Lab 14 04-11-2022

November 3, 2022

Program to implement text classification using Support vector machine.

Ajay Jeevan Jose

1 Support Vector Machine

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import sklearn
```

1.1 Load Dataset

[2]:
```
df = pd.read_csv('iris.csv')
df.describe()
```

[2]:

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
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

1.2 Preprocessing

[3]:
```
x = df.iloc[0:100,0:2]
y = df.iloc[0:100,-1]
print(x)
print(y)
```

```
    sepal_length  sepal_width
0            5.1          3.5
1            4.9          3.0
2            4.7          3.2
```

```
3            4.6          3.1
4            5.0          3.6
..           …          …
95           5.7          3.0
96           5.7          2.9
97           6.2          2.9
98           5.1          2.5
99           5.7          2.8

[100 rows x 2 columns]
0         setosa
1         setosa
2         setosa
3         setosa
4         setosa
         …
95    versicolor
96    versicolor
97    versicolor
98    versicolor
99    versicolor
Name: species, Length: 100, dtype: object
```

1.2.1 Standardise feature

[4]:
```
from sklearn.preprocessing import StandardScaler
scalar = StandardScaler()
x_std = scalar.fit_transform(x)
print(x)
```

```
    sepal_length  sepal_width
0            5.1          3.5
1            4.9          3.0
2            4.7          3.2
3            4.6          3.1
4            5.0          3.6
..           …          …
95           5.7          3.0
96           5.7          2.9
97           6.2          2.9
98           5.1          2.5
99           5.7          2.8

[100 rows x 2 columns]
```

1.3 Creating SVM Classifier

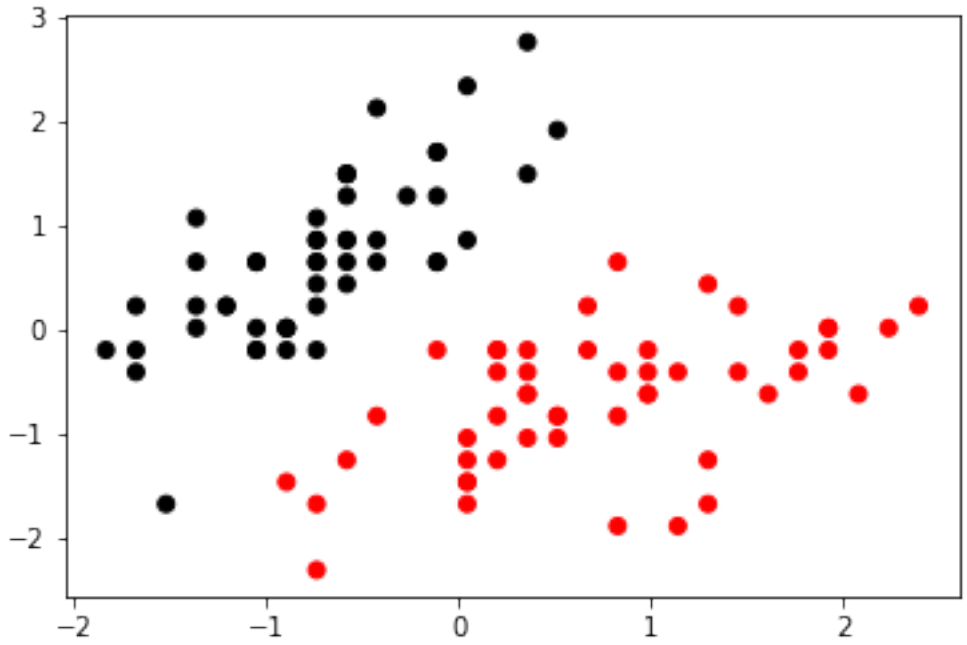
[5]:
```
from sklearn.svm import SVC
svc = SVC(C=1.0, kernel='linear', degree=3, gamma='auto')
```

[6]:
```
model = svc.fit(x_std,y)
```

1.3.1 Plotting Data Points

[7]:
```
color = ['black' if c=='setosa' else 'red' for c in y]
plt.scatter(x_std[:,0], x_std[:,1], c=color)
```

[7]:
```
<matplotlib.collections.PathCollection at 0x7fdae4c3deb0>
```



[15]:
```
print(y)
```

```
0          setosa
1          setosa
2          setosa
3          setosa
4          setosa
          ...
95     versicolor
96     versicolor
97     versicolor
98     versicolor
```

```
99    versicolor
Name: species, Length: 100, dtype: object
```

1.3.2 Creating Hyperplane

[8]:
```
w =svc.coef_
print(w)
```

```
[[ 1.93014264 -1.60829995]]
```

1.4 Visualisation

[17]:
```
color = ['black' if c == 'setosa' else 'red' for c in y]
plt.scatter(x_std[:,0], x_std[:,1], c=color)
w = svc.coef_[0]
a = -w[0] / w[1]
xx = np.linspace(-2.5, 2.5)
yy = a * xx - (svc.intercept_[0]) / w[1]
plt.plot(xx, yy)
plt.axis("off"), plt.show();
```