Project Proposal
Doordash Delivery Time Prediction

Paris Patel

Engineering Management, San Jose State University

BUS 235C: Data Mining

Dr. Shaonan Tian

November 4, 2022

Background

The dataset taken into consideration, "Delivery Time Prediction" was provided as a take-home assignment by Doordash during one of their interview processes. This data is a subset of the deliveries received by Doordash in early 2015 in a few subset of cities. It contains about 200,000 rows, each corresponding to one unique delivery and has sixteen different variables detailing order time and delivery time and other related factors.

Objective

The delivery time prediction is similar to determining ETA (expected time of arrival at destination) on Google or Apple maps and expected time of order delivery on e-commerce websites like Amazon and Ebay. For such software applications, accuracy of time estimation is essential to enhance customer experience. Hence, purpose of conducting this study is to learn what attributes could affect delivery time prediction and what would be the best way to predict it. It will also help to understand what business tradeoffs will be accepted and what will be the no-goes for the final model to be selected.

Delivery time prediction has long been a part of logistic cycle in ecommerce and delivery service providers. Customer satisfaction is highly related to on time delivery of products or services. Owing to the current pandemic, the demand has moved to online supply chain business and the pressure has been more than ever on increasing customer satisfaction and improving user experience. Hence, it became increasingly more important to accurately predict time to gain more profits in the business.

About the Data

Data Source

This data has been selected from the Data Projects section of Stratascratch. The details of the data can be found here: https://platform.stratascratch.com/data-projects/delivery-duration-prediction

Data Collection

This data was collected by Doordash as a part of its daily operations. It dates back to early 2015 and includes subset of the cities where Doordash operated and has 197,428 records.

Variables

This dataset has 16 variables:

Time features

- `market_id`: A city/region in which DoorDash operates, e.g., Los Angeles, given in the data as an id

- `created_at`: Timestamp in UTC when the order was submitted by the consumer to DoorDash. (Note this timestamp is in UTC, but in case you need it, the actual timezone of the region was US/Pacific)

- `actual_delivery_time`: Timestamp in UTC when the order was delivered to the consumer

Store features

- `store_id`: an id representing the restaurant the order was submitted for
- `store_primary_category`: cuisine category of the restaurant, e.g., italian, asian
- `order_protocol`: a store can receive orders from DoorDash through many modes. This field represents an id denoting the protocol

Order features

- `total_items`: total number of items in the order
- `subtotal`: total value of the order submitted (in cents)
- `num_distinct_items`: number of distinct items included in the order
- `min_item_price`: price of the item with the least cost in the order (in cents)
- `max_item_price`: price of the item with the highest cost in the order (in cents)

Market features

DoorDash being a marketplace, we have information on the state of marketplace when the order is placed, that can be used to estimate delivery time. The following features are values at the time of created_at (order submission time):

- `total_onshift_dashers`: Number of available dashers who are within 10 miles of the store at the time of order creation
- `total_busy_dashers`: Subset of above total_onshift_dashers who are currently working on an order
- `total_outstanding_orders`: Number of orders within 10 miles of this order that are currently being processed.

Predictions from other models

We have predictions from other models for various stages of delivery process that we can use:

- `estimated_order_place_duration`: Estimated time for the restaurant to receive the order from DoorDash (in seconds)
- `estimated_store_to_consumer_driving_duration`: Estimated travel time between store and consumer (in seconds)

Organizing Data

Following variables were converted to categorical class: market_id, store_id, store_primary_category, order_protocol. The variables demonstrating time features were converted to date-time class: created_at and actual_delivery_time.

Further, following new columns were also added to the dataset:

- actual_delivery_duration: time duration between the order created (created_at) and and delivery of the order (actual_delivery_time). This is the target variable we will try to predict using different regression and classification models.
- busy_dasher_ratio: ratio of busy dashers to toal dashers during the time of the order
- estimated_non_prep_duration: sum of estimated_order_place_duration and estimated_store_to_consumer_driving_duration.
- percent_distinct_item_total was added later to make more
- avg_price_per_item crfrd
- price_range_of_items csdc

Data Cleanup

There were missing values in certain columns of the data which were handled in the following manners:

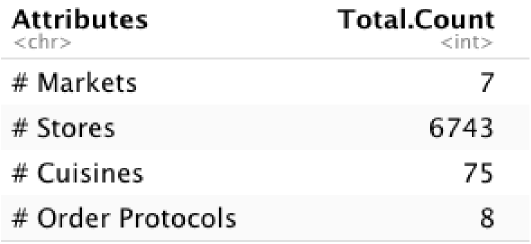
- store_primary_category: The missing values here were replaced by the most frequent primary category associated to the store_id.
- busy_dasher_ratio: There were erratic values in the values of total available values and busy dashers, which resulted in infinite values. The records with these values were removed as it was an error made during data collection and cannot be handled otherwise.
- Missing values that couldn't be filled were deleted.

Preliminary Analysis

This section will focus on descriptive statistics of the data, including the central tendencies, distribution of data and correlation between different variables.

Categorical Values

The first step was to understand how many markets, stores, cuisines and order protocols were included in the data. Cuisines and order protocols could be the factors that could affect the total delivery time.



Attributes <chr>	Total.Count <int>
# Markets	7
# Stores	6743
# Cuisines	75
# Order Protocols	8

Table 1 Categorical variables

Distribution

The distribution of markets were taken into considered. The dataset doesn't seemed balanced when considering the number of records taken from the market. However, it shouldn't affect the final prediction of the time taken for the delivery, since the essential aspect of time taken would be the distance between the store and delivery address.

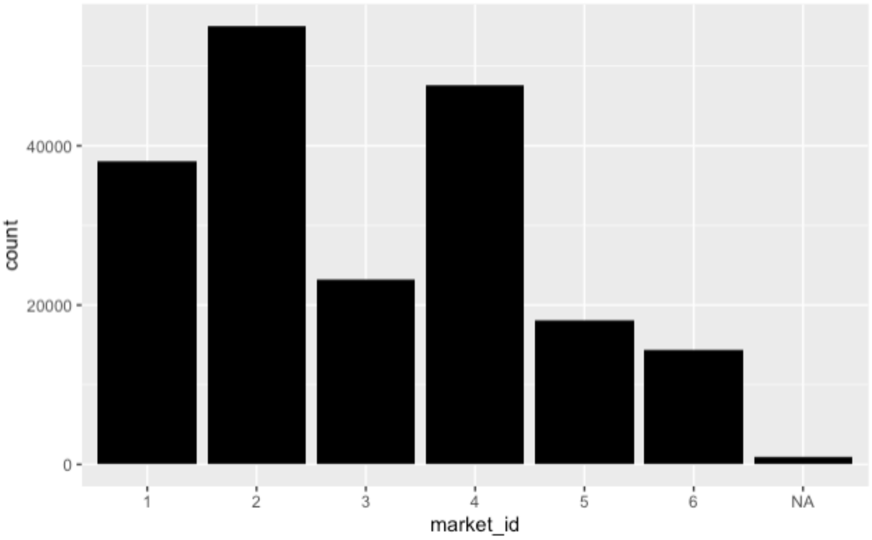


Figure 1 Distribution of market id's

For distribution, it seems important to study how often different protocols were used to place an order. As can be seen from the bar plot below, order protocol 1 was the most used, followed by protocol 3 and protocol 5.

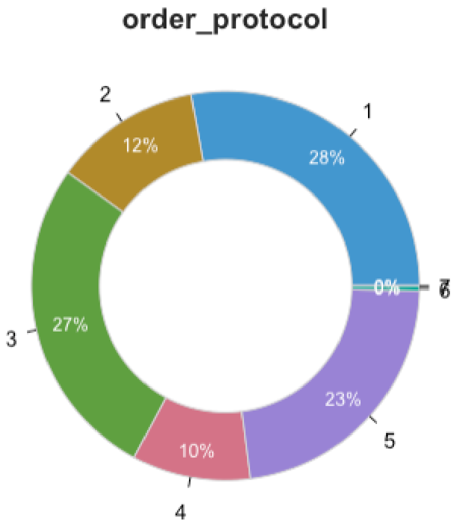
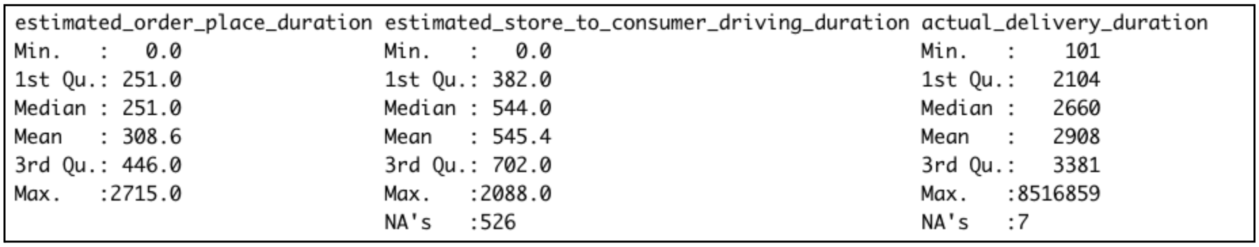


Figure 2 Order Protocol Distribution

Central Tendencies

`actual_delivery_duration` has a very high range of values as can be noticed from Table 2. The distribution of these values is right skewed as the mean and median have huge difference and mean is greater than median. This can be confirmed with the distribution plot of `actual_delivery_duration` in figure 3.



```
estimated_order_place_duration estimated_store_to_consumer_driving_duration actual_delivery_duration
Min.   :   0.0                 Min.   :   0.0                               Min.   :    101
1st Qu.: 251.0                 1st Qu.: 382.0                               1st Qu.:   2104
Median : 251.0                 Median : 544.0                               Median :   2660
Mean   : 308.6                 Mean   : 545.4                               Mean   :   2908
3rd Qu.: 446.0                 3rd Qu.: 702.0                               3rd Qu.:   3381
Max.   :2715.0                 Max.   :2088.0                               Max.   :8516859
                               NA's   :526                                  NA's   :7
```

Table 2 Central tendencies of time duration estimates

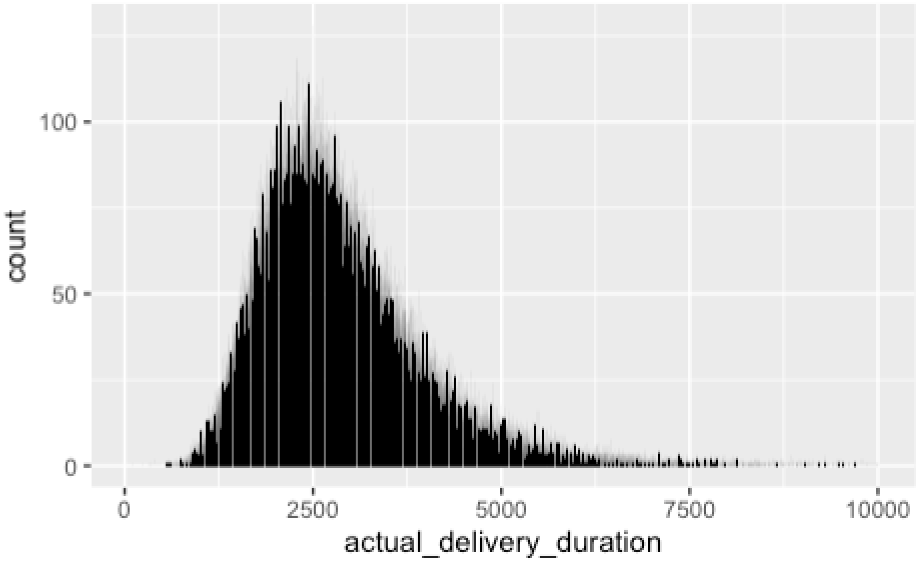


Figure 3 Distribution of `actual_delivery_duration`

Redundancy Removal:

There were 100 columns in the dataset after creating dummies for the categorical variables. Out of these, the redundant variables were removed based on the coefficient of correlation. As it is evident from the figure 4, there was an issue with store_primary_categoryindonesian as it had a 0 standard deviation. The top correlated variables that were removed were `total_busy_dashers`, `total_onshift_dashers`, `market_id`, and `order_protocol`. Correlation coefficients greater than 0.5 were considered high and the features were removed. After correcting data for redundancy, 82 variables were remaining. The correlation values before and after removal of correlated values can be seen in tables 4 and 5.

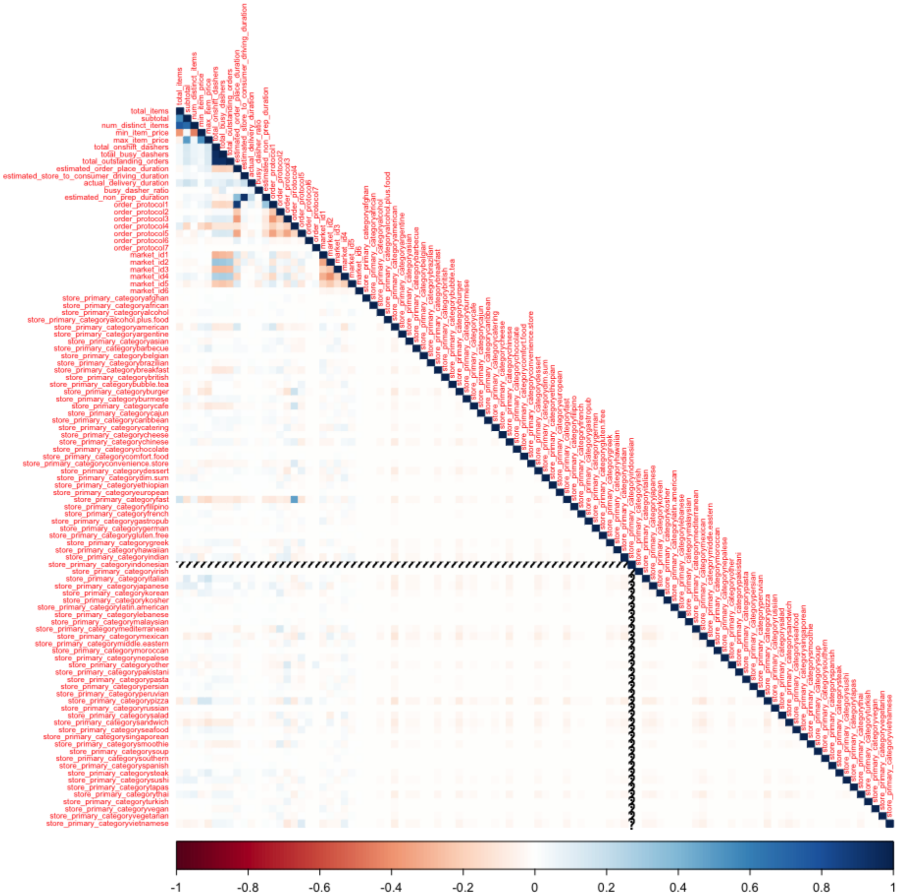
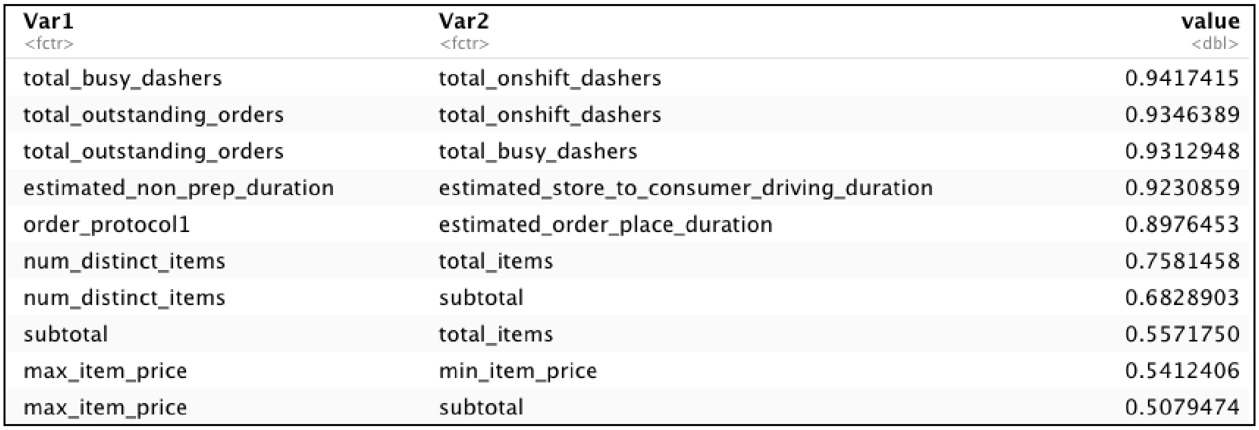
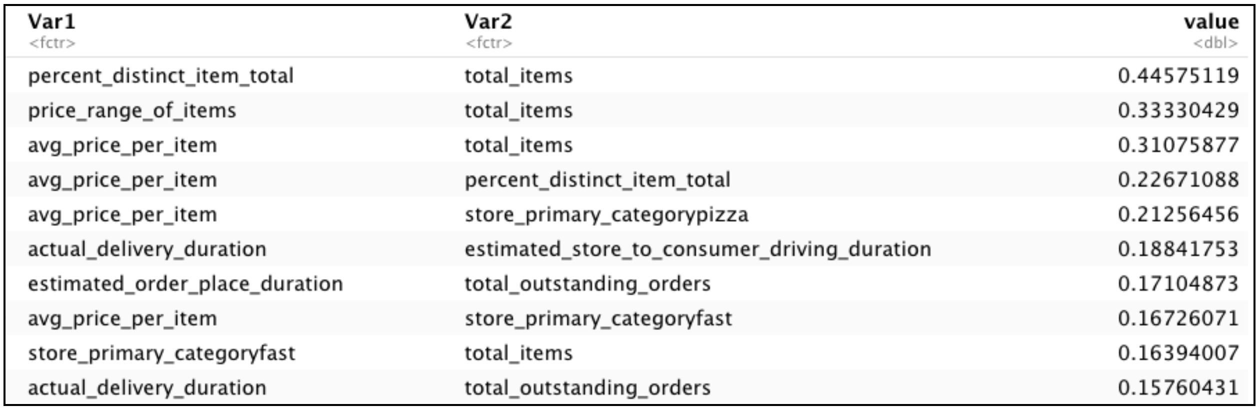


Figure 4 Correlation plot



Var1 <fctr>	Var2 <fctr>	value <dbl>
total_busy_dashers	total_onshift_dashers	0.9417415
total_outstanding_orders	total_onshift_dashers	0.9346389
total_outstanding_orders	total_busy_dashers	0.9312948
estimated_non_prep_duration	estimated_store_to_consumer_driving_duration	0.9230859
order_protocol1	estimated_order_place_duration	0.8976453
num_distinct_items	total_items	0.7581458
num_distinct_items	subtotal	0.6828903
subtotal	total_items	0.5571750
max_item_price	min_item_price	0.5412406
max_item_price	subtotal	0.5079474

Table 4 Correlation values before removal of variables

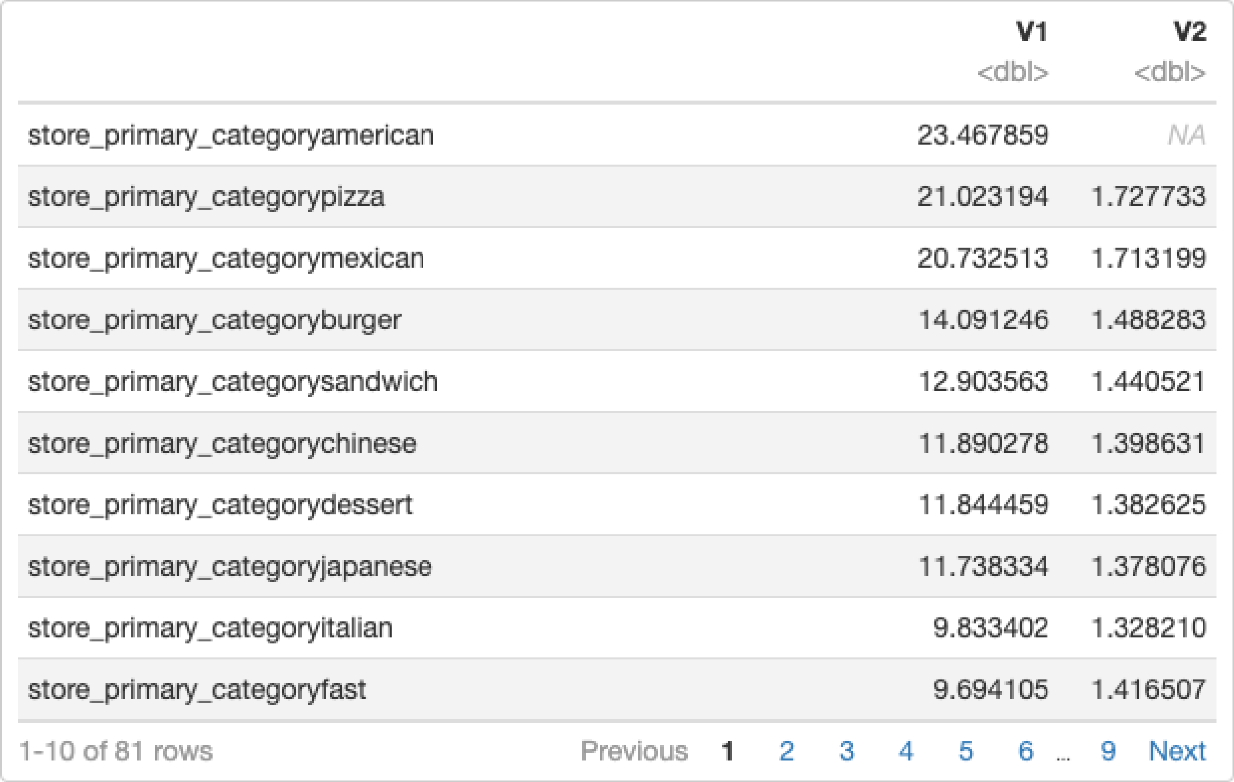


Var1	Var2	value
<fctr>	<fctr>	<dbl>
percent_distinct_item_total	total_items	0.44575119
price_range_of_items	total_items	0.33330429
avg_price_per_item	total_items	0.31075877
avg_price_per_item	percent_distinct_item_total	0.22671088
avg_price_per_item	store_primary_categorypizza	0.21256456
actual_delivery_duration	estimated_store_to_consumer_driving_duration	0.18841753
estimated_order_place_duration	total_outstanding_orders	0.17104873
avg_price_per_item	store_primary_categoryfast	0.16726071
store_primary_categoryfast	total_items	0.16394007
actual_delivery_duration	total_outstanding_orders	0.15760431

Table 5 Correlation values after removal of variables

Multicollinearity Test

Variance Inflation Factor was used to check multicollinearity and there were 3 variables that had VIF score of over 15. Variables were dropped iteratively to reduce the VIF score. Only one variable had to be removed from the 82 remaining variables, which brought down the VIF score for all variables to less than 2 (Refer Table 6).



	V1	V2
	<dbl>	<dbl>
store_primary_categoryamerican	23.467859	NA
store_primary_categorypizza	21.023194	1.727733
store_primary_categorymexican	20.732513	1.713199
store_primary_categoryburger	14.091246	1.488283
store_primary_categorysandwich	12.903563	1.440521
store_primary_categorychinese	11.890278	1.398631
store_primary_categorydessert	11.844459	1.382625
store_primary_categoryjapanese	11.738334	1.378076
store_primary_categoryitalian	9.833402	1.328210
store_primary_categoryfast	9.694105	1.416507

1-10 of 81 rows

Table 6 VIF scores before (V1) and after (V2) removal of colliear variables

Outlier handling

The distribution is right skewed, hence great amount of data could be lost if the outliers were removed. Only one data point that was extremely far from the distribution (`actual_delivery_duration` = 332482) was removed from the data.

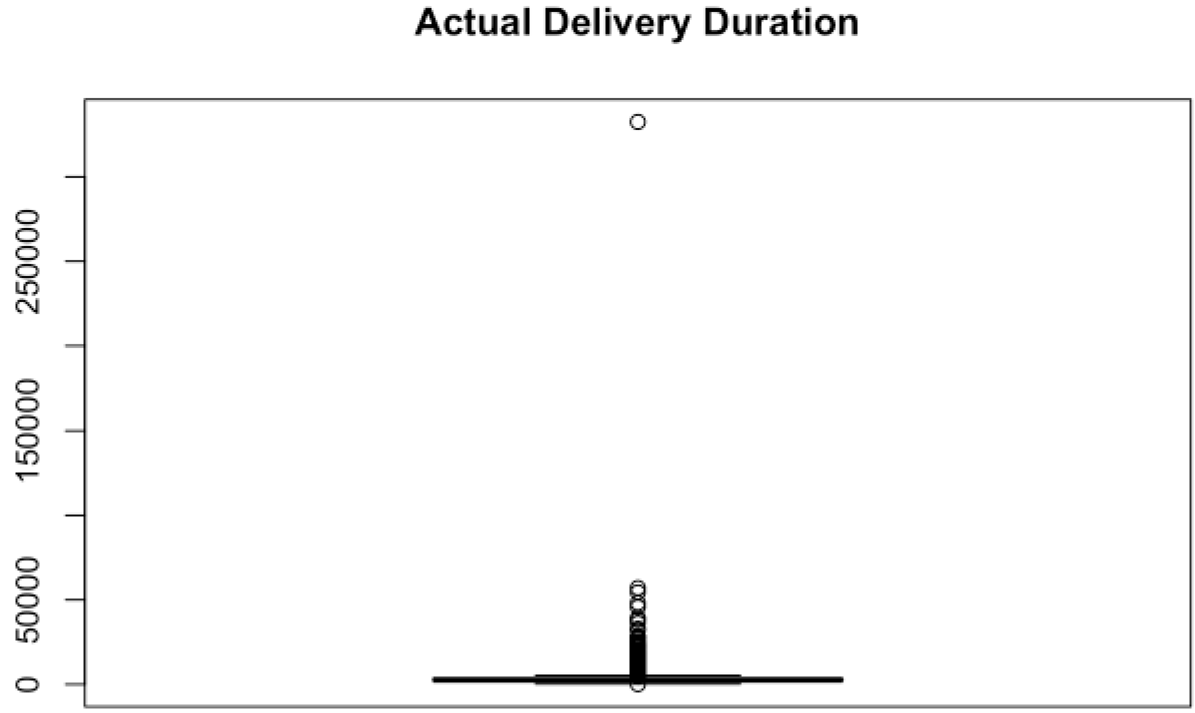


Figure 5 Outlier detection for `actual_delivery_duration`

Feature Selection

After the preprocessing of the data, there were 80 response variables still remaining in the dataset. It'd be great to use all the variables, but due to limitation on computational capacities and complexities involved, tradeoffs are required to be made to use the resources efficiently while also not compromising on the quality of the models.

Forward step method was used to determine the number of variables to use for the model. Based on the RMSE and Adjusted R^2 score, at least 20 variables were required to get an efficient model. The minimum RMSE that could be achieved as displayed by forward step seelction was 1073.277 seconds, which was about 0.1 seconds less than the median RMSE achieved through different combinations of variables. Maximum Adjusted R^2 was achieved with 58 variables, however the difference achieved with introducing so many variables was insignificant. Hence as a balance, it was decided to move forward with 40 variables. These were selected using coef() function.

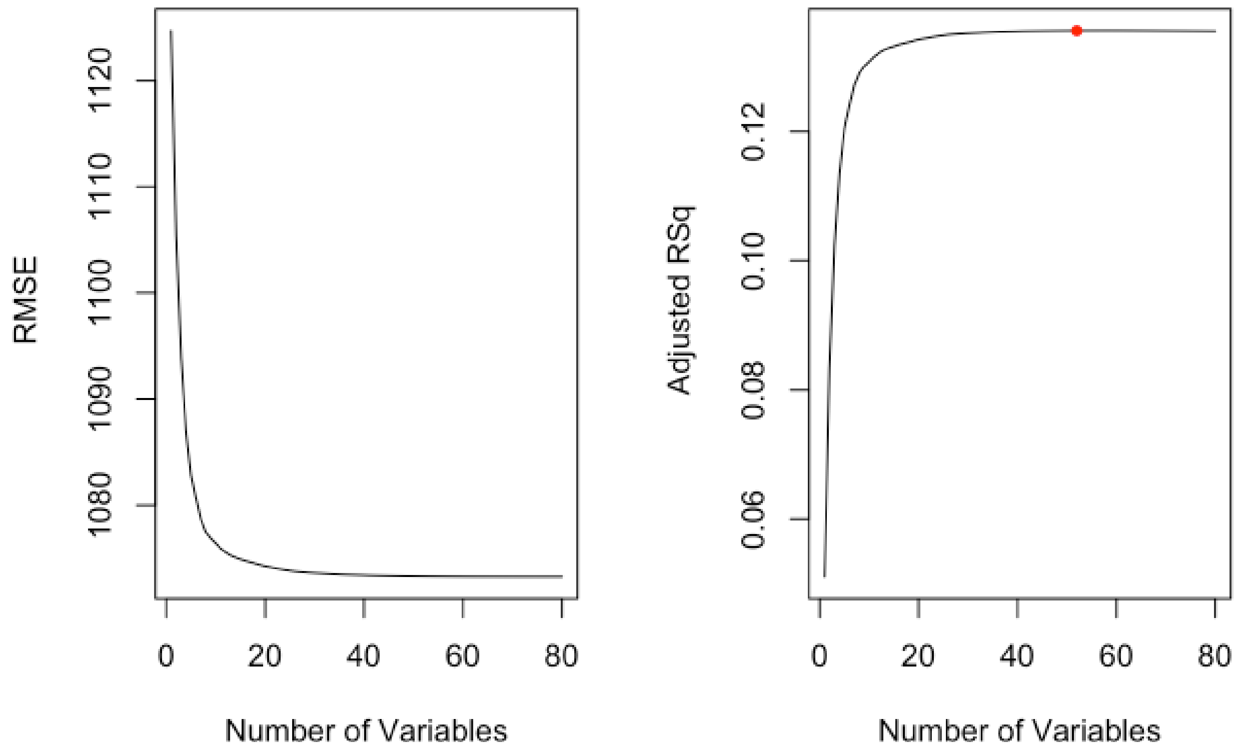


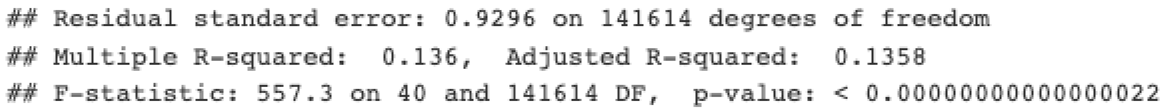
Figure 6 RMSE and Ajusted R^2 for forward step feature selection method

Model building and Evaluation

To decide which regressor will have best performance, I decided to study performances of linear regression, random forest and gradient boosting models. Details of those models can be found below.

Linear Regression

Since the F value is greater than 1, it is clear from the summary that there is a significant correlation between the predictor and responder variables. Nonetheless, the model is reasonably fit given that the overall p-value is lesser than 2.2e-16.



```
## Residual standard error: 0.9296 on 141614 degrees of freedom
## Multiple R-squared:  0.136,  Adjusted R-squared:  0.1358
## F-statistic: 557.3 on 40 and 141614 DF,  p-value: < 0.00000000000000022
```

Figure 7 Model evaluation of Linear Regressor

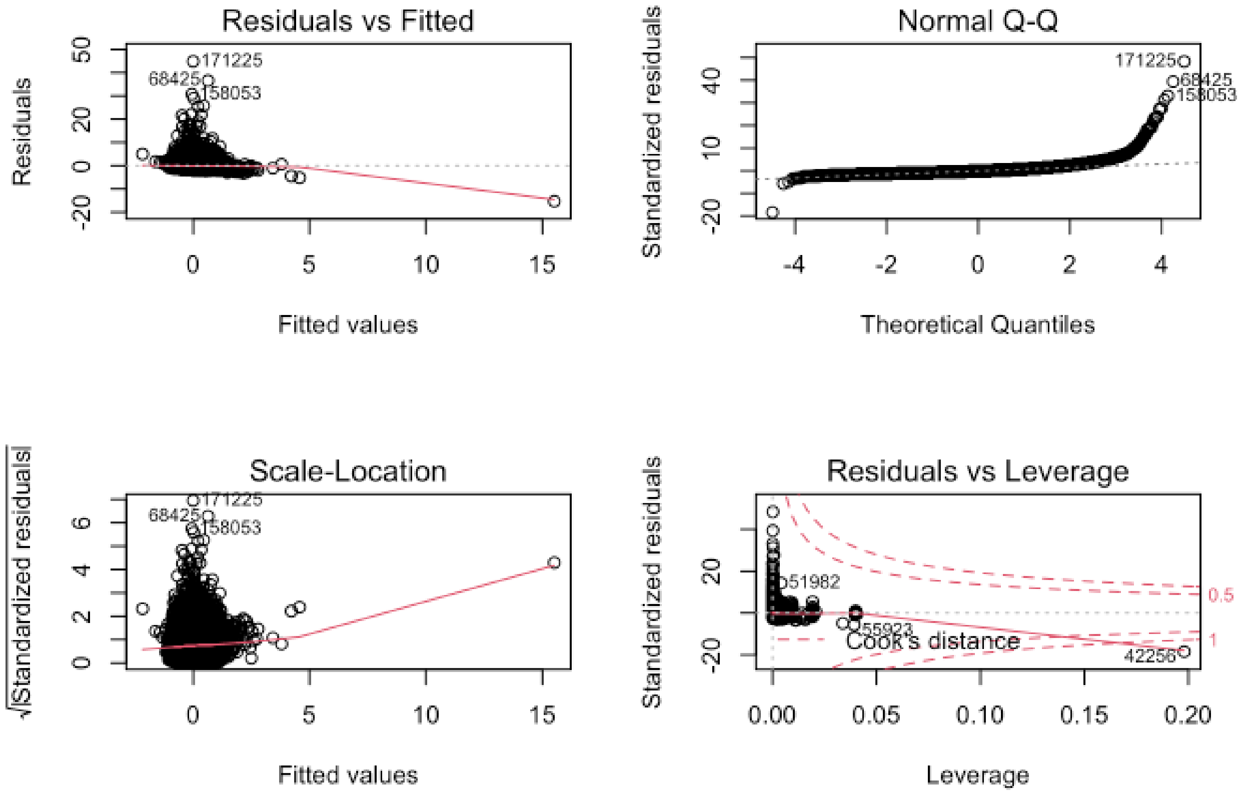
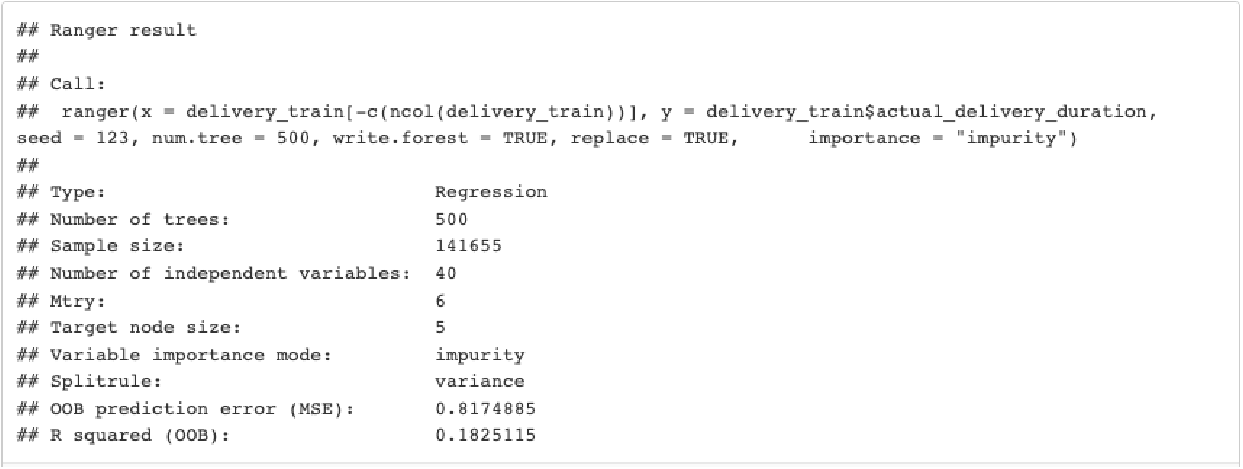


Figure 8 Residual Analysis of Linear Regressor

Random forest

Due to heavy computation requirements, randomForest() function took too long to provide the required output. Hence ranger() function from ranger package was used to generate this random forest model. On scaled variables, the MSE achieved is 0.82 with an R^2 of 0.182. The trees were generated on basis of impurity of the nodes.

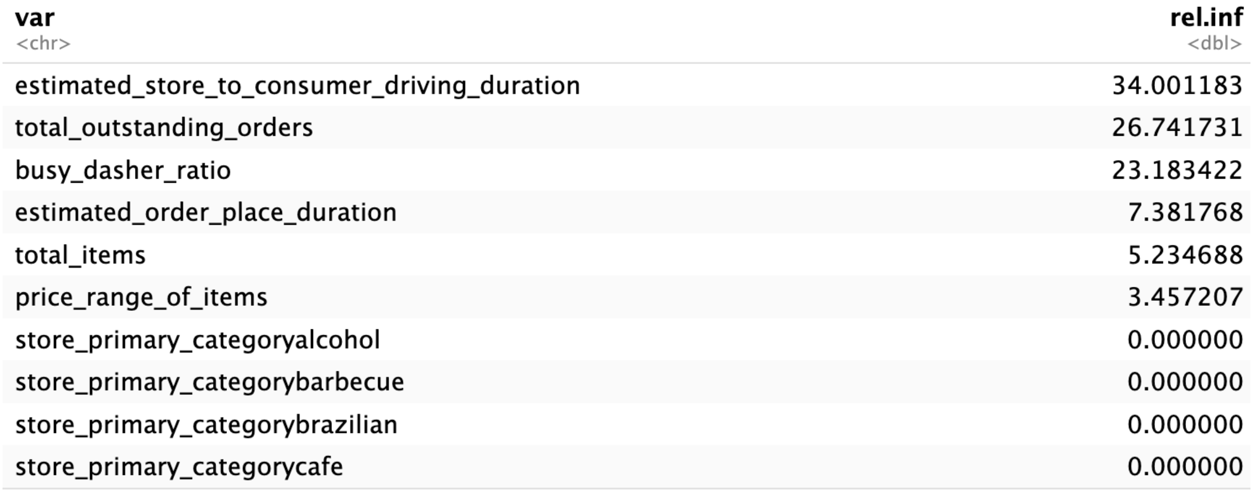


```
## Ranger result
##
## Call:
##  ranger(x = delivery_train[-c(ncol(delivery_train))], y = delivery_train$actual_delivery_duration,
seed = 123, num.tree = 500, write.forest = TRUE, replace = TRUE,      importance = "impurity")
##
## Type:                             Regression
## Number of trees:                  500
## Sample size:                      141655
## Number of independent variables:  40
## Mtry:                             6
## Target node size:                 5
## Variable importance mode:         impurity
## Splitrule:                        variance
## OOB prediction error (MSE):       0.8174885
## R squared (OOB):                  0.1825115
```

Figure 9 Random Forest Model

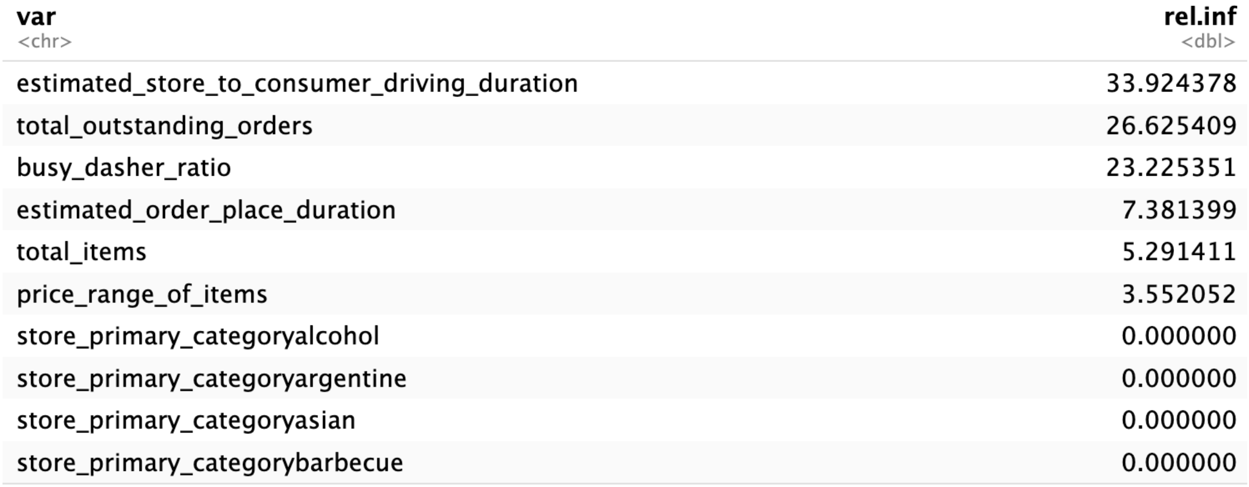
Gradient Boost

From the results obtained, it can be seen that that the `estimated_store_to_consumer_driving_duration`, `total_outstanding_orders`, and `busy_dasher_ratio` have greatest relative influence in predicting `actual_delivery_duration`.



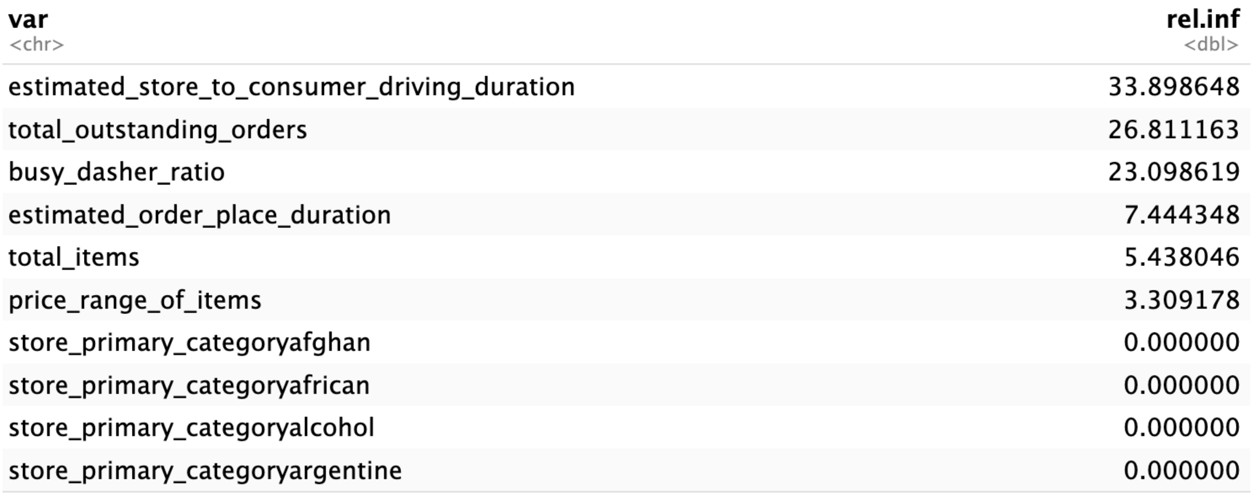
var <chr>	rel.inf <dbl>
estimated_store_to_consumer_driving_duration	34.001183
total_outstanding_orders	26.741731
busy_dasher_ratio	23.183422
estimated_order_place_duration	7.381768
total_items	5.234688
price_range_of_items	3.457207
store_primary_categoryalcohol	0.000000
store_primary_categorybarbecue	0.000000
store_primary_categorybrazilian	0.000000
store_primary_categorycafe	0.000000

Table 7 Relative Influence with 40 variables



var <chr>	rel.inf <dbl>
estimated_store_to_consumer_driving_duration	33.924378
total_outstanding_orders	26.625409
busy_dasher_ratio	23.225351
estimated_order_place_duration	7.381399
total_items	5.291411
price_range_of_items	3.552052
store_primary_categoryalcohol	0.000000
store_primary_categoryargentine	0.000000
store_primary_categoryasian	0.000000
store_primary_categorybarbecue	0.000000

Table 8 Relative Influence with 40 variables



var <chr>	rel.inf <dbl>
estimated_store_to_consumer_driving_duration	33.898648
total_outstanding_orders	26.811163
busy_dasher_ratio	23.098619
estimated_order_place_duration	7.444348
total_items	5.438046
price_range_of_items	3.309178
store_primary_categoryafghan	0.000000
store_primary_categoryafrican	0.000000
store_primary_categoryalcohol	0.000000
store_primary_categoryargentine	0.000000

Table 9 Relative Influence with 58 variables

Results

Main metric to be considered during this model development was root mean square error. The RMSE for random forest regressor is lowest, but due to high computational requirements, I'll skip using it. The next best option is to use linear regressor as our model for duration prediction.

	Linear Regressor	Random Forest Regressor	Gradient Boosting Regressor
In-sample	1067.89	776.74	1077.17
Out-of-sample	1095.52	1067.73	1107.28

Table 10 In-sample and Out-of-sample RMSE Performance

# Variables	Linear Regressor	Random Forest Regressor	Gradient Boosting Regressor
20	3075.321	3075.313	1177.619
40	1225.248	1233.432	1177.582
58	1225.488	1225.716	1177.582

Table 11 Out-of-sample RMSE Performance

Conclusion

The best model to use for prediction considering the tradeoffs between the resource capacity and performance will be the classic old linear regression model. The performance maybe improved using other models such as quadratic regression, XGBoost or Light GBM which are greatly used recently in the industry.

References

Potters, C. (2022, July 26). *Variance Inflation Factor (VIF)*. Investopedia. Retrieved December 6, 2022, from https://www.investopedia.com/terms/v/variance-inflation-factor.asp#

StrataScratch - Delivery Duration Prediction. (n.d.). StrataScratch. Retrieved November, 2022, from https://platform.stratascratch.com/data-projects/delivery-duration-prediction

Wickham, H., & Grolemund, G. (2016). *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data*. O'Reilly.

Wright, M. N., Wager, S., & Probst, P. (2022, June 18). ranger: A Fast Implementation of Random Forests. Retrieved December 6, 2022, from https://cran.r-project.org/web/packages/ranger/ranger.pdf