Lead Scoring Case Study

By

Simran Singh
and
Rajna Ameen

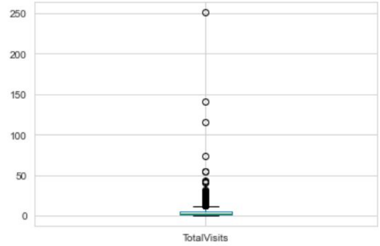
PROBLEM STATEMENT

- An education company ‘X Education’ sells online courses. Each day, many professionals land on their website. When these people fill up a form, they are classified to be a lead.
- Once these leads are acquired, sales team starts to contact them. Some of the leads get converted while most do not (lead conversion rate is very poor).
- To make this process more efficient, the company wishes to identify the most potential leads, also known as ‘Hot Leads’. If they successfully identify hot leads, the lead conversion rate will go up.
- The company required us to build a model where we needed to assign a lead score to 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 (most likely to convert) whereas a lower score would mean that the lead is cold.
- The CEO, in particular, gave a ballpark of the target lead conversion rate to be around 80%.

Step 1: inspecting and cleaning the data

- Checked the shape, data types and statistical summary of the dataframe
- Ensured that the datatypes of each column are assigned accurately
- For the Missing Value Treatment, as per industry standards, we decided to
    - Drop columns with more than 40% null values
    - Drop rows of the columns with <2% missing values
    - Imputed for the columns with missing values between 2% and 40%
    - For imputation we went with median for numerical columns as (columns were skewed) and mode for Categorical columns

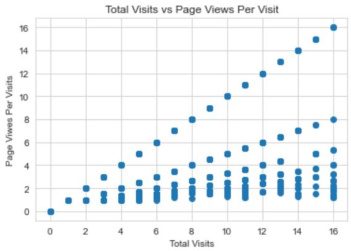
Sanity checks on Categorical Columns

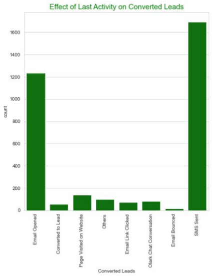


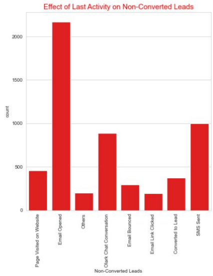
- Dropped ‘Prospect ID’ as it not needed because we already have 'Lead Number' as an identification column
- Dropped highly skewed columns (where >80% of the rows having same value) since columns with biased data will not contribute to our model
- Replaced select with np.NaN everywhere in the dataframe
- Manipulated some columns to bucket < 2% values per column, so that unnecessary creation of dummy variables can be avoided
- Outlier Detection and Treatment
  - Capped the outliers at 99th percentile for the column ‘Total Visits’
  - Page Views Per Visit had continuous values above the upper fence but because they did not resemble outliers we decided to allow them
- Checked the % of retained rows
- Got rid of sales team generated data by dropping the concerned columns

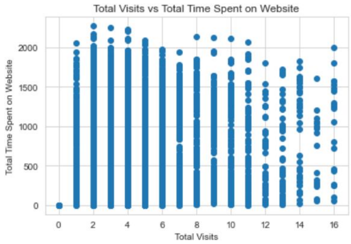
Exploratory Data Analysis

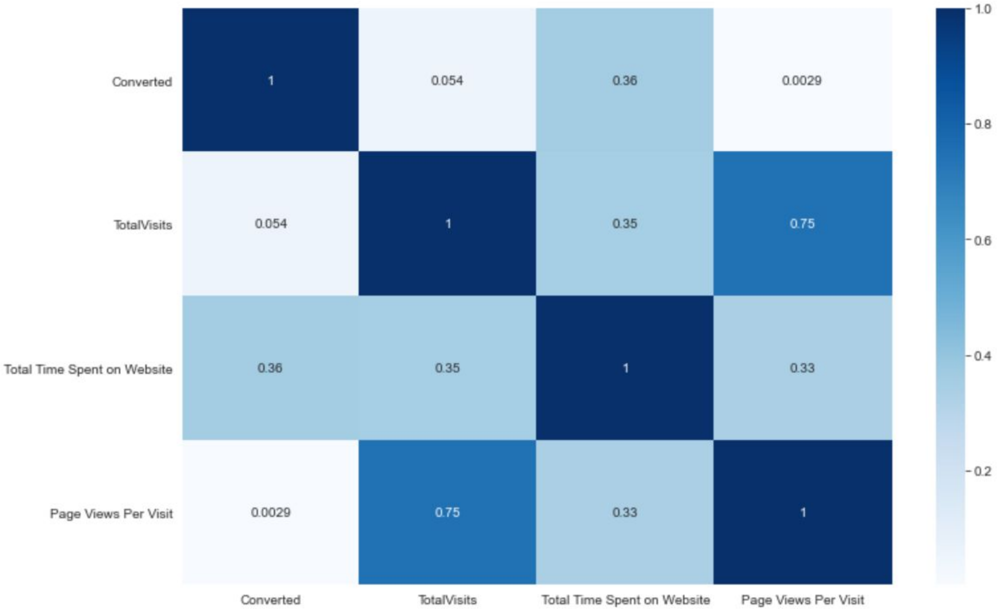
- Imbalance percentage was found to be 62:38
- Bivariable and Multivariate analysis was then done on the data using a variety of plots


Total Visits vs Page Views Per Visit


Effect of Last Activity on Converted Leads


Effect of Last Activity on Non-Converted Leads


Total Visits vs Total Time Spent on Website



	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit
Converted	1	0.054	0.36	0.0029
TotalVisits	0.054	1	0.35	0.75
Total Time Spent on Website	0.36	0.35	1	0.33
Page Views Per Visit	0.0029	0.75	0.33	1

Step 2: Handling categorical variables

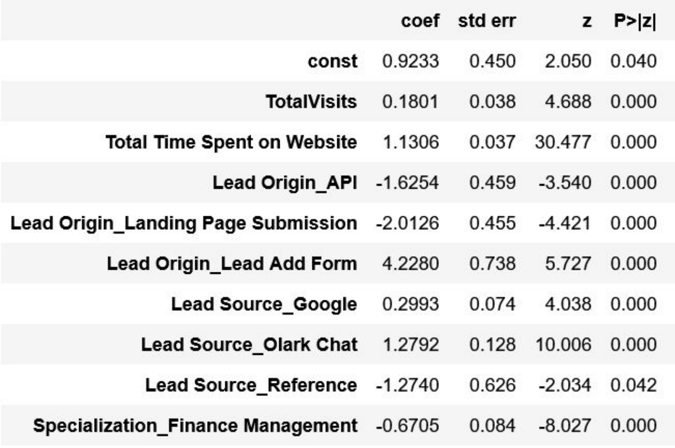
- Converted ‘A free copy of Mastering The Interview’, a binary variable having Yes/No to 1/0
- Creating dummy features (one-hot encoding) for categorical variables with multiple levels

Step 3 & 4: Train-Test Split & Feature Scaling

- Split the data into training set and testing set
- Made use of Standard Scaler and applied fit transform on the columns 'TotalVisits' and 'Total Time Spent on Website'
- Checked the old conversion rate (which came to out be 37.85 %) before moving to model building

Step 5: Model Building

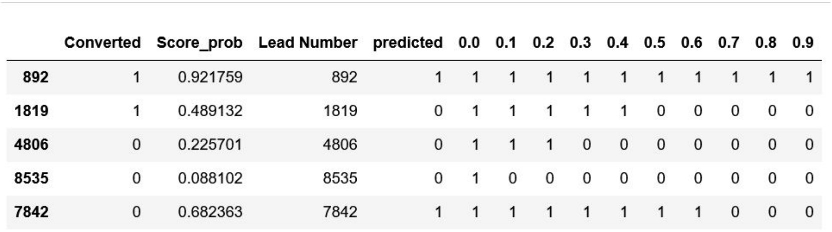
- Made use of Generalized Linear Model under Statsmodels library to build a binary classification logistic model as our first model
- Performed Feature Selection using RFE to select 20 variables
- Next, Performed Manual Selection for selecting variables. Did multiple iterations of model building, dropping irrelevant columns with each iteration, using the following approach:
    - Dropping variables with high p-values (>0.05) and high VIF (>5) both
    - Next, dropping the insignificant variables with p-values > 0.05
    - Lastly, dropping variables with VIF >5
- Arrived at our final model which has 9 features
- Calculated metrics beyond accuracy of the final model, before before finding the optimal cut-off



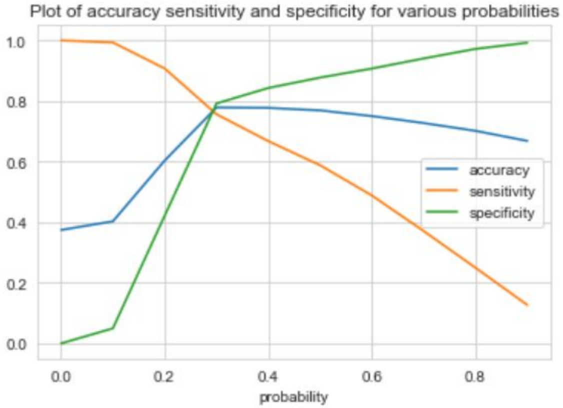
	coef	std err	z	P>|z|
const	0.9233	0.450	2.050	0.040
TotalVisits	0.1801	0.038	4.688	0.000
Total Time Spent on Website	1.1306	0.037	30.477	0.000
Lead Origin_API	-1.6254	0.459	-3.540	0.000
Lead Origin_Landing Page Submission	-2.0126	0.455	-4.421	0.000
Lead Origin_Lead Add Form	4.2280	0.738	5.727	0.000
Lead Source_Google	0.2993	0.074	4.038	0.000
Lead Source_Olark Chat	1.2792	0.128	10.006	0.000
Lead Source_Reference	-1.2740	0.626	-2.034	0.042
Specialization_Finance Management	-0.6705	0.084	-8.027	0.000

Step 6: Finding Optimal Cutoff Point

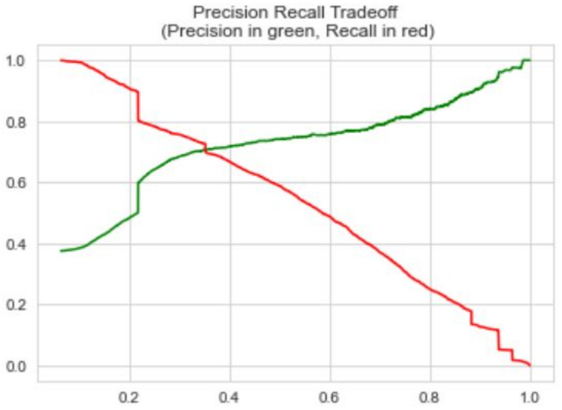
- Calculated accuracy sensitivity and specificity for various probability cutoffs and stored in a dataframe.
- Calculated the optimal cutoff using Sensitivity-Specificity-Accuracy curve and Precision-Recall curve.
- For this case study we had to focus on maximising Sensitivity (same as Recall) while also not compromising on Accuracy & Specificity & Precision.
- After careful trial and error we found our best results at 0.24



	Converted	Score_prob	Lead Number	predicted	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
892	1	0.921759	892	1	1	1	1	1	1	1	1	1	1	1
1819	1	0.489132	1819	0	1	1	1	1	1	0	0	0	0	0
4806	0	0.225701	4806	0	1	1	1	0	0	0	0	0	0	0
8535	0	0.088102	8535	0	1	0	0	0	0	0	0	0	0	0
7842	0	0.682363	7842	1	1	1	1	1	1	1	1	0	0	0



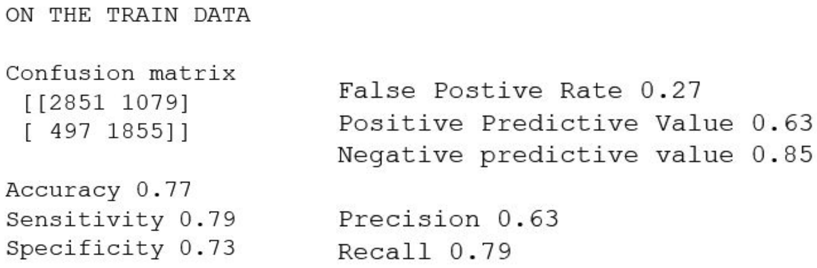
Plot of accuracy sensitivity and specificity for various probabilities



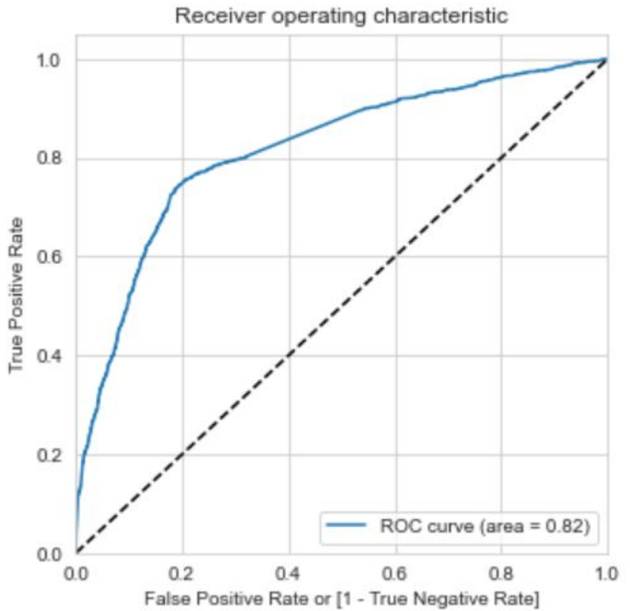
Precision Recall Tradeoff
(Precision in green, Recall in red)

Step 7: Making predictions and Calculating Evaluation Metrics on the train data

- Applied the cutoff of 0.24 and made predictions on the Train set.
- Finally, calculated the evaluation metrics on the Train set.
- We were happy with the sensitivity of the final model, 79% as even the problem statement had given a ‘ballpark of around 80%’.
- Next we plotted the ROC curve and got the AUC at 0.82

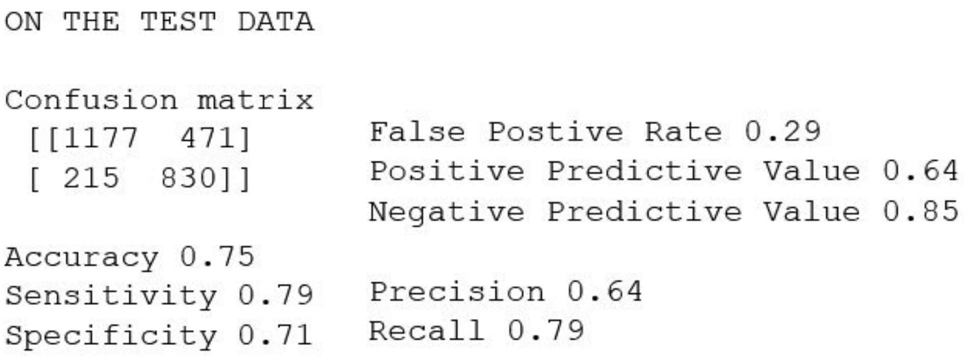


```
ON THE TRAIN DATA

Confusion matrix
 [[2851 1079]              False Postive Rate 0.27
 [ 497 1855]]              Positive Predictive Value 0.63
                           Negative predictive value 0.85
Accuracy 0.77
Sensitivity 0.79           Precision 0.63
Specificity 0.73           Recall 0.79
```

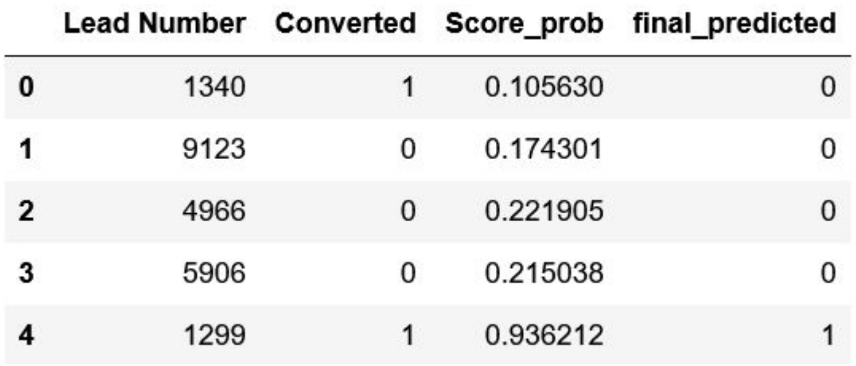


Step 8: Making predictions on the test set and Evaluating The Model

- Scaled the Test Set
- Calculated the probabilities and made predictions on the Test Set
- Applied the cutoff of 0.24 on the Test set as well
- Finally we checked the evaluation metrics on the test set



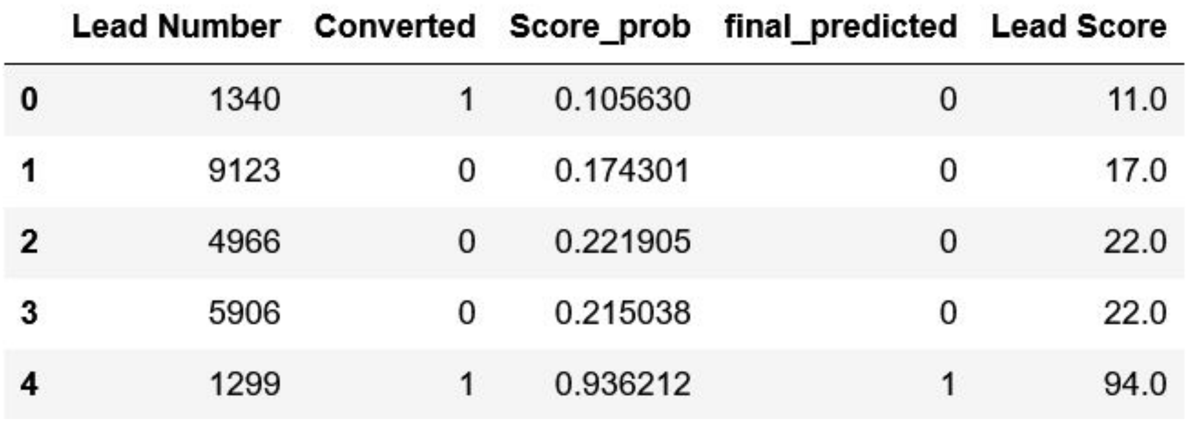
```
ON THE TEST DATA

Confusion matrix
 [[1177  471]      False Postive Rate 0.29
 [ 215  830]]      Positive Predictive Value 0.64
                   Negative Predictive Value 0.85

Accuracy 0.75
Sensitivity 0.79   Precision 0.64
Specificity 0.71   Recall 0.79
```



	Lead Number	Converted	Score_prob	final_predicted
0	1340	1	0.105630	0
1	9123	0	0.174301	0
2	4966	0	0.221905	0
3	5906	0	0.215038	0
4	1299	1	0.936212	1

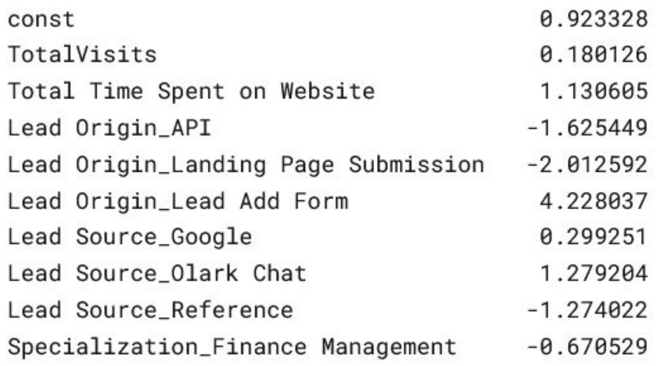
Step 9 & 10: Calculating Lead Score and Lead Conversion Rate

- We calculated the Lead Score by multiplying the conversion probability (Score_prob column) with 100
- We also calculated our Lead Conversion Rate and got it as 79.43 %



	Lead Number	Converted	Score_prob	final_predicted	Lead Score
0	1340	1	0.105630	0	11.0
1	9123	0	0.174301	0	17.0
2	4966	0	0.221905	0	22.0
3	5906	0	0.215038	0	22.0
4	1299	1	0.936212	1	94.0

EXPLANATION OF RESULTS IN BUSINESS TERMS



```
const                                  0.923328
TotalVisits                            0.180126
Total Time Spent on Website            1.130605
Lead Origin_API                       -1.625449
Lead Origin_Landing Page Submission   -2.012592
Lead Origin_Lead Add Form              4.228037
Lead Source_Google                     0.299251
Lead Source_Olark Chat                 1.279204
Lead Source_Reference                 -1.274022
Specialization_Finance Management     -0.670529
```

- Lead Score was assigned to every lead as the ‘conversion probability * 100’
- Leads with calculated Lead Score greater than 24 (or conversion probability > 0.24) are to be considered 'Hot Leads'
- The lead conversion rate after applying the final model was calculated as 79.43%
- The attached image describes the most important features.
- The three most important features obtained from the final model can be described as:
  - Lead Origin (Lead Add Form) - positive effect, so chances of being a hot lead increase the most if lead origin is ‘Lead Add Form’
  - Lead Origin (Landing Page Submission) - negative effect, so chances of being a hot lead decrease the most if lead origin is ‘Lead Page Submission’
  - Lead Origin (API) - negative effect, so chances of being a hot lead reduce if lead origin is ‘API’

Recommendations

- X Education should focus on features that increase the chances of a lead being a ‘Hot Lead’ and be aware of those features that reduce the chances of a lead being a hot lead
- Features that positively affect the probability of a lead becoming a hot lead are (in decreasing order of effect):
  - Lead Origin_Lead Add Form
  - Lead Source_Olark Chat
  - Total Time Spent on Website
  - Lead Source_Google
  - Total Visits
- Features that negatively affect the probability of a lead becoming a hot lead are (in decreasing order of effect):
  - Lead Origin_Landing Page Submission
  - Lead Origin_API
  - Lead Source_Reference
  - Specialization_Finance Management