

UNIT-4 (8Marksand16Marks)

1. Explain the various ensemble learning techniques?

Ensemble methods are techniques that aim at improving the accuracy of results in models by combining multiple models instead of using a single model. The combined models increase the accuracy of the results significantly. This has boosted the popularity of ensemble methods in machine learning.

Categories of Ensemble Methods

Ensemble methods fall into two broad categories, i.e., sequential ensemble techniques and parallel ensemble techniques. **Sequential ensemble techniques** generate base learners in a sequence, e.g., Adaptive Boosting (AdaBoost). The sequential generation of base learners promotes the dependence between the base learners. The performance of the model is then improved by assigning higher weights to previously misrepresented learners.

- In **parallel ensemble techniques**, base learners are generated in a parallel format, e.g., random forest. Parallel methods utilize the parallel generation of base learners to encourage independence between the base learners. The independence of base learners significantly reduces the error due to the application of averages.
- The majority of ensemble techniques apply a single algorithm in base learning, which results in homogeneity in all base learners. Homogenous base learners refer to base learners of the same type, with similar qualities. Other methods apply heterogeneous base learners, giving rise to heterogeneous ensembles. Heterogeneous base learners are learners of distinct types.

Main Types of Ensemble Methods

1. Bagging

- Bagging, the short form for bootstrap aggregating, is mainly applied in classification and regression. It increases the accuracy of models through decision trees, which reduces variance to a large extent. The reduction of variance increases accuracy, eliminating overfitting, which is a challenge to many predictive models.
- Bagging is classified into two types, i.e., bootstrapping and aggregation. **Bootstrapping** is a sampling technique where samples are derived from the whole population (set) using the replacement procedure. The sampling with replacement method helps make the selection procedure randomized. The base learning algorithm is run on the samples to complete the procedure.
- **Aggregation** in bagging is done to incorporate all possible outcomes of the prediction and randomize the outcome. Without aggregation, predictions will not be accurate because all outcomes are not put into consideration. Therefore, the aggregation is based on the probability bootstrapping procedures or on the basis of all outcomes of the predictive models.

Bagging is advantageous since weak base learners are combined to form a single strong learner that is more stable than single learners. It also eliminates any variance, thereby reducing the overfitting of models. One limitation of bagging is that it is computationally expensive. Thus, it can lead to more bias in models when the proper procedure of bagging is ignored.

2. Boosting

- Boosting is an ensemble technique that learns from previous predictor mistakes to make better predictions in the future. The technique combines several weak base learners to form one strong learner, thus significantly improving the predictability of models. Boosting works by arranging weak learners in a sequence, such that weak learners learn from the next learner in the sequence to create better predictive models.
- Boosting takes many forms, including gradient boosting, Adaptive Boosting (AdaBoost), and XGBoost (Extreme Gradient Boosting). AdaBoost uses weak learners in the form of decision trees, which mostly include one split that is popularly known as decision stumps. AdaBoost's main decision stump comprises observations carrying similar weights.
- Gradient boosting adds predictors sequentially to the ensemble, where preceding predictors correct their successors, thereby increasing the model's accuracy. New predictors are fit to counter the effects of errors in the previous predictors. The gradient of descent helps the gradient booster identify problems in learners' predictions and counter them accordingly.

3. Stacking

Stacking, another ensemble method is often referred to as stacked generalization. This technique works by allowing a training algorithm to ensemble several other similar learning algorithm predictions. Stacking has been successfully implemented in regression, density estimations, distance learning, and classifications. It can also be used to measure the error rate involved during bagging.

Variance Reduction

Ensemble methods are ideal for reducing the variance in models, thereby increasing the accuracy of predictions. The variance is eliminated when multiple models are combined to form a single prediction that is chosen from all other possible predictions from the combined models. An ensemble of models combines various models to ensure that the resulting prediction is the best possible, based on the consideration of all predictions.

Simple Ensemble Techniques

In this section, we will look at a few simple but powerful techniques, namely:

1. Max Voting
2. Averaging
3. Weighted Averaging

2. Explain in detail about k-means algorithm?

K-Means Clustering Algorithm

K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science. In this topic, we will learn what is K-means clustering algorithm, how the algorithm works, along with the Python implementation of k-means clustering.

What is K-Means Algorithm?

K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in

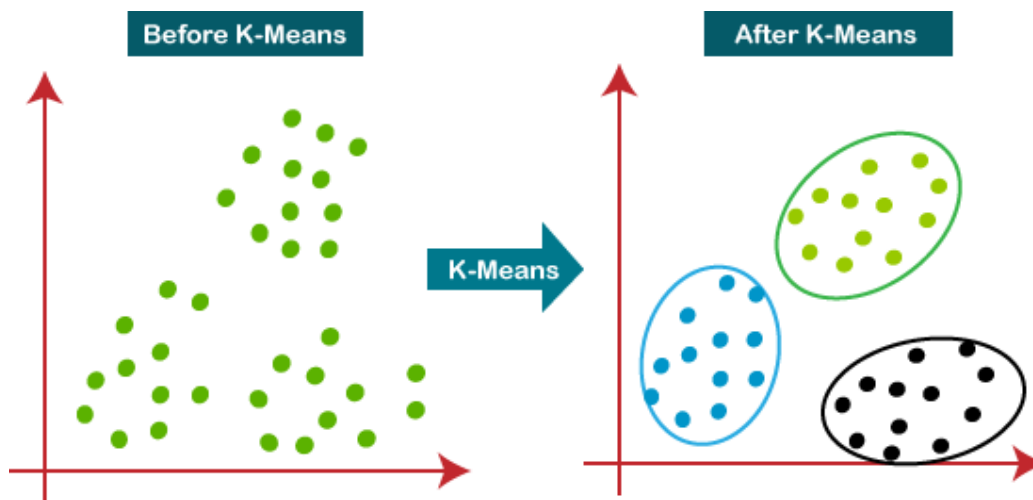
the process, as if $K=2$, there will be two clusters, and for $K=3$, there will be three clusters, and so on.

- It is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs only one group that has similar properties.
- It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.
- It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.
- The algorithm takes the unlabeled dataset as input, divides the dataset into k -number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm

The k -means clustering algorithm mainly performs two tasks:

- Determines the best value for K center points or centroids by an iterative process.
- Assign each data point to its closest k -center. Those data points which are near to the particular k -center, create a cluster.

Hence each cluster has data points with some commonalities, and it is away from other clusters. The below diagram explains the working of the K -means Clustering Algorithm:



How does the K-Means Algorithm Work?

The working of the K -Means algorithm is explained in the below steps:

Step-1: Select the number K to decide the number of clusters.

Step-2: Select random K points or centroids. (It can be other from the input dataset).

Step-3: Assign each data point to their closest centroid, which will form the predefined K clusters.

Step-4: Calculate the variance and place a new centroid of each cluster.

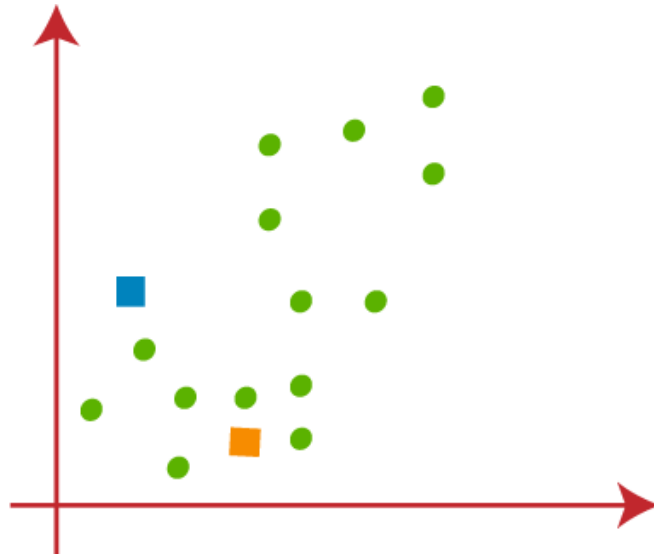
Step-5: Repeat the third steps, which mean reassign each datapoint to the new closest centroid of each cluster.

Step-6: If any reassignment occurs, then goto step-4 else go to FINISH.

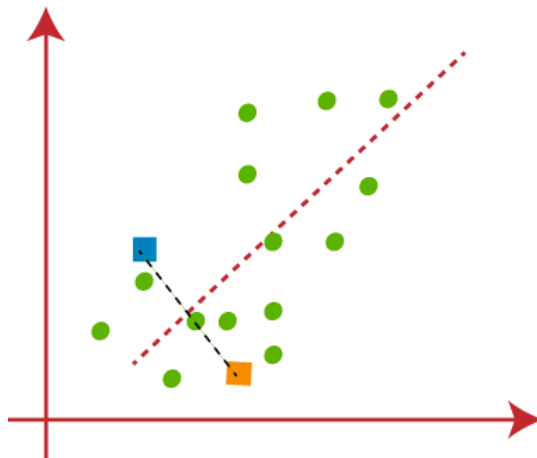
Step-7: The model is ready.

Let's understand the above steps by considering the visual plots:

- Suppose we have two variables M1 and M2. The x-y axis scatter plot of these two variables is given below: Let's take number k of clusters, i.e., $K=2$, to identify the dataset and to put them into different clusters. It means here we will try to group these datasets into two different clusters.
- We need to choose some random k points or centroid to form the cluster. These points can be either the points from the dataset or any other point. So, here we are selecting the below two points as k points, which are not the part of our dataset. Consider the below image:



Now we will assign each data point of the scatter plot to its closest K-point or centroid. We will compute it by applying some mathematics that we have studied to calculate the distance between two points. So, we will draw a median between both the centroids. Consider the below image:



From the above image, it is clear that points left side of the line is near to the K1 or blue centroid, and points to the right of the line are close to the yellow centroid. Let's color them as blue and yellow for clear visualization. As we need to find the closest cluster, so we will repeat the process by choosing a **new centroid**. To choose the new centroids, we will compute the center of gravity of these centroids, and will find new centroids as below: Next, we will reassign each datapoint to the new centroid. For this, we will repeat the same process of finding a median line.

How to choose the value of "K number of clusters" in K-means Clustering?

The performance of the K-means clustering algorithm depends upon highly efficient clusters that it forms. But choosing the optimal number of clusters is a big task. There are some different ways to find the optimal number of clusters, but here we are discussing the most appropriate method to find the number of clusters or value of K. The method is given below:

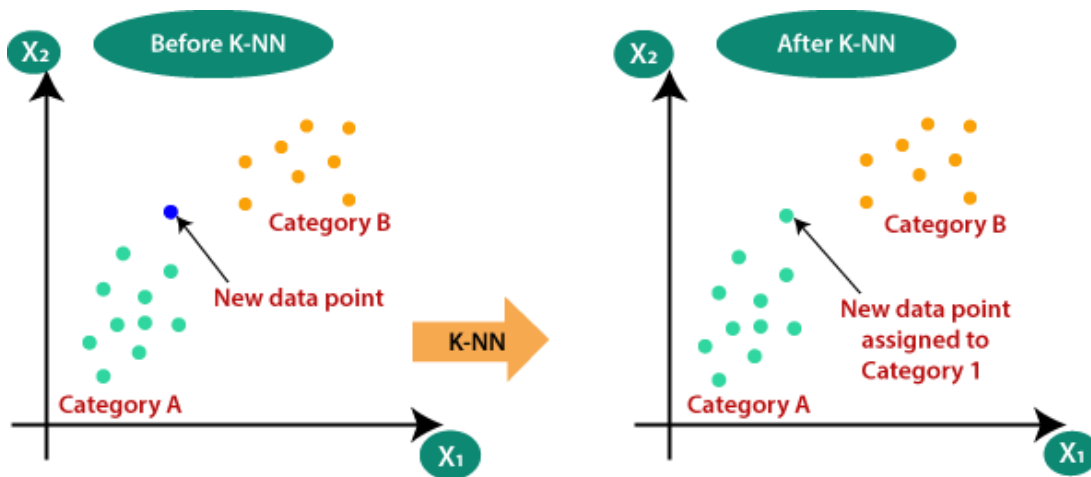
3. Explain details about KNN algorithm?

K-Nearest Neighbor (KNN) Algorithm for Machine Learning

- K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and puts 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. This means when new data appears then it can be easily classified into a well suited category by using K-NN algorithm.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
- It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is most similar to the new data.
- **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.

Why do we need a K-NN Algorithm?

Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x_1 , 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:

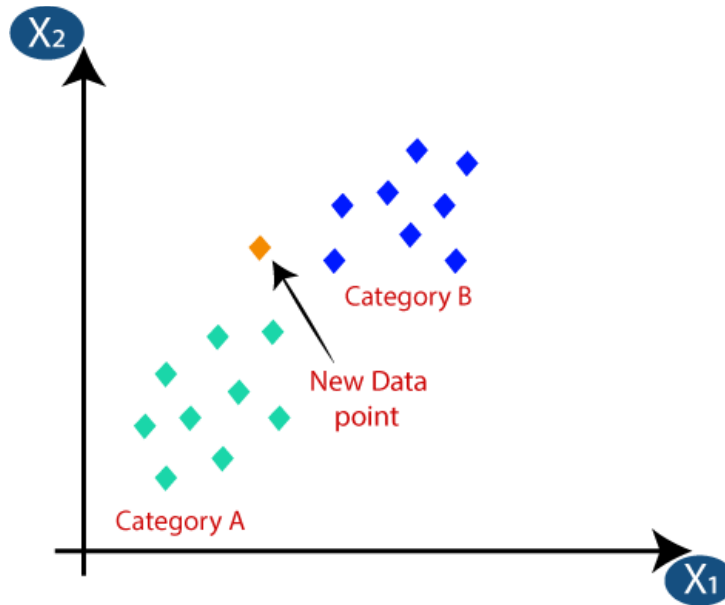


How does K-NN work?

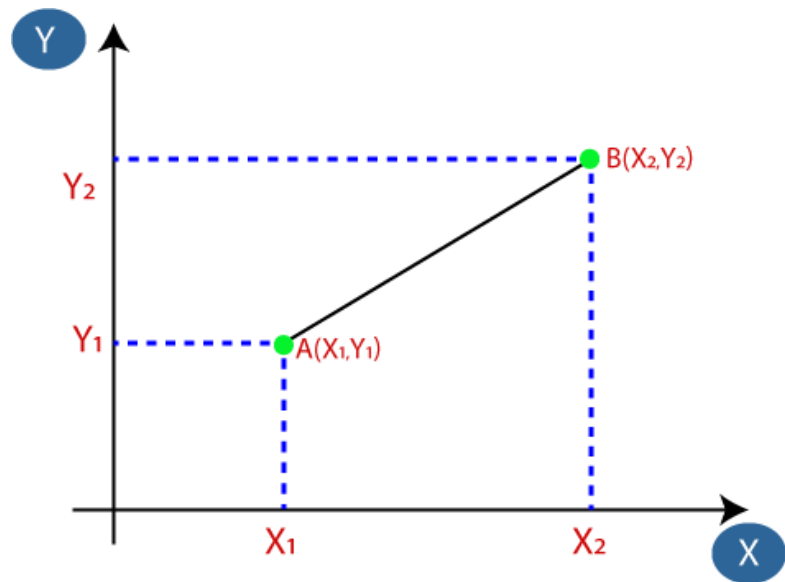
The K-NN working can be explained on the basis of the below algorithm:

- **Step-1:** Select the number K of the neighbors
- **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
- **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
- **Step-4:** Among these k neighbors, count the number of the datapoints in each category.
- **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
- **Step-6:** Our model is ready.

Suppose we have a new data point and we need to put it in the required category. Consider the below image:



- Firstly, we will choose the number of neighbors, so we will choose the $k=5$.
- Next, we will calculate the **Euclidean distance** between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry. It can be calculated as:



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

- By calculating the Euclidean distance we got the nearest neighbors, as three nearest neighbors in category A and two nearest neighbors in category B. Consider the below image:



- As we can see the 3 nearest neighbors are from category A, hence this new datapoint must belong to category A.

How to select the value of K in the K-NN Algorithm?

Below are some points to remember while selecting the value of K in the K-NN algorithm:

- There is no particular 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.

Advantages of KNN Algorithm:

- It is simple to implement.
- It is robust to the noisy training data
- It can be more effective if the training data is large.

Disadvantages of KNN Algorithm:

- 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.

4. Explain in detail about Gaussian mixture models and expectation maximization?

EM algorithm in GMM

In statistics, EM (expectation maximization) algorithm handles latent variables, while GMM is the Gaussian mixture model.

- ✓ Gaussian mixture models (GMMs) are a type of machine learning algorithm. They are used to classify data into different categories based on the probability distribution. Gaussian mixture models can be used in many different areas, including finance, marketing and so much more.
- ✓ Gaussian Mixture Models (GMMs) give us more flexibility than K-Means. With GMMs we assume that the data points are Gaussian distributed; this is a less restrictive assumption than saying they are circular by using the mean. That way, we have two parameters to describe the shape of the clusters: the mean and the standard deviation!
- ✓ Taking an example in two dimensions, this means that the clusters can take any kind of elliptical shape (since we have standard deviation in both the x and y directions). Thus, each Gaussian distribution is assigned to a single cluster. In order to find the parameters of the Gaussian for each cluster (e.g the mean and standard deviation) we will use an optimization algorithm called Expectation–Maximization (EM). Take a look at the graphic below as an illustration of the Gaussians being fitted to the clusters. Then we can proceed to the process of Expectation–Maximization clustering using GMMs.
- ✓ Gaussian mixture models (GMM) are a probabilistic concept used to model real-world data sets. GMMs are a generalization of Gaussian distributions and can be used to represent any data set that can be clustered into multiple Gaussian distributions. The Gaussian mixture model is a probabilistic model that assumes all the data points are generated from a mix of Gaussian distributions with unknown parameters.
- ✓ A Gaussian mixture model can be used for clustering, which is the task of grouping a set of data points into clusters. GMMs can be used to find clusters in data sets where the clusters may not be clearly defined. Additionally, GMMs can be used to estimate the probability that a new data point belongs to each cluster. Gaussian mixture models are also relatively robust to outliers, meaning that they can still yield accurate results even if there are some data points that do not fit neatly into any of the clusters. This makes GMMs a flexible and powerful tool for clustering data.
- ✓ It can be understood as a probabilistic model where Gaussian distributions are assumed for each group and they have means and co variances which define their parameters. GMM consists of two parts – mean vectors (μ) & covariance matrices (Σ). A Gaussian distribution is defined as a continuous probability distribution that takes on a bell-shaped curve. Another name for Gaussian distribution is the normal distribution.
- ✓ GMM has many applications, such as density estimation, clustering, and image segmentation.

For density estimation, GMM can be used to estimate the probability density function of a set of data points. For clustering, GMM can be used to group together data points that come from the same Gaussian distribution. And for image segmentation, GMM can be used to partition an image into different regions.

- ✓ Gaussian mixture models can be used for a variety of use cases, including identifying customer segments, detecting fraudulent activity, and clustering images. In each of these examples, the Gaussian mixture model is able to identify clusters in the data that may not be immediately obvious. As a result, Gaussian mixture models are a powerful tool for data analysis and should be considered for any clustering task.

Expectation-maximization (EM) method in relation to GMM

In Gaussian mixture models, an expectation-maximization method is a powerful tool for estimating the parameters of a Gaussian mixture model (GMM). The expectation is termed E and maximization is termed M. Expectation is used to find the Gaussian parameters which are used to represent each component of gaussian mixture models. Maximization is termed M and it is involved in determining whether new data points can be added or not.

- ✓ The expectation-maximization method is a two-step iterative algorithm that alternates between performing an expectation step, in which we compute expectations for each data point using current parameter estimates and then maximize these to produce a new gaussian, followed by a maximization step where we update our gaussian means based on the maximum likelihood estimate.
- ✓ The EM method works by first initializing the parameters of the GMM, then iteratively improving these estimates. At each iteration, the expectation step calculates the expectation of the log-likelihood function with respect to the current parameters. This expectation is then used to maximize the likelihood in the maximization step. The process is then repeated until convergence. Here is a picture representing the two-step iterative aspect of the algorithm

The EM algorithm consists of two steps: the E-step and the M-step. Firstly, the model parameters can be randomly initialized. In the E-step, the algorithm tries to guess the value of based on the parameters, while in the M-step, the algorithm updates the value of the model parameters based on the guess of the E-step. These two steps are repeated until convergence is reached. The algorithm in GMM is repeat until convergence.

Optimization uses the Expectation Maximization algorithm, which alternates between two steps:

1. E-step: Compute the posterior probability over given our current model - i.e. how much do we think each Gaussian generates each datapoint.

2. M-step: Assuming that the data really was generated this way, change the parameters of each Gaussian to maximize the probability that it would generate the data it is currently responsible for.

The K-Means Algorithm:

1. Assignment step: Assign each datapoint to the closest cluster
2. Refitting step: Move each cluster center to the center of gravity of the data assigned to it

The EM Algorithm:

1. E-step: Compute the posterior probability over our current model
2. M-step: Maximize the probability that it would generate the data it is currently responsible for.