UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

FIB

Web environment for extraction and graphic analysis of classification models through data

Rubén Martín Campos

Bachelor Thesis

Specialization in Computer Science

Director: Miquel Sànchez Marrè,

Computer Science department

Bachelor Degree in Informatics Engineering

Universitat Politècnica de Catalunya

Facultat d'Informàtica de Barcelona

26th April 2021

Contents

0. Abstract

0.1. English

The main goal of this thesis is to design and implement a web framework for building classification models that help as much as possible the final users.

This framework will include the tasks associated with data pre-processing, extraction of classifier models and tasks associated with postprocessing like interpretation and validation of models through graphical analysis.

It will also include model customization, allowing the user to make adjustments to improve the quality of the classifier; feature selection, being able to save the resultant dataset for future uses; a prediction section, in order to predict data through a CSV file, and finally, portability of your work, by downloading both the fitted model and the predicted data.

0.2. Catalan

L'objectiu principal d'aquesta tesi és dissenyar i implementar un entorn web per crear models de classificació, que ajudin el màxim possible a l'usuari final.

L'entorn web inclourà tasques associades al preprocessat de dades, extracció de models classificadors i tasques relacionades amb el postprocessat com poden ser la interpretació i validació de models mitjançant anàlisi gràfica.

També s'inclouran la personalització de models, permetent a l'usuari fer canvis per millorar la qualitat del classificador; selecció de variables, podent guardar el conjunt de dades resultant per un ús futur; una secció per fer prediccions mitjançant un fitxer CSV, i finalment, portabilitat del treball realitzat, mitjançant la descàrrega de tant el model entrenat com les prediccions calculades.

0.3. Spanish

El objetivo principal de esta tesis es diseñar e implementar un framework web para crear modelos de clasificación, que ayuden lo máximo posible al usuario final.

El entorno web incluirá tareas asociadas al preprocesado de datos, extracción de modelos clasificadores y tareas relacionadas con el postprocesado como interpretación y validación de modelos mediante análisis gráfico.

También se incluirá la personalización de modelos, permitiendo al usuario hacer ajustes para mejorar la calidad del clasificador; selección de variables, pudiendo guardad el conjunto de datos resultante para un uso futuro; una sección para hacer predicciones mediante un fichero CSV, y finalmente, portabilidad del trabajo realizado, mediante la descarga de tanto el modelo entrenado como las predicciones calculadas.

1. Introduction

This thesis will face the problem of classification in machine learning and will provide a user interface for the user to extract models and graphically analyse them.

Classification is a systematic grouping of observations into categories, the problem in machine learning tries to predict the class to which it belongs, for instance there is a dataset called 'Iris Dataset' [1] in which the observations are the dimensions of different flowers and the objective is detecting to which species it belongs (Setosa, Versicolor or Virginica).

1.1. Objectives and sub-objectives

Firstly, a **web-based framework** will allow us to reach more people because it is easy to access and to use. Since most of the OS have a web explorer included or allow the user to have its preferred one, it results in a very accessible application. Basically, there's no need to install anything.

In order to have **good graphics** We will be using Dash [2], It's an Open Source project from Plotly. This will provide our app with one of the best tools for creating graphics in Python, and because it's open source it will be improving with time.

Customization and interpretability are another key part of this project, it's our intention to create an app that helps as much as possible the user to understand what is going on behind with the model, therefore they can ultimately improve the way it works through customization and end with more insights.

These are the objectives that this project seeks to achieve, in order to have that some tasks have been selected as **technical objectives:**

- **Data pre-processing:** In order to extract features and apply models to the data we need to process it. This step has different parts depending on the data and the needs of the methods apply, but a general schema can be followed:
    - **Treatment of the NAs values:** What to do when the data is missing. Sometimes you delete the row or the column, but maybe with some analysis, a value that does not hurt the distribution of the data can be applied.
    - **Categorical variables:** These variables are transformed to some numeric values; this step depends on the categorical values but generally some encoding can be applied. For instance, if the categorical variable tells us whether the subject is a male or a female, we can encode it as 0 and 1, respectively.
    - **Feature extraction:** Usually datasets have a lot of information, but not all of it has the same importance, knowing that we can use different algorithms to reduce the dimensionality of the dataset. Once we have that reduced dataset it is much faster and easier to work with it. Some technics that can be applied are the following:
        - **PCA:** Principal Component Analysis, selects the most relevant features of the dataset, those that are not correlated.
        - **MDS:** Multi-Dimensional Scaling, can measure similarity or dissimilarity of the data and visualize it.
- **Extraction of classification models:** The goal of this step is to give the user some models that suit a specific dataset or a guidance to choose one. In order to achieve this, we can set up the next sub-objectives:
    - **Test different models:** In order to choose the best model, we need to test and rank the possible models. Again, this step depends on the data, not all of the models will be applicable. Some models usually work well with classification are the following:
        - **Random Forest Classifier:** This algorithm is based on classification trees. The algorithm takes into account the prediction that every tree made and decides the final prediction. With a high number of trees this algorithm is very robust.

        - **SVM:** Support Vector Machine classification, this point is more a category than an algorithm since support machine can have different kernels and perform in different ways, but the linear and the radial basis kernel perform well with classification.
        - **Knn:** K nearest neighbors, simple and good, this algorithm tries to find K groups of data. Each observation is selected by the closest cluster, which is what we want, for it to belong to the most similar group.
- **Post-processing tasks:** These tasks relate to the interpretation and validation of the model.
    - **Interpretation:** A good way to understand the final results is with the help of graphics. Which can be:
        - Confusion Matrix of the final classification.
        - Plot of some important variable's distribution.
        - Visualization of the reduced model, by feature extraction.
    - **Validation:** In order to trust the model prediction, we need to assign some score to the predictions made, aside from that we would like to see a summary of how the model is performing.
        - Metrics and Scores of each model.
        - Visualizing the end classification, if possible.
- **Customization of the models:** This objective will be available every time the user wants to modify the model, it will have as a consequence that all the progress calculated will have to be recomputed. But is a necessity if the model lacks some important characteristic.

1.2. Requirements

The requirements of this project are the following:

- The data has to be given by the user, in order to help him. Some commonly known examples will be provided just in case.
- The dataset must be related to a classification problem.

- The app needs a storage system in order to keep track of the models and the graphics.
- The user must be able to easily download the results of its work.

1.3. Potential obstacles and risks

The biggest obstacle in my opinion is **time**, I think the idea of this project is good, but maybe time can interfere with its quality.

Implementation problems are also serious, they can delay the project a lot because of some **error or bug** that cannot be found.

The Dash library is not fully developed and maybe there are components that are not available, as a risk there may be functionalities that cannot be implemented. It should not be the case, but it can happen.

Finally, Coronavirus needs to be mentioned, the risk of being diagnosticated as a positive case exists and this could delay the project.

2. Context

This is a Bachelor's Thesis of the Informatics Engineering degree, major in Computing, and it is done in the Barcelona School of Informatics (FIB) [3] of the Universitat Politècnica de Catalunya (UPC), directed by Miquel Sànchez Marrè.

2.1. State of art

The idea of a framework for machine learning is not new, there are several options to analyse each one with its pros and cons. We will take a look at the most popular ones and try to elaborate a list of what is offered and what lacks.

- **Rapid Miner [6]:** Desktop Application for getting results fast.
    - **Pros:**
        - Labour world oriented (Deployment option).
        - Fast getting results.
        - Lots of options.
    - **Cons:**
        - Moderate complexity, hard to understand.
        - Low model customization.
        - Simple graphics.
- **Big ML [7]:** Web application for machine learning.
    - **Pros:**
        - Solves different machine learning problems.
        - Comes with example datasets for trying the application.
        - Web application.
    - **Cons:**
        - Very poor graphics.
        - Supervised classification does not show which model applies, hard to trust.
        - Unable to modify existing models.
- **Orange [8]:** Desktop Application for Data Mining & Predictive analysis.

    - **Pros:**
        - GUI Based, no need to code to mine data and get insights.
        - Wonderful visuals.
        - Lots of options.
        - Fast.
    - **Cons:**
        - Lots of options.
        - Not a web application, so it's harder to reach more people.
        - Can't add your own models unless you create it with orange.
- **Knime [9]:** Desktop Application for Data Mining.
    - **Pros:**
        - GUI Based.
        - Lots of models.
        - Deployment options.
    - **Cons:**
        - Overwhelming UI, it has a lot of options, text and windows.
        - Hard to learn to use.
        - Simple graphics.
        - Not web based.

In our opinion **Orange** is the most complete and intuitive of all, it is well balanced between utility and complexity.

But no application is perfect, each one has chosen to improve some points in exchange of losing into other ones.

2.2. Terms and concepts

This thesis addresses the problem of Classification, which consists in trying to assign a label to any given set of data. This label is often called class, target or variable and the problem could have several labels.

For instance, we can classify the mail we receive into spam or not, or even differentiate if a person suffers from a disease or not.

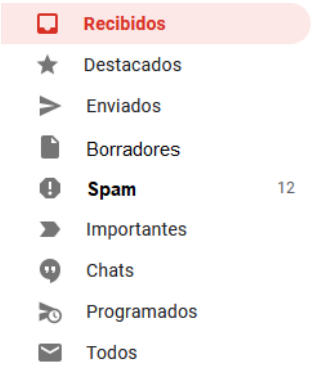

Figure 2.1: Gmail's [4] Inbox Spam

In order to work with classification problems, we need a table of data related to the problem, called dataframe or dataset, this will help us extract information about the problem so we can use an algorithm to classify the data.

This algorithm will not always be the same, we will have a set of algorithms we can use and we will try to select the one that works the best with the current data on the problem.

As for classification algorithm evaluation, we are in the same situation, the best algorithm fir any dataset does not exist and we need to use a several algorithms in order to evaluate our classification algorithm.

For binary classification problems, there is a well-known curve that allows us to compare different algorithms, it's called ROC Curve (Receiver Operating Characteristics) and shows the trade-off between the true positive rate and the false positive rate. [5] For multiclassification problems, we will use the same concept but we will compare the difference predicting each class.

- **Target Class:** Is the class we want to identify, e.g.: Spam in mail inbox.
- **True Positive Ratio:** Is the quotient between the correctly identified elements belonging to the target class (TP, True Positive) and the addition of the last ones to the elements that belonged to the target class but weren't properly identified (FN, False Negative).

$$TPR = \frac{TP}{TP + FN}$$

- **False Positive Ratio:** Is the quotient between the elements wrongly identified as belonging to the target class, but (FP, False Positive) and the addition of the last

ones to the elements that were correctly identified as not belonging to the target class (TN, True Negative).

$$FPR = \frac{FP}{FP + TN}$$

Lastly, we will create graphics to better understand the model created and its logic behind. We will start with some basic but very useful techniques in classification problems, one of them is the Confusion Matrix and compares each class to its classification. It lets the end user clearly visualize the performance of its classification model.

this last step is very important since it allows us to trust what is happening by watching the final results, this last step belongs to a field called Interpretative Machine Learning (IML) and it has improved a lot over the years helping us to extract important insights.

2.3. Selection of tools used

We will discuss the available options to carry on the project, since we are creating a web application related to machine learning there are two main aspects we need to define:

2.3.1. The Language

Two of the most used languages for machine learning are Python [14] and R [21], and both languages have frameworks to create a web application with. Therefore, we needed to decide which language to use, Python [14] or R [21],.

Both options will lead to similar results, but with little differences, the framework option for R [21] doesn't use the same graphing library, by default, as any of the Python [14] frameworks, but both will have almost the same plots. Another difference could be the libraries used for obtaining the machine learning models, in Python [14] we chose to use Sklearn [22], but this option is not available in R [21], and so on...

Even though both resultant applications will have their differences they could do the same thing, so it is a matter of which one I can use better in order to solve the problem raised.

In this case Python [14] is chosen mainly because I have much more experience and I thought I could do much more and better work in the same amount of time than using R [21].

2.3.2. Project management

We can implement the following frameworks, combined with basic website languages:

1. **Flask [23] and HTML+JS+CSS:** Flask is a micro web framework written in Python, in this case it is very common to use HTML+JS+CSS to handle the frontend and Flask as a backend allowing us to navigate the website and to use python for machine learning in the back.
2. **Dash [2] and CSS:** Dash is a productive Python framework for building web analytic applications and it is written on top of Flask, Plotly.js [24], and React.js [25]. We can easily notice that dash is a framework that has all the Flask functionality and more, it allows us to create common web structures like inputs, lists, tables, etc... with less effort than using raw HTML+JS+CSS.

As long as there are no drawbacks for using Dash instead of Flask, Dash is our best option by far. There could be potential problems that could make us choose Flask instead of Dash, for example if Dash forced us to use something either not efficient of that we dislike, but it is not the case, furthermore they are giving support to help people be able to do as much as possible with their framework.

2.4. Stakeholders

The web application aims for helping people with classification problems, from the start with data pre-processing until the end with graphics and understanding the model.

Therefore, there are many stakeholders, which will be listed below:

- **Students and teachers:** I believe this tool will help a lot in the process of learning and teaching machine learning, mainly classification problems. In the scope of FIB, UPC, the app could be used in the next subjects:

- **Artificial Intelligence (IA):** It could help introducing the "next" related subject, Machine Learning, since it is easy to use, the teacher could do a live demonstration and show both the problem and the solution. For the current progress of the application the problem will be a classification problem.
- **Machine Learning (APA):** In this subject the app will be more useful, even though at its current state it solves classification problems, which is not the entire scope of the subject. Aside from solving classification problems, the teacher can also show some data pre-processing and some other concepts related to Machine Learning, e.g., different classification models, feature extraction, model performance metrics, final result plots, etc...

- **Companies:** Having an environment that can help you understand the classification models used in a project is extremely useful, it can help not only the ones that program the model but also everyone along the company, directors, managers, salespersons, ...

People interested in classification problems: This tool can offer the user a great experience, improve its performance, creating classification models and help him produce beautiful graphics for better understanding.

2.5. Knowledge Integration

During the project development, several aspects covered throughout the degree have been integrated in order to improve the quality of the proposed solution for the project. We will describe the aspects used in the project and in which subjects they were obtained.

2.5.1. Interaction and Interface Design

This subject helped me understand how the programming interfaces work, in my opinion the most important aspect obtained for this project is the concept of slots and signals. This concept is encompassed by callbacks, and they correspond respectively to the outputs and inputs used in Dash [2].

In this subject I also obtained all the basics of how to design user interfaces and other important concepts such as colour models like RGB, used in CSS.

2.5.2. Programming projects

In Programming projects, I learned about Object Oriented programming, how to manage big projects and some basic programming patterns. One key concept that I learned is being able to store and load object from RAM to Disk, I can't express enough how important this aspect is, it allows the app to store the trained models, to have different projects and datasets... without it this project will be much simpler and overall a worse solution to the problem.

2.5.3. Probability and Statistics

This subject gave me the basic knowledge of mathematics, allowing me to understand and learn Machine Learning. It helped the project indirectly on so many areas and directly helped in the early stages of the project, those related to understanding and cleaning the datasets; later into the project it helped understanding and creating the results of every model.

2.5.4. Machine Learning

The most relevant subject, this project is focused on Machine Learning, specifically in extracting and analysing classification models through the data.

Therefore, this subject is a key aspect for the project, it is necessary to know how the structure of the website is going to be built, in an intuitive way to process the data and obtain the desired results; but also, to understand the concepts behind each tab in the project and program accordingly to what is needed. Almost every aspect of this subject is helpful to develop the project, starting from the simpler ones like data pre-processing, continuing with learning and understanding the different models and ending in result visualization and interpretation.

2.6. Conclusions

As we saw in the last chapter, the idea to create frameworks to help people with Data Mining and, Machine Learning models and graphics is not new.

Nevertheless, there are a few points all of the applications above lack or don't do well enough. These points are the next ones:

- Maximizing the amount of people their application reach.
- Making it easy to learn and intuitive.
- Creating good graphics for the user.
- Allowing customization of the models.

Even though there are some good applications for doing similar tasks I think there is much more to do and improve. This project's main goal is to create a web application that reaches as many people as possible and helps them to work with Data Mining Classification problems giving them tools to improve and learn.

I will also address the points the studied applications lack, in order to create a software that makes a difference.

3. Methodology and design

3.1. Methodology

This project's task will be organized using the Kanban methodology, which is simple, intuitive and helps clearly visualize what tasks are done, which are in progress and which are still remaining to be done.

For back-ups, I will be storing copies of the code in different hard drives and also in the cloud, the documentation will be stored in the same way. This is because there is no need to share and work on the same code, since only one person is working on it.

3.2. Methodology tools

As we mentioned above, only one person will be coding therefore I won't be using GitHub [17]. I will instead store copies of the code in different hard drives and also in the Cloud, **Google Drive [16]**. The same system will be applied to documentation.

The web app **Trello [18]** will be used to organize the project using the Kanban methodology, It's one of the best apps for this methodology and allows you to invite people, which will be incredibly useful for improving the communication between my thesis director and I.

We will start with 3 tabs, "**To do**", "**Doing**" and "**Done**", but if we need another useful tab it can always be added in the future.

3.3. Application Design

The application will be designed following the requirements and objectives stated in sections 1.1 and 1.2. In order to give the application, the ability to save its state, we created the concept of a 'Project', a project has datasets related and models. Also, when you work with a dataset you may want to test different models to find which one suits better your problem and gives you the best results, for that reason a dataset can have

several models associated. All of those aspects need to be persistent, for that we will use several python functions that allow us to save objects to disk.

3.3.1. Use cases

According to the objectives and the requirements defined, the global flow of the application needs to have the following steps: preprocessing, model extraction, postprocessing, predicting and some tool to save the project. Therefore, the **global use case** of the application is the next one:

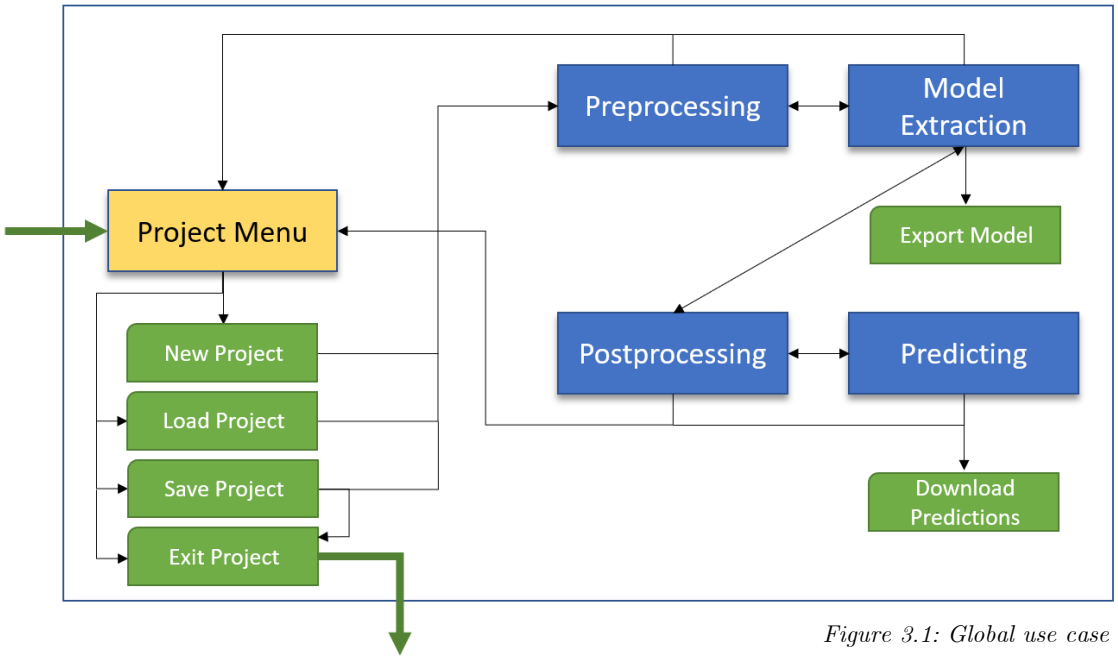

Figure 3.1: Global use case

It's important to notice that you can save the project and exit in any stage in the project, when exiting the project, you can decide if you want to save or not.

Also, the usual flow of the application will be first deciding which project you want to start or load, then preprocess the data, extract models for that data, go to postprocessing in order to evaluate the models and finally using the model you want in order to predict. It is also possible to go back in the flow if you need to change anything.

3.3.1.1. Preprocessing use case

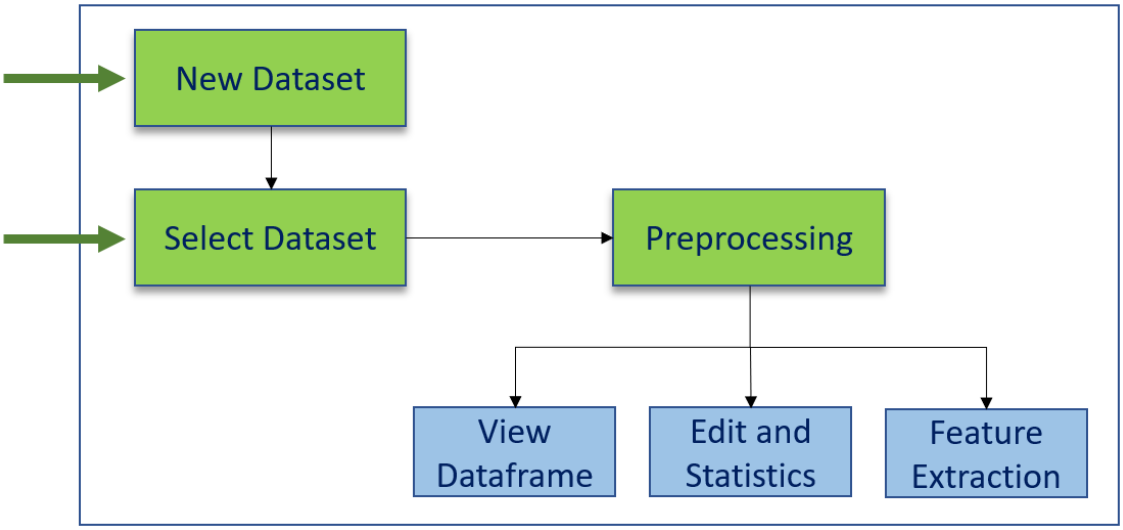

Figure 3.2: Preprocessing use case

The green boxes correspond to different windows of the final web app, inside the preprocessing use case, the New Dataset allows the user to add datasets to the project, once added the user can select though the Select Dataset window, finally the selected dataset can be preprocessed in the next window named 'Preprocessing'. The blue boxes are also windows that allow the user to view the data frame, perform some modifications, like removing NA values, converting categorical to numerical or vice versa, etc., and finally performing Feature Extraction and save the results as a new Dataset.

3.3.1.2. Model Extraction use case

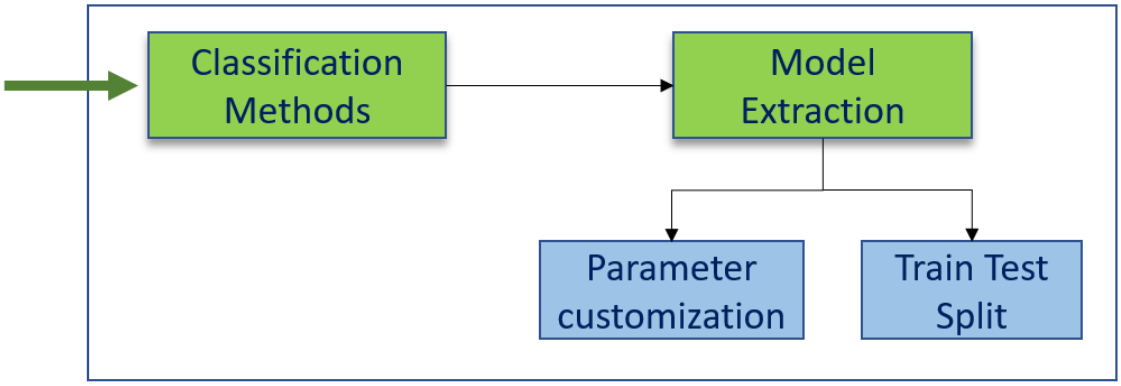

Figure 3.3: Model Extraction use case

In this use case the user selects the methods to work with the selected dataset, through the 'Classification Methods' window, once selected it is possible to tweak the parameters of each classifier and finally start the training once the 'Train Test Split' parameters have been selected.

3.3.1.3. Postprocessing use case

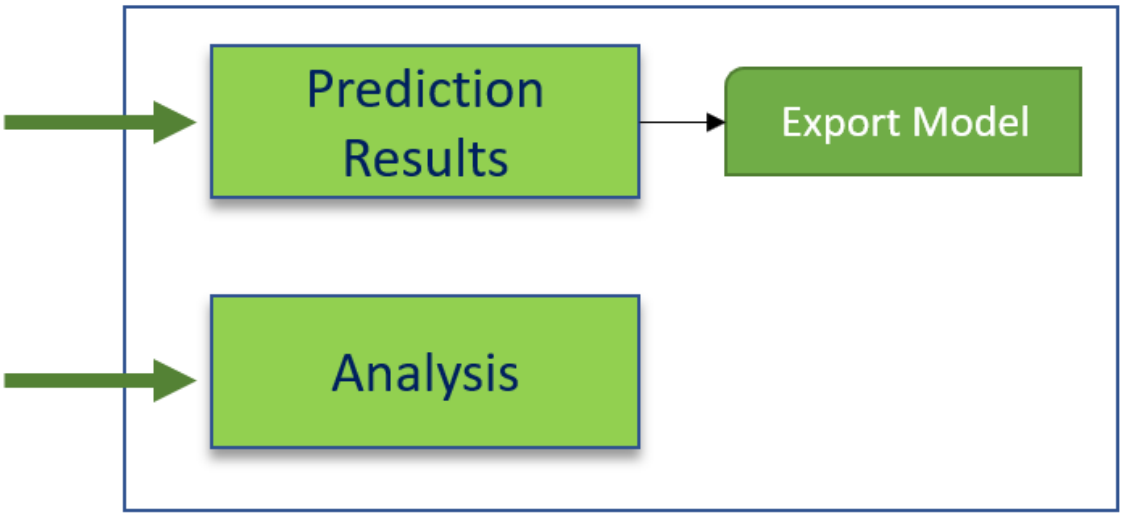

Figure 3.4: Postprocessing use case

In postprocessing use case it is possible to see how well each model performed the test predictions, the average precision, the accuracy, the F1 score and the confusion matrix are provided, also if the user likes the performance of the classifier and wants to export it is possible.

Moreover, in the Analysis window the user can see the Roc Curve for each classifier, if the classification is binary, or for each class if the problem corresponds to multiclassification.

3.3.1.4. Predict use case

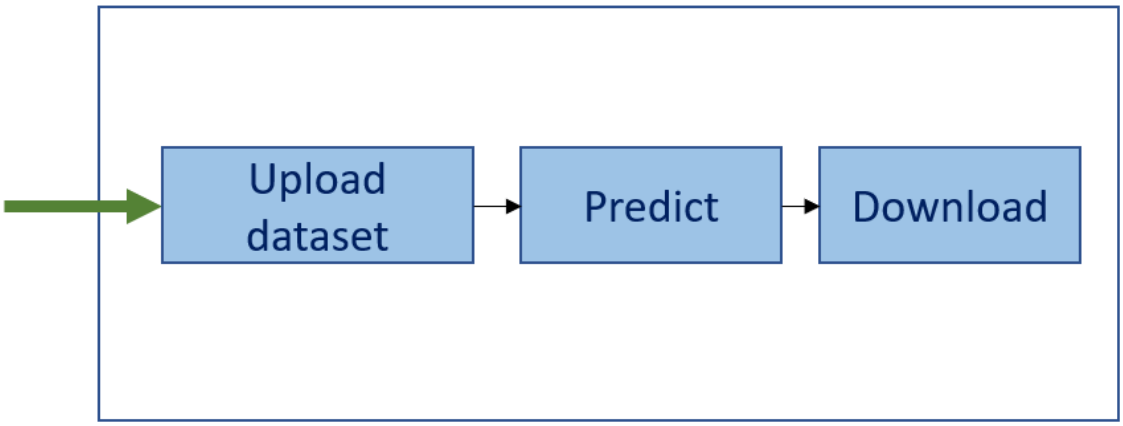

Figure 3.5: Predict use case

The predict use case has only one window, the user can upload a dataset in order to perform a bulk prediction and view the results, if they are needed, outside the application context, it is possible to download them.

3.3.2. Conceptual Model and 3 layer Architecture

3.3.2.1. Conceptual Model

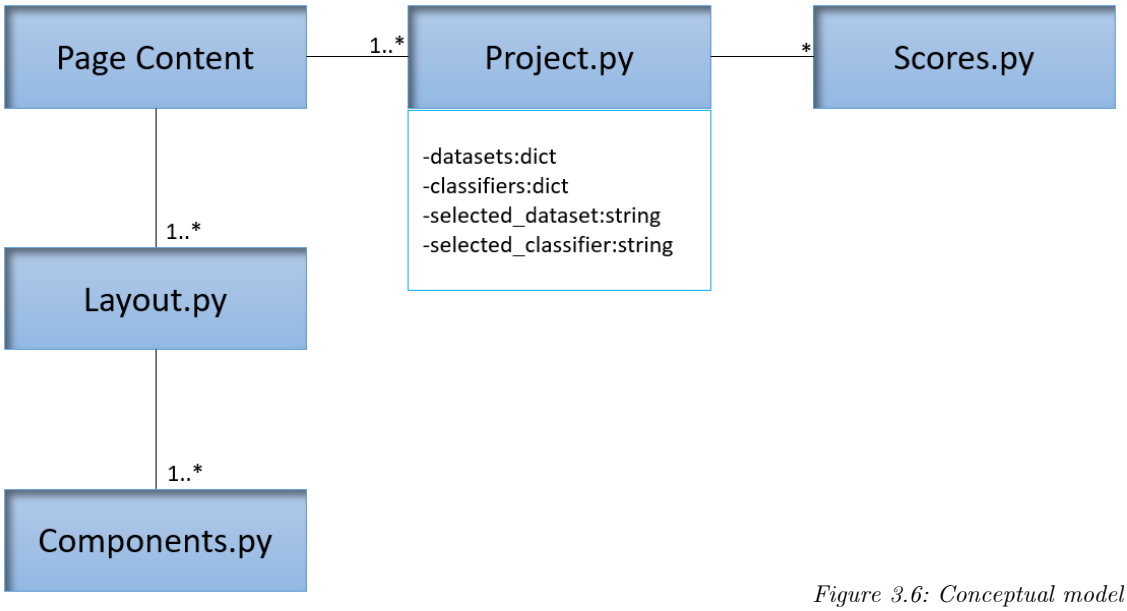

Figure 3.6: Conceptual model

Above we can see the Conceptual Model, it does not show the functions because there are so many that it would not fit. The Page Content is where it gets displayed the page layouts that we create, this is done through the server app provided by dash, that responds to the user requests returning our layouts and the information provided by the project class.

There are N layouts one corresponding to each page/window we want to display in the application. Some layouts are dynamic since they receive the information from the project class, through the main file of the application, and generate the corresponding layout. We can see the code for that serves the different layouts in the application:

```
@app.callback(
    Output("page-content", "children"),
    [Input("url","pathname")]
)
def render_page_content(pathname):
    print(pathname)
    if pathname == "/":
        return home_layout
    elif pathname == "/preprocessing":
        return preprocessing_layout
    elif pathname == '/model-extraction':
        return model_extraction_layout
    elif pathname == '/postprocessing':
        return postprocessing_layout
    elif pathname == '/predicting':
        return predicting_layout
    elif pathname == '/new-project':
        return new_project_layout
    elif pathname == '/load-project':
        global projects
        lps = projects.get_project_names()
        pjs = []
        for name in lps:
            pj = {"label":name, "value":name}
            pjs.append(pj)
        return create_project_layout(pjs, projects.current_project)
    elif pathname == '/save-project':
        current_project = projects.current_project
        modified = projects.modified
        return save_project_layout(current_project, modified)
    elif pathname == '/close-project':
        current_project = projects.current_project
        modified = projects.modified
        return close_project_layout(current_project, modified)
    else:
        return dbc.Jumbotron(
            [
                html.H1("404: Not Found", className="text-danger"),
                html.Hr(),
                html.P(f"The pathname {pathname} was not recognised..."),
            ]
        )
```

Figure 3.7: Page rendering code

In the components class we can find some functions and objects that are used a lot in the project, for instance the tables displaying the dataset information generates its base structure in the components file and adapted to show the dataset vales in the layout file.

Finally, the scores class contains the metrics used in the project and the wrapper for the feature importances algorithm we use. The algorithm used is RFECV (recursive feature elimination and cross-validated selection) that returns an object containing all the classifiers and its cross-validation scores, with that object we can retrieve the feature importances of the best classifier. We can see the code for the wrapper below:

```python
def calculate_feature_importance(est, X, y, step, cv):
    try:
        selector = RFECV(est, step=1, cv=5)
        selector = selector.fit(X, y)

        feature_importances = selector.estimator_.feature_importances_
        zeros = 0

        if len(feature_importances) < len(selector.ranking_):
            scores = []
            for i, rank in enumerate(selector.ranking_):
                if rank == 1:
                    scores.append(feature_importances[i - zeros])
                else:
                    scores.append(0)
                    zeros += 1

            feature_importances = scores

        return feature_importances, selector.ranking_, zeros, selector.support_
    except Exception as e:
        print(e)
        return None
```

Figure 3.8: RFECV implementation

We can see that we need to process the returned information, this is because the best classifier will most probably don't have all the variables in the dataset, but we want to see that as a result, we need the application to tell us which variables are not important. In order to detect the least important variables we can just check that the rank assigned by the algorithm is different than 1, which means it has a feature importance of 0.

3.3.2.2. 3 Layer Architecture

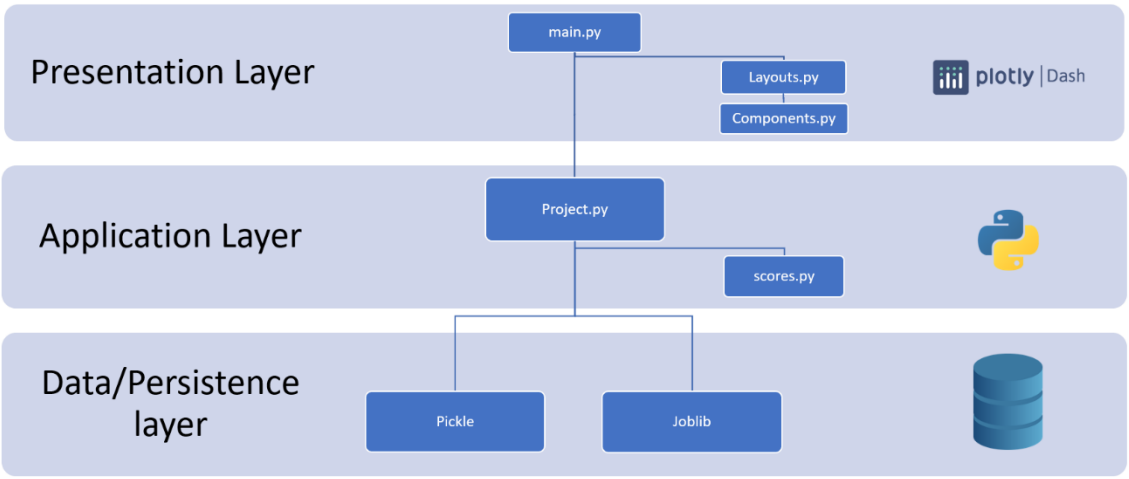

Figure 3.9: 3 Layer architecture

The first file in the presentation layer is the main.py, this is the dash server which routes the requests that the web application receives, serving the corresponding page layouts and updating the contents of the page dynamically through callbacks. The layouts and the components are provided by its corresponding class, also influenced by the data provided by the application layer.

In the application layer we mainly find the Project.py file and the scores.py, this is because the methodology agile was used, I consider that a good improve to the application will be refactoring the project class and extracting classes, maybe a class for the datasets and another one for the classifiers...

For the data layer, since we don't use a database the data layer corresponds to the libraries that help our application save everything, we need to recover the application state, this is done with pickle, for python objects like dictionaries or lists, and joblib for saving the trained classifiers.

3.4. Changes in methodology

All of the specified methodology was respected and used, we will justify why nothing was changed.

- **Trello, Kanban Methodology:** I find this methodology very illustrating, simple and clear. In this project, we needed to keep track of the tasks that have been done, the ones that are in progress and the pending to do, there is no need to make things harder than they are. You can also add comments to each task, making possible to keep track of little steps in each task fulfilment.
- **Google Drive, back-ups:** The main alternative option to back-up your project is GitHub, but is slightly more complex to Google Drive, since I don't want to share the project, in order to work, I preferred the option of Google Drive. That was the main reason I chose this option, and now in the middle of the project, it makes no sense to switch between them, I had no problems with Google Drive so it wasn't necessary to change.
- **Validations:** The functions and its implementation were validated in the same way it was described in the last section, considering the time we have is limited I think it is the best option because we ensure that the current implementation works in the current scenario and we use less time compared to an option that implements unit tests for every single function and the global implementation. If the project was bigger, with several classes and interactions between them, we could consider using unit tests.

4. Implementation aspects

In this section some details about the code will be given, explaining with details how the use cases have been implemented.

4.1. Preprocessing use case

This use case has 3 windows, they can be accessed by selecting the desired tab in the menu.

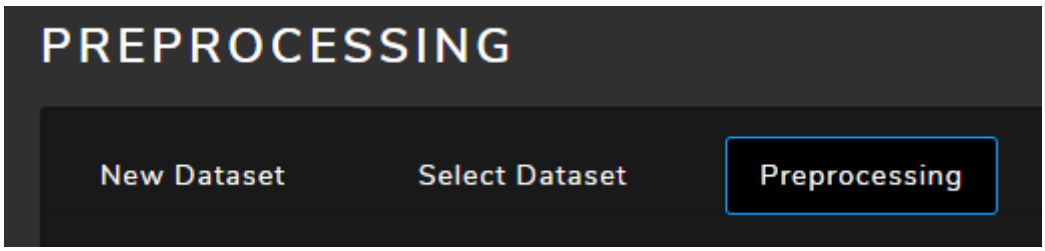

Figure 4.1: Preprocessing tabs

The first two tabs are very straight forward so we will focus in the preprocessing tab, this tab has a Control Menu that allows you to navigate the different options that this tab gives you.

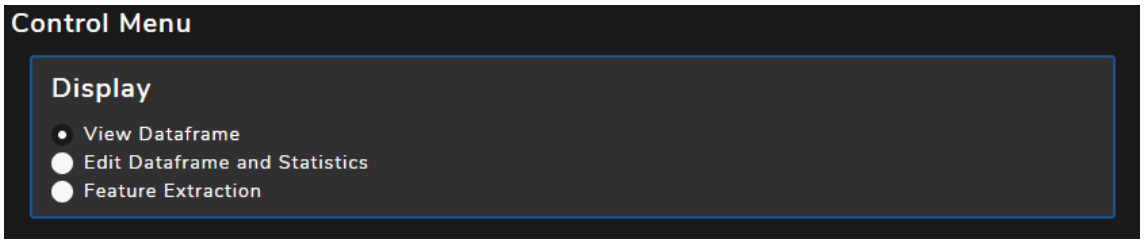

Figure 4.2: Preprocessing control meny

The View Dataframe option allows the use to see the selected dataframe occupying as much screen as possible. The data is displayed in a table with a dynamic header that follows the scroll of the page.

In the Edit Dataframe and Statistics we find the following options: 'Select the Target Feature', 'NA Value Treatment', 'Categorical Values' and 'Discretization'. The code of

the Categorical Value transformation and the Discretization transformation is the following:

```
def transform_categorical_to_numerical(df, categorical_column):
    if categorical_column in df.columns:
        column = df[categorical_column].tolist()
        mapping = [ x for x in enumerate(list(set(column))) ]
        print(mapping)
        df[categorical_column] = [ x for x in map(lambda x: [ i for i, cat in mapping if cat == x ][0], column) ]
        return df, mapping


def transform_to_discrete(df, discrete_column, discrete_intervals):
    if discrete_column in df.columns:
        column = df[discrete_column].tolist()

        min_v = min_wrapper(column)
        max_v = max_wrapper(column)
        rnge = max_v - min_v
        interval_size = rnge/discrete_intervals

        mapping = [ (min_v + interval_size*i, min_v + interval_size*(i+1)) for i in range(0,discrete_intervals) ]

        df[discrete_column] = [ select_discrete_value(mapping, x) for x in column ]
        return df, mapping
```

Figure 4.3: 'Edit dataframe and statistics' code for categorical and numerical transformation

The Categorical to numerical enumerates the different values in the selected column, and assigns the corresponding numerical value to each category. As an improvement, it wouldn't be hard to select the desired numerical values for each category.

In the discretization transformation the user decides how many intervals wants to create, once selected the number of intervals we divide create a mapping that helps to decide where each value is mapped. If the value is inside the two values in the pair

(min_v + interval_size * i, min_v + interval_size*(i+1))

then we assign the value i, for i in range from 0 to the number of discrete intervals (The assignment is done inside the 'select_discrete_value' function).

Finally, in the Feature Extraction tab, we use the algorithm RFECV (Recursive Feature Elimination and Cross-Validated selection), we can see an image of the code that wraps the SK-learn function in the section 3.3.2. In this tab the user selects the Steps and the Cross Validation folds to perform and the application shows you the resultant Feature importance and allows you to save the dataset without the discarded variables. Results for the heart disease dataset, with 1 step and 10 cross validation folds.

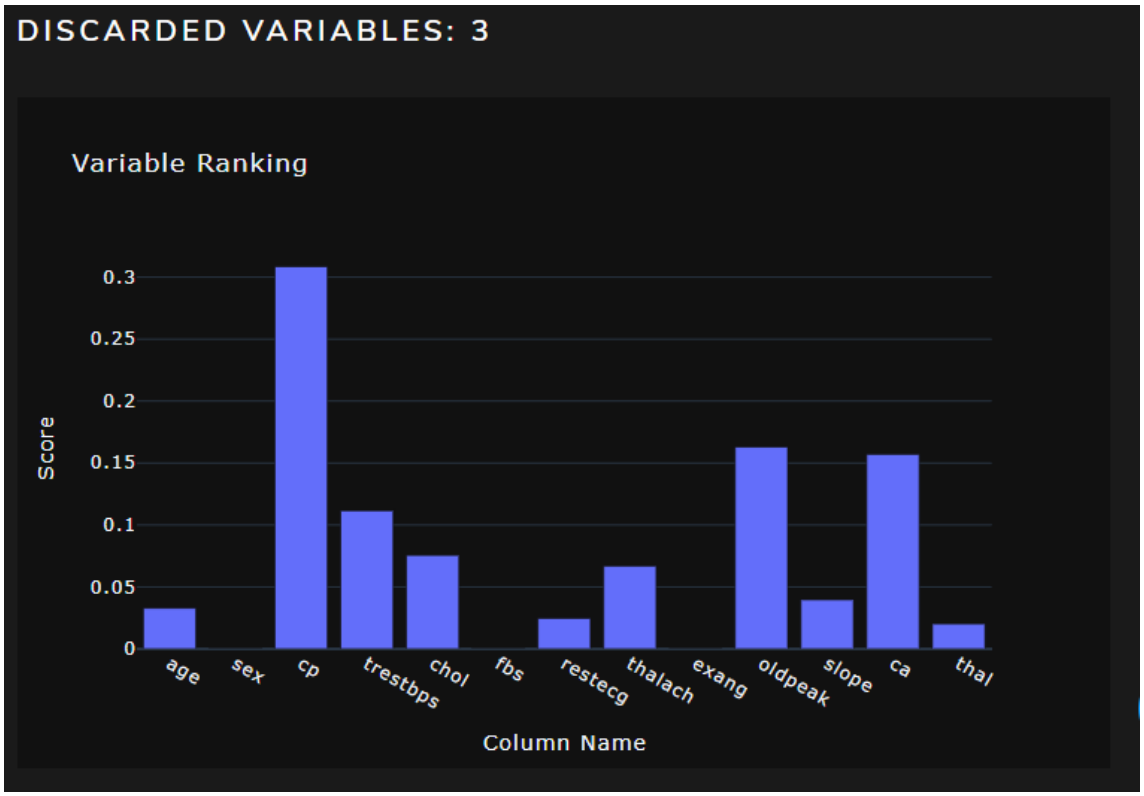

Figure 4.4: RFECV variable ranking

4.2. Model extraction use case

The Model Extraction tab has two windows, the first one selecting the corresponding models that the user wants to use in this problem.

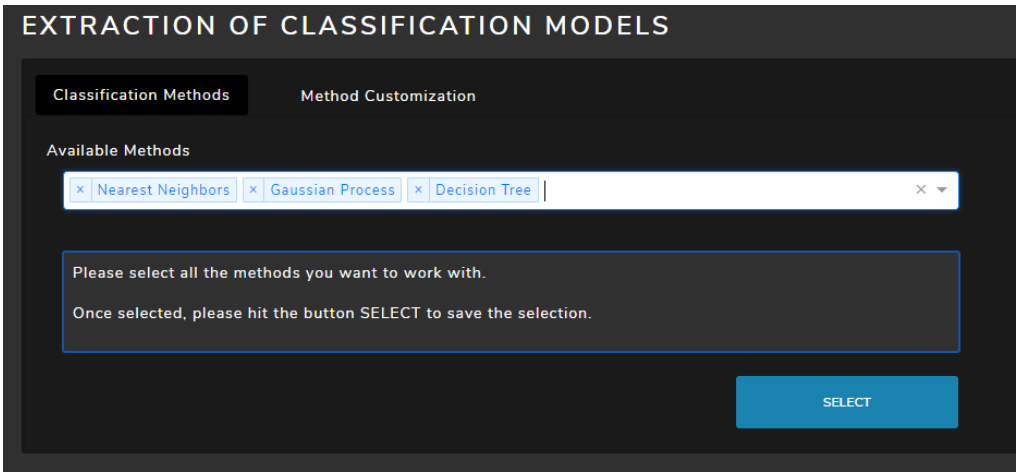

Figure 4.5: 'Extraction of classification models' tabs

The user has a list of all the available classifiers in the sklearn library, and can assign as much as he wishes since the selection is multiple, as we can see in the image above.

In the 'Method Customization' tab we can modify all the parameters in the classifier, and the 'train test split' parameters in order to start the training. Generating a form for the user to modify each parameter of every different classifier was a tricky, since the form has to be generated dynamically for all the classifiers. The dynamically generated form for Nearest Neighbors classifier is the following:

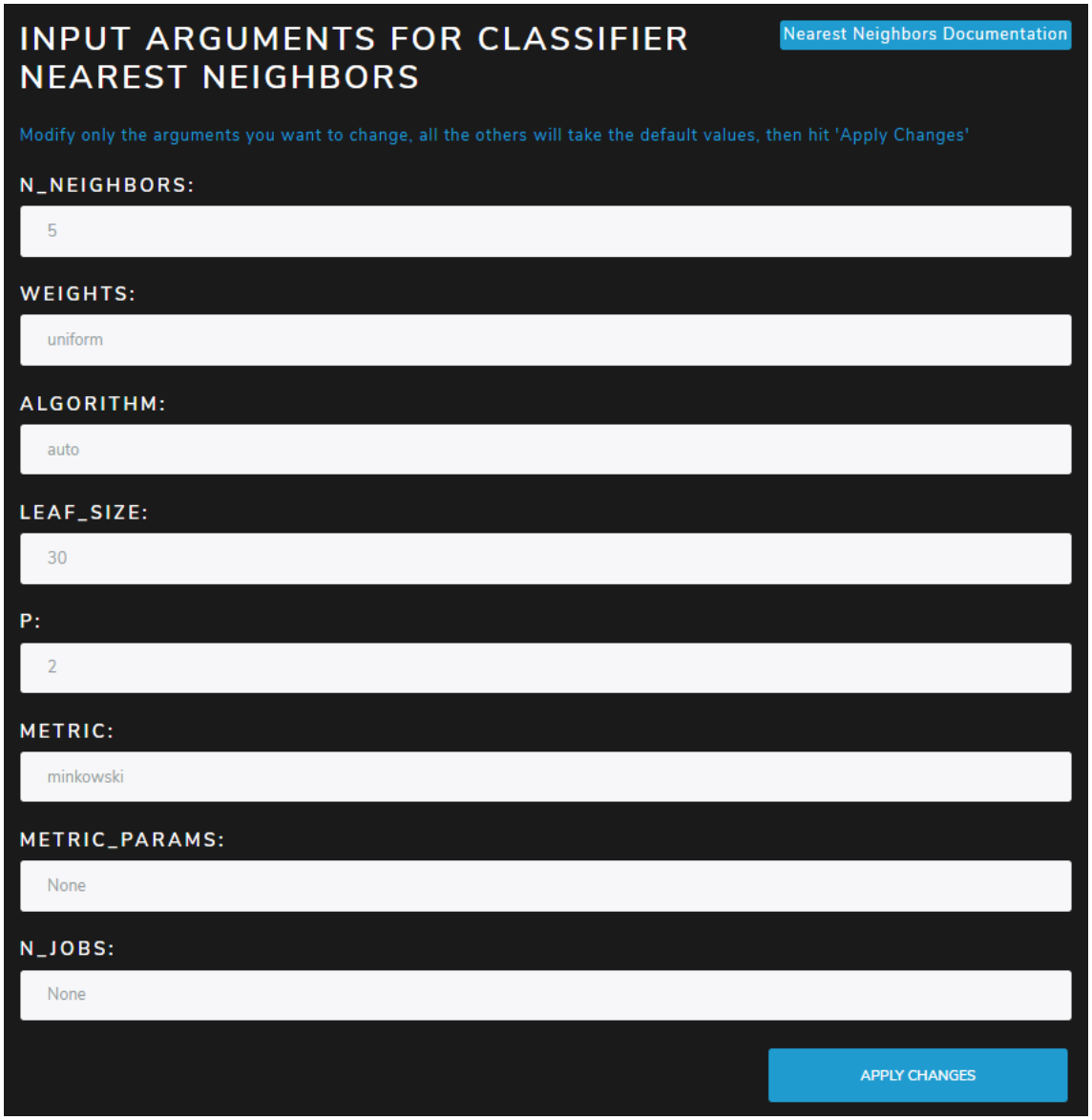

Figure 4.6: KNN argument input

If we don't know what an argument does, we can easily access the classifier documentation at the top right button.

In order to solve that problem, we needed to make use of the inspect python library, allowing use to retrieve the classifier arguments and its default values. Another tricky part was transforming the user input from string to its corresponding type, for that we used regular expressions, we can see the code that solves that problem:

```
def __cast_classifier_argument__(self, argument):
    int_regex = '^[-+]?[0-9]+$'
    simple_float = '^[-+]?[0-9]+.[0-9]+$'
    scientific_float = '^[-+]?[0-9]+e[-+][0-9]+$'
    float_regex = f'{simple_float}|{scientific_float}'
    dict_regex = '^\{.+:.+[,.+:.+]*\}$'
    pair_regex = '^(.+,.+)$'
    bool_regex = '^True$|^False$'

    # We eat spaces
    argument = re.sub(r"\s+", "", argument)

    if re.search(pair_regex, argument):
        la = argument.split(',')[0][1:]
        la = self.__cast_classifier_argument__(la)
        ra = argument.split(',')[1][:-1]
        ra = self.__cast_classifier_argument__(ra)
        return (la, ra)
    elif re.search(float_regex, argument):
        return float(argument)
    elif re.search(int_regex, argument):
        return int(argument)
    elif re.search(dict_regex, argument):
        argument = argument.replace('\'', '\"')
        return json.loads(argument)
    elif re.search(bool_regex, argument):
        return argument == 'True'
    else: #String case
        return argument
```

Figure 4.7: Parsing code for classifier arguments

We can see that our function detects a lot of different types, all of the types that are needed in the classifiers. Note that we also preserve strings, if it does not match any of the types then it must be a string.

In the event that there is any given error in the classifier instantiation, the user will be prompt with a dialog that helps detecting that error and fixing it.

4.3. Postprocessing use case

In the postprocessing tab we can see the model evaluation, in the first tab the user can find the following metrics:

- **Average Precision:** Summarizes a precision-recall curve as the weighted mean of precisions achieved at each threshold, with the increase in recall from the previous threshold used as the weight.

$$\mathrm{AP} = \sum_n (R_n - R_{n-1}) P_n$$

- **Accuracy:** Accuracy is closeness of the measurements to a specific value.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- **F1 Score:** The F1 score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. The relative contribution of precision and recall to the F1 score are equal.

$$F1\ Score = \frac{2 * (precision * recall)}{precision + recall}$$

- **Confusion Matrix:** Is a specific table layout metric where $C_{i,j}$ is equal to the number of observations known to be in group and predicted to be in group, where C is the Confusion Matrix and i and j go from 0 to n-1 where n is the number of observations.

In the next tab called 'Analysis' we find a ROC curve, if the dataset selected corresponds to a binary classification problem, then there will be as much curves as classifiers trained, therefore we can easily compare how well are performing in comparison to the other ones. It is possible to select any combination of classifiers to compare.

If instead the problem we are facing is a multiclassification one, there will be a curve for each class in the problem, and we can change the classifier we are using to calculate the ROC curves. Like in the binary case we can toggle the curves displayed in order to compare them easier.

4.4. Predicting use case

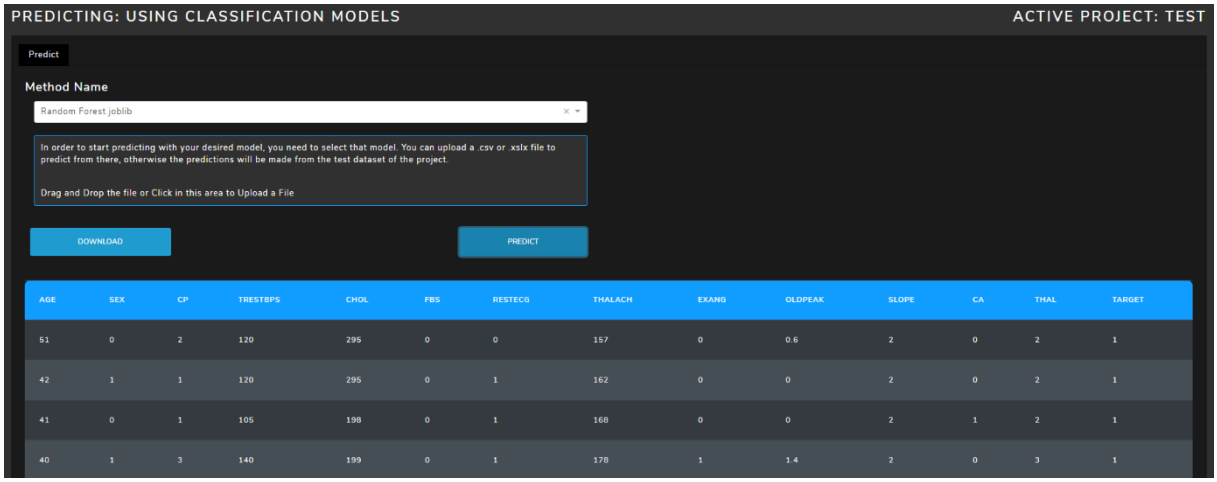

Figure 4.8: Predicting use case layout

The predicting use case is very straight forward, as it can be seen in the image you can select the method you want to use for the prediction, once selected if you click in the blue outline rectangle you can upload a dataset to perform the prediction, this area reacts changing the colour and the mouse pointer when hovering so it is easier to spot, also if you read the description, it tells you to click in the area or drag and drop the files to that specific rectangle.

4.5. Validation

As for model validation, some verification process will be applied, as mentioned before in 1.1. In order to thoroughly verify that the app and everything related to the models are well programmed, some tests will be applied. First **checking each function and then the global implementation.** In order to test the web application we need some classification datasets, I selected the following: "Iris Data Set" [1], "Diabetes Dataset" [19] and "Heart Disease" [20], the first corresponds to a multiclassification problem. In the

preprocessing part I duplicated the datasets and added some modifications, introduced some missing values, changed categorical columns to numerical and vice versa.

Moreover, some **reunions** will be done **periodically** to report the progress and obtain some feedback. By doing this, we can redirect the project from possible deviations of the established goal or errors accumulating.

As a final test, I will ask different end users, for example to the thesis director and some friends with a basic level in informatics, to test the application without giving them many indications. This will help me see some possible design mistakes or things I will never try that could raise an error.

5. Temporal Planning

5.1. Description of tasks

This project's duration is approximately 4 months, starting at September 21st and ending at January 25th, this dates and timings are approximated at the beginning of the project and can be altered by changes or obstacles. Assuming I can successfully invest 4 hours every day during 126 days it makes a total of 504 hours.

5.1.1. Project management

These tasks include all the management and planification of the project. The timings have been estimated taking into account the assumption of working 4 hours per day.

- **M1 – Context and scope of the project:** Project's context and scope definition take about 28 hours.
  Microsoft Word [10] is needed and also some basic programs for taking screenshots for the project's documentation.
- **M2 – Time planning:** Creating the project planning and all the support graphics takes about 16 hours. Microsoft Word [10] and Microsoft Project [11] is required in order to create the Gannt diagram and the summary of the tasks.
- **M3 – Budget and sustainability:** Creating a budget plan and a sustainability analysis takes about 16 hours. Microsoft Word [10] and Microsoft Excel [12] is required in order to help the computation of the budget and sustainability.
- **M4 – Final Document:** This task consists in integrating all the previous tasks M1, M2 and M3 in order to create the final project's Management document. It requires about 16 hours and Microsoft Word [10] to create the document.
- **M5 – Reports of the progress:** Every completed task requires a feedback of the director of the project, a report will be delivered and meetings could be arranged to analyse the documents provided and the project's evolution. This task takes approximately 1 hour and requires Gmail [4] and Skype [13].

5.1.2. Project development

- **DV1 – Web's UI Structure:** We need to program a basic web environment for each part of the application (Pre-processing, model results, etc...). This is just a general structure for organizing the web app, the navigation bar, the different routes, the colour palette, etc...

  This task will take 16 hours.
- **DV2 – Persistency of the data:** We need a way to store models in the web application so the user can work, customize and download if needed.

  12 hours will be needed to finish this task.
- **DV3 – Add different models to the app:** Some classification models from the SK-Learn [13] library will be added. These models will allow us to make predictions and classification, so they are an essential part of this project. This task will take 10 hours.
- **DV4 – Model Customization:** Adding a module that allows model customization and its corresponding UI. In order to finish this task 24 hours will be needed.
- **DV5 – Pre-processing module:** We need to create a module for processing the current's problem data. This task will include the improvement of the basic UI for the Pre-processing module. 20 hours will be required in order to finish this task.
- **DV6 – Extraction of classification models:** In this task a set of classification methods and feature extraction techniques will be applied and evaluated to the data. After that evaluation, the results will be presented to the user allowing him to choose the one he considers the best.

  This task will require 30 hours.
- **DV7 – Post processing module:** This task consists in taking the results obtained and processing them, obtaining graphs that will be presented to the user providing new relevant information about the problem.

  This task will require 20 hours.

5.1.3. Project documentation

We need to keep track of the more technical advances or events of the project.

- **DC1 – Technical documentation for each task we complete:** We need to document as much as possible the code we create and the technical functionalities of the application, thus we will comment those aspects as we create new functionalities.

  This task will require 116 hours.
- **DC2 – Write documentation of the application's usage:** Each application must have a 'help' section and so the documentation must have a description of what each functionality of the app does. This task is concerned with documenting those aspects.

  This task will require 20 hours.
- **DC3 – Conclusions and further work:** When the application is in its final state, we can write our final conclusion about it and also comment out the things that can be done to improve the application.

  This task will require 12 hours.
- **DC4 – Final revision:** The last step is to check that every aspect of the documentation is in order and fix the possible mistakes.

  This task will require 20 hours.

5.2. Description of Resources and Roles

In order to perform the lately described tasks some resources and two main roles have been selected.

The resources chosen are either software or hardware that will help us develop our project:

- **PC:** My personal computer has Windows 10 installed, an AMD Ryzen Processor and 16 GB of RAM.
- **Microsoft Word [10]:** A word processor developed by Microsoft.
- **Microsoft Excel [12]:** A spreadsheet developed by Microsoft.
- **Python [14]:** An interpreted, high-level and general-purpose programming language.
- **Dash [2]:** Is a Python framework for building web applications. It built on top of Flask, Plotly.js, React and React Js.

- **Visual Studio Code [15]:** Is a freeware source-code editor made by Microsoft.
- **Google Drive [16]:** Is a file storage and synchronization service developed by Google.

The Main roles needed in this project are the following:

- **Consultant:** Is an experienced member of the team, has knowledge of project management and different fields, in this case programming, machine learning and API programming.
- **Programmer:** A qualified person to carry on the task of programming, in this case knowledge of Python, machine learning and web programming is required.

5.3. Estimates and the Gantt

As for resources used and persons on the team performing tasks, a more detailed description of the workload division is provided in the previous section.

ID	Name	Time(h)	Predecessors	Resources
M	**Project Management**	**94 horas?**		**PC, Microsoft Word**
M1	Context and scope	28 horas		PC, Microsoft Word
M2	Time planning	16 horas	M1	PC, Microsoft Word, Microsoft Project
M3	Budget and sustainability	16 horas	M2	PC, Microsoft Word, Microsoft Excel
M4	Final document	16 horas	M3	PC, Microsoft Word
M5	Reports of the progress	18 horas	M1	PC, Microsoft Word
DV	**Project development**	**132 horas**		**PC, Microsoft Word**
DV1	Web's UI Structure	16 horas	M5	PC, Microsoft Word, Python
DV2	Persistency of the data	12 horas	DV1	PC, Microsoft Word, Python
DV3	Add different models to the app	10 horas	DV2	PC, Microsoft Word, Python
DV4	Model Customization	24 horas	DV3	PC, Microsoft Word, Python
DV5	Pre-processing module	20 horas	DV4	PC, Microsoft Word, Python
DV6	Extraction of classification models	30 horas	DV5	PC, Microsoft Word, Python
DV7	Post processing module	20 horas	DV6	PC, Microsoft Word, Python
DC	**Project documentation**	**168 horas**		**PC, Microsoft Word**
DC1	Technical documentation	116 horas	DV1	PC, Microsoft Word
DC2	Write App documentation	20 horas	DC1	PC, Microsoft Word
DC3	Conclusion and further work	12 horas	DC2	PC, Microsoft Word
DC4	Final review	20 horas	DC3	PC, Microsoft Word
	Total:	**394 horas**		**PC, Microsoft Word**

Table 5.1: Task to perform

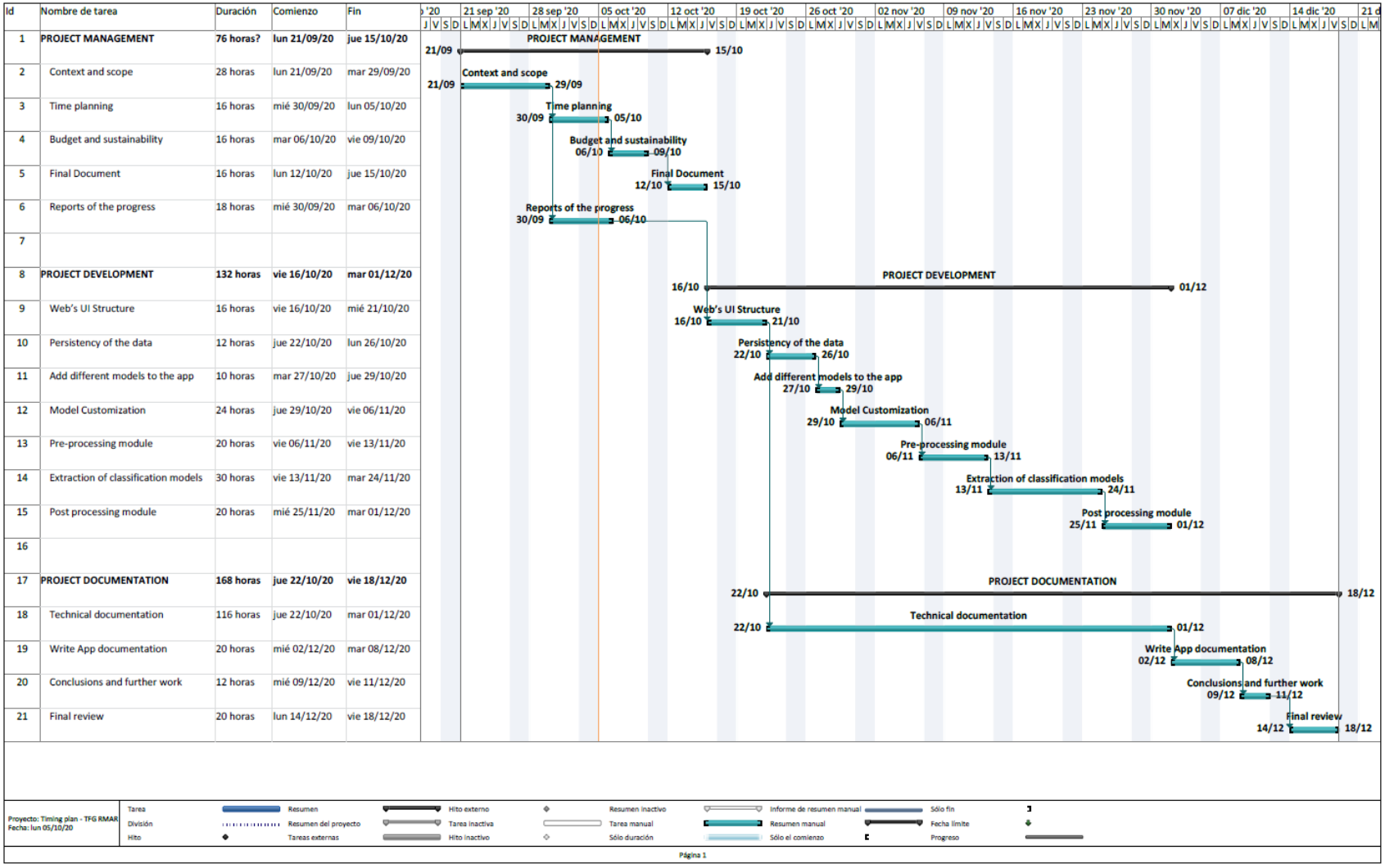

Figure 5.1: Gantt Diagram

5.4. Risk management: alternative plans and obstacles

- **Technical research taking too long (Medium risk):** There are some algorithms new to me, in the task **DV6**, and it is possible that any of them could be more complicated than I expect. This would result in some hours lost looking for the right way to implement it.

  It would be a serious problem if too many hours were lost because of this issue; in this case the alternative plan would be looking for another algorithm that could replace it.

  These algorithms are a small part of the thesis, it would be nice to have them, but in the worst-case scenario not having one of them would not impact the final result by much.
- **Design Problems (Medium risk):** There is the possibility of not being able to make a design in a certain way. This is because the Dash library is a wrapper for web developing specialized in graphics for Python, also the people programming Dash has taken some design decisions that either you like it or you don't. Therefore, if you wanted to design some component of your web in a certain way, but it is not available you need to redesign at least that component, and maybe this has some implications on previous or following designs.

  The best thing we can do to minimize this problem's impact of our project is to do the design as soon as possible, and this is why the task **DV1** is the first development task.
- **Bugs (Medium-Low risk):** Bugs are always there and can delay a project by some hours depending on the importance of the task and the difficulty to solve the bug. There's room for fixing bugs in the schedule, but if the code is as clean as possible and documenting while you program the probability to have important bugs is drastically reduced. Also, new functionalities will be tested when implemented and documented.
- **Documentation issues (Low risk):** As shown in **Figure 5.1: Gantt Diagram** the documentation task is a big time-consuming task, so it has to be treated carefully. The best solution to minimize the risk of not having a good

documentation is to manage well the time, in this case it was decided to document as soon as possible, which is right after a new functionality is programmed.

5.5. Planning Changes

5.5.1. Follow up milestone

There were several changes, most of them are related to miscalculation tasks complexity, but there's one obstacle that was not expected. The changes were the next ones:

- By December of 2020, I was finishing the last exams of the degree, but they ending up being harder than I expected, therefore I needed to invest more hours than expected, the number of hours were significant enough to prevent me from catching up or ending the project in time (in the same term). In order to reschedule and finish the project, I had to postpone to project to the next term. The cost of this obstacle is approximately two weeks of work, the weeks I needed to prepare and pass the exams.

- New tasks appeared; those tasks are related to the persistence of the project. We need the concept of project, with the options of create, load, save and exit. This allows the user to work on different projects and switch between them. This task was necessary and it was not directly added to the planning, so it increases the cost of the project, it also increases by a little the cost of the persistency task.

- The task "DV6 – Extraction of classification models" is harder than expected, it is necessary to add all the model parameters to the customization tab as inputs in order for the user to properly modify and create the model, this was maybe 10 hours or more. The only alternative to this implementation was to pass all the parameters in one input, separated by a comma or some

special character, but this alternative was not a valid option since it went against the idea of the project, which is make easy and accessible.

- The documentation tasks DC1 and DC2 changed, they become 'follow up documentation', 'final documentation' and 'Alerts and User guidance' improving the project helping the user dynamically (This last task will be implemented in parallel of the development tasks). We realise the number of hours needed in order to create a complete and clear documentation for the user is too high, with every detail of how the application works, and also that the objective of the project is not to create that type of documentation, other aspects are required. This project requires some functional documentation, with a resume of the application, where we can include details of how to use it; It also requires some project management details. Therefore, because all of the reasons mentioned before, it is not realistic to invest that much time creating another version of the documentation, with only technical details. On the other hand, it is a fact that every software needs a learning curve, and some explanation is needed for the user to understand how to use the application. In order to solve that need, we have the final documentation, which will have a resume of the project; a resume of how to use the application and the web app will be improved with some notifications and messages that will be sent to the user for guidance purposes and warnings every time a mistake is made.

The new estimated tasks planning is the following (changes appear in italics):

ID	Name	Time(h)	Predecessors	Resources
M	**Project Management**	**112 hours**		**PC, Microsoft Word**
M1	Context and scope	28 hours		PC, Microsoft Word
M2	Time planning	16 hours	M1	PC, Microsoft Word, Microsoft Project
M3	*Reports of the progress*	*28 hours*	*M2*	PC, Microsoft Word, Microsoft Project
M4	Budget and sustainability	16 hours	M3	PC, Microsoft Word, Microsoft Excel
M5	Final document	16 hours	M4	PC, Microsoft Word
M6	*Final Report*	18 hours	M5	PC, Microsoft Word
DV	**Project Development**	**226 hours**		**PC, Microsoft Word**
DV1	Web's UI Structure	50 hours	M5	PC, Microsoft Word, Python
DV2.1	Persistency of the data	*24* hours	DV1	PC, Microsoft Word, Python
DV2.2	*Project Menu implementation*	*18* hours	*DV2.1*	*PC, Microsoft Word, Python*
DV3	Add different models to the app	*18* hours	DV2.2	PC, Microsoft Word, Python
DV4	Model Customization	36 hours	DV3	PC, Microsoft Word, Python
DV5	Pre-processing module	30 hours	DV4	PC, Microsoft Word, Python
DV6	Extraction of classification models	*50* hours	DV5	PC, Microsoft Word, Python
DV7	Post processing module	30 hours	DV6	PC, Microsoft Word, Python
DV8	*Alerts and User guidance*	*15* hours	-	*PC, Microsoft Word, Python*
DC	**Project Documentation**	**133 hours**		**PC, Microsoft Word**
DC1	Follow up Documentation	60 hours	DV1	PC, Microsoft Word
DC2	Final documentation	36 hours	DC1	PC, Microsoft Word
DC3	Conclusion and further work	17 hours	DC2	PC, Microsoft Word
DC4	Final review	20 hours	DC3	PC, Microsoft Word
	Total:	**471 hours**		**PC, Microsoft Word**

Table 5.2: Tasks to perform, with changes

5.5.2. Final milestone

All of the objectives have been achieved, but some little changes were done, some adjustments to the amount of hours worked have been done in the Table 2, in order to be more precise. Regarding to the user interface, there are mainly little tweaks or improvements, but still, they are worth mentioning.

- In the feature extraction tab instead of using PCA, proposed as an example in the objectives section, RFECV was used (Recursive Feature Elimination and Cross-Validated selection). This option was chosen over PCA mainly because it allowed me to plot the variable importance, PCA in the other hand, tries to find the principal components of the dataset, therefore it does not make sense to plot the importance of it feature.
- Moving into the user interface, the following changes were added:
    - Axis for the Confusion Matrix are displayed in a more intuitive way (True positive diagonal goes from bottom-left of the matrix to the top-right)
    - The Feature Selection plots are moved to the Pre-Processing tab.
    - Adding the "Export Model" option, that allows the user to download the model.

Below you can see the final Gantt diagram, once every task has been performed and estimated, the final diagram includes the examination period and it is possible to see which tasks were done before and after that period.

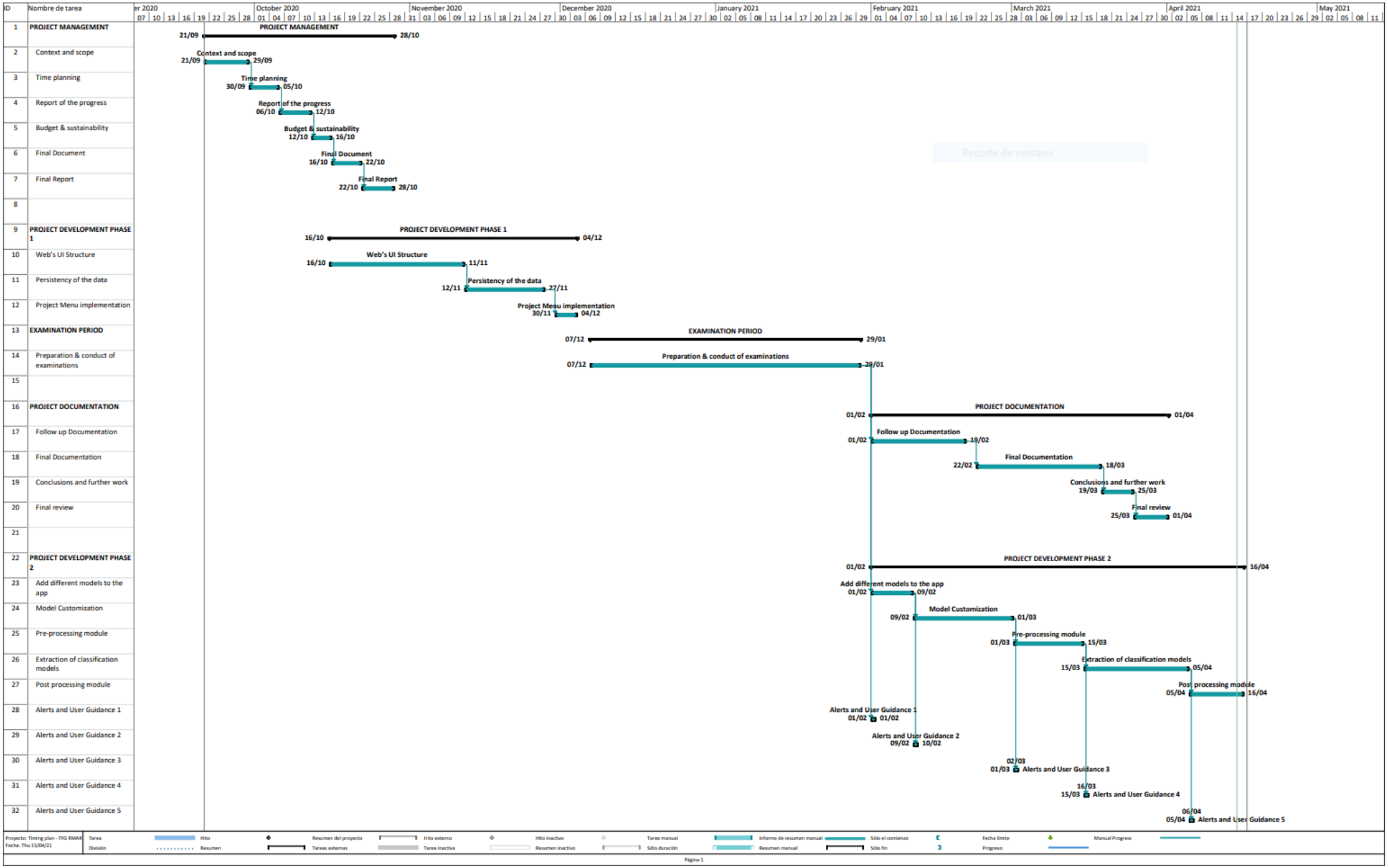

Figure 5.2: Final Gantt

5.6. Development state

5.6.1. Follow up milestone

All of the Project Management tasks have been done, regarding Project Development it is currently in progress the tasks DV7, with the exception of one complementary tab in the DV5 task. The task DV8 is in continuous progress due to the nature of it, its purpose is to improve the user interface to help the user in real time, providing him feedback on its actions.

Finally, from the Project Documentation it remains pending to do the tasks DC2, DC3 and DC4, which are related to the ending of the project.

5.6.2. Final milestone

All the objectives and tasks of the project have been achieved.

6. Economic, sustainability and legislative analysis

6.1. Economic analysis

In order to identify the costs, we need to take into account the next things. Firstly, each task of the project has different requirements and responsibilities, therefore different people should carry on that task and the cost of that activity would be different.

In this project the responsibility is mine and I will do these tasks, but the real cost of the project will be computed as it should be done, assuming that the work load is properly divided.

6.1.1. Identification of costs

The different job positions were defined in 5.2, keep in mind that the names of those job positions may vary depending on the team or the company, we will focus on how much they are paid, it will be specified in euros paid per hour.

- **Consultant:** Since the responsibility and the knowledge needed to be Consultant is way higher to be just a Programmer, we will assume that the **Consultant** is paid twice as much, based on Glassdor [28].
- **Programmer:** We will assume that the Programmer is paid 10€/hour, as it is established in the FIB's cooperation agreement for Industrial Practices [29].

On the other hand, the amortisation of the tools used in this project and indirect costs will be computed.

Budget Estimation

Activity	Amount (€)	Observations
M1 - Context and scope	560,00 €	Consultant, 28 hours
M2 - Time planning	320,00 €	Consultant, 16 hours
M3 - Budget and sustainability	320,00 €	Consultant, 16 hours
M4 - Final document	320,00 €	Consultant, 16 hours
M5 - Reports of the progress	180,00 €	Programmer, 18 hours
DV1 - Web's UI Structure	180,00 €	Programmer, 14 hours; Consultant, 2 hours
DV2 - Persistency of the data	150,00 €	Programmer, 9 hours; Consultant, 3 hours
DV3 - Add different models to the app	110,00 €	Programmer, 9 hours; Consultant, 1 hours
DV4 - Model Customization	300,00 €	Programmer, 18 hours; Consultant, 6 hours
DV5 - Pre-processing module	200,00 €	Programmer, 20 hours
DV6 - Extraction of classification	300,00 €	Programmer, 30 hours
DV7 - Post processing module	200,00 €	Programmer, 20 hours
DC1 - Technical Documentation	1.320,00 €	Programmer, 100 hours; Consultant, 16 hours
DC2 - Write app documentation	220,00 €	Programmer, 18 hours; Consultant, 2 hours
DC3 - Conclusion and further work	160,00 €	Programmer, 12 hours; Consultat, 2 hours
DC4 - Final review	200,00 €	Programmer, 20 hours
Total CPA	**5.040,00 €**	**· Total CPA: Total personnel costs by activity (Gantt activities)**
Hardware		
Desktop Computer	128,80 €	Purchase price: 1400€, Estimated life: 7000 hours
Acer Laptop	97,89 €	Purchase price: 760€, Estimated life: 5000 hours
BenQ Monitor	1,50 €	Purchase price: 200€, Estimated life: 40000 hours
Mouse	12,00 €	Purchase price: 80€, Estimated life: 2000 hours
Keyboard	0,45 €	Purchase price: 30€, Estimated life: 20000 hours
WD Elements HDD	6,27 €	Purchase price: 100€, Estimated life: 6000 hours
Software		
Microsoft Office	28,00 €	4 month subscription
Microsoft Project	33,60 €	4 month subscription
Space		
Desk	28,57 €	
Chair	14,29 €	
Internet	240,00 €	4 months
Electricity	216,00 €	4 months
Total GC	**807,36 €**	· Total Costs imputed generically (not detailed by activity)
Total Cost (Total CPA + Total GC)	**5.847,36 €**	· Total Costs
Contingency	877,10 €	· Fixed a contingency level (understood as a safety margin) for example of 15%, is calculated as: (Total CPA + Total CG)*15%
Total DC+IC + Contingency	**6.724,47 €**	
Research taking too long	3,00 €	(Cost=15; risk=20%)
Design problems	4,50 €	(Cost=15; risk=30%)
Bugs	1,50 €	(Cost=5; risk=30%)
Im	0,80 €	(Cost=8; risk=5%)
Total incidentals (or unforseen costs):	**98,00 €**	
TOTAL:	**6822,47 €**	This is the total cost: CPA+CG+Contingency+Incidentals

Figure 6.1: Costs Table

Final Budget

Activity	Amount (€)	Observations
M1 - Context and scope	560,00 €	Consultant, 28 hours
M2 - Time planning	320,00 €	Consultant, 16 hours
M3 - Reports of the progress	280,00 €	Programmer, 28 hours
M4 - Budget and sustainability	320,00 €	Consultant, 16 hours
M5 - Final document	320,00 €	Consultant, 16 hours
M6 - Final report	180,00 €	Programmer, 18 hours
DV1 - Web's UI Structure	520,00 €	Programmer, 48 hours; Consultant, 2 hours
DV2.1 - Persistency of the data	260,00 €	Programmer, 22 hours; Consultant, 2 hours
DV2.2 - Project Menu implementation	200,00 €	Programmer, 16 hours; Consultant, 2 hours
DV3 - Add different models to the app	200,00 €	Programmer, 16 hours; Consultant, 2 hours
DV4 - Model Customization	400,00 €	Programmer, 32 hours; Consultant, 4 hours
DV5 - Pre-processing module	320,00 €	Programmer, 28 hours; Consultant, 2 hours
DV6 - Extraction of classification	540,00 €	Programmer, 46 hours; Consultant, 4 hours
DV7 - Post processing module	320,00 €	Programmer, 28 hours; Consultant, 2 hours
DV7 - Alerts and user guidance	160,00 €	Programmer, 14 hours; Consultant, 1 hours
DC1 - Follow up Documentation	650,00 €	Programmer, 55 hours; Consultant, 5 hours
DC2 - Final documentation	370,00 €	Programmer, 33 hours; Consultant, 3 hours
DC3 - Conclusion and further work	190,00 €	Programmer, 15 hours; Consultat, 2 hours
DC4 - Final review	230,00 €	Programmer, 17 hours; Consultat, 3 hours
Total CPA	**6.340,00 €**	**· Total CPA: Total personnel costs by activity (Gantt activities)**
Hardware		
Desktop Computer	128,80 €	Purchase price: 1400€, Estimated life: 7000 hours
Acer Laptop	97,89 €	Purchase price: 760€, Estimated life: 5000 hours
BenQ Monitor	1,50 €	Purchase price: 200€, Estimated life: 40000 hours
Mouse	12,00 €	Purchase price: 80€, Estimated life: 2000 hours
Keyboard	0,45 €	Purchase price: 30€, Estimated life: 20000 hours
WD Elements HDD	6,27 €	Purchase price: 100€, Estimated life: 6000 hours
Software		
Microsoft Office	28,00 €	4 month subscription
Microsoft Project	33,60 €	4 month subscription
Space		
Desk	28,57 €	
Chair	14,29 €	
Internet	240,00 €	4 months
Electricity	216,00 €	4 months
Total GC	**807,36 €**	· Total Costs imputed generically (not detailed by activity)
Total Cost (Total CPA + Total GC)	**7.147,36 €**	· Total Costs
Contingency	1.072,10 €	· Fixed a contingency level (understood as a safety margin) for example of 15%, is calculated as: (Total CPA + Total CG)*15%
Total DC+IC + Contingency	**8.219,47 €**	
Research taking too long	3,00 €	(Cost=15; risk=20%)
Design problems	4,50 €	(Cost=15; risk=30%)
Bugs	1,50 €	(Cost=5; risk=30%)
Im	0,80 €	(Cost=8; risk=5%)
Total incidentals (or unforseen costs):	**9,80 €**	
TOTAL:	**8.229,27 €**	This is the total cost: CPA+CG+Contingency+Incidentals

Figure 6.2: Final Costs Table

If we compare the final budget with the estimated one, we can see that the final cost is approximately 1000€ higher than expected, this is in one hand due to the changes described in section 5.5, but mainly due to the initial estimation was underestimating the true cost of the tasks. In order to not repeat this error again, a first estimation needs to be made, and later on add a safety margin, for example a 15%.

6.1.2. Cost estimates

As shown in Figure 6.1: Costs Table, the cost of the project is divided into activities (CPA), other elements that increase the cost (GC) and finally we add a contingency margin of 15% and an incidental amount. The cost of the project can be computed by the next formula:

$$Total\ Cost = CPA + GC + (CPA + GC) * 0.15 + Incidentals$$

On the cost per activity section, the amount is a combination of the hours of each type of worker, e.g. DV1's cost: 220€ = 14h*10€ + 2h*20€.

The consultant is paid that much because he has an important position in the project and a lot of knowledge, the programmer in the other hand can be an intern with informatics knowledge, which is paid 10€/h.

Following with the General Cost, the hardware section are tools used in the development of the project, to calculate its cost relative to this project we need to use the amortization formula.

$$Amortization = \frac{purchasePrice * UsedHours}{EstimatedLifeHours}$$

The Desk and the Chair also follow the amortization formula, but when it comes to Software or internet and electricity, we measure the cost in a different way. The resources are paid monthly, so the cost is simply a multiplication of the months they were used by the price per month.

Finally, we need to mention the incidental costs. In order to calculate the cost of a possible incident we need to evaluate how probable is for an incident to happen and

how much time will it take to fix it. Once we thought that we just apply the next formula:

$$IncidentCost = recoverCost * incidentRisk$$

All the project costs have been calculated using the mentioned formulas with Microsoft Excel [12], generating the table shown in Figure 6.1: Costs Table.

6.1.3. Management control

In order to control the budget and ensure the project's planning, we need to define some metrics and procedures that help us detect deviations.

First is to annotate each step along the project, if we finish a task, we need to calculate the time and resources needed to finish that task to quantify the difference with our initial estimation.

Since the costs are divided as we did in section 6.1.1 the budget control will be divided too. First, we encounter the project's activities, therefore in order to compute its deviation we need to do a sum of deviations.

- **Activity deviation:** This metric shows us how is the performance of activities going. A positive Activity deviation means activities are taking longer than expected to finish, this mean either the estimation wasn't realistic enough or a low performance of the responsible person.

$$Activity\ deviation = \sum_{a=1}^{n} \big(cost(a) - costEstimation(\mathrm{a})\big) * hoursActivity_a$$

- **Amortization deviation:** This metric allows us to analyse the usage of the necessary tools of the project. In this case a positive deviation means those tools have been used more than expected, probably because a bad estimation.

$$Amortization\ Deviation = \sum_{t=1}^{n} (cost(\mathrm{t}) - costEstimation(\mathrm{t})) * hoursUsed_t$$

- **Incident deviation:** If this metric is negative means that no incident has happened or that its cost wasn't that high. In my project this metric should be

negative because the probability of a risk to happen is always less than 0.5, this means it's more likely to **not** happen.

$$Incident\ Deviation = \sum_{i=1}^{n} (cost(\mathrm{i}) - costEstimation(\mathrm{i})) * hoursUsed_i$$

- **Total cost deviation:** Overall metric that indicates how much the project ended up costing.

$$Total\ Cost\ Deviation = \sum_{t=1}^{n} (cost(\mathrm{t}) - costEstimation(\mathrm{t})) * hoursUsed_t$$

Note: I calculated the deviation of costs relative to its estimation, therefore when a cost is higher than expected is positive, and when is less than expected is negative. To simplify, a higher deviation means a higher cost in euros, and a lower deviation results in a lower cost in euros.

6.2. Sustainability report

6.2.1. Economic dimension

- **Have you estimated the cost of undertaking the project (human and material resources)?**

  The cost estimation of the project is done in section 0, it has been analysed both human and material resources. I think the cost of the project is reasonable for its scope.

- **What decisions have you taken to reduce the cost? Have you quantified these savings? Is the expected cost similar to the final cost? Have you justified any differences (lessons learnt)?**

  Just one decision was taken to reduce the cost, it was related to documentation and it is explained in section 5.5. The documentation hours are reduced to 133 from 168 estimated, using the tables of budget to calculate the cost in euros it ends up being a saving of 460€.

  The final cost was higher than expected, this is because some unexpected tasks needed to be added to achieve the project goals, and also the costs were

underestimated. The lesson learnt is to add a safety margin to the estimation of hours, this way if some unexpected tasks appear or we slightly underestimate any tasks this will mitigate the cost difference.

- **How is the problem that you wish to address resolved currently (state of art)?**

  The closest solution to what I want to address is Orange [8], but it is a Desktop application instead of a web one.

- **What cost do you estimate the project will have during its useful life? Could this cost be reduced to increase viability?**

  The cost of the project once it has been developed is minimum, it only requires of a computer to run and obviously electricity that powers it. It would be hard to reduce the cost even more.

- **Have you considered the cost of adaptations/updates/repairs during the useful life of the project?**

  We could try to serve the app in a server and try to make it public, this would need an update in order to adapt the app security increasing this way the cost. An extensive study would be need in order to estimate the cost of all the modifications and updates, and carefully analyse if we could somehow earn money, regaining the cost of the initial inversion.

- **In what ways will your solution economically improve existing solutions?**

  The fact that my solution involves a web application helps reaching more people since it makes it easier to access the app because the user just needs internet access.

- **Could situations occur that are detrimental to the project's viability?**

  Not in the scope of the actual project, that aims to be a tool that helps people with classification problems. If we think in ways to earn money with this tool there could be the risk of someone developing this kind of web application first than us, or doing a better work, if that happens, we could lose the interest of people and this way will be very hard to earn money, but this is not the objective of this project.

6.2.2. Environmental dimension

- **Have you estimated the environmental impact of undertaking the project?**

  The project's biggest impact on the environment is electrical, also materials like the desktop PC and the laptop could have an impact on the environment if we are not careful when it's useful life ends.

- **Have you quantified the environmental impact of undertaking the project? What measures have you taken to reduce the impact? Have you quantified this reduction?**

  According to the initial estimate the biggest impacts are electricity usage and my desktop PC or laptop ending its useful life. Since both of my computers are still perfectly working, we will focus on electricity usage. In this case since the amount of hours of this project are 644h, the price of the electricity is 28,74€ per 100 kWh and my pc and monitor approximately uses 0,7kWh we have a resultant cost of: 129,56€.

  The measures taken to reduce the impact are optimizing the consume of both the computer and the monitor, this impact hasn't been quantified due to the small impact on the final cost.

- **Have you considered how to minimise the impact for example by reusing resources?**

  We need to recycle components of the PC and the laptop when the time comes, also the electrical consume needs to be reduced, for example by reducing the brightness on the screen and having efficient components for each computer.

- **If you carried out the project again, could you use fewer resources?**

  Yes, I could have restricted myself to only use the computer PC, this will slightly reduce the final cost of the project.

- **How is the problem that you wish to address resolved currently (state of the art)? In what ways will your solution environmentally improve existing solutions?**

  This type of projects doesn't affect much the environment, they just need to be careful with electricity and with their computers once they need to be replaced. My solution for mitigating this problem is the one mentioned above.

- **What resources do you estimate will be used during the useful life of the project? What will be the environmental impact of these resources?**

  The resources in order to use are very minimal, you need to have a computer, but the real cost will be the cost of using that computer during the hours of usage of the project.

- **Will the project enable a reduction in the use of other resources? Overall, does the use of the project improve or worsen the ecological footprint?**

  I think the project will reduce the amount of hours that the final user spends in front of the pc working in classification problems, this will improve the ecological footprint.

- **Could situations occur that could increase the project's ecological footprint?**

  If the cost of the electricity rises the ecological footprint of the project will worsen.

6.2.3. Social dimension

- **What do you think undertaking the project has contributed to you personally?**

  The possibility to work on all of the stages of a big project, including graphics, design and project management. Also, it is my last step for ending the informatics bachelor degree, a very important point in my life.

- **Has undertaking this project led to meaningful reflections at the personal, professional or ethical level among the people involved?**

  It made me realize that a enjoy the process of designing interfaces because it involves a creative process and it ends up being very satisfying to me. Also, I realized the difficulty and the cost of carrying out a project, which is higher than I expected.

- **How is the problem that you wish to address resolved currently (state of art)?**

There are applications, but with they don't reach as much people as they could, and there are some aspects that could be improved.

- **Who will benefit from the use of the project? Could any group be adversely affected by the project? To what extent?**

  This project will mainly benefit beginners in machine learning and also people who spend a lot of time creating classifier models, with this tool they will save many hours of work. I don't think this project will affect adversely anyone, maybe it could add some competition to the existent tools described in the **State of art** section.

- **To what extent does the project solve the problem that was established initially?**

  It mainly solves the case of classification problems, which is what it aims for, but in order to solve the problem fully, and provide a better tool than the existent ones, it will be necessary to expand the project and cover all the alternative problems in machine learning, not only classification.

- **In what ways will your solution socially improve (quality of life) existing? Is there a real need for the project?**

  It would help a lot of people since it will have a good user interface and it is easy to get in touch with. I think there is a need for my project since everybody should be able to take profit of Machine Learning and Data Mining.

- **Could situations occur in which the project adversely affects a specific population segment? Could the project create any kind of dependency that puts users in a weak position?**

  No, the project's objective is to help people, it will be a disaster if it ended up doing the opposite.

  This project tries to be as general as possible so it does not create dependencies to the user, in the alleged case that there existed a weakening dependency this project allows the easy portability, to any other context, of the obtained results.

6.3. Identification of laws and integration

This project is an application in which the user can upload information. Therefore, data laws need to be reviewed. The laws that may affect this project are as follows:

6.3.1. Intellectual property

Intellectual property [26] is made up of a series of rights of a personal and/or patrimonial nature that attribute to the author and other owners the disposition and exploitation of their works and services.

To comply with this law, it is necessary to cite all sources of information used in the project, and in the case of software, to respect the license and the terms of use.

6.3.2. LOPD

The Organic Law on Protection of Personal Data [27] guarantees and protects the processing of personal data, public liberties and fundamental human rights, and especially of personal and family honour and privacy.

This law obliges all individuals, companies and organizations that possesses personal data to comply a series of requirements and apply certain security measures.

In conclusion, intellectual property rights [26] have been respected during the development of this project, citing sources and respecting the licenses and terms of use of the software used.

The organic law on data protection [27] is also respected but because it does not apply. Although this project is a web application, its scope is to work locally on the user's computer and therefore this law would not apply, since the data would not be sent to any external server, the only one who may possess personal information is the user, about himself.

7. Conclusion

I am very satisfied with the obtained results of this project, all the proposed objectives were achieved successfully, even though some new tasks appeared, making the project harder than the initial estimation, a solution has been found with a reasonable cost.

I consider the User Interface to be very intuitive and appealing, with an elegant touch.

After all the worked time, I would like to add some future work ideas that I think will help the project become a better version of itself. I will mainly focus on improving two points, broaden the amount of people that can use the application and making the application capable of solving other machine learning problems.

For the first improvement, I would start creating the concept of user in the application, adding a login, a database and strengthen the protection of the user information.

In the second case, I would start with regression, in order to finish with the supervised learning, but we could also start the other way around, the final objective will be implementing both.

Bibliography

[1] "UCI Machine Learning Repository: Iris Data Set." https://archive.ics.uci.edu/ml/datasets/iris (accessed Apr. 17, 2021).

[2] "Dash | Plotly." https://plotly.com/dash/.

[3] "Facultat d'Informàtica de Barcelona |." https://www.fib.upc.edu/ (accessed Sep. 26, 2020).

[4] Google, "Gmail," *01/04/2004*, 2004. www.gmail.com.

[5] "Machine Learning Classifiers. What is classification? | by Sidath Asiri | Towards Data Science." https://towardsdatascience.com/machine-learning-classifiers-a5cc4e1b0623 (accessed Sep. 27, 2020).

[6] "RapidMiner | Best Data Science & Machine Learning Platform." https://rapidminer.com/ (accessed Sep. 28, 2020).

[7] "BigML.com." https://bigml.com/ (accessed Sep. 28, 2020).

[8] "Orange Data Mining - Data Mining." https://orange.biolab.si/ (accessed Sep. 28, 2020).

[9] "KNIME | Open for Innovation." https://www.knime.com/ (accessed Sep. 28, 2020).

[10] "Microsoft Word: software de procesamiento de textos | Office." https://www.microsoft.com/es-es/microsoft-365/word (accessed Oct. 04, 2020).

[11] "Software de administración de proyectos | Microsoft Project." https://www.microsoft.com/es-es/microsoft-365/project/project-management-software (accessed Oct. 04, 2020).

[12] "Microsoft Excel, software de hojas de cálculo, prueba gratuita de Excel."

https://www.microsoft.com/es-es/microsoft-365/excel (accessed Oct. 04, 2020).

[13] Microsoft, "Skype, Communication tool for chats and calls," [Online]. Available: https://www.skype.com/es/.

[14] "Welcome to Python.org." https://www.python.org/ (accessed Mar. 03, 2021).

[15] "Visual Studio Code - Code Editing. Redefined." https://code.visualstudio.com/ (accessed Mar. 03, 2021).

[16] Google, "Google Drive," [Online]. Available: drive.google.com.

[17] "GitHub," [Online]. Available: github.com.

[18] "Trello," [Online]. Available: https://trello.com/es.

[19] "Pima Indians Diabetes Database | Kaggle." https://www.kaggle.com/uciml/pima-indians-diabetes-database (accessed Apr. 17, 2021).

[20] "Heart Disease UCI | Kaggle." https://www.kaggle.com/ronitf/heart-disease-uci (accessed Apr. 17, 2021).

[21] "R: The R Project for Statistical Computing." https://www.r-project.org/ (accessed Mar. 12, 2021).

[22] "scikit-learn: machine learning in Python — scikit-learn 0.23.2 documentation." https://scikit-learn.org/stable/ (accessed Oct. 04, 2020).

[23] "Welcome to Flask — Flask Documentation (1.1.x)." https://flask.palletsprojects.com/en/1.1.x/ (accessed Mar. 12, 2021).

[24] "Plotly JavaScript Graphing Library | JavaScript | Plotly." https://plotly.com/javascript/ (accessed Mar. 12, 2021).

[25] "React – A JavaScript library for building user interfaces."

https://reactjs.org/ (accessed Mar. 12, 2021).

[26] "La propiedad intelectual en general | Ministerio de Cultura y Deporte." https://www.culturaydeporte.gob.es/cultura/propiedadintelectual/la-propiedad-intelectual/preguntas-mas-frecuentes/la-propiedad-intelectual.html (accessed Mar. 17, 2021).

[27] "Ley Orgánica de Protección de Datos de Carácter Personal - Wikipedia." https://en.wikipedia.org/wiki/Ley_Orgánica_de_Protección_de_Datos_de_Carácter_Personal (accessed Mar. 17, 2021).

[28] "Glassdoor." https://www.glassdoor.es/Sueldos/barcelona-consultant-sueldo-SRCH_IL.0,9_IM1015_KO10,20.htm.

[29] "Industrial Practices | FIB - Barcelona School of Informatics." https://www.fib.upc.edu/en/companies/industrial-practices (accessed Oct. 11, 2020).