Lead Scoring Case Study
Lead Conversion Analysis for X Education

SUBMITTED BY :

- Vunna Praveen Kumar
- Praful Pillay
- Sathish Madhiyalagan

Contents

- Problem Statement
- Objectives
- Data Description
- Data Preparation and EDA
- Data Dictionary
- Model Building Approach
- Model Evaluation
- Model Results
- Business Implications
- Summary Report
- Conclusion

Problem Statement :

- X Education sells online courses to industry professionals.
- Many professionals visit the website, browse courses, and fill out forms to express interest.
- Leads are acquired through marketing efforts and referrals.
- The sales team contacts these leads, but the conversion rate is around 30%.
- The goal is to identify "Hot Leads" to increase the conversion rate to around 80%.

Objectives :

- Build a logistic regression model to assign a lead score between 0 and 100.
- Improve the lead conversion process by focusing on high-potential leads.
- Address additional problems presented by the company in the future.

Data Description :

- Dataset with 9000 data points.
- Various attributes such as Lead Source, Total Time Spent on Website, Last Activity, etc.
- Target variable: 'Converted' (1 = Converted, 0 = Not Converted).

Data Preparation and EDA :

- Print information about the dataset, such as the data types of each column and the number of non-null values.
- Calculate and display summary statistics (e.g., mean, standard deviation, min, max) for numerical columns.
- Identify and print the number of missing values in each column to assess data completeness.
- Determine and print the number of duplicate rows in the dataset, if any.
- Plot a histogram to visualize the distribution of the target variable 'Converted' , which represents Lead Source.

Then we go with **Univariate analysis,Bivariate Analysis and Multivariate Analysis.**

- Univariate Analysis (categorical variables or numerical variables)

  Univariate analysis examines individual variables to understand their distribution and properties, using techniques like histograms and summary statistics.

- Bivariate Analysis (Categorical vs. Categorical ,Categorical vs. Numerical and Numerical vs. Numerical)

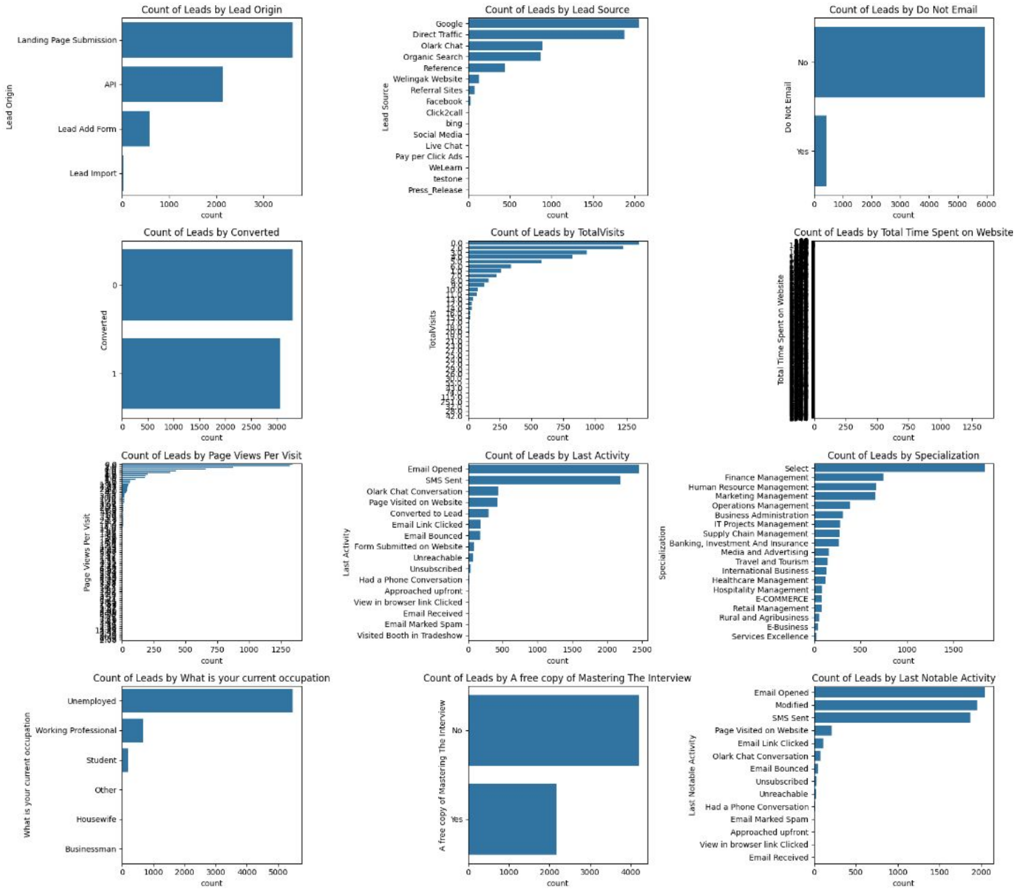
  Bivariate analysis explores relationships between pairs of variables, revealing patterns and correlations through scatter plots and cross-tabulations.

- Multivariate Analysis (Its combined all)

  Multivariate analysis delves into interactions among multiple variables, uncovering complex patterns and relationships using techniques like PCA, factor analysis, and multiple regression.

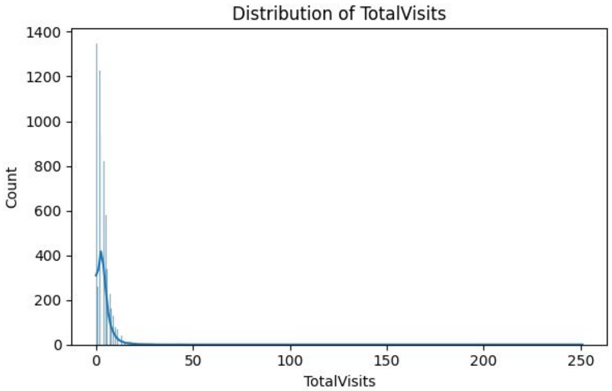
Histograms for numerical variables

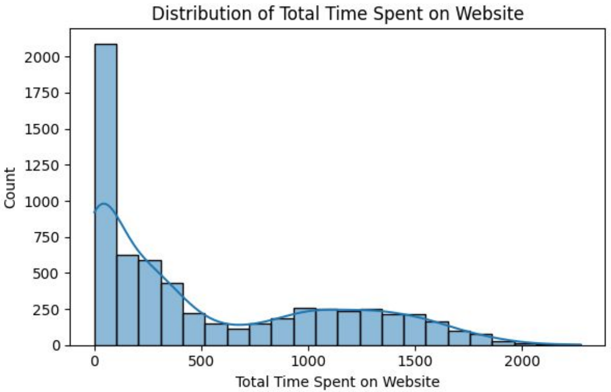
Univariate analysis

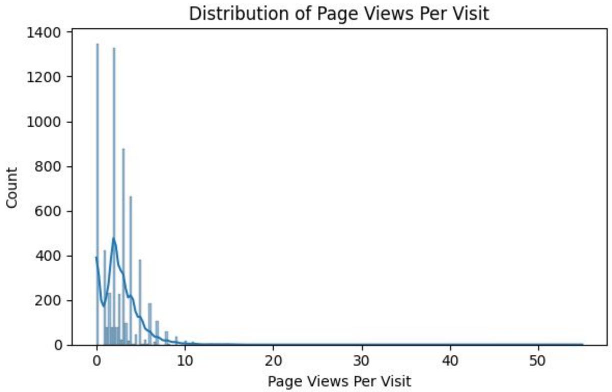


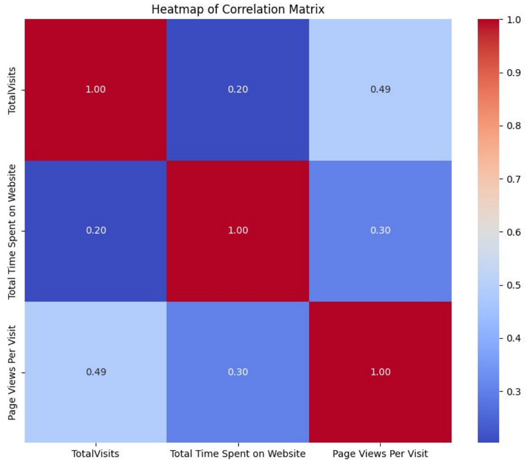
Bivariate Analysis

Categorical vs. Categorical





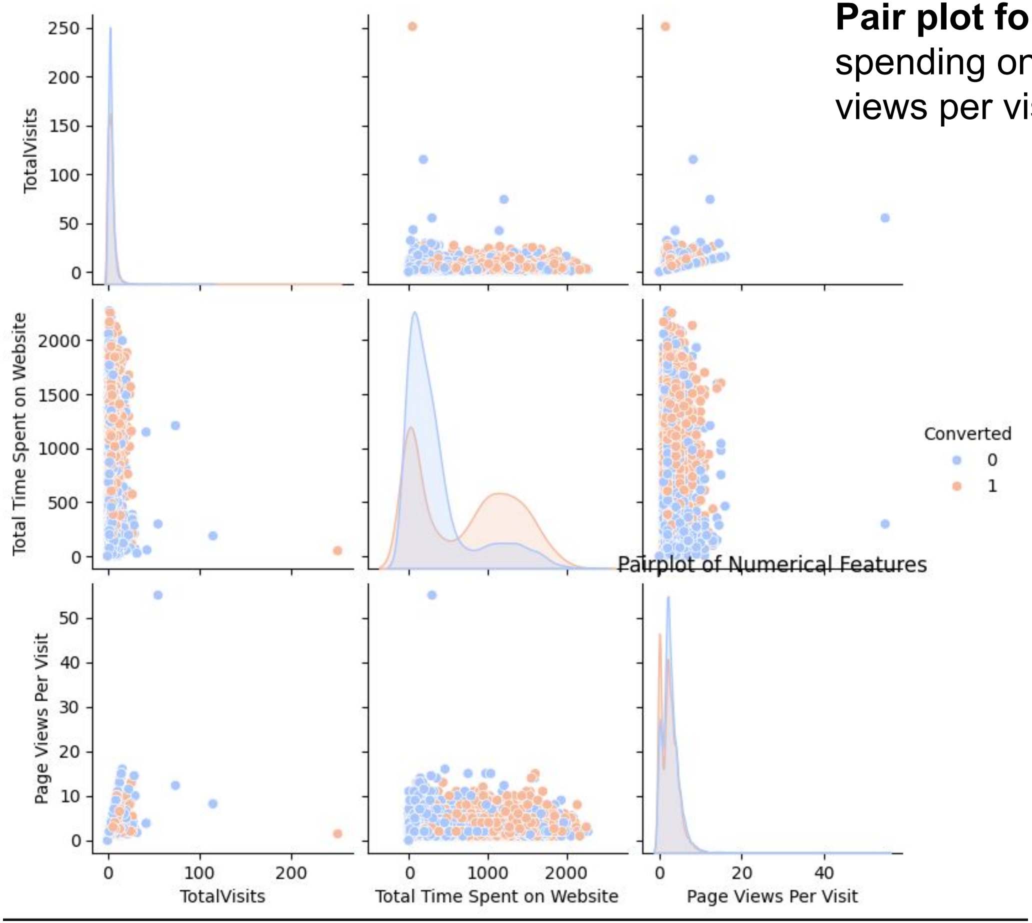




Heat map for totalVisits ,time spending on website and page views per visit

Multivariate Analysis

Pair plot for totalVisits ,time spending on website and page views per visit



In the preprocessing phase, it's crucial to identify and remove any columns that are not relevant to the prediction task or those that do not contribute meaningful information to the model. Based on my statement, it seems four columns that are not correct or necessary for this feature.

That four columns are ('TotalVisits', 'Page Views Per Visit', 'Last Activity', 'Lead Source', 'Specialization')

One more time check ,Missing Values and null values ,Categorical to encode numerical format

Dummy Variables Creation

- Transform categorical variables into numerical dummy variables to prepare data for analysis and modeling.
- Convert `True` and `False` values to `1` and `0`

- Columns : ['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']

- At the same time Create dummy variables and drop the 'Select' level.like Concatenate dummy variables and drop original categorical columns.

Splitting the Data into Training and Testing Sets :

- To effectively train and evaluate a machine learning model, it is crucial to divide the dataset into training and testing sets. This allows the model to learn from the training set and be evaluated on unseen data in the testing set, helping to ensure that the model generalizes well to new data.

- Import the train_test_split function from the sklearn.model_selection module. This function is used to split the dataset into training and testing sets.

- Set the random seed using np.random.seed(0). This ensures that the splitting of data is reproducible, meaning that every time you run the code, the training and testing sets will have the same rows.

- Use the train_test_split function to split the DataFrame into training and testing sets. The train_size=0.7 parameter specifies that 70% of the data should be used for training, while the test_size=0.3 parameter specifies that 30% should be used for testing. The random_state=100 parameter ensures that the split is consistent every time the code is run.

Rescaling the Features

- Specify the list of numerical columns that need to be scaled.Import the MinMaxScaler class from the sklearn.preprocessing module to perform scaling.
- Create an instance of the MinMaxScaler.Fit the MinMaxScaler on the training set's numerical columns and transform the data, scaling the values to a range of [0, 1].
- Display summary statistics for the scaled numerical columns to understand their distribution after scaling.
- Create a heatmap to visualize the correlations between the scaled numerical features in the training set.
- This visualization helps identify strongly correlated features, which can inform feature selection and engineering decisions in the modeling process.

Adding Variable and Features selection

- Feature Added: 'TotalVisits'
- Model Evaluation: Used Ordinary Least Squares (OLS) regression.
- Result: Achieved an R-squared value of 0.414, indicating that the 'TotalVisits' feature alone explains 41.4% of the variance in the target variable.

Added Additional Numerical Features:

- Features Added: 'Total Time Spent on Website', 'Page Views Per Visit'
- Model Evaluation: Included these features in the regression model.
- Result: The model's performance improved, with an R-squared value of 0.621

Repeated the process by adding all numerical columns to the feature set

Variance Inflation Factor (VIF) Checking

- The code you provided calculates the Variance Inflation Factor (VIF) for each feature variable in your dataset. The VIF helps assess multicollinearity among predictor variables in a regression analysis.
- Variables with high VIF values, typically above 5, indicate strong multicollinearity, suggesting that those variables are highly correlated with other predictor variables in the model.
- Based on the results of the VIF calculation, you should identify variables with high VIF values and consider dropping them from your model to address multicollinearity issues
- Look for variables with high VIF values, typically above 5, indicating multicollinearity.

Dropping the variable and updating the model

Look for variables with high VIF values, typically above 5, indicating multicollinearity.

[Lead Source_Reference,
Last Notable Activity_Had a Phone Conversation,
What is your current occupation_Housewife,
What is your current occupation_Working Professional,
]

Variables with p-values less than the chosen significance level (usually 0.05) are considered statistically significant.

Model Evaluation and Performance

1. Predicting Probabilities

- Train Set Predictions:
  - Example Probabilities:
    - 0.3001, 0.1420, 0.1276, 0.2916, 0.9548
  - Predicted Values (Threshold 0.5):
    - 0, 0, 0, 0, 1

2. Confusion Matrix (Train Set)

- Confusion Matrix:
  - True Negatives (TN): 1929
  - False Positives (FP): 383
  - False Negatives (FN): 560
  - True Positives (TP): 1589

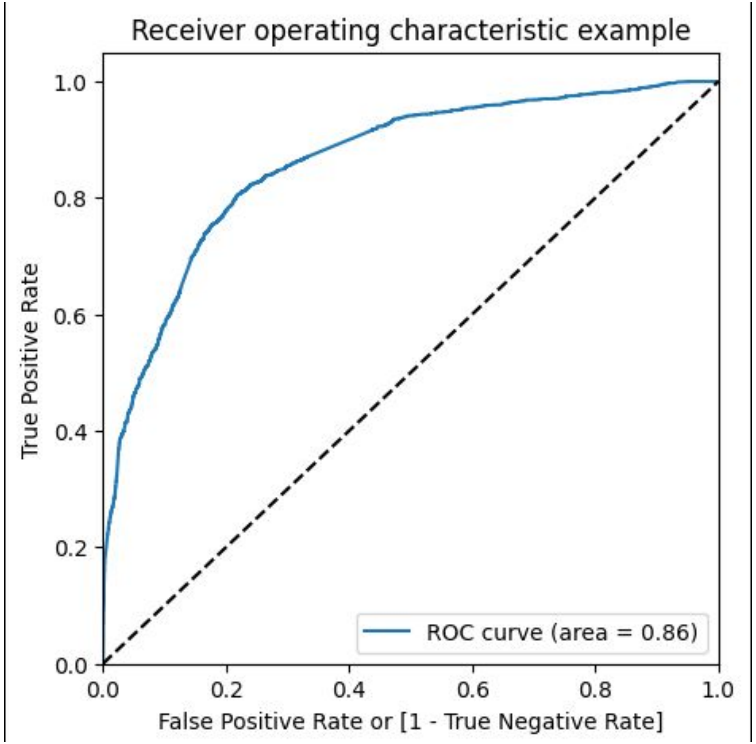
3. Accuracy and Metrics

- Accuracy: 0.749
- Sensitivity (Recall):0.739
- Specificity: 0.834

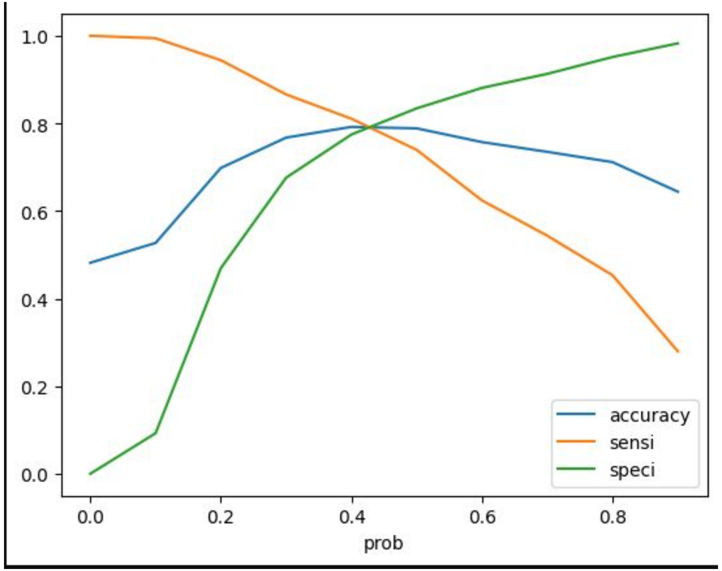
4. ROC Curve Analysis

- ROC Curve:
  - AUC Score:0.86 (Strong performance)
- Visual Representation: ROC curve showing True Positive Rate vs. False Positive Rate

ROC Curve Analysis



Optimal Cutoff Analysis



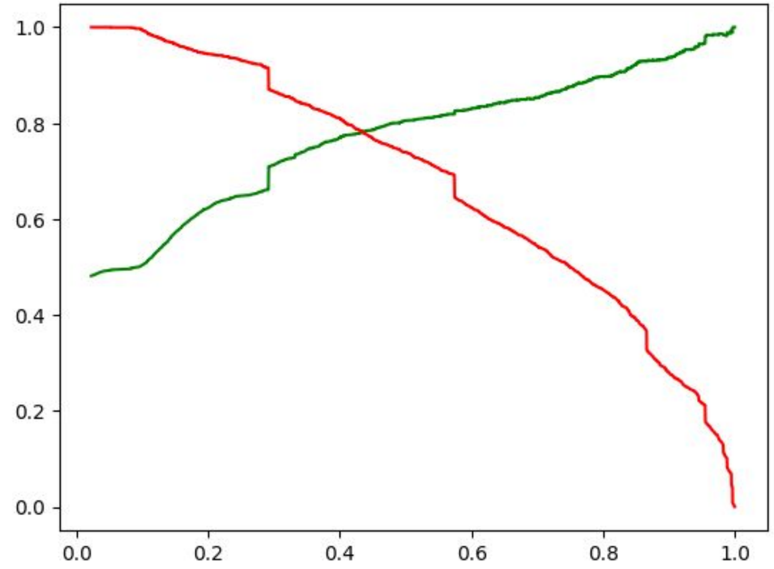
5. Optimal Cutoff Analysis

- Optimal Cutoff Threshold:0.42
- Revised Accuracy:0.785
- Confusion Matrix (Test Set):
  - True Negatives (TN): 786
  - False Positives (FP): 210
  - False Negatives (FN): 202
  - True Positives (TP): 714

6. Precision-Recall Tradeoff

- Precision: 0.777
- Recall:0.793
- Visual Representation: Precision and Recall vs. Threshold Curve

Precision-Recall Curve



7. Model Suitability

- Target Conversion Rate: ~80%
- Model Recall Achieved: 80%
- Business Fit: Model aligns well with business objectives and expected lead conversion rates

Conclusion:

The logistic regression model demonstrates strong performance with an ROC-AUC of 80%, and precision-recall metrics indicating suitability for business needs. Adjusted threshold improves accuracy and meets target recall rate.

Recommendations:

Fine-tune the cutoff threshold based on business objectives and desired balance between precision and recall. The current threshold of 0.42 provides a good balance but should be monitored for further optimization.

Explore additional features or interactions that could improve model performance.