CREDIT CARD FRAUD PREDICTION

1.INTRODUCTION

1.1Overview:

Credit Card Fraud Prediction is the process which is used to predict fraud.
The finance and banking is very important sectors in our present generation, where almost every human has to deal with bank either physically or online. Nowadays most of E-commerce application systems are done through credit card and online net banking. Credit card fraud can be defined as "Unauthorized account activity of a person for which the account was not inteded.
So, using Logistic Regression we predict whether the transactions are fraud or not in this project.

1.2Purpose

It is important for credit card companies to be able to recognize fraudulent credit card transactions so that customers are not charged for the items that they did not purchase. So, this project helps to predit whether the transactions are fraud or not.

2.LITERATURE SURVEY

2.1Existing problem

Methods to solve this problem are:

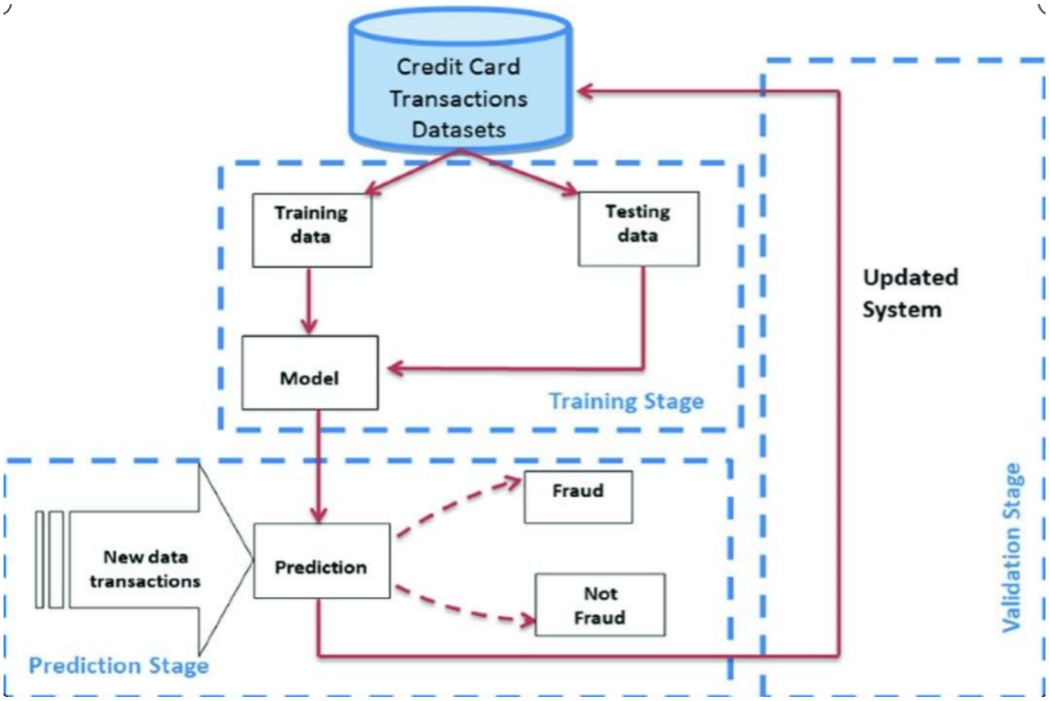
- Logistic Regression
- Decision Tree
- Random Forest
- Naive Bayes
- ANN Model

2.2Proposed solution

Solution that I suggest is Logistic Regression :
Logistic Regression works with sigmoid function because the sigmoid function can be used to classify the output that is dependent feature and it uses the probability for classification of the dependent feature.
This algorithm works well with less amount of data set because of the use of sigmoid function. If the value of the sigmoid function is greature than 0.5 the output will be 1.If the output of the sigmoid function is less than 0.5 then the output is considered as 0.

3.THEORITICAL ANALYSIS

3.1BlockDiagram



3.2Hardware /Software designing

i)Hardware Requirement:

- Windows 7 and above (64 bit)
- RAM : 4GB
- Processor: Minimum pentium 2 266 MHz processor
- Browsers: Chrome

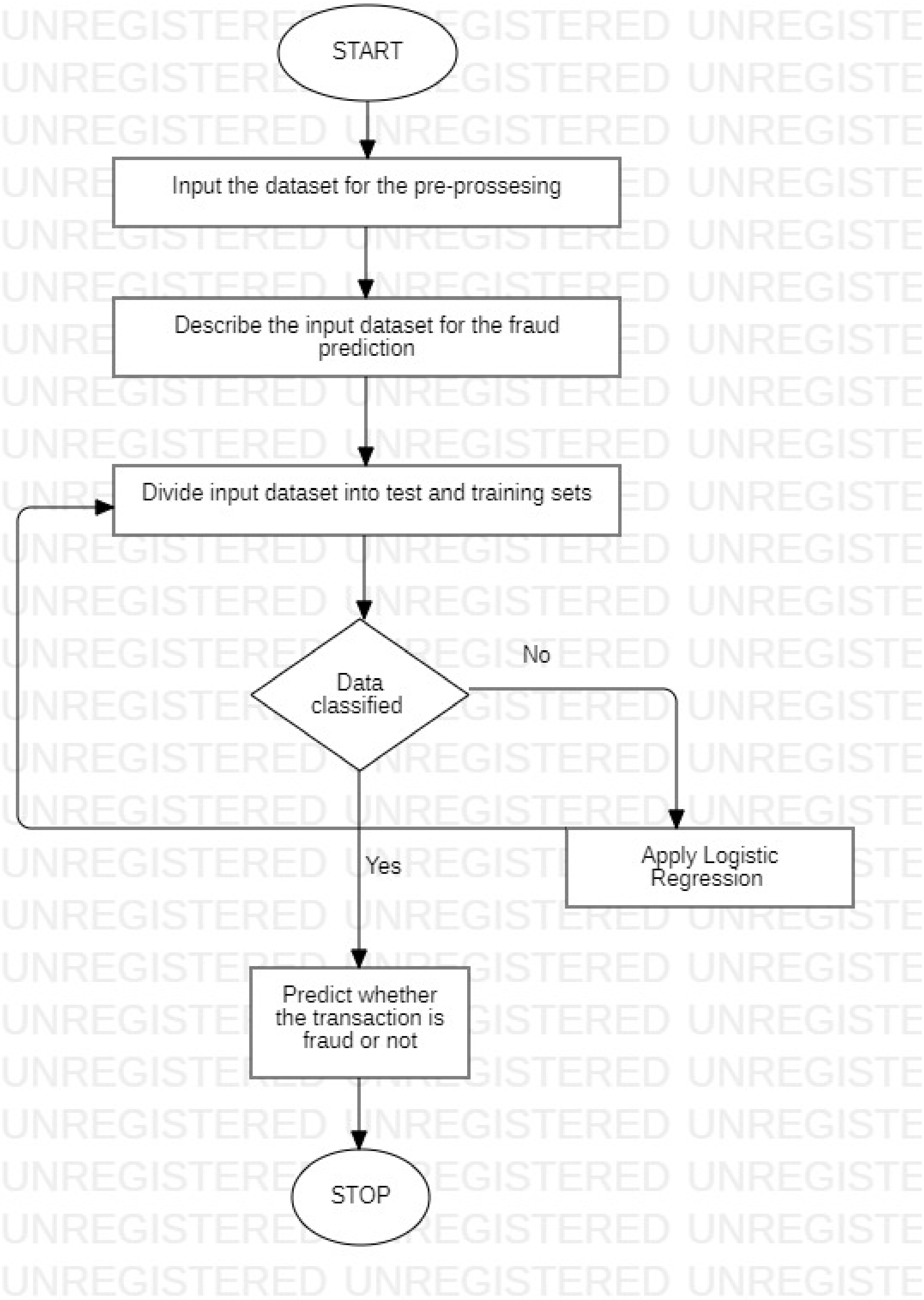
ii)Software Requirement

- Java JDK 10
- Weka
- Eclipse IDE

4.EXPERIMENTAL INVESTIGATIONS

Analysis is made to find out which method is best to predict whether the transactions made are fraud or not

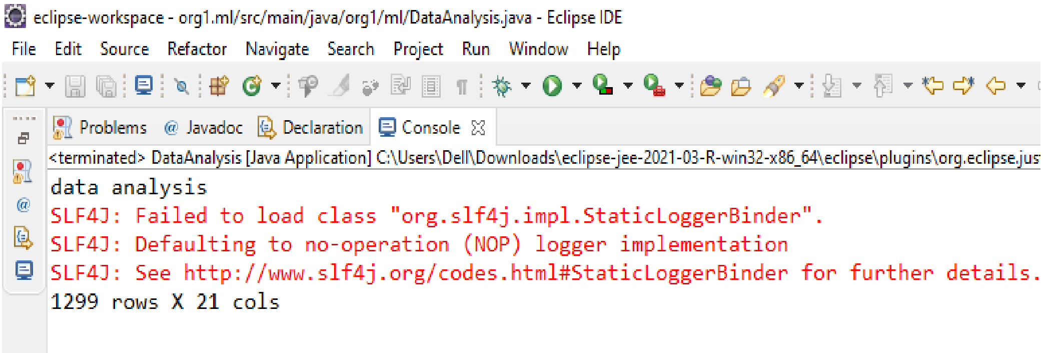
5.FLOWCHART


START
Input the dataset for the pre-prossesing
Describe the input dataset for the fraud prediction
Divide input dataset into test and training sets
Data classified
No
Yes
Apply Logistic Regression
Predict whether the transaction is fraud or not
STOP

UNREGISTERED UNREGISTERED UNREGISTE

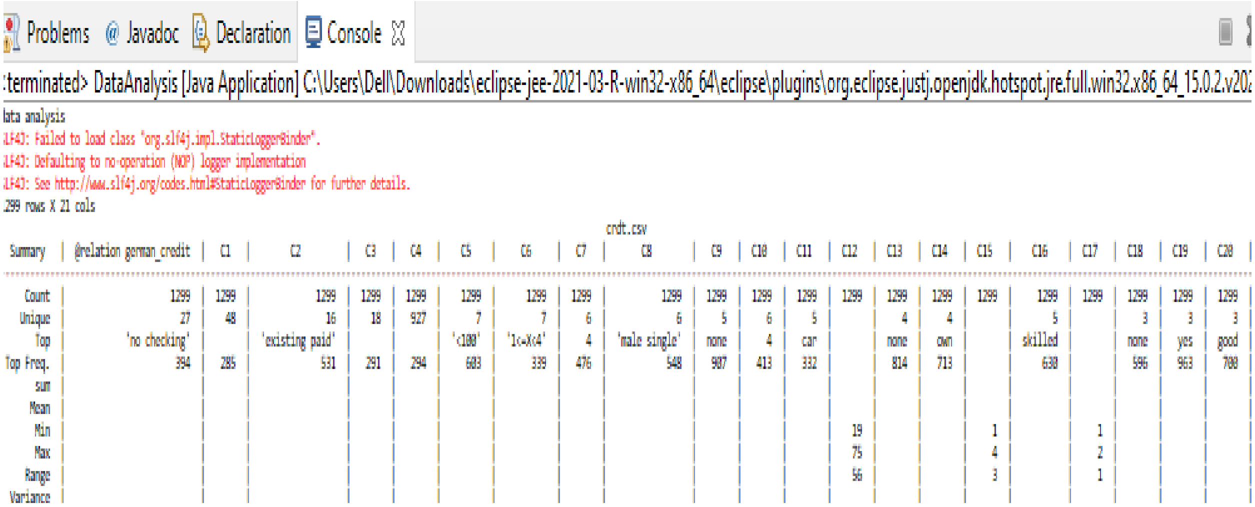
6.RESULT

SHAPE:


eclipse-workspace - org1.ml/src/main/java/org1/ml/DataAnalysis.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Problems @ Javadoc Declaration Console

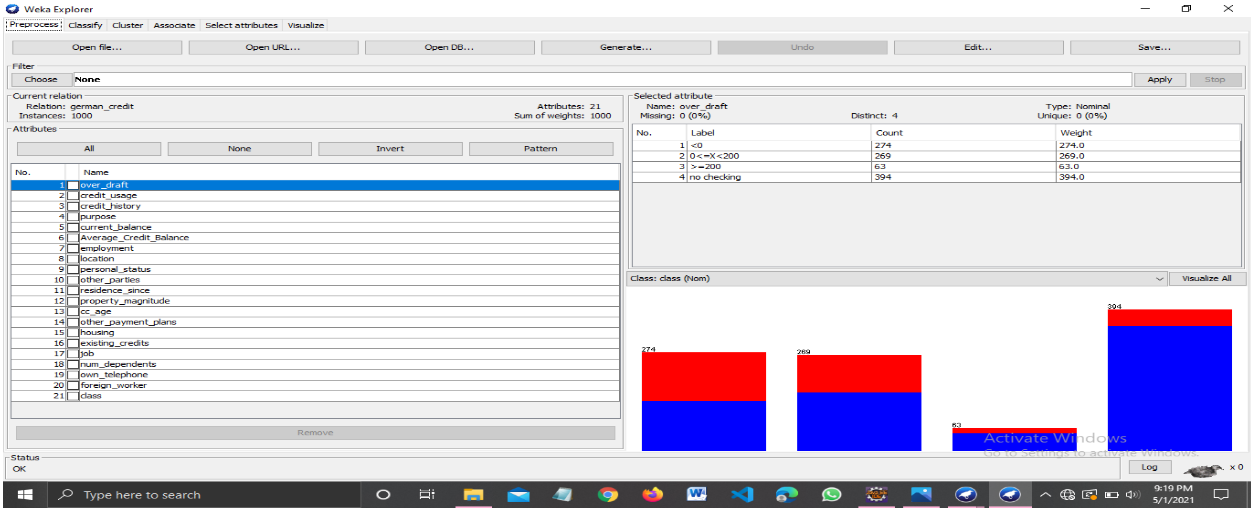
```
<terminated> DataAnalysis [Java Application] C:\Users\Dell\Downloads\eclipse-jee-2021-03-R-win32-x86_64\eclipse\plugins\org.eclipse.jus
data analysis
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
1299 rows X 21 cols
```

Summary



```
:terminated> DataAnalysis [Java Application] C:\Users\Dell\Downloads\eclipse-jee-2021-03-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_15.0.2.v202
lata analysis
iLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
iLF4J: Defaulting to no-operation (NOP) logger implementation
iLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
.299 rows X 21 cols
                                                                                    crdt.csv
 Summary | @relation german_credit | C1 |       C2        | C3 |  C4 |  C5    |  C6     | C7 |      C8       | C9 | C10 | C11 | C12 | C13 | C14 | C15 |   C16   | C17 | C18 | C19 | C20 |
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
   Count |                    1299 |1299|            1299 |1299| 1299|   1299 |    1299 |1299|          1299 |1299| 1299| 1299| 1299| 1299| 1299| 1299|    1299 | 1299| 1299| 1299| 1299|
  Unique |                      27 |  48|              16 |  18|  927|      7 |       7 |   6|             6 |   5|    6|    5|     |    4|    4|     |       5 |     |    3|    3|    3|
     Top |           'no checking' |    | 'existing paid' |    |     | '<100' | '1<=X<4'|   4| 'male single' |none|    4|  car|     | none|  own|     | skilled |     | none|  yes| good|
Top Freq.|                     394 | 285|             531 | 291|  294|    603 |     339 | 476|           548 | 907|  413|  332|     |  814|  713|     |     630 |     |  596|  963|  700|
     sum |                         |    |                 |    |     |        |         |    |               |    |     |     |     |     |     |     |         |     |     |     |     |
    Mean |                         |    |                 |    |     |        |         |    |               |    |     |     |     |     |     |     |         |     |     |     |     |
     Min |                         |    |                 |    |     |        |         |    |               |    |     |     |   19|     |     |    1|         |    1|     |     |     |
     Max |                         |    |                 |    |     |        |         |    |               |    |     |     |   75|     |     |    4|         |    2|     |     |     |
   Range |                         |    |                 |    |     |        |         |    |               |    |     |     |   56|     |     |    3|         |    1|     |     |     |
Variance |                         |    |                 |    |     |        |         |    |               |    |     |     |     |     |     |     |         |     |     |     |     |
```

Structure



Filter: None

Current relation
Relation: german_credit
Instances: 1000
Attributes: 21
Sum of weights: 1000

Attributes

No.		Name
1	□	over_draft
2	□	credit_usage
3	□	credit_history
4	□	purpose
5	□	current_balance
6	□	Average_Credit_Balance
7	□	employment
8	□	location
9	□	personal_status
10	□	other_parties
11	□	residence_since
12	□	property_magnitude
13	□	cc_age
14	□	other_payment_plans
15	□	housing
16	□	existing_credits
17	□	job
18	□	num_dependents
19	□	own_telephone
20	□	foreign_worker
21	□	class

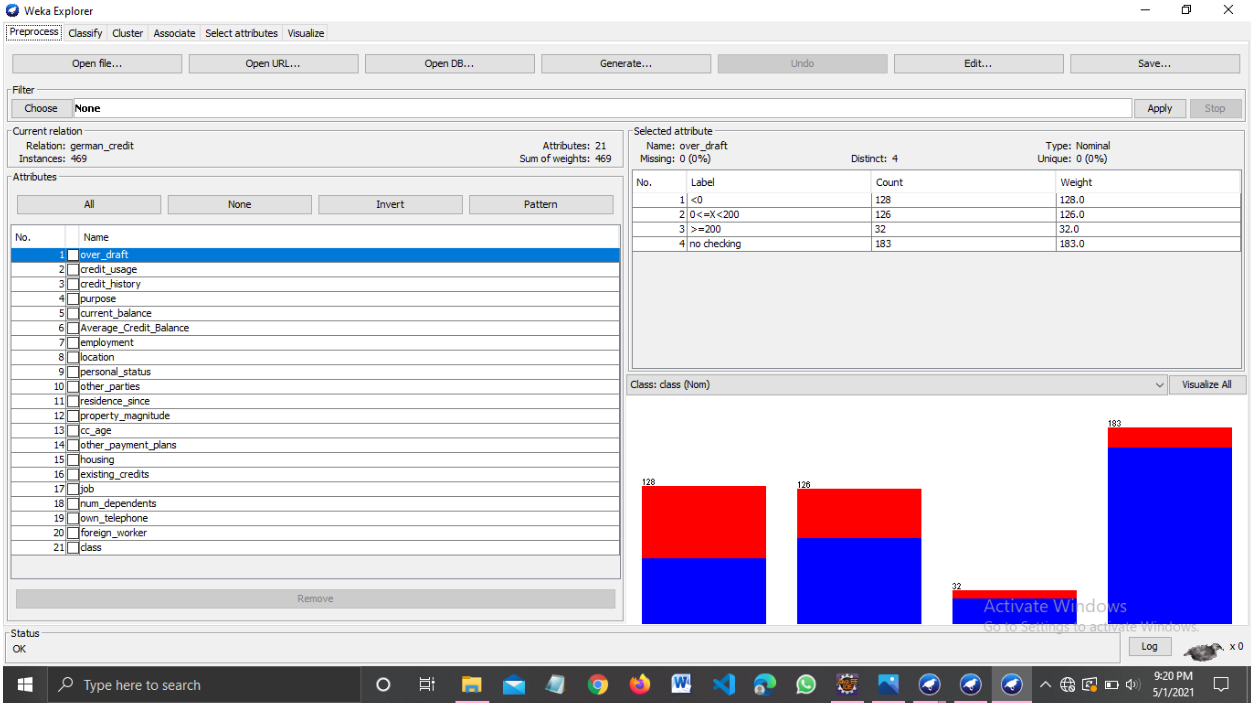
Selected attribute
Name: over_draft
Missing: 0 (0%)
Distinct: 4
Type: Nominal
Unique: 0 (0%)

No.	Label	Count	Weight
1	<0	274	274.0
2	0<=X<200	269	269.0
3	>=200	63	63.0
4	no checking	394	394.0

Class: class (Nom)

Status
OK

Train Data



Weka Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Open file... | Open URL... | Open DB... | Generate... | Undo | Edit... | Save...

Filter

Choose None | Apply | Stop

Current relation

Relation: german_credit Attributes: 21

Instances: 469 Sum of weights: 469

Attributes

All | None | Invert | Pattern

No.	Name
1	over_draft
2	credit_usage
3	credit_history
4	purpose
5	current_balance
6	Average_Credit_Balance
7	employment
8	location
9	personal_status
10	other_parties
11	residence_since
12	property_magnitude
13	cc_age
14	other_payment_plans
15	housing
16	existing_credits
17	job
18	num_dependents
19	own_telephone
20	foreign_worker
21	class

Remove

Selected attribute

Name: over_draft Type: Nominal

Missing: 0 (0%) Distinct: 4 Unique: 0 (0%)

No.	Label	Count	Weight
1	<0	128	128.0
2	0<=X<200	126	126.0
3	>=200	32	32.0
4	no checking	183	183.0

Class: class (Nom) | Visualize All

Activate Windows
Go to Settings to activate Windows.

Status

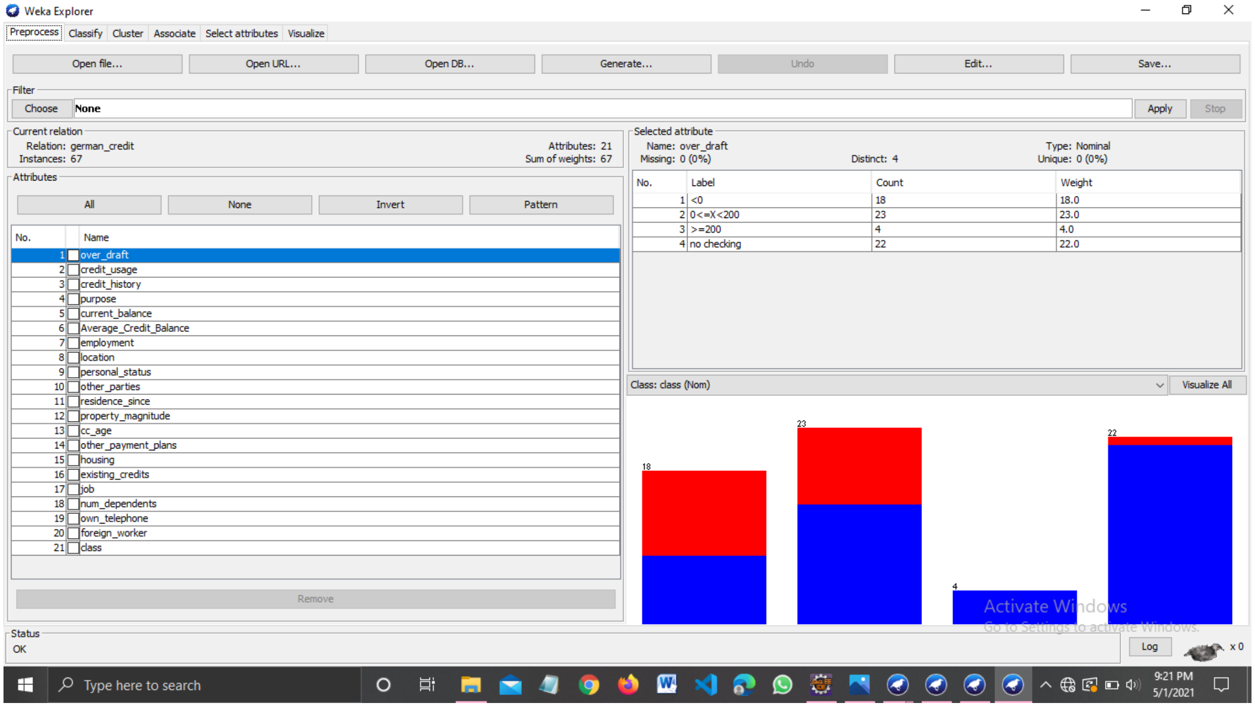
OK

Log x 0

Type here to search

9:20 PM
5/1/2021

Test Data



Weka Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Open file... | Open URL... | Open DB... | Generate... | Undo | Edit... | Save...

Filter

Choose None | Apply | Stop

Current relation

Relation: german_credit Attributes: 21

Instances: 67 Sum of weights: 67

Attributes

All | None | Invert | Pattern

No.	Name
1	over_draft
2	credit_usage
3	credit_history
4	purpose
5	current_balance
6	Average_Credit_Balance
7	employment
8	location
9	personal_status
10	other_parties
11	residence_since
12	property_magnitude
13	cc_age
14	other_payment_plans
15	housing
16	existing_credits
17	job
18	num_dependents
19	own_telephone
20	foreign_worker
21	class

Remove

Selected attribute

Name: over_draft Type: Nominal

Missing: 0 (0%) Distinct: 4 Unique: 0 (0%)

No.	Label	Count	Weight
1	<0	18	18.0
2	0<=X<200	23	23.0
3	>=200	4	4.0
4	no checking	22	22.0

Class: class (Nom) | Visualize All

Activate Windows
Go to Settings to activate Windows.

Status

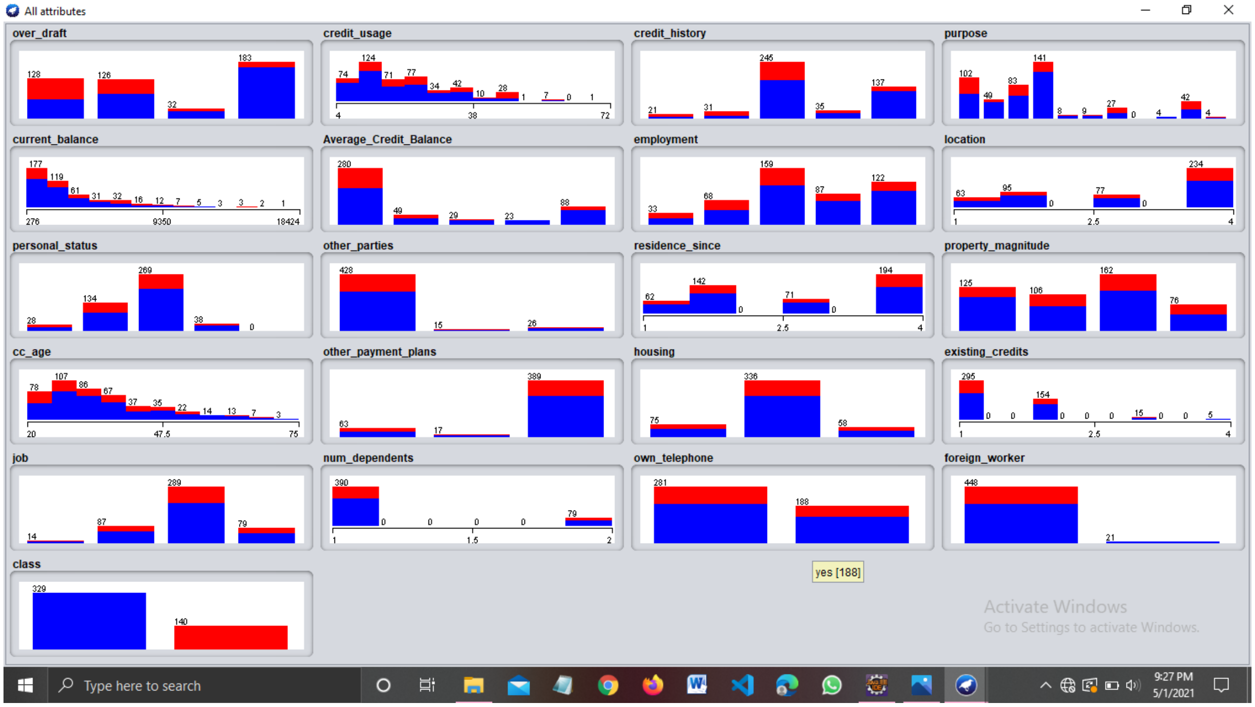
OK

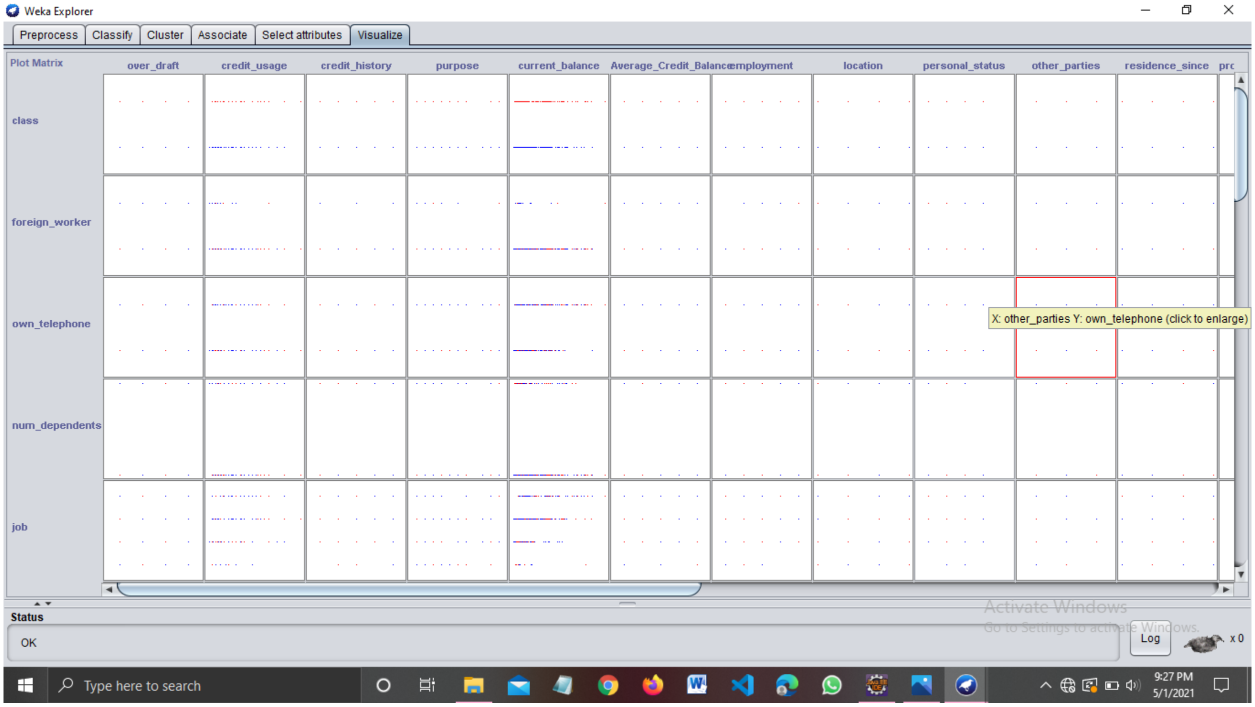
Log x 0

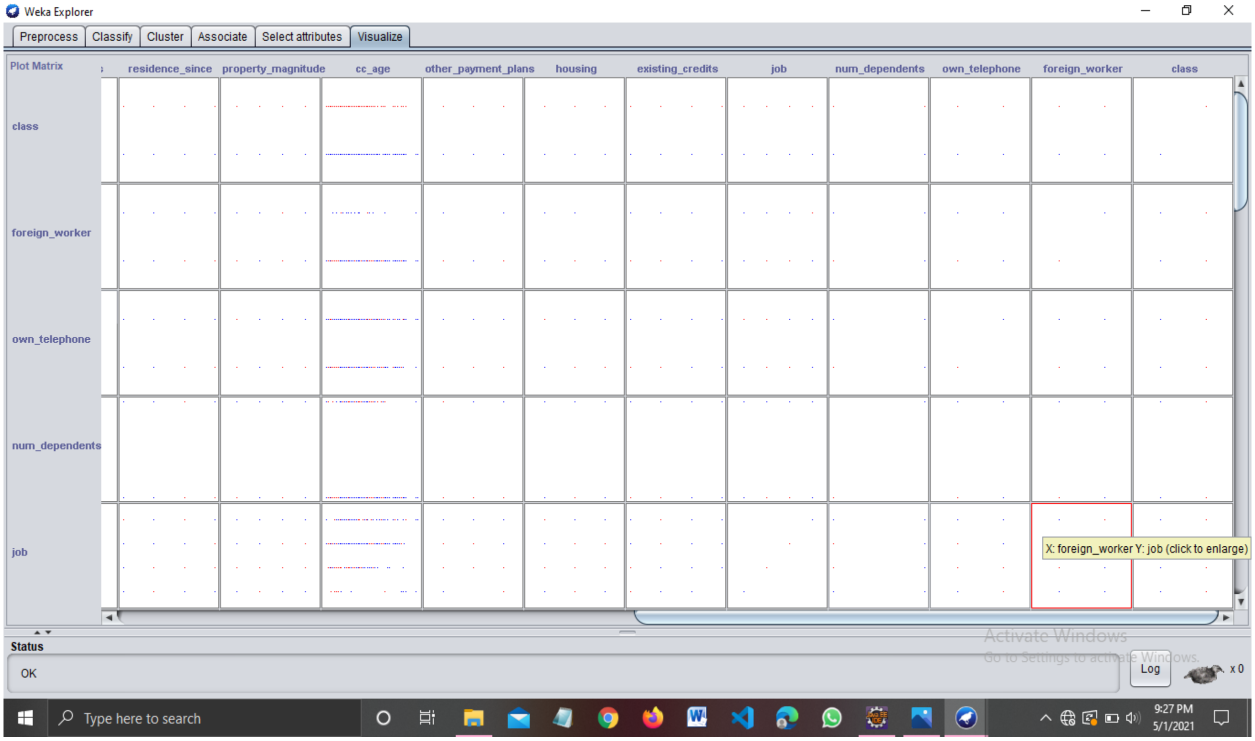
Type here to search

9:21 PM
5/1/2021

Visualization

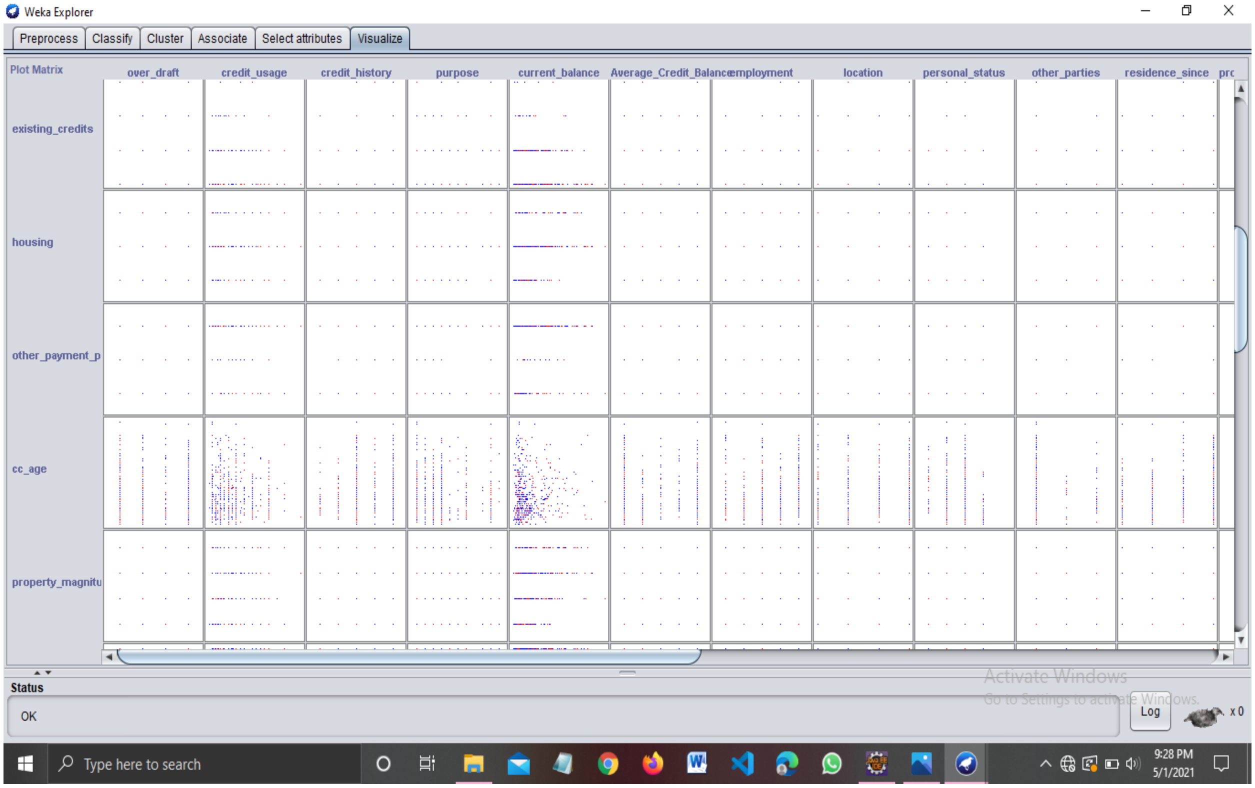






Status

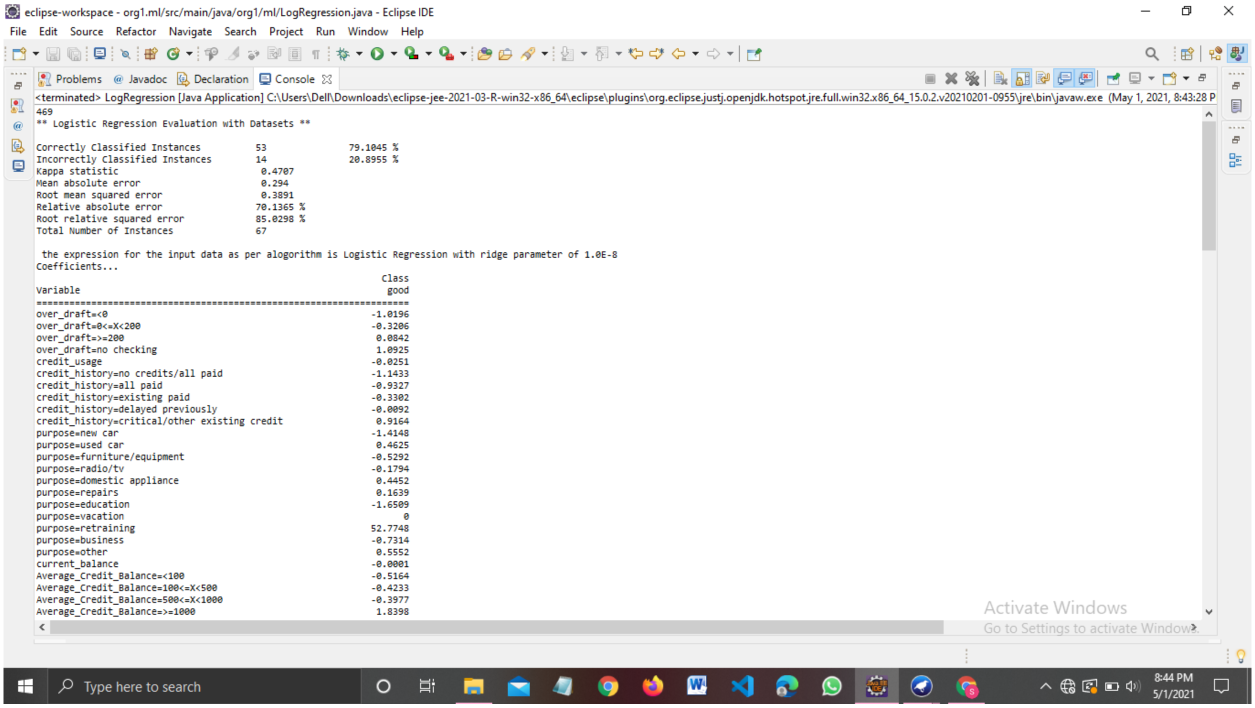
OK



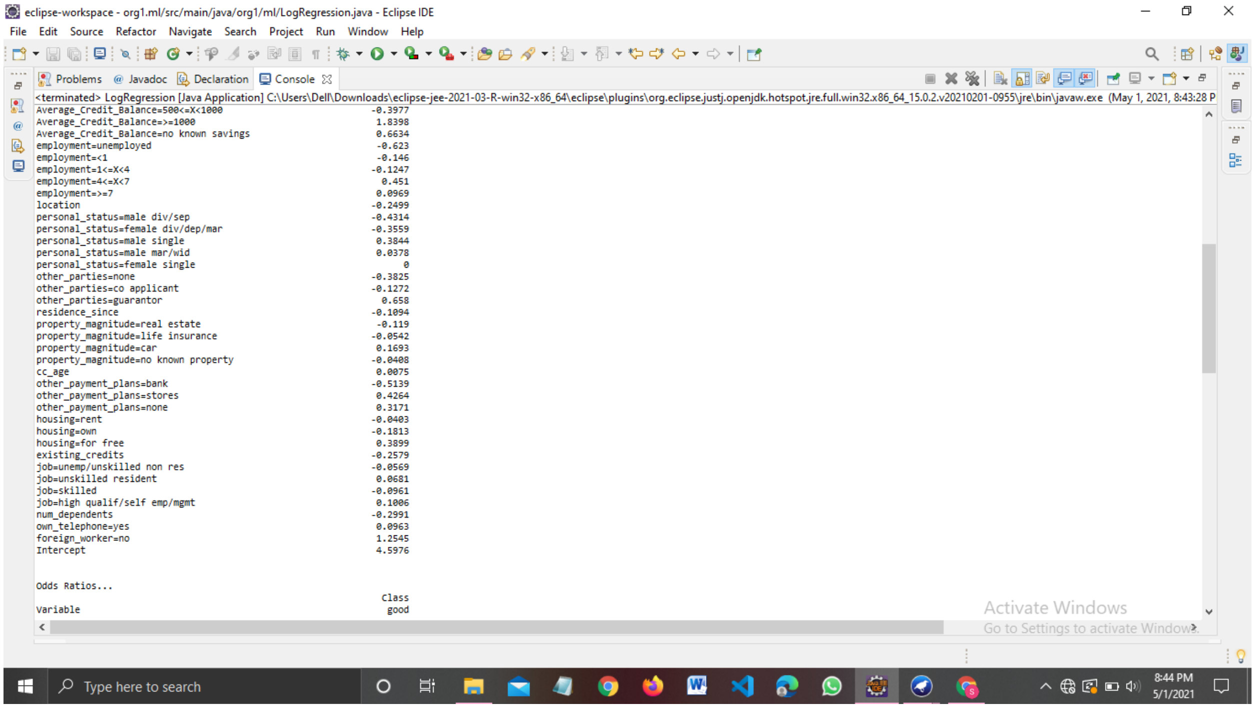
Status

OK

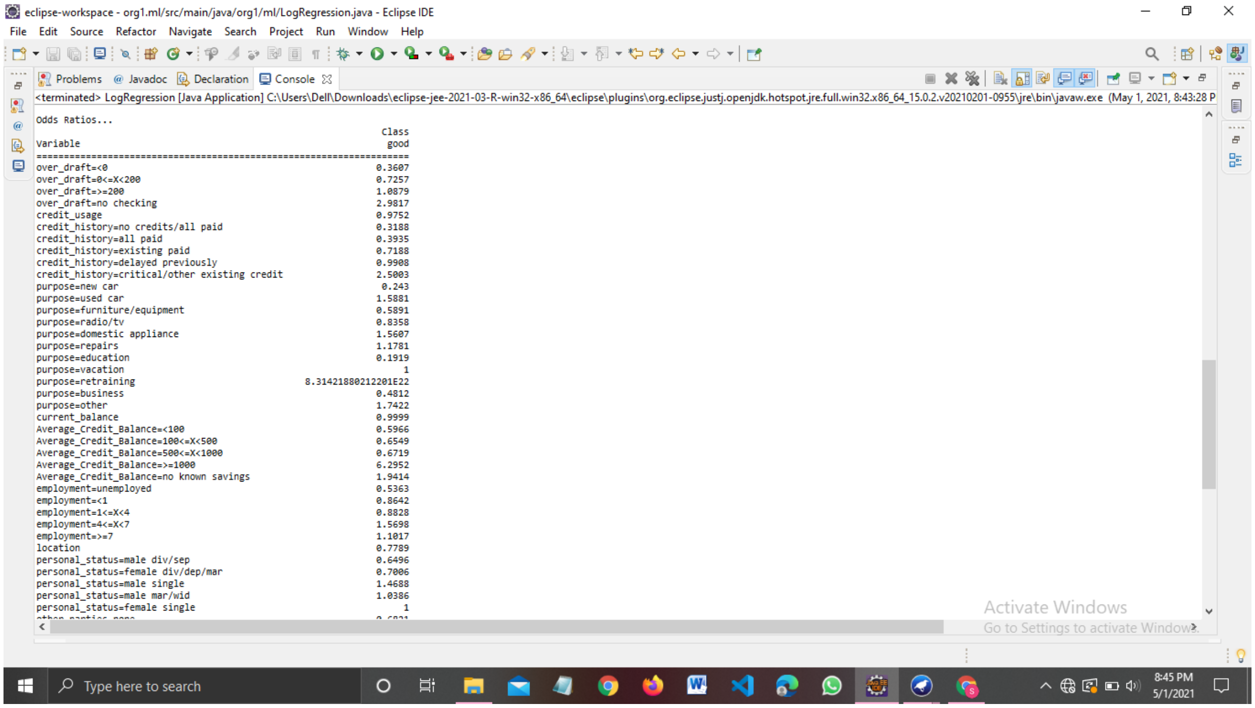
Logistic Regression


eclipse-workspace - org1.ml/src/main/java/org1/ml/LogRegression.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Problems @ Javadoc Declaration Console
Activate Windows
Go to Settings to activate Windows.
Type here to search
8:44 PM
5/1/2021

```
<terminated> LogRegression [Java Application] C:\Users\Dell\Downloads\eclipse-jee-2021-03-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_15.0.2.v20210201-0955\jre\bin\javaw.exe  (May 1, 2021, 8:43:28 P
469
** Logistic Regression Evaluation with Datasets **

Correctly Classified Instances          53               79.1045 %
Incorrectly Classified Instances        14               20.8955 %
Kappa statistic                          0.4707
Mean absolute error                      0.294
Root mean squared error                  0.3891
Relative absolute error                 70.1365 %
Root relative squared error             85.0298 %
Total Number of Instances               67

 the expression for the input data as per alogorithm is Logistic Regression with ridge parameter of 1.0E-8
Coefficients...
                                                     Class
Variable                                              good
==========================================================
over_draft=<0                                      -1.0196
over_draft=0<=X<200                                -0.3206
over_draft=>=200                                    0.0842
over_draft=no checking                              1.0925
credit_usage                                       -0.0251
credit_history=no credits/all paid                 -1.1433
credit_history=all paid                            -0.9327
credit_history=existing paid                       -0.3302
credit_history=delayed previously                  -0.0092
credit_history=critical/other existing credit       0.9164
purpose=new car                                    -1.4148
purpose=used car                                    0.4625
purpose=furniture/equipment                        -0.5292
purpose=radio/tv                                   -0.1794
purpose=domestic appliance                          0.4452
purpose=repairs                                     0.1639
purpose=education                                  -1.6509
purpose=vacation                                         0
purpose=retraining                                 52.7748
purpose=business                                   -0.7314
purpose=other                                       0.5552
current_balance                                    -0.0001
Average_Credit_Balance=<100                        -0.5164
Average_Credit_Balance=100<=X<500                  -0.4233
Average_Credit_Balance=500<=X<1000                 -0.3977
Average_Credit_Balance=>=1000                       1.8398
```

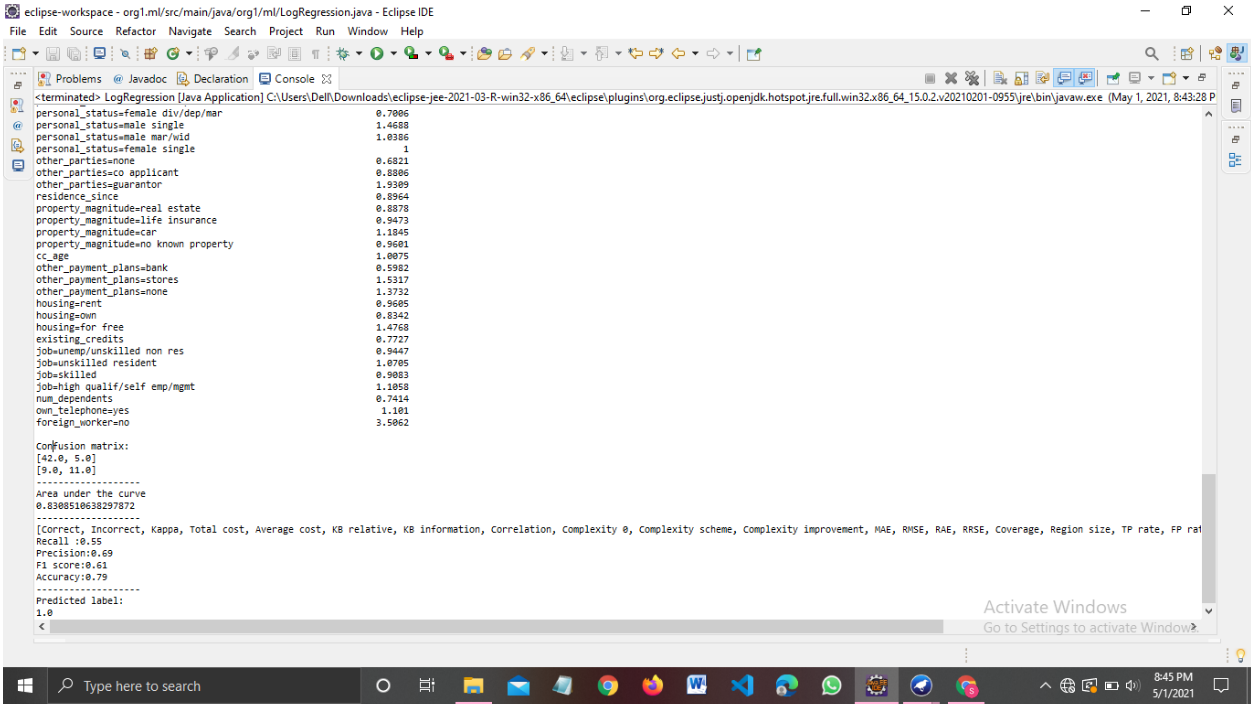

eclipse-workspace - org1.ml/src/main/java/org1/ml/LogRegression.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Problems @ Javadoc Declaration Console
Activate Windows
Go to Settings to activate Windows.
Type here to search
8:44 PM
5/1/2021

```
<terminated> LogRegression [Java Application] C:\Users\Dell\Downloads\eclipse-jee-2021-03-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_15.0.2.v20210201-0955\jre\bin\javaw.exe  (May 1, 2021, 8:43:28 P
Average_Credit_Balance=500<=X<1000                 -0.3977
Average_Credit_Balance=>=1000                       1.8398
Average_Credit_Balance=no known savings             0.6634
employment=unemployed                               -0.623
employment=<1                                       -0.146
employment=1<=X<4                                  -0.1247
employment=4<=X<7                                    0.451
employment=>=7                                      0.0969
location                                           -0.2499
personal_status=male div/sep                       -0.4314
personal_status=female div/dep/mar                 -0.3559
personal_status=male single                         0.3844
personal_status=male mar/wid                        0.0378
personal_status=female single                            0
other_parties=none                                 -0.3825
other_parties=co applicant                         -0.1272
other_parties=guarantor                              0.658
residence_since                                    -0.1094
property_magnitude=real estate                      -0.119
property_magnitude=life insurance                  -0.0542
property_magnitude=car                              0.1693
property_magnitude=no known property               -0.0408
cc_age                                              0.0075
other_payment_plans=bank                           -0.5139
other_payment_plans=stores                          0.4264
other_payment_plans=none                            0.3171
housing=rent                                       -0.0403
housing=own                                        -0.1813
housing=for free                                    0.3899
existing_credits                                   -0.2579
job=unemp/unskilled non res                        -0.0569
job=unskilled resident                              0.0681
job=skilled                                        -0.0961
job=high qualif/self emp/mgmt                       0.1006
num_dependents                                     -0.2991
own_telephone=yes                                   0.0963
foreign_worker=no                                   1.2545
Intercept                                           4.5976


Odds Ratios...
                                                     Class
Variable                                              good
```



eclipse-workspace - org1.ml/src/main/java/org1/ml/LogRegression.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Problems @ Javadoc Declaration Console
Activate Windows
Go to Settings to activate Windows.
Type here to search
8:45 PM
5/1/2021

```
<terminated> LogRegression [Java Application] C:\Users\Dell\Downloads\eclipse-jee-2021-03-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_15.0.2.v20210201-0955\jre\bin\javaw.exe (May 1, 2021, 8:43:28 P
Odds Ratios...
                                                               Class
Variable                                                        good
====================================================================
over_draft=<0                                                 0.3607
over_draft=0<=X<200                                           0.7257
over_draft=>=200                                              1.0879
over_draft=no checking                                        2.9817
credit_usage                                                  0.9752
credit_history=no credits/all paid                            0.3188
credit_history=all paid                                       0.3935
credit_history=existing paid                                  0.7188
credit_history=delayed previously                             0.9908
credit_history=critical/other existing credit                 2.5003
purpose=new car                                                0.243
purpose=used car                                              1.5881
purpose=furniture/equipment                                   0.5891
purpose=radio/tv                                              0.8358
purpose=domestic appliance                                    1.5607
purpose=repairs                                               1.1781
purpose=education                                             0.1919
purpose=vacation                                                   1
purpose=retraining                               8.31421880212201E22
purpose=business                                              0.4812
purpose=other                                                 1.7422
current_balance                                               0.9999
Average_Credit_Balance=<100                                   0.5966
Average_Credit_Balance=100<=X<500                             0.6549
Average_Credit_Balance=500<=X<1000                            0.6719
Average_Credit_Balance=>=1000                                 6.2952
Average_Credit_Balance=no known savings                       1.9414
employment=unemployed                                         0.5363
employment=<1                                                 0.8642
employment=1<=X<4                                             0.8828
employment=4<=X<7                                             1.5698
employment=>=7                                                1.1017
location                                                      0.7789
personal_status=male div/sep                                  0.6496
personal_status=female div/dep/mar                            0.7006
personal_status=male single                                   1.4688
personal_status=male mar/wid                                  1.0386
personal_status=female single                                      1
```



eclipse-workspace - org1.ml/src/main/java/org1/ml/LogRegression.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Problems @ Javadoc Declaration Console
Activate Windows
Go to Settings to activate Windows.
Type here to search
8:45 PM
5/1/2021

```
<terminated> LogRegression [Java Application] C:\Users\Dell\Downloads\eclipse-jee-2021-03-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_15.0.2.v20210201-0955\jre\bin\javaw.exe (May 1, 2021, 8:43:28 P
personal_status=female div/dep/mar                            0.7006
personal_status=male single                                   1.4688
personal_status=male mar/wid                                  1.0386
personal_status=female single                                      1
other_parties=none                                            0.6821
other_parties=co applicant                                    0.8806
other_parties=guarantor                                       1.9309
residence_since                                               0.8964
property_magnitude=real estate                                0.8878
property_magnitude=life insurance                             0.9473
property_magnitude=car                                        1.1845
property_magnitude=no known property                          0.9601
cc_age                                                        1.0075
other_payment_plans=bank                                      0.5982
other_payment_plans=stores                                    1.5317
other_payment_plans=none                                      1.3732
housing=rent                                                  0.9605
housing=own                                                   0.8342
housing=for free                                              1.4768
existing_credits                                              0.7727
job=unemp/unskilled non res                                   0.9447
job=unskilled resident                                        1.0705
job=skilled                                                   0.9083
job=high qualif/self emp/mgmt                                 1.1058
num_dependents                                                0.7414
own_telephone=yes                                              1.101
foreign_worker=no                                             3.5062

Confusion matrix:
[42.0, 5.0]
[9.0, 11.0]
-------------------
Area under the curve
0.8308510638297872
-------------------
[Correct, Incorrect, Kappa, Total cost, Average cost, KB relative, KB information, Correlation, Complexity 0, Complexity scheme, Complexity improvement, MAE, RMSE, RAE, RRSE, Coverage, Region size, TP rate, FP rat
Recall :0.55
Precision:0.69
F1 score:0.61
Accuracy:0.79
-------------------
Predicted label:
1.0
```

Final output:

For Default threshold 0.5

Confusion matrix:

[42.0, 5.0]

[9.0, 11.0]

Area under the curve is 0.8308510638297872

[Correct, Incorrect, Kappa, Total cost, Average cost, KB relative, KB information, Correlation, Complexity 0, Complexity scheme, Complexity improvement, MAE, RMSE, RAE, RRSE, Coverage, Region size, TP rate, FP rate, Precision, Recall, F-measure, MCC, ROC area, PRC area]

Recall :0.55

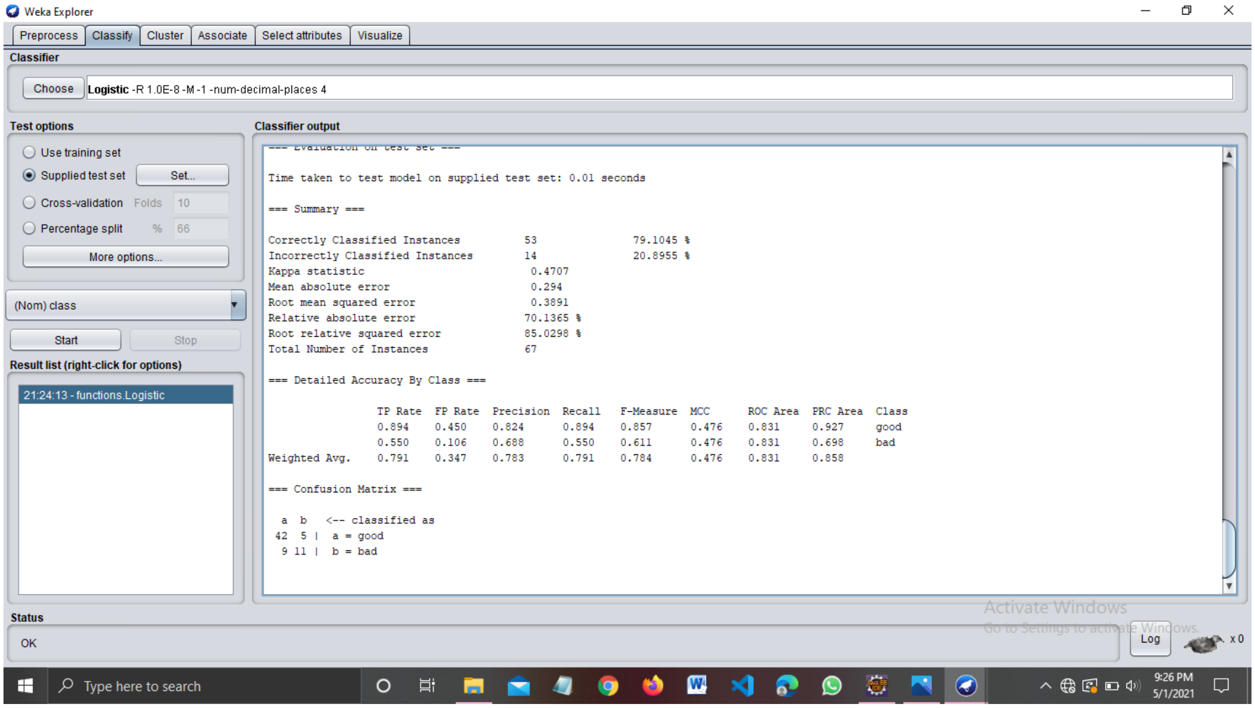
Precision:0.69

F1 score:0.61

Accuracy:0.79

Predicted label: 1.0(FRAUD)

WEKA:



Classifier

Logistic -R 1.0E-8 -M -1 -num-decimal-places 4

Test options

○ Use training set
● Supplied test set
○ Cross-validation Folds 10
○ Percentage split % 66

(Nom) class

Result list (right-click for options)

21:24:13 - functions.Logistic

Classifier output

```
=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correctly Classified Instances          53               79.1045 %
Incorrectly Classified Instances        14               20.8955 %
Kappa statistic                          0.4707
Mean absolute error                      0.294
Root mean squared error                  0.3891
Relative absolute error                 70.1365 %
Root relative squared error             85.0298 %
Total Number of Instances               67

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.894    0.450    0.824      0.894    0.857      0.476    0.831     0.927     good
                 0.550    0.106    0.688      0.550    0.611      0.476    0.831     0.698     bad
Weighted Avg.    0.791    0.347    0.783      0.791    0.784      0.476    0.831     0.858

=== Confusion Matrix ===

  a  b   <-- classified as
 42  5 |  a = good
  9 11 |  b = bad
```

Status

OK

7.ADVANTAGES & DISADVANTAGES

Advantages:

- Logistic regression is easier to implement, interpret and very efficient to train.
- It is very fast at classifying unknow records.
- Good accuracy for many simple data sets and it performs well when the dataset ie linearly separable

Disadvantages

- The major limitation of Logistic Regression is the assumption of linearity between the dependent and independent variables.
- It can only predict discrete functions. Hence, the dependent variable of Logistic Regression is bound to the discrete number set.

8.APPLICATIONS

This solution can be applied for Fraud prediction in banking and finance.

9.CONCLUSION

Area under the curve is 0.8308510638297872

- Recall: 0.55
- Precision: 0.69
- f1 score: 0.61
- Accurcy: 0.79
- Prediction label: 1.0

Overall model could be improved with more data

10.FUTURE SCOPE

The most popular area of current fraud prediction research has been in credit card. Custom models or targeted modelling enhance the accuracy of fraud prediction by pulling customer-specific data points. In future, this technique will be standardized across all card associations and banks.

11.BIBILOGRAPHY

https://www.kaggle.com/mlg-ulb/creditcardfraud

APPENDIX

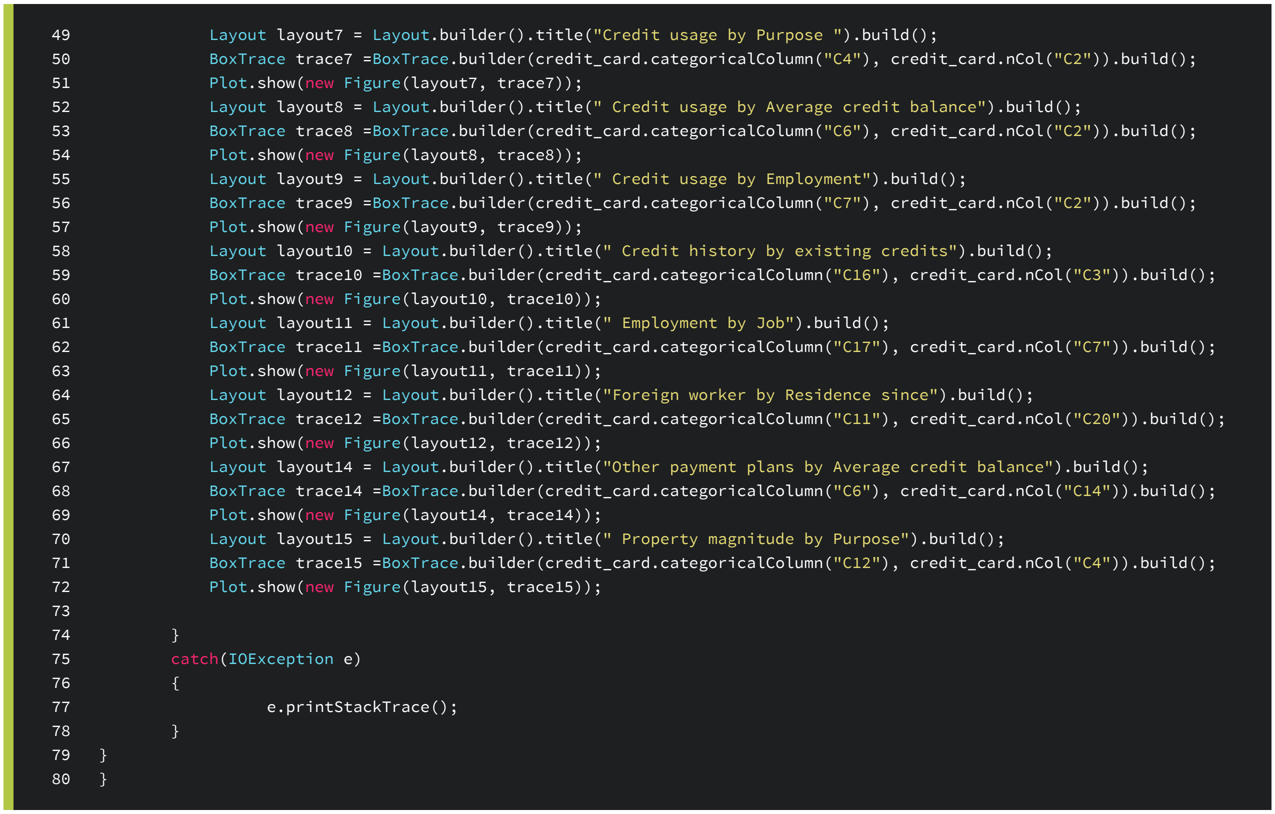
Source code : pom.xml

```
<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>org1.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>
```

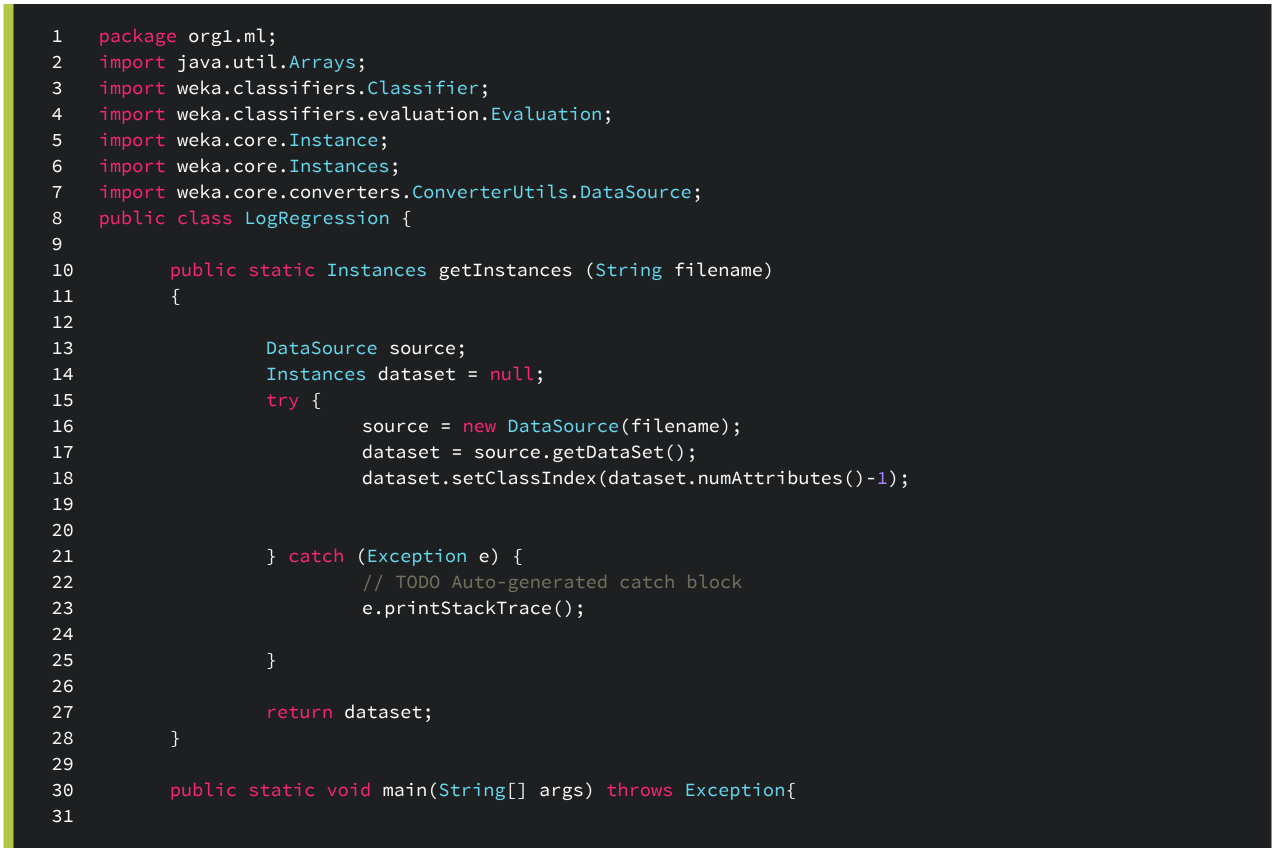
Data Analysis: (DataAnalysis.java)

```
package org1.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 credit_card=
Table.read().csv("C:\\Users\\Dell\\eclipse-workspace\\org1.ml\\src\\main\\java\\org1\\ml\\crdt.csv");

                System.out.println(credit_card.shape());
                System.out.println(credit_card.first(7));
                System.out.println(credit_card.last(7));
                System.out.println(credit_card.summary());
                System.out.println(credit_card.structure());
////   Histogram
                Layout layout1 = Layout.builder().title("Distribution Over draft").build();
            HistogramTrace trace1 = HistogramTrace.builder(credit_card.nCol("C1")).build();
            Plot.show(new Figure(layout1, trace1));
            Layout layout2 = Layout.builder().title("Distribution of  Credit usage").build();
            HistogramTrace trace2 = HistogramTrace.builder(credit_card.nCol("C2")).build();
            Plot.show(new Figure(layout2, trace2));
            Layout layout3 = Layout.builder().title("Distribution of Credit history").build();
            HistogramTrace trace3 = HistogramTrace.builder(credit_card.nCol("C3")).build();
            Plot.show(new Figure(layout3, trace3));
            Layout layout4 = Layout.builder().title("Distribution of Purpose").build();
            HistogramTrace trace4 = HistogramTrace.builder(credit_card.nCol("C4")).build();
            Plot.show(new Figure(layout4, trace4));
            Layout layout5 = Layout.builder().title("Distribution of Current balance").build();
            HistogramTrace trace5 = HistogramTrace.builder(credit_card.nCol("C5")).build();
            Plot.show(new Figure(layout5, trace5));
            Layout layout6 = Layout.builder().title("Distribution of  Average credit balance").build();
            HistogramTrace trace6 = HistogramTrace.builder(credit_card.nCol("C6")).build();
            Plot.show(new Figure(layout6, trace6));
            Layout layout13 = Layout.builder().title("Distribution of Employment").build();
            HistogramTrace trace13 = HistogramTrace.builder(credit_card.nCol("C7")).build();
            Plot.show(new Figure(layout13, trace13));


```

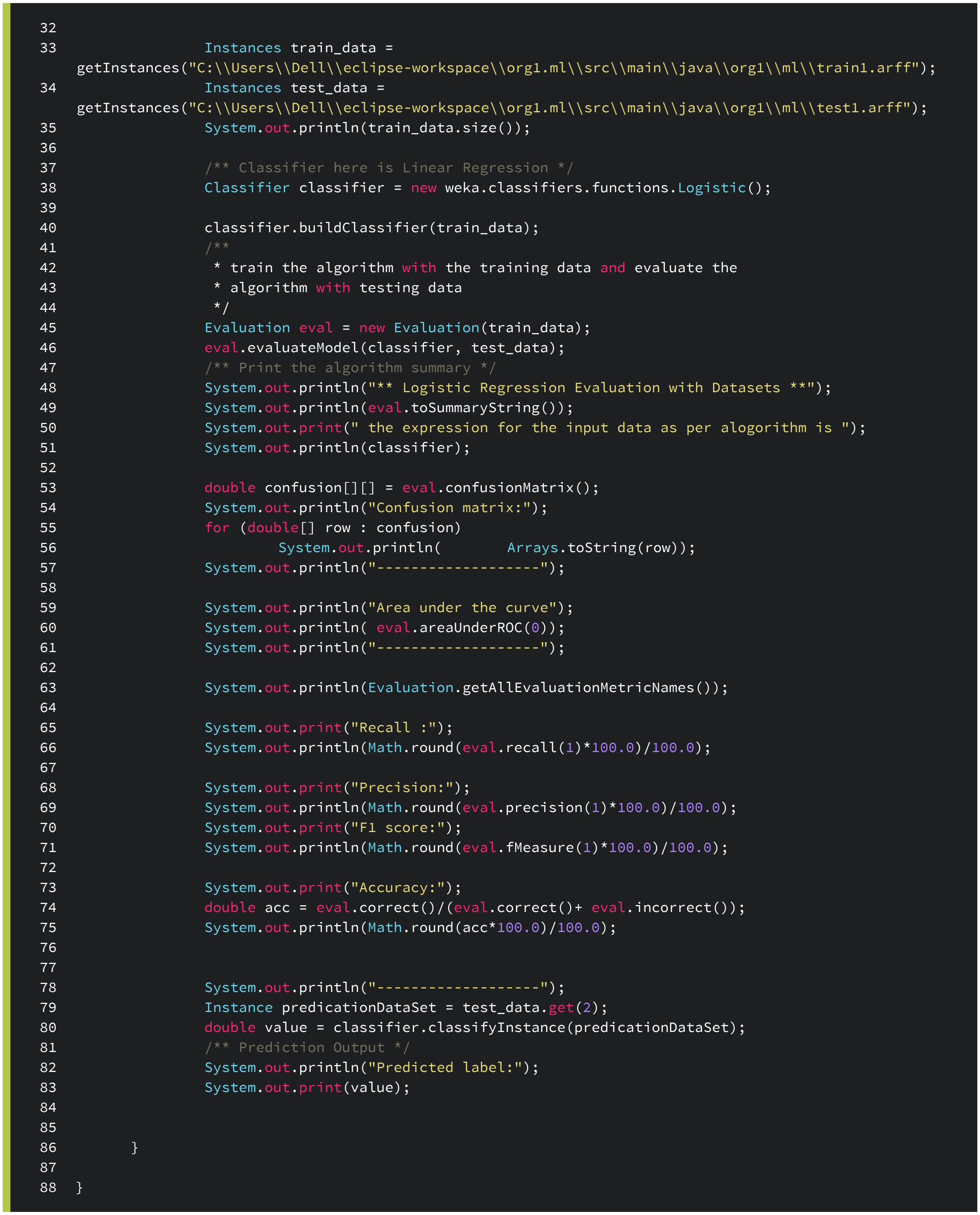


```java
            Layout layout7 = Layout.builder().title("Credit usage by Purpose ").build();
            BoxTrace trace7 =BoxTrace.builder(credit_card.categoricalColumn("C4"), credit_card.nCol("C2")).build();
            Plot.show(new Figure(layout7, trace7));
            Layout layout8 = Layout.builder().title(" Credit usage by Average credit balance").build();
            BoxTrace trace8 =BoxTrace.builder(credit_card.categoricalColumn("C6"), credit_card.nCol("C2")).build();
            Plot.show(new Figure(layout8, trace8));
            Layout layout9 = Layout.builder().title(" Credit usage by Employment").build();
            BoxTrace trace9 =BoxTrace.builder(credit_card.categoricalColumn("C7"), credit_card.nCol("C2")).build();
            Plot.show(new Figure(layout9, trace9));
            Layout layout10 = Layout.builder().title(" Credit history by existing credits").build();
            BoxTrace trace10 =BoxTrace.builder(credit_card.categoricalColumn("C16"), credit_card.nCol("C3")).build();
            Plot.show(new Figure(layout10, trace10));
            Layout layout11 = Layout.builder().title(" Employment by Job").build();
            BoxTrace trace11 =BoxTrace.builder(credit_card.categoricalColumn("C17"), credit_card.nCol("C7")).build();
            Plot.show(new Figure(layout11, trace11));
            Layout layout12 = Layout.builder().title("Foreign worker by Residence since").build();
            BoxTrace trace12 =BoxTrace.builder(credit_card.categoricalColumn("C11"), credit_card.nCol("C20")).build();
            Plot.show(new Figure(layout12, trace12));
            Layout layout14 = Layout.builder().title("Other payment plans by Average credit balance").build();
            BoxTrace trace14 =BoxTrace.builder(credit_card.categoricalColumn("C6"), credit_card.nCol("C14")).build();
            Plot.show(new Figure(layout14, trace14));
            Layout layout15 = Layout.builder().title(" Property magnitude by Purpose").build();
            BoxTrace trace15 =BoxTrace.builder(credit_card.categoricalColumn("C12"), credit_card.nCol("C4")).build();
            Plot.show(new Figure(layout15, trace15));

        }
        catch(IOException e)
        {
                e.printStackTrace();
        }
}
}
```

Logistic Regression(LogRegression.java)



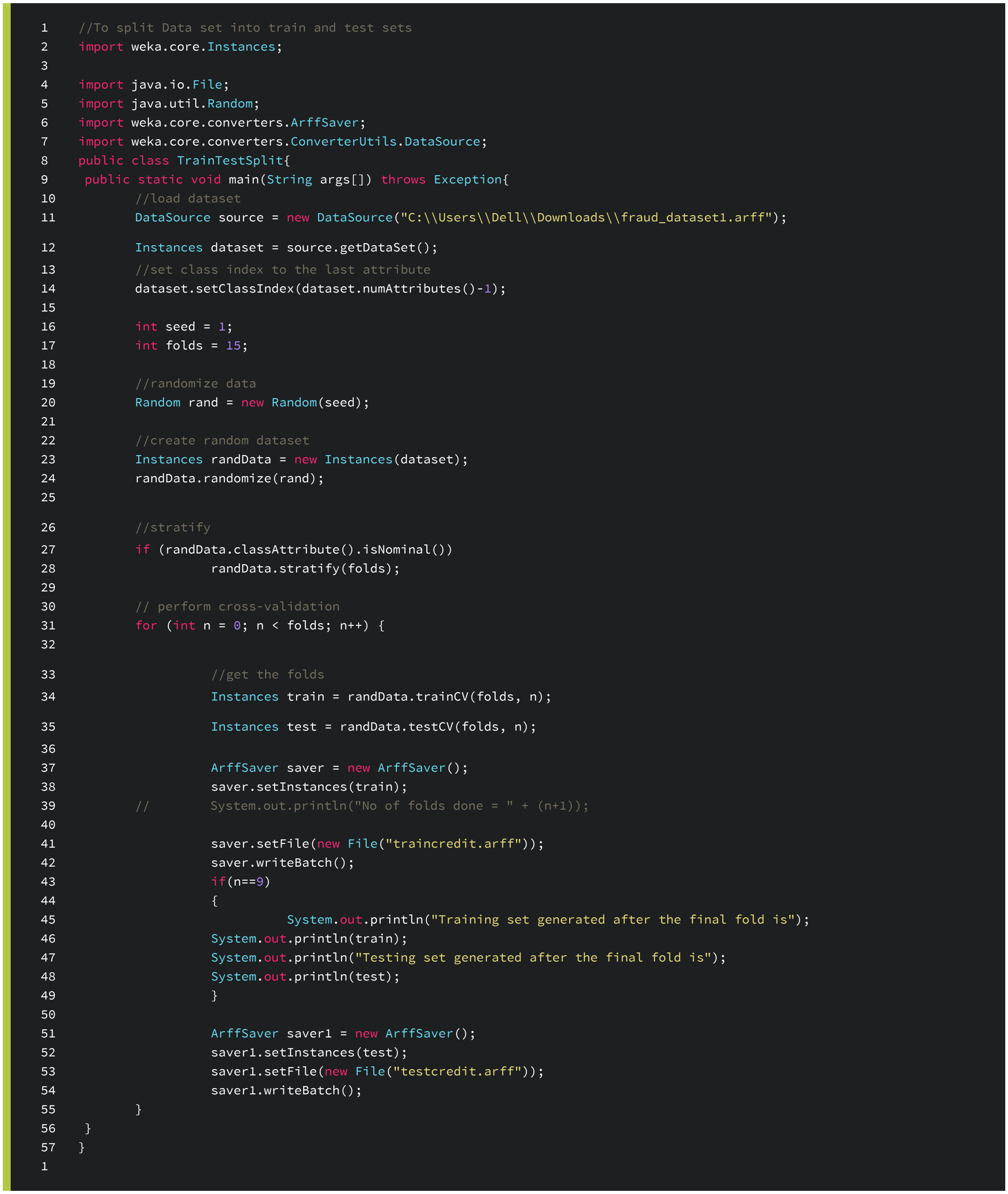
```java
package org1.ml;
import java.util.Arrays;
import weka.classifiers.Classifier;
import weka.classifiers.evaluation.Evaluation;
import weka.core.Instance;
import weka.core.Instances;
import weka.core.converters.ConverterUtils.DataSource;
public class LogRegression {

        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{

```



```java

               Instances train_data = getInstances("C:\\Users\\Dell\\eclipse-workspace\\org1.ml\\src\\main\\java\\org1\\ml\\train1.arff");
               Instances test_data = getInstances("C:\\Users\\Dell\\eclipse-workspace\\org1.ml\\src\\main\\java\\org1\\ml\\test1.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 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(Evaluation.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);


      }

}
```

Java code to split dataset into Test and Train

data:(TrainTestSplit.java)



```java
//To split Data set into train and test sets
import weka.core.Instances;

import java.io.File;
import java.util.Random;
import weka.core.converters.ArffSaver;
import weka.core.converters.ConverterUtils.DataSource;
public class TrainTestSplit{
 public static void main(String args[]) throws Exception{
        //load dataset
        DataSource source = new DataSource("C:\\Users\\Dell\\Downloads\\fraud_dataset1.arff");
        Instances dataset = source.getDataSet();
        //set class index to the last attribute
        dataset.setClassIndex(dataset.numAttributes()-1);

        int seed = 1;
        int folds = 15;

        //randomize data
        Random rand = new Random(seed);

        //create random dataset
        Instances randData = new Instances(dataset);
        randData.randomize(rand);

        //stratify
        if (randData.classAttribute().isNominal())
                randData.stratify(folds);

        // perform cross-validation
        for (int n = 0; n < folds; n++) {

                //get the folds
                Instances train = randData.trainCV(folds, n);
                Instances test = randData.testCV(folds, n);

                ArffSaver saver = new ArffSaver();
                saver.setInstances(train);
        //      System.out.println("No of folds done = " + (n+1));

                saver.setFile(new File("traincredit.arff"));
                saver.writeBatch();
                if(n==9)
                {
                        System.out.println("Training set generated after the final fold is");
                System.out.println(train);
                System.out.println("Testing set generated after the final fold is");
                System.out.println(test);
                }

                ArffSaver saver1 = new ArffSaver();
                saver1.setInstances(test);
                saver1.setFile(new File("testcredit.arff"));
                saver1.writeBatch();
        }
 }
}

```