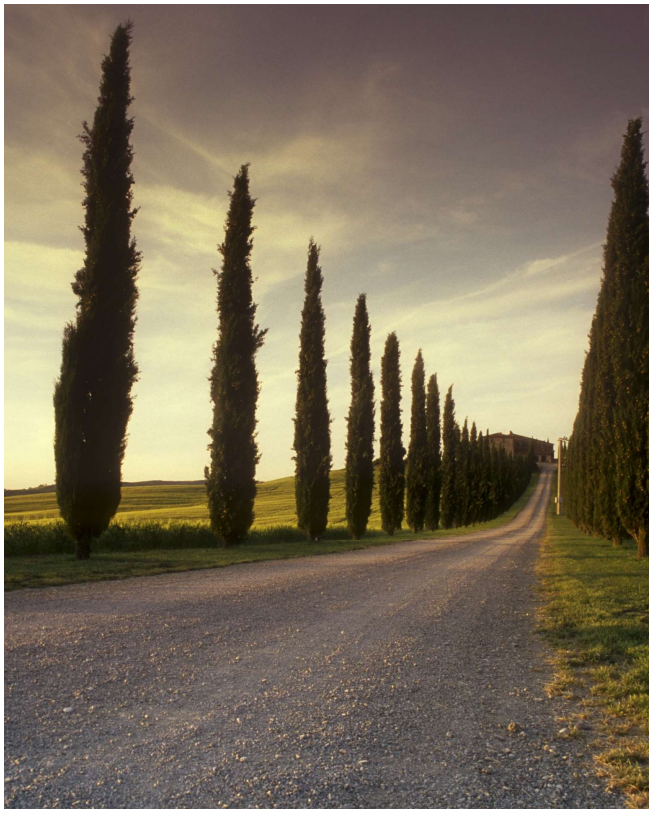

PREDICTING BANK LOAN DEFAULT

Instructor: Prof. Keying Ye

Group project report for partial fulfillment of the course.

Ambassador Negash, Cambrey Sullivan, Kevin Sablan and Roxanne Salinas

Advanced Statistical Learning and Data Mining-Fall,2019

Table of Contents

List of Tables

List of Figures

1. Introduction

The dataset used has 30,000 observations and looks at credit default from Customersâ€™ in Taiwan. The source of the dataset is a website for UCI Machine Learning Repository[1]. The response variable is a binary variable indicating whether a given customer defaulted on his/her loan. Besides the response variable, the dataset has 23 explanatory variables.

The explanatory variables are:

- X_1: Amount of the given credit (NT dollar): it includes both the individual consumer credit and his/her family (supplementary) credit.
- X_2: Gender (1 = male; 2 = female).
- X_3: Education (1 = graduate school; 2 = university; 3 = high school; 4 = others).
- X_4: Marital status (1 = married; 2 = single; 3 = others).
- X_5: Age (year).
- $X_6 - X_{11}$: History of past payment. These variables track the past monthly payment records (from April to September 2005) as follows:
  - X_6: the repayment status in September 2005;
  - X_7: the repayment status in August 2005; . . .;
  - X_{11}: the repayment status in April 2005. The measurement scale for the repayment status is: -1 = pay duly; 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.
- $X_{12} - X_{17}$: Amount of bill statement (NT dollar).
  - X_{12}: amount of bill statement in September, 2005;
  - X_{13}: amount of bill statement in August, 2005; . . .;
  - X_{17}: amount of bill statement in April, 2005.
- $X_{18} - X_{23}$: Amount of previous payment (NT dollar).
  - X_{18}: amount paid in September, 2005;
  - X_{19}: amount paid in August, 2005; . . .;
  - X_{23}: amount paid in April, 2005.

The response variable is:

- y: default payment (Yes = 1, No = 0)

The main objective of the project is to produce an adequate prediction model that helps predict credit defaults using explanatory variables identified in the dataset. To achieve this, the project will compare various classification methods to identify and forecast loan defaulters. The methods we used are logistic regression, LDA, QDA, Naïve Bayes, and KNN for the original data as well as Regularization methods, such as LASSO, Ridge and Elastic Net. In addition, logistic regression, LDA, QDA, Naïve Bayes, and KNN were performed on the principal components of the explanatory variables.

These models were trained on a training dataset, which is 70% of the entire dataset, and tested on the remaining holdout dataset for accuracy and sensitivity. Finally, a champion model, with the smallest test error, is selected as the model of choice.

[1] https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients

2. Exploratory Data analysis

2.1 The Response Variable

Before we conduct any modeling, we explored the data using plots, correlation matrices and summary statistics. Figure 1 shows the relative proportion of defaulters vis-à-vis non-defaulters in the dataset.

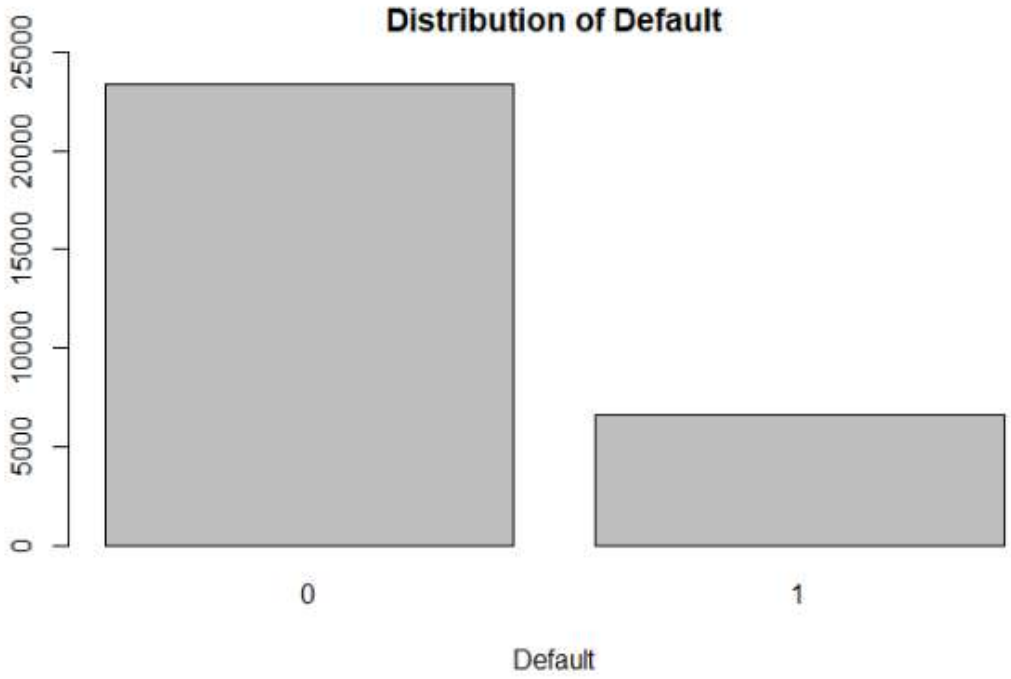

Figure 1: Distribution of Response Variable (Default)

As we see above, defaulters constitute 6,636 individuals of the total 30,000 individuals included in the dataset. This accounts for 22.12% of all customers. The nonproportionality of classes means that assessing the performance of models solely based on test accuracy could be misleading. For example, a model that predicts all test observations as zero has a 78% accuracy rate. However, this does not mean that model is performing well. With this as a backdrop, we tried to assess each individual model based on its performance on a list of statistics besides accuracy.

2.2 Explanatory Variables

One of the common problems associated with explanatory variables is a problem of multicollinearity. We found that the bill amount variables ($X_{12} - X_{17}$) were highly correlated with one another. The following correlation matrix on those variables indicate, their mutual correlation coefficients are greater than 0.9 in absolute value.

Table 1: Correlation matrix for highly correlated variables

	row	col	corr
BILL_AMT2	13	12	0.951
BILL_AMT1	12	13	0.951
BILL_AMT3	14	13	0.928
BILL_AMT2.1	13	14	0.928
BILL_AMT4	15	14	0.924
BILL_AMT3.1	14	15	0.924
BILL_AMT5	16	15	0.940
BILL_AMT6	17	15	0.901
BILL_AMT4.1	15	16	0.940
BILL_AMT6.1	17	16	0.946
BILL_AMT4.2	15	17	0.901
BILL_AMT5.1	16	17	0.946

Graphically, we can also observe that there are highly linear relationships between these variables:

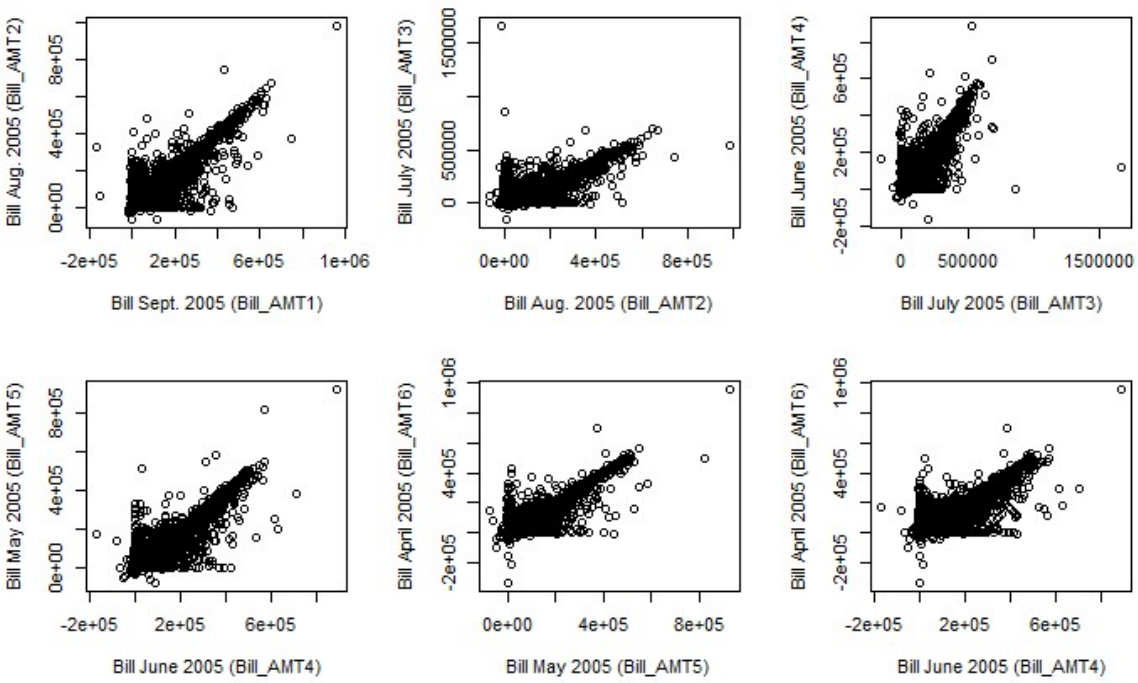

Figure 2: Scatter plot for highly correlated variables

Multicollinearity is not a detrimental problem for forecasting, however, in estimation problems, it could interfere in the decision to determine the effect of a variable on a response. Since the variance of the estimated coefficients for correlated variables inflate, the significance of these variables on the response can be hard to assess. Moreover, strong correlation means fixing all variables to explain the effect of one turns out to be a philosophical conundrum.

3. Classification Models

3.1 Logistic Regression without Regularization

One of the most popular classification methods is logistic regression. Given the explanatory variables, logistic regression assumes that each response variable follows a Bernoulli distribution with probability of $p(y_i = 1|\boldsymbol{X})$.

$$y_i|\boldsymbol{X} \sim Bernoulli(p(y_i = 1|\boldsymbol{X}))$$

Since the link function for $p(y_i = 1|\boldsymbol{X})$ is a logistic distribution, finding the coefficients of the variables involve maximizing the following distribution with respect to the coefficients:

$$L(y_i|\boldsymbol{X}) \propto \prod_{i=1}^{n} \left(\frac{1}{1+e^{-XB}}\right)^{\sum_{i=1}^{n} y_i} \left(\frac{e^{-XB}}{1+e^{-XB}}\right)^{n-\sum_{i=1}^{n} y_i}$$

After fitting logistic regression on the training dataset, we assessed model performance on the test dataset using a confusion matrix.

Table 2: Confusion Matrix for logistic regression without regularization

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 6828  215
##          1 1461  496
##
##                Accuracy : 0.8138
##                  95% CI : (0.8056, 0.8218)
##     No Information Rate : 0.921
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.2895
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.8237
##             Specificity : 0.6976
##          Pos Pred Value : 0.9695
##          Neg Pred Value : 0.2534
##              Prevalence : 0.9210
##          Detection Rate : 0.7587
##    Detection Prevalence : 0.7826
##       Balanced Accuracy : 0.7607
##
##        'Positive' Class : 0
##
```

As seen in table 2, nearly a third of defaulters, 216 of the total 711, are misclassified as nondefaulters. As such, this method of classification on the given dataset may not be the best method of classification. Also, the fitted logistic regression on the training dataset has numerous coefficients which are not statistically significant (p-value >0.05) as seen in table 3. These exceptions are likely due to the multicollinearity present in the billing amount variables.

Table 3: Coefficient estimates for logistic regression without regularization

```
## Call:
## glm(formula = default ~ ., family = binomial, data = d.train.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1800  -0.7051  -0.5403  -0.2625   4.0468
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.385e+01  8.592e+01  -0.161 0.871890
## LIMIT_BAL   -8.198e-07  1.910e-07  -4.293 1.76e-05 ***
## SEX2        -1.321e-01  3.678e-02  -3.593 0.000327 ***
## EDUCATION1   1.087e+01  8.591e+01   0.127 0.899272
## EDUCATION2   1.079e+01  8.591e+01   0.126 0.900042
## EDUCATION3   1.072e+01  8.591e+01   0.125 0.900710
## EDUCATION4   9.778e+00  8.591e+01   0.114 0.909386
## EDUCATION5   9.247e+00  8.591e+01   0.108 0.914286
## EDUCATION6   1.058e+01  8.591e+01   0.123 0.902030
## MARRIAGE1    1.993e+00  8.259e-01   2.413 0.015816 *
## MARRIAGE2    1.815e+00  8.260e-01   2.198 0.027971 *
## MARRIAGE3    2.025e+00  8.407e-01   2.408 0.016020 *
## AGE          6.414e-03  2.237e-03   2.867 0.004144 **
## PAY_0        5.884e-01  2.123e-02  27.714  < 2e-16 ***
## PAY_2        8.716e-02  2.406e-02   3.623 0.000291 ***
## PAY_3        5.790e-02  2.713e-02   2.134 0.032815 *
## PAY_4        4.577e-02  3.008e-02   1.522 0.128097
## PAY_5        7.044e-03  3.243e-02   0.217 0.828019
## PAY_6        1.550e-02  2.669e-02   0.581 0.561286
## BILL_AMT1   -6.198e-06  1.354e-06  -4.578 4.70e-06 ***
## BILL_AMT2    2.555e-06  1.782e-06   1.434 0.151646
## BILL_AMT3    2.423e-06  1.586e-06   1.528 0.126537
## BILL_AMT4    2.606e-07  1.611e-06   0.162 0.871524
## BILL_AMT5    7.108e-07  1.879e-06   0.378 0.705196
## BILL_AMT6   -3.339e-07  1.472e-06  -0.227 0.820499
## PAY_AMT1    -1.479e-05  2.870e-06  -5.154 2.54e-07 ***
## PAY_AMT2    -1.139e-05  2.625e-06  -4.338 1.44e-05 ***
## PAY_AMT3    -8.838e-07  1.971e-06  -0.449 0.653790
## PAY_AMT4    -5.500e-06  2.314e-06  -2.376 0.017480 *
## PAY_AMT5    -4.315e-06  2.224e-06  -1.940 0.052334 .
## PAY_AMT6    -3.929e-06  1.669e-06  -2.354 0.018595 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 22279  on 20999  degrees of freedom
## Residual deviance: 19419  on 20969  degrees of freedom
## AIC: 19481
##
## Number of Fisher Scoring iterations: 11
```

The ROC plot for the fitted logistic regression is shown below in figure 3. The area under the curve is 0.7127.

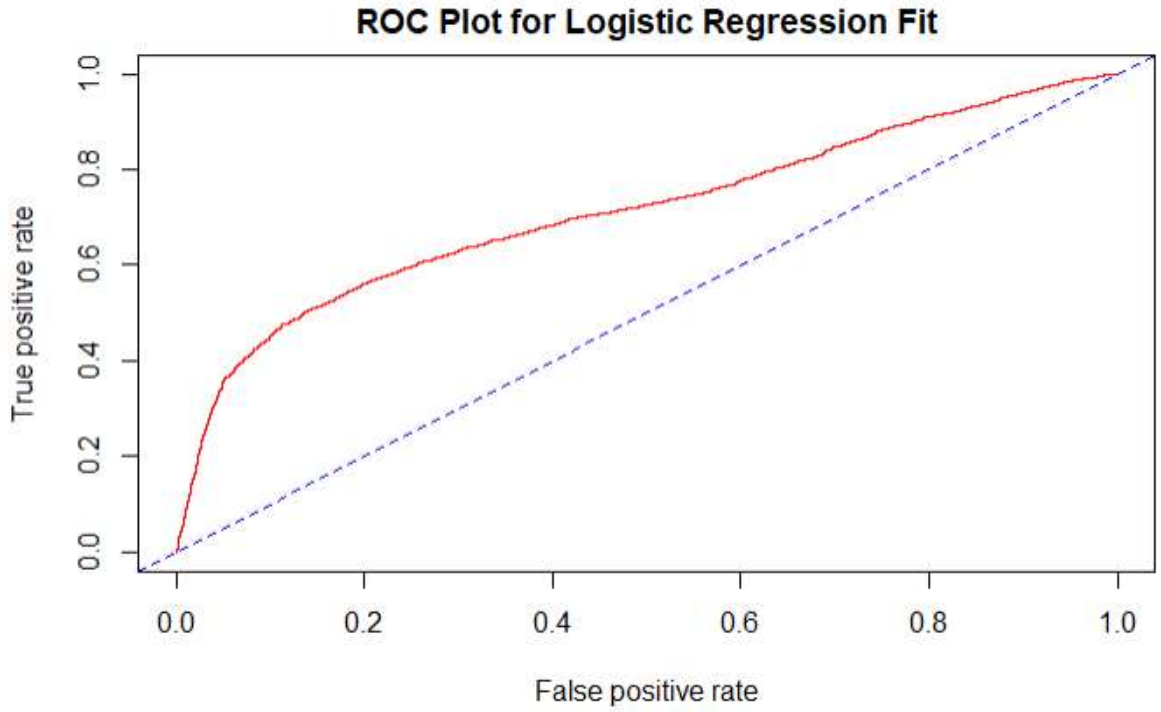

Figure 3: ROC plot for logistic regression without regularization

3.2 Logistic Regression with Regularization

After assessing that fitting logistic regression on dataset without regularization is not a good fit, we turned to three regularization methods: Ridge, LASSO, and Elastic Net.

3.2.1 Ridge Logistic Regression

For this method, we chose the tuning parameters (λ) using cross validation method on the entire dataset. Two tuning paraments are selected to fit the Ridge Logistic regression (minimum lambda and 1 SE lambda). The following plot shows the one-standard error lambdas and minimum lambda.

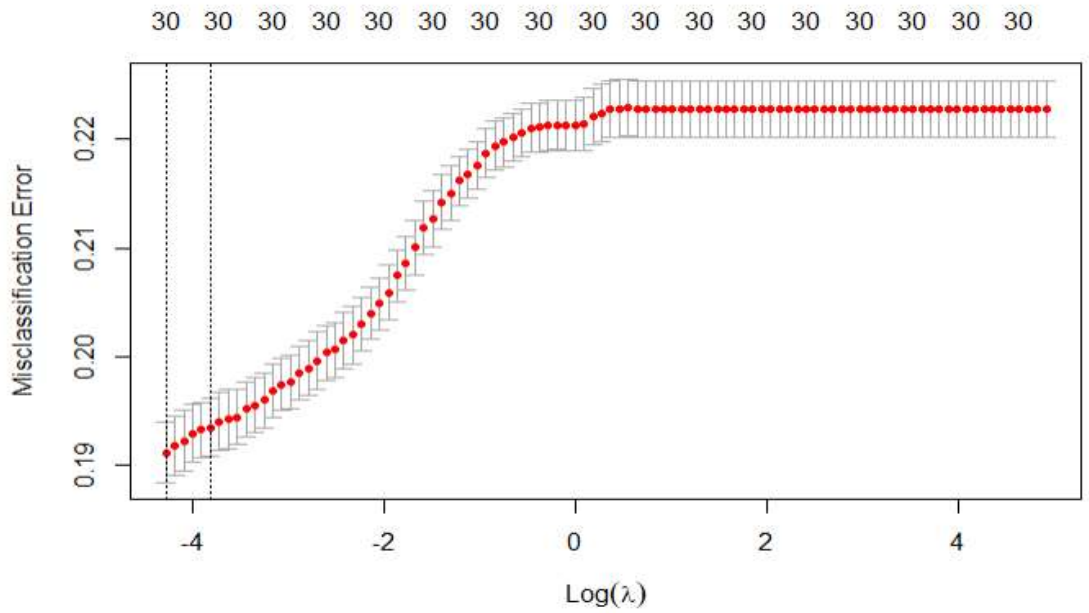

Figure 4: Misclassification error plot for various lambda-Ridge

3.2.1.1 Ridge Regression with Minimum Lambda

The ridge logistic regression coefficient estimates with minimum lambda are:

Table 4: Coefficient estimates for ridge logistic regression with minimum lambda

```
## 31 x 1 sparse Matrix of class "dgCMatrix"
##                          s0
## (Intercept) -1.210840e+00
## LIMIT_BAL   -8.390587e-07
## SEX2        -1.193283e-01
## EDUCATION1   9.436524e-02
## EDUCATION2   2.375011e-02
## EDUCATION3  -4.398191e-02
## EDUCATION4  -8.182685e-01
## EDUCATION5  -1.200684e+00
## EDUCATION6  -1.697019e-01
## MARRIAGE1    1.199454e-01
## MARRIAGE2   -5.004661e-02
## MARRIAGE3    1.453733e-01
## AGE          5.908732e-03
## PAY_0        5.188216e-01
## PAY_2        1.029111e-01
## PAY_3        6.451965e-02
## PAY_4        4.489231e-02
## PAY_5        1.988744e-02
## PAY_6        2.132288e-02
## BILL_AMT1   -1.880769e-06
## BILL_AMT2   -2.367929e-07
## BILL_AMT3    3.301174e-07
## BILL_AMT4    3.817079e-07
## BILL_AMT5    3.528550e-07
## BILL_AMT6    1.834191e-07
## PAY_AMT1    -9.492205e-06
## PAY_AMT2    -7.557825e-06
## PAY_AMT3    -2.104387e-06
```

```
## PAY_AMT4    -5.331571e-06
## PAY_AMT5    -4.522466e-06
## PAY_AMT6    -3.692031e-06
```

As we can see, some of the estimated coefficients are very close to zero, yet none of them are zeros. The test confusion matrix based on this regression, shown in table 5, indicates that 1,823 credit card holders who defaulted were misclassified as nondefaulters.

Table 5: Confusion matrix for ridge logistic regression with minimum lambda

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 6982 1823
##          1   61  134
##
##                Accuracy : 0.7907
##                  95% CI : (0.7821, 0.799)
##     No Information Rate : 0.7826
##     P-Value [Acc > NIR] : 0.03154
##
##                   Kappa : 0.0886
##
##  Mcnemar's Test P-Value : < 2e-16
##
##             Sensitivity : 0.99134
##             Specificity : 0.06847
##          Pos Pred Value : 0.79296
##          Neg Pred Value : 0.68718
##              Prevalence : 0.78256
##          Detection Rate : 0.77578
##    Detection Prevalence : 0.97833
##       Balanced Accuracy : 0.52991
##
##        'Positive' Class : 0
```

The ROC curve for this model is shown below and its AUC is 0.7125.

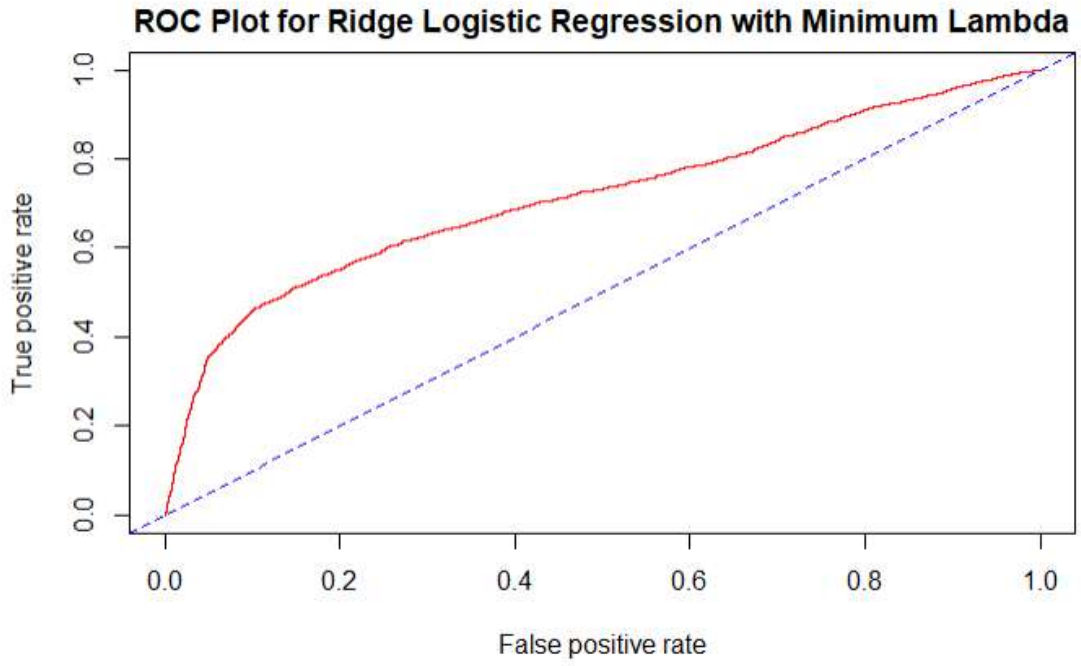

Figure 5: ROC plot for ridge logistic regression minimum lambda

As we see above, this model is worse than the non-regularized logistic regression model. Now, we fit a ridge logistic regression with 1se lambda.

3.2.1.2 Ridge Regression with 1 SE lambda

The ridge regression coefficient estimates with 1 se lambda are:

Table 6: Coefficient estimates for ridge logistic regression with 1se lambda

```
## 31 x 1 sparse Matrix of class "dgCMatrix"
##                          s0
## (Intercept) -1.169516e+00
## LIMIT_BAL   -8.144330e-07
## SEX2        -1.099593e-01
## EDUCATION1   7.368113e-02
## EDUCATION2   1.728290e-02
## EDUCATION3  -3.951150e-02
## EDUCATION4  -7.068906e-01
## EDUCATION5  -1.012134e+00
## EDUCATION6  -1.603468e-01
## MARRIAGE1    9.669302e-02
## MARRIAGE2   -6.463325e-02
## MARRIAGE3    1.197590e-01
## AGE          5.382720e-03
## PAY_0        4.610568e-01
## PAY_2        1.131001e-01
## PAY_3        6.717697e-02
## PAY_4        4.649886e-02
## PAY_5        2.772981e-02
## PAY_6        2.647251e-02
## BILL_AMT1   -1.238980e-06
## BILL_AMT2   -3.329483e-07
## BILL_AMT3    1.000959e-08
## BILL_AMT4    1.804207e-07
```

```
## BILL_AMT5    2.003845e-07
## BILL_AMT6    1.346421e-07
## PAY_AMT1    -7.955619e-06
## PAY_AMT2    -6.034975e-06
## PAY_AMT3    -2.158064e-06
## PAY_AMT4    -4.786576e-06
## PAY_AMT5    -4.165057e-06
## PAY_AMT6    -3.424683e-06
```

As we can see, some of the estimated coefficients are very close to zero, yet none of them are zeros. The test confusion matrix based on this regression is shown in table 7. This model also has a high misclassification rate for defaulters.

Table 7: Confusion matrix for ridge logistic regression with 1se lambda

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 6995 1852
##          1   48  105
##
##                Accuracy : 0.7889
##                  95% CI : (0.7803, 0.7973)
##     No Information Rate : 0.7826
##     P-Value [Acc > NIR] : 0.07403
##
##                   Kappa : 0.0702
##
##  Mcnemar's Test P-Value : < 2e-16
##
##             Sensitivity : 0.99318
##             Specificity : 0.05365
##          Pos Pred Value : 0.79066
##          Neg Pred Value : 0.68627
##              Prevalence : 0.78256
##          Detection Rate : 0.77722
##    Detection Prevalence : 0.98300
##       Balanced Accuracy : 0.52342
##
##        'Positive' Class : 0
##
```

The ROC curve for ridge regression with 1se lambda is shown below.

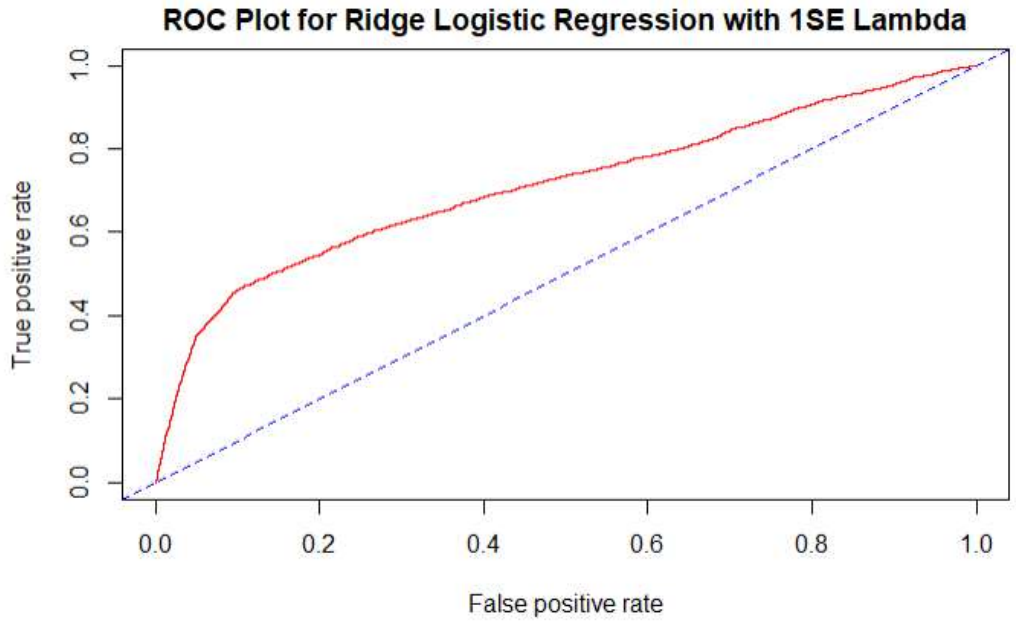

Figure 6: Roc curve for ridge logistic regression with 1se lambda

As we saw above, ridge logistic regressions (based on minimum and 1se lambdas) do not perform better than logistic regression without regularization.

3.2.2 LASSO Logistic Regression

Using a method similar to the one used for ridge regression, we obtained the tuning parameter using cross validation and then fit two types of LASSO Logistic Regression. The following plot shows the cross validated tuning parameters for LASSO logistic regression.

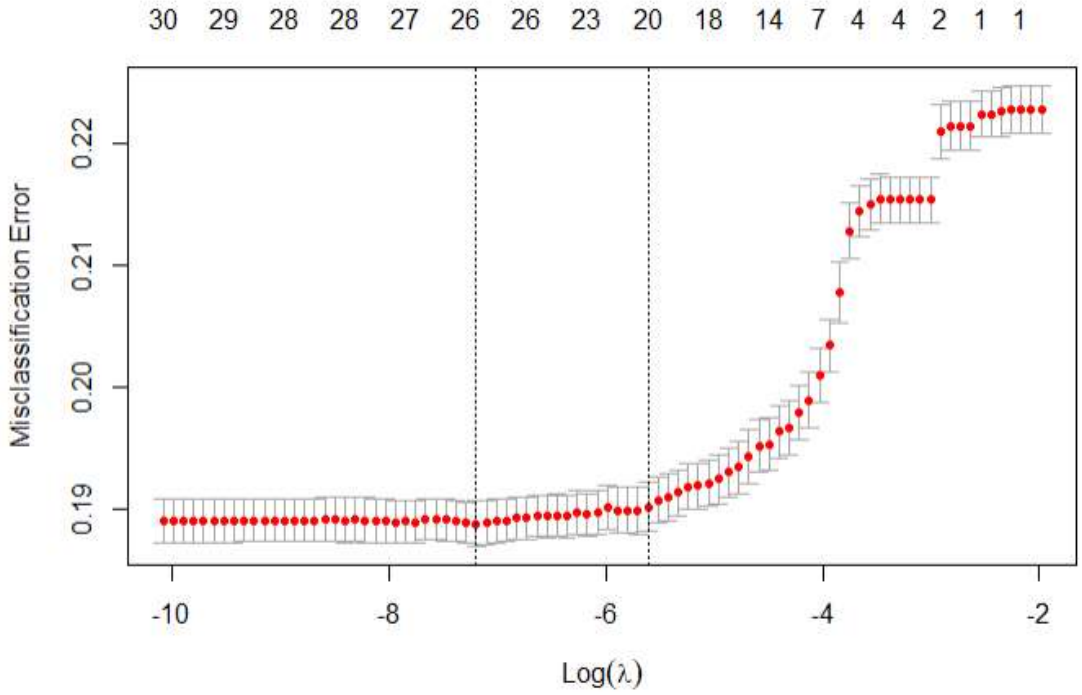

Figure 7: Misclassification error plot for various lambda-Lasso

3.2.2.1 LASSO Logistic Regression with Minimum Lambda

The following table shows the LASSO logistic regression coefficient estimates with minimum lambda. As we see below, four coefficient estimates are zero. Note that two of these are the variables which were deemed correlated with each other.

Besides those variables, numerous other variables have coefficients very close to zero. Limit Balance, Payment Amount 2 to 6 and bill amount 1 and 3 to 5 have coefficient estimates very close to zero. As such, the LASSO logistic regression with minimum lambda as a tuning parameter effectively suggests a model with a smaller number of variables.

Table 8: Coefficient estimates for Lasso logistic regression with minimum lambda

```
## 31 x 1 sparse Matrix of class "dgCMatrix"
##                          s0
## (Intercept) -2.326335e+00
## LIMIT_BAL   -8.246865e-07
## SEX2        -1.313988e-01
## EDUCATION1   3.494931e-01
## EDUCATION2   2.659141e-01
## EDUCATION3   1.903564e-01
## EDUCATION4  -7.321057e-01
## EDUCATION5  -1.265316e+00
## EDUCATION6   4.356424e-02
## MARRIAGE1    9.894517e-01
## MARRIAGE2    8.112206e-01
## MARRIAGE3    1.019470e+00
## AGE          6.378224e-03
## PAY_0        5.879765e-01
## PAY_2        8.667621e-02
## PAY_3        5.833678e-02
## PAY_4        4.542738e-02
## PAY_5        7.630786e-03
## PAY_6        1.576287e-02
## BILL_AMT1   -5.822968e-06
## BILL_AMT2    2.253020e-06
## BILL_AMT3    2.340979e-06
## BILL_AMT4    2.537612e-07
## BILL_AMT5    4.588014e-07
## BILL_AMT6   -6.345666e-08
## PAY_AMT1    -1.446401e-05
## PAY_AMT2    -1.128469e-05
## PAY_AMT3    -8.842692e-07
## PAY_AMT4    -5.440183e-06
## PAY_AMT5    -4.486477e-06
## PAY_AMT6    -3.857628e-06
```

The confusion matrix from this model is shown in table 9. Like with the Ridge regression models, LASSO with minimum lambda also has a high misclassification rate on defaulters.

Table 9: Confusion matrix for lasso logistic regression with minimum lambda

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 6971 1788
##          1   72  169
```

```
##
##                Accuracy : 0.7933
##                  95% CI : (0.7848, 0.8017)
##     No Information Rate : 0.7826
##     P-Value [Acc > NIR] : 0.006598
##
##                   Kappa : 0.1114
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.98978
##             Specificity : 0.08636
##          Pos Pred Value : 0.79587
##          Neg Pred Value : 0.70124
##              Prevalence : 0.78256
##          Detection Rate : 0.77456
##    Detection Prevalence : 0.97322
##       Balanced Accuracy : 0.53807
##
##        'Positive' Class : 0
##
```

The ROC curve from this model is:

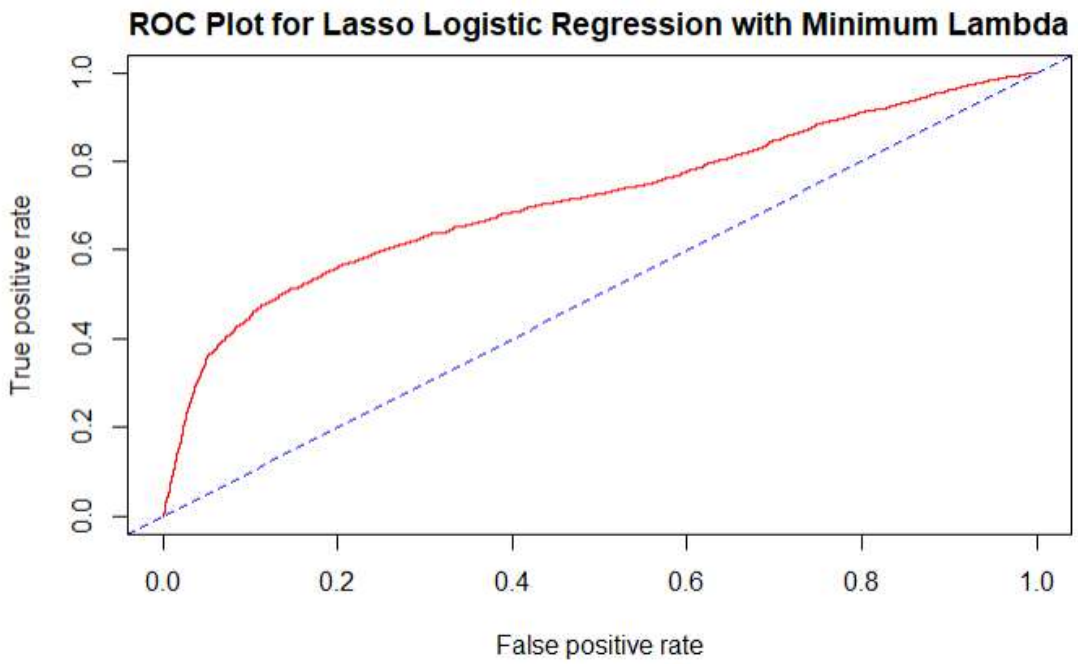

Figure 8: ROC curve for lasso logistic regression with minimum lambda

Again, this model is not as good as the model fitted without regularization. Next, we fitted a LASSO logistic regression with 1 SE lambda.

3.2.2.2 LASSO Logistic Regression with 1 SE Lambda

Table 10 shows the LASSO Logistic Regression Coefficient estimates. As before, 6 of the variables have zero coefficient estimates. This method immediately removes 6 variables from the model. Five of the coefficients for Bill Amount (2-6) and one coefficient for Payment Amount (3) are estimated to be zero.

Besides those variables, numerous other variables have coefficients very close to zero. Limit Balance, Payment Amount 1, 2, 4,5 and 6 have coefficient estimates very close to zero. As such, the LASSO logistic regression with 1se lambda as a tuning parameter effectively suggests a model with 10 variables, instead of the original 23.

Table 10: Coefficient estimates for lasso logistic regression with 1se lambda

```
## 31 x 1 sparse Matrix of class "dgCMatrix"
##                         s0
## (Intercept) -1.197324e+00
## LIMIT_BAL   -7.780231e-07
## SEX2        -5.210082e-02
## EDUCATION1   .
## EDUCATION2   .
## EDUCATION3   .
## EDUCATION4   .
## EDUCATION5  -6.300498e-01
## EDUCATION6   .
## MARRIAGE1    7.130293e-02
## MARRIAGE2   -1.948011e-02
## MARRIAGE3    .
## AGE          2.776764e-03
## PAY_0        5.754877e-01
## PAY_2        7.923960e-02
## PAY_3        5.941603e-02
## PAY_4        3.763596e-02
## PAY_5        8.553320e-03
## PAY_6        2.999931e-03
## BILL_AMT1   -1.126776e-06
## BILL_AMT2    .
## BILL_AMT3    .
## BILL_AMT4    .
## BILL_AMT5    .
## BILL_AMT6    .
## PAY_AMT1    -5.801902e-06
## PAY_AMT2    -3.342218e-06
## PAY_AMT3     .
## PAY_AMT4    -1.883623e-06
## PAY_AMT5    -1.767717e-06
## PAY_AMT6    -1.252120e-06
```

The test confusion matrix from this model is as follows:

Table 11: Confusion matrix for lasso logistic regression with 1se lambda

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 6994 1853
##          1   49  104
##
##                Accuracy : 0.7887
##                  95% CI : (0.7801, 0.7971)
##     No Information Rate : 0.7826
##     P-Value [Acc > NIR] : 0.08151
##
##                   Kappa : 0.0692
```

```
##
##  Mcnemar's Test P-Value : < 2e-16
##
##             Sensitivity : 0.99304
##             Specificity : 0.05314
##          Pos Pred Value : 0.79055
##          Neg Pred Value : 0.67974
##              Prevalence : 0.78256
##          Detection Rate : 0.77711
##    Detection Prevalence : 0.98300
##       Balanced Accuracy : 0.52309
##
##        'Positive' Class : 0
##
```

The ROC curve for this model is:

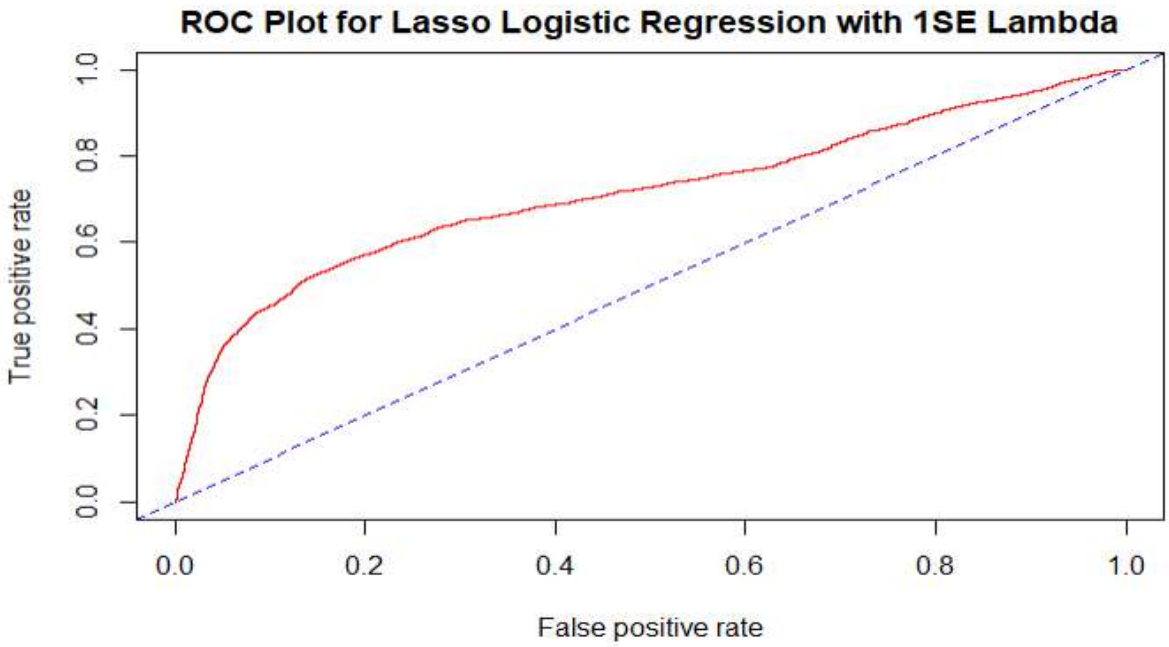

Figure 9: ROC curve for lasso logistic regression with 1se lambda

Again, LASSO logistic regression model with 1SE lambda as a tuning parameter does not perform better than the logistic regression without regularization.

3.2.3 Elastic Net Logistic Regression

For this model, as a matter of convenience, we chose alpha to be 0.5. Similar to what we did with Ridge and Lasso logistic regression, we chose the tuning parameter using cross validation on the entire dataset. The following plot shows numerous estimates of the tuning parameter estimated through cross validation.

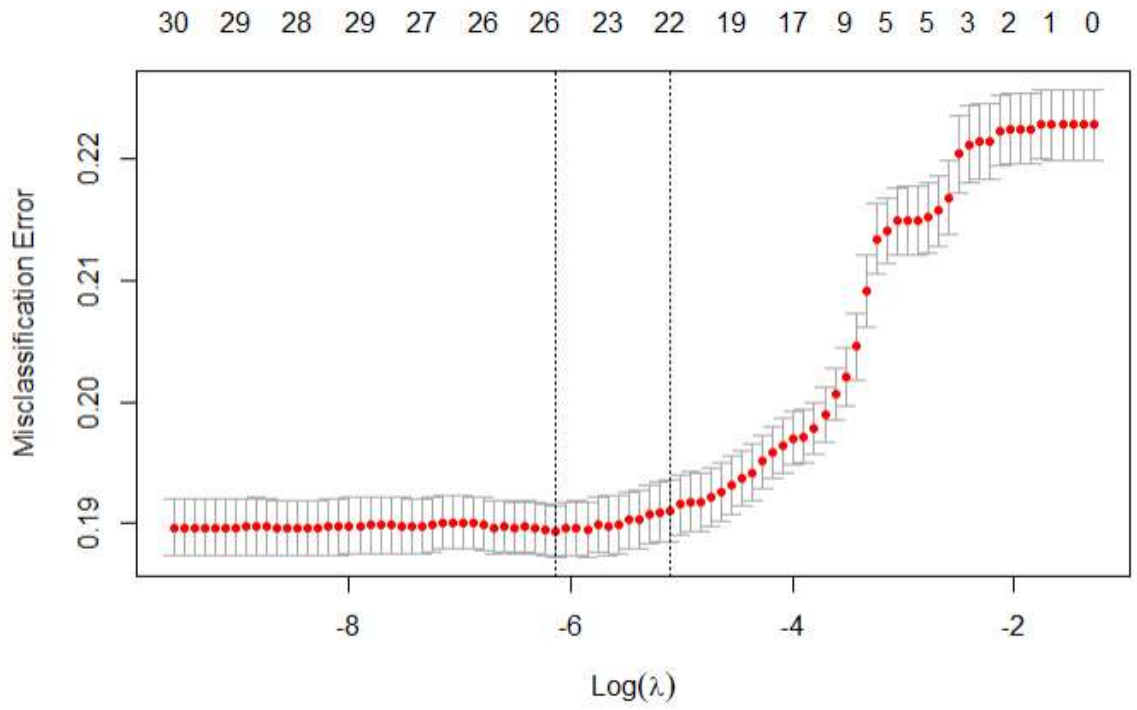

Figure 10: Misclassification error plot for various lambda-Elastic net

We are also cognizant of the fact that elastic net logistic regression gives value somewhere in between ridge and Lasso. Since the ridge and lasso fits do not perform better than the none-regularized logistic regression, we do not expect this model to perform better than the none-regularized logistic regression. For the sake of exercise and to show the robustness of our research, we fitted this model to prove what we have already expected.

3.2.3.1 Elastic Net Logistic Regression with Minimum Lambda

The following table shows the coefficient estimates for elastic net logistic regression with minimum lambda.

Table 12: Coefficient estimate for elastic net logistic regression with minimum lambda

```
## 31 x 1 sparse Matrix of class "dgCMatrix"
##                        s0
## (Intercept) -1.246880e+00
## LIMIT_BAL   -7.918567e-07
## SEX2        -1.102462e-01
## EDUCATION1   5.935857e-02
## EDUCATION2   .
## EDUCATION3  -4.536444e-02
## EDUCATION4  -6.699104e-01
## EDUCATION5  -1.231449e+00
## EDUCATION6   .
## MARRIAGE1    1.570050e-01
## MARRIAGE2    .
## MARRIAGE3    1.188401e-01
## AGE          5.478777e-03
## PAY_0        5.798298e-01
## PAY_2        8.480178e-02
## PAY_3        6.282912e-02
## PAY_4        4.612999e-02
## PAY_5        1.114418e-02
## PAY_6        1.744204e-02
```

```
## BILL_AMT1   -1.558654e-06
## BILL_AMT2    .
## BILL_AMT3    .
## BILL_AMT4    1.919094e-07
## BILL_AMT5    1.447701e-07
## BILL_AMT6    .
## PAY_AMT1    -9.598983e-06
## PAY_AMT2    -7.351511e-06
## PAY_AMT3    -8.643330e-07
## PAY_AMT4    -4.580962e-06
## PAY_AMT5    -3.888413e-06
## PAY_AMT6    -3.227459e-06
```

Note that four variables have coefficient estimates equal to zero. All other coefficients that are estimated to be zero with lasso have estimates very close to zero in this model.

The following table shows the test confusion matrix.

Table 13: Confusion matrix for elastic net logistic regression with minimum lambda

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 6976 1808
##          1   67  149
##
##                Accuracy : 0.7917
##                  95% CI : (0.7831, 0.8)
##     No Information Rate : 0.7826
##     P-Value [Acc > NIR] : 0.01827
##
##                   Kappa : 0.0982
##
##  Mcnemar's Test P-Value : < 2e-16
##
##             Sensitivity : 0.99049
##             Specificity : 0.07614
##          Pos Pred Value : 0.79417
##          Neg Pred Value : 0.68981
##              Prevalence : 0.78256
##          Detection Rate : 0.77511
##    Detection Prevalence : 0.97600
##       Balanced Accuracy : 0.53331
##
##        'Positive' Class : 0
##
```

The ROC plot for this model is:

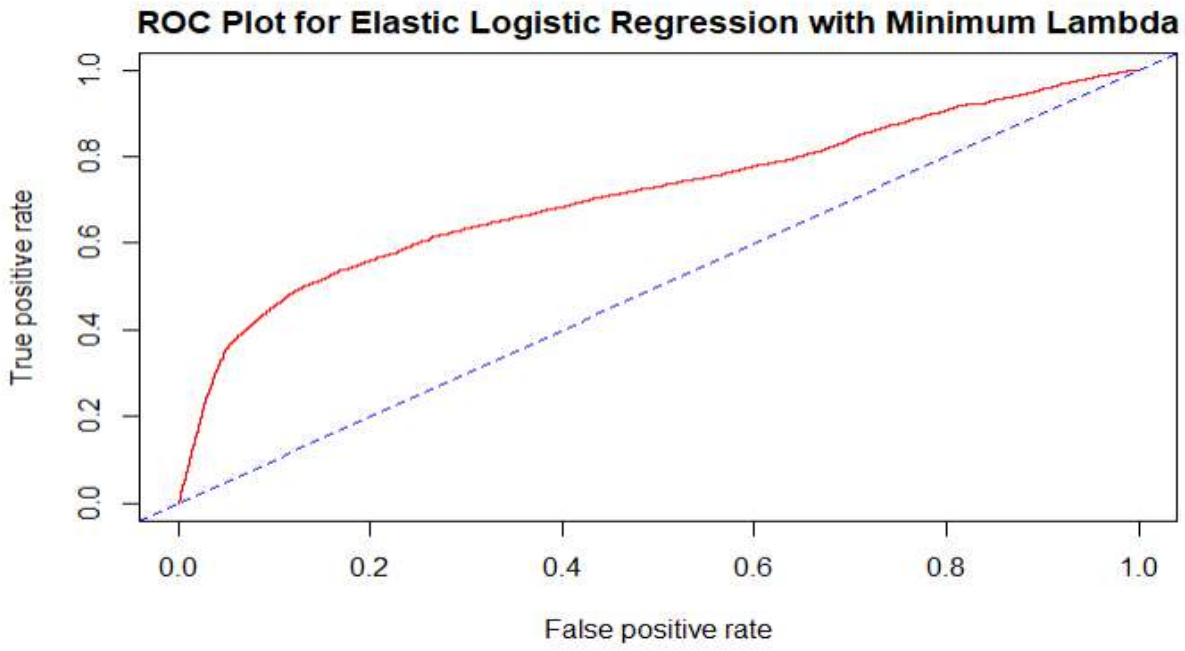

Figure 11: ROC curve for elastic net logistic regression with minimum lambda

As expected, this model (Elastic Net logistic regression with minimum lambda) is not better than the logistic model without regularization.

3.2.3.2 Elastic Net Logistic Regression with 1SE Lambda

The following table shows the coefficient estimates for elastic net logistic regression with 1se lambda.

Table 14: Coefficient estimates for elastic net logistic regression with 1se lambda

```
## 31 x 1 sparse Matrix of class "dgCMatrix"
##                          s0
## (Intercept) -1.167891e+00
## LIMIT_BAL   -7.602935e-07
## SEX2        -8.335531e-02
## EDUCATION1   1.692709e-02
## EDUCATION2   .
## EDUCATION3   .
## EDUCATION4  -2.905334e-01
## EDUCATION5  -8.952351e-01
## EDUCATION6   .
## MARRIAGE1    6.796845e-02
## MARRIAGE2   -5.411133e-02
## MARRIAGE3    .
## AGE          3.946661e-03
## PAY_0        5.613162e-01
## PAY_2        8.710651e-02
## PAY_3        6.085171e-02
## PAY_4        4.283697e-02
## PAY_5        1.144969e-02
## PAY_6        1.263461e-02
## BILL_AMT1   -1.239962e-06
## BILL_AMT2    .
## BILL_AMT3    .
## BILL_AMT4    .
## BILL_AMT5    .
```

```
## BILL_AMT6    .
## PAY_AMT1    -7.495105e-06
## PAY_AMT2    -5.096905e-06
## PAY_AMT3    -3.845449e-08
## PAY_AMT4    -3.247873e-06
## PAY_AMT5    -2.892191e-06
## PAY_AMT6    -2.299447e-06
```

This model has 8 zero coefficient estimates and several more that are close to zero. The test confusion matrix for this model is:

Table 15: Confusion matrix for elastic net logistic regression with 1se lambda

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 6988 1834
##          1   55  123
##
##                Accuracy : 0.7901
##                  95% CI : (0.7815, 0.7985)
##     No Information Rate : 0.7826
##     P-Value [Acc > NIR] : 0.04184
##
##                   Kappa : 0.0819
##
##  Mcnemar's Test P-Value : < 2e-16
##
##             Sensitivity : 0.99219
##             Specificity : 0.06285
##          Pos Pred Value : 0.79211
##          Neg Pred Value : 0.69101
##              Prevalence : 0.78256
##          Detection Rate : 0.77644
##    Detection Prevalence : 0.98022
##       Balanced Accuracy : 0.52752
##
##        'Positive' Class : 0
##
```

The ROC curve for this model is:

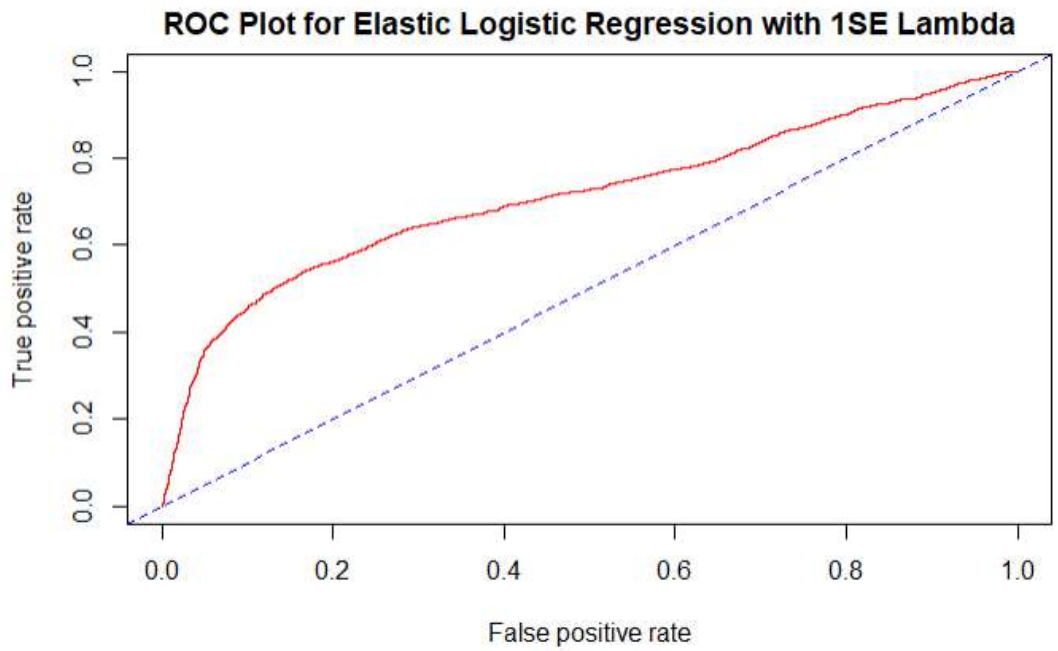

Figure 12: ROC curve for elastic net logistic regression with 1se lambda

As we see above, the elastic net logistic regression with 1SE lambda does not perform better than logistic regression without regularization.

For the dataset we have, regularization does not seem to produce better classification than that of a logistic regression model without regularization.

3.3 Linear Discriminant Analysis

Suppose that the joint distribution of $\boldsymbol{X}_{nxp}$ for population π_1 and π_2 are given by:

$$f_i(\boldsymbol{X}) = \frac{1}{(2\pi)^{p/2}|\sum_i|^{1/2}} exp\left[-\frac{1}{2}(\boldsymbol{X}-\boldsymbol{\mu}_i)'\sum_i^{-1}(\boldsymbol{X}-\boldsymbol{\mu}_i)\right] \; for\; i = 1, 2$$

When, $\sum_1 = \sum_2 = \sigma^2 I$, we have a classification problem that can be approached with linear discriminant analysis.

LDA makes predictions by comparing the likelihood equations for k populations. If the likelihood of the k^{th} population is the highest given the $\boldsymbol{x}$ values, the observation belongs to the k^{th} class. It is called linear because, the score equation is linear on the **x's.**

The following table below shows the coefficient estimates for linear discriminant analysis.

Table 16: Coefficient estimates of linear discriminant analysis

```
## Coefficients of linear discriminants:
##                     LD1
## LIMIT_BAL -7.318948e-07
## SEX       -1.280725e-01
## EDUCATION -1.268751e-01
## MARRIAGE  -1.475228e-01
## AGE        1.169514e-02
```

```
## PAY_0      6.964112e-01
## PAY_2      1.475844e-01
## PAY_3      6.784774e-02
## PAY_4      4.563729e-02
## PAY_5      1.734251e-02
## PAY_6      1.408649e-02
## BILL_AMT1 -4.976915e-06
## BILL_AMT2  1.146277e-06
## BILL_AMT3  7.249930e-07
## BILL_AMT4  8.208421e-08
## BILL_AMT5  4.662013e-07
## BILL_AMT6 -1.137659e-08
## PAY_AMT1  -6.044924e-06
## PAY_AMT2  -3.314688e-06
## PAY_AMT3  -2.443997e-07
## PAY_AMT4  -2.936573e-06
## PAY_AMT5  -2.375356e-06
## PAY_AMT6  -1.659504e-06
```

The confusion matrix below for LDA results in an accuracy of .8148 and a specificity of .2678.

Table 17: Confusion matrix of linear discriminant analysis

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 6809 1433
##          1  234  524
##
##                Accuracy : 0.8148
##                  95% CI : (0.8066, 0.8228)
##     No Information Rate : 0.7826
##     P-Value [Acc > NIR] : 2.399e-14
##
##                   Kappa : 0.3012
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9668
##             Specificity : 0.2678
##          Pos Pred Value : 0.8261
##          Neg Pred Value : 0.6913
##              Prevalence : 0.7826
##          Detection Rate : 0.7566
##    Detection Prevalence : 0.9158
##       Balanced Accuracy : 0.6173
##
##        'Positive' Class : 0
##
```

The ROC plot below for LDA results in an AUC of .71 denoting the model as a good classifier.

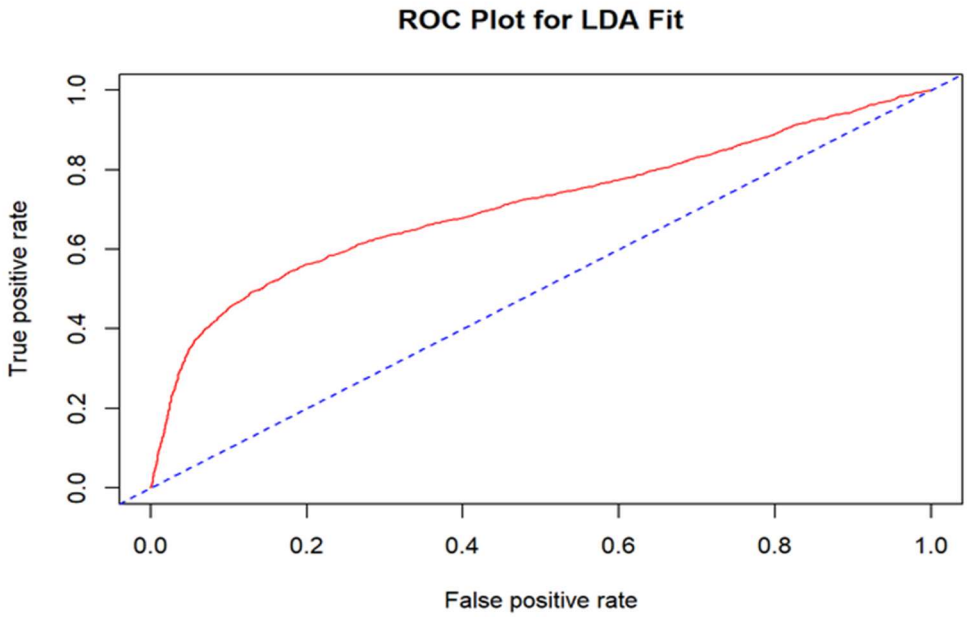

Figure 13: ROC curve of linear discriminant analysis

3.4 Quadratic Discriminant Analysis

The Quadratic Discriminant Analysis (QDA) is almost similar to the LDA above. The difference is that there is no assumption that the covariance of each classes is identical. As such,

$$\sum_1 \neq \sum_2$$

Unlike linear discriminant analysis, this analysis is called quadratic discriminant analysis because the score function is quadratic on the x's. Again, a decision is made by comparing score functions for all classes.

The output below contains the group means similar to LDA. However, it does not contain the coefficients like that of the linear discriminant analysis because the QDA classifier involves a quadratic, rather than a linear function of the predictors.

Table 18: Group means for quadratic discriminant analysis

```
## Call:
## qda(default ~ ., data = train.data)
##
## Prior probabilities of groups:
##         0         1
## 0.7771905 0.2228095
##
## Group means:
##   LIMIT_BAL      SEX EDUCATION MARRIAGE      AGE      PAY_0      PAY_2
## 0  178922.0 1.614852  1.841186 1.556951 35.38741 -0.2105263 -0.3020648
## 1  128786.1 1.561017  1.892071 1.532379 35.69160  0.6815559  0.4703997


##        PAY_3      PAY_4      PAY_5      PAY_6 BILL_AMT1 BILL_AMT2 BILL_AMT3
## 0 -0.3196495 -0.3578825 -0.3912138 -0.4095337  51999.87  49599.26  47228.44
## 1  0.3665313  0.2635178  0.1724728  0.1160504  49067.60  47879.81  45802.48
```

```
##   BILL_AMT4 BILL_AMT5 BILL_AMT6 PAY_AMT1 PAY_AMT2 PAY_AMT3 PAY_AMT4 PAY_AMT5
## 0  43381.92  40361.24  39014.92 6196.894 6459.934 5633.731 5274.959 5307.233
## 1  42654.90  39870.90  38571.10 3342.226 3346.063 3435.748 2980.726 3142.485


##   PAY_AMT6
## 0 5712.659
## 1 3322.505
```

The confusion matrix QDA results in an accuracy of .5238 and a specificity of .7951.

Table 19: Confusion matrix for quadratic discriminant analysis

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 3158  401
##          1 3885 1556
##
##                Accuracy : 0.5238
##                  95% CI : (0.5134, 0.5341)
##     No Information Rate : 0.7826
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1482
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.4484
##             Specificity : 0.7951
##          Pos Pred Value : 0.8873
##          Neg Pred Value : 0.2860
##              Prevalence : 0.7826
##          Detection Rate : 0.3509
##    Detection Prevalence : 0.3954
##       Balanced Accuracy : 0.6217
##
##        'Positive' Class : 0
##
```

The ROC plot below for QDA results in an AUC of 0.71 denoting the model as a good classifier.

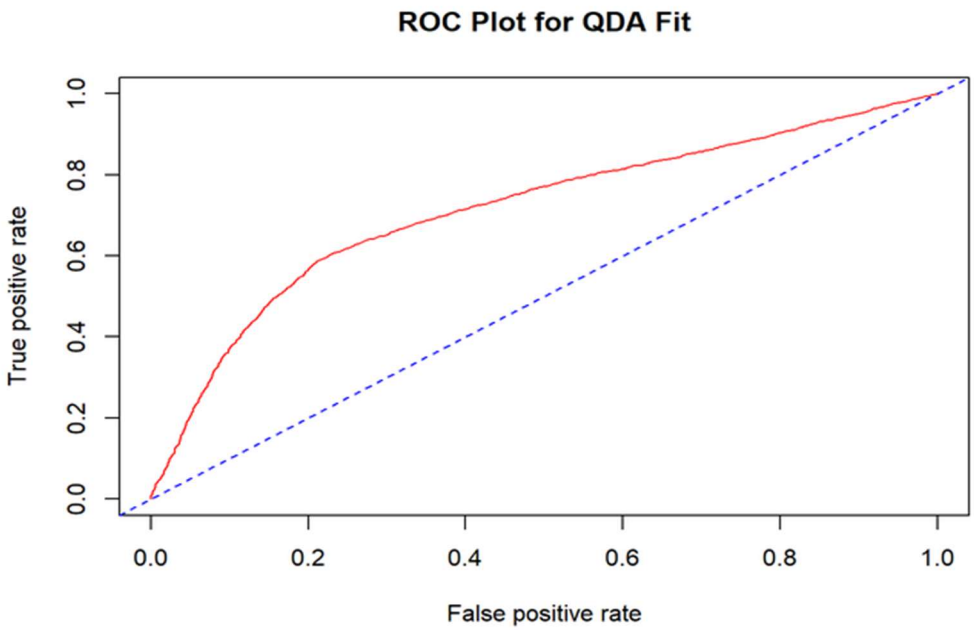

Figure 14: ROC curve for quadratic discriminant analysis

3.5 Naïve Bayes

The Naïve Bayes method is a simple supervised learning algorithm based on applying Bayes' theorem. The method is best suited when the dimensions of inputs is high. In the case of Naïve Bayes, we assume the covariance matrices to take the following form:

$$\boldsymbol{\Sigma}_1 = \sigma_1^2\boldsymbol{\Sigma}\ and\ \boldsymbol{\Sigma}_2 = \sigma_2^2\boldsymbol{\Sigma}$$

Where the two populations covariance share the same covariance structure after $\sigma_1^2\ and\ \sigma_2^2$ are factored out from their respective covariance. Naïve Bayes score function is quadratic on the x's. The confusion matrix below for Naïve Bayes results in an accuracy of .7508 and a specificity of .5989.

Table 20: Confusion matrix of naive Bayes analysis

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 5585  785
##          1 1458 1172
##
##                Accuracy : 0.7508
##                  95% CI : (0.7417, 0.7597)
##     No Information Rate : 0.7826
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.3486
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.7930
##             Specificity : 0.5989
##          Pos Pred Value : 0.8768
##          Neg Pred Value : 0.4456
```

```
##              Prevalence : 0.7826
##          Detection Rate : 0.6206
##    Detection Prevalence : 0.7078
##       Balanced Accuracy : 0.6959
##
##        'Positive' Class : 0
##
```

The ROC plot below for Naïve Bayes results in an AUC of 0.73 denoting the model as a good classifier.

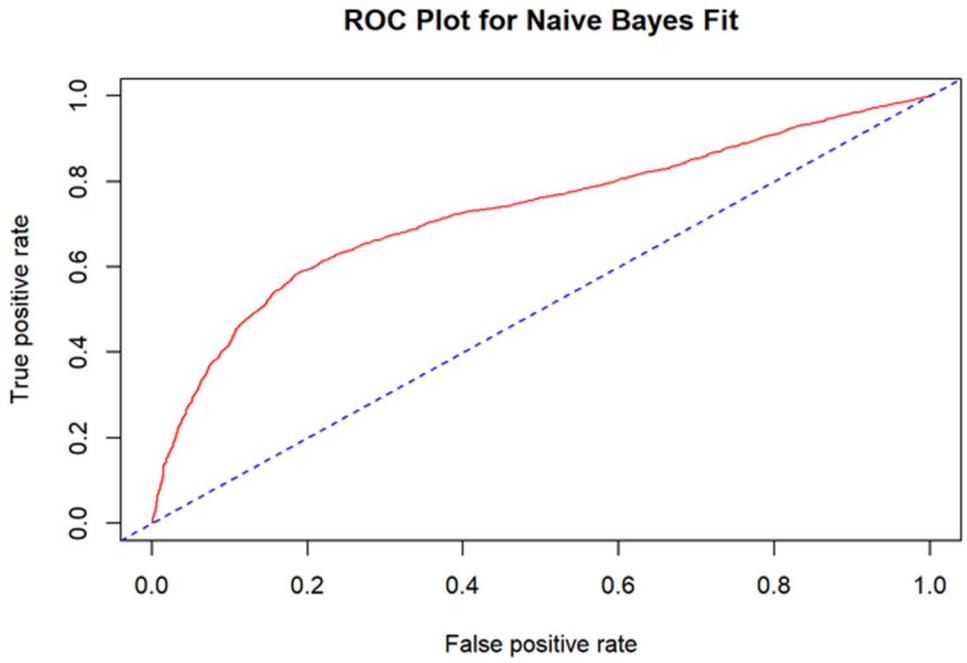

Figure 15: ROC curve of naive bayes analysis

3.6 K-Nearest Neighbors

The KNN is a classifier that takes positive integer K and first identifies K points that are nearest to x_0, represented by N_0. Then, it estimates the conditional probability for class j based on the fraction of points N_0 that have a response equal to j.

The estimated conditional probability can be stated as

$$\Pr(Y=j|X=x_0) = \frac{1}{k}\sum_{i\varepsilon N_0} I(y_i = j)$$

The KNN classifier then applies the Bayes theorem and yields the classification with the highest probability.

3.6.1 1 Nearest Neighbors

The confusion matrix below 1 nearest neighbor results in an accuracy of .6969 and a specificity of .3020.

Table 21: Confusion matrix for 1NN analysis

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 5681 1366
##          1 1362  591
```

```
##
##                Accuracy : 0.6969
##                  95% CI : (0.6873, 0.7064)
##     No Information Rate : 0.7826
##     P-Value [Acc > NIR] : 1.0000
##
##                   Kappa : 0.1087
##
##  Mcnemar's Test P-Value : 0.9542
##
##             Sensitivity : 0.8066
##             Specificity : 0.3020
##          Pos Pred Value : 0.8062
##          Neg Pred Value : 0.3026
##              Prevalence : 0.7826
##          Detection Rate : 0.6312
##    Detection Prevalence : 0.7830
##       Balanced Accuracy : 0.5543
##
##        'Positive' Class : 0
##
```

3.6.2 5 Nearest Neighbors

The confusion matrix for 5 nearest neighbors is results in an accuracy of .7953 and a specificity of .1978.

Table 22: Confusion matrix for 5NN analysis

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 6447 1570
##          1  596  387
##
##                Accuracy : 0.7593
##                  95% CI : (0.7504, 0.7681)
##     No Information Rate : 0.7826
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1379
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.9154
##             Specificity : 0.1978
##          Pos Pred Value : 0.8042
##          Neg Pred Value : 0.3937
##              Prevalence : 0.7826
##          Detection Rate : 0.7163
##    Detection Prevalence : 0.8908
##       Balanced Accuracy : 0.5566
##
##        'Positive' Class : 0
##
```

3.6.3 10 Nearest Neighbors

The confusion matrix below for 10 nearest neighbor results in an accuracy of .769 and a specificity of .1497.

Table 23: Confusion matrix for 10NN analysis

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 6628 1664
##          1  415  293
##
##                Accuracy : 0.769
##                  95% CI : (0.7602, 0.7777)
##     No Information Rate : 0.7826
##     P-Value [Acc > NIR] : 0.9991
##
##                   Kappa : 0.118
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.9411
##             Specificity : 0.1497
##          Pos Pred Value : 0.7993
##          Neg Pred Value : 0.4138
##              Prevalence : 0.7826
##          Detection Rate : 0.7364
##    Detection Prevalence : 0.9213
##       Balanced Accuracy : 0.5454
##
##        'Positive' Class : 0
##
```

3.7 Principal Component Analysis

We use principal component analysis (PCA) to transform the number of correlated variables into smaller number of uncorrelated variables called principle components. PCA is most commonly used to condense the information contained in many original variables into a smaller set of new composite dimensions, with a minimum loss of information.

Let $\boldsymbol{X}_{nxp}$ have a covariance matrix $\boldsymbol{\Sigma}$, with eigen value-eigenvector pairs $(\lambda_1, \boldsymbol{\varepsilon}_1)$, $(\lambda_2, \boldsymbol{\varepsilon}_2)$, ...,

$(\lambda_p, \boldsymbol{\varepsilon}_p)$ where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$.

Let $\boldsymbol{W}_{\boldsymbol{p\,x\,p}} = (\boldsymbol{\varepsilon}_1, \boldsymbol{\varepsilon}_2, \ldots, \boldsymbol{\varepsilon}_p)$

Then, $\boldsymbol{T} = \boldsymbol{XW}$ are the principal components of **X.** The first **L** principal components are then expressed as $\boldsymbol{T}_{\boldsymbol{L}} = \boldsymbol{XW}_{\boldsymbol{L}}$ where $\boldsymbol{W}_{\boldsymbol{L}} = (\boldsymbol{\varepsilon}_1, \boldsymbol{\varepsilon}_2, \ldots, \boldsymbol{\varepsilon}_L)\ for\ L \leq p$

In our case, we standardized the **X** variables. As such, the principal components are obtained from the correlation matrix of $\boldsymbol{Z}_{n\,x\,p}$,

where,

$\boldsymbol{Z} = \left(\boldsymbol{\Sigma}^{1/2}\right)^{-1}(\boldsymbol{X} - \boldsymbol{\mu})$ *with* $Cov(\boldsymbol{Z}) = \boldsymbol{\rho}$, *which is a correlation matrix with* $(\lambda_1, \boldsymbol{\varepsilon}_1)$,

$(\lambda_2, \boldsymbol{\varepsilon}_2), \ldots, (\lambda_p, \boldsymbol{\varepsilon}_p)$ eigenvalue - eigenvector pairs. However, the $(\lambda_i, \boldsymbol{\varepsilon}_i)$ derived from $\boldsymbol{\Sigma}$ are, in general, not same as those derived from $\boldsymbol{\rho}$.

Then the principal components (**S**) derived from **Z** are expressed as:

$$\boldsymbol{S} = \boldsymbol{ZW} = \left(\boldsymbol{\Sigma}^{1/2}\right)^{-1}(\boldsymbol{X} - \boldsymbol{\mu})\boldsymbol{W}, \qquad and\ the\ first\ L\ principal\ componenets\ of\ \ \boldsymbol{Z}\ are$$

$$\boldsymbol{S_L} = \boldsymbol{ZW_L} = \left(\boldsymbol{\Sigma}^{1/2}\right)^{-1}(\boldsymbol{X} - \boldsymbol{\mu})\boldsymbol{W_L} \text{ where } \boldsymbol{W_L}\ is\ a\ pxL\ matrix\ of\ eigenvectors\ of\ \boldsymbol{\rho}.$$

The following table displays the results after transforming the data into components using PCA. Here we select the first 15 principal components since it explains 95.7% of the variation of the independent variables.

Table 24: Importance of Principal Components table

```
## Importance of components:
##                            PC1    PC2     PC3     PC4     PC5     PC6    C7
## Standard deviation     2.5579 2.0244 1.24538 1.21337 1.01254 0.97837 0.967
## Proportion of Variance 0.2845 0.1782 0.06743 0.06401 0.04458 0.04162 0.046
## Cumulative Proportion  0.2845 0.4627 0.53010 0.59411 0.63869 0.68031 0.777
##                            PC8     PC9    PC10    PC11    PC12    PC13  PC14
## Standard deviation     0.94213 0.93341 0.88483 0.85603 0.82633 0.75561 0.241
## Proportion of Variance 0.03859 0.03788 0.03404 0.03186 0.02969 0.02482 0.0228
## Cumulative Proportion  0.75836 0.79624 0.83028 0.86214 0.89183 0.91665 0.9395
##                           PC15   PC16    PC17   PC18    PC19    PC20    PC21
## Standard deviation     0.63533 0.5098 0.49913 0.4344 0.36302 0.26487 0.20195
## Proportion of Variance 0.01755 0.0113 0.01083 0.0082 0.00573 0.00305 0.00177
## Cumulative Proportion  0.95700 0.9683 0.97913 0.9873 0.99307 0.99612 0.99789
##                           PC22    PC23
## Standard deviation     0.1590 0.15238
## Proportion of Variance 0.0011 0.00101
## Cumulative Proportion  0.9990 1.00000
```

The graph below also displays the variance, proportion, and cumulative proportion of the PCA.

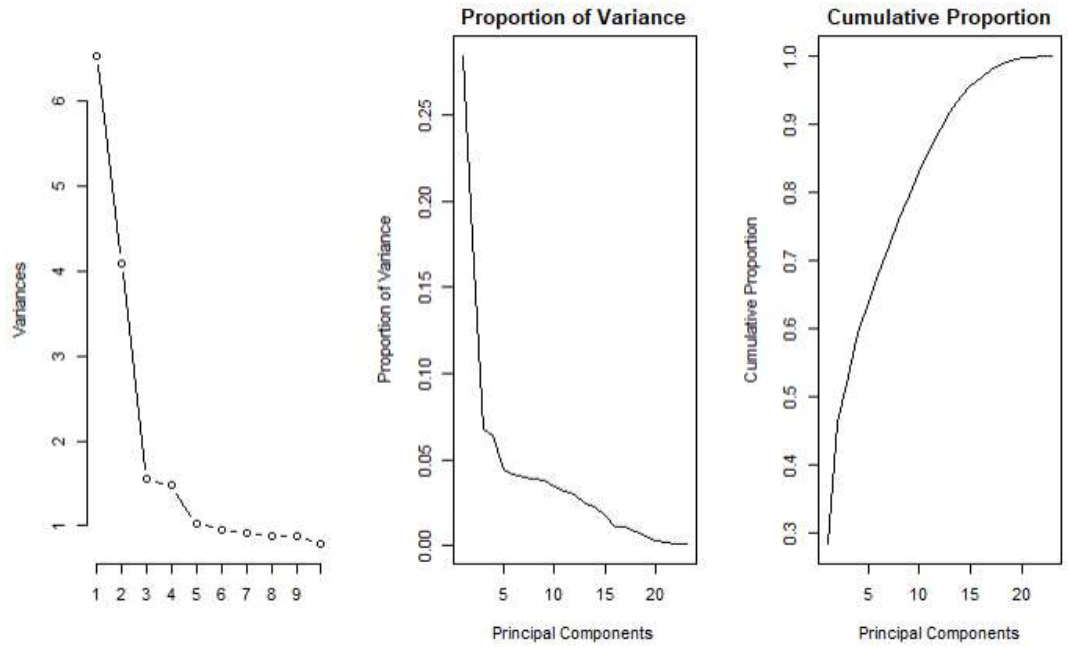

Figure 16: Scree, proportion of variance and cumulative proportion plots

As we see above, the first 15 principal components retain 95.7% of the variation in all 23 independent variables. This means, we have reduced the dimension of the independent variables by 8 with minimum loss of variability of the independent variables.

Based on the first 15 principal components, we fitted logistic regression, linear discriminant analysis, quadratic discriminant analysis, naïve bayes, KNN=1, KNN=5, and KNN=10.

3.7.1 Logistic Regression on Principal Components

The table below shows the coefficient estimates for logistic regression on PCA. However, it is impossible to interpret the coefficients since PCA reduces the dimensions of the data and transforms them into components.

Table 25: Coefficient estimates of logistic regression on 15 principal components

```
##
## Call:
## glm(formula = default ~ ., family = binomial, data = train.pca.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1749  -0.7006  -0.5506  -0.2805   3.7212
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.464919   0.019906 -73.593  < 2e-16 ***
## PC1          0.076086   0.008365   9.096  < 2e-16 ***
## PC2         -0.411818   0.011674 -35.277  < 2e-16 ***
## PC3          0.193850   0.023856   8.126 4.44e-16 ***
## PC4          0.082934   0.024791   3.345 0.000822 ***
## PC5         -0.135958   0.018789  -7.236 4.61e-13 ***
## PC6         -0.116189   0.025351  -4.583 4.58e-06 ***
## PC7          0.005401   0.027343   0.198 0.843430
## PC8          0.061461   0.031934   1.925 0.054272 .
## PC9         -0.021671   0.034335  -0.631 0.527936
## PC10        -0.082381   0.033299  -2.474 0.013360 *
## PC11         0.017050   0.044394   0.384 0.700929
## PC12         0.420388   0.022811  18.429  < 2e-16 ***
## PC13         0.058170   0.023963   2.427 0.015204 *
## PC14         0.045507   0.026555   1.714 0.086594 .
## PC15         0.334986   0.026761  12.518  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 22279  on 20999  degrees of freedom
## Residual deviance: 19529  on 20984  degrees of freedom
## AIC: 19561
##
## Number of Fisher Scoring iterations: 5
```

The confusion matrix below for logistic regression on PCA results in an accuracy of 0.813 and a specificity of 0.698.

Table 26: Confusion matrix of logistic regression on 15 principal components

```
## Confusion Matrix and Statistics
##
##           Reference
```

```
## Prediction    0    1
##          0 6834  209
##          1 1474  483
##
##                Accuracy : 0.813
##                  95% CI : (0.8048, 0.821)
##     No Information Rate : 0.9231
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.2832
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.8226
##             Specificity : 0.6980
##          Pos Pred Value : 0.9703
##          Neg Pred Value : 0.2468
##              Prevalence : 0.9231
##          Detection Rate : 0.7593
##    Detection Prevalence : 0.7826
##       Balanced Accuracy : 0.7603
##
##        'Positive' Class : 0
##
```

The ROC plot below for logistic regression on PCA results in an AUC of 0.709.

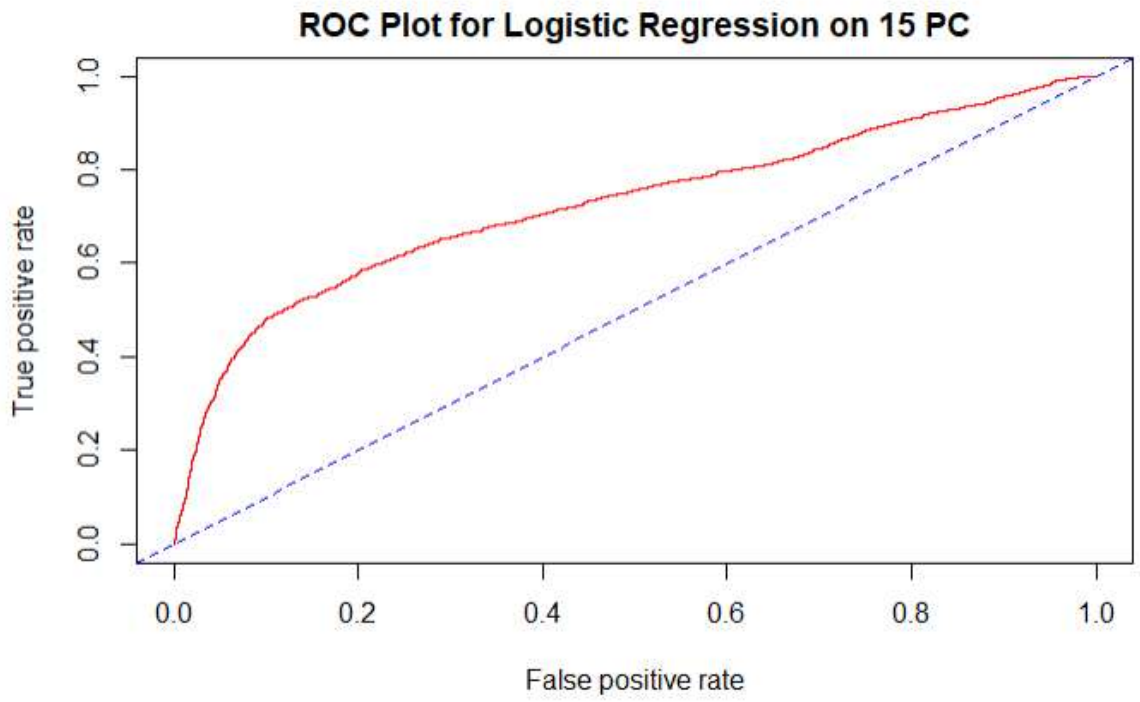

Figure 17: ROC curve of logistic regression on 15 principal components

3.7.2 LDA on Principal Components

The following table below shows the coefficient estimates for LDA on PCA. However, it is impossible to interpret the coefficients since PCA reduce the dimensions of the data and transform them into components.

Table 27: Coefficient estimates of LDA on 15 principal components

```
## Call:
## lda(default ~ ., data = train.pca.data)
##
## Prior probabilities of groups:
##         0         1
## 0.7771905 0.2228095
##
## Group means:
##          PC1        PC2         PC3         PC4         PC5          PC6
## 0 -0.1225284  0.3072276 -0.01381045  0.02832730  0.01976383  0.009016457
## 1  0.4251741 -1.0946230  0.06995205 -0.06671546 -0.08457388 -0.052108600
##            PC7          PC8           PC9         PC10         PC11        PC12
## 0 -0.004560266 -0.002656818 -0.0006838582  0.002817074 -0.008998477 -0.06536074
## 1 -0.003397728  0.013519765  0.0057861491 -0.015766462  0.029673742  0.23700825
##          PC13         PC14        PC15
## 0 -0.01043197 -0.007668732 -0.02577882
## 1  0.03146562  0.008191120  0.09741854
##
## Coefficients of linear discriminants:
##               LD1
## PC1   0.102214128
## PC2  -0.433970748
## PC3   0.087280135
## PC4  -0.064896957
## PC5  -0.132849447
## PC6  -0.103133282
## PC7   0.002391771
## PC8   0.030726586
## PC9   0.004269961
## PC10 -0.032113514
## PC11  0.049727326
## PC12  0.543813545
## PC13  0.100927918
## PC14  0.031708111
## PC15  0.387741337
```

The confusion matrix for LDA on PCA results in an accuracy of 0.8127 and a specificity of 0.2596.

Table 28: Confusion matrix of LDA on 15 principal components

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 6806 1449
##          1  237  508
##
##                Accuracy : 0.8127
##                  95% CI : (0.8044, 0.8207)
##     No Information Rate : 0.7826
##     P-Value [Acc > NIR] : 1.01e-12
##
##                   Kappa : 0.291
```

```
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9663
##             Specificity : 0.2596
##          Pos Pred Value : 0.8245
##          Neg Pred Value : 0.6819
##              Prevalence : 0.7826
##          Detection Rate : 0.7562
##    Detection Prevalence : 0.9172
##       Balanced Accuracy : 0.6130
##
##        'Positive' Class : 0
##
```

The ROC plot below for LDA on PCA results in an AUC of 0.5037.

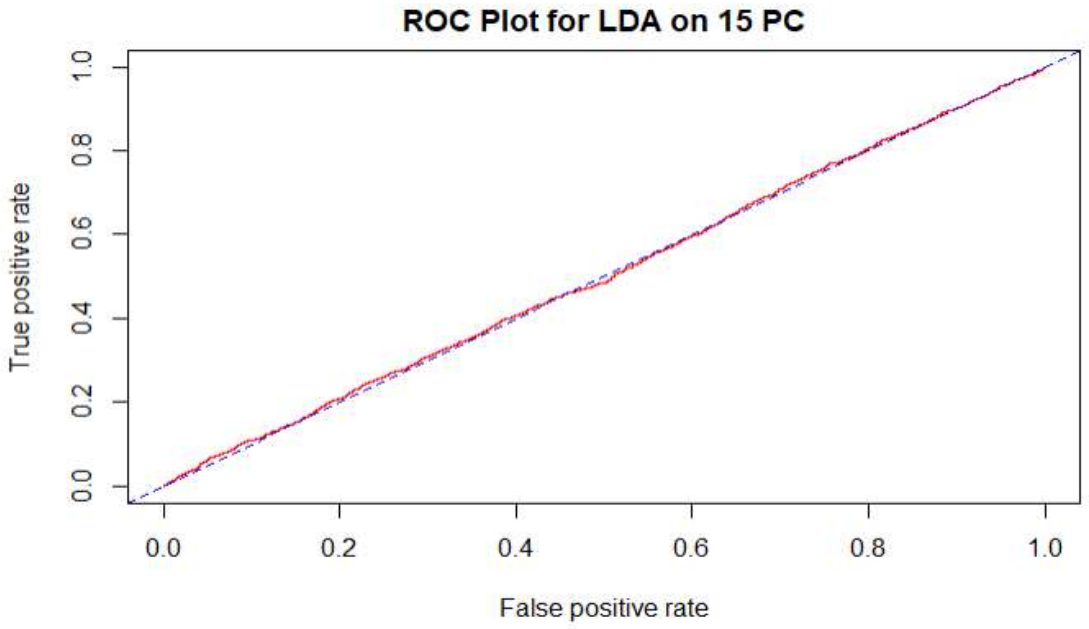

Figure 18: ROC curve of LDA on 15 principal components

3.7.3 QDA on Principal Components

Below are the mean coefficients for QDA on PCA. Similar to the original QDA, there are no coefficients because the QDA classifier involves a quadratic, rather than a linear function of the predictors.

Table 29: QDA group means for 15 principal components

```
## Call:
## qda(default ~ ., data = train.pca.data)
##
## Prior probabilities of groups:
##         0         1
## 0.7771905 0.2228095
##
## Group means:
##          PC1        PC2         PC3         PC4         PC5          PC6
```

```
## 0 -0.1225284  0.3072276 -0.01381045  0.02832730  0.01976383  0.009016457
## 1  0.4251741 -1.0946230  0.06995205 -0.06671546 -0.08457388 -0.052108600
##            PC7          PC8           PC9         PC10         PC11        PC12
## 0 -0.004560266 -0.002656818 -0.0006838582  0.002817074 -0.008998477 -0.06536074
## 1 -0.003397728  0.013519765  0.0057861491 -0.015766462  0.029673742  0.23700825
##          PC13         PC14        PC15
## 0 -0.01043197 -0.007668732 -0.02577882
## 1  0.03146562  0.008191120  0.09741854
```

The confusion matrix for QDA on PCA results in an accuracy of 0.51 and a specificity of 0.7931.

Table 30: Confusion matrix of QDA on 15 principal components

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 3043  405
##          1 4000 1552
##
##                Accuracy : 0.5106
##                  95% CI : (0.5002, 0.5209)
##     No Information Rate : 0.7826
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1353
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.4321
##             Specificity : 0.7931
##          Pos Pred Value : 0.8825
##          Neg Pred Value : 0.2795
##              Prevalence : 0.7826
##          Detection Rate : 0.3381
##    Detection Prevalence : 0.3831
##       Balanced Accuracy : 0.6126
##
##        'Positive' Class : 0
##
```

The ROC plot for QDA on PCA results in an AUC of 1 denoting the model as a perfect classifier.

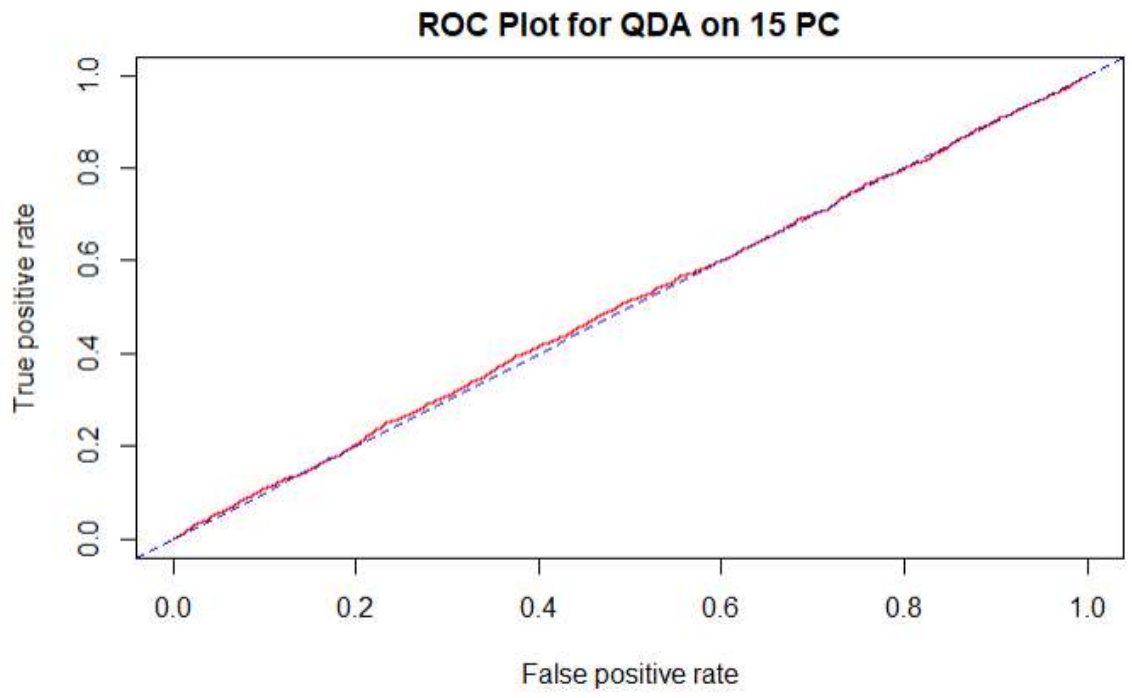

Figure 19: ROC curve of QDA on 15 principal components

3.7.4 Naïve Bayes on Principal Components

The confusion matrix for Naïve Bayes on PCA results in a .7373 accuracy and a .5933 specificity.

Table 31: Confusion matrix of naive bayes on 15 principal components

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 5475  796
##          1 1568 1161
##
##                Accuracy : 0.7373
##                  95% CI : (0.7281, 0.7464)
##     No Information Rate : 0.7826
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.3244
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.7774
##             Specificity : 0.5933
##          Pos Pred Value : 0.8731
##          Neg Pred Value : 0.4254
##              Prevalence : 0.7826
##          Detection Rate : 0.6083
##    Detection Prevalence : 0.6968
##       Balanced Accuracy : 0.6853
##
##        'Positive' Class : 0
##
```

The ROC plot for naïve bayes on PCA results in an AUC of .97 denoting the model as a good classifier.

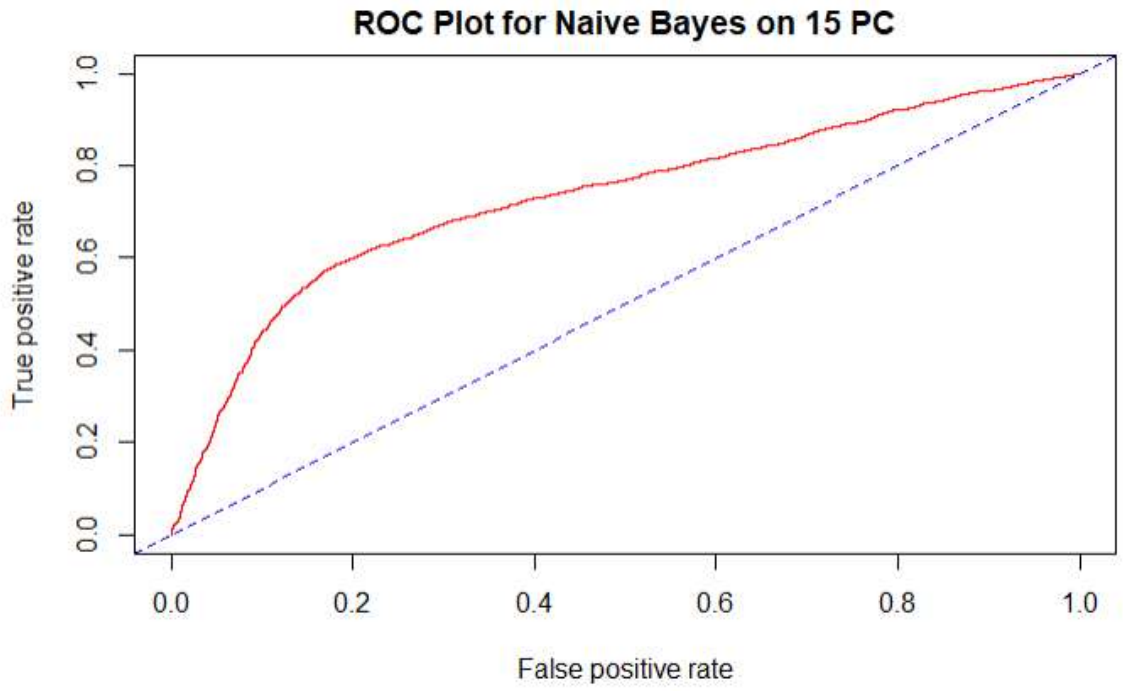

Figure 20: ROC curve of naive bayes on 15 principal components

3.7.5 KNN on Principal Components

Principle Component Analysis on the data was also fitted into K-Nearest Neighbors. Here, we use KNN=1, KNN=5, and KNN=10.

3.7.5.1 1NN on Principal Components

The confusion matrix for 1NN on PCA results in a .9431 accuracy rating and a .8380 specificity rating.

Table 32: Confusion matrix of 1NN on 15 principal components

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 6848  317
##          1  195 1640
##
##                Accuracy : 0.9431
##                  95% CI : (0.9381, 0.9478)
##     No Information Rate : 0.7826
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.829
##
##  Mcnemar's Test P-Value : 8.918e-08
##
##             Sensitivity : 0.9723
##             Specificity : 0.8380
##          Pos Pred Value : 0.9558
```

```
##          Neg Pred Value : 0.8937
##              Prevalence : 0.7826
##          Detection Rate : 0.7609
##    Detection Prevalence : 0.7961
##       Balanced Accuracy : 0.9052
##
##        'Positive' Class : 0
##
```

3.7.5.2 5NN on Principal Components

The confusion matrix below for 5NN on PCA displays a .9439 accuracy rating and a .7976 specificity rating.

Table 33: Confusion matrix of 5NN on 15 principal components

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 6935  395
##          1  108 1562
##
##                Accuracy : 0.9441
##                  95% CI : (0.9392, 0.9488)
##     No Information Rate : 0.7826
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.8266
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9847
##             Specificity : 0.7982
##          Pos Pred Value : 0.9461
##          Neg Pred Value : 0.9353
##              Prevalence : 0.7826
##          Detection Rate : 0.7706
##    Detection Prevalence : 0.8144
##       Balanced Accuracy : 0.8914
##
##        'Positive' Class : 0
##
```

3.7.5.3 10NN on Principal Components

The confusion matrix for 10NN on PCA results in a .9354 accuracy rating and a 0.7547 specificity rating.

Table 34: Confusion matrix of 10NN on 15 principal components

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 6943  487
##          1  100 1470
##
##                Accuracy : 0.9348
##                  95% CI : (0.9295, 0.9398)
##     No Information Rate : 0.7826
```

```
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7936
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9858
##             Specificity : 0.7511
##          Pos Pred Value : 0.9345
##          Neg Pred Value : 0.9363
##              Prevalence : 0.7826
##          Detection Rate : 0.7714
##    Detection Prevalence : 0.8256
##       Balanced Accuracy : 0.8685
##
##        'Positive' Class : 0
##
```

4. Results

In the tables below, the performances of all models fitted for this project are listed. In total, we have fitted 20 models. 8 logistic models (1 on the original data, 6 with regularization and 1 on 15 principal components), 2 linear discriminant models (1 on the original data and 1 on 15 principal components), 2 quadratic discriminant models (1 on the original data and 1 on 15 principal components), 2 naïve bayes models (1 on the original data and 1 on 15 principal components), 2 KNN for K=1 models (1 on the original data and 1 on 15 principal components), 2 KNN for K=5 models (1 on the original data and 1 on 15 principal components), 2KNN for K=10 models (1 on the original data and 1 on the first 15 principal components).

On the first set of models fitted on the original data (table 35), LDA performs well on overall accuracy. Since we have an imbalanced data, with a smaller proportion of defaulters (figure 1), selecting a model based on its performance on accuracy is not recommended. Since the goal of the project is to identify loan defaulters with a minimum loss of misclassification, comparing specificity is an obvious method of choice to identify a model that does not lead to recommending loans to possible defaulters. Therefore, based on specificity QDA will be the model of choice if we intend to retain the data during modeling.

Table 35: Statistics for 7 classification models on the original data

	Fitted Models						
	Logistic	**LDA**	**QDA**	**KNN=1**	**KNN=5**	**KNN=10**	**Naive Bayes**
Accuracy	0.8138	0.8148	0.5238	0.6969	0.7593	0.7690	0.7508
Sensitivity	0.8237	0.9668	0.4484	0.8066	0.9154	0.9411	0.7930
Specificity	0.6976	0.2678	0.7951	0.3020	0.1978	0.1497	0.5989
Kappa	0.2895	0.3012	0.1482	0.1087	0.1379	0.1180	0.3486
AUC	0.7127	0.7084	0.7134	NA	NA	NA	0.7267

For comparison purposes (table 36), we have also fitted six regularized logistic regression models (Ridge, Lasso and Elastic Net) with different choices of tuning parameter (λ). Based on specificity, logistic regression without any regularization performs better than ridge, lasso, and elastic net logistic regression with minimum and 1se tuning parameter (λ).

Table 36: Statistics for 7 logistic regression models

	Fitted Models					
	Logistic	**Ridge Min**	**Ridge 1se**	**Lasso Min**	**Lasso 1se**	**Elastic Min**
Min Lambda	NA	0.0138	NA	5E-05	NA	0.0029
1se Lambda	NA	NA	0.0290	NA	0.006393292	NA
Accuracy	0.8138	0.7907	0.7889	0.7933	0.7887	0.7917
Sensitivity	0.8237	0.9913	0.9932	0.9898	0.9930	0.9905
Specificity	0.6976	0.0685	0.0537	0.0864	0.0531	0.0761
Kappa	0.2895	0.0886	0.0702	0.1114	0.0692	0.0982
AUC	0.7127	0.7125	0.7109	0.7131	0.7122	0.7131

Finally (table 37), after identifying the first 15 principal components of the scaled $\boldsymbol{X_{n\,x\,p}}$ matrix, we fitted logistic, LDA, QDA KNN for K=1, 5 and 10, and naïve bayes models.

Table 37: Statistics for models fitted with 15 principal components

	Fitted Models with 15 PC						
	Logistic.pca	**LDA.pca**	**QDA.pca**	**KNN=1.pca**	**KNN=5.pca**	**KNN=10.pca**	**Naive Bayes.pca**
Accuracy	0.8130	0.8127	0.5106	0.9431	0.9439	0.9354	0.7373
Sensitivity	0.8226	0.9663	0.4321	0.9723	0.9845	0.9857	0.7774
Specificity	0.6980	0.2596	0.7931	**0.8380**	0.7976	0.7547	0.5933
Kappa	0.2832	0.2910	0.1353	0.8290	0.8259	0.7961	0.3244
AUC	0.7091	0.7060	0.7215	NA	NA	NA	0.7095

Table 38: Comparison of models on specificity

	On Original Data	Regularized	On 15 PC
	QDA	**Lasso.min Logistic**	**KNN=1.pca**
Accuracy	0.5238	0.7933	0.9431
Sensitivity	0.4484	0.9898	0.9723
Specificity	0.7951	0.0864	**0.838**
Kappa	0.1482	0.1114	0.829
AUC	0.7134	0.7131	NA

Table 39: Comparison of models based on accuracy

	On Original Data	Regularized	On 15 PC
	LDA	**Lasso.min Logistic**	**KNN=5.pca**
Accuracy	0.8148	0.7933	0.9439
Sensitivity	0.9668	0.9898	0.9845
Specificity	0.2678	0.0864	0.7976
Kappa	0.3012	0.1114	0.8259
AUC	0.7084	0.7131	NA

If we take specificity as the criteria of choice (based on our response distribution, rightly so), KNN for K=1 model on the first 15 principal components is the model of choice. However, KNN for K=5 is the model of choice based on accuracy. The difference of test error between these two models is negligible.

For the sake of robustness, we calculated accuracy, sensitivity and specificity of these models for 100 different iterations of training and test data. The mean results are listed in the table below:

Table 40: Mean confusion matrices statistic for 100 iterations of training and test data

Results of 100 Iterations		
	1NN.pca	**5NN.pca**
Accuracy	0.9478356	0.9446311
Sensitivity	0.9766983	0.9859190
Specificity	0.8460610	0.7990507

Based on the results above the 1NN on the first 15 principal components performs better that of the 5NN model on 100 different iterations of training and test data.

5. Conclusion

We started this project with the aim of developing a model that reasonably identifies loan defaulters. Given certain features of a potential client, our mission was to predict whether that client will default on his/her loans.

From the get-go, our exploratory data analysis identified serious cases of multicollinearity on the independent variables. We tried to mitigate this problem using regularization methods (such as LASSO, RIDGE and Elastic Net) and dimension reduction method, namely, principal component analysis.

The other challenge we have identified in the data is the case of imbalance response on the default variable. Ideally, a response variable with equal number of classes makes modeling fun and easier. However, our data has 22%-78% distribution of default vs non default. This suggests that a mere measure of test accuracy would not be a perfect measure of model performance. As such, to stay in line with project objective, we agreed to concentrate more on test specificity of a fitted model than its test accuracy.

In all we fitted 20 separate models with varying degree of data manipulation and dimension reduction methods. In general, the KNN models fitted on the first 15 principal components perform better in test accuracy and sensitivity. Among them, the 1NN model on the first 15 principal components have a slightly higher specificity. As such, we chose this model to be the model of choice.

The 1NN model is more flexible than the 5NN model. This makes us skeptical to choose the 1NN as a model of choice without further investigation. The reason is the 1NN model on the first 15 principal components may not perform as well as it does on new datasets. To explore this skepticism more, we fitted each model on 100 different training PC datasets and compared their performance on their respective PC test datasets. The mean values of those statistics are reported in the results section of this report. Surprisingly, on average 1NN performs better in accuracy and specificity. On average, 5NN model performs better than 1NN on sensitivity. Since our project is more concerned with identifying defaulters, the 1NN model on the first 15 principal components is the model of choice.

For this dataset, 1NN on the first 15 principal components is a champion model.