LEADING SCORE CASE STUDY

IDENTIFYING HOT LEADS

ASHUTOSH SINGH

Lead Score Summary

Problem Description:

An education company named X Education sells online courses to industry professionals. On any given day, many professionals who are interested in the courses land on their website and browse for courses.

The company markets its courses on several websites and search engines like Google. Once these people land on the website, they might browse the courses or fill up a form for the course or watch some videos. When these people fill up a form providing their email address or phone number, they are classified to be a lead. Moreover, the company also gets leads through past referrals. Once these leads are acquired, employees from the sales team start making calls, writing emails, etc. Through this process, some of the leads get converted while most do not. The typical lead conversion rate at X education is around 30%.

X Education has appointed you to help them select the most promising leads, i.e. the leads that are most likely to convert into paying customers. The company requires you to build a model wherein you need to assign a lead score to each of the leads such that the customers with higher lead score have a higher conversion chance and the customers with lower lead score have a lower conversion chance. **The CEO, in particular, has given a ballpark of the target lead conversion rate to be around 80%**.

Goal of the Case Study is

Build a logistic regression model to assign a lead score between 0 and 100 to each of the leads which can be used by the company to target potential leads. A higher score would mean that the lead is hot, i.e. is most likely to convert whereas a lower score would mean that the lead is cold and will mostly not get converted.

Approach:

From above problem description we conclude that the above problem is the classification problem, hence we choose logistic Regression to calculate the Lead rate.

Below are the steps followed to solve this problem .

1. Data Reading and Understanding

Here we tried to get the look and feel of the data, we observed following things

- Number of rows and columns
- Data types of each columns
- Checking first few rows how data looks

Shape of Dataset

In [3]:
```
lead_score.shape
```
Out[3]:
```
(9240, 37)
```

Checking The Datatype Of Each Columns

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

- Checking how the data is spread.

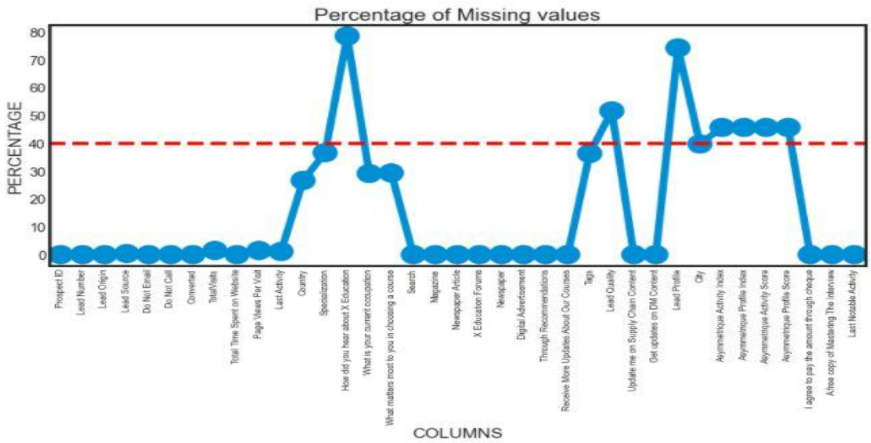
Statistical Summary Of The Dataset

- For Numerical Columns

In [5]:
```
lead_score.describe().transpose()
```
Out[5]:

	count	mean	std	min	25%	50%	75%	max
Lead Number	9240.0	617188.435606	23405.995698	579533.0	596484.5	615479.0	637387.25	660737.0
Converted	9240.0	0.385390	0.486714	0.0	0.0	0.0	1.00	1.0
TotalVisits	9103.0	3.445238	4.854853	0.0	1.0	3.0	5.00	251.0
Total Time Spent on Website	9240.0	487.698268	548.021466	0.0	12.0	248.0	936.00	2272.0
Page Views Per Visit	9103.0	2.362820	2.161418	0.0	1.0	2.0	3.00	55.0
Asymmetrique Activity Score	5022.0	14.306252	1.386694	7.0	14.0	14.0	15.00	18.0
Asymmetrique Profile Score	5022.0	16.344883	1.811395	11.0	15.0	16.0	18.00	20.0

➢ Checking for duplicates.



There are 17 columns with null values. 7 columns have more than 40% unknowns which we should drop as imputing these columns will introduce bias.

3.4 Duplicate Analysis duplicate

```
In [12]: print("Total number of duplicate values in Prospect ID column :" , lead_score.duplicated(subset = 'Prospect ID').sum())
         print("Total number of duplicate values in Lead Number column :" , lead_score.duplicated(subset = 'Lead Number').sum())

Total number of duplicate values in Prospect ID column : 0
Total number of duplicate values in Lead Number column : 0
```

Both the Prospect ID and Lead number are unique columns and hence we wont need for prediction

2 Data Cleaning

Here we checked for discrepancies in the Dataset

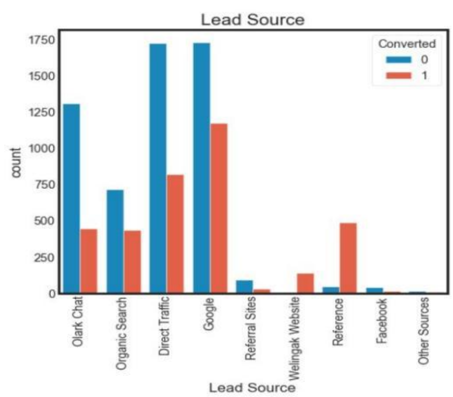
➢ Checking for any column names correction
➢ Checking for null values and imputing them with appropriate methods
➢ We used mode imputation for categorical columns.
➢ We used mean imputation for numerical columns, if there is no skewness in data.
➢ We used median imputation for numerical columns, if there is skewness in the data.

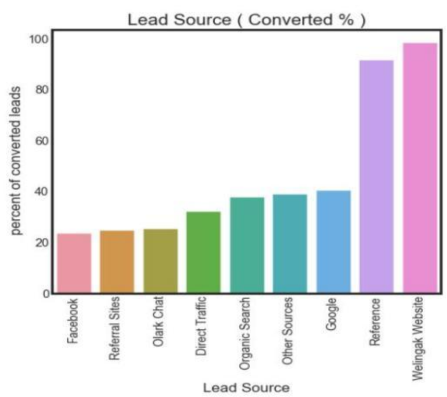
Out[21]:

	Column_name	Labels	No_of_labels	Missing_Value	Percentage_of_Missing_Value
19	City	[nan, Mumbai, Thane & Outskirts, Other Metro C...	6	3669	39.71
6	Specialization	[nan, Business Administration, Media and Adver...	18	3380	36.58
8	What matters most to you in choosing a course	[Better Career Prospects, nan, Flexibility & C...	3	2709	29.32
7	What is your current occupation	[Unemployed, Student, nan, Working Professiona...	6	2690	29.11
5	Country	[nan, India, Russia, Kuwait, Oman, United Arab...	38	2461	26.63
4	Last Activity	[Page Visited on Website, Email Opened, Unreac...	17	103	1.11
1	Lead Source	[Olark Chat, Organic Search, Direct Traffic, G...	21	36	0.39
0	NaN	[API, Landing Page Submission, Lead Add Form, ...	5	0	0.00
14	Digital Advertisement	[No, Yes]	2	0	0.00
20	I agree to pay the amount through cheque	[No]	1	0	0.00
18	Get updates on DM Content	[No]	1	0	0.00
17	Update me on Supply Chain Content	[No]	1	0	0.00
16	Receive More Updates About Our Courses	[No]	1	0	0.00
15	Through Recommendations	[No, Yes]	2	0	0.00
11	Newspaper Article	[No, Yes]	2	0	0.00
13	Newspaper	[No, Yes]	2	0	0.00
12	X Education Forums	[No, Yes]	2	0	0.00
10	Magazine	[No]	1	0	0.00
9	Search	[No, Yes]	2	0	0.00
3	Do Not Call	[No, Yes]	2	0	0.00
2	Do Not Email	[No, Yes]	2	0	0.00
21	A free copy of Mastering The Interview	[No, Yes]	2	0	0.00

3 Data Visualization and Outlier Treatment

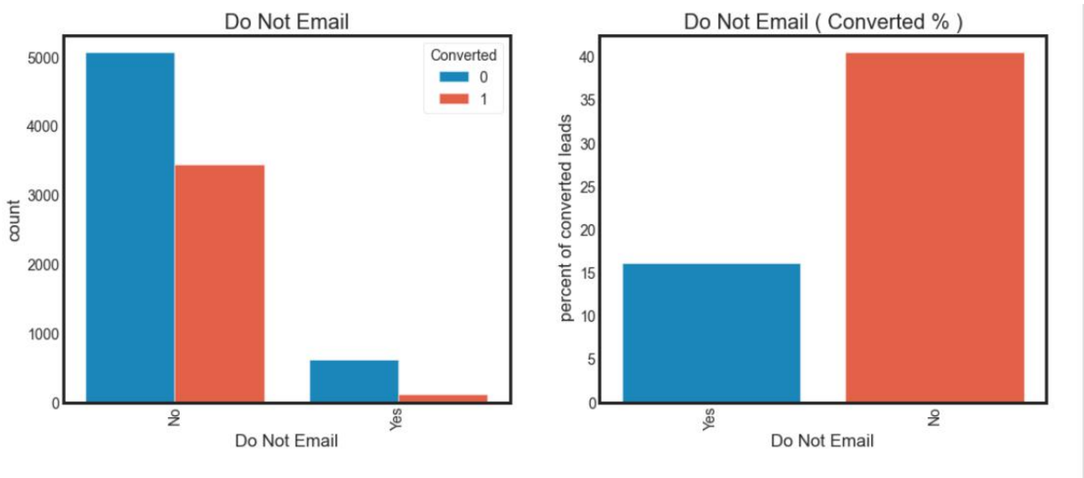
- We performed univariate analysis on categorical column to see which columns makes more sense and removed those columns whose variance is nearly zero.
- We performed bivariate analysis on categorical columns to see how they vary w.r.t Converted column.
- We performed univariate analysis on numerical columns by plotting box plots to see are there any outliers in the data or not.
- We performed bivariate analysis on numerical columns with Converted column to see how the leads are related to these columns.
- We have used IQR method to treat the outliers in the data set.
- In this step we also plotted the correlation matrix to identify the columns which are
- correlated.



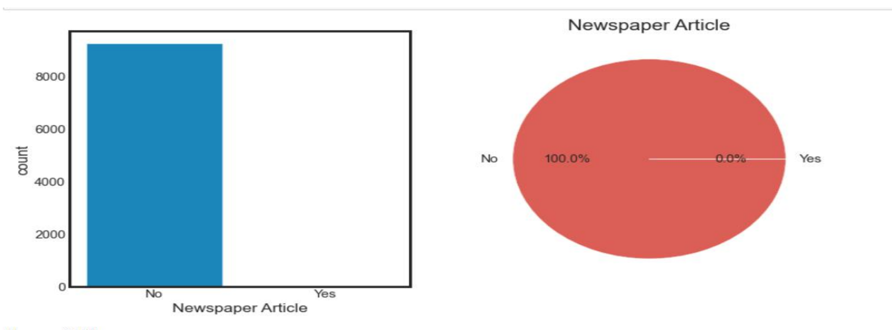


89]:

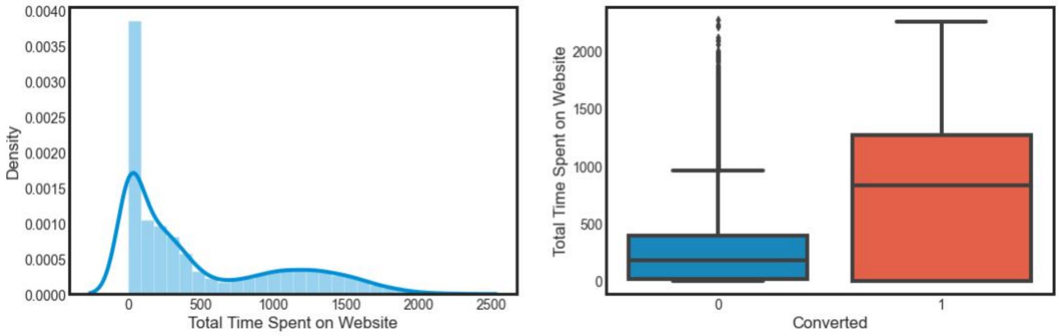
- The source of most leads was Google, and 40% of the leads converted, followed by Direct Traffic,Organic search and Olark chat where around 35%, 38% and 30% converted respectively.
- A lead that came from a reference has over 90% conversion from the total of 534.
- Welingak Website has almost 100% lead conversion rate. This option should be explored more to increase lead conversion



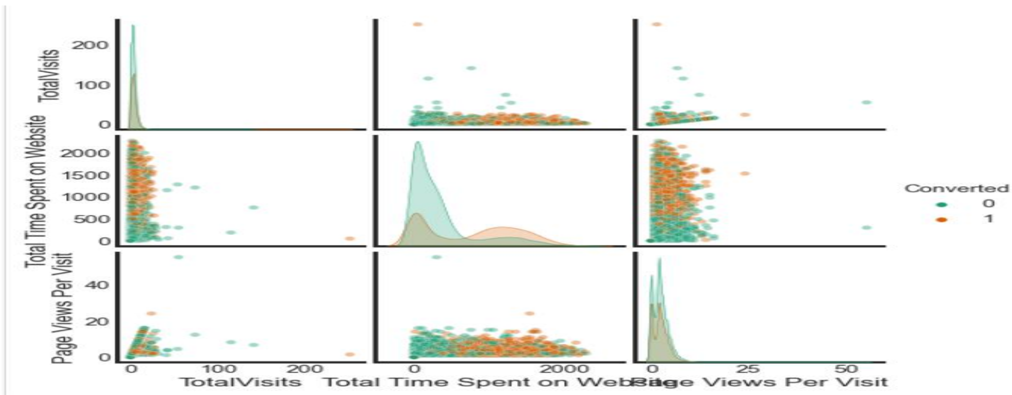
- Majority of the people are ok with receiving email (~92%)
- People who are ok with email has conversion rate of 40%
- People who have opted out of receive email has lower rate of conversion (only 15%)



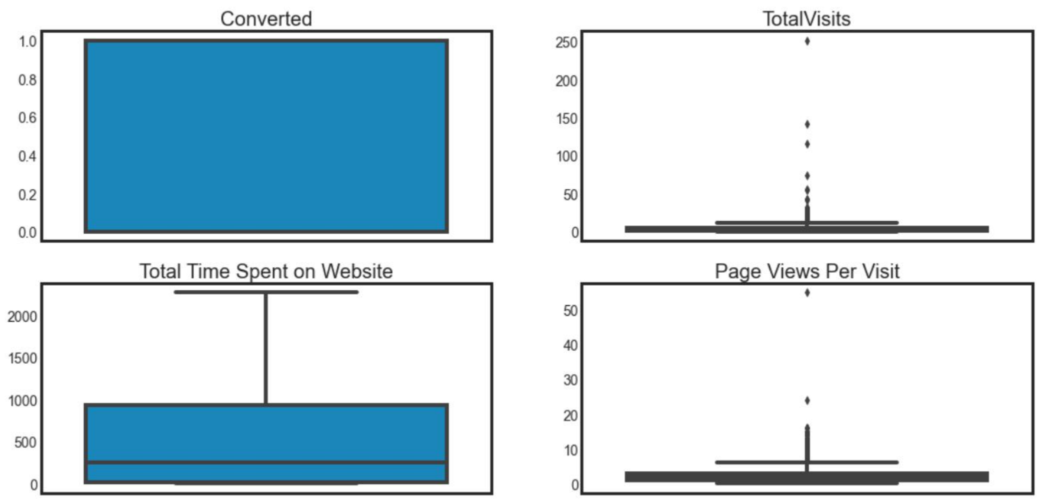
- Newspaper Article data are very skewed and can be deleted as they will not add any value to the model.



➢ The people who spend more time on website has more probalbity to converted in leads

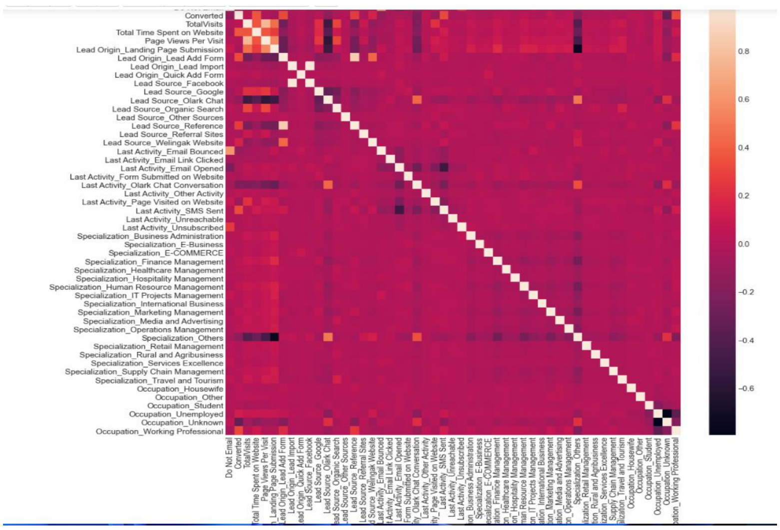


➢ Data is not normally distributed.



➢ Though outliers in TotalVisits and Page Views Per Visit shows valid values, this will misclassify the outcomes and consequently create problems when making inferences with the wrong model. Logistic Regression is heavily influenced by outliers

	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit
count	9240.000000	9240.000000	9240.000000	9240.000000
mean	0.385390	3.179221	487.698268	2.255105
std	0.486714	2.761219	548.021466	1.779471
min	0.000000	0.000000	0.000000	0.000000
10%	0.000000	0.000000	0.000000	0.000000
25%	0.000000	1.000000	12.000000	1.000000
50%	0.000000	3.000000	248.000000	2.000000
75%	1.000000	5.000000	936.000000	3.000000
90%	1.000000	7.000000	1380.000000	5.000000
95%	1.000000	10.000000	1562.000000	6.000000
99%	1.000000	10.000000	1840.610000	6.000000
max	1.000000	10.000000	2272.000000	6.000000



4 Feature Scaling

At this stage our data was very clean and no outliers. We know that logistic regression takes the input parameters as numerical values. Hence, we converted all the categorical columns to numerical.

➢ Columns which have only two levels “Yes” and “No” were converted to numerical using binary mapping.

6.2 Convert Binary Categories

In [125]:
```
binary_col=categorical_missing(lead_score,lead_score.columns)
binary_col=binary_col[binary_col['No_of_labels']==2]
binary_col[['Column_name','No_of_labels']]
```

Out[125]:

	Column_name	No_of_labels
2	Do Not Email	2
3	Converted	2

In [126]:
```
lead_score['Do Not Email'].value_counts()
```

Out[126]:
```
No      8506
Yes      734
Name: Do Not Email, dtype: int64
```

In [127]:
```
def binary_map(x):
    return x.map({'Yes':1,'No':0})
```

In [128]:
```
lead_score['Do Not Email']=lead_score[['Do Not Email']].apply(binary_map)
```

In [129]:
```
lead_score['Do Not Email'].value_counts()
```

Out[129]:
```
0    8506
1     734
Name: Do Not Email, dtype: int64
```

- Columns which have more than two levels were converted to dummies using pd.get_dummies function.Now, the data contained only numerical columns and dummy variables.

6.3 Dummy Variables

In [130]:
```
categorical_missing(lead_score,lead_score.columns)
```

Out[130]:

	Column_name	Labels	No_of_labels	Missing_Value	Percentage_of_Missing_Value
0	Lead Origin	[API, Landing Page Submission, Lead Add Form, ...	5	0	0.0
1	Lead Source	[Olark Chat, Organic Search, Direct Traffic, G...	9	0	0.0
2	Do Not Email	[0, 1]	2	0	0.0
3	Converted	[0, 1]	2	0	0.0
4	TotalVisits	[0.0, 5.0, 2.0, 1.0, 4.0, 8.0, 10.0, 6.0, 3.0,...	11	0	0.0
5	Total Time Spent on Website	[0, 674, 1532, 305, 1428, 1640, 71, 58, 1351, ...	1731	0	0.0
6	Page Views Per Visit	[0.0, 2.5, 2.0, 1.0, 4.0, 6.0, 2.67, 5.0, 3.0,...	91	0	0.0
7	Last Activity	[Page Visited on Website, Email Opened, Unreac...	11	0	0.0
8	Specialization	[Others, Business Administration, Media and Ad...	19	0	0.0
9	Occupation	[Unemployed, Student, Unknown, Working Profess...	7	0	0.0

In [131]:
```
Dummy_var=['Lead Origin','Lead Source','Last Activity','Specialization','Occupation']
```

In [132]:
```
lead=pd.get_dummies(data=lead_score,columns=Dummy_var,drop_first=True)
lead.head()
```

Out[132]:

	Do Not Email	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Lead Origin_Landing Page Submission	Lead Origin_Lead Add Form	Lead Origin_Lead Import	Lead Origin_Quick Add Form	Lead Source_Facebook	Lead Source_Google	Lead Source_Olark Chat	So
0	0	0	0.0	0	0.0	0	0	0	0	0	0	1	
1	0	0	5.0	674	2.5	0	0	0	0	0	0	0	
2	0	1	2.0	1532	2.0	1	0	0	0	0	0	0	
3	0	0	1.0	305	1.0	1	0	0	0	0	0	0	
4	0	1	2.0	1428	1.0	1	0	0	0	0	1	0	

5 rows × 51 columns

➢ Before proceeding for model building, we have rescaled all numerical columns by using standard Scaler method.

6.5 Feature Scaling

```
In [138]: feature_scaling=['TotalVisits','Total Time Spent on Website','Page Views Per Visit']
```

```
In [139]: ## initailizing the object of Standaradscaler
scaler=StandardScaler()
X_train[feature_scaling]=scaler.fit_transform(X_train[feature_scaling])
```

```
In [140]: X_train.head()
```

Out[140]:

	Do Not Email	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Lead Origin_Landing Page Submission	Lead Origin_Lead Add Form	Lead Origin_Lead Import	Lead Origin_Quick Add Form	Lead Source_Facebook	Lead Source_Google	Lead Source_Olark Chat	Source
1871	0	-1.149699	-0.885371	-1.266675	0	0	0	0	0	0	1	
6795	0	0.299722	0.005716	-0.516439	1	0	0	0	0	0	0	
3516	0	0.662077	-0.691418	0.143543	0	0	0	0	0	0	1	
8105	0	0.662077	1.365219	1.553761	1	0	0	0	0	1	0	
3934	0	-1.149699	-0.885371	-1.266675	0	0	0	0	0	0	1	

5 rows × 50 columns

5 Model Building

➢ We have used Recursive Feature Elimination Technique to remove attributes and built a model on those attributes that remain. RFE uses the model accuracy to identify which attributes (and combination of attributes) contribute the most to predicting the target attribute.

7.1 RFE for Feature Reduction

```
In [143]: #using the RFE to Reduce the feautre count from 51 to 20
LR=LogisticRegression()
RFE=RFE(LR,20)
RFE=RFE.fit(X_train,y_train)
```

```
In [144]: list(zip(X_train.columns, RFE.support_, RFE.ranking_))
```

```
In [145]: #checking the column which Rfe has selected
rfe_col=X_train.columns[RFE.support_]
rfe_col
```

```
Out[145]: Index(['Do Not Email', 'Total Time Spent on Website',
       'Lead Origin_Landing Page Submission', 'Lead Origin_Lead Add Form',
       'Lead Source_Facebook', 'Lead Source_Olark Chat',
       'Lead Source_Welingak Website', 'Last Activity_Email Opened',
       'Last Activity_Olark Chat Conversation', 'Last Activity_Other Activity',
       'Last Activity_SMS Sent', 'Last Activity_Unreachable',
       'Last Activity_Unsubscribed', 'Specialization_Hospitality Management',
       'Specialization_Others', 'Specialization_Retail Management',
       'Specialization_Rural and Agribusiness', 'Occupation_Housewife',
       'Occupation_Unknown', 'Occupation_Working Professional'],
      dtype='object')
```

```
In [146]: # columns which have not been selected by Rfe
X_train.columns[~RFE.support_]
```

- In this step we made the model stable by using stats library, where we checked the p-values to be less than 0.05 and vif values to be under 5. Variance inflation factor(vif) is used to treat the multicollinearity.

7.2 Mannual Feature Reduction

For Manual Feature Reduction, the following methods will be followed in order to reduce the features until we reach reasonable amount of feature count and maintain Sensitivity of the modeal =>80%

- High P-Value
- High VIF

MODEL 1

In [148]:
```
features=list(rfe_col)
Lr_model1,X_train1=logistic_model(features)
```

In [149]:
```
Lr_model1.summary()
```

In [150]:
```
calculate_VIF(X_train)
```

MODEL 2

We will remove `Occupation_Housewife` feature due to high P-value of 0.999

In [151]:
```
# Remove 'Occupation_Housewife ' from RFE features list
features.remove('Occupation_Housewife')
Lr_model2,X_train2=logistic_model(features)
Lr_model2.summary()
```

Do Not Email	-1.1811	0.182	-6.492	0.000	-1.538	-0.824
Total Time Spent on Website	1.0651	0.040	26.711	0.000	0.987	1.143
Lead Origin_Landing Page Submission	-1.0227	0.128	-7.972	0.000	-1.274	-0.771
Lead Origin_Lead Add Form	2.8029	0.203	13.794	0.000	2.405	3.201
Lead Source_Olark Chat	1.0993	0.123	8.940	0.000	0.858	1.340
Lead Source_Welingak Website	2.4629	0.750	3.285	0.001	0.993	3.932
Last Activity_Email Opened	0.7288	0.110	6.636	0.000	0.514	0.944
Last Activity_Olark Chat Conversation	-0.6068	0.191	-3.169	0.002	-0.982	-0.231
Last Activity_Other Activity	2.2419	0.488	4.592	0.000	1.285	3.199
Last Activity_SMS Sent	1.8672	0.111	16.782	0.000	1.649	2.085
Last Activity_Unreachable	0.8487	0.368	2.303	0.021	0.126	1.571
Last Activity_Unsubscribed	1.3906	0.485	2.865	0.004	0.439	2.342
Specialization_Hospitality Management	-0.9951	0.327	-3.040	0.002	-1.637	-0.353
Specialization_Others	-0.9785	0.123	-7.927	0.000	-1.220	-0.737
Occupation_Unknown	-1.0818	0.088	-12.357	0.000	-1.253	-0.910
Occupation_Working Professional	2.3966	0.190	12.627	0.000	2.025	2.769

In [158]:
```
calculate_VIF(X_train[features])
```

Out[158]:

	Features	VIF
2	Lead Origin_Landing Page Submission	2.97
13	Specialization_Others	2.77
6	Last Activity_Email Opened	2.55
9	Last Activity_SMS Sent	2.28
4	Lead Source_Olark Chat	2.18
7	Last Activity_Olark Chat Conversation	1.77
3	Lead Origin_Lead Add Form	1.63
14	Occupation_Unknown	1.61
0	Do Not Email	1.27
5	Lead Source_Welingak Website	1.27
1	Total Time Spent on Website	1.25
15	Occupation_Working Professional	1.21
11	Last Activity_Unsubscribed	1.08
8	Last Activity_Other Activity	1.03
10	Last Activity_Unreachable	1.03
12	Specialization_Hospitality Management	1.02

- Once the stable model was created, we predicted probabilities on the train set and created a new column predicted with 1 if probability is greater than .5 else 0.

Creating new column 'predicted' with 1 if lead|_Prob > 0.5 else 0

In [163]:
```
y_train_pred_final['predicted(0.5)'] = y_train_pred_final.Converted_Prob.map(lambda x: 1 if x > 0.5 else 0)
# Let's see the head
y_train_pred_final.head()
```

Out[163]:

	Converted_IND	Converted_Prob	Prospect_IND	predicted(0.5)
0	0	0.523486	1871	1
1	0	0.113305	6795	0
2	0	0.336733	3516	0
3	0	0.818686	8105	1
4	0	0.292254	3934	0

- We calculated the confusion matrix on this predicted column to the actual converted column. We also calculated the metrics sensitivity, specificity, precision, recall and accuracy. We also plotted roc curve to find the area under the curve.

In [164]:
```
matrix=confusion_matrix(y_true=y_train_pred_final['Converted_IND'],y_pred=y_train_pred_final['predicted(0.5)'])
matrix
```

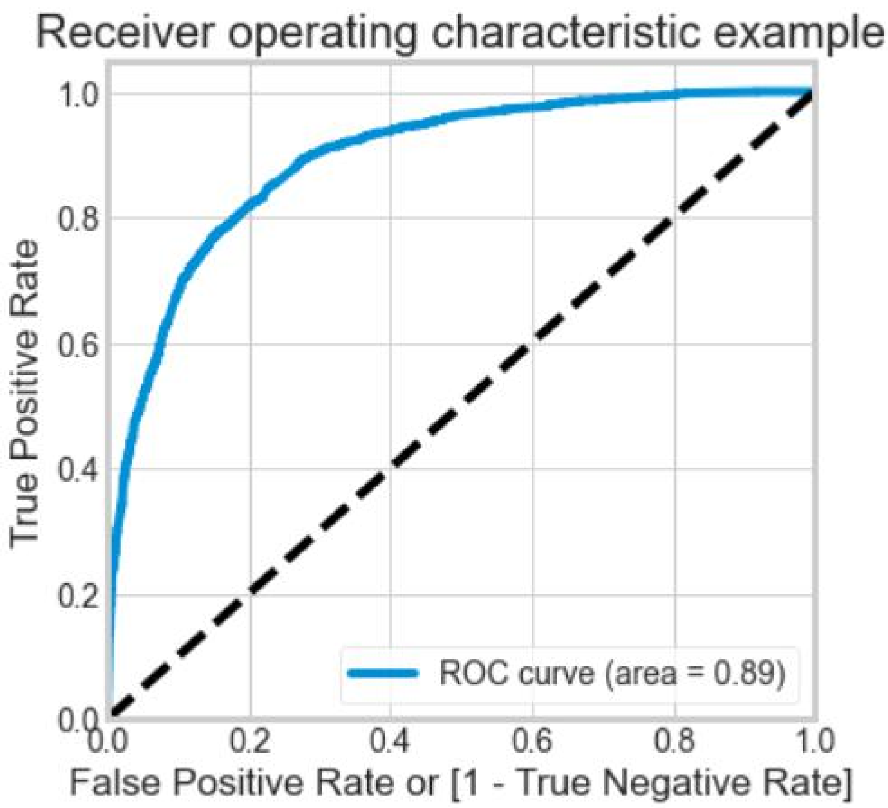
Out[164]:
```
array([[3548,  454],
       [ 717, 1749]], dtype=int64)
```

In [165]:
```
accuracy_score(y_true=y_train_pred_final['Converted_IND'],y_pred=y_train_pred_final['predicted(0.5)'])
```

Out[165]:
```
0.8189548546691404
```

In [166]:
```
lg_metrics(matrix)
```

```
Model Accuracy value is            :  81.9 %
Model Sensitivity value is         :  70.92 %
Model Specificity value is         :  88.66 %
Model Precision value is           :  79.39 %
Model Recall value is              :  70.92 %
Model True Positive Rate (TPR)     :  70.92 %
Model False Positive Rate (FPR)    :  11.34 %
Model Poitive Prediction Value is  :  79.39 %
Model Negative Prediction value is :  83.19 %
```



6 Model Evaluation on Train Test

- In the step 5 we took 0.5 as the cut-of. To confirm that it was the best cut, we calculated the probabilities with different cut-offs.

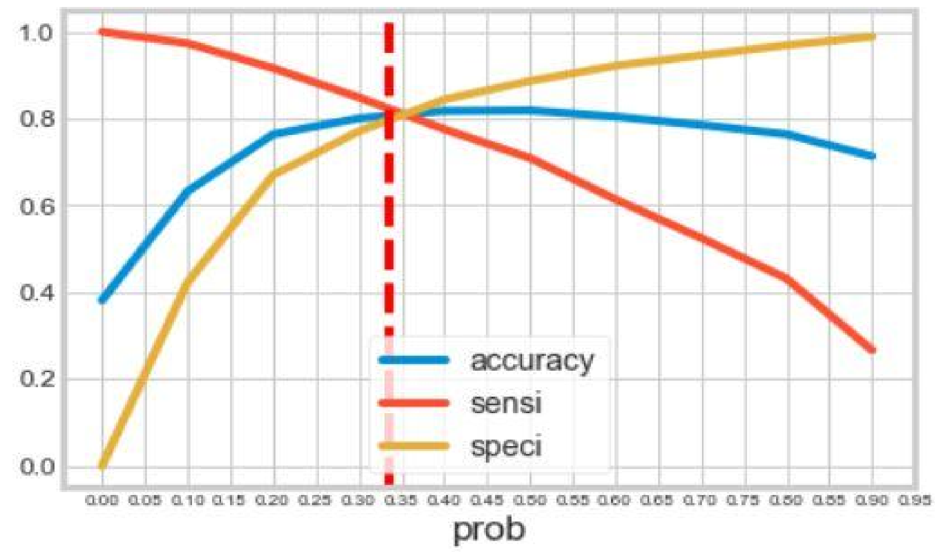
	Converted_IND	Converted_Prob	Prospect_IND	predicted(0.5)	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0	0	0.523486	1871	1	1	1	1	1	1	1	0	0	0	0
1	0	0.113305	6795	0	1	1	0	0	0	0	0	0	0	0
2	0	0.336733	3516	0	1	1	1	1	0	0	0	0	0	0
3	0	0.818686	8105	1	1	1	1	1	1	1	1	1	1	0
4	0	0.292254	3934	0	1	1	1	0	0	0	0	0	0	0

- With probabilities from 0.0 to 0.9, we calculated the 3 metrics -accuracy, sensitivity and specificity.

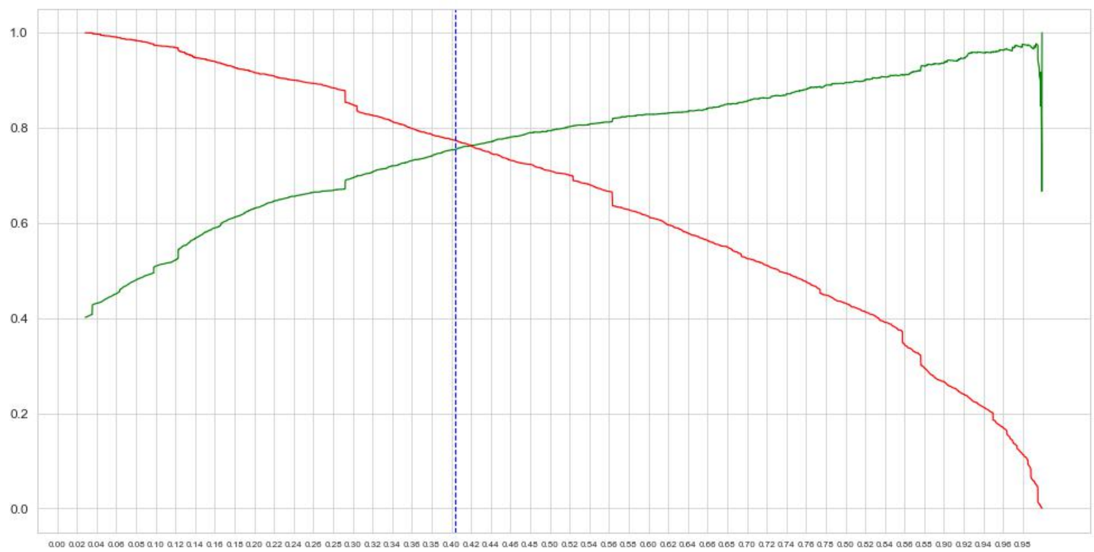
	prob	accuracy	sensi	speci	Precision	Recall
0.0	0.0	0.381262	1.000000	0.000000	1.000000	0.000000
0.1	0.1	0.632653	0.973236	0.422789	0.509554	0.973236
0.2	0.2	0.763760	0.916058	0.669915	0.631006	0.916058
0.3	0.3	0.800402	0.848743	0.770615	0.695118	0.848743
0.4	0.4	0.817718	0.775345	0.843828	0.753646	0.775345
0.5	0.5	0.818955	0.709246	0.886557	0.793917	0.709246
0.6	0.6	0.804267	0.613950	0.921539	0.828228	0.613950
0.7	0.7	0.785250	0.525142	0.945527	0.855915	0.525142
0.8	0.8	0.764069	0.431468	0.969015	0.895623	0.431468
0.9	0.9	0.713358	0.266423	0.988756	0.935897	0.266423

- To make predictions on the train dataset, optimum cutoff of 0.35 was found from the intersection of sensitivity, specificity and accuracy as shown in below figure:

```
<Figure size 1800x864 with 0 Axes>
```



- To make predictions on the test dataset, optimum cutoff was considered as obtained from Precision recall graph of the train dataset as shown below figure:



Inferences:

- Based on Precision- Recall Trade off curve, the cutoff point seems to 0.40 We will use this threshold value for Test Data Evaluation

7 Prediction On Test Datastet

After finalizing the optimum cutoff and calculating the metrics on train set, we predicted the data on test data set. Below are the observations:

Train Data:

```
In [173]: lg_metrics(s)

Model Accuracy value is              :  80.71 %
Model Sensitivity value is           :  81.79 %
Model Specificity value is           :  80.03 %
Model Precision value is             :  71.63 %
Model Recall value is                :  81.79 %
Model True Positive Rate (TPR)       :  81.79 %
Model False Positive Rate (FPR)      :  19.97 %
Model Poitive Prediction Value is    :  71.63 %
Model Negative Prediction value is   :  87.71 %
```

```
In [174]: # Classification Record : Precision, Recall and F1 Score
          print(classification_report( y_train_pred_final['Converted_IND'], y_train_pred_final['final_predicted_1(0.35)'] ) )

              precision    recall  f1-score   support

           0       0.88      0.80      0.84      4002
           1       0.72      0.82      0.76      2466

    accuracy                           0.81      6468
   macro avg       0.80      0.81      0.80      6468
weighted avg       0.82      0.81      0.81      6468
```

```
In [175]: print("F1 Score: {}".format(f1_score(y_train_pred_final['Converted_IND'], y_train_pred_final['final_predicted_1(0.35)']

F1 Score: 0.7637258614161304
```

Test Data:

```
In [195]: lg_metrics(s)

Model Accuracy value is            :  80.7 %
Model Sensitivity value is         :  81.83 %
Model Specificity value is         :  79.96 %
Model Precision value is           :  72.73 %
Model Recall value is              :  81.83 %
Model True Positive Rate (TPR)     :  81.83 %
Model False Positive Rate (FPR)    :  20.04 %
Model Poitive Prediction Value is  :  72.73 %
Model Negative Prediction value is :  87.08 %
```

```
In [197]: # Classification Record : Precision, Recall and F1 Score
print(classification_report( y_pred_final['Converted_IND'], y_pred_final['final_predicted'] ) )

              precision    recall  f1-score   support

           0       0.87      0.80      0.83      1677
           1       0.73      0.82      0.77      1095

    accuracy                           0.81      2772
   macro avg       0.80      0.81      0.80      2772
weighted avg       0.81      0.81      0.81      2772
```

Inferences:

The sensitivity value on Test data is 81.83% vs 80.79% in Train data. The accuracy values is 80.7%. It shows that model is performing well in test data set also and is not over-trained.

8 Final Model Feature

Let's look into final model features and coefficients

```
ut[199]: Do Not Email                              -1.18
         Total Time Spent on Website                1.07
         Lead Origin_Landing Page Submission       -1.02
         Lead Origin_Lead Add Form                  2.80
         Lead Source_Olark Chat                     1.10
         Lead Source_Welingak Website               2.46
         Last Activity_Email Opened                 0.73
         Last Activity_Olark Chat Conversation     -0.61
         Last Activity_Other Activity               2.24
         Last Activity_SMS Sent                     1.87
         Last Activity_Unreachable                  0.85
         Last Activity_Unsubscribed                 1.39
         Specialization_Hospitality Management     -1.00
         Specialization_Others                     -0.98
         Occupation_Unknown                        -1.08
         Occupation_Working Professional            2.40
         dtype: float64
```

<Figure size 864x576 with 0 Axes>