**PRACTICAL:2**

**AIM:**

For given Dataset (you may continue to use the same processed dataset from experiment 1 only for this experiment), perform the following using Python Pandas and scikit-learn library or by writing your own user-defined function:

1. Perform Data Standardization and Normalization
2. Select the 10 best features of the data using different statistical scoring methods. (Hint: Chi-Squared Statistical Test is a good scoring method)
3. Split the data into training and testing sets in a ratio of 80:20.

**CODE:**

import pandas as pd

df=pd.read\_csv('fullspecs.csv')

df=df.dropna()

df

df.drop(4,inplace=True)

df.replace(to\_replace='Yes',value='1',inplace=True)

df.replace(to\_replace='No',value='0',inplace=True)

df.rename(columns={'2019 Acura RDX Specs: FWD w/Technology Pkg':'column\_1','2019 Acura RDX Specs: FWD w/Advance Pkg':'column\_2','2019 Acura RDX Specs: FWD w/A-Spec Pkg':'column\_3'},inplace=True)

df

df=df.astype({'column\_1':float,'column\_2':float,'column\_3':float})

df.dtypes

df.describe()

df.isnull().sum()

y=df['column\_1']

y

x=df[['column\_1','column\_2','column\_3']]

x

x1=df.iloc[0:2,1:3]

x1

print(x.shape)

print(y.shape)

x.head()

y.head()

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

x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.2,random\_state=15)

x\_train

x\_test

y\_train

y\_test

print(x\_train.shape)

print(y\_train.shape)

print(y\_test.shape)

from sklearn.preprocessing import StandardScaler

scaler=StandardScaler()

print(scaler.fit(x\_train))

scaler=StandardScaler().fit(x\_train)

std\_x=scaler.transform(x\_train)

std\_x

from sklearn.preprocessing import Normalizer

transformer=Normalizer().fit(x\_test)

transformer

transformer=Normalizer().fit(x\_test)

transformer=scaler.transform(x\_test)

transformer

df.head(1)

from sklearn.feature\_selection import SelectKBest

from sklearn.feature\_selection import chi2

bestfeaters=SelectKBest(score\_func=chi2,k=3)

fit=bestfeaters.fit(x,y)

dfscores=pd.DataFrame(fit.scores\_)

dfcolumns=pd.DataFrame(x.columns)

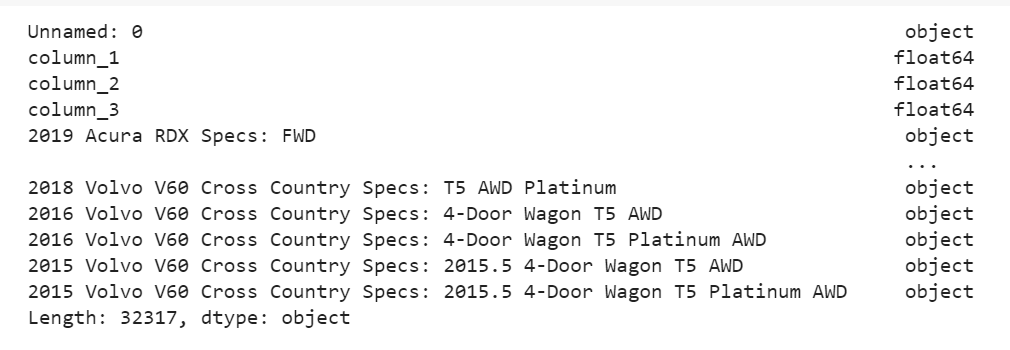
featureScores=pd.concat([dfcolumns,dfscores],axis=1)

featureScores

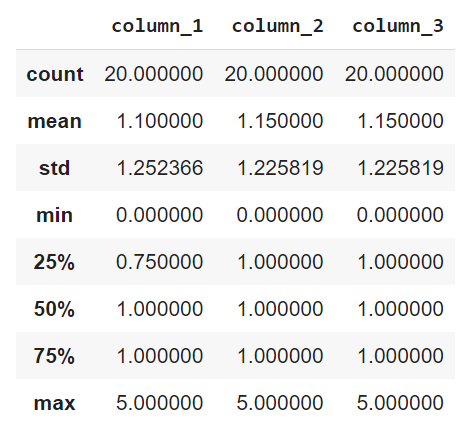
featureScores.columns=['Spec','Scores']

featureScores.nlargest(3,'Scores')

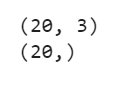
**OUTPUT:**



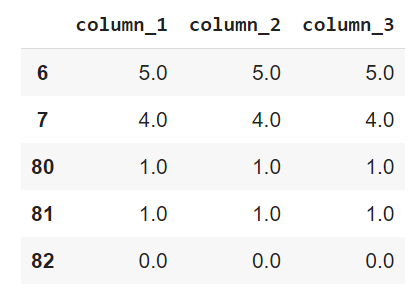
*Details of df*



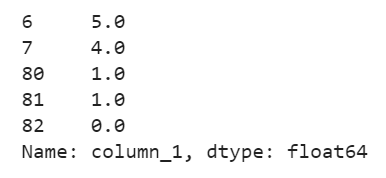
*Output of describe()*



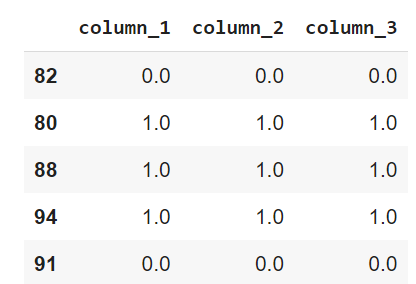
*Dimensions of x and y*



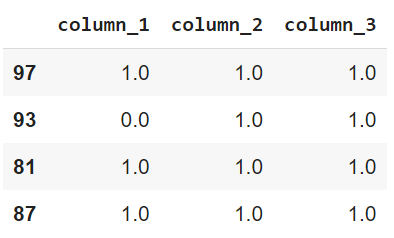
*x.head()*



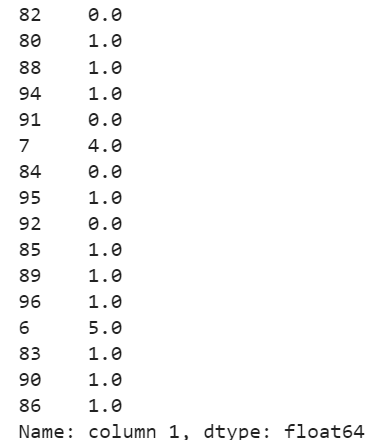
*y.head()*



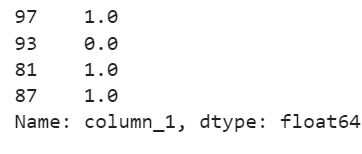
*Glimpses of x\_train*



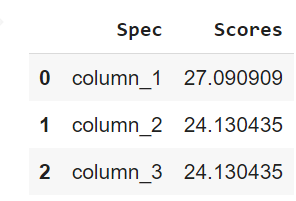
*X\_test*



*Y\_train*



*Y\_test*



*Feature Scores*