Lead Scoring Case Study Summary

Problem Statement:

X Education sells online courses to industry professionals. X Education needs help in selecting the potential leads, i.e. the leads that are most likely to convert into paying customers.

The company needs a model 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, in particular, has given a ballpark of the target lead conversion rate to be around 80%

Solution Summary:

Step1: Reading and Understanding Data.

By using the basic info and describe methods we understood the datatypes and distribution of data. Also understood shape and data imbalance.

Step2: Data Cleaning:

The variables having higher percentage of null values were dropped and for the remaining variables, if numerical, we imputed the missing values with median and if categorical, for lower percentage of null values, we imputed the missing values with mode and for significant percentage of missing values we all together created another category with appropriate name depending on the variable.

Out of 3 numerical variables, 2 had outliers present in them, so in order to treat outliers, we capped the values at 99th percentile.

Step3: Exploratory Data Analysis

EDA is the most important step to understand the data thoroughly and to derive some initial inference based on the visualization. Some of the categorical variables were having significant number of categories which would have made out model too complex as well as less interpretable if the categories were kept as it is. So, we went ahead and merged some of the less populated categories together or with some other suitable category.

Firstly, we did univariate and bivariate analysis for categorical variables, in which we observed category wise distribution of each variable as well as against target variable. The numerical variables were analysed using boxplots for outliers' identification and heatmaps for correlation.

After the EDA, we dropped few variables which had little to no variance.

Step4: Creating Dummy Variables

We created dummy variables and dropped the first column from each dummy variable category to avoid multicollinearity. As we wanted some customization in the column names, we created dummy variables for each category step by step.

Step5: Test Train Split:

We divided the data into training and testing data in proportion of 70% and 30% respectively with random state 44 in order to get same output every time the code is executed.

Step6: Feature Rescaling and building first model

MinMax scale was used for scaling the numerical variables so that the coefficients from our model would be able to correctly explain the numerical variables along with categorical variables.

Using stats model, we built our first model with all the variables and the model had a lot of features with p value greater than 0.05 which had to be eliminated as they were not significant for prediction. So, one way was to eliminate these features one by one, by giving elimination preference to high p value and high VIF, high p value and low VIF, low p value and high VIF and lastly, low p value and low VIF. And the cut-off of elimination feature was, p value greater than 0.05, VIF greater than 5.

Step7: Feature selection using RFE:

But instead of doing all the step-by-step feature elimination, we used RFE to eliminate less significant features and only retain top "n" features as per the requirement of user

Using the Recursive Feature Elimination we went ahead and selected the 20 top important features. Using the statistics generated, we recursively tried looking at the P-values and VIF values in order to select the most significant features that should be present and dropped the insignificant features.

Finally, we arrived at the 18 most significant variables. The VIF's for these variables were also found to be good. We then created a dataframe having converted probability values and with initial assumption that a probability value of more than 0.5 means Converted(1) else Not 0. Based on this assumption, we derived the Confusion Metrics and calculated the overall Accuracy of the model.

We also calculated the 'Sensitivity', 'Specificity' as well as 'Precision', 'recall' matrices to understand how reliable the model is.

Step8: Plotting the ROC Curve

We then tried plotting the ROC curve for the features and the curve came out be pretty decent with an area coverage of 89% which further solidified the of the model.

Step9: Finding the Optimal Cutoff Point

To find out the optimal probability cut-off point where **'**Accuracy', 'Sensitivity' and 'Specificity' would almost be equivalent, we plotted the variation of 'Accuracy', 'Sensitivity', and 'Specificity' curve against the variation in probability from 0.1 to 0.9 with step of 0.1.

We got the optimal probability cut-off point as 0.37 and based on this we recalculated our evaluation metrices as below,

Accuracy – 81.33%

Sensitivity – 80.58%

Specificity – 81.80%

As we got our results in 80s, the main objective of model building has been accomplished.

Step10: Computing the Precision and Recall metrics

As we wanted to correctly predict the actual converted leads, True positive rate, and some of the industries follow sensitivity-specificity view and some follow precision-recall view, so we calculated precision and recall too.

Precision - 79.63%

Recall - 71.5%

Based on the Precision and Recall tradeoff, we got a cut off value of approximately 0.42 which was giving almost similar precision and recall values.

Step11: Making Predictions on Test Set

Finally, we predicted the conversion probability using our trained model on the test data set. Then we implemented the learnings to the test model and calculated the conversion probability found out the evaluation metrices as below,

Accuracy - 81.32%,

Sensitivity - 79.90%

Specificity - 82.17%.

Precision - 72.87%

Recall - 79.90

By comparing the test results with train results, we can say that our model is a good fit.