SUMMARY

LEAD CASE STUDY

Problem Statement and Requirement:

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 business wants you to create a model in which you give each lead a lead score so that leads with higher lead scores have a higher chance of converting, while leads with lower lead scores have a lower chance of converting.

Importing and Inspecting Dataset:

Import all the required libraries as a first step. Leads Dataset has been imported to a data frame. Inspect the columns and the shape. To understand data types and column structure, checked the description and column information.

Data Cleaning:

Examine the data set for null values. The data shows that there are numerous column values as Select. We must replace each of those with NaN because it is clear that no value has been chosen. The percentage of missing values for each column are then determined. Columns with a high percentage of missing values (>45) have been dropped from the dataset. As the sales team added the Tags column, which had no use for analysis, it was dropped. The city column was dropped because it had around 40 missing values and a limited number of categories.

Missing Values Treatment:

Now, each variable in columns with fewer missing percentage value is examined. For categorical columns we have imputed with mode. For numerical columns we have imputed with median checking value counts. In order to reduce the number of dummy variables and improve comprehension, some columns where there are more categories combined percentage is very less we imputed all those values with 'Others'.

Dropping Highly Skewed variables:

We have dropped highly skewed columns like Country, do not call, do not mail, magazine, newspaper, etc. after checking unique categories and value counts. Since the Prospect ID is not required for analysis, we have dropped it. Also, we have dropped all the extra columns.

Analysing Outlier:

In this instance, after checking boxplots, outlier treatment is carried out by updating all values >=99 percentile to 99 percentile using the capping technique.

Analysing Data Imbalance ratio:

From data imbalance ratio, we can see leads not converted are higher compared to leads converted.

EDA:

To check the trends between columns, we have plotted pairplots for all numerical variables and bar graphs for categorical variables. To examine the correlation between the columns, we created a

heatmap. Total Time Spent on Website has been found to have the strongest correlation to Converted (the target variable).

Preparing Data For Model Building:

Binary (Yes/No) variables are transformed to 1/0. Dummy variables were created for each categorical column. The original columns for which the dummies were created were dropped.

Test-Train Split:

We imported in the necessary libraries. Two data frames, X and Y, were created. X will have feature variables and Y will have response/target variable i.e. Converted. The data has now been split into train values (70%) and test values (30%).

Scaling:

We have imported standard scaler from sklearn library. All the three numerical columns from the train set are scaled using Standard scaler technique. We know TotalVisits, Total Time Spent on Website,Page Views Per Visit is on different scale value compared to other columns which are in binary scale, so we can use rescaling to get the coefficients comparable with other variable coefficients.

Model Building:

We build a logistic regression model using train dataset. Feature selection is done by using RFE technique with 15 variables as output. Now that we had those 15 columns, we checked the model. Several variables have high p values. It is best to drop these variables because they add unnecessary complexity to the model and don't really aid in prediction. We can see variable having high P value. Drop first 'What is your current occupation_Housewife' variable to check further. Rre-run the model using the selected variables. Let's predict the value on the train set once we see that the p value is less than 0.05.

Predictions on Train Set:

After we got the predicted values on train set, we created a dataframe with the actual converted flag and the predicted probabilities. Next created new column predicted with 1 if converted probability > 0.5 else 0. We obtained an overall accuracy of 80% after importing metrics and checking the confusion matrix.

We have now verified the VIF values, and they are all good for all variables. We did not need to eliminate any additional variables, so we continued to make predictions solely based on this model. We checked metrics from confusion matrix and sensitivity was around 65%. We must develop a model with high sensitivity in order to identify hot leads that will convert.

ROC and Optimum Point:

We plotted the ROC curve that shows the tradeoff between sensitivity and specificity. We found the optimal cut off probability after plotting accuracy sensitivity and specificity for various range of probabilities. From the curve, observed around 0.28 is the optimum point. But since we have to build a model with good sensitivity, we took 0.3 as cutoff probability. Now we checked the sensitivity and it comes around 77%.

Predictions on Test Set:

We have now made predictions on the test set and added the Lead number to the index. Appended test and predicted final set. We obtained 79% overall accuracy and 77% sensitivity, which is good and in line with train set.

Lead Score Variable:

Finally, we combined two train and test datasets that already contained a generated converted probability column. Then we added a column called Lead Score that denotes the likelihood that a customer will convert. We multiplied probability by 100, the Lead Score variable, for the sales team's benefit.