PREDICT TAXI TRIP DURATION

Problem statement

Predict New York City taxi trip duration

Useful for taxi company's **fleet management** – how many taxis will be there in a particular location for a given time point

Also beneficial for **customer satisfaction** if the taxi company has an app for booking rides

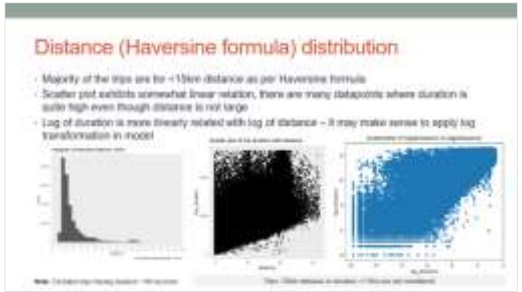
- Referred to Kaggle to extract the required data
- Objective is to predict taxi trip duration
- Available data elements include pickup date and time, geo-coordinates, number of passengers, vendor ID, and a few other variables
- Data is available for 1.5M trips which can be used to train a machine learning model

Approach: overall

- **Data exploration and cleaning**
  - Various data dimensions are visualized to understand how the data is distributed
  - Some elements are visualized in combination of others to check for any interaction
  - Identified some irregularities in the data while exploring it - those irregularities will be discarded as they are not in large scale
- **Additional feature creation**
  - Additional features have been created using the existing information to help predict trip duration
    - Distance related to the earth between pickup & drop-off coordinates,
    - Pickup time related metrics: time of pickup day, pickup day of week, whether pickup day aligns with any holiday, etc.
    - Some proxy information about traffic: number of trips around a given pickup location at similar time and similar day
- **Model training and evaluation**
  - We used regression, and tree-based models for training
  - RMSLE for goodness of fit – SKlearn recommends this over RMSE when targets have exponential growth

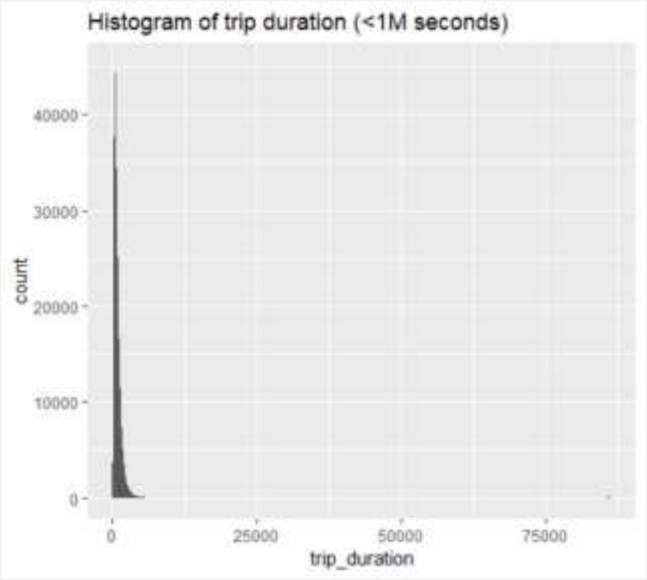
Data exploration

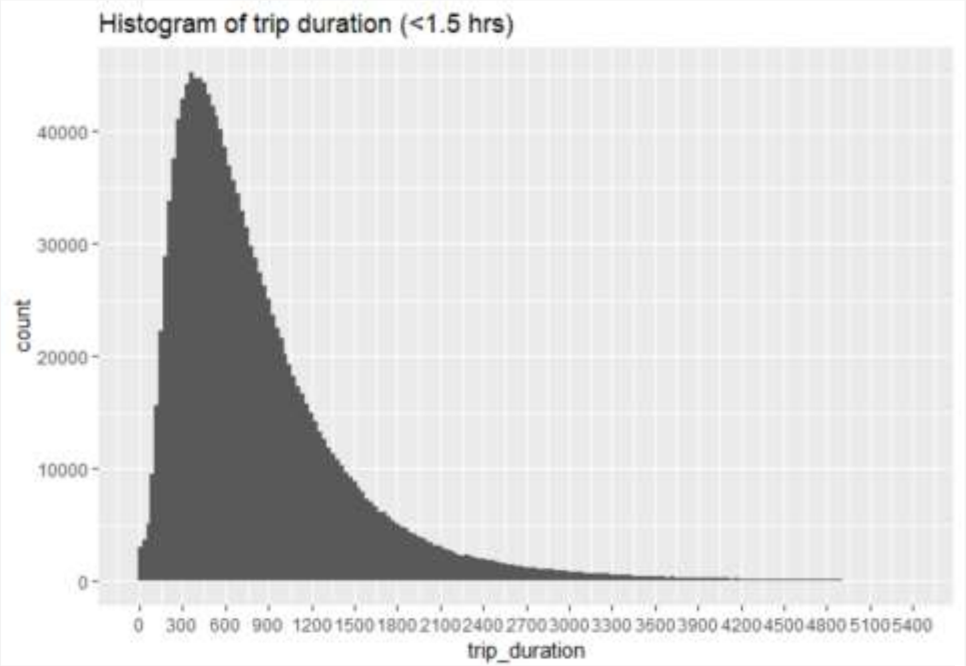
- **There are no missing data elements, however some outliers are noticed**
  - Number of **passengers is 0** in 60 trips out of 1.5M trip records
  - **Trip duration** is abnormally high, **>1M seconds**, in only 4 trips; also, there are only **3K trips (0.2% records)** whose **duration >1.5 hrs**
  - Some geo coordinates fall outside NYC; and some lie on the Pacific ocean if we visualize!
  - ~450 trips had Haversine distance between geo coordinates >32km
- **Other observations**
  - 5K trips ended in <30 seconds! - these could be trips that were immediately terminated or just bad data
  - Only **4 trips** exist where **number of passengers >6**, we may combine them in a bucket as >=6 passengers

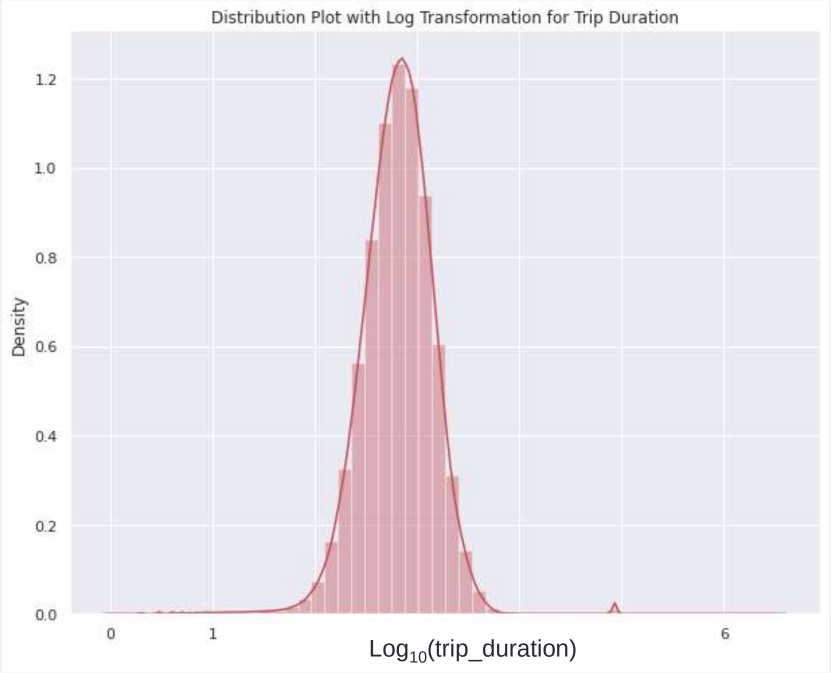


Distribution of trip duration

- Mode of the trip duration distribution is around 400 seconds, <7 minutes
- It has a huge right tail (positively skewed)
- It is not Normally distributed, closer to Gamma distribution
  - Since we are applying RMSLE, log transformation makes it more symmetric and closer to Normal distribution





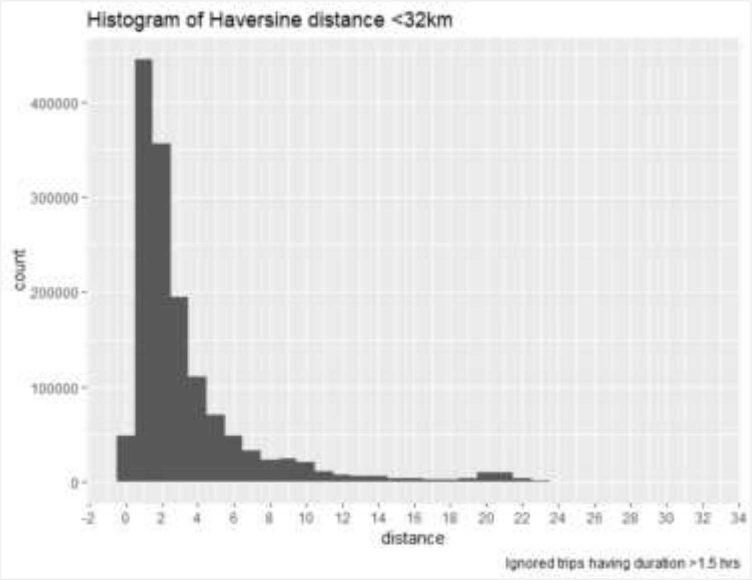


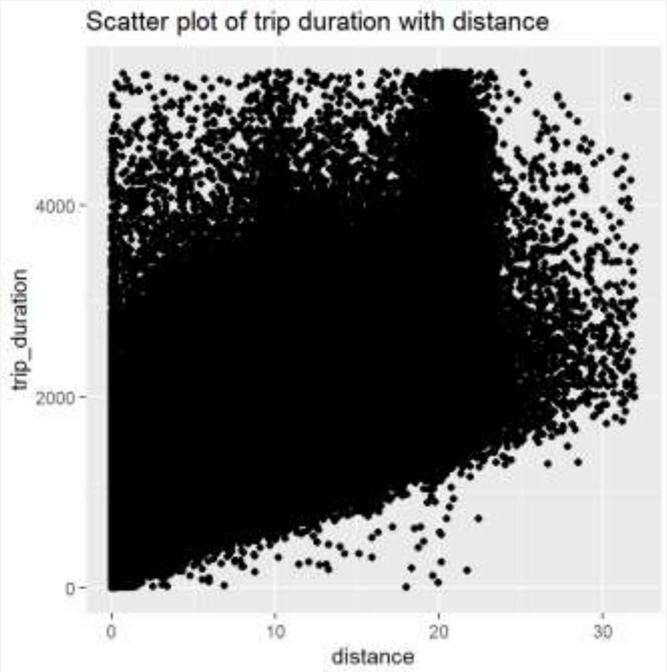
Bin width: 30 seconds

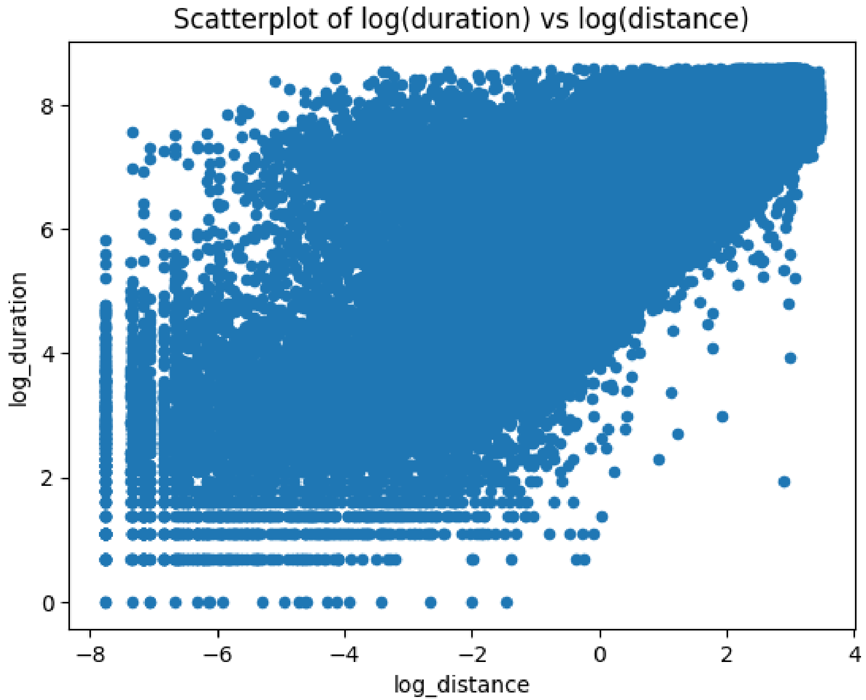
Note: Excluded trips having duration >1M seconds

Distance (Haversine formula) distribution

- Majority of the trips are for <15km distance as per Haversine formula
- Scatter plot exhibits somewhat linear relation, there are many datapoints where duration is quite high even though distance is not large
- Log of duration is more linearly related with log of distance – it may make sense to apply log transformation in model





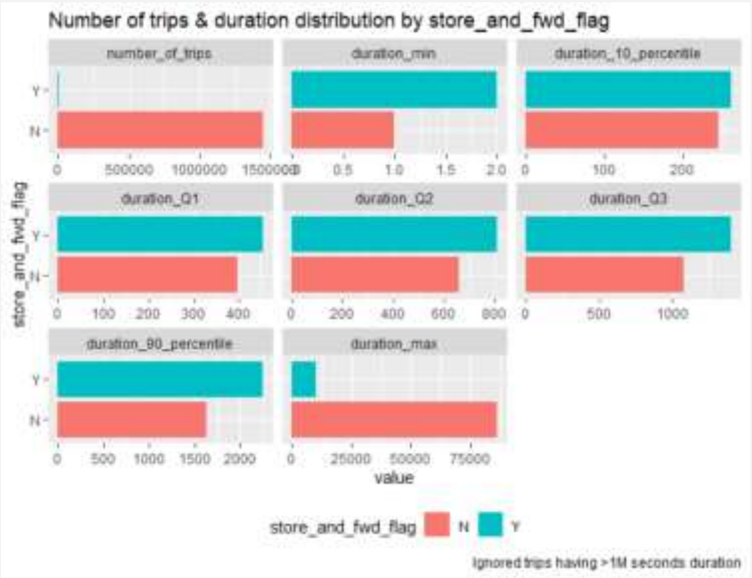


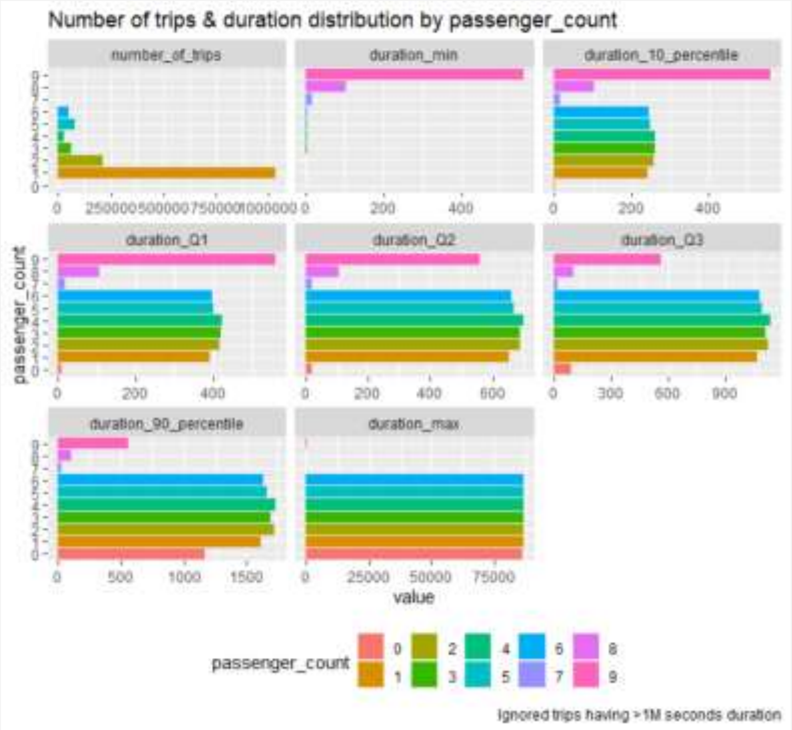
Note: Excluded trips having duration >1M seconds

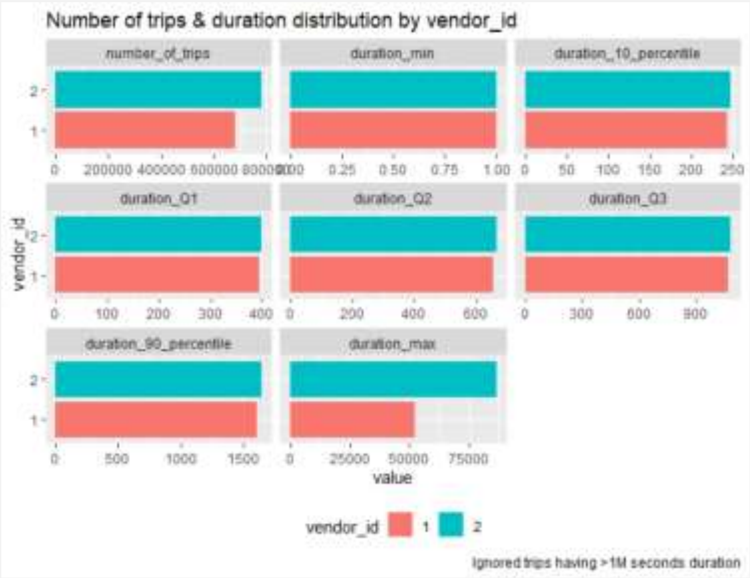
Trips >32km distance or duration >1.5hrs are not considered

Number of trips and distribution of duration by categorical features

- Trip duration is usually higher for store_fwd_flag='Y'; also for passenger count 2 to 4
- Vendor 2 taxis take ~2-3% more time than vendor 1







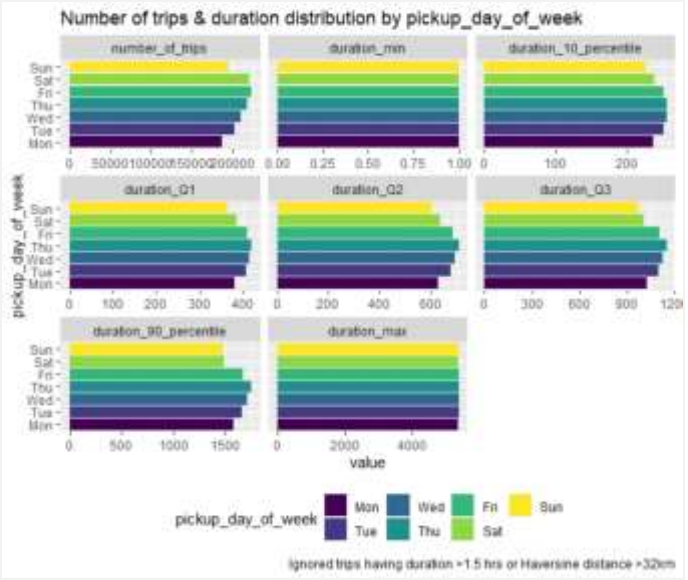
store_and_fwd_flag - flag indicating whether the trip record was held in vehicle memory before sending to the vendor. Quite sparse data (N: 99.4%, Y: 0.6%)

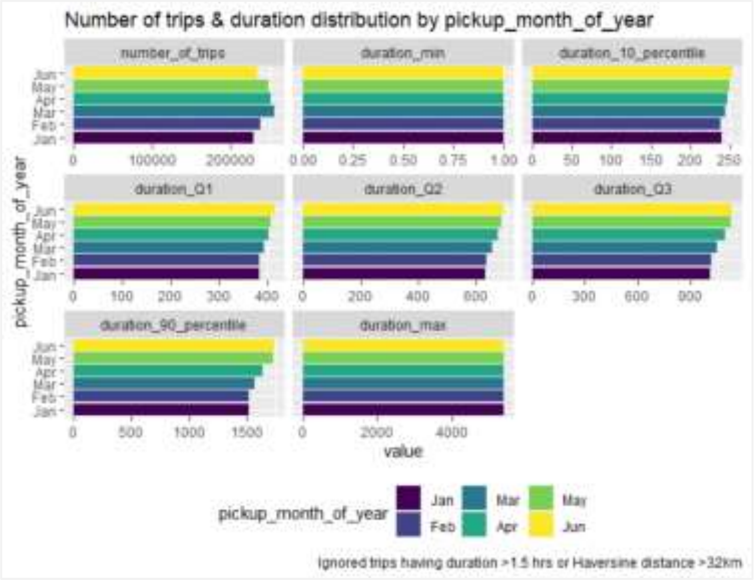
passenger_count – number of passengers in a trip. Intuitively it does not make much sense that # of passengers will impact trip duration

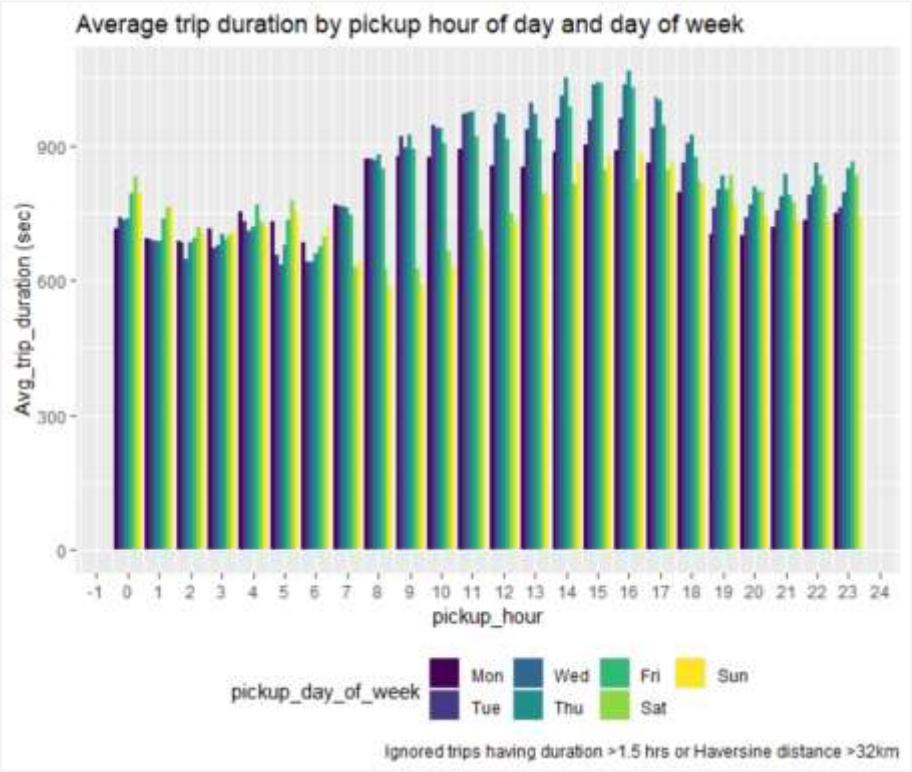
vendor_id - code indicating the provider associated with the trip. Only 2 distinct ids (1: 46.5%, 2: 53.5%)

Trip duration distribution by time dimension

- Trip durations are usually high during Tue-Fri
- Trip duration gradually increases as summer period approaches
- Average trip duration is higher on weekends after midnight until early morning
- Average trip duration gradually increases throughout the day on weekdays



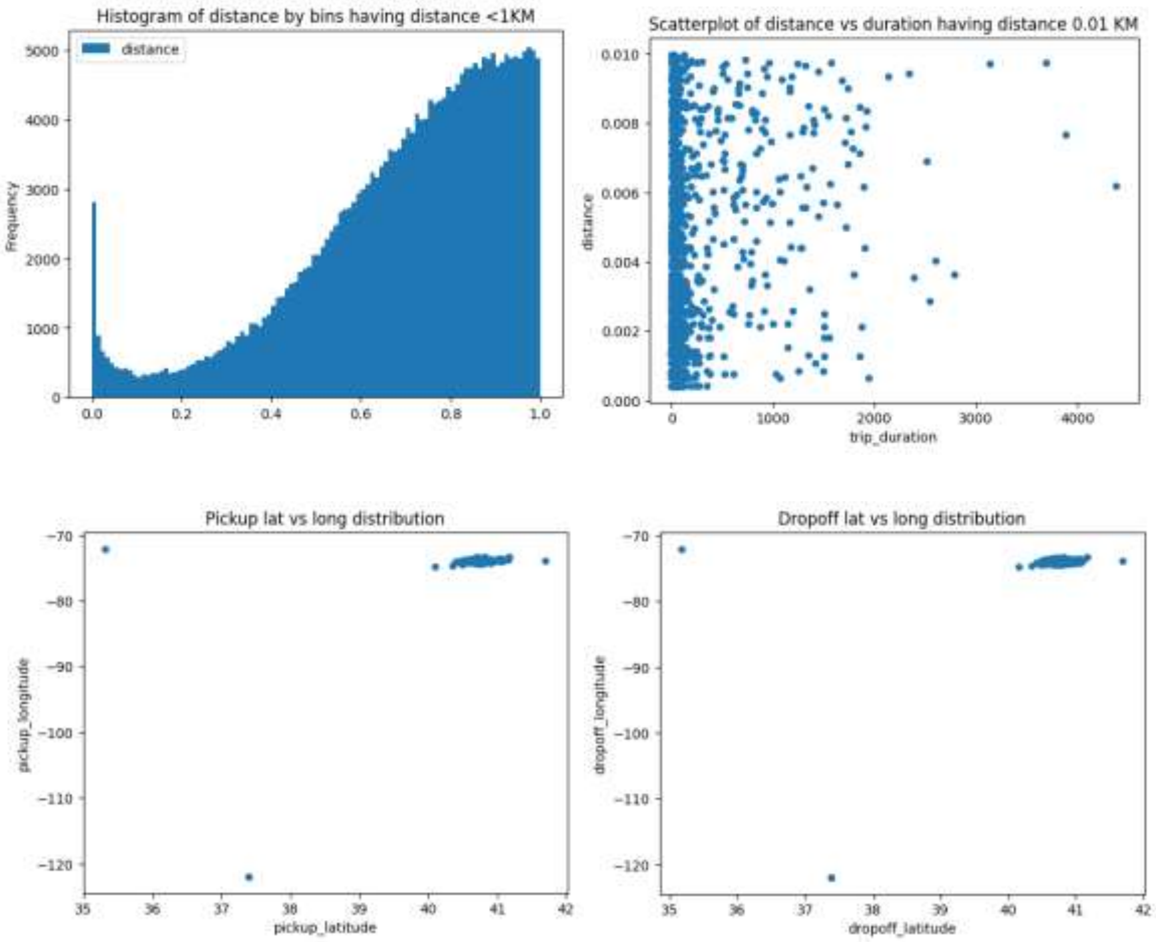




Some more outliers on co-ordinates and distance observed

- Lots of trips are there where distance is very, very small
  - ~**6K trips are with same pickup & drop-off locations**, hence distance became 0!
  - ~**3K trips** have Haversine **distance <10 meters** - could be bad data
- Still in some of those cases the duration is all over the place, upto ~1 hour.

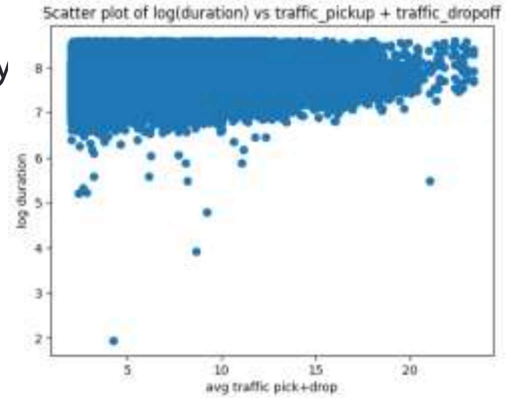
- Most of the pickup/drop-off co-ordinates are concentrated in a relatively small area
  - We may **discard the data points** where pickup/drop-off **latitude is less than 39**



Note: Excluded trips having duration >1.5 hrs, or distance >32 KM

Feature engineering

- We **removed** some of the identified **outlier cases which are minimal**, not to distort the analysis
  - distance >32km, or distance <0.01km, or duration >1.5 hrs, or pickup_latitude <39 or passenger count =0
- **New variables** added:
  - **Distance** (Haversine formula) based on pickup and drop-off coordinates
  - Categorized passenger count into buckets: 1, 2 to 4, >4
  - **Pickup day of week**, categorized as midweek (Tue, Wed, Thu), and kept other 4 days separate
  - **Pickup hour in a day** by buckets: 00-04, 05-06, 07-10, 11-13, 14-18, 19-23
  - Interaction of above two categorical features as day of week and hour in a day
  - **Pickup month of the year**
  - Whether the **pickup day coincides with holiday**, or **day before** holiday, or **day after** holiday
  - **Proxy of traffic**: average number of trips within a small region (+/- 0.01 of latitude or longitude), within a small time-window (15mins window of a day), and by day type as per the day of week bucket defined above
    - If any of the attributes in test data are not found in traffic calculated as above, then calculate average of (avg_traffic_given_location, avg_traffic_given_timewindow, avg_traffic_given_daytype) in test data, and use that as a proxy



Modelling approach

- Multiple model forms tested
  - **Gamma Regressor**: as the trip duration showed some form of Gamma distribution.
    - Considered trip duration in original scale as dependent
  - **Random Forest** Regressor
  - **Histogram-based Gradient Boosting** Regressor
    - Considered log(trip duration) as dependent for both the tree-based models above
  - Continued with **Histogram-based Gradient Boosting**, which produced better results than others prior to hyper-parameter tuning
- Data was **split** into **train and validation** (90:10)
- **Standardized** the data before training, although it does not matter much for linear regression or tree-based models
- **Categorical variables** were **one-hot encoded** for regression and random forest models. Histogram-based gradient boosting model **natively supports categorical data**
- **5-fold cross-validation** was used for checking model performance
- **Grid search** was done for tuning **hyper-parameters**

Model outcomes

Initial model performance

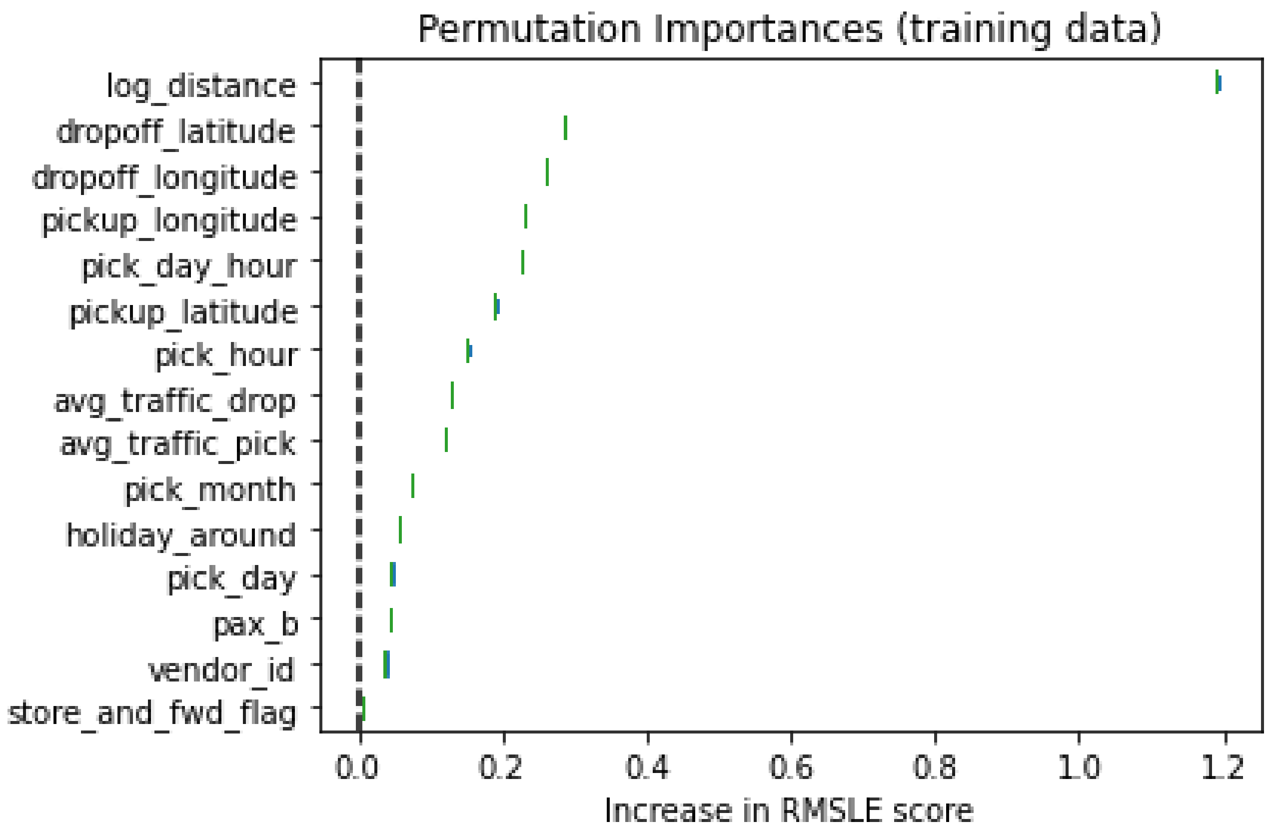
- **Gamma** Regressor produced **RMSLE on training: 0.43**, validation: 0.49
- **Random Forest** produced average **RMSLE** on **3-fold cross-validation: 0.3758**
  - Considered “min_samples_leaf=10” and “max_features='sqrt'” as otherwise it was taking huge time to train
- **Histogram-based Gradient Boosting** produced average **RMSLE** on **3-fold cross-validation: 0.3681**

Hyper-parameter tuning

- Considered below parameters of HistGB for tuning via Grid Search:
  - learning_rate, max_bins, min_samples_leaf, max_iter
- **Tuned HistGB** produced average **RMSLE on 5-fold cross-validation: 0.3284**; and on validation data: 0.4643
  - Note that the traffic information was already extracted for the training data, hence cross-validation produced better result than on validation data
- **Final tuned model** was **trained** on the full data (**combination of train & validation**), and submitted to Kaggle for evaluation (private: 0.4225, public: 0.4244)

Model feature importance

- Feature importance extracted using Sklearn's 'permutation importance' API
  - It calculated a baseline metric on the training dataset.
  - Next, a feature column is permuted and the metric is evaluated again.
  - The permutation importance is defined to be the difference between the baseline metric and metric from permutating the feature column
- It clearly shows **distance** as the **most important feature on training data**, followed by co-ordinates, and the engineered features (interaction of day-hour, traffic, …)



Next steps

- There were many data points having distance <0.01 KM; instead of discarding them, since we applied log transformation on distance, maybe it may help to add an **indicator variable to flag if distance is very small**. (log(0) tends to negative infinity!)

- We could have used all the passenger counts instead of bucketing them as 1, 2 to 4, >4 etc. as we can build HistGB model without increasing the number of variables.

- Creating **cluster of co-ordinates based on traffic, duration** etc. may help improving the predictive power. Using co-ordinates as numeric variable does not make much sense, rather using them to create categorical features could have made more sense.

- **Some other modeling algorithm** e.g., neural network could be tried.

APPENDIX

Additional visuals

Pickup locations outside NYC, some on the ocean!