Lead Scoring Case Study

Ho Trung Hieu

hotrunghieu94@gmail.com

Business Problem Statement

- An education company named X Education sells online courses to industry professionals. On any given day, many professionals who are interested in the courses land on their website and browse for courses.
- The company 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%.
- Goal:

1. To identify the features that contributes to predict Lead Conversion.

2. Identifying Hot Leads by generating Lead Score for all leads, so that leads having higher Lead Scores can be contacted with priority for achieving Higher Lead Conversion Rate

Check data and prepare data

- Checking missing values
- Look at special signs in categorical variables

```
#check missing values
round(lead_df.isnull().sum()/lead_df.shape[0], 2)
```

✓ 0.0s

```
Prospect ID                                    0.00
Lead Number                                    0.00
Lead Origin                                    0.00
Lead Source                                    0.00
Do Not Email                                   0.00
Do Not Call                                    0.00
Converted                                      0.00
TotalVisits                                    0.01
Total Time Spent on Website                    0.00
Page Views Per Visit                           0.01
Last Activity                                  0.01
Country                                        0.27
Specialization                                 0.16
How did you hear about X Education             0.24
What is your current occupation                0.29
What matters most to you in choosing a course  0.29
Search                                         0.00
Magazine                                       0.00
Newspaper Article                              0.00
X Education Forums                             0.00
Newspaper                                      0.00
Digital Advertisement                          0.00
Through Recommendations                        0.00
Receive More Updates About Our Courses         0.00
Tags                                           0.36
...
Asymmetrique Profile Score                     0.46
I agree to pay the amount through cheque       0.00
A free copy of Mastering The Interview         0.00
Last Notable Activity                          0.00
dtype: float64
```

Prepare the initial data

- For features requiring users to select but they don't select anything, convert to NaN
- Checking the asymmetrique features, keep only the necessary ones and remove the rest

```
# For features requiring users to select but they don't select anything, convert to NaN

lead_df = lead_df.replace('Select', np.nan)
```

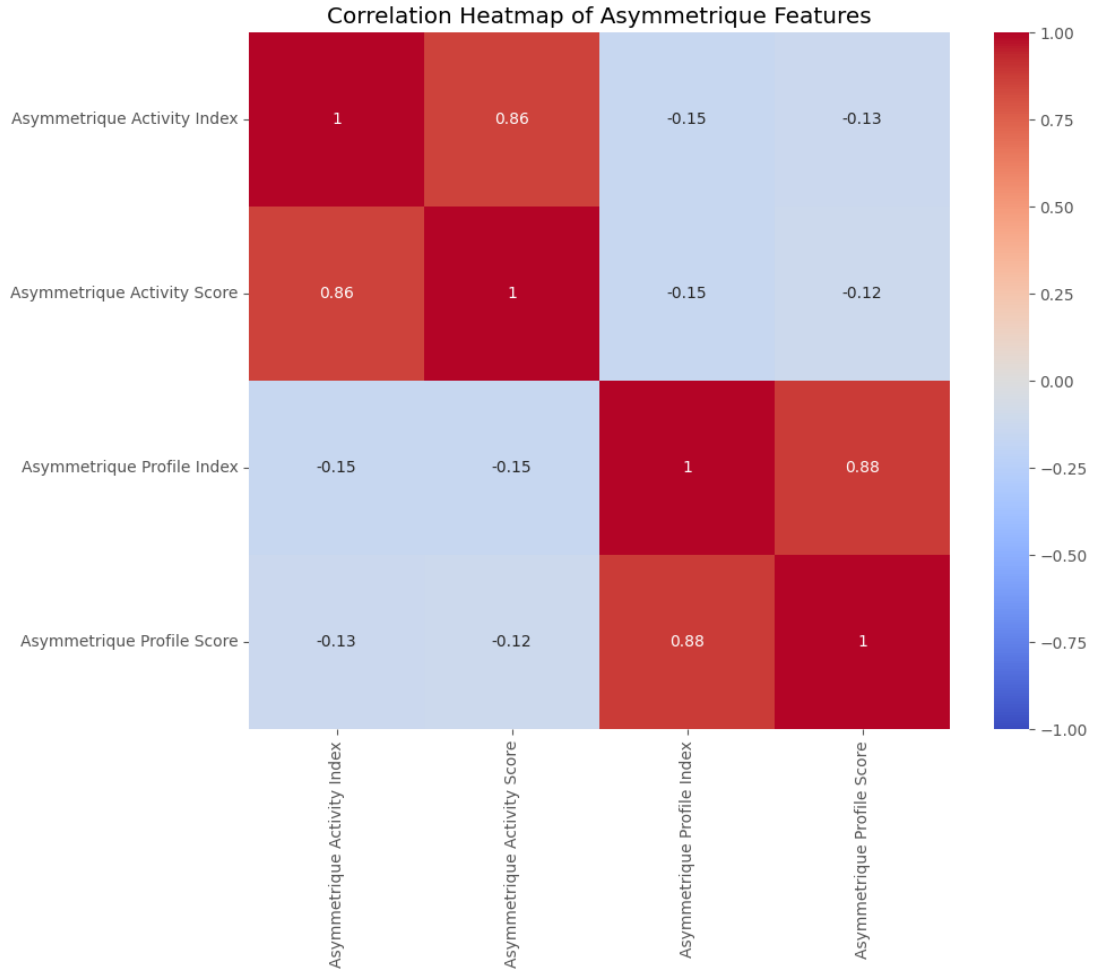
✓ 0.0s

Asymmetrique features

- Checking the asymmetrique features, keep only the necessary ones and remove the rest

```
Unique values in Asymmetrique Activity Index:
[ 2.  3.  1. nan]

Unique values in Asymmetrique Activity Score:
[15. 14. 13. 17. 16. 11. 12. 10.  9.  8. 18. nan  7.]

Unique values in Asymmetrique Profile Index:
[ 2.  3.  1. nan]

Unique values in Asymmetrique Profile Score:
[15. 20. 17. 18. 14. 16. 13. 19. 12. nan 11.]
```

```
# Calculate and display correlation matrix for Asymmetrique features
asymmetrique_features = ['Asymmetrique Activity Index', 'Asymmetrique Activity Score',
                         'Asymmetrique Profile Index', 'Asymmetrique Profile Score']

correlation_matrix = lead_df[asymmetrique_features].corr()

print("Correlation Matrix for Asymmetrique Features:")
print(correlation_matrix)

# Create a heatmap visualization for better interpretation
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(10,8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', vmin=-1, vmax=1)
plt.title('Correlation Heatmap of Asymmetrique Features')
plt.show()
```



Missing values

- Remove any columns with missing percentage >70%
- The remaining columns with missing percentage <70%, I impute using mode or Others for catergorical data

```
# Fill null values with specified values for each column
lead_df['Lead Quality'] = lead_df['Lead Quality'].fillna('Not sure')
lead_df['City'] = lead_df['City'].fillna('Mumbai')
lead_df['Specialization'] = lead_df['Specialization'].fillna('Other')
lead_df['Tags'] = lead_df['Tags'].fillna('Will revert after reading the email')
lead_df['What matters most to you in choosing a course'] = lead_df['What matters most to you in choosing a course'].fillna('Better Career Prospects')
lead_df['What is your current occupation'] = lead_df['What is your current occupation'].fillna('Unemployed')
lead_df['Country'] = lead_df['Country'].fillna('India')

# Verify the changes by checking null percentages again
null_percentages = (lead_df.isnull().sum() / len(lead_df) * 100).round(2).sort_values(ascending=False)
print("\nUpdated Null Value Percentages:")
print(null_percentages)
```

✓ 0.0s

```
Updated Null Value Percentages:
Asymmetrique Activity Index            45.65
Asymmetrique Profile Index             45.65
TotalVisits                             1.48
Page Views Per Visit                    1.48
```

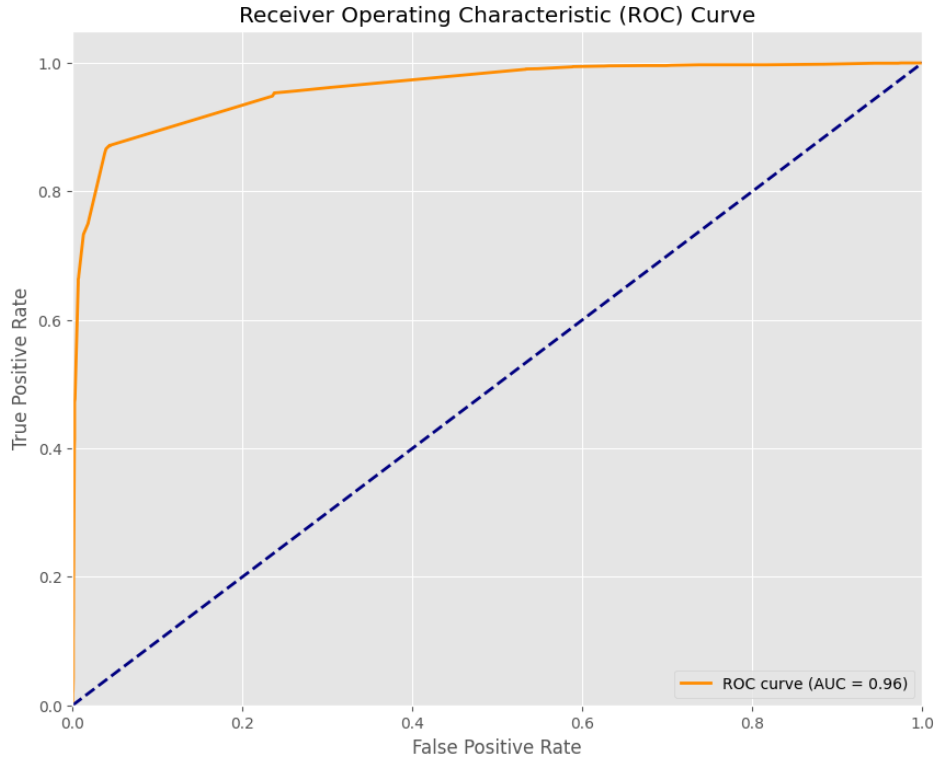
Check and remove unbalance columns

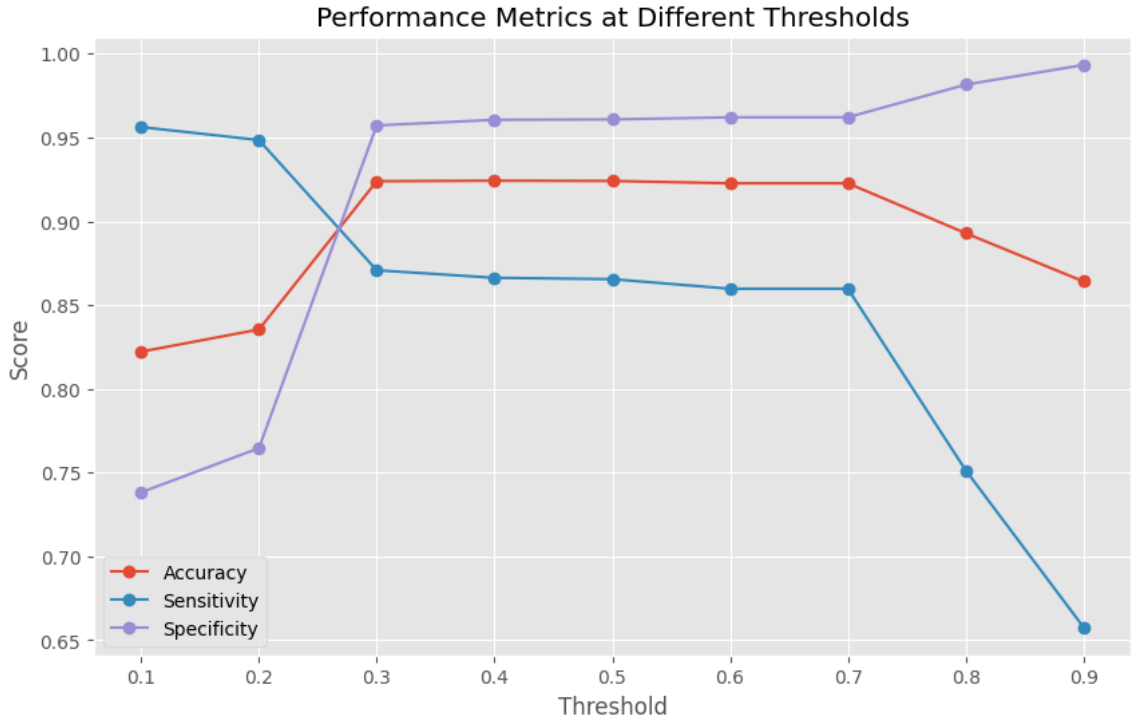
Columns with super skewed distribution is then checked and removed

```
# Drop specified unbalanced columns
columns_to_drop = ['Do Not Email', 'Do Not Call', 'Country',
                   'What is your current occupation',
                   'What matters most to you in choosing a course',
                   'Magazine', 'Newspaper Article', 'X Education Forums',
                   'Newspaper', 'Digital Advertisement',
                   'Through Recommendations',
                   'Update me on Supply Chain Content',
                   'Get updates on DM Content',
                   'I agree to pay the amount through cheque']

lead_df = lead_df.drop(columns=columns_to_drop)

# Verify remaining columns
print("Remaining columns in dataset:")
print(lead_df.columns.tolist())
```

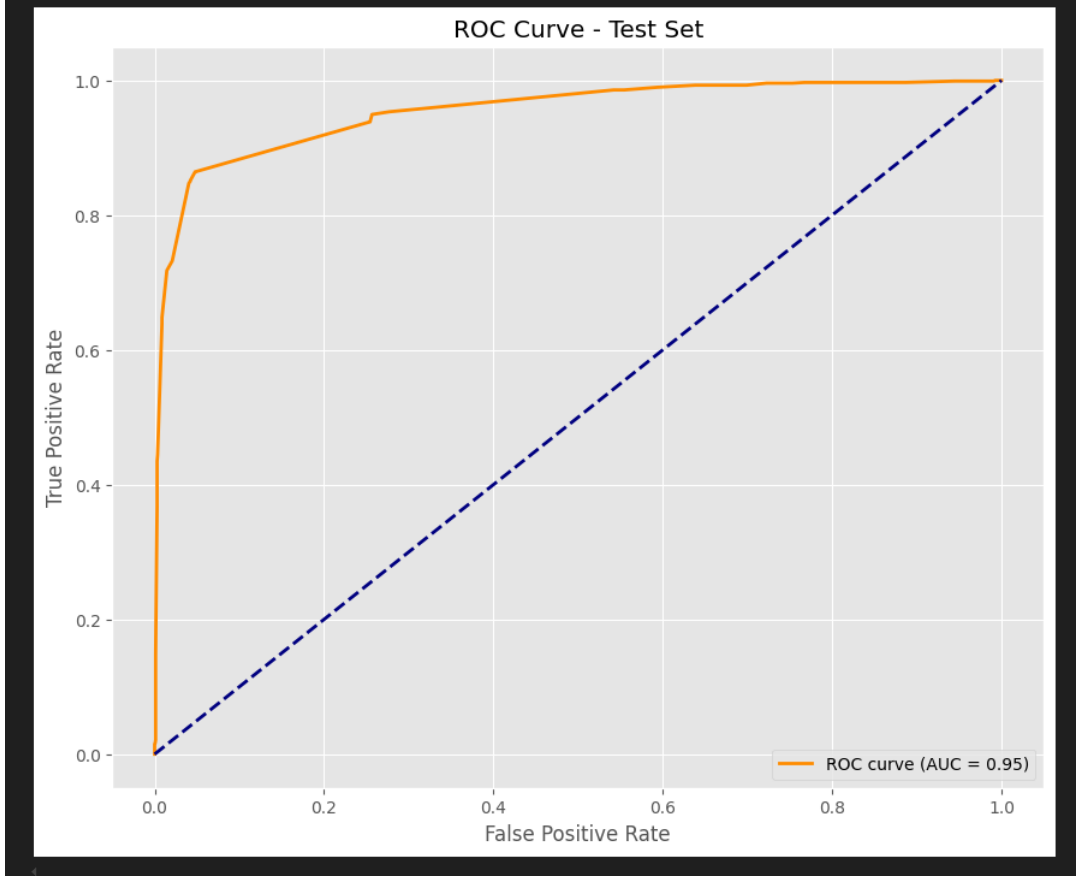
Build the logistic regression model

- Test-train split
- StandardScaler
- RFE feature selection
- Eveluate with Statmodels
- Finding optimal cut-off value





```
Test Set Performance Metrics:
Accuracy: 0.920
Precision: 0.912
Sensitivity: 0.865
Specificity: 0.952
```



Fit the model to the test data and feature importance

const	-1.942030
Lead Origin_Lead Add Form	1.086138
Lead Source_Welingak Website	3.403782
Tags_Busy	1.354397
Tags_Closed by Horizzon	9.318964
Tags_Lost to EINS	10.132080
Tags_Ringing	-2.569160
Tags_Will revert after reading the email	4.792272
Tags_invalid number	-28.940255
Tags_number not provided	-16.181205
Tags_switched off	-2.933269
Tags_wrong number given	-8.047377
Lead Quality_Not sure	-4.134132
Lead Quality_Worst	-3.368213
Last Notable Activity_Modified	-1.055338