```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
# from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
```

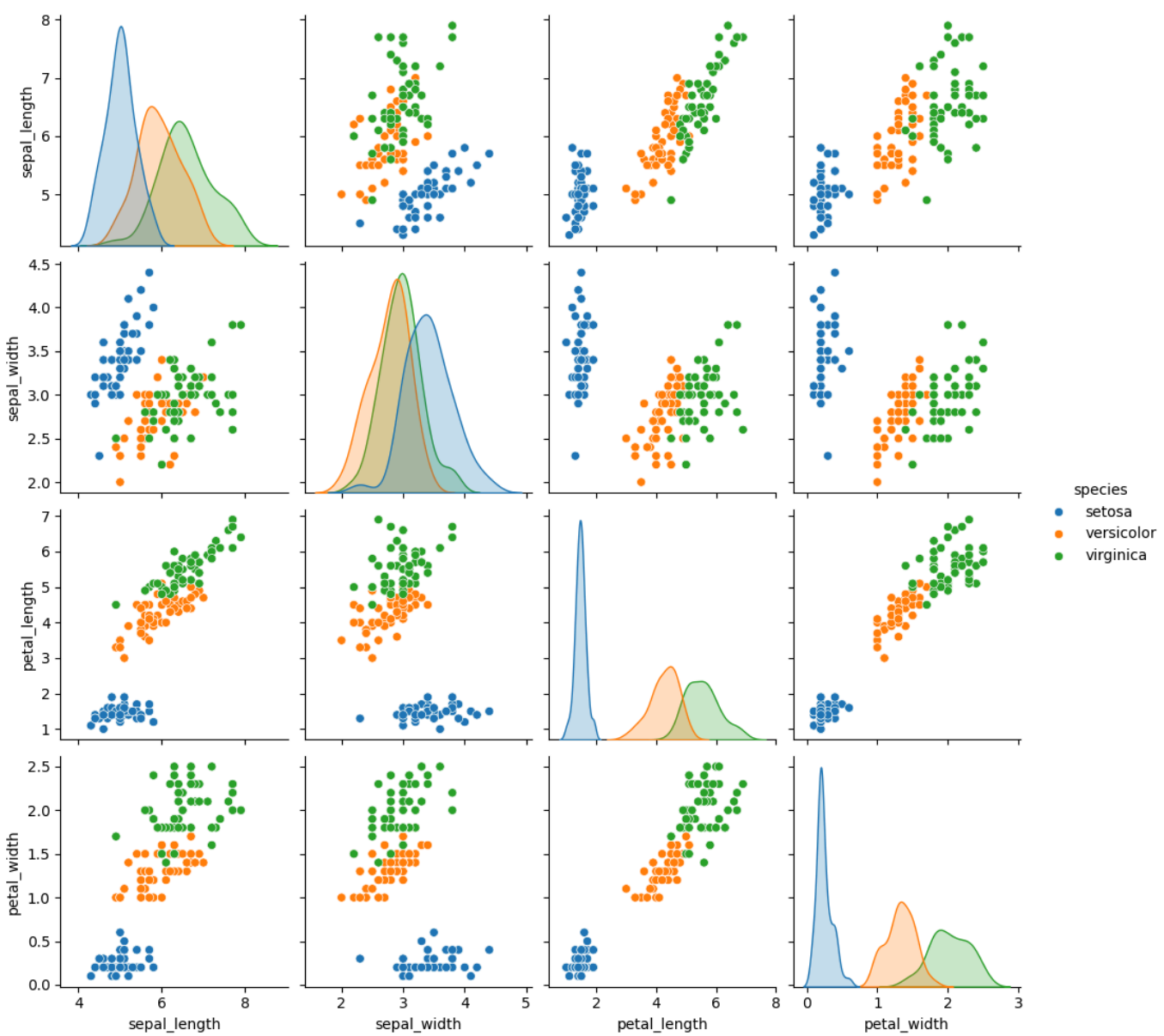
Start coding or generate with AI.

```
import pandas as pd
iris = pd.read_csv('/content/iris.csv')


# Display first few rows
print(iris.head())
```

```
   sepal_length  sepal_width  petal_length  petal_width 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 (EDA)
import seaborn as sns # Import the seaborn library and assign it to the alias 'sns'
import matplotlib.pyplot as plt # Import the matplotlib library for plotting
sns.pairplot(iris, hue='species')
plt.show()
```




```
X = iris.drop(columns=['species'])
y = iris['species']


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


from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)



from sklearn.linear_model import LogisticRegression

model = LogisticRegression()
```

```
model.fit(X_train, y_train)
```

```
LogisticRegression
LogisticRegression()
```

```
y_pred = model.predict(X_test)


from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

  accuracy = accuracy_score(y_test, y_pred)
  print(f'Accuracy: {accuracy:.2f}')
  print('Classification Report:\n', classification_report(y_test, y_pred))
  print('Confusion Matrix:\n', confusion_matrix(y_test, y_pred))
```

```
Accuracy: 1.00
Classification Report:
                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

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