PRESIDENCY UNIVERSITY

BENGALURU

MODULE 3

ADVANCED MACHINE LEARNING CONCEPTS

PRESIDENCY UNIVERSITY

Private University Estd. in Karnataka State by Act No. 41 of 2013

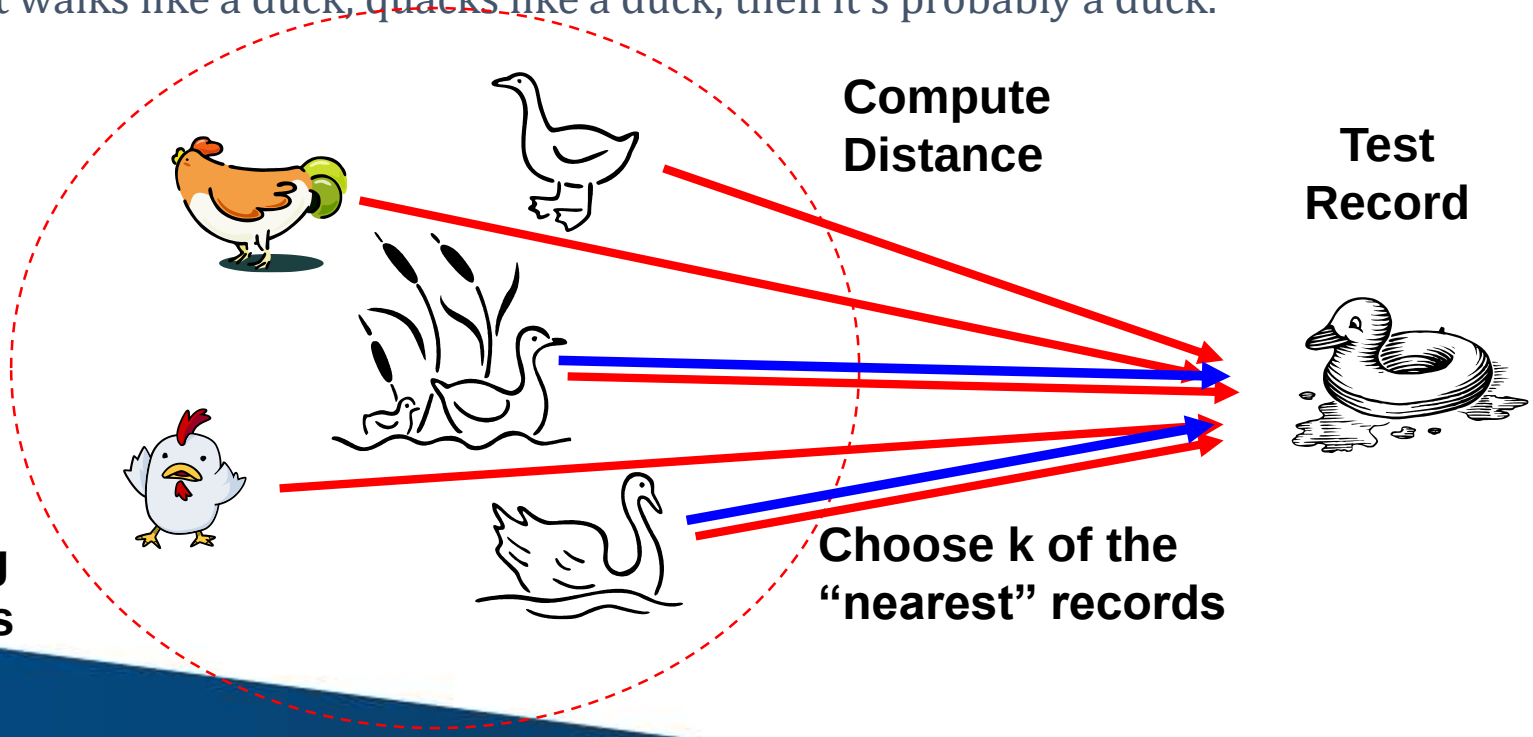
40 YEARS OF ACADEMIC WISDOM

CONTENT

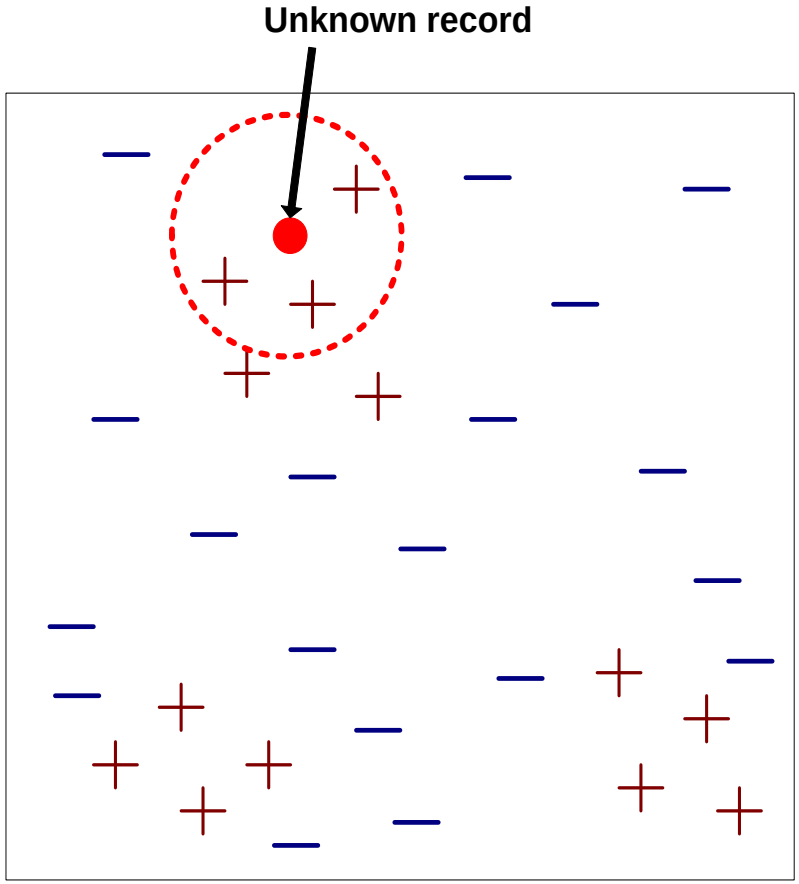
- Nearest Neighbor techniques
  - KNN
- Cost functions and Optimization technique
  - Introduction to gradient descent
  - Applications on Linear Regression
- Ensemble Learning Algorithms
  - Bagging(Random Forest)
  - Boosting(Ada Boost)
  - XG Boost

NEAREST NEIGHBOR TECHNIQUES

- Nearest Neighbor Analysis is a method for classifying cases based on their similarity to other cases.
- In machine learning, it was developed as a way to recognize patterns of data without requiring an exact match to any stored patterns, or cases.
- If it walks like a duck, quacks like a duck, then it's probably a duck.



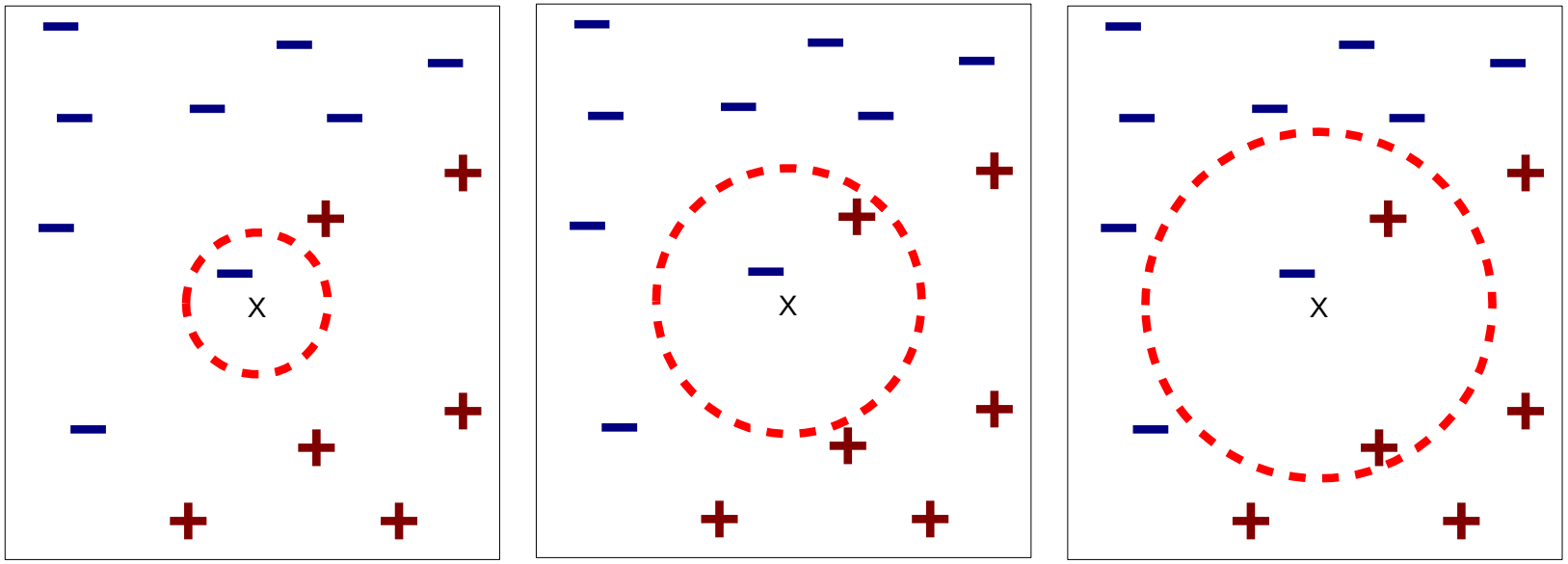
NEAREST NEIGHBOR TECHNIQUES



- Requires three things
  - The set of stored records
  - Distance Metric to compute distance between records
  - The value of *k*, the number of nearest neighbors to retrieve

- To classify an unknown record:
  - Compute distance to other training records
  - Identify *k* nearest neighbors
  - Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)

NEAREST NEIGHBOR TECHNIQUES



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

K-nearest neighbors of a record x are data points that have the k smallest distance to x

NEAREST NEIGHBOR TECHNIQUES

- Compute distance between two points:
  - Euclidean distance

$$d(p,q) = \sqrt{\sum_i (p_i - q_i)^2}$$

  - Manhatten distance

$$d(p,q) = \sum_i |p_i - q_i|$$

  - q norm distance

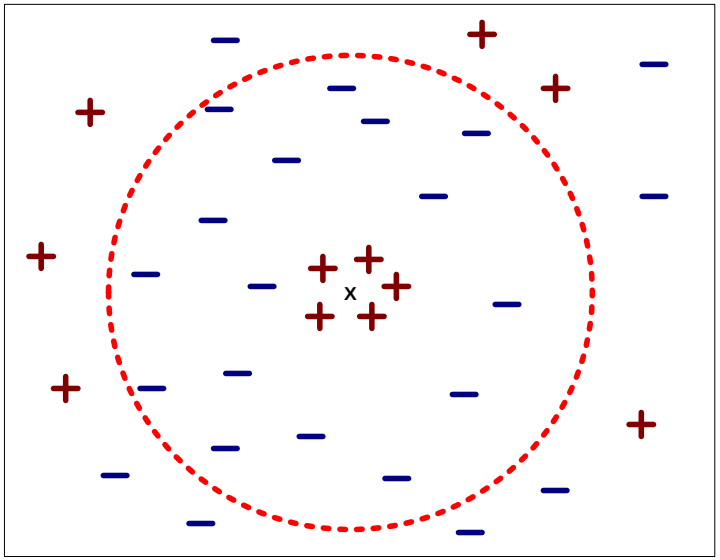
$$d(p,q) = (\sum_i |p_i - q_i|^q)^{1/q}$$

K NEAREST NEIGHBOR CLASSIFICATION

- Compute distance between two points:
  - Euclidean distance

$$d(p,q) = \sqrt{\sum_i (p_i - q_i)^2}$$

- Determine the class from nearest neighbor list
  - take the majority vote of class labels among the k-nearest neighbors
  - Weigh the vote according to distance
    - weight factor, $w = 1/d^2$
- Choosing the value of k:
  - If k is too small, susceptible to over fitting, due to noise points in the training data.
  - If k is too large, neighborhood may include points from other classes.



K NEAREST NEIGHBOR CLASSIFICATION

- Scaling issues
  - Attributes may have to be scaled to prevent distance measures from being dominated by one of the attributes
  - Example:
    - height of a person may vary from 1.5m to 1.8m
    - weight of a person may vary from 90lb to 300lb
    - income of a person may vary from $10K to $1M
  - Solution: Normalize the vectors to unit length

K NEAREST NEIGHBOR CLASSIFICATION

1. Let k be the no. of nearest neighbors and $\boldsymbol{D}$ be the set of training examples.

2. for each test example z = (x', y') do

2.1 compute $d(x', x)$, the distance between z and every example $(x,y) \in \boldsymbol{D}$.

2.2 Select $\boldsymbol{D_z} \subseteq \boldsymbol{D}$, the set of k closest training examples to z.

2.3 $$y' = \underset{v}{\boldsymbol{argmax}} \sum_{(x_i, y_i) \in D_z} \boldsymbol{I}(v = y_i)$$

2.4 end for

K NEAREST NEIGHBOR CLASSIFICATION

- Example: The table represents our data set. We have two columns — **Brightness** and **Saturation**. Each row in the table has a class of either **Red** or **Blue**.

BRIGHTNESS	SATURATION	COLOR
40	20	RED
50	50	BLUE
60	90	BLUE
10	25	RED
70	70	BLUE
60	10	RED
25	80	BLUE
20	35	?

K NEAREST NEIGHBOR CLASSIFICATION

$d1 = \sqrt{(20 - 40)^2 + (35 - 20)^2}$
$= \sqrt{400 + 225}$
$= \sqrt{625}$
$= 25$

$d2 = \sqrt{(20 - 50)^2 + (35 - 50)^2}$
$= \sqrt{900 + 225}$
$= \sqrt{1125}$
$= 33.54$

$d3 = \sqrt{(20 - 60)^2 + (35 - 90)^2}$
$= \sqrt{1600 + 3025}$
$= \sqrt{4625}$
$= 68.01$

$d4 = \sqrt{(20 - 10)^2 + (35 - 25)^2}$
$= \sqrt{100 + 100}$
$= \sqrt{200}$
$= 14.14$

$d5 = \sqrt{(20 - 70)^2 + (35 - 70)^2}$
$= \sqrt{2500 + 1225}$
$= \sqrt{3725}$
$= 61.03$

$d6 = \sqrt{(20 - 60)^2 + (35 - 10)^2}$
$= \sqrt{1600 + 625}$
$= \sqrt{2225}$
$= 47.17$

$d7 = \sqrt{(20 - 25)^2 + (35 - 80)^2}$
$= \sqrt{25 + 2025}$
$= \sqrt{2050}$
$= 45.27$

If k= 5, then distance for 1st five nearest value should be noted -> majority Color be red. So, For Brightness -> 20, Saturation -> 35, Color be **RED**

K NEAREST NEIGHBOR CLASSIFICATION

BRIGHTNESS	SATURATION	COLOR	DISTANCE
40	20	RED	25
50	50	BLUE	33.54
60	90	BLUE	68.01
10	25	RED	14.14
70	70	BLUE	61.03
60	10	RED	47.17
25	80	BLUE	45.27
20	35	RED	

K NEAREST NEIGHBOR CLASSIFICATION

Strengths	Weaknesses
• Simple and effective • Makes no assumptions about the underlying data distribution • Fast training phase	• Does not produce a model, which limits the ability to find novel insights in relationships among features • Slow classification phase • Requires a large amount of memory • Nominal features and missing data require additional processing

COST FUNCTIONS AND OPTIMIZATION TECHNIQUE

- A Machine Learning model should have a very high level of accuracy in order to perform well with real-world applications. But how to calculate the accuracy of the model, i.e., how good or poor our model will perform in the real world?
- In such a case, the Cost function comes into existence.
- It is an important machine learning parameter to correctly estimate the model.
- ***A cost function is an important parameter that determines how well a machine learning model performs for a given dataset.*** It calculates the difference between the expected value and predicted value and represents it as a single real number.
- "***Cost function is a measure of how wrong the model is in estimating the relationship between X(input) and Y(output) Parameter***." A cost function is sometimes also referred to as Loss function.
- By minimizing the value of the cost function, we can get the optimal solution.

GRADIENT DESCENT: MINIMIZING THE COST FUNCTION

- "***Gradient Descent is an optimization algorithm which is used for optimizing the cost function or error in the model.***"
- It enables the models to take the gradient or direction to reduce the errors by reaching to least possible error.
- Gradient descent is an iterative process where the model gradually converges towards a minimum value, and if the model iterates further than this point, it produces little or zero changes in the loss. This point is known as convergence, and at this point, the error is least, and the cost function is optimized.

$$\theta_j = \theta_j - \alpha \frac{\partial J}{\partial \theta}$$

- In the gradient descent equation, alpha is known as the learning rate.

GRADIENT DESCENT: MINIMIZING THE COST FUNCTION

Types of Cost Function

Cost functions can be of various types depending on the problem.

- *Regression Cost Function:* Regression models are used to make a prediction for the continuous variables such as the price of houses, weather prediction, loan predictions, etc. When a cost function is used with Regression, it is known as the "Regression Cost Function."

  Error= Actual Output-Predicted output

  Three commonly used Regression cost functions

  1. **Means Error**
  2. **Mean Squared Error(MSE)**
  3. **Mean Absolute Error(MAE)**

- *Binary Classification cost Functions:* Classification models are used to make predictions of categorical variables, such as predictions for 0 or 1, Cat or dog, etc. The cost function used in the classification problem is known as the Classification cost function.
- *Multi-class Classification Cost Function:* A multi-class classification cost function is used in the classification problems for which instances are allocated to one of more than two classes.

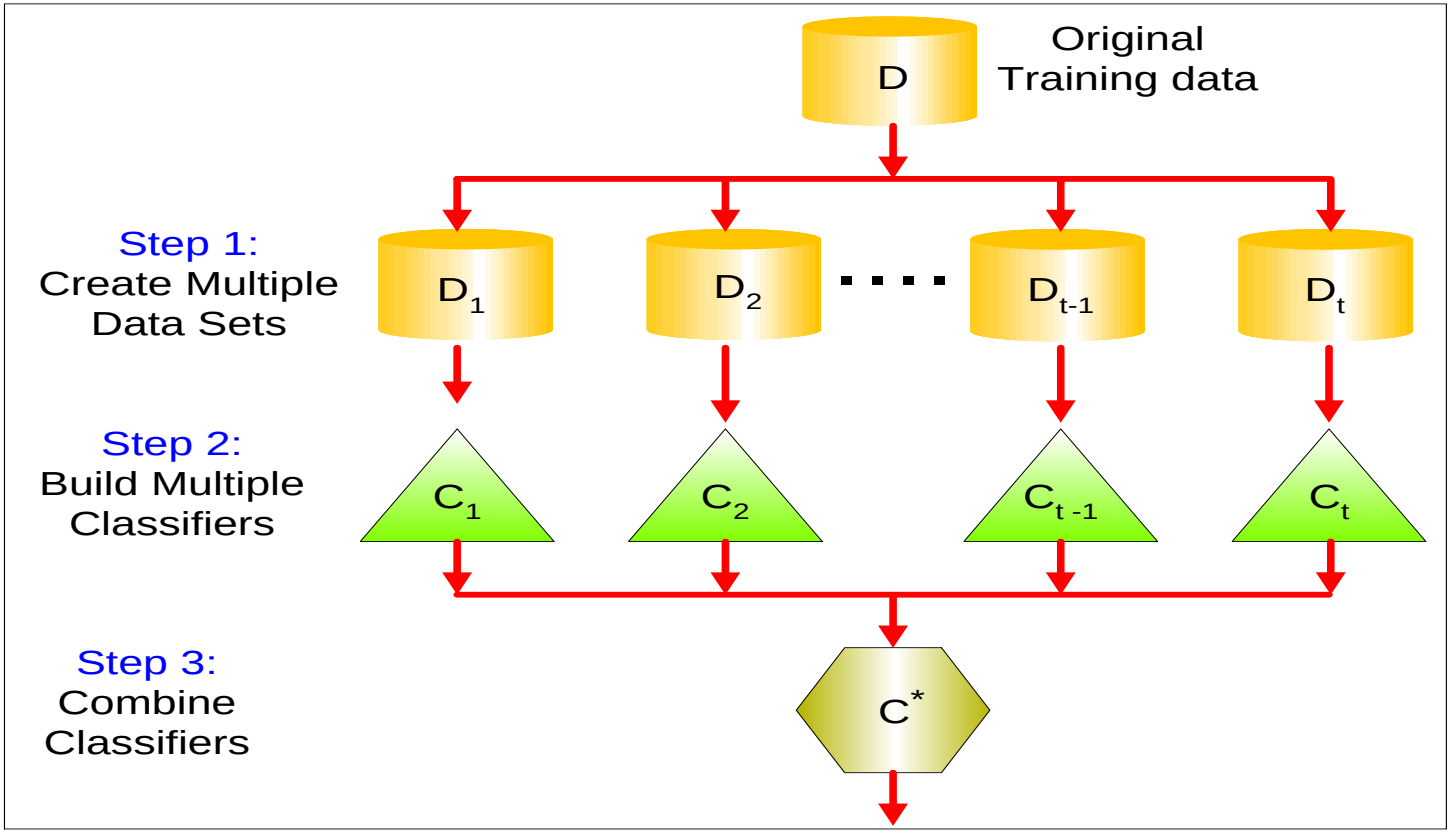
GRADIENT DESCENT: APPLICATIONS ON LINEAR REGRESSION

- Gradient Descent is an algorithm that finds the best-fit line for a given training dataset in a smaller number of iterations.
- A Linear Regression model converging to optimum solution using Gradient Descent.
- Gradient Descent is used for multiple linear regression as an iterative algorithm use in loss function to find the global minima. The loss can be any differential loss function.
- **Real time applications:** Calculate the gradients of slopes in and around school, like staircases and ramps.

ENSEMBLE LEARNING ALGORITHM

- Ensemble means combining multiple models.
- Predict class label of previously unseen records by aggregating predictions made by multiple classifiers
- Improves the classification accuracy
- Predicted output of the base classifiers is combined by majority voting
- Build different experts and let them vote.
- Build many models and combine them.
- Only through averaging do we get at the truth!
- It's too hard (*impossible*?) to build a single model that works best
- Two types of approaches:
  - Models that don't use randomness
  - Models that incorporate randomness

ENSEMBLE LEARNING ALGORITHM



BAGGING(BOOTSTRAP AGGREGATION)

- Main Assumption:
  - Combining many unstable predictors to produce a ensemble (stable) predictor.
  - Unstable Predictor: small changes in training data produce large changes in the model.
    - e.g. Neural Nets, trees
    - Stable: SVM (sometimes), Nearest Neighbor.
- Hypothesis Space
  - Variable size (nonparametric):
    - Can model any function if you use an appropriate predictor (e.g. trees)

BAGGING ALGORITHM

Given data: $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$

For $m = 1 : M$

- Obtain bootstrap sample D_m from the training data D
- Build a model $G_m(\mathbf{x})$ from bootstrap data D_m

- Regression

$$\hat{y} = \frac{1}{M} \sum_{m=1}^{M} G_m(\mathbf{x})$$

- Classification:
  - Vote over classifier outputs $G_1(\mathbf{x}), \ldots, G_M(\mathbf{x})$

BAGGING – RANDOM FOREST ALGORITHM

- Random Forest Algorithm widespread popularity stems from its user-friendly nature and adaptability, enabling it to tackle both classification and regression problems effectively.
- Random Forest Algorithm is that it can handle the data set containing ***continuous variables,*** as in the case of regression, and ***categorical variables,*** as in the case of classification. It performs better for classification and regression tasks.
- ***Real Time application:*** A student named X wants to choose a course after his 10+2, and he is confused about the choice of course based on his skill set. So he decides to consult various people like his cousins, teachers, parents, degree students, and working people. He asks them varied questions like why he should choose, job opportunities with that course, course fee, etc. Finally, after consulting various people about the course he decides to take the course suggested by most people.

BAGGING – RANDOM FOREST ALGORITHM

Steps Involved in Random Forest Algorithm

Step 1: In the Random forest model, a subset of data points and a subset of features is selected for constructing each decision tree. Simply put, n random records and m features are taken from the data set having k number of records.

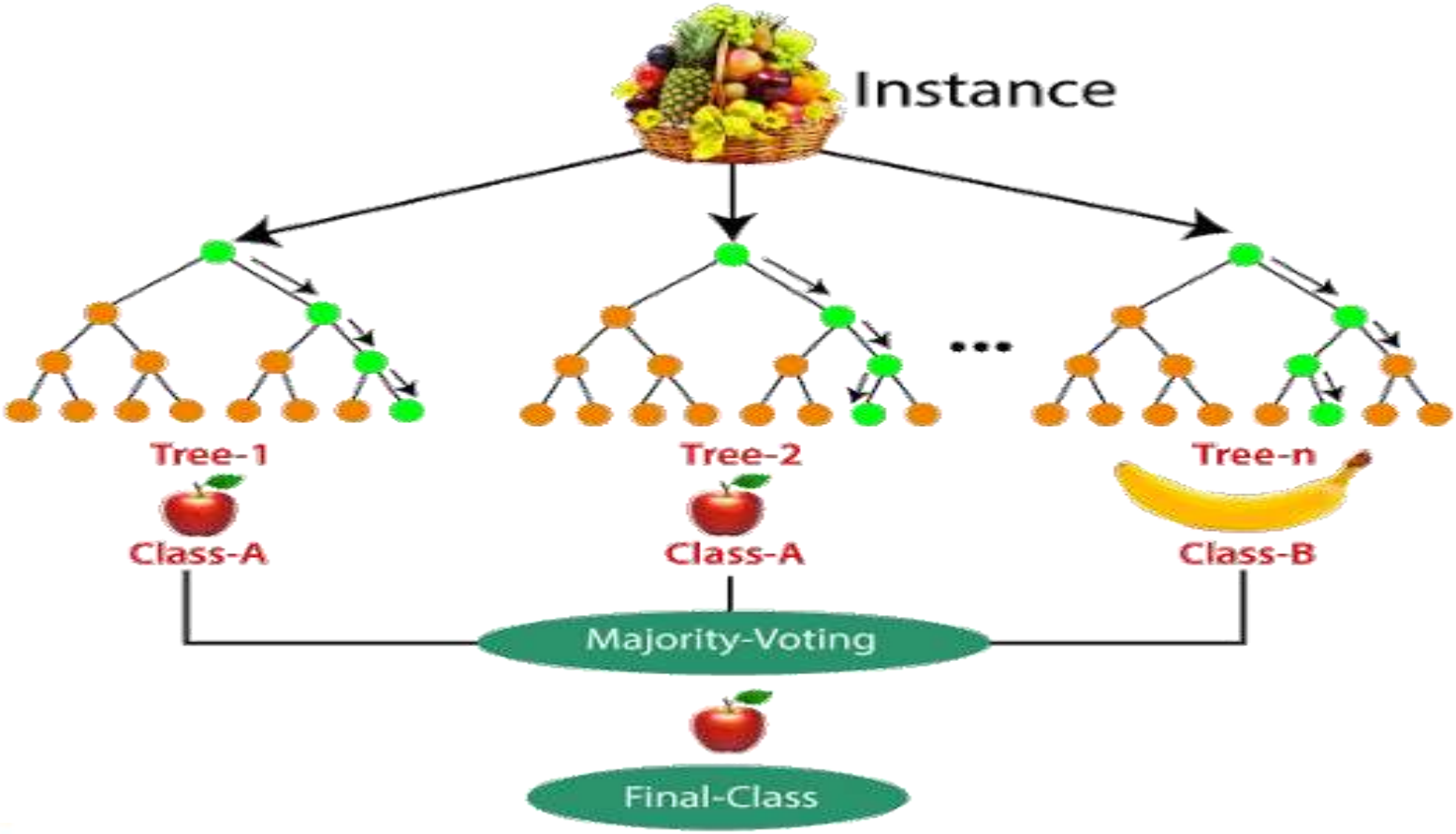
Step 2: Individual decision trees are constructed for each sample.

Step 3: Each decision tree will generate an output.

Step 4: Final output is considered based on ***Majority Voting or Averaging*** for Classification and regression, respectively.

Example: Consider the fruit basket as the data as shown in the figure below. Now n number of samples are taken from the fruit basket, and an individual decision tree is constructed for each sample. Each decision tree will generate an output, as shown in the figure. The final output is considered based on majority voting. In the below figure, you can see that the majority decision tree gives output as an apple when compared to a banana, so the final output is taken as an apple.

BAGGING – RANDOM FOREST ALGORITHM



ADVANTAGES & DISADVANTAGES

ADVANTAGES

- It reduces over fitting in decision trees and helps to improve the accuracy
- It is flexible to both classification and regression problems
- It works well with both categorical and continuous values
- It automates missing values present in the data
- Normalizing of data is not required as it uses a rule-based approach.

DISADVANTAGES

- It requires much computational power as well as resources as it builds numerous trees to combine their outputs.
- It also requires much time for training as it combines a lot of decision trees to determine the class.
- Due to the ensemble of decision trees, it also suffers interpretability and fails to determine the significance of each variable.

BOOSTING

- **Boosting** is an ensemble modeling technique that attempts to build a strong classifier from the number of weak classifiers.
- It is done by building a model by using weak models in series. Firstly, a model is built from the training data. Then the second model is built which tries to correct the errors present in the first model.
- This procedure is continued and models are added until either the complete training data set is predicted correctly or the maximum number of models are added.

TRAINING THE BOOSTING MODEL

- *Initialise the dataset and assign equal weight to each of the data point.*
- *Provide this as input to the model and identify the wrongly classified data points.*
- *Increase the weight of the wrongly classified data points.*
- *if (got required results)*
  Goto step 5
  else
  Goto step 2
- *End*

ADVANTAGES & DISADVANTAGES

- ***Improved Accuracy*** – Boosting can improve the accuracy of the model by combining several weak models' accuracies and averaging them for regression or voting over them for classification to increase the accuracy of the final model.
- ***Robustness to Over fitting*** – Boosting can reduce the risk of over fitting by reweighting the inputs that are classified wrongly.
- ***Better handling of imbalanced data*** – Boosting can handle the imbalance data by focusing more on the data points that are misclassified
- ***Better Interpretability*** – Boosting can increase the interpretability of the model by breaking the model decision process into multiple processes.

DISADVANTAGES

- Boosting Algorithms are vulnerable to the outliers
- It is difficult to use boosting algorithms for Real-Time applications.
- It is computationally expensive for large datasets

ADA BOOST

- AdaBoost is a boosting algorithm that also works on the principle of the stagewise addition method where multiple weak learners are used for getting strong learners.
- The value of the alpha parameter, in this case, will be indirectly proportional to the error of the weak learner, Unlike Gradient Boosting in XGBoost, the alpha parameter calculated is related to the errors of the weak learner, here the value of the alpha parameter will be indirectly proportional to the error of the weak learner.
- **ADVANTAGES:** The flexible AdaBoost can also be used for accuracy improvement of weak classifiers and cases in image/text classification. **DISADVANTAGES:** AdaBoost uses a progressively learning boosting technique.

XGBOOST

- In addition to the gradient boosting technique, XG Boost is another boosting machine learning approach.
- The full name of the XG Boost algorithm is the eXtreme Gradient Boosting algorithm, which is an extreme variation of the previous gradient boosting technique.
- The key distinction between XG Boost and Gradient Boosting is that XGBoost applies a regularization approach.
- It is a regularized version of the current gradient-boosting technique. Because of this, XGBoost outperforms a standard gradient boosting method, which explains why it is also faster than that.
- Additionally, it works better when the dataset contains both numerical and categorical variables.
- **ADVANTAGES :** It makes it easy to scale up on multicore machines or clusters.
- **DISADVANTAGES:** **It** can over-fit the data, especially if the trees are too deep with noisy data.

THANK YOU