DOORDASH: **TOTAL DELIVERY DURATION PREDICTION**

Alphonse N. Akakpo: alpnelson5@gmail.com

1 Introduction

1.1 Background

The prediction of total delivery duration is an interesting problem due to the challenges involved in getting the most appropriate metric for model evaluation which will be consistent with the key performance indicators (KPI) of the business entity. Another challenge is getting the best set of features for the model. Besides the set of features provided, there are other features such as the *time of the day* and *the day of the week* which may be significant in predicting the total delivery duration. These two features can provide information on the traffic pattern at different times of the day or for different days of the week.

1.2 Project Summary

Just like any machine learning model, the metric for evaluation it is of utmost importance. During the initial phase model of building, the mean absolute error (MAE) was used. A major drawback of MAE for this particular problem is its symmetric nature, i.e., assuming the actual delivery duration is 5 minutes, MAE penalizes a prediction of 4 minutes equally as it would penalize a prediction of 6 minutes. This makes the MAE (and other similar symmetric loss functions) not suitable for evaluating models for the problem since the cost of under-prediction is more than that of over-prediction. This function can updated to account for penalizing too late/early deliveries more than slightly late/early deliveries.

A custom loss function(**custom_mae**) that takes into consideration the preference of over-prediction to under-prediction was developed. After developing and evaluating three models namely: Random Forest, XGBoost, and Simple Linear Regression models, the Random Forest model performs the best, in both cases of employing the traditional MAE as well as the custom cost function. With an MAE of 228 on the validation data-set and 697 on the training data-set, the Random Forest model was selected. This implies that on the training data, there is an absolute deviation of 228 seconds in delivery time on average, whereas the average absolute deviation in delivery time on the validation data is 697 seconds. Further analysis on the validation dataset shows that, for late deliveries, the

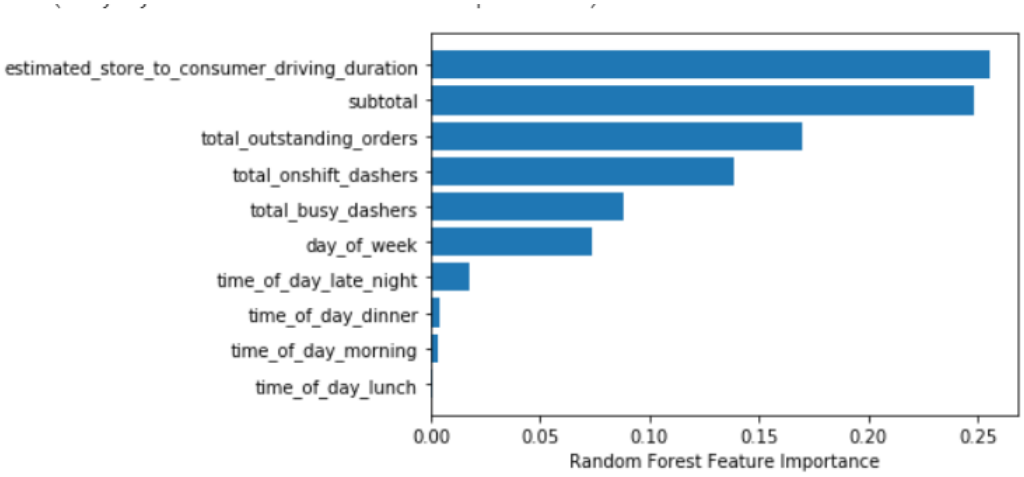

Figure 1: Factors that contribute most to the Model

average delay in seconds is 860 whereas on average, deliveries made before the estimated time is 558 seconds early.

Figure 1 shows the factors that contribute the most to the chosen model above. Feature importance of a model is basically the order of importance of each of the features to the model or the rank of features based on their respective contributions to predicting the response in the model. As expected, distance to the customer from the store plays the most significant role in in determining the delivery duration (expected delivery time). Followed closely is the subtotal of the customer's order. Information on total outstanding orders, and total on-sight dashers also contribute significantly to the model. This is consistent with expectations because if the number of outstanding orders is more than dashers available, it can lead to delays in delivery and on-time deliveries vice versa.

2 New Features

Getting the right features is one of the most crucial parts of any machine learning modelling process. Getting these features are derived from business intuition. Most of the features discussed below are factors which mostly can cost delays in delivery time. Since delays in delivery are the most crucial part of this problem(i.e underestimation of total delivery duration), I classify the the various factors that can cause this into four categories below.

1. Delays from the restaurant
2. Road traffic

3. number of dashers available

4. And also, the inability of the dasher locating the house/apartment

The proposed features which can aid in improving this prediction problem can be classified under at least one of the factors above.

- **Doordash promotion for dashers:** From the knowledge that doordash already has some promotions for dashers mostly during busy periods. This which I believe mostly influence the number of dashers on available to dash. Total on-sight dashers which is one of the main features of importance in the model selected , any factors that affects this can immensely add information to modelling this problem.

- **Categorizing the stores:** Though most delays from stores/restaurants are mostly due to the queue available at the time, categorizing stores/restaurants into the kind of food/order they sell can be informative on getting informed on where delays are more likely to be. That is, grouping fast food joints like MacDonald's/ Burger King into one category whereas restaurants like steakhouse can be grouped into another category.

- **Categorizing location population size:** It is clearly distinctive the traffic situation in small cities and big cities. As difficult as estimating the traffic situation in any city/town offline is, getting some information on the location(the population in the are) will be informative in estimating the traffic situation in any city at anytime. For this reason, adding this as a feature can help improve the model performance

3 Deploying New Models

A/B testing is use to assess the new model performance before fully deploying. This is a carefully organised process in order not to hurt user experience. There are costs and benefits associated with this experimental design approach in assessing model performance on real time data. This is a carefully organised process because the distribution of variables might change over time or some unexpected happenings. And also you want to be able to test your new model without hurting user experience in the process. Outlined below are the steps of A/B testing before deploying a new model.

- Deciding on the metric to use: Deciding on a metric to use is the key part in any experimental design in other to evaluate your system/ business model. The metric for problems of this kind are mostly be the ones used to during the training stage such as RMSE, MSE, MAE. Custom metrics such as dasher efficiency, which is the number of on time deliveries dashers are able to do within the period of time, can also be used in this case. To actually evaluate this using this metric, any bias from individual dashers can be reduced by rating dashers based on how far off there delivery time if off the predicted time.

- Write down hypotheses: Just like any statistical inference problem, we need to set up a hypothesis for the experiment. For this situation, the null hypothesis will be the **'there is no difference between the new model and its predecessor'**. Whereas the alternate hypothesis is **'the new model is better than the old model'**.

- After this, one has to determine the sample size(number of users) for the experiment.

- The experiment is then run for a period of time, where the old model is used on a portion of the sample and the one model used on the rest (let's say from the sample size selected 70% uses the old model whiles the rest 30% uses the new model), this is to ensure that most users' experience is not hurt. The users in experiment is dashers(doordash driver). Running the experiment for a period of two weeks

- Using the metric determined above, a statistical test such as T-test (best use two-tailed test to check direction of effect) can be performed after which the null hypothesis is rejected or not.

- A decision is made to deploy or not deployed based on the null hypotheses. The new model is not deployed if the null hypothesis is not rejected. If the null hypothesis is rejected, the decision to deploy is based on how much more the new model is performing than the old one. The difference has to be significant enough to be worth the resources needed to deploy.