Dealing with imbalanced data

Supervised Learning Project

Mathieu Pont and Lucas Rodrigues Pereira

January 2020

Abstract

1 Introduction

In the context of the Master's Degree on Machine Learning for Data Science, at the University of Paris (Descartes), we have been given the task to discuss some supervised algorithms on synthetic data sets and imbalanced data sets, as part of the "Supervised Learning" course.

2 Experiments and Results

2.1 Synthetic Data Sets

At first we run many different supervised algorithms on synthetic data sets presented below.

	#samples	#features	#classes
Flame	240	2	2
Spiral	312	2	3
Aggregation	788	2	7

Table 1: Synthetic data sets description.

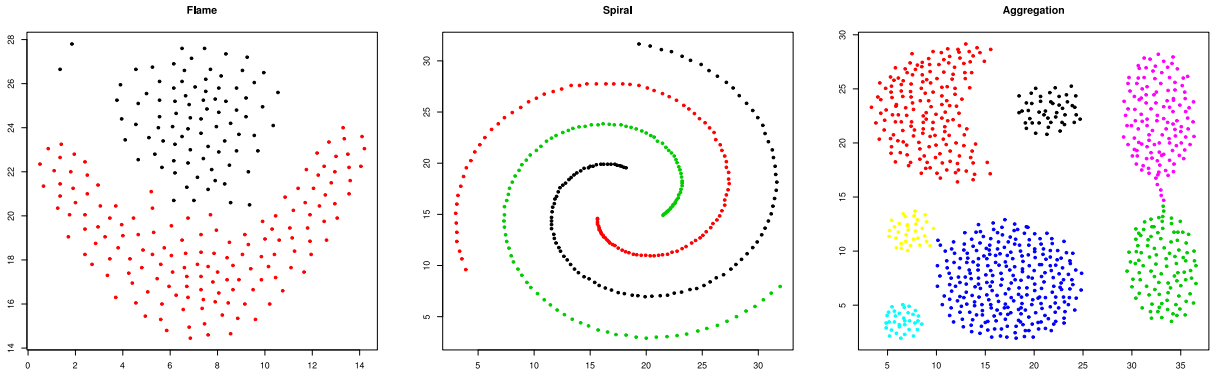


Figure 1: Synthetic data sets plot (from left to right: Flame, Spiral and Aggregation).

Each data set has a peculiarity making it interesting to test algorithms on. For example Flame is not linearly separable and has outliers. Spiral is a hard known classification problem where the class are not linearly separable and where the decision boundaries can be complicated to find. Finally Aggregation has many classes with different shape and size, some of them are close, for example the two classes at the right can fool a CAH with a minimum criterion.

Moreover, only Flame and Aggregation are imbalanced contrary to Spiral. We have decided to use here a random under-sampling method that will be described in 2.2.

Method	**Flame**	**Spiral**	**Aggregation**
Logistic Regression	0.90 ± 0.14	0.31 ± 0.15	**1.00**
LDA	0.90 ± 0.11	0.31 ± 0.20	0.99 ± 0.01
QDA	0.96 ± 0.10	0.27 ± 0.12	**1.00**
KNN	**0.99 ± 0.02**	0.94 ± 0.10	**1.00**
CART	0.97 ± 0.10	0.91 ± 0.13	**1.00**
Gaussian Naive Bayes	0.96 ± 0.10	0.27 ± 0.10	**1.00**
SVM	**0.99 ± 0.03**	0.93 ± 0.13	**1.00**
AdaBoost	0.98 ± 0.04	0.81 ± 0.14	0.81 ± 0.03
GradientBoosting	0.96 ± 0.10	0.88 ± 0.12	0.99 ± 0.02
Random Forest	0.97 ± 0.08	0.89 ± 0.13	**1.00**
ExtraTrees	**0.99 ± 0.03**	**0.96 ± 0.10**	**1.00**

Table 2: Mean accuracy and standard deviation with k-fold cross-validation ($k = 10$).

For all data sets the under-sampling method highly increases the results compared to those without it. Overall the results are very good except for Spiral due to its non-linear classes boundaries. However, some algorithms perform well on the latter, for example kNN, since the points of each cluster are well grouped together it is not fooled by the other clusters. ExtraTrees has very good results and is one of the best algorithm for each data set.

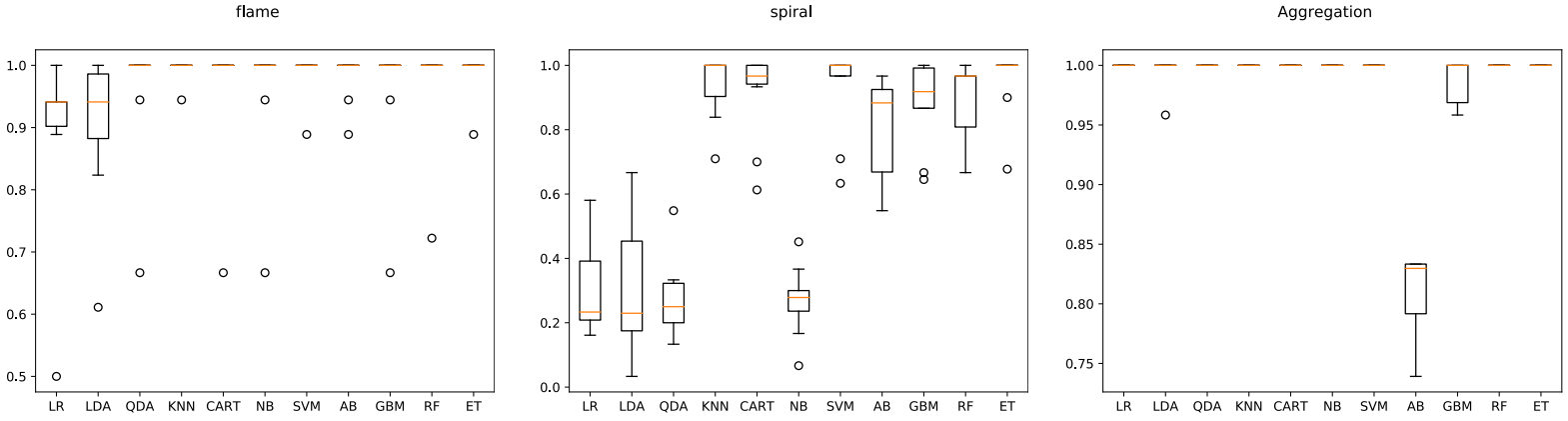


Figure 2: Algorithm comparison for each data set, boxplot of all folds for each algorithm.

The variance of folds for Flame and Aggregation are very low compared to Spiral due to its peculiarity. Interestingly, LDA still manages to have a good result for a fold, but has however a big variance due to its linear decision boundaries.

2.2 Dealing with imbalanced data

Most machine learning algorithm will struggle to fit a classifier when few classes are over represented in the dataset. For example, if a class is presented 1% of the time, disregarding it will guarantee 99% accuracy. The confusion matrix to describe the quality of a model is pointless in this case.

Many approaches have been proposed over the years, from over-sampling to under-sampling, both aim to balance the data set. The first one consists in generating samples of the class having the less. The second one consists in selecting samples from the class having the more.

The main drawback of over-sampling is that the test set could not be independent of the train set anymore leading to an over-estimation of the metrics.

The main drawback of under-sampling is that we can loose important information by selecting samples. We can fix this by using specific methods that will be described here after.

Other methods combine both kinds of sampling, such as ROSE [Lunardon et al., 2014] or SMOTE [Chawla et al., 2002].

In this study, we have chosen to experiment different methods of under-sampling, because the over-sampling, for extremely imbalanced datasets can lead to overfitting.

2.2.1 Under-sampling.

Under-sampling is commonly used to tackle imbalanced datasets. The idea is to select observations of the bigger class, keeping only as many observations as in the smaller class. Three under-sampling methods will we tested: Random, NearMiss and Centroids. The selection of observations is made so as to keep the same amount of observations for each class. Therefore, it is limited to the size of the smallest class.

- **Random**: random observations were selected.
- **NearMiss**: the observations that are closest to the smaller class were selected.
- **Centroid**: using K-means, the observations closest to the centroids of each class were selected.

The Python package *imblearn* was used in order to apply these methods.

2.3 Visa Premier

2.3.1 Preliminary Study

Each sample of this data set corresponds to a client of a bank that is described by its behaviours. The idea is to predict a binary variable corresponding to the fact the client is in possession of the Visa Premier card or not.

	#samples	**#features**	**#classes**	**Balance**
Visa Premier	1073	47	2	0.33

Table 3: Visa Premier fraud data set description.

The data set is quite imbalanced with almost 2/3 of samples being in the first class (not in possession of the Visa Premier card).

Pre-processing. The variable *matricul* is an identifier and has a unique value for each sample making it not important for our analysis. Moreover, the variable *nbimpaye* has the value of zero for each sample. Therefore, we have decided to remove these variables. The variable *nbbon* and *mtbon* have the same value for each sample except only one, we can consider it as an outlier and remove it and the two variables cited.

Most of the variables are quantitative except 7 that are qualitative:

- **departem**: residency department.
- **ptvente**: selling point.
- **sexe**: sexe.
- **sitfamil**: marital status.
- **csp**: socio-professional category.
- **sexer**: sexe (binary values)
- **codeqlt**: bank-assessed customer quality (ordinal).

We see that the variable *sexe* appears twice, one with character values and the other one with binary values. The same comment is true for the variable to predict *cartevpr*. Therefore, we can remove these duplicates and keep the numerical variables.

After this pre-processing the final dataset has the following dimension: 1072×42 where the last column corresponds to the label.

424 values seem to be missing and have "." as a value, involving the following variables: *departem*, *codeqlt*, *agemvt* and *nbpaiecb*. Many methods exist to handle missing data, one of the easier way is to replace the missing values by the mean, median or mode of the variable. For the two qualitative variables we have chosen the mode, and for the two quantitative variables the median since they are discrete variables (in order to keep the notion of discrete).

Exploratory Analysis. We saw through boxplots that each variable have more or less the same descriptive statistics among samples of one class and samples of the other. It means that none of them, alone, is discriminative according to these criteria. Due to the hight number of variables and because information given by these boxplots is kind of pointless we have decided to not show them here.

The variables do not have at all same range of values, for example, some of them are in francs whereas other represent the age, some normalization could be interesting in the supervised classification.

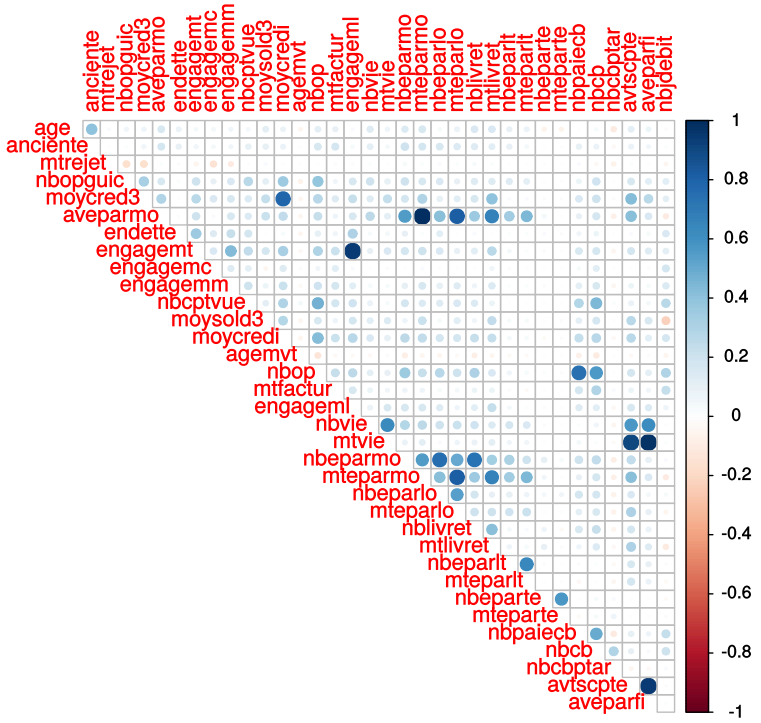


Figure 3: Variables correlation for the Visa Premier data set .

mteparmo	*aveparmo*	0.99
aveparfi	*mtvie*	0.98
aveparfi	*avtscpte*	0.95
engageml	*engagemt*	0.95
avtscpte	*mtvie*	0.90
mteparlo	*mteparmo*	0.82
mteparlo	*aveparmo*	0.82
moycredi	*moycred3*	0.79
nbeparlo	*nbeparmo*	0.76

aveparfi	Total of financial savings assets in francs
aveparmo	Total of monetary savings assets in francs
avtscpte	Total of assets over all account
engagemt	Total of commitments in francs
engageml	Long-term commitment
nbeparlo	Number of home savings products
nbeparmo	Number of monetary savings products
moycredi	Mean of credit transactions in kF
moycred3	Mean of net credit transactions of the last 3 months in kF
mteparlo	Amount of home savings products in francs
mteparmo	Amount of monetary savings products in francs
mtvie	Amount of life insurance products in francs

Table 4: Highest correlations between variables and their definition.

Some variables seem to be highly correlated or even completely for some of them. Interestingly the number of life insurance products (*mtvie*) is extremely correlated with the total of financial savings assets (*aveparfi*), the latter being also highly correlated with the total of assets (*avtscpte*) itself correlated with *mtvie*. These three variables are highly correlated with each other. The same comment can be made for *mteparmo*, *aveparmo* and *mteparlo* meaning that monetary and home savings products are linked.

PCA is a classical method to visualize in two dimensions a data set, but the latter can't take into account qualitative variables. We can however run PCA without these variables and have a pretty good result since we don't have much qualitative variables. Nevertheless, it would have been even better to use an algorithm that can take into account both types of variables, this is the purpose of the FAMD (Factor Analysis of Mixed Data).

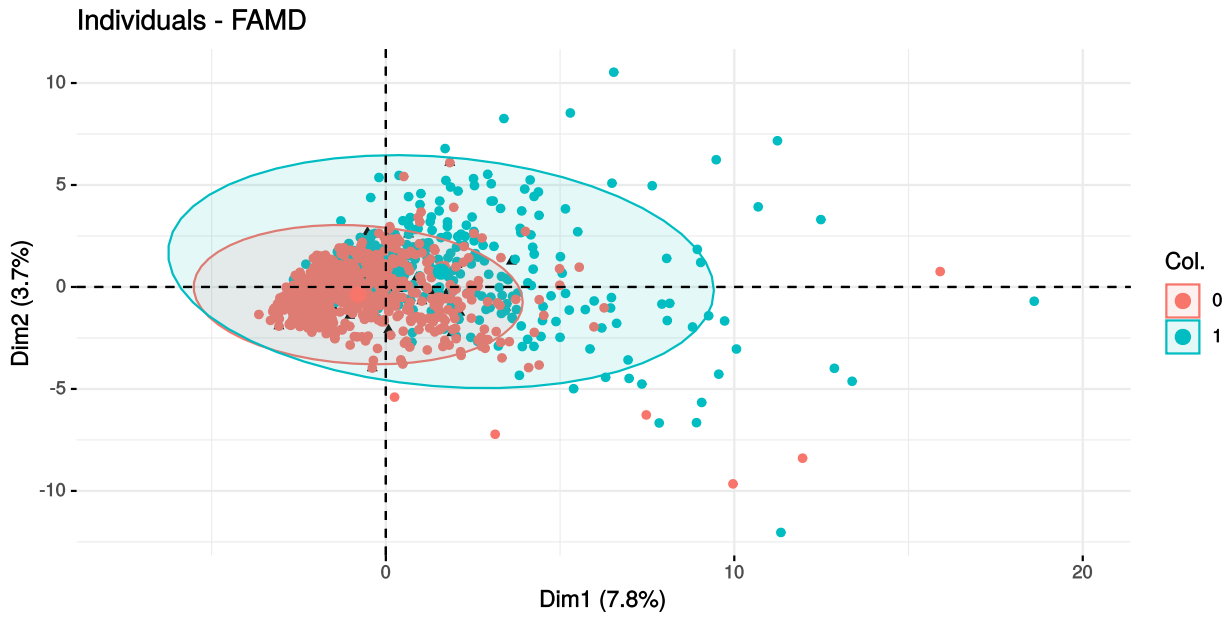
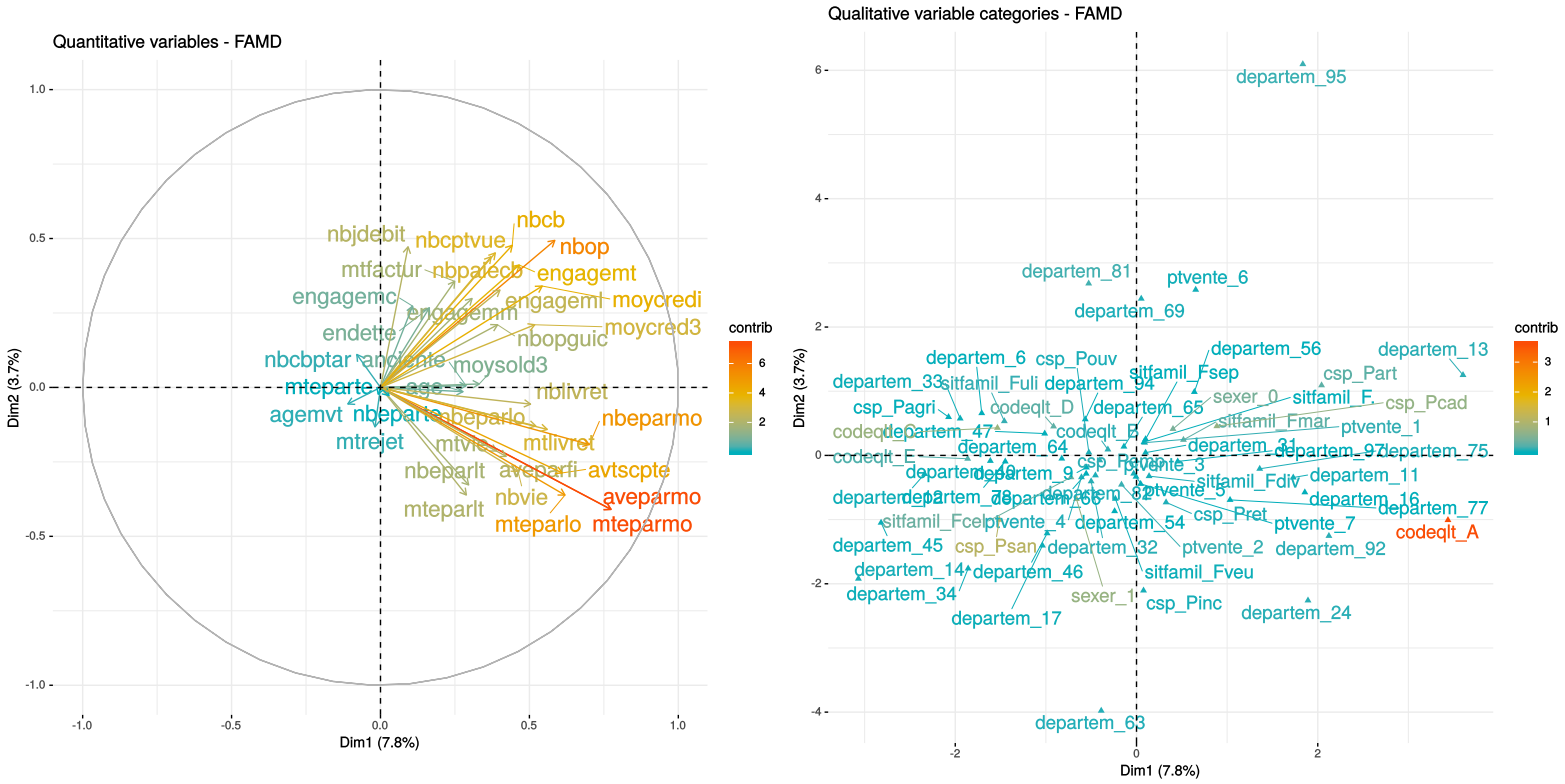


Figure 4: Individuals factor map of FAMD.



(a) Quantitative variables factor map.

(b) Qualitative variables factor map.

Figure 5: Variables factor maps of FAMD.

On the individuals factor map of Figure 4 we see that both classes are overlapping. The cluster of those who have not the Visa Premier card seems to be "inside" the cluster of those who have it. The latter is more like an ellipsoid than the other (that is more a circle) with a shape kind of ellongated to the right and a little to the top. Therefore we can say that variables that will be highly correlated with the first two components could be important to discriminate them.

We see in the qualitative variables factor map of Figure 5b that this kind of variable seems to not bring much variance to the data since their contribution to the components are much lower than for quantitative variables. However, the modality "A" of the qualitative variable "codeqlt" seems to have a high contribution and is highly correlated with the first component, this variable represents a customer quality defined by the bank.

We found that PCA and FAMD give more or less the same result (for this data set) but the latter gives in addition information about qualitative variables. However, the latter explains less variance (10%) than PCA (25%) for the first two components. This can be explained by the fact that qualitative variables are not implicated much in the variance of the data set and that FAMD is bringed down when trying to take into account this kind of variable.

2.3.2 Classification

At first we have removed the qualitative variables to do a supervised classification with the same algorithms than for synthetic data sets. Since we saw with FAMD that, overall, qualitative variables do not participate much to the variance of the data set we can more or less remove them safely. However, we

could have taken (correctly) into account this kind of variables by transforming them into a *complete disjunctive table*.

In order to deal with this unbalanced data set and have meaningful results we used the three under-sampling methods presented above.

Method	Centroid	NearMiss	Random
Logistic Regression	0.83 ± 0.04	**0.91 ± 0.03**	0.86 ± 0.04
LDA	0.81 ± 0.03	**0.88 ± 0.03**	0.85 ± 0.02
QDA	0.57 ± 0.09	**0.80 ± 0.07**	0.74 ± 0.07
KNN	0.76 ± 0.07	**0.84 ± 0.03**	0.80 ± 0.04
CART	0.83 ± 0.05	**0.87 ± 0.05**	0.85 ± 0.04
Gaussian Naive Bayes	0.71 ± 0.09	**0.83 ± 0.03**	0.76 ± 0.02
SVM	0.85 ± 0.06	**0.90 ± 0.03**	0.87 ± 0.05
AdaBoost	0.88 ± 0.06	**0.93 ± 0.04**	0.92 ± 0.04
GradientBoosting	0.90 ± 0.09	**0.94 ± 0.04**	0.91 ± 0.08
Random Forest	0.88 ± 0.04	**0.93 ± 0.04**	0.90 ± 0.04
ExtraTrees	0.87 ± 0.03	**0.95 ± 0.02**	0.90 ± 0.06

Table 5: Mean accuracy and standard deviation with k-fold cross-validation ($k = 10$) for three types of under sampling methods.

Without a doubt, that is the NearMiss under-sampling method that performs the better here, followed by the Random method. Interestingly, QDA performs worst than LDA for each experiment, as pinpoint by [James et al., 2014] it can be the case for some experimental setup, for example when we have few samples, the estimation of the additional parameters of QDA (compared to LDA) can make it worse "since it t a more exible classifier than necessary".

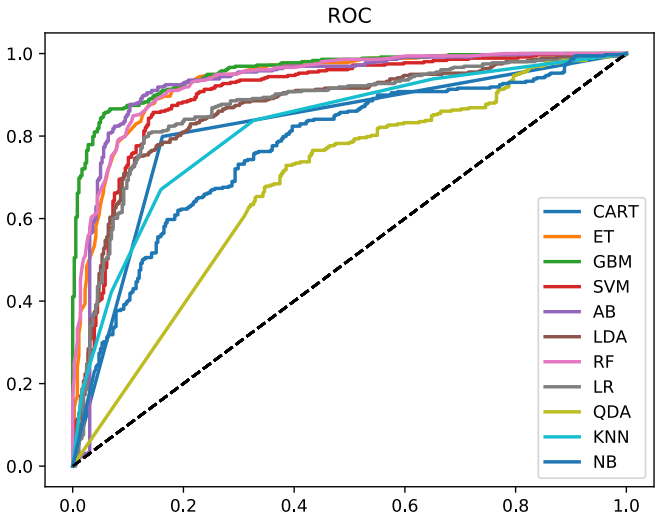


Figure 6: ROC curves with **Centroid** under-sampling.

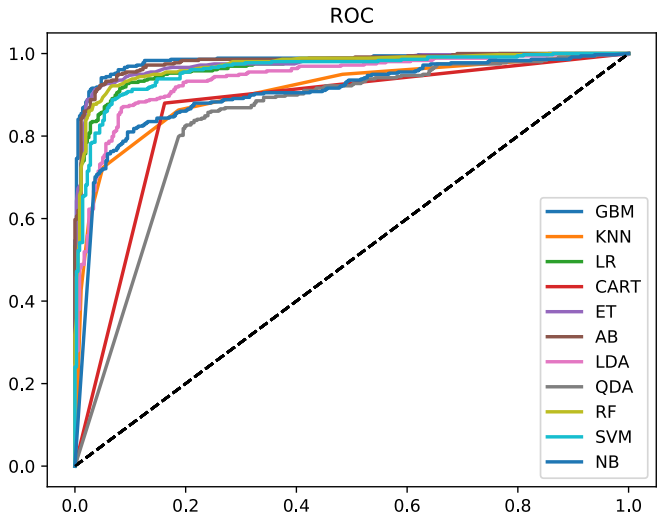


Figure 7: ROC curves with **NearMiss** under-sampling.

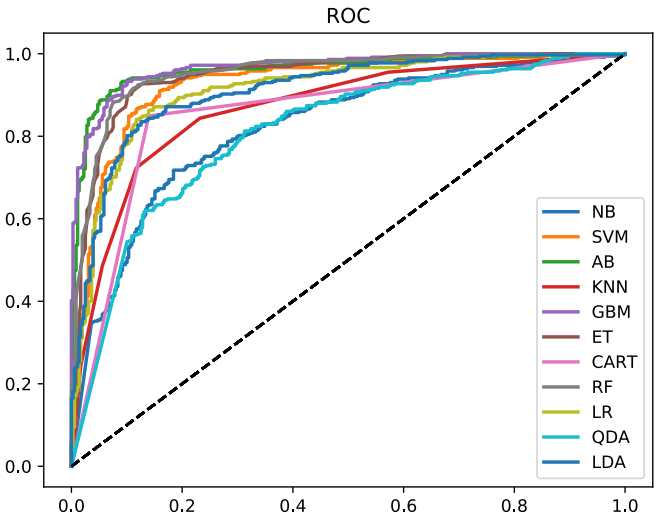


Figure 8: ROC curves with **Random** under-sampling.

Method	Centroid	NearMiss	Random
Logistic Regression	0.87	**0.97**	0.91
LDA	0.87	**0.94**	0.91
QDA	0.68	**0.84**	0.81
KNN	0.82	**0.91**	0.87
CART	0.82	**0.86**	0.85
Gaussian Naive Bayes	0.77	**0.90**	0.82
SVM	0.90	**0.96**	0.93
AdaBoost	0.93	**0.98**	0.96
GradientBoosting	0.96	**0.98**	0.97
Random Forest	0.94	**0.97**	0.95
ExtraTrees	0.93	**0.97**	0.95

Table 6: AUC score.

NearMiss still outperforms the other under-sampling methods when we compare AUC score. However, the best algorithms are not the same, here, the two boosting algorithms (AdaBoost and GradientBoosting) are the best whereas it was Extra-Trees when we compare using accuracy. Moreover, by selecting these features QDA manages to do better than LDA unlike when we were using all the quantitative variables. Actually it was because some of the variables were collinear and like it was stated by [Næs and Mevik, 2001] LDA and QDA can suffer from collinear variables.

2.3.3 Features Importance

The decision trees (CART, Random Forest and Extra-Trees) and boosting (Adaboost and Gradient-Boosting) algorithms provide a score for each feature representing its importance for the classification. These score can be useful to select most important variables.

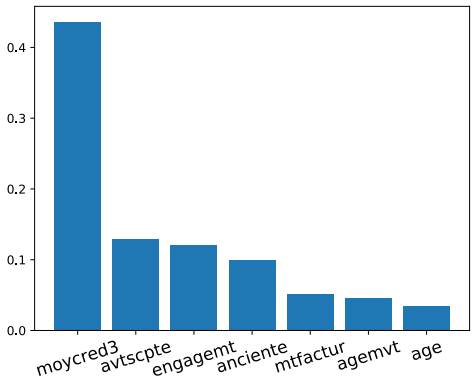


Figure 9: Features importance for **CART**.

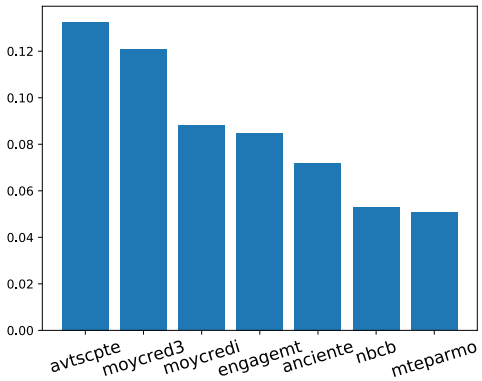


Figure 10: Features importance for **Random Forest**.

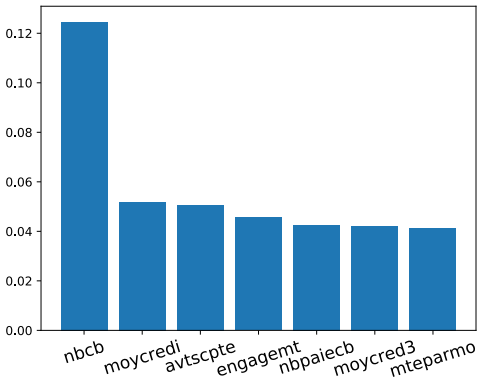


Figure 11: Features importance for **Extra-Trees**.

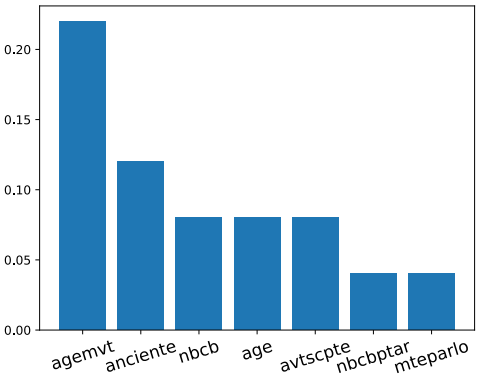


Figure 12: Features importance for **AdaBoost**.

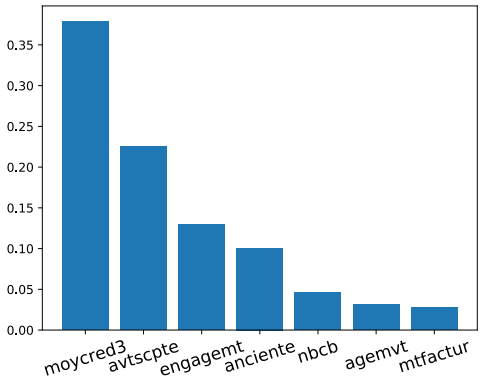


Figure 13: Features importance for **GradientBoosting**.

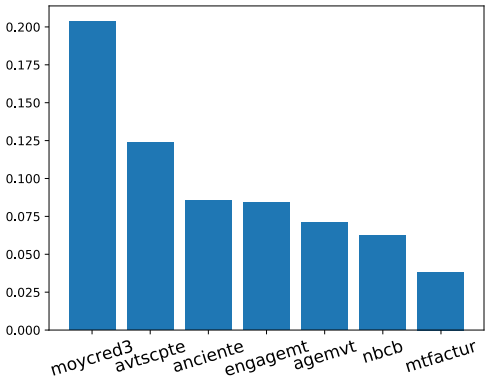


Figure 14: **Average** features importance.

Then we have decided to run again our experiments with a selection of variables by using only those having the most importance according to these algorithms. Since the NearMiss under-sampling gave the best results we have decided to use it in order to compare different sets of selected variables.

Method	Original	7 MIF	5 MIF	3 MIF
Logistic Regression	**0.97**	0.95	0.93	0.90
LDA	**0.94**	0.91	0.82	0.74
QDA	0.84	**0.94**	0.92	0.90
KNN	0.91	**0.94**	**0.94**	0.92
CART	0.86	**0.89**	0.87	0.79
Gaussian Naive Bayes	0.90	**0.94**	0.92	0.90
SVM	**0.96**	0.95	0.94	0.93
AdaBoost	**0.98**	0.97	0.96	0.93
GradientBoosting	**0.98**	**0.98**	0.97	0.92
Random Forest	0.97	**0.98**	0.97	0.92
ExtraTrees	0.97	**0.98**	**0.98**	0.92

Table 7: AUC score for NearMiss with different set of variables. MIF: Most Important Features.

"m MIF" represents the execution of the algorithm on the m most important features (on average) instead of all of them. It is interesting to see that on average 7 MIF does better than execution on all quantitative variables even if it removed 28 variables. It means that the other variables bring more noise than information relative to classification and therefore can be discarded for the prediction. Moreover, 5 MIF performs very well too even if it is not better than 7 MIF it is very close. The AUC score begin to decrease significantly with 3 MIF meaning that we have removed too many variables.

2.4 Credit Card Fraud

Having identified the profile most prone to acquiring the Visa Premier card, we now tackle another issue, equally tricky. Based on the Credit Card Fraud dataset, we will perform learn to classify potentially fraudulent credit card transactions.

	#samples	#features	#classes	Balance
Credit Card Fraud	284,807	30	2	0.00172

Table 8: Credit card fraud data set description.

As can be seen from the table above, this dataset is highly imbalanced, which makes harder for algorithms to learn and for us to evaluate results. For example, if we never classify one transaction as fraud, we would get 99,99% accuracy. In order to fix this we will use some under-sampling methods like it was previously discussed.

2.4.1 Preliminary Study

The dataset consists of 31 variables: Time, Amount, Class and V1...V28 (28 variables which are a result of a PCA transformation). What these variables stand for we are not able to know. Below we have graphic representations of the distribution of the known variables.

Pre-processing. The dataset contains no missing data. Moreover, the variables are already scaled. On the other hand, there are approximately 1.8k transactions with amount equal to zero. As these transactions are distributed between fraud and non-fraud classes, we have decided to keep them, because no information whether this is an error or not has been given. So no further data pre-processing is required.

Exploratory Analysis. The Amount of the transactions from both classes follow the same distribution (skewed to the left). Similarly, no conclusion can be taken from the Time distribution of each class.

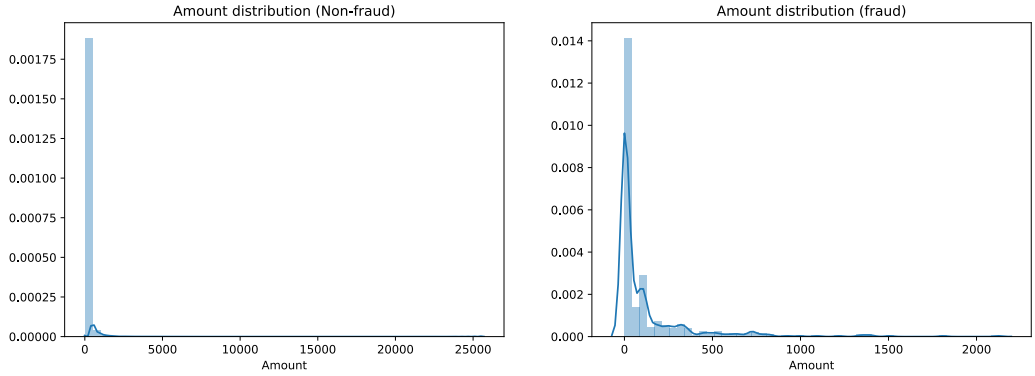


Figure 15: Distribution of Amount.

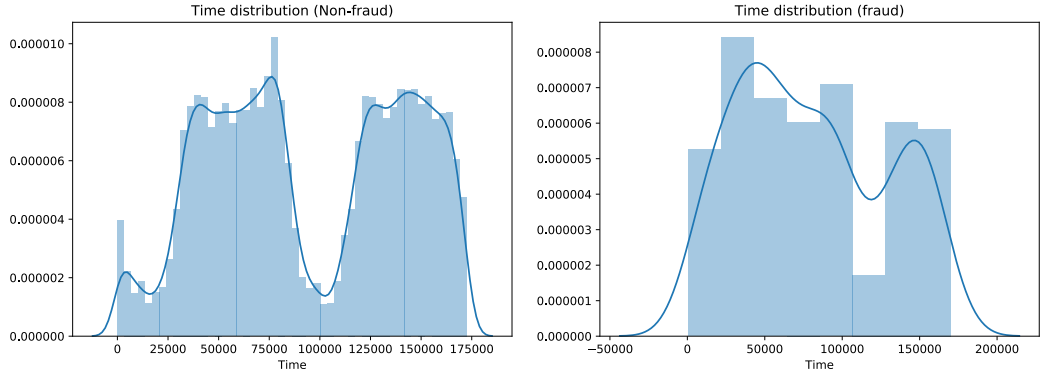


Figure 16: Distribution of Time.

The imbalance makes it much trickier to tackle this problem. The correlation matrix below shows no correlation among most of the variables. In the next section, we will bypass this imbalance issue by experimenting some sampling algorithms and exposing the hidden characteristics of the dataset.

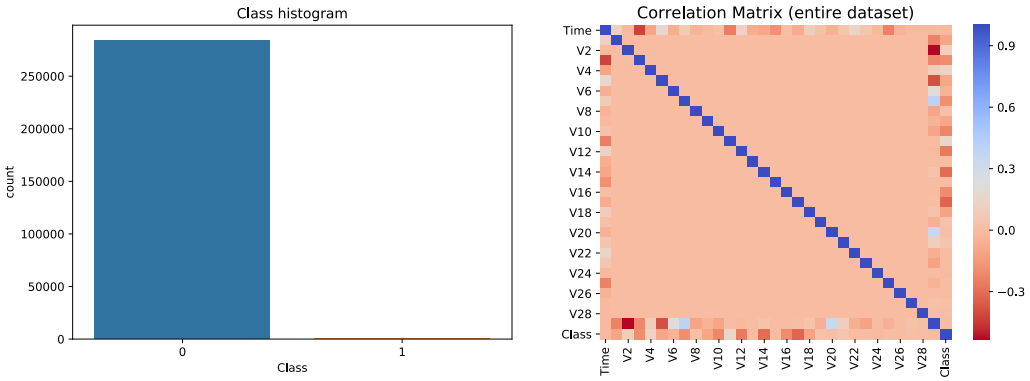


Figure 17: Distribution of Class and correlation matrix.

After the under-sampling, we can finally expose some characteristics of the dataset previously hidden by the imbalance. Now we can see how variables V1, V3, V5, V6, V7, V10, V12, V14, V16, V17 and V18 are negatively correlated to the fraud / non-fraud classification, while V2, V4 and V11 are positively correlated to it.

Nevertheless, plotting the two first principal components still does not give a visual idea of the classes.

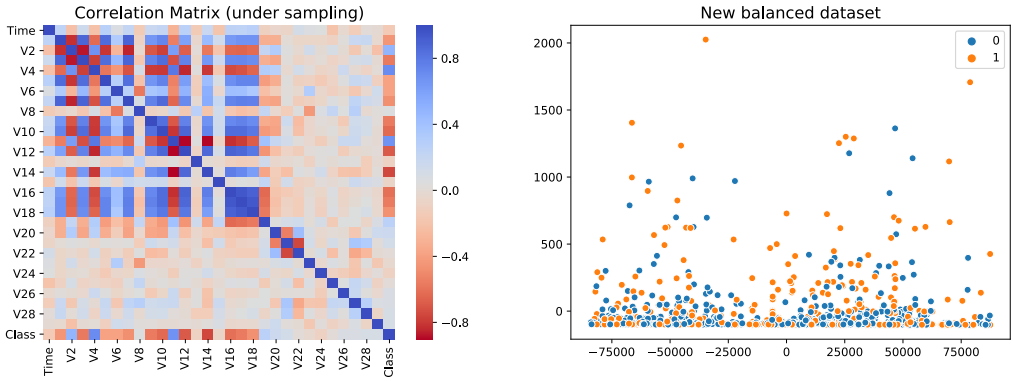


Figure 18: Distribution of Amount.

2.4.2 Classification

Following the same principal as in previous sections, we have implemented 11 supervised machine learning algorithms. Their accuracy for each sub-sampling is listed in the table below.

Method	Centroid	NearMiss	Random
Logistic Regression	**0.95** ± 0.04	0.94 ± 0.03	0.92 ± 0.01
LDA	**0.92** ± 0.05	0.89 ± 0.05	0.91 ± 0.01
QDA	**0.98** ± 0.01	0.92 ± 0.02	0.92 ± 0.02
KNN	0.30 ± 0.12	**0.80** ± 0.19	0.64 ± 0.05
CART	**0.95** ± 0.12	0.94 ± 0.03	0.90 ± 0.02
Gaussian Naive Bayes	**0.90** ± 0.05	**0.90** ± **0.06**	0.86 ± 0.02
SVM	0.50 ± 0.00	**0.81** ± 0.07	0.54 ± 0.01
AdaBoost	**0.99** ± 0.00	0.93 ± 0.09	0.92 ± 0.02
GradientBoosting	**0.99** ± 0.01	0.95 ± 0.02	0.93 ± 0.02
Random Forest	**0.99** ± 0.00	0.96 ± 0.02	0.94 ± 0.01
ExtraTrees	**0.99** ± 0.00	0.95 ± 0.02	0.94 ± 0.01

Table 9: Mean accuracy and standard deviation with k-fold cross-validation ($k = 10$) for three types of under sampling methods.

The centroid under-samplig method seems to have increased separability between the two classes. The NearMiss approach, however, performed well and with more stability (as for KNN and SVM). Moreover, this approach has the benefit to keep observations closer to the limit between the two classes, which in a way guarantee better results for new data.

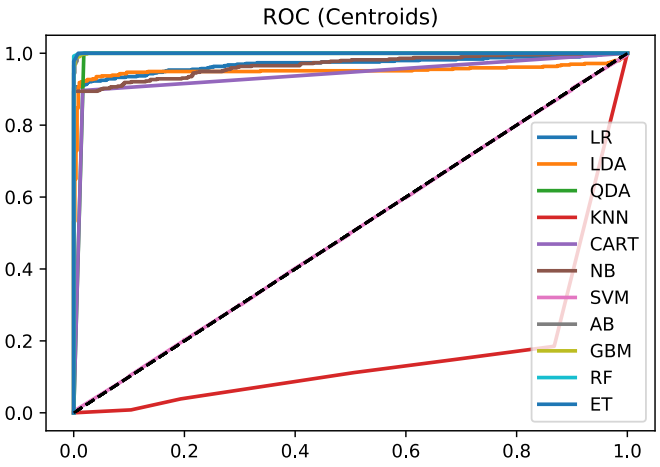


Figure 19: ROC curves with **Centroid** under-sampling.

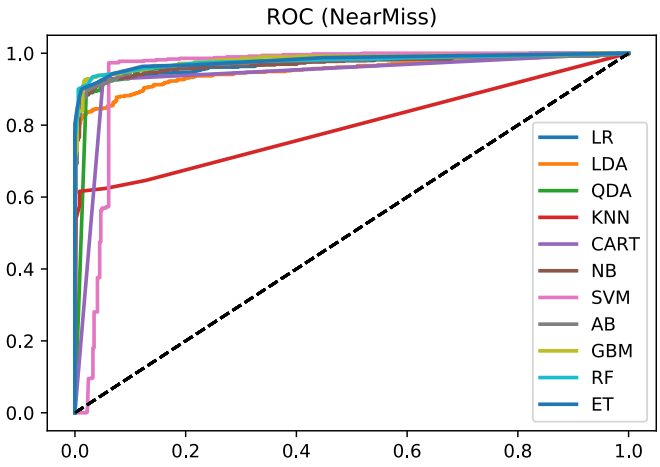


Figure 20: ROC curves with **NearMiss** under-sampling.

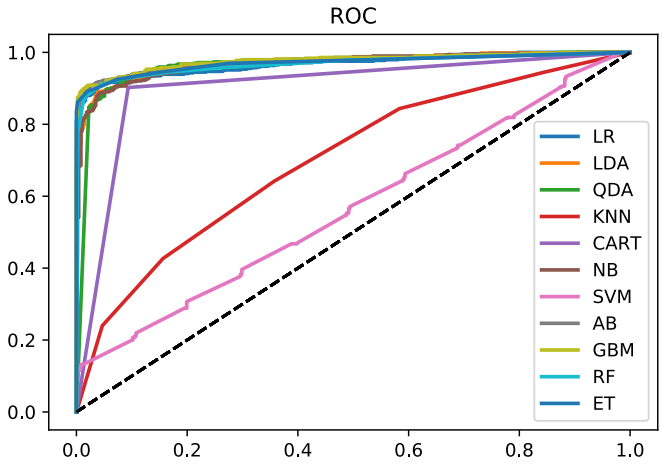


Figure 21: ROC curves with **Random** under-sampling.

Method	Centroid	NearMiss	Random
Logistic Regression	**0.97**	**0.97**	**0.97**
LDA	0.95	0.95	**0.97**
QDA	**0.99**	0.97	0.96
KNN	0.16	**0.80**	0.70
CART	**0.94**	**0.94**	0.90
Gaussian Naive Bayes	**0.97**	**0.97**	0.96
SVM	0.50	**0.95**	0.56
AdaBoost	**0.99**	0.97	0.97
GradientBoosting	**0.99**	0.98	0.98
Random Forest	**0.99**	0.98	0.97
ExtraTrees	**0.99**	0.98	0.97

Table 10: AUC score.

The AUC score gives much better results for NearMiss unlike when we compare with the accuracy even if the best results still belong to Centroid. The same comment can be made with Random, it manages to be the best for some algorithms like Logistic Regression or LDA.

2.4.3 Features Importance

As stated before for the Visa Premier data set, we can have the importance of features according to the boosting and decision trees algorithms.

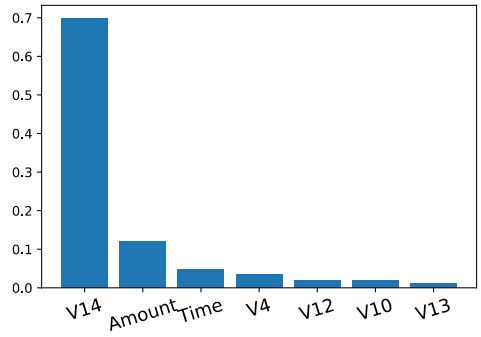


Figure 22: Features importance for **CART**.

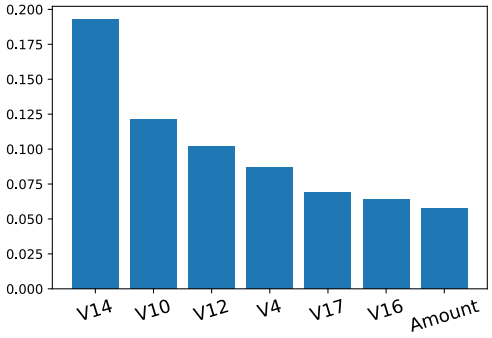


Figure 23: Features importance for **Random Forest**.

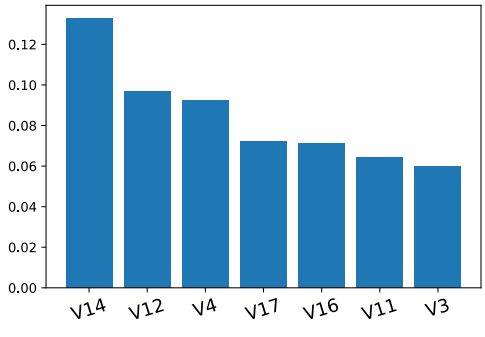


Figure 24: Features importance for **Extra-Trees**.

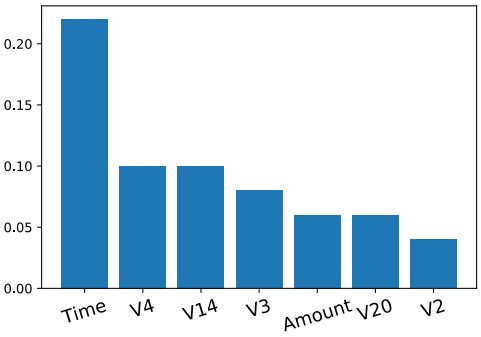


Figure 25: Features importance for **AdaBoost**.

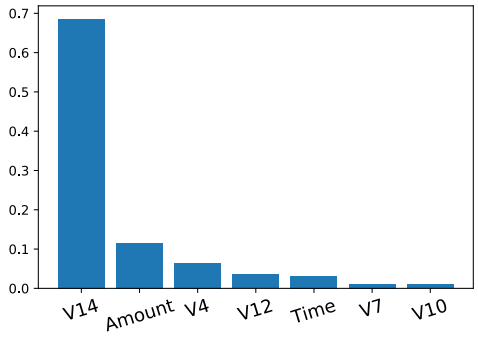


Figure 26: Features importance for **GradientBoosting**.

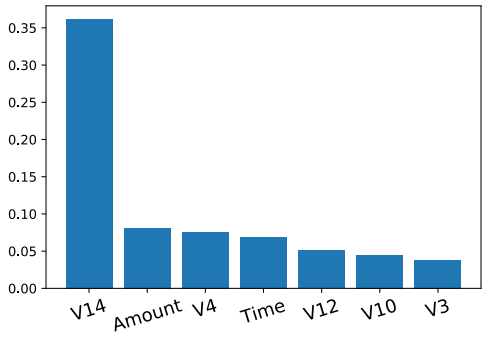


Figure 27: **Average** features importance.

The V14 variable seems to be very important for most of the methods. Based on the results above, we have decided to re-run the experiment selecting only the most important variables and comparing the result with the best under-sampling result, in this case Centroid under-sampling.

Method	**Original**	**7 MIF**	**5 MIF**	**3 MIF**
Logistic Regression	0.95	**0.97**	**0.97**	**0.97**
LDA	0.92	**0.96**	**0.96**	**0.96**
QDA	0.98	**0.99**	**0.99**	**0.99**
KNN	**0.30**	0.16	0.16	0.16
CART	0.95	**0.96**	0.95	0.95
Gaussian Naive Bayes	0.90	**0.97**	**0.97**	**0.97**
SVM	**0.50**	0.36	0.41	0.35
AdaBoost	**0.99**	**0.99**	**0.99**	**0.99**
GradientBoosting	**0.99**	**0.99**	**0.99**	**0.99**
Random Forest	**0.99**	**0.99**	**0.99**	**0.99**
ExtraTrees	0.99	**1.00**	**1.00**	**1.00**

Table 11: AUC score for Centroid with different set of variables. MIF: Most Important Features.

Like for Visa Premier, the selection of variables by taking the most important ones globally increases the results and manages to has a perfect AUC score with ExtraTrees.

3 Conclusion

We saw that decisions trees and boosting algorithms can provide good results but above all they can score the importance of a feature for the classification. This score is very helpful, especially when we have a lot of variables to know who are the most important. Moreover it can help us to select some of them and can even increase classification result as we saw.

Moreover under-sampling methods are crucial to deal with imbalanced data set and have meaningful result such as a non-biased accuracy, AUC score is a good indicator too.

As future work we could try other method to deal with imbalanced data set such as over-sampling or methods combining both kinds of sampling as SMOTE or ROSE. Moreover, it could have been interesting to try to run algorithms on Credit Card Fraud with only the V14 variable.

Finally, our github repository is open source and free to use[1].

References

[Chawla et al., 2002] Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357.

[James et al., 2014] James, G., Witten, D., Hastie, T., and Tibshirani, R. (2014). *An Introduction to Statistical Learning: With Applications in R.* Springer Publishing Company, Incorporated.

[Lunardon et al., 2014] Lunardon, N., Menardi, G., and Torelli, N. (2014). Rose: a package for binary imbalanced learning. *R Journal*, 6:79–89.

[Næs and Mevik, 2001] Næs, T. and Mevik, B.-H. (2001). Understanding the collinearity problem in regression and discriminant analysis. *Journal of Chemometrics*, 15:413 – 426.

[1] https://github.com/MatPont/Supervised_Project