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.

Loan Eligibility Prediction

A test set is used to validate the model.

2.2 Proposed solution :

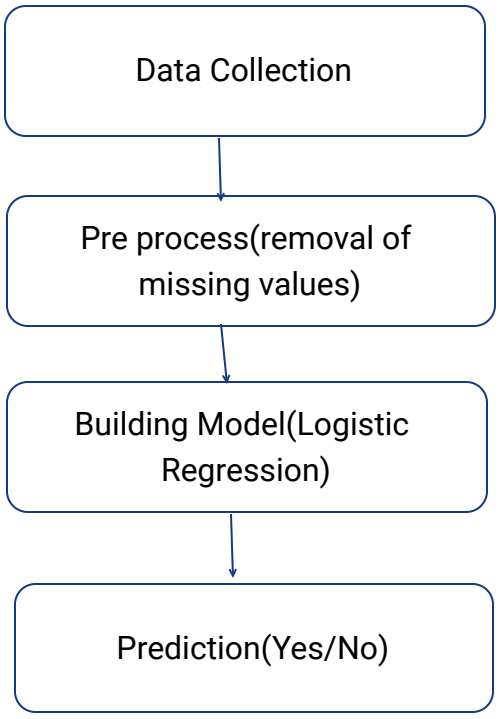
Logistic Regression:

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:



Loan Eligibility Prediction

3.2 Software Designing:

The software used for this project are :

➤ WEKA 3.8.5

➤ Java version 10

➤ Eclipse neon IDE.

4.EXPERIMENTAL INVESTIGATION:

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)
Applicant Income	Applicant income
Coapplicant Income	Coapplicant Income
Loan Amount	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)

Loan Eligibility Prediction

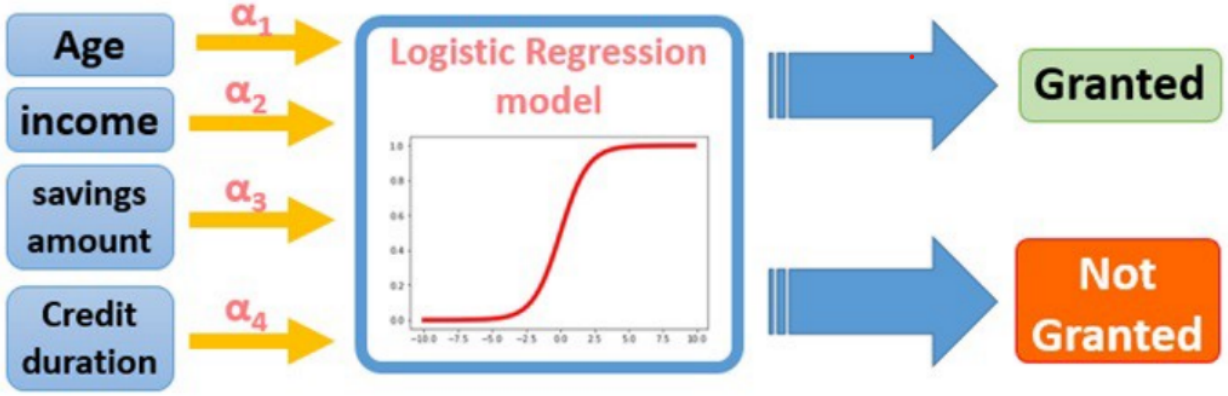
4.2 Pre processing:

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.

5.FLOWCHART:

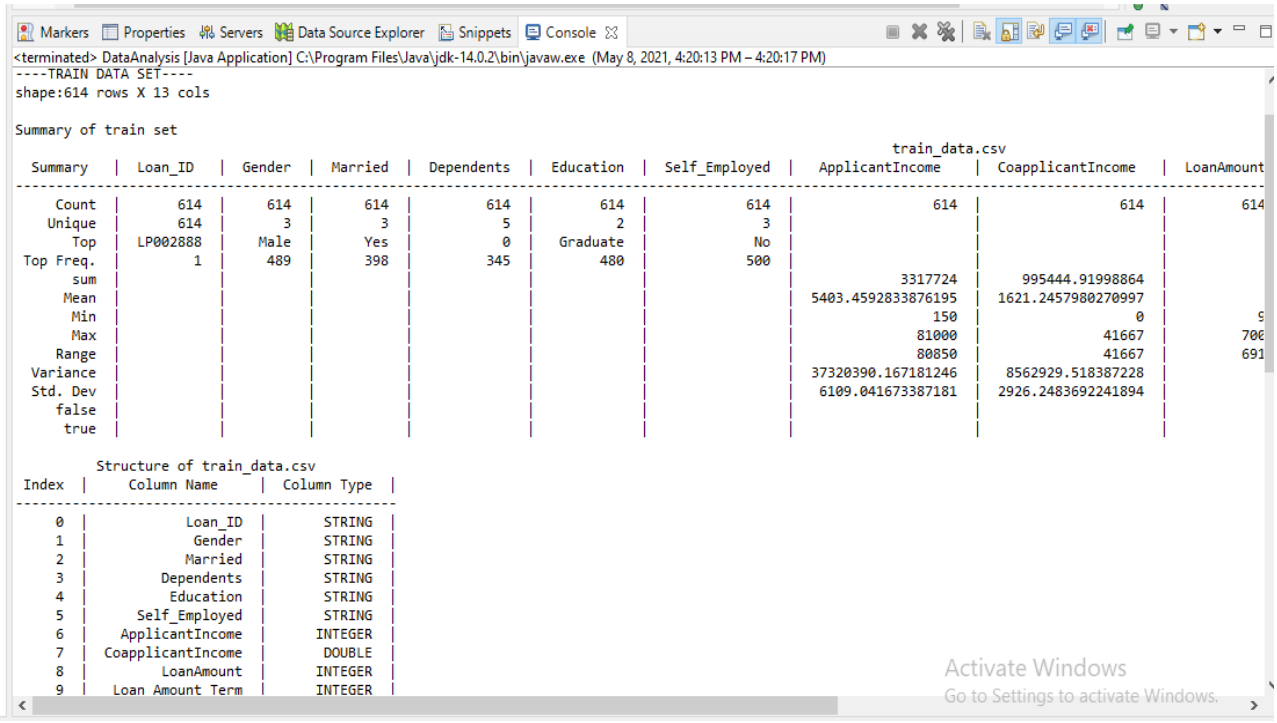


6.RESULT:

The accuracy for the built model is 81%, Precision is 91%.

A)ECLIPSE RESULT:

Loan Eligibility Prediction



Markers | Properties | Servers | Data Source Explorer | Snippets | Console

<terminated> DataAnalysis [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe (May 8, 2021, 4:20:13 PM – 4:20:17 PM)

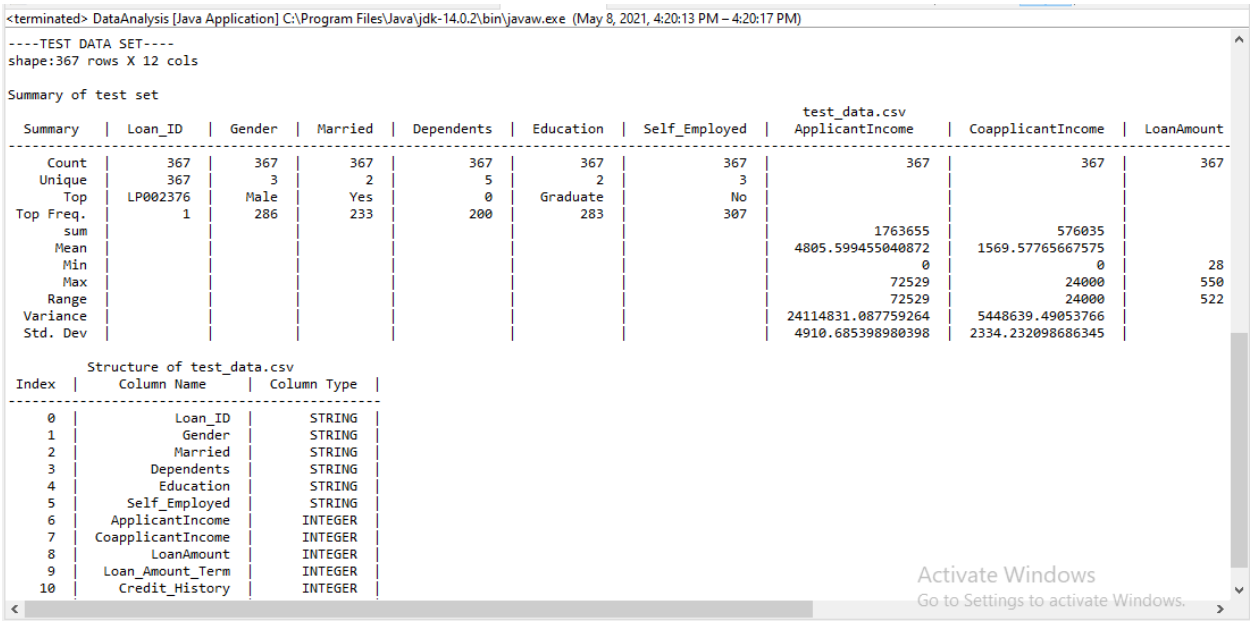
```
----TRAIN DATA SET----
shape:614 rows X 13 cols

Summary of train set
```

train_data.csv

Summary	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount
Count	614	614	614	614	614	614	614	614	614
Unique	614	3	3	5	2	3			
Top	LP002888	Male	Yes	0	Graduate	No			
Top Freq.	1	489	398	345	480	500			
sum							3317724	995444.91998864	
Mean							5403.4592833876195	1621.2457980270997	
Min							150	0	9
Max							81000	41667	700
Range							80850	41667	691
Variance							37320390.167181246	8562929.518387228	
Std. Dev							6109.041673387181	2926.2483692241894	
false									
true									

Structure of train_data.csv

Index	Column Name	Column Type
0	Loan_ID	STRING
1	Gender	STRING
2	Married	STRING
3	Dependents	STRING
4	Education	STRING
5	Self_Employed	STRING
6	ApplicantIncome	INTEGER
7	CoapplicantIncome	DOUBLE
8	LoanAmount	INTEGER
9	Loan_Amount_Term	INTEGER



<terminated> DataAnalysis [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe (May 8, 2021, 4:20:13 PM – 4:20:17 PM)

```
----TEST DATA SET----
shape:367 rows X 12 cols

Summary of test set
```

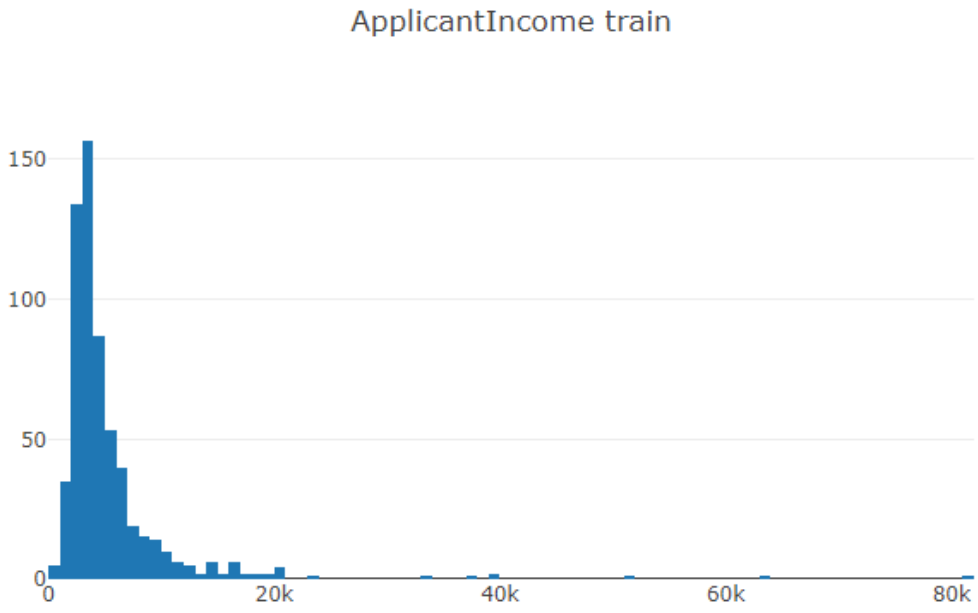
test_data.csv

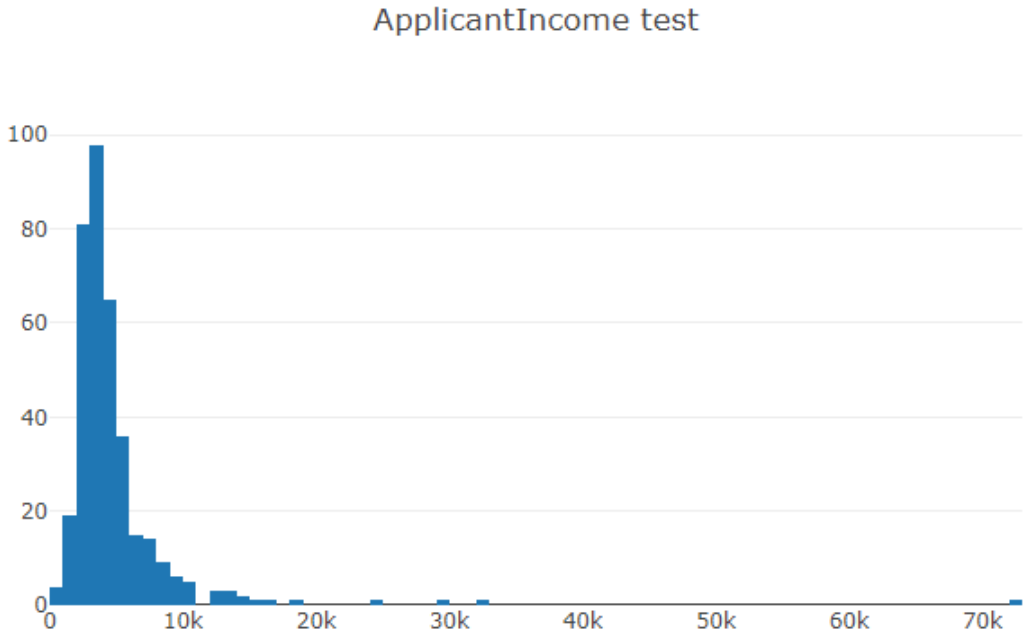
Summary	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount
Count	367	367	367	367	367	367	367	367	367
Unique	367	3	2	5	2	3			
Top	LP002376	Male	Yes	0	Graduate	No			
Top Freq.	1	286	233	200	283	307			
sum							1763655	576035	
Mean							4805.599455040872	1569.57765667575	
Min							0	0	28
Max							72529	24000	550
Range							72529	24000	522
Variance							24114831.087759264	5448639.49053766	
Std. Dev							4910.685398980398	2334.232098686345	

Structure of test_data.csv

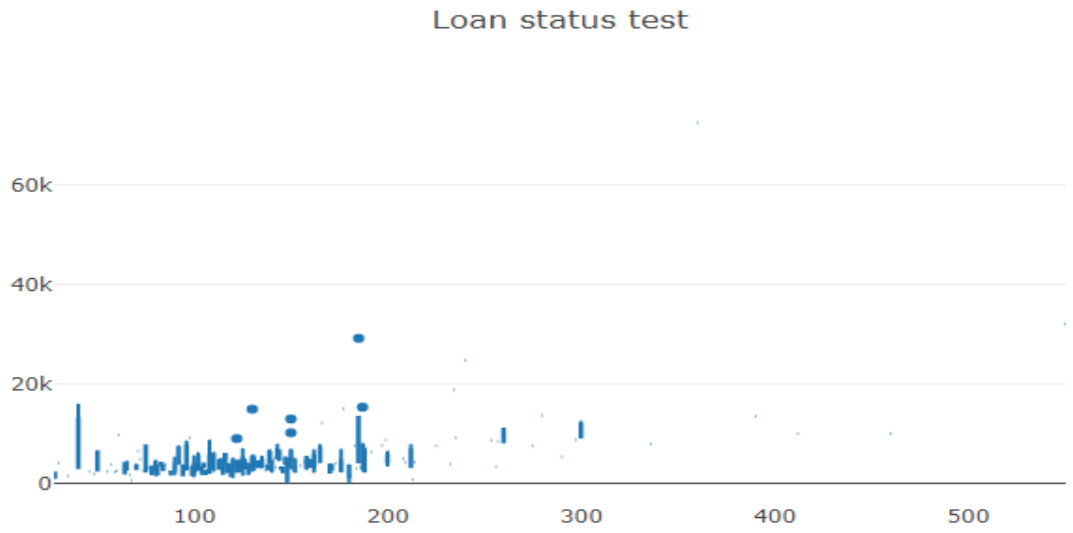
Index	Column Name	Column Type
0	Loan_ID	STRING
1	Gender	STRING
2	Married	STRING
3	Dependents	STRING
4	Education	STRING
5	Self_Employed	STRING
6	ApplicantIncome	INTEGER
7	CoapplicantIncome	INTEGER
8	LoanAmount	INTEGER
9	Loan_Amount_Term	INTEGER
10	Credit_History	INTEGER

Loan Eligibility Prediction





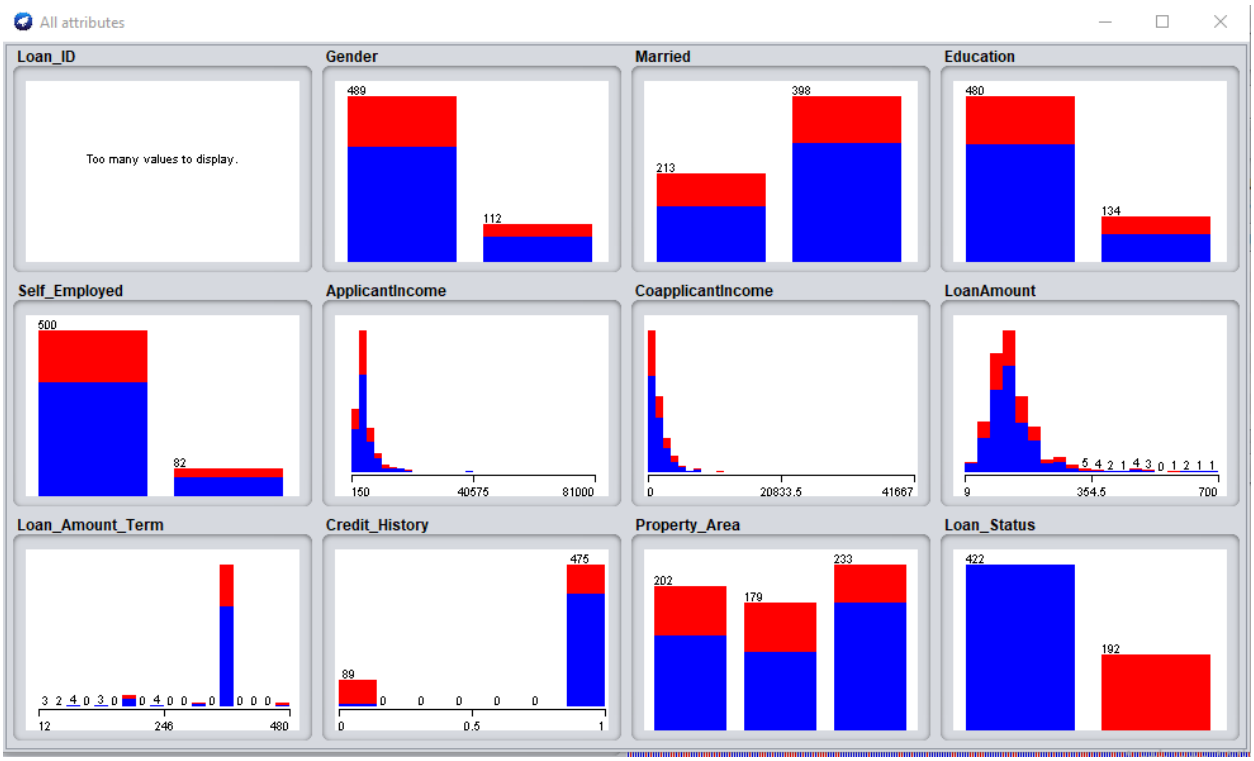
Loan Eligibility Prediction



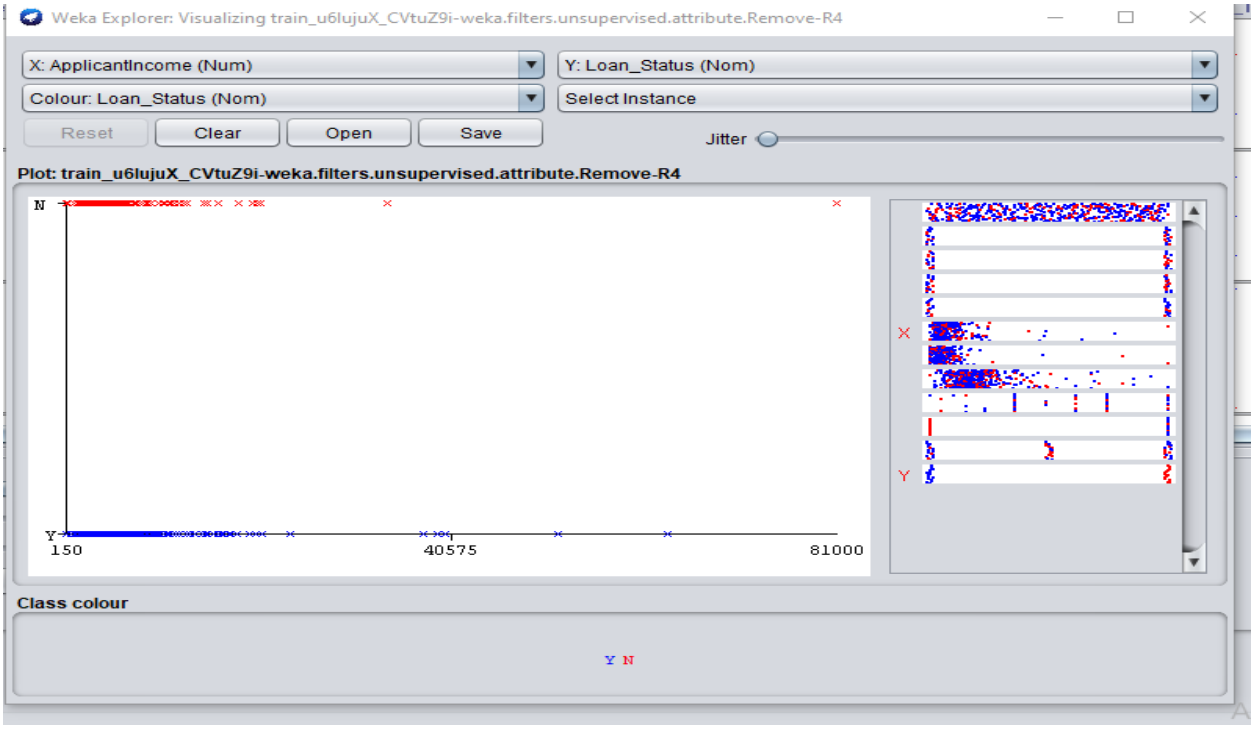
Loan Eligibility Prediction

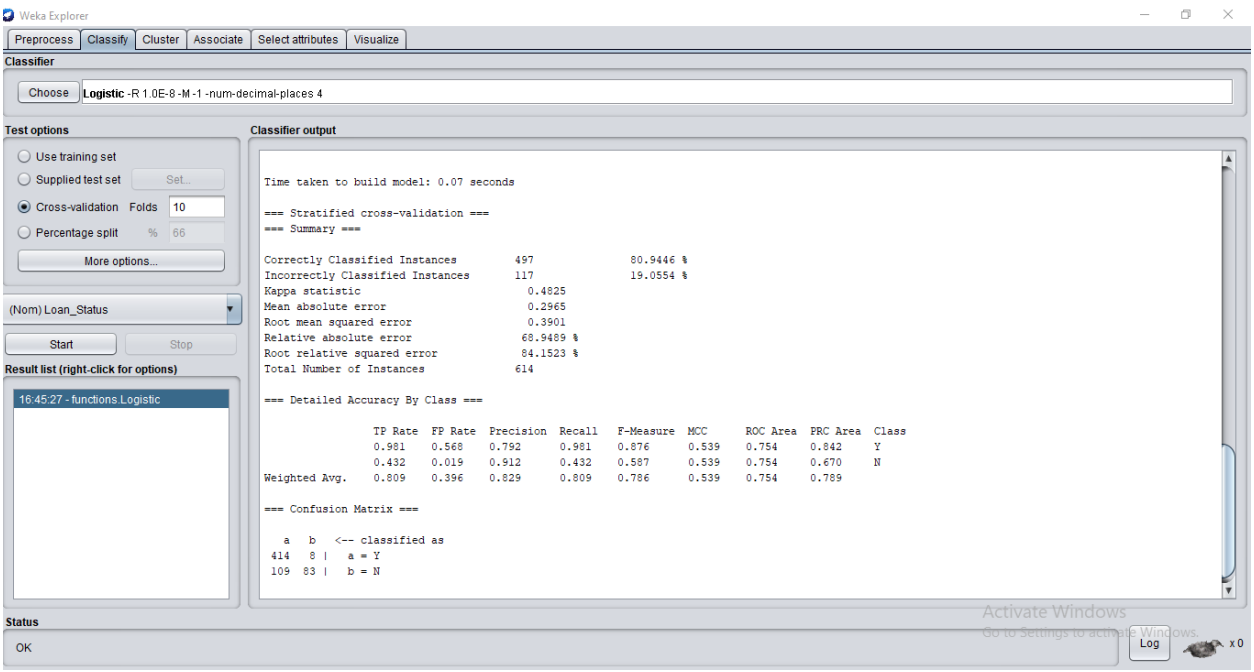
B)WEKA GUI Result:

TRAIN DATASET:

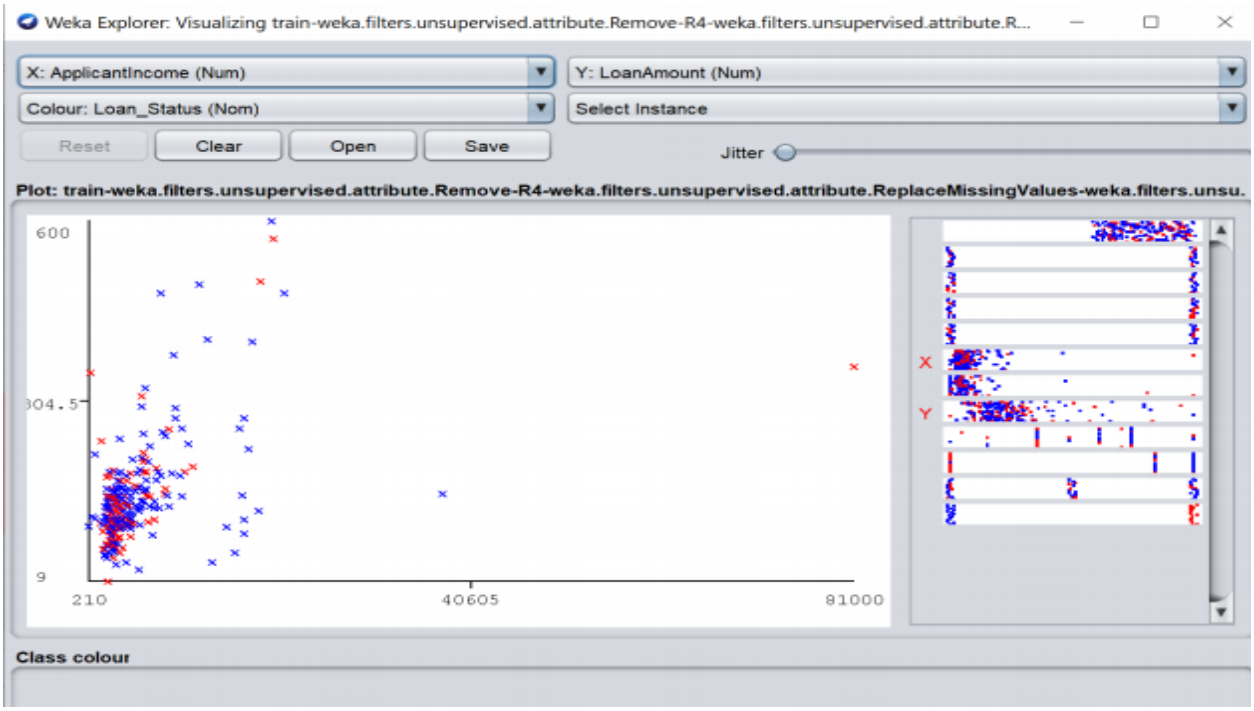


Loan Eligibility Prediction

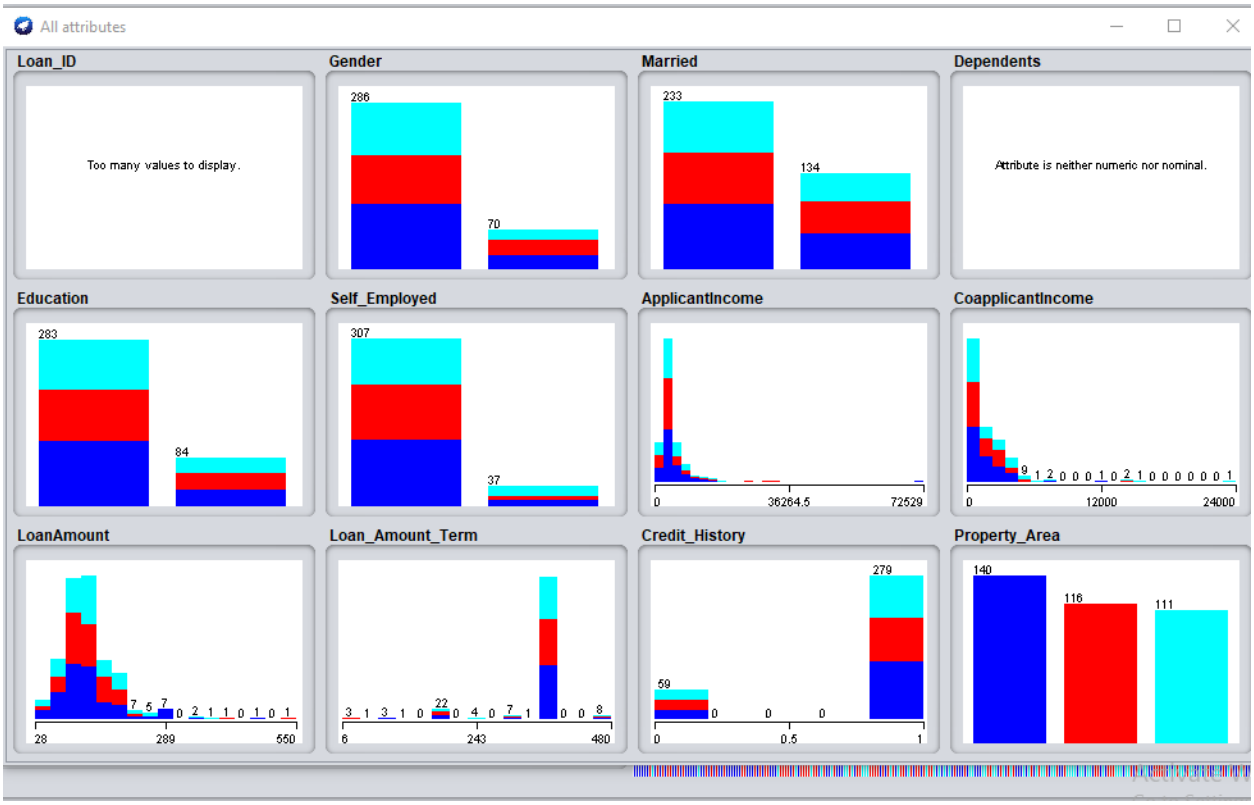




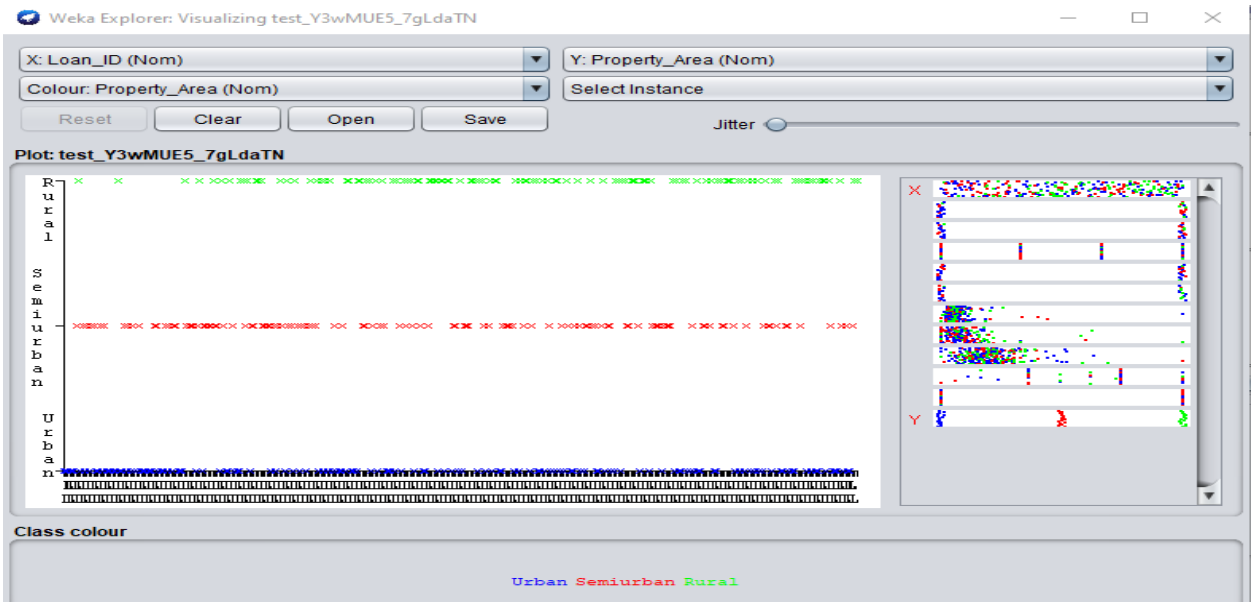
Loan Eligibility Prediction

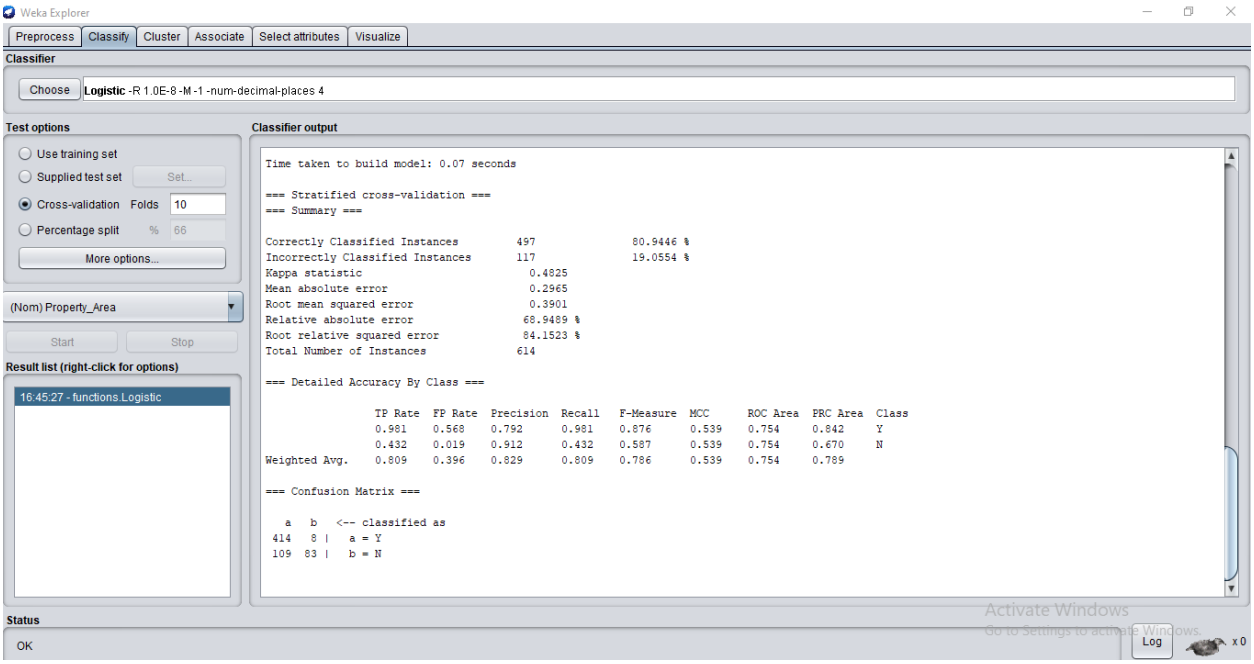


TEST DATASET:

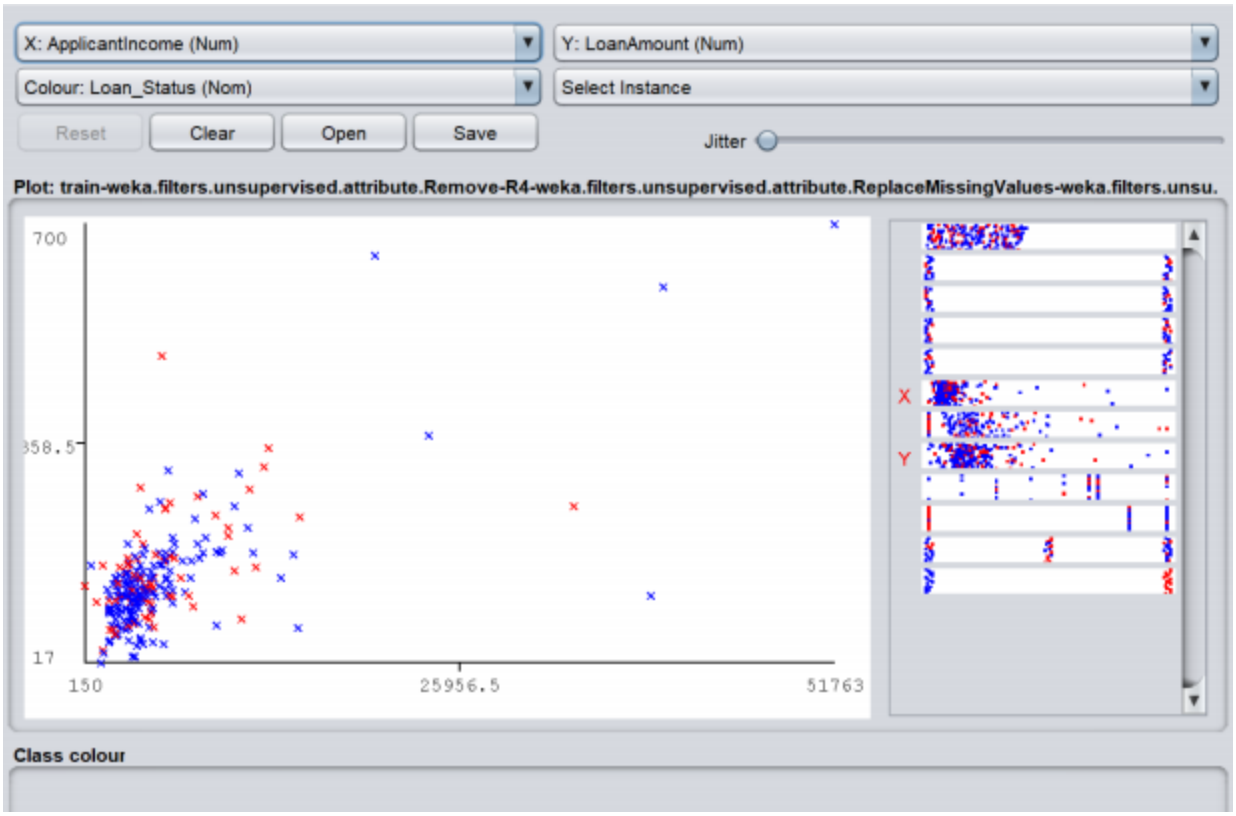


Loan Eligibility Prediction





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:

Loan Eligibility Prediction

➤ Data preparation can be tedious in Logistic Regression as both scaling andnormalization 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:

Loan Eligibility Prediction

Source Code:

A) Data Analysis Code:

```
package org.ml;
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("C:\\.eclipse\\org.ml\\src\\main\\java\\org\\ml\\train_data.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("C:\\.eclipse\\org.ml\\src\\main\\java\\org\\ml\\test_data.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());
        //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_dat
a.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("LoanAmount"),loantest_dat
a.nCol("ApplicantIncome")).build();
        Plot.show(new Figure(layout4, trace4));
        }
                catch (IOException e) {
      // TODO Auto-generated catch block
                        e.printStackTrace();
                        }
                }
        }
```

Loan Eligibility Prediction

B) Logistic Regression Model:

```
package org.ml;
import java.util.Arrays;
import weka.classifiers.Classifier;
import weka.classifiers.evaluation.Evaluation;
import weka.core.Instance;
import weka.core.Instances;
import weka.core.converters.ConverterUtils.DataSource;
public class LogRegression {
	public static Instances getInstances (String filename)
	{
		DataSource source;
		Instances dataset = null;
		try {
			source = new DataSource(filename);
			dataset = source.getDataSet();
			dataset.setClassIndex(dataset.numAttributes()-1);
		} catch (Exception e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
		return dataset;
	}
	public static void main(String[] args) throws Exception{
		Instances train_data
=getInstances("C:\\.eclipse\\org.ml\\src\\main\\java\\org\\ml\\train_data.arff");
		Instances test_data
=getInstances("C:\\.eclipse\\org.ml\\src\\main\\java\\org\\ml\\test_data.arff");
		System.out.print("The size of train data is:");
		System.out.println(train_data.size());
		System.out.print("The size of test data is:");
		System.out.println(test_data.size());
```

```
/** Classifier here is Linear Regression */
Classifier classifier = new weka.classifiers.functions.Logistic();
/** */
classifier.buildClassifier(train_data);
/**
* train the alogorithm with the training data and evaluate the
* algorithm with testing data
*/
Evaluation eval = new Evaluation(train_data);
eval.evaluateModel(classifier, test_data);
/** 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("------------------");
System.out.println(eval.getAllEvaluationMetricNames());
System.out.print("Recall :");
System.out.println(Math.round(eval.recall(1)*100.0)/100.0);
System.out.print("Precision:");
System.out.println(Math.round(eval.precision(1)*100.0)/100.0);
System.out.print("F1 score:");
System.out.println(Math.round(eval.fMeasure(1)*100.0)/100.0);
System.out.print("Accuracy:");
double acc = eval.correct()/(eval.correct()+ eval.incorrect());
System.out.println(Math.round(acc*100.0)/100.0);
System.out.println("------------------");
Instance predicationDataSet = test_data.get(2);
double value = classifier.classifyInstance(predicationDataSet);
```

```
            /** Prediction Output */
            System.out.println("Predicted label:");
            System.out.print(value);
        }

}
```