Lead Scoring Case Study

By:

Manoj Kumar Dara

Meenakshy M

Problem Statement

An education company named X Education sells online courses to industry professionals. On any given day, many professionals who are interested in the courses land on their website and browse for courses. 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%. X Education needs help to select the most promising leads, i.e. the leads that are most likely to convert into paying customers. A model is required to be built wherein a lead score is assigned to each of the leads such that the customers with higher lead score have a higher conversion chance and the customers with lower lead score have a lower conversion chance. The CEO has given a ballpark of the target lead conversion rate to be around 80%

APPROACH

1. Data Reading and Understanding:

To observe data present in the Leads.csv file, we observed following things:

- Number of rows and columns
- Data types of each columns
- Checking first few rows how data looks
- Checking how the data is spread
- Checking for duplicates, if any

2. Data Cleaning

To go through the data further for any shortcomings in the dataset.

- Checked for null values percentage and removed those columns which are above 40%.
- Imputed some of the columns which we thought were necessary like Lead Quality which had important data regarding Lead Conversion.
- Converted all the “selects” to unknown.
- Removed 1% rows of null values as they are very less.

3. Data Visualization

- We performed bivariate analysis on categorical columns to see how they vary w.r.t Converted column.
- We performed bivariate analysis on numerical columns with Converted column to see how the leads are related to these columns.
- In this step we use the correlation matrix to identify the columns which are correlated.

4. Feature Scaling

At this stage our data was clean. We know that logistic regression takes the input parameters as numerical values. Hence, we converted all the categorical columns to numerical .

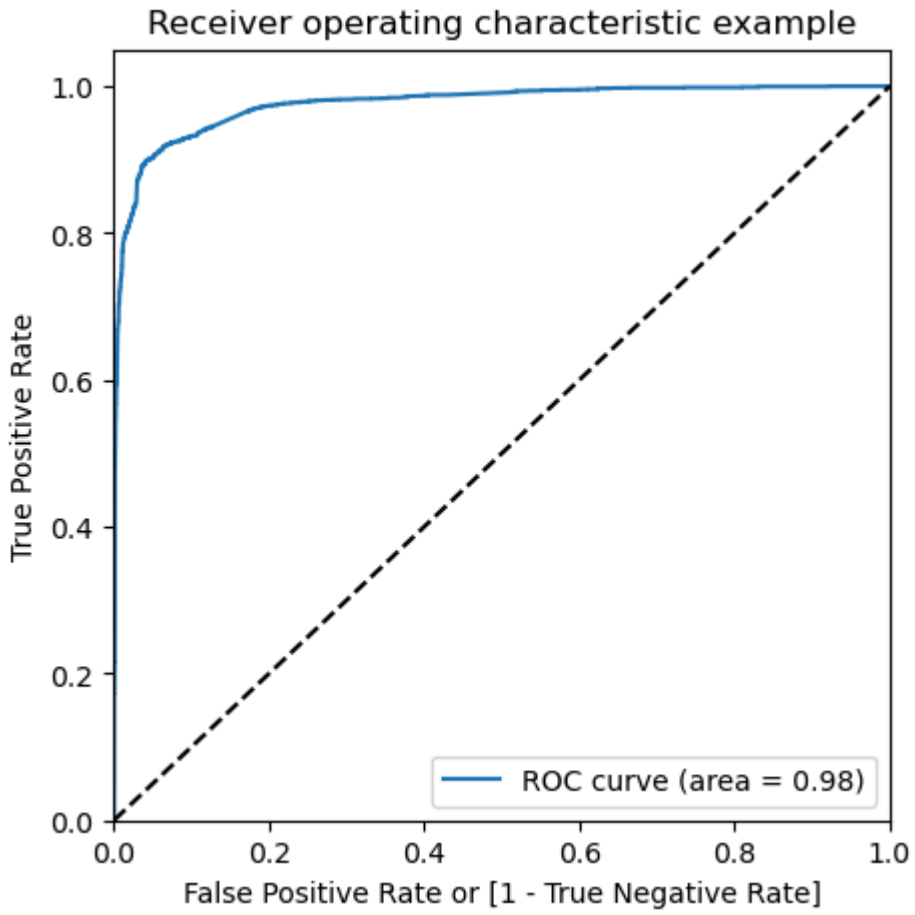
- Columns which have only two levels “Yes” and “No” were converted to numerical using binary mapping.
- Columns which have more than two levels were converted to dummies using the pd.get_dummies function. Now, the data contained only numeric columns and dummy variables.
- Before proceeding for model building, we have rescaled all numerical columns by using the MinMax Scaler method.

5. Model Building

We have used Recursive Feature Elimination Technique to remove attributes and built a model on those attributes that remain. RFE uses the model accuracy to identify which attributes (and combination of attributes) contribute the most to predicting the target attribute. In this step we made the model stable by using the stats library, where we checked the p-values to be less than 0.05 and VIF values to be under 5. Variance inflation factor(VIF) is used to treat multicollinearity.

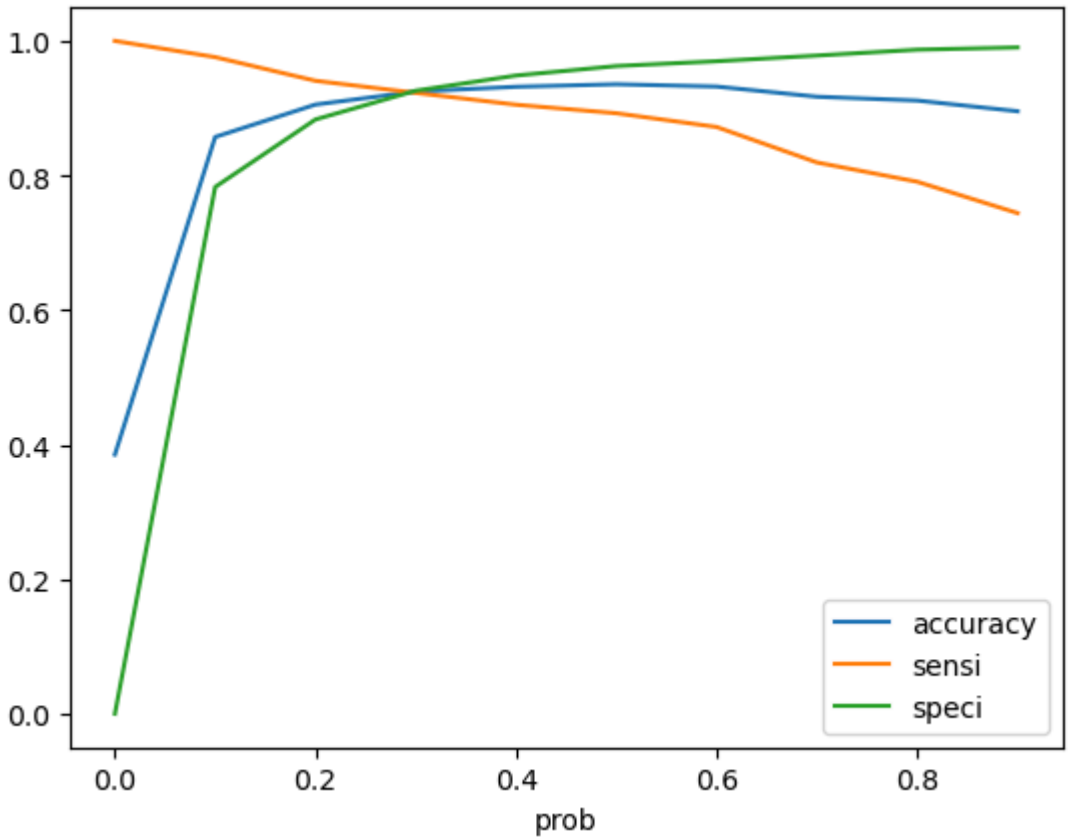
- Once the stable model was created, we predicted probabilities on the train set and created a new column predicted with 1 if probability is greater than 0.5 else 0.

- We calculated the confusion matrix on this predicted column to the actual converted column. We also calculated the metrics sensitivity, specificity, precision, recall, accuracy and plotted ROC curve to find the area under the curve.



6. Model evaluation on Train Set

- In step 5 we took 0.5 as the cut-of. To confirm that it was the best cut, we calculated the probabilities with different cut-offs.
- With probabilities from 0.0 to 0.9, we calculated the 3 metrics -accuracy, sensitivity and specificity
- To make predictions on the train dataset, optimum cutoff of 0.3 was found from the intersection of sensitivity, specificity and accuracy as shown in below figure:


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
prob
accuracy
sensi
speci

7. Predictions on Test Dataset:

After finalizing the optimum cutoff and calculating the metrics on the train set, we predicted the data on the test data set. Below are the observations:

Train Data Set:

- Accuracy: 92%
- Sensitivity: 92%
- Specificity : 93%

Test Data Set:

- Accuracy: 91%
- Sensitivity: 90%
- Specificity: 92%

Conclusion of the Analysis

- The company should make calls which are closed by Horizzon CRM Tools first.
- The Company should make calls to the leads coming from "Tags_Will revert after reading the email".
- The company should make calls to the leads coming from the lead sources "Welingak Websites" and "Reference" as these are more likely to get converted.
- The company should make calls to the leads who spent "more time on the websites" as these are more likely to get converted.
- The company should make calls to the leads coming from the lead sources "Olark Chat" as these are more likely to get converted.
- The company should not make calls to the leads whose lead origin is "Landing Page Submission" as they are not likely to get converted.
- The company should not make calls to the leads. Last Notable Activity is less.
- The company should not make calls to the leads whose lead Quality is worst.