PROJET TITLE - LOAN ELIGIBILITY PREDECTION

NAME - CHAKKA JAHNAVI

SBID - SB20210110764

1- INTRODUCTION

1.1- Overview

I have considered two files which are Train data and Test data next reading data from Train and Test files using JAVA is done, then later on the subcategories of files are understood and the type of data is displayed in the console. Then all the data is represented in the form of a graph which is an overview/analysis of all the data present in the Train and Test data files.

After the visualization and cross verifying the data using WEKA software all the inputs, outputs and necessary data would be present for the cross-validation. Therefore, for any program that needs to be processed using ML(Machine Learning), there would be 80% of train data and 20% of test data after the cross-validation the result would be displayed in the console, and loan eligibility would be predicted.

1.2 - Purpose

Home loan eligibility is defined as a set of criteria basis which a financial institution assesses the creditworthiness of a customer to avail and repay a particular loan amount. Home loan eligibility depends on criteria such as age, financial position, credit history, credit score, other financial obligations etc.

Housing loan eligibility is primarily dependent on the income and repayment capacity of the individual(s).There are other factors that determine the eligibility of home loans such as age, financial position, credit history, credit score, other financial obligations etc.

In banks, people are allowed to calculate the loan eligibility to minimize this I have come up with a loan eligibility prediction to decrease the workload of the bank employee.

2 - LITERATURE SURVEY

2.1- Existing problem

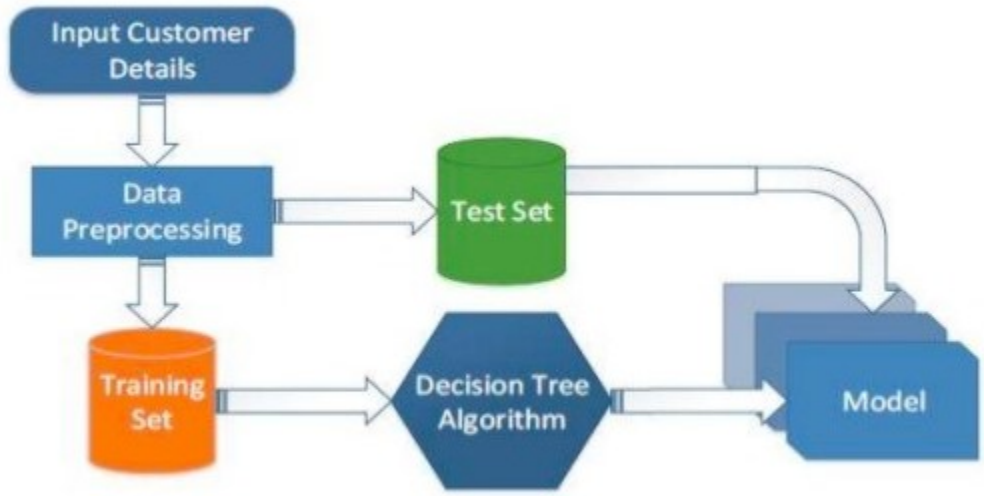
Loan Eligibility is defined as a set of criteria basis which a financial institution assesses the creditworthiness of a customer to avail and repay a particular loan amount. There are various types of loans like home loan,car loan etc. Generally, calculating the eligibility for loan is quite difficult.

2.2 - Proposed solution

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed.The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide.

3 - THEORITICAL ANALYSIS

3.1 - Block diagram



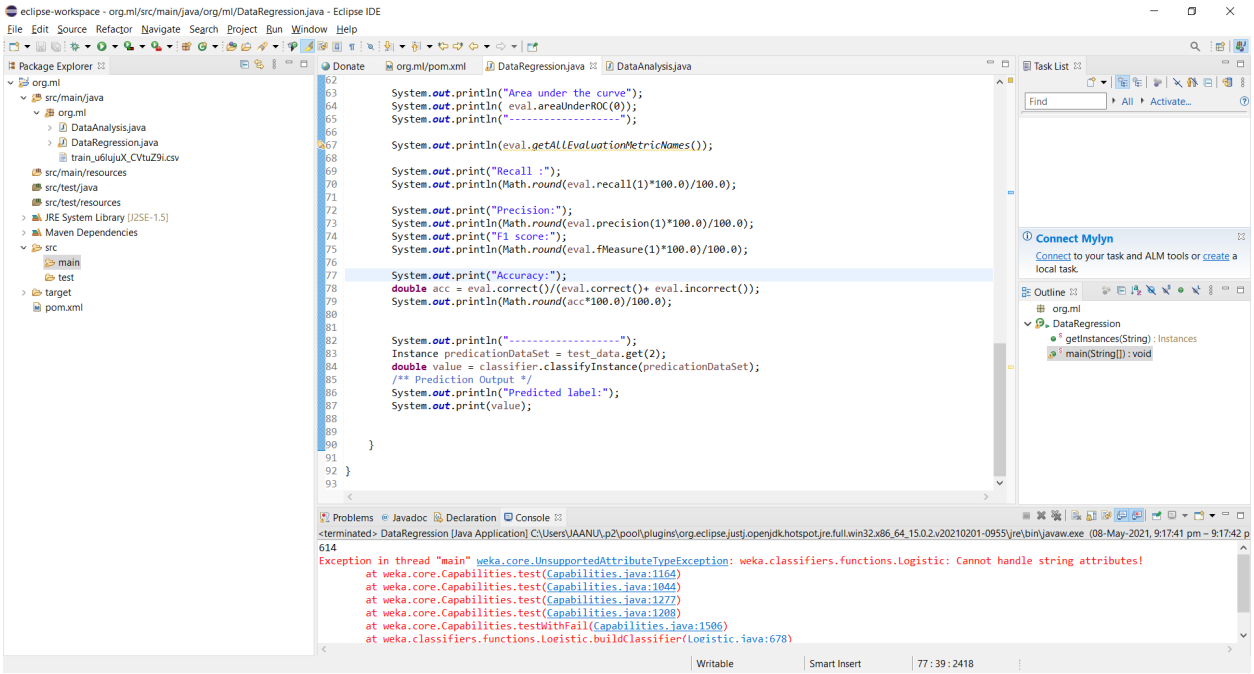
Architecture of Proposed Model

3.2 - Hardware / Software designing

A data set/ data collection is a collection of data. In the case of tabular data, a data set corresponds to one or more database tables, where every column of a table represents a particular variable, and each row corresponds to a given record of the data set in question. The data set lists values for each of the variables, such as height and weight of an object, for each member of the data set. Each value is known as a datum. Data sets can also consist of a collection of documents or files.

4 - FLOWCHART

5 - RESULT



6 - ADVANTAGES & DISADVANTAGES

ADVANTAGES:-

- The selection process for giving a Loan will be easy.
- Man work will be decreased.
- Mistakes will become zero.
- People will get fair loans according to the income, age, etc.
- Low cost maintanence.

DISADVANTAGES:-

- Prof of the given information will not be verified.
- A small mistake will change the whole analysis of the information.
- If we did'nt update then we will get old analysis.

7 - APPLICATIONS

The areas where the Loan prediction application is used are

- Commercial Banks
- Industrial Banks
- Co-operative Banks
- Savings Banks
- Central Banks

8 - CONCLUSION

Therefore, this application will reduce the man work by decreasing of selecting the eligible people for loan and decrease the man-made mistakes.

9 - FUTURE SCOPE

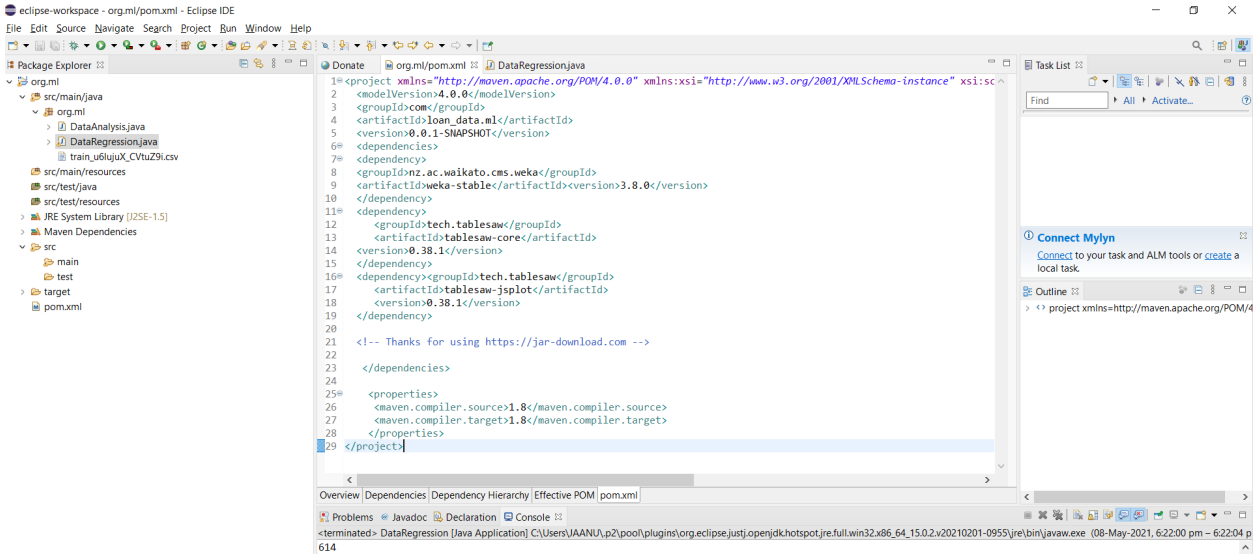
The further things which I can add to this application are to be able to verify the proof of the person and to be able to distinguish the difference between original and duplicate copy.

10 - BIBILOGRAPHY

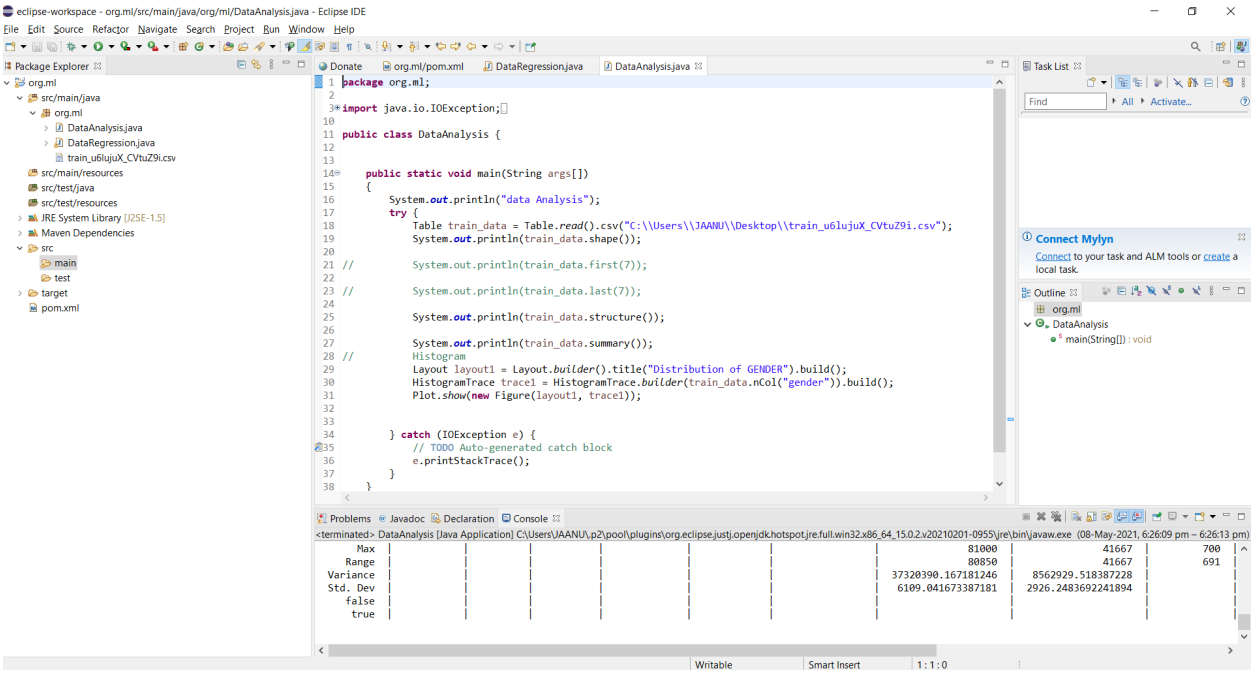
- https://www.bankbazaar.com/personal-loan/loans-for-construction.html
- https://www.ijrte.org/wp-content/uploads/papers/v7i4s/E2026017519.pdf

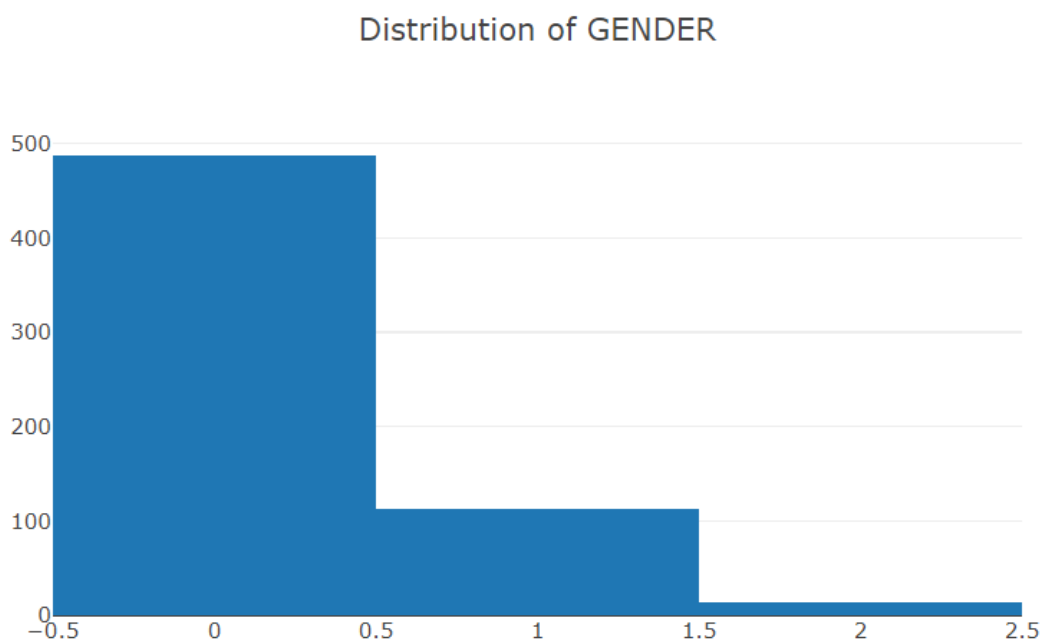
APPENDIX

pom.xml:-



DataAnalysis:-





DataRegression:-

eclipse-workspace - org.ml/src/main/java/org/ml/DataRegression.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

- org.ml
  - src/main/java
    - org.ml
      - DataAnalysis.java
      - DataRegression.java
      - train_u6lujuX_CVtuZ9i.csv
  - src/main/resources
  - src/test/java
  - src/test/resources
  - JRE System Library [J2SE-1.5]
  - Maven Dependencies
  - src
    - main
    - test
  - target
  - pom.xml

Donate | org.ml/pom.xml | DataRegression.java | DataAnalysis.java

```java
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 DataRegression {

    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;
    }

```

eclipse-workspace - org.ml/src/main/java/org/ml/DataRegression.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

- org.ml
  - src/main/java
    - org.ml
      - DataAnalysis.java
      - DataRegression.java
      - train_u6lujuX_CVtuZ9i.csv
  - src/main/resources
  - src/test/java
  - src/test/resources
  - JRE System Library [J2SE-1.5]
  - Maven Dependencies
  - src
    - main
    - test
  - target
  - pom.xml

Donate | org.ml/pom.xml | DataRegression.java | DataAnalysis.java

```java
    }

    public static void main(String[] args) throws Exception{

        Instances train_data = getInstances("C:\\Users\\JAANU\\Desktop\\train_u6lujuX_CVtuZ9i.csv.arff");
        Instances test_data = getInstances("C:\\Users\\JAANU\\Desktop\\test_Y3wMUE5_7gLdaTN.csv.arff");
        System.out.println(train_data.size());

        /** Classifier here is Linear Regression */
        Classifier classifier = new weka.classifiers.functions.Logistic();
        /** */
        classifier.buildClassifier(train_data);


        /**
         * train the algorithm 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 algorithm 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("-------------------");

```

eclipse-workspace - org.ml/src/main/java/org/ml/DataRegression.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

- org.ml
  - src/main/java
    - org.ml
      - DataAnalysis.java
      - DataRegression.java
      - train_u6lujuX_CVtuZ9i.csv
  - src/main/resources
  - src/test/java
  - src/test/resources
  - JRE System Library [J2SE-1.5]
  - Maven Dependencies
  - src
    - main
    - test
  - target
  - pom.xml

Donate | org.ml/pom.xml | DataRegression.java | DataAnalysis.java

```java

        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);


    }

}
```