Laboratory assignment

Component 3

Authors: Patricia Moga, Ciobanu Sergiu-Tudor
Group: 6

January 13, 2026

1 K-Nearest Neighbor

1.1 Brief Description of Employed ML Technique

K-Nearest Neighbors (KNN) is a supervised machine learning algorithm that makes predictions by finding the k closest data points to a new sample and deciding the result based on them, usually through majority voting. It is non-parametric, meaning it does not assume any specific data distribution, and it is a lazy learner because it does not build a model during training; instead, it stores the entire dataset and performs the required calculations only when a prediction is needed. KNN typically uses a distance measure such as Euclidean distance to identify the nearest points. The choice of k is important, since very small values can make the model sensitive to noise, while very large values can oversimplify the decision.

The goal is to determine whether a passenger is Satisfied or Neutral/Dissatisfied based on their flight experience. Using features such as seat comfort, service quality, delays, and other travel-related attributes from the Airline Passenger Satisfaction dataset, the model learns to assign each passenger to one of the two classes. The objective is to build a classifier that can reliably distinguish between satisfied and unsatisfied passengers using the available data.

1.2 Design of the Learning Task

The learning task is formally designed using the component-based framework:

Target Function to be Learned (Formal Definition)

The target function $f^*(x)$ maps each passenger, represented by a feature vector $x \in \mathbb{R}^n$, to one of the two satisfaction classes:

$$f^*(x) : \mathbb{R}^n \to \{\text{Satisfied}, \ \text{Neutral/Dissatisfied}\}$$

This function represents the ideal relationship between a passenger's characteristics and their true satisfaction level. Since the real function is unknown, the goal of the learning process is to approximate $f^*(x)$ as accurately as possible using the labeled data available in the dataset.

Learning Hypothesis (Approximation of the Target Function)

The hypothesis function $h(x)$ provides an approximation of the unknown target function by assigning each new passenger to a class based on its nearest neighbors in the training set. For a given input x, the function selects the k closest data points and predicts the class that appears most frequently among them:

$$h(x) = \arg \max_{c \in \{\text{Satisfied, Neutral/Dissatisfied}\}} |\{ x_i \in N_k(x) \mid y_i = c \}|$$

Here, the expression $|\{ x_i \in N_k(x) \mid y_i = c \}|$ represents the number of neighbors among the k closest points whose class label is c. In other words, $h(x)$ returns the class with the majority vote among the k nearest neighbors of x. This makes the hypothesis a direct, distance-based approximation of the true satisfaction function.

Representation of the Learned Function

In the K-Nearest Neighbors algorithm, the learned function does not rely on explicit parameters, since KNN is a non-parametric and lazy learning method. Instead, the model is represented by the elements required to apply the distance-based decision rule during prediction. Specifically, the classifier stores the entire training dataset $(X_{\text{train}}, y_{\text{train}})$, the chosen distance metric (typically the Euclidean distance), and the selected value of k. These components determine how similarity between data points is evaluated and how the final prediction is made. Because no parameters are learned during training, the representation of the model is defined entirely by the stored data and the rules used to identify the nearest neighbors.

Learning Algorithm (K-Nearest Neighbor)

In the K-Nearest Neighbors algorithm, the learning process consists of preparing the data and storing the training instances that will be used during prediction. After preprocessing the dataset, such as normalizing the feature values, the algorithm keeps the training samples available for distance comparisons. When a new input x needs to be classified, the algorithm computes its distance to all training points using the chosen metric. These distances are then sorted, and the k closest points are selected as the nearest neighbors. The predicted class is obtained by identifying the most frequent label among these neighbors. The final decision, therefore, follows directly from the similarity between the new instance and the stored training data.

2 K-Means Clustering

2.1 Brief Description of Employed ML Technique

The learning problem is defined as unsupervised classification, commonly referred to as clustering. The objective is to discover hidden patterns within the airline passenger satisfaction data by dividing the dataset $(\mathcal{X})$ into a pre-defined number of K distinct subsets or clusters $(\mathcal{C}_1, \ldots, \mathcal{C}_K)$. The goal is to assign passenger instances to clusters such that objects within a cluster reveal high similarity regarding their features (flight metrics, passenger demographics, service ratings), and low similarity with instances in other clusters.

2.2 Design of the Learning Task

The learning task is formally designed using the component-based framework:

Target Function to be Learned (Formal Definition)

- **Input Space $(\mathcal{X})$:** The set of all passenger feature vectors, $\mathbf{x}_i \in \mathbb{R}^M$, where $M = 22$ is the number of cleaned, numerical features (demographics, ratings, delays, flight distance).

- **Output Space ($\mathcal{C}$):** The finite system of K subsets, $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_K\}$, such that their union covers the entire input space ($\mathcal{C}_1 \cup \ldots \cup \mathcal{C}_K = \mathcal{X}$).The value of K is determined through a preprocessing step to ensure cluster quality.

- **Target ($\boldsymbol{f^*}$):** The optimal partitioning function, defined by the set of centroids $\{\boldsymbol{\mu}_j\}_{j=1}^{K}$, that minimizes the dissimilarity within each cluster to achieve the goal of maximizing intra-cluster similarity. This is formally done by minimizing the Within-Cluster Sum of Squares (WCSS), which sums the squared Euclidean distance between each point and its assigned cluster centroid.

$$f^*(\boldsymbol{\mu}) = \arg\min_{\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_K} \sum_{i=1}^{N} \min_{j=1,\ldots,K} ||\mathbf{x}_i - \boldsymbol{\mu}_j||^2$$

Learning Hypothesis (Approximation of the Target Function)

The learning hypothesis ($\hat{\boldsymbol{f}}$) is the approximation of the target function found by the clustering algorithm after training. It represents the final cluster assignment function.

- **Hypothesis ($\hat{f}$):** The function that assigns any data point $\mathbf{x}$ to the cluster j whose centroid ($\hat{\boldsymbol{\mu}}_j$) is nearest to $\mathbf{x}$ in the Euclidean space.

$$\hat{f}(\mathbf{x}) = \arg\min_j ||\mathbf{x} - \hat{\boldsymbol{\mu}}_j||^2$$

Representation of the Learned Function

The representation is the direct, physical output of the clustering algorithm. The final coordinates of the **K** centroids in the M-dimensional feature space: $\hat{\boldsymbol{\mu}} = \{\hat{\boldsymbol{\mu}}_1, \ldots, \hat{\boldsymbol{\mu}}_K\}$. These centroids act as the prototypes or data concepts defining the average passenger characteristics for each identified segment.

Learning Algorithm (K-Means)

The K-Means algorithm is an iterative partitioning method designed to minimize the WCSS objective function. The procedure maximizes the intra-cluster similarity by following a repeated two-step process:

1. **Assignment:** Each passenger instance is assigned to the cluster with the nearest mean (centroid).
2. **Update:** The mean (centroid) of each cluster is recomputed based on the new set of assigned instances.

This iterative process is repeated until the centroids no longer move significantly, although the algorithm is not guaranteed to converge to the global optimum.