Different Distance Methods

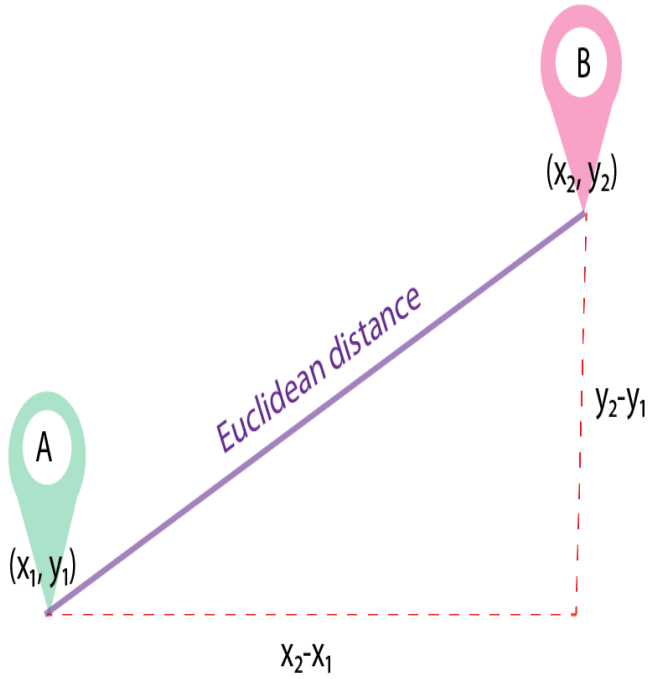
- Euclidean Distance
- Manhattan Distance
- Minkowski Distance
- Hamming Distance
- Cosine Distance

Euclidean Distance

It represents the shortest distance between two points.

$$D_e = \left(\sum_{i=1}^{n} (p_i - q_i)^2 \right)^{1/2}$$

where, n = number of dimensions
pi, qi = data points



https://medium.com/analytics-vidhya/role-of-distance-metrics-in-machine-learning-e43391a6bf2e

Microsoft SAP edunet foundation

Manhattan Distance

The Sum of absolute differences between points across all the dimensions.

$$D_m = \sum_{i=1}^{n} |p_i - q_i|$$

where,

n = number of dimensions

pi, qi = data points

Minkowski Distance

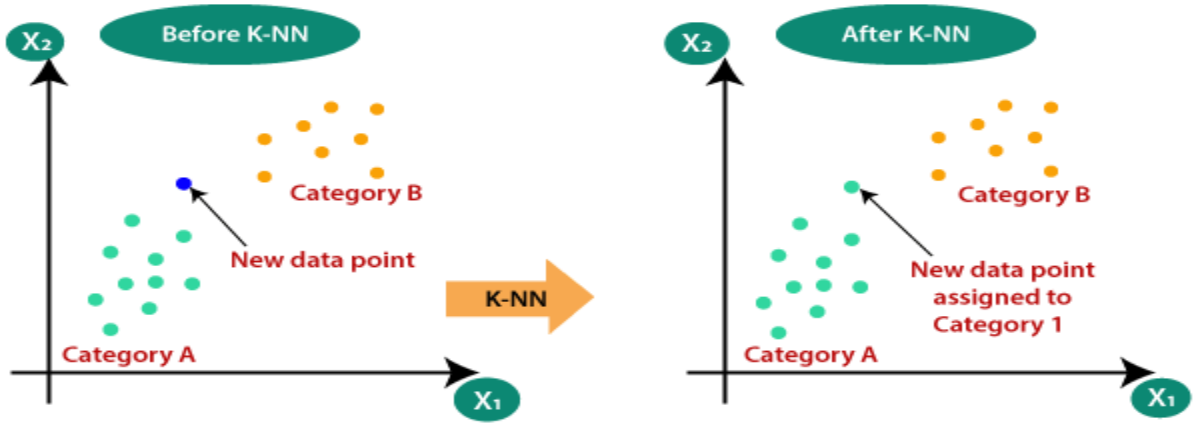
- Minkowski Distance is the generalized form of Euclidean and Manhattan Distance.
- Calculates the distance between two points.

$$D = \left(\sum_{i=1}^{n} |p_i - q_i|^p \right)^{1/p}$$

Where “p” is the order parameter.
p=1: Manhattan distance.
p=2: Euclidean distance.

K-Nearest Neighbor (KNN)

- Supervised Learning technique and Non-parametric algorithm.
- Used for Regression as well as for Classification.
- Assumes the similarity between the new case and available cases.
- Put the new case into the category that is most similar to the available categories.



https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning

K Nearest Neighbors Implementation

- Import the KNeighborsClassifier module.

```
from sklearn.neighbors import KNeighborsClassifier
```

- Create KNN classifier object

```
model = KNeighborsClassifier(n_neighbors=3)
```

- Fit your model

```
# Train the model using the training sets
model.fit(features,label)
```

- Prediction

```
#Predict Output
predicted= model.predict([[0,2]])
print(predicted)
```

REFERENCES

1. https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn
2. https://medium.com/analytics-vidhya/role-of-distance-metrics-in-machine-learning-e43391a6bf2e

THANK YOU