📅 DAY 20 — K-Nearest Neighbors (KNN)

Goal : Understand instance-based learning and distance-based decisions

---

1️⃣ What Is KNN?

KNN is a **lazy learning algorithm**.

> It does NOT build a model.
> It stores the data and decides at prediction time.

Simple Idea:

1. Look at the **K nearest data points**
2. Let them vote
3. Pick the majority class (or average for regression)

2️⃣ Why KNN is Different

Algorithm	Learns Model?
Linear / Logistic	Yes
KNN	No

KNN:

- No training phase
- Expensive prediction
- Memory heavy

3️⃣ Distance Metric

KNN depends entirely on **distance**.

Common distances:

- Euclidean (default)
- Manhattan

- Minkowski

```
distance = sqrt((x1 - x2)**2 + (y1 - y2)**2)
```

Distance = similarity

closer = more similar

4️⃣ Why Scaling Is CRITICAL for KNN 🔥

If features are not scaled:

```
salary ( 100000) dominates age (30)
```

Result:

- Distance is meaningless
- Wrong neighbors selected

Always scale features before KNN

5️⃣ Choosing K

K value	Effect
Small K (1-3)	Noisy, overfitting
Large K	Smooth, underfitting

Rule of thumb:

```
K ≈ sqrt(n)
```

6️⃣ Classification vs Regression with KNN

Task	Decision
Classification	Majority vote
Regression	Average value

same algorithm, different output.

7️⃣ Bias-Variance Tradeoff in KNN

- Small K → Low bias, high variance
- Large K → High bias, low variance

This is the **heart of ML tradeoffs**.

8️⃣ When KNN Works Well (IMPORTANT)

✔️ Small datasets

✔️ Low-dimensional data

✔️ Clean data

❌ Large datasets

❌ High dimensions (curse of dimensionality)

9️⃣ Implementation KNN in Sklearn

```
from sklearn.neighbors import KNeighborsClassifier

model = KNeaighborsClasifier(n_neighbors=5)
model.fit(X_train, y_train)
```

🔟 KNN in Real Life (Intuition)

- Recommendation systems
- Similar product search
- Image similarity
- Face recognition (early systems)