PROJECT TITLE - LOAN ELIGIBILITY PREDICTION

NAME - J. Sri Varsha Meghana

SBID - SB20210085417

1.INTRODUCTION

1.1 Overview of the project

The term loan refers to a type of credit vehicle in which a sum of money is lent to another party in exchange for future repayment of the principle value or amount.

For this problem, we have two CSV files: train file and test files.

- The Train filewill be used for training the model, i.e. our model will learn from this file. It contains all the independent variables and the target variable.
- Test file contains all the independent variables, but not the target variable. We will apply the model to predict the target variable for the test data.

READING THE DATA

A copy of the train and test data will be done so that even if we have to make any changes in these datasets we would not lose the original datasets.

UNDERSTANDING DATA

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.

MISSING VALUE IMPUTATION

We will treat the missing values in all the features one by one.We can consider these methods to fill the missing values:

- For numerical variables: imputation using mean or median
- For categorical variables: imputation using mode

CROSS VALIDATION

Then the cross validation between the train and test data will be done and the output is therefore displayed on the console.

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 Problems

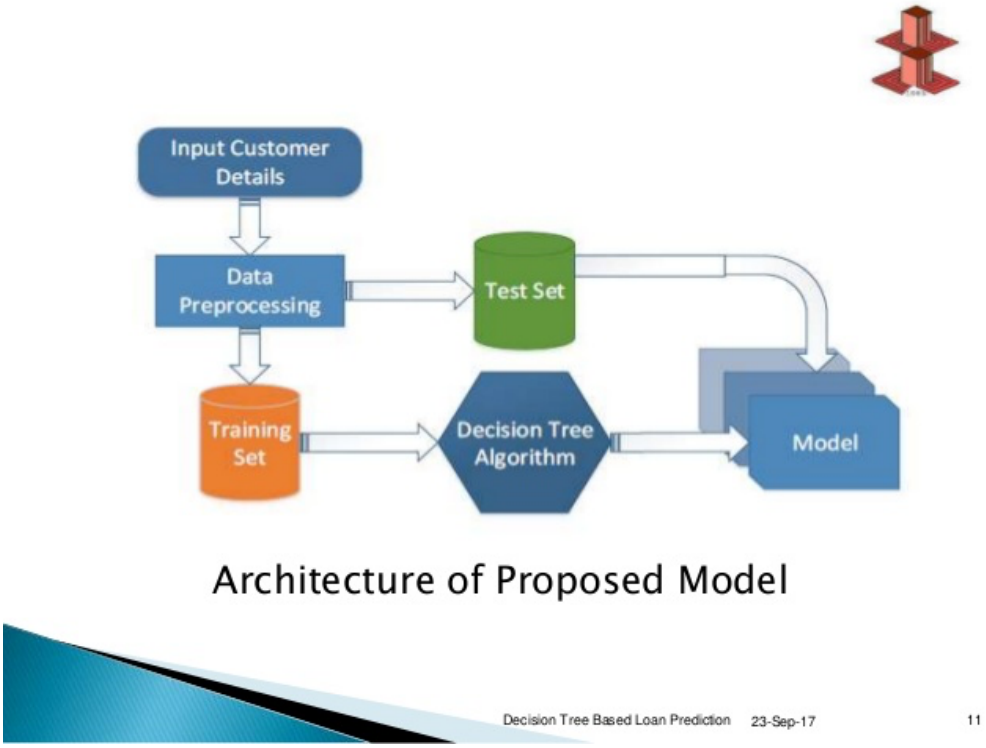
Loan Eligibility is defined as a set of criteria basis which a financial institution assesses the creditworthiness of a customer to avail and repay a particular loan amount. There are various types of loans like home loan,car loan etc. Generally, calculating the eligibility for loan is quite difficult.

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

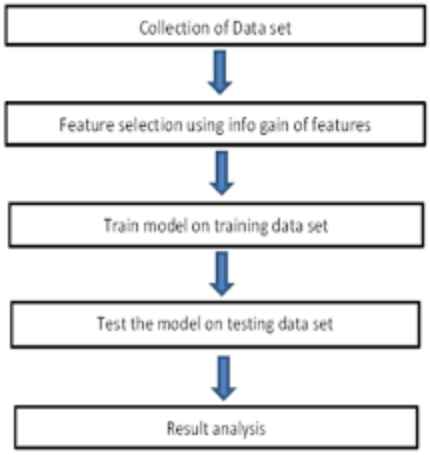


3.2 Hardware/Software Designing

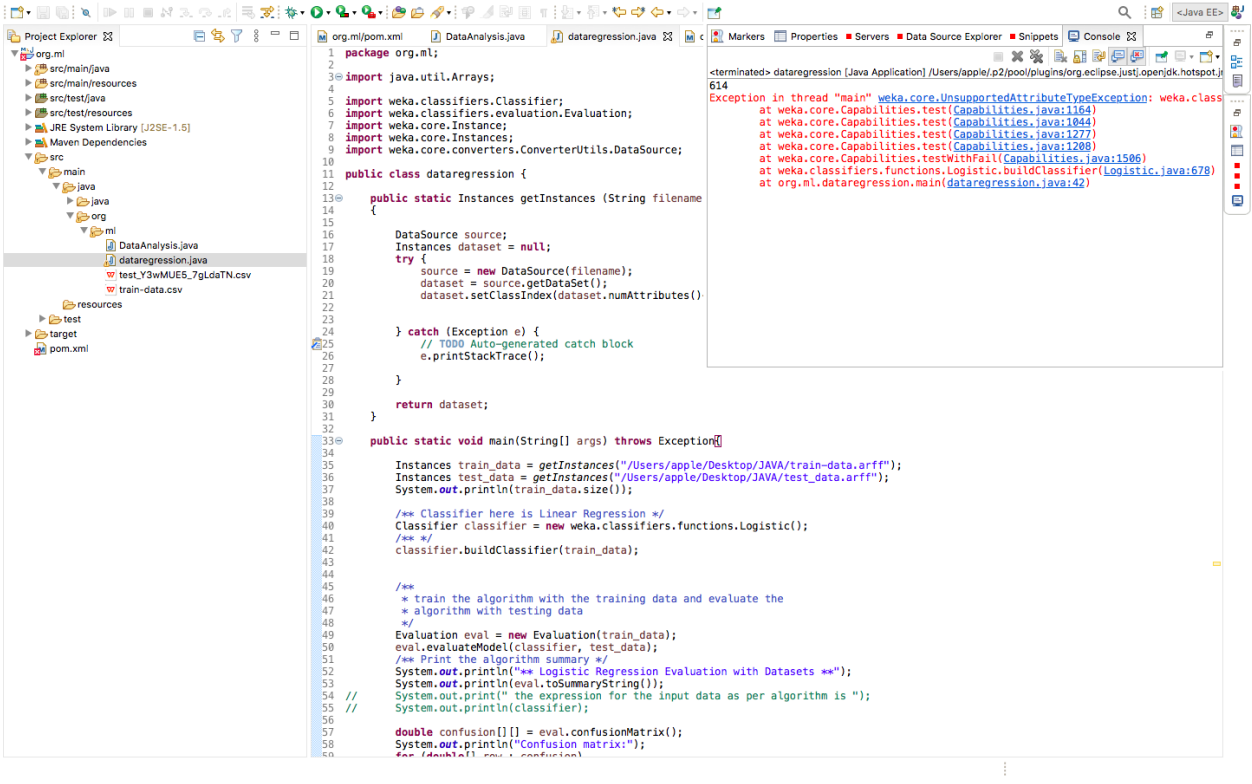
A data set/ data collection is a collection of data. In the case of tabular data, a data set corresponds to one or more database tables, where every column of a table represents a particular variable, and each row corresponds to a given record of the data set in question. The data set lists values for 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.

Cross-validation is a technique for evaluating ML models by training several ML models on subsets of the available input data and evaluating them on the complementary subset of the data. Use cross-validation to detect overfitting, ie, failing to generalize a pattern.

4 .FLOWCHART



5 .RESULT



6 .ADVANTAGES AND 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 the selection of eligible people for loan and decrease man-made mistakes.

9.FUTURE SCOPE

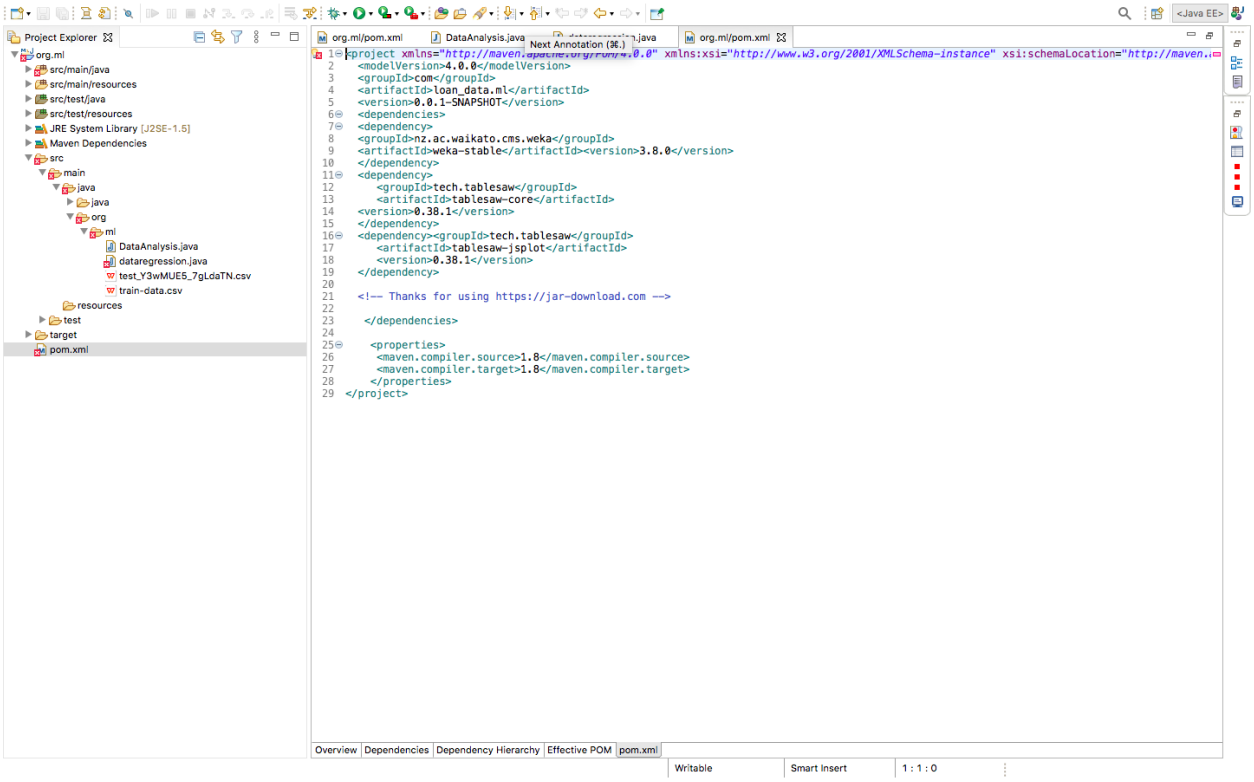
The further things which I can add to this application are to be able to verify the proof of the person and to be able to distinguish the difference between original and duplicate copy.

10. BIBILOGRAPHY

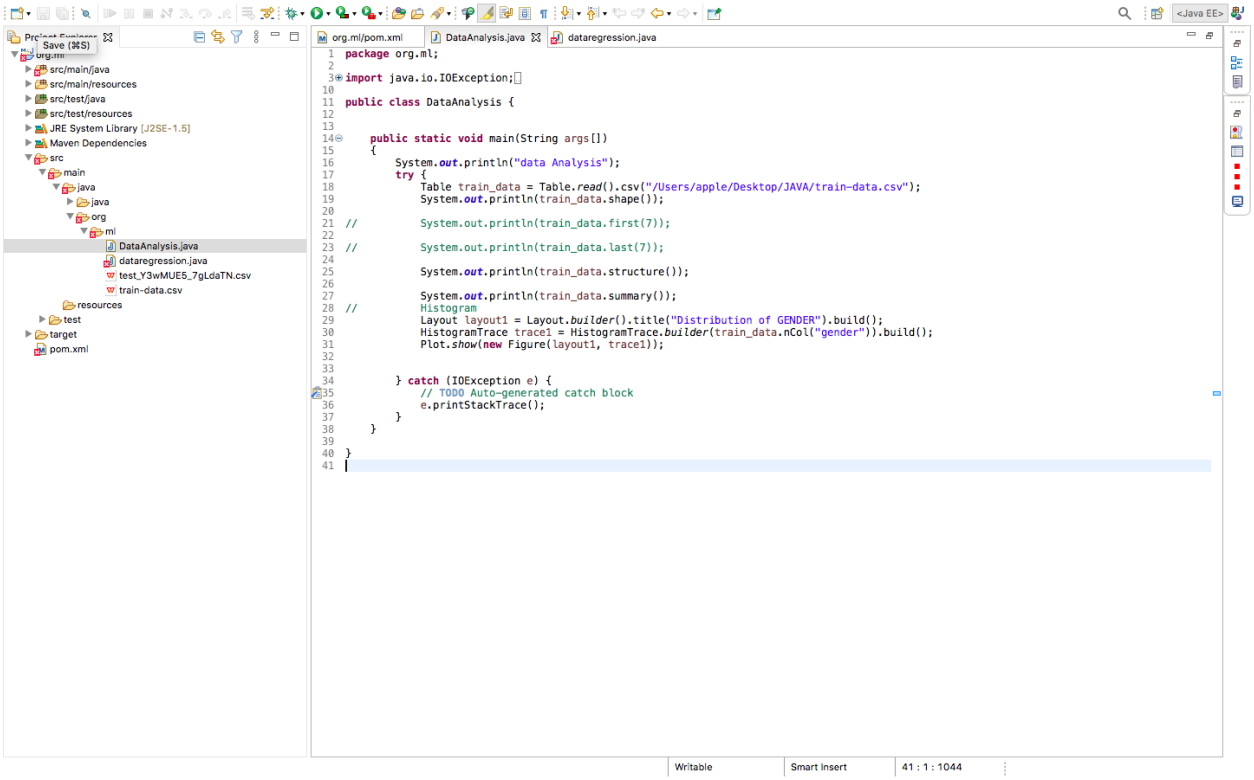
https://www.bankbazaar.com/personal-loan/loans-for-construction.html
https://www.ijrte.org/wp-content/uploads/papers/v7i4s/E2026017519.pdf

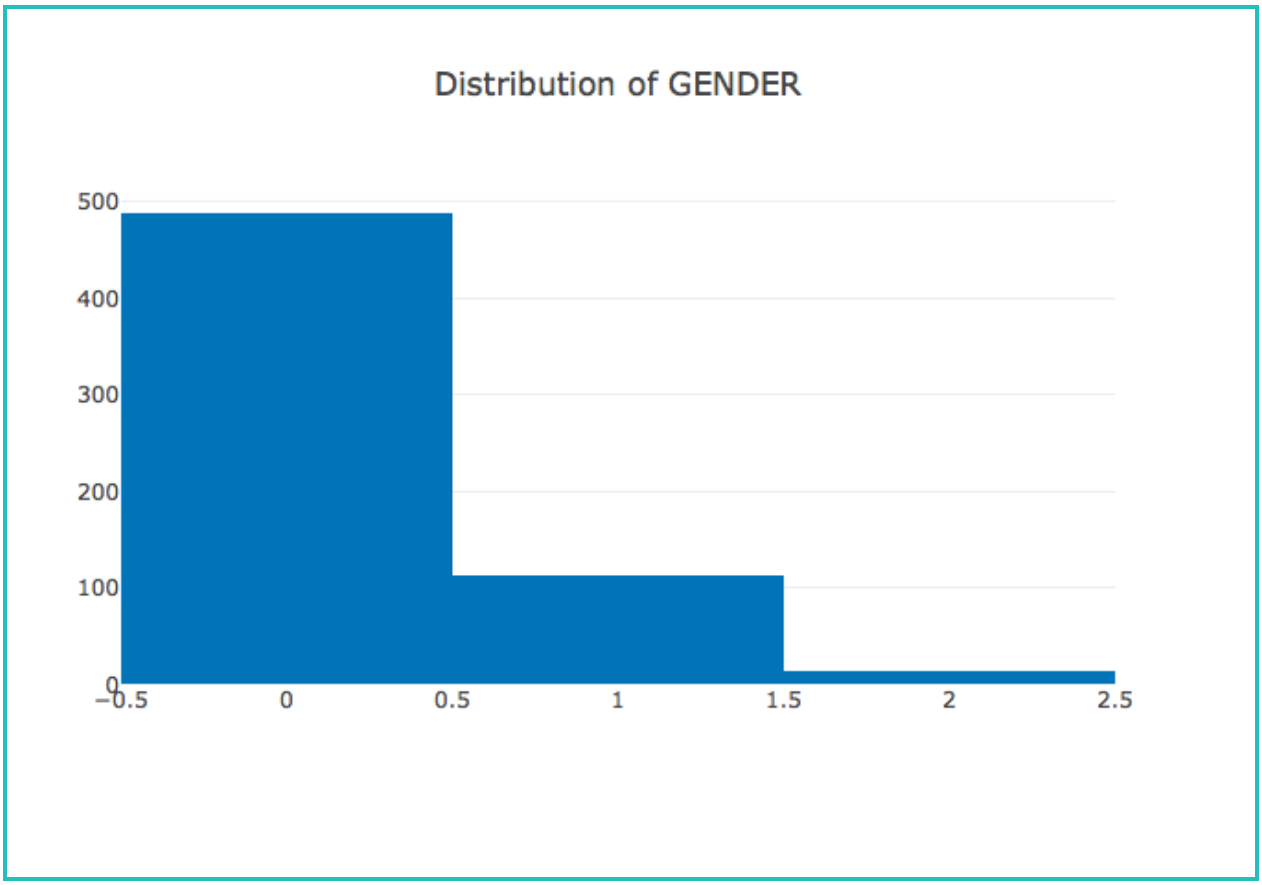
11.APPENDIX

>POM.XML

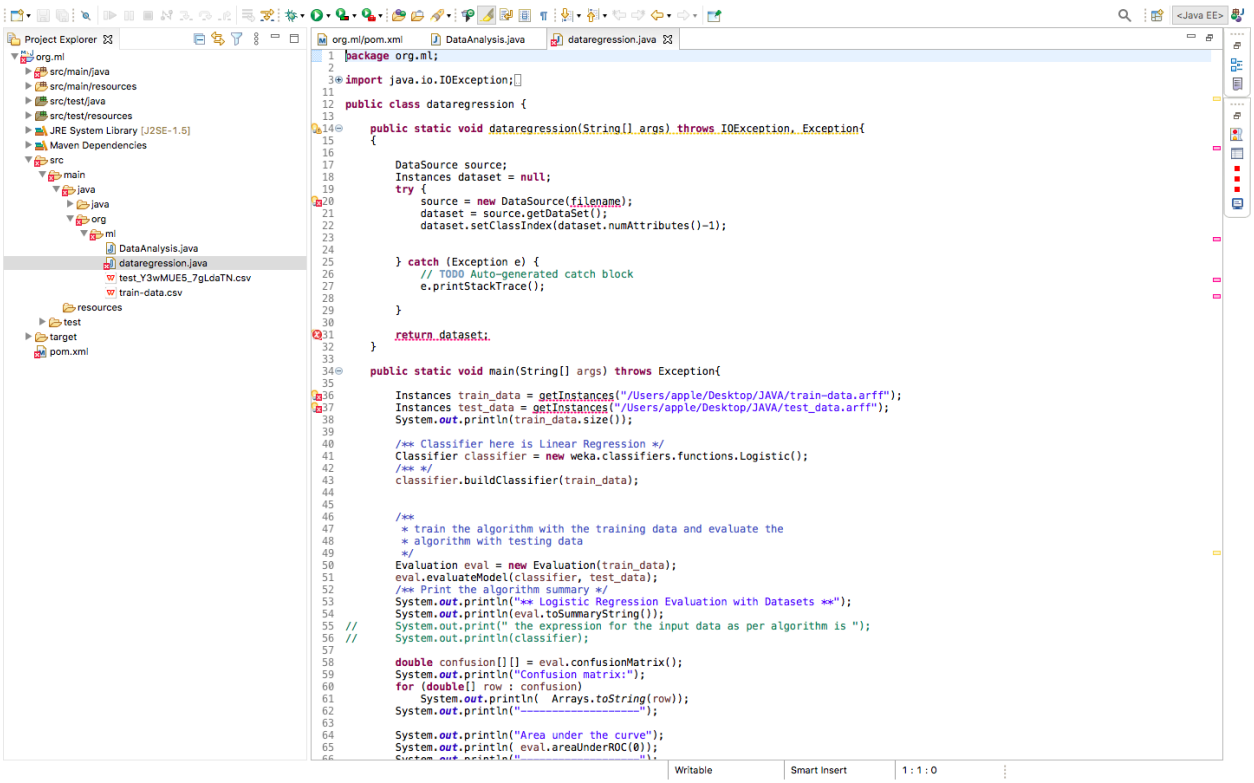


>DATA SET TESTING

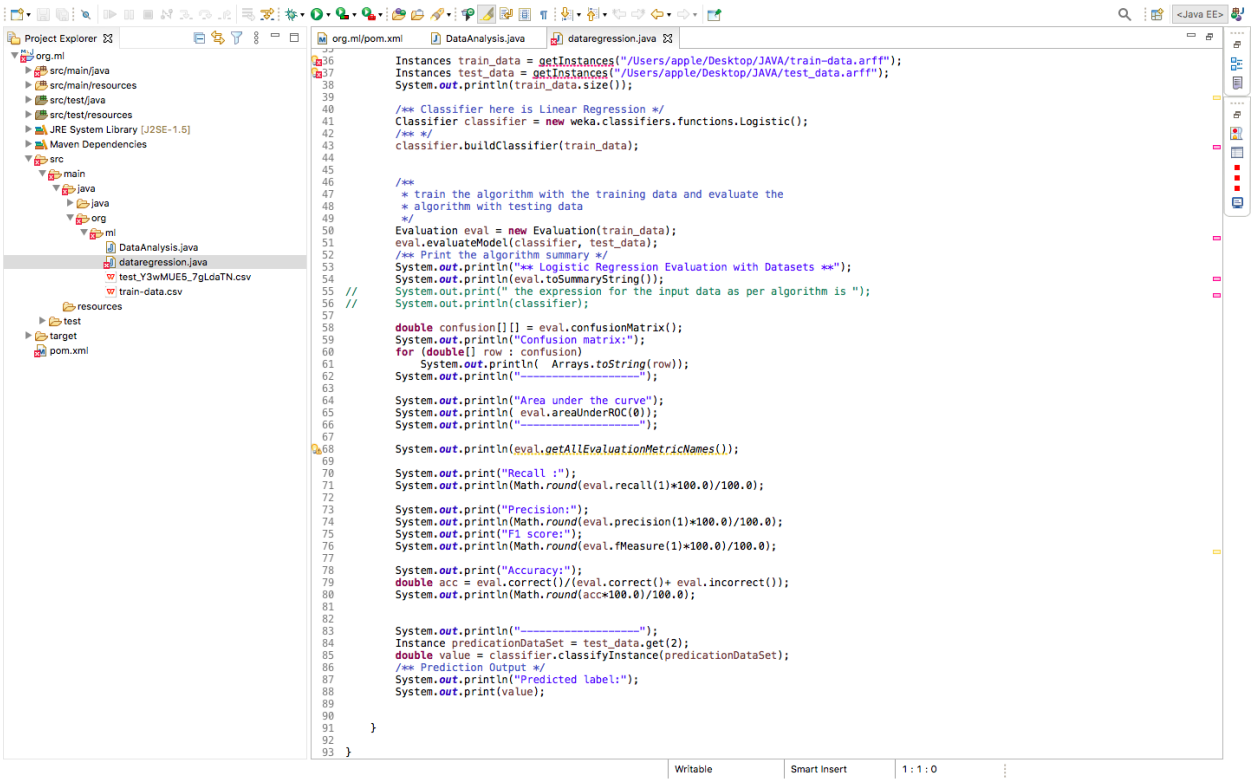




>CROSS VALIDATION




```
package org.ml;

import java.io.IOException;

public class dataregression {

    public static void dataregression(String[] args) throws IOException, Exception{
    {

        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("/Users/apple/Desktop/JAVA/train-data.arff");
        Instances test_data = getInstances("/Users/apple/Desktop/JAVA/test_data.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("----------------------");

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




```
        Instances train_data = getInstances("/Users/apple/Desktop/JAVA/train-data.arff");
        Instances test_data = getInstances("/Users/apple/Desktop/JAVA/test_data.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("--------------------");

        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);


    }
}
```