LEAD SCORING CASE STUDY

BY

SAI PRIYA & RAOOF

TABLE OF CONTENT

Problem Statement

- An education company named sells online courses to industry professionals,
- Once these people land on the website and fill the form with either their mail address or phone number they are marked as lead,
- Once these leads are acquired, employees from the sales team starts contacting them, then if the lead enrolls into the course they are classified as converted,
- The percentage of lead conversion is about 30, that is for every 100 leads contacted only 30 were successfully converted,
- So as to make this process more efficient, the company wishes to identify the most potential leads, also known as ‘Hot Leads’ and thereby directing their resources towards the hot leads,
- The plan is to achieve lead conversion rate of 80 percent and create a model which could help to do so by providing the a lead score between 0 and 100, where higher is the score hotter the lead is,
- And make this model deployable in future.

APPROACH TO SOLUTION

This problem can be solved in the following manner using CRISP-DM→

- Data reading and understanding (basic loading the data and looking at its parameters)
- Data cleaning (treating null values and outliers, performing sanity checks, encoding/decoding)
- EDA (exploratory data analysis)
- Model Preparation (Data pre processing)
- Model Building (Training the data)
- Model Evaluation (Testing the data)
- Model Deployment

Data reading and Data understanding

- Data reading basically means reading the structured data (CSV in this case) into a python data frame structure.
- Once we're done with reading the data into a data frame,
- We will analyze the shape, numerical summary, basic information and various metrics for our data frame as shown in figure below

`As we can see our data frame has 9240 rows and 37 columns`

```
In [6]: # Checking the shape of dataset

        lead_df.shape

Out[6]: (9240, 37)
```

Further we can performed numerical summary and saw the basic information regarding our

Data frame, during which we observed that all of the columns had correct data types associated with them, there were some unwanted columns (noise) and there were certain columns with plethora of missing values, all of these issues we shall deal with during the process of data cleaning.

Data Cleaning with EDA

- First we removed all columns with more than 40 percent missing data or null values,
- Then we checked for duplicate values in data-set after which we imputed the null values with appropriate value and simultaneously performed univariate and bivariate analysis on columns

```python
# Checking percentage of Null values present in each column

(((lead_df.isnull().sum())*100)/lead_df.shape[0]).sort_values(ascending = False)
```

```
How did you hear about X Education               78.463203
Lead Profile                                     74.188312
Lead Quality                                     51.590909
Asymmetrique Profile Score                       45.649351
Asymmetrique Activity Score                      45.649351
Asymmetrique Profile Index                       45.649351
Asymmetrique Activity Index                      45.649351
City                                             39.707792
Specialization                                   36.580087
Tags                                             36.287879
What matters most to you in choosing a course    29.318182
What is your current occupation                  29.112554
Country                                          26.634199
TotalVisits                                       1.482684
Page Views Per Visit                              1.482684
Last Activity                                     1.114719
Lead Source                                       0.389610
Do Not Call                                       0.000000
Converted                                         0.000000
Total Time Spent on Website                       0.000000
Do Not Email                                      0.000000
Last Notable Activity                             0.000000
X Education Forums                                0.000000
Search                                            0.000000
Magazine                                          0.000000
Newspaper Article                                 0.000000
A free copy of Mastering The Interview            0.000000
Newspaper                                         0.000000
Digital Advertisement                             0.000000
Through Recommendations                           0.000000
Receive More Updates About Our Courses            0.000000
Update me on Supply Chain Content                 0.000000
Get updates on DM Content                         0.000000
I agree to pay the amount through cheque          0.000000
Lead Origin                                       0.000000
dtype: float64
```

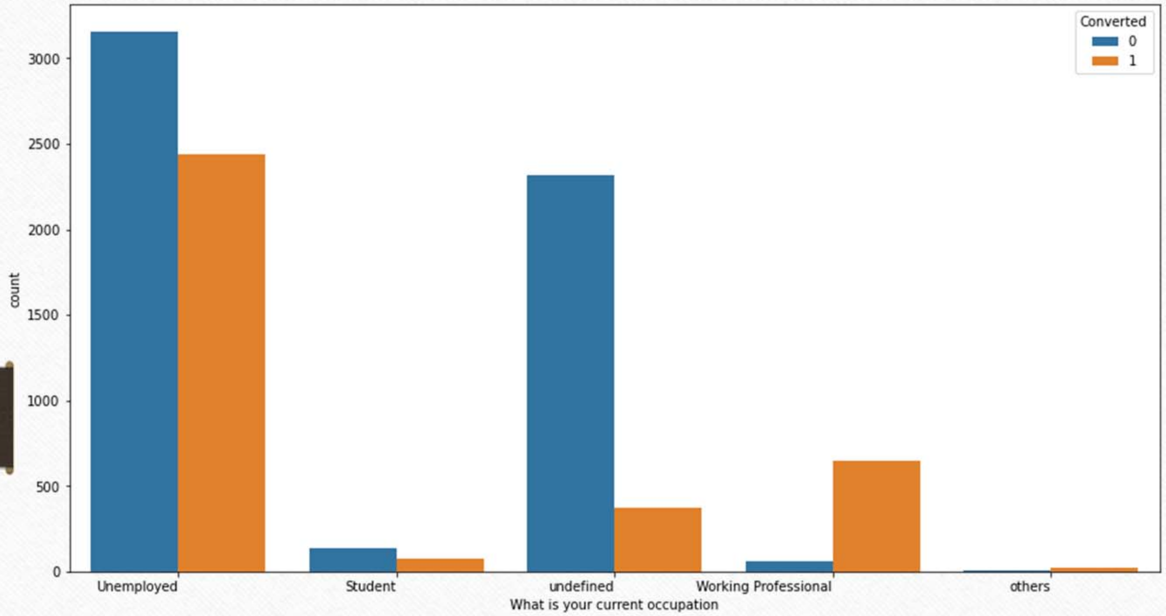
Before cleaning ←

After cleaning →

```python
# Here we can see that our function has worked successfully

((lead_df.isnull().sum()/lead_df.shape[0])*100).sort_values(ascending = False )
```

```
City                                             39.707792
Specialization                                   36.580087
Tags                                             36.287879
What matters most to you in choosing a course    29.318182
What is your current occupation                  29.112554
Country                                          26.634199
TotalVisits                                       1.482684
Page Views Per Visit                              1.482684
Last Activity                                     1.114719
Lead Source                                       0.389610
Last Notable Activity                             0.000000
Do Not Email                                      0.000000
Do Not Call                                       0.000000
Converted                                         0.000000
Total Time Spent on Website                       0.000000
Search                                            0.000000
A free copy of Mastering The Interview            0.000000
Magazine                                          0.000000
Newspaper Article                                 0.000000
X Education Forums                                0.000000
Newspaper                                         0.000000
Digital Advertisement                             0.000000
Through Recommendations                           0.000000
Receive More Updates About Our Courses            0.000000
Update me on Supply Chain Content                 0.000000
Get updates on DM Content                         0.000000
I agree to pay the amount through cheque          0.000000
Lead Origin                                       0.000000
dtype: float64
```



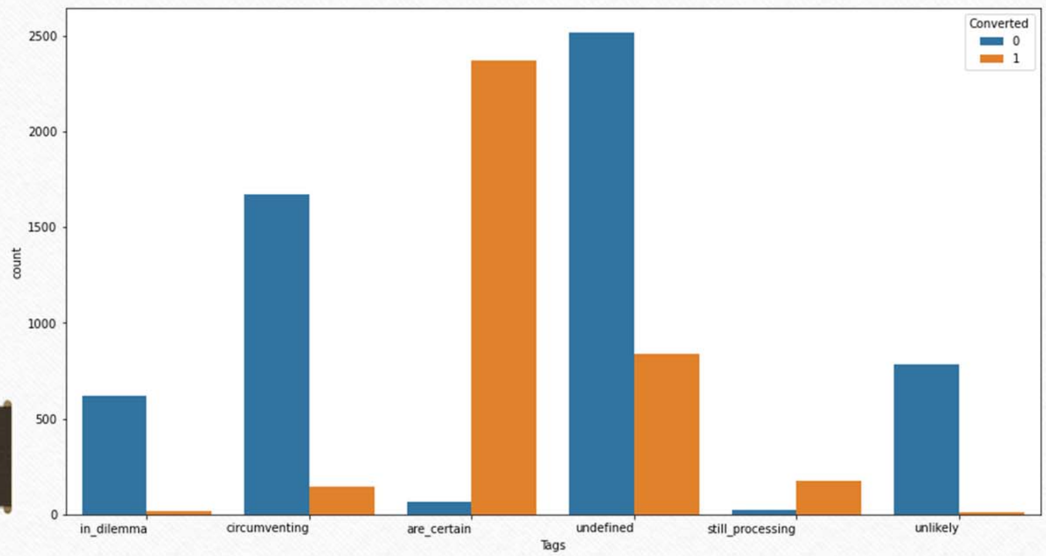
The variable 'What is your current occupation' needed a through cleaning too, so first we imputed the missing values with label 'undefined' , Then we clubbed the variables having counts less than 100 together,

```
ConvRate('What is your current occupation')
```

	What is your current occupation	Conversion_rate
3	Working Professional	91.64
4	others	73.53
0	Unemployed	43.59
1	Student	37.14
2	undefined	13.75

As we can observe from the plot and conversion rate ,'Working Professional' has a 'phenomenal conversion rate' of more than 91 percent, whereas 'students' and 'others' are seemingly on the lower end of the spectrum . While the most of the leads are unemployed.

The conversion Rate after clubbing



The variable `TAGS` needed a through cleaning, so first we imputed the missing values with label `undefined` as it seemed most appropriate,
Then we segmented the variables into various labels based upon the attitude of the lead towards marketing team,

The conversion Rate after Segmenting →

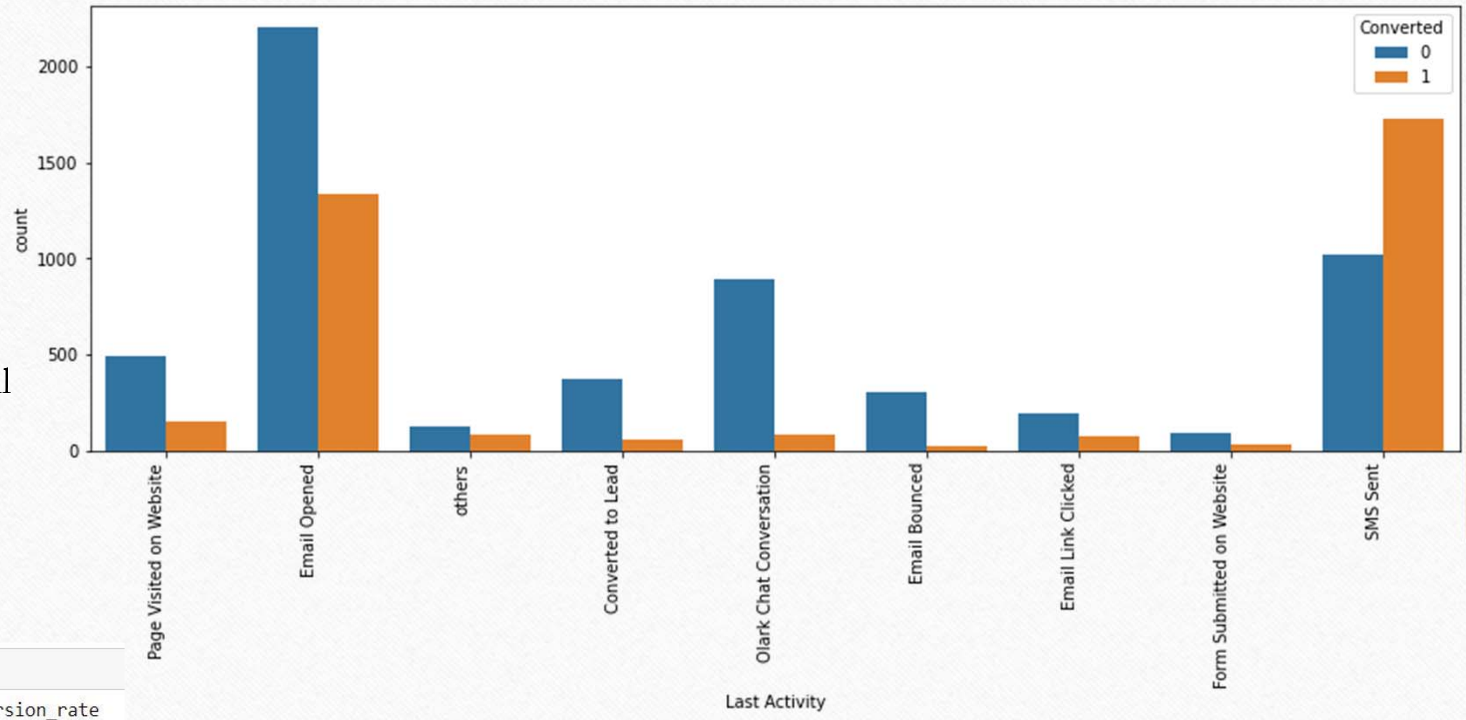
```
ConvRate('Tags')
```

	Tags	Conversion_rate
2	are_certain	97.21
4	still_processing	88.89
3	undefined	24.93
1	circumventing	8.08
0	in_dilemma	2.99
5	unlikely	1.76

As we can observe from the plot and conversion rate →

- We can confirm that 'Certain' leads do tend to successfully convert by a whopping 97 percent rate, and
- As expected 'unlikely' has a very low conversion rate of about 1.76 percent

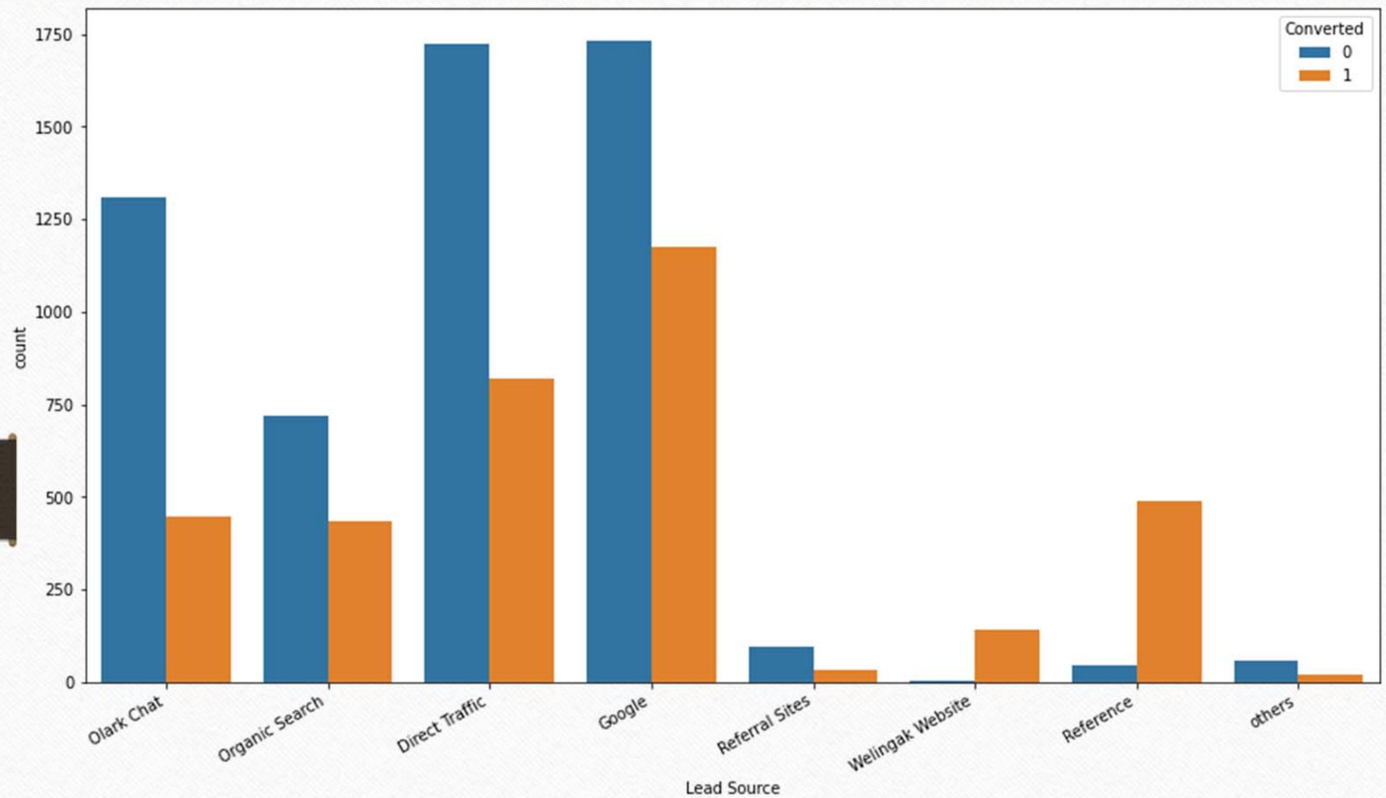
The feature 'Last activity' had some null values which were replaced by the mode of the feature, while all values with small occurrence were clubbed together




```
ConvRate('Last Activity')
```

	Last Activity	Conversion_rate
8	SMS Sent	62.91
2	others	40.98
1	Email Opened	37.68
6	Email Link Clicked	27.34
7	Form Submitted on Website	24.14
0	Page Visited on Website	23.59
3	Converted to Lead	12.62
4	Olark Chat Conversation	8.63
5	Email Bounced	7.98

We can observe that 'SMS sent' is indeed the label with highest conversion rate, 62.9 percent to be precise while, 'Email Bounced' have the lowest, that is roughly 8 percent as expected




```
ConvRate('Lead Source')

         Lead Source  Conversion_rate
5   Welingak Website            98.59
6          Reference            91.76
3             Google            40.43
1     Organic Search            37.78
2     Direct Traffic            32.17
7             others            28.21
0         Olark Chat            25.53
4     Referral Sites            24.80
```

```
count('Lead Source')

Google              2909
Direct Traffic      2543
Olark Chat          1755
Organic Search      1154
Reference            534
Welingak Website     142
Referral Sites       125
others                78
Name: Lead Source, dtype: int64
```

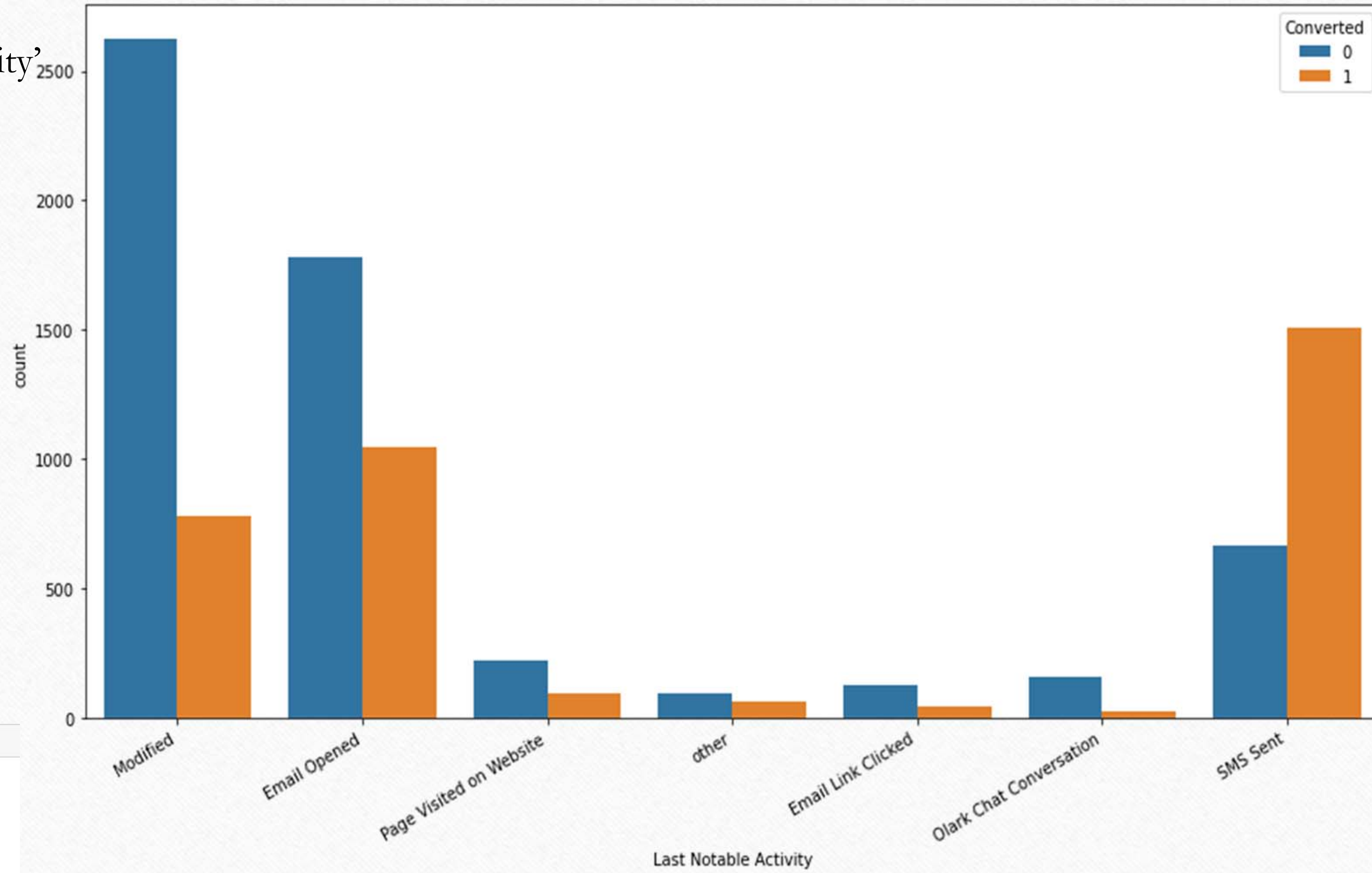
`Welingak website` has a very high conversion rate despite low occurrence, same can be said about reference while google has about 40% of conversion rate.

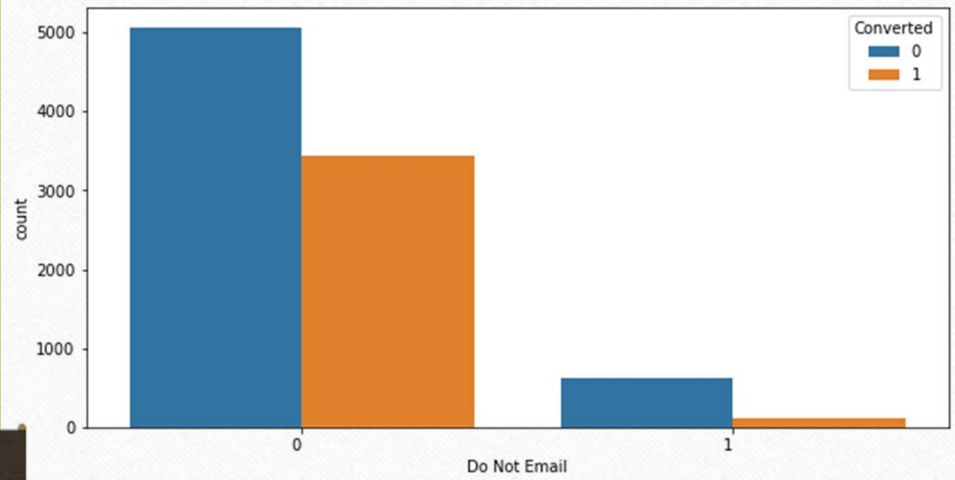
For feature 'last notable activity'

- `SMS sent` and `email opened` have both high occurrence and good conversion rate,
- where as `modified` has the highest occurrence and very low conversion rate.

```
ConvRate('Last Notable Activity')
```

	Last Notable Activity	Conversion_rate
6	SMS Sent	69.43
3	other	39.38
1	Email Opened	36.93
2	Page Visited on Website	29.25
4	Email Link Clicked	26.01
0	Modified	22.98
5	Olark Chat Conversation	13.66





From the analysis of feature ‘Do Not Email’ it can be noted that

1. `SMS sent` and `email opened` have both high occurrence and good conversion rate
2. where as `modified` has the highest occurrence and very low conversion rate

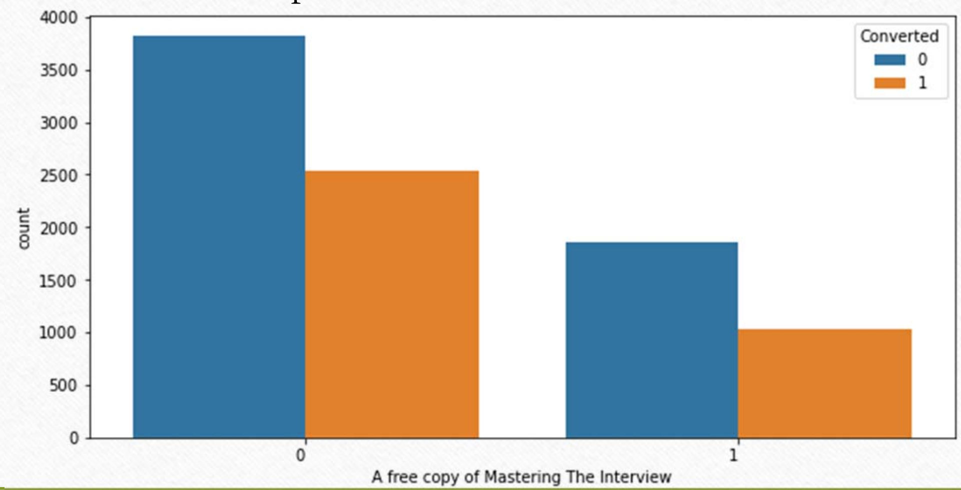
```
ConvRate('Do Not Email')
```

	Do Not Email	Conversion_rate
0	0	40.48
1	1	16.08

Ones who have opted for the free copy have both lower occurrence and conversion rate (though only slightly) than their counterparts.

```
ConvRate('A free copy of Mastering The Interview')
```

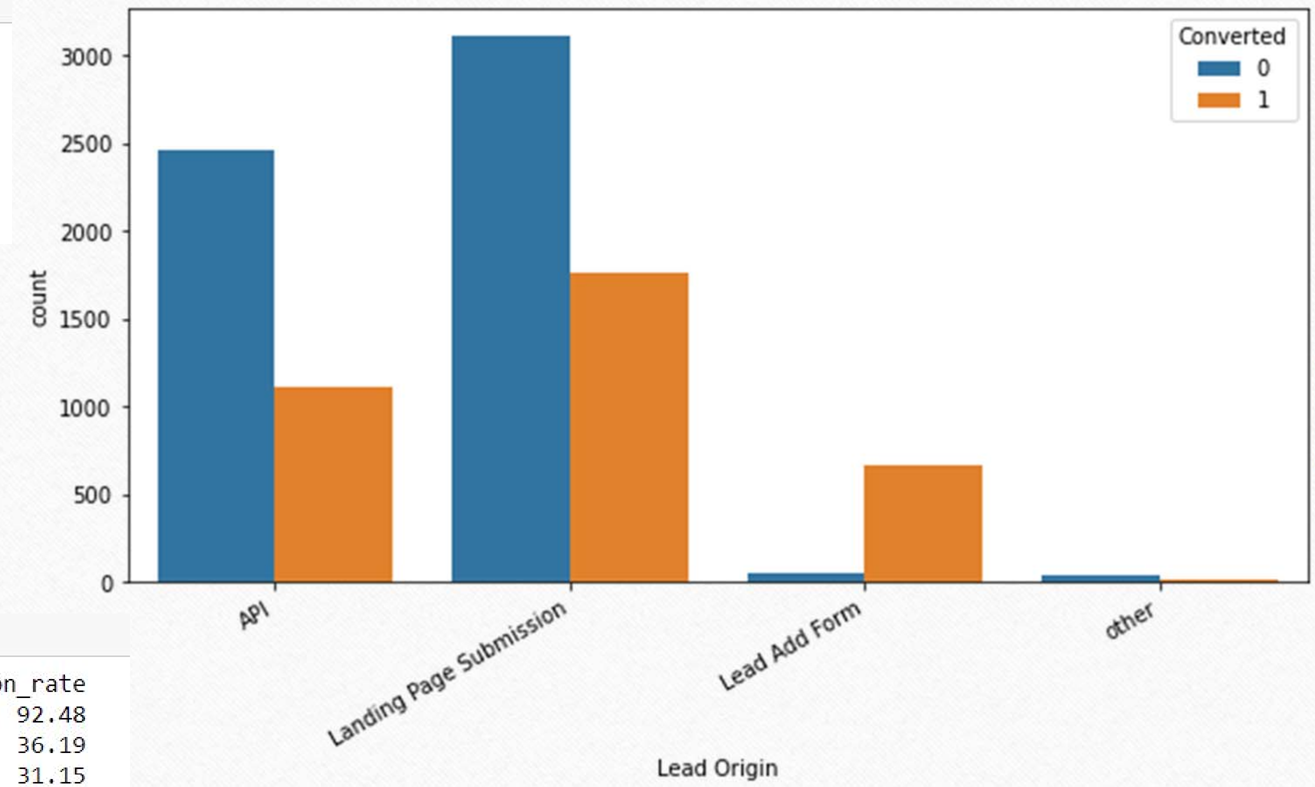
	A free copy of Mastering The Interview	Conversion_rate
0	0	39.85
1	1	35.66



```
count('Lead Origin')
```

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

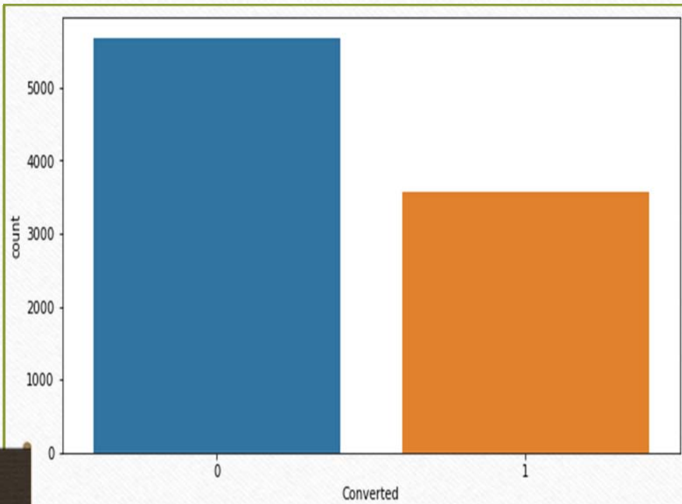
There were no null values to impute but only clubbing of labels with very few occurrence was needed.




```
ConvRate('Lead Origin')
```

	Lead Origin	Conversion_rate
2	Lead Add Form	92.48
1	Landing Page Submission	36.19
0	API	31.15
3	other	25.00

`Landing page submission` has highest occurrence while `lead add form` has lowest.



← This is our target variable

```
count('Converted')   # Analysing our target vriable
```

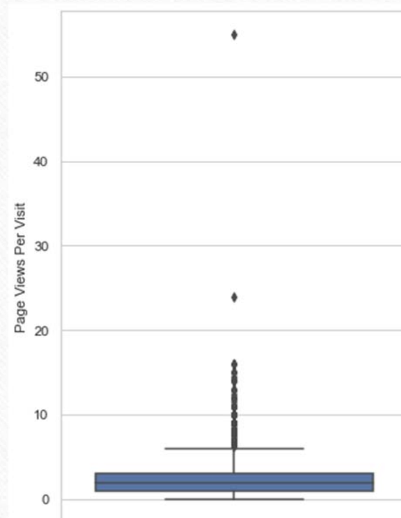
```
0    5679
1    3561
Name: Converted, dtype: int64
```

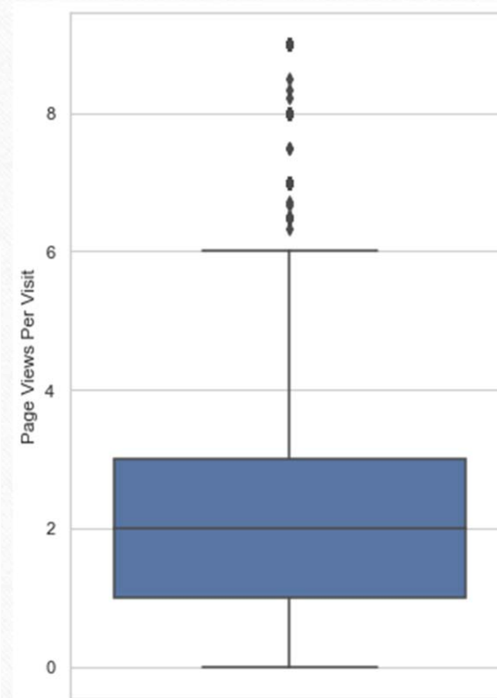
The average rate of conversion for our model is 38.54%. (Target Variable)

Before Outlier treatment →

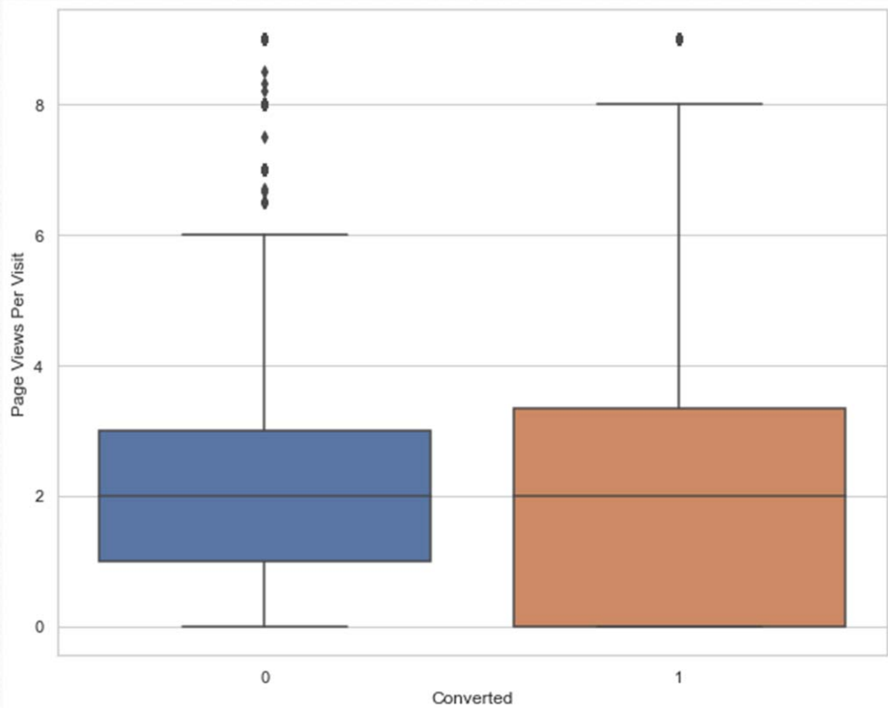
```
lead_df['Page Views Per Visit'].describe(percentiles = [0.01,0.05,0.95,0.99])
```

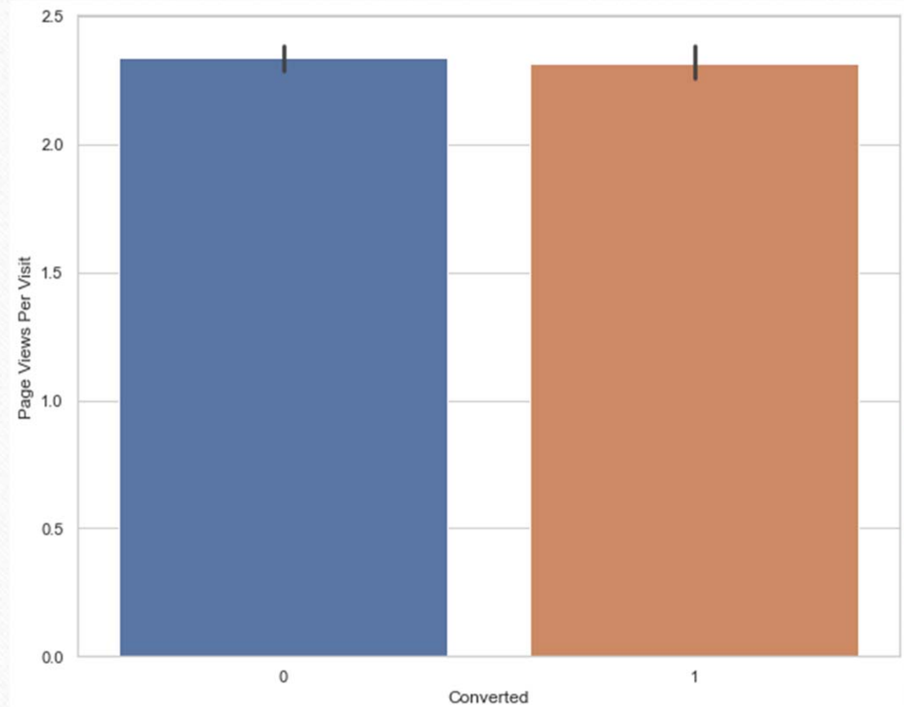
```
count    9103.000000
mean        2.362820
std         2.161418
min         0.000000
1%          0.000000
5%          0.000000
50%         2.000000
95%         6.000000
99%         9.000000
max        55.000000
Name: Page Views Per Visit, dtype: float64
```



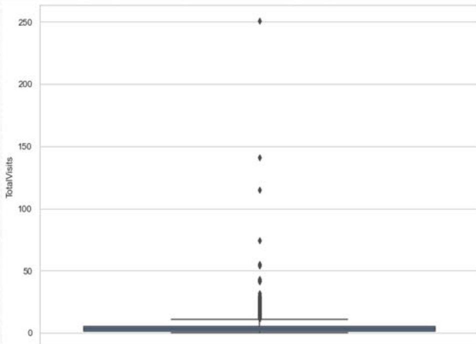


After outlier treatment





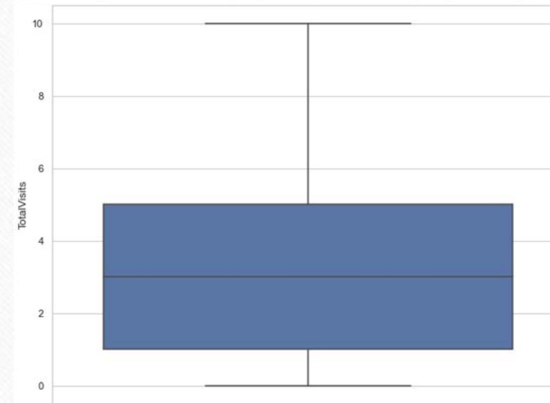
Although the Median is more or less same for both converted and not converted leads, in feature 'Page views per visit' , the IQR of converted is a little larger or more diverse

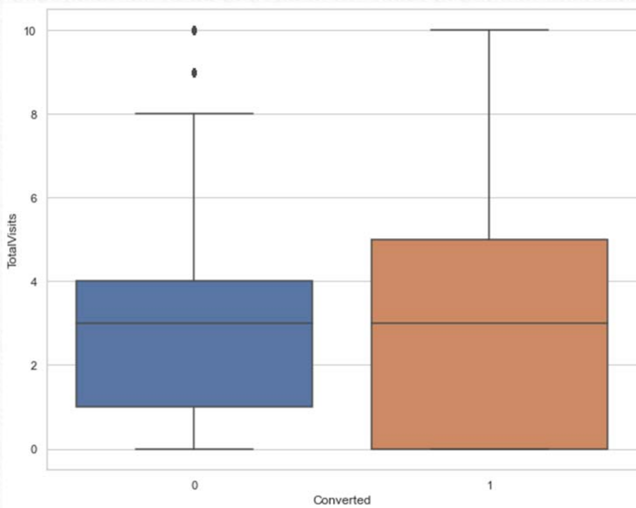


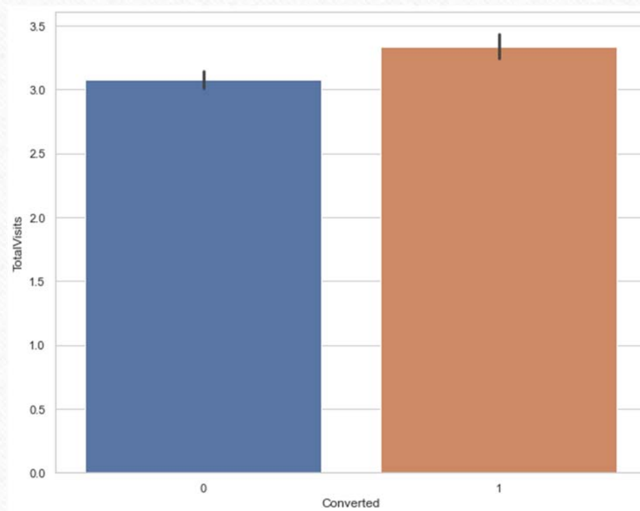
As we can see Converted leads are more for `totalvisits` than unconverted leads

←before treatment

After treatment →

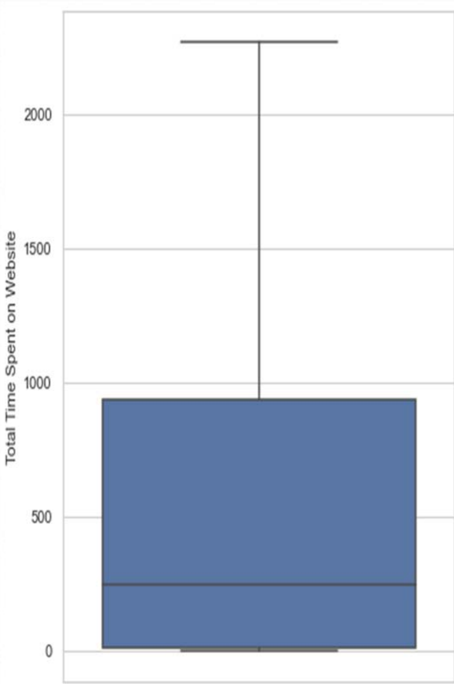


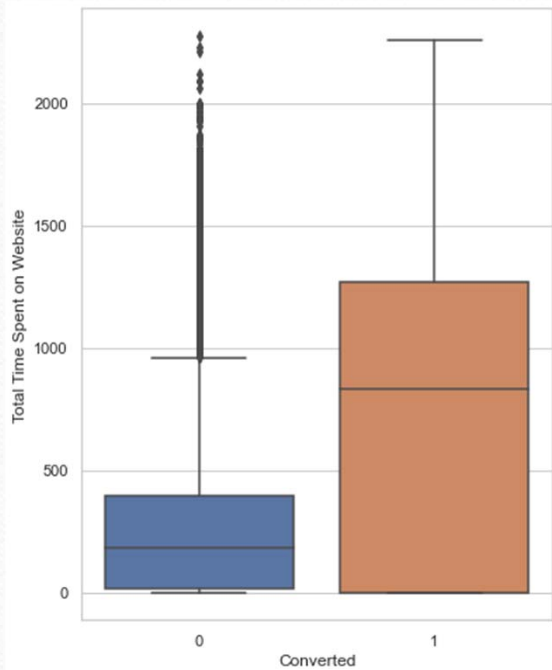


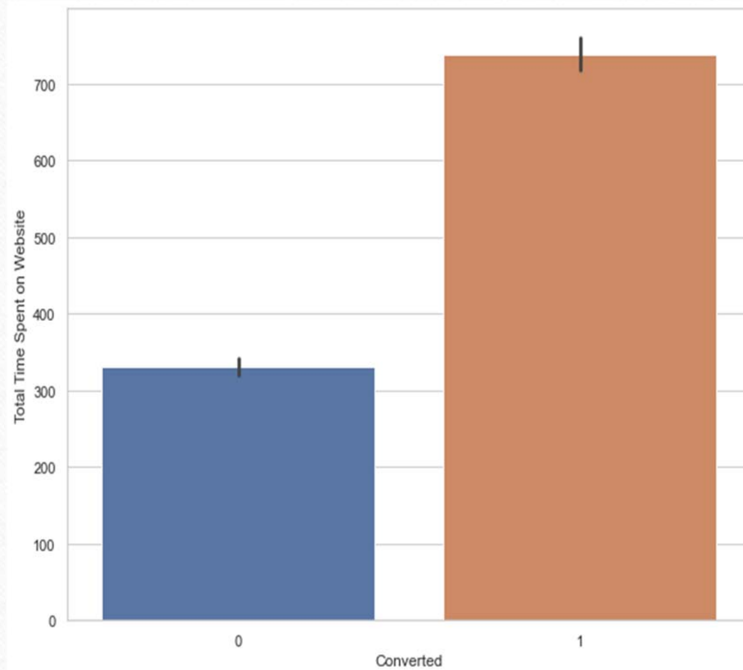



```
: lead_df['TotalVisits'].describe(percentiles

: count    9103.000000
  mean        3.445238
  std         4.854853
  min         0.000000
  1%          0.000000
  5%          0.000000
  50%         3.000000
  75%         5.000000
  80%         5.000000
  85%         6.000000
  90%         7.000000
  95%        10.000000
  99%        17.000000
  max       251.000000
  Name: TotalVisits, dtype: float64
```







Median for converted is very high for `Total Time Spent on Website`, so higher the lead spends time on website more likely they are to convert. Some can be concluded from bar chart

Heatmaps between all numerical columns present in our dataset

From above plot, it is evident that -:

1. \`Total time spent on website\` and \`converted\` are \`positively correlated\`
2. \`Converted\` and \`do not Email\` are \`negatively correlated\`

Model Preparation And Building

```
# Using the class LogisticRegression we will build a function

from sklearn.linear_model import LogisticRegression
#### 2.1 Dealing with Missing data   A.K.A   Null values for categorical columns whilst performing univariate analysis
from sklearn.feature_selection import RFE

lr =  LogisticRegression()

rfe = RFE(lr,)                                   #  using RFE prioritizing 15 variables, to begin with
rfe = rfe.fit(X_train, y_train)
```

←Building 1st model

```
lead_df = pd.get_dummies(lead_df, drop_first = True)
lead_df.info()
```

← Dummy Variable creation

```
# Performing a train-test split on our dataset with 70% to 30% ratios respectively

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

← Test-train split

```
lr4 = sm.GLM(y_train,X_train_rfe, family = sm.families.Binomial())
res = lr4.fit()
res.summary()
```

Generalized Linear Model Regression Results

Dep. Variable:	Converted	No. Observations:	6468
Model:	GLM	Df Residuals:	6455
Model Family:	Binomial	Df Model:	12
Link Function:	logit	Scale:	1.0000
Method:	IRLS	Log-Likelihood:	-1422.0
Date:	Wed, 13 Oct 2021	Deviance:	2843.9
Time:	19:12:29	Pearson chi2:	7.23e+03
No. Iterations:	7		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	1.7485	0.105	16.622	0.000	1.542	1.955
Total Time Spent on Website	1.0342	0.056	18.430	0.000	0.924	1.144
Lead Origin_Lead Add Form	2.0813	0.241	8.650	0.000	1.610	2.553
Lead Source_Olark Chat	0.4245	0.140	3.024	0.002	0.149	0.700
Do Not Email_1	-1.8709	0.256	-7.317	0.000	-2.372	-1.370
Last Activity_SMS Sent	1.8985	0.118	16.094	0.000	1.667	2.130
What is your current occupation_undefined	-3.7978	0.122	-31.244	0.000	-4.036	-3.560
Tags_circumventing	-5.5569	0.170	-32.615	0.000	-5.891	-5.223
Tags_financial_issues	-3.0636	1.132	-2.707	0.007	-5.282	-0.845
Tags_in_dilemma	-5.1487	0.320	-16.109	0.000	-5.775	-4.522
Tags_unlikely	-5.8930	0.379	-15.538	0.000	-6.636	-5.150
Last Notable Activity_Modified	-1.0712	0.111	-9.646	0.000	-1.289	-0.854
Last Notable Activity_other	1.1386	0.402	2.833	0.005	0.351	1.926

	Features	VIF
10	Last Notable Activity_Modified	1.78
5	What is your current occupation_undefined	1.54
2	Lead Source_Olark Chat	1.42
4	Last Activity_SMS Sent	1.33
3	Do Not Email_1	1.25
0	Total Time Spent on Website	1.22
6	Tags_circumventing	1.19
1	Lead Origin_Lead Add Form	1.17
8	Tags_in_dilemma	1.16
9	Tags_unlikely	1.16
11	Last Notable Activity_other	1.14
7	Tags_financial_issues	1.01

All VIF values are under control

Final Model Metrics and Parameters Before selecting optimal point

```
print(res.params)
```

```
const                                        1.748483
Total Time Spent on Website                  1.034173
Lead Origin_Lead Add Form                    2.081347
Lead Source_Olark Chat                       0.424474
Do Not Email_1                              -1.870891
Last Activity_SMS Sent                       1.898454
What is your current occupation_undefined   -3.797782
Tags_circumventing                          -5.556923
Tags_financial_issues                       -3.063557
Tags_in_dilemma                             -5.148652
Tags_unlikely                               -5.892963
Last Notable Activity_Modified              -1.071245
Last Notable Activity_other                  1.138596
dtype: float64
```

```
# Overall accuracy of model

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

```
0.9135745207173779
```

```
# Elements of Confusion matrix

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

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

```
0.875506893755069
```

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

```
0.9370314842578711
```

```
FP/ float(TN+FP)       # FPR (FALSE POSITIVE RATE)
```

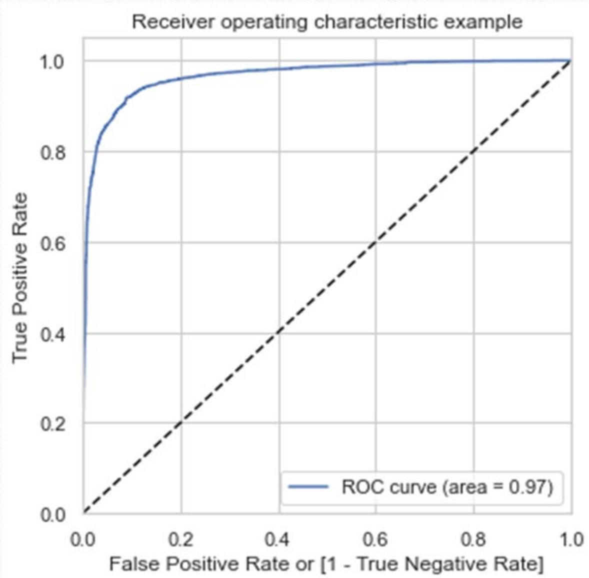
```
0.06296851574212893
```

```
TP / float(TP+FP)      # positive predicted value
```

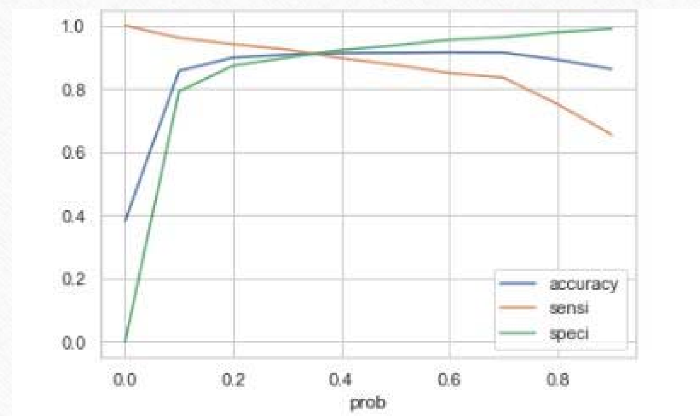
```
0.8954790543343011
```

```
TN / float(TN+ FN)     # negitive predicted value
```

```
0.9243283214197683
```



	prob	accuracy	sensi	speci
0.0	0.0	0.381262	1.000000	0.000000
0.1	0.1	0.856524	0.961476	0.791854
0.2	0.2	0.898887	0.940795	0.873063
0.3	0.3	0.908318	0.923763	0.898801
0.4	0.4	0.912801	0.896999	0.922539
0.5	0.5	0.913575	0.875507	0.937031
0.6	0.6	0.914966	0.849959	0.955022
0.7	0.7	0.914502	0.835361	0.963268
0.8	0.8	0.892239	0.751419	0.979010
0.9	0.9	0.862554	0.655718	0.990005



Our ROC curve covers 0.97 times area, which is a very good score since the closest the score is to 1, better is our predective model

From the curve above, 0.3 is the optimum point to take as cutoff probability.

```
metrics.accuracy_score(y_train_final_pred.Converted, y_train_final_pred.final_Predicted)
```

0.9083178726035869

```
c_matrix_2 = metrics.confusion_matrix(y_train_final_pred.Converted, y_train_final_pred.final_Predicted )
c_matrix_2
```

```
array([[3597,  405],
       [ 188, 2278]], dtype=int64)
```

```
TP = c_matrix_2[1,1] # true positive
TN = c_matrix_2[0,0] # true negatives
FP = c_matrix_2[0,1] # false positives
FN = c_matrix_2[1,0] # false negatives
```

```
# Sensitivity

TP / float(TP+FN)
```

0.9237631792376317

```
# Specificity

TN / float(TN+FP)
```

0.8988005997001499

```
# False postive rate

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

0.10119940029985007

```
# Positive predictive value

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

0.8490495713753261

```
# Negative predictive value

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

0.950330250990753

Final Model Metrics and Parameters after selecting optimal point

```
c_matrix = metrics.confusion_matrix(y_train_final_pred.Converted,y_train_final_pred.final_Predicted )
c_matrix
```

```
array([[3597,  405],
       [ 188, 2278]], dtype=int64)
```

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

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

```
# Precision

precision_score(y_train_final_pred.Converted , y_train_final_pred.final_Predicted)
```

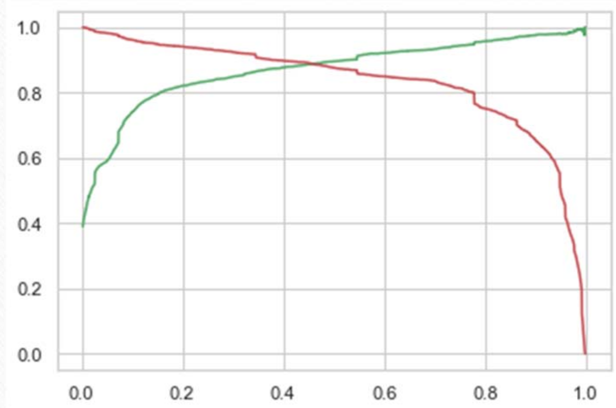
0.8490495713753261

```
# Recall

recall_score(y_train_final_pred.Converted, y_train_final_pred.final_Predicted)
```

0.9237631792376317

Precision Recall Trade-off Curve →



Final Model Evaluation

Final Featuers and thier coeffients :-

const	:	1.748483
Total Time Spent on Website	:	1.034173
Lead Origin_Lead Add Form	:	2.081347
Lead Source_Olark Chat	:	0.424474
Do Not Email_1	:	-1.870891
Last Activity_SMS Sent	:	1.898454
What is your current occupation_undefined	:	-3.797782
Tags_circumventing	:	-5.556923
Tags_financial_issues	:	-3.063557
Tags_in_dilemma	:	-5.148652
Tags_unlikely	:	-5.892963
Last Notable Activity_Modified	:	-1.071245
Last Notable Activity_other	:	1.138596

For Training Dataset :-

Accuracy : 90.83 %

Sensitivity : 92.38%

Specificity : 89.88%

This is a very good score, our model performs really good on train dataset

For Test Dataset :-

Accuracy : 91.09%

Sensitivity : 93.52%

Specificity : 89.51%

This is a very good score, our model performs really good on test dataset,

Also, the difference between train and test data set is very small, which indicates that our model is a very good fit

So looking over the above parameters it can be said that -:

`Lead Origin_Lead Add Form` contributes highest in helping securing a lead, followed by `Last Activity_SMS Sent`,

while `Tags_unlikely` contributes highest in turning away a lead so we should definitely avoid those.

CONCLUSION

AVOID

- Tags_circumventing (Negative):

The more the lead is avoiding the contact from team higher are their chance of failing to convert, or becoming cold lead.

- Tags_in_dilemma (Negative):

The more doubt the lead is showing, the higher are their chances of not getting converted and thus becoming a cold lead.

- Tags_unlikely (Negative):

If the lead is already engaged is some other course or if they are not eligible for the enrolling (being a diploma holder, already a student etc.) they will not get converted and therefore they will be termed as cold heat

Go After

- Lead Origin_Lead Add Form:

If the lead has originated via Add form, then they are most likely to convert.

- Last Activity_SMS Sent:

If the last activity is done via SMS, then there is a very high probability of successful lead conversion.

- Last Notable Activity_other:

If the lead having last notable activity as other their chances of getting converted are high

Also, those leads should be target who have shown interest in the past and were classified as `are_certain` during EDA. And people what are visiting website for longer amount of time.