MÁSTER UNIVERSITARIO EN INGENIERÍA DE TELECOMUNICACIONES

PREDICTIVE AND DESCRIPTIVE LEARNING

HALF-TERM ACTIVITIES REPORT:

Developing Machine Learning Models

Case Study: Obstructive Sleep Apnea

Jaime Pérez Sánchez

Course 2019/20

Table of Contents

1. Problem Description

1.1. Case Study: Obstructive Sleep Apnea

Obstructive Sleep Apnea (OSA) is the most common type of sleep apnea and a potentially serious sleep disorder. It is characterized by the appearance of repeated episodes of (partial or total) upper airway obstruction during sleep. The most commonly used metric to measure the severity of the disease is the Apnea-Hypopnea Index (AHI) and represents the amount of times the patient stops breathing per hour. Typical **symptoms** of OSA are [1]:

- Excessive daytime sleepiness
- Loud snoring
- Morning headache
- Not rested after sleeping
- Abrupt awakenings accompanied by gasping or choking
- Dry mouth or sore throat on awakening
- High blood pressure (hypertension)
- Nighttime sweating
- Decreased libido

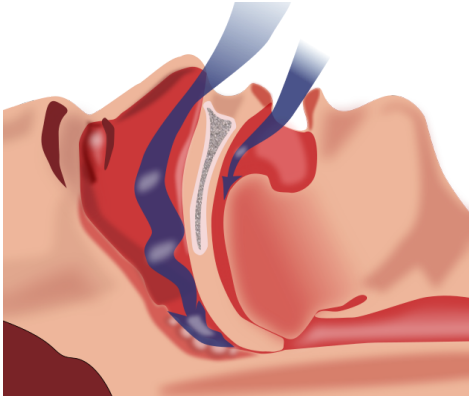

Figure 1. Obstructive Sleep Apnea

Is more likely to suffer from OSA [2] when the patient is overweight or obese, has a long or thick neck, or have smaller airways in the nose, throat or mouth. A larger than average tongue or deviated septum can also block the airway in many people. Other common **risk factors** include:

- Smoking
- Diabetes
- High blood pressure
- Being at risk for heart failure or stroke
- A family story of sleep apnea
- Neck circumference greater than 40 cm
- Be over 40 years of age
- To be a man (Men are twice as likely to suffer OSA)

Diagnosing OSA is not an easy task. The most commonly used method is to stay overnight in a sleep lab with sensors that monitor your body activity (polysomnography). In Spain the waiting list for polysomnography is more than 1 year. With this in mind, the **objective of this project** is to provide additional information to doctors so that they can diagnose the disease earlier and more accurately. In particular, we will try to predict the Apnea-Hypopnea Index (AHI) in patients, taking into account basic clinical information and risk factors. Typically in adults:

- An AHI of less than 5 is considered *normal*
- An AHI between 5 and 15 is considered *mild*
- An AHI between 15 and 30 is considered *moderate*
- An AHI greater than 30 is considered *severe*

1.2. Machine Learning Approach

To attempt to solve the proposed problem will intend to use Machine Learning approaches. In the first place we will try to solve as a **regression** problem where the target variable is the Apnea-Hypopnea Index (AHI), that is, the number of apnea episodes during the patient's sleep per hour. Afterwards, it will be approached as a **classification** problem where we must try to predict whether male patients are healthy (AHI ≤ 10) or suffer from severe sleep apnea (AHI ≥ 30). The Machine Learning models that will be used are those provided by the Python's library *Scikit-learn* [3].

Before the use of Machine Learning models, the used dataset must be cleaned, processed and transformed in order to feed the algorithms. This dataset has been provided in the Moodle website of the subject and contains basic clinical information such as: age, weight, height, cervical perimeter... Also, the target variable that we have to predict: Apnea-Hypopnea Index (AHI); and other data that may be risk factors such as whether the patient is a smoker, other diseases, etc…

For a better generalization and confidence in the obtained results, the cross-validation method **K-Fold** will be used. Through this method, we will divide the dataset into **5** random sets of the same size (20%), and evaluated one by one taking the rest as training data in each case. Therefore, the evaluation metrics will be conducted with the predictions of the entire dataset. Figure 2 shows a schematic illustration of this.

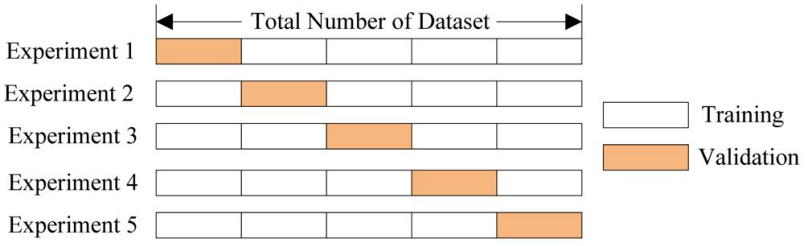

Figure 2. Cross-validation K-Fold diagram

2. Data Wrangling

2.1. Data Description

In this section we will explain and briefly explore the variables that compose the used dataset. First we will explore the dataset with **clinical information**:

- **Patient:** Indicates the patient identification label. This variable could be used as an index. There is a repeated value (P0363) probably due to a data entry error.

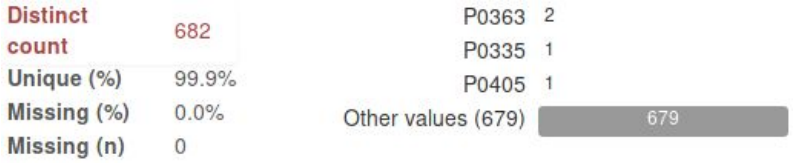

Figure 3. 'Patient' variable analysis

- **Commentaries:** Contains comments about the patient or just an identification code. It does not seem relevant for the predictive model, so we will not use this column.
- **Audios lying:** Seems to indicate only whether audios of the lying patient have been recorded. It does not seem relevant for the predictive model, so we will not use this column.

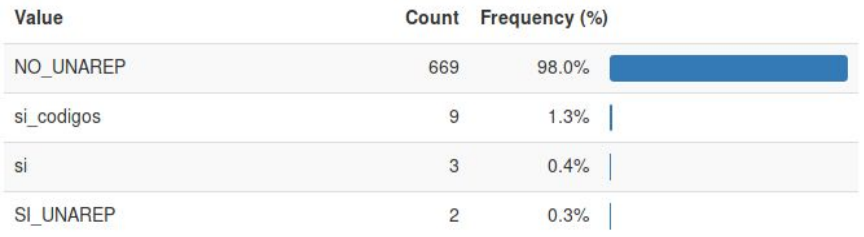

Figure 4. 'Audios lying' variable analysis

- **Photos:** Seems to indicate only whether photos have been taken of the patient. It does not seem relevant for the predictive model, so we will not use this column.

Figure 5. 'Photos' variable analysis

- **Audio fs kHz:** It seems to indicate only the frequency modes in the audio recordings, with 2 possible values. It does not seem relevant for the predictive model, so we will not use this column.

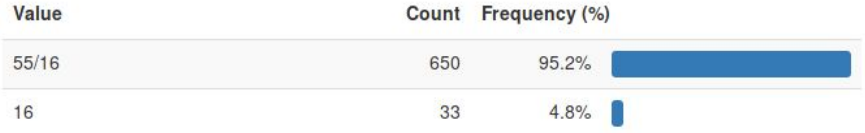

Figure 6. 'Audio fs kHz' variable analysis

- **Gender:** Indicates the gender of the patient. As it is a categorical variable formed by Strings ("hombre" or "mujer"), it will have to be transformed later in order to be introduced to the model.

Value	Count	Frequency (%)
hombre	488	71.4%
mujer	195	28.6%

Figure 7. 'Gender' variable analysis

- **EPWORTH:** Since 91.9% of the values in this column are missing, it has been decided not to use this variable for the model.

Figure 8. 'EPWORTH' variable analysis

- **AHI:** Indicates the number of apnea episodes during sleep per hour of the patient. As explained above, this is the target variable of the model. There is a 5% of lost samples, later it will be decided how to handle those missing data.

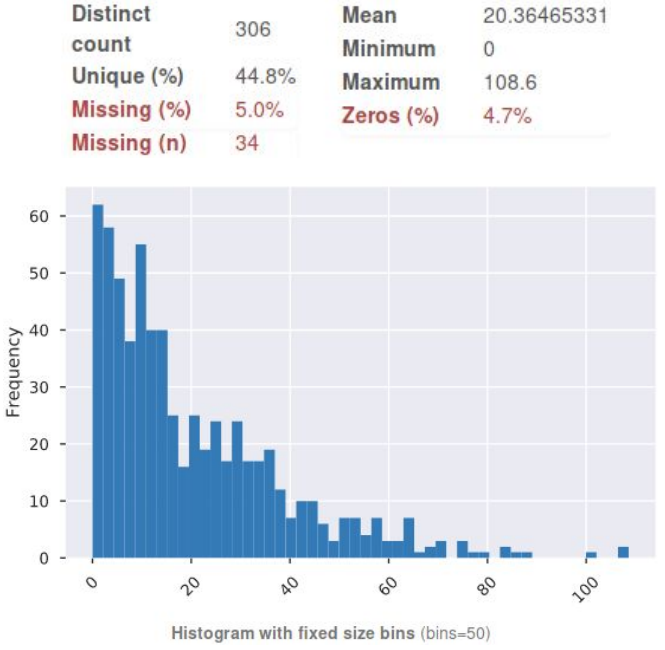

Figure 9. 'AHI' variable analysis

- **AHI Supine:** Indicates the number of apnea episodes during sleep per hour of patient, in supine position. It has been decided that this variable will not be used in the models, as it is included in the AHI column.

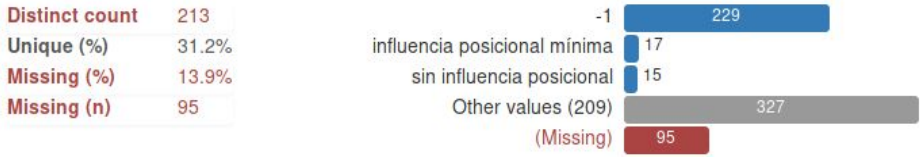

Figure 10. 'AHI Supine' variable analysis

- **AHI Lateral:** Indicates the number of apnea episodes during sleep per hour of patient, in supine position. It has been decided that this variable will not be used in the models, as it is included in the AHI column.

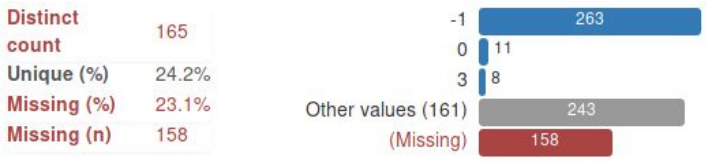

Figure 11. 'AHI Lateral' variable analysis

- **Weight:** Indicates the patient's weight. It has a 1% of lost values and a -1 value, which will be handled later.

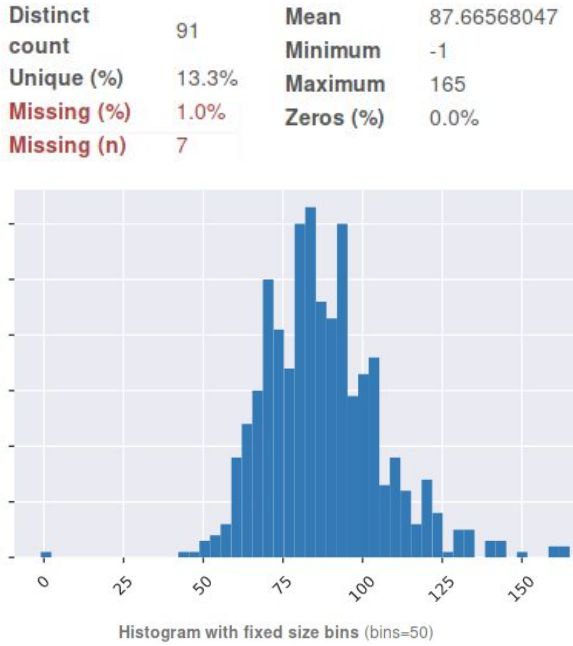

Figure 12. 'Weight' variable analysis

- **Height:** Indicates the patient's weight. It has a 0.9% of lost values and a -1 value, which will be handled later.

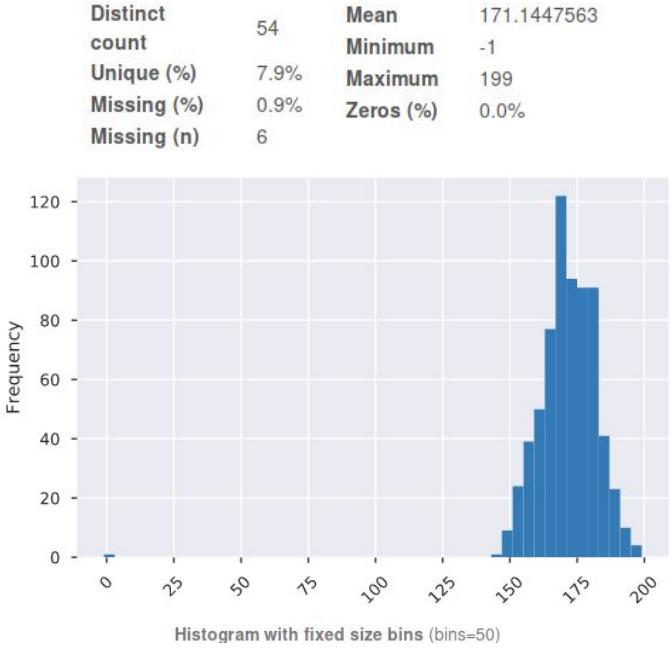

Figure 13. 'Height' variable analysis

- **BMI:** Indicates the patient's Body Mass Index (BMI). All values are -1 (or missing) but, as it is computed from height and weight, we will be able to calculate all the values ourselves later.

Figure 14. 'BMI' variable analysis

- **Age:** Indicates the patient's age. It has a 0.7% of missing values and three -1 values, which will be handled later.

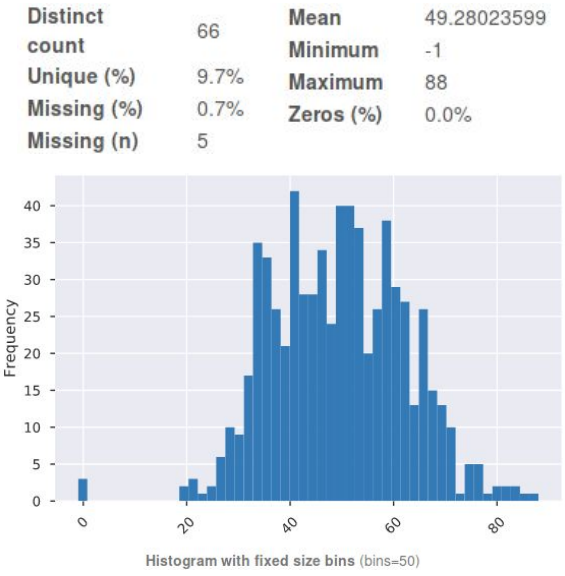

Figure 15. 'Age' variable analysis

- **Cervical Perimeter:** Indicates the cervical perimeter of the patient. It has a 0.7% of missing values and seven -1 values, which will be handled later.

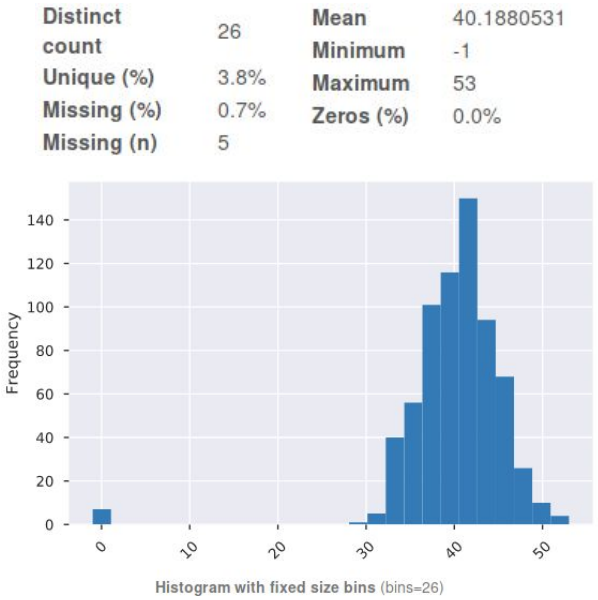

Figure 16. 'Cervical Perimeter' variable analysis

- **Smoker:** Indicates if the patient is a smoker. As it is a categorical variable formed by Strings, it will have to be transformed later in order to be introduced to the model. It has 0.4% lost values and a 2.3% of "ns" (i.e. not know) values that we will also assume as missing.

Value	Count	Frequency (%)
no	373	54.6%
si	165	24.2%
antiguo	119	17.4%
ns	16	2.3%
poco	6	0.9%
si (poco)	1	0.1%
(Missing)	3	0.4%

Figure 17. 'Smoker' variable analysis

- **Snorer:** Indicates if the patient snores while sleeping. It has 0.4% missing values and 25.9% "ns" values assumed as missing, so it has been decided that this variable will not be used in the model.

Value	Count	Frequency (%)
si	466	68.2%
ns	177	25.9%
no	18	2.6%
CPAP	12	1.8%
no con CPAD	4	0.6%
si sin CPAP	1	0.1%
si´(protesis boca para dormir)	1	0.1%
poco	1	0.1%
(Missing)	3	0.4%

Figure 18. 'Snorer' variable analysis

- **Diseases:** Indicates other diseases related to the respiratory system suffered by the patient. As the number of possible diseases indicated is very large, it has been decided not to use this variable for the model, but it is proposed as a future line of work.

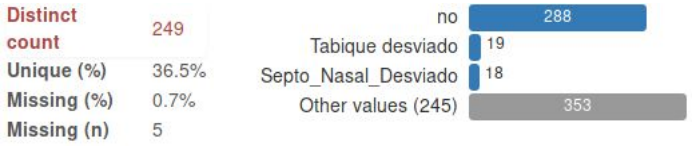

Figure 19. 'Diseases' variable analysis

- **Room/Noise:** It has not been possible to clarify exactly what this variable indicates, it is probably related to the audio recordings of the patients while they sleep. Therefore, it has been decided not to use it in the model, but it is proposed as a future line of work.

Figure 20. 'Room/Noise' variable analysis

- **Image:** It seems to indicate anomalies or peculiarities of patients' photographs. It has been decided not to use this variable for the model.

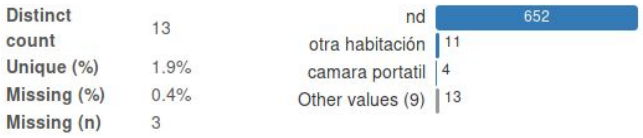

Figure 21. 'Image' variable analysis

- **Dialect:** Indicates the patient's dialect. As the number of possible dialects is very large, and is not considered to be relevant for prediction, it has been decided not to use it in the model.

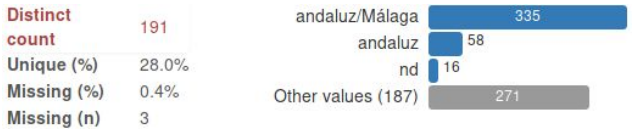

Figure 22. 'Dialect' variable analysis

- **Distance-Ext-Eyes:** It seems to indicate a distance related to the position of the patient's eyes. Since the 55.3% of the values are missing, it has been decided not to use this variable in the model.

Figure 23. 'Distance-Ext-Eyes' variable analysis

- **Distance-Chin-Lobule:** It seems to indicate a distance related to the position of the patient's chin and lobes. Since the 55.3% of the values are missing, it has been decided not to use this variable in the model.

Figure 24. 'Distance-Chin-Lobule' variable analysis

- **Fatigue:** Indicates if the patient is fatigued regularly. Since the 73.8% of the values are missing, it has been decided not to use this variable in the model.

Figure 25. 'Fatigue' variable analysis

- **Focusing:** Indicates if the patient has difficulty concentrating on their daily life. Since the 73.8% of the values are missing, it has been decided not to use this variable in the model.

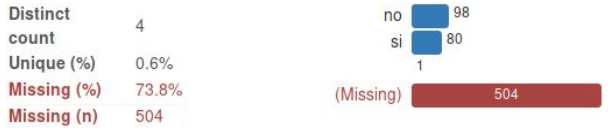

Figure 26. 'Focusing' variable analysis

- **Breathing Loss at Night:** Indicates if the patient has noticed breathing deprivation during sleep. Since the 96.3% of the values are missing, it has been decided not to use this variable in the model.

Figure 27. 'Breathing Loss at Night' variable analysis

- **Hypertension:** Indicates if the patient suffers from hypertension. Since the 96.3% of the values are missing, it has been decided not to use this variable in the model.

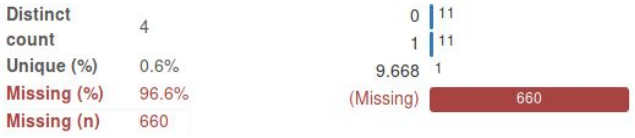

Figure 28. 'Hypertension' variable analysis

- **EstHSOP:** Since 91.9% of the values in this column are missing, it has been decided not to use this variable for the model.

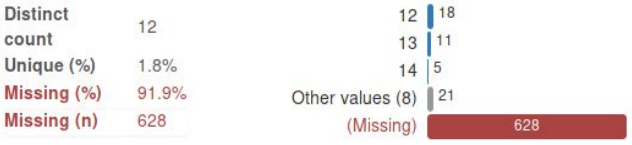

Figure 29. 'EstHSOP' variable analysis

In the Moodle web site of the course it has also been provided another dataset with bandwidths of the male patients, in which Feature Selection techniques could be applied. In this assignment, however, its analysis is proposed as a future line of work.

2.2. Data Preparation

As a result of the analysis of the variables carried out on the clinical dataset, it was finally decided to use the following columns to feed the models:

- Patient
- Gender
- Weight
- Height
- Age
- Smoker
- Cervical Perimeter
- BMI (This column will be computed later with the weight and height data)
- AHI

2.2.1. Missing Values

The first step in data preparation is to handle the **missing values** (or -1). There is a great variety of methods to handle these values, but in this case it has been decided to use the

naive method, which is to eliminate those rows from the dataset. Since the amount of missing values is not excessively high in the selected columns, as we can observe in Table 1 and Figure 30, it is not expected to affect model results significantly.

	Patient	Gender	Weight	Height	Age	Smoker	Cervical	AHI
NaNs	0	0	7	6	5	3	5	34
-1	0	0	1	1	3	0	7	0

Table 1. Missing values on the selected variables (clinical data)

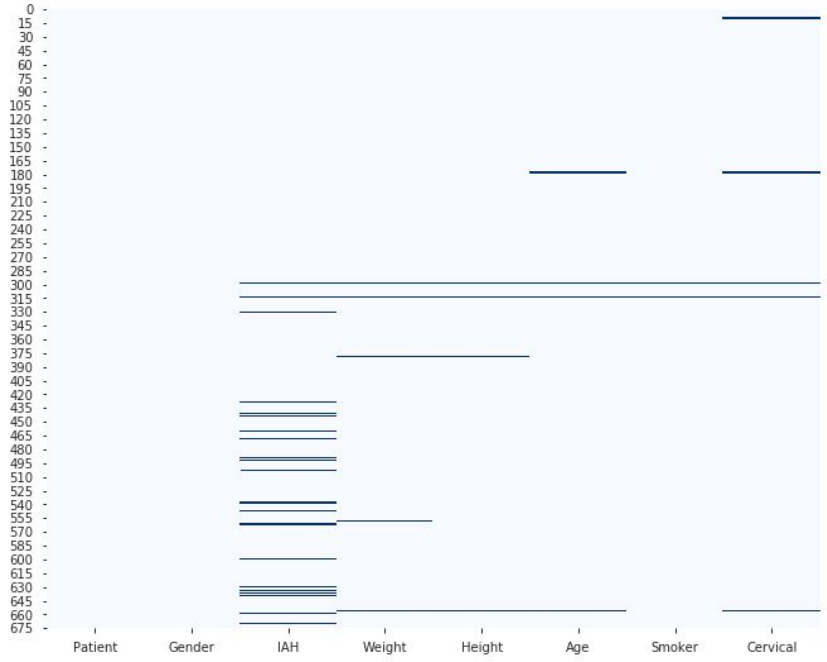

Figure 30. Heatmap of missing values on the selected variables (clinical data)

2.2.2. Label Encoding 'Smoker'

In the analysis carried out previously, it has been observed that "Smoker" is a categorical variable with the values shown in Figure 17. First, the "ns" values have been transformed into NaNs for later deletion. It has also been considered that "poco" and "si (poco)" are actually the same value. Since this is a quantitative variable, that is, it indicates the level at which the patient smokes, it has been considered that the most appropriate is to use Label Encoding as shown in the following:

Label	Code
Non-smoker ("no")	0
Former smoker ("antiguo")	1
Light smoker ("poco")	2
Smoker ("si")	3

Table 2. Label Encoding 'Smoker'

2.2.3. One Hot Encoding ‘Gender’

In the analysis carried out previously, it has been observed that “Gender” is a categorical variable with the values shown in Figure 7 (“hombre” or “mujer”). In this case it is not a quantitative variable, as it only indicates the gender of the patient, so it has been decided to perform One Hot Encoding to this variable. This means to create a new column for each possible value, that is to say, one column to indicate when the patient is a man and another to indicate when it is a woman.

2.2.4. Setting ‘Patient’ to Index

In the analysis carried out previously, it has been observed that in the variable 'Patient' there is a repeated label (‘P0363’). In order to make this variable the index of the DataFrame, we must first replace one of the repeated values with another new label. It has been decided to replace one of them with the unused value 'P9999'.

2.2.5. Computing ‘BMI’

In order to add more information to the models and thus facilitate predictions, a new column 'BMI' has been computed from the Weight and Height data, with the following equation:

$$BMI = \frac{Weight_{[kg]}}{Height^2_{[m]}}$$

2.2.6. Computing ‘log(AHI+1)’

In order to facilitate model predictions, it has been tested to change the AHI target variable by its logarithm. However, when analyzing the results, it did not imply a significant change in the precision of the predictions, so it was finally decided not to use this variable. The justification for adding 1 to the AHI before making the logarithm is to avoid that the 0 values are transformed into ‘-Infinity’.

2.2.7. Computing ‘OSA’ for Classification Models

In the classification models it is intended to predict whether male patients are healthy (i.e. AHI ≤ 10) or have severe OSA (i.e. AHI ≥ 30). For this purpose, a new variable 'OSA' has been computed that only has these two possible values to classify, as shown in Table 3.

Label	Code
Healthy (AHI ≤ 10)	0
Severe (AHI ≥ 30)	1

Table 3. Label Encoding ‘OSA’

3. Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is an approach to analyzing datasets to summarize their main characteristics, discover patterns, spot anomalies, test hypothesis... Often with visual methods. In this case we want to discover if there are direct correlations between the selected variables to feed into the models.

In the problem description we have shown that there are risk factors which lead to a higher AHI such as being a smoker, a high BMI, being over 40 years old… Therefore, what we are trying to observe in this analysis is whether these initial assumptions are accurate for our dataset. Our first approach is to visualize the **correlation matrix**:

Figure 31. Correlation Heatmap on the selected variables (clinical data)

We can remark that:

- There is a certain correlation (~0.4) between the 'AHI' and the variables 'Cervical Perimeter', 'Weight' and 'BMI'.
- The 'Cervical Perimeter' is correlated (~0.6 - 0.7) with 'Weight' and male patients.

A good approach for visualizing the relationship between two variables and their statistical distributions is shown in Figure 32. In this case we are analyzing the relationship between 'AHI' and the variables 'Cervical' and 'Weight'. Where we can observe that there is some linear relationship, but there is also a lot of variance in the data.

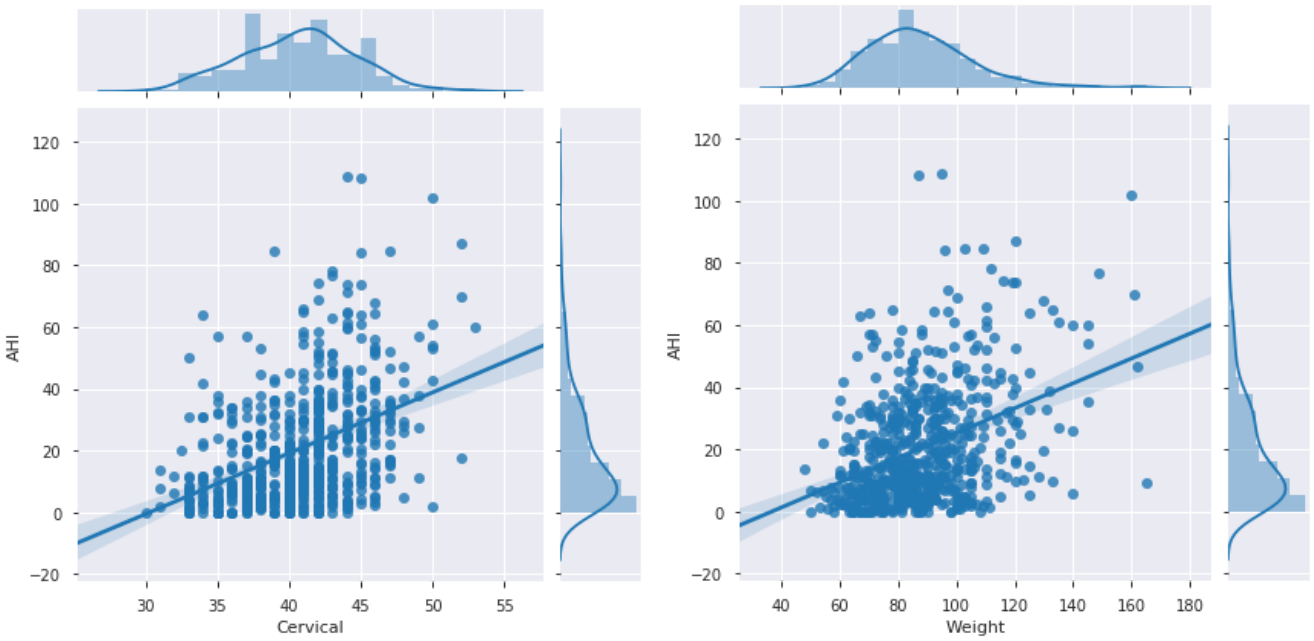

Figure 32. Relationships between ‘AHI’ and the variables ‘Cervical’ and ‘Weight’

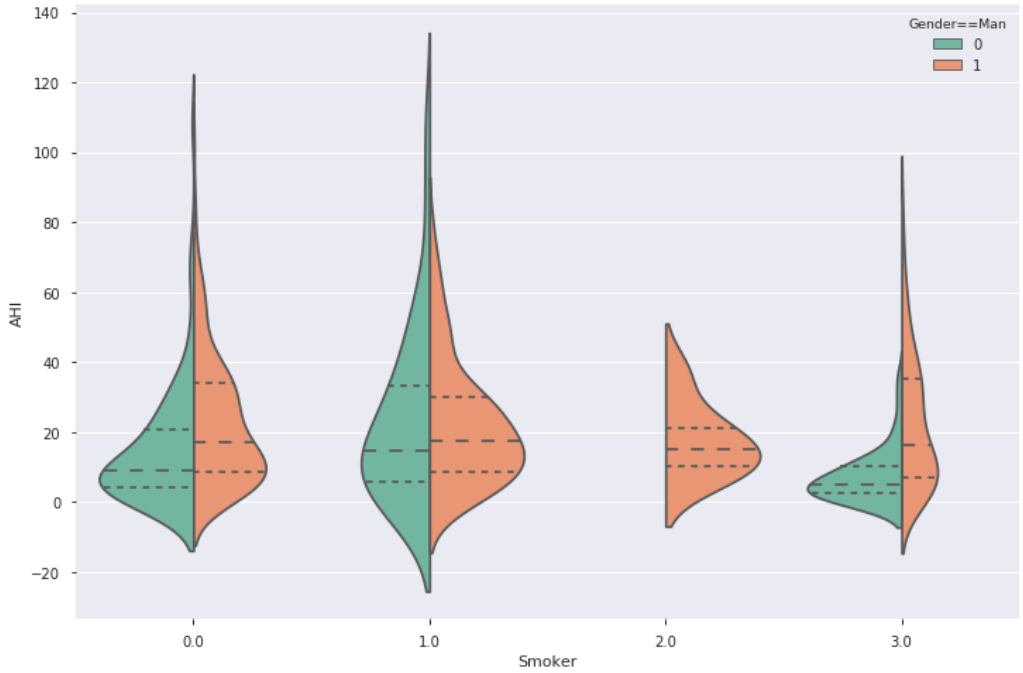

Figure 33. Relationships between ‘AHI’ and the variables ‘Smoker’ and ‘Gender’

Figure 33 shows the relationship between AHI and categories of ‘Smoker’ shown in Table 2. We can notice that, apparently, there is no clear relationship between them.

In Figure 34 and Figure 35 we can observe the relationships between the numerical variables. The categories of the patient's gender and the ‘Smoker’ variable are shown in different colors. We can note that in terms of the patient's gender there are notable separations in some relationships and variables; but as we have deduced previously, there are no clear differentiations with the variable ‘Smoker’.

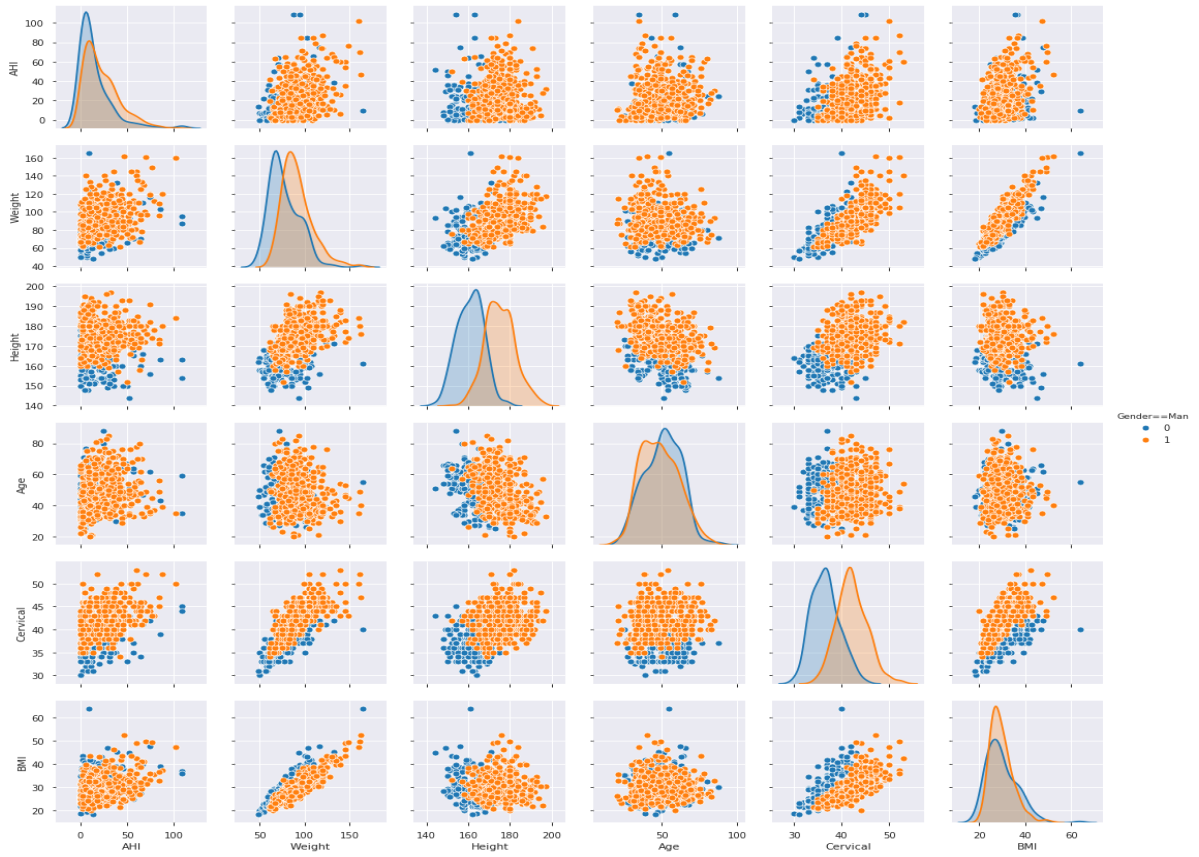

Figure 34. Scatter matrix of the numerical variables (clinical data) splitted by 'Gender'

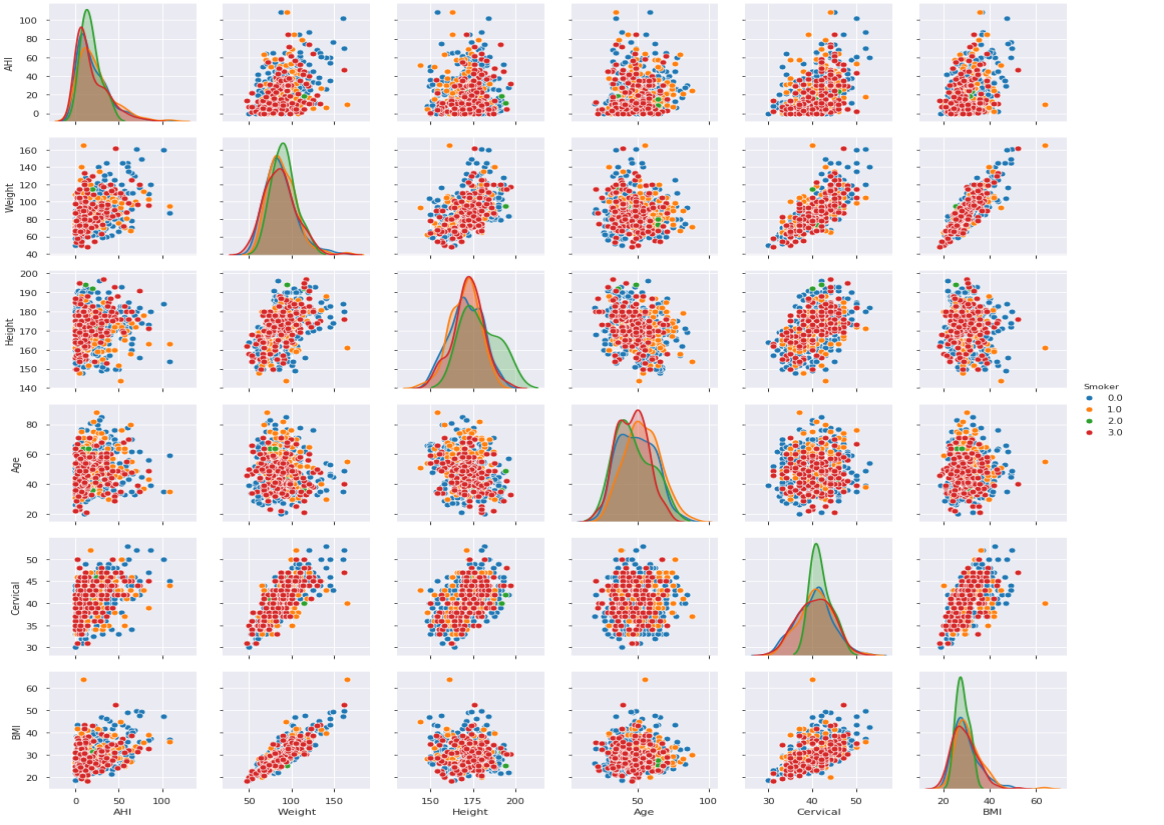

Figure 35. Scatter matrix of the numerical variables (clinical data) splitted by 'Smoker'

For the **classification** models, in which we add the ‘OSA’ column, we can display the following distributions.

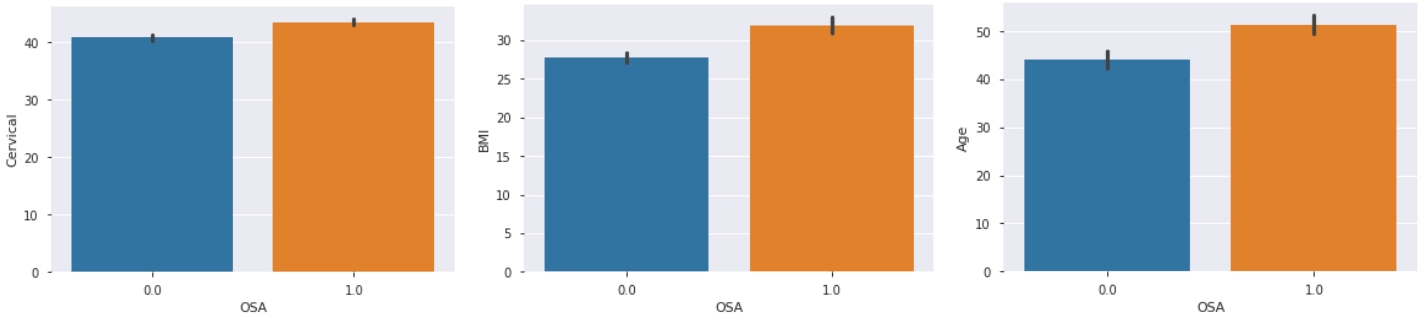

Figure 36. Relationships between ‘OSA’ and the variables ‘Cervical’, ‘BMI’ and ‘Age’

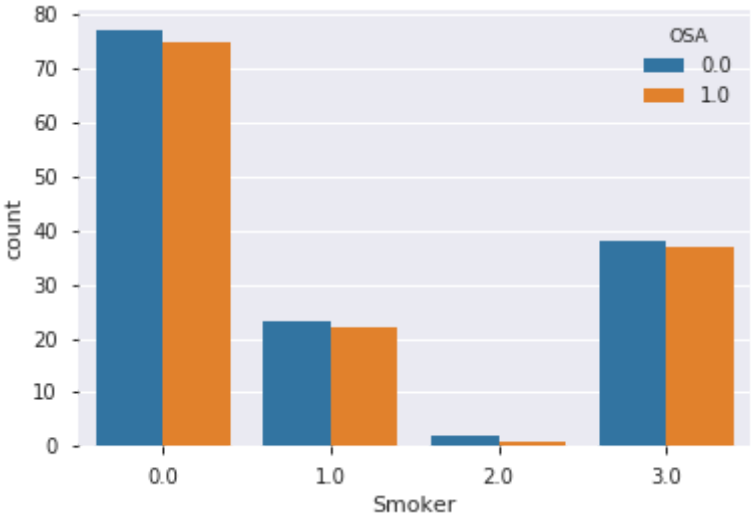

Figure 37. Relationship between ‘OSA’ and the variable ‘Smoker’

In Figure 36 we can appreciate that there is a certain relationship between the 'OSA' variable and the other variables analysed. Although these differences between healthy patients and those with severe OSA are not as evident as expected. In Figure 37 we can also verify that through the different categories of the variable 'Smoker' there is no clear separation between healthy patients and patients with severe OSA, which does not correspond much to the description of the problem.

4. Machine Learning Models

The methodology followed in the implementation of Machine Learning models has been to try to automate the process as much as possible. In the following we will study the data preprocessing methods used and the families of the implemented algorithms, both for the problem of regression and classification. All the code of the project is available in my Github account [4] [5] and in Google Collab [6] [7].

4.1. Data Preprocessing

In general, Machine Learning algorithms benefit from the normalization and transformation of data into training time and performance. It also helps them to process strange data such as outliers. In this project three preprocessing methods have been tested, in addition to introducing the raw data, to feed into the models. We will now briefly explain them.

- **Polynomial Features:** Generate a new feature matrix consisting of all polynomial combinations of the features with degree less than or equal to an specified degree. For example, if an input sample is two dimensional and of the form [a, b], the degree-2 polynomial features are [1, a, b, a^2, ab, b^2].
- **Standard Scaler:** Standardize features by subtracting the mean and scaling to unit variance, of each variable. Is a common preprocessing method in many algorithms, but it might behave badly if the variables do not (more or less) look like standard normally distributed data.
- **MinMax Scaler:** Transforms features by scaling each variable to a given range. The most typical ranges are between 0 and 1, or between -1 and 1.

4.2. Implemented Models

In this section we will briefly explain the main families of the regression and classification algorithms implemented in this project. The regression analysis aims to estimate the relationship between a dependent variable (target variable 'AHI') and multiple independent variables (features). The classification analysis aims to identify to which of the possible categories the observations belong, based on the input variables (features).

- **Generalized Linear Models:** Set of algorithms that is characterized by the intention of estimating the variable target as a linear combination of features. Models that implement reguralizations to coefficients are also included. To this family belong algorithms like: Linear Regression, Ridge, Lasso, Elastic-Net, Logistic Regression, LARS, Orthogonal Matching Pursuit (OMP), Bayesian Regression, Stochastic Gradient Descent (SGD)... The simplest mathematical notation of this family can be expressed as:

$$\hat{y}(w, x) = w_0 + w_1 x_1 + \ldots + w_p x_p$$

- **Support Vector Machines:** A set of supervised learning algorithms based on the construction of hyperplanes in multi-dimensional spaces. They offer great advantages

such as high performance with high number of dimensions, efficient in terms of memory and versatile for many different types of problems and data.

- **Nearest Neighbors:** This set of algorithms has applications in both supervised and unsupervised learning. They are based on the search for samples that are closer in distance to the new point studied. The number of samples sought can be a constant (K-NN) or vary depending on the local density of points (Radius-based Neighbor). Despite its simplicity, nearest neighbors has been successful in a large number of classification and regression problems.
- **Gaussian Processes:** Set of nonparametric supervised learning algorithms based on stochastic processes, designed for regression and probabilistic classification problems. Since the prediction is probabilistic (Gaussian) we can empirically compute confidence intervals. The major disadvantage is that they lose efficiency in high-dimensional spaces.
- **Decision Trees:** Set of non-parametric supervised learning methods used in both classification and regression problems. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from data features. They require little data preparation, are simple to interpret and can be visualized.
- **Ensemble Methods:** The goal of ensemble methods is to combine the prediction of several bse estimators in order to provide generalizability and robustness over a single estimator. There are two main families: averaging methods (in which predictions are averaged) and boosting methods (in which the estimators are constructed sequentially).
- **XGBoost and CatBoost:** Algorithms based on optimized and distributed gradient boosting on parallel decision trees. Is able to solve a wide variety of problems quickly, efficiently and flexibly.
- **Neural Networks:** Set of methods inspired by the behavior of biological neural networks. They offer great advantages such as the ability to learn high nonlinear models or learn in real time. It is trained using Backpropagation. The *Scikit-learn* library offers the implementation of a Multi-layer Perceptron, a supervised learning algorithm that learns the function $f(\cdot) : R^m \rightarrow R^0$ by training on a dataset.

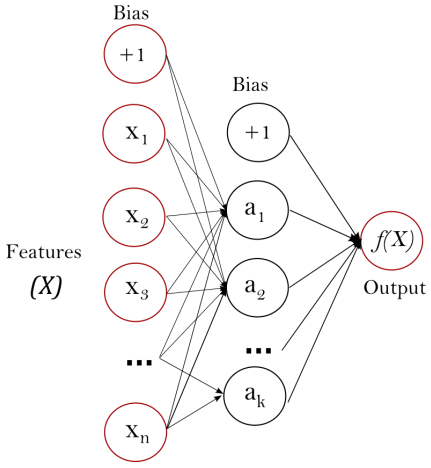

Figure 38. Multi-layer Perceptron diagram with one hidden layer

5. Results

5.1. Evaluation

5.1.1. Regression

Analyzing the results we can notice that there are many models close to the best solutions obtained, there is not a model that stands out from the others. Even so, we can observe that the best results have been obtained in the algorithms ***Kernel Ridge*** and ***Gaussian Process***, applying the **Standard scaling** in the preprocessing of the data. In the following we show visualizations and tables obtained from these outcomes with Standard scaling, the rest of the results and visualizations can be found in this jupyter notebook [8] [9].

Model	R^2	Max Error	MAE ± STD	RMSE
Linear_Regression	0.204	85.64	12.55 ± 16.62	16.62
Ridge_Regression	0.204	85.40	12.55 ± 16.62	16.62
Lasso	0.178	82.49	12.87 ± 16.88	16.88
ElasticNet	-0.010	89.01	14.58 ± 18.72	18.72
LARS	0.056	87.71	14.04 ± 18.10	18.10
Lasso_LARS	0.122	85.91	13.44 ± 17.46	17.46
OMP	0.158	82.17	12.90 ± 17.09	17.09
Bayesian_Ridge	0.198	84.18	12.59 ± 16.69	16.69
Bayesian_ARD	0.198	84.18	12.59 ± 16.69	16.69
Passive_Aggressive	-0.228	85.01	16.40 ± 20.57	20.65
RANSAC	-0.029	101.52	13.42 ± 17.61	18.89
Theil_Sen_Regressor	0.189	89.80	12.41 ± 16.73	16.77
Huber_Regressor	0.177	91.59	12.15 ± 16.68	16.90
Kernel_Ridge	0.213	84.91	12.49 ± 16.53	16.53
SVM_Linear	0.161	92.80	12.10 ± 16.66	17.06
SVM_C-support	0.046	130.42	12.78 ± 17.92	18.19
SVM_Nu-support	0.187	88.55	12.49 ± 16.74	16.79
SGD	0.198	85.19	12.61 ± 16.68	16.68
K-neighbors	0.117	87.54	12.86 ± 17.51	17.51
K-neighbors_Radius	-0.010	89.01	14.58 ± 18.72	18.72

Gaussian_Process	0.211	84.99	12.50 ± 16.54	16.54
PLS_Regressor	0.197	83.98	12.62 ± 16.69	16.69
Decision_Tree	-0.409	91.20	16.39 ± 22.11	22.11
Gradient_Boosting	0.172	67.89	12.75 ± 16.96	16.96
Bagging_Regressor	0.049	84.61	13.65 ± 18.11	18.16
Random_Forest	0.170	84.37	12.73 ± 16.94	16.97
Extra_Tree	-0.518	83.10	17.14 ± 22.95	22.95
AdaBoost	-0.006	79.88	15.58 ± 17.06	18.69
MLP	0.198	84.15	12.49 ± 16.68	16.68
XGBoost	0.158	88.50	12.71 ± 17.09	17.09
CatBoost	0.123	82.05	12.93 ± 17.44	17.44

Table 4. Regression results with Standard scaling

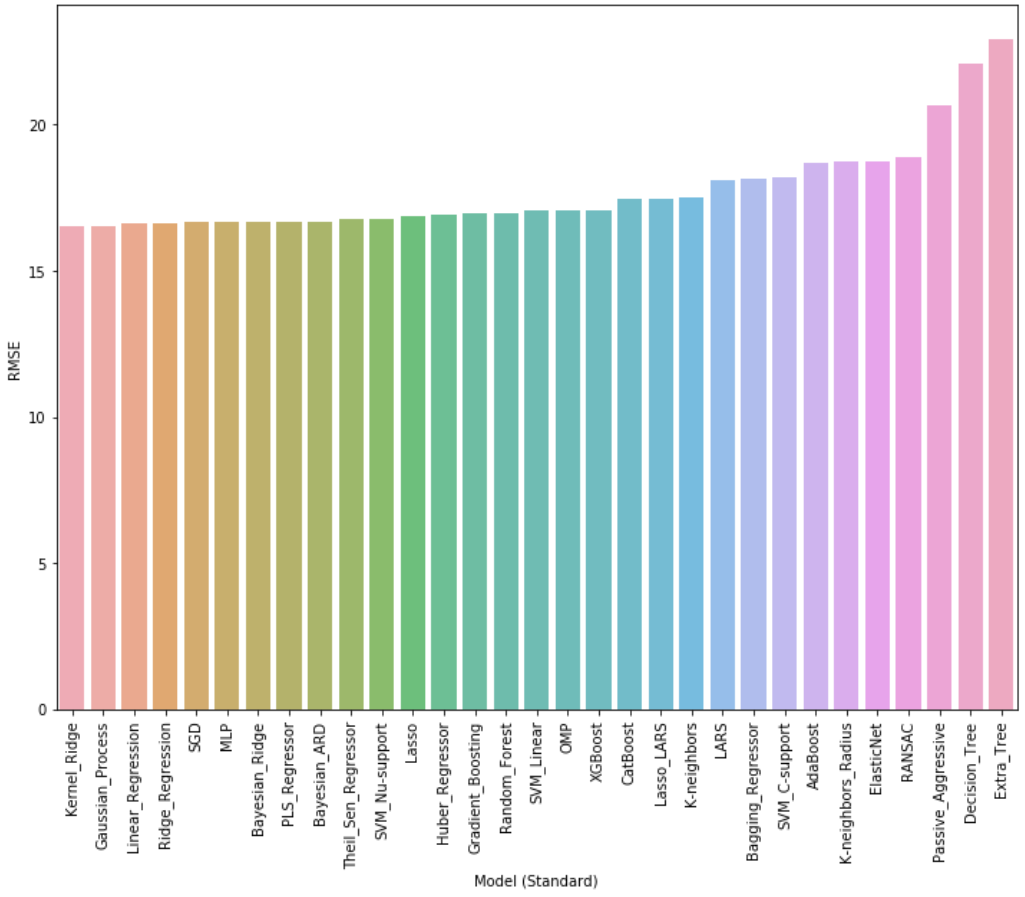

Figure 39. Regression results with Standard scaling comparing RMSE

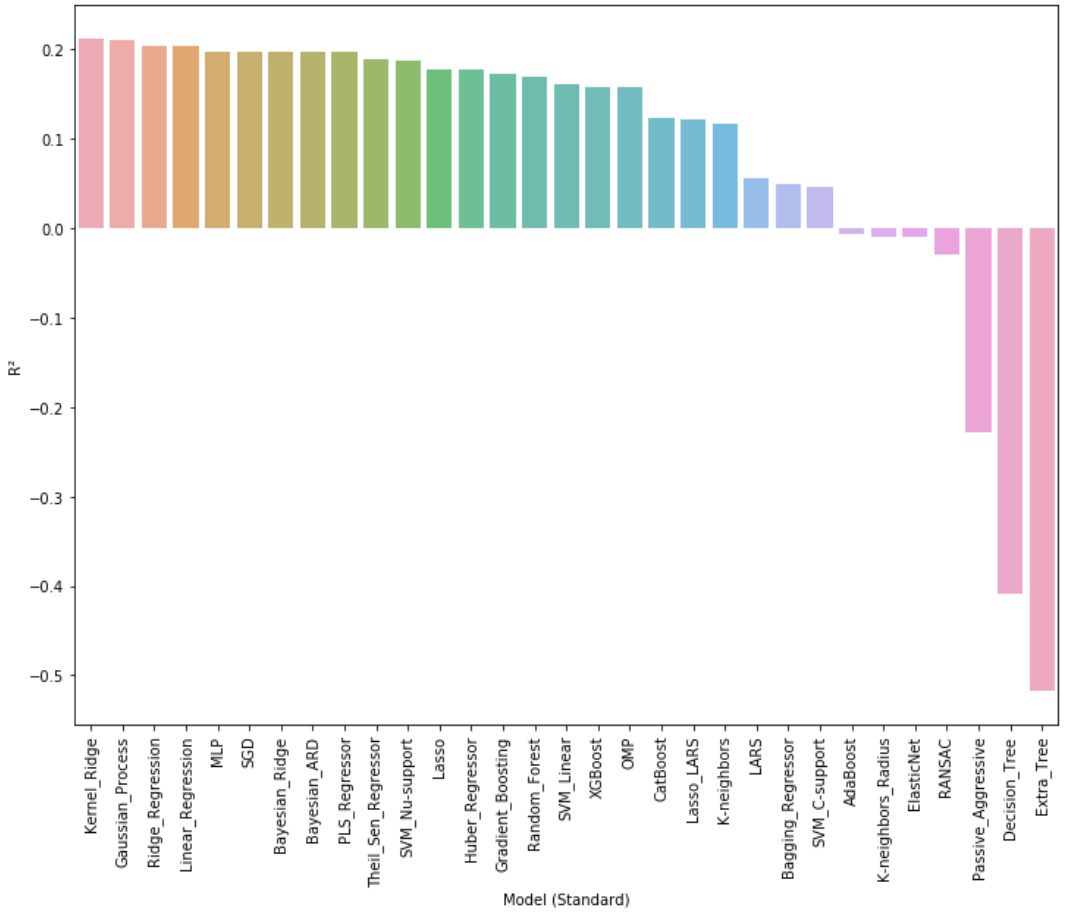

Figure 40. Regression results with Standard scaling comparing R^2

5.1.2. Classification

Analyzing the results we can notice that there are a few models close to the best solutions obtained, there is not a model that stands out from the others. Even so, we can observe that the best results have been obtained in the algorithms ***CatBoost***, ***Random Forest*** and ***K-Neighbors***, without applying any scaling in the preprocessing of the data (raw data). In the following we show visualizations and tables obtained from these outcomes, the rest of the results and visualizations can be found in this jupyter notebook [8] [9].

Model	Precision	Recall	F1-Score	Balanced Accuracy
Logistic_Regression	0.68	0.68	0.676	0.676
Ridge_Classifier	0.68	0.68	0.676	0.676
SGD_Classifier	0.62	0.62	0.617	0.619
Perceptron	0.65	0.62	0.594	0.614
Passive_Aggressive	0.54	0.54	0.534	0.536
NaiveBayes_Bernoulli	0.43	0.43	0.424	0.427
NaiveBayes_Multinomial	0.66	0.66	0.661	0.661

SVM_Linear	0.67	0.67	0.663	0.664
SVM_C-support	0.70	0.69	0.692	0.693
SVM_Nu-support	0.66	0.65	0.654	0.656
K-neighbors	0.72	0.72	0.720	0.720
K-neighbors_Radius	0.68	0.68	0.676	0.676
Neighbor_Nearest-Centroid	0.68	0.68	0.674	0.675
Gaussian_Process	0.68	0.68	0.676	0.676
AdaBoost	0.64	0.64	0.636	0.637
Bagging_Classifier	0.67	0.67	0.668	0.668
Ensemble_Extra_Trees	0.68	0.68	0.679	0.679
Gradient_Boosting	0.70	0.70	0.698	0.698
Random_Forest	0.72	0.72	0.720	0.720
Decision_Tree	0.67	0.67	0.669	0.669
Extra_Tree	0.62	0.62	0.622	0.622
MLP	0.66	0.67	0.507	0.578
XGBoost	0.68	0.68	0.680	0.680
CatBoost	0.72	0.72	0.724	0.724

Table 5. Classification results without scaling (raw data)

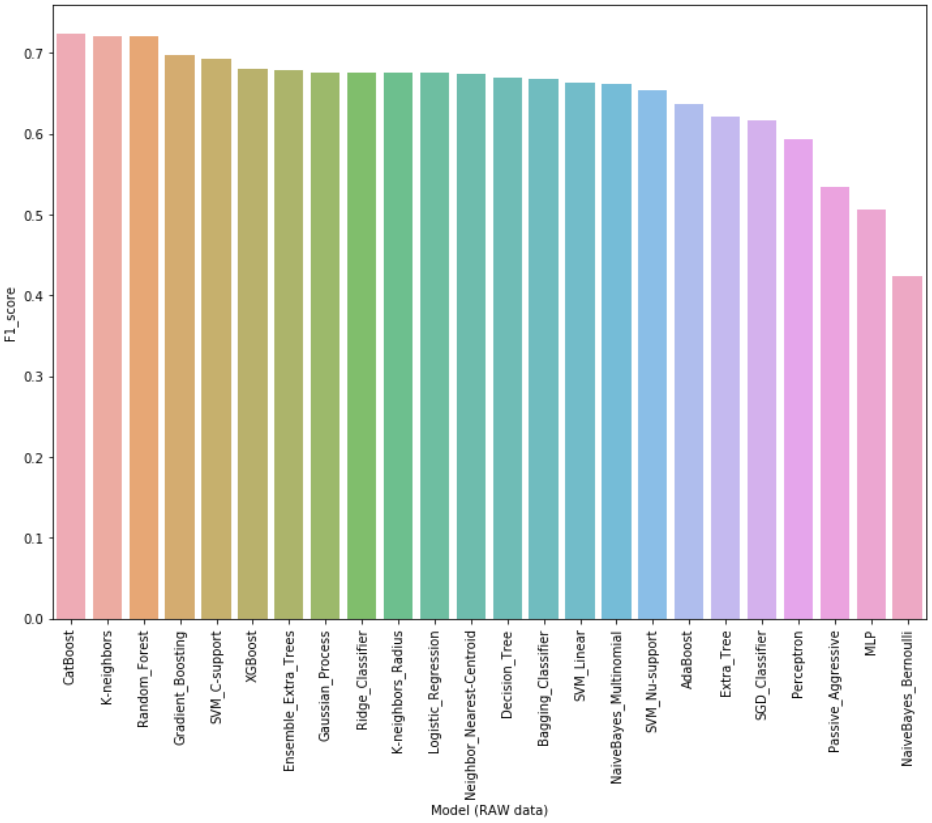

Figure 41. Classification results without scaling comparing F1-Score

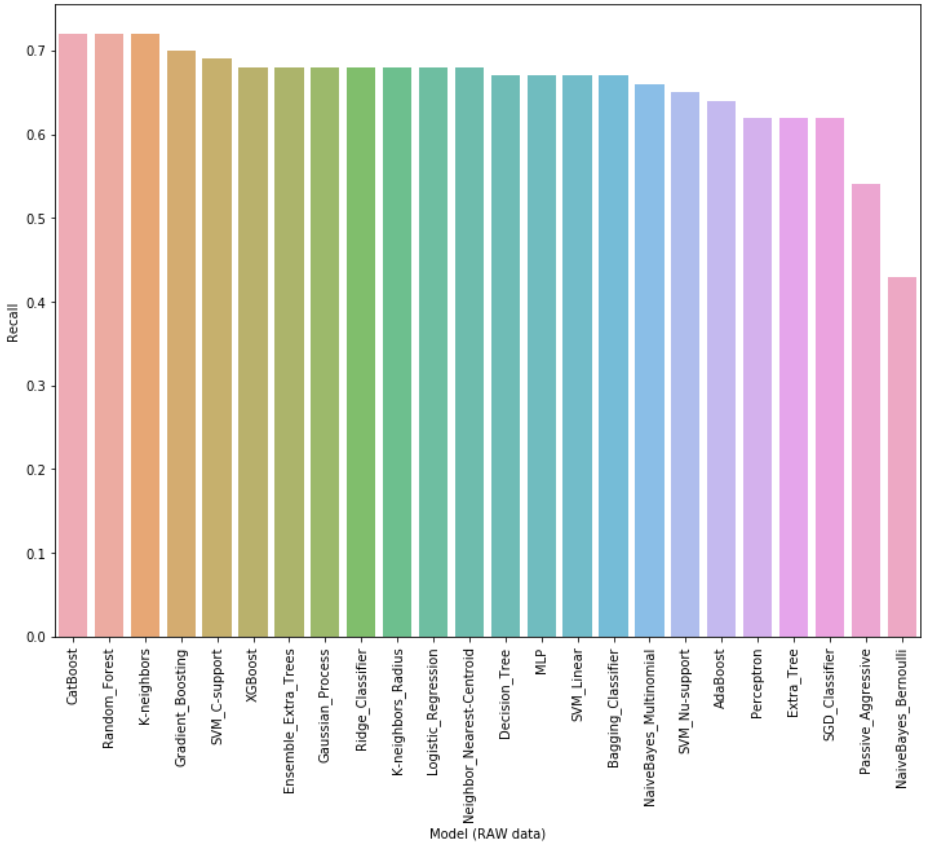

Figure 42. Classification results without scaling comparing Recall

5.2. Discussion

From the analysis carried out in the previous section, in the **regression** problem, the methods shown in Table 6 have been selected as the ones that provide the best results.

Model	R^2	Max Error	MAE ± STD	RMSE
Kernel_Ridge	0.213	84.91	12.49 ± 16.53	16.53
Gaussian_Process	0.211	84.99	12.50 ± 16.54	16.54

Table 6. Selected models for regression problem

The results obtained are quite far from what was expected. With the data obtained and the methods used, no regression model has been found that predicts with sufficient precision to be useful in a real-world deployment. In Figure 43 we can visualize the comparison between the real values and those predicted by the *Kernel Ridge* model. In the same way for the Gaussian Process model in Figure 44.

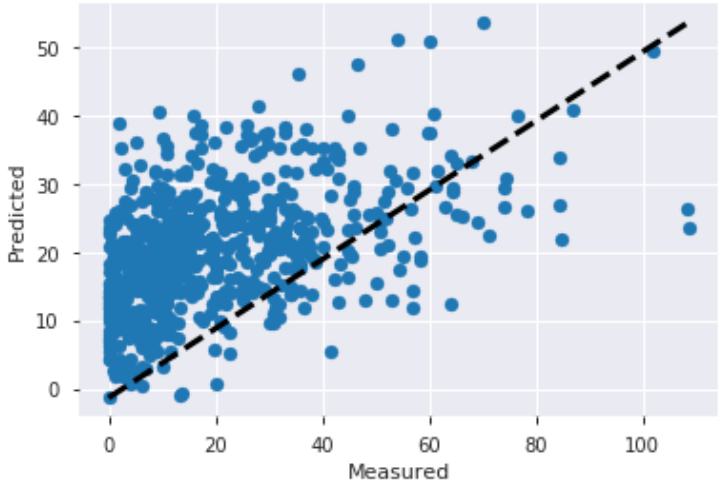

Figure 43. Comparison of actual values with predictions from the Kernel Ridge model

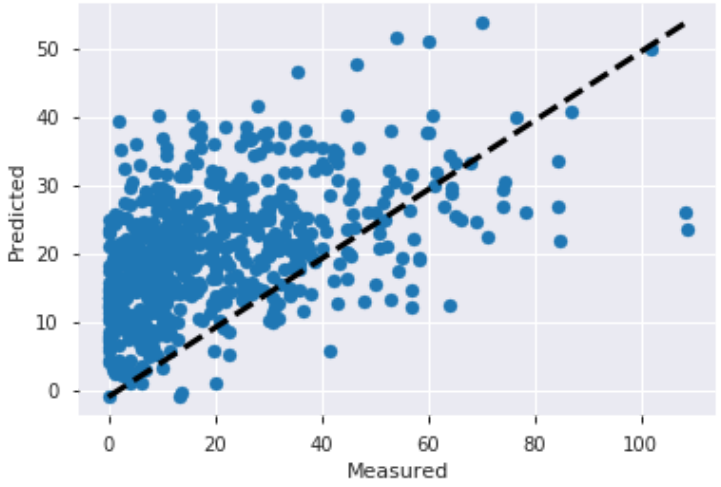

Figure 44. Comparison of actual values with predictions from the Gaussian Process model

In these figures we can note that, despite a certain correlation between real values and predictions, the variance between them is very high and in some cases the models make very remarkable mistakes. If all the predictions were correct, the points of the figures should follow the discontinuous line. The positive aspect is that the selected models belong to the *White Box* algorithm group. That is, their results can be explained from the input variables, providing information on why they have obtained these outcomes.

In addition, the fact that much more complex models are not the ones leading the problem resolution, may indicate that the weak results are not caused by a bad choice of models or hyperparameters, but that the problem is presumably in the data itself. Either because of lack of samples or because the complexity of the problem is too high to be obtained with the used variables.

From the analysis carried out in the previous section, in the **classification** problem, the methods shown in Table 7 have been selected as those that provide the best results.

Model	Precision	Recall	F1-Score	Balanced Accuracy
K-neighbors	0.72	0.72	0.720	0.720
Random_Forest	0.72	0.72	0.720	0.720
CatBoost	0.72	0.72	0.724	0.724

Table 7. Selected models for classification problem

In this case the algorithms have certainly achieved good results. Probably due to the problem composition itself, it is easier to obtain good performance in binary classification than in a regression approach. The following figures show the confusion matrices and ROC curves of the selected models.

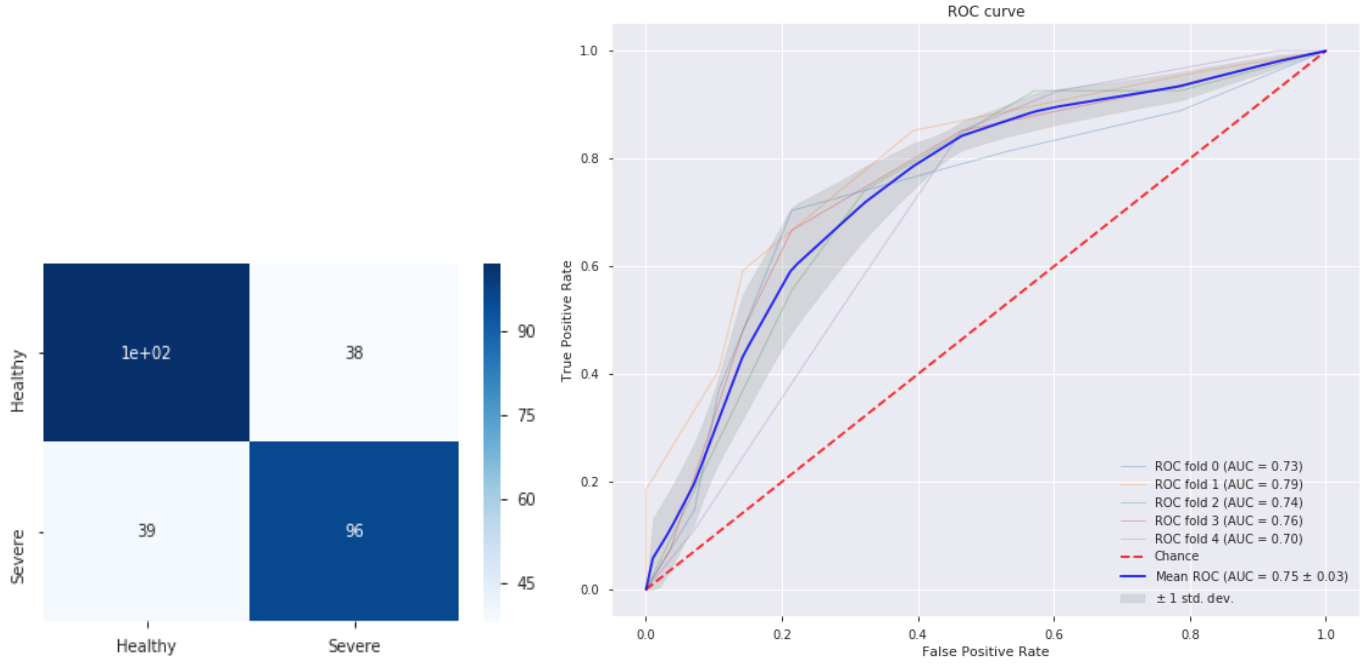

Figure 45. Confusion matrix and ROC curve from K-Neighbors model

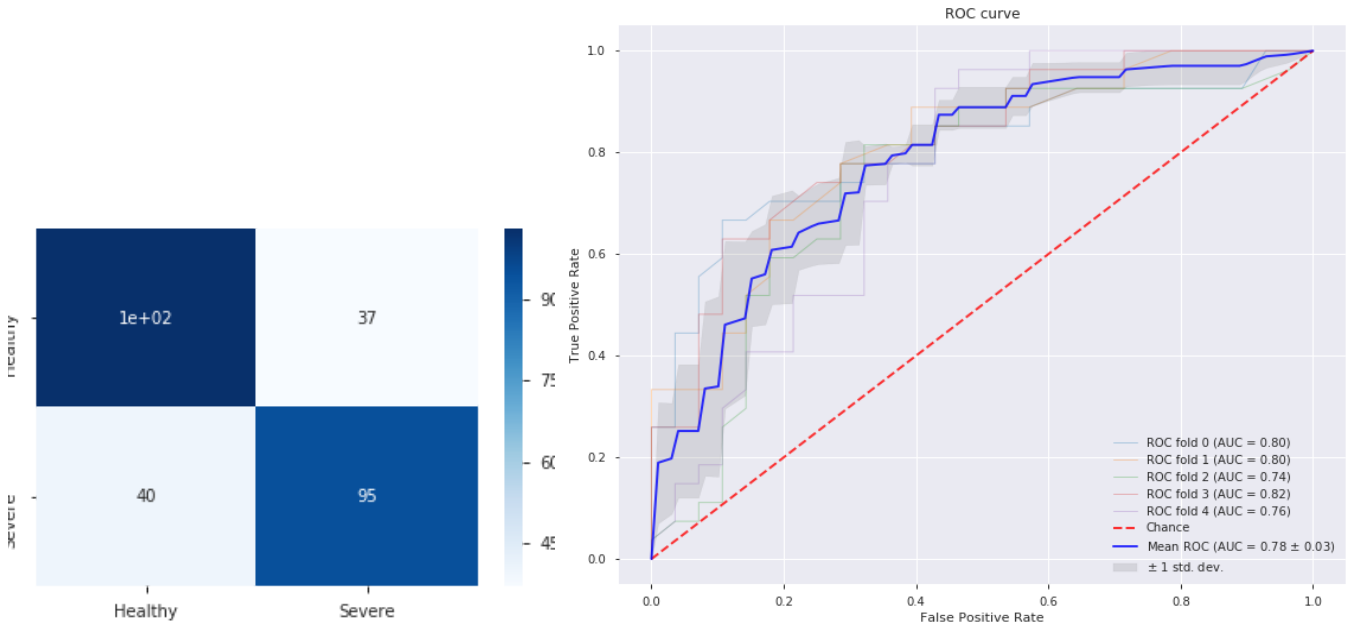

Figure 46. Confusion matrix and ROC curve from Random Forest model

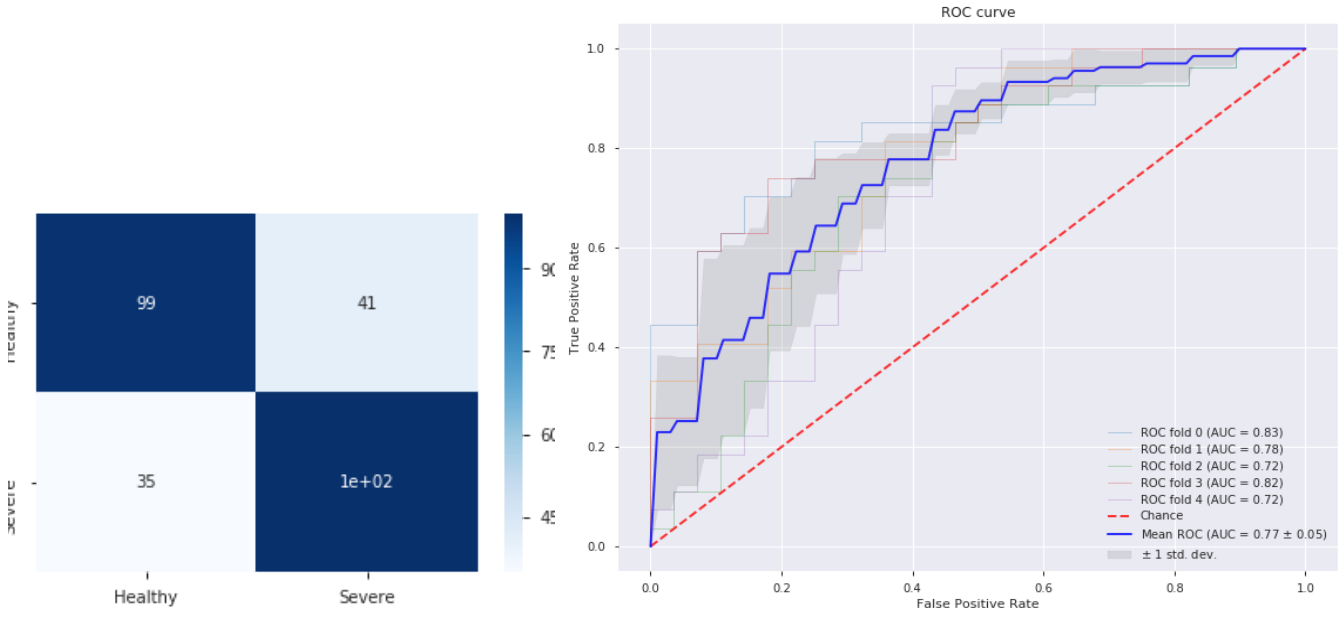

Figure 45. Confusion matrix and ROC curve from CatBoost model

In the figures we can notice that the models obtain relatively good results. In problems related to the detection of diseases, the minimization of the False Negatives (FN) from the confusion matrix is usually considered of much greater importance. That is to say, to minimize the number of patients with severe disease that have been classified as healthy. Therefore, according to this criterion, we should choose the *Random Forest* model.

However, the explainability of the results is also often very important in this type of problem. Two of the selected models belong to the Black Box group (*CatBoost* and *Random Forest*) and the other one to the White Box group (*K-Neighbors*). Therefore, according to this criterion we should choose the *K-Neighbors* model.

6. Conclusions

In this project a comprehensive methodology has been developed to predict Obstructive Sleep Apnea (OSA) using Machine Learning models. This methodology, widely used and studied, is based on *state of the art* Machine Learning model development guidelines. It is described in detail in the first section of the work that complements this, from the *Machine Learning Lab* subject.

My experience in this field due to my work as a researcher at the Polytechnic University of Madrid (Department of Electronic Engineering), has made that the development of this project has been carried out without outstanding issues. Despite the theoretical knowledge, I had never used before some of the techniques applied in this project. For this reason it has been useful for me to put my knowledge into practice and acquire experience in this field. All the code of the project is available in my Github account [4] [5] and in Google Collab [6] [7].

Regarding the obtained results, the models that achieved the best scores in the problems of regression and classification were shown during the report. In the regression it has not been possible to obtain a model good enough to be able to be deployed in a real world use case. On the other hand, the classification model, as it is a simpler task to solve for the algorithms, has achieved results that could be considered good enough. Despite this, it has been concluded that these bad results are not due to the bad choice of models or hyperparameters, but are derived from the dataset used. In order to solve this, it is proposed to obtain more samples, and even the possibility of acquiring more relevant variables that can make the models useful in real world scenarios. All the results and visualizations can be found in this jupyter notebook [8] [9].

The analysis of the requirements (computational and time) of the models used is proposed as a future line of work. Since, for its deployment in the real world this are very important factors to take into account when deciding which models will be used.

7. References

[1] «Obstructive Sleep Apnea», [Online]. Available: https://www.mayoclinic.org/diseases-conditions/obstructive-sleep-apnea/symptoms-causes/syc-20352090 [Accessed 10 November 2019]

[2] «Obstructive Sleep Apnea Explained», [Online]. Available: https://www.webmd.com/sleep-disorders/guide/understanding-obstructive-sleep-apnea-syndrome [Accessed 10 November 2019]

[3] «Scikit-Learn Library, [Online]. Available: https://scikit-learn.org/stable/ [Accessed 10 November 2019]

[4] Jupyter Notebook with regression approach code, [Online]. Available: https://github.com/jaimeperezsanchez/Machine_Learning_Lab/blob/master/CaseStudy_OSA/CODE_Python/Notebooks/1_Regression_OSA_Extra.ipynb [Accessed 10 November 2019]

[5] Jupyter Notebook with classification approach code, [Online]. Available: https://github.com/jaimeperezsanchez/Machine_Learning_Lab/blob/master/CaseStudy_OSA/CODE_Python/Notebooks/2_Classification_OSA.ipynb [Accessed 10 November 2019]

[6] Jupyter Notebook with regression approach code on Google Collab, [Online]. Available: https://drive.google.com/file/d/1_BaVajc6zaqngJLSN_vzwwn9lS5qS2NP/view?usp=sharing [Accessed 10 November 2019]

[7] Jupyter Notebook with classification approach code on Google Collab, [Online]. Available: https://drive.google.com/file/d/1LdWIC2MSEiHzccP1Tq0qRCYCJ8PeTt0P/view?usp=sharing [Accessed 10 November 2019]

[8] Jupyter Notebook with model results and analysis, [Online]. Available: https://github.com/jaimeperezsanchez/Machine_Learning_Lab/blob/master/CaseStudy_OSA/CODE_Python/Notebooks/Results.ipyn [Accessed 10 November 2019]

[9] Jupyter Notebook with model results and analysis on Google Collab, [Online]. Available: https://drive.google.com/file/d/1UvLXJFKYAQ1LJ038HtDrU7gbajTcDI3Z/view?usp=sharing [Accessed 10 November 2019]