BT2103 Project

17 November 2022

Student Name	Student Number
Cheong Wen Wei	A0233582E
Gao Heng	A0234014X
Hom Lim Jun How	A0235131W
Lee Jun Wei	A0230329M

Contents

Overview

Problem Description

This project endeavours to accurately predict customers who will default on their bills from those who will pay promptly. More importantly, being able to accurately predict customers who will default would allow banks to minimised potential losses that would potentially be written off as bad debt. Therefore, a stronger emphasis is being placed on being able to accurately predict customers who will default.

Data

The dataset used for this project contains information of 30,000 credit card holders obtained from a bank in Taiwan. Each credit card holder is described by 23 feature attributes, a unique customer identification corresponding to each credit card holder as well as each credit card holder's default status.

Exploratory Data Analysis

Structure of the Data

The first crucial step is to find out more about the dataset. By exploring the structure of the data, it can be discerned that all variables read in were of type integer. However, it is clear that variables V3 (Gender), V4 (Education Level), V5 (Marital Status) and V7 to V12 (Repayment Status) should not be treated as integers. Instead, the aforementioned variables would be converted to a factor to better represent the data.

```
## 'data.frame':    30000 obs. of  25 variables:
##  $ V1 : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ V2 : int  20000 120000 90000 50000 50000 50000 500000 100000 140000 20000 ...
##  $ V3 : int  2 2 2 2 1 1 1 2 2 1 ...
##  $ V4 : int  2 2 2 2 2 1 1 2 3 3 ...
##  $ V5 : int  1 2 2 1 1 2 2 2 1 2 ...
##  $ V6 : int  24 26 34 37 57 37 29 23 28 35 ...
##  $ V7 : int  2 -1 0 0 -1 0 0 0 0 -2 ...
##  $ V8 : int  2 2 0 0 0 0 0 -1 0 -2 ...
##  $ V9 : int  -1 0 0 0 -1 0 0 -1 2 -2 ...
##  $ V10: int  -1 0 0 0 0 0 0 0 0 -2 ...
##  $ V11: int  -2 0 0 0 0 0 0 0 0 -1 ...
##  $ V12: int  -2 2 0 0 0 0 0 -1 0 -1 ...
##  $ V13: int  3913 2682 29239 46990 8617 64400 367965 11876 11285 0 ...
##  $ V14: int  3102 1725 14027 48233 5670 57069 412023 380 14096 0 ...
##  $ V15: int  689 2682 13559 49291 35835 57608 445007 601 12108 0 ...
##  $ V16: int  0 3272 14331 28314 20940 19394 542653 221 12211 0 ...
##  $ V17: int  0 3455 14948 28959 19146 19619 483003 -159 11793 13007 ...
##  $ V18: int  0 3261 15549 29547 19131 20024 473944 567 3719 13912 ...
##  $ V19: int  0 0 1518 2000 2000 2500 55000 380 3329 0 ...
##  $ V20: int  689 1000 1500 2019 36681 1815 40000 601 0 0 ...
##  $ V21: int  0 1000 1000 1200 10000 657 38000 0 432 0 ...
##  $ V22: int  0 1000 1000 1100 9000 1000 20239 581 1000 13007 ...
##  $ V23: int  0 0 1000 1069 689 1000 13750 1687 1000 1122 ...
##  $ V24: int  0 2000 5000 1000 679 800 13770 1542 1000 0 ...
##  $ V25: int  1 1 0 0 0 0 0 0 0 0 ...
```

Missing Values

In this section, a check was performed to identify any missing or N.A. values. From the check, it was identified that there were no missing or N.A. values in the data. Thereafter, a summary of the variables is shown with the exception of V1 which is the customer identification.

```
##        V2                V3              V4              V5       
##  Min.   :  10000   Min.   :1.000   Min.   :0.000   Min.   :0.000  
##  1st Qu.:  50000   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.000  
##  Median : 140000   Median :2.000   Median :2.000   Median :2.000  
##  Mean   : 167484   Mean   :1.604   Mean   :1.853   Mean   :1.552  
##  3rd Qu.: 240000   3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:2.000  
##  Max.   :1000000   Max.   :2.000   Max.   :6.000   Max.   :3.000  
##        V6              V7                V8                V9         
##  Min.   :21.00   Min.   :-2.0000   Min.   :-2.0000   Min.   :-2.0000  
##  1st Qu.:28.00   1st Qu.:-1.0000   1st Qu.:-1.0000   1st Qu.:-1.0000  
##  Median :34.00   Median : 0.0000   Median : 0.0000   Median : 0.0000  
##  Mean   :35.49   Mean   :-0.0167   Mean   :-0.1338   Mean   :-0.1662  
##  3rd Qu.:41.00   3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 0.0000  
##  Max.   :79.00   Max.   : 8.0000   Max.   : 8.0000   Max.   : 8.0000  
##       V10               V11               V12               V13         
##  Min.   :-2.0000   Min.   :-2.0000   Min.   :-2.0000   Min.   :-165580  
##  1st Qu.:-1.0000   1st Qu.:-1.0000   1st Qu.:-1.0000   1st Qu.:   3559  
##  Median : 0.0000   Median : 0.0000   Median : 0.0000   Median :  22382  
##  Mean   :-0.2207   Mean   :-0.2662   Mean   :-0.2911   Mean   :  51223  
##  3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.:  67091  
##  Max.   : 8.0000   Max.   : 8.0000   Max.   : 8.0000   Max.   : 964511  
##       V14              V15               V16               V17        
##  Min.   :-69777   Min.   :-157264   Min.   :-170000   Min.   :-81334  
##  1st Qu.:  2985   1st Qu.:   2666   1st Qu.:   2327   1st Qu.:  1763  
##  Median : 21200   Median :  20089   Median :  19052   Median : 18105  
##  Mean   : 49179   Mean   :  47013   Mean   :  43263   Mean   : 40311  
##  3rd Qu.: 64006   3rd Qu.:  60165   3rd Qu.:  54506   3rd Qu.: 50191  
##  Max.   :983931   Max.   :1664089   Max.   : 891586   Max.   :927171  
##       V18               V19               V20               V21        
##  Min.   :-339603   Min.   :     0   Min.   :      0   Min.   :     0  
##  1st Qu.:   1256   1st Qu.:  1000   1st Qu.:    833   1st Qu.:   390  
##  Median :  17071   Median :  2100   Median :   2009   Median :  1800  
##  Mean   :  38872   Mean   :  5664   Mean   :   5921   Mean   :  5226  
##  3rd Qu.:  49198   3rd Qu.:  5006   3rd Qu.:   5000   3rd Qu.:  4505  
##  Max.   : 961664   Max.   :873552   Max.   :1684259   Max.   :896040  
##       V22              V23                V24                V25        
##  Min.   :     0   Min.   :     0.0   Min.   :     0.0   Min.   :0.0000  
##  1st Qu.:   296   1st Qu.:   252.5   1st Qu.:   117.8   1st Qu.:0.0000  
##  Median :  1500   Median :  1500.0   Median :  1500.0   Median :0.0000  
##  Mean   :  4826   Mean   :  4799.4   Mean   :  5215.5   Mean   :0.2212  
##  3rd Qu.:  4013   3rd Qu.:  4031.5   3rd Qu.:  4000.0   3rd Qu.:0.0000  
##  Max.   :621000   Max.   :426529.0   Max.   :528666.0   Max.   :1.0000  
```

Distribution of the Data

Target Variable

The distribution of the target variable was explored using the table function in R.

```
# Default Table
table(data$V25)
```

```
##
##     0     1
## 23364  6636
```

It can be observed that the dataset is imbalance with approximately 78% of the 30,000 observations being not default while the remaining 22% make up the default customers.

Caterogical Variables

The distribution of the categorical variables were explored using the table function in R.

```
#Gender Table
table(data$V3)
```

```
##
##     1     2
## 11888 18112
```

```
#Education Table
table(data$V4)
```

```
##
##     0     1     2     3     4     5     6
##    14 10585 14030  4917   123   280    51
```

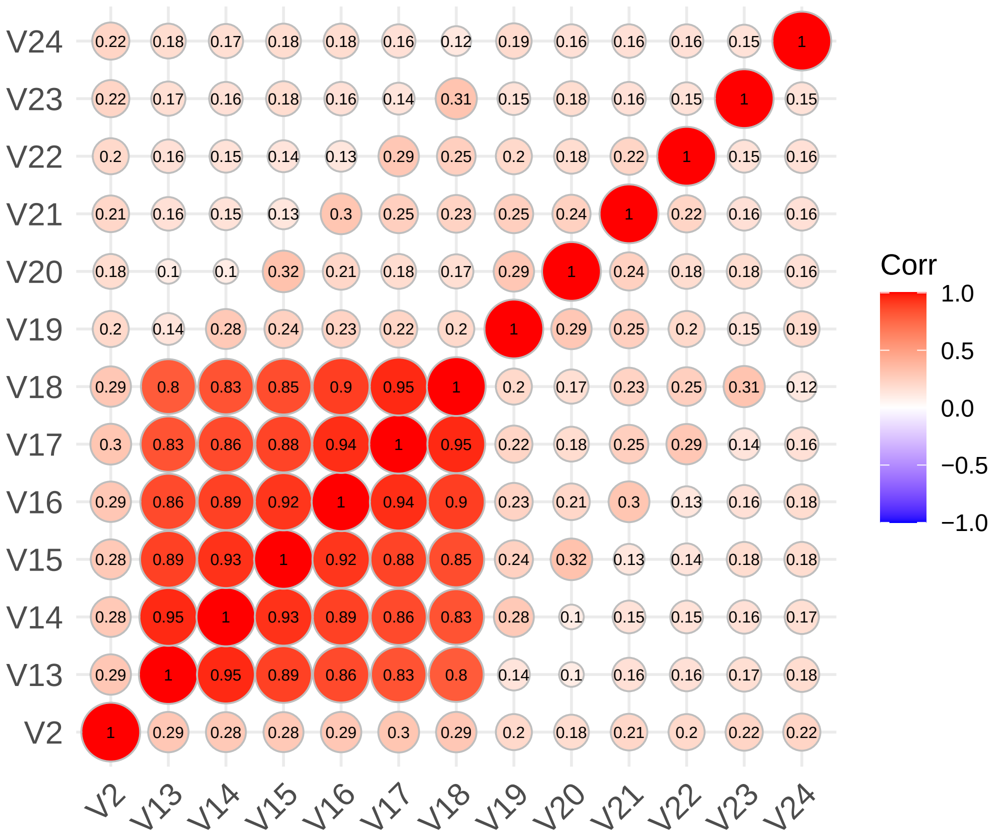
```
#Marital Status
table(data$V5)
```

```
##
##     0     1     2     3
##    54 13659 15964   323
```

It was observed that Education has unknown observations (values of 5 and 6) and Marriage has unknown observations (value of 0). These inconsistencies will be addressed subsequently under the Data Pre-Processing Section.

Continuous Variables

The correlation matrix was used to check the degree of association among the continuous variables from the dataset. From the visualisation, it is evident that V13, V14, V15, V16, V17 and V18 are highly correlated. This is probably due to autocorrelation where the bill amount from the month before affects the bill amount in the current month. As such, feature engineering would be used to overcome the autocorrelation which would be elaborated under the Data Pre-Processing section.



Data Pre-Processing and Feature Engineering

As highlighted in the Exploratory Data Analysis section, there were inconsistencies with the data as well as the problem of autocorrelation. In order to address the inconsistencies in values for the Education, observations that had 0, 4, 5 or 6 as the value for Education would be categorised under 4 as "Others". Similarly, observations that had 0 under the Marriage feature would be categorised under the value 3 as "Others".

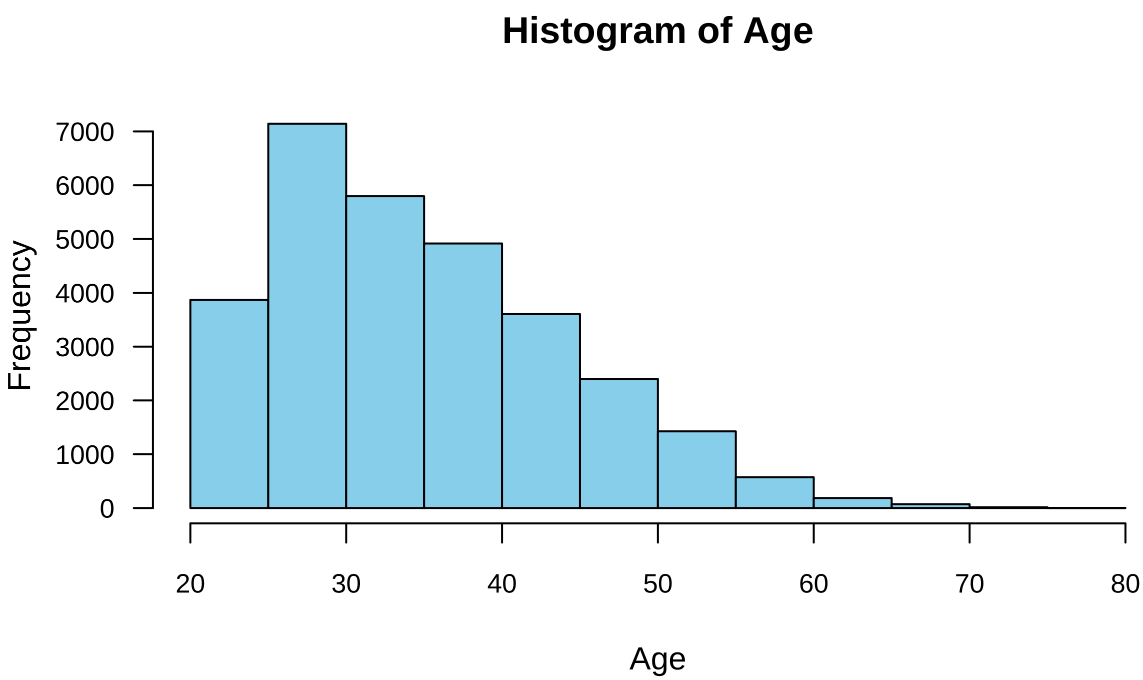
In order to resolve the possible autocorrelation among the features, V13 to V18 as well as V19 to V24, 2 new features would be introduced to represent V13 to V18 and V19 to V24. The first new feature is `mean_col_13_18` which is the average of V13 to V18. Similarly, the second new feature is `mean_col_19_24` would be the average of V19 to V24.

```
#Making a Gender column
data_v$Gender = ifelse(data$V3 == 1, "Male", "Female")

# Firstly modify Education values, change values that are not 1,2,3 to 4.
data$V4 = ifelse(data$V4%in%c(0,4,5,6), 4, data$V4)

# Making an Education column (1 = graduate school;
#       2 = university; 3 = high school; 4 = others).
data_v$Education <- factor(data$V4,
          labels = c("Graduate School", "University", "High School","Others"))

#Replacing Marriage 0 to 3 (1 = married; 2 = single; 3 = others)
data$V5 = ifelse(data$V5 == 0, 3,data$V5)

data_v$`Marital Status` <- factor(data$V5,
                            labels = c("Married", "Single", "Others"))

#Changing data type to factors
data$V5 <- as.factor(data$V5)
data$V4 <- as.factor(data$V4)
data$V3 <- as.factor(data$V3)

data$V7 <- as.factor(data$V7)
data$V8 <- as.factor(data$V8)
data$V9 <- as.factor(data$V9)
data$V10 <- as.factor(data$V10)
data$V11 <- as.factor(data$V11)
data$V12 <- as.factor(data$V12)

#Feature engineering: compress 6 columns to 1 by finding the average of each observation
data_MOD <- mutate(data, mean_col_13_18 = rowMeans(select(data,V13:V18), na.rm = TRUE))
data_MOD <- mutate(data_MOD, mean_col_19_24 = rowMeans(select(data,V19:V24),
                                                       na.rm = TRUE))
```

After the pre-processing of the data, plots were created to visualised the "cleaned" data.

Feature: Age

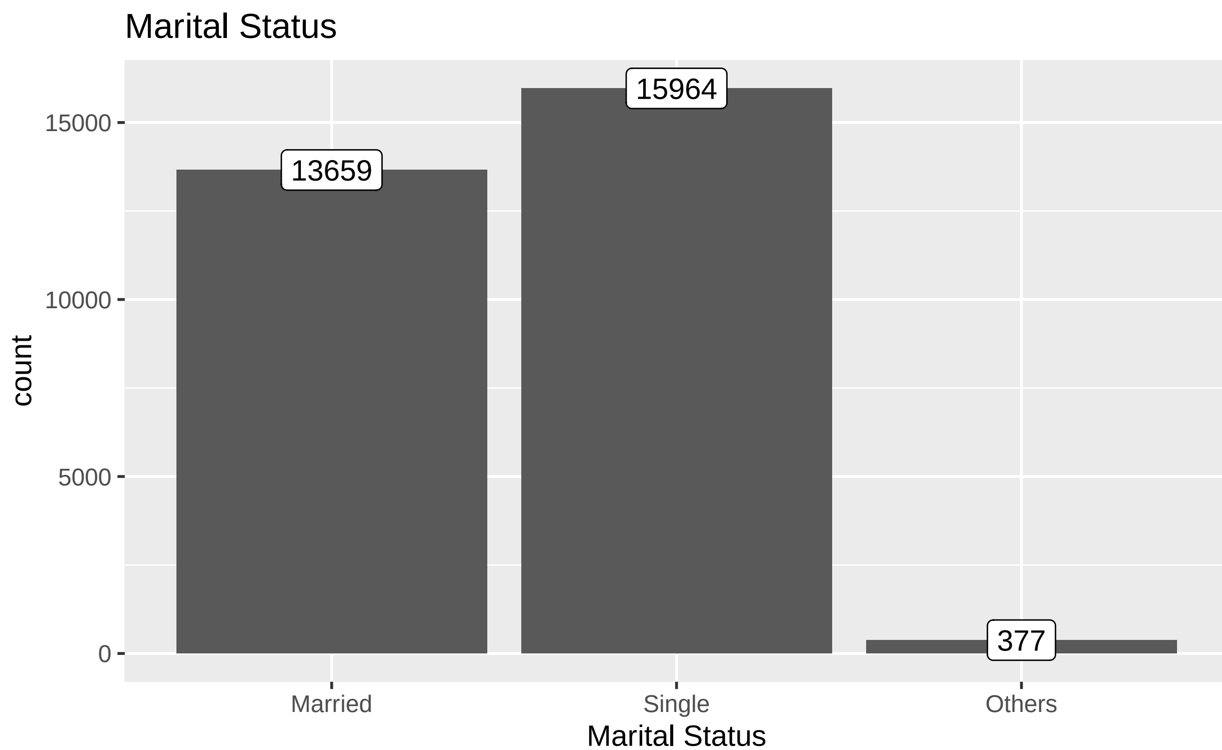
The age distribution across the 30,000 credit card holders is shown below.



From the above histogram, it appears to be positively skewed with most of the credit card holders being younger than 50 years old.

Feature: Marital Status

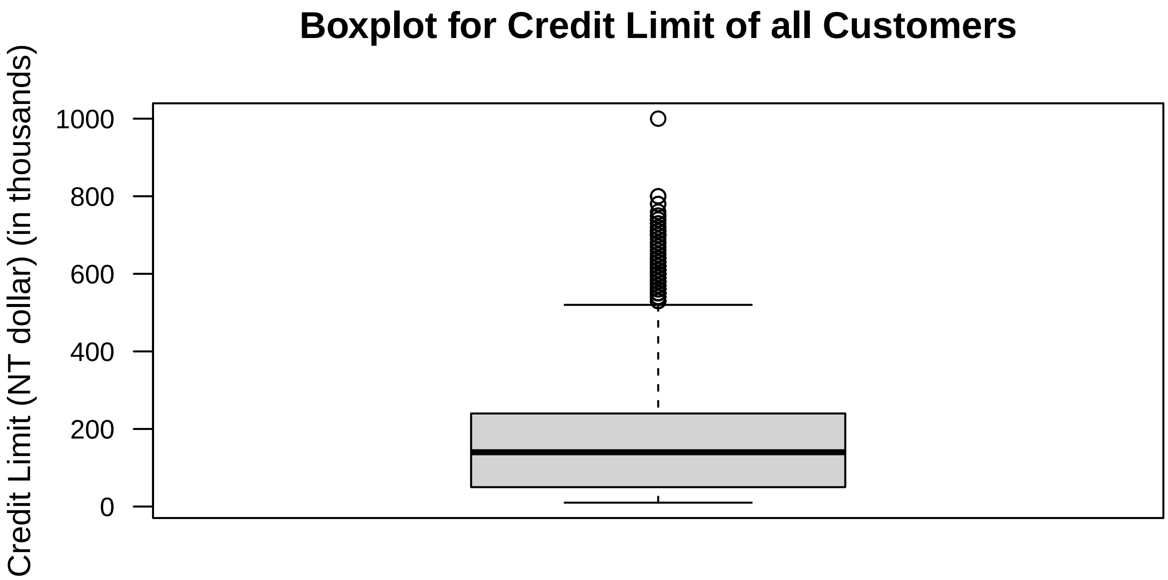
The marital status distribution across the 30,000 credit card holders is shown below.



From the above bar plot, only a few customers have marital status of "Others" while the majority are either "Married" or "Single" with a slightly higher frequency of "Single" customers.

Feature: Credit Limit Balance

The credit limit balance distribution across the 30,000 credit card holders is shown below.



From the above box plot, the median credit limit of the 30,000 customers is approximately TWD160,000 with a few outliers that have a credit limit of approximately TWD550,000 or higher.

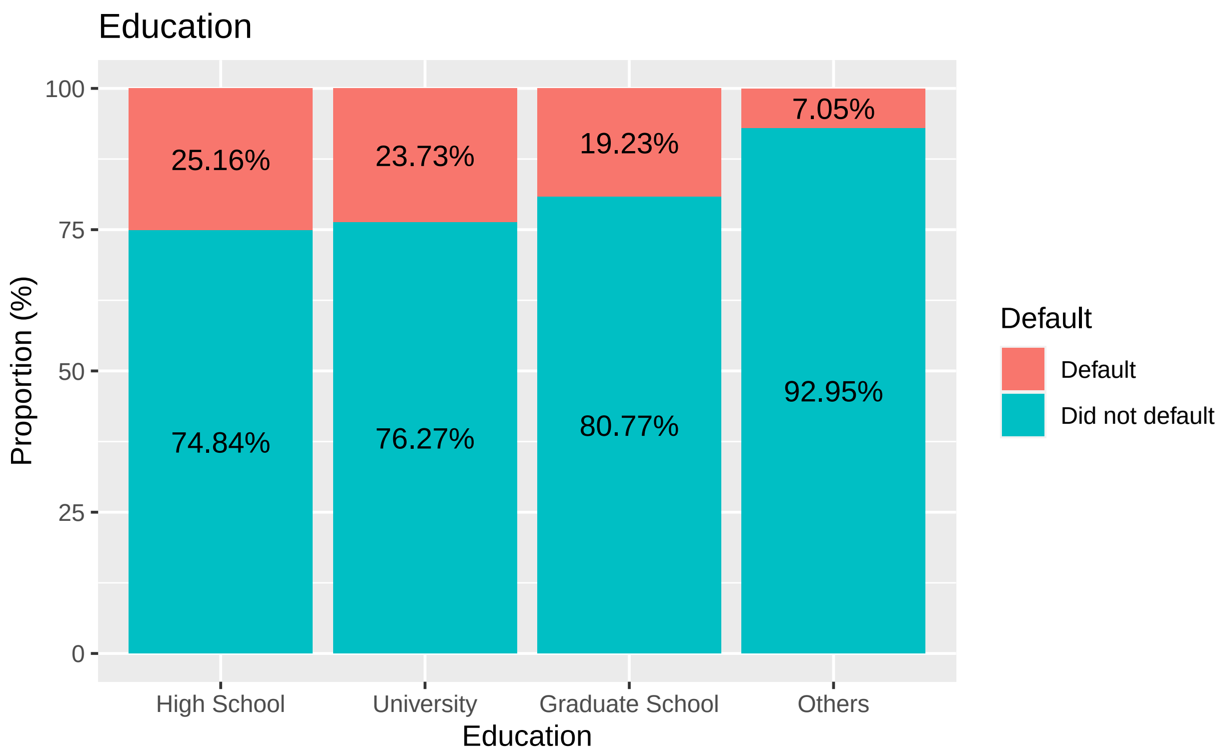
Feature: Gender

The gender distribution across the 30,000 credit card holders is shown below.

From the above stacked bar plot, there is slightly more female than male customers with similar proportion of defaults within each gender group.

Feature: Education

The education distribution across the 30,000 credit card holders is shown below.



The above stacked bar plot shows the default proportion within the individual education level group. From the plot, it appears that customers who have "Others" as their education level has the least proportion of default.

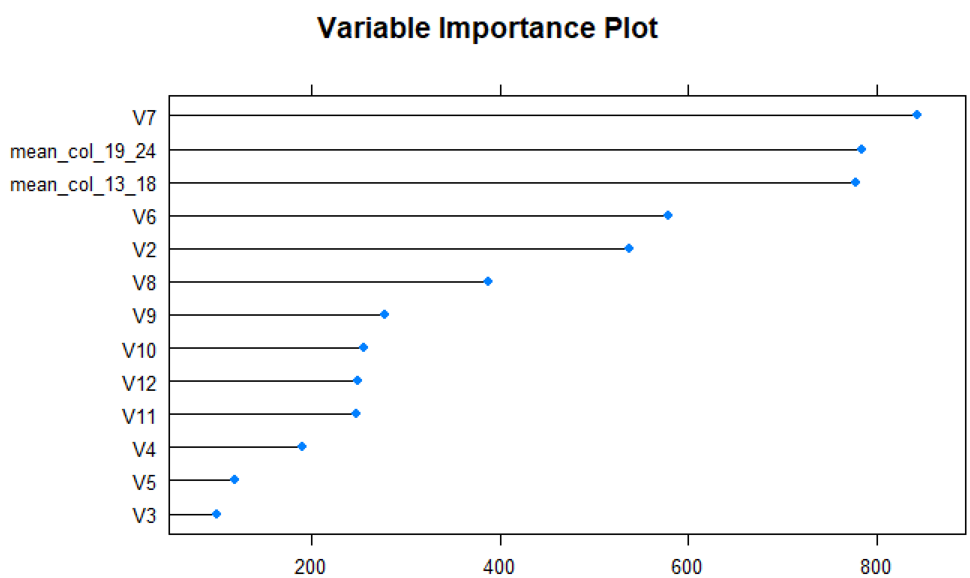
Partioning Data

Prior to the training of the models, the dataset would now be split into 75% training data and 25% testing data.

```
set.seed(1234)
n = length(data$V1)
index <- 1:nrow(data)
testindex <- sample(index, trunc(n)/4)
test.data <- data_MOD[testindex,]
train.data <- data_MOD[-testindex,]
```

Feature Selection and Model Selection

In order to prevent overfitting of the models, it is prudent to find the optimal number of features to build the models such that it is robust and has the ability to generalise. As such, one method to find the optimal number of features to use would be to construct the Variable Importance Plot. This was accomplished by first creating a random forest using 10-fold cross-validation and plotting the variable importance of the random forest.



From the Variable Importance Plot, it can be seen that V7 (Repayment Status in September 2005) has the highest importance value, indicating that V7 should definitely be included in the models. In order to ensure that the models are representative and would not overfit, the features chosen to be included in the models, based on its individual importance, are `V7`, `mean_col_19_24` (the average payment amount), `mean_col_13_18` (the average bill amount payable), `V6` (Age), `V2` (credit limit) and `V8` (Repayment Status in August 2005).

Logistic Regression Model

The first model built to predict whether a customer would default on his or her payments is the logistic regression model. A logistic regression model is appropriate because the target variable is discrete (either default or not default). One benefit of building a logistic regression model is that it is easy to build and train the model. However, a drawback of logistic regression is the assumption that the target variable has a linear relationship with the independent variables.

```
##     actual
## pred    0    1
##    0 4855  754
##    1  977  914
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

Based on the confusion matrix generated by the prediction of the logistic regression model, there are a total of 5,769 correctly classified defaults and non-defaults. More importantly, there are 754 defaulters that were incorrectly classified as non-defaulters.

After running the model, below are the results of the logistic regression model.

Accuracy	Specificity	Area under ROC Curve	F1-Score
0.77	0.48	0.69	0.85

Support Vector Machine

The second model built to predict whether a customer would default on his or her payment is the Support Vector Machine (SVM). Based on the features chosen, the support vector machine was trained using a linear kernel, a cost of 10 and class weights of 0.17 for non-default and 0.83 for default.

```
##      actual
## pred    0    1
##    0 4964  813
##    1  868  855
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

Based on the confusion matrix generated by the prediction of the support vector machine, there are a total of 5,819 correctly classified defaults and non-defaults. More importantly, there are 813 defaulters that were incorrectly classified as non-defaulters.

After running the model, below are the results of the support vector machine.

Accuracy	Specificity	Area under ROC Curve	F1-Score
0.78	0.5	0.68	0.86

Neural Network

The third model built to predict whether a customer would default on his or her payment is the Neural Network. Using the features chosen, the neural network has 6 input neurons, 15 hidden neurons in the hidden layer and 2 output neurons. Additional parameters include a max iteration of 1,000 a decay of 0.01 as well as using entropy (maximum conditional likelihood).

Based on the confusion matrix generated by the prediction of the neural network, there are a total of 5,316 correctly classified defaults and non-defaults. More importantly, there are 930 defaulters that were incorrectly classified as non-defaulters.

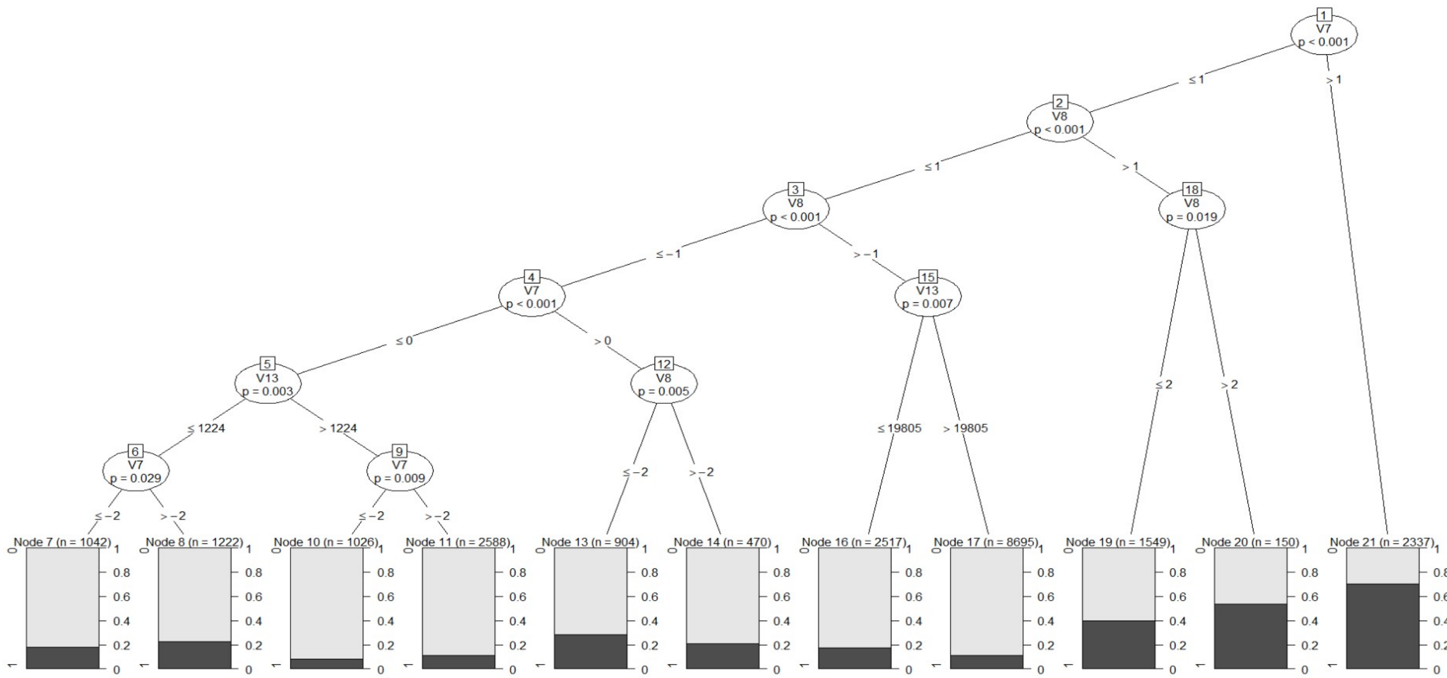
After running the model, below are the results of the neural network.

```
##      actual
## pred    0    1
##    0 5316  930
##    1  516  738
```

Accuracy	Specificity	Area under ROC Curve	F1-Score
0.81	0.59	0.68	0.88

Decision Tree

The last model built to predict whether a customer would default on his or her payment is the Decision Tree. Similar to the previous models, the Decision Tree utilized the 6 features chosen during the feature selection process.




```
##    tree.predict_test
##        0    1
##   0 5598  234
##   1 1159  509

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases
```

Based on the confusion matrix generated by the prediction of the decision tree, there are a total of 5,598 correctly classified defaults and non-defaults. More importantly, there are 234 defaulters that were incorrectly classified as non-defaulters.

After running the model, below are the results of the decision tree.

Accuracy	Specificity	Area under ROC Curve	F1-Score
0.81	0.31	0.63	0.89

Model Evaluation

The metrics employed to evaluate the models are accuracy, specificity, area under ROC curve and F1-score. Due to the data set being unbalanced, Accuracy is not a good metric to compare across the models. Thus, Area under Roc Curve and F1-Score are considered instead, which are better for imbalanced data. Specificity is also used to check for overfitting of the models.

	Accuracy	Specificity	Area under ROC Curve	F1-Score
Logistic Regression	0.77	0.48	0.69	0.85
Support Vector Machine	0.78	0.50	0.68	0.86
Neural Network	0.81	0.59	0.68	0.88
Decision Tree	0.81	0.31	0.63	0.89

Based on the results of the 4 different models using the evaluation metrics selected, it is observed that the neural network model produces better results as compared to the other models.

Improvements

Previously, it is observed that the data set is heavily imbalanced, thus a potential way to improve the model to obtain a better prediction accuracy could be to balance the data set. Oversampling or under sampling can be utilized to introduce a bias to select more samples from one class than from another to obtain a balanced data set. Below, we have run an oversampling and undersampling method to balance the data for demonstration purposes.

Oversampling

```
# #OVERSAMPLING
oversampled_train_data <- ovun.sample(V25 ~ ., data = train.data, method = "over",
                         N =  2*nrow(subset(train.data, train.data$V25 == 0)))$data

table(oversampled_train_data$V25)
```

```
##
##     0     1
## 17532 17532
```

Undersampling

```
# UNDERSAMPLING
undersampled_train_data <- ovun.sample(V25 ~ ., data = train.data, method = "under",
                         N =  2*nrow(subset(train.data, train.data$V25 == 1)))$data

table(undersampled_train_data$V25)
```

```
##
##    0    1
## 4968 4968
```