D8_jithin_SVM-Usecase

August 17, 2018

1 SVM Use Case

This dataset contains information of users in a social network. So those informations are the UserID, the Gender, the Age and, the Estimated Salary. This social network has several business clients which can put their ads on the social network and one of their clients is a car company who has just launched their brand-new luxury SUV for a for ridiculous price. And we are trying to see which of these users of the social network are going to buy this brand-new SUV. The last column of the dataset tells that if yes or no the user bought this SUV. Build a model using SVM to predict if a user is going to buy or not the SUV.

```
In [1]: import pandas as pd
        import matplotlib.pyplot as plt
        from sklearn.model_selection import train_test_split
        from sklearn.svm import SVC,LinearSVC
        from sklearn.preprocessing import LabelEncoder
        from sklearn.metrics import classification_report
        from sklearn.model_selection import cross_validate


        df=pd.read_excel('social_network_ads.xlsx')

In [2]: df.isnull().sum()

Out[2]: User ID            0
        Gender             0
        Age                0
        EstimatedSalary    0
        Purchased          0
        dtype: int64

In [3]: df.apply(lambda x:[x.unique()])

Out[3]: User ID            [[15624510, 15810944, 15668575, 15603246, 1580...
        Gender                                              [[Male, Female]]
        Age                [[19, 35, 26, 27, 32, 25, 20, 18, 29, 47, 45, ...
        EstimatedSalary    [[19000, 20000, 43000, 57000, 76000, 58000, 84...
        Purchased                                                  [[0, 1]]
        dtype: object
```

```
In [4]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 5 columns):
User ID            400 non-null int64
Gender             400 non-null object
Age                400 non-null int64
EstimatedSalary    400 non-null int64
Purchased          400 non-null int64
dtypes: int64(4), object(1)
memory usage: 15.7+ KB
```

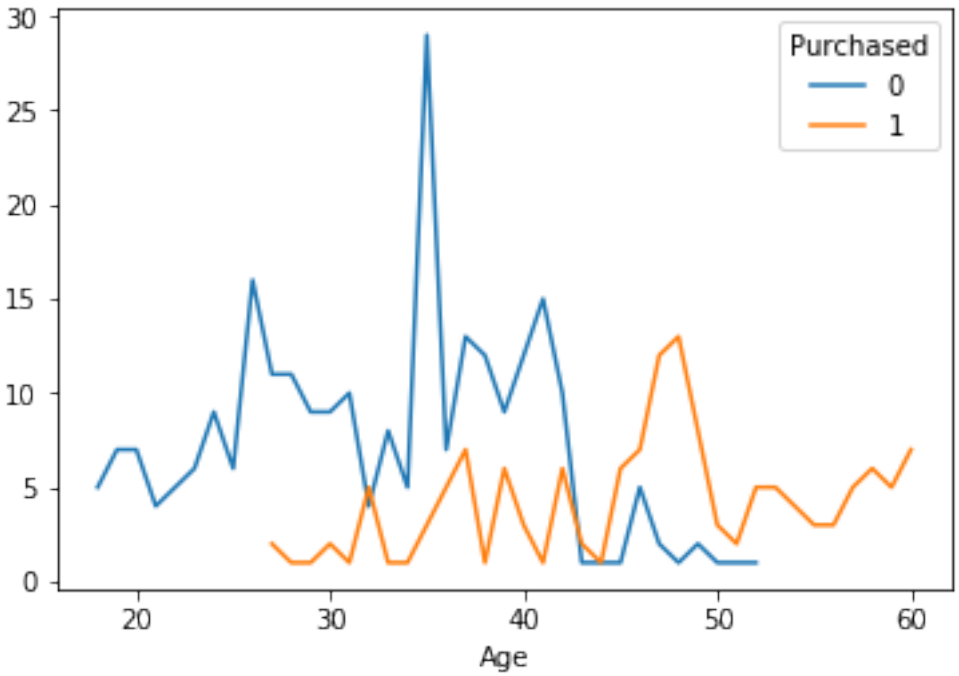
1.0.1 Data Exploration

```
In [5]: temp=pd.pivot_table(data=df, index='Gender', values='User ID',columns='Purchased', aggfu
        temp['Percent']=round(temp[1]*100/temp['All'],2)
        temp
```

```
Out[5]: Purchased    0    1  All  Percent
        Gender
        Female     127   77  204    37.75
        Male       130   66  196    33.67
        All        257  143  400    35.75
```

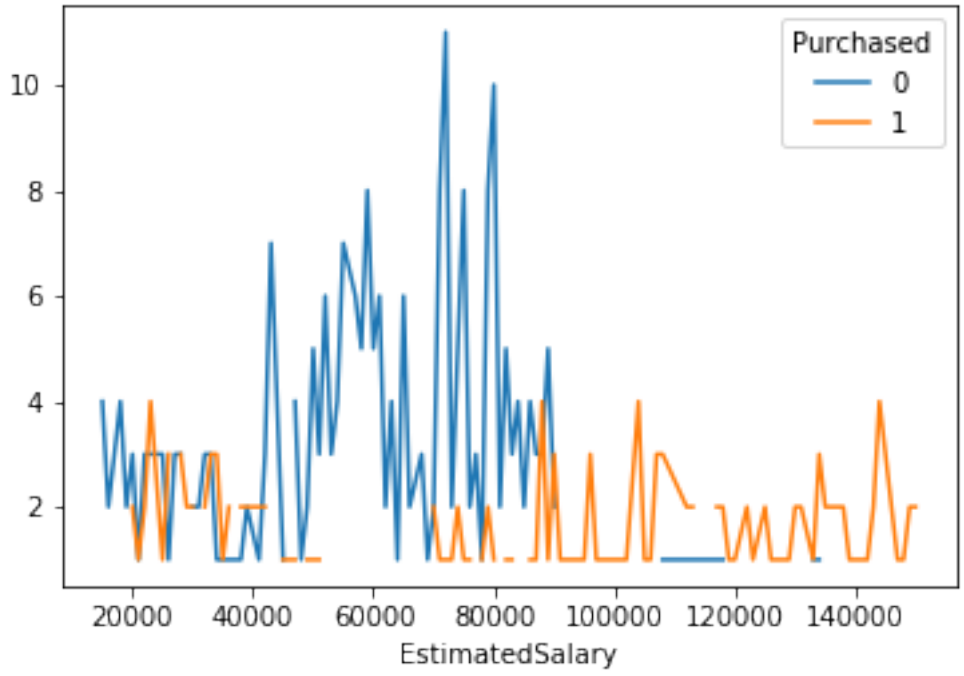
```
In [6]: pd.pivot_table(data=df, index='Age', values='User ID',columns='Purchased', aggfunc='coun
```

```
Out[6]: <matplotlib.axes._subplots.AxesSubplot at 0x7f146de28940>
```



```
In [7]: pd.pivot_table(data=df, index='EstimatedSalary', values='User ID',columns='Purchased', a
```

```
Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x7f1462c3f5f8>
```



1.1 Model Preparation

```
In [8]: df["Gender"] = df["Gender"].astype('category')
        df["Gender"] = df["Gender"].cat.codes
        df["Purchased"] = df["Purchased"].astype('category')
        df["Purchased"] = df["Purchased"].cat.codes
```

```
In [9]: y=df['Purchased']
        x=df.drop(['Purchased','User ID'],axis=1)
        x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.2,random_state=5)
```

```
In [ ]: model=SVC(class_weight='balanced',kernel='linear')
        model.fit(x_train,y_train)
        y_pred=model.predict(x_test)

        print(classification_report(y_pred,y_test))
```

	precision	recall	f1-score	support
0	0.74	0.95	0.83	41
1	0.93	0.64	0.76	39
avg / total	0.83	0.80	0.79	80

In [ ]:
```
score=cross_validate(model, x,y, scoring=['precision','accuracy','recall'],cv=4, return_
print("Accuracy after Cross Validation :",round(score['test_accuracy'].mean(),2))
print("Precision after Cross Validation:",round(score['test_precision'].mean(),2))
print("Recall after Cross Validation:",round(score['test_recall'].mean(),2))
```