Data Science - Report (Group 11)

ISABEL SANTOS RAMOS SOARES (89466), Instituto Superior Técnico
JOÃO RIBEIRO DIAS (89484), Instituto Superior Técnico
RODRIGO BORGES PESSOA DE SOUSA (89535), Instituto Superior Técnico

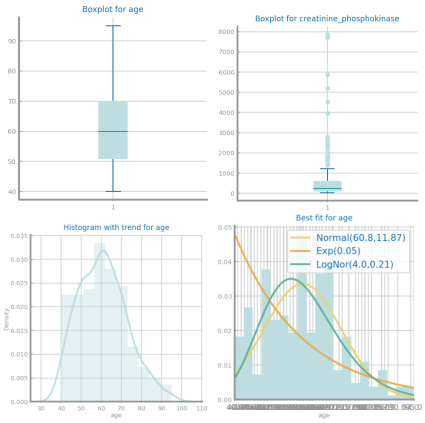
1 INTRODUCTION

For this project we used two datasets to analyze, visualize and manage the information associated to these same datasets, taking advantage of the best of this information and using it to classify all records. We will use **QOT** to represent the "qsar oral toxicity" dataset and **HFCR** to represent the "heart failure clinical records" dataset.

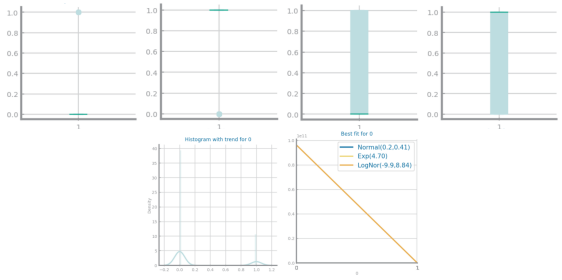
2 DATA PROFILING

2.1 Dimensionality and Distribution

Using a bar chart for each dataset, we noticed that the number of records was much higher than number of variables, which is satisfactory to avoid the 'Curse of Dimensionality' problem. Related to HFCR, the number of records is 300 and the number of variables 13, either numeric (represented by an Integer) or Binary, being 'anaemia', 'diabetes', 'high blood', 'sex' 'smoking' and the target variable 'DEATH EVENT' the binary ones. QOT has 8992 records and 1025 features, being all binary (either 0's and 1's or in the case of the variable 1024 negative and positive). To analyse the data distribution we made use of boxplots, to describe it through its mean, quartiles and outliers, and we noticed that we would need to treat the variables later due to their different scales. In the figure 1a we show the variable age and the variable creatinine phosphokinase, and only the second one with outliers. We also to display a function that would fit the HFCR age variable and we compared the histograms with Normal, Exponential or LogNormal functions. We found that, in HFCR for instance, the age is similar to Normal. As the QOT variables are all binary, there are only four possible distributions as we can see in the first line of histograms of 1. Then, the functions chosen to QOT do not provide information, because the variables are binary. HFCR has also some binary ones.



(a) HFCR

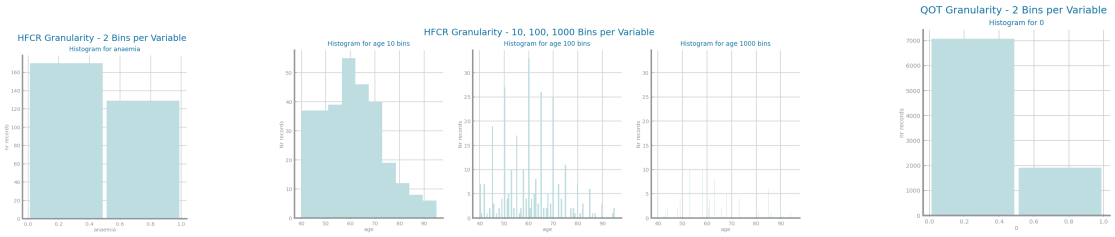


(b) QOT

Fig. 1. Distribution

2.2 Granularity

In order to study the granularity of the data stored in each one of the datasets we graphed histograms for each one of the variables where we varied the number of bins. Binary Values can be represented with 2 bins since all records have one of two values, as such the variables *anaemia*, *diabetes*, *high_blood_pressure*, *sex*, *smoking* from HFCR and all of the variables from QOT can be represented using 2 bins, as shown in figure 2b. When it comes to the rest of the HFCR variables we tried with 10, 100 and 1000 bins to get an idea of what would be the best fit for each one of this variables, as represented in the figure 2a.

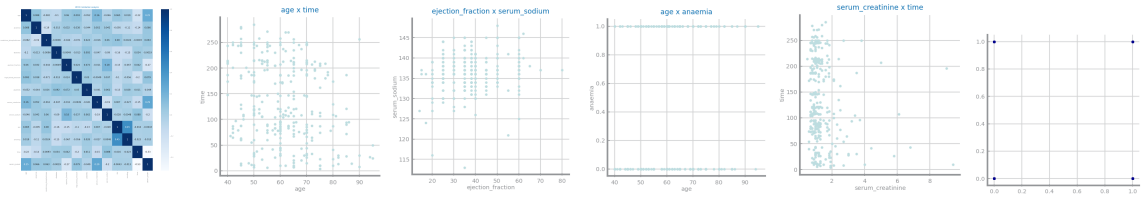


(a) HFCR (b) QOT

Fig. 2. Granularity

2.3 Sparcity and Correlation

In the graphs of 3a we tried to show four of the most common results we obtained. In this, the graph that better describes two correlated variables is the one with the variables ejection fraction and serum sodium (the second graph), that seems to increase linearly. As anaemia is a binary variable, it is easy to understand the third graph, and we get no information about the correlation in this case. The first and the fourth graphs represent also two uncorrelated variables. For QOT, since all the variables are binary, the relation between each others are all equal to the graph 3b, which does not bring any information regarding their correlation.



(a) HFCR (b) QOT

Fig. 3. Correlation matrix and Sparcity

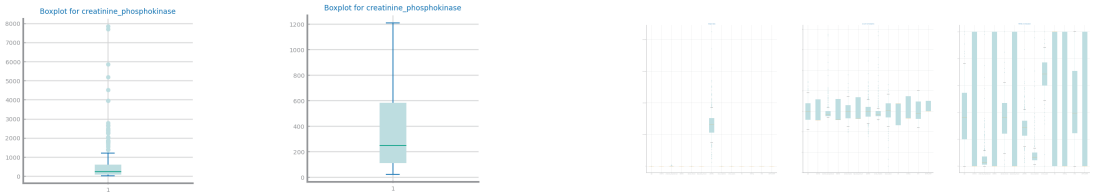
3 DATA PREPARATION

3.1 Missing Values, Outliers and Scaling

We analyzed the number of missing values per variable by plotting them through a bar chart. Then, we observed on both datasets, that there are no missing values.

For the treatment of the outliers, we used Winsorization, that consists on transforming their values to the nearest maximum or minimum. As we can observe in 4b, we have now a better boxplot when compared to the one in 4a for creatinine phosphokinase. In QOT we do not need to remove outliers, since the variables are binary.

When it comes to Scaling it only makes sense to analyse it in the HFCR dataset, since in the QOT all variables are binary. For the HFCR dataset we analysed the scaling through boxplots plotted side by side, showing the original, scaling using Z-score normalization and Min Max normalization, as shown in 4c. In the supervised classifiers we decided to use Z-Score scaling since it is more resilient to outliers and to alterations on the datasets' minimum and maximum. But for clustering we obtained the better results with Min Max normalization.

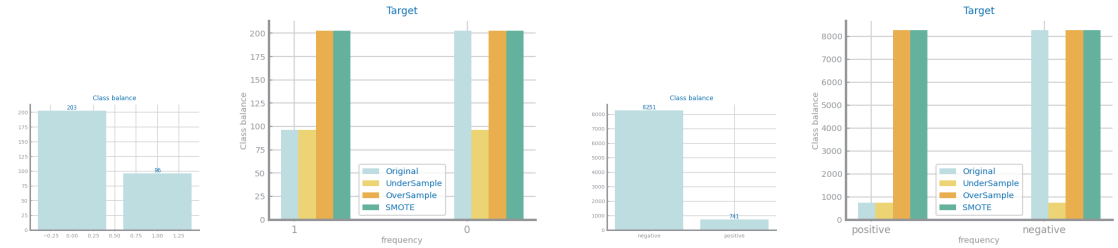


(a) With outliers (b) Without outliers (c) No Scaling, Scaling with Z-Score and Scaling with Min Max

Fig. 4. HFCR - Outliers and Scaling

3.2 Balancing

The analysis of balancing techniques in these datasets is extremely important since, as we can see in figures 5a and 5c, both datasets are unbalanced. We used *Undersample*, *Oversample* and *SMOTE*. In the figure 5b and 5d we display the class balance before and after following each one of this techniques, for both datasets.



(a) HFCR - Original Class Balances (b) HFCR - Class Balances after techniques (c) QOT - Original Class Balances (d) QOT - Class Balances after techniques

Fig. 5. Balancing

3.3 Feature Selection

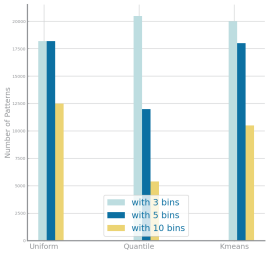
For clustering we applied different variance thresholds in order to explore how the removal of features affected the dataset. When it comes to the supervised techniques, since we can use the target variable, we applied RFECV, which uses a linear estimator to select some of the features of the dataset based on their importance. Of course feature selection is much more important in QOT than HFCR since the first has a large number of features.

4 UNSUPERVISED

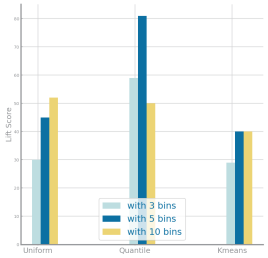
4.1 Association Rules

We executed two different approaches depending on the dataset. Since HFCR is not that large in number of records or features, we were able to run it using a minimum support of 0.02 and explore all the possibilities of length of patterns. The pattern mining for HFCR involved the need for a bin discretizer since some of its attributes are continuous or at least numeric. There are three possible strategies for the bin discretizer: uniform in (each bin has the same 'width'),

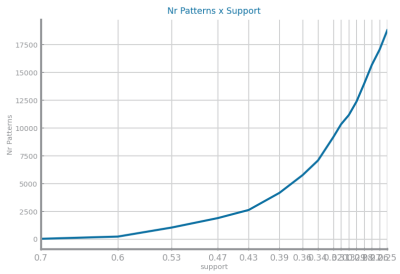
quantile (each bin has the same number of elements) or K means (all the elements in each bin have same nearest center of a 1D k-means cluster).



(a) HFCR Number of Patterns per Strategy and Number



(b) HFCR Lift Score per Strategy and Number



(c) QOT Patterns x Support

Fig. 6. Pattern Mining

We conclude that with techniques uniform and K means the results are similar, the big difference comes with the quantile strategy, where we verify that there are less patterns / rules found as seen in 6a), but they tend to have a greater degree of confidence and lift of the rules achieved 6b), so we decided to follow up with this strategy.

There is still another important factor involved with the bin discretizer, the number of bins. We tested with three different values 3, 5 and 10. We verify again through 6a and 6b that even tough that for 10 there are a lot less patterns / rules discovered the best lift is achieved with a number of bins of 5.

Even tough the different strategies cause a little disruption in the number of patterns achieved and their quality, most of the rules achieved with this parameters reoccur in them, which gives us a greater confidence on the rules achieved, the following rule is the top rule by lift:

Per minimum support: $age[58, 63[, sex, injection\_fraction[14, 30[, serum\_sodium[136, 138[$ ===> $platelets[196000, 237000[smoking, times[147, 210.4[$ (with a lift of 99.67)
Per minimum confidence: $anaemia, serum\_creatinine[0.9, 1[, time[147, 210.4[$ ===> $creatinine\_phosphokinase[110.2, 176.8[, age[40, 50[, serum\_sodium[136, 138[$ (with a lift of 299)

On the other hand QOT doesn't need any sort of bin discretizer since all non-target variables are binary. The problem come from its size both in records and variables, so it wasn't possible for us to run it with a minimum support as low as the one with HFCR. We solved this problem by running QOT first with a min support of 0.57, before we reached the 'explosion' of patterns, and then running with 0.25 but limiting the length of the pattern to a maximum of 3 6c.

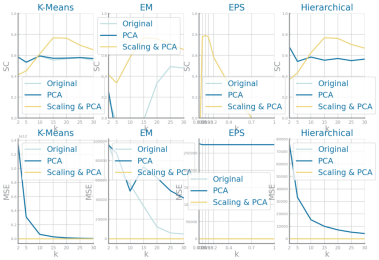
There was no big difference in the number of patterns achieved with the different methods. We expected that the lift score of the rules achieved with the minimum support of 0.25 to be much better, but by analysing the top 10 of the rules achieved we verify that in fact the lift score with 0.25 was not that much better, we suspect that the limit in length affected negatively the confidence of the rules acquired, since with 0.57 the top 10% of the rules all had a bigger length than 3. This is the top rule by lift acquired both per minimum support and minimum confidence:

With a minimum support of 0.25: 626, 15 ===> 125 (with a lift of 3.82)
With a minimum support of 0.57: 222, 467, 960, 437 ===> 288, 16, 473 (with a lift of 1.74)

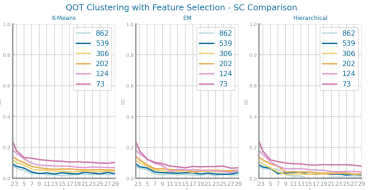
4.2 Clustering

For clustering we chose age and time for HFCR as they the uncorrelated variables that gave us the best results. According to the results obtained in 7a we see greater results in all metrics for scaling with PCA, since the mean squared error (MSE) is normally the smallest and silhouette coefficient (SC) the biggest. We observe a decrease of MSE with the increase of clusters for both datasets. Continuing with HFCR, using the elbow rule, regarding 7d, we choose 10 as the number of clusters in K-Means, EM and Hierarchical, according to the biggest decrease in error. For EPS, Density-based, we choose eps=0.05 with 19 clusters, also untied the two small errors with the greatest SC. We can see in the figure 7h, that there are some similar results, but with the best result for chebyshev with eps=0.05 and k=19 (less MSE and more SC). With Hierarchical Metric, in 7i, the best result was for a complete link, with cityblock and k=3, not too different from average. In the figure 8 we show the visual difference between the smallest number of clusters and the chosen number for K-Means, Expectation Maximization and Hierarchical and the best ones for the other metrics. They were much better than with the original data. The figures 7e and 7f also show the creation of the two new variables by PCA. For the QOT dataset, we show in 7b that feature selection with less variables improves clustering (SC). In 7c, after applying Feature Selection and PCA the results improve significantly for all those metrics. Using the elbow rule, in this case regarding to 7g, we should choose 5 clusters for K-Mean and 10 for EM and Hierarchical. EPS did not give us satisfactory results, but we got eps=0.1 with 131 clusters. The same happened with density-based metric with the best results with chebyshev, eps=0.03 and 1565 clusters. The best hierarchical metric was with an average link, again similar to the complete one, as expected, with chebyshev and k=3.

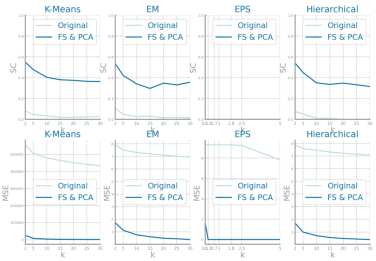
We used maximum absolute error and davies boulding to support our results with MSE and SC (shown in the figures 7h, 7i, 7j and 7k, but used for all metrics). We have a decrease of MSE with the decrease of clusters for both datasets.



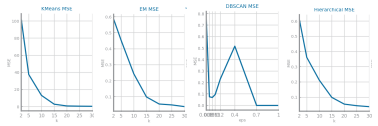
(a) HFCR Original vs PCA vs Scaling with PCA (SC up, MSE down)



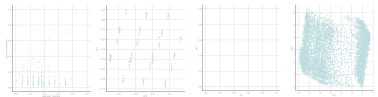
(b) QOT (SC) Feature Selection Comparison



(c) QOT Original vs Feature Selection and PCA (SC up, MSE down)

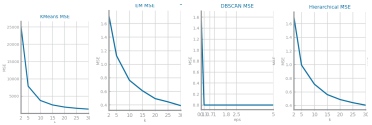


(d) HFCR MSE Comparison

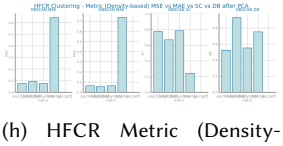


(e) HFCR PCA

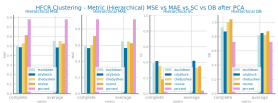
(f) QOT PCA



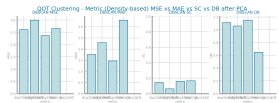
(g) QOT MSE Comparison



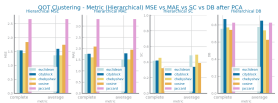
(h) HFCR Metric (Density-based) MSE, MAE, SC and DB



(i) HFCR Metric (Hierarchical) MSE, MAE, SC and DB

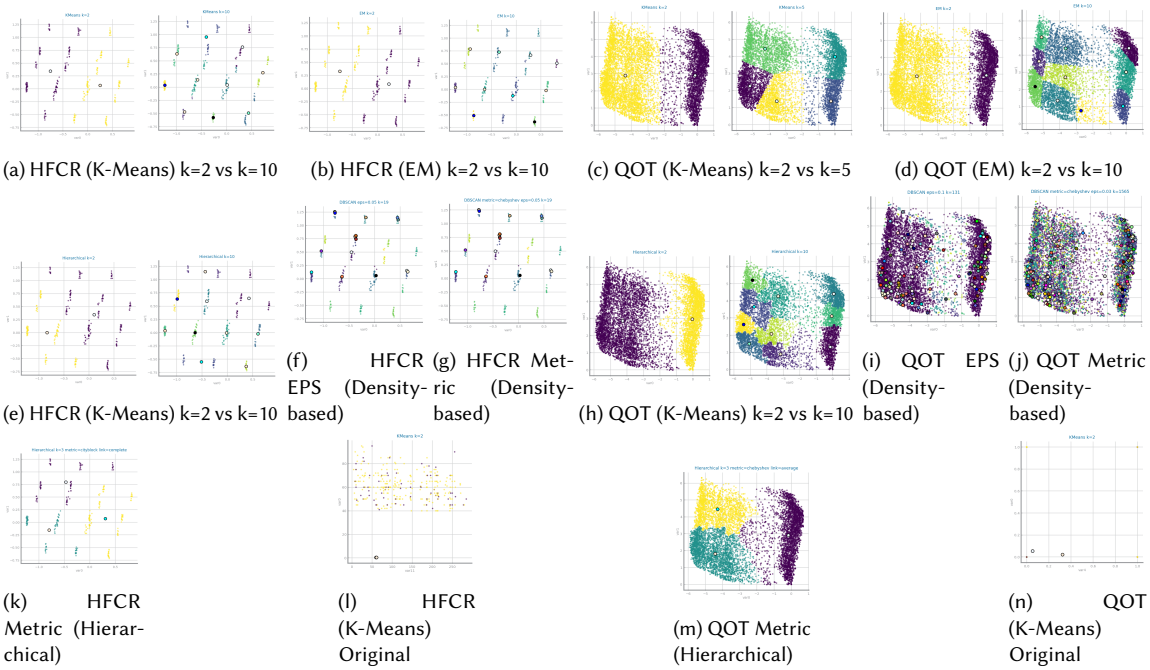


(j) QOT Metric (Density-based) MSE, MAE, SC and DB



(k) QOT Metric (Hierarchical) MSE, MAE, SC and DB

Fig. 7. Clustering - Comparison



(a) HFCR (K-Means) k=2 vs k=10 (b) HFCR (EM) k=2 vs k=10 (c) QOT (K-Means) k=2 vs k=5 (d) QOT (EM) k=2 vs k=10

(e) HFCR (K-Means) k=2 vs k=10 (f) HFCR EPS (Density-based) (g) HFCR Metric (Density-based) (h) QOT (K-Means) k=2 vs k=10 (i) QOT EPS (Density-based) (j) QOT Metric (Density-based)

(k) HFCR Metric (Hierarchical) (l) HFCR (K-Means) Original (m) QOT Metric (Hierarchical) (n) QOT (K-Means) Original

Fig. 8. Clustering - Scatter-Plots

5 CLASSIFICATION

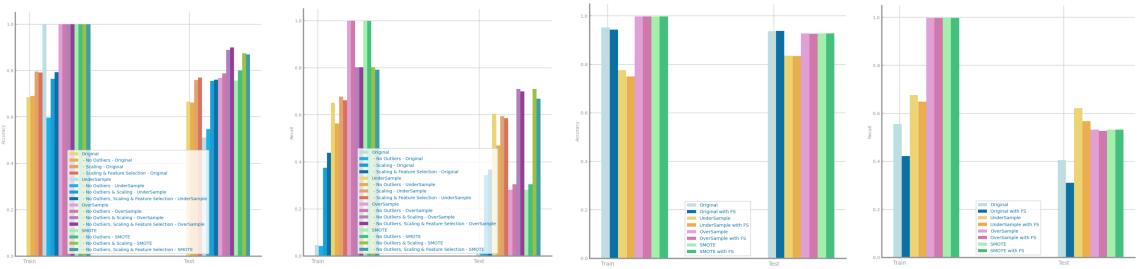
In our Classification experiments we tested the performance of different techniques, such as *outliers removal*, *scaling*, *balancing* and *feature selection*. Feature selection was more important in QOT than HFCR, as the first has much more variables. It is also important to explain that we removed outliers and scaled the data for HFCR, so that the magnitude of the variables would not influence the results. Then used a stratified kfold to split it, in response to the small number of records in this dataset. We used train-test split for QOT, because of the large number of records. We balanced the training data for both datasets accordingly. To increase our confidence in the results obtained we measured the evaluation metrics for each fold in HFCR and calculated the mean. We also made multiple runs with QOT.

5.1 K Nearest Neighbours

As we can see from the figure 9a the outliers removal increases a little the accuracy in HFCR. The major improve is with scaling in HFCR due to the fact that KNN is a distance-based algorithm.
For HFCR the accuracy increases with OverSample and SMOTE and in QOT the accuracies are similar between balancing techniques and original as seen in the figures 9a and 9c. However, the accuracy is not enough and if we look at the recall comparison in figures 9b and 9d, we can see a relevant increase when we balance the data in both datasets, because in the original one the data is unbalanced. Feature selection does not change by much the accuracy, neither in HFCR nor QOT, but we should consider SMOTE with feature selection the best model for both, since it is simpler, calling the Occam's razor.

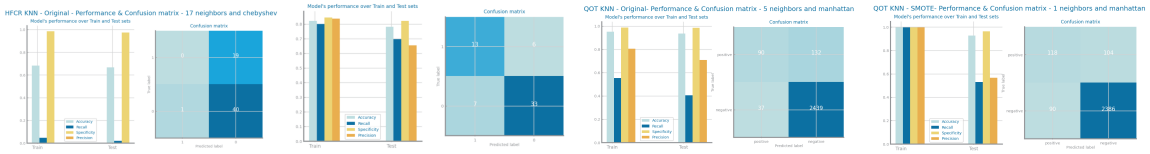
By comparing 10a with 10b and 10c with 10d we understand why the recall is very low for a strategy with outliers, no scaling, no balancing and without feature selection (Original), compared with the treated solution, since almost all

positive records are classified as negatives. The specificity maintains its high values, classifying the negative values accordingly. The confusion matrices helped us confirming these results.



(a) HFCR Accuracy (b) HFCR Recall (c) QOT Accuracy (d) QOT Recall

Fig. 9. Accuracy and Recall for KNN

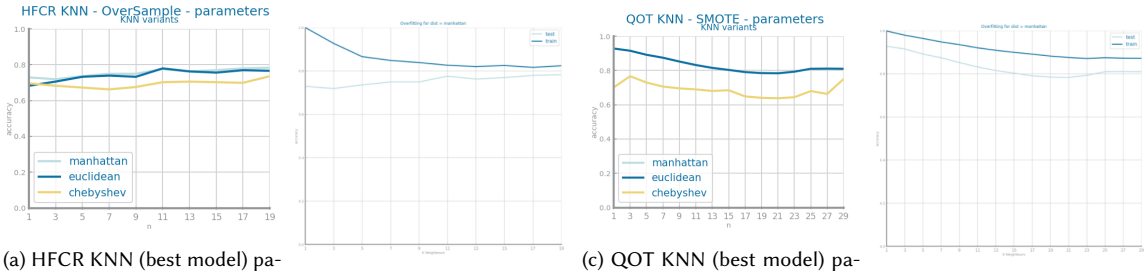


(a) HFCR KNN - Original- Performance & Confusion matrix (b) HFCR KNN (best model) Performance & Confusion matrix (c) QOT KNN - Original- Performance & Confusion matrix (d) QOT KNN (best model) Performance & Confusion matrix

Fig. 10. Performance and confusion matrix for KNN

By analysing the figures 11a and 11c, we concluded that, among manhattan, euclidean and chebyshev, the best parameters were with manhattan. With 17 neighbours and 1 neighbour, respectively for HFCR and QOT.

We concluded that the models did not enter into overfitting, since in both the test curve follows the train curve variance (11b and 11d).



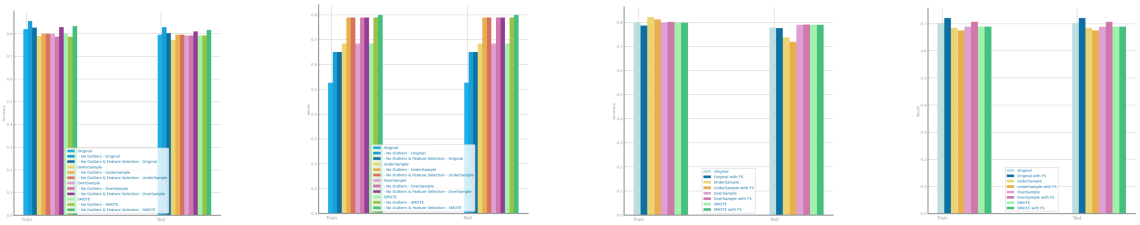
(a) HFCR KNN (best model) parameters (b) HFCR Overfitting - KNN (c) QOT KNN (best model) parameters (d) QOT Overfitting - KNN

Fig. 11. Parameters and overfitting for KNN

5.2 Naive Bayes

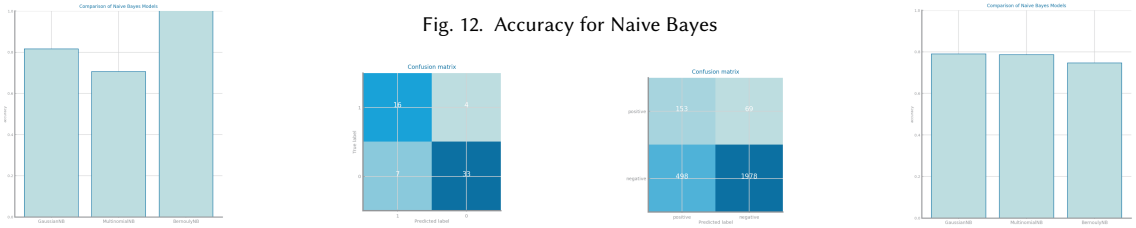
According to the histogram 12a and 12c, we can conclude that the accuracy is very similar for all the techniques, what is expected of Naive Bayes since it uses conditional probabilities, and they remain almost the same, not taking much benefit neither from scaling nor feature selection. When we balanced the data the recall increases as it did with KNN (12b and 12d). We should consider a model with feature selection since it is simpler. For SMOTE with outliers removal

and feature selection, Bernoulli gave us the best results in HFCR (13a). In QOT (13d) the best results were achieved with Oversample with feature selection. The best was Gaussian, although values did not vary much with the other techniques.



(a) HFCR Accuracy (b) HFCR Recall (c) QOT Accuracy (d) QOT Recall

Fig. 12. Accuracy for Naive Bayes



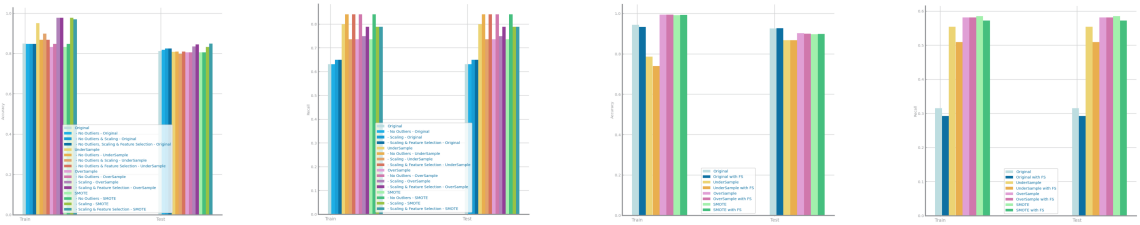
(a) HFCR Gaussian vs Multinomial vs Bernoulli (b) HFCR (Gaussian) (c) QOT (Gaussian) (d) QOT Gaussian vs Multinomial vs Bernoulli

Fig. 13. Comparison of Gaussian vs Multinomial vs Bernoulli; and confusion matrix

5.3 Decision Trees

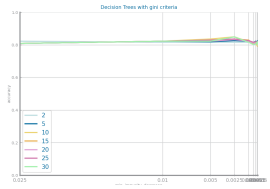
In 14a and 14c, we conclude that for both the datasets the accuracies are similar without balancing, but as with before the recall (14b and 14d) is much lower without balancing. The accuracy slightly increases with scaling in HFCR, but nothing significant, because decision trees are not based on the values itself but on the thresholds that maximize the impurity decrease of each node. So again, we choose the simpler model, in this case SMOTE with scaling and feature selection for HFCR, and Oversample with Feature Selection for QOT.

We experimented with varying several factors: criteria (entropy or gini), max depth of the tree generated and minimum impurity decrease of each node split. We decided to plot for each of the criteria the accuracy according to the max depth in order to identify the parameters that maximized the accuracy of the tree, achieving the following figures 15a and 15c. We conclude that the parameters that maximize HFCR's accuracy were gini criteria, maximum depth of 10 and minimum impurity decrease of 0.0025, when it comes to QOT the parameters were with entropy criteria, maximum depth of 30 and minimum impurity decrease of 0.000025.

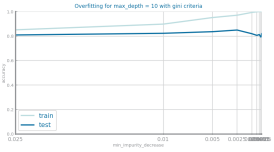


(a) HFCR Accuracy (b) HFCR Recall (c) QOT Accuracy (d) QOT Recall

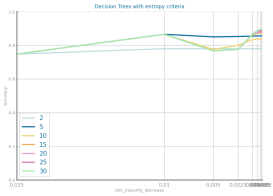
Fig. 14. Accuracy and Recall for Decision Trees



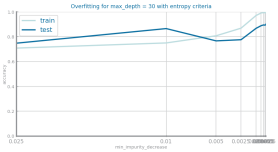
(a) HFCR Decision Trees (best model) parameters



(b) HFCR Overfitting



(c) QOT Decision Trees (best model) parameters



(d) QOT Overfitting

Fig. 15. Parameters and overfitting for Decision Trees

We then tested the trees for overfitting according to the minimum impurity decrease since it could be the main factor for a tree to be too much adjusted to the training set. From the figure 15b we conclude that the model achieved for HFCR is in fact the best, since after 0.0025 the test line starts to decline and train still increases. From 15d we conclude the same for the model achieved for QOT, since test achieves a maximum for 0.000025. The first node of each one of the trees generated and therefore the most discriminating variable is *time* and *235*, respectively for HFCR and QOT, the size of the tree was exponentially bigger with the QOT than with HFCR, this was expected since the maximum depth was bigger and the minimum impurity decrease was lower, which made more nodes to be split.

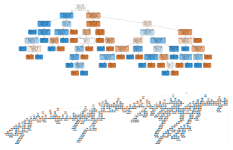
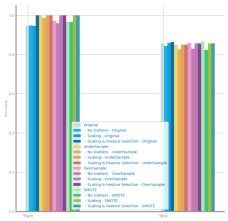


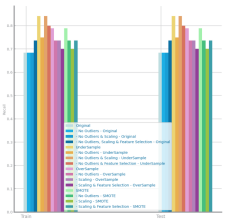
Fig. 16. Trees representation - HFCR top and QOT bottom

5.4 Random Forests

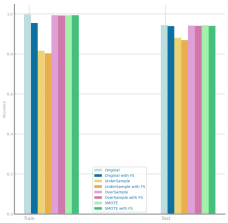
As we did for the other classifiers, we made an accuracy and recall histogram for both datasets (17) and we noticed that the QOT results clearly benefited from balancing with Undersample. There is a little improvement with scaling in HFCR. There is a little decrease on recall with Feature Selection but negligible and, since we prefer a simpler model, we decided to follow Undersample with Feature Selection for both, with HFCR having the addition of outliers removal.



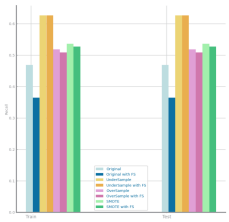
(a) HFCR Accuracy



(b) HFCR Recall



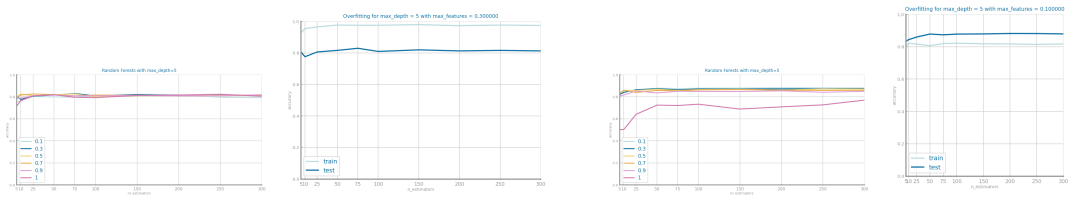
(c) QOT Accuracy



(d) QOT Recall

Fig. 17. Accuracy and Recall for Random Forests

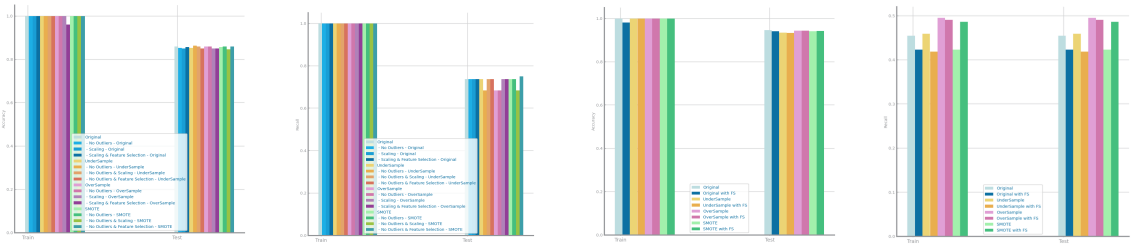
The best parameters for HFCR were with maximum depth of 5, maximum features of 0.30 and 75 estimators. For QOT the parameters that maximize the results were a depth of 5 as well, 0.10 maximum features and 200 estimators. But as we can see in the figures the number of estimators has almost no influence in the results. However, it is interesting to notice in 18b, that there is a great decrease in accuracy with a maximum features of 1, even lower with few estimators. This could mean that there are some features that deteriorate the accuracy of this ensemble.
There is no overfitting and, in fact, both for HFCR and QOT the lines tend to stabilize with the number of estimators, but of course we want the simplest model possible.



(a) HFCR (b) QOT

Fig. 18. Parameters (left) & Overfitting (right) for Random Forests

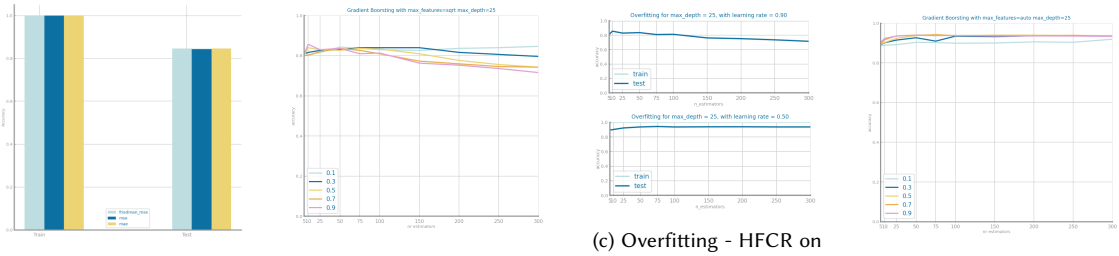
5.5 Gradient Boosting



(a) HFCR Accuracy (b) HFCR Recall (c) QOT Accuracy (d) QOT Recall

Fig. 19. Accuracy and Recall for Gradient Boosting

As for the other classifiers, we made an accuracy and recall histogram for both datasets, the variation was not big but we decided to choose due to a slight increase in recall and with the intent of choosing the simplest model, SMOTE with Feature Selection for HFCR and Oversample with Feature Selection for QOT. For HFCR, we used three different criterias: friedman mse, mae and mse. In 20a, we can see their accuracies for HFCR, there is small to none variation between them, so we decided to follow in both datasets the friedman mse criteria, since for QOT the increase in time by exploring with this different criterias would be greatly increased and most likely have no benefits. We plotted the accuracy according to the maximum depth, maximum features, learning rate and number of estimators, we conclude that the best results, with an accuracy of 0.86, are when the maximum depth of 25, maximum features following a square root, learning rate of 0.90 and a number of estimators of 10 for HFCR (20b) and for QOT (20b), the best results were achieved with a maximum depth of 25, maximum features with the criteria auto, learning rate of 0.50 and 75 estimators, resulting on an accuracy of 0.94. We then tested for overfitting according to the maximum depth and learning rate. From 20c, we conclude for HFCR that when number of estimators is superior to 10, there is overfitting, so this was the best choice and for QOT we conclude that when number of estimators is superior to 75 the test line seems to stabilize.



(a) HFCR - Criteria (b) HFCR - Parameters (c) Overfitting - HFCR on top and QOT on bottom (d) QOT - Parameters

Fig. 20. Criteria & Overfitting & Parameters analysis of accuracy according to number of estimators and learning rate