BT2103 Project

2023-03-23

1. Introduction of dataset & Modelling of the problem

The data set consists of 30,000 credit card holders information which is obtained from a bank in Taiwan. There are 23 feature attributes (V2 - V24) and 1 target variable which are further explained below. The target feature (V25) is predicted to be a binary value 0 (= not default) or 1 (= default).

This study reviewed the literature and used the following 23 variables as explanatory variables:

- ID (V1): Indexation/ID of each individual (From 1 to 30,000).
- LIMIT_BAL (V2): Amount of the given credit in NT dollar. It includes both the individual consumer credit and his/her family (supplementary) credit.
- SEX (V3): Gender of the individual (1 = male; 2 = female).
- EDUCATION (V4): Education level of the individual (1 = graduate school; 2 = university; 3 = high school; 4 = others).
- MARRIAGE (V5): Marital status of the individual (1 = married; 2 = single; 3 = others).
- AGE (V6): Age of the individual in terms of years.
- PAY_0, PAY_2 to PAY_6 (V7 - V12): History of past payment. We tracked the past monthly payment records (from April to September, 2005) as follows: PAY_0 (V7) = the repayment status in September, 2005; PAY_2 (V8) = the repayment status in August, 2005; . . .;PAY_6 (V12) = the repayment status in April, 2005. The measurement scale for the repayment status is: -2 = no consumption; -1 = pay duly; 0 = the use of revolving credit; 1 = payment delay for one month; 2 = payment delay for two months; . . .; 8 = payment delay for eight months; 9 = payment delay for nine months and above.
- BILL_AMT1 to BILL_AMT6 (V13-V18): Amount of bill statement in NT dollar. BILL_AMT1 (V13) = amount of bill statement in September, 2005; BILL_AMT2 (V14) = amount of bill statement in August, 2005; . . .; BILL_AMT6 (V18) = amount of bill statement in April, 2005.

- PAY_AMT1 to PAY_AMT6 (V19-V24): Amount of previous payment in NT dollar. PAY_AMT1 (V19) = amount paid in September, 2005; PAY_AMT2 (V20) = amount paid in August, 2005; . . .; PAY_AMT6 (V24) = amount paid in April, 2005.

With the given data set and features attributes, we wish to seek out ways to classify our credit card holders as either a defaulter or a non-defaulter so that the bank can make appropriate decisions to approve the loan depending on the resulting classification. This is because a bank would want to avoid misclassifying a defaulting customer as a non-defaulter, which would result in increased financial risk and losses for the bank. Likewise, the bank would also wish to avoid the scenario of misclassifying a non-defaulting customer as a defaulter, which would result in them losing potential revenue and profits from a customer. Therefore, it is important to correctly classify the customer in order to maximise their profits and reduce their risk and losses.

2.1 Exploratory data analysis

From the initial overview of the data set, we can see that although there are no missing values in any of the observations, there are some values that seem to differ from the supposed range of values. For example, under *EDUCATION (V4)*, the supposed range of values is from `1` to `4`. However, we observe that there are 14, 280 and 51 observations with value of `0`, `5` and `6` respectively. Since there is already an existing value `4` that corresponds to `Others`, we will reclassify these observations under `Others` instead to retain as many observations as possible.

Table 1: Distribution of Education (V4)

V4	n
0	14
1	10585
2	14030
3	4917
4	123
5	280
6	51

Similarly under *MARRIAGE (V5)*, the supposed range of values is from `1` to `3`. However, we observe that there are 54 observations with value of `0`. For these observations, Since there is already an existing value `3` that corresponds to `Others`, we will reclassify these observations under `Others` instead to retain as many observations as possible.

Table 2: Distribution of Marital Status (V5)

V5	n
0	54
1	13659
2	15964
3	323

Furthermore, we also observed that for *BILLAMT1 to BILLAMT6 (V13 to V18)*, there are observations with negative numbers. We found that there are a total of 1930 observations with negative bill statements. We treat such values as occurrences where the bank refunds money to credit card holders, thereby resulting in negative bill amounts. As these are realistic scenarios, we decided not to remove or modify these observations.

Table 3: Frequency of observation with negative bill amount

n
1930

2.2 Visualisation of Observations

We can also analyse different attributes to uncover any potential patterns or imbalance in the observations.

Firstly, we can look at the distribution of defaulter status within our data set to check if the dataset is balanced or imbalanced. We can observe that the data set is imbalanced as among all the observations, 77.9% of the observations are indicated to be of non-defaulter status, and only 22.1% of the observations are indicated to be of defaulter status.

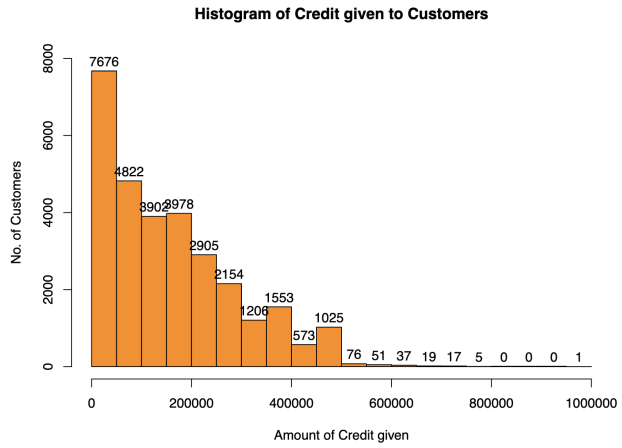
Table 4: Frequency of Credit Card Holder based on Default Status (0 = Non-defaulter, 1 = Defaulter)

V25	n	Percentage
0	23364	77.88
1	6636	22.12

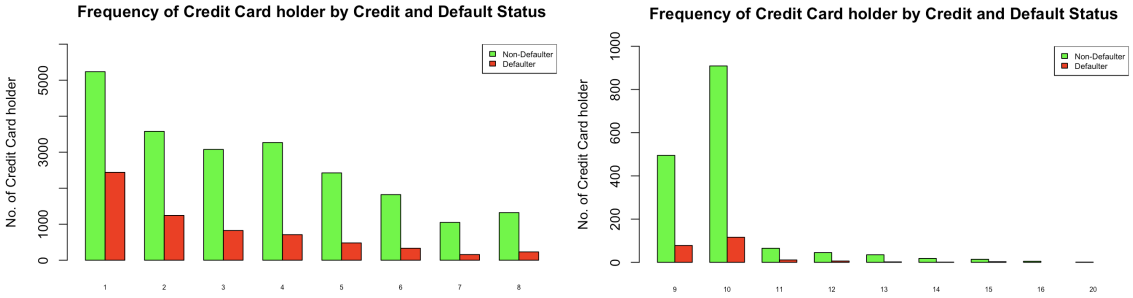
Secondly, we will look at the amount of given credit to the credit card holder as well as the family's credit.

Secondly, we can also inspect the amount of given credit to the credit card holder as well as the family's credit. We observe that majority of the customers falls between `0` to `500,000` in terms of credit given, with most customers (7676) of them, receiving `0` to `50,000` in credit. Though there is a lone customer granted `1,000,000` in credit, which seems to differ greatly

from most other credit card holders, this could potentially be attributed to the customer having a high net worth which would lead to the customer being approved for a much larger credit limit. As such, though this customer seems to be an outlier, the observation is considered realistic and not anomalous. As such, the record will be retained.



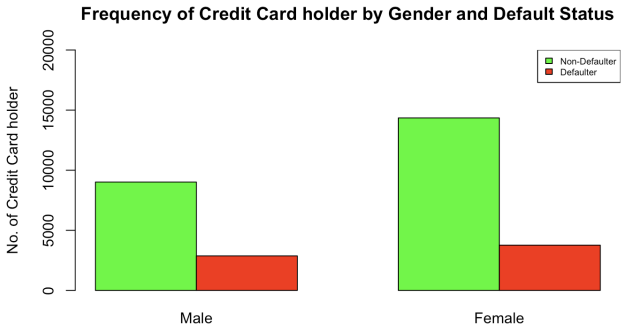
Among the different bin groupings in the histogram, we further sub-divided the observations based on the default status of the credit card holder. We observe that majority of the defaulters and non-defaulters are also found in the `0` to `50,000` grouping. For the remaining groupings, the proportion of non-default status is greater than the default status, contributing to the imbalance dataset with regards to the default status.



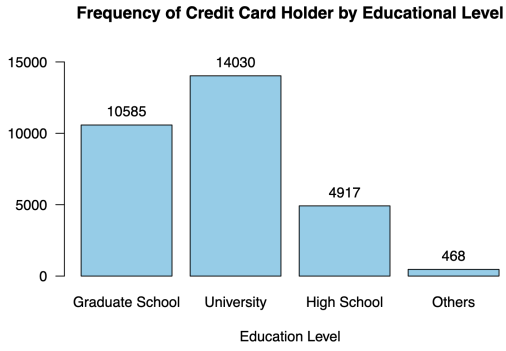
Upon further inspection, we can also observe that although gender is relatively balanced with 60.4% female and 39.6% male credit card holders, there is a higher proportion of defaults for male credit card holders at 24.2%, as compared to 20.8% for the female credit card holders.

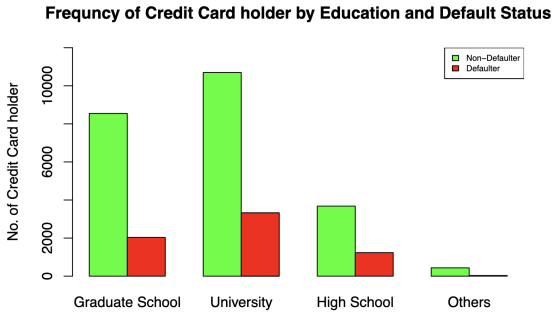
Table 5: Frequency of Credit Card Holder by Gender (1 = Male, 2 = Female)

V3	n	Percentage
1	11888	39.62667
2	18112	60.37333

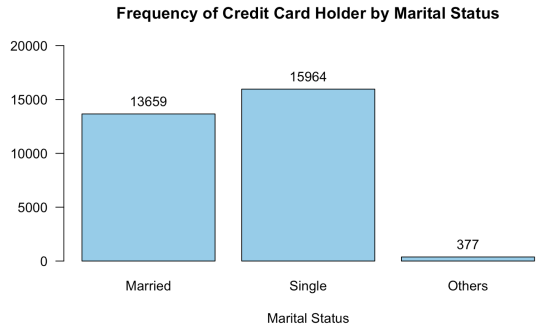


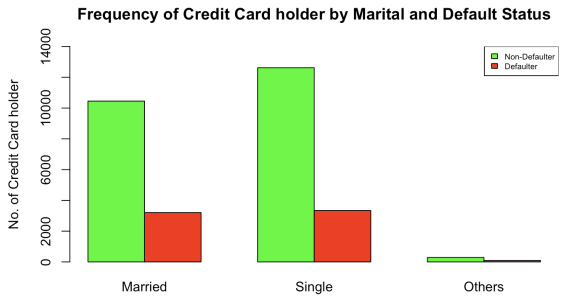
However for education level, we observe that most of the credit card holders are either from Graduate School or University which accounts for 35.2% and 46.8% of the observations respectively. When looking at the default status, credit card holders with `Others` as their indicated educational level actually have the highest defaulter percentage at 25.1%, though it is the least common educational level for the credit card holders. This is in comparison to the University and Graduate School educational levels, which have a proportion of defaulter status of 23.7% and 19.2% respectively.





With regards to marital status, it is relatively balanced with 45.5% of the credit card holder being `Married`, 53.2% of them being `Single` and a small minority of 1.3% being `Others`. Similarly, the proportion of defaulters is relatively equal between the 3 marital statuses, where the proportion is 23.5%, 20.9% and 23.6% for `Married`, `Single` and `Others` marital status respectively.

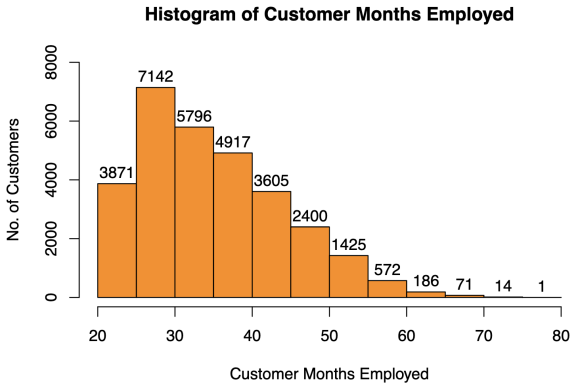


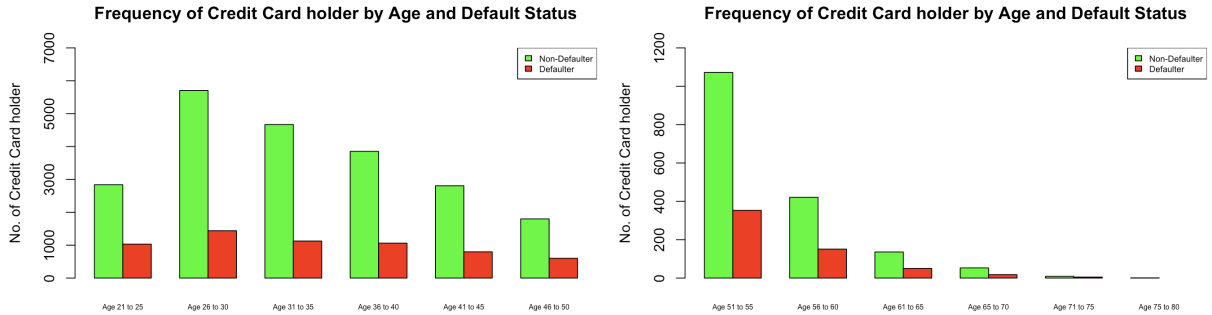


We observe that most credit card holders are between the ages of 20 to 55. When comparing the age and the default status, we can observe that majority of the defaulters are between the age 20 to 50 and fewer are observed in the age group of 51 to 80. This could be caused by their lifestyle habits and expenses, which is potentially higher for the younger generations.

Table 6: Frequency distribution by Age

Age.Group	Freq
(20,25]	3871
(25,30]	7142
(30,35]	5796
(35,40]	4917
(40,45]	3605
(45,50]	2400
(50,55]	1425
(55,60]	572
(60,65]	186
(65,70]	71
(70,75]	14
(75,80]	1





We can also observe that among the different possible observation result for history of past payment, over 50% of the observations for most of the states (`2` to `8`) in each of the variable (V7 to V12) are defaulters.

This is not surprising because for observed states `2` to `8`, the credit card holder has already delayed their payments for at least 2 months. The fact that they are already delaying their credit payments can likely be attributed to their inability to pay the stipulated amount. Therefore, the odds of them being a defaulter is much higher.

Table 7: Proportion of Defaulter in PAY0, PAY2 to PAY6 (V7 to V12)

Data observed	V7	V8	V9	V10	V11	V12
-2	0.1322943	0.1827076	0.1853121	0.1925023	0.1968764	0.2004086
-1	0.1677805	0.1596694	0.1559448	0.1589590	0.1619426	0.1698606
0	0.1281129	0.1591227	0.1745115	0.1832878	0.1885289	0.1884441
1	0.3394794	0.1785714	0.2500000	0.5000000	0.0000000	0.0000000
2	0.6914136	0.5561497	0.5155800	0.5232669	0.5418888	0.5065076
3	0.7577640	0.6165644	0.5750000	0.6111111	0.6348315	0.6413043
4	0.6842105	0.5050505	0.5789474	0.6666667	0.6071429	0.6326531
5	0.5000000	0.6000000	0.5714286	0.5142857	0.5882353	0.5384615
6	0.5454545	0.7500000	0.6086957	0.4000000	0.7500000	0.7368421
7	0.7777778	0.6000000	0.8148148	0.8275862	0.8275862	0.8260870
8	0.5789474	0.0000000	0.6666667	0.5000000	0.0000000	0.0000000

3. Data Pre-processing

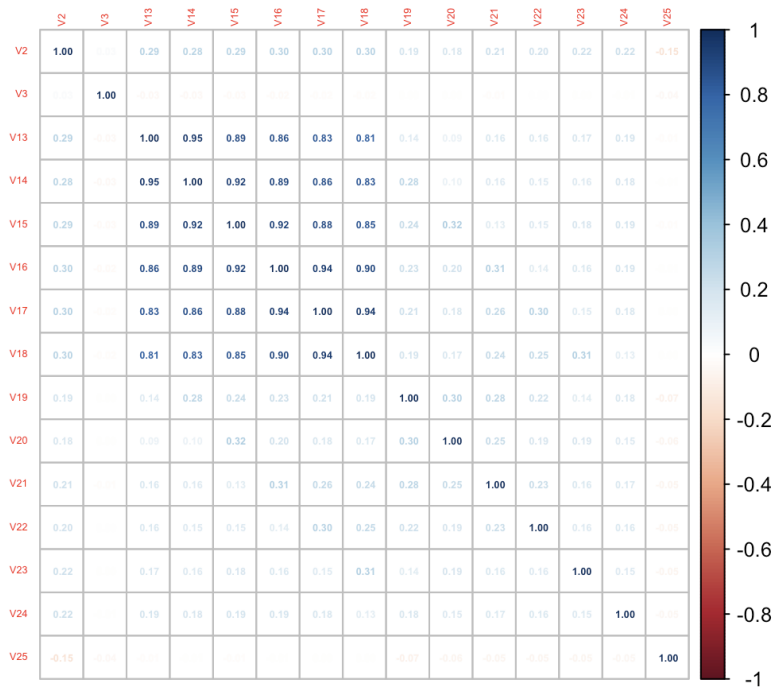
We can first split the data into 2 subsets, 1 being the `trainset`, and the other being the `testset`, in the respective proportion of 3:1. This allows us to be able to train our models effectively, and gauge the subsequent models' effectiveness on the `testset` in an unbiased manner. In addition, we will be scaling our data to ensure that all the features contribute equally to the modeling process. Scaling the dataset will help us to ensure that all features are equally weighted, making our model more accurate and robust. The process we will be scaling our data is normalisation, which rescales our data to a specific range, [0,1], depending on the minimum and maximum values of the data, ensuring that different variables with different ranges and units have a comparable scale.

4. Feature Selection

As the dataset is unbalanced, where majority of the dataset comprises non-defaulters, our feature selection methods have to take the unbalanced nature of the dataset into consideration as well.

Firstly, with regards to feature selection, we can start by removing highly correlated variables. This can be done by plotting a correlation matrix and subsequently identifying which

feature variables have a high correlation with one another. We can then remove such highly correlated variables to ensure that the feature variables are linearly independent from one another.



Based on the results of the plotted correlation matrix, we can keep 1 variable out of V13 - V18, as they are all highly correlated to one another.

Next, given that the data is unbalanced, to avoid our models producing biased results that skew towards the majority class, we can employ over/under sampling to mitigate the class imbalance issue, thereby improving the performance of our models later.

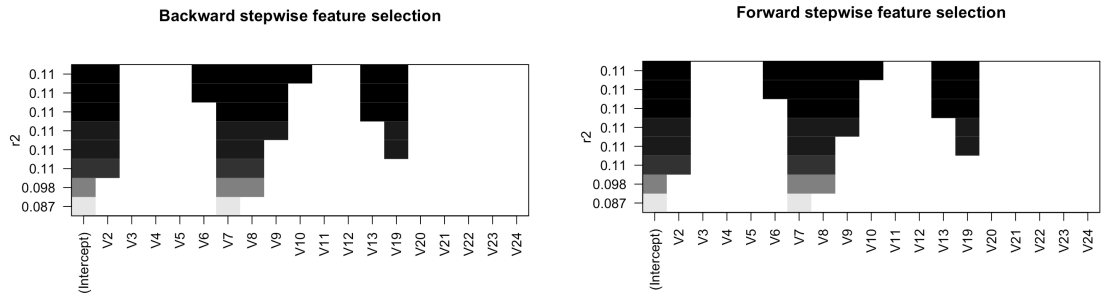
Table 8: Distribution of V25 in train set before processing the data

Var1	Freq
0	17529
1	4971

Table 9: Distribution of V25 after train set processing the data

Var1	Freq
0	11153
1	11347

We can perform wrapper method feature selection (both forward and backward stepwise feature selection) to select for feature variables that are most important and useful for predicting the target variable:



The results of both backward and forward stepwise feature selection presented the same set of suitable variables: {V2, V6, V7, V8, V9, V10, V13, V19}.

In addition, we can also perform Random Forest feature selection to determine the importance of the variables and choose the resultant subset.

Table 10: Random Forest Feature Seletion

	%IncMSE	IncNodePurity
V20	192.048306	747.5466
V19	166.521981	624.3835
V7	160.121247	695.8177
V8	89.112995	464.9068
V9	82.650594	347.3590
V10	81.690808	321.0603
V11	76.738405	287.9677
V12	59.373503	232.2481
V21	44.129320	218.6094
V2	33.921650	176.4468
V22	32.567364	192.0364
V24	19.568373	166.1272
V13	19.474844	170.1103
V23	16.072146	164.7610
V4	6.553056	146.1013
V6	5.183650	145.5613
V3	4.231615	151.0968
V5	3.445181	143.7316

Based on the results above of the random forest feature selection, a suitable set of variables are {V7, V8, V9, V10, V11, V12, V19, V20}.

Model 1: Logistic Regression

Logistic regression is a statistical method used to analyse data in order to make predictions about the probability of a binary outcome. It is a computationally efficient method, making it suitable for large datasets and is able to handle both continuous and categorical predictors. This is suitable for us as it is in line with our current dataset which also have categorical and continuous features. However, the logistic regression model assumes that the predictors and outcomes have a linear relationship. In addition, it requires the independent variables are independent of each other and do not have significant outliers or suffer from multicollinearity. To help aid this model, our pre-processing has removed variables that are highly correlated from each other and irrelevant features are also eliminated during feature selection.

In this case, we are using logistic regression to predict the default status of customers in the next month. We used our processed balanced train dataset to train our model before testing it on the scaled test data. For this model, we used 2 sets of variables taken from backward-forward feature selection as well as the random forest feature selection and will be using the results to evaluate which set of variables would be selected for training the other models.

After passing in the scaled test data with parameter type "Response" to obtain a output between range 0 to 1, we plot the ROC curve to find the AUC and the k-s statistic to find the cutoff in order to find the optimal threshold value. Results above the threshold value was considered to be defaulted and below to be considered as to not have defaulted.

```
##
## Call:
## glm(formula = V25 ~ V2 + V6 + V7 + V8 + V9 + V10 + V13 + V19,
##     family = binomial, data = processed_balanced_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1665  -1.0846   0.4398   1.0638   2.8731
##
## Coefficients:
##             Estimate Std. Error z value             Pr(>|z|)
## (Intercept) -0.88393    0.04831 -18.295 < 0.0000000000000002 ***
## V2          -0.97130    0.10192  -9.530 < 0.0000000000000002 ***
## V6           0.37226    0.06798   5.476       0.000000043453 ***
## V7           3.27700    0.11779  27.821 < 0.0000000000000002 ***
## V8           0.94040    0.10370   9.069 < 0.0000000000000002 ***
## V9           0.74298    0.11753   6.321       0.000000000259 ***
## V10          0.63169    0.11374   5.554       0.000000027977 ***
## V13         -1.16414    0.19016  -6.122       0.000000000925 ***
## V19         -7.68307    0.88370  -8.694 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 31190  on 22499  degrees of freedom
## Residual deviance: 28392  on 22491  degrees of freedom
## AIC: 28410
##
## Number of Fisher Scoring iterations: 4
```

Table 11: Confusion Matrix for logistic regression model with backward-forward feature selection

	1 (Predicted)	0 (Predicted)
1	912	753
0	973	4862

For the logistic regression model using variables selected from the backward-forward feature selection, we achieved an accuracy of 76.99%, an average class accuracy of 69.05% and an area under ROC-curve (AUC score) of 0.7196. The confusion matrix for the model can also be observed above.

```
##
## Call:
## glm(formula = V25 ~ V20 + V19 + V7 + V8 + V9 + V10 + V11 + V12,
##     family = binomial, data = processed_balanced_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0431  -1.1004   0.4482   1.0737   2.6957
##
## Coefficients:
##             Estimate Std. Error z value             Pr(>|z|)
## (Intercept) -1.17155    0.03225 -36.326 < 0.0000000000000002 ***
## V20         -6.63045    1.19185  -5.563        0.00000002649 ***
## V19         -8.84762    0.89316  -9.906 < 0.0000000000000002 ***
## V7           3.25793    0.11752  27.722 < 0.0000000000000002 ***
## V8           0.90083    0.10390   8.670 < 0.0000000000000002 ***
## V9           0.69132    0.11886   5.816        0.00000000601 ***
## V10          0.53633    0.12308   4.358        0.00001315302 ***
## V11          0.30814    0.12679   2.430                0.0151 *
## V12          0.17362    0.11820   1.469                0.1418
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 31190  on 22499  degrees of freedom
## Residual deviance: 28535  on 22491  degrees of freedom
## AIC: 28553
##
## Number of Fisher Scoring iterations: 4
```

Table 12: Confusion Matrix for logistic regression model with random forest feature selection

	1 (Predicted)	0 (Predicted)
1	870	795
0	729	5106

For the logistic regression model using variables selected from the random forest feature selection, we achieved an accuracy of 79.69%, an average class accuracy of 69.88% and an AUC score of 0.7139. The confusion matrix for the model can also be observed above.

It can be observed that some variables such as V12, with p-values of 0.1418 which is larger than 0.05, that were selected by our random forest feature selection is not significant when predicting whether a customer will default or not. Comparatively, all variables selected by the backwards and forwards feature selection are significant, with p-values less than 0.05. In addition, the logistic regression model using variables taken from backward and forward feature selection have a higher AUC score, meaning that it is able to better distinguish between the default and non-default. Hence, we will be using the variables selected from our backward forward feature selection for the subsequent models as they are significant in predicting whether a customer will default or not.

Model 2: Support Vector Machine

A Support Vector Machine is an algorithm that can to used for classification. It finds a hyperplane that can separate the data into our 2 classes, default and non-default, and is chosen to be as far away as the closest data points from each of the classes. This model is able to work well with more complex data, but is highly sensitive to the kernel function as well as other parameters. In addition, it is computationally expensive and not efficient compared to other models.

We used a Support Vector Machine, specifically of type "C-Classification" for binary classification to predict the default status of customers in the next month. Each model will similarly be trained using our processed balanced training dataset and will then be tested on our scaled test dataset. Since SVM is very sensitive to its different parameters, we attempted different models in order to observe the difference in performance.

```
##
## Call:
## svm(formula = fmla, data = processed_balanced_train, type = "C-classification",
##     kernel = "linear")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  1
##
## Number of Support Vectors:  17787
```

Table 13: Confusion Matrix for C-Classification SVM model

	1 (Predicted)	0 (Predicted
1	1044	621
0	1691	4144

For the model above, we did not specify any parameters to have an estimation of the performance of a basic SVM model. We obtained an accuracy of 69.17%, an average class accuracy of 66.86% and an AUC score of 0.6686121. The confusion matrix for this model can also be observed above.

```
##
## Call:
## svm(formula = fmla, data = processed_balanced_train, type = "C-classification",
##     kernel = "linear", cross = 10, cost = 1)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  1
##
## Number of Support Vectors:  17787
```

Table 14: Confusion Matrix for C-Classification with 10-fold cross validation SVM model

	1 (Predicted)	0 (Predicted)
1	1044	621
0	1691	4144

To improve the first model, we performed 10-fold cross validation to prevent overfitting and allow the model to be a more generalised one. For this model, We obtained a similar accuracy of 69.17%, an average class accuracy of 66.86% and an AUC score of 0.6686121. The confusion matrix for this model can also be observed above.

```
##
## Call:
## svm(formula = fmla, data = processed_balanced_train, type = "C-classification",
##     kernel = "linear", cross = 10, cost = 1, class.weights = class_weights)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  1
##
## Number of Support Vectors:  17761
```

Table 15: Confusion Matrix for C-Classification with 10-fold cross validation and weighted SVM model

	1 (Predicted)	0 (Predicted)
1	1018	647
0	1547	4288

To further improve the previous model, we added class weights to penalise customers who default by placing more emphasis on detecting customers who are likely to default. For this model, We obtained an accuracy of 70.74%, an average class accuracy of 67.31% and an AUC score of 0.6731436. The confusion matrix for this model can also be observed above.

Neural Network Model

Neural network models is also computationally expensive and can be prone to overfitting, depending on the hyperparameters inputted for the model, but they are able to capture complex nonlinear relationships in the data where during model training, the model adjusts the weights of the connections between its neurons to minimize the difference between the predicted and actual values.

As such, with our set of selected variables, we can run a neural network model in order to generate an accurate predictive classification of defaulters and non-defaulters.

Table 16: Confusion Matrix for neural network model

	1 (Predicted)	0 (Predicted)
1	849	816
0	676	5159

Above shows the confusion matrix based on the prediction done by the neural network model.

Random Forest Model

As the random forest model constructs and combines decision trees in order to generate its predictive model, it greatly reduces the chance of model overfitting. Furthermore, due to its ability to handle high dimension datasets like the one used, as well as its robust capacity to handle outliers, we decided to run the random forest model as well.

Table 17: Confusion Matrix for random forest model

	1 (Predicted)	0 (Predicted)
1	1097	568
0	1531	4304

The confusion matrix of the Random Forest model is illustrated above.

6. Model Evaluation

For our model evaluation, we will be using Default (1) as our positive instance and Non-Default (0) as our negative instance for our calculations and interpretations.

Table 18: Model Performance Metrics

Model	Acc	Avg_Class_Accuracy	Recall	Precision	F1_Score	AUC
Logistic Regression	0.7968000	0.6987934	0.5225225	0.5440901	0.5330882	0.7139351
SVM	0.7074667	0.6686121	0.6270270	0.3817185	0.4745455	0.6686121
SVM w/ K-fold	0.6917333	0.6686121	0.6270270	0.3817185	0.4745455	0.6686121
SVM w/ weighted K-fold	0.7074667	0.6731436	0.6114114	0.3968811	0.4813239	0.6731436
Neural Network	0.8010667	0.6970286	0.5099099	0.5567213	0.5322884	0.6970286
Random Forest	0.7201333	0.6982383	0.6588589	0.4174277	0.5110645	0.6982383

We will be able to make a more in-depth evaluation of the models through the various metrics obtained. Specifically, we can gauge the models' performances through `Accuracy`, `Average Class Accuracy`, `Precision`, `Recall`, `F1` and `Area Under ROC Curve`.

However, it should be worth noting that since the dataset is very imbalanced in nature, the `Accuracy` metric should not be used as a good gauge of a model's performance. Instead, `Average Class Accuracy` will be a better metric to use, because it considers the accuracy of each class separately and subsequently averages them such that the accuracy is weighted by the classes. This can be illustrated using a trivial model where all instances are predicted as non-default. This particular trivial model would have a good accuracy of 77.8% but would only have an average class accuracy of 50%. Since all our models have an average class accuracy greater than 50%, we would be considering all models as valid and useful for the bank.

Based on the `Area Under ROC Curve` metric, the Logistic Regression model performed the best, indicating that it was the best in distinguishing and classifying between default and non-default observations. The `Average Class Accuracy` metric, which measures the average accuracy per class, is useful given that the dataset is highly imbalanced. With regards to this metric, the Logistic Regression model also performed the best, indicating that despite the imbalanced dataset, the Logistic Regression model was the most accurate at correctly identifying and classifying defaulters and non-defaulters.

With regards to the `Precision`, the Neural Network model performed the best.`Precision` is a measure of the proportion of correctly predicted positive instances out of all instances predicted as positive, including both true positives and false positives. As such, this indicates that the Neural Network model was best at avoiding false positives. In the context of a bank, the Neural Network model was the best at avoiding the misclassification of non-defaulters as defaulters.

The Random Forest model performed the best at the `Recall` metric. Since `Recall` is a measure of the proportion of correctly predicted positive instances out of all actual positive instances, this means that the Random Forest model was the most capable at avoiding false negatives. Within the context of a bank, the Random Forest model was the best at avoiding the misclassification of defaulters as non-defaulters.

The `F1 Score` metric is a harmonic mean of both `Precision` and `Recall`. As such, since the Logistic Regression model performed the best in terms of the `F1 Score` metric, it achieved the best balance between `Precision` and `Recall`, and is overall well-performing in both aspects and is equally capable of avoiding false positives and false negatives during classification.

From the perspective of a bank, there are 2 key considerations for the bank:

1) Minimise the number of defaulters misclassified as non-defaulters
2) Minimise the number of non-defaulters misclassified as defaulters

For consideration 1, this is particularly significant as misclassifying defaulters as non-defaulters would pose a large financial risk on the bank when issuing loans, especially when

the bank is tight on cash as the borrower might delay repayments. As such, if the priority of the bank was to reduce such misclassification occurrences, the `Recall` metric should be a key consideration of the bank when gauging model performance, which means that the Random Forest model should be the most relevant model for the bank to use when making predictive classifications of its customers, as it would greatly reduce the amount of risk that the bank encounters.

Consideration 2 is also important as misclassifying non-defaulters as defaulters would mean that the bank would lose out on a group of potential loan customers and by extension, lose out on potential revenue. As such, under this scenario, the `Precision` metric should be the prioritised and key consideration of the bank when evaluating model performance, which would indicate that the Neural Network model should be used in order to minimise the number of non-defaulters incorrectly classified as defaulters.

Overall, however, due to the strong performance of the Logistic Regression model in the `F1 Score` metric, it should be the primary model that the bank uses to evaluate its customers' risk of default, as it offers great performance in both `Precision` and `Recall`.

7. Possible Improvements

General

Instead of splitting the dataset into only train and test, we should experiment splitting into 3 sub-categories: train, validation and test. By using a separate validation set, we can evaluate the model's performance on a dataset that was not used during training and ensure that the model generalizes well to new data. Moreover, we can also use this validation set to tune the model's hyperparameters to improve its performance. The test set would then be used to evaluate the final performance of the model because it provides an unbiased estimate of the model's performance on new data that was not used during training. This overall helps to reduce the chance of overfitting and further improve our models' performance on actual data.

Support Vector Model

Tuning of the parameters such as choosing a different kernel, as the class may not be linearly separable and using a polynomial kernel may provide a more accurate model. Other parameters such as `cost`, `class.weights` and `margin` can also be modified and tested with different thresholds to possibly obtain a more accurate and performance model.

Neural Network

We can increase the number of hidden layers to the neural network in order to capture the benefits of using a deep-learning learning network. By adding more layers to our model, it increases the ability to learn the complex patterns in the data and make more accurate

predictions. This is done as the network perform more complex transformations on the input data, allowing it to learn more intricate patterns. However, adding too many layers will lead to overfitting. Hence, we should increase the number of hidden layers by a suitable amount to obtain a more accurate model while not over-fitting. The use of validation set mentioned previously in order to validate the number of layers before using the model on the test set can help further improve the model.

Random Forest and Neural Network

In addition, we can also use hyperparameter tuning methods such as Grid Search or Random Search in order to find the optimal hyperparameters so that our Neural Network and Random Forest Models can be further optimised. However, due to the dimensionality of the dataset, the performance optimisations through these methods will take a long time to evaluate, and will possibly require greater computing power.