Author: Kunjan Mhaske

Counterfeit Currency Detection using Decision Tree and Naive Bayes

Dataset: For this project, Banknote Authentication Dataset is taken into consideration. In this dataset, there are 4 Attributes datapoints having continuous datapoints:

1) Variance of wavelet transformed image (WTI)
2) Skewness of WTI
3) Curtosis of WTI
4) Entropy of image.

Also, 2 Classes which are class 0 - Fake and class 1 – Original. The dataset has 1372 observation values.
Prediction outcome based on random choice of class prediction: Accuracy in range of 50 – 52%
Prediction outcome based on Majority class (viz. class 0 in this case) choice: Accuracy is 55.54 %

Decision Tree: It is decision support tool that uses tree-like model of decisions and their probable outcomes based on the predefined conditions. The goal is to create a model that predicts the value of a target variable based on several input variables. Internal nodes are labelled with input feature. The arcs coming from a node labelled with an output feature or the arc leads to a subordinate decision node on different input feature. The conjunctions of features that lead to the class labels which are at the leaf nodes. The tree can be learned the information by splitting the source data into subsets based on the attribute value tests.

In this project, the split-point is based on the mean of 2 consecutive values pairs from entire values of that particular attribute and compare them with each other based on maximum gain that branch could attain after calculating that subset entropy and whole set entropy including all attributes. This will give perfectly divide the attribute values into two branches and ease the decision process while parsing testing data. This process is repeated on each derived subset in a recursive manner. The recursion is completed when the all attributes are used in the branching process. At that point, the leaf node is labelled with prediction based on most class values in that subset.

Naïve Bayes Classifier: It is family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features. It assigns class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. It assumes that the value of a particular feature is independent of the value of any other feature, given the class variable. For example, a fruit may be considered to be a banana if it is yellow, cylindrical, about 6 cm in diameter. A naïve bayes classifier considers each of these features to contribute independently to the probability that this fruit is a banana, regardless of any possible correlations between color, shape and diameter features. An advantage of naive Bayes is that it only requires a small number of training data to estimate the parameters necessary for classification.

In this project, the classifier learns the data based on the median and variance values of specific feature/attribute values and finding its probability density and made decision by comparing it with class values from each data instance and extracting out the best probable class label for that particular instance of test data.

How to run the code: Python 3 is used with numpy, scipy, math, random libraries
Command line *python3 decision_tree.py* for decision tree implementation.
Note: **banknote.txt** dataset file is needed to be in same folder as decision_tree.py
Command line *python3 naive_bayes.py* for naïve bayes classifier
Note: You can alter the Global Variable "R" to restrict the re-run of program. By default, it is set to run 5 times for cross validation to get proper range of results.

Cross Validation: For this project, the data is partitioned into 5 equal parts and performed the validation on the data as follows:

a) First take 1st part which is beginning 20% data as training data and rest of 80% as testing data.
b) Then for second iteration, take 2nd part as training data which is 20% of data which is not taken in previous iteration and rest of 80% data for testing which may contains previously (iterations) taken training and testing data.
c) Do this for 5 times then the 100% of data is considered for testing and training.
d) Calculate the accuracy at each iteration and save it for the final result.
e) At the end, calculate the average of the accuracies obtained in the previous steps and this will be the final accuracy of particular classifier.
Note: One could obtain higher precision by running the classifier multiple times on same dataset with random sequence.

Observations and Results:

Bank Note Dataset: 1372 Samples (762 of class 0 and 610 of class 1), with 4 Features and 2 Classes

Cross Validation: Decision Tree - Average Accuracy of 5 re-runs: **97.39085 %**

Sample Range of Training (20%)	Sample Range of Testing (80%)	Testing Points	Correct Prediction Counts	Accuracy in Percentages
0 - 273	273 – 1371	1098	985	89.70
274 - 547	0 – 273, 547-1371	1098	972	88.52
548 – 822	0 – 547, 822 – 1371	1097	978	89.15
823 – 1096	0-822, 1096-1371	1098	1024	93.26
1097 – 1371	0-1097	1097	1046	95.35

Table: Result of only 1st Run of Decision Tree

By Parsing Complete data for both Training and Testing: 100% accuracy

----

Cross Validation: Naïve Bayes - Average Accuracy of 5 re-runs: **95.134699 %**

Sample Range of Training (20%)	Sample Range of Testing (80%)	Accuracy in Percentages
0 – 273	273 – 1371	85.42
274 – 547	0 – 273, 547-1371	85.42
548 – 822	0 – 547, 822 – 1371	83.40
823 – 1096	0-822, 1096-1371	92.98
1097 – 1371	0-1097	95.35

Table: Result of only 1st Run of Naïve Bayes Classifier

By Parsing Complete data for both Training and Testing: 70.84548 % Accuracy

Conclusion:

Decision trees are very flexible and easy to implement as well as to debug. It uses rationally distinguishing values such as entropy, gain of whole subset based on specific features and gives priority to maximum outcome. No pre-processing is required before parsing data to tree. Although, the accuracy level they give is much more, decision trees are prone to overfitting. Hence, the reliability shatters for new unknown data.

On other hand, Naïve Bayes shows lesser overfitting and it is smaller in size to implement as well as faster in processing the data. It can give very good results on the limited (small) amount of data in comparison with decision trees which required larger data.

In conclusion, I believe Naïve Bayes Classifiers are more reliable and robust than Decision Trees.