Lead Scoring Case Study – Summary Document

Problem Statement:

To build a Logistic Regression model to predict the conversion probability for leads in an ed tech company, assigning a score between 0-100 against each lead and validating the findings against actual conversion results.

Approach

The team worked together to analyze the data file and went about with the basic analysis, data cleaning, handling of null values etc.

There were multiple variables which had over 3000 null values which were removed, including Lead Quality, various Asymmetrique score indices, Tags.

```
In [10]: leads_df.isnull().sum().sort_values(ascending=0)

Out[10]: Lead Quality                                     4767
         Asymmetrique Activity Index                      4218
         Asymmetrique Profile Score                       4218
         Asymmetrique Activity Score                      4218
         Asymmetrique Profile Index                       4218
         Tags                                             3353
         Lead Profile                                     2709
         What matters most to you in choosing a course    2709
         What is your current occupation                  2690
         Country                                          2461
         How did you hear about X Education               2207
         Specialization                                   1438
         City                                             1420
         Page Views Per Visit                              137
         TotalVisits                                       137
         Last Activity                                     103
```

Variables such as City, Country etc were also removed since they didn't seem to have any impact on the outcome of conversion.

"Lead Profile", and 'How did you hear about X Education' were also removed as they had a high number of the value "Select" which was as good as null.

This was followed by the dummy variable creation for the categorical variables. Post this we went on to build the model. Around 31% of the leads had to be removed for null values in addition to multiple columns that seemed redundant or where the data were dominated by a single value. These included 'Do Not Call', 'Search', 'Magazine', 'Newspaper Article', 'X Education Forums', 'Newspaper', 'Digital Advertisement', 'Through Recommendations', 'Receive More Updates About Our Courses', 'Update me on Supply Chain Content', 'Get updates on DM Content', 'I agree to pay the amount through cheque'

Categorical variables removed after creating dummies - 'Lead Origin', 'Lead Source', 'Do Not Email', 'Last Activity', 'Specialization', 'What is your current occupation', 'A free copy of Mastering The Interview', 'Last Notable Activity'

Model Building

We used a minmax scaler to scale the numerical variables('TotalVisits','Total Time Spent on Website','Page Views Per Visit') with a 70:30 split on Train vs Test data.

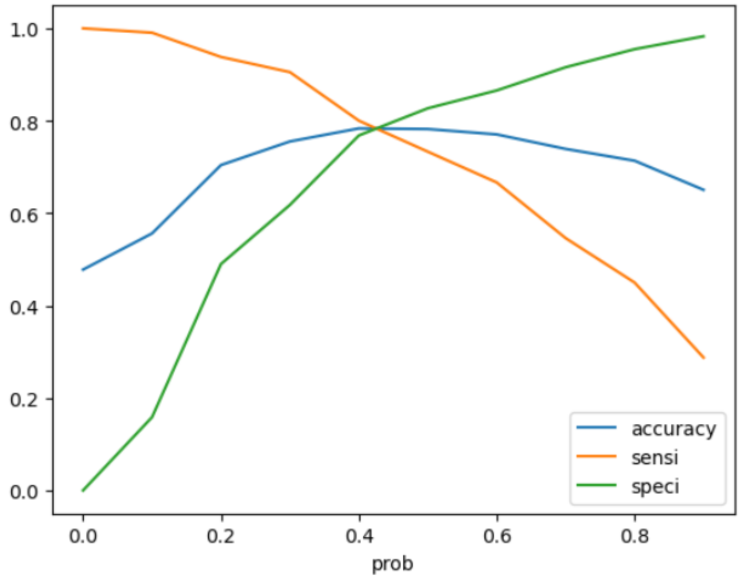
The logistic regression model was built with up to 1000 iterations for 15 variables. Post this we looked at the variables with P>0.05 and high vif values which indicated high correlation between the variables. Following this we were left with the below variables and co-efficients.

	coef	std err	z	P>|z|	[0.025	0.975]
const	-0.6474	0.585	-1.107	0.268	-1.793	0.498
TotalVisits	4.0447	1.199	3.375	0.001	1.696	6.394
Total Time Spent on Website	4.3198	0.184	23.421	0.000	3.958	4.681
Lead Origin_Lead Add Form	3.5342	0.227	15.553	0.000	3.089	3.980
Lead Source_Olark Chat	1.5566	0.126	12.366	0.000	1.310	1.803
Lead Source_Welingak Website	2.0778	0.752	2.764	0.006	0.604	3.551
Do Not Email_Yes	-1.5573	0.193	-8.079	0.000	-1.935	-1.179
Last Activity_Converted to Lead	-1.1403	0.238	-4.795	0.000	-1.606	-0.674
Last Activity_Olark Chat Conversation	-1.3210	0.184	-7.163	0.000	-1.682	-0.960
Last Activity_SMS Sent	1.0674	0.084	12.740	0.000	0.903	1.232
What is your current occupation_Student	-1.3919	0.617	-2.255	0.024	-2.602	-0.182
What is your current occupation_Unemployed	-1.4870	0.581	-2.559	0.010	-2.626	-0.348
What is your current occupation_Working Professional	1.3025	0.613	2.125	0.034	0.101	2.504
Last Notable Activity_Unreachable	2.5712	0.814	3.158	0.002	0.975	4.167

While we then started with an arbitrary value of 0.5 against the final calculated probability, we were then able to arrive at the optimum cutoff value by using the ROC curve and plotting accuracy, sensitivity and specificity, which took us to the value of 0.42 as cutoff.

In [87]:
```
# Let's plot it as well

cutoff_df.plot.line(x='prob', y=['accuracy','sensi','speci'])
plt.show()
```



As you can see that around 0.42, you get the optimal values of the three metrics. So let's choose 0.42 as our cutoff now.

Using this cutoff value we tagged the output variables and then calculated the accuracy and sensitivity

Accuracy: 78.65%
Sensitivity: 78.89%
Specificity: 78.45%

In [89]:
```
# Let's check the accuracy now

metrics.accuracy_score(y_train_pred_final.Converted,
                       y_train_pred_final.final_predicted)
```

Out[89]: `0.7865949338713293`

In [90]:
```
# Creating confusion matrix once again

confusion2 = metrics.confusion_matrix(y_train_pred_final.Converted, y_train_pred_final.final_predicted )
confusion2
```

Out[90]:
```
array([[1827,  502],
       [ 450, 1682]], dtype=int64)
```

In [91]:
```
# Let's evaluate the other metrics as well

TP = confusion2[1, 1]  # true positive
TN = confusion2[0, 0]  # true negatives
FP = confusion2[0, 1]  # false positives
FN = confusion2[1, 0]  # false negatives
```

In [92]:
```
# Calculate Sensitivity

TP/(TP+FN)
```

Out[92]: `0.7889305816135085`

In [93]:
```
# Calculate Specificity

TN/(TN+FP)
```

Out[93]: `0.7844568484328038`

Post this, we then tested the model against the test data, and the model showed the following parameters.

Accuracy: 79.86%
Sensitivity: 79.2%
Specificity: 80.49%

```
In [111]: # Let's check the overall accuracy

          metrics.accuracy_score(y_pred_final['Converted'], y_pred_final.final_predicted)

Out[111]: 0.7986401673640168
```

```
In [112]: confusion2 = metrics.confusion_matrix(y_pred_final['Converted'],
                                                y_pred_final.final_predicted)
          confusion2

Out[112]: array([[788, 191],
                 [194, 739]], dtype=int64)
```

```
In [113]: TP = confusion2[1, 1]  # true positive
          TN = confusion2[0, 0]  # true negatives
          FP = confusion2[0, 1]  # false positives
          FN = confusion2[1, 0]  # false negatives
```

```
In [114]: # Calculate sensitivity
          TP / float(TP+FN)

Out[114]: 0.7920685959271169
```

```
In [115]: # Calculate specificity
          TN / float(TN+FP)

Out[115]: 0.804902962206333
```

We finally calculated the precision and recall values as well which were necessary in addressing the subjective questions that were also part of the assignment.

Precision and Recall

```
In [116]: confusion2[1, 1] / (confusion2[0, 1] + confusion2[1, 1])

Out[116]: 0.7946236559139785
```

```
In [117]: # Recall
          confusion2[1, 1] / (confusion2[1, 0] + confusion2[1, 1])

Out[117]: 0.7920685959271169
```

This is how the team went about the given problem statement and proceeded to create the various documents needed for the assignment.

We finally narrowed down the key variables influencing a conversion to the number of site visits, time spent on the website as well as Lead Source while removing a lot of the redundant variables that seemingly had little effect on the conversions.