* Importing All The Necessary Modules

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.preprocessing import StandardScaler

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import confusion\_matrix, accuracy\_score

After importing all the Necessary modules,

Here we are Uploading the Dataset which we are using for the

Analysis.

**Dataset Name :- “ Leads.csv”**

# uploading data

lead\_df = pd.read\_csv("Leads.csv")

To upload the data we have used the above Command.

After Successfully uploading the data, now we are checking the no of Records

And no columns with the help of the following command.

#checking no of records and no of columns

lead\_df.shape

* **Understanding the Dataset**

First we checked for the null values and missing values in the given dataset.

After that we count the null values and NAN values for each column in our data set.

After understanding the dataset we dropped the ‘ Prospect ID ’ and

‘ Lead number ’ Columns as they are just working as indexes.

There will not be any significance in output or prediction of these two columns.

#Dropping Prospect ID and Lead Number columns as they are just working as index

#There will no be any significance in output of prediction of this two columns

lead\_df = lead\_df.drop(['Prospect ID','Lead Number'],axis=1)

Later we calculated the percentage of null values in each column.

# percentage of null values in each column

print("Percentage of null values:\n")

for i in lead\_df.columns:

if(lead\_df[i].isnull().sum() != 0):

print(i+":", round((lead\_df[i].isnull().sum()/lead\_df.shape[0])\* 100,2))

1. Here we can see there are some columns which are having more than 40% null values.
2. Even though we replaced null values it will create a biased significant to output variable.
3. Due to this unreliable significant is get created to output variable
4. so that it is better to drop this columns

Name of the Dropped Columns :-

* 'Lead Quality',
* 'Asymmetrique Activity Index',
* 'Asymmetrique Profile Index',
* 'Asymmetrique Activity Score',
* 'Asymmetrique Profile Score'.

# Analyzing each categorical Features

Here we are performing operations on the particular columns to Analyze the Features of the Given categorical Data.

* Using ‘ Lead Source ‘ column :-

lead\_df['Lead Source'].unique()

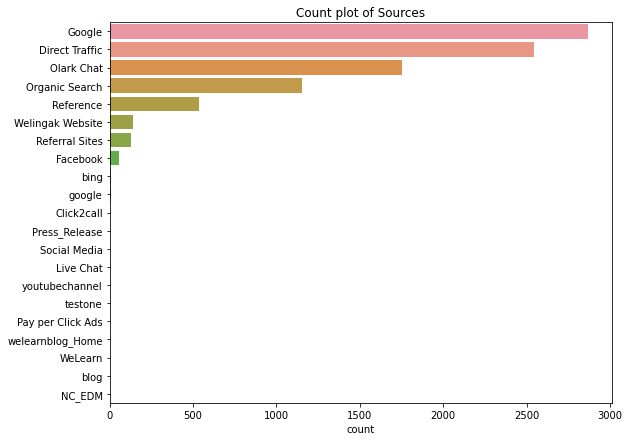
# count plot of sources

plt.figure(figsize = (9, 7))

sns.countplot(y = lead\_df["Lead Source"].values, order = lead\_df["Lead Source"].value\_counts().index)

plt.title("Count plot of Sources")

plt.show()

 By graph we can see majority is of google so that we can replace null value with google

lead\_df['Lead Source'] = lead\_df['Lead Source'].fillna('Google')

* Using ‘ Country’ column :-

lead\_df['Country'].unique()

# count plot of Country

plt.figure(figsize = (9, 10))

sns.countplot(y = lead\_df["Country"].values, order = lead\_df["Country"].value\_counts().index)

plt.title("Count plot of Country")

plt.show()



* Using ‘City’ column :-

lead\_df["City"].unique()

lead\_df[lead\_df["City"].isin(["Mumbai", "Thane & Outskirts", "Other Cities of Maharashtra"])]["Country"].isnull().sum()

for i in ["Mumbai", "Thane & Outskirts", "Other Cities of Maharashtra"]:

lead\_df.loc[lead\_df.City == i, "Country"] = "India"

lead\_df.loc[lead\_df.Country == "unknown", "Country"] = "Unknown"

lead\_df["Country"] = lead\_df["Country"].fillna("Unknown")

* Using ‘Specialization’ column :-

lead\_df['Specialization'].unique()

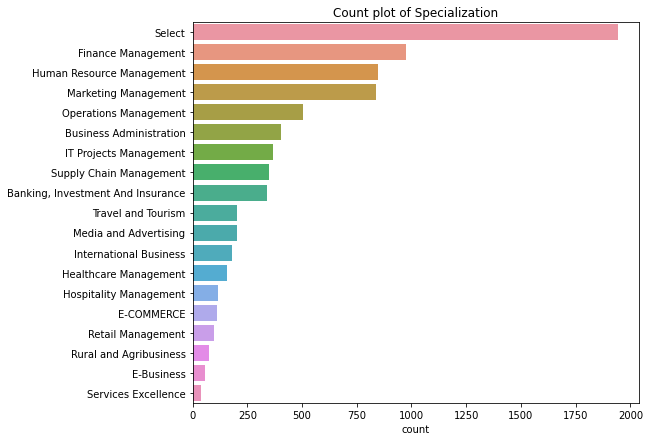
# count plot of Specialization

plt.figure(figsize = (8, 7))

sns.countplot(y = lead\_df["Specialization"].values, order = lead\_df["Specialization"].value\_counts().index)

plt.title("Count plot of Specialization")

plt.show()



lead\_df['Specialization'] = lead\_df['Specialization'].replace('Select',np.nan)

lead\_df['Specialization']=lead\_df['Specialization'].fillna('Unknown')

Here we are Replacing all the ‘Select’ values with nan values.

After that we have filled these nan values with ‘Unknown’.

* Using ‘**What is your current occupation’** column :-

lead\_df['What is your current occupation'].unique()

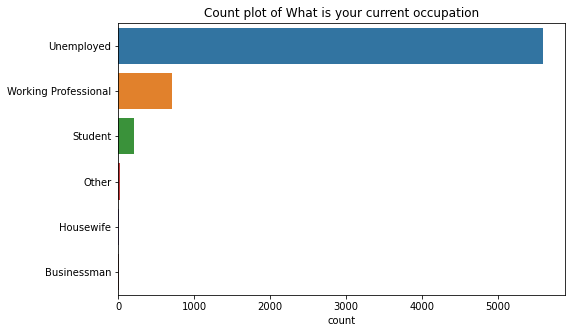
# count plot of What is your current occupation

plt.figure(figsize = (8, 5))

sns.countplot(y = lead\_df['What is your current occupation'].values, order = lead\_df['What is your current occupation'].value\_counts().index)

plt.title('Count plot of What is your current occupation')

plt.show()



We can replace null value by the mode of that column

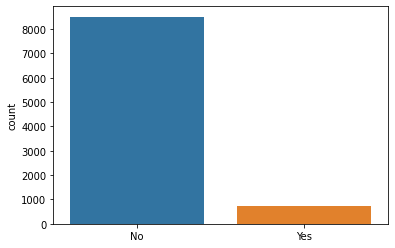
lead\_df['What is your current occupation'] = lead\_df['What is your current occupation'].fillna(lead\_df['What is your current occupation'].mode()[0])

* Using ‘**Do Not Email’** column :-

lead\_df['Do Not Email'].unique()

sns.countplot(x=lead\_df['Do Not Email'].values)

plt.show()



* Using **‘Last Activity’** column :-

lead\_df['Last Activity'].unique()

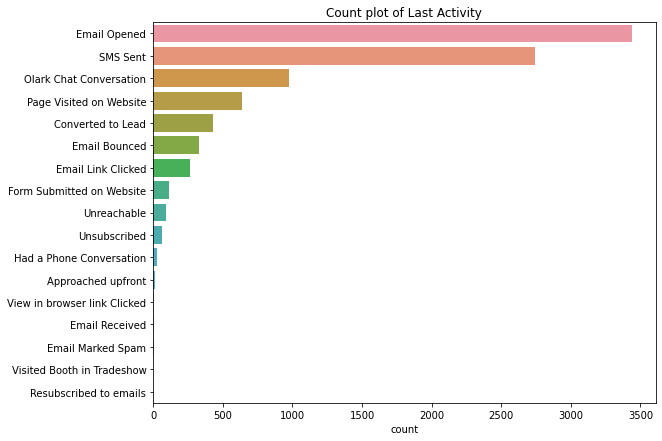
# count plot of Last Activity

plt.figure(figsize = (9, 7))

sns.countplot(y = lead\_df['Last Activity'].values, order = lead\_df['Last Activity'].value\_counts().index)

plt.title('Count plot of Last Activity')

plt.show()



From this it is clear that we can replace nan values with mode of the last activity

lead\_df['Last Activity'] = lead\_df['Last Activity'].fillna(lead\_df['Last Activity'].mode()[0])

* Using **‘How did you hear about X Education’** column :-

lead\_df['How did you hear about X Education'].unique()

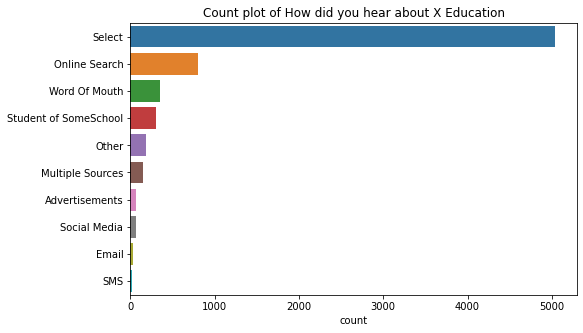
# count plot of How did you hear about X Education

plt.figure(figsize = (8, 5))

sns.countplot(y = lead\_df['How did you hear about X Education'].values, order = lead\_df['How did you hear about X Education'].value\_counts().index)

plt.title('Count plot of How did you hear about X Education')

plt.show()



Select is invalid, we have to replace it.

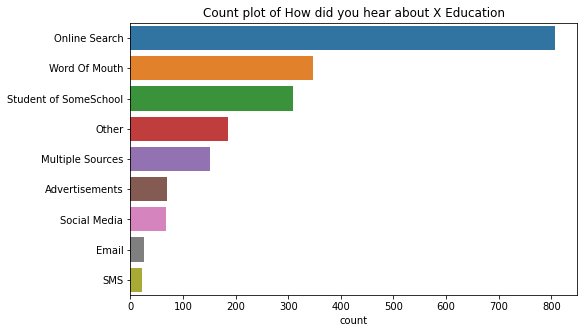
lead\_df['How did you hear about X Education'] = lead\_df['How did you hear about X Education'].replace('Select',np.nan)

plt.figure(figsize = (8, 5))

sns.countplot(y = lead\_df['How did you hear about X Education'].values, order = lead\_df['How did you hear about X Education'].value\_counts().index)

plt.title('Count plot of How did you hear about X Education')

plt.show()



* Using **‘What matters most to you in choosing a course’** column :-

lead\_df['What matters most to you in choosing a course'].unique()

# count plot of What matters most to you in choosing a course

plt.figure(figsize = (8, 5))

sns.countplot(x = lead\_df['What matters most to you in choosing a course'].values, order = lead\_df['What matters most to you in choosing a course'].value\_counts().index)

plt.title('Count plot of What matters most to you in choosing a course')

plt.show()



1. From above figure and observation we can say that there is null value present in feature
2. as graph shows there is majority of better career prospects
3. sso that we can replace null value with better career prospects

lead\_df['What matters most to you in choosing a course'] = lead\_df['What matters most to you in choosing a course'].fillna('Better Career Prospects')

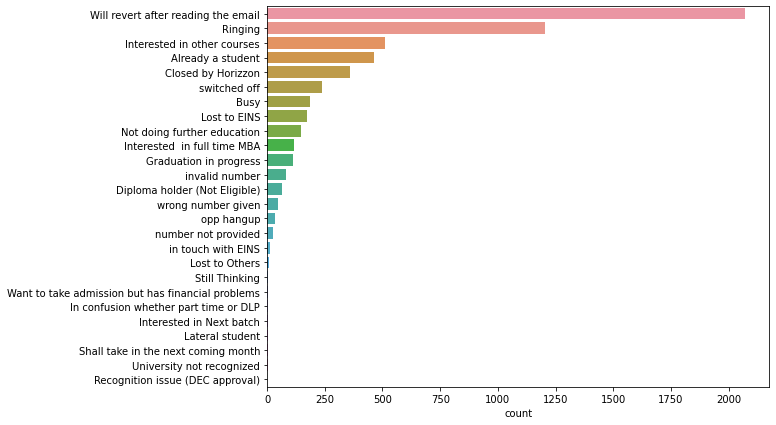
* Using **‘Tags’** column :-

lead\_df['Tags'].unique()

plt.figure(figsize = (9, 7))

sns.countplot(y = lead\_df['Tags'].values, order = lead\_df["Tags"].value\_counts().index)

plt.show()



with the help of graph we have replace the null value in tag with mode of that column

lead\_df['Tags'] = lead\_df['Tags'].fillna(lead\_df['Tags'].mode()[0])

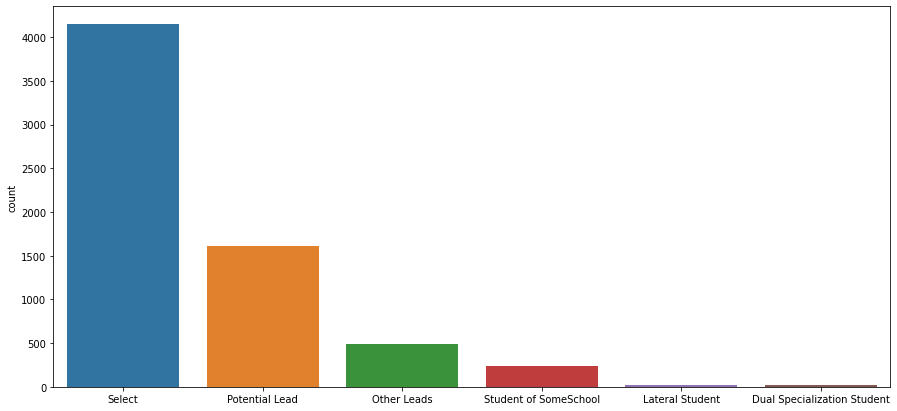
* Using **‘Lead Profile’** column :-

lead\_df['Lead Profile'].unique()

plt.figure(figsize = (15, 7))

sns.countplot(x = lead\_df['Lead Profile'].values, order = lead\_df["Lead Profile"].value\_counts().index)

plt.show()



we are replacing select with null because we don't know the profile of them and can not placed any other category

lead\_df['Lead Profile'] = lead\_df['Lead Profile'].replace('Select',np.nan)

* Using **'City'**  column :-

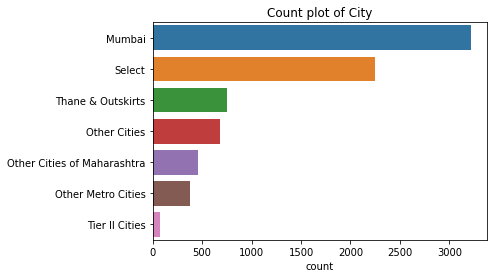
lead\_df['City'].unique()

# count plot of City

sns.countplot(y = lead\_df['City'].values, order = lead\_df['City'].value\_counts().index)

plt.title('Count plot of City')

plt.show()



* here we will replace missing value with null first
* then will replace null value with mode of column

lead\_df['City']=lead\_df['City'].replace('Select',np.nan)

lead\_df[lead\_df['Country'].isin(['India'])]['City'].isnull().sum()

lead\_df.loc[lead\_df.Country == 'India', 'City'] = 'Mumbai'

lead\_df[lead\_df['Country'].isin(['India'])]['City'].isnull().sum()

lead\_df['City'] = lead\_df['City'].fillna('Unknown')

lead\_df.isna().sum()

#From above we can drop column which are having more than 50% null values

lead\_df = lead\_df.drop(['How did you hear about X Education','Lead Profile'],axis=1)

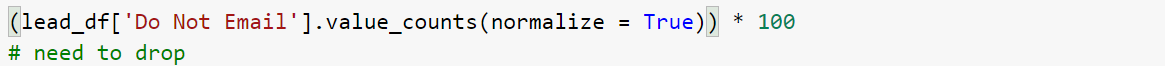
# Checking biased categorical Data

* Till now we have replaced missing values and null values.
* But from observation we have found that we have imbalanced data in our dataset.
* We will need to clear the imbalanced feature data.
* We are taking a call that if data is biased by 70% then we are going to drop that column.

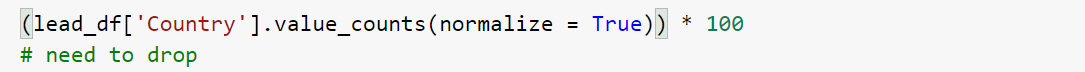


* In the following code cells we will be calculating distribution of different values in the said column.

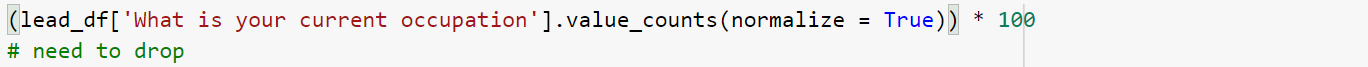


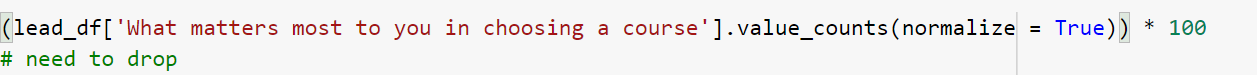


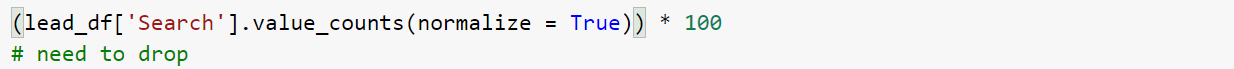


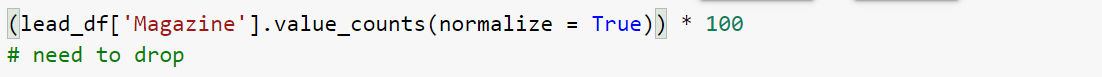


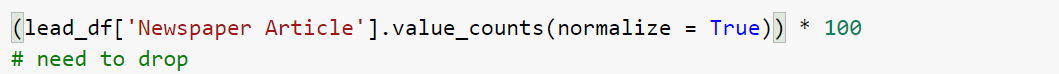


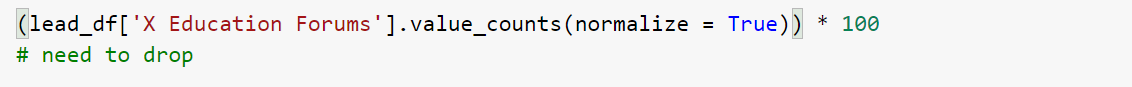


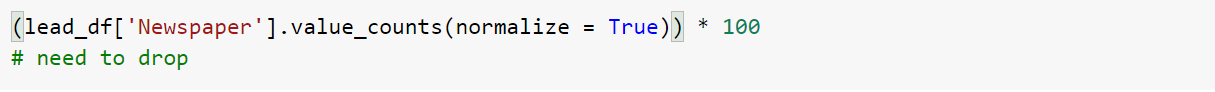


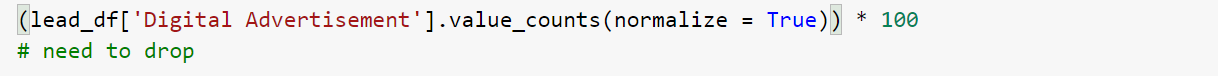


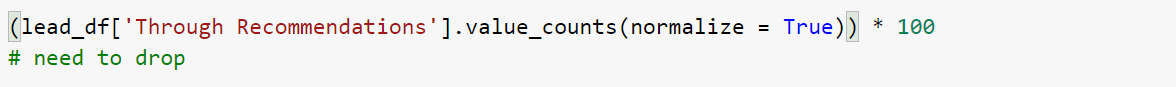


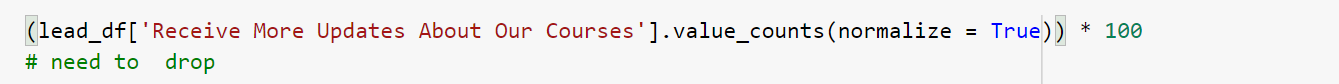




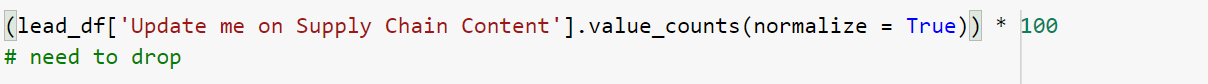


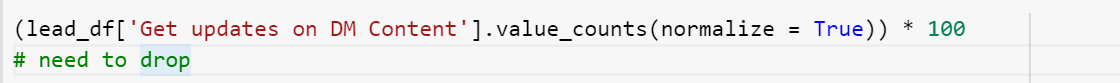






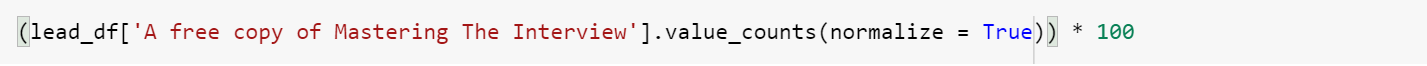






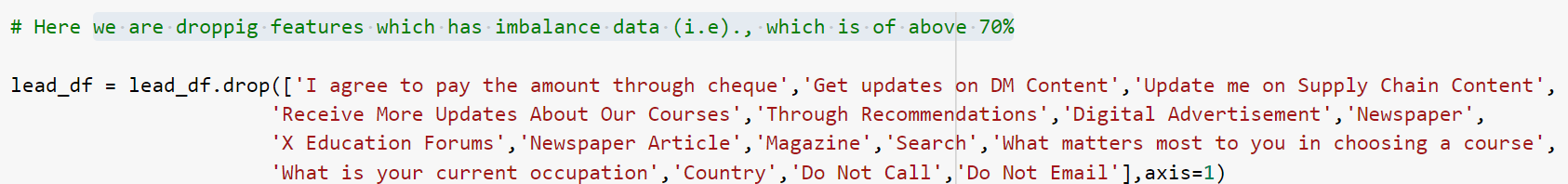






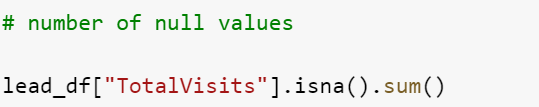


* **we are dropping features which has imbalance data (i.e)., which is of above 70%**

****

## Analyzing Numerical Values

* **Finding number of null values**

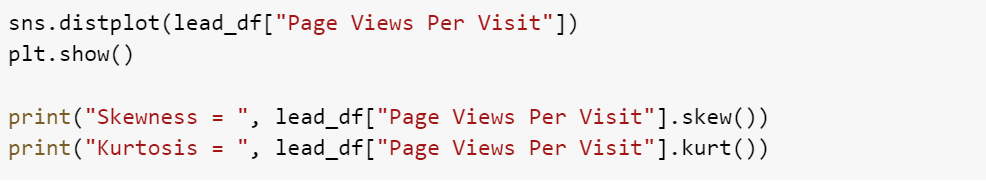
****

****

* **Mean imputation works better if the distribution is normally-distributed or has a Gaussian distribution.**
* **While median imputation is preferable for skewed data(be it right or left).**

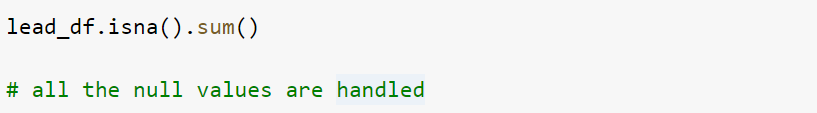
****

* Here also it is positively skewed.
* We can replace null values with median.



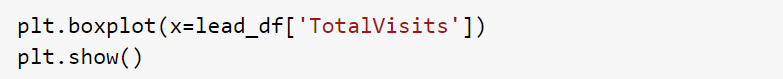


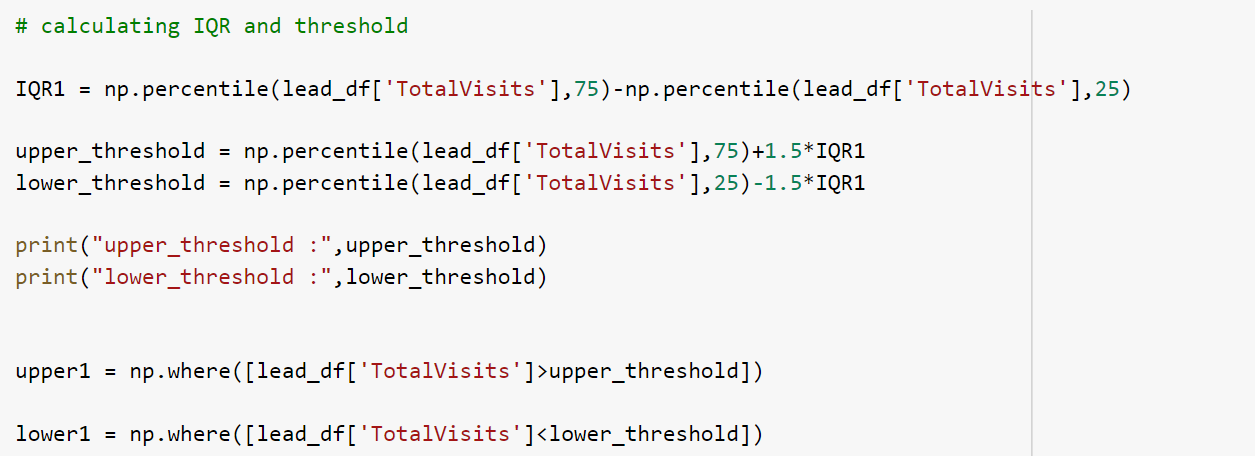
* **all the null values are handled**

****

## Finding Outlier

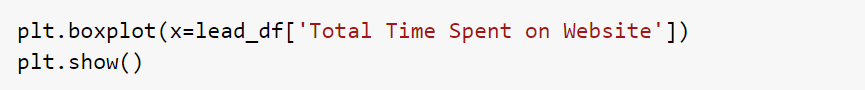
All the outliers from TotalVisits are deleted

****

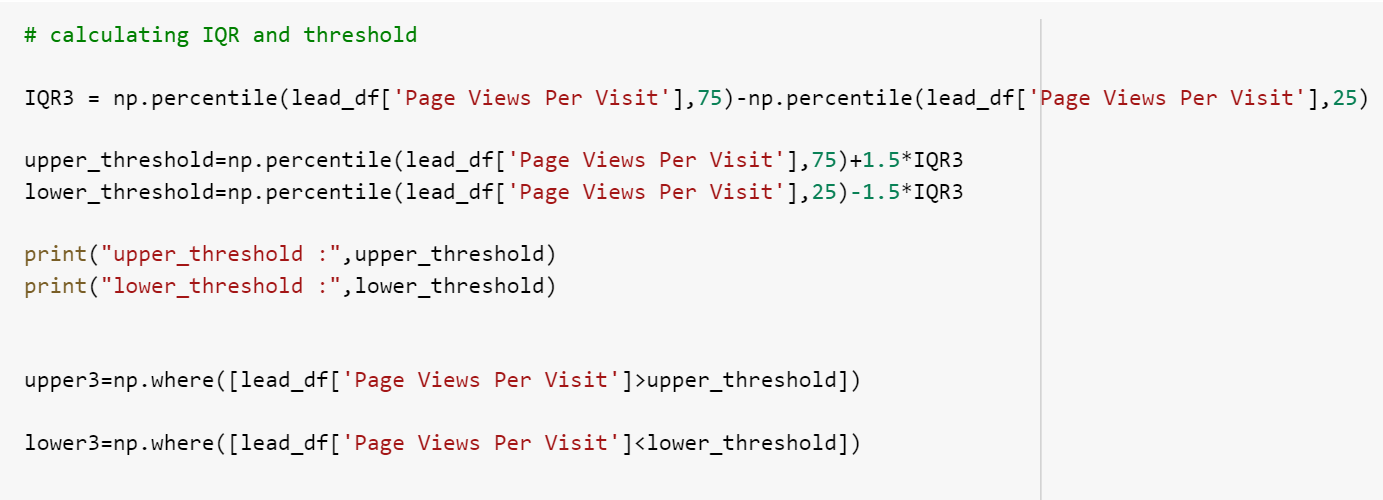
****



No outliers present in Total Spent on Website

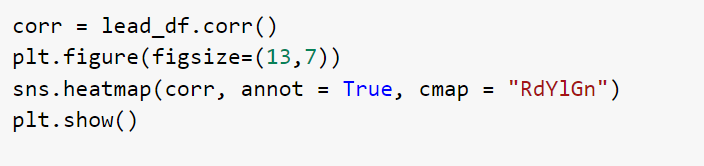


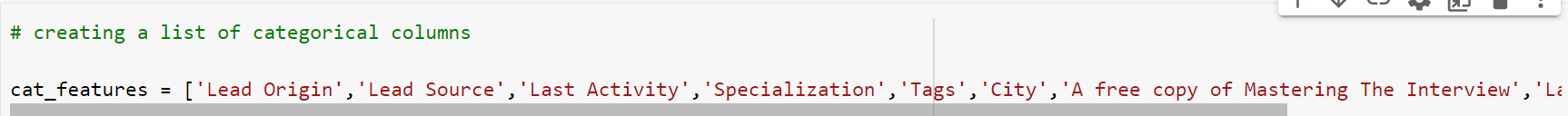


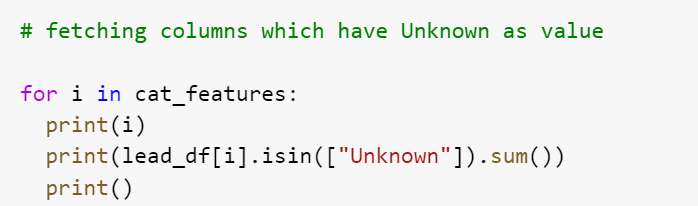


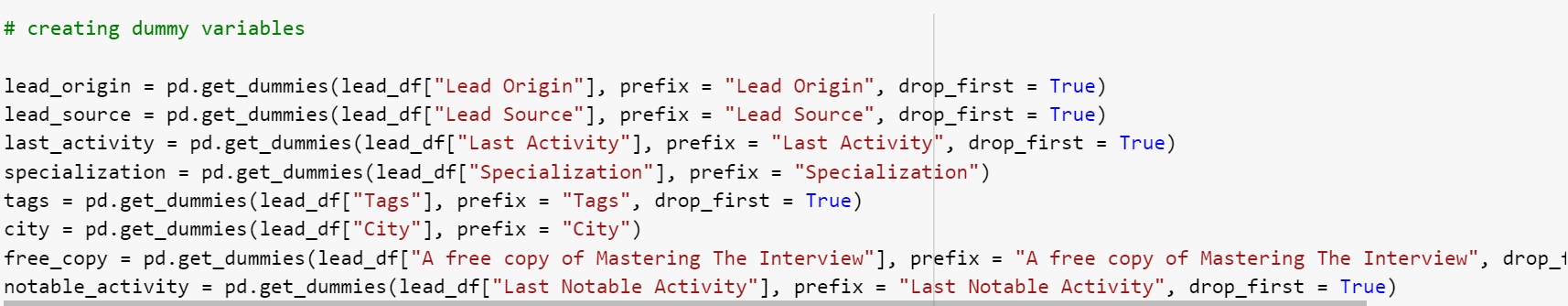


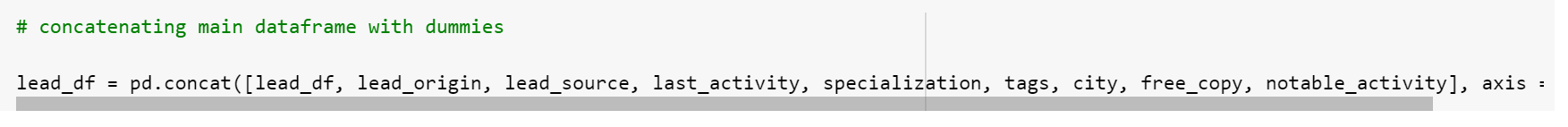
* As all outliers are deleted and null values are replaced, now the data is clean.

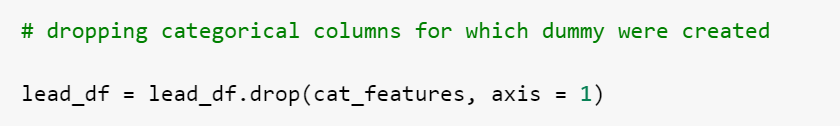


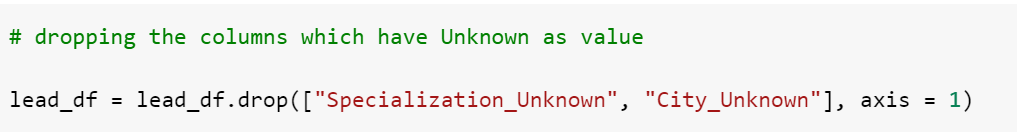




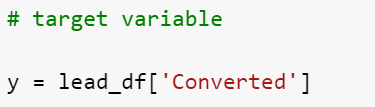


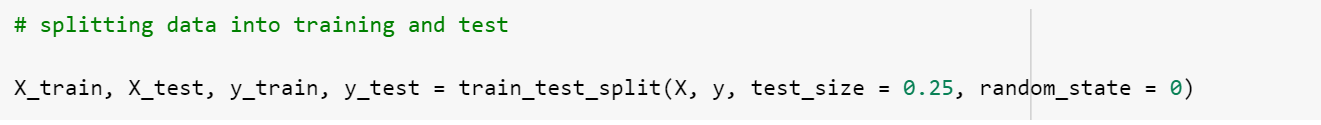


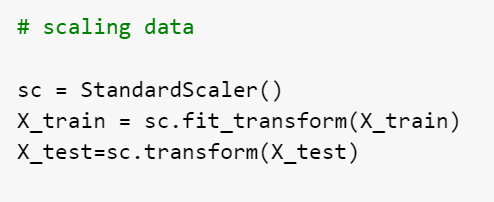


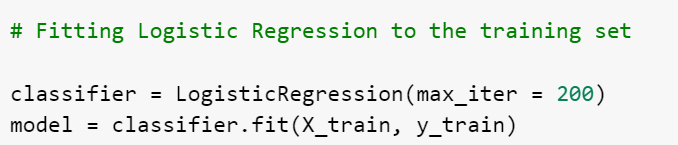


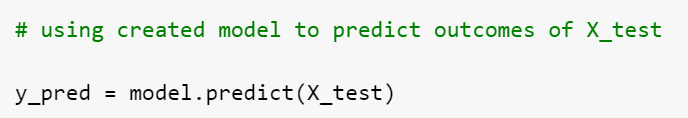


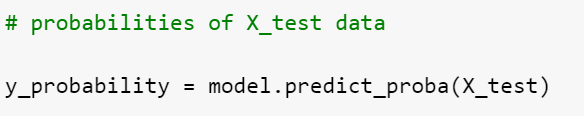




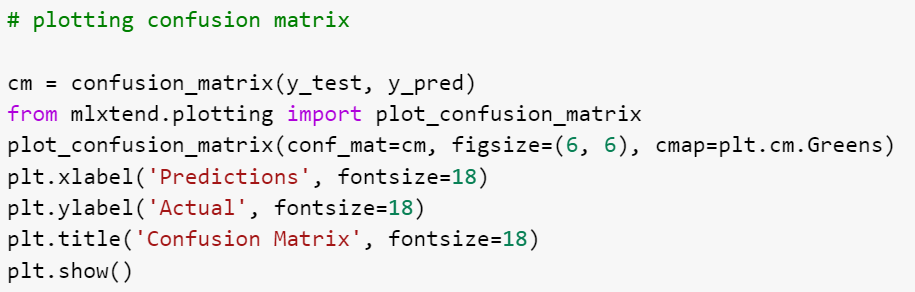


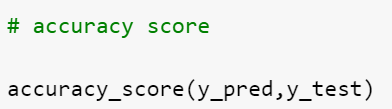












* **The accuracy of the model is 89.49%**