



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Escola d'Enginyeria de Barcelona Est

TREBALL FI DE GRAU

Grau en Enginyeria Química

**DESENVOLUPAMENT D'UN SISTEMA DE MONITORITZACIÓ
DE PLANTA PER A LA DETECCIÓ DE MALFUNCIONS AMB
EINES AUTOMÀTIQUES DE RECONeixEMENT DE PATRONS**



Memòria i Annexos

Autor: Martí Tena Tena
Director: Moisès Graells
Co-Director: Gerard Companya
Convocatòria: Juny 2022

Resum

Aquest projecte té com a objectiu la ideació d'un prototip de programa per a la creació d'un sistema d'entrenament i aplicació de models predictius per a plantes de producció, amb la intenció de crear un detector de fallades en controlador. Per aquest cas en concret, es va seleccionar el sistema de control anti-surge, ja que és un sistema utilitzat en moltes plantes químiques.

Aquest programa buscarà usar els nous mitjans que ofereixen els serveis integrats al núvol per a estudiar com aquestes aplicacions poden beneficiar a les fàbriques a gestionar de manera deslocalitzada les seves dades.

Després d'un estudi de les diferents tecnologies possibles amb les quals es podria realitzar aquesta tasca es va escollir utilitzar les eines d'Unisim per a la Simulació d'una planta, Python per a la gestió de les dades i el control dels diferents programes, Azure (Conjunt de serveis integrats al núvol creat per Microsoft) per l'emmagatzematge de dades, la creació del model, i la implementació d'aquest.

El programa ha estat ideat per a poder funcionar tant amb Unisim com amb una suposada planta real, ja que es farà servir el protocol MQTT (*Message Queuing Telemetry Transport*), un protocol de missatgeria que està sent utilitzat actualment en la indústria.

S'ha aconseguit generar un programa útil per a la detecció d'errors, amb el qual es demostra la capacitat de les tecnologies de Machine Learning per a la creació de plantes intel·ligents, amb poca intervenció humana.

Resumen

Este proyecto tiene como objetivo la ideación de un prototipo de programa para la creación de un sistema de entrenamiento y aplicación de modelos predictivos para plantas de producción, con la intención de crear un detector de fallos en controlador. Por este caso en concreto, se seleccionó el sistema de control anti-surge, puesto que se un sistema simple utilizado en muchas plantas químicas.

Este programa buscará utilizar los nuevos medios que ofrecen los servicios integrados en la nube para estudiar como estas aplicaciones pueden beneficiar a las fábricas a gestionar de manera deslocalizada sus datos.

Después de un estudio de las diferentes tecnologías posibles con las que se podría realizar esta tarea se escogió utilizar las herramientas de Unisim para la Simulación de una planta, Python para la gestión de los datos y el control de los diferentes programas, Azure (Conjunto de servicios integrados a la nube creada por Microsoft) por el almacenamiento de datos, la creación del modelo, y la implementación del mismo.

El programa ha sido ideado para poder funcionar tanto con Unisim como con una supuesta planta real, puesto que se utilizará el protocolo MQTT (Message Queuing Telemetry Transporte), un protocolo de mensajería que está siendo utilizado actualmente en la industria.

Se ha conseguido generar un programa útil para la detección de errores, con el cual se demuestra la capacidad de las tecnologías de Machine Learning para la creación de plantas inteligentes, con poca intervención humana.

Abstract

This project aims to develop a prototype program for the creation of a predictive model training and application system for production plants, with the intention of creating a controller fault detector. For this specific case, the anti-surge control system was selected, as it is a simple system used in many chemical plants.

This program will seek to use the new means offered by integrated cloud services to study how these applications can benefit the manufactures from relocating their data.

After a study of the different possible technologies with which this task could be performed, it was chosen to use the Unisim tools for the Simulation of a plant, Python for the management of data and control of the different programs, Azure (Common-integrated Cloud Services Set up by Microsoft) for data storage, model creation, and implementation of the data.

The program has been designed to be able to operate both with Unisim and with a supposed real plant, as the MQTT protocol (Message Queuing Telemetry Transport) will be used, a messaging protocol that is currently being used in the industry.

A useful program for error detection has been created, demonstrating the ability of Machine Learning technologies to create intelligent plants, with little human intervention.



Agraïments

Aquest treball no hauria sigut possible sense l'ajuda dels dos directors, en Moisès Graells i en Gerard Campanya, els quals han aportat ajuda de seguiment setmanalment, guiant-me i solucionant els dubtes que podien sorgir durant el transcurs del projecte.

També voldria agrair a l'UPC per oferir les diferents llicències i eines per a poder desenvolupar el projecte sense grans costos.

Per últim, agrair a la meua família el suport emocional durant el transcurs del treball i de tot el grau, sense el qual no hauria sigut possible assolir-ho.



Glossari

Algorisme= Conjunt de regles per a resoldre un problema en un nombre finit de passos.

Sobreajustament = Acte d'entrenar un algorisme d'aprenentatge només en uns casos en concret, fent que aquest model sigui molt concret per a alguns casos, provocant que el model tingui més errors en casos no inclosos en l'entrenament.

CSV = "Coma-separated values" és un format de document per representar dades d'una taula, les columnes se separen per comes, i les files per salts de pàgina.

JSON = JSON és un estàndard obert basat en text dissenyat per a intercanvi de dades llegible per humans.

Broker = Programa intermediari que regula i tradueix els missatges entre dues aplicacions.

Índex

RESUM	I
RESUMEN	II
ABSTRACT	III
AGRAÏMENTS	V
GLOSSARI	VI
1. INTRODUCCIÓ	3
1.1. Origen del treball	3
1.2. Objectius del treball	3
1.3. Abast del treball	3
2. INDÚSTRIA 4.0	5
2.1. Les anteriors revolucions industrials	5
2.2. La Quarta Revolució Industrial.....	7
2.2.1. CPS	7
2.2.2. IoT	7
2.2.3. Computació en el Núvol (cloud computing)	8
2.2.4. Intel·ligència artificial.....	8
2.3. Machine Learning.....	8
2.4. Serveis en el Núvol.....	9
2.4.1. Emmagatzematge en el núvol.....	9
2.4.2. IoT Hub.....	10
2.4.3. Machine Learning en el Núvol.....	10
3. SOFTWARE DE SIMULACIÓ DE PROCESSOS	11
3.1. Simulació Dinàmica.....	11
4. DESENVOLUPAMENT	12
4.1. Selecció de les eines.....	12
4.1.1. Selecció Simulador.....	12
4.1.2. Selecció servei en el núvol.....	12
4.1.3. Selecció “llenguatge de programació”	13
4.1.4. Selecció protocol comunicació.....	13
4.1.5. Selecció model ML.....	14

4.1.6.	Selecció “case study”	14
4.2.	Implementació i configuració	16
4.2.1.	Preparacions prèvies	16
4.2.2.	Presa de dades	19
4.2.3.	Creació del model	21
4.2.4.	Utilitzar el model predictiu	22
5.	MODEL MATEMÀTIC	23
5.1.	Ajust d’un model	23
5.2.	Modelització en Azure	23
5.3.	Voting ensemble	23
5.4.	TruncatedSVDWrapper/ElasticNet	24
5.4.1.	ElasticNet	24
5.4.2.	TruncatedSVDWrapper	24
6.	VALIDACIÓ	26
6.1.	Mètriques del model	26
6.2.	Comprovació pràctica	27
6.2.1.	Funcionament correcte	27
6.2.2.	Funcionament incorrecte	29
7.	ANÀLISI DE L’IMPACTE AMBIENTAL	33
	CONCLUSIONS	35
	PRESSUPOST I/O ANÀLISI ECONÒMICA	37
	BIBLIOGRAFIA	41
	ANNEX 1 CODI PROGRAMA	43
	ANNEX 2 EXPLICACIÓ MODEL ELASTICNET	52



1. Introducció

1.1. Origen del treball

Actualment l'indústria està patint una modernització, buscant automatitzar i millorar l'eficiència de les plantes de producció

Aquest treball ha sigut ideat a partir d'una idea inicial dels directors del projecte. Prèviament s'havia realitzat un treball de fi de màster, amb el que es van generar eines que s'han utilitzat en aquest projecte.

1.2. Objectius del treball

Aquest treball té com a objectiu la creació d'un prototip amb el qual demostrar la utilitat de les noves eines de les que es disposen en la indústria, i com aquestes eines poden beneficiar a una transició cap a fàbriques més automatitzades, més segures, i més eficients. Es pretén aconseguir la implementació d'un model predictiu de Machine Learning en el núvol, per a comprovar si una planta està funcionant de la manera esperada.

Per a assolir-ho s'han marcat els objectius següents:

- Crear una aplicació per a crear models que permetin detectar fallades en una planta.
- Demostrar la utilitat i la viabilitat dels models predictius i de la computació en el núvol.
- Comparar diferents eines amb les quals es pot dur a terme aquest projecte, i seleccionar les més indicades.
- Incloure protocols de comunicació IoT, per a oferir la possibilitat de substituir el simulador per una planta real

1.3. Abast del treball

El que es busca amb aquest treball és comprovar la utilitat de les eines seleccionades, així com oferir facilitats per a futures millores que puguin fer altres possibles projectes. El projecte estarà centrat en un cas concret de simulació, d'on s'extrauran només les dades necessàries per a la predicció. Només es crearà un model per al control antisurge a manera d'exemple pràctic.

Aquestes dades es penjaran al núvol mitjançant el protocol MQTT. Un cop al núvol s'entrenarà un model matemàtic, el qual es podrà implementar de manera remota.

2. Indústria 4.0

Una revolució industrial es coneix com a un procés de transformació que pateix la societat. Aquests canvis afecten la producció, l'economia i la demografia de les societats on afecta.

Els principals motors per a una revolució industrial són, segons l'autor Jeremy Rifkin, noves comunicacions, noves fonts d'energia, i nous mitjans de transport.[1]

Aquestes revolucions venen produïdes per innovacions científiques i tecnològiques, que altera la capacitat de producció, així com les relacions productives.

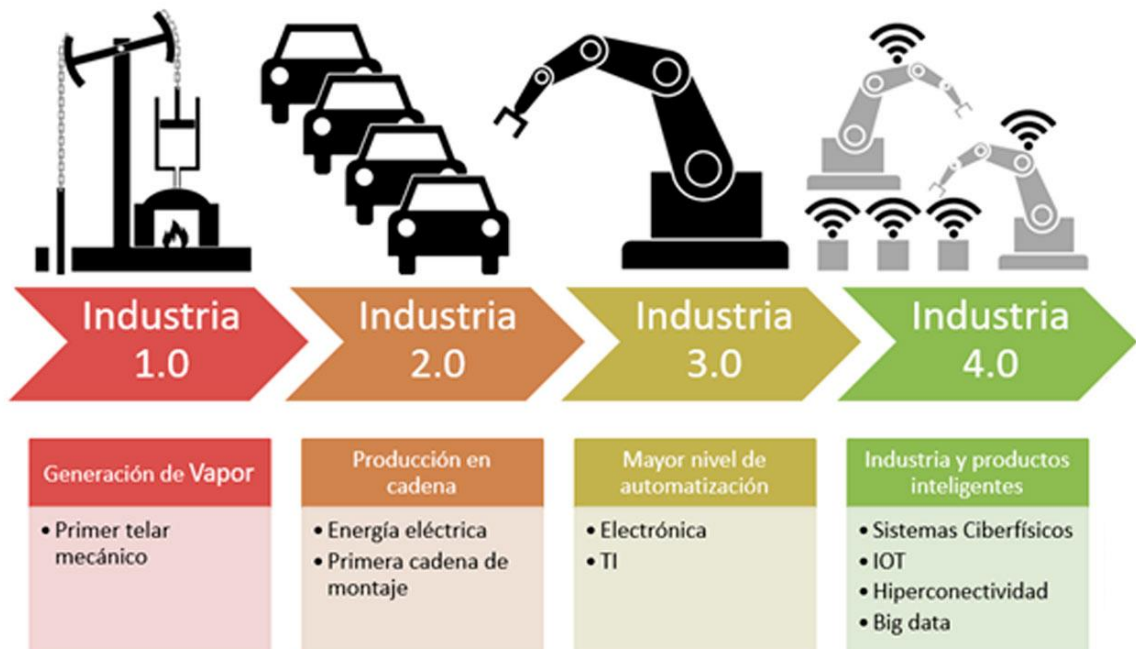


Figura 1 Les 4 Revolucions Industrials (Font: <https://www.avansis.es/industria-4-0/industria-4-0-revolucion-evolucion>)

2.1. Les anteriors revolucions industrials

Els historiadors estableixen tres revolucions industrials, des de la segona meitat del segle XVII fins la primera dècada del segle XX.

Primera revolució industrial

La Primera Revolució Industrial, també coneguda simplement com La Revolució Industrial, va ser impulsada per la creació de la màquina de vapor. Es va iniciar a Gran Bretanya i es va estendre per l'Europa Occidental fins a l'any 1840. Es considera el conjunt de transformacions més gran que ha patit la societat des del Neolític.

Va incloure innovacions com la industrialització de la majoria de processos productius, la creació del ferrocarril o el vaixell de vapor, així com la impressió en massa de text o el telègraf. La població va augmentar en gran manera, i van créixer els nuclis urbans.

Segona revolució industrial

La Segona Revolució Industrial és referida als canvis produïts entre 1870 fins a l'inici de la Primera Guerra Mundial l'any 1914.

Durant aquest període, va accelerar molt la industrialització gràcies a noves fonts d'energia com el gas o l'electricitat, noves eines de transport com els automòbils o l'avió i nous materials com l'acer o el petroli. Així com noves comunicacions mitjançant la ràdio o el telèfon.

Es van introduir les primeres cadenes de muntatge.

Aquests canvis van provocar l'inici de la globalització. [2]

Tercera revolució industrial

La Tercera Revolució Industrial, també coneguda com la Revolució Científicotecnològica (RCT), es va originar als Estats Units i al Japó, durant la dècada dels 70.

L'electrònica i la informàtica, van transformar la indústria així com l'economia del món occidental. L'augment de l'automatització, mitjançant ordinadors i robots, va provocar la disminució de la necessitat de mà d'obra en la indústria, fent que una gran part de la població es dediqui al sector terciari, augmentant l'especialització dels treballadors.

2.2. La Quarta Revolució Industrial

Actualment, estem vivint un nou procés de canvi en la producció, aquesta revolució és coneguda com a Indústria 4.0. Aquest concepte va ser nomenat per Klaus Schwab, creador del Fòrum Econòmic Mundial, l'any 2016.

Aquesta revolució, seria la primera en la història que s'ha pogut preveure, això provoca que companyies i els centres de recerca puguin modelar com s'avança en aquest procés. També causa que la indústria 4.0 sigui una prioritat en la recerca i en el desenvolupament en tot el món. [3]

Els principals components de l'indústria 4.0 són:

- Els sistemes ciberfísics o CPS (de l'anglès Cyberphysical System)
- Internet de les coses o IoT (de l'anglès Internet of Things)
- La computació en el núvol
- Intel·ligència Artificial

2.2.1. CPS

Un CPS és un sistema físic controlat i monitorat per algoritmes basats en computació i internet, aquests sistemes connecten mecanismes físics entre ells mitjançant la computació i intercanviant dades per alterar el funcionament dels mateixos. A diferència d'un sistema incrustat, el qual està dissenyat per realitzar un baix nombre de funcions diferents, un sistema CPS presenta una combinació més alta de connexions i coordinació de diferents elements físics i computacionals.

Alguns exemples serien l'automòbil autònom, els electrodomèstics moderns o les plantes industrials autogestionades.

Amb la incorporació dels algoritmes, es busca reduir la interacció humana així com l'augment de la precisió del control i la capacitat d'anàlisi d'una quantitat de dades molt més elevada, impossible de gestionar per les persones, sense ajuda de la computació.

2.2.2. IoT

Internet de les coses o IoT, descriu tot sistema amb dispositius físics a través de xarxes sense fil, minimitzant la intervenció humana.

En els últims anys, és innegable l'ingrés d'internet en la societat a tot nivell, com és el cas de cotxes intel·ligents, nous electrodomèstics o cases domòtiques.

La indústria està integrant ràpidament aquestes tecnologies, en el que es coneix com IIoT (Industrial Internet of Things).

2.2.3. Computació en el Núvol (cloud computing)

La computació en el núvol és el subministrament de serveis informàtics a través d'internet. Aquests serveis són flexibles i fàcilment escalable.

Els principals beneficis que aporta el "cloud computing" són [4]:

- Eliminació de la inversió inicial necessària en Hardware i Software, ja que els pagaments es fan de manera gradual durant l'ús dels serveis.
- Els serveis oferts són escalables, pel que es poden seleccionar només els serveis que s'utilitzen, podent-se variar en qualsevol moment i des de qualsevol part del món.
- Son serveis molt segurs, amb bon rendiment i accessibles de manera no local.

2.2.4. Intel·ligència artificial

La intel·ligència artificial o IA, es pot definir com la capacitat de les màquines d'usar algorismes, aprendre de dades i utilitzar-ho per a prendre decisions. Aquest comportament pot ser anàleg a com les persones aprenen i treballen. El benefici d'utilitzar les màquines, és la capacitat que tenen de fer càlculs complexos, amb una gran quantitat de dades, sent capaces de reconèixer patrons ocults per a l'ull humà. Per altra banda, poden actuar de manera constant i fàcilment escalable.

2.3. Machine Learning

El Machine Learning, o aprenentatge màquina, és una rama de la IA amb la qual, utilitzant gran quantitat de dades es pot alimentar un algorisme, conegut com a model.

Els models d'aprenentatge automàtic són algorismes d'ordinador que fan servir dades per fer estimacions o decisions.

Els models d'aprenentatge automàtic difereixen dels algorismes tradicionals en com estan dissenyats. Per modificar o millorar un algorisme tradicional, s'ha de modificar el codi, en canvi, un algorisme d'aprenentatge automàtic utilitza les dades per millorar en una tasca específica. És a dir, afegint noves dades o modificant-les, es pot modificar el model.

Aquests models es poden implementar de manera que puguin fer prediccions de manera autònoma i continua, sense necessitat de cap tipus d'intervenció humana. [5]

2.4. Serveis en el Núvol

En el núvol, per a desenvolupar el projecte ens faran falta tres eines diferents, l'emmagatzematge, la recepció de dades mitjançant IoT i el servei de Machine Learning del núvol.

2.4.1. Emmagatzematge en el núvol

Un dels avantatges d'utilitzar el núvol, és la capacitat de poder emmagatzemar dades no localitzades en un ordinador en concret, per poder accedir a elles des de qualsevol màquina connectada a la xarxa d'internet.

Per altra banda, per a empreses o projectes que puguin tenir una gran variació de la memòria necessària al llarg del temps, l'emmagatzematge en el núvol ofereix la capacitat d'adaptar el cost a la memòria necessària, evitant que s'hagin de comprar grans quantitats de memòria física per a projectes temporals.

L'emmagatzematge pot ser estructurat o no estructurat.

- Emmagatzematge estructurat:

Dades que s'ajusten a un esquema fix, per tant, totes les dades han de tenir els mateixos camps o propietats.

- Emmagatzematge no estructurat:

En un emmagatzematge no estructurat, les dades poden ser de diferents classes i amb diferents propietats, per exemple arxius de text, imatges o arxius de dades.

2.4.2. IoT Hub

Un IoT Hub permet la comunicació entre dispositius IoT i el núvol.

Aquests serveis ofereixen una comunicació segura, amb eines per l'autenticació, l'administració de dispositius i la facilitat per escalar els serveis.

Mitjançant aquest servei, es pot rebre els missatges amb MQTT i guardar-los en el magatzem del núvol, d'on es poden recollir per a tota mena d'usos. En el cas d'aquest projecte, s'utilitzarà per a recopilar dades de manera anàloga a com es faria en una planta industrial.

2.4.3. Machine Learning en el Núvol

L'aprenentatge màquina en el núvol, permet crear, entrenar i implementar models. Aquestes eines permeten diferents aproximacions al ML, la gent experimentada pot implementar models de manera ràpida i eficaç, mentre que la gent que no és científica de dades ni programadora pot aconseguir models útils de manera senzilla i intuïtiva.

3. Software de simulació de processos

Aquest tipus de software són eines que mitjançant models matemàtics i equacions físiques, intenten simular el comportament del món real de la manera més precisa possible.

Aquests softwares s'utilitzen per a poder estudiar, dissenyar o preveure com actuaria un sistema físic en certes condicions, sense que sigui necessari la construcció del sistema o el càlcul de tots els diferents estats a estudiar. Això redueix en una gran magnitud els costos econòmics, monetaris, de temps i ambientals.

En la indústria química, els softwares de simulació són molt utilitzats per a poder dissenyar plantes per a futures construccions, ja que permeten valorar diferents possibilitats, buscar els valors òptims per a les diferents variables, que permetin tenir una planta amb els mínims costos i els màxims beneficis. També permeten provar alternatives als processos actuals per a estudiar implementar canvis en la planta.[6]

3.1. Simulació Dinàmica

Els simuladors poden treballar de manera estacionària o dinàmica.

La simulació estàtica no treballa amb la variable temporal, és útil per a l'enginyer que vol dissenyar una planta, ja que permet trobar l'equipament que reduirà el cost i maximitzarà la producció. En canvi, la simulació dinàmica estudia la planta amb evolució temporal, mitjançant gradients de pressió i equacions de balanç de volum, per tant, és l'indicat per operacions de control d'una planta.

Utilitza el Pressure-Flow Solver, el qual resol l'estat en cada instant de temps mitjançant la integració d'aquestes equacions. La simulació dinàmica considera diferents nusos al llarg del cas, on es calcula l'energia i la massa que conté.

La simulació dinàmica aporta informació sobre si la planta serà capaç d'arribar a l'estat estacionari de manera segura i sense problemes.

4. Desenvolupament

4.1. Selecció de les eines

4.1.1. Selecció Simulador

Hi ha diversos simuladors de processos químics que es podrien utilitzar per a aquesta tasca, alguns dels exemples són:

- Aspen HYSYS
- Aspen Plus
- HoneyWell Unisim Design

Els simuladors d'Aspen són els més utilitzats en la indústria. Tot i tenir eines semblants, l'Aspen Plus s'usa més per plantes de químics i productes farmacèutics, mentre que l'Aspen HYSYS és més indicat per a treballar amb hidrocarburs i la indústria petrolera.

L'Unisim és molt similar a la principal competència de Aspen. Tot i ser una mica menys intuïtiu en l'àmbit gràfic, aporta les mateixes eines i és robust. Té simulació dinàmica, necessària per a l'objectiu d'aquest treball.

Per aquest treball s'utilitzarà Unisim, ja que en un treball anterior Kevin Martínez Lopez va crear una eina amb Python per a poder llegir i escriure dades. Això permet no haver d'invertir tant de temps en crear la connexió i poder centrar-se en la finalitat del projecte.

4.1.2. Selecció servei en el núvol

La tecnologia en el núvol es un sector en creixement, per això les principals empreses informàtiques del món tenen els seus propis serveis.

- Amazon Web Services (AWS)
- Microsoft Azure
- Google Cloud

Totes aquestes opcions tenen eines semblants, i per aquest projecte podria ser útil qualsevol d'elles.

La plataforma escollida ha sigut Microsoft Azure, ja que gràcies a Azure For Students Microsoft ofereix un any de serveis gratuïts, així com 100 \$ per gastar en els productes de pagament.

A demés, Microsoft té un conjunt de guies per a aprendre conceptes generals i guies per a aprendre a utilitzar les pròpies eines d'Azure.



Figura 2 Logotip Azure

4.1.3. Selecció “llenguatge de programació”

Per al llenguatge de programació s'utilitzarà Python. Aquest llenguatge és un dels més usats, degut a la seva “simplicitat” és molt indicat per aquella gent que utilitza programes simples per a funcions no específiques de programació. Té una gran quantitat de biblioteques disponibles, la qual cosa facilita en gran mesura la creació d'aplicacions.



Figura 3 Logotip Python

En els últims anys ha guanyat molta influència degut a la seva capacitat en el Machine Learning, la intel·ligència artificial i la ciència de dades.

A més d'això, aquest és el llenguatge après durant la carrera i amb el que tinc més coneixements.

4.1.4. Selecció protocol comunicació

La comunicació amb el núvol es pot fer mitjançant diferents protocols de comunicació, actualment els principals protocols utilitzats son HTTP i MQTT. [7]

MQTT té una gran utilitat per a aparells que no disposen d'una gran amplada de banda, ja que té la capacitat d'enviar missatges molt més lleugers i de manera més optimitzada. Això ho fa un protocol indicat per a aplicacions IoT.

HTTP, en canvi, està més orientat a missatges més pesats. És fiable per a sistemes antics.

Com que aquesta aplicació estarà orientada a la transmissió permanent de dades, MQTT és més indicat. A més, Azure ofereix un servei de MQTT-Broker (IoT Hub).

4.1.5. Selecció model ML

Dins de l'abast d'aquest treball no està conèixer tots els possibles models de regressió, per tant, s'utilitzarà l'eina d'AutoML que ens ofereix Azure, s'estudiaran aquells models que aportin una error de desviació mitjana quadràtica similar i òptima.

Els models que menys error tenen son:

Taula 1 Resultats Models AutoML

Nom algoritme	Error de desviació mitjana quadràtica	Error absolut mitjà
Voting Ensemble	0.00119	0.007478
TruncatedSVDWrapper/ElasticNet	0.00119	0.007478
StackEnsemble	0.00121	0.007505

Tant Voting Ensemble i TruncatedSVDWrapper/ElasticNet són igual de eficients. A l'apartat 6 de la memòria es parla més sobre el funcionament d'aquest models.

4.1.6. Selecció "case study"

Com aquest prototip busca demostrar la capacitat d'utilitzar els models de predicció de l'aprenentatge màquina i els serveis en el núvol en la indústria química, el "case study" emprat no és l'important.

Per això s'estudiarà un compressor i el control per evitar el "surge", ja que és una fallida típica en la indústria, a demès Unisim ofereix tant un controlador com un generador de fallides, aquestes eines seran molt útils. El controlador ens assegura que a l'hora de recopilar dades per a l'entrenament els valors d'obertura de la vàlvula seran els indicats per a evitar el surge.

El generador de fallides dona la possibilitat de simular una possible fallida del controlador, per tant ens permet comprovar si el model predictiu podria donar-nos informació de quan s'estan produint mal funcions.

Compressor

Un compressor és una màquina utilitzada per a reduir el volum d'un fluid a la vegada que augmenta la seva pressió i temperatura. Són màquines tèrmiques, que transmeten energia de la màquina al fluid

corresponent en forma d'energia de flux. L'ús més habitual és el de compressió de l'aire, per a tasques neumàtiques, com inflar rodes, accionar vàlvules, equips de lubricació neumàtica. També s'utilitza amb altres fluids per desplaçar-los al llarg de la planta, o pel control de la pressió d'un corrent (per exemple per una reacció on sigui necessària una pressió mínima) [8] [9]



Figura 4 Compressor comercial [Font: <https://www.amazon.es/Compresor-Aire-Litros-2200W-Cilindros/dp/B01CLQZV8Q>]

Compressors dinàmics

Aquests compressors funcionen mitjançant una màquina amb rotació contínua, per on passa l'aire, que augmenta de velocitat i adquireix energia cinètica.

A diferència dels compressors de desplaçament positiu, aquests compressors funcionen de manera contínua, són més petits i produeixen molta menys vibració. Són els més utilitzats en la indústria química, ja que permeten un treball continu. [10] [11]

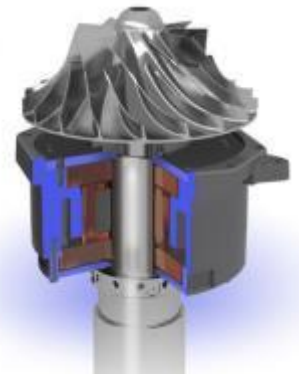


Figura 5 Compressor Dinàmic [Font: mundocompresor.com]

Surge

El "surge" és una mal funció que es produeix en els compressors dinàmics.

Aquest fenomen es produeix quan un corrent d'aire oscil·la en la direcció axial del compressor, provocant que el gradient de pressió varii de manera periòdica, podent arribar a ser negatiu. El "surge" provoca pulsacions de canvis de pressió que poden provocar greus fallides mecàniques, així com un major desgast de la maquinària.

Aquest fenomen es considera el causant d'aproximadament el 25% de les aturades d'emergència en les instal·lacions que utilitzen aquest tipus de compressor.

Control Anti-Surge

Per a evitar situacions crítiques, s'instal·la una recirculació, amb una vàlvula que regula el pas. Aquesta recirculació provoca que el flux del compressor no arribi als mínims de treball necessaris.

L'esquema d'Unisim d'aquest sistema és el següent:

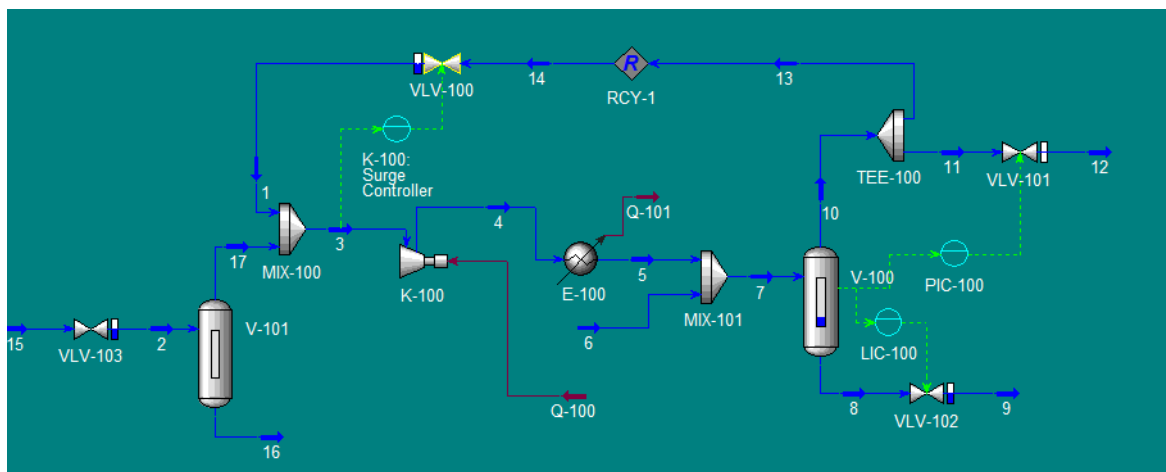


Figura 6 Esquema planta amb control anti-surge

4.2. Implementació i configuració

4.2.1. Preparacions prèvies

Per poder utilitzar el programa, s'ha d'haver preparat prèviament certes condicions, en el simulador, l'interpret de Python i el núvol.

Preparacions cas simulador

Per a poder comunicar les dades de simulació amb el Python, s'usen les anomenades "DataTable" d'unisim, aquestes són taules que monitoren certs estats del simulador i permeten tant llegir-los com escriure'ls.

Per al projecte actual, les dades necessàries seran, el "Head", la capacitat i la velocitat del compressor, i l'obertura de la vàlvula de control.

Per tant, s'ha de crear una DataTable que contengui aquestes quatre dades, i donar permisos de lectura, per a poder extreure aquesta informació del programa.

Per altra banda, també és necessari que el cas de simulació d'on es vagin a agafar les dades, no tingui cap error i tingui un controlador anti-surge. Això és necessari, ja que si les dades no són correctes, el model ens donarà respostes errònies, i no es podrà preveure correctament la malfunció del control.

Preparacions Intèrpret Python

Per aquest programa, es necessiten diverses llibreries (conjunts de funcions i classes prèviament creades) que no formen part de Python, per tant, si no s'instal·len, el programa no pot funcionar. Aquestes llibreries són les següents:

- matplotlib
- azure-iot-device
- azure-storage-blob
- numpy
- pandas
- pywin32
- colorama

Per evitar problemes amb la instal·lació dels paquets, s'utilitzarà la funció de "requirements.txt" amb la qual es pot instal·lar un conjunt de llibreries amb només un clic.

Aquesta funcionalitat ha sigut molt útil durant el desenvolupament del projecte, ja que com s'ha hagut de programar des d'una màquina virtual oferta per la UPC (Ravada), aquesta màquina virtual es reinicia cada vegada. Aquesta funció ajuda a estalviar molt de temps.

Preparacions Núvol

Per a utilitzar els serveis en el núvol d'Azure, és necessària una subscripció de pagament, Microsoft ofereix comptes per estudiants.

Un cop es posseeix un compte, és necessari crear els tres components esmentats en l'apartat 4.4 així com un clúster de procés.

Emmagatzematge

L'emmagatzematge emprat serà el Blob Storage, un emmagatzematge no estructurat que treballa amb Blobs. Un Blob és un arxiu per al núvol amb el qual es treballa de manera anàloga a com es guarden els arxius en un ordinador normal, de manera que un blob pot ser un arxiu de qualsevol mena.

Dins d'aquest blob storage, s'ha de crear un container, on es guardaran els arxius. En aquest container, les dades s'organitzaran segons el format amb el que es pengin. En el cas actual, serà endreçat per la data de pujada.

IoT Hub

S'ha de crear un IoT Hub, un broker de MQTT que rebrà les dades enviades, on es crea un dispositiu que simularà l'emissor de dades de la planta.

Aquest dispositiu s'ha de vincular amb un endpoint mitjançant l'encaminament de missatges. Aquest endpoint ha de ser l'emmagatzematge desitjat, això provocarà que les dades enviades per aquest dispositiu, amb el format indicat ("nomcarpeta/YY/mm/dd/hh/MM"), seran enviades i guardades en el Blob Storage.

Machine Learning

Azure té l'anomenat Microsoft Azure Machine Learning Studio. En aquesta aplicació, s'ha de crear un Workspace. Es pot crear de manera directa a través de la web o mitjançant un programa personalitzat amb Python.

Dins d'aquest servei, és necessari crear instàncies de procés, o clústers de procés, que serviran per a computar l'entrenament del model.

Una instància o clúster de procés és una infraestructura que permet a l'usuari executar treballs dins d'un entorn virtual ofert per Azure. Aquests serveis es paguen proporcionalment amb l'ús, per la qual cosa tenen un cost escalable dependent de les necessitats.

El prototip està format per tres principals funcions. La presa de dades, l'entrenament del model i la implementació d'aquest.

4.2.2. Presa de dades

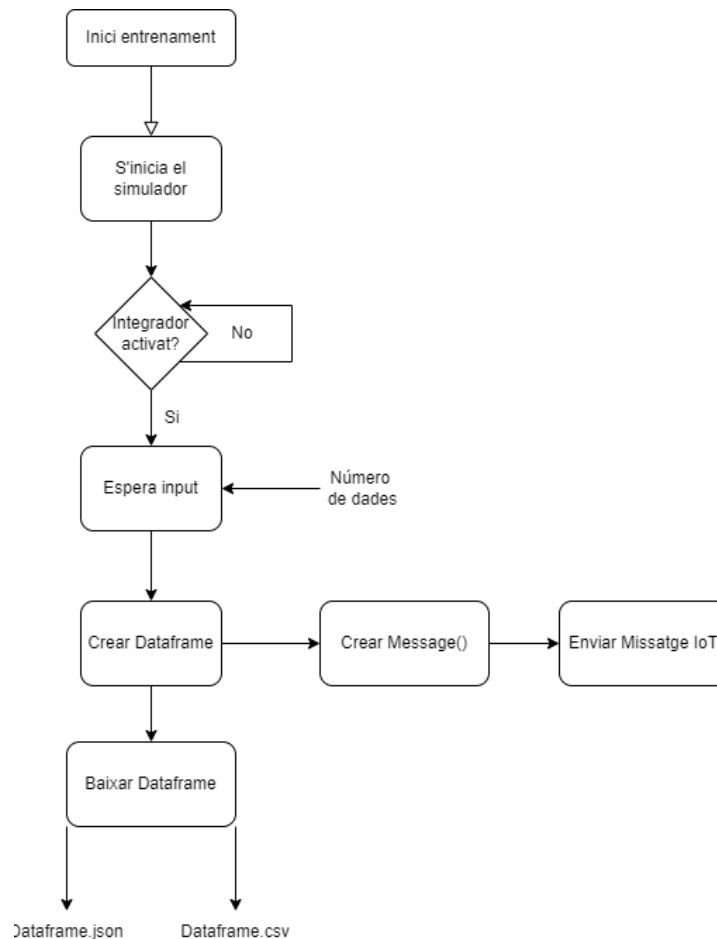


Figura 7 Diagrama presa de dades

Aquesta funció permet l'obtenció de dades amb les quals poder entrenar el model. Aquestes dades seran agafades d'unisim, però aquest simulador podria ser substituït per un conjunt de sensors en una planta real, sense haver de variar en gran manera el codi, i sense afectar a l'entrenament ni a la implementació del model. Això és degut a que està pensat per a treballar amb el protocol MQTT.

El primer que fa el programa és iniciar el simulador Unisim amb el cas desitjat. Aquest cas ha d'estar preparat com s'ha explicat prèviament.

Un cop iniciat el simulador, el programa espera fins que s'activa l'integrador, un cop activat s'espera un input amb el número de dades a agafar. A causa del protocol MQTT, el límit és de 256kB. Aleshores agafa dades cada segon fins a tenir el número de dades prèviament escollit. Tot i que el temps limita la quantitat de dades preses, si s'utilitza un simulador, es pot variar la relació temps real/ temps simulat. Això es pot aprofitar per simular l'evolució d'una planta al llarg de més temps.

Quan es tenen totes les dades, es crea un Dataframe (classe de taules de dades). Les dades recopilades acaben amb l'estructura següent:

Taula 1 Exemple Dataframe

Id.	Speed (rpm)	Head (N*m/kg)	Capacity (cfm)	ValveOpening (%)
0	1250	3281.16	2294.03	42.378
1	1250	3281.636	2292.466	42.342
2	1250	3282.537	2291.466	42.305

Aquest Dataframe serà el cos del missatge que s'enviarà mitjançant el protocol MQTT. Per utilitzar aquest protocol es crea un objecte de la classe missatge, afegint al format el nom de la carpeta on ha de guardar-se, afegint la data i l'hora on ha sigut creat.

Gràcies a això, les dades es guardaran endreçades en carpetes amb el format "nomcarpeta/YY/mm/dd/hh/MM"

Tot i que està fora de l'abast del projecte actual, amb aquest format es poden automatitzar la selecció de dades per a la creació dels models, ja que les dades queden guardades de manera estructurada i normalitzada.

Les dades rebudes i emmagatzemades al núvol seran Json amb el format que s'observa a la figura 8.

```
{
  "EnqueuedTimeUtc": "2022-05-09T18:01:07.5460000Z",
  "Properties": {},
  "SystemProperties": {
    "messageId": "IoTmartitena/Carpeta1/2022/05/09/20/00",
    "connectionDeviceId": "Test",
    "connectionAuthMethod": {
      "\scope": "\device",
      "\type": "\sas",
      "\issuer": "\iothub",
      "\acceptingIpFilterRule": null
    },
    "connectionDeviceGenerationId": "637841737696447055",
    "contentType": "application/json",
    "contentEncoding": "utf-8",
    "enqueuedTime": "2022-05-09T18:01:07.5460000Z"
  },
  "Body": [
    [12500.0, 11703.48182417543, 1049.5216796914524, 9.779580201678106],
    [12500.0, 11631.410121892031, 1062.0482413545292, 9.778229030417492],
    [12500.0, 11631.410121892031, 1062.0482413545292, 9.778229030417492],
    [12500.0, 10014.51577554296, 1168.1862517500313, 9.765074981882007],
    [12500.0, 10014.51577554296, 1168.1862517500313, 9.765074981882007],
    [12500.0, 10084.843782228068, 1157.3402680003292, 9.76622975602426],
    [12500.0, 10084.843782228068, 1157.3402680003292, 9.76622975602426],
    [12500.0, 10150.852598286183, 1147.0853533433187, 9.767316625040525],
    [12500.0, 10150.852598286183, 1147.0853533433187, 9.767316625040525]
  ]
}
```

Figura 8 Exemple missatge guardat al núvol

Aquest missatge no és apte perquè Azure l'entengui a l'hora de fer el Dataset, per tant, és necessari transformar-lo en un arxiu CSV.

Per fer això, des de Python es baixa l'arxiu, on s'aïlla el "body", i es transforma en un CSV, que torna a ser penjat a una carpeta d'Azure Storage.

4.2.3. Creació del model

Un cop es tenen les dades penjades en format CSV, s'ha de crear i entrenar el model. Per a això és necessari crear l'anomenat Dataset. Azure és capaç de crear aquests Datasets de manera semiautomàtica.

Selecciónant un o més arxius en format CSV, escollint el tipus de conjunt Tabular i dient que és un arxiu delimitat separat per comes, s'obté el Dataset. A la figura 9, es veu la configuració per aquest cas i el resultat.

Id.	Column1	Speed	Head	Capacity	ValveOpening
1	0	1250	3281,16	2294,03	42,378
2	1	1250	3281,636	2292,466	42,342
3	2	1250	3282,088	2291,05	42,305
4	3	1250	3282,537	2289,512	42,27

Figura 9 Configuració i resultat del Dataset

Un cop es té el Dataset llest, mitjançant l'eina AutoML, es crea un nou experiment, seleccionant el Dataset, tipus de tasca (en aquest cas un model de regressió) i la columna que volem que sigui l'objectiu de predicció.

AutoML ofereix diferents tipus de validació de models, incloent-hi un mètode automàtic, que detecta quina és la més indicada per la tasca. Per a avaluar el model, s'utilitzarà una partició de les dades aportades. La recomanació és fer servir entre el 10% i el 30% de les dades. S'ha seleccionat un 20%.

Amb la configuració feta, mitjançant un clúster de procés creat prèviament (apartat 4.2.1.) s'executa l'entrenament dels models.

Aquest procés pot durar hores, encara que el temps es pot reduir amb clústers de procés més potents, però també més cars.

Un cop acabada l'execució d'AutoML, es mostren els models que han aconseguit un menor error de desviació mitjana quadràtica. Es pot seleccionar el model desitjat per implementar-lo com a servei web.

Nombre del algoritmo	Explicado	Error de desviación media cuadrática normalizada ↑	Muestreo	Fecha de creación	Duración
VotingEnsemble	Ver explicación	0.00119	100.00 %	2 de jun. de 2022 19:00	4 s
<input checked="" type="radio"/> TruncatedSVDWrapper, ElasticNet	Ver explicación	0.00119	100.00 %	2 de jun. de 2022 18:09	4 s
StackEnsemble	Ver explicación	0.00121	100.00 %	2 de jun. de 2022 19:02	4 s

Figura 10 Implementació model

El procés d'implementació també pot durar hores, però un cop acabat es podrà consumir directament.

4.2.4. Utilitzar el model predictiu

Per utilitzar el model, només cal conèixer l'URL del model, i utilitzar una funció que ofereix el mateix Azure, editant aquesta funció es pot obtenir la predicció per utilitzar-la. En el cas d'aquest projecte, el que es fa és comparar la predicció de la posició de la vàlvula amb la posició de l'actuador de la vàlvula.

Això permet saber si la vàlvula està en la posició on l'actuador ens diu que està. Gràcies a aquesta comparació es pot saber si l'actuador està funcionant de la manera esperada.

A l'apartat 6.2. es mostra com és l'output del programa i com aquest utilitza el model.

5. Model matemàtic

Un model de Machine Learning es la sortida generada quan s'entrena un algoritme amb dades. Un cop generat el model, donada una entrada s'obté una sortida.

Els models necessaris per obtenir la predicció sobre l'obertura de la vàlvula seran models predictius, que utilitzaran algoritmes de regressió lineal, ja que el que es busca és la relació entre un conjunt de característiques i una variable objectiu continua. En aquest cas la variable objectiu serà un número real entre el 0 i el 100, ja que es busca el percentatge d'obertura d'una vàlvula.

5.1. Ajust d'un model

Entrenar un model implica trobar quins paràmetres del model minimitzen la diferència entre els valors predits i els valors observats.

Una funció d'ajust és una funció, normalment polinòmica, que passa prop del conjunt de dades donat.

No és necessari que aquest model passi per les dades, sinó que aporti la mínima diferència amb aquestes dades. Per valorar quin model és millor per a cada cas s'utilitzen diferents ajustos, un dels més utilitzats és l'ajust per mínims quadrats i certes variacions d'aquest.

5.2. Modelització en Azure

Azure actua com el que es coneix com a "caixa opaca", és a dir, l'usuari no pot conèixer els càlculs realitzats per a entrenar el model. El que sí que es pot conèixer són les desviacions, residus i R^2 , s'explica el cas concret del projecte a l'apartat 6.1.

5.3. Voting ensemble

Els algoritmes anomenats "Voting Ensemble" fan la mitjana de diferents models de regressió per poder obtenir una millor capacitat de predicció que amb un únic model.

Aquests models combinen diferents hipòtesis per a trobar una hipòtesis millor. Gràcies a això, es poden evitar biaixos generats al utilitzar una única hipòtesis. El problema d'aquest mètode, es que augmenta en gran mesura la capacitat de computació necessària, ja que augmenta el número de càlculs a fer. Pot ser útil per a aconseguir resultats més fiables en el cas on no hi hagi molta confiança en cap algoritme.

Degut a que en el cas d'aquest estudi l'error mitjà d'aquest model es pràcticament idèntic que l'error mitjà del TruncatedSVDWrapper/ElasticNet, es considera que es més eficient utilitzar aquest segon, ja que necessitarà menys computació i per tant menys diners i temps.

Per al nombre de dades utilitzat en el projecte, la diferència no serà notable, però si es volen crear models més extensos o amb més dades, l'ideal es reduir la computació necessària.

5.4. TruncatedSVDWrapper/ElasticNet

L'AutoML d'azure, ens indica que el model amb menys error és l'anomenat TruncatedSVDWrapper/ElasticNet.

Aquest mètode utilitza dos algoritmes diferents, el primer es TruncatedSVDWrapper, un algoritme de transformació de dades. Per a l'entrenament s'utilitza ElasticNet.

5.4.1. ElasticNet

Aquest model utilitza l'ajust de ElasticNet, que correspon a l'equació (1).

$$MIN\left[\frac{\sum_{i=1}^n \left(y_i - \left(\beta_0 + \sum_{j=1}^p \beta_j \cdot x_{ij}\right)\right)^2}{2n} + \lambda \left(\alpha \sum_{j=1}^p |\beta_j| + \frac{1-\alpha}{2} \sum_{j=1}^p \beta_j^2 \right)\right] \quad (1)$$

Aquest ajust s'explica més específicament a l'annex 2

5.4.2. TruncatedSVDWrapper

Aquest algoritme es el que es coneix com un algoritme de selecció de característiques, és a dir, s'encarrega de seleccionar d'un grup de característiques, quines són aquelles més rellevants. Per tant, d'un grup amb n objectes d'entrada, s'obtenen k objectes de sortida, sent $k < n$. ($k, n \in \mathbb{N}^*$)

Això serveix per a que l'algoritme d'entrenament pugui treballar a més velocitat, per a reduir la complexitat del model i el sobre ajustament.

Els mètodes "wrapper" és un procés de selecció on s'intenten totes les possibles combinacions de característiques i s'escull la millor seguint un criteri indicat per a cada cas. En el cas del projecte, s'utilitzarà com a criteri el mínim error de desviació mitjana quadràtica. [12]

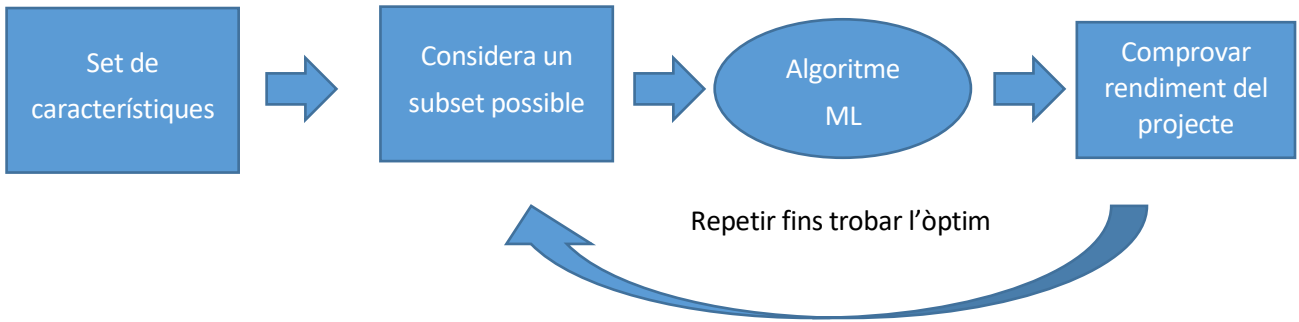


Figura 11 Diagrama de flux funcionament Wrapper

6. Validació

6.1. Mètriques del model

Azure ens ofereix diferents mètriques per a conèixer com d'ajustat està el model per al cas que ha sigut entrenat. Per al model utilitzat les mètriques obtingudes son les següents:

- Error de desviació quadràtica normalitzada = 0.0011935
- Error absolut mitjà = 0.00747802
- $R^2 = 0.99959$

Les mètriques semblen prou bones, ja que tenen un error molt baix i un R^2 molt proper a 1

També es pot veure la gràfica de residus (diferència entre la predicció i les dades correctes) a la figura 12.

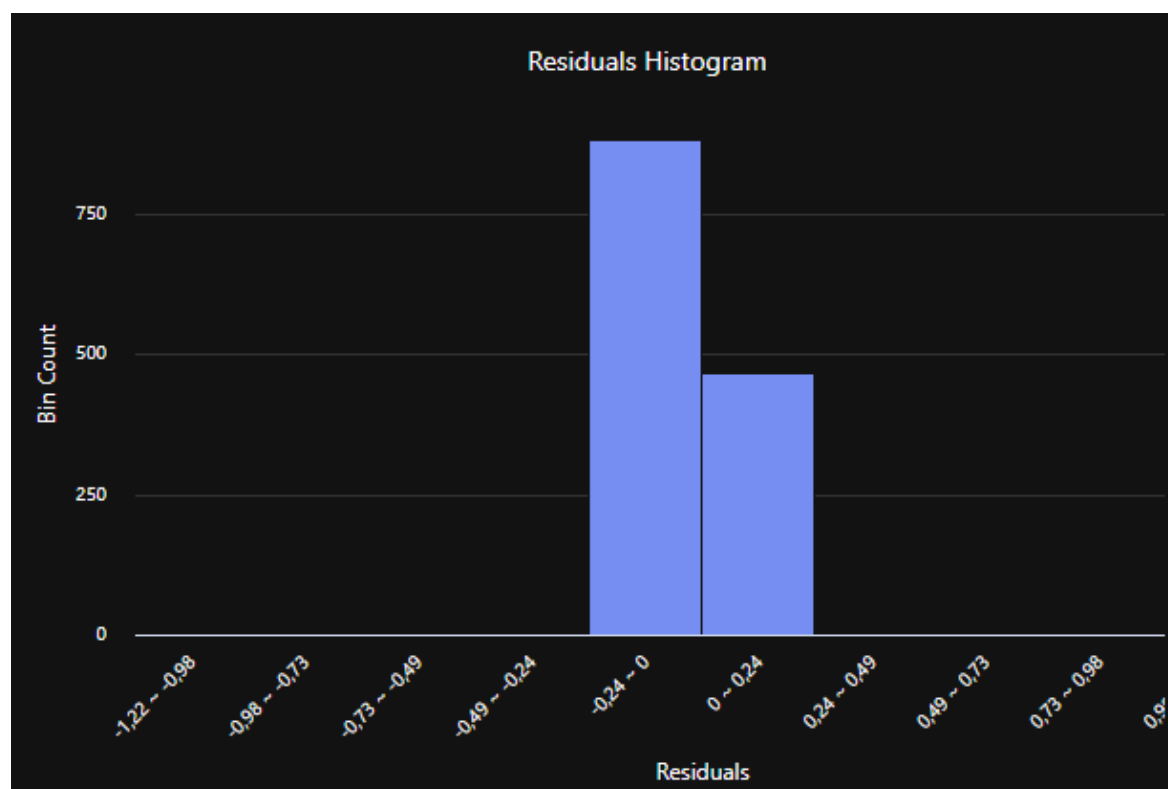


Figura 12 Histograma de residus del model

Es pot veure com els residus estan compresos entre -0.24 i +0.24. Sembla que no hi ha cap biaix que provoqui que totes les prediccions estiguin desplaçades, ja que sembla que tenen una distribució bastant equitativa al voltant del 0.

6.2. Comprovació pràctica

Per comprovar la utilitat del model, s'ha utilitzat el mateix cas simulat que en l'entrenament, però per a simular les fallades en la vàlvula, s'utilitzarà l'opció de "malfunction" que ofereix Unisim.

The screenshot shows the configuration window for a valve upset simulation. On the left, under 'Upset Types', several radio buttons are listed: 'Actuator Failure' (selected), 'Positioner Offset', 'Leakage', 'Plugging Factor', 'Stem Inertia', 'Sticky Stem', and 'Positioner Hunting'. Below this, there are two checkboxes: 'Use Malfunction' (checked) and 'Use as Plugging Restriction Only' (unchecked). On the right, the 'Description' field is empty. The 'Status' section shows 'Valve Stem Status' with a dropdown menu currently set to 'Normal'. The 'Upset Details' section includes: 'Fail Position' with a dropdown set to 'Fail Open', 'Duration' with a dropdown set to 'Hold', 'Delay Time' with a text input field containing '000:00:00.00', and 'ESD Trip State' with a dropdown set to 'Fail Shut'. An 'Enable Upset' button is located at the bottom right of the configuration area.

Figura 13 Opcions per a simular errors en la vàlvula

6.2.1. Funcionament correcte

Quan el funcionament de l'actuador i la vàlvula està funcionant correctament, s'espera que la predicció i l'obertura de la vàlvula siguin molt similars.

El programa ofereix el resultat de manera gràfica i numèrica.

De manera gràfica, es mostra un gràfic en temps real que mostra l'evolució de l'estat predit i de les dades experimentals. S'observa un exemple en la figura 14.

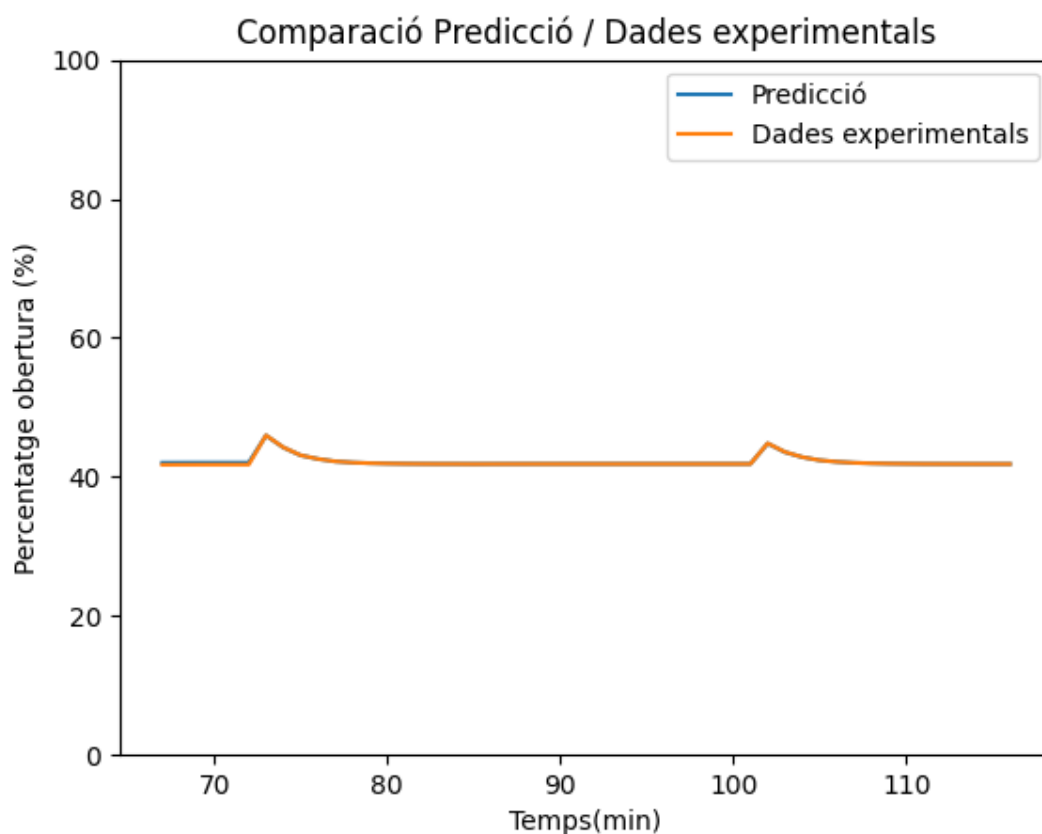


Figura 14 Gràfic amb funcionament correcte

S'han aplicat canvis a l'estat en els instants $t \approx 74$ i $t \approx 105$, canviant lleugerament la velocitat del compressor per a generar una alteració en l'estat.

S'observa que la predicció és prou propera a la posició real de la vàlvula.

De manera numèrica, es mostra la diferència calculada entre la predicció i la posició real en cada instant. Com es veu en la figura 15.

```
Diferencia Predicció/Real = 0.11447825735493922
Diferencia Predicció/Real = 0.022053145292829868
Diferencia Predicció/Real = 0.025116213842309776
Diferencia Predicció/Real = 0.014563768742192451
Diferencia Predicció/Real = 0.006239623911973524
```

Figura 15 Output numèric de sortida funcionament correcte

Avaluar quin percentatge significaria un error a solucionar és feina de l'operari de la planta, qui coneix quins valors són acceptables per al funcionament correcte.

S'ha habilitat una simple línia de codi amb la que es pot escollir a partir de quin percentatge de diferència es considera com error. Quan se supera aquest valor el missatge es veurà en vermell, com a la figura 16.

```
Diferencia Predicció/Real = 15.63432031789663
Diferencia Predicció/Real = 15.36903901074929
Diferencia Predicció/Real = 15.201412500735415
Diferencia Predicció/Real = 15.103028391037903
Diferencia Predicció/Real = 15.054830882893867
```

Figura 16 Output numèric de sortida incorrecte

6.2.2. Funcionament incorrecte

Per comprovar que es detectaria de manera correcta un error de funcionament, se simulen diferents fallides.

Alguns d'aquests errors poden portar al sistema a estats no entrenats pel model, el que pot provocar que les prediccions no siguin molt fiables. Això es pot solucionar entrenant el model amb més dades, o amb dades que representin estats extrems.

Es pot considerar que en aquests casos l'error és notable, i, per tant no es precisa una predicció tant exacta.

Offset

Un error possible és un offset, és a dir, quan la vàlvula té un desplaçament respecte a la posició indicada.

Simulant un offset de 20% el programa respon de la manera següent.

```
Diferencia Predicció/Real = 20.00742303247876
Diferencia Predicció/Real = 20.007537374867557
Diferencia Predicció/Real = 20.0072550782068
Diferencia Predicció/Real = 20.00726931299545
```

Figura 17 Output numèric offset 20%

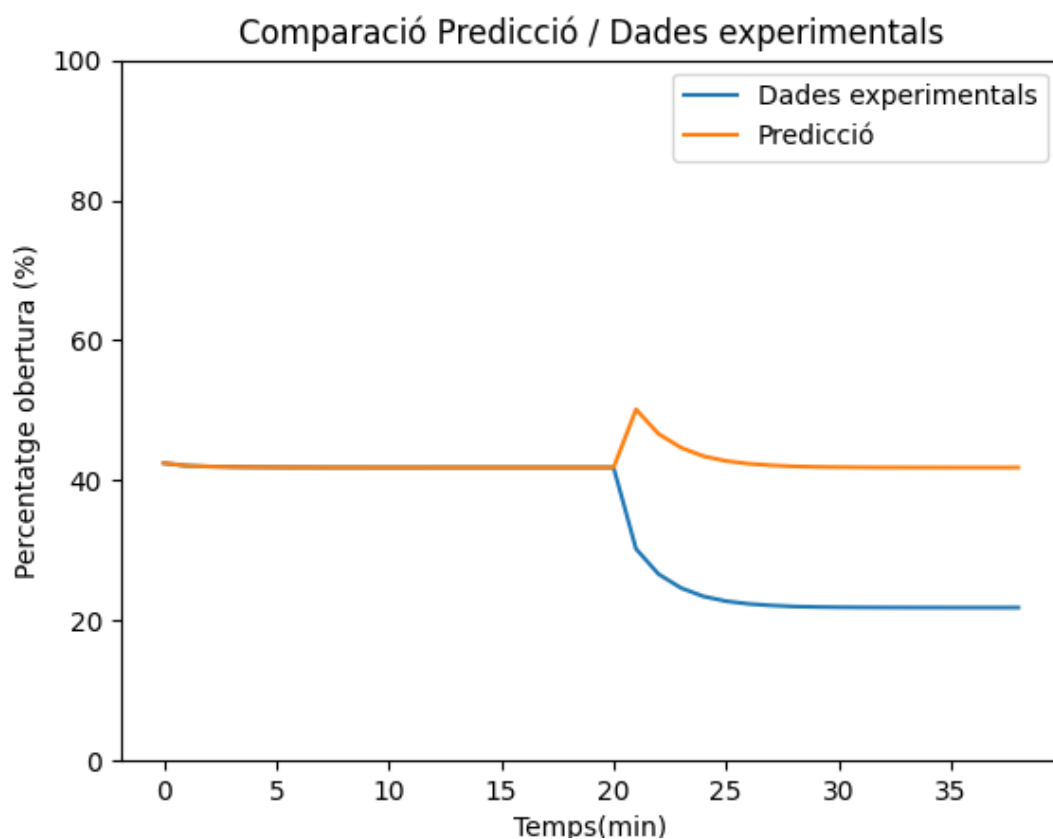


Figura 18 Gràfic amb Offset del 20%

Es veu com apareix la diferència del 20% degut a l'error que hem provocat.

Error d'obertura

Un error possible és que la vàlvula es quedi oberta completament. Donant els resultats següents.

```
Diferencia Predicció/Real = 13.257000000000005
Diferencia Predicció/Real = 13.275999999999996
Diferencia Predicció/Real = 13.289000000000001
Diferencia Predicció/Real = 13.296000000000006
```

Figura 19 Output numèric error vàlvula oberta

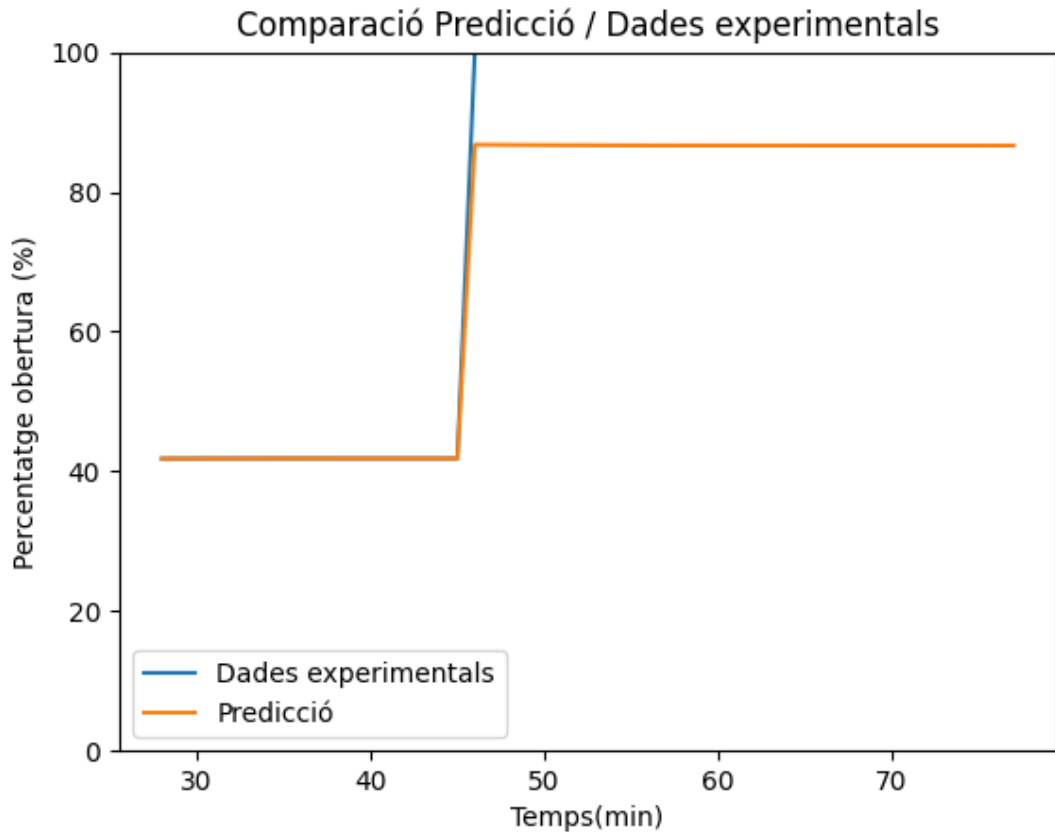


Figura 20 Gràfica error vàlvula oberta

Error vàlvula no respon

Quan l'actuador de la vàlvula deixa de funcionar, moltes vegades es manté la mateixa posició que tenia quan l'actuador falla. Això provoca que si l'estat de la planta varia, la vàlvula no modificarà la seva posició.

En el següent gràfic es veu com la predicció si que varia però la posició de la vàlvula no. En l' instant $t \approx 17$ es modifica la velocitat del reactor de 1200 a 1300 rpm.

```
Diferencia Predicció/Real = 1.081783878919829
Diferencia Predicció/Real = 1.081783878919829
Diferencia Predicció/Real = 3.2027838789198313
Diferencia Predicció/Real = 3.2327838789198324
```

Figura 21 Output numèric error vàlvula no respon

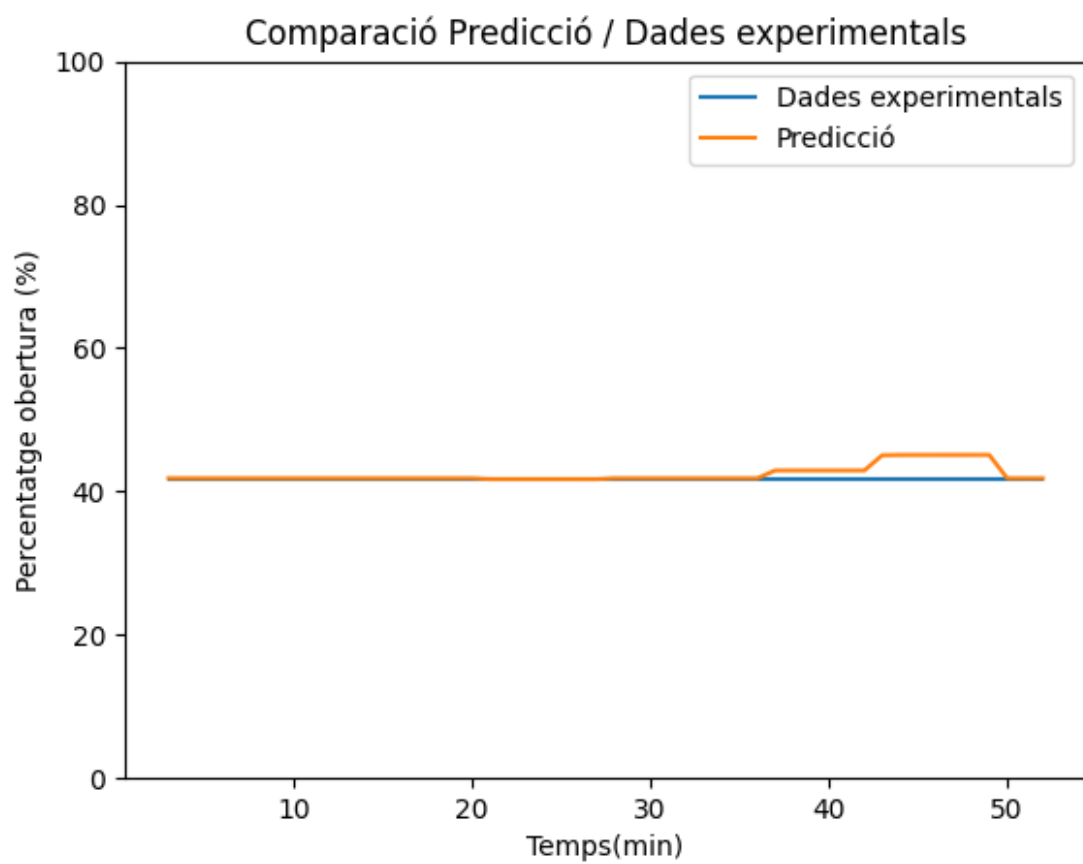


Figura 22 Gràfic error vàlvula no respon

7. Anàlisi de l'impacte ambiental

Aquest projecte només ha utilitzat eines informàtiques, per tant, no es generen residus que es puguin tenir un impacte ambiental considerable. L'ús d'ordinadors així com el serveis en el núvol consumeixen electricitat i, per tant, certa contaminació. Aquest consum es pot considerar menyspreable en la magnitud usada durant el projecte.

El prototip creat en aquest projecte està destinat a evitar fallades en plantes reals, on la utilització del model creat durant el treball o algun model similar pot evitar l'emissió de productes químics i la pèrdua d'aquests. Per tant, es pot considerar que l'aplicació creada seria beneficiosa per al medi ambient.

Conclusions

A la finalització d'aquest treball s'ha pogut concloure el següent:

- S'ha pogut crear una aplicació capaç de detectar la fallada de l'actuador d'una vàlvula anti-surge en un simulador de processos químics dinàmic.
- S'han comparat diferents eines amb les que realitzar l'objectiu del treball i s'han escollit les següents.
 - Unisim, com a simulador de processos
 - Python, com a llenguatge de programació
 - Microsoft Azure, per a servei al núvol
- S'ha aconseguit la connexió i l'obtenció de dades d'una planta simulada amb Unisim mitjançant el llenguatge de programació Python.
- S'ha desenvolupat un programa capaç de recollir les dades d'una planta simulada, i tractar-les de manera que siguin útils per a la creació de models predictius.
- S'ha creat un model de predicció mitjançant l'aprenentatge automàtic capaç de predir la posició real de la vàlvula en cada moment a partir de les dades obtingudes sobre l'estat de la planta simulada.
- S'ha pogut demostrar la utilitat que tenen aquests models per a entendre un sistema en temps real.
- S'han utilitzat els serveis en el núvol d'Azure, per l'emmagatzematge, la recepció de dades, la creació del model i la implementació d'aquest.
- S'han inclòs protocols de comunicació MQTT, amb els que es facilita una possible aplicació en una planta real.
- Aquest projecte podrà ser utilitzat com a exemple per a crear diferents aplicacions, amb les que detectar fallades en diferents elements d'una planta.

Pressupost i/o Anàlisi Econòmica

En aquest apartat s'hi ha d'incloure el pressupost desglossat i agrupat per capítols, incloent la mesura dels mitjans emprats i la valoració econòmica dels mateixos (costos d'enginyeria, mà d'obra, materials, etc.).

El pressupost d'aquest projecte té 3 grups de costos:

- Els costos materials, en aquest cas només el ús de l'ordinador durant el transcurs del projecte.
- El cost de les llicències dels programes utilitzats.
- El cost dels serveis d'Azure utilitzats.

Per al temps d'ús de les llicències i de l'ordinador s'ha calculat 5 mesos, i tot i el projecte no ha sigut l'únic ús que se li ha donat, ho suposarem així per a simplificar els costos

Cost Ordinador

S'ha utilitzat l'ordinador personal, el qual es va comprar per 1500€, i amb una vida total d'aproximadament 10 anys.

Cost Llicències

- Microsoft Windows
- Microsoft Office
- Unisim Design

Cost serveis Azure

Microsoft Azure ofereix una eina de control de pagaments i despeses. Mitjançant aquesta eina podem obtenir els costos desglossats entre els diferents serveis.

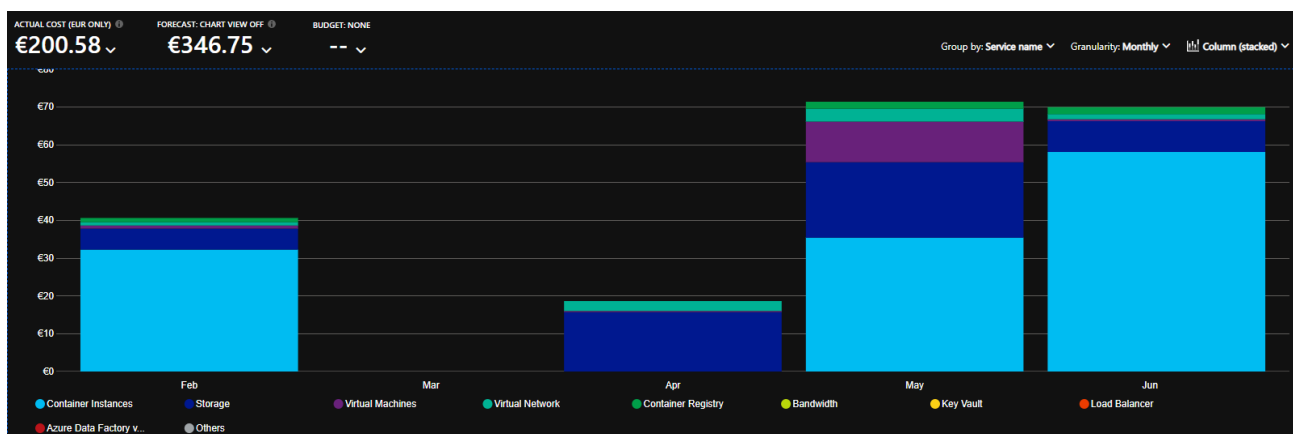


Figura 23 Costos Azure desglossats per servei i temps

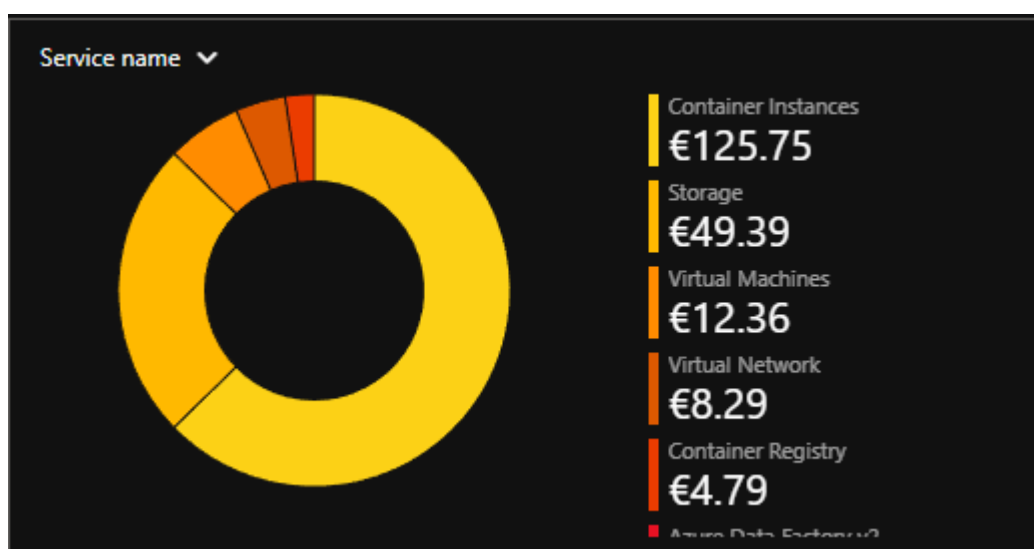


Figura 24 Costos Azure desglossats per servei

S’observa que el cost ve principalment del que Azure anomena Container Instancies. Això es refereix als processos executats en els clústers de procés explicats en l’apartat 5.2.1.

L’Storage, també té un cost important, referit a l’emmagatzematge de dades.

Cost total

Taula 2 Costos projecte

Element	Cost Total (€)	Vida útil
Ordinador	1400	10 anys
Microsoft Office	100	1 any

Microsoft Windows	145	5 anys
Microsoft Azure	200	-
Honeywell Unisim	20000	-
Total	21845	-

El cost total de la realització del projecte és d'aproximadament 22000€, tot i que no és valorada la amortització de l'ordinador i de la llicència d'Unisim. Per tant, amb aquesta inversió es podrien realitzar futurs projectes sense haver de comprar la llicència o l'ordinador.

Bibliografia

- [1] Vice. “The Third Industrial Revolution: A Radical New Sharing Economy.” YouTube. YouTube, February 13, 2018. <https://www.youtube.com/watch?v=QX3M8Ka9vUA>.
- [2] Bilbao, Luis Maria, and Ramon Lanza. HISTORIA ECONÓMICA. Madrid: Universidad Autónoma de Madrid. Departamento de Análisis Económico: Teoría Económica e Historia Económica, 2009.
- [3] Hermann, Mario, Tobias Pentek, and Boris Otto. “Design Principles for Industrie 4.0 Scenarios.” 2016 49th Hawaii International Conference on System Sciences (HICSS), 2016. <https://doi.org/10.1109/hicss.2016.488>.
- [4] “¿Qué Es La Informática En La Nube? Guía Para Principiantes: Microsoft Azure.” Guía para principiantes | Microsoft Azure. Accessed June 14, 2022. <https://azure.microsoft.com/es-es/overview/what-is-cloud-computing/>.
- [5] TomReidNZ. “Introduction to Machine Learning - Learn.” Learn | Microsoft Docs. Accessed June 14, 2022. <https://docs.microsoft.com/en-us/learn/modules/introduction-to-machine-learning/>.
- [6] “Chemengguy.” ChemEngGuy. Accessed June 14, 2022. <https://www.chemicalengineeringguy.com/the-blog/process-simulation/what-is-unisim/>.
- [7] “IBM Bluemix and the Internet of Things - Workshop.” SlideShare a Scribd company. Accessed June 14, 2022. https://www.slideshare.net/gjuljo/ibm-bluemix-and-the-internet-of-things-workshop?qid=e4d489ba-e587-4d38-b479-0f83931fa3f7&v=&b=&from_search=4.
- [8] “Química.” compair. Accessed June 14, 2022. <https://www.compair.com/es-es/industries-and-applications/chemical>.
- [9] “Compresor (Máquina).” Wikipedia. Wikimedia Foundation, March 24, 2022. [https://es.wikipedia.org/wiki/Compresor_\(m%C3%A1quina\)](https://es.wikipedia.org/wiki/Compresor_(m%C3%A1quina)).
- [10] Widman, Richard, and Omar Linares. “Compresores – Su Funcionamiento y Mantenimiento.” Accessed June 14, 2022. https://widman.biz/boletines_informativos/56.pdf.
- [11] Atlas Copco. “Dos Principios Básicos De Compresión: Compresión De Desplazamiento Y Compresión Dinámica.” Atlas Copco. Accessed June 14, 2022. <https://www.atlascopco.com/es-es/compressors/wiki/compressed-air-articles/displacement-and-dynamic-compression>.
- [12] “Feature Selection Using Wrapper Method - Python Implementation.” Analytics Vidhya, December 29, 2020. <https://www.analyticsvidhya.com/blog/2020/10/a-comprehensive-guide-to-feature-selection-using-wrapper-methods-in-python/>.
- [13] Sanz, Gerard. “Desarrollo De Un Control Anti Surge Usando Redes Neuronales En Una Planta Virtual.” Universitat Politècnica de Catalunya, January 2022.
- [14] Martínez Lopez, Kevin “DEVELOPMENT OF MACHINE LEARNING STRATEGIES FOR FAULT DIAGNOSIS IN VIRTUAL PLANTS (DIGITAL TWINS)” Universitat Politècnica de Catalunya



Annex 1 Codi Programa

Classe Base

```
class BaseTFG():

    def __init__(self, parent=None):
        filepath = "i"
        self.obrir_simulador()

    def obrir_simulador(self):
        self.filepath = r"C:\Users\vdi.eebe\Desktop\FinalClases-
FINAL\C1.usc"
        self.unisimpath = r'C:/Program Files (x86)/Honeywell/UniSim
Design R481/unisimdesign.tlb'
        ConexioUnisim.obj = ConexioUnisim.UnisimConnection(self.filepath,
self.unisimpath)
        self.dadestotals = 0
        ConexioUnisim.unisim = ConexioUnisim.obj.OpenCase()
        x = 0
        while x < 1:
            if ConexioUnisim.unisim[0].Solver.Integrator.IsRunning:
                x = 1
        self.agafardades()
        """def comprovarActivacio(self):
if ConexioUnisim.unisim[0].Solver.Integrator.IsRunning:
    self.BotoAgafarDades.setEnabled(True)"""

    def agafardades(self):
        # self.Pantalles.setCurrentIndex(2)
        print("Numero maxim de dades (*100)")
        self.maxim_dades = int(input()) * 100
        print("MaximDades: " + str(self.maxim_dades))
        self.LlegirTaula()
        self.crear_llista_tuples()

    def hora(self):
        return datetime.now()

    def LlegirTaula(self):
        TableDict = ConexioUnisim.obj.ReadDataTable('ProcData1')

    def crear_llista_tuples(self):
        self.lista = []
        lista_valores_capacity = []
        lista_valores_head = []
        surgesigno = []
        stonewallsigno = []
        zonatrabajo = []
        lista_valores_corregida = []
```

```

lista_valor_aperutra_valvula = []
lista_valores_corregida_ok = []
self.hora_inici = self.hora()

while ConexioUnisim.unisim[0].Solver.Integrator.IsRunning:
    data = ConexioUnisim.unisim[0].DataTables('ProcData1')
    time.sleep(1)
    values = data.GetAllValues()
    self.lista.append(values)
    # self.Pantalles.setCurrentIndex(2)
    """
    Me dice distancias a la que estan y si es zona buena
    d=list(zip(surgesigno,stonewallsigno,zonatrabajo))
    dok=d[-1]
    """
    self.lista_valores_corregida_ok = lista_valores_corregida[-1]

"""self.Nomtaula.setText(str(self.lista_valores_corregida_ok))"""
print(self.lista_valores_corregida_ok)

self.diferenciadetemps = self.hora() - self.hora_inici
print(self.diferenciadetemps)
if self.diferenciadetemps >= timedelta(seconds=600):

    print("Llista feta")
    self.actualitzarDades()
    self.crearDataFrame()

    break

    """if len(self.lista) >= self.maxim_dades:
    print("Llista feta")
    self.crearDataFrame()
    break"""

def actualitzarDades(self):
    self.dadestotals += len(self.lista)
def crearDataFrame(self):
    # Converteix la llistat de tuples a DataFrame
    data = self.lista
    self.df = pd.DataFrame(data, columns=['Speed', 'Head',
'Capacity', 'ValveOpening'])
    self.dades_json = json.dumps(data)
    print("DataframeCreada")
    self.baixarDF()

def baixarDF(self):
    self.TextDirBaixar = r"C:\Users\vdi.eebe\Desktop\Dataframe"
    self.df.to_json(self.TextDirBaixar + ".json")

```

```
self.df.to_csv(self.TextDirBaixar + ".csv")
print("Baixat")

asyncio.run(self.penjarDF(self.dadestotals))

async def penjarDF(self, dt):
    time.sleep(1)
    await MQTT.enviarDF(self.dades_json)
    print("json enviat")
    #await MQTT.enviarDF(self.df)
    #print("CSV enviat")
    """filename = "Dataframe.csv"
    PenjarTxt.PenjarArxiuAzure.PenjarUnJson(filename, filename)"""
    print(dt)
    if dt<=self.maxim_dades:
        default_loop = asyncio.get_running_loop()
        default_loop.stop()
        self.LlegirTaula()
        self.crear_llista_tuples()
def enviarMissatge(self):
    return
```

Classe Connexió Unisim

```
class UnisimConnection(object):
    ''' All the functions related to Unisim communication '''

    # Open Unisim Case
    def __init__(self, filepath, unisimpath):
        '''
        Set all the global variables and do the conversion for C classes
        to Python language.
        The CLSID may be wrong because this script was written for Unisim
        in case is wrong we can look for the good ones
        '''
        self.casepath = filepath
        self.uniapp =
win32com.client.Dispatch("UnisimDesign.Application.NewInstance")
        self.simcase = self.uniapp.SimulationCases.Open(filepath)
        ## Call to Unisim backdoor class on the register
        self.BackDoor = makepy.gencache.GetClassForCLSID("{red}")
        # Define class for FixedDataTables, Datatalbe and VarDefinition
lybraries
        self.VarDefinition = makepy.gencache.GetClassForCLSID("{B37CCBF0-
E29E-11D0-A418-00A0C923285C}")
        sys.argv = ["makepy", unisimpath]
        makepy.main()
        return

    def OpenCase(self):
        ''' Open Hysys case & Activate the select case '''
```

```

self.uniapp.Visible = False
self.simcase.Activate()
return [self.simcase, self.uniapp]

def KillUnisim(self, simCase=''):
    ''' Kill Hysys at the end of the run or when somethin goes wrong
'''
    if self.simcase != '':
        self.simcase.Close(False)
    return self.uniapp.Quit()

def ReadDataTable(self, TableName):
    ''' Read Information from Process data table and convert it to a
python dictionary'''
    # Look for the Process Data Table of the Unisim case
    ProcessDataTable = self.simcase.DataTables(TableName)
    ProcessDataTable.StartTransfer()
    vd = ProcessDataTable.GetAllVarDefinitions()
    # Extract all Tags info and save it into a dict
    DataTable = {}
    idict = {}
    for x, value in enumerate(vd):

        vd = self.VarDefinition(value)
        idict[vd.Tag] = vd
    ProcessDataTable.EndTransfer()
    DataTable[TableName] = idict
    return DataTable

def WriteTagsDataTable(self, Values, dataTable_name):
    '''Write all tags to selected process data table'''
    # try:
    # ProcessDataTable.StartTransfer()
    # Write all the values back to the ProcessDataTable
    ProcessDataTable = self.simcase.DataTables(dataTable_name)
    ProcessDataTable.StartTransfer()
    getvalues = ProcessDataTable.GetAllVarDefinitions()
    for count, value in enumerate(getvalues):
        tempDict = getvalues[count]
        dictindex = 0
        vd = self.VarDefinition(value)
        if (vd.accessMode == 3 or vd.accessMode == 2): # Write and
Write/Read
            if not (np.isnan(Values)):
                vd.Variable.SetValue(Values, vd.Units)
                dictindex += 1
    ProcessDataTable.EndTransfer()

```


EnviarDF

```
path_to_root_cert = '/MQTT/certificat/digicert.crl'  
device_id = "Test"  
sas_token = "HostName=IoTmartitena.azure-  
devices.net;DeviceId=Test;SharedAccessSignature=SharedAccessSignature  
sr=IoTmartitena.azure-  
devices.net%2Fdevices%2FTest&sig=BYrf8836W98%2ByYQ3ffINmbWU%2Fj7HlH%2FXM  
XvyfJjM7A%3D&se=1648649435"  
iot_hub_name = "IoTmartitena"
```

```
async def enviarDF(miss):  
    time.sleep(1)  
    missatge=ferclasseMissatge(miss)  
  
    print("tipus missatge:")  
    print(missatge.content_type)  
    print("data:")  
    print(missatge.data)  
    time.sleep(1)  
    print("si")  
    await client.send_message(missatge)  
  
    print("fet")
```

```
def ferclasseMissatge(missatge):  
    now = datetime.now()  
    current_time = now.strftime("%Y/%m/%d/%H/%M")  
    formatCorrecte=r"IoTmartitena/Carpeta1/"+ current_time  
    miss=azure.iot.device.Message(missatge,message_id=formatCorrecte,content_  
encoding="utf-8",content_type="application/json")  
    return miss
```

Connectar Workplace

```
subscription_id = os.getenv("SUBSCRIPTION_ID", default="0c8e38d9-e695-  
490a-a4db-580845967383")  
resource_group = os.getenv("RESOURCE_GROUP", default="GrupRecursos")  
workspace_name = os.getenv("WORKSPACE_NAME", default="Workplace")  
workspace_region = os.getenv("WORKSPACE_REGION", default="West Europe")  
  
from azureml.core import Workspace  
def conectarWorkplace():  
    try:  
        ws = Workspace(subscription_id = subscription_id, resource_group  
= resource_group, workspace_name = workspace_name)  
        # write the details of the workspace to a configuration file to
```

```

the notebook library
    ws.write_config()

    print("Workspace configuration succeeded. Skip the workspace
creation steps below")
    return ws
except:
    print("Workspace not accessible. Change your parameters or create
a new workspace below")

ws=conectarWorkplace()
print(ws.webservices)

```

CrearCSVdeJson:

```

def CrearCSVdeJson():
    STORAGEACCOUNTURL = "https://carpetamarti.blob.core.windows.net"
    STORAGEACCOUNTKEY =
"eZNPpRpl/V+WLowolzcmDFLXrE/f0b7ZoKfU7i3mXm0toaBd0kBF3CbcTJAmQPM9lSIhGGr8K
MUXpFq+iCdEHeQ=="
    CONTAINERNAME = "rebutiot"
    BLOBNAME = "IoTmartitena/01/2022/06/02/14/12.json"

    blob_service_client_instance = BlobServiceClient(
        account_url=STORAGEACCOUNTURL, credential=STORAGEACCOUNTKEY)

    blob_client_instance = blob_service_client_instance.get_blob_client(
        CONTAINERNAME, BLOBNAME, snapshot=None)
    file_name=
r"C:\Users\mtt99\OneDrive\Escritorio\Eebe\2022\TFG\FinalClases\Baixades\A
rxiuPerDataFrame03.csv"
    blob_data = blob_client_instance.download_blob()
    data_bytes = blob_data.readall()
    data_string = data_bytes.decode("utf-8")
    data_json = json.loads(data_string)
    df = pd.DataFrame(data_json["Body"], columns=['Speed', 'Head',
'Capacity', 'ValveOpening'])
    datacsv = df.to_csv(file_name, encoding='utf-8')
    penjador = PenjarArxiuAzure()
    PenjarArxiuAzure.PenjarTotaCarpetaCSV(penjador)

```

Utilitzar Predicció:

```

class CompararResultats():

    def __init__(self, parent=None):
        filepath = "i"
        """self.x = np.linspace(0, 10, 100)
        self.y = np.linspace(9.76,9.8,100)

        plt.ion()

```

```
self.figure, ax = plt.subplots(figsize=(30, 6))
self.line1, = ax.plot(self.x, self.y)
plt.title("Dynamic Plot of sinx", fontsize=25)
plt.xlabel("X", fontsize=18)
plt.ylabel("sinX", fontsize=18)"""
self.obrir_simulador()

def obrir_simulador(self):
    self.filepath = r"C:\Users\vdi.eebe\Desktop\FinalClases-
FINAL\C2.usc"
    self.unisimpath = r'C:/Program Files (x86)/Honeywell/UniSim
Design R481/unisimdesign.tlb'
    ConexioUnisim.obj = ConexioUnisim.UnisimConnection(self.filepath,
self.unisimpath)
    self.dadestotals = 0
    ConexioUnisim.unisim = ConexioUnisim.obj.OpenCase()
    x = 0
    while x < 1:
        if ConexioUnisim.unisim[0].Solver.Integrator.IsRunning:
            x = 1
    self.agafardades()
    """def comprovarActivacio(self):
if ConexioUnisim.unisim[0].Solver.Integrator.IsRunning:
    self.BotoAgafarDades.setEnabled(True)"""

def agafardades(self):
    # self.Pantalles.setCurrentIndex(2)
    """print("Numero maxim de dades (*100)")
self.maxim_dades = int(input()) * 100
print("MaximDades: " + str(self.maxim_dades))"""
    self.LlegirTaula()
    self.AgafarDades()

def hora(self):
    return datetime.now()

def LlegirTaula(self):
    TableDict = ConexioUnisim.obj.ReadDataTable('ProcData1')

def AgafarDades(self):
    self.iteracio=0
    self.lista = []
    self.llistaresultats = []
    while ConexioUnisim.unisim[0].Solver.Integrator.IsRunning:
        data = ConexioUnisim.unisim[0].DataTables('ProcData1')
        time.sleep(1)
        self.values = data.GetAllValues()
        self.lista.append(self.values)
        # self.Pantalles.setCurrentIndex(2)
```

```

        self.iteracio+=1
        Predicció =
Utilitzar_model.compararDades(self.values[0],self.values[1],self.values[2
])
        Resultats=[self.values[3],Predicció]
        self.llistaresultats.append(Resultats)
        dataframe
=pd.DataFrame(self.llistaresultats,columns=["Actual","Predicció"])
        #dataframe.plot(x="Temps",y=['Predicció',"Actual"], kind =
'line')
        if len(self.llistaresultats)<= 10:
            plt.plot(dataframe)
        else:
            plt.plot(dataframe[-10:])
            plt.pause(0.05)
            plt.cla()
            diferencia = abs(Predicció-self.values[3])

            #self.fer_grafica(predicció=Predicció,valor=self.values[3])
            if diferencia < 0.0070:
                print("Diferencia Predicció/Real = " + str(diferencia))
            else:
                print("Diferencia Predicció/Real = " + Fore.RED
+str(diferencia) + Fore.WHITE)

if __name__ == '__main__':
    MyWindow = CompararResultats()

```

Utilitzar Model:

```

def allowSelfSignedHttps(allowed):
    # bypass the server certificate verification on client side
    if allowed and not os.environ.get('PYTHONHTTPSVERIFY', '') and
getattr(ssl, '_create_unverified_context', None):
        ssl._create_default_https_context =
ssl._create_unverified_context

allowSelfSignedHttps(True) # this line is needed if you use self-signed
certificate in your scoring service.

def compararDades(speed,head,capacity):
    data={
        "Inputs": {
            "data": [
                {
                    "Column2": "example_value",
                    "Speed": speed,

```

```
        "Head": head,
        "Capacity": capacity
    }
]
}

}

body = str.encode(json.dumps(data))

url = 'http://30bee71f-fce2-47b4-8d79-
01b91b97f986.westeurope.azurecontainer.io/score'
api_key = '' # Replace this with the API key for the web service
headers = {'Content-Type':'application/json',
'Authorization':('Bearer '+ api_key)}

req = urllib.request.Request(url, body, headers)

try:
    response = urllib.request.urlopen(req)

    result = response.read()
    string=str(result)

numero_resultat=float(str(string[15]+string[16]+string[17]+string[18]+str
ing[19]+string[20]))
    print(numero_resultat)
    return numero_resultat

except urllib.error.HTTPError as error:
    print("The request failed with status code: " + str(error.code))

    # Print the headers - they include the request ID and the
timestamp, which are useful for debugging the failure
    print(error.info())
    print(error.read().decode("utf8", 'ignore'))
```

Annex 2 Explicació Model ElasticNet

ElasticNet

ElasticNet es un model de regressió lineal. Per a entendre el funcionament d'Elasticnet, es necessari conèixer els conceptes de Lasso, Ridge i regressió.

Regressió lineal

Un model de regressió lineal considera que donat un conjunt d'observacions, la variable de sortida "Y_i" es relaciona de forma amb les variables de explicatives X₁, X₂...X_p d'acord amb l'equació 2.

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip} + \epsilon \quad (2)$$

També es pot escriure com:

$$y_i = \beta_0 + \sum_{j=1}^p \beta_j \cdot x_{ij} \quad (3)$$

β_0 = ordenada a l'origen, valor mitjà de la sortida "Y" quan totes les variables de regressió son 0

β_j = paràmetres del model, mesuren la influencia de les variables explicatives

ϵ = residu o error, recull el efecte de totes les variables que existeixen però no es recullen en el model

Els valors de β es desconeixen

Ajust per mínims quadrats

Usualment, per a valorar la qualitat del model, s'utilitza l'ajustament per mínims quadrats.

Aquest ajustament valora com a millor model aquell que té la menor suma de desviacions entre cada dada i la recta, elevades al quadrat.

$$MIN[\sum_{i=1}^n e^2] = MIN[\sum_{i=1}^n (y_i - \hat{y})^2] \quad (4)$$

Regularització

Les estratègies de regularització son penalitzacions que se li apliquen a l'ajust per mínims quadrats per a minimitzar el sobre ajustament, reduir la variància, atenuar l'efecte de la correlació entre predictors i minimitzar l'influència en el model dels predictors menys importants.

La regularització Lasso

Lasso o “least absolute shrinkage and selection operator”, també coneguda com L1, penalitza la suma dels valors absoluts dels coeficients de regressió. Això força que els valors dels coeficients de predicció tendeixin a 0.

Aquesta regularització exclou aquelles predictors que tinguin menys influència.

El coeficient de Lasso és el següent:

$$\|\beta\|_1 = \sum_{j=1}^p |\beta_j| \quad (5)$$

Amb les equacions 3,4 i 5 obtenim l'equació a minimitzar seguint mínims quadrats amb Lasso:

$$\sum_{i=1}^n \left(y_i - (\beta_0 + \sum_{j=1}^p \beta_j \cdot x_{ij}) \right)^2 + \lambda * \sum_{j=1}^p |\beta_j| \quad (6)$$

El grau d'aquesta penalització es controla mitjançant el valor de λ , per tant si el valor de $\lambda=0$ no s'aplicarà cap penalització, com més creix λ més influent serà la penalització i menys ho serà els mínims quadrats.

La regularització Ridge

La regularització Ridge, o L2, de manera similar a Lasso, penalitza la suma dels coeficients elevats al quadrat.

$$\|\beta\|_1 = \sum_{j=1}^p \beta_j^2 \quad (7)$$

Aquesta penalització busca reduir tots els coeficients de manera proporcional, sense arribar al 0. Així, l'equació a minimitzar amb Ridge serà l'equació 8.

$$\sum_{i=1}^n \left(y_i - (\beta_0 + \sum_{j=1}^p \beta_j \cdot x_{ij}) \right)^2 + \lambda * \sum_{j=1}^p \beta_j^2 \quad (8)$$

Lambda té la mateixa utilitat que en el cas anterior.

ElasticNet

ElasticNet és una regularització que combina L1 i L2 de la manera següent:

$$\|\beta\|_L = \left(\alpha \sum_{j=1}^p |\beta_j| + \frac{1-\alpha}{2} \sum_{j=1}^p \beta_j^2 \right) \quad (9)$$

Aquesta regularització regula quan de cada penalització (L1 o L2) s'aplica depenent en l'hiperparàmetre α .

α és un valor entre 0 i 1. Quan el valor és pròxim a 0, s'aplica sobretot Ridge, i quan es pròxim a 1 s'aplica Lasso.

La funció definitiva a minimitzar, serà per tant la funció 10.

$$\text{MIN} \left[\frac{\sum_{i=1}^n \left(y_i - \left(\beta_0 + \sum_{j=1}^p \beta_j \cdot x_{ij} \right) \right)^2}{2n} + \lambda \left(\alpha \sum_{j=1}^p |\beta_j| + \frac{1-\alpha}{2} \sum_{j=1}^p \beta_j^2 \right) \right] \quad (10)$$

Hiperparàmetres model final

El model que ha generat AutoML té els paràmetres següents:

$$\alpha = 0.05357$$

$$\lambda = 0.11421$$

Amb aquests paràmetres, podem veure que L2 té molta més importància en ElasticNet, ja que α és proper a 0, això vol dir que ElasticNet treballa pràcticament com si fos Ridge.

Lambda és petita, per tant la influència d'aquestes regularitzacions no serà molt gran, prenent més força els mínims quadrats.