Lead Scoring Using Logistic Regression

1. Problem Statement

- **Objective**: The company aims to develop a logistic regression model to assign a lead score between 0 and 100 for each of the leads. The score will indicate the likelihood of the lead converting, where a higher score signifies a higher probability of conversion. The goal is to use these lead scores to prioritize leads that are more likely to convert.
- **Business Goal**: By identifying leads with higher conversion potential (hot leads), the company can allocate resources more effectively to maximize conversion rates.
- **Scope**: The model should be adaptable to future changes in the dataset or company requirements, allowing the company to continuously refine their targeting strategy.

2. Data Exploration and Preprocessing

- **Dataset**: The dataset contains information on potential leads, with features like `Lead Source`, `TotalVisits`, `Total Time Spent on Website`, `Lead Quality`, `Country`, and `City`. These features are believed to influence whether a lead will convert.
- **Missing Values**:
  - Missing values were checked and handled by imputing numerical columns with the mean value and categorical columns with the mode.
  - No rows were dropped, and missing data was managed efficiently using imputation techniques.
- **Feature Engineering**:
  - Categorical variables (e.g., `Lead Source`, `Country`) were encoded using one-hot encoding to convert them into numerical format.
  - We also transformed some continuous features like `Total Visits` and `Total Time Spent on Website` using scaling methods to normalize them.
- **Target Variable**: The target variable is `Converted`, which indicates whether a lead converted (1) or did not convert (0).

3. Logistic Regression Model

- **Model Choice**: Logistic Regression was chosen for this task because the goal is to predict the likelihood of binary events (conversion vs. no conversion). Logistic regression provides the probability of a lead converting and assigns a score between 0 and 1.
- **Training the Model**:
  - The dataset was split into a training set (80%) and a testing set (20%).
  - The logistic regression model was trained using the training data, and the target variable (`Converted`) was predicted for the testing set.
  - No hyperparameter tuning was performed in this basic model setup, but parameters like `max_iter` were adjusted for convergence.
- **Code for Model Training**: Explained in the Python Notebook.

4. Model Evaluation

- **Metrics Used**:

- **Accuracy**: This metric tells us the proportion of correct predictions (both true positives and true negatives) made by the model.
- **Precision, Recall, F1-Score**: These metrics were calculated to assess how well the model handles both positive (converted) and negative (non-converted) predictions.
- **ROC-AUC**: The area under the ROC curve provides an aggregate measure of performance across all possible classification thresholds.

- **Results**:
  - The model performed with an accuracy of , and the precision, recall, and F1-score were evaluated as follows:

Precision: **0.7257844474761255**
Recall: 0.49396471680594245
F1-Score: 0.5878453038674033

▪

The ROC-AUC was calculated to be **0.7497588351780182,** indicating good classification perfor mance.

- **Confusion Matrix**:
  - The confusion matrix was used to identify the number of true positives, false positives, true negatives, and false negatives. The results show how well the model is distinguishing between leads that convert and those that do not.

5. Business Implications and Recommendations

- **Targeting Strategy**:
  - The lead scores can be used by the marketing and sales teams to prioritize their efforts. Leads with higher scores (e.g., above 80) are more likely to convert, and thus, they should be targeted more aggressively.
  - Low-scoring leads (e.g., below 30) should be deprioritized or targeted with less effort.
- **Resource Allocation**:
  - By focusing on high-scoring leads, the company can optimize their marketing budget, ensuring that efforts are concentrated on the leads with the highest potential for conversion.
- **Future Model Adjustments**:
  - The model can be easily updated as new data becomes available. Additionally, if the company introduces new features (e.g., lead demographic data), the model can be retrained to incorporate these changes.
  - The company can experiment with other models such as Random Forest or XGBoost to see if they provide better performance in the future.

6. Handling Future Problems (from the company's document)

- **Problem 1: Changes in Lead Sources**
  - If the company introduces new lead sources (e.g., social media, offline events), the model can be retrained to include these new sources. Categorical variables will be re-encoded to reflect new categories.

- **Problem 2: Feature Updates**
  - If the company adds new features (e.g., lead behavior data from their website), these can be included in the model, and feature engineering will be required to preprocess the new data.
- **Problem 3: Class Imbalance**
  - If the data becomes imbalanced (e.g., too few conversions), techniques like oversampling or SMOTE can be used to balance the classes before training the model.

7. Conclusion

- The logistic regression model provides a reliable method for predicting lead conversion, and the results show that the model is able to predict lead scores that align well with actual conversions.
- The model can be easily updated and adjusted to accommodate changes in the dataset or business requirements.
- The lead scoring system helps the company make data-driven decisions for prioritizing leads, which will likely improve conversion rates and reduce wasted marketing efforts.

Appendix

- **Code Snippets**: Include the Python code used for training the logistic regression model, evaluating the model, and generating the necessary metrics.
- **Visualizations**: Include the following charts and graphs:
  - **ROC Curve**: For model evaluation.
  - **Confusion Matrix**: To assess classification performance.
  - **Feature Importance**: If applicable, showing the importance of different features in predicting conversion.
- Precision: 0.7257844474761255
- Recall: 0.49396471680594245
- F1 Score: 0.5878453038674033