Project Proposal

Topic : K-Nearest Neighbours

Group Members:

Manasvini Nittala (mn777)

Sahithi Reddy Sakinala (ss4362)

Jahnavi Manchala (jm2658)

K - Nearest Neighbors or KNN algorithm is a simple algorithm that uses the entire dataset in its training phase. Whenever a prediction is required for an unseen data instance, it searches through the whole training dataset for k-most similar instances. The data with the most similar instance is finally returned as the prediction. This algorithm is used to solve classification model problems.

KNN suggests that if **you are similar to your neighbors, you are one of them**. The k-nearest neighbor's algorithm uses a straightforward approach to perform classification. K-nearest neighbor or K-NN algorithm creates an imaginary boundary to classify the data. When new data points come in, the algorithm will try to predict that to the nearest boundary line.

When tested with a new example, it looks through the training data and finds the k training examples closest to the new example. It then assigns the most common class label (among those k-training examples) to the test example.

In this project, we aim to explore, understand, and implement the k-nearest neighbor's algorithm.

We aim to run the algorithm for different values of k, as selecting the optimal K value to achieve maximum accuracy of the model is always challenging. To get the exact k-value, we need to test the model for every expected k-value. **Small values of K aren't suitable** for classification. The optimal K value usually found is the **square root of N,** where N is the total number of samples. We can use an error plot or accuracy plot to find the most favorable K value, **then choose the K value as having a minimum error rate.**

For a comprehensive explanation of the working of this algorithm, we will demonstrate the implementable approach to perceive the superior value of K in the KNN algorithm.

We also aim to provide an instructional video for implementing the above-stated algorithm.

Since the concept of K nearest neighbors was not covered in class, we would like to explore this domain, understand, and implement the relevant algorithm, and analyze its complexities.