K-Means

K-means is unsupervised machine learning algorithm used for clustering. It organizes data into groups based on their similarity. Algorithm works by selecting one central point called centroid and then each data point is assigned to the closest centroid forming a cluster. Here, K represents the number of clusters to be formed. After forming clusters, we calculate Euclidean distance of each cluster and since K-Means clustering cannot see the best clustering, the only option is to keep the track of these clusters and their total variance, and then do the whole thing again with different starting points. The value of K is determined using graph between reduction in variance and values of K, then the elbow point in graph represents the value of K.

Support Vector Machine

Support Vector Machine (SVM) is a supervised machine learning algorithm used for classification and regression tasks. While it can handle regression problems, SVM is particularly well-suited for classification tasks. The algorithm maximizes the margin between the closest points of different classes. The key idea behind the SVM algorithm is to find the hyper-plane that best separates two classes by maximizing the margin between them. This margin is the distance from the hyper-plane to the nearest data points (support vectors) on each side. When data is not linearly separable (i.e., it can’t be divided by a straight line), SVM uses a technique called kernels to map the data into a higher-dimensional space where it becomes separable. This transformation helps SVM find a decision boundary even for non-linear data. A kernel is a function that maps data points into a higher-dimensional space without explicitly computing the coordinates in that space. This allows SVM to work efficiently with non-linear data by implicitly performing the mapping. Kernels can be linear, polynomial and radial basis function.

DBScan

DBScan is a density-based clustering algorithm that groups data points that are closely packed together and marks outliers as noise based on their density in the feature space. It identifies clusters as dense regions in the data space, separated by areas of lower density. Eps is the radius of the neighbourhood around a data point. A common method to determine eps is by analysing the k-distance graph. There is also the minimum number of required within the eps radius to form a dense region. DBSCAN works by categorizing data points into three types: core points: which have a sufficient number of neighbours within a specified radius (eplison), border points: which are near core points but lack enough neighbors to be core points themselves, noise points: which do not belong to any cluster. By iteratively expanding clusters from core points and connecting density-reachable points, DBSCAN forms clusters without relying on rigid assumptions about their shape or size.

Decision Tree

A decision tree is a supervised learning algorithm used for both classification and regression tasks. It models decisions as a tree-like structure where internal nodes represent attribute tests, branches represent attribute values, and leaf nodes represent final decisions or predictions. Therefore many decision trees can be made according to the given attributes. So for the selection measure of decision tree there are two methods that can be used: 1. Information gain 2. Gini index. Information Gain tells us how useful a question (or feature) is for splitting data into groups. It measures how much the uncertainty decreases after the split. A good question will create clearer groups, and the feature with the highest Information Gain is chosen to make the decision. Gini Index is a metric to measure how often a randomly chosen element would be incorrectly identified. It means an attribute with a lower Gini index should be preferred. f we have a group of people where all bought the product (100% Yes), the Gini Index is 0, indicating perfect purity. But if the group has an equal mix of Yes and No, the Gini Index would be 0.5, showing higher impurity or uncertainty.

Random Forest

A Random Forest is a collection of decision trees that work together to make predictions and in this, we do voting of all the tress to make prediction. It takes different random parts of the dataset to train each tree and then it combines the results by averaging them. This approach helps improve the accuracy of predictions. Random Forest is based on ensemble learning. Ensemble learning combines the predictions of multiple models to make a stronger, more reliable prediction. Multiple Decision Trees are created from the training data. Each tree is trained on a random subset of the data (with replacement) and a random subset of features. This process is known as bagging or bootstrap aggregating. Each Decision Tree in the ensemble learns to make predictions independently. The final prediction is made by combining the predictions of all the Decision Trees. This is typically done through a majority vote (for classification) or averaging (for regression).