Assignment-2.1

1. Please Find the following for the given diagram below Accuracy.
   a. Accuracy
   b. Recall
   c. Precision
   d. F1-Score

n = 165	Predicted: No	Predicted: Yes	
Actual: No	Tn =50	FP=10	60
Actual: Yes	Fn=5	Tp=100	105
	55	110	

a) Accuracy = $\frac{TP+TN}{FN+FP+TP+TN}$ = 0.91

b) Recall = $\frac{TP}{TP+FN}$ = 0.95

c) Precision = $\frac{TP}{TP+FP}$ = 0.91

d) F1-Score = $2\frac{PRE \times REC}{PRE+REC}$ = 0.93

2. Explain the following algorithms in detail
    1) K Nearest Neighbor
    2) Naïve Bayes
    3) Decision Tree

K Nearest Neighbor

K Nearest neighbor classifier classify unlabeled examples by assigning the class of similar labeled examples.

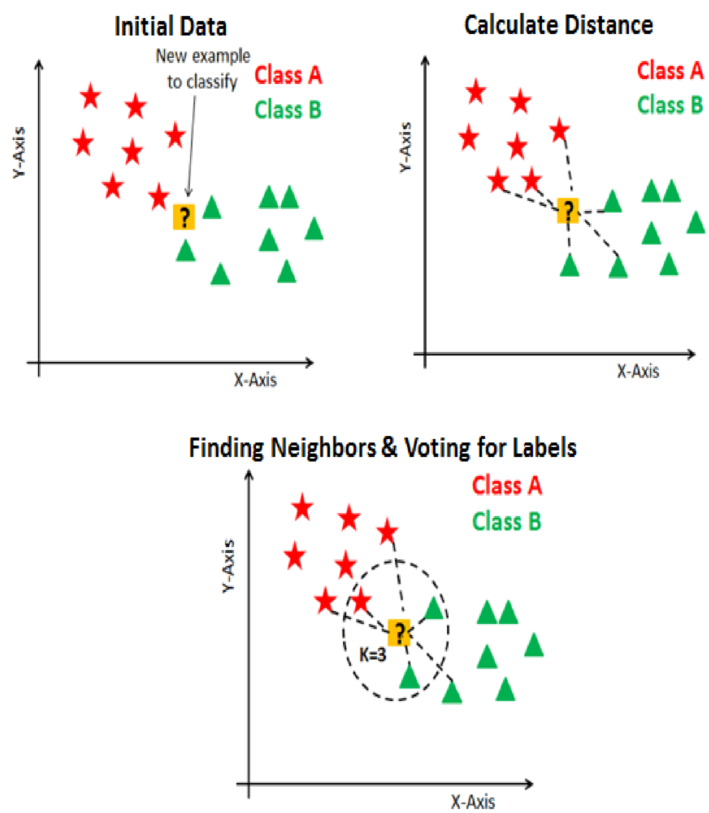
Real Life Application: K Nearest neighbor algorithm are most widely used in computer vision algorithm, predicting whether a person enjoy a movie or music recommendation and pattern in genetic data.

KNN Algorithm:

Below are the basic steps in KNN algorithm:

- ➢ Choose a number k and a distance metric
- ➢ Find the k nearest neighbors of the sample
- ➢ Assign the class label by majority vote.

KNN algorithm treats the features as co-ordinate in multidimensional feature space. It basically uses Euclidean distance between the target feature and the unlabeled feature.



Naïve Bayes

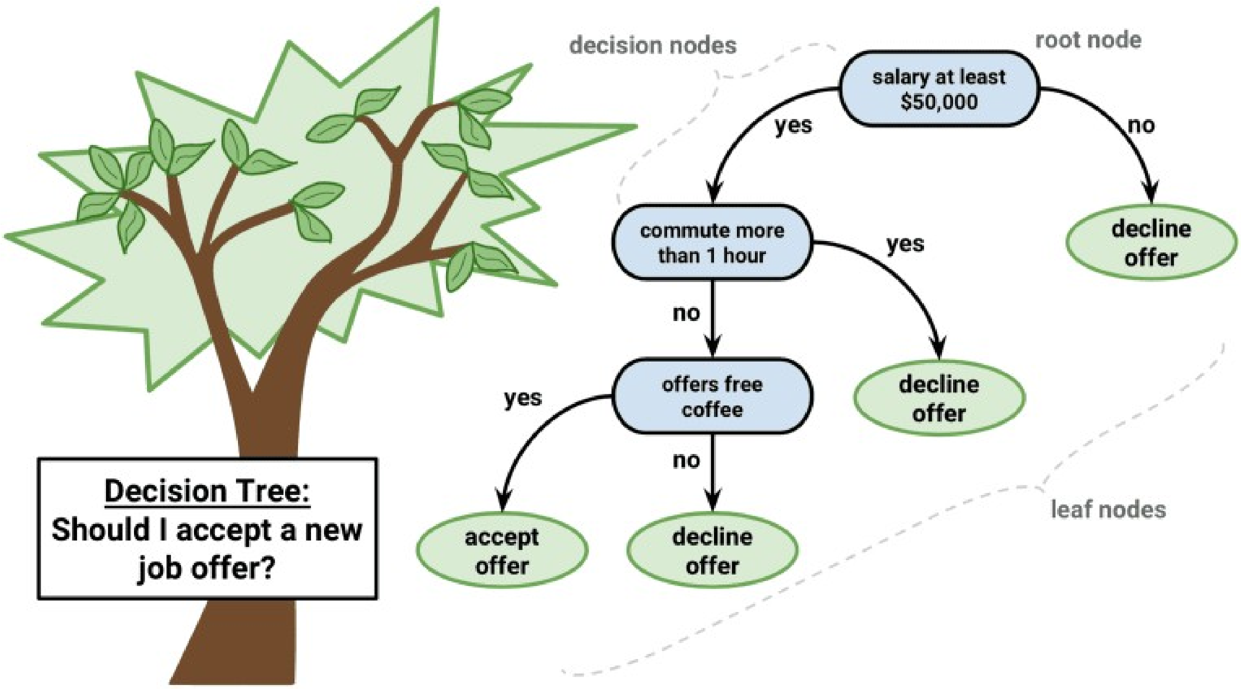
Naïve Bayes algorithms says that:

The probability of label L for class C, given the evidence provided by features F1 through Fn is equal to the probabilities of each piece of evidence conditioned on the class labels, the prior probabilities of the class label and a scaling factor 1/Z, which converts the likelihood values into probabilities.

$$P\left(C_L \middle| F_1 \ldots F_n\right) = \frac{1}{z} p\left(C_L\right) \prod_{i=1}^{n} p\left(F_i \middle| C_L\right)$$

Naïve Bayes algorithm Named as such because it makes some 'naïve' assumption about the data. Naïve Bayes assume that the all the features in the datasets are equally important and independent. This assumptions are rarely true in most real-world application.s

Decision Tree



Decision Tree can be thought of as breaking down data by making a decision based of a series of questions.

Decision Tree algorithms starts at the tree root and split the data on the feature that results in the largest information gain.

The Decision tree hopes to find the splits that reduce the entropy, ultimately increasing information gain (Homogeneity within the groups).

$$\textbf{Entropy (S)} = \sum_{i=1}^{c} -p_i \log_2(p_i)$$

Where

S = Segment of data

c = number of classes

p_i = Proportion of values falling into class i

Information Gain (F) = Entropy (S1) – Entropy (S2)

After split there are more than one partitions so the total entropy resulting from split is the sum of the entropy of each of the n partitions weighted by the proportion of examples falling in the partition (p_i)

$$\text{Total Entropy} = \sum_{i=1}^{n} w_i \, Entropy(p_i)$$