```
# @title Import Required Libraries
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix
from sklearn.ensemble import RandomForestClassifier


# @title Load the Dataset
df = pd.read_csv("IRIS.csv")
df.head()
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 150,\n  \"fields\": [\
n    {\n      \"column\": \"sepal_length\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0.8280661279778629,\n
\"min\": 4.3,\n        \"max\": 7.9,\n        \"num_unique_values\":
35,\n        \"samples\": [\n          6.2,\n          4.5,\n
5.6\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"sepal_width\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.4335943113621737,\n        \"min\":
2.0,\n        \"max\": 4.4,\n        \"num_unique_values\": 23,\n
\"samples\": [\n          2.3,\n          4.0,\n          3.5\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"petal_length\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1.7644204199522617,\n        \"min\": 1.0,\n        \"max\": 6.9,\n
\"num_unique_values\": 43,\n        \"samples\": [\n          6.7,\n
3.8,\n          3.7\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"petal_width\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.7631607417008414,\n        \"min\":
0.1,\n        \"max\": 2.5,\n        \"num_unique_values\": 22,\n
\"samples\": [\n          0.2,\n          1.2,\n          1.3\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"species\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 3,\n        \"samples\": [\n          \"Iris-
setosa\",\n          \"Iris-versicolor\",\n          \"Iris-
virginica\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"df"}
```

```
# @title Understand the Dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

```
df.isnull().sum()
```

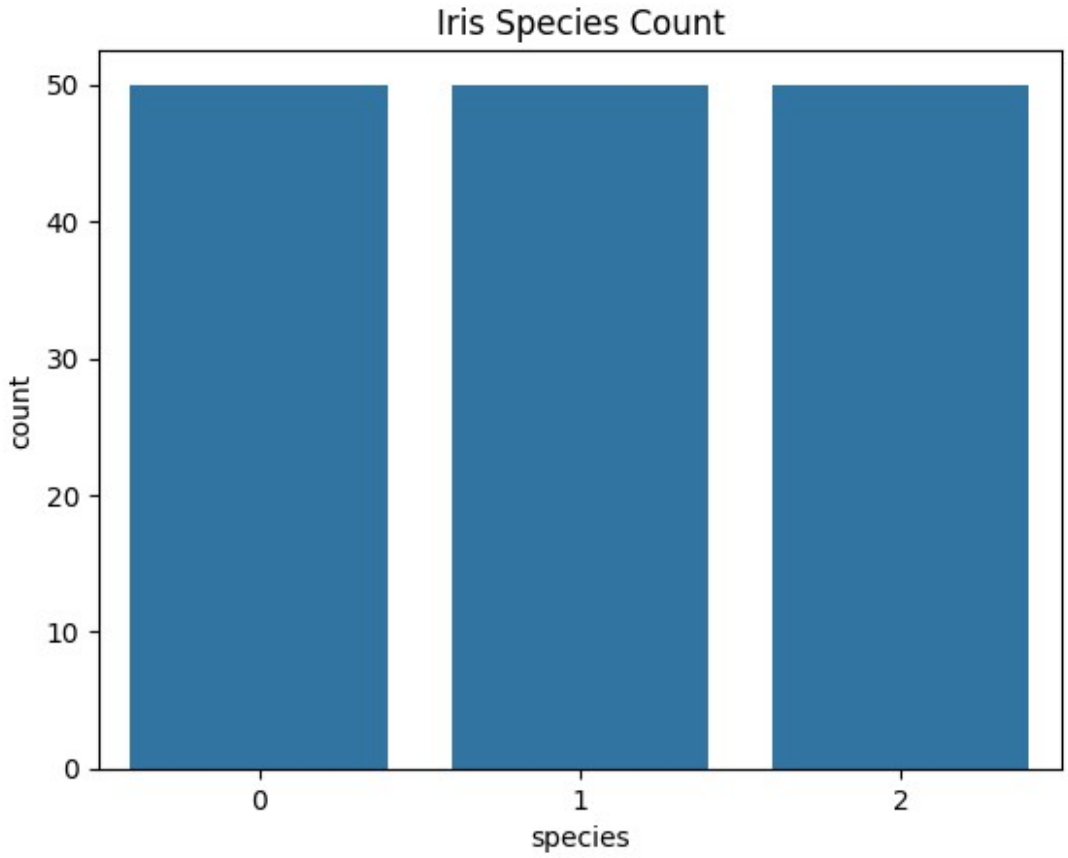
```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

```
df.describe()
```

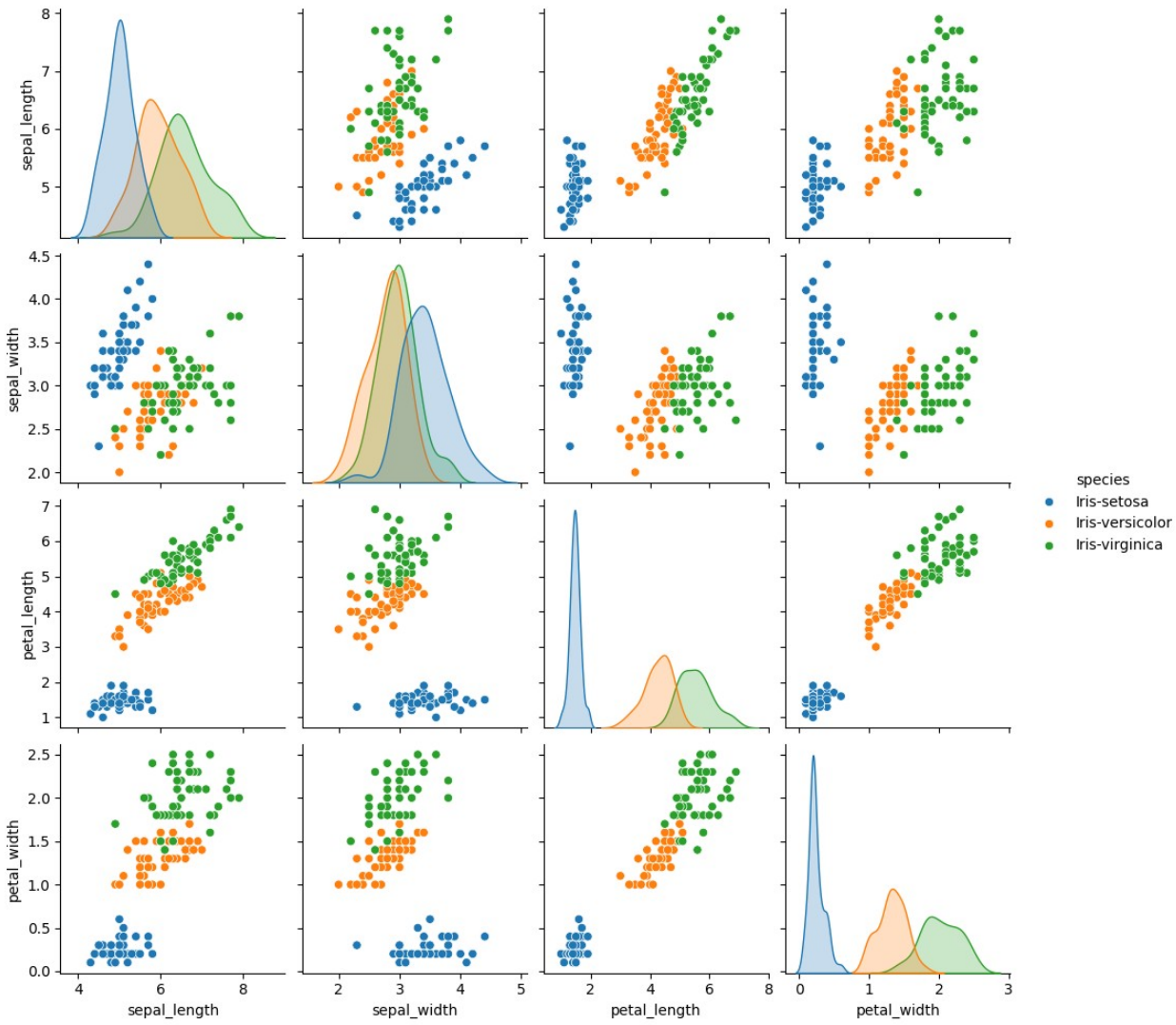

```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 8,\n  \"fields\": [\n
{\n      \"column\": \"sepal_length\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 51.24711349471842,\n
\"min\": 0.8280661279778629,\n        \"max\": 150.0,\n
\"num_unique_values\": 8,\n        \"samples\": [\n
5.843333333333334,\n          5.8,\n          150.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"sepal_width\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
52.08647211421483,\n        \"min\": 0.4335943113621737,\n
\"max\": 150.0,\n        \"num_unique_values\": 8,\n
\"samples\": [\n          3.0540000000000003,\n          3.0,\n
150.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"petal_length\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 51.835227940958106,\n        \"min\":
1.0,\n        \"max\": 150.0,\n        \"num_unique_values\": 8,\n
\"samples\": [\n          3.758666666666666,\n          4.35,\n
150.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"petal_width\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 52.63663424340991,\n        \"min\":
0.1,\n        \"max\": 150.0,\n        \"num_unique_values\": 8,\n
```

\"samples\": [\n   1.1986666666666668,\n   1.3,\n
150.0\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"species\",\n   \"properties\": {\n   \"dtype\": \"number\",\
n   \"std\": 52.69404575122032,\n   \"min\": 0.0,\n
\"max\": 150.0,\n   \"num_unique_values\": 5,\n
\"samples\": [\n   1.0,\n   2.0,\n
0.8192319205190405\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe"}

```
# @title Exploratory Data Analysis (EDA)
sns.countplot(x='species', data=df)
plt.title("Iris Species Count")
plt.show()
```




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




```
# @title Encode Target Variable
le = LabelEncoder()
df['species'] = le.fit_transform(df['species'])

# @title Feature Selection
X = df.drop(columns=['species'])
y = df['species']

# @title Train-Test Split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

# @title Build the Model
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

RandomForestClassifier(random_state=42)
```

```
# @title Make Predictions
y_pred = model.predict(X_test)

# @title Model Evaluation
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 1.0
```

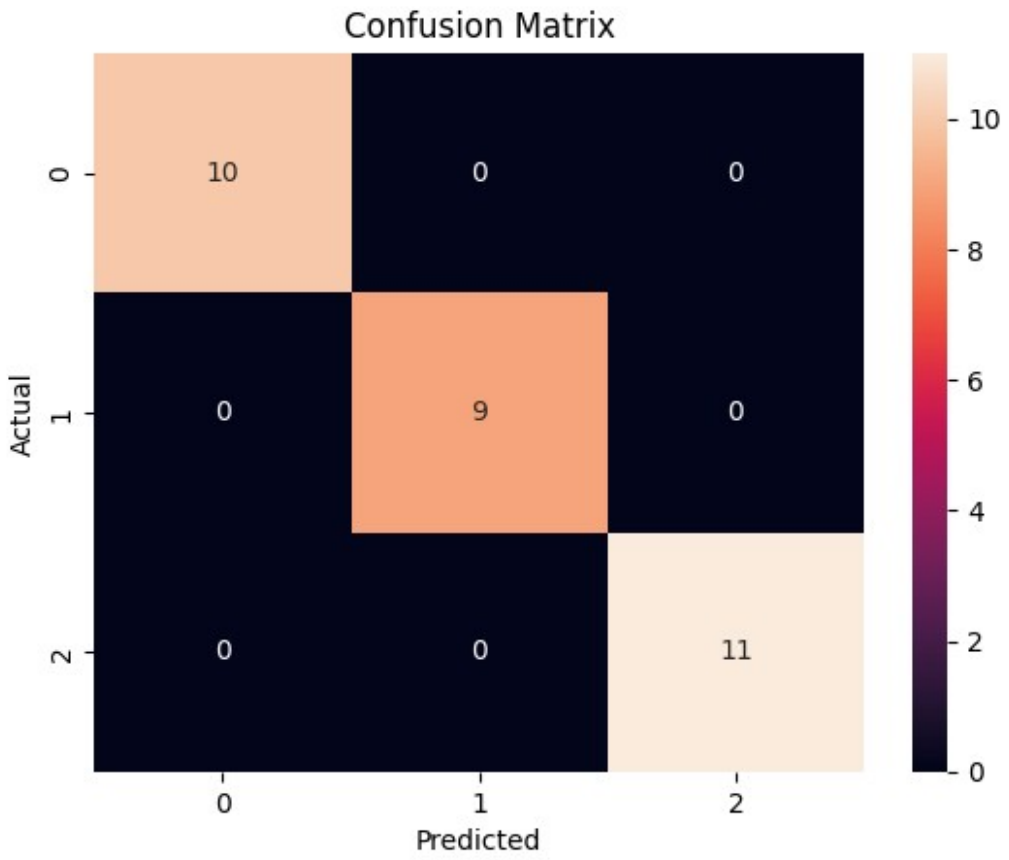
```
print(classification_report(y_test, y_pred))
```

```
              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
```

```
# @title Model Evaluation
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt='d')
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```




```
# @title Feature Importance
feature_importance = pd.DataFrame({
    'Feature': X.columns,
    'Importance': model.feature_importances_
}).sort_values(by='Importance', ascending=False)

feature_importance
```


```
{"summary":"{\n  \"name\": \"feature_importance\",\n  \"rows\": 4,\n
\"fields\": [\n    {\n      \"column\": \"Feature\",\n
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 4,\n        \"samples\": [\n
\"petal_width\",\n          \"sepal_width\",\n
\"petal_length\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Importance\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.2112530882478962,\n        \"min\":
0.030386812473242528,\n        \"max\": 0.43999397414456937,\n
\"num_unique_values\": 4,\n        \"samples\": [\n
0.4215215887397244,\n          0.030386812473242528,\n
0.43999397414456937\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"feature_importance"}
```

```
# @title Conclusion
print("Iris Flower Classification model built successfully.")
print("Petal length and petal width are the most important features.")
```

```
Iris Flower Classification model built successfully.
Petal length and petal width are the most important features.
```