**#!pip install scikit-learn pandas numpy dtale notebook**

**import pandas as pd**

**import numpy as np**

**#import dtale**

**from sklearn.impute import SimpleImputer**

**from sklearn.compose import ColumnTransformer**

**from sklearn.preprocessing import LabelEncoder,OneHotEncoder**

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

**from sklearn.preprocessing import StandardScaler**

**data={' Region' : ['India','Brazil','USA','Brazil','USA','India','Brazil','India','nan','India'],**

**' Age':['49','32','35','43','45','40','nan','53','55','42'],**

**' Income':['86400','57600','64800','73200','nan','69600','62400','94800','99600','80400'],**

**' Online Shopper':['No','Yes','No','No','Yes','Yes','No','Yes','No','Yes']**

**}**

**dataset=pd.DataFrame(data)**

**print(dataset)**

Region Age Income Online Shopper

0 India 49 86400 No

1 Brazil 32 57600 Yes

2 USA 35 64800 No

3 Brazil 43 73200 No

4 USA 45 nan Yes

5 India 40 69600 Yes

6 Brazil nan 62400 No

7 India 53 94800 Yes

8 nan 55 99600 No

9 India 42 80400 Yes

**X=dataset.iloc[:,:-1].values**

**Y=dataset.iloc[:, -1].values**

**print(X)**

**print(Y)**

[['India' '49' '86400']

['Brazil' '32' '57600']

['USA' '35' '64800']

['Brazil' '43' '73200']

['USA' '45' 'nan']

['India' '40' '69600']

['Brazil' 'nan' '62400']

['India' '53' '94800']

['nan' '55' '99600']

['India' '42' '80400']]

['No' 'Yes' 'No' 'No' 'Yes' 'Yes' 'No' 'Yes' 'No' 'Yes']

**imputer = SimpleImputer(missing\_values=np.nan,strategy="mean")**

**imputer = imputer.fit(X[:,1:])**

**X[:,1:] = imputer.transform(X[:,1:])**

**print(X)**

[['India' 49.0 86400.0]

['Brazil' 32.0 57600.0]

['USA' 35.0 64800.0]

['Brazil' 43.0 73200.0]

['USA' 45.0 76533.33333333333]

['India' 40.0 69600.0]

['Brazil' 43.77777777777778 62400.0]

['India' 53.0 94800.0]

['nan' 55.0 99600.0]

['India' 42.0 80400.0]]

[5]:

**labelencoder\_X = LabelEncoder()**

**X[:,0]=labelencoder\_X.fit\_transform(X[:,0])**

**ct=ColumnTransformer([("Region",OneHotEncoder(),[0])],remainder='passthrough')**

**X=ct.fit\_transform(X)**

**labelencoder\_Y=LabelEncoder()**

**Y=labelencoder\_Y.fit\_transform(Y)**

**print(X)**

**print(Y)**

[[0.0 1.0 0.0 0.0 49.0 86400.0]

[1.0 0.0 0.0 0.0 32.0 57600.0]

[0.0 0.0 1.0 0.0 35.0 64800.0]

[1.0 0.0 0.0 0.0 43.0 73200.0]

[0.0 0.0 1.0 0.0 45.0 76533.33333333333]

[0.0 1.0 0.0 0.0 40.0 69600.0]

[1.0 0.0 0.0 0.0 43.77777777777778 62400.0]

[0.0 1.0 0.0 0.0 53.0 94800.0]

[0.0 0.0 0.0 1.0 55.0 99600.0]

[0.0 1.0 0.0 0.0 42.0 80400.0]]

[0 1 0 0 1 1 0 1 0 1]

[6]:

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

**print(X\_train)**

[[0.0 1.0 0.0 0.0 42.0 80400.0]

[1.0 0.0 0.0 0.0 32.0 57600.0]

[1.0 0.0 0.0 0.0 43.77777777777778 62400.0]

[0.0 1.0 0.0 0.0 53.0 94800.0]

[1.0 0.0 0.0 0.0 43.0 73200.0]

[0.0 1.0 0.0 0.0 49.0 86400.0]

[0.0 1.0 0.0 0.0 40.0 69600.0]]

[13]:

**sc\_X = StandardScaler()**

**X\_train = sc\_X.fit\_transform(X\_train)**

**X\_test = sc\_X.transform(X\_test)**

**print(X\_train)**

[[-0.8660254 0.8660254 0. 0. -0.2029809 0.44897083]

[ 1.15470054 -1.15470054 0. 0. -1.82168936 -1.41706417]

[ 1.15470054 -1.15470054 0. 0. 0.08478949 -1.0242147 ]

[-0.8660254 0.8660254 0. 0. 1.5775984 1.62751925]

[ 1.15470054 -1.15470054 0. 0. -0.04111006 -0.14030338]

[-0.8660254 0.8660254 0. 0. 0.93011502 0.94003267]

[-0.8660254 0.8660254 0. 0. -0.52672259 -0.43494049]]

[ ]: