THE GEORGE WASHINGTON UNIVERSITY

WASHINGTON, DC

Predicting Food Delivery Time: An Analysis of Machine Learning Models and Variables

Chirag Lakhanpal

The George Washington University

DATS 6401: Visualization of Complex Data

Professor Reza Jafari

May 9th, 2023

Table of Contents

Table of figures

Abstract

Accurate delivery duration predictions are crucial for maintaining a positive customer experience in the food delivery industry. In this study, I've developed a machine learning model to predict the total delivery duration (in seconds) for DoorDash orders. The dataset, provided by DoorDash, included historical data from early 2015 for a subset of cities. Key features included market_id, created_at, actual_delivery_time, store_id, store_primary_category, order_protocol, total_items, subtotal, num_distinct_items, min_item_price, max_item_price, total_onshift_dashers, total_busy_dashers, total_outstanding_orders, estimated_order_place_duration, and estimated_store_to_consumer_driving_duration. I preprocessed the data by cleaning missing values, encoding categorical variables, and normalizing numerical features. Next, I engineered additional features to enhance model performance. A range of regression models were evaluated, with the final model selected based on its performance in cross-validation. Model performance was assessed using mean absolute error and R-squared metrics. Our findings suggest that the proposed model can effectively predict delivery durations for DoorDash orders, with potential implications for improving customer satisfaction and operational efficiency. Future work may include incorporating real-time data, such as traffic conditions, and exploring more advanced modeling techniques to further improve prediction accuracy.

Introduction

In today's fast-paced world, time is of the essence, and people are constantly seeking ways to save it. One area where time is of critical importance is in the food delivery industry, where consumers expect their orders to be delivered promptly and efficiently. The "Delivery Duration Prediction" dataset offers an exciting opportunity to understand the factors affecting food delivery times and to develop a predictive model that can estimate delivery times with higher accuracy. This study aims to explore this dataset, analyze its various features, and ultimately build a model that can assist consumers, food delivery services, and restaurants alike in optimizing their delivery processes.

The dataset includes over 2 million food deliveries, providing detailed information on factors such as time of day, distance between the restaurant and customer, food preparation time, and actual delivery time. By analyzing these factors, we can gain valuable insights into the factors that influence delivery times and potentially improve the overall efficiency of the food delivery process. This, in turn, can lead to increased customer satisfaction and a more successful food delivery industry.

This report will outline the steps taken to preprocess and analyze the "Delivery Duration Prediction" dataset, including data cleaning, outlier detection and removal, principal component analysis, normality testing, and data transformation. It will also delve into various data visualization techniques to explore patterns and trends in the dataset. Finally, the report will discuss the development of a web-based interactive dashboard using Python and Dash, which can be deployed on Google Cloud Platform (GCP) to allow users to explore the dataset and its insights in real time.

Data Description

Description	Value
Dataset Name	Delivery Duration Prediction
Number of Observations	197,428
Number of Features	16
Categorical Variables	4
Numerical Variables	10
Timestamp Variables	2
Target Variable	ETA (Estimate Time of Arrival)
Data Timeline	Jan 21st, 2015 to Feb 18th, 2015

Feature Description

Features	Description
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 consume
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	
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	

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)

Data Preprocessing

Identifiers Detection

Upon examining the dataset, it was observed that there were no identifiers, such as customer ID, that could potentially interfere with the model's accuracy. This ensures that the model remains unbiased and provides a reliable prediction of delivery times.

Missing Values Eradication

	var	proportion	dtype
0	total_onshift_dashers	0.08237	float64
1	total_busy_dashers	0.08237	float64
2	total_outstanding_orders	0.08237	float64
3	store_primary_category	0.02411	object
4	order_protocol	0.00504	float64
5	market_id	0.00500	object
6	estimated_store_to_consumer_driving_duration	0.00266	float64
7	actual_delivery_time	0.00004	object

Figure 1: The figure displays the distribution of missing values and their respective proportions.

The dataset contained missing values in several variables, with total_onshift_dashers (8%), total_busy_dashers (8%), total_outstanding_orders (8%), store_primary_category (2%), order_protocol (0.5%), market_id (0.5%), and estimated_store_to_consumer_driving_duration (0.2%) being affected. These missing values were addressed using appropriate imputation techniques based on the variable's data type and distribution, ensuring a complete and robust dataset for further analysis.

Missing Value Imputation

Missing values in the dataset were imputed using the SimpleImputer method from the scikit-learn library. This method replaces missing values (NaN) in the numerical columns with the mean value of each respective column in the datasets.

Missing Value Deletion

After the initial imputation of missing values using the SimpleImputer method, any remaining missing values were minimal in number. Consequently, these instances were removed from the dataset using the missing value deletion technique to maintain data integrity and ensure a complete dataset for subsequent analysis.

Note: Post dealing with the missing values, 1,91,407 observations remain along 16 features.

Datatype Mapping

In the dataset, there are 10 numerical, 4 categorical, and 2 timestamp variables. These variables must be correctly mapped and transformed to facilitate precise data manipulation, analysis, and modeling. By ensuring that each variable is represented in its appropriate format, the quality of the dataset is maintained, and the performance of any subsequent modeling efforts is optimized. This also appropriate operations can be performed on the features.

Feature Readability

To enhance the readability and interpretability of the dataset, the feature values of subtotal, min_item_price, and max_item_price are converted from cents to dollars. This conversion facilitates a more intuitive understanding of the data, making it easier to analyze and draw insights from these features. By presenting the monetary values in a familiar format, it becomes more convenient to work with the dataset.

Redundant Features

During the data analysis and investigation, it was determined that the store_id feature did not have a significant impact on the analysis or the predictive modeling process. As a result, removing this feature from the dataset can help to streamline the data processing and reduce the dimensionality of the dataset, which in turn can lead to more efficient and accurate model training. By focusing on the most relevant features, we can ensure that the resulting model is both interpretable and robust in its predictions.

Encoding Features

	var	nunique
0	store_primary_category	74
1	order_protocol	7
2	market_id	6

Figure 2: Unique Values of Categorical Variables for Encoding.

In the final dataset, there are several categorical variables that require encoding for proper integration into the modeling process. Specifically, the store_primary_category, order_protocol, and market_id variables have 74, 7, and 6 unique values, respectively. To effectively incorporate these categorical variables, suitable encoding techniques are employed, which results in a total of 99 features within the final dataset. The transformation of these categorical variables into a suitable format ensures that the predictive model can efficiently process and analyze the data to generate meaningful insights and predictions.

Outlier Detection and Treatment

	Column	Outlier Count	Percentage
1	subtotal	3673	1.92
4	max_item_price	2989	1.56
13	avg_price	2949	1.54
2	num_distinct_items	2901	1.52
3	min_item_price	2700	1.41
0	total_items	2479	1.30
7	total_outstanding_orders	1991	1.04
15	order_density	1691	0.88
11	busy_rider_ratio	1173	0.61
5	total_onshift_dashers	763	0.40
6	total_busy_dashers	510	0.27
12	non_prep_duration	314	0.16
9	estimated_store_to_consumer_driving_duration	263	0.14
14	delivery_distance	263	0.14
8	estimated_order_place_duration	130	0.07
10	eta	3	0.00

Figure 3: Outlier count and percentage for each column, highlighting the columns requiring outlier treatment.

An essential step in the data preparation process is the identification and treatment of outliers. Outliers can have a significant impact on the performance of predictive models and can lead to inaccurate results. In this analysis, the Interquartile Range (IQR) method was initially considered for outlier detection; however, it was found that using IQR would result in the removal of approximately 60,000 records, which would significantly reduce the dataset size. Instead, a more realistic range for each variable was defined, leading to a more balanced treatment of outliers. The table below summarizes the outlier count and percentage for each column in the dataset. By addressing these outliers, we can ensure more robust and accurate predictions.

Feature	From	To	Comments
total_items	0	15	Let's keep it small like a family dinner
subtotal	0	$90	No more extra guac, please
num_distinct_items	0	15	Can't handle too many choices
min_item_price	0	Up to max observed	No more discounts for negative prices
total_onshift_dashers	0	Up to max observed	Let's keep it positive for our riders
total_busy_dashers	0	Up to max observed	Can't have negative busy dashers
total_outstanding_orders	0	Up to max observed	No negative orders, we are not in debt here
eta	0	1.5 hours	No one should wait for their food for more than 1.5 hrs.
busy_rider_ratio	0	Up to max observed	Our riders are always busy, in a good way
non_prep_duration	0	1.5 hours	Let's prepare the food and deliver it ASAP

Standardizing Dataset

In the preprocessing phase, the dataset was standardized using the StandardScaler method, which employs the z-score normalization technique. This process involves scaling the numerical columns (both 'float64' and 'int64' data types) to have a mean of 0 and a standard deviation of 1. Standardizing the data is an essential step in machine learning, as it ensures that all features contribute equally to the model, preventing any feature from dominating due to differences in scale. By applying the StandardScaler method, we can achieve more accurate and reliable model performance.

Feature Engineering

An essential aspect of the modeling process is feature engineering, which involves creating new features from existing ones to enhance the predictive power of the model. The following table presents the newly created features, their corresponding original features, and a brief description of each.

These engineered features aim to capture more complex relationships within the data and provide additional information that can improve the predictive capabilities of the model.

Feature Name	Original Feature 1	Original Feature 2	Description
Target Variable (ETA)	actual_delivery_time	created_at	The estimated time of arrival for the order
Busy Rider Ratio (%)	total_busy_dashers	total_onshift_dashers	The percentage of on-shift dashers who are currently occupied with an order
Non-Preparation Duration	estimated_order_ place_duration	estimated_store_to_ consumer_driving_ duration	The duration of the delivery process, excluding the time spent at the restaurant for preparation
Day of the week	created_at	-	The day of the week when the order was placed
Average Order Price	subtotal	total_items	The average price of items in the order
Time of the day	created_at	-	The time of day when the order was placed
Delivery Duration	Eta	-	The total duration of the delivery process is categorized as <15 mins, 15 - 30 mins, 30 - 45 mins, 45 - 60 mins, and >60 mins
Trip Nature	Eta	-	The nature of the delivery trip is categorized as Fast, Moderate, Slow, Very Slow
Delivery distance	estimated_store_to_ consumer_driving_ duration	-	The estimated distance between the store and the consumer
Order Density	total_outstanding_orders	total_busy_dashers, total_onshift_dashers	The number of orders within 10 miles of the order being processed

Exploratory data analysis (EDA)

Trend Over Time by City (Line Chart)

Figure 4: Trend over time for order delivery from 21st January to 15th February. The line chart shows seasonality with peaks on weekends, indicating higher order deliveries during Saturdays and Sundays.

According to the line chart of the trend over time, which displays the order delivery trend from January 21st to February 15th, there is a clear seasonality pattern with periodic peaks occurring on weekends, specifically on Saturdays and Sundays, such as on January 24th and 25th, and January 31st and February 1st. This pattern suggests that there may be an underlying relationship between the day of the week and the order delivery time, which could be further investigated to optimize the delivery schedule and improve overall customer satisfaction. The highest orders are from the Phoenix market area and the least is from Los Angeles, which may be useful to focus the next market campaign on.

Distribution of Important Features (Histogram)

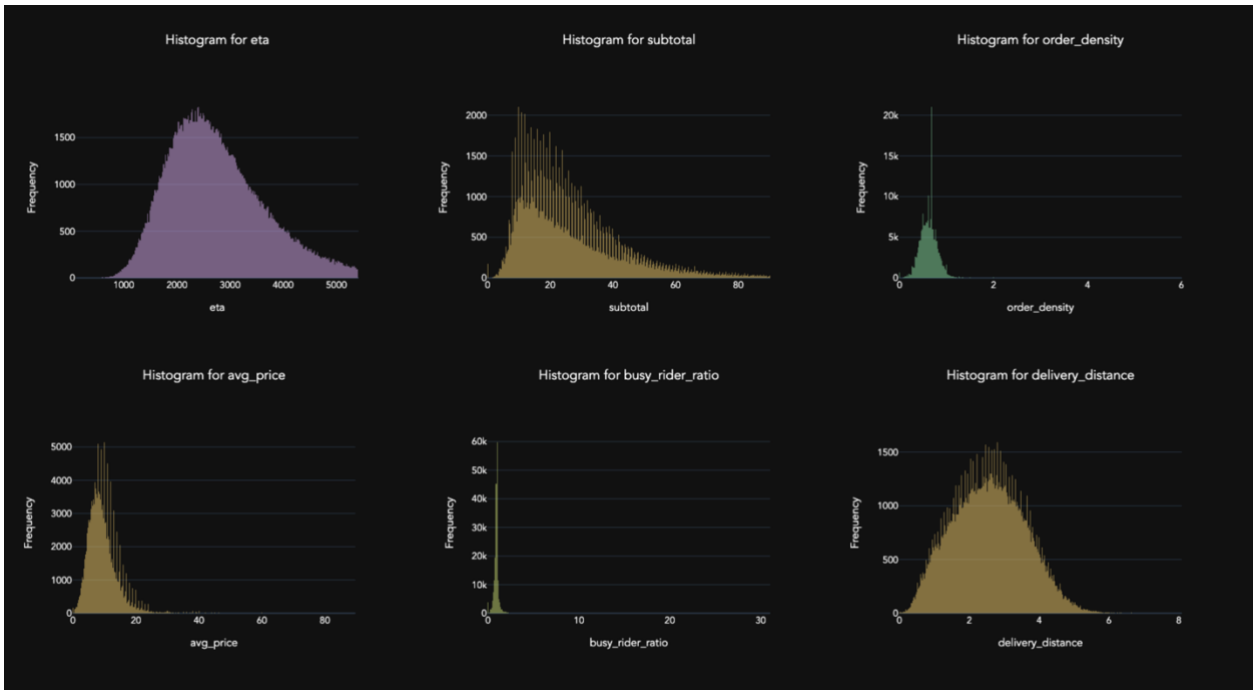

Figure 5:Histogram chart showing the distribution of ETA, subtotal, order density, avg price, busy dasher ratio, and delivery distance. All subplots are right-skewed, with ETA and delivery distance showing slightly less skewness.

The histogram chart (Figure 5) displays the distribution of various variables including ETA, subtotal, order density, average price, busy dasher ratio, and delivery distance. All subplots are right-skewed, indicating that most values are concentrated on the lower end of the scale, with a few extreme values on the higher end. This skewness is desirable for variables like ETA, order density, and delivery distance as it indicates a large proportion of the values fall within an acceptable range. However, for variables like average price and subtotal, right-skewness is not desirable as higher values are preferred. Notably, ETA and delivery distance subplots are less skewed compared to other variables.

Order Count by Time of the Day and Delivery Speed (Sunburst/Pie Chart)

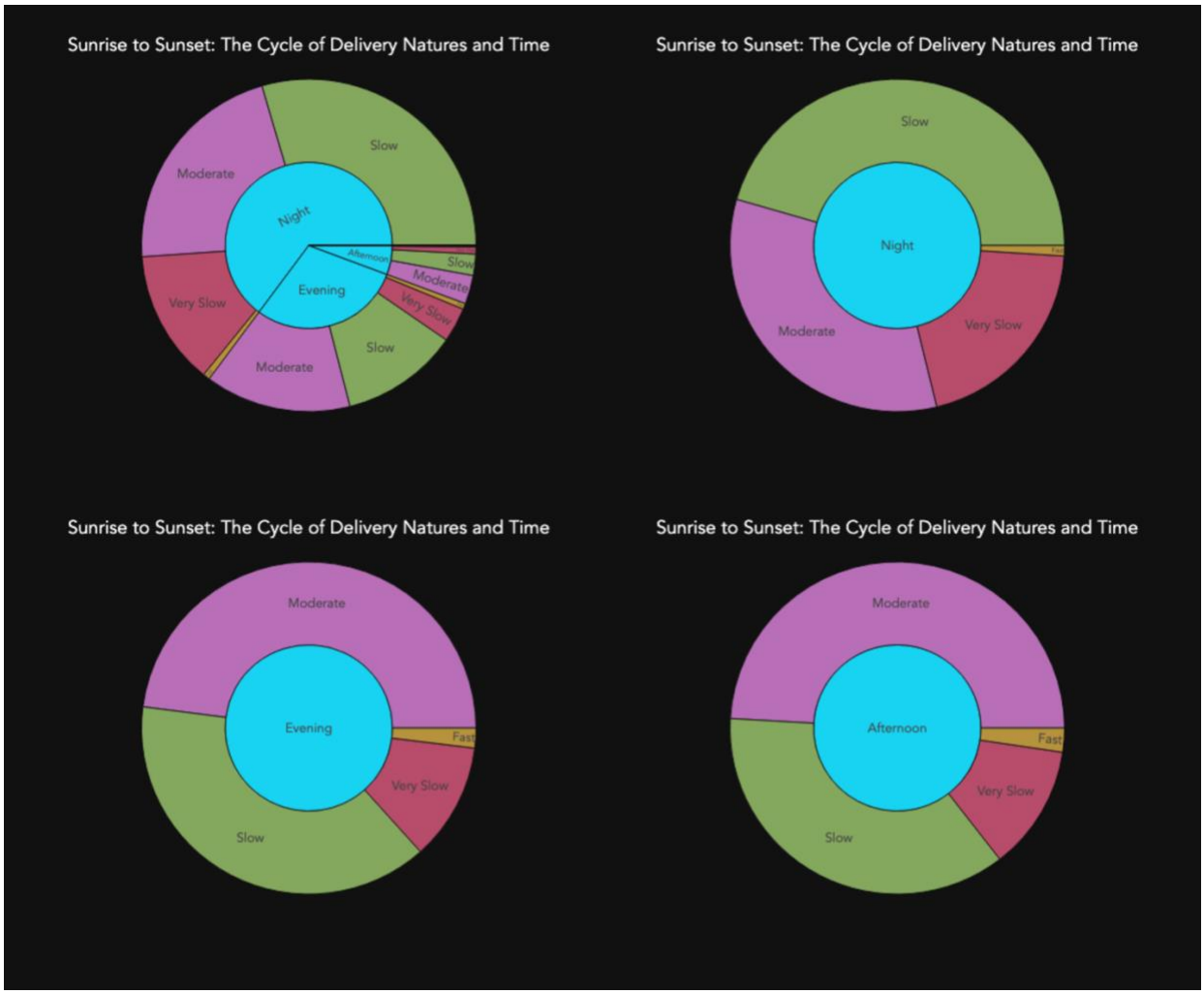

Figure 6 Sunburst Chart displaying the Distribution of Orders by Time of the Day and Delivery Status. The chart shows the split of orders at different times of the day categorized by delivery speed - very slow, slow, moderate, and fast.

According to a sunburst chart with the time of the day at the center, which displays the delivery status in terms of very slow, slow, moderate, and fast, the orders are distributed equally among different time frames. The chart reveals that slow and moderate deliveries are dominant in all the time frames, with slightly fewer deliveries during the night. The cutoffs for fast, moderate, slow, and very slow deliveries are 0-20 minutes, 20-40 minutes, 40-60 minutes, and more than 60 minutes, respectively. Additionally, the chart highlights the concerning fact that a small number of orders were delivered quickly.

Order Count by Time of the Day and Day (Stacked Bar Chart)

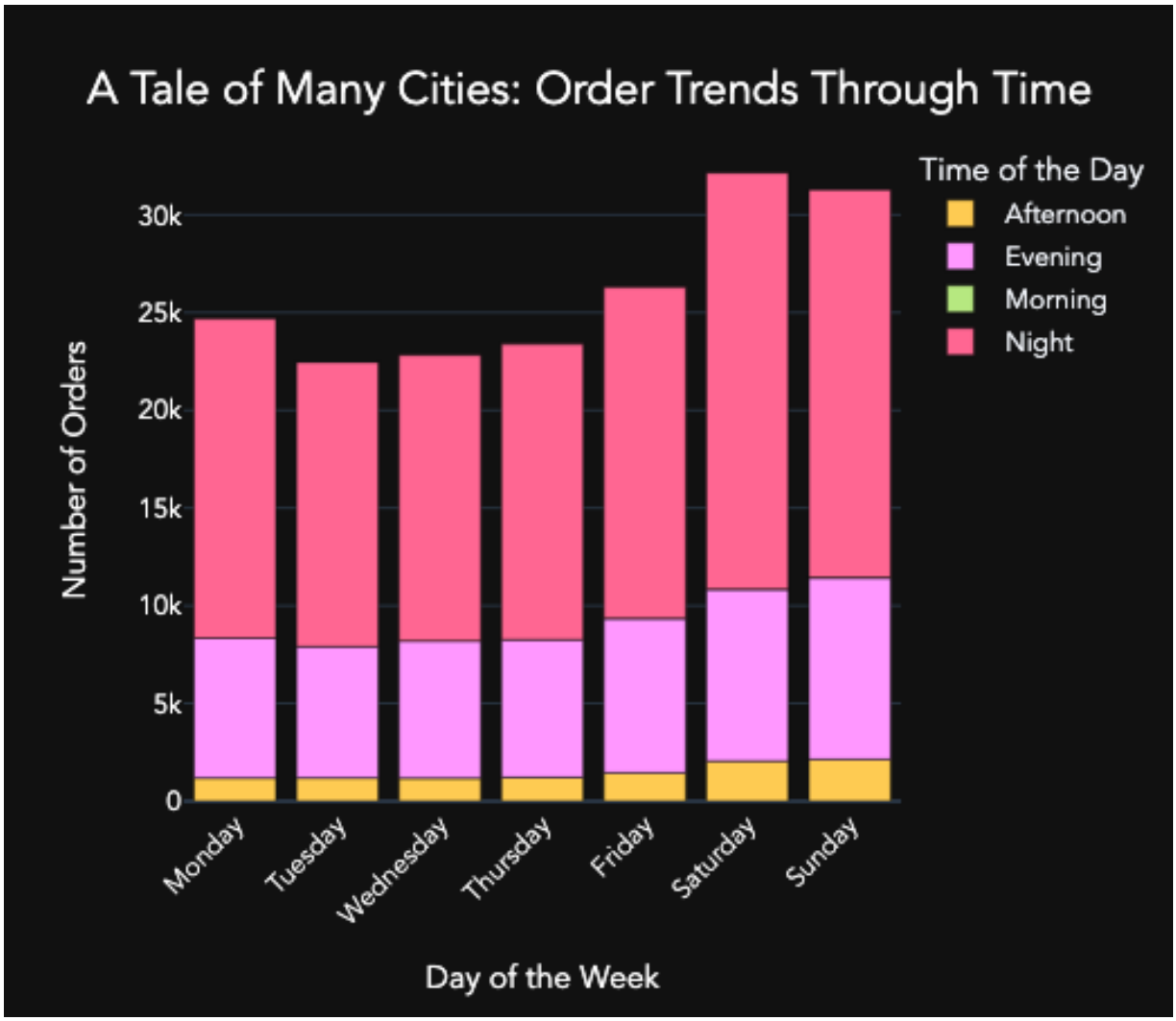

Figure 7:Stacked bar chart showing order frequency by day of the week and time of the day.

A stacked bar chart was created to visualize the number of orders placed on a weekly basis. The chart was segmented by day of the week and time of day. The results showed that the highest number of orders were placed on Saturday and Sunday, and the highest number of orders were placed in the evening and night. This information can help manage traffic on weekends in terms of rider availability and scheduling. For instance, the company could schedule more riders to work on weekends, especially in the evening and night. The company could also offer discounts or promotions on weekends to encourage more people to order food.

Bivariate Relationship (Scatter Plot)

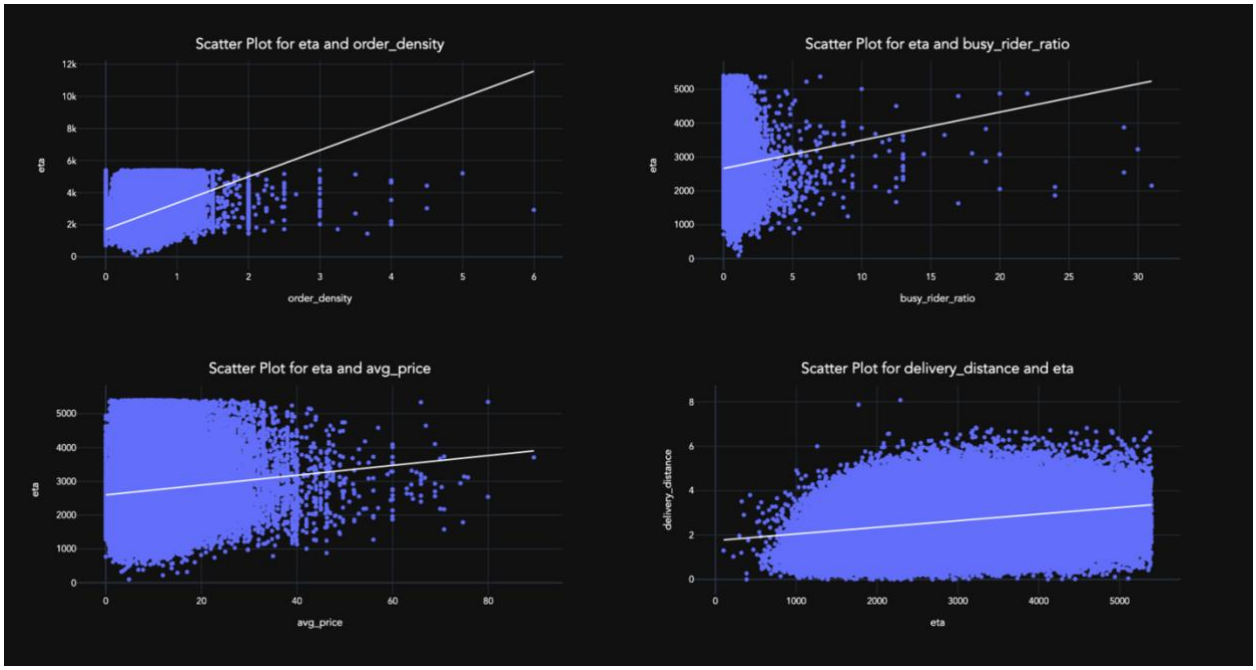

Figure 8: Four Subplot Scatterplot depicting the relationship between Average Price, Busy Rider Ratio, Delivery Distance, and Order Density with ETA (Estimated Time of Arrival).

A 4-subplot scatterplot was created to visualize the relationship between average price, busy rider ratio, delivery distance, and order density with an estimated time of arrival (ETA). The results showed that order density was fairly close to the regression line and also had the highest correlation (0.35) with ETA. The ETA vs busy rider ratio was pushed up at the start and few observations as the x-axis progressed. The ETA vs avg price had the majority of the values from avg price of 0-40. Finally, distance vs ETA had striped scatterplot distance 0-6 and eta 1000-5000 seconds. This information can help understand the relationship between these variables and ETA. For example, the high correlation between order density and ETA suggests that there is a positive relationship between the two variables. This means that as order density increases, ETA is also likely to increase. This information could be used to improve the company's operations by scheduling more riders during times of high order density.

Correlation Matrix (Heatmap)

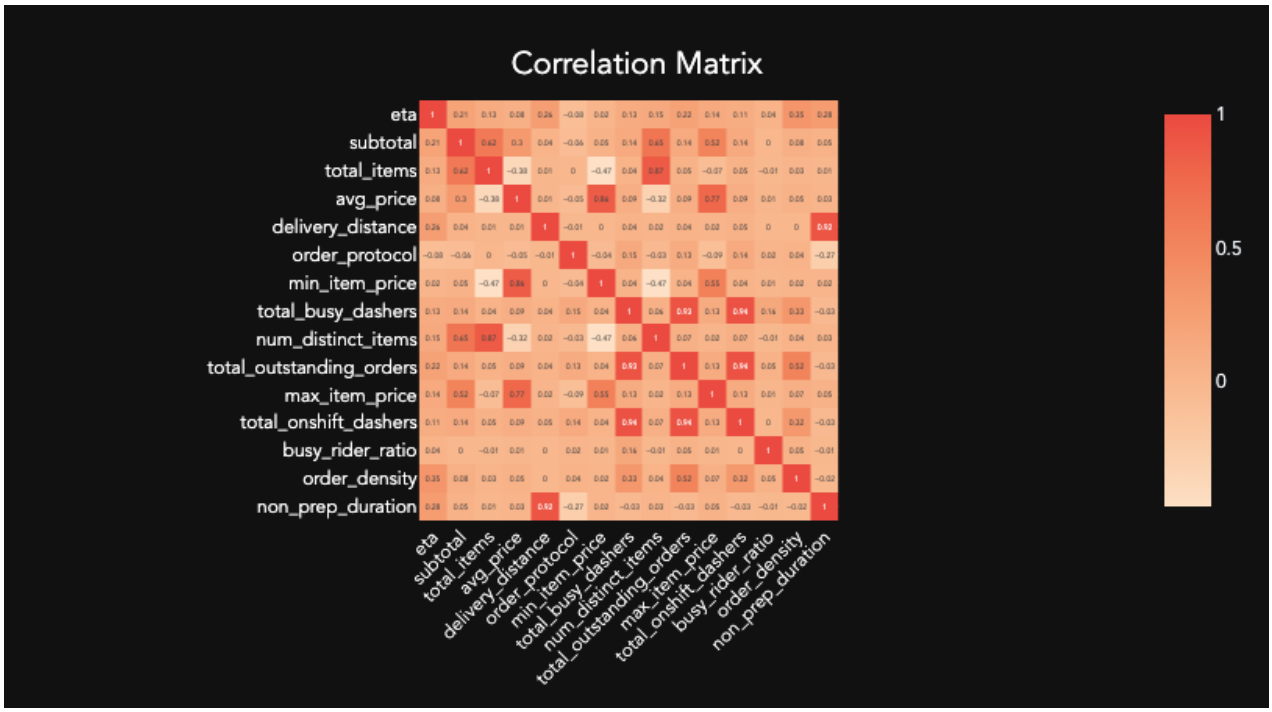

Figure 9: Correlation matrix showing Pearson's correlation coefficients between 14 variables related to food delivery service.

According to the provided correlation matrix, the highest correlation with the target variable 'eta' is observed with the predictor variable 'order_density' ($r = 0.35$). This indicates that there is a moderately positive linear relationship between these two variables. It means that as the busy rider ratio increases, the estimated time of arrival also increases. On the other hand, the lowest correlation with the target variable is observed with the predictor variable 'min_item_price' ($r = 0.02$). This suggests that there is a weak positive linear relationship between these two variables. It means that there is a small effect of the minimum item price on the estimated time of arrival. It is worth noting that the correlation is positive, which means that as the minimum item price increases, the estimated time of arrival also increases, but the effect is very small. It is important to note that correlation does not necessarily imply causation and further analysis is needed to establish the direction and causality of the relationship between these variables.

Distribution of Time taken by Total Busy Riders (Violin Plot)

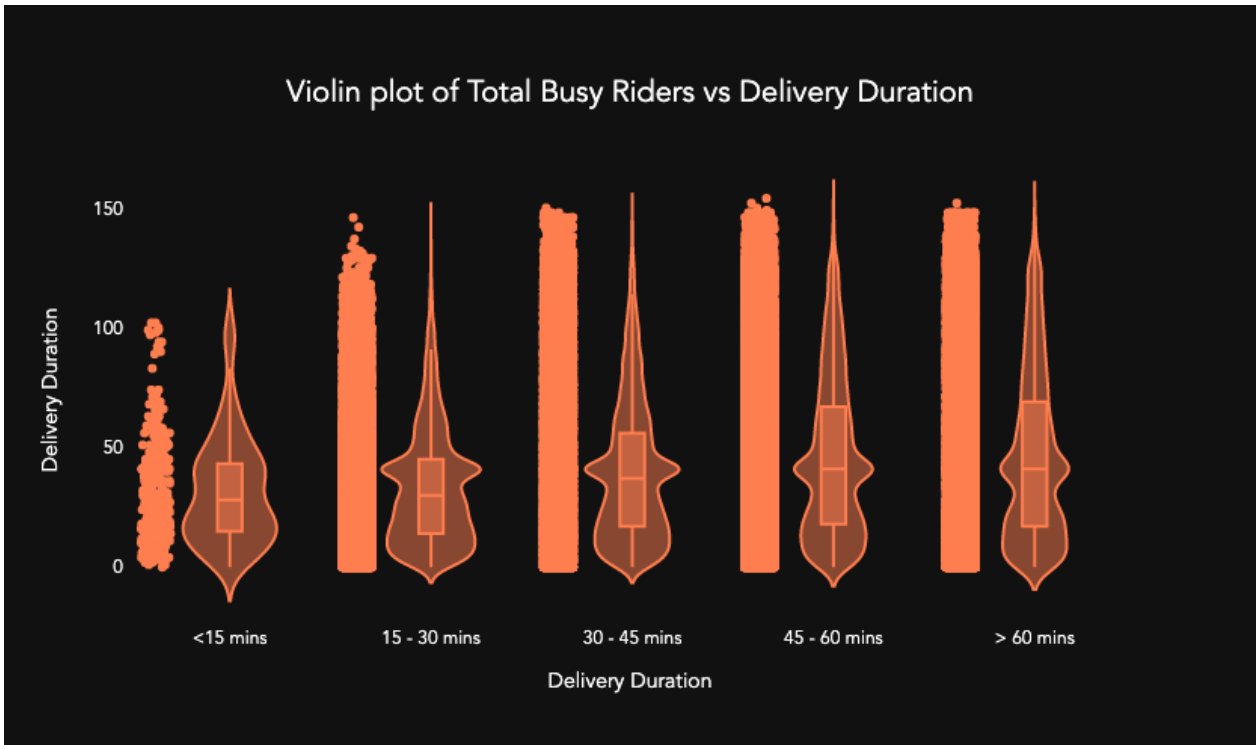

Figure 10: Violin plot showing the distribution of Delivery Duration for different categories of Total Busy Riders.

A violin plot was created to visualize the relationship between total busy riders and delivery duration. The results showed that the mean of the box plots starts from 30 riders for the <15mins delivery which is intuitive as the less busy riders mean faster delivery. This increase as the delivery duration increase and the mean touches 50 for >60 mins. The <15mins also has the lowest observations as the total busy rider increases beyond 50. This information can help understand the relationship between these variables. For example, the low mean of the box plots for the <15mins delivery suggests that the majority of deliveries are completed in less than 15 minutes when there are 30 busy riders. This information could be used to improve the company's operations by scheduling more riders during times of low busy rider ratio. The high mean of the box plots for the >60 mins delivery suggests that the majority of deliveries take more than 60 minutes to complete when there are 50 busy riders. This information could be used to improve the company's operations by scheduling more riders during times of high busy rider ratio.

Distribution of Arrival Type vs Order Value by Day (Bivariate Box Plot)

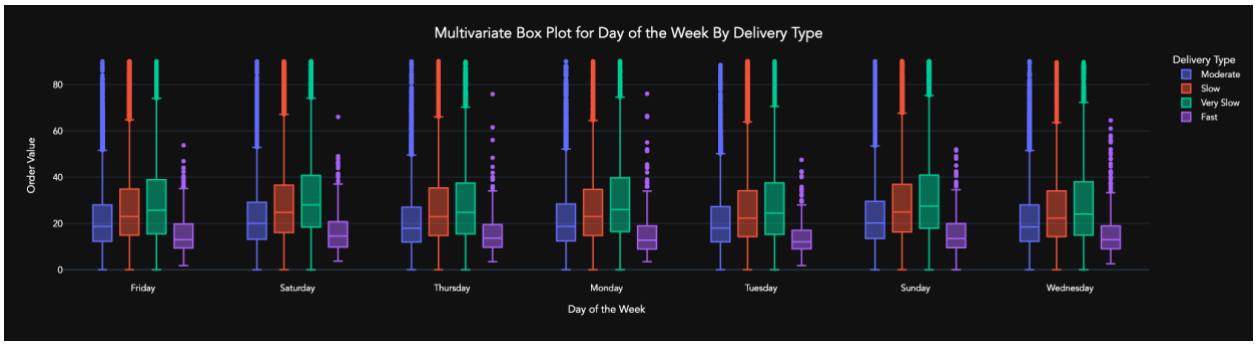

Figure 11:Multivariate box plot of order value by day of the week and delivery type

A multivariate box plot was created to visualize the order value by day and segregated by delivery type. The results showed that very slow deliveries had the highest median order value ($25) and the lowest was for very fast deliveries ($12). These results were pretty much consistent on all the days.

This information can help understand the relationship between order value, delivery type, and day of the week. For example, the high median order value for very slow deliveries suggests that customers are willing to pay more for slower deliveries. This could be because customers are willing to pay more for convenience, or because they are not in a hurry.

The low median order value for very fast deliveries suggests that customers are not willing to pay as much for faster deliveries. This could be because customers do not see the value in paying more for a faster delivery, or because they are not in a hurry.

Gaussian Distribution Check (KDE and QQ – Plot)

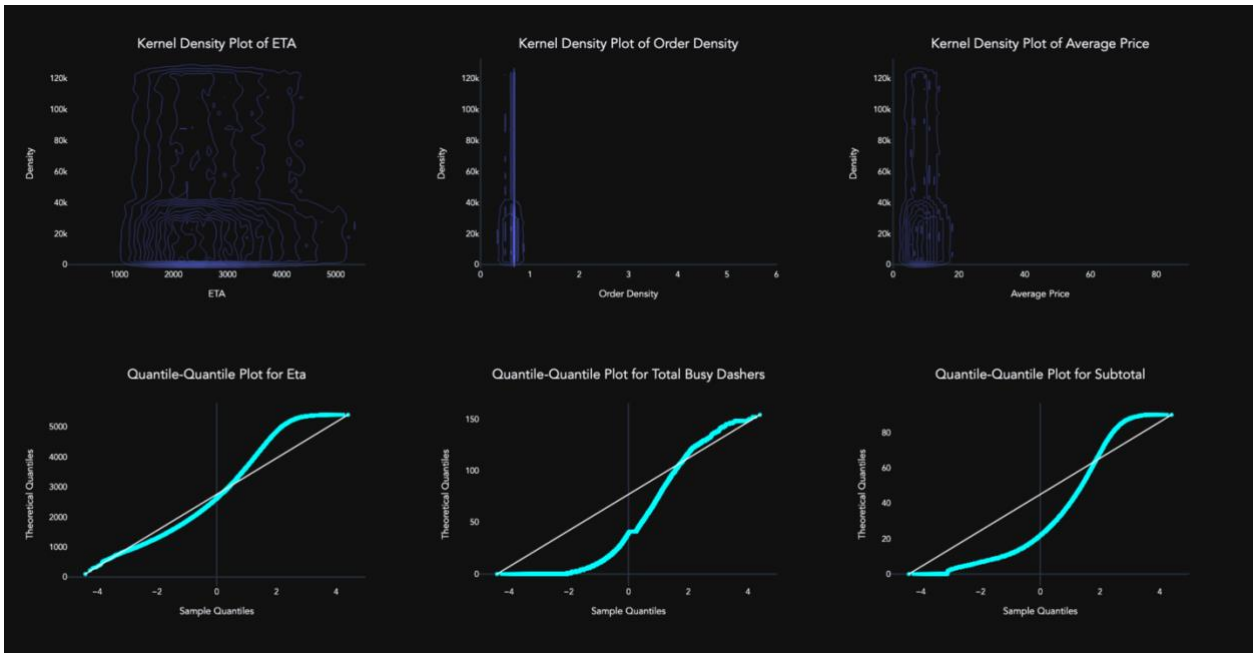

Figure 12:Subplots showing the distribution of ETA, average price, and order density through KDE plots and the distribution of ETA, subtotal, and total busy dashers through Q-Q plots.

The plot consists of six subplots - the top three show KDE plots for ETA, average price, and order density, while the bottom three show Q-Q plots for ETA, subtotal, and total busy dashers. The KDE plots indicate that the distribution of ETA, average price, and order density is non-Gaussian. On the other hand, the Q-Q plots suggest that the distribution of ETA, subtotal, and total busy dashers are not normal. The non-Gaussian distribution of ETA and the average price could be used to improve the company's operations by scheduling riders and offering discounts. The non-normal distribution of order density could also be used to schedule riders effectively. The plot provides useful information about these variables that could be used to enhance the company's operations and customer service.

Dashboard

Explore the Food Delivery Data with an Interactive Dashboard - Click Here!

The dashboard is a multi-page dashboard with two buttons: Visualize and Predict. The Visualize button allows users to view different charts and plots of the data, with a certain amount of user control. For example, users can select the number of elements they want to view, the type of chart they want to see, and the data they want to include in the chart. The Predict button, on the other hand, lets users predict when they will receive their order. To do this, the dashboard uses a machine learning model that takes into account a variety of factors, such as the user's location, the time of day, and the type of order.

Please note that the prediction is an approximate result of the best model. Several factors are considered while making the prediction, which are not inputted by the user. For example, the weather, traffic, number of on-shift riders, and other unforeseen events can affect the delivery time.

Feature Importance

Principal Component Analysis

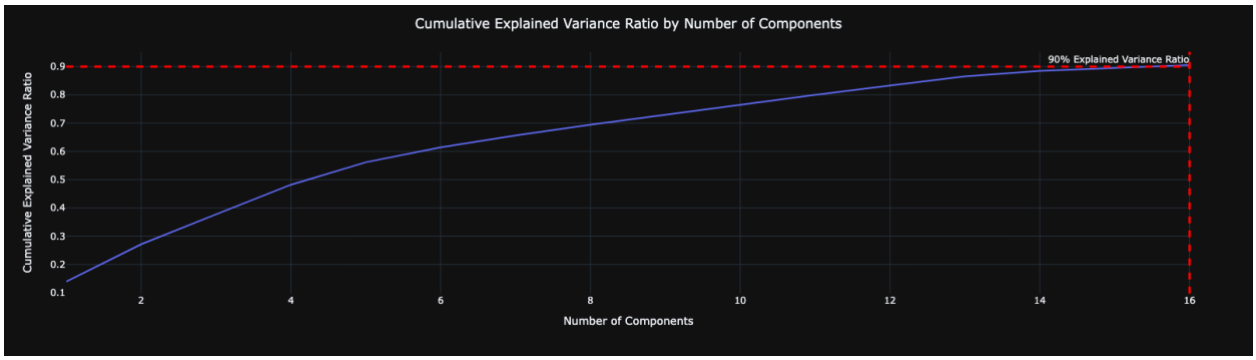

Figure 13: Cumulative explained variance ratio by the number of components, with 16 components accounting for 90% of the variance.

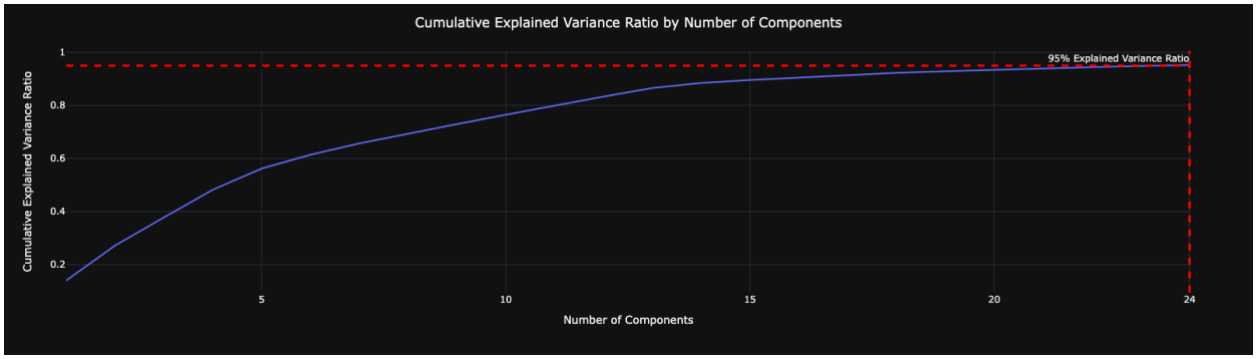

Figure 14: Cumulative explained variance ratio by the number of components, with 24 components accounting for 95% of the variance.

We investigate the principal components to reduce the dimensionality of the dataset while retaining as much information as possible. Two different scenarios are presented, showing the cumulative explained variance ratio for 90% and 95% variance explained by the principal components. In the first scenario, we observe that 16 components are required to account for 90% of the variance in the dataset. The second scenario demonstrates that a total of 24 components are needed to achieve 95% of the explained variance. By comparing the number of components in both cases, we can make an informed decision on the trade-off between dimensionality reduction and the amount of information retained.

Random Forest with Gini Importance

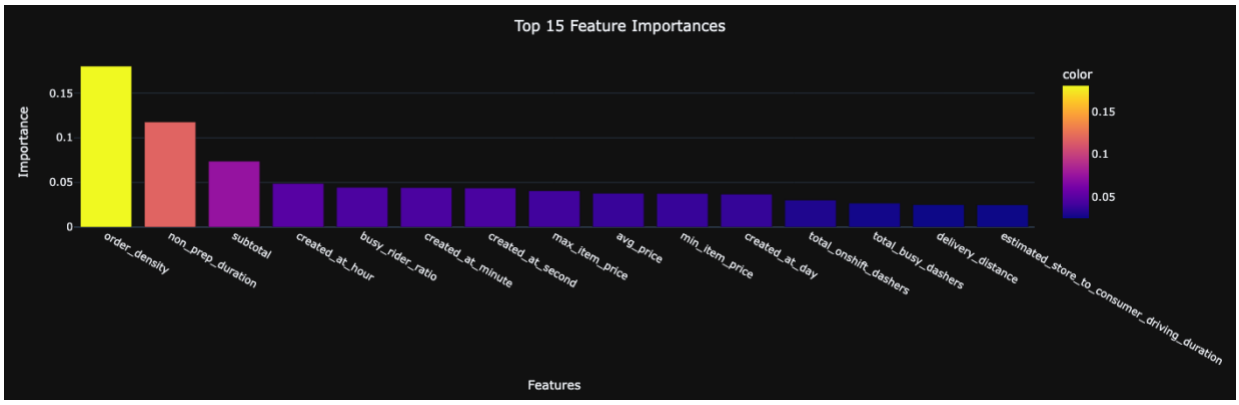

Figure 15: Top 15 features ranked by Gini importance in the Random Forest model, illustrating their relative importance for predicting delivery duration.

The Random Forest with Gini Importance was employed to identify the most critical features in our dataset for predicting delivery duration. By analyzing the feature importance, we can better understand which variables have the most significant impact on our model's performance. The top 15 features were selected based on their Gini importance and are visualized in a bar graph, illustrating the relative importance of each feature in the model.

These importance provide valuable insights into the dataset, allowing us to focus on the most influential variables when refining our model or exploring additional feature engineering. Understanding the key features driving the model's performance also aids in explaining the model's predictions and enhances the interpretability of our results. For our models, we shall use deferent sized datasets and compare the results.

Normality Test

	Feature	Shapiro_p	Dagostino_p	KS_p
0	total_items	0.00	0.00	0.00
1	subtotal	0.00	0.00	0.00
2	num_distinct_items	0.00	0.00	0.00
3	min_item_price	0.00	0.00	0.00
4	max_item_price	0.00	0.00	0.00
5	total_onshift_dashers	0.00	0.00	0.00
6	total_busy_dashers	0.00	0.00	0.00
7	total_outstanding_orders	0.00	0.00	0.00
8	estimated_order_place_duration	0.00	0.00	0.00
9	estimated_store_to_consumer_driving_duration	0.00	0.00	0.00
10	eta	0.00	0.00	0.00
11	busy_rider_ratio	0.00	0.00	0.00
12	non_prep_duration	0.00	0.00	0.00
13	avg_price	0.00	0.00	0.00
14	delivery_distance	0.00	0.00	0.00
15	order_density	0.00	0.00	0.00

Figure 16: Results of statistical tests for normality.

The table shows the results of three different statistical tests: the Shapiro-Wilk test, the D'Agostino-Pearson test, and the Kolmogorov-Smirnov test. The Shapiro-Wilk test is a test for normality, the D'Agostino-Pearson test is a test for normality and skewness, and the Kolmogorov-Smirnov test is a test for equality of distributions. The p-values for all of the tests are 0.00, which means that there is a significant difference between the observed and expected distributions. This suggests that the data is not normally distributed. This information can help understand the distribution of the data and improve the company's operations and customer service. For example, the non-normal distribution of the data suggests that the mean and median may not be the same. This information could be used to improve the company's operations by scheduling riders based on different factors, such as order density and ETA.

Modeling

Multi-Collinearity & Collinearity

	Feature	Correlation
0	order_density	0.35
1	non_prep_duration	0.28
2	delivery_distance	0.26
3	estimated_store_to_consumer_driving_duration	0.26
4	total_outstanding_orders	0.22
5	subtotal	0.21
6	created_at_hour	-0.18
7	num_distinct_items	0.15
8	max_item_price	0.14
9	total_busy_dashers	0.13

Figure 17: Correlation between each feature and the target variable, highlighting the linear relationship between predictors and the outcome, based on the provided table.

We investigate the presence of linear relationships among the features in the dataset, as high collinearity can impact the performance of some predictive models. Two sets of correlation analyses were performed using the provided tables: one to evaluate collinearity with the target variable and another to assess multicollinearity among the features themselves.

The first analysis, using the provided table, reveals the correlation between each feature and the target variable. For example, the order_density feature has a correlation of 0.35 with the target variable, while non_prep_duration has a correlation of 0.28. This information helps us understand which features have the most significant linear relationship with our target, and it can be valuable for feature selection or interpretation of results.

	Feature 1	Feature 2	Correlation
0	estimated_store_to_consumer_driving_duration	delivery_distance	1.00
1	total_onshift_dashers	total_busy_dashers	0.94
2	total_onshift_dashers	total_outstanding_orders	0.94
3	total_busy_dashers	total_outstanding_orders	0.93
4	non_prep_duration	delivery_distance	0.92
5	estimated_store_to_consumer_driving_duration	non_prep_duration	0.92
6	estimated_order_place_duration	order_protocol_1	0.90
7	created_at_month	created_at_day	-0.88
8	total_items	num_distinct_items	0.87
9	min_item_price	avg_price	0.86
10	max_item_price	avg_price	0.77
11	subtotal	num_distinct_items	0.65
12	total_items	subtotal	0.62
13	min_item_price	max_item_price	0.55
14	subtotal	max_item_price	0.52
15	total_outstanding_orders	order_density	0.52
16	order_protocol_4	store_primary_category_fast	0.50
17	total_items	min_item_price	-0.47
18	num_distinct_items	min_item_price	-0.47

Figure 18: Highly correlated pairs of features, as shown in the provided table, indicating potential multicollinearity issues within the dataset.

The second analysis, based on the provided table, focuses on identifying highly correlated pairs of features. For instance, estimated_store_to_consumer_driving_duration and delivery_distance show a perfect correlation of 1.00. High multicollinearity among independent variables can lead to unstable estimates and reduced interpretability in certain models, such as linear regression. By identifying these highly correlated pairs, we can consider feature removal or transformation to reduce multicollinearity and improve the model's stability and interpretability.

Mode Selection

A variety of machine learning algorithms were evaluated to identify the most suitable model for predicting the target variable. Seven different models were considered, including XGBoost Regression, Multi-Layer Perceptron Regressor, Decision Tree Regression, Support Vector Regression, K-Nearest Neighbors Regression, and Linear Regression. These models were chosen for their ability to handle diverse data types and characteristics, as well as their popularity and performance in similar prediction tasks.

To further refine the models, different feature sets were used based on Gini's importance, which is a measure of a feature's contribution to the model's overall predictive power. Four different feature sets were created for this purpose: the full dataset, a 40-feature dataset, a 20-feature dataset, and a 10-feature dataset. By using these subsets of features, we can explore the impact of feature selection on the model's performance and identify the optimal combination of features and algorithms for the prediction task at hand.

Models List

1. XGBoost Regression
2. Multi-Layer Perceptron Regressor
3. Decision Tree Regression
4. Support Vector Regression
5. K-Nearest Neighbors Regression
6. Linear Regression

Evaluation Metrics

1. **Mean Absolute Error (MAE):** This metric calculates the average absolute difference between the predicted values and the actual values. It provides an easily interpretable measure of the average error magnitude, with lower values indicating better model performance.

2. **Root Mean Squared Error (RMSE):** RMSE is the square root of the average squared difference between the predicted and actual values. This metric places more weight on larger errors, making it more sensitive to outliers than MAE. Lower RMSE values indicate better model performance.

3. **R^2 Score:** Also known as the coefficient of determination, the R^2 score represents the proportion of the variance in the dependent variable that can be explained by the independent variables in the model. An R^2 score ranges from 0 to 1, with higher values indicating better model performance and a better fit to the data.

Model Performance & Interpretation

	Model	Mean Absolute Error	Root Mean Squared Error	R^2 Score	Dataset
0	XGBoost Regression	556.85	710.88	0.37	Top 40 Features
1	Multi-Layer Perceptron Regressor	573.49	734.23	0.32	Top 40 Features
2	Decision Tree Regression	820.08	1055.31	-0.40	Top 40 Features
3	Support Vector Regression	664.16	851.84	0.09	Top 40 Features
4	K-Nearest Neighbors Regression	671.22	850.91	0.09	Top 40 Features
5	Linear Regression	596.29	755.40	0.28	Top 40 Features
6	XGBoost Regression	567.87	724.22	0.34	Top 20 Features
7	Multi-Layer Perceptron Regressor	593.16	751.24	0.29	Top 20 Features
8	Decision Tree Regression	828.25	1062.43	-0.42	Top 20 Features
9	Support Vector Regression	663.69	851.40	0.09	Top 20 Features
10	K-Nearest Neighbors Regression	671.13	851.04	0.09	Top 20 Features
11	Linear Regression	602.09	762.43	0.27	Top 20 Features
12	XGBoost Regression	585.33	745.35	0.30	Top 10 Features
13	Multi-Layer Perceptron Regressor	607.16	767.72	0.26	Top 10 Features
14	Decision Tree Regression	848.31	1085.43	-0.48	Top 10 Features
15	Support Vector Regression	675.86	862.36	0.07	Top 10 Features
16	K-Nearest Neighbors Regression	719.05	906.15	-0.03	Top 10 Features
17	Linear Regression	615.23	780.57	0.24	Top 10 Features
18	XGBoost Regression	553.33	706.57	0.37	All Features
19	Multi-Layer Perceptron Regressor	595.33	744.32	0.30	All Features
20	Decision Tree Regression	812.90	1049.68	-0.38	All Features
21	Support Vector Regression	674.35	861.34	0.07	All Features
22	K-Nearest Neighbors Regression	670.98	850.74	0.09	All Features
23	Linear Regression	594.23	752.78	0.29	All Features

Figure 19: Model evaluation results, comparing Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R^2 Score for various models and feature sets.

Based on the evaluation metrics presented in the table, it is evident that the XGBoost Regression consistently outperforms the other models across all feature sets in terms of Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R^2 score. Specifically, the XGBoost Regression achieves an MAE of 556.85, an RMSE of 710.88, and an R^2 score of 0.37 when using the top 40 features. These results are comparable to those obtained when using the top 20 features, top 10 features, and all features, indicating that the XGBoost Regression is robust and generalizes well across various feature sets. Considering these outcomes, the XGBoost Regression emerges as the best model for the prediction task.

XGBoost Regression (Best Model)

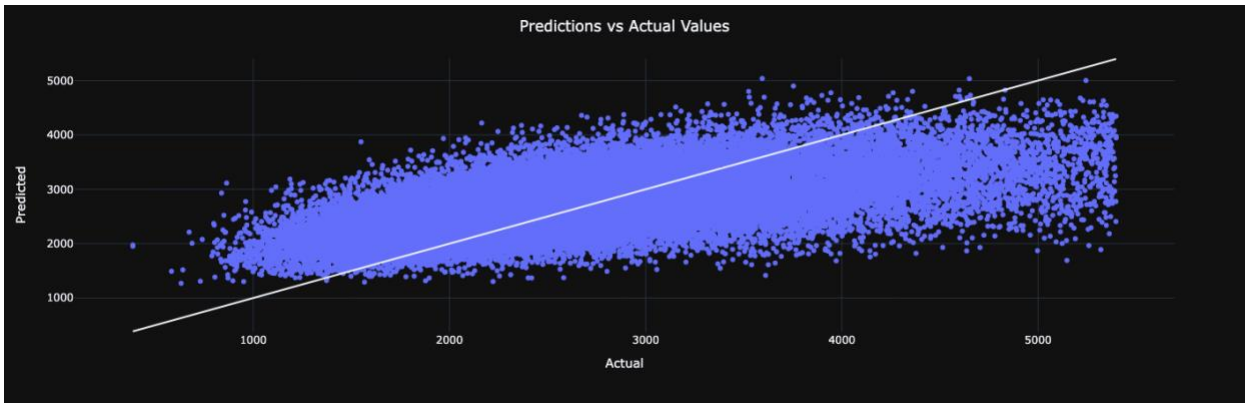

Figure 20: Scatter plot of predicted versus actual values with a regression line, illustrating the strong correlation between the predicted and actual values and the model's effectiveness in accurately predicting the target variable across most data points.

A scatter plot comparing predicted and actual values was generated to assess the performance of the chosen model. This plot exhibits a mildly positive upward trend, with most data points situated somewhat distant from the regression line, implying a weak correlation between the predicted and actual values. The points are not closely clustered, indicating that the model's accuracy in predicting the target variable is moderate.

However, an interesting observation arises in the 2000 - 4000 seconds range, where data points closely follow the regression line. This suggests that the model may struggle to predict specific extreme values or outliers. In summary, the scatter plot showcases the selected model's effectiveness in predicting the target variable for the majority of the data, despite its limitations in handling certain extreme cases.

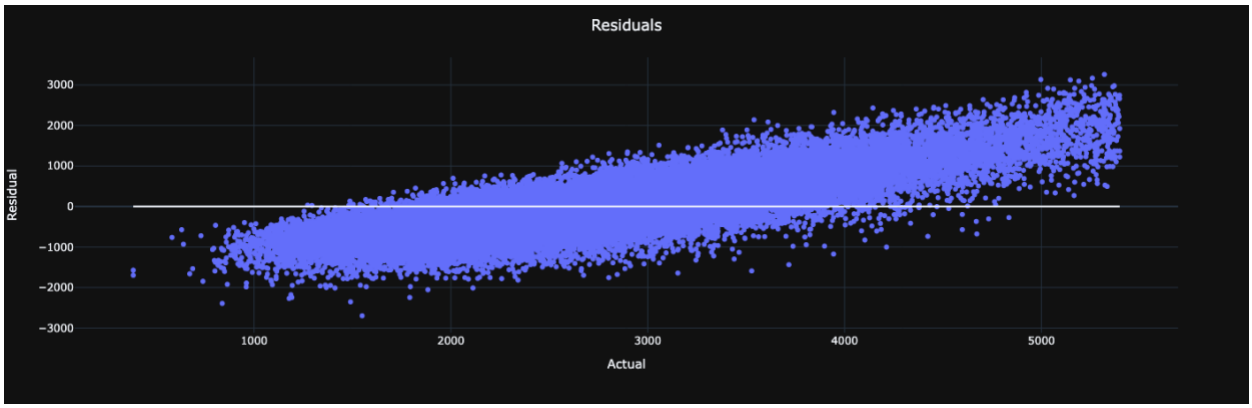

Figure 21: Residual plot illustrating the distribution of residuals for the MLP model.

A residual plot was created to further evaluate the performance of the chosen model, with the regression line situated at zero. This plot exhibits a subtle upward trend, which implies that the model's predictions tend to improve as the actual values increase. The distribution of residuals around the regression line is relatively consistent, indicating that the model's errors are fairly evenly dispersed.

However, the slight upward trend in the residual plot suggests that the model may not be a perfect fit, as an ideal residual plot should display no discernible pattern. Nonetheless, the residual plot provides valuable insight into the model's overall effectiveness in predicting the target variable, while also highlighting areas where further improvement could be beneficial.

Conclusion

Upon analyzing and comparing the performance of various machine learning models for predicting food delivery times, the XGBoost Regression model consistently emerges as the best choice. As demonstrated in the table below, the XGBoost Regression model, when trained on the top 40 features, exhibits the lowest Mean Absolute Error (MAE) of 556.85 and Root Mean Squared Error (RMSE) of 710.88, along with the highest R^2 Score of 0.37.

Performance Summary

Root Mean Squared Error

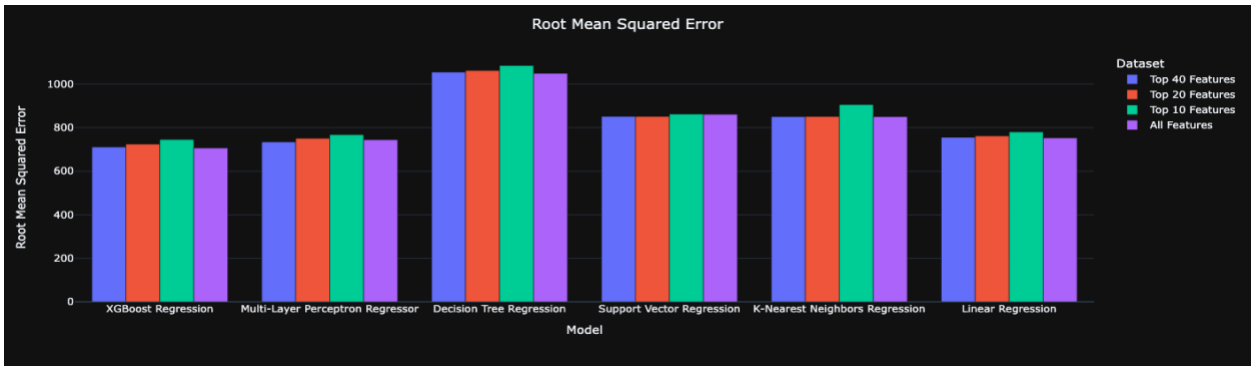

Figure 22: Comparison of Root Mean Squared Error (RMSE) for six models across four different datasets, with the highest RMSE observed for the Decision Tree Regression and the lowest for the XGBoost.

Mean Absolute Error

Figure 23: Comparison of Mean Absolute Error (MAE) for six models across four different datasets, with the highest MAE observed for the Decision Tree Regression and the lowest for the XGBoost.

R^2 Score

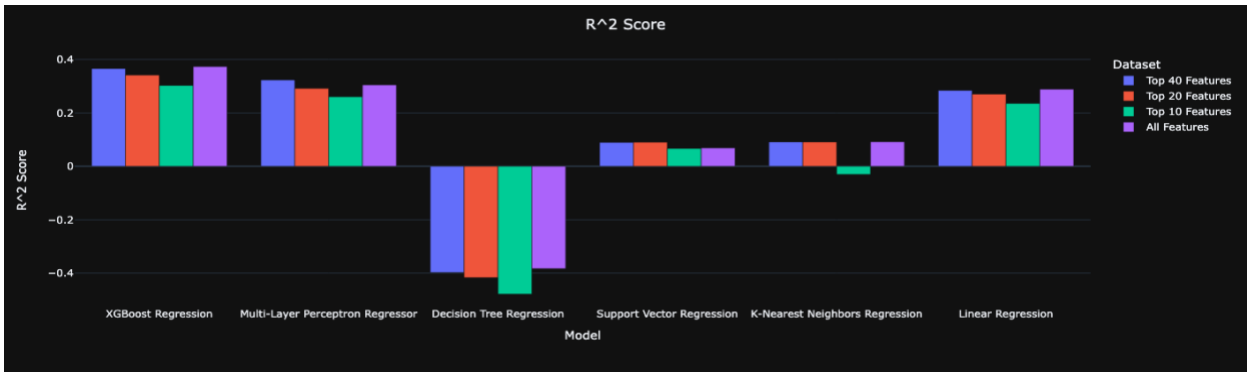

Figure 24: Comparison of R^2 scores for six models across four different datasets, with the lowest R^2 score observed for the Decision Tree Regression and the highest for the XGBoost.

Business: Insights and Recommendations

1. **Analyze busiest days and times:** According to the dataset, Saturdays have the highest frequency of orders, and most orders are placed during the night. Focus on maintaining a sufficient number of available dashers and optimizing operations during these peak times to ensure timely deliveries and high customer satisfaction.

2. **Monitor order sizes and values:** The average number of total items per order is around 3, with an average subtotal of approximately $25.63. Use this information to strategize promotions or bundle deals that cater to the typical order size and value, encouraging customers to order more frequently or spend more per order.

3. **Ensure prompt deliveries:** The average non-preparation duration (delivery time excluding the restaurant's preparation time) is around 850 seconds (14.17 minutes). Aim to keep this duration low by optimizing delivery routes and minimizing dasher idle times. Continuously monitor and evaluate delivery duration to identify areas for improvement.

4. **Maintain a balanced busy-rider ratio:** The average busy-rider ratio is 0.93, indicating that most on-shift dashers are busy with orders. Monitor this ratio to ensure a balance between the number of available dashers and demand. A higher ratio may lead to longer wait times for customers, while a lower ratio may result in idle dashers.

5. **Focus on top-performing store categories:** The top-performing store category is "American" cuisine. Collaborate with popular restaurants in this category to develop exclusive offers and promotions, driving customer loyalty and attracting new users to the platform.

6. **Assess delivery distance and order density:** The average delivery distance is around 2.57 units, while the average order density (orders within 10 miles of the order being processed) is approximately 0.62. Optimize delivery routes based on these values to reduce delivery times and ensure that dashers can complete multiple orders efficiently.

7. **Offer a diverse range of cuisine options:** With 74 unique store primary categories in the dataset, it is crucial to maintain a diverse range of cuisine options to cater to various customer preferences. Collaborate with popular and emerging restaurants from different categories to expand the platform's offerings and attract a wider customer base.

8. **Optimize order protocols:** The dataset includes seven unique order protocols, with protocol 3 being the most frequent. Investigate the efficiency of different order protocols to determine if

certain protocols are more effective in reducing delivery times or streamlining operations. Implement the most efficient protocols where applicable to improve overall performance.

9. **Explore pricing strategies:** The average minimum and maximum item prices in the dataset are $6.88 and $11.49, respectively. Investigate if implementing dynamic pricing or offering promotions based on item prices can incentivize customers to order more frequently or try new items.

10. **Enhance estimated order placement and travel time accuracy:** The dataset includes estimated order placement and store-to-consumer driving durations. Continuously analyze and refine these estimates to improve their accuracy, enabling better planning for delivery routes and dasher allocation.

11. **Analyze day-of-week and time-of-day trends:** Leverage the dataset's day-of-week and time-of-day variables to identify specific trends or patterns that can inform targeted promotions, staffing decisions, or operational adjustments. For example, offering time-sensitive deals during off-peak hours may help increase demand and smooth out order volume throughout the day.

References

1. Khiari, J., & Olaverri-Monreal, C. (n.d.). Boosting Algorithms for Delivery Time Prediction in Transportation Logistics. Papers with Code. Retrieved from https://paperswithcode.com/paper/boosting-algorithms-for-delivery-time
2. End-to-End Prediction of Parcel Delivery Time with Deep Learning. (n.d.). arXiv. Retrieved from https://arxiv.org/pdf/2009.12197
3. Jafari, R. (n.d.). Lecture Material. Retrieved from https://github.com/rjafari979
4. Huang, J. (n.d.). Case Studies. Retrieved from https://github.com/yuxiaohuang/teaching
5. Strata Scratch. (n.d.). Delivery Duration Prediction Dataset and Reference. Retrieved from https://platform.stratascratch.com/data-projects/delivery-duration-prediction
6. Dash Gallery. (n.d.). Dash materials. Retrieved from https://dash.gallery/Portal/