Alexandra Whiteford
February 2023

Predicting Food Delivery Times

Introduction

Why?

Food delivery is a type of courier service in which a consumer receives food from a restaurant, shop, or an independent food delivery business. Typically, orders are placed electronically through a mobile app or a restaurant's website. The deliveries often are composed of meals, desserts, drinks, snacks, or buffet orders, and are shipped in boxes or bags to be handled with care by the delivery driver until it reaches the customer. It is common for delivery drivers to make their deliveries using cars, but in larger cities where the restaurants and homes are not far from each other, the deliveries are often completed by bicycles or motorized scooters.

As food delivery services remain an integral part of city life, the wait times for these orders can often seem unreliable or incorrect. Whether the delivery driver is arriving too late or too early, or there is no communication, or the food has arrived cold, it can be difficult to predict just how long the order will take getting from the chef's hands to a customer's doorstep. The main goal of this project is to predict the most accurate travel time between locations to ensure the customer is left satisfied.

Audience

This report is directed towards data science officials or driver relations professionals at online food ordering companies, such as GrubHub or DoorDash, so that more reliable estimations about delivery wait times can be made and lead to more satisfied customers. For the purpose of this project, I have built a prediction model based on the input of a sample of food delivery data and outputting the most feasible pick-up and delivery times to be shared with the driver and the customer.

Goal

The ideal estimation for food delivery windows are within 5 minutes or less of the predicted delivery time. This must be done so that the estimated drop-off times are honest and reflect accurate estimations for both the customers and the drivers. There are several factors that go into predicting how long it will take a delivery driver to get from point A to point B, and we will explore these variables in the dataset. These variables will determine if the delivery time will be longer or shorter based on, for example, the traffic.

The Data

Source

This dataset was downloaded from Kaggle, and the data was collected as a sample from past deliveries, including the time it took from start to finish and the factors that directly impacted that time. The dataset is titled “Food Delivery Dataset”, and includes the following columns:

- ID: The unique identifier of the individual delivery orders.
- Delivery_person_ID: The unique identifier of the person completing the delivery route.
- Delivery_person_Age: Age of the person completing the delivery route.
- Delivery_person_Ratings: Rating between 1 and 5 of the person completing the delivery route.
- Restaurant_latitude: North-South coordinate of the restaurant location as the delivery starting point.
- Restaurant_longitude: East-West coordinate of the restaurant location as the delivery starting point.
- Delivery_location_latitude: North-South coordinate of the customer location as the delivery end point.
- Delivery_location_longitude: East-West coordinate of the customer location as the delivery end point.
- Order_Date: The date that the order was placed on in dd-mm-yyyy format.
- Time_Orderd: The time of day that the order was placed.
- Time_Order_Picked: The time of day that the order was picked up from the restaurant supplier by the delivery driver.
- Weatherconditions: The state of the weather at the time the order was placed.
- Road_traffic_density: How dense the traffic conditions were during the delivery period.
- Vehicle_condition: The condition of the delivery driver’s vehicle.
- Type_of_order: The type of food that was ordered for delivery (Snack, Drink, Meal, Buffet)
- Multiple_deliveries: The number of additional deliveries the delivery person is making during the trip.
- Festival: Whether or not the festival coincided with a festival in the area (Yes/No)
- City: What kind of living environment the delivery was made to (City, Metropolitan, Semi-Urban)

Methods

It is my understanding that the most common way to estimate the length of time a customer is expected to wait for their food order to be completed and delivered depends on a simple estimation of how long it will take the restaurant to prepare the order, how long it will take for the delivery person to pick up the prepared order from the restaurant, and finally the amount of time it typically takes to drive from that location to the delivery address. While this seems straightforward, there are many other components that can affect the amount of time that it will

take for the driver to get the order to the customer's door, such as several of the dataset columns listed above. While exploring and beginning to clean the data, some of these factors stood out to me as strong variables for delivery time correlation, while others seemed insignificant regarding their impact on the amount of time taken to deliver the order.

The main variable to be focused on in this dataset is the Time_taken(min) column, which will then be predicted using training and testing data that is to be created. From there, food delivery companies can then use the algorithm to predict how long it will take to complete the order and delivery. The performance metrics for delivery times can be categorized as:

- Early/On-Time delivery: > 5 minutes early
- Exactly On-Time: +/- 0 minutes
- Basically On-Time: < 5 minutes late
- A bit late: < 10 minutes late
- Late: 10+ minutes

After a customer is waiting for an order that was predicted to be delivered more than 10 minutes prior, it doesn't matter how much longer the order takes, it is categorically late and the customer will be unsatisfied. This dissatisfaction could lead to the loss of customers, poor reviews, and frustrated delivery drivers who are at risk of receiving poor tips. When this lateness occurs, it is very likely that the time prediction algorithm did not take into account several factors such as traffic density, weather, or vehicle condition.

The main components of a delivery's timing, as mentioned prior, include the time it takes for the supplier to prepare the order (also known as prep time), the time it takes for the delivery person to pick up the order (also known as pick-up time), and the time it takes for the delivery person to travel with the order to the delivery location (also known as delivery time). However, there are hyperparameters that also must be taken into account. While doing research on the topic, I discovered that several food delivery services simply add an additional 5-10 minutes to the estimated delivery time so that if there is a delay in the delivery it will still appear on-time, or additionally if there are no delays in the trip then the delivery will be considered early and leave the consumer pleased with the swift process. While this practice may seem manageable in theory, it is not always guaranteed to leave customers happy due to the wavering uncertainty that each delivery will be made in a timely and efficient manner. The goal of this project is to use the data provided to predict the most accurate delivery times in order to increase positive reviews and satisfied customers.

Data Wrangling

Below is an overview of some of the main issues that I encountered during the data cleaning process:

1. Problem: Getting rid of null or missing values
    a. Solution: Using a for loop on the columns with loc[ ] and np.nan

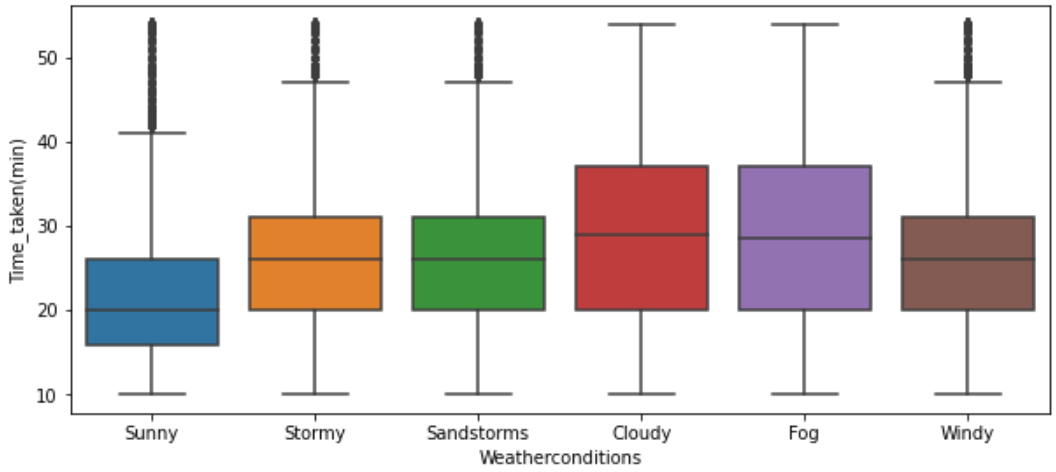
b. Additional solution: Use dropna( ) with the subset and fillna( ) to replace any null values using ffill method

2. Problem: Convert Time_taken(min) from an object to a float value
    a. Solution: Using a for loop to change the type from str to float using astype( ) so that it may be interpreted as a numeric value for our predictions
3. Problem: Unknown distance, only coordinates given
    a. Solution: Import geopy packages and input the coordinates for the starting points and the delivery destinations to output the distance in kilometers
        i. Major Issue with this: A portion of the distance results appear to be in the 2000-7000 km range regarding the distance traveled for deliveries. This does not make sense as this is much too far of a distance for a driver to travel. Likely the data is either incorrect or my distance formula is not properly applied, so it is not entirely reliable. Therefore, this distance column will not be used for final analyses but rather for initializing and exploring the data.

Data Visualization

Once the dataset was cleaned, and all the necessary packages had been imported, I used seaborn's boxplot on 5 different variables against Time_taken(min) in order to get a better visual representation of how these variables impact the length of delivery time.

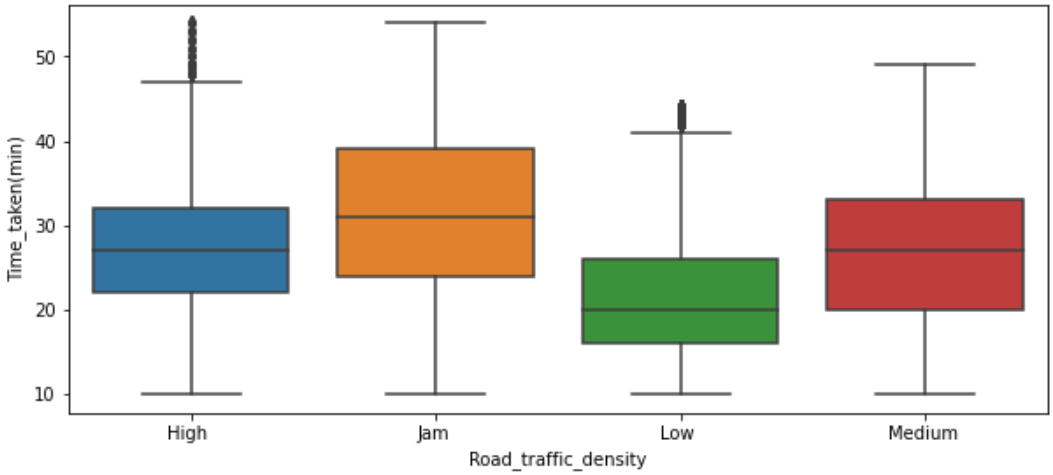
Weather conditions



The boxplots here verify the assumption that on clear, sunny days the route between the restaurant and the delivery location will be the quickest, taking 20 minutes on average. The estimates on stormy and windy days are in the middle range, taking between 20-30 minutes. The weather that offers the widest range in time are cloudy and foggy days, taking between 20-36 minutes to complete the delivery, with an average of 30 minutes. Based on this visualization, it is

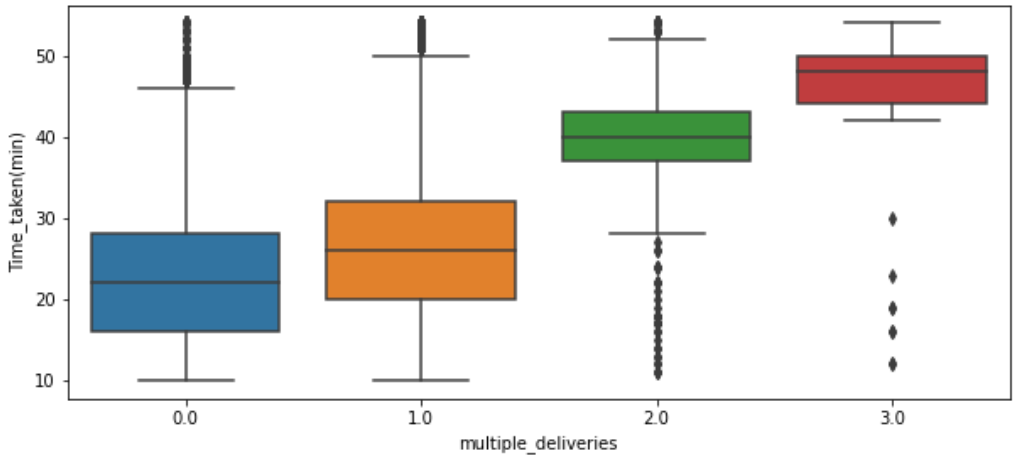
clear that weather has an impact on how long it will take the delivery person to reach the destination from the supplier.

Road Traffic Density



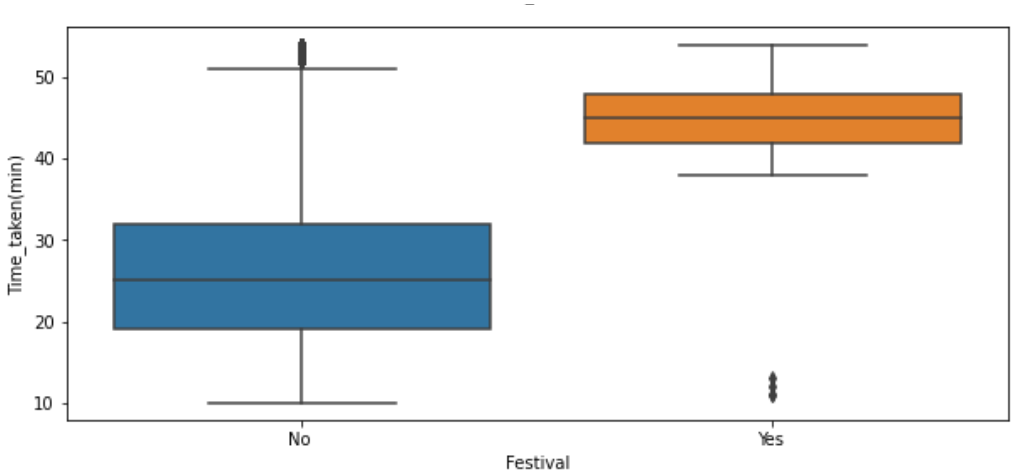
The boxplots here represent a very clear-cut visualization for the impact of road traffic density on delivery time. Where there is low traffic density, the route to the delivery location is clear and direct, taking about 15-25 minutes on average. When the traffic begins to pick up to medium density, the route takes a bit longer, ranging from 20-32 minutes. When the traffic is quite high in density, the estimated time could typically take between 22-32 minutes, roughly the same as in medium-density traffic. When the roads are really jammed and the delivery driver is stuck among the slow-moving traffic, the delivery often takes between 25-40 minutes to complete. This suggests a strong correlation between road traffic density and time taken to reach the customer from the supplier.

Multiple Deliveries



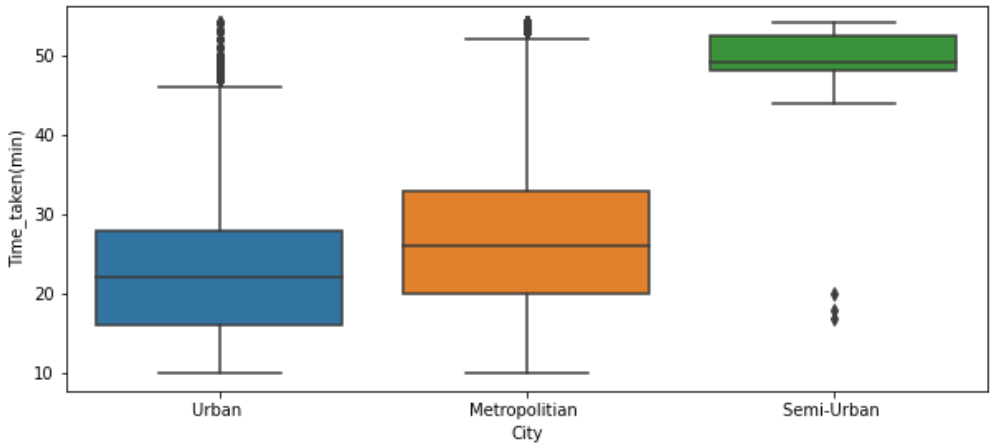
The amount of additional deliveries taken during a delivery route appears to have a very significant impact on how long it takes to complete an order. This was slightly predictable, considering that the delivery driver has to stop at one initial delivery location to drop off one order before moving on to drop off the next order, leaving the second order to take more time than the first. If a driver takes as many as 3 additional deliveries, totalling 4 in one journey, it will leave the final delivery to take the longest. One single delivery takes between 16-28 minutes on average, an additional delivery takes between 20-32 minutes on average, two additional deliveries take between 36-42 minutes on average with many outliers, and three additional deliveries take between 45-50 minutes on average. While this data visualization proves significant, it is also worth noting the high amount of outliers in each boxplot.

Festival



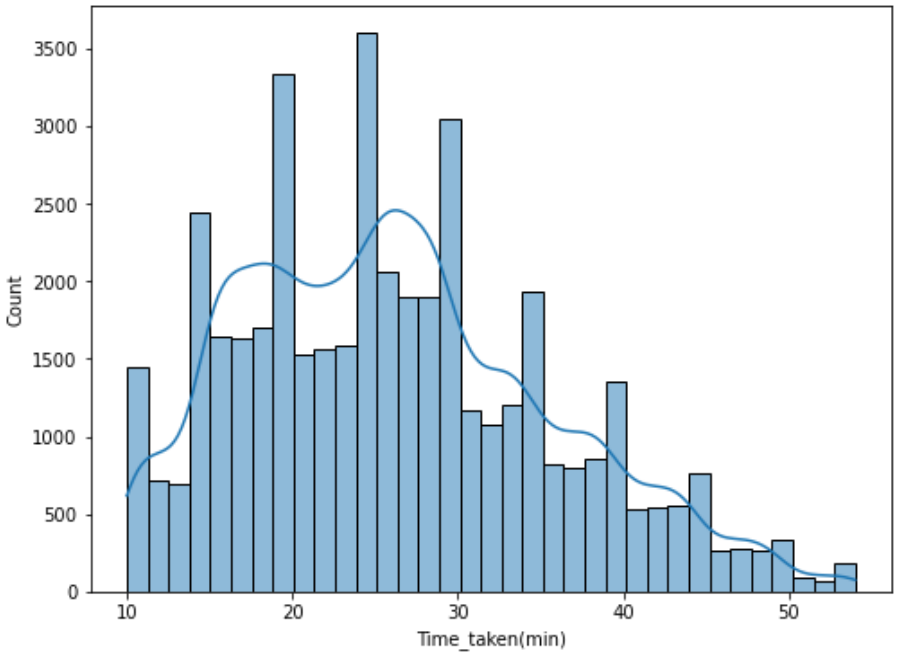
When there is a festival in the area that the delivery route must travel through to reach the delivery location, there will likely be heavy traffic or road closures to account for the large amount of space that the festival needs for activities. Therefore, when there is a festival in the area during the time that the delivery is being made, it will often take the driver much longer to reach the delivery location. The average time for delivery with no festival is 25 minutes, while the average time for delivery with a festival in town is 45 minutes. While only about 2% of delivery drivers encounter festivals during their routes (896 out of 45,365), this is significant enough of an impact on time that it must be noted.

City Type



The boxplots here display the type of urban living environment that the order is being delivered to. Urban describes a typical downtown city area, most typically in smaller cities, while Metropolitan describes a larger metropolis of a city, which often has more people, homes, and restaurants, and therefore may take a bit more time to deliver to some addresses. The times for urban and metropolitan are similar, however with more square footage to cover, the metropolitan deliveries take a little bit longer on average, about 5 minutes more. The Semi-Urban living environment, on the other hand, takes much longer. These living environments are typically much more spread out and further away as they can stretch to the outskirts of the city and beyond that, similar to suburban homes. Due to the further distance that must be traveled by the delivery driver, these deliveries take much longer, averaging 50 minutes.

Time taken



Above is a histogram of the average times taken for deliveries to be completed, with a majority of the deliveries taking between 20-30 minutes. The highest point of the average is at about 27 minutes, as can be seen at the peak of the kernel density estimation line. This plot has 35 bins, and what I find interesting about this plot is that it appears a large portion of the data has been recorded in multiples of 5. My assumption here is that the data was being entered as delivery time rounded to the nearest multiples of 5 quite often and recorded as such. Based on this assumption, the data is not as precise as it could be, which is not ideal but still manageable as it still provides us with estimations within a 5 minute range as many delivery companies often include as a buffer time frame. Therefore, this is not a concern to us, but should still be taken note of when interpreting results later on.

Modeling

Before any modeling steps can be done, a train/test split must occur. We must first define our X and y values so that the data may be split into four pieces: X_train, y_train, X_test, and y_test. X is defined as the independent variable, and contains all of the columns in the data excluding Time_taken(min), while y is defined as the dependent variable, which is solely the Time_taken(min) column. The X value is the input variables, and the y value is the output, also known as the value we are trying to predict, which is time.

Additionally, before the data is split into four parts, dummy variables must be created so that the categorical variables may be converted into integer values for modeling evaluation. We then call the pandas function get_dummies on X and y, and we are ready to split the data. The train size is set to 0.75:

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.75, random_state=123)
```

Linear Regression

The Linear Regression model is one of the simplest regression methods, searching for and calculating the relationships among the variables in the dataset in order to create predictions. The main values that I will be interpreting the results of are the Root Mean Squared Error (RMSE) and the R-Squared (R2) values. See the model metrics comparison table below for all modeling results. The RMSE is approximately 0.1454 which is very low; this tells us that the residuals and variance were very similar to what was anticipated. This confirms that our predicted values align closely with the observed values, however the R2 value is approximately 0.0214 which is lower than is ideal. This tells us that while the data may be following a significant trendline, the data is also noisy and has high variability. This indicates that although the predicted values fall further from the regression line in this case, the predicted values are still able to give us information about the trendline.

K-Nearest Neighbors

The K-Nearest Neighbors works to classify the proximity of a predicted value to others so that they may be grouped or classified together. The RMSE value once again tells us the difference between the observed values and the predicted values, and that value is about 0.1516 which is very low and thus very good for our correlation between our variables and the predicted values. The R2 value is -0.0502 which is not good at all, as a negative R2 value means that the model's predictions are bad and less accurate than with our linear regression model. Using the K-Neighbors Classifier was not successful in this case, even though the RMSE value was low as we were hoping.

Random Forest Regressor

The Random Forest Regressor uses a number of decision tree splits on several sub-samples in the data to help us determine how to improve the accuracy of the predictive data through averaging in order to control over-fitting. Once again, we have gotten a small, negative R2 value of -0.0463, which tells us that this model poorly fits the data. However, with a small RMSE value of 0.1503, the data still shows a strong correlation between our observed variables and our predicted values. Using the Random Forest Regressor was not an ideal model for this dataset, therefore I will try one more separate model for this data set.

Extreme Gradient Boosting Regressor

The XGB Regressor is a modeling regressor that provides predictive accuracy by minimizing the prediction error overall. It is a powerful technique that boosts weak trees or groups of the data in order to form a stronger tree, providing a lower variance. Here, it provided a RMSE value of 0.149, which is fairly low, once again, and tells us that the data has strong correlation between the predicted values and the actual observed values. The R2 value is the lowest one yet, at -0.069, telling us that while the model may correlate well with the observed values, the model doesn't fit the data all that well. The XBG Regressor is not the best model in this case due to the negative R2 value, but the RMSE is promising.

	RMSE	**R2**
LinearRegression	0.14537212334877098	0.021389304824339917
KNNeighbors	0.15159863732138348	-0.05022433040104393
RandomForestRegressor	0.15028156528536282	-0.04635969968657615
XGBRegressor	0.14904654	-0.06903199681413394

Overall, the most confident modeling prediction is done with the Linear Regression model. The use of this model for predicting the time taken for food deliveries provides confidence in the estimated times produced by the coding and algorithms. By using linear regression modeling to get our positive correlations between variables in the dataset, then, ideally, food delivery companies will be able to use these algorithms to predict their own delivery times more accurately.

Constraints

There are a handful of constraints when it comes to making a food delivery within the predicted amount of time, especially when ordering in a large city. One major constraint that many algorithms do not account for is accessibility to parking. For example, there are often few parking options outside of restaurants and apartment buildings, making it more difficult for the driver to legally and safely park in order to remain on-time. Apartment buildings often require the customer to let the delivery person inside for access to the correct address' doorstep, which can take several minutes if the customer is not prepared for the order's arrival. This is another important reason for why these delivery times must be as exact as possible; if the customer expects it to be delivered at a different time than it arrives, they may not be prepared to let the driver inside the building in a timely manner. Other times, many buildings have security issues that take several minutes to address. Additionally, customers can often input complicated delivery instructions or have a hard-to-interpret address. In order to avoid this issue, the address data must be entered in matching formats (apartment number, floor, building, zip code, etc).

Another constraint is the restaurant or supplier not being ready with the order at the time that was anticipated. This leaves delivery drivers waiting longer for the order to be prepared and therefore arriving later than expected to the delivery address, and possibly leaving an unsatisfied customer. One recommendation to aid this problem would be to have the driver contact the customer to inform them of the delay due to the restaurant, so that way the customer is made aware of both the delay in delivery and that the delivery driver is not at fault. The goal is to keep the customer satisfied with the delivery process from start to finish, ensuring that the driver gets a good rating and a good tip.

Recommendations for Businesses

There are several changes that management of food delivery businesses such as GrubHub and DoorDash can make in order to improve their customer satisfaction. In order to more accurately predict the delivery time, the company must be sure to observe many of the factors and variables mentioned above that contribute to a potential delay such as weather, traffic density, and the amount of deliveries. Particularly in major metropolitan cities where there is often lots of complex housing, it is important to verify the delivery address and, when possible, have the customer note the parking and security when placing the order.

Another recommendation is to limit the amount of additional deliveries the drivers can make in one trip. Ideally, the driver will only accept up to one additional delivery (total of 2) for time efficiency and for freshness in order to keep all customers satisfied. If there are too many orders to fulfill and only a limited number of drivers, causing multiple additional deliveries to be accepted, then the times must be accurately adjusted to account for this, as well as an investment in temperature-containing delivery bags. This way, even if the delivery time is longer than the customer was hoping for, the provided delivery time will still be honest and accurate.

Summary

Customer satisfaction is a key ingredient in keeping a business profitable, reputable, and trustworthy. By providing the customer with the most accurate delivery time estimation possible, the company is ensuring the customer that they will have their food or drink order delivered to their doorstep and ready to be enjoyed at a specific time. Even so, the concept of adding a +/- 5 minute buffer to the predicted delivery time helps to account for any parking or security constraints, and the customer may be none the wiser. By ensuring the delivery will be made by a certain time, not only will the customer be satisfied, but the company leaders will be satisfied as well knowing that their deliveries are being made in the most time-efficient manner possible.