Import Libraries

In [ ]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
from sklearn import metrics
```

Load Dataset

In [ ]:
```
iris = load_iris()
iris_df = sns.load_dataset("iris")
```

Basic EDA

In [ ]:
```
iris_df.head()
```

Out[ ]:

	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

In [ ]:
```
iris_df.isnull().sum()
```

Out[ ]:
```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

In [ ]:
```
iris_df.describe()
```

Out[ ]:

	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

In [ ]:
```
iris_df['species'].unique()
```

Out[ ]:
```
array(['setosa', 'versicolor', 'virginica'], dtype=object)
```

In [ ]:
```
print("Setosa : ")
iris_df[iris_df['species']  == 'setosa'].describe()
```

```
Setosa :
```

Out[ ]:

	sepal_length	sepal_width	petal_length	petal_width
count	50.00000	50.000000	50.000000	50.000000
mean	5.00600	3.428000	1.462000	0.246000
std	0.35249	0.379064	0.173664	0.105386
min	4.30000	2.300000	1.000000	0.100000
25%	4.80000	3.200000	1.400000	0.200000
50%	5.00000	3.400000	1.500000	0.200000
75%	5.20000	3.675000	1.575000	0.300000
max	5.80000	4.400000	1.900000	0.600000

In [ ]:
```
print("Versicolor : ")
iris_df[iris_df['species']  == 'versicolor'].describe()
```

```
Versicolor :
```

Out[ ]:

	sepal_length	sepal_width	petal_length	petal_width
count	50.000000	50.000000	50.000000	50.000000
mean	5.936000	2.770000	4.260000	1.326000
std	0.516171	0.313798	0.469911	0.197753
min	4.900000	2.000000	3.000000	1.000000
25%	5.600000	2.525000	4.000000	1.200000
50%	5.900000	2.800000	4.350000	1.300000
75%	6.300000	3.000000	4.600000	1.500000
max	7.000000	3.400000	5.100000	1.800000

In [ ]:
```
print("Virginica : ")
iris_df[iris_df['species']  == 'virginica'].describe()
```

```
Virginica :
```

Out[ ]:

	sepal_length	sepal_width	petal_length	petal_width
count	50.00000	50.000000	50.000000	50.00000
mean	6.58800	2.974000	5.552000	2.02600
std	0.63588	0.322497	0.551895	0.27465
min	4.90000	2.200000	4.500000	1.40000
25%	6.22500	2.800000	5.100000	1.80000
50%	6.50000	3.000000	5.550000	2.00000
75%	6.90000	3.175000	5.875000	2.30000
max	7.90000	3.800000	6.900000	2.50000

Target Variable is Type of Species

In [ ]:
```
X = iris_df.drop(columns='species')
y = iris_df['species']
```

Partition Data for training and testing the model and standardize the data

In [ ]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=9)
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

KNN Model

In [ ]:
```
knn_model = KNeighborsClassifier(n_neighbors=5)
knn_model.fit(X_train_scaled, y_train)
```

Out[ ]:
```
KNeighborsClassifier
KNeighborsClassifier()
```

Predict the species for the test data

In [ ]:
```
y_pred = knn_model.predict(X_test_scaled)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 1.0
```

In [ ]:
```
knn_model.score(X_test_scaled, y_test)
```

Out[ ]:
```
1.0
```

In [ ]:
```
comparision = pd.DataFrame({'Predicted':y_pred, 'Actual':y_test})
comparision
```

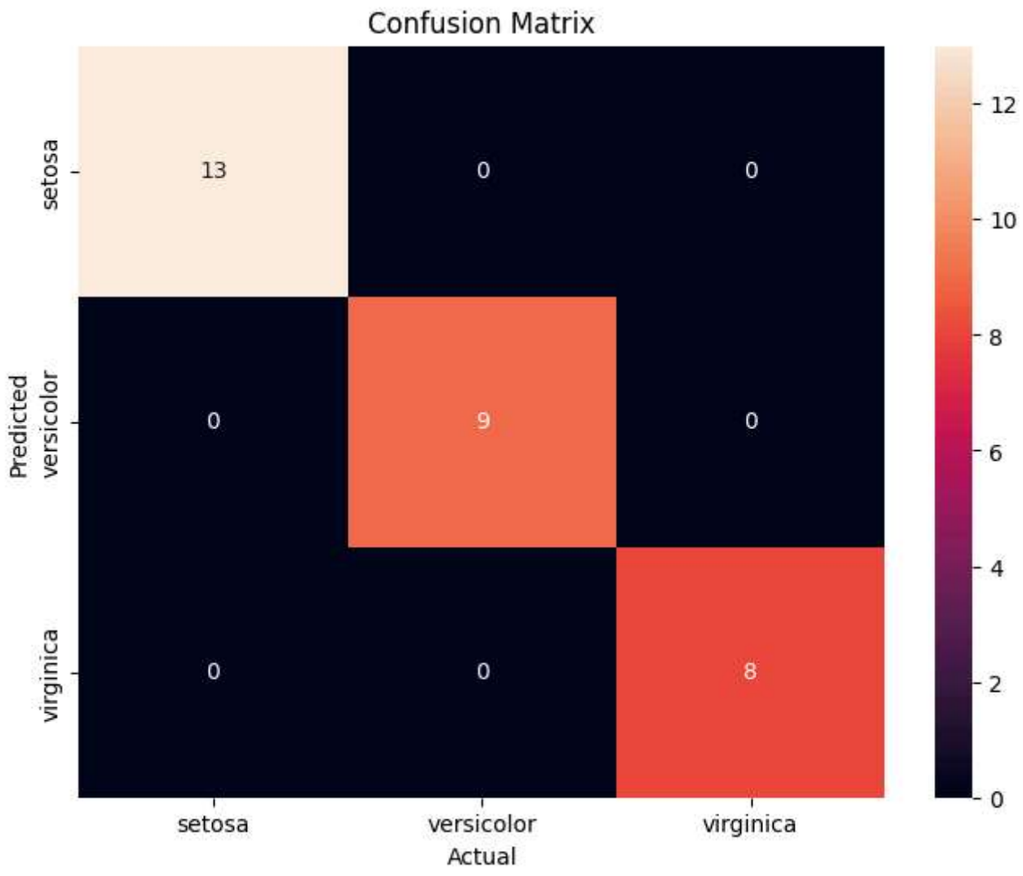
Out[ ]:

	Predicted	Actual
135	virginica	virginica
90	versicolor	versicolor
145	virginica	virginica
147	virginica	virginica
60	versicolor	versicolor
37	setosa	setosa
26	setosa	setosa
3	setosa	setosa
75	versicolor	versicolor
9	setosa	setosa
25	setosa	setosa
73	versicolor	versicolor
98	versicolor	versicolor
94	versicolor	versicolor
39	setosa	setosa
81	versicolor	versicolor
31	setosa	setosa
55	versicolor	versicolor
101	virginica	virginica
47	setosa	setosa
28	setosa	setosa
27	setosa	setosa
111	virginica	virginica
20	setosa	setosa
118	virginica	virginica
89	versicolor	versicolor
43	setosa	setosa
144	virginica	virginica
8	setosa	setosa
137	virginica	virginica

In [ ]:
```
confusion_matrix = metrics.confusion_matrix(y_test, y_pred)
confusion_matrix
```

Out[ ]:
```
array([[13,  0,  0],
       [ 0,  9,  0],
       [ 0,  0,  8]], dtype=int64)
```

In [ ]:
```
species_names = iris_df['species'].unique()
plt.figure(figsize=(8, 6))
sns.heatmap(confusion_matrix, annot=True, xticklabels=species_names, yticklabels=species_names)
plt.xlabel('Actual')
plt.ylabel('Predicted')
plt.title('Confusion Matrix')
plt.show()
```



In [ ]: