LEAD SCORING CASE STUDY

Problem Statement:

An education company named X Education markets its courses on several websites and search engines like Google. Once these people land on the website, they might browse the courses or fill up a form for the course or watch some videos. When these people fill up a form providing their email address or phone number, they are classified to be a lead. Moreover, the company also gets leads through past referrals. Once these leads are acquired, employees from the sales team start making calls, writing emails, etc. Through this process, some of the leads get converted while most do not. The typical lead conversion rate at X education is around 30%.

Now, although X Education gets a lot of leads, its lead conversion rate is very poor. For example, if, say, they acquire 100 leads in a day, only about 30 of them are converted. To make this process more efficient, the company wishes 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.

Business Goal:

- To help the company to select the most promising leads, i.e. the leads that are most likely to convert into paying customers.
- To come up with a model wherein it needs to assign a lead score to each of the leads such that the customers with a higher lead score have a higher conversion chance and the customers with a lower lead score have a lower conversion chance.

Data Overview:

- The shape of the dataset leads is (9240, 37).
- The target variable in this case study, is the column 'Converted' which tells whether a past lead was converted or not. Wherein 1 means it was converted and 0 means it wasn't converted.

Data Preprocessing:

- Most of the columns have level Select. These are the cases where user has not selected anything while filling the form. There are similar to missing values and these can be replaced with nan in python.
- It is noticed that most of the columns have missing values and some have high percentage of missing values. We will drop the features having missing values almost above 40%.

- Impute the missing values with its mode if it is categorical variable and with median if it is numerical and has skewness in data.

Data Visualization:

- EDA is performed to understand the dataset.
- We have 38.5% of the leads who have converted and 61.5% who haven't. It seems data is balanced.
- A univariant, bivariant and multivariant analysis is performed to understand data and how the features are related to each other.
- It is noticed that, we don't very high correlation between target variable and other variables.

Data Preparation:

- Most of the features has Yes/No binary values and it can be replaced with 1/0 numerical values.
- There are nominal features for which we have applied one hot encoding by creating dummy variables.
- It is noticed there are outliers and it is handled by replacing it with median.
- Next, we have chosen highly positively correlated and negatively correlated features to target variable for building the model.

Model Building:

- We have split the dataset into train and test dataset with 70% and 30%
- Further, scaling is performed using Standard Scaler.
- Since the target variable is a categorical variable, we have applied Logistic Regression and used Recursive Feature Elimination technique to choose 11 features initially.
- After creating the model using stats model with 11 features, it is noticed that p-value of all the coefficients looks good but couple of variables have high VIF (high correlation).
- Rebuild the model (model2) after dropping one feature with high correlation.
- The second model that is built seems to have p-values and VIF within the range, thus we can proceed to perform model evaluation on train and test datasets.

Model Evaluation:

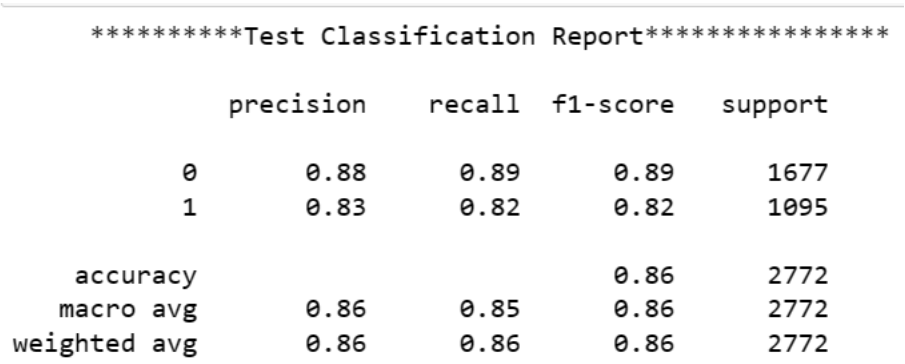
- The performance metrics accuracy, sensitivity and specificity for various probability cutoffs are plotted and identified 0.3 has the optimum cutoff.
- At 0.3 cutoff, the scores of accuracy, sensitivity and specificity are 0.83, 0.86 and 0.82 respectively.
- Later we plotted precision and recall curve and identified the optimum cutoff point at 0.4
- Further we proceeded with 0.4 cutoff and calculated the precision and recall metrics on train set which are as below:

    - accuracy: 0.87
    - precision: 0.83
    - recall: 0.82
    - f1_score: 0.82
    - roc auc score: 0.85

- The above metrics we derived seems doing good. Now we will evaluate the model on test dataset.
- We derived below metrics after evaluating the model on test dataset.

  - accuracy: 0.86
  - precision: 0.83
  - recall: 0.82
  - f1_score: 0.82
  - roc auc score: 0.85

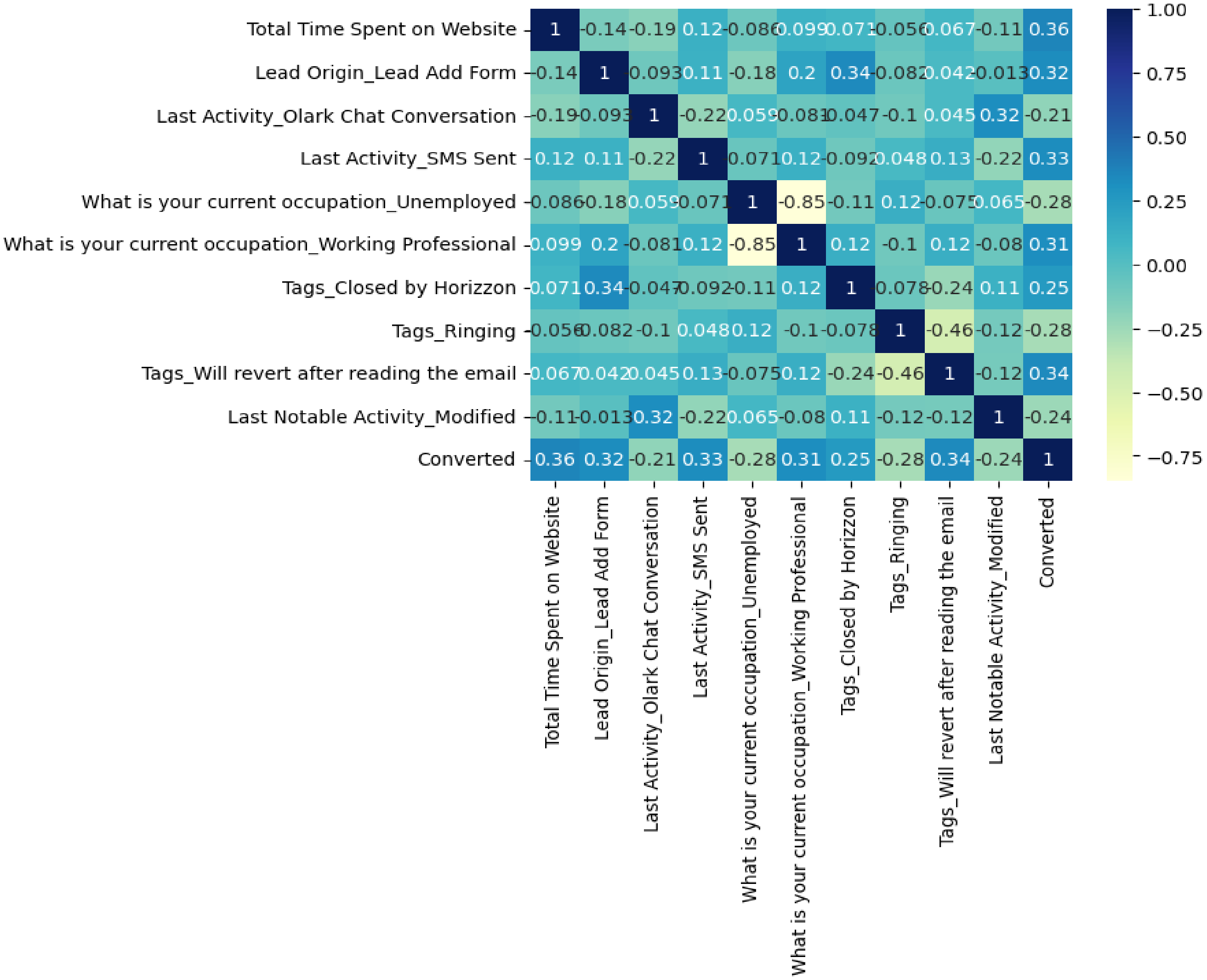
- Based on the above metrics on both train and test datasets, we conclude that this is a decent model.



```
          **********Test Classification Report****************

                  precision    recall  f1-score   support

               0       0.88      0.89      0.89      1677
               1       0.83      0.82      0.82      1095

        accuracy                           0.86      2772
       macro avg       0.86      0.85      0.86      2772
    weighted avg       0.86      0.86      0.86      2772
```

Conclusion:

As we noticed we derived a reasonable metrics with the model we have built. Hence the model does a decent job. Below are the final predictors that can be considered to predict if a lead would get converted or not.


Total Time Spent on Website
Lead Origin_Lead Add Form
Last Activity_Olark Chat Conversation
Last Activity_SMS Sent
What is your current occupation_Unemployed
What is your current occupation_Working Professional
Tags_Closed by Horizzon
Tags_Ringing
Tags_Will revert after reading the email
Last Notable Activity_Modified
Converted
1 -0.14 -0.19 0.12 -0.086 0.099 0.071 -0.056 0.067 -0.11 0.36
-0.14 1 -0.093 0.11 -0.18 0.2 0.34 -0.082 0.042 -0.013 0.32
-0.19 -0.093 1 -0.22 0.059 -0.081 -0.047 -0.1 0.045 0.32 -0.21
0.12 0.11 -0.22 1 -0.071 0.12 -0.092 0.048 0.13 -0.22 0.33
-0.086 -0.18 0.059 -0.071 1 -0.85 -0.11 0.12 -0.075 0.065 -0.28
0.099 0.2 -0.081 0.12 -0.85 1 0.12 -0.1 0.12 -0.08 0.31
0.071 0.34 -0.047 -0.092 -0.11 0.12 1 -0.078 -0.24 0.11 0.25
-0.056 -0.082 -0.1 0.048 0.12 -0.1 -0.078 1 -0.46 -0.12 -0.28
0.067 0.042 0.045 0.13 -0.075 0.12 -0.24 -0.46 1 -0.12 0.34
-0.11 -0.013 0.32 -0.22 0.065 -0.08 0.11 -0.12 -0.12 1 -0.24
0.36 0.32 -0.21 0.33 -0.28 0.31 0.25 -0.28 0.34 -0.24 1
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75