LEAD SCORING – SUMMARY

Problem statement

X Education sells courses online. Leads and metadata are generated from various sources. Teams are assigned to capture potential leads and convert them into hot leads. We have to create a machine learning model for the company which predicts and assigns a lead score to each lead based on different variables available from historical data. This is a logistic regression problem because we are predicting the classification of the leads as converted or not with a probability of conversion.

Solution

We need to find the hot leads - leads that have a higher probability of getting converted. The teams will spend more time on those and not waste time and resources on leads with low score.

The Dataset

The original dataset was loaded using pandas. It had 9240 rows and 37 columns. However, when we look carefully at the dataset, it becomes immediately clear that a couple of columns are irrelevant to our study. A few columns have a singular value only. This means that we have to look deeper in the data.

Data Cleaning and Data Preparation

We explored the data statistically and visually to get an idea of null values, outliers, how the data is distributed, which columns are redundant etc.

Some of the columns had string values “Select”. This means that users did not a particular value in the form provided to them by the company. It was treated as a null value and replaced by “np.nan”.

There were columns with a high number of missing values. Columns where missing values were greater than 45% were dropped.

We then identified columns which were not adding much value to the understanding of the data and target variable - Converted. After checking value counts for several columns along with a quick EDA, we found that a lot of elements in the categorical variables could be taken away from the dataset. Hence, we dropped those columns.

For the numeric values, we constructed boxplots and checked the correlations between the variables through a heatmap.

Dummy Variables

We converted two object type columns to binary variables since they had two values that could be mapped as 1 for Yes and 0 for No. We then proceeded to create dummy variables for rest of the categorical columns.

Train-Test split

The split was done at 70% and 30% for train and test data respectively. For numeric values, we used Standard Scaler.

Model Building

We used RFE to obtain the top 15 relevant variables for building our logistic regression. The model was built and variables with VIF < 5 and p-value < 0.05 were kept. Since everything seemed as expected, no further models were built.

Model Evaluation

We took an initial assumption that probability of more than 0.5 means 1, else 0. We then created a dataframe with converted probability values. Based on this assumption, a confusion matrix was made, and we calculated accuracy, sensitivity, and specificity for the train set.

After this, the optimum cut off value was found using the ROC curve. The optimum cut off was fixed at 0.3.

Precision and Recall

The Precision and Recall values were also calculated and a Precision – Recall trade-off graph was plotted.

Prediction on Test set

Prediction was done on the test data frame and the confusion matrix was re-created. Model evaluation metrics like accuracy, sensitivity, and specificity were calculated.

Finding the hot leads

Given the ballpark of the target lead conversion rate to be 80%, we found out the important features that the teams must pursue in our final model.