## Exercise 5

Given a dataset that contains information about different types of flowers (e.g., Iris dataset), perform classification using the k-Nearest Neighbors (kNN) algorithm. Evaluate the performance of the model by calculating its accuracy and visualize the results using appropriate techniques.

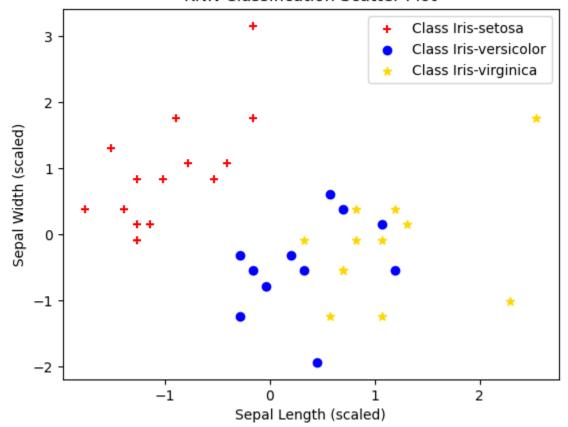
```
In [1]: import matplotlib.pyplot as plt
        import pandas as pd
        from sklearn.metrics import accuracy_score
        from sklearn.neighbors import KNeighborsClassifier
        from sklearn.model selection import train test split
        from sklearn.preprocessing import LabelEncoder, StandardScaler
In [2]: | df = pd.read_csv("iris_dataset.csv")
        df.head()
Out[2]:
           sepal_length sepal_width petal_length petal_width
                                                              target
        0
                    5.1
                               3.5
                                           1.4
                                                       0.2 Iris-setosa
                   4.9
                               3.0
                                                      0.2 Iris-setosa
                                           1.4
         2
                   4.7
                               3.2
                                           1.3
                                                      0.2 Iris-setosa
         3
                   4.6
                               3.1
                                           1.5
                                                      0.2 Iris-setosa
                   5.0
                               3.6
                                           1.4
                                                      0.2 Iris-setosa
In [3]: X = df[["sepal_length", "sepal_width", "petal_length", "petal_width"]]
        Y = df["target"]
In [4]: X train, X test, Y train, Y test = train test split(X, Y, test size=0.25,
        random state=42)
In [5]: | scaler = StandardScaler()
        X train scaled = scaler.fit transform(X train)
        X test scaled = scaler.transform(X test)
In [6]: encoder = LabelEncoder()
        Y_train_enc = encoder.fit_transform(Y_train)
        Y_test_enc = encoder.transform(Y_test)
In [7]: knn = KNeighborsClassifier(n_neighbors=3)
```

1 of 2 1/9/25, 16:44

knn.fit(X train scaled, Y train enc)

```
Out[7]:
                                          i ?
               KNeighborsClassifier
         KNeighborsClassifier(n_neighbors=3)
In [8]: Y pred = knn.predict(X test scaled)
 In [9]: | accuracy = accuracy_score(Y_test_enc, Y_pred)
         print(f"The KNN Classifier is {accuracy * 100:.0f}% accurate")
        The KNN Classifier is 100% accurate
In [10]: labels = encoder.classes_
         markers = ["+", "o", "*"]
         colors = ["red", "blue", "gold"]
         for i, label in enumerate(labels):
             class_points = (Y_pred == i)
             plt.scatter(X_test_scaled[class_points, 0], X_test_scaled[class_points, 1],
         label=f'Class {label}', marker=markers[i], color=colors[i])
             plt.title("KNN Classification Scatter Plot")
             plt.xlabel("Sepal Length (scaled)")
             plt.ylabel("Sepal Width (scaled)")
             plt.legend()
```

## KNN Classification Scatter Plot



2 of 2 1/9/25, 16:44