Load the Dataset

In [1]:
```
# Using Scikit-learn's Built-in Dataset
from sklearn.datasets import load_iris
import pandas as pd

# Load the iris dataset
iris = load_iris()

# Convert to a DataFrame
df = pd.DataFrame(data=iris.data, columns=iris.feature_names)

# Add the target column (species)
df['species'] = iris.target

# Map target numbers to species names
df['species'] = df['species'].map({0: 'setosa', 1: 'versicolor', 2: 'virginica'})

# Display first few rows
df.head()
```

Out[1]:

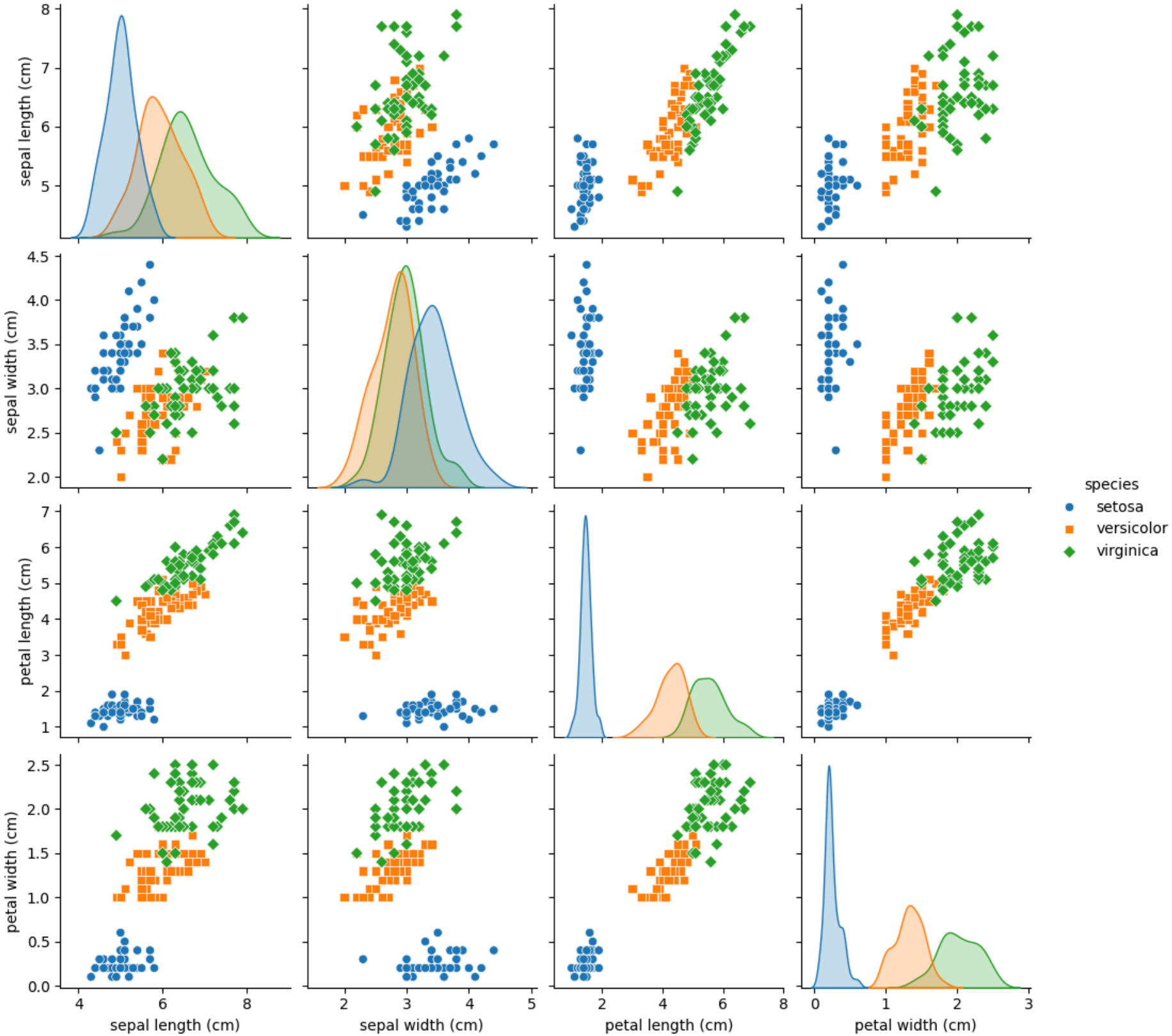
	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

Exploratory Data Analysis

In [2]:

```
import matplotlib.pyplot as plt
import seaborn as sns

# Check dataset info
print(df.info())

# Summary statistics
print(df.describe())

# Visualize pairwise relationships
sns.pairplot(df, hue='species', markers=["o", "s", "D"])
plt.show()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   sepal length (cm)  150 non-null    float64
 1   sepal width (cm)   150 non-null    float64
 2   petal length (cm)  150 non-null    float64
 3   petal width (cm)   150 non-null    float64
 4   species            150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
None
       sepal length (cm)  sepal width (cm)  petal length (cm)  \
count         150.000000        150.000000         150.000000   
mean            5.843333          3.057333           3.758000   
std             0.828066          0.435866           1.765298   
min             4.300000          2.000000           1.000000   
25%             5.100000          2.800000           1.600000   
50%             5.800000          3.000000           4.350000   
75%             6.400000          3.300000           5.100000   
max             7.900000          4.400000           6.900000   

       petal width (cm)  
count        150.000000  
mean           1.199333  
std            0.762238  
min            0.100000  
25%            0.300000  
50%            1.300000  
75%            1.800000  
max            2.500000
```


sepal length (cm)
sepal width (cm)
petal length (cm)
petal width (cm)
species
setosa
versicolor
virginica

Prepare Data for Training

In [3]:
```
# Separate Features & Labels
X = df.drop(columns=['species'])  # Features (measurements)
y = df['species']  # Labels (flower species)
```

In [4]:
```
# Split Data into Training & Testing Sets
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Train a Machine Learning Model

In [5]:
```
from sklearn.linear_model import LogisticRegression

# Train the model
model = LogisticRegression(max_iter=200)
model.fit(X_train, y_train)
```

Out[5]:

LogisticRegression

```
LogisticRegression(max_iter=200)
```

Evaluate Model Performance

In [6]:
```
from sklearn.metrics import accuracy_score, classification_report

# Predict on test data
y_pred = model.predict(X_test)

# Model accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f"Model Accuracy: {accuracy:.2f}")
```

```
# Classification report
print(classification_report(y_test, y_pred))
```

```
Model Accuracy: 1.00
              precision    recall  f1-score   support

      setosa       1.00      1.00      1.00        10
  versicolor       1.00      1.00      1.00         9
   virginica       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
```

Make Predictions on New Data

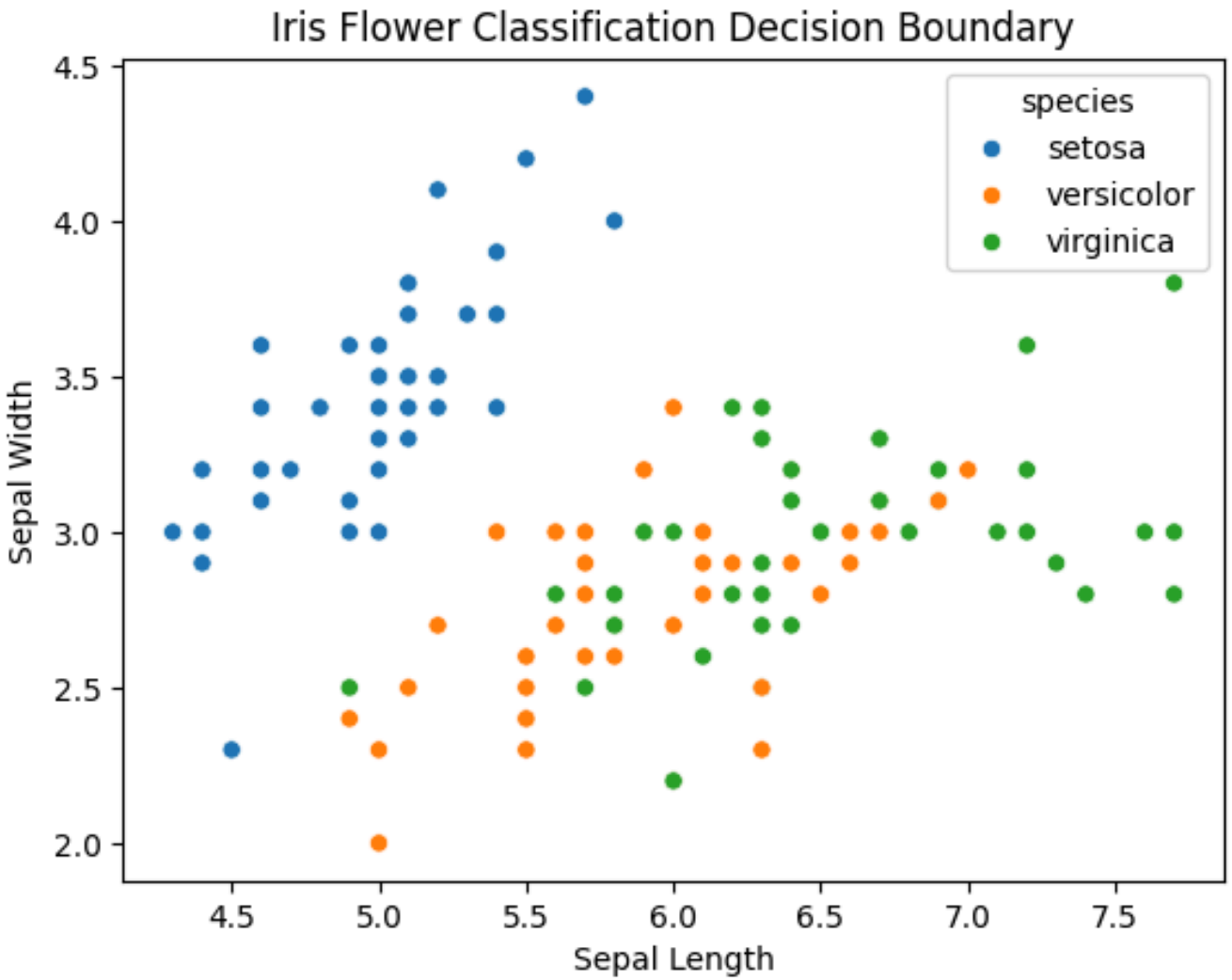
In [8]:
```
# Test the model with new iris measurements.

import pandas as pd

# Create a DataFrame with the same feature names as the training data
new_sample = pd.DataFrame([[5.1, 3.5, 1.4, 0.2]], columns=X_train.columns)

# Predict the species
predicted_species = model.predict(new_sample)
print("Predicted Species:", predicted_species[0])
```

```
Predicted Species: setosa
```

Visualize Results

In [9]:
```
from sklearn.svm import SVC

# Train SVM model for visualization
svm_model = SVC(kernel='linear')
svm_model.fit(X_train[['sepal length (cm)', 'sepal width (cm)']], y_train)
```

```
# Create a scatter plot
sns.scatterplot(x=X_train['sepal length (cm)'], y=X_train['sepal width (cm)'], hue=y_train)
plt.xlabel("Sepal Length")
plt.ylabel("Sepal Width")
plt.title("Iris Flower Classification Decision Boundary")
plt.show()
```



In [ ]: