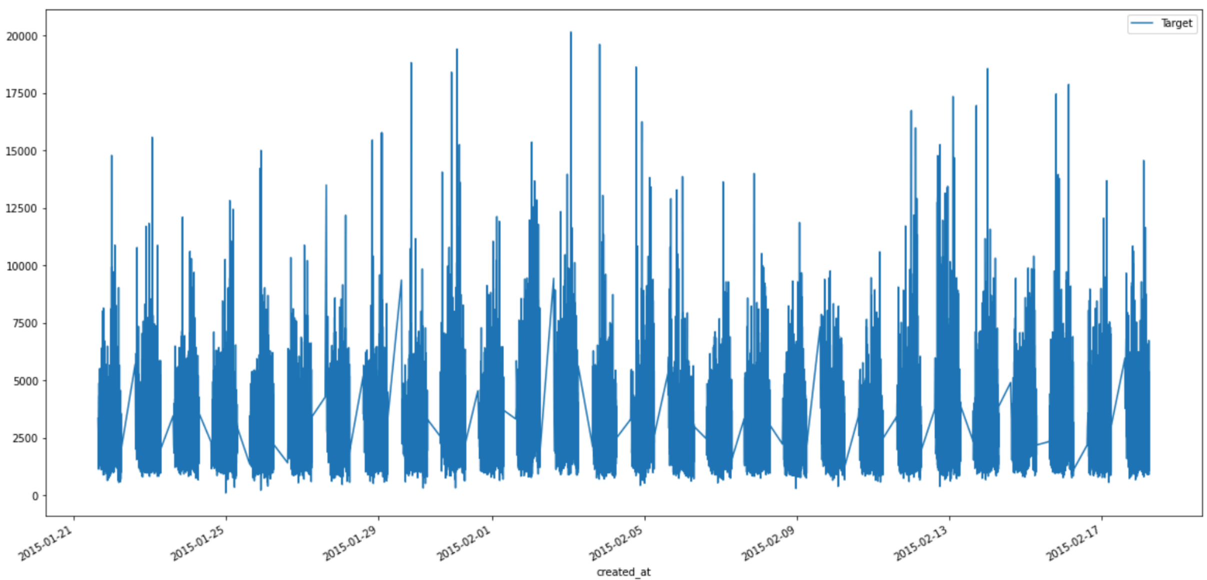
Nicholas Reimold

Can we Predict the DoorDash Delivery Time?

As many of us have stayed home progressively more, food delivery services have been used at a rapidly increasing pace. Within these apps, there is a lot of competition, and the slightest differences can make all the difference in customer experience. DoorDash, like many others, gives customers an estimate of when their order will be delivered. This prediction can have a wide range of effects depending on the accuracy. Underpredicting the order time is twice as harmful as over predicting and bigger prediction errors are much worse than smaller errors. One can imagine that a customer planning for an order to arrive at 8:00pm will be much more pleasantly surprised that it arrived at 7:50pm than at 8:10pm. For this exercise, I attempted to predict delivery time given a set of order features.

Below is a visualization of how the target(seconds until delivery from order creation) varied by day. There quite a few outliers on each day, but the majority of observations top out at around 1.5hours(5400s).

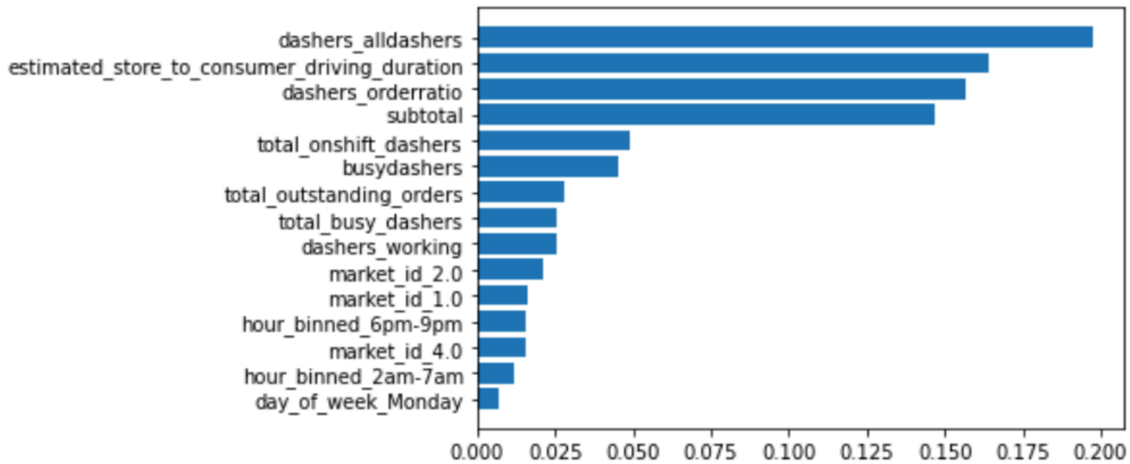


At a high level here are some of the features I considered and some intuition behind them:

- Order subtotal: the higher an order subtotal, the bigger a potential profit a dasher could make, making it attractive
- Available dashers: higher availability makes it more likely order will be picked up quickly
- Outstanding orders: A higher number of orders makes it likely a lot of dashers will be working given the large opportunity for orders
- Estimated driving duration: This is an output from another model that makes up a chunk of the order time

These few features illustrate that there is promise in this dataset, as these are important factors in how quickly an order will be delivered. From this initial set, I created some additional variables and then built a model predicting the target time. My approach with model building was to build a solid baseline model and then optimize it to minimize order underprediciton. Since this is twice as costly as overprediction, it is important for customer experience.

From my final model, here are the most important features:



Some important variables in the model include available dashers ratio, the dashers to order ratio, estimated driving duration, and subtotal. These are mostly features we picked out from the initial dataset that looked promising. The dashers_alldashers ratio is a representation of the percentage of available dashers, which is a good description of how available workers are in the area. There are also some various market intricacies that are important to the model, such as market id and order time, which all combine to make a more stable prediction.

Overall, this model produced a Root Mean Squared Error of 811.5 on a test set, which is decent for this dataset. This represents an error of about 13.5 minutes, which is not perfect, but it is a good start. I did make some tradeoffs in order to make the model underpredict less than it overpredicts, which it only does at a 33% rate now. This resulted in an average prediction error increase of about 30 seconds, but I'm betting that this will be beneficial for customer satisfaction. Some additional features that could be helpful in building a more accurate model are:

- Traffic data: this could give a better idea of how long an order will take to be delivered
- Restaurant order time: knowing the metrics of how fast a restaurant usually prepares an order will give a good idea of how long to budget on this side of the equation
- Number of items/item prices: this would enable to tune the order time, given that a high number of items is likely to take longer
- Customer information: how many times has a customer ordered and had the order be late or early- while this may not help with the prediction time, it could help in terms of customer satisfaction because we don't want to consistently underdeliver to the same customer

Now that we have our model built, we may want to replace a current model in production. To evaluate this, I would consider many of the error metrics(how big/small) and whether we underpredict or overpredict. There are tradeoffs with both of these and given a similar error rate, I would want the overprediction rate to be better in my model, or vice versa. I would also want to do a more thorough validation of the evaluation data. I don't know how representative the test data is of orders in general, given that it is only over a week. I'd want to evaluate both my model and the current model on a more robust dataset. Given that all these checks pass, finally I would evaluate the model for business impact. Although it is accurate on average, I want to make sure it holds true for specific cohorts. i.e. if the model is accurate for market 1 but not market 2, I may need to reexamine. It looks fine for the current dataset, but using more data and a wider timeframe could vary it. This part is all about ensuring a consistent user experience, because at the end of the day, that is the goal of this model.

In conclusion, this was a very interesting problem to explore. While there are certainly other variables to explore, many of the current metrics, such as available dashers and estimated driving time, proved very predictive. I believe this model strikes a good balance between error rate optimizing and customer satisfaction, making it a good model to start with and build out as we get more features.