ISSN No. 0976-5697

Volume 4, No. 10, Sep-Oct 2013

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Predictive Mining on Loan Data using Rattle

Venkata Sreedhar Ventrapragada
Department of Information Technology
Vignan's Institute of Information Technology
Duvvada, Visakhapatnam- 530049, A.P., INDIA.
v.v.sreedhar@live.in

Abstract: Data mining on large database has been a major concern in research community due to the difficulty in analyzing huge volumes of data. Data mining refers to extracting or mining knowledge from large amounts of data. This paper attempts to explain data mining techniques, particularly "classification", to predict which loan applicants are "risky" and which are "safe" for loan data set. For this purpose we use Rattle—"R Analytical Tool To Learn Easily" which is a powerful platform for data mining. Some preprocessing techniques such as Data Cleaning, Relevance Analysis, Data Transformation and Data Reduction have to be applied to improve the accuracy and efficiency of the classification process i.e removing noise by filling missing values, performing correlation analysis for identifying redundancies, scaling data to a specified range and performing principal component analysis (PCA) for reducing the dimensionality of data set. The main goal is to classify the loan applicants by classification based on decision tree method.

Keywords: Classification, Rattle, preprocessing, PCA, decision tree method.

I. INTRODUCTION

Data Mining is also known as Knowledge discovery in database. Extraction of interesting patterns or knowledge from huge amount of data**.** Data Mining is the discovery of knowledge of analyzing enormous set of data by extracting the meaning of the data and then predicting the future trends and also helps companies to take sound decisions, based on knowledge and information. Data mining software **RATTLE** is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified [1][2]. The Data mining concept is introduced because of the reasons like Automated data collection tools and mature database technology lead to tremendous amounts of data stored in databases, data warehouses and other information repositories.

RATTLE (the R Analytical Tool To Learn Easily) nothing but software provides R console increasingly provides a powerful platform for data mining [3].The Rattle package provides a graphical user interface specially for data mining using R. It also provides a stepping stone toward using R as a programming language for data analysis. It presents statistical and visual summaries of data, transforms data into forms that can be readily modeled. Rattle provides considerable data mining functionality by exposing the power of the R Statistical Software through a graphical user interface. The capabilities of R are extended through user-submitted packages, which allow specialized statistical techniques, graphical devices, as well as import/export capabilities to many external data formats. Rattle uses these packages - RGtk2, pmml, colorspace, ada, amap, arules, biclust, cba, descr, doBy, e1071, ellipse, fEcofin, fBasics, foreign, fpc, gdata, gtools, gplots, gWidgetsRGtk2, Hmisc, kernlab, latticist, Matrix, mice, network, nnet, odfWeave, party and many more packages are available. The packages like rattle, outliers, rjava, plot, tree, cluster, MASS, akima, abc, and many more are available [4]. The rattle package is very important in the RATTLE which provides datasets and a GUI called DATA MINER. In data miner using tabs provided, we can simply generate the cleaned data and classification too [5]. The code will be appeared for a particular action when we click the log button.

II. LOAN DATA SET

The loan dataset contains thirteen attributes namely ID, Age, Employment, Education, Marital, Occupation, Income, Gender, Deduction, Hours, IGNORE_Accounts, RISK_Adjustment and LOAN_Decision. The detailed description of the dataset is given below

Table I. Loan Data Set

Attributes	Levels	Storage	NAs
Age		Integer	0
Employment	8	Integer	75
Education	16	Integer	
Marital	6	Integer	
Occupation	14	Integer	76
Income		Double	
Gender	2	Integer	
Deduction		Double	
Hours		Integer	8
RISK_Adjustment		Integer	
LOAN_Decision	2	Integer	

© 2010, IJARCS All Rights Reserved

Variable	Levels
Employment	Consultant,Private,PSFederal,PSLocal,PSState,SelfEmp,Unemployed,Volunteer
Education	Associate, Bachelor, College, Doctorate, HSgrad, Master, Preschool Professional, Vocational, Yr10, Yr11, Yr12, Yr1t4, Yr5t6, Yr7t8, Yr9
Marital	Absent, Divorced, Married, Married-spouse-absent, Unmarried, Widowed
Occupation	Cleaner, Clerical, Executive, Farming, Home, Machinist, Military, Professional, Protective, Repair, Sales, Service, Support, Transport
Gender	Female, Male
LOAN Decision	risk, safe

Figure 1. Introduction to loan data set

III. DATA PREPROCESSING

Data preprocessing is a data mining technique that involves transforming raw data into an understandable format [6]. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors. Data preprocessing is a proven method of resolving such issues. Data preprocessing prepares raw data for further processing. Raw data is highly susceptible to noise, missing values, and inconsistency. The quality of data affects the data mining results. In order to help improve the quality of the data and, consequently, of the mining results raw data is pre-processed so as to improve the efficiency and ease of the mining process. Data preprocessing is one of the most critical steps in a data mining process which deals with the preparation and transformation of the initial dataset[7].

Data preprocessing methods are divided into following categories:

A. Data Cleaning
B. Relevant Analysis
C. Data Transformation
D. Data Reduction

A. Data Cleaning:

Real-world data tend to be incomplete, noisy, and inconsistent. Data cleaning (or data cleansing) routines attempt to fill in missing values, smooth out noise while identifying outliers, and correct inconsistencies in the data. This method is responsible for performing the following:

(a). Removing or replacing missing values
(b). Identifying and removing outliers

a) Removing or replacing missing values:

There are five different ways for replacing the missing values. They are:

i. Replacing missing values with zero's
ii. Replacing missing values with attribute mean
iii. Replacing missing values with attribute median
iv. Replacing missing values with attribute mode
v. Replacing missing values with a global constant

a. Replacing missing values with zero's

```
setwd ("E:/Program Files/R/R-2.15.0/library/rattle/csv")
loan.with.na<-read.csv ("loan.csv", header=TRUE)
rmv<-loan.with.na
rmz<-rmv$Hours
rmz
```

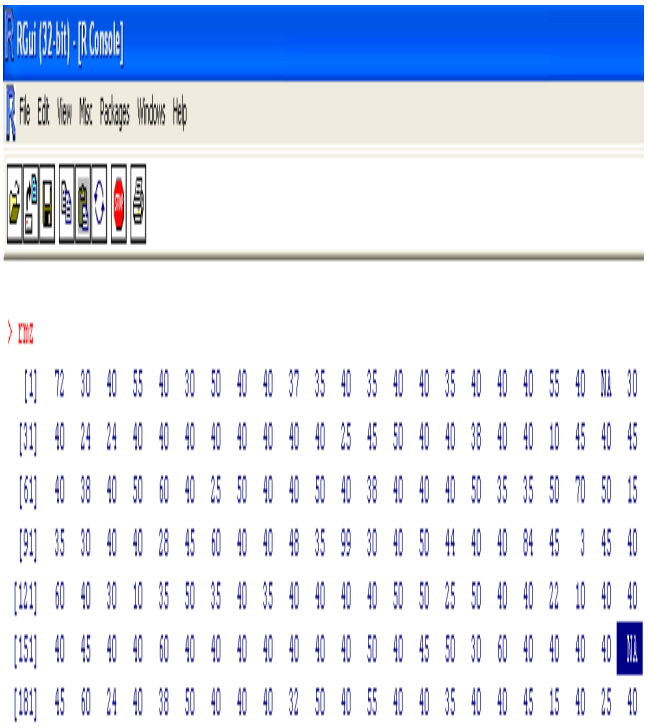


Figure 2. Replasing missing values with zero's

```
rmz[is.na(rmz)]<-0
rmz
```

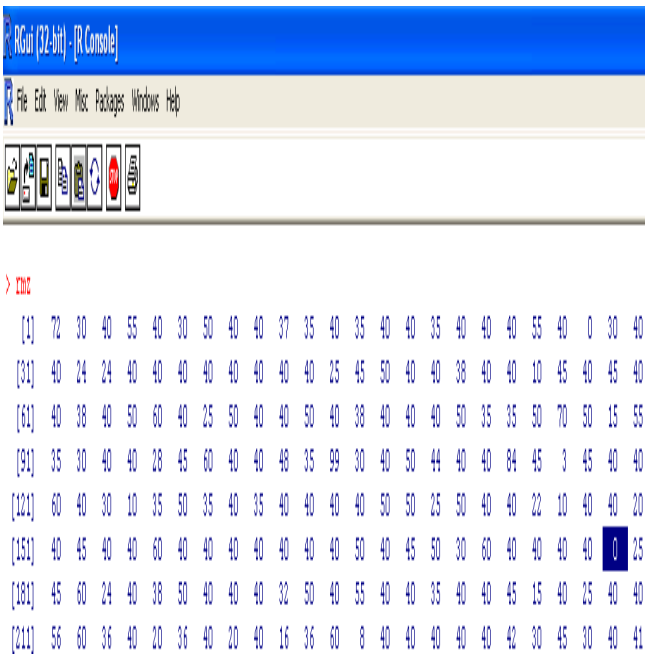


Figure 2.1.

b. Replacing missing values with attribute mean:

```
rmm<-rmv$Hours
rmm[is.na (rmm)] <-mean (rmm, na.rm=TRUE)
rmm
```

© 2010, IJARCS All Rights Reserved

Figure 3. Replacing missing values with attribute mean

c. Replacing missing values with attribute median:

rmme<-rmv$Hours
rmme[is.na (rmme)] <-median (rmme, na.rm=TRUE)
rmme

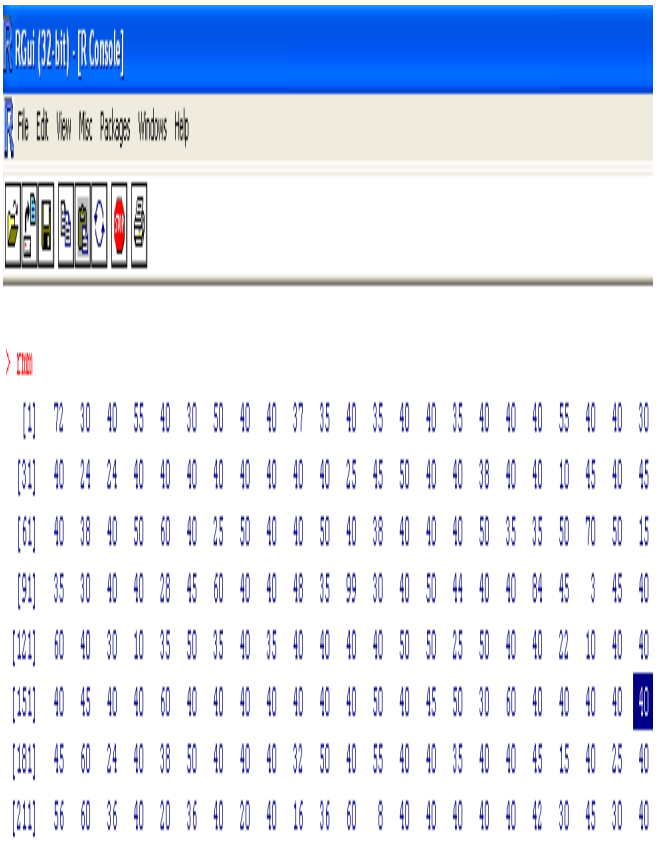


Figure 4. Replacing missing values with attribute median

d. Replacing missing values with attribute mode

rmmo<-rmv$Occupation
rmmo

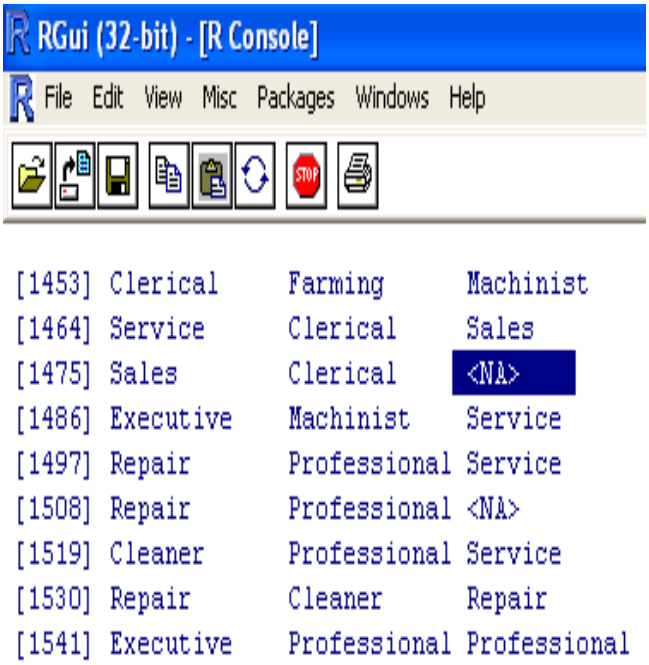


Figure 5. Replacing missing values with attribute mode

a<- factor (rmmo)
r<-table (a)
r

Cleaner Clerical Executive Farming Home Machinist Military Professional Protective Repair Sales
91 233 289 58 5 139 1 247 40 225 206 Service Support Transport
211 49 107

rmmo [is.na (rmmo)] <-"Executive"
rmmo

Figure 6.

e. Replacing missing values with a global constant:

rmg<-rmv$Employment
rmg

© 2010, IJARCS All Rights Reserved

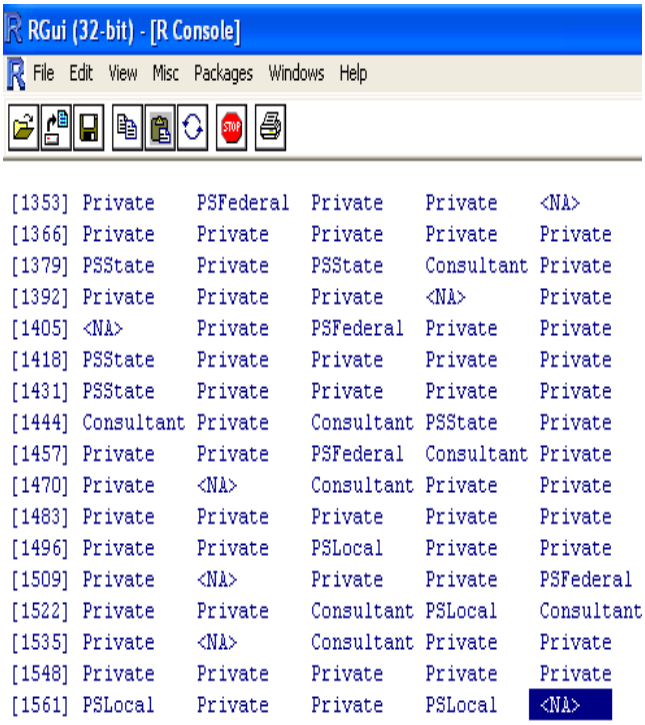


Figure 7. Replacing missing values with a global constat

rmg<-as.character (rmg)
rmg [is.na (rmg)] <-"Unknown"

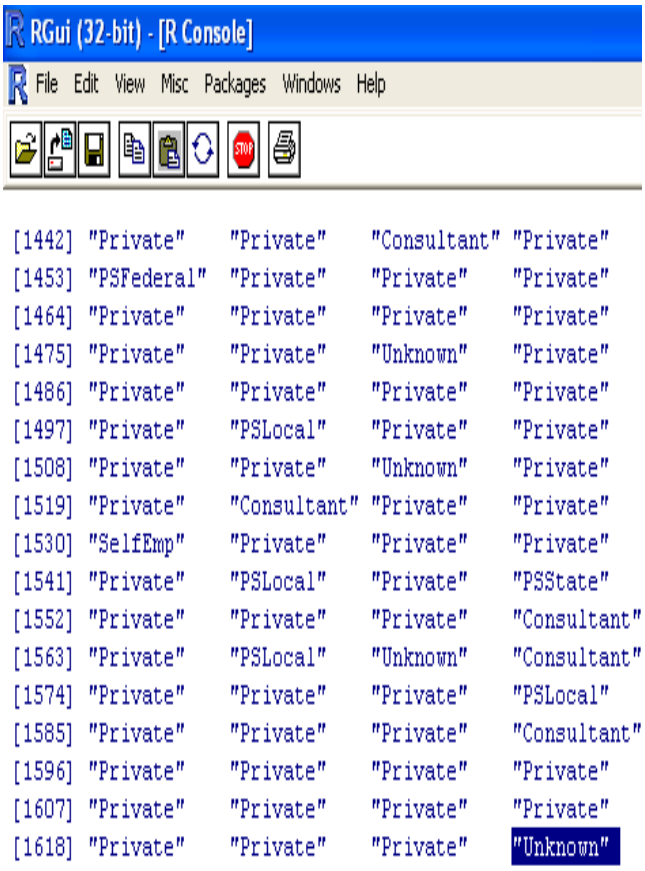


Figure 7.1

Removing missing values

Set working directory
setwd ("E:/Program Files/R/R-2.15.0/library/rattle/csv")
loan.with.na<-read.csv ("loan.csv", header=TRUE)
loan.with.na

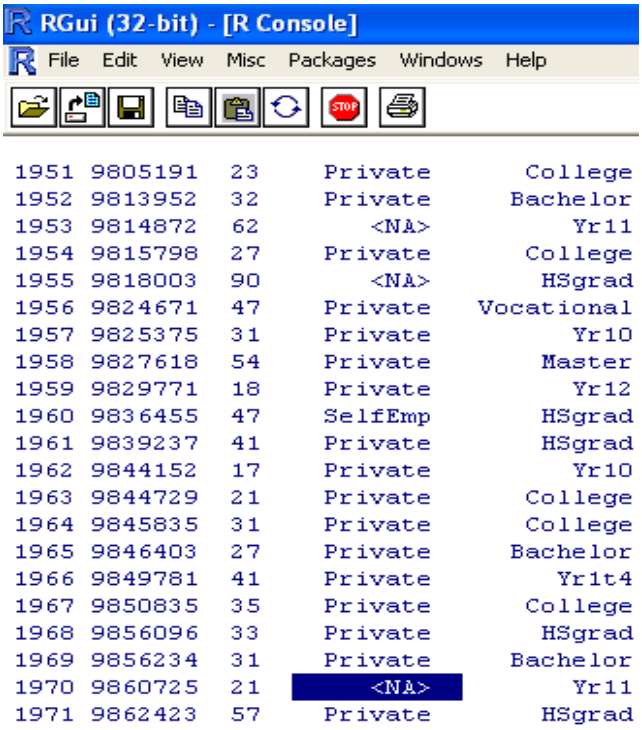


Figure 8. Replacing missing values

dim(loan.without.na)
[1] 2001 13
loan.without.na<-na.omit (loan.with.na)
loan.without.na

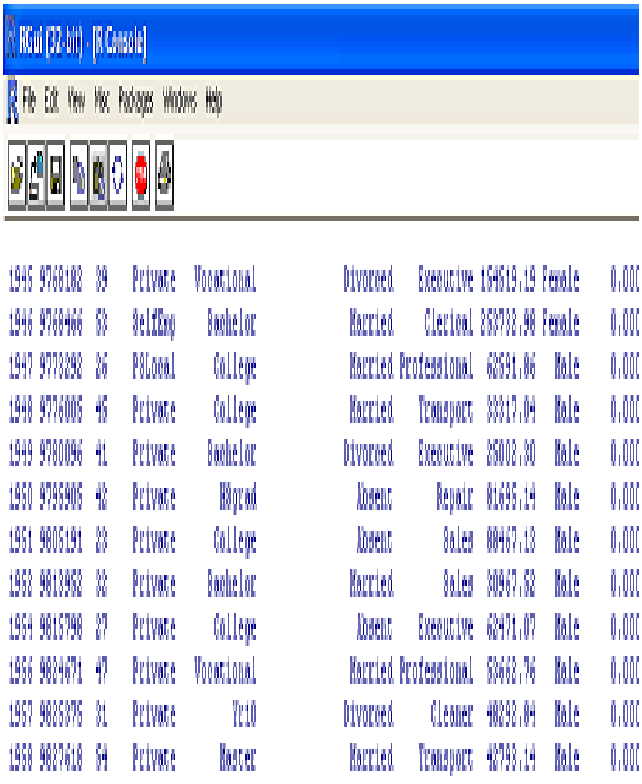


Figure 8.1

dim(loan.without.na)
[1] 1851 13

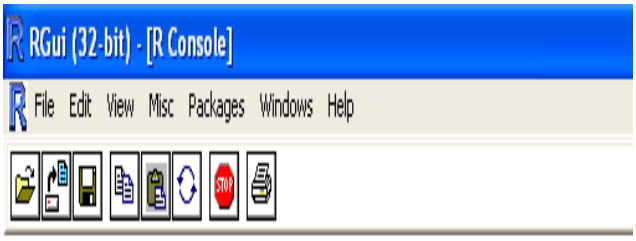
b) Identifying and removing outliers:

In this, the outliers are first identified and then removed.

library (outliers)

© 2010, IJARCS All Rights Reserved

```
out = outlier (loan.without.na$Hours, logical=TRUE)
find_outlier = which (out==TRUE, arr.ind=TRUE)
loan.out = loan.without.na [-find_outlier,]
```




```
1943 9750086  39    Private       HSgrad              Absent
1945 9768182  39    Private   Vocational          Divorced
1946 9768466  53    SelfEmp     Bachelor           Married
1947 9773292  26    PSLocal      College           Married
1948 9776005  45    Private      College           Married
1949 9780096  41    Private     Bachelor          Divorced
1950 9795905  42    Private       HSgrad              Absent
1951 9805191  23    Private      College              Absent
1952 9813952  32    Private     Bachelor           Married
1954 9815798  27    Private      College              Absent
```

Figure 9. Identifying and removing outliers

```
dim (loan.out)
[1] 1850   13
```

B. Relevant Analysis:

Many of the attributes in the data may be redundant. Correlation analysis can be used to identify whether any two given attributes are statistically related. For numerical attributes, we can evaluate the correlation between two attributes by computing the correlation coefficient. There are three methods for finding correlation coefficient namely Pearson's method, Spearman method and Kendall method.

```
loan.cor<-subset (loan.out, select=c (Age, Income, Hours))
plot (loan.cor$Age, cor$Income)
```

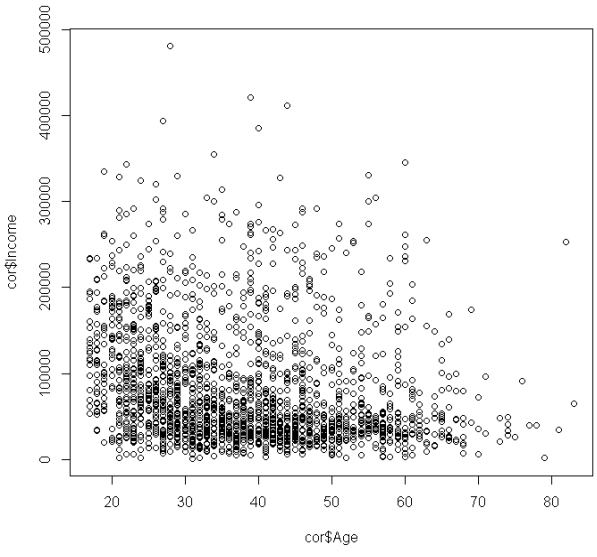


Figure 10. Relevant Analysis

```
plot (loan.cor$Age, cor$Hours)
```

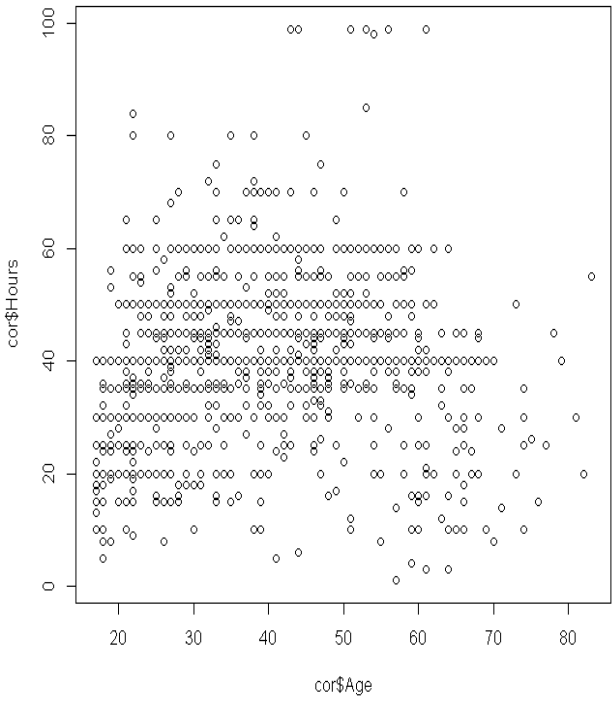


Figure 10.1

```
pairs (loan.cor)
```

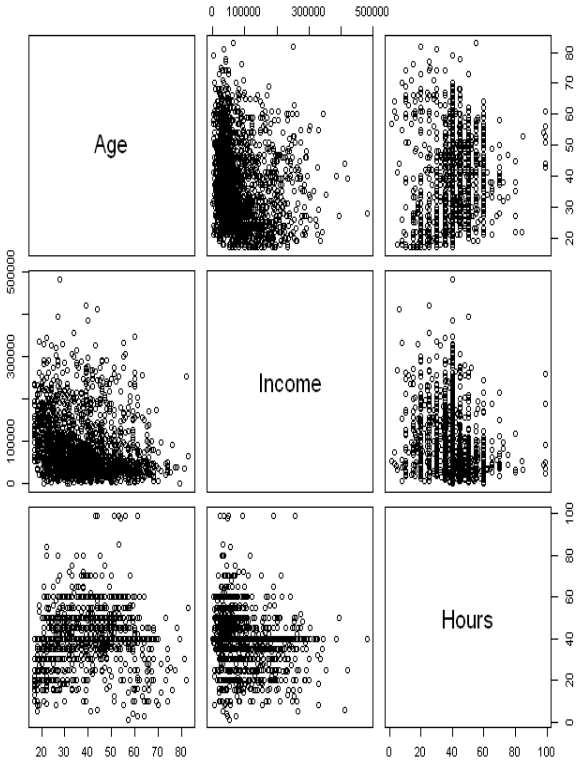


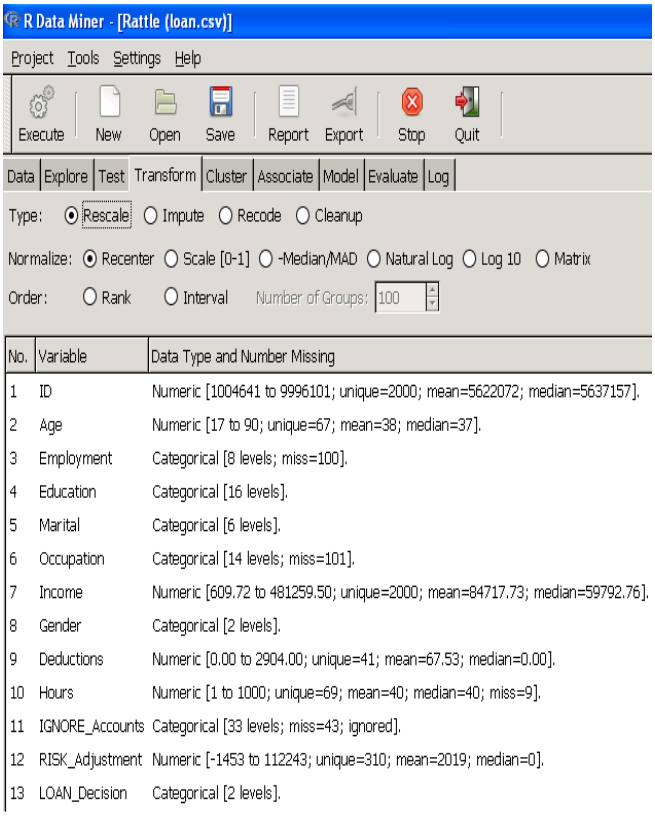
Figure 10.2

```
cor (loan.cor)
```

C. Data Transformation:

In data transformation, data are transformed or consolidated into forms appropriate for mining process. We have various transformations supported by Rattle. The R data miner has a transform tab shown below.

© 2010, IJARCS All Rights Reserved



No.	Variable	Data Type and Number Missing
1	ID	Numeric [1004641 to 9996101; unique=2000; mean=5622072; median=5637157].
2	Age	Numeric [17 to 90; unique=67; mean=38; median=37].
3	Employment	Categorical [8 levels; miss=100].
4	Education	Categorical [16 levels].
5	Marital	Categorical [6 levels].
6	Occupation	Categorical [14 levels; miss=101].
7	Income	Numeric [609.72 to 481259.50; unique=2000; mean=84717.73; median=59792.76].
8	Gender	Categorical [2 levels].
9	Deductions	Numeric [0.00 to 2904.00; unique=41; mean=67.53; median=0.00].
10	Hours	Numeric [1 to 1000; unique=69; mean=40; median=40; miss=9].
11	IGNORE_Accounts	Categorical [33 levels; miss=43; ignored].
12	RISK_Adjustment	Numeric [-1453 to 112243; unique=310; mean=2019; median=0].
13	LOAN_Decision	Categorical [2 levels].

Figure 11. Data Transformation

Data can be transformed by generalizing it to higher-level concepts. For example numeric values of the attribute age can be generalized to discrete range such as “young”, “middle_age”, “senior”. We have two primary transformation techniques namely **normalization and recoding**.

Normalization involves

(a). Data re-center
(b). Data scaling[0-1]
(c). Normalizing with Median/MAD
(d). Normalizing with Natural log

a. Data re-center:

This approach subtracts the mean value of the variable from each observation’s value of variables (to re-center the variable) and then divide the values by their standard deviation, which rescales the value back to a range within a few integer values around zero.

```
library (rattle)
td<-data2
td$RRC_Income<- scale (td$Income)
```

b. Data scaling [0-1]:

This approach simply recodes the data so that all values are between o and 1. This is done by subtracting minimum value from the variable’s value for each observation and then dividing by the difference between the minimum and maximum values.

```
library (reshape)
td$R01_Deductions<-rescaler (td$Deductions, “range”)
```

c. Normalizing with Median/MAD:

In this approach instead of using mean and standard deviation, we subtract the median and divide by median absolute deviation (MAD)

```
library (reshape)
td$RMD_Hours<-rescaler (td$Deductions, “robust”)
```

d. Normalizing with Natural log:

This is called logarithmic transformation. Here the transformation is done by using natural logarithm function. This is particularly used when the variable has outliers with extremely large values.

```
td$RLG_Age<-log (td$Deductions)
```

If the result is “infinite” that is log(0), then those values are treated as NAs.

```
td$RLG_Age [td$RLG_Age == -inf]<-NA
```

e. Recoding involve:

a) Grouping or binning attribute values by Quantile (equal count)
b) Grouping attribute values by KMeans
c) Grouping attribute values by Equal Widths

Out of the above transformation techniques we perform re-centering and grouping attribute values by Quantile method.

```
Library (rattle)
td<-data2
td$RRC_Income<- scale (td$Income)
td$BQ3_Age<-binning (td$Age, bins=3, method="quantile")
td$Income=NULL
td$Age=NULL
td$ID=NULL
td$IGNORE_Accounts=NULL
td
```

Figure 12

D. Data Reduction:

Data reduction techniques can be applied to obtain a reduced representation of the dataset that is much smaller in volume but maintains the integrity of the original data. Dimensionality reduction is one of the data reduction techniques which encodes or transforms data to obtain a reduced representation of original data [8]. There are two types of dimensionality reductions. If the original data can

© 2010, IJARCS All Rights Reserved

be reconstructed from the compressed data without any loss of information, the data reduction is called **lossless.** Similarly if we can reconstruct data with loss of information is called lossy.

Principal component analysis is one of the popular effective methods of lossy dimensionality reduction. PCA can provide insights into the importance of variables in explaining the variation found within a dataset. A principal component is a numeric linear combination of the values of other variables in the dataset that captures maximal variation in the data. The main goals of PCA is to

(a). Identify how different variables work together to create the dynamics of the system
(b). Reduce the dimensionality of the data
(c). Filter some of the noise in the data
(d). Compress the data
(e). Prepare the data for further analysis using other techniques.

In PCA, uncorrelated PC's are extracted by linear transformations of the original variables so that the first few PC's contain most of the variations in the original dataset.

These PCs are extracted in decreasing order of importance so that the first PC accounts for as much of the variation as possible and each successive component accounts for a little less.

Bi-plot display is a visualization technique for investigating the inter-relationships between the observations and variables in multivariate data

PC scores are the derived composite scores computed for each observation based on the eigenvectors for each PC.
PC loadings are correlation coefficients between the PC scores and the original variables.

```
data1<-loan.out
data2<-cbind(data1$Age,data1$Employment,data1$Education,data1$Marital,data1$Occupation,data1$Income,data1$Gender,data1$Deduction,data1$Hours,data1$RISK_Adjustment,data1$LOAN_Decision)
colnames(data2) <- c("age","emp","edu","mtrl","occu","inc","gen","deduc","hr","risk","decision")
duplicated (data2)
```

```
FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE TRUE TRUE
```

Figure 13. Data Reduction

```
data2<-data2 [! duplicated (data2),]
pairs (data2)
```

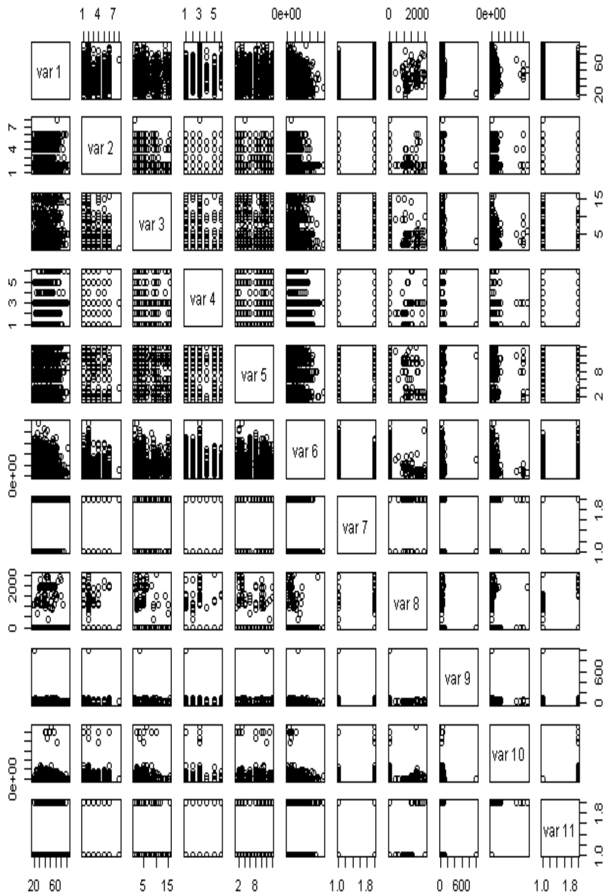


Figure 13.1

```
arc.pca1 <- princomp (data2, scores=TRUE, cor=TRUE)
summary (arc.pca1)
plot (arc.pca1)
```

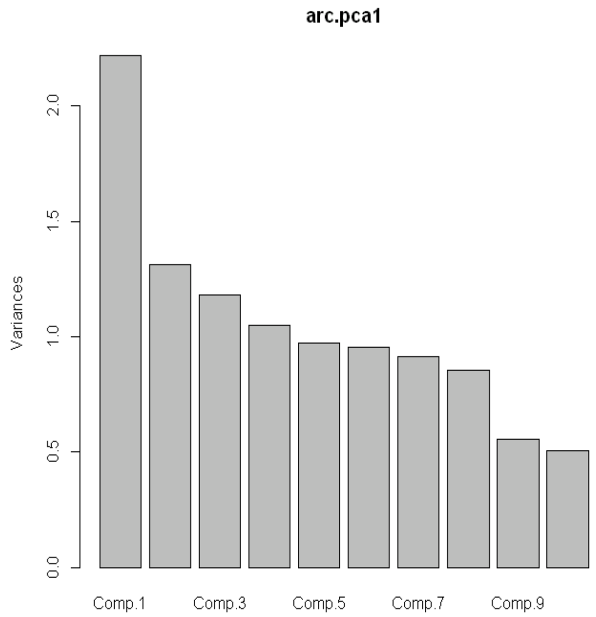


Figure 13.2

```
biplot (arc.pca1)
```

© 2010, IJARCS All Rights Reserved

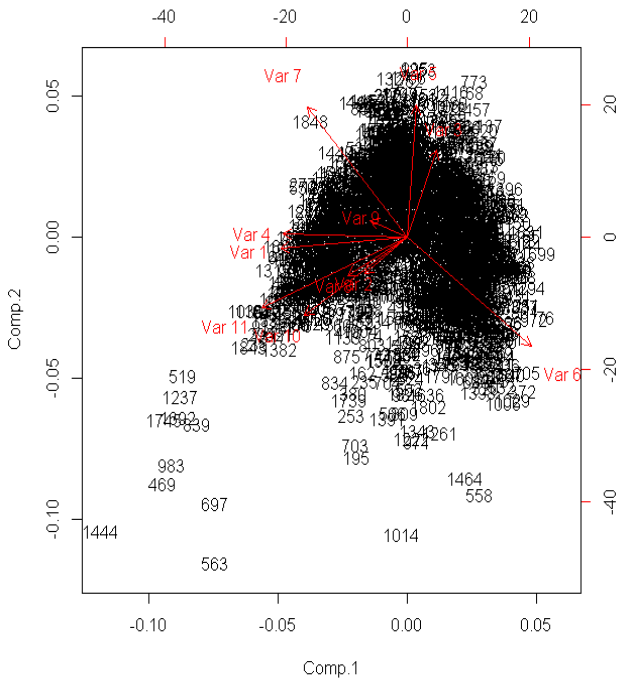


Figure 13.3

arc.pca1$loadings

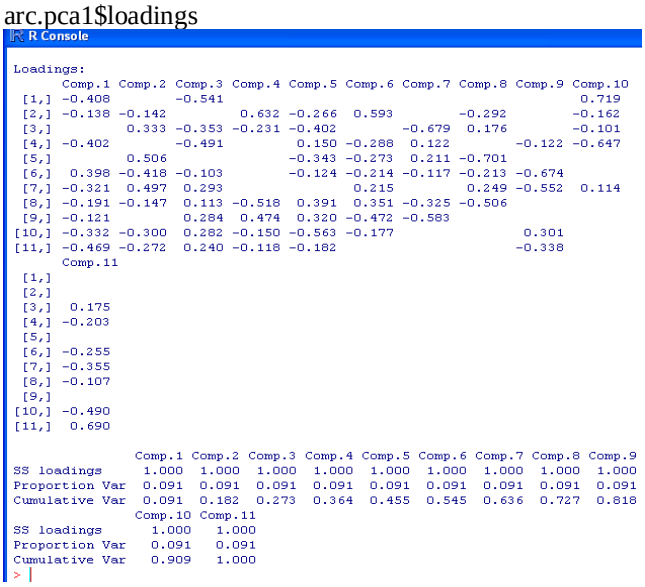


Figure 13.4

IV. CLASSIFICATION AND PREDICTION STAGE

Decision tree induction is the learning of decision trees from class-labeled training tuples. A **decision tree** model is one of the most common data mining models. It is popular because the resulting model is easy to understand. It is a flow chart like tree structure, where each **internal node** denotes a test on the attribute, each **branch** represents an outcome of the test, and each **leaf node** holds a class label. The top most node in a tree is the **root** node.

We can do classification in R data miner, which is very simple and easy to understand. The algorithms used in the decision tree induction are recursive partitioning approach. There are two algorithms namely traditional and conditional. Here we use traditional algorithm to build our decision tree. It is implemented in the rpart package. It is comparable to CART. The conditional tree algorithm is implemented in the party package. It builds trees in a conditional inference framework. In R data miner firstly we have to load the cleaned dataset. This is shown below.

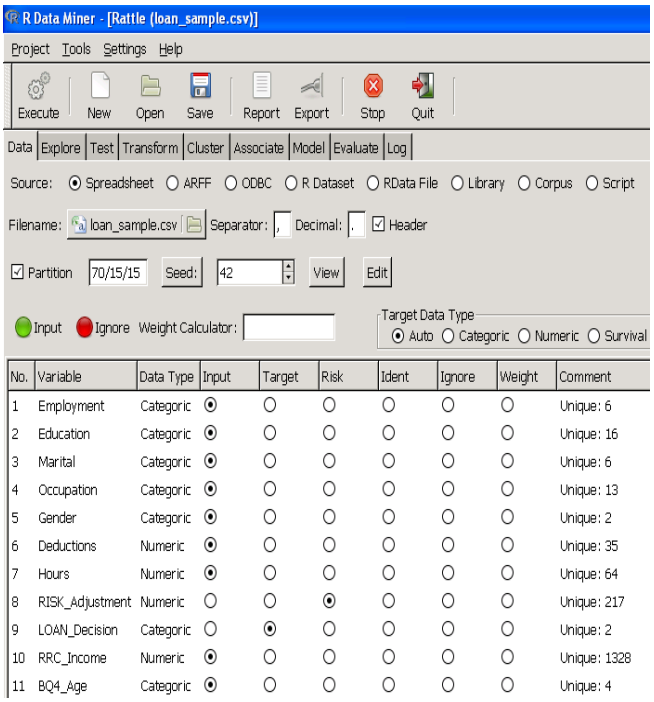


Figure 14. Classification and Prediction

To generate a model R data miner provides model tab which has six model options namely

a. Tree
b. Forest
c. Boost
d. SVM
e. Linear
f. Neural net

Decision tree and classification rules are produced by clicking tree option as we chosen to adapt decision tree based induction method.

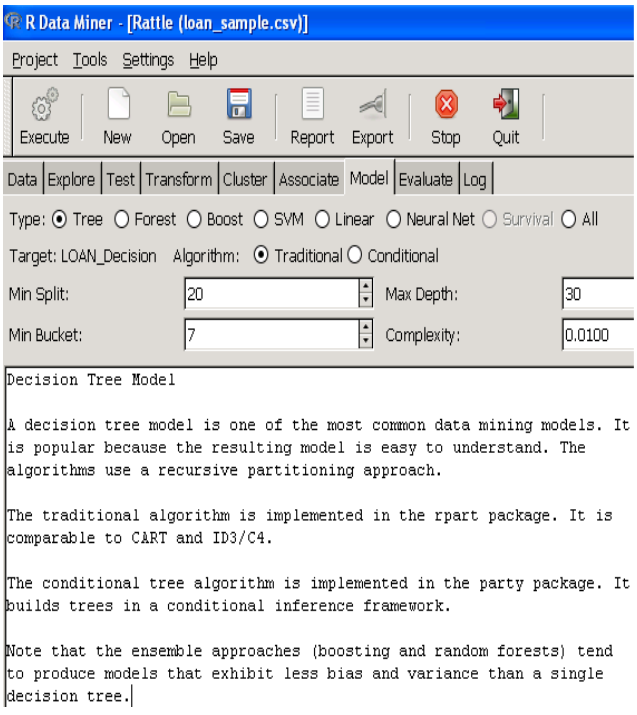


Figure 14.1

© 2010, IJARCS All Rights Reserved

The decision tree classification model can be built using either of the two packages [9]:

a) "rpart"
b) "tree"

Here the classification model is built using "rpart" package. The summary of the decision tree classification model in R data miner is shown below.

```
Summary of the Decision Tree model for Classification (built using 'rpart'):

n=1314 (84 observations deleted due to missingness)

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

 1) root 1314 313 risk (0.76179604 0.23820396)
   2) Marital=Absent,Divorced,Married-spouse-absent,Unmarried,Widowed 722  47 risk (0.93490305 0.06509695) *
   3) Marital=Married 592 266 risk (0.55067568 0.44932432)
     6) Occupation=Cleaner,Farming,Machinist,Repair,Service,Transport 285  71 risk (0.75087719 0.24912281) *
     7) Occupation=Clerical,Executive,Professional,Protective,Sales,Support 307 112 safe (0.36482085 0.63517915)
      14) Education=College,HSGrad,Tr10,Tr12,Tr5t6,Tr7t8,Tr9 129  61 risk (0.52713178 0.47286822)
        28) BQ4_Age=[17,28],(28,37] 39  11 risk (0.71794872 0.28205128) *
        29) BQ4_Age=(37,48],(48,90] 90  40 safe (0.44444444 0.55555556)
          58) Employment=Consultant 15   3 risk (0.80000000 0.20000000) *
          59) Employment=Private,PSFederal,PSLocal,PSState,SelfEmp 75  28 safe (0.37333333 0.62666667)
           118) RRC_Income< -0.789549 20   6 risk (0.70000000 0.30000000) *
           119) RRC_Income>=-0.789549 55  14 safe (0.25454545 0.74545455) *
      15) Education=Associate,Bachelor,Doctorate,Master,Professional,Vocational,Yr11 178  44 safe (0.24719101 0.75280899) *

Classification tree:
rpart(formula = LOAN_Decision ~ ., data = crs$dataset[crs$train,
    c(crs$input, crs$target)], method = "class", parms = list(split = "information"),
    control = rpart.control(usesurrogate = 0, maxsurrogate = 0))

Variables actually used in tree construction:
[1] BQ4_Age    Education  Employment Marital    Occupation RRC_Income

Root node error: 313/1314 = 0.2382

n=1314 (84 observations deleted due to missingness)

        CP nsplit rel error  xerror     xstd
1 0.192588      0   1.00000 1.00000 0.049334
2 0.027157      2   0.73482 0.75719 0.044529
3 0.025559      5   0.65176 0.75719 0.044529
4 0.010000      6   0.62620 0.74441 0.044233
```

Figure 14.2

V. DECISION TREE GENERATED

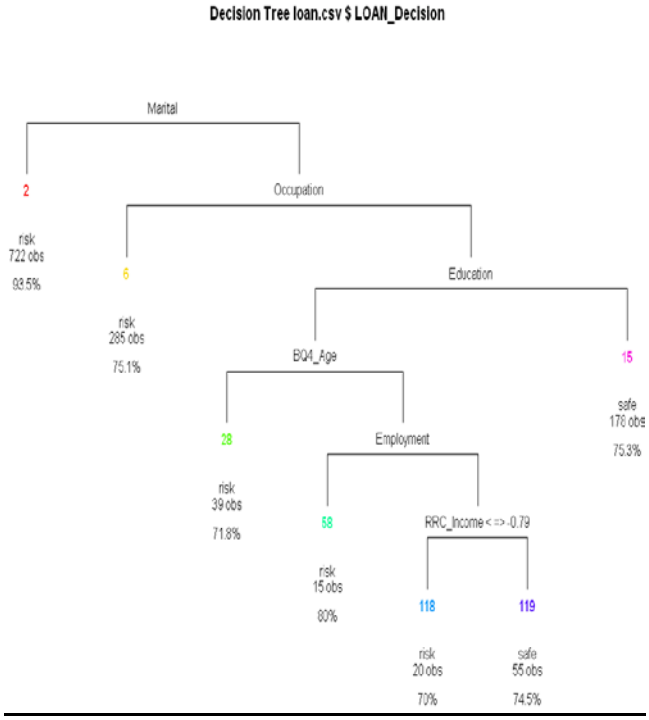


Figure 15. Decision Tree

VI. CONCLUSION

Thus by performing the above stages we can predict which loan applicants are "risky" and which are "safe" for a loan dataset. This prediction is very useful in taking decision for a bank whether to issue or reject loans for any individual [6][10].

VII. REFERENCES

[1] Qasem A. Al-Radaideh, Ahmad Al Ananbeh, and Emad M. Al-Shawakfa, "A Classification Model For Predicting The Suitable Study Track For School Students Classification", Volume8, Issue2, IJRRAS.

[2] Qasem A. Al-Radaideh, Emad M. Al-Shawakfa, and Mustafa I. Al-Najjar, "Mining Student Data Using Decision Trees", The 2006 International Arab Conference on Information Technology (ACIT'2006).

[3] Graham J Williams, "Rattle: A Data Mining GUI for R", Contributed Research Articles.

[4] Nevine M. Labib, and Michael N. Malek, "Data Mining for Cancer Management in Egypt

[5] Case Study: Childhood Acute Lymphoblastic Leukemia" World Academy of Science, Engineering and Technology 8 2005.

[6] Burcu Kalender, "Analysis of loan customers' characteristics", Data Minig Project.

[7] Graham Williams, "Data Mining with Rattle and R", The Art of Extracting Data for Knowledge Discovery, Springer.

[8] Jiawi Han & Micheline Kamber, "Data Mining- Concepts and Techniques", Haracourt India.

[9] Arun Pujari, "Data Mining Techniques", University Press.

[10] Margaret H Dunham, "Data Mining Introductory And Advanced Topics", Pearson Education.

[11] Jean-Marc Adamo, Springer, "Data Mining for Association Rules and Sequential Patterns".

© 2010, IJARCS All Rights Reserved