Lead Score Case Study

GROUP MEMBERS:

1. HARSHAVARDHINI J SUJATHA

2. AISHWARYA ANAND

Problem Statement

1. An education company named X Education sells online courses to industry professionals.
2. The typical lead conversion rate at X education is very poor, only around 30%.
3. To make this process more efficient, the company wishes to identify the most potential leads, also known as ‘Hot Leads’.

Business Objective

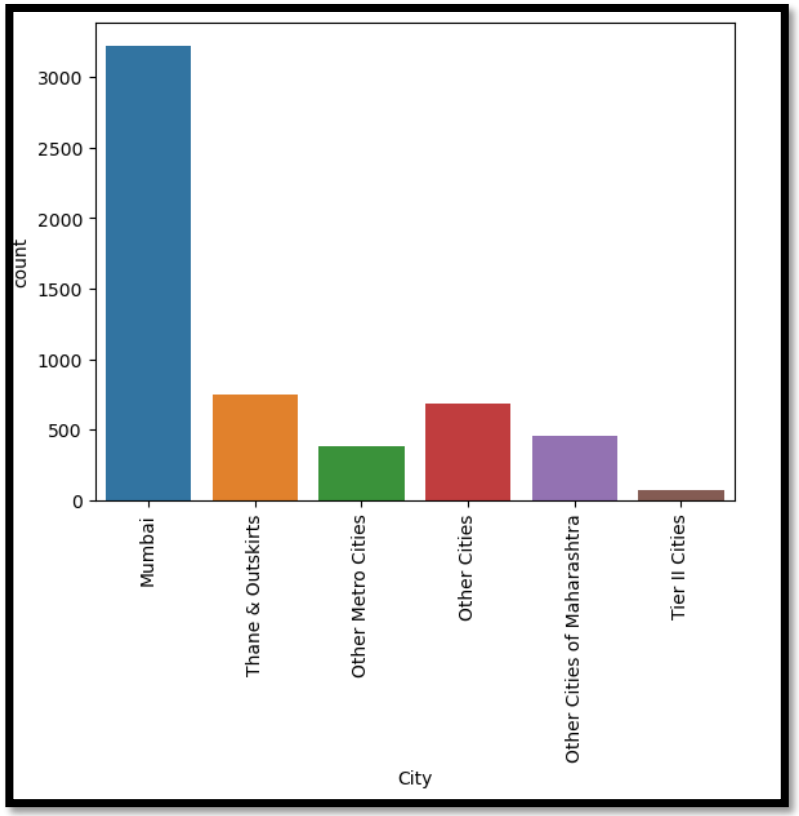
1. X education wants to know most promising leads.
2. Build a model that assign lead score to every lead and thus help in identifying ‘Hot Leads’.
3. Deployment for future use.

Solution Approach

1. Source Data & Data Cleanup:
   - Handling Missing values – Drop columns with high percentage of missing values and high data unbalance.
   - Imputing missing values in Columns.
2. Exploratory Data Analysis:
   - Bivariate plots
   - Creating Dummy variables for Categorical variables
   - Checking for Outliers
3. Splitting Data into Test-Train Datasets.
4. Feature Scaling
5. Model Building
   - Feature Selection using RFE
6. Model Evaluation:
   - Optimal Cutoff point
   - Precision and Recall trade-off
7. Applying the best model in Test data

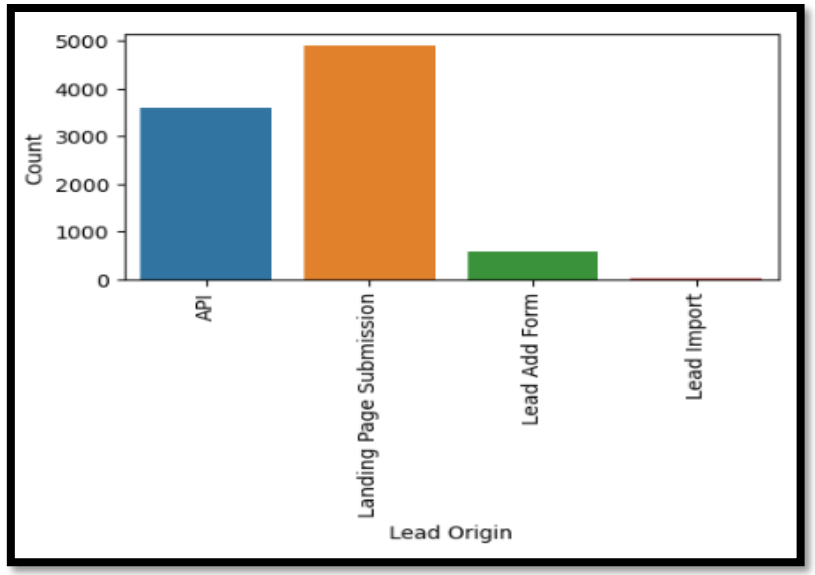
Source Data & Data Cleanup

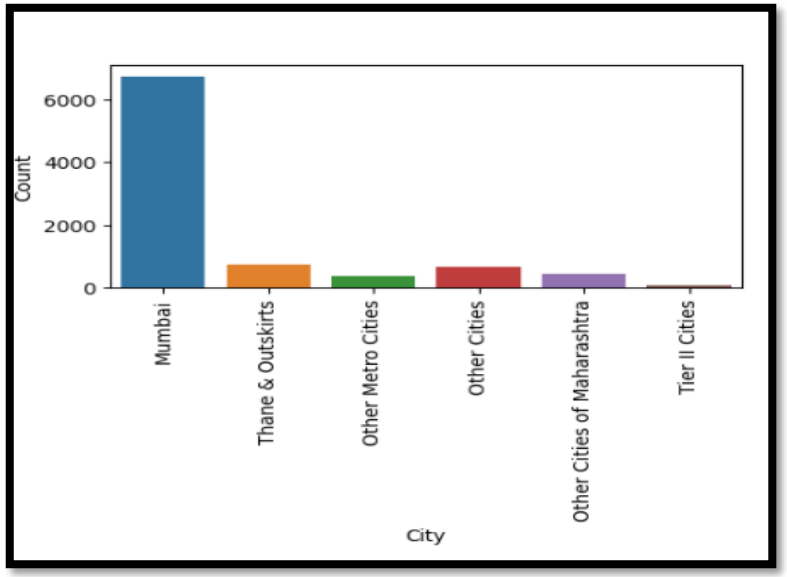
1. Initial data had 37 Columns and 9240 rows.
2. Replace all the columns which has value “Select” with Blank value.
3. Drop all the columns with high percentage of null values (~ 40%).
4. Handle missing values:
   - Impute missing values in the columns.
   - Drop columns with imbalanced data.

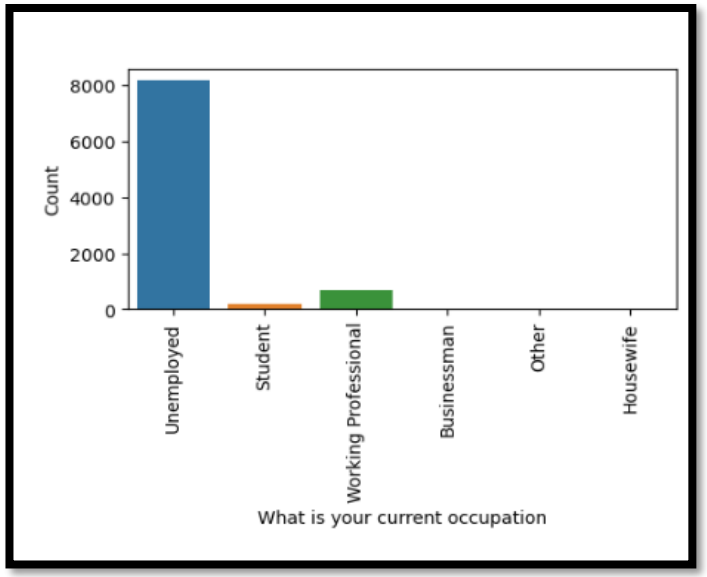


Exploratory Data Analysis

1. Bivariate Analysis: Perform Bivariate analysis of any concurrent relation between any two variables and the target variable i.e. ‘Converted’.
2. Creating Dummy variables for Categorical variables

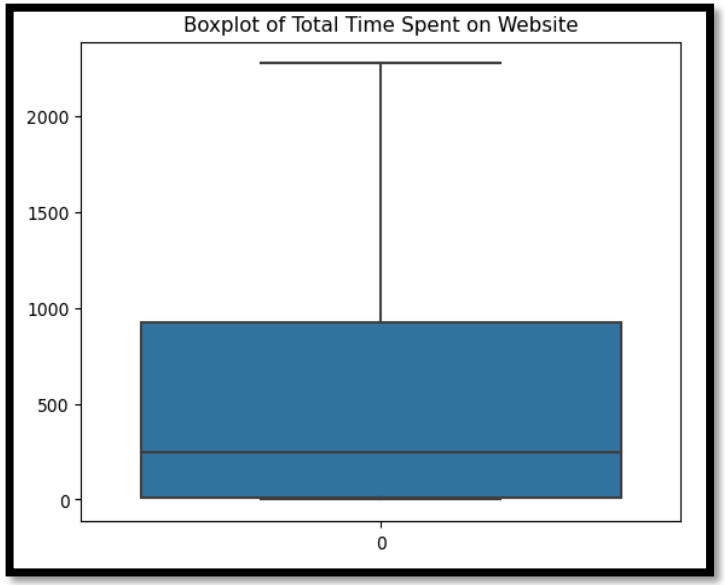


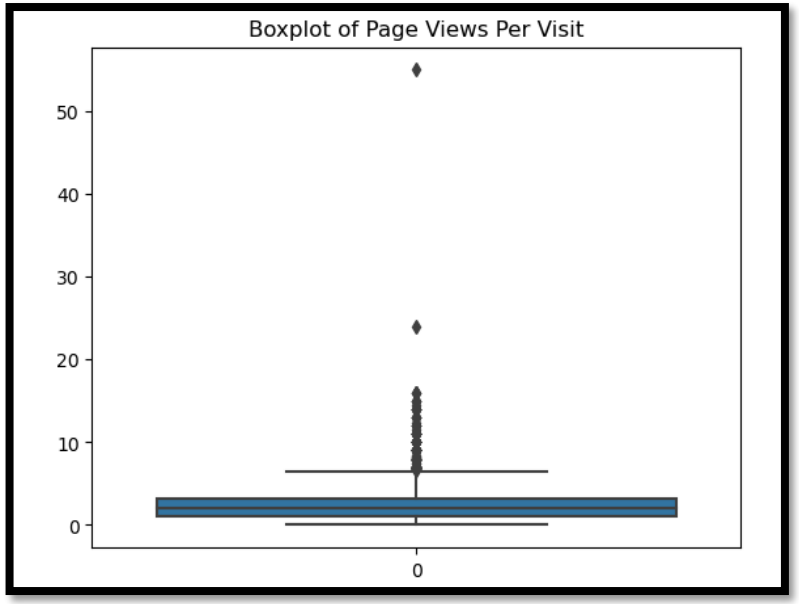


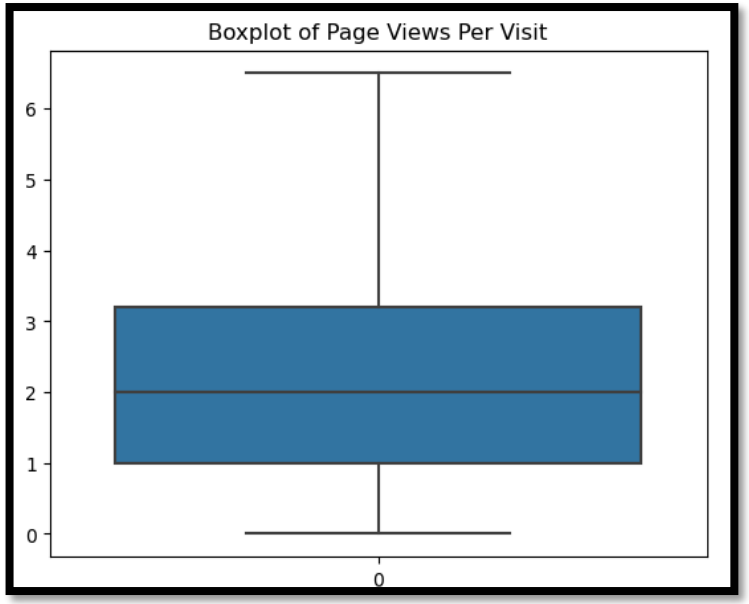


Exploratory Data Analysis (Cont.)

3. Checking for Outliers and removing them.







Splitting Data into Test-Train Datasets

Splitting the data into Test-Train data sets in the ratio of 7:3 to help us build the model on the training set and finally test the model on the test data set.

```
X_train,X_test,y_train,y_test=train_test_split(X,y,train_size=0.7,random_state=100)
print(X_train.shape,y_train.shape)
print(X_test.shape,y_test.shape)
```

```
(6351, 152) (6351,)
(2723, 152) (2723,)
```

Feature Scaling

Normalizing the range of independent variables using the standard scaling method.

```
scaler=StandardScaler()
X_train[['TotalVisits','Total Time Spent on Website','Page Views Per Visit']]=scaler.fit_transform(X_train[['TotalVisits'
X_train.head()
```

	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Lead Origin_Landing Page Submission	Lead Origin_Lead Add Form	Lead Origin_Lead Import	Lead Source_Direct Traffic	Lead Source_Facebook	Lead Source_Google	So
3009	-0.431325	-0.160255	-0.161929	1	0	0	1	0	0	
1012	-0.431325	-0.540048	-0.161929	1	0	0	1	0	0	
9226	-1.124566	-0.888650	-1.247280	0	0	0	0	0	0	
4750	-0.431325	1.643304	-0.161929	1	0	0	1	0	0	
7987	0.608537	2.017593	0.109409	1	0	0	1	0	0	

5 rows × 152 columns

Model Building

1. Run the Logistic Regression model on training dataset.
2. Perform feature selection using RFE.
3. Remove the variables with high p-value and VIF values.
4. Iterate the process until all the process has p-value less than 0.05 and Variance Inflation factor values of all the predictor variables are less than 5 indicating no multicollinearity issues

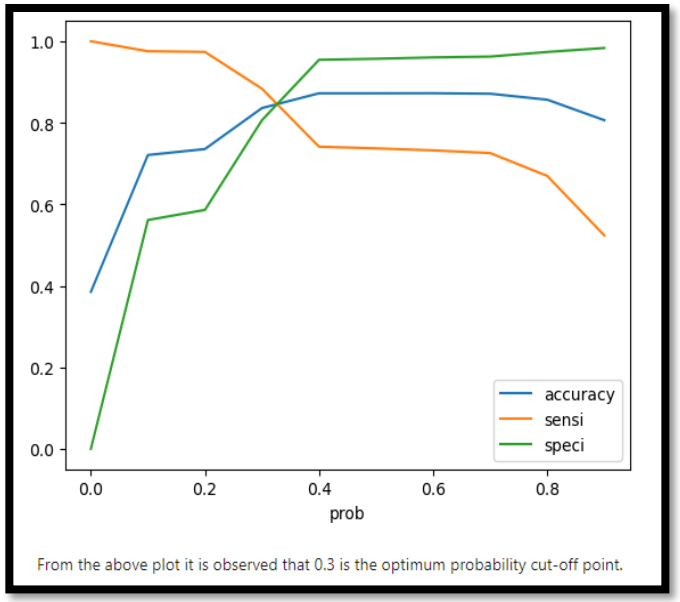
Model Evaluation

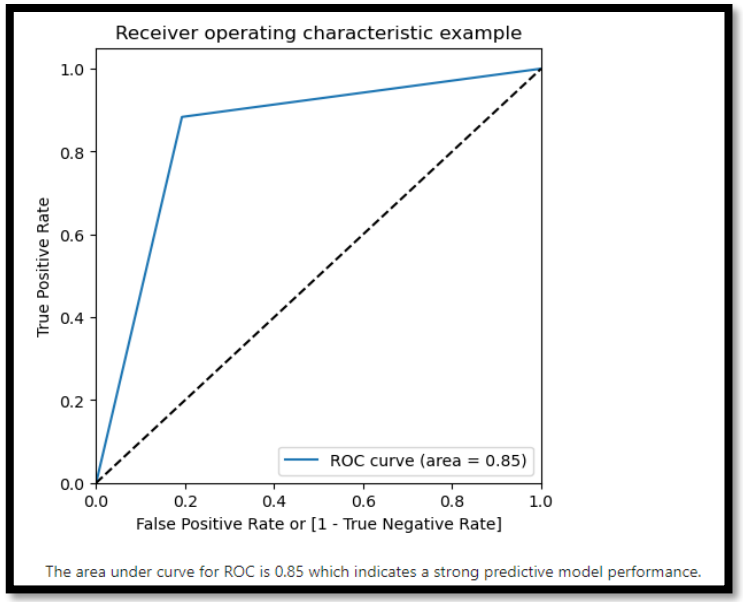
Perform model evaluation using:

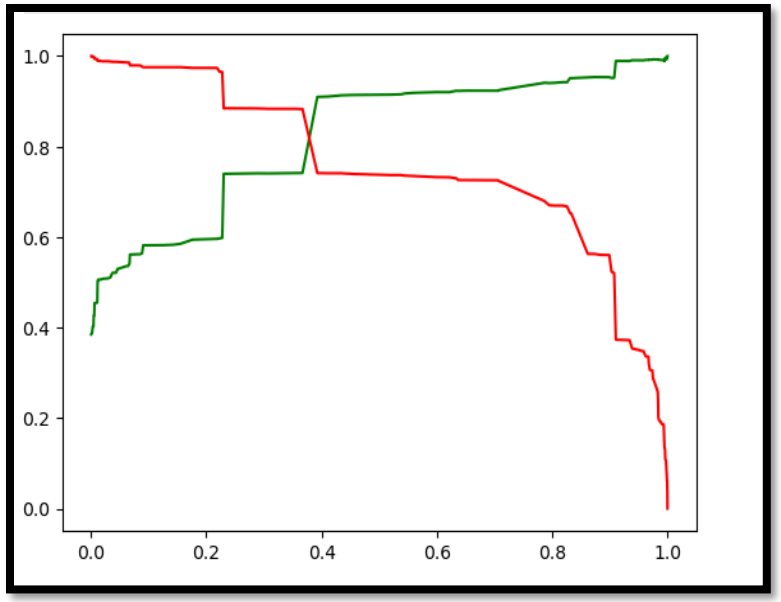
1. Optimal Cutoff Point
2. Precision and Recall Trade-off

```
## Confusion matrix
from sklearn import metrics
confusion_matrix1=metrics.confusion_matrix(y_test_pred_final.Converted,y_test_pred_final.Predicted)
confusion_matrix1
```

```
array([[1386,  348],
       [ 136,  853]], dtype=int64)
```







Model Predictions

Run the model on Test dataset to make predictions.

Results:

- Accuracy = 0.82
- Recall = 0.86
- Precision = 0.71
- Sensitivity of test data = 0.86
- Specificity of test data = 0.799

Conclusion:

- The model performs consistently well on both the train and test datasets, with accuracies around 0.82-0.83, it demonstrates a strong overall predictive capability. Sensitivity and specificity scores above 0.80 reflect the model's ability to effectively identify both positive and negative cases.

- The top three variables in the model that contribute most to the probability of a lead getting converted are **Tags_Closed by Horizzon, Tags_Lost to EINS, and Tags_Will revert after reading the email.**

- These variables indicates varying degrees of readiness and potential for conversion, influencing the overall likelihood of a lead becoming a customer.