Only for Reference and basic understanding of flow

Dataset:

x1	x2	Class (y)
1	2	A
2	3	A
3	3	A
6	5	B
7	6	B
8	8	B

- **Test point (x1 = 5, x2 = 5)**

Step-by-Step Process:

Step 1: Calculate the Euclidean Distance

Step 1: Calculate the Euclidean Distance

To calculate the Euclidean distance between a test point $P(x_1, x_2)$ and each data point $D(x_1, x_$ we use the formula:

$$d(P, D) = \sqrt{(x_{1,test} - x_{1,data})^2 + (x_{2,test} - x_{2,data})^2}$$

Let's calculate the distances between the test point (5, 5) and each of the training points.

1. **Distance to (1, 2)**:

$$d((5,5),(1,2)) = \sqrt{(5-1)^2 + (5-2)^2} = \sqrt{16+9} = \sqrt{25} = 5$$

2. **Distance to (2, 3)**:

$$d((5,5),(2,3)) = \sqrt{(5-2)^2 + (5-3)^2} = \sqrt{9+4} = \sqrt{13} \approx 3.61$$

3. **Distance to (3, 3)**:

$$d((5,5),(3,3)) = \sqrt{(5-3)^2 + (5-3)^2} = \sqrt{4+4} = \sqrt{8} \approx 2.83$$

4. **Distance to (6, 5)**:

$$d((5,5),(6,5)) = \sqrt{(5-6)^2 + (5-5)^2} = \sqrt{1+0} = \sqrt{1} = 1$$

5. **Distance to (7, 6)**:

$$d((5,5),(7,6)) = \sqrt{(5-7)^2 + (5-6)^2} = \sqrt{4+1} = \sqrt{5} \approx 2.24$$

6. **Distance to (8, 8)**:

$$d((5,5),(8,8)) = \sqrt{(5-8)^2 + (5-8)^2} = \sqrt{9+9} = \sqrt{18} \approx 4.24$$

Step 2: Find the 3 Nearest Neighbors

The distances we calculated are:

- **(1, 2)**: Distance = 5
- **(2, 3)**: Distance ≈ 3.61
- **(3, 3)**: Distance ≈ 2.83
- **(6, 5)**: Distance = 1
- **(7, 6)**: Distance ≈ 2.24
- **(8, 8)**: Distance ≈ 4.24

Now, we need to find the **3 closest neighbors**. From the distances, the 3 smallest are:

- (6, 5) with distance = 1
- (7, 6) with distance ≈ 2.24
- (3, 3) with distance ≈ 2.83

Step 3: Determine the Class of the Test Point

The classes of the 3 nearest neighbors are:

- (6, 5) → **Class B**
- (7, 6) → **Class B**
- (3, 3) → **Class A**

Now, we take the majority vote from the 3 nearest neighbors:

- **Class B** appears 2 times
- **Class A** appears 1 time

Since **Class B** is the majority, the **predicted class** for the test point (5, 5) is **Class B**.

Final Answer:

- The test point (5, 5) is predicted to belong to **Class B**.

KNN PROGRAM

```
# Import necessary libraries
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# Step 1: Load the Iris dataset
iris = load_iris()
X = iris.data  # Features
y = iris.target  # Target labels

# Step 2: Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Step 3: Feature scaling (important for KNN)
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Step 4: Initialize the KNN classifier
# You can change the value of k here
k = 3
knn = KNeighborsClassifier(n_neighbors=k)

# Step 5: Train the model
knn.fit(X_train, y_train)
```

```
# Step 6: Make predictions on the test set
y_pred = knn.predict(X_test)

# Step 7: Evaluate the model's performance
accuracy = accuracy_score(y_test, y_pred)
print(accuracy)
```