File Edit View Insert Runtime Tools Help

ommands + Code + Text

[1]
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

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
```

[2]
```
iris = load_iris()
X = pd.DataFrame(iris.data, columns=iris.feature_names)
y = iris.target
```

Commands + Code + Text

```
[2]  iris = load_iris()
     X = pd.DataFrame(iris.data, columns=iris.feature_names)
     y = iris.target

     # Optional: To view dataset
     df = X.copy()
     df['target'] = y
     df.head()
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
[3]  scaler = StandardScaler()
     X_scaled = scaler.fit_transform(X)
```

```
[4]  X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)
```

```
[5]  # Try different K values
     for k in range(1, 11):
         knn = KNeighborsClassifier(n_neighbors=k)
         knn.fit(X_train, y_train)
         y_pred = knn.predict(X_test)
         print(f"K = {k}, Accuracy = {accuracy_score(y_test, y_pred):.2f}")
```

```
K = 1, Accuracy = 0.97
K = 2, Accuracy = 1.00
K = 3, Accuracy = 1.00
K = 4, Accuracy = 1.00
K = 5, Accuracy = 1.00
K = 6, Accuracy = 1.00
K = 7, Accuracy = 1.00
K = 8, Accuracy = 1.00
K = 9, Accuracy = 1.00
K = 10, Accuracy = 1.00
```

What can I help you build?

```
[6]  knn = KNeighborsClassifier(n_neighbors=3)
     knn.fit(X_train, y_train)
     y_pred = knn.predict(X_test)

     print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
     print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

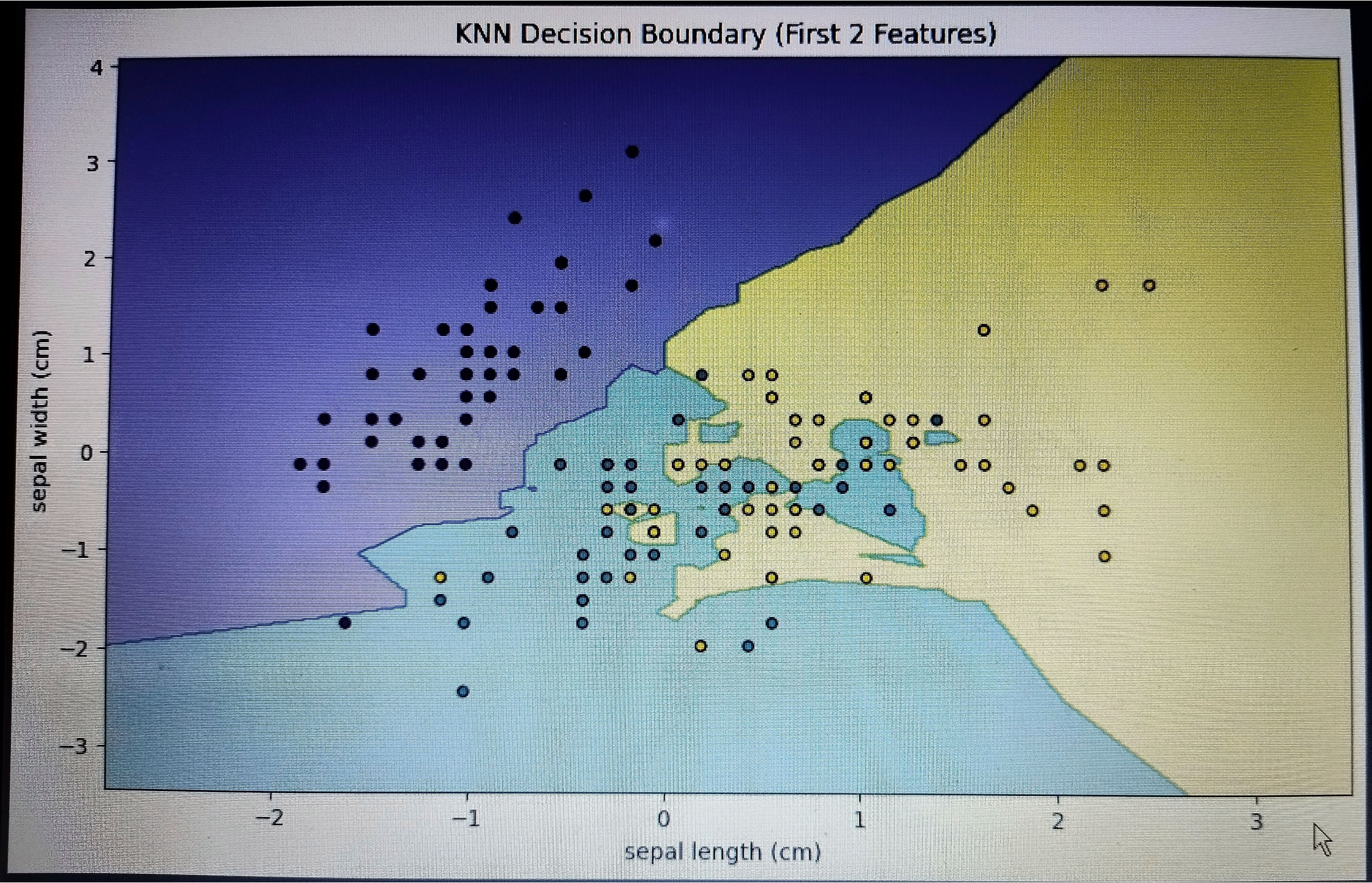
```
Confusion Matrix:
 [[10  0  0]
 [ 0  9  0]
 [ 0  0 11]]

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        10
           1       1.00      1.00      1.00         9
           2       1.00      1.00      1.00        11

    accuracy                           1.00        30
   macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30
```

nmands + Code + Text

[7]

```
X_train_simple, X_test_simple, y_train_simple, y_test_simple = train_test_split(X_simple, y, test_size=0.2, random_state=42)

knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train_simple, y_train_simple)

# Plotting
h = .02
x_min, x_max = X_simple[:, 0].min() - 1, X_simple[:, 0].max() + 1
y_min, y_max = X_simple[:, 1].min() - 1, X_simple[:, 1].max() + 1
xx, yy = np.meshgrid(np.arange(x_min, x_max, h),
                     np.arange(y_min, y_max, h))

Z = knn.predict(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)

plt.figure(figsize=(10, 6))
plt.contourf(xx, yy, Z, alpha=0.4)
plt.scatter(X_simple[:, 0], X_simple[:, 1], c=y, edgecolor='k', s=20)
plt.title("KNN Decision Boundary (First 2 Features)")
plt.xlabel(iris.feature_names[0])
plt.ylabel(iris.feature_names[1])
plt.show()
```

What can I help you build?


KNN Decision Boundary (First 2 Features)
sepal width (cm)
sepal length (cm)
4
3
2
1
0
−1
−2
−3
−2
−1
0
1
2
3