## DV Laboratory Part B - Exercise 3

Develop a k-nearest neighbours classifier model based on the Iris dataset. Make use of visualization tools such as scatter plots to visualize the data.

```
In [1]: import matplotlib.pyplot as plt
        from sklearn.model selection import train test split
        from sklearn.neighbors import KNeighborsClassifier
        import pandas as pd
        from sklearn.metrics import accuracy score
        from sklearn.preprocessing import LabelEncoder
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
         1
                   4.9
                               3.0
                                           1.4
                                                      0.2 Iris-setosa
                   4.7
                                                      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 fields = ["sepal length", "sepal width", "petal length", "petal width"]
        X = df[x fields]
        y = df["target"]
In [4]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42
In [5]: knn = KNeighborsClassifier(n neighbors=3)
        knn.fit(X train, y train)
Out[5]:
                                          i ?
               KNeighborsClassifier
        KNeighborsClassifier(n_neighbors=3)
In [6]: y pred = knn.predict(X test)
In [7]: | accuracy = accuracy score(y test, y pred)
        print(f"Accuracy = {accuracy * 100} %")
        label_encoder = LabelEncoder()
        y test encoded = label encoder.fit transform(y test)
        y pred encoded = label encoder.transform(y pred)
       Accuracy = 100.0 %
```

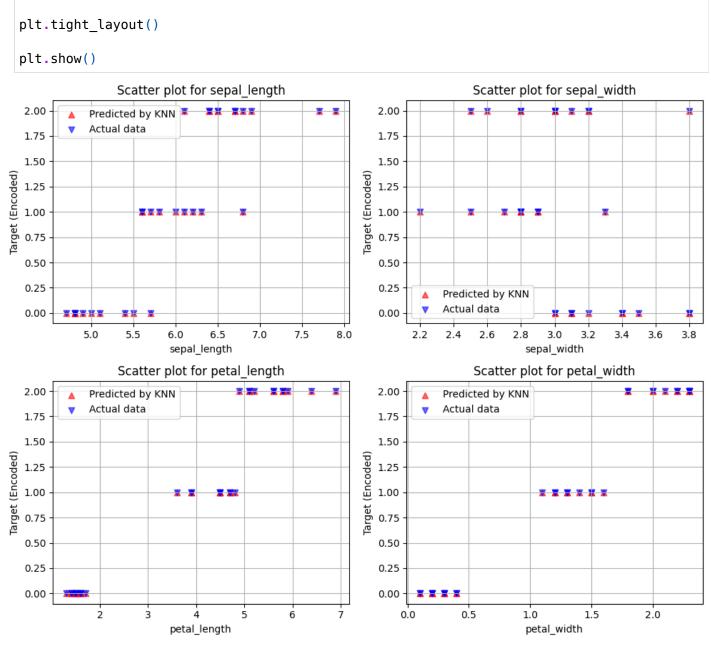
1 of 2 12/24/24, 18:00

```
In [8]: fig, axes = plt.subplots(2, 2, figsize=(10, 8))
    axes = axes.flatten()

for i, field in enumerate(x_fields):
        axes[i].scatter(X_test[field], y_pred_encoded, color="red", marker="^", label="Prediaxes[i].scatter(X_test[field], y_test_encoded, color="blue", marker="v", label="Actuaxes[i].set_title(f"Scatter plot for {field}")
        axes[i].set_xlabel(f"{field}")
        axes[i].set_ylabel("Target (Encoded)")
        axes[i].legend()
        axes[i].grid(True)

plt.tight_layout()

plt.show()
```



2 of 2 12/24/24, 18:00