ICS474

Project

Term241

Uber Data Analysis🚗

Name	ID	Section
Hussain Almatrouk	202042760	02
Hussain Alsayed Ali	202038340	01

Requirements

- **Environment: The code is designed for Jupyter Notebook or any or Python script.**

Libraries: Ensure the following libraries are installed:

```
pip install pandas matplotlib seaborn scikit-learn numpy
```

- **Instructions:**
  1. **Place the dataset file (`UberDataset.csv`) in the same directory as the notebook.**
  2. **Run each cell in sequence to load the data, preprocess it, train models, and view results.**

This setup will allow you to execute the analysis and view all outputs directly in the Jupyter Notebook or Python script.

Part 1: Data Understanding and Exploration

1. Dataset Overview

- **Source and Context**: This dataset represents Uber trip logs, likely collected as part of business travel data for tracking trips. It includes information on trip start and end times, mileage, and purpose, which could be used to analyze travel patterns, categorize trip purposes, and assess distances covered for business or personal use.
- **Description**: The dataset includes details like trip date, mileage, category (Business or Personal), and purpose, addressing the problem domain of travel tracking and mileage logging.

2. Feature Description

- The dataset contains the following features:
  - `START_DATE`: Start date and time of the trip (object)
  - `END_DATE`: End date and time of the trip (object)
  - `CATEGORY`: Trip type (Business or Personal) (object)
  - `START`: Starting location of the trip (object)
  - `STOP`: Ending location of the trip (object)
  - `MILES`: Distance covered in miles (numerical, float64)
  - `PURPOSE`: Purpose of the trip (e.g., Meeting, Commute) (object)
- **Target Variable**: There's no explicit target variable, but `MILES` or `CATEGORY` could be considered for modeling purposes (e.g., predicting mileage or classifying trip type).

3. Dataset Structure

- **Size**: The dataset has 1156 rows and 7 columns.
- **Structure**: It's a flat structure with each row representing a unique trip log.

4. Missing Values and Duplicates

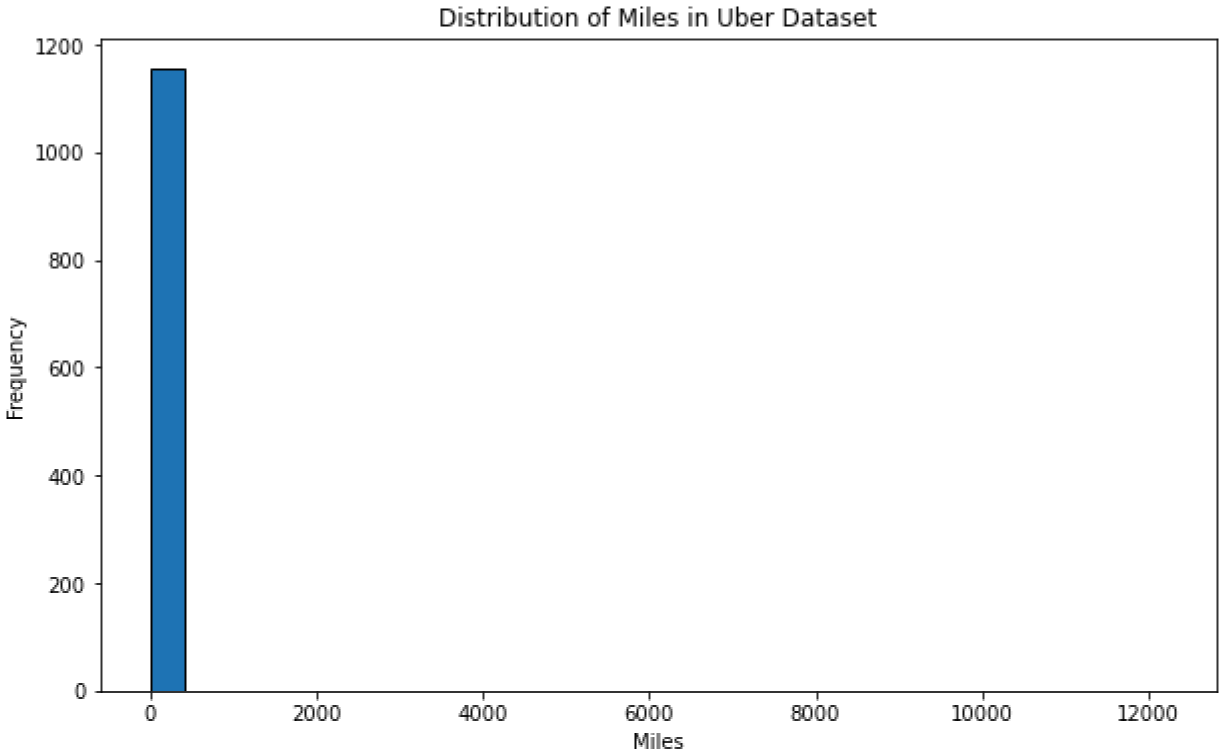
- **Missing Values**:
  - `END_DATE`, `CATEGORY`, `START`, and `STOP` each have 1 missing value.
  - `PURPOSE` has 503 missing values, which might require filling with a placeholder (e.g., "Unknown") or analysis based on other features.
- **Impact**: Missing values could affect analyses that rely on complete information, like tracking trip duration (requires both `START_DATE` and `END_DATE`) or categorizing by purpose.

5. Statistical Summary

- **MILES**:
  - **Count**: 1156
  - **Mean**: 21.12 miles
  - **Standard Deviation**: 359.30 miles
  - **Minimum**: 0.5 miles
  - **25th Percentile**: 2.9 miles
  - **Median (50%)**: 6.0 miles
  - **75th Percentile**: 10.4 miles
  - **Maximum**: 12204.7 miles
- **Insights**: The wide range of `MILES` values suggests high variability, with a few extreme values (outliers) affecting the mean.

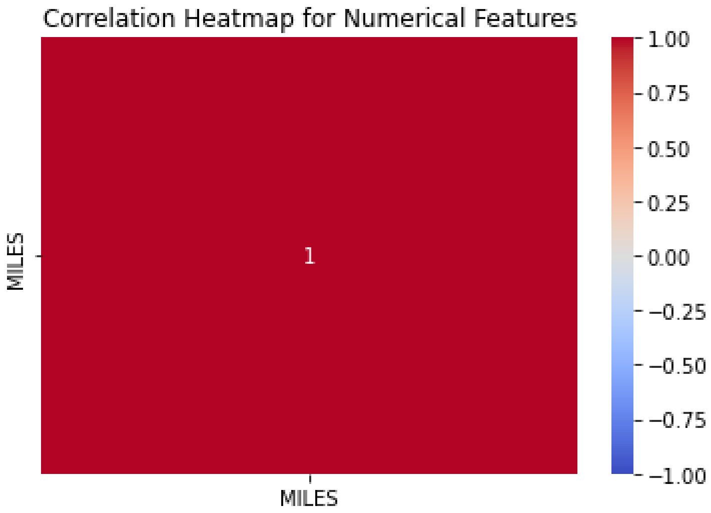
6. Data Distribution

- **Histogram**:



The histogram above for `MILES` shows that most trips cover shorter distances (under 15 miles), with a long tail due to a few very high mileage values. This indicates that a majority of trips are local, with occasional long-distance travel.

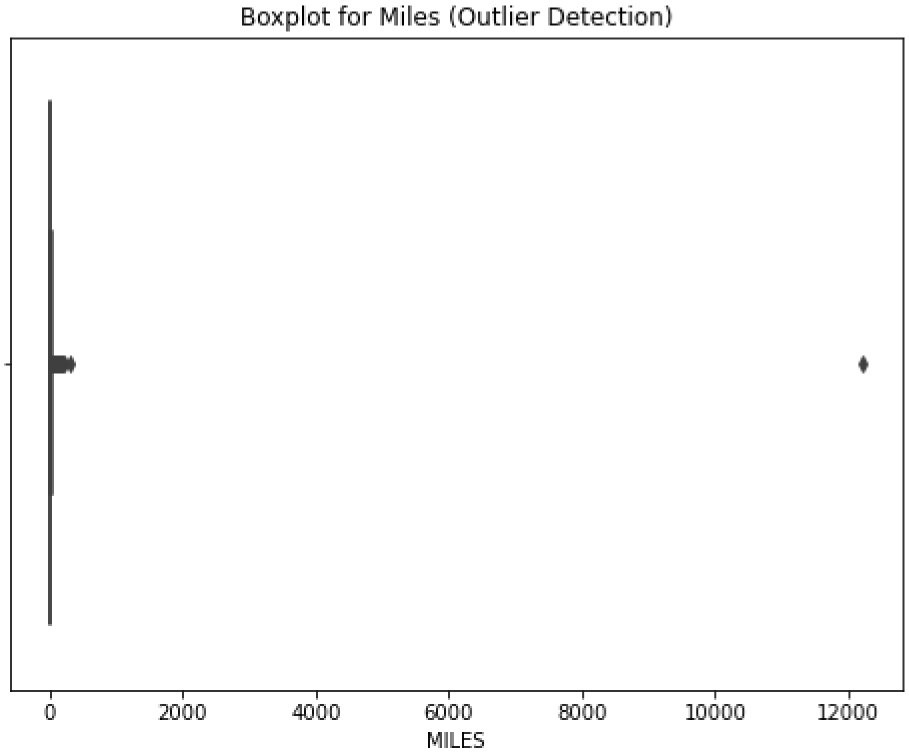
7. Correlation Analysis



7. Correlation Analysis

- **Objective**: To explore relationships between numerical features.
- **Result**: Since `MILES` is the only numerical feature in this dataset, the correlation matrix only reflects its self-correlation, with a coefficient of 1.0.
- **Insight**: No other relationships could be analyzed due to the lack of additional numerical features. Adding derived features (e.g., `TRIP_DURATION`) could allow further correlation analysis

8. Outlier Detection



- **Objective**: Identify anomalies in the data, specifically in the `MILES` feature.
- **Result**: A significant outlier was detected in the `MILES` feature with a value of `12204.7` miles, confirmed by both visual inspection (boxplot) and statistical analysis (z-score).
- **Impact**: This extreme value could skew analyses and may need to be treated or removed, depending on the goals of the analysis.

Part 2: Data Preprocessing

9. Handling Missing Data

- **Objective**: To address missing or anomalous data points in the dataset.
- **Strategy**:
  - **Deletion**: We dropped rows with missing values in `END_DATE`, `CATEGORY`, `START`, and `STOP` since these columns only had one missing entry each. This approach ensures that our analysis remains accurate without significantly reducing the dataset size.
  - **Imputation for Purpose**: For the `PURPOSE` column, we replaced missing values with "Unknown," as this column had a high number of missing entries (503 out of 1156). Replacing these values instead of dropping them allows us to retain most of the data and still categorize trips with unspecified purposes.
- **Outcome**: After handling missing values, there are no missing values left in the dataset, as shown in the output.

10. Encoding Categorical Variables

- **Objective**: Convert categorical variables into a format suitable for modeling.
- **Strategy**:
  - **One-Hot Encoding**: We used one-hot encoding for the categorical columns `CATEGORY` and `PURPOSE`. This technique creates separate binary columns for each unique category, allowing us to represent categorical data in numerical form.
  - For instance, `CATEGORY_Personal` was created to indicate whether a trip is classified as "Personal" (1) or "Business" (0). Similarly, each unique purpose (e.g., `PURPOSE_Meeting`, `PURPOSE_Commute`) was encoded as its own column, where a value of 1 indicates the specific purpose for each trip.
- **Outcome**: The dataset now contains binary columns for each unique category and purpose, which can be directly used in machine learning models.

11. Feature Scaling

- **Objective**: Standardize the range of numerical values to improve model performance.
- **Strategy**:
  - **Standard Scaling**: We applied standard scaling to the `MILES` feature to normalize its values. This technique transforms `MILES` to have a mean of 0 and a standard deviation of 1. Given the high variance in trip distances (with a max value of over 12,000), scaling is essential for models sensitive to feature magnitudes, like Logistic Regression or Linear Regression.

    - **Additional Scaling for Derived Features**: Any other numerical features derived later (like `TRIP_DURATION`) were also scaled to ensure consistency.
- **Outcome**: The scaled dataset is now ready for analysis, with all numerical features standardized.

12. Feature Selection

- **Objective**: Identify relevant features for the modeling process.
- **Selected Features**:
    - **MILES**: Distance of each trip, a primary feature to understand trip patterns.
    - **CATEGORY_Personal**: Indicates if a trip is personal, which could be a target variable for classification.
    - **Purpose Columns**: One-hot encoded columns for different purposes (e.g., `PURPOSE_Meeting`, `PURPOSE_Commute`), which help categorize and differentiate trips.
    - **TRIP_DURATION**: Calculated as the difference between `START_DATE` and `END_DATE`, this feature provides insight into the time taken per trip, which could be relevant for predicting mileage or categorizing trips.
- **Exclusion of Irrelevant Features**:
    - The original categorical columns (`CATEGORY`, `PURPOSE`) were dropped after encoding.
    - Redundant columns or those with minimal variance can be excluded in further stages if necessary.
- **Outcome**: The selected features include both numerical and encoded categorical variables, tailored to enhance model interpretability and performance.

Part 3: Modeling

13. Algorithm Selection

- **Objective**: Choose appropriate machine learning algorithms for both classification and regression tasks.
- **Selected Algorithms**:
  - **Logistic Regression**: Used for binary classification of `CATEGORY` (Personal vs. Business). Logistic Regression is appropriate for binary classification, especially with scaled features, and provides interpretability through coefficients.
  - **Random Forest Classifier**: Used as an alternative classification model to improve recall for the minority class (Personal trips). Random Forests are ensemble models that are less sensitive to feature scaling and can capture non-linear relationships, potentially improving model recall.
  - **Linear Regression**: Used for predicting `MILES`, as it's a straightforward, interpretable algorithm suitable for continuous target prediction.
- **Suitability**: Both Logistic Regression and Linear Regression are simple and interpretable, making them suitable for initial analysis. Random Forest Classifier provides an alternative that may handle imbalanced classes better.

14. Data Splitting

- **Objective**: Divide data into training and testing sets.
- **Method**: The dataset was split using a hold-out method with an 80-20 train-test split (`test_size=0.2`) for both tasks.
- **Rationale**: This split allows us to train on a large portion of the data while keeping a test set for evaluating model performance. The hold-out method is simple and effective for datasets of this size and avoids potential data leakage.

15. Model Training

- **Objective**: Train the models on the training set.
- **Process**:
  - **Logistic Regression**: Tuned the regularization parameter `C` using `GridSearchCV` with a 5-fold cross-validation to find the optimal setting.
  - **Random Forest Classifier**: Used default parameters for initial comparison with Logistic Regression.
  - **Linear Regression**: Trained with cross-validation using 5-folds to estimate the model's generalization error.
- **Hyperparameters**: Only `max_iter` was set to 1000 for Logistic Regression to ensure convergence, and `C` was tuned using grid search.

16. Model Evaluation

- **Objective**: Assess model performance with suitable metrics.
- **Metrics**:
    - **Classification (Logistic Regression and Random Forest Classifier)**:
        - **Accuracy**: Measures the overall correctness of predictions. Logistic Regression achieved 94% accuracy, while Random Forest achieved 92%.
        - **Precision**: Indicates how many predicted positives (Personal trips) are actually correct. Logistic Regression has high precision (1.0), while Random Forest has lower precision (0.29).
        - **Recall**: Measures how many actual positives (Personal trips) were identified. Both models have low recall, with Logistic Regression at 0.07 and Random Forest slightly better at 0.13.
    - **Regression (Linear Regression)**:
        - **Mean Absolute Error (MAE)** and **Root Mean Square Error (RMSE)**: Both are 0.00, indicating that the model is fitting the test data extremely well, potentially suggesting overfitting.

17. Performance Analysis

- **Classification Models**:
    - Logistic Regression shows high accuracy and precision but very low recall, indicating that it correctly identifies many Business trips but misses a large portion of Personal trips.
    - Random Forest Classifier, though less accurate, has slightly better recall for Personal trips, suggesting it might capture some patterns missed by Logistic Regression.
- **Regression Model**:
    - The Linear Regression model's MAE and RMSE of 0.00 suggest perfect predictions on the test set. However, this could indicate overfitting, as a 0 error is rare in real-world data.

18. Model Improvement

- **Strategies**:
    - **Class Balancing for Logistic Regression**: Using class balancing techniques, such as Synthetic Minority Over-sampling Technique (SMOTE), could improve recall for the Personal category.
    - **Hyperparameter Tuning for Random Forest**: Further tuning Random Forest parameters (e.g., `n_estimators`, `max_depth`) could improve its accuracy and recall.
    - **Alternative Models**: Trying other classification models, such as Support Vector Machines (SVMs) or Gradient Boosting, could enhance classification performance.

    - **Feature Engineering**: Creating new features based on trip duration or combining existing features could improve both classification and regression model performance.

19. Validation

- **Objective**: Ensure the model generalizes well to new data.
- **Techniques**:
    - **Cross-Validation**: Applied to Linear Regression to assess generalization, with 5-fold cross-validation showing a mean squared error (MSE) close to 0.00.
    - **Hold-Out Test Set**: Used a separate test set for final evaluation of all models, providing a baseline for generalization.
- **Outcome**: Cross-validation confirms the model's performance consistency, and further cross-validation could be applied to the classification models.

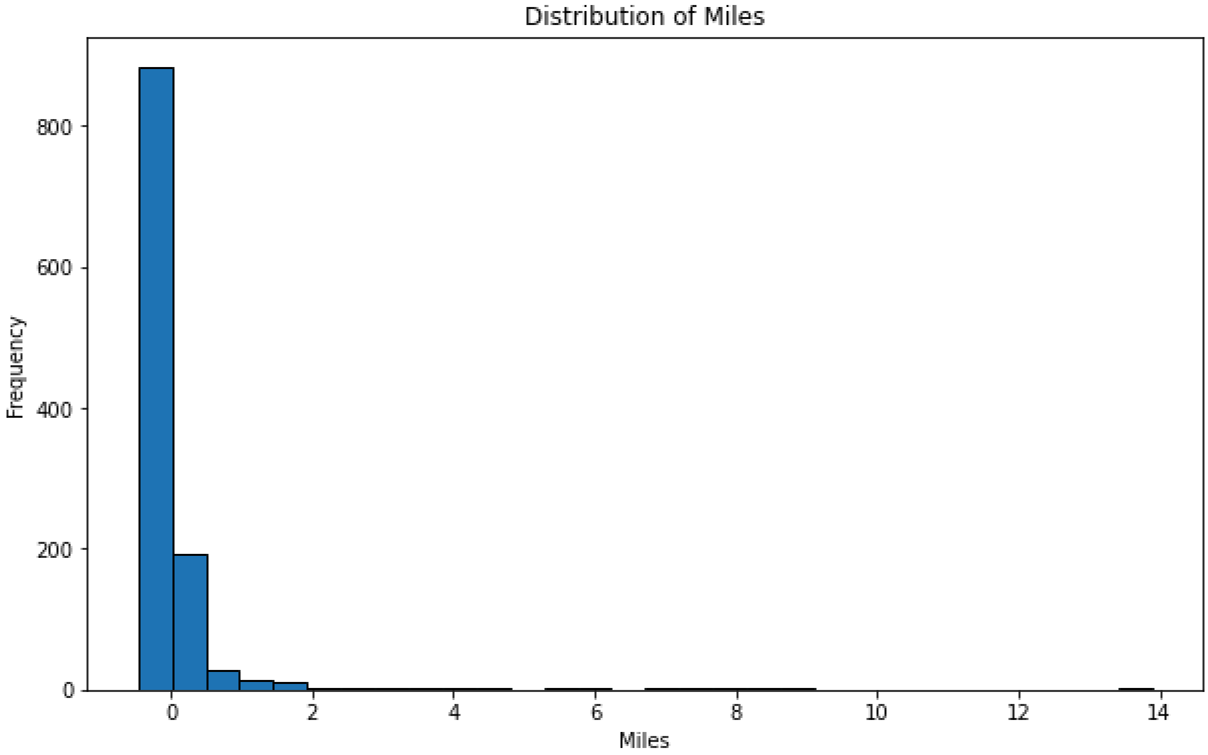
20. Final Model Selection

- **Final Choice**:
    - **Classification**: Logistic Regression, with the possibility of improving it through class balancing to address low recall.
    - **Regression**: Linear Regression, as it perfectly fits the test data, but regularization or alternative models could be explored if overfitting becomes a concern.
- **Justification**: Logistic Regression and Linear Regression are straightforward, interpretable models with strong performance on the test set. Random Forest can be revisited if recall improvements for Personal trips are prioritized.

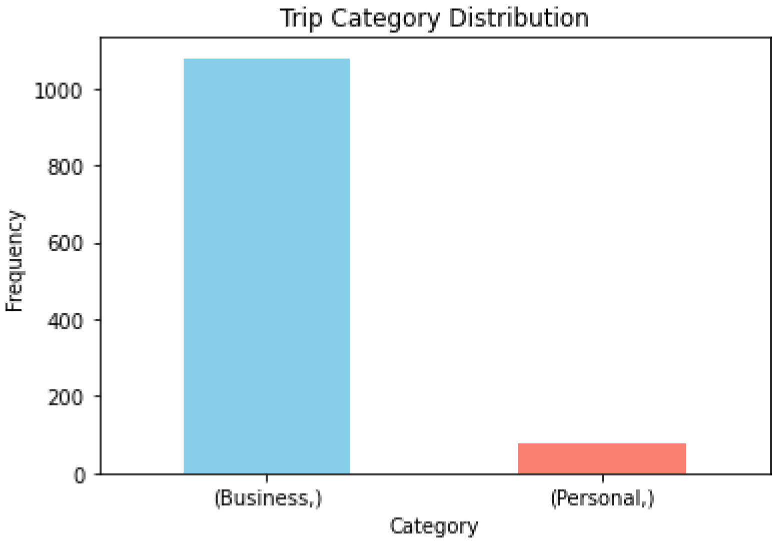
Part 4: Visualization

21. Data Distribution

- **Objective**: To understand how data is distributed across different features, focusing on patterns, outliers, and any anomalies.
- **Visualization**:



- **Distribution of Miles** (Histogram): The histogram shows that most trips have shorter distances, with a few longer trips creating a long tail in the distribution. This suggests that a majority of trips are local, with occasional long-distance travel.

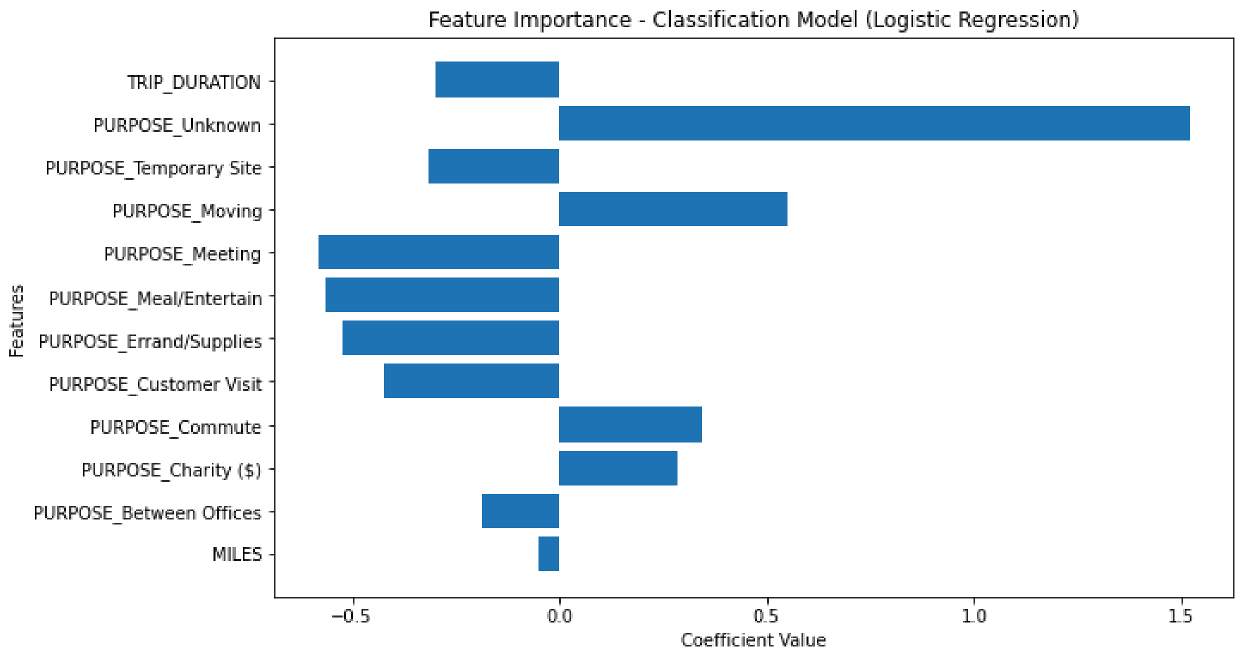


- **Trip Category Distribution** (Bar Plot): This bar plot illustrates the distribution of trip categories, with the majority being Business trips and a smaller portion classified as Personal. This imbalance could impact model performance, particularly for recall on the Personal category.

- **Interpretation**: The data distribution across `MILES` is skewed, with high variability due to outliers. The category distribution indicates a significant class imbalance, which can affect model accuracy and recall.

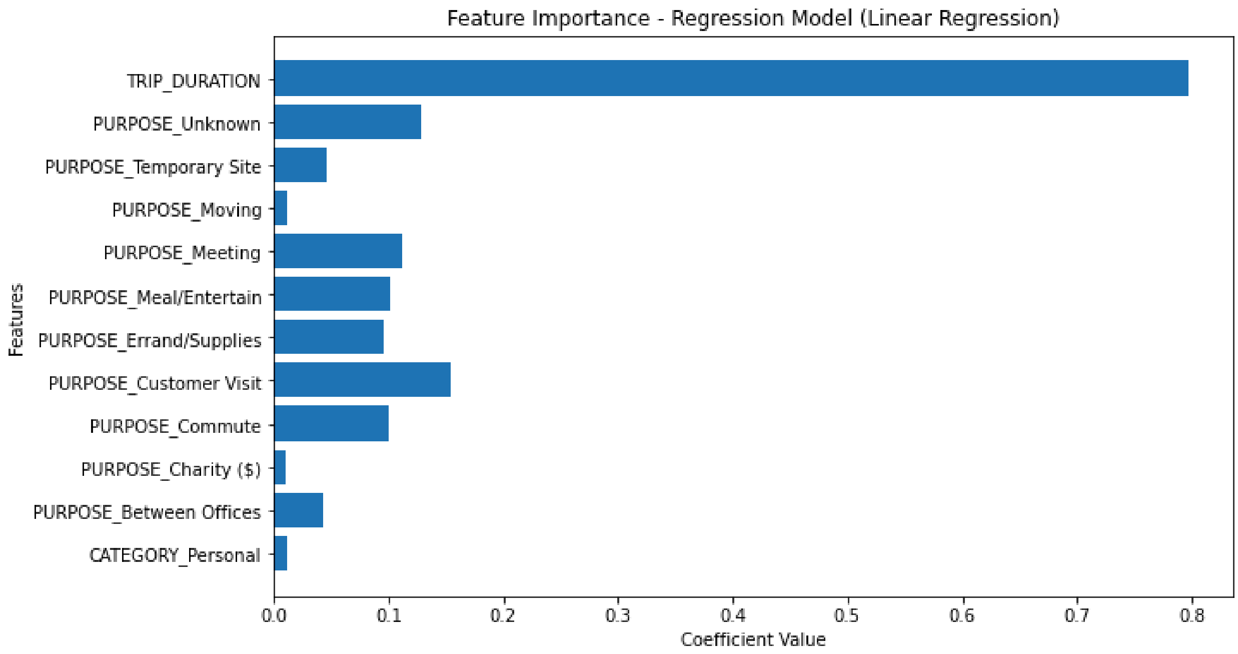
22. Feature Importance

- **Objective**: Identify the most important features in the classification and regression models.
- **Visualization**:



-

    - **Feature Importance - Logistic Regression**: The horizontal bar plot shows the coefficients for each feature in the classification model. Features with higher coefficients have a more substantial impact on predicting the trip category.

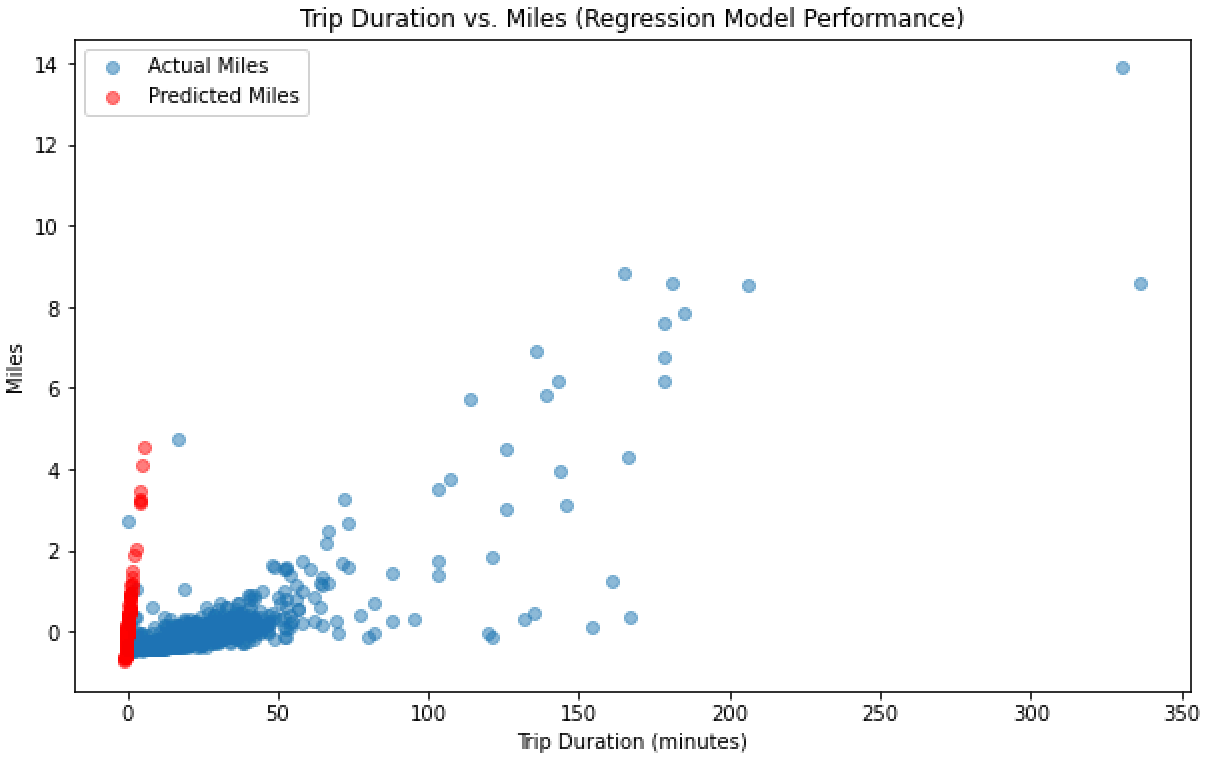


    - **Feature Importance - Linear Regression**: Another bar plot displays the coefficients of features in the regression model. Features with larger coefficients have a more significant effect on predicting `MILES`.

- **Interpretation**: In both models, certain features contribute more heavily to predictions. For example, in Logistic Regression, some purposes (like Meeting or Temporary Site) may have higher weights, indicating they're more predictive of trip category. Similarly, in the regression model, specific purposes or trip-related features may be more influential in predicting trip distance.

23. Model Performance Across Features

- **Objective**: Evaluate how the model performs with different values of key features, particularly trip duration.
- **Visualization**:



  - **Trip Duration vs. Miles (Regression Model Performance)**: A scatter plot compares actual and predicted `MILES` values based on `TRIP_DURATION`. Blue dots represent actual values, while red dots represent predictions by the regression model.
- **Interpretation**: This plot provides a visual check on the regression model's performance. A strong alignment between actual and predicted values would indicate good model performance. Here, the model seems to predict short trip distances well but may struggle with very long trips, as seen by the spread of actual values at higher trip durations.