An Explanatory Analysis of Food Delivery Using Doordash Data

SI 618 Project II Report

Sherry Zhang – xingyuez

Dec 2022

Motivation

Have you experienced that after a long, tiring weekday, you came home totally exhausted so you ordered a food delivery? You were told that your food would arrive in 30 minutes, then you stared at your screen and waited and waited and waited. A half-hour after placing your order, your food was still being prepared. Ten minutes later your driver finally picked up your order. Another ten minutes later you found your driver got stuck in traffic, and finally, one hour after placing your order, your food finally arrived. The experience was so frustrating not only because it took one hour for your order to arrive, but also due to the fact that the delivery time was much longer than estimated. As a customer who does not wish to wait forever for my food delivery, I would like to use data analysis skills to examine what influences delivery duration and how well we can predict it. Meanwhile, for a company like Doordash, a reasonable delivery time estimate is very important, since it has a significant impact on customer satisfaction.

Data Source

The dataset I will be using in the project is from the take-home assignment in the recruitment process for the data science positions at DoorDash. It is publicly available on the stratascratch website. The data contains a subset of DoorDash deliveries received in early 2015 in a subset of cities. Each row in this file represents a unique delivery. There has been noise added to the dataset by DoorDash in an attempt to obscure certain details of the business.

This dataset contains 197428 orders and 15 features including city/region id, restaurant type, food price, number of available dashers, and other critical features to explore these orders. A detailed description of features will be found on the stratascratch website.

Research Questions

There are three major questions I want to explore for this dataset:

1. How does time affect the delivery time?
    - Does delivery time vary with the day of the week?

    - Will the order take longer to arrive during rush hour?
2. Is there a difference in delivery time between different prices and types of food?
    - Does the expensive food take longer to arrive?
    - Does restaurant type have some impact?
3. How well we can predict the delivery time?
    - What features are important and what are not?
        - Can we use PCA to reduce the dataset dimensions while still preserving the most of information?
    - Which model gives us the most accurate prediction and how accurate is it?

Methods

In this section, I will describe how I processed each research question from three aspects: data preparation, missing value or outlier treatment, and challenges encountered.

Q1: How does time affect the delivery time?

x_1 = day of the week, x_2 = hour of the day, y = delivery time, here I'd like to find out the relationship between x_1 and y, x_2 and y.

Data preparation:

- Used `to_datetime` function in the pandas package to convert features `"created_at", "actual_delivery_time"` into Datetime type.
- Subtract `"created_at" and "actual_delivery_time"` to get the actual total delivery duration, i.e. y
- `dt.weekday, dt.hour` functions were applied to get the day of the week, i.e, x_1, and the hour of the day, i.e.x_2, when the order was placed

Missing value and outliers treatment:

There are no missing values in `"created_at" and "actual_delivery_time"` features. However, outliers exist in `"actual_total_delivery_durations ("created_at" - "actual_delivery_time"`), the minimum delivery time is 101 seconds (1.68 mins) and the

maximum is 8.5×10^6 seconds (99 days), they do not seem to make sense. There are 1091 orders that arrived after more than 2 hours after the order was placed, which represents just 0.55% of the whole dataset. Since it is rare for an order to arrive more than 2 hours or less than 3 minutes after it is placed, so I decided to remove these rows.

Challenges:
Since these timestamps in the dataset are in UTC, while the actual timezone of the region was US/Pacific, tz_localize, tz_convert were used to convert the DateTime features into the correct time zone.

Q2: Is there a difference in delivery time between different prices and types of food?

Food price vs delivery duration

x = total price of the food, y = delivery time, I want to know the relationship between x_1 and y, x_2 and y.

Data preparation:
This analysis simply involved using a scatter/joint plot to see the correlation between x_1 and y. Not much data manipulation was needed for preparing data for analysis. Nonetheless, based on the joint plot as seen later, the total price of food is highly right-skewed, thus I decided to use `qcut` function to discretize food price i.e. `"subtotal"` into three almost equal-sized buckets: cheap, medium, and expensive, then used `groupby`, box plot to visualze the difference in delivery duration among these three groups.

Missing value and outliers treatment:
There are no missing values in `"subtotal"` column.
Some zeros exist in the `"subtotal"` column. The food price being equal to 0 is likely due to customers using coupons, whereas it is more likely due to an error in data entry. I decided to fill the 0 with the median of other values in the `"subtotal"` column. The reason why I used

median instead of mean is that the total price of the order is highly right-skewed thus median is more representative of the whole column.

Challenges:

As the total order price is highly right-skewed, it is difficult to discern a pattern from the scatter plot since most points are concentrated on the left side. Consequently, the continuous price must be divided into categories with a similar amount of data in each.

Restaurant type vs delivery duration

x = type of the restaurant, y = delivery time, I need to explore the relationship between x_1 and y, x_2 and y.

Data preparation:

This dataset contains too many types of restaurants, 74 in total, so it would be more meaningful and feasible to compare duration time within the most popular restaurant types.

- `groupby` and `sort_values` were applied to extract the 10 most popular restaurant types, from which, I picked five types: American, Mexican, Chinese, Japanese, and dessert for the following analysis.
- `qcut` was used to categorize the delivery time into 3 almost equal-sized bins: "fast", "normal", and "slow"

Missing value and outliers treatment:

There are no outliers in the categorical data.

Since nonvalues in the restaurant type column accounted for only 2.5% of the total dataset, I decided to simply drop 4724 rows containing nans in `"store_primary_category".`

Challenges:

I initially intended to use the five types for comparison, while the top five in popularity were: American, pizza, Mexican, burger, and sandwich. Well, four of them have an obvious overlap. American, pizza, burger, and sandwich can all be categorized as American foods. This led me to choose other types of restaurants that are more representative.

How well we can predict the delivery time?

Data preparation:
Used one-hot to encode categorical data including different types of restaurants, market_id (A city/region in which DoorDash operates), order_protocol: (a store can receive orders from DoorDash through many modes) etc. I then calculated the correlation between all variables in the dataset, and dropped or created new variables to represent the highly correlated variables. Next, I attempted to reduce the dataset's dimension using PCA. Afterward, I split the dataset into 80% training and 20% testing, and tested six models (Ridge, DecisionTree, RandomForest, XGBoost, LGBM, MLP) on the dataset, selecting the best model with the lowest RMSE (Root Mean Square Deviation).

Missing value and outliers treatment:
Rows containing null value takes only about 11% of the whole dataset, it is reasonable to drop all the these rows without losing much information.

Challenges:
There are almost 100 columns in the dataset after one-hot encoding, which means there might be redundant features. These features are highly correlated with others and therefore doesn't add any new knowledge to prediction models. Thus I had to check all of them and delete some features with high correlation, or created new features to represent high-correlated features if the they contain some valuable information.

Analysis and Results

In this section, I will briefly describe the workflow of my source code and present the analysis results along with the visualizations in order to draw a conclusion regarding the research questions.

Q1: How does time affect the delivery time?

Q1a. Does delivery time vary with the day of the week?

After extracting the day of the week from the datetime variable `"created_at"` (when the order was submitted by the consumer to DoorDash) as described in the Method part, I used boxplot to visualize the relationship between the day of week that order created and the delivery duration.

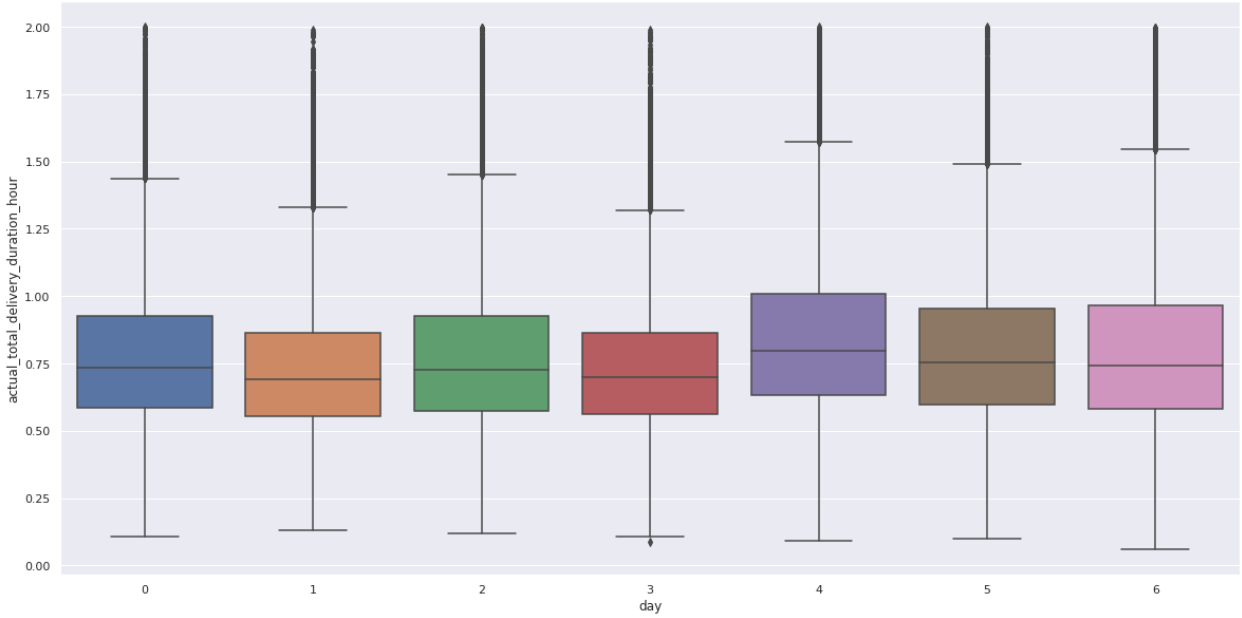

Figure 1: Average delivery duration vs day in a week

From the boxplot above, we can see that delivery duration does not vary much across different days of the week.

Since it was not easy to tell directly from the graph if there was a correlation between the day of week and duration time, I wanted to conduct an ANOVA test to validate this. Considering the dataset is too large (196330 rows on the current step), and a sample size cannot be too large for an ANOVA since we will almost invariably get statistically significant results. I randomly sampled 200 rows and conducted ANOVA test on the sampled dataset.

	sum_sq	df	F	PR(>F)
C(day)	8.160627e+06	6.0	1.238754	0.288259
Residual	2.075147e+08	189.0	NaN	NaN

Table 1: Anova Test for Day of the week when the order was created ~ Delivery duration

Based on the results, the p value is 0.288 which is significantly high. It's fair to assume that there is no significant difference in delivery duration in different days of the week.

Q1b. Will the order take longer to arrive during rush hour?

Similas as Q1a, I extracted the hour of a day the datetime variable "created_at" and used violin plot to visualize how the average delivery duration will change over one day.

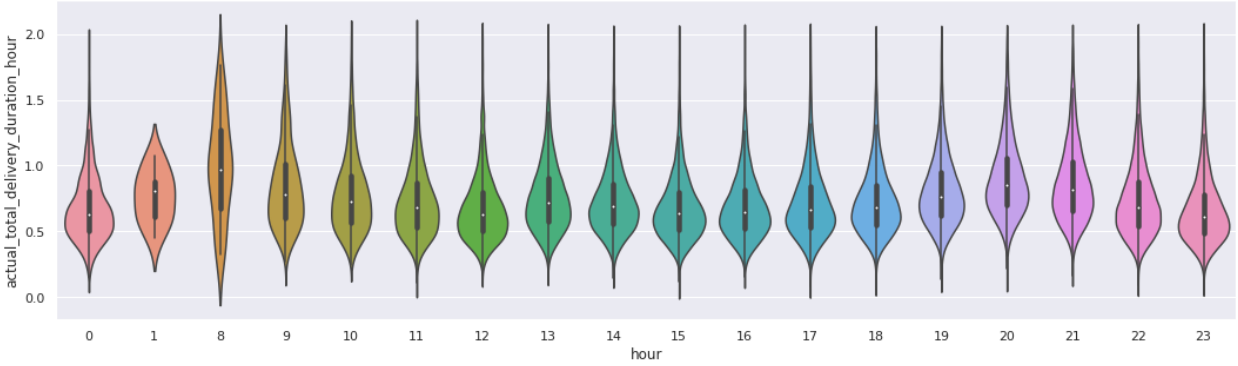

Figure 2: Average delivery duration vs hour in a day

Delivery duration also does not vary much across different time in a day. Among the hourly data we have, the order placed at 8am has the **longest delivery duration** and **the largest variance**. The longest average delivery duration can be explained by the heavy traffic during morning rush hour. The largest variance can be explained by the small number of orders placed at 8 am; only 36 orders were placed at that time, accounting for only 0.018% of the dataset.

Q2: Is there a difference in delivery time between different prices and types of food?

Q2a: Does the expensive food take longer to arrive?

It is reasonable to assume that the more expensive the food, the longer it may take to prepare, and thus, the longer the delivery time. To test my hypothesis, I made a regression plot to check the total price of the order and the delivery duration. From the plot below, we can see that tell the delivery duration increase slightly with the total value of the order.

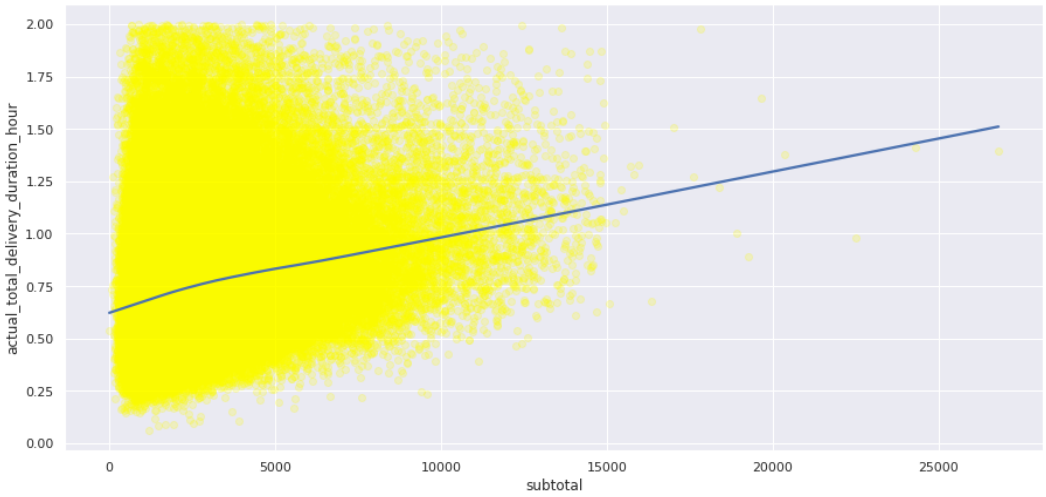

Figure 3: Average delivery duration (in hour) vs total food price (in cent)

I also used correlation function to get the correlation between these two variables:

```
data["subtotal"].corr(data['actual_total_delivery_duration'])
```

and got a 0.23 correlation coefficient, which can be considered as a weak positive relationship.

From the plot above, it is actually head to discern a pattern since most points are concentrated on the left side, thus I divided the price into three bins: "cheap", "medium", "expensive", each category contains a similar amount of data. Then I used groupby to calculate the mean delivery time in each category, and boxplot to visualize the difference among three food price levels. The table and figure below prove the hypothesis above, that the more expensive the food, the longer it will take for the order to arrive.

subtotal_cat	avg
cheap	0.721840
medium	0.774145
expensive	0.857126

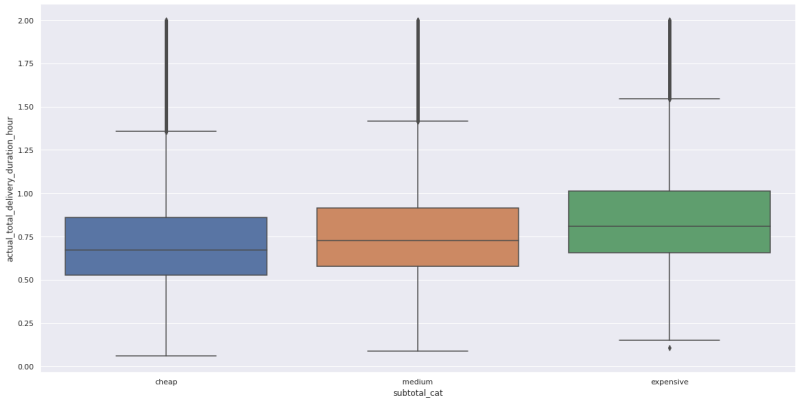

Figure 4: Average delivery duration (in hour) vs food price level

Q2b: Does restaurant type have some impact on delivery duration?

As described in the method part, I chose five restaurants types from the ten most popular restaurant types: American, Mexican, Chinese, Japanese, and dessert to compare delivery duration among them. And I also cut the delivery time into 3 almost equal-sized bins: "fast", "normal", and "slow".Then I used a mosaic plot to see what the portion of fast or slow order for different types of restaurants.

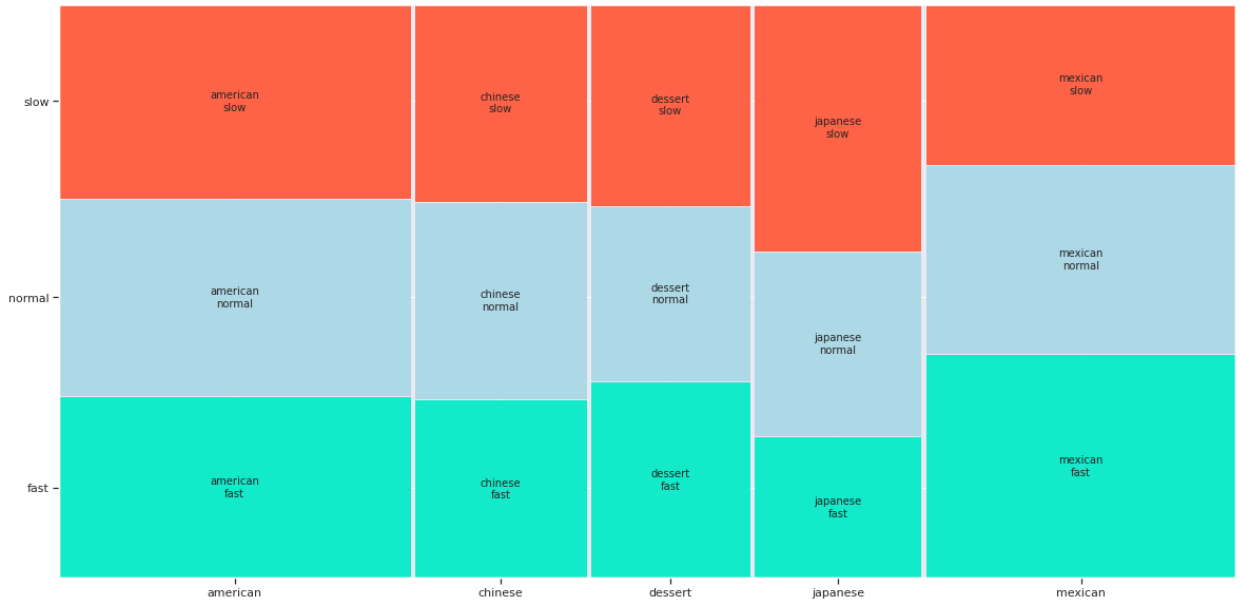

Figure 5: delivery duration level vs restaurant type

As we can see, food from Japanese restaurants takes longer to arrive than food from Mexican restaurants. That makes sense if you have been to a sushi place like sadako or of Rice and Men in Ann Arbor, it takes around at least ten minutes if you order rolls, but getting your food at Chipotle might take just two minutes.

Q3: How well we can predict the delivery time?

Q3a: What features are important and what are not?

Collinear features are considered redundant and repetitive since they share the same information as other features. Using correlation function, I calculated how closely the variables are related to

each other and extracted the 20 features with the highest correlation. From the table below, we can see that variables like “total_on_shift_dashers”, “total_outstanding_orders”, “order_protocol_1.0” etc are highly correlated with other variables, thus needed to be removed.

Top Absolute Correlations		
total_onshift_dashers	total_busy_dashers	0.943602
	total_outstanding_orders	0.936331
total_busy_dashers	total_outstanding_orders	0.932995
estimated_store_to_consumer_driving_duration	estimated_non_prep_duration	0.923815
estimated_order_place_duration	order_protocol_1.0	0.900518
total_items	num_distinct_items	0.757643
subtotal	num_distinct_items	0.680902
total_items	subtotal	0.554352
min_item_price	max_item_price	0.540874
subtotal	max_item_price	0.509666
order_protocol_4.0	category_fast	0.501208
num_distinct_items	min_item_price	0.447064
market_id_2.0	market_id_4.0	0.395948
total_items	min_item_price	0.389650
total_onshift_dashers	hour	0.376068
order_protocol_1.0	order_protocol_3.0	0.373279
estimated_order_place_duration	order_protocol_3.0	0.363383
total_outstanding_orders	hour	0.363116
estimated_order_place_duration	estimated_non_prep_duration	0.361938
total_busy_dashers	hour	0.351390
dtype: float64		

Table 2: Top 20 features with high correlations

Then I used the built-in attribute `.feature_importances_` in the Random Forest algorithm to evaluate feature importance based on gini impurity after dropping the variables with high collinearity. The most important feature is “estimated_store_to_consumer_driving_duration”, which indicates how far between store and consumer. Other important attributes include “total_outstanding_orders”, which are the number of orders within 10 miles of this order that are currently being processed; “busy_dashers_ratio”, “ avg_price_per_item”, “price_range_of_items”, “day”, “percent_distinct_item_of_total”, “estimated_order_place_duration”, “category_pizza”.

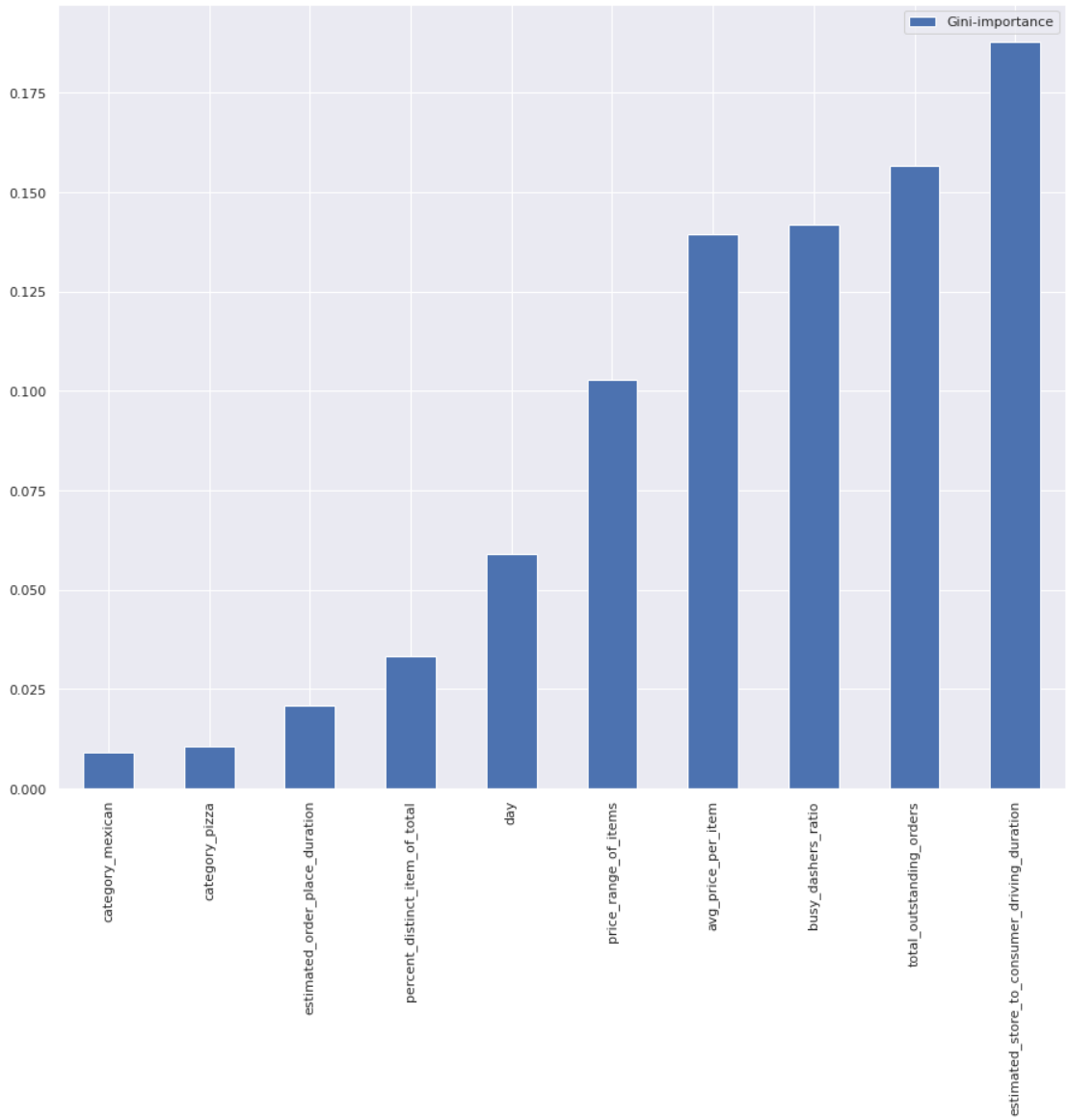

Figure 6: the 10 most important features evaluated based on gini impurity

Q3b: Can we use PCA to reduce the dataset dimensions while still preserving the most of information?

"Elbow" plot, i.e. the eigenvalue vs the number of components was different from any elbow plot I've seen before, it seems the turning point occur at 5 features while more than 60 features have eigenvalue greater than 1, which is a common criteria that we should retain factors, thus we may not be able to directly use 5 components to represent the whole dataset.

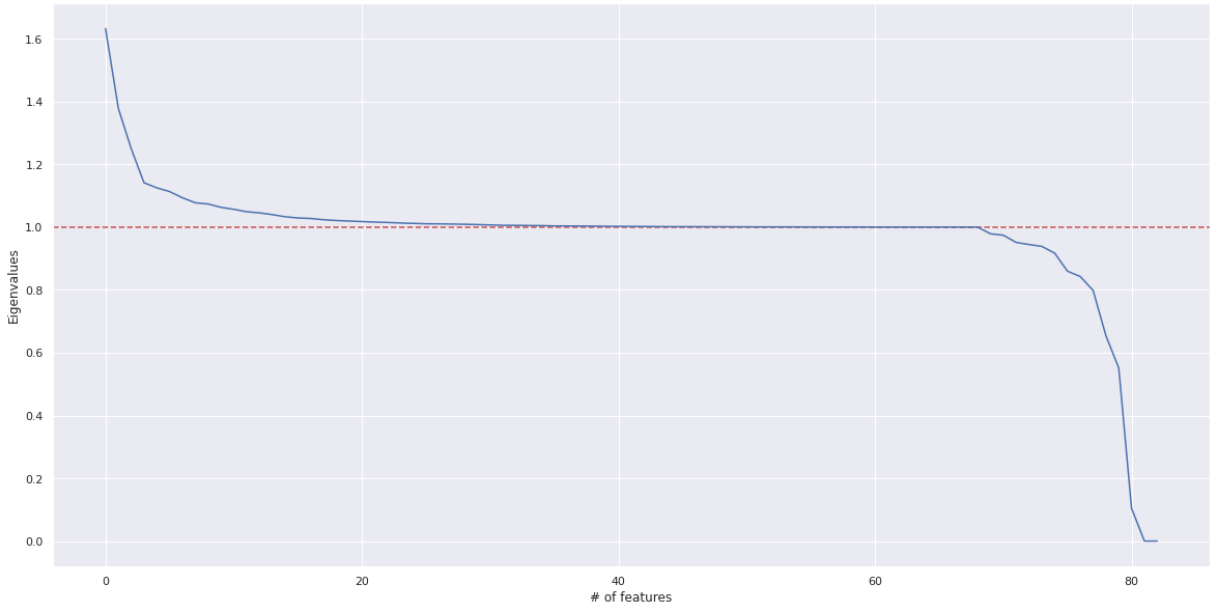

Figure 7:PCA eigenvalues vs number of features

I plot the cumulative proportion of variance explained against the number of features. Based on the plot below, 80% of the dataset can be explained by at least 60 representative features, which makes the PCA transformation useless since we already have 80 features and could select the most important ones based on the feature importance calculated above.

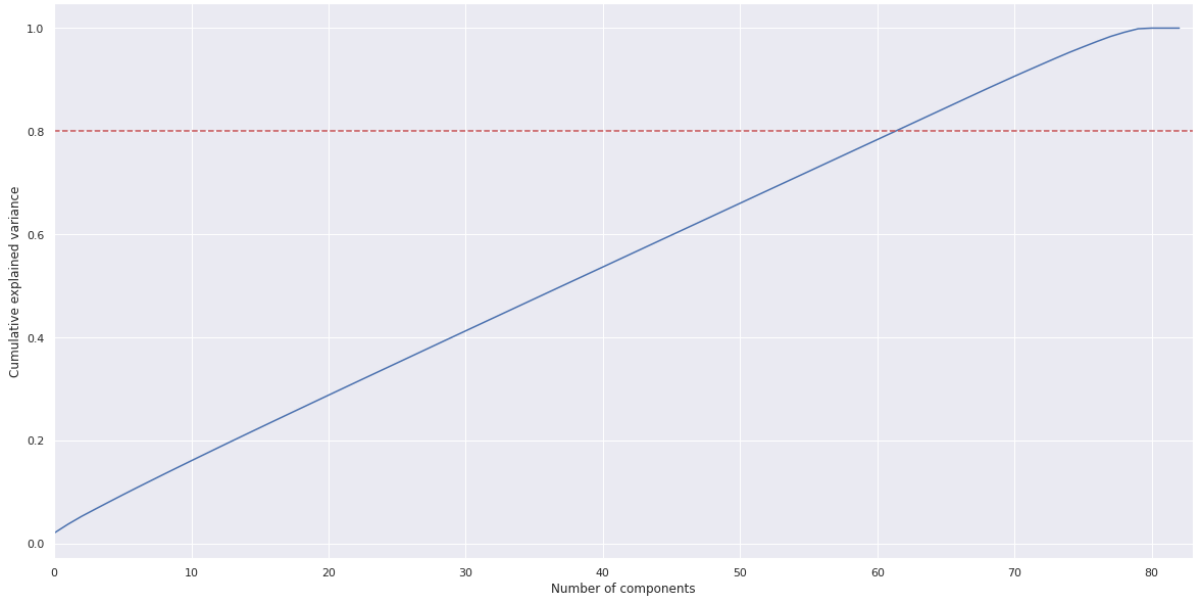

Figure 7:cumulative proportion of variance explained vs the number of features

Q3c: Which model gives us the most accurate prediction and how accurate is it?

Data was splited into 80% training and 20% test, and features are scaled using StandardScaler or MinMaxScaler, 60, 40, 10 most important features were selected to be put into six regression models. And the combination of MinMaxScaler, using 60 features, and Light GBM(Light Gradient Boosting Machine) gave the best prediction with a RMSE equals to 873, that indicates we can predict the delivery duration using 60 features within an error controlled in 14.5 mins.

Conclusions

Based on the analysis results, I can give answers to the previous research questions:

1. The delivery duration time does not vary much with the day within a week, which means you will wait for almost the same time no matter on which day you order
2. Delivery time doesn't vary much within a day, but if you order at 8 a.m., food takes longer to arrive
3. Expensive food takes longer to prepare thus longer to arrive, and different restaurant types affect the duration time. If you are hungry, don’t order fancy food, it would better to order some fast food from Mexican restaurants.
4. The distance you are from the restaurant largely determines the duration of your order, which is quite obvious. Also, the total order price, the number of items you order, and how many dashers are busy make a huge difference.
5. Even with these features and model, it's still difficult to predict delivery duration precisely. However, Light GBM can give the best prediction of when your order will arrive within an average error of 14.5 minutes (with extra weight added to larger prediction errors).