Lead Scoring Case Study Summary

Case Study Problem Statement

X Education is an ed-tech firm that targets working professionals and provide them online courses. The company markets its courses on several websites and search engines like Google. They reach out to customers using emails, phone calls, referrals, etc. Their business goal is to increase the conversion rate from 30% to 80% i.e. the leads changing to customers. The objective is to identify the most potential leads, also known as 'Hot Leads'. If they successfully identify this set of leads, the lead conversion rate should go up as the sales team will now be focusing more on communicating with the potential leads rather than making calls to everyone.

Solution (Our Approach)

Our goal is to target hot leads i.e. the leads which have higher chances of converting. For this we built a logistic regression model which gives probabilistic lead score to each lead and also identify the features that play major role in decision making. Leads with higher lead score will have higher chance of converting. This will not only give leads with higher chances of conversion but will also let the sales representative to focus more on hot leads instead of wasting time on leads with no or lower chances of conversion.

The following are steps that we used to achieve the solution for the mentioned problem statement:

1. **Inspecting the data**

   We started with checking the data i.e. count of columns, tuples. Then we checked the data for missing values. We concluded that the data was almost clean with a few missing values.

2. **Feature Analysis and Cleaning**

   - We started with replacing 'Select' values with np.NaN as the 'Select" value is as good as nan.
   - Then we dropped a few columns that were of no use.
   - Since we replaced 'Select' values with NaNs, we again check the percentage of missing values in each column.
   - We dropped all the columns with more than 45% missing values.
   - Once we had data ready for cleaning and preprocessing, we separated continuous and categorical features.
   - We started with cleaning the categorical columns which included imputing NaNs, grouping values with lower frequency under one tag.
   - We again dropped some columns which were highly unbalanced because of 97-98% presence of one class.
   - With cleaning the categorical feature we also plotted graphs for them to get clearer picture and distribution of data.

- Now coming to numerical features, we started with EDA, we plotted Conversions against the numerical feature. We found out that there were presence of outliers in good number.
- We removed the outlier using the statistical method from all the numerical variables.
- We again plotted boxplots for the numerical features and also we plotted a heatmap to check correlations of the features.

3. Encoding/ Dummy Variables Creation

Feature encoding or dummy variable creation is required as the logistic regression model requires numerical data. We used two techniques to for this step:

- Binary Encoding: We used binary encoding for features with two two classes i.e yes and no. For this we created a function that maps Yes to 1 and No to 0.
- Creating Dummy Variables: We used this to create dummies of all other categorical features and in this we dropped one class from all features in order to decrease the number of features.

4. Train-Test Data Split

We have taken 70-30 ratio for train test data respectively.

5. Data Transformation [Scaling the features]

We have used Standard Scaler to scale all the numerical features.

6. Model Development

We have used statsmodel's GLM module for modeling. For feature selection we have used statsmodel's VIF and Sklearn's RFE.

The model building is an iterative procedure so we took the following considerations:

- There are 63 features, hence trying out different combinations will be tedious.
- We will be using Sklearn RFE and statsmodels VIF in conjunction to filter out the significant features.
- We will be using GLM with statsmodels for our logistic regression models.

We devised a criteria for feature selection:

- Remove feature if high p-value
- Features should have p-value<=0.05 [it signifies that features are statistically significant]
- Features should have VIF <=5 [it signifies absense of multicollinearity]
- ALWAYS REMOVE ONE FEATURE AT A TIME.

Once we finalized the features and created the final model, we concluded with 12 features which play major role in decision making.

7. Model Validation

- We used all the metrics like accuracy, specificity, sensitivity, precision, recall to validate our model.

- We plotted the ROC curve to check how good our model is and for further optimization.
- To find optimal cutoff threshold, we took several cut-offs in consideration and plotted a graph to with accuracy, specificity, sensitivity to find the optimal cut-off for scoring.
- We found 0.26 as the optimal cutoff with 96% Specificity and 86% Sensitivity on the test data prediction.