Trip Duration Prediction Project Summary

Awab Medhat Erwa - January 9, 2025

Introduction

In this project, I built a model that predicts the total ride duration of taxi trips in New York City.

This competition was held by Kaggle, the primary dataset is

provided by the NYC Taxi and Limousine Commission which includes pickup time, geo-coordinates, number of passengers, and several other variables.

EDA Findings

1- Trip Duration Analysis

As we can see in Figure 1 the log(trip_duration) represents a gaussian distribution, also there is some outliers, most of the the data is around 5.5 - 7, which is around 250 - 1000 seconds.

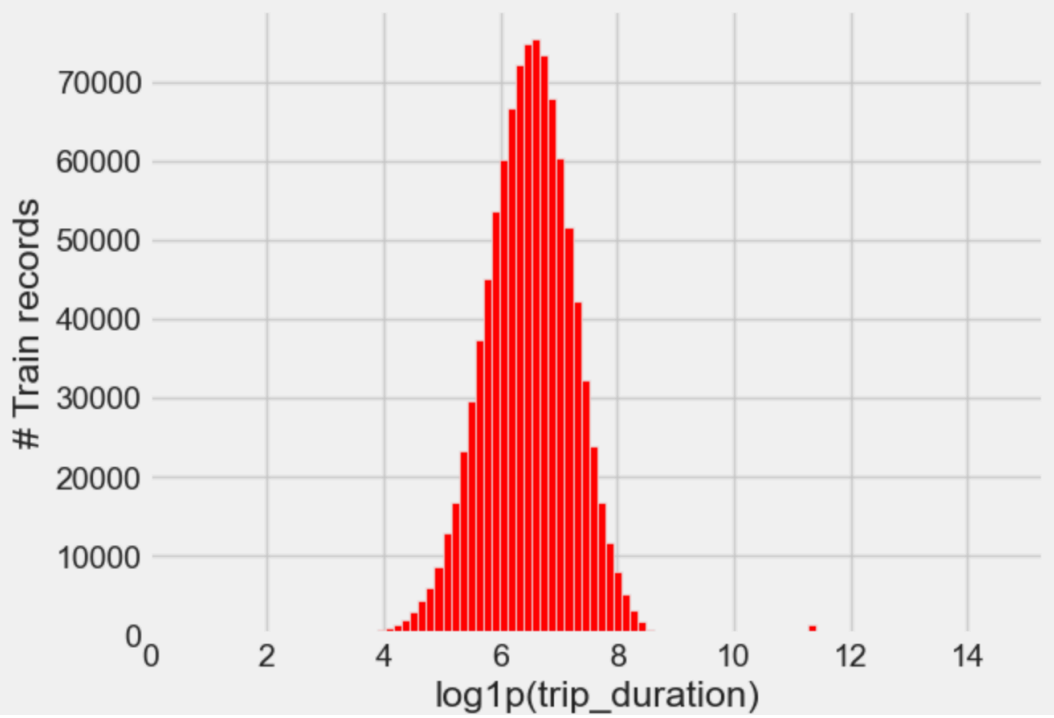

Figure 1

Also the data covers a specific period of time in 2016, from January to July as we can see in Figure 2.

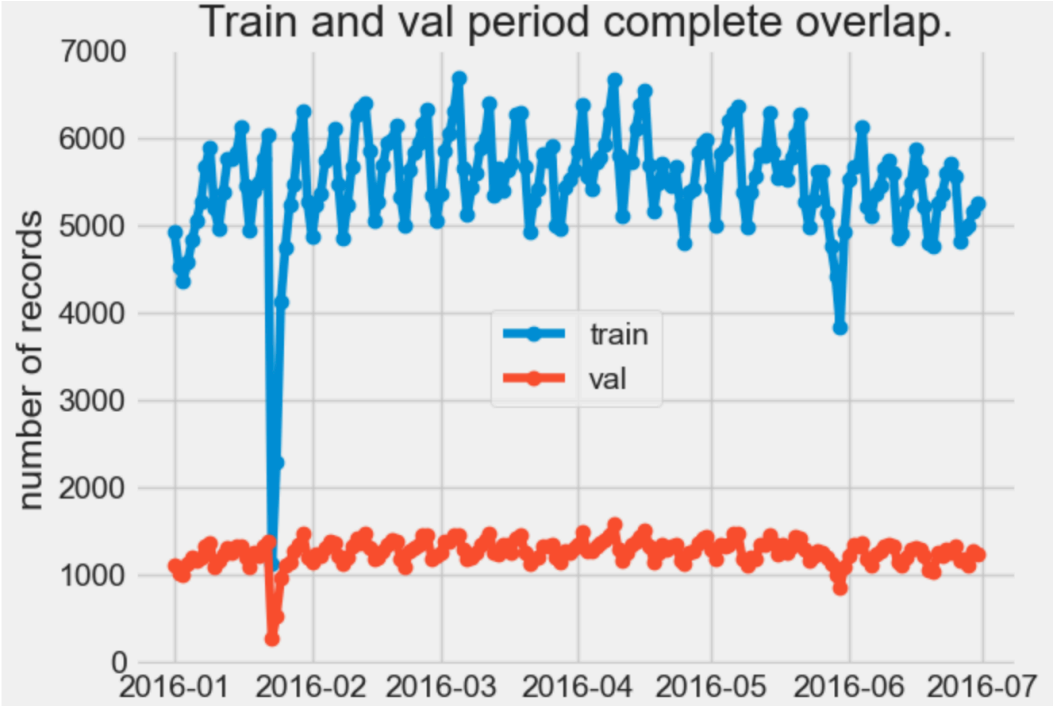

Figure 2

2 - Trip Duration and Pickup time

Pickup time is a critical feature for analyzing trip duration, as we know timing play a big factor in real world for trips duration, so I made a box blot for analyzing trip duration over different time periods, including: month, day, weekday, and hour.

Looking at Figure 3, it's obvious that the biggest factors are weekday and hour, and nothing special in month and day. In weekdays, weekends have less trip duration (0 = Monday, 1 = Tuesday..etc), in hour, when pick up in pm specially in afternoon, trip duration is higher.

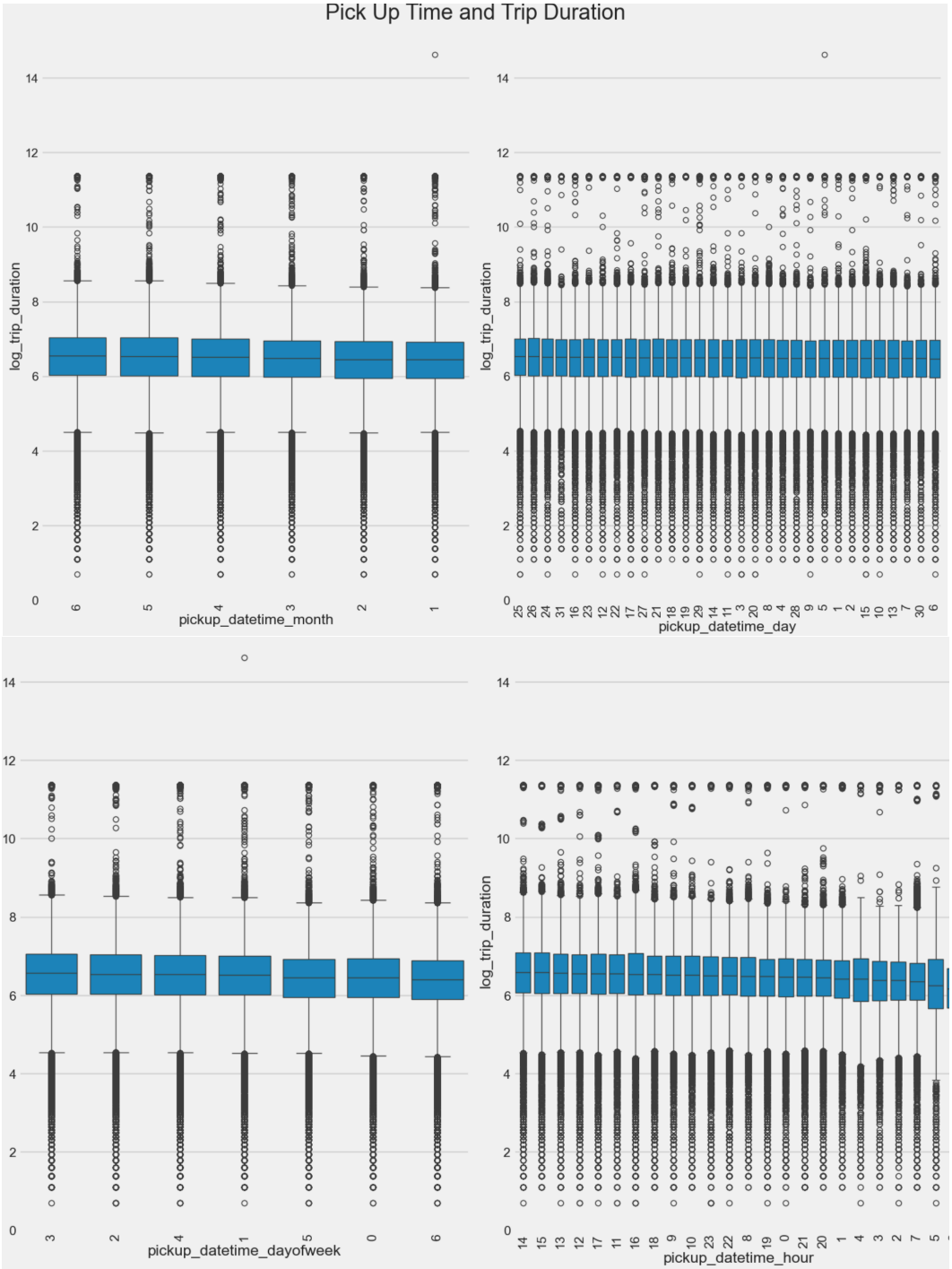

Figure 3

3 - Trip Duration and Passenger Count

From Figure 4 we can see that passenger count from 1 to 6 don't vary much in trip duration, 0 and 7 seems outliers since they occur very few times.

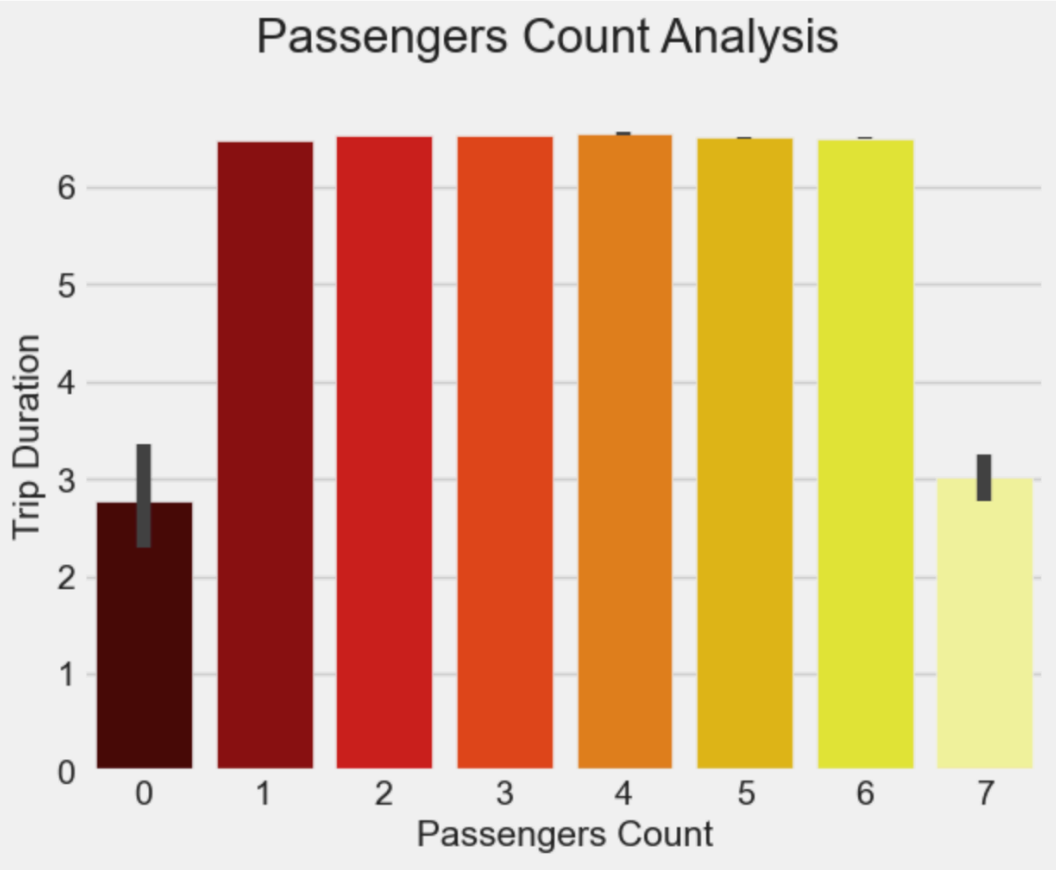

Figure 4

Feature Engineering

1 - Adding Distance Feature

Adding the distance to our features is significantly important as there is high positive relationship between trip duration and distance, also I added compass bearing from pickup to drop-off and the center latitude and center longitude between the pickup and drop-off.

2 - Performing Principal Components Analysis (PCA)

PCA is normally used for dimensionality reduction, but here I used it to transform longitude and latitude coordinates as we can see in Figure 5.

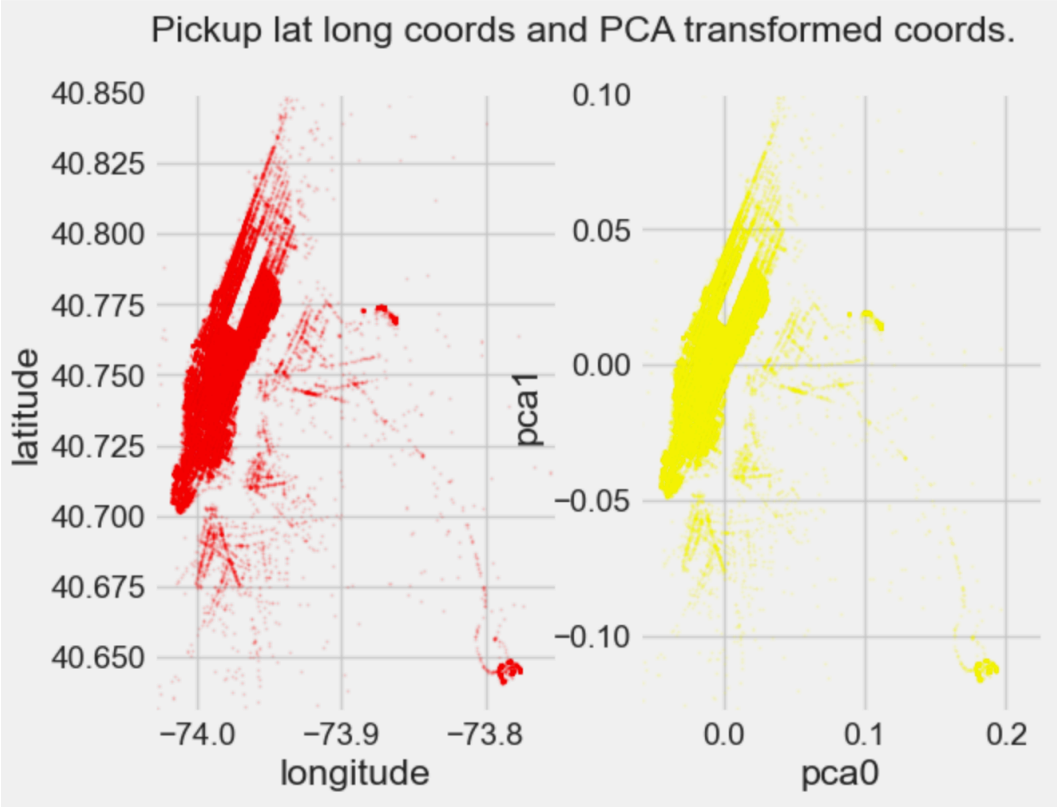

Figure 5

3 - Adding Clusters

Here I used k-means clustering to make every (latitude, longitude) point belong to a cluster to help modeling as we can see in Figure 6.

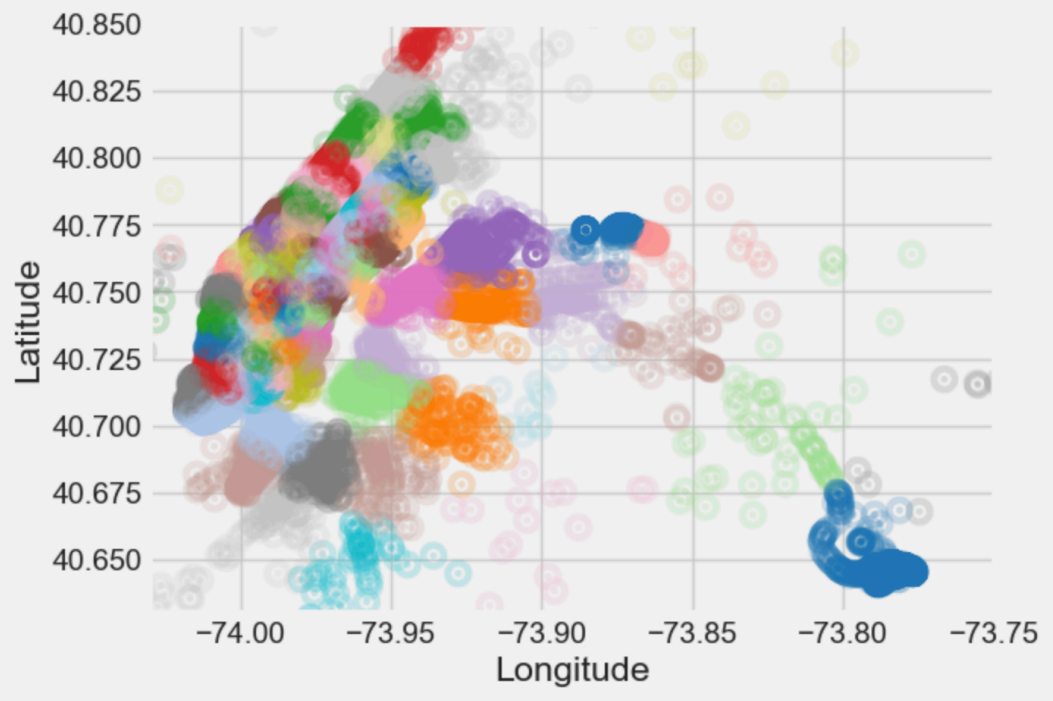

Figure 6

Modeling

1 - Data Preprocessing

Before giving the data to our machine learning algorithm, I did several transformations to it, first I split the features into 4 parts each of them are transformed with PolynomialFeatures either with degree 5 or 6, then I took one or two features from each of the 4 parts, and made a combination of them and transform it with PolynomialFeatures with degree 7, after that I did a StandardScaler then log transformation. The reason behind splitting the features into 4 parts is for saving memory and computing power as I have limited resources, you can see the full pipeline in Figure 7.

2 - Final Model

In the final model I used a regularized linear regression, Ridge with alpha=1, giving the following scores:-

Metric	Train	Validation	Test
R2	0.677961075091914	0.655190462155564	0.678533146245086

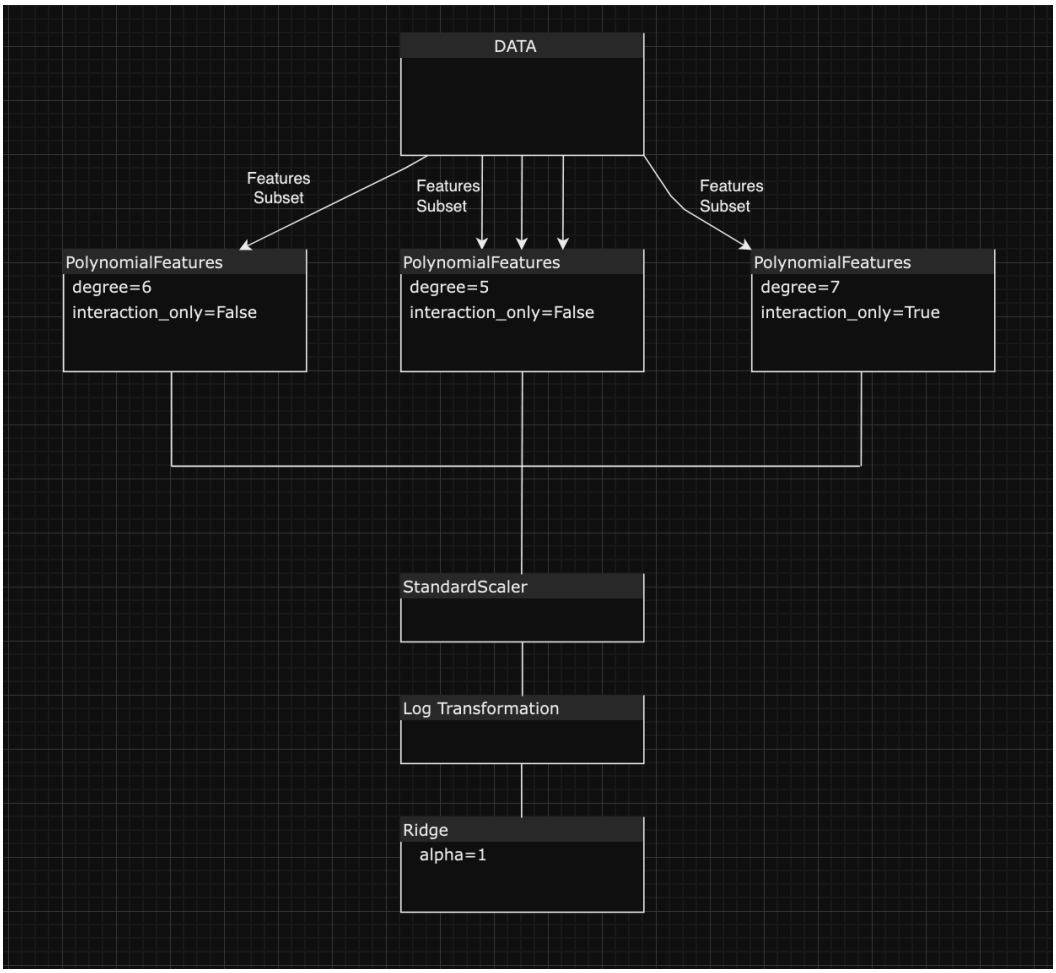

Figure 7