INTRODUCTION:

Credit scoring problem is widely used in many public organizations such as banks or private organizations such as credit card companies etc. They use credit scores which is a numerical value representing the creditworthiness of a person. This is evaluated by the analysis of person's portfolio which may include his/her revenues, information about mortgages, the property owned by person.

PROBLEM STATEMENT:

The purpose of the study is to build a classifier to predict the granting of retail credits. To this purpose we have a database of recent concessions of credits done by analysts. The goal of the classifier is to imitate the analysts and decision support. For this problem, we are provided with a real data. The credit scoring problem in this case study is the decision of granting loans to the customer to buy an asset.

METHODOLOGY:

The credit scoring problem dealt in this case study is building a classifier to predict whether a customer should be granted for credit or not. The methodology to do such task is divided into following steps:

1) Preprocessing the data
2) Feature Extraction
3) Building a classifier
4) Cross validation for selecting best classifier
5) Applying classifier on test data
6) Cost computation

The data is divided into two parts: training and test data set which represents a sample from the whole population. The training dataset is preprocessed first in order to take some actions with the missing values and outlier detection etc. Both steps are described in details in later sections.

DATA DESCRIPTION:

The data description is given as follows:
Total number of variables: 14

Variable Name	Type	Description
Opinion	Categorical	Response Variable with 2 modalities (1 – positive, 2 – negative).
Number.of.employment.years	Continuous	Variables for the years the person worked.
House.type	Categorical	Variable for the type of house where the person is living; 1-rented, 2-ownerwithdeed, 3-private, 4-ignore, 5-parents, 6-others.
term.in.months	Continuous	Variable indicating whether the person registers for house or not; 1-No, 2-Yes.
Age	Continuous	Variable indicating the age of person.
Marital.status	Categorical	Variable for the marital status of the person; 1-single, 2-married, 3-widower, 4-separated, 5-dicorced.
Registers	Categorical	Variable indicating the installment terms in months.
Type.of.employment	Categorical	Variable indicating the type of employment the person have; 1- Permanent, 2-Temporaray, 3-Self, 4-Others.
Expenses	Continuous	Variable indicating the expenses of the person.
income	Continuous	Variable indicating the total income of the person.
total.wealth	Continuous	Variable indicating the total wealth a person have.
mortgage.left	Continuous	Variable indicating the mortgage left to pay by the person.
amount.of.money.solicitated	Continuous	Variable indicating the total wealth of the person taking into account the mortgage left to pay.
price.of.good.to.buy	Continuous	Variable indicating the price of good the person wants to buy.

PREPROCESSING OF THE DATA:

After analysing the data from the summary of the data, we can see that there are missing values in the original dataset.

1. Missing values handling:

The steps to handle the missing values are as following:

Updating the missing values to NA, to do this there are 2 cases

- The numerical variables which appear as "99999999". Since it was stated in the problem statement that the missing values are denoted by "99999999" in the dataset. So it is converted to NA for convenience of R language.
- The categorical variables which appear as "0". Since it was not mentioned that it is missing value or it is a value in the dataset. We converted it into missing values by considering that due to some mistake in the data, the missing values are also denoted by 0.

```
input$income[input$income=="99999999"] <- NA
input$mortgage.left [input$mortgage.left =="99999999"] <- NA
input$total.wealth [input$total.wealth =="99999999"] <- NA
input$total.wealth [input$total.wealth =="99999999"] <- NA
input$Type.of.employment [input$Type.of.employment =="0"] <- NA
input$House.type [input$House.type =="0"] <- NA
```

2. Outlier detection

   Since from the given training data, the data is well prepared by the experts thus we have no outlier thus no correction is need for this case. However, for testing data set we will need to consider this step.

We factor the variables which should be the categorical variable and label the modalities.

Variable Name	Number of Modalities	Modalities
House.type	6	1-Rented , 2-Ownerwithdeed , 3-Private 4-Ignore , 5-Parents , 6-others
Marital.status	5	1-Single , 2-Married, 3-Widower

		4-Separated, 5-divorced
Registers	2	1-No, 2-Yes
Type.of.employment	4	1-Permanent , 2-Temporary 3-Self, 4-Others
Opinion	2	5-Positive , 6-Negative

The missing values are filled using the process of multiple imputations which is done using MICE library in R. The data set is replicated 5 times and equal weights are assigned to them.

CHI-SQUARE TEST

To evaluate the link between each category of variables, a chi-square test is performed. The more significant the test is, the more the considered category and categorical variable are linked.

```
> cat3$test.chi2
                   p.value df
House.type               0  5
Marital.status           0  4
Registers                0  1
Type.of.employment       0  3
```

From the result, it can be seen that p.value of all categorical variables are less than 0.05 (at 95% confident). Thus the categorical variables here are significant.

Next, in order to see the link between one category of opinion (Positive and Negative) and another category of another categorical variable of the data set, the function compares two proportions:

a) The proportion of individuals who possess the second category among those who possess the first
b) The global percentage of individuals who possess the second category

```
> cat3$category
$Positive
                              Cla/Mod    Mod/Cla     Global      p.value     v.test
Type.of.employment=Permanent 78.90824 68.8919796 63.0676490 0.000000e+00        Inf
Registers=No                 77.66377 91.2378303 84.8626926 0.000000e+00        Inf
Marital.status=married       74.51070 75.8924432 73.5766912 0.000000e+00        Inf
House.type=ownerwithdeed     80.82538 53.3889662 47.7160080 0.000000e+00        Inf
Marital.status=widower       82.60870  1.7617061  1.5405224 9.506871e-47  14.357897
House.type=parents           71.77388 17.0700046 17.1801741 2.594413e-02  -2.227047
House.type=private           68.18182  4.5897079  4.8626926 1.153856e-21  -9.562105
House.type=ignore            54.54545  0.2781641  0.3683858 2.398147e-27 -10.833130
Marital.status=divorced      56.00000  0.6490496  0.8372405 2.446641e-51 -15.072763
```

```
Marital.status=single          67.08861 19.6569309 21.1654387 1.391029e-170 -27.840828
Type.of.employment=Others      57.62082  2.8743625  3.6034829 5.927798e-181 -28.684661
Type.of.employment=Self        67.18275 22.5312935 24.2263898 6.453192e-196 -29.860297
Marital.status=separated       51.16279  2.0398702  2.8801072 6.792724e-290 -36.387170
Type.of.employment=Temporaray  45.25386  5.7023644  9.1024782  0.000000e+00       -Inf
Registers=Yes                  41.81416  8.7621697 15.1373074  0.000000e+00       -Inf
House.type=others              51.44864  5.4334724  7.6289350  0.000000e+00       -Inf
House.type=rented              62.48118 19.2396847 22.2438044  0.000000e+00       -Inf

$Negative
                                Cla/Mod    Mod/Cla     Global       p.value     v.test
Type.of.employment=Temporaray  54.74614 17.9493366  9.1024782  0.000000e+00        Inf
Registers=Yes                  58.18584 31.7249698 15.1373074  0.000000e+00        Inf
House.type=others              48.55136 13.3413752  7.6289350  0.000000e+00        Inf
House.type=rented              37.51882 30.0603136 22.2438044  0.000000e+00        Inf
Marital.status=separated       48.83721  5.0663450  2.8801072 6.792724e-290  36.387170
Type.of.employment=Self        32.81725 28.6369119 24.2263898 6.453192e-196  29.860297
Type.of.employment=Others      42.37918  5.5006031  3.6034829 5.927798e-181  28.684661
Marital.status=single          32.91139 25.0904704 21.1654387 1.391029e-170  27.840828
Marital.status=divorced        44.00000  1.3268999  0.8372405  2.446641e-51  15.072763
House.type=ignore              45.45455  0.6031363  0.3683858  2.398147e-27  10.833130
House.type=private             31.81818  5.5729795  4.8626926  1.153856e-21   9.562105
House.type=parents             28.22612 17.4668275 17.1801741  2.594413e-02   2.227047
Marital.status=widower         17.39130  0.9650181  1.5405224  9.506871e-47 -14.357897
Type.of.employment=Permanent   21.09176 47.9131484 63.0676490  0.000000e+00       -Inf
Registers=No                   22.33623 68.2750302 84.8626926  0.000000e+00       -Inf
Marital.status=married         25.48930 67.5512666 73.5766912  0.000000e+00       -Inf
House.type=ownerwithdeed       19.17462 32.9553679 47.7160080  0.000000e+00       -Inf
```

From the result it can be seen that Type of employment = Permanent is over represented (as v-test>0) while Type.of.employment=Temporaray is under represented (as v-test<0) among individual who has positive response for credit approval.

From the figures, we found that 78% of the individuals who has positive response belong to permanent employee while 54% of the individuals who has negative response belong to temporary employee.

For each category of “opinion” and each continuous variable, we use quanti.var to see the global description of the variable by the quantitative variables with the square correlation coefficient and the p-value of the test in a one-way analysis of variance (assuming the hypothesis of homoscedsticity which means the variables in the sequence or vector have the same finite variance).

```
> cat3$quanti.var
                                    Eta2       P-value
Number.of.employment.years  6.419749e-02  0.000000e+00
term.in.months              1.186941e-02  0.000000e+00
Age                         7.206822e-03  0.000000e+00
income                      4.291116e-02  0.000000e+00
amount.of.money.solicitated 2.561571e-02  0.000000e+00
total.wealth                6.298688e-03 1.134928e-309
mortgage.left               7.319774e-04  1.524544e-37
Expenses                    5.498273e-04  1.282363e-28
price.of.good.to.buy        7.874732e-05  2.674827e-05
```

FEATURE EXTRACTION:

In this step, we made some variables as categorical variables. This is done by using decision trees. For each explanatory variable, a classification decision tree is constructed and then the predicted values are plotted against the explanatory variable. Next, it is checked if the plot obtained is a linear plot i.e. there is a linear relationship between the predicted values and the explanatory variable. If it is not linear, the explanatory variable is converted as a categorical variable. This feature extraction will help in reducing the number of features so that the model doesn't overfit the data leading to bad results.

Two extra variables are also added to new dataset with imputed values.

1) Funding ratio : - Funding ratio is introduced taking into account value of the total wealth of the person and the mortgage left to pay and price of good that she/he would like to buy to express the ratio of money solicitated of an asset purchased. The lower the funding ratio, the riskier the loan is for a bank.

$$\boldsymbol{Funding\ ratio = \frac{amount\ of\ money\ solicitated}{price\ of\ good\ to\ buy} * 100}$$

2) Saving Capacity : Saving Capacity is introduced to express the ability or power of borrower in saving money perspective taking into account his/her income, expense, mortgage, and the amount of money solicitated. The lower the saving capacity, the riskier the loan is for a bank.

$$\boldsymbol{Saving\ Capacity = \frac{income - expenses - \frac{mortgage\ left}{100}}{\frac{amount\ of\ money\ solicitated}{term\ in\ months}} * 100}$$

In summary, from the graph obtained we found that following variables can be taken as linear: age, income, amount.of.money.solicitated, mortgage, price.of.good.to.buy while total.wealth, expenses, Number.of.employment.years, term.in.months, funding.ratio and saving.capacity we will need to transform them to Categorical or Binary as it is not linear and have discrete values.

From the result obtained from decision tree, we transform the variables as below, the binary decisions are made on the basis of continuous features by determining a threshold that divides the range of values into intervals correlated with decisions (e.g. saving capacity). In addition, multiple splits is used as many partitions as distinct values (e.g. Number.of.employment.years):

Variable Name	Type	Description
Number.of.employment.years	Categorical	[Number.of.employment.years<1] <- 1 [Number.of.employment.years>=1 & Number.of.employment.years< 5] <- 2 [Number.of.employment.years>=5] <- 3
term.in.months	Categorical	[term.in.months==12] <- 1 [term.in.months==24] <- 2 [term.in.months==36] <- 3 [term.in.months==48] <- 4 [term.in.months==60] <- 5
Expenses	Categorical	[Expenses< 44.5] <- 1 [Expenses>=44.5 & Expenses< 85.5] <- 2 [Expenses>= 85.5 & Expenses < 145] <- 3 [Expenses>= 145] <- 4
total.wealth	Categorical	total.wealth [p[,1]>0.4] <- 1 total.wealth [p[,1]<=0.4] <- 0
Funding.ratio	Categorical	[funding.ratio< 70] <- 1 [funding.ratio>=70 & funding.ratio< 98] <- 2 [funding.ratio>=99] <- 3
Saving.capacity	Categorical	[saving.capacity< 1] <- 1 [saving.capacity>=1] <- 2

DIFFERENT CLASSIFIERS:

To select the best classifier, several options are used. The four options for the classifiers which were used are stated below.

1) Classification trees: This classifier uses decision tree which is implemented using "rpart" function in R package. The maximum depth of the tree is taken to be 4 and cp to be 0.001.

```
n= 7345

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 7345 2127 Positive (0.7104152 0.2895848)
   2) Type.of.employment=Permanent,Self,Others 6440 1512 Positive
(0.7652174 0.2347826)
     4) Registers=No 5030  842 Positive (0.8326044 0.1673956)
       8) income>=27.5 4551  618 Positive (0.8642057 0.1357943) *
       9) income< 27.5 479  224 Positive (0.5323591 0.4676409)
        18) House.type=ownerwithdeed,private,parents 369  139 Positive
(0.6233062 0.3766938) *
        19) House.type=rented,others 110   25 Negative (0.2272727
0.7727273) *
     5) Registers=Yes 1410  670 Positive (0.5248227 0.4751773)
      10) House.type=ownerwithdeed,parents 879  299 Positive (0.6598407
0.3401593) *
```

```
      11) House.type=rented,private,ignore,others 531  160 Negative
(0.3013183 0.6986817)
        22) Age< 31.5 145   70 Positive (0.5172414 0.4827586) *
        23) Age>=31.5 386   85 Negative (0.2202073 0.7797927) *
   3) Type.of.employment=Temporaray 905  290 Negative (0.3204420
0.6795580)
     6) total.wealth>=7500 45   10 Positive (0.7777778 0.2222222)
      12) income>=95 35    0 Positive (1.0000000 0.0000000) *
      13) income< 95 10    0 Negative (0.0000000 1.0000000) *
     7) total.wealth< 7500 860  255 Negative (0.2965116 0.7034884)
      14) amount.of.money.solicitated< 415 95   40 Positive (0.5789474
0.4210526)
        28) Expenses=less miser 25    0 Positive (1.0000000 0.0000000) *
        29) Expenses=miser 70   30 Negative (0.4285714 0.5714286) *
      15) amount.of.money.solicitated>=415 765  200 Negative (0.2614379
0.7385621) *
```

2) Kernel Support Vector Machines: This classifier uses support vector machines which is implemented in “ksvm” function of the R package. The most important parameter for the SVM is cost which is taken as 5 in this case.

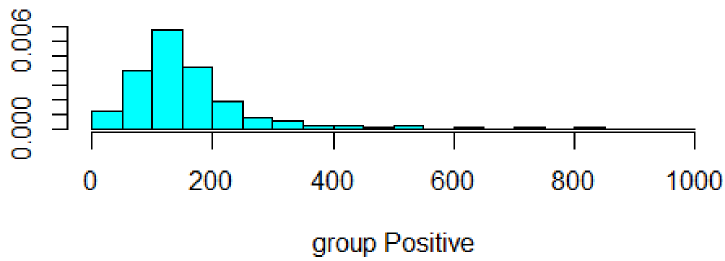
```
Support Vector Machine object of class "ksvm"

SV type: C-svc  (classification)
 parameter : cost C = 5

Gaussian Radial Basis kernel function.
 Hyperparameter : sigma =  6.99062114061969e-07

Number of Support Vectors : 3935

Objective Function Value : -16660.27
Training error : 0.185841
```

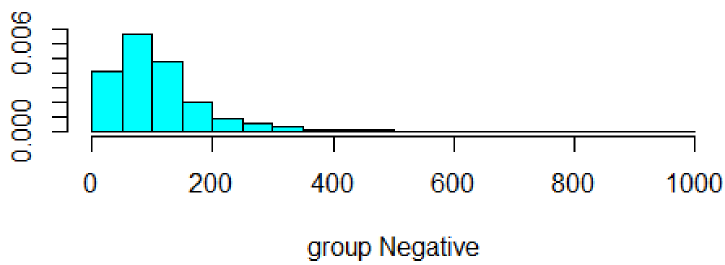
3) Linear Discriminant Analysis: This classifier uses linear discriminant analysis which is implemented in “lda” function of the R package.

```
Call:
lda(data$Opinion ~ data$Age + data$income +
data$amount.of.money.solicitated +
    data$price.of.good.to.buy, data = data, CV = FALSE)

Prior probabilities of groups:
 Positive  Negative
0.7104152 0.2895848

Group means:
         data$Age data$income data$amount.of.money.solicitated
data$price.of.good.to.buy
Positive 37.68781   131.90303                          1015.69
1467.893
Negative 34.84720    95.98072                          1167.36
1487.309

Coefficients of linear discriminants:
                                          LD1
data$Age                         -0.026545312
data$income                      -0.009817864
```

```
data$amount.of.money.solicitated  0.002093565
data$price.of.good.to.buy        -0.000960713
```

Visualizing the Results:





4) Generalized Linear Model: This model builds a logistic regression model which is implemented using "glm" function implemented in R package.

```
Call:  glm(formula = data$Opinion ~ data$Number.of.employment.years + 
    data$House.type + data$term.in.months + data$Age + data$Marital.status 
+ 
    data$Registers + data$Type.of.employment + data$Expenses + 
    data$income + data$total.wealth + data$mortgage.left + 
data$amount.of.money.solicitated + 
    data$price.of.good.to.buy + data$saving.capacity + data$funding.ratio, 
    family = binomial(link = "logit"), data = data)

Coefficients:
                      (Intercept)    data$Number.of.employment.years  
                        5.412e-01                         -6.424e-01  
     data$House.typeownerwithdeed             data$House.typeprivate  
                       -8.775e-01                         -3.552e-02  
            data$House.typeignore             data$House.typeparents  
                       -1.077e-01                         -7.373e-01  
            data$House.typeothers                data$term.in.months  
                       -7.869e-02                          6.322e-03  
                         data$Age          data$Marital.statusmarried  
                       -4.316e-04                         -2.698e-01  
         data$Marital.statuswidower      data$Marital.statusseparated  
                        1.455e+00                          1.087e+00  
        data$Marital.statusdivorced                  data$RegistersYes
```

```
                   -1.476e+00                            1.695e+00
data$Type.of.employmentTemporaray       data$Type.of.employmentSelf
                   1.893e+00                             2.720e-01
    data$Type.of.employmentOthers           data$Expensesless miser
                   5.485e-01                             3.534e-01
     data$Expensesless spenthrift           data$Expensesspenthrift
                   1.030e+00                            -8.709e+00
                     data$income                  data$total.wealth
                   -9.382e-03                           -4.379e-05
            data$mortgage.left   data$amount.of.money.solicitated
                   8.331e-05                             1.979e-03
       data$price.of.good.to.buy               data$saving.capacity
                   -7.941e-04                                   NA
            data$funding.ratio
                   -7.543e-03

Degrees of Freedom: 7344 Total (i.e. Null);  7319 Residual
Null Deviance:      8840
Residual Deviance: 6245        AIC: 6297
```

CROSS VALIDATION:

The cross validation is performed particularly 10-fold cross validation method. Since the data set is formed by replicating the initial training data replicated 5 times and merged together. So dividing the initial data into 10 parts and assigning the same fold number to each row in 5 sets of initial data set. Then, the 10-fold cross validation is performed and each time the error is computed and stored in a vector. This error computation is done in the same way for all four models and at the end the average error is computed for all models. It has been found that the least mean error is obtained when the **decision tree classifier** is used. The table below shows the mean error obtained by all four models by the process of 10-fold cross validation.

We considered error as the measure for model selection. From the result obtained, we observed that “Classification tree” provided least cross validation error followed by Generalized Linear Models. Therefore, we will choose “Classification tree” as our classifier model. In particular, the decision tree approach has become a popular technique for developing credit scoring models because the resulting decision trees are easily interpretable and visualized. The parameter we used for building tree model is given below. We consider to build tree such that it compromises following 2 aspects

1) A low training error
2) A tree that is not too large

```
tree<-rpart(data$Opinion~.,data=data,control=rpart.control(cp=0.0001,maxdepth = 4, method="class"))
```

COMPARISON OF MODELS:

MODEL	MEAN ERROR
Classification tree	0.1995916
Kernel Support Vector Machines	0.2054186
Linear Discriminant Analysis	0.2754255
Generalized Linear Models	0.2006807

TEST DATASET:

After the selection of best choice for the classifier for this credit scoring problem which is classification trees, the classifier is applied on the test dataset. The preprocessing of the test data is carried out similarly as done while preprocessing of training dataset. The missing values indicated as “9999999” and “0” in the data are replaced by NAs for the convenience of R language. The attributes which were used as categorical variables while training the classifier are also made as categorical in the test dataset. Next, the two other attributes “funding ratio” and “saving capacity” are also added.

- Missing Values:

The missing values in the test data set are filled by the method of imputation similarly as done in the training data set. The missing values are filled according to the training dataset so that both datasets come from same population. To do this, the training dataset (which is the set of replicating the data five times) is divided into five parts. Each part is merged with test data separately. Then using the MICE package in R, the imputation of missing values is performed on five datasets. After performing the imputation, the test data is separated from all five datasets and these five subsets are merged together to produce final test dataset which contains no missing values.

- Classifier on Test Dataset:

The “decision tree classifier” is applied on the obtained test data set after the imputation of missing values. The settings of this classifier in R are the same as obtained in the training of the classifier. The maximum depth of the tree is 4. The error on this test data set is found out to be equal to 19.95 %.

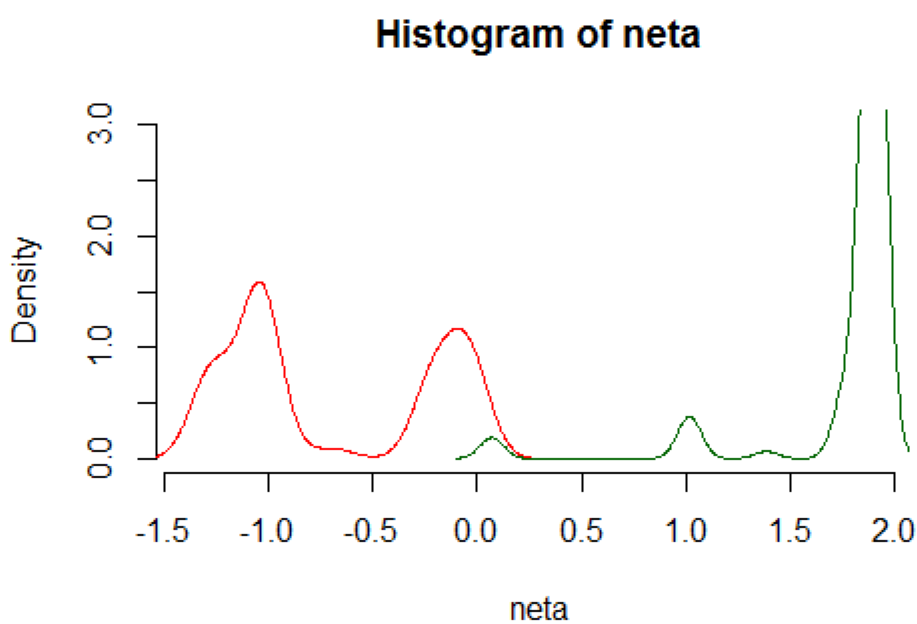
COST COMPUTATION:

There may be some situations in which a person is a good person but yet he is not granted the credit or may be the person is bad, but he is granted with the credit. In both situations, the bank is at loss. To deal with this situation in this grey area, bank might need to dedicate person to investigate and perform a judgmental review on particular case. This step came with the human resource cost which consider as an expense for organization. Thus, we consider to compute the cost taking into account the human resource cost and the cost of wrongly classified. From this cost obtained, one can adjust the threshold to compromise the grey area and the expense that may occurred to be used as final model.

A density value is computed using the formula below. This has been used since we are using the decision trees.

$$\boldsymbol{\eta = \log_{10}\left(\frac{p}{1-p}\right); p\ being\ the\ predicted\ probability\ of\ opinion}$$

After this, it is divided into two sub probabilities where Opinion is positive and negative. This is denoted by η_+ and η_- . The density plot is plotted as given below.



Now a prior probability is set for both cases when Opinion is positive and negative. This means that the probability that a person is being granted credits is 0.55 and the probability that a person is being refused credit is 0.53.

$$p_+ = 0.55$$

$$p_- = 0.53$$

The threshold probabilities have been set in both cases which is computed as below.

$$thres - pos = \log_{10}\left(\frac{p_+}{1 - p_+}\right) = 0.2006707$$

$$thres - neg = \log_{10}\left(\frac{p_-}{1 - p_-}\right) = 0.1201443$$

Now the probabilities that when a person is good but refused the credits and when a person is bad but granted the credit are computed using the formulas given below.

$$P(Opinion = Positive|Person = bad) = sum(\eta_- > thres - pos)/length(\eta_-)$$

$$P(Opinion = Negative|Person = good) = sum(\eta_+ > thres - neg)/length(\eta_+)$$

$P(Opinion = Positive\|Person = bad)$	**= 0**
$P(Opinion = Negative\|Person = good)$	**= 0.02724028**

The manual probability is computed using the formula given below.

$$\pi_- = 0.13$$

$$Prob_{manual} = P(\eta \leq \eta_+) - P(Opinion = Negative|Person = good) * (1 - \pi_-) + P(\eta \geq \eta_-) - P(Opinion = Positive|Person = bad) * \pi_-$$

$$Prob_{manual} = 0.003541236$$

COST COMPUTATION:

$$Cost_{manual} = 20 * Prob_{manual} = 0.07082472$$

$$Cost_{Opinion=positive|Person=bad} = 1000 * P(Opinion = Positive|Person = bad) * \pi_-$$

$$Cost_{Opinion=negative|Person=good}s = 1000 * 0.14 * P(Opinion = Negative|Person = good) * (1 - \pi_-)$$

$Cost_{manual}$	**0.07082472**
$Cost_{Opinion=positive\|Person=bad}$	**0**
$Cost_{Opinion=negative\|Person=good}$	**3.317866**

CONCLUSION

Nowadays, credit scoring model becomes more and more important as it helps financial organization to decide whether or not to grant the credit. This work is done by taking into account the historical data and extracting the features in order to access the creditworthiness of the applicants. The model is derived using classification tree since it has least error comparing to another algorithms. However, there is a case where it is required analyst to make the final decision on credit approval and to minimize the resources and time the cost is introduced to assist the proper threshold.