K – Nearest Neighbor Classifier

Dr. Kuppusamy .P
Associate Professor / SCOPE

Parametric vs Non-parametric

Parametric

- Set of fixed parameters uses to determine a probability model.
- The best fit in Linear regression model with one dependent variable and one independent variable is the regression line equations with optimized parameters intercept and coefficient.
- In classification algorithm, get a decision boundary that separates different target classes. In Logistic regression, get a decision boundary by optimizing parameters.

Non-Parametric

- No need to make any assumption of parameters for the given population
- Number of parameters grows ((No fixed) with the size of the training dataset i.e., learning algorithm needs to keep around an entire training set, even after training.

Example:

- KNN algorithm

Instance Based Learning

- Instance-based learning (*lazy* learning) is typically not "transforming" the training instances into more general "statements".
- Instead, the given training data is simply stored and, when a new test instance is encountered, a set of similar, related instances is retrieved from memory that are used to classify the new test instance.
- Hence, instance-based learners never form an explicit general hypothesis regarding the target function.
- Instance-based learners simply compute the classification of each new test instance as needed.

Training data

$$\left(x^{(1)}, y^{(1)}\right), \left(x^{(2)}, y^{(2)}\right), \cdots, \left(x^{(N)}, y^{(N)}\right)$$

Learning

Not Learning anything.

Testing

$$h(x) = y^{(k)}, \text{ where } k = \text{argmin}_{\text{i}}\, D\left(x, x^{(i)}\right)$$

Example: K-Nearest Neighbour algorithm

K-Nearest Neighbour (Classification Algorithm)

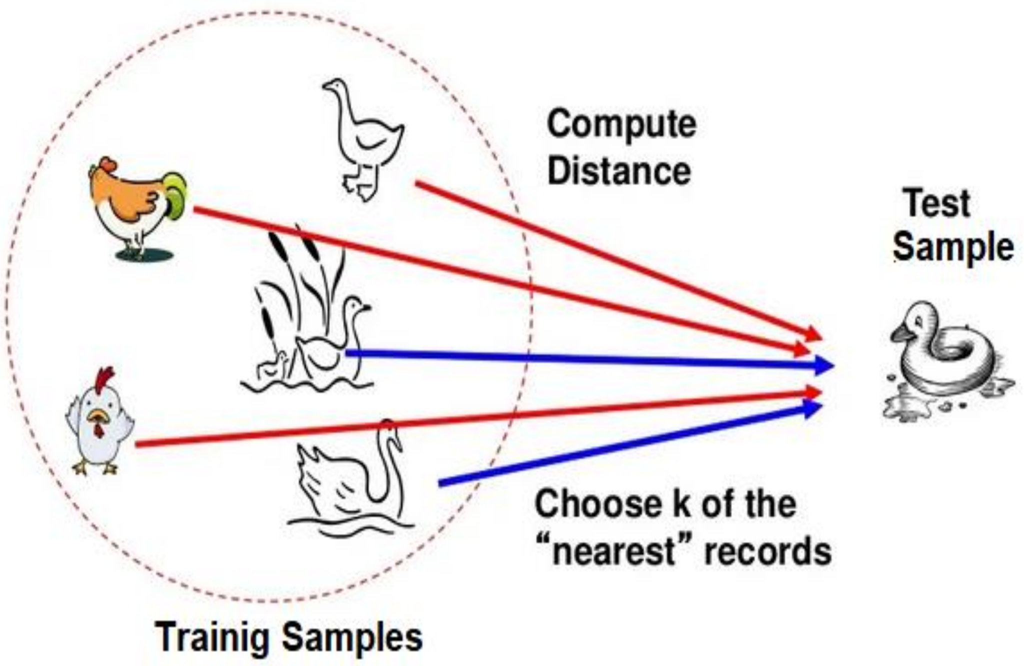
- K-Nearest Neighbour is Supervised Learning technique.
- K-NN is a non-parametric algorithm i.e., it does not make any assumption on underlying data.
- k-NN assumes that all instances are points in some n-dimensional space. k - number of neighbors considered.
- K-NN algorithm considers the similarity between the new test data point and training data points.
- It does not learn from the training dataset at the training phase instead it stores the dataset and it performs an action on the dataset at the time of new test data enters into classification. It is called Lazy learning .
- Assign new test data point into the category that is most similar to the training data points categories.

Example:

- Let consider an image of a creature that looks similar to cat and dog. The kid wants to identify either it is a cat or dog.
- KNN model recognizes (measures) the similar features of the new data point (i.e., cats and dogs images) based on the most similar features to classify either cat or dog category.

k-Nearest Neighbors (Classification Algorithm)

- Initially, Collect the training dataset.
- Select the k (positive) value for nearest neighbors.
- Take a new datapoint from test dataset
  - Apply the distance metric (Euclidean or Manhattan) to calculate the distance of **k** number of **neighbors.**
  - Arrange the calculated distance in ascending order.
  - Consider the k top values from calculated distance.
  - Among these k neighbors, count the number of the data points in each category.
  - Assign the new data points to that category for which the number of the neighbor is maximum.



Distance Metrics

1. Manhattan Distance (L_1 norm or L_1 metric):

- Let consider two points (x_1,y_1) and (x_2,y_2), in an N-dimensional vector space. It is defined as the sum of absolute distance between coordinates in corresponding dimensions $|x_1 - x_2| + |y_1 - y_2|$.
- Manhattan distance between the m vectors $D(X,Y) = \sum_{i=1}^{m} |x_i - y_i|$

- Used in *integrated circuits where wires only run parallel to the X or Y axis.*
- E.g., two vectors, x= (3, 6, 8, 9) and y = (1, 7, 8, 10).
- Manhattan distance = |3 – 1| + |6 – 7| + |8 – 8| + |9 – 10| = 4.

2. Euclidean Distance (L_2 norm or L_2 metric):

- Square root of the sum of the squared differences of the elements in the two vectors using Pythagorean theorem.

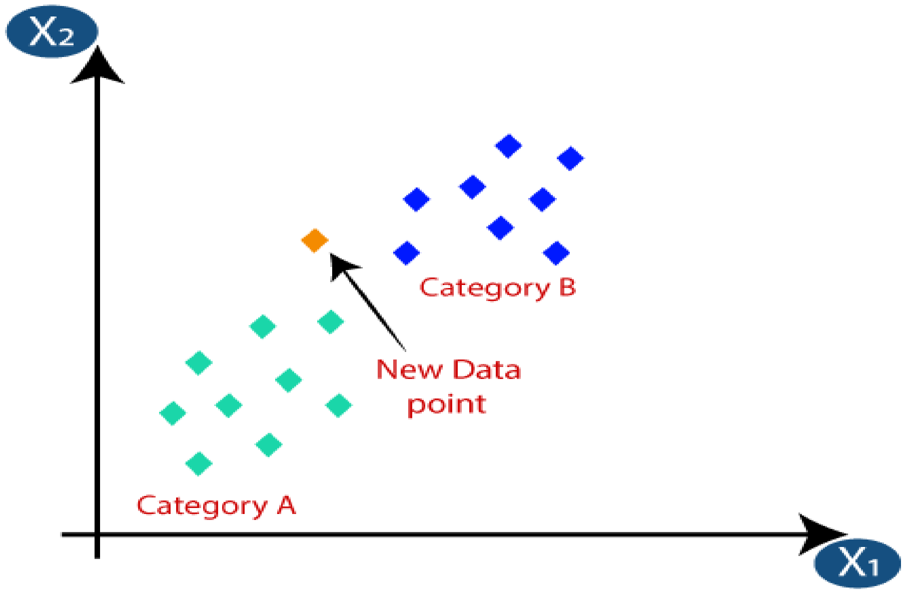
- Euclidean distance between two vectors X and Y is $D(X,Y) = |X - Y| = \sqrt{\sum_{i=1}^{m}\left((x_i - y_i)\right)^2}$
- x_i - X axis values in the coordinate plane and y_i - Y axis in the coordinate plane.

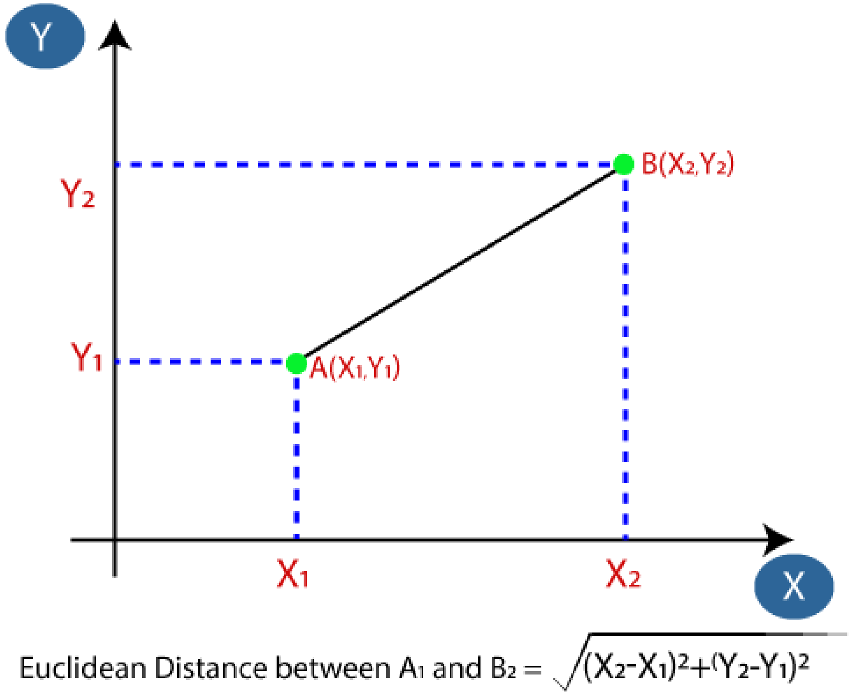
3. Minkowski Distance

- $D(X,Y) = \left(\sum_{i=1}^{m} |x_i - y_i|^r\right)^{1/r}$

K-Nearest Neighbour Example

- Receives new data point:
- Initially, choose the number of neighbors the k=5.

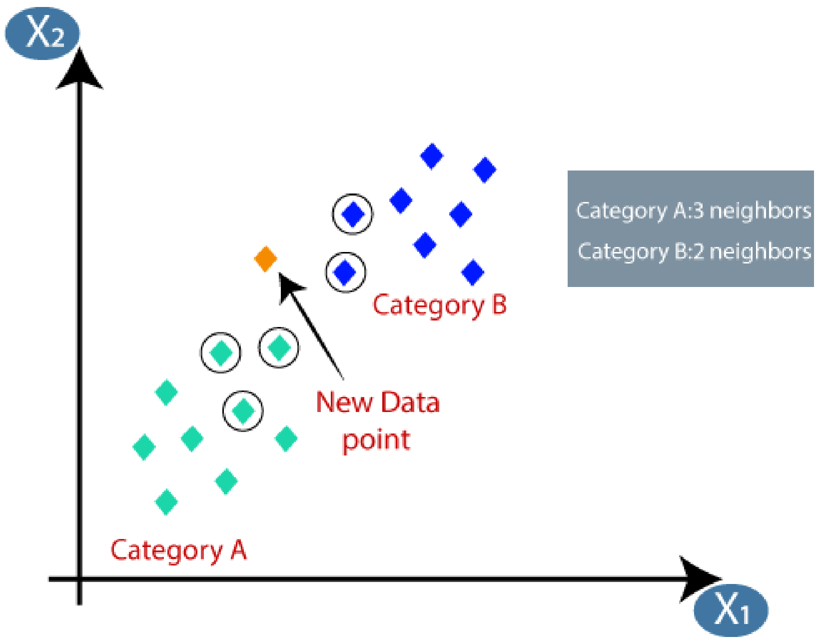




- Calculate the **Euclidean distance** between the new data point and training data points.

K-Nearest Neighbour Example

- Three nearest neighbors in category A and two nearest neighbors in category B.
- So, new data point must belong to category A.



K-Nearest Neighbour Example

- Given Training Dataset

Customer	Age	Salary	No. of Debit cards	Class
David	35	35K	3	No
Anu	22	50K	2	Yes
Hari	63	200K	1	No
Dainty	59	170K	1	No
Sundar	25	40K	4	Yes

- Select k value, here k=3

K-Nearest Neighbour Example

- Select new test data point from test dataset
- {Sam, 37, 50K, 2}.
- Find the class?

Customer	Age	Salary	No. of Debit cards	Class
David	35	35K	3	No
Anu	22	50K	2	Yes
Hari	63	200K	1	No
Dainty	59	170K	1	No
Sundar	25	40K	4	Yes
Sam	**37**	**50K**	**2**	**?**

K-Nearest Neighbour Example, K=3

- Calculate the distance between test datapoint and all training data points.

Customer	Age	Salary	No. of Debit cards	Class	Distance
David	35	35K	3	No	**Sqrt[$(35-37)^2+(35-50)^2+(3-2)^2$] = 15.16**
Anu	22	50K	2	Yes	**Sqrt[$(22-37)^2+(50-50)^2+(2-2)^2$] = 15**
Hari	63	200K	1	No	**Sqrt[$(63-37)^2+(200-50)^2+(1-2)^2$] = 152.23**
Dainty	59	170K	1	No	**Sqrt[$(59-37)^2+(170-50)^2+(1-2)^2$] = 122**
Sundar	25	40K	4	Yes	**Sqrt[$(25-37)^2+(40-50)^2+(4-2)^2$] = 15.74**
Sam	**37**	**50K**	**2**	**?**	

K-Nearest Neighbour Example, K=3

- Arrange the calculated distance in ascending order.

Customer	Age	Salary	No. of Debit cards	Class	Distance
Anu	22	50K	2	Yes	**15**
David	35	35K	3	No	**15.16**
Sundar	25	40K	4	Yes	**15.74**
Dainty	59	170K	1	No	**122**
Hari	63	200K	1	No	**152.23**
Sam	**37**	**50K**	**2**	**?**	

K-Nearest Neighbour Example, K=3

- Consider the **k=3 top values** from calculated distance.

Customer	Age	Salary	No. of Debit cards	Class	Distance
Anu	22	50K	2	Yes	**15**
David	35	35K	3	No	**15.16**
Sundar	25	40K	4	Yes	**15.74**
Dainty	59	170K	1	No	**122**
Hari	63	200K	1	No	**152.23**
Sam	**37**	**50K**	**2**	**?**	

K-Nearest Neighbour Example, K=3

- Among these k neighbors, count the number of the data points in each category.

Customer	Age	Salary	No. of Debit cards	Class	Distance
Anu	22	50K	2	Yes	**15**
David	35	35K	3	No	**15.16**
Sundar	25	40K	4	Yes	**15.74**
Dainty	59	170K	1	No	**122**
Hari	63	200K	1	No	**152.23**
Sam	**37**	**50K**	**2**	**Yes**	

Selecting K value

- 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 computationally expensive.
- Cross Validation test KNN algorithm with different values of K
- From set of K values, can select optimal value.

Data points Split

1. Outliers: Observations that lie at an abnormal distance from all the data points. Most of these are extreme values. Removing these observations will increase the accuracy of the model.

2. Prototypes: Minimum points in training set required to recognize non-outlier points.

3. Absorbed points: These are points that are correctly identified to be non-outlier points.

Practical Implementation

- Data Pre-processing step
- Fitting the K-NN algorithm to the Training set

```
1.from sklearn.neighbors import KNeighborsClassifier

2.classifier= KNeighborsClassifier(n_neighbors=5, metric='minkowski', p=2 )

3.classifier.fit(x_train, y_train)
```

- Predicting the test result
- Test accuracy of the result(Creation of Confusion matrix)
- Visualizing the test set result.

Applications

- Handwritten character classification using nearest neighbour in large databases.
- Fast content-based image retrieval
- Classify program behaviour as normal or intrusive.
- Fault Detection in Semiconductor Manufacturing Processes

Advantages

- It is very simple algorithm to understand and interpret.
- It is very useful for nonlinear data because there is no assumption about data in this algorithm.
- It is a versatile algorithm as it can be used for classification as well as regression.

Disadvantages

- For large data, the prediction stage might be slow.
- k-NN is subject to the curse of dimensionality (i.e., presence of many irrelevant attributes)
- k-NN needs adequate distance measure
- Accuracy depends on the quality of the data
- Sensitive to the scale of the data and irrelevant features
- Require high memory – need to store entire training data and computationally expensive

References

1. Tom M. Mitchell, Machine Learning, McGraw Hill , 2017.

2. EthemAlpaydin, Introduction to Machine Learning (Adaptive Computation and Machine Learning), The MIT Press, 2017.

3. Wikipedia