G20 - Data Science

JOÃO MARQUES, 58722, JOÃO SILVA, 76691, and JOÃO BRAVO, 84390, Instituto Superior Técnico

1 DATA PROFILING

The first dataset (the heart failure dataset, henceforth referred to as Heart) has 299 records and 13 variables. Note that the *Time* variable is disregarded since it represents when the patient died/was no longer followed. Therefore, it does not make sense to use it for classifying.

The second dataset (the toxicity dataset, henceforth referred to as Toxic) has 8992 records and 1025 variables.

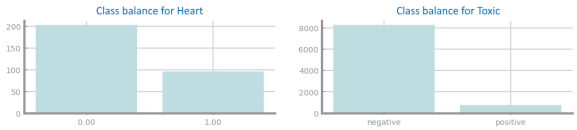

Fig. 1. Class balance

Neither dataset had any missing values, and both had far more records than variables, making it more likely that the data contains useful patterns on said variables. In both cases, the last variable is the target, which is binary (true or false, although with different representations in each set). Both datasets also had a clear imbalance in terms of the target classes, as can be seen in Figure 1, with the imbalance for Toxic being particularly high. In both datasets, each record has no relation to any other. In Heart, each record represents a patient, while, in Toxic, each record represents a chemical.

Heart has 3 floats as features and the rest of the variables are integers. It is likely that *age* and *platelets* being floats is a consequence of dealing with missing values in the original dataset. The researchers may have estimated those values and added them to their dataset, since the *platelets* float value is the average for that column, and the *age* float value is close to the average of its column. For *age*, the format of the float entries does not match the rest of the values, and, for *platelets*, as this metric is supposed to be a strict count for a certain volume, it cannot be anything but an integer. Since these abnormalities have no real impact, they were ignored.

In terms of granularity, 6 features are binary, with most of the others having a histogram with a reasonable distribution if we divide their respective range into 10 bins. *Platelets* still has a somewhat smooth looking histogram with 25 bins. The same is valid for *serum_creatinine* and *creatinine_phosphokinase* with 50 bins.

Four of the variables seem to have quite a few outliers, since, as we can see in Figure 2, these four have quite a lot of values outside the whiskers. The dataset very clearly does not cover the entire domain. Also in Figure 2, we can see two examples of sparsity. *Age* x *ejection_fraction* is the exception, displaying quite a good coverage. Many binary variables x others also have a decent coverage, but most of the other plots look, however, like *age* x *serum_creatinine*.

Looking at the correlations, the highest is 0.45 between *smoking* and *sex*, and the highest correlation between a feature and the classes is 0.29. No variable appears to be highly correlated to each other.

Toxic has 1024 integers as its features. The target itself is parsed as an object but we changed it to be a category. The data granularity is always binary, being 0 or 1 for the features, and *negative* or *positive* for the target. Checking the

Authors' address: João Marques, 58722, joao.g.marques@tecnico.ulisboa.pt; João Silva, 76691, joaobernardosilva@tecnico.ulisboa.pt; João Bravo, 84390, joaofbravo@tecnico.ulisboa.pt, Instituto Superior Técnico, Lisbon.

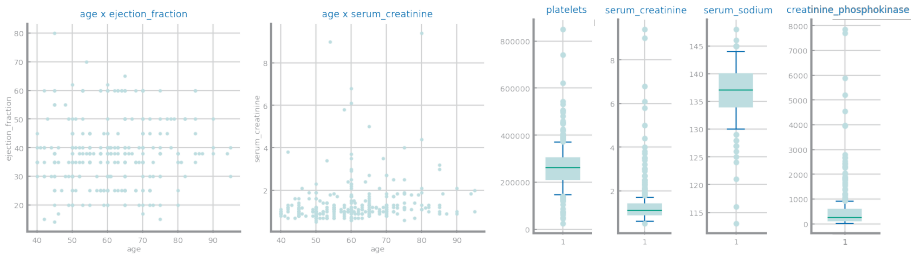

Fig. 2. Examples of sparsity and some distributions for Heart

data distribution for the features, we can see that the vast majority of them has an overwhelming majority of 0. This indicates that these molecular signatures are rare, and, even if they were directly correlated to the toxicity, they would not help much with the classification, as they rarely appear.

Fig. 3. Correlation matrix with the most correlated variables of Toxic

There are many variables in Toxic that seem to be highly correlated. In Figure 3, we can see the correlation matrix filtered to only show lines/columns with a correlation of 0.97 and above. The equivalent matrix for a threshold of 0.95 is a 27 by 27 matrix. However, considering the dataset has 1025 variables, the possibly correlated variables are a small minority. It should be noted that there is no correlation between any features and the target.

2 DATA PREPARATION

In order to decide what pre-processing should be applied to each dataset, we ran tests with different options for each of the 5 classifiers used. While some options are used universally for all classifiers, we sometimes chose to do different data preparations for a classifier or more, in order to get the most out of said classifier(s).

As mentioned before, there were no missing values on either dataset. They were, however, very unbalanced, so we started by balancing both datasets. We tried Under Sampling, Over Sampling and SMOTE. In both cases, Over Sampling was the preferred option. In Figure 4, we can see the improvement for Decision Tree, and a similar variation occurred for the other classifiers. Although there was a slight loss of accuracy, the balancing fixed the critical lack of precision the classifiers were having.

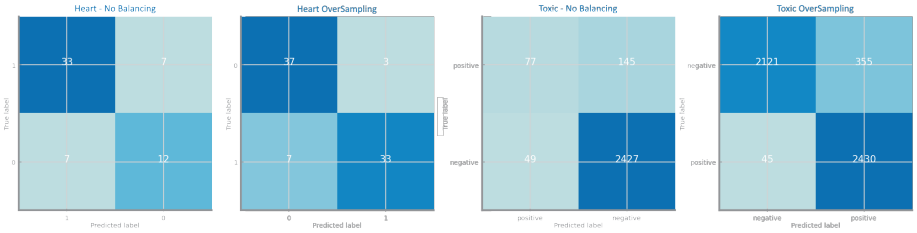

Fig. 4. Original vs Over Sampling Confusion Matrix for Heart and Toxic using Decision Tree

To test KNN, we scaled Heart only. For the other classifiers the scaling added nothing to the performance. Since both z-score and min-max gave the same result for KNN, we decided to go with min-max to always have the values within the [0,1] interval. Toxic was not scaled since it was only composed of binary 0 or 1 variables.

We created a new feature for Heart called *gfr*, calculated used Equation 1. We created this after talking with a domain expert (doctor) and getting the feedback that serum creatinine is not usually used for diagnosis, but rather to create this metric. The ethnicity of the patients is not listed in the set, but the paper that is linked to this set claims to have gotten this data from Pakistan so it was assumed the last part of the formula would not apply.

$$GFR = 186 \times (S_c)^{-0.203} \times (0.742 \text{ if female}) \times (1.210 \text{ if African-American}) \tag{1}$$

We also took the opportunity to ask what original features the expert would select, which lead to the manual selection of *age*, *diabetes*, *ejection_fraction*, *high_blood_pressure*, *sex* and *smoking*. Together with the generated *gfr*, this gave us a version of Heart with 7 features.

We then ran feature selection using K-best for both datasets. For Heart, we ran feature selection on the original features plus *gfr* to compare with the manual selection. We did not want to remove outliers manually since there did not seem to be a clear way to identify a record as an outlier when considering all the features. So, we tested if some of the algorithms for outlier removal (Isolation Forest, Elliptic Envelope, and Local Outlier Factor) improved the performance for the classifiers.

Table 1. Options selected for each classifier

	Heart		Toxic	
Classifier	Feature Selection	Outlier Removal	Feature Selection	Outlier Removal
KNN	8-best	Isolation Forest - 0.05	200-best	-
Naïve Bayes	manual by expert	-	300-best	-
Decision Tree	6-best	Isolation Forest - 0.01	300-best	-
Random Forest	5-best	-	300-best	-
Gradient Boosting	4-best	Local Outlier Factor 0.1	200-best	-

For these last two steps, the best approach varies depending on the classifier. In Table 1, we have listed the choices made when preparing each dataset for each classifier, based solely on model perfomance, with preference for simpler models in case of similar results. Note that, for Toxic, there did not seem to be outliers among the records, and none of the methods tested yielded a significant improvement for any of the classifiers. Also for Toxic, one never needs more than 300 features to keep a similar performance, meaning that, while individual correlations do occur, most features are likely multiply correlated in some way.

Fig. 5. Performance for several feature combinations for Heart using Naïve Bayes

For Heart, there seem to be a few outliers, and automatic removal slightly improves the performance for KNN, Random Forest and Gradient Boosting by around 3% of accuracy. The automatic feature selection finds a better or comparable performance with less features than the manual for most classifiers. For KNN, we picked the 8-best since it gave a slightly better performance (+3%) by adding 1 more feature. It is interesting to note that Naïve Bayes has an exceptionally good performance with the manual selection by the expert, as we can see in Figure 5. None of the automatic selections managed to achieve a similar result.

3 UNSUPERVISED LEARNING

For unsupervised learning, we used the original datasets with the targets removed.

3.1 Association Rules

To apply the following pattern mining techniques, we first preprocessed Heart by discretizing its features with a one-hot encoding, using equal-width intervals. According to our granularity studies, we decided to set the number of intervals equal to the number of bins that best describe the general data without losing significant information (as indicated in Section 1), and took into account that binary features were already dummified, except for *sex*, for which we created two binary dummies, one for each gender.

Table 2. Some of the interesting rules found

Heart					Toxic				
Prior	Consequence	Support	Conf.	Lift	Prior	Post	Support	Conf.	Lift
anaemia_0, high_blood_pressure_1, diabetes_0, ejection_fraction_3	*smoking_0*	0.03	1	1.47	72,93,62	42	0.34	1.00	1.51
anaemia_0, high_blood_pressure_1, creatinine_phosphokinase_0, sex_0, ejection_fraction_3	*smoking_0, diabetes_1*	0.01	1	3.15	72,70	93	0.13	1	1.66
creatinine_phosphokinase_0, age_4, serum_creatinine_2, serum_sodium_6	*anaemia_1, diabetes_1, platelets_4*	0.001	1	74.75	16,26,60	58,91	0.02	0.98	40.30

We did not consider equal-frequency discretization, since domain experts (doctors) always seem to use equal-range intervals in their studies. Besides, this percentile-based split would not preserve the probability distribution of each feature, transforming all into uniform distributions, which is usually less representative of the data. Dummification was not necessary for Toxic, since boolean features are, by definition, already discretized. However, due to the large nature of this dataset, we opted to transform all features into boolean types, in order to reduce the memory required to represent each one.

Toxic was also reduced by choosing the best features through K-best. Due to the high number of rules, we chose the some of the interesting results, which can be seen in table 2. We can observe that there is a very high confidence that a female (*sex*=0) with high blood pressure and ejection fraction has diabetes. This is not very frequent, as seen by the low support, and the high lift tells us the prior and the consequence are quite dependent on one another.

As for Toxic, it is more difficult to extract information due to the fact that we are in presence of molecular fingerprints and little information is known about that.

3.2 Clustering

In order to reduce the dimensionality of the dataset, the principal component analysis (PCA) was the feature extraction method used. However, for Heart, all the information is almost evenly spread across the 11 variables. Therefore, PCA becomes unuseful, resulting in a big information loss for all significant reductions. For Toxic, 80% of the information is contained in 222 variables and, so, the PCA is mandatory in this case, reducing from 1024 variables greatly.

Starting with Heart, we investigate the relation *age* vs. $ejection\_fraction$. Ejection fraction is a measurement, expressed as a percentage, of how much blood the left ventricle pumps out with each contraction. A normal heart's ejection fraction may be between 50% and 70% and less than 40% may be evidence of heart failure. This makes it an interesting study case. As for clustering, we started with the K-means method. Since only some variables follow a normal distribution, the EM method did not give great results. Density and Hierarchical methods were also tested but omitted for simplicity sake in the present study. By the Elbow Rule (Figure 6), we chose k=5.

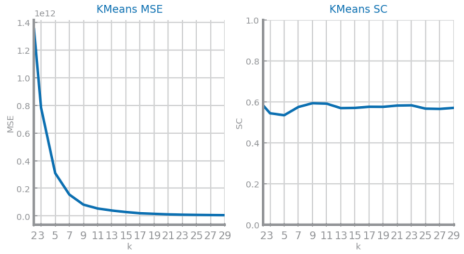

Fig. 6. Elbow Rule and SC index. Raw data.

We also experimented with selecting the best features through K-best and got $age/ejection\_fraction$ vs. $serum\_creatinine$ with similar clusters. All the clusters for k=5 can be seen in Figure 7.

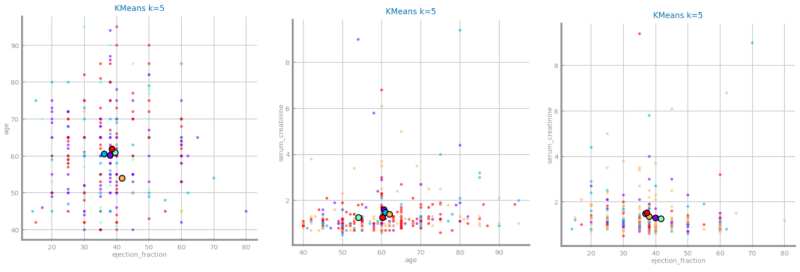

Fig. 7. Clustering $k = 5$. Raw data.

There is a perfect index $SC \simeq 0.6$ for the original data but a messy scatter plot. The SC values indicates that the clusters are well separated and cohesive, the fact that we cannot observe that in the graph for the 2 best variables indicates that these two values would not be enough to explain much of the variance. Furthermore, these 11 variables are plotted in a 2D graph which difficults the observation. Despite that, the mean squared error is very large.

The best scaling was the *min_max* which gave an MSE in the order of hundreds, but $SC \simeq 0.2$, which is quite poor. Shown by the graphs, the centroids indicate heart illnesses between the participants in the study.

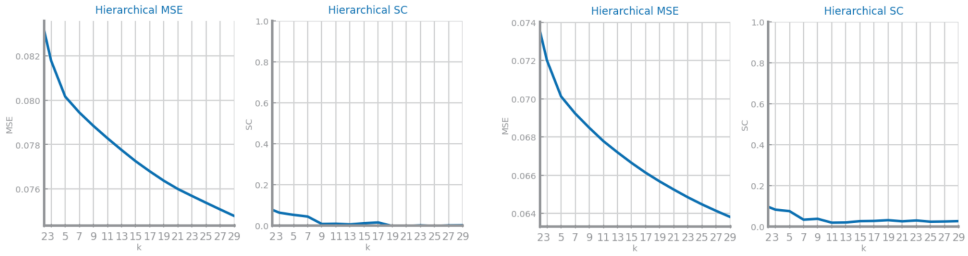

Fig. 8. Original and PCA data comparison.

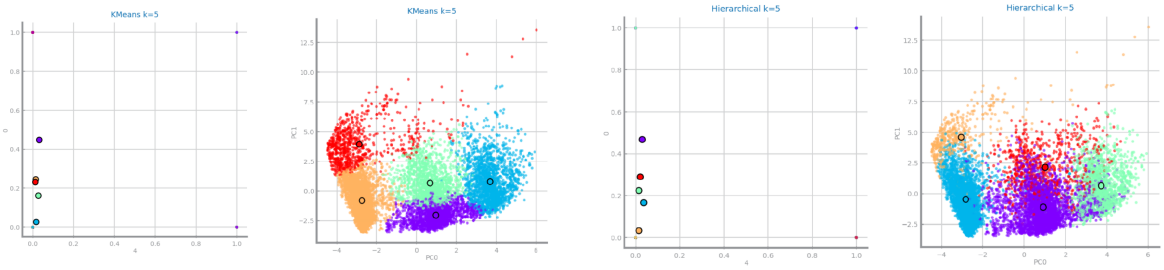

Fig. 9. Original and PCA data for $k = 5$ clusters.

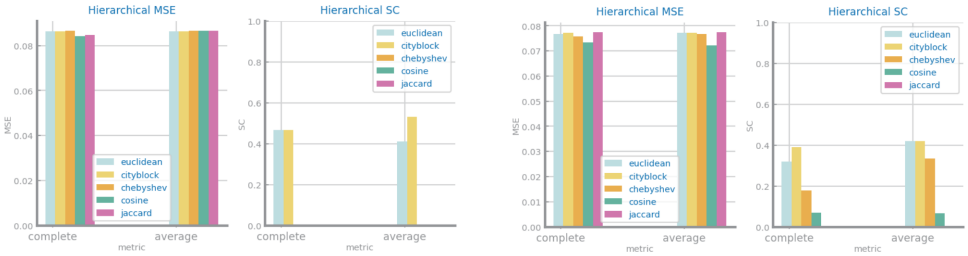

Fig. 10. Metrics and error comparison.

For Toxic, through the Elbow Rule, we chose $k = 5$ for the number of clusters. In general, the MSE is lower and SC is higher for the PCA data than for the original data (for K-means, EM and Hierarchical methods). As for the metrics, in Hierarchical clustering, the results are the same. However, the most metrics are shown in PCA data. In the Density method, all metrics have errors associated in the original dataset and only *cityblock* and *cosine* for the PCA data. For the clusters scatterplots, the original data shows only 4 points (combinations of 0 and 1) and the correspondent centroids. For the linearly independent data transformed by the PCA method, the visualization is much more illustrative and clean. A few examples of the present analysis can be seen in Figures 9, 8, 10.

4 CLASSIFICATION

In terms of training strategies, we used cross-validation with 5 folds for Heart and hold-out for Toxic. For more exact performance results, we could have run cross-validation or multiple samples over Toxic, but, given the dataset size, we decided on hold-out due to time constraints. The random state variable was set to a fixed number to ensure that the variation between the classifiers or the processing options was not due to a different split of the data or to the randomness inherent to some of the classifiers. This also applies to the study already mentioned during data preparation. Any graph that shows accuracy without explicitly marking test and train is showing the testing accuracy. We did not worry about other metrics such as precision since the balancing done made sure these two values lined up approximately[1]. Any confidence interval shown/mentioned will be at the 95% confidence level.

4.1 Naïve Bayes

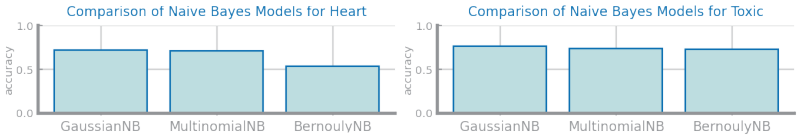

Fig. 11. Comparison of Naïve Bayes Models

Since Naïve Bayes has no parameters, we tested 3 estimators for both datasets, as shown in Figure 11. For Heart, Gaussian got the best average performance with an accuracy of 72 ± 1%. For Toxic, the best estimator was also Gaussian with an accuracy of 76%. With no configurable parameters, this classifier clearly cannot overfit.

4.2 KNN

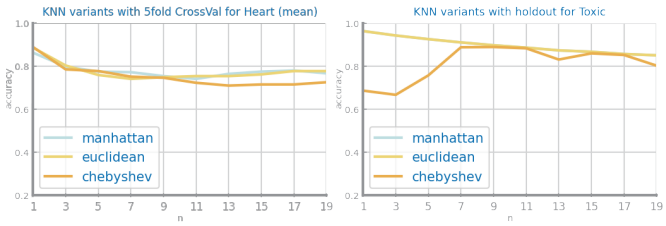

Fig. 12. Comparison of KNN using different distance metrics

[1]Naive Bayes was the exception. As shown in data preparation, it took the manual selection to achieve this alignment

For KNN, we tried three variations using different distance metrics. As shown in Figure 12, the highest accuracy obtained for Heart was 89 ± 1% with Chebyshev or Manhattan distance and 1 neighbour. All distances have their peak accuracy at 1 neighbour. This should indicate that the records with *DEATH_EVENT* are, for the most part, separated from the records without it, within the vector space. Since all the distances performed similarly, there is not much we can infer concerning the dataset.

Also in Figure 12, we can see that the highest accuracy was obtained with either Manhattan or Euclidean distance and 1 neighbour, at 96%.

Overall, adding more neighbours seems to always reduce accuracy. There is no issue of overfitting for KNN with both train and test curves following the same pattern.

4.3 Decision Trees

For decision trees, we varied the *min impurity decrease* and the *maximum depth*. In Figures 13 and 14, we can see that the best *min impurity* is the lowest in the set, as expected. However, for Heart, there is not much difference if we increase the *min impurity*, and it will likely lead to a simpler tree. The criteria also behaves similarly for both datasets, with all values having a confidence interval of around 1%.

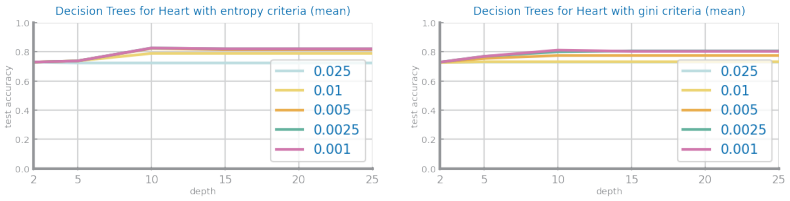

Fig. 13. Performance of Decision Trees for Heart

In terms of *maximum depth*, for Heart, it enters a plateau after it reaches 10, with an accuracy of 84 ± 0.4%. Considering that it is running on a selection of the 6 best features, this means it is splitting using some features multiple times, which is coherent with the medical knowledge[2]

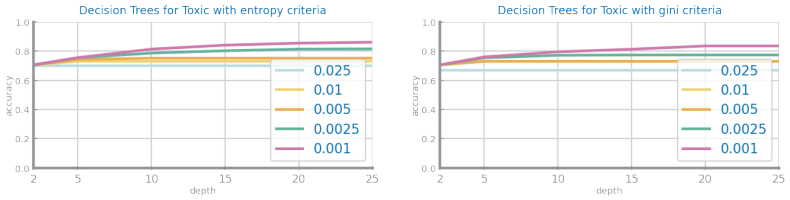

Fig. 14. Performance of Decision Trees for Toxic

Toxic does not quite hit a plateau, but the gains for depth beyond 20 are negligible. The peak accuracy is at 25 with 86%.

There was no overfitting for either dataset, and the equivalent graphs for the train set follow the same patterns, with Heart hitting the plateau at the same point.

[2]Problems are often estimated if measure X is above or below a certain value, or if measure X is above A, or above B only if Y is above C, for example.

4.4 Random Forests

For random forests, we fixed the *maximum depth* to the respective peaks of each dataset in decision trees. We tried the same two criteria we used for decision trees, and tried a range of *max features* and number of estimators.

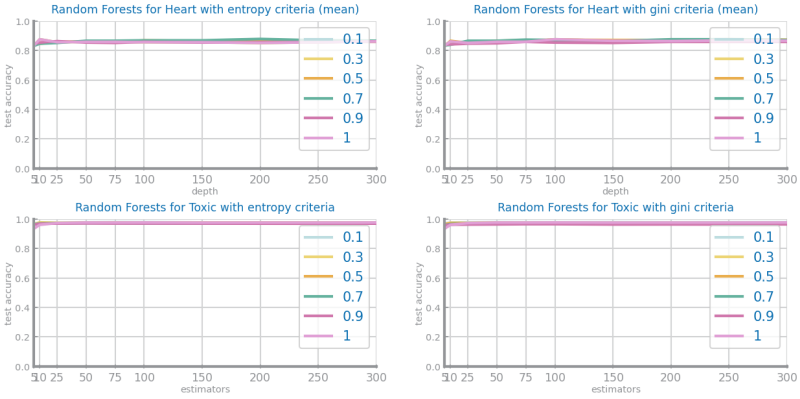

Fig. 15. Performance of Random Forests for Heart (max depth = 10) and Toxic (max depth = 25)

As we can see in Figure 15, the accuracy does not fluctuate much past 25 estimators. The *max features* also does not seem to have as much of an impact on the accuracy. This indicates that, with the selected depths, the model has no need for too many estimators and does not "care" too much about how many features it puts in each tree, since the depth is enough to ensure a good performance.

The peak accuracy for Heart is 89 ± 0.7% with 0.7 of the total features selected, 200 estimators and the entropy criterion, while the peak accuracy for Toxic is 98% with 0.1 of the total features selected, 75 estimators and the entropy criterion.

As would be expected with such flat lines, we found no overfitting. The equivalent graphs for the train accuracy follow the same patterns.

4.5 Gradient Boosting

For gradient boosting, we kept the best *maximum depth* from decision trees, and, after identifying how little *max features* impacted random forest, we fixed it at 0.5.

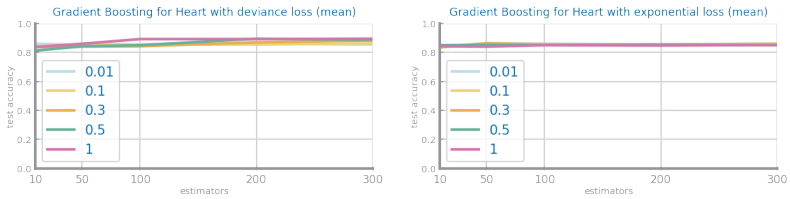

Fig. 16. Performance of Gradient Boosting for Heart, using max depth of 10 and 0.5 max features

As shown in Figure 16, the gradient boosting parameters actually have a slight impact for Heart, with the deviance *loss function* providing better results than the exponential *loss function*. The best *learning rate* with the deviance loss is 1,

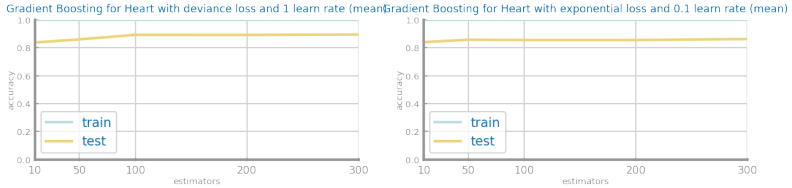

Fig. 17. Overfitting of Gradient Boosting for Heart, using max depth of 10, 0.5 max features

and the peak ($91 \pm 0.3\%$)occurs with 100 estimators. The accuracy slowly goes down with further increases in estimators. This is the first instance we have had of overfitting. In Figure 17, we can see that, while the training accuracy remains constant, the testing accuracy has the mentioned peak. We included the exponential loss, since it also overfits, even though it does have the best results.

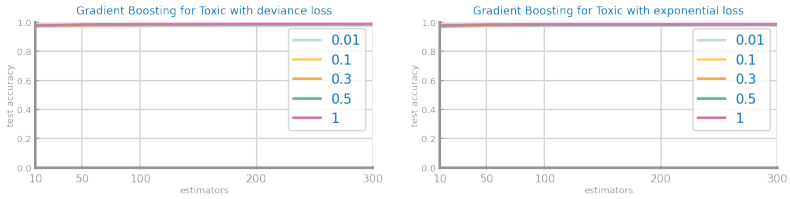

Fig. 18. Performance of Gradient Boosting for Toxic, using max depth of 25 and 0.5 max features

In Figure 18, we can see that, whatever combination of parameters we use, the accuracy is the same, being quite high. The peak is 99.2%, with deviance as the *loss function*, a *learning rate* of 1 and 200 estimators. Even during the data preparation, gradient boosting always had good results at each phase, with an original accuracy of 94.4%. Therefore, it is no surprise that, with the prepared data, it has such a high accuracy. Once again, with such flat lines, it is no surprise that there was no overfitting.

5 CONCLUSION

After all the work with the classifiers, the importance of data preparation became clear. Although parametrization can improve the classification for most classifiers, targeted data preparation gave us large gains in accuracy that we could not get otherwise. It also seems to have given the results more stability, with Heart's confidence interval never going over 1%. The fact that the expert feature selection always performed rather well for Heart, while taking a few minutes for the expert to decide, also highlights the importance of including domain experts in data science.

6 IMPROVEMENT STRATEGIES

For unsupervised learning, we would like to test a manual discretization, using the intervals for the variables that the medical community uses. For supervised learning, we would like to run Toxic with multiple sampling and cross validation for clearer results.