K - Nearest Neighbour(KNN)

Introduction

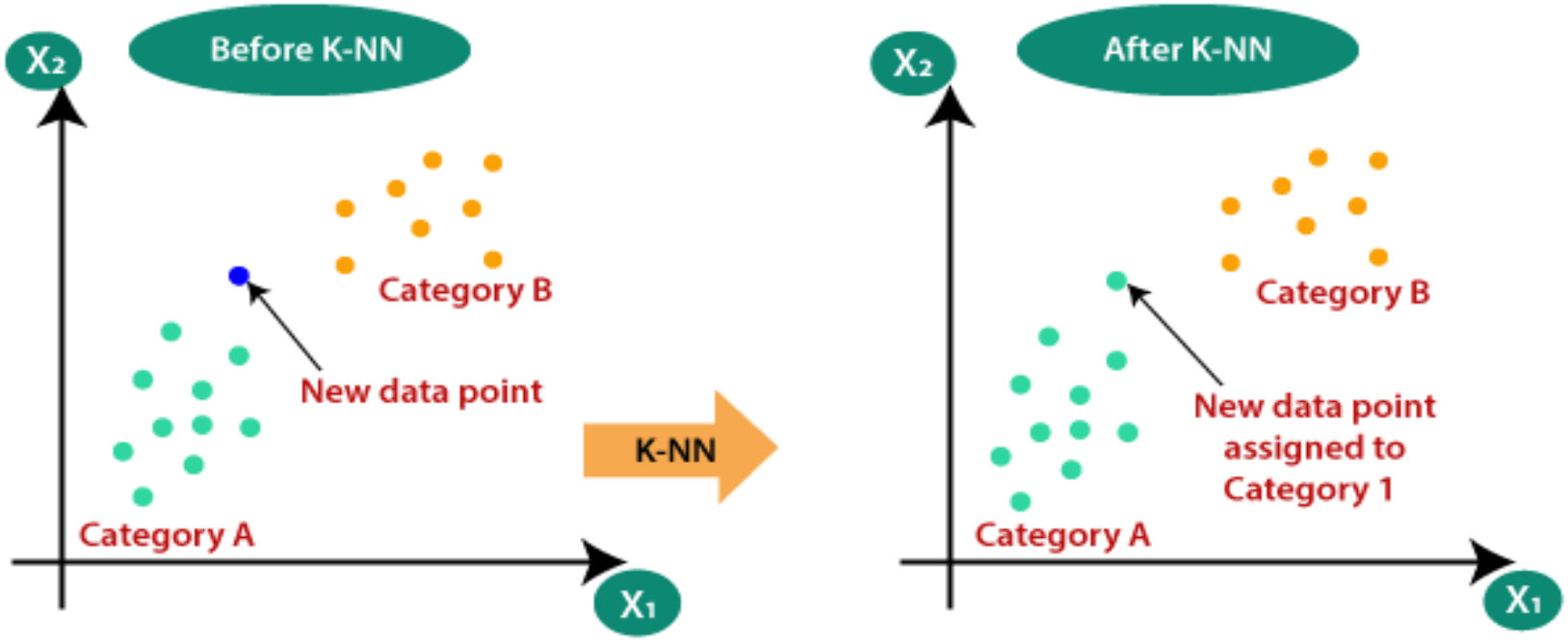
What is KNN?

K Nearest Neighbour is a simple algorithm that stores all the available cases and classifies the new data or case based on a similarity measure. It is mostly used to classifies a data point based on how its neighbours are classified.

- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
- K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
- It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

Why Do We Need KNN

- 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:



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.

Euclidean Distance

$$dist = \sqrt{\sum_{k=1}^{p}(a_k - b_k)^2}$$

Minkowski Distance

$$dist = \sum_{k=1}^{p} |a_k - b_k|^{r}\,^{1/r}$$

Manhattan Distance

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

KNN Example

For example classifying Defaulter/Non-Defaulter based on Age and Loan.

We need to predict Andrew default status (Yes or No).

Customer	Age	Loan	Default
John	25	40000	N
Smith	35	60000	N
Alex	45	80000	N
Jade	20	20000	N
Kate	35	120000	N
Mark	52	18000	N
Anil	23	95000	Y
Pat	40	62000	Y
George	60	100000	Y
Jim	48	220000	Y
Jack	33	150000	Y
Andrew	48	142000	?

We need to predict Andrew default status by using Euclidean distance

Calculate Euclidean distance for all the data points.

Customer	Age	Loan	Default	Euclidean distance
John	25	40000	N	1,02,000.00
Smith	35	60000	N	82,000.00
Alex	45	80000	N	62,000.00
Jade	20	20000	N	1,22,000.00
Kate	35	120000	N	22,000.00
Mark	52	18000	N	1,24,000.00
Anil	23	95000	Y	47,000.01
Pat	40	62000	Y	80,000.00
George	60	100000	Y	42,000.00
Jim	48	220000	Y	78,000.00
Jack	33	150000	Y	8,000.01
Andrew	**48**	**142000**	**?**	

First Step calculate the Euclidean distance dist(d) = Sq.rt $(x_1-y_1)^2 + (x_2-y_2)^2$
= Sq.rt$(48-25)^2 + (142000 - 40000)^2$
dist (d_1) = 1,02,000.

We need to calcuate the distance for all the datapoints

With K=5, there are two Default=N and three Default=Y out of five closest neighbors. We can say default status for Andrew is 'Y' based on the major similarity of 3 points out of 5.

Customer	Age	Loan	Default	Euclidean distance	Minimum Euclidean Distance
John	25	40000	N	1,02,000.00	
Smith	35	60000	N	82,000.00	
Alex	**45**	**80000**	**N**	**62,000.00**	**5**
Jade	20	20000	N	1,22,000.00	
Kate	**35**	**120000**	**N**	**22,000.00**	**2**
Mark	52	18000	N	1,24,000.00	
Anil	**23**	**95000**	**Y**	**47,000.01**	**4**
Pat	40	62000	Y	80,000.00	
George	**60**	**100000**	**Y**	**42,000.00**	**3**
Jim	48	220000	Y	78,000.00	
Jack	**33**	**150000**	**Y**	**8,000.01**	**1**
Andrew	48	142000	?		

Let assume K = 5

Find minimum euclidean distance and rank in order (ascending)

In this case, 5 minimum euclidean distance. With k=5, there are two Default = N and three Default = Y out of five closest neighbors.

We can say Andrew default stauts is 'Y' (Yes)

Note : K-NN is also a lazy learner because it doesn't learn a discriminative function from the training data but "memorizes" the training dataset instead.

➢ Pros of KNN

1. Simple to implement
2. Flexible to feature/distance choices
3. Naturally handles multi-class cases
4. Can do well in practice with enough representative data

➢ Cons of KNN

1. Need to determine the value of parameter K (number of nearest neighbors)
2. Computation cost is quite high because we need to compute the distance of each query instance to all training samples.
3. Storage of data
4. Must know we have a meaningful distance function.

➢ How to select optimal value of K for KNN

1. There are no pre-defined statistical methods to find the most favorable value of K.
2. Initialize a random K value and start computing.
3. Choosing a small value of K leads to unstable decision boundaries.
4. The substantial K value is better for classification as it leads to smoothening the decision boundaries.
5. **Derive a plot between error rate and K denoting values in a defined range. Then choose the K value as having a minimum error rate.**