Lead Score Case Summary Report:

Inspection of Data:

1. We have checked the dimensions of data provided in Leads.csv using df.shape, excel is having 9240 rows and 37 columns.

2. Using df.info command, we have checked the datatypes of 37 columns, it's having datatype in int, float and object.

Cleansing of Data:

1. Using df.duplicated() command, we have checked the whether the provided data is having any duplicate or not. There's no duplicate record.

2. Using code - round(100*(df.isnull().sum())/len(df), 2).sort_values(ascending=False), we have tried to find the percentage of missing values a column is having and eventually dropped the columns having missing values more than 30%, same exercise has been performed 1 more time and columns having more than 15% of missing value has been dropped.

3. We have performed imputation on columns - "City", "Specialization" with "Select" to let the clients know where selections of city and specializations are missing.

4. For columns like "Total Visits", "Page Views Per Visit", "Last Activity", "Lead Source", we have imputed it with the mode of variable.

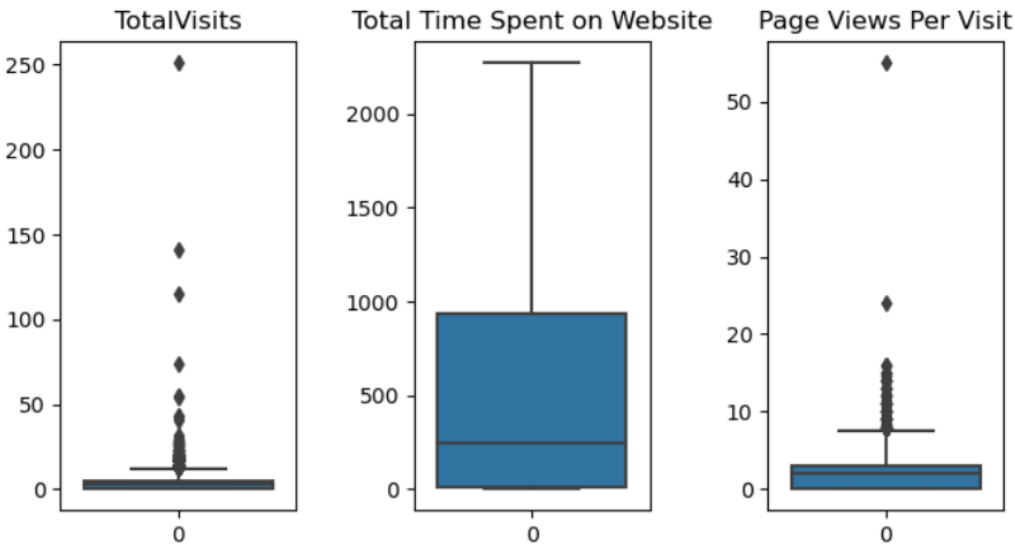
5. We have dropped the columns containing same single value as it won't useful for analysing of data, using below code:

   Leads = Leads.drop(['Magazine', 'Receive More Updates About Our Courses', 'Update me on Supply Chain Content', 'Get updates on DM Content', 'I agree to pay the amount through cheque'], axis=1)

6. We have dropped the columns containing maximum value as 'No' as it won't useful for analysing of data.

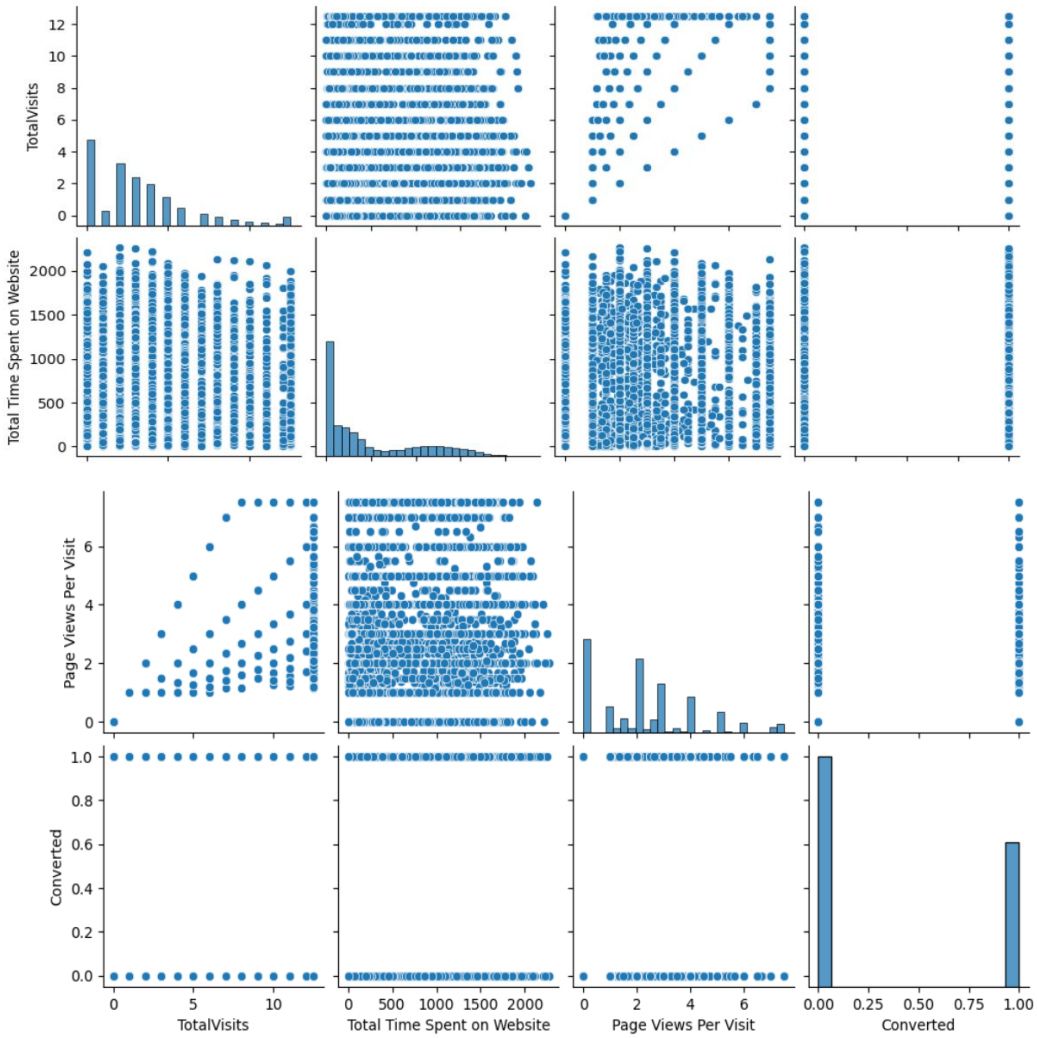
Checking for Outliers:

1. We have checked for outliers in continuous variables i.e. 'Total Visits', 'Total Time Spent on Website', 'Page Views Per Visit' and we can see for variables 'Total Visits' and 'Page Views Per Visit', there are outliers present towards upper level, hence we have capped the outliers with "Quartile3+1.5*IQR".

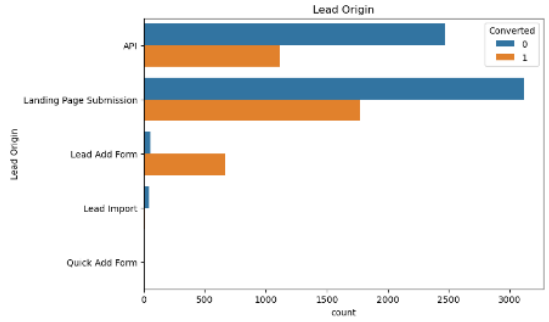


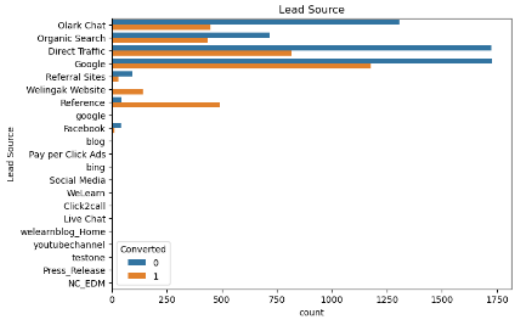
Visualizing the Data:

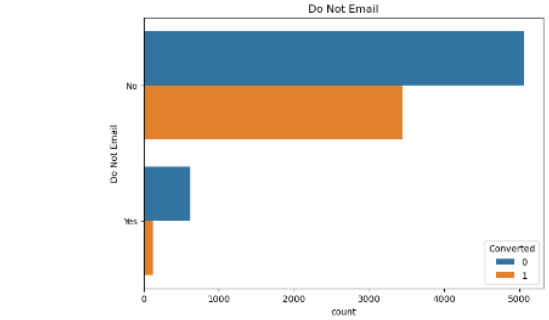
1. We have created pairplot for all numeric variables:

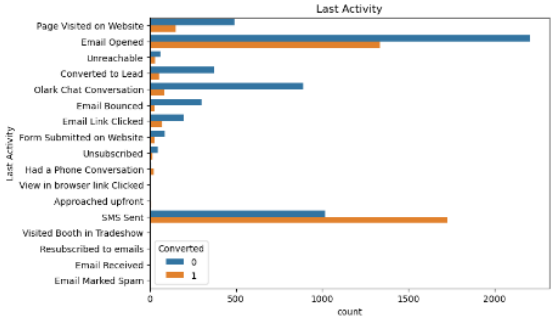


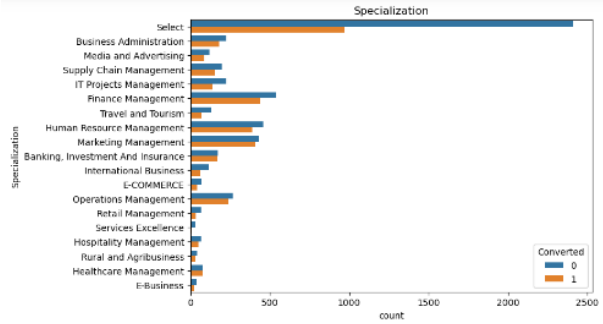
2. For all categorical variables, we have created countplot :

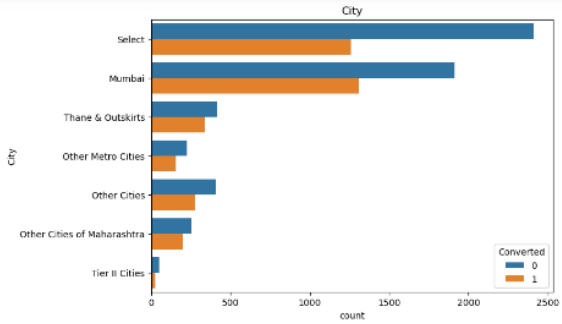


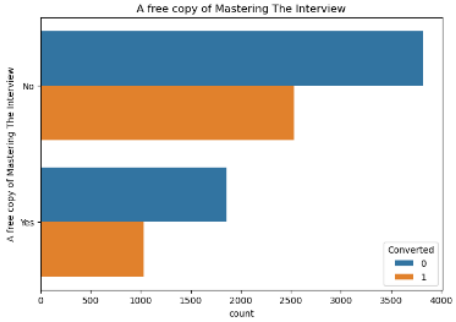


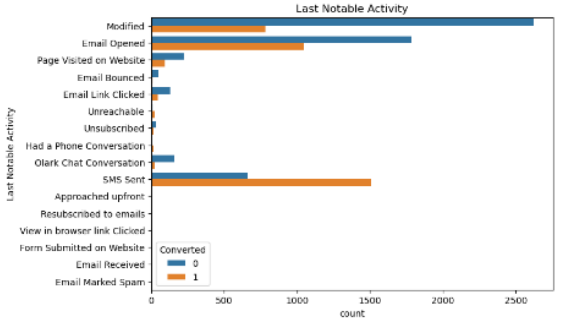












<u>Data preparation:</u>

1. We have converted the values for binary variables such as 'Do Not Email' & 'A free copy of Mastering The Interview' to 0/1 from Yes/No.

Out[36]:

	Prospect ID	Lead Number	Lead Origin	Lead Source	Do Not Email	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Last Activity	Specialization	City	A free copy of Mastering The Interview	Last Notable Activity
0	7927b2df-8bba-4d29-b9a2-b6e0beafe620	660737	API	Olark Chat	0	0	0.0	0	0.0	Page Visited on Website	Select	Select	0	Modified
1	2a272436-5132-4136-86fa-dcc88c88f482	660728	API	Organic Search	0	0	5.0	674	2.5	Email Opened	Select	Select	0	Email Opened
2	8cc8c611-a219-4f35-ad23-fdfd2656bd8a	660727	Landing Page Submission	Direct Traffic	0	1	2.0	1532	2.0	Email Opened	Business Administration	Mumbai	1	Email Opened
3	0cc2df48-7cf4-4e39-9de9-19797f9b38cc	660719	Landing Page Submission	Direct Traffic	0	0	1.0	305	1.0	Unreachable	Media and Advertising	Mumbai	0	Modified
4	3256f628-e534-4826-9d63-4a8b88782852	660681	Landing Page Submission	Google	0	1	2.0	1428	1.0	Converted to Lead	Select	Mumbai	0	Modified

2. We have created the dummy variables for categorical variables such as “City” & “Specialization” and dropped the dummy variables having “Select”.

In [40]:
```
# Creating dummy variables for 'Specialization' and 'City' columns and dropping the 'Select' columns of both the variables.

# Creating dummy variables for the variable 'Specialization'
SP = pd.get_dummies(Leads['Specialization'], prefix='Specialization', dtype = int)
# Dropping MultipleLines_No phone service column
SP1 = SP.drop(['Specialization_Select'], axis = 1)
#Adding the results to the master dataframe
Leads = pd.concat([Leads,SP1], axis=1)

# Creating dummy variables for the variable 'City'.
City = pd.get_dummies(Leads['City'], prefix='City', dtype = int)
City1 = City.drop(['City_Select'], axis = 1)
# Adding the results to the master dataframe
Leads = pd.concat([Leads,City1], axis=1)
```

3. We have dropped the repeated variables as well.

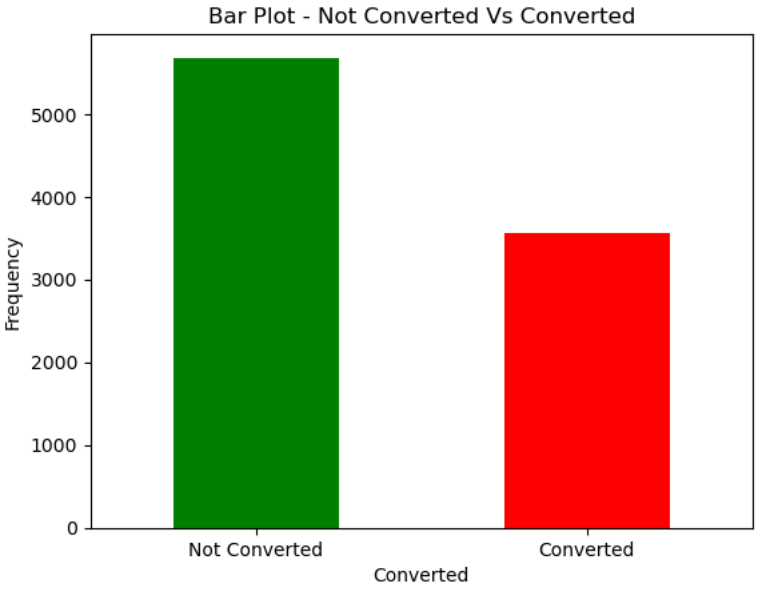
Test – Train Split:

1. We have split the data into training set & testing set i.e. 'TotalVisits','Total Time Spent on Website','Page Views Per Visit' are training sets and find out the conversion rate using below code, which is 38%:

In [53]:
```
### Checking the Conversion Rate
converted = (sum(Leads['Converted'])/len(Leads['Converted'].index))*100
converted
```

Out[53]: 38.53896103896104

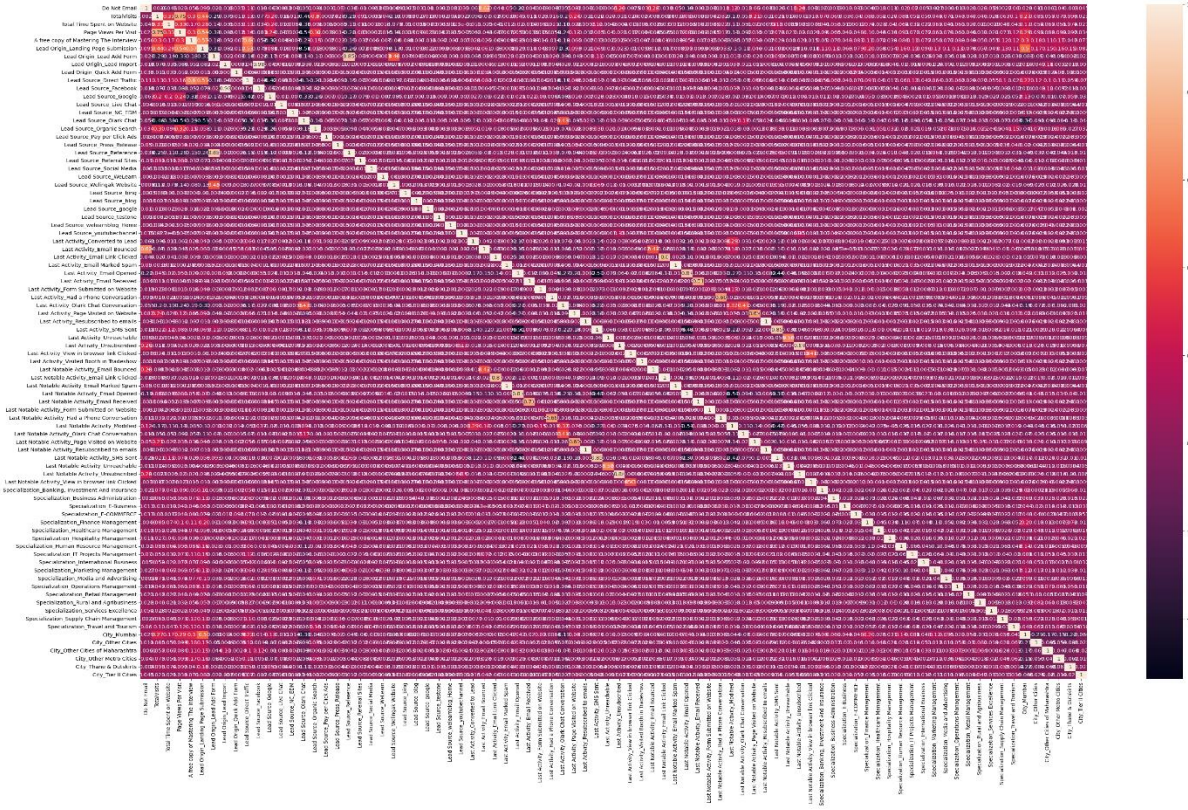
2. We have plotted the graph for Converted vs. Not Converted as below:



We have around 38% conversion rate

Looking for Correlations:

1. Since there are lot of variables present, we could not able to get much information about correlation matrix.



Model building:

1. We have created our first multivariate logistic regression model using all features present in the dataset, below are the summary output:

	coef	std err	z	P>|z|	[0.025	0.975]
const	3.916e+15	4.58e+07	8.54e+07	0.000	3.92e+15	3.92e+15
Do Not Email	-7.082e+14	4.43e+06	-1.6e+08	0.000	-7.08e+14	-7.08e+14
TotalVisits	1.673e+14	1.36e+06	1.23e+08	0.000	1.67e+14	1.67e+14
Total Time Spent on Website	4.803e+13	9.51e+05	5.05e+07	0.000	4.8e+13	4.8e+13
Page Views Per Visit	-6.935e+13	1.49e+06	-4.66e+07	0.000	-6.93e+13	-6.93e+13
A free copy of Mastering The Interview	-3.387e+13	2.71e+06	-1.25e+07	0.000	-3.39e+13	-3.39e+13
Lead Origin_Landing Page Submission	-8.298e+14	3.68e+06	-2.25e+08	0.000	-8.3e+14	-8.3e+14
Lead Origin_Lead Add Form	7.191e+14	1.3e+07	5.52e+07	0.000	7.19e+14	7.19e+14
Lead Origin_Lead Import	2.906e+14	4.8e+07	6.06e+06	0.000	2.91e+14	2.91e+14
Lead Origin_Quick Add Form	2.528e+15	6.8e+07	3.72e+07	0.000	2.53e+15	2.53e+15

2. We have created our 2nd logistic model using RFE and selected the 15 features, output as below:

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.2162	0.067	3.209	0.001	0.084	0.348
Do Not Email	-1.6571	0.163	-10.148	0.000	-1.977	-1.337
Total Time Spent on Website	1.1176	0.038	29.131	0.000	1.042	1.193
Lead Origin_Lead Add Form	4.0163	0.185	21.762	0.000	3.655	4.378
Lead Source_Olark Chat	1.1546	0.100	11.535	0.000	0.958	1.351
Lead Source_Welingak Website	1.6810	0.745	2.255	0.024	0.220	3.142
Last Activity_Had a Phone Conversation	1.6643	0.660	2.521	0.012	0.370	2.958
Last Activity_Olark Chat Conversation	-1.0867	0.188	-5.782	0.000	-1.455	-0.718
Last Notable Activity_Email Link Clicked	-1.8547	0.259	-7.175	0.000	-2.361	-1.348
Last Notable Activity_Email Opened	-1.3446	0.084	-16.035	0.000	-1.509	-1.180
Last Notable Activity_Modified	-1.8740	0.092	-20.392	0.000	-2.054	-1.694
Last Notable Activity_Olark Chat Conversation	-1.6501	0.365	-4.522	0.000	-2.365	-0.935
Last Notable Activity_Page Visited on Website	-1.5824	0.184	-8.589	0.000	-1.944	-1.221
Specialization_Finance Management	0.3353	0.105	3.181	0.001	0.129	0.542
Specialization_Healthcare Management	0.4050	0.253	1.603	0.109	-0.090	0.900
Specialization_Rural and Agribusiness	0.7301	0.371	1.969	0.049	0.003	1.457

3. We have run VIFs and created multiple models and dropped columns specialization_healthcare_management, specialization_rural, Lead_Source_Welingak_Website, Last_Activity_Had_a_Phone_Conversation, Specialization_Finance Management and agribusiness as p-value is high. Finally at model 7, p value and VIFs both are low for rest of the features.

   Model 2:

Out[69]:

	Features	VIF
6	Last Activity_Olark Chat Conversation	1.90
3	Lead Source_Olark Chat	1.65
9	Last Notable Activity_Modified	1.56
2	Lead Origin_Lead Add Form	1.31
10	Last Notable Activity_Olark Chat Conversation	1.31
4	Lead Source_Welingak Website	1.22
1	Total Time Spent on Website	1.18
8	Last Notable Activity_Email Opened	1.15
0	Do Not Email	1.11
12	Specialization_Finance Management	1.11
7	Last Notable Activity_Email Link Clicked	1.02
11	Last Notable Activity_Page Visited on Website	1.02
13	Specialization_Healthcare Management	1.02
5	Last Activity_Had a Phone Conversation	1.01
14	Specialization_Rural and Agribusiness	1.01

Final model 7:

Out[84]:

	Features	VIF
4	Last Activity_Olark Chat Conversation	1.89
3	Lead Source_Olark Chat	1.65
7	Last Notable Activity_Modified	1.51
8	Last Notable Activity_Olark Chat Conversation	1.30
1	Total Time Spent on Website	1.17
0	Do Not Email	1.11
6	Last Notable Activity_Email Opened	1.09
2	Lead Origin_Lead Add Form	1.08
5	Last Notable Activity_Email Link Clicked	1.02
9	Last Notable Activity_Page Visited on Website	1.02

Model Evaluation:

1. We have created a new column as Predicted to flag the data as 1 if Converted_Prob is greater than 0.5.

Out[88]:

	Converted	Converted_Prob	ProsID	predicted
0	0	0.279202	1871	0
1	0	0.257711	6795	0
2	0	0.324738	3516	0
3	0	0.859075	8105	1
4	0	0.186040	3934	0

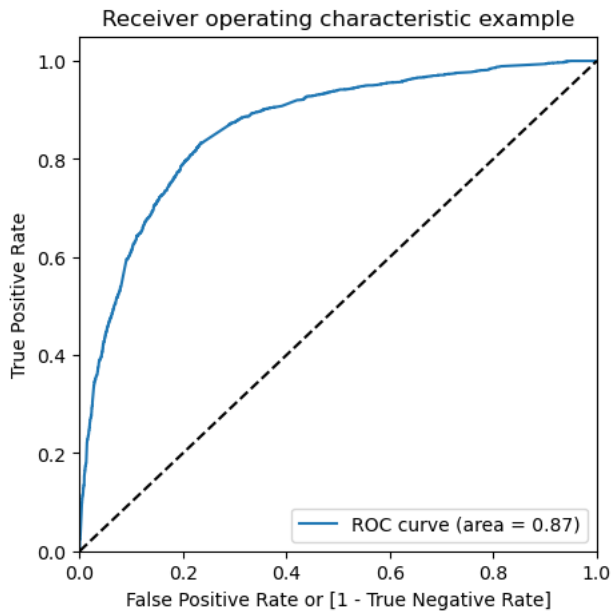
Metrics beyond simply accuracy:

```
In [92]: 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 [93]: # Let's see the sensitivity of our logistic regression model
         TP / float(TP+FN)

Out[93]: 0.6695052716950527

In [94]: # Let us calculate specificity
         TN / float(TN+FP)

Out[94]: 0.8740629685157422

In [95]: # Calculate false postive rate - predicting converted when customer does not have converted
         print(FP/ float(TN+FP))

         0.12593703148425786

In [96]: # positive predictive value
         print (TP / float(TP+FP))

         0.7661252900232018

In [97]: # Negative predictive value
         print (TN / float(TN+ FN))

         0.8110364015766288
```

ROC Curve:

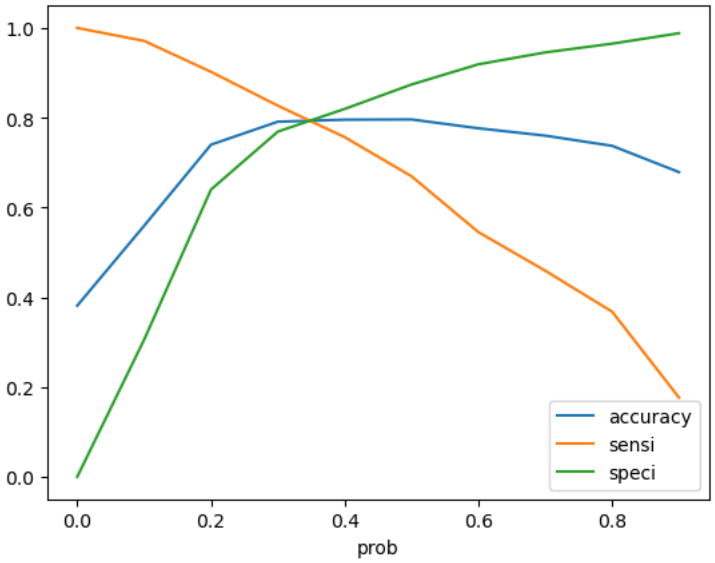
An ROC curve demonstrates several things:

- It shows the trade off 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.



Finding the Optimal Cutoff Point:

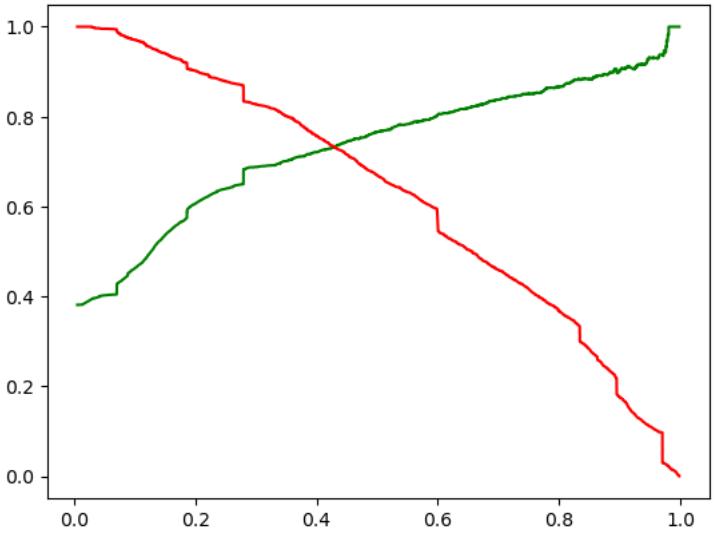
1. Optimal cutoff probability is that prob where we get balanced sensitivity and specificity.



From the curve above, 0.37 is the optimum point to take it as a cutoff probability.

Precision and Recall:

Trade off Curve



Making prediction on test set:

```
In [140]: TP = confusion2[1,1] # true positive
          TN = confusion2[0,0] # true negatives
          FP = confusion2[0,1] # false positives
          FN = confusion2[1,0] # false negatives

In [141]: # Let's see the sensitivity of our logistic regression model
          TP / float(TP+FN)

Out[141]: 0.7808219178082192

In [142]: # Let us calculate specificity
          TN / float(TN+FP)

Out[142]: 0.8127608825283243

In [143]: # Let's calculate Precision Score
          precision_score(y_pred_final.Converted, y_pred_final.final_predicted)

Out[143]: 0.7313943541488451

In [144]: # Let's calculate Recall Score
          recall_score(y_pred_final.Converted, y_pred_final.final_predicted)

Out[144]: 0.7808219178082192
```

Assigning lead score value:

1. We have derived lead score value from previously calculated converted_prob and using y_train_pred_final. In the end, each row is having a lead score value.

Lead_Score_df

	Do Not Email	Do Not Call	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	...	City	Asymmetrique Activity Index	Asymmetrique Profile Index	Asymmetrique Activity Score	Asymmetrique Profile Score	I agree to pay the amount through cheque	A free copy of Mastering The Interview	Last Notable Activity	ProsID	Lead Score
	No	No	0	0.0	0	0.00	...	Select	02.Medium	02.Medium	15.0	15.0	No	No	Modified	0	19
	No	No	0	5.0	674	2.50	...	Select	02.Medium	02.Medium	15.0	15.0	No	No	Email Opened	1	34
	No	No	1	2.0	1532	2.00	...	Mumbai	02.Medium	01.High	14.0	20.0	No	Yes	Email Opened	2	75