CM2604 Machine Learning

Supervised Machine Learning - Part 1

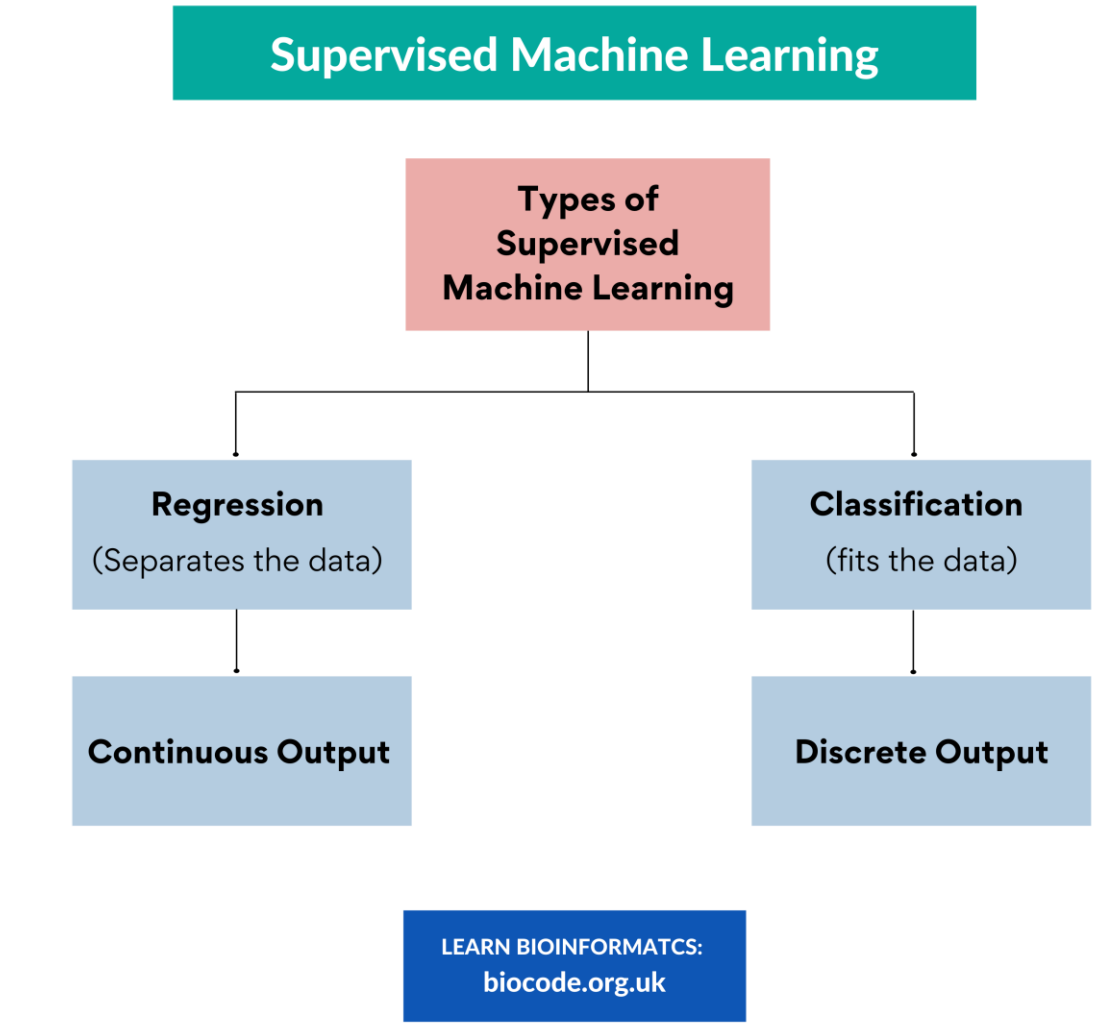
Week 03 | Prasan Yapa

ROBERT GORDON UNIVERSITY ABERDEEN
TEF Gold
INFORMATICS INSTITUTE OF TECHNOLOGY

Overview

- General Framework of Supervised Learning
- Nearest Neighbor
- K-NN
- Random Forest

General Framework of Supervised Learning

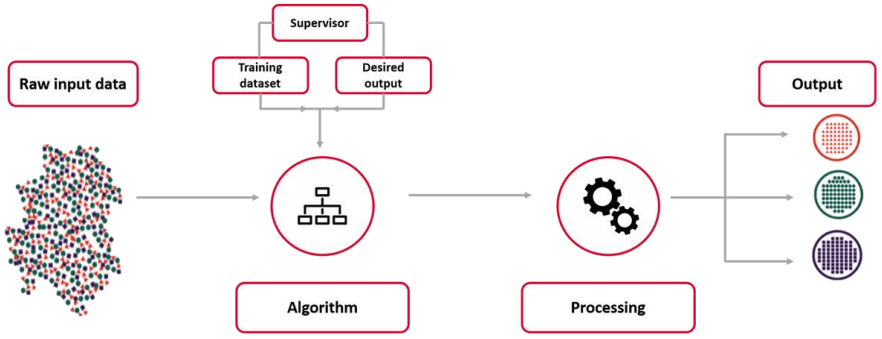


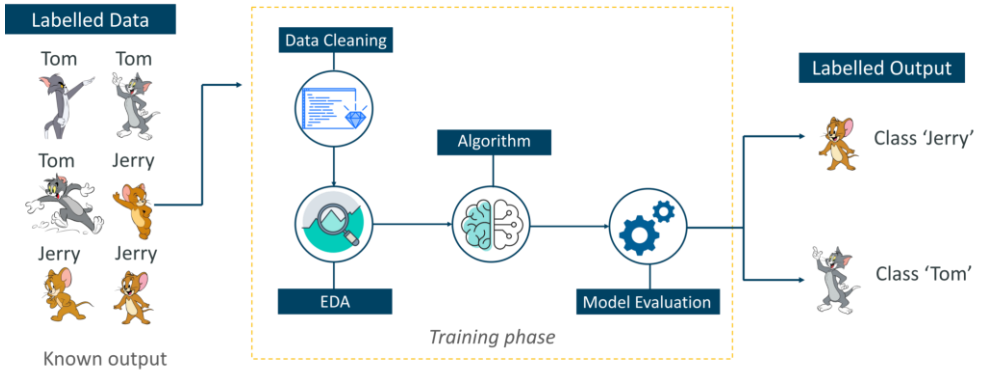
General Framework

- Supervised machine learning is generally used to classify data or make predictions.
- We have training data in the form of d-dimensional points, each with an associated label.
- The label might be a binary number, or an integer denoting the class of a data point in case of more than 2 classes. This leads to a problem of classification.
- The label might be a real number, in which case the problem would be one of regression.

General Framework

- Simply put, we want to learn a mapping from the data points to labels, using the training data, such that this mapping helps us get reliable estimates of the labels of the test data points.
- If there is a function that captures this relationship perfectly, we would have solved the problem satisfactorily if our learning algorithm can learn that function from the training data.
- However, there is generally a significant bit of noise in this mapping. Also, dealing with complex function classes is in general not easy.





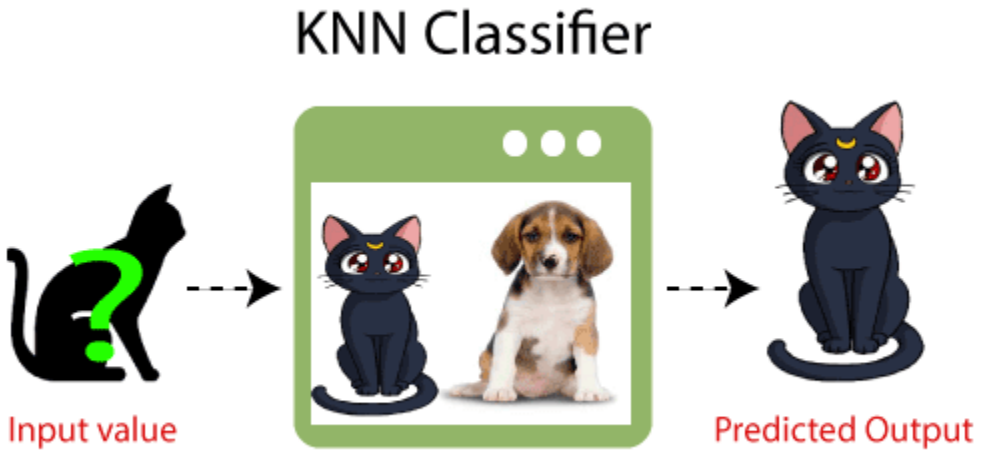
Nearest Neighbors

K-Nearest Neighbor (K-NN)

- K-NN is one of the simplest ML algorithms.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity.
- K-NN algorithm can be used for Regression as well as for Classification.

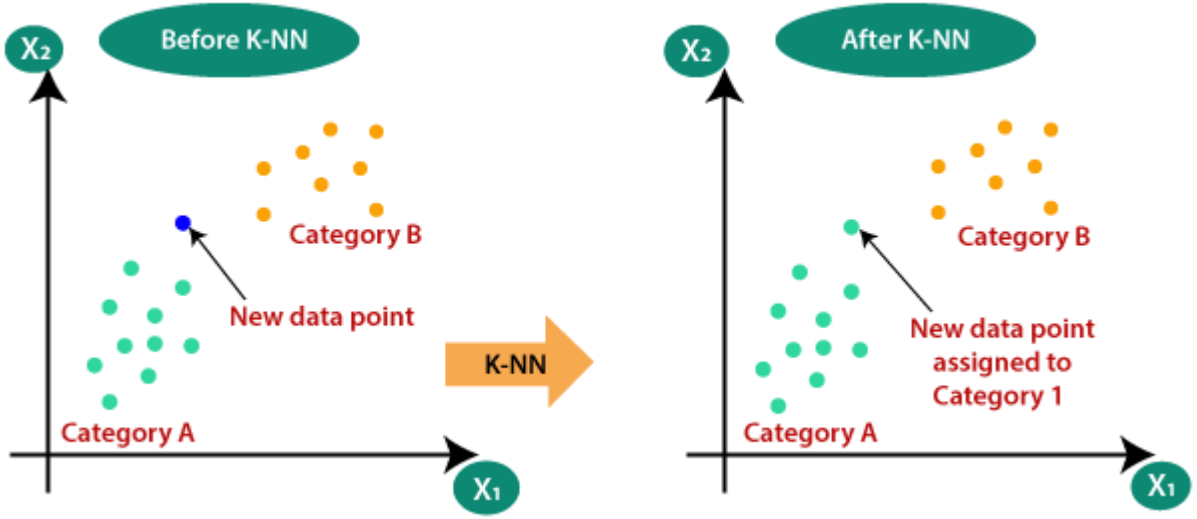
K-Nearest Neighbor (K-NN)

- Example:
  - Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So, for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.



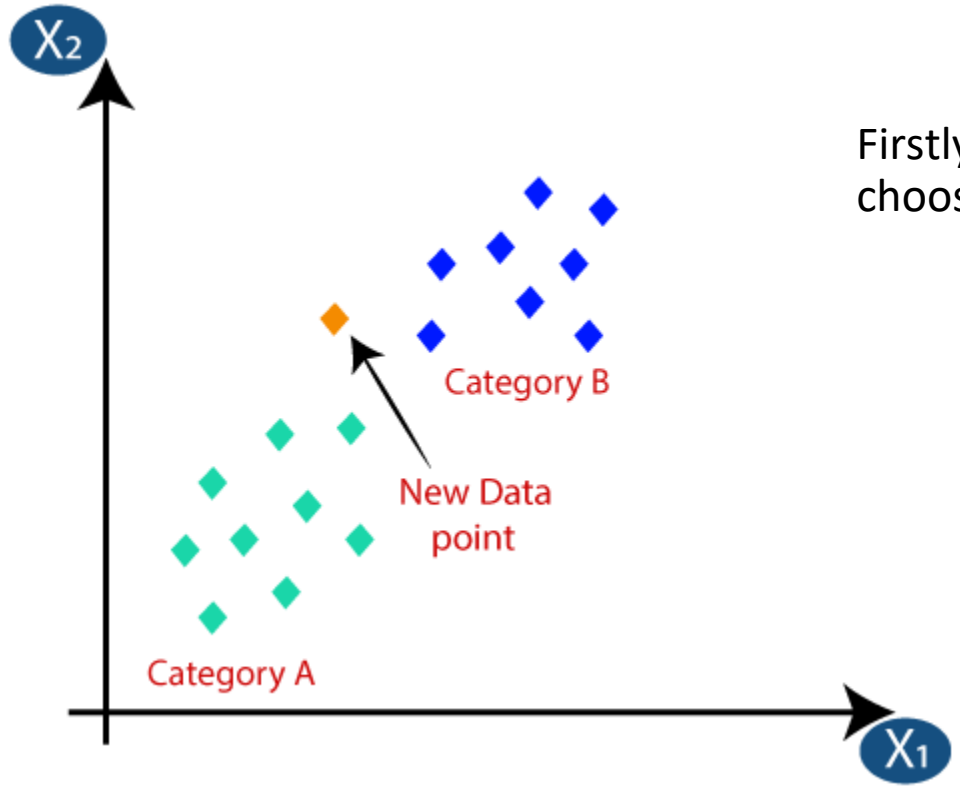
K-Nearest Neighbor (K-NN)

- Suppose there are two categories, i.e., Category A and Category B, and we have a new data point *x1*, so this data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset. Consider the below diagram:



How K-NN Works? Step 1

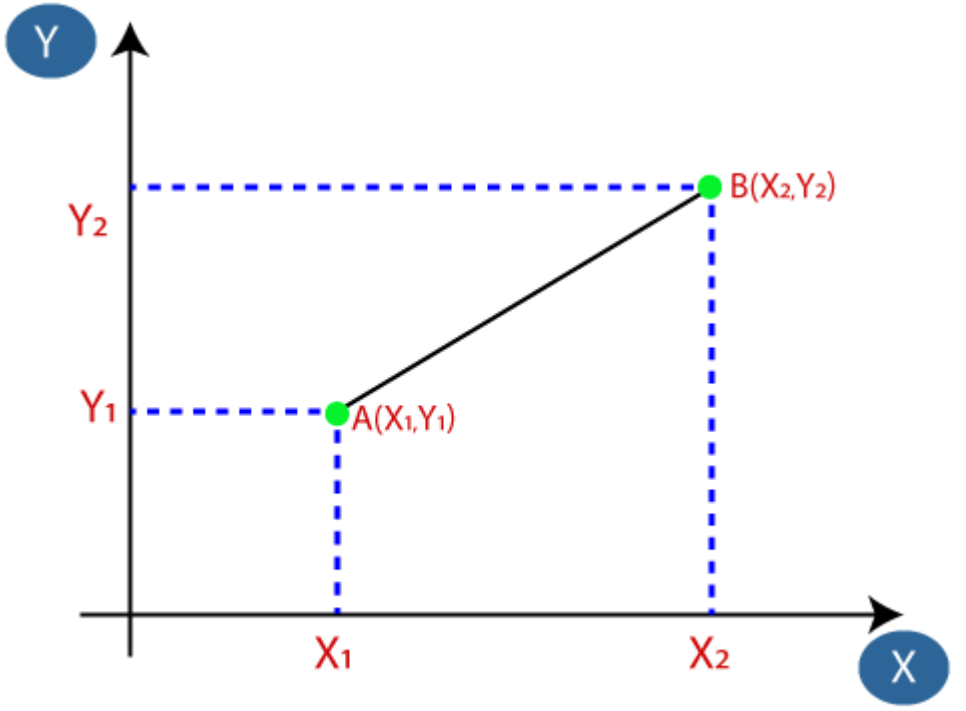
- Suppose we have a new data point, and we need to put it in the required category.



Firstly, we will choose the number of neighbors, so we will choose the k=5.

How K-NN Works? Step 2

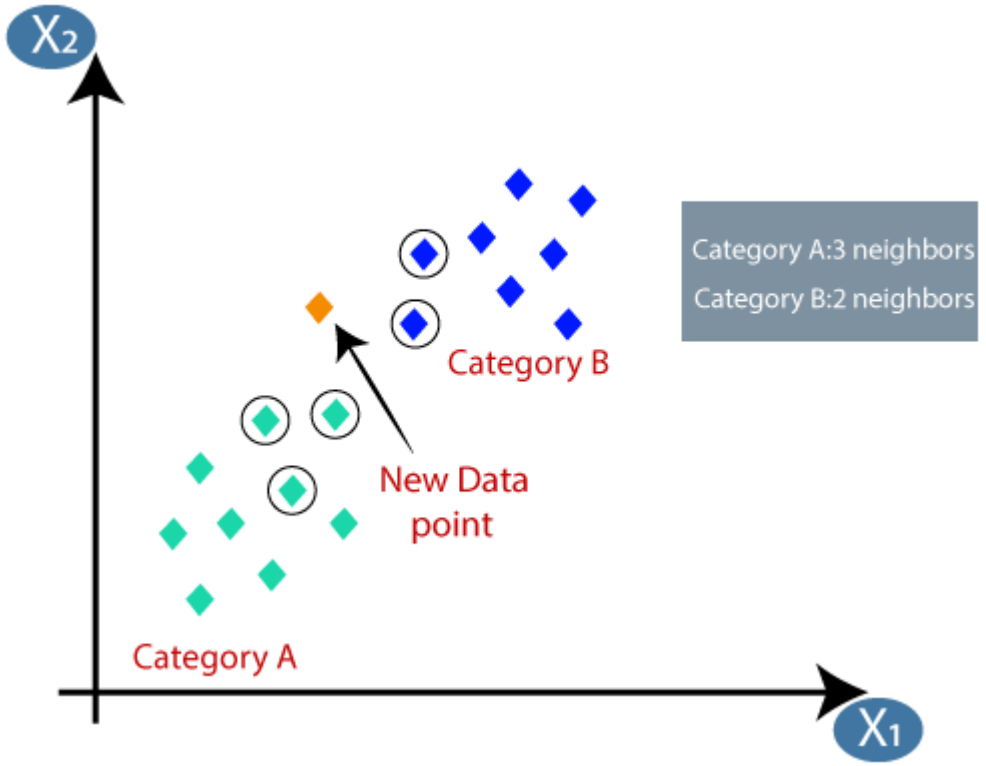
- Next, we will calculate the Euclidean distance between the data points.
- The Euclidean distance is the distance between two points.



Euclidean Distance between A_1 and $B_2 = \sqrt{(X_2-X_1)^2+(Y_2-Y_1)^2}$

How K-NN Works? Step 3

- By calculating the Euclidean distance, we got the nearest neighbors.
- Three nearest neighbors in category A and two nearest neighbors in category B.
- As we can see the 3 nearest neighbors are from category A, hence this new data point must belong to category A.



How to Select the Value of K?

- There is no way to determine the best value for “K”.
- So, we need to try some values to find the best out of them. The most preferred value for K is 5.
- A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.
- Large values for K are good, but it may find some difficulties.

Pros & Cons of K-NN

- Pros
  - It is simple to implement.
  - It is robust to the noisy training data.
  - It can be more effective if the training data is large.
- Cons
  - Always needs to determine the value of K which may be complex some time.
  - The computation cost is high because of calculating the distance between the data points for all the training samples.

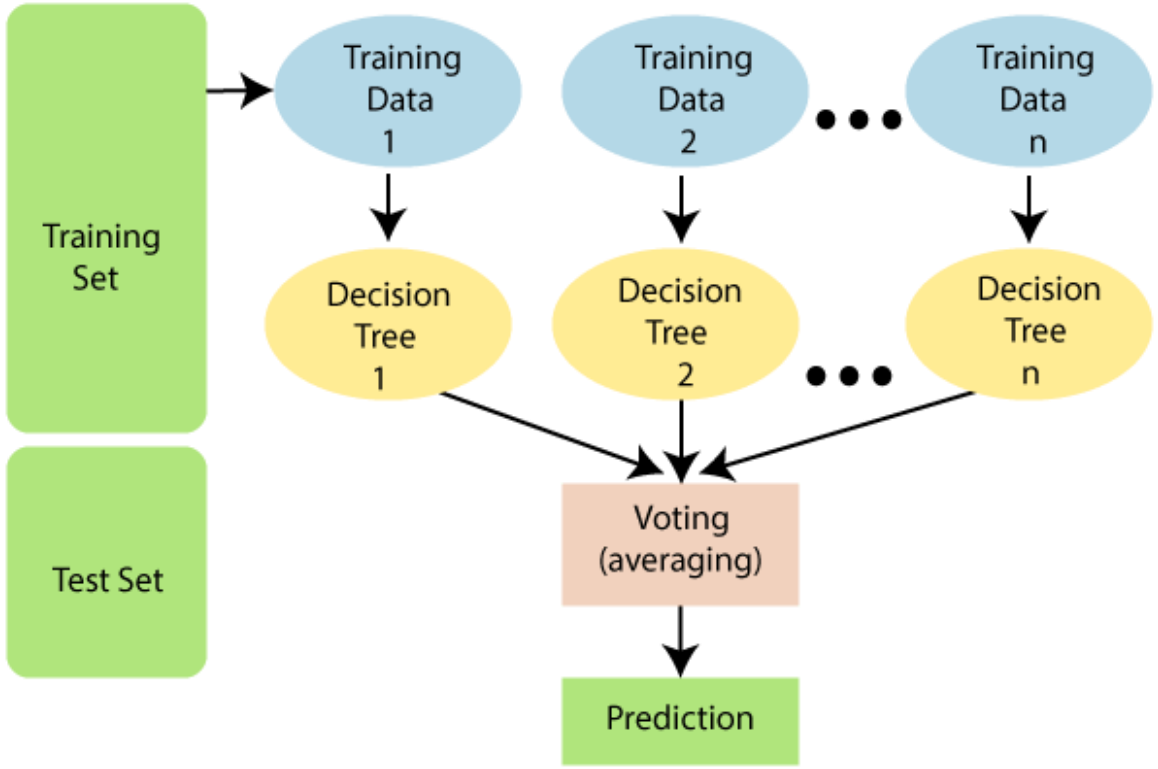
Random Forest

Random Forest

- Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique.
- It can be used for both Classification and Regression problems in ML.
- It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem.
- Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset.

Working Model of Random Forest

- It is possible that some decision trees may predict the correct output, while others may not.
- But together, all the trees predict the correct output.
- There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results.
- The predictions from each tree must have very low correlations.



Why Random Forest?

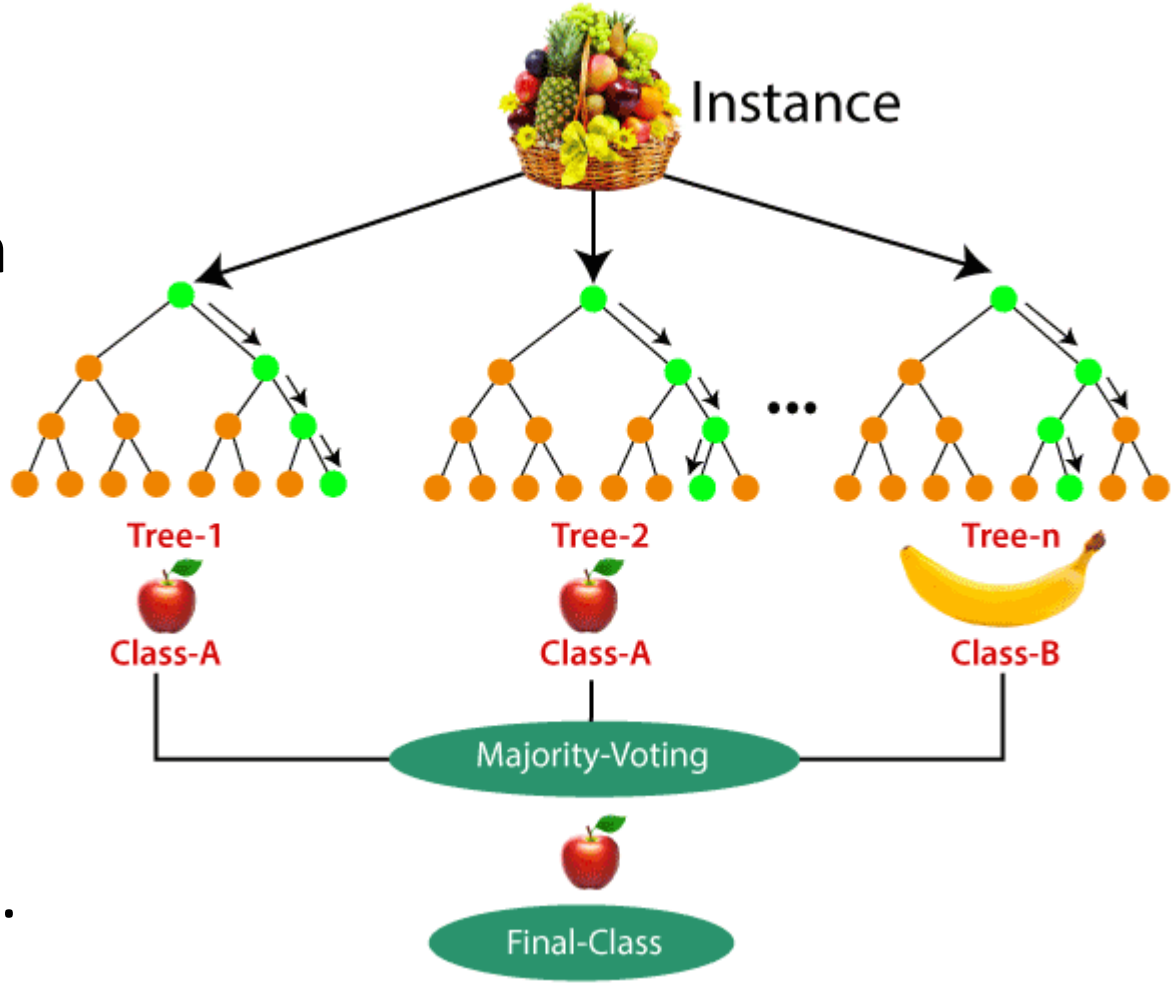
- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.

How Random Forest Works?

- Step-1: Select random K data points from the training set.
- Step-2: Build the decision trees associated with the selected data points.
- Step-3: Choose the number N for decision trees that you want to build.
- Step-4: Repeat Step 1 & 2.
- Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

How Random Forest Works?

- Suppose there is a dataset that contains multiple fruit images. So, this dataset is given to the Random forest classifier. The dataset is divided into subsets and given to each decision tree. During the training phase, each decision tree produces a prediction result, and when a new data point occurs, then based on the majority of results, the Random Forest classifier predicts the final decision.



Applications of Random Forest

- Banking: Banking sector mostly uses this algorithm for the identification of loan risk.
- Medicine: With the help of this algorithm, disease trends and risks of the disease can be identified.
- Land Use: We can identify the areas of similar land use by this algorithm.
- Marketing: Marketing trends can be identified using this algorithm.

Pros & Cons of Random Forest

- Pros
  - It can perform both Classification and Regression tasks.
  - It is capable of handling large datasets with high dimensionality.
  - It enhances the accuracy of the model and prevents the overfitting issue.
- Cons
  - Although random forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks.

Questions