Lead Scoring Case Study Summary

Problem Statement:

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

The company needs a model wherein a lead score is assigned to each lead so that the customers with higher lead scores have a higher conversion chance and the customers with lower lead scores have a lower conversion chance.

The CEO has given a ballpark of the target lead conversion rate to be around 80%.

Solution Summary:

Step 1: Reading and Understanding Data.

We have used the Jupiter notebook for the python coding. We have imported the Leads data provided to the notebook using python's panda library and analysed the data.

Step2: Data Cleaning

We dropped the variables with a high percentage of NULL values. This step also included imputing the missing values as and where required with median values in case of numerical variables and the creation of new classification variables in case of categorical variables. The outliers were identified and removed.

Step 3: Data Analysis.

Then we started with the Exploratory Data Analysis of the data set to get a feel of how the data is oriented. In this step, some variables were identified to have only one value in all rows. These variables were dropped.

Step 4: Creating Dummy Variables.

We went on with creating dummy data for the categorical variables.

Step 5: Test Train Split

The next step was to divide the data set into test and train sections with a proportion of 70-30% values.

Step 6: Feature Rescaling.

We used the Min Max Scaling to scale the original numerical variables. Then using the stats model, we created our initial model, which would give us a complete statistical view of all the parameters of our model.

Step 7: Feature selection using RFE.

Using the Recursive Feature Elimination, we went ahead and selected the 15 top important features. Using the statistics generated, we recursively tried looking at the P-values to select the most significant values that should be present and dropped the insignificant values.

Finally, we arrived at the 13 most significant variables. The VIF's for these variables were also found to be good.

We then created the data frame having the converted probability values and we had an initial assumption that a probability value of more than 0.5 means 1 else 0.

Based on the above assumption, we derived the Confusion Metrics and calculated the overall Accuracy of the model.

We also calculated the **'Sensitivity'** and the **'Specificity'** matrices to understand how reliable the model is.

Step 8: Plotting the ROC Curve

We then tried plotting the ROC curve for the features and the curve came out to be decent with an area coverage of 98% which further solidified the model.

Step 9: Finding the Optimal Cut-off Point

Then we plotted the probability graph for the **'Accuracy', 'Sensitivity',** and **'Specificity'** for different probability values. The intersecting point of the graphs was considered the optimal probability cut-off point. The cut-off point was found to be 0.32

Based on the new value we could observe that close to 93% of values were rightly predicted by the model.

We could also observe the new values of the **'Accuracy=93.4%**, **'Sensitivity=93.04%'**, and **'Specificity=93.5%'**.

Also calculated the lead score and figured that the final predicted variables approximately gave a target lead prediction of 90%

Step 10: Computing the Precision and Recall metrics

We also found out the Precision and Recall metrics values came out to be 89.7% and 93.04% respectively on the train data set.

Step 11: Making Predictions on Test Set

Then we implemented the learnings to the test model and calculated the conversion probability based on the **Sensitivity** and **Specificity** metrics and found the **Accuracy** value to be **94.1%; Sensitivity=93.76%; Specificity= 94.33%.**