

Assignment No: - 3

Decision Tree

Problem Statement: -

Apply appropriate ML algorithm on a dataset collected by obtaining the person's information like age, gender, annual income, spending score, work experience... to predict the person's profession.

Objective:

The main objective is to develop a predictive model using K-Nearest Neighbors(KNN) to classify the person into profession, whether he/she is a Doctor, Engineer, Artist, Lawyer, etc.

To achieve this, we've collected data, including Customer ID, Gender, Age, Annual Income, and Spending Score. The objective is to build a machine learning model that can accurately predict whether the person is a Engineer or Doctor or Lawyer based on these features.

S/W Packages and H/W apparatus used: OS: Windows, Kernel: Python 3, Tools: Google Colab

Libraries and packages used: NumPy, Pandas, Matplotlib, Scikit-Learn

Theory:

K-Nearest Neighbors (KNN) Algorithm:

K-Nearest Neighbors (KNN) is a supervised learning algorithm used for classification and regression tasks. It predicts the class or value of a data point by analyzing the majority class or average value of its k nearest neighbors in the feature space.

Methodology:

- **Data Preprocessing:**

Handle missing values and encode categorical variables if necessary.

Scale the features to ensure they are on a similar scale, as KNN is sensitive to the scale of the features.

- **Train-Test Split:**

Split the dataset into training and testing sets to train the KNN model on a subset of the data and evaluate its performance on unseen data.

- **Applying KNN Algorithm:**

Fit the KNN model to the training data, specifying the number of neighbors (k) to consider. For each data point in the testing set, find its k nearest neighbors based on a distance metric (e.g., Euclidean distance).

Predict the class of the data point based on the majority class of its k nearest neighbors.

- **Model Evaluation:**

Evaluate the performance of the KNN model using metrics such as accuracy, precision, recall, and F1-score on the testing set.

Tune the hyperparameters, such as the number of neighbors (k), using techniques like cross-validation to improve model performance.

Advantages and Applications & Limitations/Example:

Advantages:

- **Simple and Intuitive:** KNN is a straightforward algorithm that is easy to understand and implement, making it suitable for beginners and quick prototyping.
- **Non-Parametric:** KNN is a non-parametric algorithm, meaning it makes no assumptions about the underlying distribution of the data. It can capture complex decision boundaries and is robust to outliers and noisy data.
- **Versatile:** KNN can be applied to both classification and regression tasks, making it a versatile algorithm for various types of predictive modeling problems.
- **Adaptive Learning:** KNN is an instance-based learning algorithm that does not require explicit training. It adapts to new data points during the prediction phase, making it suitable for dynamic or evolving datasets.
- **Effective with Small Datasets:** KNN performs well with small to medium-sized datasets, where the computation time is not a significant concern. It does not require the calculation of explicit model parameters, resulting in fast training times.

Applications:

- **Classification:** KNN is commonly used for classification tasks such as image recognition, text categorization, and spam detection. For example, in image recognition, KNN can classify images based on their similarity to known images in the dataset.
- **Regression:** KNN can also be applied to regression tasks such as predicting house prices, stock prices, or weather forecasts. For instance, in predicting house prices, KNN can estimate the price of a house based on the prices of similar houses in the neighborhood.
- **Recommendation Systems:** KNN is used in recommendation systems to suggest items or products to users based on their similarity to other users or items. For example, in movie recommendation systems, KNN can recommend movies to users based on their similarity to other users who have similar movie preferences.
- **Anomaly Detection:** KNN can be employed for anomaly detection in areas such as fraud detection, network intrusion detection, or manufacturing quality control. It identifies data points that deviate significantly from the majority of the data and may indicate anomalies or outliers.
- **Pattern Recognition:** KNN is utilized in pattern recognition tasks such as handwriting recognition, facial recognition, and speech recognition. It classifies patterns based on their similarity to known patterns in the dataset, enabling accurate recognition and classification.

Limitations:

- **Computationally Expensive:** As the size of the dataset grows, the computation time required to find the nearest neighbors increases significantly, making KNN impractical for large datasets or real-time applications.
- **Sensitive to Noise and Outliers:** KNN is sensitive to noisy data and outliers, as they can significantly impact the distance calculations and affect the model's predictions. Preprocessing techniques like outlier removal and feature scaling are essential to mitigate this issue.
- **Curse of Dimensionality:** In high-dimensional feature spaces, the distance between data points may become less meaningful, leading to degraded performance of the KNN

algorithm. Dimensionality reduction techniques may be necessary to address this limitation.

- **Need for Optimal Choice of K:** The performance of the KNN algorithm depends on the choice of the number of neighbors (k). A suboptimal choice of k may result in underfitting or overfitting of the model. Cross-validation techniques can be used to select the optimal value of k .
- **Imbalanced Datasets:** KNN may perform poorly on imbalanced datasets, where one class is significantly more prevalent than the others. In such cases, techniques like oversampling, undersampling, or using weighted distances can help improve the model's performance.

Example:

Consider a dataset containing information about customers' shopping habits, including features such as age, income, and spending score. Suppose we want to predict whether a new customer will be a high-spender or a low-spender based on their demographic characteristics.

In this scenario, we can apply the K-Nearest Neighbors (KNN) algorithm to classify the new customer into one of the two categories (high-spender or low-spender) based on the similarity to its nearest neighbors in the feature space. By analyzing the distances between the new customer and its neighbors, KNN can predict the customer's spending behavior accurately.

However, KNN may face challenges if the dataset is large, contains noisy data, or has imbalanced classes. Additionally, selecting the optimal value of k is crucial to ensure the model's performance. Despite these limitations, KNN remains a powerful and versatile algorithm for classification and regression tasks in various domains.

Working / Algorithm:

Step-1: Select the number K of the neighbors

Step-2: Calculate the Euclidean distance of K number of neighbors

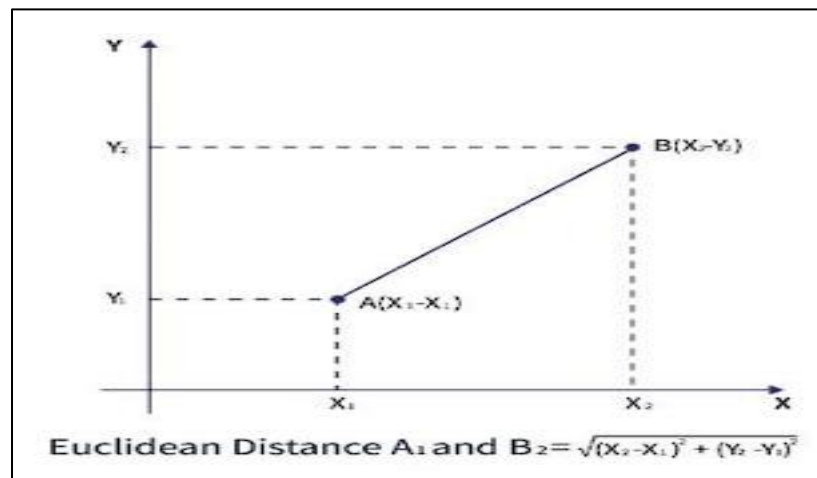
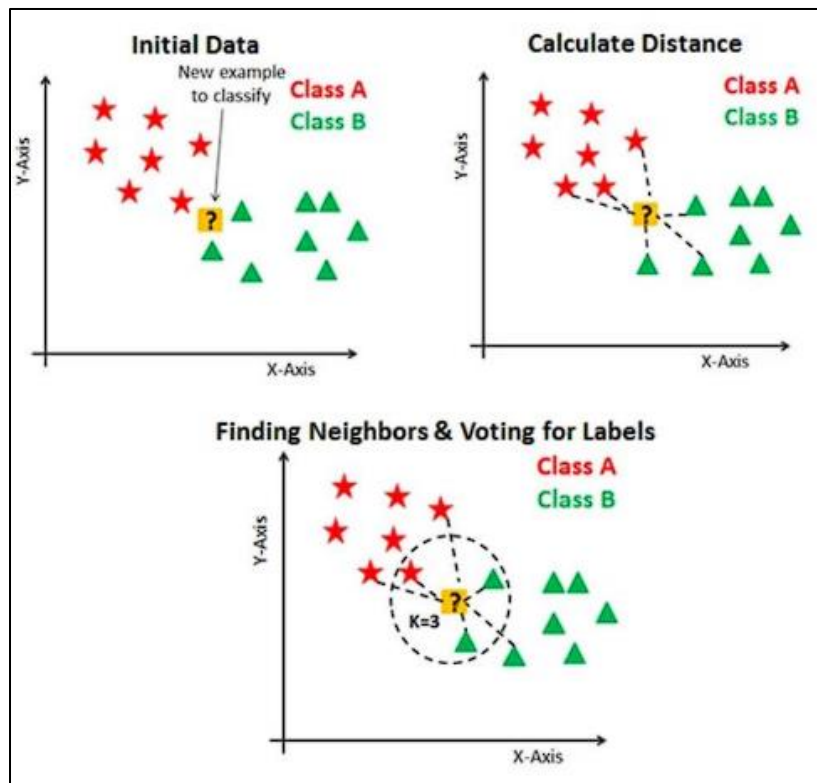
Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.

Step-4: Among these k neighbors, count the number of the data points 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.

Diagram:



Conclusion:

In conclusion, the K-Nearest Neighbors (KNN) algorithm offers a simple yet effective approach to predictive modeling, particularly for classification tasks like predicting customer responses to special offers in the cosmetics shop. By leveraging the advantages of simplicity, adaptability, and versatility, KNN can provide valuable insights and actionable predictions for optimizing marketing strategies. However, it's essential to consider the limitations of KNN, such as computational complexity, sensitivity to noise, and the need for optimal hyperparameter selection, to ensure the robustness and reliability of the model in real-world applications.