# Credit Defaults: Naïve Bayes and Random Forest Classifiers

**Alberto Ruiz Benítez de Lugo Hernández & Rémi Ounadjela – City, University of London**

## Motivation and Description

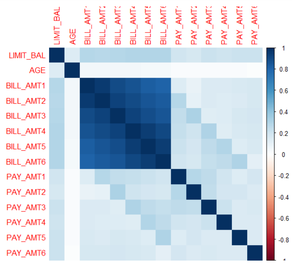
This project aims to predict whether a customers will default on their bank credit card or not using two classification techniques, Naïve Bayes (NB) and Random Forest (RF).
The motivation for this selection is the methodological difference between these two models, probabilistic method against decision trees, respectively. This data was obtained from UCI Machine learning repository [1]. Whilst the overall accuracy of both models is important, here the objective will be to predict which clients default on their credit card (True Negatives in the confusion matrix).

## Initial Analysis of the Data including Basic Statistics

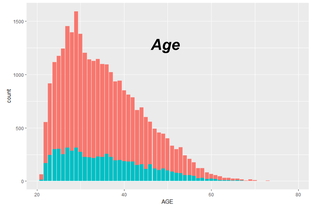
**Original dataset**
- 24 different predictors and 30,000 observations

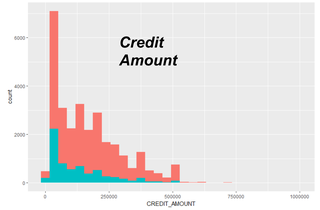
**Processed dataset**
- 7 different predictors and 29,601 observations

- The **correlation matrix** showed essential insight, there is high correlation between many predictors. As a result, we decided to combine these predictors into a single variable, *"**Pay on Time**"*, in order to enhance independency among variables, simplify complexity and avoid overfitting [4].
- Some predictors were deleted because they duplicated financial information.
- Some errors were discovered, these represents 1.33% of our data, so we deleted them without fear of losing information.
- This data has a binary response:
  *Default credit card* (**1**) or *Paid credit card* (**0**).
- *Processed data predictors:*
  *5* ***categorical*** *predictors, 2* ***continuous*** (Age, Credit Amount).
- *"Age"* shows a clear normal Gaussian distribution with a positive skew. Whereas, *"Credit Amount"* needs optimization to define a good density distribution.




| Measure | Credit Amount (NT Dollars) | Age |
|---|---|---|
| Mean | 167551 | 35.5 |
| Median | 140000 | 34 |
| Mode | 50000 | 29 |
| Minimum | 10000 | 21.00 |
| Maximum | 1000000 | 79.00 |
| Std. deviation | 129747.7 | 9.2 |




## Hypothesis

We expect, according to theory, Random Forest (RF) to perform better than Naïve Bayes (NB). To evaluate this assumption, we have pre-processed the data to boost the independence between predictors (NB), in order to evaluate whether this assumption is true or not in fair conditions.
Furthermore, our data could be considered as a large dataset (>20.000 observations). Thus, according to the nature of Random Forest, low bias and high variance, it should be perform better than Naïve Bayes on this dataset.

## References


1. https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients#
2. Leo Brieman, 'Random Forests', Machine Learning 45, 2 – 6
3. Deepanshu Bhalla. 2014. *RANDOM FOREST EXPLAINED IN SIMPLE TERMS*. [ONLINE] Available at: http://www.listendata.com/2014/11/random-forest-with-r.html. [Accessed 1 December 2016].
4. *D. Hand and K. Yu, "Idiot Bayes, not so stupid after all", International Statistical Review, Vol. 69, No. 3. (2001), pp. 385-398).*
5. *Eibe Frank, Leonard E. Trigg, Geoffrey Holmes, and Ian H. Witten. Naive Bayes for regression (technical note). Machine Learning, 41(1):5-25, 2000)*
6. Bengio, Y., & Grandvalet, Y. (2005). Bias in estimating the variance of k-fold cross-validation. Statistical modeling and analysis for complex data problems, 75–95.
7. Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection.

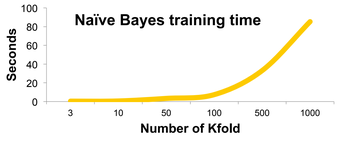

| Algorithm | Naïve-Bayes(NB) | Random Forest (RF) |
|---|---|---|
| Summary | NB sets a parametric probabilistic model based on Bayes Theorem, assuming independence between predictors. It is a ***high bias*** and ***low variance*** classifier. (Only needs few observations to provide good results). | RF, based on decision trees, is made of a sample of training data, and a number of random predictors. Then, the trees are combine to classify. It is a ***low bias*** and ***high variance*** classifier. |
| Pros | • Naïve Bayes is a ***probabilistic*** *model*, thus is easy to understand (*high interpretability*) and easy to train.<br>• **Very quick training** and **low computational cost**.<br>• The **simplicity** of the model usually brings better results than other complex solutions, as *Hand (2001) says "...based on the typically false assumption that the predictor variables are independent, can be highly effective, and often more effective than sophisticated rules". [4]* | • **Non-probabilistic**. It deals well with overfitting due to absence of discretionary decisions and random nature of each individual tree [2].<br>• Provides a summary of the most/least useful predictor to the classification (or regression) problem.<br>• **Low bias**, thus its performance is better with large datasets, provide better results under this circumstance. |
| Cons | • Good results, but **limited** by simplicity, RF is better.<br>• We lose the ability to exploit the interactions between features due to independent assumption [5].<br>• **Zero frequency issue**. If we get zero observations in our dataset for a particular class, the frequency-based probability will be zero and this will affect the posterior probability. ***Laplacian correction*** needed. | • **High variance**. It needs a large datasets to provide good results and has **low interpretability** of them.<br>• High **computational cost.** Also, **training** and **execution time** are usually slow.<br>• RF cannot handle noise very well.<br>• It needs methods to reduce the variance, such as bagging. |

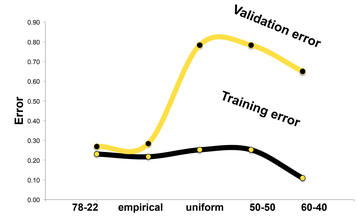
## Description of Training and Evaluation Methodology

**Test (20%)** | **Training (80%)**

### *Naïve Bayes – Kfold (Cross Validation)*

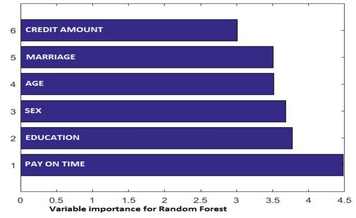
- 20% test set – **80% training set (23,680 obs).**
- **Cross validation** (*kfold*) was applied due to the probabilistic nature of Naïve Bayes. The bins obtained from this method provide a generalized training error.
- Predicted responses were calculated using the **minimum training error** obtained from cross validation. In order to use the maximum likelihood possible while predicting.
- Different ***probability distributions*** were applied to evaluate each predictor result. The best performance was obtained by graphical analysis and optimization (fitcnb).
- Different ***priors*** were applied to evaluate results, the best output comes from the *class probability distribution* observed in the dataset (78% paid – 22% default). Reducing the difference between training error and validation error.
- ***The number of kfold*** applied increase the execution time while training the model. The errors obtained remain constant no matter the number of kfolds. Then, a regular cross validation (kfold = 10) was selected.

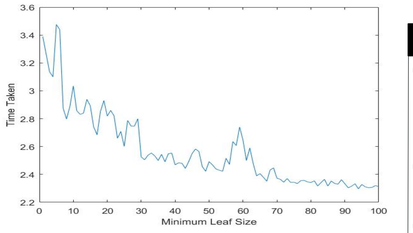



### *Random Forest – Bagging (Bootstrapping)*

- 20% test set – **80% training set (23,680 obs)**
- Every decision tree suffers ***high variance*** due to its nature [7]. The Bootstrap Aggregation, or **Bagging**, was applied to reduce this variance while training.
- The errors are evaluated using Out-of-Bag error (OOB) – a measure of misclassification
- Bagging provides a **summary of the importance** of each predictor. In our case, the most important predictor is ***"Pay on Time"***, the self-made attribute created to reduce complexity and correlation.
- The **execution time** decreases while *minimum leaf size* increases. Because a higher leaf is equal to smaller trees, which take less time to compute.
- Increases in the *number of trees and/or random predictors* are directly related to rises in **computational cost** and **execution time**.




## Lessons Learned and Future Work

**Lesson learned:**
- Naïve Bayes depends on discrete decisions while selecting probability distributions and data processing. Whereas, Random Forest performance depends heavily on hyperparameter optimization.
- The bias and variance set the main difference between these two models. If the dataset was smaller, the NB could reach better results, because the RF requires more observations to be trained, while NB require less amount of observations.

**Future work:**
- A future approach could be the **normalization of all predictors to categorical attributes** during the data processing stage. This way could reduce even more the complexity of the model and increase results while using Naïve Bayes.
- Regarding Random Forest, **more parameters could be optimised**. For instance, the percentage of data used during bagging.

| ***Model*** | ***Precision*** | ***Recall*** |
|---|---|---|
| **Naïve Bayes** (without Cost Matrix) | 0% | 0% |
| **Naïve Bayes** (using Cost Matrix) | 30% | 21% |
| **Random Forest** (without Cost Matrix) | 47% | 2,7% |
| **Random Forest** (using Cost Matrix) | 35% | 34% |

## Choice of Parameters and Experimental Results

### *Naïve Bayes (parameters)*

- **Probability Distributions**

Our predictors have different probability distributions
  - ***Multivariant multinomial distribution (mvmn)*** for *categorical predictors (aditive smoothing included in this distribution,* ***Laplacian correction****)*
  - ***Normal Gaussian distributions*** for continuous predictor "**Age**", *obtained by graphical analysis (histogram)*
  - ***Kernel distribution (box, width =15695)*** for continuous predictor "**Credit Amount**", obtained by hyperparameter optimization (fitcnb).

- **Prior**

Prior probability distribution = [0.78, 0.22]
Equal to 78% (paid class) and 22% (default class) by calculating the class probabilities from the total observations given.

- **Cost matrix: (both models)**

*Our results were unbalanced*. The accuracy of the models were good, but after a deeper analysis using the confusion matrix, we discovered that our system had bad results due to **misclassification** by unbalanced data. Thus a *Cost Matrix* was applied to avoid this bad performance while predicting.

Cost Matrix = $\begin{pmatrix} 0 & 1{,}5 \\ 3{,}5 & 0 \end{pmatrix}$, according to the best experimental result.

*Without Cost matrix*

**Naïve Bayes – Confusion Matrix**

| | Predicted paid (0) | Predicted default (1) |
|---|---|---|
| Actual paid (0) | 4662 | 0 |
| Actual default (1) | 1259 | 0 **Objective** |
| **Model Accuracy Rate** | **78.71 %** | |

*While using Cost matrix*

**Naïve Bayes – Confusion Matrix**

| | Predicted paid (0) | Predicted default (1) |
|---|---|---|
| Actual paid (0) | 4041 | 621 |
| Actual default (1) | 993 | 266 **Objective** |
| **Model Accuracy Rate** | **72.87 %** | |

### *Random Forest (parameters)*

- **Number of Grown Trees**

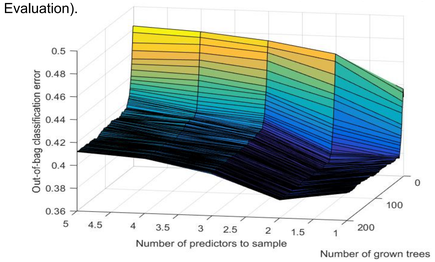
Higher number of trees may increase computational cost and execution time. After a graphical analysis, **50 trees minimize the error (OBB) with minimum computational cost**. More than 50 trees keep the error constant and increase computational cost and time.

- **Number of Predictors to Sample**

The number of random predictors used on each tree can also be optimised. After a graphical analysis, it is clear that selecting **2 predictors** provides the minimum amount of error.

- **Minimum Leaf Size**

Small leaf size requires more computational power and increase noise. Generally a larger amount of short trees (high leaf) is the best approach. However, In our case best the leaf size is equal to 1 (see Analysis and Evaluation).




*Without Cost matrix*

**Random Forest – Confusion Matrix**

| | Predicted paid (0) | Predicted default (1) |
|---|---|---|
| Actual paid (0) | 4624 | 38 |
| Actual default (1) | 1225 | 34 **Objective** |
| **Model Accuracy Rate** | **78.60%** | |

*While using Cost matrix*

**Random Forest – Confusion Matrix**

| | Predicted paid (0) | Predicted default (1) |
|---|---|---|
| Actual paid (0) | 3886 | 776 |
| Actual default (1) | 833 | 426 **Objective** |
| **Model Accuracy Rate** | **72.57%** | |

### *Experimental Results:*

- Both models reduce their accuracy rate with the **Cost Matrix** implemented. However, the ***quality of the predictions*** *(objective)* increases in both cases, i.e. less overfitting.
- The number of trees, number of random predictors and minimum leaf size, obtained in Random Forest by optimization present good results and **low computational cost.**
- The **prior** assumption in Naïve Bayes increases the accuracy.
- Kernel distribution applied on "Credit Amount" presents better results than a normal gaussian distribution.

## Analysis and Evaluation

- The chosen **minimum leaf size** in Random Forest is 1 (deep tree), as increasing minimum leaf did not lead to any improvement in the given experimental results (see Matlab code). Although, the *graph on the left* shows, as expected, that the **computational time** decreases significantly when a higher minimum leaf size is applied. Based on the obtained results, our analysis is that our model has a few predictors (just 7), thus a deep tree is not so costly as expected according to theory, *Kohavi (1995)* [7]. Furthermore, our RF only takes few minutes to compute with a minimum leaf size of 1, thus there is no need in our specific case to improve computational time.
- Both models reach more **quality** results while the **Cost Matrix** is applied, but obtain a smaller Accuracy Rate. The **Accuracy Rate** only do not secure good performance, we need to focus on further indicator such as ***Precision*** and ***Recall***, as well as the number (nominal) of well predicted observations, to select a model.
- The main objective in this project is estimate the ***credit card default (1)***. As said, without the Cost Matrix our models reach higher accuracy levels, ***but with low quality,*** *because our objective is to maximize* ***True Negatives***. This is very clear in NB (*confusion matrix*), where the model predict zero default cases due to unbalanced data. Thus, cost matrix is needed to correct this unbalance and predict well.
- Independently whether Cost Matrix is applied or not, *Random Forest* always performs better than *Naïve Bayes* regarding our objective (True Negatives, *default cases (1)*), providing more well predicted observations. ***Thus, we can confirm our initial hypothesis.*** This can be explained by the low bias of RF, which provides strong results in large datasets while *bagging* reduces its variance.
- Regarding ***zero frequency issues*** in Naïve Bayes, we focus on pre-processing data (missed values) to avoid this behavior. Moreover, the probability distribution chosen for categorical predictors is *Multivariate Multinomial (mvmn),* which includes a ***additive smoothing*** *(Laplace correction)* to prevent bad results in the posterior.
- The Naïve Bayes model works better, in theory, using ***categorical predictors*** *(simplify* complexity*)* and ***low correlation*** attributes, *Bengio (2005)* [6]. Then, *the* ***"Pay on Time"*** predictor we created improves the performance of the NB Algorithm. Because this predictor reduces correlation in the model, that otherwise could lead to increased complexity, as correlated attributes could increase their own importance by duplicating results. This variable is also the most importance in RF, what evidences a good data pre-processing method.