FACULTAT D'INFORMÀTICA DE BARCELONA

TREBALL DE FI DE GRAU

Anàlisi i millora de l'algorisme Las Vegas Filter per a la selecció de variables en models predictius.

Autor:
Ruben Martínez Escobar

Director:
Lluís A. Belanche

Grau en Enginyeria Informàtica

Especialitat de Computació

Gener 19, 2021

Resum

Des de temps prehistòrics l'ésser humà s'ha diferenciat de la resta d'animals per la seva racionalitat, fet que impulsa a l'espècie a la recerca del coneixement. Aquest comportament únic ha estat el catalitzador principal de tota la nostra avançada evolució. Una evolució en la qual trobem importants fets que ens han marcat i canviat per sempre. Des del control del foc, l'escriptura, la llum i molts altres invents humans. Actualment ens trobem davant d'un altre descobriment el qual està tenint un gran impacte sobre tota la població, les ciències de les dades. Aquest nou paradigma que a priori pot semblar de menys precedència, està canviant la nostra evolucionada recerca del coneixement per complet i ens dota d'una capacitat d'anàlisi mai vista.

Aquest projecte elaborat a la *Universitat Politècnica de Catalunya*, tractarà la problemàtica de la selecció de variables per a la construcció de models predictius. Un dels problemes de més rellevància en les ciències de les dades i que per culpa de l'apogeu del *Big data* cada cop cobra més importància. Aquest problema serà estudiat amb un enfocament poc habitual, ja que partirem de la base d'un algorisme probabilístic que amb un seguit d'optimitzacions intentarem millorar el seu rendiment.

L'algorisme probabilístic a optimitzar és el *Las Vegas Filter* (LVF), un algorisme principalment utilitzat com a mètode de filtre i amb una mesura d'avaluació anomenada inconsistència. Durant el transcurs del projecte s'estudiaran optimitzacions ja proposades per altres autors i s'elaboraran diverses propostes de noves optimitzacions. A causa de la seva naturalesa probabilística, trobem un nivell d'aleatorietat molt alt en l'algorisme, el qual en alguns aspectes empitjora el rendiment d'aquest. Enfocarem les optimitzacions proposades en aquest projecte a resoldre aquest problema i a preservar els beneficis que ens aporta aquesta aleatorietat.

Durant el transcurs del projecte tractarem principalment el LVF com un mètode de filtre, però per a resoldre el problema de la pèrdua de precisió en els models predictius construïts amb els subconjunts de variables solució donats pel LVF, serem pioners en la introducció d'un enfocament de mètode híbrid en el LVF, el qual ens obligarà a estudiar també optimitzacions ja proposades utilitzant el LVF com a un mètode d'embolcall.

Resumen

Desde tiempos prehistóricos el ser humano se ha diferenciado del resto de animales por su racionalidad, hecho que impulsa a la especie en busca del conocimiento. Este comportamiento único ha sido el catalizador principal de toda nuestra avanzada evolución. Una evolución en la cual encontramos importantes hechos que nos han marcado y cambiado por siempre jamás. Desde el control del fuego, la escritura, la luz y otros muchos inventos humanos. Actualmente nos encontramos ante otro descubrimiento el cual está teniendo un gran impacto sobre toda la población, las ciencias de los datos. Este nuevo paradigma que a priori puede parecer de menos precedencia, está cambiando nuestra evolucionada investigación del conocimiento por completo y nos dota de una capacidad de análisis nunca vista.

Este proyecto realizado en la *Universitat Politècnica de Catalunya*, tratará la problemática de la selección de variables para la construcción de modelos predictivos. Uno de los problemas de más relevancia en las ciencias de los datos y que por culpa del apogeo del *Big data* cada vez cobra más importancia. Este problema será estudiado con un enfoque poco habitual, puesto que partiremos de la base de un algoritmo probabilístico que con una serie de optimizaciones intentaremos mejorar su rendimiento.

El algoritmo probabilístico a optimizar es el *Las Vegas Filter* (LVF), un algoritmo principalmente utilizado como método de filtro y con una medida de evaluación llamada inconsistencia. Durante el transcurso del proyecto se estudiarán optimizaciones ya propuestas por otros autores y se elaborarán varias propuestas de nuevas optimizaciones. A causa de su naturaleza probabilística, encontramos un nivel de aleatoriedad muy alto en este algoritmo, el cual en algunos aspectos empeora el rendimiento de este. Enfocaremos las optimizaciones propuestas en este proyecto a resolver este problema y a preservar los beneficios que nos aporta esta aleatoriedad.

Durante el transcurso del proyecto trataremos principalmente el LVF como un método de filtro, pero para resolver el problema de la pérdida de precisión en los modelos predictivos construidos con los subconjuntos de variables solución dados por el LVF, seremos pioneros en la introducción de un enfoque de método híbrido en el LVF, el cual nos obligará a estudiar también optimizaciones ya propuestas utilizando el LVF como un método de envoltorio.

Abstract

Since prehistoric times, human beings have been differentiated from other animals by their rationality, a fact that drives the species in search of knowledge. This unique behaviour has been the main catalyst for all our advanced evolution. An evolution in which we find important facts that have marked and changed us forever. From the control of fire, writing, light and many other human inventions. Today we are facing another discovery which is having a great impact on the whole population, the data sciences. This new paradigm, which at first sight may seem to be of lesser precedence, is changing our evolved research of knowledge completely and giving us a capacity for analysis never seen before.

This project, carried out at the *Universitat Politècnica de Catalunya*, will deal with the problem of selecting features for the construction of predictive models. This is one of the most important problems in the data sciences and one that is becoming increasingly important due to the rise of Big Data. This problem will be studied with an unusual approach, since we will start from a probabilistic algorithm that with a series of optimizations we will try to improve its performance.

The probabilistic algorithm to be optimised is the *Las Vegas Filter* (LVF), an algorithm mainly used as a filtering method and with an evaluation measure called inconsistency During the course of the project optimisations already proposed by other authors will be studied and several proposals for new optimisations will be developed. Due to its probabilistic nature, we found a very high level of randomness in the algorithm, which in some aspects worsens the performance of the algorithm. We will focus the optimizations proposed in this project to solve this problem and to preserve the benefits that this randomness brings us.

During the course of the project, we will mainly treat the LVF as a filter method, but to solve the problem of loss of accuracy in predictive models built with the subsets of solution variables given by the LVF, we will pioneer the introduction of a hybrid method approach in the LVF, which will force us to study also optimisations already proposed using the LVF as a wrapper method.

Agraïments

Primerament, agrair a en Lluís per haver-me brindat aquesta fantàstica oportunitat de poder fer el meu treball de fi de grau amb ell. Les circumstàncies han estat difícils a causa de la pandèmia, però a estat un excel·lent director del projecte i he après molt dels seus consells, sense ell aquest projecte no hauria estat possible.

Moltes gràcies a la meva família pel suport i la confiança que m'han transmès durant tot el grau, sense ells tampoc hagués estat possible aquest projecte.

Agrair també als amics coneguts a la universitat durant aquests anys de grau, sempre han estat propers i m'han ajudat en tot moment.

Per acabar, agrair també als professors tinguts durant el transcurs del grau, els quals m'han ensenyat una quantitat immensa d'increïbles coneixements.

Índex

Índex de Taules

Índex de Figures

Capítol 1

Introducció i abast

En aquest capítol s'introdueix i es contextualitza el projecte explicant els seus conceptes essencials, definint el problema i els actors que es troben implicats. També, es defineix l'abast del projecte; determinant els objectius del treball, els seus respectius requeriments i els possibles obstacles que podien aparèixer.

1.1 Definició de conceptes

Abans d'iniciar-nos amb el problema a tractar en aquest treball de fi de grau, convé definir alguns conceptes fonamentals per entendre el projecte i el seu plantejament.

1.1.1 Ciència de les dades

La ciència de les dades és un nou paradigma científic (nombrat al 1962 per John W. Tukey [1]) el qual utilitza mètodes dramàticament diferents dels del passat per al descobriment del coneixement. És un camp interdisciplinari el qual està fortament relacionat amb l'aprenentatge automàtic, mineria de dades i el big data. Aquest camp neix del concepte d'unificar el anàlisi de dades i l'estadística amb el fi d'entendre i analitzar un fenomen determinat.

Per a facilitar el context definiré breument els conceptes enumerats amb relació a les ciències de dades.

- **Aprenentatge automàtic**: És un subcamp de la ciència dels computadors que té com a objectiu l'estudi d'algorismes que construeixen models matemàtics per a conjunts de dades, amb els quals podem realitzar prediccions.
- **Mineria de dades**: És un camp interdisciplinari entre l'estadística i la ciència dels computadors el qual té com a objectiu principal descobrir patrons en grans volums de conjunts de dades.
- **Big data**: És un terme que referència als conjunts de dades de grans dimensions, els quals a causa de la seva gran mida han de ser tractats amb mètodes no tradicionals.

1.1.2 Algorisme probabilístic

Un Algorisme probabilístic és aquell algorisme el qual el seu comportament no és determinat només per la seva entrada, sinó que també està condicionat per la presa de decisions aleatòries. D'aquesta manera el resultat de l'algorisme vindrà definit per la seva entrada i per les decisions aleatòries de l'algorisme [2]. Dintre del paradigma dels algorismes probabilístics trobem dos grans grups:

- **Algorismes de Monte Carlo**: Aquest tipus d'algorismes es caracteritzen per aturar-se sempre en un temps finit, però la seva solució no sempre serà correcta. És molt important per tant fitar la seva probabilitat d'error, però cal tenir en compte que com més baixa és aquesta probabilitat, més temps d'execució de l'algorisme [3].
- **Algorismes de Las Vegas**: Aquests algorismes mai donen una resposta incorrecta, poden donar la solució correcta o bé informar que no s'ha trobat cap solució correcta. El seu temps d'execució pot arribar a no tenir fita i quantes més execucions realitzem més probabilitat existeix de trobar una solució correcta. Cal destacar que dintre d'aquest grup trobem els algorismes de Sherwood els quals sempre troben solució i és correcta [4].

Considerem un algorisme de Las Vegas o un algorisme de Monte Carlo eficient si per qualsevol de les seves entrades té un temps d'execució fitat per una funció polinòmica de la mida de l'entrada [3]. Quin és millor, Montecarlo o Las Vegas? La resposta depèn del problema a resoldre: en alguns problemes, una solució incorrecta pot ser catastròfica.

Cal remarcar que l'aleatorietat total no es pot aconseguir amb computadors. Recordem que els nostres computadors utilitzen procediments deterministes, el que significa que amb una determinada entrada amb el mateix entorn sempre s'obtindrà una mateixa sortida, per tan sempre existiran una sèrie de factors que definiran el seu resultat. El que sí que podem obtenir amb els nostres computadors és l'anomenada pseudoaleatorietat; generacions de seqüències de nombres que no presenten cap patró o regularitat aparent des d'un punt de vista estadístic, però que són generats amb algorismes deterministes. D'aquesta manera quan ens referim a aleatorietat en aquest TFG, tècnicament estarem parlant de pseudoaleatorietat.

1.1.3 Model predictiu

Un model predictiu és la principal eina utilitzada per l'anàlisi de dades predictives, aquests models són capaços de realitzar prediccions basades en patrons extrets de dades. En l'anàlisi de dades considerem una predicció com l'assignació d'un valor a qualsevol variable desconeguda, per tant les prediccions són aplicables a qualsevol camp sempre que tinguem dades suficients sobre el camp. Aquestes prediccions ajudaran a organitzacions o persones a prendre decisions [5].

Tots aquests models són entrenats per a poder realitzar aquestes prediccions mitjançant diferents tècniques d'aprenentatge automàtic amb les quals s'intentarà obtenir una predicció amb la màxima probabilitat d'encert possible.

1.1.4 Conjunt de dades

També conegut en anglès com a dataset, és una col·lecció de dades que en aquest projecte considerarem sempre tabulada, per tant en un format de taula (files i columnes). On les files representen instàncies (també conegudes com a observacions) i cada columna representa una característica (o també anomenada variable) d'aquestes instàncies. Aquestes variables les podem classificar en tres tipus genèrics segons els seu domini de valors.

- **Variable numèrica**: Variable la qual només pot prendre un valor numèric.

- **Variable categòrica**: Variable la qual només pot prendre un, d'un nombre limitat de possibles valors. A més, els seus valors no estan ordenats.
- **Variable binària**: Variable la qual només pot prendre un, de dos valors possibles.

Un conjunt de dades és l'entrada que rep qualsevol tècnica d'aprenentatge automàtic per a la construcció d'un model predictiu.

1.2 Descripció del problema

Aquest treball té la finalitat de donar una alternativa pel famós problema de la selecció de variables (en anglés, conegut com a *feature selection problem*) per al procés de construcció d'un model predictiu. Un problema molt estudiat i el qual el podem resoldre amb moltes tècniques diferents. A continuació es defineix el problema de la selecció de variables: Definim X com el conjunt de variables d'un conjunt de dades, amb una cardinalitat de $|X| = n$. Tenim dues versions del problema[6]:

- **Versió continua:** Consisteix en l'assignació de pes w_i a cada variable $x_i \in X$ de tal manera que si ordenem els pesos decreixentment, es preserva l'ordre de la rellevància teòrica de les variables al conjunt de dades. Aquesta versió permet utilitzar totes les variables en l'aprenentatge del model però amb una rellevància diferent basada en la solució obtinguda.
- **Versió binària:** Consisteix en l'elecció d'un subconjunt de les variables x que maximitzi una certa mesura relacionada amb la rellevància del subconjunt de variables. Amb aquesta versió només es selecciona el subconjunt de variables de la solució obtinguda, on totes s'utilitzen equitativament en el procés d'aprenentatge del model.

Amb una fita mínima de la mesura emprada pel càlcul de la rellevància podem transformar la solució de la versió contínua en una solució de la versió binària.

Podem declarar una formalització del problema d'aquesta manera: Sigui $\mathcal{J}$ una mesura generalitzada d'avaluació del rendiment a ser maximitzada definida com $\mathcal{J} : \mathcal{P}(X) \Rightarrow \mathbb{R}^+ \cup \{0\}$. On entendrem durant tot el projecte $\mathcal{P}(X)$ com el conjunt de les parts de X. Sigui $c(x) \geq 0$ el cost de la variable x (cost de mesura, cost de càlcul, etc.) i $c(X') = \sum_{x \in X'} c(x)$, per $X' \in \mathcal{P}(X)$. Definim també $C_X = c(X)$ com el cost de totes les variables del conjunt de dades. Finalment assumim que c es additiva, tal que $c(X' \cup X'') = c(X') + c(X'')$ [7].

Ara ja podem definir correctament el problema, en ell trobem dos escenaris diferents [7]:

- **Quan imposem un màxim pel cost total de les variables que pertanyen a la solució:** Fixem $C_0 \leq C_X$. Hem de buscar la $X' \in \mathcal{P}(X)$ de màxima $\mathcal{J}(X')$ d'entre les solucions que compleixen $c(X') \leq C_0$.
- **Quan imposem un mínim per la mesura de l'avaluació del rendiment:** Fixem $\mathcal{J}_0 \geq 0$. Hem de buscar la $X' \in \mathcal{P}(X)$ de mínim $c(X')$ d'entre les solucions que compleixen $\mathcal{J}(X') \geq \mathcal{J}_0$.

Amb aquests dos escenaris si obtenim un subconjunt de variables òptim, aquest subconjunt no té perquè ser l'única solució òptima.

En aquest TFG tractarem la versió binària d'aquest problema degut a que és la versió que millor s'adapta a la solució que es proposa. Definirem el problema basantnos en l'escenari on fixem un mínim per la mesura de l'avaluació del rendiment. Tots aquests conceptes seran justificants més endavant quan es tracti la solució que es vol donar al problema.

Per a la validació de les solucions del problema, les variables que formen el subconjunt escollit és classificaran en tres categories:

- **Variables rellevants**: Aquestes variables tenen una influència positiva en la construcció del model i el seu paper no pot ser assumit per cap altre variable.
- **Variables irrellevants**: Aquestes variables no tenen una influència positiva en la construcció del model, és a dir, no aporten cap millora al model.
- **Variables redundants**: Aquestes variables són les que poden prendre el mateix rol que una altre variable del mateix subconjunt de variables i tindran la mateixa influència en la construcció del model.

Gràcies a aquesta classificació es desenvoluparà un model avaluatiu on es ponderarà la solució en un rang de [0, 1].

1.3 Actors Implicats

Aquest projecte va dirigit clarament als investigadors del camp de les ciències de les dades en especial a les disciplines d'estadística multivariant, d'aprenentatge automàtic i de mineria de dades, ja que en totes elles apareix constantment el problema de la selecció de variables. També va dirigit als usuaris de les ciències de les dades en general.

Potencialment podria ser utilitzat per qualsevol usuari de les ciències de les dades, que vulgui utilitzar o estudiar una visió alternativa al problema de la selecció de variables.

Pel que fa els beneficiats, en primer lloc es beneficiarien els usuaris de les ciències de les dades, ja que tindrien al seu abast una solució senzilla a l'hora de seleccionar les variables per realitzar el procés d'aprenentatge d'un model predictiu. Si anem més enllà és molt difícil de predir els possibles beneficiats, a causa del fet que ara mateix les ciències de les dades es troben a tots els sectors influint a la majoria de la població mundial i a moltes empreses, per tant indirectament podrien ser beneficiats.

1.4 Objectius del projecte

L'objectiu principal d'aquest projecte és millorar l'algorisme LVF proposant noves versions d'aquest algorisme. Això significaria avançar en l'estudi d'algorismes probabilístics en l'àmbit del problema de selecció de variables. Aquest objectiu és molt genèric i el podem dividir en subobjectius per a assolir-lo:

- **Estudi de la literatura del LVF:** Realització d'un estudi exhaustiu de les investigacions realitzades anteriorment al LVF per a aprofitar informació o no repetir optimitzacions ja investigades.
- **Generació de conjunts de dades artificials:** És important generar artificialment aquests conjunts de dades per a tenir un major control i llibertat sobre ells per a una anàlisi millor.
- **Implementació d'un algorisme generador de problemes de selecció de variables:** Aquest algorisme ha de ser capaç de modificar els conjunts de dades anteriors per a afegir variables irrellevants i variables redundants correctament.
- **Selecció d'un algorisme d'aprenentatge:** Estudi de diferents algorismes de aprenentatge i selecció del que millor s'hi adeqüi al nostre perfil de dades.
- **Implementació d'un algorisme avaluador de subconjunts de variables:** Aquest algorisme ha de ser capaç d'avaluar i aportar informació sobre les variables seleccionades.
- **Implementació inicial del LVF:** Aprofitar una implementació existent i adaptar-la o realitzar una implementació des de zero de l'algorisme LVF.
- **Implementació de millores de l'algorisme LVF:** Aportació de noves idees per a noves optimitzacions de l'algorisme LVF i implementar-les.
- **Avaluació i anàlisi de les noves versions proposades del LVF:** Realitzar l'avaluació i l'anàlisi de les millores que es van implementant amb l'ajuda de l'algorisme avaluador de subconjunts de variables.
- **Implementació inicial del LVI:** Implementació d'una optimització ja estudiada del LVF anomenada Las Vegas Incremental [8], aquesta optimització és explicada en el capítol Estat de l'art, en la secció Millores proposades del LVF.
- **Implementació inicial del QBB:** Implementació d'una optimització ja estudiada del LVF anomenada Quick Branch & Bound [9]. L'optimització és explicada en el capítol Estat de l'art, en la secció Millores proposades del LVF.
- **Aplicació de millores a l'algorisme LVI:** Adaptació i aplicació de les millores desenvolupades per l'algorisme LVF a la millora ja existent LVI.
- **Aplicació de millores a l'algorisme QBB:** Adaptació i aplicació de les millores desenvolupades per l'algorisme LVF a la millora ja existent QBB.
- **Avaluació i anàlisi de les noves versions proposades del LVI i QBB:** Anàlisi de les noves versions del LVI i QBB proposades amb l'ajuda de l'algorisme avaluador de subconjunts de variables.
- **Ampliació de les propostes amb una metodologia híbrida:** Realització d'una última investigació amb la qual introduirem optimitzacions basades en metodologies híbrides i d'embolcall.

En el Capítol 10, *Planificació temporal*, s'explica en detall les fases d'aquest projecte i és concreta que inicialment es desenvoluparà un producte viable mínim el qual haurà de complir unes funcionalitats mínimes requerides. Cal remarcar que en termes de subobjectius planifiquem que el producte viable mínim assoleixi els vuit primers subobjectius definits. És a dir, els subobjectius a partir dels referents

a l'algorisme LVI es treballaran des de la base del producte viable mínim. Els altres subobjectius també s'intentaran millorar un cop desenvolupat el producte viable mínim però ja s'hauran treballat prèviament.

1.5 Requeriments

En aquesta secció s'esmentaran els requeriments principals d'aquest projecte.

1.5.1 Requeriments funcionals

Com a requeriment funcional d'aquest projecte trobem que la versió final que es presenti del LVF tingui la capacitat de proveir solucions de qualitat envers el problema de la selecció de variables.

1.5.2 Requeriments no funcionals

Seguidament es defineixen els principals requeriments no funcionals:

- Compatibilitat amb tots els conjunts de dades amb variables categòriques, variables binàries i variables numèriques discretes.
- Temps de còmput no excessivament alt.
- Fàcil entrada del conjunt de dades per a l'usuari final.
- Mínim nombre de llibreries externes utilitzades.
- Codi de fàcil enteniment per a habilitar la reutilització de la informació en futurs estudis.

1.6 Possibles obstacles i riscos

A continuació s'exposen els principals problemes que poden sorgir durant l'elaboració d'aquest projecte. Aquests problemes són obstacles i riscos que s'assumeixen que poden comportar imprevistos i contratemps al projecte.

- **Errors de disseny:** Aquests errors es poden produir a l'hora de dissenyar els algorismes o les millores, són molt greus pel fet que si no es rectifiquen a temps, poden comportar pèrdues enormes de temps.
- **Errors d'implementació:** Es donen a l'hora d'implementar els algorismes al llenguatge de programació. Poden aparèixer al compilador o el que és pitjor identificar-los per un comportament diferent de l'esperat en la fase d'anàlisi.
- **Errors d'anàlisi:** Aquests errors apareixen per a càlculs o observacions errònies en la fase d'anàlisi dels algorismes. És molt important evitar-los, ja que poden portar la investigació a conclusions errònies.
- **Risc de no trobar cap millora notable:** Al ser un projecte d'investigació es corre el risc de no trobar cap optimització que millori notablement les prestacions de l'algorisme original. No s'assoliria l'objectiu principal del projecte però el treball serviria per evitar possibles investigacions ja realitzades aquí.

- **Risc de malaltia o lesió:** Risc molt difícil de predir però que pot succeir. Afectaria principalment a la planificació horària del projecte esmentada en el Capítol 10, *Planificació temporal*. A causa del fet que incapacitaria a l'autor a seguir temporalment amb la planificació.
- **Avaria de hardware:** També és un risc molt difícil d'anticipar però que pot aparèixer. Aquest obstacle consistiria en l'avaria de l'ordinador portàtil de l'autor amb el qual desenvolupa el projecte. Aquest risc no afectaria tan directament a la planificació horària sinó que perjudicaria el pressupost econòmic, ja que es necessitaria comprar un ordinador portàtil nou.

Capítol 2

Estat de l'art

En aquest capítol s'explicaran algunes alternatives que s'han proposat per al problema de la selecció de variables i incidirem especialment en l'elecció de la solució que s'exposa en aquest treball i algunes optimitzacions ja estudiades.

2.1 Estudi solucions existents

Existeixen moltes estratègies diferents per abordar aquest problema i ha estat tema d'estudi durant molts anys, molt superficialment trobem dues alternatives genèriques quant a l'implicació de l'usuari:

- **Selecció manual:** L'usuari realitza un estudi manual per a detectar les variables rellevants que necessita el seu model. Aquest estudi pot basar-se en molts aspectes de les ciències de les dades. A continuació alguns exemples: estudi de la col·linealitat, Anàlisis de les Components Principals, Anàlisi de Correspondències Múltiples...
- **Selecció automatitzada:** L'usuari executa un algorisme de selecció de subconjunts de variables (en anglès, *Feature Subset Selection Algorithm*[6]) el qual li proporciona el subconjunt de variables rellevants.

Els mètodes emprats per al problema de la selecció de variables, els podem classificar en quatre grups diferents d'acord amb les característiques del seu criteri d'evaluació[10]:

- **Mètodes de filtre**: En anglès, *Filter methods*, aquests mètodes són coneguts per ser utilitzats en el preprocessat de les dades i són independents a l'algorisme d'aprenentatge del model. La seva complexitat computacional és baixa però no garanteixen una millora sobre l'algorisme d'aprenentatge, això és degut al fet que no basen la seva solució en el rendiment d'aquest algorisme. Són mètodes que defineixen un rànquing entre variables segons la seva correlació, Anàlisis de les Components Principals, variància, etc.
- **Mètodes d'embolcall**: En anglès, *Wrapper methods*, aquests mètodes utilitzen la precisió del model predictiu que s'utilitzarà per a avaluar el rendiment de les variables. Aquesta solució té un cost computacional molt elevat ja que per cada comprovació s'ha de realitzar el procés d'aprenentatge del model. Amb conjunts de dades d'una gran dimensionalitat aquesta solució no és viable.
- **Mètodes incrustats**: En anglès, *Embedded methods*, aquests mètodes incorporen la selecció de variables en el procés d'aprenentatge i normalment són específics per cada algorisme d'aprenentatge. Per exemple seria el cas dels arbres de decisió.

- **Mètodes híbrids**: En anglès, *Hybrid methods*, aquests mètodes van ser proposats per a combinar les millors característiques dels mètodes de filtre i d'embolcall. Són una combinació d'aquests dos mètodes, usualment s'aplica el mètode de filtre per a reduir el nombre de variables candidates i després s'aplica el mètode d'embolcall per a precisar la selecció.

Aquest treball es va decidir desenvolupar pensant en un algorisme que es basés en un mètode de filtre o un mètode híbrid, a causa del fet que cada cop els conjunts de dades tenen una dimensionalitat més i més gran. Per tant es busca una solució que pugui ser aplicada als conjunts de dades de grans dimensions i això ens fa no comptar amb els mètodes d'embolcall, ja que pel seu elevat temps d'execució no compleixen aquesta funció correctament. Pel que fa als mètodes incrustats en treballar amb ells no es posseeix una gran llibertat perquè van fortament lligats amb l'algorisme d'aprenentatge i això limitaria molt l'aplicabilitat del treball.

2.2 Algorismes de selecció de variables amb mètodes de filtre

En aquesta secció explicarem famosos algorismes de selecció de subconjunts de variables els quals el seu criteri d'avaluació està basat en mètodes de filtre. Per tant estudiarem per sobre altres alternatives al problema de la selecció de variables amb una tipologia de criteri d'avaluació similar a el nostre.

S'ha decidit no incidir en els algorismes de selecció de variables amb criteris d'avaluació basats en mètodes d'embolcall o mètodes incrustats a causa del fet que s'allunyen molt del nostre camp d'estudi (el qual ja és prou gran) i la teoria divergeix molt a la tractada en aquest projecte. Tot i això, molts dels algorismes exposats poden convertir-se en algorismes basats en mètodes d'embolcall adaptant la seva mesura d'avaluació.

2.2.1 Algorismes SFG/SBG

Aquests són dos algorismes clàssics molt generalitzats. El SFG (de l'anglès, *Sequential Forward Generation*) iterativament afegeix variables a un subconjunt de variables inicial, no repetint mai les que ja ha afegit i intenta millorar una mesura d'avaluació $\mathcal{J}$. Aquest procés es repetirà fins que es deixi de millorar la mesura d'avaluació $\mathcal{J}$. A continuació trobem el pseudocodi pel SFG.

Algorisme 1: SFG

Entrada:
$S(X)$ - Conjunt de dades S descrit per X
$\mathcal{J}$ - Mesura d'avaluació
Sortida:
X' - Subconjunt de variables resultat

$X' := \emptyset$
repeat
  $x' := argmax\{\mathcal{J}(S(X' \cup \{x\}))|x \in X \backslash X'\}$
  $X' := X' \cup \{x'\}$
until *cap millora en* $\mathcal{J}$ ***or*** $X' = X$

El SBG (de l'anglès, *Sequential Backward Generation*) es basa en el mateix concepte que el SFG, però s'inicia amb totes les variables i iterativament es van eliminant aquestes variables del conjunt de variables inicial. El procés serà repetit fins que es deixi de millorar o s'empitjori la mesura d'avaluació $\mathcal{J}$. Seguidament s'exposa el pseudocodi pel SBG.

Algorisme 2: SBG

Entrada:
$S(X)$ - Conjunt de dades S descrit per X
$\mathcal{J}$ - Mesura d'avaluació
Sortida:
X' - Subconjunt de variables resultat

$X' := X$
repeat
  $x' := argmax\{\mathcal{J}(S(X'\backslash\{x\}))|x \in X'\}$
  $X' := X'\backslash\{x'\}$
until *cap millora en* $\mathcal{J}$ ***or*** $X' = \emptyset$

A part de poder-se aplicar en mètodes de filtratge també es poden aplicar a mètodes d'embolcall, on la mesura d'avaluació $\mathcal{J}$ és la precisió d'algun model predictiu. S'identifiquen com W-SFG i W-SBG, on la W indica el terme anglès *Wrapper*, embolcall.

Doak[11], l'any 1992 va reportar que el SBG tendia a donar un millor rendiment envers el SFG quan el conjunt de dades tractat tenia un nombre reduït de variables, Aha i Bankert[12] van arribar a una conclusió similar exposant que quan el conjunt de dades tenia poques variables rellevants el SBG tenia un millor rendiment. Contràriament quan trobem un gran nombre d'instàncies rellevants és preferible l'ús del SFG.

El principal problema amb aquests algorismes és el seu elevat temps de còmput, ja que són d'ordre exponencial, per tant, presenten una difícil aplicació en conjunts de dades de grans volums.

2.2.2 Algorisme RELIEF

L'algorisme RELIEF (Kira i Rendell, 1992 [13]) funciona exclusivament per a mètodes de filtre. L'algorisme selecciona aleatòriament una instància I del conjunt de dades S, determina el *near hit* (la instància més pròxima a I de la mateixa classe que I) i el *near miss* (la instància més pròxima a I d'una altra classe que I). La idea principal és que la variable és més rellevant per I quan més separa I del seu *near miss* i quan menys separa I del seu *near hit*. Per tant, l'algorisme calcularà per totes les instàncies aleatòries aquestes diferències de distàncies per a totes les variables i desarà els resultats en un vector, el qual representarà mitjançant pesos la rellevància de cada variable. A continuació es mostra l'algorisme RELIEF.

Algorisme 3: RELIEF

Entrada:
$S(X)$ - Conjunt de dades S descrit per X on $n = |X|$
d - Mesura de distància
p - Persentatge de mostres
Sortida:
w - Vector amb els pesos de les variables

$m := p|S|$
Inicialització de $w[]$ *a* 0
repeat m **times**
  $I := InstanciaAleatoria(S)$
  $I_{nh} := NearHit(I, S)$
  $I_{nm} := NearMiss(I, S)$
  for each $i \in [1..n]$ **do**
    $w[i] := w[i] + d_i(I, I_{nm})/m - d_i(I, I_{nh})/m$
  end
end

Aquest algorisme ha estat força investigat i es va arribar a la conclusió que no realitzava una bona discriminació entre les variables redundants i seleccionava les variables correlacionades en comptes de les rellevants, per tant la solució podia trobar-se lluny de l'òptima (Dash et al., 1997 [14]). Té un cost computacional de $\mathcal{O}(m \cdot |X| \cdot D)$, on m és el nombre de repeticions, $|X|$ el nombre de variables i D el cost de la mesura de distància.

Es va proposar una versió millorada d'aquest algorisme (RELIEF-F[15]) on se seleccionaven les k instàncies més similars (pertanyents a la mateixa o diferent classe, respectivament) i es computaven les seves mitjanes.

2.2.3 Algorisme FOCUS

L'algorisme FOCUS[16] utilitza com a mesura d'avaluació $\mathcal{J}$ la consistència, una mesura d'avaluació la qual serà explicada en detall en la secció Las Vegas Filter, d'aquest capítol. Aquest algorisme comença avaluant cada subconjunt de variables format per una única variable del conjunt de variables total. Posteriorment avalua tots els subconjunts de variables possibles formats per dues variables i així successivament. L'algorisme s'aturarà quan trobi un subconjunt de variables el qual tingui un grau de consistència superior al llindar definit.

El mateix autor, H. Almuallim, va proposar una segona versió anomenada FOCUS-2[17] la qual introduïa un mètode per a podar els subespais de solucions candidates que ja no podien arribar al llindar del grau de consistència, aquesta millora és aplicable gràcies a la propietat de monotonia que presenta la mesura d'avaluació, consistència. A continuació trobem el pseudocodi de l'algorisme FOCUS.

Algorisme 4: FOCUS

Entrada:
$S(X)$ - Conjunt de dades S descrit per X
$\mathcal{J}$ - Mesura d'avaluació (consistència)
$\mathcal{J}_0$ - Llindar mínim d'acceptació de $\mathcal{J}$
Sortida:
X' - Subconjunt de variables resultat

```
for i ∈ [1..|X|] do
    for each X' ⊂ X, with |X'| = i do
        if J(S(X')) ≤ J_0 then
            stop
        end
    end
end
```

2.2.4 Algorismes B&B/ABB

L'algorisme B&B (de l'anglès, *Branch and Bound)*), és un algorisme de cerca òptim, el qual donat un llindar β (especificat per a l'usuari), realitza una cerca en profunditat en l'arbre de possibles solucions, iniciant-se amb totes les variables i en baixar un nivell de profunditat a l'arbre, elimina una de les variables de la solució actual, així successivament. L'algorisme segueix una estratègia d'eliminació per a reduir l'espai de solucions; com que l'algorisme utilitza la consistència com a mesura d'avaluació $\mathcal{J}$, aprofita la monotonia d'aquesta funció per a descartar les branques de l'arbre on el node arrel presenti una consistència menor a β.

L'algorisme ABB (de l'anglès, *Automatic Branch and Bound*) és una variant de l'algorisme B&B la qual automatitza la selecció del paràmetre llindar. Aquesta automatització es basa en el fet que l'algorisme selecciona el grau de consistència que té el conjunt de dades amb totes les variables com a paràmetre llindar. Gràcies a la monotonia que presenta la consistència, aquest valor serà el mínim que es pugui aconseguir amb el conjunt de dades, fet que no significa que un subconjunt de variables més reduït que el conjunt total no pugui tenir aquest mateix valor. Seguidament s'exposa l'algorisme ABB.

Algorisme 5: Automatic Branch and Bound (ABB)

Entrada:
$\mathcal{J}$ - la mesura d'avaluació (monòtona)
$S(X)$ - una mostra S descrita pel conjunt de variables X
Sortida:
L - Millors solucions trobades

Procediment recursiu:
Q := $\emptyset$
for $x \in X$ **do**
  $enqueue(Q, \{X - x\})$
end
while *not empty(Q)* **do**
  $X' := dequeue(Q)$
  if $validar(X')$ **and** $\mathcal{J}(S(X')) \leq \mathcal{J}_o$ **then**
    $L' = inserir(L', X')$
    $ABB(S(X'), \mathcal{J}, L')$
  end
end

Inicialització:
$\mathcal{J}_o := \mathcal{J}(S(X))$
$L' := X$
$ABB(\mathcal{J}, S(X), L')$ `// Crida inicial al procediment`
k :=mida del subconjunt de variables més petit de L'
L := elements de L' amb mida k

La funció *validar(X')* és l'encarregada de validar si el node ha estat podat o en cas contrari és vàlid. En la implementació original, les solucions es representen com una cadena de bits de la mida del nombre de variables del conjunt de dades, on 1 indica que la variable d'aquella posició es troba al subconjunt i 0 altrament. La validació es porta a terme comprovant la distància de Hamming entre la solució que representa el node fill a comprovar i les dels nodes podats. Si la distància de Hamming amb qualsevol node podat és d'1, el node fill a comprovar és fill d'algun node podat per tant és un node que hem de podar.

L'inconvenient d'aquests algorismes són els seus elevats costs de còmput pel fet que són exponencials, i ens veiem limitats en les mides dels conjunts de dades on aplicar-los. Dash i Liu[18] exposen que l'algorisme FOCUS és eficient quan trobem $|X_R|$ baix, en canvi l'algorisme ABB és eficient quan $|X| - |X_R|$ és petit, on $|X|$ és el nombre de variables totals i $|X_R|$ el nombre de variables rellevants.

2.3 Las Vegas Filter

En aquesta secció explicarem amb molta més profunditat i detall l'algorisme LVF. El LVF (*Las Vegas Filter*) és l'algorisme base que prendrà aquest treball per a analitzar possibles aplicacions de millores en ell. Com el seu nom indica és un algorisme probabilístic de tipus Las Vegas, però no es troba dins el grup d'algorismes de Sherwood, ja que no sempre la solució que proposa és la correcta.

Aquest algorisme genera de manera aleatòria i repetidament subconjunts de variables i computa la seva inconsistència. La finalitat d'aquest algorisme és trobar el subconjunt de variables de mida més reduït amb el qual es pugui assolir la

inconsistència mínima que pot tenir el conjunt de dades total o el predefinit per l'usuari.

Continuarem amb la notació definida a la secció 1.2, *Descripció del problema*, del Capítol 1, *Introducció i abast*. Definim S com el nostre conjunt de dades, la inconsistència a X' bé definida com dues instàncies de S les quals són equivalents quan només considerem les variables que pertanyen a X' i pertanyen a dues classes diferents, aquestes classes són dues categories (també anomenades modalitats) diferents d'una variable que no pertany al subconjunt X' però si al conjunt X. [19] D'aquesta manera podem definir el comptatge de la inconsistència d'una instància $A \in S$ de tal manera:

$$CI_{X'}(A) = X'(A) - \max_k X'_k(A)$$

on $X'(A)$ és el nombre d'instàncies de S iguals que A utilitzant només les variables que pertanyen a X'. Entenem doncs $X'_k(A)$ com el nombre d'instàncies de S de la classe (com l'hem definit en el cas de dues instàncies) k igual a A utilitzant només les variables contingudes en X' [20]. Finalment podem declarar el grau d'inconsistència d'un subconjunt de variables en una mostra S de tal manera:

$$I(X') = \frac{\sum_{A \in S} CI_{X'}(A)}{|S|}$$

Per tant obtenim el següent llindar $I(X') \in [0, 1)$.

Cal remarcar que a causa de la naturalesa de la mesura d'avaluació (el grau d'inconsistència) aquest algorisme només pot treballar amb variables categòriques, variables binàries i variables numèriques discretes (variables que no tenen un nombre infinit de valors entre dos valors qualsevol). Per tant, si es vol treballar amb variables numèriques contínues amb aquest algorisme, les variables s'hauran de discretitzar abans.

Un altre requisit que imposa la mesura d'avaluació (el grau d'inconsistència), és la no utilització de variables identificadores d'instàncies, és a dir, aquelles variables les quals amb únicament elles en el conjunt de dades, garanteixen que el conjunt de dades no té inconsistència. Òbviament, aquest tipus de variables són irrellevants per al classificador. Per tant, el problema pot ser solucionat eliminant aquest tipus de variables. Si no tenim un coneixement previ, amb una execució del LVF podrem detectar aquest tipus de variables i posteriorment eliminar-les.

A continuació, és mostra l'algorisme de Las Vegas Filter en format de pseudocodi, cal tenir en compte que la funció $SubconjuntAleatori(X)$ genera un subconjunt aleatori de variables del conjunt X i que la funció $append(L, X')$ afegeix a la llista L el subconjunt X'. [7]

Algorisme 6: Las Vegas Filter

Entrada:
max - el nombre màxim d'iteracions
$\mathcal{J}$ - la mesura d'avaluació
$S(X)$ - una mostra S descrita pel conjunt de variables X
Sortida:
L - Llista de les solucions equivalents trobades

```
L := []
Best := X
J_o := J(S(X))
repeat max times
    X' := SubconjuntAleatori(X)
    if J(S(X')) ≥ J_o then
        if |X'| < |Best| then
            Best := X'
            L := X'
        end
        else
            if |X'| = |Best| then
                L := append(L, X')
            end
        end
    end
end
```

Com podem observar és un algorisme força senzill i amb molt potencial de millora. Un altre aspecte positiu que té per a la seva elecció és la capacitat de poder regular el nivell de bondat de la solució, el qual el podem especificar fitant la inconsistència d'acceptació, també podem controlar el seu temps d'execució fàcilment alterant el paràmetre max que delimita el nombre d'iteracions a realitzar. Aquest paràmetre max també influirà en el cost computacional de l'algorisme, ja que vindrà descrit per $\mathcal{O}(max \cdot U)$, on entenem U com el cost computacional utilitzat per al càlcul de la mesura d'avaluació $\mathcal{J}$, a causa de que és el procediment que té un cost computacional més alt trobat a dins del bucle i max com el nombre d'iteracions realitzades pel LVF.

És important remarcar que el LVF pertany al conjunt d'algorismes anomenats *anytime algorithms*[21], algorismes que la qualitat de la seva solució millora gradualment quan augmenta el seu temps de computació, per tant el paràmetre max també condiciona la bondat del resultat. Un altre aspecte a favor que té el LVF, és que quan troba una possible solució la pot mostrar per pantalla, una característica beneficiosa, ja que podem obtenir solucions pròximes a l'òptima mentre l'algorisme segueix treballant per trobar la solució òptima.

Totes aquestes propietats beneficioses exposades fan d'aquest algorisme un molt bon candidat a possibles millores, i és per aquest motiu que s'ha preferit utilitzar aquest algorisme com a base d'aquest projecte i no els anteriors exposats.

2.4 Millores proposades del LVF

Durant aquesta secció exposarem algunes de les millores que ja han estat proposades per a millorar el rendiment de l'algorisme LVF o adaptar-lo a certes situacions. Els costos computacionals d'elles es debatran en el capítol 7, *Millores finals del LVF*, ja que van estretament lligats amb la mesura d'avaluació emprada.

2.4.1 Las Vegas Incremental

L'algorisme *Las Vegas Incremental* (LVI), és una millora de l'algorisme LVF la qual es basa en el fet que no és necessari utilitzar totes les instàncies del conjunt de dades S per a avaluar la consistència. L'algorisme s'inicia amb una porció S_o de les instàncies totals de S, si el LVF utilitzant únicament aquesta porció S_o troba una solució suficientment bona, la qual també dona bons resultats amb la resta d'instàncies de S, para. Si no és així, l'algorisme afegeix a S_o les instàncies que pertanyen a $S \backslash S_o$ que produeixen la inconsistència. Es repetirà el procés amb aquest nou subconjunt S_o i s'iterarà fins que és trobi un subconjunt suficientment bo.

Aquesta millora pot utilitzar qualsevol mesura d'avaluació, però en aquest projecte es treballarà únicament amb aquesta millora definint com a mesura d'avaluació $\mathcal{J}$, la inconsistència. A continuació s'exposa l'algorisme.

Algorisme 7: Las Vegas Incremental (LVI)

Entrada:
max - el nombre màxim d'iteracions
$\mathcal{J}$ - la mesura d'avaluació (inconcsistència)
$S(X)$ - una mostra S descrita pel conjunt de variables X
p - percentatge de les instàncies utilitzada inicialment
Sortida:
X' - Millor solució trobada

$\mathcal{J}_o := \mathcal{J}(S(X))$
$S_o = PorcioInicial(S, p)$
$S_f = S \backslash S_o$
repeat forever
  $X' := LVF(max, \mathcal{J}, S_o(X))$
  if $\mathcal{J}(S(X')) \leq \mathcal{J}_o$ **then**
    return X'
  end
  else
    $C := \{$ *elements de* S_f *que provoquen inconsistencia, utlitzant* $X'\}$
    $S_o := S_o \cup C$
    $S_f := S_f \backslash C$
  end
end

Els autors de l'algorisme LVI (Liu i Setiono [8]) reporten que l'efectivitat d'aquesta millora és més obvia quan el conjunt de dades té una mida gran. En conjunts de dades amb menys de 100 instàncies no s'aprecia una millora, per culpa dels *overheads* que aporta aquesta millora. En canvi, en incrementar aquest nombre d'instàncies, cada cop s'evidencia més la millora.

Els autors també reporten que el conjunt de dades incial S_o no pot tenir una mida massa petita ni massa gran. Si S_o és massa petit, després de la primera iteració s'hauran trobat moltes instàncies que provoquen inconsistència i s'hauran afegit a S_o, el que alentirà el funcionament de l'algorisme. En canvi, si és massa gran l'estalvi computacional, no serà massa evident. Els autors suggereixen utilitzar el 10% de les instàncies totals de S per a S_o o un valor proporcional al nombre de variables de S[8].

2.4.2 Quick Branch and Bound

L'algorisme *Quick Branch and Bound*[9] (QBB) és un algorisme híbrid compost pel LVF i per l'ABB (algorisme exposat en aquest capítol en la secció anomenada, *Algorismes de selecció de variables amb mètodes de filtre*). La principal idea d'aquest algorisme, és utilitzar el LVF per a trobar bons subconjunts de variables per a iniciar una cerca més exhaustiva amb l'ABB. A continuació, es descriu l'algorisme.

Algorisme 8: Quick Branch and Bound (QBB)

Entrada:
max - el nombre màxim d'iteracions
$\mathcal{J}$ - la mesura d'avaluació
$S(X)$ - una mostra S descrita pel conjunt de variables X
p - percentatge d'ús del LVF
Sortida:
X' - Millor solució trobada

$\mathcal{J}_o := J(S(X))$
$max_{LVF} := \lfloor max \times (p/100) \rceil$
$max_{ABB} := max - max_{LVF}$
$X' := LVF(max_{LVF}, \mathcal{J}, S(X))$
$X' := ABB(S(X'), max_{ABB}, \mathcal{J})$

Els autors[9] d'aquest algorisme van reportar que el QBB podia ser en general, més eficient que el LVF, el FOCUS i l'ABB en termes de mitjanes de temps d'execució i solucions seleccionades. També, Dash i Liu[22] reporten que la repartició equivalent de temps d'execució entre el LVF i l'ABB és una solució robusta pel que fa al llindar de transacció entre un algorisme i un altre, esmenten que normalment utilitzant aquest llindar d'equivalència s'aconsegueixen les millors solucions.

2.4.3 Las Vegas Wrapper

Aquesta versió del LVF anomenada LVW (*Las Vegas Wrapper*[23]) utilitza l'algorisme LVF per a generar les possibles solucions candidates i utilitza com a mesura d'avaluació $\mathcal{J}$ la precisió d'encert d'un model predictiu escollit, el qual serà construït per cada subconjunt de variables candidat.

En conseqüència el temps de còmput d'aquest algorisme serà molt superior al del LVF perquè per cada iteració haurà de construir un model predictiu. Tot i això, amb aquesta versió podem declarar un llindar mínim de precisió el qual sabrem que sempre serà assolit per a la solució del LVW, una característica que no posseeix el LVF.

2.5 Beneficis del problema de selecció de variables

Gràcies a l'apogeu de les ciències de les dades cada cop trobem més conjunts de dades amb unes dimensionalitats més grans. És molt important remarcar que no sempre més és millor, ja que tenir més variables implica que cada cop trobem més variables irrellevants i redundants en ells. Per tant més pot significar menys [24].

A continuació exposo els principals beneficis d'una bona reducció de variables[25]:

- **Facilitar la visualització de les dades:** Al reduir la dimensionalitat de les dades es poden reconèixer més fàcilment les tendències de les dades. Aquest fet també pot resultar que els algorismes d'aprenentatge generin models amb la mateixa validesa (o millors) però molt més simples que amb totes les variables.
- **Reduir els requisits de mesura i emmagatzematge:** Al requerir menys variables podem estalviar costos, temps de mesurament (en alguns sectors poden ser molt elevats) i emmagatzematge d'aquestes dades.
- **Reduir temps del procés d'entrenament del model:** Amb conjunts de dades més petits els temps d'execució de les fases d'aprenentatge i de classificació es veuen molt millorades.
- **Millorar el rendiment del model predictiu:** La precisió del classificador es pot veure incrementada com a resultat d'una bona selecció de variables, pel fet que reduïm el soroll que produeixen les variables enganyoses.

Capítol 3

Metodologia i rigor

En aquest capítol es tractaran conceptes molt importants a l'hora de definir l'execució d'un projecte i assolir una bona organització, es tracten de la metodologia de treball, les eines de desenvolupament i el mètode de validació. També, al final del capítol, s'introdueix una sinopsi del projecte per a facilitar al lector el seguiment del fil de la investigació.

3.1 Metodologia de treball

Quant a la metodologia de treball necessitem escollir una que s'adeqüi a un plaç de temps escàs, ja que el projecte no durarà més de set mesos. També aquesta metodologia ha de permetre tenir un contacte regular amb el director del treball per a un correcte seguiment.

La metodologia escollida és Scrum[26], una metodologia de desenvolupament àgil que ens brinda les necessitats descrites anteriorment, també flexibilitat i adaptabilitat als possibles canvis que apareguin en el projecte durant la seva execució.

Aplicarem una aproximació a aquesta metodologia la qual es basarà en reunions setmanals entre l'autor i el director del treball via Google Meet[27] per a aclarir possibles dubtes i planificar les noves tasques, també d'aquesta manera el director podrà tenir un seguiment exhaustiu del treball.

3.2 Eines de desenvolupament

En aquesta secció esmentaré les eines principals utilitzades durant el projecte:

- **Llenguatge de programació:** Pel llenguatge de programació es va decidir utilitzar R[28] a causa de l'experiència prèvia amb ell de l'autor i la seva bona adaptabilitat al projecte. R és un software lliure el qual la seva primera versió va ser llençada l'any 1993 [29]. Aquest llenguatge de programació i entorn és perfecte per computació estadística i gràfics[28] i té una forta orientació a objectes.[30]

- **Entorn integrat de desenvolupament:** També es va decidir seleccionar RStudio [31] per l'experiència prèvia de l'autor i l'excel·lent adaptabilitat amb el llenguatge de programació. RStudio és un entorn integrat de desenvolupament de software lliure per al llenguatge de programació R el qual ens aporta una bona quantitat d'eines per facilitar-nos la programació, com ara una consola, un editor de sintaxi, un visualitzador de gràfics, etc [32].

- **Sistema de tipografia:** Per la realització de la documentació es va decidir utilitzar el conjunt de macroinstruccions LaTeX[33] per al sistema de tipografia Tex[34]. Això és degut al fet que LaTeX aporta una gran qualitat tipogràfica al document i un molt bon nivell de formalisme.
- **Software planificador de projectes:** Per a l'edició del diagrama de Gantt es va decidir utilitzar el servei web Gantter[35]. Es va decidir utilitzar aquesta alternativa gràcies a la fàcil edició del diagrama i a l'alt nivell estètic resultant que comporta un fàcil enteniment del diagrama.

3.3 Mètode de validació

Gràcies a la metodologia de treball seleccionada, el director podrà avaluar els progressos realitzats en el projecte en les reunions setmanals de seguiment i d'aquesta manera, també podrà comprovar el correcte assoliment de la planificació esmentada al Capítol 10, Planificació temporal.

Aquesta tipologia de validació tan periòdica, ens assegura una excel·lent validació durant tot el transcurs del projecte i també proporciona a l'autor de més temps de reacció de cara a possibles errors, ja que en realitzar aquestes validacions entre períodes reduïts de temps és molt més fàcil detectar abans els errors i discutir-los.

3.4 Sinopsi

En aquesta secció introduirem amb una vista general els punts principals que seguirà aquest treball de fi de grau amb la finalitat de què el lector segueixi més fàcilment el fil de la investigació. Dividirem aquesta explicació en les diferents fases que seguirà el projecte.

3.4.1 Treball previ

Aquesta fase va ser la primera que és dur a terme en el treball de fi de grau, en ella es va realitzar un exhaustiu estudi de la literatura del LVF i dels algorismes de selecció de variables basats en mètodes de filtre. D'aquesta manera l'autor obtenir un bon coneixement del domini del problema.

Posteriorment en aquesta fase, es va decidir que els conjunts de dades que serien utilitzats per a l'estudi de les millores que plantejaríem inicialment del LVF serien d'origen artificial i els generaríem nosaltres. Aquest fet, ens dotaria d'una millor capacitat d'estudi sobre les millores, ja que podríem classificar fàcilment les variables d'aquests conjunts de dades en variables rellevants, irrellevants o redundants i saber quines variables se seleccionen quan executem les millores.

3.4.2 Fase inicial

Després de la maduració dels conceptes principals de la fase de treball previ, es va començar la fase inicial construint els conjunts de dades artificials. Com van ser construïts per nosaltres, el treball explica aquest procés de construcció i es detallen els resultats.

Per a un tractament equilibrat per a totes les versions del LVF, es va definir una metodologia d'avaluació de millores que es seguirà estrictament en totes les experimentacions. En ella es defineix tot el procés d'avaluació d'una modificació del LVF qualsevol, també s'introdueixen conceptes claus com els classificadors que s'empraran durant el projecte per al càlcul de la *accuracy* dels models predictius generats amb els subconjunts de variables solucions de les diferents versions del LVF.

També, en aquesta metodologia d'avaluació, es defineix l'indicador *score* el qual utilitzarem per a mesurar la validesa dels subconjunts de variables dels conjunts de dades artificials. Un cop ja vam tenir definida aquesta metodologia, vam començar a estudiar, implementar i avaluar les primeres millores del LVF.

Aquestes millores inicials del LVF seguiran un procés incremental, és a dir, s'afegiran petites modificacions a l'algorisme base LVF, s'avaluaran mitjançant la metodologia d'avaluació definida i finalment es decidirà si afegir la modificació a la solució incremental, la qual serà presentada com a la versió final d'aquestes millores inicials. Algunes d'aquestes millores seran excloents entre elles, en conseqüència, haurem d'escollir les que ens aportin un millor rendiment.

L'objectiu principal d'aquestes modificacions és incrementar el rendiment de l'algorisme LVF. Aquest objectiu s'intentarà assolir equilibrant l'arbitrarietat de l'algorisme i la cerca en espais de solucions prometedors, i així obtenir un algorisme amb una variabilitat inferior i per tant més robust.

3.4.3 Fase intermèdia

Un cop seleccionada la millora final presentada en les millores del LVF inicials, donarem un enfocament diferent a l'estudi i s'iniciarà la fase intermèdia. Perquè els resultats del nostre projecte tinguin més credibilitat, a partir d'aquesta fase utilitzarem conjunts de dades reals.

En conseqüència, en iniciar aquesta fase intermèdia cercarem set conjunts de dades reals que reuneixin característiques diferents quant a dimensions, tipologia de variables, àrea del conjunt de dades, etc. Els quals emprarem en l'estudi de les millores finals del nostre projecte. Seleccionarem intencionadament alguns conjunts de dades amb variables contínues per a analitzar el comportament de les millores utilitzant un bon mètode de discretització en aquests conjunts de dades, ja que a causa de la naturalesa de la mesura d'avaluació que utilitzem (la inconsistència) no podem tractar amb variables contínues directament.

Aquestes anomenades millores finals partiran de la modificació seleccionada en l'estudi de les millores inicials i estudiarem el seu comportament amb conjunts de dades reals. Es realitzarà una última millora a la versió del LVF perquè millori el seu rendiment amb aquests conjunts de dades més complexes. Quan ja tinguem la versió definitiva, s'aplicaran diferents millores del LVF ja existents a ella, en concret al QBB i al LVI. Per tant, tindrem la possibilitat d'observar el comportament de la nostra millora del LVF combinada amb altres millores ja existents.

En aquesta fase s'analitzaran els resultats finals de les versions tractades en ella, i es denotaran les diferències en el comportament entre les millores ja existents i la que proposem.

3.4.4 Fase final

Amb la finalitat de millorar l'*accuracy* de les millores finals presentades, desenvoluparem una última investigació la qual estudiarà diferents optimitzacions situacionals per a l'algorisme, aquestes optimitzacions ja no tindran un enfocament basat en un mètode de filtre, sinó que estaran basades en mètodes híbrids i d'embolcall. Compararem l'enfocament híbrid que proposem amb l'enfocament d'embolcall tradicional en el LVF, conegut com a *Las Vegas Wrapper*.

Per a finalitzar, s'analitzaran els resultats de les noves millores i es compararan amb les versions basades en mètodes de filtre. L'objectiu a tractar serà el guany d'*accuracy* respecte a una reducció de variables inferior i a un temps d'execució molt més elevat.

Capítol 4

Conjunts de dades

En aquest capítol s'expliquen tots els conjunts de dades utilitzats en el transcurs d'aquest projecte per a l'experimentació de les millores estudiades del LVF. Es donarà molta èmfasi en aquesta explicació perquè el lector comprengui perfectament els conjunts de dades empleats, el perquè de la seva utilització i d'aquesta manera tindrà un millor context en les experimentacions desenvolupades.

4.1 Requeriments dels conjunts de dades

Amb la finalitat de facilitar la realització i la posterior anàlisi de l'experimentació de les diferents millores implementades del LVF, vam decidir establir un conjunt de requeriments, els quals han d'assolir tots els conjunts de dades que intervinguin en totes les experimentacions realitzades en el projecte.

D'aquesta manera, en tenir definits un seguit de requeriments generals per als conjunts de dades podem avaluar amb una metodologia similar els resultats dels experiments en els diferents conjunts de dades. També aquests requeriments ens asseguren el correcte funcionament del conjunt de dades amb qualsevol versió de l'algorisme LVF implementada que es vulgui testejar.

A continuació, s'exposen els requeriments que han de complir els nostres conjunts de dades:

- **Control de la tipologia de les variables:** Únicament poden contenir variables binàries, variables categòriques o bé variables discretes, a causa de la mesura d'avaluació que utilitza el LVF, el grau d'inconsistència.
- **Control de les variables rellevants:** S'ha de conèixer quines són les variables rellevants del nostre conjunt de dades. Aquelles que milloren el model predictiu.
- **Control de les variables irrellevants:** També s'ha de conèixer quines són les variables irrellevants del nostre conjunt de dades. Aquelles que no milloren el model predictiu.
- **Control de les variables redundants:** De les variables rellevants del nostre conjunt de dades també s'ha de conèixer si altres variables rellevants del conjunt de dades aporten una informació molt similar o igual per a detectar així variables potencialment redundants.
- **Control de la dimensionalitat:** És important comptar amb conjunts de dades de diferents dimensionalitats per a un millor estudi de l'experimentació. Però

en aquest TFG ens veiem limitats pel que fa a l'experimentació de conjunts de dades de dimensionalitats molt grans, pel fet que el temps d'execució de l'algorisme en ordinadors de gamma mitjana és massa elevat.

- **Estandardització del nom de les variables:** Per a facilitar el desenvolupament de l'algorisme encarregat d'avaluar la validesa de les solucions proposades per les diferents versions del LVF, totes les variables dels conjunts de dades hauran de seguir el mateix estàndard en el seu nom. Aquest estàndard el definim de la següent manera; tota variable rellevant s'anomenarà com x_i on la i serà substituïda per un dels valors que representa i no podran repetir-se. Definim i com $\{i \in \mathbb{N} : 1 \leq i \leq n_r\}$, on n_r és el nombre de variables rellevants. Les variables irrellevants seran anomenades com irr_j on definim j com $\{j \in \mathbb{N} : 1 \leq j \leq n_{irr}\}$ entenen n_{irr} com el nombre de variables irrellevants. La notació per les variables redundants seguirà el mateix patró i serà reb_u on definim u com $\{u \in \mathbb{N} : 1 \leq u \leq n_{red}\}$ i n_{red} com el nombre de variables redundants. En el cas de les variables redundants, després de la notació esmentada s'afegirà un punt i el nom de la variable que redunda. Per exemple, $red_1.x_2$ indica que és la variable redundant número 1 i que repeteix a la variable $x2$. Per acabar, la variable objectiu serà anomenada com *target*.

- **Classificació de la variable objectiu**:S'ha de poder acomplir una tasca de classificació per a la variable objectiu del conjunt de dades. És a dir, amb l'ajuda de les altres variables (variables predictores) poder elaborar un model classificador per a classificar les instàncies en les classes de la variable objectiu. No tractarem regressions en aquest projecte.

- **Posició de la variable objectiu:** Per a facilitar el desenvolupament de la funció que calcula el grau d'inconsistència dels conjunts de dades, la variable objectiu del conjunt de dades es trobarà sempre en l'última posició en la fila de variables.

4.2 Conjunts de dades inicials

Els conjunts de dades que es descriuen en aquesta secció van ser desenvolupats durant la fase de treball previ del TFG, aquesta fase es trova descrita en el Capítol 10, *Planificació temporal*.

La primera qüestió que ens vam plantejar d'acord amb la selecció d'aquests conjunts de dades per la fase inicial de l'estudi, va ser el seu origen. Ja que hi trobem tres possibilitats:

- **Conjunt de dades real**: Els valors de les variables contingudes en el conjunt de dades s'han extret d'observacions verídiques sobre un cert domini.

- **Conjunt de dades sintètic**: Els valors de les variables contingudes en el conjunt de dades s'han generat de manera artificial.

- **Conjunt de dades híbrid**: Alguns dels valors de les variables contingudes en el conjunt de dades s'han extret d'observacions verídiques sobre un cert domini i altres valors han estat generats artificialment.

Amb un conjunt de dades real és difícil adaptar la mida a les nostres necessitats i el control de les variables rellevants, irrellevants i redundants requereix un previ

estudi del conjunt de dades. En canvi, si desenvolupem un conjunt de dades des de zero i el generem en base els nostres requeriments, tindrem un control molt més exhaustiu sobre les nostres variables i el dotarem de la dimensionalitat que nosaltres considerem millor per a l'experimentació.

Cal remarcar que es té consciència que els conjunts de dades sintètics simples no es poden extrapolar en contexts reals, però com s'ha comentat en el Capítol 3, *Metodologia i rigor*,aquesta fase inicial de l'estudi té la finalitat d'analitzar i realitzar millores incrementals en l'algorisme base LVF, però aquests resultats no seran els finals del projecte, per tant podem realitzar aquesta primera fase amb conjunts de dades sintètics, els quals ens donen un major control i enteniment del comportament de l'algorisme però no presentar-los com a resultats vàlids per al final del projecte. Per tant, utilitzarem un origen sintètic pels nostres conjunts de dades inicials.

4.2.1 Procés de creació

Amb aquesta primera qüestió ja resolta, vam definir el procés de creació dels diferents conjunts de dades que ens ajudarien en l'estudi de les millores del LVF a la fase inicial. Aquest procés el podem veure representat en la figura 4.1 i consta dels següents processos:

- **Disseny del conjunt de dades base:** Primerament establirem les característiques que ens agradaria que el nostre conjunt de dades complís. Un cop definides aquestes característiques, pensarem el nombre de variables del conjunt de dades el qual interpretem com a m, el seu respectiu domini i el nombre d'instàncies totals que interpretem com a n. Per a finalitzar, estudiarem una funció la qual aplicar a les nostres $m-1$ variables per a adquirir un grau d'inconsistència zero i generar la variable objectiu.

- **Creació dels conjunts de dades base:** Generarem aleatòriament totes les variables menys la variable objectiu del nostre conjunt de dades n vegades, és a dir per a cada instància, respectant el rang del domini de cada variable. Aplicarem a les n instàncies la funció $f(x_1, x_2, ..., x_{m-1})$ dissenyada en la primera fase, on interpretem x_i com a una variable la qual no és la variable objectiu. D'aquesta manera el resultat que ens doni la funció f per a cada instància serà la seva respectiva variable objectiu. En realitzar aquest procés ja tindrem construït el conjunt de dades. Aquest procés es repetirà tres vegades amb diferents llavors, d'aquesta manera tindrem tres conjunts de dades que apliquen la mateixa f, pero amb valors a les variables diferents. En aquests tres conjunts de dades, totes les seves variables seran rellevants, ja que totes han participat en la creació de la variable objectiu, per tant aquests conjunts de dades seran els nostres conjunts de dades base, els quals els hi haurem d'afegir més variables.

- **Disseny algorisme modificador del conjunt de dades:** En aquesta fase dissenyarem i implementarem un algorisme el qual modificarà els conjunts de dades base afegint-hi variables irrellevants i redundants. A partir dels paràmetres d'entrada de l'algorisme podrem afegir el nombre de variables irrellevants i redundats que necessitem. L'algorisme modificador del conjunt de dades serà explicat més en profunditat en la secció 4.2.4 d'aquest capítol. És important que aquestes modificacions s'apliquin sobre tres conjunts de dades bases

diferents perquè si s'apliquessin només sobre un, aquest algorisme base podria condicionar l'experimentació de les diferents versions del LVF. En tenir més diferència entre conjunts de dades reduïm aquest risc.

- **Aplicació de l'algorisme als conjunts de dades base:** Per a finalitzar el procés, aplicarem l'algorisme modificador als tres conjunts de dades base. En aquestes tres execucions modificarem el nombre de variables irrellevants i variables redundants a afegir. D'aquesta manera tindrem conjunts de dades molt similars però amb diferent nombre de variables irrellevants i redundants. Aquest fet facilita l'obtenció de conclusions respecte al comportament de les diferents versions del LVF.

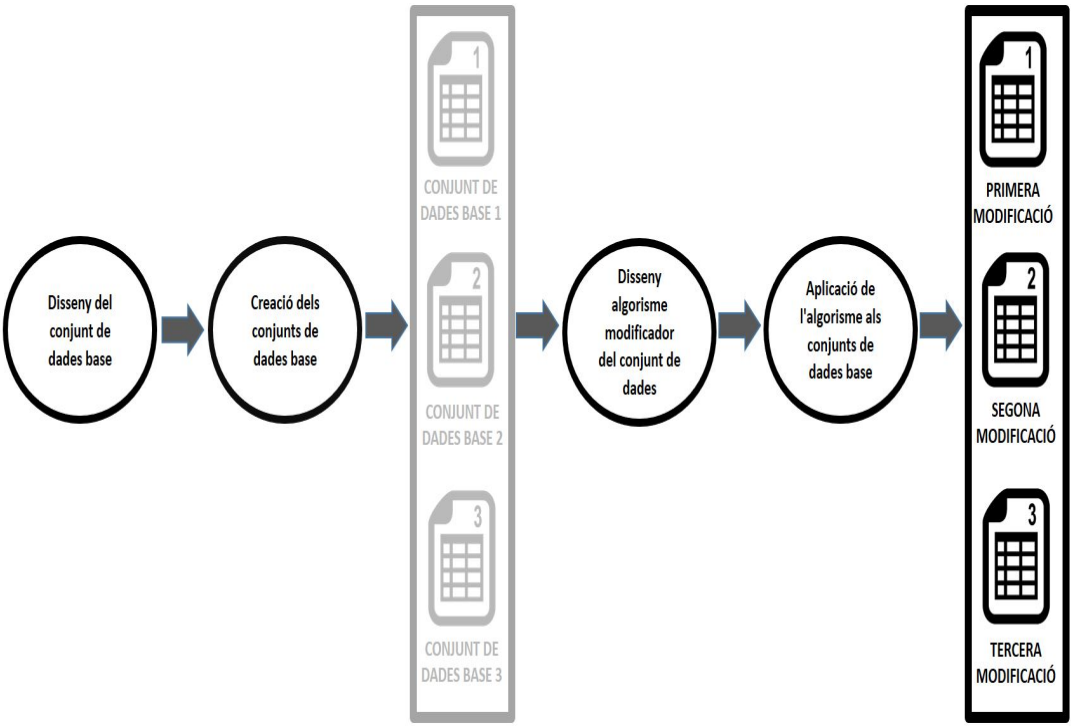


FIGURE 4.1: Diagrama del procés de creació dels conjunts de dades

4.2.2 Disseny dels conjunts de dades base

Vam decidir realitzar el procés esmentat anteriorment tres vegades, és a dir desenvolupar tres funcions f diferents i vam generar nou conjunt de dades inicials a partir d'elles (tres conjunts de dades per a cada f). A partir d'ara classificarem en tres blocs els conjunts de dades inicials, segons la funció f que s'ha aplicat en ells.

Amb aquests tres blocs de conjunts de dades inicials es busca observar el rendiment de les diferents versions de l'algorisme LVF en diferents entorns, per tant tenen característiques diferents. A continuació s'explicaran els tres blocs de conjunts de dades base.

Bloc 1

En aquest bloc trobem els conjunts de dades més simples amb la finalitat de què l'algorisme s'executi amb un temps d'execució molt baix, d'aquesta manera serà

molt ràpid realitzar proves amb ells. Els conjunts de dades d'aquest bloc estan formats per 500 instàncies amb 4 variables explicatives i un target binari. Com s'ha esmentat en els requeriments anteriors, totes les variables són categòriques. A la taula 4.1 trobem el nom i el domini d'aquestes variables.

Nom de la variable	**Domini**
x1	{0, 1}
x2	{0, 1}
x3	{0, 1}
x4	{0, 1}
target	{0, 1}

TAULA 4.1: Informació de les variables del conjunt de dades base 1

Per a la generació de la variable target, és a dir la variable objectiu dels conjunts de dades del bloc, utilitzarem una funció f^1 que pren aquesta forma $f^1 : \{0,1\}^k \rightarrow \{0,1\}$, per tant és una funció booleana que podem definir de la següent manera:

$$f^1(x1_i, x2_i, x3_i, x4_i) = x1_i \cap (x2_i \cup x3_i \cup x4_i) = target_i$$

on i representa l'aplicació sobre una instància, d'aquesta manera podem definir la i formalment com $\{i \in \mathbb{N} : 1 \leq i \leq n\}$, on n és el nombre d'instàncies.

Bloc 2

La finalitat d'aquest bloc de conjunts de dades era molt similar a la de l'anterior. Es volia un bloc amb conjunts de dades amb 400 instàncies i amb el mateix nombre de variables que l'anterior, però que a la vegada compliqués el càlcul d'inconsistència a l'algorisme. Això ho podem obtenir canviant les variables de binàries a numèriques discretes.

Per aquesta tasca vam decidir utilitzar un conjunt de dades sintètic anomenat *Balance Scale*[36] el qual el podem trobar a l'UCI Machine Learning Repository[37], el qual simula una balança on les variables $x1$ i $x3$ representen el pes de l'esquerra de la balança i el pes de la dreta respectivament. Les variables $x2$ i $x4$ indiquen la distància en la qual es troba aquest pes de l'esquerra i de la dreta respecte al centre de la balança. A la taula 4.2 podem observar el domini de les variables. A causa del fet que aquest conjunt de dades té més de 400 instàncies vam poder realitzar una selecció d'instàncies aleatòria per a poder definir els tres conjunts de dades del bloc amb diferents instàncies.

Nom de la variable	**Domini**
x1	{1, 2, 3, 4, 5}
x2	{1, 2, 3, 4, 5}
x3	{1, 2, 3, 4, 5}
x4	{1, 2, 3, 4, 5}
target	{B, L, R}

TAULA 4.2: Informació de les variables del conjunt de dades base 2

El càlcul de la variable objectiu també s'aconsegueix d'una funció f^2 molt simple, la qual podem expressar formalment d'aquesta manera:

$$f^2(x1_i, x2_i, x3_i, x4_i) = \begin{cases} B & \text{si } x1_i \times x2_i = x3_i \times x4_i \\ L & \text{si } x1_i \times x2_i > x3_i \times x4_i \\ R & \text{si } x1_i \times x2_i < x3_i \times x4_i \end{cases}$$

on i representa l'aplicació sobre una instància, d'aquesta manera podem definir la i formalment com $\{i \in \mathbb{N} : 1 \leq i \leq n\}$, on n és el nombre d'instàncies. Per tant, al aplicar $f^2(x1_i, x2_i, x3_i, x4_i) = target_i$ per tota i, calculem la variable objectiu per a totes les instàncies d'aquests conjunts de dades.

Bloc 3

En els conjunts de dades d'aquest bloc, es volia obtenir un nombre de variables rellevants més elevat. A part d'aquest increment de variables, també es volia incrementar el nombre d'instàncies que formen els conjunts de dades. És evident que per conseqüència d'aquest increment de les dimensionalitats el temps d'execució de la comprovació de les millores del LVF es veurà molt incrementat respecte als conjunts de dades dels blocs anteriors.

Per aquests objectius es va decidir que els conjunts de dades tinguessin 2000 instàncies, i que emulessin el clàssic problema de paritat. A la taula 4.3 podem observar les variables que formen aquests conjunts de dades i el seu domini.

Nom de la variable	**Domini**
x1	{0, 1}
x2	{0, 1}
x3	{0, 1}
x4	{0, 1}
x5	{0, 1}
x6	{0, 1}
x7	{0, 1}
x8	{0, 1}
x9	{0, 1}
x10	{0, 1}
target	{0, 1}

TAULA 4.3: Informació de les variables del conjunt de dades base 3

Aquest famós problema de la paritat calcula la variable objectiu aplicant la següent funció f^3:

$$f^3(x1_i, ..., x10_i) = \begin{cases} 0 & \text{si } ((\sum_{j=1}^{10} xj_i) \mod 2) = 0 \\ 1 & \text{si } ((\sum_{j=1}^{10} xj_i) \mod 2) = 1 \end{cases}$$

on i representa l'aplicació sobre una instància, d'aquesta manera podem definir la i formalment com $\{i \in \mathbb{N} : 1 \leq i \leq n\}$, on n és el nombre d'instàncies. En altres paraules, quan el nombre de variables amb valor 1 sigui parell, la variable objectiu serà 0, d'altra manera serà 1.

Aquests tres conjunts de dades d'aquest tercer bloc seran especialment difícils de classificar, a causa del fet que en ells no es compleix el principi de similitud.

Normalment, les solucions que s'han de classificar en una mateixa classe presenten una gran similitud entre les seves variables predictores, per aquest motiu molts classificadors intenten aprofitar-se d'aquest principi de similitud entre les instàncies d'una mateixa classe.

En canvi, en el cas del problema de paritat, aquest principi no es compleix, ja que un mínim canvi en una instància ja fa canviar la seva classe de la variable objectiu (en canviar una sola variable de 0 a 1 passa de senar a parell o viceversa). Per tant, no esperem un gran percentatge d'encert pel que fa al classificador aplicat en els conjunts de dades d'aquest bloc.

4.2.3 Creació dels conjunts de dades base

Per a la creació d'aquests conjunts de dades s'ha utilitzat el llenguatge de programació R, i s'ha utilitzat per emmagatzemar-los l'estructura de dades bidimensional *data.frame* que incorpora R, ja que aquesta estructura de dades permet utilitzar operacions molt útils a l'hora de tractar amb conjunts de dades.

No entrarem en detalls tècnics de la implementació dels algorismes generadors dels conjunts de dades perquè entendríem massa la mida de la memòria i tampoc és la finalitat d'aquest projecte.

En aquesta secció explicarem breument la creació dels conjunts de dades base que trobem a cada bloc definit anteriorment. Definirem les següents funcions per a ajudar-nos amb l'explicació:

- ***GeneracióBernoulli(n, p):*** Funció que generarà n valors independents els quals seran obtinguts a partir d'una distribució de Bernoulli amb una probabilitat p. Per tant, aquests valors només tindran un rang de {0 ,1} com a valor.
- ***Constructor(X):*** Funció constructora de la nostra estructura de dades que construirà el conjunt de dades amb el paràmetre X.
- ***Aplicador(f, X):*** Funció que aplicarà la funció f al conjunt de dades X i afegirà el resultat de cada instància en una última columna del conjunt de dades X.

Podem observar amb les funcions anteriors, que la generació de valors de les variables predictives es realitzarà mitjançant una generació aleatòria mitjançant la funció *GeneracióBernoulli(n, p)*. Cada generació aleatòria simularà una generació realitzada amb una distribució de Bernoulli, gràcies al fet que totes les generacions que realitzem només podran prendre dos valors. Aquests dos valors tindran la mateixa probabilitat de ser seleccionats i evidentment serà de 0.5.

A continuació s'exposarà un pseudocodi per a cada bloc de conjunts de dades base (exceptuant el bloc 2). Aquest pseudocodi és una petita aproximació a la implementació en R que ajudarà en l'explicació de la construcció dels conjunts de dades de cada bloc.

Cal recordar, que cada codi s'ha executat tres cops en el seu respectiu bloc, per a generar tres conjunts de dades amb diferents valors a les variables en cada bloc.

Bloc 1

En el següent pseudocodi podem veure que primerament generem les 4 variables predictives amb la funció explicada anteriorment, després construïm el conjunt de dades amb aquestes variables i finalment apliquem la funció generadora de la variable objectiu f_1 al conjunt de dades.

Algorisme 9: Generador conjunts de dades bloc 1

Entrada: f_1 - Funció generadora de la variable objectiu
Sortida: X - Conjunt de dades resultat

for $i = 0;\ i < 4;\ i = i + 1$ **do**
  $x_i = GeneracioBernoulli(500, 0.5)$
end
$X = Constructor(x_1, x_2, x_3, x_4)$
$Aplicador(f^1, X)$

Bloc 2

En aquest bloc, ja que hem importat un conjunt de dades ja generat no exposarem cap algorisme generador. Al conjunt de dades importat li hem donat un preprocessat en el qual s'han canviat el nom a les variables segons l'estàndard proposat en l'apartat 4.1, *Requeriments dels conjunts de dades*, i hem canviat de posició la seva variable objectiu. S'han seleccionat diferents instàncies del conjunt de dades importat per a la generació dels tres conjunts de dades d'aquest bloc.

Bloc 3

La generació d'aquests conjunts de dades és molt similar a la generació dels conjunts de dades base del bloc 1. Primerament generem les variables explicatives amb la funció *GeneracióBernoulli(n, p)*, posteriorment construïm el conjunt de dades amb els deu vectors de variables resultants i finalment apliquem la funció f^3 per a construir la variable objectiu *target*.

Algorisme 10: Generador conjunts de dades bloc 3

Entrada: f_3 - Funció generadora de la variable objectiu
Sortida: X - Conjunt de dades resultat

for $i = 0;\ i < 10;\ i = i + 1$ **do**
  $x_i = GeneracioBernoulli(2000, 0.5)$
end
$X = Constructor(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10})$
$Aplicador(f^3, X)$

4.2.4 Algorisme modificador del conjunt de dades

En aquesta secció s'estudiarà l'algorisme implementat per a la modificació dels conjunts de dades base. Aquest algorisme ha de ser capaç de realitzar les següents modificacions en el conjunt de dades d'entrada:

- **Modificació de la mida del conjunt de dades:** L'algorisme ha de poder afegir o eliminar instàncies del conjunt de dades d'entrada perquè el conjunt de dades

resultant tingui el nombre d'instàncies indicat en el paràmetre d'entrada de l'algorisme.

- **Afegir variables irrellevants:** L'algorisme ha de ser capaç d'afegir el nombre de variables irrellevants que el paràmetre d'entrada corresponent indiqui.
- **Afegir variables redundants:** L'algorisme ha de poder afegir el nombre de variables redundants que paràmetre d'entrada corresponent indiqui. També en el nom de la variable, s'ha de mostrar a quina variable rellevant repeteixen.

Com en l'apartat anterior, no entrarem en tecnicisme de la implementació en R. Però si donarem una explicació generalitzada en format de pseudocodi, per al correcte enteniment d'ell cal definir les següents funcions:

- ***AjustMida(X, n):*** Elimina o generà instàncies per al conjunt de dades X per tal que aquest conjunt de dades X tingui n instàncies en total. El cas de la generació d'instàncies segueix el mateix procediment de l'apartat anterior.
- ***GeneracióAleatòria(min, max, n):*** Aquesta funció genera n nombres en un rang de $\{i \in \mathbb{N} : min \leq i \leq max\}$ on i representa el possible nombre generat. Tots els nombres que es troben en l'interval de possibles candidats tenen la mateixa probabilitat a ser escollits.
- ***AfegirVariable(X, x, exemple):*** Funció que afegeix la variable x al conjunt de dades X amb el nom $exemple$.
- ***DesordenarVariables(X):*** Les posicions de les variables del conjunt de dades X es veuen alterades aleatòriament.

A continuació, s'exposa el pseudocodi utilitzat per a la modificació dels tres blocs de conjunts de dades, per tant és aplicable a tots els nostres conjunts de dades base.

Algorisme 11: Modificador conjunts de dades

Entrada:
X - Conjunt de dades base a modificar
n_I - Nombre de variables irrellevants a afegir a X
n_{Rd} - Nombre de variables redundants a afegir a X
T - Nombre d'instàncies esperat del conjunt de dades
min - Rang mínim dels valors de les variables
max - Rang màxim dels valors de les variables
Sortida:
X - Conjunt de dades modificat amb mida T, n_I nombre de variables irrellevants i n_{Rd} nombre de variables redundants

$n_R = nombreVariables(X)$
$X = AjustMida(X, T)$
$M = X$
for $i = 0;\ i < n_{Rd};\ i = i + 1$ **do**
  $j = GeneraciAleatoria(1, n_R, 1)$
  $AfegirVariable(M, x_j, reb_i.x_j)$
end
for $i = 0;\ i < n_I;\ i = i + 1$ **do**
  $x_I = GeneraciAleatoria(min, max, T)$
  $AfegirVariable(M, x_I, irr_i)$
end
$DesordenarVariables(M)$
$X = M$

Podem observar que inicialment obtenim el nombre de variables del conjunt de dades X, per tant la quantitat de les seves variables rellevants. Posteriorment, ajustarem la mida correcta del conjunt de dades amb la funció $AjustMida(X, T)$ comentada anteriorment.

El primer *for* ens servirà per a afegir el nombre de variables redundants al conjunt de dades que indica el paràmetre n_{Rd}. Observem que per cada iteració seleccionem una variable aleatòria a copiar i l'afegim al conjunt de dades amb el nom correcte.

El segon *for* l'utilitzarem per a afegir les variables irrellevants al conjunt de dades amb el seu nom correcte. Aquestes variables irrellevants estaran generades de manera aleatòria amb la funció *GeneracióAleatoria(min, max, T)* explicada anteriorment. El paràmetre n_I indica el nombre de variables irrellevants a afegir.

Finalment, desordenem les variables que es troben en el conjunt de dades i ja tindríem el conjunt de dades modificat correctament.

4.2.5 Modificacions dels conjunts de dades base

En aquesta secció s'expliquen les diferents modificacions realitzades per a cada conjunt de dades base aplicant l'algorisme modificador de conjunts de dades.

Com s'ha explicat en la secció de la creació dels conjunts de dades base d'aquest capítol, tenim tres conjunts de dades base per a cada bloc, cada un d'aquests conjunts de dades serà modificat amb l'algorisme esmentat. Aquestes tres modificacions tindran les següents finalitats:

- **Modificació 1:** L'estudi de les millores de l'algorisme LVF en un conjunt de dades on la proporció de variables irrellevants i redundants és equilibrada. Per tant, aquesta modificació tindrà el mateix nombre de variables irrellevant que de variables redundants.
- **Modificació 2:** L'estudi de les millores de l'algorisme LVF en un conjunt de dades amb una gran proporció de variables irrellevants per sobre de variables redundants. D'aquesta manera aquesta modificació tindrà variables irrellevants, però no tindrà variables redundants.
- **Modificació 3:** L'estudi de les millores de l'algorisme LVF en un conjunt de dades amb una gran proporció de variables redundants per sobre de variables irrellevants. En conseqüència, aquesta modificació tindrà variables redundants, però no tindrà variables irrellevants.

A continuació, es mostren les modificacions realitzades en les diferents versions dels conjunts de dades base. Els conjunts de dades que ara ja seran finals, seran descrits mitjançant un seguit de taules. És evident que no es pot mostrar tot el seu contingut per qüestions d'espai, però sí que es mostraran les variables de cada conjunt de dades, la seva tipologia, el seu domini i les proporcions de les seves categories.

Pel fet que en aquest apartat és necessari diferenciar els tres conjunts de dades que formen cada bloc ens referirem a ells com a conjunt de dades base i, on $i =$

$\{1, 2, 3\}$ és l'identificador del conjunt de dades tractat en el bloc. A la figura 4.1, Diagrama del procés de creació dels conjunts de dades, podem veure il·lustrat aquesta nomenclatura per a un bloc.

Bloc 1

Aquestes tres diferents modificacions es realitzen una a una sobre els tres conjunts de dades del bloc 1. Totes les modificacions d'aquests conjunts de dades tindran un total de 500 instàncies.

Modificació 1: Aquesta modificació afegeix tres variables irrellevants i tres variables redundants al conjunt de dades base 1 del bloc 1. A partir d'ara, ens referirem a aquest conjunt de dades com al conjunt de dades lògic 1 (CDL1). A la taula 4.4 podem apreciar la informació principal d'aquest conjunt de dades.

Nom de la variable	**Tipus de variable**	**Domini**	**Proporció de les categories**		
			Categoria	**Quantitat**	**Percentatge**
x4	Variable rellevant	{0, 1}	0	239	47.8%
			1	261	52.2%
irr1	Variable irrellevant	{0, 1}	0	257	51.4%
			1	243	48.6%
reb3.x3	Variable redundant (repeteix a x3)	{0, 1}	0	248	49.6%
			1	252	50.4%
x2	Variable rellevant	{0, 1}	0	258	51.6%
			1	242	48.4%
x1	Variable rellevant	{0, 1}	0	261	52.2%
			1	239	47.8%
reb1.x1	Variable rellevant (repeteix a x1)	{0, 1}	0	261	52.2%
			1	239	47.8%
irr2	Variable irrellevant	{0, 1}	0	267	53.4%
			1	233	46.6%
irr3	Variable irrellevant	{0, 1}	0	243	48.6%
			1	257	51.4%
reb2.x3	Variable redundant (repeteix a x3)	{0, 1}	0	248	49.6%
			1	252	50.4%
x3	Variable rellevant	{0, 1}	0	248	49.6%
			1	252	50.4%
target	Variable objectiu	{0, 1}	0	297	59.4%
			1	203	40.6%

TAULA 4.4: Informació de les variables del conjunt de dades CDL1

Modificació 2: En aquesta modificació s'afegeixen sis variables irrellevants i cap variable redundant al conjunt de dades base 2 del bloc 1. Des d'aquest moment, ens referirem a aquest conjunt de dades com al conjunt de dades lògic 2 (CDL2). A la taula 4.5 es mostra la informació principal d'aquest conjunt de dades.

Nom de la variable	Tipus de variable	Domini	Proporció de les categories		
			Categoria	Quantitat	Percentatge
x2	Variable rellevant	{0, 1}	0	256	51.2%
			1	244	48.8%
irr6	Variable irrellevant	{0, 1}	0	266	53.2%
			1	234	46.8%
irr3	Variable irrellevant	{0, 1}	0	244	48.8%
			1	256	51.2%
irr1	Variable irrellevant	{0, 1}	0	258	51.6%
			1	242	48.4%
irr4	Variable irrellevant	{0, 1}	0	238	47.6%
			1	262	52.4%
x1	Variable rellevant	{0, 1}	0	229	45.8%
			1	271	54.2%
irr5	Variable irrellevant	{0, 1}	0	251	50.2%
			1	249	49.8%
x4	Variable rellevant	{0, 1}	0	268	53.6%
			1	232	46.4%
x3	Variable rellevant	{0, 1}	0	241	48.2%
			1	259	51.8%
irr2	Variable irrellevant	{0, 1}	0	266	53.2%
			1	234	46.8%
target	Variable objectiu	{0, 1}	0	269	53.8%
			1	231	46.2%

TAULA 4.5: Informació de les variables del conjunt de dades CDL2

Modificació 3: En la tercera modificació s'afegeixen sis variables redundants i cap variable irrellevant al conjunt de dades base 3 pertanyent al bloc 1. En els següents apartats, ens referirem a aquest conjunt de dades com al conjunt de dades lògic 3 (CDL3). A la taula 4.6 podem veure la informació principal d'aquest conjunt de dades.

Nom de la variable	Tipus de variable	Domini	Proporció de les categories		
			Categoria	Quantitat	Percentatge
x3	Variable rellevant	{0, 1}	0	228	45.6%
			1	272	54.4%
reb2.x3	Variable redundant (repeteix a x3)	{0, 1}	0	228	45.6%
			1	272	54.4%
reb3.x3	Variable redundant (repeteix a x3)	{0, 1}	0	228	45.6%
			1	272	54.4%
reb6.x3	Variable redundant (repeteix a x3)	{0, 1}	0	228	45.6%
			1	272	54.4%
x4	Variable rellevant	{0, 1}	0	263	52.6%
			1	237	47.4%
reb1.x1	Variable redundant (repeteix a x1)	{0, 1}	0	265	53%
			1	235	47%
reb5.x2	Variable redundant (repeteix a x2)	{0, 1}	0	243	48.6%
			1	257	51.4%
x1	Variable rellevant	{0, 1}	0	265	53%
			1	235	47%
x2	Variable rellevant	{0, 1}	0	243	48.6%
			1	257	51.4%
reb4.x3	Variable redundant (repeteix a x3)	{0, 1}	0	228	45.6%
			1	272	54.4%
target	Variable objectiu	{0, 1}	0	298	59.6%
			1	202	40.4%

TAULA 4.6: Informació de les variables del conjunt de dades CDL3

Bloc 2

Les següents tres modificacions són realitzades una a una sobre els tres conjunts de dades base del bloc 2 que hem generat anteriorment. Totes aquestes modificacions d'aquests conjunts de dades tindran 400 instàncies en total.

Modificació 1: Aquesta primera modificació afegeix tres variables irrellevants i tres variables redundants al conjunt de dades base 1 del bloc 2. Des d'aquest moment, ens referirem a aquest conjunt de dades com al conjunt de dades de l'escala d'equilibris 1 (CDE1). A la taula 4.7 podem apreciar la informació principal d'aquest conjunt de dades.

Modificació 2: En aquesta modificació s'afegeixen sis variables irrellevants i cap variable redundant al conjunt de dades base 2 del segon bloc. A partir d'ara, ens referirem a aquest conjunt de dades com al conjunt de dades de l'escala d'equilibris 2 (CDE2). A la taula 4.8 podem veure la informació principal d'aquest conjunt de dades.

Modificació 3: Aquesta modificació afegeix sis variables redundants i cap variable irrellevant al conjunt de dades base 3 del segon bloc. En els següents apartats, ens referirem a aquest conjunt de dades com al conjunt de dades de l'escala d'equilibris 3 (CDE3). A la taula 4.9 podem observar la informació principal d'aquest conjunt de dades.

Nom de la variable	**Tipus de variable**	**Domini**	**Proporció de les categories**		
			Categoria	**Quantitat**	**Percentatge**
x3	Variable rellevant	{1, 2, 3, 4, 5}	1	76	19%
			2	82	20.5%
			3	76	19%
			4	84	21%
			5	82	20.5%
irr1	Variable irrellevant	{1, 2, 3, 4, 5}	1	93	23.25%
			2	72	18%
			3	93	23.25%
			4	65	16.25%
			5	77	19.25%
irr2	Variable irrellevant	{1, 2, 3, 4, 5}	1	69	17.25%
			2	73	18.25%
			3	88	22%
			4	89	22.25%
			5	81	20.25%
x1	Variable rellevant	{1, 2, 3, 4, 5}	1	79	19.75%
			2	75	18.75%
			3	84	21%
			4	78	19.5%
			5	84	21%
x4	Variable rellevant	{1, 2, 3, 4, 5}	1	70	17.5%
			2	77	19.25%
			3	82	20.5%
			4	88	22%
			5	83	20.75%
reb1.x2	Variable redundant (repeteix a x2)	{1, 2, 3, 4, 5}	1	80	20%
			2	72	18%
			3	83	20.75%
			4	89	22.25%
			5	76	19%
reb2.x4	Variable redundant (repeteix a x4)	{1, 2, 3, 4, 5}	1	70	17.5%
			2	77	19.25%
			3	82	20.5%
			4	88	22%
			5	83	20.75%
x2	Variable rellevant	{1, 2, 3, 4, 5}	1	80	20%
			2	72	18%
			3	83	20.75%
			4	89	22.25%
			5	76	19%
irr3	Variable irrellevant	{1, 2, 3, 4, 5}	1	71	17.75%
			2	79	19.75%
			3	76	19%
			4	81	20.25%
			5	93	23.25%
reb3.x4	Variable redundant (repeteix a x4)	{1, 2, 3, 4, 5}	1	70	17.5%
			2	77	19.25%
			3	82	20.5%
			4	88	22%
			5	83	20.75%
target	Variable objectiu	{B, L, R}	B	31	7.75%
			L	183	45.75%
			R	186	46.5%

TAULA 4.7: Informació de les variables del conjunt de dades CDE1

Nom de la variable	Tipus de variable	Domini	Proporció de les categories		
			Categoria	Quantitat	Percentatge
irr6	Variable irrellevant	{1, 2, 3, 4, 5}	1	79	19.75%
			2	89	22.25%
			3	75	18.75%
			4	82	20.5%
			5	75	18.75%
irr2	Variable irrellevant	{1, 2, 3, 4, 5}	1	89	22.25%
			2	78	19.5%
			3	72	18%
			4	81	20.25%
			5	80	20%
x4	Variable rellevant	{1, 2, 3, 4, 5}	1	80	20%
			2	79	19.75%
			3	84	21%
			4	79	19.75%
			5	78	19.5%
irr6	Variable irrellevant	{1, 2, 3, 4, 5}	1	88	22%
			2	85	21.25%
			3	83	20.75%
			4	71	17.75%
			5	73	18.25%
x3	Variable rellevant	{1, 2, 3, 4, 5}	1	78	19.5%
			2	80	20%
			3	77	19.25%
			4	73	18.25%
			5	92	23.25%
irr5	Variable irrellevant	{1, 2, 3, 4, 5}	1	90	22.5%
			2	78	19.5%
			3	88	22%
			4	81	20.25%
			5	63	15.75%
x1	Variable rellevant	{1, 2, 3, 4, 5}	1	79	19.75%
			2	80	20%
			3	76	19%
			4	78	19.5%
			5	87	21.75%
irr3	Variable irrellevant	{1, 2, 3, 4, 5}	1	86	21.5%
			2	77	19.25%
			3	83	20.75%
			4	75	18.75%
			5	79	19.75%
x2	Variable rellevant	{1, 2, 3, 4, 5}	1	76	19%
			2	80	20%
			3	92	23.25%
			4	82	20.5%
			5	70	17.5%
irr1	Variable irrellevant	{1, 2, 3, 4, 5}	1	88	22%
			2	88	22%
			3	93	23.25%
			4	63	15.75%
			5	68	17%
target	Variable objectiu	{B, L, R}	B	29	7.25%
			L	186	46.5%
			R	185	46.25%

TAULA 4.8: Informació de les variables del conjunt de dades CDE2

Nom de la variable	**Tipus de variable**	**Domini**	**Proporció de les categories**		
			Categoria	**Quantitat**	**Percentatge**
x2	Variable rellevant	{1, 2, 3, 4, 5}	1	88	22%
			2	84	21%
			3	74	18.5%
			4	80	20%
			5	74	18.5%
reb1.x4	Variable redundant (repeteix a x4)	{1, 2, 3, 4, 5}	1	84	21%
			2	86	21.5%
			3	72	18%
			4	79	19.75%
			5	79	19.75%
reb3.x3	Variable redundant (repeteix a x3)	{1, 2, 3, 4, 5}	1	82	20.5%
			2	81	20.25%
			3	89	22.25%
			4	81	20.25%
			5	67	16.75%
x4	Variable rellevant	{1, 2, 3, 4, 5}	1	84	21%
			2	86	21.5%
			3	72	18%
			4	79	19.75%
			5	79	19.75%
reb6.x3	Variable redundant (repeteix a x3)	{1, 2, 3, 4, 5}	1	82	20.5%
			2	81	20.25%
			3	89	22.25%
			4	81	20.25%
			5	67	16.75%
x3	Variable rellevant	{1, 2, 3, 4, 5}	1	82	20.5%
			2	81	20.25%
			3	89	22.25%
			4	81	20.25%
			5	67	16.75%
reb4.x2	Variable rellevant (repeteix a x2)	{1, 2, 3, 4, 5}	1	88	22%
			2	84	21%
			3	74	18.5%
			4	80	20%
			5	74	18.5%
x1	Variable rellevant	{1, 2, 3, 4, 5}	1	81	20.25%
			2	69	17.25%
			3	85	21.25%
			4	84	21%
			5	81	20.25%
reb2.x2	Variable rellevant (repeteix a x2)	{1, 2, 3, 4, 5}	1	88	22%
			2	84	21%
			3	74	18.5%
			4	80	20%
			5	74	18.5%
reb5.x1	Variable rellevant (repeteix a x1)	{1, 2, 3, 4, 5}	1	81	20.25%
			2	69	17.25%
			3	85	21.25%
			4	84	21%
			5	81	20.25%
target	Variable objectiu	{B, L, R}	B	31	7.75%
			L	185	46.25%
			R	184	46%

TAULA 4.9: Informació de les variables del conjunt de dades CDE3

Bloc 3

Les següents modificacions són realitzades una a una sobre els diferents conjunts de dades del tercer bloc, generats anteriorment. Totes les modificacions d'aquests conjunts de dades tindran un total de 2000 instàncies. A causa del fet que trobem més instàncies i més variables en els conjunts de dades d'aquest bloc, s'afegeixen un major nombre de variables en aquestes modificacions.

Modificació 1: S'afegeixen quatre variables irrellevants i quatre variables redundants al conjunt de dades base 1 del tercer bloc. A partir d'ara, ens referirem a aquest conjunt de dades com al conjunt de dades de paritat 1 (CDP1). A la taula 4.10 podem apreciar la informació principal d'aquest conjunt de dades.

Nom de la variable	Tipus de variable	Domini	Proporció de les categories		
			Categoria	Quantitat	Percentatge
irr1	Variable irrellevant	{0, 1}	0	987	49.35%
			1	1013	50.65%
x7	Variable rellevant	{0, 1}	0	1014	50.7%
			1	986	49.3%
x6	Variable rellevant	{0, 1}	0	999	49.95%
			1	1001	50.05%
irr4	Variable irrellevant	{0, 1}	0	964	48.2%
			1	1036	51.8%
x2	Variable rellevant	{0, 1}	0	1000	50%
			1	1000	50%
x3	Variable rellevant	{0, 1}	0	987	49.35%
			1	1013	50.65%
x8	Variable rellevant	{0, 1}	0	990	49.5%
			1	1010	50.5%
reb3.x6	Variable redundant (repeteix a x6)	{0, 1}	0	999	49.95%
			1	1001	50.05%
x1	Variable rellevant	{0, 1}	0	964	48.2%
			1	1036	51.8%
x4	Variable rellevant	{0, 1}	0	1015	50.75%
			1	985	49.25%
reb2.x8	Variable redundant (repeteix a x8)	{0, 1}	0	990	49.5%
			1	1010	50.5%
x10	Variable rellevant	{0, 1}	0	1032	51.6%
			1	968	48.4%
irr2	Variable irrellevant	{0, 1}	0	994	49.7%
			1	1006	50.3%
x9	Variable rellevant	{0, 1}	0	990	49.5%
			1	1010	50.5%
reb4.x8	Variable redundant (repeteix a x8)	{0, 1}	0	990	49.5%
			1	1010	50.5%
x5	Variable rellevant	{0, 1}	0	974	48.7%
			1	1026	51.3%
reb1.x8	Variable redundant (repeteix a x8)	{0, 1}	0	990	49.5%
			1	1010	50.5%
irr2	Variable irrellevant	{0, 1}	0	1010	50.5%
			1	990	49.5%
target	Variable objectiu	{0, 1}	0	1007	50.35%
			1	993	49.65%

TAULA 4.10: Informació de les variables del conjunt de dades CDP1

Modificació 2: Aquesta modificació afegeix sis variables irrellevants i cap variable redundant al conjunt de dades base 2 del tercer bloc. En els següents apartats, ens referirem a aquest conjunt de dades com al conjunt de dades de paritat 2 (CDP2). A la taula 4.11 podem veure la informació principal d'aquest conjunt de dades.

Nom de la variable	Tipus de variable	Domini	Proporció de les categories		
			Categoria	Quantitat	Percentatge
x7	Variable rellevant	{0, 1}	0	1013	50.65%
			1	987	49.35%
x1	Variable rellevant	{0, 1}	0	1037	51.85%
			1	963	48.15%
irr8	Variable irrellevant	{0, 1}	0	1015	50.75%
			1	985	49.25%
irr5	Variable irrellevant	{0, 1}	0	1008	50.4%
			1	992	49.6%
irr1	Variable irrellevant	{0, 1}	0	1001	50.05%
			1	999	49.95%
x3	Variable rellevant	{0, 1}	0	944	47.2%
			1	1056	52.8%
x9	Variable rellevant	{0, 1}	0	1012	50.6%
			1	988	49.4%
x6	Variable rellevant	{0, 1}	0	971	48.55%
			1	1029	51.45%
irr3	Variable irrellevant	{0, 1}	0	1008	50.4%
			1	992	49.6%
x8	Variable rellevant	{0, 1}	0	982	49.1%
			1	1018	50.9%
x4	Variable rellevant	{0, 1}	0	1026	51.3%
			1	974	48.7%
irr6	Variable irrellevant	{0, 1}	0	1002	50.1%
			1	998	49.9%
x10	Variable rellevant	{0, 1}	0	1016	50.8%
			1	984	49.2%
irr2	Variable irrellevant	{0, 1}	0	968	48.4%
			1	1032	51.6%
irr7	Variable irrellevant	{0, 1}	0	1038	51.9%
			1	962	48.1%
x2	Variable rellevant	{0, 1}	0	999	49.95%
			1	1001	50.05%
x5	Variable rellevant	{0, 1}	0	1045	52.25%
			1	955	47.75%
irr4	Variable irrellevant	{0, 1}	0	1023	51.15%
			1	977	48.85%
target	Variable objectiu	{0, 1}	0	1009	50.45%
			1	991	49.55%

TAULA 4.11: Informació de les variables del conjunt de dades CDP2

Modificació 3: Aquesta última modificació afegeix sis variables redundants i cap variable irrellevant al conjunt de dades base 3 del bloc 3. Des d'aquest moment, ens referirem a aquest conjunt de dades com al conjunt de dades de paritat 3 (CDP3). A la taula 4.12 podem observar la informació principal d'aquest conjunt de dades.

Nom de la variable	Tipus de variable	Domini	Proporció de les categories		
			Categoria	Quantitat	Percentatge
x5	Variable rellevant	{0, 1}	0	1030	51.5%
			1	970	48.5%
x1	Variable rellevant	{0, 1}	0	988	49.4%
			1	1012	50.6%
reb2.x7	Variable redundant (repeteix a x7)	{0, 1}	0	1047	52.35%
			1	953	47.65%
reb6.x7	Variable redundant (repeteix a x7)	{0, 1}	0	1047	52.35%
			1	953	47.65%
reb1.x10	Variable redundant (repeteix a x10)	{0, 1}	0	994	49.7%
			1	1006	50.3%
reb8.x4	Variable redundant (repeteix a x4)	{0, 1}	0	981	49.05%
			1	1019	50.95%
reb4.x2	Variable redundant (repeteix a x2)	{0, 1}	0	965	48.25%
			1	1035	51.75%
x3	Variable rellevant	{0, 1}	0	974	48.7%
			1	1026	51.3%
x2	Variable rellevant	{0, 1}	0	965	48.25%
			1	1035	51.75%
reb5.x8	Variable redundant (repeteix a x8)	{0, 1}	0	1025	51.25%
			1	975	48.75%
x7	Variable rellevant	{0, 1}	0	1047	52.35%
			1	953	47.65%
reb7.x4	Variable redundant (repeteix a x4)	{0, 1}	0	981	49.05%
			1	1019	50.95%
x8	Variable rellevant	{0, 1}	0	1025	51.25%
			1	975	48.75%
x4	Variable rellevant	{0, 1}	0	981	49.05%
			1	1019	50.95%
x9	Variable rellevant	{0, 1}	0	1007	50.35%
			1	993	49.65%
x6	Variable rellevant	{0, 1}	0	972	48.6%
			1	1028	51.4%
reb3.x4	Variable redundant (repeteix a x4)	{0, 1}	0	981	49.05%
			1	1019	50.95%
x10	Variable rellevant	{0, 1}	0	994	49.7%
			1	1006	50.3%
target	Variable objectiu	{0, 1}	0	1051	52.55%
			1	949	47.45%

TAULA 4.12: Informació de les variables del conjunt de dades CDP3

4.3 Conjunts de dades finals

En aquesta secció explicarem els conjunts de dades seleccionats per a la fase final de la investigació, on ja tenim la millora del LVF molt desenvolupada. Per tant, aquests conjunts de dades seran necessaris per a realitzar la valoració final de les millores del LVF.

Per a desenvolupar correctament aquesta tasca necessitem uns conjunts de dades els més pròxims a la realitat possible. Aquests conjunts de dades també han de mostrar una dificultat elevada per a la seva correcta classificació, fet que és possible

trobar en conjunts de dades sintètics, com per exemple el problema de les dues espirals[38]. Però, el problema amb aquest conjunt de dades i amb els conjunts de dades sintètics en general, és que sabem a priori que existeix una solució capaç de resoldre'ls, fet que a la realitat no sempre és veritat. Per tant, si haguéssim desenvolupat tota la nostra investigació amb conjunts de dades sintètics, les propietats esmentades d'ells podrien haver esbiaixat els resultats i afectat a la seva regularitat en trobar els conjunts de variables adequats[39].

Una alternativa utilitzada per millorar la semblança dels conjunts de dades sintètics als conjunts de dades reals, és afegir soroll estocàstic al conjunt de dades. Aquesta opció, però té dos desavantatges respecta utilitzar un conjunt de dades real:

- Pot aparèixer biaix a la selecció de l'algorisme que construirà el model predictiu pel tipus de procés utilitzat per a generar les dades; si utilitzem unes dades generades per un soroll Gaussià multidimensional, un classificador de funció de base radial basat en una distribució gaussiana, tindria resultats excel·lents, ja que el model emprat en la construcció del classificador i el model emprat en la generació del conjunt de dades són molt similars[39].
- És molt difícil saber quina quantitat de soroll i de quin tipus s'ha d'aplicar en el nostre conjunt de dades per a aproximar-lo a un domini real.

Per evitar-nos aquests problemes, els conjunts de dades emprats en la fase final de la investigació tindran un origen real. Aquest fet, també ens garanteix que el nostre estudi final és rellevant per a alguns dominis reals, almenys en els que hem emprat amb els nostres conjunts de dades finals. Per a emfatitzar aquesta evidència utilitzarem conjunts de dades reals pertanyents a diferents dominis reals. Els conjunts de dades emprats s'han extret del repositori *University of California, Irvine Machine Learning Repository*[37].

En aquests conjunts de dades, òbviament no s'ha seguit cap procés de creació gràcies al fet que no els hem construït nosaltres. L'objectiu era no interferir massa en els conjunts de dades reals i d'aquesta manera observa el rendiment verídic de les millores en aquests conjunts de dades. Tot i això, s'han hagut d'adaptar en alguns detalls al nostre algorisme i s'ha realitzat un breu preprocessat en alguns d'ells.

En tots els conjunts de dades, s'ha situat la variable objectiu (*target*) en l'última posició de les variables, fet que indica a l'algorisme quina és la variable objectiu.

S'ha volgut testar el rendiment de l'algorisme amb conjunts de dades que continguin variables contínues. A causa del fet que la nostra mesura d'avaluació no permet treballar amb conjunts de dades amb variables contínues, ha estat necessari realitzar la discretització de les variables contínues dels conjunts de dades.

4.3.1 Discretització de les variables

La discretització de variables consisteix en la transformació dels valors d'una variable contínua en un nombre finit d'intervals, en el qual a cada interval s'associa un valor numèric discret. Aquest procés el podem descompondre en dues tasques:

- Trobar el nombre d'intervals discrets adequats. La majoria d'algorismes de discratització no realitzen aquesta tasca i necessiten que l'ususari especifiqui el nombre d'intervals.

- Trobar el nombre d'intervals discrets adequats. La majoria d'algorismes de discretització no duen a terme aquesta tasca i necessiten que l'usuari especifiqui el nombre d'intervals.

Ja que no pretenem condicionar la discretització, es va decidir utilitzar un algorisme de discretització capaç d'acomplir les dues tasques esmentades, perquè si definíssim el nombre d'intervals a utilitzar, estaríem condicionant aquesta discretització. L'algorisme de discretització CAIM[40] compleix amb aquest requeriment.

També, aquest algorisme es troba dins del conjunt d'algorismes de discretització supervisats, és a dir, realitza la seva discretització tenint en compte la interdependència entre les classes de la variable objectiu i els valors de les variables explicatives (atributs).

La idea principal d'aquest algorisme, és discretitzar la variable en el menor nombre d'intervals i maximitzar la interdependència classe-atribut. Fet que s'adapta excepcionalment al nostre algorisme LVF, ja que:

- En contenir un nombre d'intervals discrets menors per variable, es facilitarà la cerca del grau d'inconsistència. Per tant, el temps d'execució es reduirà envers un nombre d'intervals majors per variable.
- En maximitzar la interdependència classe-variable, el rendiment del classificador serà superior que amb l'ús d'un algorisme de discretització no supervisat, ja que la discretització s'ha vist condicionada amb la variable objectiu.

D'aquesta manera, s'utilitzarà aquest algorisme quan algun conjunt de dades necessiti l'aplicació d'una discretització de variables.

4.3.2 Algorisme CAIM

A continuació, explicarem breument el funcionament d'aquest algorisme; caldrà definir els següents conceptes. Definirem n com el nombre d'instàncies totals del conjunt de dades a discretitzar, on cada instància pertany a una classe del conjunt de classes C, definim F com qualsevol de les variables contínues que es volen discretitzar, ens referirem a la discretització aplicada a F com D, la qual discretitzarà el domini continu de F en m intervals discrets: $D : \{[d_o, d_1], (d_1, d_2], ..., (d_{m-1}, d_m]\}$.

A partir d'aquestes dades definirem; q_{ir} com el nombre de valors continus que es troben a l'interval r, és a dir a $(d_{r-1}, d_r]$, i pertanyent a la classe i de C. M_{+r} com el nombre de valors continus de F a l'interval r i finalment max_r com el màxim valor entre tots els valors de q_{ir} per a l'interval r.

Ja podem definir el criteri de discretització que seguirà aquest algorisme; aquest criteri de discretització és el CAIM[40], la maximització de la interdependència classe-atribut (en anglès, *Class-Attribute Interdependency Maximization*), el qual mesura la dependència entre C i la variable discretitzada D per la variable contínua F:

$$CAIM(C, D|F) = \frac{\sum_{r=1}^{n} \frac{max_r^2}{M_{+r}}}{n}$$

Aquest criteri de discretització serà emprat com a mesura heurística en l'algorisme CAIM, a continuació s'exposa el pseudocodi de l'algorisme.

Algorisme 12: Generador conjunts de dades bloc 3

Entrada:
F - Variables continues
S - Instàncies del conjunt de dades on es troben F
C - Classes de la variable objectiu
Sortida:
D - Discretització de les variables continues F

for $f \in F$ **do**
  $d_o = min(f)$
  $d_n = max(f)$
  $B =$ Ordenar ascendentment valors f i inicialitzar tots els possibles límits dels intervals de f
  $D = [d_o, d_n],\ k = 1$ `// Discretització inicial`
  $GlobalCAIM = 0$
  repeat
    $(CAIM, d_i) = MillorLimit(C, S, D, \{B \backslash D\})$
    if $((CAIM > GlobalCAIM)$ ***or*** $(k < |C|))$ **then**
      $GlobalCAIM = CAIM$
      $D = D \cap d_i$
    end
  until $((CAIM \leq GlobalCAIM)$ ***and*** $(k \geq |C|))$
end

La funció $MillorLimit(C, S, D, G)$ provarà tots els límits interiors que es troben a G afegint-los a D individualment, i seleccionarà el que obtingui un major CAIM amb les instàncies S i les classes C.

Per tant, podem observar que l'algorisme utilitza un enfocament aproximatiu (en anglès, *greedy*), a causa del gran nombre de possibles combinacions d'intervals que trobem a l'espai de solucions. D'aquesta manera cercarem un màxim local pel CAIM, però no l'òptim, ja que no és viable computacionalment la cerca d'aquest. L'algorisme va afegint un a un els intervals que aporten un major CAIM fins que ja no es pot millorar el CAIM, però no prova totes les combinatòries.

Els autors[40] reporten que l'algorisme tendeix a utilitzar un nombre d'intervals reduït, concretament $|C|$, on C són les classes del nostre problema. Per tant, l'algorisme per discretitzar una única variable executarà $\mathcal{O}(|C|)$ vegades la funció $MillorLimit()$. Aquesta funció té un cost de $\mathcal{O}(M \cdot |C|^2)$, on M és el nombre de valors diferents en la variable a discretitzar, per tant el bucle té un cost de $\mathcal{O}(M \cdot |C|^2) \cdot \mathcal{O}(|C|) = \mathcal{O}(M \cdot |C|^3)$. Pel que fa a la part prèvia al bucle, trobem un cost de $\mathcal{O}(M \cdot \log M)$, referent a l'ordenació dels valors candidats a ser límits dels intervals. Gràcies al fet que $|C|$ tendeix a tenir valors petits, la discretització d'una variable té un cost de $\mathcal{O}(M \cdot \log M)$. El que provoca que es pugui aplicar a conjunts de dades grans.

A continuació, passarem a explicar individualment els conjunts de dades que s'utilitzaran per a aquesta part final del projecte. S'ha intentat que aquests conjunts de dades a part de pertànyer a diferents àrees, tinguin característiques diferents en base la seva mida, tipologia de variables, nombre de classes, etc. D'aquesta manera

els resultats del projecte seran més robustos.

4.3.3 Conjunt de dades Ionosphere

El conjunt de dades ionosphere[41] tracta la problemàtica de la detecció d'electrons lliures a la ionosfera (part de l'atmosfera terrestre, ionitzada permanentment a causa de la radiació solar). Aquest és un camp d'estudi molt important, ja que aquestes zones d'electrons lliures permeten que aparegui un efecte de reflexió quan enviem senyals de ràdio o altres tipus d'ones electromagnètiques, i facilitar conseqüentment la transmissió de missatges.

Els valors del conjunt de dades es van obtenir a partir de la transmissió d'ones d'un conjunt d'antenes d'alta freqüència. A partir dels senyals rebuts amb 17 números de polsos diferents, es va aplicar per cada nombre de pols una funció d'autocorrelació la qual ens retorna 2 atributs continus. Per tant, per a cada instància trobem 34 atributs continus i la variable objectiu binària indicant si provocà reflexió o no.

A causa del fet que trobem variables contínues en el conjunt de dades, s'ha aplicat la discretització explicada anteriorment. Com era d'esperar, s'ha obtingut una discretització en dos intervals per a totes les variables contínues del conjunt de dades (el CAIM tendeix a utilitzar un nombre d'intervals equivalent al nombre de classes del conjunt de dades). A continuació, es mostra la informació principal del conjunt de dades.

Nombre d'instàncies	351
Nombre d'atributs	34
Discretització aplicada	Cert
Tipologia dels atributs	Binaris
Tipologia de la variable objectiu	Binària

TAULA 4.13: Característiques del conjunt de dades *Ionosphere*

S'ha obtingut un conjunt de dades amb totes les variables de tipologia binària, amb un bon nombre d'instàncies i variables. S'espera una reducció d'un gran nombre de variables en l'experimentació d'aquest conjunt de dades, ja que es té el coneixement previ que es troben moltes variables que causen soroll a la predicció. Ens referirem a aquest conjunt de dades amb l'acrònim CDI.

4.3.4 Conjunt de dades Mushroom

El clàssic conjunt de dades Mushroom[42] agrupa com a variables característiques de diferents bolets pertanyents a 23 espècies diferents de bolets de les famílies Agaricus i Lapiota. Cada espècie bé identificada com comestible, verinós o no recomanada la seva comestibilitat a causa de falta d'informació. Aquestes dues últimes classes es troben contingudes en una classe definida com no comestible. Per tant la finalitat del conjunt de dades és la classificació dels bolets en comestible o no comestibles.

En aquest conjunt de dades totes les variables són categòriques, en conseqüència, no hem d'aplicar cap discretització a les variables. Seguidament, es mostren les característiques principals del conjunt de dades.

Nombre d'instàncies	8124
Nombre d'atributs	22
Discretització aplicada	Falç
Tipologia dels atributs	Categòrics
Nombre de categories dels atributs	Variats
Tipologia de la variable	Binària

TAULA 4.14: Característiques del conjunt de dades *Mushrooms*

Trobem un elevat nombre d'instàncies i un bon nombre d'atributs en aquest conjunt de dades. S'espera que l'*accuracy* obtinguda amb les prediccions d'aquest conjunt de dades siguin molt bones, pel fet que la tasca de classificació amb ell és força senzilla. També, gràcies al seu elevat nombre d'instàncies els temps d'execució seran alts, ja que el temps de computar la inconsistència del conjunt de dades és elevat. Ens referirem a aquest conjunt de dades amb l'acrònim CDM.

4.3.5 Conjunt de dades Congressional Voting Records

En el conjunt de dades Congressional Voting Records[43] es troben 16 votacions claus dels Congressistes de la Cambra de Representants dels EUA a la segona sessió de 1984 del 98è Congrés. Per tant, hi ha 16 atributs en aquest conjunt de dades, un per a cada votació. La finalitat d'aquest conjunt de dades és la identificació del partit polític del votant. Trobem la representació de dos partits polítics, el partit republicà i el partit demòcrata.

Per a cada votació trobem tres modalitats, les quals tenen aquests significats:

- A favor: Va votar a favor, va aparellar a favor o es va anunciar a favor.
- En contra: Va votar en contra, va aparellar en contra o es va anunciar en contra.
- Desconegut: Vot per a evitar conflicte d'interès, no votat o no ha donat a conèixer la seva posició.

Decidim preservar les tres modalitats perquè creiem que la modalitat desconegut pot aportar informació a la predicció, gràcies al fet que en algunes votacions a alguns votants segons el seu partit polític podien abstenir-se per a evitar conflictes. A sota, es mostren les principals característiques del conjunt de dades.

Nombre d'instàncies	435
Nombre d'atributs	16
Discretització aplicada	Falç
Tipologia dels atributs	Categòrics
Nombre de categories dels atributs	3
Tipologia de la variable objectiu	Binària

TAULA 4.15: Característiques del conjunt de dades *Congressional Voting Records*

En aquest conjunt de dades trobem un bon nombre d'instàncies però una menor quantitat de variables respecte als altres conjunts de dades. És possible que trobem una menor reducció de variables per aquest fet. És interessant gràcies al fet que l'àrea

d'aquest conjunt de dades és molt diferent de la dels altres utilitzats. Ens referirem a aquest conjunt de dades amb l'acrònim CDV.

4.3.6 Conjunt de dades Connectionist Bench (Sonar, Mines vs. Rocks)

En el conjunt de dades Connectionist Bench[44], trobem 111 instàncies obtingudes a partir del rebot d'uns senyals generats per un sonar amb diferents freqüències sobre un cilindre metàl·lic, en diversos angles i diferents condicions. També trobem 97 instàncies obtingudes amb un procediment similar però sobre diferents roques.

Cada instància conté 60 variables amb un valor continu entre [0, 1] i representen l'energia dins d'una banda de freqüència concreta. El cilindre metàl·lic amb el qual es realitza part de l'experimentació simula una mina. En conseqüència, la tasca d'aquest algorisme és la classificació de les instàncies segons siguin roques o mines. A causa del fet que trobem atributs continus en el conjunt de dades, s'ha aplicat la discretització explicada sobre aquests. Seguidament s'exposen les característiques principals del conjunt de dades.

Nombre d'instàncies	208
Nombre d'atributs	60
Discretització aplicada	Cert
Tipologia dels atributs	Binaris
Tipologia de la variable objectiu	Binaris

TAULA 4.16: Característiques del conjunt de dades *Connectionist Bench (Sonar, Mines vs. Rocks)*

Aquest conjunt de dades es caracteritza per a tenir un alt nombre de variables i un bon nombre d'instàncies, la discretització aplicada ha discretitzat tots els atributs en dos intervals discrets. Com que tenim moltes variables i només dues classes a la variable objectiu, és molt possible que obtinguem una alta reducció en les variables. Ens referirem a aquest conjunt de dades amb l'acrònim CDC.

4.3.7 Conjunt de dades Waveform Database Generator (Version 2)

El conjunt de dades Waveform Database Generator (Version 2)[45] és una modificació del conjunt de dades *Waveform Database Generator*[46]. En aquest conjunt de dades per instància trobem 21 atributs que descriuen la forma d'una ona generada artificialment i 19 atributs que afegeixen un soroll amb mitjana 0 i variància 1. Aquest soroll afegit és la diferència entre la versió original del conjunt de dades i la versió seleccionada.

S'ha decidit seleccionar la versió amb soroll perquè ens permetrà valorar més correctament el potencial de reducció de variables de les millores del LVF. Les instàncies s'han de classificar en tres tipus d'ones generals, aquestes classes són generades a partir d'una combinació de 2 de 3 ones "base". Com que tots els atributs del conjunt de dades són continus s'ha aplicat la discretització. A continuació, es mostra el resultat.

Nombre d'instàncies	5000
Nombre d'atributs	40
Discretització aplicada	Cert
Tipologia dels atributs	Categòrics
Nombre de categories dels atributs	3
Tipologia de la variable objectiu	Categòrica
Nombre de categories de la variable objectiu	3

TAULA 4.17: Característiques del conjunt de dades *Waveform Database Generator (Version 2)*

Gràcies al previ coneixement que tenim de l'existència d'un gran nombre de variables irrellevants al conjunt de dades, esperem una gran reducció de variables en les diferents execucions de les millores de l'algorisme LVF. Podem observar a la taula que s'ha discretitzat el valor de les variables contínues en tres intervals discrets. Cal remarcar també que s'espera un elevat temps d'execució amb aquest conjunt de dades respecte als anteriors, ja que consta de moltes instàncies i de moltes variables. Ens referirem a aquest conjunt de dades amb l'acrònim CDW.

4.3.8 Conjunt de dades Large Soybean Database

En el conjunt de dades Large Soybean Database[47] trobem com a instàncies, exemplars de soia que pateixen diferents tipus de malalties o lesions. Els atributs d'aquestes instàncies són categòrics i descriuen característiques claus de la planta per a la detecció d'aquests casos. La variable objectiu identifica la malaltia o lesió de la planta.

El conjunt de dades conté un gran nombre de classes a la variable objectiu, concretament 19. En estudiar el conjunt de dades, s'ha observat que 4 classes no es troben justificades, a causa del fet que trobem un nombre ínfim d'instàncies amb aquestes classes. S'ha pres la decisió d'eliminar les instàncies pertanyents a aquestes classes perquè no alterin la classificació del model predictiu, ja que no hi ha les suficients per a realitzar un correcte aprenentatge. A continuació, s'exposen les característiques del conjunt de dades resultant.

Nombre d'instàncies	307
Nombre d'atributs	35
Discretització aplicada	Falç
Tipologia dels atributs	Categòrics
Nombre de categories dels atributs	Variats
Tipologia de la variable objectiu	Categòrica
Nombre de categories de la variable objectiu	15

TAULA 4.18: Característiques del conjunt de dades *Large Soybean Database*

Podem apreciar que és un conjunt de dades amb un bon nombre d'instàncies i de variables. El fet que el fa diferenciar respecte als altres conjunts de dades és la gran quantitat de classes que es troben contingudes en la variable objectiu. Per tant, tenim l'oportunitat d'estudiar si amb aquesta peculiaritat també es reduiran les variables explicatives. Ens referirem a aquest conjunt de dades amb l'acrònim CDB.

4.3.9 Conjunt de dades SPECT Heart

En el conjunt de dades SPECT Heart[48] és descriu el diagnòstic cardíac d'imatges de tomografia computada per emissió de protons únics (SPECT). Cada instància correspon a un pacient al qual se li ha realitzat aquesta prova SPECT, i cada variable de la instància indica si s'ha trobat un determinat patró característic al SPECT.

En conseqüència, tots els atributs del conjunt de dades són binaris. L'objectiu d'aquest conjunt de dades és la classificació de les observacions en normals o anormals segons el diagnòstic dels cardiòlegs. Ens referirem a aquest conjunt de dades amb l'acrònim CDH. Seguidament, trobem les característiques del conjunt de dades.

Nombre d'instàncies	267
Nombre d'atributs	22
Discretització aplicada	Falç
Tipologia dels atributs	Binaris
Tipologia de la variable objectiu	Binària

TAULA 4.19: Característiques del conjunt de dades *SPECT Heart*

Capítol 5

Metodologia d'avaluació de millores

Durant un procés d'investigació és molt important definir correctament una bona metodologia per a la realització de l'experimentació i l'estudi dels resultats, la qual es pugui aplicar uniformement en els diferents casos a valorar i s'adapti perfectament a les necessitats i finalitats del projecte.

A causa del fet que aquest TFG és un projecte d'investigació, s'ha intentat donar molta importància en aquest aspecte del treball. D'aquesta manera s'ha realitzat un bon nombre d'experimentacions sobre les diferents millores implementades del LVF i s'ha recollit una bona quantitat de dades i indicadors d'aquestes experimentacions per poder elaborar un bon anàlisi d'elles i seguir una metodologia avaluativa de qualitat.

En aquest capítol s'explicarà la metodologia d'avaluació que s'ha seguit per a valorar la qualitat de les millores proposades per al LVF, així com les dades i els indicadors necessaris per a l'estudi. També, s'explicaran els classificadors probabilístics seleccionats per l'estudi dels percentatges d'encert dels models predictius generats a partir de les variables seleccionades a les solucions.

5.1 Procés d'avaluació

En aquest projecte entenem com a metodologia avaluativa tot el procés que va des de l'experimentació a l'estudi posterior d'aquesta experimentació. És important estandarditzar aquest procés i aplicar-lo en les mateixes condicions sobre les diferents versions implementades del LVF. En la figura 5.1 podem observar les diferents etapes que formen el nostre procés d'avaluació, els components d'aquesta metodologia seran explicats en les següents seccions.

Cal remarcar que l'objectiu d'aquest procés és poder extreure bones conclusions de les diferents versions implementades. Gràcies a aquestes conclusions s'han pogut detectar les carències i beneficis de les diferents versions i progressivament millorar el seu rendiment basant-nos en els resultats d'aquest procés d'avaluació.

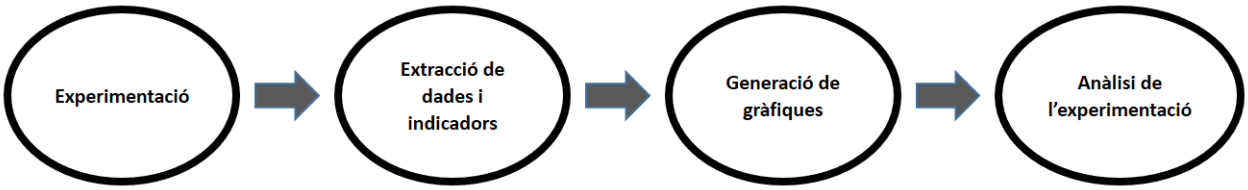


FIGURE 5.1: Diagrama del procés d'avaluació

5.2 Experimentació

En aquesta secció explicarem el primer pas del procés d'avaluació d'una versió qualsevol del LVF, l'experimentació. Sobreentenem que en arribar a aquest procés ja tenim la versió del LVF a tractar correctament implementada i llesta per a ser executada.

L'experimentació l'entenem com a l'execució de la versió del LVF que volem estudiar amb uns paràmetres determinats i amb un dels conjunts de dades esmentats en el capítol 4, Conjunts de dades.

5.2.1 Pautes per a l'experimentació

L'experimentació és un pas clau per a obtenir una bona fiabilitat en les nostres conclusions extretes en l'anàlisi de l'experimentació, per tant, haurem de definir unes pautes per a totes les experimentacions en les quals es busqui una comparació directa de resultats.

- **Execució en el mateix entorn:** És molt important executar les diferents versions implementades del LVF en el mateix entorn, d'aquesta manera podem confiar en els temps d'execució obtinguts, ja que les diferents execucions s'han portat a terme amb un entorn amb les mateixes prestacions. L'entorn en aquest cas serà l'ordinador portàtil de l'autor.
- **Alt nombre de repeticions de l'execució:** Realitzant un gran nombre de repeticions de l'experiment, en aquest cas l'execució d'una versió del LVF, podem minimitzar l'error d'aleatorietat present en un algorisme probabilístic com és el cas del LVF. Totes les versions testades seran executades cinquanta cops per a cada experimentació.
- **Execució amb el mateix grau d'inconsistència:** És molt important també a l'hora de comparar dues versions diferents del LVF sobre un mateix conjunt de dades, executar-les amb el mateix paràmetre que defineix l'acceptació màxima d'inconsistència. En aquest TFG, el grau d'inconsistència definit com a llindar sempre serà l'obtingut en calcular la inconsistència del conjunt de dades amb totes les variables amb el qual s'està treballant. D'aquesta manera si utilitzem el formalisme emprat en el Capítol 2, *Estat de l'art*, i definim X com el conjunt de variables d'un conjunt de dades S, qualsevol solució $\mathcal{A}$ que es presenti pel conjunt de dades S, ha de complir $I(X) \geq I(\mathcal{A})$.
- **Mateix nombre d'iteracions per versió:** Totes les versions de l'algorisme controlen el seu nombre d'iteracions mitjançant el paràmetre *max*, totes les versions de l'algorisme s'executaran amb el paràmetre *max* amb un valor de 100.

- **R com a llenguatge:** Totes les versions seran executades amb el llenguatge de programació emprat en aquest treball de fi de grau, R.

En l'experimentació de les millores inicials que trobem al capítol 6, *Millores inicials del LVF*, en finalitzar l'execució de la versió a comprovar, el subconjunt de variables obtingut com a solució serà seleccionat per a construir un model predictiu classificador. S'ha decidit utilitzar un classificador de la família dels classificadors bayesians ingenus (en anglès, *Naive Bayes*), gràcies al fet que s'adapta de manera excel·lent a les nostres necessitats.

En canvi, en l'experimentació de les millores finals que trobem al capítol 7, *Millores finals del LVF*, per a cada subconjunt de variables resultant de l'execució de la millora, construirem dos models predictius classificadors. El primer model predictiu serà construït amb el mateix classificador que utilitzem per a les millores inicials (*Naive Bayes*) i el segon classificador serà construït amb un aprenentatge basat en un arbre de decisió (en anglès, *decision tree*). Un enfocament de modelatge predictiu simple i que amb variables categòriques usualment dóna un bon rendiment. D'aquesta manera, no només basarem els resultats finals de la nostra experimentació en un sol model predictiu i aconseguirem una robustesa major en els resultats.

5.2.2 Naive Bayes

Els classificadors *Naive Bayes*, són una família de classificadors probabilístics els quals es fonamenten en el teorema de Bayes. El terme *Naive* (en català, ingenu) prové del funcionament d'aquests classificadors, ja que apliquen un seguit d'hipòtesis simplificadores les quals resulten en la hipòtesi d'independència de les variables predictores.

El teorema de Bayes és una proposició plantejada per Thomas Bayes en 1763, la qual permet inferir la probabilitat d'un succés a partir del coneixement que es té de successos relacionats. Aquesta proposició s'expressa amb la següent fórmula:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

on entenem $P(A)$ com la probabilitat de A i $P(A|B)$ com la probabilitat de A condicionada per B, és a dir la probabilitat que succeeixi A sabent que B és veritat. És important remarcar que aquesta condicionalitat no imposa un ordre temporal entre A i B. Es dóna la mateixa interpretació per a $P(B)$ i $P(B|A)$ però per les seves respectives variables.

És important remarcar també que la proposició anterior prové del següent sistema:

$$\begin{cases} P(A|B) = \dfrac{P(A \wedge B)}{P(B)} \\ P(B|A) = \dfrac{P(A \wedge B)}{P(A)} \end{cases} \tag{5.1}$$

Del qual podem extreure:

$$P(A \wedge B) = P(A|B)P(B) = P(B|A)P(A)$$

On entenem $P(A \wedge B)$ com la probabilitat que succeeixi A i B.

Ara que ja tenim explicat el teorema de Bayes, el contextualitzarem amb la família de classificadors *Naive Bayes*. Si volguéssim utilitzar la probabilitat condicionada per a classificar una instància definida com $(x_1, x_2, ..., x_n)$ en una classe (podem entendre classe com una categoria de la seva variable objectiu). Hauríem de cercar la classe c_{max} que maximitzes la probabilitat de la instància pertanyent a ella, tal com expressa aquest formalisme.

$$c_{max} = \arg\max_{c_j \in \mathcal{C}} P(c_j | x_1, x_2, ..., x_n)$$

On $\mathcal{C}$ és el conjunt de classes possibles. Tractar aquest problema amb les taules de probabilitat d'aquesta formulació seria intractable amb un nombre elevat de variables, ja que s'haurien de calcular un total de $(\prod_{i=1}^{n} |x_i|) \times |\mathcal{C}|$ entrades, on entenem $|x_i|$ com el nombre de categories d'aquella variable i n com el nombre de variables totals. Per tant observem que és exponencial respecte el nombre de variables. Per a reduir la complexitat d'aquest càlcul primerament aplicarem el teorema de Bayes:

$$c_{max} = \arg\max_{c_j \in \mathcal{C}} P(c_j | x_1, x_2, ..., x_n) = \arg\max_{c_j \in \mathcal{C}} \frac{P(x_1, x_2, ..., x_n | c_j) P(c_j)}{P(x_1, x_2, ..., x_n)}$$

Com que el denominador $P(x_1, x_2, ..., x_n)$ no depèn de $\mathcal{C}$, no cal que *Naive Bayes* el calculi. Per a reduir la complexitat del càlcul $P(x_1, x_2, ..., x_n | c_j) P(c_j)$ *Naive Bayes* aplica una suposició d'independència entre les variables, el que significa que considera les variables com a independents entre elles. Aquesta suposició, permet realitzar el càlcul d'aquesta manera:

$$P(x_1, x_2, ..., x_n | c_j) P(c_j) = \prod_{i=1}^{n} P(x_i | c_j)$$

La qual cosa ens permet reduir el nombre d'estimacions de probabilitat, quedant així $(\sum_{i=1}^{n} |x_i|) \times |\mathcal{C}|$, un nombre d'entrades fàcil de computar. Per tant, *Naive Bayes* realitza la cerca de la classe que maximitza la probabilitat de què la instància pertanyi a ella d'aquesta manera:

$$c_{max} = \arg\max_{c_j \in \mathcal{C}} \frac{P(x_1, x_2, ..., x_n | c_j) P(c_j)}{P(x_1, x_2, ..., x_n)} = \arg\max_{c_j \in \mathcal{C}} P(c_j) \prod_{i=1}^{n} P(x_i | c_j)$$

5.2.3 Naive Bayes en els nostres conjunts de dades

Gràcies a la suposició d'independència entre variables i a la simplicitat d'aquesta família de classificadors el temps de construcció del classificador no és elevat, és relativament baix. Un requisit indispensable pel nostre treball, ja que haurem de construir molts classificadors diferents. La generació de models predictius no pot fer de coll d'ampolla a la nostra investigació.

Aquesta suposició d'independència de variables comportarà una precisió menor respecte altres classificadors més complexos, però si apliquéssim aquests classificadors més complexos, no gaudiríem d'un temps d'execució baix i se'ns faria impossible realitzar totes les experimentacions plantejades. Aquesta suposició afectarà

especialment al conjunt de dades de paritat (CDP), ja que en entendre les variables com independents s'accentuarà el principi de similitud del qual s'ha parlat en el Capítol 4, *Conjunts de dades*. Aquest fet s'ajunta també que en aquest conjunt de dades, totes les variables són dependents entre elles, tot junt causarà prediccions poc precises per part del nostre classificador.

Una altra particularitat clau per a la selecció d'aquesta família de classificadors ha estat l'excel·lent adaptació que tenen amb els conjunts de dades amb variables categòriques o numèriques discretes, la tipologia de variables que utilitzen els nostres conjunts de dades.

La implementació seleccionada de la família de classificadors *Naive Bayes* ha estat la que trobem en el paquet klaR [49] de R, una implementació estàndard del *Naive Bayes*. Per a cada conjunt de dades adaptarem el paràmetre m del *Laplace smoothing*.

El *Laplace smoothing* és una tècnica que s'utilitza en estadística per suavitzar les dades categòriques. Aquesta tècnica és introduïda per a resoldre el problema de la probabilitat zero. Aquest problema consisteix a propagar el valor zero d'una probabilitat. Com s'ha comentat anteriorment, *Naive Bayes* assumeix la independència entre variables, per tant realitza el càlcul de probabilitats amb un productori així; $P(x_1, x_2, ..., x_n|c_j)P(c_j) = \prod_{i=1}^{n} P(x_i|c_j)$, en conseqüència quan una d'aquestes probabilitats sigui $P(x_i|c_j) = 0$ aquest fet es propagarà i independentment dels valors de les altres probabilitats s'obtindrà $P(x_1, x_2, ..., x_n|c_j)P(c_j) = 0$.

El *Laplace smoothing* canviarà el càlcul de les probabilitats perquè no s'obtinguin probabilitats amb valor zero i en conseqüència evitem la propagació. Aquest càlcul es realitzarà de la següent manera; definim n com el nombre d'instàncies amb $c = c_j$, n_c com el nombre d'instàncies amb $c = c_j$ i $x = x_i$, p és una estimació a priori de probabilitat de $p(x_i|c_j)$ i m és una constant que determina el pes de p en relació amb les dades observades.

$$p(x_i|c_j) = \frac{n_c + mp}{n + m}$$

Nosaltres utilitzarem per p les proporcions de classe del nostre conjunt de dades i per a m el nombre de classes. D'aquesta manera no obtindrem probabilitats amb valor zero, ja que no tractarem amb classes que no tinguin representació en el conjunt de dades (p o m mai tindran valor zero).

5.2.4 Arbre de decisió

Utilitzarem un aprenentatge basat en arbres de decisió, el qual utilitzarà un arbre de decisió com a model predictiu. Aquest algorisme funciona particionant repetidament les dades en diversos subespais, de manera que els resultats de cada subespai final siguin tan homogenis com sigui possible. Aquest enfocament s'anomena tècnicament particionament recursiu.

Aquestes particions en diferents subespais és realitzant mitjançant les regles de divisió. Aquestes regles es produeixen dividint repetidament les variables predictores (en el nostre cas en modalitats), començant per la variable que té més associació amb la variable objectiu. El procés continua fins que es compleixen alguns criteris d'aturada predeterminats.

Podem dividir els arbres de decisió en dos tipus, segons la tipologia de la variable a predir:

- **Arbres de regressió**: Quan la variable objectiu és contínua.
- **Arbres de classificació**: Quan la variable objectiu és discreta.

Com que en tots els nostres conjunts de dades acomplim una tasca de classificació amb variables discretes, utilitzarem els arbres de classificació. En conseqüència, aquesta explicació té un enfocament directe en ells.

Aquest arbre de decisió està format per nodes de decisió, branques i nodes fulla. L'arrel estarà a dalt i els nodes fulla a baix, l'arbre creixerà des de l'arrel i guanyarà profunditat. Cada node de decisió correspon a un únic atribut i a una regla de divisió d'aquest atribut, el qual separarà les instàncies en modalitats d'aquest atribut (ja que només tractem amb variables discretes). Els nodes fulla de l'arbre, representen les classes de la variable objectiu amb les quals s'han de classificar les instàncies.

Utilitzarem la implementació anomenada *rpart*[50] (*Recursive Partitioning And Regression Trees*), la qual trobem en el paquet *rpart*[51] de R. Amb aquesta implementació, l'arbre deixarà de créixer quan es compleixen algun dels tres criteris següents[52]:

- Tots els nodes fulla d'un arbre pertanyen a una mateixa classe.
- Cap regla de divisió pot particionar les instàncies assolint un nombre mínim predeterminat en cada node fulla.
- El nombre d'instàncies al node fulla arriba al mínim predeterminat.

Imposem aquest nombre mínim d'instàncies a cada node fulla per a evitar d'aquesta manera el problema del *overfitting*, és a dir, controlem la profunditat d'aquest arbre amb la finalitat de no adaptar-lo en excés al conjunt de dades d'entrenament i d'aquesta manera poder generalitzar el model predictiu amb altres instàncies i obtenir un bon resultat.

Les regles de divisió del nostre arbre de classificació es basen en el fet que la població de les subparticions sigui la més pura possible. Amb la paraula pura, ens referim que en cada subpartició apareguin el mínim nombre d'instàncies amb classes de la variable objectiu que no corresponen a aquell subpartició. Les dues mesures més utilitzades per aquesta mesura de puresa són la impuresa de Gini i el guany d'informació (Entropia).

Nosaltres utilitzarem la impuresa de Gini, la qual calcula el grau d'impuresa seguint aquesta fórmula:

$$I_G(s) = \sum_{i=1}^{m} s_i(1 - s_i)$$

On $I_G(s)$ significa la impuresa de Gini del conjunt d'instàncies s, s_i és la proporció d'instàncies respecte al conjunt total s que pertanyen a la classe i, on m és el nombre total de classes de la variable objectiu. La impuresa de Gini ens donarà un valor dins del rang $[0, 1]$, on 0 significa el valor mínim d'impuresa i 1 la màxima impuresa. Aquest índex s'aplicarà a totes les subparticions i mitjançant una mitjana dels resultats l'algorisme sabrà la bondat d'aplicar una regla de divisió.

5.2.5 Arbre de decisió en els nostres conjunts de dades

Els arbres de decisió són models predictius simples, igual que *Naive Bayes*, però tenen un enfocament diferent. En ser fàcils de computar i tenir un comportament molt diferent del classificador ja utilitzat, són ideals per al nostre projecte, ja que podrem observar diferències entre l'ús de diferents classificadors i la reducció de variables.

El *Naive Bayes* utilitzarà totes les variables del conjunt de dades per a realitzar la classificació, en canvi, els arbres de decisió normalment no. Ja que si aquests arbres de decisió utilitzessin variables que aporten soroll en la classificació, es generaria *overfitting* en el model predictiu i les seves prediccions amb dades noves no serien bones.

Per a evitar aquest fet, és molt important podar el creixement de l'arbre abans de l'aparició d'aquest fenomen. Per tant, s'haurà de predefinir el nivell de complexitat que podrà tenir l'arbre de classificació resultant perquè en l'avaluació creuada no s'obtingui una *accuracy* molt baixa. Aquest procediment era necessari automatitzar-lo, ja que per cada subconjunt de variables i cada conjunt de dades aquesta complexitat variarà, i si definim un estàndard perdrem molta *accuracy* en les nostres prediccions.

Per a l'automatització de la selecció de la complexitat de l'arbre de decisió, s'utilitzarà la tècnica del *one-standard-error rule* que dicta; seleccionar el model predictiu amb menor complexitat que no sigui més que un error estàndard pitjor que el millor model quant a l'error de la validació encreuada[53]. Per tant, cada cop que vulguem entrenar un model basant el seu aprenentatge en arbres de decisió, construirem diferents arbres de decisió amb diferents complexitats, realitzarem amb cadascun d'ells una validació encreuada de 20 iteracions i seleccionarem com a model predictiu l'arbre amb menor complexitat i una mitjana de l'*accuracy* que no sigui més que un error estàndard pitjor de l'*accuracy* mitjana del millor arbre de decisió.

Aquest procediment ens garanteix de solucions amb un bon nivell d'*accuracy* (respecte al màxim possible) i un valor d'*accuracy* fiable, ja que en seleccionar els arbres amb menor complexitat ens evitem cap classe de biaix degut al *overfitting*.

En conclusió, els arbres de decisió utilitzaran només aquelles variables més rellevants del conjunt de dades, i basaran la seva predicció en menys variables. Per tant, si la nostra reducció de variables selecciona les variables més rellevants del conjunt de dades molt possiblement, tindrem molt bones prediccions amb menys variables.

5.3 Extracció de dades i indicadors numèrics

En aquesta secció s'expliquen totes les dades que extraiem de l'experimentació, i la creació d'un indicador que ens serveix per avaluar el grau de correctesa de les solucions donades a l'experimentació. És important seleccionar correctament les dades les quals es volen analitzar posteriorment en l'anàlisi de l'experimentació per a facilitar l'obtenció de conclusions.

5.3.1 Dades importants a extreure

A continuació, s'enumeren i s'expliquen les dades que s'extreuen per cada execució de qualsevol versió del LVF. Cal remarcar, com s'ha comentat en la secció anterior, que en cada experimentació es realitzen cinquanta execucions amb la mateixa versió, per tant tindrem cinquanta valors de cada dada extreta.

- **Subconjunt solució:** Nom de les variables seleccionades com a solució per l'execució de l'algorisme.
- **Mida del subconjunt de variables:** Mida del subconjunt de variables resultat de l'execució de l'algorisme.
- **Variables rellevants:** Nombre de variables rellevants seleccionades a la solució de l'algorisme.
- **Variables redundants:** Nombre de variables redundants seleccionades a solució de la versió del LVF.
- **Variables irrellevants:** Nombre de variables irrellevants seleccionades a la solució de l'algorisme.
- **Temps total d'execució:** Temps que ha trigat la versió del LVF en executar-se.
- **Precisió del classificador:** Precisió obtinguda del model predictiu construït a partir de les variables seleccionades a la solució de la versió de LVF.
- **Diferència de precisió:** Diferència de precisió respecte a la precisió obtinguda amb el classificador construït amb totes les variables del conjunt de dades envers el classificador construït amb les variables seleccionades a la solució de l'algorisme.

Totes aquestes dades ens serviran com a paràmetres avaluadors de la solució, i els podrem emprar en l'anàlisi de l'experimentació per a tenir una major informació del comportament de les diferents versions experimentades del LVF.

Cal destacar, que pels conjunts de dades finals no es realitzarà la classificació de variables en variables rellevants, irrellevants o redundants. Això és degut al fet que com no són generats per nosaltres no tenim una gran precisió sobre aquesta classificació. Es van intentar utilitzar eines estadístiques com el test d'independència de la chi-quadrat per a la detecció de variables redundants, però els resultats no eren massa clars, ja que encara que fossin dues variables dependents podien aportar diferent informació, fet que no entra dins la definició de redundant d'aquest projecte. Per tant, les dades; variables rellevant, variables irrellevants i variables redundants no s'extrauran dels conjunts de dades finals.

L'extracció d'aquests paràmetres s'ha realitzat amb la metodologia següent; per a cada iteració de l'experimentació s'ha desat el resultat de l'execució de la versió del LVF a comprovar i també si hi ha guardat el seu temps d'execució. t_{exe} (restant el temps inicial del sistema t_o del temps del sistema un cop finalitzada l'execució t_f, $t_{exe} = t_f - t_o$).

Posteriorment si treballem amb un conjunt de dades inicial, amb el subconjunt resultat de la versió del LVF es classifiquen les variables segons la seva tipologia

(variables rellevants, variables redundants i variables irrellevants), aquest procés és simple gràcies al fet de tenir anomenades les variables segons la seva tipologia. És important, però, quan trobem una variable redundant en la solució comprovar que també hi trobem la variable rellevant que redunda o una altra variable redundant que repeteix la mateixa variable en el subconjunt, si no aquesta variable passa a ser rellevant, ja que no repetirà a cap en el subconjunt de variables solució.

Per a finalitzar, extraiem la precisió obtinguda del classificador construït amb la solució de l'execució del LVF. Per a obtenir la diferència de precisió realitzem la següent subtracció $P(C_{\mathcal{A}}) - P(C_X)$. On definim $P : \mathcal{P}(X) \rightarrow [0, 1]$, que expressa la precisió del classificador, $C_{\mathcal{A}}$ el classificador construït amb el subconjunt de variables solució i C_X el classificador construir amb totes les variables del conjunt de dades original X.

5.3.2 Precisió dels models predictius

Per a la validació dels models predictius construits, s'utilitzarant dues tècniques diferents. Amb aquestes tècniques és té com a objectiu obtenir una molt bona aporximació de l'*accuracy* dels models predictius.

La primera tècnica serà emprada per els models predictius construits a partir dels conjunts de dades inicials. Aquests conjunts de dades tenen la particularitat que el seu origen és artificial i estan generats per nosaltres mateixos. En conseqüència podem generar el nombre d'instàncies que desitjem. D'aquesta manera, emplearem una validació creuada on el conjunt de dades d'entrenament serà l'emprat per a la reducció de variables (descrit en el capítol 4, *Conjunts de dades*) i el conjunt de dades de prova serà un conjunt de dades generat tal i com és decriu en el capítol 4, *Conjunts de dades*, però amb un nombre molt elevat d'instàncies (50000).

Al realitzar una validació amb un conjunt de dades de prova amb tantes instàncies i tant equilibrat es garanteix una *accuray* molt pròxima a la real.

La segona tècnica que utilitzarem serà emprada per els models predictius construits a partir dels conjunts de dades finals. A causa del fet que aquests conjunts de dades no són generats per nosaltres i són reals, no podem controlar la generació de noves instàncies correctament. Per tant, utilitzarem un altre enfocament. Farem ús d'una validació creuada de 20 iteracions (en anglés, *20-fold cross-validation*) per al càlcul de l'*accuracy* dels models predictius. Per tant, cada conjunt de dades original descrit en el capítol 4, *Conjunts de dades*, serà dividits en 20 subconjunts de dades amb els quals és contruirant 20 models predictius, un per cada iteració, on en cada iteració seleccionem un subconjunt de dades com a conjunt de dades de prova (cada iteració amb un diferent) i la resta com a conjunt de dades d'entrenament.

L'*accuracy* serà calculada com la mitjana aritmètica de les 20 iteracions. Considerem que utilitza 20 iteracions són suficients per a obtenir una *accuracy* fiable, a causa del fet que amb els conjunts de dades amb els quals traballem ja tenen un bon nombre d'instàncies.

5.3.3 Indicador avaluador dels subconjunts de variables

Els paràmetres avaluadors com la precisió del classificador, la diferència de precisió i el temps total d'execució podem estudiar-los directament sense aplicar cap transformació en ells. Però en el cas dels conjunts de dades inicials tenim els paràmetres avaluadors; variables rellevants, variables irrellevants i variables redundants, amb els quals s'ha d'idear un indicador perquè tracti conjuntament aquests tres paràmetres i pugui puntuar la correctesa de la solució basant-se en la classificació de les variables.

Per a poder realitzar aquesta qualificació numèrica s'ha decidit adaptar un indicador emprat pel director del projecte en l'estudi *Review and Evaluation of Feature Selection Algorithms in Synthetic Problems*[54], el qual ens permetrà mitjançant les dades anteriors expressades, avaluar la bondat de la solució resultant de l'execució de la versió de l'algorisme LVF.

Descripció de l'indicador

Aquesta mesura d'avaluació l'anomenarem s i es basarà en un criteri de similitud entre la solució a avaluar i al subconjunt de variables òptim. Cal remarcar que si trobem variables redundants en el conjunt de dades, molt possiblement el subconjunt de variables òptim no serà únic. Definim formalment el criteri de similitud $s : \mathcal{P}(X) \times \mathcal{P}(X) \to [0,1]$ entre subconjunts de X, on $s(X_1, X_2) > s(X_1, X_3)$ indica que X_2 és més similar a X_1 que X_3, també satisfent que $s(X_1, X_2) = 1 \iff X_1 = X_2$ i $s(X_1, X_2) = s(X_2, X_1)$. Entenem X com el conjunt total de variables, el qual podem particionar d'aquesta manera $X = X_R \cup X_I \cup X_{R'}$, sent $X_R, X_I, X_{R'}$ els subconjunts de variables rellevants, irrellevants i redundants respectivament. Anomenem com X^* qualsevol subconjunt de variable amb la solució òptima i $\mathcal{A}$ com el subconjunt de variables resultant de l'execució de la versió del LVF.

Definim $\mathcal{A}_R = \mathcal{A} \cap X_R$, $\mathcal{A}_I = \mathcal{A} \cap X_I$ i $\mathcal{A}_{R'} = \mathcal{A} \cap X_{R'}$. Definim també $\mathcal{A}_T = \mathcal{A} \cap X_T$, on T significa un d'aquests subíndexs $\{R, I, R'\}$. D'aquesta manera tenim $\mathcal{A} \subseteq X$ i $\mathcal{A} = \mathcal{A}_R \cup \mathcal{A}_I \cup \mathcal{A}_{R'}$. Definim en termes de similitud l'avaluador $S_x(\mathcal{A}) : \mathcal{P}(X) \to [0,1]$ per a tot $\mathcal{A} \subseteq X$, $S_x(\mathcal{A}) = s(\mathcal{A}, X^*)$. On $S_x(\mathcal{A}) > S_x(\mathcal{A}')$ indica que $S_x(\mathcal{A})$ té més similitud a X^* que $S_x(\mathcal{A}')$. Aquest avaluador numèric $S_x(\mathcal{A})$ permet penalitzar tres situacions possibles:

- Falta de variables rellevants a $\mathcal{A}$.
- Aparició de variables irrellevants a $\mathcal{A}$.
- Aparició de variables redundants a $\mathcal{A}$.

Definirem un nou conjunt de paràmetres $\alpha = \{\alpha_R, \alpha_I, \alpha_{R'}\}$, els quals indicaran el pes de les diferents penalitzacions anteriors. Les α_T s'entenen respectivament en l'ordre que s'han representat les situacions penalitzables i han de complir $\sum \alpha_T = 1$.

Cal remarcar, que les variables rellevant i les variables redundants estan fortament relacionades, ja que una variable és redundant depenent de quines variables rellevants es trobin a $\mathcal{A}$. D'aquí que possiblement la solució òptima no sigui única si existeix redundància a X. Per a expressar aquesta redundància, classificarem les variables de X en classes d'equivalència, on a cada classe trobarem les variables que

aporten la mateixa informació. Per tant, definim formalment una relació binària entre dues variables $x_i, x_j \in X$ com: $x_i \sim x_j \iff x_i$ i x_j representen la mateixa informació. Per tant $\sim$ és una relació d'equivalència, definim $X/\sim$ com el conjunt quocient de X sota $\sim$. Cal observar que tota solució òptima haurà de tenir la mateixa mida de X_R i tenir un element de cada subconjunt $(X_R \cup X_{R'})/\sim$.

Construcció de l'indicador

Per a la construcció d'aquest indicador necessitem definir també els següents conjunts; $\rho_{\mathcal{A}} = (\mathcal{A}_R \cup \mathcal{A}_{R'})/\sim$ (les classes d'equivalència on trobem les variables rellevants i redundants barrejades per al subconjunt solució de la versió del LVF), $\rho_X = (X_R \cup X_{R'})/\sim$ (el mateix però pel conjunt original de variables) i $\rho_{\mathcal{A}\subseteq X} = \{x \in \rho_X \mid \exists y \in \rho_{\mathcal{A}}, y \subseteq x\}$ (el subconjunt format per les classes d'equivalència senceres de ρ_X les quals contenen un o més elements que apareixen a $\rho_{\mathcal{A}}$). Per finalitzar definirem pel conjunt Q, $F(Q) = \sum_{x\in Q}(|x| - 1)$. La qual cosa ens permet expressar el quocient entre el nombre de variables redundants seleccionades pel LVF i la suma de la mida de les classes d'equivalència que tenen algun element en $\mathcal{A}$, d'aquesta manera:

$$\frac{F(\rho_{\mathcal{A}})}{F(\rho_{\mathcal{A}\subseteq X})}$$

Cal destacar que tant en el numerador, tant com en el denominador la mida de cada classe d'equivalència se li resta 1, això és degut al fet que aquesta variable restada és la variable rellevant.

Ara ja estem llestos per definir com avaluarem les tres incidències descrites anteriorment. La falta de variables rellevants l'avaluarem de la següent manera:

$$R_A = \frac{|\rho_{\mathcal{A}}|}{|X_R|}$$

És a dir, el quocient entre el nombre de variables rellevants pertanyents a $\mathcal{A}$ i el nombre total de variables rellevants de X. L'aparició de variables rellevants serà avaluada així:

$$I_A = 1 - \frac{|\mathcal{A}_I|}{|X_I|}$$

La divisió entre el nombre de variables irrellevants trobades a $\mathcal{A}$ i el nombre de variables irrellevants de X restades a 1. Per acabar, l'aparició de variables redundants serà avaluada amb la següent expressió:

$$R'_{\mathcal{A}} = \begin{cases} 0 & \text{si } F(\rho_{\mathcal{A}\subseteq X})) = 0 \\ 1 - \frac{F(\rho_{\mathcal{A}})}{F(\rho_{\mathcal{A}\subseteq X})} & \text{altrament} \end{cases}$$

Un cop definides les anteriors avaluacions, hem d'ajuntar els resultats i expressarlos conjuntament en un sol indicador amb rang [0,1]. Aquí entren en joc els paràmetres α_T definits anteriorment, els quals defineixen el pes de cada incidència:

$$S_X(\mathcal{A}) = s(\mathcal{A}, X^*) = \alpha_R R_{\mathcal{A}} + \alpha_I I_{\mathcal{A}} + \alpha_{R'} R'_{\mathcal{A}}$$

Definirem ara l'ordre de severitat de les incidències. La més severa serà la mancança de variables rellevants, a causa del fet que baixaran en gran manera el rendiment del classificador. La seguirà l'aparició de variables irrellevants, ja que aportaran soroll al conjunt de dades i podrà afectar directament també al rendiment del classificador. Per últim, la incidència menys severa serà l'aparició de variables redundants, ja que no aporten informació enganyosa, però tampoc aporten informació beneficiosa. Per al compliment de l'ordre de severitat exposat, s'hauran de complir aquestes dues condicions:

- Seleccionar una variable irrellevant és millor que no seleccionar una variable rellevant: $\frac{\alpha_R}{|X_R|} > \frac{\alpha_I}{|X_I|}$
- Seleccionar una variable redundant és millor que seleccionar una variable irrellevant: $\frac{\alpha_I}{|X_I|} > \frac{\alpha_{R'}}{|X_{R'}|}$

Definim $\underline{\alpha}_T = \frac{\alpha_T}{|X_T|}$, amb l'objectiu d'expressar que no té la mateixa severitat seleccionar una variable irrellevant quan en trobem tres en el conjunt de dades original que quan en trobem quatre en el conjunt de dades original. Aquest exemple el podem extrapolar a les altres dues situacions a penalitzar. Per la correctesa d'aquest nou paràmetre definim; $(|X_T| = 0) \rightarrow (\alpha_T = 0)$, d'aquesta manera evitem caure en el clàssic error lògic de la divisió per 0.

Per finalitzar, declarem les següents equacions per a assolir el compliment de les anteriors inequacions exposades:

$$\frac{\gamma}{2}\underline{\alpha}_R = \underline{\alpha}_I \qquad \frac{2\gamma}{3}\underline{\alpha}_I = \underline{\alpha}_{R'} \qquad \alpha_R + \alpha_I + \alpha_{R'} = 1$$

On $\gamma \in (0, 1]$ expressa la relació de pesos entre α_T. En aquest projecte utilitzarem $\gamma = 1$ ja que d'aquesta manera quan $|X_R|, |X_R|$ i $|X_R|$ són equivalents, obtenim les següents relacions:

- α_R és almenys dues vegades més important que α_I.
- α_I és almenys 1.5 vegades més important que $\alpha_{R'}$.

Optimització de l'indicador:

Per a una major precisió d'aquest indicador s'ha optimitzat la seva avaluació respecte a la mancança de variables rellevants (R_A). Definim $rel : \mathcal{P}(x) \rightarrow [0, 1]$ com la rellevància que té la variable x en el conjunt de dades X. Aquests valors estan normalitzats, és a dir, s'ha de complir $\sum_{x \in X} rel(x) = 1$.

La versió anterior R_A no té en compte la rellevància individual de cada variable, és a dir, tracte totes les variables com si aportessin la mateixa informació al conjunt de dades, fet que rarament succeeix en els conjunts de dades. Exemplificarem el cas; si tenim dos subconjunts de variables Y' i Y'' que contenen les següents variables rellevants, $Y'_R = \{x_1, x_2, x_4\}$ i $Y''_R = \{x_2, x_3, x_4\}$, sabem que la rellevància de les variables en el conjunt de variables rellevants original Y_R és $\{rel(x_1) = 0.3, rel(x_2) = 0.3, rel(x_3) = 0.1, rel(x_4) = 0.3\}$, la nostra solució obtindria $R_{Y'_R} = R_{Y''_R}$, quan clarament la solució Y'_R és millor pel fet que es compleix $(\sum_{x \in Y'_R} rel(x)) > (\sum_{x \in Y''_R} rel(x))$, per tant conté una rellevància total més alta.

En conseqüència, per a evitar aquest problema i tenir un indicador de major qualitat, redefinim l'avaluació de la mancança de variables rellevants com a:

$$R^o_{\mathcal{A}} = \sum_{x \in \widetilde{\rho}_{\mathcal{A}}} rel(x)$$

On entenem $\widetilde{\rho}_{\mathcal{A}}$ com el subconjunt de variables format per cada variable representant de cada classe d'equivalència de $\rho_{\mathcal{A}}$. Entenem variable representant d'una classe d'equivalència $e \in \rho_{\mathcal{A}}$, com qualsevol variable pertanyent a la classe d'equivalència e, ja que totes les variables que es troben en la mateixa classe d'equivalència, aporten la mateixa informació (per la definició de les classes d'equivalència de $\rho_{\mathcal{A}}$).

Per a la correcta aplicació del nou avaluador R^o, també hem de redefinir $S_X(\mathcal{A})$ incorporant el nou avaluador R^o.

$$S_X(\mathcal{A}) = s(\mathcal{A}, X^*) = \alpha_R R^o_{\mathcal{A}} + \alpha_I I_{\mathcal{A}} + \alpha_{R'} R'_{\mathcal{A}}$$

Aquest indicador optimitzat és l'utilitzat durant el transcurs del projecte i serà anomenat *score*. La seva funció serà la reducció de la classificació per tipologia de variables (rellevant, irrellevant, redundant) dels conjunts de dades inicials a un sol indicador.

5.4 Generació de gràfiques

En aquesta part del procés d'avaluació de les millores del LVF, es realitzarà la representació visual dels paràmetres avaluadors i l'indicador *score* (en els conjunts de dades inicials) extrets en l'experimentació.

La generació de les gràfiques és important realitzar-la correctament, ja que en tenir tantes dades per paràmetre avaluador s'ha d'intentar perdre la menor informació possible en la representació gràfica. També, serà de gran importància a causa del fet que en gran part les conclusions del procés d'avaluació, seran condicionades per la qualitat d'aquestes gràfiques. En el projecte trobarem dos tipus de gràfics, els quals s'exposen a continuació:

- **Diagrama de caixa**: Ens referirem a ell amb el seu terme anglès, *boxplot*. Aquest tipus de gràfic ens permet representar les dades perdent menys informació que altres tipus de gràfics, ja que permet expressar no només la mediana d'uns certs valors, sinó que també permet expressar el recorregut interquartílic (la "caixa"), on l'extrem inferior simbolitza el quartil Q1 i l'extrem superior el quartil Q3.
- **Diagrama de barres**: Serà utilitzar per a representar diferents distribucions de probabilitat discretes utilitzades durant el projecte.

Durant aquest TFG s'utilitzarà el *boxplot* per sobre del gràfic lineal pel fet que gràcies a ell i el recorregut interquartílic podrem expressar si una millora presenta molta variància entre les seves solucions o no, fàcilment. El mateix es pot aplicar en el temps d'execució i altres dades de l'estil.

5.5 Anàlisi de l'experimentació

Aquest és l'últim pas en el procés d'avaluació d'una millora del LVF, en aquest pas estudiarem les gràfiques obtingudes en aquest procés d'avaluació i compararem els resultats amb els resultats d'altres versions del LVF.

Aquesta comparació de resultats ens indicarà el rendiment de la millora introduïda en aquesta versió del LVF i podrem observar l'evolució que es va adquirint respecte a la versió original del LVF.

En la fase inicial de la investigació de les millores, aquesta anàlisi tindrà com a objectiu verificar si la millora proposada augmenta el rendiment de la millor versió obtinguda fins ara del LVF. Aquest fet és degut a la metodologia incremental que utilitzem en aquesta part del projecte per a obtenir millores del LVF. Per tant, si l'anàlisi ens indica que es troba evidència que la millora proposada augmenta el rendiment de l'anterior millor versió proposada, la nova millora passarà a ser la millor versió actual del LVF.

En l'experimentació d'aquestes millores inicials, les conclusions de l'anàlisi de l'experimentació estaran basats principalment en l'indicador *score*, el temps d'execució, *accuracy* del classificador i diferència de *accuracy*, els quals han estat explicats en la secció *Extracció de dades i indicadors numèrics*, d'aquest capítol.

En l'última fase de la investigació, però, la finalitat principal d'aquesta anàlisi serà documentar el rendiment de la versió a testar en un entorn real, per tant aquests resultats hauran de ser tractats amb molta delicadesa, ja que seran els resultats finals del projecte i el rendiment real de la versió testada.

A causa del fet que amb els conjunts de dades finals no tenim l'indicador *score* (ja que no classifiquem les dades), les conclusions de l'anàlisi de l'experimentació amb les versions finals es basaran en la mida del subconjunt de variables, el temps d'execució, l'*accuracy* i la diferència d'*accuracy* dels dos classificadors construïts (amb *Naive Bayes* i arbres de decisió).

Capítol 6

Millores inicials del LVF

En aquest capítol explicarem les millores que es van proposar i implementar en la fase inicial del projecte. Com s'ha comentat en el capítol 3, *Metodologia d'avaluació de millores*, aquesta primera fase de les millores partiran de l'algorisme LVF original, per tant sense cap modificació prèvia, i intentarem aplicar petites modificacions per a millorar el seu rendiment.

A cada millora proposada s'aplicarà el procés d'experimentació explicat en el Capítol 3, *Metodologia d'avaluació de millores*, en tots els conjunts de dades de la fase inicial presentats. Les modificacions que millorin aquestes avaluacions seran afegides a la millor versió obtinguda, per tant seguirem una metodologia incremental respecte a la creació de la millora final que es plantejarà.

Les experimentacions a les quals ens referim en aquest capítol, es troben a la secció *Resultats millores inicials* del capítol 13, *Resultats de l'experimentació*. És important remarcar que per motius d'extensió de la memòria s'ha preferit no exposar gràfics repetits durant el text que ja es troben ben contextualitzats en el capítol esmentat.

6.1 Inconsistència

En la secció *Las Vegas Filter*, del Capítol 2, *Estat de l'art*, hem definit formalment i explicat les característiques de la mesura d'avaluació que utilitzaran les nostres modificacions de l'algorisme LVF, la inconsistència. En aquesta secció explicarem l'algorisme que utilitzaran totes les nostres millores de la fase inicial per al càlcul de la inconsistència. A continuació s'hi exposa el pseudocodi:

Algorisme 13: Càlcul inconsistència

Entrada:
$S(X)$ - Conjunt de dades S descrit per X, on $n = |X|$ i x_n és la variable objectiu
Sortida:
Inconsistència - Inconsistència de $S(X)$

$X_d := \{X \backslash x_n\}$
$S_u(X_d) := uniques(S(X_d))$
$m := 0$
for each $i \in S_u(X_d)$ **do**
  $S_i(X) := iguals(i, S(X))$
  $m := m + max(S_i(X_d), S_i(X))$
  $S(X) := S(X) \backslash S_i(X)$
end
Inconsistència $:= \frac{(|S(X)| - m)}{|S(X)|}$

Com podem observar en el codi, primerament obtindrem totes les instàncies úniques S_u del nostre conjunt de dades S descrit per X, entenem com a instàncies úniques el conjunt d'instàncies no repetides descrites per les variables descriptives X_d, on definim X_d com el conjunt de totes les variables menys la variable objectiu x_n. La funció *uniques()* du a terme aquesta tasca en el codi.

Per a cada instància de S_u, cercarem a S totes les instàncies que tenen els mateixos valors a les variables X_d (realitzat per la funció *iguals()* en el codi), posteriorment per a cada subconjunt d'instàncies repetides cercarem quina és la classe de la variable objectiu x_n que més representació té i obtindrem la seva quantitat d'aparicions totals en el subconjunt (realitzat per la funció *max()* al codi), definim aquest valor com m_i on la i identifica el subconjunt d'instàncies repetides. Definim també $m = \sum_{i=1}^{|S_u|} m_i$ i finalment podem definir el càlcul final de la inconsistència:

$$\textit{inconsistència} = \frac{(|S(X)| - m)}{|S(X)|}$$

Cal destacar que si en la funció *uniques()* per a cada instància única obrim una entrada en una taula de *hash* i introduïm com a valors les instàncies que tenen els mateixos valors per les variables X_d conjuntament amb les seves respectives variables objectius x_n, quan executem la funció *iguals()* podrem consultar la taula de *hash* per a cercar les instàncies repetides per a cada instància única d'una manera més eficient. Utilitzant aquest mecanisme de *hash* podem calcular la inconsistència de $S(X)$ aproximadament amb un cost computacional de $\mathcal{O}(|S|)$, on $|S|$ és el nombre d'instàncies totals del conjunt de dades. Fet que diferència aquesta mesura d'avaluació de les altres mesures univariades utilitzades comunament, ja que solen tenir costos computacionals més elevats. D'aquesta manera definirem el cost computacional del càlcul de la inconsistència com $\mathcal{O}(|S|)$.

És important també remarcar que existeix la possibilitat de treballar amb la consistència, la qual ve definida amb la següent expressió: $\textit{consistència} = 1 - \textit{inconsistència}$. Nosaltres, però, evitarem la transformació i treballarem amb la inconsistència.

6.2 Versió original

Com a versió inicial utilitzarem la versió original del LVF presentada en el capítol 2, *Estat de l'art*. Recordem que aquesta versió utilitza una generació de successors (solucions candidates) totalment aleatòria. Aquesta propietat presenta una característica molt beneficiosa pel problema de la selecció de variables; la possibilitat de moure'ns per tot l'espai de possibles solucions i conseqüentment no limitar la cerca en un subespai de solucions similars. Aquesta propietat és difícil de trobar en molts algorismes *greedy*, ja que usualment limiten la seva cerca a solucions similars al màxim local trobat. Per tant, haurem d'intentar mantenir aquesta propietat durant el transcurs de les millores.

Per altra banda, aquesta propietat condiciona molt el nivell de fiabilitat de la solució donada per l'algorisme. Podem apreciar-ho en l'experimentació realitzada,

ja que en la majoria de conjunts de dades hem obtingut un *score* amb una gran variabilitat. Per a millorar la fiabilitat de l'algorisme haurem de reduir la seva variabilitat. En conclusió, les millores proposades tindran l'objectiu de trobar un equilibri entre la reducció d'aquesta variabilitat i la conservació de la possibilitat de moure'ns per tot l'espai de possibles solucions.

A continuació, recordem l'algorisme LVF per a una millor contextualització de les millores.

Algorisme 14: Las Vegas Filter

Entrada:
max - el nombre màxim d'iteracions
$\mathcal{J}$ - la mesura d'avaluació
$S(X)$ - una mostra S descrita pel conjunt de variables X
Sortida:
L - Llista de les solucions equivalents trobades

```
L := []
Best := X
J_o := J(S(X))
repeat max times
    X' := SubconjuntAleatori(X)
    if J(S(X')) ≥ J_o then
        if |X'| < |Best| then
            Best := X'
            L := X'
        end
        else
            if |X'| = |Best| then
                L := append(L, X')
            end
        end
    end
end
```

És molt important remarcar que a causa del fet que utilitzem el LVF amb la inconsistència com a mesura d'avaluació, el cost computacional del LVF serà de $\mathcal{O}(max \cdot |S|)$, on max és el nombre d'iteracions i $|S|$ el nombre d'instàncies del conjunt de dades. Aquest cost és així perquè iterarem max vegades i el procediment que trobem de més cost dins del bucle és el càlcul de la inconsistència el qual s'ha definit anteriorment com $\mathcal{O}(S)$.

Durant totes les millores proposades en aquest capítol el cost computacional de totes les optimitzacions es mantindrà en $\mathcal{O}(max \cdot |S|)$, ja que no afegirem dintre del bucle procediments computacionalment més complexos que el càlcul de la inconsistència, i aquest càlcul de la inconsistència es mantindrà igual.

6.3 Modificacions d'acceptació de successors

En aquesta secció presentem dues modificacions molt similars, les quals són excloents entre elles. Aquestes millores parteixen de la idea d'ajustar dinàmicament la mida màxima dels subconjunts de variables candidats tenint en compte la millor

solució trobada per al moment. Aquesta idea és molt necessària per a millorar el rendiment de l'algorisme, ja que a partir del fet que el LVF trobi una solució de mida l amb una inconsistència inferior al llindar d'acceptació, tot candidat amb un nombre de variables superior a l no podrà optar a ser la solució final. D'aquesta manera, totes les iteracions que computin la inconsistència dels candidats amb un nombre de variables superior a l seran inservibles.

Per a evitar aquest poc aprofitament de les iteracions, proposem que l'algorisme es limiti a cercar solucions d'una mida màxima determinada, cada cop que es redueix el nombre de variables de la millor solució trobada. Per a seleccionar aquest llindar màxim de mida tenim dues opcions:

- Llindar màxim equivalent al nombre de variables de la millor solució actual, l.
- Llindar màxim inferior al nombre de variables de la millor solució actual, $l - 1$.

Aquestes dues propostes corresponent respectivament a les dues primeres modificacions de l'algorisme.

6.3.1 Modificació 1

Aquesta primera modificació en la qual limitem l'acceptació dels subconjunts de variables candidats en aquells que tinguin una mida inferior o igual a la millor resposta actual, no canvia gaire l'estructura de l'algorisme.

Afegirem la nova condició posteriorment de generar aleatòriament el subconjunt de variables candidat. Si el subconjunt candidat té una mida superior al nombre de variables de la millor solució actual, aquest candidat quedarà invalidat i es generarà un de nou. Aquest procés serà repetit fins que es trobi un candidat que tingui un nombre de variables inferior o equivalent a la millor solució actual. A continuació es presenta el pseudocodi de la primera millora.

Algorisme 15: Las Vegas Filter Modificació 1

Entrada:
max - el nombre màxim d'iteracions
$\mathcal{J}$ - la mesura d'avaluació
$S(X)$ - una mostra S descrita pel conjunt de variables X
Sortida:
L - Llista de les solucions equivalents trobades

```
L := []
Best := X
J_o := J(S(X))
repeat max times
    n := |X| + 1
    while n > |Best| do
        X' := SubconjuntAleatori(X)
        n := |X'|
    end
    if J(S(X')) ≤ J_o then
        if |X'| < |Best| then
            Best := X'
            L := X'
        end
        else
            L := append(L, X')
        end
    end
end
```

Podem observar en els resultats de l'experimentació una millora en l'*score* i uns temps d'execució similars respecte a la versió original, això és degut al fet que ja no malgastem iteracions en candidats que no ens milloraran la solució actual. És a dir, aquest estalvi ens permet emprar més iteracions per a explorar un espai de solucions més prometedor. Més concretament, aquesta millora de l'*score* la podem apreciar en els conjunts de dades CDL1, CDE1 i CDE2.

6.3.2 Modificació 2

En aquesta segona modificació només acceptarem com a subconjunts candidats aquells que tinguin una mida inferior a la millor resposta actual. Aquesta condició afecta directament a la solució de l'algorisme, ja que ara només retornarà un subconjunt de dades i no un llistat de subconjunts de dades.

Aquest canvi de solució és causat per l'addició de la nova condició, la qual inhabilita la possibilitat de trobar solucions amb el mateix nombre de variables i amb una inconsistència inferior al llindar determinat. Aquesta equivalència de bondat entre solucions no podrà existir ja que quan és trobi una solució, el nombre de variables de la següent solució trobada haurà de ser estrictament inferior a l'actual, per la condició afegida.

La condició s'ha afegit igual que en la modificació anterior, però de manera més estricta, ja que els conjunts de dades amb el mateix nombre de variables a la millor solució actual, no seran acceptats. A continuació trobem el pseudocodi de l'algorisme.

Algorisme 16: Las Vegas Filter Modificació 2

Entrada:
max - el nombre màxim d'iteracions
$\mathcal{J}$ - la mesura d'avaluació
$S(X)$ - una mostra S descrita pel conjunt de variables X
Sortida:
Best - Millor solució trobada

$Best := X$
$X' := X$
$\mathcal{J}_o := \mathcal{J}(S(X))$
repeat *max* **times**
  while $|X'| \geq |Best|$ **do**
    $X' := SubconjuntAleatori(X)$
  end
  if $\mathcal{J}(S(X')) \geq \mathcal{J}_o$ **then**
    $Best := X'$
    if $|Best| = 1$ **then stop**
  end
end

Podem observar en l'experimentació, que aquesta modificació també millora els resultats de l'*score* respecte a la versió original. Un resultat lògic tenint en compte la seva semblança amb la versió anterior. També podem observar que els seus temps d'execució són molt similars respecte a la versió original de l'algorisme.

6.3.3 Conclusió

Els resultats de les dues versions proposades són molt similars en quant els indicadors *score*, temps d'execució i *accuracy*. Podem fixar-nos però en el fet que l'*score* de la segona modificació és millor en els conjunts de dades CDP, els conjunts de dades amb més variables dels generats. Estudiant aquesta observació podem entendre que la modificació 2 tendirà a tenir un millor rendiment pel que fa a la reducció de variables en conjunts de dades amb moltes variables.

Aquesta hipòtesi és força òbvia, ja que amb la modificació 2 cada cop que trobem una solució correcta, reduïm l'espai de possibles solucions de mida inferior més que amb la modificació 1, i no perdem cap candidat amb aquesta mida inferior. Per tant, aquesta major reducció de l'espai de possibles solucions portarà a una major reducció de variables en conjunts de dades amb moltes variables redundats o irrellevants.

Per altra banda, amb la modificació 2 al només aportar una possible solució podria ser que l'algorisme perdi fiabilitat en termes d'*accuracy*, ja que l'algorisme no avalua aquesta mesura i pot variar la seva bondat respecte a la inconsistència indicada.

Entre aquestes dues modificacions decidim escollir la segona, a causa del fet que prioritzem la reducció de variables i preferim un algorisme amb solució única, ja que així no comporta d'un processat posterior. Si s'obtingués una gran variabilitat en l'*accuracy* dels models predictius generats a partir de les solucions de les nostres modificacions, ens replantejaríem aplicar la modificació 1. Per tant, afegim la modificació 2 a la nostra versió incremental.

6.4 Modificació en la generació dels subconjunts candidats

Aquesta millora parteix des de la modificació 2 i té com a objectiu reduir el temps d'execució de la generació dels candidats amb una mida inferior a la millor solució actual.

La necessitat d'aquesta millora neix de la ineficient imposició de les condicions de mida de les solucions candidates de les modificacions 1 i 2. Aquesta ineficiència és deguda al fet que generem els subconjunts aleatoris de variables sota cap condició i posteriorment comprovem si compleix la condició de la mida. En un hipotètic cas on tenim un conjunt de dades amb un alt nombre de variables i la nostra millor solució ha reduït en gran part aquestes variables, l'algorisme malgastarà temps de còmput buscant un subconjunt de variables amb una mida inferior al millor, ja que la probabilitat de seleccionar poques variables serà baixa (perquè el conjunt de dades té moltes variables).

Existeixen tècniques més eficients per a acomplir aquesta tasca de selecció de subconjunts de variables aleatoris sota la imposició d'una certa mida determinada. Per tant, en aquesta modificació aplicarem un procediment més eficient.

6.4.1 Modificació 3

La tècnica que emprarem per a la generació dels subconjunts de variables candidats és la següent; primer amb un generador de nombres aleatoris, on tots els nombres tenen la mateixa probabilitat de ser escollits, generem la mida que tindrà el nostre subconjunt de variables. Aquesta generació es veu fitada entre $[1, |Best|)$ on $|Best|$ és el nombre de variables de la millor solució trobada fins al moment. Posteriorment, amb la mida ja definida del subconjunt de variables candidat, se seleccionaran aleatòriament n variables de totes les variables que formen el conjunt de variables original, on n és el nombre de variables aconseguit de la generació aleatòria anterior. En la selecció aleatòria de variables, també totes les variables tenen la mateixa probabilitat de ser seleccionades.

Per tant, realitzant aquest procés sempre generarem subconjunts de variables amb una mida inferior a la millor actual i no caldrà repetir aquestes generacions aleatòries com en les modificacions anteriors. Seguidament trobem el pseudocodi de la millora.

Algorisme 17: Las Vegas Filter Modificació 3

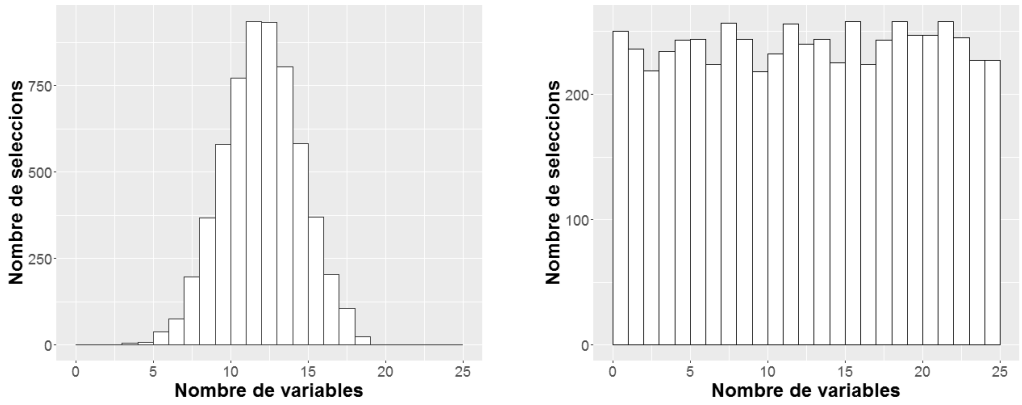
Entrada:
max - el nombre màxim d'iteracions
$\mathcal{J}$ - la mesura d'avaluació
$S(X)$ - una mostra S descrita pel conjunt de variables X
Sortida:
Best - Millor solució trobada

$Best := X$
$\mathcal{J}_o := \mathcal{J}(S(X))$
repeat *max* **times**
  $n := GeneradorAleatoriFitat(1, |Best|)$
  $X' := SubconjuntAleatori(X, n)$
  if $\mathcal{J}(S(X')) \geq \mathcal{J}_o$ **then**
    $Best := X'$
    if $|Best| = 1$ **then stop**
  end
end

Aquesta versió presenta resultats molt similars a la seva predecessora, la modificació 2. Però, sorprenentment pel que fa a l'indicador *score*pels conjunts de dades CDP, aquesta versió millora amb claredat les versions anteriors. Un resultat que a priori pot resultar estrany, però que té la seva respectiva explicació.

Al canviar la generació dels subconjunts de variables, s'ha canviat també la distribució de probabilitats pel que fa a la mida del subconjunt de variables. És a dir, la modificació 1 i 2, generaven els subconjunts candidats com la versió original de l'algorisme, la qual dicta que tota variable té la mateixa probabilitat a aparèixer. Per tant, podem comprendre aquesta distribució de probabilitats com una distribució binomial, on la probabilitat d'èxit p és 0.5 i el nombre de repeticions independents n és el nombre de variables. Per tant cada repetició indica si la respectiva variable s'inclou en el subconjunt de variables candidat.

En canvi, a la nova versió aquesta distribució de probabilitats s'ha vist alterada, ja que ara totes les mides dels subconjunts de variables tenen la mateixa probabilitat a ser seleccionats, ja que primer es selecciona la mida i després les variables que formen el subconjunt. Hem executat els dos generadors 3000 vegades amb un conjunt de 25 variables per observar les seves respectives diferències pel que fa a la distribució de probabilitat de la mida de la solució. A continuació s'exposen els resultats.



(A) Versió original (B) Modificació 3

FIGURE 6.1: Distribució de les mides dels subconjunts candidats

Com podem observar en el generador de la versió original, la mida segueix una distribució binomial amb una p de 0.5, el que fa que la mida més obtinguda com a resultat oscil·li entre la meitat del nombre de variables totals i a partir d'aquest valor decreixi simètricament. Aquest fet és el que provoca que en conjunts de dades on trobem moltes variables irrellevants i redundants l'algorisme les redueixi lentament, ja que sempre optarà a mides més pròximes a la meitat de la mida total i no a inferiors. Encara que la modificació 1 i 2 limitin la cerca a solucions de mides inferiors a l'actual, aquesta distribució també afectarà en la variabilitat de les mides de les solucions, ja que en solucions d'un terç de la mida total o inferiors tendiran a seleccionar la mida màxima del llindar sempre.

En canvi, escollint abans la mida del subconjunt de variables com en el cas de la modifiacació 3, tindrem una probabilitat que tendirà a ser equivalent per totes les mides possibles dels subconjunts de variables candidats els quals es troben en l'interval $[1, |Best|)$, donant així una major variabilitat pel que fa a la mida de les solucions proposades.

Aquest fet ha estat el culpable d'aquesta millora de rendiment en conjunts de dades amb moltes variables no rellevants, de la modificació 3, envers les altres versions exposades. Pel que fa als temps d'execució, no s'aprecia gran diferència perquè el benefici de la nova generació de subconjunts de variables és ínfim respecte el càlcul de la inconsistència dels subconjunts de dades.

6.5 Modificacions en la distribució de probabilitat de la mida dels subconjunts candidats

A causa de l'evidència trobada en l'apartat anterior de la dependència entre la distribució emprada pel generador de mides i el rendiment de l'algorisme, decidim incidir més profundament en la utilització de diferents distribucions de probabilitats.

A l'apartat anterior hem observat que el rendiment de l'algorisme amb un generador basat en una distribució de probabilitats equitativa era superior al d'un basat en una distribució binomial amb $p = 0.5$. Però això no significa que sigui la distribució òptima, de fet la distribució de probabilitats equitativa pateix d'un defecte

que la distribució binomial no patia; si totes les mides tenen la mateixa probabilitat, hi haurà moltes execucions que utilitzin un nombre de variables poc realista, és a dir, utilitzaran molt poques variables, fet que rebaixa la probabilitat de què el candidat sigui correcte, per tant, augmenta la variabilitat de l'algorisme i baixa la seva fiabilitat.

La solució proposada per a trobar un equilibri entre la gran variabilitat que ens dóna l'ús d'una distribució equitativa i l'estancament de mida que ens provoca la distribució original de l'algorisme, és utilitzar un reajustament dinàmic de la distribució de probabilitats que utilitza el generador de mides dels candidats segons el nombre de variables de la millor solució actual. És a dir, cada cop que trobem una solució correcta amb una mida inferior, utilitzarem una distribució de probabilitats per la selecció del nombre de variables dels candidats amb uns paràmetres diferents.

6.5.1 Modificació 4

Per aquesta primera versió on utilitzarem un reajustament dinàmic de distribucions farem servir un generador aleatori basat en una distribució Gausiana per a observar el rendiment de l'algorisme. Sabem que no és la distribució més apropiada a causa del fet que és una distribució continua i l'ideal seria utilitzar una distribució discreta. Però, gràcies als seus dos paràmetres d'entrada (mitjana μ i desviació tipus σ) i realitzant un arrodoniment a la sortida de la distribució al nombre natural més pròxim, podrem observar el seu rendiment amb moltes distribucions normals diferents.

La idea d'emprar una distribució normal prové d'ajustar la mitjana μ de la distribució en la mida més prometedora de la cerca i utilitzar un cert nivell de desviació tipus σ per ampliar l'espai de cerca. Per tant aquests dos paràmetres estan condicionants amb el nombre de variables de la millor solució actual, i d'aquesta manera gradualment s'aniran adaptant a les necessitats de la cerca. En aquesta modificació són definits així, on $|Best|$ és la mida de la millor solució actual:

- Mitjana $\mu = \frac{|Best|}{d_\mu}$, on d_μ és un paràmetre d'entrada que hem d'adaptar. Sabem que $d_\mu \geq 1$, ja que hem de trobar solucions amb una mida inferior a $|Best|$. Com més gran sigui aquest valor, la mitjana de la distribució serà més petita.

- Desviació tipus $\sigma = \frac{|Best|}{d_\sigma}$, on d_σ és un paràmetre d'entrada a adaptar. Com més gran sigui aquest valor, menys desviació tipus trobarem a la distribució.

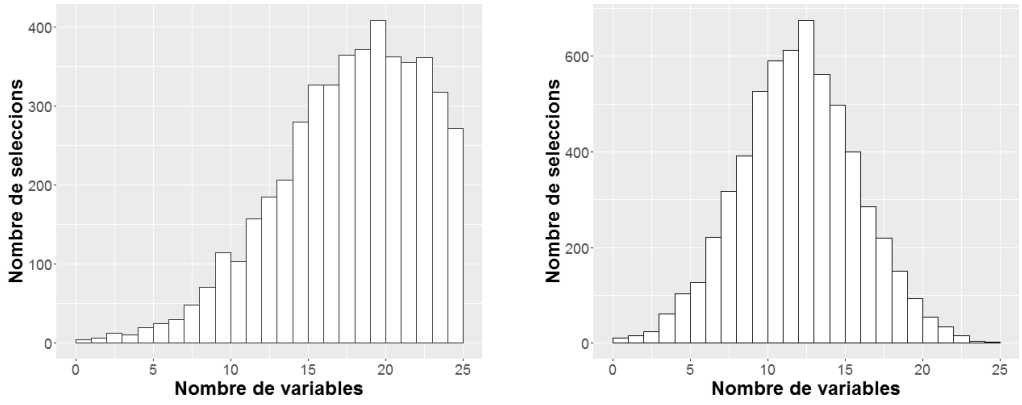
A continuació, s'exposa el pseudocodi de la modificació.

Algorisme 18: Las Vegas Filter Modificació 4

Entrada:
max - el nombre màxim d'iteracions
$\mathcal{J}$ - la mesura d'avaluació
$S(X)$ - una mostra S descrita pel conjunt de variables X
d_μ - Divisor de la mitjana del generador de nombre de variables
d_σ - Divisor de la desviació tipus del generador de nombre de variables
Sortida:
$Best$ - Millor solució trobada

$Best := X$
$\mathcal{J}_o := \mathcal{J}(S(X))$
repeat max **times**
  $n := GeneradorAleatoriNormalitat(|Best|/d_\mu, |Best|/d_\sigma)$
  $X' := SubconjuntAleatori(X, n)$
  if $\mathcal{J}(S(X')) \geq \mathcal{J}_o$ **then**
    $Best := X'$
  end
end

Es van realitzar diferents experimentacions per a la selecció de paràmetres i es recomana utilitzar un d_μ de 1.25, per tant un 80% del valor de $|Best|$ com a mitjana i un d_σ de 4. A continuació, es mostren les distribucions de les mides dels candidats obtinguts de 6000 execucions del generador aleatori amb aquests paràmetres.



(A) $|Best| = 25$ (B) $|Best| = 15$

FIGURE 6.2: Distribució de les mides dels subconjunts candidats, Modificació 4

Podem observar en les gràfiques de l'experimentació com el generador aleatori adapta la generació dinàmicament amb el valor de $|Best|$, la mida de la millor solució actual. La mediana recau en una mida poc inferior al millor actual, aquest fet comportarà una reducció de variables gradual i amb menys variabilitat, per tant l'algorisme guanyarà fiabilitat.

Pel que fa a els resultats, hem obtingut una millora de l'*score* envers les anteriors versions, la qual la podem apreciar en la majoria d'experiments però més notablement amb els conjunts de dades CDL1, CDE1 i CDP1. També cal notar que s'ha reduït en molts casos la variabilitat de la nostra solució pel que fa l'*score*. Els temps d'execució s'han elevat respecte a les altres versions, això és degut principalment

perquè amb aquesta versió no executem iteracions amb molt poques variables, sinó que aprofitem totes les execucions en testar candidats més reals, però com normalment tenen més nombre de variables, el càlcul de la inconsistència és més lent.

En les següents dues modificacions, aplicarem distribucions discretes en el generador de nombres aleatoris, ja que s'adapten millor a la casuística del problema de generar la mida dels subconjunts de dades. Amb el coneixement que hem obtingut amb aquesta versió podrem definir fàcilment els paràmetres de les noves distribucions de probabilitat.

6.5.2 Modificació 5

En aquesta modificació utilitzarem un generador de nombres aleatoris basat en una distribució de Poisson en comptes de la distribució Normal. Recordem que la distribució de Poisson segueix la següent funció de probabilitat:

$$P(X = k) = e^{-\lambda}\frac{\lambda^k}{k!} \text{ , on } k = 0,1,2,...$$

Com en la modificació anterior, haurem de definir correctament el paràmetre d'entrada del nostre generador de nombres aleatoris basats en una distribució Poisson. Aquest paràmetre és el paràmetre λ de la distribució de Poisson, el qual és entès com la freqüència esperada del fenomen modelat per la distribució. Per tant, ajustarem aquest paràmetre d'acord amb les nostres necessitats. Utilitzarem la mateixa tècnica emprada en la versió anterior, definirem un paràmetre d'entrada d_λ el qual divideix la mida de la millor solució actual i el resultat és el valor de λ que s'utilitza en el generador.

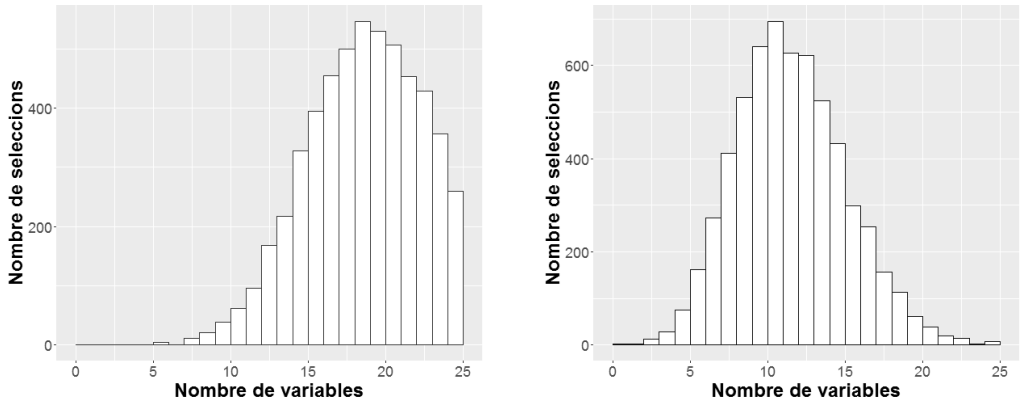
Vam observar a la modificació anterior, que seleccionar com a mediana una reducció del 20% respecte a la mida de la millor solució actual, ens oferia els millors resultats. Reutilitzarem aquest coneixement, i recomanem definir el paràmetre d_λ amb el valor de 1.25. A continuació s'exposa el pseudocodi de la millora.

Algorisme 19: Las Vegas Filter Modificació 5

Entrada:
max - el nombre màxim d'iteracions
$\mathcal{J}$ - la mesura d'avaluació
$S(X)$ - una mostra S descrita pel conjunt de variables X
d_λ - Divisor del paràmetre lambda del generador de nombres de variables
Sortida:
Best - Millor solució trobada

$Best := X$
$\mathcal{J}_o := \mathcal{J}(S(X))$
repeat *max* **times**
  $n := GeneradorAleatoriPoisson(|Best|/d_\lambda)$
  $X' := SubconjuntAleatori(X, n)$
  if $\mathcal{J}(S(X')) \geq \mathcal{J}_o$ **then**
    $Best := X'$
  end
end

Gràcies al fet que ara utilitzem una distribució de probabilitat discreta en el generador de nombres aleatoris, ens estalviem l'arrodoniment del resultat. Com en la modificació anterior, hem executat 6000 vegades el generador d'aquesta versió amb un conjunt de 25 variables i hem capturat els resultats, s'exposen seguidament.



(A) $|Best| = 25$ (B) $|Best| = 15$

FIGURE 6.3: Distribució de les mides dels subconjunts candidats, Modificació 5

Observem que les distribucions resultants són molt similars a la generació basada en una distribució Normal arrodonida. Si les estudiem amb detall, observem que tenen una variància diferent de les generades amb la versió anterior. Aquest fet s'observa clarament en les execucions amb el paràmetre $|Best| = 25$, això és a causa del fet que com la variància en la modificació 4 està condicionada amb el valor de $|Best|$, aquesta té un valor més alt.

Amb aquesta modificació hem obtingut uns resultats molt similars a la modificació 4, aquest fet és degut a la similitud que hem pogut comprovar que existeix entre els dos mètodes per a generar la mida dels subconjunts de variables candidats.

6.5.3 Modificació 6

Aquesta serà l'última modificació en la qual testejarem una nova distribució per a la generació de la mida dels subconjunts de variables candidats. En aquesta versió utilitzarem un generador de nombres aleatoris basat en una distribució binomial. Cal remarcar, que la funció de probabilitat que segueix una distribució binomial és la següent:

$$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k}, \text{ on } \binom{n}{k} = \frac{n!}{k!(n-k)!}$$

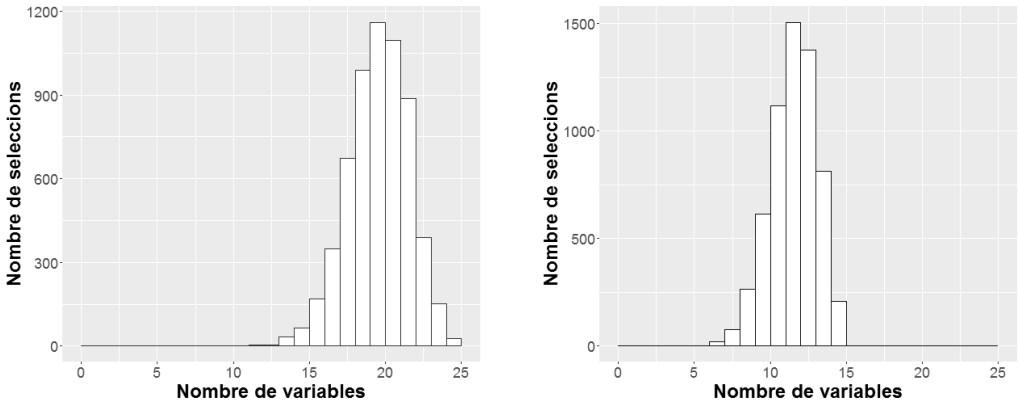
En aquesta millora, definirem com a nombre d'assaigs n la mida de la millor solució obtinguda fins al moment ($|Best|$), i la probabilitat d'èxit de cada assaig p vindrà definida per un paràmetre d'entrada el qual anomenarem també p. Perquè l'algorisme tingui un rendiment igual de beneficiós al de les anteriors dues modificacions, recomanem utilitzar el paràmetre p amb un valor de 0.8. Ja que això provocarà que la mediana de les mides dels subconjunts candidats sigui un 20% menor al de la mida de la millor solució trobada. A continuació, mostrem el pseudocodi de la versió.

Algorisme 20: Las Vegas Filter Modificació 6

Entrada:
max - el nombre màxim d'iteracions
$\mathcal{J}$ - la mesura d'avaluació
$S(X)$ - una mostra S descrita pel conjunt de variables X
p -Paràmetre probabilitat d'èxit del generador de mides
Sortida:
Best - Millor solució trobada

$Best := X$
$\mathcal{J}_o := \mathcal{J}(S(X))$
repeat *max* **times**
  $n := GeneradorAleatoriBinomial(|Best|, p)$
  $X' := SubconjuntAleatori(X, n)$
  if $\mathcal{J}(S(X')) \geq \mathcal{J}_o$ **then**
    $Best := X'$
  end
end

Com s'ha realitzat en les dues versions anteriors, s'ha executat 6000 vegades el generador de mides d'aquesta modificació amb un conjunt de 25 variables. Seguidament, trobem els resultats.



(A) $|Best| = 25$ (B) $|Best| = 15$

FIGURE 6.4: Distribució de les mides dels subconjunts candidats, Modificació 6

Trobem severes diferències pel que fa a aquests resultats envers les altres distribucions exposades anteriorment. Aquesta versió limita la mida màxima dels subconjunts de variables a la mida de la millor solució actual trobada, una característica necessària, la qual s'ha estudiat en la secció 3, *Modificacions d'acceptació de successors*, d'aquest capítol. Amb les altres distribucions aquesta característica no es compleix. També, en els resultats d'aquesta versió notem una menor variància en les dues gràfiques, ja que trobem menys resultats emprats però amb més seleccions. Aquest fet pot ajudar a l'algorisme a guanyar fiabilitat, ja que l'ajust de l'espai de cerca és més sever.

Quant als resultats de l'experimentació amb els conjunts de dades de la fase inicial, trobem millors resultats per l'indicador *score* respecte a les dues anteriors

versions. En el CDL1, CDE1 i CDE2 s'observa clarament la diferència. El temps d'execució a penes es modifica respecte a les dues modificacions anteriors.

6.5.4 Conclusió

S'ha pogut observar amb les experimentacions realitzades, que la idea d'aplicar un reajustament dinàmic de la distribució de probabilitats que utilitza el generador de mides dels candidats segons la mida de la millor solució actual, aporta una millora en l'indicador *score*, respecte a les anteriors versions. Això significa que amb aquest reajustament dinàmic obtenim millors reduccions de variables, per tant, solucions amb menys variables irrellevants i redundants.

Tot i reduir un nombre més elevat de variables, l'*accuracy* no redueix el seu valor, de fet, en alguns casos augmenta molt lleugerament respecte a les versions anteriors. Els temps d'execució s'han elevat respecte a les versions anteriors a causa de no executar iteracions amb molt poques variables, sinó que només testar subconjunts amb mides més prometedores, els quals en tenir un nombre de variables més elevat, el temps de càlcul de la seva inconsistència és major.

En obtenir millors resultats amb la modificació 6 i considerar els beneficis explicats que ens aporta aquesta distribució respecte a les altres dues, decidim seleccionar la distribució binomial, per al reajustament dinàmic de la distribució de probabilitats del generador de mides dels candidats. Per tant, seleccionarem la modificació esmentada i seguirem treballant amb aquesta versió.

6.6 Modificacions en les probabilitats de selecció de les variables

Ja hem estudiat les diferents distribucions de probabilitats per a la generació de les mides dels subconjunts candidats, però encara no hem estudiat la probabilitat de selecció de les diferents variables que es troben contingudes en el conjunt de variables a reduir. A causa del fet que no tenim cap coneixement previ de les variables, haurem d'aplicar el coneixement que guanyi l'algorisme sobre les variables a mesura que vagi iterant amb elles.

Aquest coneixement que pot adquirir l'algorisme sobre les diferents variables, és el grau d'inconsistència que s'obté quan formen part del subconjunt de variables candidat. A partir d'aquesta idea desenvoluparem les següents modificacions.

6.6.1 Modificació 7

En aquesta versió canviarem la probabilitat de selecció de les variables que formen els subconjunts candidats. En les versions anteriors les probabilitats de totes les variables eren equivalents, per tant no hi havia variables que destaquessin. Amb aquesta modificació, iniciarem totes les probabilitats de selecció de les variables amb el mateix valor, i conforme els seus resultats en base el grau d'inconsistència, incrementarem o disminuïm la probabilitat de selecció de les diferents variables.

Més concretament, inicialitzarem totes les variables amb una probabilitat de 0.5 (per a facilitar l'aplicabilitat, treballarem amb probabilitats no normalitzades), i les

variables que formin part del subconjunt candidat de cada iteració seran disminuïdes un 0.05 si el subconjunt que formen té una inconsistència superior al llindar d'acceptació. En canvi, si tenen una inconsistència inferior o igual al llindar d'acceptació, les seves probabilitats augmentaran un 0.05.

Definim un llindar mínim per a les probabilitats de 0.05, per tant, mai tindrem probabilitats inferiors a 0.05 en cap variable. A continuació és mostra el pseudocodi de la modificació.

Algorisme 21: Las Vegas Filter Modificació 7

Entrada:
max - el nombre màxim d'iteracions
$\mathcal{J}$ - la mesura d'avaluació
$S(X)$ - una mostra S descrita pel conjunt de variables X
p - Paràmetre indicador de la probabilitat d'èxit del generador de mides
Sortida:
$Best$ - Millor solució trobada

$Best := X$
$\mathcal{J}_o := \mathcal{J}(S(X))$
$prob := init(0.5, |X|)$ `// Inicialització de les probabilitats a 0.5`
repeat max **times**
  $n := GeneradorAleatoriBinomial(|Best|, p)$
  $X' := SubconjuntAleatori(X, n, prob)$
  if $\mathcal{J}(S(X')) \geq \mathcal{J}_o$ **then**
    $Best := X'$
    for each $x' \in X'$ **do**
      $prob[x'] := prob[x'] + 0.05$
    end
  end
  else
    for each $x' \in X'$ **do**
      if $prob[x'] > 0.05$ **then**
        $prob[x'] := prob[x'] - 0.05$
      end
    end
  end
end

Amb aquest canvi en les probabilitats de selecció de les variables que formen els subconjunts candidats, intentem guiar l'algorisme a espais de solucions més prometedors. D'aquesta manera les variables que tendeixin a tenir millors resultats seran seleccionades més vegades per a formar els subconjunts candidats.

Cal destacar, que gràcies a anar dinàmicament reajustant la mida màxima dels subconjunts de variables candidats, un mateix subconjunt de variables no serà repetit molts cops, ja que:

- Si el subconjunt té una inconsistència superior al llindar màxim, les probabilitats de les variables baixaran, per tant la probabilitat d'obtenir aquest subconjunt de variables s'anirà reduint.
- Altrament, si el subconjunt té una inconsistència inferior o igual al llindar màxim, la probabilitat d'obtenir un mateix subconjunt de variables amb la

mateixa mida és molt baixa, a causa del reajustament dinàmic de la mida dels subconjunts candidats.

Els resultats de l'experimentació mostren una lleugera millora en l'indicador *score* respecte a les versions anteriors, apreciable en les experimentacions amb CDL2, CDE2 i CDP3. Pel que fa als temps d'execució, es mantenen similars als de l'anterior versió 6. En conseqüència, aquesta millora és força positiva, gràcies al fet que els subconjunts candidats tenen mides similars als de l'anterior versió (per tant, el temps de càlcul del nivell d'inconsistència es mantenen), però les variables que els formen tendeixen a tenir menys inconsistència respecte a la variable *target*.

6.6.2 Modificació 8

Aquesta última modificació de la fase inicial, és igual que l'anterior, però hi afegim una fita superior en les probabilitats de les variables amb un valor de 0.95.

La idea d'aquesta fita, és tenir un control major sobre les probabilitats i d'aquesta manera tenir definit el seu rang de valors entre [0.05, 0.95]. És una simple modificació, però és necessària perquè a priori no sabem si amb aquesta fita millorarà el rendiment de l'algorisme. A continuació exposem el pseudocodi de la modificació.

Algorisme 22: Las Vegas Filter Modificació 8

Entrada:
max - el nombre màxim d'iteracions
$\mathcal{J}$ - la mesura d'avaluació
$S(X)$ - una mostra S descrita pel conjunt de variables X
p - Paràmetre indicador de la probabilitat d'èxit del generador de mides
Sortida:
Best - Millor solució trobada

$Best := X$
$\mathcal{J}_o := \mathcal{J}(S(X))$
$prob := init(0.5, |X|)$ `// Inicialització de les probabilitats a 0.5`
repeat *max* **times**
  $n := GeneradorAleatoriBinomial(|Best|, p)$
  $X' := SubconjuntAleatori(X, n, prob)$
  if $\mathcal{J}(S(X')) \geq \mathcal{J}_o$ **then**
    $Best := X'$
    for each $x' \in X'$ **do**
      if $prob[x'] < 0.95$ **then**
        $prob[x'] := prob[x'] + 0.05$
      end
    end
  end
  else
    for each $x' \in X'$ **do**
      if $prob[x'] > 0.05$ **then**
        $prob[x'] := prob[x'] - 0.05$
      end
    end
  end
end

Quant als resultats d'aquesta versió en l'experimentació, s'han obtingut resultats molt similars a l'anterior versió, tant en l'indicador *score*, l'*accuracy* i el temps d'execució. Per tant, aquesta fita superior gairebé no ha afectat al rendiment de l'algorisme.

Encara que aquesta fita superior no afecti el rendiment de l'algorisme, es decideix seleccionar aquesta última versió com la versió final de la fase inicial, això és degut al fet que amb aquesta fita tenim un major control sobre el valor de les probabilitats.

6.7 Conclusió

Durant el transcurs d'aquestes modificacions de l'algorisme LVF s'ha intentat rebaixar el grau d'aleatorietat d'aquest algorisme i dotar-lo d'un major encert en l'espai de solucions a buscar. Aquesta reducció d'aleatorietat, tenia l'objectiu d'augmentar la fiabilitat de l'algorisme i la seva capacitat de reduir subconjunts de variables, mantenint el llindar de la inconsistència.

Podem dir que s'ha assolit aquest objectiu per als conjunts de dades amb els quals s'ha realitzat l'experimentació, ja que la versió final d'aquesta sèrie de modificacions incrementals, la modificació 8, ha obtingut uns resultats millors en tots els conjunts de dades testats, pel que fa l'*score*. Per tant, té una capacitat de reducció més alta que la versió original, també s'ha reduït la variabilitat dels nostres resultats, tal com és mostra en els resultats de la nostra experimentació, indicador de què l'algorisme ha obtingut un guany en la seva fiabilitat. Aquest fet és molt positiu, ja que hem aconseguit preservar el cost computacional en $\mathcal{O}(max \cdot |S|)$ com l'algorisme original i a la vegada hem millorat el seu rendiment.

Pel que fa al temps d'execució, observem que ha augmentat respecte a la versió original del LVF, el causant d'aquest increment de temps d'execució és que en aquesta nova versió es seleccionen subconjunts de variables d'acord amb la millor mida trobada i no s'utilitzaran iteracions per a comprovar subconjunts de variables amb mides ínfimes. Trobem indicis d'aquest fet en l'experimentació, ja que aquesta diferència en el temps d'execució s'observa en els conjunts de dades CDP, els quals tenen un nombre d'instàncies major, és a dir, un temps de càlcul d'inconsistència major i conseqüentment la diferència entre calcular la inconsistència amb poques variables envers amb moltes variables incrementa.

Aquest temps d'execució addicional, és un preu a pagar el qual no podem evitar si volem que el nostre algorisme cerqui en els espais de solucions més prometedors. Aquest increment de temps, no afegeix un cost addicional al cost computacional de l'algorisme i per tant podem permetre'ns aquest augment de temps amb l'objectiu de què l'algorisme tingui un millor rendiment.

Amb la versió que exposem com a final d'aquestes millores inicials, s'han aconseguit també uns lleugers millors resultats pel que fa l'*accuracy* envers la versió original. Els resultats de la nostra versió milloren també l'*accuracy* aconseguida amb totes les variables, això és degut al fet que la reducció de variables provoca que s'elimini el soroll afegit en els conjunts de dades i permeti al Naive Bayes realitzar millors prediccions. És curiós que en l'experimentació que utilitzem els conjunts de dades CDE1 i CDE3, l'*accuracy* del Naive Bayes augmenti tant respecte a la versió

amb totes les variables, ja que als classificadors de la família Naive Bayes són difícils d'enganyar amb el soroll.

Gràcies a la classificació de les variables en rellevants, irrellevants i redundants, s'ha pogut observar també que el LVF i les optimitzacions realitzades amb ell acostumen a eliminar un major nombre de variables redundants que de variables irrellevants.

Les variables irrellevants per si soles són molt inconsistents envers la variable objectiu, però quan es troben dins d'un subconjunt de variables només poden rebaixar o mantenir igual la inconsistència, a causa de la monotonia que mostra la mesura d'avaluació. Per contra, les variables redundants, per culpa d'aportar la mateixa informació que alguna de les variables del subconjunt de dades només optaran a mantenir el mateix nivell d'inconsistència.

S'ha obtingut també que les variables rellevants són les menys eliminades dels subconjunts de variables, aquest fet és conseqüència de què són les variables que rebaixen en major manera la inconsistència del conjunt de dades. En la secció *Resultats per tipologia de variables* del capítol 13, *Resultats de l'experimentació*, mostrem una petita part dels resultats obtinguts per tipologia de variables per a exposar aquest fet.

Capítol 7

Millores finals del LVF

En aquest capítol proposarem l'optimització definitiva basada en les millores incrementals estudiades en el capítol 6, *Millores inicials del LVF*. Posteriorment estudiarem algunes optimitzacions ja proposades per altres autors, aplicarem en elles la nostra optimització definitiva i analitzarem els seus rendiments.

També estudiarem la metodologia híbrida i d'embolcall en el LVF proposant diverses optimitzacions per a cadascuna. Cal remarcar, que els conjunts de dades emprats amb aquestes optimitzacions són els que anomenem com conjunts de dades finals, explicats en el capítol 4, *Conjunts de dades*.

És molt important remarcar que els resultats de les experimentacions amb optimitzacions basades en mètodes de filtre es troben en la secció *Resultats de les millores finals del LVF basades en mètodes de filtre* i els resultats de les optimitzacions basades en mètodes híbrids i d'embolcall es troben en la secció *Resultats de les millores finals del LVF basades en mètodes híbrids i d'embolcall*, les dues seccions es troben al capítol 13, *Resultats experimentació*.

A causa de la gran extensió de la memòria s'ha preferit mostrar la majoria de resultats en el capítol esmentat i no repetir-ne l'aparició en el capítol actual. Totes les gràfiques consten d'una breu descripció amb l'identificador del conjunt de dades al qual fan referència i el mesurament que prenen.

7.1 Las Vegas Adaptative

Las Vegas Adaptative (LVA) és la versió definitiva que proposem com a la selecció de les millors modificacions incrementals estudiades en el capítol 6, *Millores inicials del LVF*. El terme *adaptative*, prové de la capacitat de l'algorisme per adaptar la seva cerca probabilística en espais de solucions prometedors i no tenir un comportament totalment aleatori com el té la versió original del LVF.

Aquesta versió final parteix de la Modificació 8 del capítol anterior, la qual prèviament ja s'ha explicat i argumentat. Realitzant un estudi més exhaustiu d'aquesta Modificació 8, es va observar que totes les probabilitats de les variables tendien a finalitzar amb valors molt baixos, fet que impedia extreure el màxim rendiment de l'ús de probabilitats individuals de les variables, ja que les diferències entres les probabilitats eren ínfimes i par tant totes presentaven probabilitats similars.

En l'estudi d'aquesta casuística, es va arribar a la conclusió de què no podíem augmentar les probabilitats amb el mateix valor amb el qual les disminuíem, ja que

l'algorisme tendia a obtenir més subconjunts de variables amb una inconsistència major a l'acceptada. En conseqüència, les probabilitats de totes les variables disminuïen gradualment.

Aquesta conclusió esmentada considerem que es pot veure afectada pel conjunt de dades, ja que si es realitzés l'experimentació amb conjunts de dades amb moltes més variables rellevants que variables irrellevants trobaríem que l'algorisme tendiria a obtenir més solucions correctes que incorrectes. Per tant, decidim afegir dos nous paràmetres a l'algorisme:

- *alpha*(α): Paràmetre regulador de l'increment de les probabilitats quan el subconjunt de dades candidat té una inconsistència inferior o igual al llindar màxim.
- *beta*(β): Paràmetre regulador del decrement de les probabilitats quan el subconjunt de dades candidat té una inconsistència superior al llindar màxim.

A part de la conclusió esmentada, també podem deduir a causa de la monotonia que presenta la inconsistència, que a mesura que avanci l'execució de l'algorisme, més resultats amb una inconsistència inferior al llindar obtindrem, en conseqüència d'escollir subconjunts de variables amb una mida cada cop més petita.

Per a regular aquest fet s'ha decidit que el valor de l'increment o decrement de probabilitat de les variables canviï de manera dinàmica en l'execució de l'algorisme. Aquesta idea l'aconseguim portar a terme basant el valor de l'addició o subtracció de probabilitat a partir de la mida del subconjunt de variables avaluat, augmentant l'increment i disminuint el decrement de probabilitats quan el subconjunt de variables és més petit.

Definim $inc : \mathcal{P}(X) \rightarrow [0, 0.98]$ com l'increment de probabilitats del subconjunt de variables X i definim $dec : \mathcal{P}(X) \rightarrow [-0.98, 0]$ com el decrement de probabilitats del subconjunt de variables X. Definim també $X^{'}$ i $X^{''}$ com dos conjunts de dades diferents. Per assolir la idea mencionada en l'anterior paràgraf, s'ha d'assolir:

- Quan $|X^{'}| > |X^{''}|$, llavors $inc(X^{'}) < inc(X^{''})$ i $dec(X^{'}) > dec(X^{''})$.
- Altrament, quan $|X^{'}| = |X^{''}|$, llavors $inc(X^{'}) = inc(X^{''})$ i $dec(X^{'}) = dec(X^{''})$.

D'aquesta manera quan la millora es trobi avançada en l'execució incrementarà més la probabilitat de les variables que conformen els conjunts de variables solucions, ja que aquests subconjunts de variables tindran una mida inferior a la inicial i per tant una probabilitat inferior d'assolir el nivell d'inconsistència. Definirem el valor de l'increment i el decrement de tal manera:

$$inc(X') = \alpha \times \frac{(|X| - |X'|)}{|X|} \qquad dec(X') = -beta \times \frac{(|X'|)^2}{(|X|)^2}$$

On X és el conjunt de variables total d'on s'extreu el subconjunt de variables X'. Podem apreciar també que s'han introduït els paràmetres reguladors α i β els quals s'han explicat anteriorment.

L'increment *inc* definit compleix els requeriments anteriorment comentats perfectament i és senzill de calcular. El paràmetre α permetrà a l'usuari augmentar o

disminuir l'increment de probabilitat. En canvi, el paràmetre β permetrà disminuir o incrementar el decrement de probabilitat.

Inicialment el decrement *dec* va ser definit com $(dec(X') = beta \times (|X'|/|X|))$, una formulació la qual també compleix els requisits descrits i és més similar a la de l'increment *inc*, però en experimentar amb ella, vam decidir canviar-la per a l'actual, ja que necessitàvem una diferència major entre els subconjunts amb diferents nombres de variables. Elevant al quadrat la proposta inicial obtenim una funció la qual disminueix molt poc la probabilitat de les variables dels subconjunts de variables amb molt poques variables, propietat positiva pel fet que normalment serà molt difícil trobar subconjunts de variables molt petits que compleixin la inconsistència imposada.

En aquesta versió també canviarem les fites imposades en l'apartat anterior Modificació 8, perquè ja no tractem amb addicions i subtraccions de probabilitat constants i per tant, podem definir amb més llibertat aquestes fites. La fita inferior de la probabilitat per variable serà de 0.01 i la fita superior de 0.99, és a dir un rang de [0.01, 0.99]. D'aquí prové la fita definida en (inc) i *dec* de 0.98 i -0.98, respectivament, ja que són els casos màxims d'addició i subtracció de probabilitats que accepta el nostre algorisme. Per a simplificar la definició de l'algorisme LVA, separarem la definició de les funcions increment *inc* i decrement *dec* de l'algorisme. Les trobem a continuació:

Algorisme 23: Increment

Entrada:
X' - Subconjunt de variables a augmentar la probabilitat de selecció
prob - Vector de les probabilitats de selecció de variables
n - Mida del conjunt de variables original
α - Paràmetre regulador de l'addició de probabilitat de variables
Sortida:
prob - Vector de les probabilitats de selecció de variables actualitzat

```
for each x' ∈ X' do
    prob[x'] := prob[x'] + (α × ((n − |X'|)/n))
    if prob[x'] > 0.99 then
        prob[x'] := 0.99
    end
end
```

Algorisme 24: Decrement

Entrada:
X' - Subconjunt de variables a augmentar la probabilitat de selecció
prob - Vector de les probabilitats de selecció de variables
n - Mida del conjunt de variables original
β - Paràmetre regulador de la subtracció de probabilitat de variables
Sortida:
prob - Vector de les probabilitats de selecció de variables actualitzat

```
for each x' ∈ X' do
    prob[x'] := prob[x'] − (β × ((|X'|)²/(n)²))
    if prob[x'] < 0.01 then
        prob[x'] := 0.01
    end
end
```

Com s'ha comentat anteriorment, el valor òptim dels paràmetres α i β és condicionat per al conjunt de dades on s'emprarà l'algorisme, ja que les característiques de les seves variables definiran el seu comportament. Tot i això, hem decidit experimentar amb diferents valors per a aquests paràmetres dins del rang [0.1, 1.0]. S'han realitzat 20 execucions per a cada combinatòria entre α i β en els conjunts de dades inicials que obtenien pitjors resultats en les modificacions anteriors, un per a cada bloc. Per acabar, s'han expressat els resultats de la mitjana aritmètica de l'*score* obtingut en diferents matrius on trobem 4 graus de color que simbolitzen la bondat del resultat. El resultat el trobem en la secció *Resultats experimentació amb LVA i alpha/beta*, del capítol 13, *Resultat experimentació*.

Observem en aquests resultats que l'*score* mitjà obtingut amb valors alts de α i valors baixos de β són els que tenen una major robustesa, ja que solen obtenir millors *scores*. Aquesta particularitat és molt observable en els resultats del conjunt de dades CDE2. Decidim utilitzar com a paràmetre α el valor 0.8 i com a paràmetre β el valor 0.2, ja que són els que presenten millors resultats, i conten de robustesa, ja que els seus *"veïns"* en la matriu també tenen resultats bons. Per tant, d'ara endavant utilitzarem aquests valors per als paràmetres descrits en totes les experimentacions relacionades amb el LVA i les seves optimitzacions successores.

És important destacar que el LVA continua preservant el cost computacional de l'algorisme LVF original, ja que no s'han afegit procediments dintre del bucle que tinguin un cost més alt que el càlcul de la inconsistència. Per tant, definim el cost computacional del LVA com a $\mathcal{O}(max \cdot |S|)$, on és reutilitzen les nomenclatures definides. Seguidament es mostra el pseudocodi de l'algorisme.

Algorisme 25: Las Vegas Filter Adaptative (LVA)

Entrada:
max - El nombre màxim d'iteracions
$\mathcal{J}$ - La mesura d'avaluació
$S(X)$ - Una mostra S descrita pel conjunt de variables X
p - Paràmetre indicador de la probabilitat d'èxit del generador de mides
α - Paràmetre regulador de la suma de probabilitat de variables.
β - Paràmetre regulador de la resta de probabilitat de variables.
Sortida:
Best - Millor solució trobada

$Best := X$
$\mathcal{J}_o := \mathcal{J}(S(X))$
$prob := init(0.5, |X|)$ `// Inicialització de les probabilitats a 0.5`
repeat *max* **times**
  $n := GeneradorAleatoriBinomial(|Best|, p)$
  $X' := SubconjuntAleatori(X, n, prob)$
  if $\mathcal{J}(S(X')) \geq \mathcal{J}_o$ **then**
    $Best := X'$
    $prob := Increment(X', prob, |X|, \alpha)$
  end
  else
    $prob := Decrement(X', prob, |X|, \beta)$
  end
end

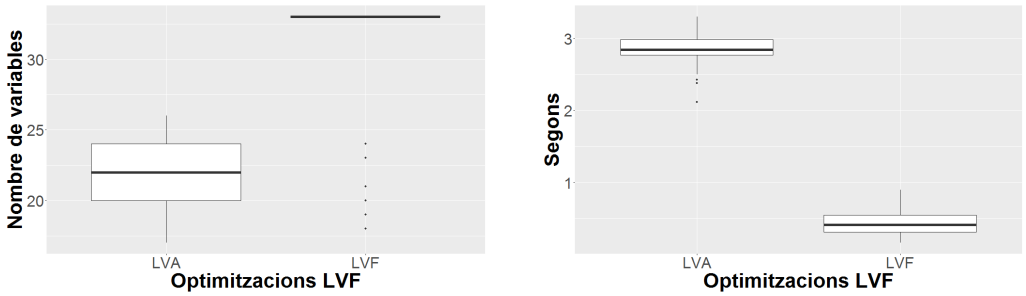
Amb els canvis que s'han aplicat en l'algorisme hem controlat la tendència que patien les probabilitats de les variables en la Modificació 8. És a dir, aquesta versió ja no sofreix d'un decrement tan gran en les variables i trobem una variabilitat més alta entre les probabilitats de les variables durant el transcurs de l'execució de l'algorisme.

En els resultats de l'experimentació podem observar que el LVA ha obtingut com a resultats subconjunt de variables amb una mida significativament inferior als obtinguts amb el LVF. L'únic conjunt de dades on trobem resultats similars pel que fa a la reducció de variables és en el conjunt de dades CDB. En el conjunt de dades CDI, trobem una gran diferència, el LVA aconsegueix reduir en mediana 11 variables del nombre de variables original, mentre que el LVA en mediana no aconsegueix reduir cap (Veure figura 7.1). D'aquesta manera podem afirmar que les optimitzacions aplicades en el LVA milloren la seva efectivitat en la reducció de variables en aquests conjunts de dades.

El temps d'execució en les diverses experimentacions realitzades a estat major el de l'optimització LVA que el de l'algorisme original LVF. La major diferència la trobem en el conjunt de dades CDC (Veure figura 7.1). Aquestes diferències en els temps d'execució no han estat massa significatives en la majoria d'exeprimentacions. D'aquesta manera basant-nos en la reducció de variables i els temps d'execució emprats per a cada algorisme recomenem el LVA envers el LVF pel que fa els resultats extrets en aquesta experimentació.

L'*accuracy* obtinguda amb els models predictius construïts amb el *Naive Bayes* i l'aprenentatge basat en arbres de decisió, ha disminuït lleugerament respecte a l'aconseguida amb el LVF en les diferents experimentacions. La mitjana d'aquestes

diferències en l'*accuracy* en cap experimentació ha superat el 4%, però només en el conjunt de dades CDC el LVA ha obtingut unes *accuracys* més altes. Aquest fet és conseqüència de què el LVA realitza una reducció de variables major que el LVF i per tant deu eliminar algunes variables que milloren l' *accuracy*. Aquest comportament no és un defecte del LVA, ja que tots els subconjunts de variables que obté l'optimització compleixen la restricció imposada per la mesura d'avaluació. Per tant, observem que la inconsistència no sempre ens guia cap a una millora o estancament en l'*accuracy*, si no que la seva funcionalitat és més similar a una heurística i no sempre és conseqüent amb l'*accuracy*.



(A) Mida a CDI (B) Segons a CDC

FIGURE 7.1: Resultats obtinguts amb LVF i LVA

7.2 Las Vegas Incremental amb Las Vegas Adaptative

En aquesta secció tractarem amb l'optimització *Las Vegas Incremental* (LVI)[8], la qual s'ha definit i explicat anteriorment en la secció, *Millores proposades del LVF* del capítol 2, *Estat de l'art*. En conseqüència, no tornarem a incidir en l'explicació. Inicialment, comprovarem el rendiment d'aquesta optimització amb els conjunts de dades finals i posteriorment, estudiarem com afecta introduir el LVA en el LVI.

7.2.1 Las Vegas Incremental

Recordem que l'algorisme LVI té l'objectiu de reduir els temps d'execució del LVF en conjunts de dades amb grans dimensionalitats. Aquesta tasca l'assoleix no utilitzant totes les instàncies del conjunt de dades, sinó que intentarà utilitzar les mínimes per a obtenir una bona reducció. Cal remarcar que l'algorisme en cap moment intenta millorar la reducció de variables que proposa el LVF, ja que el disseny de l'algorisme no pretenia millorar aquest aspecte del LVF.

Aquests aspectes esmentats s'han de tenir en compte a l'hora de valorar el rendiment de l'algorisme, ja que és evident que té un enfocament diferent del proposat per a nosaltres en la millora LVA. L'algorisme original LVI s'ha testat en els conjunts de dades finals seleccionats. S'ha definit la mida del subconjunt d'instàncies inicial com el 10% de la mida del conjunt d'instàncies original. Aquest valor és el recomanat pels autors de l'algorisme LVI[8].

Gràcies al fet que el LVI en les seves execucions repeteix molt poques vegades la seva iteració més exterior, la qual crida al LVF perquè realitzi max iteracions per a seleccionar el conjunt de dades a testejar, podem considerar que manté el cost computacional del LVF original, el qual és de $\mathcal{O}(max \cdot |S|)$. S'ha de considerar que

les iteracions realitzades amb el LVF constaran d'una mida de S força més reduïda que l'original, aquí es troba l'estalvi de temps de computació.

En els resultats de l'experimentació podem observar que aquesta optimització del LVF conserva la capacitat de reducció de variables del LVF, ja que presenta resultats molt similars a ell. Una particularitat molt beneficiosa, ja que recordem que no estem utilitzant totes les instàncies com fa el LVF. En l'experimentació amb els conjunts de dades CDW i CDV fins i tot ha estat capaç de reduir un nombre més elevat de variables.

En totes les experimentacions realitzades el LVI ha estat l'algorisme amb un temps d'execució menor, aquest fet és degut a no utilitzar totes les instàncies dels conjunts de dades i d'aquesta manera rebaixar els temps de còmput del càlcul de la inconsistència (veure figura 7.2). Per tant recomanem aquest algorisme en conjunts de dades d'altes dimensionalitats on el temps de còmput sigui un aspecte molta a tenir en compte.

L'*accuracy* obtinguda dels models predictius construïts amb aquesta versió ha estat molt similar a la que hem obtingut amb el LVF. Aquest fet denota que les variables seleccionades entre els algorismes han estat molt similars, ja que també hem obtingut la mateixa mida en els subconjunts de variables, com s'ha comentat anteriorment.

7.2.2 Las Vegas Incremental amb Las Vegas Adaptative

Combinar l'optimització de *Las Vegas Incremental* amb l'optimització desenvolupada en aquest projecte *Las Vegas Adapatative*, resulta molt senzill. Ja que el LVI funciona cridant a la versió original de l'algorisme LVF, s'ha decidit canviar aquesta crida perquè executi el LVA i adaptar els paràmetres d'entrada de l'algorisme amb els que necessita el LVA per a funcionar correctament. Ens referirem a aquesta combinació com *Las Vegas Incremental Adaptative* (LVI-A).

Aquesta combinació d'optimitzacions a priori és força interessant, ja que el LVI destaca per la millora de rendiment quant al temps d'execució en conjunts de dades amb grans dimensionalitats i el LVA aporta una fiabilitat major que el LVF en conjunts de dades amb un nombre de variables elevat. Per tant és interessant el comportament d'aquesta combinació d'optimitzacions en aquest tipus de conjunts de dades de grans mides. A continuació mostren el pseudocodi de la integració del LVA en el LVI.

Algorisme 26: Las Vegas Incremental Adaptative (LVI-A)

Entrada:
max - El nombre màxim d'iteracions
$\mathcal{J}$ - La mesura d'avaluació (inconsistència)
$S(X)$ - Una mostra S descrita pel conjunt de variables X
p - Percentatge de les instàncies utilitzades inicialment
Sortida:
X' - Millor solució trobada

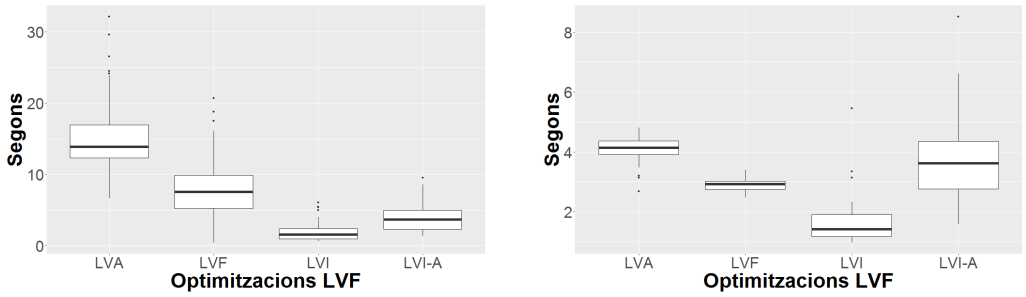
$\mathcal{J}_o := \mathcal{J}(S(X))$
$S_o = PorcioInicial(S, p)$
$S_f = S \backslash S_o$
repeat forever
  $X' := LVA(max, \mathcal{J}, S_o(X), p, \alpha, \beta)$
  if $\mathcal{J}(S(X')) \leq \mathcal{J}_o$ **then**
    return X'
  end
  else
    $C :=$ { *elements de* S_f *que provoquen inconsistència, utilitzant* X'}
    $S_o := S_o \cup C$
    $S_f := S_f \backslash C$
  end
end

Com que el LVA té el mateix cost computacional que el LVF, podem reutilitzar l'explicació del cost temporal en aquesta optimització i definir el seu cost temporal com a $\mathcal{O}(max \cdot |S|)$, el qual bé definit per l'execució del LVA.

Els resultats d'aquesta experimentació indiquen que introduir el LVA al LVI provoca clarament una major reducció de les variables respecte al LVI. En totes les experimentacions dels conjunts de dades podem observar aquest fenomen, menys en CDI i CDH que presenten la mateixa reducció de variables. En CDC podem observar que la diferència entre medianes és de 6 variables, reduint el LVI-A una quantitat de variables molt superior.

Aquest benefici en el potencial de reducció de variables es veu ofuscat per l'increment en els temps d'execució que presenta el LVI-A respecte el LVI. Uns temps d'execució molt similars als que presenta el LVA, això significa que perdem la finalitat principal d'aquesta versió, un temps d'execució més reduït en conjunts de dades de grans dimensionalitats (veure figura 7.2). Per aquest motiu, és preferible utilitzar directament el LVA, ja que presenta un major potencial de reducció de variables i un temps d'execució molt similar al LVI-A (pels conjunts de dades experimentats).

Aparentment la causa d'aquest increment de temps són els *overheads* que provoca la substitució del LVF pel LVA. Ja que el LVI realitza diverses crides al LVA durant la seva execució, aquesta diferència en els temps d'execució del LVF respecte al LVA es fa notar més. Respecte a l'*accuracy*, obtenim resultats molt similars a la versió original del LVI, lleugerament menors en alguns conjunts de dades com en el CDB, a causa del fet que el LVI-A ha reduït més les variables.



(A) Segons a CDM (B) Segons a CDV

FIGURE 7.2: Resultats obtinguts amb LVF, LVA, LVI, LVI-A

7.3 Quick Branch and Bound amb Las Vegas Adaptative

Aquest apartat girarà en torn a l'optimització *Quick Branch and Bound* (QBB)[9], introduïda i explicada en la secció *Millores proposades del LVF* del capítol 2, *Estat de l'art*. Utilitzarem una estructura similar a l'apartat anterior, començarem emprant la versió original del QBB en els conjunts de dades finals i per acabar, analitzarem la combinació del QBB amb la nostra optimització LVA.

7.3.1 Quick Branch and Bound

L'optimització *Quick Branch and Bound* és la combinació de l'algorisme LVF amb l'algorisme *Automatic Branch and Bound* (ABB); comença el LVF proposant una reducció inicial i acaba l'ABB intentant reduir-la més. La idea és molt bona perquè el LVF utilitza una cerca aleatòria, en canvi, l'ABB utilitza una cerca exhaustiva en l'arbre de solucions.

En aquesta versió, sí que esperem una reducció major de variables respecte a la versió original del LVF, ja que els autors[9] de l'optimització van declarar que normalment tenia un rendiment superior al LVF, ABB i al FOCUS. També, els autors afirmen que repartir equitativament el temps d'execució per a cada algorisme és una repartició molt robusta, ja que normalment obté els millors resultats[22]. Per a controlar el temps d'execució de cada algorisme utilitzarem el nombre d'iteracions que utilitza cadascun. En conseqüència, utilitzarem el mateix nombre d'iteracions per al LVF que per a l'ABB (50 per a cada algorisme).

A causa de que controlem l'optimització ABB amb un nombre determinat d'iteracions que anomenem max_{ABB} l'optimització ABB que utilitzem per al QBB tindrà un cost computacional de $\mathcal{O}(max_{ABB} \cdot |S|)$ ja que l'operació que es realitzarà amb un cost més alt per a cada iteració serà el càlcul de la inconsistència, el qual té un cost computacional de $\mathcal{O}(|S|)$. En aquest cas el LVF tindrà un cost computacional de $\mathcal{O}(max_{LVF} \cdot |S|)$, ja que realitzarà max_{LVF} iteracions, per tant com que aquestes crides es realitzen una darrera l'altre, podem determinar que el cost total de còmput del ABB és de $\mathcal{O}(max \cdot |S|)$, on $max = max_{ABB} + max_{LVF}$.

Els resultats obtinguts amb el QBB han estat molt positius pel que fa al seu potencial de reducció de variables, ja que presenta resultats amb un nombre de variables inferiors a totes les optimitzacions estudiades fins ara. D'aquesta manera el QBB redueix més les mides de les respostes que el LVA, en la nostra experimentació. La

diferència és altament observable en l'experimentació dels conjunts de dades CDV, CDW i CDH. Respecte a la versió original del LVF presenta una gran millora en la selecció de solucions amb menys variables (veure figura 7.3).

Els temps d'execució de les experimentacions són molt similars als de l'optimització LVA. Per tant són superiors a la versió original del LVF i al LVI. Tot i això, la diferència de temps respecte al LVF és molt lleugera. De manera clara només la trobem diferenciada en l'experimentació amb el conjunt de dades CDC. Per tant aquesta versió presenta un molt bon rendiment.

En termes d'*accuracy*, podem observar que aquesta versió presenta un lleuger decrement respecte a les altres alternatives estudiades. Això és degut al fet que les solucions presentades tenen un nombre de variables inferior, i com estem observant en aquesta experimentació aquest fet fa que la mitjana de l'*accuracy* és redueixi. Com ja s'ha explicat abans, aquest fet amb aquesta mesura d'avaluació no el podem controlar i l'algorisme està acomplint la seva tasca correctament, ja que basa els seus resultats en aquesta mesura d'avaluació.

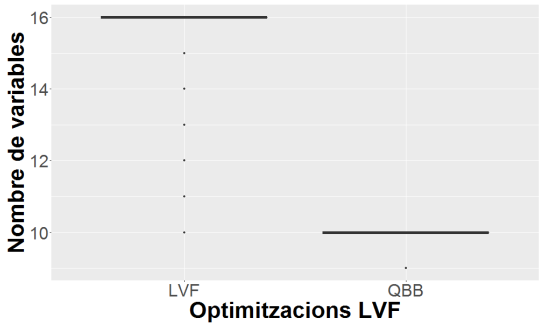


FIGURE 7.3: Mida dels resultats obtinguts amb LVF i QBB a CDV

7.3.2 Quick Branch and Bound amb Las Vegas Adaptative

Per a combinar l'enfocament que dóna l'optimització QBB amb la nostra optimització, substituirem la crida al LVF original per a la del LVA. És a dir, la primera cerca en l'espai de solucions es realitzarà amb l'optimització LVA i posteriorment al subconjunt de variables solució aplicarem l'ABB. També, haurem d'adaptar els paràmetres d'entrada de l'algorisme per incloure els necessaris per al LVA. Anomenarem aquesta combinació com *Adaptative Quick Branch and Bound* (QBB-A).

En les execucions del QBB-A, similarment amb les del QBB original, definirem la meitat de les iteracions totals per als dos algorismes involucrats (LVA i ABB), d'aquesta manera la comparació amb l'algorisme QBB estarà sota les mateixes condicions. A priori, amb aquesta combinació esperem obtenir una major reducció de variables de la que obtenim amb el QBB, ja que el LVA ha reduït més variables per execució que el LVF en la nostra experimentació. A continuació és mostra el pseudocodi del QBB-A.

Algorisme 27: Adaptative Quick Branch and Bound (QBB-A)

Entrada:
max - El nombre màxim d'iteracions
$\mathcal{J}$ - La mesura d'avaluació
$S(X)$ - Una mostra S descrita pel conjunt de variables X
p - Percentatge d'ús del LVA
Sortida:
X' - Millor solució trobada

$\mathcal{J}_o := J(S(X))$
$max_{LVA} := \lfloor max \times (p/100) \rceil$
$max_{ABB} := max - max_{LVA}$
$X' := LVA(max_{LVA}, \mathcal{J}, S_o(X), p, \alpha, \beta)$
$X' := ABB(S(X'), max_{ABB}, \mathcal{J})$

Per a determinar el cost computacional del QBB-A podem utilitzar l'explicació emprada per el QBB ja que la part que canvia de l'algorisme (substitució del LVF pel LVA) preserva el mateix cost computacional. D'aquesta manera, definim el cost computacional del QBB-A com $\mathcal{O}(max \cdot |S|)$, per tant, el mateix que el LVF.

Els resultats de l'experimentació amb aquesta optimització són molt similars als del QBB original. És a dir, hem obtingut una molt bona reducció de variables, superior a totes les altres optimitzacions i al LVF original. En els resultats del QBB-A podem apreciar que redueix lleugerament més que la versió original del QBB. En conseqüència, el QBB-A és l'optimització amb la qual hem aconseguit un major potencial de reducció de variables.

En contra partida els temps d'execució augmenten lleugerament respecte al QBB a causa del fet que l'execució inicial del LVA és lleugerament més lenta que amb el LVF. Pel que fa a l'*accuracy*, en general trobem resultats molt similars als obtinguts amb la versió QBB original, això és degut al fet que donen com a solució subconjunts de dades amb mides similars validats amb la mateixa mesura d'avaluació. En l'experimentació amb el conjunt de dades CDV, en els models predictius generats amb *Naive Bayes* amb les solucions del QBB-A, podem observar que presenten una *accuracy* major a les altres optimitzacions, per tant en aquesta experimentació aparenta que reduir molt el soroll de les variables ha aconseguit millors prediccions. Aquest fet l'esperàvem observar en altres conjunts de dades, però no ha estat així i haurem d'estudiar el cas a part (Veure Figura 7.4).

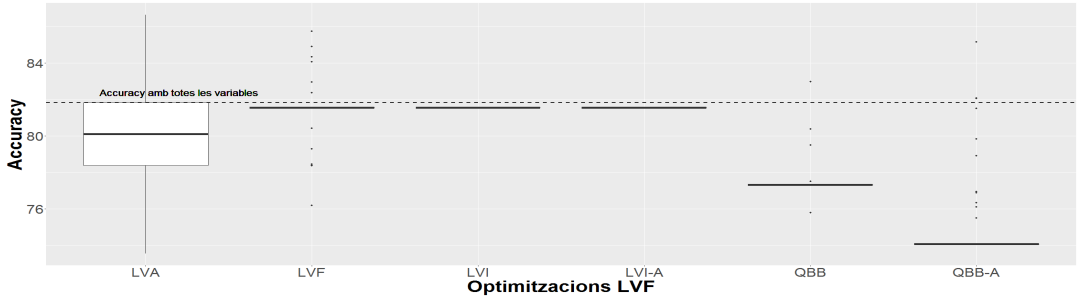
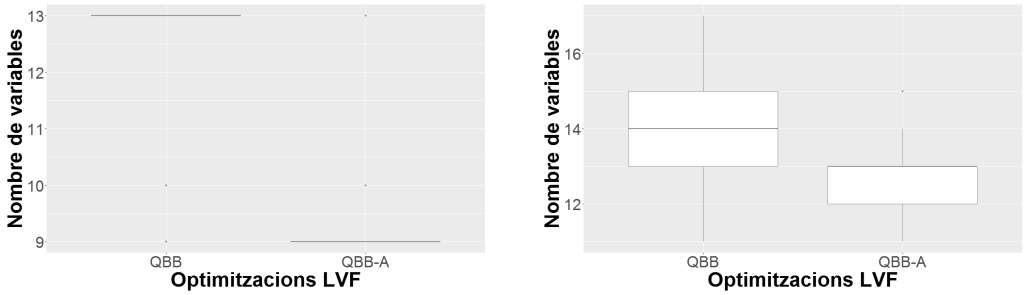


FIGURE 7.4: Accuracy de les optimitzacions basades en mètodes de filtre amb *Niave Bayes* a CDI

Posteriorment de l'experimentació explicada, es va intentar donar més pes en el QBB i el QBB-A a l'algorisme inicial, perquè és el que hem modificat. Vam definir que el 75% de les iteracions les executés el primer algorisme (en el QBB el LVF i en el QBB-A el LVA) i que el 25% restant les executes l'ABB. Per tant, $p = 75$. Com a resultat vam obtenir que el LVF empitjorava la seva reducció de variables i que el LVA es mantenia igual, inclús millorava en alguns casos.

Aquest fet és conseqüència de què el LVA aconsegueix una major reducció de variables i amb menys variabilitat respecte al LVF. Per tant, les solucions que el LVA transferia a l'ABB ja eren d'una mida menor i en donar-li més pes, la diferència s'ha vist incrementada. A continuació, mostrem dos dels resultats obtinguts en el canvi de percentatges del QBB-A i el QBB.



(A) Realitzada a CDV (B) Realitzada a CDC

FIGURE 7.5: Nombre de variables dels resultats de l'experimentació amb $p = 75\%$

En conclusió, amb aquestes experimentacions hem pogut observar com el LVA té un potencial de reducció de variables major que el LVF. No podem dictaminar cap optimització per sobre de les altres, ja que cadascuna té les seves qualitats. Basant-nos en els resultats de la nostra experimentació, podem recomanar.

- Si es treballa amb un conjunt de dades amb grans dimensionalitats i el temps d'execució és un factor molt important que pot arribar a generar un efecte de coll d'ampolla, recomanem la utilització del LVI, ja que amb un curt temps d'execució (envers les altres funcions) obté bones reduccions de variables.
- En canvi, si volem prioritzar al màxim la reducció de variables i el temps d'execució passa a un segon pla, es recomana l'optimització QBB-A, ja que és la que normalment obté unes majors reduccions de variables.

7.4 Las Vegas Hybrid

En aquesta secció presentem una nova optimització per al LVF. Aquesta nova optimització és molt situacional perquè el seu temps d'execució es disparà respecte a les altres optimitzacions estudiades. Aquesta versió té l'objectiu d'acabar amb el decrement de l'*accuracy* observat en les reduccions de variables de les optimitzacions estudiades anteriorment.

Aquesta optimització parteix del LVF i del LVW. Recordem que el LVW va ser presentat i explicat en la secció *Millores del LVF*, del capítol 2, *Estat de l'art*. Aquest algorisme aplica en el LVF un enfocament de mètode d'embolcall i utilitza com a

mesura d'avaluació $\mathcal{J}$ l'*accuracy* d'un model predictiu generat amb els subconjunts de variables de les solucions candidates. Un enfocament el qual condiciona que l'algorisme tingui un temps d'execució extremadament alt depenent de l'algorisme d'aprenentatge seleccionat, ja que ha d'entrenar i validar tants models predictius com iteracions de l'algorisme.

Amb l'optimització que plantegem la qual anomenem *Las Vegas Hybrid*, pretenem donar una idea mixta entre els mètodes de filtre i els mètodes d'embolcall, és a dir emprar un enfocament de mètode híbrid. Usualment els mètodes híbrids realitzen una primera cerca amb un mètode de filtre i amb la solució obtinguda apliquen un mètode d'embolcall per a validar-la o precisar-la. Nosaltres utilitzarem una metodologia similar per a la nostra optimització; l'optimització presentarà el mateix comportament que LVF, però utilitzarem dues mesures d'avaluació. La primera serà la inconsistència, la qual serà calculada per a tots els subconjunts de variables candidats. Si el subconjunt de variables presenta una inconsistència menor o igual al llindar màxim i la mida del subconjunt de variables candidat és inferior a la millor solució actual, s'entrenarà un model predictiu i s'avaluarà l'*accuracy* que s'obté amb ell. Si s'obté una *accuracy* màxima a un llindar mínim estipulat la solució serà correcta i passarà a ser la millor solució trobada.

A l'hora de desenvolupar l'optimització proposem dues opcions a escollir:

- **Llindar mínim d'*accuracy* constant**: El llindar mínim d'*accuracy* serà el mateix durant l'execució de tot l'algorisme.
- **Llindar mínim d'*accuracy* dinàmic**: El llindar mínim d'*accuracy* de l'algorisme variarà i serà l'*accuracy* obtinguda amb el model predictiu de la millor solució trobada fins al moment.

Nosaltres basarem l'algorisme en el llindar mínim d'*accuracy* dinàmic ja que l'enfocament d'aquesta secció és millorar l'*accuray* en les reduccions dels subconjunts de variables, per tant aquest llindar farà que l'algorisme acabi seleccionant la millor *accuracy* obtinguda.

En canvi, si només ens interesses un llindar mínim a superar, recomanem l'altra versió, ja que no descartarà tants candidats i és més probable trobar solucions amb un menor nombre de variables. A continuació és mostra el pseudocodi del LVH.

Algorisme 28: Las Vegas Hybrid (LVH)

Entrada:
max - El nombre màxim d'iteracions
$\mathcal{J}_{\mathcal{I}}$ - La primera mesura d'avaluació (la inconsistència)
$S(X)$ - Una mostra S descrita pel conjunt de variables X
$\mathcal{J}_{\mathcal{A}}$ - La segona mesura d'avaluació (l'*accuracy* d'un model predictiu determinat)
Sortida:
$Best$ - Millor solució trobada

$Best := X$
$\mathcal{J}_{\mathcal{I}o} := \mathcal{J}_{\mathcal{I}}(S(X))$
$\mathcal{J}_{\mathcal{A}best} := \mathcal{J}_{\mathcal{A}}(S(X))$
repeat max **times**
  $X' := SubconjuntAleatori(X)$
  if $(|X'| < |Best|$ **and** $\mathcal{J}_{\mathcal{I}}(S(X')) \leq \mathcal{J}_{\mathcal{I}o})$ **then**
    if $\mathcal{J}_{\mathcal{A}}(S(X')) \geq \mathcal{J}_{\mathcal{A}best}$ **then**
      $Best := X'$
      $\mathcal{J}_{\mathcal{A}best} := \mathcal{J}_{\mathcal{A}}(S(X'))$
    end
  end
end

La clau d'aquest algorisme és primer utilitzar la inconsistència com a mesura d'avaluació per a eliminar totes aquelles solucions que contenen variables que provoquen inconsistència (normalment aquestes variables aporten soroll a l'hora de la predicció), i posteriorment emprar l'*accuracy* per a millorar la selecció de solucions d'acord amb l'*accuracy*.

El problema que pot aparèixer en aquesta versió és utilitzar les dues mesures d'avaluació en cada iteració quan només busquem maximitzar una d'elles. Aquest problema pot aparèixer si no ajustem bé la inconsistència i definim un llindar massa feble o bé tractem amb un conjunt de dades el qual les variables no aporten inconsistència (fet molt estrany). Aquest problema seria greu, ja que el LVW ens aportaria la mateixa qualitat en les solucions i s'estalviaria el càlcul de la inconsistència, cal estudiar el comportament del LVH envers el LVW en els conjunts de dades seleccionats per a arribar a conclusions.

Un altre punt feble d'aquesta versió, el qual la majoria d'algorismes basats en mètodes d'embolcall o híbrids pateixen és que les variables escollides com a solució només estaran basades en un algorisme d'aprenentatge, per tant, la generalització a altres mètodes pot fallar, ja que pot succeir que funcionin millor amb altres variables.

El cost computacional d'aquesta versió canviarà severament respecte a les anteriors optimitzacions, ja que ara no només calcularem la inconsistència sinó que també calcularem l'*accuracy* dels models predictius construïts amb els candidats amb menys inconsistència que el llindar definit màxim. Aquest fet fa que el cost computacional de l'algorisme ascendeixi cap a $\mathcal{O}(max \cdot C)$, on definim C com el cost computacional de l'entrenament del model predictiu seleccionat i el cost computacional de la validesa del model predictiu. Amb aquest cost C en la majoria de situacions es complirà $C > |S|$, en termes de cost computacional per aquesta raó el nostre cost computacional passa a ser $\mathcal{O}(max \cdot C)$. La definició de C serà utilitzada en els següents apartats amb el mateix significat.

S'ha experimentat aquesta optimització amb els conjunts de dades finals utilitzant l'*accuracy* obtinguda de la construcció de models predictius basats en arbres de decisió com a segona mesura d'avaluació. S'ha decidit escollir els arbres de decisió per sobre del *Naive Bayes*, ja que amb ells s'han obtingut millors resultats quant a l'*accuray* quan generem models predictius amb poques variables. Tant l'algorisme d'aprenentatge basat en arbres de decisió com el mètode d'extracció de l'*accuracy* són els mateixos emprats en el càlcul de l'*accuracy* resultat de les optimitzacions anteriors. D'aquesta manera per a la selecció del millor arbre de decisió utilitzem la tècnica *one-standard-error rule* i pel càlcul de l'*accuracy* una validació creuada de 20 iteracions, aquests conceptes es troben explicats en el capítol 5, *Metodologia d'avaluació de millores*.

Els resultats d'aquesta optimització mostren un significatiu guany d'*accuracy* respecte a les optimitzacions basades en mètodes de filtre. Fet esperat, ja que utilitzem l'*accuracy* com una de les mesures d'avaluació. La diferència més significativa entre l'*accuracy* del LVH i la de les optimitzacions anteriors l'observem en l'experimentació realitzada en el conjunt de dades CDB, on per mitjana trobem l'*accuracy* del LVH un 7.5% més alta.

Per altra banda, els temps d'execució han augmentat d'una manera significativa. En la majoria de casos es denota la tendència que com més complexa sigui la generació del model predictiu la diferència entre els temps d'execució augmentarà molt, això és observable en les experimentacions amb conjunts de dades més simples (CDH, CDI) on la diferència a penes és apreciable, en canvi, en un el conjunt de dades CDM que presenta un nivell de complexitat més elevat la diferència és molt alta.

De l'anterior afirmació podem concloure que la utilització del LVH o optimitzacions que utilitzin l'*accuracy* com a mesura d'avaluació tindran un temps d'execució molt elevat en les següents situacions:

- El conjunt de dades presenta una gran complexitat quant a la seva classificació (alt nombre d'instàncies, variables, classes).
- L'algorisme de classificació seleccionat és molt complexa i presenta un temps d'entrenament o validació molt elevat.

Pel que fa al nombre de variables dels subconjunts de variables resultats del LVH trobem un gran decrement quant al potencial de reducció de variables, ja que la majoria de solucions tenen mides força més grans que les obtingudes amb optimitzacions basades en mètodes de filtre. Aquest fet és greu, ja que hauríem de trobar un equilibri entre *accuracy* i una bona selecció de variables.

7.5 Las Vegas Hybrid amb Las Vegas Adaptative

En aquest apartat intentarem augmentar el potencial de reducció de variables del LVH. En l'experimentació amb el LVH, s'han observat *outliers* en els resultats que tenien mides inferiors a la mediana de resultats i obtenien bona *accuracy*. Aquest fet evidencia que podem augmentar la reducció de variables en el LVH.

Per a millorar aquesta reducció de variables combinarem el LVH amb el major potencial de reducció que presenta el LVA respecta la versió original del LVF. En conseqüència, introduirem la nova mesura d'avaluació basada en l'*accuracy* al LVA. S'ha decidit que els subconjunts de variables que tinguin una inconsistència inferior al llindar màxim, però que no presentin una *accuracy* major a l'existent també rebin una penalització en la seva probabilitat de selecció. A continuació trobem el pseudocodi de l'optimització, la qual ens referirem a ella com a *Las Vegas Hybrid Adaptative*, LVH-A.

Algorisme 29: Las Vegas Hybrid Adaptative (LVH-A)

Entrada:
max - El nombre màxim d'iteracions
$\mathcal{J}_{\mathcal{I}}$ - La primera mesura d'avaluació (la inconsistència)
$S(X)$ - Una mostra S descrita pel conjunt de variables X
p - Paràmetre indicador de la probabilitat d'èxit del generador de mides
α - Paràmetre regulador de la suma de probabilitat de variables.
β - Paràmetre regulador de la resta de probabilitat de variables.
$\mathcal{J}_{\mathcal{A}}$ - La segona mesura d'avaluació (l'*accuracy* d'un model predictiu determinat)
Sortida:
$Best$ - Millor solució trobada

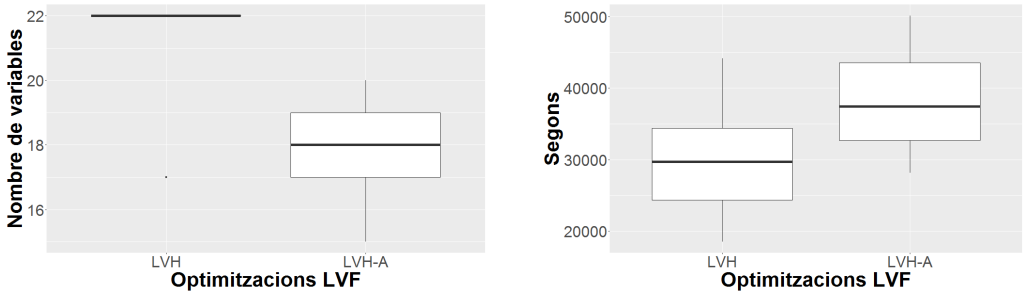
$Best := X$
$\mathcal{J}_{\mathcal{I}o} := \mathcal{J}_{\mathcal{I}}(S(X))$
$\mathcal{J}_{\mathcal{A}best} := \mathcal{J}_{\mathcal{A}}(S(X))$
$prob := init(0.5, |X|)$ `// Inicialització de les probabilitats a 0.5`
repeat max **times**
  $n := GeneradorAleatoriBinomial(|Best|, p)$
  $X' := SubconjuntAleatori(X, n, prob)$
  if $\mathcal{J}_{\mathcal{I}}(S(X')) \geq \mathcal{J}_{\mathcal{I}o}$ **then**
    if $\mathcal{J}_{\mathcal{A}}(S(X')) \geq \mathcal{J}_{\mathcal{A}best}$ **then**
      $Best := X'$
      $\mathcal{J}_{\mathcal{A}best} := \mathcal{J}_{\mathcal{A}}(S(X'))$
      $prob := Increment(X', prob, |X|, \alpha)$
    end
    else
      $prob := Decrement(X', prob, |X|, \beta)$
    end
  end
  else
    $prob := Decrement(X', prob, |X|, \beta)$
  end
end

Com que no afegim procediments amb un cost major que C dintre del bucle i tampoc apliquem noves computacions dintre de C, es manté el cost computacional en $\mathcal{O}(max \cdot C)$ en aquesta versió LVH-A.

Els resultats obtinguts amb aquesta optimització envers el LVH, denoten una major reducció de les variables, ja que en la majoria de les experimentacions s'han obtingut subconjunts de variables amb una mida inferior als obtinguts amb LVH. Aquest fet és molt observable en l'experimentació realitzada amb els conjunts de dades CDH, CDI i CDV (veure figura 7.6).

Tot i això, aquests resultats encara es troben lluny de les reduccions obtingudes amb les optimitzacions basades en mètodes de filtre. En general, el temps d'execució del LVH-A ha augmentat respecte al LVH, però si analitzem detalladament els resultats, podem observar que aquest augment de temps en els conjunts de dades on l'algorisme d'aprenentatge automàtic necessita més temps de còmput (CDM i CDW), no hi existeix una diferència proporcional gran (veure figura 7.6).

Un aspecte molt positiu d'aquesta versió és que tot i reduir més el nombre de variables que el LVH, trobem que l'*accuracy* obtinguda és més alta que amb el LVH. Això significa que amb menys variables aconseguim obtenir una *accuracy* La qual també és superior a l'obtinguda amb totes les variables. En l'experimentació realitzada amb el conjunt de dades CDB es pot observar aquest fet exageradament, la diferència entre mitjanes és de 7.5%, favorable al LVH-A.



(A) Mida a CDH (B) Segons a CDM

FIGURE 7.6: Resultats obtinguts amb LVH i LVH-A

7.6 Las Vegas Hybrid amb Adaptative Quick Branch and Bound

Aquest benefici d'*accuracy* amb els subconjunts de variables proposats pel LVH-A amb una mida inferior dels obtinguts pel LVH denoten que la reducció de variables gradual pot guiar-nos cap a millors solucions, on les variables que causen soroll en la predicció poden ser més fàcilment detectades.

A partir d'aquesta premissa sorgeix la idea de combinar el gran poder de reducció que té el QBB-A (Recordem que és l'optimització que ha reduït més els subconjunts de variables) amb l'enfocament de mètode híbrid que dóna el LVH.

Per al correcte funcionament d'aquesta combinació, haurem de definir una modificació per l'algorisme ABB perquè pugui funcionar amb les dues mesures d'avaluació que utilitza el LVH. Seguirem la mateixa idea i calcularem la inconsistència per a tots els candidats i el càlcul de l'*accuracy* serà emprat en aquells subconjunts de variables que superin els requeriments de la primera mesura d'avaluació. La modificació de l'algorisme l'anomenarem *Automatic Branch and Bound Hybrid* (ABB-H), a continuació s'exposa l'algorisme.

Algorisme 30: Automatic Branch and Bound Hybrid (ABB-H)

Entrada:
max - El nombre màxim d'iteracions
$\mathcal{J}_\mathcal{I}$ - La primera mesura d'avaluació (la inconsistència)
$\mathcal{J}_\mathcal{A}$ - La segona mesura d'avaluació (l'*accuracy* d'un model predictiu determinat)
$S(X)$ - Una mostra S descrita pel conjunt de variables X
Sortida:
$Best$ - Millor solució trobada

Procediment recursiu:
$Q := \emptyset$
for $x \in X$ **do**
  $enqueue(Q, \{X - x\})$
end
while $((not\ empty(Q))$ **and** $(it < max))$ **do**
  $X' := dequeue(Q)$
  if $((validar(X'))$ **and** $(\mathcal{J}(S(X')) \leq \mathcal{J}_o))$ **then**
    if $\mathcal{J}_\mathcal{A}(S(X')) \geq \mathcal{J}_{\mathcal{A}best}$ **then**
      $Best = X'$
      $\mathcal{J}_{\mathcal{A}best} = \mathcal{J}_\mathcal{A}(S(X'))$
    end
    $ABB(\mathcal{J}_\mathcal{I}, \mathcal{J}_\mathcal{A}, S(X'), Best, \mathcal{J}_{\mathcal{A}best}, it)$
  end
  $it = it + 1$
end

Inicialització:
$\mathcal{J}_{\mathcal{I}o} := \mathcal{J}_\mathcal{I}(S(X))$
$\mathcal{J}_{\mathcal{A}best} := \mathcal{J}_\mathcal{A}(S(X))$
$Best := X$
$it := 0$
$ABB(\mathcal{J}_\mathcal{I}, \mathcal{J}_\mathcal{A}, S(X), Best, \mathcal{J}_{\mathcal{A}best}, it)$ `// Crida inicial al procediment`

Un cop explicat el QBB-H, ja podem definir correctament l'optimització basada en la combinació del LVH i el QBB-A. Aquesta optimització primer realitzarà una selecció de variables amb l'optimització LVH-A, la qual s'ha definit en la secció anterior. Les variables seleccionades seran utilitzades com entrada del ABB-H, el qual realitzarà una cerca més exhaustiva en aquest subconjunt de variables més reduït. Per tant, repliquem el funcionament del QBB-A però adaptant-lo a la nova mesura d'avaluació afegida. Ens referirem a aquesta optimització com a *Adaptative Quick Branch and Bound Hybrid* (QBB-HA). Seguidament, trobem el seu pseudocodi.

Algorisme 31: Adaptative Quick Branch and Bound Hybrid (QBB-HA)

Entrada:
max - El nombre màxim d'iteracions
$\mathcal{J}_{\mathcal{I}}$ - La primera mesura d'avaluació (la inconsistència)
$S(X)$ - Una mostra S descrita pel conjunt de variables X
p - Percentatge d'ús del LVA
$\mathcal{J}_{\mathcal{A}}$ - La segona mesura d'avaluació (l'*accuracy* d'un model predictiu determinat)
Sortida:
X' - Millor solució trobada

$max_{LVA} := \lfloor max \times (p/100) \rceil$
$max_{ABB} := max - max_{LVA}$
$X' := LVH - A(max_{LVA}, \mathcal{J}_{\mathcal{I}}, S_o(X), p, \alpha, \beta, \mathcal{J}_{\mathcal{A}})$
$X' := ABB - AH(max_{ABB}, \mathcal{J}_{\mathcal{I}}, \mathcal{J}_{\mathcal{A}}, S(X'))$

L'ABB-H presentarà un cost computacional de $\mathcal{O}(max_{ABB} \cdot C)$, ja que el nombre d'utilitzacions del cost C es veu fitat per max_{ABB}, ja que són els nodes vàlids que pot explorar l'ABB-H. Tenint aquest concepte en compte i sabent que el LVH-A té un cost de $\mathcal{O}(max_{LVH-A} \cdot C)$, on max_{LVH-A} són el nombre d'iteracions que utilitzarà, demostrem que el cost del QBB-HA serà de $\mathcal{O}(max \cdot C)$, on $max = max_{LVH-A} + max_{ABB}$. En conseqüència manté el cost computacional del LVH.

Com que no afegim procediments amb un cost major que C dintre del bucle i tampoc apliquem noves computacions dintre de C, es manté el cost computacional en $\mathcal{O}(max \cdot C)$ en aquesta versió LVH-A.

Per a la realització de l'experimentació amb el QBB-HA, s'ha decidit utilitzar el mateix nombre d'iteracions pel LVH-A i pel QBB-HA, $p = 50$. Aquesta repartició d'iteracions ha estat la mateixa utilitzada en l'experimentació del QBB, també amb el QBB-A i ens ha donat bons resultats en les dues. Els valors dels paràmetres del LVA-H són els mateixos que s'han utilitzat en l'experimentació del LVA.

Els resultats de l'experimentació indiquen que el QBB-HA redueix aproximadament la mateixa quantitat de variables que l'optimització LVH-A, ja que no s'aprecien diferències significatives entre les mitjanes dels nombres de variables seleccionades. Fet el qual no s'esperava, ja que es creia que s'obtindria una major reducció de variables. Aquest succés és conseqüència d'endurir la selecció de la millor solució; gràcies al fet que ara anteposem l'*accuracy* a la mida de la solució, la reducció de les variables serà més lenta, en conseqüència, el subconjunt de variables que donarà el LVH-A com a punt de partida al ABB-H serà d'una mida força major, fet que empitjorarà el rendiment del ABB-H, ja que en ser un mètode de cerca exhaustiu el qual s'ha restringit en un nombre d'iteracions determinades, quedarà molt més espai de solucions sense explorar.

Els temps d'execució d'aquesta optimització augmenten molt lleugerament respecte als del LVH-A. Aquest augment de temps a penes és significatiu i es redueix proporcionalment al temps d'execució total conforme el conjunt de dades amb el qual es realitza l'experimentació és més complex.

Pel que fa l'*Accuracy* aconseguida amb el QBB-HA, obtenim uns valors molt similars respecte a la versió LVH-A, exceptuant l'experimentació realitzada amb el conjunt de dades CDI, en el qual apreciem una lleugera diferència entre les mitjanes favorable al QBB-HA.

D'aquesta manera segons els resultats de la nostra experimentació el QBB-H i el LVH-A presenten un rendiment molt similar, per sobre quant a variables reduïdes i *accuracy* obtinguda envers el LVH, però amb un temps d'execució superior al del LVH.

En conclusió, si necessitem una reducció de variables la qual contempli l'*accuracy* del model i el temps d'execució ha de ser reduït seleccionaríem el LVH. En canvi, si disposem de la capacitat d'executar temps d'execucions un 50% més llargs que els del LVH, recomanem molt fortament la utilització del LVH-A o del QBB-HA, ja que milloren molt la selecció de variables envers el LVH.

A priori, en l'optimització QBB-HA recomanem l'ús del mateix nombre d'iteracions pel LVH-A i l'ABB-H. Però, per a una major reducció de variables utilitzant el QBB-HA recomanem realitzar diferents proves augmentant el pes del LVH-A en l'algorisme, d'aquesta manera adaptarem la importància de la cerca inicial amb el nostre conjunt de dades.

7.7 Las Vegas Wrapper

Per a avaluar correctament les optimitzacions basades en una metodologia híbrida, necessitem estudiar el comportament de la versió LVW. D'aquesta manera podrem observar les principals diferències entre l'enfocament que dona el LFW i el LVH.

L'algorisme original *Las Vegas Wrapper* definit per Huan Liu i Rudy Setiono[55], presenta una estructura diferent del LVF. A continuació, mostrem el pseudocodi del LVW original.

Algorisme 32: Las Vegas Wrapper Original

Entrada:
k_{max} - El nombre màxim d'iteracions
$\mathcal{J}$ - La mesura d'avaluació (*accuracy* d'un model predictiu determinat)
$S(X)$ - Una mostra S descrita pel conjunt de variables X
Sortida:
$Best$ - Millor solució trobada

$k = 0$
$Best := X$
$\mathcal{J}_{Best} = 0$
repeat
  $X' := SubconjuntAleatori(X)$
  if $((\mathcal{J}(S(X')) > \mathcal{J}_{Best})$ **or** $((|X'| < |Best|)$ **and** $(\mathcal{J}(S(X')) = \mathcal{J}_{Best})))$ **then**
    $Best = X'$
    $\mathcal{J}_{Best} = \mathcal{J}(S(X'))$
    $k = 0$
  end
  $k = k + 1$
until $\mathcal{J}_{Best}$ no és actualitzat en k_{max} vegades $(k = k_{max})$

Podem observar que la seva regulació d'iteracions ha canviat respecte a la versió original del LVF, ja no trobem el paràmetre *max* que defineix el nombre màxim d'iteracions, sinó que ara l'algorisme s'aturarà quan realitzi K iteracions i no aconsegueixi millorar el resultat. Pel que fa a la resta de l'algorisme, és molt similar a la versió original del LVF, canviant la mesura d'avaluació per a l'*accuracy* d'un model predictiu construït a partir dels subconjunts de variables candidats. També, cal remarcar que la mida dels subconjunts de variables passa a un segon pla, ja que l'algorisme prioritza l'*accuracy* per sobre de la mida dels subconjunts de variables i només es valorarà al trobar dues solucions amb la mateixa *accuracy*.

Aquest canvi en la regulació d'iteracions empitjora l'estudi de l'algorisme envers les altres optimitzacions, ja que amb aquest algorisme el control d'iteracions no és tan exacte. És a dir, no es pot garantir que realitzi 100 iteracions com hem definit en el capítol 5, *Metodologia d'avaluació de millores*. Aquest fet suposa un problema, ja que estem tractant amb *anytime algorithms*[21], el que significa que la qualitat de les seves solucions millora gradualment a mesura que s'incrementa el seu temps de computació. En conseqüència, per a realitzar un estudi correcte, és necessari definir un nombre d'iteracions equivalents per a les diferents optimitzacions.

Per a solucionar aquest problema, s'ha decidit adaptar l'algorisme LVW original a una versió més similar a la versió original del LVF, i per tant més similar al LVH. A continuació, es mostra l'adaptació realitzada.

Algorisme 33: Las Vegas Wrapper Adaptat

Entrada:
max - El nombre màxim d'iteracions
$\mathcal{J}$ - La mesura d'avaluació (*accuracy* d'un model predictiu determinat)
$S(X)$ - Una mostra S descrita pel conjunt de variables X
Sortida:
$Best$ - Millor solució trobada

$Best := X$
$\mathcal{J}_{Best} := \mathcal{J}(S(X))$
repeat max **times**
  $X' := SubconjuntAleatori(X)$
  if $((\mathcal{J}(S(X')) > \mathcal{J}_{Best})$ **or** $((|X'| < |Best|)$ **and** $(\mathcal{J}(S(X')) = \mathcal{J}_{Best})))$ **then**
    $Best := X'$
    $\mathcal{J}_{Best} = \mathcal{J}(S(X'))$
  end
end

S'ha modificat el regulador d'iteracions de la versió original del LVW i s'ha convertit en l'utilitzat pel LVF original. La resta de l'algorisme és equivalent al LVW original. Gràcies a aquesta adaptació podrem estudiar més correctament el rendiment d'aquesta optimització envers les altres existents del LVF, ja que preserva les característiques del LVW però amb un llindar d'iteracions fix.

El cost computacional del LVW també bé definit com $\mathcal{O}(max \cdot C)$, ja que realitzarà max iteracions i per a cada iteració realitzarà l'aprenentatge d'un model predictiu i el validarà (extraurà *accuracy*), per tant el cost C.

Amb l'experimentació realitzada amb el LVW s'ha pogut observar una gran diferència respecte als temps d'execució obtinguts amb les optimitzacions basades en una metodologia híbrida. El cas del conjunt de dades CDW ha estat l'excepció, ja que els temps d'execució s'han igualat, aquest fet és perquè el cost del càlcul de la inconsistència en aquest conjunt de dades s'assimila al càlcul de l'*accuracy*, a causa del fet que en les experimentacions amb optimitzacions basades en mètodes de filtre ja obteníem uns temps d'execució extremadament alts.

Hem de considerar també que l'algorisme d'aprenentatge seleccionat no presenta gaire complexitat i dintre del domini d'algorismes d'aprenentatge automàtic és ràpid de calcular. Per tant, si seleccionéssim un algorisme amb una complexitat major, la gran diferència obtinguda en el temps d'execució respecte a les optimitzacions híbrides s'incrementaria molt més.

Quant a la mida dels subconjunts de variables solució, trobem que s'ha incrementat el rendiment envers les optimitzacions híbrides, ja que trobem que en l'experimentació dels conjunts de dades CDM, CDC, CDW s'ha realitzat una reducció similar, en canvi en els 4 conjunt de dades restants s'ha obtingut una reducció significativament major (fins i tot amb diferències de 7 variables entre mitjanes).

Aquesta major reducció de variables és deguda al fet que amb aquesta versió només utilitzem una única mesura d'avaluació i per tant els subconjunts candidats no hauran de passar pel filtratge de la inconsistència. La majoria de les solucions del LVW tenien nivells d'inconsistència molt baixos, encara que algunes d'elles eren lleugerament superiors al llindar d'inconsistència màxim definit per les anteriors optimitzacions. Aquests resultats ens indiquen que les solucions amb nivells alts d'inconsistència normalment podran reduir-se mantenint una *accuracy* similar o augmentant-la.

L'*accuracy* ha millorat molt lleugerament respecte a les optimitzacions híbrides, els casos on trobem una major millora són en l'experimentació amb el conjunt de dades CDH (amb un augment de 2.1%) i l'experimentació amb el conjunt de dades CDI (amb un augment de 1.2%). D'aquesta manera no trobem grans diferències entre el LVW i les optimitzacions basades en una metodologia híbrida pel que fa a l'*accuracy* obtinguda.

7.8 Las Vegas Wrapper Adaptative

En aquesta última optimització estudiada aplicarem un enfocament de metodologia d'embolcall a l'optimització *Las Vegas Adaptative*. L'algorisme manté la mateixa estructura que el LVA, però canviarem la mesura d'avaluació original (la inconsistència), per l'*accuracy*, la qual hem estat emprant en el LVW i les anteriors optimitzacions basades en metodologies híbrides.

Presentem aquesta optimització amb l'objectiu d'obtenir una major reducció de variables envers el LVW. Recordem que el LVW ja no dóna tanta importància a la reducció de variables, sinó que busca maximitzar l'*accuracy*, en canvi, amb aquesta versió i el seu ajustament dinàmic de les probabilitats de les mides dels subconjunts candidats heretat del LVA, obtindrem una reducció gradual de la mida dels candidats. Creiem que aquest ajustament pot ser beneficiós per la cerca de la millor solució

gràcies al fet que facilitarà a l'algorisme a generar candidats sense variables causants de soroll per a la predicció. L'ajustament de les probabilitats de cada variable també ajudarà a aquesta causa.

Cal remarcar que aquest ajustament dinàmic en aquesta versió funciona com una heurística, ja que no treballem amb un llindar definit, sinó que busquem maximitzar l'*accuracy*, per tant, es podria donar el cas que existeixin millors solucions amb una mida més elevada de la qual estem reduint. En canvi, en els casos anteriors on treballàvem amb la inconsistència i un llindar constant, sabíem que si trobàvem solucions correctes amb una certa mida, només podíem millorar els resultats cercant solucions amb una mida inferior.

També cal anunciar que amb aquest canvi de mesura d'avaluació, perdem una característica important envers l'anterior mesura d'avaluació i és que l'*accuracy* no presenta monotonia. Aquest fet impossibilita l'aplicació de l'ABB, ja que el seu procés de poda té el requisit d'emprar una mesura d'avaluació que presenti monotonia. Si no és així, l'algorisme descartarà solucions d'una mida inferior per què els seus nodes pares no obtenen uns resultats suficientment bons.

A aquesta optimització l'anomenarem *Las Vegas Wrapper Adaptative* (LVW-A) i la trobem exposada a continuació.

Algorisme 34: Las Vegas Wrapper Adaptative (LVW-A)

Entrada:
max - El nombre màxim d'iteracions
$\mathcal{J}$ - La mesura d'avaluació (*accuracy* d'un model predictiu determinat)
$S(X)$ - Una mostra S descrita pel conjunt de variables X
p - Paràmetre indicador de la probabilitat d'èxit del generador de mides
α - Paràmetre regulador de la suma de probabilitat de variables
β - Paràmetre regulador de la resta de probabilitat de variables
Sortida:
$Best$ - Millor solució trobada

$Best := X$
$\mathcal{J}_{Best} := \mathcal{J}(S(X))$
$prob := init(0.5, |X|)$ `// Inicialització de les probabilitats a 0.5`
repeat max **times**
  $n := GeneradorAleatoriBinomial(|Best|, p)$
  $X' := SubconjuntAleatori(X, n, prob)$
  if $((\mathcal{J}(S(X')) > \mathcal{J}_{Best})$ **or** $((|X'| < |Best|)$ **and** $(\mathcal{J}(S(X')) = \mathcal{J}_{Best})))$ **then**
    $Best := X'$
    $\mathcal{J}_{Best} = \mathcal{J}(S(X'))$
    $prob := Increment(X', prob, |X|, \alpha)$
  end
  else
    $prob := Decrement(X', prob, |X|, \beta)$
  end
end

En el LVW-A no s'afegeixen procediments més costos computacionalment que C per a cada iteració per tant conservarà el cost computacional del LVW original, el qual es $\mathcal{O}(max \cdot C)$.

Els resultats de totes les experimentacions amb els diferents conjunts de dades mostren que s'ha obtingut una major reducció de variables del LVW-A envers el LVW en tots els casos. Aquesta diferència la podem observar clarament en els conjunts de dades CDI, CDC, CDW, CDB, on la diferència entre medianes és de 4 variables aproximadament.

Els temps d'execució obtinguts en les experimentacions són molt similars al LVW, on alguns casos trobem un major temps pel LVW i en altres pel LVW-H. La diferència d'aquests casos és ínfima. Per tant, el temps d'execució no és diferencial. L'*accuracy* tampoc és diferencial entre les dues optimitzacions, ja que l'obtinguda amb el LVW-A no divergeix a penes de l'obtinguda amb el LVW, podem considerar que han obtingut uns resultats igual de bons quant a l'*accuracy*.

D'aquesta manera basant-nos en la nostra experimentació recomanem l'ús de l'optimització LVW-A per sobre del LVW, ja que obté un temps d'execució i una *accuracy* molt similars al LVW i aconsegueix reduir més el conjunt de variables original que el LVW. En conseqüència, normalment obtindrem subconjunts de variables d'una mida menor i una molt bona *accuracy*.

Respecta el dilema d'escollir entre l'enfocament híbrid i l'enfocament d'embolcall, s'han de considerar diversos factors.

- **Grans dimensions del conjunt de dades**: Si el conjunt de dades presenta una dimensionalitat molt gran, molt probablement l'execució de l'algorisme d'entrenament del model predictiu o la validació del model predictiu tindran un cost d'execució molt alt, en conseqüència el temps d'execució amb el LVW-A pot arribar a ser un problema. En aquest cas, per a treballar amb un temps d'execució més accessible i obtenir una bona *accuracy* recomanem l'ús del LVH-A o el QBB-HA, ja que realitzen una reducció de variables i preserven una *accuracy* per sobre d'un llindar determinat.

- **Gran complexitat de l'algorisme d'aprenentatge**: Si l'algorisme d'aprenentatge que es vol utilitzar presenta una gran complexitat en la seva computació, recomanem l'ús del LVH-A o el QBB-HA, ja que gràcies als seus enfocaments no construiran un model predictiu per a tots els candidats, només per als més prometedors. D'aquesta manera podrem obtenir una bona *accuray* en la reducció de variables sense veure'ns tan limitats pel temps d'execució de l'algorisme.

- **Cap restricció en torn el temps d'execució**: Si no tenim cap restricció entorn el temps d'execució o el nostre conjunt de dades no té grans dimensionalitats i l'algorisme d'aprenentatge no té un gran temps d'execució recomanem l'ús del LVW-A, ja que obtindrà solucions amb un nombre de variables inferiors i amb una *accuracy* molt similar o superior envers els mètodes híbrids proposats.

Capítol 8

Conclusions

En aquest capítol recollim moltes de les conclusions que s'han anat exposant durant el transcurs de la memòria, les quals hem obtingut durant la realització d'aquest treball de fi de grau.

El camp dels algorismes de selecció de variables és un camp que està en continu creixement i té un domini molt extens, en aquest projecte s'ha estudiat una ínfima part d'aquesta disciplina i s'ha apreciat el gran esforç tecnològic que està emprant l'ésser humà a través de noves propostes i estudis els quals uneixen moltes disciplines diferents per a millorar les solucions envers el problema de la selecció de variables.

Tots aquests esforços no són en va, ja que el problema de la selecció de variables cada cop cobrà més importància a causa del gran apogeu del Big Data i les tecnologies basades en l'anàlisi de dades. Amb les experimentacions realitzades en aquest projecte amb els conjunts de dades reals, **s'ha pogut observar que en tots els conjunts de dades apareixien variables que aportaven soroll a les prediccions i empitjoraven el seu rendiment**. Aquest fet indica que aquesta selecció de variables és necessària, ja que;

- Permet estalviar cost computacional a l'hora d'entrenar els models predictius i realitzar les prediccions.
- Els models predictius construïts a partir del subconjunt de variables rellevants (aquelles que no aporten soroll) tendiran a obtenir millors prediccions als construïts amb totes les variables d'un conjunt de dades al qual no li hem aplicat una selecció de variables.
- Permet estalviar esforços en els mesuraments de les variables que només aporten soroll.
- En comptar amb conjunts de dades amb un menor nombre de variables es facilita la visualització de patrons a les dades.
- Reduirem els requisits d'emmagatzemat del conjunt de dades.

Cal destacar també que en comptar cada cop amb conjunts de dades amb major dimensionalitats trobem també un nombre més elevat de variables que aporten soroll o són redundants.

Aquest increment de les dimensionalitats del conjunt de dades també cada cop restringeix més els algorismes de selecció de variables amb costs computacionals alts, per aquest motiu en aquest treball ens vam decantar per l'estudi del LVF, ja que

no presenta un cost computacional significativament alt i té un enfocament molt diferent de la resta d'opcions.

El nostre estudi ha utilitzat conjunts de dades reals i conjunts de dades artificials, el que ha ajudat a entendre els punts forts de cadascun. **Els conjunts de dades artificials ens permeten un major estudi del comportament de l'algorisme**, ja que podem comprendre tots els aspectes referents al conjunt de dades com la rellevància de cada variable, la redundància de cadascuna, etc. En canvi **els conjunts de dades reals acompleixen la tasca de validació de l'algorisme d'una manera molt més acurada**, ja que permeten aplicar directament l'algorisme en certs dominis reals els quals són molt difícils de recrear artificialment.

L'objectiu principal de les millores proposades inicialment del LVF era millorar el rendiment del LVF; solucions amb el mínim nombre de variables (que compleixin el requisit d'inconsistència) i amb poca variabilitat de la mida entre elles. **Per a millorar el rendiment de l'algorisme, s'han aplicat optimitzacions les quals redueixen l'arbitrarietat que resulta perjudicial per a l'algorisme**. Aquestes modificacions es recullen en l'optimització ***Las Vegas Adaptative*** **(LVA)** i es basen principalment en:

- La **mida dels subconjunts de variables successors** es defineix mitjançant un **generador de nombres aleatoris basat en una distribució binomial**.
- La distribució binomial del generador es generarà a partir de la mida de la millor solució trobada, per tant tindrem un **ajustament dinàmic de les probabilitats de les diferents mides dels subconjunts candidats**.
- Les **probabilitats de selecció de les variables** es veuran **incrementades o disminuïdes conforme els resultats** amb els subconjunts de variables els quals estiguin incloses.

Amb l'experimentació realitzada s'ha pogut observar que el **LVA aconsegueix majors reduccions de variables preservant el llindar màxim d'inconsistència envers el LVF**. Aquest és un aspecte molt positiu, ja que hem aconseguit millorar el rendiment de l'algorisme, aquesta millora de rendiment és molt difícil de quantificar exactament, ja que segons el conjunt de dades que s'utilitzi en l'experimentació s'obtenen rangs de beneficis diferents (aquest fet, és aplicable a totes les optimitzacions mesurades). Per tant, el benefici està condicionat pel conjunt de dades, però és apreciable en totes les experimentacions. També s'ha pogut concloure gràcies a les experimentacions realitzades amb els conjunts de dades artificials que tant el **LVA com el LVF, redueixen millor les variables redundants que les variables irrellevants**.

A la primera part de les millores finals del LVF, inicialment s'ha buscat estudiar el rendiment del LVA envers la versió original del LVF i algunes optimitzacions ja proposades per altres autors (LVI i QBB). Posteriorment, s'ha estudiat el rendiment obtingut de combinar aquestes optimitzacions amb el LVA. En els resultats d'aquestes optimitzacions s'ha observat que **la reducció de les variables venia acompanyada d'un lleuger empitjorament quant a l'*accuracy*** obtinguda amb el *Naive Bayes* i els arbres de decisió. Fet que ens indica que **obtenir solucions amb la inconsistència mínima, no condiciona totalment a què l'*accuracy* no disminueixi respecte a altres solucions amb el mateix nivell d'inconsistència o major**.

Per evitar aquest decrement de l'*accuracy* s'ha proposat la segona part de les millores finals del LVF, en la qual s'ha introduït el ***Las Vegas Hybrid* (LVH)**, una nova optimització del LVF la qual utilitzava un **enfocament híbrid** entre els mètodes de filtre i els mètodes d'embolcall. El LVH **utilitza inicialment la inconsistència per a eliminar els candidats** que superen el llindar màxim definit i posteriorment **cerca la millor solució basant-se en l'*accuracy*** obtinguda amb els models predictius construïts a partir dels diferents subconjunts de variables candidats. També, s'ha estudiat l'aplicació de diverses optimitzacions ja definides al LVH i s'ha estudiat el rendiment del LVW per a poder realitzar una comparativa amb el LVH i les seves optimitzacions.

Amb els resultats obtinguts de totes les experimentacions realitzades podem definir quines optimitzacions són més idònies per a la selecció de variables basant-nos en uns certs factors:

Si podem **acceptar un lleuger empitjorament de l'*accuracy*** respecte l'obtingut amb totes les variables del conjunt de dades, recomanem aquestes optimitzacions en aquestes situacions:

- Si es treballa amb un **conjunt de dades amb grans dimensionalitats** i el temps d'execució és un factor molt important que pot arribar a generar un efecte de coll d'ampolla, es recomana la **utilització del LVI**, ja que és l'optimització amb un temps d'execució menor gràcies al fet que no usa totes les instàncies del conjunt de dades. La seva selecció de variables manté la qualitat del LVF.
- En canvi, si **no treballem amb un conjunt de dades amb grans dimensionalitats** i no ens veiem tan limitats respecte al temps d'execució, recomanem l'**ús del QBB-A**, ja que és l'optimització amb la qual hem aconseguit majors reduccions de variables.

Si per contra, **no podem acceptar un empitjorament de l'*accuracy*** i desitgem millorar-la, recomanem les següents optimitzacions en aquestes diverses situacions (Cal destacar que aquestes optimitzacions basades en mètodes híbrids i d'embolcall augmenten molt el temps d'execució, si no és acceptable el temps de computació ni pels mètodes híbrids, haurem d'emprar les optimitzacions anteriors):

- Si el **conjunt de dades té una gran dimensionalitat**, molt probablement l'execució de l'algorisme d'entrenament del model predictiu o la validació del model predictiu tindran un cost d'execució molt alt, en conseqüència necessitarem una optimització la qual només realitzi el càlcul de l'*accuracy* en els candidats a priori més prometedors. Recomanem l'**ús del LVH-A o el QBB-HA**, ja que realitzaran una selecció de variables preservant un llindar mínim d'inconsistència.
- Si l'**algorisme d'aprenentatge** que es vol utilitzar presenta una **gran complexitat en la seva computació**, recomanem l'**ús del LVH-A o el QBB-HA**, ja que similarment que en el cas anterior, evitarem construir el model d'aprenentatge per a tots els candidats i només serà construït en els candidats més prometedors. En conseqüència no estarem tan restringits pel que fa al temps d'execució.
- Si per contra, no tenim **cap restricció entorn del temps d'execució** o el nostre conjunt de dades i algorisme d'aprenentatge no presenten les anteriors característiques, recomanem optar per la **utilització del LVW-A**, ja que obtindrà

solucions amb un nombre de variables inferiors i amb una *accuracy* molt similar o superior envers els mètodes híbrids proposats.

Capítol 9

Treball Futur

Durant la realització d'aquest projecte s'han detectat diferents oportunitats per a nous estudis en el LVF i les optimitzacions proposades. Aquestes noves propostes es desviaven molt del nostre projecte i no han estat tractades, ja que requereixen un estudi amb un enfocament més directe cap a elles.

En aquest projecte s'ha emprat majoritàriament com a mesura d'avaluació la inconsistència, fet que ha restringit la utilització de conjunts de dades amb variables contínues. En aquest treball es va decidir no basar les optimitzacions en la mesura d'avaluació, però el LVF i les optimitzacions definides, com per exemple el LVA, suporten l'ús de diferents mesures d'avaluació. En aquest sentit, és un algorisme molt polivalent i les seves respectives optimitzacions també. Cal anar amb compte, però, amb l'optimització QBB i les seves derivades, ja que aquest grup d'optimitzacions requereixen que la mesura d'avaluació presenti una monotonia similar a la que presenta la inconsistència.

En el nostre projecte hem experimentat amb les versions LVW i LVW-A, les quals gràcies al seu enfocament d'embolcall ens permeten tractar amb variables contínues, però a un cost computacional molt gran. També hem proposat l'opció de discretitzat el conjunt de dades abans d'aplicar la selecció de variables, però aquesta alternativa usualment produeix pèrdua d'informació en el conjunt de dades. Existeixen mesures d'avaluació alternatives basades en mètodes de filtre que redueixen aquest cost computacional i es podrien aconseguir solucions de qualitat. Podem prendre com a exemple l'algorisme ReliefF, el qual aplica una metodologia de filtre i es pot aplicar a conjunts de dades amb variables contínues.

D'aquesta manera proposem l'estudi del LVF i les optimitzacions definides en aquest treball amb mesures d'avaluació que divergeixin de la inconsistència i amb les quals podem aplicar la selecció de variables amb conjunts de dades que continguin variables contínues. Aquest nou enfocament podria obrir portes a estudiar models predictius basats en regressió i no només en classificació com s'ha fet en aquest projecte, ja que les variables contínues usualment donen un major rendiment que les categòriques en prediccions basades en regressió.

A part d'aquesta proposta, també tenim la possibilitat d'iniciar un estudi només centrat en el rendiment de les optimitzacions basades en mètodes híbrids definits en aquest projecte envers les optimitzacions basades en mètodes d'embolcall del LVF.

Aquest estudi hauria de comptar amb diferents conjunts de dades de diferents característiques (dimensions, tipologia de variables, etc.) i també de diferents algorismes d'aprenentatge amb rangs de complexitats diversos. En conclusió, s'ha

d'estudiar amb més aproximació quina alternativa és més rentable en diverses situacions i quina pèrdua de fiabilitat tenim del LVH al LVW.

Per a acabar, durant l'estudi del LVF s'ha detectat un gran potencial de paral·lelització en ell. Aquesta qualitat prové de la seva senzillesa, ja que les seves iteracions podrien ser dividides en diferents *threads* i realitzar les execucions independents. Els *threads* al finalitzar les seves execucions només haurien de posar en comú les seves solucions i cercar la que conté una mida inferior.

També el càlcul de la inconsistència presenta una bona oportunitat de paral·lelització, gràcies al fet que el seu càlcul no és necessari que es realitzi de manera seqüencial, ja que a penes té estats que depenguin de computacions anteriors.

En conseqüència, un investigador amb coneixements més directes en paral·lelització i concurrència podria optimitzar en gran manera el rendiment del LVF.

Capítol 10

Planificació temporal

En aquest capítol es descriurà la planificació temporal mitjançant tasques, amb l'objectiu d'assolir els objectius i subobjectius plantejats en el Capítol 1, *Introducció i abast*, i poder finalitzar el treball en el termini estimat. També s'enumeraran els principals recursos necessaris.

L'inici d'aquest projecte es va donar el 13 de Juliol i es preveu realitzar la lectura al torn de gener, d'aquesta manera el projecte ha de finalitzar el 18 de gener (una setmana abans del torn). Per tant, el termini de temps per a la realització del projecte és de 189 dies. La facultat ens aporta una xifra orientativa de 540 hores de durada la qual utilitzarem com a referència. Durant aquests dies es va treballar 4 hores diàriament (caps de setmanes inclosos), si realitzem els càlculs, ens resulten hores de més però aquest fet ens dotava de la capacitat d'ajustar-nos a imprevistos i ser més flexibles.

Gràcies al fet que no han sorgit imprevistos de cap mena s'ha pogut ampliar el projecte amb les hores de més que es van estimar en la fita inicial en la gestió del risc. Aquestes hores han estat emprades en l'aplicació d'una metodologia híbrida al LVF i l'estudi d'una metodologia d'embolcall existent per al LVF. Per a adaptar aquests canvis, les tasques finals s'han vist reconvertides en aquesta ampliació, ja que les anteriors tasques es van poder assolir abans de l'esperat.

10.1 Recursos necessaris

Durant aquest projecte es necessiten certs recursos bàsics pel seu correcte desenvolupament. A continuació trobem els principals recursos necessaris:

- **Recursos humans:** L'autor i el director del projecte.
- **Recursos de hardware:** S'utilitzarà un ordinador portàtil per a desenvolupar el projecte a causa del baix consum que genera.
- **Recursos de Software:** La majoria del programari que s'utilitzarà serà gratuït (tots els softwares menys el Gantter). Principalment s'usarà R[28] i RStudio[31] per la realització del projecte. Quant a la documentació, es realitzarà en LaTeX[33] i l'elaboració del diagrama de Gantt[56] amb l'ajuda de Gantter [35] un software especialitzat en la seva elaboració.

Podem apreciar que aquest projecte no necessita molts recursos, el que facilita la seva execució, ja que ens veiem condicionats per pocs factors externs.

10.2 Descripció de les tasques

En aquesta secció es detallaran les tasques realitzades, s'han intentat tractar el màxim d'individualment possible per a una millor planificació i flexibilitat. Les trobem agrupades en diferents conjunts de tasques per a una fàcil distinció de les diferents fases del projecte.

10.2.1 GP - Gestió del projecte

Aquest conjunt de tasques fa referència a totes les tasques assolides per a una correcta gestió del projecte.

- **GP.1 - Contextualització i Abast del projecte:** Es defineixen els conceptes bàsics del projecte, s'exposa i es justifica el problema, es determinen els objectius i la metodologia que se seguirà. Necessita una important recerca de l'estat de l'art perquè són les bases del nostre projecte.
- **GP.2 - Planificació temporal:** Es presenta una descripció de totes les tasques, una planificació temporal envers elles, la qual esta suportada amb un diagrama de Gantt i un informe de la gestió del risc. Aquesta tasca és molt important per al projecte, ja que marca el compàs de tot ell.
- **GP.3 - Pressupost i sostenibilitat:** S'elabora un informe de sostenibilitat on es tracten les tres diferents dimensions; econòmica, social i ambiental. També es defineix el pressupost, presentarà totes les partides pel desenvolupament de les activitats descrites en aquesta secció i s'estima el seu cost.
- **GP.4 - Reunions amb el director:** Conjunt de reunions periòdiques realitzades amb el director del projecte amb l'objectiu de millorar el seguiment i la resolució de dubtes. Aproximadament mitja hora per reunió.
- **GP.5 - Documentació:** Part clau en el desenvolupament de qualsevol projecte. Es tracta de l'elaboració de la memòria final del projecte. Aquesta documentació es realitza paral·lelament durant tot el desenvolupament del projecte amb altres tasques per a una evolució més eficient del treball.
- **GP.6 - Presentació:** Aquesta tasca es realitza un cop finalitzat el projecte. Es prepara el material de suport per a la presentació, el guió i un conjunt d'assaigs per a l'exposició del treball al torn de lectura.

10.2.2 TP - Treball previ

Com que el projecte va ser iniciat en període de vacances l'autor va realitzar un treball previ en el projecte amb un nivell de seguiment diferent de la metodologia seleccionada (Scrum[26]). Aquestes tasques realitzades serveixen de base per a la següent fase del projecte.

- **TP.1 - Estudi de la literatura:** Investigació del problema de la selecció de variables i posteriorment el cas concret del LVF[7].
- **TP.2 - Generació dels primers conjunts de dades artificials:** Es van generar tres conjunts de dades artificials per a l'avaluació futura de les millores implementades del LVF.

- **TP.3 - Primera implementació del LVF, LVI i QBB:** Millora de la implementació trobada a internet en R del LVF i implementació del LVI basada en l'article *Incremental Feautre Selection*[8] i del QBB[9].

10.2.3 FI - Fase inicial

En aquesta fase es van elaborar les eines base per al desenvolupament del projecte. Podem entendre aquesta fase com el desenvolupament d'un producte viable mínim (de l'anglès, *Minimum Viable Product*), un producte limitat però que compleix totes les funcionalitats, en el nostre cas, el software d'una millora per al LVF, però no el definitiu. En el Capítol 1, *Introducció i abast*, es defineixen els subobjectius que ha de complir aquest producte viable mínim.

- **FI.1 - Implementació d'un algorisme generador de problemes de selecció de variables:** Desenvolupament d'un algorisme el qual pugui afegir variables irrellevants i variables redundants als conjunts de dades generats en el treball previ.
- **FI.2 - Estudi de diferents algorismes d'aprenentatge:** Treball de recerca en l'algorisme més adequat i que tingui un millor rendiment pels nostres conjunts de dades.
- **FI.3 - Implementació inicial d'un algorisme avaluador de subconjunts de variables:** Desenvolupament d'un algorisme el qual sigui capaç d'avaluar diferents subconjunts de variables basant-se en el nombre de variables irrellevants i variables redundants seleccionades. També de la precisió del model predictiu generat.
- **FI.4 - Estudi inicial de millores pel LVF:** Recerca d'idees prometedores per a la millora del LVF.
- **FI.5 - Implementació de les millores inicials pel LVF:** Les millores seleccionades en la tasca anterior s'implementen R.
- **FI.6 - Avaluació de les millores inicials implementades del LVF:** Comparativa entre les optimitzacions implementades del LVF amb l'ajuda de l'algorisme avaluador de subconjunts de variables.

10.2.4 FM - Fase intermèdia

En aquesta fase es va tenir l'objectiu de millorar el producte viable mínim proposat en la fase anterior. És a dir, estudiar noves millores de l'algorisme que podem afegir en el producte viable mínim per a millorar el seu rendiment. També en aquesta fase s'inicià la investigació del LVI i QBB (introduït en el Capítol 2, *Estat de l'art*).

- **FM.1 - Millora de l'algorisme avaluador de subconjunts de variables:** Estudi de l'aplicació d'una puntuació de la rellevància de les variables rellevants perquè influeixi en l'avaluació que realitza l'algorisme.
- **FM.2 - Estudi de les millores pel LVF:** Realització d'una nova cerca de possibles noves optimitzacions per al LVF a partir dels resultats obtinguts.
- **FM.3 - Implementació de les millores pel LVF:** Implementació de les millores del LVF seleccionades en la tasca anterior.

- **FM.4 - Avaluació de les millores implementades del LVF:** Comparativa entre les optimitzacions implementades del LVF amb l'ajuda de la nova versió de l'algorisme avaluador de subconjunts de variables.
- **FM.5 - Estudi de les millores pel LVI i QBB:** Amb les millors optimitzacions del LVF és realitza un estudi de la seva aplicabilitat al LVI i al QBB.
- **FM.6 - Implementació de les millores pel LVI i QBB:** Implementació de les millores seleccionades en la tasca anterior per al LVI i al QBB.
- **FM.7 - Avaluació de les millores implementades del LVI i QBB:** Comparativa entre les optimitzacions implementades del LVI i del QBB amb l'ajuda de la nova versió de l'algorisme avaluador de subconjunts de variables.

10.2.5 FF - Fase final

A la fase final, primerament es van desenvolupar diferents comparatives entre les versions optimitzades dels algorismes i les seves versions originals. Posteriorment, es van estudiar dos enfocaments nous per a la selecció de variables amb el LVF. Es va tractar d'aplicar metodologies híbrides i d'embolcall com a optimitzacions del LVF. Aquests nous enfocaments van ser estudiats i comparats entre si conjuntament amb les anteriors optimitzacions basades en un mètode de filtre. Es van aportar les conclusions i els resultats finals del projecte.

- **FF.1 - Comparativa entre les millors optimitzacions del LVF, el LVI i el QBB:** Comparativa mitjançant diferents avaluacions i anàlisis de les millors optimitzacions obtingudes per al LVI, per al QBB i per al LVF i les seves respectives versions originals.
- **FF.2 - Estudi de la metodologia híbrida en el LVF:** Introducció de la metodologia híbrida en el LVF. Aquest és un camp inexplorat el qual s'estudia amb la definició d'una nova optimització anomenada *Las Vegas Hybrid* (LVH).
- **FF.3 - Estudi de la metodologia d'embolcall en el LVF:** Estudi de la metodologia d'embolcall en el LVF. Aquest camp ja ha estat explorat i s'ha plantejat una optimització anomenada *Las Vegas Wrapper* (LVW), però és necessari estudiar-lo per comparar els resultats amb el LVH.
- **FF.4 - Estudi d'optimitzacions pel LVH i el LVW:** Estudi de les diferents optimitzacions a aplicar en el LVH i en el LVW per a millorar el seu rendiment.
- **FF.5 - Anàlisi comparatiu final d'optimitzacions:** Anàlisis del comportament, rendiment i validesa de les diferents optimitzacions exposades. Es cerca definir en quines situacions és òptim l'ús de cadascuna. Es compara la metodologia de filtre, la metodologia d'embolcall i la metodologia híbrida amb les nostres diferents optimitzacions.

10.3 Estimacions de les tasques

A continuació a la Taula 10.1 trobem la duració en hores de cada tasca, les seves dependències i els recursos necessaris. S'han emprat un total de 614 hores de treball i les dependències s'entenen com prerequisits.

ID	Tasca	Dependències	Temps	Recursos
GP	**Gestió del Projecte**	**-**	**145 h**	**-**
GP.1	Contextualització i Abast del projecte:	TP.1	25 h	PC, LaTeX
GP.2	Planificació temporal	GP.1	15 h	PC, LaTeX
GP.3	Pressupost i sostenibilitat	GP.1	15 h	PC, LaTeX
GP.4	Reunions amb el director	GP.1	20 h	PC, LaTeX
GP.5	Documentació	-	60 h	PC, LaTeX
GP.6	Presentació	FF.5	10 h	PC, LaTeX
TP	**Treball previ**	**-**	**52 h**	**-**
TP.1	Estudi de la literatura	-	30 h	PC
TP.2	Generació dels primers conjunts de dades artificials	TP.1	7 h	PC, R, RStudio
TP.3	Primera implementació del LVF, LVI i QBB	TP.1	15 h	PC, R, RStudio
FI	**Fase inicial**	**-**	**107 h**	**-**
FI.1	Algorisme generador de problemes de selecció de variables	TP.2	10 h	PC, R, RStudio
FI.2	Estudi de diferents algorismes d'aprenentatge	FI.1	7 h	PC, R, RStudio
FI.3	Algorisme avaluador de subconjunts de variables	FI.2	15 h	PC, R, RStudio
FI.4	Estudi inicial de millores pel LVF	TP.3	30 h	PC
FI.5	Implementació de les millores inicials pel LVF	FI.4	15 h	PC, R, RStudio
FI.6	Avaluació de les millores inicials implementades del LVF	FI.5	30 h	PC, R, RStudio
FM	**Fase intermèdia**	**-**	**190 h**	**-**
FM.1	Millora de l'algorisme avaluador de subc. de variables	FI.3	20 h	PC, R, RStudio
FM.2	Estudi de les millores pel LVF	FI.6	40 h	PC
FM.3	Implementació de les millores pel LVF	FM.2	15 h	PC, R, RStudio
FM.4	Avaluació de les millores implementades del LVF	FM.3	40 h	PC, R, RStudio
FM.5	Estudi de les millores pel LVI i QBB	FM.4	30 h	PC
FM.6	Implementació de les millores pel LVI i QBB	FM.5	15 h	PC, R, RStudio
FM.7	Avaluació de les millores implementades del LVI i QBB	FM.6	30 h	PC, R, RStudio
FF	**Fase final**	**-**	**120 h**	**-**
FF.1	Comparativa entre les millores del LVF, el LVI i el QBB	FM.7	20 h	PC, R, RStudio
FF.2	Estudi de la metodologia híbrida en el LVF	FM.4	20 h	PC, R, RStudio
FF.3	Estudi de la metodologia d'embolcall en el LVF	FM.7	20 h	PC, R, RStudio
FF.4	Estudi d'optimitzacions pel LVH i el LVW	FF.1	30 h	PC, R, RStudio
FF.5	Anàlisi comparatiu final d'optimitzacions	FF.4	30 h	PC, R, RStudio

TAULA 10.1: Estimació de les tasques

10.4 Diagrama de Gantt

Nombre	Inicio	Fin
TP - Treball previ	**13/07/2020**	**04/09/2020**
TP.1 - Estudi de la literatura	13/07/2020	10/08/2020
TP.2 - Generació dels primers conjunts de dades artificials	10/08/2020	19/08/2020
TP.3 - Primera implementació del LVF, LVI i QBB	19/08/2020	04/09/2020
FI - Fase inicial	**14/09/2020**	**23/10/2020**
FI.1 - Algorisme generador de problemes de selec. de variables	14/09/2020	18/09/2020
FI.2 - Estudi de diferents algorismes d'aprenentatge	18/09/2020	22/09/2020
FI.3 - Algorisme avaluador de subconjunts de variables	22/09/2020	28/09/2020
FI.4 - Estudi inicial de millores pel LVF	28/09/2020	08/10/2020
FI.5 - Implementació de les millores inicials pel LVF	08/10/2020	14/10/2020
FI.6 - Avaluació de les millores inicials implementades del LVF	14/10/2020	23/10/2020
FM - Fase intermèdia	**23/10/2020**	**24/12/2020**
FM.1 - Millora de l'algorisme avaluador de subc. de variables	23/10/2020	30/10/2020
FM.2 - Estudi de les millores pel LVF	30/10/2020	12/11/2020
FM.3 - Implementació de les millores pel LVF	12/11/2020	17/11/2020
FM.4 - Avaluació de les millores implementades del LVF	17/11/2020	01/12/2020
FM.5 - Estudi de les millores pel LVI i QBB	01/12/2020	11/12/2020
FM.6 - Implementació de les millores pel LVI i QBB	11/12/2020	15/12/2020
FM.7 - Avaluació de les millores implementades del LVI i QBB	15/12/2020	24/12/2020
FF - Fase final	**24/12/2020**	**18/01/2021**
FF.1 - Comparativa entre les millores del LVF, el LVI i el QBB	24/12/2020	28/12/2020
FF.2 - Estudi de la metodologia híbrida en el LVF	28/12/2020	01/01/2021
FF.3 - Estudi de la metodologia d'embolcall en el LVF	01/01/2021	05/01/2021
FF.4 - Estudi d'optimitzacions pel LVH i el LVW	05/01/2021	11/01/2021
FF.5 - Anàlisi comparatiu final d'optimitzacions	11/01/2021	18/01/2021
GP - Gestió del projecte	**14/09/2020**	**25/01/2021**
GP.1 - Contextualització i Abast del projecte:	21/09/2020	29/09/2020
GP.2 - Planificació temporal	29/09/2020	05/10/2020
GP.3 - Pressupost i sostenibilitat	05/10/2020	12/10/2020
GP.4 - Reunions amb el director	05/10/2020	18/01/2021
GP.5 - Documentació	14/09/2020	18/01/2021
GP.6 - Presentació	18/01/2021	25/01/2021

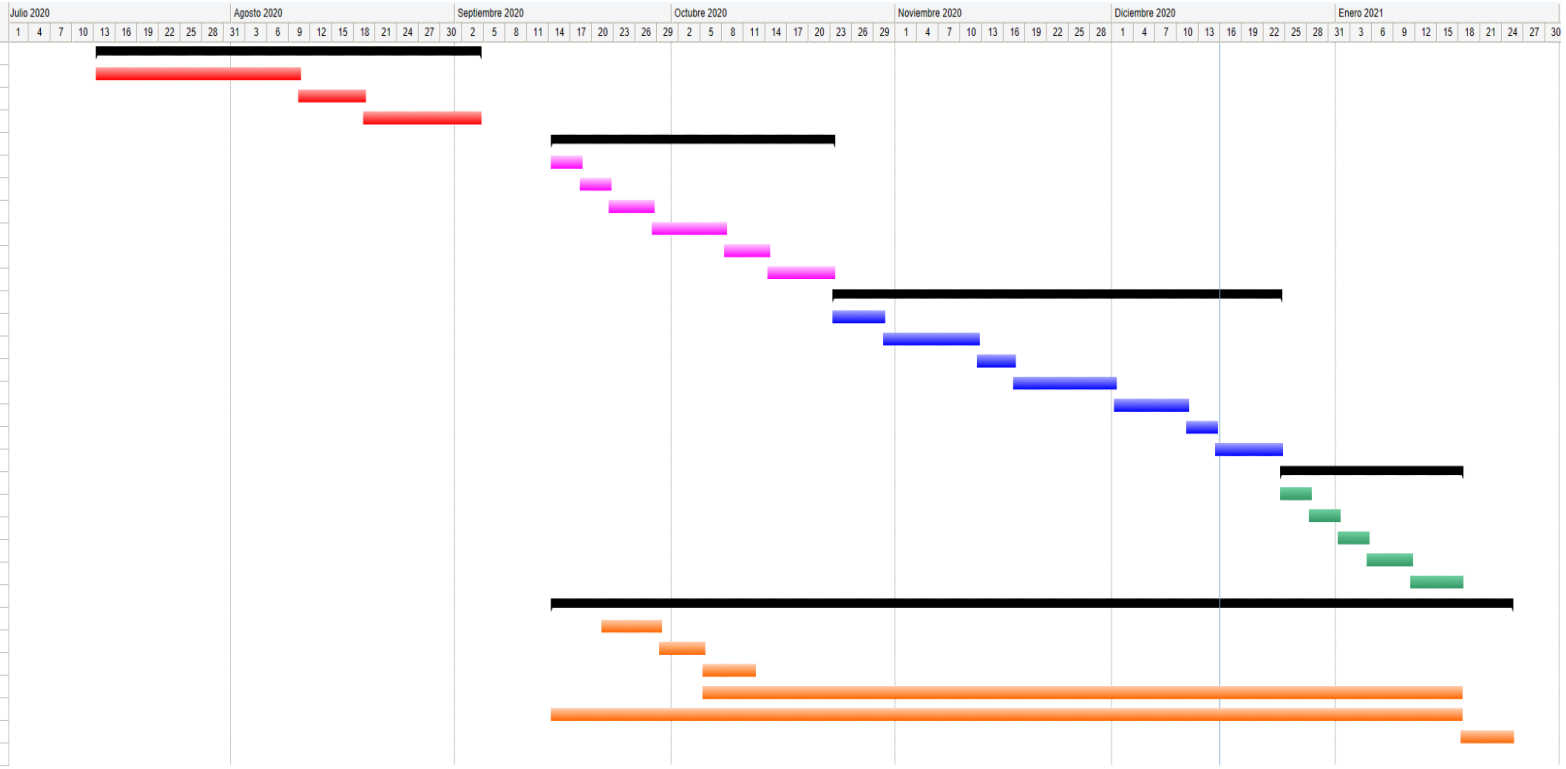


FIGURA 10.1: Diagrama de Gantt, generació pròpia amb l'eina Gantter

10.5 Gestió del risc

Com en tot projecte, trobem certs esdeveniments adversos els quals poden sorgir durant el transcurs d'aquest. És important preveure i anteposar-se a aquests problemes amb antelació. Aquesta previsió la realitzem prepararan des de l'inici del projecte un seguit d'alternatives que ens ajudarien a adaptar-nos a l'ocurrència d'algun dels problemes pronosticats.

En el Capítol 1, *Introducció i abast*, vam definir sis possibles riscos els quals podien convertir-se en obstacles pel desenvolupament del projecte. A continuació, explicarem el pla d'anticipació sobre aquests riscos.

Primerament, incidirem en el tractament de tres dels sis riscos definits. Aquests són els errors de disseny, errors d'implementació i errors d'anàlisi. Aquests tres riscos s'intentaran prevenir gràcies a la utilització d'una metodologia Agile com la Scrum[26] (definida en el Capítol 3, Metodologia i rigor), ja que gràcies a un seguiment més periòdic del director, el treball estarà més controlat, això reduirà en gran manera la probabilitat de cometre algun error dels llistats i també facilitarà la detecció de l'error en cas que es produeixi.

Quant al quart risc que vam definir en el capítol llistat, el risc de no trobar cap millora notable, el projecte no obtindria una solució correcta al problema de la millora de l'algorisme, però si aportaria un seguit d'estudis de possibles alternatives les quals reforçarien futures investigacions en el domini. Per tant, és molt important treballar en els formalismes del projecte per a facilitar la possible reutilització de la informació.

Una altra tipologia d'obstacles contemplada són els problemes greus, com ara la falta per malaltia o lesió. A l'estimació total de la duració de les tasques del projecte podem observar que s'ha obtingut un nombre d'hores amb setanta hores per sobre de la referència del temps estimat pel projecte. Aquestes hores de més són les que podríem adaptar davant d'aquest problema. També gràcies a la divisió en fases del projecte podríem reduir el pes de la fase intermèdia i arribar al termini de l'entrega correctament. Recordem que això és possible gràcies al fet que la fase inicial ja ens proporciona un seguit de millores per l'algorisme, és a dir un producte viable mínim, el qual la fase intermèdia intentarà millorar.

En referència a l'obstacle produït per una avaria de hardware, més concretament l'avaria de l'ordinador portàtil de l'autor, és un obstacle molt poc probable d'aparèixer però si es donés només afectaria la part econòmica del projecte, és a dir al pressupost. D'aquesta manera la solució seria comprar un ordinador nou per l'autor amb els diners de la partida de contingències, per a una millor explicació de la part econòmica, veure el Capítol 11, *Gestió econòmica*.

És important també remarcar l'especial situació en la qual ens trobem generada per la pandèmia de la COVID-19. Es preveuen nous tancaments de les universitats i possibles nous confinaments. Gràcies al fet que el projecte es desenvolupa individualment per l'autor amb la supervisió del director i no amb tot un equip, aquesta situació no afecta d'una manera tan directa al desenvolupament del projecte. La pandèmia ha causat que les reunions amb el director s'hagin de realitzar digitalment amb l'eina Google Meet[27] i l'autor realitza tot el projecte des de casa seva.

Durant el transcurs del projecte no es va donar cap d'aquests esdeveniments i es van poder utilitzar les hores estimades de més per a ampliar el projecte, conseqüentment millorar la qualitat de l'estudi.

Capítol 11

Gestió econòmica

En aquest capítol s'estudien quins són els factors econòmics implicats en aquest treball de fi de grau i així, determinar la seva viabilitat econòmica. Els costos han estat categoritzats en costos de personal i activitats (CPA), genèrics (CG), de contingència i d'imprevistos. També hi trobem descrit el control d'aquests costos durant el projecte.

11.1 Costos de personal i activitats

En aquesta secció partirem de la planificació per tasques realitzada en el capítol 10, *Planificació temporal*. Aquesta planificació per tasques és la que expressa el diagrama de Gantt, per tant, aquests costos estan lligats amb el diagrama i amb la planificació temporal per tasques. Abans d'entrar en detall del cost de cada tasca determinarem els papers que intervenen en aquest projecte i el seu sou mitjà. Podem trobar tres rols ben definits en aquest projecte:

- **Director del projecte**: Realitzarà les tasques relacionades amb la gestió i presentació del projecte.
- **Data Scientist**: En català, científic de les dades. Realitzarà les tasques d'investigació i anàlisis.
- **Programador**: Realitzarà les tasques referents a la programació.

Per a cadascun d'aquests tres rols s'ha investigat el seu sou mitjà a l'empresa de selecció de personal PagePersonnel[57], ja que faciliten les mitges dels sous en brut dels seus processos de selecció[58]. El paper de data scientist i programador serà emprat per l'autor del treball.

A la Taula 11.1 s'exposen els salaris per hora de cada rol en brut, la seguretat social que ha de pagar l'empresa per hora i la retribució total per hora. Pel càlcul del sou amb la seguretat social afegida s'ha multiplicat el sou brut per 1.3.

Rol	Sou brut hora	SS	Retribució
Director del projecte	25	7.5	32.5
Data Scientist	18.5	5.55	24.05
Programador	16	4.8	20.8

TAULA 11.1: Retribucions de cada perfil

Podem observar a la Taula 11.2 el nombre d'hores utilitzat per a cada rol amb totes les activitats i fases del projecte. També el sou brut per personal de cadascuna

d'elles i el sou brut amb la seguretat social de l'empresa afegida. En conclusió, el cost total del personal en brut per a totes les activitats seria de 11,956.50€ i amb la seguretat social afegida de **15,543.45€**.

ID	Tasca	Dir. projecte	Data Scientist	Programador	Cost	Cost amb SS
GP	**Gestió del Projecte**	**115 h**	**30 h**	**0 h**	**3430 €**	**4459 €**
GP.1	Contextualització i Abast del projecte:	25 h	0 h	0 h	625 €	812.5 €
GP.2	Planificació temporal	15 h	0 h	0 h	375 €	487.5 €
GP.3	Pressupost i sostenibilitat	15 h	0 h	0 h	375 €	487.5 €
GP.4	Reunions amb el director	10 h	10 h	0 h	435 €	565.5 €
GP.5	Documentació	40 h	20 h	0 h	1370 €	1781 €
GP.6	Presentació	10 h	0 h	0 h	250 €	325 €
TP	**Treball previ**	**0 h**	**40 h**	**12 h**	**932 €**	**1211.6 €**
TP.1	Estudi de la literatura	0 h	30 h	0 h	555 €	721.5 €
TP.2	Generació dels primers conjunts de dades artificials	0 h	7 h	0 h	129.5 €	168.35 €
TP.3	Primera implementació del LVF, LVI i QBB	0 h	3 h	12 h	247.5 €	321.75 €
FI	**Fase inicial**	**0 h**	**95 h**	**12 h**	**1949.5 €**	**2534.35 €**
FI.1	Algorisme generador de problemes de selecció de variables	0 h	10 h	0 h	185 €	240.5 €
FI.2	Estudi de diferents algorismes d'aprenentatge	0 h	7 h	0 h	129.5 €	168.35 €
FI.3	Algorisme avaluador de subconjunts de variables	0 h	15 h	0 h	277.5 €	360.75 €
FI.4	Estudi inicial de millores pel LVF	0 h	30 h	0 h	555 €	721.5 €
FI.5	Implementació de les millores inicials pel LVF	0 h	3 h	12 h	247.5 €	321.75 €
FI.6	Avaluació de les millores inicials implementades del LVF	0 h	30 h	0 h	555 €	721.5 €
FM	**Fase intermèdia**	**0 h**	**166 h**	**24 h**	**3455 €**	**4491.5 €**
FM.1	Millora de l'algorisme avaluador de subc. de variables	0 h	20 h	0 h	370 €	481 €
FM.2	Estudi de les millores pel LVF	0 h	40 h	0 h	740 €	962 €
FM.3	Implementació de les millores pel LVF	0 h	3 h	12 h	247.5 €	321.75 €
FM.4	Avaluació de les millores implementades del LVF	0 h	40 h	0 h	740 €	962 €
FM.5	Estudi de les millores pel LVI i QBB	0 h	30 h	0 h	555 €	721.5 €
FM.6	Implementació de les millores pel LVI i QBB	0 h	3 h	12 h	247.5 €	321.75 €
FM.7	Avaluació de les millores implementades del LVI i QBB	0 h	30 h	0 h	555 €	721.5 €
FF	**Fase final**	**5 h**	**90 h**	**25 h**	**2190 €**	**2847 €**
FF.1	Comparativa entre les millores del LVF, el LVI i el QBB	0 h	20 h	0 h	370 €	481 €
FF.2	Estudi de la metodologia híbrida en el LVF	0 h	20 h	0 h	370 €	481 €
FF.3	Estudi de la metodologia d'embolcall en el LVF	0 h	20 h	0 h	370 €	481 €
FF.4	Estudi d'optimitzacions pel LVH i el LVW	0 h	5 h	25 h	492.5 €	640.25 €
FF.5	Anàlisi comparatiu final d'optimitzacions	5 h	25 h	0 h	587.5 €	763.75 €
TOTAL	**Total CPA del projecte**	**120 h**	**421 h**	**73 h**	**11956.5 €**	**15543.45 €**

TAULA 11.2: Estimació del cost de personal a les tasques de la planificació temporal.

11.2 Costos Genèrics

En aquesta secció es descriuran els costos genèrics del projecte, són tots els costos independents a les tasques. Considerem com a costos genèrics les amortitzacions, la factura d'internet, el consum elèctric i l'espai de treball.

11.2.1 Amortitzacions

Són aquells costos els quals no s'han d'imputar de manera completa en el nostre projecte, ja que són amortitzables en altres tasques o projectes. Només imputem el cost de la part proporcional que s'ha treballat en aquest projecte. És a dir 7 mesos, els planificats en el Capítol 10, *Planificació temporal*.

- **Hardware**: Per la realització del projecte només es va necessitar un ordinador de gamma mitjana d'un cost de 750€. Hisenda permet amortitzar el hardware en 4 anys, per tant 48 mesos. Així és que, l'amortització d'aquest recurs és de $(7/48) \times 750€ =$ **109.38€**.

- **Software**: Tot el programari utilitzat és gratuït exceptuant el software Gantter[35], el qual té un cost mensual de 5 euros. Per tant, el cost del recurs és de $5€ \times 7 = \mathbf{35€}$. Aquest tipus de cost no és pròpiament una amortització, ja que és una subscripció mensual, però normalment els costos de software són amortitzables, per aquest motiu s'ha introduït en aquesta secció.

D'aquesta manera podem concloure que el cost total de les amortitzacions és de $109.38€ + 35€ = \mathbf{144.38€}$.

11.2.2 Factura d'internet

És indispensable per l'elaboració del projecte una connexió a internet, per tant hem d'afegir el cost proporcional d'una tarifa mensual d'internet. Com que s'han treballat diàriament 4 hores i el cost mensual de la tarifa d'internet utilitzada és de 43.95€, el cost del recurs és de $7 \times 43.95€ \times (4/24) = \mathbf{51.28€}$.

11.2.3 Consum elèctric

Segons Selectra[59], un comparador de tarifes de llum, el cost mitjà actual de la llum a Espanya és de 0.10445€/kWh [60]. A partir d'aquesta informació i saben que en totes les 614 hores treballades es requerirà l'ús del portàtil i de l'internet, podem estimar els costos com podem veure a la Taula 11.3.

Dispositiu	**Potència**	**Hores**	**Consum**	**Cost**
Ordinador portàtil	60 W	614 h	36.84 kWh	3.85 €
Router WiFi	14 W	614 h	8.596 kWh	0.90 €
Total	**-**	**-**	**45.436 kWh**	**4.75 €**

TAULA 11.3: Consum elèctric dels dispositius

11.2.4 Espai de treball

Atès a les circumstàncies especials de la COVID-19 tot el projecte es va desenvolupar des de casa i per tant no es contemplen costos en espai de treball.

11.2.5 Cost genèric total

La Taula 11.4 mostra el resum dels costos genèrics del projecte, i el total de la suma d'aquests costos.

Domini	**Cost**
Amortitzacions	144.38 €
Factura d'internet	51.28 €
Consum elèctric	4.75 €
Espai de treball	0 €
Total CG	**200.41 €**

TAULA 11.4: Estimacions dels costos genèrics

11.3 Contingències

És important contemplar un sobrecost afegit per a cobrir obstacles o imprevistos no anticipats. Tenint en compte que en aquest projecte es desenvolupa una investigació amb un alt nivell d'incertesa, la qual pot propiciar problemes no previstos, es va decidir fixar un 15% de sobrecost en contingències.

Calcularem el cost d'aquesta manera: $(Total\ CPA + Total\ CG) \times 0.15$.
Per tant, **cost contingències** = $(15543.45\ € + 200.41\ €) \times 0.15 =$ **2361.58 €**.

11.4 Imprevistos

Per últim, afegirem el cost que suposaria afrontar els diferents problemes previsibles definits en el Capítol 10, *Planificació temporal*. Aquests costos s'afegeixen multiplicats per l'estimació de la probabilitat de què succeeixin. Només són presents aquells problemes que suposen un increment del pressupost i s'ha prescindit d'afegir els problemes relacionats amb l'avaria de hardware, lesions o malalties, ja que són massa difícils de predir i es contemplen en les contingències. A la Taula 11.5 podem veure-hi el càlcul.

Problema	**Cost Real**	**Probabilitat**	**Cost**
Error de disseny (Increment 20h disseny)	481 €	15%	72.15€
Error d'implementació (Increment 10h implementació)	208 €	20%	41.60€
Error d'anàlisi (Increment 20h anàlisi)	481 €	15%	72.15€
Total Imprevistos	-	-	**185.90 €**

TAULA 11.5: Estimacions dels costos per imprevistos

11.5 Cost total del projecte

Podem observar a la Taula 11.6 el cost del projecte amb tots els costos totals de les seccions anteriors.

Tpidus de cost	**Cost**
CPA	15543.45 €
CG	200.41 €
Contingències	2361.58 €
Imprevistos	185.90 €
Total	**18291.34 €**

TAULA 11.6: Estimació total del cost del projecte

11.6 Control de gestió

Per al correcte seguiment del pressupost presentat es van realitzar periòdicament revisions dels costos per tasca. D'aquesta manera podríem detectar desviacions en els nostres costos estimats i a quines tasques es produeixen.

Es van utilitzar els següents indicadors numèrics:

Desviació en cost = *(Cost Estimat − Cost Real) × Consum Hores Real*
Desviació en eficiència = *(Cost Hores Estimat − Cost Hores Real) × Cost Estimat*
Desviació total en cost = *Cost Estimat − Cost Real*
Desviació total en hores = *Hores Estimades − Hores Reals*

Aquestes desviacions si són degudes als imprevistos definits, es va decidir utilitzarar la partida d'imprevistos per cobrir-ne els costos. Si aquesta partida no fos suficient, es contemplarien dues opcions; retallar la investigació en algunes tasques i així alliberar costos (gràcies al fet que primer desenvolupem un producte viable mínim, aquesta opció és aplicable) o recórrer a l'ús de la partida de contingència.

En canvi, si les desviacions provinguessin d'imprevistos no anticipats com per exemple un error de hardware, malaltia o lesió, s'utilitzaria la partida de contingència per afrontar els costos afegits a causa d'aquest problema. Si amb l'ajuda d'aquesta partida tampoc es pogués afrontar l'imprevist, s'haurien de retallar tasques de la investigació (com s'ha explicat anteriorment) per a poder assegurar arribar correctament al termini de l'entrega.

Capítol 12

Informe de sostenibilitat

En tot projecte d'enginyeria és necessari realitzar un informe de sostenibilitat en el qual es valorin les tres dimensions que inclou la sostenibilitat; la dimensió ambiental, la dimensió econòmica i la dimensió social. En aquest capítol, es presenta l'informe de sostenibilitat d'aquest projecte.

A mesura que els temps avança i la tecnologia prospera, l'esforç per a transmetre el coneixement referent a la sostenibilitat ha d'anar en augment, perquè els reptes proposats per la sostenibilitat són més difícils d'assolir i la nostra societat ha de ser conscient d'ells i conèixer les seves conseqüències. Per aquesta raó és molt important que des de l'enfocament de les TIC assumim aquest repte amb responsabilitat i aportem solucions de qualitat, ja que tenim un gran impacte sobre les tres dimensions.

Per concloure, usualment en l'elaboració d'un projecte les dimensions econòmiques i socials solen guanyar un pes extra envers la dimensió ambiental, ja que aporten un benefici més directe a l'empresa. Aquesta és una pràctica viciosa la qual en un futur s'haurà d'eliminar, i anivellar les tres dimensions per igual, ja que si no regulem la petjada ecològica i no aportem solucions amb l'objectiu de minimitzar-la, les nostres generacions futures tindran greus problemes amb el medi ambient.

12.1 Dimensió ambiental

L'impacte ambiental que ha suposat aquest projecte ha estat definit principalment pel consum energètic de l'ordinador portàtil amb el qual l'autor ha desenvolupat el projecte. S'ha realitzat una aproximació del consum energètic basada en el consum de l'ordinador portàtil i s'ha obtingut un consum de 36.84 kWh, el que és equivalent a 132624000J. Un impacte ambiental molt baix, el qual s'ha intentat minimitzar amb una sèrie d'accions emprades, les quals seguidament explicarem. L'ordinador portàtil que l'autor ha utilitzat en el desenvolupament del TFG també ha estat emprat al llarg de tot el grau, d'aquesta manera s'amortitza el cost ambiental de fabricació i el seu consum energètic és considerablement inferior a un ordinador de sobretaula. També, en desenvolupar-se des de casa s'ha aprofitat la xarxa d'internet d'ella i no s'ha generat un consum energètic extra. S'ha estalviat l'ús del paper, així doncs amb aquesta limitació els documents han estat tots en format virtual i conseqüentment s'ha reduït l'impacte ambiental.

Com que s'han aportat solucions que tenen un molt bon equilibri entre qualitat de la solució i cost computacional envers molts algorismes actuals (sigui el cas del LVA), això pot implicar una millora ambiental, ja que l'usuari és capaç de resoldre

el problema de selecció de variables sense un cost computacional extremadament alt, el que podem veure traduït com un consum d'energia menor. Aquest fet es veu amplificat si contextualitzem amb el gran increment de la mida que estan sofrint els conjunts de dades amb el *Big data*. La nostra solució té la capacitat de tractar amb grans conjunts de dades i aportar informació per a la seva reducció de variables, d'aquesta manera pot reduir el consum energètic de manteniment i recol·lecció d'aquestes variables, també d'entrenament i predicció dels models predictius.

12.2 Dimensió econòmica

El cost total del projecte ha estat descrit detalladament en el capítol 11, *Gestió econòmica*, on s'han desglossat els costos segons la seva tipologia. Com s'exposa en el capítol 10, *Planificació temporal*, havíem sobreestimat un seguit d'hores per a ajustar més correctament el pressupost, aquestes hores finalment s'han acabat utilitzant totes, gràcies a elles hem pogut acomplir un projecte de més qualitat. Aquest fet no ens ha permès rebaixar el cost del projecte en aquest aspecte com s'havia plantejat en la fita inicial del projecte. En canvi, amb les mesures preses per reduir l'impacte ambiental sí que s'ha obtingut un estalvi, principalment per l'estalvi energètic que han suposat.

El projecte s'ha ajustat correctament amb el pressupost presentat en el capítol 11, *Gestió econòmica*, aquest objectiu ha estat fàcil d'assolir, ja que era un projecte que no necessitava molts recursos per a la seva realització. Pel que fa als usuaris que utilitzessin les optimitzacions amb un baix cost computacional desenvolupades en aquest projecte podrien gaudir de l'estalvi energètic que s'ha detallat en la dimensió ambiental d'aquest informe, el qual podem traslladar a la dimensió econòmica com un estalvi de costos en energia respecte a les altres solucions existents.

12.3 Dimensió social

El desenvolupament d'aquest projecte m'ha aportat en l'àmbit personal coneixements tècnics de la problemàtica de la selecció de variables (un problema que apareix en molts àmbits de la ciència de les dades) especialment en els algorismes de selecció de variables basats en mètodes de filtre i en major mesura del LVF. D'ençà que vaig iniciar-me amb l'especialitat de computació, el camp de les ciències de les dades m'ha interessat molt i poder-me emportar un bon aprenentatge sobre ell amb l'ajuda de tot un professional en el sector com és en Lluís m'ha omplert molt en l'àmbit personal i professional. A part dels coneixements tècnics del domini del projecte, he adquirit un coneixement teòric i pràctic d'aspectes claus en la gestió d'un projecte, ja sigui la seva correcta contextualització, planificació, realització del pressupost, elaboració de l'informe de sostenibilitat, etc. També, he pogut valorar els sacrificis que comporta la realització d'un procés d'investigació i la gran importància que prenen aspectes com l'experimentació i l'anàlisi de resultats.

Aquest projecte no té un impacte social gran, ja que només afecta directament als usuaris de les ciències de les dades que utilitzin les optimitzacions o als investigadors de les ciències de les dades interessats en les optimitzacions o l'estudi. Aquests usuaris i investigadors tindran alternatives noves per a abordar el problema de la selecció de variables.

Realment no existeix una necessitat real del projecte com a tal, ja que existeixen moltes eines per a solucionar aquest problema. Però el fet d'investigar noves vies per a afrontar el problema, pot comportar trobar noves solucions de més qualitat per aquest problema de tanta importància en el món de les ciències de les dades.

Capítol 13

Resultats experimentació

En aquest apèndix, exposarem alguns resultats de l'experimentació desenvolupada durant tot el projecte. Dividirem els resultats en dues seccions diferents; els resultats de les experimentacions referents a les millores inicials deL LVF i els resultats referents a les millores finals del LVF.

13.1 Resultats millores incials del LVF

Aquests resultats són els obtinguts amb l'experimentació amb les millores inicials del LVF amb els conjunts de dades inicials (els conjunts de dades artificials). Aquests resultats són els mencionats en el capítol 6, *Millores inicials del LVF*. Al final de la secció trobem un breu recull d'alguns resultats de l'experimentació classificats per tipologia de variables.

13.1.1 Resultats de l'score

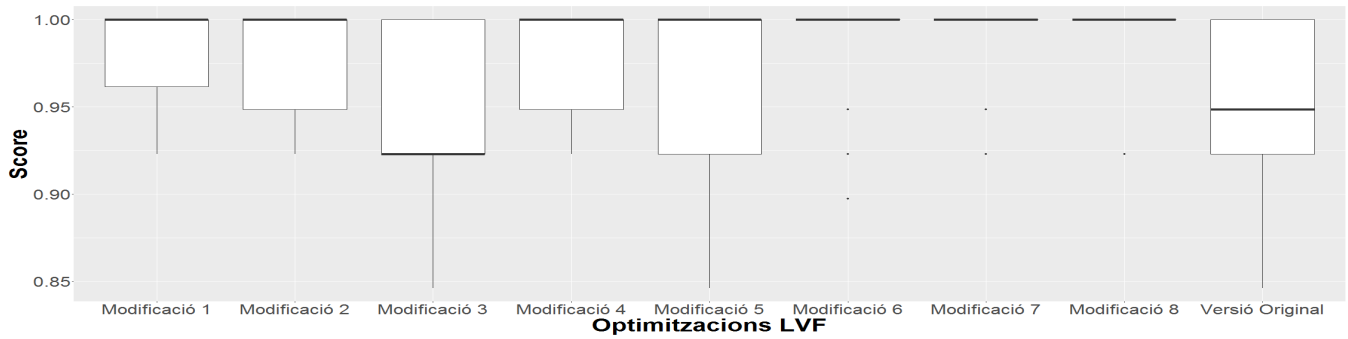


FIGURA 13.1: Score de les millores de la fase inicial a CDL1

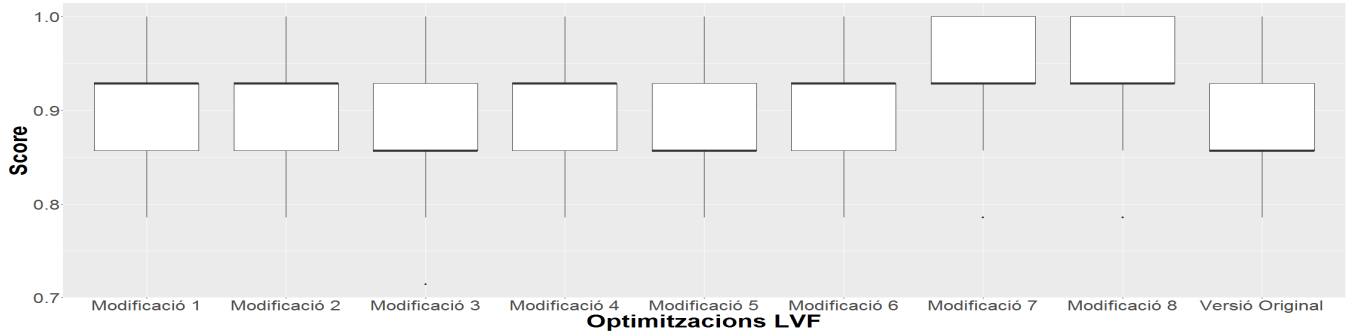


FIGURA 13.2: Score de les millores de la fase inicial a CDL2

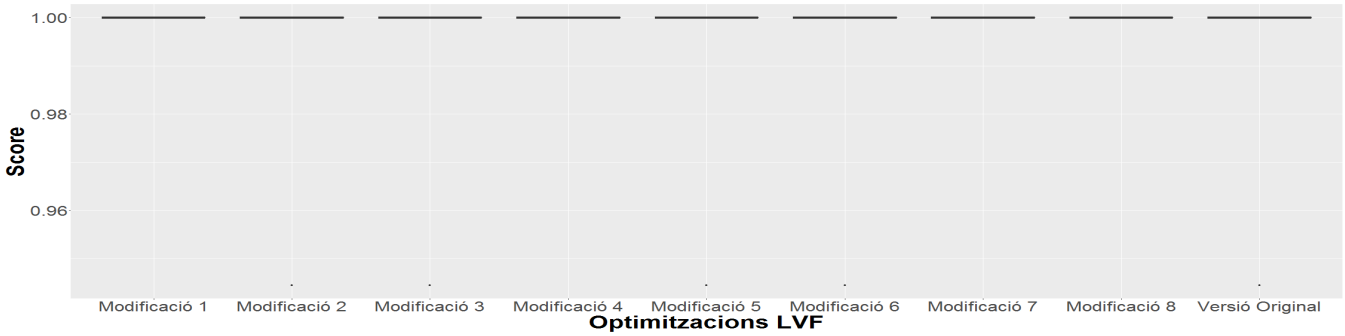


FIGURA 13.3: Score de les millores de la fase inicial a CDL3

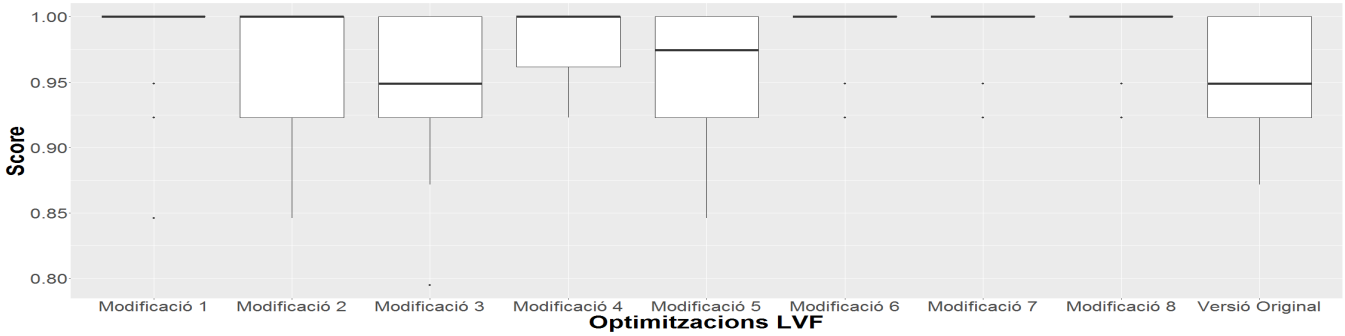


FIGURA 13.4: Score de les millores de la fase inicial a CDE1

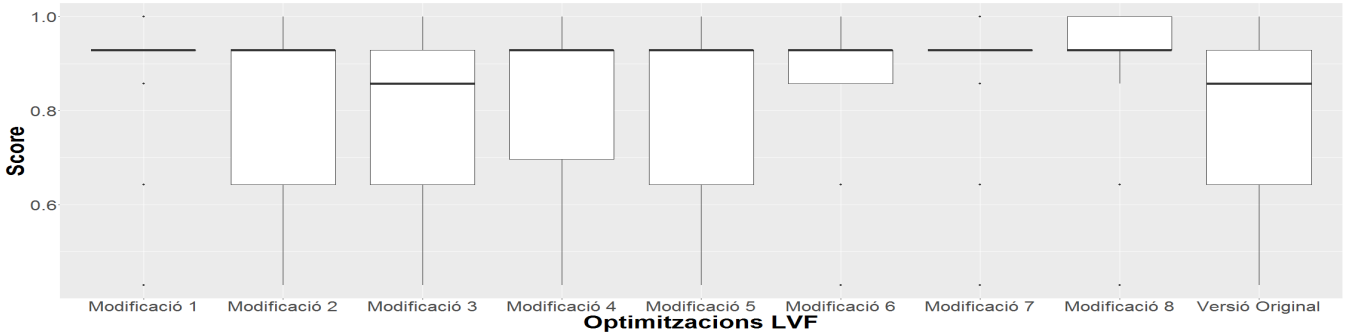


FIGURA 13.5: Score de les millores de la fase inicial a CDE2

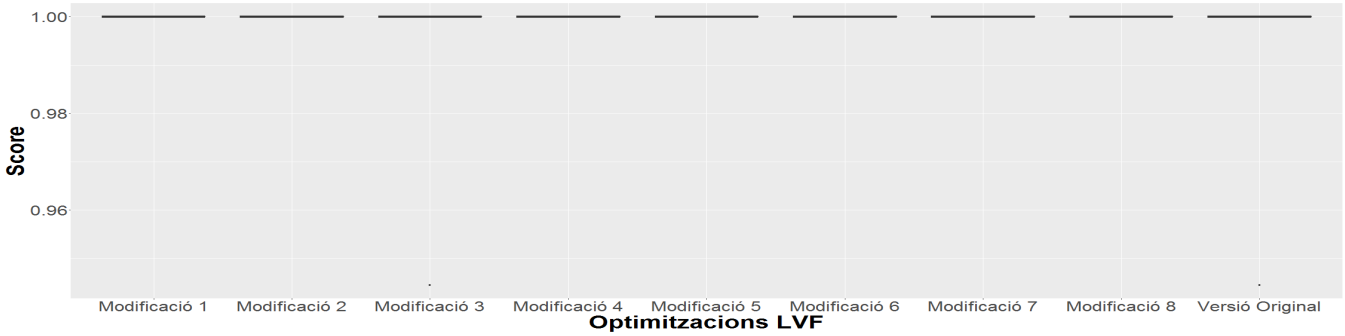


FIGURA 13.6: Score de les millores de la fase inicial a CDE3

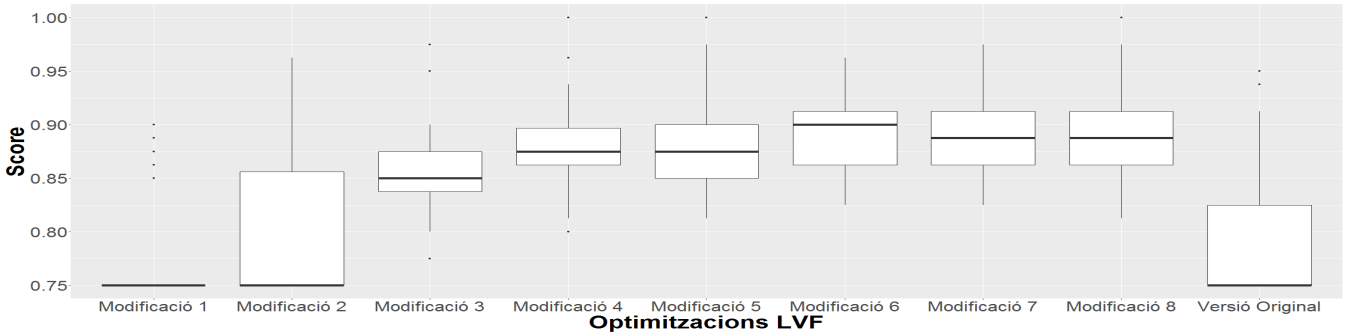


FIGURA 13.7: Score de les millores de la fase inicial a CDP1

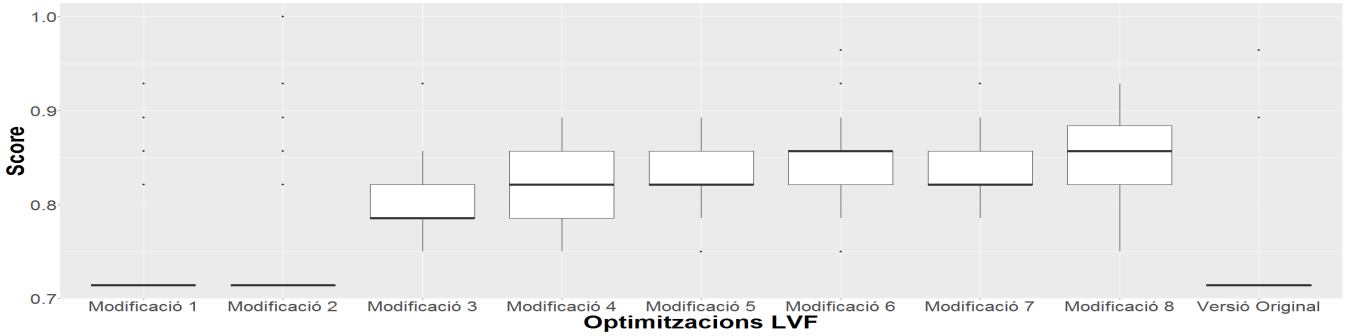


FIGURA 13.8: Score de les millores de la fase inicial a CDP2

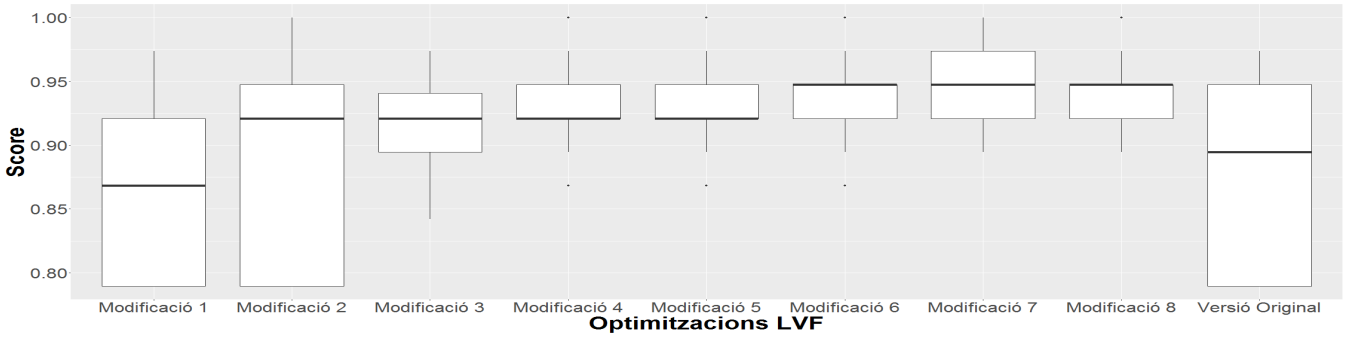


FIGURA 13.9: Score de les millores de la fase inicial a CDP3

13.1.2 Resultats del temps d'execució

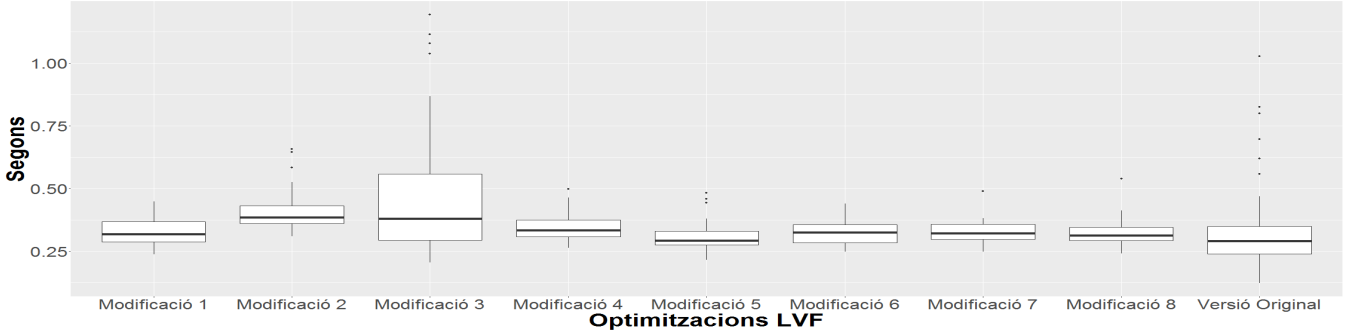


FIGURA 13.10: Temps d'execució de les millores de la fase inicial a CDL1

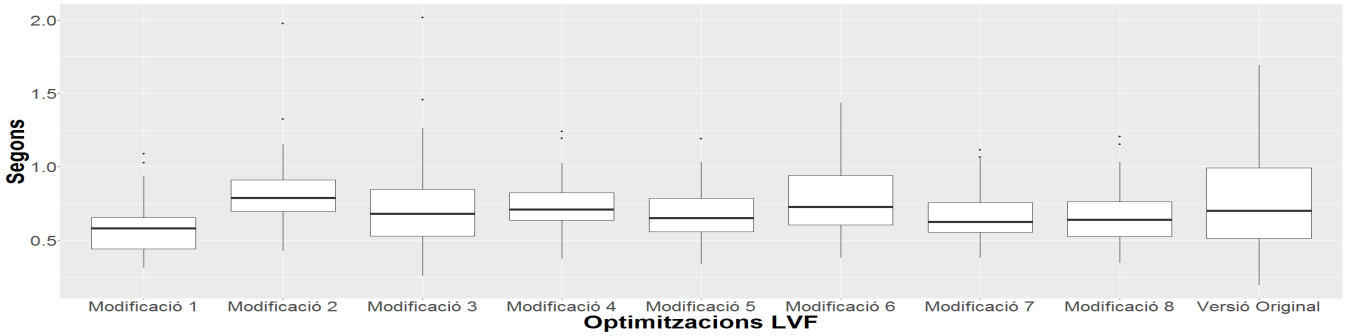


FIGURA 13.11: Temps d'execució de les millores de la fase inicial a CDL2

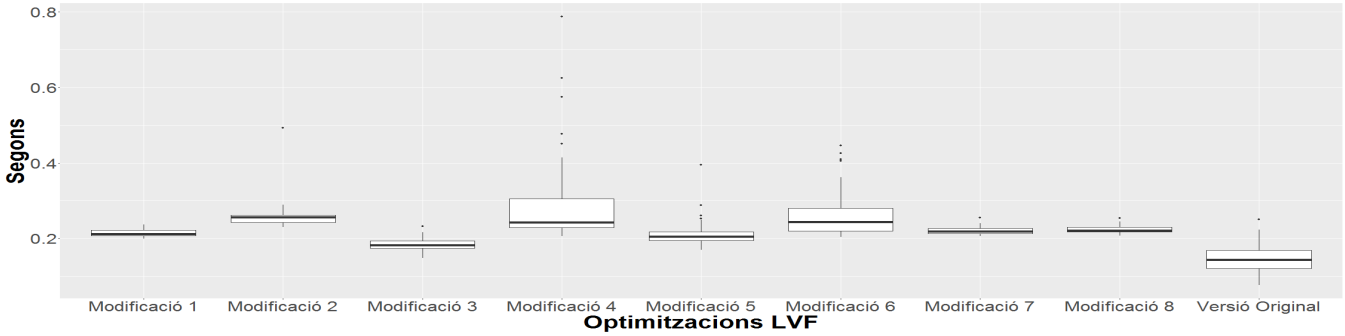


FIGURA 13.12: Temps d'execució de les millores de la fase inicial a CDL3

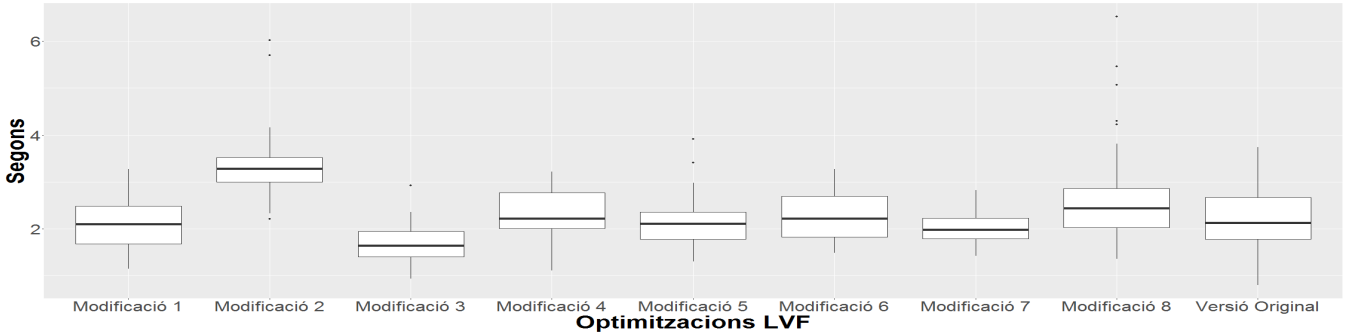


FIGURA 13.13: Temps d'execució de les millores de la fase inicial a CDE1

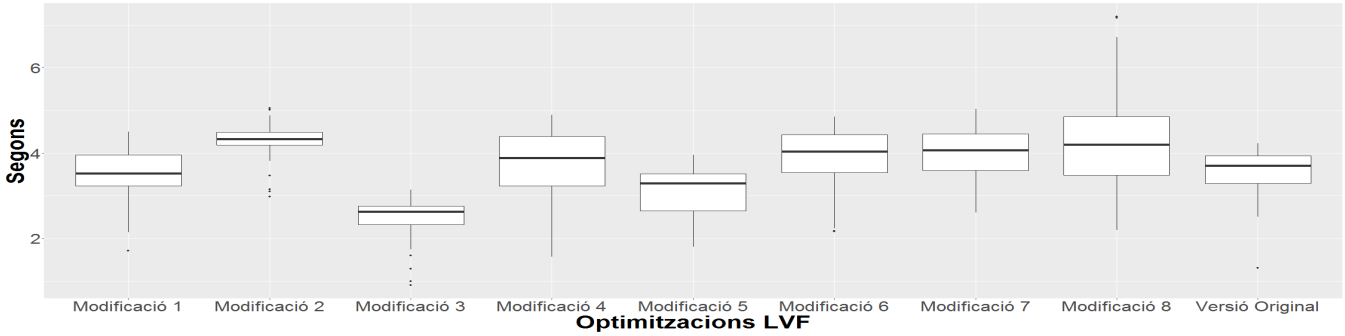


FIGURA 13.14: Temps d'execució de les millores de la fase inicial a CDE2

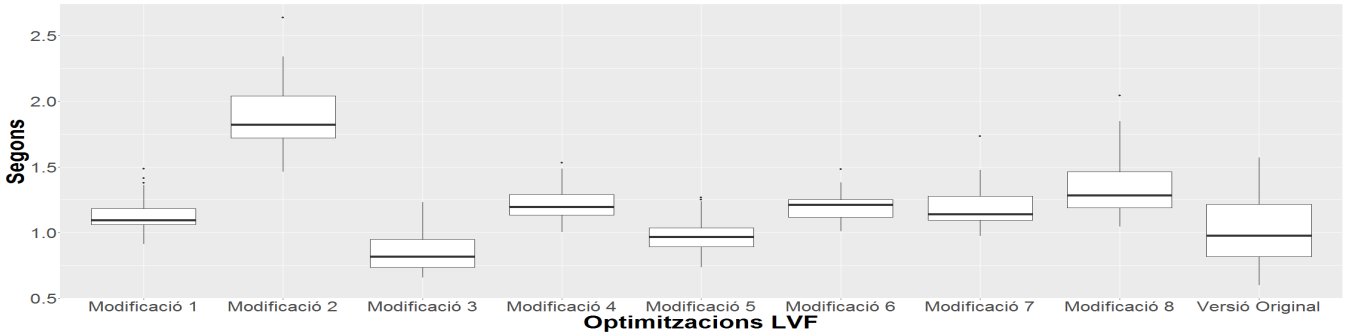


FIGURA 13.15: Temps d'execució de les millores de la fase inicial a CDE3

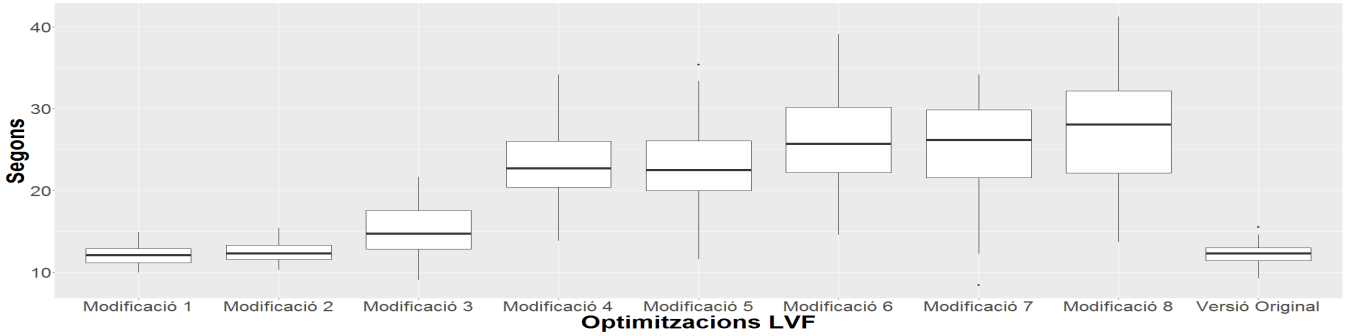


FIGURA 13.16: Temps d'execució de les millores de la fase inicial a CDP1

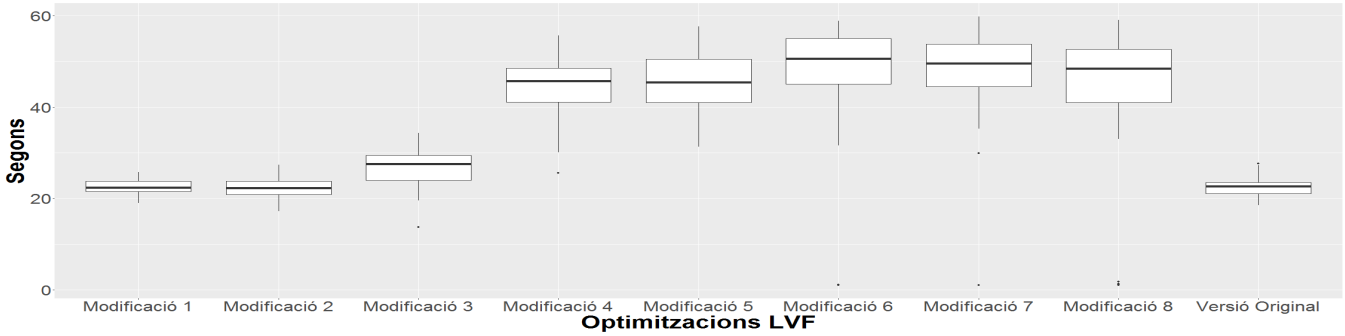


FIGURA 13.17: Temps d'execució de les millores de la fase inicial a CDP2

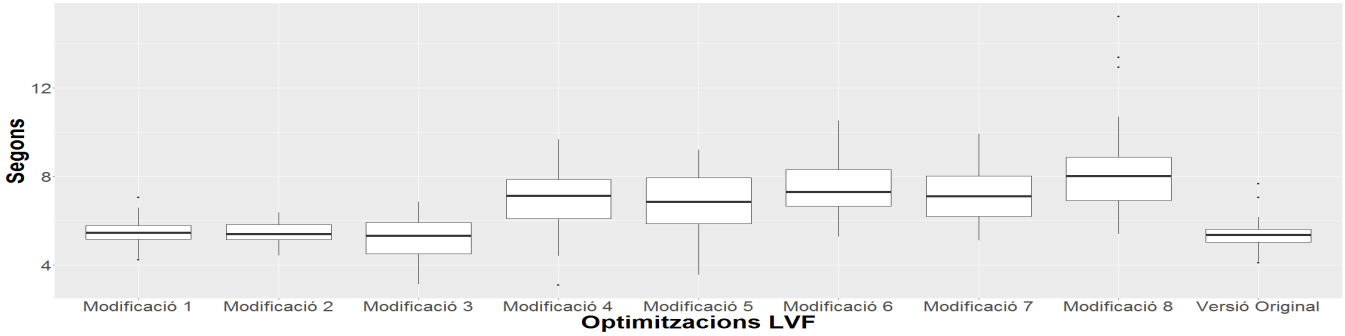


FIGURA 13.18: Temps d'execució de les millores de la fase inicial a CDP3

13.1.3 Resultats de l'*accuracy* amb *Naive Bayes*

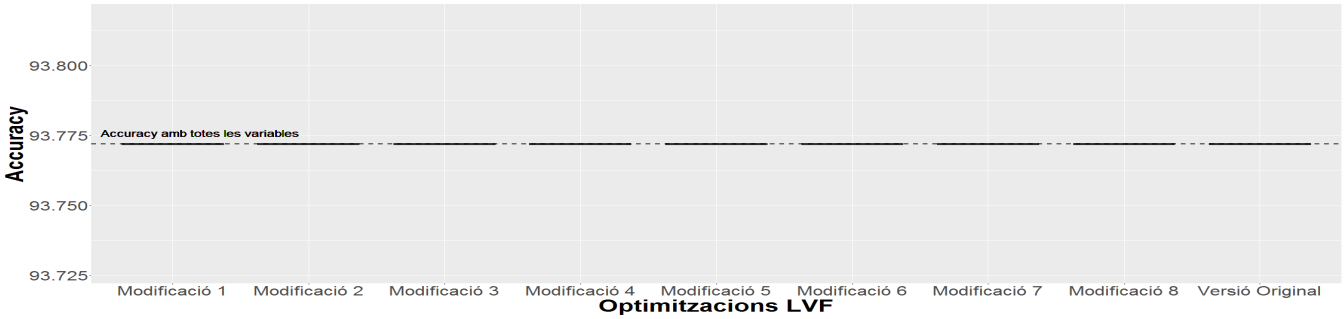


FIGURA 13.19: Accuracy de les millores de la fase inicial a CDL1

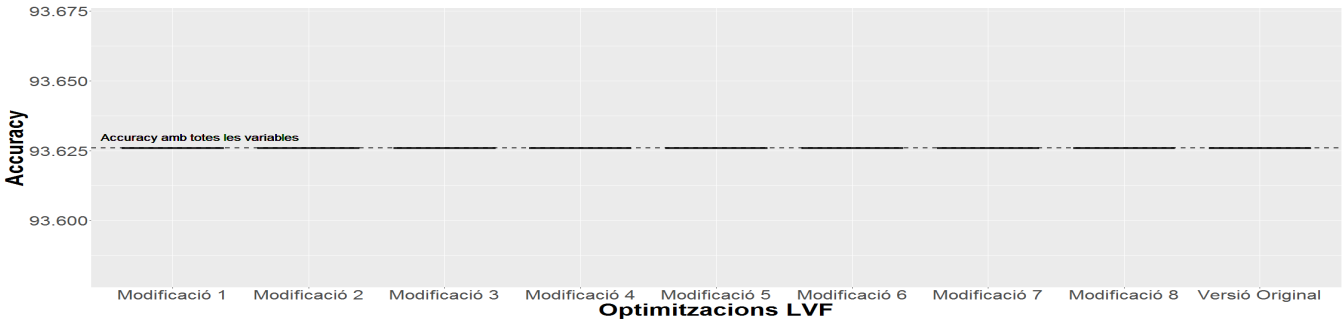


FIGURA 13.20: Accuracy de les millores de la fase inicial a CDL2

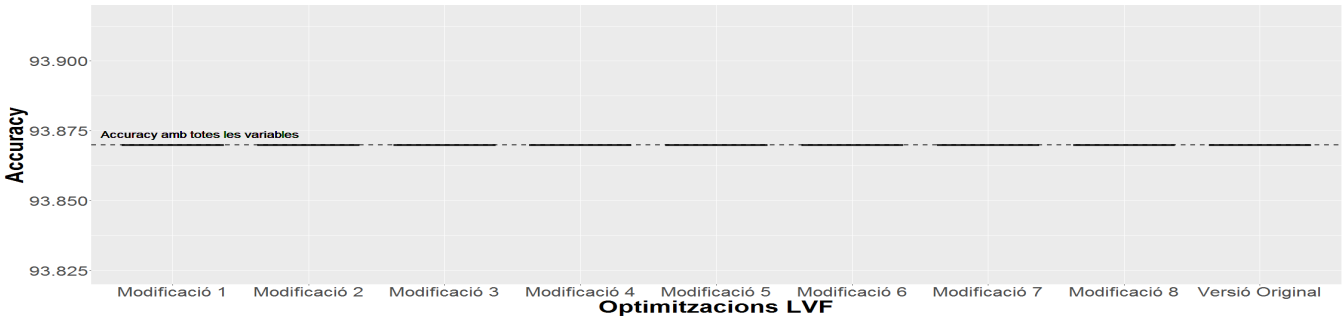


FIGURA 13.21: Accuracy de les millores de la fase inicial a CDL3

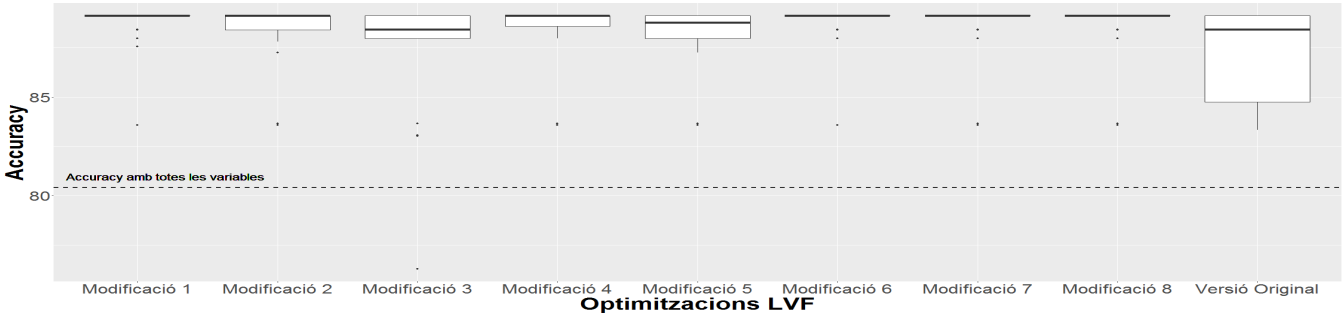


FIGURA 13.22: Accuracy de les millores de la fase inicial a CDE1

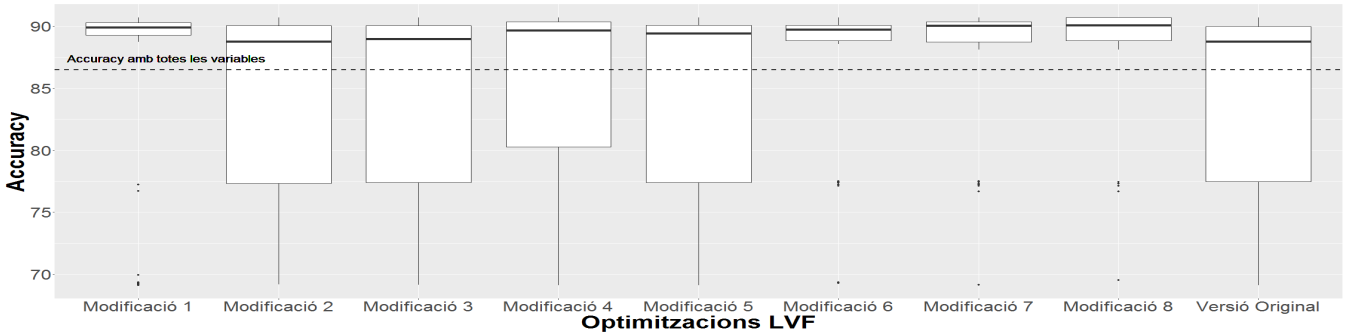


FIGURA 13.23: Accuracy de les millores de la fase inicial a CDE2

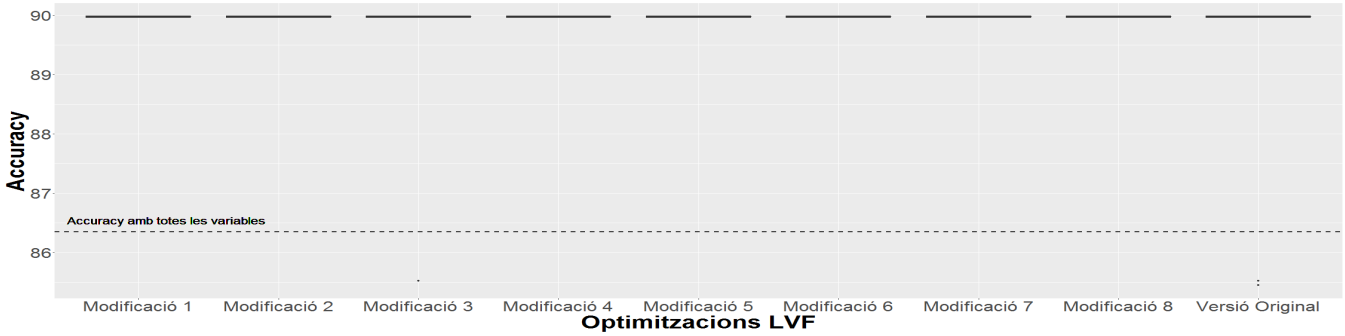


FIGURA 13.24: Accuracy de les millores de la fase inicial a CDE3

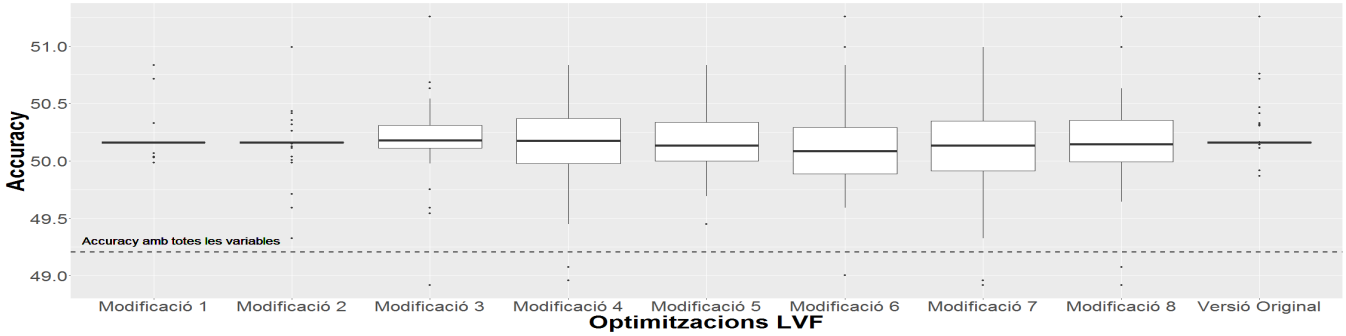


FIGURA 13.25: Accuracy de les millores de la fase inicial a CDP1

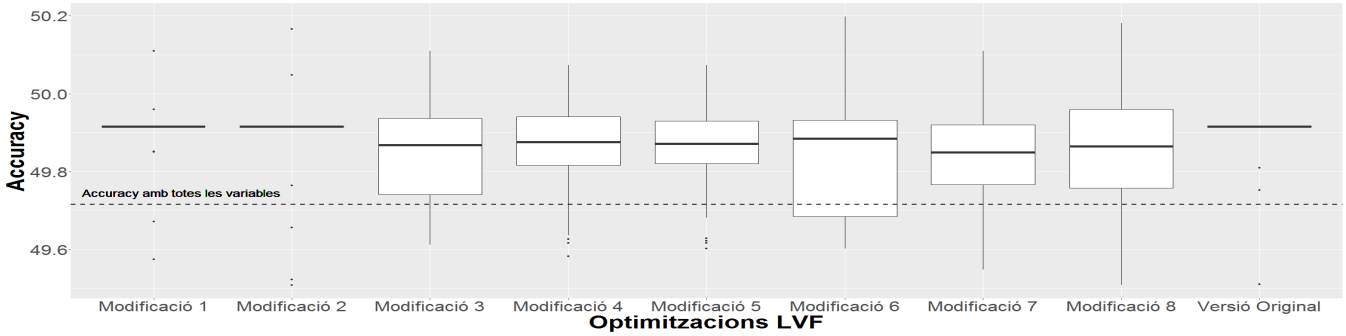


FIGURA 13.26: Accuracy de les millores de la fase inicial a CDP2

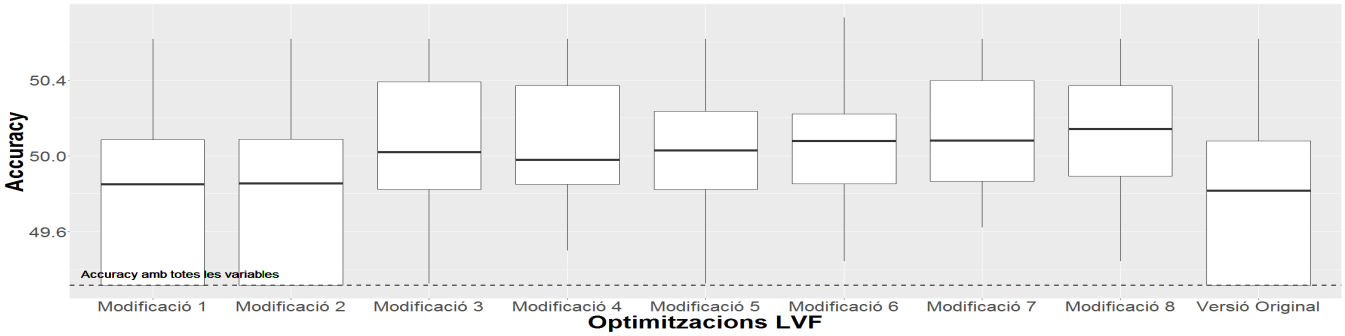
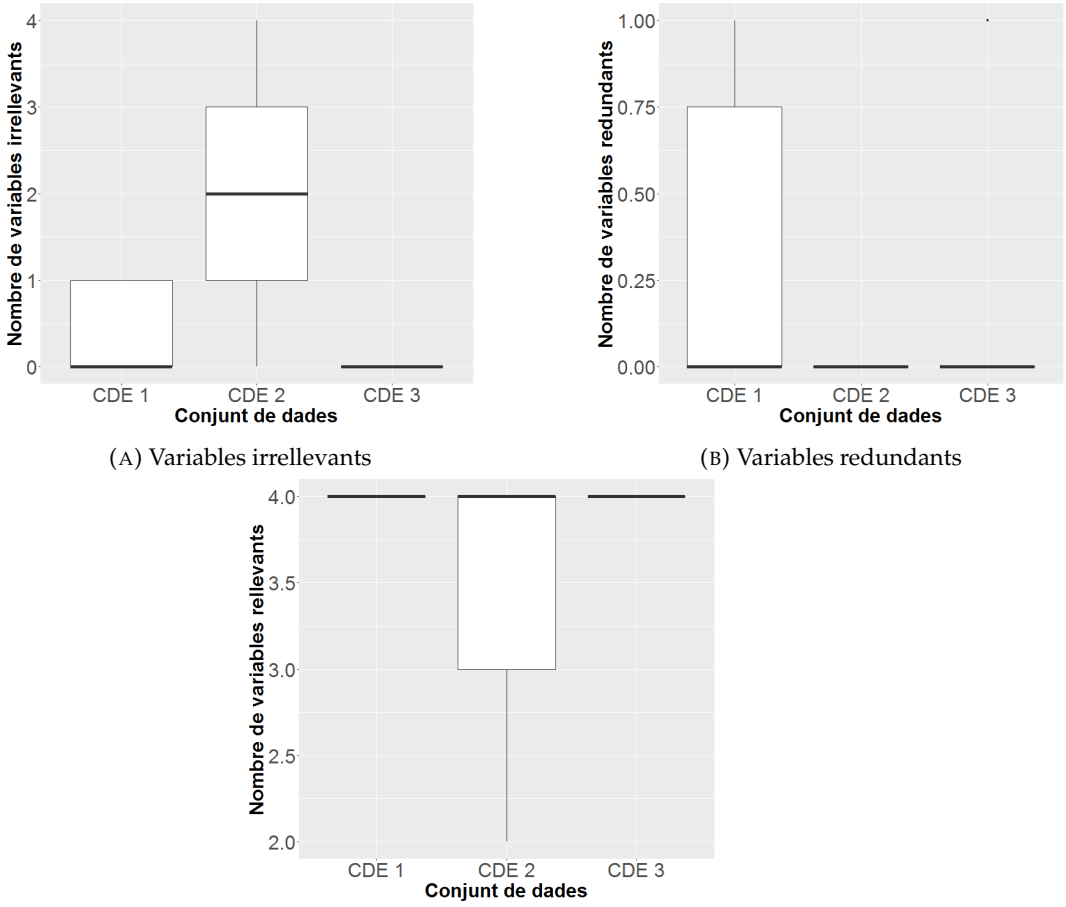


FIGURA 13.27: Accuracy de les millores de la fase inicial a CDP3

13.2 Resultats per tipologia de variables

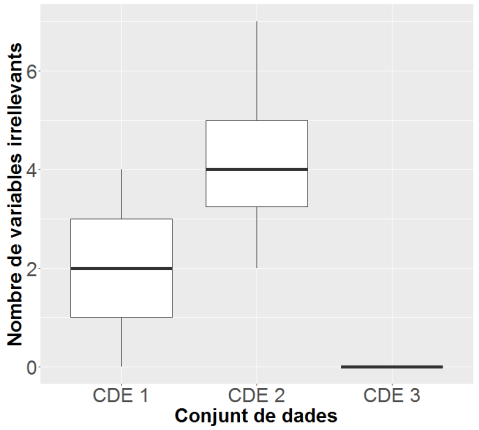


(A) Variables irrellevants

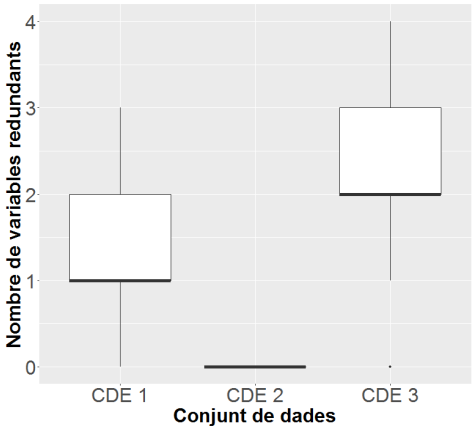
(B) Variables redundants

(C) Variables rellevants

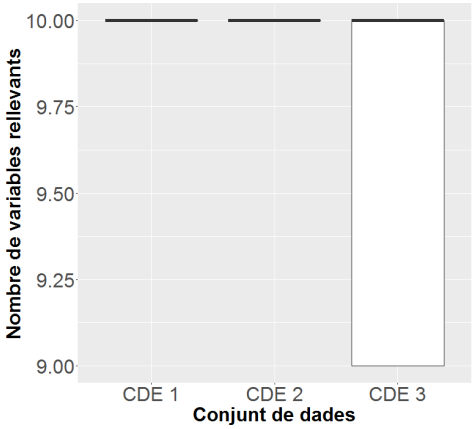
FIGURA 13.28: Nombre de variables seleccionades segons tipologia amb la versió original del LVF a CDE



(A) Variables irrellevants



(B) Variables redundants



(C) Variables rellevants

FIGURA 13.29: Nombre de variables seleccionades segons tipologia amb la modificació 8 del LVF a CDP

13.3 Resultats de les millores finals del LVF basades en mètodes de filtre

En aquesta secció exposarem alguns dels resultats obtinguts amb l'experimentació de les millores finals del LVF que es basen en mètodes de filtre. Aquestes experimentacions s'han realitzat amb els conjunts de dades finals (conjunts de dades reals), exceptuant l'experimentació realitzada pels paràmetres *alpha* i *beta* del LVA. Aquests resultats es mencionen en el capítol 7, *Millores finals del LVF*.

13.3.1 Resultats de l'experimentació amb LVA i els seus paràmetres *alpha/beta*

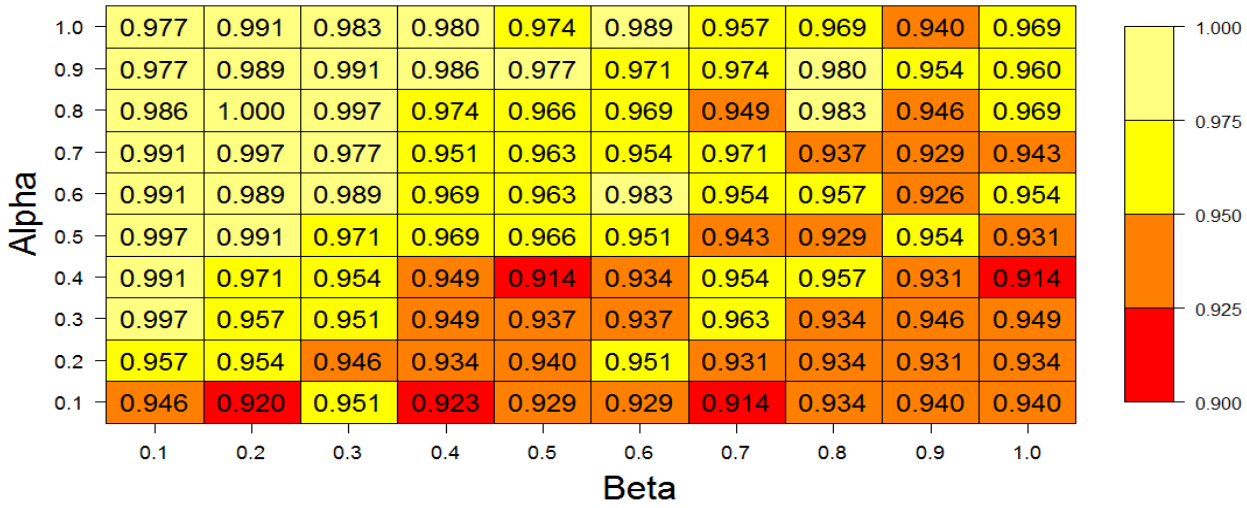


FIGURA 13.30: *Score* mitjà de l'experimentació amb LVA i *alpha*/*beta* a CDL2

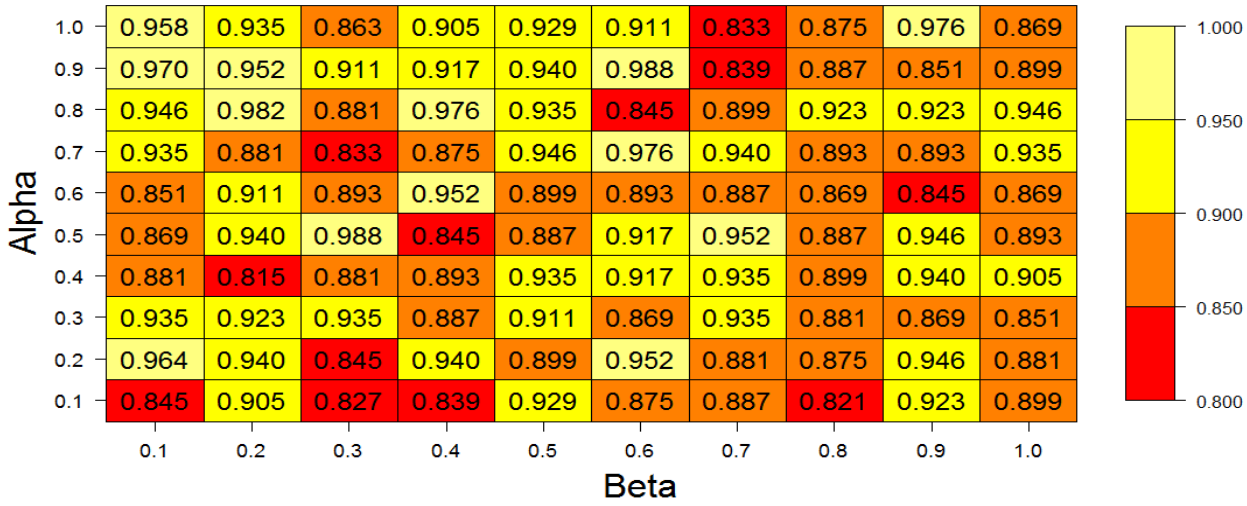


FIGURA 13.31: *Score* mitjà de l'experimentació amb LVA i *alpha*/*beta* a CDE2

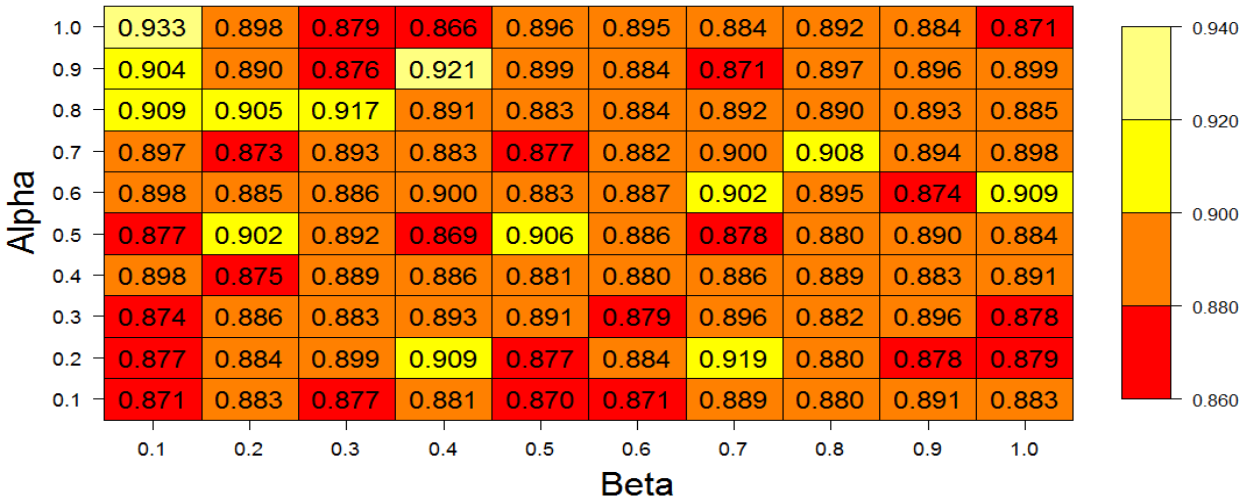


FIGURA 13.32: *Score* mitjà de l'experimentació amb LVA i *alpha*/*beta* a CDP1

13.3.2 Resultats del nombre de variables seleccionades

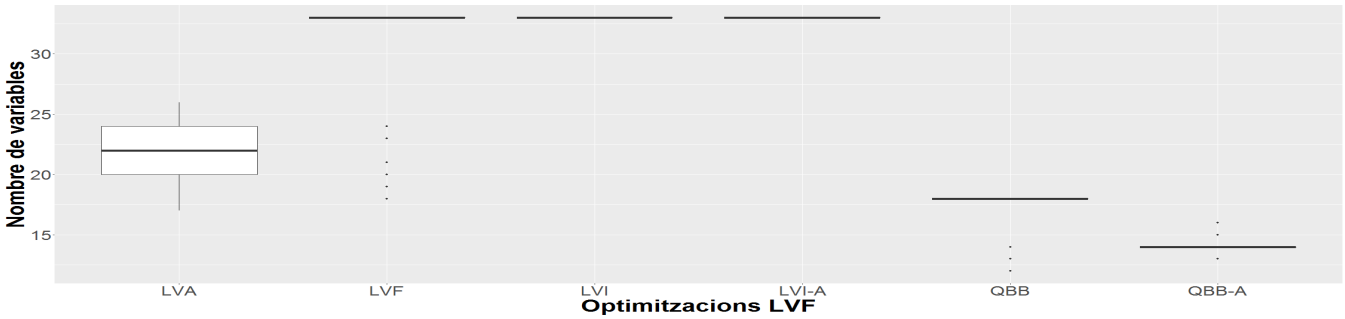


FIGURA 13.33: Nombre de variables seleccionades en les optimitzacions basades en mètodes de filtre a CDI

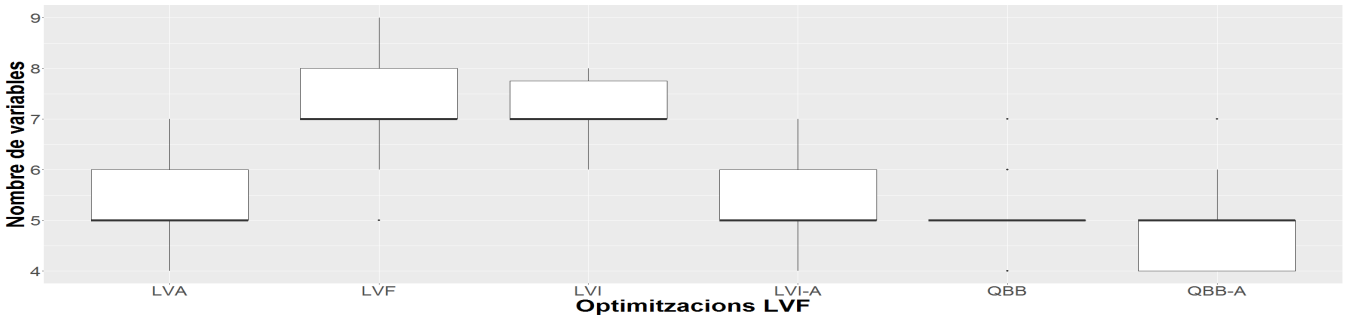


FIGURA 13.34: Nombre de variables seleccionades en les optimitzacions basades en mètodes de filtre a CDM

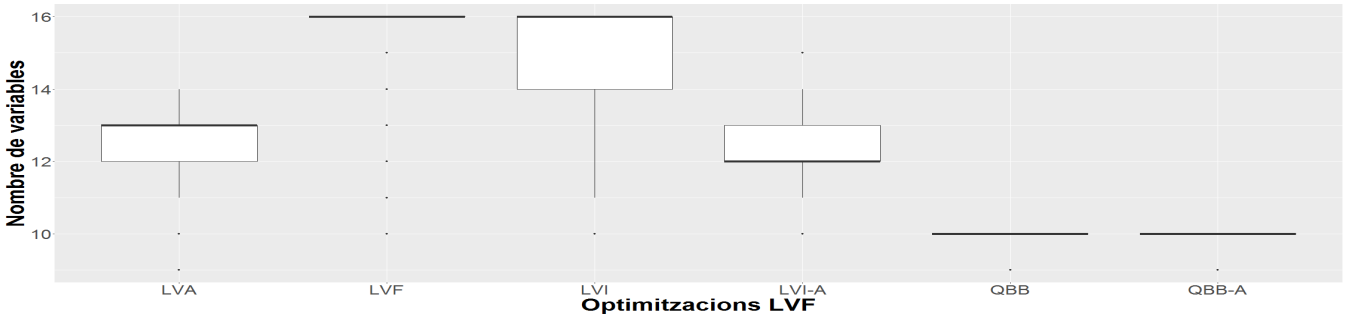


FIGURA 13.35: Nombre de variables seleccionades en les optimitzacions basades en mètodes de filtre amb a CDV

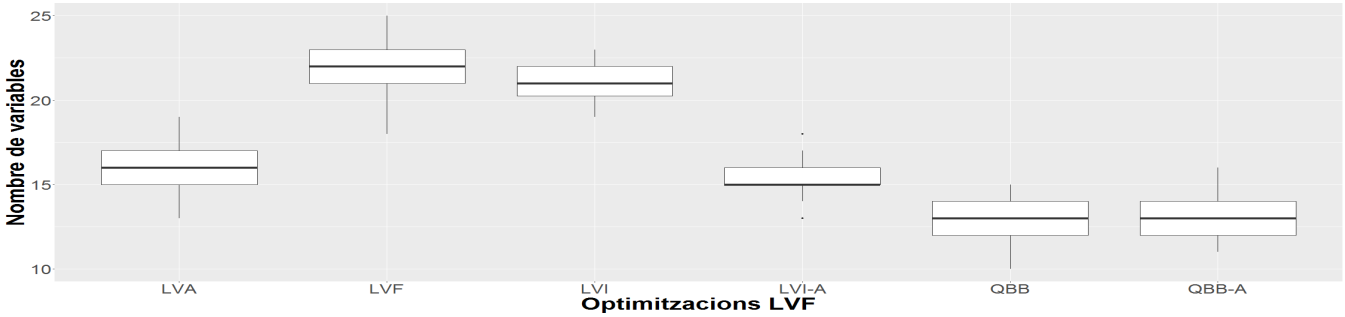


FIGURA 13.36: Nombre de variables seleccionades en les optimitzacions basades en mètodes de filtre a CDC

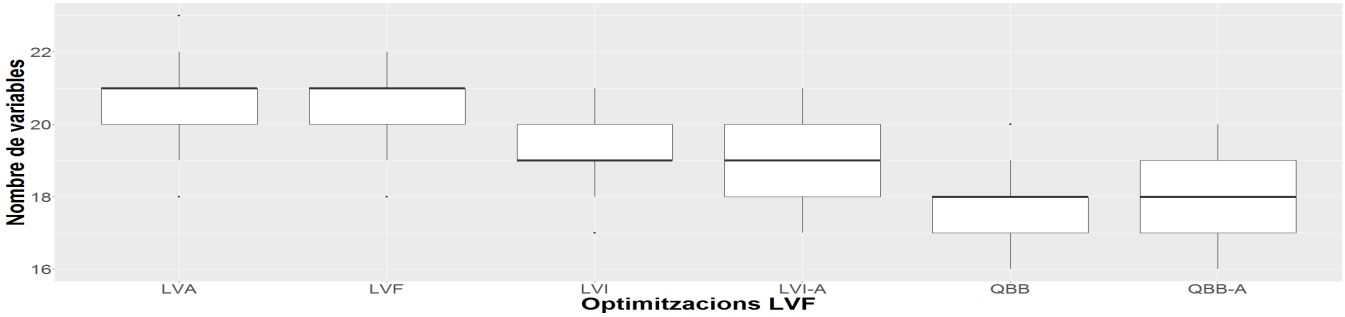


FIGURA 13.37: Nombre de variables seleccionades en les optimitzacions basades en mètodes de filtre a CDW

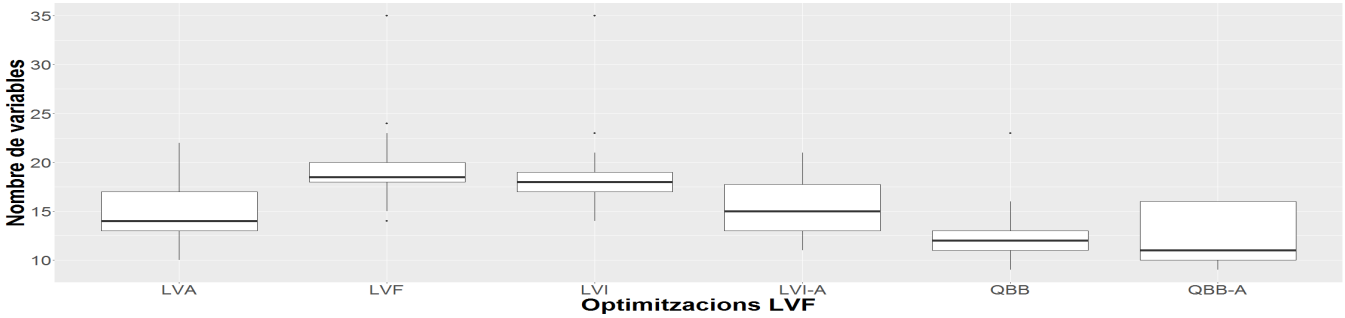


FIGURA 13.38: Nombre de variables seleccionades en les optimitzacions basades en mètodes de filtre a CDB

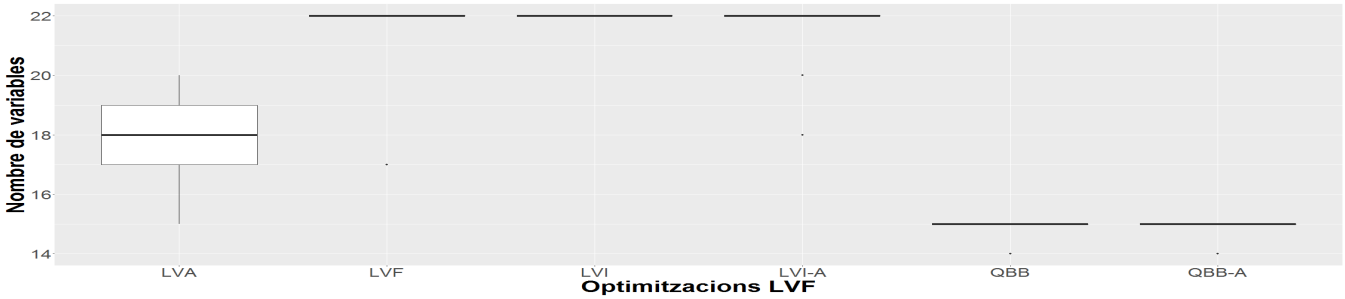


FIGURA 13.39: Nombre de variables seleccionades en les optimitzacions basades en mètodes de filtre a CDH

13.3.3 Resultats del temps d'execució

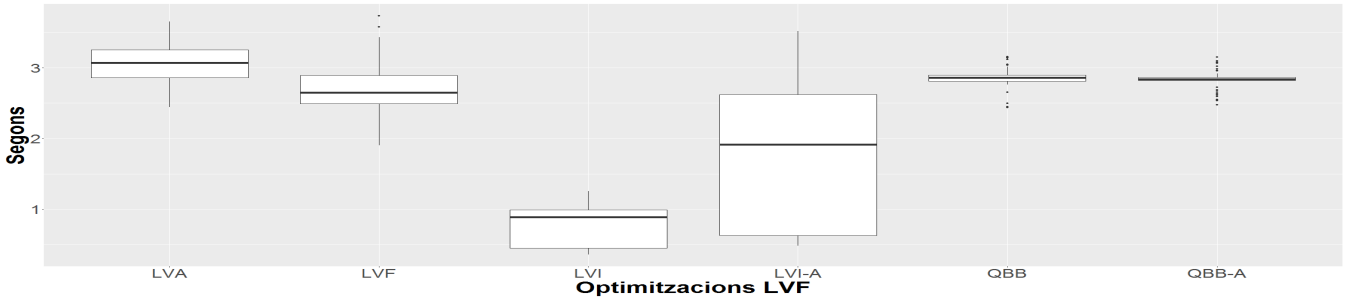


FIGURA 13.40: Temps d'execució en les optimitzacions basades en mètodes de filtre a CDI

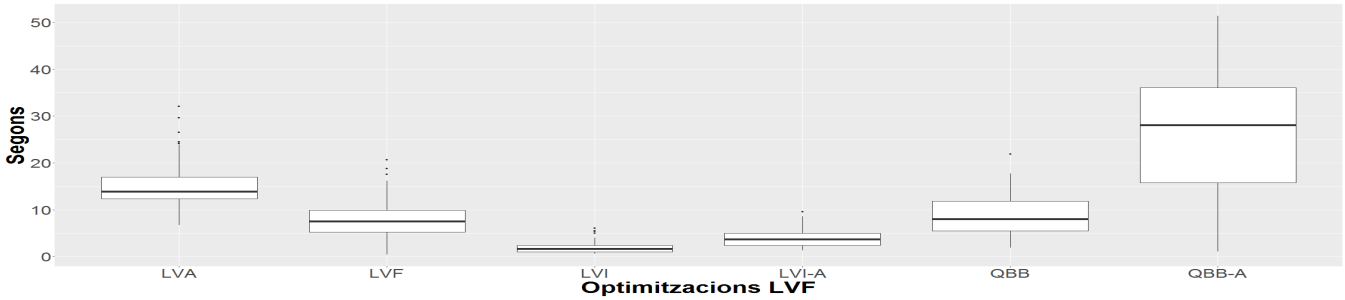


FIGURA 13.41: Temps d'execució en les optimitzacions basades en mètodes de filtre a CDM

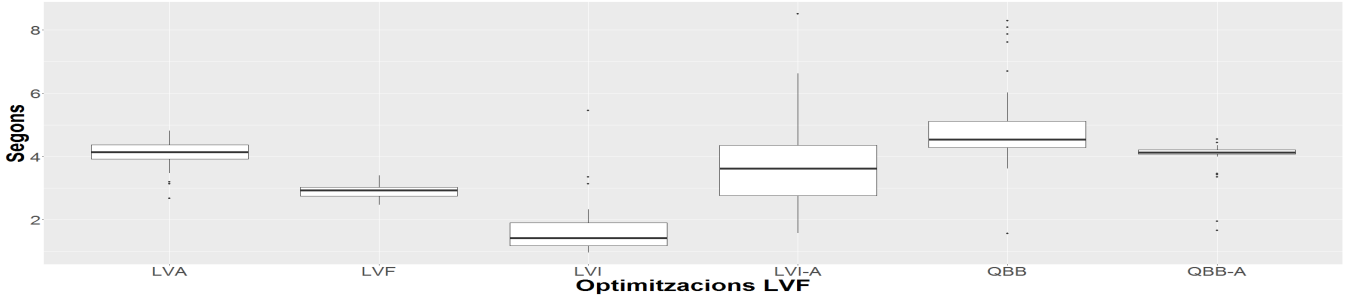


FIGURA 13.42: Temps d'execució en les optimitzacions basades en mètodes de filtre amb a CDV

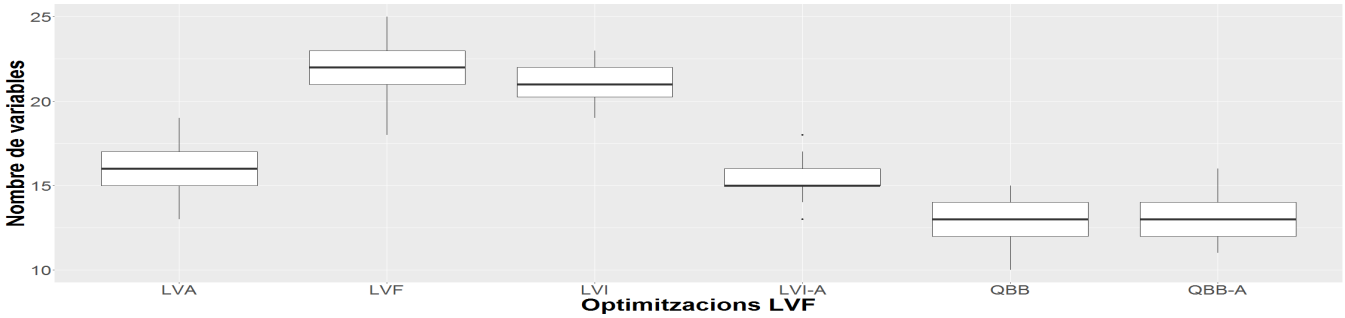


FIGURA 13.43: Temps d'execució en les optimitzacions basades en mètodes de filtre a CDC

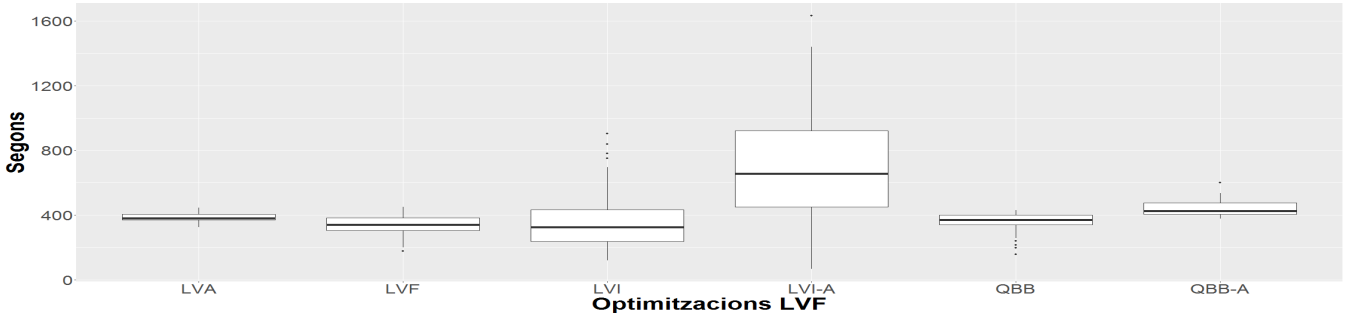


FIGURA 13.44: Temps d'execució en les optimitzacions basades en mètodes de filtre a CDW

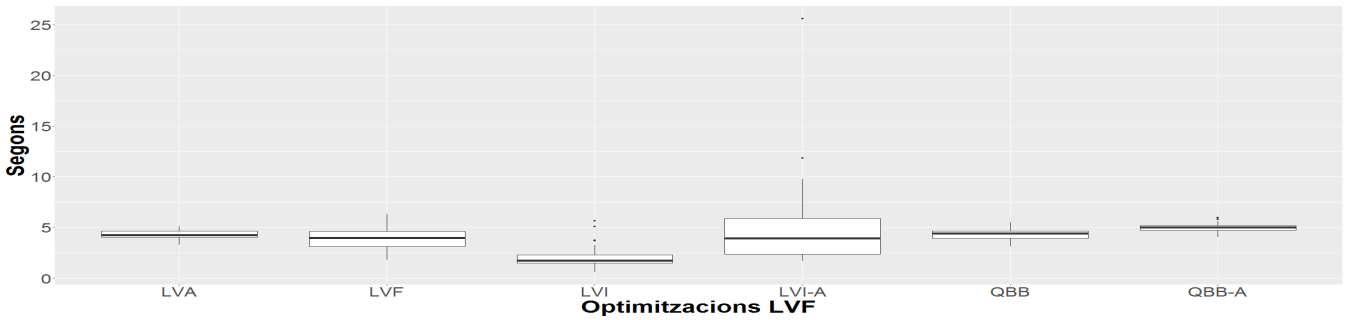


FIGURA 13.45: Temps d'execució en les optimitzacions basades en mètodes de filtre a CDB

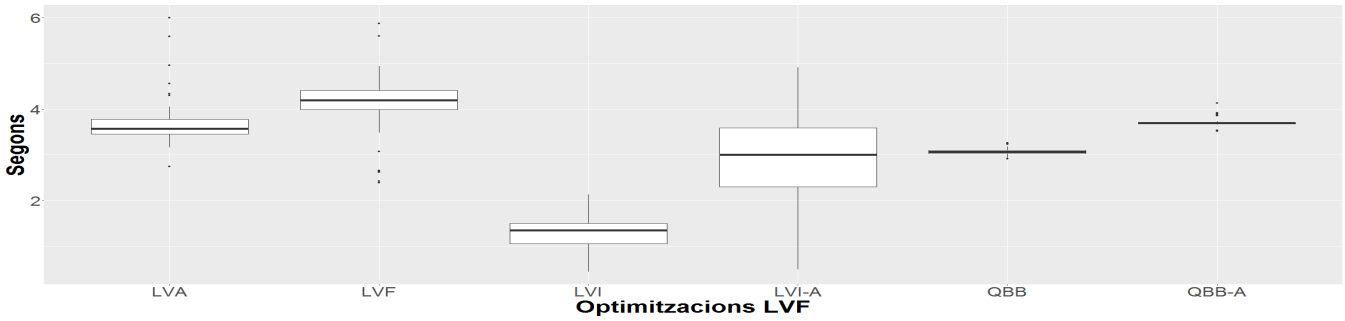


FIGURA 13.46: Temps d'execució en les optimitzacions basades en mètodes de filtre a CDH

13.3.4 Resultats de l'*accuracy* amb *Naive Bayes*

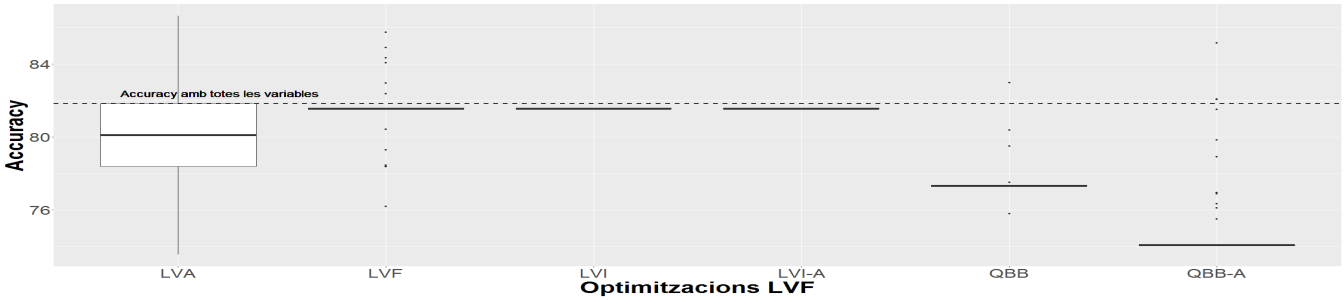


FIGURA 13.47: Accuracy de les optimitzacions basades en mètodes de filtre amb *Niave Bayes* a CDI

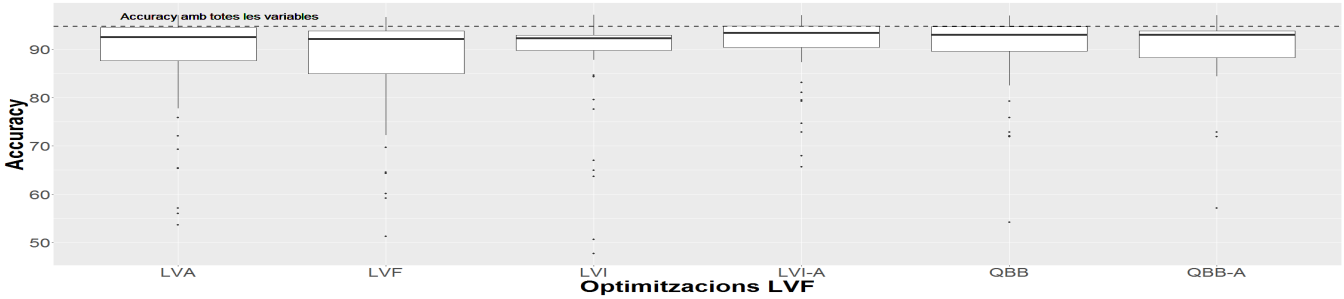


FIGURA 13.48: Accuracy de les optimitzacions basades en mètodes de filtre amb *Niave Bayes* a CDM

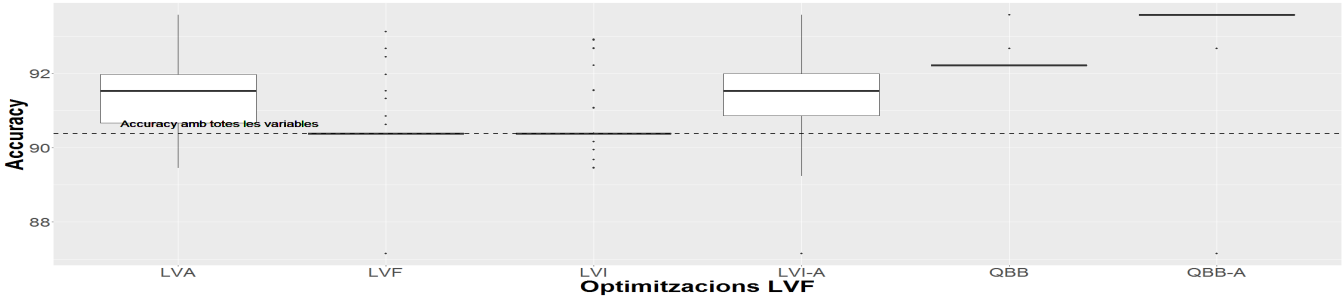


FIGURA 13.49: Accuracy de les optimitzacions basades en mètodes de filtre amb *Niave Bayes* a CDV

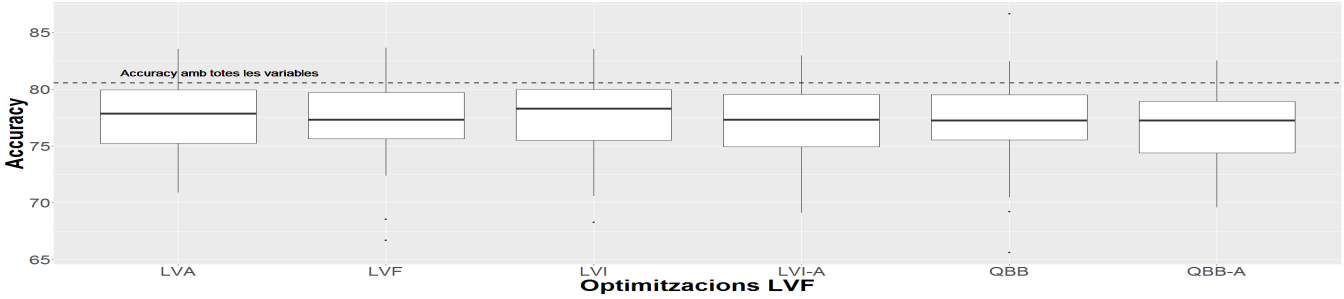


FIGURA 13.50: Accuracy de les optimitzacions basades en mètodes de filtre amb *Niave Bayes* a CDC

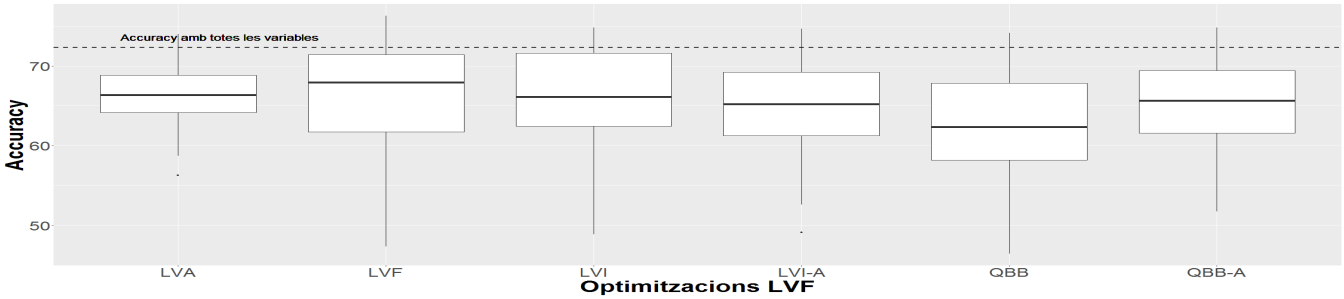


FIGURA 13.51: Accuracy de les optimitzacions basades en mètodes de filtre amb *Niave Bayes* a CDW

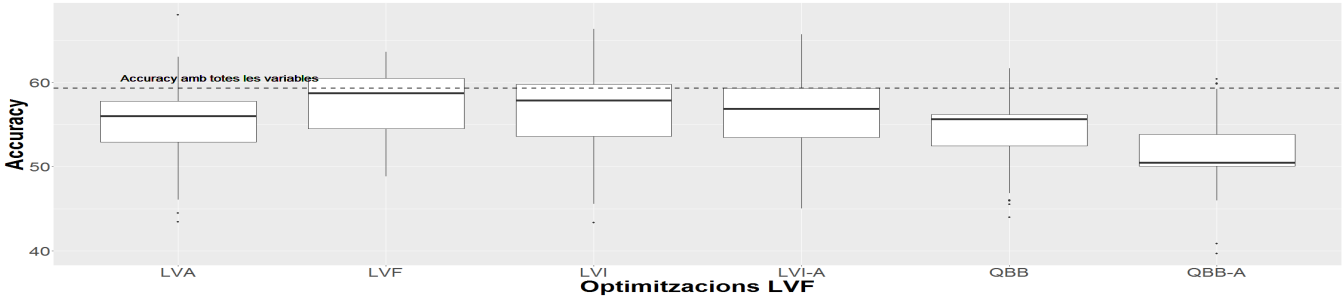


FIGURA 13.52: Accuracy de les optimitzacions basades en mètodes de filtre amb *Niave Bayes* a CDB

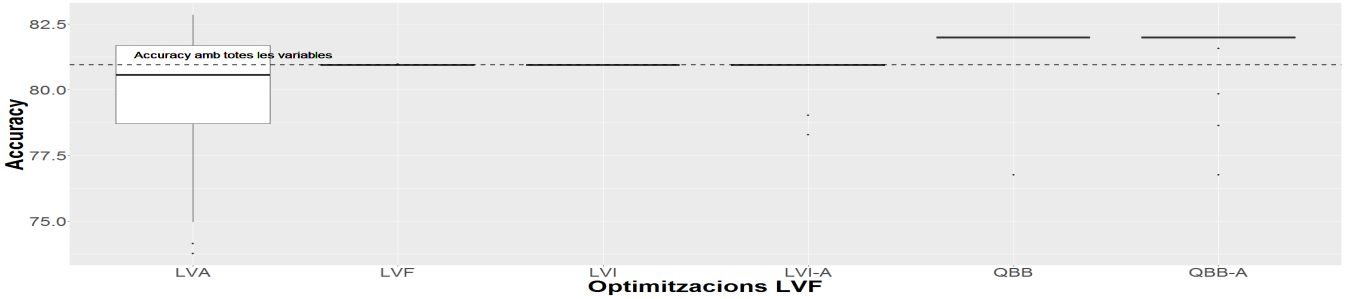


FIGURA 13.53: Accuracy de les optimitzacions basades en mètodes de filtre amb *Niave Bayes* a CDH

13.3.5 Resultats de l'*accuracy* amb arbres de decisió

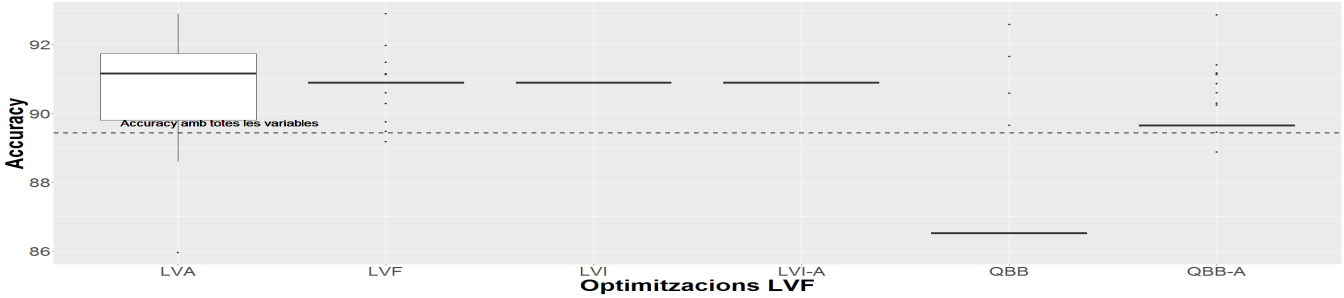


FIGURA 13.54: Accuracy de les optimitzacions basades en mètodes de filtre amb arbres de decisions a CDI

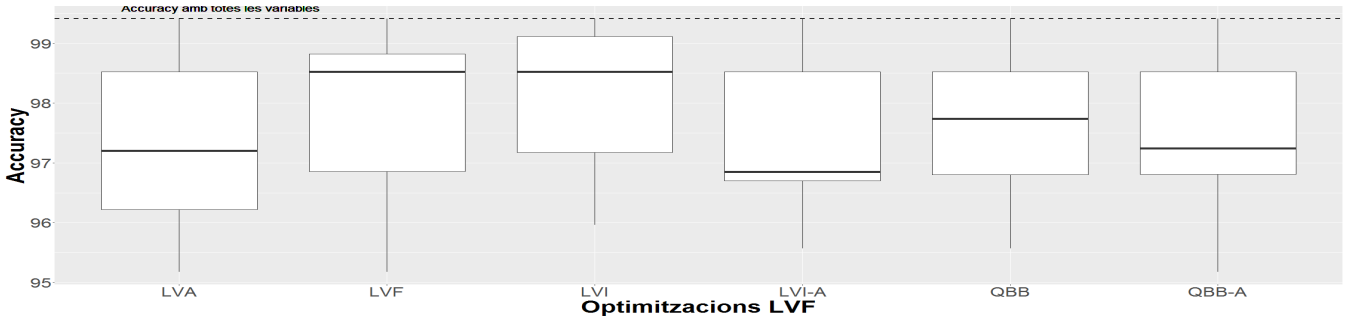


FIGURA 13.55: Accuracy de les optimitzacions basades en mètodes de filtre amb arbres de decisions a CDM

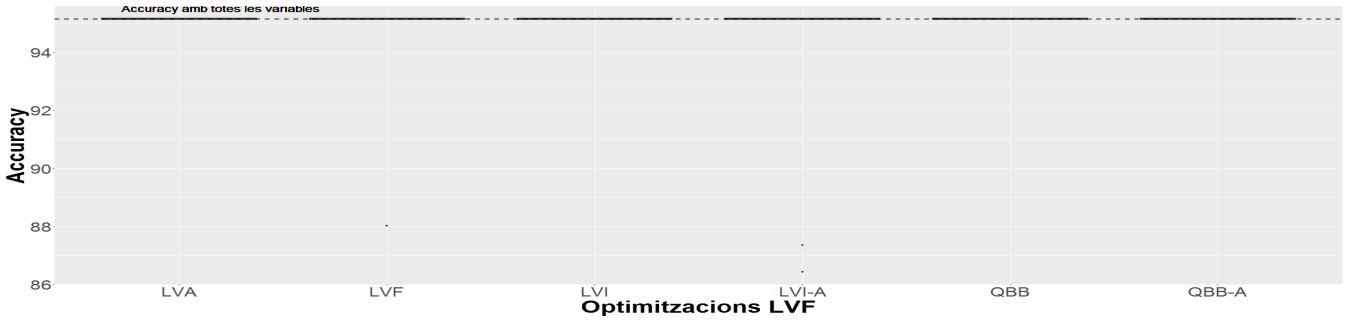


FIGURA 13.56: Accuracy de les optimitzacions basades en mètodes de filtre amb arbres de decisions a CDV

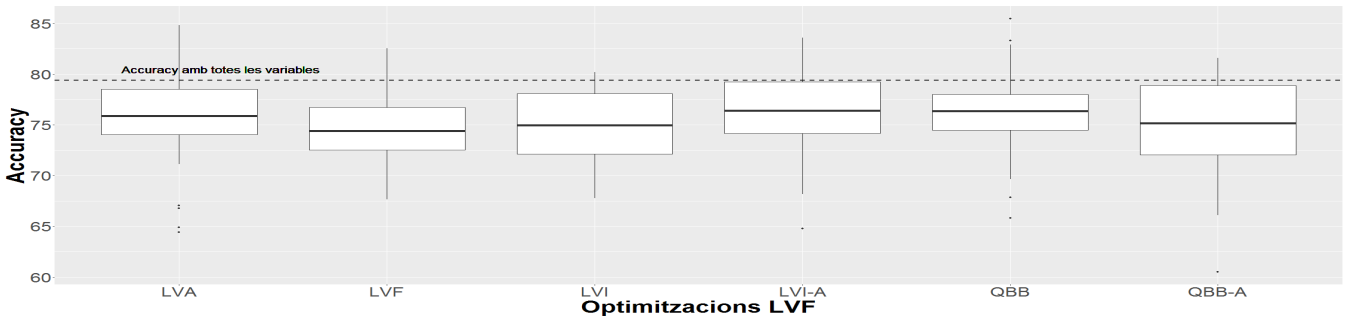


FIGURA 13.57: Accuracy de les optimitzacions basades en mètodes de filtre amb arbres de decisions a CDC

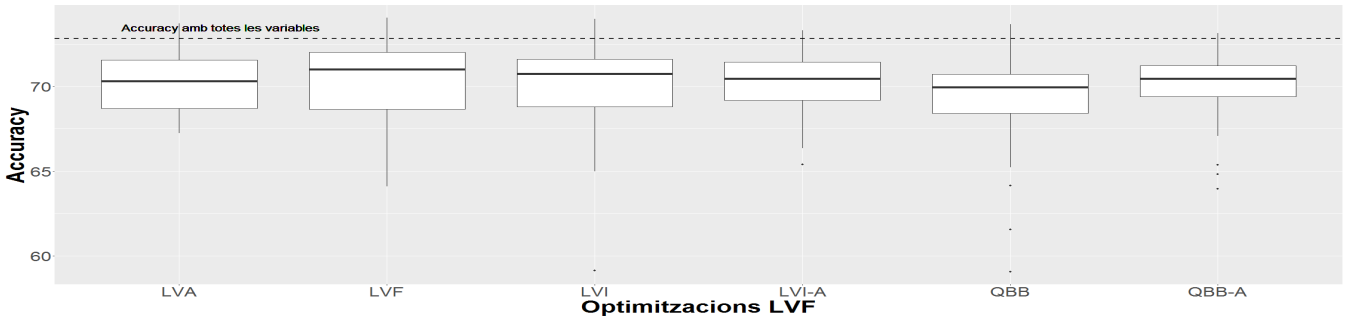


FIGURA 13.58: Accuracy de les optimitzacions basades en mètodes de filtre amb arbres de decisions a CDW

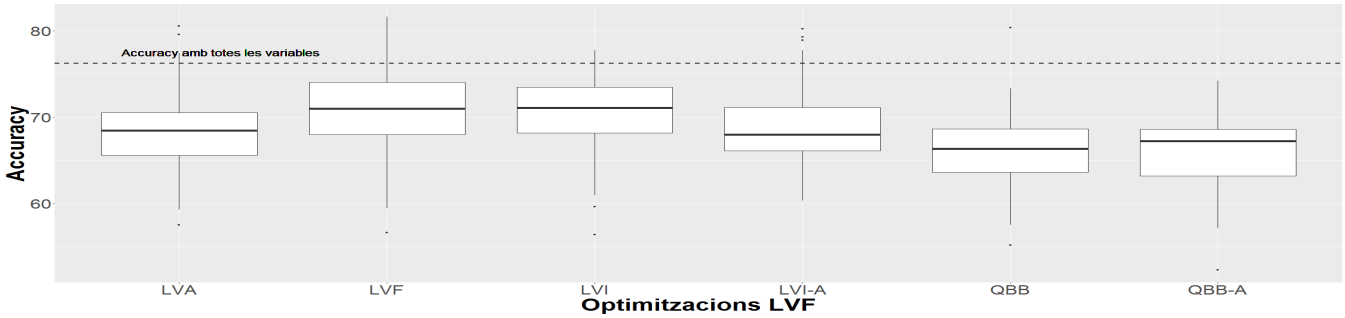


FIGURA 13.59: Accuracy de les optimitzacions basades en mètodes de filtre amb arbres de decisions a CDB

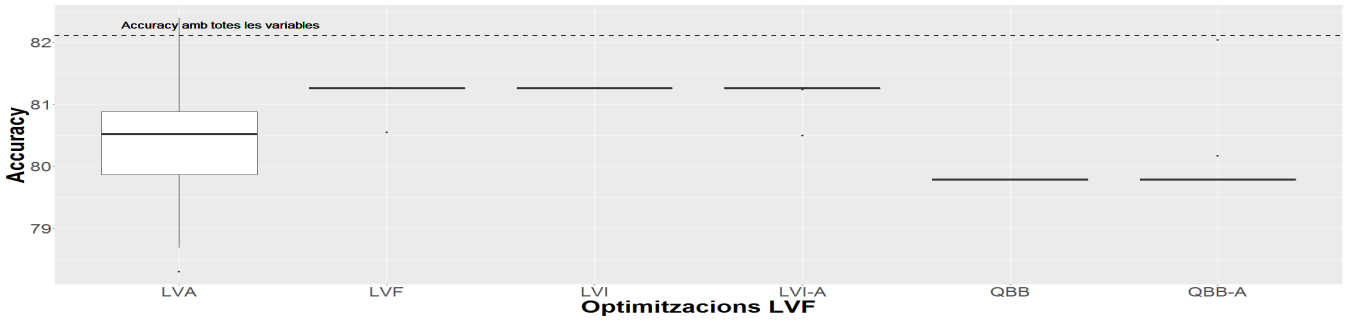


FIGURA 13.60: Accuracy de les optimitzacions basades en mètodes de filtre amb arbres de decisions a CDH

13.4 Resultats de les millores finals del LVF basades en mètodes híbrids i d'embolcall

En aquesta secció exposarem els resultats obtinguts amb l'experimentació de les millores finals del LVF que es basen en mètodes híbrids i d'embolcall. Aquestes experimentacions s'han realitzat amb els conjunts de dades finals. Aquests resultats es mencionen en el capítol 7, *Millores finals del LVF*.

13.4.1 Resultats del nombre de variables seleccionades

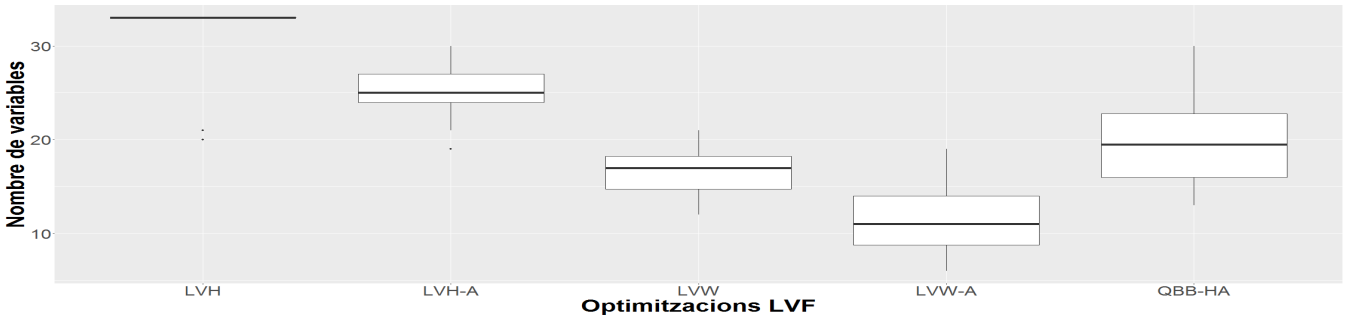


FIGURA 13.61: Nombre de variables seleccionades en les optimitzacions basades en mètodes híbrids i d'embolcall a CDI

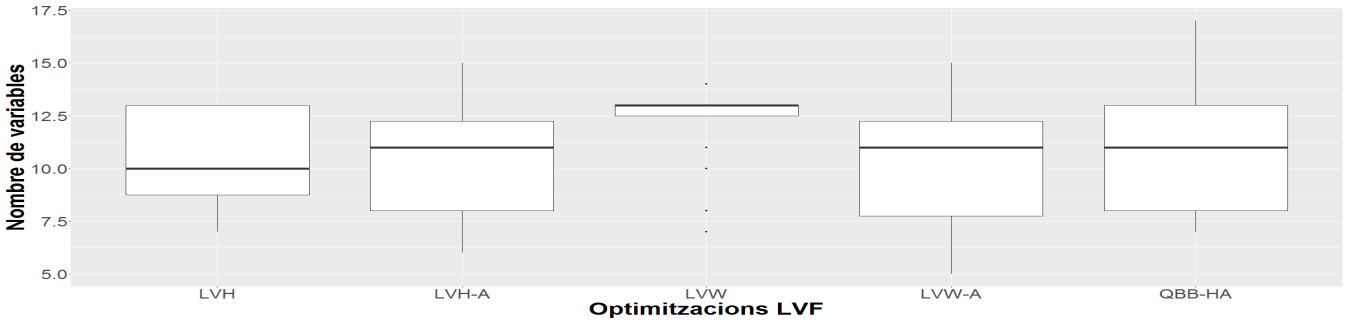


FIGURA 13.62: Nombre de variables seleccionades en les optimitzacions basades en mètodes híbrids i d'embolcall a CDM

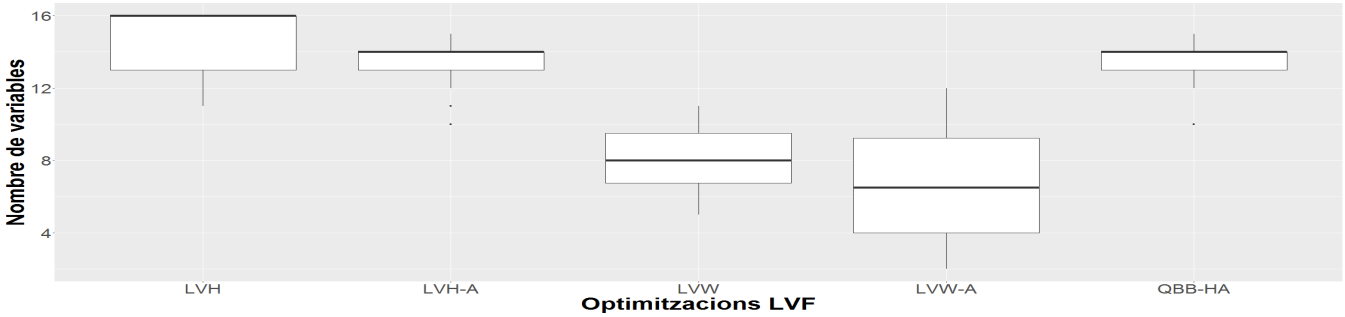


FIGURA 13.63: Nombre de variables seleccionades en les optimitzacions basades en mètodes híbrids i d'embolcall a CDV

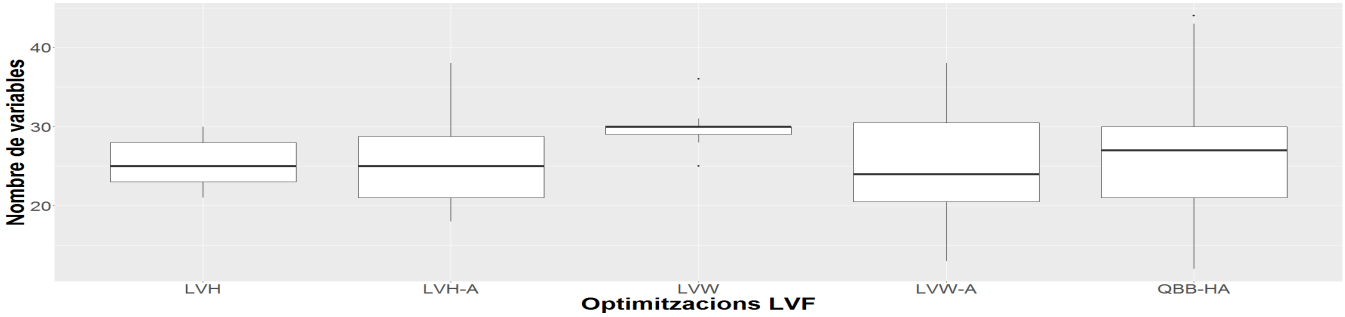


FIGURA 13.64: Nombre de variables seleccionades en les optimitzacions basades en mètodes híbrids i d'embolcall a CDC

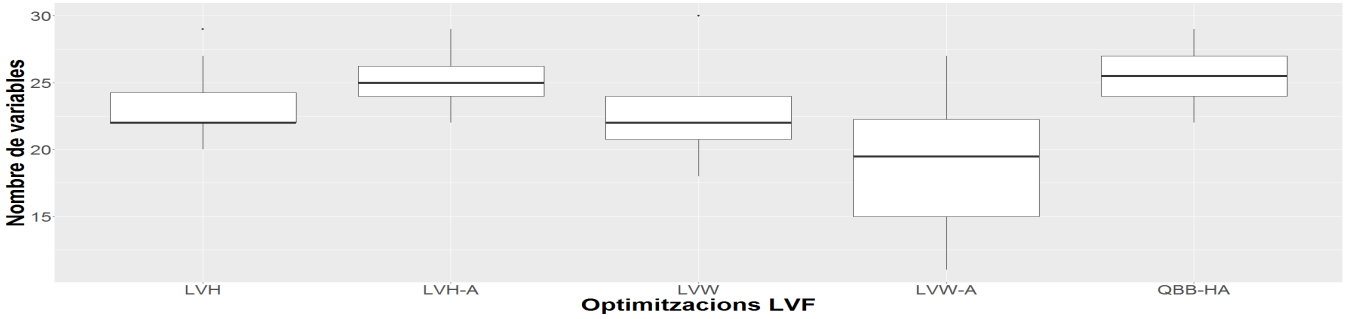


FIGURA 13.65: Nombre de variables seleccionades en les optimitzacions basades en mètodes híbrids i d'embolcall a CDW

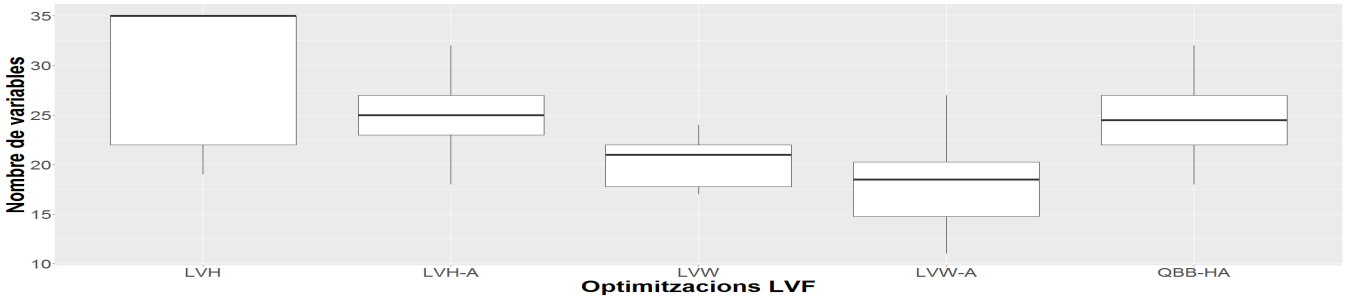


FIGURA 13.66: Nombre de variables seleccionades en les optimitzacions basades en mètodes híbrids i d'embolcall a CDB

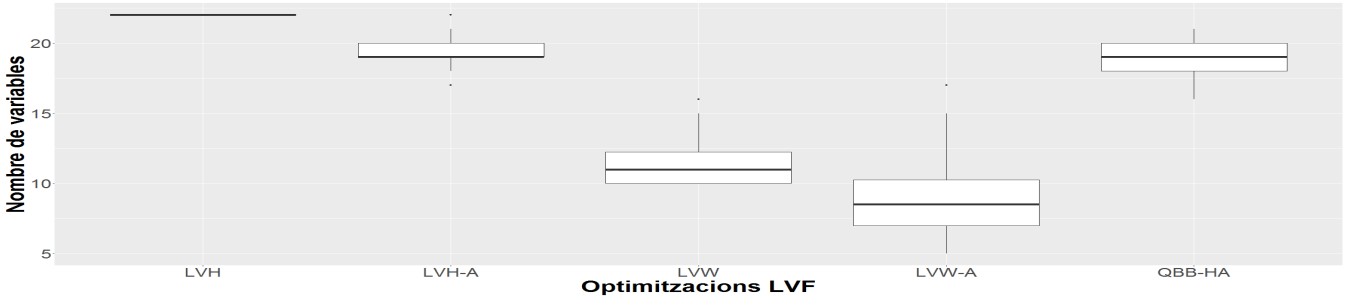


FIGURA 13.67: Nombre de variables seleccionades en les optimitzacions basades en mètodes híbrids i d'embolcall a CDH

13.4.2 Resultats del temps d'execució

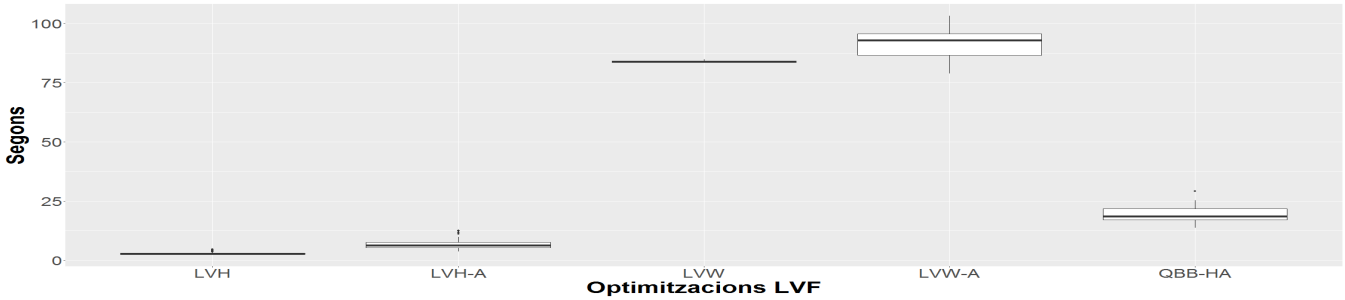


FIGURA 13.68: Temps d'execució en les optimitzacions basades en mètodes híbrids i d'embolcall a CDI

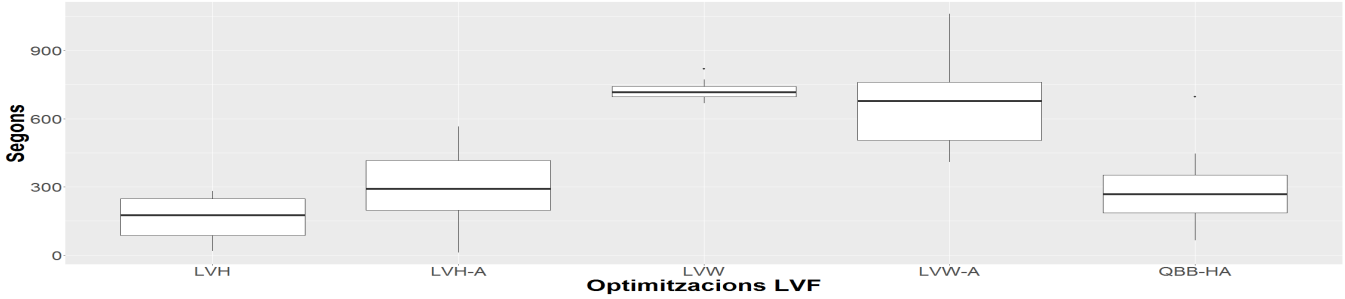


FIGURA 13.69: Temps d'execució en les optimitzacions basades en mètodes híbrids i d'embolcall a CDM

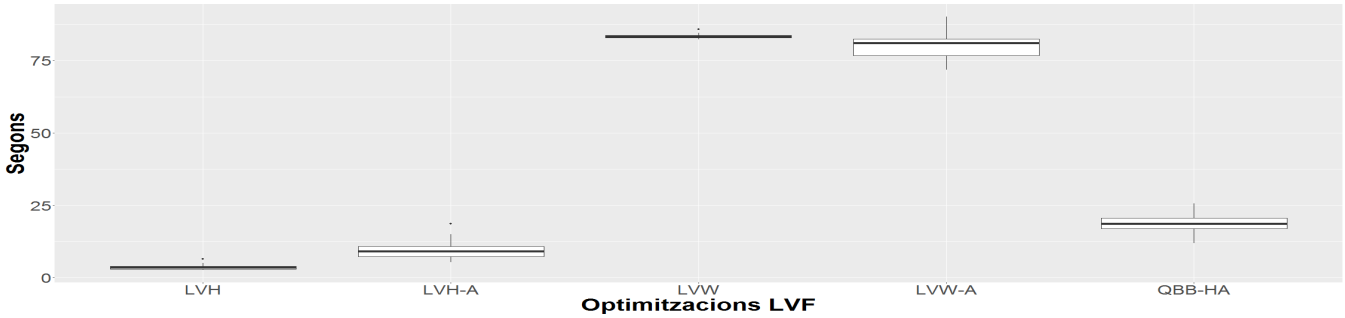


FIGURA 13.70: Temps d'execució en les optimitzacions basades en mètodes híbrids i d'embolcall a CDV

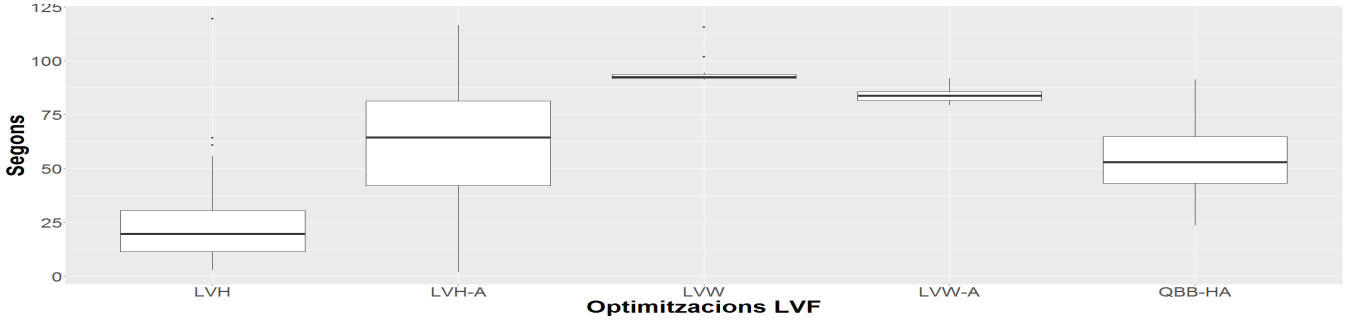


FIGURA 13.71: Temps d'execució en les optimitzacions basades en mètodes híbrids i d'embolcall a CDC

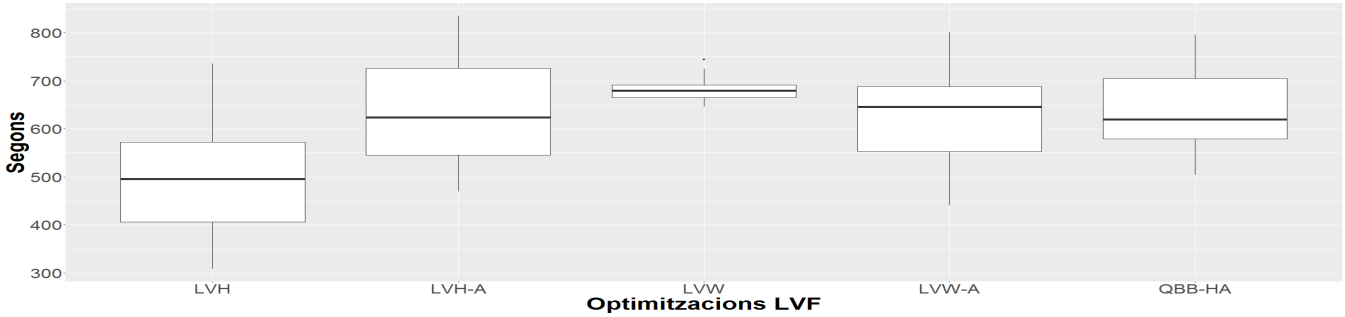


FIGURA 13.72: Temps d'execució en les optimitzacions basades en mètodes híbrids i d'embolcall a CDW

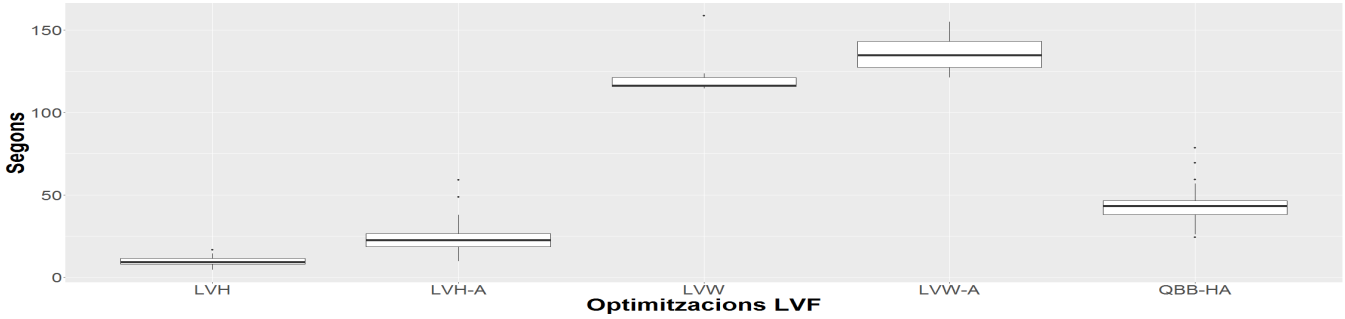


FIGURA 13.73: Temps d'execució en les optimitzacions basades en mètodes híbrids i d'embolcall a CDB

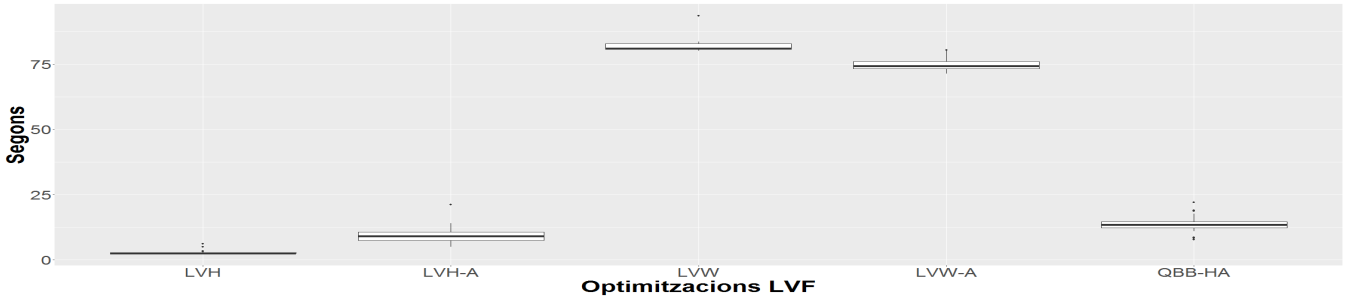


FIGURA 13.74: Temps d'execució en les optimitzacions basades en mètodes híbrids i d'embolcall a CDH

13.4.3 Resultats de l'*accuracy* amb arbres de decisió

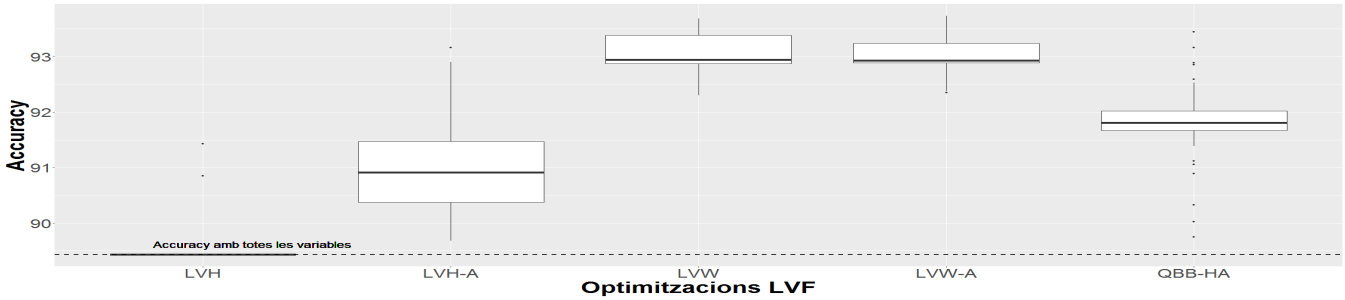


FIGURA 13.75: Accuracy de les optimitzacions basades en mètodes híbrids i d'embolcall amb arbres de decisions a CDI

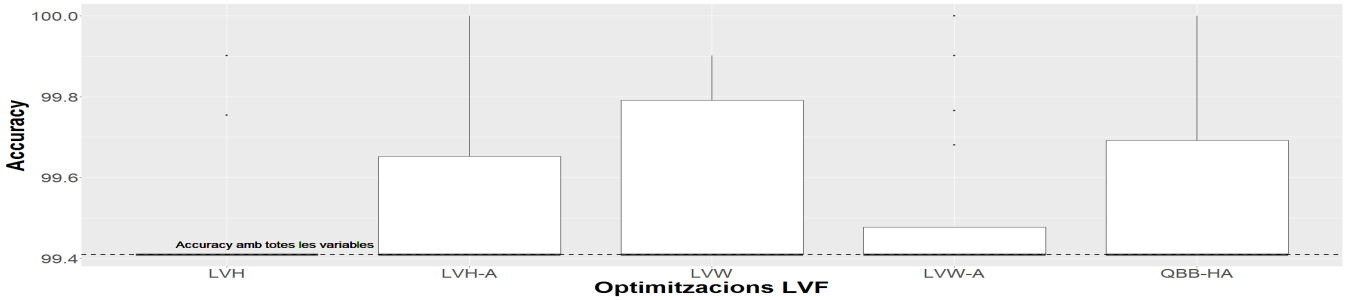


FIGURA 13.76: Accuracy de les optimitzacions basades en mètodes híbrids i d'embolcall amb arbres de decisions a CDM

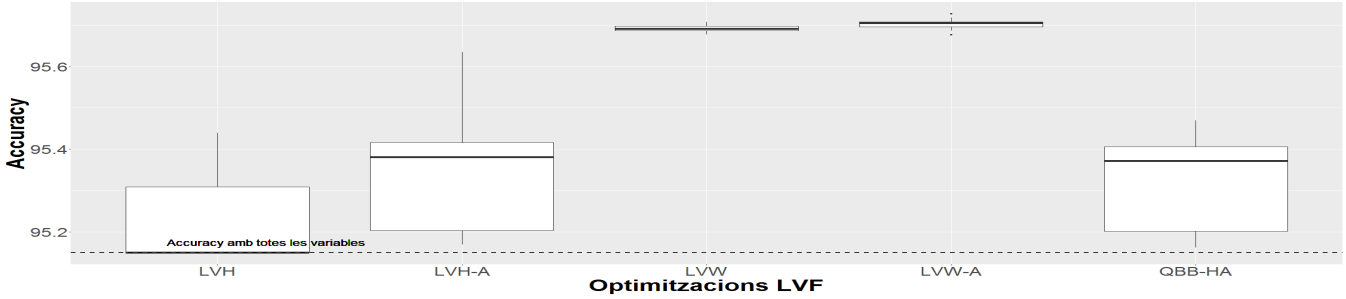


FIGURA 13.77: Accuracy de les optimitzacions basades en mètodes híbrids i d'embolcall amb arbres de decisions a CDV

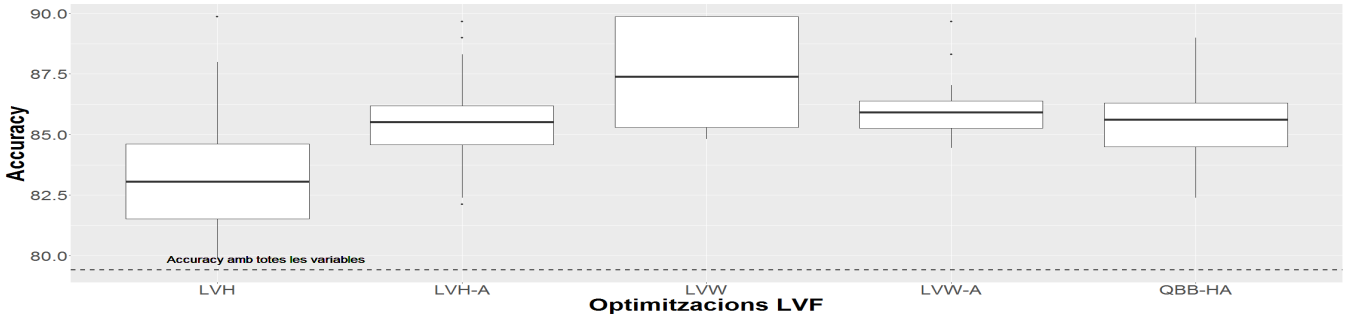


FIGURA 13.78: Accuracy de les optimitzacions basades en mètodes híbrids i d'embolcall amb arbres de decisions a CDC

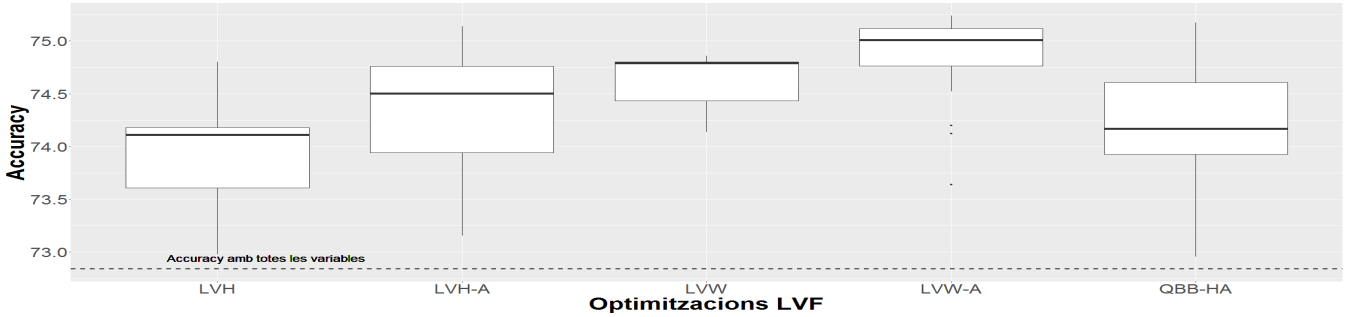


FIGURA 13.79: Accuracy de les optimitzacions basades en mètodes híbrids i d'embolcall amb arbres de decisions a CDW

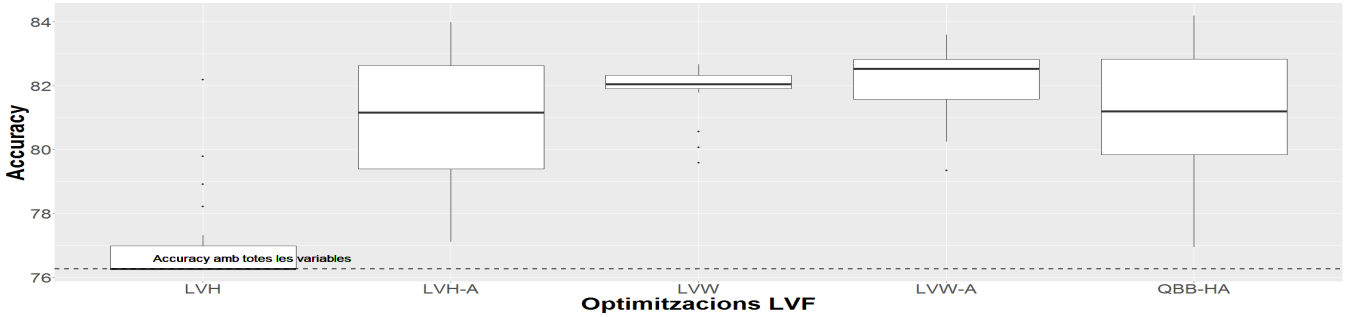


FIGURA 13.80: Accuracy de les optimitzacions basades en mètodes híbrids i d'embolcall amb arbres de decisions a CDB

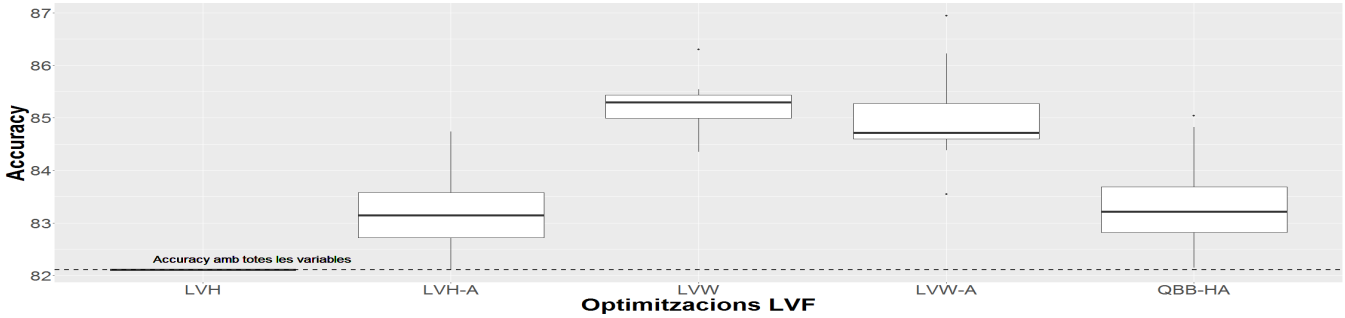


FIGURA 13.81: Accuracy de les optimitzacions basades en mètodes híbrids i d'embolcall amb arbres de decisions a CDH

Acrònims

A

ABB Automatic Branch and Bound
ABB-H Automatic Branch and Bound Hybrid

B

B&B Branch and Bound

C

CAIM Class-Attribute Interdependency Maximization
CDB Conjunt de dades Large Soybean Database
CDC Conjunt de dades Connectionist Bench (Sonar, Mines vs. Rocks)
CDE Conjunt de dades de l'escala d'equilibris
CDH Conjunt de dades SPECT Heart
CDI Conjunt de dades Ionosphere
CDL Donjunt de dades lógic
CDM Conjunt de daades Mushroom
CDP Conjunt de dades de paritat
CDV Conjunt de dades Congressional Voting Records
CDW conjunt de dades Waveform Database Generator (Version 2)
CG Costos genèrics
CPA Costos de personal i activitats

E

EUA Estats Units d'Amèrica

F

FF Fase final
FI Fase inicial
FM Fase intermèdia

G

GP Gestió del Projecte

K

kWh Kilowatt-hour

L

LVA Las Vegas Adaptative
LVF Las Vegas Filter
LVH Las Vegas Hybrid
LVH-A Las Vegas Hybrid Adaptative
LVI Las Vegas Incremental
LVI-A Las Vegas Incremental Adaptative
LVW Las Vegas Wrapper
LVW-A Las Vegas Wrapper Adaptative

Q

QBB Quick Branch and Bound
QBB-A Adaptative Quick Branch and Bound
QBB-HA Adaptative Quick Branch and Bound Hybrid

R

RPART Recursive Partitioning And Regression Trees

S

SBG Sequential Backward Generation
SFG Sequential Forward Generation
SPECT Single Photon Emission Computed Tomography

T

TFG Treball de fi de grau
TP Treball previ

U

UCI University of California, Irvine

W

W-SBG Wrapper Sequential Backward Generation
W-SFG Wrapper Sequential Forward Generation

Índex de paraules

Referències

[1] W. Turkey John. "The Future of Data Analysis". In: *Article of the Princeton University and Bell Telephone Laboratories* (1964). URL: https://projecteuclid.org/download/pdf_1/euclid.aoms/1177704711.

[2] H. Cormen Thomas, E. Leiserson Charles, L. Rivest Ronald, and Stein Clifford. *Introduction to Algorithms, Second Edition*. MIT Press, 2001. ISBN: 978-0262032933.

[3] Motwani Rajeev and Raghavan Prabhakar. *Randomized Algorithms*. Press Syndicate of the University of Cambridge, 1995. ISBN: 978-0521474658.

[4] J. McConnell Jeffrey. *Analysis of Algorithms: An Active Learning Approach*. Jones and Bartlett Publishers, 2001, pp. 249–250. ISBN: 978-0763716349.

[5] D. Kelleher John, Mac Namee Brian, and D'Arcy Aoife. *Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies*. MIT Press, 2015, pp. 40–45. ISBN: 978-0262029445.

[6] Belanche Lluís A., Molina Luis Carlos, and Nebot Àngela. "Feature Selection Algorithms: A Survey and Experimental Evaluation". In: *Article of the Universitat Politècnica de Catalunya, Barcelona, Spain* (), pp. 2–4.

[7] Belanche Lluís A. "Review and Evaluation of Feature Selection Algorithms in Synthetic Problems". In: *Article of the Universitat Politècnica de Catalunya, Barcelona, Spain* (2011), pp. 2–4. URL: https://arxiv.org/abs/1101.2320.

[8] Liu Huan and Setiono Rudy. "Incremental Feature Selection". In: *Article of the Nacional University of Singapore* (), pp. 7–15.

[9] M Dash and H. Liu. "Hybrid search of feature subsets". In: *Proceedings of the 15th Pacific Rim International Conference on Artificial Intelligence (PRICAI-98), Singapore* (1998), pp. 22–27.

[10] Jović A., Brkić K., and Bogunović N. "A review of feature selection methods with applications". In: *Article of the University of Zagreb* (). URL: https://bib.irb.hr/datoteka/763354.MIPRO_2015_JovicBrkicBogunovic.pdf.

[11] J Doak. "An Evaluation of Feature Selection Methods and their Application to Computer Security". In: *Technical Report CSE–92–18, Davis, CA: University of California, Department of Computer Science* (1992).

[12] D. W. Aha and R. L. Bankert. "A Comparative Evaluation of Sequential Feature Selection Algorithms". In: *In Proc. of the 5th International Workshop on Artificial Intelligence and Statistics* (1995), pp. 1–7.

[13] Kira Kenji and Rendell Larry A. *A Practical Approach to Feature Selection*. Morgan Kaufmann, 1992, pp. 249–256.

[14] Dash M., Liu Huan, and Motoda Hiroshi. "Feature Selection for Classification". In: *Intelligence Data Analysis: An International Journal* (1997), pp. 1–27.

[15] I Kononenko and T.G. Dietterich. "Estimating Attributes: Analysis and Extensions of Relief". In: *Proc. of the European Conf. on Machine Learning, Vienna* (1994), pp. 171–181.

[16] H Almuallim and T.G. Dietterich. "Learning with Many Irrelevant Features". In: *Proc. of the 9th National Conf. on Artificial Intelligence, volume 2, Anaheim, CA* (1991), pp. 547–552.

[17] H Almuallim and T.G. Dietterich. "Learning Boolean Concepts in the Presence of Many Irrelevant Features". In: *Artificial Intelligence, 69(1–2)* (1994), pp. 279–305.

[18] Dash Manoranjan and Liu Huan. "Consistency-based search in feature selection". In: *Technical Report, Department of Computer Science and Engineering, Arizona State University, USA and Electrical and Computer Engineering, Northwestern University, USA* (2003), p. 164.

[19] Dash Manoranjan, Liu Huan, and Motoda Hiroshi. "Consistency Based Feature Selection". In: *Article of the Nacional University of Singapore* (), pp. 2–3.

[20] Liu Huan and Setiono Rudy. "Scalable Feature Selection for Large Sized Databases". In: *In Proc. of the 4th World Congress on Expert Systems* (1998), pp. 68–75.

[21] M Boddy and T.L. Dean. "Deliberation scheduling for problem solving in time-constrained environments". In: *Artificial Intelligence, 67(2)* (1994), pp. 245–285.

[22] M Dash and H. Liu. "Hybrid search of feature subsets". In: *Proceedings of the 15th Pacific Rim International Conference on Artificial Intelligence (PRICAI-98), Singapore* (1998), pp. 238–249.

[23] H. Liu and R. Setiono. "Feature Selection and Classification: a Probabilistic Wrapper Approach". In: *Proc. of the 9th Int. Conf. on Industrial and Engineering Applications of AI and ES* (1996), pp. 129–135.

[24] Liu Huan and Motoda Hiroshi. *Computational Methods of Feature Selection*. Chapman and Hall/CRC, 2007, pp. 3–6. ISBN: 978-1584888789.

[25] Jensen Richard and Shen Qiang. *Computational Intelligence and Feature Selection: Rough and Fuzzy Approaches*. John Wiley and Sons, 2008, pp. 3–4. ISBN: 978-0470377918.

[26] Scrum Team. *What is Scrum?* URL: https://www.scrum.org/resources/what-is-scrum.

[27] Google Team. *Google Meet main page*. URL: https://meet.google.com/.

[28] R Team. *What is R?* URL: https://www.r-project.org/about.html.

[29] Ihaka Ross. "R: Past and future history." In: *Article of The University of Auckland* (1996), pp. 1–5. URL: https://www.stat.auckland.ac.nz/~ihaka/downloads/Interface98.pdf.

[30] Laxmi Lydia E, Shankar K, S.Sheeba Rani, and Lakshmanaprabu S.K. *Statistical Predictive Modelling through R Programming*. Evincepub Publishin, 2019, p. 216. ISBN: 978-9389125603.

[31] RStudio Team. *Main page RStudio*. URL: https://rstudio.com/.

[32] RStudio Team. *RStudio, new open-source IDE for R*. URL: https://blog.rstudio.com/2011/02/28/rstudio-new-open-source-ide-for-r.

[33] LaTeX Team. *LaTeX – A document preparation system*. URL: https://www.latex-project.org/.

[34] Tex Users Group. *TUG Main Page*. URL: https://www.tug.org/.

[35] Gantter Team. *The project management tool that's perfect for remote collaboration*. URL: https://www.gantter.com/.

[36] Tim Hume. *Balance Scale Data Set*. URL: https://archive.ics.uci.edu/ml/datasets/balance+scale.

[37] Irvine University of California. *UC Irvine Machine Learning Repository*. URL: https://archive.ics.uci.edu/ml/index.php.

[38] Fahlman Scott E. and Lebiere Christian. "The cascade-correlation learning architecture". In: *Technical Report CMU-CS-90-100, School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania, United States* (1989), pp. 6–10. URL: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.471.2777&rep=rep1&type=pdf.

[39] Prechelt Lutz. "Proben1, A Set of Neural Network Benchmark Problems and Benchmarking Rules". In: *Technical Report 21/94, Fakultät für Informatik, Universität Karlsruhe, Karlsruhe, Germany* (1994), pp. 6–7. URL: https://publikationen.bibliothek.kit.edu/39794/2050.

[40] Lukasz A. Kurgan and Krzysztof J. Cios. "CAIM Discretization Algorithm". In: *IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 2* (2004), pp. 145–153.

[41] Vince Sigillito. *Ionosphere Data Set*. URL: https://archive.ics.uci.edu/ml/datasets/ionosphere.

[42] Jeff Schlimmer. *Mushroom Data Set*. URL: https://archive.ics.uci.edu/ml/datasets/mushroom.

[43] Jeff Schlimmer. *Congressional Voting Records Data Set*. URL: https://archive.ics.uci.edu/ml/datasets/congressional+voting+records.

[44] Terry Sejnowski. *Connectionist Bench (Sonar, Mines vs. Rocks) Data Set*. URL: https://archive.ics.uci.edu/ml/datasets/Connectionist+Bench+(Sonar,+Mines+vs.+Rocks).

[45] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Waveform Database Generator (Version 2) Data Set*. URL: http://archive.ics.uci.edu/ml/datasets/waveform+database+generator+(version+2).

[46] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Waveform Database Generator (Version 1) Data Set*. URL: http://archive.ics.uci.edu/ml/datasets/waveform+database+generator+(version+1).

[47] R.S. Michalski and R.L. Chilausky. *Large Soybean Database*. URL: https://archive.ics.uci.edu/ml/datasets/Soybean+(Large).

[48] Lukasz A. Kurgan Krzysztof J. Cios. *SPECT Heart Data Set*. URL: https://archive.ics.uci.edu/ml/datasets/SPECT+Heart.

[49] Christian Roever, Nils Raabe, Karsten Luebke, Uwe Ligges, Gero Szepannek, Marc Zentgraf, and David Meyer. *klaR: Classification and Visualization*. URL: https://cran.r-project.org/web/packages/klaR/.

[50] Beth Atkinson. *Documentation: rpart function*. URL: https://www.rdocumentation.org/packages/rpart/versions/4.1-15/topics/rpart.

[51] Terry Therneau, Beth Atkinson, and Brian Ripley. *rpart: Recursive Partitioning and Regression Trees*. URL: https://cran.r-project.org/web/packages/rpart/.

[52] Zhang Zhongheng. "Decision Tree Modeling Using R". In: *Annals of Translational Medicine, Vol. 4, No. 15* (2016).

[53] Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman and Hall/CRC, 2015, p. 13. ISBN: 978-1498712163.

[54] Belanche Lluís A. "Review and Evaluation of Feature Selection Algorithms in Synthetic Problems". In: *Article of the Universitat Politècnica de Catalunya, Barcelona, Spain* (2011), pp. 6–7. URL: https://arxiv.org/abs/1101.2320.

[55] Huan Liu and Rudy Setiono. "Feature selection and classification - A probabilistic wrapper approach". In: *Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Proceedings of the Ninth International Conference, Fukuoka, Japan* (1996).

[56] G. Sedó Ramon. *El diagrama de Gantt*. URL: http://multimedia.uoc.edu/blogs/metodologia/es/el-diagrama-de-gantt/.

[57] PagePersonnel Team. *Página principal PagePersonnel*. URL: https://www.pagepersonnel.es/.

[58] PagePersonnel Team. *Salary comparison tool*. URL: https://www.pagepersonnel.es/salary-comparison-tool.

[59] Selectra Team. *Página principal Selectra*. URL: https://selectra.es/.

[60] Selectra Team. *Tarifaluzhora: Precio de la luz por horas*. URL: https://tarifaluzhora.es/.