Lead Scoring Case Study

By

Vigneshwari Chandramohan

Anilkumar Biradar

Business Requirement

An education company sells online courses. When people fill the online forms, they are classified as Leads. 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 is around 30%. To increase the conversion rate, the education institute wishes to identify the most potential leads. Once identified, the sales team could focus on communicating with the potential leads rather than making calls to everyone. Eventually, this would increase the lead conversion rate.

Problem Statement

- As a Data Analyst, we need to build a model to identify the potential leads. Each lead will be assigned a lead score.
- Customers with high lead score will have high conversion chance and the customers with low lead score has low conversion chance.
- Based on the lead score, the sales team could make call to the potential customers (high lead score), which would eventually increase the conversion rate.
- This can be achieved by building a logistic regression model and finding the accuracy, sensitivity/recall and specificity.

Analysis and Approach:

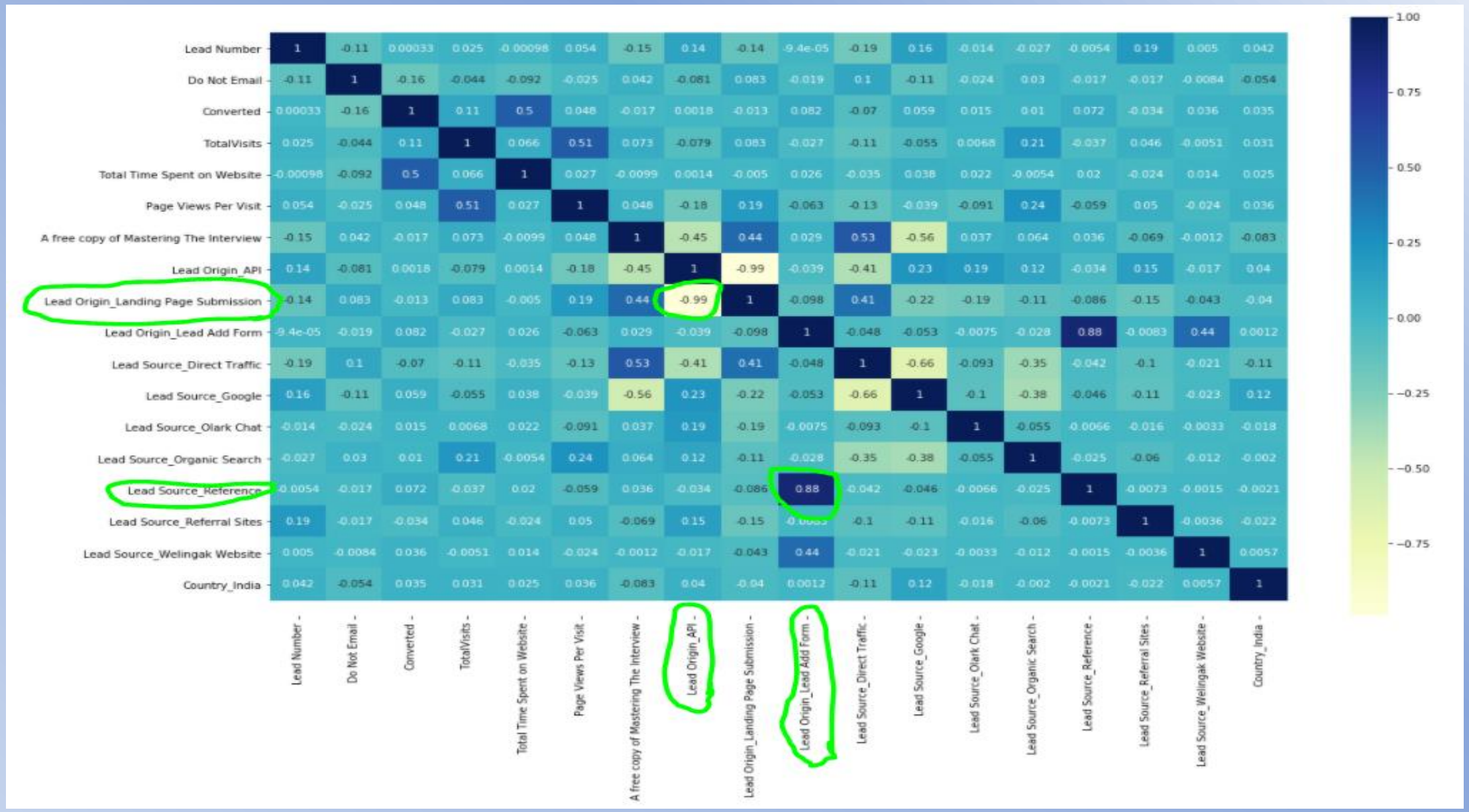
- Read the csv file and perform all data cleaning process.
- Drop the columns generated by sales team.
- Handle the null values by dropping the rows/columns or imputing it with mean, median or mode based on the requirement.
- Univariate and Bivariate analysis has been done to study the numerical and categorical variables.

Analysis and Approach: (contd.)

- Interquartile range is used to treat the outliers.
- Once the data is cleaned, check the percentage of data that has retained. In our case study, 73% of the original data has been retained.
- Convert all the categorical variables that have the value ‘Yes’ and ‘No’ to ‘1’ and ‘0’
- Create dummy variables for all the categorical variables.

Analysis and Approach: (contd.)

- Check the data imbalance in the target variable (Converted)
- Find the correlations and remove the variables that are highly correlated (positive or negative). This can be identified using heatmaps.
- <Refer the heatmap given in the next slide>


Lead Number
Do Not Email
Converted
TotalVisits
Total Time Spent on Website
Page Views Per Visit
A free copy of Mastering The Interview
Lead Origin_API
Lead Origin_Landing Page Submission
Lead Origin_Lead Add Form
Lead Source_Direct Traffic
Lead Source_Google
Lead Source_Olark Chat
Lead Source_Organic Search
Lead Source_Reference
Lead Source_Referral Sites
Lead Source_Welingak Website
Country_India
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75

Model Building (Logistic Regression)

- Use RFE (Recursive Feature Elimination) to choose at least 15 variables.
- Now, using the features selected, build the model.
- Build the logistic regression model using Generalised Linear Models.
- Based on the models statistical summary and VIF we can drop the features based on the below conditions.
  - High p-value, high VIF -> drop those variables
  - High - Low:
    - High p-value, Low VIF - remove these first
    - Low p-value, High VIF - remove these after the ones above
  - Low p-value, Low VIF -> Keep it as such. No need to drop
- The p-values for all the features should be less than 0.05 and VIF should be less than 5 has to be achieved.

Statistical Summary and VIF values

jupyter Latest_Lead_Scoring_Assignment Last Checkpoint: 17 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Run Markdown

```
logm5 = sm.GLM(y_train,X_train_5, family = sm.families.Binomial())
model_5 = logm5.fit()
model_5.summary()
```

Out[76]:

Generalized Linear Model Regression Results

Dep. Variable:	Converted	**No. Observations:**	5400
Model:	GLM	**Df Residuals:**	5391
Model Family:	Binomial	**Df Model:**	8
Link Function:	logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-2744.1
Date:	Mon, 07 Dec 2020	**Deviance:**	5488.1
Time:	22:01:44	**Pearson chi2:**	5.42e+03
No. Iterations:	5		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	1.0471	0.390	2.686	0.007	0.283	1.811
Do Not Email	-1.5896	0.170	-9.355	0.000	-1.923	-1.257
TotalVisits	0.1966	0.034	5.770	0.000	0.130	0.263
Total Time Spent on Website	1.1683	0.036	32.809	0.000	1.098	1.238
Lead Source_Direct Traffic	-1.7225	0.394	-4.374	0.000	-2.494	-0.951
Lead Source_Google	-1.5196	0.393	-3.867	0.000	-2.290	-0.749
Lead Source_Olark Chat	-1.5176	0.476	-3.188	0.001	-2.451	-0.585
Lead Source_Organic Search	-1.5643	0.399	-3.920	0.000	-2.346	-0.782
Lead Source_Referral Sites	-2.1848	0.476	-4.591	0.000	-3.118	-1.252

jupyter Latest_Lead_Scoring_Assignment Last Checkpoint: 18 min

File Edit View Insert Cell Kernel Widgets Help

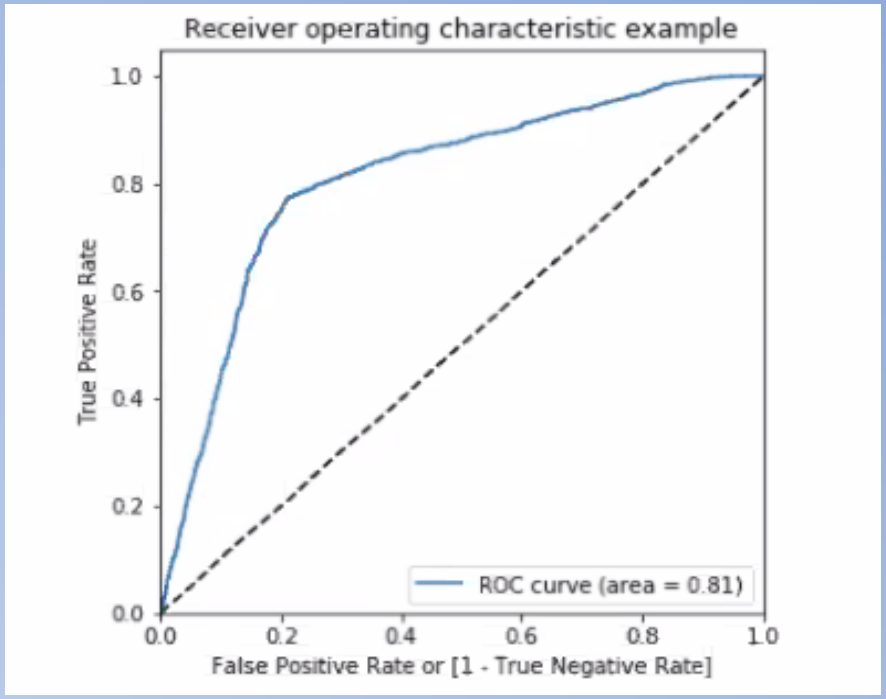
Run Markdown

```
vif['VIF'] = [variance_inflation_factor(X_train[cols
vif['VIF'] = round(vif['VIF'], 2)
vif = vif.sort_values(by = "VIF", ascending = False)
vif
```

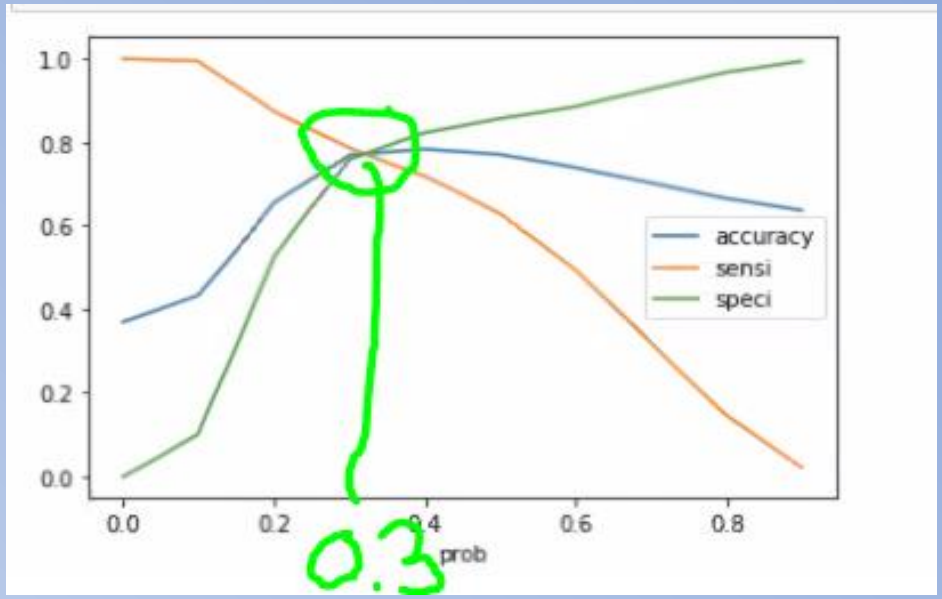
Out[77]:

	Features	VIF
0	Do Not Email	1.12
3	Lead Source_Direct Traffic	1.08
6	Lead Source_Organic Search	1.07
1	TotalVisits	1.06
2	Total Time Spent on Website	1.02
4	Lead Source_Google	1.02
5	Lead Source_Olark Chat	1.00
7	Lead Source_Referral Sites	1.00

- Now using the model predict the y-values. These values will represent the probability of occurrence.
- Define a threshold and identify whether the particular customer has been converted or not. This threshold value can be chosen in an arbitrary way.
- But to find the optimal threshold point we can go ahead with ROC curve.
- ROC curve shows the trade-off between sensitivity and specificity
- ROC curve which follows the left hand border will be more accurate.



- Define threshold value and predict the converted value using the threshold value.
- Now plot the values for accuracy, specificity and sensitivity.
- The point where accuracy, specificity and sensitivity gives you the optimal threshold.
- Based on the plot, we can decide upon the optimal threshold value as 0.3 and calculate the predicted the converted value.



- Now, calculate the accuracy, specificity, sensitivity and precision.
- Good accuracy value – shows that our model is good in predicting the leads
- Sensitivity/Recall – High recall value shows that most of the customers who are converted has been identified as converted.
- Specificity – High specificity value shows that most of the customers who are not converted has been correctly identified as not converted.

Train data:

Accuracy: 0.7696296296296297

sensitivity: 0.7859649122807018

specificity: 0.7600587371512482

Model Evaluation – Test data

- Evaluate the model created using the test data.
- Predict (probability) for y-test
- Use the same threshold which we used to predict the converted value for train dataset and predict the converted value for the test dataset.(Threshold value = 0.3 in our case as per ROC).

Model Evaluation – Test data

- Now calculate the accuracy, sensitivity and specificity.
- All these three values should be more or less same as our train dataset.

Test data:
Accuracy: 0.7579570688378978
sensitivity: 0.7975206611570248
specificity: 0.7358708189158016

Calculation of Lead Score

- `Lead Score = Converted Probability * 100`

- We have taken the optimum threshold as 0.3
- So, the customers who have lead score of 30 as considered to have higher conversion rate.

Business Recommendations

- The sales team of the education company has to focus on the potential customers who have **high lead score**.
- By doing so, all potential leads could have been identified and the lead conversion rate could be increased.