Predicting Loan Default Risk

Group 8

Erica Augustyniak
Karim Al Zeer Alhusaini
Kindeep Dhatt
Mathai Paul

December 11, 2019

Abstract

It is critical for any lending institution to assess the default risk of its potentials customers before offering any form of credit. Our aim is to best predict the risk of default given a number of features. In this project, we obtained a large data set containing features on many clients. We cleaned and transformed the data into a manageable format for modeling. We then explored the different features and extracted all the relevant correlations and links between the features. After selecting the strongest features, we moved into building multiple classification models, including Logistic Regression, Linear and Quadratic Discriminant Analysis, kNN, Random Forest, and Gradient Boosting Machine (GBM). We analyzed the performance of each model based on the ROC AUC, Sensitivity and Specificity. We found that the GBM model is the most predictive model among the above models. We concluded by proposing further analysis and improvements on the models to increase the accuracy and predictive ability.

1 Introduction

We pulled our data set from a past competition on Kaggle in which a financial institution, Home Credit, challenged Data Scientists to use its open source data to predict customer default or inability to make regular payments on a loan. The company provided different in-house and external credit attributes on 307,511 clients in an unnamed country. This is a list of existing clients who previously applied and were approved for credit with the company. The company already has financial and payment history records on these clients and is looking to create a scoring model that scores and assigns a probability of default for each customer instantaneously once they apply for new credit with the company.

There was a wide range of data, containing personal information on the client, employment history, income, address, and hundreds of normalized credit attribute data. The aim was to best utilize the data to better predict customer default. It is essentially a classification problem with a target variable of 1 (default) and 0 (non-default). We decided to apply the different classification techniques we learned through this course and assess which method is best at predicting default. Furthermore, we applied two additional techniques outside of the course content that we thought can help improve our models.

Problem statement: The problem we addressed with this project was how to predict the probability of a client defaulting on a loan given the client characteristics in the data set. We built several classification models in order to determine the model with the best predictive capability.

The criterion used to assess the models was the Area Under the Curve (AUC) for the Receiver Operating Characteristics (ROC) curve. Our goal is to get the highest possible AUC value out of the data set. We also considered the Sensitivity (True Positive Rate) and Specificity (True Negative Rate) of each model. The main tool use in this analysis was R. We also used SQL to transform the data. Additionally, we used Python and Tableau to do the Exploratory Data Analysis.

2 Data

We used a data set from Home Credit, found on Kaggle as part of a competition. The main table *application* contained over 122 different features that could be used to predict whether a client would default on their loan payment or not. The variables in the data set included the loan type, loan amount, characteristics about the client such as their income, marital status, employment status, age, household, and others. In addition to the main table, the data provided included additional larger transactional tables which contained data on the client's past application loan behavior. In total, there were 307,511 records in the raw data with each row being an individual loan. Appendix A1 shows the Entity Relationship Diagram ERD of the provided data set along with descriptions of each table

We loaded all the tables into a SQL Server Database, then joined the main *application* table with the *previous_application* and *POS_CASH_balance* tables. From the *previous_application*, we pulled the number previous applications the client has had with the company. This is to see if the client has repeatedly applied for credit but was rejected. Similarly, from the *POS_CASH_balance* table, we pulled the maximum number of days past due the client may

have had on a previous or existing loan. Given the limited computing power we had, and the size of the data, we decided to use only one table that contained enough features to build a model.

The main *application* was split into a training set and a validation (testing) set. The idea was for the test table to be used for validating the models. However, since the test table was not labeled, we used only the training table and separated it into a training set and a testing set with a 70:30 random split in R.

3 Exploratory Data Analysis

The final training table contained 124 features. Most of these features were generalized census attributes that consisted mostly of null values for most clients. We therefore decided to drop most of the features in the table and focus on only the relevant features.

The next step was to manipulate the features in the final set. Some of the features are nominal, meaning they are categorical variables that have no intrinsic order such as gender and marital status. Other features are ordinal, meaning they are categorical with an intrinsic order, such as education level. The remaining were numeric values. We converted all the categorical values into binary dummy variables with values of (0,1)

Likewise, we added two new features based on our domain knowledge. The first feature we added was monthly payment as a percentage of monthly income. We added this feature believing the debt-to-income ratio can have an impact of the loan repayment and ultimately. default. The second feature was the length of the loan or the term. We were able to extract both features from the data provided. With that, we had we had a complete set of 23 features to be used in our analysis. Appendix A2 details the feature names and their descriptions.

In examining the Target variable, we found that approximately **8%** of the final data set were positively labeled as default.

3.1 Correlation Matrix

Once the data set has been finalized, we can transition into understanding the different correlations between the different features. We build a correlation matrix which shows the Pearson Correlation coefficient. **Figure 1** below displays the correlation matrix.

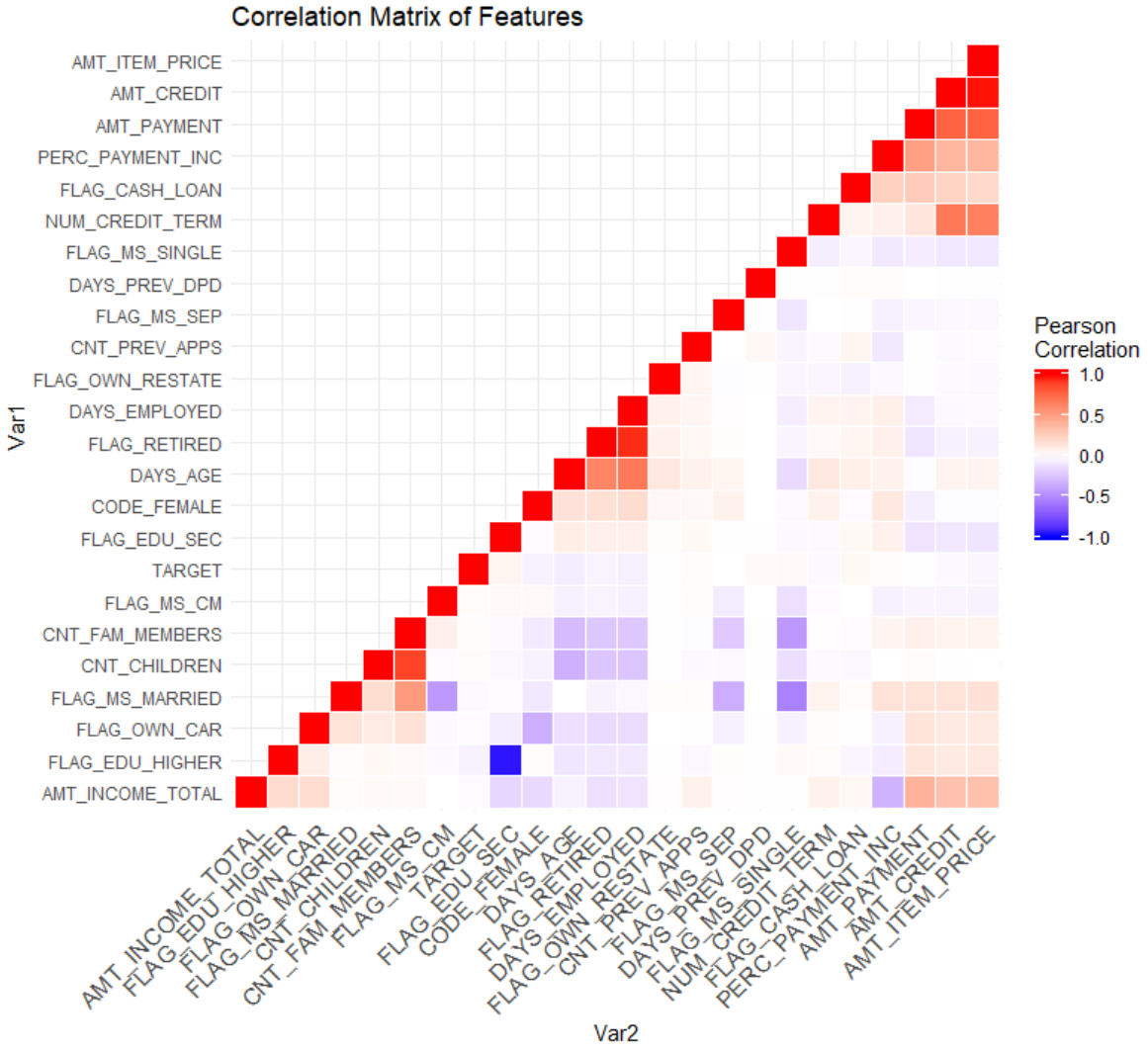

Figure 1: Correlation matrix of the final data set features

It seems the target variable does not have many strong correlations with the other features. Nonetheless, there is small negative correlation between age and default risk. In other words, the older the client is, the less likely they are to default. Similarly, there is a negative correlation between days of employment and risk of default which means there is a smaller risk of default associated with a longer period of employment.

There also seems to be a strong correlation between the price of the item that is financed, the loan amount, the payment amount, and the payment as a percentage of monthly income. To avoid falling into a co-linearity trap, some of these variables will be dropped when the models are built.

3.2 Age Deep Dive

For a deeper look, we examine the client age distribution. **Figure 2** displays the age distribution by years. We noticed, there are no anomalies and the distribution is as expected. Nonetheless, we add the default rates by age, we notice that younger people tend to have a higher chance of default, especially around the age of 30. **Figure 3** displays the age distribution and the default rate.

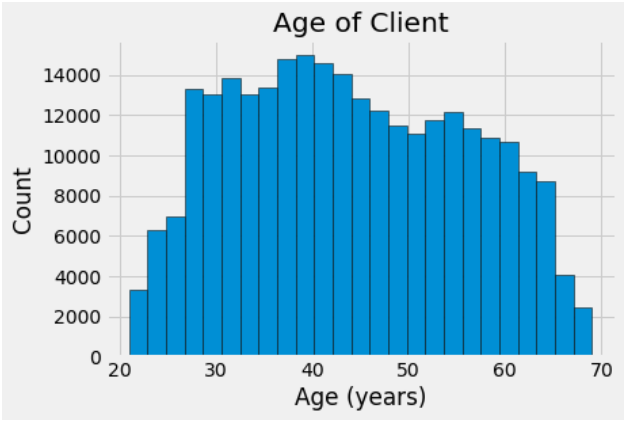

Figure 2: Client Age Distribution

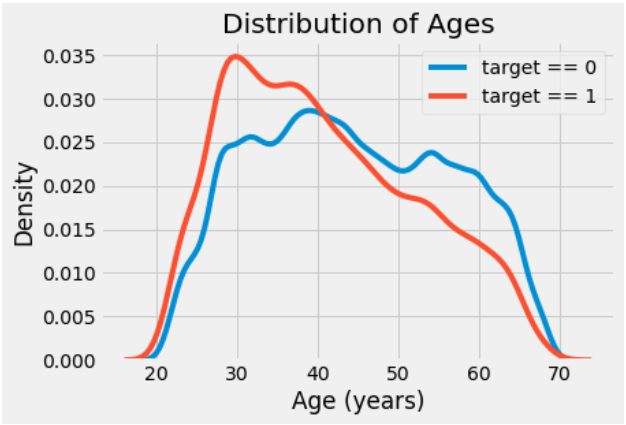

Figure 3: Default Rate by Age

4 Models

Since the response variable is a binary categorical variable, we built different classification models to predict the likelihood of default on a given loan. The models built included:

1. Logistic Regression.
2. Linear Discriminant Analysis.
3. Quadratic Discriminant Analysis.
4. k Nearest Neighbors (kNN), using k = 3,5,7
5. Random Forest
6. Gradient Boosting Machine (GBM)

4.1 Logistic Regression with all variables

The first model we ran was a logistic regression model. A logistic regression model computes the likelihood of the target variable as demonstrated in the equation below.

$$\hat{p}(\boldsymbol{x}) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X_1+...+\hat{\beta}_{23} X_{23}}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X_1+...+\hat{\beta}_{23} X_{23}}}$$

where $\hat{p}(\boldsymbol{x})$ is the likelihood of default and $x = (X_1, X_2, ..., X_{23})$.

After we ran the model, the AIC on the model was 101799. **Appendix A3** contains the full summary of the regression. We then tested this against the validation data set. **Figure 4** below displays the probability distribution of predictions.

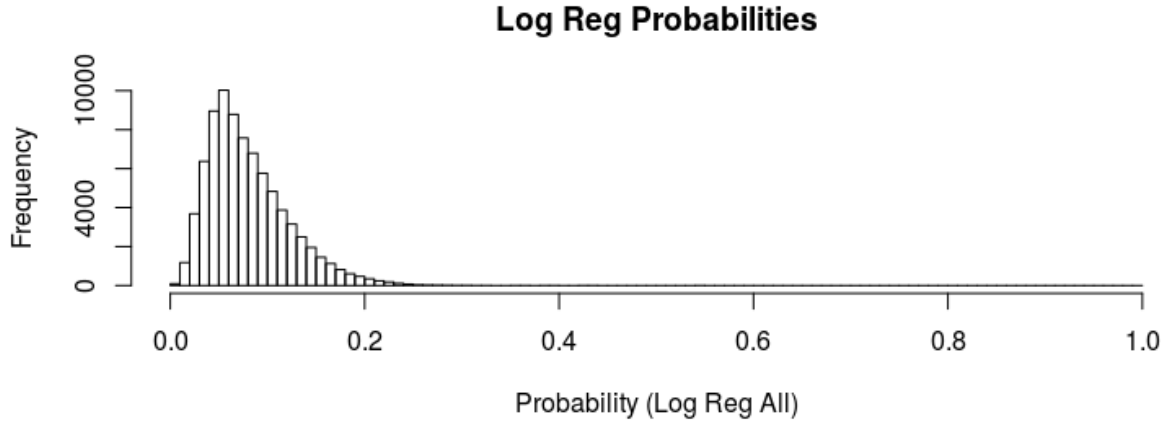

Figure 4: Predictions Probability Distribution: Log Reg

Based on the probabilities, we used a 12% threshold to to categorize the probabilities i.e. any prediction that is over 12% is considered default. **Table 1** below displays the confusion matrix attributes detailing the AUC and the accuracy of the model. **Figure 5** below displays the ROC curve for this model.

Measure	**Value**
AIC	101799
AUC	0.6607
Accuracy	86.02%
Sensitivity	91.75%
Specificity	20.79%

Table 1: Confusion Matrix Results: Logistic Regression All, validated against Test Data

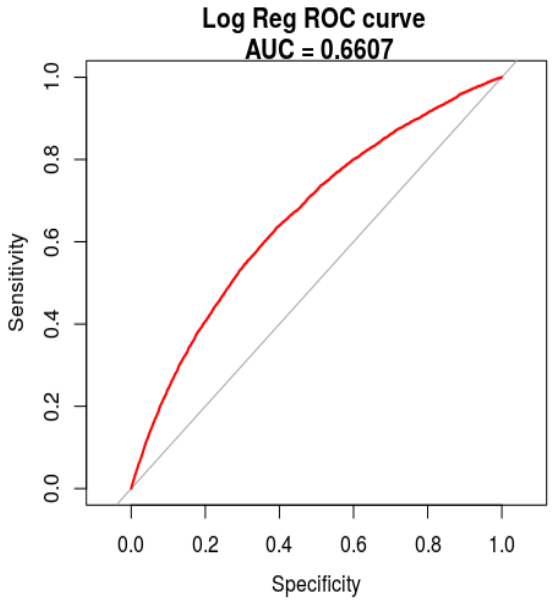

Figure 5: Full Logistic Regression ROC

The Specificity value above is low due to the response variable containing around 92% non-default cases, which causes an imbalance. We can, however, increase this value by lowering the threshold. This in turn will decrease the Sensitivity value. **Table 2** below shows the confusion matrix when the threshold is moved to 10%. The model's overall accuracy has dropped, so did the Sensitivity, while the Specificity has significantly increased.

Measure	**Value**
Accuracy	72.22%
Sensitivity	74.38%
Specificity	47.48%

Table 2: Confusion Matrix Results: Logistic Regression All, threshold 10%

4.2 Logistic Regression with step-wise feature selection

In order to determine the variables that produce the best model, we used the '*stepwise*' regression technique; the model with the lowest AIC (Akaike Information Criterion) is the model we chose to analyze. This helped us determine which variables within our data set that were significant in predicting the likelihood of defaulting on a loan. The final model we used, after removing all insignificant variables, is shown in the **Figure 6** below. Descriptions of the different variables can be found in **Appendix A2**.

```
TARGET(0,1) ~ FLAG_CASH_LOAN + CODE_FEMALE + FLAG_OWN_CAR +
AMT_INCOME_TOTAL + AMT_CREDIT + AMT_PAYMENT + AMT_ITEM_PRICE +
FLAG_EDU_HIGHER + FLAG_EDU_SEC + FLAG_MS_SEP + FLAG_MS_CM +
FLAG_MS_SINGLE + DAYS_AGE + FLAG_RETIRED + DAYS_EMPLOYED +
CNT_PREV_APPS + DAYS_PREV_DPD + PERC_PAYMENT_INC +
NUM_CREDIT_TERM
```

Figure 6: Features selected to run models

Once we selected the features, we ran the Logistic Regression Model again. **Appendix A4** shows the full results of this regression. As shown in **Table 3**, the AIC is actually lower than that of the full model. Additionally, the accuracy is less than that of the full model, at 12% threshold, though the Specificity is higher. **Figure 7** below displays the ROC curve for this model. The AUC value is the same as that of Logit model with all the features.

We can deduce that the reduced model is no better at predicting default than the full model. It is better however, at predicting Negative values, as demonstrated in the higher Specificity (True Negative Rate).

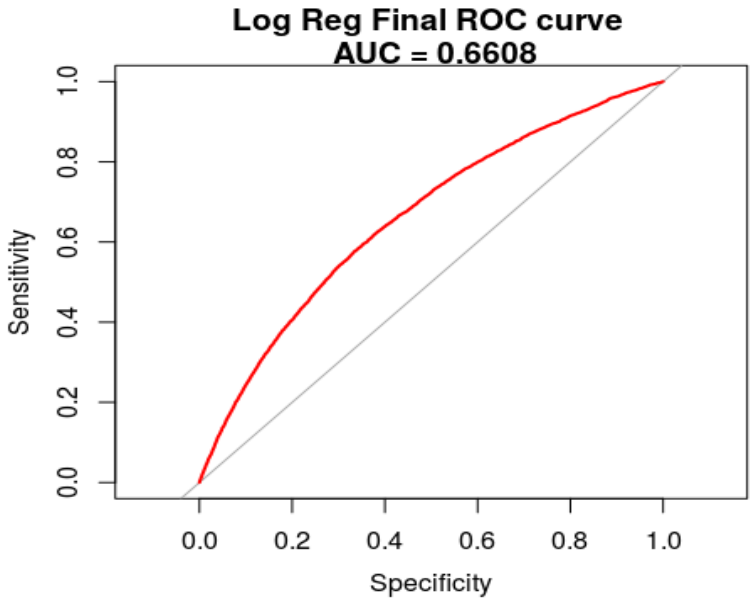

Figure 7: Reduced Logistic Regression ROC

Measure	Value
AIC	101699
AUC	0.6608
Accuracy	80.73%
Sensitivity	84.90%
Specificity	32.94%

Table 3: Confusion Matrix Results: Logistic Regression Reduced, validated against Test Data

4.3 Linear Discriminant Analysis

We proceeded to create a Linear Discriminant Analysis model. The purpose of using this technique is to test how accurate it can be at predicting default. The LDA model assumes the feature covariance matrices of both classes are the same, which results in a linear decision boundary. The LDA and QDA models work best when there are more than two classes, which is not the case in this paper. Therefore, we do not expect the results of the models to be accurate.

In R, we ran the model using the same features used in the reduced Logit model, described in **Figure 6** above. **Figure 8** below shows the ROC curve for the LDA model. **Table 4** is the confusion matrix of the posterior probability and the Target variable of the Test Data.

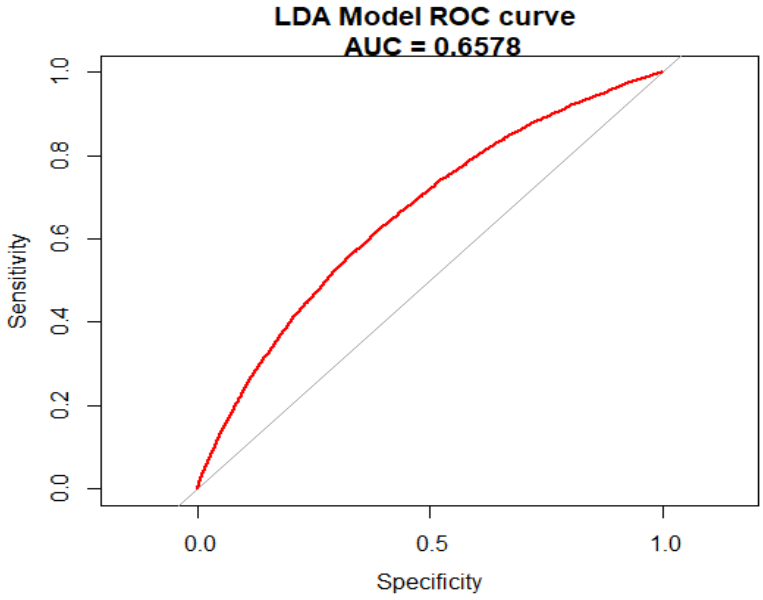

Figure 8: LDA ROC

Measure	Value
AUC	0.6578
Accuracy	91.96%
Sensitivity	99.97%
Specificity	0.0%

Table 4: Confusion Matrix Results: Linear Discriminant Analysis

The AUC of the LDA model is not drastically lower than the Logit models. However, the accuracy is quite high. The model is great at predicting Positive cases, but incapable of predicting Negative cases.

4.4 Quadratic Discriminant Analysis

As with the LDA model, we also tested a QDA model to compare the accuracy of both. Unlike the LDA model, QDA is less strict and allows different feature covariance matrices for different classes, which leads to a quadratic decision boundary. As mentioned previously, we do not expect the results of the models to be accurate since this data has a target variable

of two classes.**Table 5** shows the confusion matrix results and **Figure 9** shows the ROC curve.

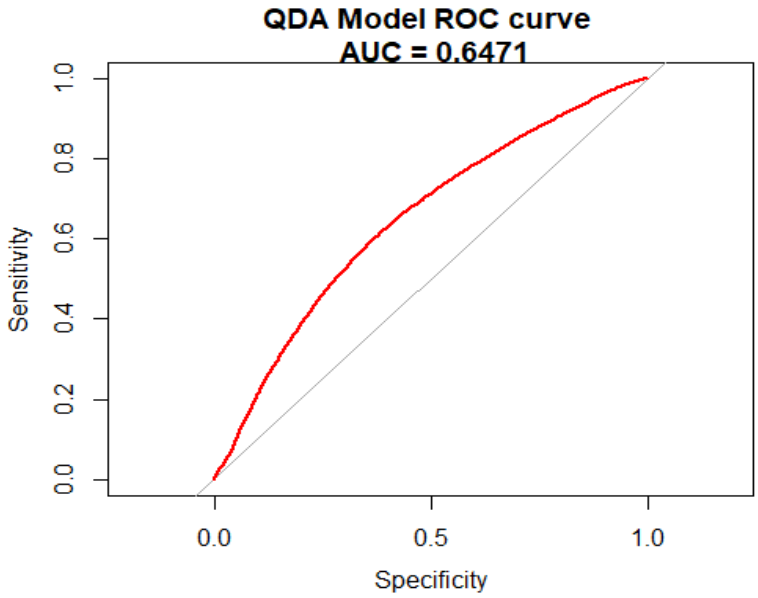

Figure 9: QDA ROC

Measure	Value
AUC	0.6471
Accuracy	89.99%
Sensitivity	97.31%
Specificity	4.77%

Table 5: Confusion Matrix Results: Quadratic Discriminant Analysis

It does appear that the QDA model has a slightly lower AUC, and is marginally less accurate than the LDA model.

4.5 kNN Model

For the k Near Neighbors model, we first had to normalize the data in order to properly run the model. We gain used the same features as with the reduced Logit model. The normalization function used was, for a given value (x):

$$normalized(x) = \frac{x - min(x)}{max(x) - min(x)} \tag{1}$$

Furthermore, we used 3 different values for k, 3,5,7. Then, calculated which value produced the highest accuracy. **Table 6**, **Table 7**, and **Table 8** show the confusion matrix results for k=3,5,7 respectively.

Measure	Value
Accuracy	90.02%
Sensitivity	97.63%
Specificity	4.42%

Table 6: Confusion Matrix Results: kNN Method, k=3

Measure	Value
Accuracy	91.34%
Sensitivity	99.11%
Specificity	2.14%

Table 7: Confusion Matrix Results: kNN Method, k=5

Measure	Value
Accuracy	91.71%
Sensitivity	99.60%
Specificity	1.12%

Table 8: Confusion Matrix Results: kNN Method, k=7

From the results above, it seems the higher the k value, the more accurate the model is. The Specificity decreases with each additional k value.

4.6 Random Forest

Once we ran the basic classification models, we decided to experiment with additional models that could be more suitable for such data set. The first of these models is the Random Forest. A Random Forest creates an ensemble of decision trees on the training set, then produces the mode of the classes of all the decision trees. One benefit of using a Random Forest over a single decision tree is to avoid falling into the over fitting trap.

We ran the model in R, without inputting a limit on the number of trees. However, as show in **Figure 10**, the error rate decreases on the first 20 trees then remains flat at near 0.

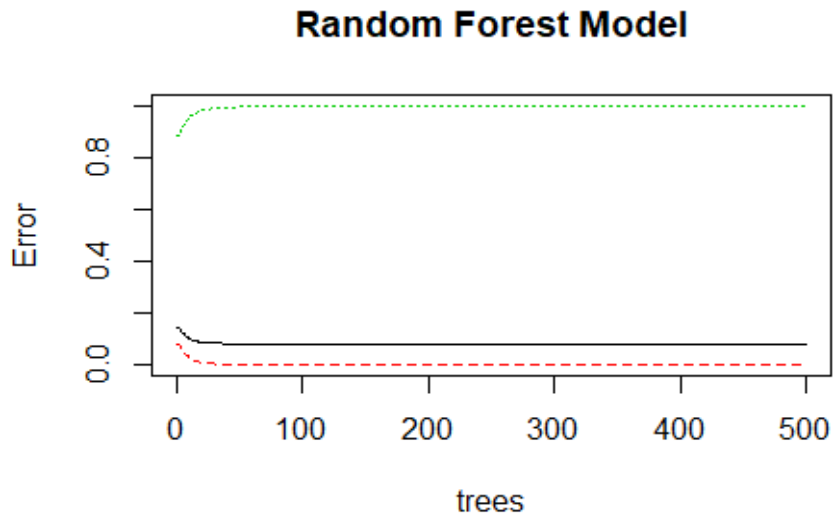

Figure 10: Random Forest number of trees and error rate

Figure 11 below shows the features that are the strongest default predictors. It shows the Mean Decrease Gini (MDI) of each predictor. The higher the value, the more important the feature is at predicting default. From the figure below, we can see that age, length of employment, and loan payment as a % of income are among the strongest predictors. Conversely, education in general is among the least accurate predictors.

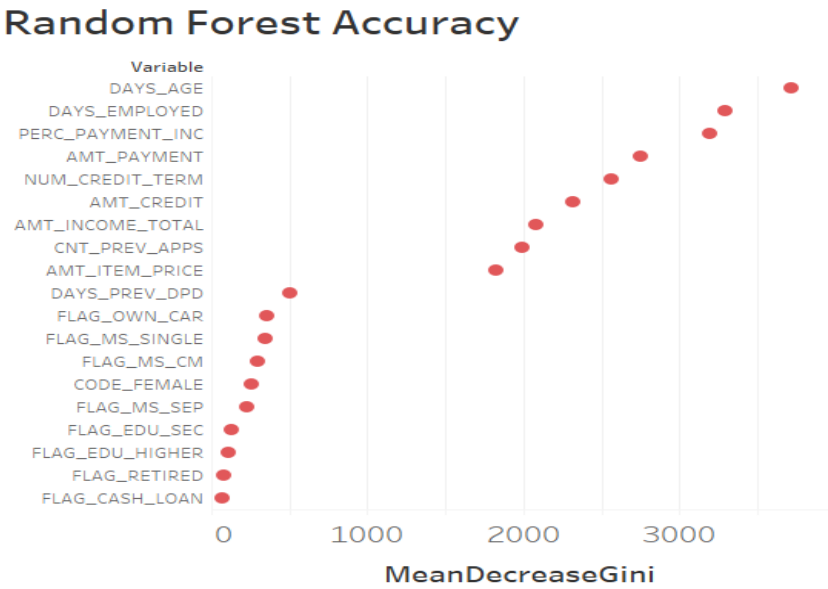

Figure 11: Random Forest: Mean Decrease Gini

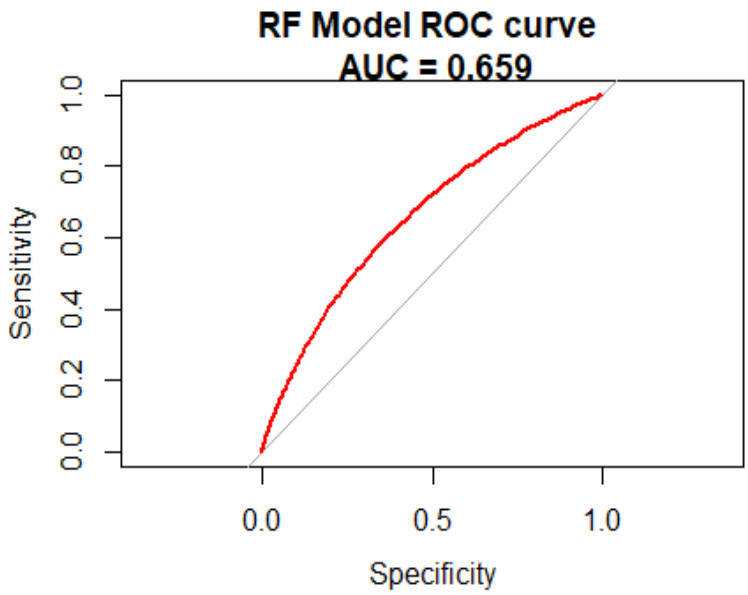

Figure 12: Random Forest ROC

Measure	Value
AUC	0.659
Accuracy	75.32%
Sensitivity	78.10%
Specificity	43.29%

Table 9: Confusion Matrix Results: Random Forest

Similarly, as shown in **Table 9** above, the AUC of this model is very close to that of the Logistic Regression Models. Nonetheless, the model is overall less accurate at 12% threshold. This is largely due to the model being a good predictor of Negative cases while remaining a solid predictor of positive cases.

4.7 Gradient Boosting Machine

The last model we used as the Gradient Boosting Machine, which is quite popular for such problems. At its essence, it is a method that uses weak learners then boosts them to become better predictors. There are different variations of this method, such as AdaBoost and XGBoost. However, in this model, we used a simple GBM function in R. **Appendix A6** shows the fucntion and parameters used to run the model.

Measure	Value
AUC	0.6904
Accuracy	85.12%
Sensitivity	90.13%
Specificity	27.57%

Table 10: Confusion Matrix Results: GBM

As shown in **Table 10** above, this model has the highest AUC of all the models in this analysis. Additionally, it is very accurate at 85% and is also moderately good at predicting Negative classes. **Figure 13** below shows the ROC curve for the GBM model.

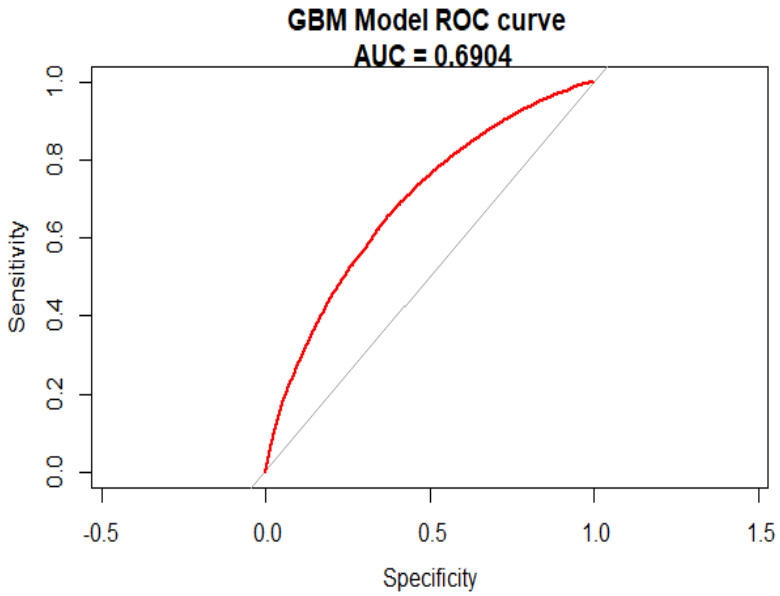

Figure 13: GBM Model Variable Relative Influence

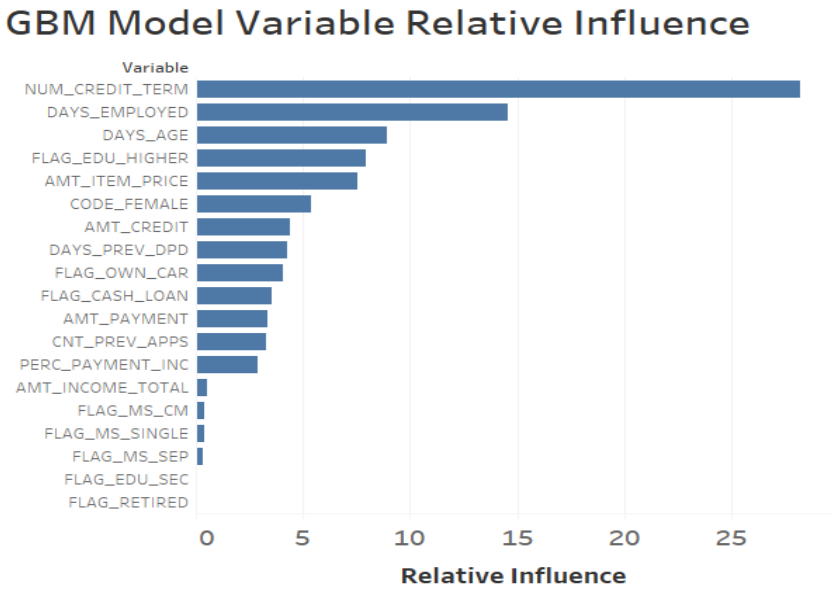

Figure 14: GBM Model ROC

Moreover, the GBM model produces a table of the relative influence of each predictor on the model. As shown in **Figure 14** below, the length of the loan is the most significant predictor, which was not the case in the Random Forest model. It does corroborate the Random Forest model in that the length of employment, age are strong predictors.

5 Conclusion and Further Analysis

After running all the models, it is clear that the GBM model is the strongest at predicting default as it had a good balance AUC and accuracy. Furthermore, the LDA and QDA models while did have a high accuracy, were not optimal for this case as they are better suited for cases when target variable contains more than 2 classes. Similarly, the logistic regression model did display a good AUC and accuracy. The kNN model had a high accuracy for higher k values. Lastly, the Random Forest model worked well at identifying the strongest predictors.

For further analysis, we hope to optimize some of the models above. First, treat for any collinearity that exists between the predictors for the logistic regression model. Second, it seems prudent to run the Random Forest model prior to running any logistic regression model as the Random Forest can help determine which features to select for building the logit model. Furthermore, we will explore different machine learning algorithms such as XGBoost, Adaboost, or Neural Network to improve the accuracy of the models built in this analysis. Lastly, we hope to overcome the computing power problem and utilize all the tables in the provided data set. Some of the tables that were not used contained important payment history information that can be essential when assessing default risk.

6 Appendix

6.1 A1: Data ERD

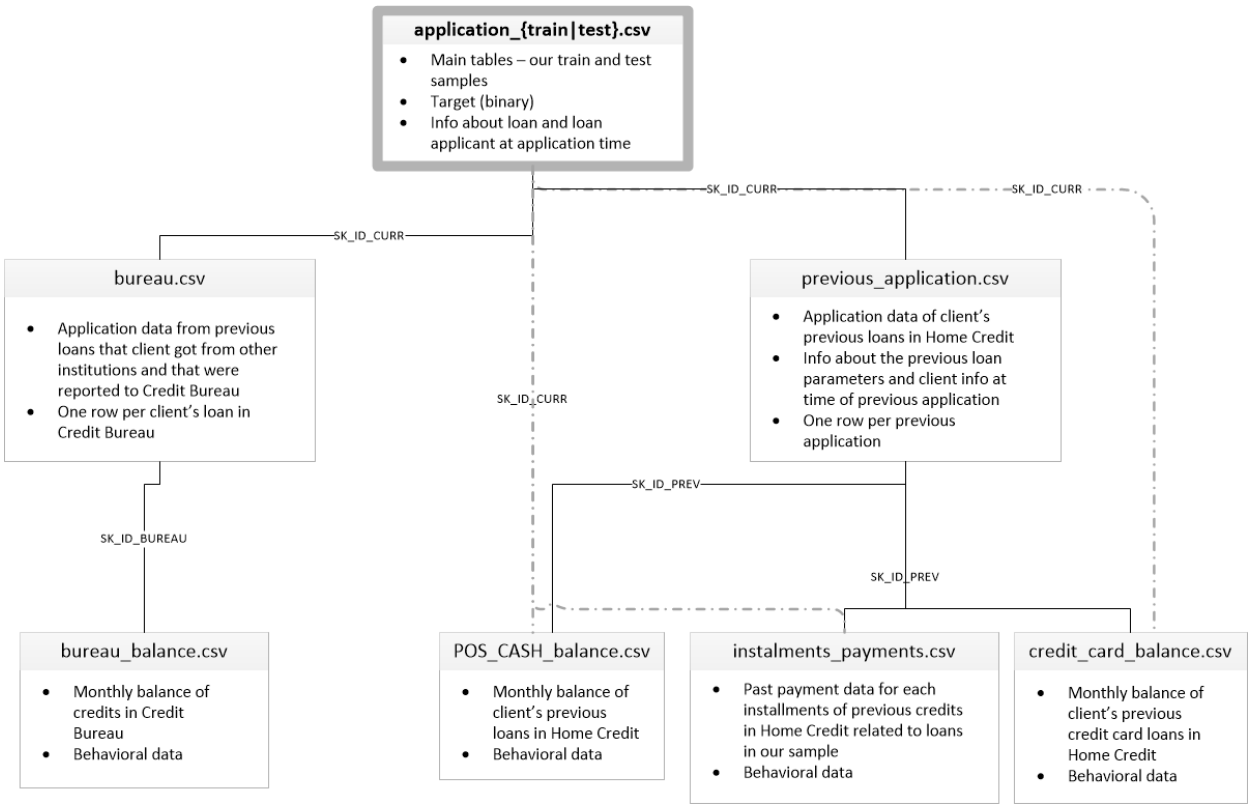

Figure 15: ERD of the of provided Kaggle data set

6.2 A2: Feature Details

Column Name	Column Description	Data Type
SK_ID_CURR	Unique client identifier	Integer
TARGET	Target Variable (1 Default, 0 no-default)	String
FLAG_CASH_LOAN	Flag indicating whether the loan is an installment cash loan (1) or a revolving credit loan (0)	Integer
CODE_FEMALE	Flag indicating the client's gender (1 Female, 0 Male)	Integer
FLAG_OWN_CAR	Flag indicating if the client owns a vehicle	Integer
FLAG_OWN_RESTATE	Flag indicating if the client owns real estate	Integer
CNT_CHILDREN	The number of client's children	Integer
AMT_INCOME_TOTAL	The client's total income in the local currency	Float
AMT_CREDIT	The amount of Credit (loan) the client is applying for	Float
AMT_PAYMENT	The monthly payment on the loan	Float
AMT_ITEM_PRICE	The price of the item the client is financing	Float
FLAG_EDU_HIGHER	Flag indicating if the client's highest education level is Higher Education	Integer
FLAG_EDU_SEC	Flag indicating if the client's highest education level is Secondary Education	Integer
FLAG_MS_MARRIED	Flag indicating if the client's martial status is Married	Integer
FLAG_MS_SEP	Flag indicating if the client's martial status is Separated	Integer
FLAG_MS_CM	Flag indicating if the client's martial status is Civil Marriage	Integer
FLAG_MS_SINGLE	Flag indicating if the client's martial status is Single	Integer
DAYS_AGE	Client age in days on the day of the application	Integer
FLAG_RETIRED	Flag indicating if the client is retired (pensioner)	Integer
DAYS_EMPLOYED	The number of days the client has been employed at their current job. 0 indicates a retired person	Integer
CNT_FAM_MEMBERS	The number of client household members	Integer
CNT_PREV_APPS	The number of previous credit applications (if any) the client has had with the financial institution	Integer
DAYS_PREV_DPD	The maximum number of days past due on a previous loan associated with the customer	Integer

Table 11: Final data set feature names, descriptions, and data types

6.3 A3: Logistic Regression with all features

```
Call:
glm(formula = TARGET ~ ., family = "binomial", data = train_data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.6762  -0.4514  -0.3692  -0.2986   3.1855

Coefficients: (1 not defined because of singularities)
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)       -1.572e+00  1.149e-01 -13.684  < 2e-16 ***
FLAG_CASH_LOAN     4.064e-01  3.912e-02  10.388  < 2e-16 ***
CODE_FEMALE       -4.218e-01  1.926e-02 -21.903  < 2e-16 ***
FLAG_OWN_CAR      -3.024e-01  2.011e-02 -15.041  < 2e-16 ***
FLAG_OWN_RESTATE   9.845e-03  1.891e-02   0.521  0.60260
CNT_CHILDREN      -2.766e-03  1.234e-02  -0.224  0.82267
AMT_INCOME_TOTAL   1.246e-07  8.554e-08   1.456  0.14535
AMT_CREDIT         2.485e-06  1.627e-07  15.275  < 2e-16 ***
AMT_PAYMENT        4.985e-06  1.854e-06   2.689  0.00717 **
AMT_ITEM_PRICE    -3.318e-06  1.576e-07 -21.058  < 2e-16 ***
FLAG_EDU_HIGHER   -6.173e-01  7.232e-02  -8.536  < 2e-16 ***
FLAG_EDU_SEC      -1.467e-01  6.926e-02  -2.118  0.03420 *
FLAG_MS_MARRIED   -1.189e-02  4.718e-02  -0.252  0.80099
FLAG_MS_SEP        1.392e-01  5.606e-02   2.483  0.01302 *
FLAG_MS_CM         1.438e-01  5.237e-02   2.746  0.00604 **
FLAG_MS_SINGLE     1.041e-01  5.130e-02   2.030  0.04240 *
DAYS_AGE          -5.231e-05  2.821e-06 -18.541  < 2e-16 ***
FLAG_RETIRED       1.405e+00  8.687e-02  16.168  < 2e-16 ***
DAYS_EMPLOYED     -9.830e-05  5.270e-06 -18.652  < 2e-16 ***
CNT_FAM_MEMBERS           NA         NA      NA       NA
CNT_PREV_APPS      2.702e-02  1.900e-03  14.220  < 2e-16 ***
DAYS_PREV_DPD      2.412e-02  2.484e-03   9.709  < 2e-16 ***
PERC_PAYMENT_INC   1.071e+00  1.260e-01   8.500  < 2e-16 ***
NUM_CREDIT_TERM    6.180e-03  2.781e-03   2.222  0.02629 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 105926  on 189201  degrees of freedom
Residual deviance: 101659  on 189179  degrees of freedom
AIC: 101705

Number of Fisher Scoring iterations: 6
```

Figure 16: Full Logistic Regression Model Results

6.4 A4: Reduced Logistic Regression with all features

```
Call:
glm(formula = TARGET ~ FLAG_CASH_LOAN + CODE_FEMALE + FLAG_OWN_CAR +
    AMT_INCOME_TOTAL + AMT_CREDIT + AMT_PAYMENT + AMT_ITEM_PRICE +
    FLAG_EDU_HIGHER + FLAG_EDU_SEC + FLAG_MS_SEP + FLAG_MS_CM +
    FLAG_MS_SINGLE + DAYS_AGE + FLAG_RETIRED + DAYS_EMPLOYED +
    CNT_PREV_APPS + DAYS_PREV_DPD + PERC_PAYMENT_INC + NUM_CREDIT_TERM,
    family = "binomial", data = train_data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.6763  -0.4514  -0.3693  -0.2986   3.1837

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)      -1.582e+00  1.020e-01 -15.514  < 2e-16 ***
FLAG_CASH_LOAN    4.052e-01  3.901e-02  10.386  < 2e-16 ***
CODE_FEMALE      -4.211e-01  1.914e-02 -22.008  < 2e-16 ***
FLAG_OWN_CAR     -3.026e-01  2.007e-02 -15.080  < 2e-16 ***
AMT_INCOME_TOTAL  1.239e-07  8.565e-08   1.447  0.14793
AMT_CREDIT        2.491e-06  1.621e-07  15.369  < 2e-16 ***
AMT_PAYMENT       4.986e-06  1.853e-06   2.691  0.00713 **
AMT_ITEM_PRICE   -3.323e-06  1.570e-07 -21.164  < 2e-16 ***
FLAG_EDU_HIGHER  -6.168e-01  7.230e-02  -8.531  < 2e-16 ***
FLAG_EDU_SEC     -1.461e-01  6.924e-02  -2.110  0.03490 *
FLAG_MS_SEP       1.501e-01  3.545e-02   4.233 2.31e-05 ***
FLAG_MS_CM        1.555e-01  2.768e-02   5.618 1.94e-08 ***
FLAG_MS_SINGLE    1.165e-01  2.425e-02   4.803 1.56e-06 ***
DAYS_AGE         -5.193e-05  2.712e-06 -19.147  < 2e-16 ***
FLAG_RETIRED      1.407e+00  8.667e-02  16.232  < 2e-16 ***
DAYS_EMPLOYED    -9.837e-05  5.268e-06 -18.674  < 2e-16 ***
CNT_PREV_APPS     2.706e-02  1.898e-03  14.255  < 2e-16 ***
DAYS_PREV_DPD     2.411e-02  2.484e-03   9.705  < 2e-16 ***
PERC_PAYMENT_INC  1.067e+00  1.257e-01   8.485  < 2e-16 ***
NUM_CREDIT_TERM   6.104e-03  2.776e-03   2.199  0.02787 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 105926  on 189201  degrees of freedom
Residual deviance: 101659  on 189182  degrees of freedom
AIC: 101699

Number of Fisher Scoring iterations: 6
```

Figure 17: Reduced Logistic Regression Model Results

6.5 A6: GBM Model in R

```
gbm.final <- gbm (TARGET ~ FLAG_CASH_LOAN + CODE_FEMALE + FLAG_OWN_CAR +
                    AMT_INCOME_TOTAL + AMT_CREDIT + AMT_PAYMENT + AMT_ITEM_PRICE +
                    FLAG_EDU_HIGHER + FLAG_EDU_SEC + FLAG_MS_SEP + FLAG_MS_CM +
                    FLAG_MS_SINGLE + DAYS_AGE + FLAG_RETIRED + DAYS_EMPLOYED +
                    CNT_PREV_APPS + DAYS_PREV_DPD + PERC_PAYMENT_INC + NUM_CREDIT_TERM,
                  distribution = "gaussian",
                  data=train_data,
                  n.trees = 1000,
                  interaction.depth = 4,
                  shrinkage = 0.01,
                  cv.folds = 3)
```

Figure 18: GBM Model in R