Day 27 - SVM

In [1]:

import pandas as pd  
import numpy as np  
  
dataset = pd.read\_csv("Dataset/train.csv")  
dataset = dataset.drop(["Name","Ticket","Cabin","PassengerId"],axis=1)  
  
dataset.head()

Out[1]:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Survived | Pclass | Sex | Age | SibSp | Parch | Fare | Embarked |
| 0 | 0 | 3 | male | 22.0 | 1 | 0 | 7.2500 | S |
| 1 | 1 | 1 | female | 38.0 | 1 | 0 | 71.2833 | C |
| 2 | 1 | 3 | female | 26.0 | 0 | 0 | 7.9250 | S |
| 3 | 1 | 1 | female | 35.0 | 1 | 0 | 53.1000 | S |
| 4 | 0 | 3 | male | 35.0 | 0 | 0 | 8.0500 | S |

In [2]:

from sklearn import preprocessing  
  
#from sklearn.cross\_validaNtion import train\_test\_split  
from sklearn.model\_selection import train\_test\_split  
  
from sklearn.metrics import accuracy\_score  
from sklearn.metrics import confusion\_matrix  
  
from sklearn import svm  
  
le=preprocessing.LabelEncoder()  
  
le.fit(dataset["Sex"])  
print(le.classes\_)  
dataset["Sex"] = le.transform(dataset["Sex"])  
  
le.fit(dataset["Embarked"])  
print(le.classes\_)  
dataset["Embarked"] = le.transform(dataset["Embarked"])  
  
dataset['Age'] = np.round(dataset['Age'])  
dataset['Fare'] = np.round(dataset['Fare'])  
  
  
#changes columns with float type to int  
float\_col = dataset.select\_dtypes(include=['float64'])  
for col in float\_col.columns.values:  
 dataset[col] = dataset[col].astype('int64')

['female' 'male']  
['C' 'Q' 'S']

In [3]:

# Changing continious to categorical using quartiles  
dataset['Age'] = np.where((dataset.Age <= 22),0,dataset.Age)  
dataset['Age'] = np.where(((dataset.Age > 22) & (dataset.Age <= 32)),1,dataset.Age)  
dataset['Age'] = np.where(((dataset.Age > 32) & (dataset.Age <= 45)),2,dataset.Age)  
dataset['Age'] = np.where((dataset.Age > 45),3,dataset.Age)  
  
#dataset['Fare'] = np.where((dataset.Fare <= 14),0,dataset.Fare)  
#dataset['Fare'] = np.where((dataset.Fare > 14), 1,dataset.Fare)  
  
dataset['Fare'] = np.where((dataset.Fare <= 8),0,dataset.Fare)  
dataset['Fare'] = np.where(((dataset.Fare > 8) & (dataset.Fare <= 14)),1,dataset.Fare)  
dataset['Fare'] = np.where(((dataset.Fare > 14) & (dataset.Fare <= 31)),2,dataset.Fare)  
dataset['Fare'] = np.where((dataset.Fare > 31),3,dataset.Fare)  
  
dataset.head(10)

Out[3]:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Survived | Pclass | Sex | Age | SibSp | Parch | Fare | Embarked |
| 0 | 0 | 3 | 1 | 0 | 1 | 0 | 0 | 2 |
| 1 | 1 | 1 | 0 | 2 | 1 | 0 | 3 | 0 |
| 2 | 1 | 3 | 0 | 1 | 0 | 0 | 0 | 2 |
| 3 | 1 | 1 | 0 | 2 | 1 | 0 | 3 | 2 |
| 4 | 0 | 3 | 1 | 2 | 0 | 0 | 0 | 2 |
| 5 | 0 | 3 | 1 | 2 | 0 | 0 | 0 | 1 |
| 6 | 0 | 1 | 1 | 3 | 0 | 0 | 3 | 2 |
| 7 | 0 | 3 | 1 | 0 | 3 | 1 | 2 | 2 |
| 8 | 1 | 3 | 0 | 1 | 0 | 2 | 1 | 2 |
| 9 | 1 | 2 | 0 | 0 | 1 | 0 | 2 | 0 |

In [4]:

def accuracy\_fun(feature):  
 print("======================================================================================")  
 print(" === ",feature," ===\n")  
 y=dataset[feature]  
 X=dataset.drop([feature],axis=1)  
   
 print(X.head())  
 print()  
 print("Total count of (",feature,") - dependent data : ", y.count())  
   
 X\_train,X\_test,y\_train,y\_test = train\_test\_split(X,y,test\_size=0.3,random\_state=0)  
   
 clf=svm.SVC(gamma=0.01,C=100)  
   
 y\_pred=clf.fit(X\_train,y\_train).predict(X\_test)  
   
 print("\n Accuracy score : ",accuracy\_score(y\_test,y\_pred,normalize=True))  
 print("\n Confusion Matrix")  
 print(confusion\_matrix(y\_test,y\_pred))  
 print("\n======================================================================================\n\n")  
   
  
   
features = ['Survived','Pclass','Sex','Age','SibSp','Parch','Fare','Embarked']  
  
for feature in features:  
 accuracy\_fun(feature)

======================================================================================  
 === Survived ===  
  
 Pclass Sex Age SibSp Parch Fare Embarked  
0 3 1 0 1 0 0 2  
1 1 0 2 1 0 3 0  
2 3 0 1 0 0 0 2  
3 1 0 2 1 0 3 2  
4 3 1 2 0 0 0 2  
  
Total count of ( Survived ) - dependent data : 889  
  
 Accuracy score : 0.7790262172284644  
  
 Confusion Matrix  
[[132 25]  
 [ 34 76]]  
  
======================================================================================  
  
  
======================================================================================  
 === Pclass ===  
  
 Survived Sex Age SibSp Parch Fare Embarked  
0 0 1 0 1 0 0 2  
1 1 0 2 1 0 3 0  
2 1 0 1 0 0 0 2  
3 1 0 2 1 0 3 2  
4 0 1 2 0 0 0 2  
  
Total count of ( Pclass ) - dependent data : 889  
  
 Accuracy score : 0.8164794007490637  
  
 Confusion Matrix  
[[ 61 5 4]  
 [ 9 31 9]  
 [ 4 18 126]]  
  
======================================================================================  
  
  
======================================================================================  
 === Sex ===  
  
 Survived Pclass Age SibSp Parch Fare Embarked  
0 0 3 0 1 0 0 2  
1 1 1 2 1 0 3 0  
2 1 3 1 0 0 0 2  
3 1 1 2 1 0 3 2  
4 0 3 2 0 0 0 2  
  
Total count of ( Sex ) - dependent data : 889  
  
 Accuracy score : 0.7640449438202247  
  
 Confusion Matrix  
[[ 74 24]  
 [ 39 130]]  
  
======================================================================================  
  
  
======================================================================================  
 === Age ===  
  
 Survived Pclass Sex SibSp Parch Fare Embarked  
0 0 3 1 1 0 0 2  
1 1 1 0 1 0 3 0  
2 1 3 0 0 0 0 2  
3 1 1 0 1 0 3 2  
4 0 3 1 0 0 0 2  
  
Total count of ( Age ) - dependent data : 889  
  
 Accuracy score : 0.3970037453183521  
  
 Confusion Matrix  
[[27 2 38 1]  
 [ 7 3 59 2]  
 [20 6 74 2]  
 [ 3 7 14 2]]  
  
======================================================================================  
  
  
======================================================================================  
 === SibSp ===  
  
 Survived Pclass Sex Age Parch Fare Embarked  
0 0 3 1 0 0 0 2  
1 1 1 0 2 0 3 0  
2 1 3 0 1 0 0 2  
3 1 1 0 2 0 3 2  
4 0 3 1 2 0 0 2  
  
Total count of ( SibSp ) - dependent data : 889  
  
 Accuracy score : 0.7378277153558053  
  
 Confusion Matrix  
[[165 17 0 0 0 0 0]  
 [ 37 27 0 0 0 1 0]  
 [ 6 2 0 0 0 0 0]  
 [ 1 3 0 0 3 0 0]  
 [ 0 0 0 0 2 0 0]  
 [ 0 0 0 0 0 1 0]  
 [ 0 0 0 0 0 0 2]]  
  
======================================================================================  
  
  
======================================================================================  
 === Parch ===  
  
 Survived Pclass Sex Age SibSp Fare Embarked  
0 0 3 1 0 1 0 2  
1 1 1 0 2 1 3 0  
2 1 3 0 1 0 0 2  
3 1 1 0 2 1 3 2  
4 0 3 1 2 0 0 2  
  
Total count of ( Parch ) - dependent data : 889  
  
 Accuracy score : 0.7752808988764045  
  
 Confusion Matrix  
[[196 1 1 0 0]  
 [ 33 7 0 0 0]  
 [ 18 5 4 0 0]  
 [ 0 1 0 0 0]  
 [ 1 0 0 0 0]]  
  
======================================================================================  
  
  
======================================================================================  
 === Fare ===  
  
 Survived Pclass Sex Age SibSp Parch Embarked  
0 0 3 1 0 1 0 2  
1 1 1 0 2 1 0 0  
2 1 3 0 1 0 0 2  
3 1 1 0 2 1 0 2  
4 0 3 1 2 0 0 2  
  
Total count of ( Fare ) - dependent data : 889  
  
 Accuracy score : 0.7827715355805244  
  
 Confusion Matrix  
[[83 2 6 2]  
 [13 23 7 0]  
 [ 4 3 45 10]  
 [ 3 2 6 58]]  
  
======================================================================================  
  
  
======================================================================================  
 === Embarked ===  
  
 Survived Pclass Sex Age SibSp Parch Fare  
0 0 3 1 0 1 0 0  
1 1 1 0 2 1 0 3  
2 1 3 0 1 0 0 0  
3 1 1 0 2 1 0 3  
4 0 3 1 2 0 0 0  
  
Total count of ( Embarked ) - dependent data : 889  
  
 Accuracy score : 0.7528089887640449  
  
 Confusion Matrix  
[[ 0 1 48]  
 [ 0 5 17]  
 [ 0 0 196]]  
  
======================================================================================

In [ ]: