Approach and Model Evaluation Report

1. Understand the Problem and Dataset

The initial step involved loading and exploring the dataset to understand its structure, including data types, number of rows, and columns. This foundational understanding is crucial for appropriate preprocessing and feature engineering.

2. Data Exploration and Preprocessing

Data Size and Memory Optimization: Observing that the dataset's size and memory usage were considerable, I implemented downcasting techniques to reduce memory usage by half. This process involved converting object types to categorical, downcasting integer and float types, thereby optimizing the dataset for efficient processing.

Missing Data: A thorough check revealed no missing data, simplifying the preprocessing pipeline as imputation was not required.

Feature Range Analysis: Analysis of feature ranges indicated significant disparities among feature values. For instance, `Museums_Historical_Sites_and_Similar_Institutions_total_minimum_dwell` had a range of 43,044, while `driving_school_30_days_visits` had a range of only 18. This disparity highlighted the need for normalization to ensure that all features contribute equally to the model, particularly when using algorithms sensitive to feature scales.

Feature Selection: Recognizing the potential issue of high dimensionality, I performed correlation analysis to identify and remove highly correlated features. For example, features such as `automobile_30_days_visits` and `automobile_7_days_visits` were found to be highly correlated. Preference was given to features with longer time frames (e.g., _30 over _7) based on the assumption that they carry more information.

Feature Importance and Regularization: To further refine feature selection, I used `RandomForestClassifier` to gauge feature importance, retaining features with an importance score above 0.01. This approach provided a better understanding of which features were most relevant. Additionally, L1 regularization was employed to identify and retain the most positively and negatively impactful features, focusing on those with the highest contributions.

Feature Engineering: New features were engineered by creating interaction terms to enhance model interpretability. For instance:

- `automobile_visits_inter` was created by multiplying `automobile_30_days_visits` with `automobile_30_days_total_dwell_time`.
- Similar interactions were created for `driving_school` and `grocery` visits. This engineering aimed to capture relationships between visits and dwell time, potentially improving the model's performance.

3. Model Selection

Three distinct data sets (train, test, and validation) were created to ensure effective model training, validation, and evaluation. Various classifiers were tested, including `RandomForestClassifier`, `LogisticRegression`, `KNeighborsClassifier`, and `DecisionTreeClassifier`, to determine the best-performing algorithm. Each classifier operates on different principles, and evaluating them provides insights into which model is most suitable for the problem.

Pipelines included scaling and SMOTE to address feature range disparities and class imbalance, respectively. Scaling ensured all features were on a similar scale, and SMOTE addressed the imbalance by generating synthetic samples for the minority class.

Choosing Recall: Recall was selected as the primary metric due to the nature of the business problem. High recall ensures that most potential car buyers are identified, minimizing the risk of missing valuable sales opportunities. In an imbalanced dataset with few potential buyers, optimizing recall helps capture a larger proportion of true positives, which is more critical than precision in this context.

Cross-Validation: K-fold cross-validation was used to assess the generalization ability of each model, providing a robust evaluation of their performance.

4. Hyperparameter Tuning

`LogisticRegression`, which performed best based on recall, was fine-tuned using grid search on the training and validation data. The grid search optimized hyperparameters (`C` and `solver`) to enhance performance, focusing on improving recall.

5. Inference

After identifying the best model, the final performance was evaluated using a confusion matrix, which is a valuable tool for understanding classification performance. It provides insights into true positives, false positives, true negatives, and false negatives, essential for evaluating recall and precision.

Observations:

1. **Recall**: At approximately 30%, the model identifies only 30% of actual potential car buyers, missing a significant number of true positives.
2. **Precision**: At about 12%, the model's positive predictions are unreliable, with many false positives.
3. **False Positive Rate (FPR)**: The model effectively avoids false positives, rarely misclassifying non-buyers as buyers.
4. **False Negative Rate (FNR)**: The high FNR of about 70% indicates the model fails to identify many potential buyers, crucial for improving targeting strategies.

Final Conclusion: The model is effective at identifying non-car buyers but struggles with accurately identifying potential buyers, making it suitable for filtering out non-buyers. However, it is not ideal for scenarios requiring high accuracy in identifying actual buyers, as it misses many potential car buyers. Future improvements could focus on enhancing recall and precision to better align with the business objectives of identifying potential car buyers.