INTRODUCTION

Overview :

Detecting fraud transactions is of great importance for any credit card company. this is the most common problem faced by all the companies and banks now a days so this will be helpful to those companies

Purpose :

The project will be help full to many consonances which will use transaction from the clients if we able to use this project in this corporate world we can easily identify the false transactions and we can be able to stop the wrong person to use our product and get benefited and also to know who is fraud and who is the correct person in the society

LITERATURE SURVEY

Existing problem :

There are very less products to solve this kind of problems and they are very expansive and it take lot of money to effort such solutions

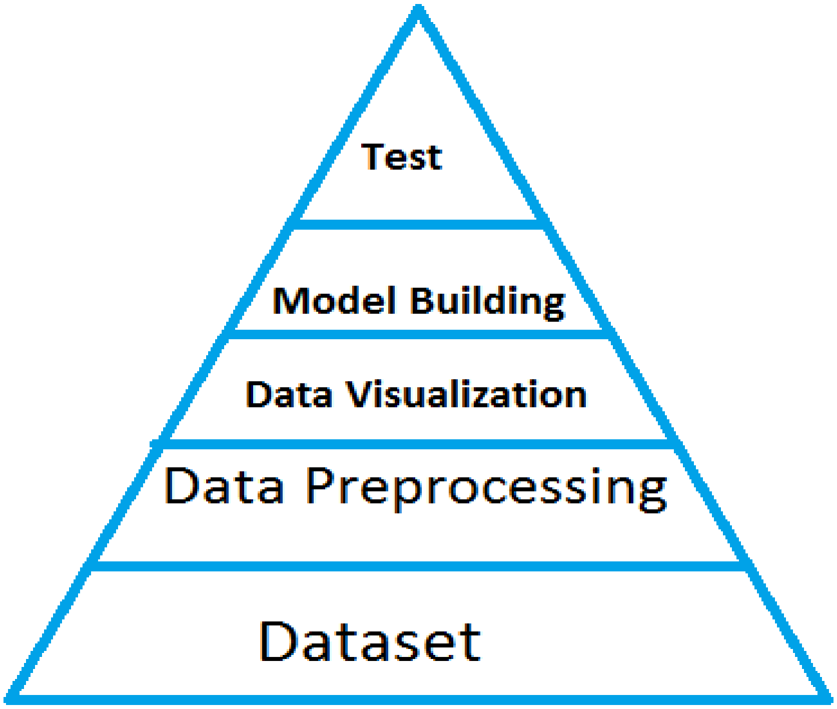
hear are some software to solve the problem of fraud diction in banking sector or in some software companies the software like get fraud is most le used in the companies and it is most expansive to effort this days

Proposed solution :

I are with a better solution this is very easy to effort and it work on the basics principles we first analysis the data and train the machine and after that we use the same data to test the machine and we will provide it the machine so now it will be identify the fraud account

THEORITICAL ANALYSIS

Block diagram :



Hardware / Software designing ing

I use some special software to do some special works in our project mainly we use three kind of software in our project they are

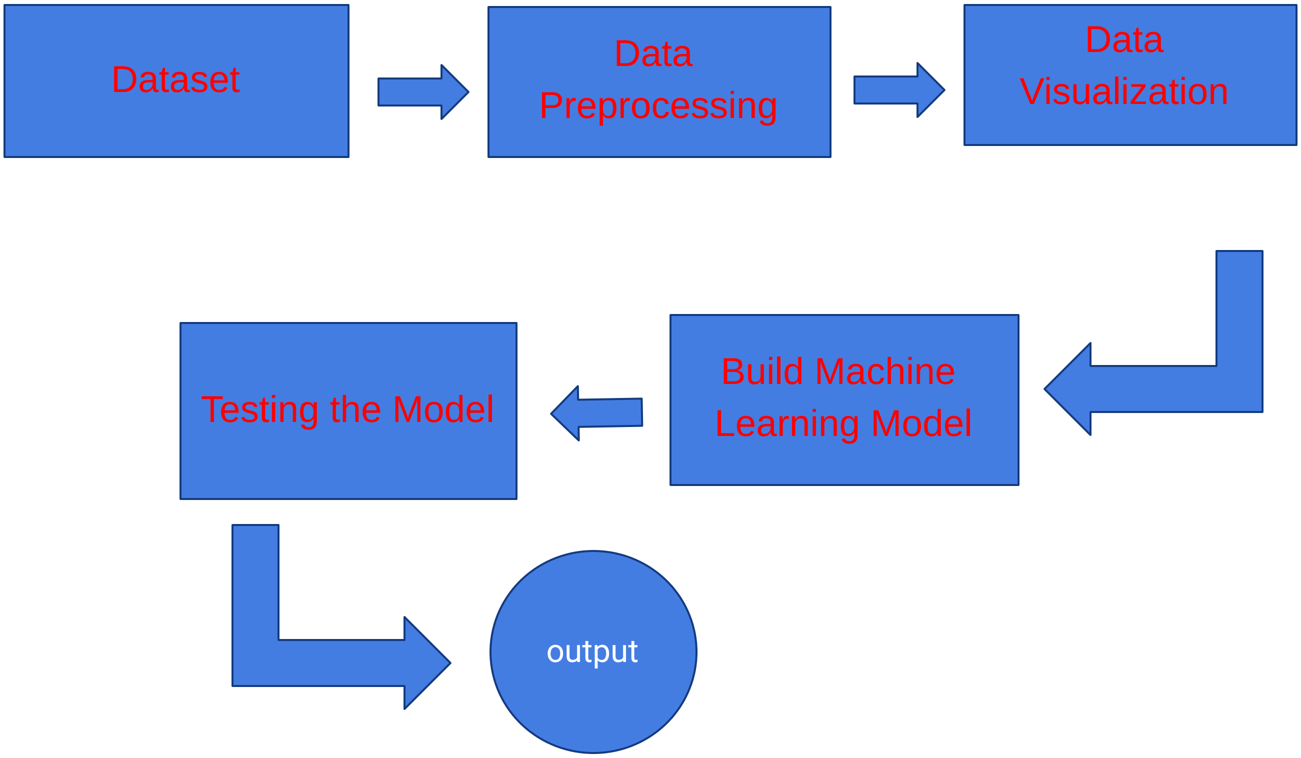
1. weak - it is used for analysis of data and to find mean , median & mode for the data
2. eclipse - it is used for analysis of data and to find mean , median & mode for the data but hear we use some codes to that work
3. java virtual machine - it is used to train and test the machine to check it is working properly or not

EXPERIMENTAL INVESTIGATIONS

I started working on the collection of data from different sources and from various wed sites it took a long time to do reaches on the data of fraud decathletes after continuous working on data I finally with data set then I started to clear the data which I done want to be in data set after that I started processing the data to clear unwanted data and then started the data in to a clear and correct set after multiple hours of working on the data set I have a clear and correct data set to visual the data now we have a correct data set I started to visual it to find mean , median and mode to make shore what data should be faded to the system after that I started to Build Machine Learning Model and I dived my data set in to two parts one is train and other is to test once my model is ready I started to train it after training is completed I started to test it to make sure the prototype is working properly or not once I furnished training and testing I am ready with the modal.

after working on the model for along time and after lot of trials and errors I finished the prototype and it is in working condition

FLOWCHART



RESULT

The out put of this project is obvious is very preface and accurate the model which e designed is very perfect and ti is correctly assuming the out put of the it will be very help full to companies and even in banking sector

ADVANTAGES & DISADVANTAGES

ADVANTAGES

This project is used in many companies and even in banking sector it is very help full and it will save lot of time efforts it will also help in to save men power wasting on it

DISADVANTAGES

The only disadvantage we have in the project is if we manipulate the code or if we give wrong data to the model it show lot of errors then the program will be companies in problem

APPLICATIONS

The application which i developed is very use full in many platforms such as banking , companies , finance and loan agency's.....ETC.

CONCLUSION

I want to conclude by saying that the project which I developed is very use and work properly in the fraud detection method this is most advanced model and easy to understand and work with it it is user friendly to.

APPENDIX

Source Code

code used in eclipses

To import required files:-

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com</groupId>
  <artifactId>org.ml</artifactId>
  <version>0.0.1-SNAPSHOT</version>
<dependencies>
  <dependency>
    <groupId>nz.ac.waikato.cms.weka</groupId>
    <artifactId>weka-stable</artifactId>
    <version>3.8.0</version>
  </dependency>
  <dependency>
    <groupId>tech.tablesaw</groupId>
    <artifactId>tablesaw-core</artifactId>
    <version>0.38.1</version>
   </dependency>
   <dependency>

<groupId>tech.tablesaw</groupId>

<artifactId>tablesaw-jsplot</artifactId>

<version>0.38.1</version>
</dependency>
<!– Thanks for using https://jar-download.com –>
  </dependencies>
   <properties>
        <maven.compiler.source>1.8</maven.compiler.source>
        <maven.compiler.target>1.8</maven.compiler.target>
    </properties>
</project>
```

to Analysis the data

```
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[])
{
System.out.println("data Analysis");
try {
Table fraud_data
Table.read().csv("F:\\TASK\\eclipse\\org.ml\\src\\main\\java\\org\\ml\\fraud-dataset.csv");
//
System.out.println(bank_data.shape());
                                                                                    //
                                                                                    //
System.out.println(bank_data.first(7));
                                                                                    //
System.out.println(bank_data.last(7));



System.out.println(fraud_data.structure());



System.out.println(fraud_data.summary());

////   His Layout layout1=Layout.builder().title("Distribution of Fraud_Risk").bui
HistogramTrace trace1= HistogramTrace.builder(fraud_data.nCol("agefrod
de")).build();Plot.show(new Figure(layout1, trace1)Layout layout3 =
Layout.builder().title("ApplicantIncome").build(); BoxTrace trace3
=BoxTrace.builder(fraud_data.categoricalColumn("Gender"),
fraud_data.nCol("LoanAmount")).build();
 Plot.show(new Figure(layout3, trace3));
```

```
} catch (IOExceptio{
                                                        // TODO
Auto-generated catch block

e.printStackTrace();
```

code used to find LogRegression

```
}package creditcard.fraud;

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:\\Users\\PERSONAL\\eclipse-workspace\\org1.ml\\src\\main\\java\\org1\\ml
\\diabetes_train.arff");
                                                                        Instances test_data =
getInstances("C:\\Users\\PERSONAL\\eclipse-workspace\\org1.ml\\src\\main\\java\\org1\\ml
\\diabetes_test.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 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);


                                                                 }

}
```

code used to find

```
import java.io.IOException;
import weka.classifiers.Evaluation;
import weka.classifiers.functions.LinearRegression;
import weka.core.Instances;
import weka.core.converters.ConverterUtils.DataSource;
public class LiR {
                                                                 public static void
main(String[] args) throws Exception {
                                                                        DataSource source
=new
DataSource("C:\\Users\\dhondik\\eclipse-workspace\\org.abc\\src\\main\\java\\org\\abc\\bo
dyfat.arff");
                                                                        Instances
dataset=source.getDataSet();

dataset.setClassIndex(dataset.numAttributes()-1);
                                                                        //linear Regression
                                                                        LinearRegression
lr=new LinearRegression();

lr.buildClassifier(dataset);


                                                                        Evaluation lreval
=new Evaluation(dataset);

lreval.evaluateModel(lr,dataset);

System.out.println(lreval.toSummaryString());


                                                                      }

}
```