Lead Scoring Case Study – Summary

Problem Statement: An education company named X Education sells online courses to industry professionals. The company markets its courses on several websites and search engines like Google. Once these people land on the website, they fill up forms providing details. X Education has appointed you to help them select the most promising leads, i.e. the leads that are most likely to convert into paying customers. The company requires you to build a model wherein you need to assign a lead score 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, in particular, has given a ballpark of the target lead conversion rate to be around 80%.

The Data has details of 9240 leads that are most likely to convert into paying customers. It includes columns like Lead Source, Total Time Spent on Website, Total Visits, Last Activity, etc. The target variable, in this case, is the column 'Converted' which tells whether a past lead was converted or not wherein 1 means it was converted and 0 means it wasn't converted. Some columns in the data have more than 30 percent of null values, which we will need to clean.

For data cleaning we first dropped Prospect ID, Lead Number, City and country as these columns have no significance in our analysis. We then dropped columns with missing values higher than 45%. There were few columns with 'Select' as one of the categories which can be considered as missing. We dropped 'Lead Profile' and 'How did you head about X Education as the number of rows with 'Select' are large. We also converted null values in Tags to a separate category named 'Unknown'. As there were only few rows with null in TotalVisits and Lead Source, we dropped those rows. In out final dataset we have 6372 rows and 13 columns.

We applied power transformation all the numerical columns since it had values right skewed. Later created dummy variables for all the categorial columns using pd.get_dummies. After dummy variables were created, we have 102 columns and data is ready for pre-processing. We applied train-test split on data by 70/30 and scaled the numerical values using min-max scaler.

Before building the first model, we selected top 15 variables using Recursive feature elimination and then used statsmodel.api for logistic regression model. We got p-values of 0.999 on three variables and rest of the p-values and VIFs were under the threshold values. Dropped these three variables one-by-one keeping check of VIFs in between.

Evaluated our model on the training dataset with random cut-off of 0.5 and we got an accuracy score of 0.91 with sensitivity of 0.98 and specificity of 0.84. Plotted the ROC curve to see trade-off between TPR and FPR also Area under the curve which we got 0.97 meaning model is performing good. We used two methods to look for optimal cut-off sensitivity-specificity trade off and precision-recall trade off. In both the methods we plotted these measures for 10 cut-off value ranging from 0 to 1. We got good sensitivity and accuracy score with a cut-off of 0.53 therefore selected it for the test dataset predictions. We got accuracy score of 0.90 on test set with sensitivity of 0.98 and specificity of 0.82. Finally gave lead scores to all leads by multiplying predicted probabilities from logistic regression model by 100. We recommended strategies of selecting leads based on different business requirements.