Credit Card Default Prediction

Anagha Patil[1], Rani Adhaduk[2], Vidhi Mistry[3]

School of Electrical Engineering and Computer Science,
University of Ottawa, Ottawa, Ontario, Canada
apati009@uottawa.ca[1]
radha067@uottawa.ca[2]
Vmist063@uottawa.ca[3]

Abstract. Credit Card fraud is the most pressing issue which has to be dealt with by card issuing companies. This problem has been around for decades. This paper focuses on the frauds taking place at the application level using the feature selection methods and re-sampling like oversampling techniques. It uses various ML techniques such as Dummy Classifier, Decision Tree, Logistic Regression, K-Neighbors, Random Forest, XGBoost, LightGBM, Neural Networks and Gaussian Naive Bayes to detect any financial frauds. The performance of each of the methods is compared based on four different parameters: Recall, F1-Score, Accuracy, and Precision. A dataset of credit cards is used to evaluate the efficiency of the different ML techniques, purely based on the filter features selection method. The outcomes of this experiment show that the accuracy and other parameters of XGBoost are the highest. But when we performed the Nemenyi test we found that Random Forest has the least average rank and hence we concluded that Random Forest performs best at all times.

Keywords: Credit card, Machine Learning techniques, Classification, Sampling, Nemenyi Test

1 Introduction

Financial information like customer transactions, financial statements, bill amount, repayment records, etc., is used for risk prediction to anticipate business performance and credit risk of individual customers. This can help reduce the uncertainty and damage caused by potential fraud. Granting or rejecting a loan of a customer is the most important decision any financial institutions have to make. This choice essentially comes down to a twofold classification issue which targets recognizing good payers from bad payers. The distinction was done by inspecting an applicant's form details by a judgemental approach. Considering the applicant's all possible relevant information, economic situations, and socio-demographic status, credit experts then decides his/her creditworthiness. Repayment behavior of credit applicants and all the related characteristic information can be stored electronically at financial institutions. This has propelled the need for using machine learning or statistical algorithms to automate credit-granting decisions.

2 Design

This project follows the common machine learning life-cycle. Refer 1. First of all, data is investigated with an aim to spot trends and patterns in the data which makes up the Exploratory Data Analysis step in the project workflow. Later, data pre-processing tasks are performed and the data gets normalized. It will be discussed later in the paper about the problem of class imbalance in the dataset which is solved by sampling. Thereafter, a subset of features is selected using feature selection techniques and this selected data is then fed to various classification models to perform classification. In the end, the results obtained from all the classifiers are compared using statistical tests to find the best classifier among all.

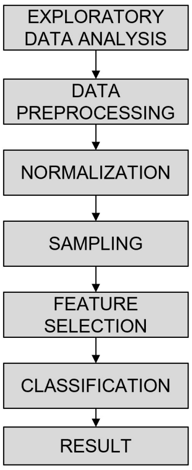


Fig. 1: Project Workflow

2.1 Technologies Used

The language of choice used for the project was Python. Scikit-learn library in python was used to create all the machine learning models. Moreover, matplotlib library was used to plot the graphs. Pandas which is an open-source library was used to perform data analysis in this project.

3 Implementation

3.1 Dataset

The dataset is obtained from the UCI Machine Learning repository, where it's listed under the name of 'default of credit card clients'. The data is of clients of a bank in Taiwan, precisely credit cardholders of that bank, collected over a 6 month period - April 2005 to September 2005. The dataset contains the details of the credit card clients in the form of 24 features. The features reflect the payment pattern and history of the cardholders. Among these 24 features, the output label is the class which takes two values 0 and 1, the values implying non-default and default respectively. Thus, the task at hand is a binary classification task: default or non-default. In total, there are 30,000 data samples in the dataset. The features in the dataset are : `limit-bal`,`education`, `marriage`, `sex`, `age`, `class`, `pay-1`, `pay-2`, `pay-3`, `pay-4`, `pay-5`, `pay-6`, `bill-amt1`, `bill-amt2`, `bill-amt3`, `bill-amt4`, `bill-amt5`, `bill-amt6`, `pay-amt1`, `pay-amt2`, `pay-amt3`, `pay-amt4`, `pay-amt5`, `pay-amt6`.

Here `class` is the output label. The `limit-bal` is the credit card limit of the customers. The features `education`, `marriage`, `sex`, and `age` are self-explanatory.The `pay-n` features are the status of the payment whether it was paid duly or there was some delay in the payment for the six months - April to September. The `bill-amtn` features is the amount of bill for the n^{th} month whereas the `pay-amtn` is the amount paid by the customer for that particular n^{th} month.

3.2 Data Exploratory Analysis

We begin with the exploratory analysis of the data with an aim to find interesting patterns in the data and understand the dataset better.

Null Values : No null values were found in the dataset.

Class Imbalance : The output class can take up value 0 signifying ‘non-default’ or 1 signifying ‘default’. Majority of the data instances belonged to the non-default class. See figure 2. Precisely, there were 23364 samples for the non-default category whereas only 6636 default instances. Thus, there exists class imbalance in the dataset.

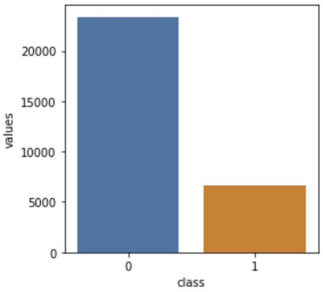


Fig. 2: Class Imbalance

Anomalies and Outliers : Since the dataset had many features that represented payment information. We checked for anomalies in the dataset. We checked for all the payments that had transactions greater than 400,000. Upon closer inspection, it was concluded that those transactions were genuine and all those clients' payment history included similar high amount transactions with high credit limits.

Correlation : The correlation among the features was also checked using Pearson's Correlation coefficient. The correlation matrix was plotted and is shown in 3. From the correlation matrix, we can infer that there is a correlation between the pay-n and bill-amt features, and also among the bill-amt features. Moreover, the class label is also shown to be correlated with the pay-n features, indicating they must offer insights while predicting the class labels.

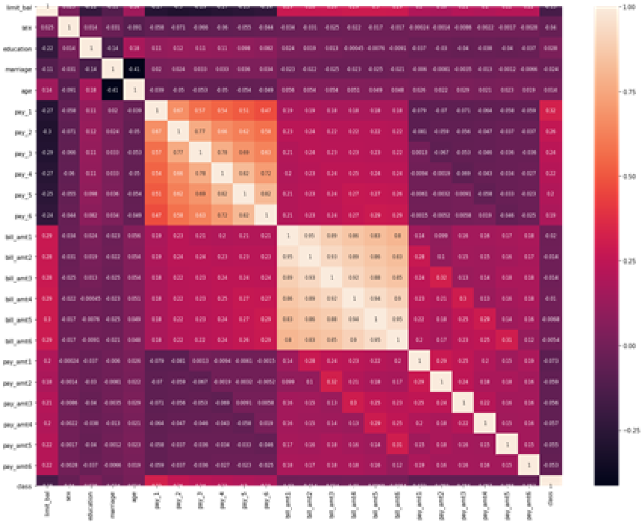


Fig. 3: Correlation Matrix

Client Demographics: The demographics of clients were inspected with an aim to spot trends among them. Moreover, the demographics were also looked into to check for any inconsistencies among them. And, some inconsistencies were detected for two of the features. The user demographics such as age, education, and marriage were probed.

Gender : There were two values for this feature: male and female. The count by value bar chart for age is shown in figure 4a. The figure (right-side) also shows charts for the number of defaults per each gender type per age. It's apparent from the charts that the bank had more female clients and eventually more defaults with female clients.

Education : This feature took 7 values from 0 to 7 each representing a level of education of the client. However, the values 0, 5, and 6 were not defined by the source of the data. It was unclear what these values represented. The count by value bar chart for education is shown in figure 4b. The figure (right-side) also shows charts for a number of defaults per each education type per age. It's apparent from the charts that the bank had most clients that had a university degree.

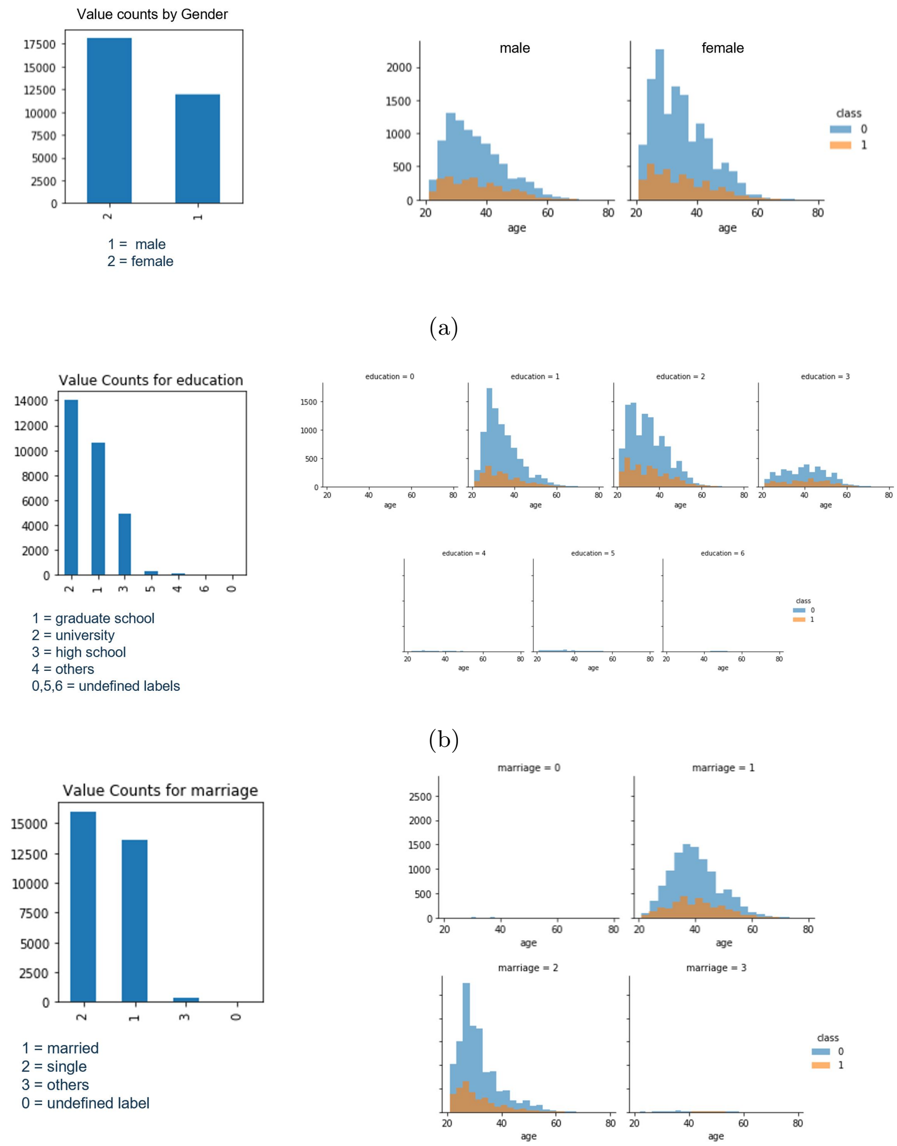
Marriage : This feature took 4 values from 0 to 3 each representing the marital status of the client. However, the value 0 was not defined by the source of the data. It was unclear what marital status this value represented. The count by value bar chart for marriage is shown in figure 4c. The figure (right-side) also shows charts for a number of defaults per marital status per age. It's apparent from the charts that the bank had around the same number of married and single clients.

3.3 Data Pre-processing

Noisy Data: Upon exploratory data analysis, issues with the data were encountered. There were some features that had undefined labels, in the sense there was no information put by the source regarding what those labels meant. These noise in the data was handled with the following steps: For `education` feature the labels 5, 6, and 0 were not defined. We labeled all these samples as 4 i.e in *'others'* category. For `marriage` feature the label 0 wasn't defined. So, We labeled all these samples as 2 i.e in *'others'* category. The `pay-n` had undefined labels : -2 and 0. Since -1 meant paid duly and 1 meant a delay of a month, 2 meant a delay of 2 months, and so on. The samples with labels -1, 2, and 0 were relabelled as 0 indicating paid duly.

Categorical Features: The data contains categorical features - `education`, `marriage` and `sex`. These categorical features were converted into a binary set of features using one hot encoding technique. This technique converts the categorical feature into an integer encoded feature. Thereafter, a binary feature is generated for each value of this integer encoded feature. After one hot encoding the categorical features, the dataset had 30 features in total including the output class label. This dataset of 30 features was used for further classification tasks. Thus, after one hot encoding, the dataset contained the following features: `limit-bal`, `age`, `pay-1`, `pay-2`, `pay-3`, `pay-4`, `pay-5`, `pay6`, `bill-amt1`, `bill-amt2`, `bill-amt3`, `bill-amt4`, `bill-amt5`, `bill-amt6`, `pay-amt1`, `pay-amt2`, `pay-amt3`, `pay-amt4`, `pay-amt5`, `pay-amt6`, `class`, `grad-school`, `university`, `high-school`, `edu-others`, `married`, `single`, `mrg-others`, `male`, `female`.

Normalization: Since all the features had a different range, they had to be scaled and brought into a common range. Normalization technique was used



(a)

(b)

(c)

Fig. 4: Charts for user demographics : (a) Gender, (b) Education, and (c) Marriage

here. The goal of normalization is to change the values of numeric columns in the dataset to a common scale, without distorting differences in the ranges of values. Here, for this dataset, all the features were normalized and brought in the range [0,1].

3.4 Sampling

The ML algorithms have a high tendency to create classifiers that are faulty in nature. This is because these classifiers are trained with datasets that aren't balanced.As a result, the algorithms give biased output in the favour of majority class while treating the minority class as noise.This is called Class Imbalance problem. There are various ways of handling the imbalance of datasets, but in this project, oversampling is used.

3.4.1 Oversampling

SMOTE or Synthetic Minority Oversampling technique [4], is a method of oversampling which is used while resolving the problem posed by the imbalance of datasets. SMOTE is a widely accepted technique, and is used to remove the imbalance by increasing the examples of minor classes by replicating the already existing examples.

3.5 Feature Extraction

We come across hundreds of thousands of features, and keeping a note of the important ones is a complex task. Creating a model for this has three benefits: it will be simple to interpret, the variance will be reduced and as a result overfitting as well, and the time and cost for computation also decrease. Process of selecting the most important features is known as feature selection.

3.5.1 Random Forest

It is a common method used to select feature in a workflow pertaining to data science, the reason behind is the tree based strategies, the fact that how well do they improve ranking of the purity of the node. A tree is built on the value of Impurity. Impurity is high at the root of the tree and decreases as we go down. Finally the tree is pruned at a decided node and the subset of most important nodes is created[1]. We applied Random Forest method for feature selection and resulted in 20 imporatant features. Removed features are sex, education, marriage and age. This helped with decreasing the complexity of computations and the training time.

3.6 Classification

This paper aims to compare the working and output of various classifiers like Dummy Classifier, Decision Tree, Logistic Regression, K-Neighbors, Random

Forest, XGBoost, LightGBM, Neural Networks and Gaussian Naive Bayes and obtain the classifier which suits our dataset the best. Extracted features are fed into classifiers to classify the dataset and finally sort, if the instances will pay credit or not.

3.6.1 Dummy Classifier

This classifier does not provide any insights regarding the data, but is able to classify them using a set of simple rules. The behaviour of this classifier is completely independent of any kind of training data. This is the fact that the data set are completely ignored, and in their place, one of the strategies is used for predicting the label of the class. This method is only used as a standard for any other classifier. It forms an essential benefit method for those cases where there is a surety of Imbalance of class.

3.6.2 Decision Tree

It is a method to classify the data. This method generates a tree, after which a set of rules are laid out from a particular dataset. The tree is a representation of the model which contains different classes pertaining to the data. While the internal nodes signify a test on a particular attribute, the leaf denotes the class distribution. On the other hand, a branch denotes the output or the result of the test. The usage of this methods lies in its function of dividing various large sets into smaller ones. This is done by using a sequence of simple decision rules. It is one of the methods of classifying the sequential decision problems [2].

3.6.3 Logistic Regression

It is another case of linear regression.Meanwhile its assumptions of regression model is violated by the binary response variable.The advantage that is approach provides is that it is able to develop probabilistic formula for any classification. The drawback it carries is that this method cannot handle non linear problems and contain interactive effects of variables [6].

3.6.4 K-Neighbors

K-nearest neighbour, or KNN [6] classifier is the method which is based on learning analogy. Upon being presented with an unknown sample, this method searches for a pattern that is closest to the unknown sample and then assign the most common class.Biggest advantage of KNN is that it doesn't require label class prior to the classification. The drawback is its prediction accuracy which is highly susceptible to the cardinality 'k' and the measure of distance of the neighbourhood.

3.6.5 Random forest

This is a supervised ensemble learning algorithm [3], which can be used for both regression problem and for classification as well. However, it is mainly used to classify problems. More trees mean a more robust forest. Likewise, the algorithm of random forest creates a decision tree on data samples, which then receives a prediction from every single one of them and results in a best possible solution by voting. It reduces the issue of over-fitting by averaging the result, and thus proves to be a better option for single decision tree.

3.6.6 XGBoost

This ML algorithm is based on decision tree which uses framework of gradient boosting.It is usually used to solve prediction problems that involve data that is unstructured. ANNs tend to outperform all algorithms. But, when it comes to small-to-medium data, decision tree based algorithm are considered as best-in-class. XGBoost [5] are ensemble tree which uses boosting weak learner through gradient descent.

3.6.7 Light GBM

It uses framework of boosting gradient, which is similar to tree based learning algorithm. With the increasing size in data, the traditional data science algorithms are finding it hard to give faster outputs. The word 'light' is used because this method is faster. It can process larger sizes of data and use lower memory while running.This algorithm is popular as it supports GPU learning [7].

3.6.8 Artificial Neural Networks

Artificial neural networks or ANN [6] is a method that uses mathematical non linear equations to develop a successful relationship between the output and the input variables. This is achieved via a learning process.The back propagation neural network uses supervised learning and feedforward topology to classify the data.This network is basically consist of input layer,hidden layer and output layer. Each layer has several neurons. ANNs easily handle any type of non-linear problem.

3.6.9 Naive Bayesian

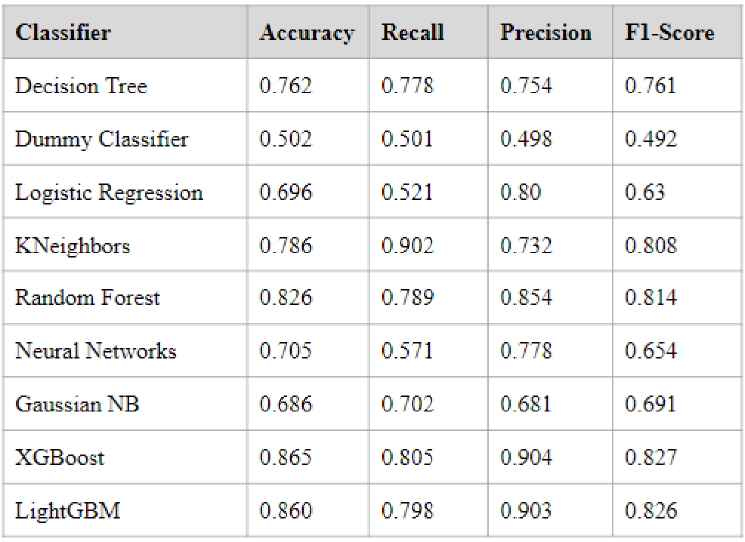
This classifier uses the Bayes theory. It works on the assumption that no two attribute are related that means the occurrences of any attribute is independent from each other. The Bayesian classifier is used when a justification is required for some other classifier, which do not make use of Bayes Theorem.The biggest disadvantage of this algorithm is its assumption but this assumption can be false and there shall be some relation in between variables [6].

4 Evaluation and Results

We compared all the nine classifiers as mentioned in Section using the performance measures: Accuracy, Recall, Precision and F1- Score. Performance metrics of each classifier is noted before and after feature selection.

4.1 Evaluation result without Feature Selection

Initially, total number of features in the dataset were 24, after perform data cleaning there were 30. We measured the performance metrics by applying each classifier. From the table 1, we can say that all the classifiers except Dummy Classifier and Logistic Regression show similar performance. Out of all, it is clear that XGBoost model gives the highest accuracy and can be noted as the best performing classifier.

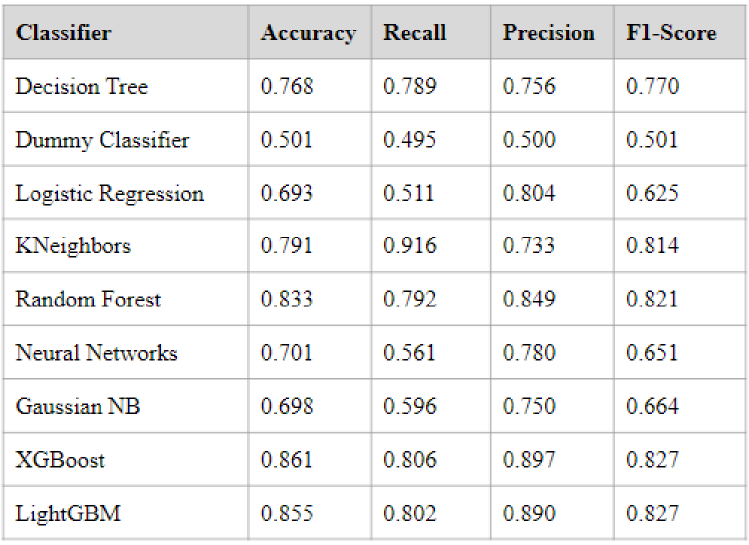


Classifier	Accuracy	Recall	Precision	F1-Score
Decision Tree	0.762	0.778	0.754	0.761
Dummy Classifier	0.502	0.501	0.498	0.492
Logistic Regression	0.696	0.521	0.80	0.63
KNeighbors	0.786	0.902	0.732	0.808
Random Forest	0.826	0.789	0.854	0.814
Neural Networks	0.705	0.571	0.778	0.654
Gaussian NB	0.686	0.702	0.681	0.691
XGBoost	0.865	0.805	0.904	0.827
LightGBM	0.860	0.798	0.903	0.826

Table 1: Evaluation without Feature selection

4.2 Evaluation result with Feature Selection

After applying feature selection using Random Forest, we got 20 important features. Now using these 20 features, we measured the performance metrics by applying the nine classifiers. Comparing Table 2 with Table 1, not much of a difference can be seen in the performance metrics. But the results have certainly shown a slight increase in the values of performance metrics of all the classifiers. However, feature selection decreased the model training time and the complexity of the model. It is clear from the Table 2 that XGboost is still the best performing model whereas, Dummy Classifier depicts the baseline model for the task at hand, and hence has the worst performance.

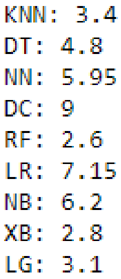


Classifier	Accuracy	Recall	Precision	F1-Score
Decision Tree	0.768	0.789	0.756	0.770
Dummy Classifier	0.501	0.495	0.500	0.501
Logistic Regression	0.693	0.511	0.804	0.625
KNeighbors	0.791	0.916	0.733	0.814
Random Forest	0.833	0.792	0.849	0.821
Neural Networks	0.701	0.561	0.780	0.651
Gaussian NB	0.698	0.596	0.750	0.664
XGBoost	0.861	0.806	0.897	0.827
LightGBM	0.855	0.802	0.890	0.827

Table 2: Evaluation with Feature selection

4.3 Nemenyi Test

Nemenyi Test is a post-hoc statistical test that helps us to analyze the results of the experimental data. We used Nemenyi Test to compare all classifiers with each other on a statistical view.
We used 10-fold cross validation which resulted in 10 values of each performance metrics. We considered 10 Accuracy values and ranked them for each classifier. After ranking, average of ranks were calculated as shown in Table 3. Random Forest(RF) has the lowest rank of 2.6 and Dummy Classifier (DC) has the highest, 9. Using these ranks CD (Critical Difference) was calculated to be 3.7988.



```
KNN: 3.4
DT: 4.8
NN: 5.95
DC: 9
RF: 2.6
LR: 7.15
NB: 6.2
XB: 2.8
LG: 3.1
```

Table 3: Average Ranks

Figure 5 shows Nemenyi diagram of all 9 classifiers. The thick line represents CD that is, how significantly the algorithms are similar to each other. As mentioned in Section 4.2, considering the first thick line denoting CD, output of the majority of the algorithms: Random Forest, XGBoost, LightGBM, K-Nearest Nighbor, Decision Tree and Nueral networks are statistically similar to each other. However, statistically, Random Forest having the lowest average rank can be concluded as best performing algorithm.

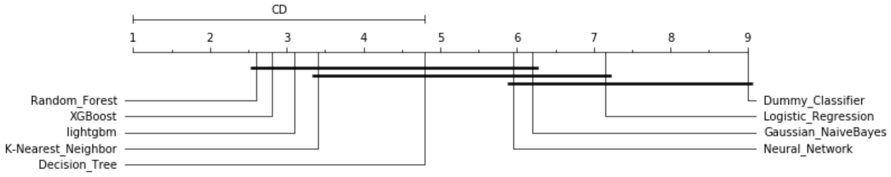


Fig. 5: Nemenyi Test Result

5 Conclusion

With the aim to analyze Credit Card Default prediction dataset and the work done, the following can be concluded:

- The dataset underwent extensive data exploratory analysis, data pre-processing, and cleaning.
- Random Forest method was used for feature selection which saved the model training time, decreased the computational cost, and enhanced the performance of the model.
- Evaluation results show that XGBoost is the best performing model.
- Statistically, using Nemenyi Test, Random Forest can be concluded as the best performing model.

References

1. Algorithm of Random Forest for feature selection `https://chrisalbon.com/machine_learning/trees_and_forests/feature_selection_using_random_forest/`
2. Hasan, Md Rajib, et al. “Single decision tree classifiers’ accuracy on medical data.” 5th International Conference on Computing and Informatics (ICOCI) 2015, 2015
3. Random Forest ensemble algorithm for classification`https://towardsdatascience.com/random-forest-a-powerful-ensemble-learning-algorithm-2bf132ba639d`
4. Showalter, Samuel, and Zhixing Wu. ”Minimizing the Societal Cost of Credit Card Fraud with Limited and Imbalanced Data.” arXiv preprint arXiv:1909.01486 (2019).

5. https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d
6. Yeh, I-Cheng, and Che-hui Lien. ”The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients.” Expert Systems with Applications 36.2 (2009): 2473-2480.
7. https://medium.com/@pushkarmandot/https-medium-com-pushkarmandot-what tgbm-how-to-implement-it-how-to-fine-tune-the-parameters-60347819b7fc