**Mathematical Algorithms**

The **K-Nearest Neighbors (K-NN)** algorithm is a **supervised machine learning** technique used for **classification** and **regression**. It works by finding the **k closest data points** (neighbors) to a given input and making predictions based on their values.

**✅ How K-NN Works**

1. Choose a value of **k** (number of neighbors to consider).
2. Calculate the **distance** (e.g., Euclidean) between the new point and all other points in the dataset.
3. Select the **k closest neighbors**.
4. **For classification**: Take a **majority vote** among the k neighbors' labels.
5. **For regression**: Take the **average** of the k neighbors' values.

**📘 Example (Classification)**

**🔹 Dataset**

| **ID** | **Height (cm)** | **Weight (kg)** | **Class** |
| --- | --- | --- | --- |
| 1 | 170 | 65 | Fit |
| 2 | 160 | 70 | Unfit |
| 3 | 180 | 80 | Fit |
| 4 | 175 | 85 | Unfit |

**🔹 New Input:**

text

CopyEdit

Height = 172 cm, Weight = 75 kg

**🧮 Step-by-Step (with k = 3)**

We use **Euclidean distance**:

Distance=(x1−x2)2+(y1−y2)2\text{Distance} = \sqrt{(x\_1 - x\_2)^2 + (y\_1 - y\_2)^2}Distance=(x1​−x2​)2+(y1​−y2​)2​

For each record:

* **To ID 1**: √((172–170)² + (75–65)²) = √(4 + 100) = √104 ≈ 10.2
* **To ID 2**: √((172–160)² + (75–70)²) = √(144 + 25) = √169 = 13
* **To ID 3**: √((172–180)² + (75–80)²) = √(64 + 25) = √89 ≈ 9.43
* **To ID 4**: √((172–175)² + (75–85)²) = √(9 + 100) = √109 ≈ 10.44

✅ Nearest neighbors (k=3): IDs 3, 1, 4

Classes: Fit, Fit, Unfit → **Majority = Fit**

🔸 **Prediction**: Fit

**📌 Code Example (sklearn)**

python

CopyEdit

import pandas as pd

from sklearn.neighbors import KNeighborsClassifier

# Data

data = pd.DataFrame({

'Height': [170, 160, 180, 175],

'Weight': [65, 70, 80, 85],

'Class': ['Fit', 'Unfit', 'Fit', 'Unfit']

})

# Features and labels

X = data[['Height', 'Weight']]

y = data['Class']

# Train K-NN model

model = KNeighborsClassifier(n\_neighbors=3)

model.fit(X, y)

# Predict for a new person

prediction = model.predict([[172, 75]])

print("Prediction:", prediction[0]) # Output: Fit

**✅ Summary**

| **Feature** | **K-NN** |
| --- | --- |
| Type | Supervised (used for classification & regression) |
| Core idea | “Tell me who your neighbors are, and I’ll tell you who you are” |
| Sensitive to | Feature scaling (requires normalization or standardization) |
| Lazy learning | No model training; prediction happens during inference |