

SVM

68_Adnan Shaikh

```
In [3]: import pandas as pd
from sklearn.datasets import load_iris
iris = load_iris()
```

```
In [4]: iris.feature_names
```

```
Out[4]: ['sepal length (cm)',
'sepal width (cm)',
'petal length (cm)',
'petal width (cm)']
```

```
In [5]: dir(iris)
```

```
Out[5]: ['DESCR',
'data',
'data_module',
'feature_names',
'filename',
'frame',
'target',
'target_names']
```

```
In [6]: iris.target_names
```

```
Out[6]: array(['setosa', 'versicolor', 'virginica'], dtype='<U10')
```

```
In [7]: df = pd.DataFrame(iris.data, columns=iris.feature_names)
df.head()
```

```
Out[7]:
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
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

```
In [8]: df['target'] = iris.target
df.head()
```

```
Out[8]:
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

```
In [9]: df[df.target==1].head()
```

```
Out[9]:
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
50	7.0	3.2	4.7	1.4	1
51	6.4	3.2	4.5	1.5	1
52	6.9	3.1	4.9	1.5	1
53	5.5	2.3	4.0	1.3	1
54	6.5	2.8	4.6	1.5	1

```
In [10]: df[df.target==2].head()
```

```
Out[10]:
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
100	6.3	3.3	6.0	2.5	2
101	5.8	2.7	5.1	1.9	2
102	7.1	3.0	5.9	2.1	2
103	6.3	2.9	5.6	1.8	2
104	6.5	3.0	5.8	2.2	2

```
In [11]: df['flower_name'] =df.target.apply(lambda x: iris.target_names[x])
df.head()
```

```
Out[11]:
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target	flower_name
0	5.1	3.5	1.4	0.2	0	setosa
1	4.9	3.0	1.4	0.2	0	setosa
2	4.7	3.2	1.3	0.2	0	setosa
3	4.6	3.1	1.5	0.2	0	setosa
4	5.0	3.6	1.4	0.2	0	setosa

```
In [12]: df[45:55]
```

```
Out[12]:
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target	flower_name
45	4.8	3.0	1.4	0.3	0	setosa
46	5.1	3.8	1.6	0.2	0	setosa
47	4.6	3.2	1.4	0.2	0	setosa
48	5.3	3.7	1.5	0.2	0	setosa
49	5.0	3.3	1.4	0.2	0	setosa
50	7.0	3.2	4.7	1.4	1	versicolor
51	6.4	3.2	4.5	1.5	1	versicolor
52	6.9	3.1	4.9	1.5	1	versicolor
53	5.5	2.3	4.0	1.3	1	versicolor
54	6.5	2.8	4.6	1.5	1	versicolor

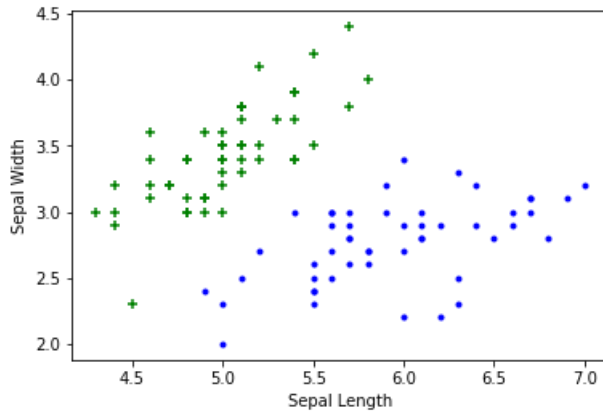
```
In [13]: df0 = df[:50]
df1 = df[50:100]
df2 = df[100:]
```

```
In [14]: import matplotlib.pyplot as plt
```

Sepal length vs Sepal Width (Setosa vs Versicolor)

```
In [15]: plt.xlabel('Sepal Length')
plt.ylabel('Sepal Width')
plt.scatter(df0['sepal length (cm)'], df0['sepal width (cm)'],color="green",marker='+')
plt.scatter(df1['sepal length (cm)'], df1['sepal width (cm)'],color="blue",marker='.')
```

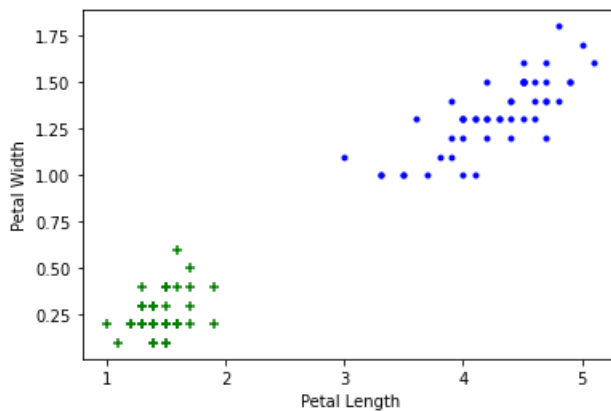
Out[15]: <matplotlib.collections.PathCollection at 0x18b41d9bd60>



Petal length vs Petal Width (Setosa vs Versicolor)

```
In [16]: plt.xlabel('Petal Length')
plt.ylabel('Petal Width')
plt.scatter(df0['petal length (cm)'], df0['petal width (cm)'],color="green",marker='+')
plt.scatter(df1['petal length (cm)'], df1['petal width (cm)'],color="blue",marker='.')
```

Out[16]: <matplotlib.collections.PathCollection at 0x18b42550400>



Train Using Support Vector Machine (SVM)

```
In [17]: from sklearn.model_selection import train_test_split
```

```
In [18]: X = df.drop(['target', 'flower_name'], axis='columns')
y = df.target
```

```
In [19]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

```
In [20]: len(X_train)
```

Out[20]: 120

```
In [21]: len(X_test)
```

Out[21]: 30

```
In [22]: from sklearn.svm import SVC
model = SVC()
```

```
In [23]: model.fit(X_train, y_train)
```

```
Out[23]: SVC()
```

```
In [24]: model.score(X_test, y_test)
```

```
Out[24]: 0.8666666666666667
```

```
In [25]: model.predict([[4.8,3.0,1.5,0.3]])
```

```
C:\Users\adnan\anaconda3\lib\site-packages\sklearn\base.py:445: UserWarning: X does not have valid  
feature names, but SVC was fitted with feature names  
warnings.warn(
```

```
Out[25]: array([0])
```

Tune parameters

1. Regularization (C)

```
In [26]: model_C = SVC(C=1)  
model_C.fit(X_train, y_train)  
model_C.score(X_test, y_test)
```

```
Out[26]: 0.8666666666666667
```

```
In [27]: model_C = SVC(C=10)  
model_C.fit(X_train, y_train)  
model_C.score(X_test, y_test)
```

```
Out[27]: 0.9333333333333333
```

1. Gamma

```
In [28]: model_g = SVC(gamma=10)  
model_g.fit(X_train, y_train)  
model_g.score(X_test, y_test)
```

```
Out[28]: 0.9333333333333333
```

1. Kernel

```
In [29]: model_linear_kernal = SVC(kernel='linear')  
model_linear_kernal.fit(X_train, y_train)
```

```
Out[29]: SVC(kernel='linear')
```

```
In [30]: model_linear_kernal.score(X_test, y_test)
```

```
Out[30]: 0.9666666666666667
```