Lead Scoring Case Study Summary

Problem Statement:

Industry professionals can purchase online courses from X Education, a company that provides education. Several experts who are interested in the courses visit their website on any given day and search for courses.

On numerous websites and search engines like Google, the firm advertises its courses. When arriving at the website, these visitors may browse the courses, submit a form for the course, or watch some videos. These persons are categorized as leads when they fill out a form with their phone number or email address. Also, the business receives leads from earlier recommendations. Once these leads are obtained, sales team members begin calling, sending emails, etc. Some leads are converted during this procedure, but most are not. At X Education, the normal lead conversion rate is roughly 30%.

In the beginning, a lot of leads are generated, but very few of them end up becoming paying clients. To increase lead conversion, you must properly nurture the potential leads during the middle stage (e.g., by educating the leads about the product and maintaining ongoing communication).

You have been asked by X Education to assist them in choosing the leads that have the best chance of becoming paying clients. 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 better chance of converting, while leads with lower lead scores have a lesser chance of converting. The desired lead conversion rate has been estimated by the CEO to be in the range of 80%.

Goals and Objectives:

This case study has a lot of objectives.

- Create a logistic regression model to provide each lead with a lead score between 0 and 100 that the business may use to target potential prospects. In contrast, a lower number would indicate that the lead is chilly and unlikely to convert, while a higher score would indicate that the lead is hot and most likely to convert.

Approach:

From the above problem description, we conclude that the above problem is the classification problem, hence we choose logistic Regression to calculate the Lead rate. Below are the steps followed to solve this problem

Step1: Reading and Understanding Data

Here we tried to get the look and feel of the data, we observed the following things

- ✓ The number of rows and columns and the data types of each column.
- ✓ Checking the first few rows of how the data looks
- ✓ Checking how the data is spread and for duplicates if any.

Step2: Data Cleaning

Here we checked for discrepancies in the dataset

- ✓ The first step to clean the dataset we chose was to drop the variables having unique values.
- ✓ Then, there were a few columns with the value 'Select' which means the leads did not choose any given option. We changed those values to Null values.
- ✓ We dropped some columns having NULL values greater than 35%.
- ✓ Next, we removed the imbalanced and redundant variables. This step also included imputing the missing values as and where required with
    - We used mode imputation for categorical columns.
    - We used mean imputation for numerical columns if there is no skewness in the data.
    - We used median imputation for numerical columns if there is skewness in the data.

Step3: Data Visualization and Outlier Treatment

Here we have done Univariate and Bivariate analysis on the clean data for both categorical and numerical variables and checked for outliers in the dataset:

- ✓ We used univariate analysis on the categorical column to determine which columns make more sense and eliminated those whose variance is almost zero.
- ✓ Additionally, a bivariate analysis was done on categorical columns to determine their variation w. r.t Converted column
- ✓ A univariate analysis was performed on numeric columns by drawing boxplots to see if there were outliers in the data. We used the IQR method to handle outliers in the dataset because only a few columns have outliers.
- ✓ To determine how the leads are related to these columns, we conducted a bivariate analysis on numerical columns with Converted columns.
- ✓ In this step, we also plotted the correlation matrix to identify the columns which are correlated.

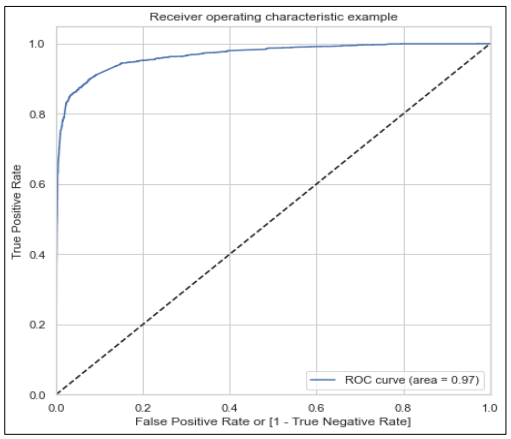
Step4: Data Preparation and Feature Scaling

At this stage, our data was very clean, and no outliers. We know that logistic regression takes the input parameters as numerical values. Hence, we converted all the categorical columns to numerical ones.

- ✓ Columns that have only two levels "Yes" and "No" were converted to numerical using binary mapping.
- ✓ Columns that have more than two levels were converted to dummies using the pd.get_dummies function.
- ✓ The next step was to divide the data set into test and train sections with a proportion of 70-30% values.

Step5: Model Building

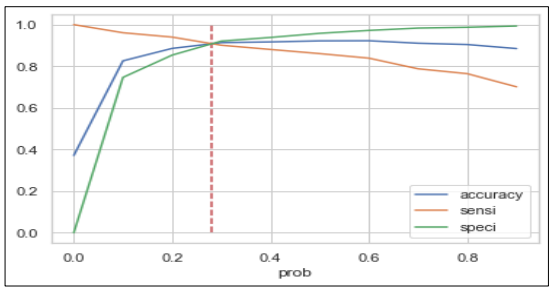
- ✓ Using the Recursive Feature Elimination, we went ahead and selected the 15 top important features. RFE uses the model accuracy to identify which attributes (and combination of attributes) contribute the most to predicting the target attribute
- ✓ Then, we build three models using the glm() function, which is part of the formula submodule of (stats models). The glm() function fits generalized linear models, a class of models that includes logistic regression.
- ✓ Using the statistics generated, we recursively tried looking at the P-values in order to select the most significant values that should be present and dropped the insignificant values.

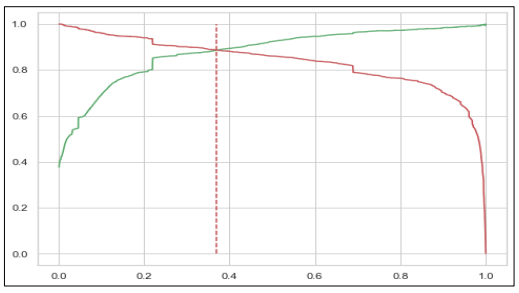


- ✓ Finally, we arrived at the 12 most significant variables. The VIFs for these variables were also found to be good. Variance inflation factor(VIF) is used to treat the multi-collinearity.
- ✓ Once the stable model was created, we predicted probabilities on the train set and created a new column predicted with 1 if the probability is > 0.5 else 0.
- ✓ We calculated the confusion matrix on this predicted column to the actually converted column and also calculated the metrics sensitivity, specificity, precision, recall, and accuracy.
- ✓ Finally, plotted the ROC curve to find the area under the curve.

Step6: Making Predictions on the Train Set

- ✓ We used 0.5 as the cut-off in step 5. We computed the probability with several cut-offs to ensure that it was the best cut.
- ✓ We calculated the 3 metrics — accuracy, sensitivity, and specificity — for probabilities ranging from 0.0 to 0.9.
- ✓ An ideal cutoff of 0.28 was discovered from the intersection of sensitivity, specificity, and accuracy to generate predictions on the training dataset, as shown in the figure:
- ✓ The best cutoff was used to create predictions on the test dataset and was derived from the training dataset's Precision recall graph, as shown in the figure:
- ✓ Observably, the trade-off between precision and recall is 0.37. We can therefore confidently decide to classify any Prospect Lead with Conversion Probability greater than 37% as a hot lead.





Step7: Making Predictions on the Test set

After finalizing the optimum cutoff and calculating the metrics on the train set, we predicted the data on the test data set. Below are the observations:

Train Data:

- ✓ -Accuracy : 92%; Sensitivity : 90%; Specificity : 92%; Precision : 92%; Recall : 86%,

Test Data:

- ✓ Accuracy : 92%; Sensitivity : 89%; Specificity : 94%; Precision : 90%; Recall : 89%

Step8: Conclusion

- ✓ The Model seems to predict the Conversion Rate very well on the final predicted model which meets the expectation of the CEO and has given a ballpark of the target lead conversion rate to be around 80%.
- ✓ The good value of sensitivity of our model will help to select the most promising leads.