CLASS: TE CMPN A
NAME: REBECCA DIAS

ROLL NO. : 19
PID: 182027

EXPERIMENT : 8A

Classification Algorithm using WEKA

Aim: - To Implement all the Classification algorithm using WEKA (Naïve Bayes, ID3, Regression).

S/W Requirement: - WEKA Tool

Theory: -

1.Application of all different Classification algorithm: Naïve Bayes, ID3, Regression:

•Naïve Bayes: As this algorithm is fast and efficient, you can use it to make real-time predictions.

•This algorithm is popular for multi-class predictions. You can find the probability of multiple target classes easily by using this algorithm.

•Email services (like Gmail) use this algorithm to figure out whether an email is a spam or not. This algorithm is excellent for spam filtering.

•Its assumption of feature independence, and its effectiveness in solving multi-class problems, makes it perfect for performing Sentiment Analysis. Sentiment Analysis refers to the identification of positive or negative sentiments of a target group (customers, audience, etc.)

•Collaborative Filtering and the Naive Bayes algorithm work together to build recommendation systems. These systems use data mining and machine learning to predict if the user would like a particular resource or not.

ID3:

The ID3 algorithm is a decision tree classification algorithm based on information entropy.

Regression:

1. Predictive Analytics:

Predictive analytics i.e. forecasting future opportunities and risks is the most prominent application of regression analysis in business. Demand analysis, for instance, predicts the number of items which a consumer will probably purchase. However, demand is not the only dependent variable when it comes to business.

2. Operation Efficiency:

Regression models can also be used to optimize business processes. A factory manager, for example, can create a statistical model to understand the impact of oven temperature on the shelf life of the cookies baked in those ovens.

3. Supporting Decisions:

Businesses today are overloaded with data on finances, operations and customer purchases. Increasingly, executives are now leaning on data analytics to make informed business decisions that have statistical significance, thus eliminating the intuition and gut feel. RA can bring a scientific angle to the management of any businesses.

2.Advantageand Disadvantageofalldifferent Classification Algorithms:

- Naïve Bayes:

Advantages:

1.This algorithm works very fast and can easily predict the class of a test dataset.

2. You can use it to solve multi-class prediction problems as it's quite useful with them.

3. Naive Bayes classifier performs better than other models with less training data if the assumption of independence of features holds.

Disadvantages:

1.If your test data set has a categorical variable of a category that wasn't present in the training data set, the Naive Bayes model will assign it zero probability and won't be able to make any predictions in this regard. This phenomenon is called 'Zero Frequency,' and you'll have to use a smoothing technique to solve this problem.

2. This algorithm is also notorious as a lousy estimator. So, you shouldn't take the probability outputs of 'predict_proba' too seriously.

- ID3:

Advantages:

1. Assuming that the space contains all decision trees,the search space is complete.

2. Good robustness and not affected by noise.

3. The instances with missing attribute values can be trained.

Disadvantages:

1. ID3 only considers sub-type features, but does not consider continuous features, such as length and density are continuous values, which cannot be used in ID3. This greatly limits the use of ID3.

2. The ID3 algorithm does not consider missing values.

3. The problem of overfitting is not considered.

- Regression:

Advantages:

1. Linear Regression is a very simple algorithm that can be implemented very easily to give satisfactory results.

2. Linear regression fits linearly seperable datasets almost perfectly and is often used to find the nature of the relationship between variables.

3. Regularization is a technique that can be easily implemented and is capable of effectively reducing the complexity of a function so as to reduce the risk of overfitting.

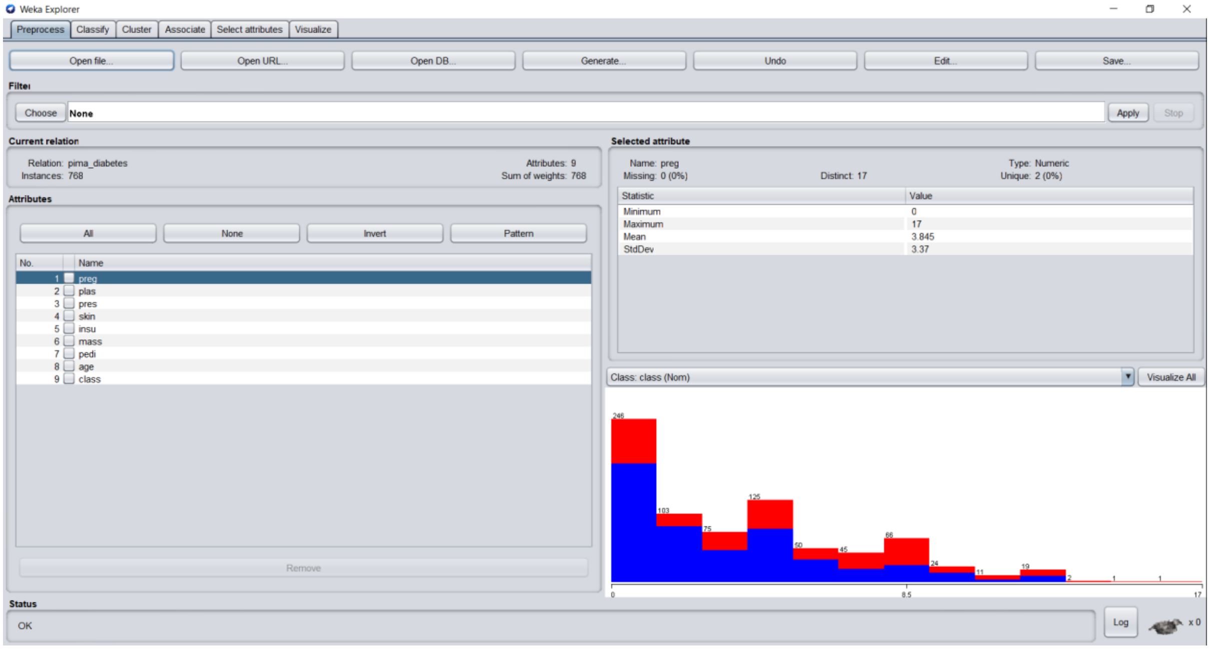
Disadvantages:

1. Underfitting: A sitiuation that arises when a machine learning model fails to capture the data properly.This typically occurs when the hypothesis function cannot fit the data well.

2. Outliers of a data set are anomalies or extreme values that deviate from the other data points of the distribution.Data outliers can damage the performance of a machine learning model drastically and can often lead to models with low accuracy.

3. Outliers can have a very big impact on linear regression's performance and hence they must be dealt with appropriately before linear regression is applied on the dataset.

Implementation: -



Filter

Choose None

Current relation

Relation: pima_diabetes
Instances: 768
Attributes: 9
Sum of weights: 768

Attributes

No.		Name
1	□	preg
2	□	plas
3	□	pres
4	□	skin
5	□	insu
6	□	mass
7	□	pedi
8	□	age
9	□	class

Selected attribute

Name: preg
Missing: 0 (0%)
Distinct: 17
Type: Numeric
Unique: 2 (0%)

Statistic	Value
Minimum	0
Maximum	17
Mean	3.845
StdDev	3.37

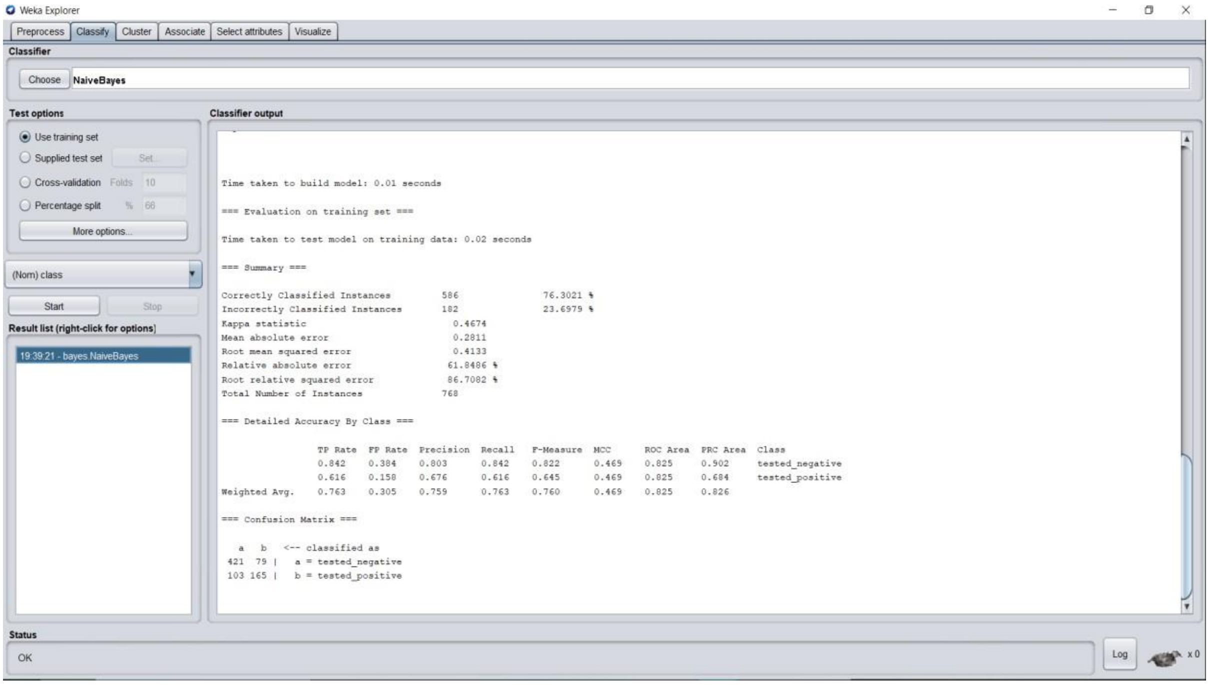
Class: class (Nom)

Status

OK

OUTPUT: -

i) Naïve Bayes:



Classifier

Choose NaiveBayes

Test options

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

(Nom) class

Result list (right-click for options)

19:39:21 - bayes.NaiveBayes

Classifier output

```
Time taken to build model: 0.01 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0.02 seconds

=== Summary ===

Correctly Classified Instances         586               76.3021 %
Incorrectly Classified Instances       182               23.6979 %
Kappa statistic                          0.4674
Mean absolute error                      0.2811
Root mean squared error                  0.4133
Relative absolute error                 61.8486 %
Root relative squared error             86.7082 %
Total Number of Instances              768

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.842    0.384    0.803      0.842    0.822      0.469    0.825     0.902     tested_negative
                 0.616    0.158    0.676      0.616    0.645      0.469    0.825     0.684     tested_positive
Weighted Avg.    0.763    0.305    0.759      0.763    0.760      0.469    0.825     0.826

=== Confusion Matrix ===

   a   b   <-- classified as
 421  79 |   a = tested_negative
 103 165 |   b = tested_positive
```

Status

OK

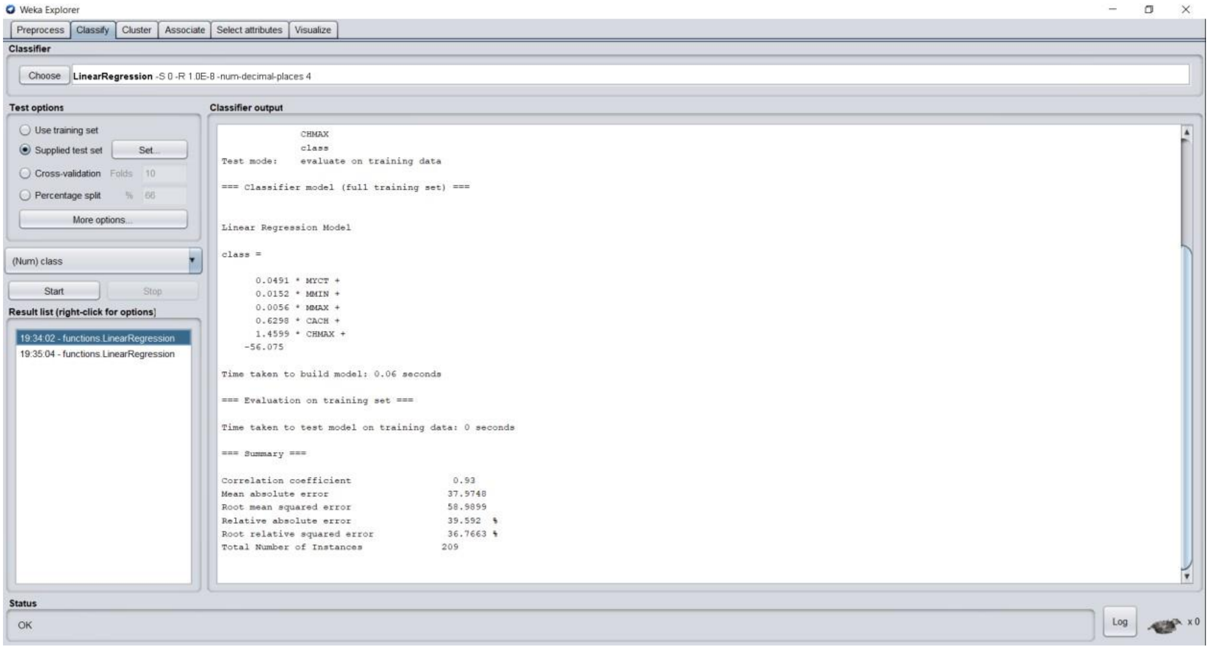
Classifier Output:

```
=== Run information===

Scheme:       weka.classifiers.bayes.NaiveBayes

Relation:     pima_diabetes
Instances:     768
Attributes: 9
            preg
            plas
            pres
            skin
            insu
            mass
            pedi
            age
            class
Test mode:      evaluate on training data

=== Classifier model (full training set) ===

Naive Bayes Classifier

                      Class
Attribute       tested_negative tested_positive
                  (0.65)        (0.35)
================================================
preg
```

```
  mean              3.4234     4.9795
  std. dev.         3.0166     3.6827
  weight sum           500        268
  precision         1.0625     1.0625

plas
  mean            109.9541   141.2581
  std. dev.        26.1114    31.8728
  weight sum           500        268
  precision         1.4741     1.4741

pres
  mean             68.1397     70.718
  std. dev.        17.9834    21.4094
  weight sum           500        268
  precision         2.6522     2.6522

skin
  mean             19.8356    22.2824
  std. dev.        14.8974    17.6992
  weight sum           500        268
  precision           1.98       1.98

insu
  mean             68.8507   100.2812
  std. dev.         98.828   138.4883
  weight sum           500        268
  precision          4.573      4.573
```

```
mass
  mean              30.3009     35.1475
  std. dev.          7.6833     7.2537
  weight sum            500        268
  precision          0.2717     0.2717
pedi
  mean               0.4297     0.5504
  std. dev.          0.2986     0.3715
  weight sum            500        268
  precision          0.0045     0.0045

age
  mean              31.2494     37.0808
  std. dev.         11.6059     10.9146
  weight sum            500        268
  precision          1.1765     1.1765



Time taken to build model: 0.01 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0.02 seconds


=== Summary ===
Correctly Classified Instances        586              76.3021 %
Incorrectly Classified Instances       182              23.6979 %
Kappa statistic                   0.4674
Mean absolute error                    0.2811
Root mean squared error                    0.4133
Relative absolute error              61.8486 %
Root relative squared error               86.7082 %
Total Number of Instances              768

=== Detailed Accuracy By Class ===

              TP Rate FP Rate Precision Recall   F-Measure  MCC
                                                                  RO
C Area PRC Area Class
              0.842     0.384     0.803     0.842     0.822     0.469     0.825    0.902
tested_negative
              0.616    0.158    0.676      0.616    0.645      0.469    0.825     0.684
tested_positive
Weighted Avg.      0.763     0.305    0.759       0.763    0.760       0.469    0.825
0.826

=== Confusion Matrix===

   a b <-- classified as
 421 79 | a = tested_negative
 103 165 |  b = tested_positive
```

ii) Linear Regression:



LinearRegression -S 0 -R 1.0E-8 -num-decimal-places 4

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

(Num) class

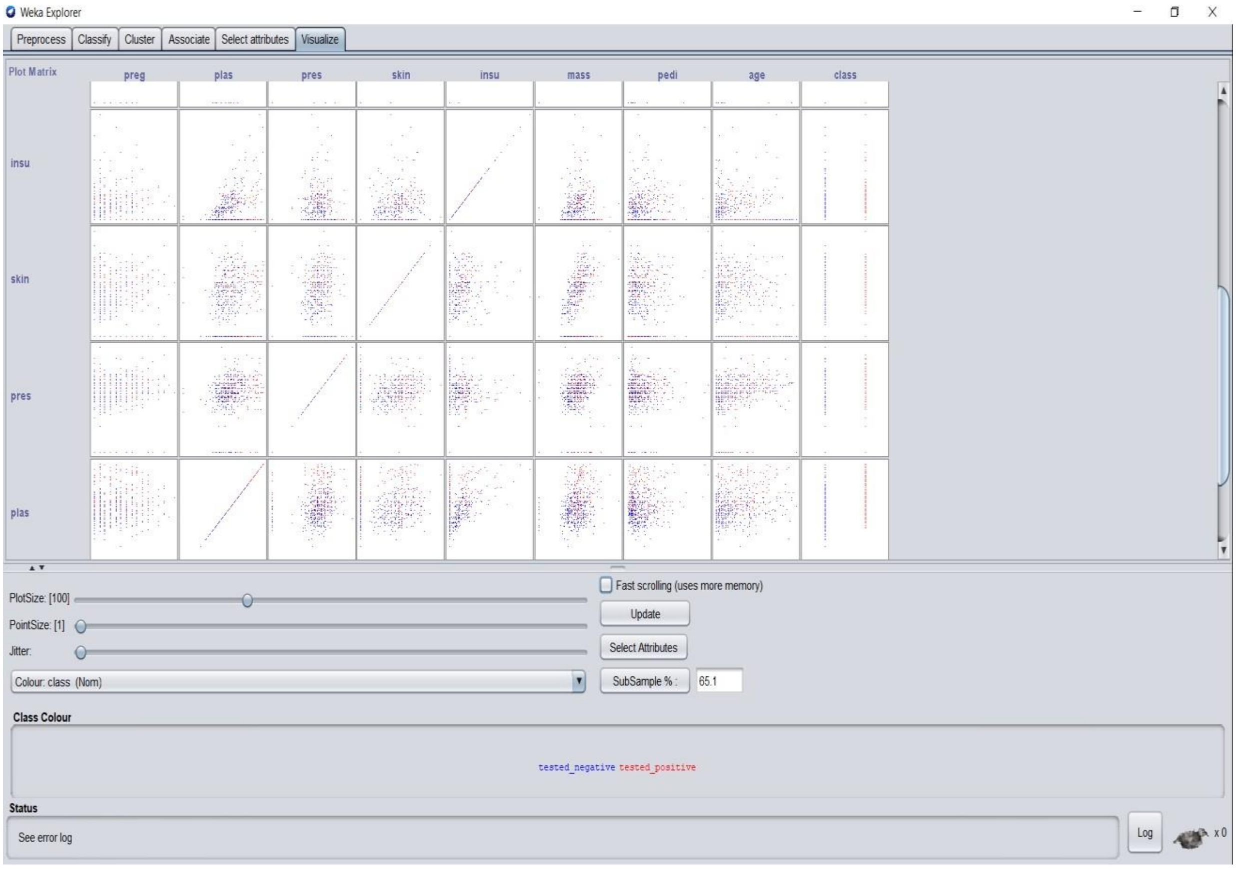
19:34:02 - functions.LinearRegression
19:35:04 - functions.LinearRegression

```
              CHMAX
              class
Test mode:    evaluate on training data

=== Classifier model (full training set) ===


Linear Regression Model

class =

      0.0491 * MYCT +
      0.0152 * MMIN +
      0.0056 * MMAX +
      0.6298 * CACH +
      1.4599 * CHMAX +
    -56.075

Time taken to build model: 0.06 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0 seconds

=== Summary ===

Correlation coefficient                  0.93
Mean absolute error                     37.9748
Root mean squared error                 58.9899
Relative absolute error                 39.592  %
Root relative squared error             36.7663 %
Total Number of Instances              209
```

Classifier Output:

```
=== Run information===


Scheme:        weka.classifiers.functions.LinearRegression -S 0 -R 1.0E-8 -
num-decimal-places 4
Relation:     pima_diabetes
Instances:     768
Attributes: 9
              preg
              plas
              pres
              skin
              insu
              mass
              pedi
              age
              class
Test mode:     evaluate on training data


=== Classifier model (full training set) ===



Linear Regression Model


age =


      1.6654 * preg +
      0.0573 * plas +
      0.1034*pres+
    -0.0911 * skin+
     1.4283 * class=tested_positive+
    14.1346
Time taken to build model: 0.02 seconds
=== Evaluation on training set ===
Time taken to test model on training data: 0.01 seconds
=== Summary ===
Correlation coefficient              0.6078
Mean absolute error                  6.7393
Root mean squared error                 9.3323
Relative absolute error             70.3002 %
Root relative squared error            79.4063%
Total Number of Instances              768
```

Visualization:



PlotSize: [100]

PointSize: [1]

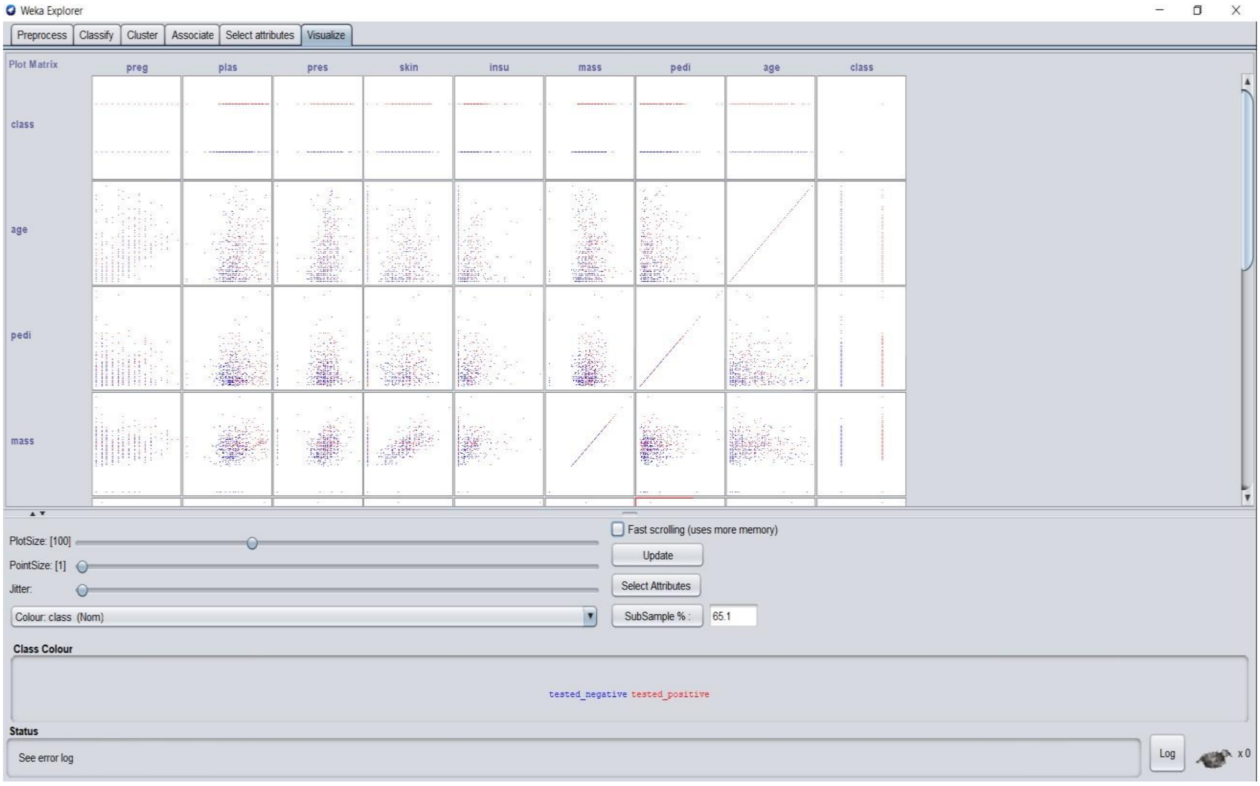
Jitter:

Colour: class (Nom)

SubSample % : 65.1

Class Colour: tested_negative tested_positive

Status: See error log



PlotSize: [100]

PointSize: [1]

Jitter:

Colour: class (Nom)

☐ Fast scrolling (uses more memory)

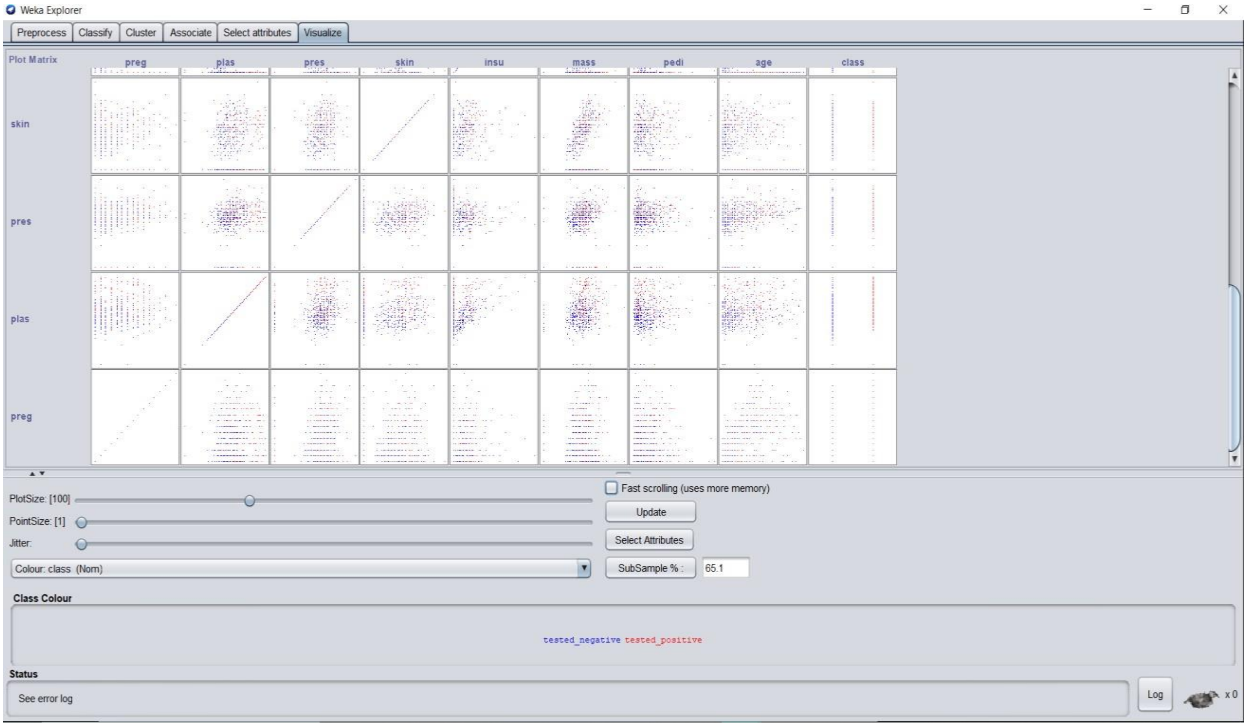
SubSample % : 65.1

Class Colour

tested_negative tested_positive

Status

See error log



PlotSize: [100]

PointSize: [1]

Jitter:

Colour: class (Nom)

☐ Fast scrolling (uses more memory)

SubSample % : 65.1

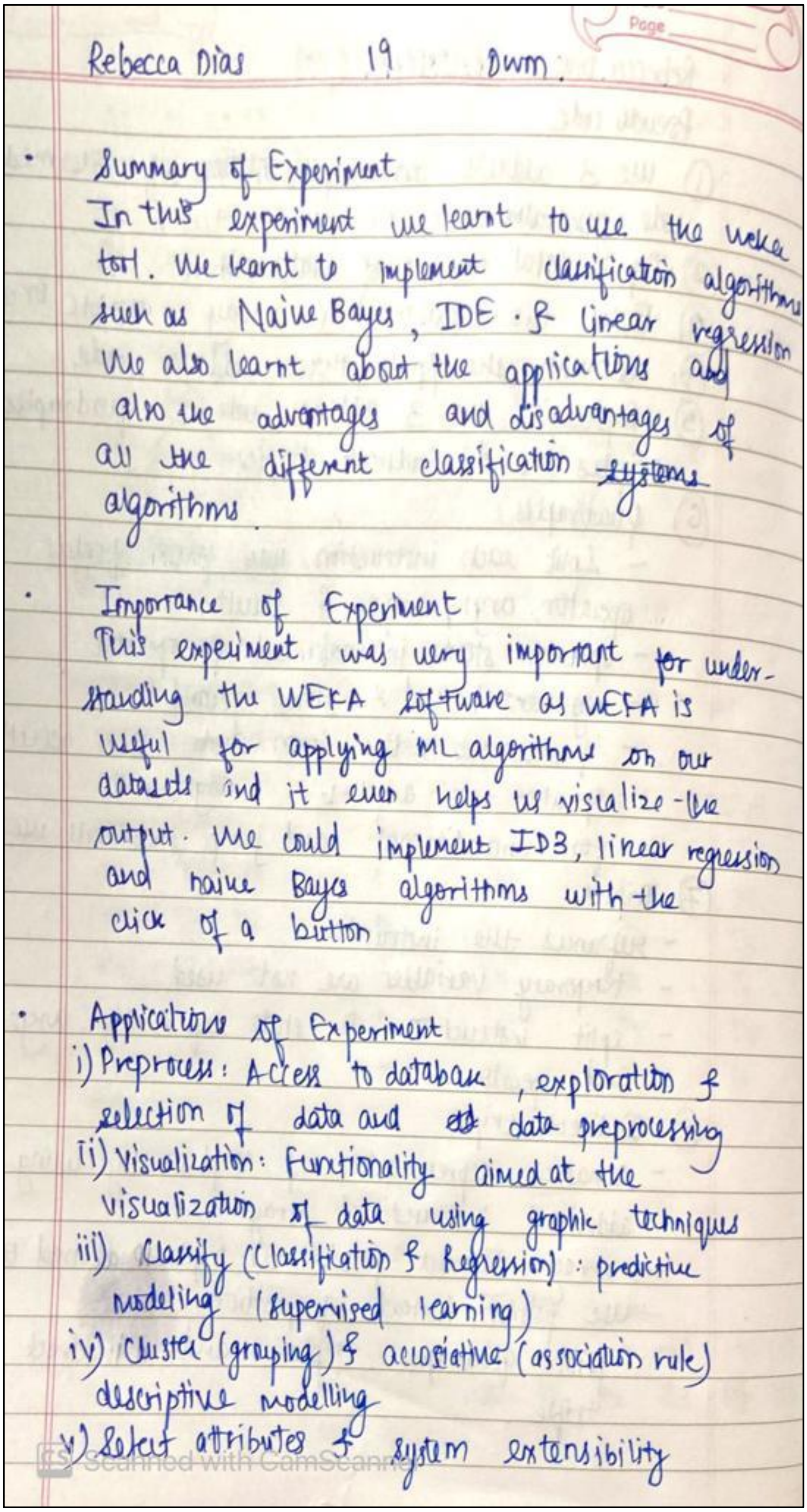
Class Colour

tested_negative tested_positive

Status

See error log

Conclusion: -



Rebecca Dias 19 Dwm.

- Summary of Experiment

In this experiment we learnt to use the weka tool. We learnt to implement classification algorithms such as Naive Bayes, IDE · & linear regression We also learnt about the applications and also the advantages and disadvantages of all the different classification ~~systems~~ algorithms.

- Importance of Experiment

This experiment was very important for understanding the WEKA software as WEFA is useful for applying ML algorithms on our datasets and it even helps us visualize the output. We could implement ID3, linear regression and naive Bayes algorithms with the click of a button

- Applications of Experiment

i) Preprocess: Access to database, exploration & selection of data and data preprocessing

ii) Visualization: Functionality aimed at the visualization of data using graphic techniques

iii) Classify (Classification & Regression): predictive modeling (supervised learning)

iv) Cluster (grouping) & accosiative (association rule) descriptive modelling

v) Select attributes & system extensibility

Scanned with CamScanner

EXPERIMENTNO.8B:ClusteringAlgorithmusingWEKA

Aim: - To Implement all the Clustering algorithm using WEKA (K Means, Agglomerative).

S/W Requirement: - WEKA Tool

Theory: -

1. Application of all different Clustering based on types (Partition and Hierarchal):

Partition:

- Academic performance
- Diagnostic systems
- Search engines
- Wireless sensor networks

Hierarchal:

1) US Senator Clustering through Twitter Can we find the party linesthrough Twitter?

2) Charting Evolution through Phylogenetic Trees How can we relate different species together?

3) Tracking Viruses through Phylogenetic Trees Can we find where a viral outbreak originated?

2. Advantage and Disadvantage of each Clustering Algorithms (Partition and Hierarchal):

Partition:

Advantages:

• Easy to implement

• With a large number of variables, K-Means may be computationally faster than hierarchical clustering (if K is small).

• k-Means may produce Higher clusters than hierarchical clustering

- An instance can change cluster (move to another cluster) when the centroids are recomputed.

Disadvantages:

- Difficult to predict the number of clusters (K-Value)
- Initial seeds have a strong impact on the final results
- The order of the data has an impact on the final results
- Sensitive to scale: rescaling your datasets (normalization or standardization) will completely change results. While this itself is not bad, not realizing that you have to spend extra a4en (on to scaling your data might bebad.

Hierarchal:

Advantages

1) No apriori information about the number of clusters required.

2) Easy to implement and gives best result in some cases.

Disadvantages

1) Algorithm can never undo what was done previously.

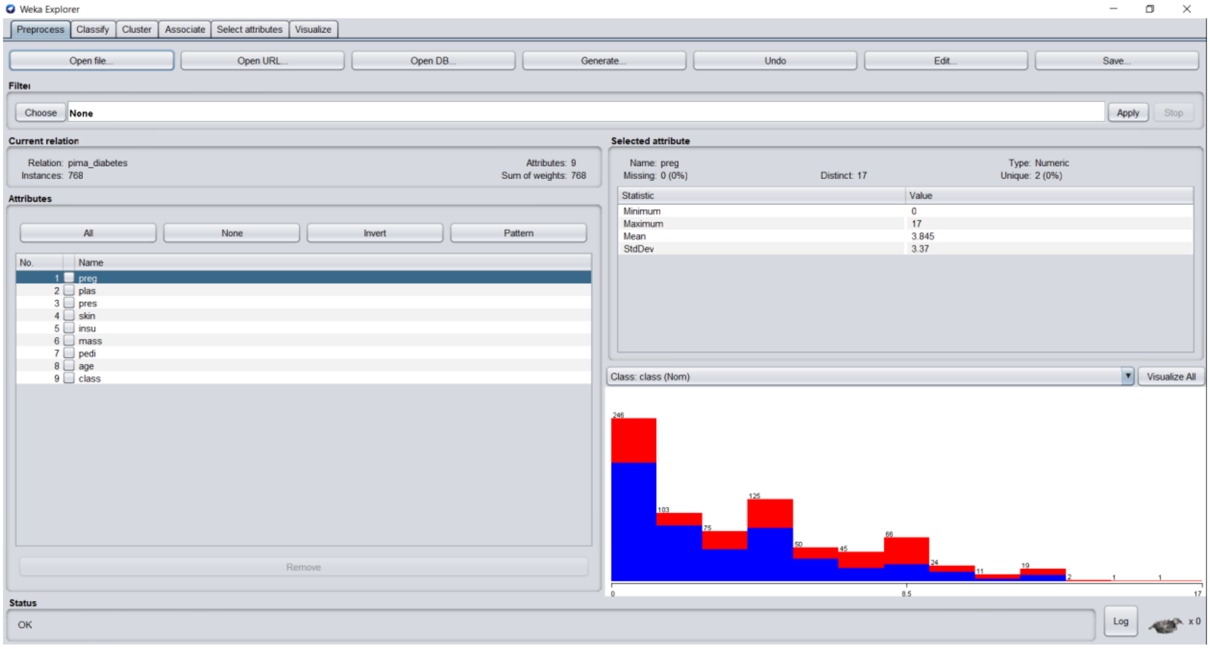
2) Time complexity of at least O (n^2 *log n*) is required, where *'n'* is the number of datapoints.

3) Based on the type of distance matrix chosen for merging different algorithms can suffer with one or more of the following:

i) Sensitivity to noise and outliers

ii) Breaking large clusters

Implementation: -



Filter: None

Current relation

Relation: pima_diabetes
Instances: 768
Attributes: 9
Sum of weights: 768

Attributes

No.		Name
1	□	preg
2	□	plas
3	□	pres
4	□	skin
5	□	insu
6	□	mass
7	□	pedi
8	□	age
9	□	class

Selected attribute

Name: preg
Missing: 0 (0%)
Distinct: 17
Type: Numeric
Unique: 2 (0%)

Statistic	Value
Minimum	0
Maximum	17
Mean	3.845
StdDev	3.37

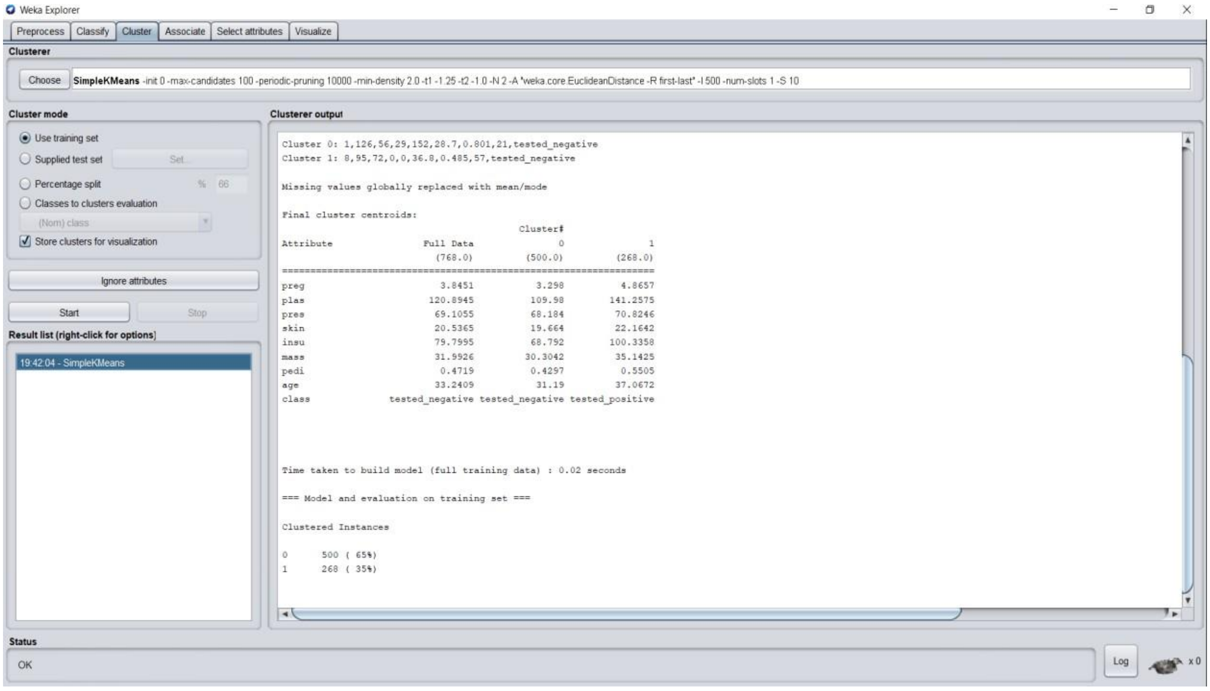
Class: class (Nom)

Status

OK

OUTPUT: -

i)K-Means:



Clusterer: SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10

Cluster mode

● Use training set
○ Supplied test set
○ Percentage split % 66
○ Classes to clusters evaluation
(Nom) class
☑ Store clusters for visualization

Result list (right-click for options)

19:42:04 - SimpleKMeans

Clusterer output

```
Cluster 0: 1,126,56,29,152,28.7,0.801,21,tested_negative
Cluster 1: 8,95,72,0,0,36.8,0.485,57,tested_negative

Missing values globally replaced with mean/mode

Final cluster centroids:
                                         Cluster#
Attribute                 Full Data              0              1
                            (768.0)        (500.0)        (268.0)
==================================================================
preg                         3.8451          3.298         4.8657
plas                       120.8945         109.98       141.2575
pres                        69.1055         68.184        70.8246
skin                        20.5365         19.664        22.1642
insu                        79.7995         68.792       100.3358
mass                        31.9926        30.3042        35.1425
pedi                         0.4719         0.4297         0.5505
age                         33.2409          31.19        37.0672
class                tested_negative tested_negative tested_positive




Time taken to build model (full training data) : 0.02 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       500 ( 65%)
1       268 ( 35%)
```

Status

OK

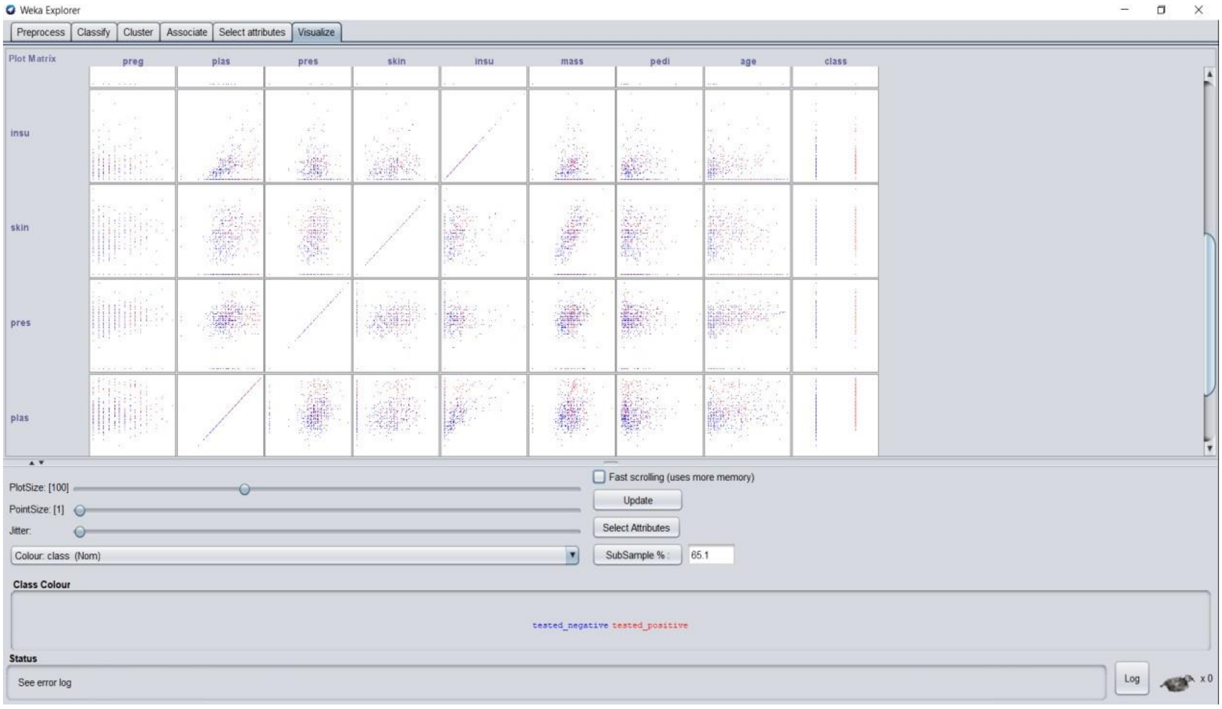
Classifier Output:

=== Run information===

```
Scheme:        weka.clusterers.SimpleKMeans -init 0 -max-candidates 100 -
periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A
"weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10

Relation:     pima_diabetes

Instances:     768

Attributes: 9

              preg

              plas

              pres

              skin

              insu

              mass

              pedi

              age

              class

Test mode:     evaluate on trainingdata



=== Clustering model (full training set) ===



kMeans

======


Number of iterations:4

Within cluster sum of squared errors: 149.5177664581119


Initial starting points(random):
```

```
Cluster 0:1,126,56,29,152,28.7,0.801,21,tested_negative
Cluster 1: 8,95,72,0,0,36.8,0.485,57,tested_negative

Missing values globally replaced with mean/mode


Final cluster centroids:
                              Cluster#
Attribute               Full Data           0             1
                    (768.0)        (500.0)        (268.0)
===========================================================
======

preg                      3.8451         3.298        4.8657
plas                    120.8945         109.98        141.2575
pres                     69.1055        68.184         70.8246
skin                    20.5365         19.664         22.1642
insu                    79.7995         68.792          100.3358
mass                      31.9926        30.3042          35.1425
pedi                      0.4719         0.4297          0.5505
age                     33.2409          31.19        37.0672
class            tested_negative tested_negative tested_positive


  Time taken to build model (full training data) : 0.02 seconds


=== Model and evaluation on training set ===


Clustered Instances

0      500(65%)
1     268 (35%)
```

Visualization:



PlotSize: [100]

PointSize: [1]

Jitter:

Colour: class (Nom)

☐ Fast scrolling (uses more memory)

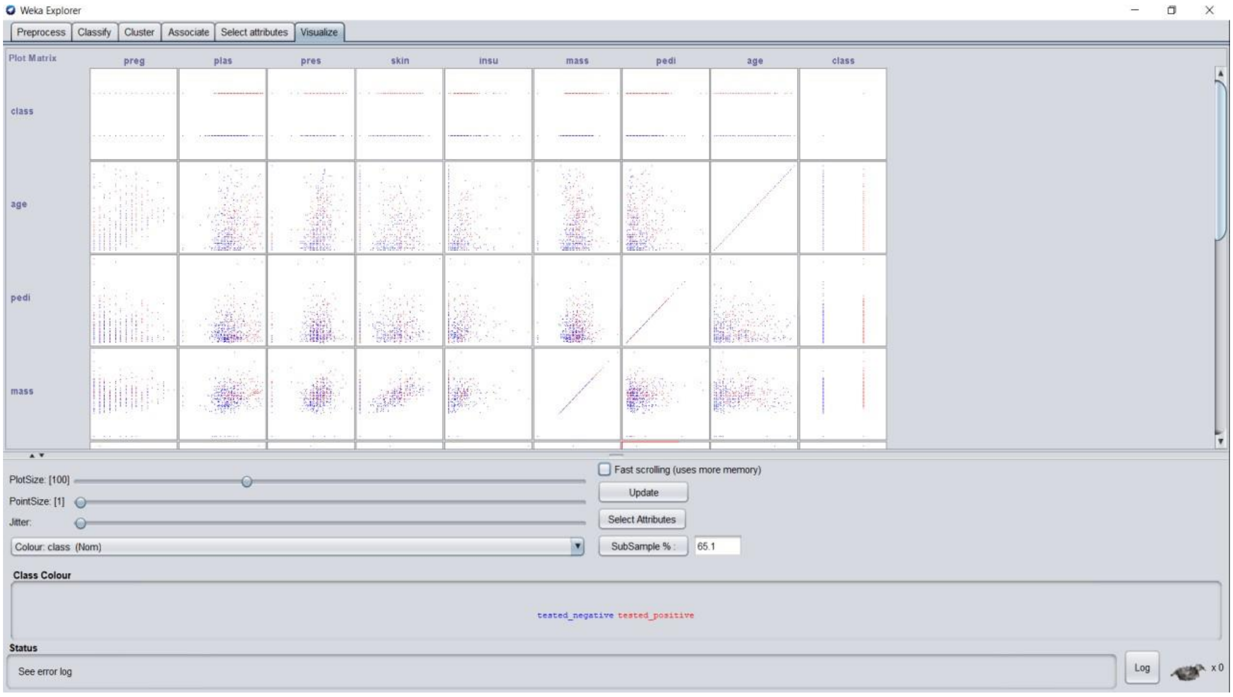
SubSample % : 65.1

Class Colour

tested_negative tested_positive

Status

See error log



PlotSize: [100]

PointSize: [1]

Jitter:

Colour: class (Nom)

☐ Fast scrolling (uses more memory)

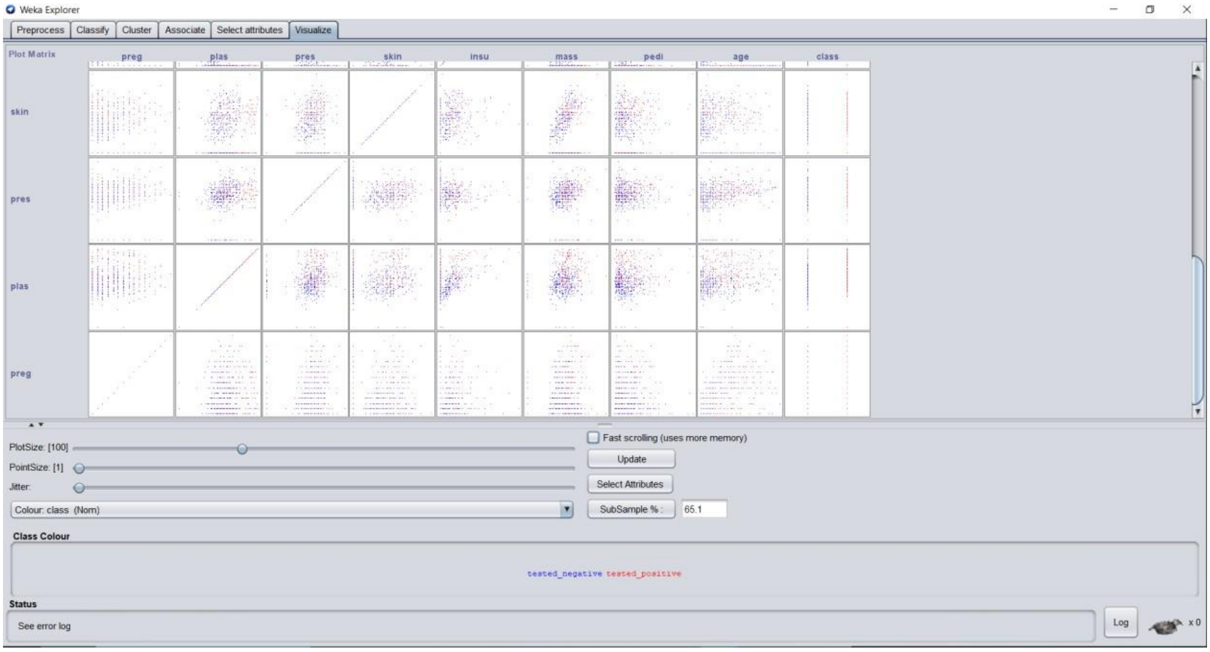
SubSample % : 65.1

Class Colour

tested_negative tested_positive

Status

See error log



PlotSize: [100]

PointSize: [1]

Jitter:

Colour: class (Nom)

□ Fast scrolling (uses more memory)

SubSample % : 65.1

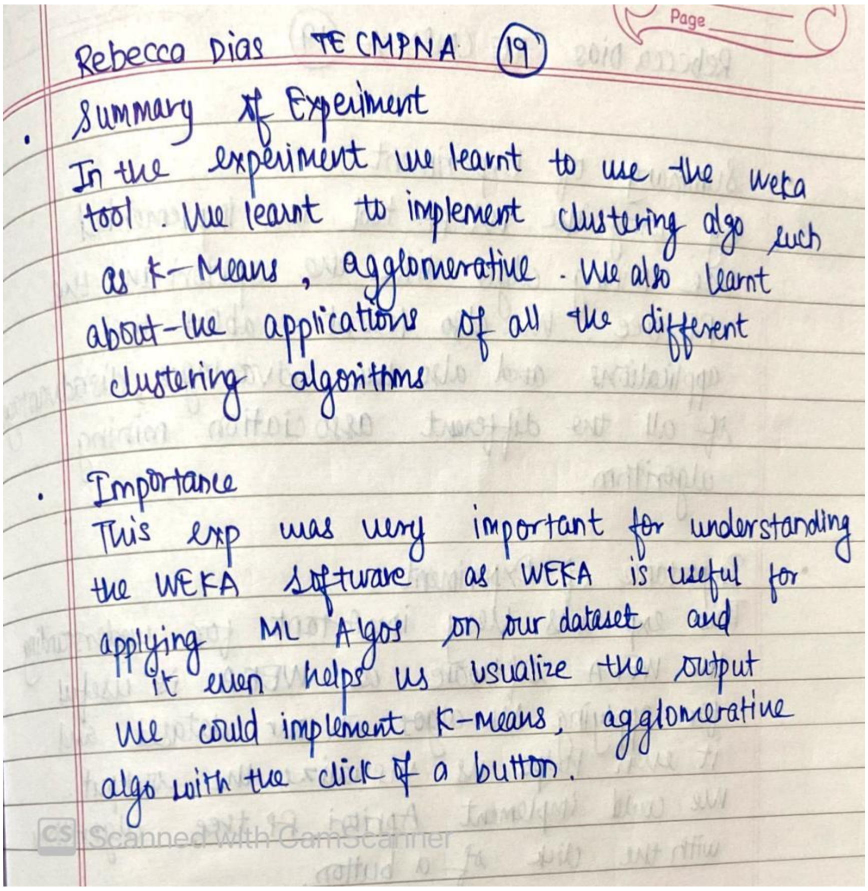
Class Colour

tested_negative tested_positive

Status

See error log

Conclusion: -



Rebecca Dias TE CMPNA 19

Page

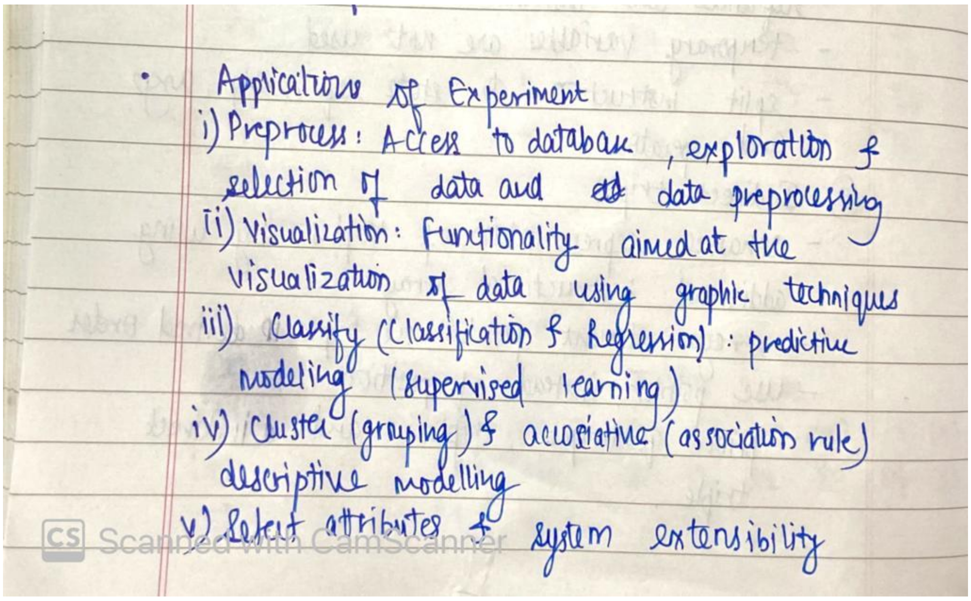
- Summary of Experiment

In the experiment we learnt to use the weka tool. We learnt to implement clustering algo such as K-Means, agglomerative. We also learnt about the applications of all the different clustering algorithms

- Importance

This exp was very important for understanding the WEKA software as WEKA is useful for applying ML Algos on our dataset and it even helps us visualize the output we could implement K-Means, agglomerative algo with the click of a button.

CS Scanned with CamScanner



- Applications of Experiment

i) Preprocess: Access to database, exploration & selection of data and ~~dd~~ data preprocessing

ii) Visualization: Functionality aimed at the visualization of data using graphic techniques

iii) Classify (Classification & Regression): predictive modeling (supervised learning)

iv) Cluster (grouping) & accosiative (association rule) descriptive modelling

v) Select attributes & system extensibility

CS Scanned with CamScanner

EXPERIMENTNO.8C:AssociationAlgorithmusingWEKA

Aim: - Association Mining like Apriori, FP tree in WEKA.

S/W Requirement: - WEKA Tool

Theory: -

1. Application of Association Algorithms:

Association algorithm has various applications in machine learning and data mining. Below are some popular applications of association rule learning:

- Market Basket Analysis: It is one of the popular examples and applications of association rule mining. This technique is commonly used by big retailers to determine the association between items.
- Medical Diagnosis: With the help of association rules, patients can be cured easily, as it helps in identifying the probability of illness for a particular disease.
- Protein Sequence: The association rules help in determining the synthesis of artificial Proteins.
- It is also used for the Cata-log Design and Loss-leader Analysis and many more otherapplications.

2. Advantage and Disadvantage of Association Algorithms:

• **Advantages:**

- Uses large item set property

- Easily parallelized

- Easy to implement

• **Disadvantages:**

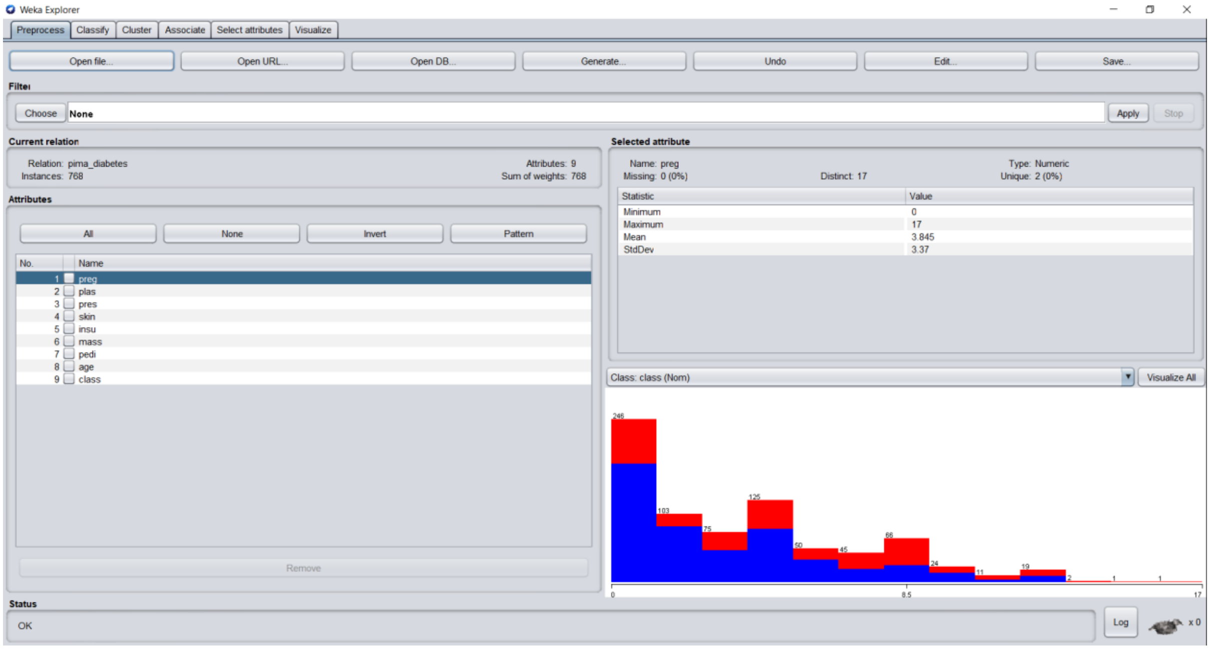
- Assumes transaction database is memory resident

- Requires many databasescans

3. How Association helps in Boosting the Business Profit?

In data mining, **association** rules are useful for analysing and predicting customer behaviour. They play an important part in customer analytics, market basket analysis, product clustering, cata-log design and store layout. **Association Algorithm** usually contains or deals with a large number of transactions. For example, customers buying a lot of goods from a grocery store, by applying this method of the **algorithm** the grocery stores can enhance their sales performance and could work effectively.

Implementation: -



Filter: None

Current relation

Relation: pima_diabetes
Instances: 768
Attributes: 9
Sum of weights: 768

Attributes

No.		Name
1	□	preg
2	□	plas
3	□	pres
4	□	skin
5	□	insu
6	□	mass
7	□	pedi
8	□	age
9	□	class

Selected attribute

Name: preg
Missing: 0 (0%)
Distinct: 17
Type: Numeric
Unique: 2 (0%)

Statistic	Value
Minimum	0
Maximum	17
Mean	3.845
StdDev	3.37

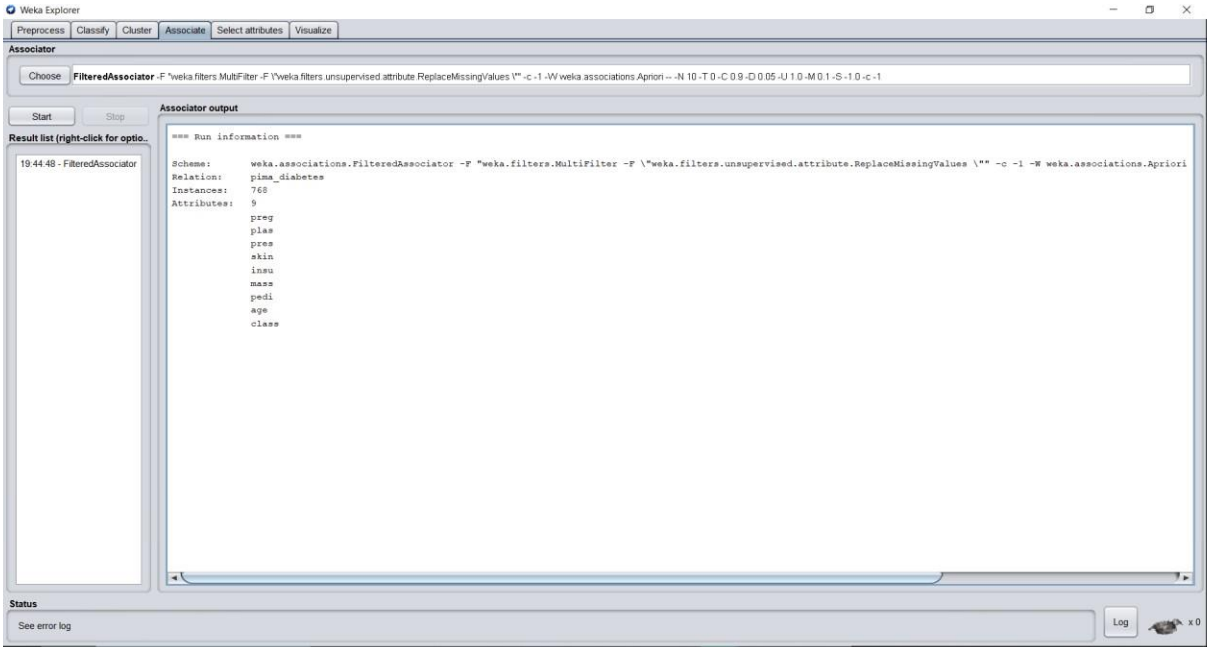
Class: class (Nom)

Status

OK

OUTPUT: -

i)Apriori:



FilteredAssociator -F "weka.filters.MultiFilter -F \"weka.filters.unsupervised.attribute.ReplaceMissingValues \"" -c -1 -W weka.associations.Apriori -- -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1

19:44:48 - FilteredAssociator

```
=== Run information ===

Scheme:       weka.associations.FilteredAssociator -F "weka.filters.MultiFilter -F \"weka.filters.unsupervised.attribute.ReplaceMissingValues \"" -c -1 -W weka.associations.Apriori
Relation:     pima_diabetes
Instances:    768
Attributes:   9
              preg
              plas
              pres
              skin
              insu
              mass
              pedi
              age
              class
```

See error log

Classifier Output:

=== Run information===

Scheme: weka.associations.FilteredAssociator -F "weka.filters.MultiFilter -F \"weka.filters.unsupervised.attribute.ReplaceMissingValues \"" -c -1 -W weka.associations.Apriori -- -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1

Relation: pima_diabetes

Instances: 768

Attributes: 9

preg

plas

pres

skin

insu

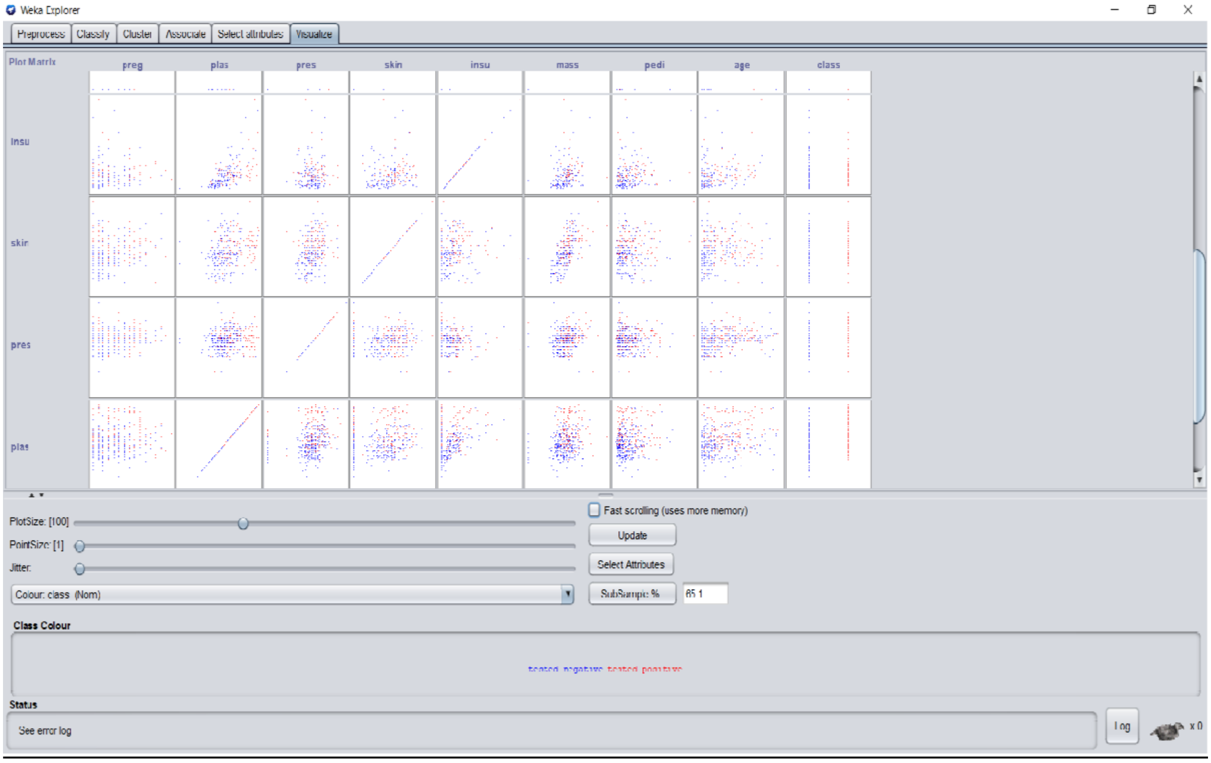
mass

pedi

age

class

Visualization:



☐ Fast scrolling (uses more memory)

PlotSize: [100]

PointSize: [1]

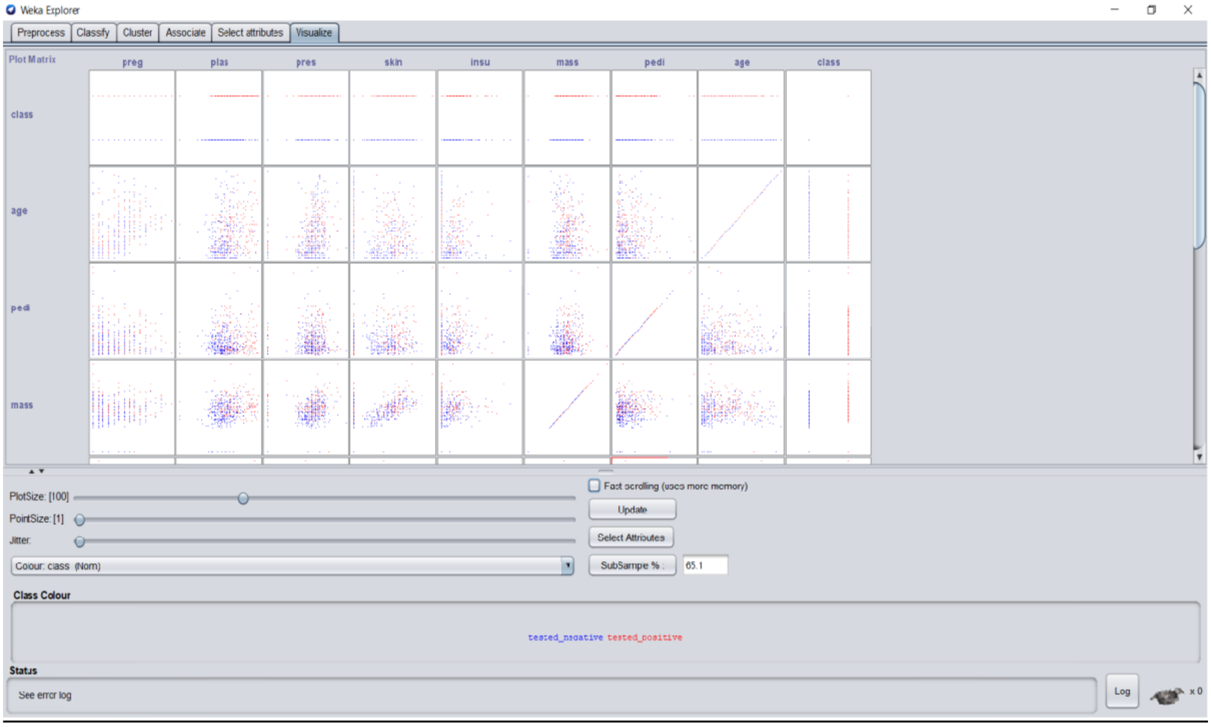
Jitter:

Colour: class (Nom)

SubSample % 65.1

tested_negative tested_positive

See error log



PlotSize: [100]

PointSize: [1]

Jitter:

Colour: class (Nom)

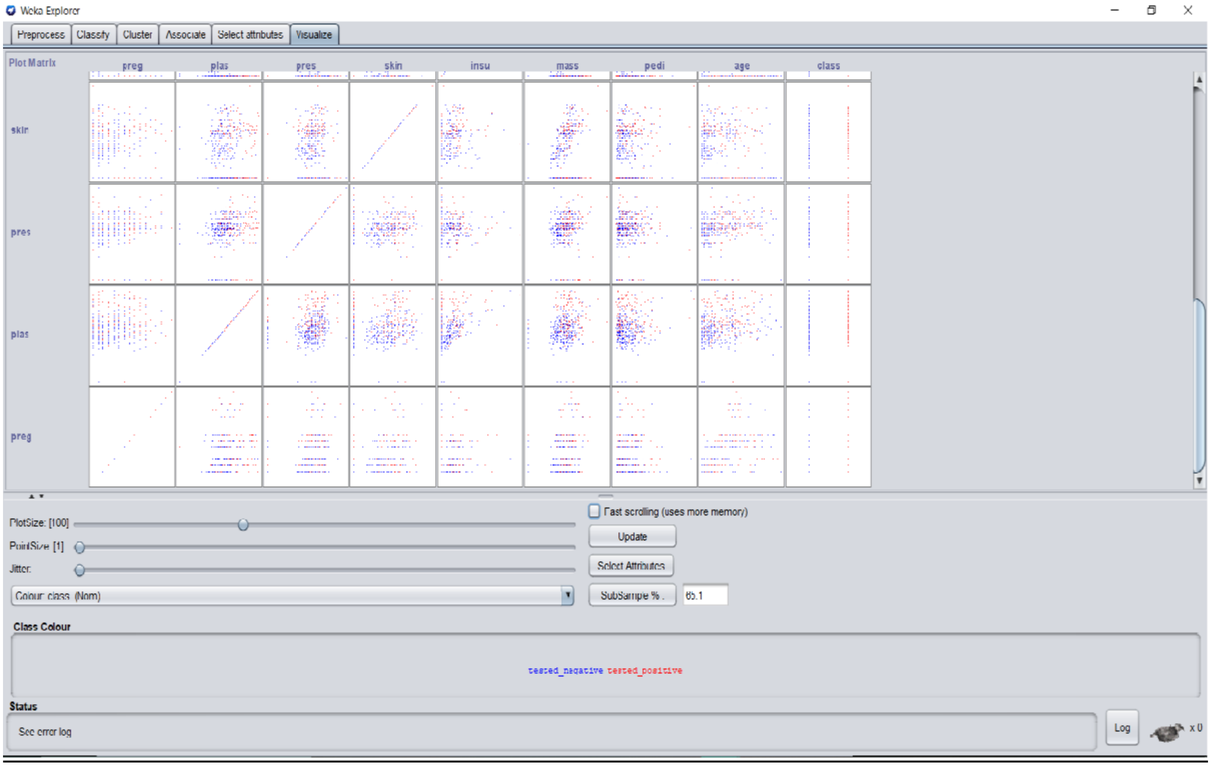
SubSample % : 65.1

Class Colour

tested_negative tested_positive

Status

See error log



PlotSize: [100]

PointSize: [1]

Jitter:

Colour: class (Nom)

SubSample % : 65.1

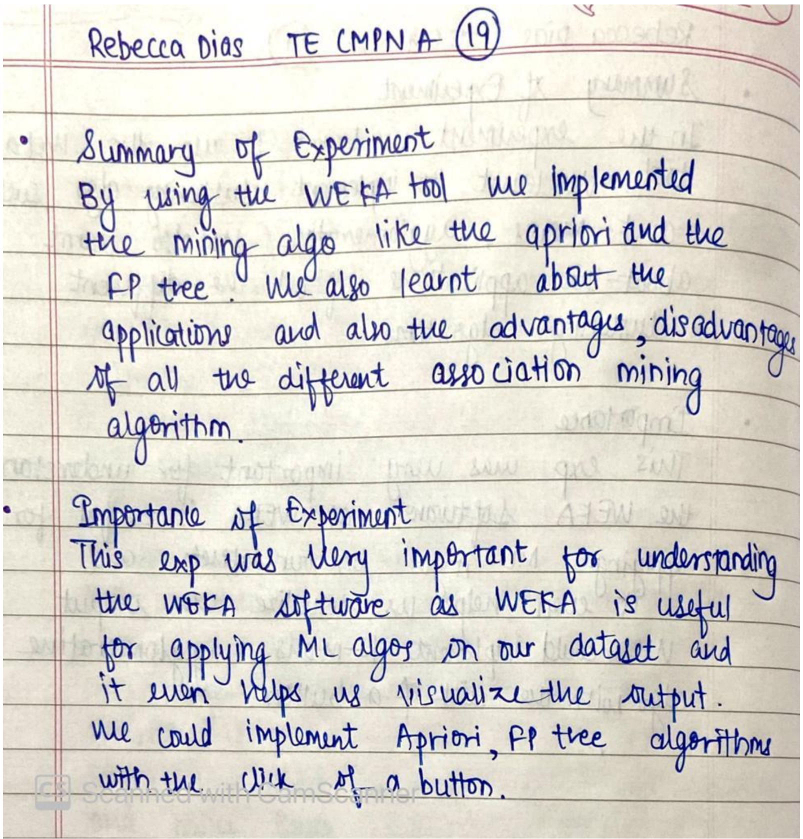
Class Colour

tested_negative tested_positive

Status

See error log

Conclusion: -



Rebecca Dias TE CMPN A (19)

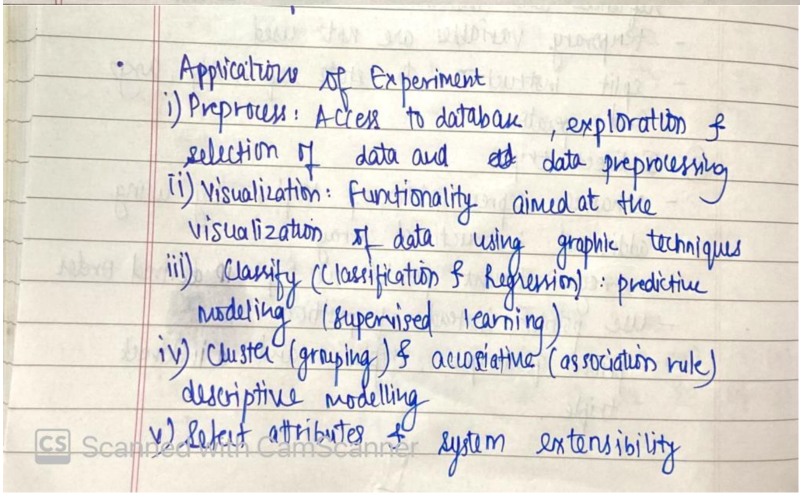
- Summary of Experiment

By using the WEKA tool we implemented the mining algo like the apriori and the FP tree. We also learnt about the applications and also the advantages, disadvantages of all the different association mining algorithm.

- Importance of Experiment

This exp was very important for understanding the WEKA software as WEKA is useful for applying ML algos on our dataset and it even helps us visualize the output. We could implement Apriori, FP tree algorithms with the click of a button.

CS Scanned with CamScanner



- Applications of Experiment

i) Preprocess: Access to database, exploration & selection of data and ~~ed~~ data preprocessing

ii) Visualization: Functionality aimed at the visualization of data using graphic techniques

iii) Classify (Classification & Regression): predictive modeling (supervised learning)

iv) Cluster (grouping) & accosiative (association rule) descriptive modelling

v) Select attributes & system extensibility

CS Scanned with CamScanner