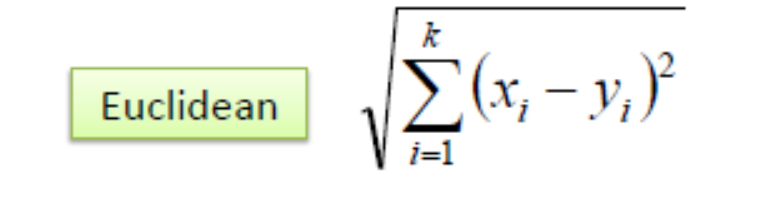
**KNN Report**

KNN

K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). A case is classified by a majority vote of its neighbors, with the case being assigned to the class most common amongst its K nearest neighbors measured by a distance function. If K = 1, then the case is simply assigned to the class of its nearest neighbor[1].

Distance Function:



Dataset

The adult dataset, hosted by The Machine Learning Group at UCI, contains census information from 1994[4]. With this data, we are tasked of predicting whether a person makes more than $50K/year. The adult dataset is a fairly large set, consisting of 48,842 instances. There are 14 attributes prescribed to each person: {income (‘>50K’ or ‘<=50K’), age, WorkClass, fnlwgt, education, education-num, marital-status, occupation, relationship, race, sex, capital-gain, capital-loss, hours-per-week, native-country}[3].

Data Preprocessing

The dataset contained ‘ ?’ which I dropped from the pandas dataframe to get the better accuracy. The dataset was reduced to 30,162 records.

Implementation

* Loaded the dataset into pandas dataframe using pd.read\_csv.
* Cleaning and preprocessing the data.
* Initialize the value of k
* Splitting the data into training and testing using ‘train\_test\_split’ from sklearn.model\_selection.
* Creating KNeighborsClassifier object and using it metrics like ‘euclidean’ to get the accuracy score, classification report and confusion matrix[2].

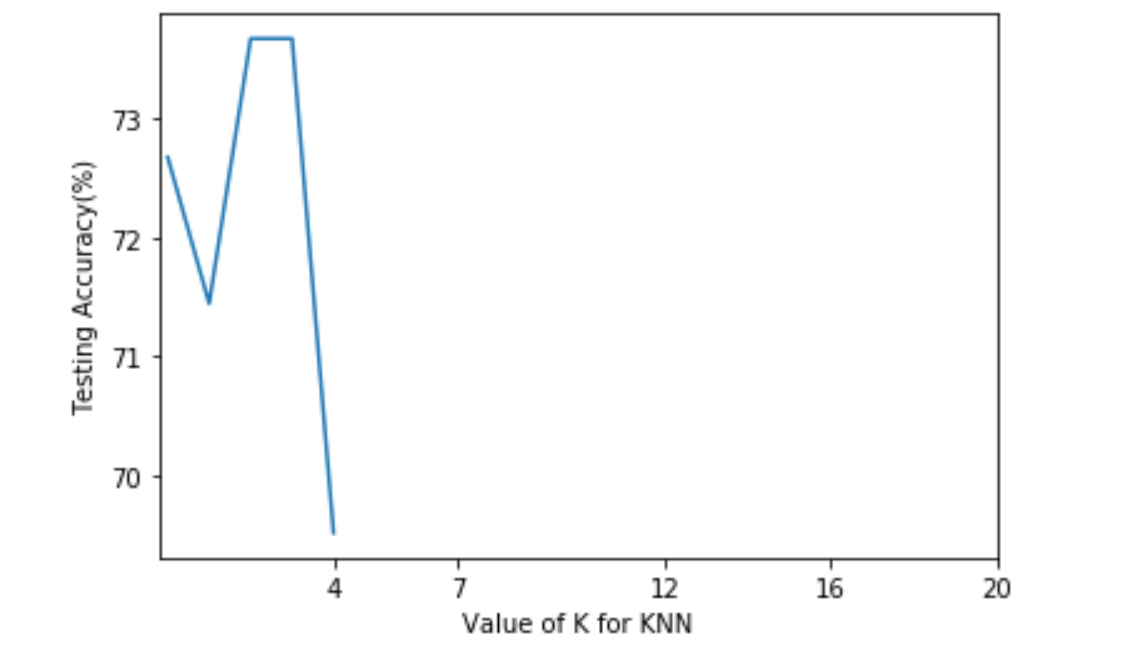
1. For getting the predicted class, iterate from 1 to total number of training data points
2. Calculate the distance between test data and each row of training data. Here we will use Euclidean distance as our distance metric since it’s the most popular method. The other metrics that can be used are Chebyshev, cosine, etc.
3. Sort the calculated distances in ascending order based on distance values
4. Get top k rows from the sorted array
5. Get the most frequent class of these rows
6. Return the predicted class

* Visualizing the KNN using ‘matplotlib’ library.

Results

As per my analysis, the got the following accuracy score for different K values:

|  |  |
| --- | --- |
|  | Accuracy Score |
| K=4 | 69.52% |
| K=7 | 71.44% |
| K=12 | 72.67% |
| K=16 | 73.66% |
| K=20 | 73.66% |



Conclusion

In KNN, finding the value of k is not easy. A small value of k means that noise will have a higher influence on the result and a large value make it computationally expensive but reduces the noise. As per my analysis, K-value increases it has more and more class labels to classify the unknown label but I think ideal value would between 15-20.

References

[1] <https://www.saedsayad.com/k_nearest_neighbors.htm>

[2] <https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/>

[3] <http://cseweb.ucsd.edu/classes/sp15/cse190-c/reports/sp15/024.pdf>

[4] <https://archive.ics.uci.edu/ml/datasets/census+income>

[5] <https://towardsdatascience.com/knn-using-scikit-learn-c6bed765be75>