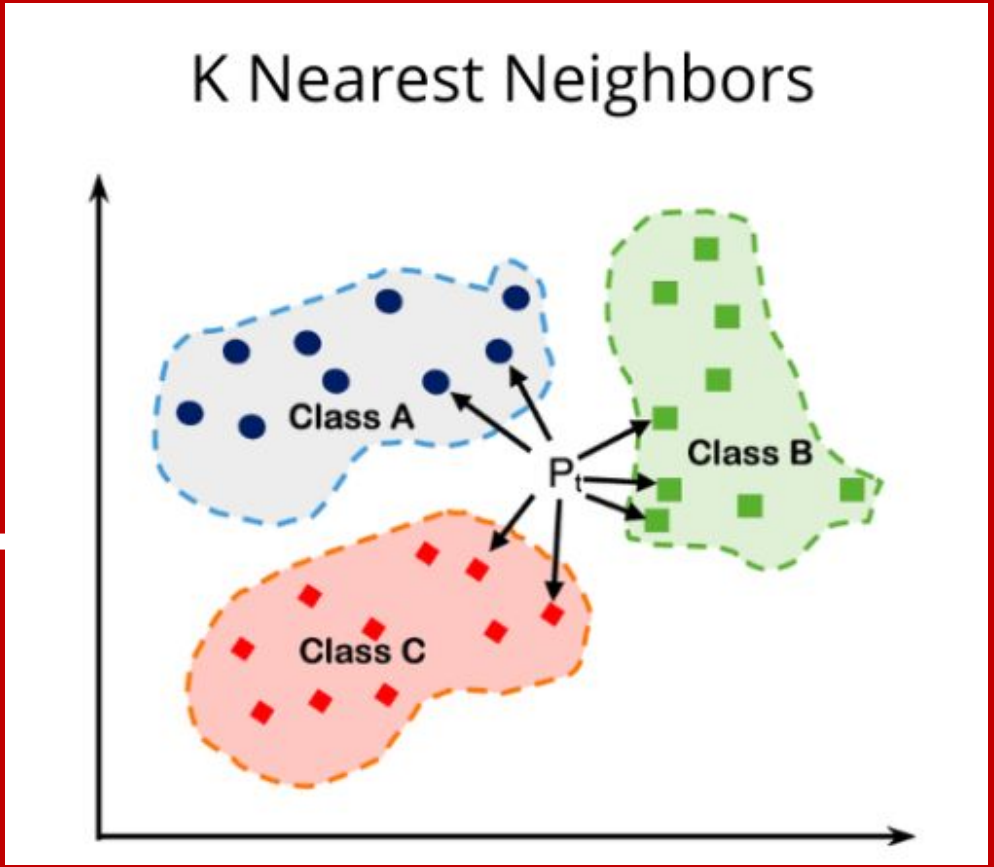
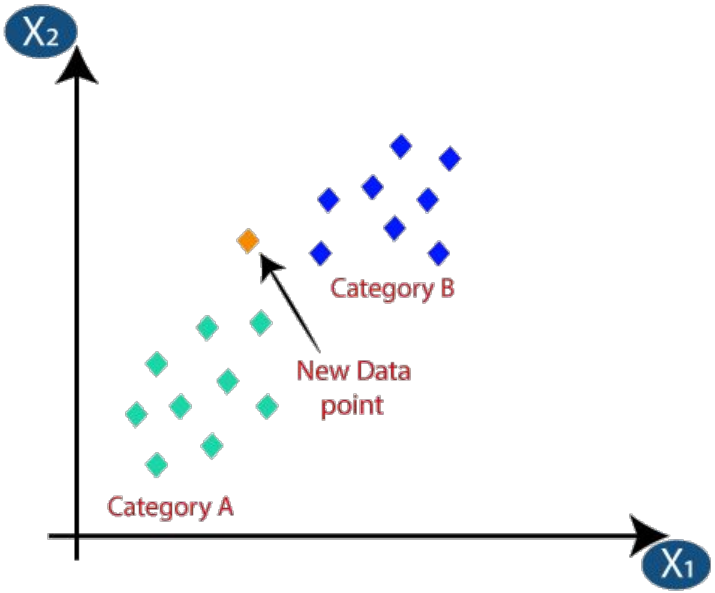
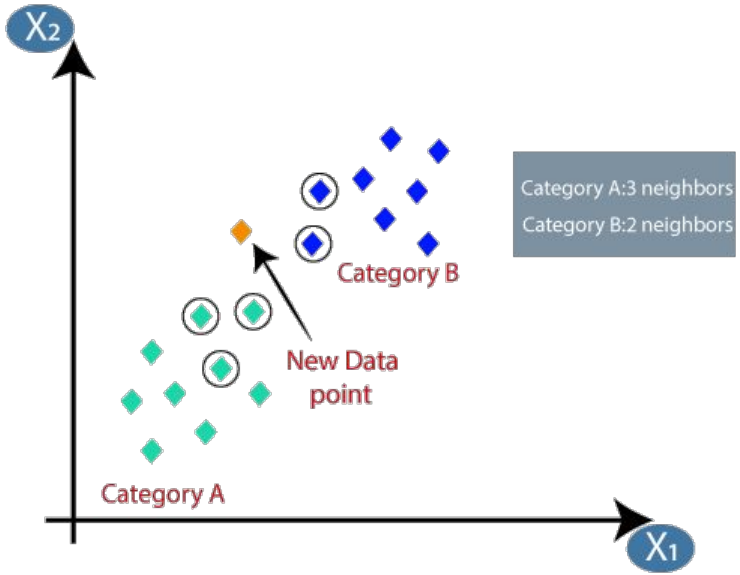
KNN

- قل لي من هم أقرب جيرانك، وسأخبرك من أنت



KNN - lazy learner algorithm





K-Nearest Neighbors (Non-Parametric Approach) classification

- Step 1 − load the training as well as test data.
- Step 2 − Next, we need to choose the value of K

Choosing K:

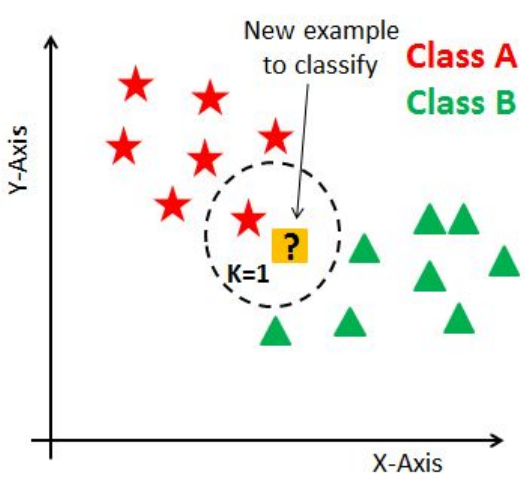
- The value of K is critical; it controls the bias-variance trade-off
- Small K: Low bias, high variance (more flexible, but sensitive to noise)
- Large K: High bias, low variance (smoother, but may overlook local patterns)

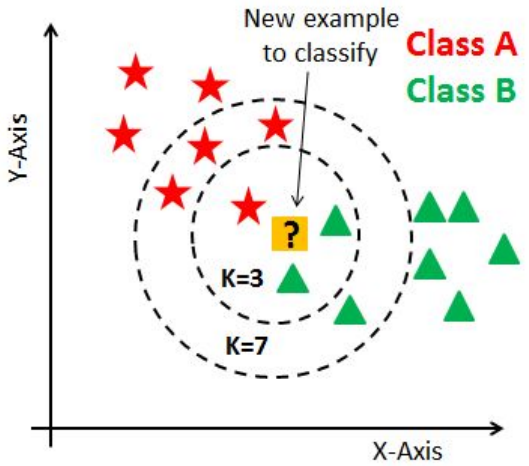
The choice of K affects the bias-variance trade-off.

3.2 − Now, based on the distance value, sort them in ascending order.

- 3.3 − Next, it will choose the top K rows from the sorted array.
- 3.4 − Now, it will assign a class to the test point based on most frequent class of these rows

How do you Decide the Number of Neighbors in KNN?





K-Nearest Neighbors Regression Example

Given Data Points:

X	Y
1	1.5
2	1.7
3	3.1
4	2.8
5	3.6
6	3.9
7	5.1
8	5.3
9	6.7
10	6.9

- **Choose K:** Let's select $K = 3$
- **Prediction Point x_0:** Let's predict the value at $x_0 = 5.5$

Calculate the distance from $x_0 = 5.5$ to all other points.

X	Y	Distance to 5.5
1	1.5	$\|5.5 - 1\| = 4.5$
2	1.7	$\|5.5 - 2\| = 3.5$
3	3.1	$\|5.5 - 3\| = 2.5$
4	2.8	$\|5.5 - 4\| = 1.5$
5	3.6	$\|5.5 - 5\| = 0.5$
6	3.9	$\|5.5 - 6\| = 0.5$
7	5.1	$\|5.5 - 7\| = 1.5$
8	5.3	$\|5.5 - 8\| = 2.5$
9	6.7	$\|5.5 - 9\| = 3.5$
10	6.9	$\|5.5 - 10\| = 4.5$

Identify the K=3 closest points to $x_0 = 5.5$:

X	Y	
5	3.6	$(\text{Distance} = 0.5)$
6	3.9	$(\text{Distance} = 0.5)$
4	2.8	$(\text{Distance} = 1.5)$

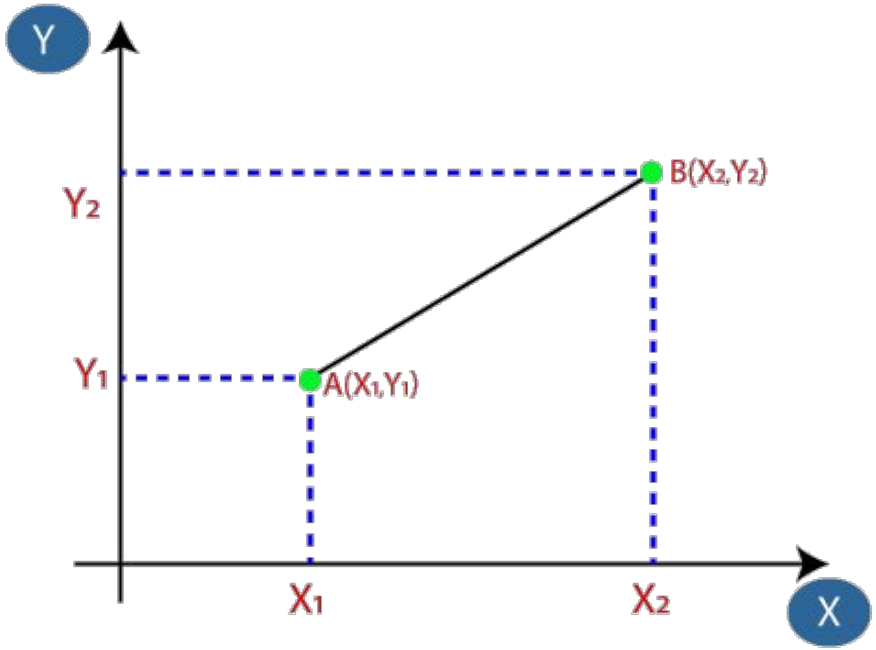
Calculate the average of the responses (Y values) of the K=3 nearest neighbors.

$$\hat{f}(5.5) = \frac{3.6 + 3.9 + 2.8}{3} = \frac{10.3}{3} = 3.43$$

How do you Decide the Number of Neighbors in KNN?

- Defining k can be a balancing act as different values can lead to overfitting or underfitting.
- Lower values of k can have **high variance**, but **low bias** [overfitting].
- larger values of k may lead to **high bias** and **lower variance** [underfitting].
- The choice of **k** will largely depend on the input data as data with more outliers or noise will likely perform better with higher values of k.
- Overall, it is recommended to have an odd number for **k** to avoid ties in classification, and cross-validation tactics can help you choose the optimal **k** for your dataset.

Euclidean Distance



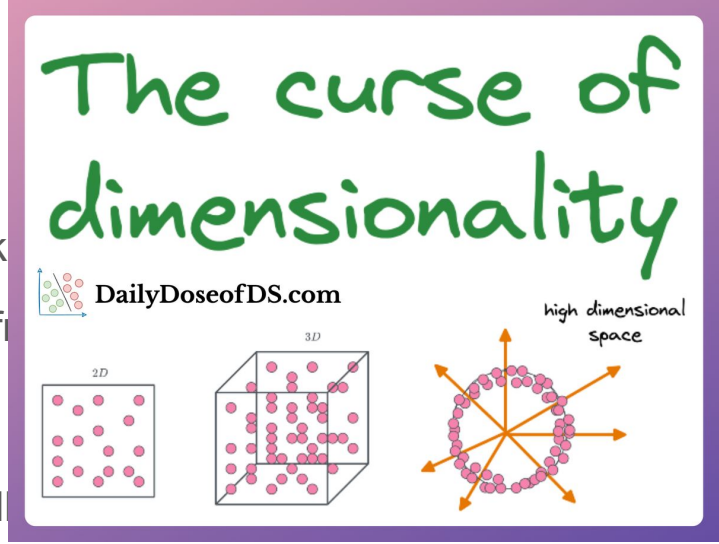
Euclidean Distance between A_1 and $B_2 = \sqrt{(X_2-X_1)^2+(Y_2-Y_1)^2}$

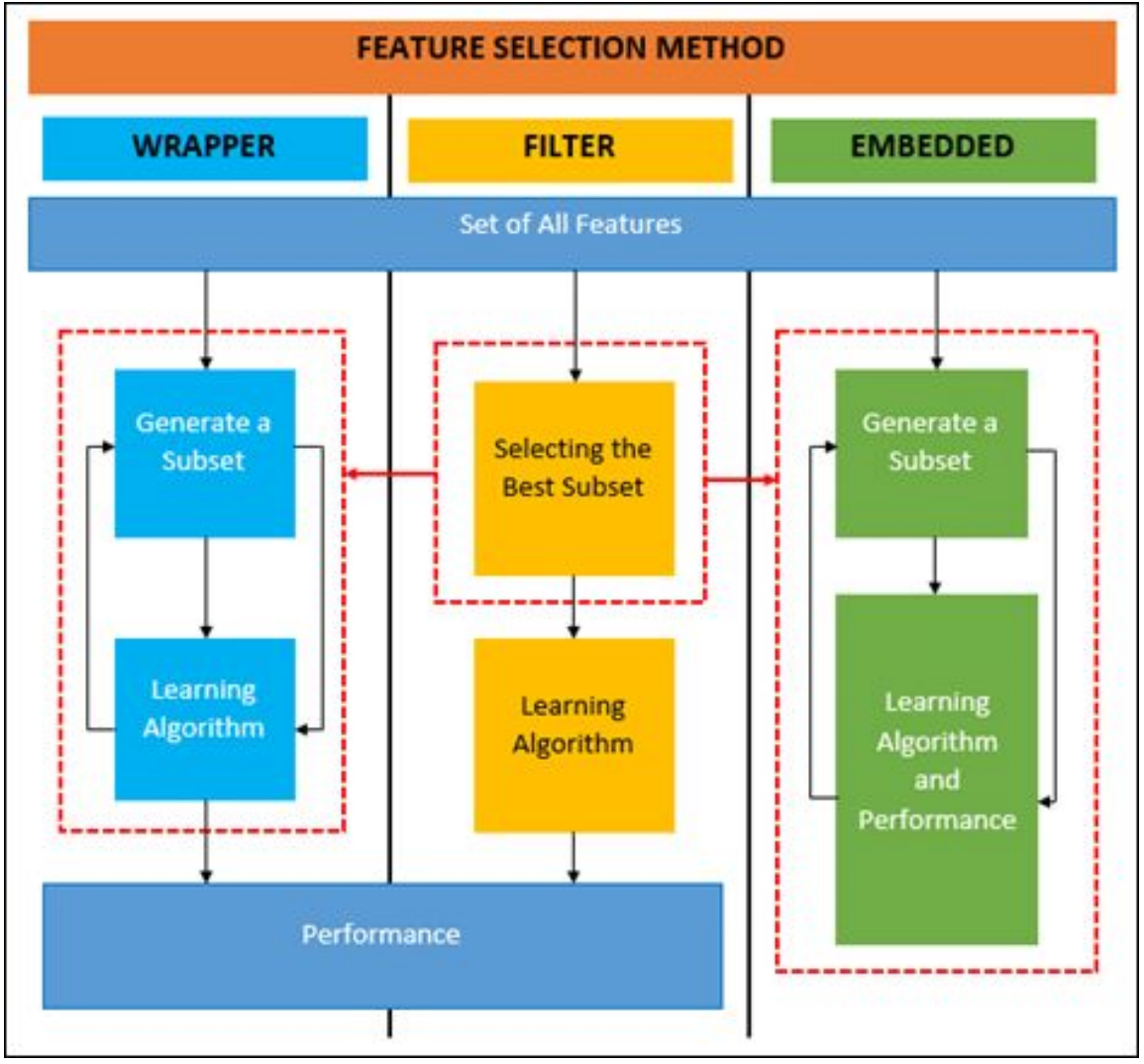
Advantages

- **Easy to implement**
- **Adapts easily**: As new training samples are added, the algorithm adjusts to account for any new data since all training data is stored into memory.
- **Few hyperparameters**: KNN only requires a k value and a distance metric.

Disadvantages

- **Does not scale well**: Since KNN is a lazy algorithm, it tak
  storage compared to other classifiers. This can be costly f
  perspective.
- **Curse of dimensionality**: The KNN algorithm tends to fal
  dimensionality, which means that it doesn't perform well with high-dimensional data inputs.
- **Prone to overfitting**: Due to the "curse of dimensionality", KNN is also more prone to overfitting. While feature selection and dimensionality reduction techniques are leveraged to prevent this from occurring, the value of k can also impact the model's behavior.




FEATURE SELECTION METHOD
WRAPPER
FILTER
EMBEDDED
Set of All Features
Generate a Subset
Learning Algorithm
Selecting the Best Subset
Learning Algorithm
Generate a Subset
Learning Algorithm and Performance
Performance

Dimension reduction (Feature extraction)

	1	2	3	4	5	6	7		200
	Height	Weight	Average blood pressure	Average heart rate	BMI	Cholesterol levels	Average cigarettes/day	...	Sugar levels
Person 1	150	80	140/90	63	36	5.0	0		99
Person 2	174	90	90/60	100	32	4.1	0		95
Person 3	183	109	120/80	95	29	3.6	1		92
Person 4	186	95	123/75	84	28	4.8	5		89
Person 5	170	67	95/60	76	23	2.7	10		100
Person 6	180	82	92/60	78	25	3.7	10		112
Person 7	165	71	124/80	81	26	3.8	0		113
Person 8	172	78	97/70	90	24	3.4	0		100
...									
Person 20	190	75	90/60	78	21	4.2	0		82

200 FACTORS (VARIABLES)

PCA

PC1	PC2	PC3	PC4	PC5
-1	3	-1	4	4
2	4	2	5	5
3	2	4	2	2
4	4	5	-4	-4
5	5	2	2	5
2	5	-4	3	2
-4	-6	5	5	-4
-3	-6	-6	2	5
8	-3	-6	-3	-6

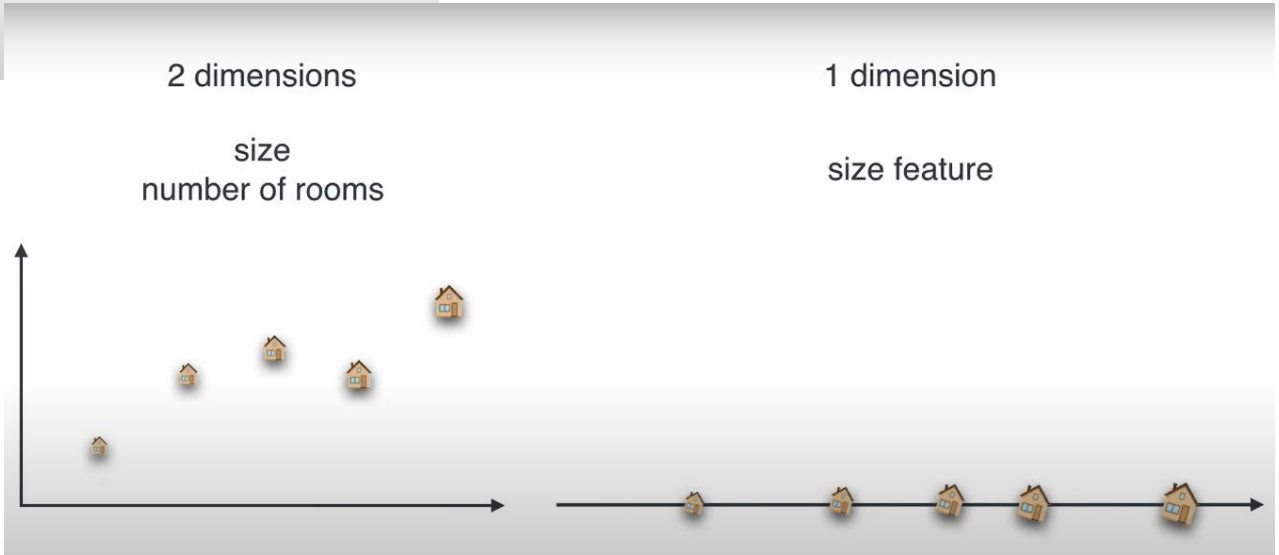
5 PRINCIPAL COMPONENTS

Housing Data

Dimension reduction (Feature extraction)

Size
Number of rooms
Number of bathrooms
→ Size feature

Schools around
Crime rate
→ Location feature



Required Data Preparation for KNN

Computational Complexity:

- Can be computationally expensive, especially with large datasets, because it requires calculating distances to all training points for each prediction

Normalization:

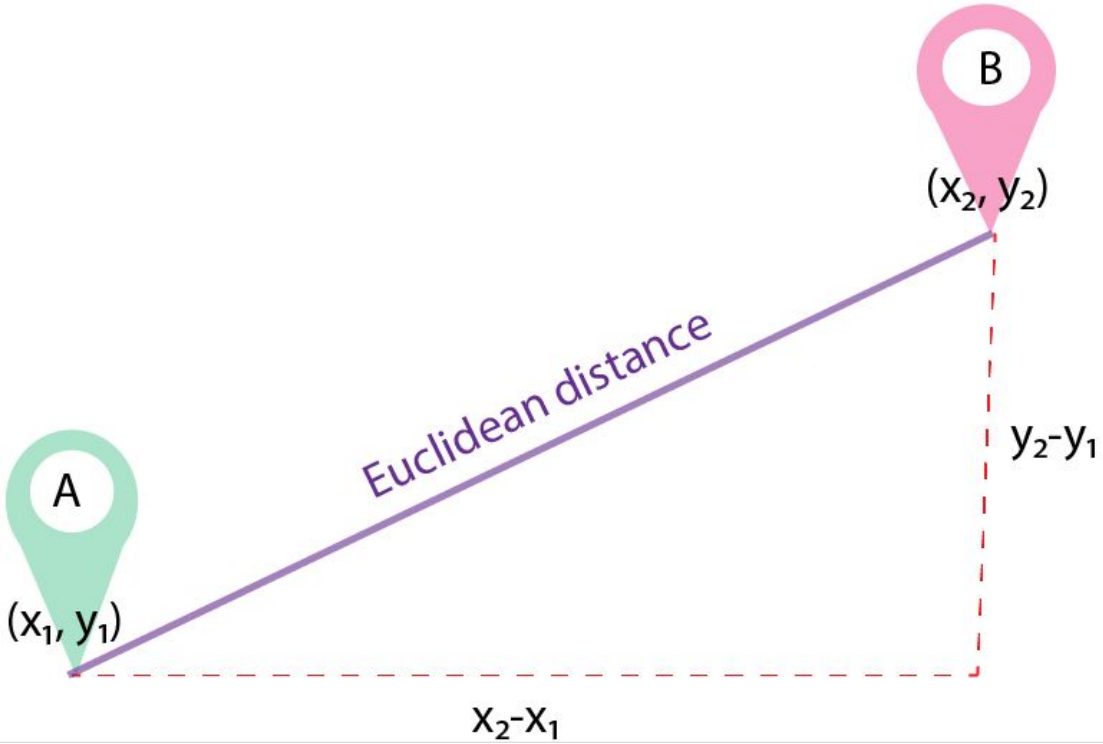
- Feature scaling (normalization) is important to ensure all features contribute equally to the distance calculations

Handling High Dimensionality:

- Performance can degrade in high-dimensional spaces (curse of dimensionality)
- Dimensionality reduction techniques (e.g., PCA) can be helpful

Distance Measures for KNN

K-Nearest Neighbors (KNN) uses distance measures to determine the proximity of data points to one another. Different distance metrics affect how KNN classifies the data points.



K-Nearest Neighbors (KNN)

Distance Measures for KNN

1. Manhattan Distance

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

Explanation

- Manhattan distance is the sum of the absolute differences of their Cartesian coordinates.
- It can be visualized as "grid-like" distances, where movement is allowed only along the grid lines (no diagonal movement).

K-Nearest Neighbors (KNN)

Distance Measures for KNN

2. Euclidean Distance

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

Explanation

- Euclidean distance measures the straight-line distance between two points in Euclidean space.
- This is the most commonly used distance measure in KNN.

K-Nearest Neighbors (KNN)

Distance Measures for KNN

3. Minkowski Distance

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

Explanation

- Minkowski distance is a generalized distance metric. It can represent both Manhattan and Euclidean distances based on the value of p:
  - When p=1, it represents Manhattan distance.
  - When p=2, it represents Euclidean distance.
- Minkowski distance is useful in vector spaces and can generalize to higher dimensions.

K-Nearest Neighbors (KNN)

Distance Measures for KNN

4. Cosine Distance/Similarity

Formula $\text{Cosine Similarity} = \cos(\theta) = \frac{a \cdot b}{|a||b|}$

Explanation

- Cosine similarity is used to measure the **similarity** between two vectors by calculating the cosine of the angle between them.
    - 1 indicates vectors pointing in the same direction.
    - 0 indicates orthogonal vectors (no similarity).
    - -1 indicates vectors pointing in opposite directions.
- Cosine **distance** is calculated as $\text{Cosine Distance} = 1 - \cos(\theta)$

Usage

- This metric is commonly used in text analysis and document comparison.

K-Nearest Neighbors (KNN)

Distance Measures for KNN

5. Hamming Distance

Formula

- Hamming distance is calculated by counting the number of bit positions in which two binary strings differ.

Explanation

- It's used primarily for comparing two **binary data strings** of equal length.

Example

- If the strings are "ABCDE" and "AGDDF," the Hamming distance is 3 because three positions differ.

متى افكر استخدم KNN

- **حسب نوع المشكلة**
  - Regression or Classifaction
- **طبيعة البيانات**
  - بيانات بسيطة وغير معقدة
  - **no linear**
  - Similar scale
- **اياك ثم اياك تستخدمها مع داتا بابعاد كثير (Curse of dimensionality) يعني عدد الفيتشر يكون كبير (مينفعششششش)**
  - ممكن تحل هاي المشكلة باستخدام الطرق الموجودة بالسلايدات المرفقة
- حساااااااااااس جدا ل outliers, Noise