**Himanshu Kumar**

**Q.1-** **Write a Python program using Scikit-learn to split the iris dataset into 80% train data and 20% test data. Out of total 150 records, the training set will contain 120 records and the test set contains 30 of those records. Train or fit the data into the model and calculate the accuracy of the model using the Decision Tree Algorithm.**

**Ans.-**

**import pandas as pd**

**import numpy as np**

**from sklearn.neighbors import KNeighborsClassifier**

**from sklearn.model\_selection import train\_test\_split**

**iris = pd.read\_csv("iris.csv")**

**iris = iris.drop('Id',axis=1)**

**X = iris.iloc[:, :-1].values**

**y = iris.iloc[:, 4].values**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.20)**

**knn = KNeighborsClassifier(n\_neighbors=7)**

**knn.fit(X\_train, y\_train)**

**for i in np.arange(1, 10):**

**knn2 = KNeighborsClassifier(n\_neighbors=i)**

**knn2.fit(X\_train, y\_train)**

**print("For k = %d accuracy is"%i,knn2.score(X\_test,y\_test))**

**Q.3- What are the advantages & Disadvantages of Decision Tree Algorithms?**

Ans.-

Advantages:

1. Compared to other algorithms decision trees requires less effort for data preparation during pre-processing.
2. A decision tree does not require normalization of data.
3. A decision tree does not require scaling of data as well.
4. Missing values in the data also do NOT affect the process of building a decision tree to any considerable extent.
5. A Decision tree model is very intuitive and easy to explain to technical teams as well as stakeholders.

**Disadvantage:**

1. A small change in the data can cause a large change in the structure of the decision tree causing instability.
2. For a Decision tree sometimes calculation can go far more complex compared to other algorithms.
3. Decision tree often involves higher time to train the model.
4. Decision tree training is relatively expensive as the complexity and time has taken are more.
5. The Decision Tree algorithm is inadequate for applying regression and predicting continuous values.

**Q.4- Describe at least one way to overcome the problem of overfitting when constructing decision trees?**

**Ans.- Decision Tree - Overfitting**

Overfitting is a significant practical difficulty for decision tree models and many other predictive models. Overfitting happens when the learning algorithm continues to develop hypotheses that reduce training set error at the cost of an  
increased test set error. There are several approaches to avoiding overfitting in building decision trees.

* **Pre-pruning** that stop growing the tree earlier, before it perfectly classifies the training set.
* **Post-pruning** that allows the tree to perfectly classify the training set, and then post prune the tree.

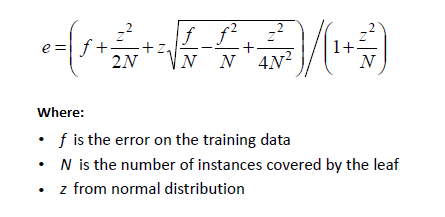
Practically, the second approach of post-pruning overfit trees is more successful because it is not easy to precisely estimate when to stop growing the tree. The important step of tree pruning is to define a criterion be used to determine the correct final tree size using one of the following methods:

1. Use a distinct dataset from the training set (called validation set), to evaluate the effect of post-pruning nodes from the tree.
2. Build the tree by using the training set, then apply a statistical test to estimate whether pruning or expanding a particular node is likely to produce an improvement beyond the training set.
   * Error estimation
   * Significance testing (e.g., Chi-square test)
3. Minimum Description Length principle : Use an explicit measure of the complexity for encoding the training set and the decision tree, stopping growth of the tree when this encoding size (size(tree) + size(misclassifications(tree)) is minimized.

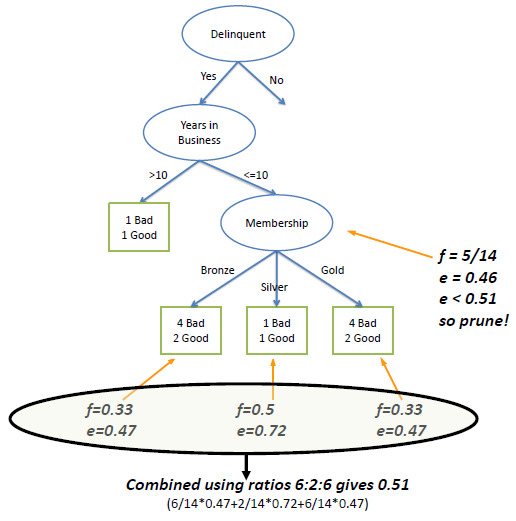
The first method is the most common approach. In this approach, the available data are separated into two sets of examples: a training set, which is used to build the decision tree, and a validation set, which is used to evaluate the impact of pruning the tree. The second method is also a common approach. Here, we explain the error estimation and Chi2 test. 

**Post-pruning using Error estimation**

Error estimate for a sub-tree is weighted sum of error estimates for all its leaves. The error estimate (*e*) for a node is:



In the following example we set *Z* to 0.69 which is equal to a confidence level of 75%.



The error rate at the parent node is 0.46 and since the error rate for its children (0.51) increases with the split, we do not want to keep the children. **Post-pruning using Chi2 test**In [Chi2 test](https://www.saedsayad.com/categorical_categorical.htm) we construct the corresponding frequency table and calculate the Chi2 value and its probability.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Bronze | Silver | Gold |
| Bad | 4 | 1 | 4 |
| Good | 2 | 1 | 2 |

Chi2 = 0.21          Probability = 0.90         degree of freedom=2

 If we require that the probability has to be less than a limit (e.g., 0.05), therefore we decide not to split the node.

**Q.5** -What are the advantages of Naïve Bayes Classifier?

Ans.- **Advantages of Naive Bayes Classifier**

The following are some of the benefits of the Naive Bayes classifier:

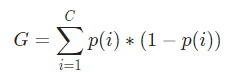
* It is simple and easy to implement
* It doesn’t require as much training data
* It handles both continuous and discrete data
* It is highly scalable with the number of predictors and data points
* It is fast and can be used to make real-time predictions
* It is not sensitive to irrelevant features

Q.6- What is gini index and how is it used in decision trees..?

**Ans.-** The Gini Index or Gini Impurity is calculated by subtracting the sum of the squared probabilities of each class from one. It favours mostly the larger partitions and are very simple to implement. In simple terms, it calculates the probability of a certain randomly selected feature that was classified incorrectly.

The Gini Index varies between 0 and 1, where 0 represents purity of the classification and 1 denotes random distribution of elements among various classes. A Gini Index of 0.5 shows that there is equal distribution of elements across some classes.

Mathematically, The Gini Index is represented by



The Gini Index works on categorical variables and gives the results in terms of “success” or “failure” and hence performs only binary split. It isn’t computationally intensive as its counterpart – Information Gain. From the Gini Index, the value of another parameter named Gini Gain is calculated whose value is maximised with each iteration by the Decision Tree to get the perfect CART

**Q7 Using the above dataset, create a Naïve Bayes Classifier.**

**Ans.-**

**Q.8- Name different types of Naïve Bayes algorithm with their use cases?**

**Ans.- N**aive Bayes is one the most popular and beginner-friendly algorithms that anyone can use. In this article, we are going to explore the Naive Bayes Algorithm.

Note: If you are more interested in learning concepts in an Audio-Visual format, We have this entire article explained in the video below. If not, you may continue reading.

## Types of Naive Bayes

Now let’s discuss different types of Naive Bayes algorithm and which is used when. So, we have three types

### Gaussian Naive Bayes

This type of Naive Bayes is used when variables are continuous in nature. It assumes that all the variables have a normal distribution. So if you have some variables which do not have this property, you might want to transform them to the features having distribution normal.

### Multinomial Naive Bayes

Next comes the multinomial Naive Bayes. This is used when the features represent the frequency.

Suppose you have a text document and you extract all the unique words and create multiple features where each feature represents the count of the word in the document. In such a case, we have a frequency as a feature. In such a scenario, we use multinomial Naive Bayes.

It ignores the non-occurrence of the features. So, if you have frequency 0 then the probability of occurrence of that feature will be 0 hence multinomial naive Bayes ignores that feature. It is known to work well with text classification problems.

### Bernoulli Naive Bayes

This is used when features are binary. So, instead of using the frequency of the word, if you have discrete features in 1s and 0s that represent the presence or absence of a feature. In that case, the features will be binary and we will use Bernoulli Naive Bayes.

Also, this method will penalize the non-occurrence of a feature, unlike multinomial Naive Bayes.

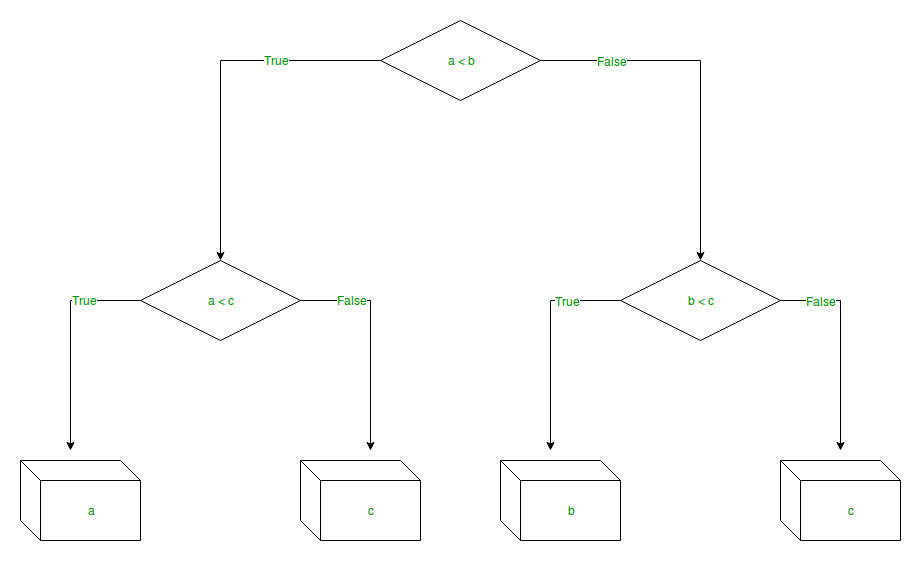
Q.9- What is Decision tree Regressor? Explain one case where you will use Decision Tree Regressor?

**Ans.-** **Decision Tree** is a decision-making tool that uses a flowchart-like tree structure or is a model of decisions and all of their possible results, including outcomes, input costs, and utility.  
Decision-tree algorithm falls under the category of supervised learning algorithms. It works for both continuous as well as categorical output variables.

The branches/edges represent the result of the node and the nodes have either:

1. Conditions [Decision Nodes]
2. Result [End Nodes]

3.The branches/edges represent the truth/falsity of the statement and take makes a decision based on that in the example below which shows a decision tree that evaluates the smallest of three numbers:

1. 
2. **Decision Tree Regression:**   
   Decision tree regression observes features of an object and trains a model in the structure of a tree to predict data in the future to produce meaningful continuous output. Continuous output means that the output/result is not discrete, i.e., it is not represented just by a discrete, known set of numbers or values.
3. **Discrete output example:** A weather prediction model that predicts whether or not there’ll be rain on a particular day.

**Q.10- What is the difference between Supervised and Unsupervised Learning**

## Ans.- Difference between Supervised and Unsupervised Learning

The following table highlights the major differences between Supervised and Unsupervised learning –

|  |  |  |
| --- | --- | --- |
| Factor | Supervised Learning | Unsupervised Learning |
| Objective | To train the algorithm for prediction. The outcome the algorithm predicts mostly occurs as per the human expectation. | To train the algorithm to find insights from the large volume of unclassified data. |
| Dataset Labelling | The datasets used in Supervised learning are labelled. | The data used in Unsupervised learning are unclassified. |
| Knowledge of Classes | The classes of data are known. | The number of classes is unknown as the model data is uncategorized and unlabelled. |
| Human Intervention | In supervised learning, human intervention is required to label the data appropriately. | The unsupervised learning makes the algorithm to take care of both; the input and the output of the data analizing but human intervention is only required for data validation. |
| Proximity with Artificial Intelligence | With remarkable amount of human intervention, Supervised learning seems distant from the real Artificial Intelligence. | With the less amount of human intervention, Unsupervised learning is very close to Artificial Intelligence. |
| Computational Complexity | It is simple and inexpensive. | It is complicated, timeconsuming, and requires more resources. |
| Learning Process | In Supervised learning, the process of training the algorithm takes place offline. | In case of unsupervised learning, the process of training the algorithms takes place in real time. |