**Executive Summary**

In the following report, we conduct an analysis on the proposed implementation of machine learning to predict loan defaulters, reducing our losses. We first begin with an overview of the problem at hand and variables. We then explore the dataset and note relationships between variables. Furthermore, due to the unbalanced nature of the dataset, we conduct SMOTE and undersample the majority while oversampling the majority.

We then investigate the robustness of different machine learning models on the dataset, making use of Accuracy, F1-Score and false negative rate to determine the most applicable model. Throughout, we discuss the extent of usefulness of each model. We end the report with an evaluation of our work, noting further areas of exploration and potential for improvements.

**1. Brief introduction of data set and data modelling problem**

The variables Sex, Education, and Marital status defined as categorical variables while the other variables are continuous variables. It also describes historical payment data which is only available for existing customers. The total proportion of defaults in the data is 22.12% which is 6,636 out of the total data set comprising of 30,000 samples.

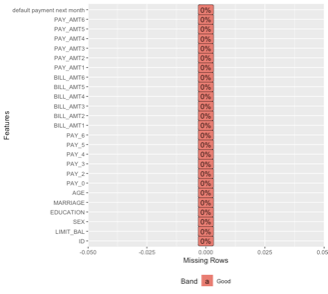
**1.1 Description of the variables**

The data consists of 30,000 rows and 25 columns of variables. Each sample corresponds to a single customer. The columns consist of the following variables:

|  |  |
| --- | --- |
| **ID:** | Individual ID |
| **Balance Limit**: | Amount of the given credit (NT dollar) - It includes both the individual consumer credit and his/her family (supplementary) credit |
| **Sex**: | Gender: 1 = male; 2 = female |
| **Education:** | Education: 1 = graduate school; 2 = university; 3 = high school;  0, 4, 5, 6 = others |
| **Marriage:** | Marital status: 1 = married; 2 = single; 3 = divorce; 0 = others |
| **Age:** | Age (year). |
| **Pay (X6 - X11)**: | History of past payment. We tracked the past monthly payment records (from April to September, 2005) as follows:   * X6 = the repayment status in September, 2005; * X7 = the repayment status in August, 2005; . . .; * X11 = the repayment status in April, 2005. * The measurement scale for the repayment status is:   o -2: No consumption & balance fully paid  o -1: Paid in full, but balance may still be positive due to recent transactions for which payment has not yet come due. 0: Paid the minimum due amt, but not the full balance (revolved the remaining balance)  o 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 Amount (X12-X17)**: | Amount of bill statement (NT dollar): X12 = amount of bill statement in September, 2005; X13 = amount of bill statement in August, 2005; . . .; X17 = amount of bill statement in April, 2005. |
| **Pay Amount (X18-X23)**: **Pay Amount (X18-X23)**: | Amount of previous payment (NT dollar).X18 = amount paid in September, 2005; X19 = amount paid in August, 2005; . . .;X23 = amount paid in April, 2005.  Amount of previous payment (NT dollar): X18 = amount paid in September, 2005; X19 = amount paid in August, 2005; . . .;X23 = amount paid in April, 2005. |
| **Default:** | The client's behavior in next month: Y=0 then not default, Y=1 then default |

**1.2 Cleanliness of the dataset**

Since the data is not collected by us, we check for missing values in the dataset. It does not contain any NA values as seen below:



**1.3 Defining the problem**

Based on the premise of the data, the problem statement is as follows:

|  |  |  |
| --- | --- | --- |
|  | How can we make use of the dataset to better predict potential customers that will default, reducing the number of customers we might lose? |  |

Hence, we will need to leverage on the data set and develop a robust model to predict customers that might default. This model will then aid the bank in predicting potential defaulters in future loans and not loan to them, reducing our losses from loan defaults.

**2. Exploratory data analysis:**

**2.1.1 Demographic Variables**

|  |  |  |
| --- | --- | --- |
| **Statistics** | **Age** | **Balance Limit** |
| Min | 21 | 10000 |
| Max | 79 | 1000000 |
| Mean | 35.49 | 167484 |
| Median | 34 | 140000 |
| Std | 9.21 | 129747.7 |
| 25% | 28 | 50000 |
| 75% | 41 | 240000 |

Table 1

The median of 34 years, with a longer tail towards the right side, up to a maximum value of 79 years and a minimum value of 21. The median and mean are quite close to each other, showing that the average age cuts the distribution in half, as seen in Table 1. The mode of age is at 29 years, which can be seen as the highest peak in Figure 2. On the basis of visual analysis, it can be seen that older customers have a higher chance of default. The distribution of balance limit has a large tail towards the higher values. The maximum value of balance limit in the data is $1,000,000 but 75% of the values are less than $240,000. This effect can also be seen in Figure 6, which indicates it to be even more pronounced in the subset of customers that default. This points to customers with low balance limit to possibly having a higher chance of default.

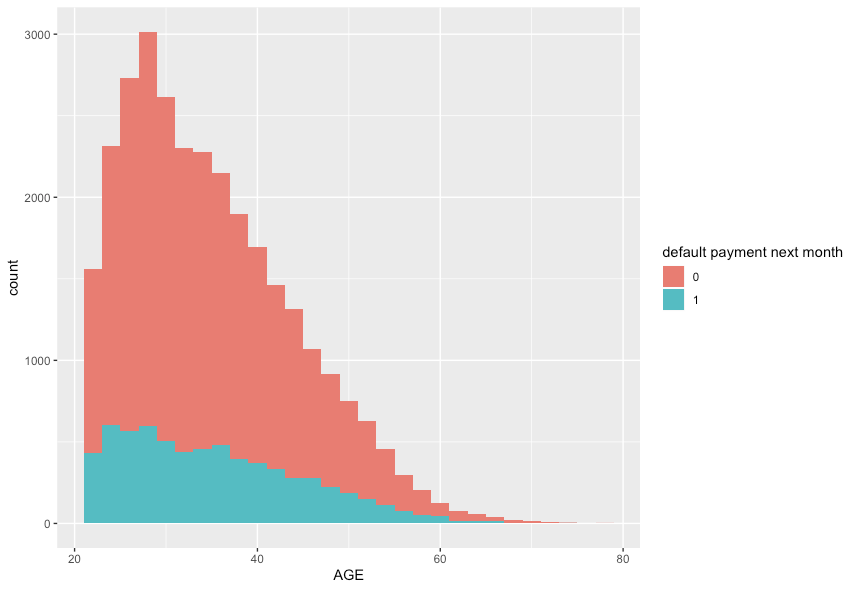
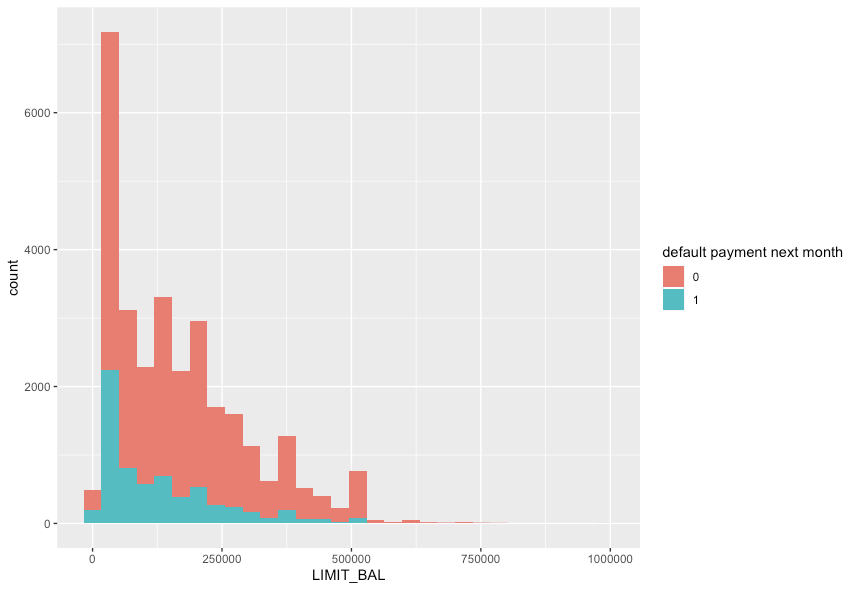


Figure 1 Figure 2

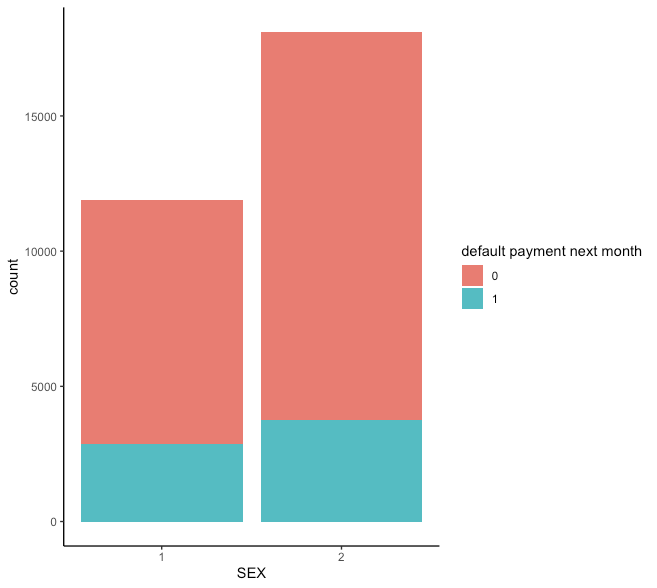
**Categorical Variables**

Figure 3

The distribution of customers into male and female is visualized in Figure 3. Approximately 60.4% of the customers in the dataset are female, and the remaining 39.6% male. The proportion of defaults among men is 24.2% and 20.8% among women.

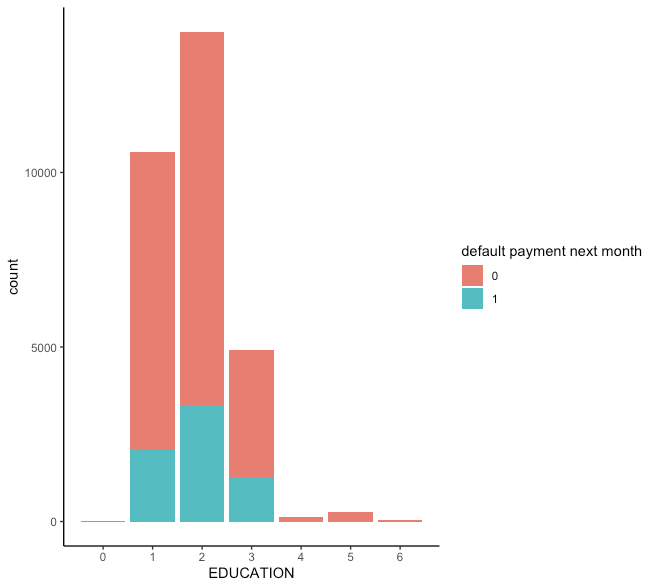


Figure 4

Distribution of customers’ education level is shown in Figure 4. The proportions of customers in each “Graduate School”, “University”, “High School” are: 19.2%, 46.8%, and 16.4%. The last category “Other” amounts to only 1.6% of the customers in the dataset. We can see that University is the most common level of education in the dataset, followed by Graduate school and High school. There are small differences in proportions of defaults in the categories, with

“Graduate School” at a default rate of 23.4%, “University” at 23.7%, and “High School” at 25.2%.

The category “Other” only has about 7% defaults, although containing such a small number of customers, it could be due to chance.

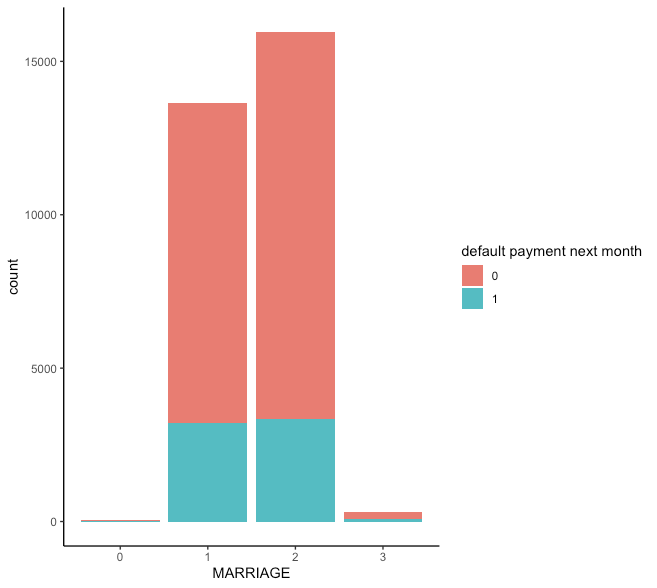


Figure 5

Distribution of Marital Status is shown in Figure 5. Marital status is mostly divided into categories “Married”, “Single”, “Divorced” and “Other” with respective proportions of 45.5%, 53.2%, 0.01% and the group “Other” containing only 0.0018% of the customers. here are small differences in proportions of defaults in the categories, with “Single” at a default rate of 20.9%, “Married” at 23.5%, “Divorced” at 26.0% and “Others” at 0.09%.

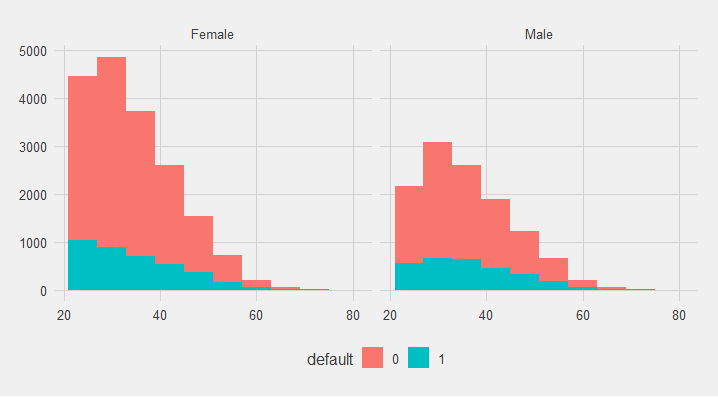
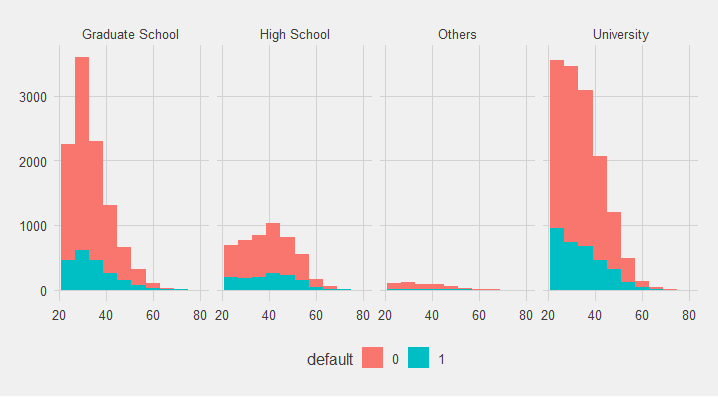
**2.1.2 Distributions of age**

Age distributions of different customer groups are visually examined using histograms of both default and non-default customers in Figure 6.

Figure 6 shows the different levels of education and distribution of customer’s age among them. Graduate School education seems to show a higher peak around 30 years of age in the non-defaulting group than in the defaulting group. This indicates that under 30 year old customers with a graduate school education are have a lower likelihood to default as compared to older customers with similar education.

Another difference in default can be found in Figure 7, where the distribution of age among defaulting men is flatter in their non-defaulting counterpart. The decline in number of customers start from 30 years among the non-defaulting group, while the proportion of of default and non-defaulters remain relatively constant after 30 years. For women, a similar effect is visible.

Some differences can be found but no clear guidelines can be made with this level of analysis. A more in-depth approach is required to meaningfully differentiate between low or high risk customers. This analysis of distribution is however useful to understand some of the characteristics of the data and to later describe the result of customer segmentation.

Figure 6. Education and Age Figure 7: Sex and Age

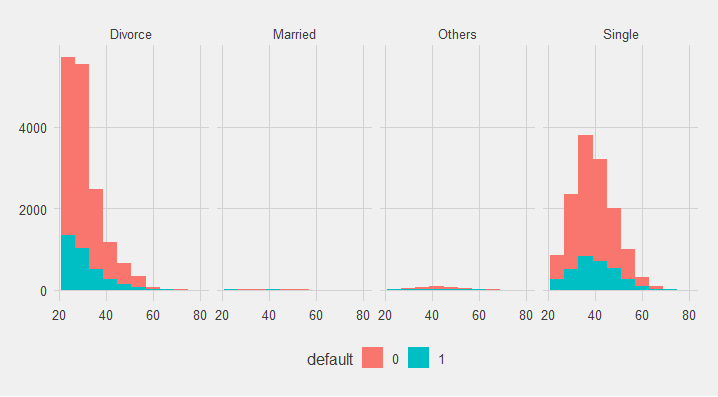


Figure 8: Marital Status and Age

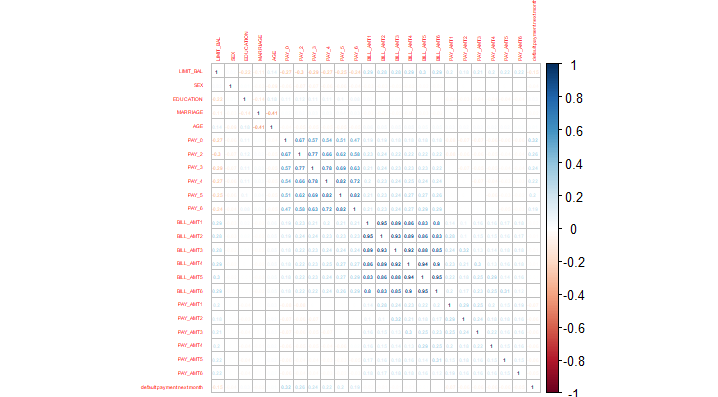
**2.1.3 Correlation Analysis**

A correlation matrix of all variables in the dataset is shown in Figure 9. The only variable with a notable correlation status with default is payment status 0.

The only variable with a notable correlation with default is the history of payment record in the month of september. The repayment status(September) has a pearson correlation coefficient of 0.32 with default. The other month’s correlations are slightly lower. This correlation is not surprising as a customer with at least a month or more delay is naturally more likely to be unable to pay for their bills in the upcoming months and thus defaulting.

The next highest correlation with default comes with limit balance at -0.15 which is slightly negative correlated. These measures indicates that customers with lower balance limits have more delays in payment and are more likely to default. Figure 1 also supports this argument since most defaulting customers have low limit balance.

Limit balance has a positive correlation coefficient of 0.14 with age and negative correlation coefficient of -0.22 with education. Age itself doesn’t seem to be correlated with default. However, Figure 2 seems to show that the proportion of defaults among age group grows with age. Negative correlation with education can be understood as customers with a higher level of education will generally have lower credit limits. Bill amounts doesn’t seem to be correlated with default. Payment amounts seem to be slightly negative correlated with default, but that is to be expected since customers with higher payments are less likely to default as they have the capacity to pay back.

Other demographic variables show little to no correlation with default.  
Figure 9

**3. Pre-Processing of Data**

**3.1.1.1 Feature Scaling of Data**

Since the dataset contains features that varies in magnitudes, units, and range. Normalisation via min-max scaling is performed to scale the features to a meaningful scale.

**3.1.1.2 Train Test Split**

Train test split is performed using the code provided. A 2:1 Test:Train data set is achieved.

**Data set balancing**

The classes of the dataset are not equally distributed, with 23364 being class 0 (Not default) and 6636 being class 1 (Default). This results in the Train data set having 7832 being class 0 (Not default) and 2168 being class 1 (Default). This would affect the accuracy calculated. Hence, to handle skewed classes, oversampling on the minority class 1 and undersampling on majority class 0 is done using SMOTE to balance the dataset.

**3.1.2 Feature Selection of Training Data**

Filtering VS Wrapping./Embedding

* Filter methods measure the relevance of features by their correlation with dependent variable while wrapper methods measure the usefulness of a subset of feature by actually training a model on it.
* Filter methods are much faster compared to wrapper methods as they do not involve training the models. On the other hand, wrapper methods are computationally very expensive as well.
* Filter methods use statistical methods for evaluation of a subset of features while wrapper methods use cross validation.
* Filter methods might fail to find the best subset of features in many occasions but wrapper methods can always provide the best subset of features.
* Using the subset of features from the wrapper methods make the model more prone to overfitting as compared to using subset of features from the filter methods.

Disadvantages of filtering :

1. No feedback
2. Criteria built in search with no reference to the learner
3. Ignores the learning problem
4. Looking at features in isolation

Advantages of Wrapping/Embedding :

1. Criteria built in the learner
2. Takes into account model bias and learning

**3.1.2.1 Embedded Method - Lasso Regression**

L1 (or LASSO) regression for generalized linear models can be understood as adding a penalty against complexity to reduce the degree of overfitting or variance of a model by adding more bias. Here, we add a penalty term directly to the cost function,

In L1 regularization, the penalty term is



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| L1 : λ Σki | wi | = λ | **w** | 1, |

where **w** is our *k-dimensional* feature vector. Through adding the L1 term, our objective function now becomes the minimization of the regularized cost, and since the penalty term grows with the value of the weight parameters (λ is just a free parameter to fine-tune the regularization strength), we can induce sparsity through this L1 vector norm, which can be considered as an intrinsic way of feature selection that is part of the model training step.

**3.1.2.2 Wrapper Method - Logistic Regression**

A wrapper method that was tried was to use logistic regression. The logistic regression model was run on the full set of training data. A p-value will be obtained for each variable in the model. Afterwards, variables that have a p-value of ≥ 0.05 are rejected. The new variables are then input into a new logistic regression model.

**3.1.2.3 Wrapper Method - Decision Tree**

One of the best ways for implementing feature selection with wrapper methods is to use Boruta package that finds the importance of a feature by creating shadow features.

It works in the following steps:

1. Firstly, it adds randomness to the given data set by creating shuffled copies of all features (which are called shadow features).
2. Then, it trains a random forest classifier on the extended data set and applies a feature importance measure (the default is Mean Decrease Accuracy) to evaluate the importance of each feature where higher means more important.
3. At every iteration, it checks whether a real feature has a higher importance than the best of its shadow features (i.e. whether the feature has a higher Z-score than the maximum Z-score of its shadow features) and constantly removes features which are deemed highly unimportant.
4. Finally, the algorithm stops either when all features get confirmed or rejected or it reaches a specified limit of random forest runs.

**3.1.2.4 Full Train Data**

We also ran our models with the full train dataset as a control to see if feature selection really plays an impact in affecting our evaluation metrics.

**4.1.1 Model Selection + Evaluation**

We will be covering 6 supervised learning models.

1. Naive Bayes
2. Logistic Regression
3. Support Vector Machines
4. Random Forest
5. Gradient boosting
6. Bagging (CART)

In each, we explain the strength and weaknesses as well as its applicability to our problem. We then provide the evaluation metrics and the best parameters/feature selection train set for the model.

We will mainly be using the following metrics to evaluate our predictions:

Accuracy, AUC, Recall and F1 Score.

Since our business objective is to reduce the number of defaults in customers, we optimise towards sensitivity(recall) because false positives(normal transactions that are flagged as likely to default) are more acceptable than false negatives(default transactions that are flagged as normal).

**4.2.1 SVM**

Industry usage : Sales forecasting

Strengths :

1. SVMs have regularization parameters that tolerates errors and avoid over-fitting
2. Kernel trick
3. Provides a good out-of-sample generalization

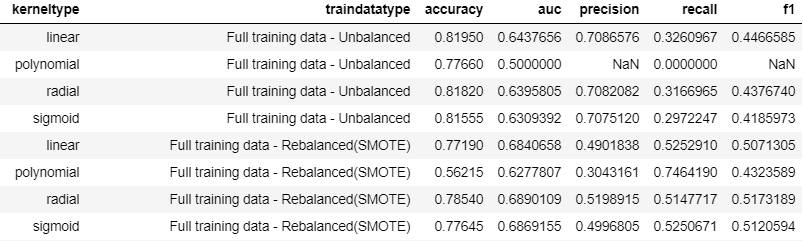
Weaknesses :

1. Bad interpretability
2. High computational cost
3. Users need to have domain knowledge to use kernel functions

Suitability for problem :

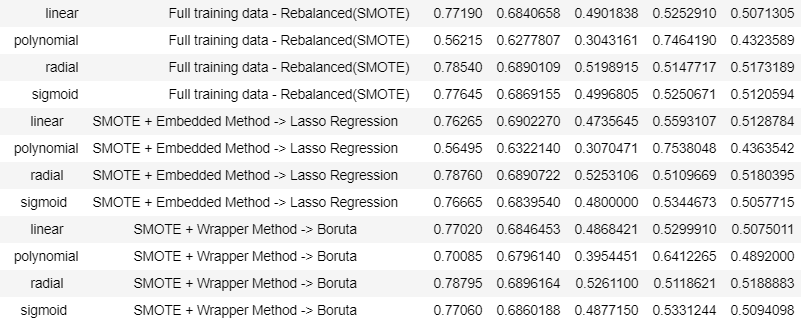
1. Reduces overfitting

We first compared between the original training set and a rebalanced training set using SMOTE to see the difference in result.

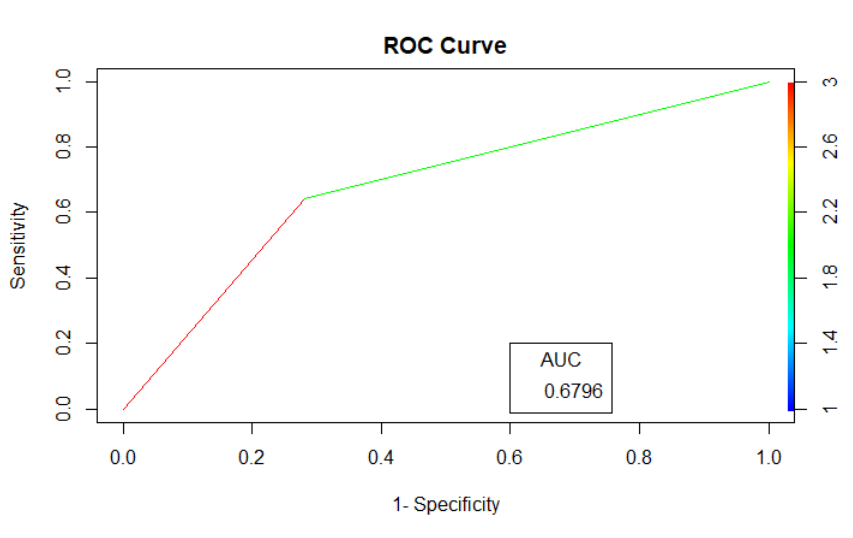
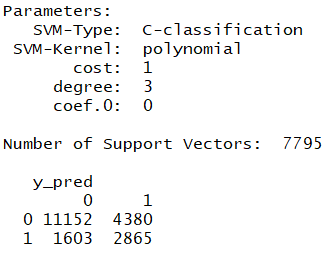


As we can see, a rebalanced dataset gives better results. We then proceed to conduct our feature selection for the various models.



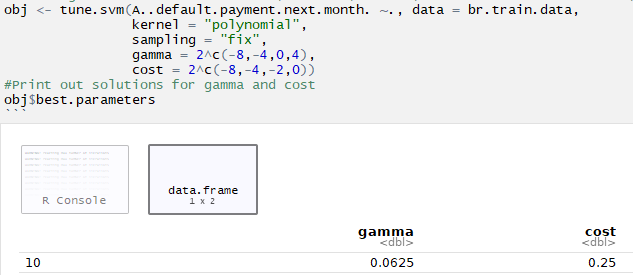


Based on all the outcomes, we decided that the best model to use is SVM-Polynomial Kernel with features that were selected from our random forest wrapper method.

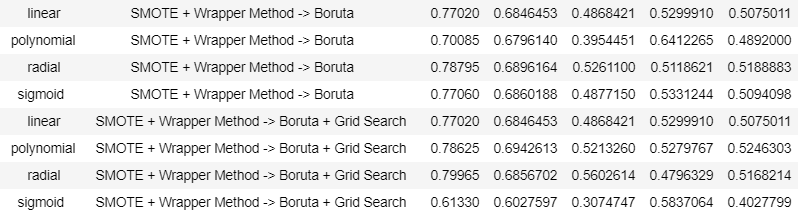


**Class 0(Non-default), Class 1(Default)**

While other models gives better recall, the tradeoff between recall and accuracy is too huge to justify the increase in recall. Thus, choosing the model with an overall high recall and accuracy score is the best option. This gives us a final accuracy of 70%, auc of 67.9%, recall of 64.1% and f1 score of 48.9%. We then conduct a simple grid search to see if other hyperparameters can result in better accuracy







After trying the new parameters obtained from grid search, it was discovered that the default settings for the polynomial model still performed better. Thus, we will stick to the default settings.

However, it is important to note that there are limitations for the polynomial kernel.

1. Tendency for polynomial kernels to [overfit](https://en.wikipedia.org/wiki/Overfitting).
2. May suffer from [numerical instability](https://en.wikipedia.org/wiki/Numerical_stability): when *x*T*y* + *c* < 1, *K*(*x*, *y*) = (*x*T*y* + *c*)*d* tends to zero with increasing *d*, whereas when *x*T*y* + *c* > 1, *K*(*x*, *y*) tends to infinity.

**4.2.2 Naive Bayes**

Industry Usage : Classification

Strengths:

1. Extremely fast for both training and prediction
2. Straightforward probabilistic prediction and easily interpretable
3. Very few tunable parameters

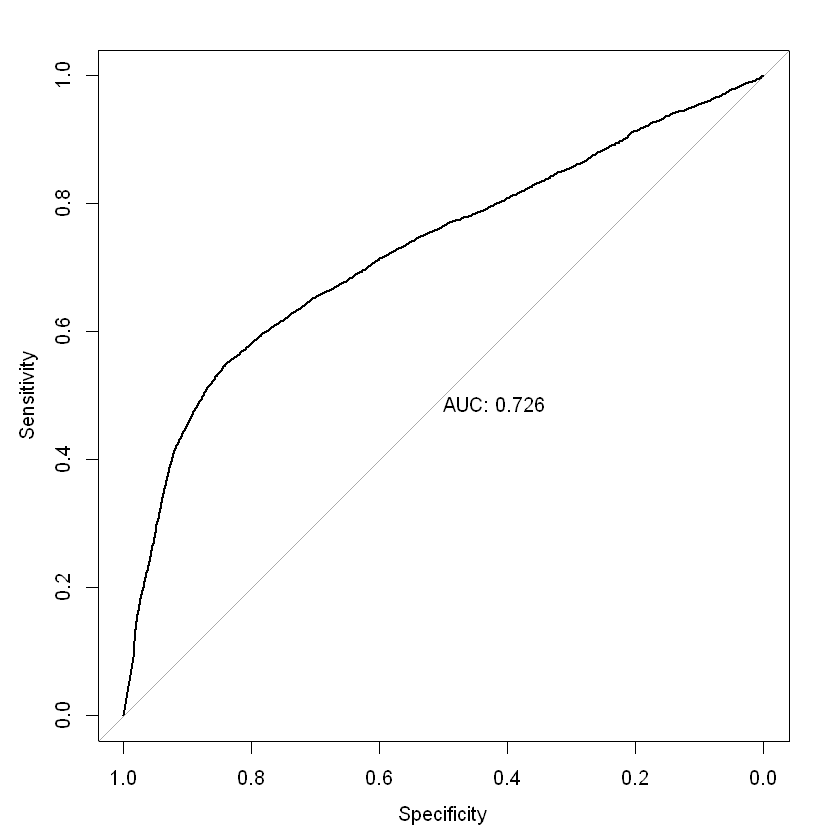
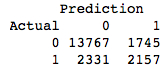
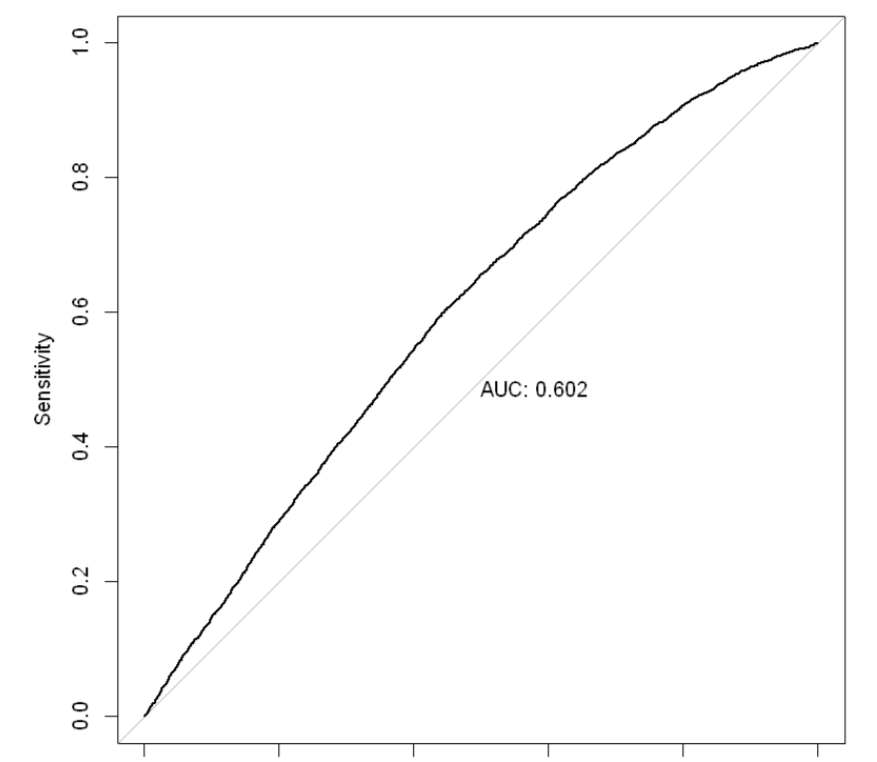
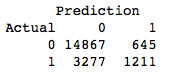
Weaknesses:

1. Requires observations to be independent of one another
2. Simple representation without opportunities for hyperparameter tuning

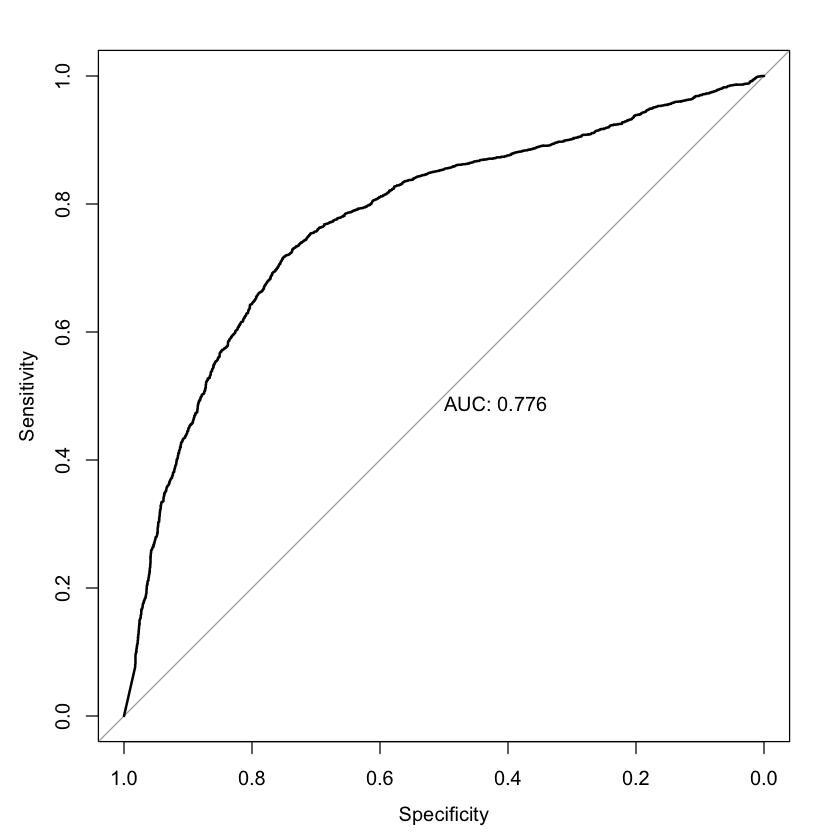
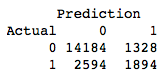
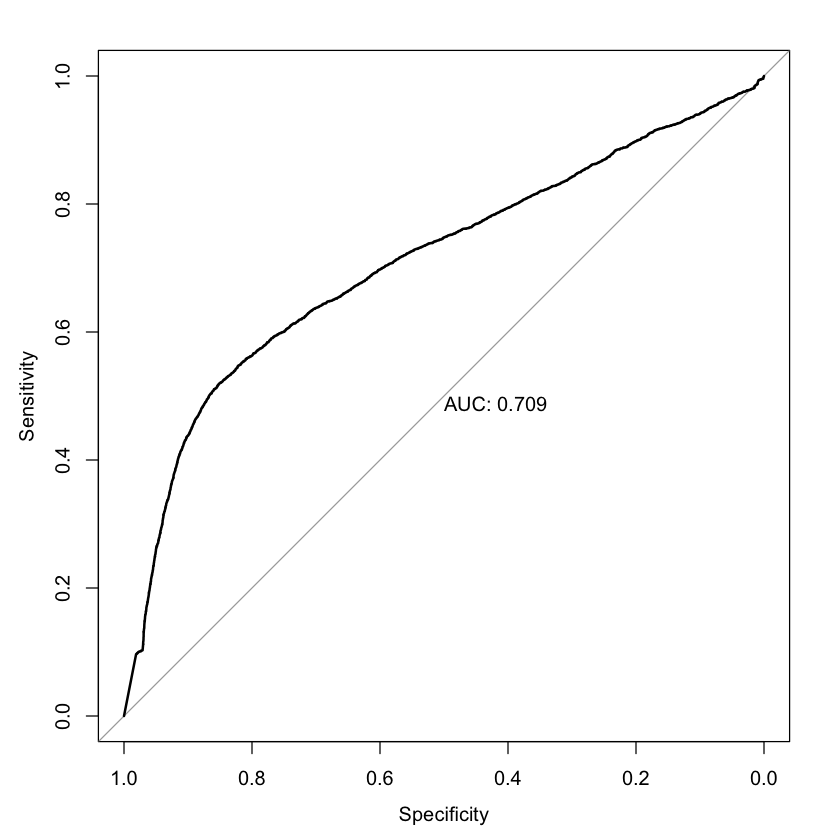
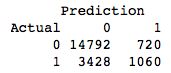
Suitability for problem:

1. Low number of observations - Naive bayes performs well on small dataset
2. Well separated categories and high-dimensional data

Original: Boruta:



After Balancing: Lasso:



**Summary Table for Naive Bayes Classifier:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Accuracy | AUC | Recall | Precision | F1-Score |
| Original | 0.8039 | 0.602 | 0.2698 | 0.6525 | 0.3818 |
| After balancing | 0.7926 | 0.709 | 0.2362 | 0.5955 | 0.3382 |
| Boruta + SMOTE | 0.7962 | 0.726 | 0.4806 | 0.5528 | 0.5142 |
| Lasso + SMOTE | 0.8039 | 0.776 | 0.4220 | 0.5878 | 0.4913 |

In this dataset, laplace smoothing is not required as there is no problem of zero probability.

Applying Boruta feature selection on the balanced data results in the best Naive Bayes model.

**4.2.3 Logistic Regression**

Industry Usage : Classification

Strengths :

1. Many ways to regularize model
2. Don’t have to worry about correlated features
3. Get p-values of features - easy removal

Weaknesses :

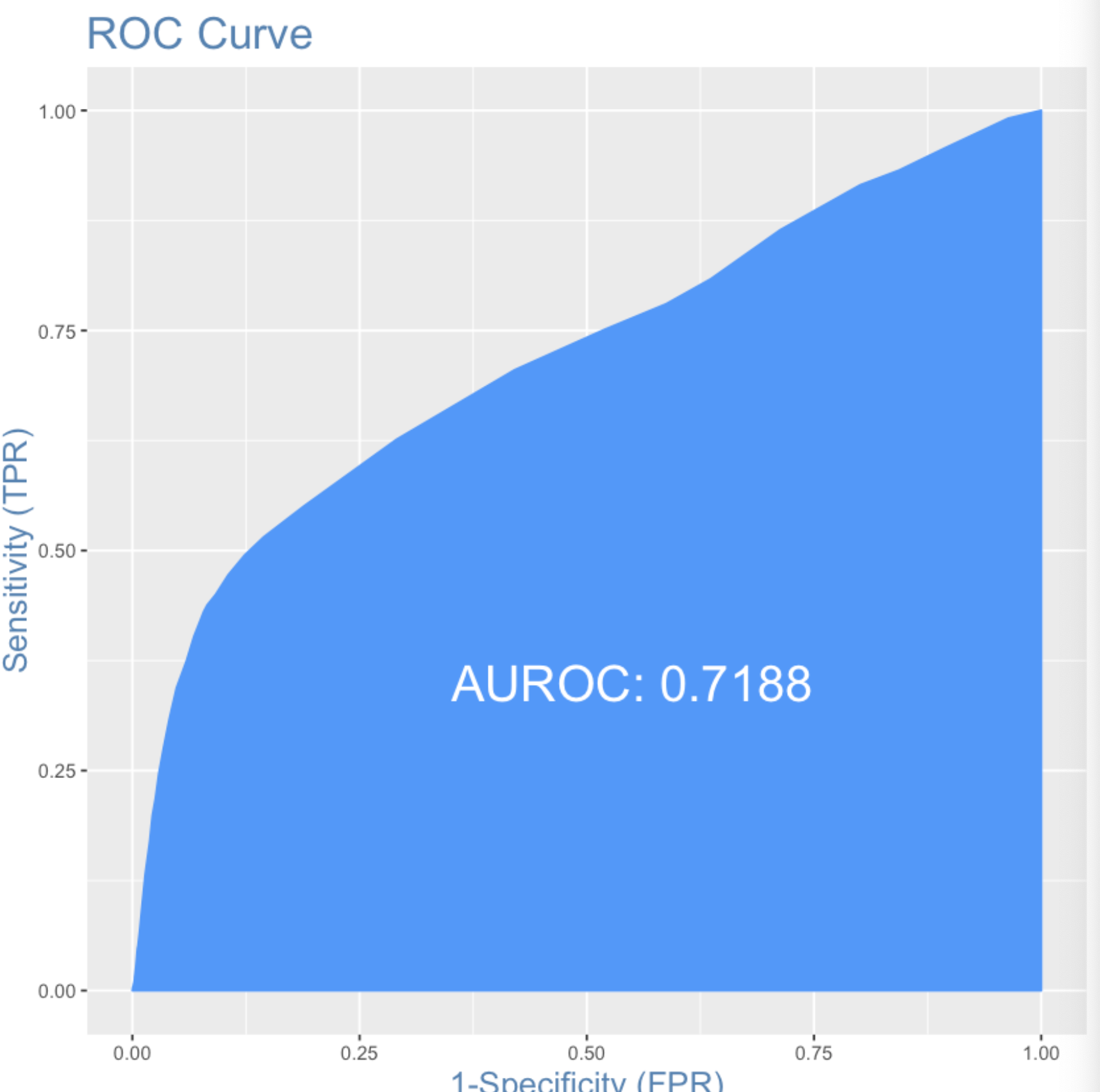
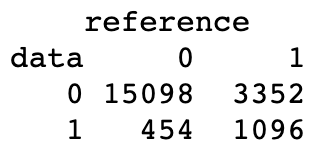
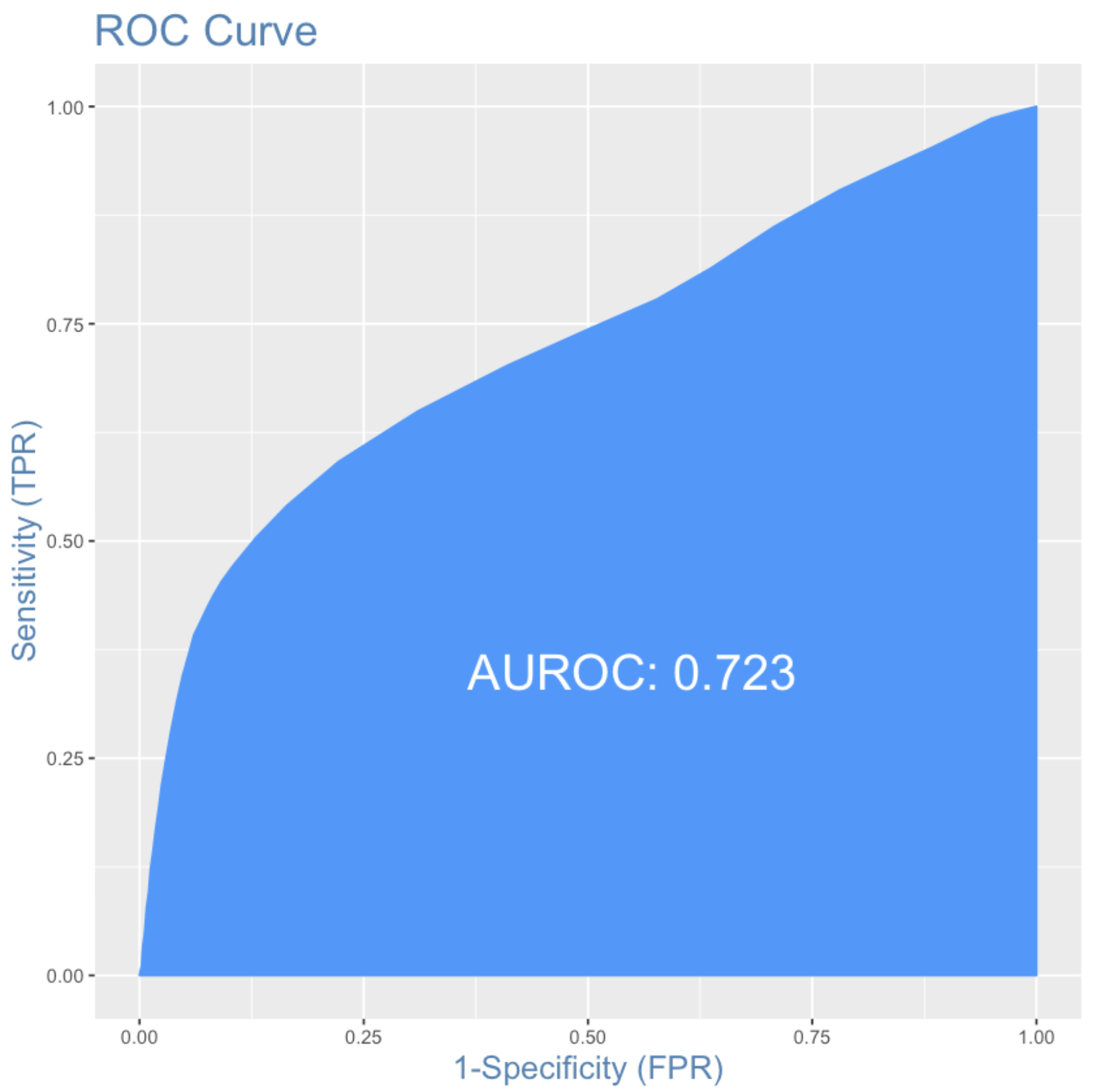
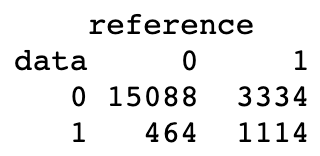
1. Requires observation to be independent of one another

Suitability for problem :

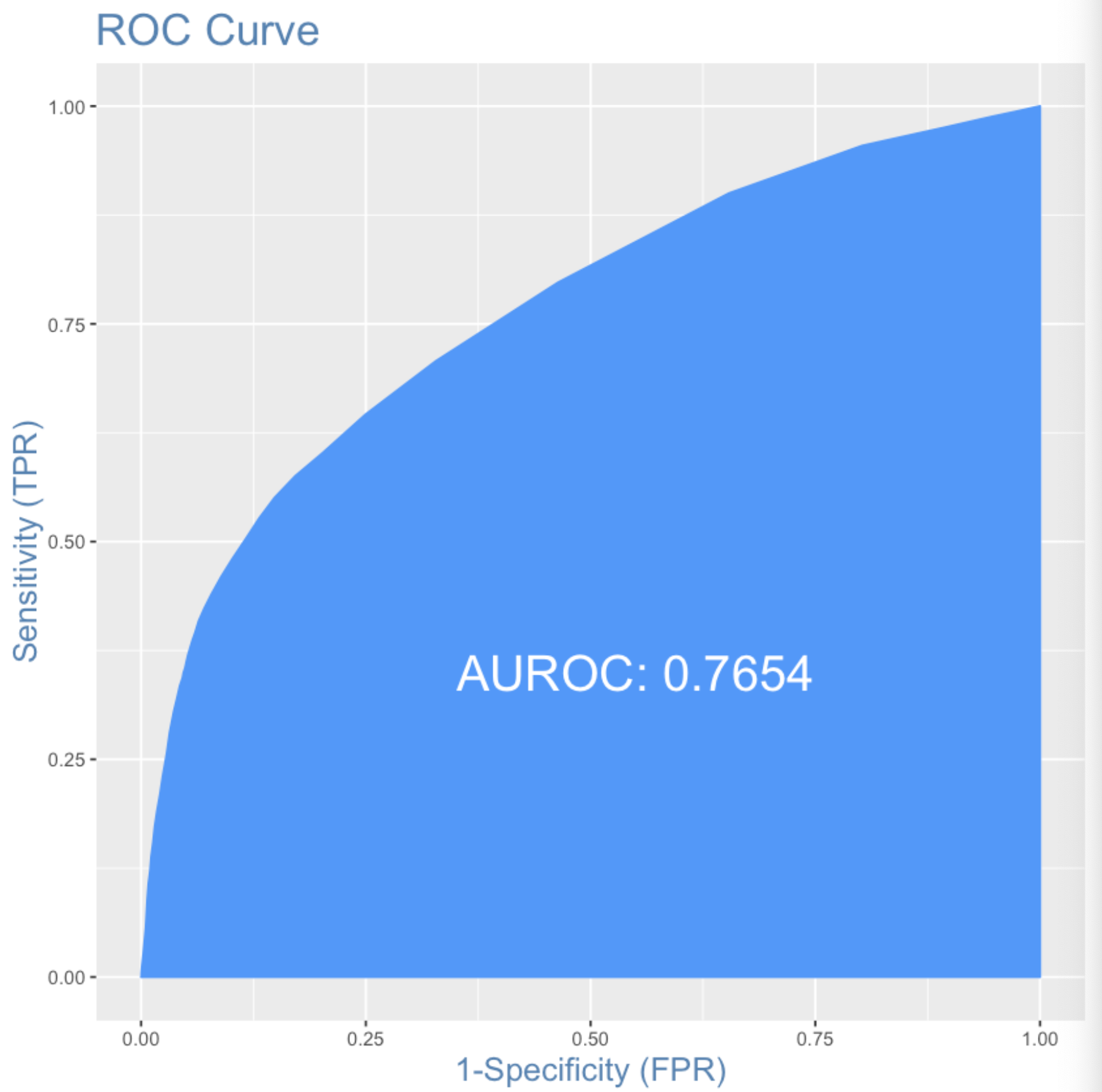
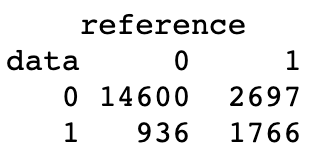
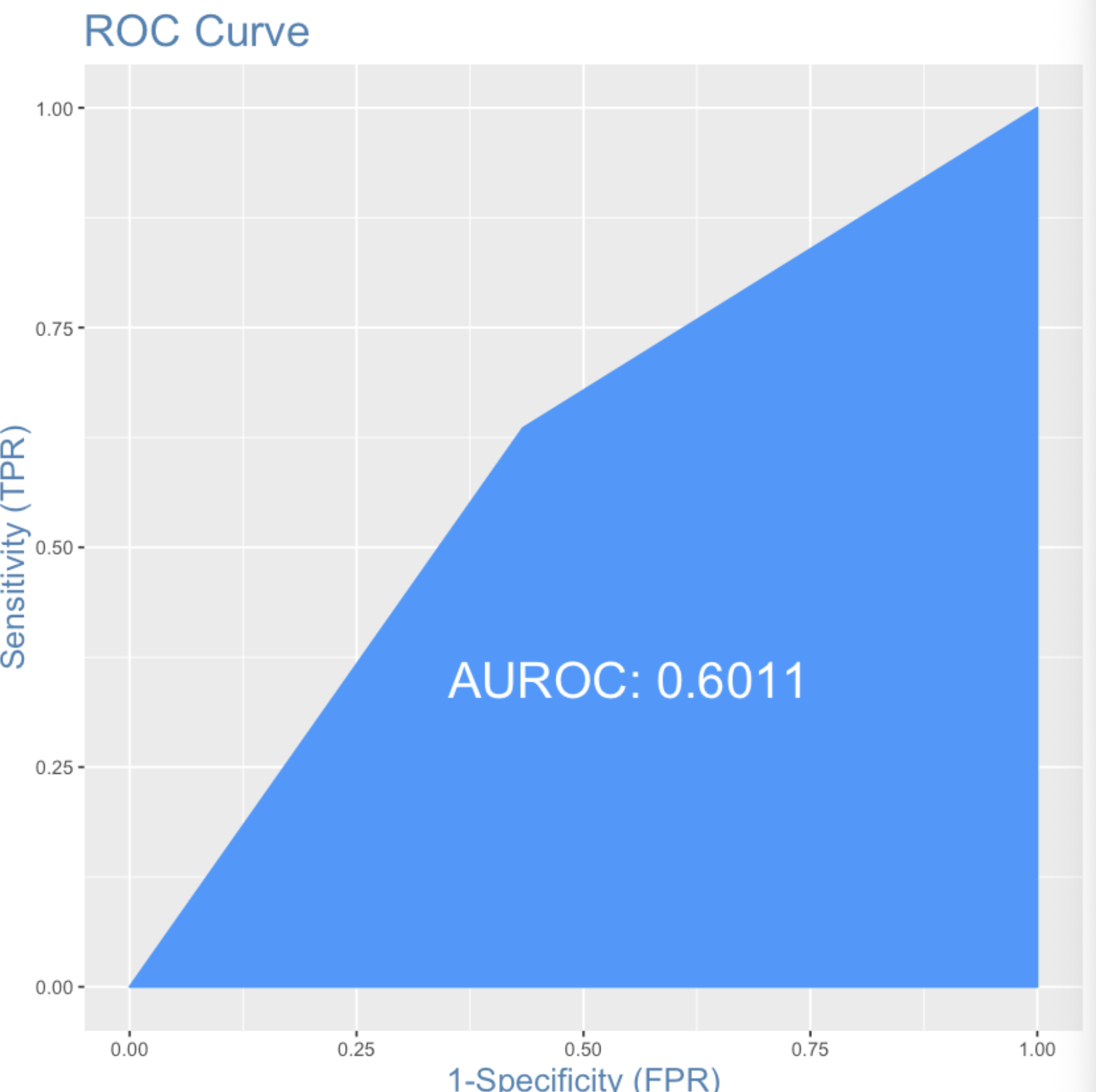
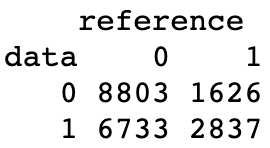
1. Regularization to prevent overfitting
2. Can easy find out which features are correlated

**Note that reference - actual, data - predicted**

|  |  |
| --- | --- |
| Original | Features selected using P-value |



|  |  |
| --- | --- |
| Balanced Dataset | Features selected using P-value |



**Summary Table for Logistic Regression:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Accuracy | AUC | Recall | Precision | F1-Score |
| Original | 81.00% | 0.723 | 25.04% | 25.5% | 0.3697 |
| P-Value selection | 80.97% | 0.7188 | 24.64% | 70.70% | 0.3654 |
| Balanced Dataset | 58.2% | 0.6011 | 63.56% | 29.64% | 0.4043 |
| Boruta | 81.83% | 0.7654 | 39.56% | 65.35% | 0.4929 |

**Limitations of Logistic Regression**

1. During the implementation of logistic regression on the dataset balanced using SMOTE, we get the warning “glm.fit: fitted probabilities numerically 0 or 1 occurred”. This indicates that a perfect fit is possible, resulting in probabilities at 1 or 0. In this case, we can consider utilizing a form of penalized regression.

**4.2.4 Random Forest**

Industry Usage : Classification and Regression

Strengths :

1. Able to deal with unbalanced and missing data
2. Fast runtimes

Weaknesses :

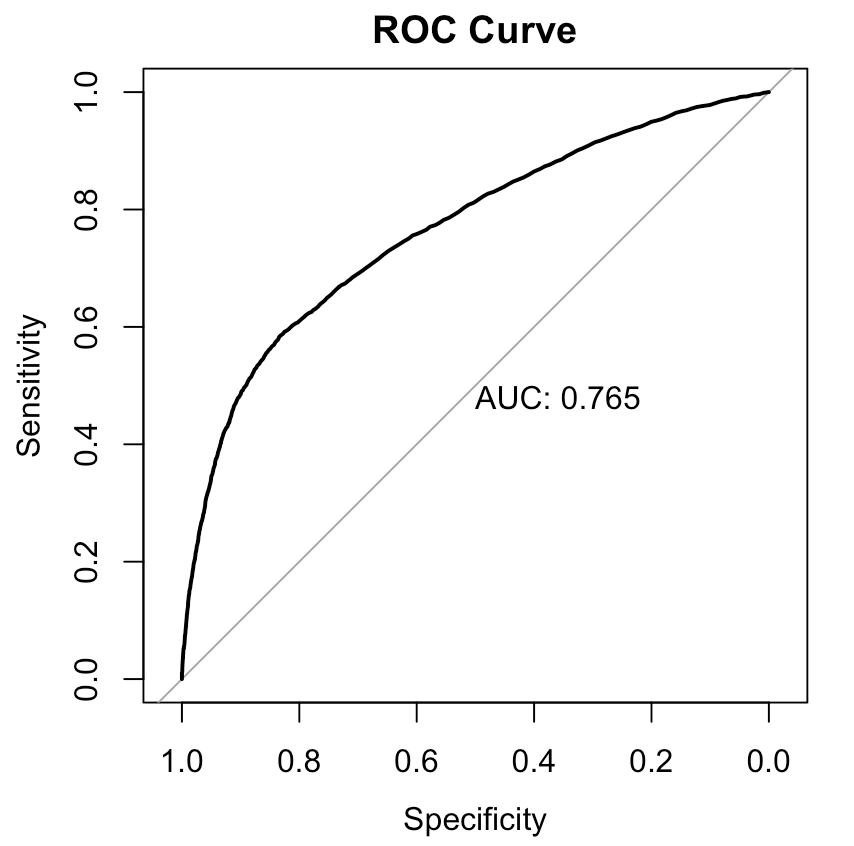
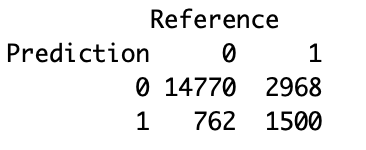
1. Cannot predict beyond range of training data
2. Tendency to overfit data-sets

Suitability for problem :

1. Unbalanced data

Original training data:

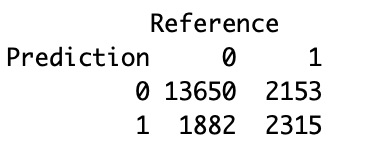
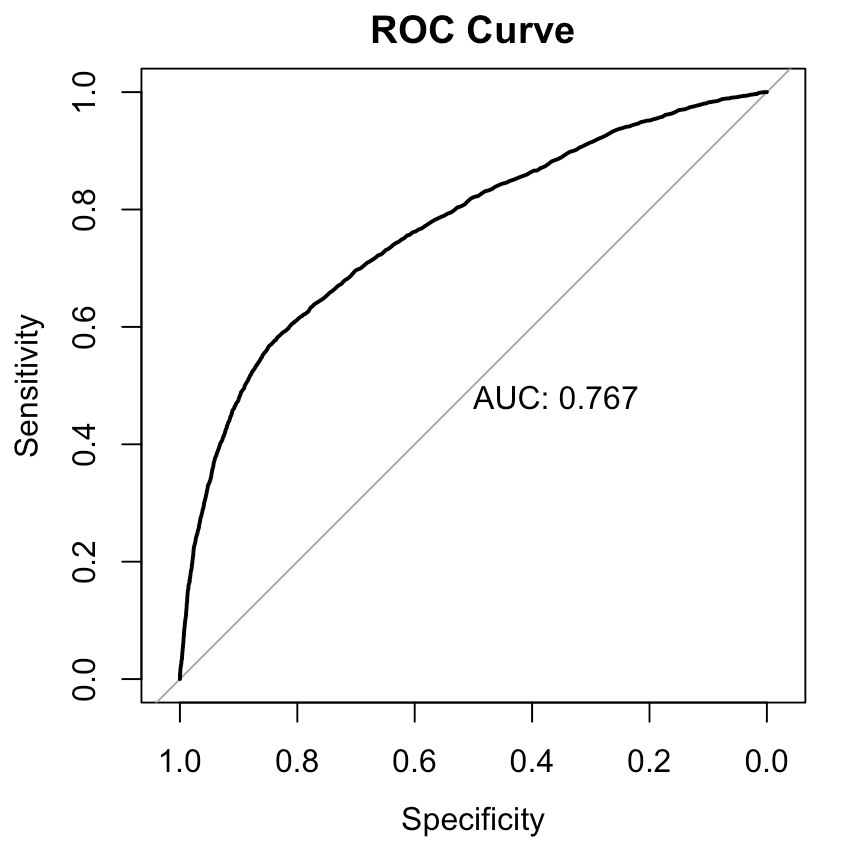
Confusion Matrix:



Applying RandomForest on the original training dataset produces an accuracy of 0.8152, precision of 0.67537, recall of 0.33979, AUC of 0.764 and F1 of 0.45212.

After balancing training data with SMOTE:

Confusion Matrix:



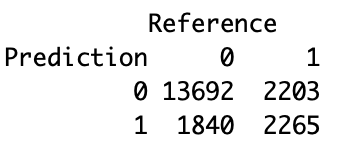
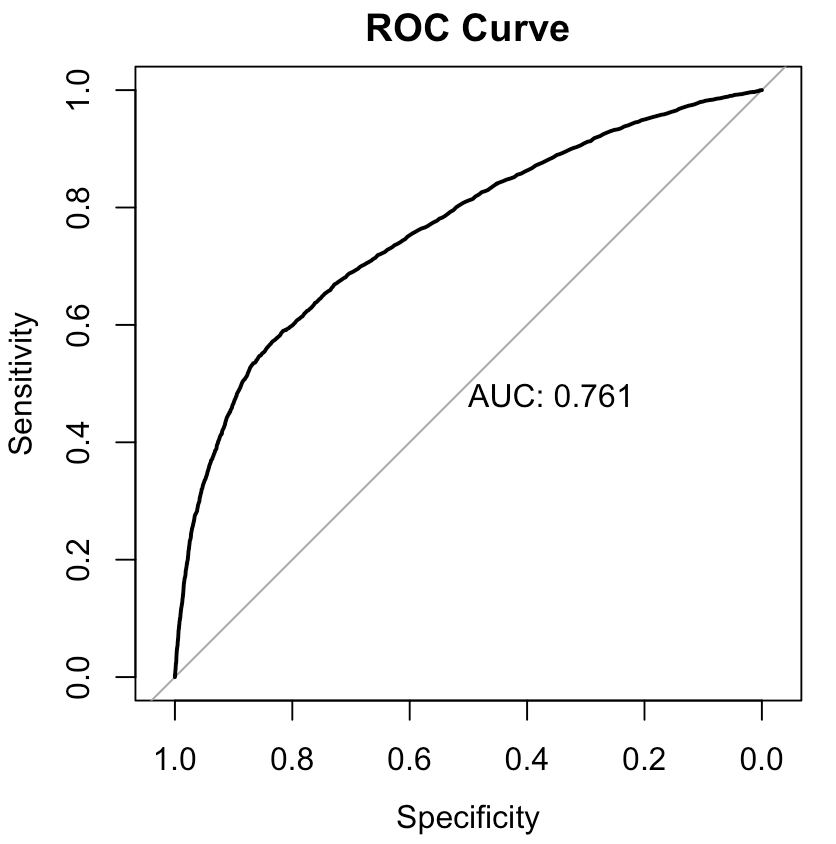
After balancing the training data, while accuracy and precision saw a slight drop, there is an improvement in the AUC and F1 score, that is 0.7666887 and 0.5343335 respectively.

While there is a drop in certain metrics, the improvement in AUC and F1 score outweighs them.

Comparing the original and balanced training data, the balanced training data is selected to run feature selection on.

After balancing with SMOTE + feature selection by LASSO:

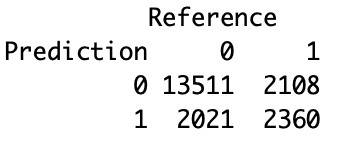
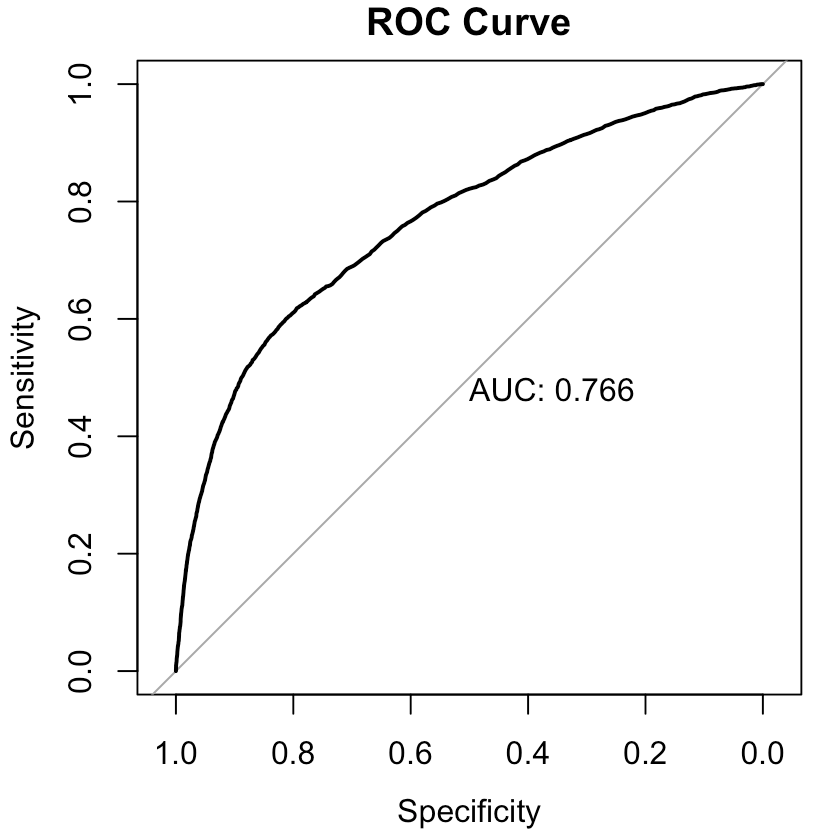
Confusion Matrix:



It is observed that the accuracy (0.79785), recall (0.5069382), AUC (0.7612923) and F1 (0.5284031) values have dropped with the selected features. Performance has worsened, thus we try doing feature selection again by Boruta.

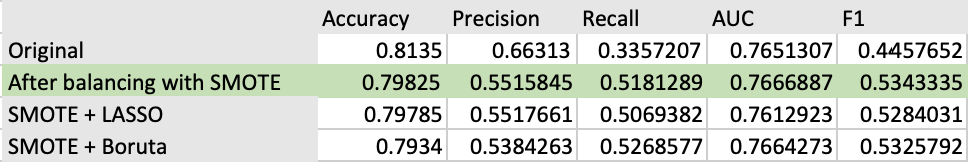
After balancing with SMOTE + feature selection by Boruta:

Confusion Matrix:



It is observed that the Accuracy (0.7934), Precision (0.5384263), AUC (0.7664273) and F1 (0.5325792) values have dropped slightly with the selected features by Boruta.

Summary:

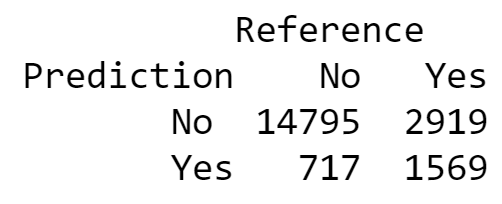
Therefore, optimal performance for RandomForest is achieved after balancing the training data with SMOTE, without feature selection.

**4.2.5 Tree Boosting (XGBoost)**

Boosting improves accuracy by having subsequent models which learn from previous errors, although there is a risk of over-fitting.

*Defaulters are represented as “Yes”, while non-defaulters are represented as “No”.*

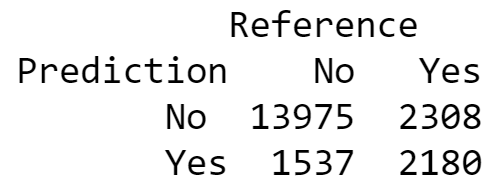
*Confusion Matrix (Original training data)*



Boosting on the unbalanced training data returned an accuracy of 0.8182, an f1 score of 0.46324, recall of 0.3496, and AUC of 0.7825

*After balancing the training data with SMOTE*

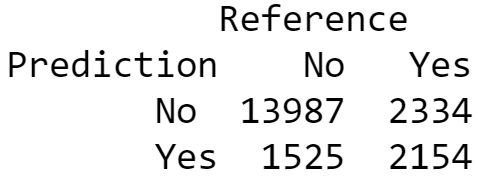
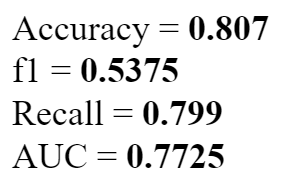
*Confusion Matrix*



After balancing the data, we can observe a marginal improvement with an f1 score of **0.5314** and recall of **0.4857**. While accuracy and AUC decreased slightly to **0.8078** and **0.7742** respectively, this is outweighed by the significant improvement in f1 score.

Using only important features selected by LASSO

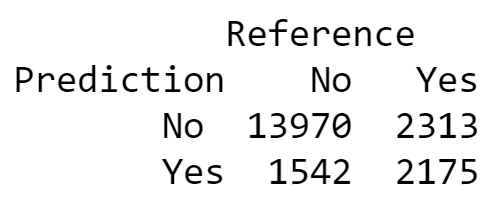
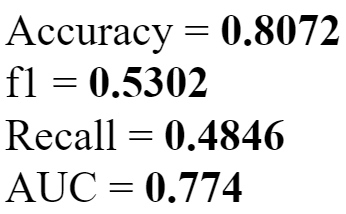
*Confusion Matrix*



All performance metrics above have not improved, thus we try again with Boruta’s features

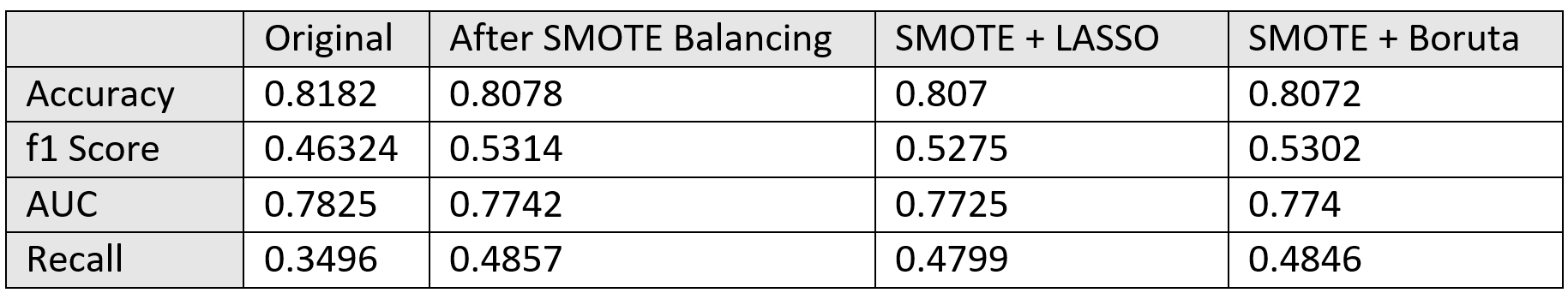
Using only important features selected by Boruta

*Confusion Matrix*



Using Boruta Feature Selection together with the balanced data results in optimal performance for our Boosting model.

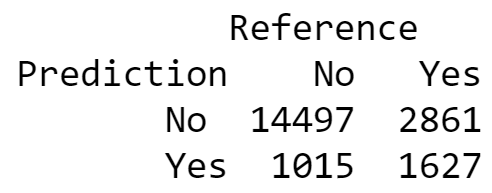
Summary Table (Boosting)



**4.2.6 Tree Bagging**

By training multiple trees with bootstrapped data, this would help to minimize variance and overfitting, as compared to regular decision trees.

*Confusion Matrix (Original unbalanced data)*



Accuracy = **0.8062**

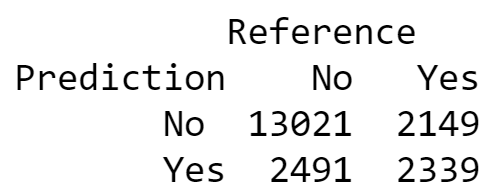
f1 = **0.45638**

Recall = **0.36252**

AUC = **0.7449**

After balancing the data with SMOTE

*Confusion Matrix*



Accuracy = **0.768**

f1 = **0.502**

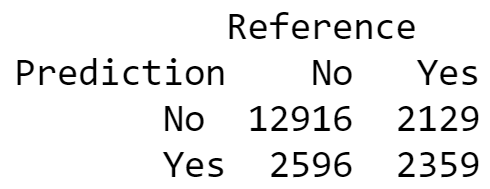
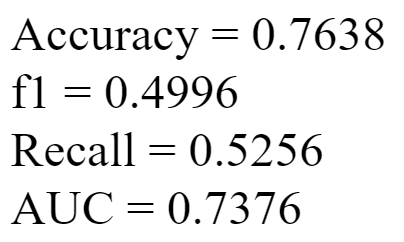
Recall = **0.5212**

AUC = **0.7401**

All of the performance metrics above have improved substantially aside from AUC, thus we can proceed with using the balanced data for our analysis.

Using only important features selected by LASSO

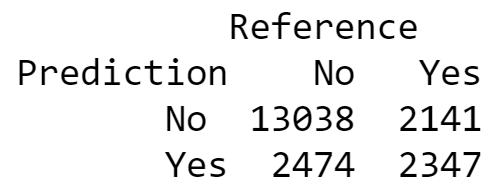
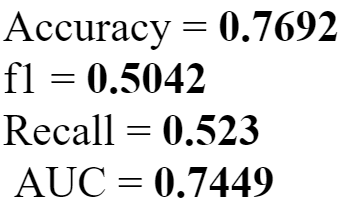
*Confusion Matrix*



All performance metrics above have decreased aside from recall, thus we will try again using Boruta’s selected features

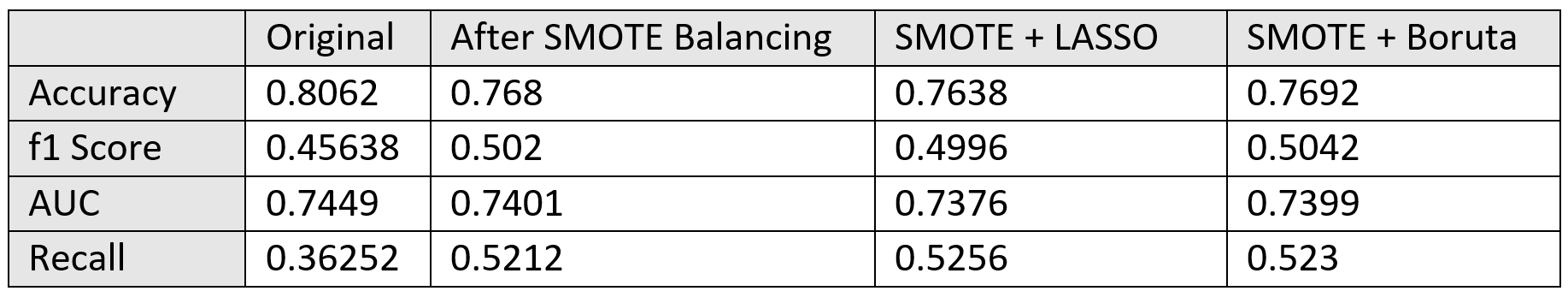
Using only important features selected by Boruta

*Confusion Matrix*



We can observe that applying Boruta Feature Selection on the balanced data results in the best performance for bagging.

Summary Table (Bagging)



Summary of best performing models

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | AUC | Accuracy | Recall | F1-Score |
| SVM - Polynomial kernel + SMOTE + Boruta feature selection | 0.679614 | 0.70085 | 0.641226 | 0.489200 |
| Random Forest + SMOTE | 0.7666887 | 0.79825 | 0.5181289 | 0.534335 |
| Boosting + SMOTE + Boruta | 0.774 | 0.8072 | 0.4846 | 0.5302 |
| Naive Bayes + SMOTE + Boruta | 0.726 | 0.7962 | 0.4806 | 0.5142 |
| Bagging + SMOTE + Boruta | 0.7399 | 0.7692 | 0.5256 | 0.5042 |
| Logistic Regression + Boruta | 0.7654 | 0.8183 | 0.3956 | 0.4929 |

While mis-classifying non-defaulters as defaulters may result in lost interest revenue, we assume that it is more important to identify potential defaulters correctly, as failing to repay loans would hurt the bank more. Thus, recall would be the most important metric, followed by F1-Score. Comparing the results of the 6 supervised machine learning models that we used, we can conclude that SVM is most suitable as it has the highest recall by far, although F1-Score is slightly lower.

**5. Improvements**

**5.1 Increasing train dataset**

One option is to look towards increasing our training set size. Currently, we are splitting the data into 10000 train data points and 20000 test data points. We can try to increase our train data by either allocating more of the test points or looking for similar data at hand or online to increase the train dataset size. Another great way to obtain new cases is to scrape data from the web or get it through an application programming interface(API).

**5.2 Detailed grid search; searching for the best hyper-parameter**Due to the time constraints of the project, we resorted to a loose grid search in our SVM model selection. While most algorithms perform fairly well out of the box using the default parameter settings, there is a possibility that we can achieve better results by testing different hyper-parameters. A finer grid search can be done and evaluated to see if it does increase our score metric. However, the more detailed the search, the longer the runtime, and the values selected may not be better than the default setting.

**5.3 Stacking models**Stacking is an ensemble method and can provide us with better performance. In stacking, we build the machine learning model in two stages. Firstly, we predict multiple results using different machine learning algorithms. We then take the results from the best models from all the various machine learning algorithms and input them into a final model. Certain libraries also provides automated machine learning where various stacking ensembles are ranked and displayed for the user.

**5.4 Feature Engineering**Although we’ve explored a few methods of feature engineering such as balancing the dataset, boruta and lasso, some techniques might work better with some algorithms. We also could explore using binning and log transform for feature engineering and subsequently test different techniques with different models to obtain the most optimal model.

**5.5 Domain knowledge**

Furthermore, we have lacked the resources to consider the impact of a loan defaulter versus someone that did not default. For example, the end goal of this model is to reduce loan defaults, hence reducing our losses. Hence, our focus was on maximizing recall. However, we did not take into account the potential revenue gain and magnitude of loss for each default and non-defaulter. Given an opportunity, we might wish to move beyond binary classification and formulate a new dataset with new target variables, with the help of a domain expert.

Lastly, the input data we have used consist of information ranging from the personal details of the client to their payment status over six months. By then, it is too late for the bank to ‘retract’ the loan - selling it off is the only way to recoup losses. Hence, this bodes the question if we should have applied our statistical learning models solely to information we have before the loan is initiated. (Eg. Age, Income, Gender, Education, Marriage).