Delivery Duration Prediction

Download Datasets

This data project has been used as a take-home assignment in the recruitment process for the data science positions at DoorDash.

Assignment

When a consumer places an order on DoorDash, we show the expected time of delivery. It is very important for DoorDash to get this right, as it has a big impact on consumer experience. In this exercise, you will build a model to predict the estimated time taken for a delivery.

Concretely, for a given delivery you must predict the total delivery duration seconds , i.e., the time taken from

- Start: the time consumer submits the order (`` `created_at` ``) to
- End: when the order will be delivered to the consumer (`` `actual_delivery_time` ``)

Data Description

The attached file `` `historical_data.csv` `` contains a subset of deliveries received at DoorDash in early 2015 in a subset of the cities. Each row in this file corresponds to one unique delivery. We have added noise to the dataset to obfuscate certain business details. Each column corresponds to a feature as explained below. Note all money (dollar) values given in the data are in cents and all time duration values given are in seconds

The target value to predict here is the total seconds value between `` `created_at` `` and `` `actual_delivery_time` ``.

Columns in historical_data.csv

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)

Practicalities

Build a model to predict the total delivery duration seconds (as defined above). Feel free to generate additional features from the given data to improve model performance. Explain:

- model(s) used,
- how you evaluated your model performance on the historical data,
- any data processing you performed on the data,
- feature engineering choices you made,
- other information you would like to share your modeling approach.

We expect the project to take 3-5 hours in total, but feel free to spend as much time as you like on it. Feel free to use any open source packages for the task.

Resources

- Video walkthrough (data preparation)
- Video walkthrough (collinearity and removing redundancies)
- Video walkthrough (handling multicollinearity and feature selection)