Lead Scoring Case study

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.

although X Education gets a lot of leads, its lead conversion rate is very poor. It has conversion rate of only 30% in a day i.e out of 100 only 30 are converted.

Company wants to identify the potential leads and make their conversion rate higher.

The company requires us to build a model wherein you 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%.

Steps followed:

Step 1: Importing libraires and Data and Reading the data.

Initially imported the necessary libraries and inspected data. Over all there are 37 columns with 9240 rows. There are numerical and categorical variables.

Step 2: Data Cleaning

After inspection, data cleaning has to be done. The data contains numerical and categorical variables. There are Null values with high percentage, replaced them with Nan and handled, dropped corresponding variables which are of no use.

Step 3: Data Preparation

Handled some of the categorical variables of binary variables Yes and No are converted to 1 and 0 for ease of process.

Created dummy variables for the remaining categorical variables and dropped the duplicate columns.

Checked for missing values and outliers and imputed them

Step 4: Test-Train Split

Divided the data into test and train data, with 70 % to train model and 30 % to test model.

Step 5: Feature Scaling

We have used standard scaler for some of the variables to standardise data as there are few of continuous values to the train data.

Step 6: EDA

In this step we have used heat map to find the correlations within the data frame and removed the highly corelated dummy variables.

Step 7: Model Building

After all the pre-processing steps are done. We have initiated the model on first training model , we build model based on logistic regression model .

first multivariate logistic regression model using all the features present in the dataset are built. There are lot of data with high P value ,simplying that that variable is statistically insignificant. So we need to eliminate some of the variables in order to build a better model.

Step 8: Feature Selection Using RFE:

We have to firstly eliminate a few features using Recursive Feature Elimination (RFE), and once we have reached a small set of variables to work, we can then use manual feature elimination.

After initiating the RFE model for feature selection, it provides a output of top 15 variables to work with based on ranking and support.

We went ahead with columns selected by RFE model, used them to build a model using statsmodels .

Then we moved to manual feature elimination (using p-values and VIFs) .

We have performed deletion of variables using p – Values and VIFs until we have variables with good values of VIF say threshold of less than 5 and we go ahead using these for making predictions.

We have made a assumption of confusion matrix for predicted value say if it is 0.5 then It is 1 and less than 0.5 is 0.

With this the confusion matrix overall accuracy is at 0.81 i.e 81%.

We also calculated metrics beyond Simply accuracy checked sensitivity :0.69 i.e 69% , specificity:0.88 i.e 88% , false positive rate : 0.11 i.e 11%, Positive predicted value : 0.79 i.e 79%, negative predicted value: 0.82 i.e 82%

Step 9: Plotting the ROC Curve

With the data using RFE model we go ahead and plot a ROC curve to show trade off between sensitivity and specificity.

After plotting the ROC Curve , the graph shows a pretty decent curve with ROC curve area of 0.88 ie 88%, which indicates the model is performing better.

Step 10: Finding Optimal Cutoff Point

We have to find the optimal cutoff point in order to get a decent accuracy, sensitivity, as well as specificity.

Based on the confusion matrix we created the probability metrics for each cutoffs from 0.0 to 0.9 .

Based on probabilities plotted a graph and the optimal cutoff observed is 0.3

From this we observed sensitivity :83% , specificity:78% , accuracy: 80%

Step 11: Precision and recall metrics:

Based on confusing matrix we have checked precision and recall metrics.

precision_score= 71%

recall_score = 82%

cutoff point for metrics based on graph is near to 0.4

Step 11: Making predictions on the test set

After all the steps we made predictions on test data.

After necessary evaluations we found the metrics of accuracy: 0.80 i.e 80%, sensitivity : 0.82 i.e 82.7%%, Specificity : 0.78 i.e 78.2%

The metrics seem to hold on the test dataset as well.

Overall the model looks decent for conversion rate.