Problem Statement:

To predict transit time (in seconds) between Pick up and Drop off Location at Pick up Location –Drop off Location – Month- Week- Day of Week- Hour combination for Chicago Taxi Trips.

Solution Approach:

Data Set Selection: As the Actual data set is over 100 million records for 2013-2017, It is tough to take every data point into RAM. So I sampled the dataset using stratified sampling approach.

- Taking 10 percent of the data for training using same proportion of pickup community area across the 100 million dataset.
  Example: If Area-1 and Area -2 consists of 0.1 and 0.15 proportion of 100 million, here we take same proportion for 10 million data points

Pre Process and Cleaning of the Data:

- Removed taxi_id null rows – As They are very few in number
- Removed taxi_second nulls – As target variable is required to build the model
- Tried Various Methods to impute taxi_miles and taxi_total nulls but none worked well.
  - Not much co-relation between taxi_miles and taxi_total
  - When Taxi_miles is null most of the times pickup and drop off areas are null so could impute only few rows
  - So removed those nulls
- Tried to build predictive model to impute drop_offlocation - > trip_miles + Trip_total + pickup_location using predictive approach but performance is decreasing so not used for final model instead removed nulls
- Removed outliers for trip_seconds where the values exceeds 99.97th percentile (around 4 hours) and falls below 5th percentile (180 seconds)
- Both Pickup and Drop off Co-ordinates are binned across 15 bins using KMeans and the distribution is shown in Fig 1
  - They are binned distance wise so that new pickup and drop off co-ordinates comes in future One can easily predict the cluster that the point belongs.

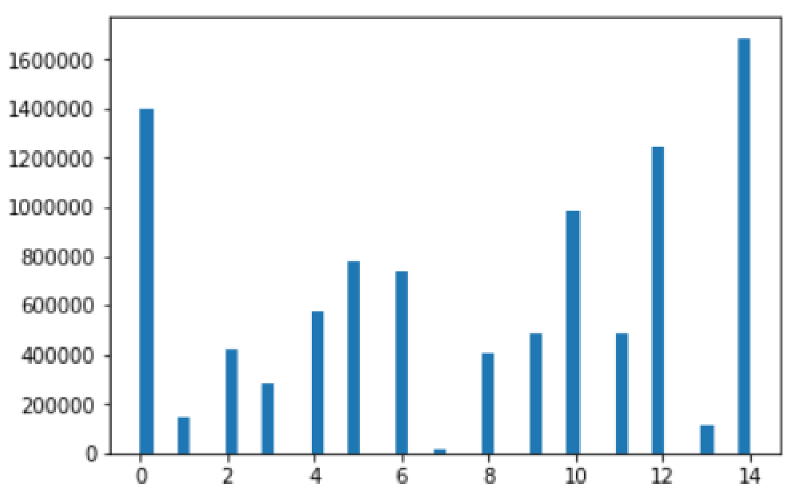
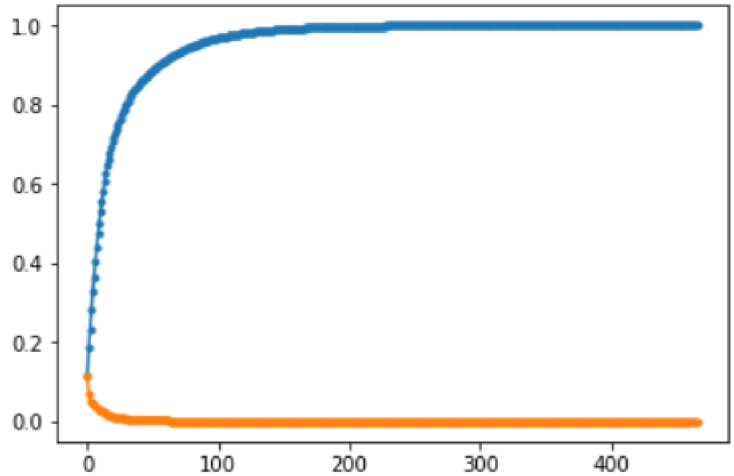
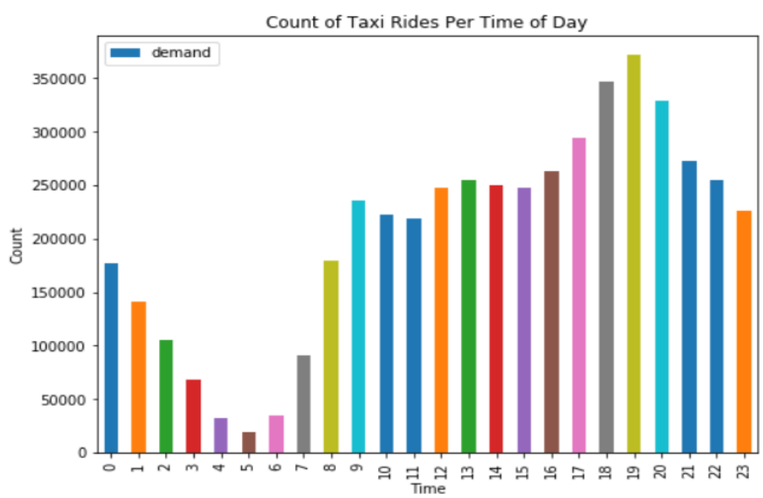


Fig 1: **Clustering using Pick up and Drop off Latitude and Longitude**

- Some of the Drop off Locations which are infrequent are grouped together. Example : All the cumulative demand greater than 0.95 were grouped as Others.
  The below plot shows the cumulative sum (blue) and actual demand (orange)



- Pick up Location was also grouped according to the same logic and follows similar distribution
- Hour is binned into 6 categories based on the demand shown below
  - [2 - 7] - 1 category
  - [8 -11] – 2 category
  - [12 – 16] – 3 category
  - [17 -20] – 4 category
  - [20 – 23 – 5 category
  - [0-1] – 6 category



- Company is also binned using the similar logic but instead taken threshold cumulative sum as 0.99
  - Missing values are imputed using taxi id and Company mapping. And Other nulls are treated as separate variable 'Others_2'

Note : Didn't share plots or insights that are not helpful to final solution

Feature Engineering:

The features that were used for predicting the model were

- Hourly Traffic Data : Calculated based on the demand at that Date and pick up hour to the drop off cluster / demand at that Date and pick up hour. Snapshot of table shown below

Month	day	year	hour_start	dropoff_cluster_label	unique_key_count
1	1	2013	0	0	17
				1	6
				3	15
				5	35
				6	25

Fig : Demand at Date and Hour to the drop off cluster

	Month	day	year	hour_start	dropoff_cluster_label	traffic_hr_cluster
0	11	22	2014	19	10	0.086806
1	11	22	2014	19	10	0.086806
2	11	22	2014	19	10	0.086806
3	11	22	2014	19	10	0.086806
4	11	22	2014	19	10	0.086806

Fig : Traffic at Date , Hour to drop off cluster

- Hourly Weather Data : Chicago Hourly weather data from Kaggle: https://www.kaggle.com/selfishgene/historical-hourly-weather-data
  - Temperature , Pressure and Humidity
  - **Missing values are imputed using exponential weighted average over 12 hours**

- Pickup Cluster and Drop off Cluster : Calculated above based on Kmeans
- Drop off Location Bin and Pick up Location : Calculated based on above mentioned logic
- Company Bin: Calculated based on above mentioned procedure
- Trip Miles , Trip Total were used directly
- Time Variables : Month, Week, Day of Week, hour bin

The categorical variables are one hot encoded which include 'company_bin', 'pickup_cluster_label', 'dropoff_cluster_label','dropoff_location_bin', 'pickup_location_bin', 'Month', 'Week', 'DayofWeek', 'hour_bin'

The continuous values are scaled which include 'trip_miles','traffic_hr_cluster', 'tolls', 'trip_total', 'humidity', 'pressure', 'temperature'

Both categorical and Continuous Values are joined and sent as input to the model for training and prediction

Model Building:

- Used Neural Networks with (266,130,65,1) architecture to predict the travel time
  - Relu is used as Activation Function for both Hidden Layers
  - Batch Normalization is Used at Hidden Layer to Improve the training performance and convergence
  - ADAM Regulizer is used for effective updating of weights

- Drop out and L2 regulizer are used to avoid the over fitting of the model
- Number of Epochs : 50
- Loss Function : MAPE
- Used 20 % of the data for validation to improve the model and test the performance.
- The MAPE for Validation data set is 22.34 for 1 million records

Results:

MAPE is 57.742 for 4335034 records of 2017 data after removing nulls with respect to taxi ids, trip seconds and co-ordinates

Various approaches experimented:

- Experimented with imputing null values using following methods
  - Direct Mapping like trip miles - > pick up and drop off location
  - Impute using Predictive modeling using Logit for drop off location -> trip miles, pick up and trip total but performance is degrading
- Experimented with Using Simple Neural Network with one hidden layer
- Experimented with Random Forest Regressor but didn't show better performance
- Tried using XGBoost but didn't converge because of memory issues.
- Used All Weather Features and Weather Text based Features but not showing better performance.
- DNN's with Batch Norm and Drop out outperformed the previous model approaches

Decision between various approaches:

- DNN's with Batch Norm and Drop out outperformed the previous model approaches with respect to MAPE

Improvement Areas for Solution

- Additional Data
  - Using More Data to build the model with better machine configuration
  - Better Imputation of Null Values
  - Real Time Traffic Data
  - Driver Details like experience, number of trips travelled, Age , Gender
  - More Details of the Location Area such as Type of Location Airport , Tech Parks, Hospital , Any mode of public transportation and many more
- Could have used GPU in Colab and experimented with more models