Assignment

Take Pclass, gender, SibSp, Parch, Embarked as dependent variable one by one and others as the independent variable and find the confusion matrix.

How can you minimize the code, using looping or function?

import pandas as pd

svmtrain\_dataset = pd.read\_csv("D:/AI\_ML\_Course/Day24/train.csv")

svmtrain\_dataset.columns

# Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',

# 'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],

# dtype='object')

svmtrain\_dataset = svmtrain\_dataset.drop("Cabin", axis=1)

svmtrain\_dataset = svmtrain\_dataset.drop("Ticket", axis=1)

svmtrain\_dataset = svmtrain\_dataset.drop('Name', axis=1)

svmtrain\_dataset.columns

# Index(['PassengerId', 'Survived', 'Pclass', 'Sex', 'Age', 'SibSp', 'Parch',

# 'Fare', 'Embarked'],

# dtype='object')

from sklearn import preprocessing

from sklearn.model\_selection import train\_test\_split

from sklearn.naive\_bayes import GaussianNB

from sklearn.metrics import accuracy\_score

from sklearn.metrics import confusion\_matrix

le = preprocessing.LabelEncoder()

le.fit(svmtrain\_dataset["Sex"])

#Out[56]: LabelEncoder()

print(le.classes\_)

# ['female' 'male']

svmtrain\_dataset["Sex"] = le.transform(svmtrain\_dataset["Sex"])

le.fit(svmtrain\_dataset["Embarked"])

print(le.classes\_)

# ['C' 'Q' 'S']

svmtrain\_dataset["Embarked"] = le.transform(svmtrain\_dataset["Embarked"])

y = svmtrain\_dataset["Survived"]

X = svmtrain\_dataset.drop(["Survived", "PassengerId"], axis=1)

y.count()

# Out[87]: 889

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=0)

X\_train.head()

# Out[89]:

from sklearn.naive\_bayes import \*

# Out[153]:

# Survived Pclass Sex SibSp Parch

# 350 0 1 1 0 0

# 124 1 3 1 1 0

# 577 0 3 0 1 0

# 422 0 3 0 1 1

# 118 0 3 0 4 2

clf = BernoulliNB()

y\_pred = clf.fit(X\_train, y\_train).predict(X\_test)

accuracy\_score(y\_test, y\_pred, normalize=True)

# Out[132]: 0.7715355805243446

confusion\_matrix(y\_test, y\_pred)

# Out[133]:

# array([[131, 26],

# [ 35, 75]], dtype=int64)

# Correct = 131+75 = 206

# Total Records = 267

# Acc Score = 206/267 = 77.15

Calculate the Accuracy score & Confusion matrix using loop function:

svmtrain\_dataset = svmtrain\_dataset.drop(["PassengerId", "Age", "Fare"], axis=1)

col = list(svmtrain\_dataset.columns)

print(col)

['Survived', 'Pclass', 'Sex', 'SibSp', 'Parch', 'Embarked']

for i in col:

y = svmtrain\_dataset[i]

X = svmtrain\_dataset.drop([i], axis=1)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=0)

clf = BernoulliNB()

y\_pred = clf.fit(X\_train, y\_train).predict(X\_test)

accuracy\_sc = accuracy\_score(y\_test, y\_pred, normalize=True)

print("The accuracy\_score is :", accuracy\_sc,"for the column :", i, end='\n')

c\_m = confusion\_matrix(y\_test, y\_pred)

print("The confusion\_matrix is :\n", c\_m, "for the column :", i, end='\n')

The accuracy\_score is: 0.7715355805243446 for the dependent variable: Survived

The confusion\_matrix is:

[[131 26]

[ 35 75]] for the dependent variable: Survived

The accuracy\_score is: 0.5880149812734082 for the dependent variable: Pclass

The confusion\_matrix is:

[[ 26 10 34]

[ 8 10 31]

[ 20 7 121]] for the dependent variable: Pclass

The accuracy\_score is: 0.7453183520599251 for the dependent variable: Sex

The confusion\_matrix is:

[[ 49 49]

[ 19 150]] for the dependent variable: Sex

The accuracy\_score is: 0.6891385767790262 for the dependent variable: SibSp

The confusion\_matrix is:

[[162 20 0 0 0 0 0]

[ 43 22 0 0 0 0 0]

[ 6 2 0 0 0 0 0]

[ 5 2 0 0 0 0 0]

[ 2 0 0 0 0 0 0]

[ 1 0 0 0 0 0 0]

[ 2 0 0 0 0 0 0]] for the dependent variable: SibSp

The accuracy\_score is: 0.7153558052434457 for the dependent variable: Parch

The confusion\_matrix is:

[[182 16 0 0 0]

[ 31 9 0 0 0]

[ 24 3 0 0 0]

[ 1 0 0 0 0]

[ 1 0 0 0 0]] for the dependent variable: Parch

The accuracy\_score is: 0.7340823970037453 for the dependent variable: Embarked

The confusion\_matrix is:

[[ 0 0 49]

[ 0 0 22]

[ 0 0 196]] for the dependent variable: Embarked