```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

```
iris_data = sns.load_dataset('iris')
```

```
print(iris_data.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
```

```
print(iris_data.describe())
```

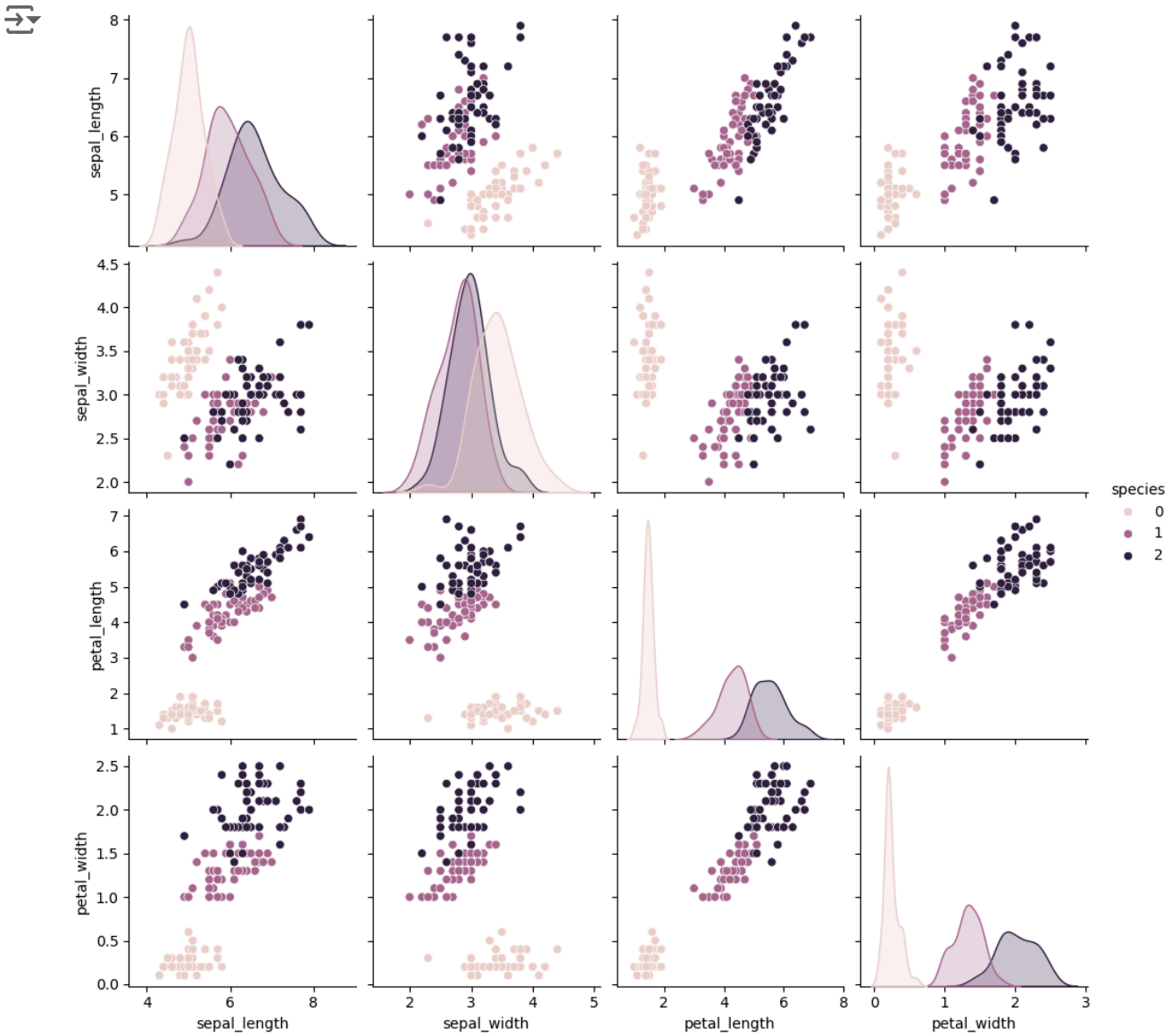
```
       sepal_length  sepal_width  petal_length  petal_width
count    150.000000   150.000000    150.000000   150.000000
mean       5.843333     3.057333      3.758000     1.199333
std        0.828066     0.435866      1.765298     0.762238
min        4.300000     2.000000      1.000000     0.100000
25%        5.100000     2.800000      1.600000     0.300000
50%        5.800000     3.000000      4.350000     1.300000
75%        6.400000     3.300000      5.100000     1.800000
max        7.900000     4.400000      6.900000     2.500000
```

```
print(iris_data.isnull().sum())
```

```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

```
iris_data['species'] = iris_data['species'].astype('category').cat.codes
```

```
sns.pairplot(iris_data, hue='species')
plt.show()
```

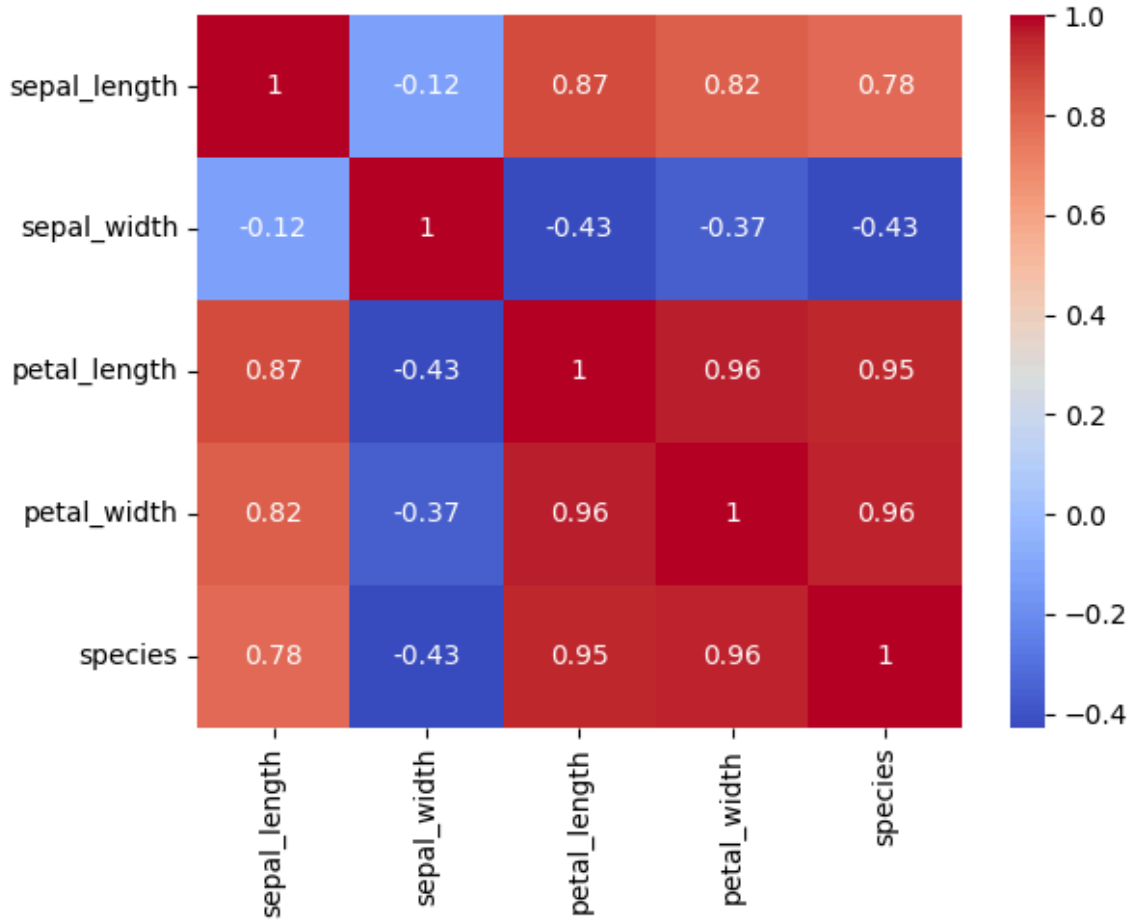



```
plt.figure(figsize=(10, 6))
```

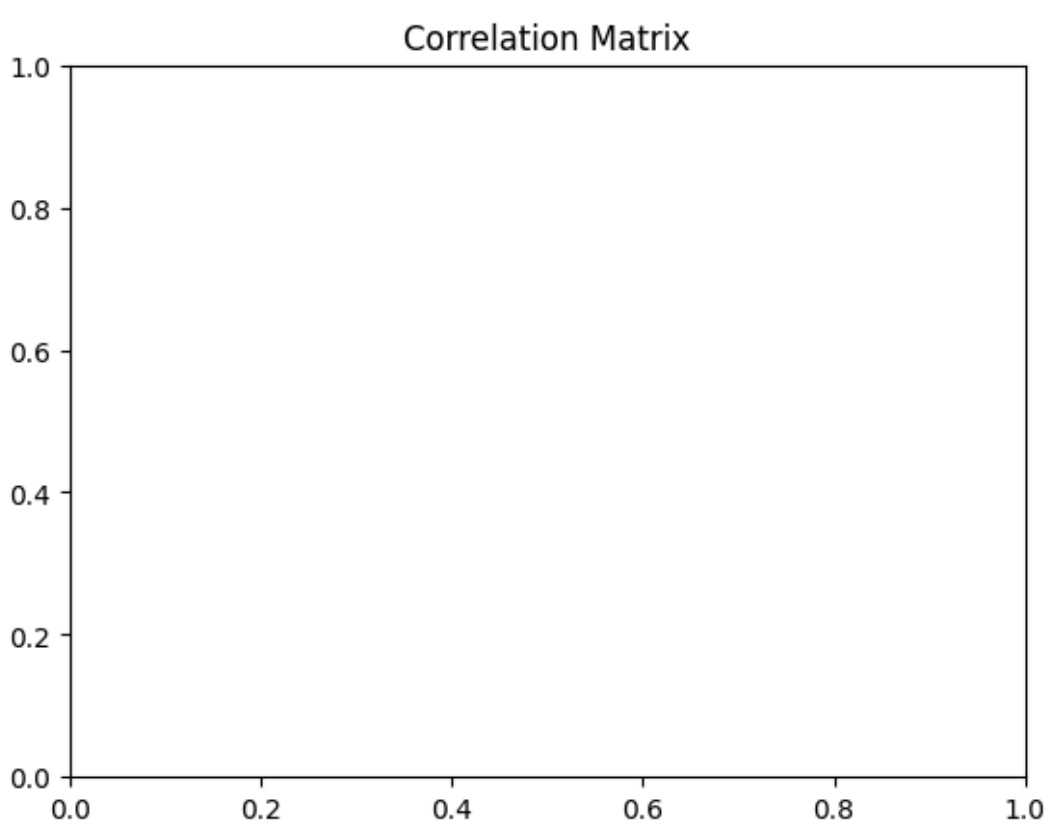
```
<Figure size 1000x600 with 0 Axes>
<Figure size 1000x600 with 0 Axes>
```

```
sns.heatmap(iris_data.corr(), annot=True, cmap='coolwarm')
```

```
<Axes: >
```




```
plt.title('Correlation Matrix')
plt.show()
```



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

y = iris_data['species']

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

dt_model = DecisionTreeClassifier()

dt_model.fit(X_train, y_train)
```

```
▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

```
y_pred = dt_model.predict(X_test)


accuracy = accuracy_score(y_test, y_pred)


conf_matrix = confusion_matrix(y_test, y_pred)


class_report = classification_report(y_test, y_pred)


print(f'Accuracy: {accuracy}')
```

```
Accuracy: 1.0
```

```
print('Confusion Matrix:')
```

```
Confusion Matrix:
```

```
print(conf_matrix)
```

```
[[10  0  0]
 [ 0  9  0]
 [ 0  0 11]]
```

```
print('Classification Report:')
```

```
Classification Report:
```

```
print(class_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