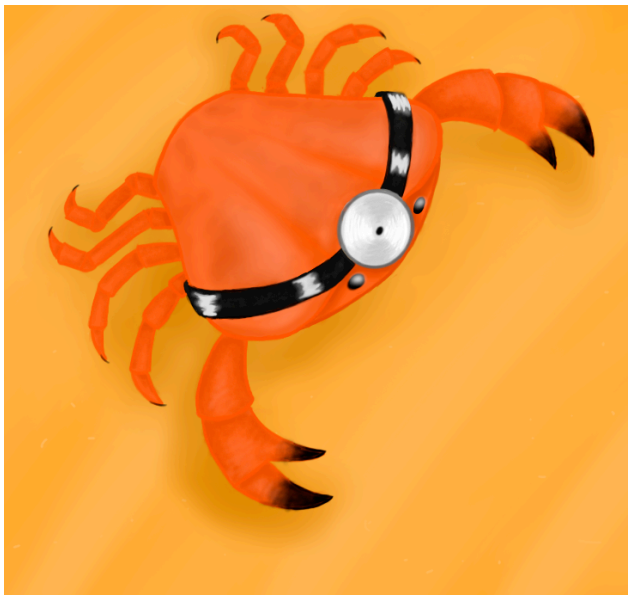
Team Nr. 20

Data Science Project



Nr 80980 Name David Calhas

Nr 81105 Name Sofia Aparício

Nr 81115 Name Rodrigo Lousada

MEIC-A SAD 2017

INDEX

1 INTRODUCTION

Data mining involves computing large datasets and determine it's patterns using machine learning combined with statistics. In machine learning provides the ability to project analytics of a determined data set. There are two predominant machine learning methods: supervised and unsupervised machine learning. In the last report some algorithms of unsupervised learning were covered, in this one we will cover supervised learning. Supervised Learning consists on receiving a train data set and, trough a specific algorithm, maps/processes a contingent function which can predict new target variables, given new examples.

This will be tested in this report comparing four supervised machine learning algorithms: KNN, Naïve Bayes, Decision Trees and Neural Networks.

KNN is the simplest of machine learning algorithms. "It is based on the principle that the samples that are similar, generally lies in close vicinity.(…) based on learning by resemblance, i.e. by comparing a given test sample with the available training samples which are similar to it." (Jadhav & Channe, 2016)

Naïve Bayes algorithm is generally used to create predictions founded on prior information and present evidence. Using the training dataset, it is possible to guess the likelihood without additional evidence. The Naïve Bayes theorem is:

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$

A - categorical outcome events
B - series of predictors

Decision Tree is the algorithm that allows to map a set of observations of a determined train data. "is a data mining induction techniques that recursively partitions a dataset of records using depth-first greedy approach or breadth-first approach until all the data items belong to a particular class." (Jadhav & Channe, 2016)

Neural Networks is "a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs." (Maureen Caudill, 1989)

2 PRE PROCESSING

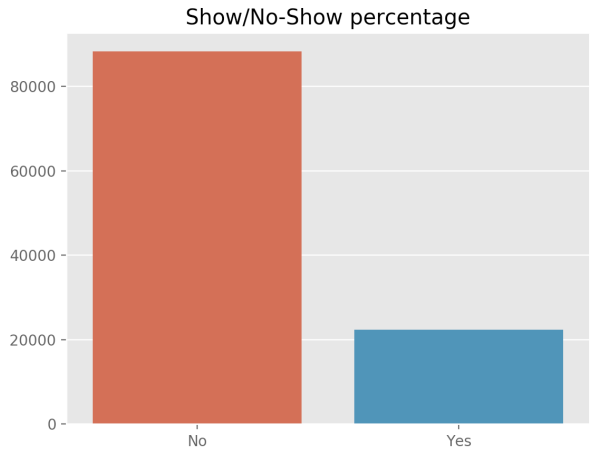
2.1 Problem 1

When pre processing the dataset of crabs, it was necessary to primarily remove the index column so it won't spoil the results. The 'sex' column of the dataset was discretized: changing the values *M* and *F*, respectively to *1* and *0*. The value that we are trying to predict is the 'sp' so it was split from the rest of the dataset into X and Y datasets before we separate them in train and test data.

2.2 Problem 2

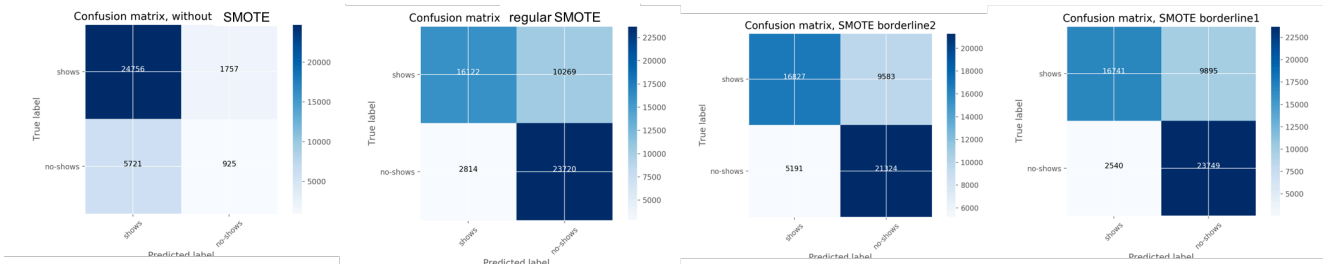
The preprocessing of the noshows dataset was almost the same as the last delivery, with minor modification.

First, to clean the data set and remove data that could make noise, the columns *AppointmentID* and *PatientId* were removed. Then, the column age was divided into categories to better comprehend the data: Baby, Infant, Child, Teenager, Young adult, Adult and Elder.



In order to make fit of the data, it was necessary to transform the columns *ScheduledDay* and *AppointmentDay*, so instead of transforming them into timestamps, we created 6 new columns: hourAppoint, *weekdayAppoint*, *monthdayAppoint*, monthAppoint, *monthSched* and *monthdaySched*, and the *ScheduledDay* and AppointmentDay columns were eliminated. Because of the same reason, to pre-process the Neighbourhood column it was necessary to relate each neighbourhood to a column, saving binary values (1 if the person leaves in that neighbourhood and 0 otherwise).

Another main alteration was to use SMOTE, in order to rebalance our data. When the knn algorithm was run for the first time in the noshows dataset, it was possible to observe that the dataset was bias/preferential, which means tending toward the most popular class, in this case the no in the class shows. So it was necessary to reduce the global quantities, disregarding the data distribution. In order to eliminate the oversampling and the under sampling we chose to use SMOTE and we analysed the confusion matrix for each type of over fitting and under fitting. Analysing the matrixes we chose the SMOTE kind borderline2.



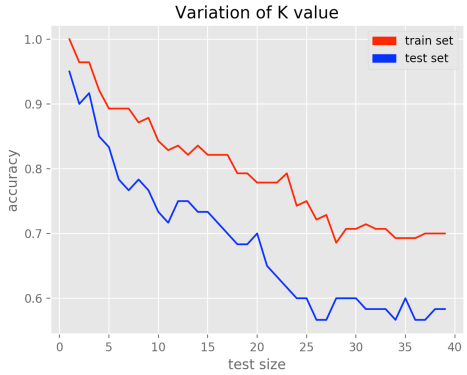
3 EXPLORATION

3.1 Problem 1

3.1.1 Methods and Parameterization

KNN

After running the K-nearest neighbor approach (Instance Based Learning), different values for k and train/test size were tested, generating the following results.



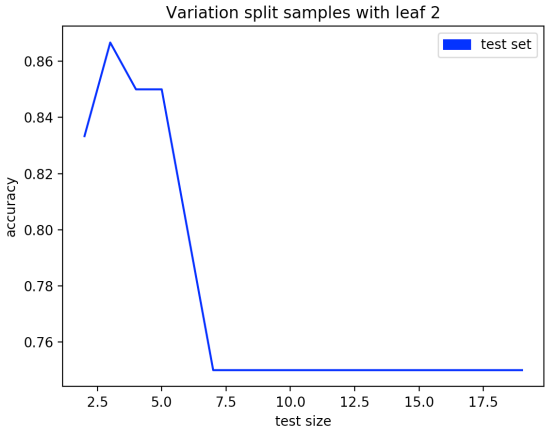
The test set accuracy drops proportionally with train data size, and is possible to determinate optimal values for both (train/test sets) with 30% of test size. For this test_size, the k value is optimal at 1, falling with bigger values.

Naïve Bayes

First it was necessary to create a data partition, using the function createDataPartition with 80% of data that goes to training, and then as created a test data with the remaining data.

Decision Tree

The decision tree tends to overfit the data, the overfit can be blamed on the noise in the attribute values and the class information present on the training data. For that reason, it was necessary to prune the tree, subtrees were replaced with leaves. Even though sometimes the unpruned tree might provide better accuracy results and the pruning is almost always based on a statistic model, there were made tests when the values of slit and leaf were altered in order to prune the tree and optimize the accuracy value. It was used the gini index criterion lectured in classes which is the default for DecisionTreeCalssifier.

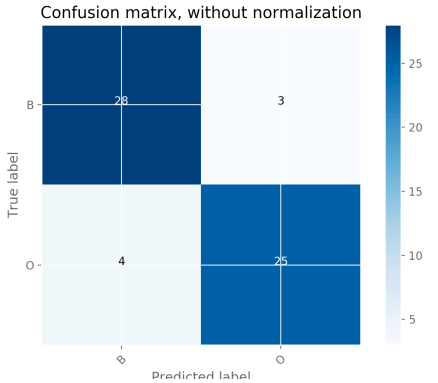


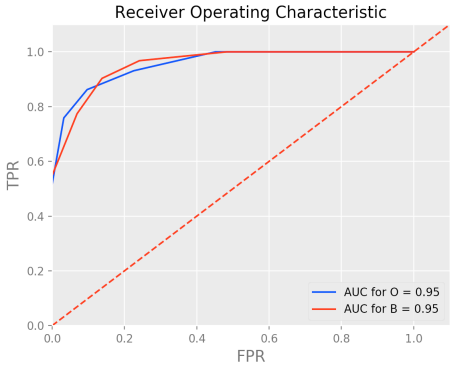
Regarding the DecisionTreeClassifier it was tested values from 1 to 20 for min_sample_leaf and from 2 to 20 for min_sample_split. After analysing different graphics generated for this analysis, the following showed great values at min_sample_leaf=2 and min_sample_lift=6. As explained above in order to prune the tree and optimize the accuracy value, the value 6 was chosen over the min_sample_lift=3.

3.1.2 Results

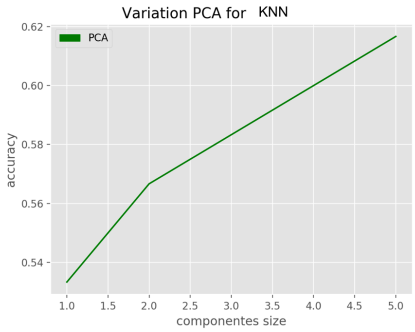
KNN

For test_size=0.3 and K=1 the following complex matrix and ROC chart were generated:





In the complex matrix generated with the optimal values, it is possible to see only 7(4+3) failed predictions which explains the results near a ROC Heaven.
Accuracy: 0,9000

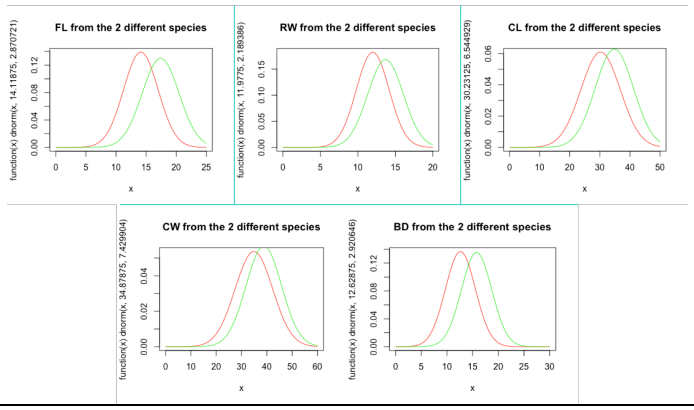


PCA converts a set of observation variables, possibly co-related, into a set of variables values linearly not co-related. This are called principal components. In the algorithm, we differentiated the the principal components number and associated it with the accuracy. As it is possible to confirm, in the PCA the accuracy is worst, and it drops when the value of the components decreases.

Naïve Bayes

Running the function naive_bayes with the traindata as argument, it was possible determine the a priori probability of each specie: B – 0.5 and O – 0.5. Regarding the tables given for each column, where was provided the mean and the standard deviation, it was possible to extract the following normal distribution for each attributes (were the red line represents the specie B and the green the specie O).

Posteriorly, the function predictand and confusionMatrix were run with the testdata and the results from the Naïve Bayes classification. It was possible to obtain the results presented in the image on the right.



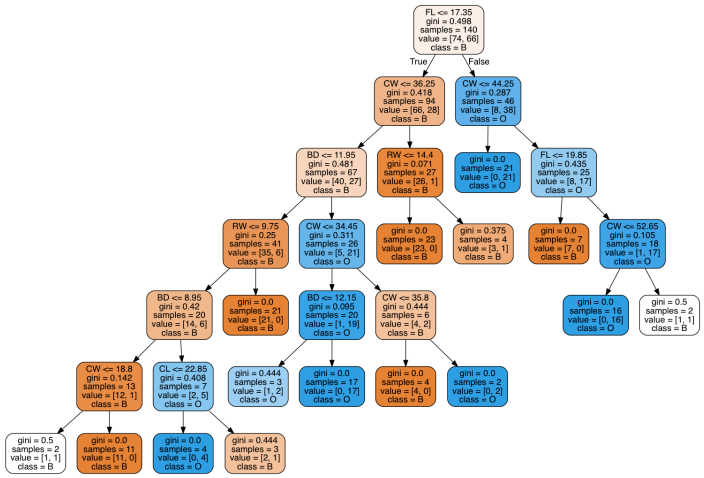

```
Confusion Matrix and Statistics

          Reference
Prediction  B  O
         B 12  9
         O  8 11

               Accuracy : 0.575
                 95% CI : (0.4089, 0.7296)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : 0.2148

                  Kappa : 0.15
 Mcnemar's Test P-Value : 1.0000

            Sensitivity : 0.6000
            Specificity : 0.5500
         Pos Pred Value : 0.5714
         Neg Pred Value : 0.5789
             Prevalence : 0.5000
         Detection Rate : 0.3000
   Detection Prevalence : 0.5250
      Balanced Accuracy : 0.5750

       'Positive' Class : B
```

Decision Tree

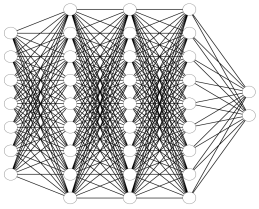
According to methods and the parametrization described for this algorithm, the tree on the left was generated. Gini index is higher for FL <= 17.35, choosing it for root of the decision tree. As it is possible to

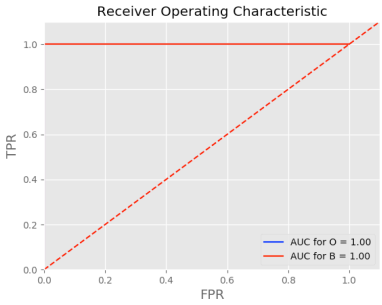
see on the second level, we have different gini indexes about the same attribute, having a higher index for a more restrictive interval of CW, this is because the gini index is calculated for each specific case. In every decision the arrow on the left is for True and the arrow on the right is for False, as described in the root.

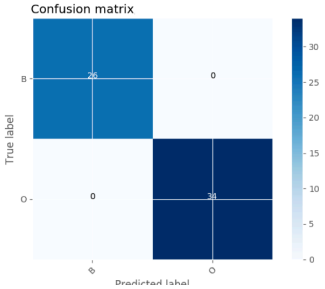
Accuracy: 0,8667



Neural Networks







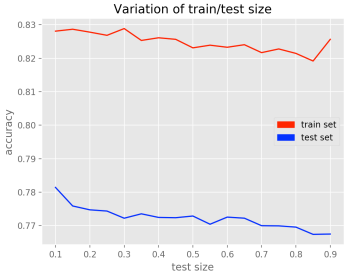
With the following methods and parameterization defined earlier is possible to achieve a ROC Heaven, which illustrates a perfect result. This is achieved by a highly dense neural network for a dataset like crabs. Standardization had a huge impact on this results, making them more regular, since without it we could get drastically different results every time we run the neural network.

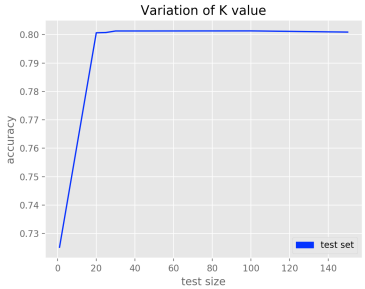
3.2 Problem 2

3.2.1 Methods and Parameterization

KNN

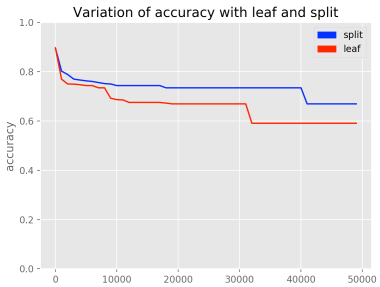
After running the K-nearest neighbour approach (Instance Based Learning), different values for k and train/test size were tested, generating the following results.





It is possible to observe that with the increasing the value of k, the accuracy also increases. Even though we can not determine for sure why the value increases exponentially until it reaches the stability point at the value k = 20, it is possible to speculate that this happens because of the large amount of data and because the data can be grouped in a big group ignoring the little amount of data that runs out of the tendency.

Decision Trees



Regarding the DecisionTreeClassifier it was tested values from 1 to 50000 for min_sample_leaf and from 2 to 50000 for min_sample_split.
After analysing the graph, it was possible to conclude that the optimal values for the accuracy are not the
ones pruned. But in this case the dataset is too big, so it is better to determine values for the split and the lift in order to better analyse the decision tree.
In order to have a better analysis, it was generated the results for min_sample_leaf=3000 and min_sample_lift=3000.

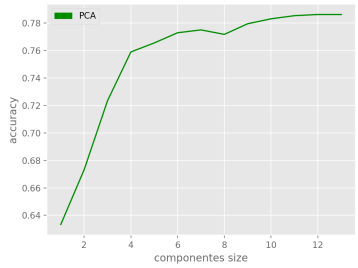
Accuracy: 0,7484

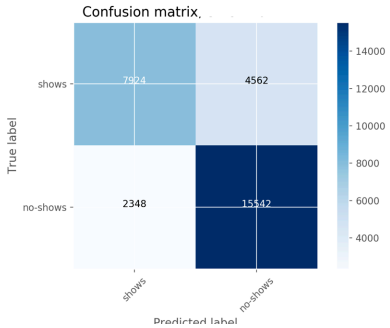
Neural Networks

Comparing the neural network parameterization designed for crabs dataset, when ran the nos-hows dataset, it was only possible to achieve accuracy level around 74%. It is normal since the number of nodes of the input layer is much higher than the crabs one. Considering the random factor mentioned before, it's only possible to achieve an accuracy level of 100% with a highly dense neural network for a dataset like nos-hows, which means to many hidden layers with a really high number of nodes in each layer. As this means a ridiculous computation time with few improvements in terms of performance, we chose to show the results for 5 hidden layers with respectively 100, 100, 90, 90, 50 nodes, once it shows a better accuracy than with 100, 100, 100, 100, 50 (around 80% against 77%), values where the accuracy begins to drop.

3.2.2 Results

KNN





For test_size=0.3 and K=20 the following complex matrix. It is possible to determine that (4562+2348)
failed the test. It was also possible to obtain the following classification report:

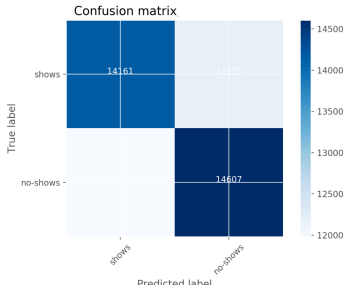
k-NN accuracy(score) for test set: 0.770855

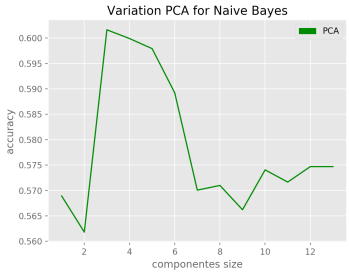
	precision	recall	f1-score	support
shows	0.76	0.64	0.69	12332
no-shows	0.78	0.86	0.82	17985
avg/total	0.77	0.77	0.77	30317

As it possible to witness, the accuracy is not 80 % as it was shown on graph of the variation of k, in the Methods and Parameterization. It is possible to explain that considering the random random factor associated with the Classifiers.

The PCA for this data set gave good results, compared to accuracy of the algorithm without the PCA.

Naïve Bayes





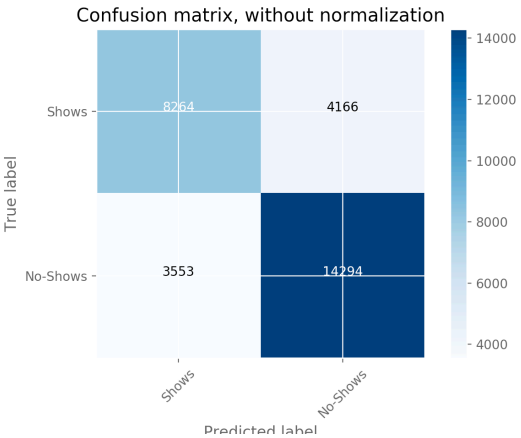
Running the function naive_bayes with the traindata as argument, it was possible to observe that the accuracy was 54.0 %.

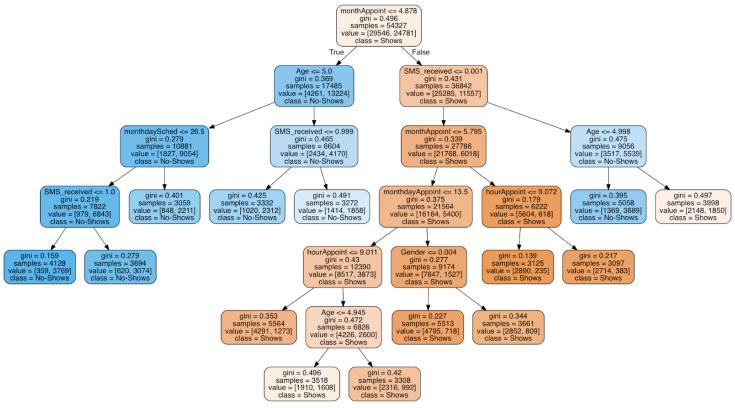
Posteriorly, the function predictand and confusionMatrix were run with the testdata and the results from the Naïve Bayes classification. It was possible to obtain the results presented in the table.

	precision	recall	f1-score	support
shows	0.54	0.54	0.54	26336
no-shows	0.55	0.55	0.55	26589
avg/total	0.54	0.54	0.54	52925

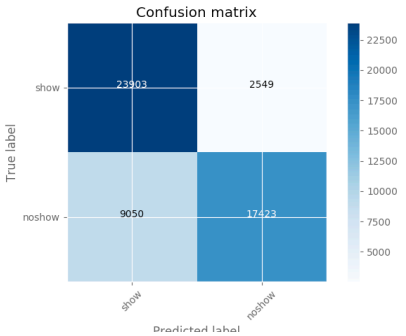
With the PCA graph presented above, it is possible to conclude that it reaches best accuracy levels than the normal Gaussian Naïve Bayes.

Decision Tree





According to methods and the parameterization described for this algorithm, the tree on the left was generated. Gini index is higher for the condition of monthAppoint <= 4.878, choosing it for root of the decision tree. As it is possible to see, if it's True it will be easier to determinate the attendance to the appointment because the subtree generated is shorter. This actually doesn't mean we would have more precise predictions from the left size. In every decision the arrow on the left is for True and the arrow on the right is for False, as described in the root.



Neural Networks

With the following methods and parameterization defined earlier is possible to achieve a really good accuracy around 80%. However, the cost of running the algorithm with this specifications is too much for an improvement of around 6%, when compared with the parameterization for the crabs dataset. The following results are achieved by a huge density in the neural network (too complex to illustrate). Standardization and oversampling had a huge impact on this results, making them more regular and improving the accuracy.

4 CRITICAL ANALYSIS

For the crabs dataset, ANNs are the most appropriate algorithms, although almost every algorithm that was implemented and tested in crabs, had a relatively good performance, excluding Naive Bayes. ANNs really looked apart with 100% accuracy for the parameterization chosen.

Regarding the no-shows dataset, we have to consider that it is a much more complex kind of data and it is really difficult, if not impossible, for a simple algorithm such as Naive Bayes, that already had a bad performance in the crabs dataset, to perform reasonably in no-shows. ANNs couldn't achieve the results like for crab dataset, needing a much more complex neural network with a horrible performance to achieve the similar results to KNN.

SMOTE had a huge impact in no-shows results, solving the problem of the unbalanced data, by oversampling methodology.

PCA improves Naive Bayes performance, however it is decreased when applied to KNN algorithm on the crabs dataset and increased with the no-shows algorithm.

It is possible to analyse the different accuracy values and complex matrixes to compare the different performances for every model.

While the crabs dataset was relatively easy to pre-process, a lot of critical choices were made for pre-processing the no-shows data set in order to improve our accuracy levels. Neighbourhoods discretization instead of deleting the attribute was really important for the result showed before.

5 CONCLUSIONS

The results presented above were the product of various occurrences for each module, algorithm and parameterization, since it was decided to take advantage of the random factor associated with the Classifiers which would have different rows assigned for the same train/test size, every time the algorithms were run.

For the crabs dataset the model that had better performance was ANNs. The crabs data set simpler than the no-shows dataset, and for that reason it was already expected that a better performance would be achieved. This can be justified using Kolmogorov complexity. It is much more challenging to describe a dataset such as no-shows, than to describe the crabs dataset, regarding both the pre processing and methods/parameterization.

For the no-shows dataset we had better results with the ANNs and KNN algorithms, however for practically the same accuracy, KNN runs with better results performance than ANN's. This proves that ANN's helps us to reach higher levels of accuracy, however for a complex dataset like no-shows that presents great performances in algorithms like KNN, it cost much more to achieve practically the same results.

Naive Bayes is a more simplistic algorithm which compared with the others presented the little success.

On overall, good results were achieved, reaching accuracies of more than 80% for both datasets. In a real context, only the ANN's for the crabs dataset is viable, but it's possible to achieve good predictions for the attendance in appointments. It is obvious that some algorithms are prepared for more complex datasets than others.