# CAB PRICE PREDICTION

Gauresh Chavan | Mohit Ruke | Kaushal Chaudhary

**Problem Statement:**

Uber and Lyft's ride prices are not constant like public transport. They are greatly affected by the demand and supply of rides at a given time. So, what exactly drives this demand? The first guess would be the time of the day; times around 9 am and 5 pm should see the highest surges on account of people commuting to work/home. Another guess would be the weather; rain/snow/hotness/humidity should cause more people to take rides.

Our aim is to try to analyze the prices of these ride-sharing apps and try to figure out what factors are driving the demand.

**Data:**

With no public data of rides/prices shared by any entity, we have used a comprehensive dataset found on Kaggle which comprises of the actual cab prices data and corresponding weather conditions

The Cab ride data covers various types of cabs for Uber & Lyft and their price for the given location. You can also find if there was any surge in the price during that time. Weather data contains weather attributes like temperature, rain, cloud, etc. for all the locations taken into consideration.

**Glossary:**

- Distance: distance between source & destination
- Cab_type: Uber of lyft
- Time_stamp: epoch time when data was queried
- Destination: destination of the ride
- Source: starting point of the ride
- Price: price estimate for the ride in USD
- Surge_multiplier: the multiplier by which price as increased, default 1
- Id: unique identifier
- Product_id: uber/lyft identifier for cab-type
- Name: visible type of the cab e.g. Uber Pool, UberXL
- *Location: location name*
- *Clouds: clouds*
- *Pressure: pressure in mb*
- *Rain: rain in inches for the last hour*
- *Time_stamp: epoch time when row data was collected*

- *Humidity: humidity in %*
- *Wind: wind speed in mph*

*And a few more…*

**Methodology:**

1. Exploratory Data Analysis (EDA)
    1.1. Comparing Uber vs Lyft ride counts
    1.2. Price surge investigation
    1.3. Exploring Fare vs Distance correlation
    1.4. Effect of time on cab fare
    1.5. Does weather impact cab fare?
2. Data pre-processing
    2.1. Imputing nulls with mean values
    2.2. Segregating data frames for Uber & Lyft and dropping unnecessary columns

    *We know there is a surge multiplier factor in Lyft where the rate is multiplied during rush hours. When prices surge, Uber does not show a multiplier and instead quotes only the higher prices up front. Lyft marks up its Prime-Time pricing with a percentage: if the rate is 50%, a fare that would normally be $10 costs $15.*

    *Keeping this mind and because our data is not skewed, we want our predictions to be calculated off those surges for Lyft separately. Hence, we segregate the data post processing into two different use cases and fit models to it.*
3. Use Case 1: Lyft Price Prediction
    3.1. Recursive Feature Elimination

    *Recursive feature elimination (RFE) is a feature selection method that fits a model and removes the weakest feature (or features) until the specified number of features is reached. Features are ranked by the model's feature importance or coefficient attributes, and by recursively eliminating a small number of features per loop, RFE attempts to eliminate dependencies and collinearity that may exist in the model.*
    3.2. Train-valid-test split (70-15-15) %
    3.3. Implementing XGBoostRegressor()
    3.4. Understanding feature importance
    3.5. Hyperparameter tuning using RandomizedSearchCV()
    3.6. Performing predictions
4. Use Case 2: Uber Price Prediction (Same steps as Use case 1)

**Results:**

| | Lyft | | Uber | |
|---|---|---|---|---|
| | *Validation Set* | *Test Set (Tuned)* | *Validation Set* | *Test Set (Tuned)* |
| *Mean Absolute Error (MAE)* | 6.14 | 1.24 | 6.17 | 1.57 |
| Mean Square Error (MSE) | 87.24 | 3.21 | 73.67 | 5.78 |
| Root Mean Square Error (RMSE) | 9.34 | 1.79 | 8.58 | 2.40 |