K-Nearest-Neighbours (KNN)

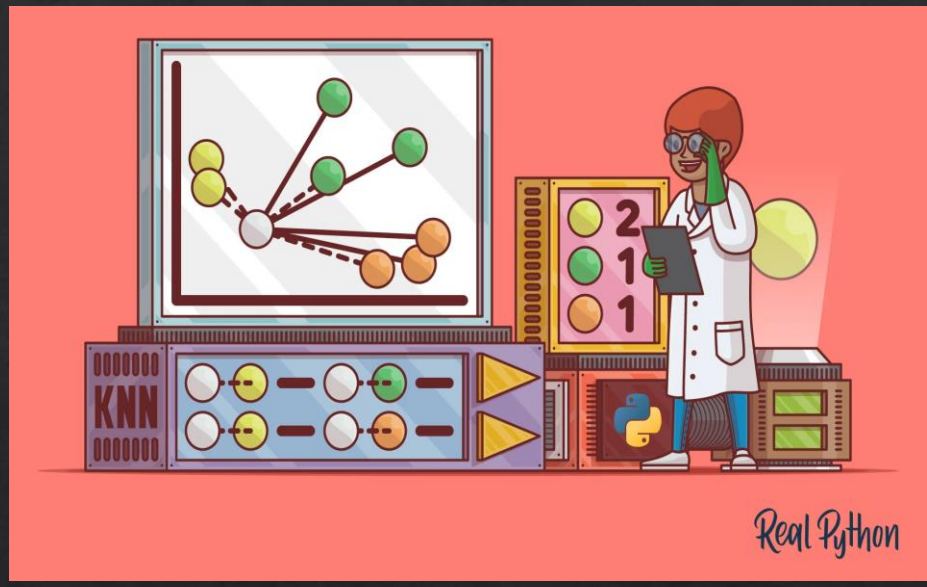
Presented by : Muhammad Zaqeem

Overview:

- K-NN
- How K-NN Works in classification and regression
- Distance Metric
- Choosing the best K
- Why K-NN

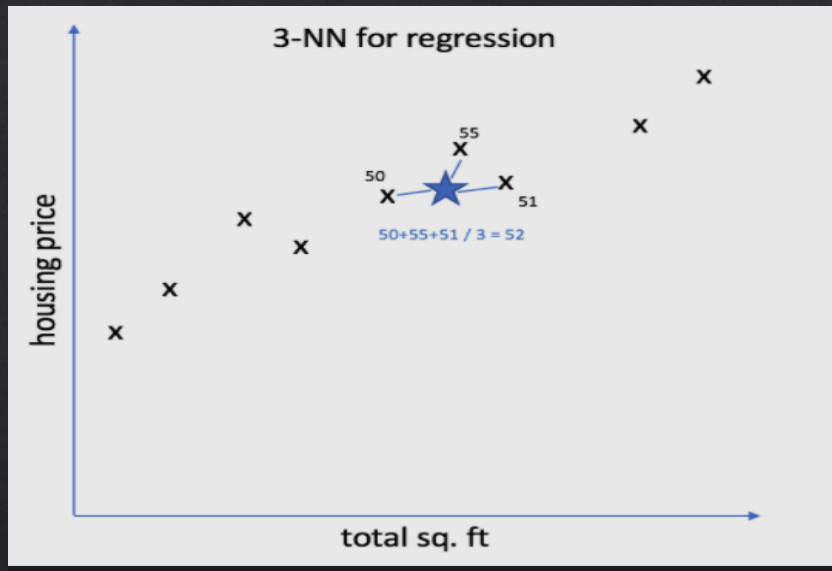
K-NN Algorithm:

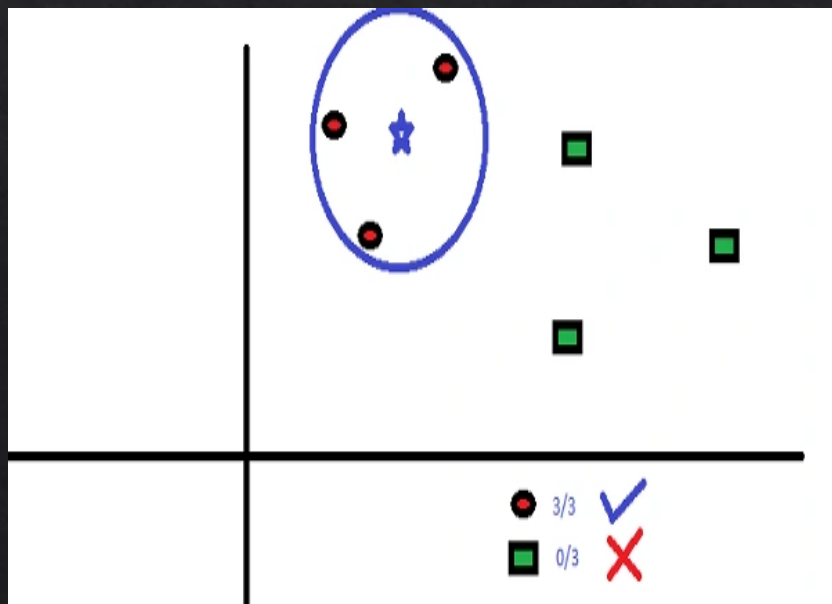
- K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a **non-parametric** and **lazy learner** algorithm.
- It is a type of **instance-based learning**, meaning that it makes decisions based on the similarity to existing data
- The KNN algorithm is straightforward and easy to understand, making it a popular choice in various domains.



How it Works:

- **Choosing K value:** First, you select the number of neighbors (K) you want to consider. Typically, K is a small integer like 3 or 5.
- **Measure Distance**: To determine how close the new data point is to the training data, the algorithm calculates the **distance** between the new data point and all other points in the dataset.
- **Find Nearest Neighbors:** Once the distances are calculated, the algorithm selects the K points with the smallest distance.
- For **regression**: It calculates the average of the values of the K nearest neighbors.
- For **classification**: It picks the most common class among the K nearest neighbors.

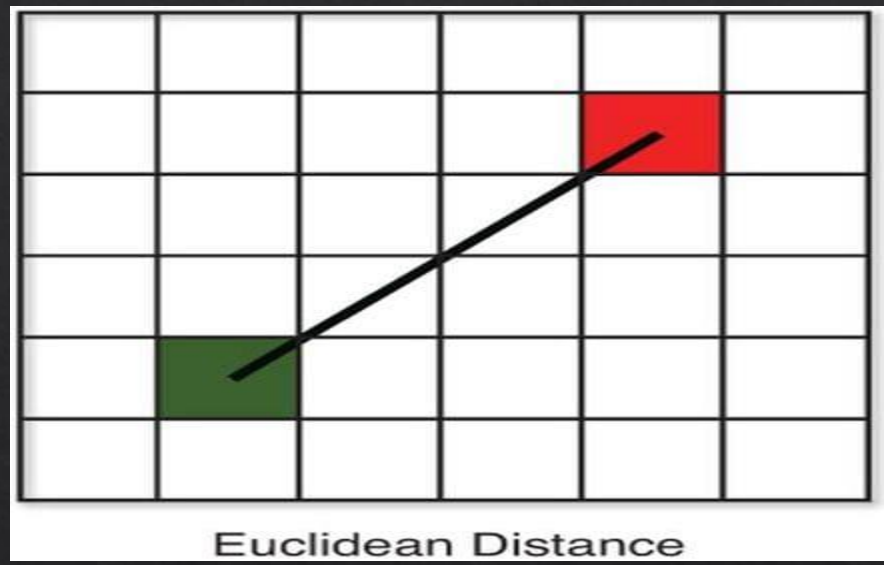




Distance Metric:

Euclidean Distance: This is the default distance metric for Scikit-Learn's KNN method. This distance measures the truest straight distance between two points.

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$



Manhattan Distance: This distance is also known as taxicab distance or city block distance, that is because the way this distance is calculated. The distance between two points is the sum of the absolute differences of their Cartesian coordinates

$$d = \sum_{i=1}^{n} |x_i - y_i|$$

Choosing the best K value:

- The parameter k in KNN signifies the number of nearest neighbors to consider when making predictions for a specific query point
- The choice of k is crucial as it impacts the model's accuracy and generalization.

To define an appropriate k:

- **Elbow Method:** Plot the error rate or accuracy against various k values and identify the point of diminishing returns, often referred to as the "elbow." This can help pinpoint a suitable k value.
- Square Root of N rule: This rule offers a quick and practical way to determine an initial k value for your KNN model, especially when no other domain-specific knowledge or optimization techniques are readily available. The rule suggests setting k to the square root of N. Here, N represents the total number of data points in the dataset.
- **K value should be odd:** Using an **odd k value** helps avoid ties in the KNN algorithm. This way, when the model counts the "votes" from neighbors, there's a clear majority, making it easier to decide on the predicted class

Why K-NN

- KNN can be used in both regression and classification predictive problems. However, when it comes to industrial problems, it's mostly used in classification since it fairs across all parameters evaluated when determining the usability of a technique
- **Prediction Power**
- **Calculation Time**
- **Ease to Interpret the Output**
- KNN algorithm fairs across all parameters of considerations. But mostly, it is used due to its ease of interpretation and low calculation time.

	Logistic Regression	CART	Random Forest	KNN
1. Ease to interpret output	2	3	1	3
2. Calculation time	3	2	1	3
3. Predictive Power	2	2	3	2