

Support Vector Machine(SVM) Example-5

Introduction

Social_Network_Ads data set ကို Support Vector Machine algorithm ဖြင့် classification လုပ်သည့်ကုန်များကို ရှင်းပြထားသည်။ data set ထဲတွင် feature များအဖြစ် အသက်(age), လိင်(Gender), လစာ(EstimatedSalary) တို့ပါဝင်သည်။ Label အဖြစ် ဝယ်သည် သို့မဟုတ် မဝယ်ခဲ့ပါ (Purchased)ပါသည်။

(၁)SVM algorithm အတွက် လိုအပ်သည့် library များကို import လုပ်သည်။

pandas, numpy , seaborn နှင့် matplotlib.pyplot တို့ကို import လုပ်သည်။

```
In [1]: # Importing the libraries

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

(၂) Comma Separated Values(.csv) ဖိုင်ကို ဖတ်၍ dataframe ပြုလုပ်သည်။

```
In [2]: # Importing the datasets
datasets = pd.read_csv('Social_Network_Ads.csv')
datasets.head(10)
```

Out[2]:

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0
5	15728773	Male	27	58000	0
6	15598044	Female	27	84000	0
7	15694829	Female	32	150000	1
8	15600575	Male	25	33000	0
9	15727311	Female	35	65000	0

```
In [3]: X = datasets.iloc[:, [2,3]].values
Y = datasets.iloc[:, 4].values
```

(၃) Checking the correlation between each feature

Feature တစ်ခုချင်းစီတို့၏ ဆက်စပ်မှု(correlation)ကို စစ်ဆေးသည်။

```
In [4]: datasets.corr()
```

Out[4]:

	User ID	Age	EstimatedSalary	Purchased
User ID	1.000000	-0.000721	0.071097	0.007120
Age	-0.000721	1.000000	0.155238	0.622454
EstimatedSalary	0.071097	0.155238	1.000000	0.362083
Purchased	0.007120	0.622454	0.362083	1.000000

(၄) Checking whether there is any null values

Missing value များ သို့မဟုတ် null value ရှိ မရှိ စစ်ဆေးသည်။

```
In [5]: datasets.isnull().sum()
```

```
Out[5]: User ID      0
        Gender      0
        Age         0
        EstimatedSalary  0
        Purchased    0
        dtype: int64
```

dataframe ၏ row နှင့် ကော်လံ အရေအတွက်ကို ဖတ်သည်။

```
In [6]: print('Shape of DataFrame(rows,columns):', datasets.shape)
```

```
Shape of DataFrame(rows,columns): (400, 5)
```

Dataset ထဲတွင် sample 400 ခု ပါဝင်ပြီး feature ၅ ခု ပါဝင်သည်။

(၅) Splitting the dataset into the Training set and Test set

```
In [7]: # Splitting the dataset into the Training set and Test set
        from sklearn.model_selection import train_test_split
        X_Train, X_Test, Y_Train, Y_Test = train_test_split(X, Y, test_size
        = 0.25, random_state = 0)
        X_Train[0:4]
```

```
Out[7]: array([[ 44, 39000],
               [ 32, 120000],
               [ 38, 50000],
               [ 32, 135000]], dtype=int64)
```

```
In [8]: X_Test[0:4]
```

```
Out[8]: array([[ 30, 87000],
 [ 38, 50000],
 [ 35, 75000],
 [ 30, 79000]], dtype=int64)
```

```
In [9]: Y_Train[0:4]
```

```
Out[9]: array([0, 1, 0, 1], dtype=int64)
```

```
In [10]: Y_Test[0:4]
```

```
Out[10]: array([0, 0, 0, 0], dtype=int64)
```

(၆) မူလ(original) ဒေတာများကို ဂရပ်ပုံ ထုတ်ကြည့်သည်။

```
In [11]: plt.scatter(X_Train[:,0],X_Train[:,1])
plt.title('Train Data Before Feature Scaling')
plt.grid()
```



မူလ(original) data set ထဲတွင် feature များ၏ တန်ဖိုးများသည် အလွန်ကွာခြားသည်။ ထို့ကြောင့် feature Scaling လုပ်ရန်လိုအပ်သည်။

(၇) Feature Scaling

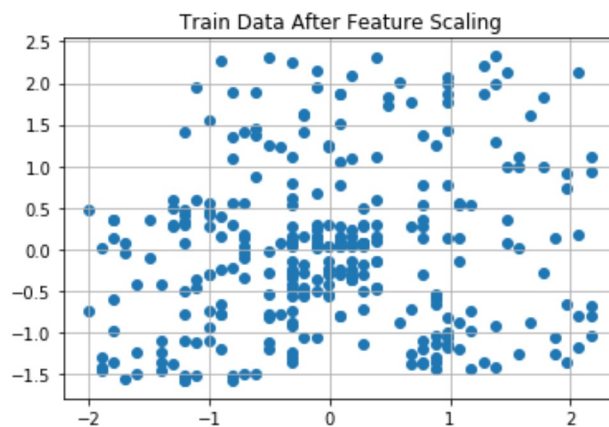
Feature Scaling လုပ်ရန် sklearn.preprocessing မှ StandardScaler ကို အသုံးပြုသည်။

```
In [12]: # Feature Scaling
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
```

Feature Scaling လုပ်သည့်အခါ Train data set (X_Train) နှင့် Test data set (X_Test) နှစ်ခုစလုံးကို တူညီသည့် Feature Scaling ပုံစံမျိုးလုပ်ရသည်။ အပေါ်မှ ဂရပ်ပုံသည် Feature Scaling မလုပ်ရသေးသည့် ပုံဖြစ်သည်။ အောက်တွင် Feature Scaling လုပ်ပြီးသည့် ဂရပ်ပုံထုတ်ကြည့်သည်။

```
In [13]: X_Train = sc_X.fit_transform(X_Train)
X_Test = sc_X.transform(X_Test)

plt.scatter(X_Train[:,0],X_Train[:,1])
plt.title('Train Data After Feature Scaling')
plt.grid()
```



(၈) Fitting the classifier into the Training set

Training set များကို ထည့်၍ SVC model တည်ဆောက်သည်။ SVC model ကို SVC classifier ဟုလည်း ခေါ်သည်။ SVC model တွင် 'linear' kernel ကို အသုံးပြုထားသည်။ C တန်ဖိုးသည် default တန်ဖိုးဖြစ်သည့် 1.0 ကို အသုံးပြုထားသည်။

```
In [14]: # Fitting the classifier into the Training set

from sklearn.svm import SVC
classifier = SVC(kernel = 'linear', random_state = 0)
classifier.fit(X_Train, Y_Train)
```

```
Out[14]: SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
      decision_function_shape='ovr', degree=3, gamma='scale', kernel='linear',
      max_iter=-1, probability=False, random_state=0, shrinking=True, tol=0.001,
      verbose=False)
```

(၉) Predicting the test set results

တည်ဆောက်ပြီးသည့် SVC model/SVC classifier ထဲသို့ Test data set (X_Test) ထည့်၍ ရလဒ်ကို ခန့်မှန်းသည်။

```
In [15]: # Predicting the test set results
Y_Pred = classifier.predict(X_Test)
```

(၁၀) Making the Confusion Matrix

ပြုလုပ်ခဲ့ပြီးသည့် classification ၏ accuracy ကို ခန့်မှန်းရန်, evaluate လုပ်ရန်အတွက် confusion matrix ကို တွက်သည်။

```
In [16]: # Making the Confusion Matrix
          from sklearn.metrics import confusion_matrix
          cm = confusion_matrix(Y_Test, Y_Pred)
          cm

Out[16]: array([[66,  2],
                [ 8, 24]], dtype=int64)
```

Training set ကို Visualising လုပ်သည်။

Visualising the Training set results

```

In [17]: from matplotlib.colors import ListedColormap
X_Set, Y_Set = X_Train, Y_Train
X1, X2 = np.meshgrid(np.arange(start = X_Set[:, 0].min() - 1, stop
                             = X_Set[:, 0].max() + 1, step = 0.01),
                     np.arange(start = X_Set[:, 1].min() - 1, stop
                             = X_Set[:, 1].max() + 1, step = 0.01))
plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(), X2.ra
vel()]).T).reshape(X1.shape),
             alpha = 0.75, cmap = ListedColormap(('red', 'green')))
plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max())

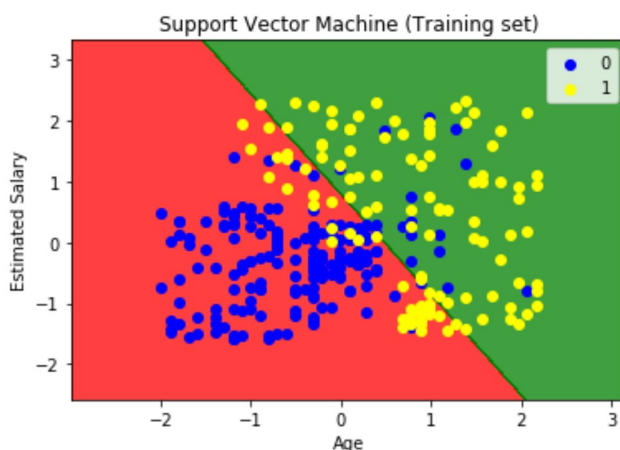
for i, j in enumerate(np.unique(Y_Set)):
    plt.scatter(X_Set[Y_Set == j, 0], X_Set[Y_Set == j, 1],
                c = ListedColormap(('blue', 'yellow'))(i), label =
j)

plt.title('Support Vector Machine (Training set)')
plt.xlabel('Age')
plt.ylabel('Estimated Salary')
plt.legend()
plt.show()

```

'c' argument looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping will have precedence in case its length matches with 'x' & 'y'. Please use a 2-D array with a single row if you really want to specify the same RGB or RGBA value for all points.

'c' argument looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping will have precedence in case its length matches with 'x' & 'y'. Please use a 2-D array with a single row if you really want to specify the same RGB or RGBA value for all points.



Test set ကို Visualising လုပ်သည်။

Visualising the Test set results

```

In [18]: from matplotlib.colors import ListedColormap
X_Set, Y_Set = X_Test, Y_Test
X1, X2 = np.meshgrid(np.arange(start = X_Set[:, 0].min() - 1, stop
                                = X_Set[:, 0].max() + 1, step = 0.01),
                    np.arange(start = X_Set[:, 1].min() - 1, stop
                                = X_Set[:, 1].max() + 1, step = 0.01))

plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(), X2.ra
vel()])).T).reshape(X1.shape),
              alpha = 0.75, cmap = ListedColormap(('red', 'green')))

plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max())
for i, j in enumerate(np.unique(Y_Set)):
    plt.scatter(X_Set[Y_Set == j, 0], X_Set[Y_Set == j, 1],
                c = ListedColormap(('blue', 'yellow'))(i), label =
j)

plt.title('Support Vector Machine (Test set)')
plt.xlabel('Age')
plt.ylabel('Estimated Salary')
plt.legend()
plt.show()

```

'c' argument looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping will have precedence in case its length matches with 'x' & 'y'. Please use a 2-D array with a single row if you really want to specify the same RGB or RGBA value for all points.

'c' argument looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping will have precedence in case its length matches with 'x' & 'y'. Please use a 2-D array with a single row if you really want to specify the same RGB or RGBA value for all points.

