UNIT IV

Introduction to Ensembles:

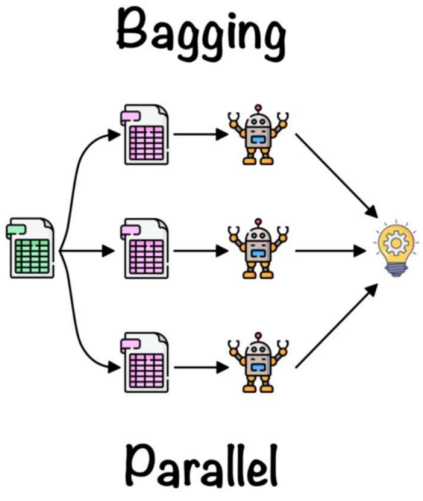
Ensemble learning is a machine learning technique that involves combining multiple models to improve the overall predictive power and stability of the model.

- Ensemble methods are widely used in classification, regression, and clustering.
- The idea behind ensemble learning is that combining the predictions of multiple models can help overcome the limitations of individual models, reduce the impact of noisy data, and improve overall accuracy.
- There are several types of ensemble methods, Bagging, Boosting and Random Forests.

Bagging:

Bagging, or bootstrap aggregating, is an ensemble method in machine learning that involves training multiple models on different samples of a training dataset and then combining their predictions to make a final prediction.

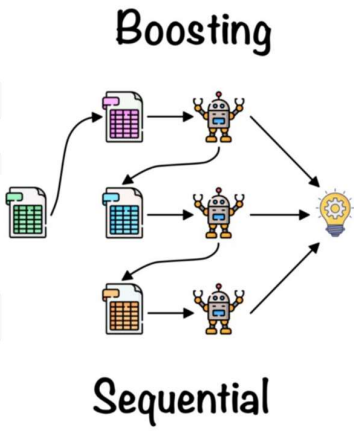
- Bagging involves randomly selecting subsets of the training data with replacement to create multiple new training sets, each of which is used to train a different model.
- Each model in the ensemble is trained independently, so they may use different algorithms or hyperparameters.
- Bagging reduces the variance of the model predictions by averaging the results of many models trained on different samples of the data.
- Bagging is often used with decision trees, where the high variance of a single tree can be reduced by averaging the results of many trees.
- Bagging can also be used with other models, such as support vector machines, neural networks, and regression models.
- One advantage of bagging is that it is relatively simple to implement and can be parallelized easily.
- One potential disadvantage of bagging is that it may not reduce the bias of the models, so it may not improve the accuracy of a model that is already underfitting the data.
- Bagging can be used for classification or regression problems and is a popular method for improving the accuracy of machine learning models.



Boosting:

Boosting is a popular ensemble method used in machine learning for improving the performance of weak learners to create strong learners. It works by combining multiple weak models to create a single strong model.

- Boosting is an iterative process that involves training a sequence of weak learners on weighted versions of the training data.
- Each weak learner is trained to improve the classification performance of the previous weak learner.
- During training, the weights of misclassified samples are increased so that the next weak learner can focus on these samples.
- After training, the weak learners are combined by giving them different weights based on their performance.
- Boosting algorithms can be divided into two categories: gradient boosting and adaptive boosting (AdaBoost).
- Gradient boosting is a method that uses gradient descent optimization to minimize a loss function by adding weak learners to the model.
- AdaBoost is a method that uses weighted samples to create multiple weak learners that are combined to form a strong learner.
- Boosting can be used with many different types of base learners, including decision trees, neural networks, and support vector machines.
- Boosting can be prone to overfitting, so it is important to tune the hyperparameters carefully.
- Boosting has been shown to be effective in many applications, including image classification, speech recognition, and natural language processing.

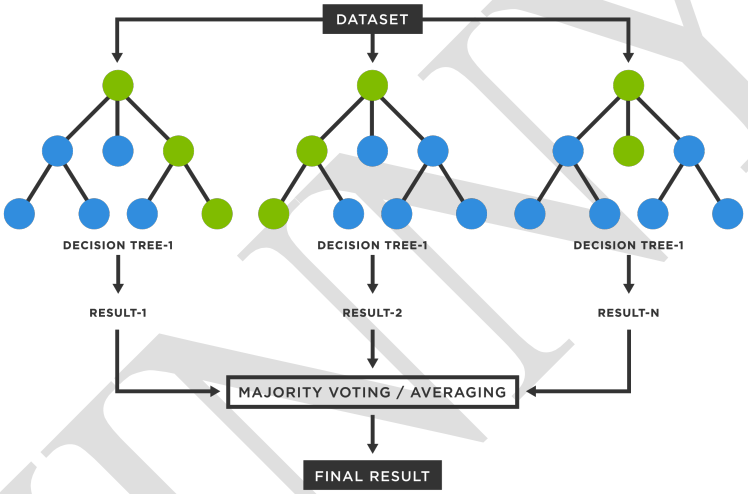


Random forest:

Random Forest is an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class or average prediction of the individual trees at test time.

- Random Forest is a collection of decision trees, where each tree is constructed using a different subset of the training data and features.
- During training, the algorithm creates a large number of decision trees, with each tree being trained on a random subset of the training data and a random subset of the features.
- The trees are constructed using a recursive partitioning algorithm that splits the data into subsets based on the value of a single feature at each node of the tree.

- The splitting criterion is usually based on information gain or Gini impurity, which measures the degree of purity of the classes in each subset.
- Once the trees are constructed, the Random Forest algorithm combines their outputs to make a final prediction for a new input.
- For classification tasks, the algorithm outputs the mode of the class predictions made by each individual tree.
- For regression tasks, the algorithm outputs the average of the predicted values made by each individual tree.
- Random Forest is a highly effective algorithm that is robust to noisy data, overfitting, and the presence of irrelevant features.
- The algorithm is often used in practical applications where high accuracy is required, such as image and speech recognition, natural language processing, and medical diagnosis.
- One of the key advantages of Random Forest is that it can provide information on the relative importance of each feature in the data, which can be useful for feature selection and interpretation of the results.



Introduction to clustering:

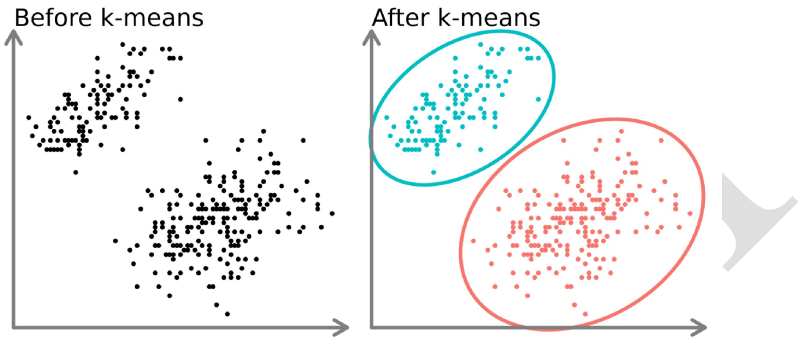
Clustering is a type of unsupervised learning technique that involves grouping a set of objects in such a way that objects in the same group or cluster are more similar to each other than to those in other groups or clusters.

- Clustering is an unsupervised learning technique, which means that the data is not labeled or pre-categorized. Instead, the algorithm identifies patterns in the data and groups similar items together.
- Clustering algorithms are used to partition data into groups, or clusters, based on similarities or distances between data points.
- Clustering can be used for a wide variety of applications, such as image segmentation, anomaly detection, customer segmentation, and data compression.
- There are many different clustering algorithms, each with its own strengths and weaknesses. Some popular algorithms include k-means, hierarchical clustering, and DBSCAN.
- The choice of clustering algorithm and the number of clusters to use depends on the nature of the data and the specific problem at hand.
- Clustering can be used for exploratory data analysis, identifying hidden patterns in the data, and creating summaries of large datasets.

K-means clustering:

K-means clustering is a popular unsupervised learning algorithm used for clustering similar data points in a dataset. The goal of K-means clustering is to partition a given dataset into K clusters, where K is a user-defined hyperparameter. Each cluster should represent a distinct group of data points that are more similar to each other than to data points in other clusters.

- **Initialization:** Select K random data points from the dataset to act as initial centroids of the K clusters.
- **Assignment:** For each data point in the dataset, calculate its distance to each centroid and assign it to the nearest centroid. This forms K clusters.
- **Update:** Recalculate the centroids of each cluster as the mean of all the data points assigned to that cluster.
- **Repeat:** Steps 2 and 3 are repeated iteratively until the clusters no longer change significantly or until a maximum number of iterations is reached.



- The optimal number of clusters can be determined using techniques such as the elbow method or silhouette analysis
- K-means clustering can only find convex-shaped clusters and is sensitive to the initial placement of centroids. It is also affected by outliers and noise in the dataset.

Expectation-Maximization Algorithm:

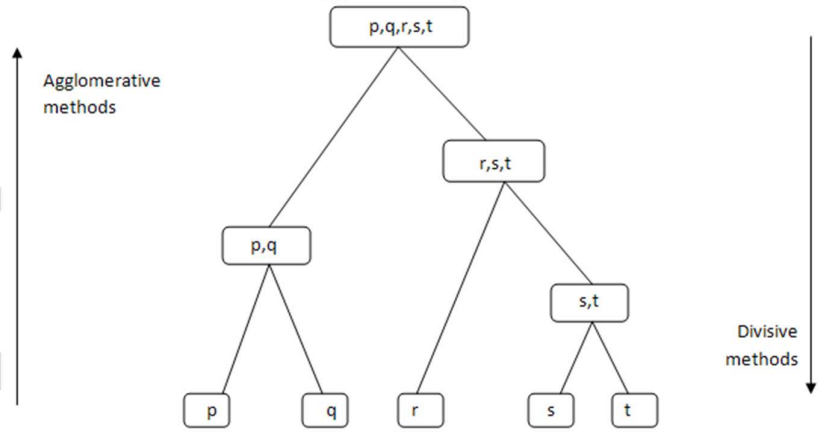
The Expectation-Maximization (EM) algorithm is a powerful technique used for finding maximum likelihood estimates of model parameters in the presence of latent or missing variables.

- The EM algorithm is used when we have incomplete data or when some data is missing.
- In the E-step, we compute the expected value of the latent variable given the observed data and current estimate of the model parameters.
- In the M-step, we update the model parameters to maximize the likelihood of the observed data, given the expected values of the latent variable computed in the E-step.
- The E-step and M-step are repeated iteratively until convergence, where the convergence is typically determined by the change in the log-likelihood of the observed data or the change in the model parameters.
- EM is an iterative algorithm that converges to a local maximum of the likelihood function, but not necessarily the global maximum.
- The EM algorithm is widely used in many applications, such as image and speech processing, bioinformatics, and natural language processing.
- One of the main challenges in using the EM algorithm is that it can be computationally expensive, particularly for large datasets or complex models.

Hierarchical clustering:

Hierarchical clustering is a popular unsupervised learning technique and it is a type of clustering algorithm that creates a tree-like structure or hierarchy of clusters.

- **Initialization:** The first step is to initialize the algorithm by assigning each data point to its own cluster.
- **Computing distance:** Next, we need to calculate the distance between each pair of clusters. There are various distance metrics that can be used, such as Euclidean distance, Manhattan distance.
- **Merging clusters:** Once we have the distances between all pairs of clusters, we need to merge the two closest clusters. There are two common methods for determining the distance between clusters: single linkage and complete linkage. Single linkage calculates the distance between the closest points in each cluster, while complete linkage calculates the distance between the farthest points in each cluster.
- **Updating distance:** After merging two clusters, we need to update the distance between the new cluster and the other clusters. There are two common methods for updating the distance: minimum distance and maximum distance. Minimum distance calculates the distance between the closest points in the new cluster and the closest points in each other cluster, while maximum distance calculates the distance between the farthest points in the new cluster and the farthest points in each other cluster.
- **Repeating steps 3 and 4**: We continue merging clusters and updating distances until all data points belong to a single cluster.
- **Dendrogram:** The result of hierarchical clustering is typically visualized as a dendrogram, which is a tree-like diagram that shows the arrangement of the clusters. The height of each branch in the dendrogram represents the distance between the clusters that were merged at that step.

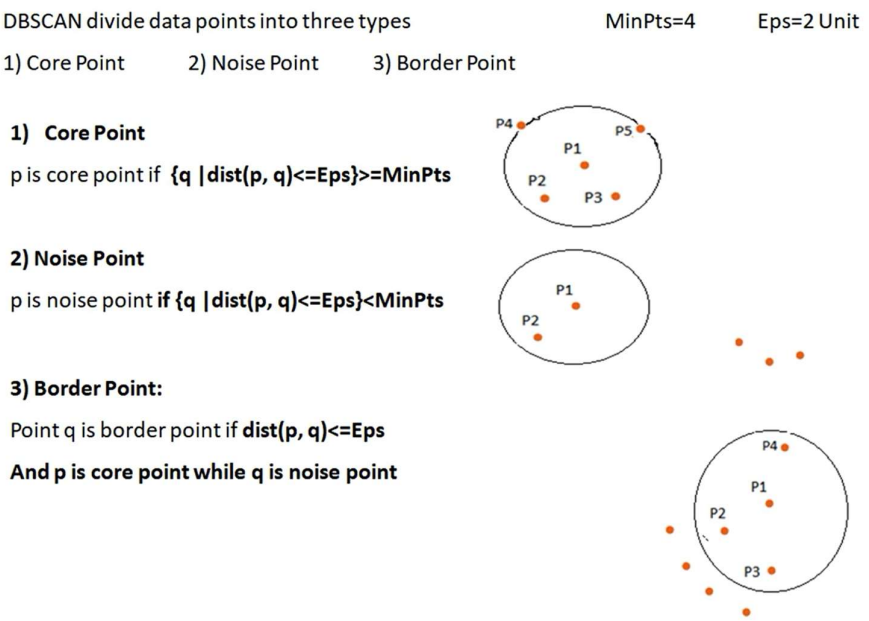


- Agglomerative clustering (also known as bottom-up clustering) starts with each data point as its own cluster and then gradually merges pairs of clusters until all points are in one cluster.
- The algorithm initially considers each data point as a separate cluster, and then repeatedly merges the two nearest clusters until all data points belong to the same cluster.
- Divisive clustering (also known as top-down clustering) starts with all data points in one cluster and then gradually divides it into smaller clusters.
- The algorithm initially considers all the data points as belonging to a single cluster, and then repeatedly divides the cluster into two smaller clusters until each data point is in its own cluster.

Density based clustering: DBSCAN:

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a popular density-based clustering algorithm used for identifying clusters in a dataset with arbitrary shapes and sizes.

- Unlike K-means and Hierarchical clustering, DBSCAN does not require the number of clusters as an input parameter. Instead, it groups together data points that are close to each other in a high-density region and separates out data points that are in low-density regions.



- A core point is a data point that has at least a minimum number of other data points (minPts) within its distance (epsilon).
- A border point is a data point that is within the distance (epsilon) of a core point, but has fewer than the minimum number of other data points (minPts) within its distance.
- A noise point is a data point that is neither a core point nor a border point. It is an outlier that does not belong to any cluster.

The DBSCAN algorithm:

- Randomly select a point from the dataset that has not been visited.
- Retrieve all points from the dataset that are density-reachable from this point within the distance epsilon.
- If the number of points retrieved is greater than or equal to the minimum number of points required to form a dense region (minPts), then a new cluster is created. Otherwise, the point is labeled as noise and the algorithm moves on to the next point in the dataset.
- Expand the cluster by recursively repeating steps 2-3 for each point in the newly created cluster.
- When no more points can be added to the cluster, the algorithm selects another unvisited point from the dataset and repeats the process until all points have been visited.

Choosing the Number of Clusters:

The number of clusters, k, is a knob to adjust the complexity of clustering.

- Clustering always finds k centers, whether they represent meaningful groups or not.
- There are various ways to fine-tune k, such as setting it based on the application, using PCA to uncover the structure of data, or setting a maximum allowed distance or reconstruction error per instance.
- In some applications, validation of the clusters can be done manually by checking if they represent meaningful groups.
- Depending on the clustering method used, the reconstruction error or log likelihood can be plotted as a function of k to look for an "elbow" where the algorithm starts dividing groups.
- In hierarchical clustering, the differences between levels in the tree can help decide on a good split.

Algorithm evaluation methods: Classification Accuracy, Confusion Matrix

Algorithm evaluation is an essential step in machine learning where we test and measure the performance of a model. Two of the commonly used evaluation methods for classification problems are classification accuracy and confusion matrix.

Classification Accuracy:

Classification accuracy measures the fraction of correctly classified instances in a dataset. It is the simplest and most common evaluation metric used for classification problems.

Confusion Matrix:

The confusion matrix shows the actual and predicted class labels for each instance in the dataset and also allows us to calculate various performance metrics, including accuracy, precision, recall, and F1-score. The following are the components of a confusion matrix:

- **True positives (TP**): The number of instances that belong to a positive class that are correctly predicted as positive by the model.
- **False positives (FP**): The number of instances that belong to a negative class but are incorrectly predicted as positive by the model.
- **True negatives (TN):** The number of instances that belong to a negative class that are correctly predicted as negative by the model.
- **False negatives (FN):** The number of instances that belong to a positive class but are incorrectly predicted as negative by the model.

Accuracy = (TP + TN) / (TP + TN + FP + FN)
Precision = TP / (TP + FP)
Recall = TP / (TP + FN)
F1-score = 2 * (Precision * Recall) / (Precision + Recall)