Trip Duration Prediction Project Report

1 Introduction :

The NYC Taxi Duration Prediction competition on Kaggle challenges participants to build a model that predicts the total ride duration of taxi trips in New York City. The primary dataset is provided by the NYC Taxi and Limousine Commission and includes information such as pickup time, geo-coordinates, number of pas sengers, and other variables

2 understand my dataset :

2.1 definition my dataset

- **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

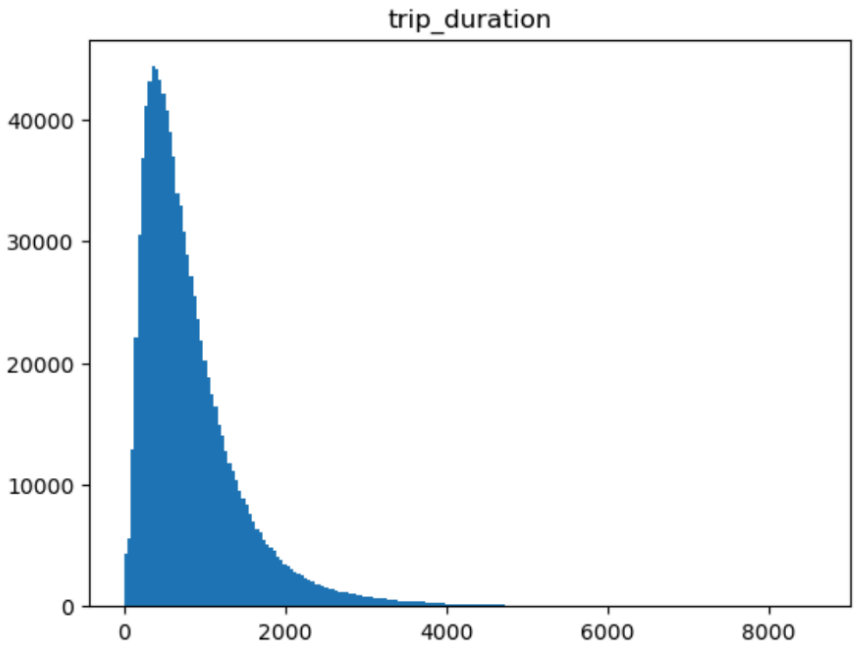
- **store_and_fwd_flag** - This flag indicates whether the trip record was held in vehicle memory before sending to the vendor because the vehicle did not have a connection to the server - Y=store and forward; N=not a store and forward trip
- **trip_duration** - duration of the trip in second

2.2 Target Variable (Trip Duration) :

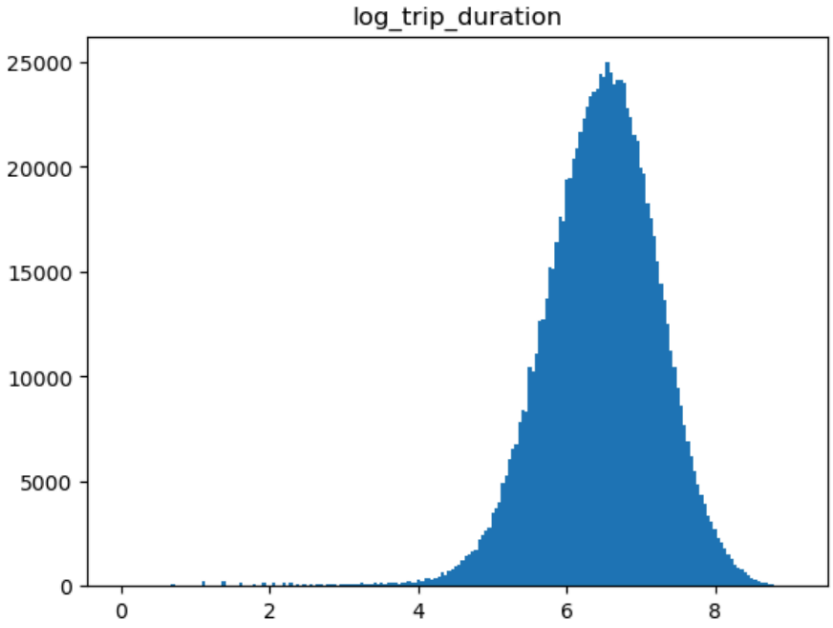
Figure: Right-Skewed Distribution

"We need a log transformation for `trip_duration`"

The figure could show a histogram plot of the `trip_duration` feature, demonstrating a right-skewed distribution with most of the data clustered on the left and a long tail extending to the right.



after apply log :



2.3 features analysis:

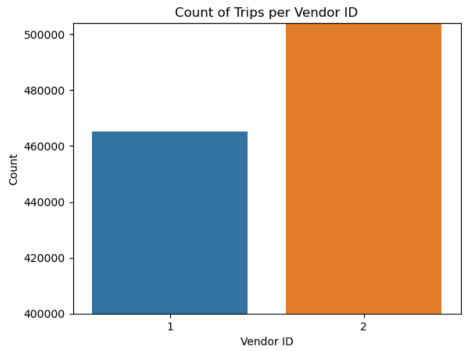
2.3.1 Numeric features:

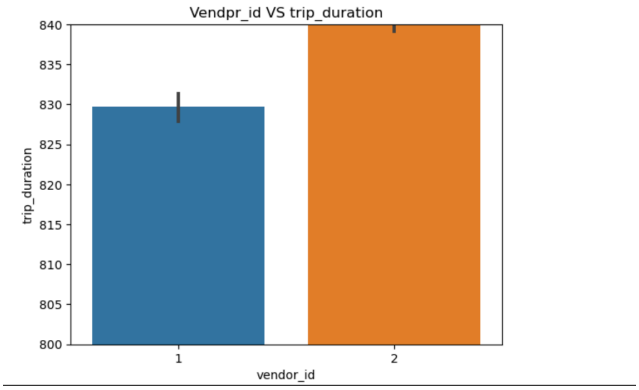
numeric features is

vendor_id,pickup_longitude,passenger_count,pickup_longitude,pickup_latitude,dropoff_longitude,dropoff_latitude

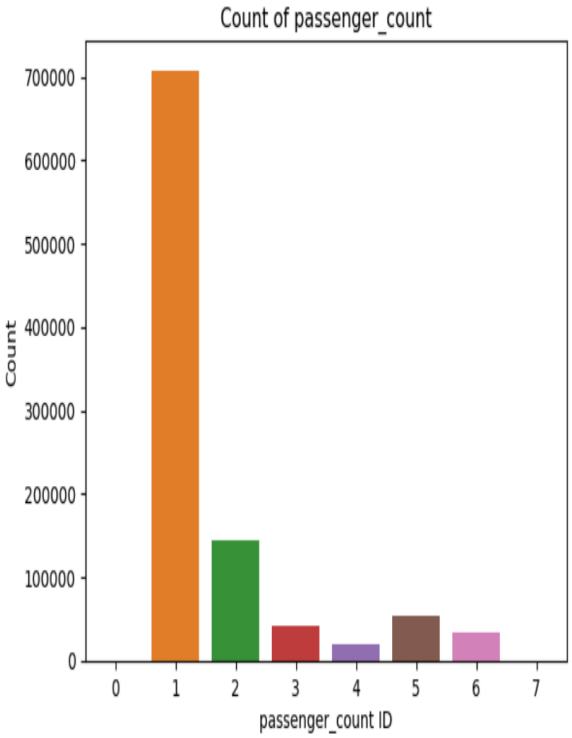
first for Discrete Features

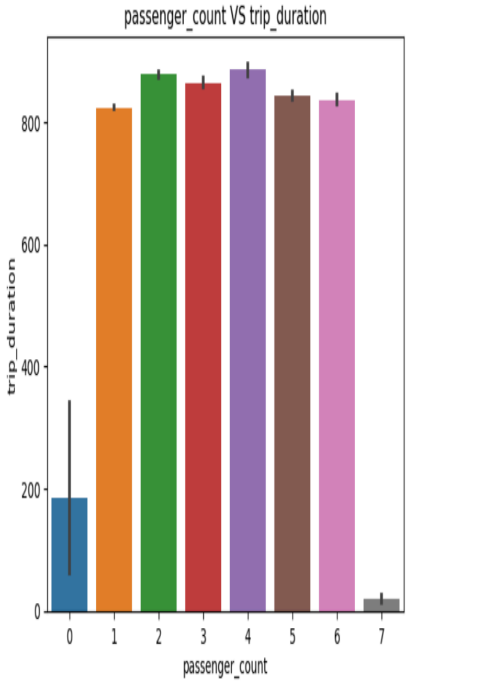
- Discrete Features is vendor_id,passenger_count
-





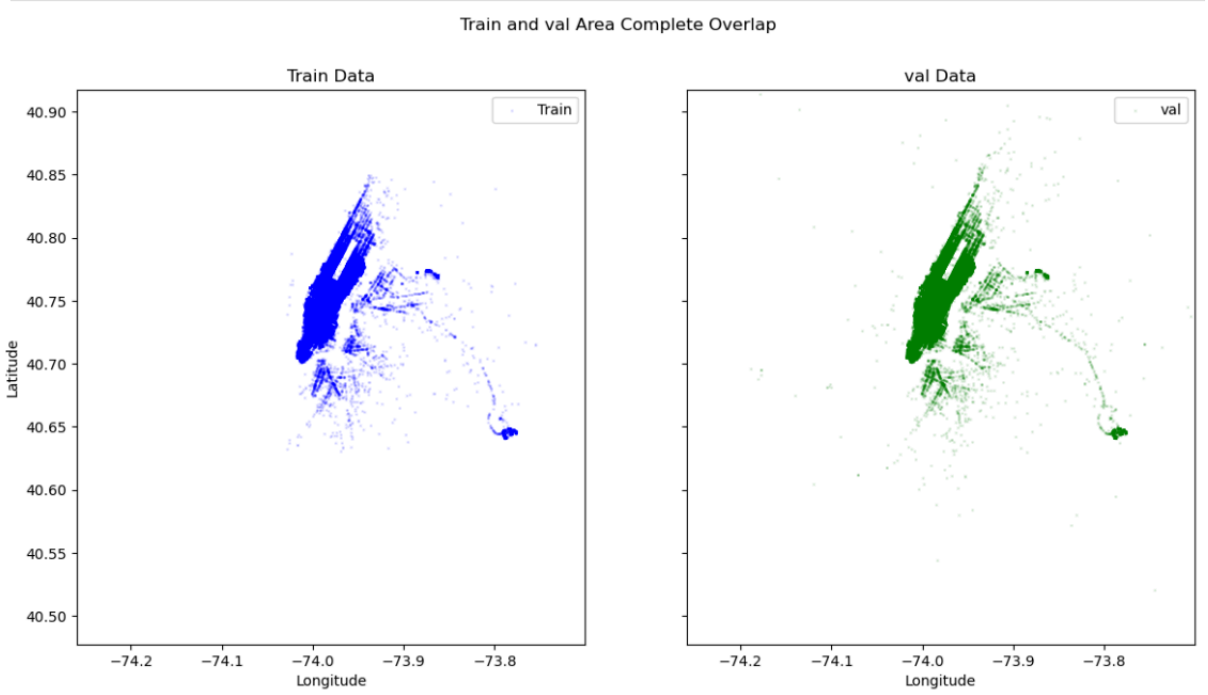
- *The difference between Vendor ID 1 and 2 is about 60,000*
- The difference between the two vendors is 10 for target , which is small





- The number of passenger counts for ID 1 is much higher than for the others
- The limit for the size of a taxi is no more than 6 passengers
- I found that the number of passengers, ranging from 1 to 6, is closely clustered and significantly impacts the target variabl

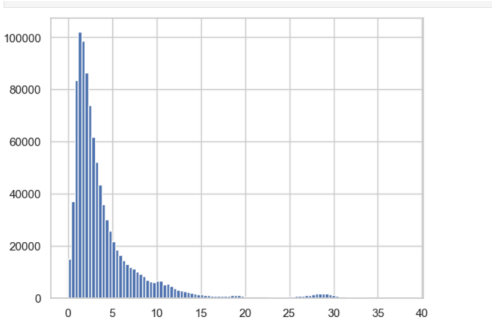
second Continuous Features



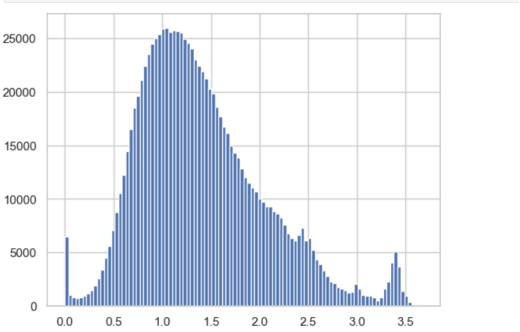
- In this case, the train and test split appears to be random. This allows us to use unsupervised learning and apply feature extraction to the full dataset.

let`s for extracted Features

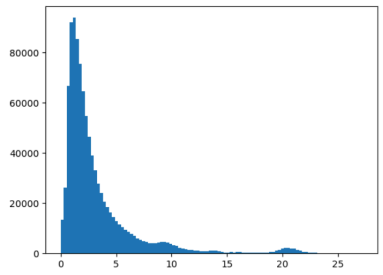
1 - distance_dummy_manhattan



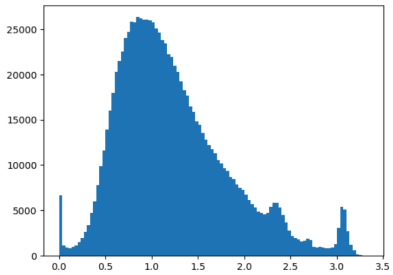
after log



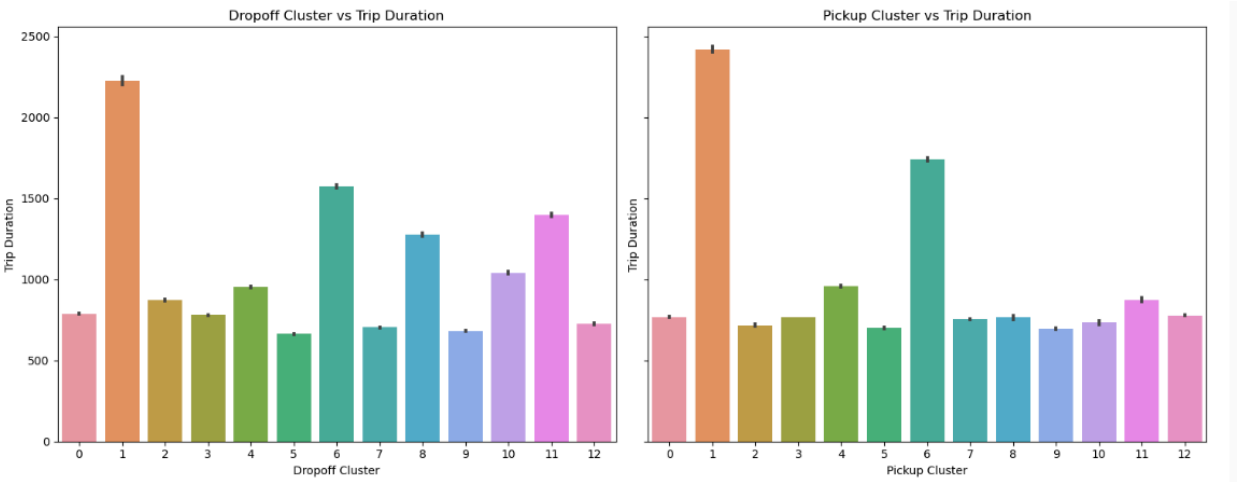
2- distance_haversine



after log



3- pickup_cluster and Dropoff Cluster



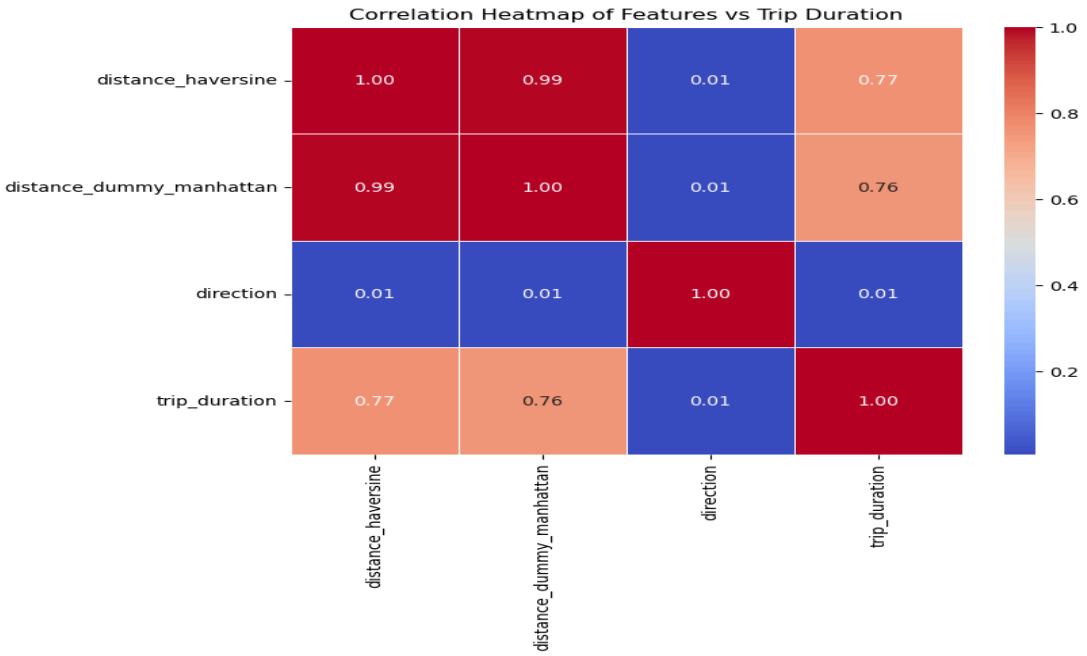
High Trip Durations:

- Both dropoff and pickup clusters have one cluster (Cluster 1) with exceptionally high trip durations, suggesting that trips starting or ending in these clusters are significantly longer.

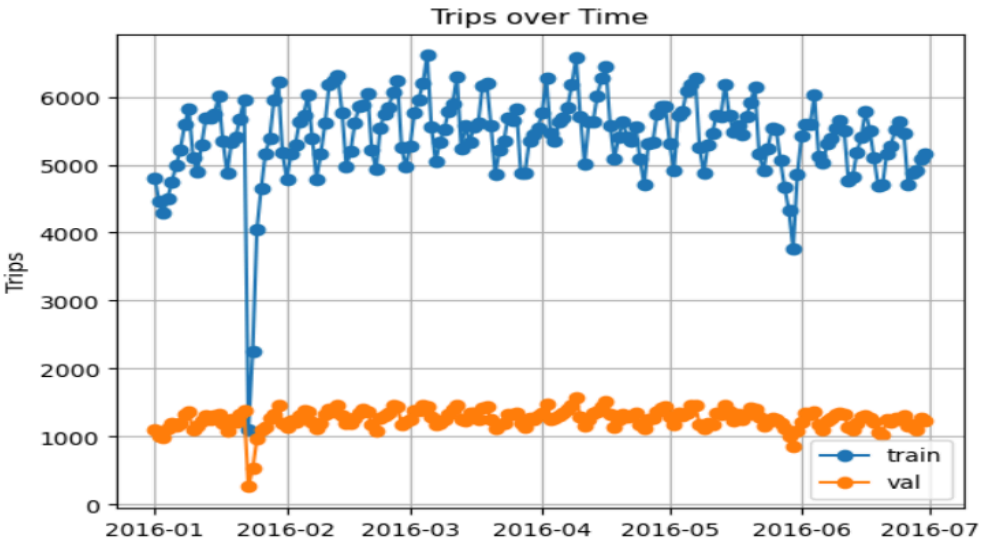
Variability:

- There is a noticeable variability in trip durations across different clusters, indicating that the location clusters play a significant role in trip duration

now let`s to correlation



2.3.2 Data Time



- It's clear that the two datasets is the same pattern but the val is more narrow

4-model

The data pipeline is designed to transform both categorical and numerical features before fitting the model. It ensures that the data is prepared appropriately for modeling, following a series of preprocessing steps. Below is a detailed breakdown of the transformations applied:

1. **Feature Splitting**:
    - The features are categorized into **categorical** and **numerical** groups based on their types, allowing for separate preprocessing.
2. **Categorical Feature Processing**:
    - We apply **One-Hot Encoding** to convert categorical variables into binary format, making the data compatible with machine learning algorithms that require numeric inputs.
3. **Numerical Feature Processing**:
    - **Standard Scaling**: Numerical features are scaled using the **Standard Scaler**. This transformation centers each feature around zero and scales it to unit variance, ensuring that all features contribute equally to the model.
    - **Polynomial Feature Expansion**: Polynomial features with a **degree of 2** are generated to capture complex interactions between numerical variables. This step allows the model to identify non-linear relationships that a simple linear model might miss.

Model Training:

- The processed data is used to train a **Ridge Regression Model** with an alpha value of `100` to regularize the model and prevent overfitting.

Result

- train RMSE = 0.4170 - R2 = 0.75

- test RMSE = 0.4270 - R2 = 0.7150

The dataset I am working with is quite large, and while using advanced models would likely result in better performance, that is not the primary focus of this project.
the main goal here is not just to achieve high performance but to demonstrate and showcase my skills in data preprocessing, feature engineering, and model building.
I want to highlight my ability to design effective pipelines, perform appropriate transformations, and implement various machine learning techniques, rather than simply optimizing for the highest accuracy. By choosing simpler models, I can emphasize the steps and thought processes behind creating a strong foundation for any future improvements. This project serves as a learning experience and a testament to my understanding of machine learning concepts.