mlp and svm

September 25, 2018

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import LabelEncoder
from sklearn.svm import SVC
```

In [2]:
```
data=pd.read_csv('clean_bmart.csv',sep=',')
data.head()
```

Out[2]:

	Unnamed: 0	Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility	Item_Type	Item_MRP	Outlet_Identifier	Outlet_Establishment_Year	Outlet_Size	Outlet_Location_Type	Outlet_Type	Item_Outlet_Sales
0	0	FDA15	9.30	Low Fat	0.016047	Dairy	249.8092	OUT049	1999	Medium	Tier 1	Supermarket Type1	3735.1380
1	1	DRC01	5.92	Regular	0.019278	Soft Drinks	48.2692	OUT018	2009	Medium	Tier 3	Supermarket Type2	443.4228
2	2	FDN15	17.50	Low Fat	0.016760	Meat	141.6180	OUT049	1999	Medium	Tier 1	Supermarket Type1	2097.2700
3	3	FDX07	19.20	Regular	0.000000	Fruits and Vegetables	182.0950	OUT010	1998	Medium	Tier 3		
4	4	NCD19	8.93	Low Fat	0.000000	Household	53.8614	OUT013	1987	High	Tier 3		

```
        3      Grocery Store           732.3800
        4  Supermarket Type1           994.7052
```

```
In [3]: X=data.loc[(data['Outlet_Location_Type']=='Tier 1')|(data['Outlet_Location_Type']=='Ti
        x=X.values[:,:]
        y=X.values[:,10]
        ley=LabelEncoder()
        ley.fit(y)
        y=ley.transform(y)
        for i in [1,3,5,7,9,11]:
            en=LabelEncoder()
            en.fit(X.values[:,i])
            x[:,i]=en.transform(x[:,i])

        x=x[:,[1,2,3,4,5,6,7,8,9,11,12]]
        X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.3)
        print(y)
        print(x)
        print(data.columns.values[[4,8,10]])
```

```
[0 0 1 ... 1 1 0]
[[156 9.3 0 ... 0 1 3735.138]
 [659 17.5 0 ... 0 1 2097.27]
 [438 16.2 1 ... 0 1 1076.5986]
 ...
 [890 8.38 1 ... 0 1 549.285]
 [1348 10.6 0 ... 1 1 1193.1136]
 [50 14.8 0 ... 1 1 765.67]]
['Item_Visibility' 'Outlet_Establishment_Year' 'Outlet_Location_Type']
```

```
In [4]: from sklearn.neural_network import MLPClassifier
```

```
In [5]: mlp=MLPClassifier(hidden_layer_sizes=(5),max_iter=1000,random_state=0)
```

```
In [6]: mlp.fit(X_train,y_train)
```

```
Out[6]: MLPClassifier(activation='relu', alpha=0.0001, batch_size='auto', beta_1=0.9,
               beta_2=0.999, early_stopping=False, epsilon=1e-08,
               hidden_layer_sizes=5, learning_rate='constant',
               learning_rate_init=0.001, max_iter=1000, momentum=0.9,
               nesterovs_momentum=True, power_t=0.5, random_state=0, shuffle=True,
               solver='adam', tol=0.0001, validation_fraction=0.1, verbose=False,
               warm_start=False)
```

```
In [7]: predictions=mlp.predict(X_test)
```

```
In [8]: from sklearn.metrics import classification_report,confusion_matrix
        print(confusion_matrix(y_test,predictions))
```

```
[[472 286]
 [ 29 765]]
```

```
In [9]: print('Accuracy: %.2f' % accuracy_score(y_test, predictions))
```

```
Accuracy: 0.80
```

```
In [10]: model=SVC(kernel='linear',C=1E10)
         model.fit(X_train,y_train)
```

```
Out[10]: SVC(C=10000000000.0, cache_size=200, class_weight=None, coef0=0.0,
           decision_function_shape='ovr', degree=3, gamma='auto', kernel='linear',
           max_iter=-1, probability=False, random_state=None, shrinking=True,
           tol=0.001, verbose=False)
```

```
In [11]: predictions=model.predict(X_test)
```

```
In [12]: print('Accuracy: %.2f' % accuracy_score(y_test, predictions))
```

```
Accuracy: 1.00
```

```
In [13]: print(confusion_matrix(y_test,predictions))
```

```
[[758   0]
 [  0 794]]
```