* **INTRODUCTION**:

The three algorithms being considered over here are (1) C4.5 decision tree analysis; (2) k-NN classifier and (3) Naïve Bayes classifier. They are taken being run from the interactive GUI of WEKA machine learning GUI.

* **C4.5 decision tree**: C4.5 is a software extension of the basic ID3 algorithm designed by Quinlan. It selects one attribute from a set of training instances. It then selects an initial subset of the training instances. Next, the attribute and the subset of instances are used to build a decision tree. The rest of the training instances are used to test the accuracy of the constructed tree. If all instances are correctly classified the algorithm stops. If an instance is incorrectly classified, it is added to the initial subset and a new tree is constructed. This iteration continues until either of the following occurs: A tree is built that classifies all instances correctly ***OR*** a tree is built from the entire training set.
* **k-NN classifier** : KNN classifier is a classification algorithm, which classifies unknown instances by relating the unknown to the known according to some distance/similarity function. KNN is a non-parametric learning algorithm, which means that it does not make any assumptions on the underlying data distribution. The KNN algorithm works by finding the k-nearest neighbors and applying a majority voting. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k-nearest neighbors (k is a positive integer, typically small).

The different distance measures used by K-NN for measuring similarity are

1. Euclidean:
2. Manhattan:
3. Minkowski:

* **Naïve Bayes classifier:** The Naive Bayes Classifier technique is based on the so-called Bayesian theorem and is particularly suited when the dimensionality of the inputs is high. Despite its simplicity, Naive Bayes can often outperform more sophisticated classification methods. Given a class variable y and a dependent feature vector x_1 through x_n, Bayes’ theorem states the following relationship:   
     
    
  Using the naïve independence assumption that :

For all i, this relationship is simplified to

Since P(x_1, \dots, x_n) is constant given the input, we can use the following classification

→

and we can use Maximum A Posteriori (MAP) estimation to estimate P(y) and P(x_i \mid y); the former is then the relative frequency of class y in the training set.

* **EXPERIMENTS AND RESULTS:**

*Iris.data- 10 fold cross validation on test data*

|  |  |  |  |
| --- | --- | --- | --- |
| Classifier | Accuracy of classification | Root mean squared error | Confusion Matrix |
| C4.5 | 96% | 0.1586 |  |
| K-NN/ K=7 | 96.6667 % | 0.1282 |  |
| Naïve Bayes | 96% | 0.155 |  |

*Iris.data- on training data*

|  |  |  |  |
| --- | --- | --- | --- |
| Classifier | Accuracy of classification | Root mean squared error | Confusion Matrix |
| C4.5 | 98% | 0.108 |  |
| K-NN/ K=7 & K=3 | 96.6667 % | K = 3; 0.0235  K = 7; 0.0337 | K=3  K = 7, |
| Naïve Bayes | 96% | 0.0324 |  |

--------------------------------------------------------------------------------------------------------------------------------------------

*Car.data- 10 fold cross validation on test data*

|  |  |  |  |
| --- | --- | --- | --- |
| Classifier | Accuracy of classification | Root mean squared error | Confusion Matrix |
| C4.5 | 96.3542% | 0.0254 |  |
| Naïve Bayes | 82.4653 % | 0.1087 |  |

*Car.data- on training data*

|  |  |  |  |
| --- | --- | --- | --- |
| Classifier | Accuracy of classification | Root mean squared error | Confusion Matrix |
| C4.5 | 98.0324 %% | 0.0167 |  |
| Naïve Bayes | 83.2176 % | 0.1069 |  |

* **DISCUSSION AND CONCLUSION:**
* For the results on iris.data and car.data it can be observed that C4.5 decision tree ID3 has the greatest discrepancy between the accuracy results of the training data and the cross validation data. The reason for this discrepancy is that Decision Trees try to over fit the data. This is not usually the case for either KNN or Naïve Bayes. Decision trees have a very *hard margin* for classification in a sense, once it classifies instances into separate groups based on feature value at a particular node. Hence they only correctly classify instances which have a similar pattern.
* Choice of k is very critical – A small value of k means that noise will have a higher influence on the result. A large value makes it computationally expensive and defeats the basic philosophy behind KNN (that points that are near might have similar densities or classes). A simple approach to select k is set k = n^(1/2). In this case it can be assumed that for iris.data, K=3, K=5 did not create neighbours which were similar in their overall patterns. The same goes for car.data set.
* From the results it can also be concluded that, KNN is the best classifier among these algorithms when the cross validation is performed. sThe C4.5 takes an edge over Naïve Bayes.

**KNN>C4.5>Naïve Bayes.**

* The main focus over here is the reason why these algorithms work so differently. C4.5 for example tries to over fit the data. KNN on the other hand is susceptible to noise in the data because, the presence of a spike in the value of a sample causes the sample to change its nearest neighbours and lands it in a region far from where it should be and completely alters the system representation. On the other hand, the Naïve Bayes suffers from a problem of making the probability of a particular random variable instance zero if one of the sub cases in the posterior distribution of the variable is 0. Hence it suffers from a memory effect where absence of an instance with a feature value different from what has been previously observed reduces the probability of occurrence of a correct instance to zero.
* The reason why the results are such is because, KNN is able to exploit the class membership of the K nearest neighbors with increasing value of K. This helps it in making better decisions.
* The reason for observing a similar range of accuracy for C4.5 and Naïve Bayes in the results is that, these methods do not usually have the capability to visualize data in a higher dimensional plane like SVMs. This causes them to limit their accuracy since they do not get a separator geometrical figure like a hyperplane as in SVMs.