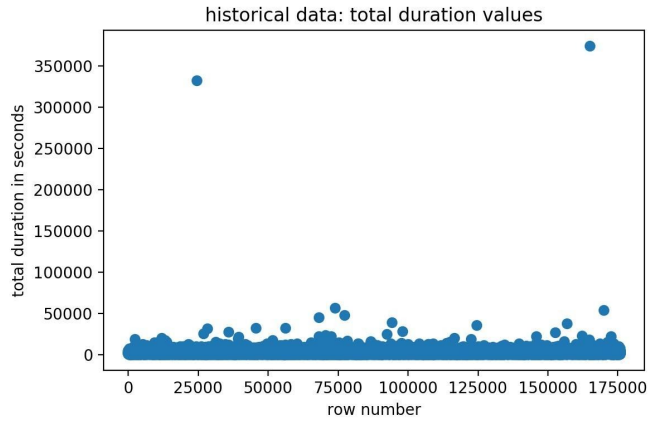
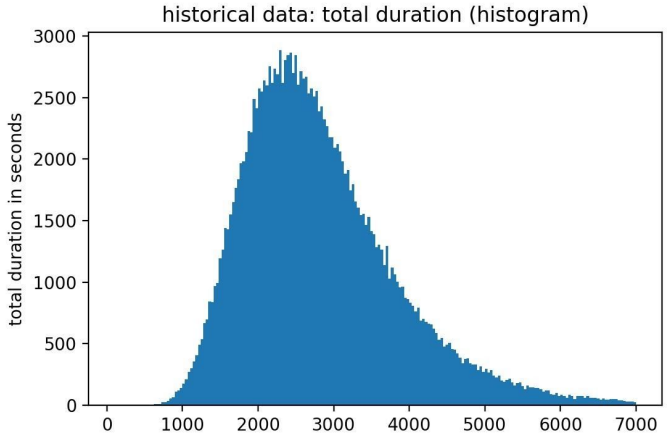
Accurately predicting the time required for delivery helps to ensure a positive customer experience. These times are highly variable and can be influenced by a range of factors. One way to estimate delivery times is to build a multi-linear regression model which incorporates information about multiple features. The following summarizes the output from a multilinear regression model trained on selected features from the historical data set provided. We are aiming to predict the dependent variable (the total delivery duration time in seconds between the order creation and arrival to the customer) based on a set of historical features related to the time requirements, store attributes, order characteristics, market features, and complementary predictive model outputs.

Multilinear regression is a powerful tool for finding key features driving a response and assigning weights to them so they can be combined to make predictions using new input data. The model itself has a few assumptions, the first and biggest being that there is a linear relationship between outcome and the independent variables or features included in the model. It also requires that residuals are normally distributed. In general these will not be met if we have extreme outliers. While there are some ways to account for outliers in a robust model when it's necessary to be able to predict extreme values, here they are discarded in favor of a better model fit and greater predictive power.

Below we see the total delivery duration and can visually identify extreme outliers with delivery times on the order of days.



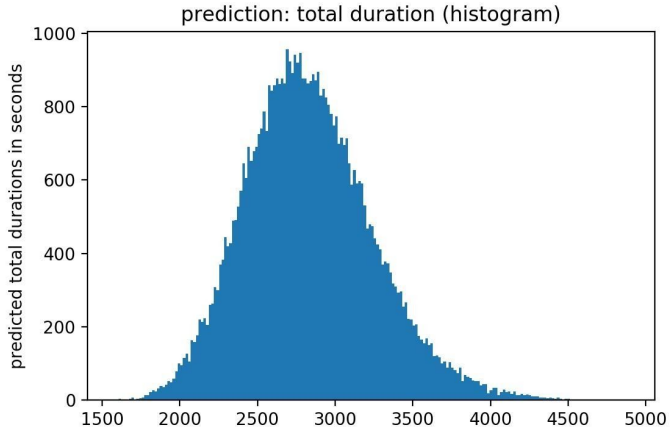
These are considered outliers and removed from the training data set. The distribution of the remaining data (below) resembles a Poisson distribution, which is not atypical for a distribution of random 'arrival times'.



The feature data in the training (historical) dataset is further cleansed by removing rows with missing values. Overall, roughly 88% (or around 170,000 rows) of the data in the training set is preserved.

Since the model includes many types of variables (continuous, categorical, rank order, timestamp) and over varying scales (100s to 1000s for cost in cents, 10s to 100s for onshift drivers), it is important to first transform and scale the data. This allows each feature an equal opportunity for inclusion in the final model instead of favoring the features with the largest values. Once the variables are transformed and scaled, each is added individually to the regression model. If the model fit improves, the variable remains within the model and if not, it is removed.

The most predictive power resulted from a model including the independent variables estimated_order_place_duration, estimated_store_to_consumer_driving_duration, total_available_drivers, total_outstanding_orders, market_id. The model has a mean absolute error of 705 seconds or about 12 minutes, which is about 25% of the average delivery time (~45 minutes). The adjusted R-squared value is 0.17. The distribution of the predicted duration times is shown in the figure below. The predicted durations are less skewed than the historical durations.



Some features on their own can increase the delivery time significantly. By looking at the Spearman correlation coefficient between each numeric variable and the delivery time, we have an indication of the strength of each bivariate relationship. Similarly, the average delivery time associated with a particular label in a categorical variable is especially high, there is likely a strong relationship and vice versa, if all labels within a category have a similar average waiting, that variable probably does not contribute much predictive power. This provides an additional means of selecting features and given additional time, could be included in additional model iterations to improve the predictive power. In the table below, the features with the five highest correlation coefficients (absolute value) are listed.

feature	**Correlation coefficient with total duration**
`estimated_store_to_consumer_driving_duration`	`0.27`
`subtotal`	`0.24`
`total_outstanding_orders`	`0.18`
`num_distinct_items`	`0.17`
`max_item_price`	`0.16`

Once implemented, a model's performance (and a comparison with a current production model) can be measured through the adjusted R-squared coefficient. This represents the proportion of variance explained by the model, or more plainly, how well the included variables impact changes in the outcome. However, the r-squared value will increase with each added variable, even when the new variable does not boost the model's predictive power. For that reason an adjusted R-squared, which penalizes models with too many included variables (overfitted models) is a better indicator of model performance.

The model's performance can also be tested through k-fold cross validation. This involves splitting the input data into training and test sets and building multiple models on subsets of the data. The average score indicates the generalized performance of the model. The average performance score resulting from cross validation and the adjusted r-square values can be compared to similar metrics from any additional models which are currently in use.

With further time and resource, the model could be improved in several ways, for instance by including variables which represent the way these features interact with each other. For example, the number of distinct items in an order may not be a huge influencer of delivery time, but the combination of the number of items with the “created at” time may have a more substantial impact, since this could be an important factor at particularly busy times of day.

The model could also be more sensitive to the correlations between variables. In some instances the given features may not represent independent observations, such as the number of orders and the number of available drivers, which likely are both related to expected demand and consequently correlated with each other.

Separate models using subsets of the data, for instance only including data specific to a given location or time of day may also improve the predictive power. Additional the model construction could have followed a ‘leave-one-out’ method where one variable is removed with each iteration (rather than adding one at each step) and additional model types could be tested, such as a neural net.

The magnitude of a delivery error matters. A complementary model could be built on cases where there was a large discrepancy between the predicted and actual delivery time. The features which drive this variability between the expected and actual outcomes could then be used with a higher weight in the full predictive model, since they are more likely to contribute to a poor customer experience.

In addition to model considerations, data availability could be used to improve the model by adding additional variables which exert a stronger influence on the delivery time. We would expect the average time it takes for a restaurant to prepare a given number of food items to be extremely sensitive to the forecasted total delivery duration. Demand features would likely also be impactful, such as the weather or major sports events which could quickly drive up the number of expected orders.
Features which influence transit time may be helpful, such as heavy traffic, extreme weather, or an address which requires climbing 5 flights of stairs. The transportation method for the driver would also be a feature of interest since a driver on a bike might be less affected by traffic and drop-off location data than a driver in a car who might be stuck finding parking at the drop-off location.