# NYC Taxi Fare Prediction

Professor Dr. Wencen Wu
Fall 2019 : CMPE 257
Team 13

Farhan Bhoraniya (014506531)
Kalyani Deshmukh (011414663)
Mukesh Mogal (014529112)
Samruddhi Patil (014550094)

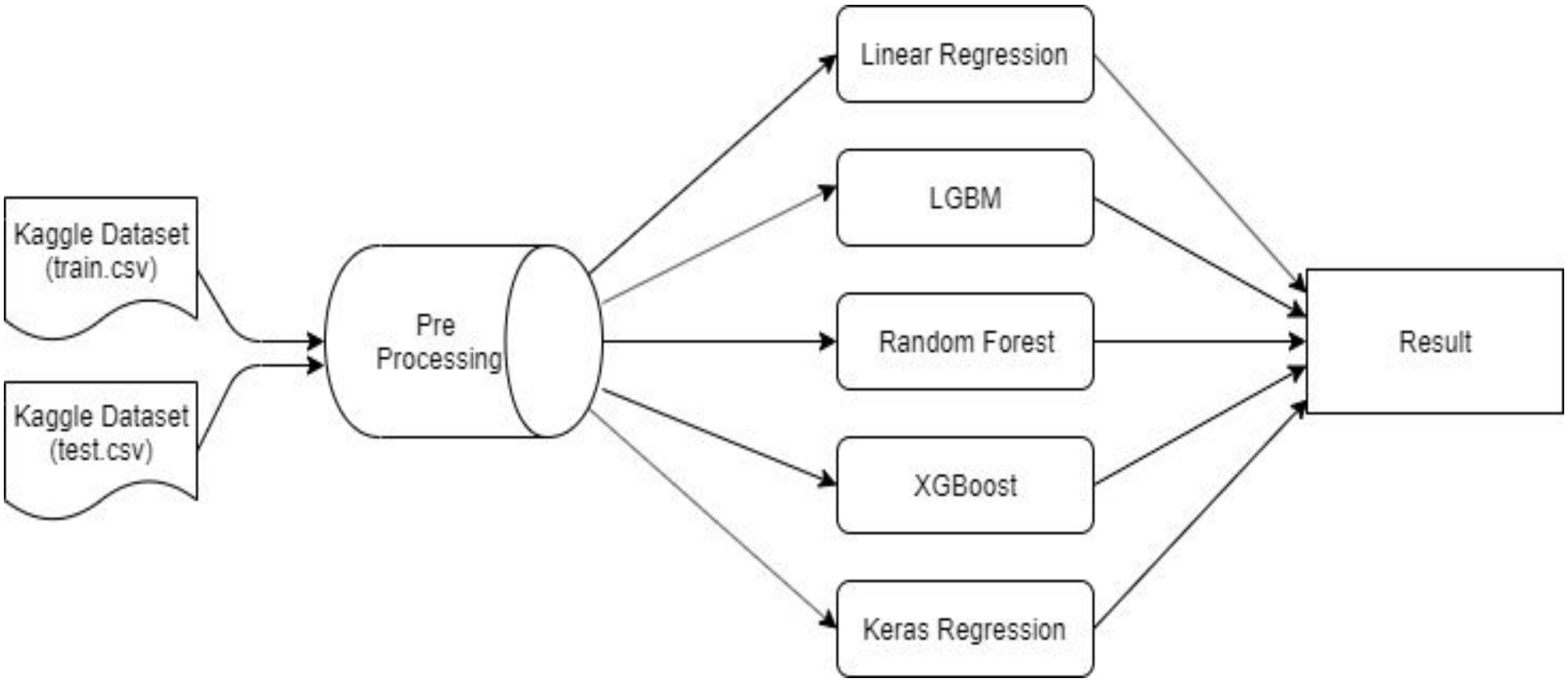
# Motivation

- Idea is to predict the NYC Taxi fare based on pickup and drop off location
- One can get approx fare by distance but it does not give you accurate fare
- Fare depends on the other factors like time of the ride, the day of the week, year of the ride as the fares increase with the inflation

# Outline

- Task
  - Predicting the NYC taxi fare
- Dataset
  - Around 55 Million rows and 7 Columns
- Models
  - Linear Regression
  - LGBM (Light Gradient Boosting Machine)
  - Random Forest
  - XGBoost
  - Keras

# Project Flow



# Before Preprocessing

| | fare_amount | pickup_longitude | pickup_latitude | dropoff_longitude | dropoff_latitude | passenger_count |
|---|---|---|---|---|---|---|
| **count** | 5.000000e+06 | 5.000000e+06 | 5.000000e+06 | 4.999964e+06 | 4.999964e+06 | 5.000000e+06 |
| **mean** | 1.134080e+01 | -7.250678e+01 | 3.991974e+01 | -7.250652e+01 | 3.991725e+01 | 1.684695e+00 |
| **std** | 9.820175e+00 | 1.280970e+01 | 8.963509e+00 | 1.284777e+01 | 9.486767e+00 | 1.331854e+00 |
| **min** | -1.000000e+02 | -3.426609e+03 | -3.488080e+03 | -3.412653e+03 | -3.488080e+03 | 0.000000e+00 |
| **25%** | 6.000000e+00 | -7.399206e+01 | 4.073491e+01 | -7.399139e+01 | 4.073404e+01 | 1.000000e+00 |
| **50%** | 8.500000e+00 | -7.398181e+01 | 4.075263e+01 | -7.398016e+01 | 4.075315e+01 | 1.000000e+00 |
| **75%** | 1.250000e+01 | -7.396711e+01 | 4.076712e+01 | -7.396367e+01 | 4.076811e+01 | 2.000000e+00 |
| **max** | 1.273310e+03 | 3.439426e+03 | 3.310364e+03 | 3.457622e+03 | 3.345917e+03 | 2.080000e+02 |

Statistical details of the training dataset
before preprocessing

# Data Preprocessing Steps

- Removed invalid number of passengers
- Changed data-types
- Removed too high or too low fares
- Removed invalid latitudes and longitudes
- Separated date time field into different fields
- Calculated Harversine distance and JFK distance

*Haversine formula:*

$$a = \sin^2(\Delta\varphi/2) + \cos\varphi_1 \cdot \cos\varphi_2 \cdot \sin^2(\Delta\lambda/2)$$

$$c = 2 \cdot \operatorname{atan2}(\sqrt{a}, \sqrt{(1-a)})$$

$$d = R \cdot c$$

*where $\varphi$ is latitude, $\lambda$ is longitude, R is earth's radius (mean radius = 6,371km);*
*note that angles need to be in radians to pass to trig functions!*

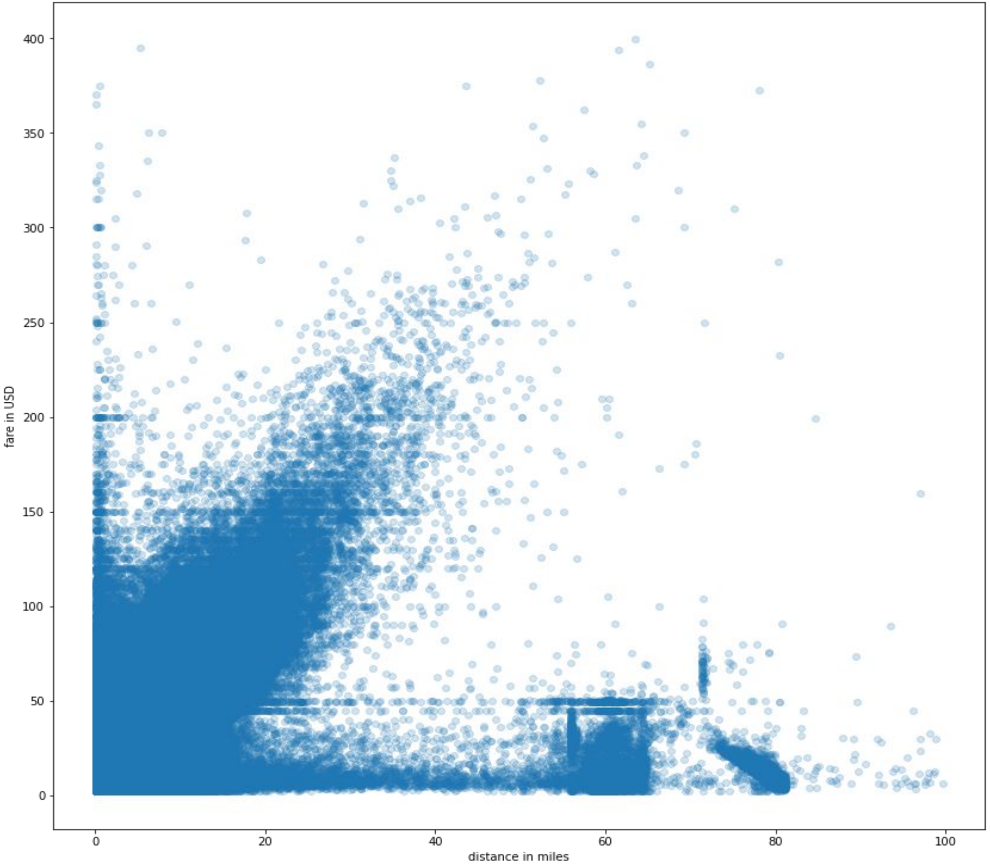
# After Preprocessing

| | fare_amount | pickup_longitude | pickup_latitude | dropoff_longitude | dropoff_latitude | passenger_count | hour | day | month | weekday | year | haversine_distnace | direction | JFK_distance |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **count** | 5.308606e+07 | 5.308606e+07 | 5.308606e+07 | 5.308606e+07 | 5.308606e+07 | 5.308606e+07 | 5.308606e+07 | 5.308606e+07 | 5.308606e+07 | 5.308606e+07 | 5.308606e+07 | 5.308606e+07 | 5.308606e+07 | 5.308606e+07 |
| **mean** | 8.139474e+00 | -4.045287e+01 | 2.022644e+01 | -4.045287e+01 | 2.022644e+01 | 1.691975e+00 | 1.351765e+01 | 1.571170e+01 | 6.268899e+00 | 3.040971e+00 | 2.011745e+03 | 2.116131e+00 | 3.186229e-01 | 1.247493e+01 |
| **std** | 9.522861e+00 | 2.544103e+01 | 1.272051e+01 | 2.544103e+01 | 1.272051e+01 | 1.307223e+00 | 6.515234e+00 | 8.685037e+00 | 3.436881e+00 | 1.949043e+00 | 1.866117e+00 | 2.492550e+00 | 1.669659e+00 | 2.160912e+00 |
| **min** | 1.110000e+00 | -7.499804e+01 | 3.903129e+01 | -7.499828e+01 | 3.901662e+01 | 1.000000e+00 | 0.000000e+00 | 1.000000e+00 | 1.000000e+00 | 0.000000e+00 | 2.009000e+03 | 1.000002e-01 | -3.141579e+00 | 2.463167e-03 |
| **25%** | 6.000000e+00 | -7.399229e+01 | 4.073662e+01 | -7.399159e+01 | 4.073563e+01 | 1.000000e+00 | 9.000000e+00 | 8.000000e+00 | 3.000000e+00 | 1.000000e+00 | 2.010000e+03 | 8.051371e-01 | -8.954245e-01 | 1.255060e+01 |
| **50%** | 8.500000e+00 | -7.398213e+01 | 4.075340e+01 | -7.398064e+01 | 4.075389e+01 | 1.000000e+00 | 1.400000e+01 | 1.600000e+01 | 6.000000e+00 | 3.000000e+00 | 2.012000e+03 | 1.363401e+00 | -1.269430e-01 | 1.289996e+01 |
| **75%** | 1.250000e+01 | -7.396851e+01 | 4.076756e+01 | -7.396560e+01 | 4.076841e+01 | 2.000000e+00 | 1.900000e+01 | 2.300000e+01 | 9.000000e+00 | 5.000000e+00 | 2.013000e+03 | 2.464340e+00 | 2.260265e+00 | 1.325358e+01 |
| **max** | 3.993300e+02 | -7.000039e+01 | 4.199717e+01 | -7.000227e+01 | 4.199811e+01 | 6.000000e+00 | 2.300000e+01 | 3.100000e+01 | 1.200000e+01 | 6.000000e+00 | 2.015000e+03 | 9.965149e+01 | 3.141593e+00 | 2.044714e+02 |

Statistical details of the training dataset
after data preprocessing

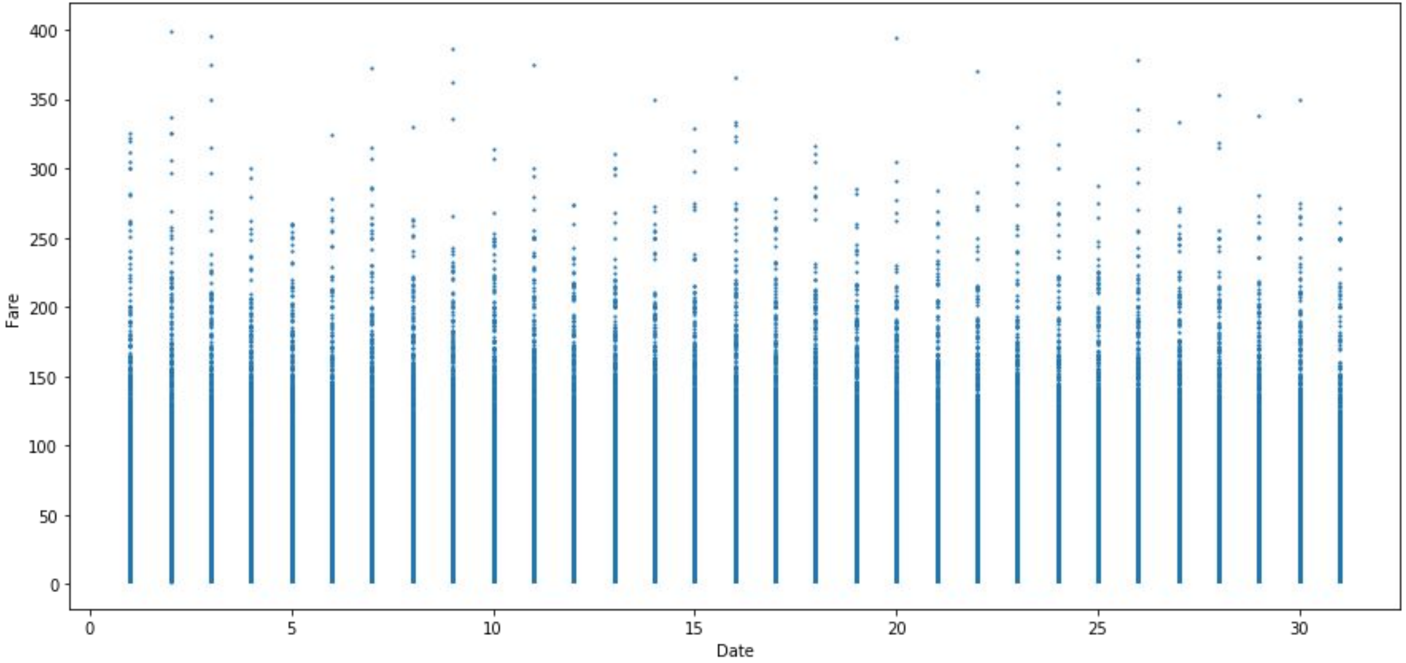
# Distance vs Fare

- This graph indicates that the fare varies with the distance almost linearly but it is also dependent on other features.



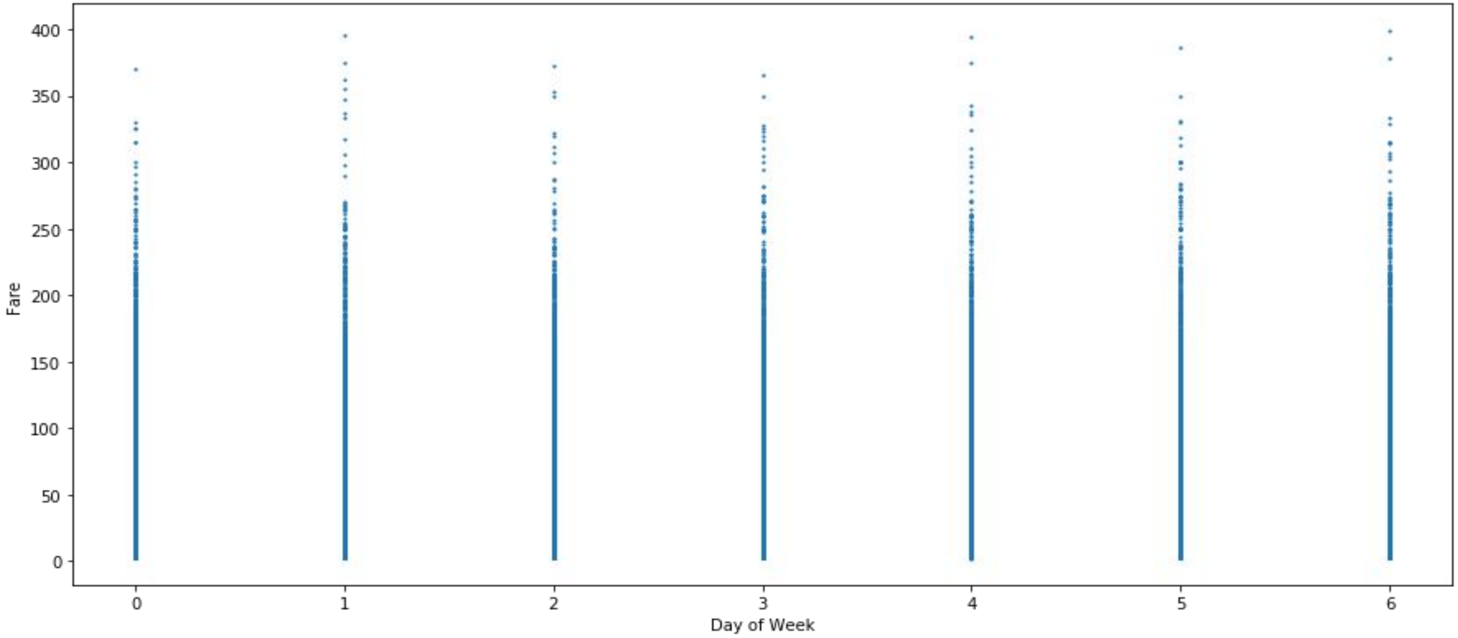
# Day of Month vs Fare

- Lack of variation of the fare w.r.t. Date make this feature insignificant.



# Day of Week vs Fare

- Similarly, day of the week is also an insignificant feature which can be removed.



# Correlation Matrix

# Model Implementation and their RMSE

| Model | RMSE |
|---|---|
| Linear Regression | 5.207 |
| LGBM | 3.381 |
| Random Forest | 3.18 |
| XGBoost | 3.047 |
| Keras Regression | 1.680 |

Root Mean Squared Error was used as a performance indicator and Keras Regression gave the best results.

# Predictions

| Data | | | Prediction | | | | |
|---|---|---|---|---|---|---|---|
| Pickup Location | Drop Off Location | Fare Amount | Linear Regression | LGBM | Random Forest | XGBoost | Keras |
| 40.77 N<br>73.97 W | 40.77 N<br>73.96 W | 10.1 | 6.645 | 6.642 | 9.64 | 6.82 | 9.58 |
| 40.74 N<br>73.98 W | 40.70 N<br>74.01 W | 16.5 | 20.866 | 13.932 | 18.66 | 17.02 | 16.09 |
| 40.73 N<br>74.00 W | 40.74 N<br>73.98 W | 7.7 | 7.3254 | 7.519 | 9.06 | 7.58 | 7.65 |
| 40.73 N<br>74.00 W | 40.82 N<br>73.92 W | 24.9 | 24.063 | 24.182 | 25.87 | 27.24 | 26.89 |

# Conclusion & Future Scope

- Results are fairy accurate and comparable with all models according to the obtained RMSE for each models.
- Models have tradeoff between accuracy, memory and execution time.
- To further improve the accuracy we can consider other features like traffic, driving speed etc.
- Also usage of Google map APIs for journey details would improve the accuracy

# Contribution

| Task | Name |
|---|---|
| Data Pre-processing | Samruddhi, Farhan |
| Feature Engineering | Kalyani, Mukesh |
| Linear Regression | Samruddhi |
| LGBM | Samruddhi |
| Random Forest | Farhan |
| XGBoost | Kalyani |
| Keras Regression | Mukesh |

# References

1. Dataset: https://www.kaggle.com/c/new-york-city-taxi-fare-prediction/data
2. Linear regression: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html
3. LGBM: https://lightgbm.readthedocs.io/en/latest/index.html
4. Random Forest: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html
5. Keras regression: https://keras.io/

# Thank You!!