**CLASSIFICATION**

**Dataset(Numerical):**

|  |  |
| --- | --- |
| **Burst Header Packet (BHP) flooding attack on Optical Burst Switching (OBS) Network Data Set** |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Data Set Characteristics:** | Text | **Number of Instances:** | 1075 | **Area:** | Computer |
| **Attribute Characteristics:** | Integer | **Number of Attributes:** | 22 | **Date Donated** | 2017-08-28 |
| **Associated Tasks:** | Classification | **Missing Values?** | N/A | **Number of Web Hits:** | 44225 |

**Source:**

Adel Rajab

Department of Computer Science and Engineering,

University of South Carolina,

Columbia, SC, USA, 29208

Phone number:(803)238-6657

rajaba '@' email.sc.edu

**Data Set Information:**

For Further information about the variables see the file in the data folder.

**Attribute Information:**

1.Node: This is the number of the sending node (numeric).

2. Utilized Bandwidth Rate: This is the normalization of Used Band width(numeric).

3. Packet Drop Rate: This is the normalization of Percentage\_Of\_Lost\_Pcaket\_Rate (numeric).

4. Reserved\_Bandwidth: Initial reserved Bandwidth assigned (given) to each node, the user (usr) in the experiments assign these values. (numeric).

5. Average\_Delay\_Time\_Per\_Sec: Average Delay Time (per second) for each node. This is (End-to End Delay). (numeric).

6. Percentage\_Of\_Lost\_Pcaket\_Rate: Percentage of Packets Drop Rate for each node (numeric).

7. Percentage\_Of\_Lost\_Byte\_Rate: Percentage of Lost Byte Rate for each node (numeric).

8. Packet Received Rate: Total received packets (per second) for each node based on Reserved\_Bandwidthâ (numeric).

9. Used\_Bandwidth: This is what each node could reserve from the Reserved\_Bandwidth(numeric).

10. Lost\_Bandwidth: The amount of lost Bandwidth by each node from Reserved\_Bandwidth(numeric).

11. Packet Size\_Byte: Packets size in Byte assigned specifically for each node to transmit. Note: 60 Byte will be added to the 1440 for the IP Header and the UDP Header ((Data size 1440 Byte) + (IP Header 40 Byte) + (UDP Header 20 Byte)) =1500 Byte (numeric).

12. Packet\_Transmitted: Total transmitted packets (per second) for each node based on the Reserved\_Bandwidth (numeric).

13. Packet\_Received: Total received packets (per second) for each node based on the Reserved\_Bandwidth (numeric).

14. Packet\_lost: Total lost packets (per second) for each node, which based on the Lost\_Bandwidth (numeric).

15. Transmitted\_Byte: Total transmitted Byte (per second) for each node (numeric).

16. Received\_Byte: Total received Byte (per second) for each node based on the Reserved\_Bandwidth (numeric).

17. 10-Run-AVG-Drop-Rate: Average packet drop rate for 10 consecutive (run) iterations (numeric).

18. 10-Run-AVG-Bandwidth-Use: Average Bandwidth utilized for 10 consecutive (run) iterations (numeric).

19. 10-Run-Delay: Average delay time for 10 consecutive (run) iterations (numeric).

20. Node Status' {B, NB, P NB}: initial classification of nodes based on Packet Drop Rate, Used\_Bandwidth and Average\_Delay\_Time\_Per\_Sec. B = Behaving, NB = Not Behaving and P NB = Potentially Not Behaving. (Categorical)

21. Flood Status: Percentage of flood per node based on Packet Drop Rate Medium and high level of BHP flood attack in case B (numeric).

22. Class ' {NB-No Block, Block, No Block, NB-Wait}: The final classification of nodes based on Packet Drop Rate, Reserved\_Bandwidth, Iteration, Used\_Bandwidth, Packet Drop Rate. This is for case B (Categorical ).

**Relevant Papers:**

Rajab, C. T. Huang, M. Alshargabi, and J. Cobb, Countering Burst Header Packet Flooding Attack in Optical Burst Switching Network, In: International Conference on Information Security Practice and Experience, Springer International Publishing, pp. 315329, Nov 16 2016.

**Dataset (Nominal):**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Mushroom Data Set**  **Abstract**: From Audobon Society Field Guide; mushrooms described in terms of physical characteristics; classification: poisonous or edible | https://archive.ics.uci.edu/ml/assets/MLimages/Large73.jpg |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **Data Set Characteristics:** | Multivariate | **Number of Instances:** | 8124 | **Area:** | Life | | **Attribute Characteristics:** | Categorical | **Number of Attributes:** | 22 | **Date Donated** | 1987-04-27 | | **Associated Tasks:** | Classification | **Missing Values?** | Yes | **Number of Web Hits:** | 792234 |   **Source:**  **Origin**: Mushroom records drawn from The Audubon Society Field Guide to North American Mushrooms (1981). G. H. Lincoff (Pres.), New York: Alfred A. Knopf  Donor:Jeff Schlimmer (Jeffrey.Schlimmer '@' a.gp.cs.cmu.edu)  **Data Set Information:**  This data set includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family (pp. 500-525). Each species is identified as definitely edible, definitely poisonous, or of unknown edibility and not recommended. This latter class was combined with the poisonous one.  **Attribute Information:**  1. cap-shape: bell=b,conical=c,convex=x,flat=f, knobbed=k,sunken=s  2. cap-surface: fibrous=f,grooves=g,scaly=y,smooth=s  3.capcolor:brown=n,buff=b,cinnamon=c,gray=g,green=r,pink=p,purple=u,red=e,white=w,yellow=y  4. bruises?: bruises=t,no=f  5. odor: almond=a,anise=l,creosote=c,fishy=y,foul=f, musty=m,none=n,pungent=p,spicy=s  6. gill-attachment: attached=a,descending=d,free=f,notched=n  7. gill-spacing: close=c,crowded=w,distant=d  8. gill-size: broad=b,narrow=n  9.gillcolor:black=k,brown=n,buff=b,chocolate=h,gray=g,green=r,orange=o,pink=p,purple=u,red=e, white=w,yellow=y  10. stalk-shape: enlarging=e,tapering=t  11. stalk-root: bulbous=b,club=c,cup=u,equal=e, rhizomorphs=z,rooted=r,missing=?  12. stalk-surface-above-ring: fibrous=f,scaly=y,silky=k,smooth=s  13. stalk-surface-below-ring: fibrous=f,scaly=y,silky=k,smooth=s  14.stalkcolorabovering:brown=n,buff=b,cinnamon=c,gray=g,orange=o,pink=p,red=e,white=w,yellow=y  15.stalkcolorbelowring:brown=n,buff=b,cinnamon=c,gray=g,orange=o,pink=p,red=e,white=w,yellow=y  16. veil-type: partial=p,universal=u  17. veil-color: brown=n,orange=o,white=w,yellow=y  18. ring-number: none=n,one=o,two=t  19.ringtype:cobwebby=c,evanescent=e,flaring=f,large=l,none=n,pendant=p,sheathing=s,zone=z  20.sporeprintcolor:black=k,brown=n,buff=b,chocolate=h,green=r,orange=o,purple=u,white=w,yellow=y  21. population: abundant=a,clustered=c,numerous=n, scattered=s,several=v,solitary=y  22. habitat: grasses=g,leaves=l,meadows=m,paths=p, urban=u,waste=w,woods=d  **Relevant Papers:**  Schlimmer,J.S. (1987). Concept Acquisition Through Representational Adjustment (Technical Report 87-19). Doctoral disseration, Department of Information and Computer Science, University of California,Irvine.  Iba,W., Wogulis,J., & Langley,P. (1988). Trading off Simplicity and Coverage in Incremental Concept Learning. In Proceedings of the 5th International Conference on Machine Learning, 73-79. Ann Arbor, Michigan.  Duch W, Adamczak R, Grabczewski K (1996) Extraction of logical rules from training data using backpropagation networks, in: Proc. of the The 1st Online Workshop on Soft Computing, 19-30.Aug.1996, pp.  Duch W, Adamczak R, Grabczewski K, Ishikawa M, Ueda H, Extraction of crisp logical rules using constrained backpropagation networks - comparison of two new approaches, in: Proc. of the European Symposium on Artificial Neural Networks (ESANN'97), Bruge, Belgium 16-18.4.1997. |
|  |

**Description of the Problem:**

Decision Trees are an important type of algorithm for predictive modeling machine learning.

Decision Trees are a non-parametric supervised learning method used for classification and

regression. The goal is to create a model that predicts the value of a target variable by

learning simple decision rules inferred from the data features.

**Advantages of Decision Trees:-**

1. Decision trees are simple to understand, interpret, visualize.
2. Decision trees implicitly perform variable screening or feature selection.
3. Decision trees require little data preparation.
4. Decision trees can handle both numerical and categorical data.
5. Decision trees can also handle single-class and multi-class classification problems.
6. Nonlinear relationships between parameters do not affect tree performance.
7. Uses a white box model. If a given situation is observable in a model, the explanation
8. for the condition is easily explained by Boolean logic.
9. The cost of using the tree (i.e., predicting data) is logarithmic in the number of data
10. points used to train the tree.
11. Possible to validate a model using statistical tests. That makes it possible to account for
12. the reliability of the model.
13. Performs well even if its assumptions are somewhat violated by the true model from
14. which the data were generated.

**Disadvantages of Decision Trees:-**

* Decision-tree learners can create over-complex trees that do not generalize the data

well. This is called overfitting. Mechanisms such as pruning, setting the minimum

number of samples required at a leaf node or setting the maximum depth of the tree

are necessary to avoid this problem.

* Decision trees can be unstable because small variations in the data might result in a

completely different tree being generated. This is called variance, which needs to be

lowered by methods like bagging, boosting etc.

* Predictions of decision trees are neither smooth nor continuous, but are piecewise

constant approximations. Therefore, they are not good at extrapolation.

* The problem of learning an optimal decision tree is known to be NP-complete under

several aspects of optimality and even for simple concepts. Consequently, practical

decision-tree learning algorithms are based on heuristic algorithms such as the greedy

algorithm where locally optimal decisions are made at each node. Such algorithms

cannot guarantee to return the globally optimal decision tree. This can be mitigated by

training multiple trees in an ensemble learner, where the features and samples are

randomly sampled with replacement.

* There are concepts that are hard to learn because decision trees do not express them

easily, such as XOR, parity or multiplexer problems.

* Decision tree learners create biased trees if some classes dominate. It is therefore

recommended to balance the data set prior to fitting with the decision tree.

**Implementation in Weka:**

1) Start ->Programs ->Weka

2) Click on explorer.

3) Click on open file.

4) Select BHP flooding attack on OBS network dataset and click on open.

5) Click on edit button which shows BHP flooding attack on OBS network table on weka.

**Procedure for Constructing Decision Tree:**

**1)** Open Start Programs Weka

**2)** Open **explorer**.

**3)** Click on **open file** and select **weather.arff**

**4)** Select **Classifier option** on the top of the Menu bar.

**5)** Select **Choose button** and click on **Tree option** and thenClick on **J48.**

**6)** Click on **Start button** and output will be displayed on the **right side** of the window.

**7)** Select the **result list** and **right click** on result list and select **Visualize Tree option**.

**8)** Then **Decision Tree** will be displayed on **new window**.

**For Numeric Data:**

**Fig.1.1: Overview :-**

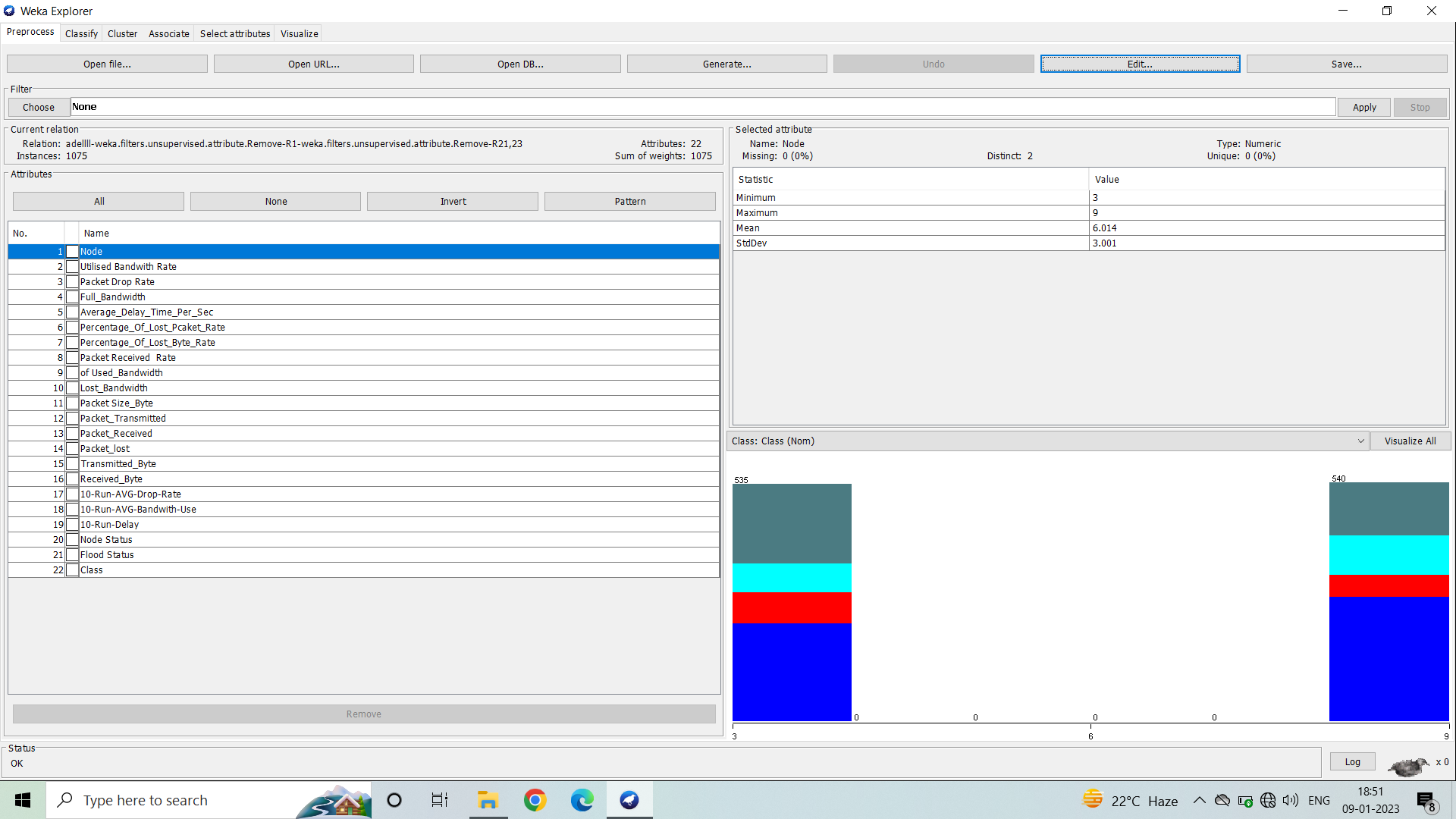


Fig.1.2: Dataset

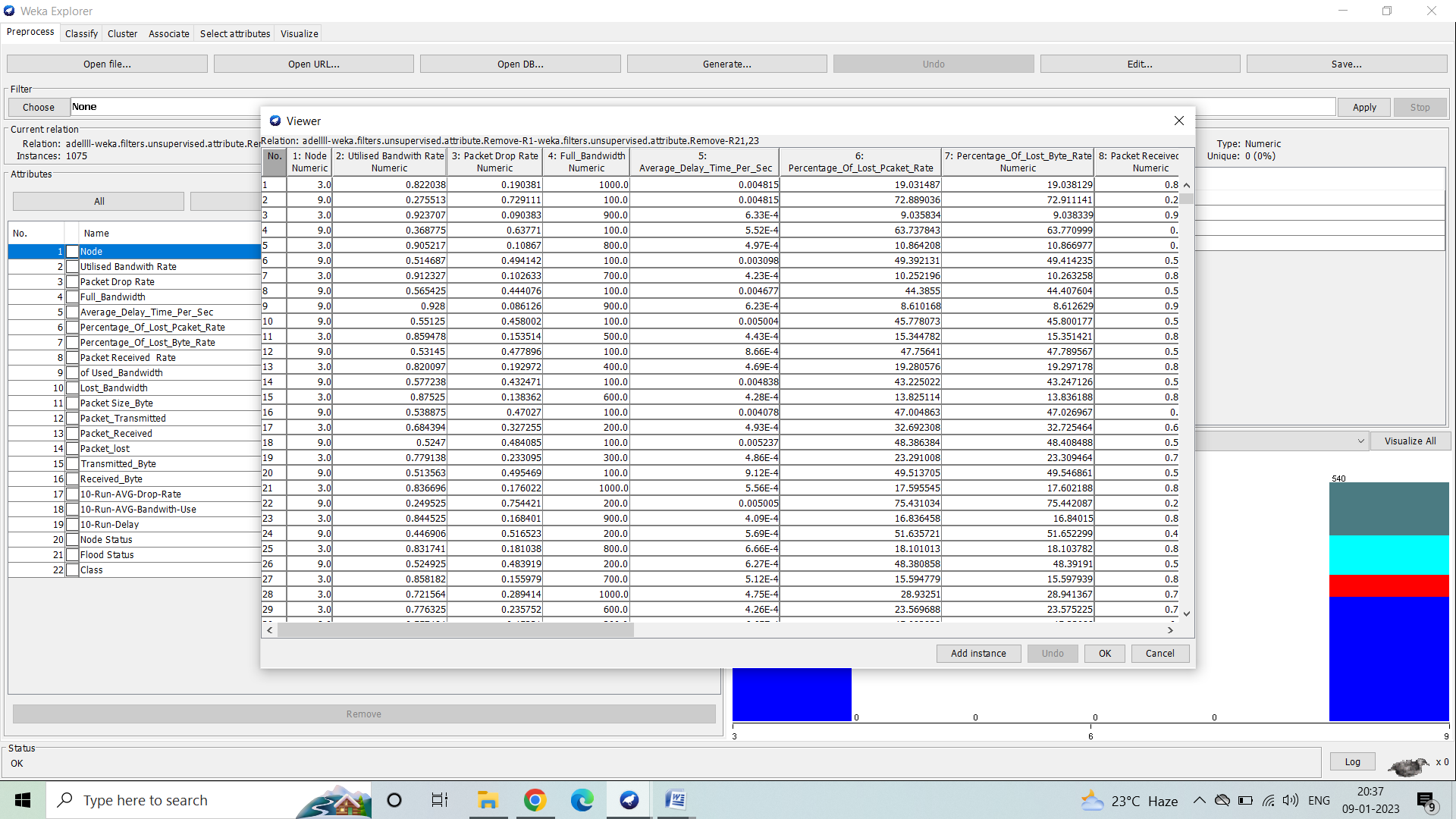
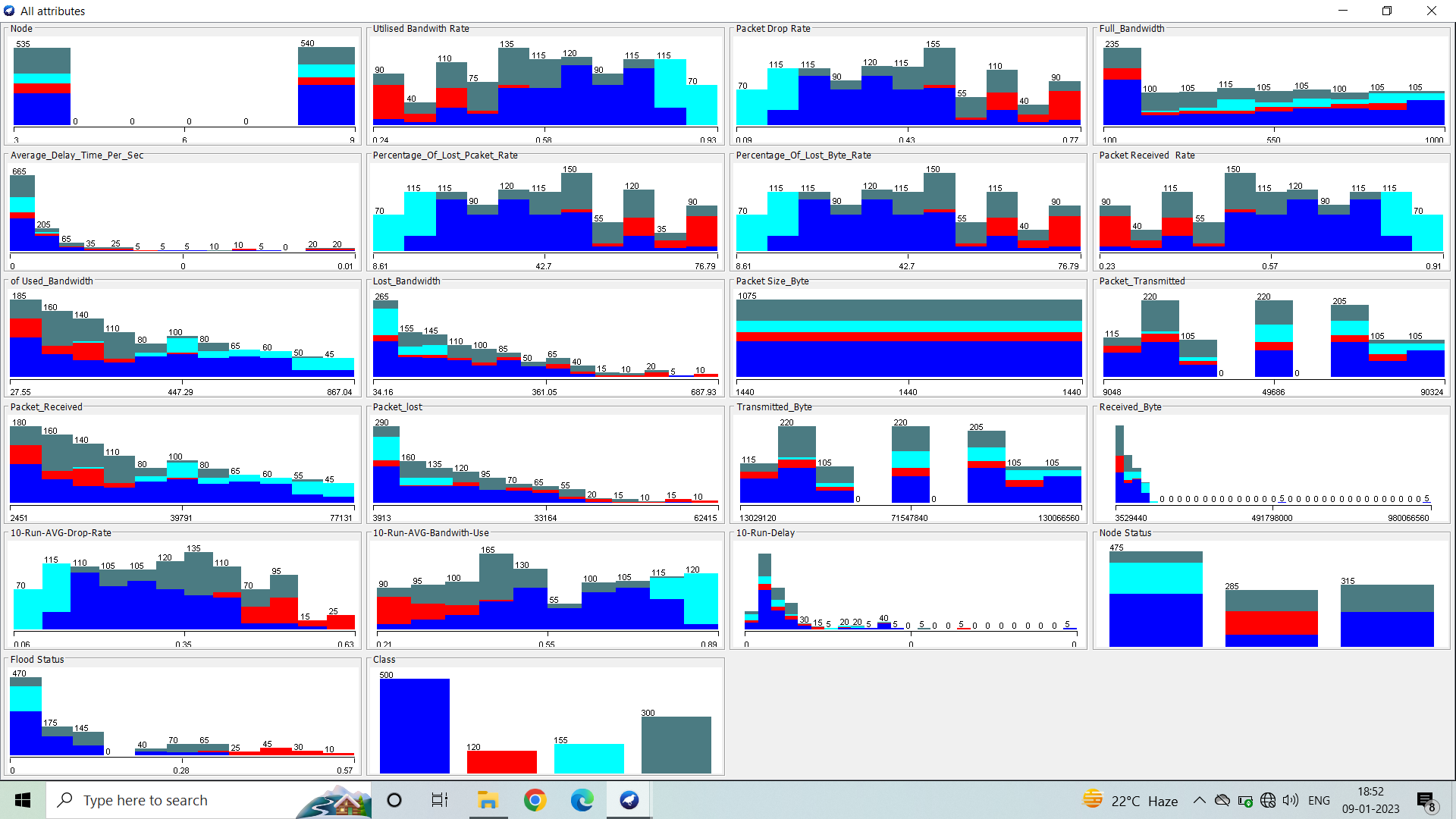
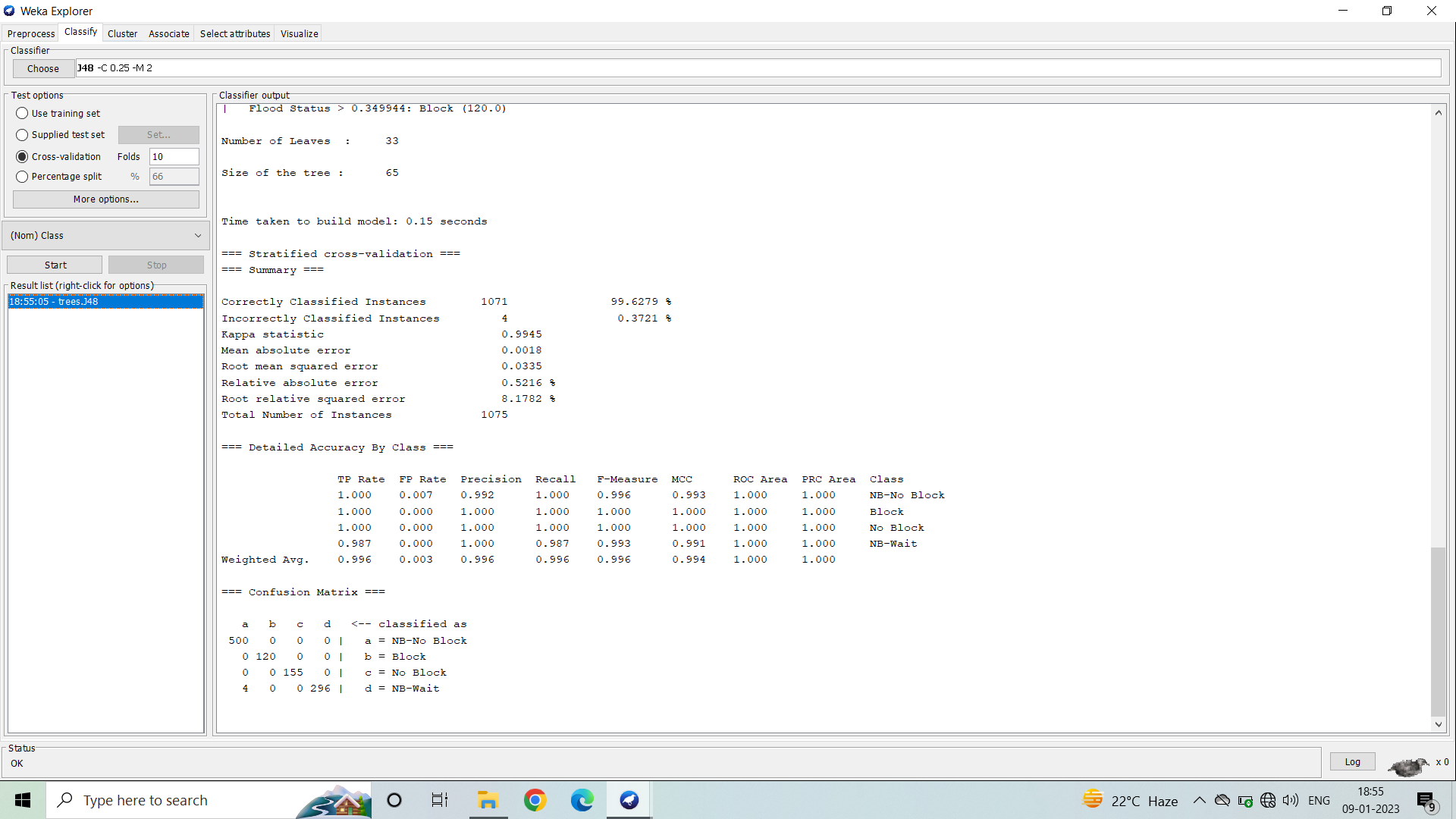


Fig.1.3: Visualization:



Result:



=== Run information ===

Scheme:weka.classifiers.trees.J48C0.25M2

Relation:adellllweka.filters.unsupervised.attribute.RemoveR1weka.filters.unsupervisd.attribute.Remove-R21,23

Instances: 1075

Attributes: 22

Node

Utilised Bandwith Rate

Packet Drop Rate

Full\_Bandwidth

Average\_Delay\_Time\_Per\_Sec

Percentage\_Of\_Lost\_Pcaket\_Rate

Percentage\_Of\_Lost\_Byte\_Rate

Packet Received Rate

of Used\_Bandwidth

Lost\_Bandwidth

Packet Size\_Byte

Packet\_Transmitted

Packet\_Received

Packet\_lost

Transmitted\_Byte

Received\_Byte

10-Run-AVG-Drop-Rate

10-Run-AVG-Bandwith-Use

10-Run-Delay

Node Status

Flood Status

Class

Test mode: evaluate on training data

Number of Leaves : 33

Size of the tree : 65

Time taken to build model: 0.12 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0.03 seconds

=== Summary ===

Correctly Classified Instances 1075 100 %

Incorrectly Classified Instances 0 0 %

Kappa statistic 1

Mean absolute error 0

Root mean squared error 0

Relative absolute error 0 %

Root relative squared error 0 %

Total Number of Instances 1075

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

1.000 0.000 1.000 1.000 1.000 1.000 1.000 1.000 NB-No Block

1.000 0.000 1.000 1.000 1.000 1.000 1.000 1.000 Block

1.000 0.000 1.000 1.000 1.000 1.000 1.000 1.000 No Block

1.000 0.000 1.000 1.000 1.000 1.000 1.000 1.000 NB-Wait

Weighted Avg. 1.000 0.000 1.000 1.000 1.000 1.000 1.000 1.000

=== Confusion Matrix ===

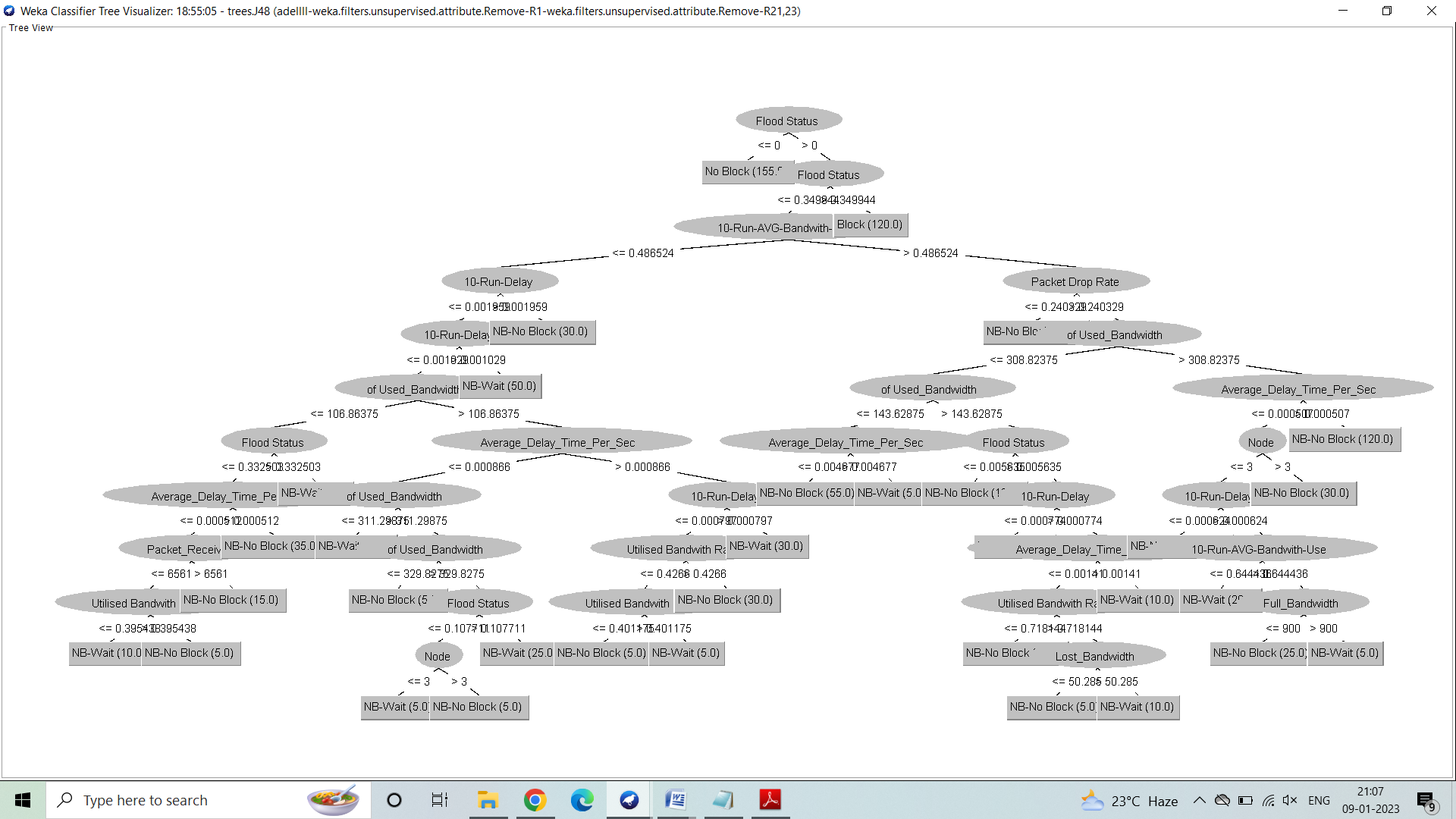
a b c d <-- classified as

500 0 0 0 | a = NB-No Block

0 120 0 0 | b = Block

0 0 155 0 | c = No Block

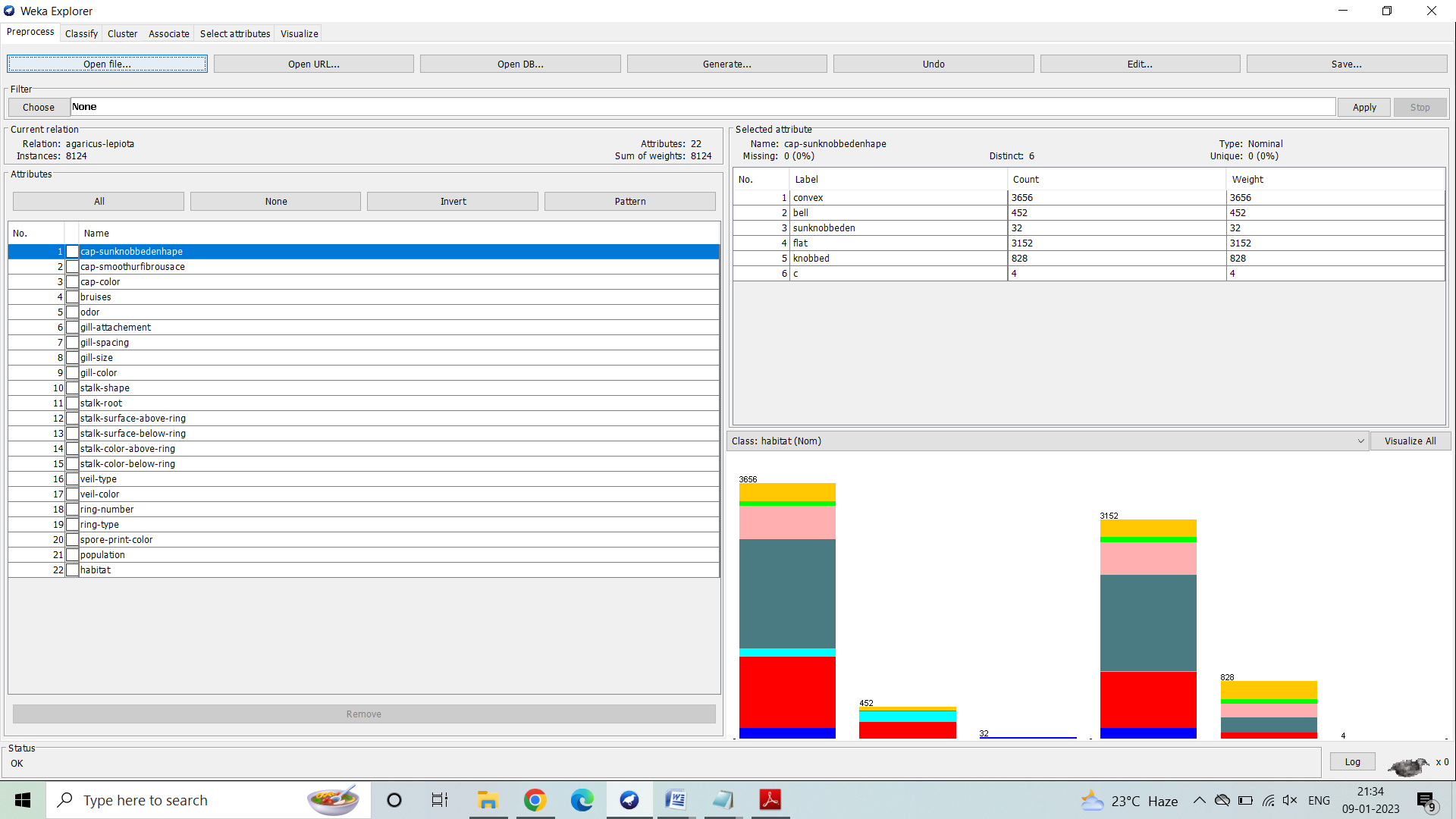
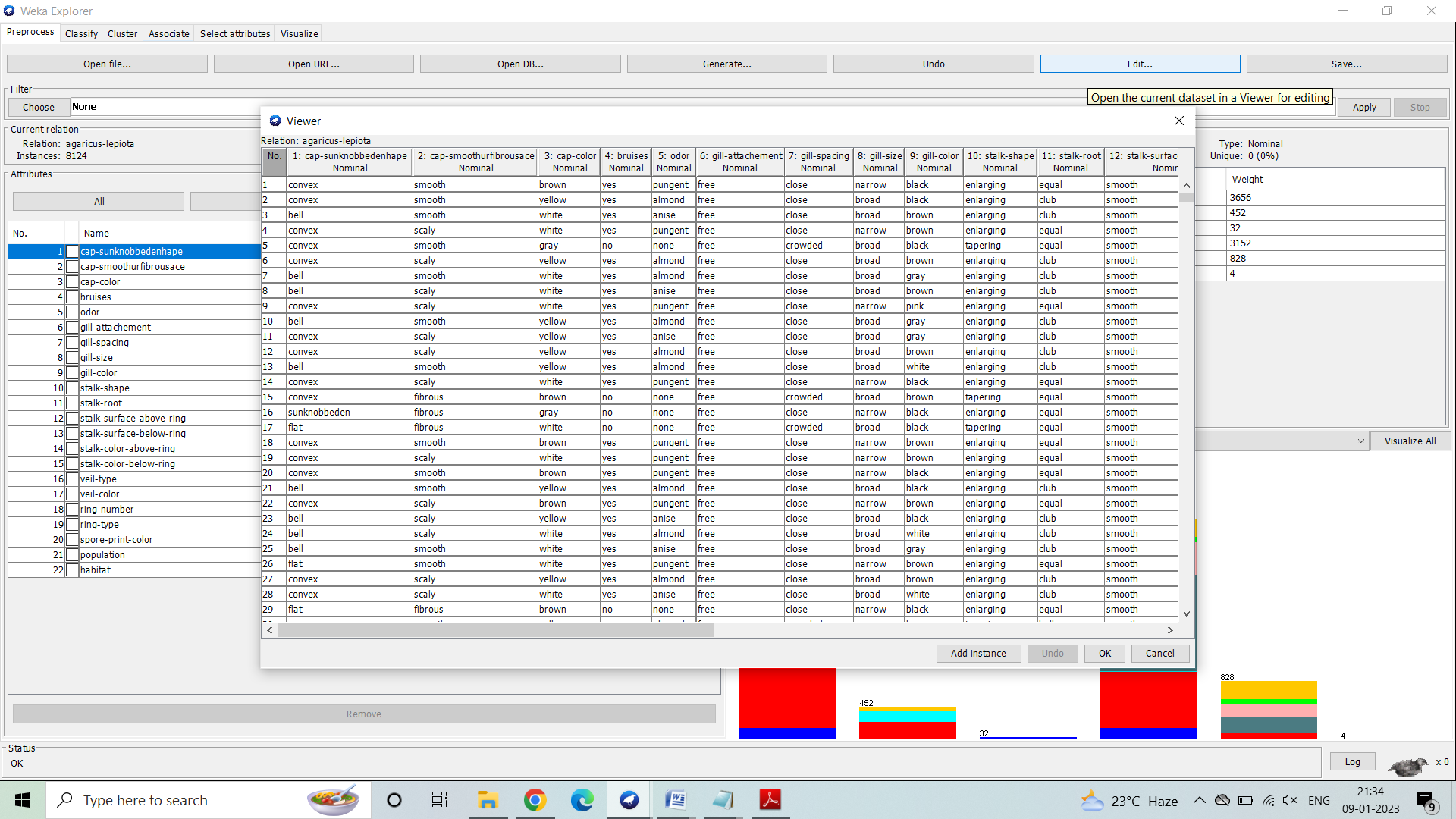
0 0 0 300 | d = NB-Wait



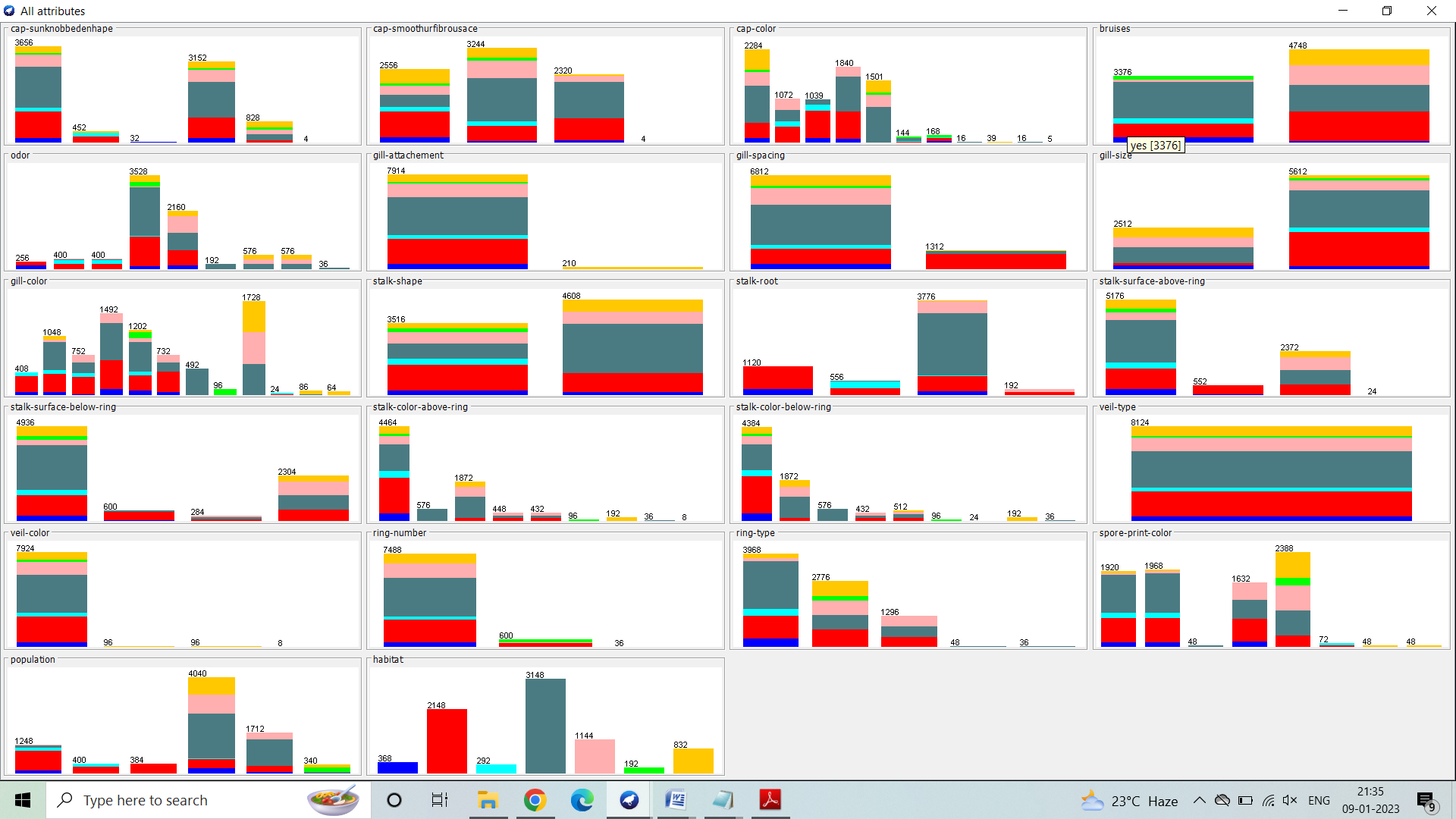
**Decision Tree has been successfully constructed for BHP** **flooding attack on OBS network** **dataset using Weka.**

**For Nominal data:**

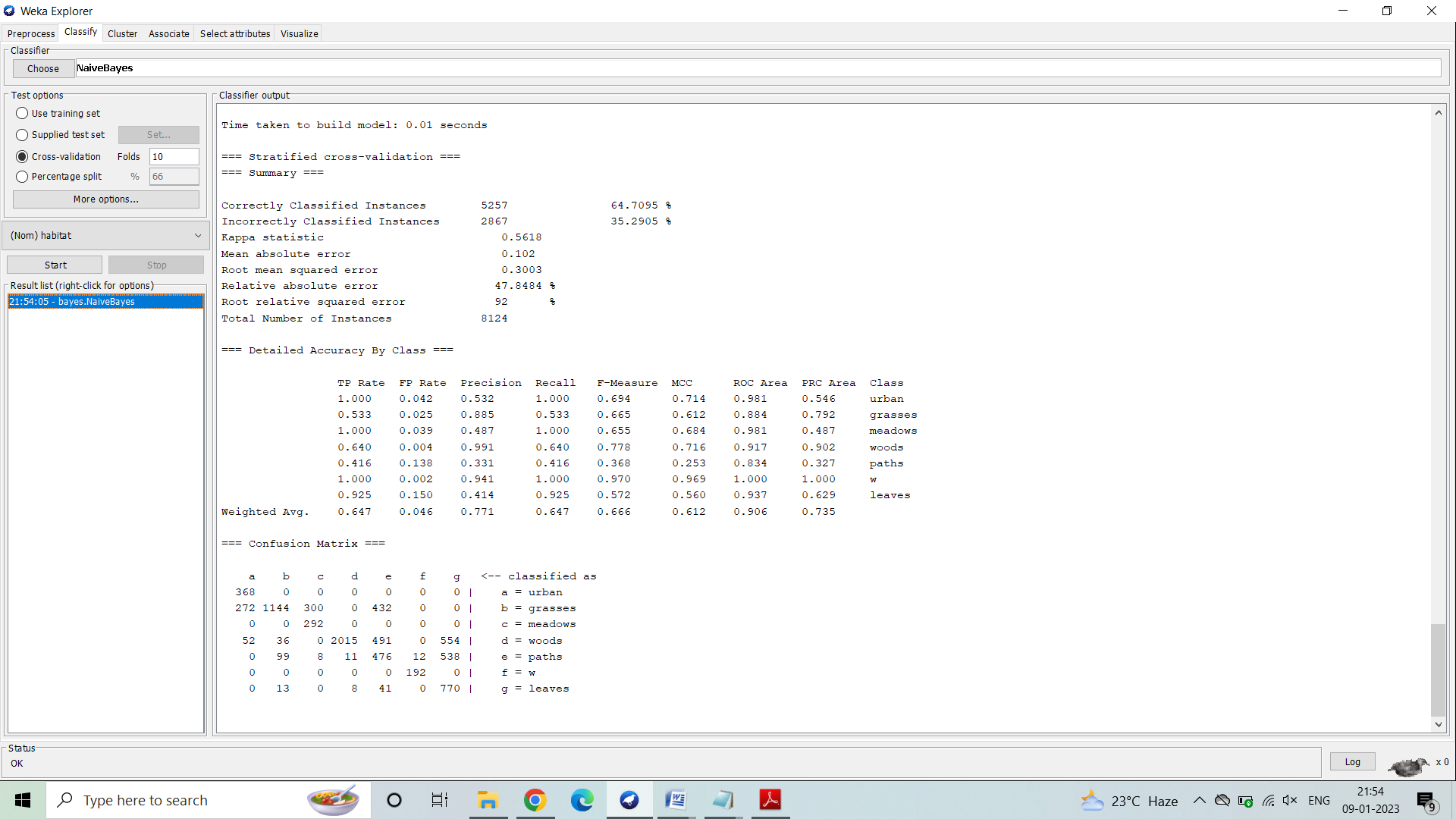
**Fig.1.1: Overview**

**Fig.1.2:Dataset**

**Fig.1.3: Visualization**

****

Result:



=== Run information ===

Scheme: weka.classifiers.trees.J48 -C 0.25 -M 2

Relation: agaricus-lepiota

Instances: 8124

Attributes: 22

cap-sunknobbedenhape

cap-smoothurfibrousace

cap-color

bruises

odor

gill-attachement

gill-spacing

gill-size

gill-color

stalk-shape

stalk-root

stalk-surface-above-ring

stalk-surface-below-ring

stalk-color-above-ring

stalk-color-below-ring

veil-type

veil-color

ring-number

ring-type

spore-print-color

population

habitat

Test mode: evaluate on training data

Number of Leaves : 62

Size of the tree : 77

Time taken to build model: 0.13 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0.05 seconds

=== Summary ===

Correctly Classified Instances 5440 66.9621 %

Incorrectly Classified Instances 2684 33.0379 %

Kappa statistic 0.5394

Mean absolute error 0.0989

Root mean squared error 0.2224

Relative absolute error 46.4197 %

Root relative squared error 68.1364 %

Total Number of Instances 8124

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

1.000 0.035 0.575 1.000 0.730 0.745 0.987 0.686 urban

0.750 0.129 0.676 0.750 0.711 0.602 0.944 0.840 grasses

0.000 0.000 ? 0.000 ? ? 0.969 0.387 meadows

0.860 0.232 0.702 0.860 0.773 0.613 0.932 0.889 woods

0.266 0.070 0.384 0.266 0.314 0.230 0.850 0.365 paths

1.000 0.000 1.000 1.000 1.000 1.000 1.000 1.000 w

0.308 0.000 1.000 0.308 0.471 0.534 0.944 0.593 leaves

Weighted Avg. 0.670 0.135 ? 0.670 ? ? 0.930 0.747

=== Confusion Matrix ===

a b c d e f g <-- classified as

368 0 0 0 0 0 0 | a = urban

272 1612 0 0 264 0 0 | b = grasses

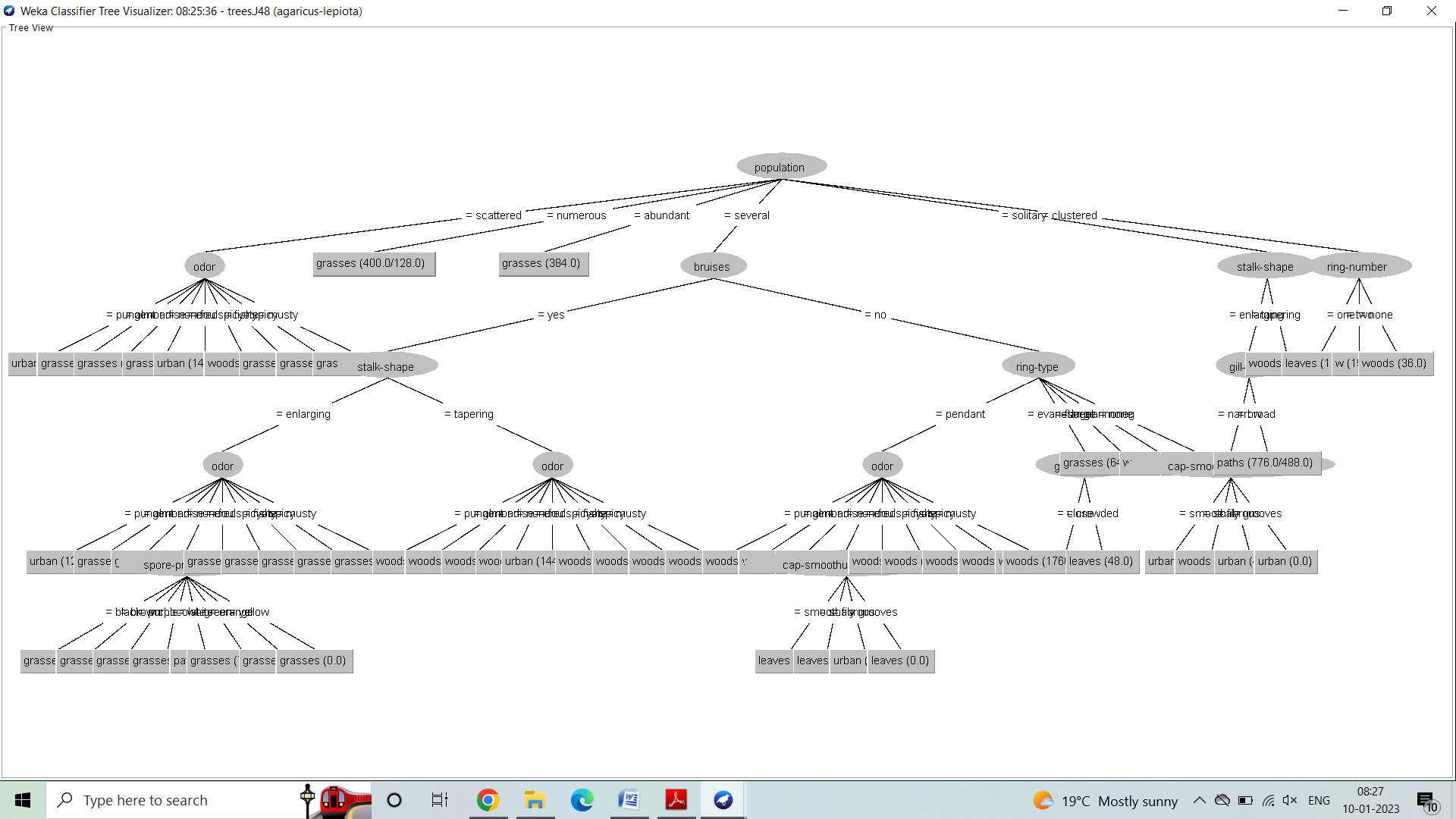
0 292 0 0 0 0 0 | c = meadows

0 216 0 2708 224 0 0 | d = woods

0 264 0 576 304 0 0 | e = paths

0 0 0 0 0 192 0 | f = w

0 0 0 576 0 0 256 | g = leaves



**Decision Tree has been successfully constructed for mushroom dataset using Weka.**

**Applying Naive Bayesian Filter:**

**Description of the Problem:**

* Bayesian Learning provides a probabilistic approach to inference.
* It is based on the assumption that the quantities of interest are governed by probability distributions and that optimal decisions can be made by reasoning about these probabilities together with observed data.
* Bayesian learning algorithms calculate explicit probabilities for hypotheses.
* Each observed training example can incrementally decrease or increase the estimated probability that a hypothesis is correct. This provides a more flexible approach to learning than algorithms that completely eliminate a hypothesis if it is found to be inconsistent with any single example.
* Prior knowledge can be combined with observed data to determine the final probability of a hypothesis. In Bayesian learning, prior knowledge is provided by asserting

1) a prior probability for each candidate hypothesis, and

2) a probability distribution over observed data for each possible hypothesis.

* Bayesian methods can accommodate hypotheses that make probabilistic predictions (e.g., hypotheses such as "this pneumonia patient has a 93% chance of complete recovery").
* New instances can be classified by combining the predictions of multiple hypotheses, weighted by their probabilities.
* They require initial knowledge of many probabilities. When these probabilities are not known in advance they are often estimated based on background knowledge, previously available data, and assumptions about the form of the underlying distributions.
* They require significant computational cost.
* They can provide a standard of optimal decision making against which other practical methods can be measured.

**Implementation in Weka:**

1) Open Start -> Programs->Weka

2) Open explorer.

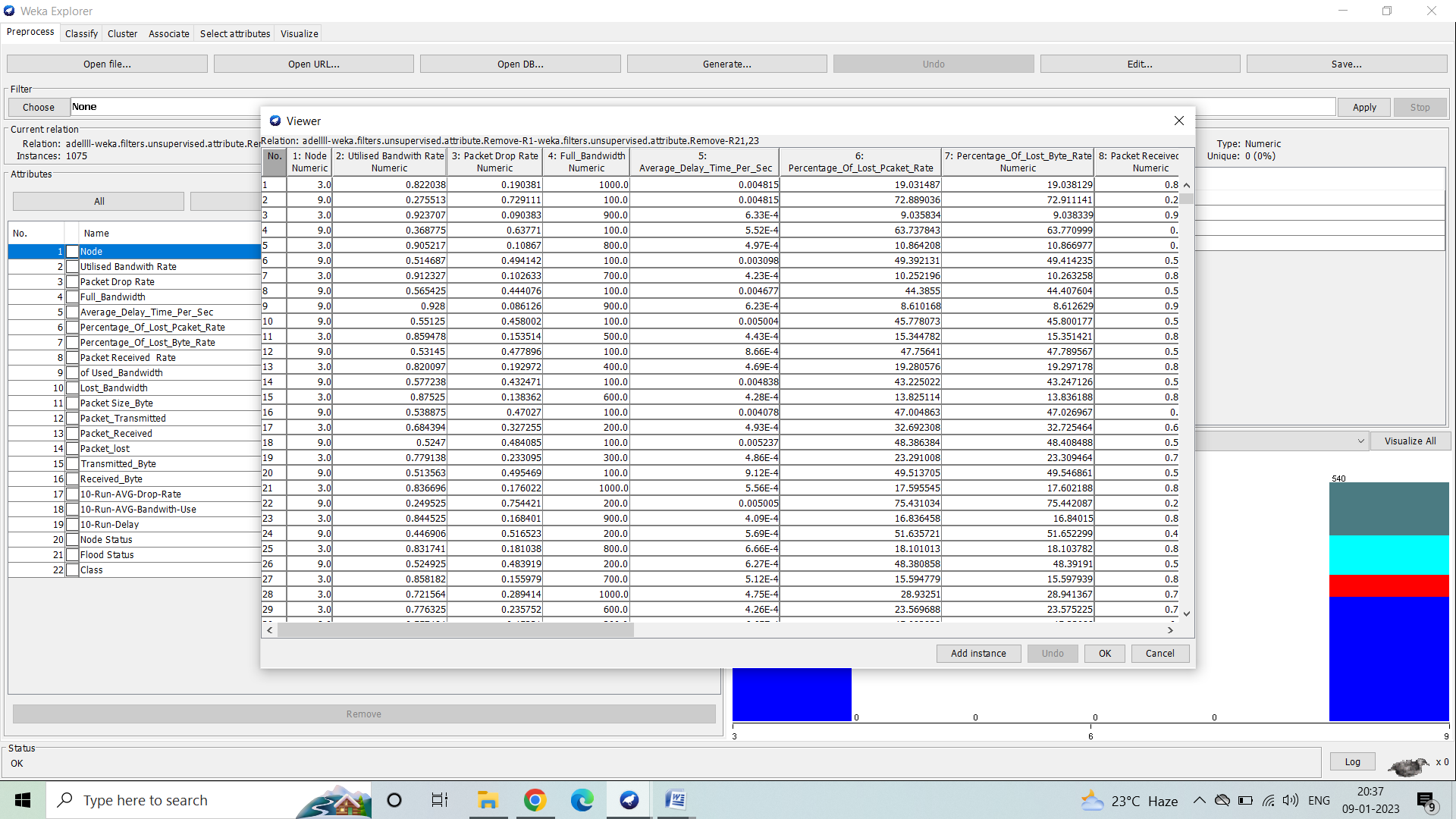
3) Click on open file and select Nursery.csv

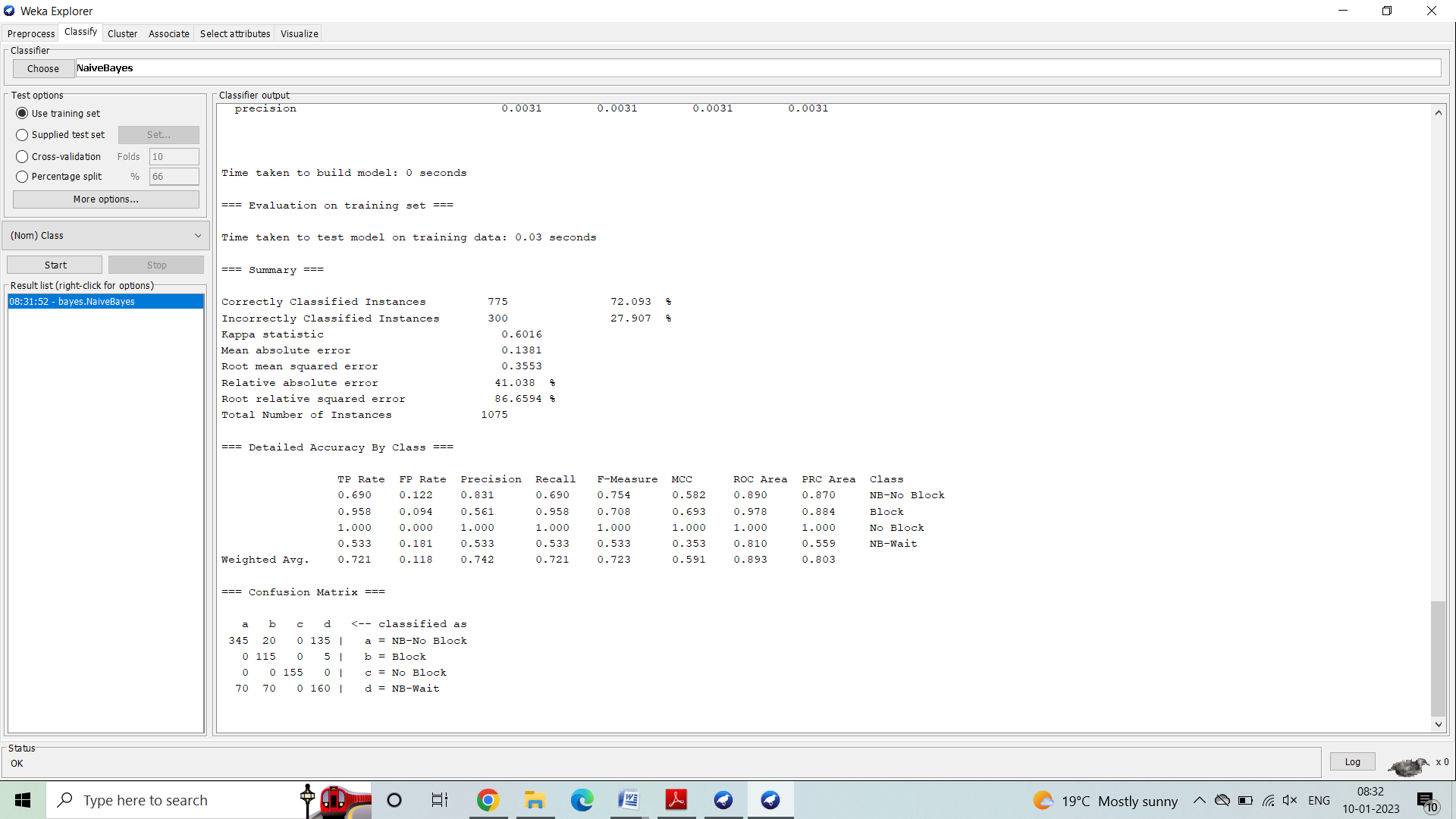
4) Select Classify option on the top of the Menu bar.

5) Select Choose button and click on weka->classifier->bayes->NaiveBayes.

6) Click on Start button and output will be displayed on the right side of the window.

**For Numerical data:**

**Dataset:**

**Result:**

=== Run information ===

Scheme: weka.classifiers.bayes.NaiveBayes

Relation:adellllweka.filters.unsupervised.attribute.RemoveR1weka.filters.unsupervisd.attribute.Remove-R21,23

Instances: 1075

Attributes: 22

Node

Utilised Bandwith Rate

Packet Drop Rate

Full\_Bandwidth

Average\_Delay\_Time\_Per\_Sec

Percentage\_Of\_Lost\_Pcaket\_Rate

Percentage\_Of\_Lost\_Byte\_Rate

Packet Received Rate

of Used\_Bandwidth

Lost\_Bandwidth

Packet Size\_Byte

Packet\_Transmitted

Packet\_Received

Packet\_lost

Transmitted\_Byte

Received\_Byte

10-Run-AVG-Drop-Rate

10-Run-AVG-Bandwith-Use

10-Run-Delay

Node Status

Flood Status

Class

Test mode: evaluate on training data

=== Evaluation on training set ===

Time taken to test model on training data: 0.03 seconds

=== Summary ===

Correctly Classified Instances 775 72.093 %

Incorrectly Classified Instances 300 27.907 %

Kappa statistic 0.6016

Mean absolute error 0.1381

Root mean squared error 0.3553

Relative absolute error 41.038 %

Root relative squared error 86.6594 %

Total Number of Instances 1075

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

0.690 0.122 0.831 0.690 0.754 0.582 0.890 0.870 NB-No Block

0.958 0.094 0.561 0.958 0.708 0.693 0.978 0.884 Block

1.000 0.000 1.000 1.000 1.000 1.000 1.000 1.000 No Block

0.533 0.181 0.533 0.533 0.533 0.353 0.810 0.559 NB-Wait

Weighted Avg. 0.721 0.118 0.742 0.721 0.723 0.591 0.893 0.803

=== Confusion Matrix ===

a b c d <-- classified as

345 20 0 135 | a = NB-No Block

0 115 0 5 | b = Block

0 0 155 0 | c = No Block

70 70 0 160 | d = NB-Wait

a b c d <-- classified as

345 20 0 135 | a = NB-No Block

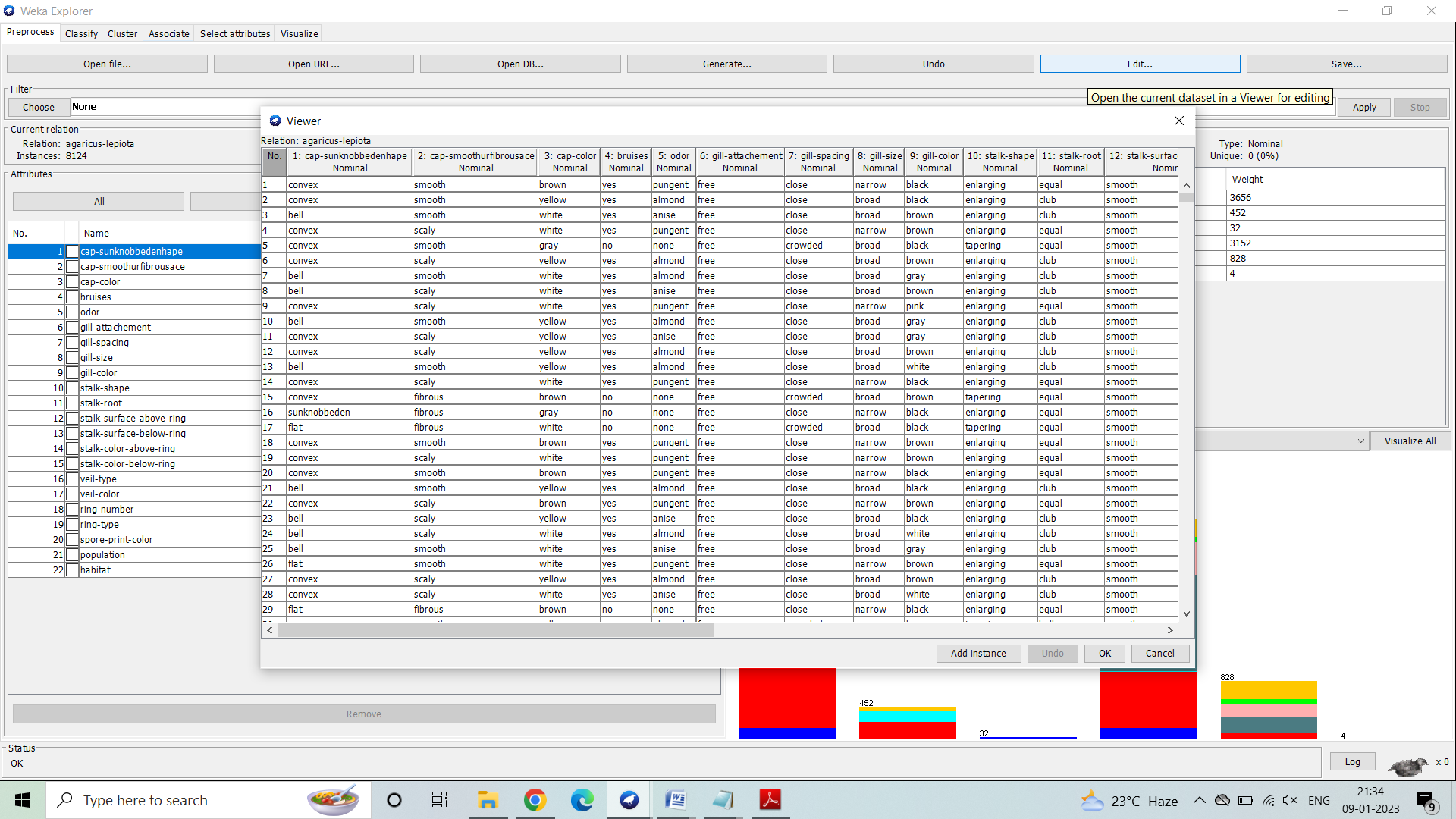
0 115 0 5 | b = Block

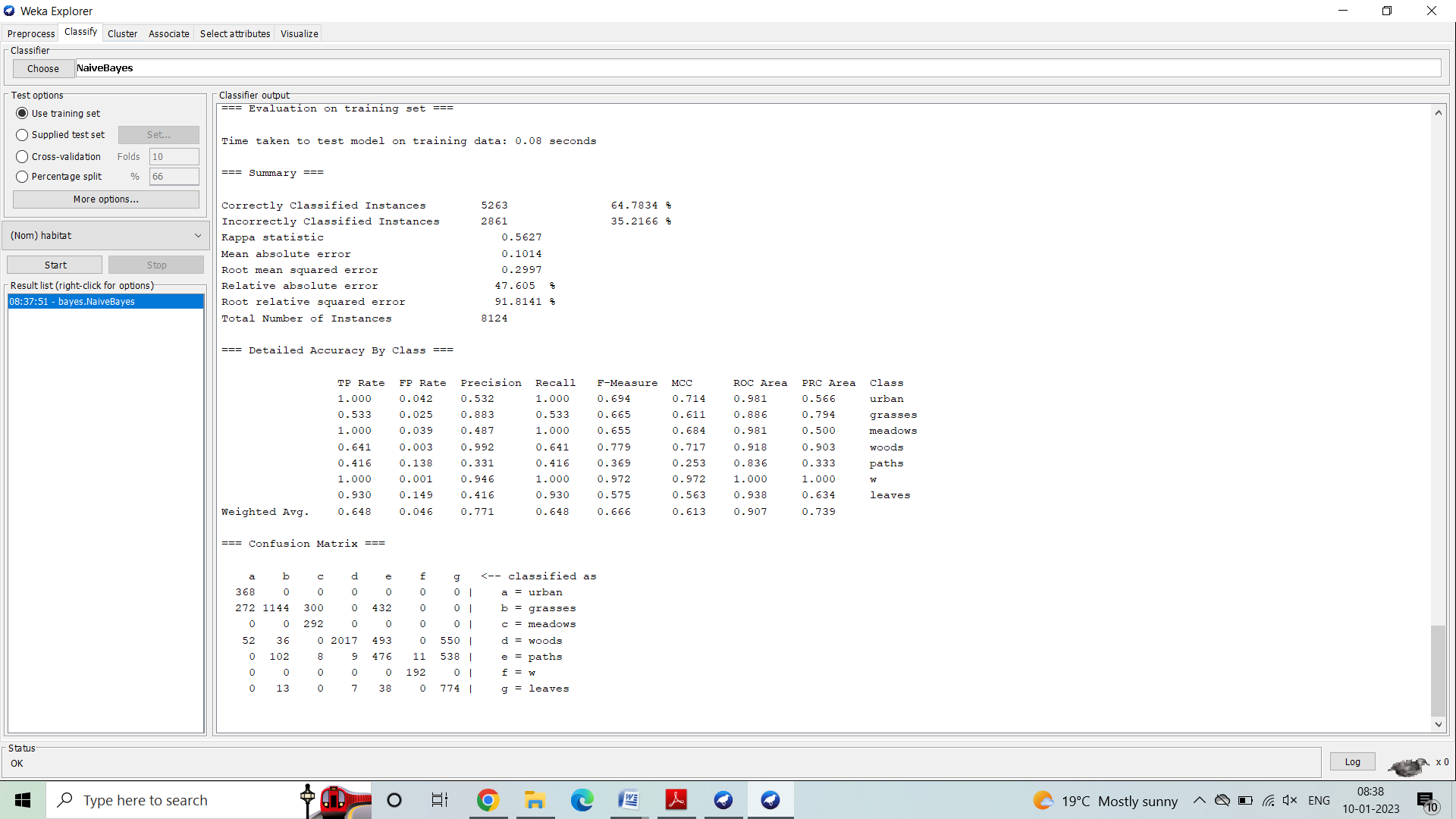
0 0 155 0 | c = No Block

70 70 0 160 | d = NB-Wait

**Bayesian Classifier has been successfully constructed for** BHP flooding attack on OBS network **dataset using Weka.**

**For Nominal data:**

Dataset:

Result:

=== Run information ===

Scheme: weka.classifiers.bayes.NaiveBayes

Relation: agaricus-lepiota

Instances: 8124

Attributes: 22

cap-sunknobbedenhape

cap-smoothurfibrousace

cap-color

bruises

odor

gill-attachement

gill-spacing

gill-size

gill-color

stalk-shape

stalk-root

stalk-surface-above-ring

stalk-surface-below-ring

stalk-color-above-ring

stalk-color-below-ring

veil-type

veil-color

ring-number

ring-type

spore-print-color

population

habitat

Test mode: evaluate on training data

=== Evaluation on training set ===

Time taken to test model on training data: 0.08 seconds

=== Summary ===

Correctly Classified Instances 5263 64.7834 %

Incorrectly Classified Instances 2861 35.2166 %

Kappa statistic 0.5627

Mean absolute error 0.1014

Root mean squared error 0.2997

Relative absolute error 47.605 %

Root relative squared error 91.8141 %

Total Number of Instances 8124

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

1.000 0.042 0.532 1.000 0.694 0.714 0.981 0.566 urban

0.533 0.025 0.883 0.533 0.665 0.611 0.886 0.794 grasses

1.000 0.039 0.487 1.000 0.655 0.684 0.981 0.500 meadows

0.641 0.003 0.992 0.641 0.779 0.717 0.918 0.903 woods

0.416 0.138 0.331 0.416 0.369 0.253 0.836 0.333 paths

1.000 0.001 0.946 1.000 0.972 0.972 1.000 1.000 w

0.930 0.149 0.416 0.930 0.575 0.563 0.938 0.634 leaves

Weighted Avg. 0.648 0.046 0.771 0.648 0.666 0.613 0.907 0.739

=== Confusion Matrix ===

a b c d e f g <-- classified as

368 0 0 0 0 0 0 | a = urban

272 1144 300 0 432 0 0 | b = grasses

0 0 292 0 0 0 0 | c = meadows

52 36 0 2017 493 0 550 | d = woods

0 102 8 9 476 11 538 | e = paths

0 0 0 0 0 192 0 | f = w

0 13 0 7 38 0 774 | g = leaves

**Bayesian Classifier has been successfully constructed for Mushroom dataset using Weka**.

**REGRESSION**

**Dataset:**

|  |  |
| --- | --- |
| **Airfoil Self-Noise Data Set**  **Abstract**: NASA data set, obtained from a series of aerodynamic and acoustic tests of two and three-dimensional airfoil blade sections conducted in an anechoic wind tunnel. |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Data Set Characteristics:** | Multivariate | **Number of Instances:** | 1503 | **Area:** | Physical |
| **Attribute Characteristics:** | Real | **Number of Attributes:** | 6 | **Date Donated** | 2014-03-04 |
| **Associated Tasks:** | Regression | **Missing Values?** | N/A | **Number of Web Hits:** | 195399 |

**Source:**

Provide the names, email addresses, institutions, and other contact information of the donors and creators of the data set.

Donor:Dr Roberto Lopezrobertolopez '@' intelnics.com

Creators:

Thomas F. Brooks, D. Stuart Pope and Michael A. MarcoliniNASA

**Data Set Information:**

The NASA data set comprises different size NACA 0012 airfoils at various wind tunnel speeds and angles of attack. The span of the airfoil and the observer position were the same in all of the experiments.

**Attribute Information:**

This problem has the following inputs:

1. Frequency, in Hertzs.

2. Angle of attack, in degrees.

3. Chord length, in meters.

4. Free-stream velocity, in meters per second.

5. Suction side displacement thickness, in meters.

The only output is:

6. Scaled sound pressure level, in decibels.

**Relevant Papers:**

T.F. Brooks, D.S. Pope, and A.M. Marcolini.

Airfoil self-noise and prediction.

Technical report, NASA RP-1218, July 1989.

K. Lau.

A neural networks approach for aerofoil noise prediction.

Masters thesis, Department of Aeronautics.

Imperial College of Science, Technology and Medicine (London, United Kingdom), 2006.

R. Lopez.

Neural Networks for Variational Problems in Engineering.

PhD Thesis, Technical University of Catalonia, 2008.

**Description of the Problem:**

Numeric prediction is the task of predicting continuous values for giveninput. For example,it is required to predict the salary of employees with 10 years ofexperience, or tomorrow’stemperature. The most widely used approach for numeric prediction is regression.

Regression analysis can be used to model the relationship between aset of predictorvariables and a response variable (which is continuous-valued). Theresponse variable isalso referred to as the predicted attribute.

Regression analysis is a good choice when all of the predictor variablesare continuousvalued as well. Many problems can be solved by linear regression, andeven more can betackled by applying transformations to the variables so that a nonlinearproblem can beconverted to a linear one. Several software packages exist to solveregression problems.

Examples include SAS, SPSS, and S-Plus.

Simple Linear regression analysis involves a response variable, y, and asingle predictorvariable, x. It is the simplest form of regression, and models y as a linearfunction of x.

That is,

y = b + wx

where the variance of y is assumed to be constant, and b and w areregression coefficientsspecifying the Y-intercept and slope of the line, respectively**.**

**Implementation in Weka:**

**Procedure:**

1) Open Start -> Programs -> Weka

2) Open explorer.

3) Click on open file and select file.

4) Select Classify option on the top of the Menu bar.

5) Select Choose button and click on

Weka ->classifiers - >functions - >Linear Regression.

1. Click on Start button and output will be displayed on the right side of the window.

Fig.1.1: Overview

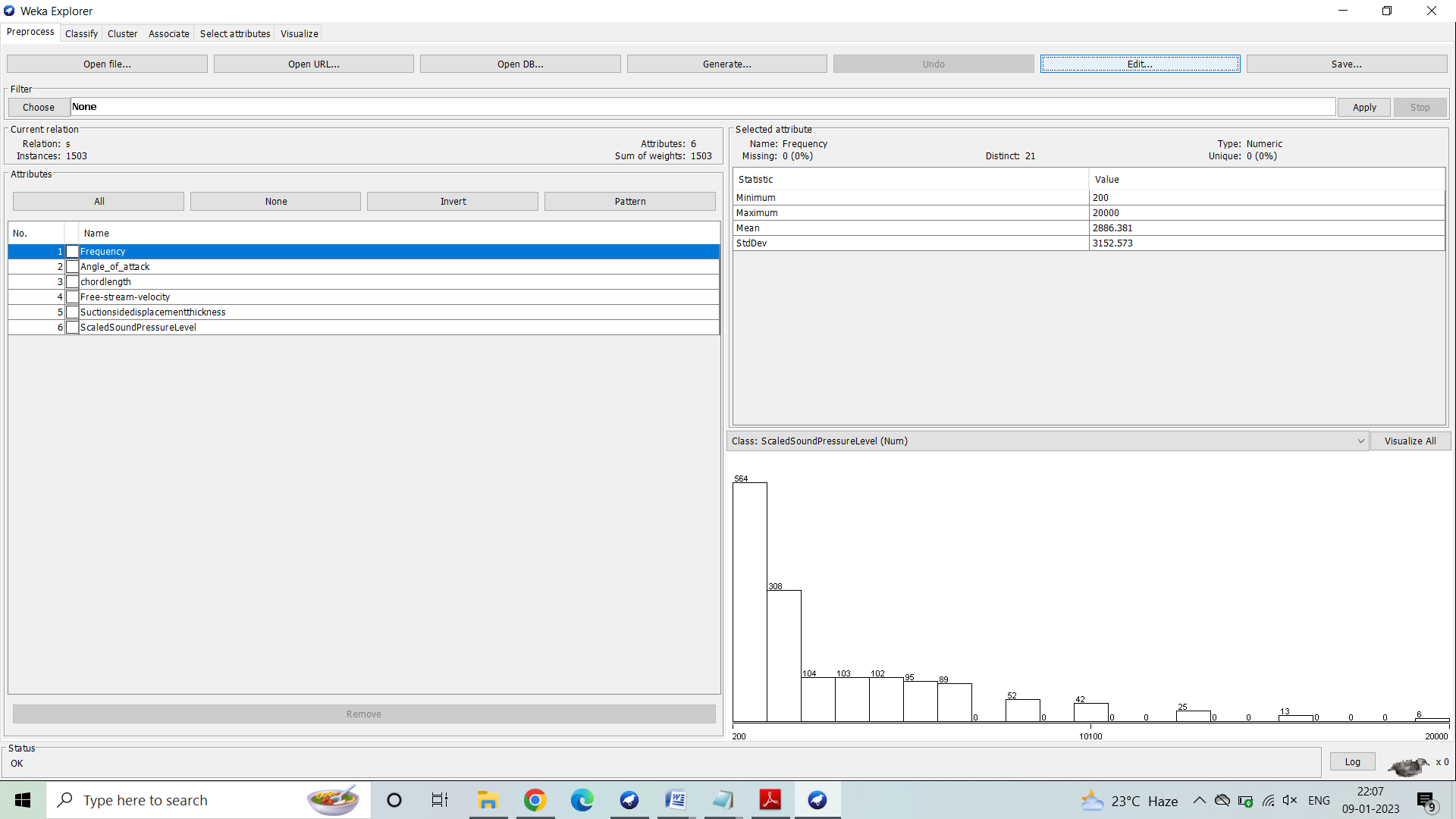


Fig.1.2: Dataset

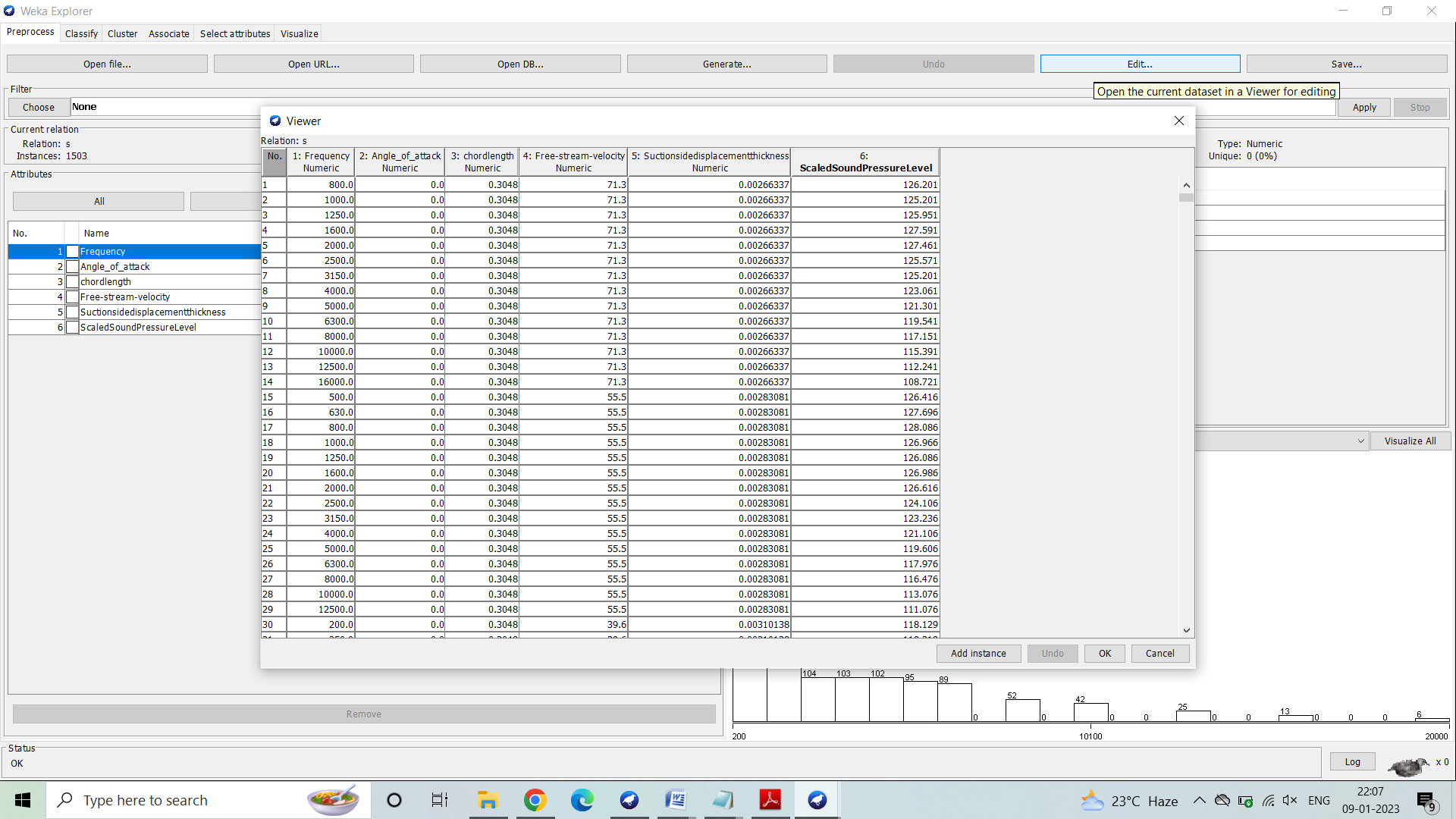
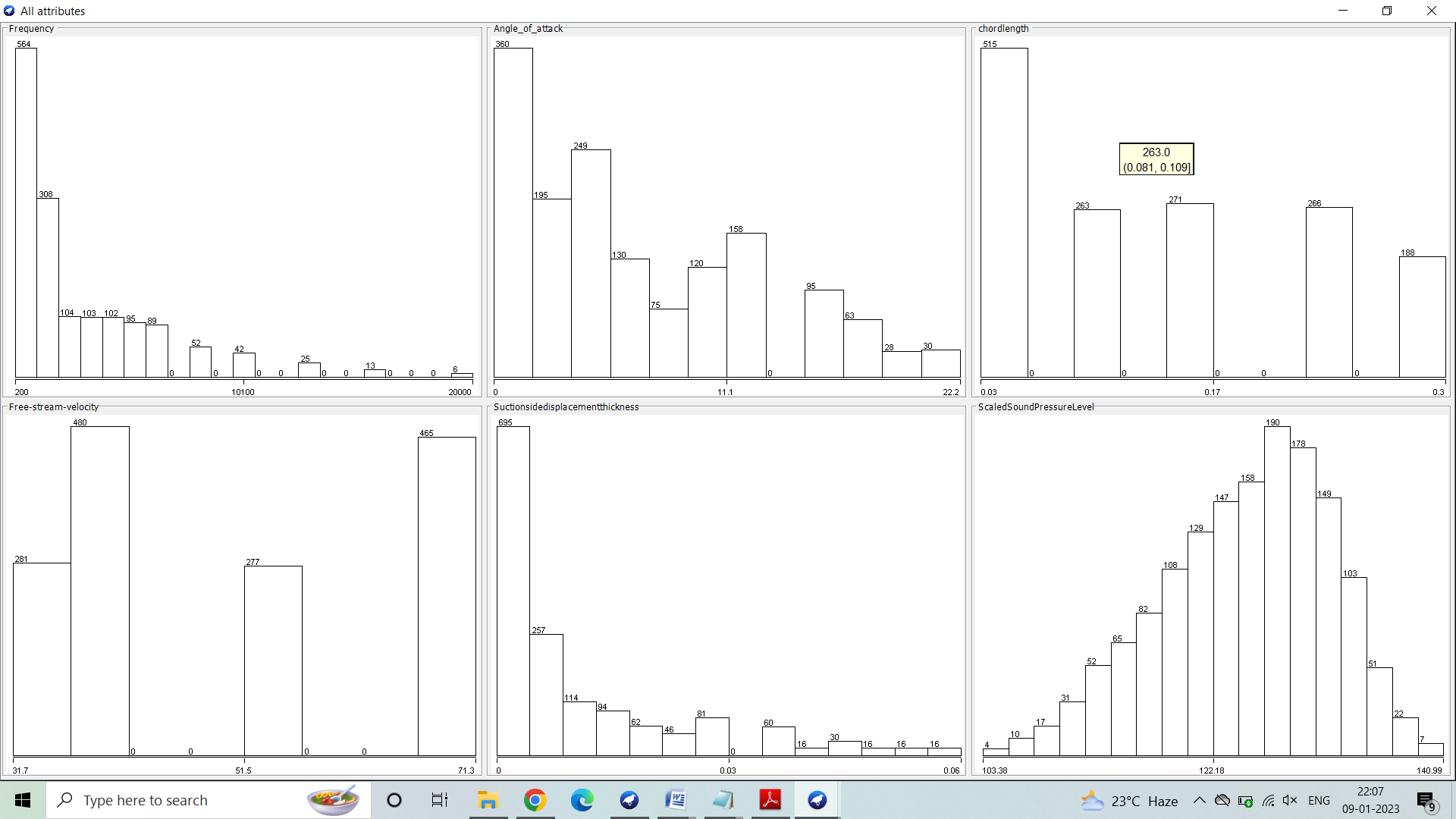
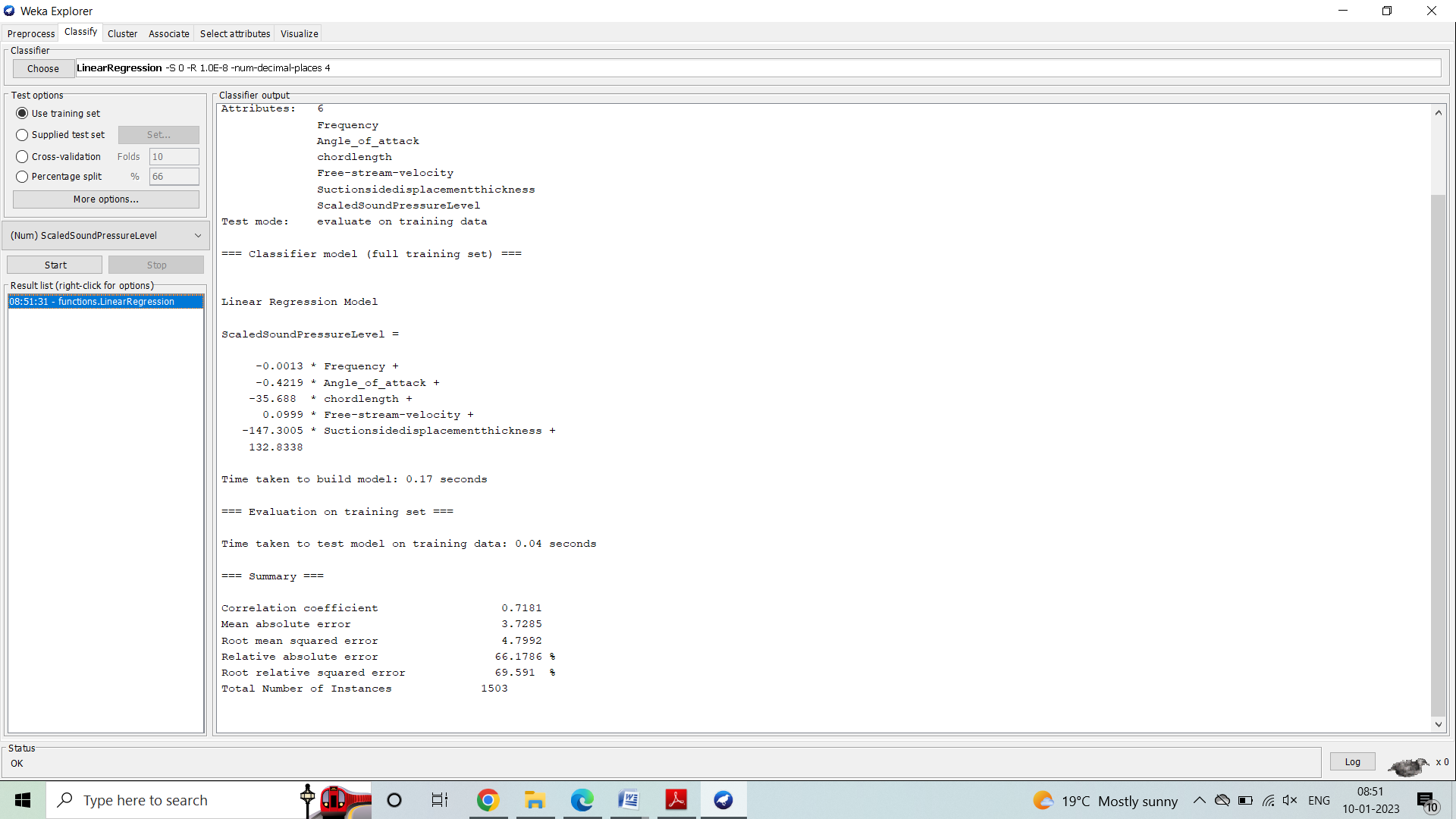


Fig.1.3: Visualization



Result:



=== Run information ===

Scheme: weka.classifiers.functions.LinearRegression -S 0 -R 1.0E-8 -num-decimal-places 4

Relation: s

Instances: 1503

Attributes: 6

Frequency

Angle\_of\_attack

chordlength

Free-stream-velocity

Suctionsidedisplacementthickness

ScaledSoundPressureLevel

Test mode: evaluate on training data

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

Linear Regression Model

ScaledSoundPressureLevel =

-0.0013 \* Frequency +

-0.4219 \* Angle\_of\_attack +

-35.688 \* chordlength + 0.0999 \* Free-stream-velocity +

-147.3005 \* Suctionsidedisplacementthickness + 132.8338

=== Evaluation on training set ===

Time taken to test model on training data: 0.04 seconds

=== Summary ===

Correlation coefficient 0.7181

Mean absolute error 3.7285

Root mean squared error 4.7992

Relative absolute error 66.1786 %

Root relative squared error 69.591 %

Total Number of Instances 1503

**Linear Regressor has been successfully constructed for Airfoil self-noise dataset using Weka**

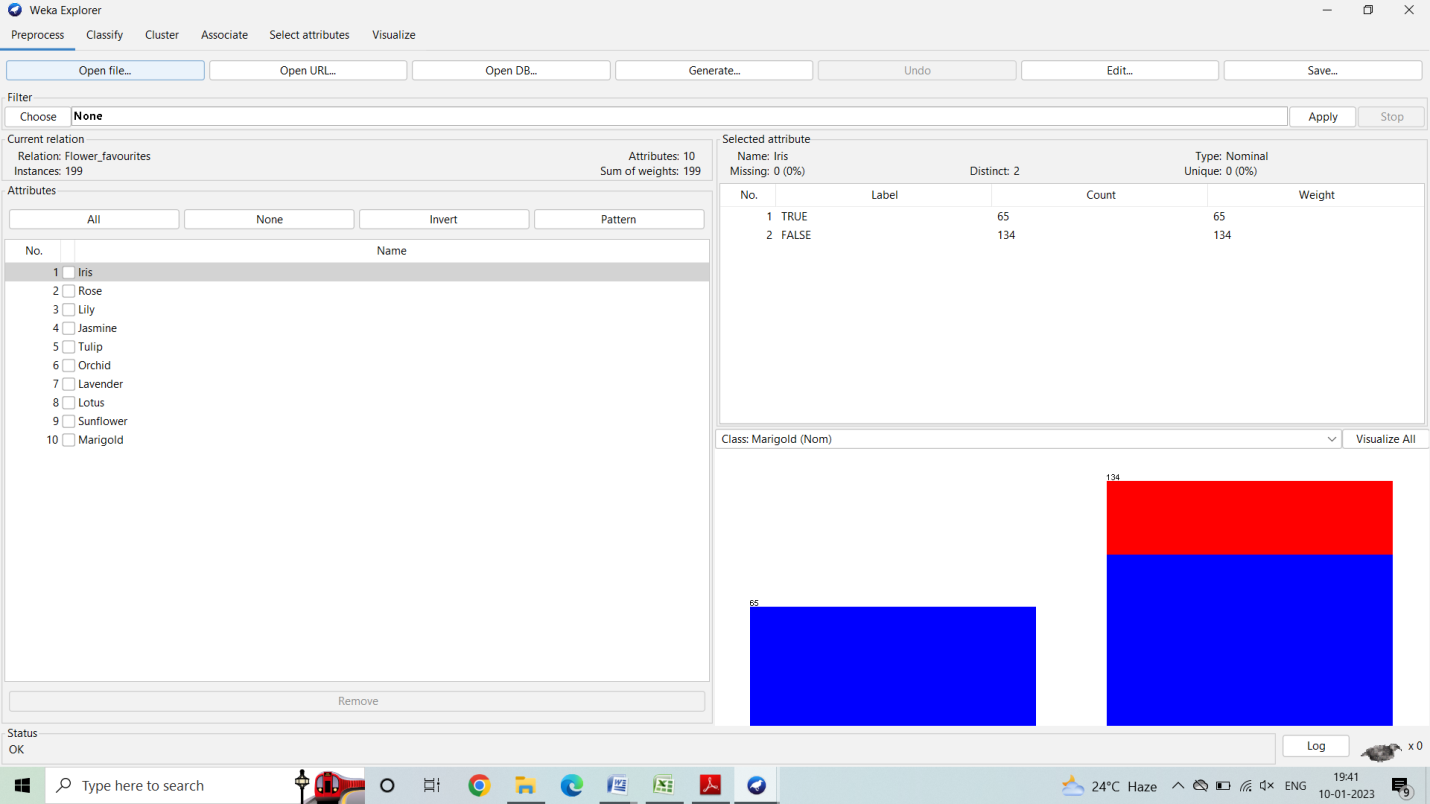
**CLUSTERING**

**Dataset:**

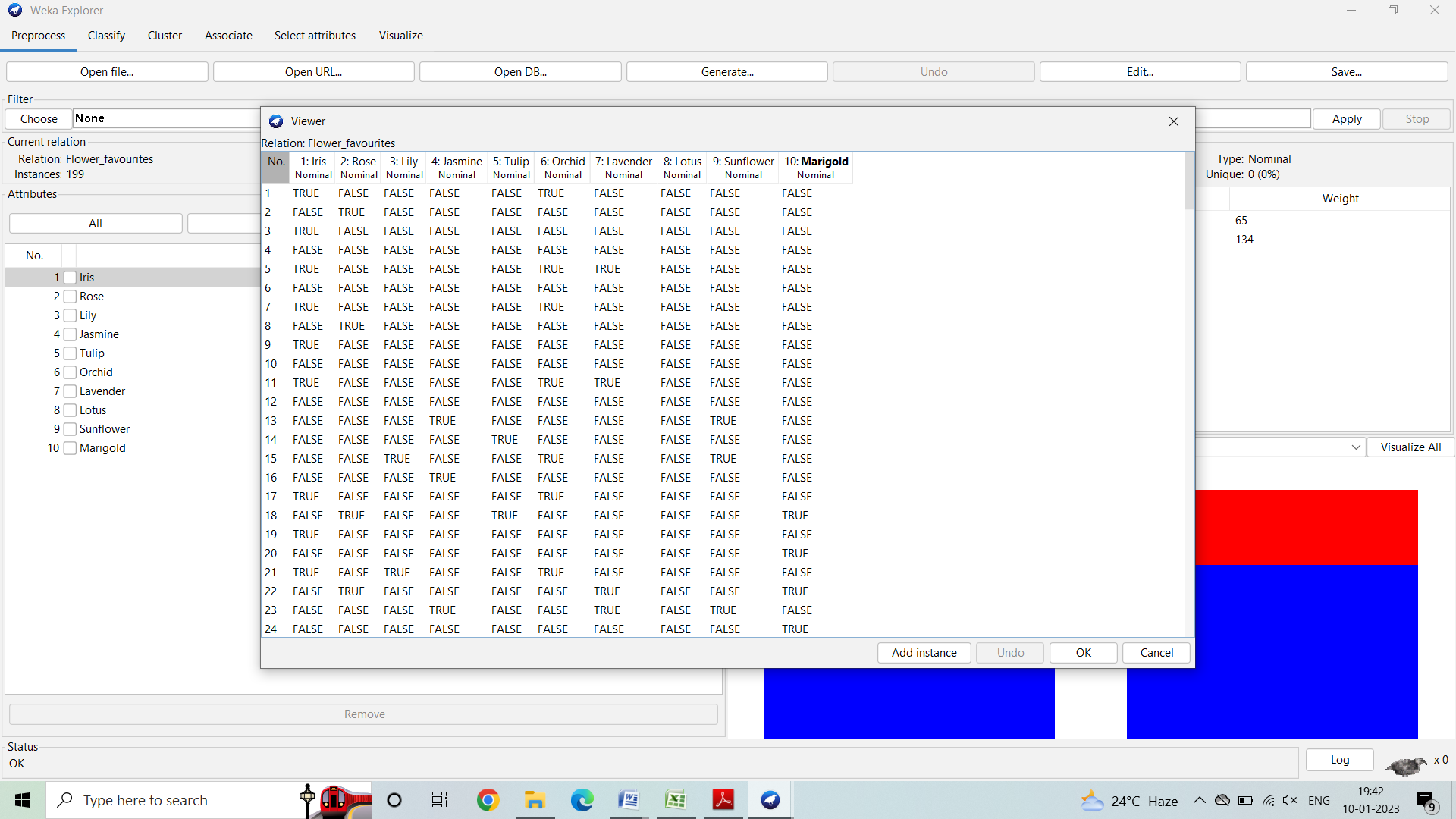
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Travel Reviews Data Set  Abstract: Reviews on destinations in 10 categories mentioned across East Asia. Each traveler rating is mapped as Excellent(4), Very Good(3), Average(2), Poor(1), and Terrible(0) and average rating is used. |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Data Set Characteristics: | Multivariate, Text | Number of Instances: | 980 | Area: | N/A | | Attribute Characteristics: | Real | Number of Attributes: | 11 | Date Donated | 2018-12-19 | | Associated Tasks: | Classification, Clustering | Missing Values? | N/A | Number of Web Hits: | 118572 |   Source:  Shini Renjith, shinirenjith '@' gmail.com  Data Set Information:  This data set is populated by crawling TripAdvisor.com. Reviews on destinations in 10 categories mentioned across East Asia are considered. Each traveler rating is mapped as Excellent (4), Very Good (3), Average (2), Poor (1), and Terrible (0) and average rating is used against each category per user.  **Attribute Information:**  Attribute 1 : Unique user id  Attribute 2 : Average user feedback on art galleries  Attribute 3 : Average user feedback on dance clubs  Attribute 4 : Average user feedback on juice bars  Attribute 5 : Average user feedback on restaurants  Attribute 6 : Average user feedback on museums  Attribute 7 : Average user feedback on resorts  Attribute 8 : Average user feedback on parks/picnic spots  Attribute 9 : Average user feedback on beaches  Attribute 10 : Average user feedback on theaters  Attribute 11 : Average user feedback on religious institutions  **Relevant Papers:**  Renjith, Shini, A. Sreekumar, and M. Jathavedan. 2018. Evaluation of Partitioning Clustering Algorithms for Processing Social Media Data in Tourism Domain In 2018 IEEE Recent Advances in Intelligent Computational Systems (RAICS), 12731. IEEE.  **Description of the Problem**:  Clustering is the process of grouping a set of data objects into multiple groups or clusters  so that objects within a cluster have high similarity, but are very dissimilar to objects in  other clusters. Dissimilarities and similarities are assessed based on the attribute values  describing the objects and often involve distance measures. Clustering as a data mining  tool has its roots in many application areas such as biology, security, business intelligence,  and Web search.  **Algorithm:**  k-means. The k-means algorithm for partitioning, where each clusters center  is represented by the mean value of the objects in the cluster.  Input: k: the number of clusters, D: a data set containing n objects.  Output: A set of k clusters.  **Method:**  (1) Arbitrarily choose k objects from D as the initial cluster centers;  2) Repeat  (3) (Re) assign each object to the cluster to which the object is the most similar,based on  the mean value of the objects in the cluster;  (4) update the cluster means, that is, calculate the mean value of the objects for each  cluster; (5) until no change;  **Implementation in Weka:**  Steps for run K-mean Clustering algorithms in WEKA  1. Open WEKA Tool.  2. Click on WEKA Explorer.  3. Click on Preprocessing tab button.  4. Click on open file button.  5. Choose Heart failure clinical record data set and open file.  6. Click on cluster tab and Choose k-mean and select use training set test option.  7. Click on start button.  **Fig.1.1: Overview**  **Screenshot (134).png**  **Fig.1.2: Dataset**    **Fig1.3:Visualization**  **Result:**    === Run information ===  Scheme: weka.classifiers.functions.LinearRegression -S 0 -R 1.0E-8 -num-decimal-places 4  Relation: tripadvisor\_review-weka.filters.unsupervised.attribute.Remove-R1  Instances: 980  Attributes: 10  Category 1  Category 2  Category 3  Category 4  Category 5  Category 6  Category 7  Category 8  Category 9  Category 10  Test mode: evaluate on training data  === Classifier model (full training set) ===  Linear Regression Model  Category 10 =  0.0819 \* Category 1 +  0.0556 \* Category 2 +  0.0928 \* Category 3 +  -0.1855 \* Category 4 +  -0.1042 \* Category 6 +  -31.834 \* Category 7 +  0.199 \* Category 8 +  0.039 \* Category 9 + 103.4844  === Evaluation on training set ===  Time taken to test model on training data: 0.02 seconds  === Summary ===  Correlation coefficient 0.7686  Mean absolute error 0.1643  Root mean squared error 0.2055  Relative absolute error 61.01 %  Root relative squared error 63.9761 %  Total Number of Instances 980  **Supervised Learning has been successfully performed using simple k-Means clustering algorithm on Travel Reviews dataset using Weka.**  **ASSOCIATION**  **Dataset:**   |  |  | | --- | --- | | **Flowers\_favourite Data Set** |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **Data Set Characteristics:** | Multivariate | **Number of Instances:** | 200 | **Area:** | Physical | | **Attribute Characteristics:** | Real | **Number of Attributes:** | 10 | **Date Donated** | 2014-03-04 | | **Associated Tasks:** | Regression | **Missing Values?** | N/A | **Number of Web Hits:** | 34757 |   **Source:**  Provide the names, email addresses, institutions, and other contact information of the donors and creators of the data set.  Donor:Dr Roberto Lopezrobertolopez '@' intelnics.com  Creators:  Thomas F. Brooks, D. Stuart Pope and Michael A. Marcolini  NASA  **Data Set Information:**  The NASA data set comprises different size NACA 0012 airfoils at various wind tunnel speeds and angles of attack. The span of the airfoil and the observer position were the same in all of the experiments.  **Attribute Information:**  This problem has the following inputs:  1. Iris  2. Rose  3. Lily  4. Jasmine  5. Tulip  6. Orchid  7. Lavender  8. Lotus  9. Sunflower  10. Marigold  **Relevant Papers:**  T.F. Brooks, D.S. Pope, and A.M. Marcolini.  Airfoil self-noise and prediction.  Technical report, NASA RP-1218, July 1989.  K. Lau.  A neural networks approach for aerofoil noise prediction.  Masters thesis, Department of Aeronautics.  Imperial College of Science, Technology and Medicine (London, United Kingdom), 2006.  R. Lopez.  Neural Networks for Variational Problems in Engineering.  PhD Thesis, Technical University of Catalonia, 2008.  **Description of the problem:**  Association Rule is a popular method for discovering interesting relations between variables in large databases. It is used to identify the most frequent item sets or frequent patterns that occur frequently in a database. To perform association rule mining we need two parameters like:  *Support*:- It is the total probability of the item sets that occurred in the transaction.  *Confidence*:-it is the ratio of probability of two item sets A, B and the probability of the item set A.  Association rule mining is used in different applications like basket analysis, catalog design, clustering, classification etc…  **Apriori Algorithm:** - To implement the association rule we use this algorithm.  Steps to follow while implementing the association rule are:  1. Find the frequent item sets where the items have the minimum support count.  2. Use the frequent item sets to generate association rules.  3. Generate the candidates by performing join operation.  4. Perform pruning operation i.e. if any one of the candidates that have a subset is not frequent in the set then delete it.  5. Also calculate support count for each candidate and delete those candidates whose minimum support count is not satisfied. |
|  |

**APRIORI:**

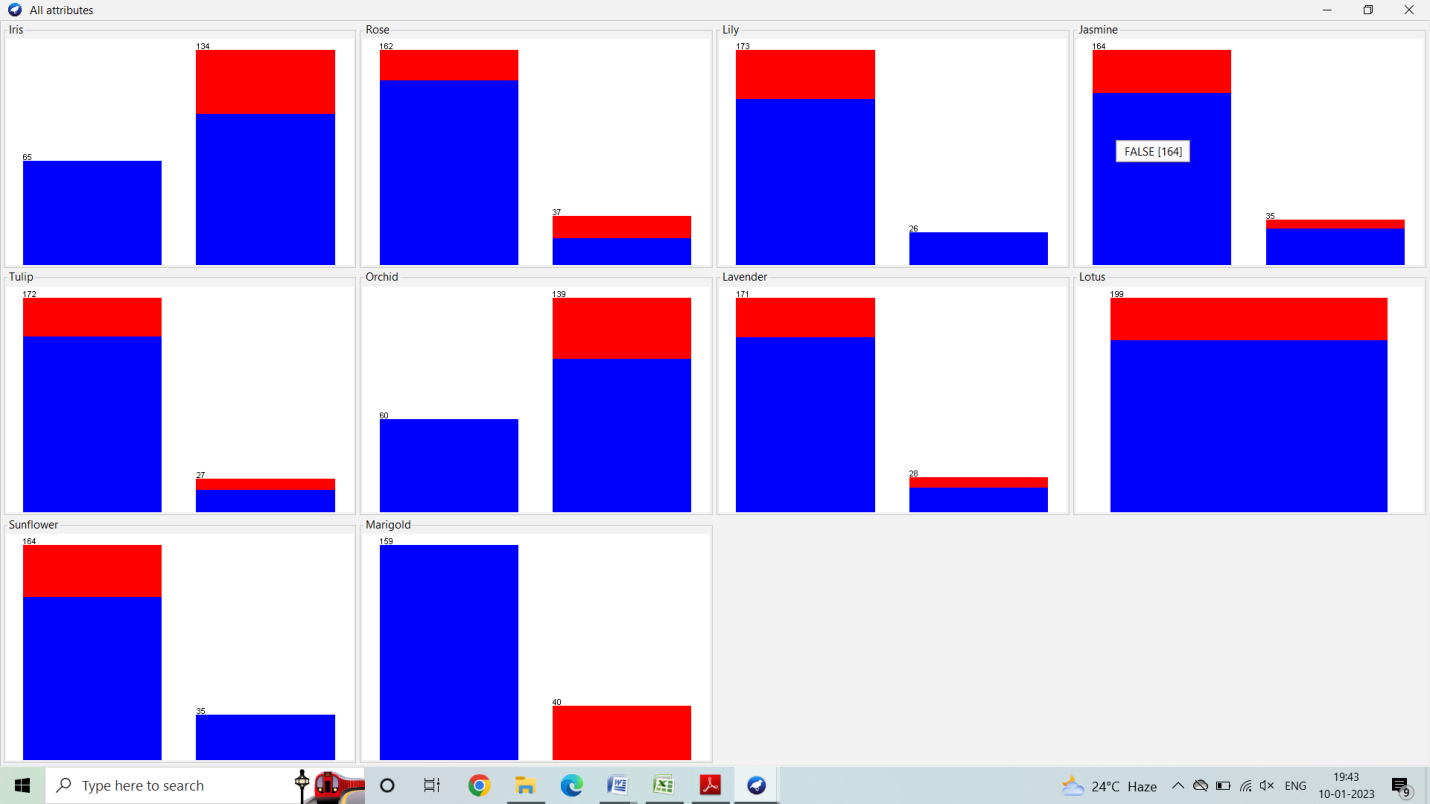
**Fig.1.1: Overview:**

****

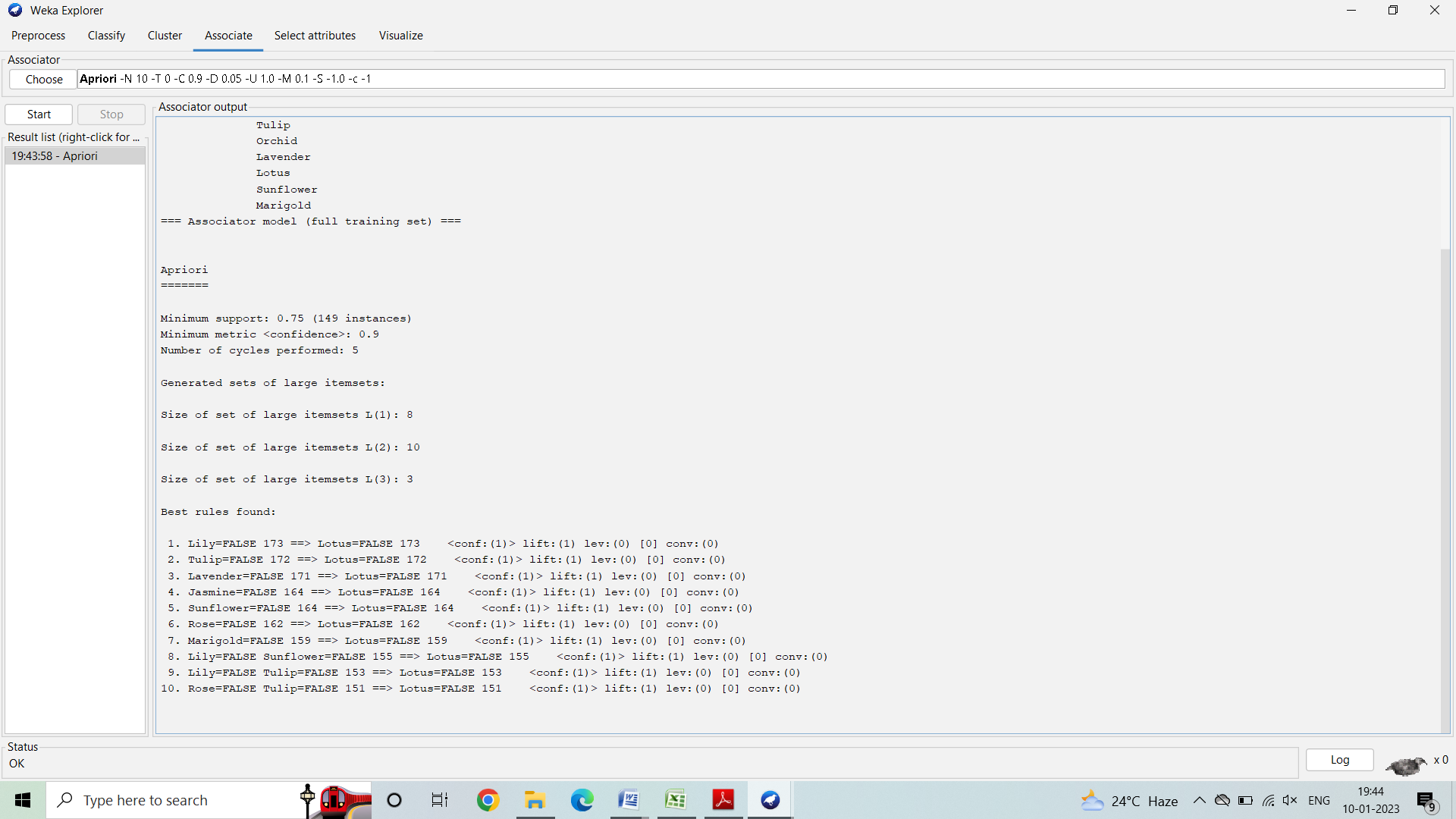
**Fig.1.2: Dataset**

****

**Fig 1.3: Visualization**

****

**Result:**

****

=== Run information ===

Scheme: 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: Flower\_favourites

Instances: 199

Attributes: 10

Iris

Rose

Lily

Jasmine

Tulip

Orchid

Lavender

Lotus

Sunflower

Marigold

=== Associator model (full training set) ===

Apriori

Minimum support: 0.75 (149 instances)

Minimum metric <confidence>: 0.9

Number of cycles performed: 5

Generated sets of large itemsets:

Size of set of large itemsets L(1): 8

Size of set of large itemsets L(2): 10

Size of set of large itemsets L(3): 3

**Best rules found:**

1. Lily=FALSE 173 ==> Lotus=FALSE 173 <conf:(1)> lift:(1) lev:(0) [0] conv:(0)

2. Tulip=FALSE 172 ==> Lotus=FALSE 172 <conf:(1)> lift:(1) lev:(0) [0] conv:(0)

3. Lavender=FALSE 171 ==> Lotus=FALSE 171 <conf:(1)> lift:(1) lev:(0) [0] conv:(0)

4. Jasmine=FALSE 164 ==> Lotus=FALSE 164 <conf:(1)> lift:(1) lev:(0) [0] conv:(0)

5. Sunflower=FALSE 164 ==> Lotus=FALSE 164 <conf:(1)> lift:(1) lev:(0) [0] conv:(0)

6. Rose=FALSE 162 ==> Lotus=FALSE 162 <conf:(1)> lift:(1) lev:(0) [0] conv:(0)

7. Marigold=FALSE 159 ==> Lotus=FALSE 159 <conf:(1)> lift:(1) lev:(0) [0] conv:(0)

8. Lily=FALSE Sunflower=FALSE 155 ==> Lotus=FALSE 155 <conf:(1)> lift:(1) lev:(0) [0] conv:(0)

9. Lily=FALSE Tulip=FALSE 153 ==> Lotus=FALSE 153 <conf:(1)> lift:(1) lev:(0) [0] conv:(0)

10. Rose=FALSE Tulip=FALSE 151 ==> Lotus=FALSE 151 <conf:(1)> lift:(1) lev:(0) [0] conv:(0)

**Association Rules have been successfully generated using Apriori algorithm using Weka.**

**FP-GROWTH:**

**Result:**

=== Run information ===

Scheme: weka.associations.FPGrowth -P 2 -I -1 -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1

Relation: Flower\_favourites

Instances: 199

Attributes: 10

Iris

Rose

Lily

Jasmine

Tulip

Orchid

Lavender

Lotus

Sunflower

Marigold

=== Associator model (full training set) ===

FPGrowth found 22 rules (displaying top 10)

1. [Orchid=FALSE]: 139 ==> [Lotus=FALSE]: 139 <conf:(1)> lift:(1) lev:(0) conv:(0)

2. [Iris=FALSE]: 134 ==> [Lotus=FALSE]: 134 <conf:(1)> lift:(1) lev:(0) conv:(0)

3. [Marigold=TRUE]: 40 ==> [Lotus=FALSE]: 40 <conf:(1)> lift:(1) lev:(0) conv:(0)

4. [Marigold=TRUE]: 40 ==> [Orchid=FALSE]: 40 <conf:(1)> lift:(1.43) lev:(0.06) conv:(12.06)

5. [Marigold=TRUE]: 40 ==> [Iris=FALSE]: 40 <conf:(1)> lift:(1.49) lev:(0.07) conv:(13.07)

6. [Orchid=FALSE, Iris=FALSE]: 117 ==> [Lotus=FALSE]: 117 <conf:(1)> lift:(1) lev:(0) conv:(0)

7. [Marigold=TRUE]: 40 ==> [Lotus=FALSE, Orchid=FALSE]: 40 <conf:(1)> lift:(1.43) lev:(0.06) conv:(12.06)

8. [Lotus=FALSE, Marigold=TRUE]: 40 ==> [Orchid=FALSE]: 40 <conf:(1)> lift:(1.43) lev:(0.06) conv:(12.06)

9. [Orchid=FALSE, Marigold=TRUE]: 40 ==> [Lotus=FALSE]: 40 <conf:(1)> lift:(1) lev:(0) conv:(0)

10. [Marigold=TRUE]: 40 ==> [Lotus=FALSE, Iris=FALSE]: 40 <conf:(1)> lift:(1.49) lev:(0.07) conv:(13.07)

**Association Rules have been successfully generated using FP-Growth algorithm using Weka.**