LEAD SCORING CASE STUDY

By-

- Sumit Singh
- Ranajit Mahapatra
- Santosh Gouda

Agenda

- Problem Statement
- Problem Approach
- Data Cleaning & EDA
- Model Creation and Evaluation
- Model Performance on test Set
- Conclusion & Recommendations

Problem Statement

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%.

The company require to build a model wherein we need to assign a lead score to each of the leads such that the customers with a higher lead score have a higher conversion chance and the customers with a 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%.

PROBLEM APPROACH

We can solve the problem by building a regression model with acceptable cutoff probabilty score which will maximise accuaracy,sensitivity & specificity. The steps envole-

- Treating missing values
- EDAs
- Creating Model
- Evaluating Model
- Calculating the performance

Data Cleaning & EDA

Removing missing values where %share is more than 30%

	attributes	missing_percentage
3	Lead Source	0.4
7	TotalVisits	1.5
9	Page Views Per Visit	1.5
10	Last Activity	1.1
11	Country	26.6
12	Specialization	36.6
13	How did you hear about X Education	78.5
14	What is your current occupation	29.1
15	What matters most to you in choosing a course	29.3
24	Tags	36.3
25	Lead Quality	51.6
28	Lead Profile	74.2
29	City	39.7
30	Asymmetrique Activity Index	45.6
31	Asymmetrique Profile Index	45.6
32	Asymmetrique Activity Score	45.6
33	Asymmetrique Profile Score	45.6

- There are various attributes where the field is ‘select’ , Replacing them with null will make the analysis Easier.
- There are various fields where 30% of nulls are present. Dropping all these values will be the option as replacing them with mean/median will create bias in the data.

Removing attributes with unique value

	attributes	unique_values
15	Magazine	1
21	Receive More Updates About Our Courses	1
22	Update me on Supply Chain Content	1
23	Get updates on DM Content	1
24	I agree to pay the amount through cheque	1

There are 5 attributes where values are unique (only one value). These attributes will not contribute to impact outcomes, so dropping these values will be better options.

Missing values imputation having missing less than 30%

	attributes	missing_percentage
3	Lead Source	0.4
7	TotalVisits	1.5
9	Page Views Per Visit	1.5
10	Last Activity	1.1
11	Country	26.6
12	What is your current occupation	29.1
13	What matters most to you in choosing a course	29.3

For values where missing value is less than 30%, imputing them with median (for continuous variables) and mode (for categorical variables).

Dropping attributes where only one value contains more than 98%

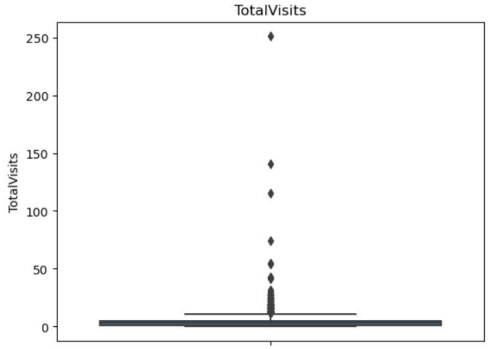
```
### Removing all unrelevant as values are heavily biased

df_leads.drop(['Prospect ID','Lead Number','Do Not Call','Country',
               'What matters most to you in choosing a course','Search',
              'Newspaper Article','X Education Forums','Newspaper','Digital Advertisement',
              'Through Recommendations'],axis=1,inplace=True)
```

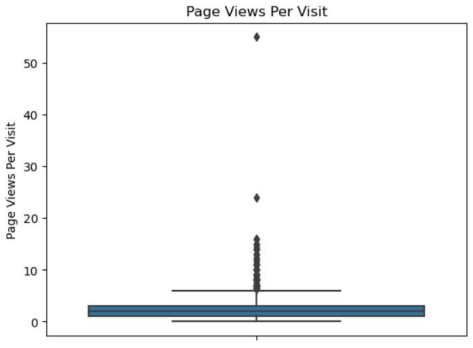
There are various attributes where 98% of values belongs to one segment,so dropping them as well as they are not useful for outcomes.

Outlier Treatment for continuous variable

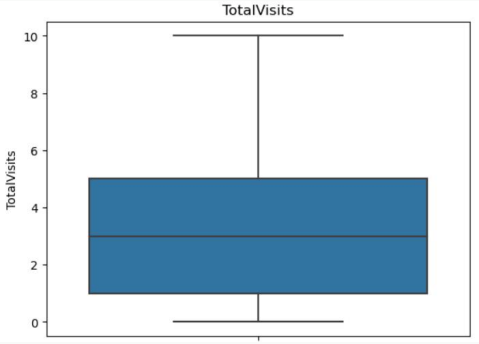
Clipping all outliers to 95%ile to reduce the impact of outlier data

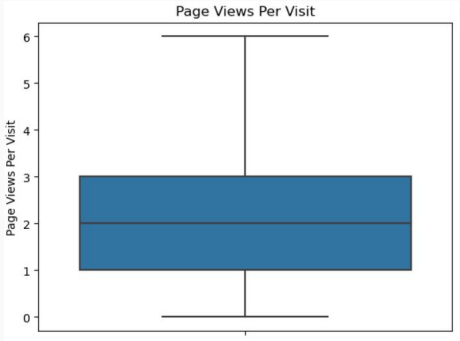
Before



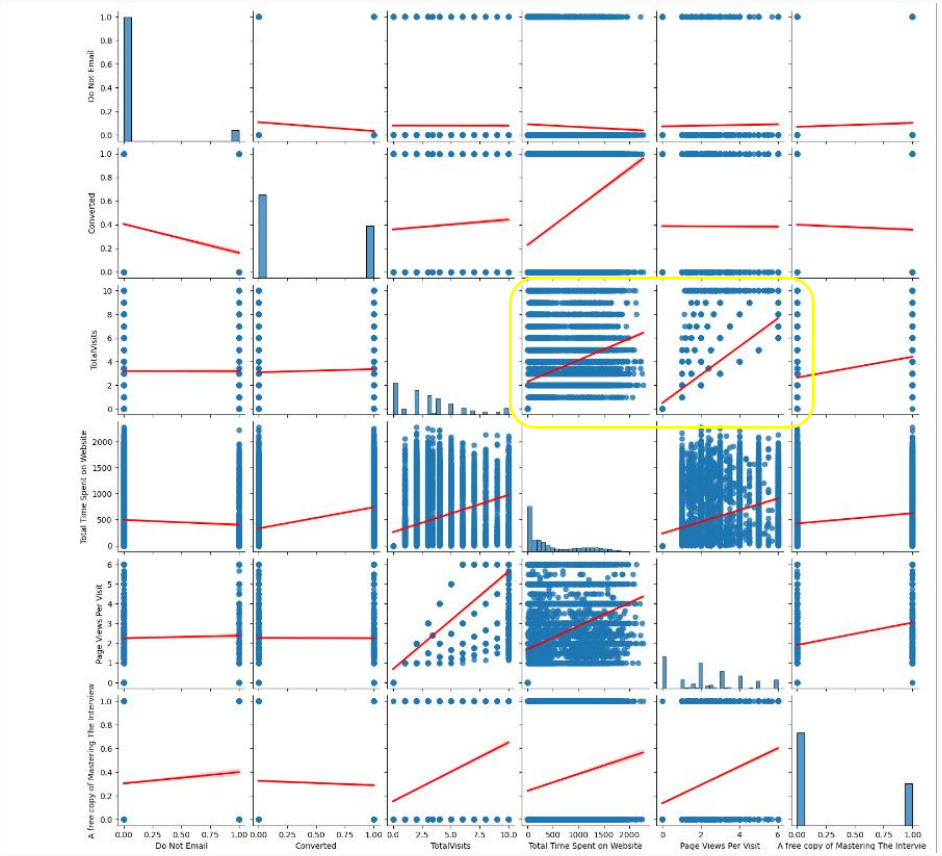


After

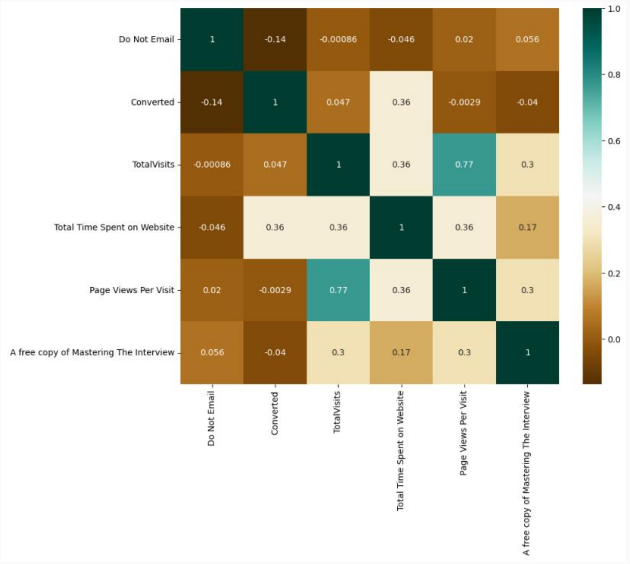




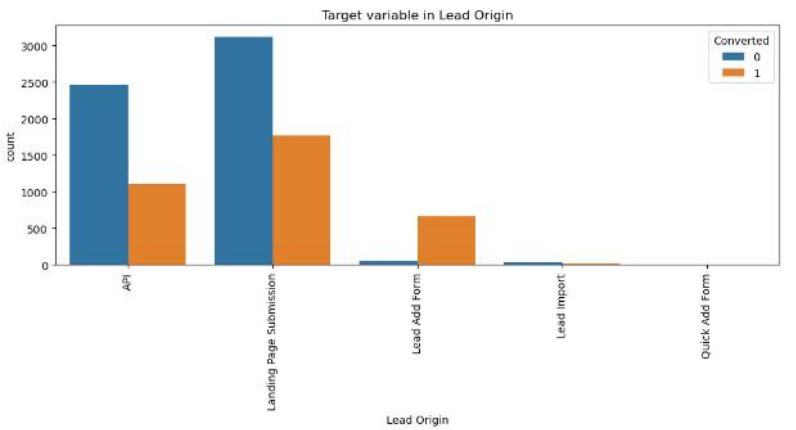
EDA on Continuous attributes

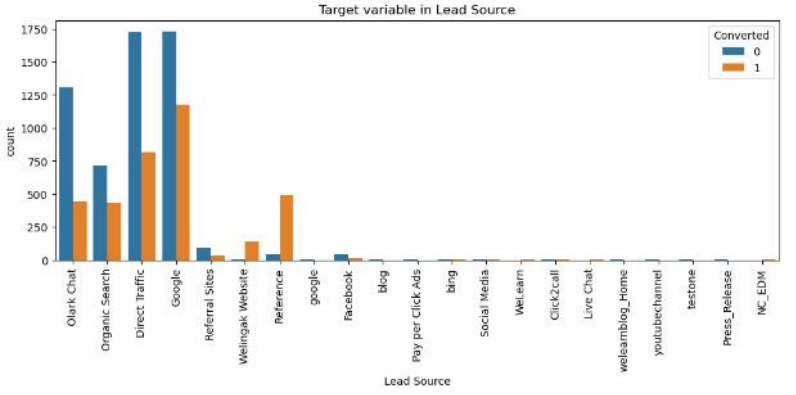


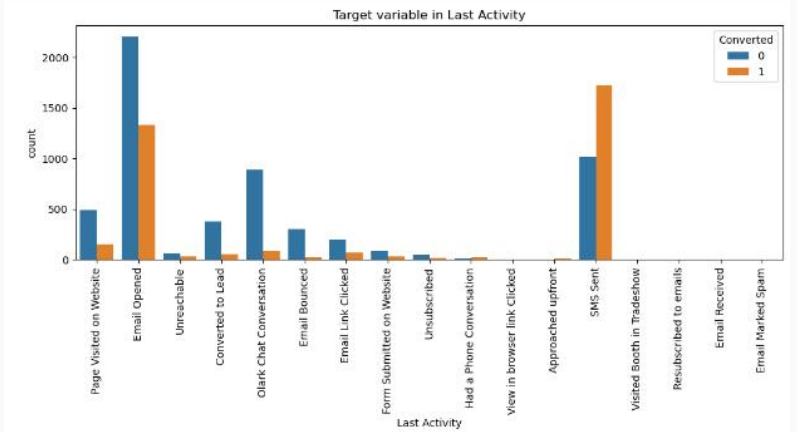
The TotalVisits are hightly corelated with Total time spent on website & Page views per visits. We could drop these columns here but we can drop them using RFE and manual observation basis p-value and VIF.

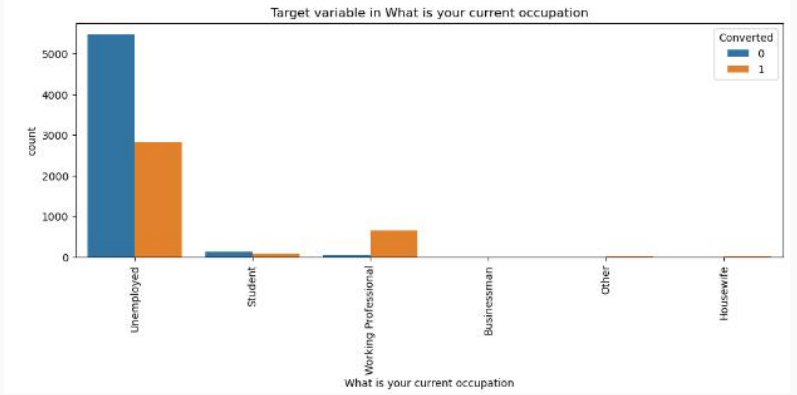


EDA on Categorical attributes









From these countplots we can figure out Lead Add form as form origin,reference and lead source,working professionals have higher conversion rate.

Model Creation and Evaluation

Feature Elimination via RFE,p-value and VIF

After RFE & 5 repetation of model fitting, eliminating all non statistically significant attributes we have reduced attributes to 11 which have acceptable p-values and VIF.

	coef	std err	z	P>|z|	[0.025	0.975]
const	-0.3635	0.070	-5.193	0.000	-0.501	-0.226
Do Not Email	-1.7494	0.155	-11.278	0.000	-2.053	-1.445
Total Time Spent on Website	3.8530	0.138	27.929	0.000	3.583	4.123
Lead Origin_Lead Add Form	3.6015	0.181	19.918	0.000	3.247	3.956
Last Activity_Converted to Lead	-1.0614	0.199	-5.329	0.000	-1.452	-0.671
Last Activity_Olark Chat Conversation	-0.8453	0.176	-4.808	0.000	-1.190	-0.501
What is your current occupation_Working Professional	2.6022	0.176	14.753	0.000	2.256	2.948
Last Notable Activity_Email Link Clicked	-1.7349	0.244	-7.096	0.000	-2.214	-1.256
Last Notable Activity_Email Opened	-1.4909	0.083	-17.895	0.000	-1.654	-1.328
Last Notable Activity_Modified	-1.7471	0.094	-18.652	0.000	-1.931	-1.563
Last Notable Activity_Olark Chat Conversation	-1.6576	0.353	-4.693	0.000	-2.350	-0.965
Last Notable Activity_Page Visited on Website	-1.6762	0.179	-9.346	0.000	-2.028	-1.325

	VIF Factor	features
8	2.0	Last Notable Activity_Modified
4	1.8	Last Activity_Olark Chat Conversation
1	1.5	Total Time Spent on Website
9	1.3	Last Notable Activity_Olark Chat Conversation
3	1.2	Last Activity_Converted to Lead
7	1.2	Last Notable Activity_Email Opened
5	1.2	What is your current occupation_Working Profes...
2	1.1	Lead Origin_Lead Add Form
0	1.1	Do Not Email
10	1.0	Last Notable Activity_Page Visited on Website
6	1.0	Last Notable Activity_Email Link Clicked

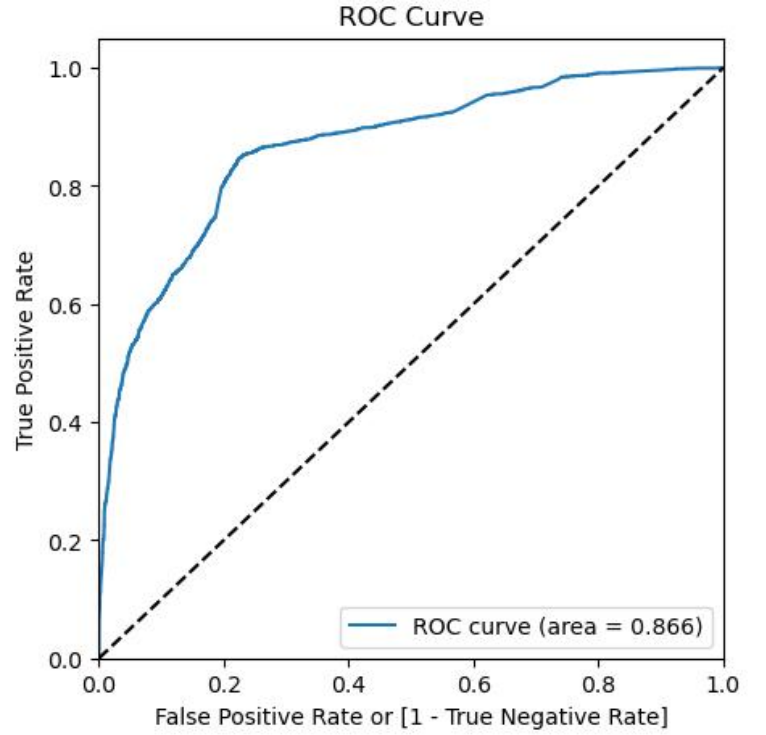
Optimum cutoff selection based on predicted probability

From Above output we can take 40% as cutoff point as in our problem statement Recall is most important parameter

Probability_bins	Cumtarget_0	Cumtarget_1	CumPrecision	Recall
0.7>=	187	1312	0.875250	0.494162
0.6>=	309	1520	0.831055	0.572505
0.5>=	559	1750	0.757904	0.659134
0.4>=	852	2129	0.714190	0.801883
0.3>=	964	2249	0.699969	0.847081
0.25>=	1086	2285	0.677840	0.860640
0.2>=	1411	2331	0.622929	0.877966
0.15>=	2134	2423	0.531709	0.912618
0.1>=	3177	2613	0.451295	0.984181
0.08>=	3238	2617	0.446968	0.985687
0.06>=	3378	2625	0.437281	0.988701
0.04>=	4036	2651	0.396441	0.998493
0.02>=	4252	2655	0.384393	1.000000
-0.001>=	4275	2655	0.383117	1.000000

Performance of model in training set

The final model has 80% of accuracy along with 80% each of specificity and sensitivity, with 86.6% of AUC.




```
# Create confusion matrix

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

```
[[3423  852]
 [ 526 2129]]
```

```
## Overall Accuracy
accuracy=metrics.accuracy_score(y_train_pred_final.Converted, y_train_pred_final.Predicted)
print(accuracy)
```

```
0.8011544011544012
```

```
### Evaluating Specificity & sensitivity

# Let's evaluate the other metrics as well

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

```
sensitivity=TP/(TP+FN)
specificity=TN/(TN+FP)

print("Sensitivity: ",sensitivity)
print("Specificity: ",specificity)
```

```
Sensitivity:  0.8018832391713747
Specificity:  0.8007017543859649
```

Model Performance on test Set

Performance of model in Test Set

The final model has 81% of accuracy along with 83% of sensitivity and ~80% of specificity on the test set. Hence the model is very stabel

```
### Accuracy

accuracy_test=metrics.accuracy_score(y_test_pred_final['Converted'], y_test_pred_final.predicted)
print(accuracy_test)
```

```
0.8108225108225108
```

```
## Confusion Metrics

confusion_test = metrics.confusion_matrix(y_test_pred_final['Converted'], y_test_pred_final.predicted
confusion_test
```

```
array([[1119,  285],
       [ 152,  754]], dtype=int64)
```

```
### Evaluating Specificity & sensitivity

# Let's evaluate the other metrics as well

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
```

```
sensitivity=TP/(TP+FN)
specificity=TN/(TN+FP)

print("Sensitivity: ",sensitivity)
print("Specificity: ",specificity)
```

```
Sensitivity:  0.8322295805739515
Specificity:  0.7970085470085471
```

Conclusion & Recommendations

- As per Model, go aggressive on leads with more than 40% by contacting them via various channels.
- Create marketing programs to target these users via various channels such as google, Facebook, Instagram
- Offer these leads additional discounts to get more conversions.
- Do Not Focus on Leads where probability is less than 40%.

THANK YOU