In [2]:

```
import numpy as np
import pandas as pd
from datetime import datetime as dt
# For Visualisation
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
# To Scale our data
from sklearn.preprocessing import scale
# Supress Warnings
import warnings
warnings.filterwarnings('ignore')
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from sklearn import metrics
from sklearn.model_selection import train_test_split
import pandas as pd
```

In [3]:

```
leads = pd.read_csv("C:/Users/sudha/Desktop/csv/Leads.csv",  sep = ',',encoding = "ISO-8859-1")
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	Lead Profile	City	As A
0	7927b2df-8bba-4d29-b9a2-b6e0beafe620	660737	API	Olark Chat	No	No	0	0.0	0	0.0	...	No	Select	Select	
1	2a272436-5132-4136-86fa-dcc88c88f482	660728	API	Organic Search	No	No	0	5.0	674	2.5	...	No	Select	Select	
2	8cc8c611-a219-4f35-ad23-fdfd2656bd8a	660727	Landing Page Submission	Direct Traffic	No	No	1	2.0	1532	2.0	...	No	Potential Lead	Mumbai	
3	0cc2df48-7cf4-4e39-9de9-19797f9b38cc	660719	Landing Page Submission	Direct Traffic	No	No	0	1.0	305	1.0	...	No	Select	Mumbai	
4	3256f628-e534-4826-9d63-4a8b88782852	660681	Landing Page Submission	Google	No	No	1	2.0	1428	1.0	...	No	Select	Mumbai	

5 rows × 37 columns

Step 2: Inspecting the Dataframe

In [5]:

```
leads.dtypes
```

Out[5]:

```
Prospect ID                                       object
Lead Number                                        int64
Lead Origin                                       object
Lead Source                                       object
Do Not Email                                      object
Do Not Call                                       object
Converted                                          int64
```

```
Converted                                          int64
TotalVisits                                      float64
Total Time Spent on Website                        int64
Page Views Per Visit                             float64
Last Activity                                     object
Country                                           object
Specialization                                    object
How did you hear about X Education                object
What is your current occupation                   object
What matters most to you in choosing a course     object
Search                                            object
Magazine                                          object
Newspaper Article                                 object
X Education Forums                                object
Newspaper                                         object
Digital Advertisement                             object
Through Recommendations                           object
Receive More Updates About Our Courses            object
Tags                                              object
Lead Quality                                      object
Update me on Supply Chain Content                 object
Get updates on DM Content                         object
Lead Profile                                      object
City                                              object
Asymmetrique Activity Index                       object
Asymmetrique Profile Index                        object
Asymmetrique Activity Score                      float64
Asymmetrique Profile Score                       float64
I agree to pay the amount through cheque          object
A free copy of Mastering The Interview            object
Last Notable Activity                             object
dtype: object
```

In [6]:

```
leads.shape
```

Out[6]:

```
(9240, 37)
```

Step 3: Data Preparation

In [7]:

```
# removing duplicate rows
leads.drop_duplicates(subset='Lead Number')
leads.shape
```

Out[7]:

```
(9240, 37)
```

In [8]:

```
total = pd.DataFrame(leads.isnull().sum().sort_values(ascending=False), columns=['Total'])
percentage = pd.DataFrame(round(100*(leads.isnull().sum()/leads.shape[0]),2).sort_values(ascending=
False)\
                          ,columns=['Percentage'])
pd.concat([total, percentage], axis = 1)
```

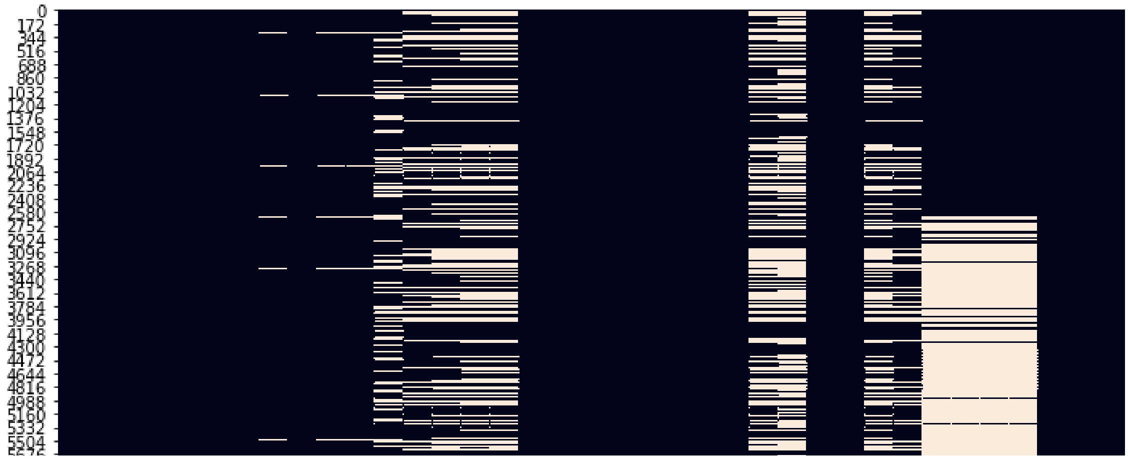
Out[8]:

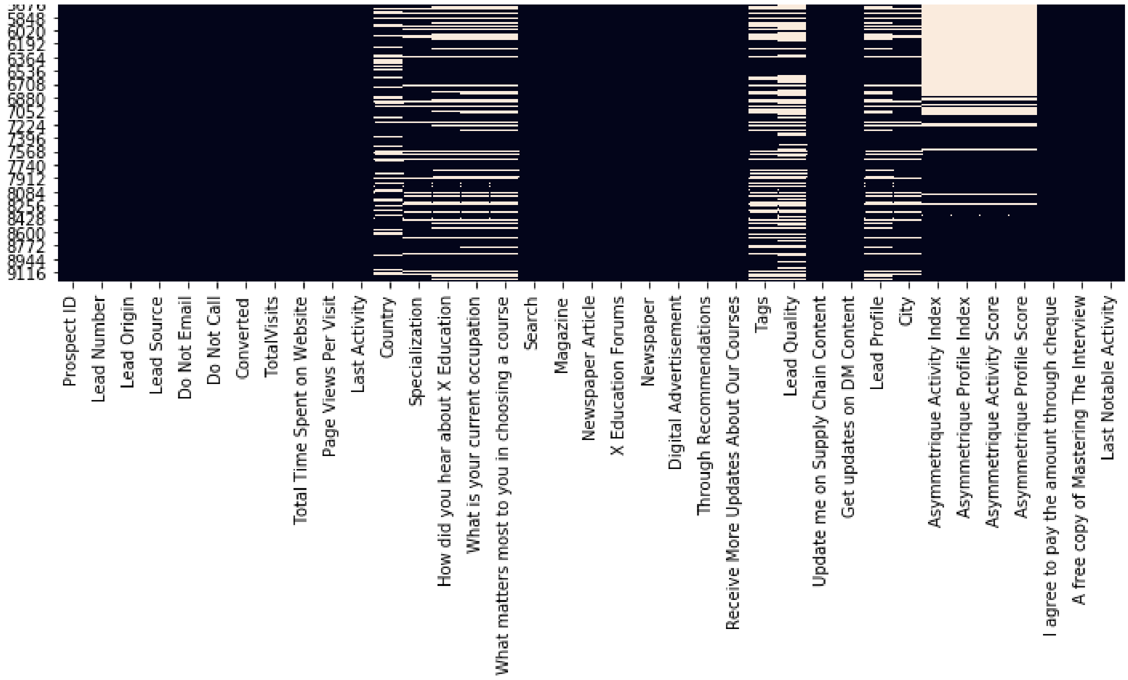
	Total	Percentage
Lead Quality	4767	51.59
Asymmetrique Profile Score	4218	45.65
Asymmetrique Activity Score	4218	45.65
Asymmetrique Profile Index	4218	45.65
Asymmetrique Activity Index	4218	45.65

	Total	Percentage
Asymmetrique Activity Index	4218	45.65
Tags	3353	36.29
What matters most to you in choosing a course	2709	29.32
Lead Profile	2709	29.32
What is your current occupation	2690	29.11
Country	2461	26.63
How did you hear about X Education	2207	23.89
Specialization	1438	15.56
City	1420	15.37
TotalVisits	137	1.48
Page Views Per Visit	137	1.48
Last Activity	103	1.11
Lead Source	36	0.39
Do Not Email	0	0.00
Do Not Call	0	0.00
Converted	0	0.00
Total Time Spent on Website	0	0.00
Lead Origin	0	0.00
Lead Number	0	0.00
Last Notable Activity	0	0.00
Newspaper Article	0	0.00
Search	0	0.00
Magazine	0	0.00
A free copy of Mastering The Interview	0	0.00
X Education Forums	0	0.00
Newspaper	0	0.00
Digital Advertisement	0	0.00
Through Recommendations	0	0.00
Receive More Updates About Our Courses	0	0.00
Update me on Supply Chain Content	0	0.00
Get updates on DM Content	0	0.00
I agree to pay the amount through cheque	0	0.00
Prospect ID	0	0.00

In [9]:

```
plt.figure(figsize=(10,10))
sns.heatmap(leads.isnull(), cbar=False)
plt.tight_layout()
plt.show()
```





In [10]:

```
leads.isnull().all(axis=0).any()
```

Out[10]:

False

In [11]:

```
leads.loc[:, (leads != 0).any(axis=0)]
leads.shape
```

Out[11]:

(9240, 37)

In [12]:

```
leads= leads.loc[:,leads.nunique()!=1]
leads.shape
```

Out[12]:

(9240, 32)

In [13]:

```
leads = leads.drop('Asymmetrique Activity Score', axis=1)
leads = leads.drop('Asymmetrique Profile Score', axis=1)
leads.shape
```

Out[13]:

(9240, 30)

In [14]:

```
leads = leads.drop('Prospect ID', axis=1)
#leads = leads.drop('Lead Number', axis=1)
leads.shape
```

Out[14]:

(9240, 29)

In [15]:

```
leads = leads.drop('What matters most to you in choosing a course', axis=1)
leads.shape
```

Out[15]:

```
(9240, 28)
```

In [16]:

```
leads = leads.drop('How did you hear about X Education', axis=1)
leads.shape
```

Out[16]:

```
(9240, 27)
```

Removing rows where a particular column has high missing values

In [17]:

```
leads['Lead Source'].isnull().sum()
```

Out[17]:

```
36
```

In [18]:

```
leads = leads[~pd.isnull(leads['Lead Source'])]
leads.shape
```

Out[18]:

```
(9204, 27)
```

In [21]:

```
leads['TotalVisits'].replace(np.NaN, leads['TotalVisits'].median(), inplace =True)
leads['Page Views Per Visit'].replace(np.NaN, leads['Page Views Per Visit'].median(), inplace
=True)
leads['Country'].mode()
```

Out[21]:

```
0    India
dtype: object
```

In [22]:

```
leads.loc[pd.isnull(leads['Country']), ['Country']] = 'India'
leads['Country'] = leads['Country'].apply(lambda x: 'India' if x=='India' else 'Outside India')
leads['Country'].value_counts()
```

Out[22]:

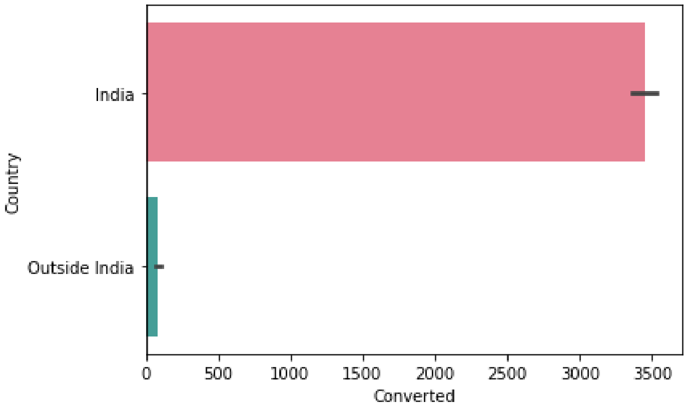
```
India            8917
Outside India     287
Name: Country, dtype: int64
```

In [23]:

```
sns.barplot(y='Country', x='Converted', palette='husl', data=leads, estimator=np.sum)
```

Out[23]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21757521a48>
```



In [24]:

```
leads['Lead Quality'].value_counts()
```

Out[24]:

```
Might be              1545
Not Sure              1090
High in Relevance      632
Worst                  601
Low in Relevance       583
Name: Lead Quality, dtype: int64
```

In [25]:

```
leads['Lead Quality'].isnull().sum()
```

Out[25]:

```
4753
```

In [26]:

```
leads['Lead Quality'].fillna("Unknown", inplace = True)
leads['Lead Quality'].value_counts()
```

Out[26]:

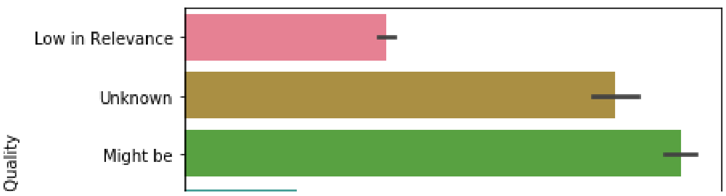
```
Unknown               4753
Might be              1545
Not Sure              1090
High in Relevance      632
Worst                  601
Low in Relevance       583
Name: Lead Quality, dtype: int64
```

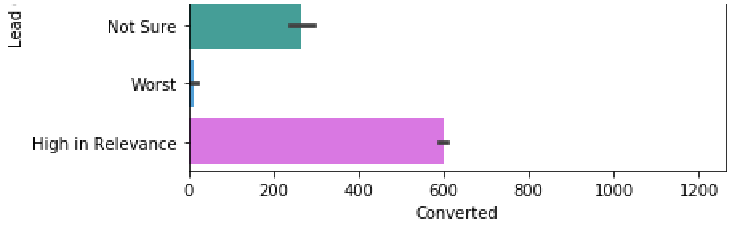
In [27]:

```
sns.barplot(y='Lead Quality', x='Converted', palette='husl', data=leads, estimator=np.sum)
```

Out[27]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x217581e1c88>
```





In [28]:

```
leads['Asymmetrique Profile Index'].value_counts()
```

Out[28]:

```
02.Medium    2771
01.High      2201
03.Low         31
Name: Asymmetrique Profile Index, dtype: int64
```

In [29]:

```
leads['Asymmetrique Profile Index'].isnull().sum()
```

Out[29]:

```
4201
```

In [30]:

```
leads['Asymmetrique Profile Index'].fillna("Unknown", inplace = True)
leads['Asymmetrique Profile Index'].value_counts()
```

Out[30]:

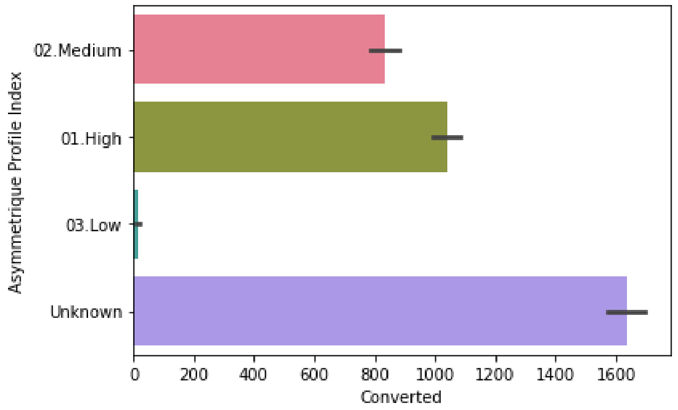
```
Unknown      4201
02.Medium    2771
01.High      2201
03.Low         31
Name: Asymmetrique Profile Index, dtype: int64
```

In [31]:

```
sns.barplot(y='Asymmetrique Profile Index', x='Converted', palette='husl', data=leads, estimator=np
.sum)
```

Out[31]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2175814f4c8>
```



In [32]:

```
leads['Asymmetrique Activity Index'].value_counts()
```

Out[32]:

```
02.Medium    3820
01.High       821
03.Low        362
Name: Asymmetrique Activity Index, dtype: int64
```

In [33]:

```
leads['Asymmetrique Activity Index'].isnull().sum()
```

Out[33]:

```
4201
```

In [34]:

```
leads['Asymmetrique Activity Index'].fillna("Unknown", inplace = True)
leads['Asymmetrique Activity Index'].value_counts()
```

Out[34]:

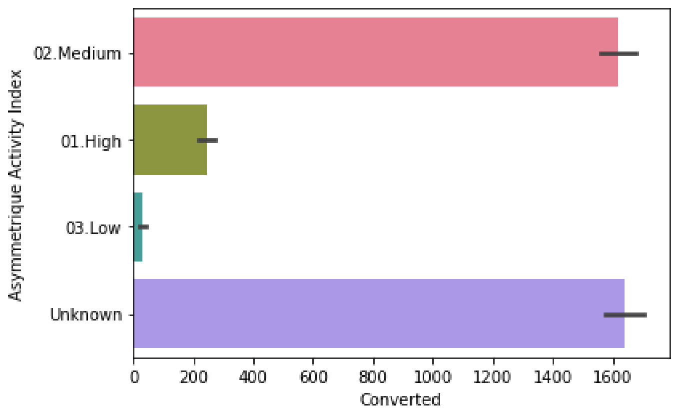
```
Unknown      4201
02.Medium    3820
01.High       821
03.Low        362
Name: Asymmetrique Activity Index, dtype: int64
```

In [35]:

```
sns.barplot(y='Asymmetrique Activity Index', x='Converted', palette='husl', data=leads, estimator=n
p.sum)
```

Out[35]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x217581bfd08>
```



In [36]:

```
leads['City'].isnull().sum()
```

Out[36]:

```
1420
```

In [37]:

```
leads['City'].fillna("Unknown", inplace = True)
leads['City'].value_counts()
```

Out[37]:

```
Mumbai                          3220
```

```
Select                          2218
Unknown                         1420
Thane & Outskirts                751
Other Cities                     686
Other Cities of Maharashtra      456
Other Metro Cities               379
Tier II Cities                    74
Name: City, dtype: int64
```

In [38]:

```
leads['City'].replace('Select', 'Unknown', inplace =True)
leads['City'].value_counts()
```

Out[38]:

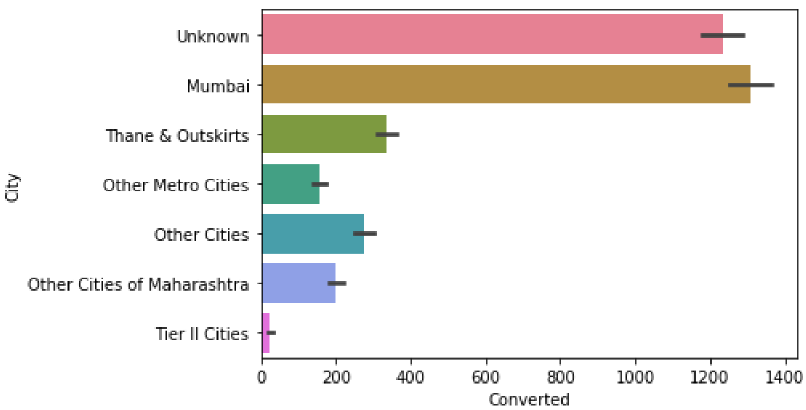
```
Unknown                         3638
Mumbai                          3220
Thane & Outskirts                751
Other Cities                     686
Other Cities of Maharashtra      456
Other Metro Cities               379
Tier II Cities                    74
Name: City, dtype: int64
```

In [39]:

```
sns.barplot(y='City', x='Converted', palette='husl', data=leads, estimator=np.sum)
```

Out[39]:

<matplotlib.axes._subplots.AxesSubplot at 0x217582ba9c8>



In [40]:

```
leads['Last Activity'].value_counts()
```

Out[40]:

```
Email Opened                    3432
SMS Sent                        2723
Olark Chat Conversation          973
Page Visited on Website          640
Converted to Lead                428
Email Bounced                    321
Email Link Clicked               267
Form Submitted on Website        116
Unreachable                       93
Unsubscribed                      59
Had a Phone Conversation          30
Approached upfront                 9
View in browser link Clicked       6
Email Marked Spam                  2
Email Received                     2
Resubscribed to emails             1
Visited Booth in Tradeshow         1
Name: Last Activity, dtype: int64
```

In [41]:

```
leads['Last Activity'].isnull().sum()
```

Out[41]:

```
101
```

In [43]:

```
leads['Last Activity'].fillna("Unknown", inplace = True)
leads['Last Activity'].value_counts()
```

Out[43]:

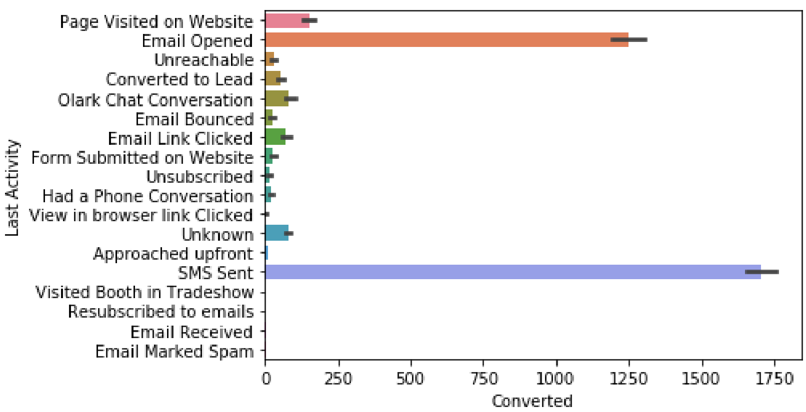
```
Email Opened                    3432
SMS Sent                        2723
Olark Chat Conversation          973
Page Visited on Website          640
Converted to Lead                428
Email Bounced                    321
Email Link Clicked               267
Form Submitted on Website        116
Unknown                          101
Unreachable                       93
Unsubscribed                      59
Had a Phone Conversation          30
Approached upfront                 9
View in browser link Clicked       6
Email Marked Spam                  2
Email Received                     2
Resubscribed to emails             1
Visited Booth in Tradeshow         1
Name: Last Activity, dtype: int64
```

In [44]:

```
sns.barplot(y='Last Activity', x='Converted', palette='husl', data=leads, estimator=np.sum)
```

Out[44]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2175828f488>
```



In [45]:

```
leads['Lead Profile'].value_counts()
```

Out[45]:

```
Select                         4115
Potential Lead                 1608
Other Leads                     487
Student of SomeSchool           241
Lateral Student                  24
```

```
Dual Specialization Student      20
Name: Lead Profile, dtype: int64
```

In [46]:

```
leads['Lead Profile'].isnull().sum()
```

Out[46]:

```
2709
```

In [48]:

```
leads['Lead Profile'].fillna("Unknown", inplace = True)
leads['Lead Profile'].value_counts()
```

Out[48]:

```
Select                          4115
Unknown                         2709
Potential Lead                  1608
Other Leads                      487
Student of SomeSchool            241
Lateral Student                   24
Dual Specialization Student       20
Name: Lead Profile, dtype: int64
```

In [49]:

```
leads['Lead Profile'].replace('Select', 'Unknown', inplace =True)
leads['Lead Profile'].value_counts()
```

Out[49]:

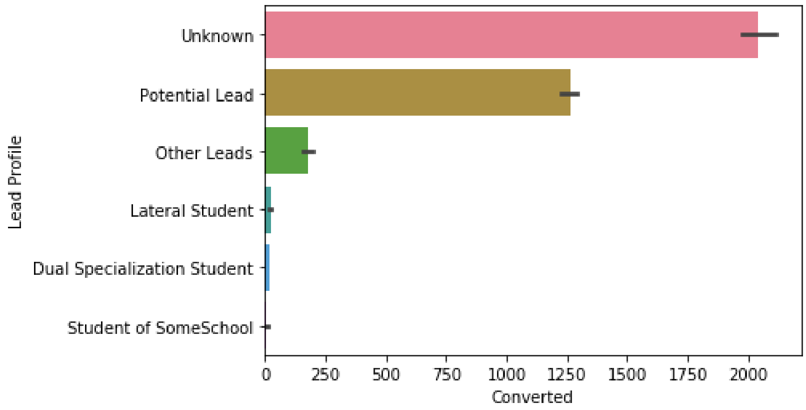
```
Unknown                         6824
Potential Lead                  1608
Other Leads                      487
Student of SomeSchool            241
Lateral Student                   24
Dual Specialization Student       20
Name: Lead Profile, dtype: int64
```

In [50]:

```
sns.barplot(y='Lead Profile', x='Converted', palette='husl', data=leads, estimator=np.sum)
```

Out[50]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x217583a7ec8>
```



In [51]:

```
leads['What is your current occupation'].value_counts()
```

Out[51]:

```
Unemployed              5567
Working Professional     704
Student                  209
Other                     16
Housewife                 10
Businessman                8
Name: What is your current occupation, dtype: int64
```

In [52]:

```
leads['What is your current occupation'].isnull().sum()
```

Out[52]:

```
2690
```

In [53]:

```
leads['What is your current occupation'].fillna("Unknown", inplace = True)
leads['What is your current occupation'].value_counts()
```

Out[53]:

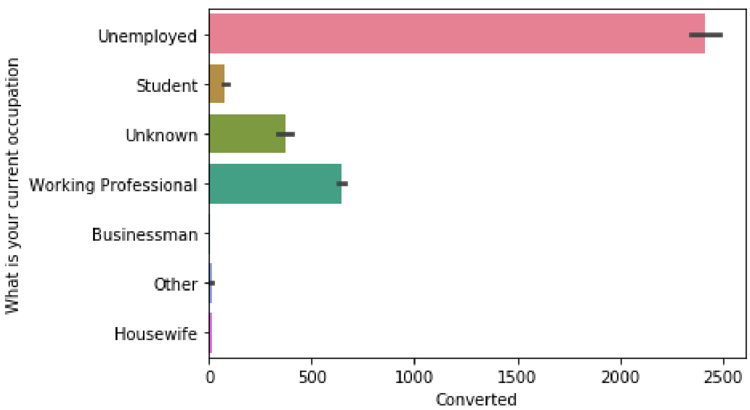
```
Unemployed              5567
Unknown                 2690
Working Professional     704
Student                  209
Other                     16
Housewife                 10
Businessman                8
Name: What is your current occupation, dtype: int64
```

In [54]:

```
sns.barplot(y='What is your current occupation', x='Converted', palette='husl', data=leads, estimat
or=np.sum)
```

Out[54]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2175842cd88>
```



In [55]:

```
leads['Specialization'].value_counts()
```

Out[55]:

```
Select                                 1914
Finance Management                      973
Human Resource Management               847
Marketing Management                    837
Operations Management                   502
Business Administration                 403
```

```
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                  158
Hospitality Management                 114
E-COMMERCE                             111
Retail Management                      100
Rural and Agribusiness                  73
E-Business                              57
Services Excellence                     40
Name: Specialization, dtype: int64
```

In [56]:

```python
leads['Specialization'].isnull().sum()
```

Out[56]:

```
1438
```

In [57]:

```python
leads['Specialization'].fillna("Unknown", inplace = True)
leads['Specialization'].value_counts()
```

Out[57]:

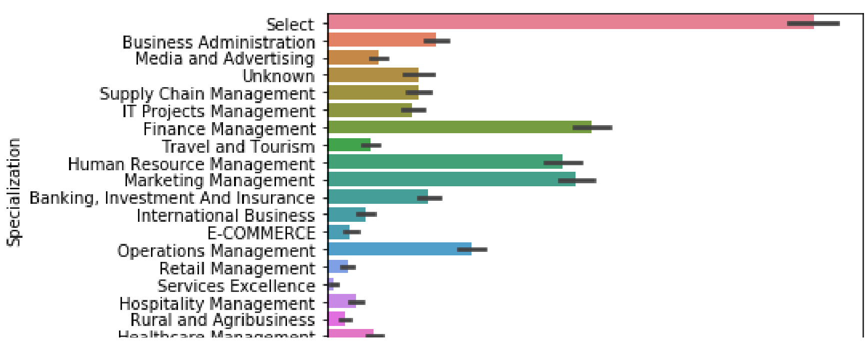
```
Select                                1914
Unknown                               1438
Finance Management                     973
Human Resource Management              847
Marketing Management                   837
Operations Management                  502
Business Administration                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                  158
Hospitality Management                 114
E-COMMERCE                             111
Retail Management                      100
Rural and Agribusiness                  73
E-Business                              57
Services Excellence                     40
Name: Specialization, dtype: int64
```

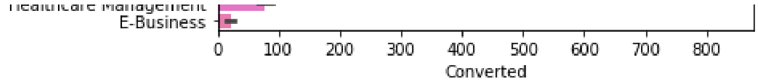
In [58]:

```python
sns.barplot(y='Specialization', x='Converted', palette='husl', data=leads, estimator=np.sum)
```

Out[58]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2175755d688>
```





In [59]:

```
leads['Tags'].value_counts()
```

Out[59]:

```
Will revert after reading the email                  2052
Ringing                                              1200
Interested in other courses                           513
Already a student                                     465
Closed by Horizzon                                    358
switched off                                          240
Busy                                                  186
Lost to EINS                                          174
Not doing further education                           145
Interested  in full time MBA                          117
Graduation in progress                                111
invalid number                                         83
Diploma holder (Not Eligible)                          63
wrong number given                                     47
opp hangup                                             33
number not provided                                    26
in touch with EINS                                     12
Lost to Others                                          7
Still Thinking                                          6
Want to take admission but has financial problems       6
In confusion whether part time or DLP                   5
Interested in Next batch                                5
Lateral student                                         3
Shall take in the next coming month                     2
University not recognized                               2
Recognition issue (DEC approval)                        1
Name: Tags, dtype: int64
```

In [60]:

```
leads['Tags'].isnull().sum()
```

Out[60]:

```
3342
```

In [61]:

```
leads['Tags'].fillna("Unknown", inplace = True)
leads['Tags'].value_counts()
```

Out[61]:

```
Unknown                                              3342
Will revert after reading the email                  2052
Ringing                                              1200
Interested in other courses                           513
Already a student                                     465
Closed by Horizzon                                    358
switched off                                          240
Busy                                                  186
Lost to EINS                                          174
Not doing further education                           145
Interested  in full time MBA                          117
Graduation in progress                                111
invalid number                                         83
Diploma holder (Not Eligible)                          63
wrong number given                                     47
opp hangup                                             33
number not provided                                    26
in touch with EINS                                     12
Lost to Others                                          7
Still Thinking                                          6
```

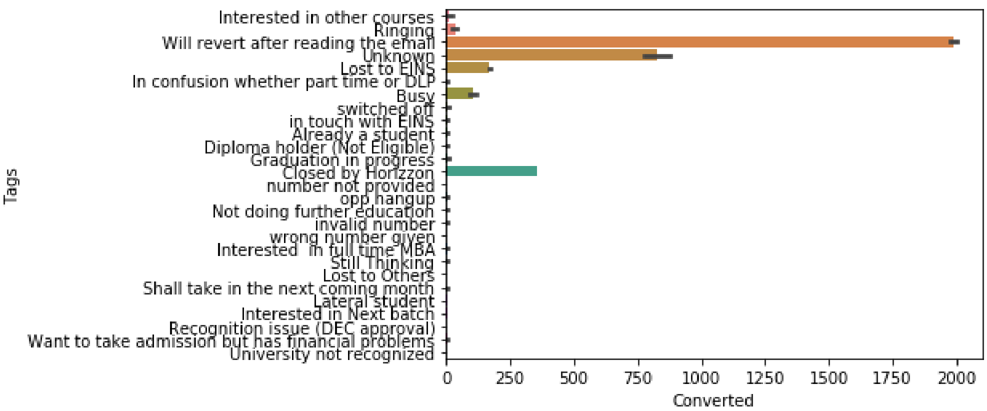
```
Want to take admission but has financial problems        6
In confusion whether part time or DLP                    5
Interested in Next batch                                 5
Lateral student                                          3
Shall take in the next coming month                      2
University not recognized                                2
Recognition issue (DEC approval)                         1
Name: Tags, dtype: int64
```

In [62]:

```
sns.barplot(y='Tags', x='Converted', palette='husl', data=leads, estimator=np.sum)
```

Out[62]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x217584b5348>
```



Reinspecting Null Values

In [63]:

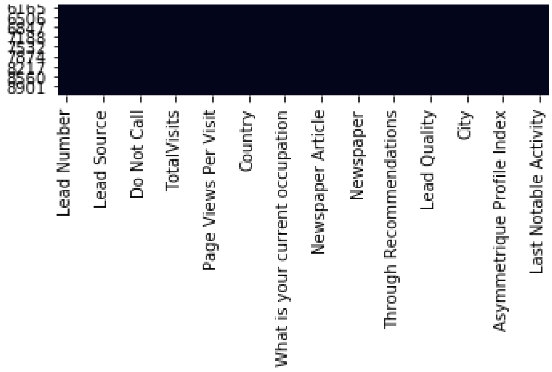
```
total = pd.DataFrame(leads.isnull().sum().sort_values(ascending=False), columns=['Total'])
percentage = pd.DataFrame(round(100*(leads.isnull().sum()/leads.shape[0]),2).sort_values(ascending=
False)\
                          ,columns=['Percentage'])
pd.concat([total, percentage], axis = 1).head()
```

Out[63]:

	Total	Percentage
Last Notable Activity	0	0.0
What is your current occupation	0	0.0
Lead Origin	0	0.0
Lead Source	0	0.0
Do Not Email	0	0.0

In [64]:

```
plt.figure(figsize=(5,5))
sns.heatmap(leads.isnull(), cbar=False)
plt.tight_layout()
plt.show()
```



Checking Outliers

In [65]:

```
leads.describe(percentiles=[.25,.5,.75,.90,.95,.99]).T
```

Out[65]:

	count	mean	std	min	25%	50%	75%	90%	95%	99%	max
Lead Number	9204.0	617194.608648	23418.830233	579533.0	596484.5	615479.0	637409.25	650513.1	655405.85	659599.46	660737.0
Converted	9204.0	0.383746	0.486324	0.0	0.0	0.0	1.00	1.0	1.00	1.00	1.0
TotalVisits	9204.0	3.449587	4.824662	0.0	1.0	3.0	5.00	7.0	10.00	17.00	251.0
Total Time Spent on Website	9204.0	489.005541	547.980340	0.0	14.0	250.0	938.00	1380.0	1562.00	1839.97	2272.0
Page Views Per Visit	9204.0	2.364923	2.145999	0.0	1.0	2.0	3.00	5.0	6.00	9.00	55.0

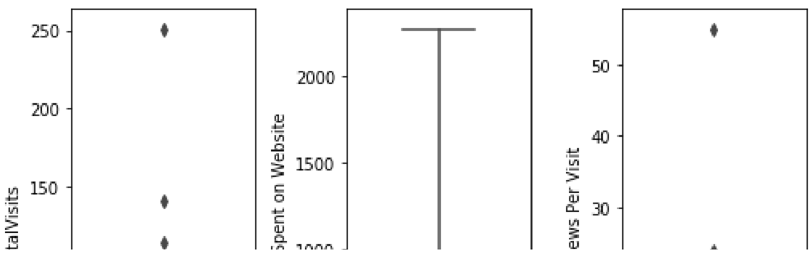
In [66]:

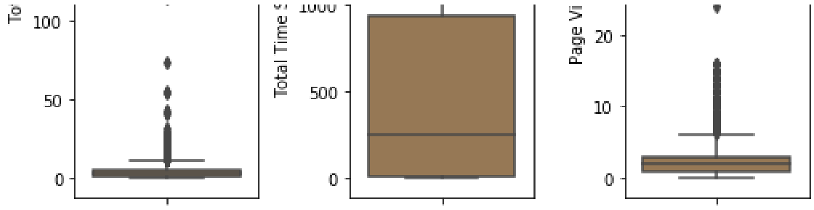
```
numeric_variables = ['TotalVisits','Total Time Spent on Website','Page Views Per Visit']
print(numeric_variables)
```

```
['TotalVisits', 'Total Time Spent on Website', 'Page Views Per Visit']
```

In [67]:

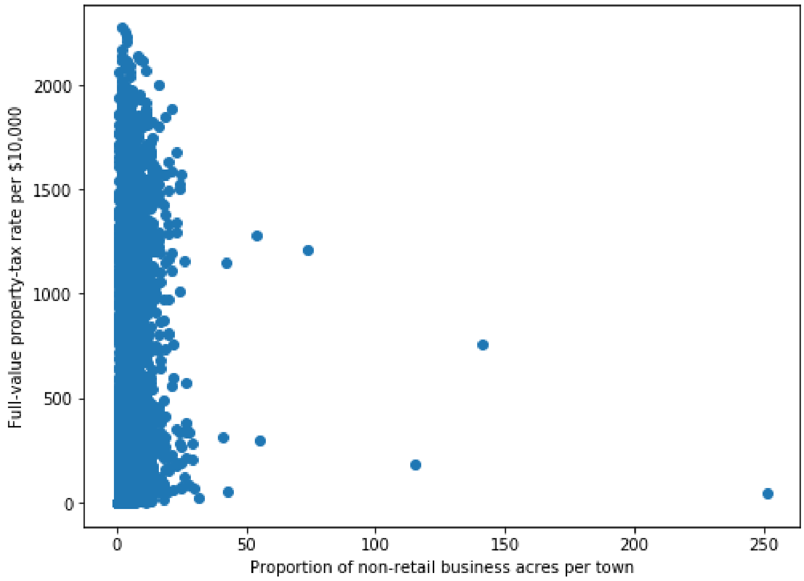
```
numeric_variables = ['TotalVisits','Total Time Spent on Website','Page Views Per Visit']

#Function to plot the distribution plot of the numeric variable list
def boxplot(var_list):
    plt.figure(figsize=(12,8))
    for var in var_list:
        plt.subplot(2,5,var_list.index(var)+1)
        #plt.boxplot(country[var])
        sns.boxplot(y=var,palette='cubehelix', data=leads)
    # Automatically adjust subplot params so that the subplotS fits in to the figure area.
    plt.tight_layout()
    # display the plot
    plt.show()

boxplot(numeric_variables)
```





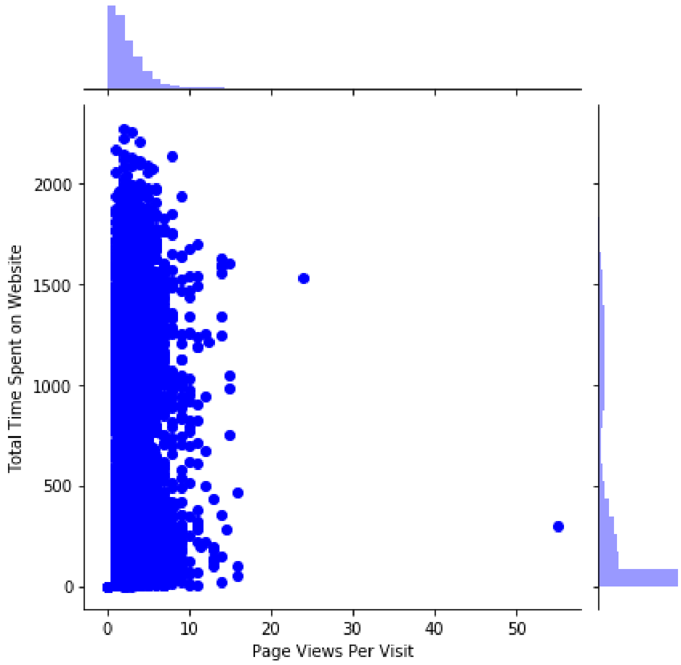
In [68]:

```
fig, ax = plt.subplots(figsize=(8,6))
ax.scatter(leads['TotalVisits'], leads['Total Time Spent on Website'])
ax.set_xlabel('Proportion of non-retail business acres per town')
ax.set_ylabel('Full-value property-tax rate per $10,000')
plt.show()
```



In [69]:

```
sns.jointplot(leads['Page Views Per Visit'],leads['Total Time Spent on Website'], color="b")
plt.show()
```



Removing outlier values based on the Interquartile distance for some of the

continuous variable

In [70]:

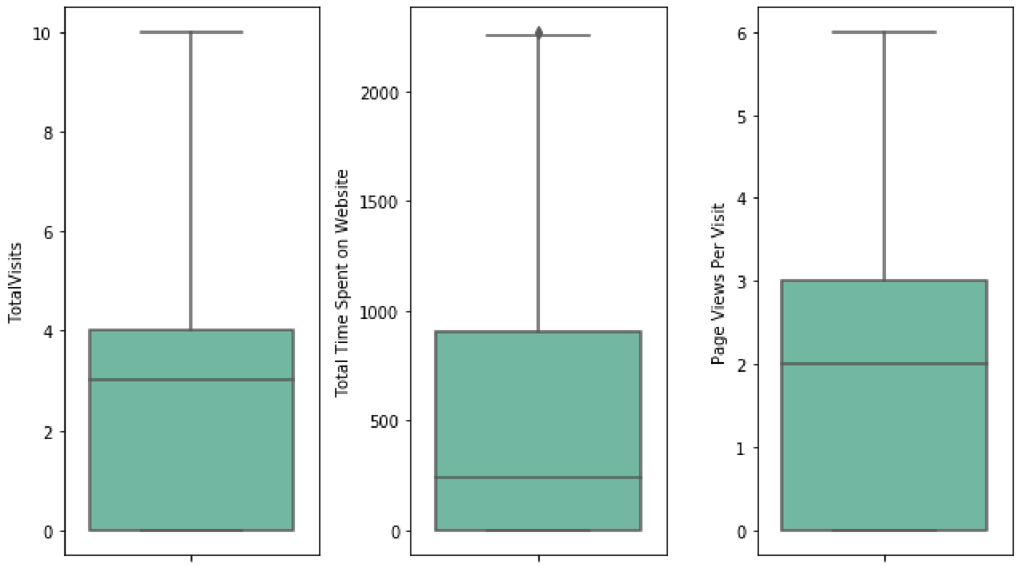
```
Q1 = leads['TotalVisits'].quantile(0.25)
Q3 = leads['TotalVisits'].quantile(0.75)
IQR = Q3 - Q1
leads=leads.loc[(leads['TotalVisits'] >= Q1 - 1.5*IQR) & (leads['TotalVisits'] <= Q3 + 1.4*IQR)]

Q1 = leads['Page Views Per Visit'].quantile(0.25)
Q3 = leads['Page Views Per Visit'].quantile(0.75)
IQR = Q3 - Q1
leads=leads.loc[(leads['Page Views Per Visit'] >= Q1 - 1.5*IQR) & (leads['Page Views Per Visit'] <=
Q3 + 1.5*IQR)]

leads.shape
```

Out[70]:

```
(8575, 27)
```

In [71]:

```
def boxplot(var_list):
    plt.figure(figsize=(15,10))
    for var in var_list:
        plt.subplot(2,5,var_list.index(var)+1)
        #plt.boxplot(country[var])
        sns.boxplot(y=var,palette='BuGn_r', data=leads)
    # Automatically adjust subplot params so that the subplotS fits in to the figure area.
    plt.tight_layout()
    # display the plot
    plt.show()

boxplot(numeric_variables)
```



In [72]:

```
leads.shape
```

Out[72]:

```
(8575, 27)
```

Converting some binary variables (Yes/No) to 0/1

In [73]:

```
varlist =  ['Search','Do Not Email', 'Do Not Call', 'Newspaper Article', 'X Education Forums',
```

```
'Newspaper',
             'Digital Advertisement','Through Recommendations','A free copy of Mastering The Intervi
ew']

# Defining the map function
def binary_map(x):
    return x.map({'Yes': 1, "No": 0})

# Applying the function to the housing list
leads[varlist] = leads[varlist].apply(binary_map)
leads.head()
```

Out[73]:

	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	...	Digital Advertisement	Throug Recommendation
0	660737	API	Olark Chat	0	0	0	0.0	0	0.0	Page Visited on Website	...	0	
1	660728	API	Organic Search	0	0	0	5.0	674	2.5	Email Opened	...	0	
2	660727	Landing Page Submission	Direct Traffic	0	0	1	2.0	1532	2.0	Email Opened	...	0	
3	660719	Landing Page Submission	Direct Traffic	0	0	0	1.0	305	1.0	Unreachable	...	0	
4	660681	Landing Page Submission	Google	0	0	1	2.0	1428	1.0	Converted to Lead	...	0	

5 rows × 27 columns

For categorical variables with multiple levels, creating dummy features

In [75]:

```
dummy1 = pd.get_dummies(leads[['Country', 'Lead Source','Lead Origin','Last Notable Activity']],
drop_first=True)

# Adding the results to the master dataframe
leads = pd.concat([leads, dummy1], axis=1)
leads.shape
```

Out[75]:

```
(8575, 105)
```

In [76]:

```
# Creating dummy variables for the variable 'Lead Quality'
ml = pd.get_dummies(leads['Lead Quality'], prefix='Lead Quality')
# Dropping the level called 'Unknown' which represents null/select values
ml1 = ml.drop(['Lead Quality_Unknown'], 1)
#Adding the results to the master dataframe
leads = pd.concat([leads,ml1], axis=1)
ml = pd.get_dummies(leads['Asymmetrique Profile Index'], prefix='Asymmetrique Profile Index')
# Dropping the level called 'Unknown' which represents null/select values
ml1 = ml.drop(['Asymmetrique Profile Index_Unknown'], 1)
#Adding the results to the master dataframe
leads = pd.concat([leads,ml1], axis=1)
# Creating dummy variables for the variable 'Asymmetrique Activity Index'
ml = pd.get_dummies(leads['Asymmetrique Activity Index'], prefix='Asymmetrique Activity Index')
# Dropping the level called 'Unknown' which represents null/select values
ml1 = ml.drop(['Asymmetrique Activity Index_Unknown'], 1)
#Adding the results to the master dataframe
leads = pd.concat([leads,ml1], axis=1)
```

```
leads = pd.concat([leads,ml1], axis=1)
ml = pd.get_dummies(leads['Tags'], prefix='Tags')
ml1 = ml.drop(['Tags_Unknown'], 1)
leads = pd.concat([leads,ml1], axis=1)
ml = pd.get_dummies(leads['Lead Profile'], prefix='Lead Profile')
ml1 = ml.drop(['Lead Profile_Unknown'], 1)
leads = pd.concat([leads,ml1], axis=1)
ml = pd.get_dummies(leads['What is your current occupation'], prefix='What is your current
occupation')
ml1 = ml.drop(['What is your current occupation_Unknown'], 1)
leads = pd.concat([leads,ml1], axis=1)
ml = pd.get_dummies(leads['Specialization'], prefix='Specialization')
ml1 = ml.drop(['Specialization_Unknown'], 1)
leads = pd.concat([leads,ml1], axis=1)
ml = pd.get_dummies(leads['City'], prefix='City')
ml1 = ml.drop(['City_Unknown'], 1)
leads = pd.concat([leads,ml1], axis=1)
ml = pd.get_dummies(leads['Last Activity'], prefix='Last Activity')
ml1 = ml.drop(['Last Activity_Unknown'], 1)
leads = pd.concat([leads,ml1], axis=1)
leads.shape
```

Out[76]:

```
(8575, 195)
```

Dropping the repeated variables

In [77]:

```
leads = leads.drop(['Lead Quality','Asymmetrique Profile Index','Asymmetrique Activity Index','Tag
s','Lead Profile',
                    'Lead Origin','What is your current occupation', 'Specialization', 'City','Last
Activity', 'Country',
                    'Lead Source','Last Notable Activity'], 1)
leads.shape
```

Out[77]:

```
(8575, 182)
```

In [78]:

```
leads.head()
```

Out[78]:

	Lead Number	Do Not Email	Do Not Call	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Search	Newspaper Article	X Education Forums	...	Last Activity_Form Submitted on Website	Last Activity_Had a Phone Conversation	Acti Cor
0	660737	0	0	0	0.0	0	0.0	0	0	0	...	0	0	
1	660728	0	0	0	5.0	674	2.5	0	0	0	...	0	0	
2	660727	0	0	1	2.0	1532	2.0	0	0	0	...	0	0	
3	660719	0	0	0	1.0	305	1.0	0	0	0	...	0	0	
4	660681	0	0	1	2.0	1428	1.0	0	0	0	...	0	0	

5 rows × 182 columns

In [79]:

```
cols = leads.columns
num_cols = leads._get_numeric_data().columns
list(set(cols) - set(num_cols))
```

Out[79]:

```
[]
```

In [80]:

```
original_leads = leads.copy()
print(original_leads.shape)
print(leads.shape)
```

```
(8575, 182)
(8575, 182)
```

Step 4: Test-Train Split

In [81]:

```
from sklearn.model_selection import train_test_split
```

In [82]:

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

Out[82]:

	Do Not Email	Do Not Call	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Search	Newspaper Article	X Education Forums	Newspaper	Digital Advertisement	...	Last Activity_Form Submitted on Website	Las Activity_Ha a Phon Conversatio
0	0	0	0.0	0	0.0	0	0	0	0	0	...	0	
1	0	0	5.0	674	2.5	0	0	0	0	0	...	0	
2	0	0	2.0	1532	2.0	0	0	0	0	0	...	0	
3	0	0	1.0	305	1.0	0	0	0	0	0	...	0	
4	0	0	2.0	1428	1.0	0	0	0	0	0	...	0	

5 rows × 180 columns

In [83]:

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

Out[83]:

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

In [84]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.7, test_size=0.3,
random_state=100)
```

Step 5: Feature Scaling

In [85]:

```
from sklearn.preprocessing import StandardScaler
```

In [86]:

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

Out[86]:

	Do Not Email	Do Not Call	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Search	Newspaper Article	X Education Forums	Newspaper	Digital Advertisement	...	Last Activity_Form Submitted on Website	Acti Conv
8529	0	0	0.969969	-0.864724	1.785283	0	0	0	0	0	...	0	
7331	0	0	0.102087	-0.215257	0.562949	0	0	0	0	0	...	0	
7688	0	0	0.102087	1.523992	0.562949	0	0	0	0	0	...	0	
92	0	0	0.536028	-0.686762	1.174116	0	0	0	0	0	...	0	
4908	0	0	-1.199737	-0.872062	-1.270553	0	0	0	0	0	...	0	

5 rows × 180 columns

In [87]:

```
X_train.describe()
```

Out[87]:

	Do Not Email	Do Not Call	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Search	Newspaper Article	X Education Forums	Newspaper	Digit Advertisemei
count	6002.000000	6002.0	6.002000e+03	6.002000e+03	6.002000e+03	6002.000000	6002.0	6002.0	6002.000000	6002.00000
mean	0.076308	0.0	6.130088e-17	1.427826e-16	1.538996e-17	0.001000	0.0	0.0	0.000167	0.00033
std	0.265512	0.0	1.000083e+00	1.000083e+00	1.000083e+00	0.031604	0.0	0.0	0.012908	0.01825
min	0.000000	0.0	-1.199737e+00	-8.720622e-01	-1.270553e+00	0.000000	0.0	0.0	0.000000	0.00000
25%	0.000000	0.0	-7.657957e-01	-8.683929e-01	-6.593854e-01	0.000000	0.0	0.0	0.000000	0.00000
50%	0.000000	0.0	1.020868e-01	-4.381673e-01	-4.821826e-02	0.000000	0.0	0.0	0.000000	0.00000
75%	0.000000	0.0	5.360281e-01	7.846274e-01	5.629489e-01	0.000000	0.0	0.0	0.000000	0.00000
max	1.000000	0.0	3.139676e+00	3.296264e+00	2.396450e+00	1.000000	0.0	0.0	1.000000	1.00000

8 rows × 180 columns

In [88]:

```
### Checking the Lead Conversion Rate
converted = (sum(leads['Converted'])/len(leads['Converted'].index))*100
converted
```

Out[88]:

```
38.04081632653061
```

Step 6: Model Building

In [89]:

```
import statsmodels.api as sm
```

In [90]:

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

Out[90]:

Generalized Linear Model Regression Results

Dep. Variable:	Converted	**No. Observations:**	6002
Model:	GLM	**Df Residuals:**	5871
Model Family:	Binomial	**Df Model:**	130
Link Function:	logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	nan
Date:	Wed, 13 May 2020	**Deviance:**	nan
Time:	11:46:21	**Pearson chi2:**	2.01e+18
No. Iterations:	100		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	5.132e+14	1.08e+08	4.73e+06	0.000	5.13e+14	5.13e+14
Do Not Email	-2.358e+14	4.66e+06	-5.06e+07	0.000	-2.36e+14	-2.36e+14
Do Not Call	-214.9875	3.31e-06	-6.49e+07	0.000	-214.987	-214.987
TotalVisits	6.628e+13	1.51e+06	4.38e+07	0.000	6.63e+13	6.63e+13
Total Time Spent on Website	1.788e+14	1.07e+06	1.67e+08	0.000	1.79e+14	1.79e+14
Page Views Per Visit	-8.408e+13	1.64e+06	-5.13e+07	0.000	-8.41e+13	-8.41e+13
Search	2.925e+14	2.9e+07	1.01e+07	0.000	2.93e+14	2.93e+14
Newspaper Article	-4.3763	3.44e-07	-1.27e+07	0.000	-4.376	-4.376
X Education Forums	53.8288	8.31e-07	6.47e+07	0.000	53.829	53.829
Newspaper	-1.37e+15	6.76e+07	-2.03e+07	0.000	-1.37e+15	-1.37e+15
Digital Advertisement	1.117e+15	4.85e+07	2.3e+07	0.000	1.12e+15	1.12e+15
Through Recommendations	-7.604e+14	5e+07	-1.52e+07	0.000	-7.6e+14	-7.6e+14
A free copy of Mastering The Interview	-3.241e+13	2.94e+06	-1.1e+07	0.000	-3.24e+13	-3.24e+13
Country_Outside India	6.538e+13	2.49e+06	2.62e+07	0.000	6.54e+13	6.54e+13
Lead Source_Direct Traffic	4.492e+14	3.98e+07	1.13e+07	0.000	4.49e+14	4.49e+14
Lead Source_Facebook	1.316e+14	2.01e+07	6.56e+06	0.000	1.32e+14	1.32e+14
Lead Source_Google	4.589e+14	3.98e+07	1.15e+07	0.000	4.59e+14	4.59e+14
Lead Source_Live Chat	3.992e+14	3.16e+07	1.26e+07	0.000	3.99e+14	3.99e+14
Lead Source_NC_EDM	5.926e+14	5.21e+07	1.14e+07	0.000	5.93e+14	5.93e+14
Lead Source_Olark Chat	5.383e+14	3.97e+07	1.36e+07	0.000	5.38e+14	5.38e+14
Lead Source_Organic Search	3.782e+14	3.98e+07	9.5e+06	0.000	3.78e+14	3.78e+14
Lead Source_Pay per Click Ads	-2.147e+15	5.21e+07	-4.12e+07	0.000	-2.15e+15	-2.15e+15
Lead Source_Press_Release	91.2283	8.38e-07	1.09e+08	0.000	91.228	91.228
Lead Source_Reference	3.398e+14	2.08e+07	1.64e+07	0.000	3.4e+14	3.4e+14
Lead Source_Referral Sites	3.557e+14	3.99e+07	8.9e+06	0.000	3.56e+14	3.56e+14
Lead Source_Social Media	-2.778e+15	5.3e+07	-5.24e+07	0.000	-2.78e+15	-2.78e+15
Lead Source_WeLearn	19.2616	3.26e-07	5.91e+07	0.000	19.262	19.262
Lead Source_Welingak Website	6.597e+14	2.1e+07	3.14e+07	0.000	6.6e+14	6.6e+14
Lead Source_bing	-2.569e+14	4.64e+07	-5.53e+06	0.000	-2.57e+14	-2.57e+14
Lead Source_blog	-2.124e+15	5.22e+07	-4.07e+07	0.000	-2.12e+15	-2.12e+15
Lead Source_google	4.949e+14	4.65e+07	1.06e+07	0.000	4.95e+14	4.95e+14
Lead Source_testone	-1.581e+15	5.22e+07	-3.03e+07	0.000	-1.58e+15	-1.58e+15

Lead Source_welearnblog_Home	-2.342e+15	5.21e+07	-4.5e+07	0.000	-2.34e+15	-2.34e+15
Lead Source_youtubechannel	-69.2819	6.62e-07	-1.05e+08	0.000	-69.282	-69.282
Lead Origin_Landing Page Submission	-1.995e+13	2.14e+06	-9.32e+06	0.000	-1.99e+13	-1.99e+13
Lead Origin_Lead Add Form	1.175e+14	3.39e+07	3.47e+06	0.000	1.18e+14	1.18e+14
Lead Origin_Lead Import	1.316e+14	2.01e+07	6.56e+06	0.000	1.32e+14	1.32e+14
Last Notable Activity_Email Bounced	-5.383e+14	3.71e+07	-1.45e+07	0.000	-5.38e+14	-5.38e+14
Last Notable Activity_Email Link Clicked	-1.235e+15	3.69e+07	-3.34e+07	0.000	-1.23e+15	-1.23e+15
Last Notable Activity_Email Marked Spam	1.99e+14	2.92e+07	6.8e+06	0.000	1.99e+14	1.99e+14
Last Notable Activity_Email Opened	-9.348e+14	3.66e+07	-2.55e+07	0.000	-9.35e+14	-9.35e+14
Last Notable Activity_Email Received	7.247e+15	6e+07	1.21e+08	0.000	7.25e+15	7.25e+15
Last Notable Activity_Form Submitted on Website	-3.129e+15	4.99e+07	-6.27e+07	0.000	-3.13e+15	-3.13e+15
Last Notable Activity_Had a Phone Conversation	-1.203e+15	4.09e+07	-2.94e+07	0.000	-1.2e+15	-1.2e+15
Last Notable Activity_Modified	-1.083e+15	3.66e+07	-2.96e+07	0.000	-1.08e+15	-1.08e+15
Last Notable Activity_Olark Chat Conversation	-1.207e+15	3.67e+07	-3.29e+07	0.000	-1.21e+15	-1.21e+15
Last Notable Activity_Page Visited on Website	-1.011e+15	3.68e+07	-2.75e+07	0.000	-1.01e+15	-1.01e+15
Last Notable Activity_Resubscribed to emails	9.2309	7.84e-07	1.18e+07	0.000	9.231	9.231
Last Notable Activity_SMS Sent	-8.544e+14	3.66e+07	-2.33e+07	0.000	-8.54e+14	-8.54e+14
Last Notable Activity_Unreachable	-1.055e+15	3.77e+07	-2.8e+07	0.000	-1.06e+15	-1.06e+15
Last Notable Activity_Unsubscribed	-1.308e+15	3.84e+07	-3.4e+07	0.000	-1.31e+15	-1.31e+15
Last Notable Activity_View in browser link Clicked	3.383e+14	6.02e+07	5.62e+06	0.000	3.38e+14	3.38e+14
Country_Outside India	6.538e+13	2.49e+06	2.62e+07	0.000	6.54e+13	6.54e+13
Lead Source_Direct Traffic	4.492e+14	3.98e+07	1.13e+07	0.000	4.49e+14	4.49e+14
Lead Source_Facebook	1.316e+14	2.01e+07	6.56e+06	0.000	1.32e+14	1.32e+14
Lead Source_Google	4.589e+14	3.98e+07	1.15e+07	0.000	4.59e+14	4.59e+14
Lead Source_Live Chat	3.992e+14	3.16e+07	1.26e+07	0.000	3.99e+14	3.99e+14
Lead Source_NC_EDM	5.926e+14	5.21e+07	1.14e+07	0.000	5.93e+14	5.93e+14
Lead Source_Olark Chat	5.383e+14	3.97e+07	1.36e+07	0.000	5.38e+14	5.38e+14
Lead Source_Organic Search	3.782e+14	3.98e+07	9.5e+06	0.000	3.78e+14	3.78e+14
Lead Source_Pay per Click Ads	-2.147e+15	5.21e+07	-4.12e+07	0.000	-2.15e+15	-2.15e+15
Lead Source_Press_Release	-2.9643	3.88e-07	-7.63e+06	0.000	-2.964	-2.964
Lead Source_Reference	3.398e+14	2.08e+07	1.64e+07	0.000	3.4e+14	3.4e+14
Lead Source_Referral Sites	3.557e+14	3.99e+07	8.9e+06	0.000	3.56e+14	3.56e+14
Lead Source_Social Media	-2.778e+15	5.3e+07	-5.24e+07	0.000	-2.78e+15	-2.78e+15
Lead Source_WeLearn	0.5151	1.06e-07	4.84e+06	0.000	0.515	0.515
Lead Source_Welingak Website	6.597e+14	2.1e+07	3.14e+07	0.000	6.6e+14	6.6e+14
Lead Source_bing	-2.569e+14	4.64e+07	-5.53e+06	0.000	-2.57e+14	-2.57e+14
Lead Source_blog	-2.124e+15	5.22e+07	-4.07e+07	0.000	-2.12e+15	-2.12e+15
Lead Source_google	4.949e+14	4.65e+07	1.06e+07	0.000	4.95e+14	4.95e+14
Lead Source_testone	-1.581e+15	5.22e+07	-3.03e+07	0.000	-1.58e+15	-1.58e+15
Lead Source_welearnblog_Home	-2.342e+15	5.21e+07	-4.5e+07	0.000	-2.34e+15	-2.34e+15
Lead Source_youtubechannel	-5.9974	5.64e-07	-1.06e+07	0.000	-5.997	-5.997
Lead Origin_Landing Page Submission	-1.995e+13	2.14e+06	-9.32e+06	0.000	-1.99e+13	-1.99e+13
Lead Origin_Lead Add Form	1.175e+14	3.39e+07	3.47e+06	0.000	1.18e+14	1.18e+14
Lead Origin_Lead Import	1.316e+14	2.01e+07	6.56e+06	0.000	1.32e+14	1.32e+14
Last Notable Activity_Email Bounced	-5.383e+14	3.71e+07	-1.45e+07	0.000	-5.38e+14	-5.38e+14
Last Notable Activity_Email Link Clicked	-1.235e+15	3.69e+07	-3.34e+07	0.000	-1.23e+15	-1.23e+15
Last Notable Activity_Email Marked Spam	1.99e+14	2.92e+07	6.8e+06	0.000	1.99e+14	1.99e+14
Last Notable Activity_Email Opened	-9.348e+14	3.66e+07	-2.55e+07	0.000	-9.35e+14	-9.35e+14
Last Notable Activity_Email Received	7.247e+15	6e+07	1.21e+08	0.000	7.25e+15	7.25e+15
Last Notable Activity_Form Submitted on Website	-3.129e+15	4.99e+07	-6.27e+07	0.000	-3.13e+15	-3.13e+15

Last Notable Activity_Had a Phone Conversation	-1.203e+15	4.09e+07	-2.94e+07	0.000	-1.2e+15	-1.2e+15
Last Notable Activity_Modified	-1.083e+15	3.66e+07	-2.96e+07	0.000	-1.08e+15	-1.08e+15
Last Notable Activity_Olark Chat Conversation	-1.207e+15	3.67e+07	-3.29e+07	0.000	-1.21e+15	-1.21e+15
Last Notable Activity_Page Visited on Website	-1.011e+15	3.68e+07	-2.75e+07	0.000	-1.01e+15	-1.01e+15
Last Notable Activity_Resubscribed to emails	-0.9981	1.25e-07	-7.96e+06	0.000	-0.998	-0.998
Last Notable Activity_SMS Sent	-8.544e+14	3.66e+07	-2.33e+07	0.000	-8.54e+14	-8.54e+14
Last Notable Activity_Unreachable	-1.055e+15	3.77e+07	-2.8e+07	0.000	-1.06e+15	-1.06e+15
Last Notable Activity_Unsubscribed	-1.308e+15	3.84e+07	-3.4e+07	0.000	-1.31e+15	-1.31e+15
Last Notable Activity_View in browser link Clicked	3.383e+14	6.02e+07	5.62e+06	0.000	3.38e+14	3.38e+14
Lead Quality_High in Relevance	-1.53e+14	5.63e+06	-2.72e+07	0.000	-1.53e+14	-1.53e+14
Lead Quality_Low in Relevance	-2.644e+14	5.45e+06	-4.85e+07	0.000	-2.64e+14	-2.64e+14
Lead Quality_Might be	-8.566e+13	4.06e+06	-2.11e+07	0.000	-8.57e+13	-8.57e+13
Lead Quality_Not Sure	1.489e+14	3.68e+06	4.04e+07	0.000	1.49e+14	1.49e+14
Lead Quality_Worst	-3.809e+14	5.57e+06	-6.83e+07	0.000	-3.81e+14	-3.81e+14
Asymmetrique Profile Index_01.High	-1.014e+14	3.86e+06	-2.63e+07	0.000	-1.01e+14	-1.01e+14
Asymmetrique Profile Index_02.Medium	3.992e+13	3.34e+06	1.2e+07	0.000	3.99e+13	3.99e+13
Asymmetrique Profile Index_03.Low	-3.278e+14	1.44e+07	-2.28e+07	0.000	-3.28e+14	-3.28e+14
Asymmetrique Activity Index_01.High	1.356e+14	4.13e+06	3.28e+07	0.000	1.36e+14	1.36e+14
Asymmetrique Activity Index_02.Medium	2.979e+13	3.34e+06	8.91e+06	0.000	2.98e+13	2.98e+13
Asymmetrique Activity Index_03.Low	-5.547e+14	5.07e+06	-1.09e+08	0.000	-5.55e+14	-5.55e+14
Tags_Already a student	-1.642e+15	6.49e+06	-2.53e+08	0.000	-1.64e+15	-1.64e+15
Tags_Busy	5.451e+14	7.61e+06	7.17e+07	0.000	5.45e+14	5.45e+14
Tags_Closed by Horizzon	5.665e+14	7.01e+06	8.08e+07	0.000	5.67e+14	5.67e+14
Tags_Diploma holder (Not Eligible)	-3.63e+15	1.11e+07	-3.27e+08	0.000	-3.63e+15	-3.63e+15
Tags_Graduation in progress	-9.141e+14	9.08e+06	-1.01e+08	0.000	-9.14e+14	-9.14e+14
Tags_In confusion whether part time or DLP	-1.231e+15	3.04e+07	-4.05e+07	0.000	-1.23e+15	-1.23e+15
Tags_Interested in full time MBA	-1.123e+15	8.87e+06	-1.27e+08	0.000	-1.12e+15	-1.12e+15
Tags_Interested in Next batch	9.227e+14	3.92e+07	2.35e+07	0.000	9.23e+14	9.23e+14
Tags_Interested in other courses	-1.13e+15	5.13e+06	-2.2e+08	0.000	-1.13e+15	-1.13e+15
Tags_Lateral student	3.688e+15	4.79e+07	7.7e+07	0.000	3.69e+15	3.69e+15
Tags_Lost to EINS	9.902e+14	7.42e+06	1.33e+08	0.000	9.9e+14	9.9e+14
Tags_Lost to Others	-2.983e+15	3.08e+07	-9.7e+07	0.000	-2.98e+15	-2.98e+15
Tags_Not doing further education	-1.377e+15	8.38e+06	-1.64e+08	0.000	-1.38e+15	-1.38e+15
Tags_Recognition issue (DEC approval)	-4.238e+15	6.89e+07	-6.15e+07	0.000	-4.24e+15	-4.24e+15
Tags_Ringing	-1.762e+15	4.4e+06	-4.01e+08	0.000	-1.76e+15	-1.76e+15
Tags_Shall take in the next coming month	3.632e+15	6.78e+07	5.36e+07	0.000	3.63e+15	3.63e+15
Tags_Still Thinking	-2.847e+15	3.42e+07	-8.33e+07	0.000	-2.85e+15	-2.85e+15
Tags_University not recognized	-3.87e+15	4.79e+07	-8.07e+07	0.000	-3.87e+15	-3.87e+15
Tags_Want to take admission but has financial problems	-2.892e+14	4.15e+07	-6.97e+06	0.000	-2.89e+14	-2.89e+14
Tags_Will revert after reading the email	3.44e+14	5.07e+06	6.79e+07	0.000	3.44e+14	3.44e+14
Tags_in touch with EINS	-3.636e+14	2.42e+07	-1.5e+07	0.000	-3.64e+14	-3.64e+14
Tags_invalid number	-3.652e+15	9.98e+06	-3.66e+08	0.000	-3.65e+15	-3.65e+15
Tags_number not provided	-4.137e+15	1.66e+07	-2.5e+08	0.000	-4.14e+15	-4.14e+15
Tags_opp hangup	-1.816e+15	1.62e+07	-1.12e+08	0.000	-1.82e+15	-1.82e+15
Tags_switched off	-1.938e+15	6.61e+06	-2.93e+08	0.000	-1.94e+15	-1.94e+15
Tags_wrong number given	-2.886e+15	1.27e+07	-2.27e+08	0.000	-2.89e+15	-2.89e+15
Lead Profile_Dual Specialization Student	9.514e+14	2.16e+07	4.4e+07	0.000	9.51e+14	9.51e+14
Lead Profile_Lateral Student	1.638e+15	1.79e+07	9.16e+07	0.000	1.64e+15	1.64e+15
Lead Profile_Other Leads	2.709e+14	4.7e+06	5.76e+07	0.000	2.71e+14	2.71e+14
Lead Profile_Potential Lead	2.365e+14	3.28e+06	7.21e+07	0.000	2.37e+14	2.37e+14

Lead Profile_Student of SomeSchool	-7.825e+13	8.03e+06	-9.75e+06	0.000	-7.83e+13	-7.83e+13
What is your current occupation_Businessman	2.694e+14	4.82e+07	5.59e+06	0.000	2.69e+14	2.69e+14
What is your current occupation_Housewife	3.808e+15	2.45e+07	1.55e+08	0.000	3.81e+15	3.81e+15
What is your current occupation_Other	7.49e+14	1.95e+07	3.85e+07	0.000	7.49e+14	7.49e+14
What is your current occupation_Student	1.077e+15	7.46e+06	1.44e+08	0.000	1.08e+15	1.08e+15
What is your current occupation_Unemployed	1.132e+15	4.32e+06	2.62e+08	0.000	1.13e+15	1.13e+15
What is your current occupation_Working Professional	1.289e+15	5.71e+06	2.26e+08	0.000	1.29e+15	1.29e+15
Specialization_Banking, Investment And Insurance	-3.572e+13	6.78e+06	-5.27e+06	0.000	-3.57e+13	-3.57e+13
Specialization_Business Administration	-1.639e+12	6.5e+06	-2.52e+05	0.000	-1.64e+12	-1.64e+12
Specialization_E-Business	-1.293e+14	1.29e+07	-1e+07	0.000	-1.29e+14	-1.29e+14
Specialization_E-COMMERCE	-1.32e+14	9.61e+06	-1.37e+07	0.000	-1.32e+14	-1.32e+14
Specialization_Finance Management	-1.27e+14	5.75e+06	-2.21e+07	0.000	-1.27e+14	-1.27e+14
Specialization_Healthcare Management	-1.92e+14	8.91e+06	-2.16e+07	0.000	-1.92e+14	-1.92e+14
Specialization_Hospitality Management	-2.093e+14	9.42e+06	-2.22e+07	0.000	-2.09e+14	-2.09e+14
Specialization_Human Resource Management	-1.421e+14	5.74e+06	-2.47e+07	0.000	-1.42e+14	-1.42e+14
Specialization_IT Projects Management	-1.772e+14	6.98e+06	-2.54e+07	0.000	-1.77e+14	-1.77e+14
Specialization_International Business	-1.779e+14	8.12e+06	-2.19e+07	0.000	-1.78e+14	-1.78e+14
Specialization_Marketing Management	-2.952e+13	5.67e+06	-5.2e+06	0.000	-2.95e+13	-2.95e+13
Specialization_Media and Advertising	-1.811e+14	7.95e+06	-2.28e+07	0.000	-1.81e+14	-1.81e+14
Specialization_Operations Management	-4.891e+13	6.22e+06	-7.86e+06	0.000	-4.89e+13	-4.89e+13
Specialization_Retail Management	-9.806e+13	1.02e+07	-9.65e+06	0.000	-9.81e+13	-9.81e+13
Specialization_Rural and Agribusiness	-3.467e+14	1.12e+07	-3.08e+07	0.000	-3.47e+14	-3.47e+14
Specialization_Select	-1.346e+14	4.18e+06	-3.22e+07	0.000	-1.35e+14	-1.35e+14
Specialization_Services Excellence	-5.492e+14	1.66e+07	-3.3e+07	0.000	-5.49e+14	-5.49e+14
Specialization_Supply Chain Management	-1.693e+14	6.74e+06	-2.51e+07	0.000	-1.69e+14	-1.69e+14
Specialization_Travel and Tourism	-3.621e+14	8.3e+06	-4.36e+07	0.000	-3.62e+14	-3.62e+14
City_Mumbai	-1.045e+14	4.61e+06	-2.27e+07	0.000	-1.05e+14	-1.05e+14
City_Other Cities	-1.11e+14	5.4e+06	-2.05e+07	0.000	-1.11e+14	-1.11e+14
City_Other Cities of Maharashtra	-4.272e+13	5.87e+06	-7.27e+06	0.000	-4.27e+13	-4.27e+13
City_Other Metro Cities	-2.093e+14	6.29e+06	-3.33e+07	0.000	-2.09e+14	-2.09e+14
City_Thane & Outskirts	-1.874e+14	5.26e+06	-3.56e+07	0.000	-1.87e+14	-1.87e+14
City_Tier II Cities	3.113e+14	1.1e+07	2.84e+07	0.000	3.11e+14	3.11e+14
Last Activity_Approached upfront	4.13e+15	2.92e+07	1.42e+08	0.000	4.13e+15	4.13e+15
Last Activity_Converted to Lead	-4.108e+13	1.06e+07	-3.88e+06	0.000	-4.11e+13	-4.11e+13
Last Activity_Email Bounced	-2.533e+14	1.17e+07	-2.16e+07	0.000	-2.53e+14	-2.53e+14
Last Activity_Email Link Clicked	4.042e+14	1.25e+07	3.24e+07	0.000	4.04e+14	4.04e+14
Last Activity_Email Marked Spam	1.99e+14	2.92e+07	6.8e+06	0.000	1.99e+14	1.99e+14
Last Activity_Email Opened	9.603e+13	9.92e+06	9.68e+06	0.000	9.6e+13	9.6e+13
Last Activity_Email Received	-2.717e+14	6.8e+07	-4e+06	0.000	-2.72e+14	-2.72e+14
Last Activity_Form Submitted on Website	3.056e+14	1.21e+07	2.52e+07	0.000	3.06e+14	3.06e+14
Last Activity_Had a Phone Conversation	2.999e+14	2.31e+07	1.3e+07	0.000	3e+14	3e+14
Last Activity_Olark Chat Conversation	-2.415e+13	1.01e+07	-2.4e+06	0.000	-2.41e+13	-2.41e+13
Last Activity_Page Visited on Website	1.142e+14	1.06e+07	1.08e+07	0.000	1.14e+14	1.14e+14
Last Activity_Resubscribed to emails	0	0	nan	nan	0	0
Last Activity_SMS Sent	3.005e+14	1e+07	3e+07	0.000	3e+14	3e+14
Last Activity_Unreachable	1.021e+14	1.43e+07	7.15e+06	0.000	1.02e+14	1.02e+14
Last Activity_Unsubscribed	7.078e+14	2.28e+07	3.1e+07	0.000	7.08e+14	7.08e+14
Last Activity_View in browser link Clicked	-5.949e+15	6.81e+07	-8.74e+07	0.000	-5.95e+15	-5.95e+15
Last Activity_Visited Booth in Tradeshow	-1.181e+14	6.9e+07	-1.71e+06	0.000	-1.18e+14	-1.18e+14

Step 7: Feature Selection Using RFE

In [91]:

```
from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression()
```

In [92]:

```
from sklearn.feature_selection import RFE
rfe = RFE(logreg, 20)             # running RFE with 20 variables as output
rfe = rfe.fit(X_train, y_train)
```

In [93]:

```
rfe.support_
```

Out[93]:

```
array([False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False,  True, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False,  True, False, False, False, False,
       False,  True,  True, False,  True,  True, False, False,  True,
       False,  True, False,  True, False,  True, False,  True, False,
       False, False, False,  True, False,  True,  True,  True,  True,
        True, False, False, False, False, False, False, False, False,
       False,  True,  True, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False,  True, False, False, False, False])
```

In [94]:

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

Out[94]:

```
[('Do Not Email', False, 9),
 ('Do Not Call', False, 160),
 ('TotalVisits', False, 87),
 ('Total Time Spent on Website', False, 13),
 ('Page Views Per Visit', False, 66),
 ('Search', False, 33),
 ('Newspaper Article', False, 150),
 ('X Education Forums', False, 151),
 ('Newspaper', False, 116),
 ('Digital Advertisement', False, 112),
 ('Through Recommendations', False, 128),
 ('A free copy of Mastering The Interview', False, 113),
 ('Country_Outside India', False, 96),
 ('Lead Source_Direct Traffic', False, 105),
 ('Lead Source_Facebook', False, 74),
 ('Lead Source_Google', False, 129),
 ('Lead Source_Live Chat', False, 142),
 ('Lead Source_NC_EDM', False, 31),
 ('Lead Source_Olark Chat', False, 12),
 ('Lead Source_Organic Search', False, 110),
 ('Lead Source_Pay per Click Ads', False, 146),
 ('Lead Source_Press_Release', False, 152),
 ('Lead Source_Reference', False, 61),
 ('Lead Source_Referral Sites', False, 72),
 ('Lead Source_Social Media', False, 148),
 ('Lead Source_WeLearn', False, 153),
```

```
('Lead Source_Welingak Website', False, 2),
('Lead Source_bing', False, 123),
('Lead Source_blog', False, 93),
('Lead Source_google', False, 90),
('Lead Source_testone', False, 138),
('Lead Source_welearnblog_Home', False, 100),
('Lead Source_youtubechannel', False, 158),
('Lead Origin_Landing Page Submission', False, 85),
('Lead Origin_Lead Add Form', False, 44),
('Lead Origin_Lead Import', False, 63),
('Last Notable Activity_Email Bounced', False, 48),
('Last Notable Activity_Email Link Clicked', False, 23),
('Last Notable Activity_Email Marked Spam', False, 98),
('Last Notable Activity_Email Opened', False, 140),
('Last Notable Activity_Email Received', False, 135),
('Last Notable Activity_Form Submitted on Website', False, 111),
('Last Notable Activity_Had a Phone Conversation', False, 46),
('Last Notable Activity_Modified', False, 24),
('Last Notable Activity_Olark Chat Conversation', False, 15),
('Last Notable Activity_Page Visited on Website', False, 117),
('Last Notable Activity_Resubscribed to emails', False, 159),
('Last Notable Activity_SMS Sent', False, 16),
('Last Notable Activity_Unreachable', False, 91),
('Last Notable Activity_Unsubscribed', False, 37),
('Last Notable Activity_View in browser link Clicked', False, 133),
('Country_Outside India', False, 114),
('Lead Source_Direct Traffic', False, 64),
('Lead Source_Facebook', False, 89),
('Lead Source_Google', False, 136),
('Lead Source_Live Chat', False, 143),
('Lead Source_NC_EDM', False, 20),
('Lead Source_Olark Chat', False, 86),
('Lead Source_Organic Search', False, 82),
('Lead Source_Pay per Click Ads', False, 145),
('Lead Source_Press_Release', False, 155),
('Lead Source_Reference', False, 47),
('Lead Source_Referral Sites', False, 84),
('Lead Source_Social Media', False, 149),
('Lead Source_WeLearn', False, 156),
('Lead Source_Welingak Website', True, 1),
('Lead Source_bing', False, 124),
('Lead Source_blog', False, 97),
('Lead Source_google', False, 94),
('Lead Source_testone', False, 139),
('Lead Source_welearnblog_Home', False, 107),
('Lead Source_youtubechannel', False, 161),
('Lead Origin_Landing Page Submission', False, 115),
('Lead Origin_Lead Add Form', False, 11),
('Lead Origin_Lead Import', False, 65),
('Last Notable Activity_Email Bounced', False, 39),
('Last Notable Activity_Email Link Clicked', False, 21),
('Last Notable Activity_Email Marked Spam', False, 106),
('Last Notable Activity_Email Opened', False, 127),
('Last Notable Activity_Email Received', False, 134),
('Last Notable Activity_Form Submitted on Website', False, 109),
('Last Notable Activity_Had a Phone Conversation', False, 58),
('Last Notable Activity_Modified', False, 3),
('Last Notable Activity_Olark Chat Conversation', False, 7),
('Last Notable Activity_Page Visited on Website', False, 108),
('Last Notable Activity_Resubscribed to emails', False, 154),
('Last Notable Activity_SMS Sent', False, 40),
('Last Notable Activity_Unreachable', False, 88),
('Last Notable Activity_Unsubscribed', False, 54),
('Last Notable Activity_View in browser link Clicked', False, 132),
('Lead Quality_High in Relevance', False, 29),
('Lead Quality_Low in Relevance', False, 102),
('Lead Quality_Might be', False, 43),
('Lead Quality_Not Sure', False, 60),
('Lead Quality_Worst', True, 1),
('Asymmetrique Profile Index_01.High', False, 75),
('Asymmetrique Profile Index_02.Medium', False, 104),
('Asymmetrique Profile Index_03.Low', False, 103),
('Asymmetrique Activity Index_01.High', False, 76),
('Asymmetrique Activity Index_02.Medium', False, 77),
('Asymmetrique Activity Index_03.Low', True, 1),
('Tags_Already a student', True, 1),
('Tags_Busy', False, 34),
```

```
 ('Tags_Closed by Horizzon', True, 1),
 ('Tags_Diploma holder (Not Eligible)', True, 1),
 ('Tags_Graduation in progress', False, 6),
 ('Tags_In confusion whether part time or DLP', False, 42),
 ('Tags_Interested  in full time MBA', True, 1),
 ('Tags_Interested in Next batch', False, 57),
 ('Tags_Interested in other courses', True, 1),
 ('Tags_Lateral student', False, 38),
 ('Tags_Lost to EINS', True, 1),
 ('Tags_Lost to Others', False, 41),
 ('Tags_Not doing further education', True, 1),
 ('Tags_Recognition issue (DEC approval)', False, 35),
 ('Tags_Ringing', True, 1),
 ('Tags_Shall take in the next coming month', False, 53),
 ('Tags_Still Thinking', False, 10),
 ('Tags_University not recognized', False, 45),
 ('Tags_Want to take admission but has financial problems', False, 32),
 ('Tags_Will revert after reading the email', True, 1),
 ('Tags_in touch with EINS', False, 59),
 ('Tags_invalid number', True, 1),
 ('Tags_number not provided', True, 1),
 ('Tags_opp hangup', True, 1),
 ('Tags_switched off', True, 1),
 ('Tags_wrong number given', True, 1),
 ('Lead Profile_Dual Specialization Student', False, 62),
 ('Lead Profile_Lateral Student', False, 14),
 ('Lead Profile_Other Leads', False, 19),
 ('Lead Profile_Potential Lead', False, 18),
 ('Lead Profile_Student of SomeSchool', False, 55),
 ('What is your current occupation_Businessman', False, 118),
 ('What is your current occupation_Housewife', False, 27),
 ('What is your current occupation_Other', False, 28),
 ('What is your current occupation_Student', False, 4),
 ('What is your current occupation_Unemployed', True, 1),
 ('What is your current occupation_Working Professional', True, 1),
 ('Specialization_Banking, Investment And Insurance', False, 73),
 ('Specialization_Business Administration', False, 69),
 ('Specialization_E-Business', False, 56),
 ('Specialization_E-COMMERCE', False, 99),
 ('Specialization_Finance Management', False, 80),
 ('Specialization_Healthcare Management', False, 126),
 ('Specialization_Hospitality Management', False, 78),
 ('Specialization_Human Resource Management', False, 79),
 ('Specialization_IT Projects Management', False, 131),
 ('Specialization_International Business', False, 137),
 ('Specialization_Marketing Management', False, 67),
 ('Specialization_Media and Advertising', False, 141),
 ('Specialization_Operations Management', False, 68),
 ('Specialization_Retail Management', False, 83),
 ('Specialization_Rural and Agribusiness', False, 119),
 ('Specialization_Select', False, 17),
 ('Specialization_Services Excellence', False, 70),
 ('Specialization_Supply Chain Management', False, 125),
 ('Specialization_Travel and Tourism', False, 30),
 ('City_Mumbai', False, 122),
 ('City_Other Cities', False, 95),
 ('City_Other Cities of Maharashtra', False, 81),
 ('City_Other Metro Cities', False, 52),
 ('City_Thane & Outskirts', False, 121),
 ('City_Tier II Cities', False, 26),
 ('Last Activity_Approached upfront', False, 71),
 ('Last Activity_Converted to Lead', False, 50),
 ('Last Activity_Email Bounced', False, 49),
 ('Last Activity_Email Link Clicked', False, 22),
 ('Last Activity_Email Marked Spam', False, 101),
 ('Last Activity_Email Opened', False, 144),
 ('Last Activity_Email Received', False, 120),
 ('Last Activity_Form Submitted on Website', False, 25),
 ('Last Activity_Had a Phone Conversation', False, 5),
 ('Last Activity_Olark Chat Conversation', False, 36),
 ('Last Activity_Page Visited on Website', False, 51),
 ('Last Activity_Resubscribed to emails', False, 157),
 ('Last Activity_SMS Sent', True, 1),
 ('Last Activity_Unreachable', False, 92),
 ('Last Activity_Unsubscribed', False, 8),
 ('Last Activity_View in browser link Clicked', False, 130),
 ('Last Activity_Visited Booth in Tradeshow', False, 147)]
```

In [95]:

```
col = X_train.columns[rfe.support_]
col
```

Out[95]:

```
Index(['Lead Source_Welingak Website', 'Lead Quality_Worst',
       'Asymmetrique Activity Index_03.Low', 'Tags_Already a student',
       'Tags_Closed by Horizzon', 'Tags_Diploma holder (Not Eligible)',
       'Tags_Interested  in full time MBA', 'Tags_Interested in other courses',
       'Tags_Lost to EINS', 'Tags_Not doing further education', 'Tags_Ringing',
       'Tags_Will revert after reading the email', 'Tags_invalid number',
       'Tags_number not provided', 'Tags_opp hangup', 'Tags_switched off',
       'Tags_wrong number given', 'What is your current occupation_Unemployed',
       'What is your current occupation_Working Professional',
       'Last Activity_SMS Sent'],
      dtype='object')
```

In [96]:

```
X_train.columns[~rfe.support_]
```

Out[96]:

```
Index(['Do Not Email', 'Do Not Call', 'TotalVisits',
       'Total Time Spent on Website', 'Page Views Per Visit', 'Search',
       'Newspaper Article', 'X Education Forums', 'Newspaper',
       'Digital Advertisement',
       ...
       'Last Activity_Email Received',
       'Last Activity_Form Submitted on Website',
       'Last Activity_Had a Phone Conversation',
       'Last Activity_Olark Chat Conversation',
       'Last Activity_Page Visited on Website',
       'Last Activity_Resubscribed to emails', 'Last Activity_Unreachable',
       'Last Activity_Unsubscribed',
       'Last Activity_View in browser link Clicked',
       'Last Activity_Visited Booth in Tradeshow'],
      dtype='object', length=160)
```

In [97]:

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

Out[97]:

Generalized Linear Model Regression Results

Dep. Variable:	Converted	**No. Observations:**	6002
Model:	GLM	**Df Residuals:**	5981
Model Family:	Binomial	**Df Model:**	20
Link Function:	logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-1264.7
Date:	Wed, 13 May 2020	**Deviance:**	2529.4
Time:	11:48:16	**Pearson chi2:**	8.56e+03
No. Iterations:	24		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-2.4929	0.090	-27.836	0.000	-2.668	-2.317
Lead Source_Welingak Website	1.6140	0.366	4.414	0.000	0.897	2.331

Lead Source_Welingak Website	1.6140	0.366	4.414	0.000	0.897	2.331
Lead Quality_Worst	-2.5504	0.761	-3.354	0.001	-4.041	-1.060
Asymmetrique Activity Index_03.Low	-2.4592	0.358	-6.869	0.000	-3.161	-1.758
Tags_Already a student	-3.8785	0.726	-5.344	0.000	-5.301	-2.456
Tags_Closed by Horizzon	5.1421	0.722	7.120	0.000	3.727	6.558
Tags_Diploma holder (Not Eligible)	-24.1871	2.82e+04	-0.001	0.999	-5.52e+04	5.52e+04
Tags_Interested in full time MBA	-3.0545	0.742	-4.117	0.000	-4.509	-1.600
Tags_Interested in other courses	-3.0288	0.330	-9.183	0.000	-3.675	-2.382
Tags_Lost to EINS	6.3792	0.831	7.677	0.000	4.751	8.008
Tags_Not doing further education	-3.7904	1.032	-3.674	0.000	-5.813	-1.768
Tags_Ringing	-4.2659	0.249	-17.107	0.000	-4.755	-3.777
Tags_Will revert after reading the email	3.5963	0.194	18.561	0.000	3.217	3.976
Tags_invalid number	-25.7192	2.7e+04	-0.001	0.999	-5.3e+04	5.29e+04
Tags_number not provided	-25.9733	4.5e+04	-0.001	1.000	-8.82e+04	8.82e+04
Tags_opp hangup	-3.5152	1.063	-3.308	0.001	-5.598	-1.433
Tags_switched off	-5.1620	0.724	-7.126	0.000	-6.582	-3.742
Tags_wrong number given	-26.1206	3.49e+04	-0.001	0.999	-6.84e+04	6.84e+04
What is your current occupation_Unemployed	2.0649	0.119	17.357	0.000	1.832	2.298
What is your current occupation_Working Professional	2.1458	0.364	5.903	0.000	1.433	2.858
Last Activity_SMS Sent	2.0390	0.112	18.174	0.000	1.819	2.259

In [98]:

```
y_train_pred = res.predict(X_train_sm)
y_train_pred[:10]
```

Out[98]:

```
8529    0.065692
7331    0.009069
7688    0.833555
92      0.076360
4908    0.076360
451     0.009069
4945    0.009069
2844    0.994975
4355    0.076360
7251    0.001051
dtype: float64
```

In [99]:

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

Out[99]:

```
array([0.06569164, 0.00906869, 0.83355546, 0.07635965, 0.07635965,
       0.00906869, 0.00906869, 0.99497496, 0.07635965, 0.00105118])
```

In [100]:

```
y_train_pred_final = pd.DataFrame({'Converted':y_train.values, 'Conversion_Prob':y_train_pred})
y_train_pred_final['LeadID'] = y_train.index
y_train_pred_final.head()
```

Out[100]:

	Converted	Conversion_Prob	LeadID
0	0	0.065692	8529

	Converted	Conversion_Prob	LeadID
1	0	0.009069	7331
2	1	0.833555	7688
3	0	0.076360	92
4	0	0.076360	4908

In [101]:

```
y_train_pred_final['predicted'] = y_train_pred_final.Conversion_Prob.map(lambda x: 1 if x > 0.5 els
e 0)

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

Out[101]:

	Converted	Conversion_Prob	LeadID	predicted
0	0	0.065692	8529	0
1	0	0.009069	7331	0
2	1	0.833555	7688	1
3	0	0.076360	92	0
4	0	0.076360	4908	0

In [102]:

```
from sklearn import metrics
```

In [103]:

```
# Confusion matrix
confusion = metrics.confusion_matrix(y_train_pred_final.Converted, y_train_pred_final.predicted )
print(confusion)
```

```
[[3647   89]
 [ 409 1857]]
```

In [104]:

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

```
0.9170276574475175
```

Checking VIFs

In [105]:

```
from statsmodels.stats.outliers_influence import variance_inflation_factor
```

In [106]:

```
vif = pd.DataFrame()
vif['Features'] = X_train[col].columns
vif['VIF'] = [variance_inflation_factor(X_train[col].values, i) for i in range(X_train[col].shape[1
])]
vif['VIF'] = round(vif['VIF'], 2)
vif = vif.sort_values(by = "VIF", ascending = False)
vif
```

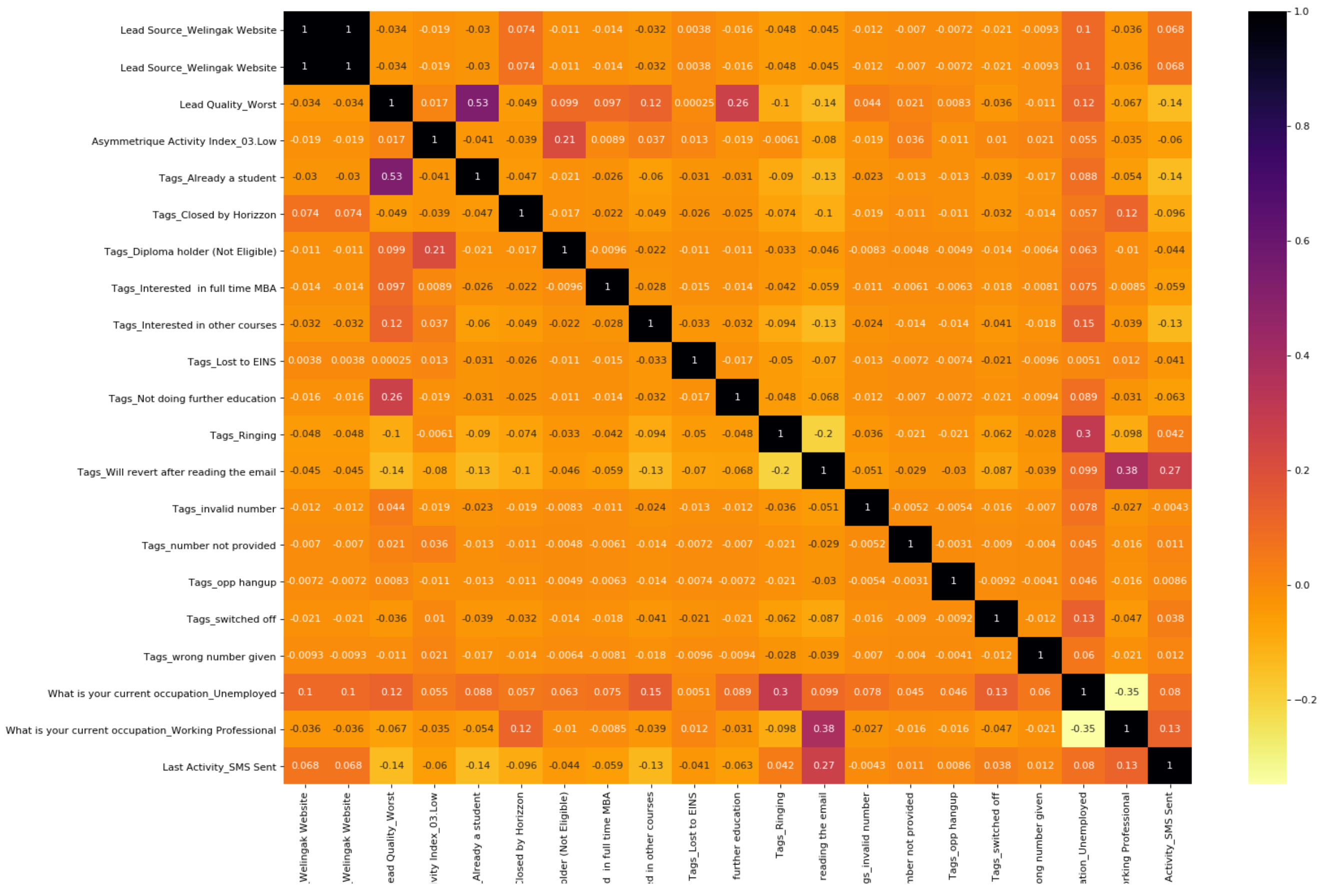
Out[106]:

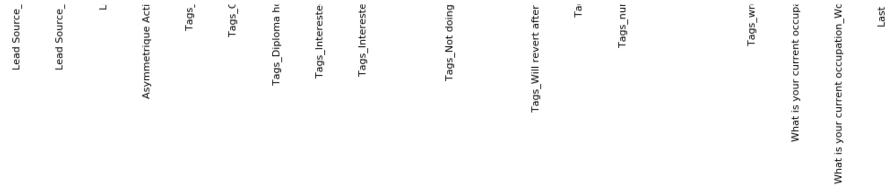
	Features	VIF
0	Lead Source_Welingak Website	inf

	Features	VIF
1	Lead Source_Welingak Website	inf
5	Tags_Closed by Horizzon	1.30
10	Tags_Not doing further education	1.27
16	Tags_switched off	1.20
6	Tags_Diploma holder (Not Eligible)	1.12
7	Tags_Interested in full time MBA	1.12
3	Asymmetrique Activity Index_03.Low	1.11
13	Tags_invalid number	1.08
9	Tags_Lost to EINS	1.07
17	Tags_wrong number given	1.04
14	Tags_number not provided	1.03
15	Tags_opp hangup	1.03
19	What is your current occupation_Working Profes...	0.80
2	Lead Quality_Worst	0.69
11	Tags_Ringing	0.62
8	Tags_Interested in other courses	0.40
4	Tags_Already a student	0.38
12	Tags_Will revert after reading the email	0.09
18	What is your current occupation_Unemployed	0.01
20	Last Activity_SMS Sent	0.00

In [107]:

```
plt.figure(figsize=(20,15), dpi=80, facecolor='w', edgecolor='k', frameon='True')

cor = X_train[col].corr()
sns.heatmap(cor, annot=True, cmap="inferno_r")

plt.tight_layout()
plt.show()
```





In [108]:

```
col = col.drop('Tags_number not provided', 1)
col
```

Out[108]:

```
Index(['Lead Source_Welingak Website', 'Lead Quality_Worst',
       'Asymmetrique Activity Index_03.Low', 'Tags_Already a student',
       'Tags_Closed by Horizzon', 'Tags_Diploma holder (Not Eligible)',
       'Tags_Interested  in full time MBA', 'Tags_Interested in other courses',
       'Tags_Lost to EINS', 'Tags_Not doing further education', 'Tags_Ringing',
       'Tags_Will revert after reading the email', 'Tags_invalid number',
       'Tags_opp hangup', 'Tags_switched off', 'Tags_wrong number given',
       'What is your current occupation_Unemployed',
       'What is your current occupation_Working Professional',
       'Last Activity_SMS Sent'],
      dtype='object')
```

In [109]:

```
X_train_sm = sm.add_constant(X_train[col])
logm3 = sm.GLM(y_train,X_train_sm, family = sm.families.Binomial())
res = logm3.fit()
res.summary()
```

Out[109]:

Generalized Linear Model Regression Results

Dep. Variable:	Converted	**No. Observations:**	6002
Model:	GLM	**Df Residuals:**	5982
Model Family:	Binomial	**Df Model:**	19
Link Function:	logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-1278.7
Date:	Wed, 13 May 2020	**Deviance:**	2557.4
Time:	11:48:48	**Pearson chi2:**	8.49e+03
No. Iterations:	24		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-2.4804	0.089	-27.881	0.000	-2.655	-2.306
Lead Source_Welingak Website	1.6459	0.366	4.503	0.000	0.929	2.362
Lead Source_Welingak Website	1.6459	0.366	4.503	0.000	0.929	2.362
Lead Quality_Worst	-2.7112	0.739	-3.668	0.000	-4.160	-1.263
Asymmetrique Activity Index_03.Low	-2.4342	0.357	-6.817	0.000	-3.134	-1.734
Tags_Already a student	-3.8015	0.724	-5.247	0.000	-5.221	-2.382
Tags_Closed by Horizzon	5.1851	0.722	7.184	0.000	3.770	6.600
Tags_Diploma holder (Not Eligible)	-24.1120	2.81e+04	-0.001	0.999	-5.51e+04	5.51e+04
Tags_Interested in full time MBA	-2.9855	0.741	-4.028	0.000	-4.438	-1.533
Tags_Interested in other courses	-2.9603	0.329	-8.996	0.000	-3.605	-2.315
Tags_Lost to EINS	6.4382	0.838	7.684	0.000	4.796	8.080
Tags_Not doing further education	-3.7070	1.031	-3.596	0.000	-5.727	-1.687
Tags_Ringing	-4.1829	0.248	-16.855	0.000	-4.669	-3.696

Tags_Will revert after reading the email	3.6368	0.193	18.834	0.000	3.258	4.015
Tags_invalid number	-25.6348	2.7e+04	-0.001	0.999	-5.3e+04	5.29e+04
Tags_opp hangup	-3.4305	1.062	-3.231	0.001	-5.512	-1.349
Tags_switched off	-5.0770	0.724	-7.013	0.000	-6.496	-3.658
Tags_wrong number given	-26.0375	3.49e+04	-0.001	0.999	-6.85e+04	6.84e+04
What is your current occupation_Unemployed	1.9949	0.118	16.969	0.000	1.764	2.225
What is your current occupation_Working Professional	2.1030	0.363	5.788	0.000	1.391	2.815
Last Activity_SMS Sent	2.0063	0.111	18.069	0.000	1.789	2.224

In [110]:

```
y_train_pred = res.predict(X_train_sm)
y_train_pred[:10]
```

Out[110]:

```
8529    0.065249
7331    0.009300
7688    0.820658
92      0.077242
4908    0.077242
451     0.009300
4945    0.009300
2844    0.994861
4355    0.077242
7251    0.000913
dtype: float64
```

In [111]:

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

Out[111]:

```
array([6.52492255e-02, 9.29987842e-03, 8.20658174e-01, 7.72422324e-02,
       7.72422324e-02, 9.29987842e-03, 9.29987842e-03, 9.94861183e-01,
       7.72422324e-02, 9.12704851e-04])
```

In [112]:

```
y_train_pred_final = pd.DataFrame({'Converted':y_train.values, 'Conversion_Prob':y_train_pred})
y_train_pred_final['LeadID'] = y_train.index
y_train_pred_final.head()
```

Out[112]:

	Converted	Conversion_Prob	LeadID
0	0	0.065249	8529
1	0	0.009300	7331
2	1	0.820658	7688
3	0	0.077242	92
4	0	0.077242	4908

In [113]:

```
y_train_pred_final['predicted'] = y_train_pred_final.Conversion_Prob.map(lambda x: 1 if x > 0.5 els
e 0)

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

Out[113]:

	Converted	Conversion_Prob	LeadID	predicted
0	0	0.065249	8529	0
1	0	0.009300	7331	0
2	1	0.820658	7688	1
3	0	0.077242	92	0
4	0	0.077242	4908	0

In [114]:

```
from sklearn import metrics
```

In [115]:

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

```
[[3641   95]
 [ 409 1857]]
```

In [116]:

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

```
0.9160279906697767
```

Checking VIFs

In [117]:

```
from statsmodels.stats.outliers_influence import variance_inflation_factor
```

In [118]:

```
vif = pd.DataFrame()
vif['Features'] = X_train[col].columns
vif['VIF'] = [variance_inflation_factor(X_train[col].values, i) for i in range(X_train[col].shape[1
])]
vif['VIF'] = round(vif['VIF'], 2)
vif = vif.sort_values(by = "VIF", ascending = False)
vif
```

Out[118]:

	Features	VIF
0	Lead Source_Welingak Website	inf
1	Lead Source_Welingak Website	inf
5	Tags_Closed by Horizzon	1.29
10	Tags_Not doing further education	1.27
15	Tags_switched off	1.19
6	Tags_Diploma holder (Not Eligible)	1.12
7	Tags_Interested in full time MBA	1.12
3	Asymmetrique Activity Index_03.Low	1.11
13	Tags_invalid number	1.08
9	Tags_Lost to EINS	1.07
16	Tags_wrong number given	1.04
14	Tags_opp hangup	1.03

	Features	VIF
18	What is your current occupation_Working Profes...	0.79
2	Lead Quality_Worst	0.69
11	Tags_Ringing	0.62
8	Tags_Interested in other courses	0.39
4	Tags_Already a student	0.38
12	Tags_Will revert after reading the email	0.09
17	What is your current occupation_Unemployed	0.01
19	Last Activity_SMS Sent	0.00

In [119]:

```
col = col.drop('Tags_wrong number given', 1)
col
```

Out[119]:

```
Index(['Lead Source_Welingak Website', 'Lead Quality_Worst',
       'Asymmetrique Activity Index_03.Low', 'Tags_Already a student',
       'Tags_Closed by Horizzon', 'Tags_Diploma holder (Not Eligible)',
       'Tags_Interested  in full time MBA', 'Tags_Interested in other courses',
       'Tags_Lost to EINS', 'Tags_Not doing further education', 'Tags_Ringing',
       'Tags_Will revert after reading the email', 'Tags_invalid number',
       'Tags_opp hangup', 'Tags_switched off',
       'What is your current occupation_Unemployed',
       'What is your current occupation_Working Professional',
       'Last Activity_SMS Sent'],
      dtype='object')
```

In [120]:

```
X_train_sm = sm.add_constant(X_train[col])
logm4 = sm.GLM(y_train,X_train_sm, family = sm.families.Binomial())
res = logm4.fit()
res.summary()
```

Out[120]:

Generalized Linear Model Regression Results

Dep. Variable:	Converted	**No. Observations:**	6002
Model:	GLM	**Df Residuals:**	5983
Model Family:	Binomial	**Df Model:**	18
Link Function:	logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-1305.1
Date:	Wed, 13 May 2020	**Deviance:**	2610.1
Time:	11:49:00	**Pearson chi2:**	8.25e+03
No. Iterations:	23		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-2.4653	0.088	-27.969	0.000	-2.638	-2.293
Lead Source_Welingak Website	1.7080	0.365	4.676	0.000	0.992	2.424
Lead Source_Welingak Website	1.7080	0.365	4.676	0.000	0.992	2.424
Lead Quality_Worst	-2.7568	0.728	-3.787	0.000	-4.184	-1.330
Asymmetrique Activity Index_03.Low	-2.3688	0.357	-6.637	0.000	-3.068	-1.669
Tags_Already a student	-3.6760	0.724	-5.080	0.000	-5.094	-2.258
Tags_Closed by Horizzon	5.2742	0.721	7.314	0.000	3.861	6.687
Tags_Diploma holder (Not Eligible)	-22.9881	1.71e+04	-0.001	0.999	-3.35e+04	3.35e+04
Tags_Interested in full time MBA	-2.8602	0.740	-3.866	0.000	-4.310	-1.410

Tags_Interested in other courses	-2.8332	0.328	-8.641	0.000	-3.476	-2.191
Tags_Lost to EINS	6.4558	0.839	7.692	0.000	4.811	8.101
Tags_Not doing further education	-3.5698	1.030	-3.467	0.001	-5.588	-1.552
Tags_Ringing	-4.0320	0.246	-16.378	0.000	-4.515	-3.550
Tags_Will revert after reading the email	3.7184	0.192	19.386	0.000	3.342	4.094
Tags_invalid number	-24.4886	1.64e+04	-0.001	0.999	-3.22e+04	3.21e+04
Tags_opp hangup	-3.2794	1.061	-3.092	0.002	-5.358	-1.201
Tags_switched off	-4.9237	0.723	-6.809	0.000	-6.341	-3.506
What is your current occupation_Unemployed	1.8623	0.115	16.189	0.000	1.637	2.088
What is your current occupation_Working Professional	2.0226	0.363	5.570	0.000	1.311	2.734
Last Activity_SMS Sent	1.9628	0.109	17.982	0.000	1.749	2.177

In [121]:

```
y_train_pred = res.predict(X_train_sm)
y_train_pred[:10]
```

Out[121]:

```
8529    0.064635
7331    0.009613
7688    0.795734
92      0.078329
4908    0.078329
451     0.009613
4945    0.009613
2844    0.994720
4355    0.078329
7251    0.000879
dtype: float64
```

In [122]:

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

Out[122]:

```
array([6.46349739e-02, 9.61261677e-03, 7.95733870e-01, 7.83285731e-02,
       7.83285731e-02, 9.61261677e-03, 9.61261677e-03, 9.94720023e-01,
       7.83285731e-02, 8.79091579e-04])
```

In [123]:

```
y_train_pred_final = pd.DataFrame({'Converted':y_train.values, 'Conversion_Prob':y_train_pred})
y_train_pred_final['LeadID'] = y_train.index
y_train_pred_final.head()
```

Out[123]:

	Converted	Conversion_Prob	LeadID
0	0	0.064635	8529
1	0	0.009613	7331
2	1	0.795734	7688
3	0	0.078329	92
4	0	0.078329	4908

In [124]:

```
y_train_pred_final['predicted'] = y_train_pred_final.Conversion_Prob.map(lambda x: 1 if x > 0.5 els
e 0)
```

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

Out[124]:

	Converted	Conversion_Prob	LeadID	predicted
0	0	0.064635	8529	0
1	0	0.009613	7331	0
2	1	0.795734	7688	1
3	0	0.078329	92	0
4	0	0.078329	4908	0

In [125]:

```
from sklearn import metrics
```

In [126]:

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

```
[[3630  106]
 [ 409 1857]]
```

In [127]:

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

```
0.9141952682439187
```

In [128]:

```
from statsmodels.stats.outliers_influence import variance_inflation_factor
```

In [129]:

```
vif = pd.DataFrame()
vif['Features'] = X_train[col].columns
vif['VIF'] = [variance_inflation_factor(X_train[col].values, i) for i in range(X_train[col].shape[1
])]
vif['VIF'] = round(vif['VIF'], 2)
vif = vif.sort_values(by = "VIF", ascending = False)
vif
```

Out[129]:

	Features	VIF
0	Lead Source_Welingak Website	inf
1	Lead Source_Welingak Website	inf
5	Tags_Closed by Horizzon	1.29
10	Tags_Not doing further education	1.26
15	Tags_switched off	1.19
6	Tags_Diploma holder (Not Eligible)	1.12
7	Tags_Interested in full time MBA	1.12
3	Asymmetrique Activity Index_03.Low	1.11
13	Tags_invalid number	1.08
9	Tags_Lost to EINS	1.06
14	Tags_opp hangup	1.02
17	What is your current occupation_Working Profes...	0.79

	Features	VIF
	Profes...	
2	~~Lead Quality_Worst~~	~~0.69~~
11	Tags_Ringing	0.61
8	Tags_Interested in other courses	0.39
4	Tags_Already a student	0.38
12	Tags_Will revert after reading the email	0.09
16	What is your current occupation_Unemployed	0.01
18	Last Activity_SMS Sent	0.00

In [130]:

```
col = col.drop('Tags_Diploma holder (Not Eligible)', 1)
col
```

Out[130]:

```
Index(['Lead Source_Welingak Website', 'Lead Quality_Worst',
       'Asymmetrique Activity Index_03.Low', 'Tags_Already a student',
       'Tags_Closed by Horizzon', 'Tags_Interested  in full time MBA',
       'Tags_Interested in other courses', 'Tags_Lost to EINS',
       'Tags_Not doing further education', 'Tags_Ringing',
       'Tags_Will revert after reading the email', 'Tags_invalid number',
       'Tags_opp hangup', 'Tags_switched off',
       'What is your current occupation_Unemployed',
       'What is your current occupation_Working Professional',
       'Last Activity_SMS Sent'],
      dtype='object')
```

In [131]:

```
X_train_sm = sm.add_constant(X_train[col])
logm5 = sm.GLM(y_train,X_train_sm, family = sm.families.Binomial())
res = logm5.fit()
res.summary()
```

Out[131]:

Generalized Linear Model Regression Results

Dep. Variable:	Converted	**No. Observations:**	6002
Model:	GLM	**Df Residuals:**	5984
Model Family:	Binomial	**Df Model:**	17
Link Function:	logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-1313.2
Date:	Wed, 13 May 2020	**Deviance:**	2626.4
Time:	11:49:12	**Pearson chi2:**	8.42e+03
No. Iterations:	23		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-2.4750	0.088	-28.020	0.000	-2.648	-2.302
Lead Source_Welingak Website	1.7339	0.365	4.747	0.000	1.018	2.450
Lead Source_Welingak Website	1.7339	0.365	4.747	0.000	1.018	2.450
Lead Quality_Worst	-2.8883	0.706	-4.092	0.000	-4.272	-1.505
Asymmetrique Activity Index_03.Low	-2.4330	0.351	-6.931	0.000	-3.121	-1.745
Tags_Already a student	-3.6149	0.723	-4.999	0.000	-5.032	-2.198
Tags_Closed by Horizzon	5.3212	0.721	7.382	0.000	3.908	6.734
Tags_Interested in full time MBA	-2.8081	0.740	-3.794	0.000	-4.259	-1.357
Tags_Interested in other courses	-2.7838	0.328	-8.493	0.000	-3.426	-2.141
Tags_Lost to EINS	6.5606	0.846	7.757	0.000	4.903	8.218

Tags_Not doing further education	-3.5144	1.030	-3.412	0.001	-5.533	-1.496
Tags_Ringing	-3.9921	0.246	-16.235	0.000	-4.474	-3.510
Tags_Will revert after reading the email	3.7631	0.192	19.646	0.000	3.388	4.138
Tags_invalid number	-24.4442	1.64e+04	-0.001	0.999	-3.22e+04	3.21e+04
Tags_opp hangup	-3.2379	1.061	-3.052	0.002	-5.317	-1.159
Tags_switched off	-4.8845	0.723	-6.756	0.000	-6.302	-3.467
What is your current occupation_Unemployed	1.8184	0.114	15.893	0.000	1.594	2.043
What is your current occupation_Working Professional	1.9876	0.362	5.486	0.000	1.277	2.698
Last Activity_SMS Sent	1.9808	0.109	18.198	0.000	1.767	2.194

In [132]:

```
y_train_pred = res.predict(X_train_sm)
y_train_pred[:10]
```

Out[132]:

```
8529    0.064888
7331    0.009483
7688    0.789866
92      0.077629
4908    0.077629
451     0.009483
4945    0.009483
2844    0.994813
4355    0.077629
7251    0.000777
dtype: float64
```

In [133]:

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

Out[133]:

```
array([6.48878261e-02, 9.48266404e-03, 7.89866093e-01, 7.76292105e-02,
       7.76292105e-02, 9.48266404e-03, 9.48266404e-03, 9.94812863e-01,
       7.76292105e-02, 7.76508332e-04])
```

In [134]:

```
y_train_pred_final = pd.DataFrame({'Converted':y_train.values, 'Conversion_Prob':y_train_pred})
y_train_pred_final['LeadID'] = y_train.index
y_train_pred_final.head()
```

Out[134]:

	Converted	Conversion_Prob	LeadID
0	0	0.064888	8529
1	0	0.009483	7331
2	1	0.789866	7688
3	0	0.077629	92
4	0	0.077629	4908

In [135]:

```
y_train_pred_final['predicted'] = y_train_pred_final.Conversion_Prob.map(lambda x: 1 if x > 0.5 els
e 0)

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

Out[135]:

	Converted	Conversion_Prob	LeadID	predicted
0	0	0.064888	8529	0
1	0	0.009483	7331	0
2	1	0.789866	7688	1
3	0	0.077629	92	0
4	0	0.077629	4908	0

In [136]:

```
from sklearn import metrics
```

In [137]:

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

```
[[3629  107]
 [ 409 1857]]
```

In [138]:

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

```
0.9140286571142953
```

In [139]:

```
from statsmodels.stats.outliers_influence import variance_inflation_factor
```

In [140]:

```
vif = pd.DataFrame()
vif['Features'] = X_train[col].columns
vif['VIF'] = [variance_inflation_factor(X_train[col].values, i) for i in range(X_train[col].shape[1
])]
vif['VIF'] = round(vif['VIF'], 2)
vif = vif.sort_values(by = "VIF", ascending = False)
vif
```

Out[140]:

	Features	VIF
0	Lead Source_Welingak Website	inf
1	Lead Source_Welingak Website	inf
5	Tags_Closed by Horizzon	1.28
9	Tags_Not doing further education	1.25
14	Tags_switched off	1.18
6	Tags_Interested in full time MBA	1.11
3	Asymmetrique Activity Index_03.Low	1.07
12	Tags_invalid number	1.07
8	Tags_Lost to EINS	1.06
13	Tags_opp hangup	1.02
16	What is your current occupation_Working Profes...	0.78
2	Lead Quality_Worst	0.67
10	Tags_Ringing	0.59

	Features	VIF
7	Tags_Interested in other courses	0.38
4	Tags_Already a student	0.37
11	Tags_Will revert after reading the email	0.09
15	What is your current occupation_Unemployed	0.01
17	Last Activity_SMS Sent	0.00

In [141]:

```
col = col.drop('Tags_invalid number', 1)
col
```

Out[141]:

```
Index(['Lead Source_Welingak Website', 'Lead Quality_Worst',
       'Asymmetrique Activity Index_03.Low', 'Tags_Already a student',
       'Tags_Closed by Horizzon', 'Tags_Interested  in full time MBA',
       'Tags_Interested in other courses', 'Tags_Lost to EINS',
       'Tags_Not doing further education', 'Tags_Ringing',
       'Tags_Will revert after reading the email', 'Tags_opp hangup',
       'Tags_switched off', 'What is your current occupation_Unemployed',
       'What is your current occupation_Working Professional',
       'Last Activity_SMS Sent'],
      dtype='object')
```

In [142]:

```
X_train_sm = sm.add_constant(X_train[col])
logm6 = sm.GLM(y_train,X_train_sm, family = sm.families.Binomial())
res = logm6.fit()
res.summary()
```

Out[142]:

Generalized Linear Model Regression Results

Dep. Variable:	Converted	**No. Observations:**	6002
Model:	GLM	**Df Residuals:**	5985
Model Family:	Binomial	**Df Model:**	16
Link Function:	logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-1342.4
Date:	Wed, 13 May 2020	**Deviance:**	2684.8
Time:	11:49:28	**Pearson chi2:**	8.52e+03
No. Iterations:	8		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-2.4751	0.088	-28.144	0.000	-2.647	-2.303
Lead Source_Welingak Website	1.8067	0.365	4.949	0.000	1.091	2.522
Lead Source_Welingak Website	1.8067	0.365	4.949	0.000	1.091	2.522
Lead Quality_Worst	-3.1794	0.670	-4.742	0.000	-4.494	-1.865
Asymmetrique Activity Index_03.Low	-2.3401	0.354	-6.605	0.000	-3.035	-1.646
Tags_Already a student	-3.4492	0.722	-4.776	0.000	-4.865	-2.034
Tags_Closed by Horizzon	5.4435	0.720	7.559	0.000	4.032	6.855
Tags_Interested in full time MBA	-2.6565	0.740	-3.591	0.000	-4.106	-1.207
Tags_Interested in other courses	-2.6347	0.327	-8.060	0.000	-3.275	-1.994
Tags_Lost to EINS	6.7102	0.862	7.786	0.000	5.021	8.399
Tags_Not doing further education	-3.3472	1.030	-3.250	0.001	-5.366	-1.329
Tags_Ringing	-3.8360	0.244	-15.709	0.000	-4.315	-3.357
Tags_Will revert after reading the email	3.8695	0.190	20.331	0.000	3.497	4.243

Tags_opp hangup	-3.0789	1.061	-2.903	0.004	-5.158	-1.000
Tags_switched off	-4.7274	0.722	-6.544	0.000	-6.143	-3.311
What is your current occupation_Unemployed	1.6711	0.112	14.926	0.000	1.452	1.891
What is your current occupation_Working Professional	1.8944	0.363	5.221	0.000	1.183	2.606
Last Activity_SMS Sent	1.9687	0.107	18.383	0.000	1.759	2.179

In [143]:

```
y_train_pred = res.predict(X_train_sm)
y_train_pred[:10]
```

Out[143]:

```
8529    0.064688
7331    0.009566
7688    0.762190
92      0.077626
4908    0.077626
451     0.009566
4945    0.009566
2844    0.994819
4355    0.077626
7251    0.000591
dtype: float64
```

In [144]:

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

Out[144]:

```
array([6.46881585e-02, 9.56568869e-03, 7.62190244e-01, 7.76256984e-02,
       7.76256984e-02, 9.56568869e-03, 9.56568869e-03, 9.94818870e-01,
       7.76256984e-02, 5.91337209e-04])
```

In [145]:

```
y_train_pred_final = pd.DataFrame({'Converted':y_train.values, 'Conversion_Prob':y_train_pred})
y_train_pred_final['LeadID'] = y_train.index
y_train_pred_final.head()
```

Out[145]:

	Converted	Conversion_Prob	LeadID
0	0	0.064688	8529
1	0	0.009566	7331
2	1	0.762190	7688
3	0	0.077626	92
4	0	0.077626	4908

In [146]:

```
y_train_pred_final['predicted'] = y_train_pred_final.Conversion_Prob.map(lambda x: 1 if x > 0.5 els
e 0)

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

Out[146]:

	Converted	Conversion_Prob	LeadID	predicted
0	0	0.064688	8529	0

	Converted	Conversion_Prob	LeadID	predicted
1	0	0.009566	7331	0
2	1	0.762190	7688	1
3	0	0.077626	92	0
4	0	0.077626	4908	0

In [147]:

```
from sklearn import metrics
```

In [148]:

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

```
[[3620  116]
 [ 409 1857]]
```

In [149]:

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

```
0.9125291569476841
```

In [150]:

```
from statsmodels.stats.outliers_influence import variance_inflation_factor
```

In [151]:

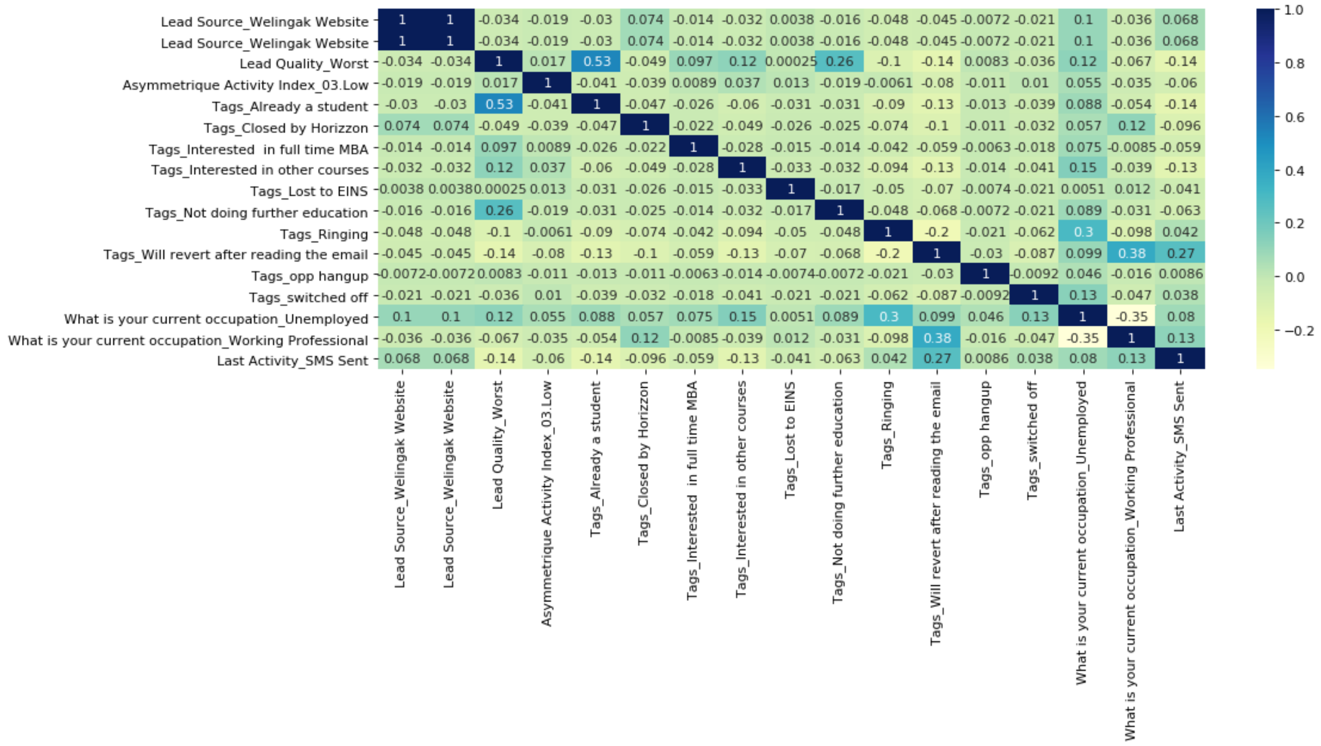
```
vif = pd.DataFrame()
vif['Features'] = X_train[col].columns
vif['VIF'] = [variance_inflation_factor(X_train[col].values, i) for i in range(X_train[col].shape[1
])]
vif['VIF'] = round(vif['VIF'], 2)
vif = vif.sort_values(by = "VIF", ascending = False)
vif
```

Out[151]:

	Features	VIF
0	Lead Source_Welingak Website	inf
1	Lead Source_Welingak Website	inf
5	Tags_Closed by Horizzon	1.26
9	Tags_Not doing further education	1.23
13	Tags_switched off	1.17
6	Tags_Interested in full time MBA	1.10
3	Asymmetrique Activity Index_03.Low	1.07
8	Tags_Lost to EINS	1.06
12	Tags_opp hangup	1.02
15	What is your current occupation_Working Profes...	0.77
2	Lead Quality_Worst	0.67
10	Tags_Ringing	0.58
7	Tags_Interested in other courses	0.38
4	Tags_Already a student	0.36
11	Tags_Will revert after reading the email	0.09
14	What is your current occupation_Unemployed	0.01
16	Last Activity_SMS Sent	0.00

In [152]:

```
plt.figure(figsize=(15,8), dpi=80, facecolor='w', edgecolor='k', frameon='True')

cor = X_train[col].corr()
sns.heatmap(cor, annot=True, cmap="YlGnBu")

plt.tight_layout()
plt.show()
```



Metrics beyond simply accuracy

In [153]:

```
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 [154]:

```
TP / float(TP+FN)
```

Out[154]:

```
0.8195057369814651
```

In [155]:

```
# Let us calculate specificity
TN / float(TN+FP)
```

Out[155]:

```
0.9689507494646681
```

In [156]:

```
# Calculate false postive rate - predicting churn when customer does not have churned
print(FP/ float(TN+FP))
```

```
0.031049250535331904
```

In [157]:

```
# positive predictive value
print (TP / float(TP+FP))
```

```
0.941206284845413
```

In [158]:

```
# Negative predictive value
print (TN / float(TN+ FN))
```

```
0.8984859766691486
```

Step 9: Plotting the ROC Curve

An ROC curve demonstrates several things:

It shows the tradeoff between sensitivity and specificity (any increase in sensitivity will be accompanied by a decrease in specificity). The closer the curve follows the left-hand border and then the top border of the ROC space, the more accurate the test. The closer the curve comes to the 45-degree diagonal of the ROC space, the less accurate the test.

In [159]:

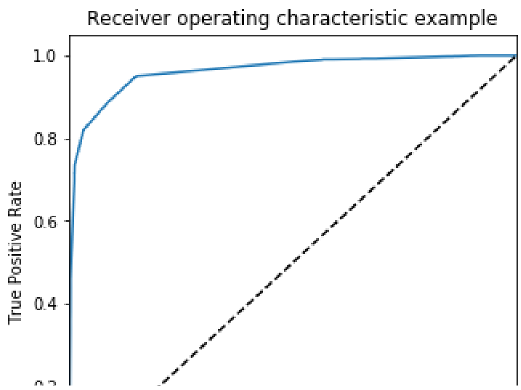
```
def draw_roc( actual, probs ):
    fpr, tpr, thresholds = metrics.roc_curve( actual, probs,
                                              drop_intermediate = False )
    auc_score = metrics.roc_auc_score( actual, probs )
    plt.figure(figsize=(5, 5))
    plt.plot( fpr, tpr, label='ROC curve (area = %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 fpr,tpr, thresholds
```

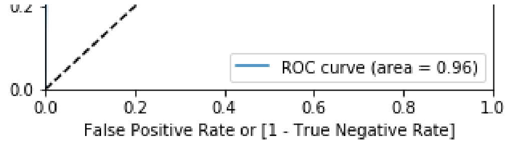
In [167]:

```
fpr, tpr, thresholds = metrics.roc_curve( y_train_pred_final.Converted, y_train_pred_final.Conversi
on_Prob, drop_intermediate = False )
```

In [168]:

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





Out[168]:

```
(array([0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
        0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
        0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 2.67665953e-04,
        2.67665953e-04, 2.67665953e-04, 5.35331906e-04, 8.02997859e-04,
        8.02997859e-04, 2.14132762e-03, 2.14132762e-03, 2.40899358e-03,
        2.40899358e-03, 2.67665953e-03, 2.67665953e-03, 2.67665953e-03,
        2.67665953e-03, 2.67665953e-03, 4.55032120e-03, 1.15096360e-02,
        1.15096360e-02, 1.15096360e-02, 1.15096360e-02, 1.15096360e-02,
        1.15096360e-02, 1.17773019e-02, 1.17773019e-02, 1.20449679e-02,
        1.20449679e-02, 3.07815846e-02, 3.07815846e-02, 3.07815846e-02,
        3.10492505e-02, 3.10492505e-02, 3.13169165e-02, 8.61884368e-02,
        8.69914347e-02, 1.45610278e-01, 1.45610278e-01, 1.45877944e-01,
        1.48019272e-01, 1.48019272e-01, 1.51498929e-01, 1.52301927e-01,
        1.53640257e-01, 1.55513919e-01, 1.56584582e-01, 1.57387580e-01,
        1.57655246e-01, 5.00802998e-01, 5.65578158e-01, 5.67184154e-01,
        5.67719486e-01, 5.87794433e-01, 5.90471092e-01, 5.91541756e-01,
        6.47216274e-01, 6.57655246e-01, 6.73715203e-01, 6.76659529e-01,
        6.76927195e-01, 6.84154176e-01, 6.84957173e-01, 6.85224839e-01,
        6.94057816e-01, 7.22965739e-01, 7.23233405e-01, 7.24036403e-01,
        8.46092077e-01, 8.46627409e-01, 8.46895075e-01, 8.57601713e-01,
        8.59475375e-01, 8.59743041e-01, 8.66167024e-01, 8.66434690e-01,
        8.67237687e-01, 8.69111349e-01, 8.91327623e-01, 8.96145610e-01,
        8.96948608e-01, 8.97483940e-01, 8.98822270e-01, 9.02034261e-01,
        9.02301927e-01, 9.02569593e-01, 9.18361884e-01, 9.23447537e-01,
        9.23715203e-01, 9.23982869e-01, 9.28533191e-01, 9.28800857e-01,
        9.29068522e-01, 9.29336188e-01, 9.29871520e-01, 9.42987152e-01,
        9.77783726e-01, 9.78319058e-01, 9.78586724e-01, 9.80192719e-01,
        9.80460385e-01, 9.80728051e-01, 9.80995717e-01, 9.81531049e-01,
        9.81798715e-01, 9.83137045e-01, 9.99464668e-01, 1.00000000e+00]),
 array([0.00000000e+00, 4.41306267e-04, 1.76522507e-03, 2.20653133e-03,
        2.64783760e-03, 7.50220653e-03, 9.26743160e-03, 1.23565755e-02,
        1.27978817e-02, 1.41218005e-02, 1.63283319e-02, 1.98587820e-02,
        2.07413945e-02, 2.33892321e-02, 4.36893204e-02, 4.36893204e-02,
        1.36804943e-01, 3.38040600e-01, 3.59664607e-01, 3.79082083e-01,
        3.79523389e-01, 4.40423654e-01, 4.50132392e-01, 4.50573698e-01,
        4.51015004e-01, 4.59399823e-01, 5.16769638e-01, 6.97705207e-01,
        6.99911739e-01, 7.00353045e-01, 7.12268314e-01, 7.13592233e-01,
        7.15357458e-01, 7.15357458e-01, 7.16681377e-01, 7.33451015e-01,
        7.34333628e-01, 8.15975287e-01, 8.16857899e-01, 8.17299206e-01,
        8.19064431e-01, 8.19505737e-01, 8.19505737e-01, 8.86142983e-01,
        8.87025596e-01, 9.46160635e-01, 9.47043248e-01, 9.47043248e-01,
        9.47484554e-01, 9.47925861e-01, 9.50573698e-01, 9.50573698e-01,
        9.51015004e-01, 9.51015004e-01, 9.51015004e-01, 9.51015004e-01,
        9.51015004e-01, 9.84995587e-01, 9.89849956e-01, 9.90732568e-01,
        9.90732568e-01, 9.90732568e-01, 9.90732568e-01, 9.90732568e-01,
        9.91615181e-01, 9.92497793e-01, 9.92497793e-01, 9.92497793e-01,
        9.92497793e-01, 9.92497793e-01, 9.92497793e-01, 9.92497793e-01,
        9.92939100e-01, 9.93821712e-01, 9.93821712e-01, 9.93821712e-01,
        9.97793469e-01, 9.97793469e-01, 9.97793469e-01, 9.97793469e-01,
        9.98234775e-01, 9.98234775e-01, 9.98234775e-01, 9.98234775e-01,
        9.98234775e-01, 9.98234775e-01, 9.99117387e-01, 9.99558694e-01,
        9.99558694e-01, 9.99558694e-01, 9.99558694e-01, 9.99558694e-01,
        9.99558694e-01, 9.99558694e-01, 1.00000000e+00, 1.00000000e+00,
        1.00000000e+00, 1.00000000e+00, 1.00000000e+00, 1.00000000e+00,
        1.00000000e+00, 1.00000000e+00, 1.00000000e+00, 1.00000000e+00,
        1.00000000e+00, 1.00000000e+00, 1.00000000e+00, 1.00000000e+00,
        1.00000000e+00, 1.00000000e+00, 1.00000000e+00, 1.00000000e+00,
        1.00000000e+00, 1.00000000e+00, 1.00000000e+00, 1.00000000e+00]),
 array([1.99998975e+00, 9.99989752e-01, 9.99963630e-01, 9.99926618e-01,
        9.99824507e-01, 9.99739607e-01, 9.99696013e-01, 9.99619996e-01,
        9.99067415e-01, 9.98921946e-01, 9.98744640e-01, 9.98652624e-01,
        9.97982408e-01, 9.97827196e-01, 9.97285124e-01, 9.96068727e-01,
        9.94818870e-01, 9.93531147e-01, 9.92330917e-01, 9.91658985e-01,
        9.90946856e-01, 9.90430784e-01, 9.85729281e-01, 9.72513628e-01,
        9.68594012e-01, 9.66532756e-01, 9.64045300e-01, 9.55451586e-01,
        9.51129211e-01, 9.48701894e-01, 9.43188923e-01, 9.38594823e-01,
        9.36681743e-01, 9.19686683e-01, 8.69333113e-01, 8.01307904e-01,
        8.00272142e-01, 7.62190244e-01, 7.41881568e-01, 7.20870097e-01,
```

```
       6.73819364e-01, 5.95510427e-01, 4.71582333e-01, 3.76039363e-01,
       3.58780052e-01, 3.09184642e-01, 2.79768967e-01, 2.78462554e-01,
       2.35885412e-01, 2.23269965e-01, 1.86945330e-01, 1.83648482e-01,
       1.28515565e-01, 1.17670662e-01, 1.01339571e-01, 9.24171186e-02,
       7.95826653e-02, 7.76256984e-02, 6.46881585e-02, 5.48630241e-02,
       5.11368216e-02, 4.13271293e-02, 3.85913902e-02, 3.77892179e-02,
       3.11094131e-02, 3.04578364e-02, 2.75807056e-02, 2.01774252e-02,
       1.93065321e-02, 1.82829397e-02, 1.74663085e-02, 1.58049406e-02,
       1.55031513e-02, 1.40202683e-02, 1.26824514e-02, 1.19300861e-02,
       9.56568869e-03, 9.47683387e-03, 9.27400777e-03, 8.04083211e-03,
       6.61749672e-03, 6.09878155e-03, 6.00129941e-03, 5.87241677e-03,
       4.67040048e-03, 4.21923408e-03, 3.94509344e-03, 3.08308090e-03,
       3.01667910e-03, 2.72443128e-03, 2.66668098e-03, 1.79056558e-03,
       1.66748292e-03, 1.51445478e-03, 1.33426161e-03, 1.30547565e-03,
       1.17880864e-03, 1.16160909e-03, 9.29384926e-04, 9.20677381e-04,
       8.56150440e-04, 7.44205863e-04, 7.39156334e-04, 6.54824520e-04,
       5.91337209e-04, 5.81186973e-04, 4.01716645e-04, 3.81344283e-04,
       2.51162123e-04, 1.74890361e-04, 1.64780647e-04, 1.28668971e-04,
       1.25889731e-04, 1.23196481e-04, 1.11246454e-04, 5.69870559e-05]))
```

Calculating the area under the curve(GINI)

In [169]:

```
def auc_val(fpr,tpr):
    AreaUnderCurve = 0.
    for i in range(len(fpr)-1):
        AreaUnderCurve += (fpr[i+1]-fpr[i]) * (tpr[i+1]+tpr[i])
    AreaUnderCurve *= 0.5
    return AreaUnderCurve
```

In [170]:

```
auc = auc_val(fpr,tpr)
auc
```

Out[170]:

```
0.9623860234430959
```

Step 10:Finding Optimal Cutoff Point

Optimal cutoff probability is that prob where we get balanced sensitivity and specificity

In [171]:

```
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 else 0)
y_train_pred_final.head()
```

Out[171]:

	Converted	Conversion_Prob	LeadID	predicted	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0	0	0.064688	8529	0	1	0	0	0	0	0	0	0	0	0
1	0	0.009566	7331	0	1	0	0	0	0	0	0	0	0	0
2	1	0.762190	7688	1	1	1	1	1	1	1	1	0	0	0
3	0	0.077626	92	0	1	0	0	0	0	0	0	0	0	0
4	0	0.077626	4908	0	1	0	0	0	0	0	0	0	0	0

In [172]:

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

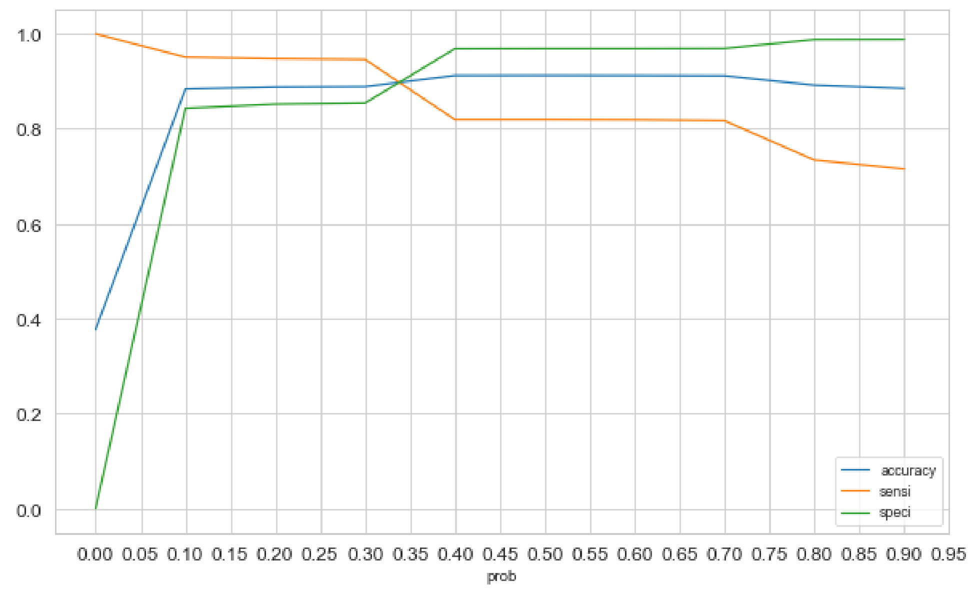
In [173]:

```
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:
    cm1 = metrics.confusion_matrix(y_train_pred_final.Converted, y_train_pred_final[i] )
    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  0.377541  1.000000  0.000000
0.1   0.1  0.884039  0.951015  0.843415
0.2   0.2  0.888204  0.947926  0.851981
0.3   0.3  0.889037  0.946161  0.854390
0.4   0.4  0.912363  0.819506  0.968683
0.5   0.5  0.912529  0.819506  0.968951
0.6   0.6  0.912363  0.819064  0.968951
0.7   0.7  0.911863  0.817299  0.969218
0.8   0.8  0.892203  0.734334  0.987955
0.9   0.9  0.885205  0.715357  0.988223
```

In [175]:

```
sns.set_style("whitegrid") # white/whitegrid/dark/ticks
sns.set_context("paper") # talk/poster
cutoff_df.plot.line(x='prob', y=['accuracy','sensi','speci'], figsize=(10,6))
# plot x axis limits
plt.xticks(np.arange(0, 1, step=0.05), size = 12)
plt.yticks(size = 12)
plt.show()
```



In [176]:

```
y_train_pred_final['final_predicted'] = y_train_pred_final.Conversion_Prob.map( lambda x: 1 if x >
0.33 else 0)

y_train_pred_final.head()
```

Out[176]:

	Converted	Conversion_Prob	LeadID	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	0.064688	8529	0	1	0	0	0	0	0	0	0	0	0	0
1	0	0.009566	7331	0	1	0	0	0	0	0	0	0	0	0	0

	Converted	Conversion_Prob	LeadID	predicted	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	final_predicted
2	1	0.762190	7688	1	1	1	1	1	1	1	1	1	0	0	1
3	0	0.077626	92	0	1	0	0	0	0	0	0	0	0	0	0
4	0	0.077626	4908	0	1	0	0	0	0	0	0	0	0	0	0

In [177]:

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

Out[177]:

```
0.9031989336887704
```

In [178]:

```
confusion1 = metrics.confusion_matrix(y_train_pred_final.Converted,
y_train_pred_final.final_predicted)
confusion1
```

Out[178]:

```
array([[3411,  325],
       [ 256, 2010]], dtype=int64)
```

In [179]:

```
TP = confusion1[1,1] # true positive
TN = confusion1[0,0] # true negatives
FP = confusion1[0,1] # false positives
FN = confusion1[1,0] # false negatives
```

In [180]:

```
# Let's see the sensitivity of our logistic regression model
TP / float(TP+FN)
```

Out[180]:

```
0.8870255957634599
```

In [181]:

```
# Let us calculate specificity
TN / float(TN+FP)
```

Out[181]:

```
0.9130085653104925
```

In [182]:

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

```
0.0869914346895075
```

In [183]:

```
print (TP / float(TP+FP))
```

```
0.860813704496788
```

In [184]:

```
print (TN / float(TN+ FN))
```

```
0.9301881647122989
```

Step 11: Precision and Recall

Precision- TP / TP + FP

In [185]:

```
precision = confusion1[1,1]/(confusion1[0,1]+confusion1[1,1])
precision
```

Out[185]:

```
0.860813704496788
```

Recall- TP / TP + FN

In [186]:

```
recall = confusion1[1,1]/(confusion1[1,0]+confusion1[1,1])
recall
```

Out[186]:

```
0.8870255957634599
```

In [187]:

```
from sklearn.metrics import precision_score, recall_score
```

In [188]:

```
precision_score(y_train_pred_final.Converted, y_train_pred_final.final_predicted)
```

Out[188]:

```
0.860813704496788
```

In [189]:

```
recall_score(y_train_pred_final.Converted, y_train_pred_final.final_predicted)
```

Out[189]:

```
0.8870255957634599
```

Precision and recall trade off

In [190]:

```
from sklearn.metrics import precision_recall_curve
```

In [191]:

```
y_train_pred_final.Converted, y_train_pred_final.final_predicted
```

Out[191]:

```
(0       0
 1       0
 2       1
 3       0
 4       0
        ..
 5997    0
```

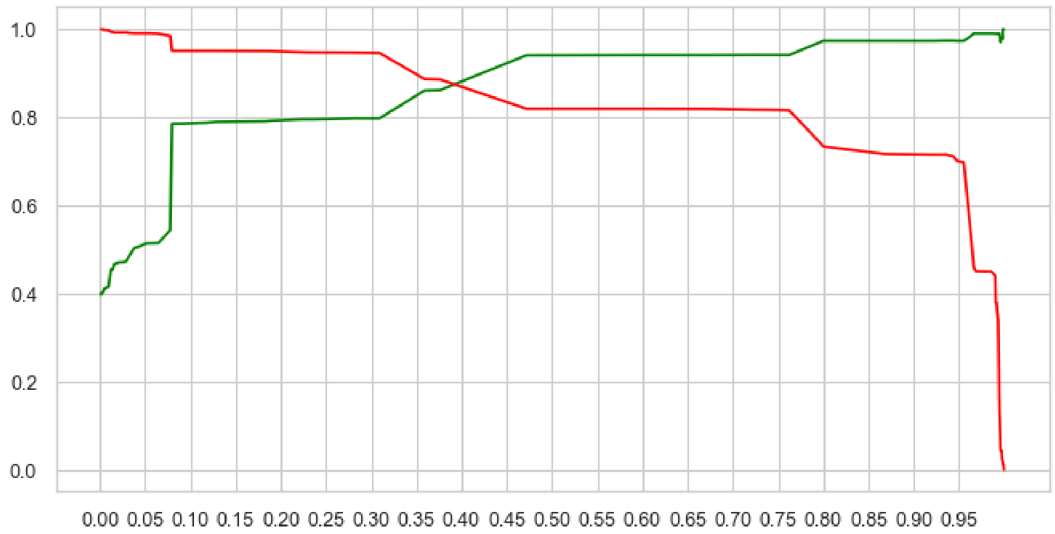
```
5998     0
5999     0
6000     1
6001     0
Name: Converted, Length: 6002, dtype: int64,
0        0
1        0
2        1
3        0
4        0
        ..
5997     0
5998     0
5999     0
6000     1
6001     0
Name: final_predicted, Length: 6002, dtype: int64)
```

In [192]:

```
p, r, thresholds = precision_recall_curve(y_train_pred_final.Converted, y_train_pred_final.Conversi
on_Prob)
```

In [193]:

```
plt.figure(figsize=(8, 4), dpi=100, facecolor='w', edgecolor='k', frameon='True')
plt.plot(thresholds, p[:-1], "g-")
plt.plot(thresholds, r[:-1], "r-")
plt.xticks(np.arange(0, 1, step=0.05))
plt.show()
```



Calculating the F1 score

F1 = 2×(Precision*Recall)/(Precision+Recall)

In [194]:

```
F1 = 2*(precision*recall)/(precision+recall)
F1
```

Out[194]:

```
0.8737231036731146
```

Step 11: Making predictions on the test set

In [195]:

In [195]:

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

Out[195]:

	Do Not Email	Do Not Call	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Search	Newspaper Article	X Education Forums	Newspaper	Digital Advertisement	...	Last Activity_Form Submitted on Website	Acti Conv
6190	0	0	-1.199737	-0.872062	-1.270553	0	0	0	0	0	...	0	
7073	0	0	0.969969	-0.615211	1.785283	0	0	0	0	0	...	0	
4519	1	0	-1.199737	-0.872062	-1.270553	0	0	0	0	0	...	0	
607	0	0	-1.199737	-0.872062	-1.270553	0	0	0	0	0	...	0	
440	0	0	1.403911	-0.094170	0.562949	0	0	0	0	0	...	0	

5 rows × 180 columns

In [196]:

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

Out[196]:

	Lead Source_Welingak Website	Lead Source_Welingak Website	Lead Quality_Worst	Asymmetrique Activity Index_03.Low	Tags_Already a student	Tags_Closed by Horizzon	Tags_Interested in full time MBA	Tags_Interested in other courses
6190	0	0	1	0	1	0	0	0
7073	0	0	0	0	0	0	0	0
4519	0	0	0	0	0	0	0	0
607	1	1	0	0	0	0	0	0
440	0	0	0	0	0	0	0	0

In [197]:

```
X_test_sm = sm.add_constant(X_test)
```

In [198]:

```
y_test_pred = res.predict(X_test_sm)
```

In [199]:

```
y_test_pred[:10]
```

Out[199]:

```
6190    0.000591
7073    0.077626
4519    0.309185
607     0.999825
440     0.077626
4247    0.077626
7431    0.008041
726     0.376039
7300    0.008041
4046    0.077626
```

```
4040    0.077626
dtype: float64
```

In [200]:

```
# Converting y_pred to a dataframe which is an array
y_pred_1 = pd.DataFrame(y_test_pred)
```

In [201]:

```
y_pred_1.head()
```

Out[201]:

	0
6190	0.000591
7073	0.077626
4519	0.309185
607	0.999825
440	0.077626

In [202]:

```
y_test_df = pd.DataFrame(y_test)
```

In [203]:

```
y_test_df['LeadID'] = y_test_df.index
```

In [204]:

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

In [205]:

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

In [206]:

```
y_pred_final.head()
```

Out[206]:

	Converted	LeadID	0
0	0	6190	0.000591
1	0	7073	0.077626
2	0	4519	0.309185
3	1	607	0.999825
4	0	440	0.077626

In [207]:

```
# Renaming the column
y_pred_final= y_pred_final.rename(columns={ 0 : 'Conversion_Prob'})
```

In [208]:

```
# Rearranging the columns
y_pred_final = y_pred_final.reindex_axis(['LeadID','Converted','Conversion_Prob'], axis=1)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-208-314d1e9da6d6> in <module>
      1 # Rearranging the columns
----> 2 y_pred_final = y_pred_final.reindex_axis(['LeadID','Converted','Conversion_Prob'], axis=1)

E:\ProgramData\Anaconda3\lib\site-packages\pandas\core\generic.py in __getattr__(self, name)
   5272             if self._info_axis._can_hold_identifiers_and_holds_name(name):
   5273                 return self[name]
-> 5274             return object.__getattribute__(self, name)
   5275 
   5276     def __setattr__(self, name: str, value) -> None:

AttributeError: 'DataFrame' object has no attribute 'reindex_axis'
```

In [ ]:

```
y_pred_final.head()
```

In [ ]:

```
y_pred_final.shape
```

In [ ]:

```
y_pred_final['final_predicted'] = y_pred_final.Conversion_Prob.map(lambda x: 1 if x > 0.33 else 0)
```

In [ ]:

```
y_pred_final.head()
```

In [209]:

```
acc_score=metrics.accuracy_score(y_pred_final.Converted, y_pred_final.final_predicted)
acc_score
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-209-f3b9218dc6e3> in <module>
----> 1 acc_score=metrics.accuracy_score(y_pred_final.Converted, y_pred_final.final_predicted)
      2 acc_score

E:\ProgramData\Anaconda3\lib\site-packages\pandas\core\generic.py in __getattr__(self, name)
   5272             if self._info_axis._can_hold_identifiers_and_holds_name(name):
   5273                 return self[name]
-> 5274             return object.__getattribute__(self, name)
   5275 
   5276     def __setattr__(self, name: str, value) -> None:

AttributeError: 'DataFrame' object has no attribute 'final_predicted'
```

In [210]:

```
confusion_test = metrics.confusion_matrix(y_pred_final.Converted, y_pred_final.final_predicted )
print(confusion_test)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-210-6c9655640d21> in <module>
----> 1 confusion_test = metrics.confusion_matrix(y_pred_final.Converted, y_pred_final.final_predic
ted )
      2 print(confusion_test)

E:\ProgramData\Anaconda3\lib\site-packages\pandas\core\generic.py in __getattr__(self, name)
   5272             if self._info_axis._can_hold_identifiers_and_holds_name(name):
   5273                 return self[name]
-> 5274             return object.__getattribute__(self, name)
   5275 
```

```
   5276     def __setattr__(self, name: str, value) -> None:

AttributeError: 'DataFrame' object has no attribute 'final_predicted'
```

In [211]:

```
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import metrics
```

In [212]:

```
import matplotlib.pyplot as plt
from mlxtend.plotting import plot_confusion_matrix

fig, ax = plot_confusion_matrix(conf_mat=confusion_test)
all_sample_title = 'Accuracy Score: {0}'.format(acc_score)
plt.title(all_sample_title, size = 12);
# Automatically adjust subplot params so that the subplotS fits in to the figure area.
plt.tight_layout()

# display the plot
plt.show()
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
<ipython-input-212-590096f1df55> in <module>
      1 import matplotlib.pyplot as plt
----> 2 from mlxtend.plotting import plot_confusion_matrix
      3 
      4 fig, ax = plot_confusion_matrix(conf_mat=confusion_test)
      5 all_sample_title = 'Accuracy Score: {0}'.format(acc_score)

ModuleNotFoundError: No module named 'mlxtend'
```

In [213]:

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

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-213-74bf28863f49> in <module>
----> 1 TP = confusion_test[1,1] # true positive
      2 TN = confusion_test[0,0] # true negatives
      3 FP = confusion_test[0,1] # false positives
      4 FN = confusion_test[1,0] # false negatives

NameError: name 'confusion_test' is not defined
```

In [214]:

```
# Let's see the sensitivity of our logistic regression model
TP / float(TP+FN)
```

Out[214]:

```
0.8870255957634599
```

In [215]:

```
# Let us calculate specificity
TN / float(TN+FP)
```

Out[215]:

```
0.9130085653104925
```

False Postive Rate

FP / TN + FP

In [216]:

```
# Calculate false postive rate - predicting churn when customer does not have churned
print(FP/ float(TN+FP))
```

```
0.0869914346895075
```

Positive Predictive Value

TP / TP + FP

In [217]:

```
# Positive predictive value
print (TP / float(TP+FP))
```

```
0.860813704496788
```

Negative Predictive Value

TN / TN + FN

In [218]:

```
# Negative predictive value
print (TN / float(TN+ FN))
```

```
0.9301881647122989
```

Precision

TP / TP + FP

In [219]:

```
Precision = confusion_test[1,1]/(confusion_test[0,1]+confusion_test[1,1])
Precision
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-219-a4c44841b451> in <module>
----> 1 Precision = confusion_test[1,1]/(confusion_test[0,1]+confusion_test[1,1])
      2 Precision

NameError: name 'confusion_test' is not defined
```

Recall

TP / TP + FN

In [220]:

```
Recall = confusion_test[1,1]/(confusion_test[1,0]+confusion_test[1,1])
Recall
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
```

```
<ipython-input-220-6c66ce64eca2> in <module>
----> 1 Recall = confusion_test[1,1]/(confusion_test[1,0]+confusion_test[1,1])
      2 Recall

NameError: name 'confusion_test' is not defined
```

F1 = 2×(Precision*Recall)/(Precision+Recall)

In [221]:

```
F1 = 2*(Precision*Recall)/(Precision+Recall)
F1
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-221-4a0ac02f9c95> in <module>
----> 1 F1 = 2*(Precision*Recall)/(Precision+Recall)
      2 F1

NameError: name 'Precision' is not defined
```

Classification Report

In [222]:

```
from sklearn.metrics import classification_report
print(classification_report(y_pred_final.Converted, y_pred_final.final_predicted))
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-222-5ac5e08ccc9d> in <module>
      1 from sklearn.metrics import classification_report
----> 2 print(classification_report(y_pred_final.Converted, y_pred_final.final_predicted))

E:\ProgramData\Anaconda3\lib\site-packages\pandas\core\generic.py in __getattr__(self, name)
   5272             if self._info_axis._can_hold_identifiers_and_holds_name(name):
   5273                 return self[name]
-> 5274             return object.__getattribute__(self, name)
   5275 
   5276     def __setattr__(self, name: str, value) -> None:

AttributeError: 'DataFrame' object has no attribute 'final_predicted'
```

In [223]:

```
from sklearn.model_selection import cross_val_score

lr = LogisticRegression(solver = 'lbfgs')
scores = cross_val_score(lr, X, y, cv=10)
scores.sort()
accuracy = scores.mean()

print(scores)
print(accuracy)
```

```
[0.85997666 0.90198366 0.90898483 0.91142191 0.91608392 0.92074592
 0.92191142 0.92298716 0.93589744 0.9369895 ]
0.9136982426363991
```

Plotting the ROC Curve for Test Dataset

In [224]:

```
def draw_roc( actual, probs ):
    fpr, tpr, thresholds = metrics.roc_curve( actual, probs,
                                              drop_intermediate = False )
    auc_score = metrics.roc_auc_score( actual, probs )
```

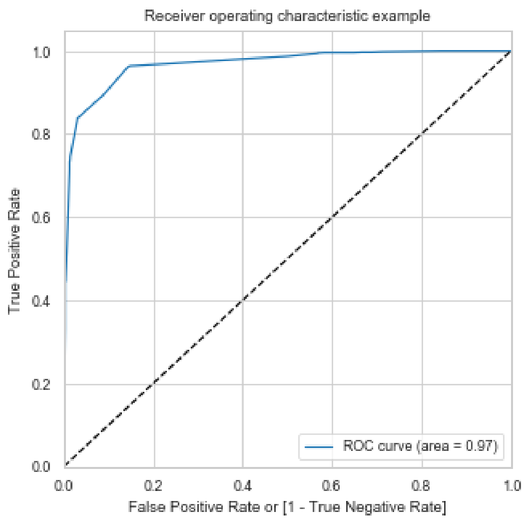
```
    plt.figure(figsize=(5, 5))
    plt.plot( fpr, tpr, label='ROC curve (area = %0.2f)' % auc_score )
    plt.plot([0, 1], [0, 1], 'k--')
    plt.xlim([0.0, 1.0])
    plt.ylim([0.0, 1.05])
    plt.xlabel('False Positive Rate or [1 - True Negative Rate]')
    plt.ylabel('True Positive Rate')
    plt.title('Receiver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

    return fpr,tpr, thresholds
```

In [225]:

```
fpr, tpr, thresholds = metrics.roc_curve( y_pred_final.Converted, y_pred_final.Conversion_Prob, dro
p_intermediate = False )
```

In [226]:

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



Out[226]:

```
(array([0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
        0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
        0.00000000e+00, 0.00000000e+00, 0.00000000e+00, 0.00000000e+00,
        0.00000000e+00, 0.00000000e+00, 6.34115409e-04, 2.53646164e-03,
        2.53646164e-03, 2.53646164e-03, 2.53646164e-03, 2.53646164e-03,
        3.80469245e-03, 3.80469245e-03, 6.34115409e-03, 1.26823082e-02,
        1.26823082e-02, 1.26823082e-02, 1.26823082e-02, 1.33164236e-02,
        1.33164236e-02, 1.39505390e-02, 1.39505390e-02, 2.98034242e-02,
        2.98034242e-02, 2.98034242e-02, 8.24350032e-02, 8.37032340e-02,
        1.43310082e-01, 1.43944198e-01, 1.45846544e-01, 1.46480659e-01,
        1.47114775e-01, 1.49651237e-01, 1.50919467e-01, 1.52821814e-01,
        1.53455929e-01, 4.98414711e-01, 5.67533291e-01, 5.68167406e-01,
        5.82752061e-01, 5.85288523e-01, 6.37285986e-01, 6.48065948e-01,
        6.66455295e-01, 6.67089410e-01, 6.72162334e-01, 6.81039949e-01,
        6.81674065e-01, 6.88649334e-01, 7.12111604e-01, 8.44007609e-01,
        8.44641725e-01, 8.59860495e-01, 8.63665187e-01, 8.64299302e-01,
        8.70006341e-01, 8.71908687e-01, 8.72542803e-01, 8.73176918e-01,
        8.91566265e-01, 8.97273304e-01, 8.98541535e-01, 8.99175650e-01,
        9.00443881e-01, 9.01077996e-01, 9.02346227e-01, 9.02980342e-01,
        9.04248573e-01, 9.04882689e-01, 9.16296766e-01, 9.23272036e-01,
        9.23906151e-01, 9.31515536e-01, 9.43563729e-01, 9.74001268e-01,
        9.74635384e-01, 9.75269499e-01, 9.75903614e-01, 9.79074192e-01,
        9.79708307e-01, 9.80342422e-01, 9.98731769e-01, 9.99365885e-01,
        1.00000000e+00]),
 array([0.        , 0.00200803, 0.0060241 , 0.00702811, 0.00903614,
        0.01305221, 0.01506024, 0.02208835, 0.0251004 , 0.02710843,
        0.03413655, 0.03614458, 0.03915663, 0.06024096, 0.14959839,
        0.33032129, 0.35240964, 0.37048193, 0.42670683, 0.4437751 ,
        0.44578313, 0.45180723, 0.51907631, 0.71485944, 0.71987952,
        0.73192771, 0.73393574, 0.73393574, 0.73493976, 0.74497992,
```

```
        0.74598394, 0.83333333, 0.83534137, 0.8373494 , 0.88855422,
        0.88855422, 0.96184739, 0.96184739, 0.96285141, 0.96285141,
        0.96285141, 0.96385542, 0.96385542, 0.96385542, 0.96385542,
        0.98694779, 0.9939759 , 0.99497992, 0.99598394, 0.99598394,
        0.99598394, 0.99598394, 0.99698795, 0.99698795, 0.99698795,
        0.99698795, 0.99698795, 0.99698795, 0.99799197, 1.        ,
        1.        , 1.        , 1.        , 1.        , 1.        ,
        1.        , 1.        , 1.        , 1.        , 1.        ,
        1.        , 1.        , 1.        , 1.        , 1.        ,
        1.        , 1.        , 1.        , 1.        , 1.        ,
        1.        , 1.        , 1.        , 1.        , 1.        ,
        1.        , 1.        , 1.        , 1.        , 1.        ,
        1.        , 1.        , 1.        ]),
 array([1.99998975e+00, 9.99989752e-01, 9.99963630e-01, 9.99926618e-01,
        9.99824507e-01, 9.99739607e-01, 9.99696013e-01, 9.99619996e-01,
        9.98921946e-01, 9.98744640e-01, 9.98652624e-01, 9.97982408e-01,
        9.97827196e-01, 9.97285124e-01, 9.94818870e-01, 9.93531147e-01,
        9.92330917e-01, 9.91658985e-01, 9.90430784e-01, 9.85729281e-01,
        9.72513628e-01, 9.66532756e-01, 9.64045300e-01, 9.55451586e-01,
        9.51129211e-01, 9.43188923e-01, 9.38594823e-01, 9.36681743e-01,
        9.08834949e-01, 8.01307904e-01, 8.00272142e-01, 7.62190244e-01,
        7.20870097e-01, 6.73819364e-01, 3.76039363e-01, 3.58780052e-01,
        3.09184642e-01, 2.35885412e-01, 1.86945330e-01, 1.83648482e-01,
        1.28515565e-01, 1.17670662e-01, 1.01339571e-01, 9.24171186e-02,
        7.95826653e-02, 7.76256984e-02, 6.46881585e-02, 5.48630241e-02,
        4.13271293e-02, 3.85913902e-02, 3.11094131e-02, 3.04578364e-02,
        2.75807056e-02, 2.27525240e-02, 2.01774252e-02, 1.82829397e-02,
        1.74663085e-02, 1.55031513e-02, 1.40202683e-02, 9.56568869e-03,
        9.47683387e-03, 8.04083211e-03, 6.61749672e-03, 6.09878155e-03,
        6.00129941e-03, 5.87241677e-03, 4.92713243e-03, 4.21923408e-03,
        3.94509344e-03, 3.08308090e-03, 3.01667910e-03, 2.95232373e-03,
        2.72443128e-03, 2.23748987e-03, 1.79056558e-03, 1.66748292e-03,
        1.51445478e-03, 1.36773766e-03, 1.33426161e-03, 1.30547565e-03,
        1.17880864e-03, 9.29384926e-04, 6.54824520e-04, 5.91337209e-04,
        5.81186973e-04, 4.76695605e-04, 4.01716645e-04, 3.81344283e-04,
        2.45737689e-04, 1.28668971e-04, 1.11246454e-04, 6.31089388e-05,
        5.69870559e-05]))
```

In [227]:

```
def auc_val(fpr,tpr):
    AreaUnderCurve = 0.
    for i in range(len(fpr)-1):
        AreaUnderCurve += (fpr[i+1]-fpr[i]) * (tpr[i+1]+tpr[i])
    AreaUnderCurve *= 0.5
    return AreaUnderCurve
```

In [228]:

```
auc = auc_val(fpr,tpr)
auc
```

Out[228]:

```
0.9678947241088641
```

As a rule of thumb, an AUC can be classed as follows,

0.90 - 1.00 = excellent 0.80 - 0.90 = good 0.70 - 0.80 = fair 0.60 - 0.70 = poor 0.50 - 0.60 = fail Since we got a value of 0.9678, our model seems to be doing well on the test dataset

Calculating Lead score for the entire dataset

Lead Score = 100 * ConversionProbability

In [229]:

```
leads_test_pred = y_pred_final.copy()
leads_test_pred.head()
```

Out[229]:

	Converted	LeadID	Conversion_Prob
0	0	6190	0.000591
1	0	7073	0.077626
2	0	4519	0.309185
3	1	607	0.999825
4	0	440	0.077626

In [230]:

```
leads_train_pred = y_train_pred_final.copy()
leads_train_pred.head()
```

Out[230]:

	Converted	Conversion_Prob	LeadID	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	0.064688	8529	0	1	0	0	0	0	0	0	0	0	0	0
1	0	0.009566	7331	0	1	0	0	0	0	0	0	0	0	0	0
2	1	0.762190	7688	1	1	1	1	1	1	1	1	1	0	0	1
3	0	0.077626	92	0	1	0	0	0	0	0	0	0	0	0	0
4	0	0.077626	4908	0	1	0	0	0	0	0	0	0	0	0	0

In [231]:

```
leads_train_pred = leads_train_pred[['LeadID','Converted','Conversion_Prob','final_predicted']]
leads_train_pred.head()
```

Out[231]:

	LeadID	Converted	Conversion_Prob	final_predicted
0	8529	0	0.064688	0
1	7331	0	0.009566	0
2	7688	1	0.762190	1
3	92	0	0.077626	0
4	4908	0	0.077626	0

In [232]:

```
lead_full_pred = leads_train_pred.append(leads_test_pred)
lead_full_pred.head()
```

Out[232]:

	LeadID	Converted	Conversion_Prob	final_predicted
0	8529	0	0.064688	0.0
1	7331	0	0.009566	0.0
2	7688	1	0.762190	1.0
3	92	0	0.077626	0.0
4	4908	0	0.077626	0.0

In [233]:

```
print(leads_train_pred.shape)
print(leads_test_pred.shape)
print(lead_full_pred.shape)
```

```
(6002, 4)
(2573, 3)
(8575, 4)
```

In [234]:

```
len(lead_full_pred['LeadID'].unique().tolist())
```

Out[234]:

```
8575
```

In [235]:

```
lead_full_pred['Lead_Score'] = lead_full_pred['Conversion_Prob'].apply(lambda x : round(x*100))
lead_full_pred.head()
```

Out[235]:

	LeadID	Converted	Conversion_Prob	final_predicted	Lead_Score
0	8529	0	0.064688	0.0	6
1	7331	0	0.009566	0.0	1
2	7688	1	0.762190	1.0	76
3	92	0	0.077626	0.0	8
4	4908	0	0.077626	0.0	8

In [236]:

```
lead_full_pred.LeadID.max()
```

Out[236]:

```
9239
```

In [237]:

```
lead_full_pred = lead_full_pred.set_index('LeadID').sort_index(axis = 0, ascending = True)
lead_full_pred.head()
```

Out[237]:

	Converted	Conversion_Prob	final_predicted	Lead_Score
LeadID				
0	0	0.031109	0.0	3
1	0	0.009566	0.0	1
2	1	0.801308	1.0	80
3	0	0.009566	0.0	1
4	1	0.955452	1.0	96

In [238]:

```
original_leads = original_leads[['Lead Number']]
original_leads.head()
```

Out[238]:

	Lead Number
0	660737
1	660728

	Lead Number
2	660727
3	660719
4	660681

In [239]:

```
leads_with_score = pd.concat([original_leads, lead_full_pred], axis=1)
leads_with_score.head(10)
```

Out[239]:

	Lead Number	Converted	Conversion_Prob	final_predicted	Lead_Score
0	660737	0	0.031109	0.0	3
1	660728	0	0.009566	0.0	1
2	660727	1	0.801308	1.0	80
3	660719	0	0.009566	0.0	1
4	660681	1	0.955452	1.0	96
5	660680	0	0.077626	0.0	8
6	660673	1	0.955452	1.0	96
7	660664	0	0.077626	0.0	8
8	660624	0	0.077626	0.0	8
9	660616	0	0.077626	0.0	8

In [240]:

```
leads_with_score.shape
```

Out[240]:

```
(8575, 5)
```

In [241]:

```
total = pd.DataFrame(leads_with_score.isnull().sum().sort_values(ascending=False),
columns=['Total'])
percentage = pd.DataFrame(round(100*(leads_with_score.isnull().sum()/leads_with_score.shape[0]),2).
sort_values(ascending=False)\
                          ,columns=['Percentage'])
pd.concat([total, percentage], axis = 1)
```

Out[241]:

	Total	Percentage
final_predicted	2573	30.01
Lead_Score	0	0.00
Conversion_Prob	0	0.00
Converted	0	0.00
Lead Number	0	0.00

Determining Feature Importance

In [242]:

```
pd.options.display.float_format = '{:.2f}'.format
new_params = res.params[1:]
new_params
```

Out[242]:

```
Lead Source_Welingak Website                          1.81
Lead Source_Welingak Website                          1.81
Lead Quality_Worst                                   -3.18
Asymmetrique Activity Index_03.Low                   -2.34
Tags_Already a student                               -3.45
Tags_Closed by Horizzon                               5.44
Tags_Interested  in full time MBA                    -2.66
Tags_Interested in other courses                     -2.63
Tags_Lost to EINS                                     6.71
Tags_Not doing further education                     -3.35
Tags_Ringing                                         -3.84
Tags_Will revert after reading the email              3.87
Tags_opp hangup                                      -3.08
Tags_switched off                                    -4.73
What is your current occupation_Unemployed            1.67
What is your current occupation_Working Professional  1.89
Last Activity_SMS Sent                                1.97
dtype: float64
```

In [243]:

```
feature_importance = new_params
feature_importance = 100.0 * (feature_importance / feature_importance.max())
feature_importance
```

Out[243]:

```
Lead Source_Welingak Website                           26.93
Lead Source_Welingak Website                           26.93
Lead Quality_Worst                                    -47.38
Asymmetrique Activity Index_03.Low                    -34.87
Tags_Already a student                                -51.40
Tags_Closed by Horizzon                                81.12
Tags_Interested  in full time MBA                     -39.59
Tags_Interested in other courses                      -39.26
Tags_Lost to EINS                                     100.00
Tags_Not doing further education                      -49.88
Tags_Ringing                                          -57.17
Tags_Will revert after reading the email               57.67
Tags_opp hangup                                       -45.88
Tags_switched off                                     -70.45
What is your current occupation_Unemployed             24.90
What is your current occupation_Working Professional   28.23
Last Activity_SMS Sent                                 29.34
dtype: float64
```

In [244]:

```
sorted_idx = np.argsort(feature_importance,kind='quicksort',order='list of str')
sorted_idx
```

Out[244]:

```
Lead Source_Welingak Website                          13
Lead Source_Welingak Website                          10
Lead Quality_Worst                                     4
Asymmetrique Activity Index_03.Low                     9
Tags_Already a student                                 2
Tags_Closed by Horizzon                               12
Tags_Interested  in full time MBA                      6
Tags_Interested in other courses                       7
Tags_Lost to EINS                                      3
Tags_Not doing further education                      14
Tags_Ringing                                           1
Tags_Will revert after reading the email               0
Tags_opp hangup                                       15
Tags_switched off                                     16
What is your current occupation_Unemployed            11
What is your current occupation_Working Professional   5
Last Activity_SMS Sent                                 8
dtype: int64
```

In [245]:

```
pos = np.arange(sorted_idx.shape[0]) + .5

featfig = plt.figure(figsize=(10,6))
featax = featfig.add_subplot(1, 1, 1)
featax.barh(pos, feature_importance[sorted_idx], align='center', color = 'tab:red',alpha=0.8)
featax.set_yticks(pos)
featax.set_yticklabels(np.array(X_train[col].columns)[sorted_idx], fontsize=12)
featax.set_xlabel('Relative Feature Importance', fontsize=14)

plt.tight_layout()
plt.show()
```

1.Selecting Top 3 features which contribute most towards the probability of a lead getting converted

In [246]:

```
pd.DataFrame(feature_importance).reset_index().sort_values(by=0,ascending=False).head(3)
```

Out[246]:

	index	0
8	Tags_Lost to EINS	100.00
5	Tags_Closed by Horizzon	81.12
11	Tags_Will revert after reading the email	57.67

2. What are the top 3 categorical/dummy variables in the model which get maximum focus in order to increase the probability of lead conversion?

1.Tags_Lost to EINS 2.Tags_Closed by Horizzon 3.Tags_Will revert after reading the email

3. X Education has a period of 2 months every year during which they hire few interns. The sales team, in particular, has around 10 interns allotted to them. So, during this phase, they wish to make the lead conversion more aggressive. So they want almost all of the potential leads (i.e. the customers who have been predicted as 1 by the model) to be converted and hence, want to make phone calls to as much of such people as possible. Suggest a good strategy they should employ at this stage.

Sensitivity with respect to our model can be defined as the ratio of total number of actual Conversions correctly predicted to the total no of actual Conversions.

Similarly, Specificity can be defined as the ratio of total no of actual non-Conversions correctly predicted to the total number of actual non-Conversions.

the aboue sensitivity diagram

4. Similarly, at times, the company reaches its target for a quarter before the deadline. During this time, the company wants the sales team to focus on some new work as well. So during this time, the company's aim is to not make phone calls unless it's extremely necessary, i.e. they want to minimize the rate of useless phone calls. Suggest a strategy they should employ at this stage.

Therefore, since X Education has already reached its target for a quarter and doesn't want to make phone calls unless it's extremely necessary, i.e. they want to minimize the rate of useless phone calls, we can choose a higher threshold value for Conversion Probability.

This will ensure the Specificity rating is very high, which in turn will make sure almost all leads who are on the brink of the probability of getting Converted or not are not selected. As a result the agents won't have to make unnecessary phone calls and can focus on some new work.

In [ ]:

```

```