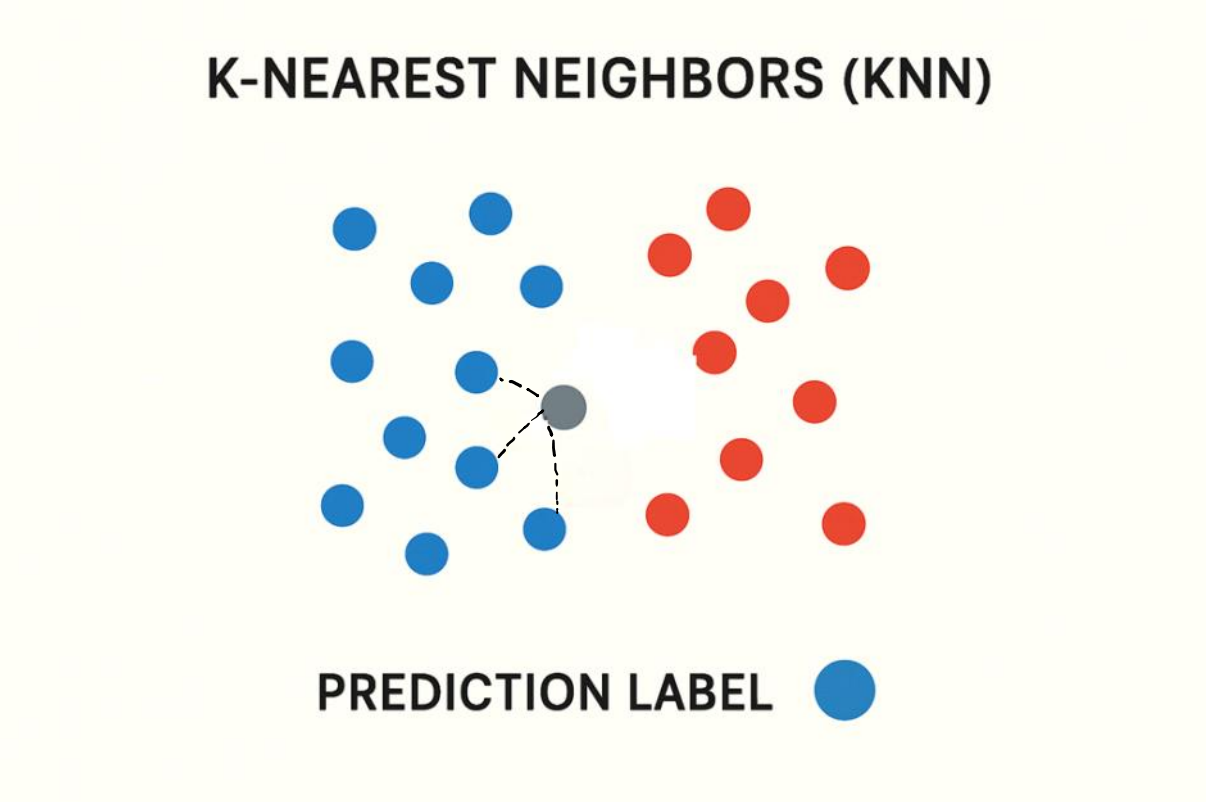
KNN (K-Nearest Neighbors)

23 August 2025 16:27



Agenda

23 August 2025 16:39

1. **Concept/Introduction of KNN**

2. **Geometric Intuition**

3. **How KNN Works**

3. **Numerical Example for Classification Problem**

4. **Numerical Example for Regression Problem**

5. **Advances and Disadvantages**

Concept/Introduction of KNN

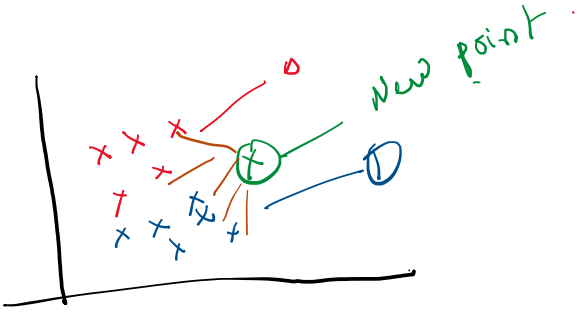
23 August 2025 16:39

- **KNN (K-Nearest Neighbors)** is a **supervised learning algorithm** used for both **classification** and **regression**.
- It is **non-parametric** (no assumptions about data distribution).
- It is a **lazy learner** (does not learn a model in training; instead, stores training data and uses it during prediction).

👉 **Key Idea**:

- To predict the output for a new data point, look at the **K nearest points** from training data.
- Use **majority vote** (classification) or **mean value** (regression).

Geometric Intuition



How KNN Works: Algorithm

23 August 2025 16:41

Steps:

1. Choose the number of neighbors (**K**).
2. Measure the **distance** between the new point and all training points.
3. Pick the **K nearest points**.
4. - **Classification**: Assign the majority class among neighbors.
   - **Regression**: Take the average of the neighbors' values.

Numerical Example for Classification Problem

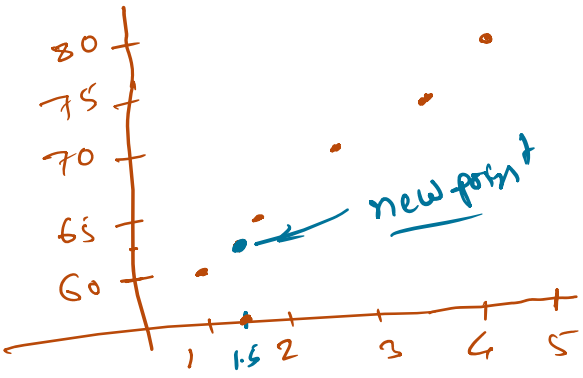
23 August 2025 16:42

Predict whether a student will pass or fail an exam based on their:

- **Study Hours per day**
- **Attendance (%)**

Sample Dataset:

Student	Study Hours	Attendance (%)	Result (Pass=1, Fail=0)
S1	1	60	0 (Fail)
S2	2	65	0 (Fail)
S3	3	70	1 (Pass)
S4	4	75	1 (Pass)
S5	5	80	1 (Pass)
S6	1.5	62	0 (Fail)



① $k = 3$

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

③ new (1.5, 62) → distance points (S_1, S_2, S_3, S_4, S_5)

new → (1.5, 62) x_1, y_2

From S_1 (1, 60) x_2, y_2

$d_1 = |1 - 1.5| + |60 - 62|$

$= 0.5 + 2 = 2.5$

From S_2 (2, 65)

$d_2 = |2 - 1.5| + |65 - 62|$

$= 0.5 + 3 = 3.5$

From S_3 (3, 70)

$= |3 - 1.5| + |70 - 62|$

From $S_3(3, 70)$

$$d_3 = |3-1.5| + |70-62|$$

$$= 1.5 + 8 = 9.5$$

From $S_4(4, 75)$

$$= |4-1.5| + |75-62|$$

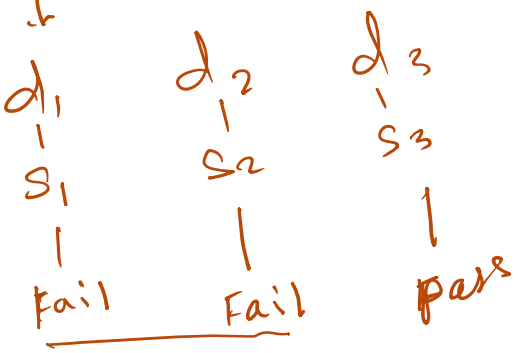
$$d_4 = 2.5 + 13 = 15.5$$

From $S_5 = (5, 80)$

$$= |5-1.5| + |80-62|$$

$$d_5 = 3.5 + 18 = 21.5$$

$d_1 = 2.5$, $d_2 = 3.5$, $d_3 = 9.5$, $d_4 = 15.5$, $d_5 = 21.5$



(4) Majority voting

Fail - 2
pass - 1
} -> Fail

Numerical Example for Regression Problem

23 August 2025 16:43

Problem: Predict **house price** based on the house **size (in sqft)**

Sample Dataset :

Point	Size (sqft)	Price (lacs)
H1	800	15.0
H2	950	17.0
H3	1100	20.0
H4	1300	23.0
H5	1500	30.0

1200

Query : **size = 1200 sqft**, use **K = 3**.

1. K = 3
2. Distance(new point) → Existing points
3. Sort according
4. top (K)
5. get the o/p values of selected point
6. calculate the Average of values

↓

Final prediction = Avg

Point	Size (sqft)	Price (lacs)
H1	800	15.0
H2	950	17.0
H3	1100	20.0
H4	1300	23.0
H5	1500	30.0

1200

K = 3

H_1 : $|1200 - 800| = 400$

H_2 : $|1200 - 950| = 250$

H_3 : $|1200 - 1100| = 100$

H_4 : $|1200 - 1300| = 100$

H_5 : $|1200 - 1500| = 300$

$H_3(100), H_4(100), H_2(250)$

↓

$$\frac{20 + 23 + 17}{k = 3} = \frac{60}{3} = 20 \text{ lacs}$$

Advantages & Disadvantages

23 August 2025 16:43

☑ Advantages

- Simple, intuitive.
- Works well with small datasets.
- No training phase (fast to fit).

❌ Disadvantages

- **Slow prediction for large datasets** → because distance to all points must be computed.
- **Sensitive to irrelevant features/scales** → because distances get skewed.
- **Curse of dimensionality** → nearest neighbors become meaningless in high dimensions.

When to Use KNN

- Baseline model for classification/regression.
- Small to medium datasets.
- When interpretability is important.

🔑 Key Takeaway

- KNN is simple but powerful.
- Must use **scaling**.
- Choosing the right **K** is critical.
- Great for teaching ML concepts and for small datasets.

Distance Metrics

23 August 2025 16:50

(a) Euclidean Distance

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

(b) Manhattan Distance

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

(c) Minkowski Distance (general form)

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

- $p = 1 \rightarrow$ Manhattan
- $p = 2 \rightarrow$ Euclidean

(d) Cosine Similarity (for text data)

$$\text{sim}(x, y) = \frac{x \cdot y}{||x||\,||y||}$$