# K-Nearest Neighbor Algorithm in Machine Learning

Student's Name

Department, Institutional Affiliation

Course Code: Course Name

Professor's Name

Date

**K-Nearest Neighbor Algorithm in Machine Learning**

Machine learning, the practice of using statistical methods to enable computers to learn without explicit programming, has opened up a myriad of new technological applications. Various supervised and unsupervised machine learning algorithms are available today that developers can select when building their applications. One such algorithm is the K-nearest neighbor (KNN), which is extensively utilized in pattern recognition and data mining. KNN is a non-parametric technique used for regression and classification. In classification, an object is categorized by the majority of the neighbors and, therefore, the output is class membership. In regression, the output is the object's property value, which is computed as the average of the values of the nearest neighbors of the object. The main principle behind the operation of the KNN algorithm is that similar objects exist close to one another.

In the KNN classification approach, a distance function is utilized to assign an object to the most common class among the k-nearest neighbors. The Euclidean distance is usually used as the distance measure for continuous variables (Zhang, 2016). The Euclidean distance between X = ($x_1$, $x_2$, $x_3$, …, $x_n$) and Y = ($y_1$, $y_2$, $y_3$, …, $y_n$) is defined as follows:

$$D(X,Y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Where x and y are points in the n-space

$x_i$ and $y_i$ are Euclidean vectors, from the initial point

n = n-space

Using the above formula, the distance between the new example (E) and other points within the training set can be calculated. The K-NN algorithm perceives all instances to correspond to points in the n-space, and each instance is defined by numerical attributes. Accordingly, the

training data comprises a set of vectors, with each vector having a class label. To classify points, the vector compares vector features with various K nearest points.

The implementation of the KNN algorithm entails several steps. It starts with loading data and initializing the K to the selected number of neighbors. Thereafter, for each example, the algorithm calculates the distance between the query and current example and adds the index and distance of the example to a collection (Harrison, 2018). Next, it sorts the collection by distances, selects the first K entries, and returns the labels of the chosen entries. In a classification problem, the algorithm returns the mode of those labels. The process of selecting the K value involves running the KNN algorithm several times using different values and then choosing the value that minimizes errors without adversely affecting the accuracy of the algorithm.

The KNN algorithm has various strengths. To start with, it is easy and simple to implement and utilize. A key feature of this algorithm is that there is no need to build a model, set parameters, and make extra assumptions. In addition, KNN can be applied to data with varying types of distribution. According to Harrison (2018), it is versatile as it can be used for regression and classification. Furthermore, if the sample is adequate, the algorithm exhibits good classification accuracy. On the downside, it takes some time to categorize a new example as one has to compute the distance from it and the other examples. Additionally, the algorithm requires a large number of samples to produce accurate results. Harrison (2018) also observed that KNN gets slower as the number of predictors, examples, and independent variables increases. The process of choosing the appropriate value of K can be difficult.

Despite the weaknesses, KNN continues to be utilized in many applications. According to Bronshtein (2017), KNN is used to compute credit ratings by comparing the financial details of a

client with those of similar people to determine their credit rating. Individuals are likely to have the same credit rating if their financial situations are similar. Accordingly, instead of computing a client's credit score, the company can simply use the KNN algorithm to predict the rating with a high degree of accuracy. Based on Bronshtein (2017), the algorithm is also used to develop recommender systems on platforms such as Netflix and Amazon. Here, the objective is to suggest to the client a product that they are likely to use based on their previous consumption. The capability to utilize KNN in practical settings makes it an important algorithm in data science.

To conclude, KNN is a machine learning algorithm that predicts the classification of a new sample point based on similarity. KNN is non-parametric as it does not make assumptions and lacks an explicit training phase. The Euclidean distance is the most commonly used measure in KNN as it determines the distance between continuous variables. Some of the strengths of this algorithm include simplicity, ability to apply it on data with different distributions, capacity to compute regression and classification, and adequate accuracy. However, it takes some time to organize data, requires large numbers of samples to produce accurate outcomes, and gets slower as requirements increase. KNN finds applications in various settings, including recommender and credit rating systems.

# References

Bronshtein, A. (2017, April 12). *A quick introduction to k-nearest neighbors algorithm*. Noteworthy – The Journal Blog. **https://blog.usejournal.com/a-quick-introduction-to-k-nearest-neighbors-algorithm-62214cea29c7**

Harrison, O. (2018, Sep. 10). *Machine learning basics with the k-nearest neighbors algorithm*. Toward Data Science. **https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761**

Zhang, Z. (2016). Introduction to machine learning: K-nearest neighbors. *Annals of Translational Medicine*, *4*(11), 218-225. 10.21037/atm.2016.03.37