# Data Science Report - Group 03

FILIPE SOUSA 90714 JOÃO GUERREIRO 90734 MIGUEL ROQUE 90758

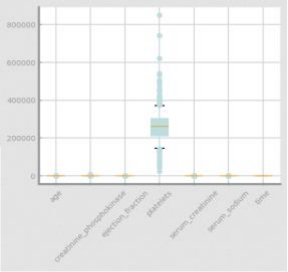
## 1 DATA PROFILING

**Dataset 1** has 299 records and 13 variables (there's a ratio of **23** records/variable), DEATH_EVENT being the **target**. Python assumes all features are represented by an int64 or float64. However, by domain expertise, we know that anaemia, diabetes, high_blood_pressure, sex, smoking and DEATH_EVENT can be better represented as a boolean since they only assume 2 different values.Concerning Data Distribution, we concluded that the variables had different ranges, especially platelets, which makes scaling particularly important. We were able to identify several outliers in our variables except for the time variable.

Regarding granularity, we studied the atomic granularity of each variable and also analysed what was the best granularity for each numeric variable that still made it possible to visualize its distribution. We also concluded that there isn't any hierarchy of concepts for the symbolic variables, either from data or domain knowledge.

To finish this data profiling, we analyzed the data sparsity and made a correlation analysis. In this dataset the variables have very little correlation (most correlation values are between 0.01 and 0.3, 0.53 being the highest). The only two pairs with maybe some relevance are {sex, smoking} and {time, DEATH_EVENT}. We removed the outliers, applied the same techniques and got slightly different results, but still very low correlation coefficients. After this analysis, we concluded that the dataset seems enough to cover the domain. If we consider only the domain ([max, min]) of the data without the outliers, the data is, for almost all pairs of variables, largely scattered around the domain, covering most of it. **Dataset 2** has 8991 records and 1025 variables, 1024 of them being **binary molecular fingerprints** (bmf) and the target - the **experimental class** (exp) - which can be **positive** (**very toxic**) or **negative** (**not very toxic**). It has a ratio of approximately **8.772** records per variable.Regarding the data distribution, 41 bmf variables have 1 as their mode, and the remaining 983 have 0. The exp class has "negative" as its mode. We reach to the conclusion that since we're working with Boolean variables, the notion of outliers had no meaning, only of anomalies. All variables follow a Bernoulli distribution, since they are binary. Regarding data granularity, since all variables are binary, they have a granularity of 2, and don't need any discretization. Regarding data sparsity, we consider it is not useful to generate scatterplots for this dataset, since all variables are binary. However, we found out that about 0.183% of the pairs of variables are missing at least one of the possible combinations ((0, 0), (0, 1), (1, 0) or (1, 1)). Also, there are many pairs of highly dependent variables. We can see that the pairs with the 3 highest correlations are (bmf760, bmf415) with 0.981484, (bmf872, bmf686) with 0.975299 and (bmf684, bmf261) with 0.972491.Regarding missing values, both datasets have none.

Figure 1 - Dataset 1 numeric variables distribution



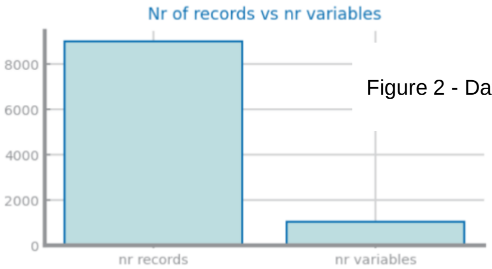


Figure 2 - Dataset 2

# 2 UNSUPERVISED

## 2.1 Association Rules

### *2.1.1 Data preparation.*

In **Dataset 1,** we started by dummifying the ‘sex’ column in ‘sex_M’ and ‘sex_F’ in order to find patterns for women too. We also dummified each numeric variable into bins of equal number of records, with different numbers of bins. We will refer to these configurations as A and B onwards (A: 10 bins for ejection_fraction and 25 for all others, B: 5 bins for ejection_fraction and 10 for all others). In **Dataset 2**, we selected features with correlation less than 0.13, so the algorithm ran in our devices without crashing.

### *2.1.2 Pattern mining*

In Dataset 1, we used a min support of 0.001 and we found 242911 patterns with configuration A and 225851 with B. In Dataset 2, we had to increase min support to 0.004 due to computational resources, and we only found 4602 rules, most probably because the 175 remaining variables (after selection) had low correlation and thus have few implication relationships between them.

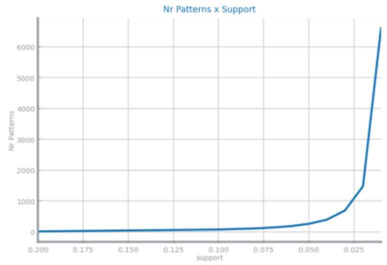


Figure 3 - Dataset 1 Configuration B

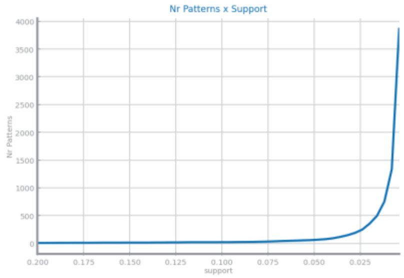


Figure 4 - Dataset 2

### *2.1.3 Association rules*

In Dataset 1, we found 11644889 association rules with configuration A and 9618208 with B. In Dataset 2, we found 4602 rules.

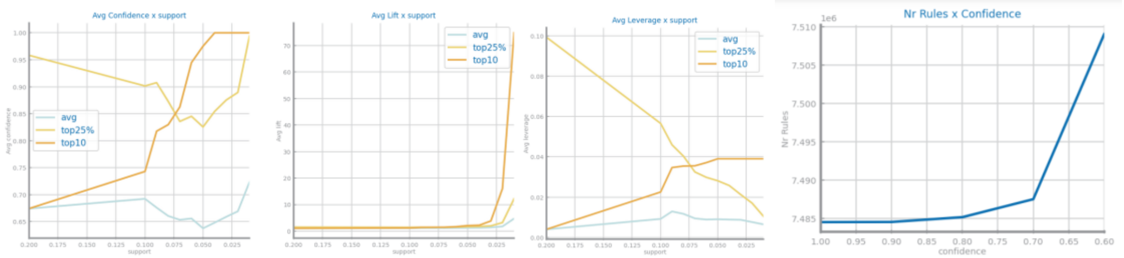


Figure 5 - Dataset 1

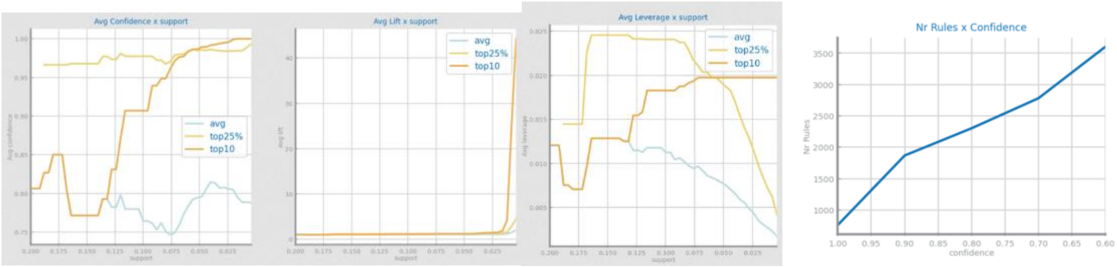
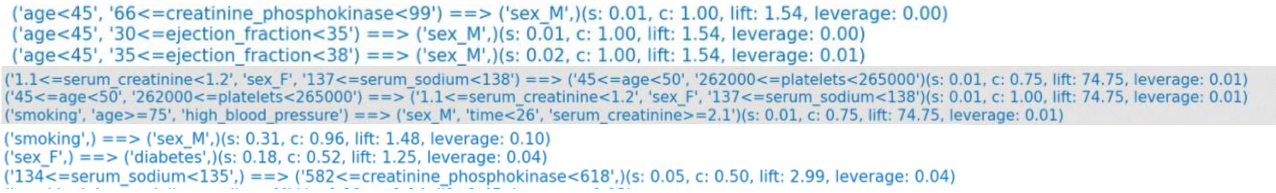


Figure 6 - Dataset 2



```
('age<45', '66<=creatinine_phosphokinase<99') ==> ('sex_M',)(s: 0.01, c: 1.00, lift: 1.54, leverage: 0.00)
('age<45', '30<=ejection_fraction<35') ==> ('sex_M',)(s: 0.01, c: 1.00, lift: 1.54, leverage: 0.00)
('age<45', '35<=ejection_fraction<38') ==> ('sex_M',)(s: 0.02, c: 1.00, lift: 1.54, leverage: 0.01)
('1.1<=serum_creatinine<1.2', 'sex_F', '137<=serum_sodium<138') ==> ('45<=age<50', '262000<=platelets<265000')(s: 0.01, c: 0.75, lift: 74.75, leverage: 0.01)
('45<=age<50', '262000<=platelets<265000') ==> ('1.1<=serum_creatinine<1.2', 'sex_F', '137<=serum_sodium<138')(s: 0.01, c: 1.00, lift: 74.75, leverage: 0.01)
('smoking', 'age>=75', 'high_blood_pressure') ==> ('sex_M', 'time<26', 'serum_creatinine>=2.1')(s: 0.01, c: 0.75, lift: 74.75, leverage: 0.01)
('smoking',) ==> ('sex_M',)(s: 0.31, c: 0.96, lift: 1.48, leverage: 0.10)
('sex_F',) ==> ('diabetes',)(s: 0.18, c: 0.52, lift: 1.25, leverage: 0.04)
('134<=serum_sodium<135',) ==> ('582<=creatinine_phosphokinase<618',)(s: 0.05, c: 0.50, lift: 2.99, leverage: 0.04)
```

Figure 7- TOP 3 rules (confidence, lift and leverage) Dataset 1

We didn’t show the top 3 rules for dataset 2 as they don’t have a very perceptible meaning.

## 2.2 Clustering

### *2.2.1 Data preparation.*

For **Dataset 1** we tested with the raw data, the data after two types of scaling (minmax and zscore), and then added Feature Extraction (PCA) for the three just mentioned. For **Dataset 2** we experimented with Feature Selection and with Feature Extraction. For each modified dataset obtained we applied several Clustering methods (K-means, Expectation Maximization, Density Based and Hierarchical). Feature Extraction theoretically may improve the clustering algorithms by generating features that produce better clusters.

### *2.2.2 K-means*

For **Dataset 1**, without preparation, the variable “platelets” was having a high impact in the clusters’ formation due to its high scale. After scaling and FE, the clusters were more well defined, as we can see below (k=7 was chosen using the elbow method for MSE). For **Dataset 2**, we present only the results after FE, as otherwise the variables are binary.

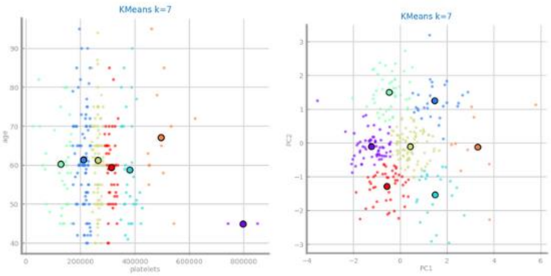


Figure 8 - Dataset 1

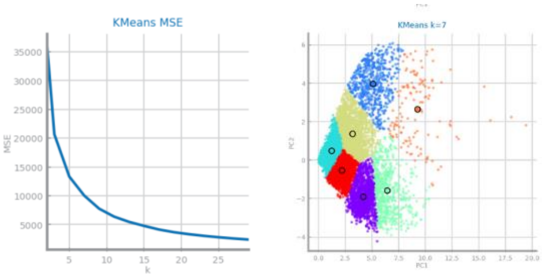
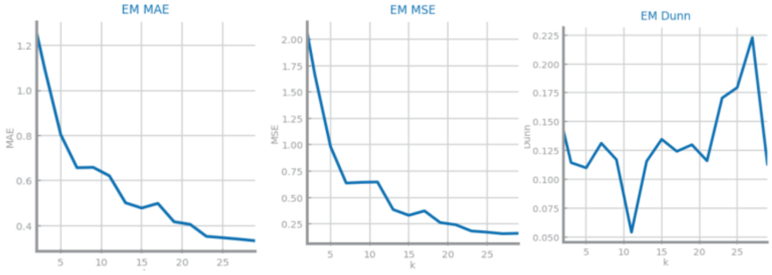


Figure 9 – Dataset 2

### 2.2.3 *Expectation Maximatizion*

For both datasets, the results and plots were similar to K-means. For dataset 1, we compare MAE and MSE and conclude that they follow similar distributions. The chosen k was also 7 for both. For that k, the silhouette is around 0.35, and remains about the same for higher k. This can be due to the clusters being very close together, and thus increasing the number of clusters doesn't increase their separability. We also computed Dunn coefficient, but its distribution was somewhat random and we couldn't reach any conclusion.



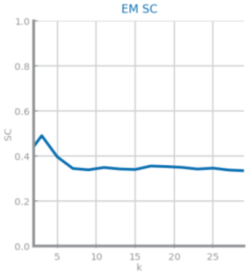
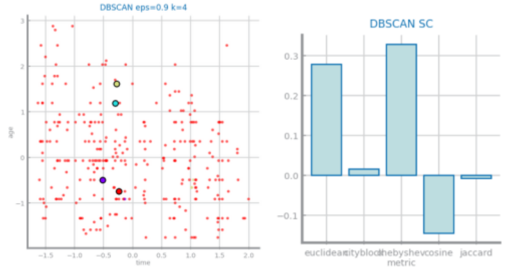


Figure 10 - Dataset 1 Figure 11 - Dataset 2

### 2.2.4 *Density Based*

For **Dataset1**, we calculated the average distance between records, and tried to choose values for epsilon near those distances. However, the results were poor, as can be seen by SC, which is even negative in cosine measure. There is usually a large cluster, and many small clusters that intersect the large one, which explains these poor results. The results were similar for **Dataset 2**, and we couldn't even plot the graph for chebyshev and jaccard measures.



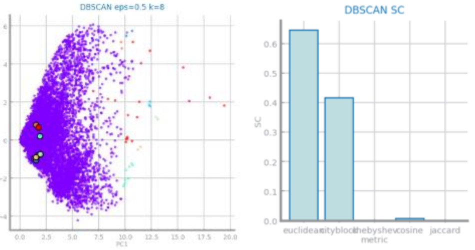


Figure 12 - Dataset 1 Figure 13 Dataset 2

### 2.2.5 *Hierarchical*

The firs image shows the hierarchical clusters for k=7 for Dataset1 with minmax scaling and FE. We can see that these preparation methods yield well-defined clusters, which is confirmed by high values for Davies-Bouldin index. For both datasets, the cosine measure was usually the best. This can be seen in the bar chart for Davies-

Bouldin index (for **Dataset 1**). We can also see that it presents lower values of MSE than most other metrics for both types of link (for **Dataset 2).**

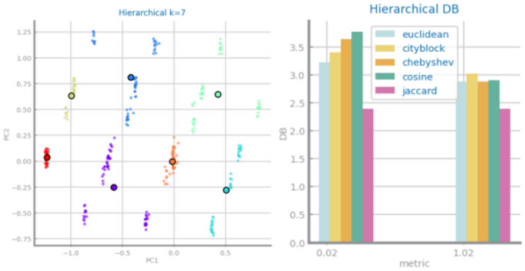


Figure 14 - Dataset 1

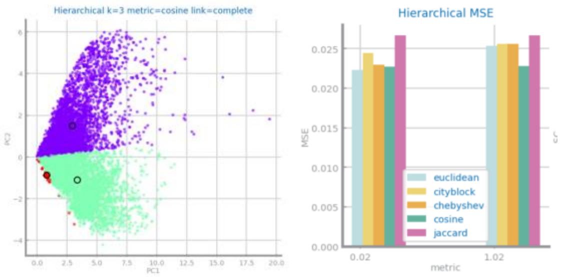
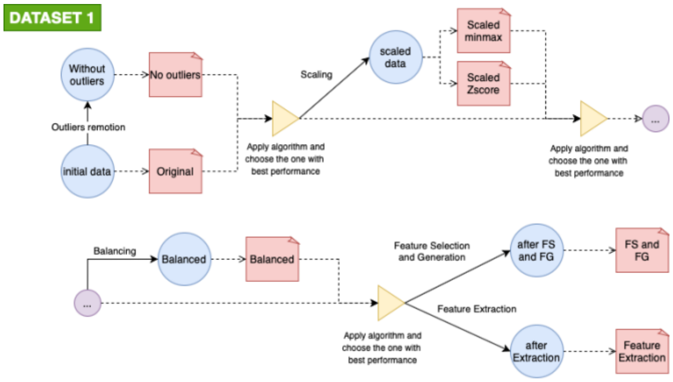
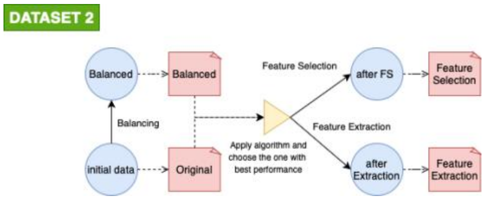


Figure 15 - Dataset 1

## 3 CLASSIFICATION

### 3.1 Data preparation for Classification algorithms





The process of data preparation for the modified datasets used to test each algorithm is represented in general by the schemes below.

We considered records as outliers if the value for one of its variables was outside the interval [(Q1 - 1.5 * IQR), (Q3 + 1.5 * IQR)]. For balancing, on dataset 1, we used SMOTE; on dataset 2 we did undersampling of the

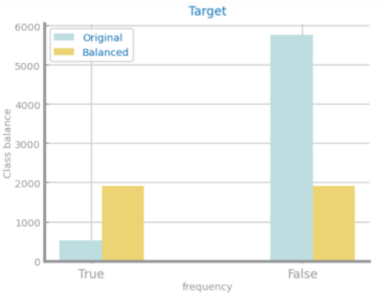


Figure 16 Balancing Dataset 2

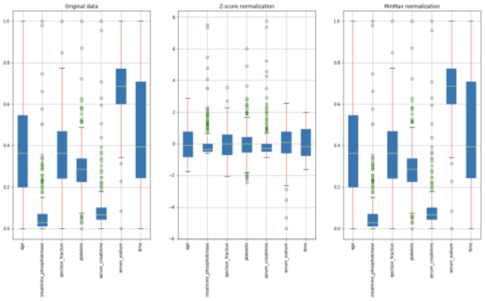


Figure 17 Scalig Dataset 1

majority class (reduced it to 2/3 of the records) and oversampling of the minority class (to match the num. records of the majority class).

For FS, we used supervised methods (we removed features with less than 0.1 correlation to the target and then applied a select K-Best using chi-square with min chi-score 50) and unsupervised methods (we removed features with more than 0.8 correlation with at least one other feature). For FG, on Dataset 1, we did sum, product and difference of all pairs of numeric variables. For the last 3 algorithms, we didn't use outliers' removal since they shouldn't affect trees. In all algorithms, except Decision Trees, we did Feature Extraction, using PCA. In dataset 1, although note referenced in the diagram, FG and PCA were always done after scaling, as the variables had very different ranges.

## 3.2 Training models

Based primarily on the size of the datasets, for Dataset 1 we used Stratified K-fold Cross Validation with 10 splits, and for Dataset 2 Hold-out (70% train / 30% test) without sampling, and for the preparation method with best results, with sampling (5 samples).

## 3.3 Evaluation metrics

For both datasets we used mean accuracy and recall, and 99% and 95% confidence intervals for accuracy in cases with multiple samples. We considered (besides accuracy) recall particularly important, as in both datasets it is important that the model is good at identifying true positives (death – Dataset 1, and toxicity – Dataset 2), as they have a higher impact than negatives, and thus need to be predicted with more certainty.

## 3.4 Algorithms

### *3.4.1 Naïve Bayes*

The z-score scaling, by assuming negative values, doesn't work for MultinomialNB, and so we only considered minmax (mm) scaling. The best results for **Dataset 1** were obtained using the data with no outliers, no scaling, with balancing and feature selection and generation. In the next graph we present the accuracy score and confidence intervals of the several modified datasets we tried and the confusion matrix for the best result. The best estimator was Gaussian.

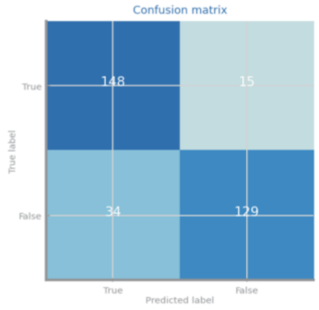


Figure 18 - Best Confusion Matrix

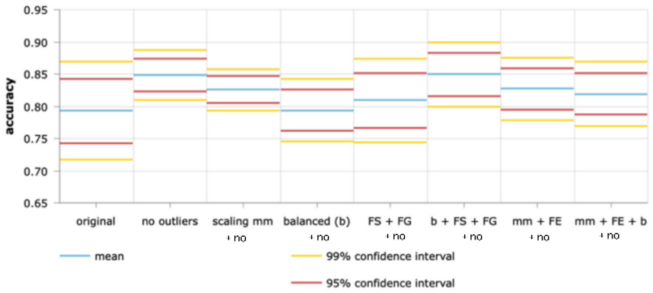


Figure 19 - Mean and confidence intervals

For **Dataset 2** without balancing the accuracy is high but this is probably only because the original dataset has 8 times more False values than True values for the target variable and the classifier is marking almost everything as False as we can see in the first confusion matrix on Figure 3. Therefore, we also considered recall, and decided that the best results for this algorithm and dataset were with balancing only (Figure 4), that achieved accuracy of 0.802 with [0.*786*, 0.804] and [0.782,0.808] as 95% and 99% confidence interval respectively and using Multinominal. This can be explained by dataset2 being composed of multiple binary variables, and thus being well modeled by a Multinomial distribution. A bar chart of the best estimator, accuracy and recall is presented in Figure 5

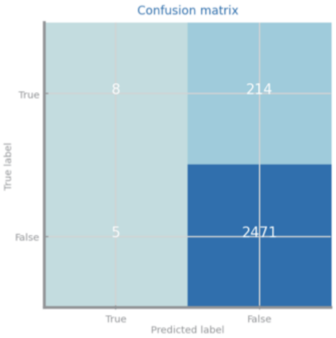


Figure 20 – Unbalanced Confusion Matrix

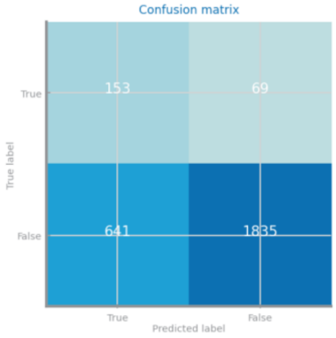


Figure 21 – Balanced Confusion Matrix

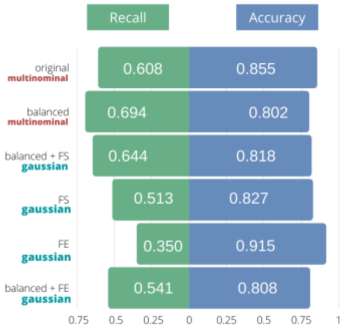


Figure 22 – Evaluation metrics

### *3.4.2 KNN*

For **Dataset 1** the best results were achieved with z-score scaling, balancing and feature selection and generation. Regarding the distance measures, Manhattan and Chebyshev were the ones that showed better results. The best value for neighbours varied mostly between 5 and 9.

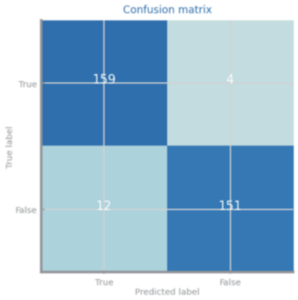


Figure 23- Best confusion matrix

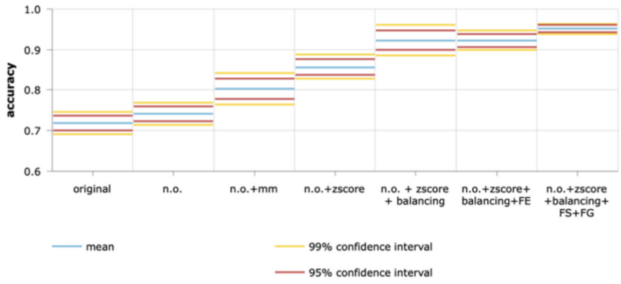


Figure 24-Mean and confidence intervals

After looking at performance indicators, we concluded that, for **Dataset 2,** we got the best results with balancing only. Note that once again, the unbalanced data had higher accuracy but little true positives and therefore the model was useless given the domain of the problem. Manhattan was the measure with best results, 1 was the best number of neighbours. The mean accuracy after balancing only and its corresponding 95% and 99% confidence intervals are 0.894, [0.89,0.897], [0.889,0.899].

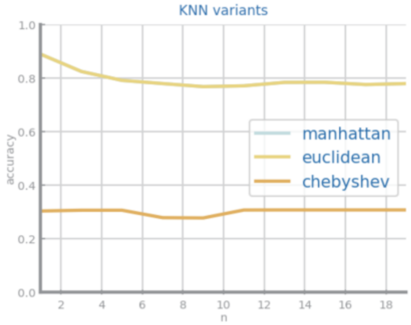
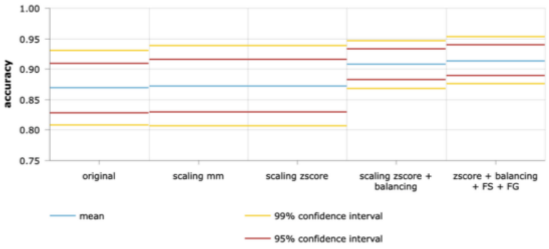


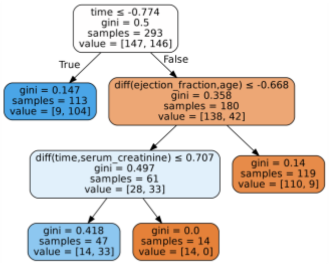
Figure 25 - Dataset 2

### 3.4.3 Decision Trees

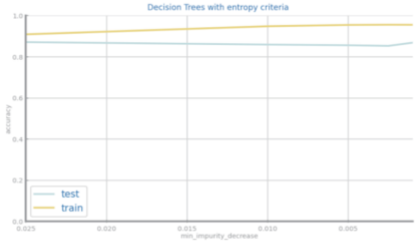
In Decision Trees the scaling had a really small impact in the results. This is likely due to scaling not changing the ordering between the values of each variable (and Decision Trees only have conditions of the form ">=", "=" or "<="). We omitted Feature Extraction, as the resulting trees would not have a useful meaning in the domain of the problems.

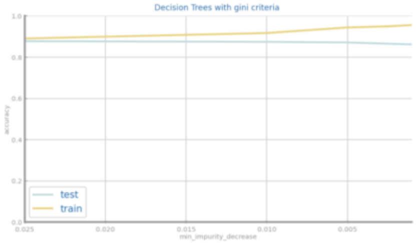
For **Dataset 1** we got the best results after **balancing** and **feature generation and selection**. The accuracies table can be found next as well as one of the best trees for one of the splits (which used gini criteria, max depth = 5 and min impurity decrease 0.02, yielding 0.939 accuracy). The accuracy for the best split was 1, although it doesn't belong in the confidence interval for this preparation method, as seen below.





Regarding Overfitting, we plotted the evolution of train and test accuracy and observed that the divergence of the values was small and therefore the overfitting didn't have a noticeable relevance.





On **Dataset 2**, we opted to only explore the results in the modified datasets after feature selection, since the original data had a high number of variables. Considering this, the best accuracy was achieved without balancing. Though, after an overfitting analysis, the results seemed to be slightly overfit for min impurity decrease less that 0.0005, whereas with balancing the difference in performance between train and test sets is almost constant, ando so there is no overfitting (see Figure 2). The best performance achieved was with feature selection and balancing with gini criteria, the mean accuracy achieved and its corresponding 99% and 95% confidence intervals are 0.899, [0.888,0.911], [0.882,0.917].

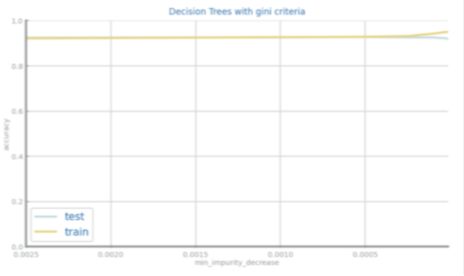


Figure 26- Unbalanced with overfitting

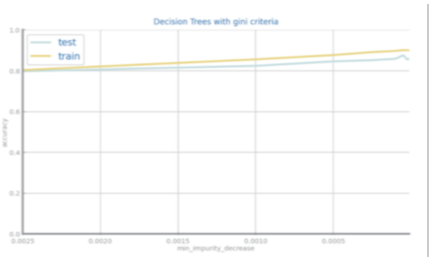
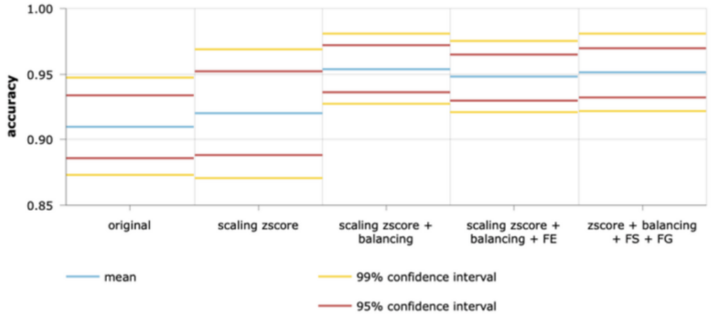
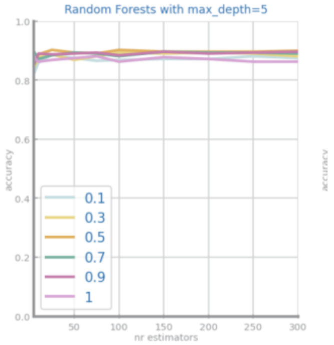


Figure 27 – No overfitting after balancing

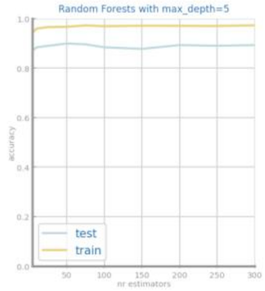
### *3.4.4 Random Forests*

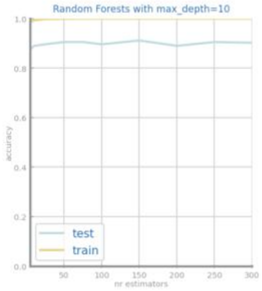
In **Dataset 1** we got the best results with scaling and balancing with and without FG and FS. It's relevant to observe that after FS and FG the algorithm was able to obtain almost the same results but with 3 less features. The most common best max depth was 5, the best number of features varied mostly between 0.1 and 0.7 and the best estimators were most times 10 or 25.

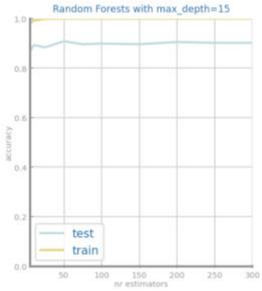


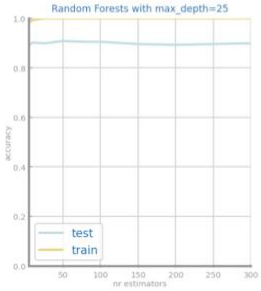


After making an overfitting analysis we concluded that there is likely no overfitting since the train and test accuracies don't diverge.



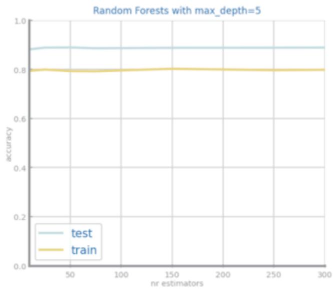


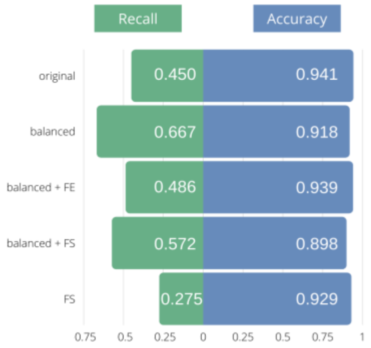




In **Dataset 2** with original we got the best accuracy, however the recall was very low due to unbalanced data, and so we considered the best preparation for this dataset was the one with balancing only.

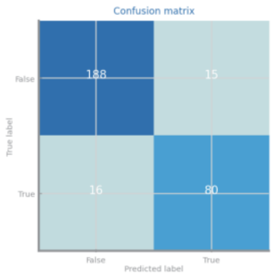
No overfitting was found either as we can see in the graphs below. It is also curious that for max_depth = 5 the best forest classifies better the test than the train set.

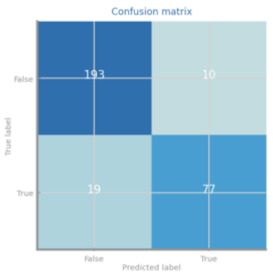


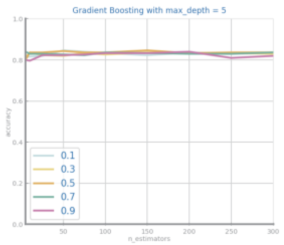


### 3.4.5 Gradient Boosting

For Gradient Boosting we experienced varying the maximum depth, learning rate and number of estimators. The best result for Dataset 1 were obtained with minmax scaling with and without balancing (see matrixes above). The mean, median and mode values for depth were 5.5, 5 and 5 respectively, for learning rate 0.38,

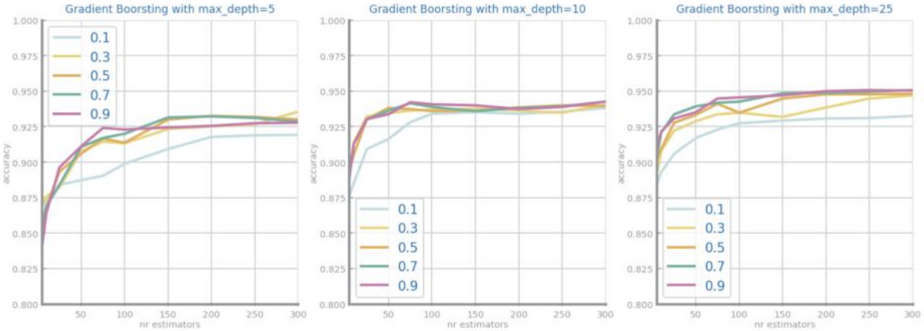






0.30 and 0.1, for n_estimators 91, 50 and 10. For the best of bests results (minmax scaling only) the accuracy and respective 99% and 95% confidence intervals were 0.903, [0.859,0.947] and [0.874,0.932].
Since the accuracy lines don't diverge on the plot presented next, we assumed that overfitting isn't relevant even though test accuracy is lower than train accuracy.

For **Dataset 2** the best result was 0.951 of accuracy, with balancing and feature selection, depth of 25, learning rate of 0.7 and 300 estimators. The evolution of accuracy regarding those parameters can be seen in the next graph.



The other steps of preparation also achieved really high accuracy but only good recall if after balancing.
There seemed to be no overfitting also, after looking to the accuracy evolution in terms of max_depth and nr of estimators as we can see in the next plots.

## 4 IMPROVEMENT STRATEGIES

Although we didn't have time to implement this strategy it came to our mind a classification method based on clusters that would work in the following way: we would start by using the results obtained on clustering to separate our data into clusters; next, for each cluster, we would train a specialized classifier for that cluster, using a classification method like GB or RF. Then, if we were given a new record, we would classify it using the classifier corresponding to the nearest cluster. This might be useful, especially in the case where each cluster doesn't separate well the positive and negative labels for the target.