Order Delivery Time Prediction

Porter Services

- Suyash Nagar

Objectives

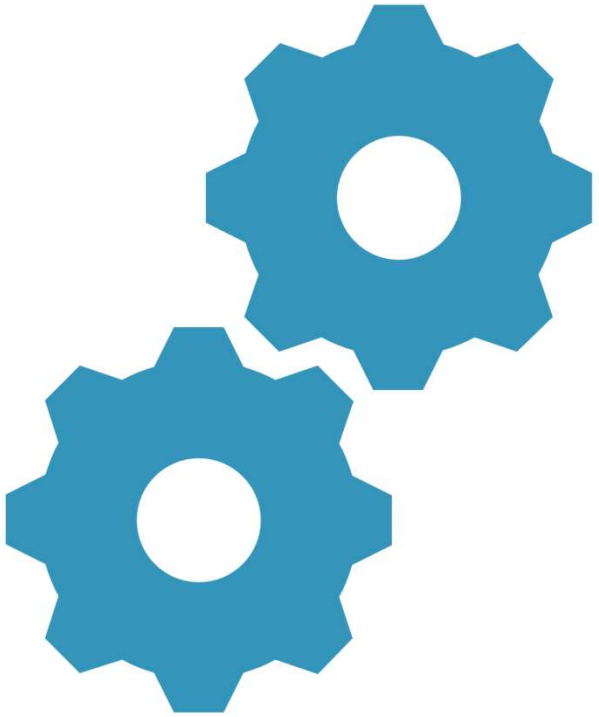
▶The objective of this assignment is to build a regression model that predicts the delivery time for orders placed through Porter. The model will use various features such as the items ordered, the restaurant location, the order protocol, and the availability of delivery partners.

The key goals are:

▶- Predict the delivery time for an order based on multiple input features

▶- Improve delivery time predictions to optimize operational efficiency

▶- Understand the key factors influencing delivery time to enhance the model's accuracy



Data Pipeline

DATA LOADING

DATA PREPROCESSING

FEATURE ENGINEERING

EXPLORATORY

MODEL BUILDING

MODEL INFERENCE

1. Data Loading

Data Understanding

Field	Description
market_id	Integer ID representing the market where the restaurant is located.
created_at	Timestamp when the order was placed.
actual_delivery_time	Timestamp when the order was
store_primary_category	Category of the restaurant (e.g., fast food, dine-in).
order_protocol	Integer representing how the order was placed (e.g., via Porter, call to restaurant, etc.).
total_items	Total number of items in the order.
subtotal	Final price of the order.
num_distinct_items	Number of distinct items in the order.
min_item_	Price of the cheapest item in the order.
max_item_price	Price of the most expensive item in the order.
total_onshift_dashers	Number of delivery partners on duty when the order was placed.
total_busy_dashers	Number of delivery partners already occupied with other orders.
total_outstanding_orders	Number of orders pending fulfillment at the time of the order.
distance	Total distance from the restaurant to the customer.

Necessary Libraries

Data Manipulation & Analysis

- pandas: Used for data manipulation, handling DataFrames, and processing structured datasets.
- numpy: Provides support for numerical computations and array processing.
- sklearn.feature_selection.RFE: Helps with Recursive Feature Elimination for feature selection.

Data Visualization

- matplotlib.pyplot: A fundamental plotting library for visualizing data.
- seaborn: Built on top of matplotlib, offering beautiful statistical visualizations.

Machine Learning & Model Building:

- sklearn.model_selection.train_test_split: Splits dataset into training and testing sets.
- sklearn.preprocessing.MinMaxScaler: Normalizes data to a specified range.
- sklearn.linear_model.LinearRegression: Implements linear regression models.
- sklearn.feature_selection.RFE: Used for feature selection in machine learning models.

Model Evaluation & Performance Metrics:

- sklearn.metrics.mean_absolute_error, mean_squared_error, r2_score: Evaluate regression model performance using various metrics.

Statistical Analysis:

- statsmodels.api: Performs advanced statistical modeling.
- statsmodels.stats.outliers_influence.variance_inflation_factor: Calculates VIF to detect multicollinearity in regression models.

Suppressing Warnings

- warnings.filterwarnings("ignore"): Suppresses unnecessary warnings to clean output

Data View & Observations

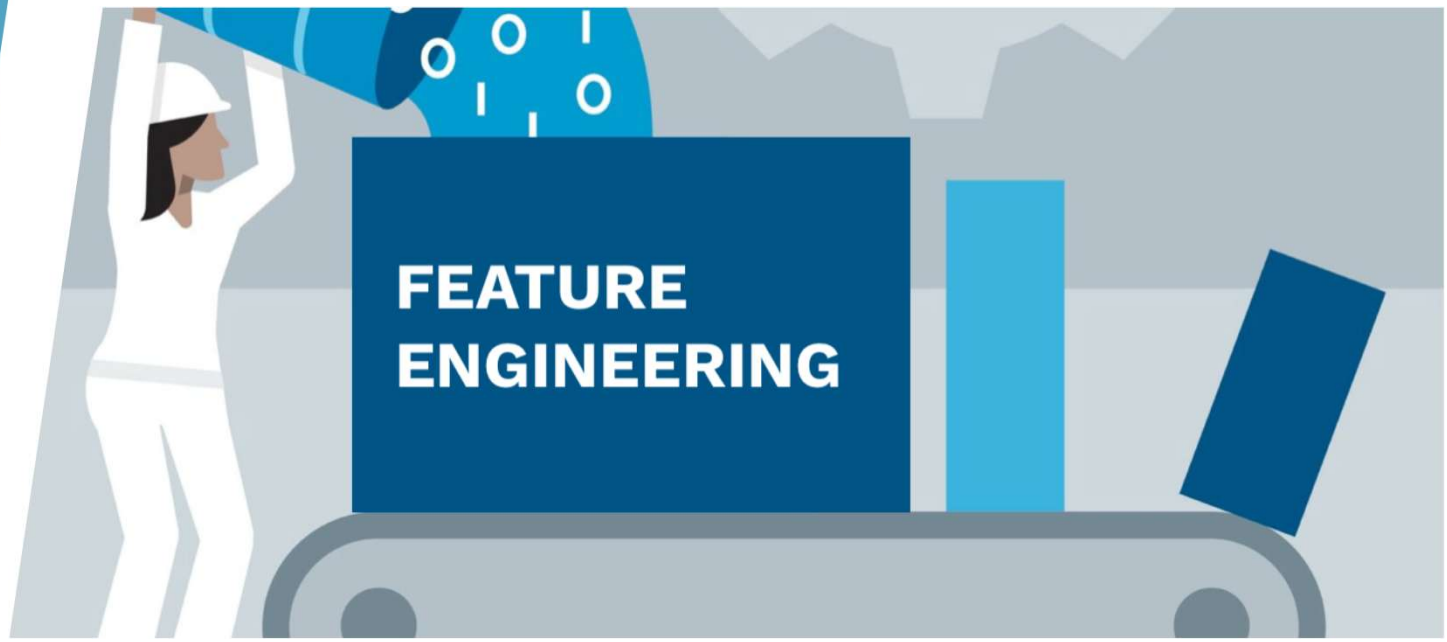
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 175777 entries, 0 to 175776
Data columns (total 14 columns):
 #   Column                    Non-Null Count   Dtype
---  ------                    --------------   -----
 0   market_id                 175777 non-null  float64
 1   created_at                175777 non-null  object
 2   actual_delivery_time      175777 non-null  object
 3   store_primary_category    175777 non-null  int64
 4   order_protocol            175777 non-null  float64
 5   total_items               175777 non-null  int64
 6   subtotal                  175777 non-null  int64
 7   num_distinct_items        175777 non-null  int64
 8   min_item_price            175777 non-null  int64
 9   max_item_price            175777 non-null  int64
 10  total_onshift_dashers     175777 non-null  float64
 11  total_busy_dashers        175777 non-null  float64
 12  total_outstanding_orders  175777 non-null  float64
 13  distance                  175777 non-null  float64
dtypes: float64(6), int64(6), object(2)
memory usage: 18.8+ MB
Info of dataset                      : None
```

Observations: Knowing Data

1. No Null, Missing or duplicate present
2. The dataset `porter_df` contains **175,777 rows & 14 columns**
3. Categorical features include **market_id, store_primary_category, & order_protocol**

To-do :

1. Date Features present as object
2. Categorical Features present as object
3. Removed negative records for min_item_price, total_onshift_dashers , total_busy_dashers and total_outstanding_orders 95/175777 records



2. Data Preprocessing and Feature Engineering

Fixing the Datatypes

- Converted Date and time from object to datetime format
- Converted below categorical features to category type
  - market_id
  - order_protocol
  - store_primary_category
- Changed datatypes to int for below
  - total_onshift_dashers
  - total_busy_dashers
  - total_outstanding_orders

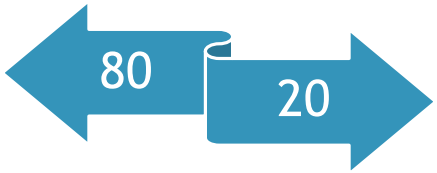
```
Data columns (total 14 columns):
 #   Column                  Non-Null Count   Dtype
---  ------                  --------------   -----
 0   market_id               175687 non-null  category
 1   created_at              175687 non-null  datetime64[ns]
 2   actual_delivery_time    175687 non-null  datetime64[ns]
 3   store_primary_category  175687 non-null  category
 4   order_protocol          175687 non-null  category
 5   total_items             175687 non-null  int64
 6   subtotal                175687 non-null  int64
 7   num_distinct_items      175687 non-null  int64
 8   min_item_price          175687 non-null  int64
 9   max_item_price          175687 non-null  int64
 10  total_onshift_dashers   175687 non-null  int32
 11  total_busy_dashers      175687 non-null  int32
 12  total_outstanding_orders 175687 non-null  int32
 13  distance                175687 non-null  float64
dtypes: category(3), datetime64[ns](2), float64(1), int32(3), int64(5)
memory usage: 14.6 MB
None
```

Feature Engineering

- Created below DateTime Features
  - ‘time_taken_mins’ Delivery time in mins
  - ‘hour’ delivery hour (0-23)
  - ‘day_of_week’ Delivery day in week (Monday to Sunday)
- Create new derived Features
  - ‘dashers_order_ratio’ Total available Dashers by Total outstanding order
  - ‘dashers_load_index’ Total outstanding order by Total onshift Dashers

```
 #   Column                  Non-Null Count   Dtype
---  ------                  --------------   -----
 0   market_id               175687 non-null  category
 1   store_primary_category  175687 non-null  category
 2   order_protocol          175687 non-null  category
 3   total_items             175687 non-null  int64
 4   subtotal                175687 non-null  int64
 5   num_distinct_items      175687 non-null  int64
 6   min_item_price          175687 non-null  int64
 7   max_item_price          175687 non-null  int64
 8   total_onshift_dashers   175687 non-null  int32
 9   total_busy_dashers      175687 non-null  int32
 10  total_outstanding_orders 175687 non-null  int32
 11  distance                175687 non-null  float64
 12  time_taken_mins         175687 non-null  float64
 13  hour                    175687 non-null  category
 14  day_of_week             175687 non-null  category
 15  isWeekend               175687 non-null  category
 16  dashers_order_ratio     175687 non-null  float64
 17  dashers_load_index      175687 non-null  float64
dtypes: category(6), float64(4), int32(3), int64(5)
memory usage: 16.4 MB
```

Split Train(80) and Test set(20) with random state




```
X Train shape: (140549, 17), y Train shape: (140549,)
X Test shape: (35138, 17), y Test shape: (35138,)
```

3. Exploratory Data Analysis on Training Data

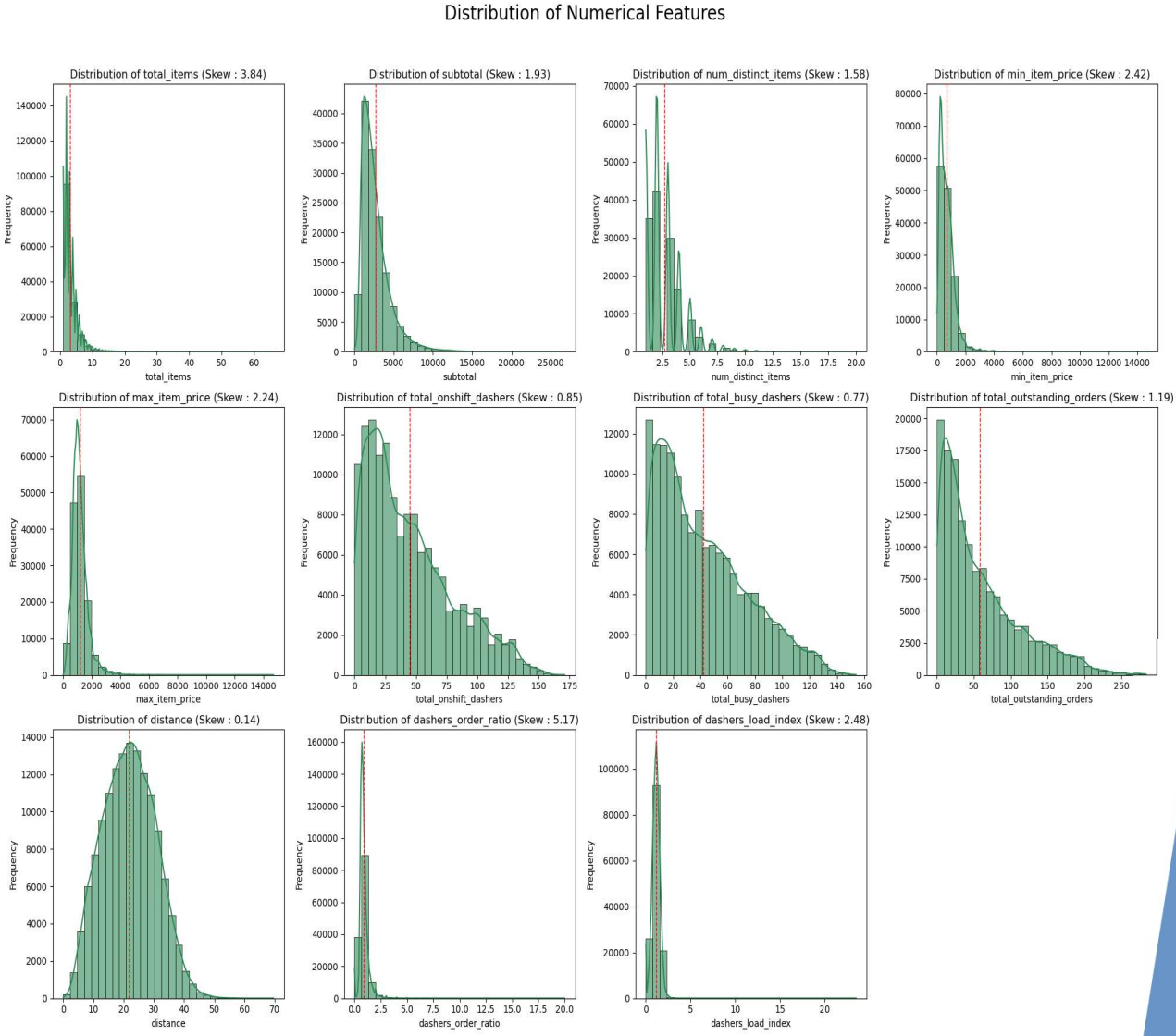
- Univariate Analysis
- Bivariate Analysis
- Multivariate Analysis

Feature Distributions - Numerical

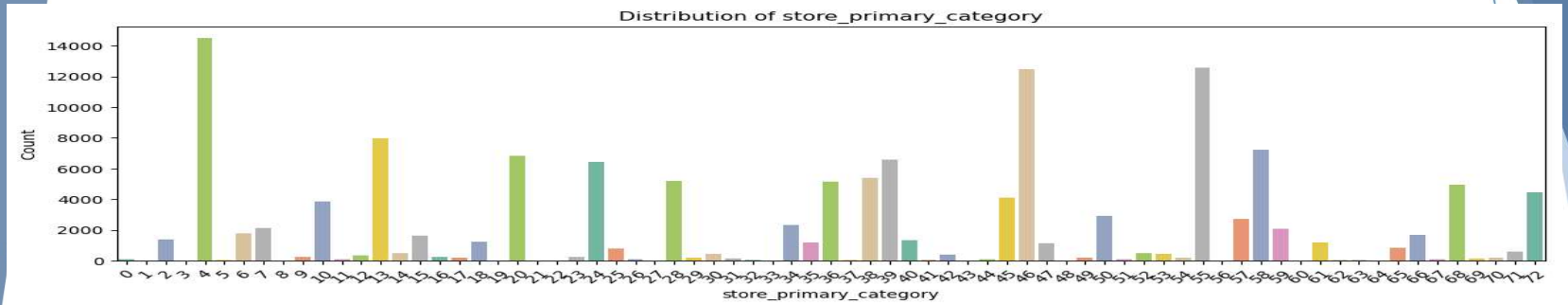
- Right-Skewed Distributions: The majority of features (total_items, subtotal, num_distinct_items, min_item_price, max_item_price, etc.) are right-skewed

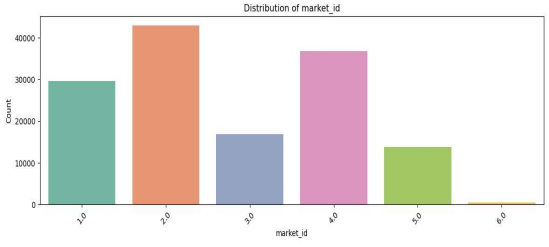
2. This indicates that while most observations are clustered at the lower end of the scale, there are considerable outliers stretching the distributions to higher values.

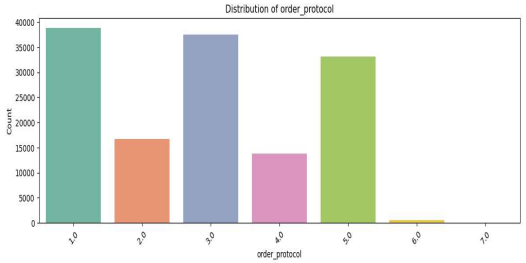
3. The distance feature stands out as nearly symmetric, implying that its values are uniformly spread around the average, which makes it suitable for models that assume normally distributed predictors without additional transformation.

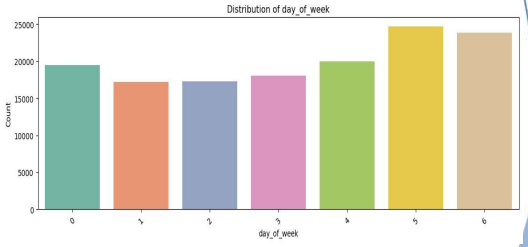


Feature Distributions - Categorical







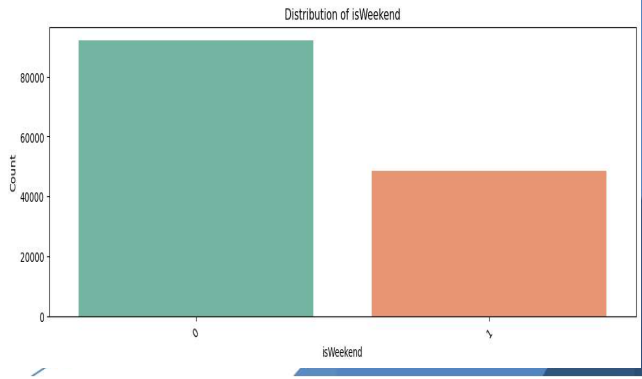


1. Market Id: Market Id 2 has the largest share followed 4

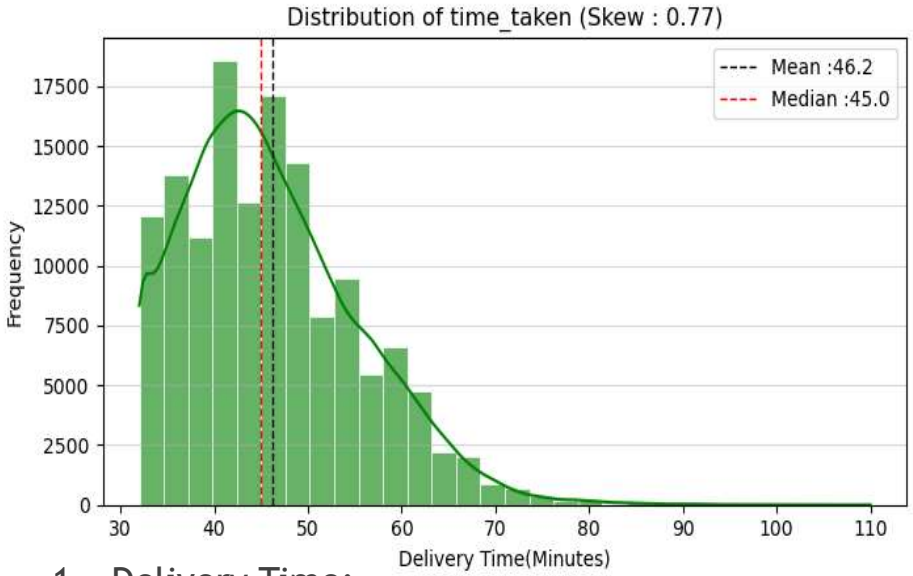
2. Store Primary Category: Category 4 is the most dominant with 14000+ counts followed by 55 & 46. Distribution varies signifiantly

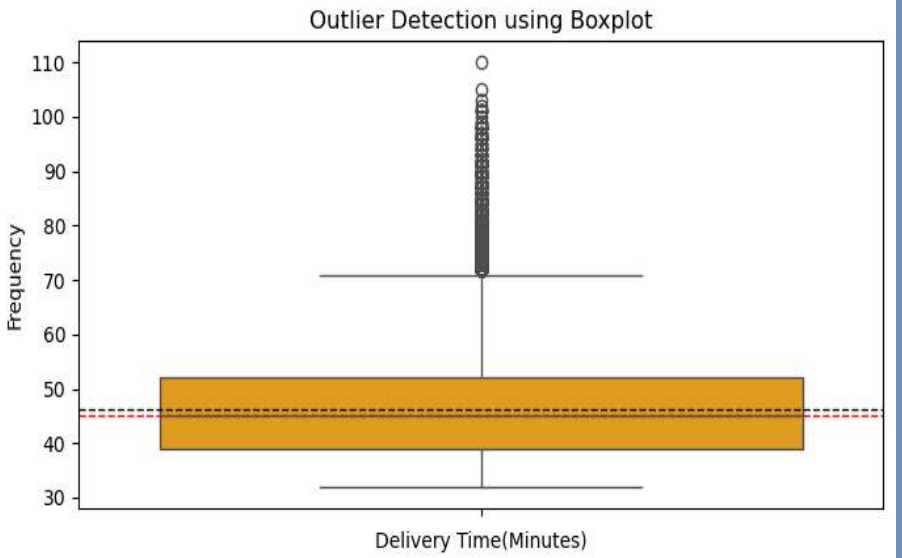
3. Order protocol: Protocol 1is the most used with close to 40K counts followed by protocol 2 and 3

4. Day of Week (Monday(0)-->Sunday(6))Weekend peak starting Friday followed by Saturday and Sunday, other days are steady



Feature Distributions – Target Variable





1. Delivery Time:

   Right Skewed more delivery with shorter time and few take much longer time

   Mean 46.2 and Median 45 are close, suggesting balanced distribution

   Box plot indicates existence of outliers above upper whiskers indicating longer delivers

Key take aways:

1. Most deliveries take around 45 minutes
2. Presence of outliers

Relationships Between Features

- `Total Items vs Delivery Time`: A slight upward trend, might suggest more items might lead to longer delivery time

- `Subtotal vs Delivery Time`: No strong correlation, indicating that higher value doesn't mean longer delivery time

- `Distinct Items vs. Delivery Time': A weak positive relationship, suggesting that orders with more variety of items could take longer to prepare and deliver.

- `Min Item Price vs. Delivery Time`: Minimal correlation, meaning cheaper or expensive items don't significantly affect delivery time.

- `Max Item Price vs. Delivery Time': Similar to the above—delivery times remain fairly consistent across different price ranges.

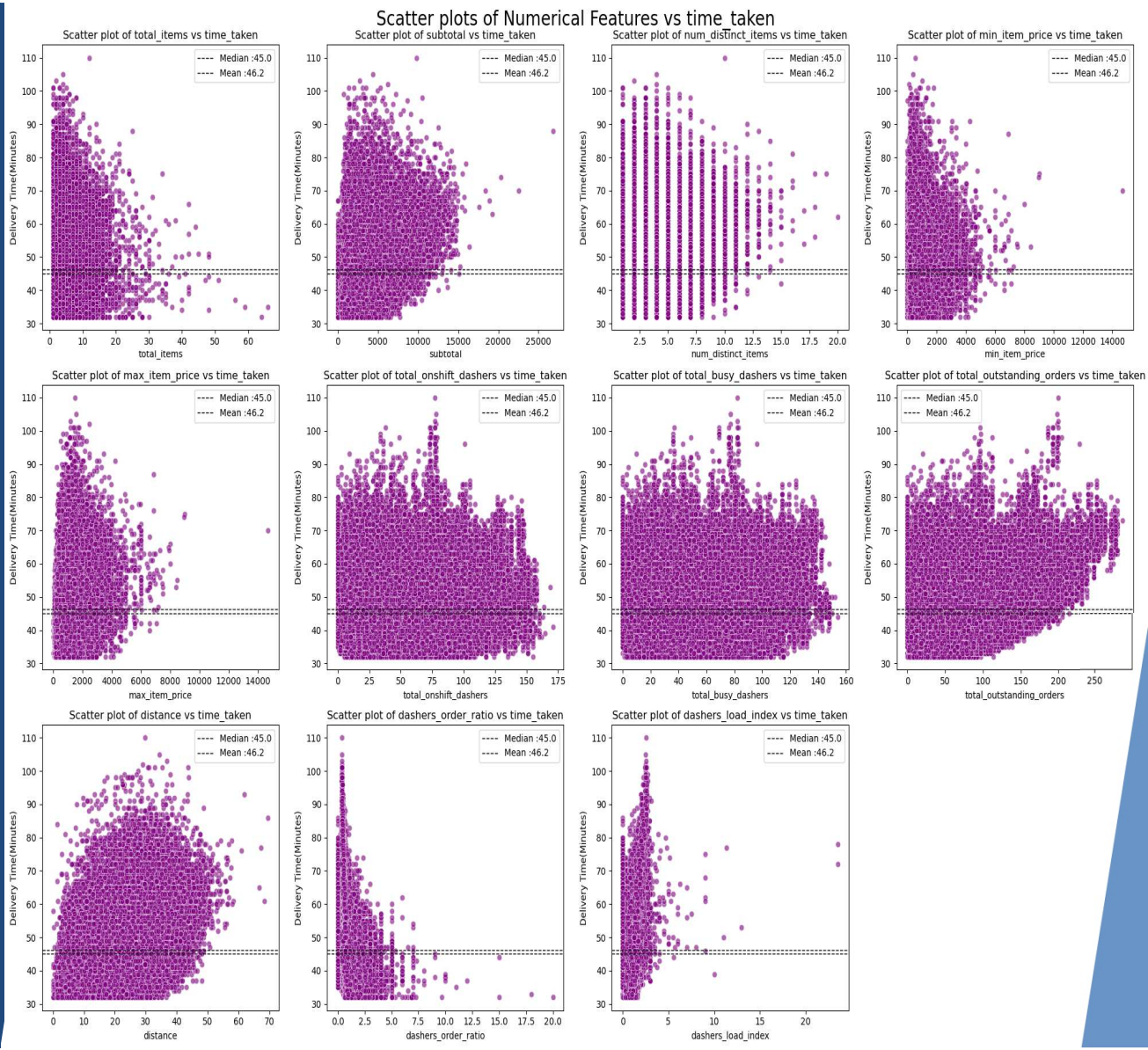
- `Total On shift Dashers vs. Delivery Time': More dashers appear to slightly reduce delivery times, confirming higher availability leads to efficiency.

- `Total Busy Dashers vs. Delivery Time`:Busier dashers correlate with higher delivery times, indicating congestion or peak-hour delays.

- `Outstanding Orders vs. Delivery Time': Strong correlation with delivery times—more pending orders cause delays, highlighting rush-hour pressure.

- `Distance vs. Delivery Time': Direct positive correlation—longer distances increase delivery time, as expected.

- `Hour of the Day vs. Delivery Time`:Peak-hour orders show higher delivery times, likely due to demand surges and traffic.

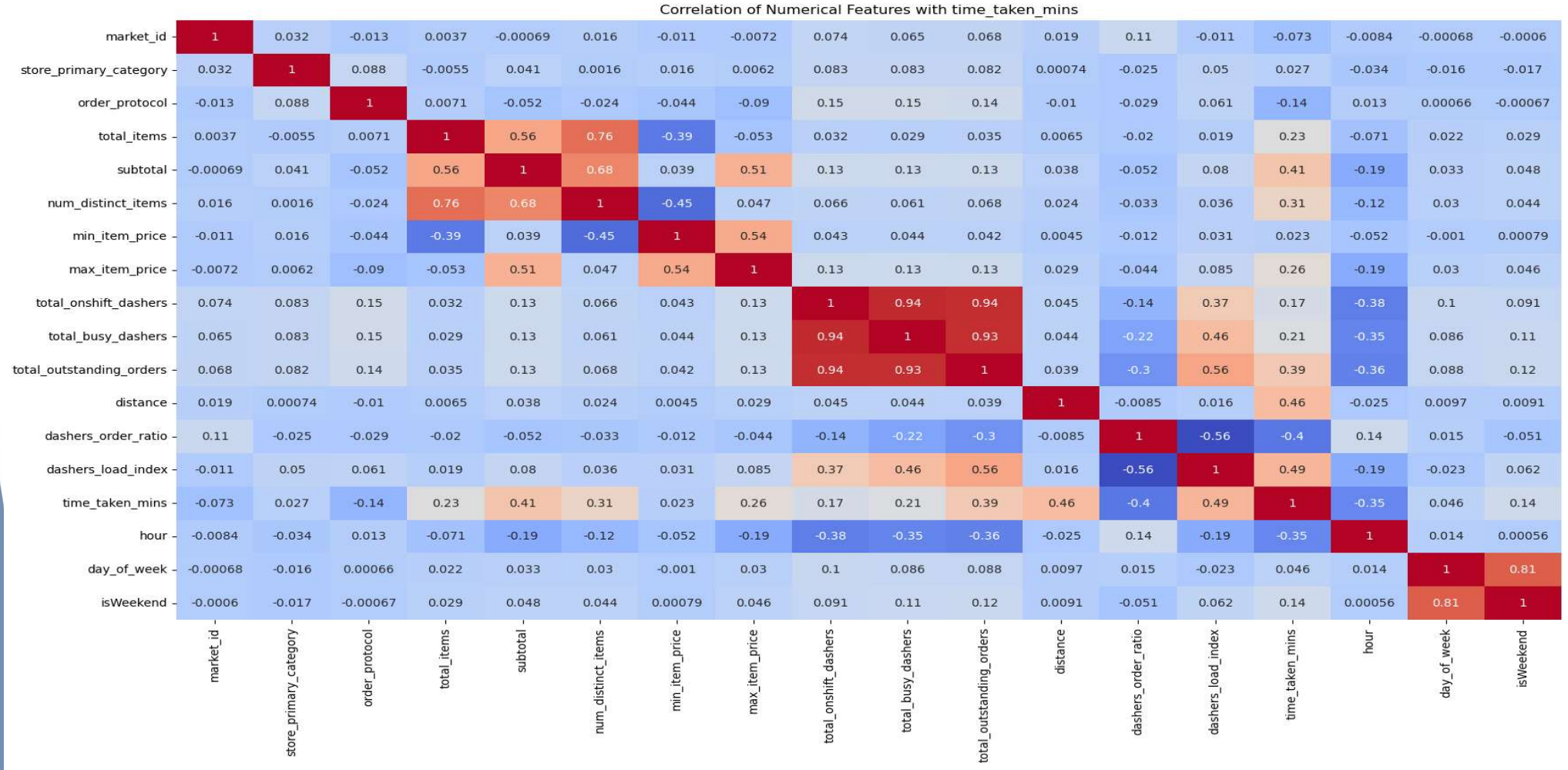


Distribution Time Taken (Hours)

1. Median Delivery Time: Ranges from **39 to 50**

2. Outliers: There are noticeable outliers in multiple hours, suggesting occasional extreme delivery times.

Display Correlation



Display Correlation - 2

Correlation Heatmap Key Highlights

Strongest Positive Correlations:

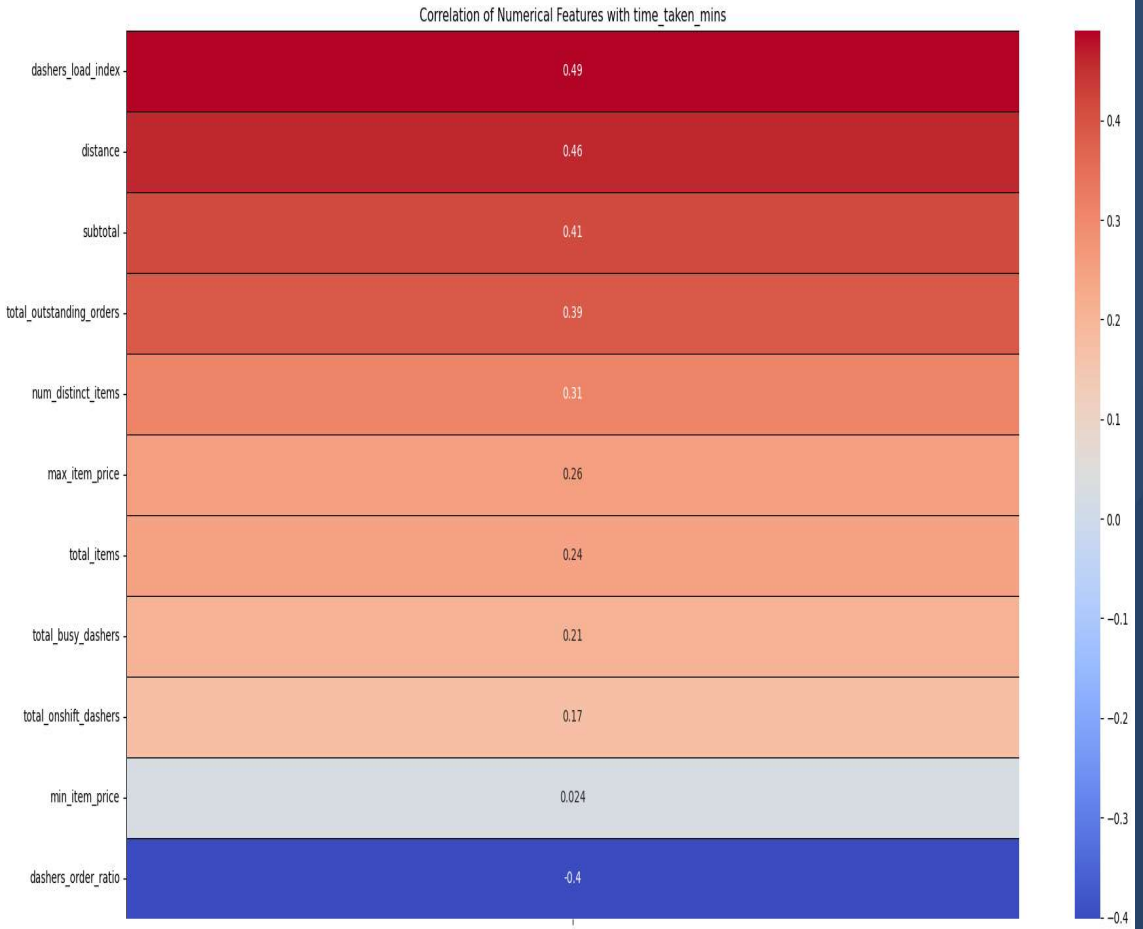
- **Dashers Load Index (0.49):** This is the highest among the features and suggests that when the dashers are under more load, the delivery time tends to increase. This factor may capture real-life constraints such as peak busy times or operational pressure.
- **Distance (0.46):** A higher distance unsurprisingly correlates with longer delivery times. The nearly linear relationship echoes that delivery time is largely influenced by how far the package has to travel.
- **Subtotal (0.41) and Total Outstanding Orders (0.39):** These features are moderately correlated with delivery time. A higher order value (subtotal) might indicate larger or more complex orders, while higher outstanding orders could imply operational bottlenecks that inadvertently slow down delivery.

Moderate to Weak Positive Correlations:

- **Number of Distinct Items (0.31), Max Item Price (0.26), Total Items (0.24), Total Busy Dashers (0.21), Total On-Shift Dashers (0.17):** These features show that while there is some relationship with delivery time, their influence is less pronounced compared to distance or load index. They might affect the delivery time in nuanced ways (e.g., the order complexity or product variety) and could interact with other factors.

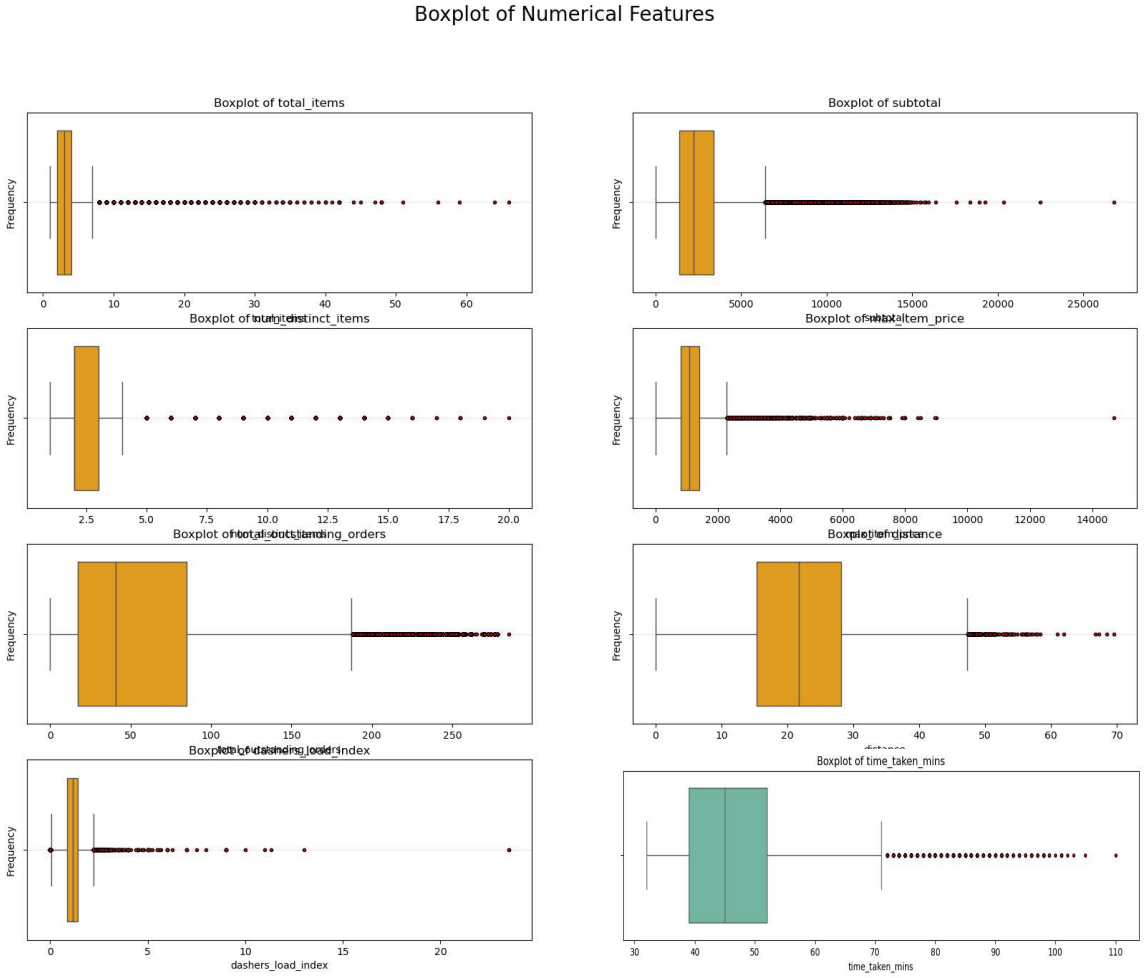
Negative Correlation:

- **Dashers Order Ratio (-0.4):** This negative correlation suggests that a higher dashers order ratio is associated with shorter delivery times. Perhaps when more dashers are assigned relative to orders, the system can handle orders more swiftly.




```
DROP Features : min_item_price,total_onshift_dashers, total_busy_dashers,
dashers_order_ratio as they are weakly correlated
```

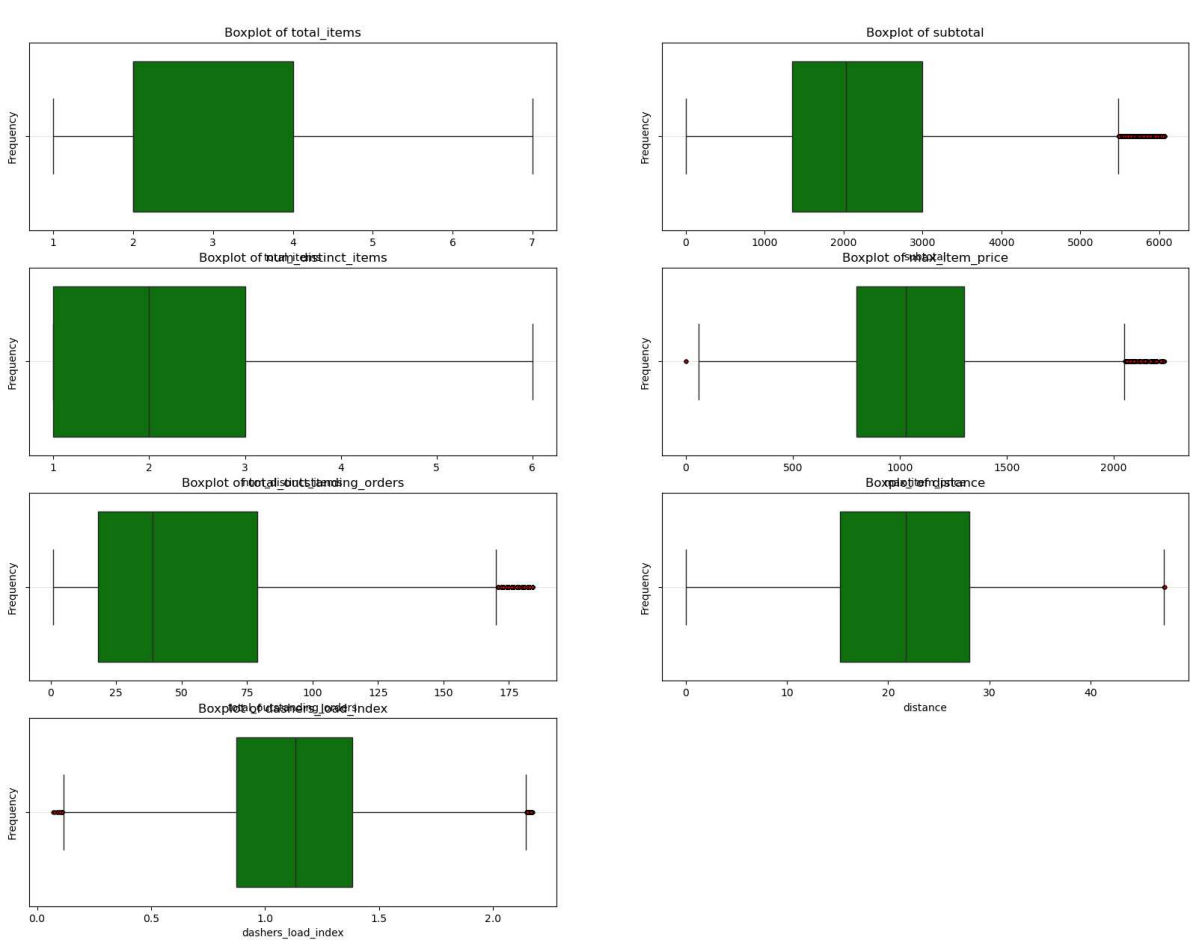
Handling Outliers



Outliers

- ❑ time_taken_mins
- ❑ total_items
- ❑ Subtotal
- ❑ num_distinct_items
- ❑ max_item_price
- ❑ total_outstanding_orders
- ❑ Distance
- ❑ dashers_load_index

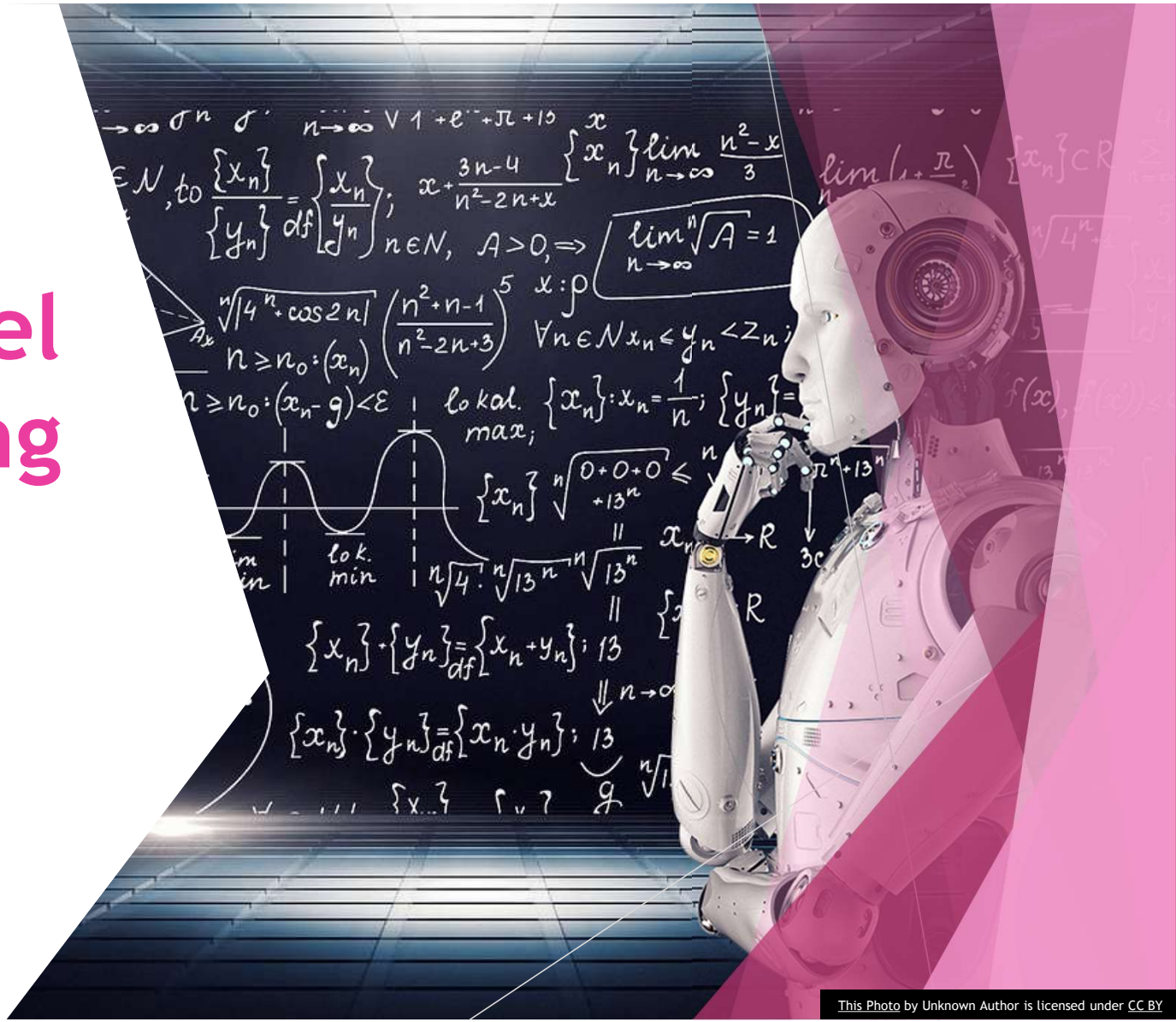
Handling Outliers -2



Outliers

- ✓ time_taken_mins
- ✓ total_items
- ✓ Subtotal
- ✓ num_distinct_items
- ✓ max_item_price
- ✓ total_outstanding_orders
- ✓ Distance
- ✓ dashers_load_index

5. Model Building



Categorical Treatment - Dummy Variables

one-hot encoding

- ✓ Encode binary dummy variables
  - ✓ store_primary_category [Top 20]
  - ✓ hour [Top 10]
- ✓ Each new column flags whether a row belongs to that category (1 if yes, 0 otherwise), then the original store_primary_category column is dropped.
- ✓ This technique is done to limit number of new feature created

one-hot encoding - get_dummies(pandas)

- ✓ Encode binary dummy variables
  - ✓ market_id
  - ✓ order_protocol
  - ✓ day_of_week

SPC_50	SPC_57	SPC_34	SPC_59	SPC_7	Hrs_2	Hrs_1	Hrs_3
0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0
0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0

market_id_2.0	market_id_3.0	market_id_4.0	market_id_5.0	market_id_6.0	order_protocol_2.0
0	0	1	0	0	0
0	1	0	0	0	0
1	0	0	0	0	0
0	0	0	0	0	0
1	0	0	0	0	0

Scaling

Scaling numerical variables standardizes feature ranges so that no single predictor dominates due to its magnitude differences.

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler(with_mean=True, with_std=True)
X_train[num] = scaler.fit_transform(X_train[num])
X_test[num] = scaler.fit_transform(X_test[num])
```

otal_items	subtotal	num_distinct_items	max_item_price	total_outstanding_orders	distance
2.244204	0.569776	0.486659	-1.218130	0.741279	0.159072
-0.496441	1.411545	-0.328762	-0.193642	-0.391711	0.375848
-1.181602	-1.316350	-1.144184	-0.954401	1.163374	-0.786441
-1.181602	-1.155107	-1.144184	-0.964544	0.052599	1.575034
1.559042	0.408533	1.302081	-0.710958	-0.302849	0.380460

Build a Linear Regression

Statsmodels – using OLS – Model 1

1. Create Initialize the mode
2. Add constant to X_train and X_test
3. Train the model
4. Predict the model
5. Calculate R2 score
6. Testing (p-value < 0.05) with multicollinearity diagnostics (VIF ≤ 5)

OLS Regression Results

Dep. Variable:	time_taken_mins	R-squared:	0.785
Model:	OLS	Adj. R-squared:	0.785
Method:	Least Squares	F-statistic:	7856.
Date:	Mon, 28 Apr 2025	Prob (F-statistic):	0.00
Time:	16:35:22	Log-Likelihood:	-3.2265e+05
No. Observations:	116454	AIC:	6.454e+05
Df Residuals:	116399	BIC:	6.459e+05
Df Model:	54		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	53.3248	0.066	802.135	0.000	53.194	53.455
total_items	-0.1771	0.029	-6.181	0.000	-0.233	-0.121
subtotal	1.4968	0.023	65.502	0.000	1.452	1.542
num_distinct_items	0.7179	0.025	28.553	0.000	0.669	0.767
max_item_price	0.1503	0.018	8.333	0.000	0.115	0.186
total_outstanding_orders	1.1373	0.019	58.488	0.000	1.099	1.175
distance	4.1515	0.011	364.006	0.000	4.129	4.174
dashers_load_index	4.2393	0.014	294.662	0.000	4.211	4.268
isWeekend	-0.0383	0.025	-1.561	0.119	-0.086	0.010

P-Value > 0.05

['isWeekend', 'SPC_55', 'SPC_58', 'SPC_24', 'SPC_39', 'SPC_45', 'SPC_50', 'SPC_57', 'order_protocol_7.0']

	feature	VIF
55	day_of_week_6	inf
8	isWeekend	inf
54	day_of_week_5	inf
0	const	34.456916
1	total_items	6.398721

1. Features with p-Value > 0.05 should be dropped
2. Vif Analysis
    1. Inf VIF is caused by day_of_week6, IsWeekend and day_of_week5 are highly correlated [Day5 and Day is Weekend so redundant] – Drop them
    2. Total_items could be dropped too

RFE to select important features

Iteration	Features Combination	MAE	MSE	R2	#Rank
7	('distance', 'dashers_load_index', 'Hrs_20', 'Hrs_19', 'Hrs_21', 'Hrs_22', 'market_id_2.0', 'market_id_4.0')	3.7487	22.6536	0.6642	1
26	('distance', 'dashers_load_index', 'Hrs_20', 'Hrs_19', 'Hrs_21', 'market_id_2.0', 'market_id_3.0', 'market_id_4.0')	3.7656	22.7601	0.6626	2
17	('distance', 'dashers_load_index', 'Hrs_20', 'Hrs_19', 'Hrs_21', 'Hrs_23', 'market_id_2.0', 'market_id_4.0')	3.7619	22.7994	0.662	3
29	('distance', 'dashers_load_index', 'Hrs_20', 'Hrs_19', 'Hrs_21', 'market_id_2.0', 'market_id_4.0', 'market_id_5.0')	3.7787	22.9248	0.6602	4
30	('distance', 'dashers_load_index', 'Hrs_20', 'Hrs_19', 'Hrs_21', 'market_id_2.0', 'market_id_4.0', 'market_id_6.0')	3.7862	23.0202	0.6588	5
101	('distance', 'dashers_load_index', 'Hrs_20', 'Hrs_21', 'market_id_2.0', 'market_id_3.0', 'market_id_4.0', 'market_id_5.0')	3.8181	23.2257	0.6557	6
81	('distance', 'dashers_load_index', 'Hrs_20', 'Hrs_21', 'Hrs_22', 'market_id_2.0', 'market_id_3.0', 'market_id_4.0')	3.8106	23.2311	0.6556	7
46	('distance', 'dashers_load_index', 'Hrs_20', 'Hrs_19', 'Hrs_22', 'market_id_2.0', 'market_id_3.0', 'market_id_4.0')	3.8204	23.2924	0.6547	8
66	('distance', 'dashers_load_index', 'Hrs_20', 'Hrs_19', 'market_id_2.0', 'market_id_3.0', 'market_id_4.0', 'market_id_5.0')	3.8292	23.2925	0.6547	9
72	('distance', 'dashers_load_index', 'Hrs_20', 'Hrs_21', 'Hrs_22', 'Hrs_23', 'market_id_2.0', 'market_id_4.0')	3.8083	23.2988	0.6546	10

Build the model with 8 Selected Features

Selected features: ['distance', 'dashers_load_index', 'Hrs_20', 'Hrs_19', 'Hrs_21', 'Hrs_22', 'market_id_2.0',

Statsmodels – using OLS – Model 1

1. Create Initialize the mode
2. Add constant to X_train and X_test
3. Train the model
4. Predict the model
5. Calculate R2 score
6. Testing (p-value < 0.05) with multicollinearity diagnostics (VIF ≤ 5)

Conclusion

1. Model Build – Successful
2. R – Squared 0.671
3. P-Value is 0 for all Feature
4. There is no VIF
5. R-squared for training set: 0.6707
6. R-squared value - Test Set : 0.6642

OLS Regression Results

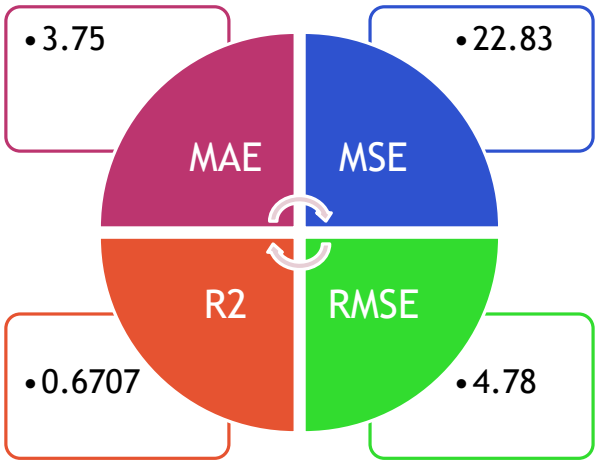
Dep. Variable:	y	R-squared:	0.671
Model:	OLS	Adj. R-squared:	0.671
Method:	Least Squares	F-statistic:	2.965e+04
Date:	Mon, 28 Apr 2025	Prob (F-statistic):	0.00
Time:	17:20:59	Log-Likelihood:	-3.4738e+05
No. Observations:	116454	AIC:	6.948e+05
Df Residuals:	116445	BIC:	6.949e+05
Df Model:	8		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	48.2367	0.023	2092.667	0.000	48.192	48.282
distance	4.2210	0.014	300.670	0.000	4.194	4.249
dashers_load_index	4.9001	0.015	335.273	0.000	4.871	4.929
Hrs_20	-4.6358	0.050	-91.927	0.000	-4.735	-4.537
Hrs_19	-3.7070	0.055	-67.452	0.000	-3.815	-3.599
Hrs_21	-4.0092	0.058	-69.219	0.000	-4.123	-3.896
Hrs_22	-2.8936	0.066	-43.674	0.000	-3.023	-2.764
market_id_2.0	-5.5595	0.034	-164.235	0.000	-5.626	-5.493
market_id_4.0	-3.3132	0.035	-93.476	0.000	-3.383	-3.244

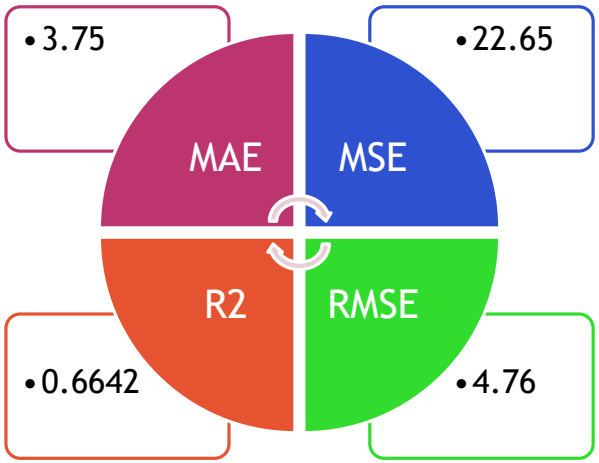
Omnibus:	5100.991	Durbin-Watson:	2.004
Prob(Omnibus):	0.000	Jarque-Bera (JB):	6473.210
Skew:	0.465	Prob(JB):	0.00
Kurtosis:	3.686	Cond. No.	5.36


Metrics and
Evaluation

Evaluation Matrices



Training Set



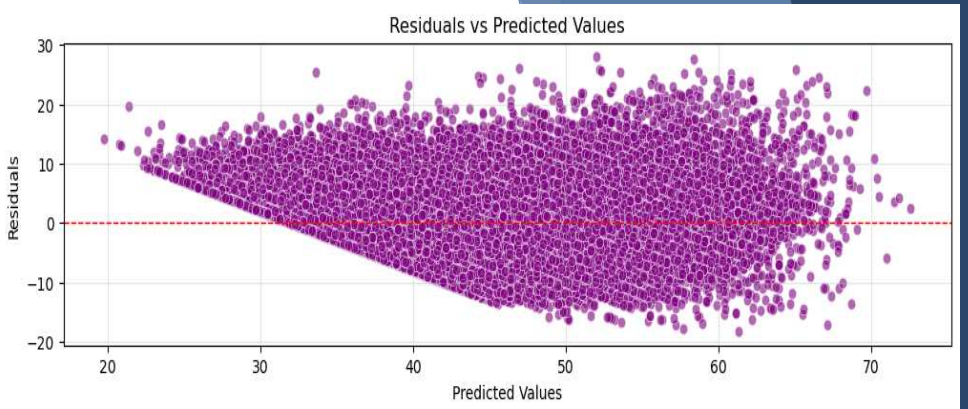
Test Set

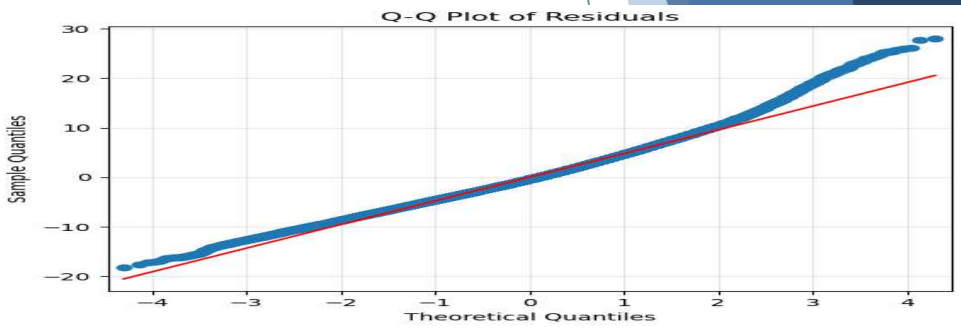
Perform Residual Analysis

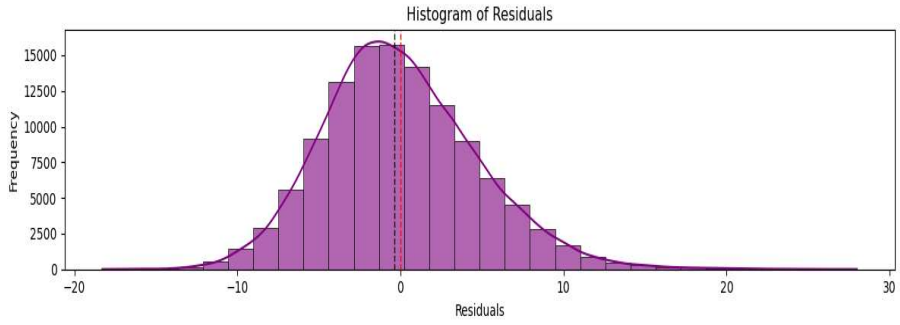
1. The Residuals are random scatter and centered around Zero, suggests the model captures the systematic part of data well.

2. Q-Q Plot: Residuals mostly align with red reference line which represent theory of a normal distribution indicative of, residuals are approximately normally distributed

Residual histogram, the residuals are approximately symmetric around zero and closely follow a normal distribution, as indicated by the smooth overlaid curve. The dashed vertical lines, likely representing measures of central tendency, confirm that most errors center near zero

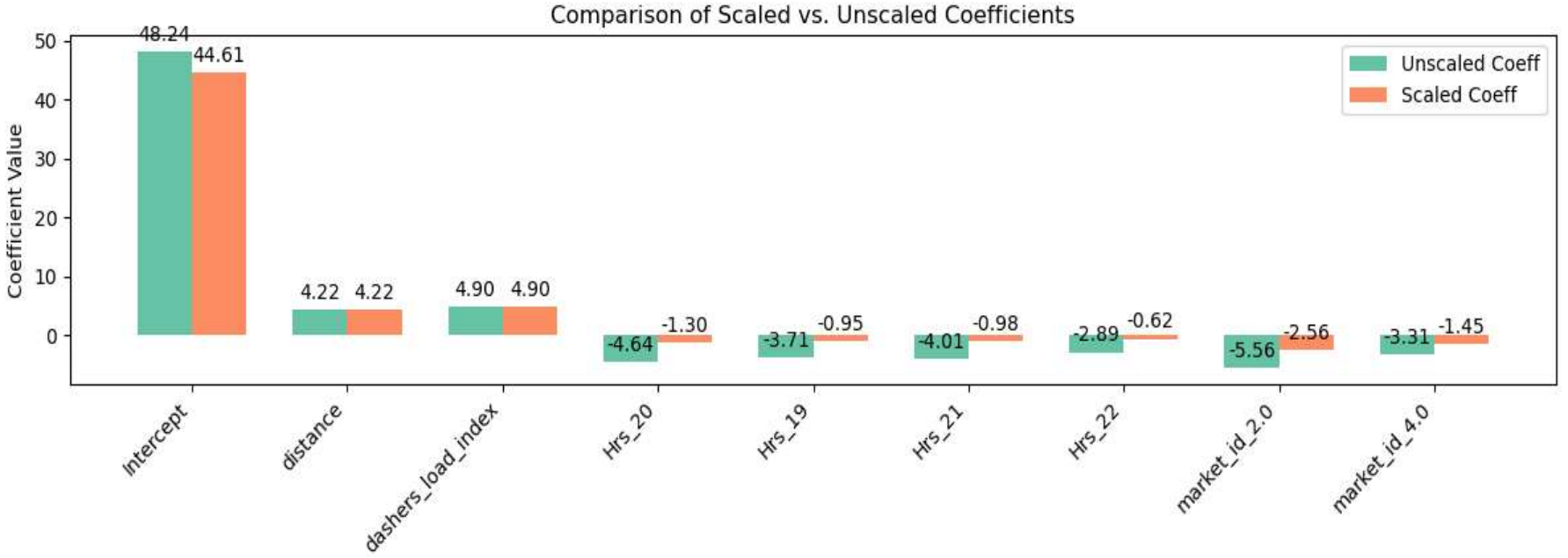






Overall, these plot supports the assumption of normality in the error terms—a key requirement in linear regression—for reliable inference and model performance.

Perform Coefficient Analysis



	Features	Unscaled Coeff	Scaled Coeff
0	Intercept	48.236717	44.612946
1	distance	4.221033	4.221033
2	dashers_load_index	4.900068	4.900068
3	Hrs_20	-4.635766	-1.301344
4	Hrs_19	-3.706996	-0.953842
5	Hrs_21	-4.009153	-0.984431
6	Hrs_22	-2.893637	-0.621141
7	market_id_2.0	-5.559537	-2.562343
8	market_id_4.0	-3.313211	-1.452055

```
Effect of a unit change in distance on Delivery Time: 4.2210 units
Effect of a unit change in dashers_load_index on Delivery Time: 4.9001 units
Effect of a unit change in Hrs_20 on Delivery Time: -4.6358 units
Effect of a unit change in Hrs_19 on Delivery Time: -3.7070 units
Effect of a unit change in Hrs_21 on Delivery Time: -4.0092 units
Effect of a unit change in Hrs_22 on Delivery Time: -2.8936 units
Effect of a unit change in market_id_2.0 on Delivery Time: -5.5595 units
Effect of a unit change in market_id_4.0 on Delivery Time: -3.3132 units
```

Summary

Data Preparation & Feature Engineering

- Data from 175,777 records were cleaned and preprocessed. Date and categorical variables were transformed, outliers were removed using the IQR method, and new features (like delivery hour, day of week, dashers load index, and dashers order ratio) were derived.

Exploratory Analysis

- Visual and statistical analyses highlighted that most features are right-skewed, with the exception of distance. Correlation analysis showed that “distance” and “dashers_load_index” are the strongest predictors of delivery time.

Model Building & Evaluation

- Multiple models were built and evaluated. The final model, using eight key features—[distance, dashers_load_index, Hrs_20, Hrs_19, Hrs_21, Hrs_22, market_id_2.0, market_id_4.0]—achieved an R^2 of approximately 0.67. Residual diagnostic plots confirmed that error assumptions, such as randomness around zero and approximate normality, were adequately met.

Coefficient Insight

- The coefficients indicate that for every additional unit of distance, delivery time increases by about 4.22 minutes. Similarly, an increase in the dashers load index contributes about 4.90 extra minutes. Certain hour-related features (e.g., Hrs_20) show a reducing effect on delivery time, suggesting temporal patterns in service efficiency.

Recommendations for Porter Services

Optimize Delivery Routes

Since distance plays a significant role in increasing delivery time, consider investing in advanced routing and navigation algorithms. This can help minimize travel distances and reduce delays.

Balance Dashers' Workload

The dashers load index is a critical factor. Review operational assignments and scheduling to avoid overloading delivery partners. Implementing dynamic dispatching or adjusting shift patterns during peak periods can mitigate delays

Leverage Temporal Patterns

With certain hours (e.g., observed through Hrs_20) leading to faster delivery times, reallocate resources or offer incentives during slower periods. This may smooth out the service delivery curve across the day.

Market-Specific Strategies

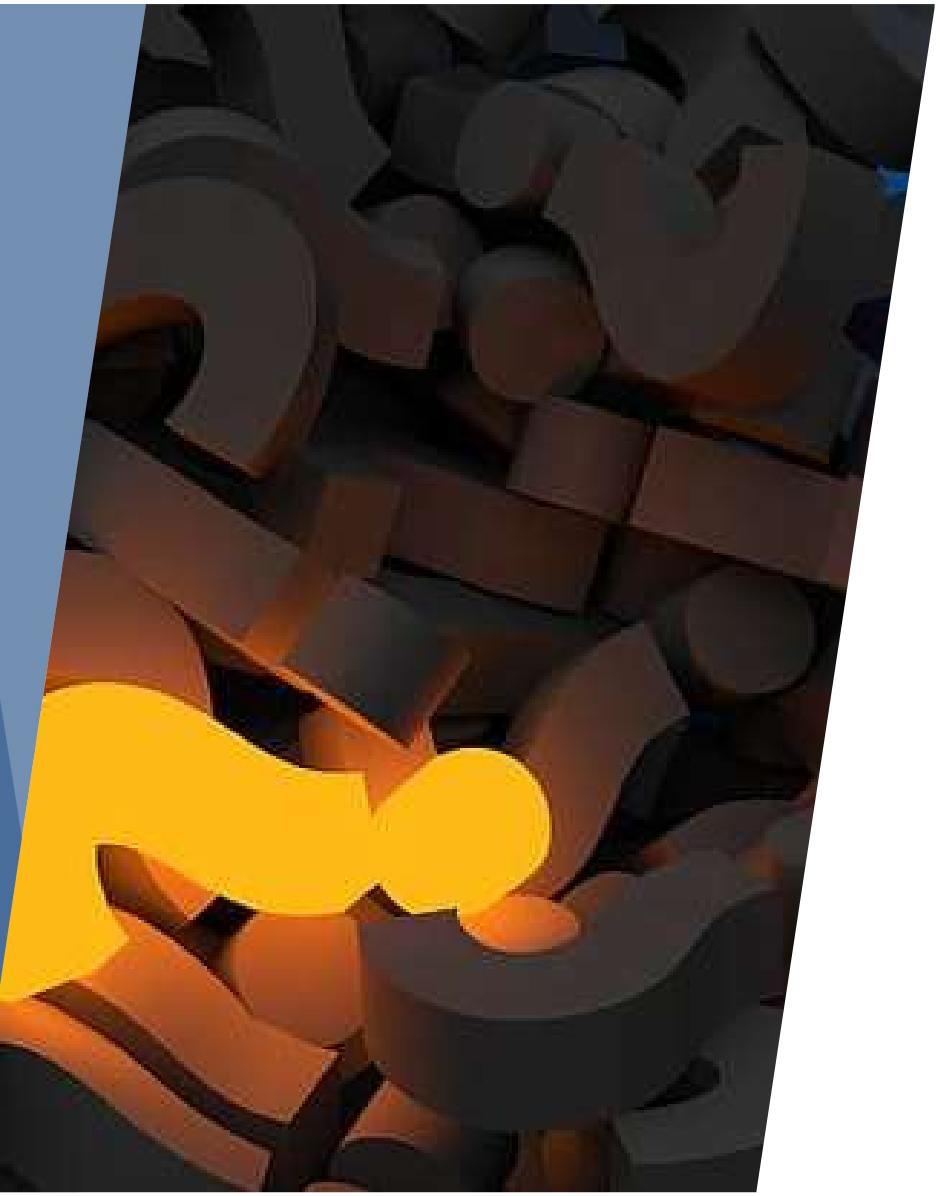
Differences in market_id effects suggest that some markets inherently experience longer or shorter delivery times. Tailor localized strategies, such as targeted staffing or traffic management partnerships, to improve performance in markets with higher delays.

Continuous Monitoring & Model Updates

As operational dynamics evolve (e.g., due to seasonal changes or expansion), routinely update the model with new data. Continuous monitoring can help detect emerging patterns and ensure that the predictive model remains accurate and actionable.

Investigate Outliers

The analyses reveal a few extreme delivery time cases. Conduct further investigations into these outliers to identify any underlying issues (such as logistic bottlenecks or external factors) and implement corrective measures.



Subjective Questions

Question 2

Question 1: Are there any categorical variables in the data? From your analysis of the categorical variables from the dataset, what could you infer about their effect on the dependent variable?

Yes, there are categorical variables in the porter dataset such as marketid, store_primary_category, order_protocil, day_of_week, isWeekend.

>`Market Id` represting diffrent zones which could impact delivery time, as certain market may have congestion or traffic leading to longer delviery time

>`Store Primary Category` Type of services offered by store may affect delivery time For instance **Fast Food** might have shorter delviery times compared to restaurants

>`Order Protocol` Some of protocols as seen in data are heavily used compared to others

>`Hour` , `Day of week` and `Weekend` Delivery time is affected by all during peak hours or if its weekend

Question 2

What does `test_size = 0.2` refer to during splitting the data into training and test sets?

Parameter `test_size =0.2` is used for splitting a dataset into training and test sets. It markes the proportions of the data allocated to train and test. when it is **0.2** means **20** of the total dataset will be used for testing while **80%** for training.

For example, If datset contains 10000 records then `Test Set` will be **2000** and `Training Set` would be **8000** records.

Question 3

Looking at the heatmap, which one has the highest correlation with the target variable?

`Distance` and ****derived feature**** `dashers_load_index` has the strongest positive correlation with target variable `time_taken_mins`, meaning longer distances significantly increase delivery times.

Question 4

What was your approach to detect the outliers? How did you address them?

I have used `IQR` method to detect outliers for each `continuous` variable. I computed `Q1` and `Q3` followed by `IQR(Q3-Q1)` and defined lower and upper bound as `Q1 - 1.5(IQR)` and `Q3 + 1.5(IQR)`.

Data outside of these bounds were taken as outliers and removed from both `training` and `test` dataset

Additionally, Box plots were used to identify outlier visually and at last `describe` function was used to validate

Question 5

Based on the final model, which are the top 3 features significantly affecting the delivery time?

Top 3 feature significantly affecting the delviery time are
1. `distance` Coeffient 4.22 --> For extra unit of distance delviery time is increased by 4.22 minutes
2. `dashers_load_index` Coeffient 4.9 --> If Dashers are loaded then it will affact the delivery time by 4.9 minutes
3. `Hrs_20` Coeffient -1.3 --> Order placed at 8PM, could lead to reduction in delivery time by 1.3 minutes, suggesting faster delivery

Question 6

Explain the linear regression algorithm in detail

Linear Regression falls under supervised learning and it used to build relationship between dependent variable `Target` and independent variables `Features` by fitting Linear equation **[y = mX +b]**.

The Goal of `Linear Regression` is to find **best-fit** line that minimizes the error between predicted and actual values.

Types of Linear Regression
1. `Simple Linear Regression` (only one independent variable)
2. `Multiple Linear Regression` (Many independent variable)

Important Assumptions
- `Linearity`: Relationship between variable is linear
- `Independence`: Features are not correlated
- `Homoscedasticity`: Constant error variance
- `No Multicollinearity`: Features should not be highly correlated

Pros
- Easy to interpret.
- Computationally efficient.

Cons
- Struggles with non-linearity.
- Sensitive to outliers.

Question 7

Explain the difference between simple linear regression and multiple linear regression

Simple Linear Regression	Multiple Linear Regression
Models the relationship betwen one independent variable (X) and a dependent variable (Y)	Works with two or more independent variables affecting dependent target variable
$Y= mX + b$	$Y= b_0 + b_1X_1 + b_2X_2 + ... + b_nX_n$
To be used when there is only single factor affecting the outcome	To be used when result is impacted by multiple factors
Predict : price of house just based on location	Predict: House price based on location, size , area, number of bedrooms etc
Complexity is low andinterpretation is easy	Complexity is high and require more analysis

Question 8

What is the role of the cost function in linear regression, and how is it minimized?

Cost or loss function measures how well the model fits the data by calculating difference between predicted and actual valus. One of the most comman industry metric is Mean Squared Error(MSE).

Cost function can be minimized by using Gradient Desent
- Calculate slope of cost function
- Adjust model parameters
- continue updating parameter untile minimum

Question 9

Explain the difference between overfitting and underfitting.

Overfitting` it occurs when model learns training data too good, including noise. With increased complexity model performs well on training data and poorly on test and unknown data. To simplify models, learn training data instead of learning its patterns.

Effects of overfitting
- High accuracy on training but poor performance on test data
- Sensitive to small variations in the data, making it unstable

`Underfitting`: Occures when a model fails to capture the pattern in data, this can be caused if model is too simple and does not observe from training data, leading to poor performance on both training and test data.

Effects of overfitting
- Low accuracy on both Train and Test

How to avoid Overfitting and Underfitting
`Overfitting`
- Use regularization techniques
- Collect more training data or mockup data(augmentation)
- Reduce model complexity

`Underfitting`
- Train for longer durations
- Ensure data hygiene
- Use complex model

Question 10

How do residual plots help in diagnosing a linear regression model?

`Residual plots` visualize the discrepancies between actual and predicted values (the residuals) to help assess key model assumptions and potential issues.

`Linearity`: A good residual plot shows a random scatter of points around zero without any distinct pattern otherwise response is not linear
`Homoscedasticity`: Constant variance across all levels of predicted values
`Normality of Errors': Residual plots like histograms or Q-Q plots are used separately, they complement scatter plots to verify whether residuals are approximately normally distributed

To Summarize, `residual plots` provide a visual check for the linear regression assumptions (linearity, constant variance, and normality) and help identify data points that might require further investigation or remediation.

Thank You

Suyash Nagar