Approach –

1. **Data Cleaning, data transformation and outlier detection**
2. **Feature engineering**
3. **Xgboost models**
4. **Catboost model**
5. **Ensemble Predictions**

Data Cleaning, data transformation and outlier detection

1. # checking for null values
2. Getting the common Id's. From the intersection value we can conclude that the testing id's are a subset of training # ID's
3. # checking for missing values
4. # converting unix time to standard time
5. Checking the distribution of the dependent variable – did log transformation because its skewed distribution.
6. Removed outliers after log transforming the dependent variable

Validation check

- Since dates are given as part of the dataset, it is essential to check whether the train and test datasets are from the same time period or different time period.

- Validation as to do K-fold cross validation or doing K-fold cross validation with respect to time
- In this case we find that the train and test dataset are for the same time period. Hence, we can do K-fold cross validaton

Feature engineering from Date time feature

1. hour_pick : hour of pick up
2. day_of_week: the day of the week the trip was done, to indetify the busy days.
3. day_of_month: which day it was in the month to account for seasonality
4. month: the month the trip was done to account for seasonality
5. is_night_time : 1 if its night time, else 0
6. late_night_time : 1 if its late night time , else 0
7. week : which week is it ?
8. min_of_pick : minute of pick
9. weather : different seasons encoded
10. quarter : which quarter is it when the trip was done

11. center_lat : latitude center (avg)
12. center_long – longitude center(avg)
13. pickup_hour_weekofyear -
14. pickup_week_hour
15. lat_diff – latitude difference between start and end latitude
16. lon_diff – Longitude difference between start and end longitude

Clustering based features

Clustered the starting latitude and longitude into clusters using Kmeans to know similar starting and ending points which might help us in determining the duration

1. start_cluster
2. end_cluster

PCA based features
Principal components of latitude and longitude combination

1. start_pca0
2. start_pca1
3. end_pca0
4. end_pca1
5. start_pca0
6. start_pca1
7. end_pca0
8. end_pca1

Metric to measure

Mean absolute error. However, I have measured the metric RMSE because RMSE has the benefit of penalizing large errors more.

Similarities
Both MAE and RMSE express average model prediction error in units of the variable of interest. Both metrics can range from 0 to ∞ and are indifferent to the direction of errors. They are negatively-oriented scores, which means lower values are better.

Differences: Taking the square root of the average squared errors has some interesting implications for RMSE. Since the errors are squared before they are averaged, the RMSE gives a relatively high weight to large errors. This means the RMSE should be more useful when large errors are particularly undesirable.

RMSE has the benefit of penalizing large errors more so can be more appropriate in some cases, for example, if being off by 10 is more than twice as bad as being off by 5. But if being off by 10 is just twice as bad as being off by 5, then MAE is more appropriate.
From an interpretation standpoint, MAE is clearly the winner. RMSE does not describe average error alone and has other implications that are more difficult to tease out and understand.
On the other hand, one distinct advantage of RMSE over MAE is that RMSE avoids the use of taking the absolute value

Build an ensemble of non linear models : XGBOOST AND CATBOOST models with 5 fold validation

I applied CatBoost, which is a machine learning algorithm that uses gradient boosting on decision trees. Since there is no linear relationship between the predictor variables and the salary, using non-linear decision boundary is good. It automatically handles categorical variables and it gives a better model without tuning in comparision to Xgboost. CatBoost has the flexibility of giving indices of categorical columns so that it can be encoded as one-hot encoding using one_hot_max_size.
one-hot encoding is used for all features with number of different values less than or equal to the given parameter value(one_hot_max_size.)

For remaining categorical columns which have unique number of categories greater than one_hot_max_size, CatBoost uses an efficient method of encoding which is similar to mean encoding but reduces overfitting.
1. Permuting the set of input observations in a random order. Multiple random permutations are generated
2. Converting the label value from a floating point or category to an integer
3. All categorical feature values are transformed to numeric values

Catboost, its an ensemble tree model which combines lot of weak predictive trees to increase overall accuracy. This is basically similar to other gradient boosting trees but in Catboost we have the following advantages

Advantages of CatBoost

- **Performance:** CatBoost provides state of the art results and it is competitive with any leading machine learning algorithm on the performance front.
- **Handling Categorical features automatically:** We can use CatBoost without any explicit pre-processing to convert categories into numbers. CatBoost converts categorical values into numbers using various statistics on combinations of categorical features and combinations of categorical and numerical features.
- **Robust:** It reduces the need for extensive hyper-parameter tuning and lower the chances of overfitting also which leads to more generalized models.

Xgboost-Pros:

- It is a very fast Ensemble algorithm
- It is a parallel implementation of Gradient Boosting machines with Regularization
- Efficient pruning, as it builds a tree to max depth and start removing splits beyond which there is no positive gain
- It has built in cross-validation

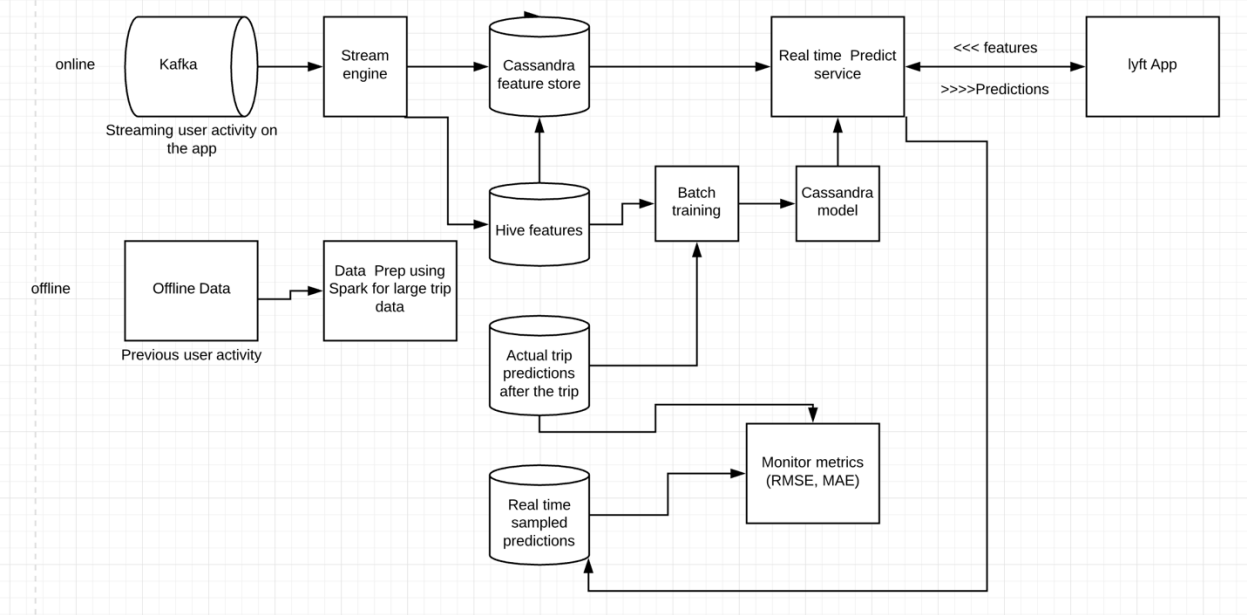
Xgboost-cons:

- Many parameters to tune
- Care should be taken to avoid overfitting

Future work given more time and computing resources

1. Hyper parameter tuning
2. Feature engineering : Aggregating average traffic features , mean encoding categorical variables
3. Run Neural networks which will automatically create features and use them in the xgboost model
4. Feature scaling which prevents the model from giving greater weightage to certain attributes as compared to others and feature selection to pruning them to reduce noise/redundancy
5. Ensemble of various models with different random seeds and use stacking to ensemble the models
6. Getting more data regarding traffic, weather and customer information, demographics will help us in having a better understanding of the prediction

Ideas for Real time implementation



Offline

In real time, we can have an offline trained model using the offline data. Batch train the model and store the model in Cassandra model repo.

Online

With live feed, we can do kafka streaming and store the data in Cassandra as key value pairs. We can use the model trained offline and predict the time taken and send it as a service to the client in the form of a restfull service API.

We can have live traffic details fed into the model real time to make accurate predictions. Given the variables considered in the initial dataset, we can generate the features in real time using spark and run the offline trained model to get a request to the client.