Lead Scoring Case Study Presentation

Presented By

Ulagammal Ramalingam

Ritesh

Rishab

Problem Statement:

- X Education aims to identify the most promising leads, referred to as "Hot Leads," that are more likely to convert into paying customers.

- The company seeks to build a predictive model that will assign a lead score to each lead, enabling the sales team to prioritize high-potential leads and focus their efforts on the leads most likely to convert.

Objectives:

- **Build a predictive model** using past lead data to assign a lead score.

- **Improve lead conversion** by focusing on the most promising leads based on the predicted scores.

- Help the **marketing team** strive toward an **80% conversion rate** by using the model to identify the **"Hot Leads"** most likely to convert.

- Ensure the model can **handle future changes** in the company's lead generation process, data, or conversion goals by keeping it **adaptive** and updated.

Goals:

- Build a **logistic regression model** that predicts the likelihood of a lead converting into a customer, with the output being a lead score between 0 and 100.

- A higher score will indicate a higher likelihood of conversion, while a lower score will indicate a lower likelihood.

Steps in Building the Model:

The model can be built by following the steps given below,

1. Data Overview
2. Data Preprocessing
3. Performing Exploratory Data Analysis (EDA)
4. Data Preparation
5. Model Building
6. Model Evaluation
7. Model Performance
8. Business Insights
9. Conclusion

1.Data Overview

In this analysis, we have been provided with one dataset,

Leads.csv :

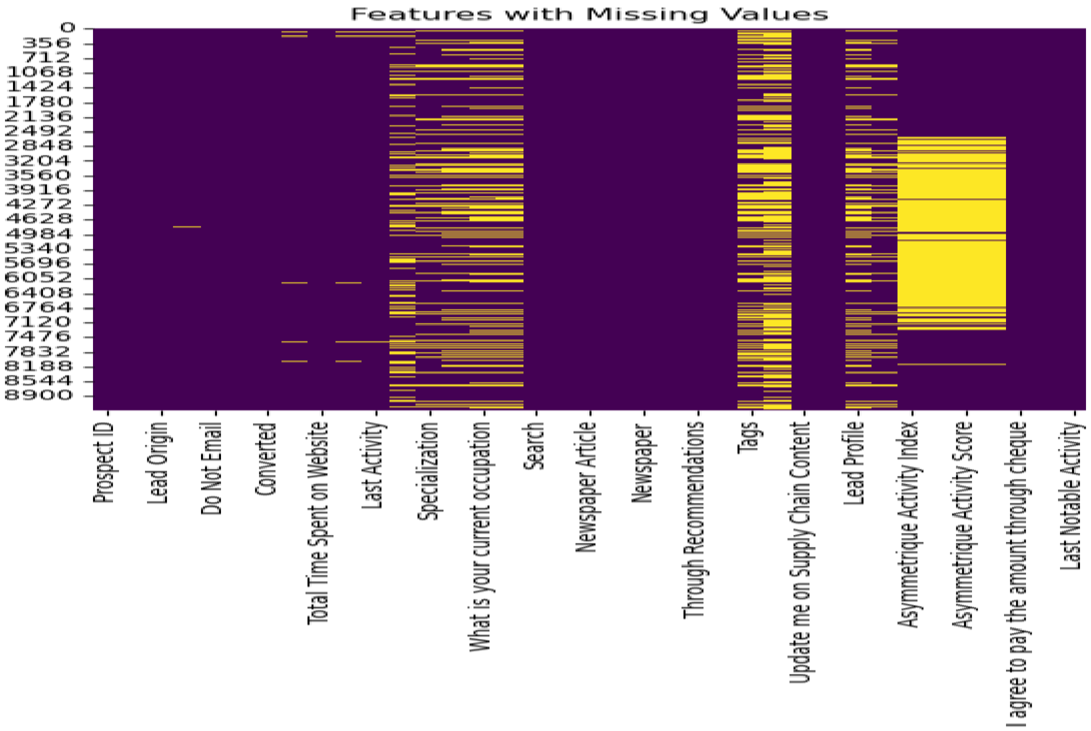
- This dataset contains various attributes such as Lead Source, Total Time Spent on Website, Total Visits, Last Activity, etc.
- It contains 9240 rows and 37 columns
- We have 29 Categorical features and 8 Numerical features

2.Data Preprocessing

- Inspect the Data Frame
- Exploring the Data Frame
- Data Cleaning
- Handling Binary Features
- Handling Categorical Features
  - Handling ‘Select’ level categories
  - Imputation on Categorical features
- Handling Outliers
- Handling Numerical Features

2.1. Inspect the Dataframe

- Drop the unique identifiers columns - ‘Prospect ID', 'Lead Number' as they don’t hold any meaningful values for the model.
- Look into the unique values in the categorical columns
- Check for missing values in the dataset to identify columns that require imputation



2.Data Preprocessing (Contd)

2.2 Data Cleaning

- Handling the 'Select' level in the features
  - ➢ The 'Select' level in categorical variables is a placeholder for when a value has not been selected or provided by the user.
  - ➢ It is as good as a missing value, and this shows the high percentage of missing values in some columns.
  - ➢ Replace 'Select' with NaN and then impute
- Dropping the column 'What matters most to you in choosing a course' as it is extremely dominant.
- Dropping the columns with only one unique value has no variability and wouldn't contribute useful information to the model.

```
Columns with one unique value:
Index(['Magazine', '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'],
      dtype='object')
```

- 'Lead source' and 'Last Activity' – Grouped the categories with a very minimal value under a single category 'Others'.
- Check for the columns having missing values> 40% in the data frame.

```
How did you hear about X Education    78.0
Lead Quality                          52.0
Lead Profile                          74.0
Asymmetrique Activity Index           46.0
Asymmetrique Profile Index            46.0
Asymmetrique Activity Score           46.0
Asymmetrique Profile Score            46.0
dtype: float64
```

- Drop the 'How did you hear about X Education' column with 78% missing value.
- Perform meaningful imputation on the other columns and anlayze the predictive power.

2.3 Handling Binary Features

- Identify all the columns with 'Yes' or 'No' values and encode it as 1 for 'Yes' and 0 for 'No.
- **'Do Not Call', 'Newspaper Article', 'X Education Forums', 'Newspaper', 'Digital Advertisement', 'Through Recommendations',** these columns are mostly dominated by a single value.
- **Consider 'Do Not Call',** this column contains only 2 'yes' which don not add any predictive power to the model.

```
Column: Do Not Call
No      9238
Yes        2
Name: Do Not Call, dtype: int64
```

- These columns have been dominated by a single value, so we can combine the features to represent the same information in a more compact form.
- Grouping the Features:
  - Create a new column to indicate whether the customer came through any marketing channels

```
# Create a new feature to indicate whether the customer came through any marketing channels
df['Came Through Marketing Channels'] = df[['Newspaper Article', 'X Education Forums', 'Newspaper', 'Digital Advertisement', 'Th
```

  - Create a new column flagging customers who opted out of communication

```
# Create a new feature flagging customers who opted out of communication
df['Not Interested in Contact'] = df['Do Not Call'] | df['Do Not Email']
```

2.Data Preprocessing (Contd)

2.4 Handling Categorical Features

Imputation on Categorical Features

- Imputed the missing values using the mode and for the columns with a high percentage of missing values imputed with a specific value 'Unknown' instead of dropping.
- After handling the missing values, we can proceed with encoding the categorical variables.

Grouping the Features

- ‘Country’, ‘Last Notable Activity’, ‘Tags’ have a large number of unique values, so we can group the similar activities.

```
# Group countries with fewer than `threshold` occurrences as "Other"
df['Country_grouped'] = df['Country'].apply(lambda x: x if country_counts[x] >= threshold else 'Other')
```

```
# Group rare tags into an 'Other' category
df['Tags_grouped'] = df['Tags'].apply(lambda x: x if tag_counts[x] >= threshold else 'Other')
```

```
# Map the activities using the defined mapping
df['Last_Notable_Activity_grouped'] = df['Last Notable Activity'].map(activity_mapping).fillna('Other')
```

- This makes the feature easier to interpret and reduces the dimensionality for encoding.

2.5 Handling Numerical Features

- Asymmetrique Activity Score, and Asymmetrique Profile Score have a large number of missing values (around 4215)
  - Checked the correlation of these with the Target variable
  - Still have weak predictive power after imputation, meaning they don't seem to significantly improve the predictive performance of the model.
  - We can drop these features

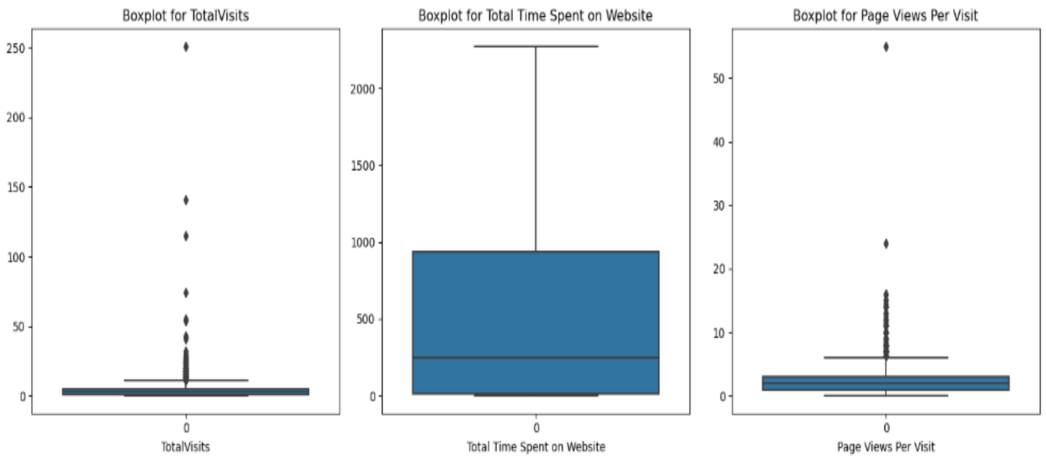
	Asymmetrique Activity Score	Asymmetrique Profile Score	Converted
Asymmetrique Activity Score	1.000000	-0.123250	0.167962
Asymmetrique Profile Score	-0.123250	1.000000	0.218571
Converted	0.167962	0.218571	1.000000

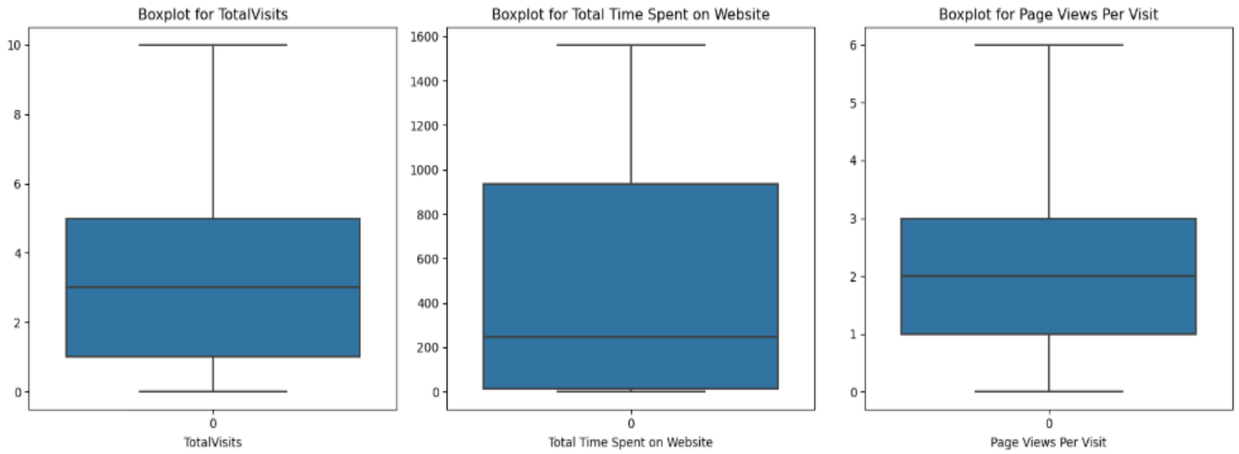
- TotalVisits and Page Views Per Visit, the missing values can be imputed with the median as they are highly skewed.

2. Data Preprocessing(Contd)

- **2.5 Handling Outliers**

  - Check for outliers in the numerical features using BoxPlot.
  - Apply Capping based on percentiles and re-check the features for outliers





3. Performing EDA

3.1.Explore the Target Variable:

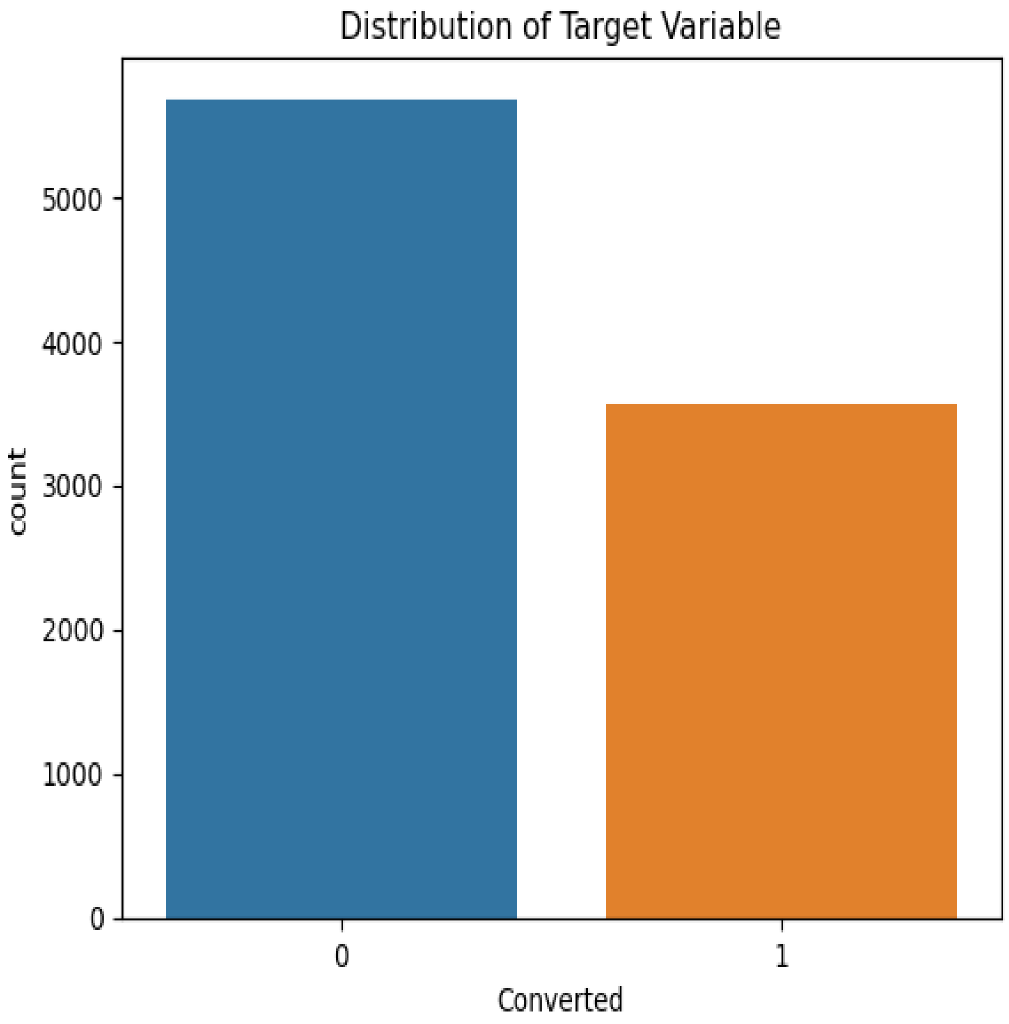
- Check the distribution to see if the data is imbalanced.

- The **"Converted"** column indicates whether a lead was converted (1) or not converted (0).

    - **5679 leads** were **not converted** (0).
    - **3561 leads** were **converted** (1).
    - **38.53 %** is the **Conversion Rate**

- Since the target variable has a higher number of non-converted leads (0) compared to converted leads (1), the dataset is imbalanced.

- To handle the imbalance, we can use **class weights** during training to make the model more sensitive to the minority class.



3. Performing EDA (Contd)

3.2. Univariate Analysis

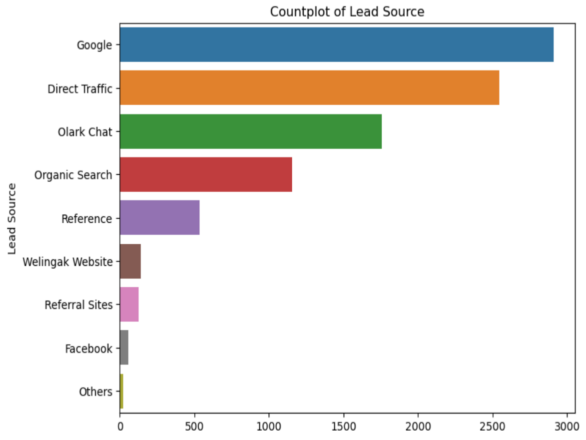
Analysis of Categorical Features

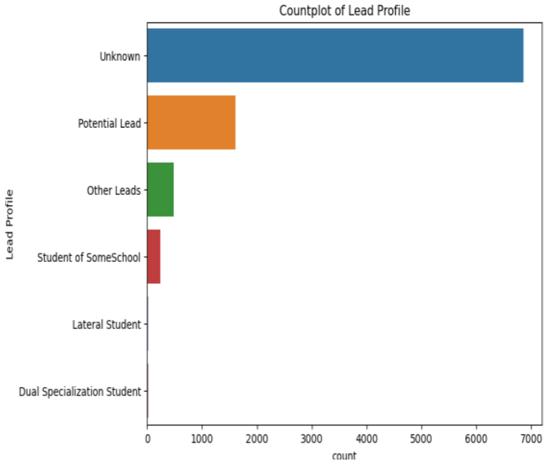
Approach:

- For each categorical column, analyze its distribution with value_counts() to find the frequencies of different categories.
- Visualize the count using count plot/ bar plot

Observations:

•**Dominant Categories:** Many features have one or two dominant categories, indicating concentrated behavior or preferences among leads.
•**Imbalance in Distribution:** Several features show imbalanced distributions, where a few categories represent most of the data. This might impact model performance and require handling techniques.
•**Potential Indicators:** Certain categories may act as strong indicators of lead behavior or conversion likelihood, offering potential predictive power.





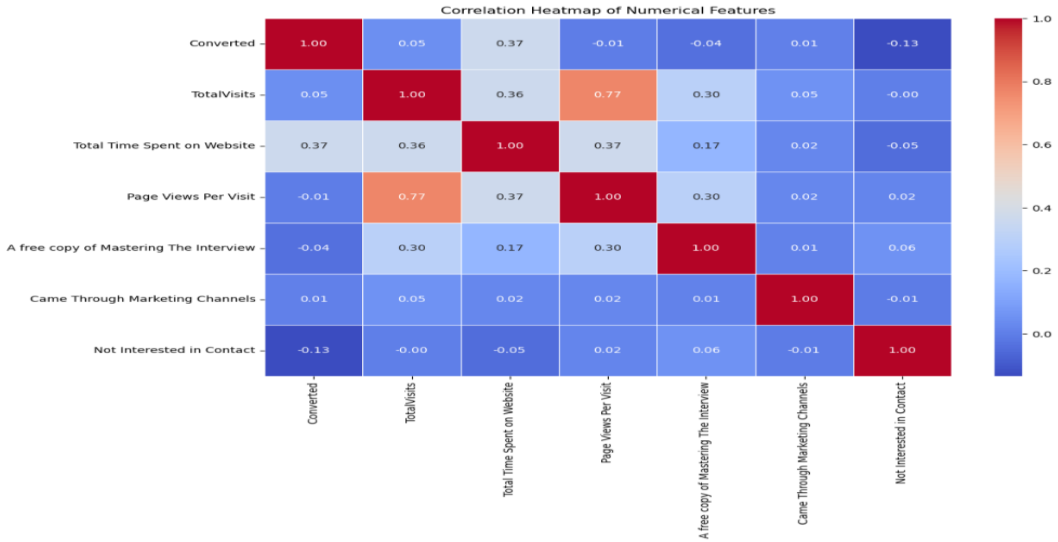
Analysis of Numerical Features

Approach:

- Calculate the summary statistics for all the numerical columns
- Use Pair-plots to understand the distribution of the data.

Observations:

- **Time Spent Matters:** Leads who spend more time on the website are more likely to convert.
- **Engagement Insight:** Visitors who explore more pages during each visit tend to spend more time overall.
- **Weak Predictors:** "A free copy of Mastering The Interview" (-0.04), "Came Through Marketing Channels" (0.01), and "Not Interested in Contact" (-0.13) show little impact on conversion.
- **Negative Indicator:** Leads who aren't interested in contact are less likely to convert.



3. Performing EDA (Contd)

3.2. Segmented Univariate Analysis

This analysis involves breaking down the univariate analysis further by segmenting the data based on the Target variable.

Analysis of Categorical Features

Approach:

- For the categorical columns, perform Chi-square Test and get the p-value
- Visualize the distribution by Target variable(Converted = 1, Not converted = 0) using Bar plot.

Observations:

- **Significant Relationships:** All features show extremely low p-values (close to 0), indicating a strong association with lead conversion.
- **Key Predictors:** Features like *Lead Origin*, *Lead Source*, *Last Activity*, *Specialization*, and *Lead Quality* are particularly significant and could serve as key predictors in the model.

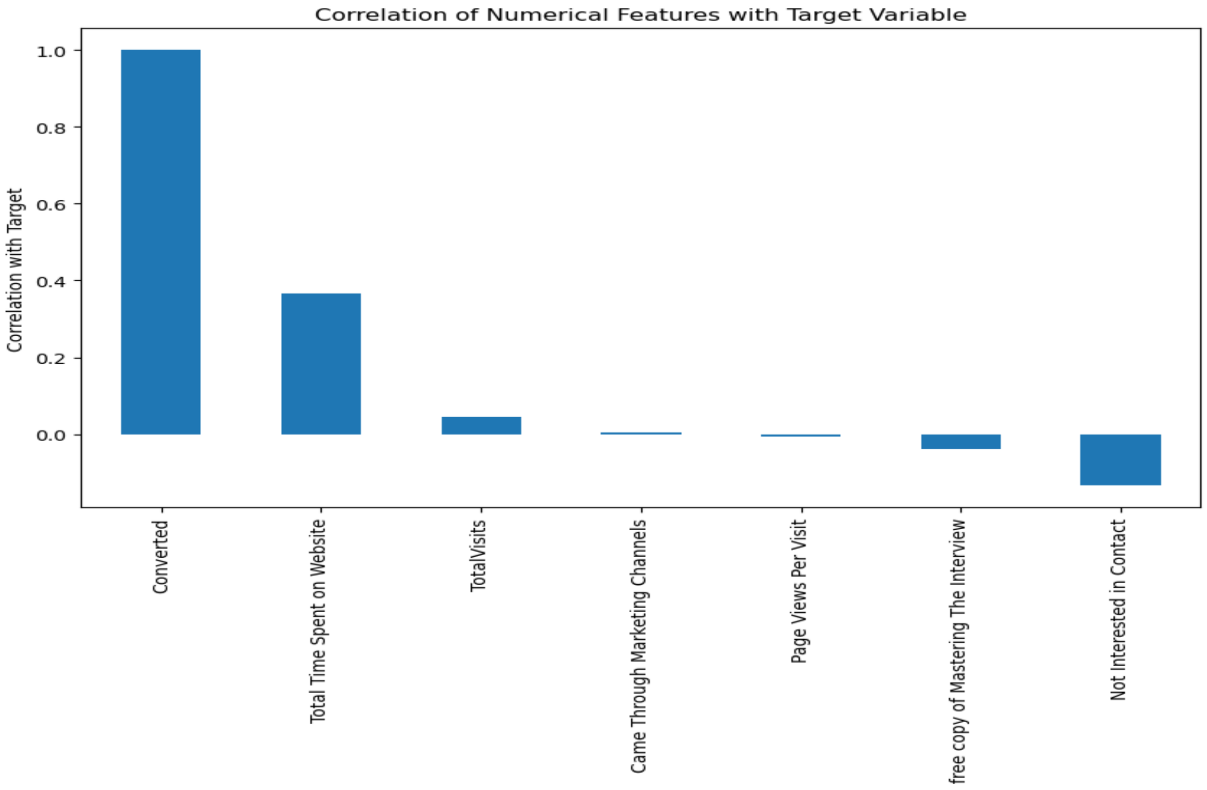
```
Chi-Square Test for Lead Origin: p-value = 1.93831790477878e-211
Chi-Square Test for Lead Source: p-value = 5.807287885111543e-220
Chi-Square Test for Last Activity: p-value = 2.20756149499419e-299
Chi-Square Test for Specialization: p-value = 2.71408416228191e-46
Chi-Square Test for What is your current occupation: p-value = 2.0048634168048997e-
Chi-Square Test for Lead Quality: p-value = 0.0
Chi-Square Test for Lead Profile: p-value = 0.0
Chi-Square Test for City: p-value = 3.858123849309725e-10
Chi-Square Test for Asymmetrique Activity Index: p-value = 1.1014830296533056e-39
Chi-Square Test for Asymmetrique Profile Index: p-value = 5.126283990633163e-33
Chi-Square Test for Country_grouped: p-value = 1.53479763840092e-09
Chi-Square Test for Tags_grouped: p-value = 1.3420084999870018e-132
Chi-Square Test for Last_Notable_Activity_grouped: p-value = 7.692313858756043e-281
```

Analysis of Numerical Features

- Group the data by Target variable: Divide the dataset based on the Target variable
- Compare the distributions and identify the notable differences in the numerical features between the groups.

Observations:

- **Total Time Spent on Website** is the most significant factor positively influencing conversion, while **Not Interested in Contact** is a slight deterrent.
- Other factors like **Total Visits** and **Marketing Channels** have weaker correlations.



4. Data Preparation

4.1. Splitting the Train-Test Data

- We'll split the dataset into 80% for training and 20% for testing.
- It's important to ensure that the data split maintains the class distribution in the target variable ('Converted').
- Use Stratified Sampling to ensure that both the training and testing datasets have a similar distribution of 'Converted' and 'Not Converted' leads.

4.2. Feature Engineering

Approach:

- **Ordinal Columns**: **Ordinal Encoding** is used where each category is mapped to a unique integer that reflects its ranking or position.
  - **Lead Quality:** ['Low in Relevance', 'Might be', 'Not Sure', 'Worst', 'High in Relevance'].This feature has a clear hierarchy or ranking (from worst to best).
  - **Asymmetrique Activity Index:** ['02.Medium', '01.High', '03.Low', 'Unknown'] is ordinal as there is a ranking from low to high.
  - **Asymmetrique Profile Index:** ['02.Medium', '01.High', '03.Low', 'Unknown']

- **Nominal Columns:** As the columns have many unique values , we can perform Target Encoding, which reduces the dimensionality and the multi-collinearity

Approach:

- Apply Target Encoding after splitting the data into training and test sets to ensure no information from the test set leaks into the training process.
- Install 'category-encoders' and then perform target encoding.
- This technique calculates the mean target value for each category and replaces the category with that mean value.
- Binary Features are being excluded in the encoding process.
- This encoding makes the model more stable.

Column	Unique Values
Lead Origin	5
Lead Profile	6
Lead Source	9
Last Activity	11
Country_grouped	5
Specialization	19
What is your current occupation	6
Tags_grouped	5
City	7
Last Notable Activity grouped	8

4. Data Preparation(Contd)

4.3. Feature Scaling

This analysis involves breaking down the univariate analysis further by segmenting the data based on the Target variable.

Approach:

- The dataset contains both continuous (float64) and discrete (int64) features, we want to apply scaling or normalization to the continuous features for better model performance.
- Exclude the Binary features and apply scaling only to the other features
- Use StandardScaler from sklearn.preprocessing to scale the features.

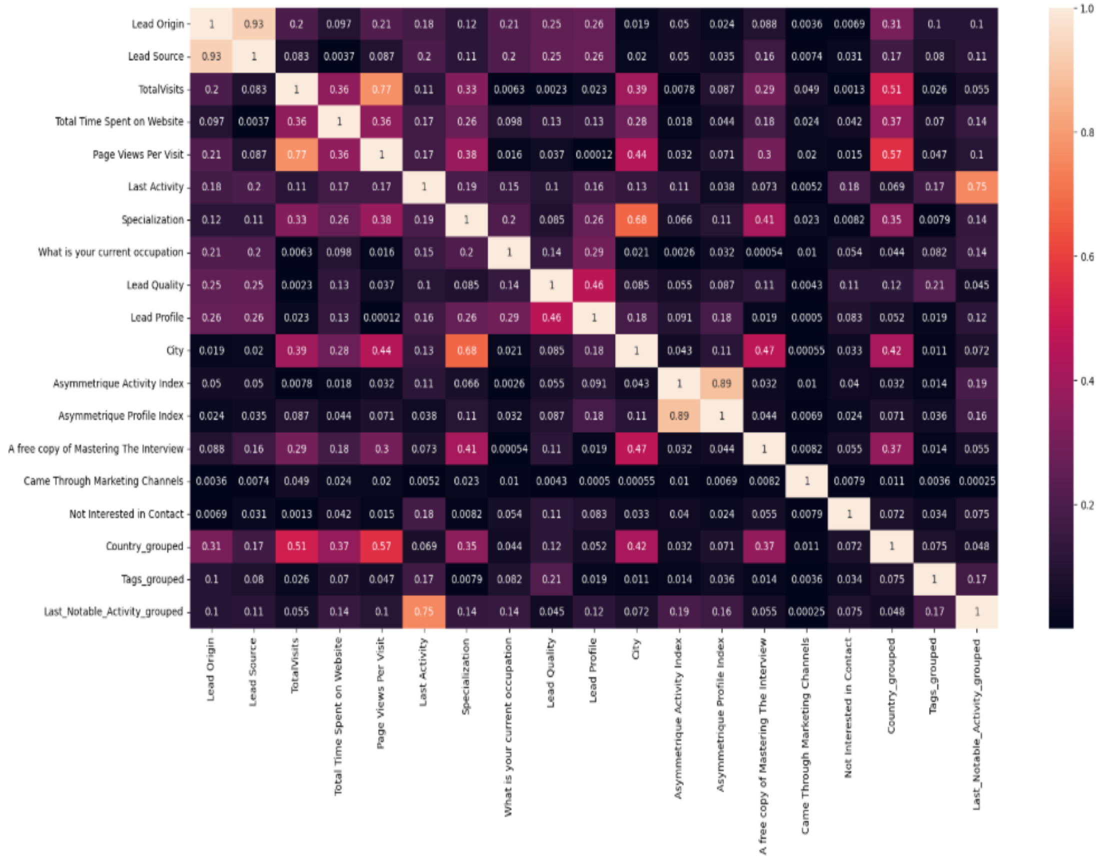
4.4. Correlation Analysis

Check Conversion Rate:

- The Conversion Rate of Leads is 38.53

Approach:

- Features with stronger correlations are likely critical predictors for scoring leads.
- Many features appear to have low correlation, that might not have strong multicollinearity.



5. Model Building

5.1 Model Selection

- The data is prepared and preprocessed, the next step is to build the predictive model using logistic regression.
- Logistic Regression, a binary classification algorithm suitable for predicting the likelihood of a lead converting (converted = 1) or not converting (converted = 0).
- The model outputs a probability score between 0 and 1, which can be scaled to a 0-100 lead score by multiplying the probability by 100.

Generalized Linear Model Regression Results

Dep. Variable:	Converted	No. Observations:	7392
Model:	GLM	Df Residuals:	7372
Model Family:	Binomial	Df Model:	19
Link Function:	Logit	Scale:	1.0000
Method:	IRLS	Log-Likelihood:	-2388.7
Date:	Fri, 14 Mar 2025	Deviance:	4777.4
Time:	19:33:35	Pearson chi2:	9.07e+03
No. Iterations:	7	Pseudo R-squ. (CS):	0.4969
Covariance Type:	nonrobust		

5. Model Building(Contd)

5.2 Model Training

5.2.1 Fit the Model

Approach:

- As we are dealing with the imbalanced classes (more non-converted leads than converted leads), we use class weights to penalize misclassification of the minority class more heavily.
- Get the statistical summary using 'statsmodel'
- It will provide insights into the coefficients and statistical significance of each feature.

Observations from the Model:

Key Positive Drivers:

- **Total Time Spent on Website:** The strongest predictor — more time spent increases conversion chances.
- **Tags, Occupation, Lead Profile, Last Activity, and Lead Origin:** These factors significantly boost conversion likelihood, making them crucial for targeting high-potential leads.

Takeaway:

- Focus on increasing engagement and time spent on the website while improving lead quality.

	coef	std err	z	P>|z|	[0.025	0.975]
const	-0.6215	0.054	-11.549	0.000	-0.727	-0.516
Lead Origin	0.4792	0.119	4.021	0.000	0.246	0.713
Lead Source	0.1666	0.114	1.457	0.145	-0.057	0.391
TotalVisits	0.3046	0.057	5.306	0.000	0.192	0.417
Total Time Spent on Website	1.1138	0.043	25.895	0.000	1.029	1.198
Page Views Per Visit	-0.4687	0.063	-7.438	0.000	-0.592	-0.345
Last Activity	0.5263	0.060	8.817	0.000	0.409	0.643
Specialization	0.2283	0.058	3.960	0.000	0.115	0.341
What is your current occupation	0.6366	0.059	10.713	0.000	0.520	0.753
Lead Quality	-1.0941	0.061	-17.894	0.000	-1.214	-0.974
Lead Profile	0.5572	0.049	11.464	0.000	0.462	0.652
City	-0.1562	0.059	-2.660	0.008	-0.271	-0.041
Asymmetrique Activity Index	-0.4747	0.091	-5.225	0.000	-0.653	-0.297
Asymmetrique Profile Index	0.4564	0.094	4.867	0.000	0.273	0.640
A free copy of Mastering The Interview	-0.1974	0.099	-1.995	0.046	-0.391	-0.003
Came Through Marketing Channels	-0.0360	0.045	-0.801	0.423	-0.124	0.052
Not Interested in Contact	-0.9084	0.173	-5.261	0.000	-1.247	-0.570
Country_grouped	0.3391	0.059	5.711	0.000	0.223	0.456
Tags_grouped	0.7196	0.060	11.974	0.000	0.602	0.837
Last_Notable_Activity_grouped	0.5804	0.054	10.705	0.000	0.474	0.687

5. Model Building(Contd)

5.2.2 Feature Selection Using RFE

Approach:

- Select the top 15 features using RFE
- Focus on maintaining a balance between simplicity and predictive power.

```
Index(['Lead Origin', 'TotalVisits', 'Total Time Spent on Website',
       'Page Views Per Visit', 'Last Activity',
       'What is your current occupation', 'Lead Quality', 'Lead Profile',
       'Asymmetrique Activity Index', 'Asymmetrique Profile Index',
       'A free copy of Mastering The Interview', 'Not Interested in Contact',
       'Country_grouped', 'Tags_grouped', 'Last_Notable_Activity_grouped'],
      dtype='object')
```

5.2.3 ReTrain the Model

Approach:

- Fit the model with the selected features and get the statistics summary
- **Model Performance**:
  - The **Pseudo R-squared (0.4957)** indicates that the model explains around **49.57%** of the variance in lead conversion — moderate to strong for a logistic regression model.
- **Strong Positive Predictors**:
  - Total Time Spent on Website (1.1224, p=0.000)
  - Lead Origin (0.6483, p=0.000)
  - Last Activity (0.5226, p=0.000)
  - Lead Profile (0.5619, p=0.000)
  - Tags_grouped (0.7197, p=0.000)
  - Country_grouped (0.2987, p=0.000)
- **Strong Negative Predictors:**
  - Lead Quality (-1.0837, p=0.000)
  - Not Interested in Contact (-0.9324, p=0.000)
  - Page Views Per Visit (-0.4457, p=0.000)
  - Asymmetrique Activity Index (-0.4354, p=0.000)
  - A free copy of Mastering The Interview (-0.2154, p=0.012)

Observations:

- The Pseudo R-squared (0.4957) indicates that the model explains around 50% of the variance in lead conversion — moderate to strong for a logistic regression model.
- This is our final model with 14 features.

5. Model Building(Contd)

5.2.4 Make Predictions
Approach:

- Use the predict() method to get predicted probabilities for each lead
- Combine the actual conversion values and predicted probabilities into a new data frame.
- We can compare the actual values (Actual) and predicted class values (Predicted_Class) to evaluate how well the model is performing.

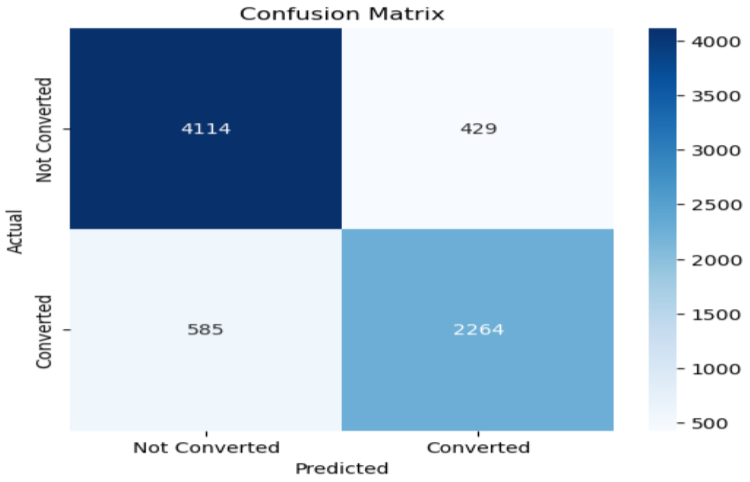
5.2.5. Actual Vs Predicted
Approach:

- Convert the predicted probabilities to binary class predictions (0 or 1) by setting a threshold.
- A common threshold is 0.5, where values greater than or equal to 0.5 are classified as 1 (converted), and those less than 0.5 are classified as 0 (not converted). This helps in decision-making.
  - Predicted probability >= 0.5 → Classified as 1 (Converted)
  - Predicted probability < 0.5 → Classified as 0 (Not Converted)

	Converted	Actual_Prob	LeadID	Predicted
0	1	0.066754	9067	0
1	0	0.281654	6093	0
2	1	0.544951	855	1
3	0	0.119310	6053	0
4	0	0.097666	292	0

5.2.6. Assess Using Confusion Matrix
Approach:

The confusion matrix will help you understand how well your model is performing for both classes.



- True Negative (TN) = 4114: These are the leads that were predicted as not converted (0) and actually not converted.
- False Positive (FP) = 429: These are the leads that were predicted as converted (1) but actually not converted.
- False Negative (FN) = 585: These are the leads that were predicted as not converted (0) but actually converted.
- True Positive (TP) = 2264: These are the leads that were predicted as converted (1) and actually converted.

Observations:

- By checking the accuracy, about 86.28% of the predictions made by the model (whether a lead would convert or not) are correct, which is a strong performance.

5. Model Building(Contd)

5.2.7 Checking VIF

Approach:

- VIF values help assess the multicollinearity between the predictor variables.
- '**Asymmetrique Profile Index 5.877003'** and **'Asymmetrique Activity Index 5.724097**' have VIF values > 5, so we can try dropping one of them and re-run the model.

	Feature	VIF
10	Asymmetrique Profile Index	5.877003
9	Asymmetrique Activity Index	5.724097
4	Page Views Per Visit	2.848453
5	Last Activity	2.608859
2	TotalVisits	2.590945
15	Last_Notable_Activity_grouped	2.455954
13	Country_grouped	1.813867
0	const	1.624957
8	Lead Profile	1.476065
7	Lead Quality	1.413701
1	Lead Origin	1.337365
3	Total Time Spent on Website	1.304747
11	A free copy of Mastering The Interview	1.201276
6	What is your current occupation	1.136500
14	Tags_grouped	1.125197
12	Not Interested in Contact	1.071385

5.2.6. Retrain the Model After Dropping

Approach:

- Fit the model after removing the 'Asymmetrique Profile Index' and re-run the model.
- Print the statistics
- Make Predictions
- Assess using confusion matrix
- The accuracy of this model is 86.01%
- Check the VIF values again

	Feature	VIF
4	Page Views Per Visit	2.828292
2	TotalVisits	2.590371
5	Last Activity	2.564351
14	Last_Notable_Activity_grouped	2.422718
12	Country_grouped	1.776812
0	const	1.620330
8	Lead Profile	1.439905
7	Lead Quality	1.413637
3	Total Time Spent on Website	1.304576
1	Lead Origin	1.296124
10	A free copy of Mastering The Interview	1.194703
6	What is your current occupation	1.136402
13	Tags_grouped	1.110095
11	Not Interested in Contact	1.069344
9	Asymmetrique Activity Index	1.064312

6. Model Evaluation

6.1 Metrics Beyond Accuracy

- These metrics provide more insight into how well your model handles the classes.

```
F1-Score: 0.8132
Sensitivity: 0.7898
Specificity: 0.9042
False Positive Rate: 0.0958
Positive Predictive Value: 0.8380
Negative Predictive value: 0.8727
```

Observations:

- **F1-Score (0.8132)**: A strong balance between precision and recall, meaning the model performs well in predicting conversions while minimizing false positives and false negatives.
- **Sensitivity (Recall) (0.7898)**: The model correctly identifies 79% of actual conversions, this cut-off point had to be optimized to get a decent value of sensitivity and for this, we will use the ROC curve.
- **Specificity (0.9042)**: The model correctly classifies 90% of non-conversions, indicating strong performance in avoiding false positives.
- **False Positive Rate (0.0958)**: A low false positive rate, meaning the model rarely misclassifies non-conversions as conversions.
- **Positive Predictive Value (Precision) (0.8380)**: 83.8% of predicted conversions are actual conversions, ensuring reliability in high-confidence predictions.
- **Negative Predictive Value (0.8727)**: 87.3% of predicted non-conversions are truly not converted, which is strong for identifying not converted

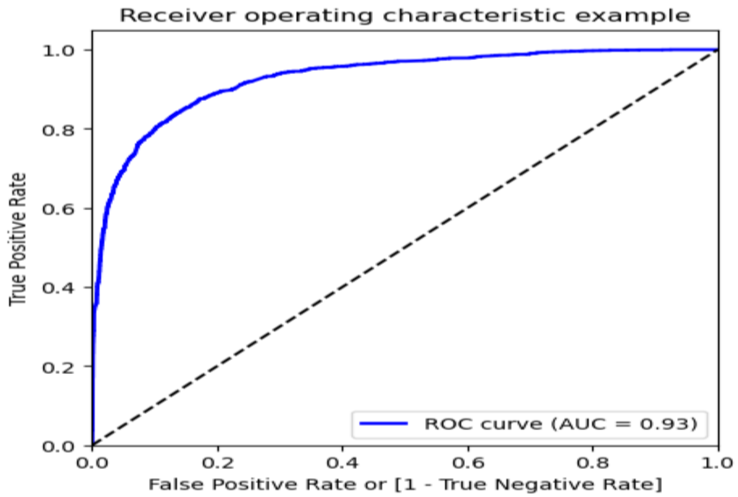
6.2. Plotting ROC Curve

Approach:

- It shows the tradeoff between sensitivity and specificity (any increase in sensitivity will be accompanied by a decrease in specificity).
- The closer the curve follows the left-hand border and then the top border of the ROC space, the more accurate the test.
- The closer the curve comes to the 45-degree diagonal of the ROC space, the less accurate the test.

Observations:

- **AUC (Area Under the Curve) = 0.93**
- This indicates an excellent model. AUC values close to 1.0 suggest a high discriminative power between converted and non-converted cases.
- 0.93 means the model correctly ranks positive instances higher than negative ones 93% of the time.

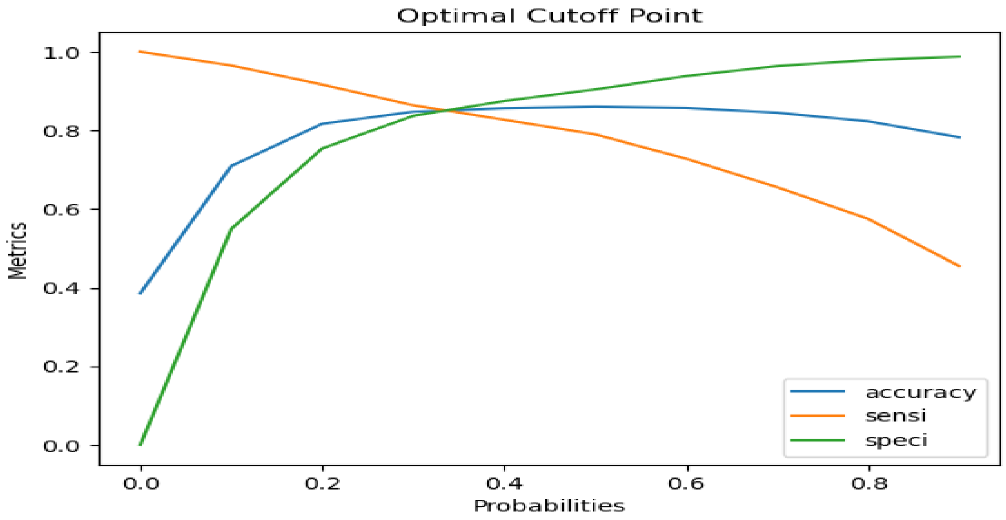


6. Model Evaluation(Contd)

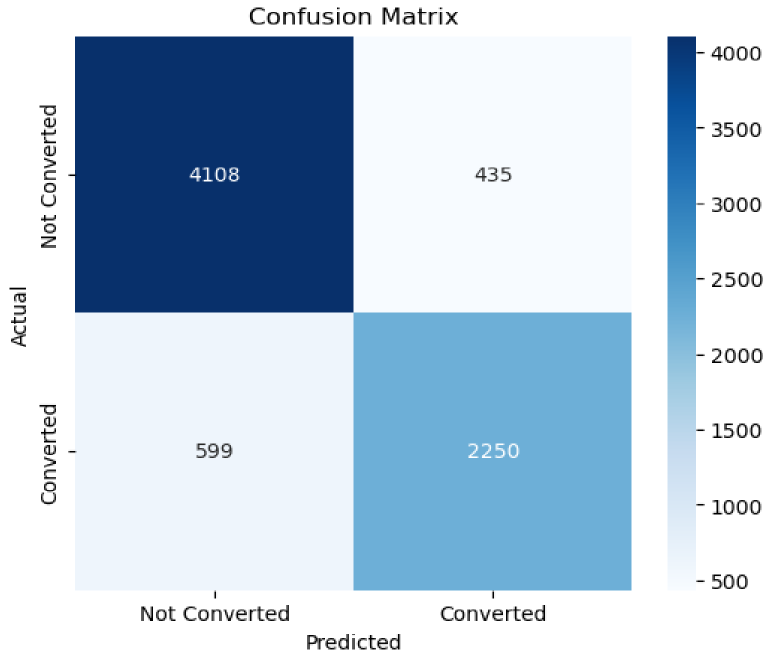
6.3 Finding Optimal Cutoff Point

Approach:

- Create columns with different probability cutoffs and calculate accuracy, sensitivity, and specificity.
- By plotting the accuracy, sensitivity, and specificity for various probabilities, we can calculate the Optimal Cutoff Point.



- The Optimal Probability Cutoff is 0.4 which can be calculated using 'Youden_index' method
- Change the threshold to 0.4 and make predictions
- Assess the model again using Confusion Matrix



- The overall accuracy is 86.01% after changing the threshold
- Calculate the Metrics Beyond accuracy

```
F1-Score: 0.8132
Sensitivity: 0.7898
Specificity: 0.9042
False Positive Rate: 0.0958
Positive Predictive Value: 0.8380
Negative Predictive value: 0.8727
```

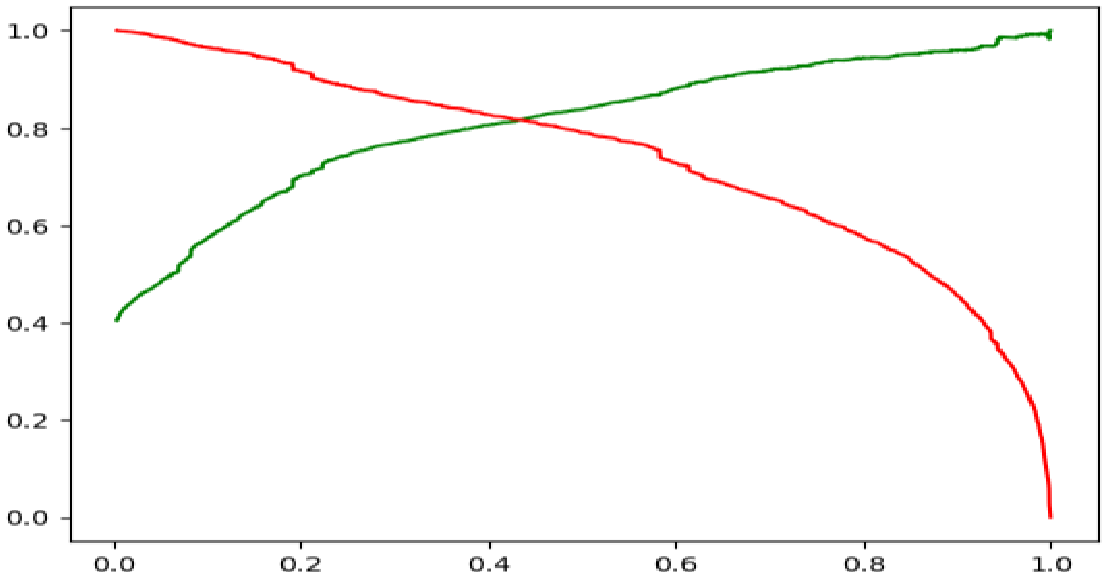
- Precision is 83.80%, meaning that out of all the leads predicted as "converted," about 83.8% actually converted.
- Recall is 78.98%, meaning that out of all actual conversions, 78.98% were correctly identified by the model.

6. Model Evaluation(Contd)

6.4 Precision And Recall Tradeoff

Approach:

- By plotting the accuracy, sensitivity, and specificity for various probabilities, we can find the Optimal Cutoff Point.



- The ideal threshold is where precision and recall balance well.

6.5 Making Predictions on the Test Set

Approach:

- Make probability predictions on the test set.
- Convert probabilities into binary class predictions using the chosen threshold.
- Evaluate the model on test data.

	Converted	LeadID	Actual_Prob	final_predicted
0	1	683	0.427488	1
1	1	1931	0.256870	0
2	0	6950	0.224922	0
3	0	2996	0.019259	0
4	0	3902	0.024198	0

6. Model Evaluation(Contd)

6.6 Model Evaluation on the Test Set

Observations:

- The overall accuracy of 83.71% on the test set is a strong result.
- The confusion matrix looks good,
  - True Negatives (TN): 969 leads correctly predicted as "Not Converted"
  - False Positives (FP): 167 leads incorrectly predicted as "Converted"
  - False Negatives (FN): 134 leads incorrectly predicted as "Not Converted"
  - True Positives (TP): 578 leads correctly predicted as "Converted"
- Calculate the metrics,
  - F1-Score (0.7934): Indicates a strong balance between precision and recall.
  - Sensitivity (0.7898): The model correctly identifies ~79% of converted leads.
  - Specificity (0.9042): The model correctly identifies ~90% of non-converted leads.
  - False Positive Rate (0.0958): Only ~9.6% of non-converted leads are incorrectly labeled as converted.
  - Positive Predictive Value (0.8380): 83.8% of leads predicted as converted actually convert.
  - Negative Predictive Value (0.8727): 87.3% of leads predicted as not converted remain unconverted.

6.7. Lead Score

Approach:

- Calculated the *Lead Score* by scaling the predicted probability from the model:

  Lead Score=Predicted Probability×100

- Classified leads into categories based on their scores:
  - **Hot Leads**: Leads with a score greater than 85, indicating a high likelihood of conversion.
- This allows the sales team to prioritize these high-scoring leads, improving efficiency and increasing the chances of conversion.

	Converted	LeadID	Actual_Prob	final_predicted	Lead_Score
10	1	3685	0.997415	1	100
13	1	9158	0.962401	1	96
22	1	6461	0.998290	1	100
23	1	6475	0.985098	1	99
34	0	745	0.943670	1	94

Result:

- The model successfully pinpoints the most promising leads, enabling a targeted sales strategy.

6. Model Evaluation(Contd)

6.6 Model Evaluation on the Test Set

Observations:

- The overall accuracy of 83.71% on the test set is a strong result.
- The confusion matrix looks good,
  - True Negatives (TN): 969 leads correctly predicted as "Not Converted"
  - False Positives (FP): 167 leads incorrectly predicted as "Converted"
  - False Negatives (FN): 134 leads incorrectly predicted as "Not Converted"
  - True Positives (TP): 578 leads correctly predicted as "Converted"
- Calculate the metrics,
  - F1-Score (0.7934): Indicates a strong balance between precision and recall.
  - Sensitivity (0.7898): The model correctly identifies ~79% of converted leads.
  - Specificity (0.9042): The model correctly identifies ~90% of non-converted leads.
  - False Positive Rate (0.0958): Only ~9.6% of non-converted leads are incorrectly labeled as converted.
  - Positive Predictive Value (0.8380): 83.8% of leads predicted as converted actually convert.
  - Negative Predictive Value (0.8727): 87.3% of leads predicted as not converted remain unconverted.

6.7. Lead Score

Approach:

- Calculated the *Lead Score* by scaling the predicted probability from the model:

  Lead Score=Predicted Probability×100

- Classified leads into categories based on their scores:
  - **Hot Leads**: Leads with a score greater than 85, indicating a high likelihood of conversion.
- This allows the sales team to prioritize these high-scoring leads, improving efficiency and increasing the chances of conversion.

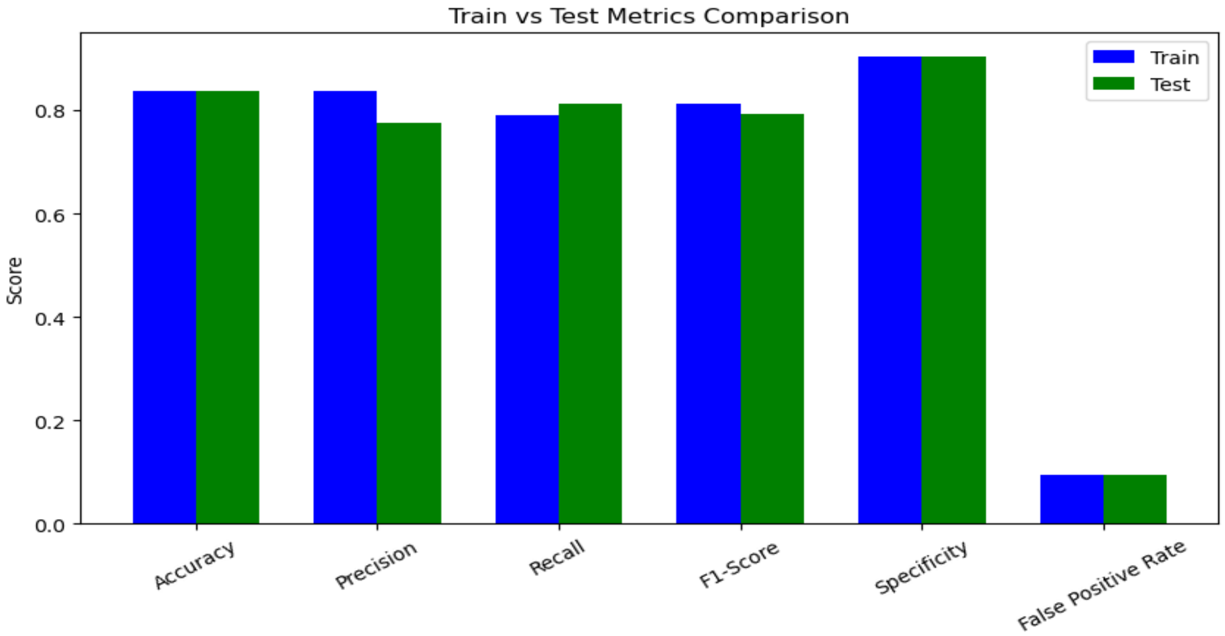
	Converted	LeadID	Actual_Prob	final_predicted	Lead_Score
10	1	3685	0.997415	1	100
13	1	9158	0.962401	1	96
22	1	6461	0.998290	1	100
23	1	6475	0.985098	1	99
34	0	745	0.943670	1	94

Result:

- The model successfully pinpoints the most promising leads, enabling a targeted sales strategy.

7. Model Performance

- Consistent Accuracy – The accuracy on the test set closely matches the train set, indicating strong generalization and minimal overfitting.
- High Specificity – The model effectively identifies non-converting leads, ensuring minimal wasted effort on low-potential leads.
- Balanced Recall – The recall remains stable between train and test, meaning the model consistently captures potential conversions.
- Strong F1-Score – A good balance between precision and recall ensures reliable lead prioritization.
- Low False Positive Rate – The model minimizes incorrect classifications, reducing unnecessary sales efforts on unlikely conversions.



Metric	Train Set	Test Set
Accuracy	83.71%	83.71%
Precision	83.80%	77.58%
Recall	78.98%	81.18%
F1-Score	81.32%	79.34%
Specificity	90.42%	90.42%
False Positive Rate	9.58%	9.58%

7. Business Insights and Recommendations

Business Problem: Aggressive Lead Conversion During the Intern Hiring Phase

Strategy:

Lower the Lead Conversion Threshold:

- ✓ Adjust the threshold from 0.5 to 0.4 to increase the pool of predicted leads.
- ✓ This allows reaching out to more leads with a reasonable chance of conversion.

Prioritized Outreach Based on Lead Scores:

- ✓ **High Priority** (Lead Score > 80): Focus efforts on leads with the highest likelihood of conversion.
- ✓ **Medium Priority** (Score 60-80): Assign interns to follow up with these leads.
- ✓ **Low Priority** (Score 40-60): Engage these leads with a combination of automated nurturing and minimal outreach.

Multi-Channel Engagement:

- ✓ Use **emails, WhatsApp, and SMS** for outreach to all lead categories, with phone calls reserved for high-priority leads.
- ✓ Automate follow-up sequences to maintain engagement over time.

Business Problem: Minimizing Calls When Quarterly Targets Are Met Early

Strategy:

Increase the Lead Conversion Threshold:

- ✓ Raise the threshold to 0.7 to focus on the **most promising leads** and avoid wasting resources on low-conversion leads.

Focus on High-Value Leads:

- ✓ Prioritize outreach to leads with scores greater than **85**, ensuring maximum return on effort.

Leverage Automated Nurturing for Lower-Priority Leads:

- ✓ Engage lower-priority leads through automated **email nurturing** sequences.
- ✓ Phone calls should only be made to those who show engagement (e.g., opened emails, or visited the website).

Conclusion:

With high accuracy, strong recall, and an effective lead scoring mechanism, the project has successfully achieved its goal in identifying the most promising leads.