ASSESSMENT CASE STUDY

Application of credit risk models in the banking industry

INDEX

DATA CLEANING

Given data has 19 variables. We are supposed to predict default with the help of logistic regression.

```
> library(readxl)
> credrisk <- read.csv("C:/Users/Animish/Downloads/Assessment Case Study Data.csv")
> str(credrisk)
'data.frame':	6000 obs. of  19 variables:
 $ i..cust_id          : int  6252029 5110070 2846491 9264318 9412980 6111903 1613014 7940321 1673336 2336197 ...
 $ acc_no              : num  6.25e+11 5.11e+11 2.85e+11 9.26e+11 9.41e+11 ...
 $ checking_balance    : chr  "< 0 USD" "1 - 200 USD" "< 0 USD" "1 - 200 USD" ...
 $ months_loan_duration: int  12 36 11 15 10 14 24 18 24 30 ...
 $ credit_history      : chr  "good" "good" "critical" "good" ...
 $ purpose             : chr  "car" "car" "car" "renovations" ...
 $ amount..USD.        : int  1274 12389 3939 1308 1924 3973 6615 2124 11938 2406 ...
 $ savings_balance     : chr  "< 100 USD" "unknown" "< 100 USD" "< 100 USD" ...
 $ employment_duration : chr  "< 1 year" "1 - 4 years" "1 - 4 years" "> 7 years" ...
 $ percent_of_income   : int  3 1 1 4 1 1 2 4 2 4 ...
 $ years_at_residence  : int  1 4 2 4 4 4 NA 4 3 NA ...
 $ age                 : int  37 37 40 38 38 22 75 24 39 23 ...
 $ other_credit        : chr  "none" "none" "none" "none" ...
 $ housing             : chr  "own" "other" "own" "own" ...
 $ existing_loans_count: int  1 1 2 2 1 1 2 2 2 1 ...
 $ job                 : chr  "unskilled" "skilled" "unskilled" "unskilled" ...
 $ dependants          : int  1 1 2 1 1 1 1 1 2 1 ...
 $ phone               : chr  "no" "yes" "no" "no" ...
 $ default             : chr  "yes" "yes" "no" "no" ...
```

Converting yes, no inputs of default column to 0, 1 using given code.

```
> credrisk$default <- factor(credrisk$default, levels = c("no", "yes"), labels = c(0, 1))
```

CHANGE FLAG VARIABLES INTO FACTOR

```
> cols <- c("checking_balance", "credit_history", "purpose", "savings_balance",
+           "employment_duration", "percent_of_income", "years_at_residence",
+           "other_credit", "housing", "job", "dependants", "phone", "default")
> credrisk[cols] <- lapply(credrisk[cols],factor)
> str(credrisk)
'data.frame':   6000 obs. of  19 variables:
 $ ï..cust_id          : int  6252029 5110070 2846491 9264318 9412980 6111903 1613014 7940321 1673336 2336197 ...
 $ acc_no              : num  6.25e+11 5.11e+11 2.85e+11 9.26e+11 9.41e+11 ...
 $ checking_balance    : Factor w/ 4 levels "< 0 USD","> 200 USD",..: 1 3 1 3 4 1 1 1 3 1 ...
 $ months_loan_duration: int  12 36 11 15 10 14 24 18 24 30 ...
 $ credit_history      : Factor w/ 5 levels "critical","good",..: 2 2 1 2 2 2 1 1 1 2 ...
 $ purpose             : Factor w/ 5 levels "business","car",..: 2 2 2 5 4 2 2 4 2 4 ...
 $ amount..USD.        : int  1274 12389 3939 1308 1924 3973 6615 2124 11938 2406 ...
 $ savings_balance     : Factor w/ 5 levels "< 100 USD","> 1000 USD",..: 1 5 1 1 1 1 1 1 1 1 ...
 $ employment_duration : Factor w/ 5 levels "< 1 year","> 7 years",..: 1 3 3 2 3 5 5 3 3 4 ...
 $ percent_of_income   : Factor w/ 4 levels "1","2","3","4": 3 1 1 4 1 1 2 4 2 4 ...
 $ years_at_residence  : Factor w/ 4 levels "1","2","3","4": 1 4 2 4 4 4 NA 4 3 NA ...
 $ age                 : int  37 37 40 38 38 22 75 24 39 23 ...
 $ other_credit        : Factor w/ 3 levels "bank","none",..: 2 2 2 2 2 2 2 2 2 2 ...
 $ housing             : Factor w/ 3 levels "other","own",..: 2 1 2 2 2 1 1 3 2 3 ...
 $ existing_loans_count: int  1 1 2 2 1 1 2 2 2 1 ...
 $ job                 : Factor w/ 4 levels "management","skilled",..: 4 2 4 4 2 2 1 2 1 2 ...
 $ dependants          : Factor w/ 2 levels "1","2": 1 1 2 1 1 1 1 1 2 1 ...
 $ phone               : Factor w/ 2 levels "no","yes": 1 2 1 1 2 1 2 1 2 1 ...
 $ default             : Factor w/ 0 levels: NA NA NA NA NA NA NA NA NA NA ...
```

Here, I changed all flag variables like character Variables into factor variables with given code.

```
> sapply(credrisk, function(x) sum(is.na(x)))
          ï..cust_id               acc_no     checking_balance months_loan_duration       credit_history
                   0                    0                    0                    0                    0
             purpose         amount..USD.      savings_balance  employment_duration    percent_of_income
                   0                    0                    0                    0                    0
  years_at_residence                  age         other_credit              housing existing_loans_count
                1320                    0                    0                    0                    0
                 job           dependants                phone              default
                   0                    0                    0                    0
>
```

With the help of sapply function, finding na values in each column. Only years_at_residence contains 1320 na values.

REPLACING NA AND CHECKING TARGET PROPORTION

```
> #Replacing the missing values in year_at_residence column by mean
> credrisk$years_at_residence[is.na(credrisk$years_at_residence)] <-
+   mean(credrisk$years_at_residence, na.rm = TRUE)
```

With this code, na values are replaced by column mean

```
> sapply(credrisk, function(x) sum(is.na(x)))
```

i..cust_id	acc_no	checking_balance	months_loan_duration	credit_history
0	0	0	0	0
purpose	amount..USD.	savings_balance	employment_duration	percent_of_income
0	0	0	0	0
years_at_residence	age	other_credit	housing	existing_loans_count
0	0	0	0	0
job	dependants	phone	default	
0	0	0	0	

```
> count <- table(credrisk$default)
> prop.table(count)

 no yes
0.7 0.3
```

Data has perfect target variable proportion

DATA PREPARATION

```
> ##splitting data set into train and test
> library(caret)
> set.seed(123)
> train.index <- createDataPartition(credrisk$default,p=0.7,list = FALSE)
> train <- credrisk[train.index,]
> test <- credrisk[-train.index,]
> count2 <- table(train$default)
> prop.table(count2)

 no yes
0.7 0.3
> count3 <- table(test$default)
> prop.table(count3)

 no yes
0.7 0.3
```

With the help of library caret, splitting the data into training and testing data.

Checking class proportion in training data.

Checking class proportion in testing data.

ITERATION 1

Fitting the logistic regression model of train data with the help of glm function.

Default is the response variable and rest of the variables are predictors.

Here, family taken in binomial.

```
> #Iteration 1
> model1 <- glm(train$default ~ ., data = train, family = binomial)
> summary(model1)

Call:
glm(formula = train$default ~ ., family = binomial, data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9244  -0.7508  -0.4071   0.7863   2.6217

Coefficients:
                              Estimate Std. Error z value Pr(>|z|)
(Intercept)                 -1.420e+00  4.401e-01  -3.226 0.001255 **
ï..cust_id                   1.246e-05  2.080e-04   0.060 0.952210
acc_no                      -1.248e-10  2.080e-09  -0.060 0.952161
checking_balance> 200 USD   -7.529e-01  1.736e-01  -4.336 1.45e-05 ***
checking_balance1 - 200 USD -3.806e-01  1.005e-01  -3.787 0.000152 ***
checking_balanceunknown     -1.824e+00  1.110e-01 -16.432  < 2e-16 ***
months_loan_duration         2.966e-02  4.408e-03   6.728 1.72e-11 ***
credit_historygood           9.157e-01  1.253e-01   7.308 2.71e-13 ***
credit_historyperfect        1.440e+00  2.036e-01   7.074 1.51e-12 ***
credit_historypoor           6.568e-01  1.596e-01   4.116 3.85e-05 ***
credit_historyvery good      1.366e+00  2.050e-01   6.665 2.64e-11 ***
purposecar                   1.231e-01  1.506e-01   0.817 0.413666
purposecar0                 -6.480e-01  3.766e-01  -1.721 0.085276 .
purposeeducation             5.829e-01  2.113e-01   2.758 0.005811 **
purposefurniture/appliances -2.508e-01  1.473e-01  -1.703 0.088557 .
purposerenovations           4.742e-01  2.858e-01   1.659 0.097087 .
amount..USD.                 8.734e-05  2.038e-05   4.286 1.82e-05 ***
```

ITERATION-1SUMMARY

```
savings_balance> 1000 USD          -1.121e+00  2.519e-01  -4.452 8.52e-06 ***
savings_balance100 - 500 USD       -2.784e-01  1.376e-01  -2.024 0.042948 *
savings_balance500 - 1000 USD      -3.611e-01  1.901e-01  -1.900 0.057493 .
savings_balanceunknown             -9.071e-01  1.221e-01  -7.430 1.08e-13 ***
employment_duration> 7 years       -4.878e-01  1.371e-01  -3.558 0.000373 ***
employment_duration1 - 4 years     -4.963e-01  1.161e-01  -4.274 1.92e-05 ***
employment_duration4 - 7 years     -1.205e+00  1.446e-01  -8.336  < 2e-16 ***
employment_durationunemployed      -2.188e-01  2.068e-01  -1.058 0.290099
percent_of_income2                  2.364e-01  1.443e-01   1.638 0.101481
percent_of_income3                  4.841e-01  1.560e-01   3.103 0.001915 **
percent_of_income4                  8.136e-01  1.394e-01   5.835 5.38e-09 ***
years_at_residence2                 8.697e-01  1.641e-01   5.298 1.17e-07 ***
years_at_residence2.84871794871795  4.464e-01  1.672e-01   2.669 0.007598 **
years_at_residence3                 7.203e-01  1.801e-01   4.000 6.35e-05 ***
years_at_residence4                 4.744e-01  1.622e-01   2.924 0.003452 **
age                                -1.329e-02  4.296e-03  -3.094 0.001974 **
other_creditnone                   -5.026e-01  1.121e-01  -4.485 7.28e-06 ***
other_creditstore                  -9.595e-03  1.978e-01  -0.049 0.961302
housingown                         -2.474e-01  1.363e-01  -1.816 0.069438 .
housingrent                         1.920e-01  1.584e-01   1.212 0.225452
existing_loans_count                2.612e-01  9.167e-02   2.849 0.004390 **
jobskilled                          1.397e-01  1.349e-01   1.036 0.300269
jobunemployed                      -9.931e-02  3.089e-01  -0.321 0.747844
jobunskilled                        5.109e-02  1.621e-01   0.315 0.752615
dependants2                        -1.512e-02  1.134e-01  -0.133 0.893890
phoneyes                           -1.937e-01  9.508e-02  -2.037 0.041603 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 5131.3  on 4199  degrees of freedom
Residual deviance: 3950.4  on 4157  degrees of freedom
AIC: 4036.4

Number of Fisher Scoring iterations: 11
```

At 5% level of significance, checking_balance, months_loan_duration, credit_history, amount..USD., employment_duration,years_at_residence, age, other_credit, existing_loans_count, dependants, phone are significant.

The model explains the data well since the value of residual deviance is lesser than null deviance.

AIC of the model is 4036.4

ITERATION 2

In the second iteration all the variables which Are significant are taken.

The model explains the data well since the value of residual deviance is lesser than null deviance.

```
> ##Iteration 2 with only significant variable
> model2 <- glm(default ~ .-ï..cust_id-acc_no-percent_of_income-purpose-housing-job,
+               data = train, family = binomial)
> summary(model2)

Call:
glm(formula = default ~ . - ï..cust_id - acc_no - percent_of_income -
    purpose - housing - job, family = binomial, data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9834  -0.7677  -0.4274   0.8429   2.6348

Coefficients:
                                     Estimate Std. Error z value Pr(>|z|)
(Intercept)                        -1.213e+00  2.977e-01  -4.074 4.61e-05 ***
checking_balance> 200 USD          -9.127e-01  1.684e-01  -5.419 6.00e-08 ***
checking_balance1 - 200 USD        -4.327e-01  9.729e-02  -4.447 8.71e-06 ***
checking_balanceunknown            -1.830e+00  1.082e-01 -16.903  < 2e-16 ***
months_loan_duration                3.633e-02  4.074e-03   8.919  < 2e-16 ***
credit_historygood                  8.883e-01  1.219e-01   7.289 3.13e-13 ***
credit_historyperfect               1.451e+00  1.950e-01   7.442 9.89e-14 ***
credit_historypoor                  6.363e-01  1.550e-01   4.105 4.04e-05 ***
credit_historyvery good             1.456e+00  2.002e-01   7.273 3.52e-13 ***
amount..USD.                        3.660e-05  1.762e-05   2.077 0.037756 *
savings_balance> 1000 USD          -1.004e+00  2.451e-01  -4.096 4.21e-05 ***
savings_balance100 - 500 USD       -2.105e-01  1.338e-01  -1.573 0.115770
savings_balance500 - 1000 USD      -4.659e-01  1.872e-01  -2.488 0.012841 *
savings_balanceunknown             -8.053e-01  1.179e-01  -6.830 8.49e-12 ***
employment_duration> 7 years       -4.896e-01  1.308e-01  -3.742 0.000183 ***
employment_duration1 - 4 years     -5.243e-01  1.125e-01  -4.660 3.17e-06 ***
employment_duration4 - 7 years     -1.183e+00  1.399e-01  -8.459  < 2e-16 ***
employment_durationunemployed      -3.170e-01  1.814e-01  -1.748 0.080420 .
years_at_residence2                 9.040e-01  1.598e-01   5.658 1.53e-08 ***
years_at_residence2.84871794871795  4.285e-01  1.626e-01   2.636 0.008386 **
years_at_residence3                 7.003e-01  1.745e-01   4.012 6.02e-05 ***
years_at_residence4                 6.216e-01  1.546e-01   4.022 5.78e-05 ***
```

```
age                         -9.993e-03  3.974e-03  -2.515 0.011919 *
other_creditnone            -4.239e-01  1.090e-01  -3.889 0.000101 ***
other_creditstore           -3.369e-02  1.931e-01  -0.175 0.861451
existing_loans_count         2.602e-01  8.785e-02   2.962 0.003052 **
dependants2                 -8.262e-03  1.092e-01  -0.076 0.939673
phoneyes                    -1.575e-01  8.565e-02  -1.839 0.065900 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 5131.3  on 4199  degrees of freedom
Residual deviance: 4062.3  on 4172  degrees of freedom
AIC: 4118.3

Number of Fisher Scoring iterations: 5
```

AIC value for iteration 2 is higher than iteration 1.

```
> vif(model2)
       checking_balance> 200 USD        checking_balance1 - 200 USD        checking_balanceunknown
                          1.1434                             1.4203                         1.3880
              months_loan_duration                 credit_historygood         credit_historyperfect
                          1.6713                             2.4311                         1.2319
               credit_historypoor             credit_historyvery good                 amount..USD.
                          1.4232                             1.5254                         1.7615
      savings_balance> 1000 USD         savings_balance100 - 500 USD  savings_balance500 - 1000 USD
                          1.0400                             1.1301                         1.0606
          savings_balanceunknown          employment_duration> 7 years  employment_duration1 - 4 years
                          1.1060                             2.0809                         1.9476
  employment_duration4 - 7 years       employment_durationunemployed            years_at_residence2
                          1.5789                             1.3627                         3.2014
years_at_residence2.84871794871795            years_at_residence3            years_at_residence4
                          2.8119                             2.2635                         3.5807
                             age                  other_creditnone               other_creditstore
                          1.3002                             1.3599                         1.3143
            existing_loans_count                        dependants2                      phoneyes
                          1.6791                             1.0672                         1.1606
```

If vif > 2 then, there is presence of multicollinearity.

Credit_history, years_at_residence and employement Duration shows multicollinearity. So these variables Will be dropped in nect iteration.

ITERATION 3

In iteration 3, most logical and intuitive variables were taken.

Those variables which don't make sense were Dropped. For eg phone.

At first credit history was showing multicollinearity but now its not showing Multicollinearity.

The model explains the data well since the value of residual deviance is lesser than null deviance.

AIC of the model is 4231.4

Also no sign of multicollinearity.

```
> model3 <- glm(default ~ checking_balance+savings_balance+months_loan_duration
+               +credit_history+other_credit, data = train, family = binomial)
> summary(model3)

Call:
glm(formula = default ~ checking_balance + savings_balance + 
    months_loan_duration + credit_history + other_credit, family = binomial, 
    data = train)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-1.7751  -0.7914  -0.4494   0.9086   2.5456  

Coefficients:
                              Estimate Std. Error z value Pr(>|z|)    
(Intercept)                  -0.957610   0.148767  -6.437 1.22e-10 ***
checking_balance> 200 USD    -0.920839   0.162776  -5.657 1.54e-08 ***
checking_balance1 - 200 USD  -0.369557   0.093632  -3.947 7.92e-05 ***
checking_balanceunknown      -1.747704   0.104284 -16.759  < 2e-16 ***
savings_balance> 1000 USD    -1.006920   0.240298  -4.190 2.79e-05 ***
savings_balance100 - 500 USD -0.215469   0.128540  -1.676 0.093684 .  
savings_balance500 - 1000 USD -0.503920  0.180649  -2.789 0.005279 ** 
savings_balanceunknown       -0.842366   0.114322  -7.368 1.73e-13 ***
months_loan_duration          0.035492   0.003178  11.169  < 2e-16 ***
credit_historygood            0.707793   0.095896   7.381 1.57e-13 ***
credit_historyperfect         1.494385   0.187794   7.958 1.75e-15 ***
credit_historypoor            0.538914   0.150210   3.588 0.000334 ***
credit_historyvery good       1.180112   0.182818   6.455 1.08e-10 ***
other_creditnone             -0.397668   0.106927  -3.719 0.000200 ***
other_creditstore             0.077230   0.186341   0.414 0.678541    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 5131.3  on 4199  degrees of freedom
Residual deviance: 4201.4  on 4185  degrees of freedom
AIC: 4231.4

Number of Fisher Scoring iterations: 5
```

```
> lrtest(model3)
Likelihood ratio test

Model 1: default ~ checking_balance + savings_balance + months_loan_duration +
    credit_history + other_credit
Model 2: default ~ 1
  #Df  LogLik  Df  Chisq Pr(>Chisq)
1  15 -2100.7
2   1 -2565.6 -14 929.85  < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
> pR2(model3)
fitting null model for pseudo-r2
          llh       llhNull            G2      McFadden          r2ML          r2CU
-2100.7041590 -2565.6300686   929.8518193     0.1812132     0.1985986     0.2815884
```

```
> hoslem.test(train$default, fitted(model3))

        Hosmer and Lemeshow goodness of fit (GOF) test

data:  train$default, fitted(model3)
X-squared = 4200, df = 8, p-value < 2.2e-16
```

```
> library(InformationValue)
> somersD(train$default, fitted(model3))
[1] 0.564642
```

TESTS

1) Likelihood ratio test

Since pcal < 0.05, we reject Ho and conclude that model is a good fit.

2) McFadden's pseudo R2 test

The value of McFadden's pseudo is almost R2>0.2 indicating that the model is a good fit

3) Hosmer Lemeshow test

Since p value< 0.05, we reject Ho and conclude that the model is a good fit.

4) Somer's D test

Somer's D value suggests that the model has a decent predictive ability.

PREDICTION FOR TRAIN DATA AND CONFUSION MATRIX

Accuracy of 0.746 is pretty good.

```
> fit <- fitted(model3, type = "response")
> head(fit, 3)
        1         2         3 
0.4448297 0.3585411 0.2759046 
> fit_class <- ifelse(fit>0.5, 1, 0)
> head(fit_class, 3)
1 2 3 
0 0 0 
```

```
> confusionMatrix(data = as.factor(fit_class), reference = train$default)
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 2652  779
         1  288  481
                                          
               Accuracy : 0.746           
                 95% CI : (0.7325, 0.7591)
    No Information Rate : 0.7             
    P-Value [Acc > NIR] : 2.301e-11       
                                          
                  Kappa : 0.3193          
                                          
 Mcnemar's Test P-Value : < 2.2e-16       
                                          
            Sensitivity : 0.9020          
            Specificity : 0.3817          
         Pos Pred Value : 0.7730          
         Neg Pred Value : 0.6255          
             Prevalence : 0.7000          
         Detection Rate : 0.6314          
   Detection Prevalence : 0.8169          
      Balanced Accuracy : 0.6419          
                                          
       'Positive' Class : 0               
```

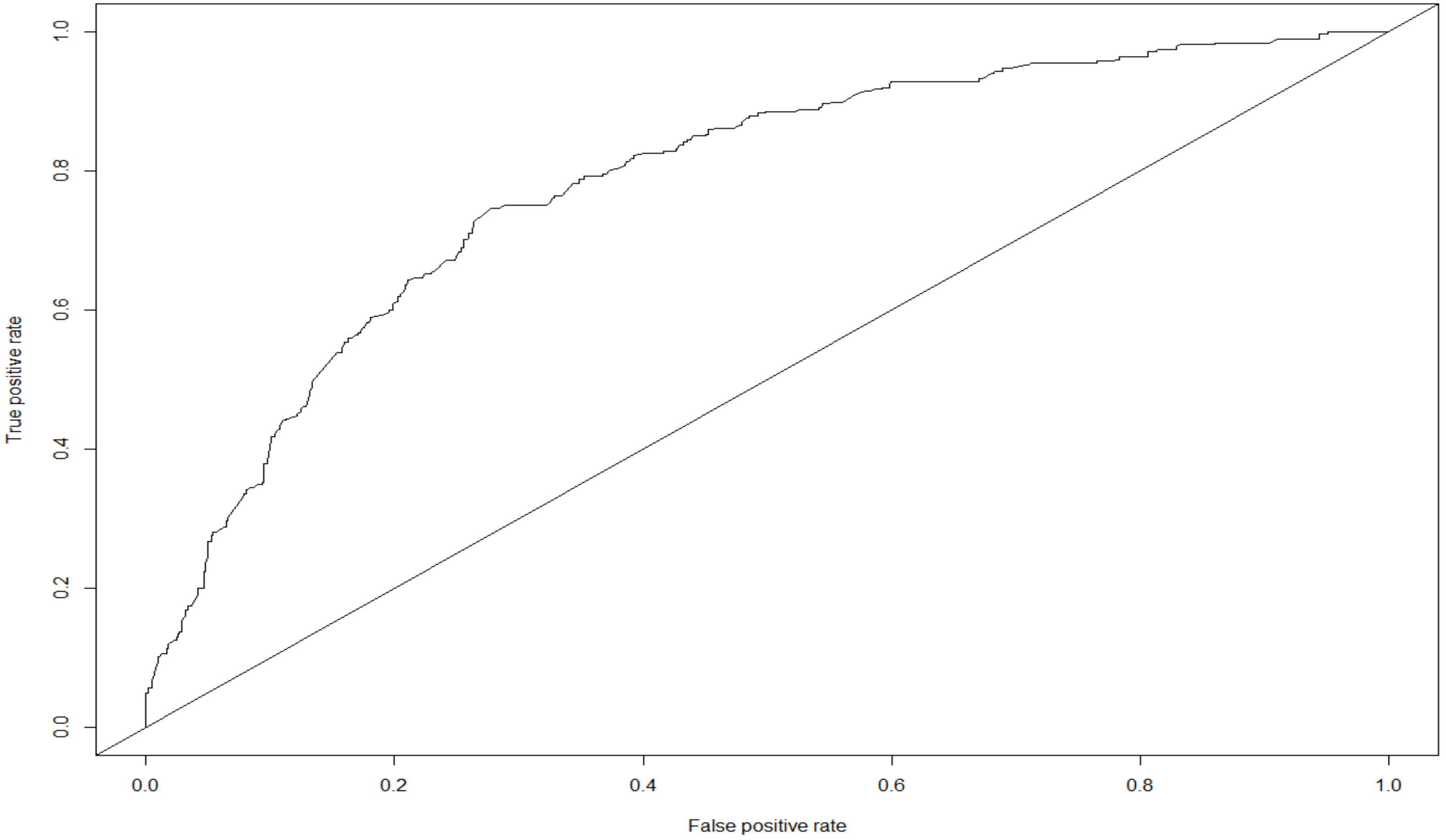
PREDICTION FOR TEST DATA AND CONFUSION MATRIX

```
> predicted <- predict(model3, test, type = "response")
> head(predicted, 3)
         5         11         12 
0.11503916 0.48192223 0.07841146 
> predicted_class <- ifelse(predicted > 0.5, 1, 0)
> head(predicted_class, 3)
 5 11 12 
 0  0  0 
> confusionMatrix(data = as.factor(predicted_class), reference = test$default)
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1146  337
         1  114  203
                                          
               Accuracy : 0.7494          
                 95% CI : (0.7288, 0.7693)
    No Information Rate : 0.7             
    P-Value [Acc > NIR] : 1.854e-06       
                                          
                  Kappa : 0.3236          
                                          
 Mcnemar's Test P-Value : < 2.2e-16       
                                          
            Sensitivity : 0.9095          
            Specificity : 0.3759          
         Pos Pred Value : 0.7728          
         Neg Pred Value : 0.6404          
             Prevalence : 0.7000          
         Detection Rate : 0.6367          
   Detection Prevalence : 0.8239          
      Balanced Accuracy : 0.6427          
                                          
       'Positive' Class : 0               
```

Again, accuracy of 74.94% is pretty good.

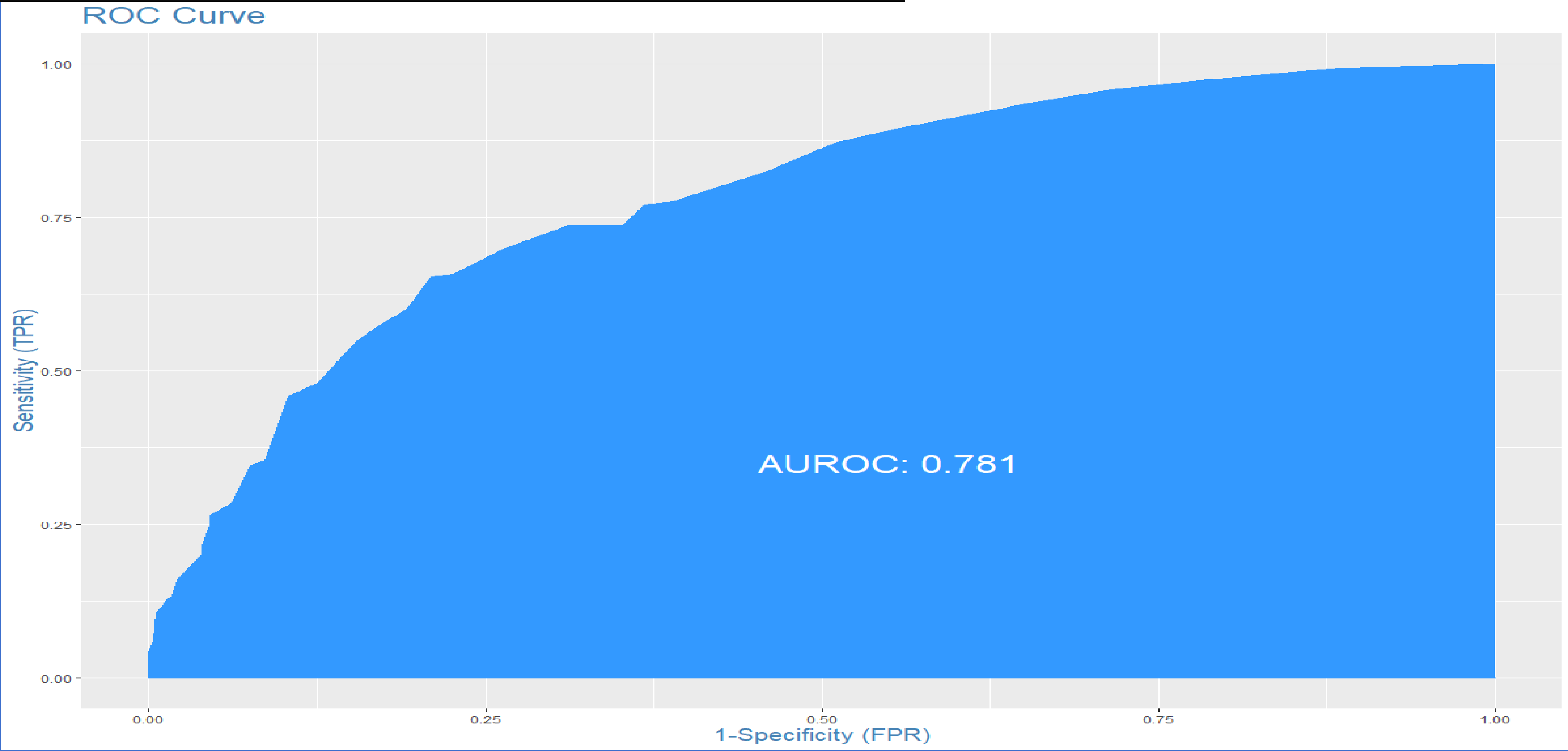
ROC CURVE OF TRAIN DATA

```
> ROC_predict_train <- prediction(fitted(model3), train$default)
> ROC_perf_train <- performance(ROC_predict_train, 'tpr', 'fpr')
> plot(ROC_perf_train)
> abline(0, 1)
> auc <- performance(ROC_predict_train, "auc")
> auc@y.values
[[1]]
[1] 0.7837261
```



```
> plotROC(test$default, predicted)
```

ROC CURVE OF TEST DATA

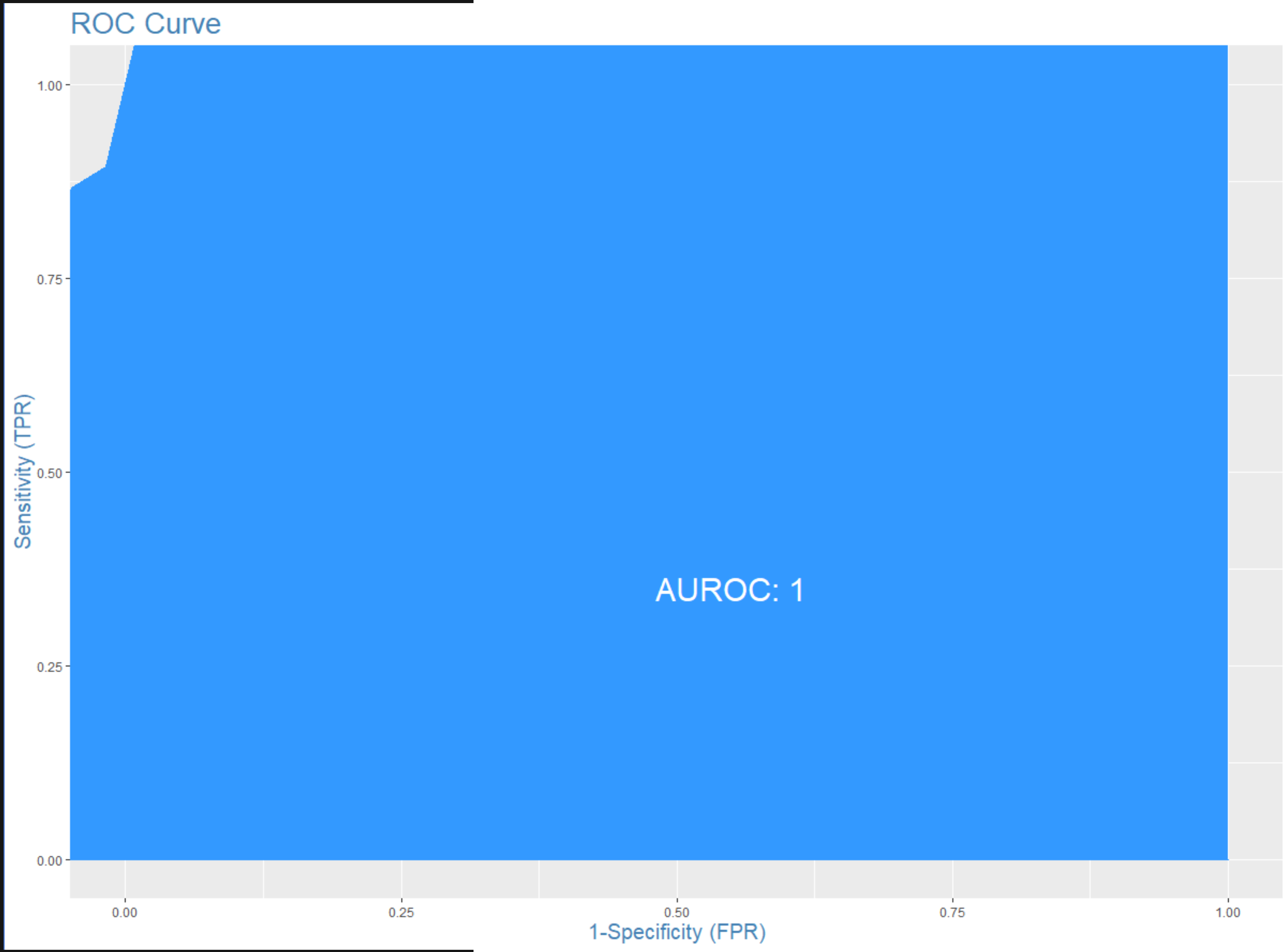


K FOLD CROSS VALIDATION

```
> train_control <- trainControl(method = "cv", number = 5)
> model4 <- train(default ~ checking_balance+savings_balance+months_loan_duration
+                 +credit_history+other_credit, data = train, method = "glm",
+                 family = binomial, trControl = train_control)
> model4
Generalized Linear Model

4200 samples
   5 predictor
   2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 3360, 3360, 3360, 3360, 3360
Resampling results:

  Accuracy   Kappa
  0.7504762  0.3354406

> #Iteration 4: Predicting values of 'Y' for Test data
> predicted_KCV <- predict(model4, test, 'prob')
> head(predicted_KCV)
           0          1
5  0.8849608 0.11503916
11 0.5180778 0.48192223
12 0.9215885 0.07841146
20 0.9433023 0.05669770
32 0.7903490 0.20965096
39 0.8663437 0.13365627
> predicted_class <- predict(model4, test)
> head(predicted_class)
[1] 0 0 0 0 0 0
Levels: 0 1
```

CONFUSION MATRIX AND ROC CURVE OF TEST DATA

```
> confusionMatrix(data = as.factor(predicted_class), reference = test$default)
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1146  337
         1  114  203

               Accuracy : 0.7494
                 95% CI : (0.7288, 0.7693)
    No Information Rate : 0.7
    P-Value [Acc > NIR] : 1.854e-06

                  Kappa : 0.3236

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9095
            Specificity : 0.3759
         Pos Pred Value : 0.7728
         Neg Pred Value : 0.6404
             Prevalence : 0.7000
         Detection Rate : 0.6367
   Detection Prevalence : 0.8239
      Balanced Accuracy : 0.6427

       'Positive' Class : 0

> library(InformationValue)
> plotROC(test$default, predicted_KCV)
```



BASIS THE INFORMATION PROVIDED, LIST DOWN AND BRIEFLY EXPLAIN THE STEPS YOU WOULD FOLLOW AS PART OF THE END-TO-END PROCESS TO DEVELOP THE INTERNAL CREDIT SCORING MODEL FOR THE RETAIL PORTFOLIO.

1. Data collection process : Collecting raw data for model building.
2. Analysing, cleaning and aggregating data: As data is not clean, we have to clean the data, remove outliers and fill na values before Proceeding to fit the model.

3. Splitting the data into training and testing.

3. Initial model running : Fitting the model from cleaned train data.
4. Variable selection : Selecting significant variables for next iteration.
5. Checking for the signs of multicollinearity.
6. Performance testing : Checking predictive power of model.
7. Statistical testing : performing various statistical tests. For eg goodness of fit tests, multicollinearity tests, etc
8. Stress testing and sensitivity analysis
9. Comparing actual vs predicted values with the help of graph.
10. Implementing the model.

ANALYSIS AND CLEANSING OF RAW DATA IS CRITICAL TO DEVELOP AN ROBUST MODEL. EXPLAIN THE DATA CLEANSING, EXPLORATORY ANALYSIS AND TRANSFORMATION ACTIVITIES YOU WOULD PERFORM OUT AS PART OF THE MODEL DEVELOPMENT EXERCISE.

1. I would first set all flag variables into appropriate class. For eg, char into factor etc,
2. Then, fill missing values by mean or get rid of the rows which contains missing values
3. Then, check for outliers. If any outliers are present then I'll get rid of them.
4. Checking summary of the data is also important.
5. If there is need of changing say date format then I will do it.
6. Merging and aggregating data.
7. Separating two parts of single columns if necessary.
8. Renaming the columns
9. Selecting data according to given condition.

LIST DOWN AND EXPLAIN VARIOUS MODELLING METHODOLOGIES / FRAMEWORKS THAT COULD BE ADOPTED FOR DEVELOPING THE CREDIT SCORING MODEL

1. **Simple linear regression:** A statistical method to mention the relationship between two variables which are continuous. Can be used if there are only two variables in question.
2. **Multiple linear regression:** A statistical method to mention the relationship between more than two variables which are continuous. Here, more than one variables can be used to predict the target.
3. **Decision tree regression:** A tree-like structure is used in these decision tree models to build classification or regression-related algorithms. Here the decision tree is incrementally developed by subsetting the given dataset into smaller chunks. Each branch of the decision tree could be a possible outcome.
4. **Logistic regression :** Logistic regression is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable in which there are only two possible outcomes.

EXPLAIN THE ACTIVITIES & CONTROL-CHECKS YOU WOULD CARRY TO ENSURE THAT EXPLANATORY VARIABLES SELECTED AS PART OF THE VARIABLE SELECTION PROCESS ARE OPTIMAL.

1. I will check for significant variable. If any variable has p value more than 5%, then I will get rid of that variable.
2. If any variable shows multicollinearity it is sensible to remove that variable.
3. If some variables that are both are significant and don't show multicollinearity but are not intuitively current will be removed.
4. It is necessary to check if selected variable don't have any missing value or misplaced value or outliers.

SUGGEST POSSIBLE USES / APPLICATIONS OF A CREDIT SCORING MODEL ACROSS VARIOUS DEPARTMENTS IN A BANK

1) Banks credit exposures typically cut across geographical locations and product lines. The use of credit risk models offers banks a framework for examining this risk in a timely manner. These properties of models may contribute to an improvement in a bank's overall ability to identify, measure and manage risk.
2) Credit risk models may provide estimates of credit risk which reflect individual portfolio composition. Hence, they provide better reflection of concentration risk compared to non-portfolio approaches.
3) This modelling methodology holds out the possibility of providing a more responsive and informative tool for risk management
4) The models offer the incentive to improve systems and data collection efforts
5) Also more accurate risk and performance-based pricing, which may contribute to a more transparent decision-making process.


18476(1)
18437(1)
4th
5th
18594 (1)
18589 (4)
18590 (1)
18567 (2)
18567 (9)
18560 (5)
18514 (1)
18390 (1)
18440
18357
18590(2)
18591(1)
18585(2)
18588(1)
-60
-70
-111

THANK YOU