In [1]:

*# Import required libraries*

**import** pandas **as** pd

**import** seaborn **as** sns

**import** matplotlib.pyplot **as** plt

**from** sklearn.datasets **import** load\_iris

**from** sklearn.ensemble **import** RandomForestClassifier

**from** sklearn.model\_selection **import** train\_test\_split

**from** sklearn.metrics **import** accuracy\_score, classification\_report, confusion\_mat

**from** sklearn.preprocessing **import** StandardScaler

**from** sklearn.decomposition **import** PCA

In [2]:

iris **=** load\_iris()

df **=** pd**.**DataFrame(data**=**iris**.**data, columns**=**iris**.**feature\_names) df['species'] **=** iris**.**target

df['species'] **=** df['species']**.**apply(**lambda** x: iris**.**target\_names[x])

print("First 5 rows of the dataset:") print(df**.**head())

First 5 rows of the dataset:

sepal length (cm) sepal width (cm) petal length (cm) petal width (cm) \

0 5.1 3.5 1.4 0.2

1 4.9 3.0 1.4 0.2

2 4.7 3.2 1.3 0.2

3 4.6 3.1 1.5 0.2

4 5.0 3.6 1.4 0.2

species

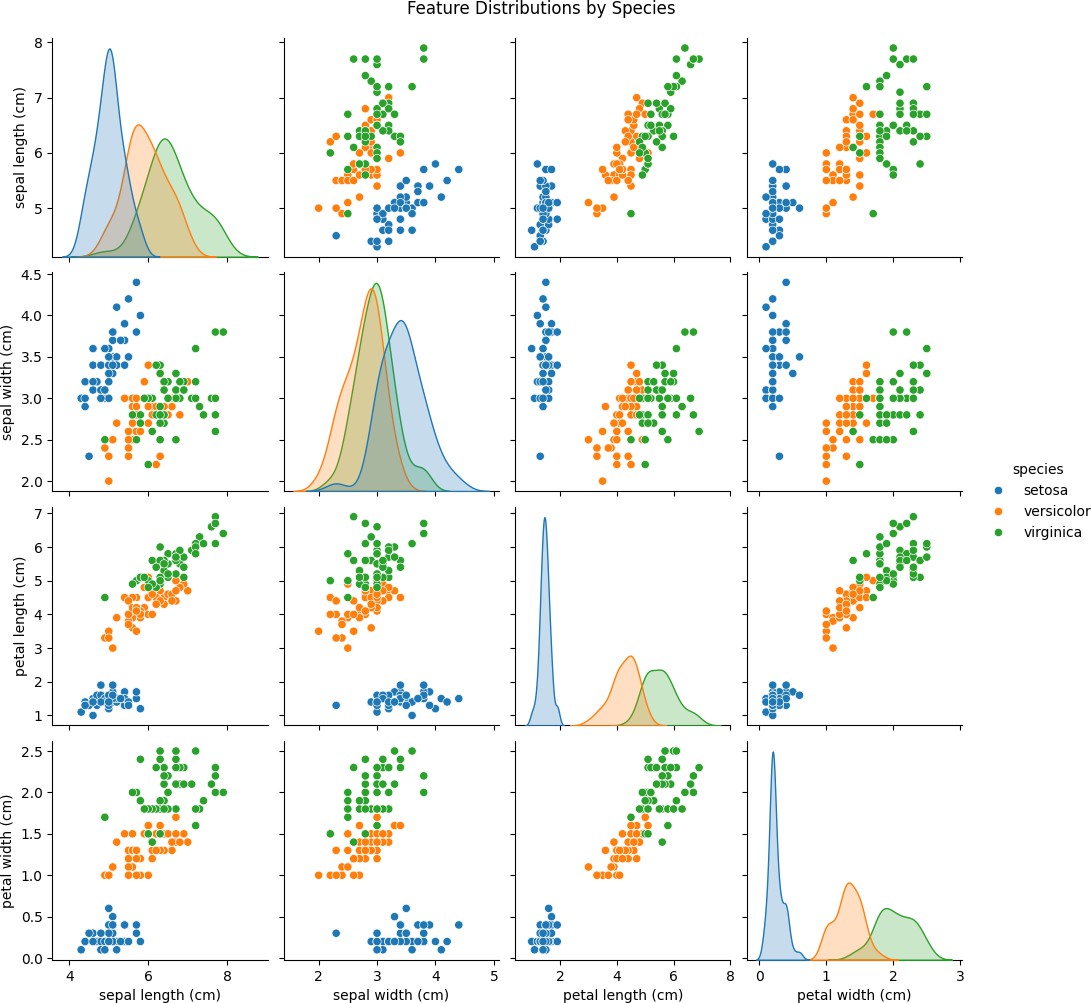
1. setosa
2. setosa
3. setosa
4. setosa
5. setosa

In [3]:

*# Plot feature distributions*

sns**.**pairplot(df, hue**=**'species')

plt**.**suptitle('Feature Distributions by Species', y**=**1.02) plt**.**show()



In [4]:

*# 2. Train a Machine Learning Classifier*

X **=** df[iris**.**feature\_names] y **=** df['species']

*# Standardize the features*

scaler **=** StandardScaler()

X\_scaled **=** scaler**.**fit\_transform(X)

*# Split into train/test*

X\_train, X\_test, y\_train, y\_test **=** train\_test\_split(X\_scaled, y, test\_size**=**0.3,

In [5]:

*# Train Random Forest Classifier*

clf **=** RandomForestClassifier(n\_estimators**=**100, random\_state**=**42) clf**.**fit(X\_train, y\_train)

y\_pred **=** clf**.**predict(X\_test)

In [6]:

*# 3. Evaluate the classifier*

print("Accuracy Score:", accuracy\_score(y\_test, y\_pred)) print("\nClassification Report:")

print(classification\_report(y\_test, y\_pred)) print("\nConfusion Matrix:")

sns**.**heatmap(confusion\_matrix(y\_test, y\_pred), annot**=True**, fmt**=**'d', xticklabels**=**i plt**.**xlabel("Predicted")

plt**.**ylabel("Actual")

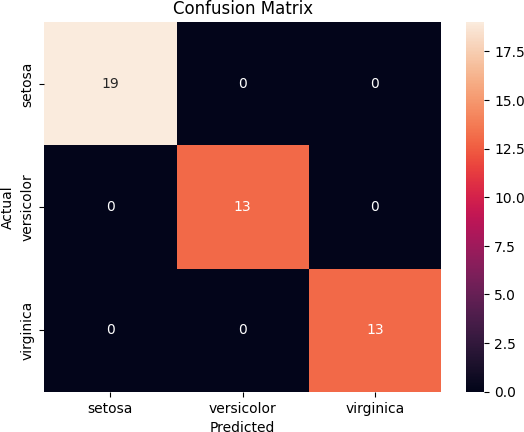
plt**.**title("Confusion Matrix") plt**.**show()

Accuracy Score: 1.0

Classification Report:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | precision | recall | f1-score | support |
| setosa | 1.00 | 1.00 | 1.00 | 19 |
| versicolor | 1.00 | 1.00 | 1.00 | 13 |
| virginica | 1.00 | 1.00 | 1.00 | 13 |
| accuracy |  |  | 1.00 | 45 |
| macro avg | 1.00 | 1.00 | 1.00 | 45 |
| weighted avg | 1.00 | 1.00 | 1.00 | 45 |

Confusion Matrix:



In [7]:

*# 4. Dimensionality Reduction and Visualization using PCA only*

pca **=** PCA(n\_components**=**2)

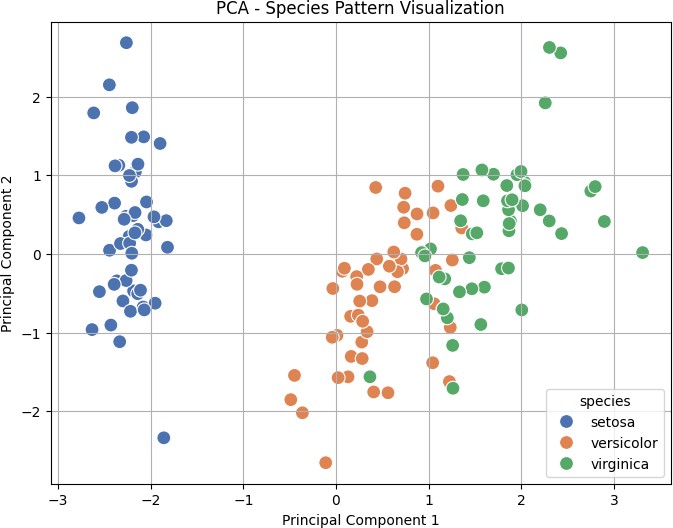
X\_pca **=** pca**.**fit\_transform(X\_scaled) plt**.**figure(figsize**=**(8,6))

sns**.**scatterplot(x**=**X\_pca[:,0], y**=**X\_pca[:,1], hue**=**y, palette**=**"deep", s**=**100) plt**.**title("PCA - Species Pattern Visualization")

plt**.**xlabel("Principal Component 1")

plt**.**ylabel("Principal Component 2") plt**.**grid(**True**)

plt**.**show()



In [ ]: