Doordash Delivery Duration Prediction

a) ***Models Used:***

Ensemble of the following two models are used with equal weights -

1. Gradient Boosted Tree (Boosting)
2. Random Forest (Bagging)

Parameters tuning is done to improve performance while carefully avoiding overfitting and considering overall training and test time. For both models 600 estimators with tree depth of 7 is found ideal.

Note: Scikit-learn and Pandas python libraries are used. ExtraTreeRegressor was initially considered but did not perform well so did not use as final model.

b) ***Performance Evaluation:***

Hold out testset: 33% hold out test set is used to evaluate model performance with RMSE. This is about 52k examples which is almost same 56k samples in the prediction json dataset to be comparable for unseen data to be used for prediction.

Training and CV set: 66% of the historic data with 5-fold cross validation is used.
Example RMSE from Random Forest with 100 estimators during Cross Validation.

```
RMSE : 1040.3801748473638
RMSE : 967.3122618704065
RMSE : 964.1000255910892
RMSE : 1087.2033836649118
RMSE : 1094.4228539689466
Mean CV RMSE : 1030.6837399885435 (CV set is about 26k)
```

This is on par with 1012 RMSE on hold out test set with 600 estimators but with double the amount of data - 52k vs CV's 26k data.

- **Bias vs Variance** is checked while doing parameter tuning(mostly with trial and error - e.g. adjusted # of features, tree_depth). Could use Grid Search but takes long time.
- Checked performance over parameters e.g. increasing n_estimators beyond 600 has only slight score improvement RMSE 954 with 1000 estimators vs 956 with 600 from Gradient Boosted Regressor - could plot validation curve.
- Experimented with # of examples e.g. instead of dropping 16k missing values for no_of_onshift(busy)_dashers, with more data from missing value imputation, model performed better - learning curve could be used as well.

c) ***Data Processing and Cleanup:*** Categorical feature “store_primary_category” is converted to numerical data type for model’s use. Several missing values is imputed with appropriate analysis and handful of the missing values were dropped as either imputing reduced performance(e.g. market_id) or did not improve performance.

Features with missing values:

The following three features have most number of missing values. Instead of throwing the rows with these missing values, these were filled with 0s(initially tried filling with average values based on each market but performance didn’t improve)

Feature	# of missing values
total_onshift_dashers	16262
total_busy_dashers	16262
total_outstanding_orders	16262

“store_primary_category” has 4760 missing values which is replaced with “other” category. This improved model’s performance a bit.

“order_protocol” has 995 missing values which is replaced with most frequent value, improves performance.

“estimated_store_to_consumer_driving_duration” has 526 missing values which is dropped as replacing with mean value reduce model’s performance .

“market_id” has 987 missing values. Dropped these samples with market_id missing values. Filling in with most frequent market or market with average overall delivery duration worsens performance.

“actual_delivery_time” has only 7 missing values and dropped these rows as no. of rows with missing value for this feature is very low and this feature is not available in the prediction dataset.

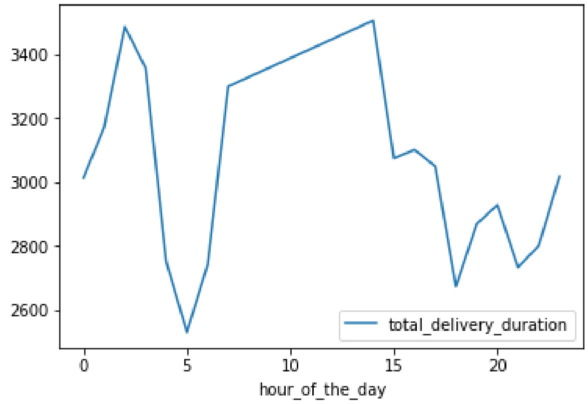
Following two features are dropped as delivery duration and hour is calculated:
'created_at', 'actual_delivery_time'(only in historic dataset)

d) ***Feature Engineering:***

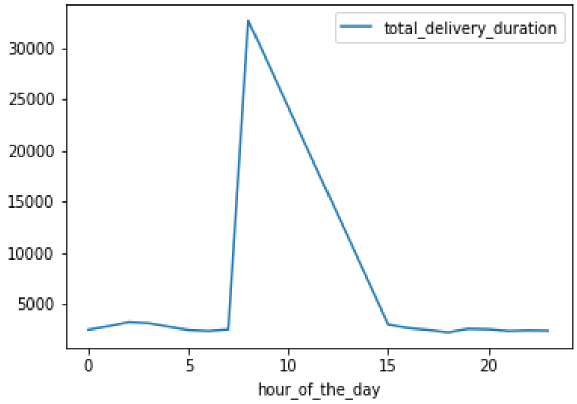
Crafted the following features from the existing features.

1. hour_of_the_day = hour of the day when the was order created_at
2. total_orders_without_dashers = total_outstanding_orders - total_onshift_dashers
3. total_available_dashers = total_onshift_dashers - total_busy_dashers
4. estimated_order_driving_duration = estimated_order_place_duration + estimated_store_to_consumer_driving_duration

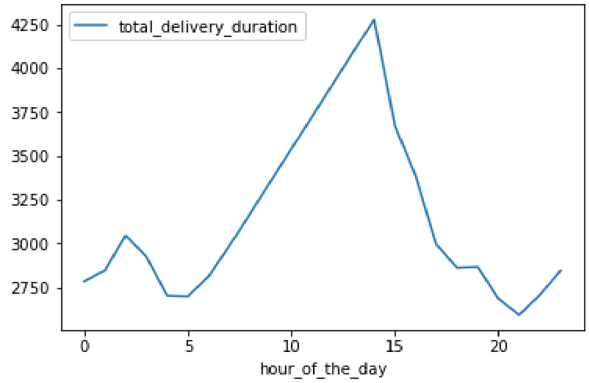
Hour_of_the_day has a high impact on the delivery duration and varies among market. Market 2 and Market 4 has much higher delivery delay than other four markets and peaks around office start time i.e 8am. Market 5 peaks around 2am in the morning possibly because of smaller city/market with no available dashers in early morning. Other 3 markets(market 1 also around 2am) peaks at lunch hour 12-2pm.



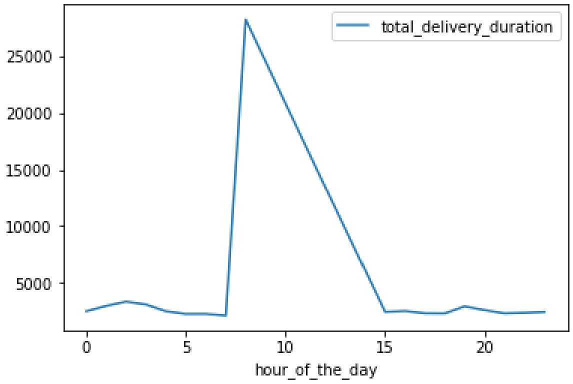
Market_id = 1



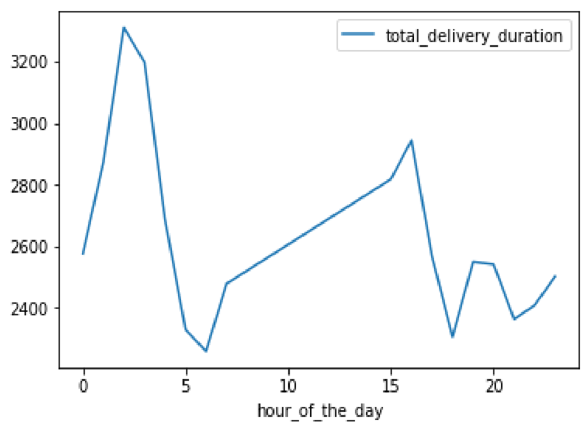
Market_id = 2



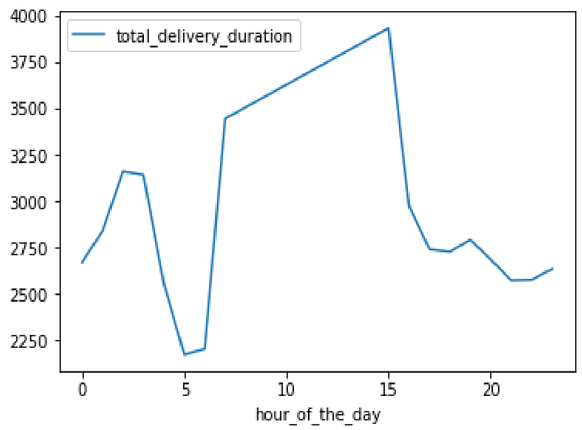
Market_id = 3



Market_id = 4



Market_id = 5

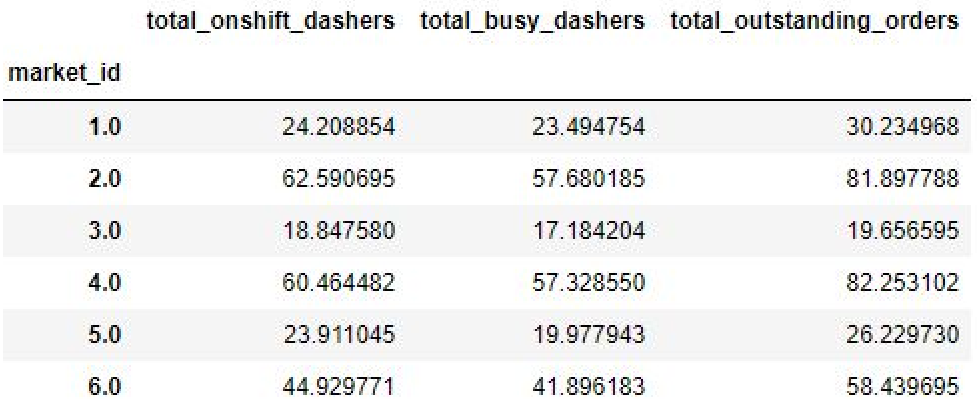


Market_id = 6

Market 2 and 4 seems to have highest # of orders as well as average # of dhasers. total_available_dashers = total_onshift_dashers - total_busy_dashers and total_orders_without_dashers = total_outstanding_orders - total_onshift_dashers have huge impact on the performance especially total_orders_without_dashers.

Market	#of order entries
2.0	53729
4.0	46430
1.0	37313
3.0	22642
5.0	17347
6.0	13946

Total orders



	total_onshift_dashers	total_busy_dashers	total_outstanding_orders
market_id			
1.0	24.208854	23.494754	30.234968
2.0	62.590695	57.680185	81.897788
3.0	18.847580	17.184204	19.656595
4.0	60.464482	57.328550	82.253102
5.0	23.911045	19.977943	26.229730
6.0	44.929771	41.896183	58.439695

Average # of dashers

Feature Importance from the models:

Feature	Gradient Boosted Trees	Random Forest
market_id	0.04365669029023432	0.05105753993343799
store_id	0.18214188346556046	0.008411425227085245
store_primary_category	0.06942987878843733	0.00713459871191111
total_items	0.028123440590522714	0.0034125112731577334
subtotal	0.08457600988596115	0.08918174593810357
num_distinct_items	0.022488186412864604	0.003890252016298437
max_item_price	0.08640867270756157	0.011283875105618514
total_outstanding_orders	0.10925444499928534	0.009426241287084301
estimated_store_to_consumer_driving_duration	0.07946245864186789	0.01195474587853766
estimated_order_driving_duration	0.0866257010524083	0.2167237353852846
total_available_dashers	0.04098652060290599	0.009793886579835863
hour_of_the_day	0.05961968287284677	0.07584599786418174
total_orders_without_dashers	0.1072264296895435	0.3892143431190055

e) *Other approaches for future improvement:*

- Stacking or Blending could be tried with the dataset to further improve the performance - require more development time.
- Also dataset could be splitted market wise and models could be generated for individual markets(or similar markets e.g. busy city with heavy volume of orders)
- Could experiment with other models like Neural Nets. If the dataset was much bigger XGBoost(for faster training) or CNNs with GPUs could be tried.

List recommendations to reduce delivery time:

a) Increase # of dashers during busy hours based on each market's peak hours when delivery delay is highest(see figures in Feature Engineering section)

b) Increase dashers' payment during peak time as subtotal/max price has high influence in delivery duration

c) Store_id seems to have high feature importance-predictive of delivery duration, so identifying stores that tends to delay in processing order, lack of available parking etc might be helpful

d) Increase dashers' payment or route more dashers when # of outstanding orders outnumbers # of on shift dashes - total_orders_without_dashers seems to have high feature importance