
**Faculty of Engineering**

**Computer Engineering Department**

# BigData report

## Supervised by

Dr. Lydia Wahid

## Presented by

Yousef Mostafa Elmahdy

Gaser Ashraf Fayez

Mohamed Salama

Ahmed Waleed

2023

# Problem definition:

the problem definition is to predict the duration of each taxi trip in the test set based on individual trip attributes.

# DataSets:

The competition dataset is based on the 2016 NYC Yellow Cab trip record data.
https://www.kaggle.com/competitions/nyc-taxi-trip-duration/data

# Data fields:

- **id** - a unique identifier for each trip
- **vendor_id** - a code indicating the provider associated with the trip record
- **pickup_datetime** - date and time when the meter was engaged
- **dropoff_datetime** - date and time when the meter was disengaged
- **passenger_count** - the number of passengers in the vehicle (driver entered value)
- **pickup_longitude** - the longitude where the meter was engaged
- **pickup_latitude** - the latitude where the meter was engaged
- **dropoff_longitude** - the longitude where the meter was disengaged
- **dropoff_latitude** - the latitude where the meter was disengaged
- **store_and_fwd_flag** - This flag indicates whether the trip record was held in vehicle memory before sending to the vendor because the vehicle did not have a connection to the server - Y=store and forward; N=not a store and forward trip
- **trip_duration** - duration of the trip in seconds

# Project pipeline:

## 1- Data Preprocessing:

Before building any model, it's important to preprocess the data and clean it as much as possible to ensure that the model is trained on high-quality data. Here are the steps we took to preprocess the data:

**a. Handling Missing Values:**

We started by checking for any missing values in the dataset. Fortunately, there were no missing values in any of the columns, so we moved on to the next step.

**b. Removing Rows with Missing Data:**

Next, we removed any rows with missing data. Although we didn't find any missing data in the first step, it's always good to double-check and make sure that the dataset is clean.

**c. Handling Datetime:**

We converted the `pickup_datetime` and `dropoff_datetime` columns to datetime format using Python's `datetime` module. This allowed us to extract additional features from the datetime column such as month, day, week, and hour. These features can be used by the model to capture any seasonality or patterns in the data.

**d. Handling Passenger Count:**

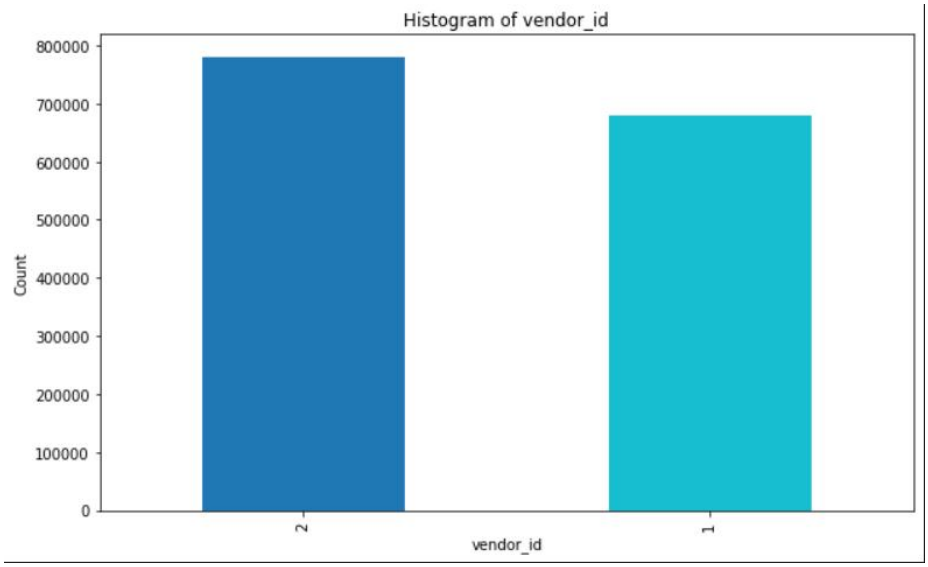
We noticed that there were trips with zero passengers, which doesn't make sense. We decided to remove any trips with zero passengers from the dataset. there were 17 trips with missing passenger count, so we removed those.

**e. Handling Trip Duration:**

Finally, we looked at the `trip_duration` column and noticed that there were some outliers. We decided to remove trips with a duration of less than 1 minute or more than 3 hours. This helped us to remove any trips that were too short or too long to be considered as part of the normal distribution of trip durations.

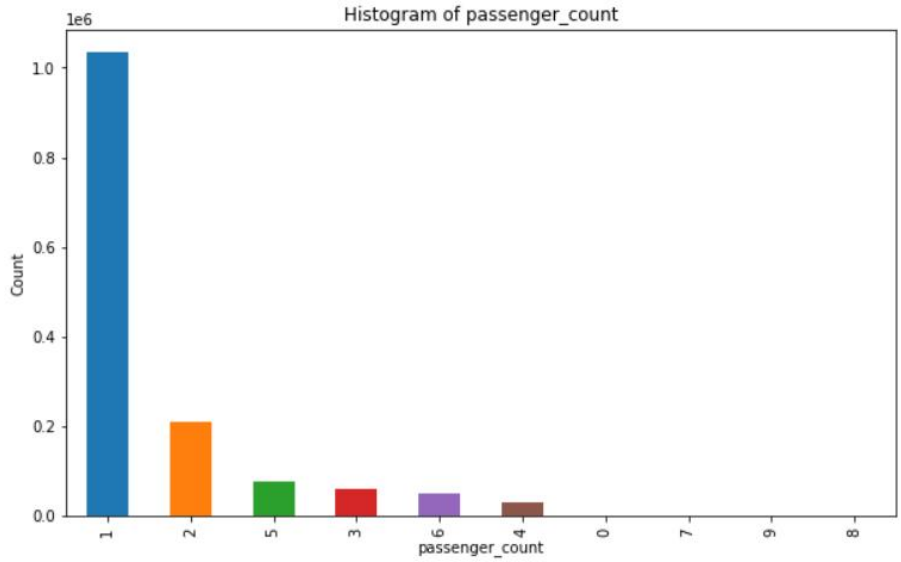
## 2- ▪ Data visualization:

## Vendor id:

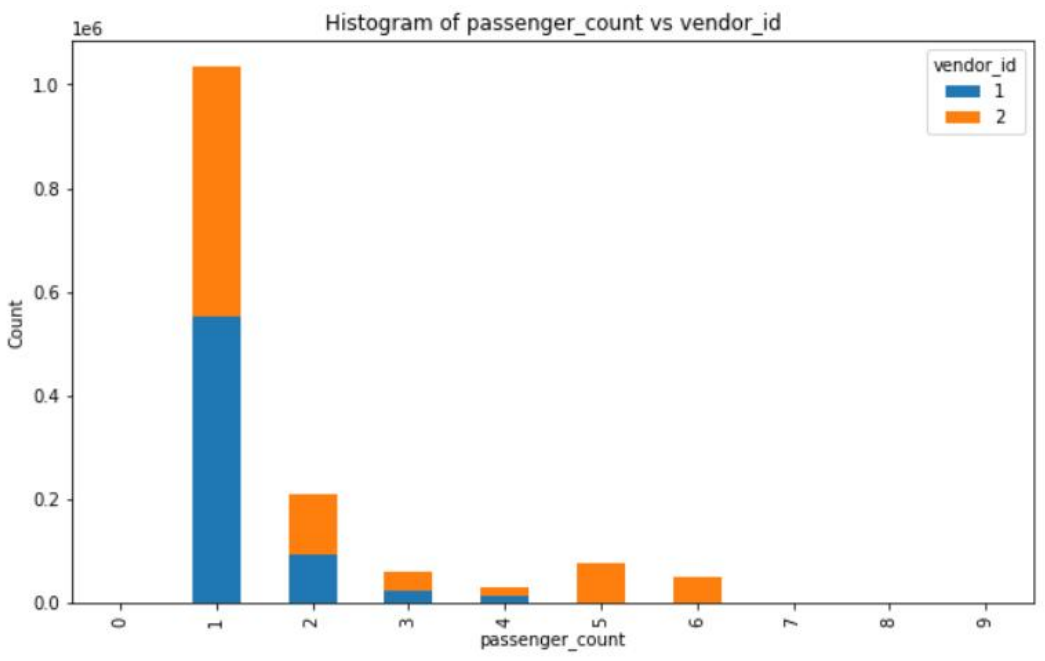



- there are only two vendors in the data set, 1 and 2
- vendor id 2 has more trips than vendor id 1

## Passenger count:

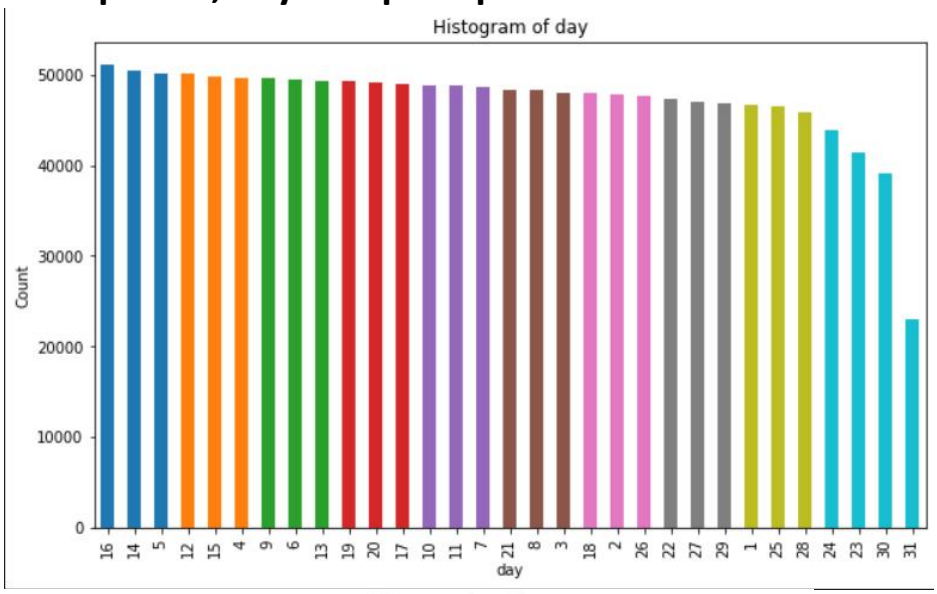


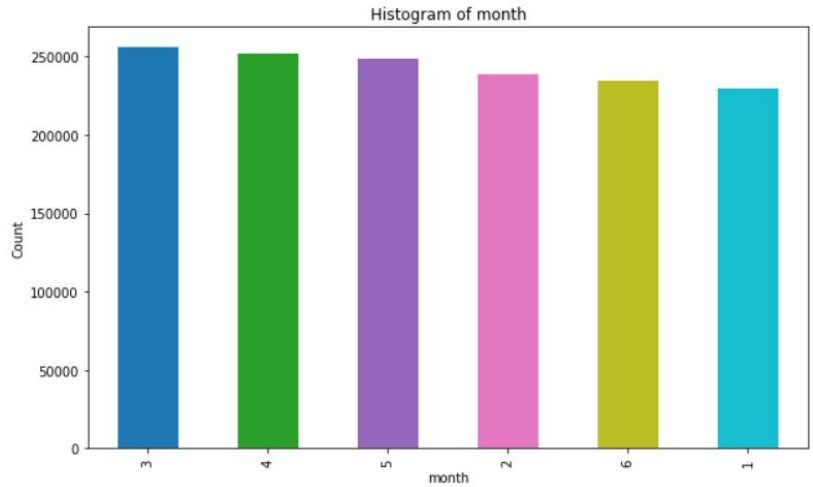

- Most of the trips have 1 or 2 passengers and there are some trips with 0 passengers

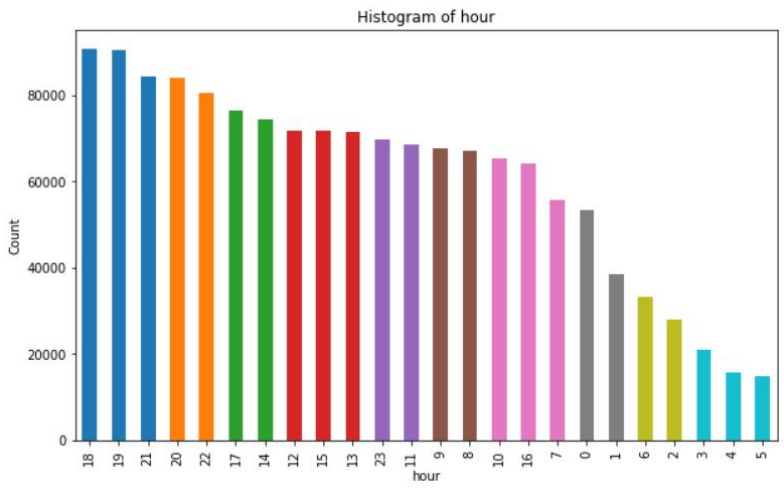



- Vendor id 2 has more trips with 5,6 passengers than vendor id 1 so vendor id 2 has more seats than vendor id 1 (big cars)

## Pickup hour, day and pickup month:

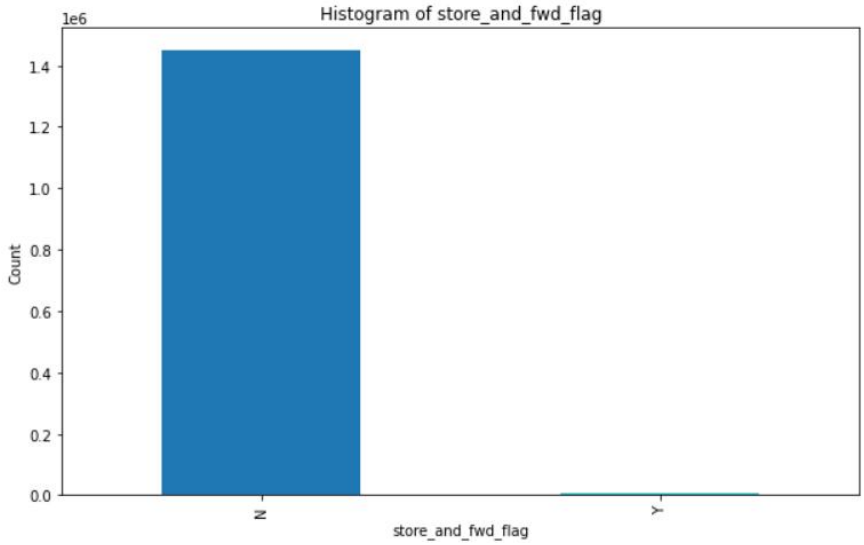


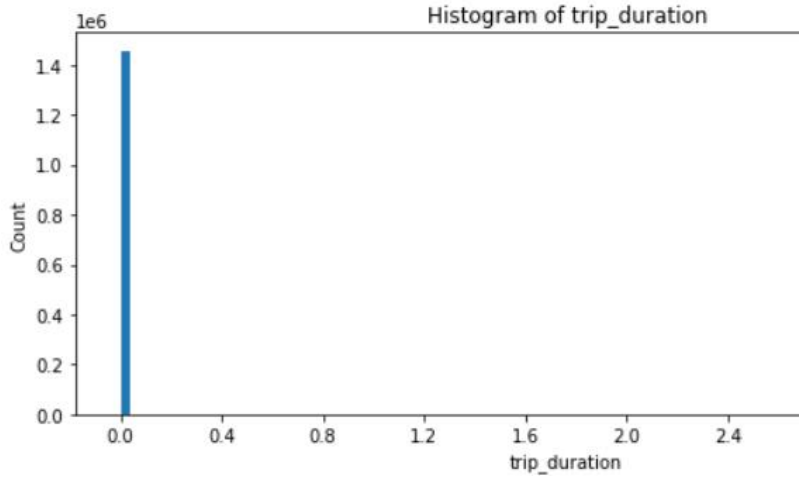

- Most hours of the day having trips are 6,7 PM

## Store and forward flag:

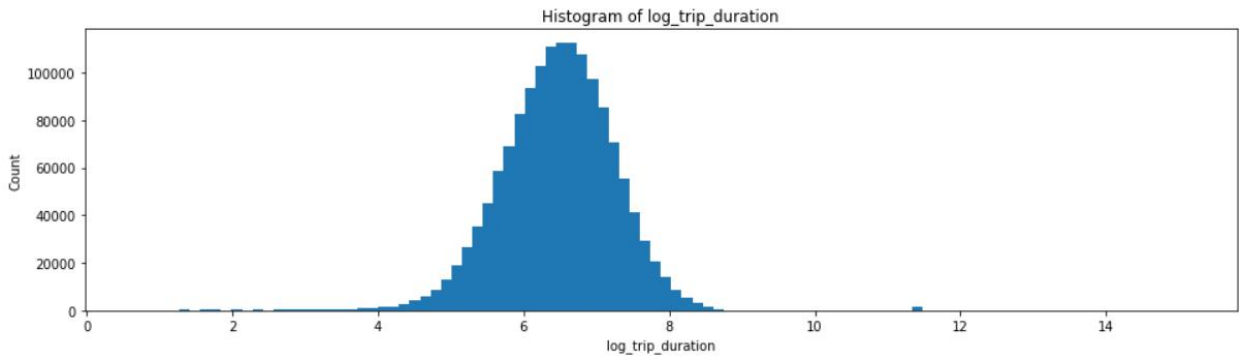



- Most of the trips are not stored and forwarded In other words, there was a direct connection between the vehicle and the server, allowing the trip data to be sent without the need for temporary storage.

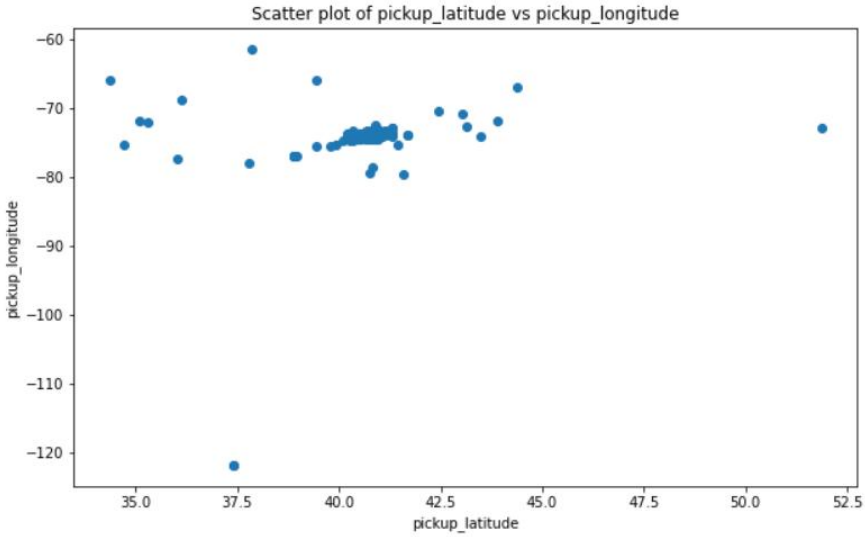
## Trip duration:




- Because most of the trips are short, the trip duration is short too so log(trip duration) is better visualization



## Pickup latitude & longitude:




- Some points are outside the main region of the data set so it is better to remove them (outliers)
- Main region is between 37.5 and 45 for latitude and -65 and -80 for longitude

## 3- Extracting insights from data:

### a. Distance features:

We extracted three distance features - Haversine distance, Manhattan distance, and direction. The Haversine distance is the great-circle distance between two points on a sphere given their longitudes and latitudes. The Manhattan distance is the distance between two points measured along the axes at right angles, i.e., the sum of the absolute differences of their coordinates. The direction is the angle between the vector connecting the pickup and dropoff points and the east-west axis.

### b. Categorical features:

We also extracted several categorical features - vendor ID, passenger count, store and forward flag, day, hour, month, and week. Vendor ID indicates which taxi company provided the ride, while passenger count indicates the number of passengers in the ride. The store and forward flag indicates whether the trip data was held in vehicle memory before sending to the vendor, and day, hour, month, and week indicate the date and time of the pickup.

### c. Encoding categorical data:

To use these categorical features in our machine learning models, we applied one-hot encoding. This involves converting categorical variables into binary vectors, where each vector corresponds to a single category. For example, the vendor ID column, which contained two categories - 1 and 2 - was split into two separate columns, with a binary 1 indicating the presence of that category and 0 indicating the absence of that category. Example in shown figure.

| month_4 | month_5 | month_6 | week_0 | week_1 | week_2 | week_3 | week_4 | week_5 | week_6 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

By extracting these features and encoding the categorical data, we were able to create a richer dataset that provided valuable insights into the factors that influence taxi trip duration.

## 4- Models:

### a. Linear regression:

A simple yet powerful machine learning algorithm that can be used to predict a continuous value from a set of features. It works by finding a linear relationship between the features and the target variable.

### b. Decision tree regression:

Another simple machine learning algorithm that can be used to predict a continuous value from a set of features. It works by building a decision tree, which is a series of yes/no questions that are used to classify the data.

### c. Random forest regression:

More complex machine learning algorithm that can be used to predict a continuous value from a set of features. It works by building a number of decision trees and then combining their predictions to get a more accurate result.

### d. XGBoost regression:

A state-of-the-art machine learning algorithm that can be used to predict a continuous value from a set of features. It works by combining decision trees with a number of other techniques to get a very accurate result.

# Results and Evaluation:

**1. Linear Regression:** This model fits a linear function to the training data, trying to find the best weights for each feature to predict the target variable. The `maxIter` parameter sets the maximum number of iterations the algorithm should run to find these weights. The `regParam` and `elasticNetParam` are regularization parameters that help prevent overfitting. The root mean squared error (RMSE) of this model on the test data is 414.42, and the R-squared score is 0.601.

**2. Decision Tree Regression:** This model builds a decision tree based on the features of the training data, splitting on the feature that maximizes the reduction in variance of the target variable. The `maxDepth` parameter sets the maximum depth of the decision tree. The RMSE of this model on the test data is 375.42, and the R-squared score is 0.601.

**3. Random Forest Regression:** This model is an ensemble of decision trees, where each tree is built on a random subset of the features and a random subset of the training data. The `numTrees` parameter sets the number of trees in the forest. The RMSE of this model on the test data is 385.52, and the R-squared score is 0.587.

**4. XGBoost Regression:** This model is a gradient boosting model that uses a more efficient tree-building algorithm than the previous models. The `maxDepth` parameter sets the maximum depth of the trees, and the `stepSize` parameter controls the learning rate. The RMSE of this model on the test data is 337.25, and the R-squared score is 0.736.

**5. XGBoost Regression without maxdepth (no limitations):** This model is a gradient boosting model that uses a more efficient tree-building algorithm than the previous models. The `maxDepth` parameter sets the maximum depth of the trees, and the `stepSize` parameter controls the learning rate. The RMSE of this model on the test data is 316.57, and the R-squared score is 0.79.

Based on the previous information, it seems that XGBoost Regression outperforms the other models, achieving the lowest RMSE and the highest R-squared score. This suggests that XGBoost Regression is able to capture the underlying patterns in the data more accurately than the other models. Furthermore, the performance of the XGBoost model appears to be enhanced by removing the maxDepth limitation, resulting in even better predictions.

However, it is important to note that the choice of the best model can depend on the specific problem and dataset being analyzed. In addition, other factors such as interpretability, computational efficiency, and scalability may also need to be considered when choosing a model.

**Addition information that should be provided:**
RMSE and R-squared are commonly used metrics to evaluate the performance of regression models.

RMSE stands for Root Mean Squared Error, which measures the average distance between the predicted values and the actual values. It calculates the square root of the average of the squared differences between the predicted and actual values. The lower the RMSE value, the better the performance of the model, as it indicates that the model is able to predict the target variable with greater accuracy.

R-squared ($R^2$) is a statistical measure that indicates the proportion of variance in the target variable that is explained by the independent variables in the model. It ranges from 0 to 1, where 0 indicates that the model does not explain any of the variance in the target variable, and 1 indicates that the model explains all of the variance in the target variable. A higher R-squared value indicates a better fit of the model to the data, as it suggests that a larger proportion of the variance in the target variable is explained by the independent variables.

Overall, both RMSE and R-squared are useful metrics to evaluate the performance of regression models, but they provide different information about the accuracy and fit of the model. It is common to use a combination of these metrics to select the best model for a specific problem.

# future work:

Moving onto future work, there are a few things that could be explored to improve the model and further the analysis:

**1. Feature engineering:** Although we have extracted and engineered several features in this project, there could be more useful features that could be generated from the existing data or external sources.

**2. Hyperparameter tuning:** While we have tuned some of the important hyperparameters for the models, further tuning could be done to find the optimal set of hyperparameters for each model, which could further improve the performance.

**3. More data:** The dataset used in this project only includes data from a limited time period. Collecting and incorporating more data could help improve the model's performance and capture more general patterns.

**4. Real-time prediction:** Deploying the model for real-time prediction could be explored, which could be useful for predicting time duration for upcoming trips and providing recommendations to users.