Roll No.	Experiment No. 08	Marks :
BATCH -		Sign :

Aim: Demonstrate Classification algorithm on data sets using data mining tools (WEKA, R tool, XL Miner, Orange etc.)

Apparatus: WEKA

Theory:

What are Classification Algorithms in Data Mining?

The Classification Algorithms in Data Mining is a Supervised Learning method used to determine the group of entirely new observations on the foundation of instruction data. Inside Classification algorithms in data mining, a process learns from the detailed words or dataset and then classifies recent brand mentions into a selection of organizations or classes.

Types of Classification Algorithms in Data Mining

1. **Logistic regression algorithms in data mining:**
   Logistic regression algorithms in data mining are calculations used to predict binary outcome: something happens or doesn't. This is displayed as Yes/No, Alive/Dead, Pass/Fail, etc.
   Impartial variables are examined to figure out the binary effect, with the results falling into one of two types. The impartial variables might be numeric or categorical; however, the impartial variable is categorical. Written like this:
   P(Y=1|X) or even P(Y=0|X) calculates the prospects of reliant adjustable Y, provided impartial adjustable X. This is used to compute the possibility of a term getting a negative or positive atmosphere (0, 1, or on a scale) or it is used to figure out the item found in a picture (tree, etc.), grass, flower, with each object providing a probability between 0 and 1.

2. **Naive Bayes algorithms in data mining:**
   Naive Bayes algorithms in data mining calculate the chance of if a datapoint belongs inside a particular category or doesn't. In-text studies, it used to categorize phrases or words as belonging to a preset "tag" (classification) or not.

3. **K-nearest Neighbors in classification algorithms in data mining**

   K-nearest neighbors (K-NN) in classification algorithms in data mining is a pattern recognition algorithm that uses instruction datasets to find the k closest relations in succeeding examples.

   When K-NN is used in classification algorithms in data mining, you calculate to put details to the group of its nearest neighbor. If k = one, then it will be put into the category to the nearest one. K is classified by a wide variety of forms of its companion.

   KNN in classification algorithms in data mining works on the exact process. It organizes the brand new data points based on the category of the vast majority of data points amongst the K neighbor, in which K is the number of neighbors to be looked at. KNN captures the thought of similarity (sometimes called distance, closeness) or proximity with some fundamental mathematical distance formulas, including Euclidean distance, Manhattan distance, etc.

4. **Decision Tree**

   A decision tree is a supervised learning algorithm that's ideal for classification troubles, as it is in a position to purchase courses on a precise fitness level. It works like a flow chart, sorting out data points into two identical groups at a period from the "tree trunk" to "branches" to "leaves," the place where the categories are finitely comparable. This produces categories within categories, allowing for organic and natural distinction with limited human supervision.

   The Decision tree is just about the most popular machine learning algorithms used. They're used for both classification algorithms in data mining and regression issues. Decision trees do human-level thinking very easily. It is straightforward to understand the data and make some great intuitions & interpretations. They allow you to see the logic for the data to interpret. Decision trees don't like black-box algorithms such as SVM, Neural Networks, etc.

5. **Random Forest**

   The random forest algorithm in classification algorithms in data mining is an enlargement of a decision tree, in which you initially establish a wide range of decision trees with instruction data, then fit the new data of yours within one of the trees as being a "random forest."

   It averages your data of yours to link it to probably the nearest tree on the datascale. Arbitrary forests are very helpful while treating the decision tree's situation of "forcing" data points inside a group unnecessarily.

   Random forest is an operative machine learning algorithm commonly used in Regression and Classification issues. It creates determination trees on various samples and takes the majority vote of theirs for average and classification in case of regression.

DEMONSTRATION (Naive Bayes)

Generated Dataset:

Relation: weka.datagenerators.classifiers.classification.RDG1-S_1_-n_100_-a_10_-c_2_-N_0_-I_0_-M_1_-R_10

No.	1: a0 Nominal	2: a1 Nominal	3: a2 Nominal	4: a3 Nominal	5: a4 Nominal	6: a5 Nominal	7: a6 Nominal	8: a7 Nominal	9: a8 Nominal	10: a9 Nominal	11: **class** Nominal
1	true	false	false	false	true	false	true	true	true	true	c0
2	true	true	false	false	false	true	true	false	true	true	c0
3	false	false	true	true	false	true	false	false	false	false	c1
4	false	true	false	true	true	false	true	true	false	false	c0
5	true	true	false	false	false	true	true	false	false	true	c0
6	true	true	true	true	false	false	true	false	true	true	c1
7	true	true	false	true	true	false	true	false	false	true	c0
8	false	false	true	false	true	true	false	true	false	true	c1
9	true	true	false	true	false	true	false	true	true	true	c0
10	false	false	true	true	true	true	true	true	false	true	c1
11	false	true	false	false	false	true	true	false	true	false	c0
12	true	false	true	false	true	false	false	false	false	true	c0
13	true	true	false	true	true	true	false	false	false	false	c1
14	false	false	false	true	true	true	false	false	true	false	c1
15	false	false	false	true	true	false	false	false	true	false	c0
16	true	false	false	true	false	false	true	false	false	false	c0
17	true	true	true	false	false	true	true	true	false	true	c1

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: weka.datagenerators.classifiers.classification.RDG1-S... Attributes: 11
Instances: 100 Sum of weights: 100

Attributes
All | None | Invert | Pattern

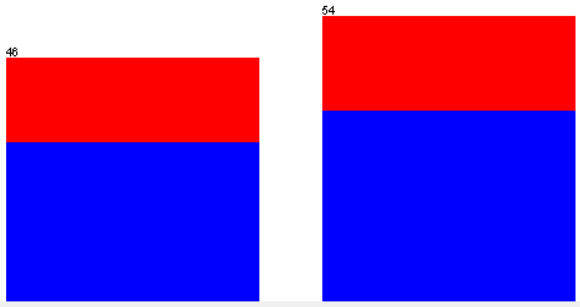
No.	Name
1	a0
2	a1
3	a2
4	a3
5	a4
6	a5
7	a6
8	a7
9	a8
10	a9
11	class

Remove

Selected attribute
Name: a0 Type: Nominal
Missing: 0 (0%) Distinct: 2 Unique: 0 (0%)

No.	Label	Count	Weight
1	false	46	46
2	true	54	54

Class: class (Nom) | Visualize All



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

Classifier
Choose NaiveBayes

Test options
- Use training set
- Supplied test set — Set...
- Cross-validation — Folds 10
- Percentage split — % 66

More options...

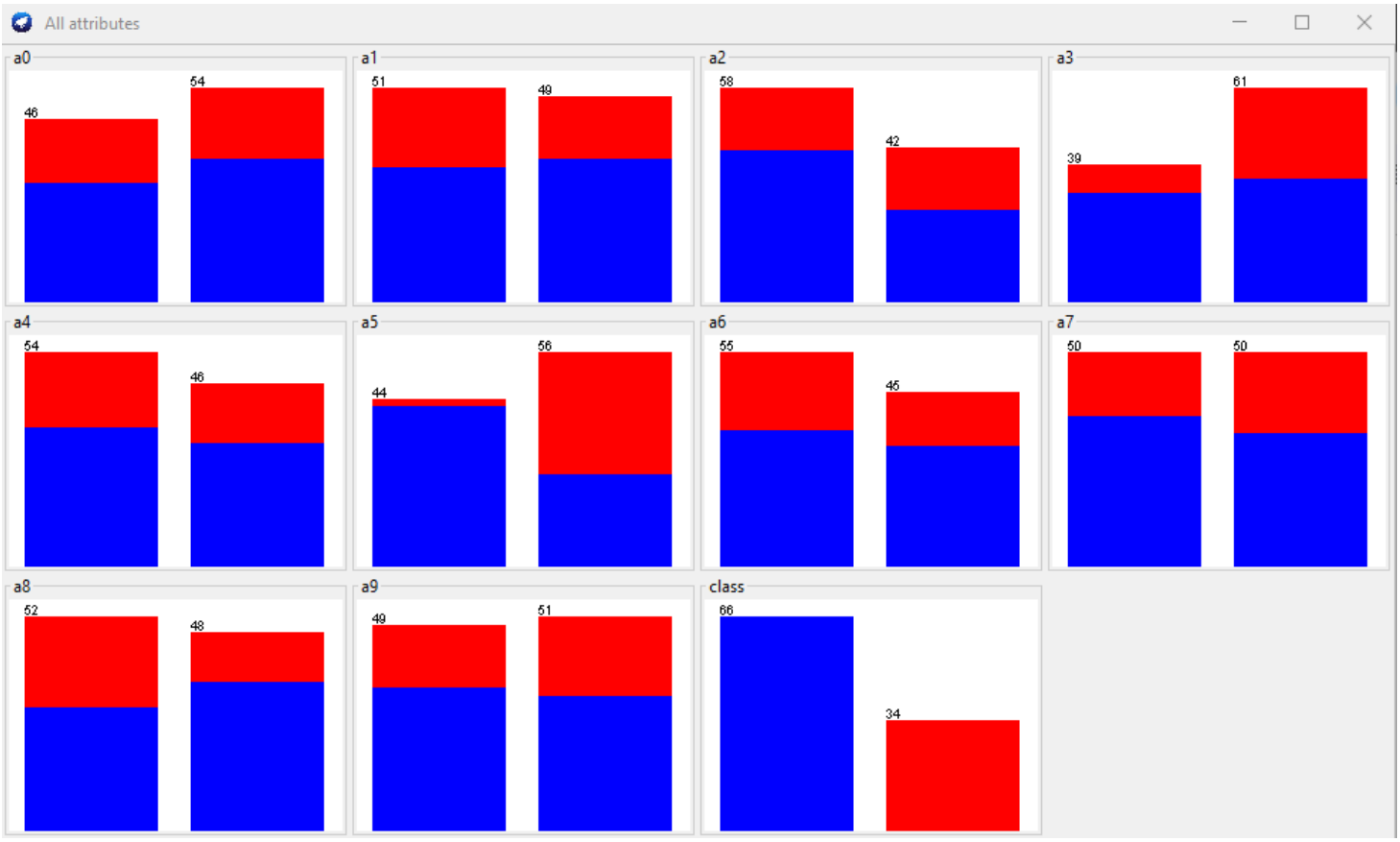
(Nom) class

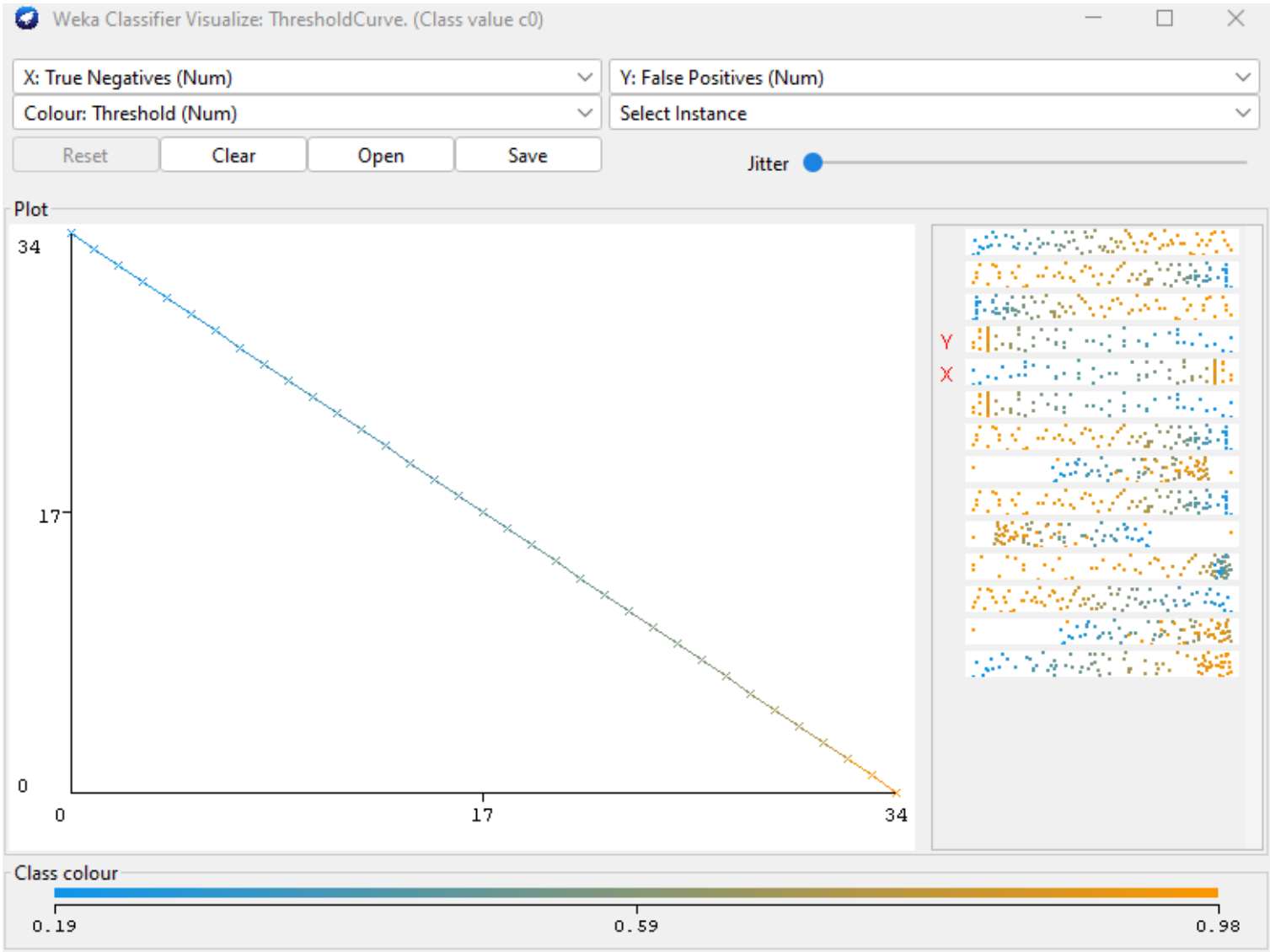
Start | Stop

Result list (right-click for options)
07:26:20 - bayes.NaiveBayes
07:28:55 - bayes.NaiveBayes

Classifier output

```
  true              33.0    20.0
  [total]           68.0    36.0



Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          77               77      %
Incorrectly Classified Instances        23               23      %
Kappa statistic                          0.4838
Mean absolute error                      0.3108
Root mean squared error                  0.3921
Relative absolute error                 68.9937 %
Root relative squared error             82.6609 %
Total Number of Instances              100

=== Detailed Accuracy By Class ===

                 TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                 0.833     0.353     0.821      0.833    0.827      0.484    0.820     0.870     c0
                 0.647     0.167     0.667      0.647    0.657      0.484    0.820     0.660     c1
Weighted Avg.    0.770     0.290     0.768      0.770    0.769      0.484    0.820     0.799

=== Confusion Matrix ===

  a  b   <-- classified as
 55 11 |  a = c0
 12 22 |  b = c1
```


All attributes
a0
46
54
a1
51
49
a2
58
42
a3
39
61
a4
54
46
a5
44
56
a6
55
45
a7
50
50
a8
52
48
a9
49
51
class
66
34



Conclusion:

In conclusion, Weka stands as a powerful open-source data mining tool, offering a user-friendly interface, a rich set of algorithms, and extensive support for preprocessing and experimentation. Its versatility makes it a valuable asset for both beginners and experts in the fields of machine learning and data analysis. Moreover, Weka provides a wide range of classification algorithms, including decision trees, support vector machines, naive Bayes, and many others, empowering users to effectively tackle classification tasks across various domains.