|  |  |
| --- | --- |
| Name Of The Student | Vaishnavi G |
| Internship Project Topic | Build a Classification Model for Drug Trials Dataset |
| Name of the Organization | TCS iON |
| Name of the Industry Mentor | Himdweep Walia |
| Name of the Institute | SRM Institute of Science and Technology |

|  |  |  |
| --- | --- | --- |
| Date | Day # | Hours Spent |
| 6/11/2022 | 26 | 5 hours |
| Activities done during the day:  **K-Nearest Neighbor(KNN) Algorithm for Machine Learning:**  **What is the k-nearest neighbors algorithm?**  The k-nearest neighbors (KNN) algorithm is a data classification method for estimating the likelihood that a data point will become a member of one group or another based on what group the data points nearest to it belong to.  The k-nearest neighbor algorithm is a type of supervised machine learning algorithm used to solve classification and regression problems. However, it's mainly used for classification problems.  KNN is a lazy learning and non-parametric algorithm.  **Why do we need a K-NN Algorithm?**  Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x1, so this data point will lie in which of these categories.  To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset. Consider the below diagram:  IMG_256  **How does K-NN work?**  The K-NN working can be explained on the basis of the below 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.  Suppose we have a new data point and we need to put it in the required category. Consider the below image:  IMG_256   * Firstly, we will choose the number of neighbors, so we will choose the k=5. * Next, we will calculate the Euclidean distance between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry. It can be calculated as:   IMG_256   * By calculating the Euclidean distance we got the nearest neighbors, as three nearest neighbors in category A and two nearest neighbors in category B. Consider the below image:   IMG_256  **How to select the value of K in the K-NN Algorithm?**  Below are some points to remember while selecting the value of K in the K-NN algorithm:   * There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5. * A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model. * Large values for K are good, but it may find some difficulties.   **Advantages of KNN Algorithm:**   * It is simple to implement. * It is robust to the noisy training data * It can be more effective if the training data is large.   **Disadvantages of KNN Algorithm:**   * Always needs to determine the value of K which may be complex some time. * The computation cost is high because of calculating the distance between the data points for all the training samples.   **To fit the KNN algorithm with K=1:**   |  | | --- | | from sklearn.neighbors import KNeighborsClassifier  data = list(zip(x, y))  knn = KNeighborsClassifier(n\_neighbors=1)  knn.fit(data, classes) |   **Steps to implement the K-NN algorithm:**   * Data Pre-processing step * Fitting the K-NN algorithm to the Training set * Predicting the test result * Test accuracy of the result(Creation of Confusion matrix) * Visualizing the test set result.   Reference:  <https://www.ibm.com/in-en/topics/knn#:~:text=The%20k%2Dnearest%20neighbors%20algorithm%2C%20also%20known%20as%20KNN%20or,of%20an%20individual%20data%20point./>  <https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761/> | | |