BT2101 Group Project

Valary Lim Wan Qian (A0190343L), Lai Yan Jean (A0190326J), Mitchell Kwong (A0182695N),
Lee Jing Xuan (A0189467H), Koay Tze Min (A0188851N)

1. Brief Introduction

1.1. Dataset

card.csv contains payment data of 30,000 credit card holders from a Taiwanese bank. We derived the below feature information from UCI's repository, so it does not account for data anomalies.

Feature		Column	Type	Description
X1	Amount of given credit (NT dollar)	LIMIT_BAL	Continuous	Includes both the individual consumer credit and his/her family (supplementary) credit
X2	Gender	SEX	Categorical	Levels: 1 = Male; 2 = Female
X3	Education	EDUCATION	Categorical	Levels: 1 = Graduate school; 2 = University; 3 = High school; 4 = Others
X4	Marital Status	MARRIAGE	Categorical	Levels: 1 = Married; 2 = Single; 3 = Others
X5	Age	AGE	Continuous	Age of customer in years

X6-11	History of monthly payment records (Apr - Sept 2005)	PAY_0
		PAY_2
		...
		PAY_6
			Continuous	X6 = Repayment status in September 2005;
				X7 = Repayment status in August 2005; …
				X11 = Repayment status in April 2005
				
				Levels:
				-1 = Pay duly;
				1 = Payment delay for 1 month;
				2 = Payment delay for 2 months; …
				9 = Payment delay for 9 months and above

X12-17	Amount of bill statement (NT dollar)	BILL_AMT1 ... BILL_AMT6	Continuous	X12 = amount of bill statement in September 2005; X13 = amount of bill statement in August 2005; ... X17 = amount of bill statement in April 2005

X18-23	Amount of previous payment (NT dollar)	PAY_AMT1
		...
		PAY_AMT6
			Continuous	X18 = amount paid in September 2005;
				X19 = amount paid in August 2005; ...
				X23 = amount paid in April 2005.

Target	Default	DEFAULT	Binary	Levels: 0 = Not default; 1 = Default

1.2. Problem

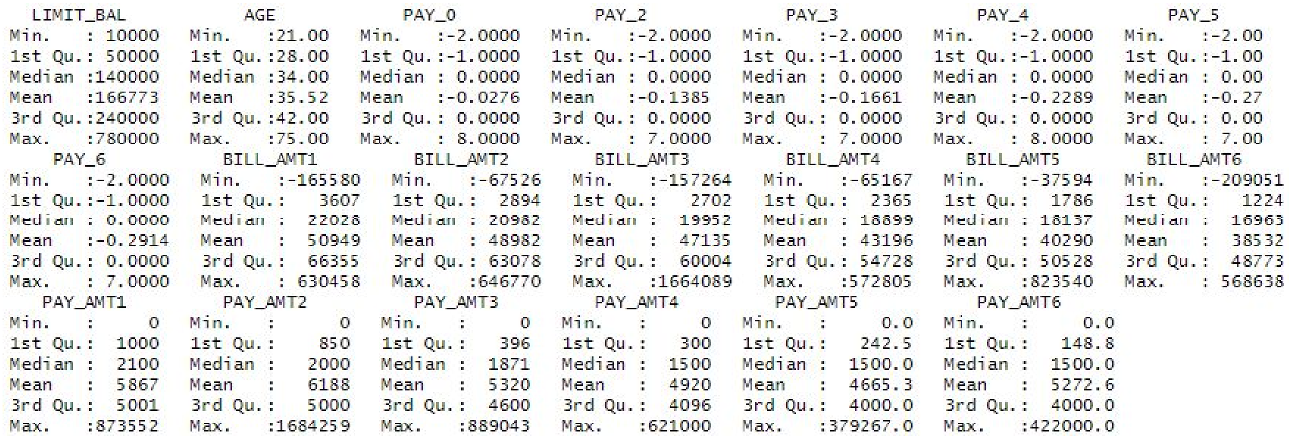
To reduce risk, banks must distinguish between credible and non-credible customers. Thus, we wish to develop classification models and find the optimal one that predicts a new customer's likelihood of defaulting on payments and thereby classify them as credible or not. This is a binary classification problem, where the target class' value is either Default (1) or Not Default (0).

2. Exploratory Data Analysis

We split the cards dataset into training and test sets, which forms 1/3 and 2/3 of the original data respectively. Data exploration, cleaning and model training are done on the former only. The latter remains unseen by the models until our final model evaluation step.

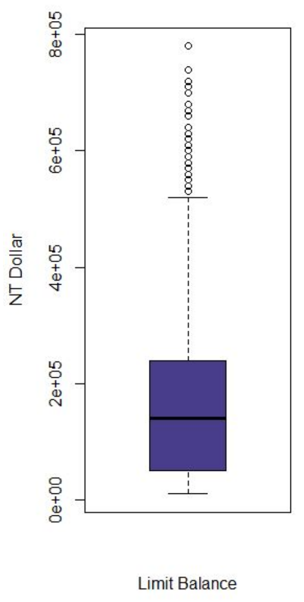
2.1. Continuous Data

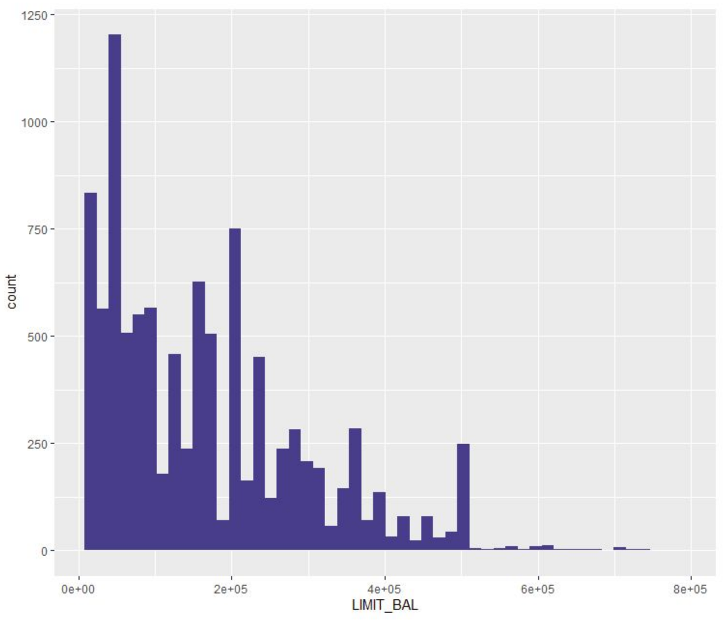
Here, we checked for anomalies, conducted hypothesis tests and correlation analyses. After ensuring that the data contains no missing values, we summarised the continuous attributes, which yielded the following results:



```
   LIMIT_BAL          AGE            PAY_0              PAY_2              PAY_3              PAY_4              PAY_5      
 Min.   : 10000   Min.   :21.00   Min.   :-2.0000   Min.   :-2.0000   Min.   :-2.0000   Min.   :-2.0000   Min.   :-2.00  
 1st Qu.: 50000   1st Qu.:28.00   1st Qu.:-1.0000   1st Qu.:-1.0000   1st Qu.:-1.0000   1st Qu.:-1.0000   1st Qu.:-1.00  
 Median :140000   Median :34.00   Median : 0.0000   Median : 0.0000   Median : 0.0000   Median : 0.0000   Median : 0.00  
 Mean   :166773   Mean   :35.52   Mean   :-0.0276   Mean   :-0.1385   Mean   :-0.1661   Mean   :-0.2289   Mean   :-0.27  
 3rd Qu.:240000   3rd Qu.:42.00   3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 0.00  
 Max.   :780000   Max.   :75.00   Max.   : 8.0000   Max.   : 7.0000   Max.   : 7.0000   Max.   : 8.0000   Max.   : 7.00  
     PAY_6           BILL_AMT1         BILL_AMT2         BILL_AMT3          BILL_AMT4         BILL_AMT5         BILL_AMT6      
 Min.   :-2.0000   Min.   :-165580   Min.   :-67526   Min.   :-157264   Min.   :-65167   Min.   :-37594   Min.   :-209051  
 1st Qu.:-1.0000   1st Qu.:   3607   1st Qu.:  2894   1st Qu.:   2702   1st Qu.:  2365   1st Qu.:  1786   1st Qu.:   1224  
 Median : 0.0000   Median :  22028   Median : 20982   Median :  19952   Median : 18899   Median : 18137   Median :  16963  
 Mean   :-0.2914   Mean   :  50949   Mean   : 48982   Mean   :  47135   Mean   : 43196   Mean   : 40290   Mean   :  38532  
 3rd Qu.: 0.0000   3rd Qu.:  66355   3rd Qu.: 63078   3rd Qu.:  60004   3rd Qu.: 54728   3rd Qu.: 50528   3rd Qu.:  48773  
 Max.   : 7.0000   Max.   : 630458   Max.   :646770   Max.   :1664089   Max.   :572805   Max.   :823540   Max.   : 568638  
    PAY_AMT1         PAY_AMT2          PAY_AMT3         PAY_AMT4         PAY_AMT5           PAY_AMT6       
 Min.   :     0   Min.   :      0   Min.   :     0   Min.   :     0   Min.   :     0.0   Min.   :     0.0  
 1st Qu.:  1000   1st Qu.:    850   1st Qu.:   396   1st Qu.:   300   1st Qu.:   242.5   1st Qu.:   148.8  
 Median :  2100   Median :   2000   Median :  1871   Median :  1500   Median :  1500.0   Median :  1500.0  
 Mean   :  5867   Mean   :   6188   Mean   :  5320   Mean   :  4920   Mean   :  4665.3   Mean   :  5272.6  
 3rd Qu.:  5001   3rd Qu.:   5000   3rd Qu.:  4600   3rd Qu.:  4096   3rd Qu.:  4000.0   3rd Qu.:  4000.0  
 Max.   :873552   Max.   :1684259   Max.   :889043   Max.   :621000   Max.   :379267.0   Max.   :422000.0  
```

50% of the customers are youths, between 21 and 34 years old (inclusive). Furthermore, the means and medians of many attributes, particularly LIMIT_BAL, BILL_AMT1 to BILL_AMT6 and PAY_AMT1 to PAY_AMT6, are significantly different. This suggests skewness in the distributions of these columns, which we confirmed by plotting their values on boxplots and histograms. The diagrams for LIMIT_BAL are shown below as an example.

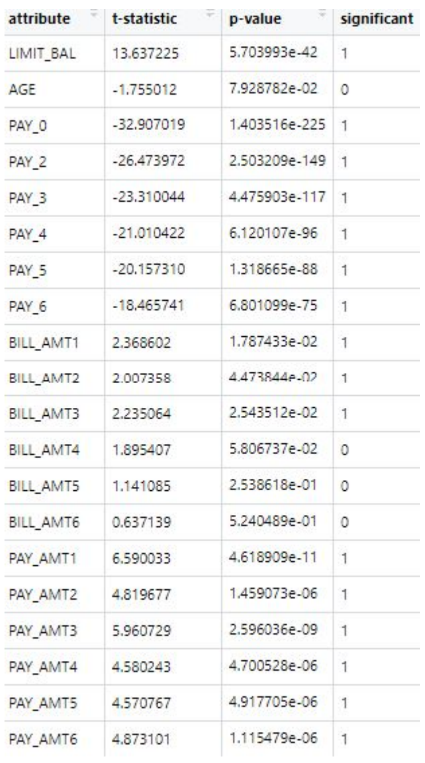




The boxplots suggest many outliers exist in these continuous data. However, considering these attributes describe credit amount of customers, we expect a large degree of variation. Hence, we considered the outliers significant to the study and did not remove them. The histograms also shows that the data tends to be positively skewed. This means a significant proportion of customers has lower limit balance, payment amount and balance statements.

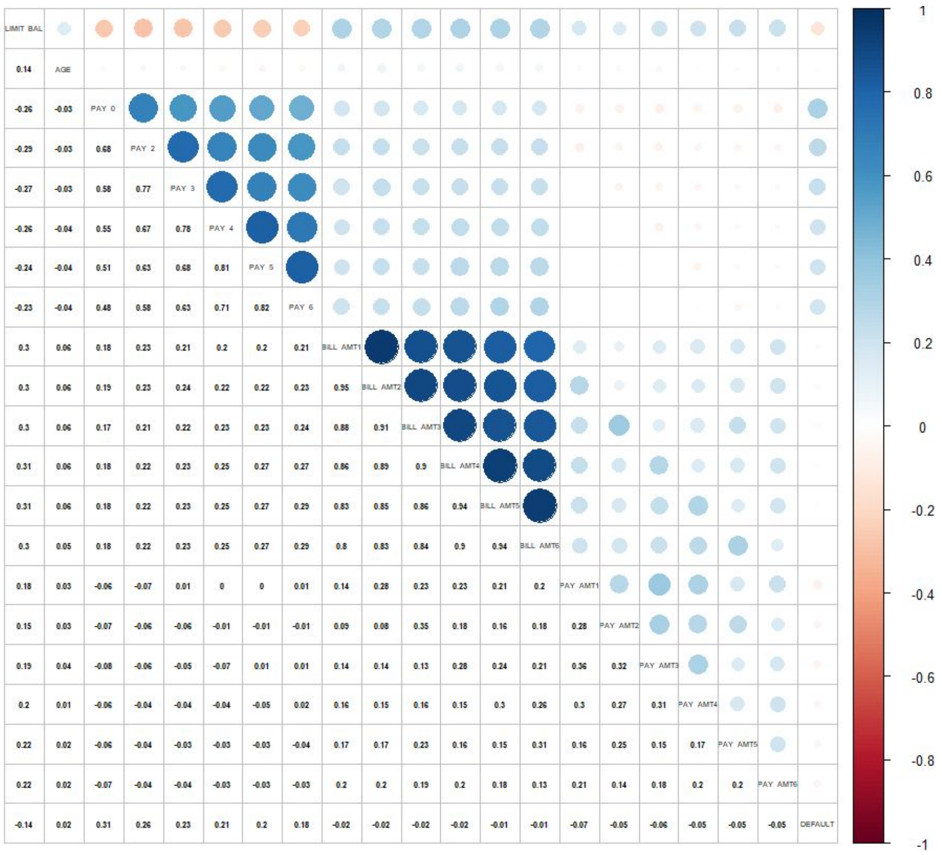
We conducted Students' t-tests to check which continuous variables are likely significant in predicting the target variable. The t-statistic and the p-value at 5% significance level are shown in the table on the right.

The table suggests that all attributes except for AGE, BILL_AMT4, BILL_AMT5 and BILL_AMT6 are significant. The significant features are likely to be used by our model to predict customers' default patterns.



attribute	t-statistic	p-value	significant
LIMIT_BAL	13.637225	5.703993e-42	1
AGE	-1.755012	7.928782e-02	0
PAY_0	-32.907019	1.403516e-225	1
PAY_2	-26.473972	2.503209e-149	1
PAY_3	-23.310044	4.475903e-117	1
PAY_4	-21.010422	6.120107e-96	1
PAY_5	-20.157310	1.318665e-88	1
PAY_6	-18.465741	6.801099e-75	1
BILL_AMT1	2.368602	1.787433e-02	1
BILL_AMT2	2.007358	4.473844e-02	1
BILL_AMT3	2.235064	2.543512e-02	1
BILL_AMT4	1.895407	5.806737e-02	0
BILL_AMT5	1.141085	2.538618e-01	0
BILL_AMT6	0.637139	5.240489e-01	0
PAY_AMT1	6.590033	4.618909e-11	1
PAY_AMT2	4.819677	1.459073e-06	1
PAY_AMT3	5.960729	2.596036e-09	1
PAY_AMT4	4.580243	4.700528e-06	1
PAY_AMT5	4.570767	4.917705e-06	1
PAY_AMT6	4.873101	1.115479e-06	1

We also examined correlations between the continuous variables and the target, which are illustrated in the matrix below.



LIMIT BAL																				
0.14	AGE																			
-0.26	-0.03	PAY 0																		
-0.29	-0.03	0.68	PAY 2																	
-0.27	-0.03	0.58	0.77	PAY 3																
-0.26	-0.04	0.55	0.67	0.78	PAY 4															
-0.24	-0.04	0.51	0.63	0.68	0.81	PAY 5														
-0.23	-0.04	0.48	0.58	0.63	0.71	0.82	PAY 6													
0.3	0.06	0.18	0.23	0.21	0.2	0.2	0.21	BILL AMT1												
0.3	0.06	0.19	0.23	0.24	0.22	0.22	0.23	0.95	BILL AMT2											
0.3	0.06	0.17	0.21	0.22	0.23	0.23	0.24	0.88	0.91	BILL AMT3										
0.31	0.06	0.18	0.22	0.23	0.25	0.27	0.27	0.86	0.89	0.9	BILL AMT4									
0.31	0.06	0.18	0.22	0.23	0.25	0.27	0.29	0.83	0.85	0.86	0.94	BILL AMT5								
0.3	0.05	0.18	0.22	0.23	0.25	0.27	0.29	0.8	0.83	0.84	0.9	0.94	BILL AMT6							
0.18	0.03	-0.06	-0.07	0.01	0	0	0.01	0.14	0.28	0.23	0.23	0.21	0.2	PAY AMT1						
0.15	0.03	-0.07	-0.06	-0.06	-0.01	-0.01	-0.01	0.09	0.08	0.35	0.18	0.16	0.18	0.28	PAY AMT2					
0.19	0.04	-0.08	-0.06	-0.05	-0.07	0.01	0.01	0.14	0.14	0.13	0.28	0.24	0.21	0.36	0.32	PAY AMT3				
0.2	0.01	-0.06	-0.04	-0.04	-0.04	-0.05	0.02	0.16	0.15	0.16	0.15	0.3	0.26	0.3	0.27	0.31	PAY AMT4			
0.22	0.02	-0.06	-0.04	-0.03	-0.03	-0.03	-0.04	0.17	0.17	0.23	0.16	0.15	0.31	0.16	0.25	0.15	0.17	PAY AMT5		
0.22	0.02	-0.07	-0.04	-0.04	-0.03	-0.03	-0.03	0.2	0.2	0.19	0.2	0.18	0.13	0.21	0.14	0.18	0.2	0.2	PAY AMT6	
-0.14	0.02	0.31	0.26	0.23	0.21	0.2	0.18	-0.02	-0.02	-0.02	-0.02	-0.01	-0.01	-0.07	-0.05	-0.06	-0.05	-0.05	-0.05	DEFAULT

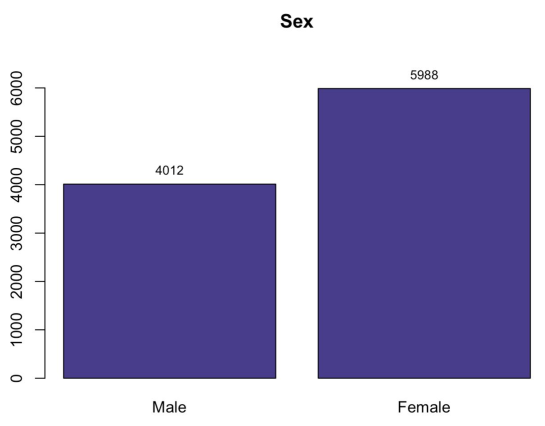
The right-most column of the matrix indicates that most attributes have low linear correlations with the DEFAULT variable. PAY_0 has the most correlation at 0.31, followed by LIMIT_BAL at -0.14. Hence, a linear regression model might not be the most useful for accurate classification.

Strong correlations exist within the PAY_0 to PAY_6 attributes, and within the BILL_AMT1 to BILL_AMT6 attributes. This is good to keep in mind especially for the naive bayes model that assumes independence between different attributes.

We chose not to perform feature selection using the filter method, i.e. remove the highly correlated features in the data-preprocessing stage. This is because our dataset contains relatively few features, so we did not want to have any loss of information by removing features before the model fitting stage.

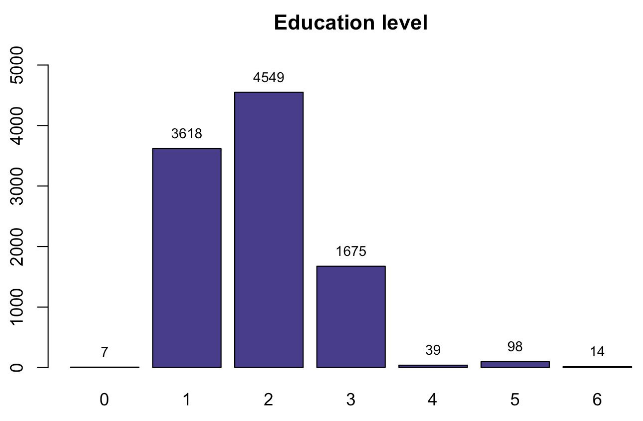
2.2 Discrete Data

We plotted bar charts of each discrete attribute to understand the distributions between its categories. We conducted Chi-squared tests as well to check for significant associations between each attribute and the target variable.



X-squared = 11.235
df = 1
p-value = 0.0008026

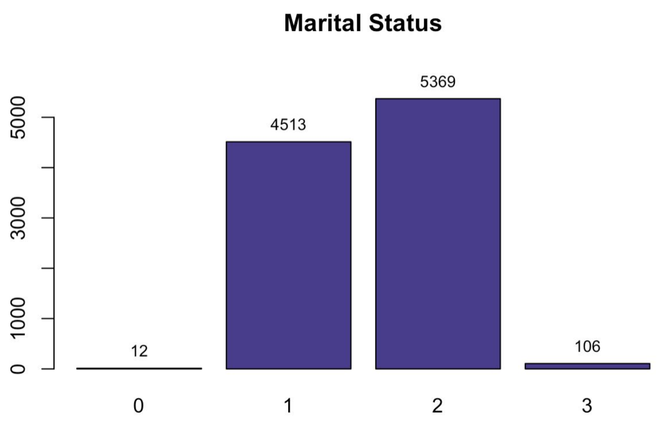
Since the p-value is less than the significance level of 0.05, the two attributes gender and default status are associated and should be included in the modelling when predicting the default status.



X-squared = 46.379
df = 6
p-value = 2.489e-08

Since the p-value is less than the significance level of 0.05, the two attributes education level and default status are associated and should be included in the modelling when predicting the default status.

From the bar chart, some of the data are categorised into class 0, 5 and 6 which are not listed in the data description given and are of unknown categories. We will need to account for this unknown data in the data pre-processing stage.

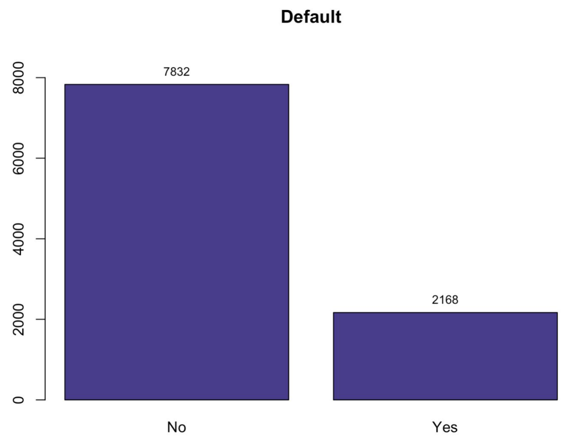


X-squared = 8.0894
df = 3
p-value = 0.0442

Since the p-value is less than the significance level of 0.05, the two attributes marital status and default status are associated and should be included in the modelling when predicting the default status.

From the bar chart, we also notice that some of the marital status data are categorised into class 0 which is not listed in the data description given and are of unknown categories. We will need to account for this unknown data in the data pre-processing stage.

2.3. Target Variable



The dataset is imbalanced. No Default (0) is the majority class, with 7852 observations making up approximately 78.3% of the samples. Default (1) is the minority class, with 2148 observations.

Models that are created from this imbalanced dataset may be biased toward predicting samples as "No Default". Thus, we will have to balance the number of samples of "Default" and "No Default" in the train dataset to ensure that our models remain unbiased.

3. Data Pre-Processing

3.1. Splitting the Data

To train our model, we carried out 10-fold cross validation across the training set and computed the selected validation metrics. K-fold cross validation was chosen over hold-out sampling as the former allows models to be trained on several different training datasets from different train-validation splits. In contrast, hold-out sampling relies on only one split, which causes the validation score to be less reliable and consistent. Thus, k-fold cross validation is better able to signal to us the performance of our model on the unseen test data sets.

3.2. Data Cleaning

The data points labelled 0, 5 and 6 for EDUCATION and 0 in MARRIAGE are unexplained in the UCI repository. For these variables, we conducted our analysis based on the train dataset, and decided to group the unknown data into a separate category. We performed the same data pre-processing steps on our test set.

3.3. Encoding Variables

We chose one-hot encoding our categorical variables instead of integer encoding. The use of integer encoding assumes that the variables have a natural ordering following that of integers. This means that the variables would be treated as ordinal variables. However, variables such as MARRIAGE are categorical nominal variables, making one-hot encoding more suitable.

3.4. Normalizing Numeric Variables

Normalization is used to standardize each variable onto a common scale. We considered min-max normalization and z-score normalization. However, our exploratory data analysis revealed the continuous variables contain significant outliers. Thus, we chose to use z-score normalization over min-max normalization for all continuous variables.

Normalization was carried out after the train-test split. The mean and standard deviation obtained from the train data was used to scale both train and test sets. This was to prevent information leakage between the train and test set.

For PAY_0 to PAY_6, we observed that the data can be further split into a categorical variable that reflects the status of the payment {>0, 0, -1, -2} and a discrete numerical variable that reflects the duration of the delay if any [1, 9]. The categorical variable was one-hot encoded and the numerical variable was z-score normalized.

3.5. Balancing the Data by Oversampling

As observed from the exploratory data analysis, the training dataset is unbalanced with the "No Default" class taking up more than 75% of the training dataset. A balanced proportion of target values is necessary to reduce bias in machine learning models, such as neural network, that work based on pattern recognition.

We chose to balance the training data by oversampling. Oversampling is done by a random selection of observations in the minority class that are duplicated up to the case where both the "Default" and "No Default" classes contain an equal number of observations. We chose to oversample rather than undersample as the latter omits observations from the majority class and this may result in the loss of important data. The test data was left unbalanced.

4. Feature and Model Selection

4.1. Validation Metrics

Precision is the proportion of customers correctly classified as default, out of all customers predicted as default. Recall is the proportion of customers correctly classified as default, out of all customers that are actually default. We prioritised these two metrics. Achieving a high recall is important for detecting a larger proportion of defaulters amongst all the actual defaulters. High precision is also important to ensure that minimal non-defaulters are wrongly categorised into the "Default" category. This would reduce the company's loss of profits.

Hence, we selected F1 statistic as our main validation metric. It ranges from 0 to 1, where a value closer to 1 indicates a better model. Using the F1 score penalises models that only do well in one of the categories (either precision or recall), and it gives a weighted average of the precision and recall, since it is defined by $F1 = 2 \times \frac{Precision * Recall}{Precision + Recall}$.

4.2. Model Selection

The following subsections outline the models we tested and the results of model fitting. Feature selection is done within the building of each model itself when necessary (wrapper or embedded method).

4.2.1. Logistic Regression

Overview
Logistic regression is a linear classification technique for categorical target variables. It classifies the observations into classes based on the posterior probability of the target variable conditioned to the input variables. This contrasts with the linear regression model, which is used if the dependent variable is continuous. In this case, the target variable is categorical - and more specifically, binary - so the use of logistic regression model is suitable.

Strengths and Limitations
Logistic regression is easily interpretable and effective in classifying linearly separable data. Additionally, the logistic regression model can be easily regularised to prevent overfitting.

However, for the model to be of value, the observations should be independent of one another. If observations are dependent, relationship between the target and independent variables will be overstated. Thus, the logistic regression model was trained on the train data without oversampling. The use of imbalanced dataset can cause the probability values to be skewed but this can be mitigated by using the `optimalCutoff` function which gives us the best cut off points for categorising. Thus, we need not worry about the imbalanced train data used to fit the model.

Model Fitting
We used the `glm` function to obtain the logistic regression model. We tried 2 different feature selection techniques and compared the mean f1 scores obtained from the k-fold cross validation to find the best model.

The first feature selection method we employed is the backward feature elimination using p-values. The p-value of each feature is observed and the maximum p-value that is above the significance level of 0.05 will be iteratively removed. The final model will only consist of features with p-value < 0.05 as a p-value < 0.05 implies the feature is a significant predictor of the target variable.

The second method used is the backward stepwise selection using `stepAIC` function. This function selects the relevant independent variables by retrieving the model that gives the lowest Akaike's Information Criteria (AIC) value. AIC is used to assess the quality of the model through comparison with related models in order to retrieve the optimal one. It measures information loss and thereby the trade-off between model accuracy and model simplicity. It also reduces model overfitting or underfitting so that it better predicts the new data inputs from the test set.

1. We first build a full model using all the given features.
2. Cross validate the model using 10-fold cross validation. Within each fold, we obtain the optimal cut off point using the train data and use the cut off value obtained to predict which class each observation in the validation set belongs to.
3. The optimal cut off is retrieved by calling the function `optimalCutoff` taking in the input `optimiseFor="Both"` rather than the `"misclasserror"` as the dataset is imbalanced and we are trying to optimise recall and precision. `optimiseFor="Both"` maximises the Youden's Index which maximises the sensitivity and specificity. Optimising for misclassification error with an imbalanced data set can cause the optimal cutoff to be skewed in favour of classifying into the majority class and hence affecting the F1 score. The optimal cut off is then used to classify the dataset into two categories "default" and "non-default".
4. We then repeat steps 1-3 using the "AIC" model which was built using the `stepAIC` function and the "Iterative" model that was constructed through feature selection using p-values. The following are the results of the cross validation.

```
           Mean F1 score
Full model     0.5082303
AIC            0.5112908
Iterative      0.5106545
```

5. From the above result, we observe that the AIC model gives the highest mean F1 score of 0.511. The AIC model is selected as the final logistic regression model. We trained the final model using the full train set and obtain the optimal cut off value.

Results and Discussion

The following are the coefficients and the relevant features in predicting whether a customer will fall under the class of "default" or "non-default" after running the logistic regression model.

```
Coefficients:
                     Estimate Std. Error z value Pr(>|z|)
(Intercept)          -0.14498    0.42022  -0.345 0.730083
LIMIT_BAL            -0.22035    0.03782  -5.826 5.66e-09 ***
SEX_male              0.12308    0.05557   2.215 0.026769 *
EDUCATION_gradSch     0.89908    0.32554   2.762 0.005748 **
EDUCATION_university  0.91509    0.32426   2.822 0.004772 **
EDUCATION_highSch     0.99440    0.32844   3.028 0.002465 **
MARRIAGE_married      0.21688    0.05617   3.861 0.000113 ***
PAY_0_unknown0       -1.02419    0.12459  -8.220  < 2e-16 ***
PAY_0_unknown2       -0.28968    0.15069  -1.922 0.054559 .
PAY_0_value           0.43479    0.04051  10.733  < 2e-16 ***
PAY_2_unknown0       -0.59023    0.26775  -2.204 0.027496 *
PAY_2_unknown2       -1.05205    0.29732  -3.538 0.000402 ***
PAY_2_payDuly        -1.06069    0.26243  -4.042 5.31e-05 ***
PAY_2_value          -0.35897    0.09307  -3.857 0.000115 ***
PAY_3_unknown0       -0.44166    0.10903  -4.051 5.11e-05 ***
PAY_3_unknown2       -0.36638    0.21341  -1.717 0.086021 .
PAY_3_payDuly        -0.50049    0.14617  -3.424 0.000617 ***
PAY_4_unknown0       -0.66152    0.24414  -2.710 0.006736 **
PAY_4_unknown2       -0.66536    0.30489  -2.182 0.029090 *
PAY_4_payDuly        -0.65959    0.26265  -2.511 0.012028 *
PAY_4_value          -0.20368    0.07873  -2.587 0.009683 **
PAY_5_unknown0       -0.38627    0.12308  -3.138 0.001699 **
PAY_5_unknown2       -0.44681    0.22106  -2.021 0.043253 *
PAY_5_payDuly        -0.45942    0.15981  -2.875 0.004042 **
PAY_6_unknown2        0.29679    0.15265   1.944 0.051867 .
PAY_6_payDuly         0.20956    0.11167   1.877 0.060580 .
PAY_6_value           0.15170    0.03605   4.208 2.58e-05 ***
BILL_AMT2             0.15285    0.03917   3.902 9.52e-05 ***
PAY_AMT1             -0.20146    0.06680  -3.016 0.002563 **
PAY_AMT2             -0.17335    0.08715  -1.989 0.046687 *
PAY_AMT3             -0.12364    0.06182  -2.000 0.045488 *
```

4.2.2. Decision Tree

Overview

The decision tree model is a sequential model that starts with all the data in the root node, before carrying out a series of classifications to split the data on a certain feature. Internal nodes represent the splitting attribute, the branches represent the outcome of the split, and all leaf nodes represent the outcome - default (1) or non-default (0).

Strengths and Limitations

The decision tree model was considered as it creates a model that is simple for the bank executives to understand and make use of quickly when assessing a new customer's likelihood of default on their debt. The decision tree employs a greedy algorithm - at every splitting stage of the decision tree, the variable which leads to the most optimal split will be selected.

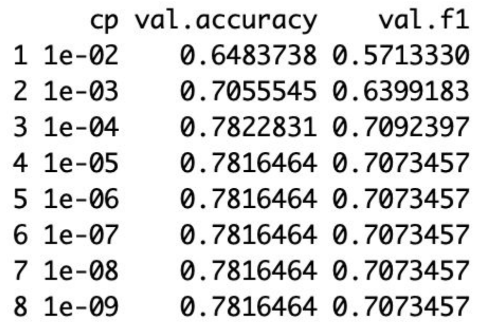
However, decision trees may overfit the noise in the data and thus may not be generalisable to other data points. The tree may not be stable, and a small change in a variable value could change the structure of the optimal decision tree. Decision trees have also a high degree of variability between different data samples from the same dataset.

Model Fitting

The R package "rpart" (Recursive Partitioning) was used to create the decision tree. The decision rules in this package are generated based on CART (Classification & Regression Trees) modelling. Several variations of the classification tree was generated by changing the parameters of the function.

1. We first plotted the simplest classification tree, with only one split using PAY_0 value as the splitting variable. We obtained an F1 score of 62.29% for the train set, and 51.32% for the test set. In order to find a better model, we ran the algorithm using 7 other values for the complexity control parameter (cp). Within each iteration, we did a 10-fold cross validation and calculated the accuracy and F1 scores. The final CP selected to train the final model and build the classification tree will be the one with the highest F1 score.
2. After the tree was built using an optimal cp value on the entire train set, the tree was pruned so that the model can be applied on the test set. The classification tree will be evaluated along with other models based on metrics calculated for both the train and test sets.

Results and Discussion



```
     cp val.accuracy    val.f1
1 1e-02    0.6483738 0.5713330
2 1e-03    0.7055545 0.6399183
3 1e-04    0.7822831 0.7092397
4 1e-05    0.7816464 0.7073457
5 1e-06    0.7816464 0.7073457
6 1e-07    0.7816464 0.7073457
7 1e-08    0.7816464 0.7073457
8 1e-09    0.7816464 0.7073457
```

After building the classification tree using different values for cp, we found that after a certain threshold ($cp = 1 \times 10^{-4}$), the accuracy and F1 score of the validation set remains constant i.e. increasing the complexity of the tree further will not be useful in achieving better predictions. The final cp value we chose to train the entire train.data on was 0.0001 which had the highest validation F1 and accuracy.

4.2.3. Random Forest

Overview
Random Forest comprises many decision trees that work together as an ensemble. Each individual tree will output a class prediction, and the most voted class will be the model's final prediction.

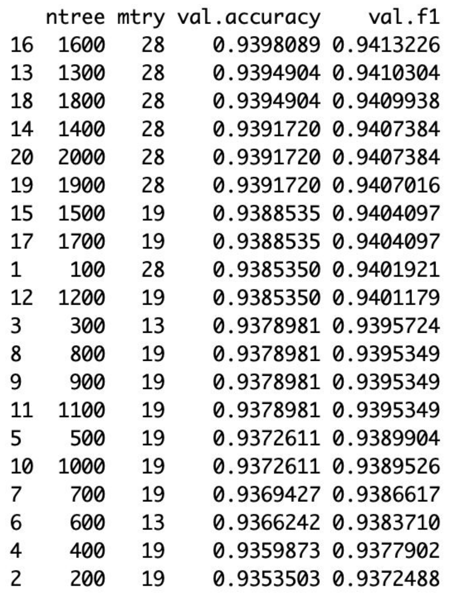
Strengths and Limitations
The Random Forest was considered as it uses the wisdom of crowds, usually resulting in more accurate model predictions. The Random Forest Model will also reduce the variance that causes errors in decision trees, by taking the most voted class as the final prediction. Overfitting on the entire train set is also reduced, as the different decision trees are trained based on different subsets of the data, and different subsets of features.

However, the Random Forest is not very interpretable - they are like "black boxes" and users will not find out how exactly the model does the classification. If the bank wishes to know how exactly the model classifies their customers, Random Forest may not be the most appropriate.

Model Fitting
The R package "randomForest" was used to create the Random Forest model. The package implements Breiman's random forest algorithm.

1. We first created a Random Forest model using the default settings (mtry = 6; ntree = 500), and obtained an accuracy of 80.2% and F1 score of 50.7% on the test set. Importance was set to TRUE so that the importance of predictors in the model will be assessed.
2. We then split the initial train data into a train-validation set to select the most optimal values for mtry (number of variables randomly sampled as candidates at each split) and ntree (number of trees to grow).
3. We tuned the model using the tuneRF() function in the randomForest package, that finds the optimal mtry values. The initial model used the default value of mtry = sqrt(p), where p is the number of variables in the data.
4. We tested different values of ntree to find an optimal number that gives the best metrics on the validation set. The initial model used the default value of ntree = 500.

Results and Discussion



```
   ntree mtry val.accuracy    val.f1
16  1600   28    0.9398089 0.9413226
13  1300   28    0.9394904 0.9410304
18  1800   28    0.9394904 0.9409938
14  1400   28    0.9391720 0.9407384
20  2000   28    0.9391720 0.9407384
19  1900   28    0.9391720 0.9407016
15  1500   19    0.9388535 0.9404097
17  1700   19    0.9388535 0.9404097
1    100   28    0.9385350 0.9401921
12  1200   19    0.9385350 0.9401179
3    300   13    0.9378981 0.9395724
8    800   19    0.9378981 0.9395349
9    900   19    0.9378981 0.9395349
11  1100   19    0.9378981 0.9395349
5    500   19    0.9372611 0.9389904
10  1000   19    0.9372611 0.9389526
7    700   19    0.9369427 0.9386617
6    600   13    0.9366242 0.9383710
4    400   19    0.9359873 0.9377902
2    200   19    0.9353503 0.9372488
```

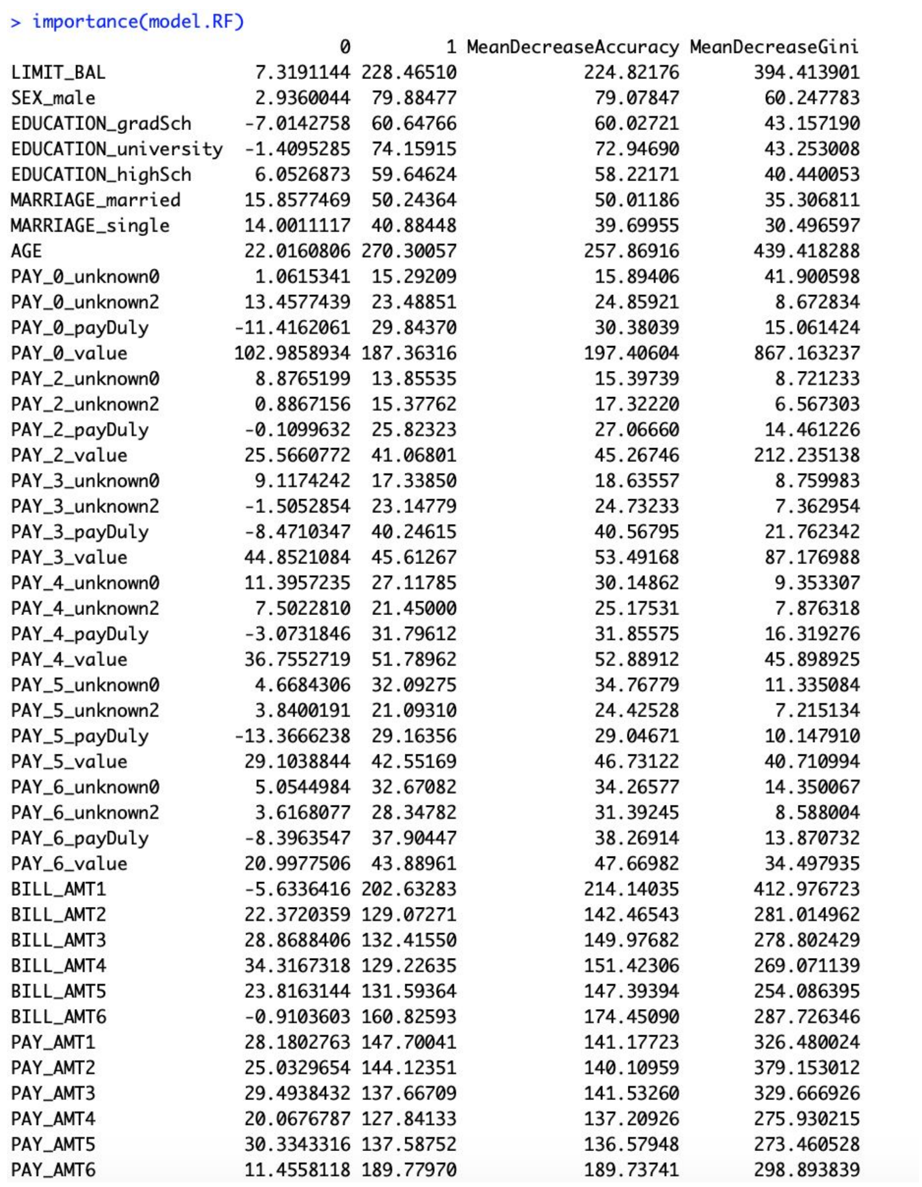
The optimal ntree and mtry is **1600** and **28** respectively as these parameters gave the highest accuracy and F1 scores.

We then built a random forest model with these parameters on the full train dataset. This model served as our final model for evaluation.

By using the `importance` function in the “randomForest” R package, we can view the relative importance of different features in our random forest model. The relative importance of features could be useful in giving the bank further insights into the features that have greater predictive value in determining which customers are more likely to default on their payments.

The table below reflects the relative importance of each feature in our random forest model. We can focus on two columns, the MeanDecreaseAccuracy and MeanDecreaseGini columns. The MeanDecreaseAccuracy reflects how much the accuracy decreases when the variable is excluded. A higher mean decrease accuracy score therefore suggests that the feature is more important. The MeanDecreaseGini column measures the decrease of Gini impurity when a variable is chosen to split the node. Thus, a higher score suggests that the feature is more important.

Based on the results of our random forest model, the History of Monthly Payment Records for September 2005 (PAY_0), Amount of Bill Statement for September 2005 (BILL_AMT1) and AGE have the greatest variable importance, and are therefore more useful in predicting if customers are likely to default on their payments.



```
> importance(model.RF)
                              0         1 MeanDecreaseAccuracy MeanDecreaseGini
LIMIT_BAL             7.3191144 228.46510            224.82176       394.413901
SEX_male              2.9360044  79.88477             79.07847        60.247783
EDUCATION_gradSch    -7.0142758  60.64766             60.02721        43.157190
EDUCATION_university -1.4095285  74.15915             72.94690        43.253008
EDUCATION_highSch     6.0526873  59.64624             58.22171        40.440053
MARRIAGE_married     15.8577469  50.24364             50.01186        35.306811
MARRIAGE_single      14.0011117  40.88448             39.69955        30.496597
AGE                  22.0160806 270.30057            257.86916       439.418288
PAY_0_unknown0        1.0615341  15.29209             15.89406        41.900598
PAY_0_unknown2       13.4577439  23.48851             24.85921         8.672834
PAY_0_payDuly       -11.4162061  29.84370             30.38039        15.061424
PAY_0_value         102.9858934 187.36316            197.40604       867.163237
PAY_2_unknown0        8.8765199  13.85535             15.39739         8.721233
PAY_2_unknown2        0.8867156  15.37762             17.32220         6.567303
PAY_2_payDuly        -0.1099632  25.82323             27.06660        14.461226
PAY_2_value          25.5660772  41.06801             45.26746       212.235138
PAY_3_unknown0        9.1174242  17.33850             18.63557         8.759983
PAY_3_unknown2       -1.5052854  23.14779             24.73233         7.362954
PAY_3_payDuly        -8.4710347  40.24615             40.56795        21.762342
PAY_3_value          44.8521084  45.61267             53.49168        87.176988
PAY_4_unknown0       11.3957235  27.11785             30.14862         9.353307
PAY_4_unknown2        7.5022810  21.45000             25.17531         7.876318
PAY_4_payDuly        -3.0731846  31.79612             31.85575        16.319276
PAY_4_value          36.7552719  51.78962             52.88912        45.898925
PAY_5_unknown0        4.6684306  32.09275             34.76779        11.335084
PAY_5_unknown2        3.8400191  21.09310             24.42528         7.215134
PAY_5_payDuly       -13.3666238  29.16356             29.04671        10.147910
PAY_5_value          29.1038844  42.55169             46.73122        40.710994
PAY_6_unknown0        5.0544984  32.67082             34.26577        14.350067
PAY_6_unknown2        3.6168077  28.34782             31.39245         8.588004
PAY_6_payDuly        -8.3963547  37.90447             38.26914        13.870732
PAY_6_value          20.9977506  43.88961             47.66982        34.497935
BILL_AMT1            -5.6336416 202.63283            214.14035       412.976723
BILL_AMT2            22.3720359 129.07271            142.46543       281.014962
BILL_AMT3            28.8688406 132.41550            149.97682       278.802429
BILL_AMT4            34.3167318 129.22635            151.42306       269.071139
BILL_AMT5            23.8163144 131.59364            147.39394       254.086395
BILL_AMT6            -0.9103603 160.82593            174.45090       287.726346
PAY_AMT1             28.1802763 147.70041            141.17723       326.480024
PAY_AMT2             25.0329654 144.12351            140.10959       379.153012
PAY_AMT3             29.4938432 137.66709            141.53260       329.666926
PAY_AMT4             20.0676787 127.84133            137.20926       275.930215
PAY_AMT5             30.3343316 137.58752            136.57948       273.460528
PAY_AMT6             11.4558118 189.77970            189.73741       298.893839
```

4.2.4. Naive Bayes

Overview

The Naive Bayes Classification model is a probabilistic one that uses Bayes Theorem to estimate conditional probabilities of the target variable given the values of the predictor variables. It begins with the raw probabilities of each class, then updates them with respect to the predictor variables, assuming independence between variables. This has several implications for our dataset.

Strengths and Limitations

A naive bayes model works well with relatively fewer data points compared to more complex models. This is especially suited to our train set which has a small number of unique data points in the minority class. However, the out of the box (OOTB) model offered by the R package “naivebayes” operates on several assumptions that might be violated by our dataset.

Firstly, any naive bayes model assumes conditional independence between variables but most features in our dataset are all likely to be dependent on each other. For example, payment amount in month t is likely to be dependent on the bill amount in month t-1 and bill amount in month t is likely to be dependent on the bill amount in month t-1.

Additionally, conditional probabilities inferred by the model are computed on the assumption that variables follow a gaussian distribution. However, our exploratory data analysis shows that most of our variables are distributed with a significant skew.

Model Fitting

The R package “naivebayes” was used to create a naive bayes model. Several possible variations in model hyperparameters were identified to improve performance or account for the limitations of the OOTB model.

1. Using the balanced train sets to fit our model results in the model inferring a different set of prior probabilities than the original train set. It might be helpful to pass the model a set of prior probabilities computed before the balancing.

2. The “naivebayes” package allows a laplace smoothing value to be passed for Laplace smoothing (additive smoothing). We will explore a range of values along a log scale to find a suitable value.

3. As highlighted in the limitations, the OOTB model uses a gaussian distribution to model numerical variables when our variables do not follow a gaussian distribution. To address this, the “naivebayes” package offers a “usekernel” flag that allow the model to estimate variable distributions with non-parametric methods.

4. Similar to the “usekernel” hyperparameter, “usepoisson” is a flag that allows the model to estimate the distribution of integer variables with a poisson distribution.

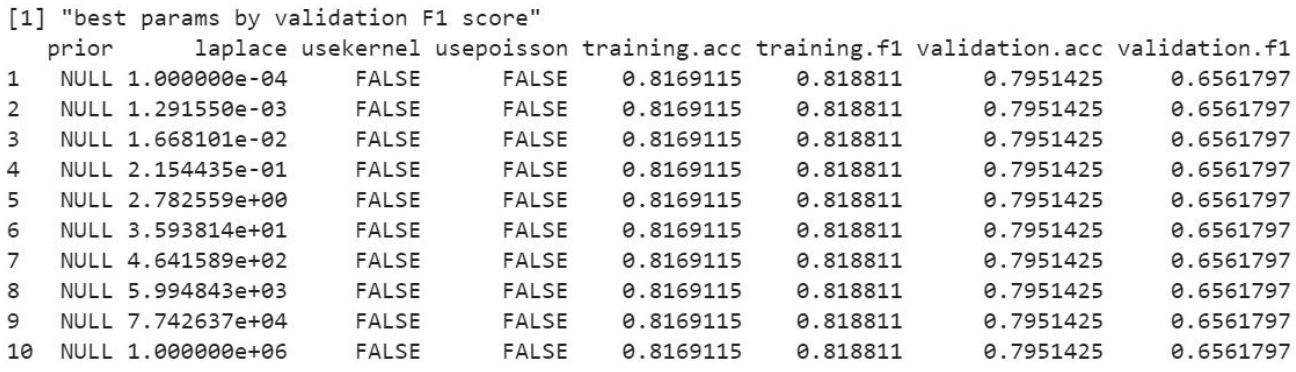
Each of these options were explored in isolation, after which a grid search was performed to find the best combinations of hyperparameters. Models were trained on a subset of the features selected by the t-test p-values in our exploratory data analysis. Some of the more notable results are discussed below and implementation details can be found in the accompanying code.

Results and Discussion

Through 10-fold cross validation, the mean accuracy and F1 score was obtained for each model. The mean values for each metric were used as benchmarks and are shown in the table below. When comparing training and validation metrics, we can see that the difference in mean F1 scores is much larger than the difference in mean accuracies. This is indicative of the general tendency for models to overfit to the majority class.

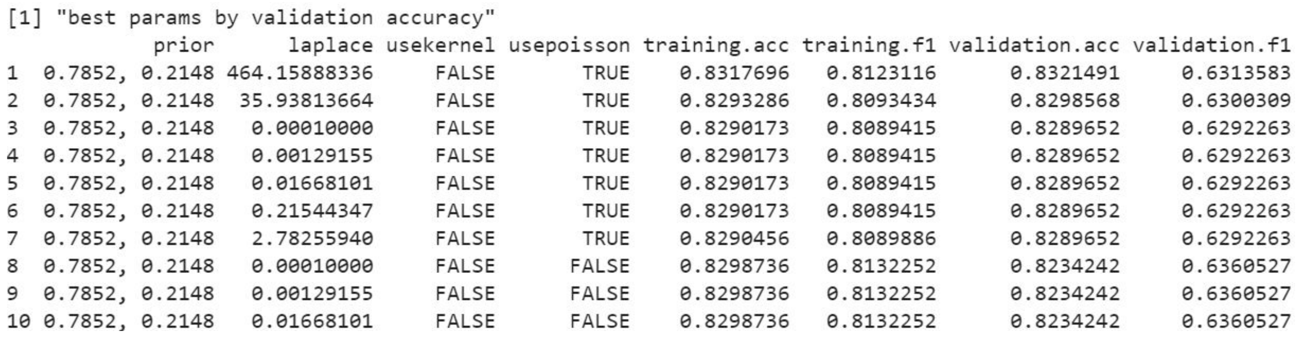
	Accuracy	F1 score
Training set	0.792	0.728
Validation set	0.724	0.350

Surprisingly, the best models by F1 score were all very close to the OOTB model. None of the top 10 models were supplied with pre-balancing class probabilities or used a kernel density estimate or poisson distribution to model variable distribution. Additionally, laplace smoothing does not seem to influence the F1 score for these models.



```
[1] "best params by validation F1 score"
   prior      laplace usekernel usepoisson training.acc training.f1 validation.acc validation.f1
1   NULL 1.000000e-04     FALSE      FALSE    0.8169115    0.818811      0.7951425     0.6561797
2   NULL 1.291550e-03     FALSE      FALSE    0.8169115    0.818811      0.7951425     0.6561797
3   NULL 1.668101e-02     FALSE      FALSE    0.8169115    0.818811      0.7951425     0.6561797
4   NULL 2.154435e-01     FALSE      FALSE    0.8169115    0.818811      0.7951425     0.6561797
5   NULL 2.782559e+00     FALSE      FALSE    0.8169115    0.818811      0.7951425     0.6561797
6   NULL 3.593814e+01     FALSE      FALSE    0.8169115    0.818811      0.7951425     0.6561797
7   NULL 4.641589e+02     FALSE      FALSE    0.8169115    0.818811      0.7951425     0.6561797
8   NULL 5.994843e+03     FALSE      FALSE    0.8169115    0.818811      0.7951425     0.6561797
9   NULL 7.742637e+04     FALSE      FALSE    0.8169115    0.818811      0.7951425     0.6561797
10  NULL 1.000000e+06     FALSE      FALSE    0.8169115    0.818811      0.7951425     0.6561797
```

In contrast, the best models by validation accuracy were all supplied with the original pre-balancing class probabilities. Additionally, a large number of them used a poisson distribution to model variable distribution for integers. Lastly, it is interesting to note that the top 2 models use a significantly larger amount of laplace smoothing than the other models.



```
[1] "best params by validation accuracy"
            prior      laplace usekernel usepoisson training.acc training.f1 validation.acc validation.f1
1  0.7852, 0.2148 464.15888336     FALSE       TRUE    0.8317696   0.8123116      0.8321491     0.6313583
2  0.7852, 0.2148  35.93813664     FALSE       TRUE    0.8293286   0.8093434      0.8298568     0.6300309
3  0.7852, 0.2148   0.00010000     FALSE       TRUE    0.8290173   0.8089415      0.8289652     0.6292263
4  0.7852, 0.2148   0.00129155     FALSE       TRUE    0.8290173   0.8089415      0.8289652     0.6292263
5  0.7852, 0.2148   0.01668101     FALSE       TRUE    0.8290173   0.8089415      0.8289652     0.6292263
6  0.7852, 0.2148   0.21544347     FALSE       TRUE    0.8290173   0.8089415      0.8289652     0.6292263
7  0.7852, 0.2148   2.78255940     FALSE       TRUE    0.8290456   0.8089886      0.8289652     0.6292263
8  0.7852, 0.2148   0.00010000     FALSE      FALSE    0.8298736   0.8132252      0.8234242     0.6360527
9  0.7852, 0.2148   0.00129155     FALSE      FALSE    0.8298736   0.8132252      0.8234242     0.6360527
10 0.7852, 0.2148   0.01668101     FALSE      FALSE    0.8298736   0.8132252      0.8234242     0.6360527
```

These disagreements in model parameters seem to indicate that the naive bayes model is prone to overfitting on our particular dataset when hyperparameters are tuned for accuracy. This is likely due to the difference in the number of unique entries between the majority and minority class. As such, F1 scores were used to rank models and the OOTB model was selected as the best model.

4.2.5. Support Vector Machine (SVM)

Overview

SVM is a technique that finds two hyperplanes to separate two groups of data, and the margin between them is maximised. In the case of linearly inseparable data, we may map it to higher dimensions by applying non-linear kernel functions. What is important, then, is to tune the parameters of the model, such as the regularisation parameter C and the kernel hyperparameters, in order to avoid over-fitting.

Strengths and Limitations

This technique works best when our two classes are distinctly separable. However, we may expect that the factors affecting whether customers default may still result in some overlapping classes. Furthermore, a SVM model is a black box, which makes it difficult to understand which factors are more important in making the predictions.

The SVM model is also non-probabilistic, as the data points are classified by placing them only either above or below the hyperplanes, hence we cannot obtain a quantifiable likelihood of an outcome. Keeping in mind these limitations, we trained a model on the balanced data. We then conducted 10-fold cross validated grid search to determine an optimal set of hyperparameters.

Model Fitting

1. All SVM functions were taken from the "e1071" package.

2. We tested linear, polynomial, radial basis and sigmoid kernels. *u* and *v* are vectors in the input space.

   linear: $u'v$

   polynomial: $(\gamma u'v + coef0)^{degree}$

   radial basis: $exp(-\gamma|u-v|^2)$

   sigmoid: $tanh(\gamma u'v + coef0)$

   Due to time constraints, the coef0 for polynomial and sigmoid kernels was left as the package's default value of 0. The degree for polynomial was left as the default of 3 as well.

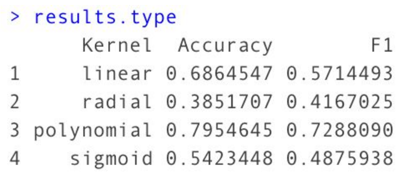
3. For all models with non-linear kernels, we varied gamma γ. Gamma defines how far the influence of a single training example reaches. It is the inverse of the radius of influence of the examples selected by the model as support vectors. A small gamma means that the decision boundary will be dependent on examples that are further away, while a high gamma means that examples that examples nearer to the decision boundary will carry more weight.

4. For all models, we also varied the cost across 2^{-2}, 2^{-1}, 1, 2^1 and 2^2. The cost parameter refers to the cost of violating constraints, i.e., within the margin made by the two hyperplanes in SVM. A smaller C value would penalise samples falling within the margin less and allows room for more misclassification than if a higher C value were used.

To limit the number of parameters searched, we focused on the gamma and cost options as they control the kernel behaviour and regulation respectively. Cost and gamma were explored in isolation, after which a grid search was performed to find the best combination of hyperparameters in order to prevent over or under fitting on the training set. The results of this process are discussed below and details can be found in the accompanying code.

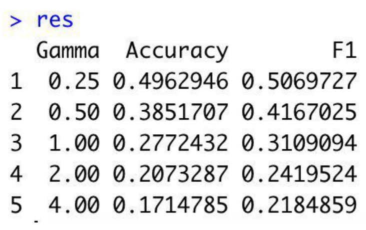
Results and Discussion

Exploration of Kernel Type



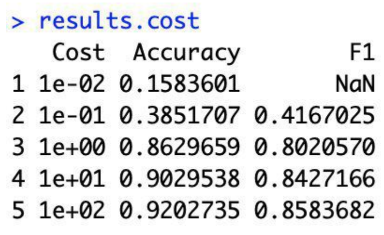
```
> results.type
     Kernel  Accuracy        F1
1    linear 0.6864547 0.5714493
2    radial 0.3851707 0.4167025
3 polynomial 0.7954645 0.7288090
4   sigmoid 0.5423448 0.4875938
```

Exploration of Gamma



```
> res
  Gamma  Accuracy        F1
1  0.25 0.4962946 0.5069727
2  0.50 0.3851707 0.4167025
3  1.00 0.2772432 0.3109094
4  2.00 0.2073287 0.2419524
5  4.00 0.1714785 0.2184859
```

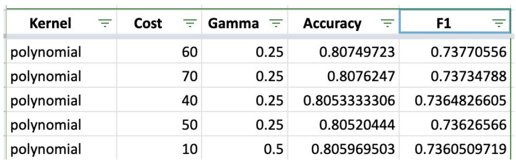
Exploration of Cost



```
> results.cost
   Cost  Accuracy        F1
1 1e-02 0.1583601       NaN
2 1e-01 0.3851707 0.4167025
3 1e+00 0.8629659 0.8020570
4 1e+01 0.9029538 0.8427166
5 1e+02 0.9202735 0.8583682
```

Based on our preliminary exploration, we found that the best Kernel type was "polynomial", the best range of gamma was 2^{-2} to 2^{-1} and the best range of cost was from 10 to 100. We performed a grid search using this narrowed range of hyperparameters to get the best combination possible.

Grid Search Results (Top 5)



Kernel	Cost	Gamma	Accuracy	F1
polynomial	60	0.25	0.80749723	0.73770556
polynomial	70	0.25	0.8076247	0.73734788
polynomial	40	0.25	0.8053333306	0.7364826605
polynomial	50	0.25	0.80520444	0.73626566
polynomial	10	0.5	0.805969503	0.7360509719

Based on the F1 scores, the optimal type, cost and gamma is polynomial, 60 and 0.25 respectively, i.e. the model has the kernel: $(0.25u'v)^3$.

We then trained an SVM model with these parameters on the full train dataset. This model served as our final model for evaluation.

4.2.6. Neural Network

Overview

A neural network uses a series of mathematical equations to model the relationship between the input and output variables, mimicking the way the human brain operates. A neural network typically comprises of an input layer, one or more hidden layers and an output layer, each consisting of several neurons. We employed the backpropagation method to train the neural network.

Strengths and Limitations

The neural network model was considered for its suitability to problems where the train data is large and noisy. The model is thus suited to our oversampled train dataset that has 15704 samples and contain errors highlighted in Section 2.

However, the neural network classifier produced is incomprehensible; Unlike other machine learning models, the complexity of the neural network creates a "black box", such that we are unable to discern the exact attributes and weightages contributing to the classification. The neural network model may not be suitable if the bank wishes to understand how the model distinguishes credible and non-credible customers.

Due to the complexity of the neural network model, the model also runs the risk of being overfitted to the training data, such that the model cannot be generalized and used to make accurate predictions for new data.

Model Fitting

The R package "nnet" was used to create the neural network model. The "nnet" package was chosen as the package performs early stopping, i.e., stops training when the validation error stops reducing, to prevent overfitting. Several variations in the training pipeline and model hyperparameters were identified to improve the performance of the neural network model.

1. The "nnet" package allows us to specify the number of hidden layers in our model. We will explore a range of values to find a suitable size for the model. As the nnet model has a limit of 1000 weights, number of layers up till 21 were tested to get as close to the 1000 weight limit as possible.

2. The "nnet" package also allows us to specify the weight decay of our model. We explored the range of decays from 0 to 1 at intervals of 0.2 to detect the best range of weight decays.

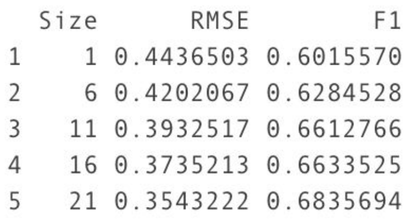
3. As the model is required to produce a binary response, we set entropy = TRUE.

4. The maximum number of iterations was set to 1000. This was an arbitrary number to set to limit the time required for training and testing different models.

The size and decay options were explored in isolation, after which a grid search was performed to find the best combination of hyperparameters. The results of this process are discussed below and details can be found in the accompanying code.

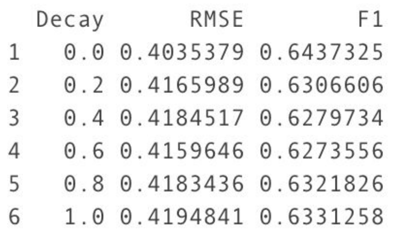
Results and Discussion

Exploration of Number of Hidden Layers



```
  Size      RMSE        F1
1    1 0.4436503 0.6015570
2    6 0.4202067 0.6284528
3   11 0.3932517 0.6612766
4   16 0.3735213 0.6633525
5   21 0.3543222 0.6835694
```

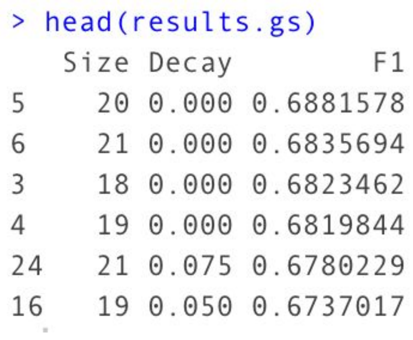
Exploration of Weight Decay



```
  Decay      RMSE        F1
1   0.0 0.4035379 0.6437325
2   0.2 0.4165989 0.6306606
3   0.4 0.4184517 0.6279734
4   0.6 0.4159646 0.6273556
5   0.8 0.4183436 0.6321826
6   1.0 0.4194841 0.6331258
```

Based on the two tables and the F1 scores above, the best size range is within [16 21], and the best decay is within the range [0.0, 0.2]. We then performed a grid search within this size and decay range.

Grid Search Results



```
> head(results.gs)
   Size Decay        F1
5    20 0.000 0.6881578
6    21 0.000 0.6835694
3    18 0.000 0.6823462
4    19 0.000 0.6819844
24   21 0.075 0.6780229
16   19 0.050 0.6737017
```

Based on the F1 scores, the optimal size and decay is 20 and 0.0 respectively.

We then trained a neural network model with these parameters on the full train dataset. This model served as our final model for evaluation.

5. Model Evaluation

5.1. Explanation of Metrics

Accuracy
Accuracy refers to the percentage of the observations that have been correctly classified (i.e., the predicted class and the actual class are the same). Accuracy is calculated using $\frac{TP+TN}{TP+FP+TN+FN}$, and acts as a quick evaluation of the various models.

F1 Statistic (As Mentioned Above)

Area Under Receiving Operator Characteristics (AUROC)
The ROC is a probability curve of True Positive Rate (TPR) against False Positive Rate (FPR) at different thresholds. AUROC measures the performance of our classification models in distinguishing between default and non-default customers. The closer the AUC is to 1, the better the model is in predicting whether a customer is credible.

Lift
Lift indicates how much higher than the actual percentage of positive instances in a decile dec is than the rate expected. For a well performing model, the lift curve should start well above 1.0 and cross 1.0 at one of the lower deciles. The values below reflect the total area under the lift chart.

Gain and Cumulative Gain
Gain is a measure how many of the positive instances in the overall test set are found in a particular decile. A chart of cumulative gain against percentiles can be plotted and the area under the cumulative gain chart calculated. The greater the area under the cumulative gain chart, the more positive test instances are found at lower deciles and thus the better the performance of the model.

5.2. Summary

The following table is a summary of the results of our model performance based on various metrics on the test dataset.

Model	Accuracy	F1 statistic	AUROC	Lift	Gains
Logistic Regression	0.758	**0.527**	**0.765**	0.703	**0.833**
SVM	0.666	0.386	0.544	**0.903**	0.582
Decision Tree	0.700	0.442	0.682	0.853	0.641
Random Forest	**0.780**	0.505	0.760	0.838	0.700
Naive Bayes	0.681	0.480	0.737	0.820	0.683
Neural Network	0.720	0.469	0.713	0.848	0.664

5.3. Discussion of Results

5.3.1. Logistic Regression

Our results show that logistic regression performs well across both quantitative and qualitative standards. As previously mentioned, logistic regression qualitatively performs better than other models due to the simplicity of implementation and the interpretability of its model parameters. The quantitative performance is discussed more below.

From the summary table, logistic regression returned the best F1 score, AUROC and gain. This was a surprisingly performant result for a model as simple as logistic regression. High AUROC and F1 indicates less bias in classifying into either classes. However, we believe that this can be explained in part by the model's comparative robustness to unbalanced training data.

As the train-test split was done randomly, we can expect the class distributions to be similar between the two datasets. Since a logistic regression model preserves the marginal probabilities of the training data, we expect that fitting the model on an unbalanced train set should improve its overall performance on a test set with a similar class distribution. In contrast, our other models require a balanced train set due to their tendency to overfit the majority class. This skews their predictions away from the original class distribution and should result in comparatively lower performance across the metrics.

5.3.2. Discussion of Sub-Optimal Models

The naive bayes model underperformed considering the relatively high training and validation accuracies and F1 scores. Interestingly, an alternate naive bayes model tuned on validation accuracy performs much better than the one tuned on validation F1 score. This seems to suggest that the choice of metric strongly affects model performance on the test set. We suspect that the disparity between results might have arisen from the small train dataset, which did not favor the use of model features that were more likely to suit our test data (e.g. Non-parametric estimates of variable distributions, laplace smoothing, etc).

Another model that performed surprisingly badly was the neural network model, especially considering the neural network's suitability for noisy datasets similar to the dataset provided. A possible explanation for its poor performance is the data-hungry nature of the neural network algorithm. While the oversampled train dataset contained 15704 samples, the number of samples may be insufficient given the noise in the data.

The SVM model also underperformed. Based on our discussion in Section 2, the data provided was relatively noisy and substantial amount of cleaning had to be done to prepare the data. Given that the SVM algorithm works by finding the hyperplane that best separates the data points of both classes (Default and No Default), having noise in the data will make data points of both classes overlap, significantly reducing the performance of SVM. Additionally, the SVM's linear decision boundary is likely to limit its performance. As our data is noisy, it is likely that it should remain linearly inseparable even after being mapped to a higher dimensional kernel space.

The decision tree and random forest models underperformed comparatively to the logistic regression model. The decision tree models categorize data at each decision node into categories. This makes the model less adept at handling continuous data as the decision tree loses information on continuous variables. As our dataset contains a number of continuous variables, the decision tree and random forest models may not be best suited to the data we are dealing with in the project.

6. Possible Improvements

6.1. Improving the Train-Test Split

For this project, we split the full dataset based on a train-test ratio of 33-67. It might be better to have more data for the training dataset, for instance, by using a 70-30 split instead. Having a greater percentage of training data will also increase the likelihood of the training data being representative of the population. Moreover, some algorithms such as the neural network algorithm are data-hungry, and perform significantly better when more data is available.

6.2. Consider Other Ensemble Models

The relatively good performance of the random forest model, as compared to the decision tree model, might be indicative of the predictive power of ensemble models. Ensemble models aggregate the predictions of several models (base models) to give the final prediction, thus reducing the variance of predictions from just one model. In addition, different ensemble models are able to modify the training process for base models through bagging, bootstrapping or stacking. This allows the ensemble model to be more robust and accurate than the base models individually. It might therefore be useful to create an ensemble model as opposed to relying on only the logistic regression model selected from our results.

6.3. Introduce Early Stopping During Model Fitting

Based on the discrepancies in F1 scores for our model on the train and test datasets, it is likely that our models are overfitted to the train dataset. Thus, a possible improvement would be to introduce early stopping during the model fitting process. Instead of training for a fixed number of iterations, we could stop the model training process as soon as the validation loss rises to prevent overfitting of the model to our data.

6.4. Gather more information on the Dataset

Data provided by the Taiwanese bank was relatively noisy. As a result we were forced to handle unknown data by either dropping entries or grouping unknown values into a separate category. While our choice to apply the latter results in smaller information loss than dropping entries, we nonetheless lose valuable information during the training phase.

It may thus be good to clarify certain data points with the bank to feed cleaner data into our models. For instance, the 0, 5 and 6 categories in EDUCATION (X3) and the 0 category in MARRIAGE (X4) could be labelled properly. Additionally, variables such as LIMIT BALANCE (X1) that currently includes both individual consumer credit and his/her family (supplementary) credit could be split into two attributes - one attribute for individual credit and another attribute for supplementary credit.