Support Vector Machine using Scikit-learn

In [104]:

```python
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report,confusion_matrix

data=pd.read_csv('iris')
data

```

Out[104]:

	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
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

data cleaning

In [86]:

```python
data.shape
```

Out[86]:

```
(150, 5)
```

In [87]:

```python
data.isnull().sum()
```

Out[87]:

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

In [88]:

```python
data.duplicated().sum()
```

Out[88]:

```
1
```

In [89]:

```
dataset= data.drop_duplicates()
dataset
```

Out[89]:

	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
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

149 rows × 5 columns

In [90]:

```
dataset.shape
```

Out[90]:

```
(149, 5)
```

In [91]:

```
dataset.count()
```

Out[91]:

```
sepal_length    149
sepal_width     149
petal_length    149
petal_width     149
species         149
dtype: int64
```

In [92]:

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 149 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   sepal_length  149 non-null    float64
 1   sepal_width   149 non-null    float64
 2   petal_length  149 non-null    float64
 3   petal_width   149 non-null    float64
 4   species       149 non-null    object 
dtypes: float64(4), object(1)
memory usage: 7.0+ KB
```

In [93]:

```
dataset.nunique()
```

Out[93]:

```
sepal_length    35
sepal_width     23
petal_length    43
petal_width     22
species          3
dtype: int64
```

In [94]:

```
dataset.describe()
```

Out[94]:

	sepal_length	sepal_width	petal_length	petal_width
count	149.000000	149.000000	149.000000	149.000000
mean	5.843624	3.059732	3.748993	1.194631
std	0.830851	0.436342	1.767791	0.762622
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.300000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

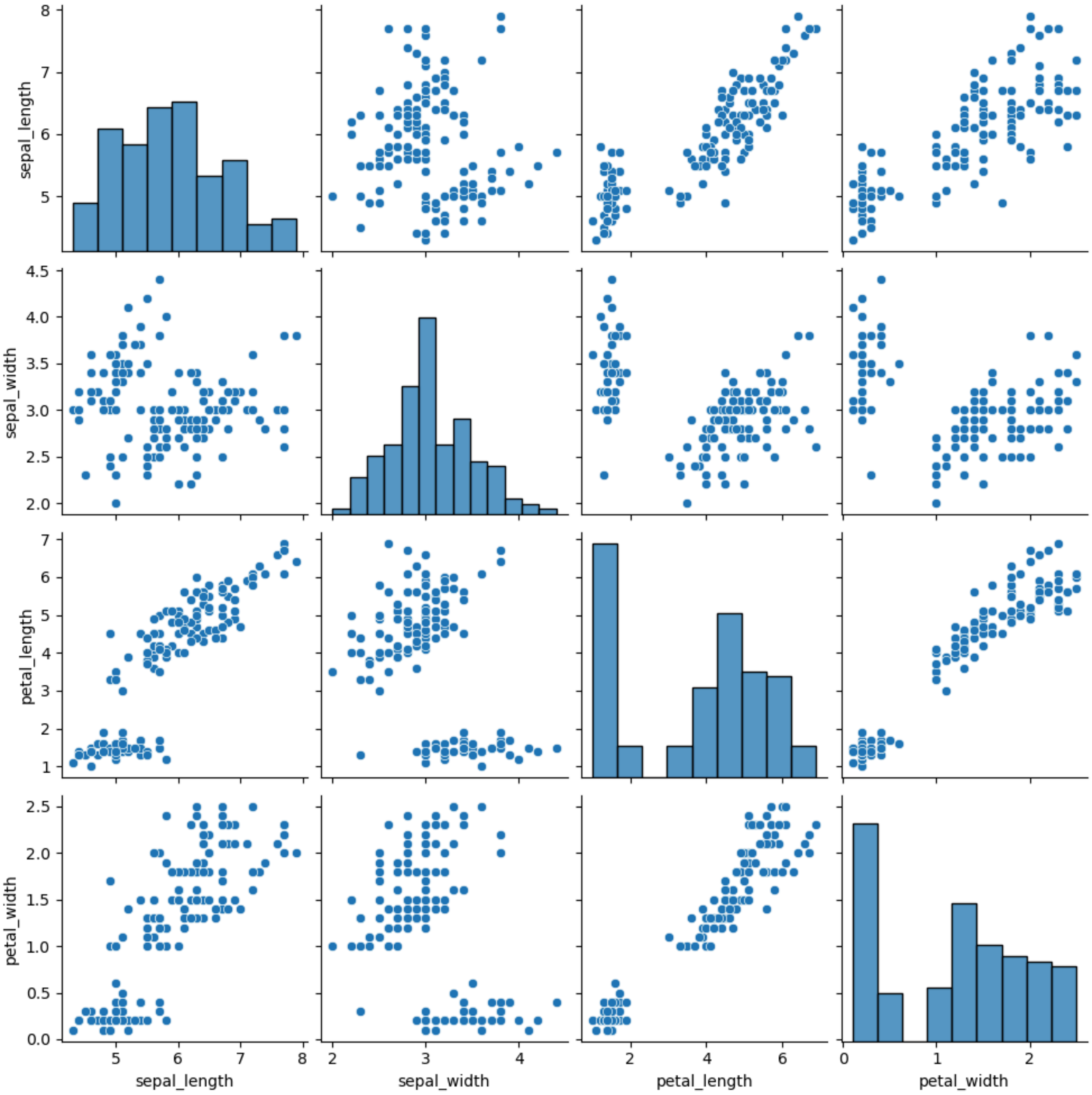
In [95]:

```
import seaborn as sns
sns.pairplot(dataset)
```

```
C:\Users\Ahmed Islam\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figur
e layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

Out[95]:

```
<seaborn.axisgrid.PairGrid at 0x1b95fad0090>
```



In [96]:

```python
X = dataset.iloc[:,[0,1,2,3]]
X
```

Out[96]:

	sepal_length	sepal_width	petal_length	petal_width
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...	...	...	...	...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

149 rows × 4 columns

In [97]:

```python
y= dataset.iloc[:,[4]]
y
```

Out[97]:

	species
0	setosa
1	setosa
2	setosa
3	setosa
4	setosa
...	...
145	virginica
146	virginica
147	virginica
148	virginica
149	virginica

149 rows × 1 columns

Train_Test_split

In [98]:

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

In [99]:

```python
print(X_train.shape)
print(X_test.shape)
```

```
(119, 4)
(30, 4)
```

In [100]:

```
print(y_train.shape)
print(y_test.shape)
```

```
(119, 1)
(30, 1)
```

Creating Model

In [101]:

```
model = SVC(kernel='linear')

model.fit(X_train, y_train)
y_pred = model.predict(X_test)


print('The Shape of Y predicted is:', y_pred.shape)
print()
print("The predicted values is:")
print(y_pred)
```

```
The Shape of Y predicted is: (30,)

The predicted values is:
['versicolor' 'setosa' 'virginica' 'versicolor' 'versicolor' 'setosa'
 'versicolor' 'virginica' 'versicolor' 'versicolor' 'virginica' 'setosa'
 'setosa' 'setosa' 'setosa' 'versicolor' 'virginica' 'versicolor'
 'versicolor' 'virginica' 'setosa' 'virginica' 'setosa' 'virginica'
 'virginica' 'virginica' 'virginica' 'virginica' 'setosa' 'setosa']
```

```
C:\Users\Ahmed Islam\anaconda3\Lib\site-packages\sklearn\utils\validation.py:1339: DataConversio
nWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of
y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

Accuracy Calculating

In [102]:

```
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:" ,accuracy)
```

```
Accuracy: 1.0
```

confusion Matrix

In [103]:

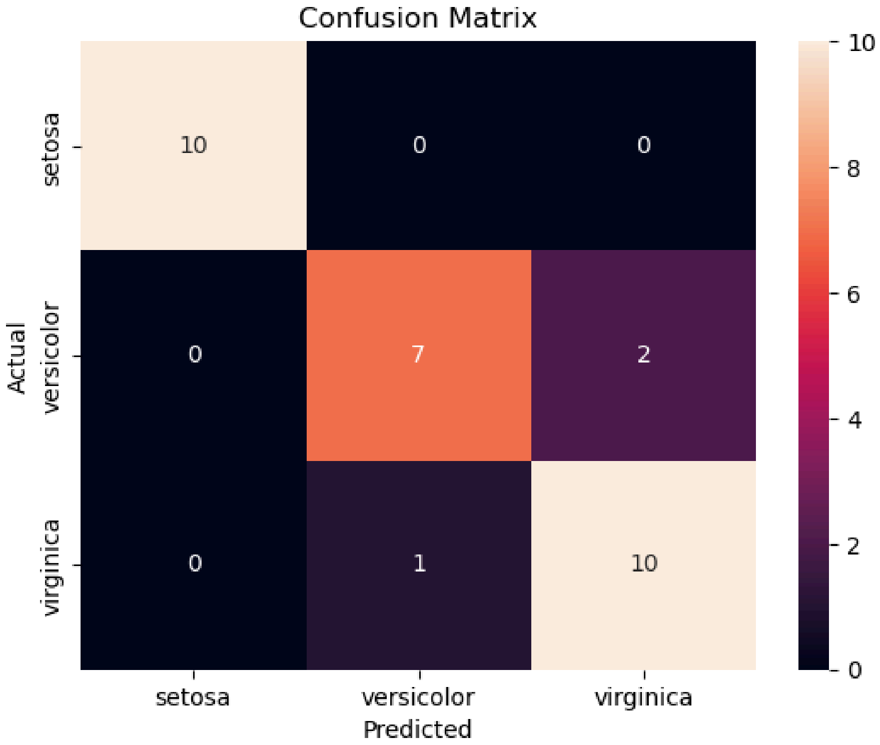
```
conf_matrix = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:")
print(conf_matrix)
```

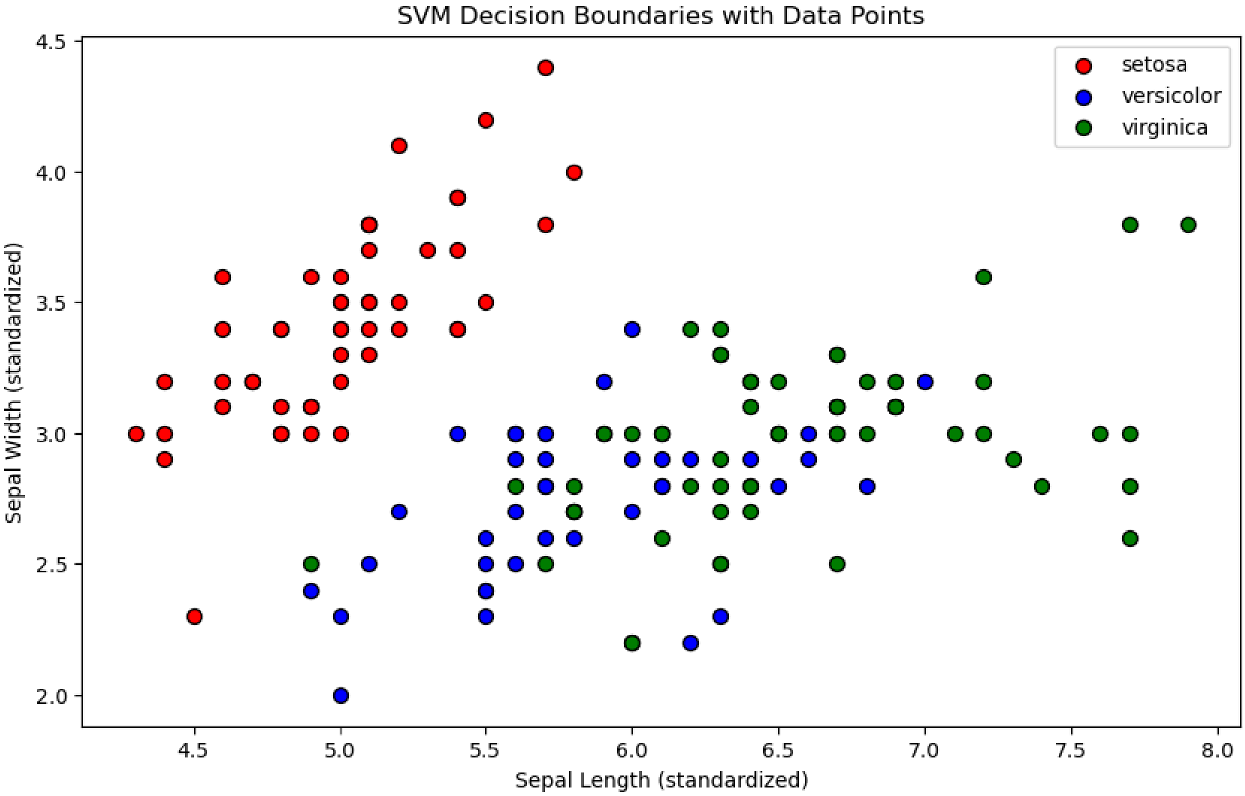
```
Confusion Matrix:
[[10  0  0]
 [ 0  9  0]
 [ 0  0 11]]
```

In [80]:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.svm import SVC

df = pd.read_csv('iris')

print("First few rows of the dataset:")
print(df.head())

print("\nMissing values in the dataset:")
print(df.isnull().sum())

X = df[['sepal_length', 'sepal_width']]
y = df['species']

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

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

model = SVC(kernel='linear')
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
print(f"\nAccuracy of the SVM model: {accuracy}")

confusion_matrix = pd.crosstab(y_test, y_pred, rownames=['Actual'], colnames=['Predicted'])
sns.heatmap(confusion_matrix, annot=True)
plt.title('Confusion Matrix')
plt.show()

plt.figure(figsize=(10, 6))
colors = {'setosa': 'red', 'versicolor': 'blue', 'virginica': 'green'}
for species in np.unique(y):
    subset = X[y == species]
    plt.scatter(subset.iloc[:, 0], subset.iloc[:, 1], color=colors[species], label=species, s

ax = plt.gca()
xlim = ax.get_xlim()
ylim = ax.get_ylim()

xx, yy = np.meshgrid(np.linspace(xlim[0], xlim[1], 200),
                     np.linspace(ylim[0], ylim[1], 200))
Z = model.decision_function(np.c_[xx.ravel(), yy.ravel()])

for i, color in zip(range(Z.shape[1]), ['red', 'blue', 'green']):
    Z_class = Z[:, i].reshape(xx.shape)
    ax.contour(xx, yy, Z_class, levels=[0], linewidths=2, colors=color)

plt.title('SVM Decision Boundaries with Data Points')
plt.xlabel('Sepal Length (standardized)')
plt.ylabel('Sepal Width (standardized)')
plt.legend()
plt.show()

```

```
First few rows of the dataset:
   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

Missing values in the dataset:
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64

Accuracy of the SVM model: 0.9
```




SVM Decision Boundaries with Data Points
setosa
versicolor
virginica
Sepal Width (standardized)
4.5
4.0
3.5
3.0
2.5
2.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
Sepal Length (standardized)