PREDICTING CREDIT CARD DEFAULT

A PREPRINT

Group Name: GROUP A
Department of Biomechanical Engineering
University College London
London, WC1E 6BT

January 11, 2021

1 Introduction

In all ages, businesses in the banking industry that offering financial services such as loan, mortgage or credit card, are facing a crucial problem, the defaulters. With the booming market of financial services, credit card has become an indispensable part of human's life as the rapid expansion of financial services in recent decades. However, this directly leads to a larger risk of credit card default. Therefore, developing a prediction model of credit card payment default is essential for banks offering credit card service. In this report, a machine learning model with the ability of predicting possible default might occur for the coming month; a trained model using data provided with various evaluation methods will be developed. The project involves data transformation & exploration, methodologies and model training & validation with multiple approaches; the final prediction will perform after the comparison of the results from different methodologies.

2 Data Transformation & Exploration

Overall, the training dataset is clean. Several tests were run on the dataset and the results will be discussed later. The goal is to perform data pre-processing on the training dataset. Two different strategies were designed for pre-processing. Both strategies include Sample Balancing, Data Cleansing, One-Hot Encoding, Normalization and Feature Selection. Clustering samples into several groups according to the similarity of features was performed in the second strategy. Several visualization approaches were applied to explore the data at different stages.

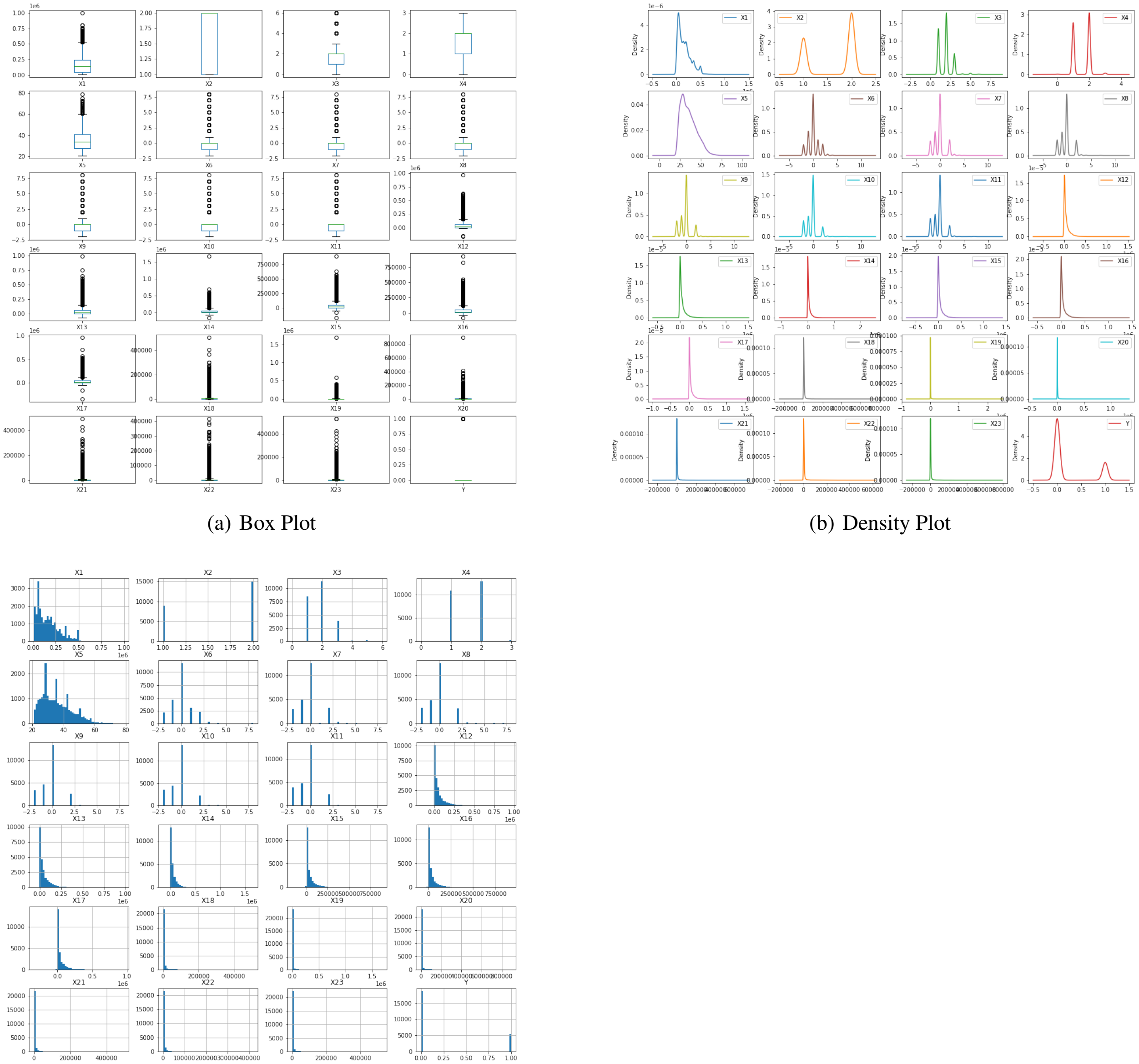
2.1 Nature of Data

The dataset can be classified into two kinds of data which are categorical and quantitative. Categorical data includes 'Gender'(X2), 'Education'(X3), 'Marital status'(X4) and 'History of past payment for each months' (X6–X11). Quantitative data includes the rest of the dataset.

By first visualizing the dataset, then integrity, uniqueness and legitimacy were checked to guarantee a reasonable dataset. No missing values, wrong data types and duplicate IDs were found. Some useless data, specifically the titles and IDs, were removed for further steps. The training set was further explored by three different distribution graphs which are box plot, density map and histogram (**Figure 1**).

From **Figure 1**, several observations were obtained as following: Outliers were detected by box plots; Categorical features contained some values that take up small proportion; The samples were not well-balanced since the number of defaults is far smaller than the number of non-defaults.

By the first observation, the rationality of the four most significant outliers was analysed. It was found that customer [2198] has large value in all quantitative features related to credit, payment and bill. Therefore, the client was concluded to be wealthier than others since this client accounts for most outliers. Therefore, the existence of this outlier was reasonable. Other outliers came from customer [5297], [20893], [12331], and these outliers only had large values in a few features, as these cases could happen due to some emergency circumstances. Thus, they also have been considered as valid data.



(a) Box Plot

(b) Density Plot

(c) Histogtam

Figure 1: Distribution Graphs

By the second observation, a *view_proportion* function was defined to return the percentage of each value grouped by categorical features. The results were used for further data cleansing.

By the third observation, sample balancing was performed, and the implementation will be introduced in the next section.

2.2 Sample Balancing

As mentioned in **Nature of Data**, the number of people who did not default(Y=0) is much larger than the number of people who did default(Y=1), which resulted in bias when predicting.

To avoid this phenomenon, SMOTENC was applied to the original training set to generate new random samples of defaulting data(Y=1). SMOTENC was used instead of SMOTE due to both categorical and quantitative features exist [1].

The distribution before and after sample balancing on the origin training set is shown in **Table 1**:

Table 1: Sample Distribution

	Y=0	Y=1
Before SMOTENC	77.625%	22.375%
After SMOTENC	50%	50%

2.3 Data Cleansing

In addition to removing the useless data, the values with a low proportion also needed to be processed. By viewing the percentage returned by *view_proportion* function, it was obvious that 'education'(X3), 'marital status'(X4) and 'history of past payment' (X6–X11) contains unknown values or values that have a negligible proportion. Therefore, these minorities were merged into one value. For example, there were only 0.1708% values of 0 and 1.1208% values of 3 in X4, thus these values were all merged to value 3. After performing Data Cleansing, the datasets were able to do feature engineering.

2.4 Feature Engineering 1: Adding Features

By exploring quantitative features, negative values were found in 'bill statement' (X12 –X17), which means that the customer prepaid some amount of money for the next month [2]. By discussion, the same amount of money should be added to the bill for next month and set 0 to the current bill statement. And if negative values appeared in the last month, it was treated as 0 since the money was paid for the next month.

Bills represent the ability to spend money and payments represent the ability to repay the bill. Intuitively, the ratio between payment and bill represents the repayment rates. When the ratio overcomes 1, which means the client paid more money than the bill requires, the client should be considered as fully paid and the ratio of pay and bill is set to 1. When the ratio is negative, which means the bill is already prepaid, therefore the ratio is set to 0.

Considering these relationships, several new features were added:

i) Total payment from May to September: **'Total_pay'**

$$= \mathbf{X18} + \mathbf{X19} + \mathbf{X20} + \mathbf{X21} + \mathbf{X23} + \mathbf{X24}$$

ii) Total bill from May to September: **'Total_bill'**

$$= \mathbf{X12} + \mathbf{X13} + \mathbf{X14} + \mathbf{X15} + \mathbf{X16} + \mathbf{X17}$$

iii) Overall repayment rate: **'tp_tb'**

$$= \frac{\mathbf{X18} + \mathbf{X19} + \mathbf{X20} + \mathbf{X21} + \mathbf{X23} + \mathbf{X24}}{\mathbf{X12} + \mathbf{X13} + \mathbf{X14} + \mathbf{X15} + \mathbf{X16} + \mathbf{X17}}$$

iv) Monthly repayment rate: **'p_b1', 'p_b2', 'p_b3', 'p_b4', 'p_b5', 'p_b6'**

$$= \frac{\mathbf{X18}}{\mathbf{X12}}, \frac{\mathbf{X19}}{\mathbf{X13}}, \frac{\mathbf{X20}}{\mathbf{X14}}, \frac{\mathbf{X21}}{\mathbf{X15}}, \frac{\mathbf{X22}}{\mathbf{X16}}, \frac{\mathbf{X23}}{\mathbf{X17}} \text{ respectively}$$

2.5 Feature Engineering 2: Normalization

Normalization maps the data into [0, 1], and removes units if exist. This helps to improve the efficiency and lowering the difficulty of training.

2.5.1 Purposes of Normalization

On one hand, Normalization is necessary for some models. For example, Normalization speeds up the convergence of gradient descent. Furthermore, some classifiers need to calculate the distance between samples. If the variance of a feature is very large, then the distance calculation mainly depends on this feature, which is contrary to the actual situation because the feature with small variance may be more important. On the other hand, Normalization can avoid numerical problems caused by extremely large values.

2.5.2 Min-Max Normalization

The formula is $\mathbf{x}' = \frac{\mathbf{x}-\mathbf{X_{min}}}{\mathbf{X_{max}}-\mathbf{X_{min}}}$

Min-Max Normalization is usually applied to quantitative features which have relatively small variances.

2.5.3 Nonlinear Normalization: Log

The formula is $\mathbf{x}' = \begin{cases} \frac{\mathbf{log1p(x)}}{\mathbf{log(X_{max})}}, & x \geq 0 \\ \mathbf{0}, & \text{otherwise} \end{cases}$

Log Normalization is usually used on quantitative features with large variances. Before applying Log Normalization, **'numpy.log1p(x)'** was used instead of 'numpy.log(x)' to deal with extremely small numbers x that 1 + x == 1 in floating-point accuracy [3]. The log value was then divided by **'log1p (X_max)'** to map them into [0,1].

2.5.4 Appliance

By the nature of the data, all quantitative features have large variances excluding 'Age'(X5). Reasonable Normalization was applied to every quantitative feature, specifically, Min-Max Normalization was applied to 'Age'(X5) and Log Normalization was applied to other quantitative features.

2.6 Feature Engineering 3: Feature Selection

A large number of features were yielded on this stage, Dimensionality Reduction and Feature Selection were considered. The correlation between features and Y value were investigated by plotting heatmap (**Figure 2**). The positive value stands for positive linear correlation and the negative value stands for negative linear correlation. The strength of linear correlation is directly proportional to the absolute value of heatmap [4].

The bottom row and the rightmost column in the heatmap represent the correlation between each feature and Y. By reviewing the values, it was found that the maximum absolute value was 0.3, thus, Principal Component Analysis approach might not be useful in this case [5]. Although, most of the features had absolute values of 0.1 or 0.2, they still had effects correlated to Y. Eventually with cautious considerations, quantitative features with value of 0 which non-correlated to Y were dropped since no categorical encoding had been done so far. The dropped features were X5, X12-X17 and 'total_bill'.

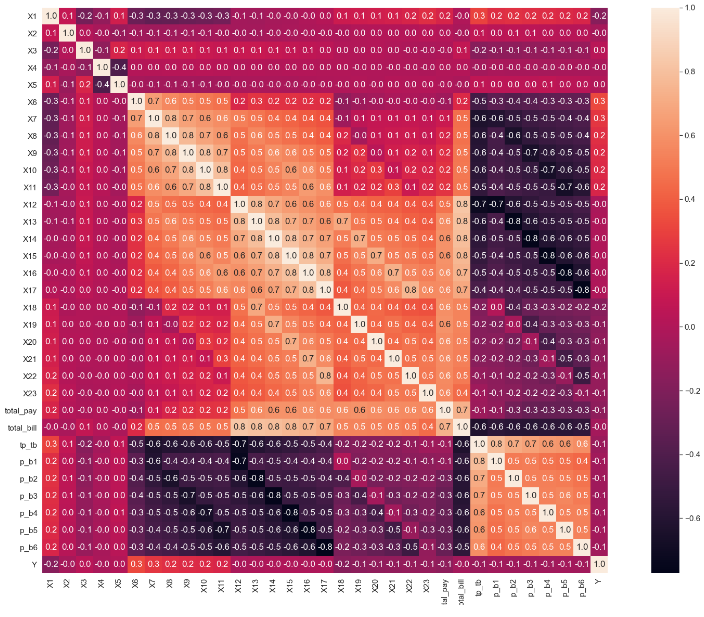


Figure 2: HeatMap

2.7 One-Hot Encoding

One-Hot Encoding is used to quantify the categorical data which generates a vector whose dimension is equal to the number of categories in the dataset. If the data belongs to the i-th category, the component of the vector is assigned with 0, and the i-th component is assigned with 1.[6]

2.7.1 Purposes of One-Hot Encoding

Integer encoding is given as the default encoding method. However, it is not suitable for categorical features which have no ordinal relationship as it causes misleading on the model. If integer encoding is performed on categorical features, the model will relate the weight to the value of the feature, which is not expected. For example, 'Gender'(X2) was originally grouped into 1 and 2, which represent male and female respectively. The integer encoding, however, would distribute more weighting on 2 as it larger than 1 in a mathematical sense. Therefore, One-Hot encoding was applied to resolve this phenomenon, categorical features were replaced by binary features where each represents a category in the categorical feature.

2.7.2 Implementation of One-Hot Encoding

After checking the nature of the new training set and testing set, it was found that categorical data did not cover the same range of categories, which directly leads to inconsistency when applying One-Hot Encoding on each set separately. Therefore, both datasets were merged, and One-Hot Encoding was performed on the merged dataset then split into training set and testing set according to their original sizes.

2.8 Sample splitting

In the original dataset, only one training set and one testing set were provided. Considering the testing set will not be given in real life, a validation set should be generated from the available training set to evaluate models. So, the original training set was split into two sets for training and validation in 80-20 ratio.

2.9 Sample Clustering

Considering that customers may be clustered into groups according to different aspect (wealthiness or level of consumption, for instance), using one model to train different groups of people may not perform well. Instead, training different groups of customers by corresponding best models may perform better. To do this, a K-means model was decided for implementing **'Combined-Model'** (which will be introduced later).

K-means clustering can be used to cluster the customers. It is a vector quantization method, which is originated from signal processing. Its purpose is to divide multiple observations into K clusters, in which each observation belongs to the cluster with the nearest mean (cluster centre or cluster centroid) as the prototype of clustering.[7]

To perform K-means clustering, a variable 'n_clusters' was defined to decide the number of clusters to form. To generate 'n_clusters' centroids, a K-means model was trained using the training set. With the use of the trained model, samples in the validation set and testing set were assigned to the corresponding cluster. In order to train, validate and test each group separately, 'n_clusters' new training sets, new validation sets and new testing sets were generated by splitting their original datasets based on different clusters.

3 Methodology Overview

By studying serval papers and research, a few data pre-processing techniques was learnt [8], such as data cleansing, Normalization, One-Hot encoding, sample balancing. However, Huawei Shan [8] merge the values with small proportion into a value with large proportion, but a different strategy which merges those values into an individual value is used. A, Venkatesh and Gracia [9] have done the feature selection but they ignored the problem of the imbalanced dataset.

The whole task can be split into two main parts, data pre-processing and model training & validation.

For data pre-processing, the nature of the datasets was studied by visualization and statistics methods. According to the observations, effective strategies were applied to pre-process the datasets. Sample Balancing and Data Cleansing result in a balanced and clean dataset for further Feature Engineering. Feature Engineering techniques, which include Normalization, Feature Adding/Selection, Categorical Feature Encoding and Sample Clustering, were applied to the datasets. In addition, a K-means model was trained to cluster similar samples in order to get separate groups, which might result in better performance in model training & validation.

For model training & validation, several models were applied as introducing in following paragraphs. The standard models were fitted with pre-processed data first, with default hyperparameters. Then, for observing better default(Y) predictions, cross validation was applied, various hyperparameters were selected and adjusted for different models. The 'Combined-Model', on the other hand, requires initial processing and comparison to select the best combination. Afterward, a comparison was made among models, by taking considerations of F1 Score, Accuracy, Recall and Precision into account, to reach the final decision.

Logistic regression is a predictive examination to portray information and will demonstrate the relationship between binary variables and more nominal, ordinal, interval or ratio-level independent variables. Sometimes logistic regressions are difficult to interpret; the Statistics tool easily allows users to conduct the analysis. For the detection of credit card fraud, logistic regression is a comparably proper model as both data and the situation of payment are binary [10].

Naïve Bayes classifiers is a typical type of probabilistic classifiers which basically applying Bayes' Theorem. A strong assumption to support this is that all features are independent to each other [11].

Linear Discriminant Analysis (LDA) is a commonly use classification technique, also a dimensionality reduction technique. It seeks a linear combination of input variables that can provide maximum separations of samples between class and a minimum separation of samples within each specific class [12].

Quadratic discriminant analysis (QDA), a similar classifier as LDA, both get observation form classes of 'Y' deriving from Gaussian distribution, whereas, for QDA each class has alternative covariance matrix. Comparing with LDA, it has more parameter to estimate than LDA. QDA gets better results when the decision boundaries are non-linear, especially in the quadratic case. For this limited dataset, QDA might be act as a reference, rather than a major model [13].

Random Forest Classifier is consisting of large amounts of binary decision trees. It classifies each new input attribute by putting it into each tree respectively, to obtain a classification from each decision tree, the model decision will be obtained from the most votes from decision trees [14].

Support Vector Machine (SVM) is a supervised machine learning model used to deal with binary classification problems. It is always used to find a hyperplane in N-dimensional space which is the number of the attributes, separating the two classes of data points. SVM aims to find the best hyperplane which has the maximum margin [15].

A Neural Network contains a train of algorithms used to observe the potential and hidden relationships in the dataset by the mechanism that imitate the action of human brain. It has the ability to work with incomplete information, the output performance will depend on the importance and usefulness of the lost information. Besides, Neural Network also allows people to learn different functions [16].

Mechanism for 'Combined-Model':

Define:

$$\begin{aligned}\mathcal{T} &= \text{Original Training Set}\\ \mathcal{V} &= \text{Original Validation Set}\\ \mathcal{X} &= \text{Original Testing Set}\end{aligned}$$

After performing sample clustering:

$$\begin{aligned}\mathcal{T} &= \{\mathcal{T}_0, \mathcal{T}_1, \cdots, \mathcal{T}_{n-1}\}\\ \mathcal{V} &= \{\mathcal{V}_0, \mathcal{V}_1, \cdots, \mathcal{V}_{n-1}\}\\ \mathcal{X} &= \{\mathcal{X}_0, \mathcal{X}_1, \cdots, \mathcal{X}_{n-1}\}\end{aligned}$$

Where $\{\mathcal{T}_n, \mathcal{V}_n, \mathcal{X}_n\}$ is the training set, validation set and testing set for cluster n.

The 'Combined-Model' looks like this:

$$\{\text{best model}_0, \text{best model}_1, \cdots, \text{best model}_{n-1}\}$$

In order to get this, several standard classifiers are cross-validated for each cluster, e.g. Logistic Regression$_1$ is the best Logistic Regression for cluster 1.

Then choose the best model for each cluster and that's 'Combined-Model':

$$\text{best model}_n = \text{Best}(\text{Logistic Regression}_n, \text{Naive Bayes}_n, \cdots, \text{Neural Network}_n)$$

For training, validation and testing:

Define

$$\begin{aligned}\text{A.fit(B)} &= \text{train model A on the dataset B}\\ \text{A.predict(B)} &= \text{predict the results of B by using model A}\end{aligned}$$

Training:

$$\text{best\_model}_n\text{.fit}(\mathcal{T}_n)$$

Validation:

$$\text{best\_model}_n\text{.predict}(\mathcal{V}_n)$$

Testing:

$$\text{best\_model}_n\text{.predict}(\mathcal{X}_n)$$

4 Model Training & Validation

4.1 Cross Validation [17]

The basic idea is to split the initial training data set into k folds. The learning algorithm is run k times, for each model, each time using all the folds but one as a training set, $\mathcal{S}\backslash\mathcal{S}_k$, and the remaining fold as a validation set, $\mathcal{S}_k$

The validation performance, for a particular model f, is averaged across all k folds to give the cross-validation loss:

$$\mathbf{L}_{CV_k}(\mathcal{E},\mathcal{S},f)=\frac{1}{k}\sum_{i=1}^{k}\mathbf{L}_{\mathcal{S}\backslash\mathcal{S}_k}(\mathcal{E},\mathcal{S}\backslash\mathcal{S}_k,f)$$

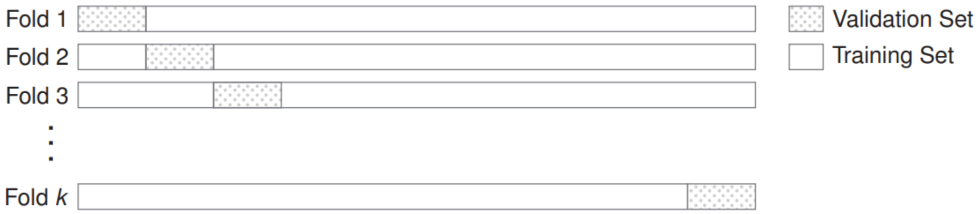


Figure 3: Cross Validation

4.2 Hyperparameter

A hyperparameter is a parameter which can be used to control learning process of models and it cannot be learnt from the training data. Choosing a hyperparameter with different sizes or types can always lead to various speed and quality of learning process. Hence, some hyperparameters with great significance have been tuned for each model and the selection of hyperparameters tuned for each model is listed in **Table 2** and the meaning of each hyperparameter is listed in **Table 3**.

4.3 Model Training

Standard classifiers were trained on the training set by a usual way. The training for **'Combined-Model'** was more complex than the standard classifiers. Cross-validation was performed per cluster per model to tune and seek the optimal hyperparameters with the same method as standard classifiers.

After cross-validation, each model had 'n_clusters' different trained versions and each version of model had a corresponding cluster. The best model for each cluster was obtained by selecting the one with the largest F1 Score (the reason for using F1 Score will be discussed in the next section) within the cluster. A function for selecting the best model for each cluster was implemented in section **Best Model Selection**. In doing so, the best model for each cluster was trained in order to get the best result.

Table 2: Selection of Hyperparameters

Classification Algorithms	**Hyperparameters**
Logistic Regression	C; Penalty; Solver
Naïve Bayes	Var_smoothing
LDA	Solver
QDA	Reg_param
Random Forest	Criterion; n_estimators
Linear SVC	C; Dual; Penalty
Neural Network	Activation; Hidden_layer_sizes; Solver

Table 3: Meaning of Hyperparameters

Hyperparameters	**Meaning**
C	Inverse of regularization strength; Smaller value of C specifies stronger regularisation
penalty	Specify the norm in the penalisation
solver	Specify the algorithm used in the optimisation problem
var_smoothing	State the part of the greatest variance in all features
reg_param	Regularize the per-class covariance estimates
criterion	Represent the function used to measure the quality of a split
n_estimators	Specify the number of trees in the forest
dual	Select the algorithm which can be used to deal with the dual or primal optimization problem
activation	Activation function for the hidden layer
hidden_layer_sizes	State the number of neurons in the corresponding hidden layer is specified by the ith element

5 Results

5.1 Model Hyperparameters

The final hyperparameters tuned for each standard classifier are listed in **Table 4**.

Table 4: Final Hyperparameters

Classification Algorithms	**Hyperparameters**
Logistic Regression	LogisticRegression(C=0.5, penalty='l1', solver='liblinear')
Naïve Bayes	GaussianNB(var_smoothing=1e-11)
LDA	LinearDiscriminantAnalysis()
QDA	QuadraticDiscriminantAnalysis(reg_param=0)
Random Forest	RandomForestClassifier(criterion='entropy', n_estimators=140)
Linear SVC	LinearSVC(C=0.05)
Neural Network	MLPClassifier(activation='tanh', hidden_layer_sizes=(10, 8, 6, 4), solver='sgd')
'Combined-Model' (n_cluster = 2)	GaussianNB(var_smoothing=0.0001),GaussianNB(var_smoothing=0.001)

5.2 Model Predictions

There are some evaluation approaches being widely used to evaluate the performance of training datasets with different models. Results obtained will be discussed in following paragraphs, by introducing and utilising several significant assessment approaches such as F1 Score, Accuracy, Recall as well as Precision[18]. The results are listed in **Table 5**.

Table 5: Results

Classification Algorithms	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.823	0.679	0.364	0.474
Naïve Bayes	0.811	0.576	0.515	0.543
Linear Discriminant Analysis (LDA)	0.823	0.667	0.386	0.489
Quadratic Discriminant Analysis (QDA)	0.777	0.492	0.540	0.515
Random Forest	0.815	0.628	0.384	0.477
Linear SVC	0.824	0.687	0.361	0.473
Neural Network	0.823	0.662	0.387	0.489
'Combined-Model' (n_cluster = 2)	0.779	0.495	0.573	0.531

Table 5 contains all of the classification algorithms applied to the training dataset and the corresponding performance results of evaluation approaches. It is clear that the QDA has the worst performance of accuracy and precision, but outstanding performance of recall. Besides, Naïve Bayes has shown the best performance of F1 Score, but awful outcome of precision. Linear SVC, Neural Network, LDA, Logistic Regression and Random Forest have produced similar results with four performance evaluation approaches. The 'Combined-Model' has shown superior performance of recall and F1 Score but atrocious outcome of accuracy and precision.

In reality, banks will pay more attention to those who have a high probability of default as this may result financial loss and burden to banks. Thus, both recall and precision play the main role in this study, to select the final model, these two parameters should be considered simultaneously. Therefore, F1 Score, which balances precision and recall, is particularly significant and contribute a higher weighting than other evaluation approaches since they can assess the ability of whether banks can accurately identify the people who are likely to default. Overall, Naïve Bayes was approved to be the best classification algorithm when dealing with this particular credit card default problem due to the outstanding performance of F1 Score and desirable outcome of the remaining evaluation approaches.

6 Final Predictions on Test Set

After comparing the results from the models, Naïve Bayes was chosen to be the final model. It was trained again with the original training set, then the model was applied on the test set which gave the following results (**Table 6**).

Table 6: Final Prediction

Classification Algorithms	Accuracy	Precision	Recall	F1 Score
Naïve Bayes	0.8075	0.544153	0.540284	0.542212

7 Conclusion

Overall, a few novel operations were performed during the whole task. In data pre-processing, negative values in the bill were managed, and a few features were added or deleted. For the training model, a **'Combined-Model'** which splits the set into serval smaller sets was brought out. And evaluation mostly focuses on the F1 Score.

Furthermore, the **'Combined-Model'** did not perform as expected on this data set. However, the performance on a larger dataset or a different number of clusters requires further research and study.

References

[1] Imbalanced-learn.org. 2021. Imblearn.Over_Sampling.SMOTENC – Imbalanced-Learn 0.7.0 Documentation.
[online] Available at: https://imbalanced-learn.org/stable/generated/imblearn.over_sampling.SMOTENC.html

[2] Kaggle.com. 2021. Default Of Credit Card Clients Dataset.
[online] Available at: https://www.kaggle.com/uciml/default-of-credit-card-clients-dataset/discussion/34608

[3] Numpy.org. 2021. Numpy.Log1p — Numpy V1.19 Manual.
[online] Available at: https://numpy.org/doc/stable/reference/generated/numpy.log1p.html

[4] Zach, V., 2021. How To Read A Correlation Matrix - Statology.
[online] Available at: https://www.statology.org/how-to-read-a-correlation-matrix/

[5] Originlab.com. 2021. Help Online - Origin Help - Principal Component Analysis.
[online] Available at: https://www.originlab.com/doc/Origin-Help/PrincipleComp-Analysis#:~:text=PCA%20should%20be%20used%20mainly,0.3%2C%20PCA%20will%20not%20help

[6] DeepAI. 2021. One Hot Encoding.
[online] Available at: https://deepai.org/machine-learning-glossary-and-terms/one-hot-encoding

[7] En.wikipedia.org. 2021. K-Means Clustering.
[online] Available at: https://en.wikipedia.org/wiki/K-means_clustering

[8] Huawei Shan. Research on Bank Credit Card Default Prediction Based on Machine Learning [J]. Hans Journal of Data Mining, 2019, 9(4): 145-152. DOI: 10.12677/hjdm.2019.94018
[online] Available at: https://doi.org/10.12677/hjdm.2019.94018

[9] Venkatesh, A. and Gracia, S. (2016) 'Prediction of Credit-Card Defaulters: A Comparative Study on Performance of Classifiers', International Journal of Computer Applications, 145(7), pp. 36–41. doi: 10.5120/ijca2016910702.

[10] What is Logistic Regression? Statistics Solutions. 2021.
[online] Available at: https://www.statisticssolutions.com/what-is-logistic-regression/

[11] Gandhi R. Naive Bayes Classifier. Medium. 2018.
[online] Available at: https://towardsdatascience.com/linear-discriminant-analysis-explained-f88be6c1e00b

[12] YANG X. Linear Discriminant Analysis, Explained. Medium. 2020. [online] Available at: https://imbalanced-learn.org/stable/generated/imblearn.over_sampling.SMOTENC.html

[13] Döring M. Linear, Quadratic, and Regularized Discriminant Analysis.Datascienceblog.net. 2018.
[online] Available at: https://www.datascienceblog.net/post/machine-learning/linear-discriminant-analysis/

[14] Yiu T. Understanding Random Forest. Medium. 2019. Imblearn.Over_Sampling.SMOTENC – Imbalanced-Learn 0.7.0 Documentation.
[online] Available at: https://towardsdatascience.com/understanding-random-forest-58381e0602d2

[15] Medium. 2020. Support Vector Machine – Introduction To Machine Learning Algorithms.
[online] Available at: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

[16] Investopedia. 2020. Neural Network Definition.
[online] Available at: https://www.investopedia.com/terms/n/neuralnetwork.asp

[17] Dariush,H. 2020. BENG0095: Model Selection & Assessment [Slides]
[online] Available at: https://moodle.ucl.ac.uk/pluginfile.php/3269917/mod_resource/content/2/6_ML_ModelSelection.pdf

[18] Medium. 2021. Metrics To Evaluate Your Machine Learning Algorithm
[online] Available at: https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234