

## EXPERIMENT 6

**TITLE:** To perform Naive Bayes, ID3 and K-Means using WEKA

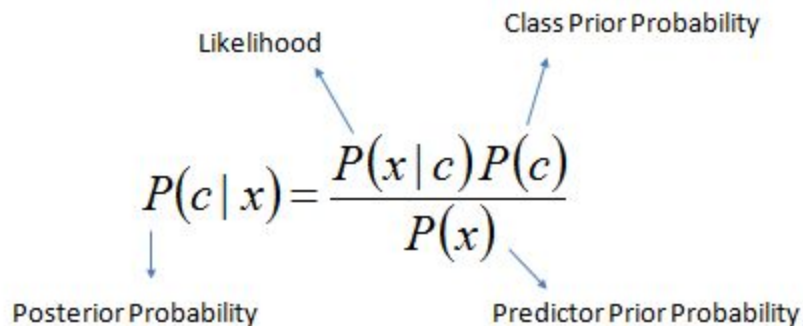
### THEORY:

#### Naive Bayes:

The Naive Bayesian classifier is based on Bayes' theorem with the independence assumptions between predictors. A Naive Bayesian model is easy to build, with no complicated iterative parameter estimation which makes it particularly useful for very large datasets. Despite its simplicity, the Naive Bayesian classifier often does surprisingly well and is widely used because it often outperforms more sophisticated classification methods.

Algorithm:

Bayes theorem provides a way of calculating the posterior probability,  $P(c|x)$ , from  $P(c)$ ,  $P(x)$ , and  $P(x|c)$ . Naive Bayes classifier assume that the effect of the value of a predictor ( $x$ ) on a given class ( $c$ ) is independent of the values of other predictors. This assumption is called class conditional independence.



The diagram shows the formula for Bayes' theorem: 
$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$
 Arrows point from the terms in the formula to their respective labels: 

- $P(c|x)$  points to "Posterior Probability"
- $P(x|c)$  points to "Likelihood"
- $P(c)$  points to "Class Prior Probability"
- $P(x)$  points to "Predictor Prior Probability"

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

- $P(c|x)$  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*.

#### Decision Tree:

Decision tree builds classification or regression models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with **decision nodes** and **leaf nodes**. A decision node (e.g., Outlook) has two or more branches (e.g., Sunny, Overcast and Rainy). Leaf node (e.g., Play) represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called **root node**. Decision trees can handle both categorical and numerical data.



Algorithm:

1. We begin with the original data set  $S$  as the root node
2. In each iteration the algorithm iterates through every unused attribute of the data set  $S$  and calculates the entropy  $H(S)$  (or information gain  $IG(A)$ ) of that attribute
3. Next it selects the attribute which has the smallest entropy (or largest information gain) value
4. The data set  $S$  is then split by the selected attribute (e.g.  $age < 50$ ,  $50 \leq age < 100$ ,  $age \geq 100$ ) to produce subsets of the data
5. The algorithm continues to recurse on each subset, considering only attributes never selected before
6. Recursion on a subset may stop in one of these cases:
  - every element in the subset belongs to the same class (+ or -), then the node is turned into a leaf and labelled with the class of the examples
  - there are no more attributes to be selected, but the examples still do not belong to the same class (some are + and some are -), then the node is turned into a leaf and labelled with the most common class of the examples in the subset
  - there are no examples in the subset, this happens when no example in the parent set was found to be matching a specific value of the selected attribute, for example if there was no example with  $age \geq 100$ . Then a leaf is created, and labelled with the most common class of the examples in the parent set

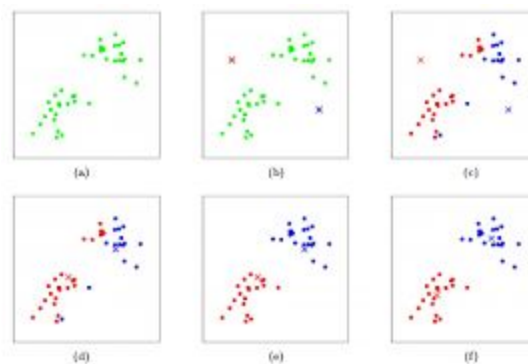
7. Throughout the algorithm, the decision tree is constructed with each non-terminal node representing the selected attribute on which the data was split, and terminal nodes representing the class label of the final subset of this branch

## K-Means:

*K*-means clustering is a type of unsupervised learning, which is used when you have unlabeled data (i.e., data without defined categories or groups). The goal of this algorithm is to find groups in the data, with the number of groups represented by the variable *K*. The algorithm works iteratively to assign each data point to one of *K* groups based on the features that are provided. Data points are clustered based on feature similarity. The results of the *K*-means clustering algorithm are:

1. The centroids of the *K* clusters, which can be used to label new data
2. Labels for the training data (each data point is assigned to a single cluster)

Rather than defining groups before looking at the data, clustering allows you to find and analyze the groups that have formed organically.



Algorithm:

### Step 1: Initialization

The first thing k-means does, is randomly choose *K* examples (data points) from the dataset (the 4 green points) as initial centroids and that's simply because it does not know yet where the center of each cluster is. (a centroid is the center of a cluster).

### Step 2: Cluster Assignment

Then, all the data points that are the closest (similar) to a centroid will create a cluster. If we're using the Euclidean distance between data points and every centroid, a straight line is drawn between two centroids, then a perpendicular bisector (boundary line) divides this line into two clusters.

### Step 3: Move the Centroid

Now, we have new clusters, that need centers. A centroid's new value is going to be the mean of all the examples in a cluster.

We'll keep repeating step 2 and 3 until the centroids stop moving, in other words, K-means algorithm is converged.

### **Arff Files:**

@relation weather

@attribute outlook {sunny, overcast, rainy}

@attribute temperature numeric

@attribute humidity numeric

@attribute windy {TRUE, FALSE}

@attribute play {yes, no}

@data

sunny,85,85,FALSE,no

sunny,80,90,TRUE,no

overcast,83,86,FALSE,yes

rainy,70,96,FALSE,yes

rainy,68,80,FALSE,yes

rainy,65,70,TRUE,no

overcast,64,65,TRUE,yes

sunny,72,95,FALSE,no

sunny,69,70,FALSE,yes

rainy,75,80,FALSE,yes

sunny,75,70,TRUE,yes

overcast,72,90,TRUE,yes

overcast,81,75,FALSE,yes

rainy,71,91,TRUE,no

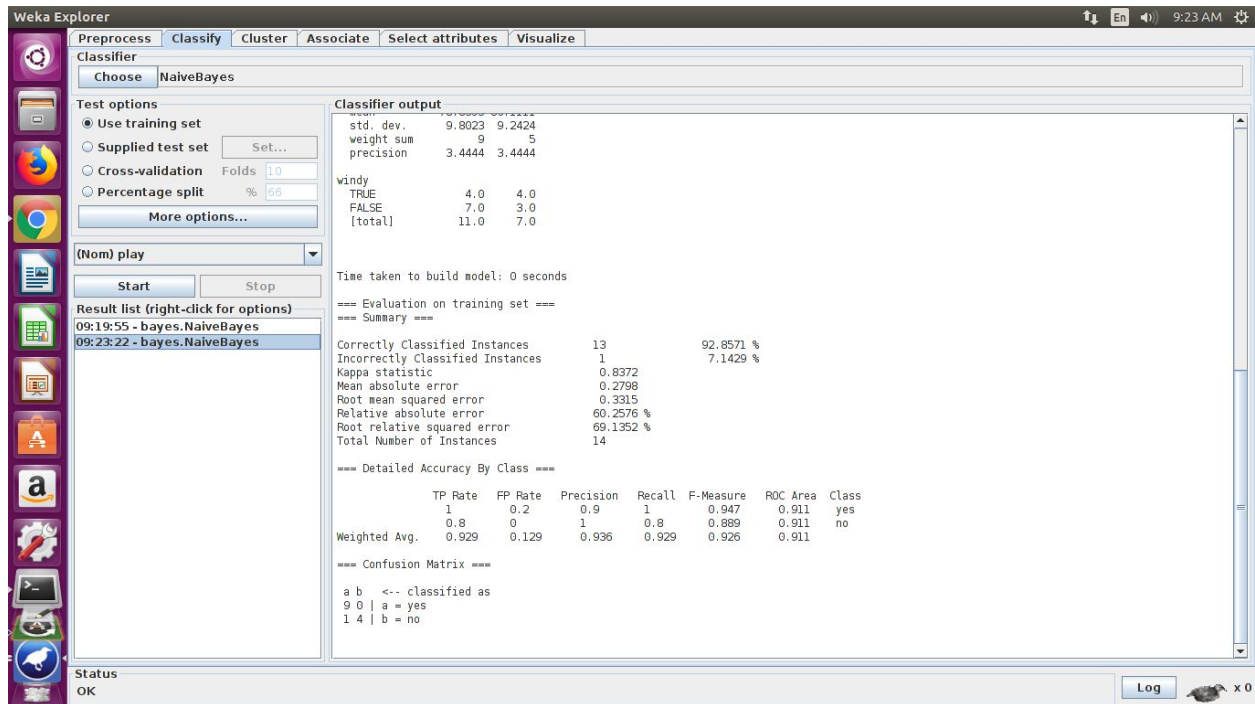


Fig 1: Classification on Weather Dataset with Naive Bayes

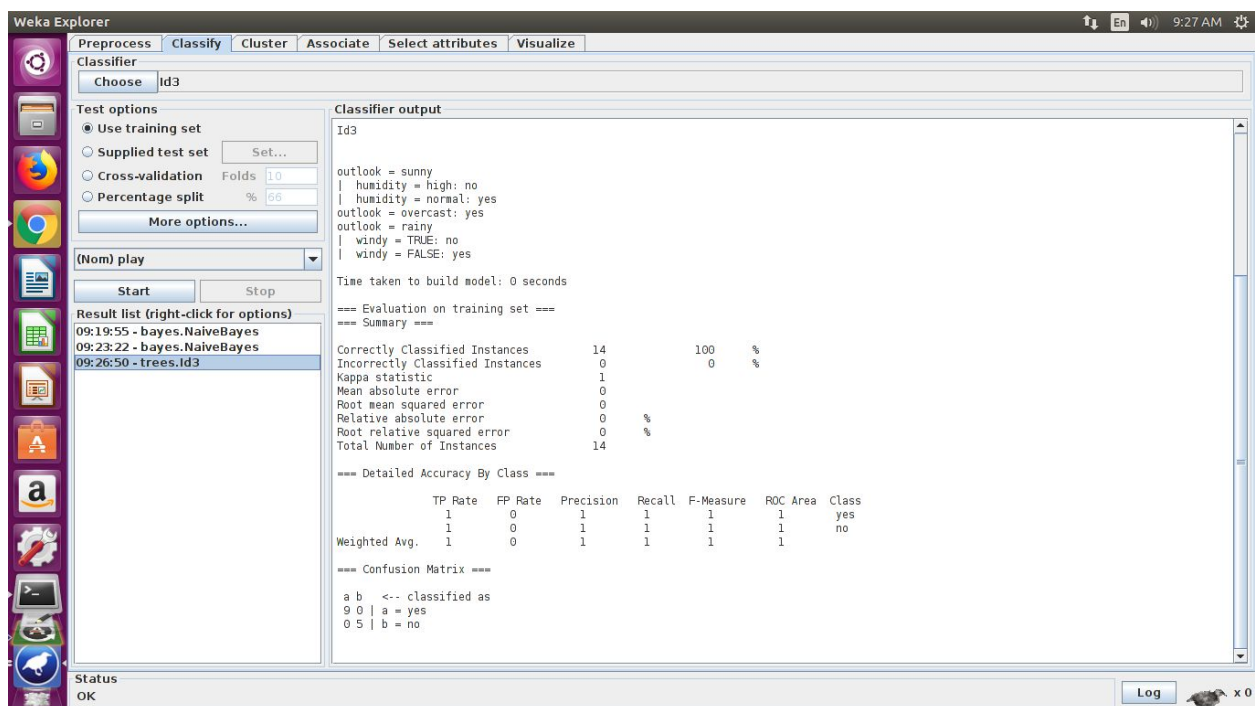


Fig 2: Classification of Weather Dataset with ID-3

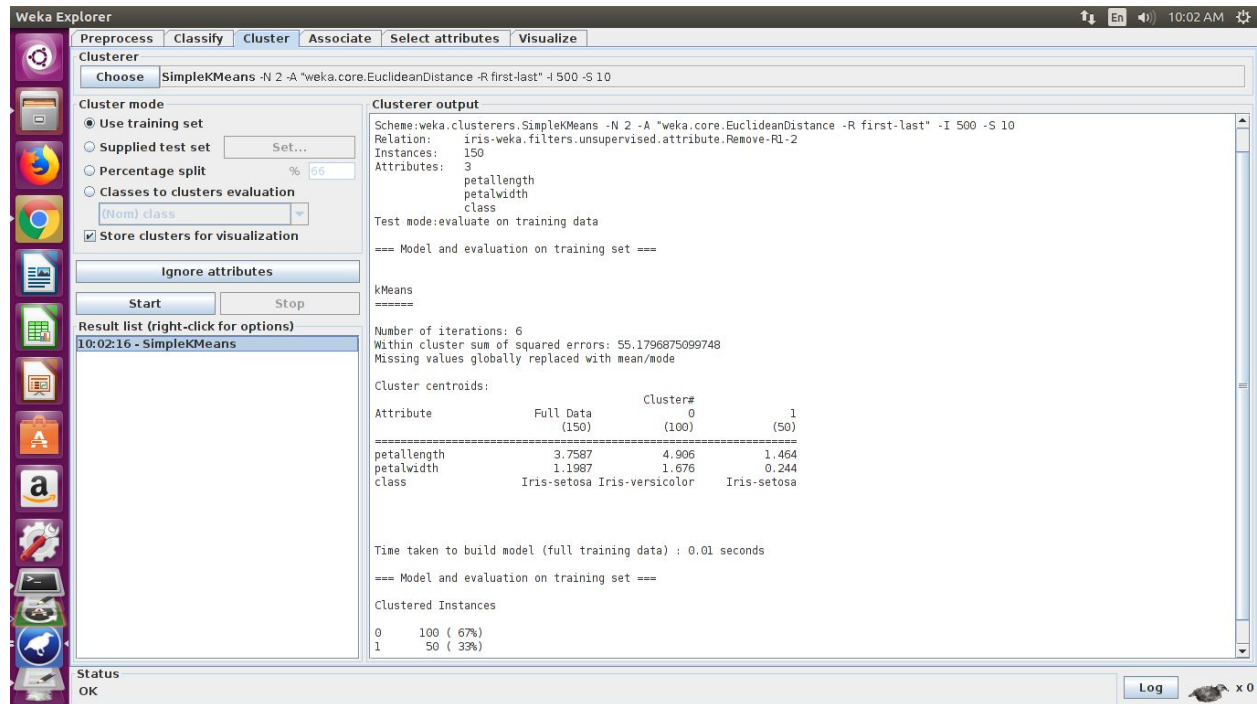


Fig 3: Clustering Iris Dataset with K-Means