LEAD SCORING CASE STUDY

Group members:

Hrudya Sreejith

Hitender Thakur

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%.

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.

Goals:

1.To build a logistic regression model to assign a lead score between 0 and 100 to each of the leads which can be used by the company to target potential leads.

2.To adjust to if the company's requirement changes in the future so you will need to handle these as well.

Approach:

- The method used in this case study is Logistic Regression. Logistic Regression is a supervised classification model. It allows you to make predictions from labelled data, if the target (output) variable is categorical.
- In this data set, our target variable is “Converted”, which implies whether a potential lead got converted as payable customer or not.

Steps:

1. Importing libraries and Understanding data : Imports necessary libraries, understands and read the given data.
2. Data Cleaning : Check for missing values and dropping the columns which are not useful for analysis.
3. Data Visualization: Univariate and Bivariate analysis using sns and matplotlib
4. Data Preparation: Creating dummy variables for categorical variables with multiple levels.
5. Train – Test Split : Splitting the data into train and test datasets.
6. Feature Scaling: Scaling using MinMaxScaler() to ensure all the variables are of same scale.
7. Correlation: Checking the correlation using heat map.
8. Model building: Using RFE method as the number of variables are huge in number.
9. Model evaluation: Evaluating the model build on train dataset.
10. Plotting ROC Curve
11. Finding optimal cut-off point
12. Making predictions on test datasets.

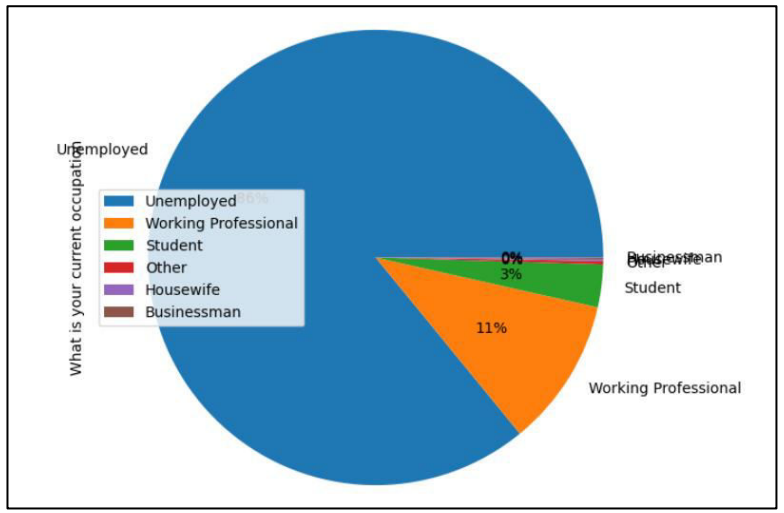
Importing libraries:

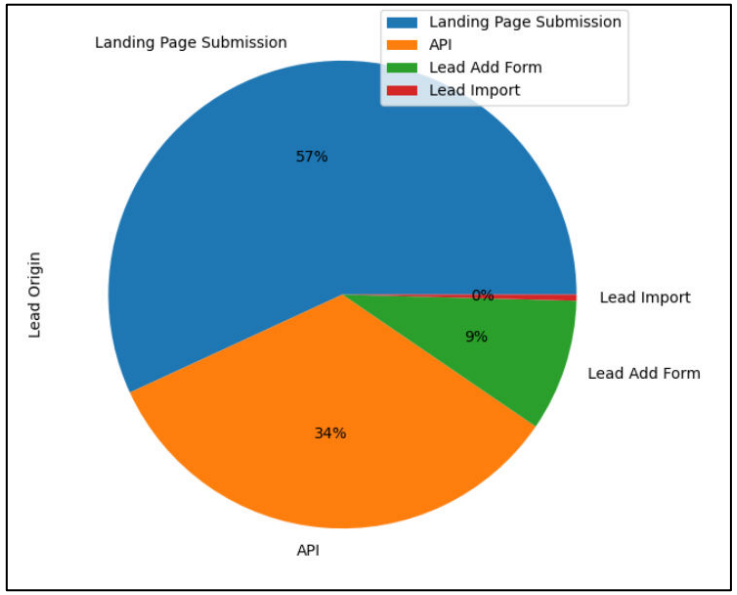
```
# Importing libraries:

import warnings
warnings.filterwarnings('ignore')

import pandas as pd, numpy as np

import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import MinMaxScaler

import statsmodels.api as sm

from sklearn.linear_model import LogisticRegression
from sklearn.feature_selection import RFE

from statsmodels.stats.outliers_influence import variance_inflation_factor
```

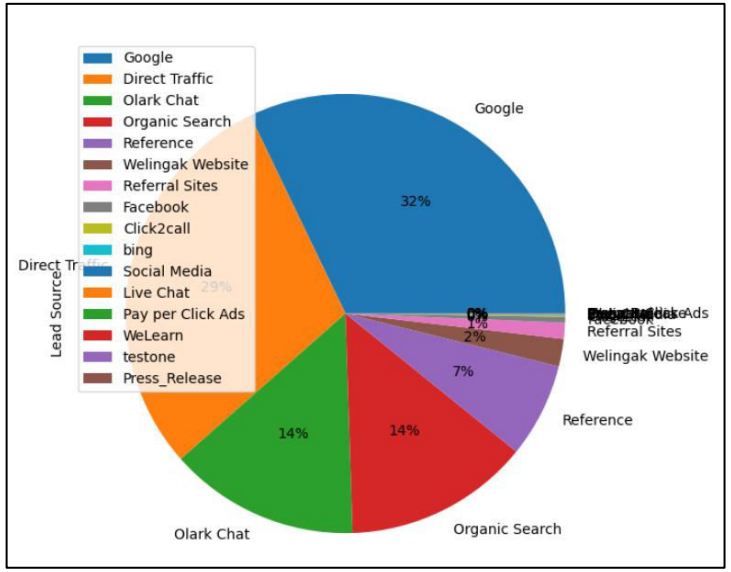
Data Cleaning:

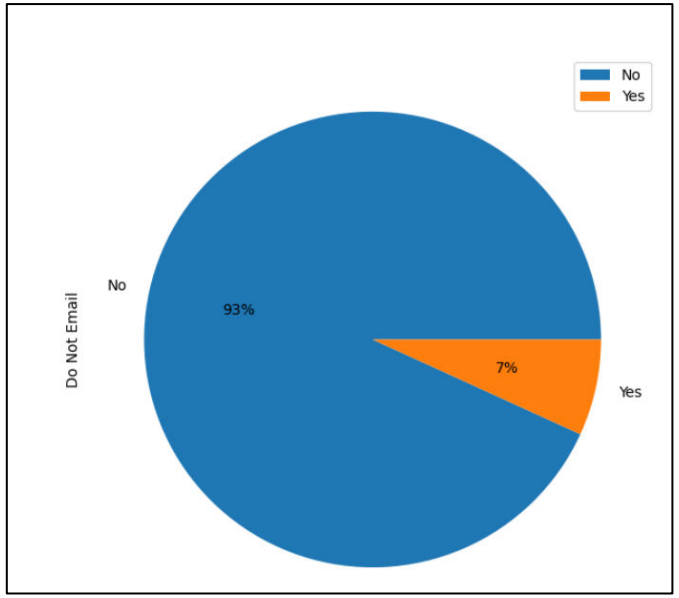
- Columns such as 'Tags', 'Lead Quality', 'Asymmetrique Activity Index','Asymmetrique Profile Index', 'Asymmetrique Activity Score',Asymmetrique Profile Score', 'City' , 'Country' are dropped as they have more than 35% missing values.
- Columns which plays no role for our analysis as most of the data points have only one value are also dropped.
- For columns "What is your current occupation", "TotalVisits", "Lead Source" and "Specialization" null value rows have been dropped.

Univariate Analysis:





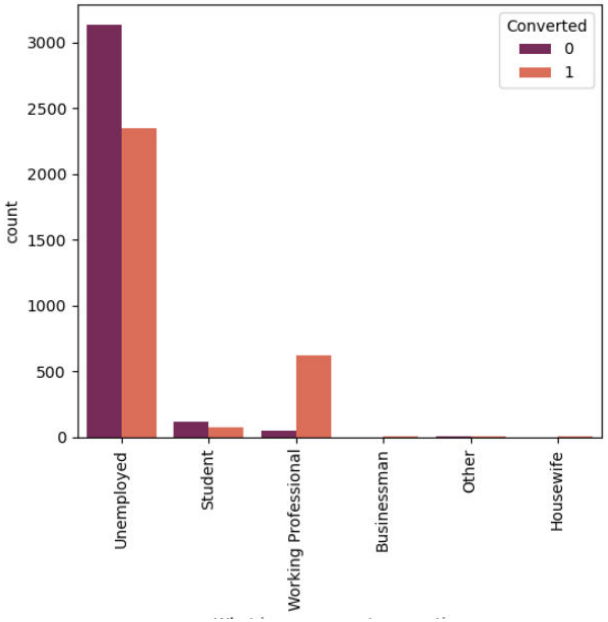


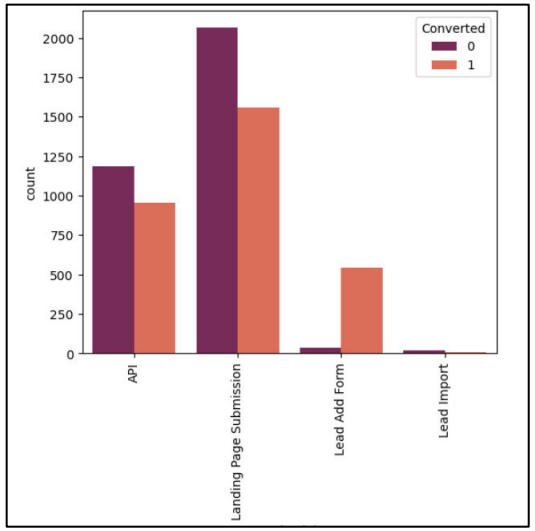


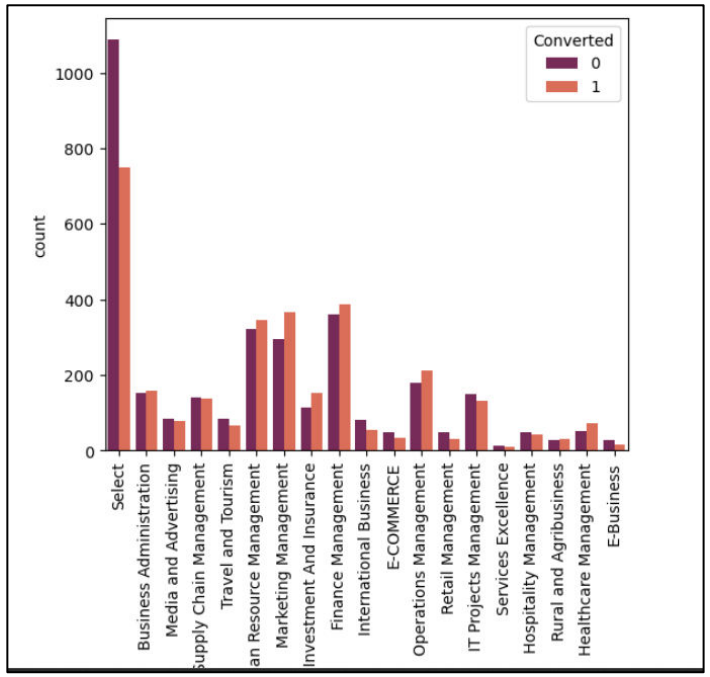
1. Land Page Submission and API are the main aspects which determined the customers to be a lead.
2. Google and Direct Traffic are the main lead sources.
3. Last Activity of most of the customers constitutes Email and SMS Sent.
4. Most of the customers are unemployed and 11% are working professionals.
5. 93% of customers does not want to be emailed about the course.

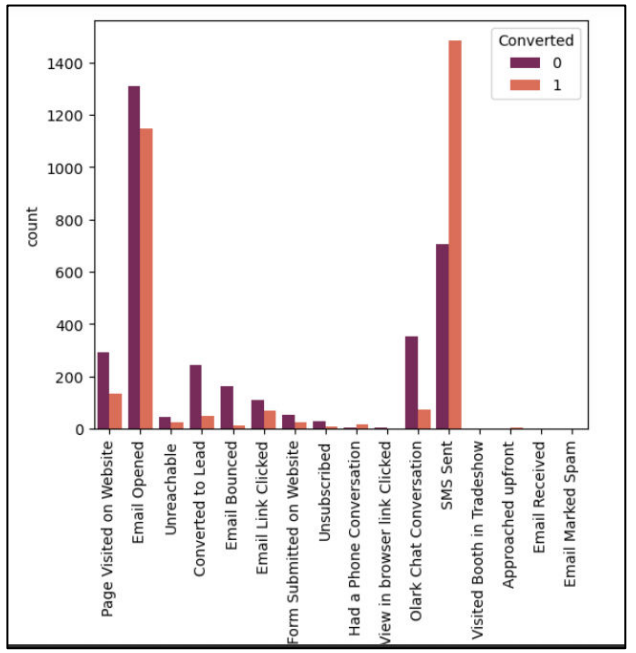
Bivariate Analysis:

1. The leads that got converted are the leads that are identified from Lead Add Form.
2. Most of the converted lead's last activity were sms sent and emails opened.
3. Banking, Finance, HR specialized leads seems to have higher chance of conversion.
4. Students have low chance of conversion when compared to high chances of unemployed and working professionals.









Data preparation:

- Created dummy variables for all categorical variables.
- Converted binary variables such as “Do Not Email” to 0/1.

```
In [47]: X_train.shape
Out[47]: (4461, 73)

In [48]: X_test.shape
Out[48]: (1912, 73)
```

Train-Test Split:

- Splitting the data with train-test size as 70-30.

Feature Scaling:

Scaling the feature "TotalVisits", "Total Time Spent on Website", "Page Views Per Visit” using MinMaxScalar

TotalVisits	Total Time Spent on Website	Page Views Per Visit
0.015936	0.029489	0.125
0.015936	0.082306	0.250
0.023904	0.034331	0.375
0.000000	0.000000	0.000
0.000000	0.000000	0.000

Model Building:

- Using RFE method to select features and running it with 20 variables.
- Assessing the models with Statsmodels.
- Columns containing p – values > 0.05 and VIF > 5 are insignificant. Hence the aim is to drop the columns with the mentioned criteria.

- **Final model:**

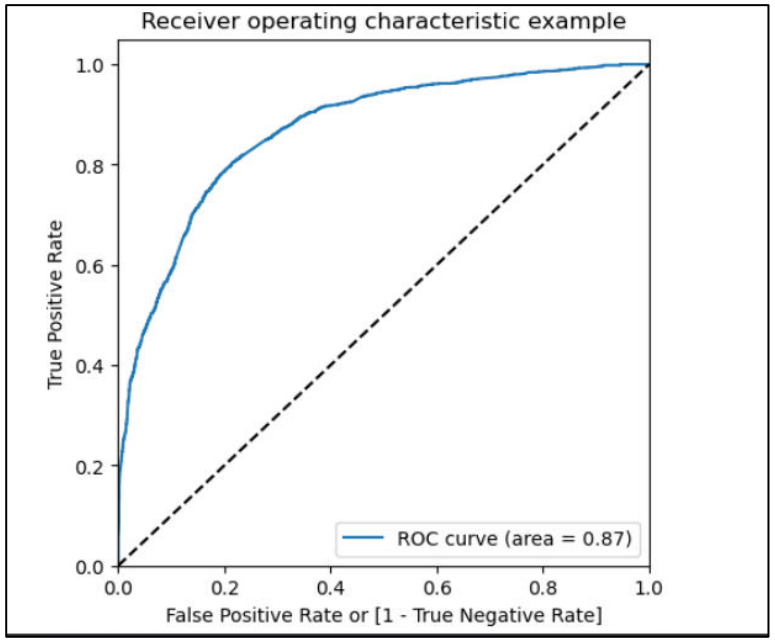
	coef	std err	z	P>|z|	[0.025	0.975]
const	0.5243	0.201	2.609	0.009	0.130	0.918
Do Not Email	-1.4684	0.194	-7.567	0.000	-1.849	-1.088
TotalVisits	8.6504	2.570	3.366	0.001	3.613	13.688
Total Time Spent on Website	4.3937	0.187	23.540	0.000	4.028	4.759
Lead Origin_Lead Add Form	4.1775	0.260	16.068	0.000	3.668	4.687
Lead Source_Olark Chat	1.5459	0.127	12.183	0.000	1.297	1.795
Lead Source_Welingak Website	2.0860	1.037	2.011	0.044	0.053	4.119
Last Activity_Had a Phone Conversation	2.6713	0.801	3.334	0.001	1.101	4.242
Last Activity_Olark Chat Conversation	-0.6211	0.191	-3.249	0.001	-0.996	-0.246
Last Activity_SMS Sent	0.9716	0.085	11.412	0.000	0.805	1.138
What is your current occupation_Student	-2.3685	0.289	-8.203	0.000	-2.934	-1.803
What is your current occupation_Unemployed	-2.5388	0.189	-13.464	0.000	-2.908	-2.169
Last Notable Activity_Modified	-0.7384	0.094	-7.890	0.000	-0.922	-0.555
Last Notable Activity_Unreachable	2.4960	0.808	3.089	0.002	0.912	4.080
Specialization_Banking, Investment And Insurance	0.5640	0.203	2.778	0.005	0.166	0.962

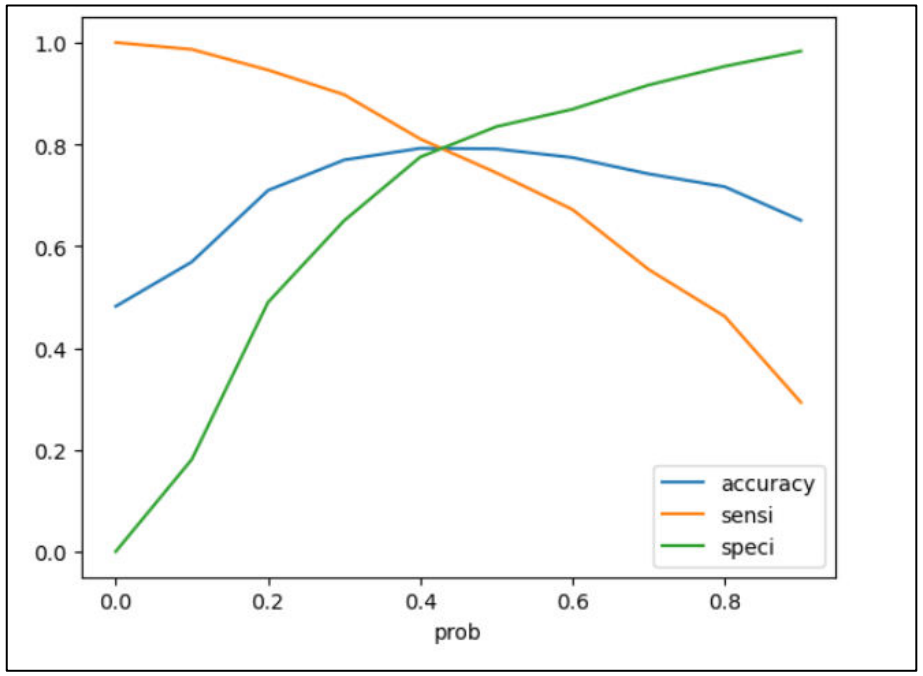
	Features	VIF
10	What is your current occupation_Unemployed	3.37
2	Total Time Spent on Website	2.00
11	Last Notable Activity_Modified	1.61
8	Last Activity_SMS Sent	1.59
1	TotalVisits	1.54
3	Lead Origin_Lead Add Form	1.46
4	Lead Source_Olark Chat	1.43
5	Lead Source_Welingak Website	1.31
7	Last Activity_Olark Chat Conversation	1.29
0	Do Not Email	1.09
9	What is your current occupation_Student	1.09
13	Specialization_Banking, Investment And Insurance	1.05
6	Last Activity_Had a Phone Conversation	1.01
12	Last Notable Activity_Unreachable	1.01

Model Evaluation:

- Model evaluated on the train dataset have an accuracy of 79% with sensitivity and specificity as 74% and 83% respectively.

Plotting ROC Curve and finding optimal cut – off:





The optimal cut – off is found to be 0.42.

Making predictions on test datasets:

```
In [109]: # Let's check the overall accuracy.
          metrics.accuracy_score(y_pred_final.Converted, y_pred_final.final_predicted)

Out[109]: 0.7824267782426778

In [110]: confusion2 = metrics.confusion_matrix(y_pred_final.Converted, y_pred_final.final_predicted )
          confusion2

Out[110]: array([[773, 223],
                 [193, 723]], dtype=int64)

In [111]: 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 [112]: # Let's see the sensitivity of our logistic regression model
          TP / float(TP+FN)

Out[112]: 0.7893013100436681

In [113]: # Let us calculate specificity
          TN / float(TN+FP)

Out[113]: 0.7761044176706827
```

Accuracy for the test dataset is found to be 78%.

Conclusion:

- The Factors that determine the potential leads (Hot leads) i.e. the leads that are most likely to get converted into paying customers are:

1. Based on customer's current occupation: **Unemployed** and **Student**.
2. The **total number of visits** made by the customer on the website.
3. The **total time spent** by the customer on the website.
4. Lead Origin : the origin identifier with which the customer was identified to be a lead was **Lead Add Form**.
5. Lead Source: Source of the lead was **Olark Chat** and **Welingak Website**.
6. Last Activity performed by the customer were either **SMS sent, phone conversation** or **Olark chat conversation**
7. Specialization in **banking, investment and insurance**.