Predicting Delivery Time

Demet Tunali

June 17, 2021

Contents

1 Overview

1.1 Executive Summary

In this study, we examine the drivers of the total delivery time of food requested through DoorDash and explore predictive models via two machine learning algorithms. Delivery time includes store to consumer driving duration and order place duration. The data covers 197.3 thousands distinct orders placed across 6 markets and 7 order protocols. The analysis is conducted using the Kaggle platform. Through the application of a gradient boosting model, we achieve 97.09% prediction accuracy. We also note that the most days of the week when the order is placed as well as the share of distinct items are not relevant for predicting the order delivery time.

1.2 Purpose of this Document

The purpose of this document is to describe the data and the steps used in predicting the food delivery time for orders placed through DoorDash. The first section describes the data processing steps. Particularly, observations about the numerical and categorical variables as well as data cleaning steps are presented. We also investigate univariate associations with the target variable. We then examine two types of ensemble models, namely the Random Forest and Gradient Boosting algorithms. The final section summarizes the conclusions.

2 Data Preparation

2.1 Data Enrichment

The dataset consists of main information associated with a typical DoorDash order. We use this information to enhance the dataset with derived features. We believe that, as far as the timing of the order is concerned, the day of the week and the calendar month might be a relevant attribute. Furthermore, we expect the ratios rather than plain levels or volumes would be more informative. Hence, we define percent of available dasher, dasher to order ratio, and share of distinct items. As far as the pricing data is concerned -such as the subtotal and the minimum/maximum item price, we argue that business input would be rather valuable to build some sensible hypothesis.

2.2 Data Assessment

When we plot the distribution of the numerical features we observe several skewed variables, such as percent of available dasher, total outstanding orders. In the meantime, due to a natural lower bound, we don't attempt to normalize these variables. We also observe several variables with accumulation around lower values along with very high records; e.g. total items, delivery time. A subsequent boxplot justify that this result is indicative of outliers. The count plot of categorical variables show that the subcategories are mostly well populated. In the meantime, created month show that the data is not representative in this respect.

2.3 Data Cleansing

The raw data for predicting delivery time consists of a sample of 197.4 thousand observations and 16 features. 13 of these features are coded as numeric while some of them are in essence time related or categorical variables.

Initial data processing steps focused on addressing data anomalies and missing values. As per the description of the dataset, none of the 16 numerical features are expected to be less than or equal to zero [1]. Thus, these invalid values are set to missing. We then proceeded to remove duplicate records, where store id and order creation time are used as key identifiers. This step delivered 197.3 thousands of unique orders.

When we tabulated the missing values, we noticed that the dataset is mostly well populated. Hence no variable is excluded due this issue. We started addressing the remaining missing data by leveraging some business intuition.[2] As such, we assumed a given store would be associated with one cuisine type. Therefore, each store is represented by the category that first appeared in the dataset.Similarly, we attempted to leverage a potential association between market id and cuisine type to impute missing market id information. This approach, however, failed due to system insufficiency.The run on the Kaggle platform outputted error before finalizing the execution of the submitted commands. Other remaining values in categorical and numerical variables are filled in using common imputation methods.

In light of the boxplots of the numerical variables, we remove outliers using the 1.5 times inter quartile range rule. Consequently, we trim 25% of the observations.Removing outliers provided a more sensible distribution of

[1] We assumed that id variables are not supposed to be zero unless there is a business support

[2] Although the gradient boosting algorithm is known to handle missing data, we chose to feed the same data input to the explored models.

the numerical values, especially for the target variable. When we look at the correlations among variables, we noticed over 90% correlation among three variables. Accordingly, we remove two features -total on shift dashers and total busy dashers.

2.4 Univariate Impact Assessment

When we analyze the correlations with the target variable, we don't observe at or more than 30% correlation in absolute value. This linear association is particularly low for minimum price item and share of distinct items. Scatterplots emphasize the lack of strong linear relationships as well. The display of the data points, looking all over the place, makes it hard to postulate any relationship that might be established with conventional regression models. Boxplots of categorical variables suggest that the distribution usually seems to vary across sub-classes. This is not, however, valid for the created month information.

3 Model Design and Performance

For predicting delivery time, we investigate two ensemble methods, namely random forest and gradient boosting algorithms. We assess model performance via accuracy rate defined as one minus mean absolute percent error.

In light of the steps described above, we dropped created month, total on-shift dashers and total busy dashers columns before we initiate any machine learning algorithm. We also encode categorical variables. For variables with many categories we apply label encoding method. In contrast, we leverage one-hot encoding for the created day variable. Finally, we define the target variable in logarithmic scale.

We first apply a simple random forest regressor with constant paramaters for maximum depth, number of estimators and minimum samples leaf. This algorithm reaches 97.03% accuracy. The subsequent feature importance plot points out that most days of the week and the share of distinct items were not widely leveraged in defining new splits. As such, removing these variables kept the accuracy rate intact. We also introduced randomization via hyperparameter tuning and addressed model variance/bias trade-off via 3-fold cross validation. These enhancement, however, did not boost accuracy.The numerical elements of the dataset comes in various scales. While some variables represent time in seconds others are dollar measurements. So we explored the 0-1 transformation of all numerical variables, including encoded ones. We demonstrated that no benefit would be obtained from such transformation.

We also trained a gradient boosting algorithm by defining a limited number of parameters. The accuracy slightly deteriorated. The best performance is obtained when the model parameters are tuned, to achieve 97.09% accuracy.

4 Conclusion

Through the application of a gradient boosting model, we achieve 97.09% prediction accuracy. We also note that most days of the week when an order is placed as well as the share of distinct items are not relevant for predicting the order delivery time.Unsurprisingly, estimated store to consumer driving duration has been a key element in establishing a strong predictive model.