ICS-474 Final Report

King Fahd University of Petroleum & Minerals

Department of Information and Computer Science

ICS474: Big Data Analytics

Dr. Muzammil Behzad

Team Members

Name	ID
Ali Jaber	202045080
Abdullah Al Muheef	202039000

Table of Contents

1. Data Understanding and Exploration

i. Dataset Overview

This project uses the **Uber dataset** from Kaggle, which records ride details, including starting and ending times, ride categories (e.g., Business or Personal), starting and ending locations, miles traveled, and ride purposes.

The primary goal is to analyze the data for patterns and insights. This includes identifying the most common ride purposes, highlighting high-mileage trips, and exploring relationships between variables like ride purpose, category, and travel time.

ii. Feature Description

- Before any edit to the dataset

- START_DATE: Start time of the trip (Datetime which is Categorical)
- END_DATE: End time of the trip (Datetime which is Categorical)
- CATEGORY: Type of ride, either "Business" or "Personal" (Categorical)
- START: Start location of the trip (Categorical)
- STOP: End location of the trip (Categorical)
- MILES: Distance covered in miles (Numerical)
- PURPOSE: Purpose of the trip, e.g., "Meal/Entertain," "Meeting" (Categorical)
- Target Variable: we make the model to classify and predict the CATEGORY.

- After editing the dataset

- START_HOUR: Approximation of start time (Categorical).
- START_DATE: Start time of the trip (Datetime, Categorical).
- END_DATE: End time of the trip (Datetime, Categorical).
- DURATION: Duration of the trip in minutes (Numerical).
- CATEGORY: Type of ride, either "Business" or "Personal" (Categorical).
- START: Start location of the trip (Categorical).
- STOP: End location of the trip (Categorical).
- DESTINATION: Destination type, either "In City" or "Out of City" (Categorical).
- MILES: Distance covered in miles (Numerical).
- PURPOSE: Purpose of the trip, e.g., "Meal/Entertain," "Meeting" (Categorical).

iii. Dataset Structure

The dataset consists of 1155 rows and 7 columns after cleaning. A "Total" row was excluded to maintain the integrity of the analysis.

```
Dataset Information:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1155 entries, 0 to 1154
Data columns (total 7 columns)
```

Image 3: Dataset Structure from the Code

iv. Missing Values and Duplicates

Missing values are primarily found in the PURPOSE column, accounting for a significant portion of the dataset. These missing values can affect analyses related to ride purposes. Furthermore, some values in the columns (START and STOP were specified as Unknown Location)

```
Missing Values:
 START_DATE      0
END_DATE         0
CATEGORY         0
START            0
STOP             0      Number of rows with 'Unknown Location' in START only: 148
MILES            0      Number of rows with 'Unknown Location' in STOP only: 149
PURPOSE        502      Number of rows with 'Unknown Location' in both START and STOP: 86
```

Image 4: Missing Values

While only one duplicate row was identified in rows 492 and 493.

```
Number of duplicate rows: 1

Duplicate Rows:
          START_DATE         END_DATE  CATEGORY   START  STOP  MILES  PURPOSE
492  6/28/2016 23:34  6/28/2016 23:59  Business  Durham  Cary    9.9  Meeting
```

Image 5: Duplicated Row

v. Statistical Summary

The MILES feature indicated an average trip length of approximately 10.57 miles, with a standard deviation highlighting significant variability in trip distances.

In CATEGORY, the most Frequent value is Business (1078 occurrences, dominating the dataset) while Personal has a lower occurrence (77), which indicates imbalance in the data.

START has 177 unique Locations and STOP has 188 Unique Locations. Most Frequent Location: Cary (201 occurrences for start, 203 for stop).

Statistical Summary for all Columns:

	START_DATE	END_DATE	CATEGORY	START	STOP	MILES
count	1155	1155	1155	1155	1155	1155.000000
unique	1154	1154	2	177	188	NaN
top	6/28/2016 23:34	6/28/2016 23:59	Business	Cary	Cary	NaN
freq	2	2	1078	201	203	NaN
mean	NaN	NaN	NaN	NaN	NaN	10.566840
std	NaN	NaN	NaN	NaN	NaN	21.579106
min	NaN	NaN	NaN	NaN	NaN	0.500000
25%	NaN	NaN	NaN	NaN	NaN	2.900000
50%	NaN	NaN	NaN	NaN	NaN	6.000000
75%	NaN	NaN	NaN	NaN	NaN	10.400000
max	NaN	NaN	NaN	NaN	NaN	310.300000

	PURPOSE
count	653
unique	10
top	Meeting
freq	187
mean	NaN
std	NaN
min	NaN
25%	NaN
50%	NaN
75%	NaN
max	NaN

Image 5: Statistical Summary for the Dataset

vi. Data Distribution

Histograms of the CATEGORY variable revealed that the majority of trips were classified as 'Business,' significantly outweighing other categories such as 'Personal' this distribution suggests that the dataset is predominantly composed of work-related travel. Further exploration of the CATEGORY variable might reveal trends or differences in trip lengths and frequencies across the categories. Additionally, visualizing other categorical variables alongside MILES could provide insights into potential patterns or anomalies.

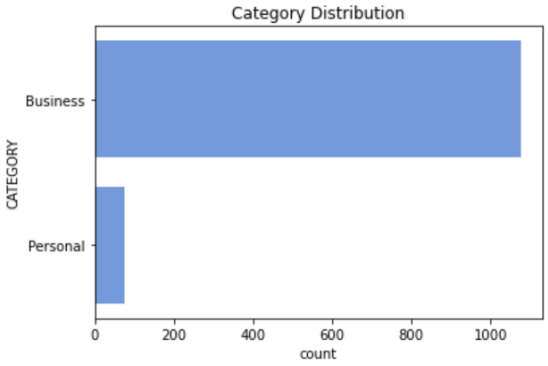


Figure 1: Category Distribution

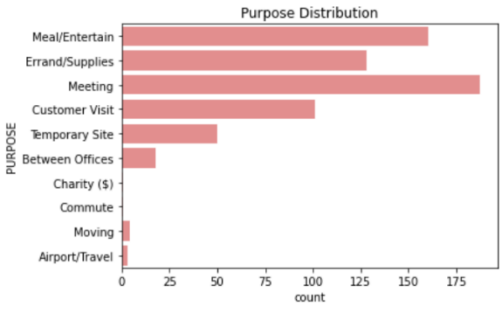


Figure 2: Purpose Distribution

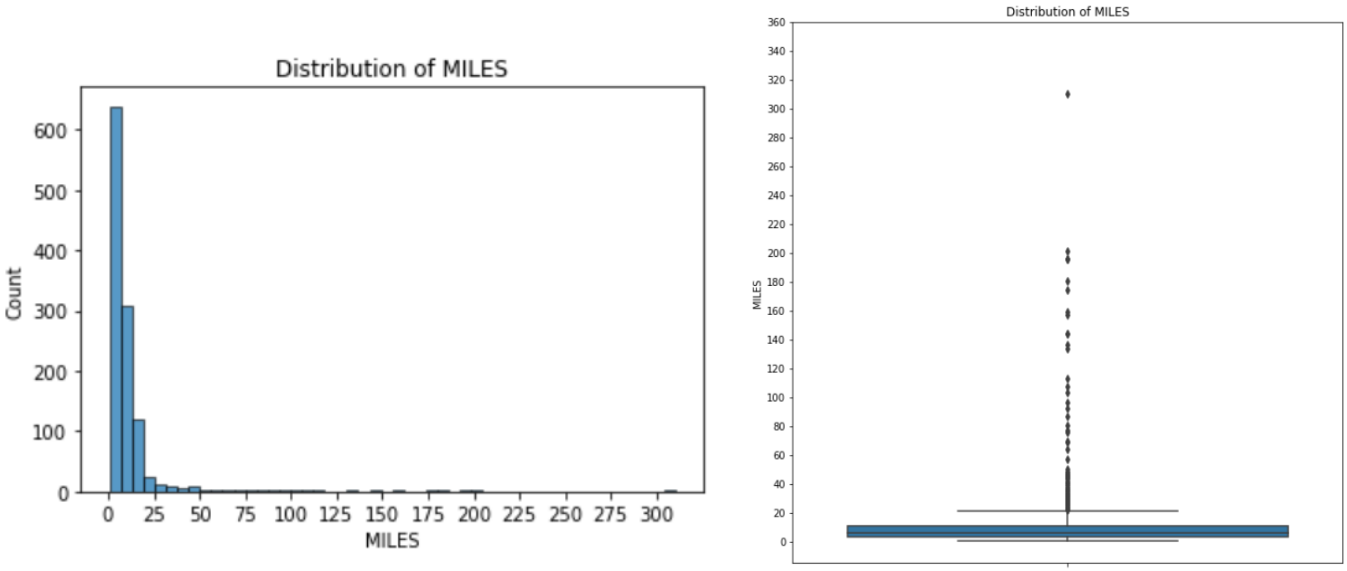


Figure 3: MILES Distribution

vii. Correlation Analysis

Since the original dataset contains only one numeric feature, performing a meaningful correlation analysis is not possible. To address this, we will encode the categorical features, enabling their inclusion in the correlation analysis. Additionally, we will create and add new features to enrich the dataset. The features we will analyze are CATEGORY (target), DURATION, MILES, PURPOSE and DESTINATION

	DURATION	CATEGORY	MILES	PURPOSE	DESTINATION
DURATION	1.0000	-0.0145	0.9160	0.1868	0.1148
CATEGORY	-0.0145	1.0000	-0.0097	-0.1560	-0.0520
MILES	0.9160	-0.0097	1.0000	0.1704	0.1362
PURPOSE	0.1868	-0.1560	0.1704	1.0000	0.1061
DESTINATION	0.1148	-0.0520	0.1362	0.1061	1.0000

Figure 4: Correlation Analysis

DURATION and MILES have a strong positive correlation (**0.9160**). This indicates that as the duration of a ride increases, the distance covered (MILES) also tends to increase.

- **CATEGORY and Other Features**:
    - **CATEGORY vs. PURPOSE**: Moderate negative correlation (**-0.1560**), suggesting that different ride purposes (e.g., "Meeting" or "Meal/Entertain") may be weakly associated with whether the ride is "Business" or "Personal."
    - **CATEGORY vs. DURATION and MILES**: Very weak negative correlations (**-0.0145** and **-0.0097**, respectively), indicating minimal relationships between the ride's length/distance and the ride type.
    - **CATEGORY vs. DESTINATION**: Weak negative correlation (**-0.0520**), suggesting that whether a ride is "Business" or "Personal" does not strongly depend on whether the destination is "In City" or "Out of City."

Insights:

- **CATEGORY is weakly correlated with most features**:
    - Features like PURPOSE show some level of association but are not strongly predictive on their own.
    - Minimal relationships with DURATION and MILES indicate that the ride's length or distance may not significantly impact the "Business" vs. "Personal" classification.

viii. Outlier Detection

We will not include the purpose to the as a feature to the model because it would reduce about half of the rows if we included. However, we include the remaining outliers such as MILES and DURATION features to reduce the overfitting due to the lack of the data.

2. Data Preprocessing

i. Handling Missing Data

As we said before, we identified two main errors in the dataset.

First, missing values were present in the PURPOSE column. These were handled by filling them with "Not Provided," ensuring that no rows were discarded and preserving the dataset's completeness. By imputing missing values, potential biases from removing incomplete rows were avoided, allowing for a more inclusive analysis.

Second, corrupted city names, such as "kar?chi," were dynamically corrected to their accurate forms using fuzzy matching. This process utilized a comprehensive dataset of global cities (worldcities.csv) to standardize entries in the START and STOP columns. The correction was automated and did not rely on a predefined list of acceptable city names, ensuring consistency and reliability across the dataset.

In addition, we have considered the Unknown locations as a missing value, which it will not be included in creating the model.

```python
import pandas as pd
from rapidfuzz import process, fuzz
from tqdm import tqdm


# Load world cities dataset with correct city names
world_cities = pd.read_csv("worldcities.csv")

# Preprocess world cities for efficient matching (removing non-alphanumeric characters and converting to lowercase)
world_cities['cleaned_city'] = world_cities['city_ascii'].apply(lambda x: ''.join(e for e in str(x) if e.isalnum()).lower())

# Function to correct city names with '?' using Levenshtein distance (fuzzy matching)
def correct_city_name(name, world_cities, max_changes=2):
    if isinstance(name, str) and '?' in name:  # Only check if '?' is in the name
        cleaned_name = ''.join(e for e in name if e.isalnum()).lower()  # Clean the city name
        if cleaned_name and cleaned_name != 'Unknown Location':  # Ignore 'Unknown Location'
            # Fuzzy match with a score threshold and maximum Levenshtein distance (max_changes)
            match = process.extractOne(cleaned_name, world_cities['cleaned_city'], scorer=fuzz.ratio)
            if match and match[1] >= 80:  # You can adjust the score threshold for accuracy
                # If Levenshtein distance is small (up to max_changes), consider it a match
                if fuzz.ratio(cleaned_name, match[0]) >= (100 - (max_changes * 10)):  # Example: 1 change -> 90%, 2 changes
                    matched_city = world_cities.loc[world_cities['cleaned_city'] == match[0], 'city_ascii'].iloc[0]
                    return matched_city
    return name  # Return original name if no match found or '?' is not in the name

# Apply the correction function with progress tracking
tqdm.pandas()
df2['START'] = df2['START'].progress_apply(lambda x: correct_city_name(x, world_cities))
df2['STOP'] = df2['STOP'].progress_apply(lambda x: correct_city_name(x, world_cities))

# Save the corrected data to a new CSV file
df2.to_csv("UberDataset_with_NewFeatures.csv", index=False)

print("City names with '?' corrected successfully!")
```

Image 6: Code for Fuzzy

ii. Encoding Categorical Variables

Encoding Techniques:

- The CATEGORY, PURPOSE, and DESTINATION columns were encoded using label encoding. Specific mappings were applied:
    - CATEGORY: {'Business': 1, 'Personal': 2}
    - PURPOSE: Multiple purpose categories mapped to numerical values (e.g., "Not Provided" = 0, "Meeting" = 3).
    - DESTINATION: {'Not Provided': 0, 'In City': 1, 'Out of City': 2}
- Label encoding was chosen because the encoded variables were used in algorithms like Decision Tree and Random Forest, which do not require one-hot encoding and can handle numerical representations of categorical data.

The unencoded features will not be used in creating the model.

	START_HOUR	START_DATE	END_DATE	DURATION	CATEGORY	START	STOP	DESTINATION	MILES	PURPOSE
0	9:15 PM	2016-01-01 21:11:00	2016-01-01 21:17:00	6.0	1	Fort Pierce	Fort Pierce	1	5.1	1
1	1:30 AM	2016-01-02 01:25:00	2016-01-02 01:37:00	12.0	1	Fort Pierce	Fort Pierce	1	5.0	0
2	8:30 PM	2016-01-02 20:25:00	2016-01-02 20:38:00	13.0	1	Fort Pierce	Fort Pierce	1	4.8	2
3	5:30 PM	2016-01-05 17:31:00	2016-01-05 17:45:00	14.0	1	Fort Pierce	Fort Pierce	1	4.7	3
4	2:45 PM	2016-01-06 14:42:00	2016-01-06 15:49:00	67.0	1	Fort Pierce	West Palm Beach	2	63.7	4

Image 6: Encoded dataset

iii. Feature Scaling

We have opted to retain the remaining outliers in features such as MILES and DURATION to mitigate the risk of overfitting, which could arise from the limited size of the dataset. Retaining these outliers ensures that the model captures the full variability

iv. Feature Selection

♦ **Selected Features**:

- DURATION: Replaces START_DATE and END_DATE to simplify temporal analysis.
- MILES: A critical numerical feature indicating trip distance, essential for predictive tasks.
- DESTINATION: Encodes whether a trip is "In City" or "Out of City," replacing redundant START and STOP columns.

- **Target Feature**:
  - CATEGORY: Encodes whether the trip is for "Business" or "Personal."
- **Excluded Features**:
  - START_DATE, END_DATE, START, STOP, and PURPOSE:
    - START_DATE and END_DATE were replaced by DURATION.
    - START and STOP were replaced by DESTINATION.
    - PURPOSE was excluded to prevent overfitting due to missing values and incomplete representation.

➢ **Justification**:

- The selected features contribute directly to the predictive task and simplify the dataset while maintaining its analytical integrity. Removing redundant or incomplete features reduces the risk of overfitting and improves model performance.

3. Modeling

i. Algorithm Selection

The goal of this task is **classification**, where the target variable CATEGORY indicates whether the trip is for "Business" or "Personal." Based on this requirement, two algorithms were selected:

➢ **Decision Tree Classifier**:

- This algorithm is easy to interpret and visualize, making it suitable for explaining the results to non-technical stakeholders.
- Decision Trees can handle both numerical and categorical data directly, without requiring one-hot encoding for categorical variables.
- It was initially chosen for its simplicity and ability to model complex decision boundaries.

➢ **Random Forest Classifier**:

- Random Forest, an ensemble method, was chosen to address the overfitting tendencies of Decision Trees.
- It combines predictions from multiple Decision Trees, increasing robustness and reducing variance.

- Random Forest is particularly effective for datasets with mixed data types (numerical and categorical) and can handle missing data better.

Decision Trees provided a baseline, while Random Forest was used for better generalization and accuracy due to its ensemble approach. Also, both algorithms are non-parametric and can adapt to the dataset without strict assumptions about the data distribution.

ii. Data Splitting

- The data was divided using the **hold-out method**, allocating 70% of the data for training and 30% for testing, ensuring that the model is trained on the majority of the data while reserving an independent set for evaluation.
- **Implementation**: This was done using the **train_test_split** function from scikit-learn, with a random state set to ensure reproducibility.

iii. Model Training

➢ **Decision Tree**:

- Trained using the entropy criterion for splitting nodes, which measures the information gain. The default hyperparameters were initially used, followed by fine-tuning (max_depth, min_samples_split, min_samples_leaf) using GridSearchCV.

➢ **Random Forest**:

- Trained with 100 estimators (trees) and default hyperparameters. Hyperparameter tuning was later applied for parameters like n_estimators, max_depth, and min_samples_split.

➢ **Training Process**:

- Models were fitted on the training data (X_train, y_train) using the scikit-learn fit() method.
- GridSearchCV was employed for hyperparameter optimization, ensuring the best parameter settings for improved performance.

iv. Model Evaluation

- **Accuracy**: Measures the proportion of correctly predicted instances out of total instances. Suitable for balanced datasets.
- **Precision and Recall**:
  - Precision: Focuses on the ratio of true positives to predicted positives.
  - Recall: Focuses on the ratio of true positives to actual positives.
  - These are essential for imbalanced datasets or tasks where specific classes are more critical.
- **F1 Score**: The harmonic mean of precision and recall, used as the primary metric to balance the trade-off between false positives and false negatives. It is particularly suitable for imbalanced datasets.

For this analysis, F1 Score is suitable choice because it Ensures a good balance between precision and recall. However, we will provide the other evaluation metrics in the report.

- **Implementation**
  - Metrics were calculated using scikit-learn functions (accuracy_score, precision_score, recall_score, f1_score) on the testing set.

v. Performance Analysis

- **Decision Tree**:
  - Accuracy: Moderate performance with overfitting observed due to the tree's depth.
  - F1 Score: Highlighted imbalances in predictions for some classes.

```
Decision Tree:

Accuracy:= 0.8697183098591549
Confusion Matrix:=
 [[245  18]
 [ 19   2]]
Precision: 0.928030303030303
Recall: 0.9315589353612167
F1 Score: 0.9297912713472486
```

Image 7: Decision Tree Performance

- **Random Forest**:

- Outperformed the Decision Tree on all metrics due to its ensemble approach.
- F1 Score was consistently higher, indicating better handling of class imbalances and more reliable predictions.

```
RandomForest:
Accuracy: 0.9014084507042254
Precision: 0.9272727272727272
Recall: 0.9695817490494296
F1 Score: 0.9479553903345725
```

Image 8: RandomForest Performance

vi. Model Improvement

➢ **Hyperparameter Tuning**:

- Applied to both models using GridSearchCV to identify optimal settings for max_depth, min_samples_split, and n_estimators.

➢ **Feature Engineering**:

- Removed redundant features (e.g., START and STOP) and replaced them with DESTINATION.

➢ **Algorithm Selection**:

- Switching to Random Forest from Decision Tree significantly improved generalizability.

```
Random Forest:
Accuracy: 0.926056338028169
Precision: 0.926056338028169
Recall: 1.0
F1 Score: 0.9616087751371115
```

Image 9: Decision Tree Performance with Hyperparameter Tuning

vii. Validation

- **K-Fold Cross-Validation**:
  - Used with 5 folds to ensure stability and robustness across different subsets of data.
  - Metrics (F1 Score, Precision, Recall) were averaged across folds to assess model consistency.
- **Rationale**:
  - Cross-validation mitigates the risk of overfitting by testing on multiple partitions of the dataset.

```
RandomForest:
Cross-Validation F1 Scores: [0.962 0.959 0.959 0.959 0.964]
Average F1 Score: 0.96
Cross-Validation Precision Scores: [0.926 0.921 0.921 0.921 0.93 ]
Average Precision Score: 0.924
Cross-Validation Recall Scores: [1. 1. 1. 1. 1.]
Average Recall Score: 1.0
```

Image 10: RandomForest Validation

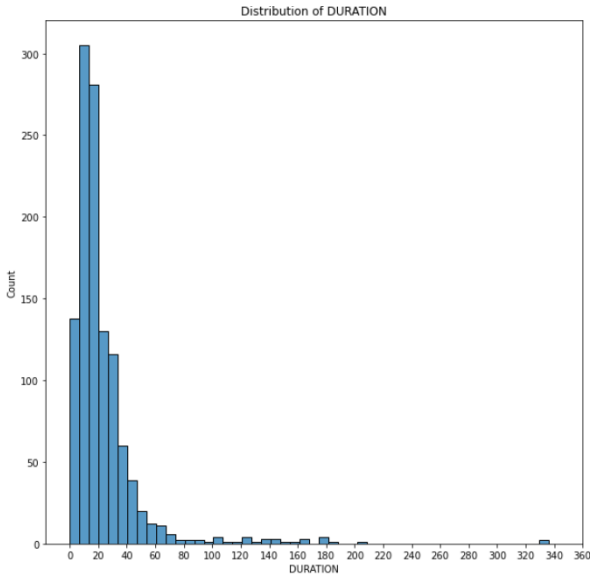
viii. Final Model Selection

- **Selected Model**: Random Forest Classifier.
- **Reasoning**:
  - Consistently outperformed the Decision Tree in terms of F1 Score, Precision, and Recall.
  - Reduced overfitting due to its ensemble nature.
  - Provided better generalization and robustness, making it more suitable for real-world deployment.

4. Visualization

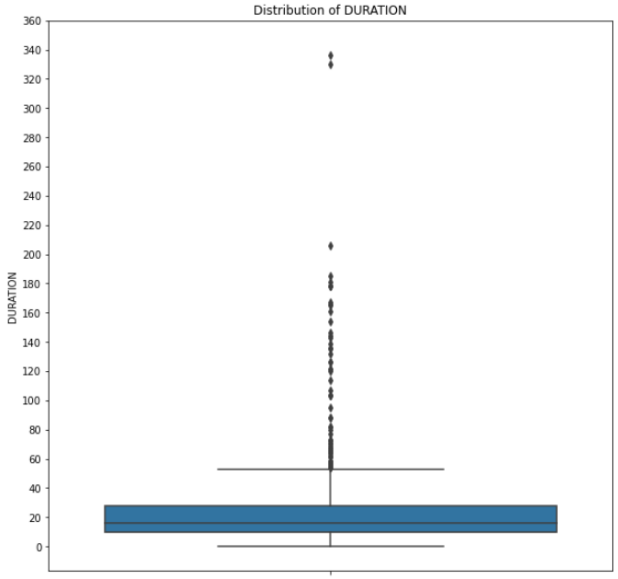
i. Data Distribution

- **Histogram of DURATION (Numeric):**
  - **Observations:**
    - The majority of rides have a short duration, with most being under 40 minutes.
    - The distribution is heavily right-skewed, with some rides having durations significantly longer than the majority (e.g., above 200 minutes).
  - **Implications:**
    - The skewness indicates that the dataset contains many short-duration rides but also a few extreme outliers.

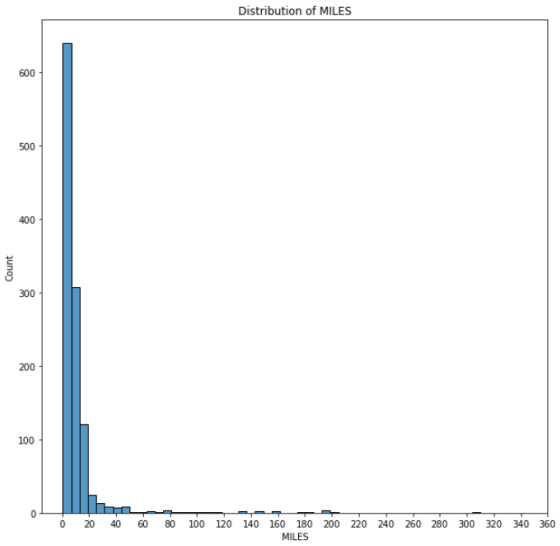


- **Boxplot of DURATION (Numeric):**
  - **Observations:**
    - Most of the rides fall within a tight range of durations (up to approximately 60 minutes).
    - A substantial number of outliers exist above this range, with extreme values reaching as high as 340 minutes.

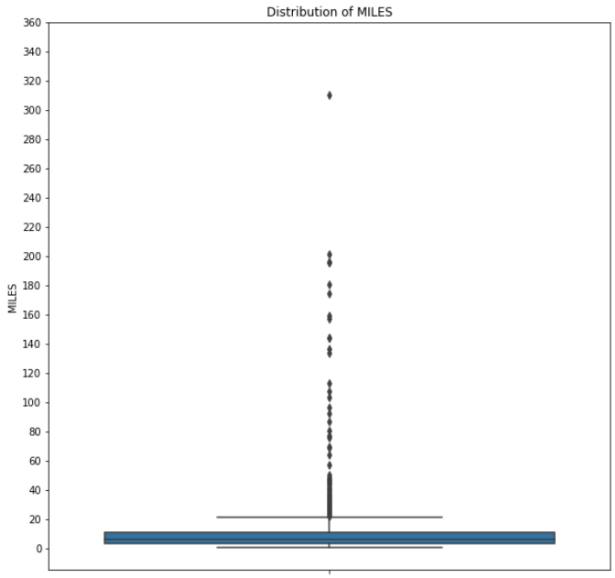
- **Implications:**
    - The outliers could indicate rare long-distance rides.



- **Histogram of MILES (Numeric):**
    - **Observations:**
        - The majority of rides are short (under 20 miles), but a long tail exists with rides exceeding 100 miles.
        - Similar to DURATION, the distribution is heavily right-skewed, with extreme values observed.
    - **Implications:**
        - Like DURATION, the skewness indicates that the dataset contains many short-distance rides but also a few extreme outliers.

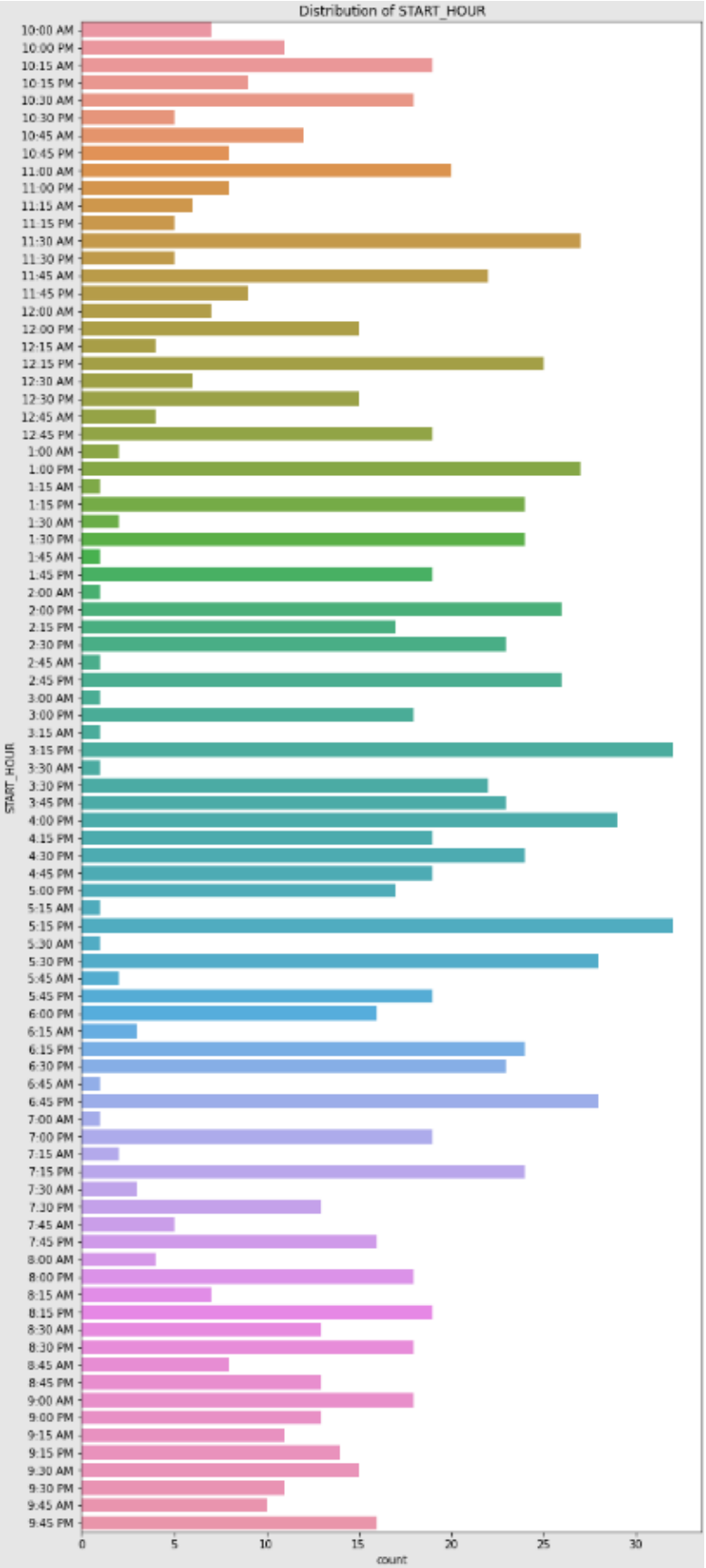


- **Boxplot of MILES (Numeric):**
  - **Observations**:
    - The majority of rides have distances concentrated below 20 miles, as indicated by the dense grouping within the box.
    - There are numerous **outliers** extending beyond the whiskers, with some rides exceeding 300 miles. These outliers represent long-distance trips and appear as individual points above the boxplot.

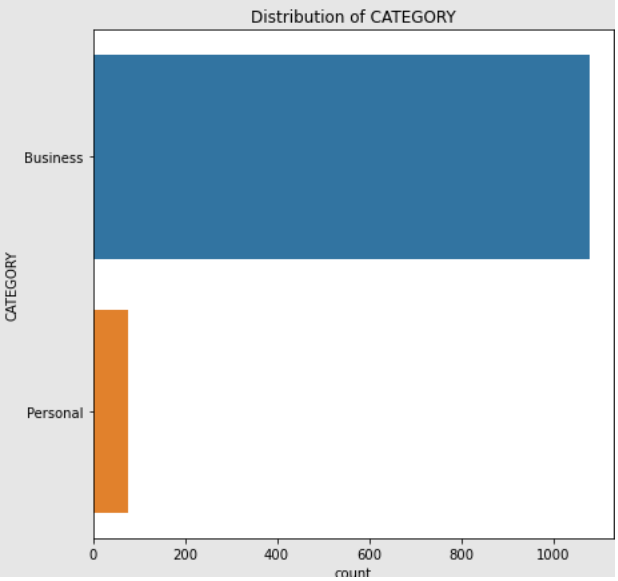


- **Distribution of START_HOUR:**
  - **Observations:**
    - Ride start times are distributed fairly evenly across different hours, with a slight increase during typical commuting hours (e.g., 7–9 AM and 4–6 PM).

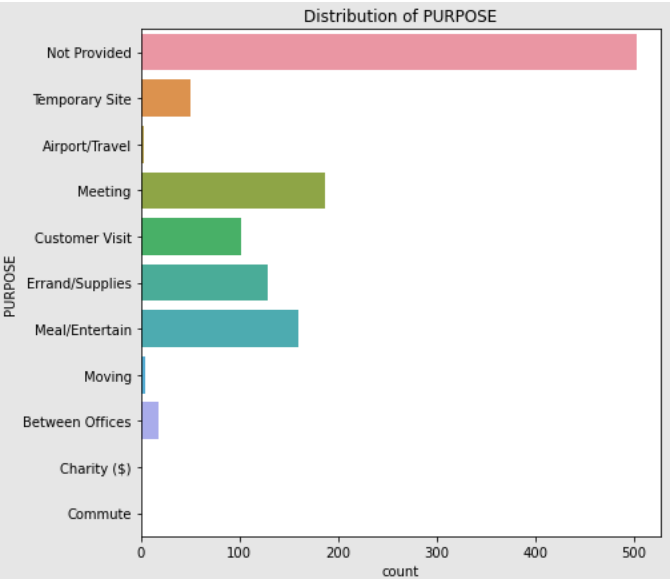
- **Bar Plot of CATEGORY:**
  - **Observations:**
    - The dataset is dominated by "Business" rides, with "Personal" rides forming a much smaller proportion.


Distribution of START_HOUR
START_HOUR
10:00 AM
10:00 PM
10:15 AM
10:15 PM
10:30 AM
10:30 PM
10:45 AM
10:45 PM
11:00 AM
11:00 PM
11:15 AM
11:15 PM
11:30 AM
11:30 PM
11:45 AM
11:45 PM
12:00 AM
12:00 PM
12:15 AM
12:15 PM
12:30 AM
12:30 PM
12:45 AM
12:45 PM
1:00 AM
1:00 PM
1:15 AM
1:15 PM
1:30 AM
1:30 PM
1:45 AM
1:45 PM
2:00 AM
2:00 PM
2:15 PM
2:30 PM
2:45 AM
2:45 PM
3:00 AM
3:00 PM
3:15 AM
3:15 PM
3:30 AM
3:30 PM
3:45 PM
4:00 PM
4:15 PM
4:30 PM
4:45 PM
5:00 PM
5:15 AM
5:15 PM
5:30 AM
5:30 PM
5:45 AM
5:45 PM
6:00 PM
6:15 AM
6:15 PM
6:30 PM
6:45 AM
6:45 PM
7:00 AM
7:00 PM
7:15 AM
7:15 PM
7:30 AM
7:30 PM
7:45 AM
7:45 PM
8:00 AM
8:00 PM
8:15 AM
8:15 PM
8:30 AM
8:30 PM
8:45 AM
8:45 PM
9:00 AM
9:00 PM
9:15 AM
9:15 PM
9:30 AM
9:30 PM
9:45 AM
9:45 PM
0
5
10
15
20
25
30
count

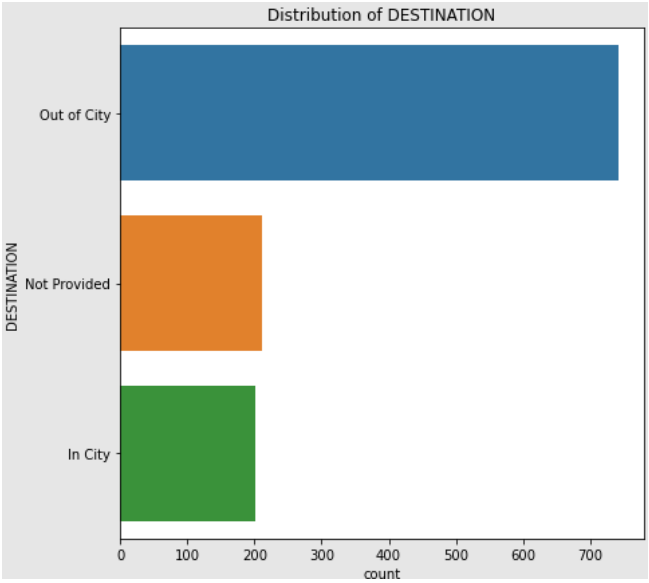
- **Implications:**
    - This class imbalance could affect model predictions, as the model may favor the majority category (Business).



- **Bar Plot of PURPOSE:**
    - **Observations:**
        - The most frequent purposes are "Meeting" and "Meal/Entertain," while other purposes like "Charity" or "Commute" have very few occurrences.
        - A large portion of entries has the purpose "Not Provided."



- **Bar Plot of DESTINATION:**
  - **Observations:**
    - "Out of City" destinations are the most frequent, followed by "In City." A significant number of rides have "Not Provided" for the destination.



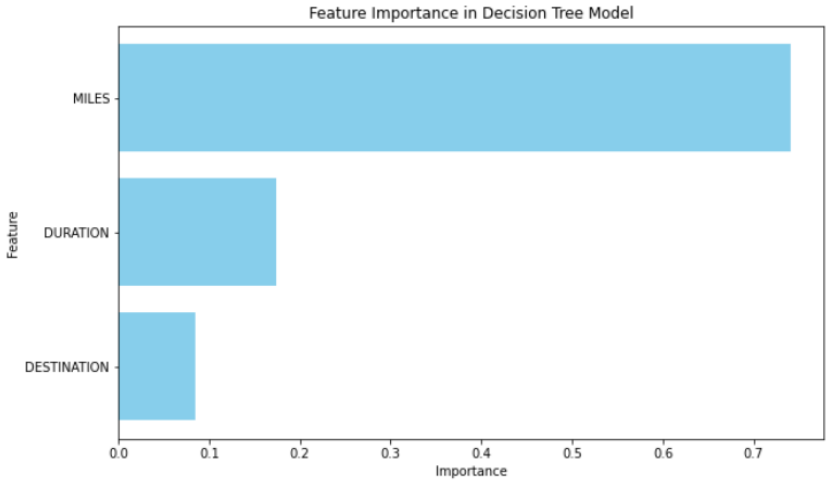
ii. Feature Importance

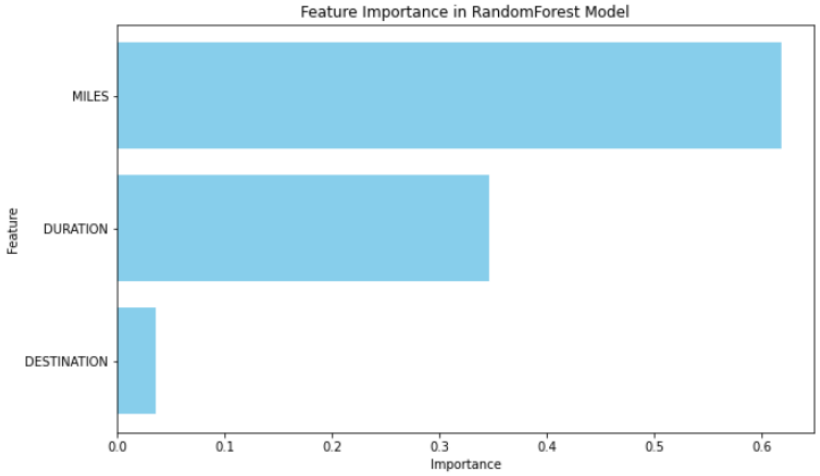
- **Most Important Feature**: MILES
  - MILES is the most influential feature in both models, emphasizing the distance as a key factor for classification.
- **Second Most Important Feature**: DURATION
  - DURATION plays a significant role, though slightly less impactful than MILES.
- **Least Important Feature**: DESTINATION
  - DESTINATION contributes minimally to the models' predictions.

Comparison Between Decision tree and RandomForest Models:

- Both models identify MILES as the most critical feature, followed by DURATION. This consistency highlights the importance of these features in classifying the "Business" vs. "Personal" categories.

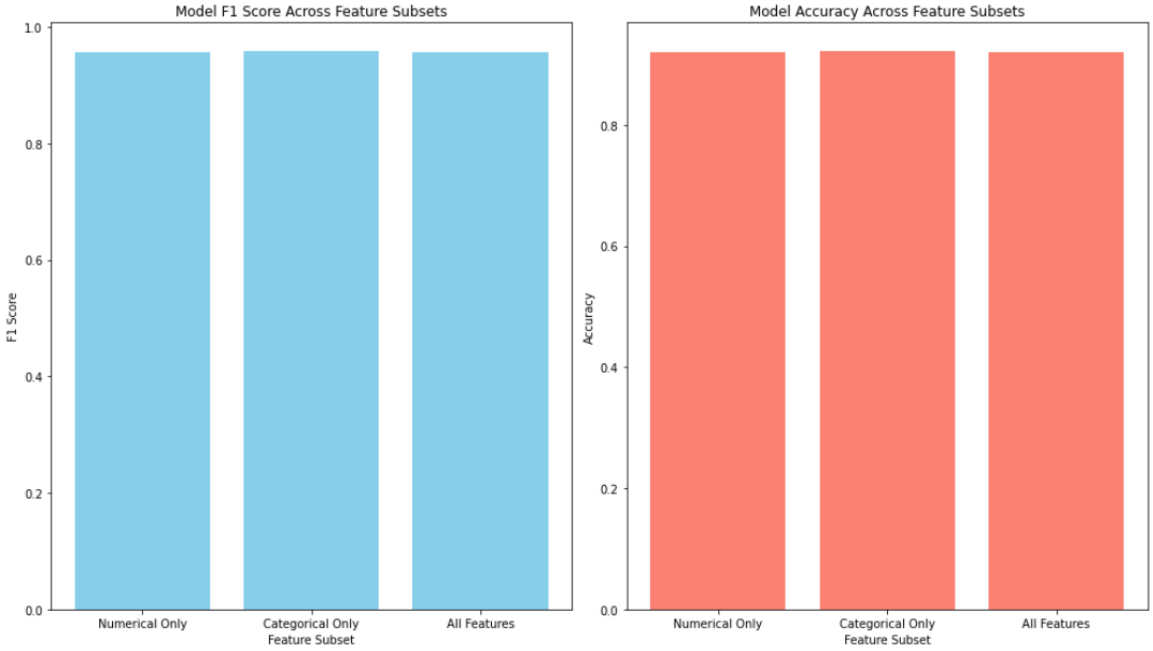
- DESTINATION is the least impactful feature in both models, suggesting that it could potentially be excluded in future iterations without significantly affecting performance.





iii. Model Performance Across Features

In this section we perform the model performance on our features for F1 Score and the Accuracy, and we found that:



- **F1 Score Across Feature Subsets**:
    - The F1 Score remains consistent across all feature subsets (Numerical Only, Categorical Only, and All Features).
    - **Implications**:
        - This suggests that either numerical or categorical features alone can provide sufficient predictive power for classifying CATEGORY.
        - Adding all features (numerical + categorical) does not improve performance, indicating potential redundancy or lack of complementary information between these feature types.
- **Accuracy Across Feature Subsets:**
    - Similar to the F1 Score, the accuracy is consistent across all feature subsets.
    - Implications:
        - The model performs equally well regardless of the subset of features used.
        - It suggests that the model's ability to classify CATEGORY is not heavily reliant on the combination of features, but rather on the

individual predictive power of key features (e.g., MILES as identified in feature importance).

- **Overall Observations:**
  - The consistency of F1 Score and Accuracy implies that the model is robust and performs well without requiring all available features.
- **Recommendations:**
  - **Feature Selection**:
    - Focus on numerical features (e.g., MILES and DURATION), as they alone yield similar performance.
    - The categorical feature (DESTINATION) may be less impactful and could be excluded for a simpler model.

5. Limitations

Small Dataset:

- The dataset contains only 1,155 entries, which limits the generalizability of any insights or predictive models developed.
- Small datasets are prone to overfitting, especially for complex models like Random Forests, as the model may capture noise rather than actual patterns.
- With such a small dataset, splitting into training and testing sets further reduces the data available for training, potentially impacting model performance.

High Number of Missing Values:

- **PURPOSE Feature**: Nearly half of the entries in the PURPOSE column are missing, reducing the utility of this feature in predictive modeling.
- Other features, such as START and STOP, also have missing values, though to a lesser extent.
- Removing rows with missing values significantly reduces the dataset size, making the already small dataset even smaller.

Presence of Outliers:

- Numerical features like MILES and DURATION contain significant outliers. For instance, some rides exceed 300 miles or last several hours, while the majority are much shorter.

- Removing these outliers would further shrink the dataset, exacerbating the issue of insufficient data.
- Retaining outliers, on the other hand, could skew model predictions and reduce performance for the majority of "normal" cases.

Class Imbalance:

- The CATEGORY feature is heavily imbalanced, with "Business" rides vastly outnumbering "Personal" rides. This imbalance makes it challenging for models to accurately classify minority instances, often leading to biased predictions favoring the majority class.
- This imbalance also limits the ability to derive meaningful insights for underrepresented categories.

Lack of Complementary Information:

- While some features (e.g., MILES and DURATION) show strong predictive power, others like DESTINATION and PURPOSE contribute minimally. This reduces the overall richness and utility of the dataset for prediction tasks.