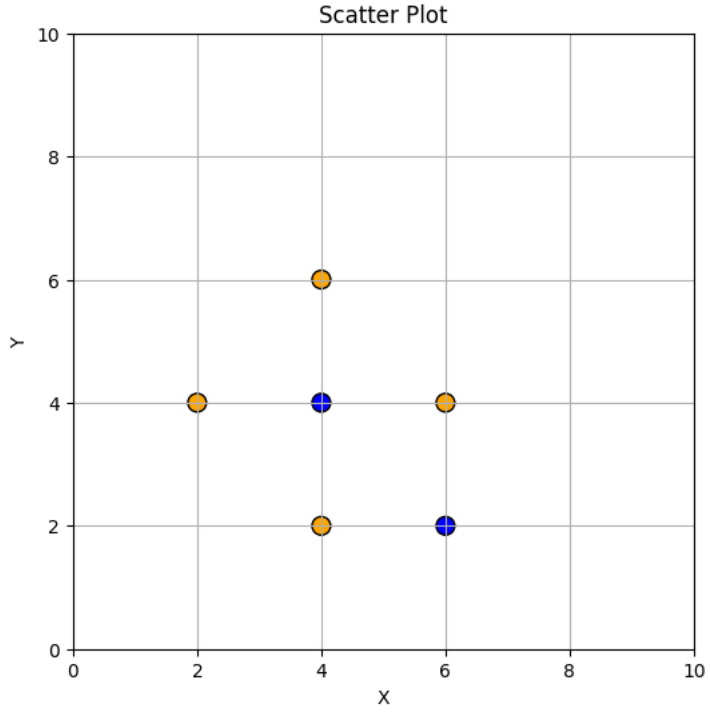
ASSIGNMENT NO : 7

Code :

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import LabelEncoder
data = {
   "X": [2, 4, 4, 6, 4, 6],
   "Y": [4, 2, 6, 4, 4, 2],
   "Class": ["Orange", "Orange", "Orange", "Orange", "Blue", "Blue"]
}
df = pd.DataFrame(data)

# Save as CSV
df.to_csv("knn_dataset.csv", index=False)
print(df)
# Assign colors based on class labels
color_map = {"Orange": "orange", "Blue": "blue"}
df["Color"] = df["Class"].map(color_map)

# Create scatter plot
plt.figure(figsize=(6, 6))
plt.scatter(df["X"], df["Y"], c=df["Color"], s=100, edgecolors='black')

# Grid and labels
plt.xticks(range(0, 11, 2))
plt.yticks(range(0, 11, 2))
plt.xlabel("X")
plt.ylabel("Y")
plt.title("Scatter Plot")
plt.grid(True)

# Show plot
plt.show()

X = df[['X', 'Y']]  # Independent variables (features)
y = df['Class']      # Target variable (labels)
le = LabelEncoder()
y_encoded = le.fit_transform(y)
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X, y_encoded)

# Predict class for point (6,6)
new_point =new_point = pd.DataFrame([[6, 6]], columns=['X', 'Y'])
predicted_class = knn.predict(new_point)
# Convert prediction back to original class label
predicted_label = le.inverse_transform(predicted_class)
```

print(f"The predicted class for point (6,6) is: {predicted_label[0]}")
Output :



The predicted class for point (6,6) is: Orange