support-vector-machines

December 19, 2023

0.1 This is a project of predicting the purchase status of consumers from a shop

[90]:
```
import pandas as pd
import numpy as np
```

[91]:
```
df = pd.read_csv("consumer.csv")
```

[92]:
```
df.head(5)
```

[92]:
```
    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
```

[93]:
```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
df.Gender = le.fit_transform(df.Gender)
df
```

[93]:
```
      User ID  Gender  Age  EstimatedSalary  Purchased
0    15624510       1   19            19000          0
1    15810944       1   35            20000          0
2    15668575       0   26            43000          0
3    15603246       0   27            57000          0
4    15804002       1   19            76000          0
..        …       …    …              …          …
395  15691863       0   46            41000          1
396  15706071       1   51            23000          1
397  15654296       0   50            20000          1
398  15755018       1   36            33000          0
399  15594041       0   49            36000          1

[400 rows x 5 columns]
```

[94]:
```
inputs = df.drop(["Purchased"],axis = "columns")
inputs
```

```
[94]:       User ID  Gender  Age  EstimatedSalary
     0    15624510       1   19            19000
     1    15810944       1   35            20000
     2    15668575       0   26            43000
     3    15603246       0   27            57000
     4    15804002       1   19            76000
     ..        …       …  …                …
     395  15691863       0   46            41000
     396  15706071       1   51            23000
     397  15654296       0   50            20000
     398  15755018       1   36            33000
     399  15594041       0   49            36000

     [400 rows x 4 columns]
```

```
[95]: targets = df["Purchased"]
      targets
```

```
[95]: 0      0
      1      0
      2      0
      3      0
      4      0
            ..
      395    1
      396    1
      397    1
      398    0
      399    1
      Name: Purchased, Length: 400, dtype: int64
```

```
[96]: from sklearn.svm import SVC
```

```
[97]: model = SVC()
```

```
[98]: model.fit(inputs,targets)
```

```
[98]: SVC()
```

```
[99]: model.score(inputs,targets)
```

```
[99]: 0.6425
```

```
[100]: from sklearn.model_selection import train_test_split
```

```
[101]: xtrain,xtest,ytrain,ytest = train_test_split(inputs,targets, test_size = 0.3)
```

```
[102]: len(xtrain)
```

```
[102]: 280
```

```
[103]: len(xtest)
```

```
[103]: 120
```

```
[104]: model.fit(xtrain,ytrain)
```

```
[104]: SVC()
```

```
[105]: model.score(xtest,ytest)
```

```
[105]: 0.6666666666666666
```

```
[ ]:
```

0.1.1 Hyper parameter tuning

```
[106]: model1 = SVC(kernel='linear')
```

```
[107]: model1.fit(xtrain,ytrain)
```

```
[107]: SVC(kernel='linear')
```

```
[108]: model1.score(xtest,ytest)
```

```
[108]: 0.7416666666666667
```

```
[109]: model2 = SVC(kernel='poly',degree=2)
```

```
[110]: model2.fit(xtrain,ytrain)
```

```
[110]: SVC(degree=2, kernel='poly')
```

```
[111]: model2.score(xtrain,ytrain)
```

```
[111]: 0.6321428571428571
```

```
[112]: model3 = SVC(kernel='rbf')
```

```
[113]: model3.fit(xtrain,ytrain)
```

```
[113]: SVC()
```

```
[114]: model3.score(xtrain,ytrain)
```

```
[114]: 0.6321428571428571
```

```
[115]: model1.predict(xtest)
```

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

0.1.2 Here we got score 0.7416666666666667 for model1, SVC(kernel='linear') which is comparitively higher than all other models. So using Model 1 we predicted the status of purchase with testing data

```
[ ]:
```