In [1]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn.linear_model import LogisticRegression
from sklearn.feature_selection import RFE
import statsmodels.api as sm
from statsmodels.stats.outliers_influence import variance_inflation_factor
from sklearn import metrics
from sklearn.metrics import precision_recall_curve
```

In [3]:
```
# Read the dataset
leads = pd.read_csv(r"C:\Users\dell\Downloads\Lead+Scoring+Case+Study\Lead Scoring Assig
leads.head()
```

Out[3]:

	Prospect ID	Lead Number	Lead Origin	Lead Source	Do Not Email	Do Not Call	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	...	Get updates on DM Content
0	7927b2df-8bba-4d29-b9a2-b6e0beafe620	660737	API	Olark Chat	No	No	0	0.0	0	0.0	...	No
1	2a272436-5132-4136-86fa-dcc88c88f482	660728	API	Organic Search	No	No	0	5.0	674	2.5	...	No
2	8cc8c611-a219-4f35-ad23-fdfd2656bd8a	660727	Landing Page Submission	Direct Traffic	No	No	1	2.0	1532	2.0	...	No
3	0cc2df48-7cf4-4e39-9de9-19797f9b38cc	660719	Landing Page Submission	Direct Traffic	No	No	0	1.0	305	1.0	...	No
4	3256f628-e534-4826-9d63-4a8b88782852	660681	Landing Page Submission	Google	No	No	1	2.0	1428	1.0	...	No

5 rows × 37 columns

In [4]:
```
leads.shape
```

Out[4]:
```
(9240, 37)
```

In [5]:
```
# Inspect the different columsn in the dataset
leads.columns
```

Out[5]:
```
Index(['Prospect ID', 'Lead Number', 'Lead Origin', 'Lead Source',
       'Do Not Email', 'Do Not Call', 'Converted', 'TotalVisits',
       'Total Time Spent on Website', 'Page Views Per Visit', 'Last Activity',
       'Country', 'Specialization', 'How did you hear about X Education',
       'What is your current occupation',
       'What matters most to you in choosing a course', 'Search', 'Magazine',
       'Newspaper Article', 'X Education Forums', 'Newspaper',
       'Digital Advertisement', 'Through Recommendations',
       'Receive More Updates About Our Courses', 'Tags', 'Lead Quality',
       'Update me on Supply Chain Content', 'Get updates on DM Content',
       'Lead Profile', 'City', 'Asymmetrique Activity Index',
       'Asymmetrique Profile Index', 'Asymmetrique Activity Score',
       'Asymmetrique Profile Score',
       'I agree to pay the amount through cheque',
       'A free copy of Mastering The Interview', 'Last Notable Activity'],
      dtype='object')
```

In [6]:
```
# Check the summary of the dataset
leads.describe()
```

Out[6]:

	Lead Number	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Asymmetrique Activity Score	Asymmetrique Profile Score
count	9240.000000	9240.000000	9103.000000	9240.000000	9103.000000	5022.000000	5022.000000
mean	617188.435606	0.385390	3.445238	487.698268	2.362820	14.306252	16.344883
std	23405.995698	0.486714	4.854853	548.021466	2.161418	1.386694	1.811395
min	579533.000000	0.000000	0.000000	0.000000	0.000000	7.000000	11.000000
25%	596484.500000	0.000000	1.000000	12.000000	1.000000	14.000000	15.000000
50%	615479.000000	0.000000	3.000000	248.000000	2.000000	14.000000	16.000000
75%	637387.250000	1.000000	5.000000	936.000000	3.000000	15.000000	18.000000
max	660737.000000	1.000000	251.000000	2272.000000	55.000000	18.000000	20.000000

In [7]:
```
# Check the info to see the types of the feature variables and the null values present
leads.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9240 entries, 0 to 9239
Data columns (total 37 columns):
 #   Column                                         Non-Null Count  Dtype  
---  ------                                         --------------  -----  
 0   Prospect ID                                    9240 non-null   object 
 1   Lead Number                                    9240 non-null   int64  
 2   Lead Origin                                    9240 non-null   object 
 3   Lead Source                                    9204 non-null   object 
 4   Do Not Email                                   9240 non-null   object 
 5   Do Not Call                                    9240 non-null   object 
 6   Converted                                      9240 non-null   int64  
 7   TotalVisits                                    9103 non-null   float64
 8   Total Time Spent on Website                    9240 non-null   int64  
 9   Page Views Per Visit                           9103 non-null   float64
 10  Last Activity                                  9137 non-null   object 
 11  Country                                        6779 non-null   object 
 12  Specialization                                 7802 non-null   object 
 13  How did you hear about X Education             7033 non-null   object 
 14  What is your current occupation                6550 non-null   object 
 15  What matters most to you in choosing a course  6531 non-null   object 
 16  Search                                         9240 non-null   object 
 17  Magazine                                       9240 non-null   object 
 18  Newspaper Article                              9240 non-null   object 
 19  X Education Forums                             9240 non-null   object 
 20  Newspaper                                      9240 non-null   object 
 21  Digital Advertisement                          9240 non-null   object 
 22  Through Recommendations                        9240 non-null   object 
 23  Receive More Updates About Our Courses         9240 non-null   object 
 24  Tags                                           5887 non-null   object 
 25  Lead Quality                                   4473 non-null   object 
 26  Update me on Supply Chain Content              9240 non-null   object 
 27  Get updates on DM Content                      9240 non-null   object 
 28  Lead Profile                                   6531 non-null   object 
 29  City                                           7820 non-null   object 
 30  Asymmetrique Activity Index                    5022 non-null   object 
 31  Asymmetrique Profile Index                     5022 non-null   object 
 32  Asymmetrique Activity Score                    5022 non-null   float64
 33  Asymmetrique Profile Score                     5022 non-null   float64
 34  I agree to pay the amount through cheque       9240 non-null   object 
 35  A free copy of Mastering The Interview          9240 non-null   object 
 36  Last Notable Activity                          9240 non-null   object 
dtypes: float64(4), int64(3), object(30)
memory usage: 2.6+ MB
```

Looks like there are quite a few categorical variables present in this dataset for which we will need to create dummy variables. Also, there are a lot of null values present as well, so we will need to treat them accordingly.

Step 1: Data Cleaning and Preparation

In [8]:

```
# Check the number of missing values in each column
leads.isnull().sum()
```

Out[8]:
```
Prospect ID                                         0
Lead Number                                         0
Lead Origin                                         0
Lead Source                                        36
Do Not Email                                        0
Do Not Call                                         0
Converted                                           0
TotalVisits                                       137
Total Time Spent on Website                         0
Page Views Per Visit                              137
Last Activity                                     103
Country                                          2461
Specialization                                   1438
How did you hear about X Education               2207
What is your current occupation                  2690
What matters most to you in choosing a course    2709
Search                                              0
Magazine                                            0
Newspaper Article                                   0
X Education Forums                                  0
Newspaper                                           0
Digital Advertisement                               0
Through Recommendations                             0
Receive More Updates About Our Courses              0
Tags                                             3353
Lead Quality                                     4767
Update me on Supply Chain Content                   0
Get updates on DM Content                           0
Lead Profile                                     2709
City                                             1420
Asymmetrique Activity Index                      4218
Asymmetrique Profile Index                       4218
Asymmetrique Activity Score                      4218
Asymmetrique Profile Score                       4218
I agree to pay the amount through cheque            0
A free copy of Mastering The Interview              0
Last Notable Activity                               0
dtype: int64
```

As you can see there are a lot of columnw which have high number of missing values. Clearly, these columns are not useful. Since, there are 9000 datapoints in our dataframe, let's eliminate the columns having greater than 3000 missing values as they are of no use to us.

In [9]:
```
# Drop all the columns in which greater than 3000 missing values are present
for col in leads.columns:
    if leads[col].isnull().sum() > 3000:
        leads.drop(col, 1, inplace=True)
```

In [10]:
```
#Check the number of null values again
leads.isnull().sum()
```

Out[10]:
```
Prospect ID                                         0
Lead Number                                         0
Lead Origin                                         0
Lead Source                                        36
Do Not Email                                        0
Do Not Call                                         0
Converted                                           0
TotalVisits                                       137
Total Time Spent on Website                         0
Page Views Per Visit                              137
Last Activity                                     103
Country                                          2461
Specialization                                   1438
How did you hear about X Education               2207
What is your current occupation                  2690
What matters most to you in choosing a course    2709
Search                                              0
Magazine                                            0
Newspaper Article                                   0
X Education Forums                                  0
Newspaper                                           0
Digital Advertisement                               0
Through Recommendations                             0
Receive More Updates About Our Courses              0
Update me on Supply Chain Content                   0
Get updates on DM Content                           0
Lead Profile                                     2709
City                                             1420
I agree to pay the amount through cheque            0
A free copy of Mastering The Interview              0
Last Notable Activity                               0
dtype: int64
```

As you might be able to interpret, the variable City won't be of any use in our analysis. So it's best that we drop it.

In [11]:
```
leads.drop(['City'], axis = 1, inplace = True)
```

In [12]:
```
# Same goes for the variable 'Country'
leads.drop(['Country'], axis = 1, inplace = True)
```

In [13]:
```
# Let's now check the percentage of missing values in each column
round(100*(leads.isnull().sum()/len(leads.index)), 2)
```

Out[13]:
```
Prospect ID                                      0.00
Lead Number                                      0.00
Lead Origin                                      0.00
Lead Source                                      0.39
Do Not Email                                     0.00
Do Not Call                                      0.00
Converted                                        0.00
TotalVisits                                      1.48
Total Time Spent on Website                      0.00
Page Views Per Visit                             1.48
Last Activity                                    1.11
Specialization                                  15.56
How did you hear about X Education              23.89
What is your current occupation                 29.11
What matters most to you in choosing a course   29.32
Search                                           0.00
Magazine                                         0.00
Newspaper Article                                0.00
X Education Forums                               0.00
Newspaper                                        0.00
Digital Advertisement                            0.00
Through Recommendations                          0.00
Receive More Updates About Our Courses           0.00
Update me on Supply Chain Content                0.00
Get updates on DM Content                        0.00
Lead Profile                                    29.32
I agree to pay the amount through cheque         0.00
A free copy of Mastering The Interview           0.00
Last Notable Activity                            0.00
dtype: float64
```

In [14]:
```
# Check the number of null values again
leads.isnull().sum()
```

Out[14]:
```
Prospect ID                                         0
Lead Number                                         0
Lead Origin                                         0
Lead Source                                        36
Do Not Email                                        0
Do Not Call                                         0
Converted                                           0
TotalVisits                                       137
Total Time Spent on Website                         0
Page Views Per Visit                              137
Last Activity                                     103
Specialization                                   1438
How did you hear about X Education               2207
What is your current occupation                  2690
What matters most to you in choosing a course    2709
Search                                              0
Magazine                                            0
Newspaper Article                                   0
X Education Forums                                  0
Newspaper                                           0
Digital Advertisement                               0
Through Recommendations                             0
Receive More Updates About Our Courses              0
Update me on Supply Chain Content                   0
Get updates on DM Content                           0
Lead Profile                                     2709
I agree to pay the amount through cheque            0
A free copy of Mastering The Interview              0
Last Notable Activity                               0
dtype: int64
```

Now recall that there are a few columns in which there is a level called 'Select' which basically means that the student had not selected the option for that particular column which is why it shows 'Select'. These values are as good as missing values and hence we need to identify the value counts of the level 'Select' in all the columns that it is present.

In [15]:

```
# Get the value counts of all the columns
for column in leads:
    print(leads[column].astype('category').value_counts())
    print('___________________________________________________')
```

```
000104b9-23e4-4ddc-8caa-8629fe8ad7f4    1
a7a319ea-b6ae-4c6b-afc5-183b933d10b5    1
aa27a0af-eeab-4007-a770-fa8a93fa53c8    1
aa30ebb2-8476-41ce-9258-37cc025110d3    1
aa405742-17ac-4c65-b19e-ab91c241cc53    1
                                       ..
539eb309-df36-4a89-ac58-6d3651393910    1
539ffa32-1be7-4fe1-b04c-faf1bab763cf    1
53aabd84-5dcc-4299-bbe3-62f3764b07b1    1
53ac14bd-2bb2-4315-a21c-94562d1b6b2d    1
fffb0e5e-9f92-4017-9f42-781a69da4154    1
Name: Prospect ID, Length: 9240, dtype: int64
```

```
579533    1
629593    1
630390    1
630403    1
630405    1
         ..
602534    1
602540    1
602557    1
602561    1
660737    1
Name: Lead Number, Length: 9240, dtype: int64
```

```
Landing Page Submission    4886
API                        3580
Lead Add Form               718
Lead Import                  55
Quick Add Form                1
Name: Lead Origin, dtype: int64
```

```
Google               2868
Direct Traffic       2543
Olark Chat           1755
Organic Search       1154
Reference             534
Welingak Website      142
Referral Sites        125
Facebook               55
bing                    6
google                  5
Click2call              4
Press_Release           2
Social Media            2
Live Chat               2
WeLearn                 1
Pay per Click Ads       1
NC_EDM                  1
blog                    1
testone                 1
welearnblog_Home        1
youtubechannel          1
Name: Lead Source, dtype: int64
```

```
No     8506
Yes     734
Name: Do Not Email, dtype: int64
```

```
No     9238
Yes       2
Name: Do Not Call, dtype: int64
```

```
0    5679
1    3561
Name: Converted, dtype: int64
___________________________________________________
0.0      2189
2.0      1680
3.0      1306
4.0      1120
5.0       783
6.0       466
1.0       395
7.0       309
8.0       224
9.0       164
10.0      114
11.0       86
13.0       48
12.0       45
14.0       36
16.0       21
15.0       18
17.0       16
18.0       15
20.0       12
19.0        9
23.0        6
21.0        6
24.0        5
25.0        5
27.0        5
22.0        3
26.0        2
28.0        2
29.0        2
54.0        1
141.0       1
115.0       1
74.0        1
55.0        1
30.0        1
43.0        1
42.0        1
41.0        1
32.0        1
251.0       1
Name: TotalVisits, dtype: int64
___________________________________________________
0       2193
60        19
75        18
74        18
127       18
        ...
1091       1
1088       1
1085       1
1084       1
2272       1
Name: Total Time Spent on Website, Length: 1731, dtype: int64
___________________________________________________
0.0     2189
2.0     1795
3.0     1196
4.0      896
```
Loading [MathJax]/extensions/Safe.js

```
1.0      651
        ...
3.57       1
3.8        1
3.82       1
3.83       1
55.0       1
Name: Page Views Per Visit, Length: 114, dtype: int64
___________________________________________________
Email Opened                    3437
SMS Sent                        2745
Olark Chat Conversation          973
Page Visited on Website          640
Converted to Lead                428
Email Bounced                    326
Email Link Clicked               267
Form Submitted on Website        116
Unreachable                       93
Unsubscribed                      61
Had a Phone Conversation          30
Approached upfront                 9
View in browser link Clicked       6
Email Received                     2
Email Marked Spam                  2
Resubscribed to emails             1
Visited Booth in Tradeshow         1
Name: Last Activity, dtype: int64
___________________________________________________
Select                                 1942
Finance Management                      976
Human Resource Management               848
Marketing Management                    838
Operations Management                   503
Business Administration                 403
IT Projects Management                  366
Supply Chain Management                 349
Banking, Investment And Insurance       338
Media and Advertising                   203
Travel and Tourism                      203
International Business                  178
Healthcare Management                   159
Hospitality Management                  114
E-COMMERCE                              112
Retail Management                       100
Rural and Agribusiness                   73
E-Business                               57
Services Excellence                      40
Name: Specialization, dtype: int64
___________________________________________________
Select                   5043
Online Search             808
Word Of Mouth             348
Student of SomeSchool     310
Other                     186
Multiple Sources          152
Advertisements             70
Social Media               67
Email                      26
SMS                        23
Name: How did you hear about X Education, dtype: int64
___________________________________________________
Unemployed              5600
Working Professional     706
Student                  210
```

```
Other                     16
Housewife                 10
Businessman                8
Name: What is your current occupation, dtype: int64
___________________________________________________
Better Career Prospects      6528
Flexibility & Convenience       2
Other                           1
Name: What matters most to you in choosing a course, dtype: int64
___________________________________________________
No     9226
Yes      14
Name: Search, dtype: int64
___________________________________________________
No    9240
Name: Magazine, dtype: int64
___________________________________________________
No     9238
Yes       2
Name: Newspaper Article, dtype: int64
___________________________________________________
No     9239
Yes       1
Name: X Education Forums, dtype: int64
___________________________________________________
No     9239
Yes       1
Name: Newspaper, dtype: int64
___________________________________________________
No     9236
Yes       4
Name: Digital Advertisement, dtype: int64
___________________________________________________
No     9233
Yes       7
Name: Through Recommendations, dtype: int64
___________________________________________________
No    9240
Name: Receive More Updates About Our Courses, dtype: int64
___________________________________________________
No    9240
Name: Update me on Supply Chain Content, dtype: int64
___________________________________________________
No    9240
Name: Get updates on DM Content, dtype: int64
___________________________________________________
Select                           4146
Potential Lead                   1613
Other Leads                       487
Student of SomeSchool             241
Lateral Student                    24
Dual Specialization Student        20
Name: Lead Profile, dtype: int64
___________________________________________________
No    9240
Name: I agree to pay the amount through cheque, dtype: int64
___________________________________________________
No     6352
Yes    2888
Name: A free copy of Mastering The Interview, dtype: int64
___________________________________________________
Modified                          3407
Email Opened                      2827
SMS Sent                          2172
```
Loading [MathJax]/extensions/Safe.js

```
Page Visited on Website          318
Olark Chat Conversation          183
Email Link Clicked               173
Email Bounced                     60
Unsubscribed                      47
Unreachable                       32
Had a Phone Conversation          14
Email Marked Spam                  2
Approached upfront                 1
Email Received                     1
Form Submitted on Website          1
Resubscribed to emails             1
View in browser link Clicked       1
Name: Last Notable Activity, dtype: int64
```

The following three columns now have the level 'Select'. Let's check them once again.

In [16]:
```python
leads['Lead Profile'].astype('category').value_counts()
```

Out[16]:
```
Select                         4146
Potential Lead                 1613
Other Leads                     487
Student of SomeSchool           241
Lateral Student                  24
Dual Specialization Student      20
Name: Lead Profile, dtype: int64
```

In [17]:
```python
leads['How did you hear about X Education'].value_counts()
```

Out[17]:
```
Select                   5043
Online Search             808
Word Of Mouth             348
Student of SomeSchool     310
Other                     186
Multiple Sources          152
Advertisements             70
Social Media               67
Email                      26
SMS                        23
Name: How did you hear about X Education, dtype: int64
```

In [18]:
```python
leads['Specialization'].value_counts()
```

Out[18]:
```
Select                                 1942
Finance Management                      976
Human Resource Management               848
Marketing Management                    838
Operations Management                   503
Business Administration                 403
IT Projects Management                  366
Supply Chain Management                 349
Banking, Investment And Insurance       338
Travel and Tourism                      203
Media and Advertising                   203
International Business                  178
Healthcare Management                   159
Hospitality Management                  114
E-COMMERCE                              112
Retail Management                       100
Rural and Agribusiness                   73
E-Business                               57
Services Excellence                      40
Name: Specialization, dtype: int64
```

Clearly the levels Lead Profile and How did you hear about X Education have a lot of rows which have the value Select which is of no use to the analysis so it's best that we drop them.

In [19]:
```
leads.drop(['Lead Profile', 'How did you hear about X Education'], axis = 1, inplace = T
```

Also notice that when you got the value counts of all the columns, there were a few columns in which only one value was majorly present for all the data points. These include Do Not Call, Search, Magazine, Newspaper Article, X Education Forums, Newspaper, Digital Advertisement, Through Recommendations, Receive More Updates About Our Courses, Update me on Supply Chain Content, Get updates on DM Content, I agree to pay the amount through cheque. Since practically all of the values for these variables are No, it's best that we drop these columns as they won't help with our analysis.

In [20]:
```
leads.drop(['Do Not Call', 'Search', 'Magazine', 'Newspaper Article', 'X Education Forum
            'Digital Advertisement', 'Through Recommendations', 'Receive More Updates Ab
            'Update me on Supply Chain Content', 'Get updates on DM Content',
            'I agree to pay the amount through cheque'], axis = 1, inplace = True)
```

Also, the variable What matters most to you in choosing a course has the level Better Career Prospects 6528 times while the other two levels appear once twice and once respectively. So we should drop this column as well.

In [21]:
```
leads['What matters most to you in choosing a course'].value_counts()
```

Out[21]:
```
Better Career Prospects      6528
Flexibility & Convenience       2
Other                           1
Name: What matters most to you in choosing a course, dtype: int64
```

In [22]:
```
# Drop the null value rows present in the variable 'What matters most to you in choosing
leads.drop(['What matters most to you in choosing a course'], axis = 1, inplace=True)
leads.isnull().sum()
```

Out[22]:
```
Prospect ID                                 0
Lead Number                                 0
Lead Origin                                 0
Lead Source                                36
Do Not Email                                0
Converted                                   0
TotalVisits                               137
Total Time Spent on Website                 0
Page Views Per Visit                      137
Last Activity                             103
Specialization                           1438
What is your current occupation          2690
A free copy of Mastering The Interview      0
Last Notable Activity                       0
dtype: int64
```

Now, there's the column What is your current occupation which has a lot of null values. Now you can drop the entire row but since we have already lost so many feature variables, we choose not to drop it as it might turn out to be significant in the analysis. So let's just drop the null rows for the column What is you current occupation.

In [23]:
```
leads = leads[~pd.isnull(leads['What is your current occupation'])]
leads.isnull().sum()
```

Out[23]:
```
Prospect ID                                   0
Lead Number                                   0
Lead Origin                                   0
Lead Source                                  36
Do Not Email                                  0
Converted                                     0
TotalVisits                                 130
Total Time Spent on Website                   0
Page Views Per Visit                        130
Last Activity                               103
Specialization                               18
What is your current occupation               0
A free copy of Mastering The Interview        0
Last Notable Activity                         0
dtype: int64
```

In [24]:
```
# Drop the null value rows in the column 'TotalVisits'
leads = leads[~pd.isnull(leads['TotalVisits'])]
leads.isnull().sum()
```

Out[24]:
```
Prospect ID                                  0
Lead Number                                  0
Lead Origin                                  0
Lead Source                                 29
Do Not Email                                 0
Converted                                    0
TotalVisits                                  0
Total Time Spent on Website                  0
Page Views Per Visit                         0
Last Activity                                0
Specialization                              18
What is your current occupation              0
A free copy of Mastering The Interview       0
Last Notable Activity                        0
dtype: int64
```

In [25]:
```
# Drop the null values rows in the column 'Lead Source'
leads = leads[~pd.isnull(leads['Lead Source'])]
leads.isnull().sum()
```

Out[25]:
```
Prospect ID                                  0
Lead Number                                  0
Lead Origin                                  0
Lead Source                                  0
Do Not Email                                 0
Converted                                    0
TotalVisits                                  0
Total Time Spent on Website                  0
Page Views Per Visit                         0
Last Activity                                0
Specialization                              18
What is your current occupation              0
A free copy of Mastering The Interview       0
Last Notable Activity                        0
dtype: int64
```

In [26]:
```
# Drop the null values rows in the column 'Specialization'
leads = leads[~pd.isnull(leads['Specialization'])]
leads.isnull().sum()
```

Out[26]:
```
Prospect ID                                   0
Lead Number                                   0
Lead Origin                                   0
Lead Source                                   0
Do Not Email                                  0
Converted                                     0
TotalVisits                                   0
Total Time Spent on Website                   0
Page Views Per Visit                          0
Last Activity                                 0
Specialization                                0
What is your current occupation               0
A free copy of Mastering The Interview        0
Last Notable Activity                         0
dtype: int64
```

Now your data doesn't have any null values. Let's now check the percentage of rows that we have retained.

In [27]:
```
print(len(leads.index))
print(len(leads.index)/9240)
```

```
6373
0.6897186147186147
```

We still have around 69% of the rows which seems good enough.

In [28]:
```
leads.head()
```

Out[28]:

	Prospect ID	Lead Number	Lead Origin	Lead Source	Do Not Email	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Last Activity	Spe
0	7927b2df-8bba-4d29-b9a2-b6e0beafe620	660737	API	Olark Chat	No	0	0.0	0	0.0	Page Visited on Website	
1	2a272436-5132-4136-86fa-dcc88c88f482	660728	API	Organic Search	No	0	5.0	674	2.5	Email Opened	
2	8cc8c611-a219-4f35-ad23-fdfd2656bd8a	660727	Landing Page Submission	Direct Traffic	No	1	2.0	1532	2.0	Email Opened	Adı
3	0cc2df48-7cf4-4e39-9de9-19797f9b38cc	660719	Landing Page Submission	Direct Traffic	No	0	1.0	305	1.0	Unreachable	
4	3256f628-e534-4826-9d63-4a8b88782852	660681	Landing Page Submission	Google	No	1	2.0	1428	1.0	Converted to Lead	

Now, clearly the variables Prospect ID and Lead Number won't be of any use in the analysis, so it's best that we drop these two variables.

In [29]:
```
leads.drop(['Prospect ID', 'Lead Number'], 1, inplace = True)
leads.head()
```

Out[29]:

	Lead Origin	Lead Source	Do Not Email	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Last Activity	Specialization	What is your current occupation
0	API	Olark Chat	No	0	0.0	0	0.0	Page Visited on Website	Select	Unemployed
1	API	Organic Search	No	0	5.0	674	2.5	Email Opened	Select	Unemployed
2	Landing Page Submission	Direct Traffic	No	1	2.0	1532	2.0	Email Opened	Business Administration	Student
3	Landing Page Submission	Direct Traffic	No	0	1.0	305	1.0	Unreachable	Media and Advertising	Unemployed
4	Landing Page Submission	Google	No	1	2.0	1428	1.0	Converted to Lead	Select	Unemployed

Dummy variable creation The next step is to deal with the categorical variables present in the dataset. So first take a look at which variables are actually categorical variables.

In [30]:

```
# Check the columns which are of type 'object'

temp = leads.loc[:, leads.dtypes == 'object']
temp.columns
```

Out[30]:

```
Index(['Lead Origin', 'Lead Source', 'Do Not Email', 'Last Activity',
       'Specialization', 'What is your current occupation',
       'A free copy of Mastering The Interview', 'Last Notable Activity'],
      dtype='object')
```

In [31]:

```
# Create dummy variables using the 'get_dummies' command
dummy = pd.get_dummies(leads[['Lead Origin', 'Lead Source', 'Do Not Email', 'Last Activi
                              'What is your current occupation','A free copy of Masterin
                              'Last Notable Activity']], drop_first=True)

# Add the results to the master dataframe
leads = pd.concat([leads, dummy], axis=1)
```

In [32]:

```
# Creating dummy variable separately for the variable 'Specialization' since it has the
# drop that level by specifying it explicitly

dummy_spl = pd.get_dummies(leads['Specialization'], prefix = 'Specialization')
dummy_spl = dummy_spl.drop(['Specialization_Select'], 1)
leads = pd.concat([leads, dummy_spl], axis = 1)
```

In [33]:

```
# Drop the variables for which the dummy variables have been created

leads = leads.drop(['Lead Origin', 'Lead Source', 'Do Not Email', 'Last Activity',
                   'Specialization', 'What is your current occupation',
                   'A free copy of Mastering The Interview', 'Last Notable Activity'], 1
```

In [34]:

```
leads.head()
```

Out[34]:

	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Lead Origin_Landing Page Submission	Lead Origin_Lead Add Form	Lead Origin_Lead Import	Lead Source_Direct Traffic	Source_Fac
0	0	0.0	0	0.0	0	0	0	0	
1	0	5.0	674	2.5	0	0	0	0	
2	1	2.0	1532	2.0	1	0	0	1	
3	0	1.0	305	1.0	1	0	0	1	
4	1	2.0	1428	1.0	1	0	0	0	

5 rows × 75 columns

Test-Train Split The next step is to split the dataset into training an testing sets.

In [35]:

```
# Put all the feature variables in X

X = leads.drop(['Converted'], 1)
X.head()
```

Out[35]:

	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Lead Origin_Landing Page Submission	Lead Origin_Lead Add Form	Lead Origin_Lead Import	Lead Source_Direct Traffic	Lead Source_Facebook	Sou
0	0.0	0	0.0	0	0	0	0	0	
1	5.0	674	2.5	0	0	0	0	0	
2	2.0	1532	2.0	1	0	0	1	0	
3	1.0	305	1.0	1	0	0	1	0	
4	2.0	1428	1.0	1	0	0	0	0	

5 rows × 74 columns

In [36]:

```
# Put the target variable in y

y = leads['Converted']
y.head()
```

Out[36]:

```
0    0
1    0
2    1
3    0
4    1
Name: Converted, dtype: int64
```

In [37]:

```
# Split the dataset into 70% train and 30% test

X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.7, test_size=0.3,
```

--**Scaling**-- Now there are a few numeric variables present in the dataset which have different scales. So let's go ahead and scale these variables.

In [38]:

```
# Scale the three numeric features present in the dataset

scaler = MinMaxScaler()
```

Loading [MathJax]/extensions/Safe.js

```
X_train[['TotalVisits', 'Page Views Per Visit', 'Total Time Spent on Website']] = scaler
X_train.head()
```

Out[38]:

	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Lead Origin_Landing Page Submission	Lead Origin_Lead Add Form	Lead Origin_Lead Import	Lead Source_Direct Traffic	Lead Source_Facebook
8003	0.015936	0.029489	0.125	1	0	0	1	0
218	0.015936	0.082306	0.250	1	0	0	1	0
4171	0.023904	0.034331	0.375	1	0	0	1	0
4037	0.000000	0.000000	0.000	0	0	0	0	0
3660	0.000000	0.000000	0.000	0	1	0	0	0

5 rows × 74 columns

Looking at the correlations Let's now look at the correlations. Since the number of variables are pretty high, it's better that we look at the table instead of plotting a heatmap

In [39]:

```
# Looking at the correlation table
leads.corr()
```

Out[39]:

	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Lead Origin_Landing Page Submission	Lead Origin_Lead Add Form	Lead Origin_Lead Import	S
Converted	1.000000	0.005651	0.313338	-0.063362	-0.117563	0.288666	-0.019269	
TotalVisits	0.005651	1.000000	0.202551	0.489039	0.267954	-0.208375	-0.043000	
Total Time Spent on Website	0.313338	0.202551	1.000000	0.303870	0.275606	-0.249493	-0.061429	
Page Views Per Visit	-0.063362	0.489039	0.303870	1.000000	0.458168	-0.340185	-0.065739	
Lead Origin_Landing Page Submission	-0.117563	0.267954	0.275606	0.458168	1.000000	-0.363764	-0.074917	
...	...	...	...	...	...	...	...	
Specialization_Retail Management	-0.018603	0.014223	0.024919	0.026099	0.070983	-0.025339	-0.007261	
Specialization_Rural and Agribusiness	0.006964	0.068015	0.018767	0.027465	0.050077	-0.018872	-0.006251	
Specialization_Services Excellence	-0.005142	0.015114	0.003203	0.015230	0.039433	-0.011155	-0.004093	
Specialization_Supply Chain Management	0.005785	0.063383	0.045386	0.052972	0.111610	-0.035065	-0.001963	
Specialization_Travel and Tourism	-0.011762	0.064384	0.037867	0.111284	0.094875	-0.045397	-0.010092	

75 rows × 75 columns

Step 2: Model Building

Let's now move to model building. As you can see that there are a lot of variables present in the dataset which we cannot deal with. So the best way to approach this is to select a small set of features from this pool of variables using RFE.

In [40]:
```
logreg = LogisticRegression()
rfe = RFE(estimator=logreg, n_features_to_select=15)
rfe.fit(X_train, y_train)
```

Out[40]:
```
▸ RFE
▸ estimator: LogisticRegression
    ▸ LogisticRegression
```

In [41]:
```
# Let's take a look at which features have been selected by RFE

list(zip(X_train.columns, rfe.support_, rfe.ranking_))
```

Out[41]:
```
[('TotalVisits', True, 1),
 ('Total Time Spent on Website', True, 1),
 ('Page Views Per Visit', False, 23),
 ('Lead Origin_Landing Page Submission', False, 8),
 ('Lead Origin_Lead Add Form', True, 1),
 ('Lead Origin_Lead Import', False, 52),
 ('Lead Source_Direct Traffic', False, 24),
 ('Lead Source_Facebook', False, 51),
 ('Lead Source_Google', False, 36),
 ('Lead Source_Live Chat', False, 44),
 ('Lead Source_Olark Chat', True, 1),
 ('Lead Source_Organic Search', False, 35),
 ('Lead Source_Pay per Click Ads', False, 43),
 ('Lead Source_Press_Release', False, 53),
 ('Lead Source_Reference', True, 1),
 ('Lead Source_Referral Sites', False, 37),
 ('Lead Source_Social Media', False, 58),
 ('Lead Source_WeLearn', False, 42),
 ('Lead Source_Welingak Website', True, 1),
 ('Lead Source_bing', False, 33),
 ('Lead Source_testone', False, 38),
 ('Do Not Email_Yes', True, 1),
 ('Last Activity_Converted to Lead', False, 25),
 ('Last Activity_Email Bounced', False, 4),
 ('Last Activity_Email Link Clicked', False, 49),
 ('Last Activity_Email Marked Spam', False, 57),
 ('Last Activity_Email Opened', False, 41),
 ('Last Activity_Email Received', False, 54),
 ('Last Activity_Form Submitted on Website', False, 28),
 ('Last Activity_Had a Phone Conversation', True, 1),
 ('Last Activity_Olark Chat Conversation', False, 5),
 ('Last Activity_Page Visited on Website', False, 26),
 ('Last Activity_SMS Sent', True, 1),
 ('Last Activity_Unreachable', False, 47),
 ('Last Activity_Unsubscribed', False, 40),
 ('Last Activity_View in browser link Clicked', False, 34),
 ('Last Activity_Visited Booth in Tradeshow', False, 48),
 ('What is your current occupation_Housewife', True, 1),
 ('What is your current occupation_Other', False, 46),
 ('What is your current occupation_Student', True, 1),
 ('What is your current occupation_Unemployed', True, 1),
 ('What is your current occupation_Working Professional', True, 1),
 ('A free copy of Mastering The Interview_Yes', False, 50),
 ('Last Notable Activity_Email Bounced', False, 3),
 ('Last Notable Activity_Email Link Clicked', False, 20),
 ('Last Notable Activity_Email Marked Spam', False, 59),
 ('Last Notable Activity_Email Opened', False, 27),
 ('Last Notable Activity_Email Received', False, 60),
 ('Last Notable Activity_Had a Phone Conversation', True, 1),
 ('Last Notable Activity_Modified', False, 2),
 ('Last Notable Activity_Olark Chat Conversation', False, 32),
 ('Last Notable Activity_Page Visited on Website', False, 31),
 ('Last Notable Activity_SMS Sent', False, 45),
 ('Last Notable Activity_Unreachable', True, 1),
 ('Last Notable Activity_Unsubscribed', False, 39),
 ('Last Notable Activity_View in browser link Clicked', False, 29),
 ('Specialization_Banking, Investment And Insurance', False, 6),
 ('Specialization_Business Administration', False, 15),
 ('Specialization_E-Business', False, 11),
 ('Specialization_E-COMMERCE', False, 9),
 ('Specialization_Finance Management', False, 14),
 ('Specialization_Healthcare Management', False, 10),
 ('Specialization_Hospitality Management', False, 55),
 ('Specialization_Human Resource Management', False, 16),
```

```
 ('Specialization_IT Projects Management', False, 18),
 ('Specialization_International Business', False, 22),
 ('Specialization_Marketing Management', False, 12),
 ('Specialization_Media and Advertising', False, 21),
 ('Specialization_Operations Management', False, 19),
 ('Specialization_Retail Management', False, 30),
 ('Specialization_Rural and Agribusiness', False, 7),
 ('Specialization_Services Excellence', False, 56),
 ('Specialization_Supply Chain Management', False, 13),
 ('Specialization_Travel and Tourism', False, 17)]
```

In [42]:

```
# Put all the columns selected by RFE in the variable 'col'
col = X_train.columns[rfe.support_]
```

Now you have all the variables selected by RFE and since we care about the statistics part, i.e. the p-values and the VIFs, let's use these variables to create a logistic regression model using statsmodels.

In [43]:

```
# Select only the columns selected by RFE
X_train = X_train[col]
```

In [44]:

```
# Fit a logistic Regression model on X_train after adding a constant and output the summ

X_train_sm = sm.add_constant(X_train)
logm2 = sm.GLM(y_train, X_train_sm, family = sm.families.Binomial())
res = logm2.fit()
res.summary()
```

Out[44]: Generalized Linear Model Regression Results

Dep. Variable:	Converted	**No. Observations:**	4461
Model:	GLM	**Df Residuals:**	4445
Model Family:	Binomial	**Df Model:**	15
Link Function:	Logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-2072.8
Date:	Wed, 06 Mar 2024	**Deviance:**	4145.5
Time:	12:01:56	**Pearson chi2:**	4.84e+03
No. Iterations:	22	**Pseudo R-squ. (CS):**	0.3660
Covariance Type:	nonrobust		

	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
const	-1.0061	0.600	-1.677	0.094	-2.182	0.170
TotalVisits	11.3439	2.682	4.230	0.000	6.088	16.600
Total Time Spent on Website	4.4312	0.185	23.924	0.000	4.068	4.794
Lead Origin_Lead Add Form	2.9483	1.191	2.475	0.013	0.614	5.283
Lead Source_Olark Chat	1.4584	0.122	11.962	0.000	1.219	1.697
Lead Source_Reference	1.2994	1.214	1.070	0.285	-1.080	3.679
Lead Source_Welingak Website	3.4159	1.558	2.192	0.028	0.362	6.470
Do Not Email_Yes	-1.5053	0.193	-7.781	0.000	-1.884	-1.126
Last Activity_Had a Phone Conversation	1.0397	0.983	1.058	0.290	-0.887	2.966
Last Activity_SMS Sent	1.1827	0.082	14.362	0.000	1.021	1.344
What is your current occupation_Housewife	22.6492	2.45e+04	0.001	0.999	-4.8e+04	4.8e+04
What is your current occupation_Student	-1.1544	0.630	-1.831	0.067	-2.390	0.081
What is your current occupation_Unemployed	-1.3395	0.594	-2.254	0.024	-2.505	-0.175
What is your current occupation_Working Professional	1.2743	0.623	2.045	0.041	0.053	2.496
Last Notable Activity_Had a Phone Conversation	23.1932	2.08e+04	0.001	0.999	-4.08e+04	4.08e+04
Last Notable Activity_Unreachable	2.7868	0.807	3.453	0.001	1.205	4.369

There are quite a few variable which have a p-value greater than 0.05. We will need to take care of them. But first, let's also look at the VIFs.

In [45]:
```
# Import 'variance_inflation_factor'
from statsmodels.stats.outliers_influence import variance_inflation_factor
```

In [46]:
```
# Make a VIF dataframe for all the variables present

vif = pd.DataFrame()
vif['Features'] = X_train.columns
vif['VIF'] = [variance_inflation_factor(X_train.values, i) for i in range(X_train.shape[
vif['VIF'] = round(vif['VIF'], 2)
vif = vif.sort_values(by = "VIF", ascending = False)
vif
```

Out[46]:

	Features	VIF
2	Lead Origin_Lead Add Form	84.19
4	Lead Source_Reference	65.18
5	Lead Source_Welingak Website	20.03
11	What is your current occupation_Unemployed	3.65
7	Last Activity_Had a Phone Conversation	2.44
13	Last Notable Activity_Had a Phone Conversation	2.43
1	Total Time Spent on Website	2.38
0	TotalVisits	1.62
8	Last Activity_SMS Sent	1.59
12	What is your current occupation_Working Profes...	1.56
3	Lead Source_Olark Chat	1.44
6	Do Not Email_Yes	1.09
10	What is your current occupation_Student	1.09
9	What is your current occupation_Housewife	1.01
14	Last Notable Activity_Unreachable	1.01

VIFs seem to be in a decent range except for three variables.

Let's first drop the variable Lead Source_Reference since it has a high p-value as well as a high VIF.

In [47]:

```
# Refit the model with the new set of features

logm1 = sm.GLM(y_train,(sm.add_constant(X_train)), family = sm.families.Binomial())
logm1.fit().summary()
```

Out[47]:

Generalized Linear Model Regression Results

Dep. Variable:	Converted	**No. Observations:**	4461
Model:	GLM	**Df Residuals:**	4445
Model Family:	Binomial	**Df Model:**	15
Link Function:	Logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-2072.8
Date:	Wed, 06 Mar 2024	**Deviance:**	4145.5
Time:	12:01:57	**Pearson chi2:**	4.84e+03
No. Iterations:	22	**Pseudo R-squ. (CS):**	0.3660
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-1.0061	0.600	-1.677	0.094	-2.182	0.170
TotalVisits	11.3439	2.682	4.230	0.000	6.088	16.600
Total Time Spent on Website	4.4312	0.185	23.924	0.000	4.068	4.794
Lead Origin_Lead Add Form	2.9483	1.191	2.475	0.013	0.614	5.283
Lead Source_Olark Chat	1.4584	0.122	11.962	0.000	1.219	1.697
Lead Source_Reference	1.2994	1.214	1.070	0.285	-1.080	3.679
Lead Source_Welingak Website	3.4159	1.558	2.192	0.028	0.362	6.470
Do Not Email_Yes	-1.5053	0.193	-7.781	0.000	-1.884	-1.126
Last Activity_Had a Phone Conversation	1.0397	0.983	1.058	0.290	-0.887	2.966
Last Activity_SMS Sent	1.1827	0.082	14.362	0.000	1.021	1.344
What is your current occupation_Housewife	22.6492	2.45e+04	0.001	0.999	-4.8e+04	4.8e+04
What is your current occupation_Student	-1.1544	0.630	-1.831	0.067	-2.390	0.081
What is your current occupation_Unemployed	-1.3395	0.594	-2.254	0.024	-2.505	-0.175
What is your current occupation_Working Professional	1.2743	0.623	2.045	0.041	0.053	2.496
Last Notable Activity_Had a Phone Conversation	23.1932	2.08e+04	0.001	0.999	-4.08e+04	4.08e+04
Last Notable Activity_Unreachable	2.7868	0.807	3.453	0.001	1.205	4.369

The variable Lead Profile_Dual Specialization Student also needs to be dropped.

In [48]:

```
# Make a VIF dataframe for all the variables present

vif = pd.DataFrame()
vif['Features'] = X_train.columns
vif['VIF'] = [variance_inflation_factor(X_train.values, i) for i in range(X_train.shape[
vif['VIF'] = round(vif['VIF'], 2)
vif = vif.sort_values(by = "VIF", ascending = False)
vif
```

Out[48]:

	Features	VIF
2	Lead Origin_Lead Add Form	84.19
4	Lead Source_Reference	65.18
5	Lead Source_Welingak Website	20.03
11	What is your current occupation_Unemployed	3.65
7	Last Activity_Had a Phone Conversation	2.44
13	Last Notable Activity_Had a Phone Conversation	2.43
1	Total Time Spent on Website	2.38
0	TotalVisits	1.62
8	Last Activity_SMS Sent	1.59
12	What is your current occupation_Working Profes...	1.56
3	Lead Source_Olark Chat	1.44
6	Do Not Email_Yes	1.09
10	What is your current occupation_Student	1.09
9	What is your current occupation_Housewife	1.01
14	Last Notable Activity_Unreachable	1.01

The VIFs are now all less than 5. So let's drop the ones with the high p-values beginning with Last Notable Activity_Had a Phone Conversation.

In [49]:
```
X_train.drop('Last Notable Activity_Had a Phone Conversation', axis = 1, inplace = True)
```

In [50]:
```
# Refit the model with the new set of features

logm1 = sm.GLM(y_train,(sm.add_constant(X_train)), family = sm.families.Binomial())
logm1.fit().summary()
```

Out[50]:

Generalized Linear Model Regression Results

Dep. Variable:	Converted	**No. Observations:**	4461
Model:	GLM	**Df Residuals:**	4446
Model Family:	Binomial	**Df Model:**	14
Link Function:	Logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-2075.6
Date:	Wed, 06 Mar 2024	**Deviance:**	4151.3
Time:	12:01:57	**Pearson chi2:**	4.85e+03
No. Iterations:	21	**Pseudo R-squ. (CS):**	0.3652
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-1.0074	0.600	-1.680	0.093	-2.183	0.168
TotalVisits	11.4562	2.686	4.265	0.000	6.192	16.720
Total Time Spent on Website	4.4237	0.185	23.900	0.000	4.061	4.787
Lead Origin_Lead Add Form	2.9481	1.191	2.475	0.013	0.613	5.283
Lead Source_Olark Chat	1.4582	0.122	11.959	0.000	1.219	1.697
Lead Source_Reference	1.2995	1.214	1.070	0.285	-1.080	3.679
Lead Source_Welingak Website	3.4159	1.558	2.192	0.028	0.362	6.470
Do Not Email_Yes	-1.5054	0.193	-7.782	0.000	-1.885	-1.126
Last Activity_Had a Phone Conversation	2.7501	0.802	3.430	0.001	1.179	4.322
Last Activity_SMS Sent	1.1826	0.082	14.364	0.000	1.021	1.344
What is your current occupation_Housewife	21.6506	1.48e+04	0.001	0.999	-2.91e+04	2.91e+04
What is your current occupation_Student	-1.1528	0.630	-1.829	0.067	-2.388	0.083
What is your current occupation_Unemployed	-1.3379	0.594	-2.252	0.024	-2.503	-0.173
What is your current occupation_Working Professional	1.2738	0.623	2.044	0.041	0.053	2.495
Last Notable Activity_Unreachable	2.7858	0.807	3.452	0.001	1.204	4.367

In [51]:

```
#Drop What is your current occupation_Housewife.
X_train.drop('What is your current occupation_Housewife', axis = 1, inplace = True)
```

In [52]:

```
# Refit the model with the new set of features

logm1 = sm.GLM(y_train,(sm.add_constant(X_train)), family = sm.families.Binomial())
logm1.fit().summary()
```

Out[52]:

Generalized Linear Model Regression Results

Dep. Variable:	Converted	**No. Observations:**	4461
Model:	GLM	**Df Residuals:**	4447
Model Family:	Binomial	**Df Model:**	13
Link Function:	Logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-2077.9
Date:	Wed, 06 Mar 2024	**Deviance:**	4155.7
Time:	12:01:57	**Pearson chi2:**	4.85e+03
No. Iterations:	7	**Pseudo R-squ. (CS):**	0.3645
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-0.4535	0.554	-0.819	0.413	-1.539	0.632
TotalVisits	11.2598	2.671	4.215	0.000	6.024	16.495
Total Time Spent on Website	4.4217	0.185	23.899	0.000	4.059	4.784
Lead Origin_Lead Add Form	2.9436	1.191	2.471	0.013	0.609	5.278
Lead Source_Olark Chat	1.4531	0.122	11.931	0.000	1.214	1.692
Lead Source_Reference	1.3014	1.214	1.072	0.284	-1.078	3.681
Lead Source_Welingak Website	3.4163	1.558	2.193	0.028	0.363	6.470
Do Not Email_Yes	-1.5080	0.194	-7.787	0.000	-1.888	-1.128
Last Activity_Had a Phone Conversation	2.7513	0.802	3.432	0.001	1.180	4.323
Last Activity_SMS Sent	1.1823	0.082	14.362	0.000	1.021	1.344
What is your current occupation_Student	-1.7021	0.588	-2.893	0.004	-2.855	-0.549
What is your current occupation_Unemployed	-1.8871	0.550	-3.433	0.001	-2.964	-0.810
What is your current occupation_Working Professional	0.7244	0.581	1.248	0.212	-0.414	1.862
Last Notable Activity_Unreachable	2.7830	0.807	3.447	0.001	1.201	4.365

In [53]:

```
#Drop What is your current occupation_Working Professional.
X_train.drop('What is your current occupation_Working Professional', axis = 1, inplace =
```

In [54]:

```
# Refit the model with the new set of features

logm1 = sm.GLM(y_train,(sm.add_constant(X_train)), family = sm.families.Binomial())
res = logm1.fit()
res.summary()
```

Loading [MathJax]/extensions/Safe.js

Out[54]:

Generalized Linear Model Regression Results

Dep. Variable:	Converted	**No. Observations:**	4461
Model:	GLM	**Df Residuals:**	4448
Model Family:	Binomial	**Df Model:**	12
Link Function:	Logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-2078.6
Date:	Wed, 06 Mar 2024	**Deviance:**	4157.2
Time:	12:01:57	**Pearson chi2:**	4.82e+03
No. Iterations:	7	**Pseudo R-squ. (CS):**	0.3643
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.2031	0.196	1.038	0.299	-0.180	0.587
TotalVisits	11.1501	2.664	4.185	0.000	5.928	16.372
Total Time Spent on Website	4.4223	0.185	23.899	0.000	4.060	4.785
Lead Origin_Lead Add Form	2.9421	1.191	2.470	0.014	0.607	5.277
Lead Source_Olark Chat	1.4526	0.122	11.935	0.000	1.214	1.691
Lead Source_Reference	1.3024	1.214	1.073	0.283	-1.078	3.682
Lead Source_Welingak Website	3.4157	1.558	2.192	0.028	0.362	6.470
Do Not Email_Yes	-1.5054	0.194	-7.777	0.000	-1.885	-1.126
Last Activity_Had a Phone Conversation	2.7551	0.802	3.437	0.001	1.184	4.326
Last Activity_SMS Sent	1.1856	0.082	14.421	0.000	1.024	1.347
What is your current occupation_Student	-2.3581	0.281	-8.390	0.000	-2.909	-1.807
What is your current occupation_Unemployed	-2.5434	0.186	-13.691	0.000	-2.908	-2.179
Last Notable Activity_Unreachable	2.7842	0.807	3.449	0.001	1.202	4.366

All the p-values are now in the appropriate range. Let's also check the VIFs again in case we had missed something.

In [55]:

```
# Make a VIF dataframe for all the variables present

vif = pd.DataFrame()
vif['Features'] = X_train.columns
vif['VIF'] = [variance_inflation_factor(X_train.values, i) for i in range(X_train.shape[
vif['VIF'] = round(vif['VIF'], 2)
vif = vif.sort_values(by = "VIF", ascending = False)
vif
```

Out[55]:

	Features	VIF
2	Lead Origin_Lead Add Form	84.18
4	Lead Source_Reference	65.06
5	Lead Source_Welingak Website	20.02
10	What is your current occupation_Unemployed	2.82
1	Total Time Spent on Website	2.00
0	TotalVisits	1.54
8	Last Activity_SMS Sent	1.51
3	Lead Source_Olark Chat	1.33
6	Do Not Email_Yes	1.08
9	What is your current occupation_Student	1.06
7	Last Activity_Had a Phone Conversation	1.01
11	Last Notable Activity_Unreachable	1.01

Step 3: Model Evaluation

Now, both the p-values and VIFs seem decent enough for all the variables. So let's go ahead and make predictions using this final set of features.

In [56]:

```
# Use 'predict' to predict the probabilities on the train set

y_train_pred = res.predict(sm.add_constant(X_train))
y_train_pred[:10]
```

Out[56]:

```
8003    0.300151
218     0.142019
4171    0.127646
4037    0.291594
3660    0.956470
207     0.194449
2044    0.178093
6411    0.949418
6498    0.075790
2085    0.982321
dtype: float64
```

In [57]:

```
# Reshaping it into an array

y_train_pred = y_train_pred.values.reshape(-1)
y_train_pred[:10]
```

Out[57]:

```
array([0.30015138, 0.14201937, 0.12764605, 0.29159395, 0.95647007,
       0.19444915, 0.17809324, 0.94941847, 0.07578958, 0.98232052])
```

Creating a dataframe with the actual conversion flag and the predicted probabilities

In [58]:

```
# Create a new dataframe containing the actual conversion flag and the probabilities pre

y_train_pred_final = pd.DataFrame({'Converted':y_train.values, 'Conversion_Prob':y_train
y_train_pred_final.head()
```

Out[58]:

	Converted	Conversion_Prob
0	0	0.300151
1	0	0.142019
2	1	0.127646
3	1	0.291594
4	1	0.956470

Creating new column 'Predicted' with 1 if Paid_Prob > 0.5 else 0

In [59]:
```
y_train_pred_final['Predicted'] = y_train_pred_final.Conversion_Prob.map(lambda x: 1 if

# Let's see the head
y_train_pred_final.head()
```

Out[59]:

	Converted	Conversion_Prob	Predicted
0	0	0.300151	0
1	0	0.142019	0
2	1	0.127646	0
3	1	0.291594	0
4	1	0.956470	1

Now that you have the probabilities and have also made conversion predictions using them, it's time to evaluate the model.

In [60]:
```
# Create confusion matrix

confusion = metrics.confusion_matrix(y_train_pred_final.Converted, y_train_pred_final.Pr
print(confusion)
```

```
[[1929  383]
 [ 560 1589]]
```

In [61]:
```
# Predicted     not_churn    churn
# Actual
# not_churn        2543      463
# churn            692       1652
```

In [62]:
```
# Let's check the overall accuracy

print(metrics.accuracy_score(y_train_pred_final.Converted, y_train_pred_final.Predicted)
```

```
0.7886124187401928
```

In [63]:
```
# Let's evaluate the other metrics as well

TP = confusion[1,1] # true positive
TN = confusion[0,0] # true negatives
FP = confusion[0,1] # false positives
FN = confusion[1,0] # false negatives
```

In [64]:
```
# Calculate the sensitivity

TP/(TP+FN)
```

Loading [MathJax]/extensions/Safe.js

Out[64]: 0.739413680781759

In [65]:
```
# Calculate the specificity

TN/(TN+FP)
```

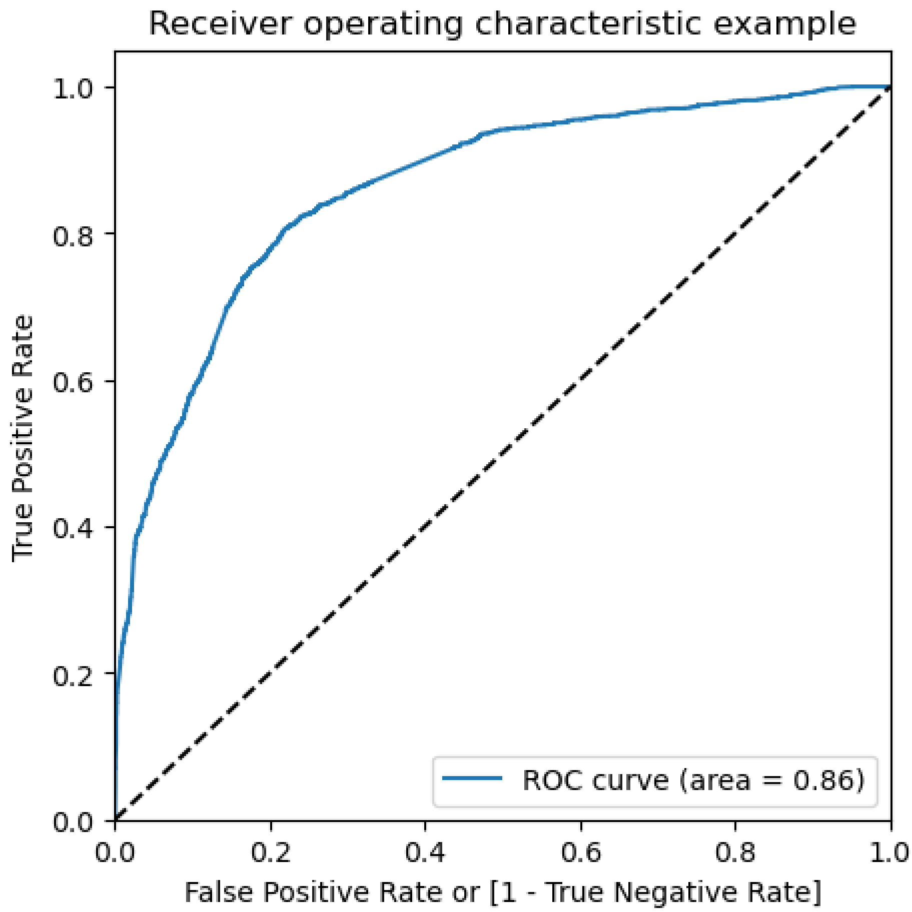
Out[65]: 0.8343425605536332

Finding the Optimal Cutoff Now 0.5 was just arbitrary to loosely check the model performace. But in order to get good results, you need to optimise the threshold. So first let's plot an ROC curve to see what AUC we get.

In [66]:
```
# ROC function

def draw_roc( actual, probs ):
    fpr, tpr, thresholds = metrics.roc_curve( actual, probs,
                                              drop_intermediate = False )
    auc_score = metrics.roc_auc_score( actual, probs )
    plt.figure(figsize=(5, 5))
    plt.plot( fpr, tpr, label='ROC curve (area = %0.2f)' % auc_score )
    plt.plot([0, 1], [0, 1], 'k--')
    plt.xlim([0.0, 1.0])
    plt.ylim([0.0, 1.05])
    plt.xlabel('False Positive Rate or [1 - True Negative Rate]')
    plt.ylabel('True Positive Rate')
    plt.title('Receiver operating characteristic example')
    plt.legend(loc="lower right")
    plt.show()

    return None
```

In [67]:
```
fpr, tpr, thresholds = metrics.roc_curve( y_train_pred_final.Converted, y_train_pred_fin
```

In [68]:
```
# Call the ROC function

draw_roc(y_train_pred_final.Converted, y_train_pred_final.Conversion_Prob)
```



The area under the curve of the ROC is 0.86 which is quite good. So we seem to have a good model. Let's also check the sensitivity and specificity tradeoff to find the optimal cutoff point.

In [69]:
```
# Let's create columns with different probability cutoffs

numbers = [float(x)/10 for x in range(10)]
for i in numbers:
    y_train_pred_final[i]= y_train_pred_final.Conversion_Prob.map(lambda x: 1 if x > i e
y_train_pred_final.head()
```

Out[69]:

	Converted	Conversion_Prob	Predicted	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0	0	0.300151	0	1	1	1	1	0	0	0	0	0	0
1	0	0.142019	0	1	1	0	0	0	0	0	0	0	0
2	1	0.127646	0	1	1	0	0	0	0	0	0	0	0
3	1	0.291594	0	1	1	1	0	0	0	0	0	0	0
4	1	0.956470	1	1	1	1	1	1	1	1	1	1	1

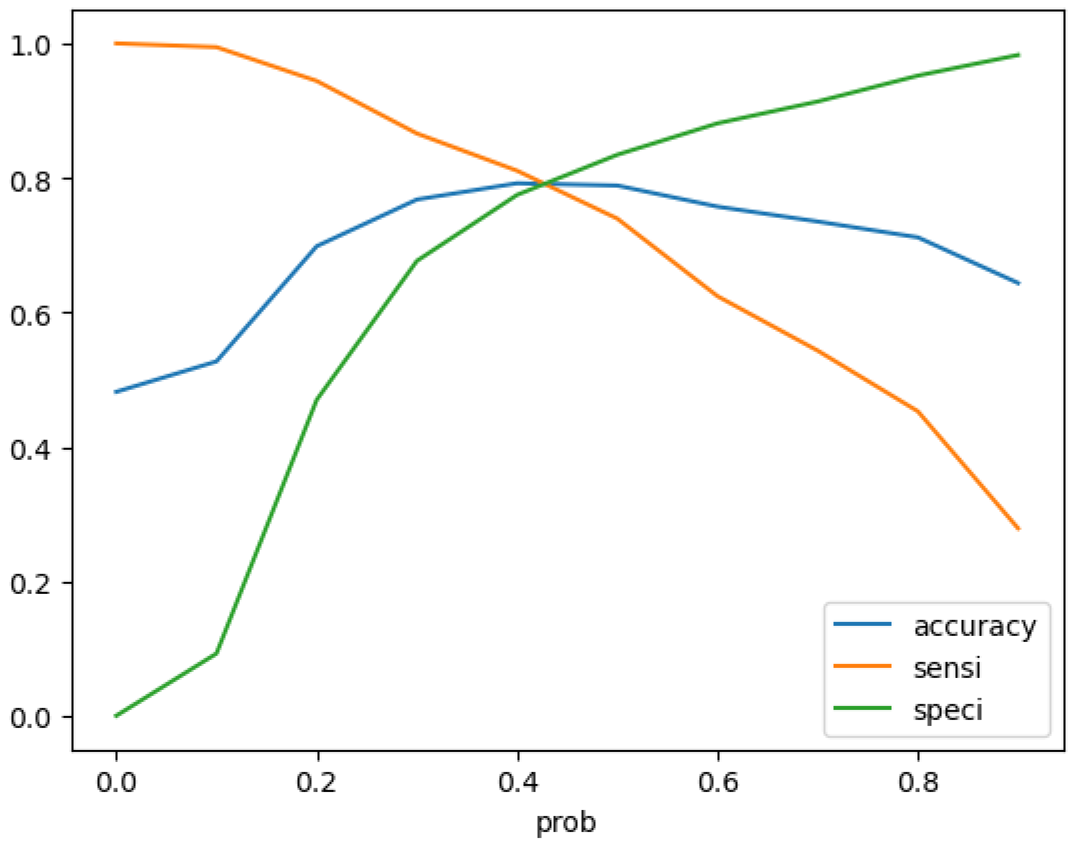
In [70]:
```
# Let's create a dataframe to see the values of accuracy, sensitivity, and specificity a

cutoff_df = pd.DataFrame( columns = ['prob','accuracy','sensi','speci'])
from sklearn.metrics import confusion_matrix

# TP = confusion[1,1] # true positive
# TN = confusion[0,0] # true negatives
# FP = confusion[0,1] # false positives
# FN = confusion[1,0] # false negatives

num = [0.0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9]
for i in num:
    rics.confusion_matrix(y_train_pred_final.Converted, y_train_pred_final[i] )
```

Loading [MathJax]/extensions/Safe.js

```
    total1=sum(sum(cm1))
    accuracy = (cm1[0,0]+cm1[1,1])/total1

    speci = cm1[0,0]/(cm1[0,0]+cm1[0,1])
    sensi = cm1[1,1]/(cm1[1,0]+cm1[1,1])
    cutoff_df.loc[i] =[ i ,accuracy,sensi,speci]
print(cutoff_df)
```

```
     prob  accuracy     sensi     speci
0.0   0.0  0.481731  1.000000  0.000000
0.1   0.1  0.527012  0.994416  0.092561
0.2   0.2  0.698274  0.944160  0.469723
0.3   0.3  0.767765  0.865984  0.676471
0.4   0.4  0.791975  0.810610  0.774654
0.5   0.5  0.788612  0.739414  0.834343
0.6   0.6  0.757229  0.624011  0.881055
0.7   0.7  0.735037  0.543043  0.913495
0.8   0.8  0.711500  0.452769  0.951990
0.9   0.9  0.643578  0.278734  0.982699
```

In [71]:
```
# Let's plot it as well

cutoff_df.plot.line(x='prob', y=['accuracy','sensi','speci'])
plt.show()
```



As you can see that around 0.42, you get the optimal values of the three metrics. So let's choose 0.42 as our cutoff now.

In [72]:
```
y_train_pred_final['final_predicted'] = y_train_pred_final.Conversion_Prob.map( lambda x

y_train_pred_final.head()
```

Out[72]:

	Converted	Conversion_Prob	Predicted	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	final_predicted
0	0	0.300151	0	1	1	1	1	0	0	0	0	0	0	0
1	0	0.142019	0	1	1	0	0	0	0	0	0	0	0	0
2	1	0.127646	0	1	1	0	0	0	0	0	0	0	0	0
3	1	0.291594	0	1	1	1	0	0	0	0	0	0	0	0
4	1	0.956470	1	1	1	1	1	1	1	1	1	1	1	1

In [73]:

```
# Let's check the accuracy now

metrics.accuracy_score(y_train_pred_final.Converted, y_train_pred_final.final_predicted)
```

Out[73]:

```
0.7910782335799148
```

In [74]:

```
# Let's create the confusion matrix once again

confusion2 = metrics.confusion_matrix(y_train_pred_final.Converted, y_train_pred_final.f
confusion2
```

Out[74]:

```
array([[1823,  489],
       [ 443, 1706]], dtype=int64)
```

In [75]:

```
# Let's evaluate the other metrics as well

TP = confusion2[1,1] # true positive
TN = confusion2[0,0] # true negatives
FP = confusion2[0,1] # false positives
FN = confusion2[1,0] # false negatives
```

In [76]:

```
# Calculate Sensitivity

TP/(TP+FN)
```

Out[76]:

```
0.7938576081898557
```

In [77]:

```
# Calculate Specificity

TN/(TN+FP)
```

Out[77]:

```
0.7884948096885813
```

Step 4: Making Predictions on the Test Set

Let's now make predicitons on the test set.

In [78]:

```
# Scale the test set as well using just 'transform'

X_test[['TotalVisits', 'Page Views Per Visit', 'Total Time Spent on Website']] = scaler.
```

In [79]:

```
# Select the columns in X_train for X_test as well

X_test = X_test[col]
X_test.head()
```

Out[79]:

	TotalVisits	Total Time Spent on Website	Lead Origin_Lead Add Form	Lead Source_Olark Chat	Lead Source_Reference	Lead Source_Welingak Website	Do Not Email_Yes	Activity a I Convers
4771	0.000000	0.000000	1	0	1	0	0	
6122	0.027888	0.029049	0	0	0	0	0	
9202	0.015936	0.416813	0	0	0	0	0	
6570	0.011952	0.378961	0	0	0	0	1	
2668	0.031873	0.395246	0	0	0	0	0	

In [80]:

```
# Add a constant to X_test

X_test_sm = sm.add_constant(X_test[col])
X_test_sm
```

Out[80]:

	const	TotalVisits	Total Time Spent on Website	Lead Origin_Lead Add Form	Lead Source_Olark Chat	Lead Source_Reference	Lead Source_Welingak Website	Do Not Email_Yes
4771	1.0	0.000000	0.000000	1	0	1	0	0
6122	1.0	0.027888	0.029049	0	0	0	0	0
9202	1.0	0.015936	0.416813	0	0	0	0	0
6570	1.0	0.011952	0.378961	0	0	0	0	1
2668	1.0	0.031873	0.395246	0	0	0	0	0
...	...	...	...	...	...	...	...	...
5828	1.0	0.011952	0.027289	0	0	0	0	0
6583	1.0	0.011952	0.152289	0	0	0	0	0
5531	1.0	0.055777	0.702025	0	0	0	0	0
3056	1.0	0.011952	0.417694	0	0	0	0	1
4088	1.0	0.019920	0.530370	0	0	0	0	0

1912 rows × 16 columns

In [81]:

```
columns_to_drop = ['Lead Source_Reference', 'What is your current occupation_Housewife',
                   'What is your current occupation_Working Professional', 'Last Notable Activ

# Drop columns only if they exist in X_test
columns_to_drop = [col for col in columns_to_drop if col in X_test.columns]
X_test.drop(columns=columns_to_drop, inplace=True, errors='ignore')
```

In [82]:

```
# Drop 'Lead Source_Reference' from X_test if it exists
X_test_subset = X_test.drop('Lead Source_Reference', axis=1, errors='ignore')

# Align X_test with X_train
X_test_subset = X_test_subset.reindex(columns=X_train.columns)

# Fill NaN values with zeros
X_test_subset.fillna(0, inplace=True)

# Add a constant term for the intercept
                = sm.add_constant(X_test_subset)
```

Loading [MathJax]/extensions/Safe.js

```
# Make predictions on the test set
y_test_pred = res.predict(X_test_const)

# Access the first 10 predictions
y_test_pred[:10]
```

Out[82]:

```
4771    0.987015
6122    0.130010
9202    0.703972
6570    0.299230
2668    0.720833
4233    0.791825
3368    0.704066
9091    0.464563
5972    0.283009
3631    0.786488
dtype: float64
```

In [83]:

```
# Converting y_pred to a dataframe

y_pred_1 = pd.DataFrame(y_test_pred)
```

In [84]:

```
y_pred_1.head()
```

Out[84]:

	0
4771	0.987015
6122	0.130010
9202	0.703972
6570	0.299230
2668	0.720833

In [85]:

```
# Converting y_test to dataframe

y_test_df = pd.DataFrame(y_test)
```

In [86]:

```
# Remove index for both dataframes to append them side by side

y_pred_1.reset_index(drop=True, inplace=True)
y_test_df.reset_index(drop=True, inplace=True)
```

In [87]:

```
# Append y_test_df and y_pred_1

y_pred_final = pd.concat([y_test_df, y_pred_1],axis=1)
```

In [88]:

```
# Check 'y_pred_final'

y_pred_final.head()
```

Out[88]:

	Converted	0
0	1	0.987015
1	0	0.130010
2	0	0.703972
3	1	0.299230
4	1	0.720833

In [89]:
```
# Rename the column

y_pred_final= y_pred_final.rename(columns = {0 : 'Conversion_Prob'})
```

In [90]:
```
# Let's see the head of y_pred_final

y_pred_final.head()
```

Out[90]:

	Converted	Conversion_Prob
0	1	0.987015
1	0	0.130010
2	0	0.703972
3	1	0.299230
4	1	0.720833

In [91]:
```
# Make predictions on the test set using 0.45 as the cutoff

y_pred_final['final_predicted'] = y_pred_final.Conversion_Prob.map(lambda x: 1 if x > 0.
```

In [92]:
```
# Check y_pred_final

y_pred_final.head()
```

Out[92]:

	Converted	Conversion_Prob	final_predicted
0	1	0.987015	1
1	0	0.130010	0
2	0	0.703972	1
3	1	0.299230	0
4	1	0.720833	1

In [93]:
```
# Let's check the overall accuracy

metrics.accuracy_score(y_pred_final['Converted'], y_pred_final.final_predicted)
```

Out[93]:
```
0.7839958158995816
```

In [94]:
```
confusion2 = metrics.confusion_matrix(y_pred_final['Converted'], y_pred_final.final_pred
confusion2
```

Out[94]:
```
array([[786, 210],
       [203, 713]], dtype=int64)
```

In [95]:
```
TP = confusion2[1,1] # true positive
TN = confusion2[0,0] # true negatives
FP = confusion2[0,1] # false positives
FN = confusion2[1,0] # false negatives
```

In [96]:
```
# Calculate sensitivity
TP / float(TP+FN)
```

Out[96]:
```
0.7783842794759825
```

In [97]:
```
# Calculate specificity
```

```
TN / float(TN+FP)
```

Out[97]: 0.7891566265060241

Precision-Recall View

Let's now also build the training model using the precision-recall view Looking at the confusion matrix again

In [98]:
```
confusion = metrics.confusion_matrix(y_train_pred_final.Converted, y_train_pred_final.Pr
confusion
```

Out[98]:
```
array([[1929,  383],
       [ 560, 1589]], dtype=int64)
```

Precision TP / TP + FP

In [99]:
```
confusion[1,1]/(confusion[0,1]+confusion[1,1])
```

Out[99]: 0.8057809330628803

Recall TP / TP + FN

In [100…
```
confusion[1,1]/(confusion[1,0]+confusion[1,1])
```

Out[100]: 0.739413680781759

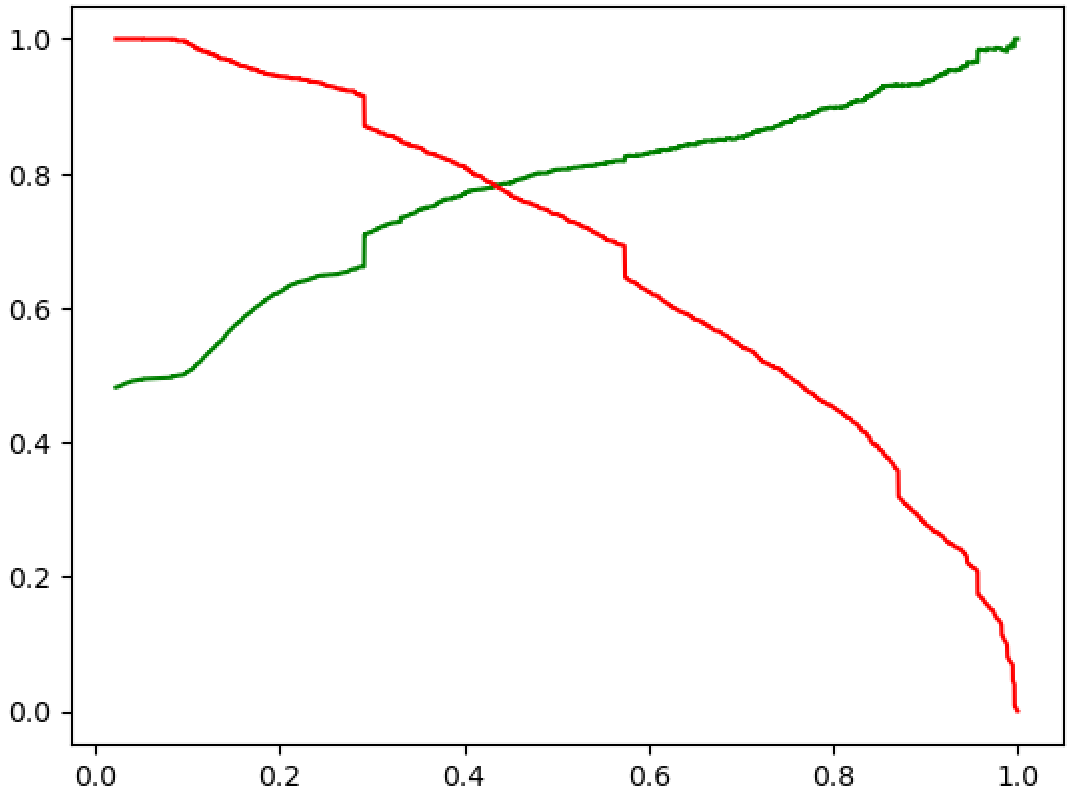
Precision and recall tradeoff

In [101…
```
y_train_pred_final.Converted, y_train_pred_final.Predicted
```

Out[101]:
```
(0       0
 1       0
 2       1
 3       1
 4       1
        ..
 4456    1
 4457    0
 4458    0
 4459    0
 4460    0
 Name: Converted, Length: 4461, dtype: int64,
 0       0
 1       0
 2       0
 3       0
 4       1
        ..
 4456    1
 4457    1
 4458    1
 4459    0
 4460    0
 Name: Predicted, Length: 4461, dtype: int64)
```

In [102…
```
p, r, thresholds = precision_recall_curve(y_train_pred_final.Converted, y_train_pred_fin
```

In [103…
```
plt.plot(thresholds, p[:-1], "g-")
plt.plot(thresholds, r[:-1], "r-")
plt.show()
```



In [104…

```
y_train_pred_final['final_predicted'] = y_train_pred_final.Conversion_Prob.map(lambda x:

y_train_pred_final.head()
```

Out[104]:

	Converted	Conversion_Prob	Predicted	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	final_predicted
0	0	0.300151	0	1	1	1	1	0	0	0	0	0	0	0
1	0	0.142019	0	1	1	0	0	0	0	0	0	0	0	0
2	1	0.127646	0	1	1	0	0	0	0	0	0	0	0	0
3	1	0.291594	0	1	1	1	0	0	0	0	0	0	0	0
4	1	0.956470	1	1	1	1	1	1	1	1	1	1	1	1

In [105…

```
# Let's check the accuracy now

metrics.accuracy_score(y_train_pred_final.Converted, y_train_pred_final.final_predicted)
```

Out[105]:

```
0.7895090786819099
```

In [106…

```
# Let's create the confusion matrix once again

confusion2 = metrics.confusion_matrix(y_train_pred_final.Converted, y_train_pred_final.f
confusion2
```

Out[106]:

```
array([[1852,  460],
       [ 479, 1670]], dtype=int64)
```

In [107…

```
# Let's evaluate the other metrics as well

TP = confusion2[1,1] # true positive
TN = confusion2[0,0] # true negatives
FP = confusion2[0,1] # false positives
FN = confusion2[1,0] # false negatives
```

Loading [MathJax]/extensions/Safe.js

In [108…

```
# Calculate Precision

TP/(TP+FP)
```

Out[108]:

```
0.784037558685446
```

In [109…

```
# Calculate Recall

TP/(TP+FN)
```

Out[109]:

```
0.7771056305258259
```

Step 4: Making Predictions on the Test Set

Let's now make predicitons on the test set.

In [110…

```
# Check if all columns in X_train are present in X_test
missing_columns = set(X_train.columns) - set(X_test.columns)

# Add missing columns to X_test with default values
for col in missing_columns:
    X_test[col] = 0

# Ensure that X_test has the same columns as X_train in the same order
X_test = X_test[X_train.columns]

# Make predictions on the test set
y_test_pred = res.predict(sm.add_constant(X_test))

# Access the first 10 predictions
y_test_pred[:10]
```

Out[110]:

```
4771    0.987015
6122    0.130010
9202    0.703972
6570    0.299230
2668    0.720833
4233    0.791825
3368    0.704066
9091    0.464563
5972    0.283009
3631    0.786488
dtype: float64
```

In [111…

```
# Converting y_pred to a dataframe

y_pred_1 = pd.DataFrame(y_test_pred)
```

In [112…

```
y_pred_1.head()
```

Out[112]:

	0
4771	0.987015
6122	0.130010
9202	0.703972
6570	0.299230
2668	0.720833

In [113…

```
# Converting y_test to dataframe

y_test_df = pd.DataFrame(y_test)
```

In [114…

```
# Remove index for both dataframes to append them side by side

y_pred_1.reset_index(drop=True, inplace=True)
y_test_df.reset_index(drop=True, inplace=True)
```

In [115…

```
# Append y_test_df and y_pred_1

y_pred_final = pd.concat([y_test_df, y_pred_1],axis=1)
```

In [116…

```
# Check 'y_pred_final'

y_pred_final.head()
```

Out[116]:

	Converted	0
0	1	0.987015
1	0	0.130010
2	0	0.703972
3	1	0.299230
4	1	0.720833

In [117…

```
# Rename the column

y_pred_final= y_pred_final.rename(columns = {0 : 'Conversion_Prob'})
```

In [118…

```
# Let's see the head of y_pred_final
y_pred_final.head()
```

Out[118]:

	Converted	Conversion_Prob
0	1	0.987015
1	0	0.130010
2	0	0.703972
3	1	0.299230
4	1	0.720833

In [119…

```
# Make predictions on the test set using 0.44 as the cutoff

y_pred_final['final_predicted'] = y_pred_final.Conversion_Prob.map(lambda x: 1 if x > 0.
```

In [120…

```
# Check y_pred_final

y_pred_final.head()
```

Loading [MathJax]/extensions/Safe.js

Out[120]:

	Converted	Conversion_Prob	final_predicted
0	1	0.987015	1
1	0	0.130010	0
2	0	0.703972	1
3	1	0.299230	0
4	1	0.720833	1

In [121…

```
# Let's check the overall accuracy

metrics.accuracy_score(y_pred_final['Converted'], y_pred_final.final_predicted)
```

Out[121]: 0.7866108786610879

In [122…

```
confusion2 = metrics.confusion_matrix(y_pred_final['Converted'], y_pred_final.final_pred
confusion2
```

Out[122]:

```
array([[801, 195],
       [213, 703]], dtype=int64)
```

In [123…

```
TP = confusion2[1,1] # true positive
TN = confusion2[0,0] # true negatives
FP = confusion2[0,1] # false positives
FN = confusion2[1,0] # false negatives
```

In [124…

```
# Calculate Precision

TP/(TP+FP)
```

Out[124]: 0.7828507795100222

In [125…

```
# Calculate Recall

TP/(TP+FN)
```

Out[125]: 0.767467248908297

In [ ]:

In [ ]:

In [ ]:

In [ ]:

Loading [MathJax]/extensions/Safe.js