Loan Eligibility Prediction

1 INTRODUCTION:

1.1 Overview :

With the enhancement in the banking sector lots of people are applying for bank loans but the bank has its limited assets which it has to grant to limited people only, so finding out to whom the loan can be granted which will be a safer option for the bank is a typical process.

1.2 Purpose:

In this project we try to reduce this risk factor behind selecting the safe person so as to save lots of bank efforts and assets. This is done by mining the Data of the previous records of the people to whom the loan was granted before and on the basis of these records the machine was trained using the machine learning model which give the most accurate result. The main objective of this project is to predict whether assigning the loan to particular person will be safe or not.

2.LITERATURE SURVEY:

2.1 Existing Problems:

Data mining is the process of analyzing data from different perspectives and extracting useful knowledge from it. Different data mining techniques include classification, clustering, association rule mining, prediction and sequential patterns, neural networks, regression etc. Classification is the most commonly applied data mining technique, which employs a set of pre-classified examples to develop a model that can classify the population of records at large.In classification, a training set is used to build the model as the classifier which can classify the data items into its appropriate classes. A test set is used to validate the model.

2.2 Proposed solution:

Loan Eligibility Prediction

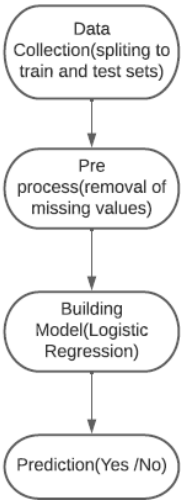
Logistic Regression:

Logistic Regression is one of the most popular machine learning algorithm, which is used for predicting the categorical dependent variable using a given set of dependent variable. Therefore the outcome must be a categorical ordiscrete value. It can be either Yes or No,True or falseinear Regression is used for solving Regression problems, whereas **Logistic regression is used for solving the classification problems**.Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.

3. Theoritical Analysis:

3.1 Block Diagram:

The steps involved in Building the data model is depicted below:



3.2 Software Designing:

The software used for this project are :

- WEKA 3.8.5,
- Java version 10

Loan Eligibility Prediction

- Eclipse neon IDE.

4.EXPERIMENTAL INVESTIGATION:

4.1 Data Collection:

The dataset collected for predicting loan default customers is predicted into Training set and testing set. Generally 60:40 ratio is applied to split the training set and testing set. The data model which was created using Logistic regression is applied on the training set and based on the test result accuracy, Test set prediction is done.attributes

Variable	Description
Loan_ID	Unique Loan ID
Gender	Male/ Female
Married	Applicant married (Y/N)
Dependents	Number of dependents
Education	Applicant Education (Graduate/ Under Graduate)
Self_Employed	Self employed (Y/N)
ApplicantIncome	Applicant income
CoapplicantIncome	Coapplicant income
LoanAmount	Loan amount in thousands
Loan_Amount_Term	Term of loan in months
Credit_History	credit history meets guidelines
Property_Area	Urban/ Semi Urban/ Rural
Loan_Status	Loan approved (Y/N)

4.2 Pre processing:

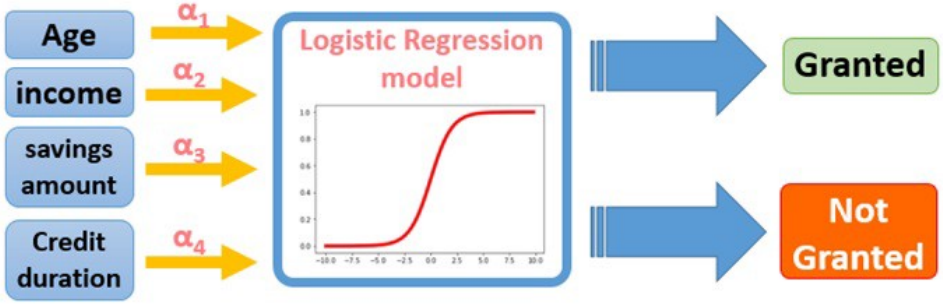
The data which was collected might contain missing values that may lead to inconsistency. To gain better results data need to be preprocessed so as to improve the efficiency o the algorithm.

4.3 Buliding Model using Logistic Regression Model:

For predicting the loan defaulter's and non defaulter's problem Logistic Regression algorithm is used. The purpose of this algorithm is to find a plane that separates two types. Y variable belongs to 1 or 0.

Loan Eligibility Prediction

5.FLOWCHART:

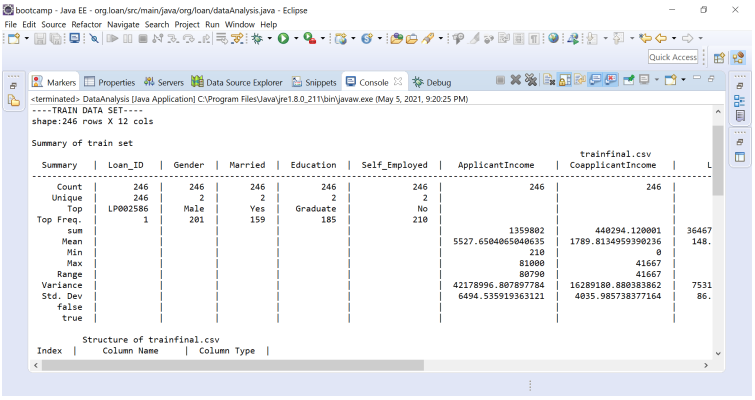


6.RESULT:

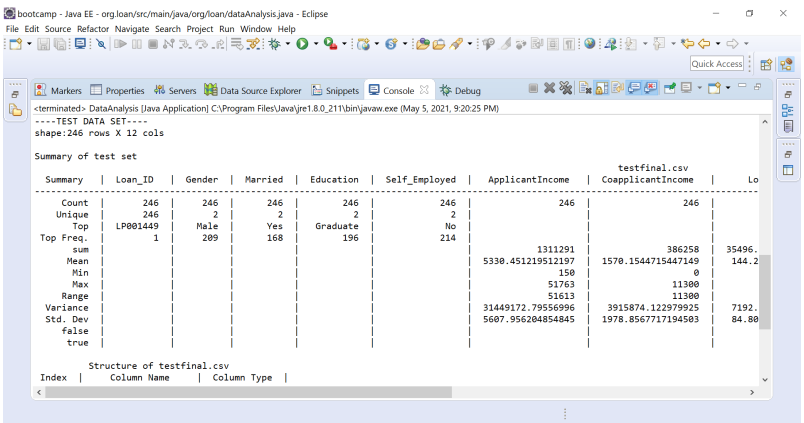
The accuracy for the built model is 85%, Precision is 95%.

A)ECLIPSE RESULT:

Data Analysis Output:

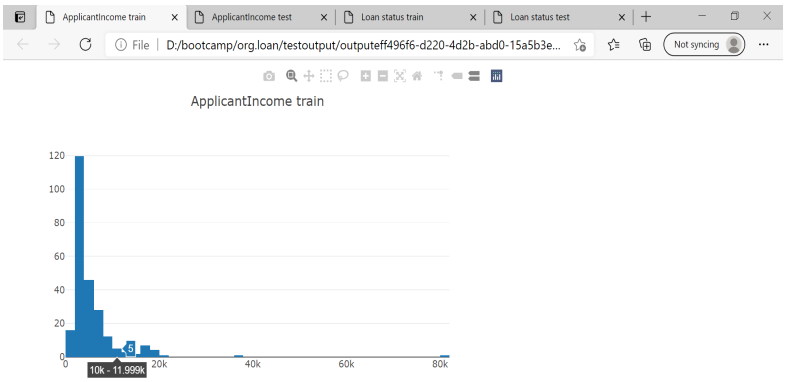


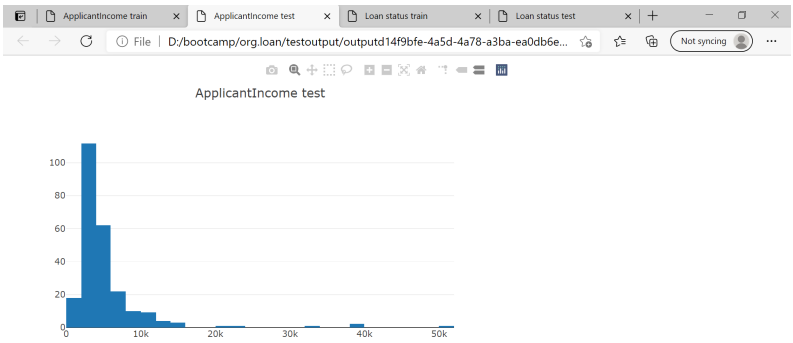
```
bootcamp - Java EE - org.loan/src/main/java/org/loan/dataAnalysis.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Markers | Properties | Servers | Data Source Explorer | Snippets | Console | Debug
<terminated> DataAnalysis [Java Application] C:\Program Files\Java\jre1.8.0_211\bin\javaw.exe (May 5, 2021, 9:20:25 PM)
----TRAIN DATA SET----
shape:246 rows X 12 cols

Summary of train set

                                                                                                                trainfinal.csv
  Summary   |  Loan_ID   |  Gender  |  Married  |  Education  |  Self_Employed  |  ApplicantIncome    |  CoapplicantIncome   |      L
------------------------------------------------------------------------------------------------------------------------------------
     Count  |        246 |      246 |       246 |         246 |             246 |                 246 |                  246 |
    Unique  |        246 |        2 |         2 |           2 |               2 |                     |                      |
       Top  |  LP002586  |     Male |       Yes |    Graduate |              No |                     |                      |
  Top Freq. |          1 |      201 |       159 |         185 |             210 |                     |                      |
       sum  |            |          |           |             |                 |             1359802 |       440294.120001  |   36467
      Mean  |            |          |           |             |                 |  5527.6504065040635 |  1789.813495939236   |   148.
       Min  |            |          |           |             |                 |                 210 |                    0 |
       Max  |            |          |           |             |                 |               81000 |                41667 |
     Range  |            |          |           |             |                 |               80790 |                41667 |
  Variance  |            |          |           |             |                 |  42178996.807897784 |  16289180.880383862  |    7531
  Std. Dev  |            |          |           |             |                 |   6494.535919363121 |  4035.985738377164   |    86.
     false  |            |          |           |             |                 |                     |                      |
      true  |            |          |           |             |                 |                     |                      |

        Structure of trainfinal.csv
 Index  |      Column Name      |  Column Type  |
```

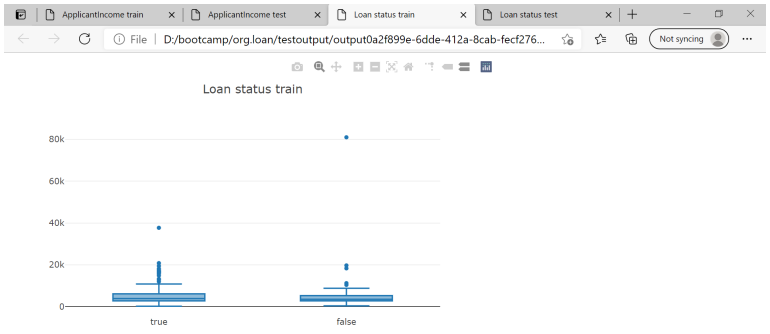


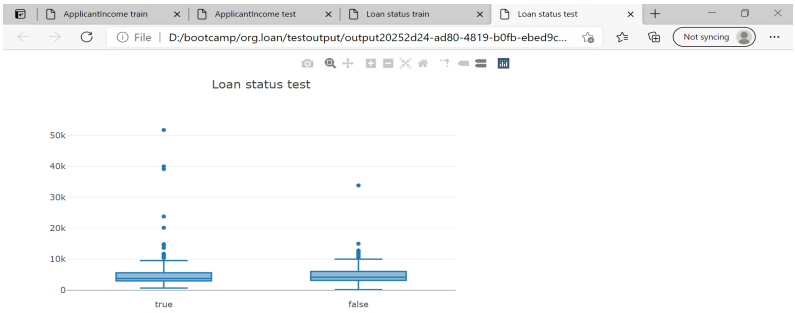
```
bootcamp - Java EE - org.loan/src/main/java/org/loan/dataAnalysis.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Markers | Properties | Servers | Data Source Explorer | Snippets | Console | Debug
<terminated> DataAnalysis [Java Application] C:\Program Files\Java\jre1.8.0_211\bin\javaw.exe (May 5, 2021, 9:20:25 PM)
----TEST DATA SET----
shape:246 rows X 12 cols

Summary of test set

                                                                                                                testfinal.csv
  Summary   |  Loan_ID   |  Gender  |  Married  |  Education  |  Self_Employed  |  ApplicantIncome    |  CoapplicantIncome   |      Lo
------------------------------------------------------------------------------------------------------------------------------------
     Count  |        246 |      246 |       246 |         246 |             246 |                 246 |                  246 |
    Unique  |        246 |        2 |         2 |           2 |               2 |                     |                      |
       Top  |  LP001449  |     Male |       Yes |    Graduate |              No |                     |                      |
  Top Freq. |          1 |      209 |       168 |         196 |             214 |                     |                      |
       sum  |            |          |           |             |                 |             1311291 |       386258         |   35496.
      Mean  |            |          |           |             |                 |  5330.451219512197  |  1570.1544715447149  |   144.2
       Min  |            |          |           |             |                 |                 150 |                    0 |
       Max  |            |          |           |             |                 |               51763 |                11300 |
     Range  |            |          |           |             |                 |               51613 |                11300 |
  Variance  |            |          |           |             |                 |  31449172.79556996  |  3915874.122979925   |    7192.
  Std. Dev  |            |          |           |             |                 |   5607.956204854845 |  1978.8567717194503  |    84.80
     false  |            |          |           |             |                 |                     |                      |
      true  |            |          |           |             |                 |                     |                      |

        Structure of testfinal.csv
 Index  |      Column Name      |  Column Type  |
```

Loan Eligibility Prediction



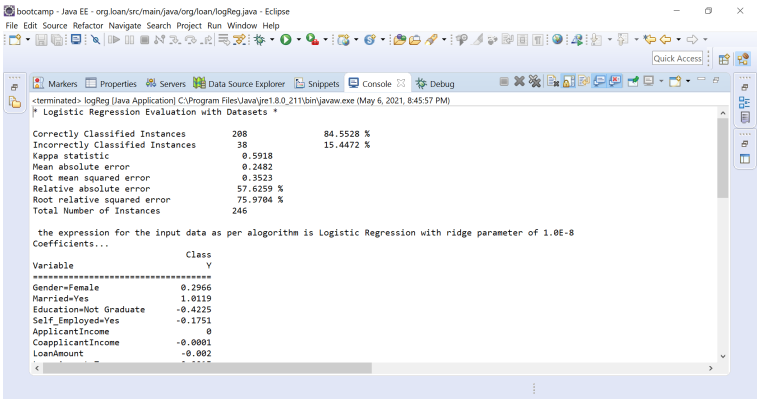


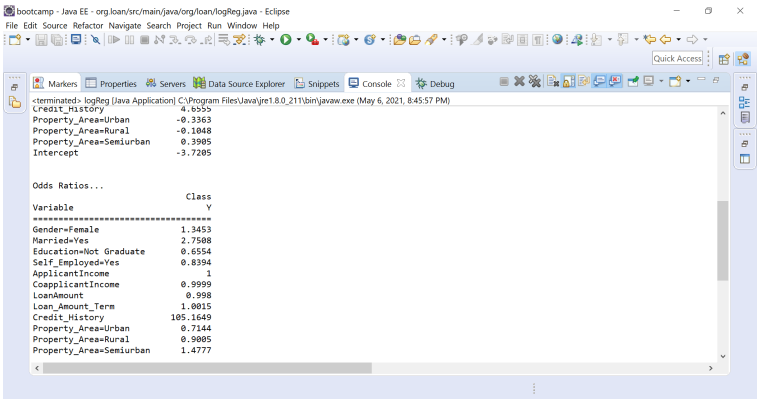


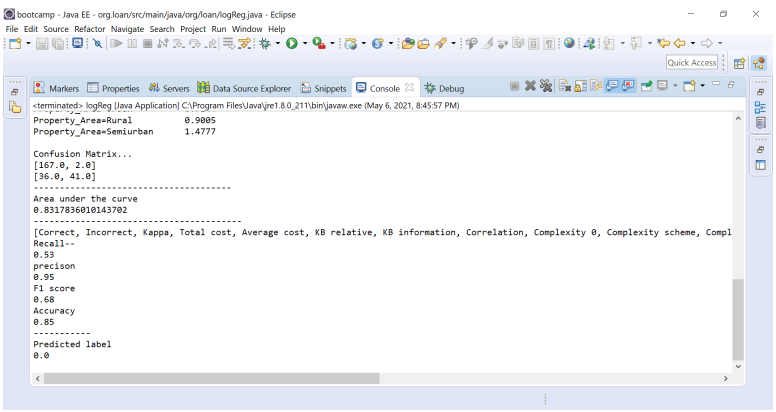


Loan Eligibility Prediction

Logistic Regression Output:



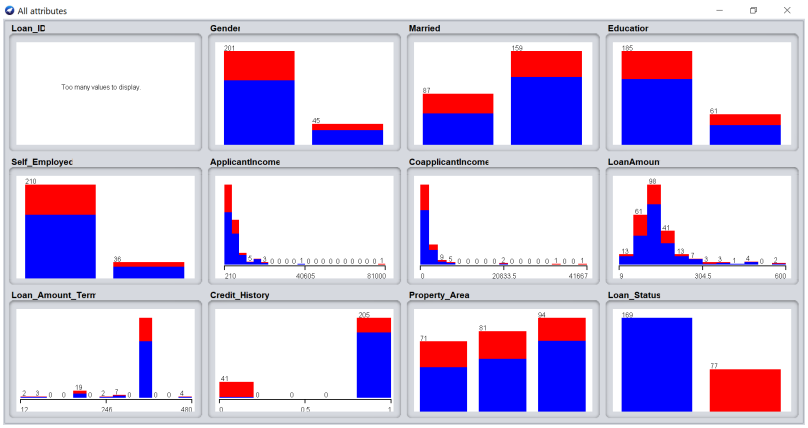


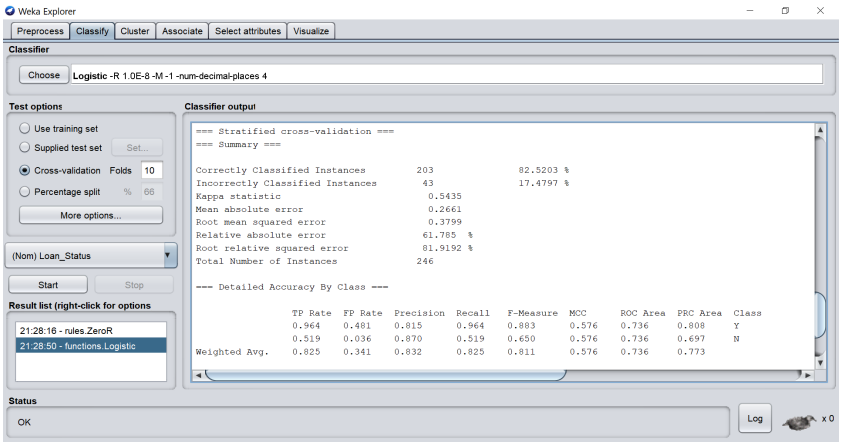


B)WEKA GUI Result:

TRAIN DATASET OUTPUT:

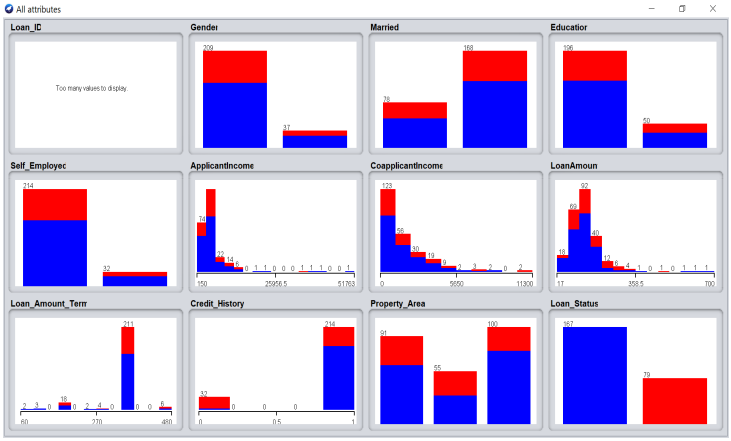
Loan Eligibility Prediction

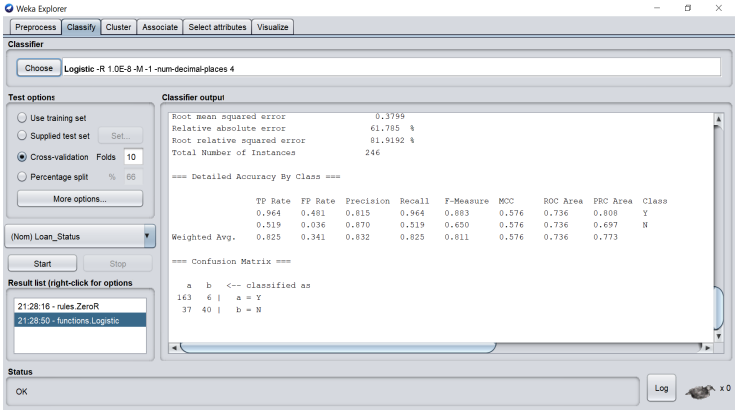




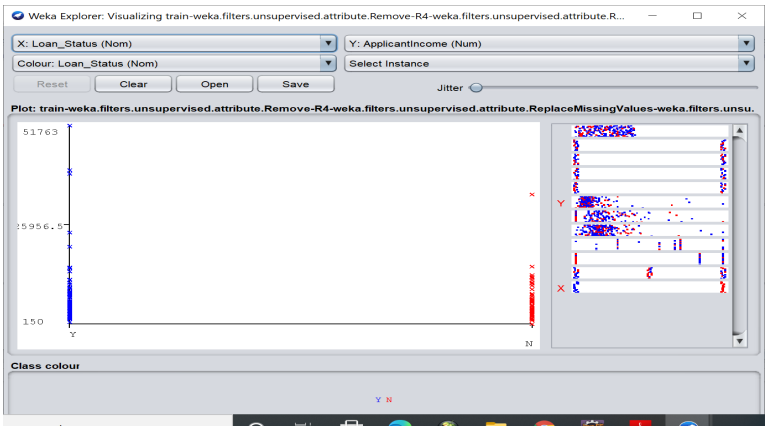
Loan Eligibility Prediction

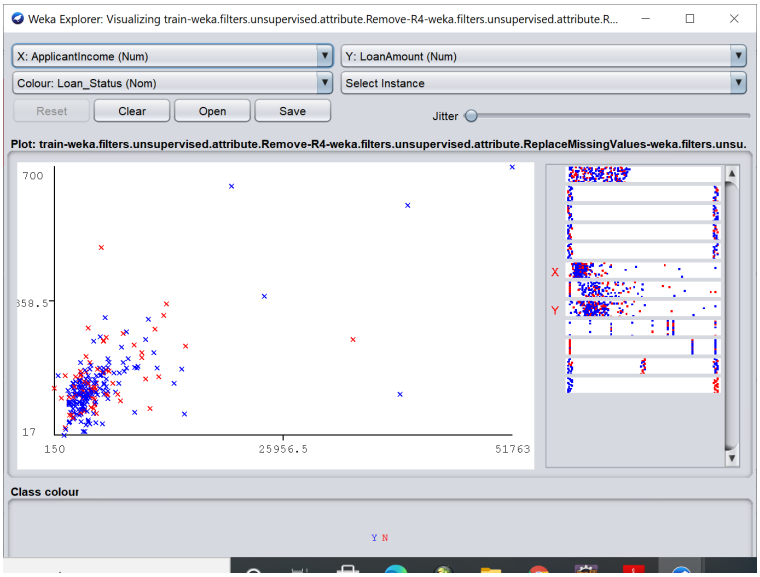
TEST DATASET OUTPUT:





Loan Eligibility Prediction





7.ADVANTAGES AND DISADVANTAGES:

7.1 Advantages:

- Compared to othe algorithm,Logistic regression will provide probablility prediction along with the classification result.
- Logistic regression can be used for large set of data.
- One of the great advantages of Logistic Regression is that when you have a complicated linear problem and not a whole lot of data it's still able to produce pretty useful predictions.

7.2 Disadvantages:

- Data preparation can be tedious in Logistic Regression as both scaling and

normalization are important requirements of Logistic Regression.

- Logistic Regression is not immune to missing data unlike some other machine learning models such as decision trees and random forests which are based on trees.

8.APPLICATOIN:

It can be used for banking sectors for predicting the eligibility of loan for the customers and predicting the customers's loan status whether he will be able to pay the loan or notby using the previous records.

9.CONCLUSION:

The analytical process started from data cleaning and processing, Missing value imputation with micepackage, thenexploratory analysis and finally model building and evaluation. The best accuracy on public test set is 0.78. Most of the Time, Applicants with high income sanctioning low amount is to more likely get approved which make sense, more likely to pay back their loans.

10.BIBILOGRAPHY:

http://www.ijetjournal.org
https://www.javatpoint.com/logistic-regression-in-machine-learning
https://holypython.com/log-reg/logistic-regression-pros-cons/

11.APPENDIX:

Source Code:

A) Data Analysis Code:

```
package org.loan;
```

Loan Eligibility Prediction

```
import java.io.IOException;

import tech.tablesaw.api.Table;
import tech.tablesaw.plotly.Plot;
import tech.tablesaw.plotly.components.Figure;
import tech.tablesaw.plotly.components.Layout;
import tech.tablesaw.plotly.traces.BoxTrace;
import tech.tablesaw.plotly.traces.HistogramTrace;

public class dataAnalysis {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		try {
			//Reading the training dataset in a dataframe using Tablesaw
			Table loantrain_data =
Table.read().csv("D:\\bootcamp\\org.loan\\src\\main\\java\\org\\loan\\trainfinal.csv");
			System.out.println("----TRAIN DATA SET----");
			System.out.print("shape:");
			System.out.println(loantrain_data.shape());
			System.out.println();
			System.out.println("Summary of train set");
			System.out.println(loantrain_data.summary());
			System.out.println();
			System.out.println(loantrain_data.structure());

			//Reading the test dataset in a dataframe using tablesaw
			Table loantest_data =
Table.read().csv("D:\\bootcamp\\org.loan\\src\\main\\java\\org\\loan\\testfinal.csv");
			System.out.println();
			System.out.println("----TEST DATA SET----");
			System.out.print("shape:");
			System.out.println(loantest_data.shape());
			System.out.println();
			System.out.println("Summary of test set");
			System.out.println(loantest_data.summary());
			System.out.println();
			System.out.println(loantest_data.structure());
```

Loan Eligibility Prediction

```
                        //histogram of variable ApplicantIncome for training dataset
                        Layout layout1 = Layout.builder().title("ApplicantIncome train").build();
                        HistogramTrace trace1 =
HistogramTrace.builder(loantrain_data.nCol("ApplicantIncome")).build();
                        Plot.show(new Figure(layout1,trace1));

        // Box Plot for variable ApplicantIncome of training data set
                    Layout layout2 = Layout.builder().title(" Loan status train").build();
                    BoxTrace trace2
=BoxTrace.builder(loantrain_data.categoricalColumn("Loan_Status"),
loantrain_data.nCol("ApplicantIncome")).build();
                    Plot.show(new Figure(layout2, trace2));

                  //histogram of variable ApplicantIncome for testing dataset
                        Layout layout3 = Layout.builder().title("ApplicantIncome test").build();
                        HistogramTrace trace3 =
HistogramTrace.builder(loantest_data.nCol("ApplicantIncome")).build();
                        Plot.show(new Figure(layout3,trace3));

        // Box Plot for variable ApplicantIncome of testing data set
                    Layout layout4 = Layout.builder().title(" Loan status test").build();
                    BoxTrace trace4
=BoxTrace.builder(loantest_data.categoricalColumn("Loan_Status"),
loantest_data.nCol("ApplicantIncome")).build();
                    Plot.show(new Figure(layout4, trace4));
                } catch (IOException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }
        }

}
```

B) Logistic Regression Model:

```
package org.loan;
import java.util.Arrays;
```

Loan Eligibility Prediction

```
import weka.classifiers.Classifier;
import weka.classifiers.Evaluation;
import weka.core.Instance;
import weka.core.Instances;
import weka.core.converters.ConverterUtils.DataSource;
public class logReg {
	public static void main(String[] args) throws Exception {
		DataSource source = new
DataSource("D:\\bootcamp\\org.loan\\src\\main\\java\\org\\loan\\trainfinal1.arff");
		Instances dataset = source.getDataSet();
		dataset.setClassIndex(dataset.numAttributes()-1);

	Classifier classifier = new weka.classifiers.functions.Logistic();
	DataSource source1 = new
DataSource("D:\\bootcamp\\org.loan\\src\\main\\java\\org\\loan\\testfinal1.arff");
	Instances dataset1 = source.getDataSet();
	dataset1.setClassIndex(dataset1.numAttributes()-1);

	classifier.buildClassifier(dataset);
	//System.out.println(classifier);

	Evaluation eval = new Evaluation(dataset);
		eval.evaluateModel(classifier, dataset1);
		/** Print the algorithm summary */
		System.out.println("* Logistic Regression Evaluation with Datasets *");
		System.out.println(eval.toSummaryString());
		System.out.print(" the expression for the input data as per alogorithm is ");
		System.out.println(classifier);

		double confusion[][] = eval.confusionMatrix();
		System.out.println("Confusion Matrix...");
		for (double[] row : confusion)
			System.out.println( Arrays.toString(row));
	  System.out.println("-----------------------------------");

	  System.out.println("Area under the curve");
	  System.out.println(eval.areaUnderROC(0));
	  System.out.println("-----------------------------------");
```

Loan Eligibility Prediction

```
        System.out.println(eval.getAllEvaluationMetricNames());;


        System.out.println("Recall--");
        System.out.println(Math.round(eval.recall(1)*100.0)/100.0);

        System.out.println("precison");
        System.out.println(Math.round(eval.precision(1)*100.0)/100.0);

        System.out.println("F1 score");
        System.out.println(Math.round(eval.fMeasure(1)*100.0)/100.0);

        System.out.println("Accuracy");
        double acc = eval.correct()/(eval.correct()+eval.incorrect());
        System.out.println(Math.round(acc*100.0)/100.0);


        System.out.println("----------");
        Instance predict = dataset1.get(120);

        double value = classifier.classifyInstance(predict);

        System.out.println("Predicted label");
        System.out.println(value);



            }

}
```