

EXPERIMENT NO 7

NAME: Vishal Shashikant Salvi

UID: 2019230069

Class: TE Comps

BATCH:C

Aim: To implement Classification and clustering Algorithm by using Weka tool.

Theory:

What is Classification?

We use the training dataset to get better boundary conditions which could be used to determine each target class. Once the boundary conditions are determined, the next task is to predict the target class. The whole process is known as classification.

Target class examples:

- Analysis of the customer data to predict whether he will buy computer accessories (**Target class: Yes or No**)
- Classifying fruits from features like color, taste, size, weight (**Target classes: Apple, Orange, Cherry, Banana**)
- Gender classification from hair length (**Target classes: Male or Female**)

Let's understand the concept of classification algorithms with gender classification using hair length (by no means am I trying to stereotype by gender, this is only an example). To classify gender (**target class**) using hair length as feature parameter we could train a model using any classification algorithms to come up with some set of boundary conditions which can be used to differentiate the male and female genders using hair length as the training feature. In gender classification case the boundary condition could be the proper hair length value. Suppose the **differentiated boundary** hair length value is 15.0 cm then we can say that if hair length is **less than 15.0 cm** then gender could be male or else female.

Classification Algorithms vs Clustering Algorithms

In clustering, the idea is not to predict the target class as in classification, it's more ever trying to group the similar kind of things by considering the most satisfied condition, **all the items in the same group should be similar and no two different group items should not be similar.**

Group items Examples:

- While grouping similar language type documents (**Same language documents are one group.**)
- While categorizing the news articles (**Same news category(Sport) articles are one group**)

Let's understand the concept with clustering genders based on hair length example. To determine gender, different similarity measure could be used to categorize male and female genders. This could be done by finding the similarity between two hair lengths and keep them in the same group if the similarity is less (**Difference of hair length is less**). The same process could continue until all the hair length properly grouped into two categories.

Basic Terminology in Classification Algorithms

- **Classifier:** An algorithm that maps the input data to a specific category.
- **Classification model:** A classification model tries to draw some conclusion from the input values given for training. It will predict the class labels/categories for the new data.
- **Feature:** A feature is an individual measurable property of a phenomenon being observed.
- **Binary Classification:** Classification task with two possible outcomes. **Eg: Gender classification (Male / Female)**
- **Multi-class classification:** Classification with more than two classes. In multi-class classification, each sample is assigned to one and only one target label. **Eg: An animal can be a cat or dog but not both at the same time.**
- **Multi-label classification:** Classification task where each sample is mapped to a set of target labels (more than one class). **Eg: A news article can be about sports, a person, and location at the same time.**

Applications of Classification Algorithms

- Email spam classification
- Bank customers loan pay willingness prediction.
- Cancer tumor cells identification.
- Sentiment analysis
- Drugs classification
- Facial key points detection
- Pedestrians detection in an automotive car driving.

Types of Classification Algorithms

Classification Algorithms could be broadly classified as the following:

- **Linear Classifiers**
 - Logistic regression
 - Naive Bayes classifier
 - Fisher's linear discriminant
- **Support vector machines**
 - Least squares support vector machines
- **Quadratic classifiers**
- **Kernel estimation**
 - k-nearest neighbor
- **Decision trees**
 - Random forests

- Neural networks
- Learning vector quantization

Naive Bayes Classifier

This is a classification technique based on an assumption of independence between predictors or what's known as Bayes' theorem. In simple terms, a **Naive Bayes classifier** assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, a Naive Bayes Classifier would consider all of these properties to independently contribute to the probability that this fruit is an apple.

To build a Bayesian model is simple and particularly functional in case of enormous data sets. Along with simplicity, Naive Bayes is known to outperform sophisticated classification methods as well.

Bayes theorem provides a way of calculating posterior probability $P(c|x)$ from $P(c)$, $P(x)$ and $P(x|c)$. The expression for Posterior Probability is as follows.

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

$$P(c | X) = P(x_1 | c) \times P(x_2 | c) \times \dots \times P(x_n | c) \times P(c)$$

Here,

- $P(c|x)$ is the posterior probability of class (target) given predictor (attribute).
- $P(c)$ is the prior probability of class.
- $P(x|c)$ is the likelihood which is the probability of predictor given class.
- $P(x)$ is the prior probability of predictor.

Why is it called Naïve Bayes?

The Naïve Bayes algorithm is comprised of two words Naïve and Bayes, Which can be described as:

- **Naïve:** It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the bases of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other.
- **Bayes:** It is called Bayes because it depends on the principle of Bayes' Theorem

Bayes' Theorem:

- Bayes' theorem is also known as **Bayes' Rule** or **Bayes' law**, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.
- The formula for Bayes' theorem is given as:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Where,

P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.

P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

P(A) is Prior Probability: Probability of hypothesis before observing the evidence.

P(B) is Marginal Probability: Probability of Evidence.

Working of Naïve Bayes' Classifier:

Working of Naïve Bayes' Classifier can be understood with the help of the below example:

Suppose we have a dataset of **weather conditions** and corresponding target variable "**Play**". So using this dataset we need to decide that whether we should play or not on a particular day according to the weather conditions. So to solve this problem, we need to follow the below steps:

1. Convert the given dataset into frequency tables.
2. Generate Likelihood table by finding the probabilities of given features.
3. Now, use Bayes theorem to calculate the posterior probability.

Example: Let's work through an example to understand this better. So, here I have a training data set of weather namely, sunny, overcast and rainy, and corresponding binary variable 'Play'. Now, we need to classify whether players will play or not based on weather condition. Let's follow the below steps to perform it.

Step 1: Convert the data set to the frequency table

Step 2: Create a Likelihood table by finding the probabilities like

Overcast probability = 0.29 and probability of playing is 0.64.

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Likelihood table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
All	5	9
	=5/14	=9/14
	0.36	0.64

			=4/14	0.29
			=5/14	0.36
			=5/14	0.36

Step 3: Now, use the Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.

Problem: Players will play if the weather is sunny, is this statement correct?

We can solve it using above discussed method,

$$\text{so } P(\text{Yes} \mid \text{Sunny}) = P(\text{Sunny} \mid \text{Yes}) * P(\text{Yes}) / P(\text{Sunny})$$

Here we have $P(\text{Sunny} \mid \text{Yes}) = 3/9 = 0.33$, $P(\text{Sunny}) = 5/14 = 0.36$, $P(\text{Yes}) = 9/14 = 0.64$

Now, $P(\text{Yes} \mid \text{Sunny}) = 0.33 * 0.64 / 0.36 = 0.60$, which has higher probability.

Naive Bayes uses a similar method to predict the probability of different class based on various attributes. This algorithm is mostly used in text classification and with problems having multiple classes.

Advantages of Naïve Bayes Classifier:

- Naïve Bayes is one of the fast and easy ML algorithms to predict a class of datasets.
- It can be used for Binary as well as Multi-class Classifications.
- It performs well in Multi-class predictions as compared to the other Algorithms.
- It is the most popular choice for **text classification problems**.

Disadvantages of Naïve Bayes Classifier:

- Naive Bayes assumes that all features are independent or unrelated, so it cannot learn the relationship between features.

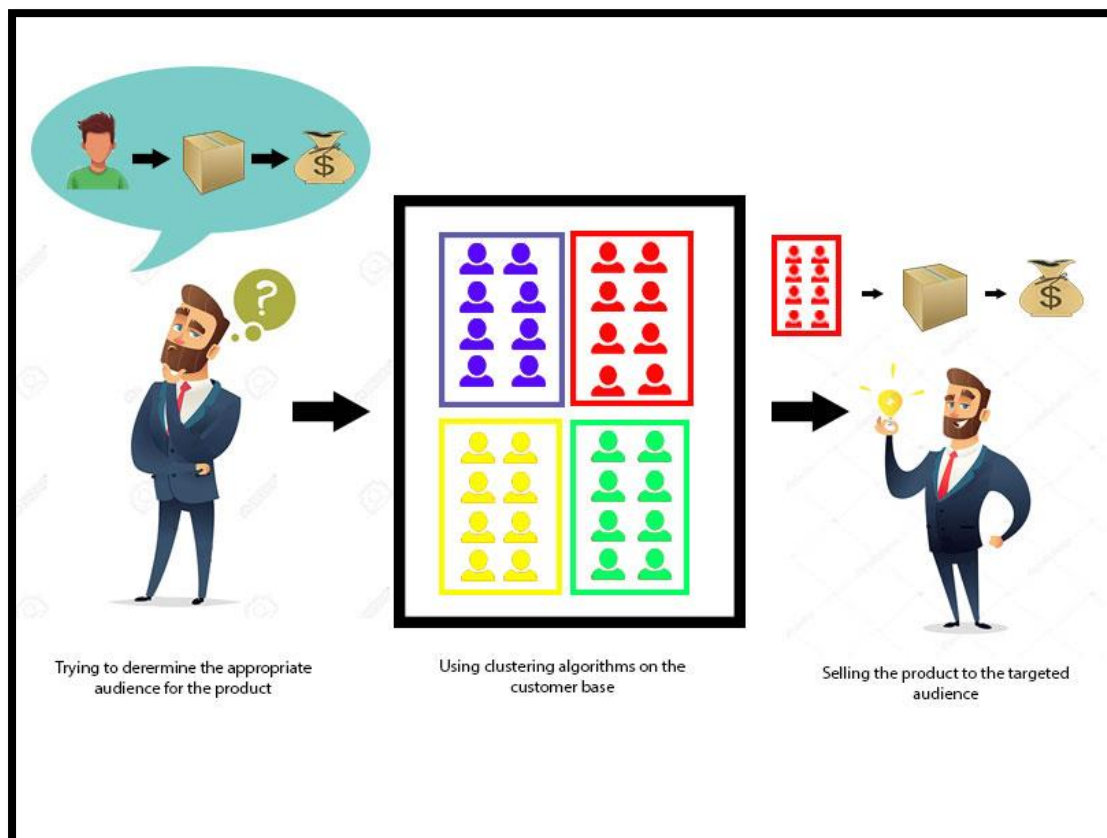
Applications of Naïve Bayes Classifier:

- It is used for **Credit Scoring**.
- It is used in **medical data classification**.
- It can be used in **real-time predictions** because Naïve Bayes Classifier is an eager learner.
- It is used in Text classification such as **Spam filtering** and **Sentiment analysis**.

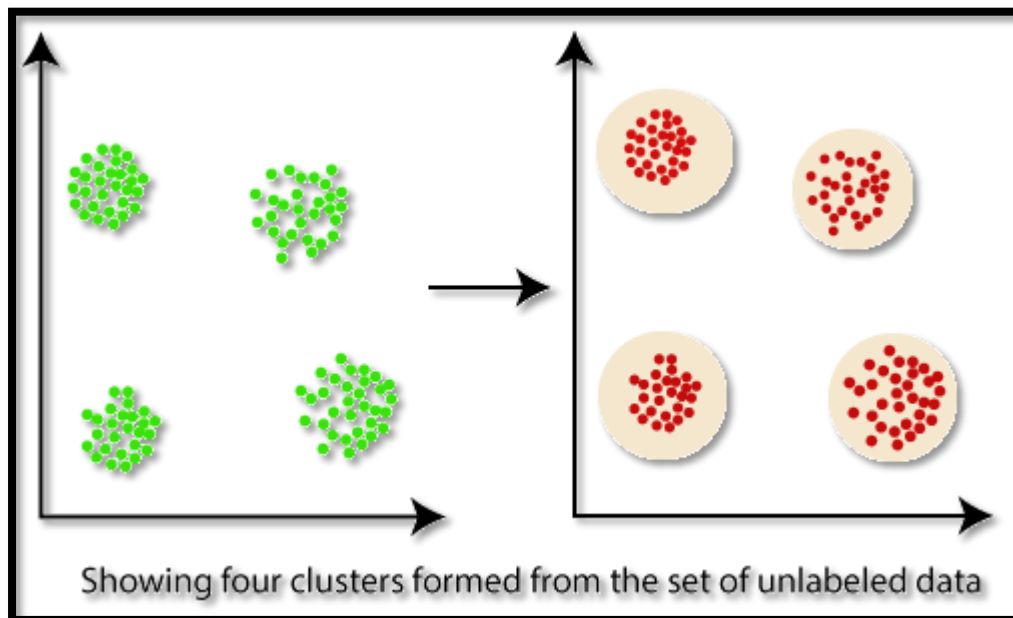
Clustering in Data Mining

Clustering is an unsupervised Machine Learning-based Algorithm that comprises a group of data points into clusters so that the objects belong to the same group.

Clustering helps to splits data into several subsets. Each of these subsets contains data similar to each other, and these subsets are called clusters. Now that the data from our customer base is divided into clusters, we can make an informed decision about who we think is best suited for this product.



Let's understand this with an example, suppose we are a market manager, and we have a new tempting product to sell. We are sure that the product would bring enormous profit, as long as it is sold to the right people. So, how can we tell who is best suited for the product from our company's huge customer base?



Clustering, falling under the category of **unsupervised machine learning**, is one of the problems that machine learning algorithms solve.

Clustering only utilizes input data, to determine patterns, anomalies, or similarities in its input data.

A good clustering algorithm aims to obtain clusters whose:

- The intra-cluster similarities are high, It implies that the data present inside the cluster is similar to one another.
- The inter-cluster similarity is low, and it means each cluster holds data that is not similar to other data.

What is a Cluster?

- A cluster is a subset of similar objects
- A subset of objects such that the distance between any of the two objects in the cluster is less than the distance between any object in the cluster and any object that is not located inside it.
- A connected region of a multidimensional space with a comparatively high density of objects.

What is clustering in Data Mining?

- Clustering is the method of converting a group of abstract objects into classes of similar objects.
- Clustering is a method of partitioning a set of data or objects into a set of significant subclasses called clusters.
- It helps users to understand the structure or natural grouping in a data set and used either as a stand-alone instrument to get a better insight into data distribution or as a pre-processing step for other algorithms

Important points:

- Data objects of a cluster can be considered as one group.
- We first partition the information set into groups while doing cluster analysis. It is based on data similarities and then assigns the levels to the groups.
- The over-classification main advantage is that it is adaptable to modifications, and it helps single out important characteristics that differentiate between distinct groups.

Applications of cluster analysis in data mining:

- In many applications, clustering analysis is widely used, such as data analysis, market research, pattern recognition, and image processing.
- It assists marketers to find different groups in their client base and based on the purchasing patterns. They can characterize their customer groups.
- It helps in allocating documents on the internet for data discovery.
- Clustering is also used in tracking applications such as detection of credit card fraud.
- As a data mining function, cluster analysis serves as a tool to gain insight into the distribution of data to analyze the characteristics of each cluster.
- In terms of biology, It can be used to determine plant and animal taxonomies, categorization of genes with the same functionalities and gain insight into structure inherent to populations.
- It helps in the identification of areas of similar land that are used in an earth observation database and the identification of house groups in a city according to house type, value, and geographical location.

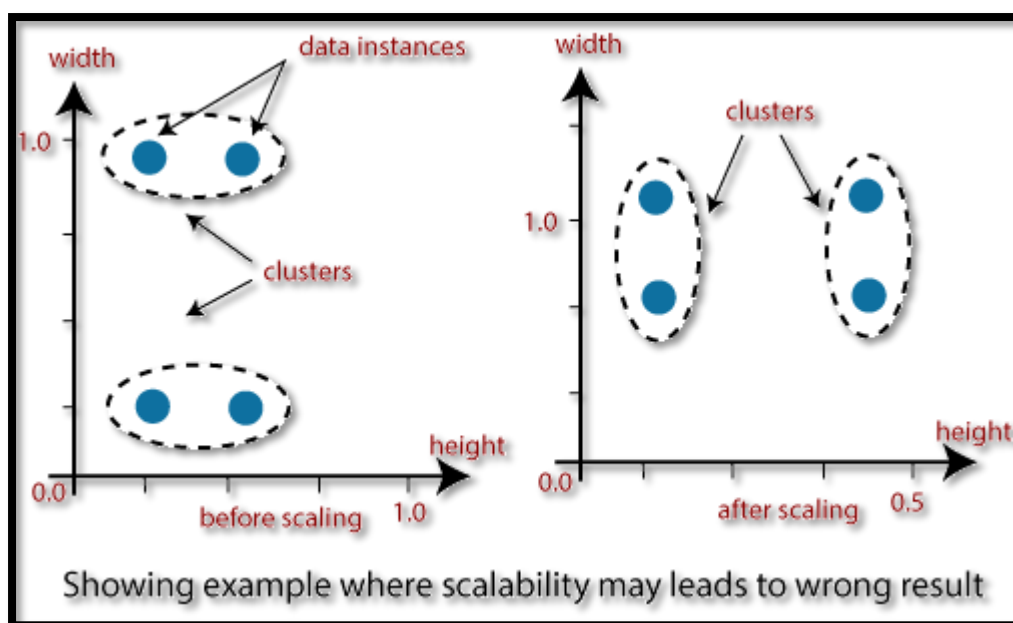
Why is clustering used in data mining?

Clustering analysis has been an evolving problem in data mining due to its variety of applications. The advent of various data clustering tools in the last few years and their comprehensive use in a broad range of applications, including image processing, computational biology, mobile communication, medicine, and economics, must contribute to the popularity of these algorithms. The main issue with the data clustering algorithms is that it can't be standardized. The advanced algorithm may give the best results with one type of data set, but it may fail or perform poorly with other kinds of data set. Although many efforts have been

made to standardize the algorithms that can perform well in all situations, no significant achievement has been achieved so far. Many clustering tools have been proposed so far. However, each algorithm has its advantages or disadvantages and can't work on all real situations.

1. Scalability:

Scalability in clustering implies that as we boost the amount of data objects, the time to perform clustering should approximately scale to the complexity order of the algorithm. For example, if we perform K-means clustering, we know it is $O(n)$, where n is the number of objects in the data. If we raise the number of data objects 10 folds, then the time taken to cluster them should also approximately increase 10 times. It means there should be a linear relationship. If that is not the case, then there is some error with our implementation process.



Data should be scalable if it is not scalable, then we can't get the appropriate result. The figure illustrates the graphical example where it may lead to the wrong result.

2. Interpretability:

The outcomes of clustering should be interpretable, comprehensible, and usable.

3. Discovery of clusters with attribute shape:

The clustering algorithm should be able to find arbitrary shape clusters. They should not be limited to only distance measurements that tend to discover a spherical cluster of small sizes.

4. Ability to deal with different types of attributes:

Algorithms should be capable of being applied to any data such as data based on intervals (numeric), binary data, and categorical data.

5. Ability to deal with noisy data:

Databases contain data that is noisy, missing, or incorrect. Few algorithms are sensitive to such data and may result in poor quality clusters.

6. High dimensionality:

The clustering tools should not only be able to handle high dimensional data space but also the low-dimensional space.

What is K-means Clustering?

K-means (Macqueen, 1967) is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. K-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining.

K-means Clustering – Example 1:

A pizza chain wants to open its delivery centres across a city. What do you think would be the possible challenges?

- They need to analyse the areas from where the pizza is being ordered frequently.
- They need to understand as to how many pizza stores have to be opened to cover delivery in the area.
- They need to figure out the locations for the pizza stores within all these areas in order to keep the distance between the store and delivery points minimum.

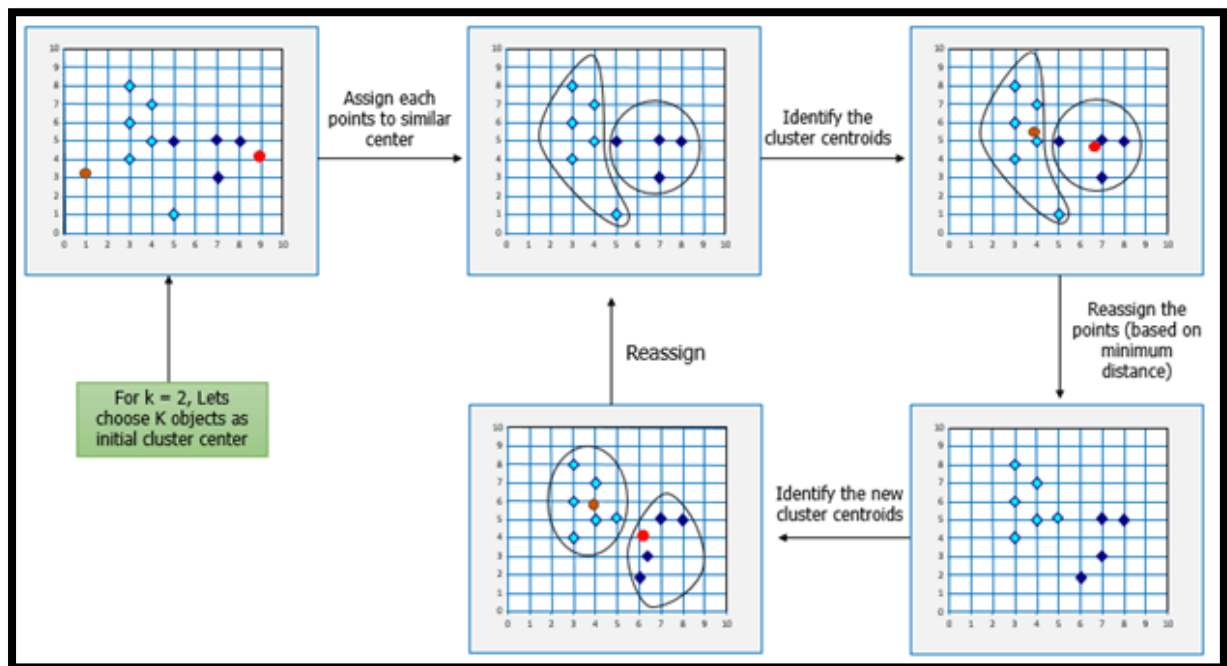
Resolving these challenges includes a lot of analysis and mathematics. We would now learn about how clustering can provide a meaningful and easy method of sorting out such real life challenges. Before that let's see what clustering is.

K-means Clustering Method:

If k is given, the K-means algorithm can be executed in the following steps:

- Partition of objects into k non-empty subsets
- Identifying the cluster centroids (mean point) of the current partition.
- Assigning each point to a specific cluster
- Compute the distances from each point and allot points to the cluster where the distance from the centroid is minimum.
- After re-allotting the points, find the centroid of the new cluster formed.

The step by step process:



Now, let's consider the problem in Example 1 and see how we can help the pizza chain to come up with centres based on K-means algorithm.

Here is another example for you, try and come up with the solution based on your understanding of K-means clustering.

K-means Clustering – Example 2:

Let's consider the data on drug-related crimes in Canada. The data consists of crimes due to various drugs that include, Heroin, Cocaine to prescription drugs, especially by underage people. The crimes resulted due to these substance abuse can be brought down by starting de-addiction centres in areas most afflicted by this kind of crime. With the available data, different objectives can be set. They are:

- Classify the crimes based on the abuse substance to detect prominent cause.
- Classify the crimes based on age groups.
- Analyze the data to determine what kinds of de-addiction centre is required.
- Find out how many de-addiction centres need to be setup to reduce drug related crime rate.

The K-means algorithm can be used to determine any of the above scenarios by analyzing the available data.

Following the K-means Clustering method used in the previous example, we can start off with a given k , following by the execution of the K-means algorithm.

Mathematical Formulation for K-means Algorithm:

$D = \{x_1, x_2, \dots, x_i, \dots, x_m\}$ à data set of m records

$x_i = (x_{i1}, x_{i2}, \dots, x_{in})$ à each record is an n -dimensional vector

$$C_j = \text{Cluster}(x_i) = \arg_j \min ||x_i - \mu_j||^2$$

$$\text{Distortion} = \sum_{i=1}^m (x_i - c_i)^2 = \sum_{j=1}^k \sum_{i \in \text{OwnedBy}(\mu_j)} (x_i - \mu_j)^2$$

(within cluster sum of squares)

Finding Cluster Centers that Minimize Distortion:

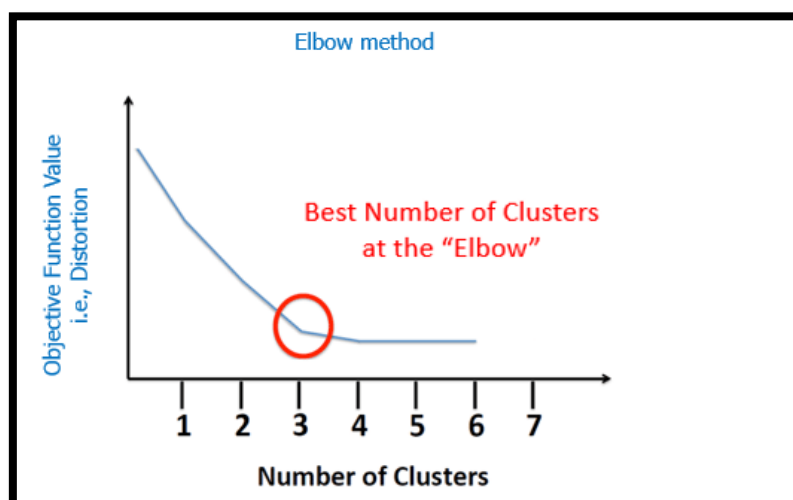
Solution can be found by setting the partial derivative of Distortion w.r.t. each cluster center to zero.

$$\frac{\partial \text{Distortion}}{\partial \mu_j} = \frac{\partial}{\partial \mu_j} \sum_{i \in \text{OwnedBy}(\mu_j)} (x_i - \mu_j)^2 = -2 \sum_{i \in \text{OwnedBy}(\mu_j)} (x_i - \mu_j) = 0 \text{ (for minimum)}$$

$$\Rightarrow \mu_j = \frac{1}{|\text{OwnedBy}(\mu_j)|} \sum_{i \in \text{OwnedBy}(\mu_j)} x_i$$

For any k clusters, the value of k should be such that even if we increase the value of k from after several levels of clustering the distortion remains constant. The achieved point is called the “Elbow”.

This is the ideal value of k , for the clusters created.



Weather.symbolic

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: weather.symbolic
Instances: 14
Attributes: 5
Sum of weights: 14

Selected attribute
Name: outlook
Missing: 0 (0%)
Distinct: 3
Type: Nominal
Unique: 0 (0%)

No.	Label	Count	Weight
1	sunny	5	5.0
2	overcast	4	4.0
3	rainy	5	5.0

Class: play (Nom) Visualize All

5 4 5

Remove

Status
OK Log x 0

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier

Choose J48 -C 0.25 -M 2

Test options

☐ Use training set
☐ Supplied test set Set...
☐ Cross-validation Folds 2
☒ Percentage split % 80
More options...

(Nom) play

Start Stop

Result list (right-click for options)

- 14:32:58 - rules.DecisionTable
- 14:33:52 - rules.DecisionTable
- 14:34:13 - bayes.NaiveBayes
- 14:34:19 - bayes.NaiveBayes
- 14:34:50 - bayes.NaiveBayes
- 14:35:58 - bayes.NaiveBayes
- 14:36:01 - bayes.NaiveBayes
- 14:38:35 - trees.RandomForest
- 14:41:03 - trees.J48

Classifier output

```
=== Evaluation on test split ===  
  
Time taken to test model on test split: 0 seconds  
  
=== Summary ===  
  
Correctly Classified Instances      3      60 %  
Incorrectly Classified Instances    2      40 %  
Kappa statistic                     0  
Mean absolute error                 0.4437  
Root mean squared error             0.5023  
Relative absolute error             93.8582 %  
Root relative squared error         102.2471 %  
Total Number of Instances          5  
  
=== Detailed Accuracy By Class ===  
  
      TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Cl  
1.000    1.000    0.600    1.000    0.750    ?      0.667    0.867    ye  
0.000    0.000    ?        0.000    ?        ?      0.667    0.583    no  
Weighted Avg.  0.600    0.600    ?        0.600    ?        ?      0.667    0.753  
  
=== Confusion Matrix ===  
  
a b  <-- classified as  
3 0 | a = yes  
2 0 | b = no
```

Status
OK Log x 0

Naive Bayes Classifier

Attribute	Class	
	yes	no
	(0.63)	(0.38)
=====		
outlook		
sunny	3.0	4.0
overcast	5.0	1.0
rainy	4.0	3.0
[total]	12.0	8.0
temperature		
hot	3.0	3.0
mild	5.0	3.0
cool	4.0	2.0
[total]	12.0	8.0
humidity		
high	4.0	5.0
normal	7.0	2.0
[total]	11.0	7.0
windy		
TRUE	4.0	4.0
FALSE	7.0	3.0
[total]	11.0	7.0

Weka Explorer

Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Classifier

Choose

J48 -C 0.25 -M 2

Test options

Use training set

Supplied test set

Cross-validation

Percentage split

Folds

%

2

80

Set...

More options...

(Nom) play

Start

Stop

Result list (right-click for options)

14:32:58 - rules.DecisionTable

14:33:52 - rules.DecisionTable

14:34:13 - bayes.NaiveBayes

14:34:19 - bayes.NaiveBayes

14:34:50 - bayes.NaiveBayes

14:35:58 - bayes.NaiveBayes

14:36:01 - bayes.NaiveBayes

14:38:35 - trees.RandomForest

14:41:03 - trees.J48

Classifier output

Time taken to build model: 0 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0 seconds

=== Summary ===

Correctly Classified Instances

Incorrectly Classified Instances

Kappa statistic

Mean absolute error

Root mean squared error

Relative absolute error

Root relative squared error

Total Number of Instances

13

1

0.8372

0.2917

0.3392

62.8233 %

70.7422 %

14

92.8571 %

7.1429 %

=== Detailed Accuracy By Class ===

TP Rate

FP Rate

Precision

Recall

F-Measure

MCC

ROC Area

PRC Area

Cl

1.000

0.200

0.900

1.000

0.947

0.849

0.922

0.947

ye

0.800

0.000

1.000

0.800

0.889

0.849

0.911

0.911

no

Weighted Avg.

0.929

0.129

0.936

0.929

0.926

0.849

0.918

0.934

=== Confusion Matrix ===

Status

OK

Log

x 0

Tree RandomForest

The screenshot shows the Weka Explorer interface with the 'Classify' tab selected. The classifier chosen is 'J48 -C 0.25 -M 2'. The 'Test options' section shows 'Percentage split' at 80%. The 'Classifier output' pane displays the following results:

```
=== Evaluation on test split ===  
Time taken to test model on test split: 0 seconds  
  
=== Summary ===  
Correctly Classified Instances      1      33.3333 %  
Incorrectly Classified Instances    2      66.6667 %  
Kappa statistic                     0  
Mean absolute error                 0.6634  
Root mean squared error             0.6968  
Relative absolute error             117.6106 %  
Root relative squared error         117.6029 %  
Total Number of Instances          3  
  
=== Detailed Accuracy By Class ===  
                TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class  
                1.000    1.000    0.333     1.000    0.500     ?       0.000    0.333    yes  
                0.000    0.000    ?         0.000    ?         ?       0.000    0.583    no  
Weighted Avg.   0.333    0.333    ?         0.333    ?         ?       0.000    0.500  
  
=== Confusion Matrix ===  
a b  <-- classified as  
1 0 | a = yes  
2 0 | b = no
```

The 'Result list' on the left shows several models, with '14:38:35 - trees.RandomForest' selected.

Tree J48

The screenshot shows the Weka Explorer interface with the 'Classify' tab selected. The classifier chosen is 'J48 -C 0.25 -M 2'. The 'Test options' section shows 'Percentage split' at 80%. The 'Classifier output' pane displays the following results:

```
=== Run information ===  
  
Scheme:      weka.classifiers.trees.J48 -C 0.25 -M 2  
Relation:    weather.symbolic  
Instances:   14  
Attributes:  5  
              outlook  
              temperature  
              humidity  
              windy  
              play  
Test mode:   split 80.0% train, remainder test  
  
=== Classifier model (full training set) ===  
  
J48 pruned tree  
-----  
  
outlook = sunny  
| humidity = high: no (3.0)  
| humidity = normal: yes (2.0)  
outlook = overcast: yes (4.0)  
outlook = rainy  
| windy = TRUE: no (2.0)  
| windy = FALSE: yes (3.0)  
  
Number of Leaves :    5  
Size of the tree :    8
```

The 'Result list' on the left shows several models, with '14:41:03 - trees.J48' selected.

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier

Choose J48 -C 0.25 -M 2

Test options

☐ Use training set
☐ Supplied test set Set...
☐ Cross-validation Folds 2
☒ Percentage split % 80
 More options...

(Nom) play

Start Stop

Result list (right-click for options)

- 14:32:58 - rules DecisionTable
- 14:33:52 - rules DecisionTable
- 14:34:13 - bayes.NaiveBayes
- 14:34:19 - bayes.NaiveBayes
- 14:34:50 - bayes.NaiveBayes
- 14:35:58 - bayes.NaiveBayes
- 14:36:01 - bayes.NaiveBayes
- 14:38:35 - trees.RandomForest
- 14:41:03 - trees.J48

Classifier output

```

=== Evaluation on test split ===

Time taken to test model on test split: 0 seconds

=== Summary ===

Correctly Classified Instances      0          0 %
Incorrectly Classified Instances    3          100 %
Kappa statistic                    -0.8
Mean absolute error                 1
Root mean squared error             1
Relative absolute error             177.2727 %
Root relative squared error         168.7695 %
Total Number of Instances          3

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
          0.000    1.000    0.000     0.000    0.000     -1.000   0.000    0.333    yes
          0.000    1.000    0.000     0.000    0.000     -1.000   0.000    0.667    no
Weighted Avg.  0.000    1.000    0.000     0.000    0.000     -1.000   0.000    0.556

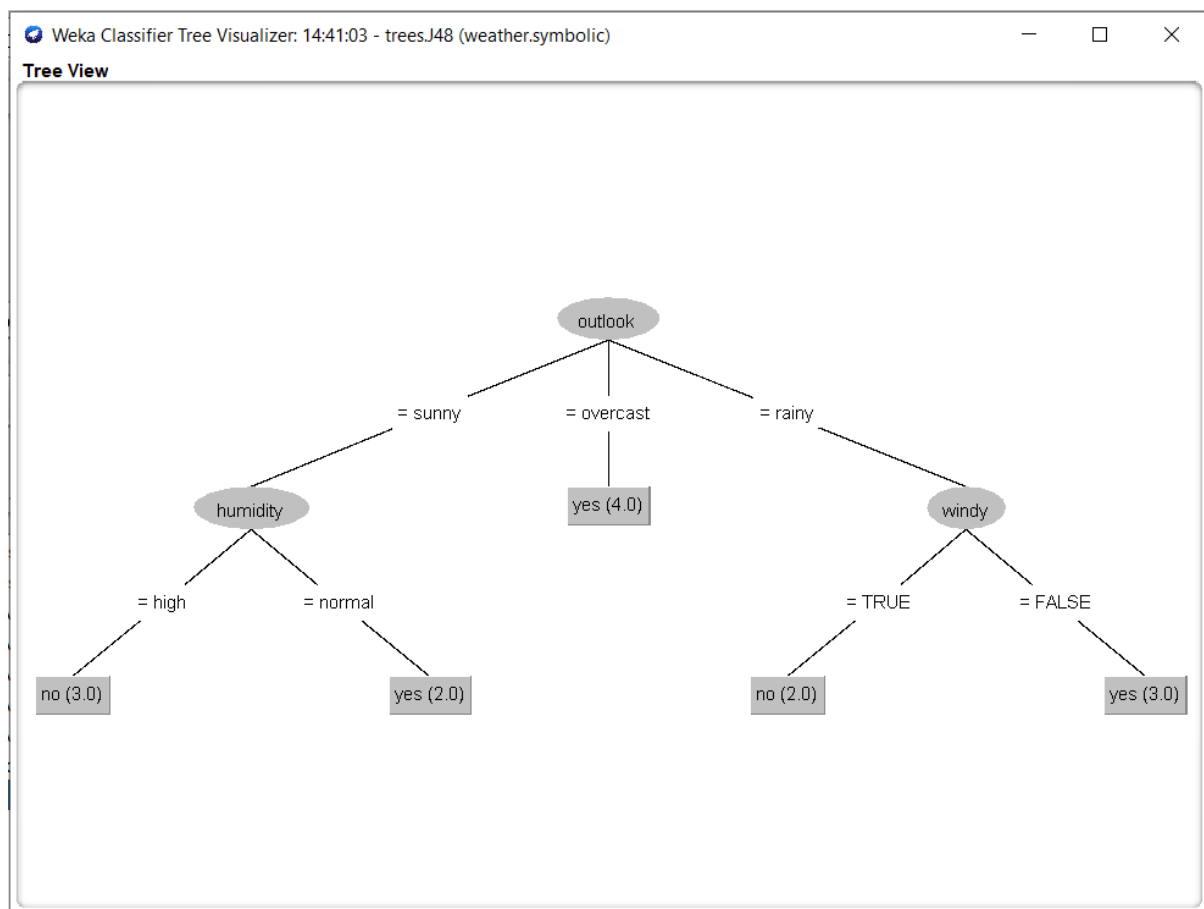
=== Confusion Matrix ===

a b  <-- classified as
0 1 | a = yes
2 0 | b = no
  
```

Status

OK Log x 0

Tree Visualizer



Wether_new.arff

*whether_new - Notepad

File Edit Format View Help

```
@relation weather
@attribute outlook {sunny, overcast, rainy}
@attribute temperature real
@attribute humidity real
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}
```

```
@data
rainy,85,85,FALSE,no
sunny,80,95,TRUE,yes
overcast,83,89,FALSE,yes
rainy,60,96,TRUE,yes
sunny,80,80,FALSE,yes
overcast,65,70,TRUE,no
overcast,64,65,TRUE,yes
sunny,72,95,TRUE,no
sunny,69,70,FALSE,yes
rainy,75,80,FALSE,yes
sunny,80,70,TRUE,yes
rainy,65,90,TRUE,yes
overcast,81,75,FALSE,no
rainy,71,91,TRUE,no
```

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: weather
Instances: 14
Attributes: 5
Sum of weights: 14

Attributes
All None Invert Pattern

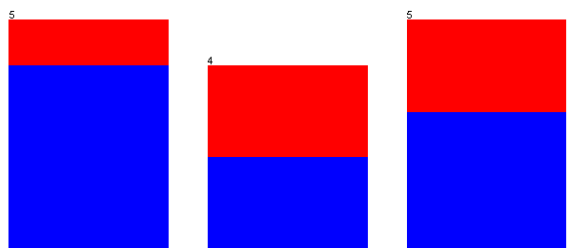
No.	Name
1	<input checked="" type="checkbox"/> outlook
2	<input type="checkbox"/> temperature
3	<input type="checkbox"/> humidity
4	<input type="checkbox"/> windy
5	<input type="checkbox"/> play

Remove

Selected attribute
Name: outlook
Missing: 0 (0%)
Distinct: 3
Type: Nominal
Unique: 0 (0%)

No.	Label	Count	Weight
1	sunny	5	5.0
2	overcast	4	4.0
3	rainy	5	5.0

Class: play (Nom) Visualize All



Status: OK Log x 0

Cross Validation = 10

The screenshot shows the Weka Explorer interface with the 'Classify' tab selected. The 'Classifier' dropdown is set to 'NaiveBayes'. Under 'Test options', 'Cross-validation' is selected with 'Folds' set to 10. The 'Result list' on the left shows two entries: '11:16:21 - bayes.NaiveBayes' and '11:24:47 - bayes.NaiveBayes', with the latter selected. The 'Classifier output' pane displays the following results:

Time taken to build model: 0 seconds

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

Correctly Classified Instances	4	28.5714 %
Incorrectly Classified Instances	10	71.4286 %
Kappa statistic	-0.5556	
Mean absolute error	0.62	
Root mean squared error	0.6659	
Relative absolute error	130.1916 %	
Root relative squared error	134.9773 %	
Total Number of Instances	14	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.444	1.000	0.444	0.444	0.444	0.444	-0.556	0.067	0.481	ye
0.000	0.556	0.000	0.000	0.000	0.000	-0.556	0.067	0.250	no
Weighted Avg.	0.286	0.841	0.286	0.286	0.286	-0.556	0.067	0.398	

=== Confusion Matrix ===

```
a b  <-- classified as
4 5 | a = yes
5 0 | b = no
```

Status: OK

Cross Validation = 5

The screenshot shows the Weka Explorer interface with the 'Classify' tab selected. The 'Classifier' dropdown is set to 'NaiveBayes'. Under 'Test options', 'Cross-validation' is selected with 'Folds' set to 5. The 'Result list' on the left shows two entries: '11:16:21 - bayes.NaiveBayes' and '11:27:33 - bayes.NaiveBayes', with the latter selected. The 'Classifier output' pane displays the following results:

Time taken to build model: 0 seconds

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

Correctly Classified Instances	6	42.8571 %
Incorrectly Classified Instances	8	57.1429 %
Kappa statistic	-0.3659	
Mean absolute error	0.5889	
Root mean squared error	0.6285	
Relative absolute error	126.1016 %	
Root relative squared error	130.6584 %	
Total Number of Instances	14	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.667	1.000	0.545	0.667	0.600	0.600	-0.389	0.089	0.486	ye
0.000	0.333	0.000	0.000	0.000	0.000	-0.389	0.089	0.253	no
Weighted Avg.	0.429	0.762	0.351	0.429	0.386	-0.389	0.089	0.403	

=== Confusion Matrix ===

```
a b  <-- classified as
6 3 | a = yes
5 0 | b = no
```

Status: OK

Split = 66

Weka Explorer

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) play

Start Stop

Result list (right-click for options)

- 11:16:21 - bayes.NaiveBayes
- 11:24:47 - bayes.NaiveBayes
- 11:27:33 - bayes.NaiveBayes
- 11:28:32 - bayes.NaiveBayes
- 11:28:39 - bayes.NaiveBayes
- 11:28:48 - bayes.NaiveBayes
- 11:29:18 - bayes.NaiveBayes

Classifier output

```
=== Evaluation on test split ===  
Time taken to test model on test split: 0 seconds  
  
=== Summary ===  
Correctly Classified Instances      3      60 %  
Incorrectly Classified Instances    2      40 %  
Kappa statistic                    0  
Mean absolute error                 0.4833  
Root mean squared error             0.5277  
Relative absolute error             102.2377 %  
Root relative squared error         107.4256 %  
Total Number of Instances          5  
  
=== Detailed Accuracy By Class ===  


|               | TP Rate | FP Rate | Precision | Recall | F-Measure | MCC | ROC Area | PRC Area | Class |
|---------------|---------|---------|-----------|--------|-----------|-----|----------|----------|-------|
| yes           | 1.000   | 1.000   | 0.600     | 1.000  | 0.750     | ?   | 0.500    | 0.639    | ye    |
| no            | 0.000   | 0.000   | ?         | 0.000  | ?         | ?   | 0.500    | 0.700    | no    |
| Weighted Avg. | 0.600   | 0.600   | ?         | 0.600  | ?         | ?   | 0.500    | 0.663    |       |

  
=== Confusion Matrix ===  
a b  <-- classified as  
3 0 | a = yes  
2 0 | b = no
```

Status

OK Log x 0

Split = 40

Weka Explorer

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 % 40
More options...

(Nom) play

Start Stop

Result list (right-click for options)

- 11:16:21 - bayes.NaiveBayes
- 11:24:47 - bayes.NaiveBayes
- 11:27:33 - bayes.NaiveBayes
- 11:28:32 - bayes.NaiveBayes
- 11:28:39 - bayes.NaiveBayes
- 11:28:48 - bayes.NaiveBayes
- 11:29:18 - bayes.NaiveBayes
- 11:30:34 - bayes.NaiveBayes
- 11:30:39 - bayes.NaiveBayes

Classifier output

```
=== Evaluation on test split ===  
Time taken to test model on test split: 0 seconds  
  
=== Summary ===  
Correctly Classified Instances      4      50 %  
Incorrectly Classified Instances    4      50 %  
Kappa statistic                    0  
Mean absolute error                 0.562  
Root mean squared error             0.5993  
Relative absolute error             112.4077 %  
Root relative squared error         119.8559 %  
Total Number of Instances          8  
  
=== Detailed Accuracy By Class ===  

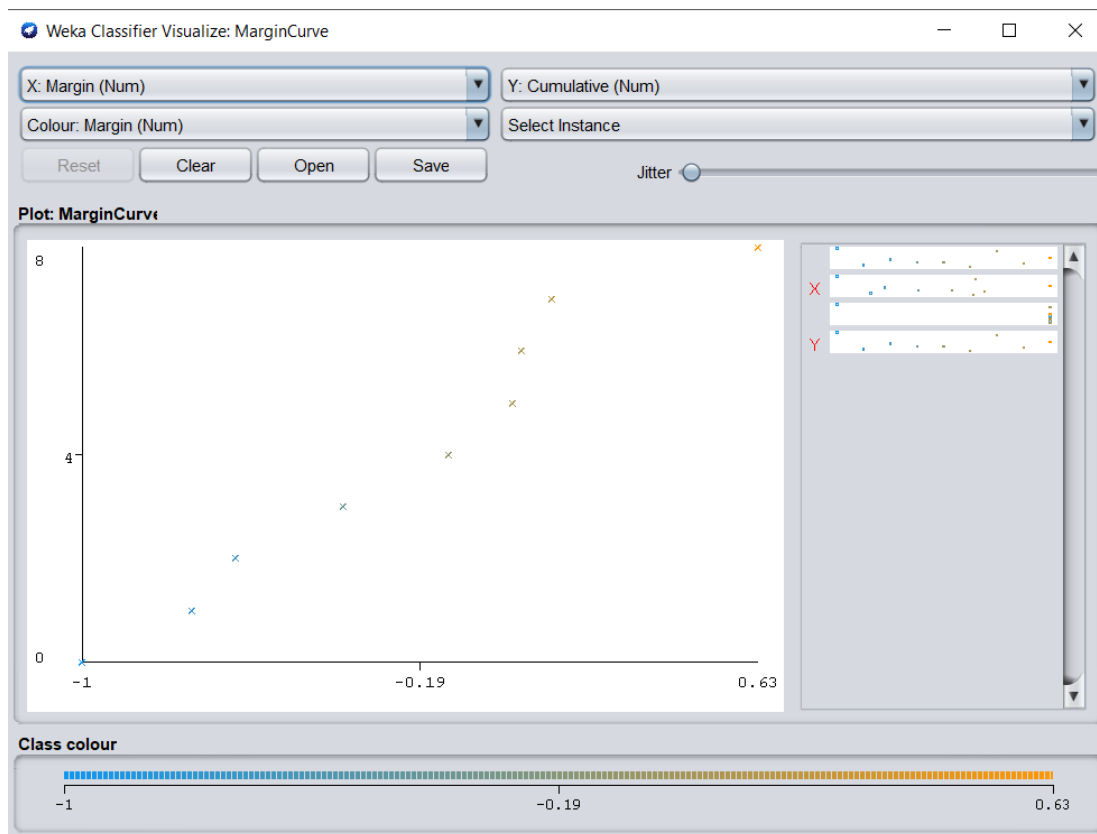

|               | TP Rate | FP Rate | Precision | Recall | F-Measure | MCC   | ROC Area | PRC Area | Class |
|---------------|---------|---------|-----------|--------|-----------|-------|----------|----------|-------|
| yes           | 0.500   | 0.500   | 0.750     | 0.500  | 0.600     | 0.000 | 0.417    | 0.780    | ye    |
| no            | 0.500   | 0.500   | 0.250     | 0.500  | 0.333     | 0.000 | 0.417    | 0.310    | no    |
| Weighted Avg. | 0.500   | 0.500   | 0.625     | 0.500  | 0.533     | 0.000 | 0.417    | 0.663    |       |

  
=== Confusion Matrix ===  
a b  <-- classified as  
3 3 | a = yes  
1 1 | b = no
```

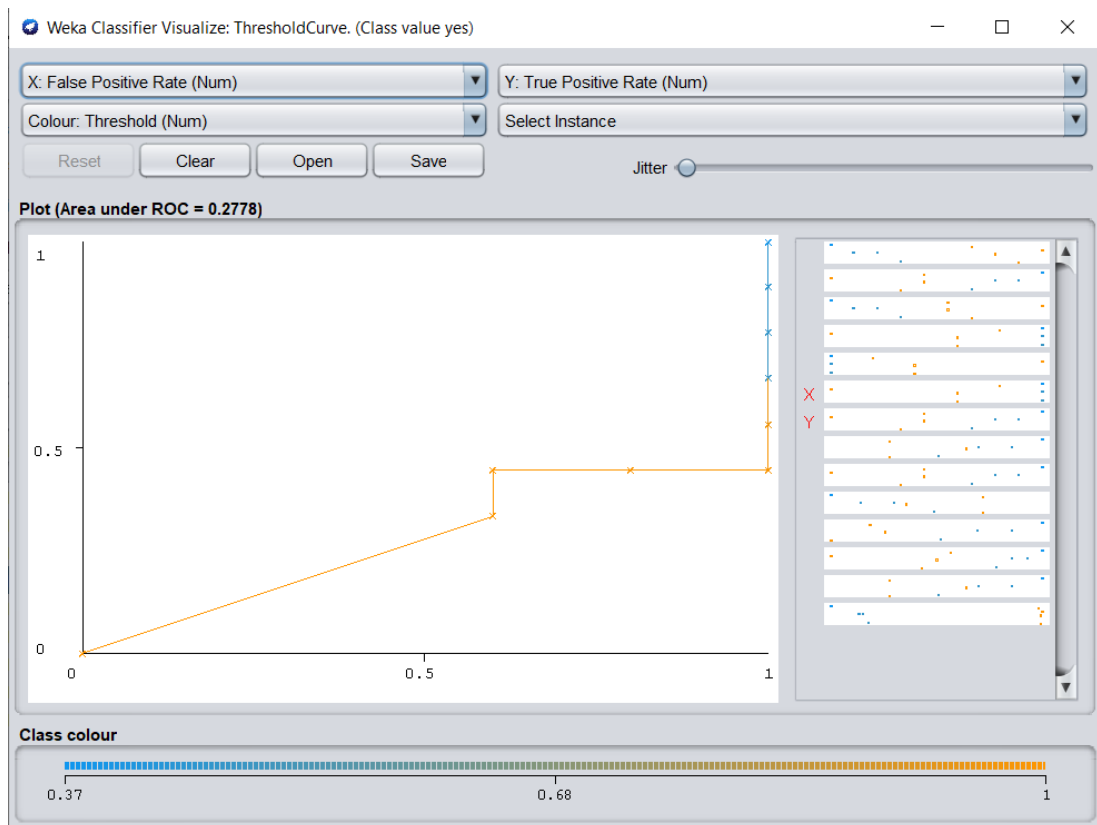
Status

OK Log x 0

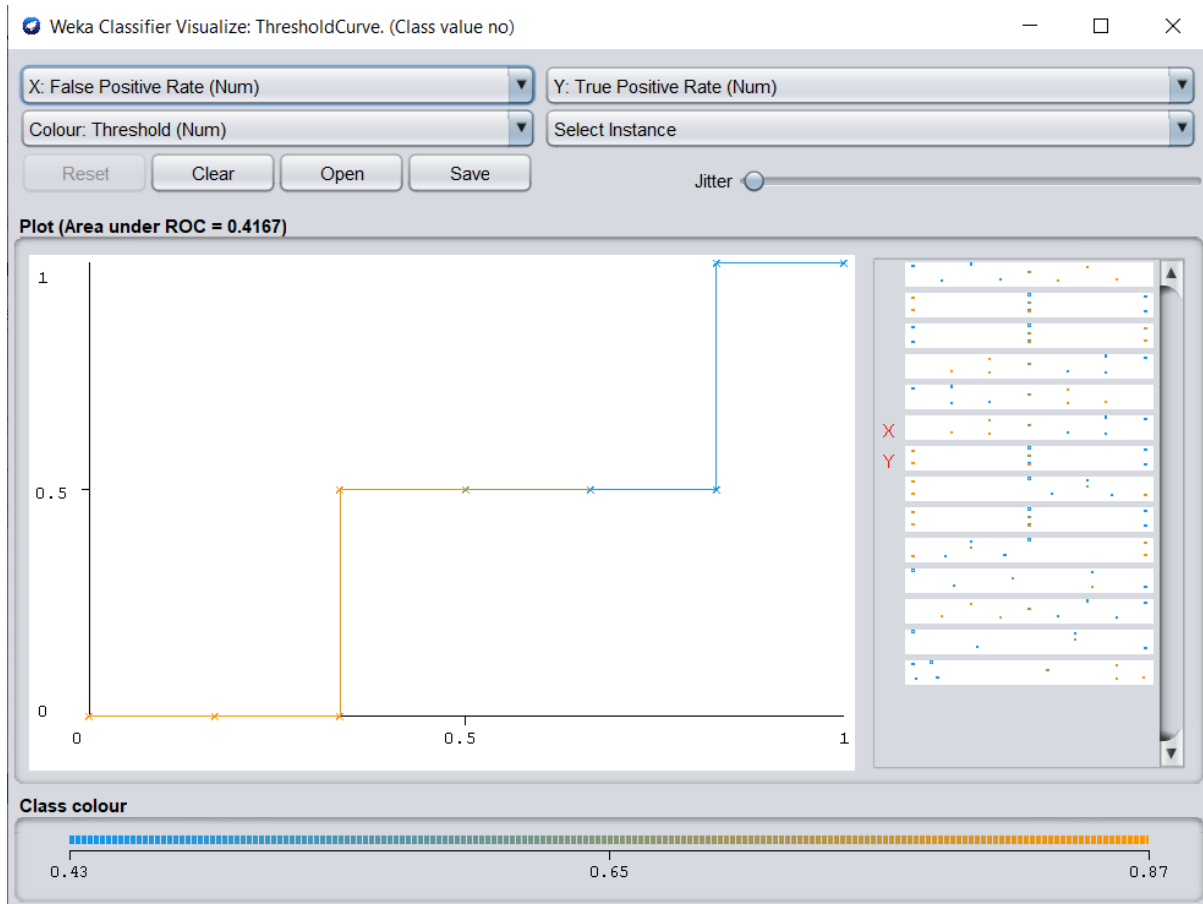
Margin Curve



Threshold Curve Class value yes



Threshold Curve Class Value No



J48 Split = 40

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier

Choose J48 -C 0.25 -M 2

Test options

☐ Use training set

☐ Supplied test set Set...

☐ Cross-validation Folds 10

☒ Percentage split % 40

More options...

(Nom) play

Start Stop

Result list (right-click for options)

- 11:16:21 - bayes.NaiveBayes
- 11:24:47 - bayes.NaiveBayes
- 11:27:33 - bayes.NaiveBayes
- 11:28:32 - bayes.NaiveBayes
- 11:28:39 - bayes.NaiveBayes
- 11:28:48 - bayes.NaiveBayes
- 11:29:18 - bayes.NaiveBayes
- 11:30:34 - bayes.NaiveBayes
- 11:30:39 - bayes.NaiveBayes
- 11:37:54 - trees.J48
- 11:38:07 - trees.J48
- 11:38:14 - trees.J48
- 11:38:22 - trees.J48

Classifier output

```

=== Evaluation on test split ===

Time taken to test model on test split: 0 seconds

=== Summary ===

Correctly Classified Instances      2      25 %
Incorrectly Classified Instances    6      75 %
Kappa statistic                    0
Mean absolute error                0.625
Root mean squared error            0.6614
Relative absolute error            125 %
Root relative squared error        132.2876 %
Total Number of Instances         8

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
      0.000    0.000    ?          0.000    ?          ?      0.500    0.750    yes
      1.000    1.000    0.250     1.000    0.400    ?      0.500    0.250    no
Weighted Avg.   0.250    0.250    ?          0.250    ?          ?      0.500    0.625

=== Confusion Matrix ===

a b  <-- classified as
0 6 | a = yes
0 2 | b = no
    
```

Status

OK Log x 0

Split = 66

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier

Choose J48 - C 0.25 - M 2

Test options

☐ Use training set
☐ Supplied test set Set...
☐ Cross-validation Folds 10
☒ Percentage split % 66
More options...

(Nom) play

Start Stop

Result list (right-click for options)

- 11:30:34 - bayes.NaiveBayes
- 11:30:39 - bayes.NaiveBayes
- 11:37:54 - trees.J48
- 11:38:07 - trees.J48
- 11:38:14 - trees.J48
- 11:38:22 - trees.J48
- 11:39:49 - trees.J48
- 11:39:55 - trees.J48
- 11:39:59 - trees.J48
- 11:40:06 - trees.J48
- 11:40:12 - trees.J48
- 11:40:17 - trees.J48
- 11:40:24 - trees.J48

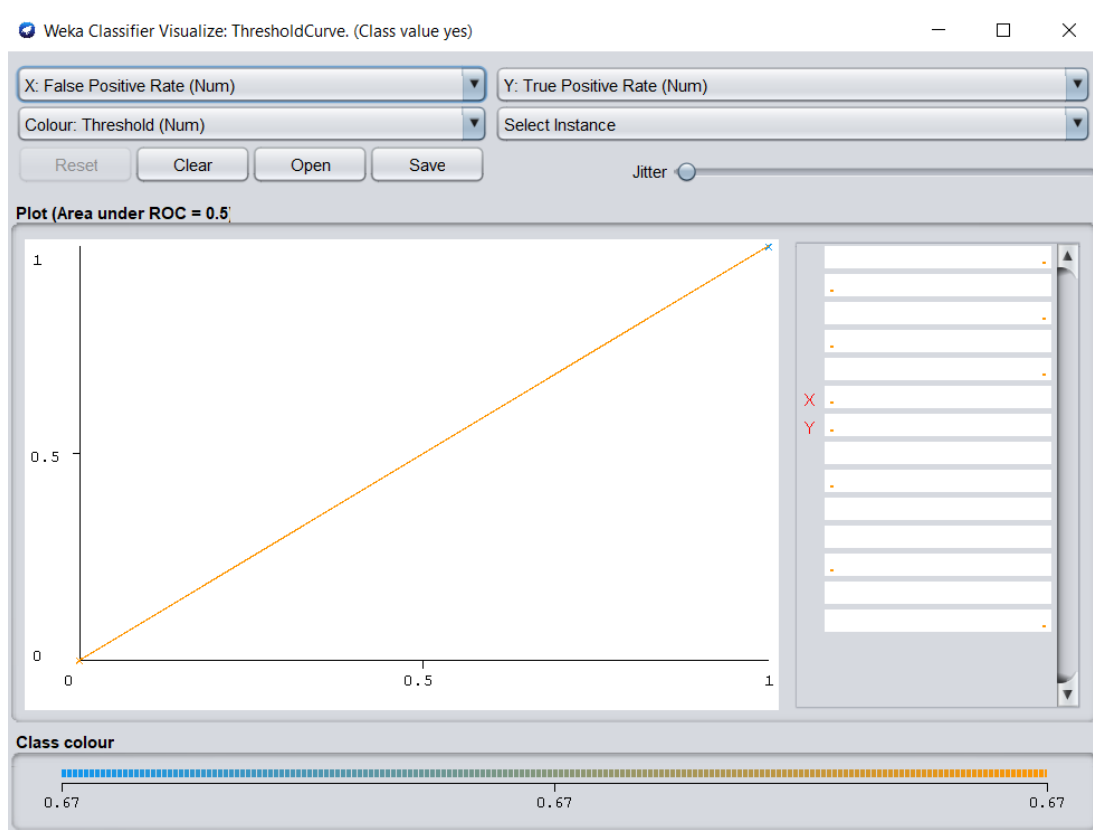
Classifier output

```
=== Evaluation on test split ===  
Time taken to test model on test split: 0 seconds  
=== Summary ===  
Correctly Classified Instances      3      60 %  
Incorrectly Classified Instances    2      40 %  
Kappa statistic                     0  
Mean absolute error                 0.4667  
Root mean squared error             0.4944  
Relative absolute error             98.7179 %  
Root relative squared error         100.6448 %  
Total Number of Instances          5  
  
=== Detailed Accuracy By Class ===  
      TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class  
      1.000    1.000    0.600    1.000    0.750    ?      0.500    0.600    yes  
      0.000    0.000    ?        0.000    ?        ?      0.500    0.400    no  
Weighted Avg.   0.600    0.600    ?        0.600    ?        ?      0.500    0.520  
  
=== Confusion Matrix ===  
a b  <-- classified as  
3 0 | a = yes  
2 0 | b = no
```

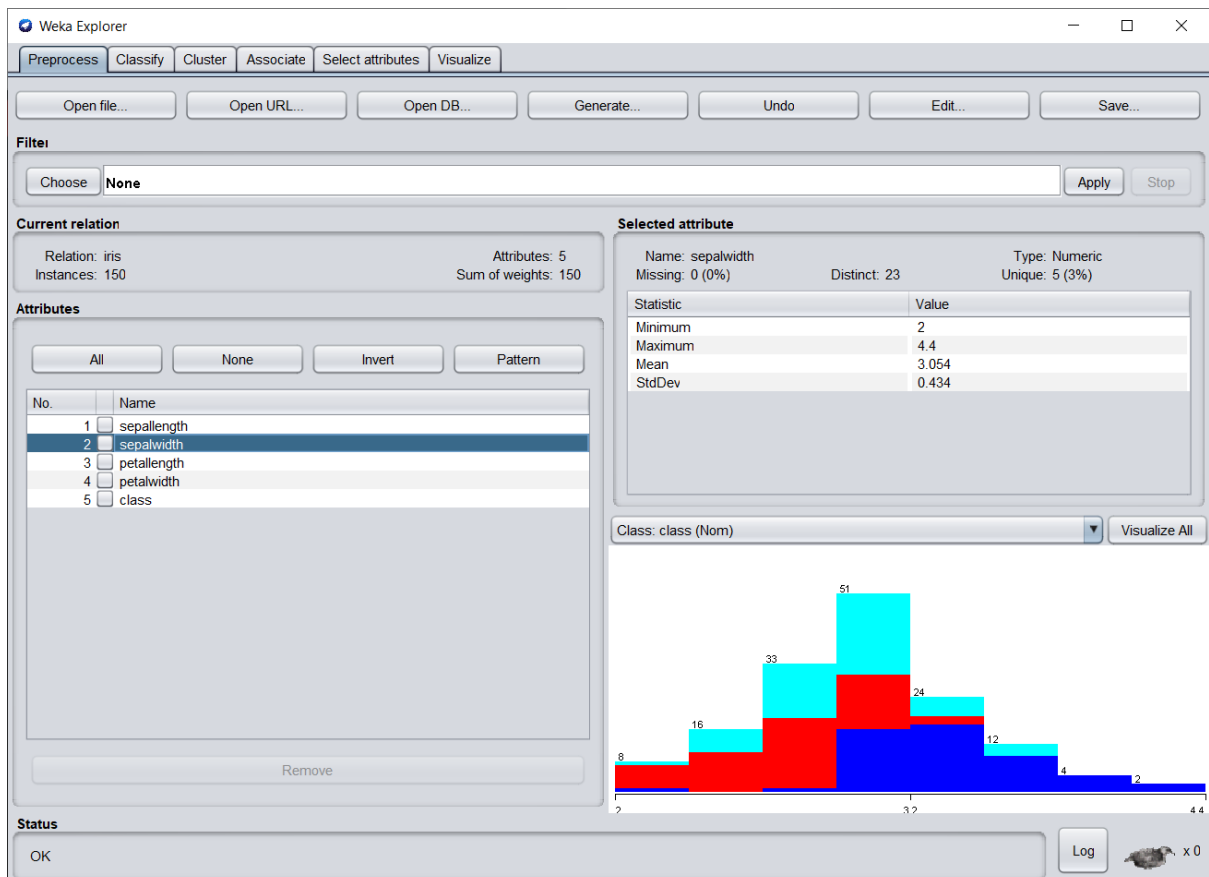
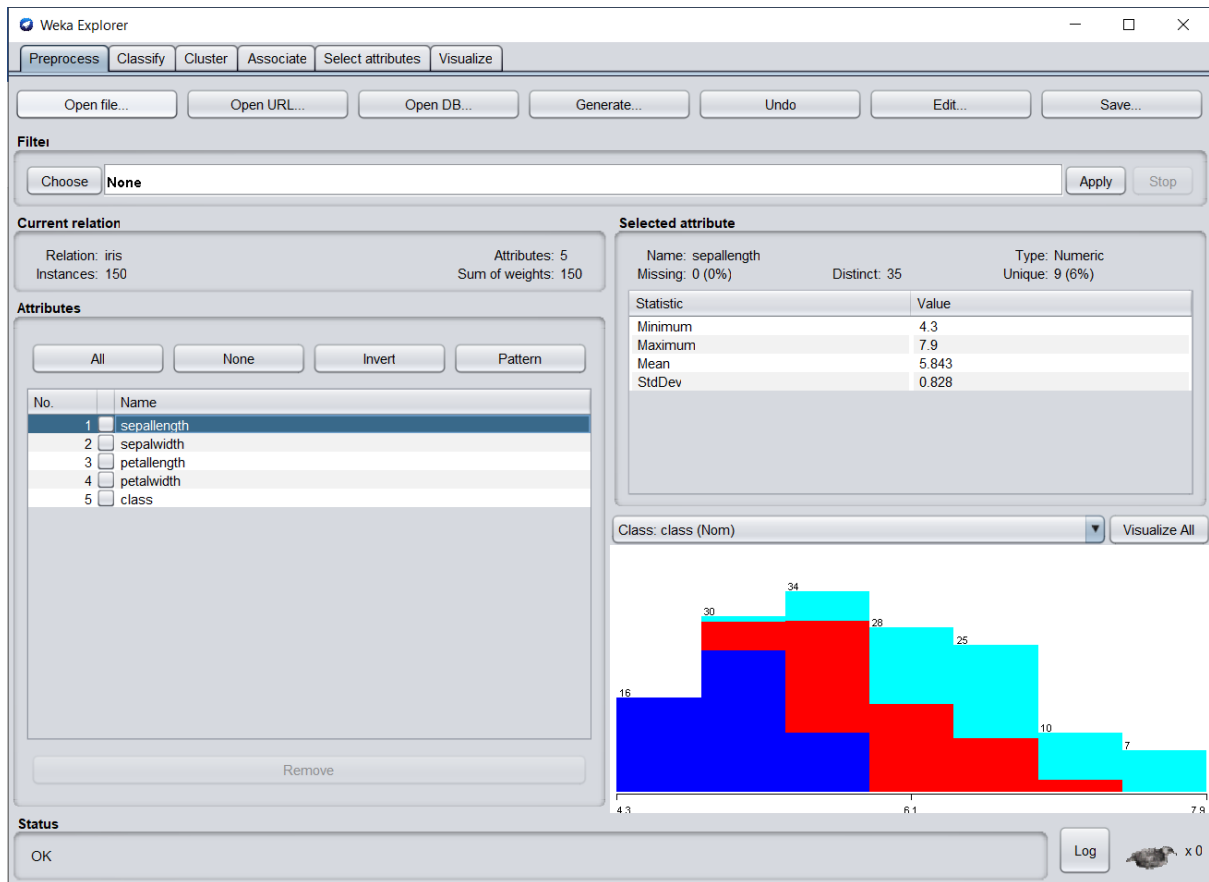
Status

OK Log x 0

Threshold Curve Class Value yes



Iris.arff



Weka Explorer

Preprocess Classify **Cluster** Associate Select attributes Visualize

Clusterer

Choose **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 Set...
☐ Percentage split % 66
☐ Classes to clusters evaluation
 (Nom) class
☒ Store clusters for visualization

Ignore attributes

Start Stop

Result list (right-click for options)

11:47:30 - SimpleKMeans

Clusterer output

```
Cluster 0: 6.1,2.9,4.7,1.4,Iris-versicolor
Cluster 1: 6.2,2.9,4.3,1.3,Iris-versicolor

Missing values globally replaced with mean/mode

Final cluster centroids:

Attribute          Full Data          Cluster#
                   (150.0)          (100.0)          (50.0)
=====
sepalength         5.8433             6.262             5.006
sepalwidth         3.054              2.872             3.418
petallength        3.7587             4.906             1.464
petalwidth         1.1987             1.676             0.244
class              Iris-setosa Iris-versicolor Iris-setosa

Time taken to build model (full training data) : 0.01 seconds

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

Clustered Instances

0      100 ( 67%)
1       50 ( 33%)
```

Status

OK Log x 0

Manhattan Distance

Weka Explorer

Preprocess Classify **Cluster** Associate Select attributes Visualize

Clusterer

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

Cluster mode

☒ Use training set
☐ Supplied test set Set...
☐ Percentage split % 66
☐ Classes to clusters evaluation
 (Nom) class
☒ Store clusters for visualization

Ignore attributes

Start Stop

Result list (right-click for options)

11:58:53 - HierarchicalClusterer
 11:59:48 - HierarchicalClusterer
 11:59:56 - SimpleKMeans
 12:02:18 - SimpleKMeans

Clusterer output

```
Cluster 0: 6.1,2.9,4.7,1.4,Iris-versicolor
Cluster 1: 6.2,2.9,4.3,1.3,Iris-versicolor

Missing values globally replaced with mean/mode

Final cluster centroids:

Attribute          Full Data          Cluster#
                   (150.0)          (98.0)          (52.0)
=====
sepalength         5.8                6.3              5
sepalwidth         3                  2.9              3.4
petallength        4.35               4.9              1.5
petalwidth         1.3                1.6              0.2
class              Iris-setosa Iris-virginica  Iris-setosa

Time taken to build model (full training data) : 0.01 seconds

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

Clustered Instances

0      98 ( 65%)
1      52 ( 35%)
```

Status

OK Log x 0

Euclidean Distance

The screenshot shows the Weka Explorer interface with the 'Cluster' tab selected. The 'Clusterer' dropdown is set to 'SimpleKMeans'. The 'Cluster mode' section has 'Use training set' selected. The 'Clusterer output' pane displays the following information:

```
Cluster 0: 6.1,2.9,4.7,1.4,Iris-versicolor
Cluster 1: 6.2,2.9,4.3,1.3,Iris-versicolor

Missing values globally replaced with mean/mode

Final cluster centroids:

Attribute          Full Data          Cluster#          0          1
                   (150.0)         (100.0)         (50.0)
=====
sepalength          5.8433             6.262            5.006
sepalwidth          3.054              2.872            3.418
petallength         3.7587             4.906            1.464
petalwidth          1.1987             1.676            0.244
class               Iris-setosa Iris-versicolor  Iris-setosa

Time taken to build model (full training data) : 0 seconds

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

Clustered Instances

0      100 ( 67%)
1       50 ( 33%)
```

The 'Result list' on the left shows several entries, with '12:04:20 - SimpleKMeans' selected. The 'Status' bar at the bottom shows 'OK' and a 'Log' button.

Hierarchical Cluster

The screenshot shows the Weka Explorer interface with the 'Cluster' tab selected. The 'Clusterer' dropdown is set to 'HierarchicalClusterer'. The 'Cluster mode' section has 'Use training set' selected. The 'Clusterer output' pane displays the following information:

```
Attributes: 5
            sepalength
            sepalwidth
            petallength
            petalwidth
            class
Test mode:  evaluate on training data

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

Cluster 0
((((((((((((((((((0.0:0.03254,0.0:0.03254):0.00913,(0.0:0.03254,0.0:0.03254):0.00913):0.

Cluster 1
((((((((((((((((((1.0:0.07344,((1.0:0.06508,1.0:0.06508):0.00066,(1.0:0.05008,1.0:0.05008)

Time taken to build model (full training data) : 0.04 seconds

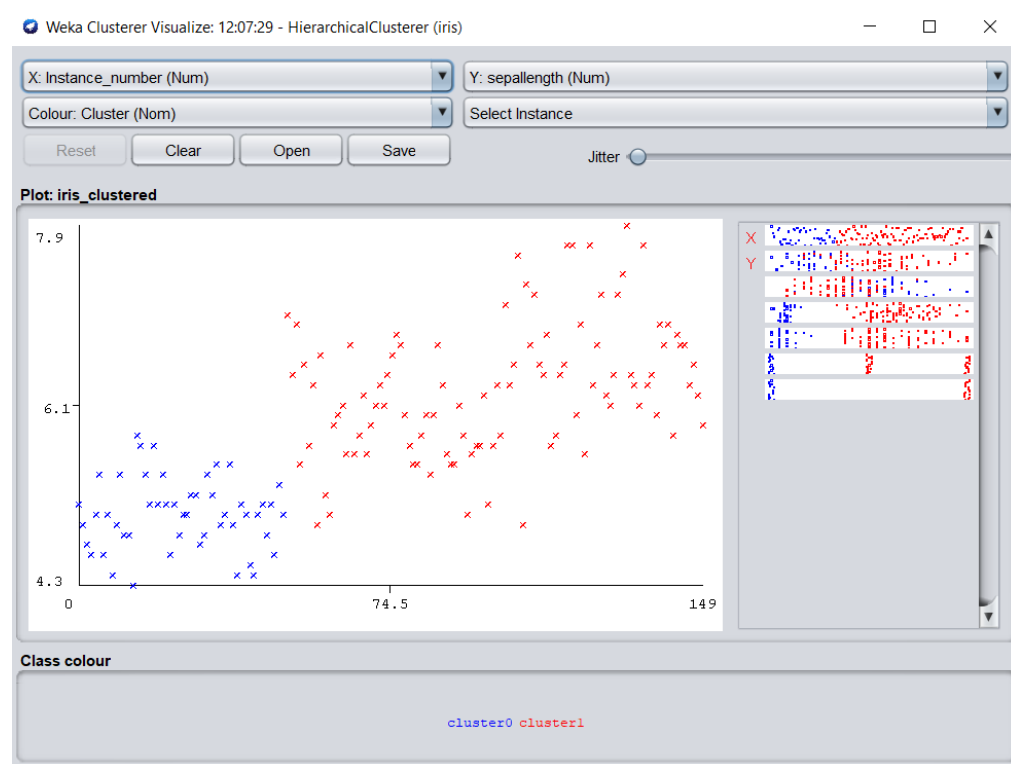
=== Model and evaluation on training set ===

Clustered Instances

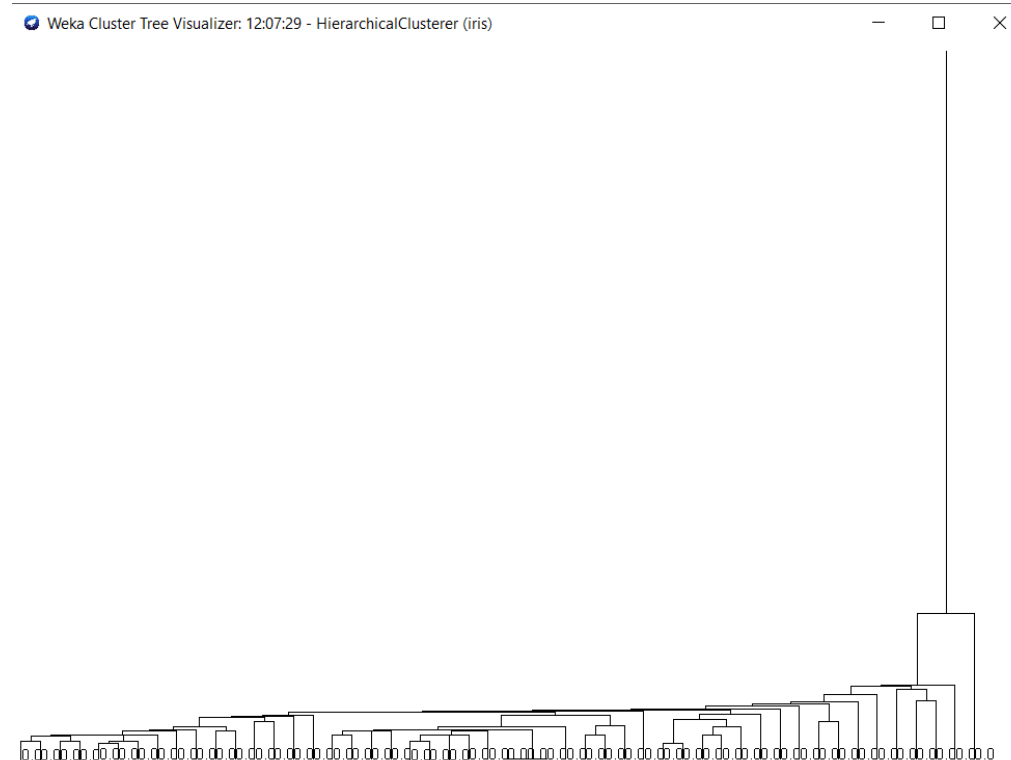
0       50 ( 33%)
1      100 ( 67%)
```

The 'Result list' on the left shows several entries, with '11:58:53 - HierarchicalClusterer' selected. The 'Status' bar at the bottom shows 'OK' and a 'Log' button.

Cluster Visualize



Dendrogram



Manhattan Distance

The screenshot shows the Weka Explorer window with the 'Cluster' tab selected. The 'HierarchicalClusterer' algorithm is chosen, with parameters set to '-N 2 -L SINGLE -P -A *weka.core.ManhattanDistance -R first-last'. The 'Cluster mode' section has 'Use training set' selected, and 'Store clusters for visualization' is checked. The 'Clusterer output' pane displays the following text:

```
Attributes: 5
            sepallength
            sepalwidth
            petallength
            petalwidth
            class

Test mode:  evaluate on training data

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

Cluster 0
((((((((((((((((((((((0.0:0.04167,0.0:0.04167):0.00306,0.0:0.04473):0,0.0:0.04473):0.013

Cluster 1
((((((((((((((((((((((1.0:0.14501,((((((((((1.0:0.08945,1.0:0.08945):0.00306,(1.0:0.06944,1.0:0.06944

Time taken to build model (full training data) : 0.03 seconds

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

Clustered Instances

0      50 ( 33%)
1     100 ( 67%)
```

The 'Result list' on the left shows a list of recent operations, with '12:10:20 - HierarchicalClusterer' selected. The 'Status' bar at the bottom shows 'OK' and a 'Log' button.

Conclusion:

Thus, From the above experiment I have understood different clustering techniques and classification techniques and also understood how to implement them using Weka Tool.