Project on Logistic Regression

Roll. No. - 404

Sannidhya
11-19-2021

Credit Risk Score Card development

AIM: To design and develop a robust internal credit risk models for analyse and quantify a potential obligor's credit risk.

In this case study we explored and analysed our data set containing 6000 records representing Default status of obligor. Our target variable is default, "1" means that the person is a defaulter and '0' means that the person is not a defaulter.

STEP 1 - Data collection

To create a model, we were provided with historical default data form our data collection team.

Data was extracted from *Assessment Case study input.csv* data.

STEP 2 - Exploratory Data Analysis

After extracting the data, we conducted following EDA:

- We used *head ()* function to return the first 10 rows of our data.
- We used *tail ()* function to return last 10 rows of our data.
- *Colnames ()* provided us with all the column names that we had in our data: -

```
> colnames(loan)
 [1] "cust_id"             "acc_no"              "checking_balance"
 [4] "months_loan_duration" "credit_history"      "purpose"
 [7] "amount..USD."        "savings_balance"     "employment_duration"
[10] "percent_of_income"   "years_at_residence"  "age"
[13] "other_credit"        "housing"             "existing_loans_count"
[16] "job"                 "dependants"          "phone"
[19] "default"
>
```

- We used *ncol ()* and *nrow ()* for determining
  Number of Rows = 6000
  Number of Columns = 19

- We used *str ()* for Compactly displaying the internal structure of our dataset.

```
> str(loan)
'data.frame':   6000 obs. of  19 variables:
 $ cust_id             : int  6252029 5110070 2846491 9264318 9412980 6111903 161301
4 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" ...
```

STEP 3 - CLEANING DATA

While doing Exploratory Data Analysis, we noticed that our data was inconsistent which needed to be corrected before using the data for any analysis.

Converting “car0” to car.

```
loan$purpose[loan$purpose == "car0"] = "car"
View(loan)
```

Finding Missing Values

After applying *sapply ()* and *is.na ()* function we noticed that there were 1320 missing values in the years at residence column. We Replaced them with mean of that column.

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

All the Integer variables which represents factor values we converted them into factor variables.

```
> str(loan)
'data.frame':   6000 obs. of  19 variables:
 $ 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   : int  3 1 1 4 1 1 2 4 2 4 ...
 $ years_at_residence  : num  1 4 2 4 4 ...
 $ 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: Factor w/ 4 levels "1","2","3","4": 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/ 2 levels "no","yes": 2 2 1 1 1 1 1 2 2 2 ...
```

STEP 4 - DATA PREPERATIONS

Target Variable Proportion

```
> # Target Variable Proportion
> tvp=table(loan$default)
> prop.table(tvp)

 no yes
0.7 0.3
```

This shows us that 30% proportion of population actually is marked default.

Before splitting the data, we changed our default columns values from (no, yes) to (0,1).

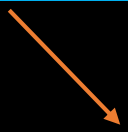
```
loan$default=factor(loan$default,levels = c("no","yes"),labels = c(0,1))
loan
```

Splitting of Data

In order to check how well our variables selected are actually performing.

Undertaking a stratified sampling and not simple sampling as our 0 and 1 are not in 50% proportion, proportion of our training and testing dataset is the same as our entire population dataset.

```
## Splitting of Data
library(caret)
set.seed(100)
s=createDataPartition(loan$default,p=0.7,list = FALSE)
loan_tr=loan[s,]
loan_test=loan[-s,]
```

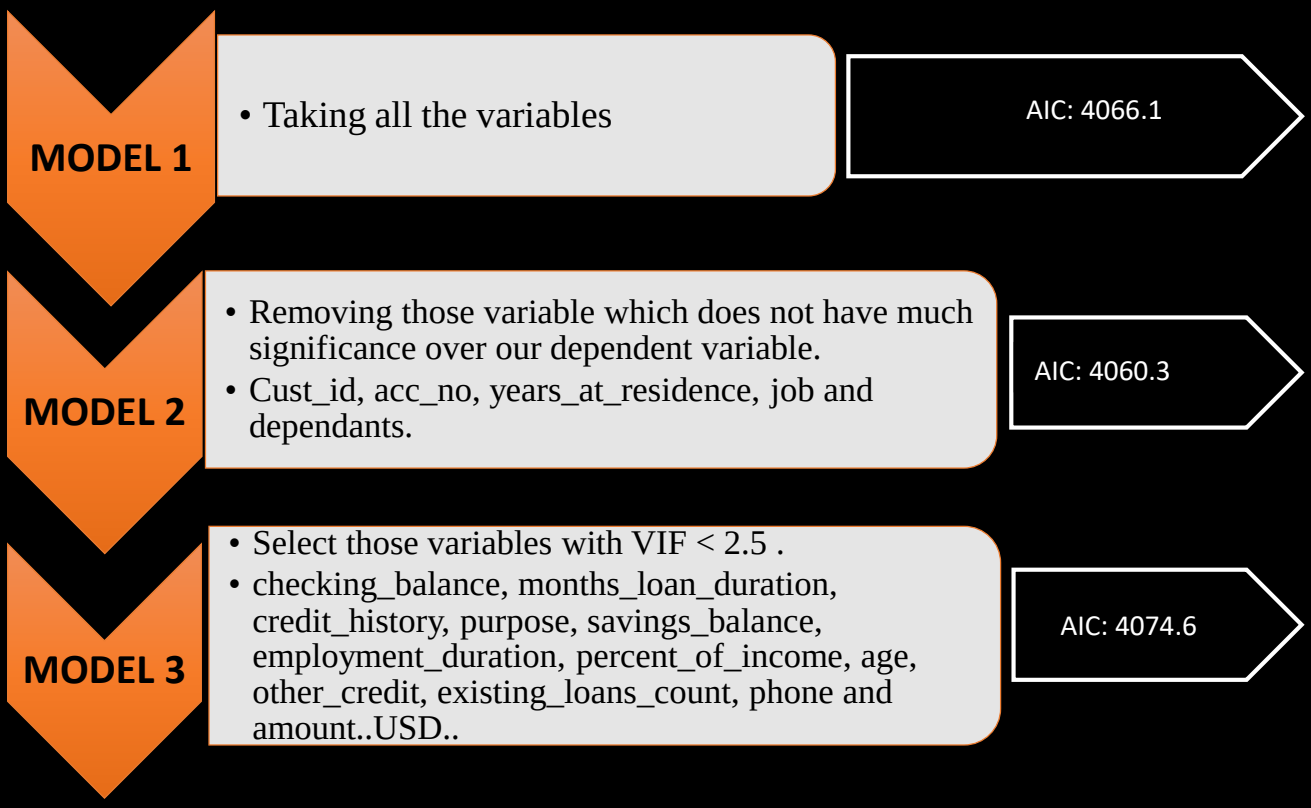


```
> tvp1=table(loan_tr$default)
> prop.table(tvp1)

  0   1
0.7 0.3
```

```
> tvp2=table(loan_test$default)
> prop.table(tvp2)

  0   1
0.7 0.3
```

STEP 5 – Modeling

ITERATIONS



ITERATION 1

We took all the variables and checked their significance.

AIC: 4066.1

Null deviance: 5131.3

Residual deviance: 3988.1

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

ITERATIONS 2

We removed all those variables which were insignificant.

We also used 3 more tests for further strengthening our models, for example:

AIC: 4060.3

Null deviance: 5131.3

Residual deviance: 3996.3

Information test

```
> IV(loan_tr$cust_id,loan_tr$default)
[1] 0
attr(,"howgood")
[1] "Not Predictive"
```

Variable Importance

```
> library(caret)
> varImp(model1)
                                Overall
cust_id                      0.04574248
acc_no                       0.04583329
checking_balance> 200 USD    4.95134216
checking_balance1 - 200 USD  4.22678408
checking_balanceunknown     16.42426304
```

Wald test

```
> regTermTest(model1,"cust_id")
Wald test for cust_id
 in glm(formula = default ~ ., family = binomial, data = loan_tr)
F =  0.002092374  on  1  and  4161  df: p= 0.96352
```

Housing variable was also somewhat not significant (0.06308) but we didn't remove it because it was simply increasing our AIC for our 2nd iteration.

Information test

```
> IV(loan_tr$housing,loan_tr$default)
[1] 0.07397974
attr(,"howgood")
[1] "Somewhat Predictive"
```

Wald test

```
> regTermTest(model1,"housing")
Wald test for housing
 in glm(formula = default ~ ., family = binomial, data = loan_tr)
F =  8.288874  on  2  and  4161  df: p= 0.00025547
```

ITERATION 3

(Assumption of VIF Threshold value of 2.5.)

For model3 we removed all the variables that were having VIF (Variation Inflation Factor) of level 2.5.

We removed Housing as it was above 2.5.

Hence there is no presence of Multicollinearity.

AIC: 4074.6

Null deviance: 5131.3

Residual deviance: 4014.6

STEP 6 – Testing

Goodness of fit test

Likelihood ratio test

```
> lrtest(model1,model2, model3)
Likelihood ratio test

Model 1: default ~ cust_id + acc_no + checking_balance + months_loan_duration +
    credit_history + purpose + amount..USD. + savings_balance +
    employment_duration + percent_of_income + years_at_residence +
    age + other_credit + housing + existing_loans_count + job +
    dependants + phone
Model 2: default ~ (cust_id + acc_no + checking_balance + months_loan_duration +
    credit_history + purpose + amount..USD. + savings_balance +
    employment_duration + percent_of_income + years_at_residence +
    age + other_credit + housing + existing_loans_count + job +
    dependants + phone) - cust_id - acc_no - years_at_residence -
    job - dependants
Model 3: default ~ checking_balance + months_loan_duration + credit_history +
    purpose + savings_balance + employment_duration + percent_of_income +
    age + other_credit + existing_loans_count + phone + amount..USD.
  #Df  LogLik Df   Chisq Pr(>Chisq)
1  39 -1994.1
2  32 -1998.2 -7  8.2034   0.314997
3  30 -2007.3 -2 18.2475   0.000109 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Since p<0.05, we can reject Ho and conclude that model 3 is a good fit.

Pseudo R^2

```
> # Pseudo R2
> library(pscl)
> pR2(model3)
fitting null model for pseudo-r2
         llh       llhNull            G2      McFadden          r2ML          r2CU
-2007.2948726 -2565.6300686  1116.6703921     0.2176211     0.2334642     0.3310236
```

McFadden = 0.2176211, The value of McFadden pseudo R2 > 0.2, indicating that the model is a good fit.

Hosmer Lemeshow test

```
> # Hosmer Lemeshow test
> library(ResourceSelection)
> hoslem.test(loan_tr$default,fitted(model3))

        Hosmer and Lemeshow goodness of fit (GOF) test

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

Since p value < 0.05, we reject Ho and conclude that the model 3 is a good fit.

Somers' D

```
> # Somers' D
> library(InformationValue)
> somersD(loan_tr$default,fitted(model3))
[1] 0.6113454
```

Value of this index should be closer to 1 and farther from −1, so this model has a decent predictive ability.

STEP 7 – Prediction

Measuring Prediction Accuracy

Predicting values for train data

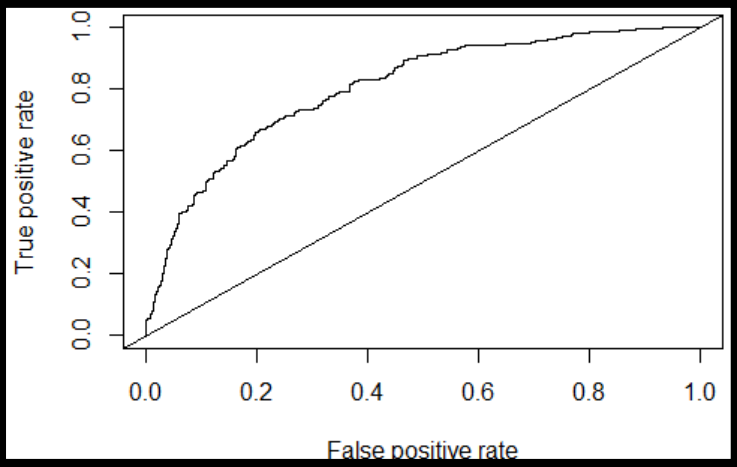
```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 2625  646
         1  315  614

               Accuracy : 0.7712
                 95% CI : (0.7582, 0.7838)
    No Information Rate : 0.7
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.411

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

            Sensitivity : 0.8929
            Specificity : 0.4873
         Pos Pred Value : 0.8025
         Neg Pred Value : 0.6609
             Prevalence : 0.7000
         Detection Rate : 0.6250
   Detection Prevalence : 0.7788
      Balanced Accuracy : 0.6901

       'Positive' Class : 0
```

Predicting values for test data

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1125  284
         1  135  256

               Accuracy : 0.7672
                 95% CI : (0.747, 0.7866)
    No Information Rate : 0.7
    P-Value [Acc > NIR] : 1.104e-10

                  Kappa : 0.3983

 Mcnemar's Test P-Value : 4.820e-13

            Sensitivity : 0.8929
            Specificity : 0.4741
         Pos Pred Value : 0.7984
         Neg Pred Value : 0.6547
             Prevalence : 0.7000
         Detection Rate : 0.6250
   Detection Prevalence : 0.7828
      Balanced Accuracy : 0.6835

       'Positive' Class : 0
```

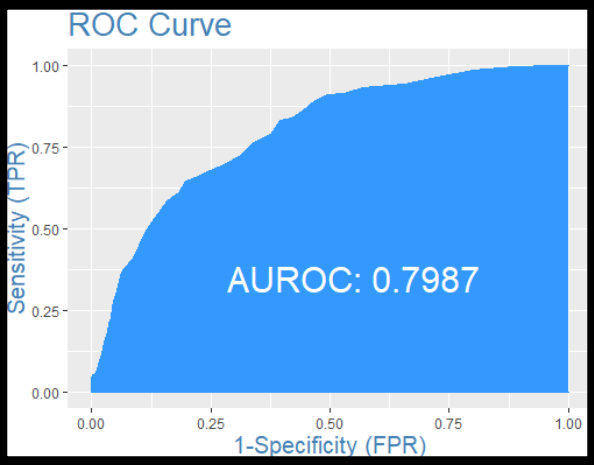
ROC curve for Train data

```
> library(ROCR)
> p_train=prediction(fitted(model3),loan_tr$default)
> perf_train=performance(p_train,'tpr','fpr')
> plot(perf_train)
> abline(0,1)
> auc=performance(p_train,"auc")
> auc@y.values
[[1]]
[1] 0.8056727
```



ROC curve for Test data

```
library(plotROC)
plotROC(loan_test$default,prediction2)
# 0.7987
```



AUROC> 0.80

This indicates that the model does a good job in discriminating between the two categories which comprise our target variable.

k fold cross validation

MODEL 4

- checking_balance, months_loan_duration, credit_history, purpose, savings_balance, employment_duration, percent_of_income, age, other_credit, existing_loans_count, phone and amount..USD..

AIC: 4074.6

Accuracy = 0.7635714

Kappa = 0.3923305

Null Deviance: 5131.3

Residual Deviance: 4014.6

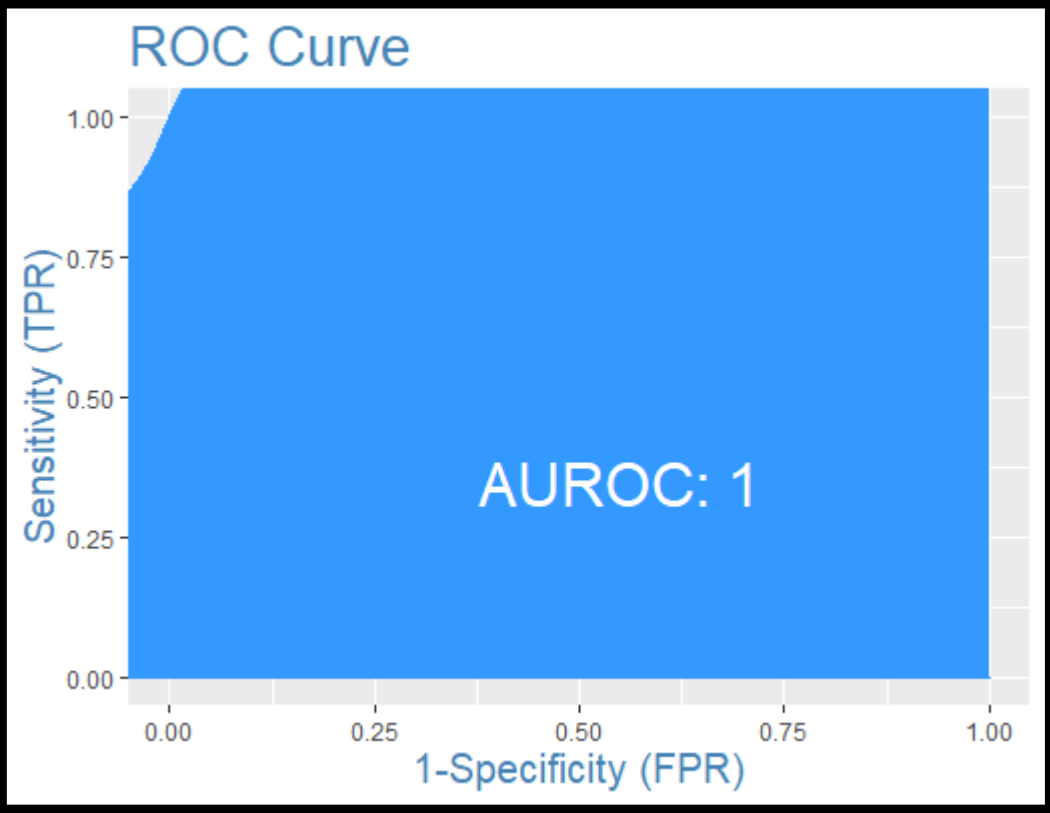
```
> model4$resample
   Accuracy     Kappa Resample
1 0.7654762 0.3912237    Fold1
2 0.7845238 0.4501823    Fold2
3 0.7738095 0.4150246    Fold3
4 0.7452381 0.3443627    Fold4
5 0.7642857 0.3903941    Fold5
```

Predicting values for Test data

```
> pred_kfcv=predict(model4,loan_test,'prob')
> head(pred_kfcv,10)
           0          1
14 0.9421411 0.05785892
15 0.5940021 0.40599790
18 0.8914522 0.10854780
19 0.8135596 0.18644041
21 0.8095265 0.19047348
22 0.6933535 0.30664646
24 0.9841830 0.01581699
32 0.7256352 0.27436477
34 0.6126527 0.38734727
35 0.2515348 0.74846517
```

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1125  284
         1  135  256

               Accuracy : 0.7672
                 95% CI : (0.747, 0.7866)
    No Information Rate : 0.7
    P-Value [Acc > NIR] : 1.104e-10

                  Kappa : 0.3983

 Mcnemar's Test P-Value : 4.820e-13

            Sensitivity : 0.8929
            Specificity : 0.4741
         Pos Pred Value : 0.7984
         Neg Pred Value : 0.6547
             Prevalence : 0.7000
         Detection Rate : 0.6250
   Detection Prevalence : 0.7828
      Balanced Accuracy : 0.6835

       'Positive' Class : 0
```

ROC Curve



Q1) 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.

There are 7 Major steps for developing a successful model. Those are:

STEP 1 - Data collection

Obtaining raw data from sources

STEP 2 - Exploratory Data Analysis

Head(), tail(), dim(), nrow(), ncol() and *str()*

STEP 3 - CLEANING DATA

Replacing all missing values in years at resident column and converting all the car0 to car.

STEP 4 - DATA PREPERATIONS

Converting Integer variables, them into factor variables and then splitting into testing and train dataset.

STEP 5 – Modeling

Building a model with selecting appropriate variables.

STEP 6 – Testing

Conducting statistical testing on the model.

STEP 7 – Prediction

Prediction values and checking its accuracy.

Q2) Analysis and cleansing of raw data is critical to develop a robust model. Explain the data cleansing, exploratory analysis and transformation activities you would perform out as part of the model development exercise.

After performing EDA (Exploratory Data Analysis) we find many data inconsistencies such as Missing values and incorrect values.

Missing Values – they were found in *"years_at_residence"* column with the help *is.na()* function.

These values will be get replaced by the mean of that data.

Incorrect Values – these were in column *"purpose"*. Values were converted from "car0" to "car".

Q3) List down and explain various modelling methodologies / frameworks that could be adopted for developing the credit scoring model.

Logistic Regression

It is a Generalized Linear Model which is used to model a binary categorical variable.

Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function.

- No Multicollinearity
- Uses Maximum Likelihood Estimation technique for parameter estimation

Decision tree

It is a non-parametric supervised machine learning method used for both classification and regression tasks.

Two types of Decision tree:

- Classification tree
- Regression tree

Q4) Explain the activities & control-checks you would carry to ensure that explanatory variables selected as part of the variable selection process are optimal

For variable selection process first step is to select all the significant values which have p value less than 0.05.

We will also use Information value and Wald test for some specific variables to achieve lowest AIC among all the iterations.

After removing insignificant variables, we check for Multicollinearity by using variation inflation factor. Here we will assume a VIF threshold value of 2.5. removing all the variables which are having vif more than 2.5.

Q5) Applications of a Credit Scoring Model

The most primary use of scoring model is to. loan underwriting and assess creditworthiness of existing obligors in the banking book.

It helps financial institutions control allocation of risk and costs with their customers.

Businesses can know immediately if they are dealing with a high-risk or low-risk customer and can operate more efficiently and reduce the cost of vital services like mortgages, car loans and credit cards.

Banks make money by selling results of the models like:

- Bankruptcies or missed payments
- Occupation
- Whether you own or rent your residence.
- The Balances
- Age
- Credit history