import pandas as pd

import numPy as np

import matplotlib. pyplot as plt

import seaborn as sns

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

from sklearn. linear\_model import LogisticRegression

from sklearn. metrics import classification\_report

from sklearn. metrics import confusion\_matrix

from sklearn.model\_selection import cross\_val\_predict

from sklearn. metrics import precision\_score, recall\_score

%matplotlib inline

ad\_data = pd.read\_excel('dataset2.xlsx')

ad\_data

ad\_data.head()

ad\_data.info()

ad\_data.describe()

sns.set\_style('whitegrid')

ad\_data['traffic\_source'].hist(bins=30)

plt.xlabel('traffic\_source')

pd.crosstab(ad\_data['country'], ad\_data['sequence\_number']).sort\_values( 1,ascending = False).tail(10)

ad\_data[ad\_data['sequence\_number']==1]['country'].value\_counts().head(10)

ad\_data['country'].value\_counts().head(10)

pd.crosstab(index=ad\_data['country'],columns='count').sort\_values(['count'], ascending=False).head(10)

ad\_data.isnull().sum()

type(ad\_data['event\_timestamp'][1])

ad\_data['event\_timestamp'] = pd.to\_datetime(ad\_data['event\_timestamp'])

ad\_data['Month'] = ad\_data['event\_timestamp'].dt.month

ad\_data['Day'] = ad\_data['event\_timestamp'].dt.day

ad\_data['Hour'] = ad\_data['event\_timestamp'].dt.hour

ad\_data["Weekday"] = ad\_data['event\_timestamp'].dt. dayofweek

ad\_data = ad\_data.drop(['event\_timestamp'], axis=1)

ad\_data.head()

sns.countplot(x = 'traffic\_source', data = ad\_data)

sns.jointplot(x = "sequence\_number", y= "daily\_time\_spent\_onsite", data =a) ad\_dat

sns.scatterplot(x = "sequence\_number", y= "daily\_time\_spent\_onsite",hue='traffic\_source', data = ad\_data)

sns.lmplot(x = "sequence\_number", y= "daily\_time\_spent\_onsite",hue='traffic\_source', data = ad\_data)

sns.pairplot(ad\_data, hue = 'traffic\_source', vars = ['daily\_time\_spent\_onsite', 'sequence\_number', 'user\_id', 'daily\_internet\_usage'],palette = 'rocket')

plots = ['daily\_time\_spent\_onsite', 'sequence\_number', 'user\_id','daily\_internet\_usage']

for i in plots:

plt.figure(figsize = (12, 6))

plt.subplot(2,3,1)

sns.boxplot(data= ad\_data, y=ad\_data[i],x='traffic\_source')

plt.subplot(2,3,2)

sns.boxplot(data= ad\_data, y=ad\_data[i])

plt.subplot(2,3,3)

sns.distplot(ad\_data[i],bins= 20,)

plt.tight\_layout()

plt.title(i)

plt.show()

fig = plt.figure(figsize = (12,10))

sns.heatmap(ad\_data.corr(), cmap='viridis', annot = True)

f,ax=plt.subplots(1,2,figsize=(14,5))

ad\_data['Month'][ad\_data['traffic\_source']==1].value\_counts().sort\_index().plot(ax=ax[0])

ax[0].set\_ylabel('Count of Clicks')

pd.crosstab(ad\_data["traffic\_source"], ad\_data["Month"]).T.plot(kind = 'bar',ax=ax[1])

ad\_data.groupby(['Month'])['traffic\_source'].sum()

plt.tight\_layout()

plt.suptitle('Months Vs Clicks',y=0,size=20)

plt.show()