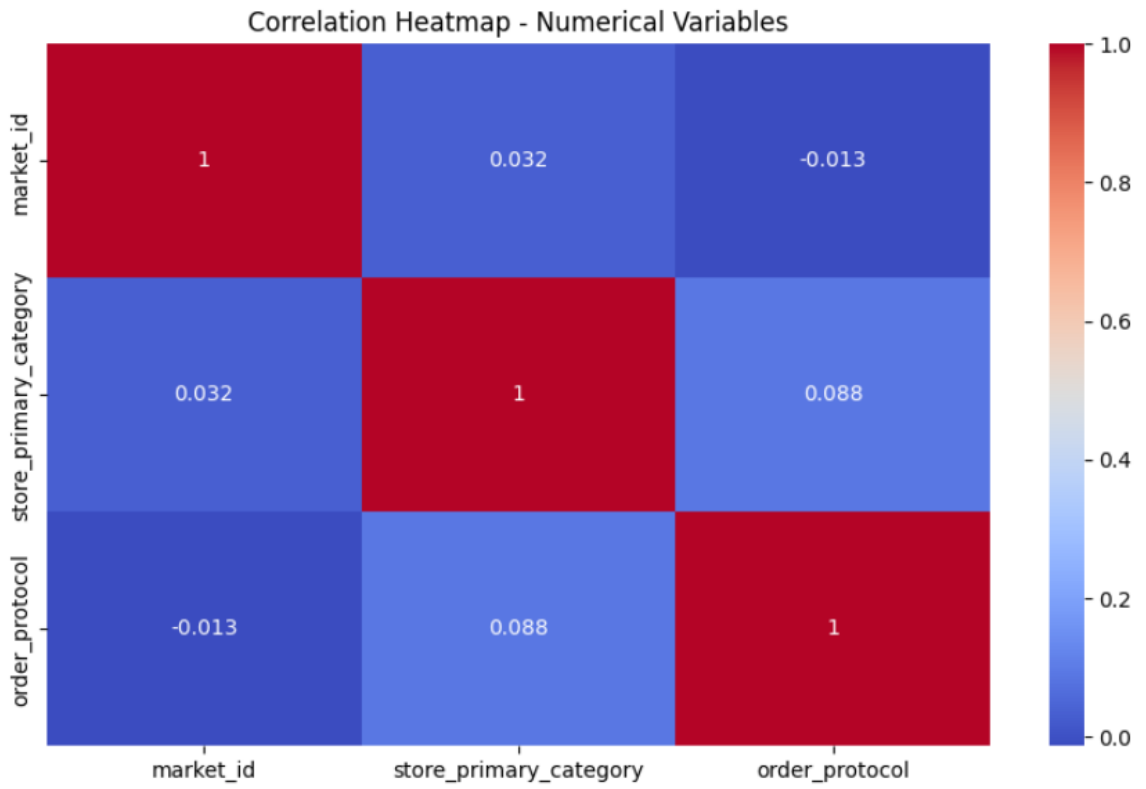
LR_Delivery_Time_Estimation

```
1 df.describe()
```

	total_items	subtotal	num_distinct_items	min_item_price	max_item_price	total_onshift_dashers	total_busy_dashers	total_outstanding_orders	distance	delivery_minutes	created_hour	created_day
count	175777.000000	175777.000000	175777.000000	175777.000000	175777.000000	175777.000000	175777.000000	175777.000000	175777.000000	175777.000000	175777.000000	175777.000000
mean	3.204976	2697.111147	2.675060	684.965433	1160.158616	44.918664	41.861381	58.230115	21.843090	46.203013	8.473441	3.222293
std	2.674055	1828.554893	1.625681	519.882924	560.828571	34.544724	32.168505	52.731043	8.748712	9.327424	8.676809	2.043874
min	1.000000	0.000000	1.000000	-86.000000	0.000000	-4.000000	-5.000000	-6.000000	0.000000	32.000000	0.000000	0.000000
25%	2.000000	1412.000000	1.000000	299.000000	799.000000	17.000000	15.000000	17.000000	15.360000	39.000000	2.000000	1.000000
50%	3.000000	2224.000000	2.000000	595.000000	1095.000000	37.000000	35.000000	41.000000	21.760000	45.000000	3.000000	3.000000
75%	4.000000	3410.000000	3.000000	942.000000	1395.000000	66.000000	63.000000	85.000000	28.120000	52.000000	19.000000	5.000000
max	411.000000	26800.000000	20.000000	14700.000000	14700.000000	171.000000	154.000000	285.000000	83.520000	110.000000	23.000000	6.000000

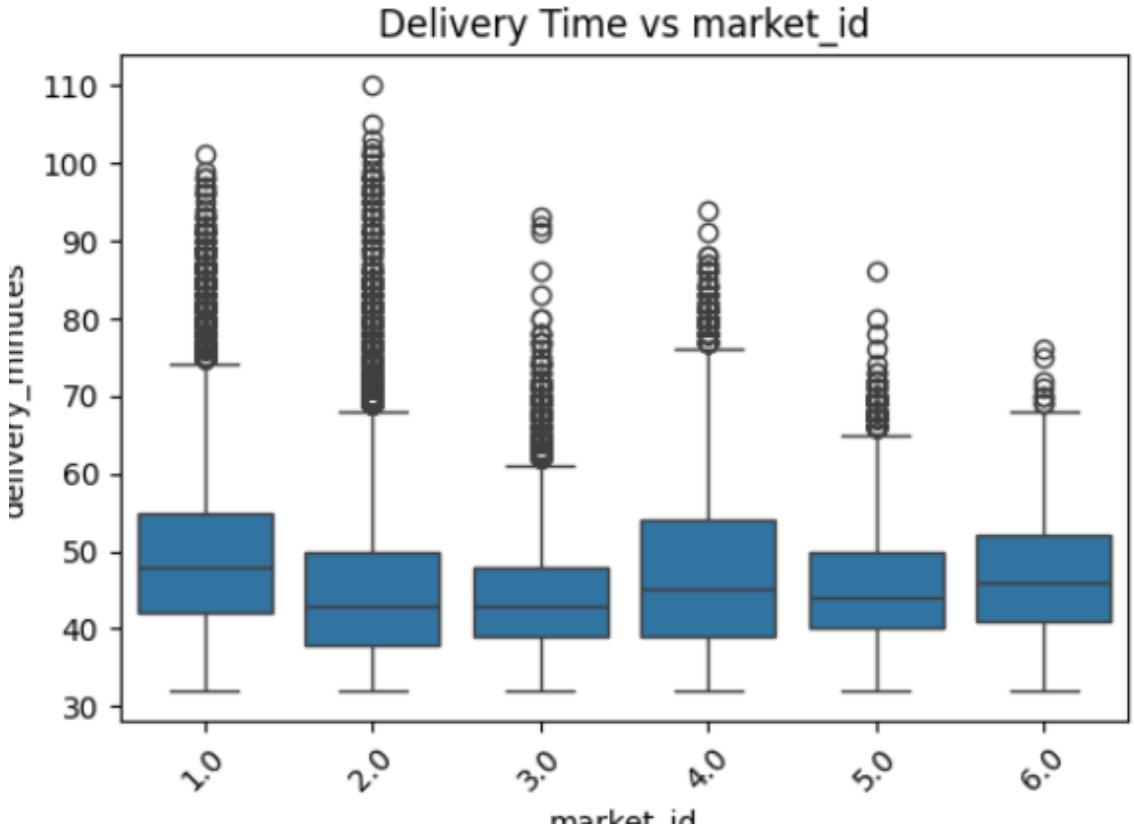
From the summary statistics, we can see that the average delivery time is around 46 minutes, with most deliveries taking between 39 and 52 minutes. The fastest delivery recorded was 32 minutes, while the longest reached 110 minutes. Looking at the distances, the average is about 22 km, with most orders falling between 15 km and 28 km, although there are some extreme outliers above 80 km. In terms of items, customers usually order around 3 items per order, and most orders are small with 2 to 4 items, but there is an unusual outlier with 411 items. The subtotal averages around 2697, but there are some suspicious values such as 0, which may indicate missing or incorrect data. Similarly, the minimum item price shows a negative value (-86), which is not realistic, while the maximum item price goes as high as 14,700, suggesting possible outliers. For driver availability, on average there are about 45 dashers on shift and 42 busy dashers, with around 58 outstanding orders, but in peak situations, this can reach as high as 285. Time features also show that orders are spread across all hours of the day (0–23) and all days of the week (0–6). Overall, while the data gives a good picture of delivery patterns, it also highlights quality issues such as negative values and extreme outliers, which should be cleaned before building a model.

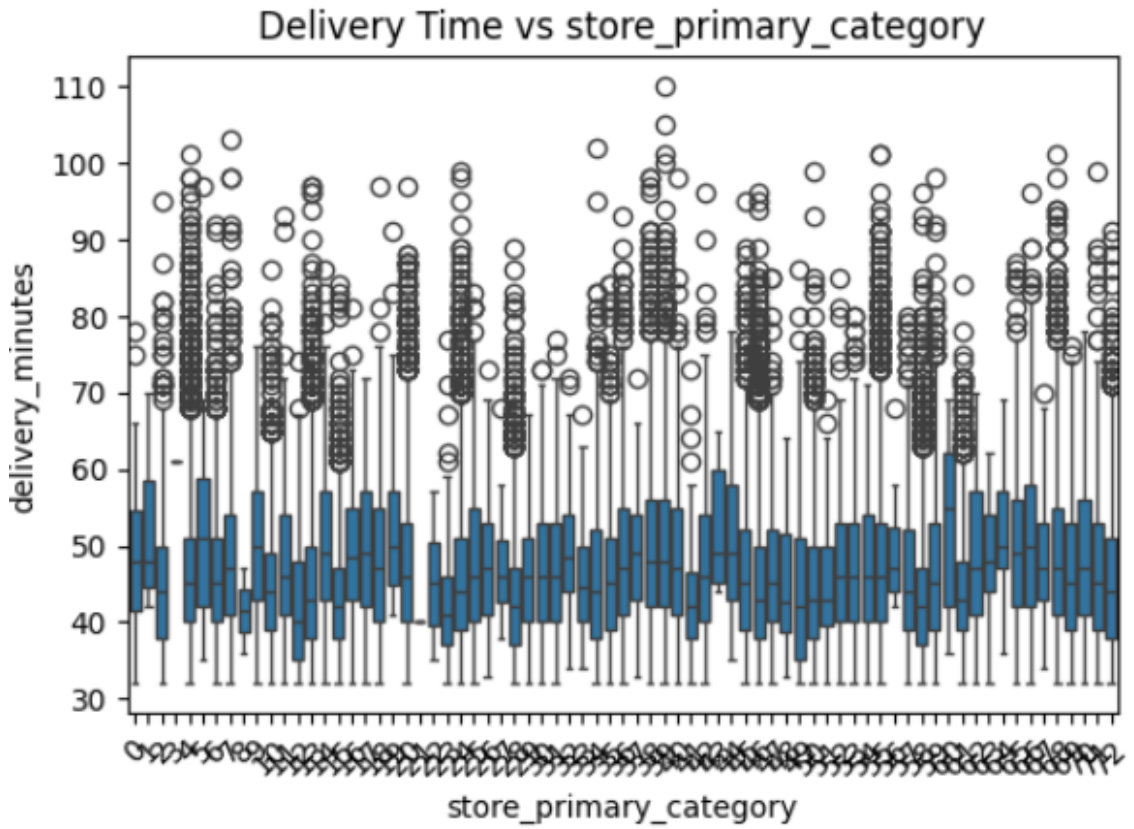
```
df_catogory=df[['market_id','store_primary_category','order_protocol']]

plt.figure(figsize=(10,6))

sns.heatmap(df_catogory.corr(), annot=True, cmap="coolwarm")

plt.title("Correlation Heatmap - Numerical Variables")

plt.show()
```



The correlation heatmap shows that there is almost no strong relationship between the categorical variables market_id, store_primary_category, and order_protocol. All the correlation values are close to zero, which means these variables are largely independent of each other. For example, market_id and store_primary_category have a correlation of only 0.032, while store_primary_category and order_protocol have a slightly higher value of 0.088, but still too weak to indicate any meaningful relationship. This suggests that each of these variables contributes unique information and they are not redundant. Therefore, it would be useful to keep all of them in the model since they do not overlap much in the information they provide.

```
cat_cols = df.select_dtypes(include=["category"]).columns

for col in cat_cols:
    plt.figure(figsize=(6,4))
    sns.boxplot(x=df[col], y=df["delivery_minutes"])
    plt.title(f"Delivery Time vs {col}")
    plt.xticks(rotation=45)
    plt.show()
```


Delivery Time vs market_id
delivery_minutes
110
100
90
80
70
60
50
40
30
1.0
2.0
3.0
4.0
5.0
6.0
market_id


Delivery Time vs store_primary_category
delivery_minutes
110
100
90
80
70
60
50
40
30
store_primary_category

The boxplot of delivery time against market_id shows that delivery times are fairly consistent across different markets, with median values mostly around 45–50 minutes. However, some markets, like market 1 and market 4, show slightly higher medians compared to others, which indicates that delivery speed may vary depending on the region. The second plot, delivery time against store_primary_category, reveals a wide variation across different store categories. While most categories have median delivery times between 40 and 50 minutes, the spread of data and presence of outliers differ greatly, suggesting that the type of store can influence delivery time, possibly due to differences in preparation time or demand. Finally, the plot of delivery time against order_protocol shows that some protocols are associated with shorter delivery times, particularly protocols 6 and 7, which have lower medians compared to others. Protocols 1 to 5, on the other hand, show similar delivery time distributions, clustered around 45–50 minutes. Overall, these insights indicate that delivery time is not only influenced by distance and order size but also by market location, store category, and the protocol used to place the order.

plt.figure(figsize=(8,4))

sns.lineplot(x="created_hour", y="delivery_minutes", data=df)

plt.title("Average Delivery Time by Hour")

plt.show()

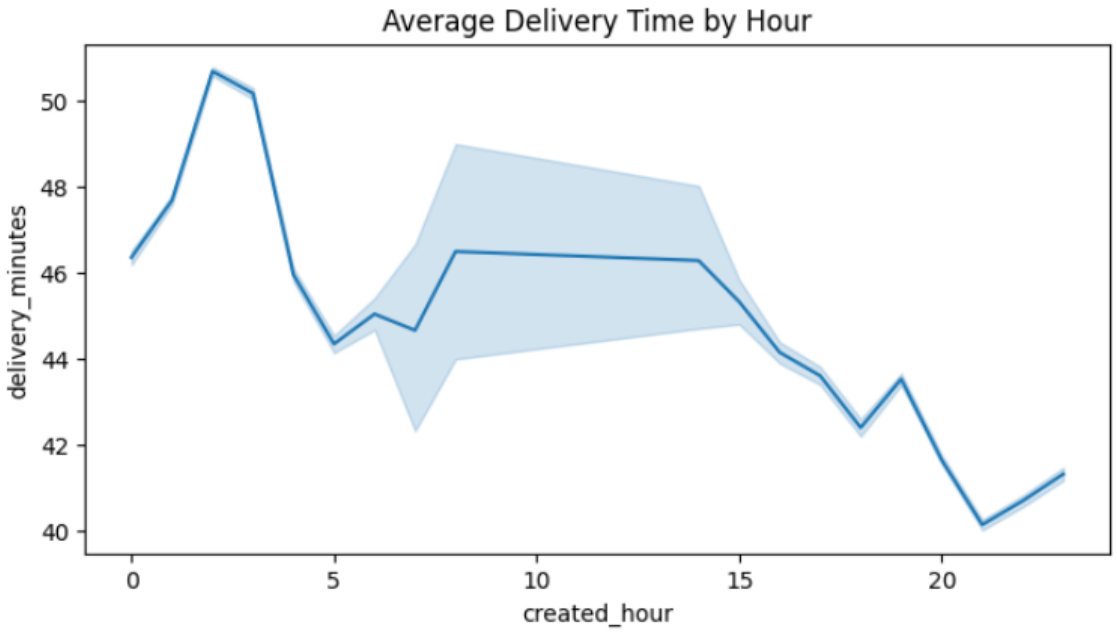
 # Lineplot: day vs delivery time

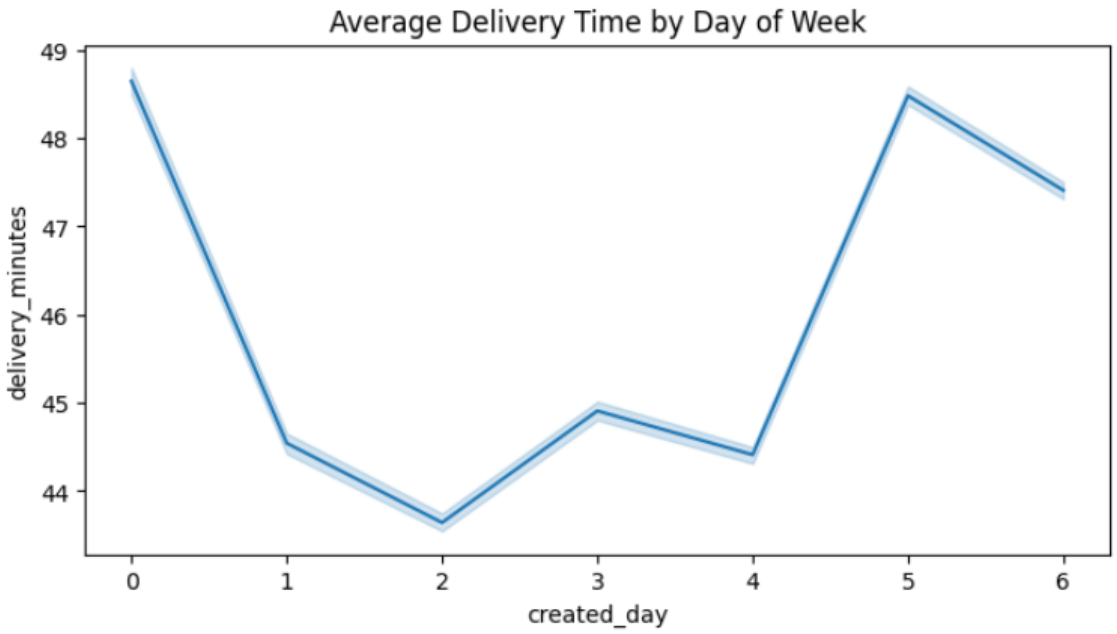
plt.figure(figsize=(8,4))

sns.lineplot(x="created_day", y="delivery_minutes", data=df)

plt.title("Average Delivery Time by Day of Week")

plt.show()



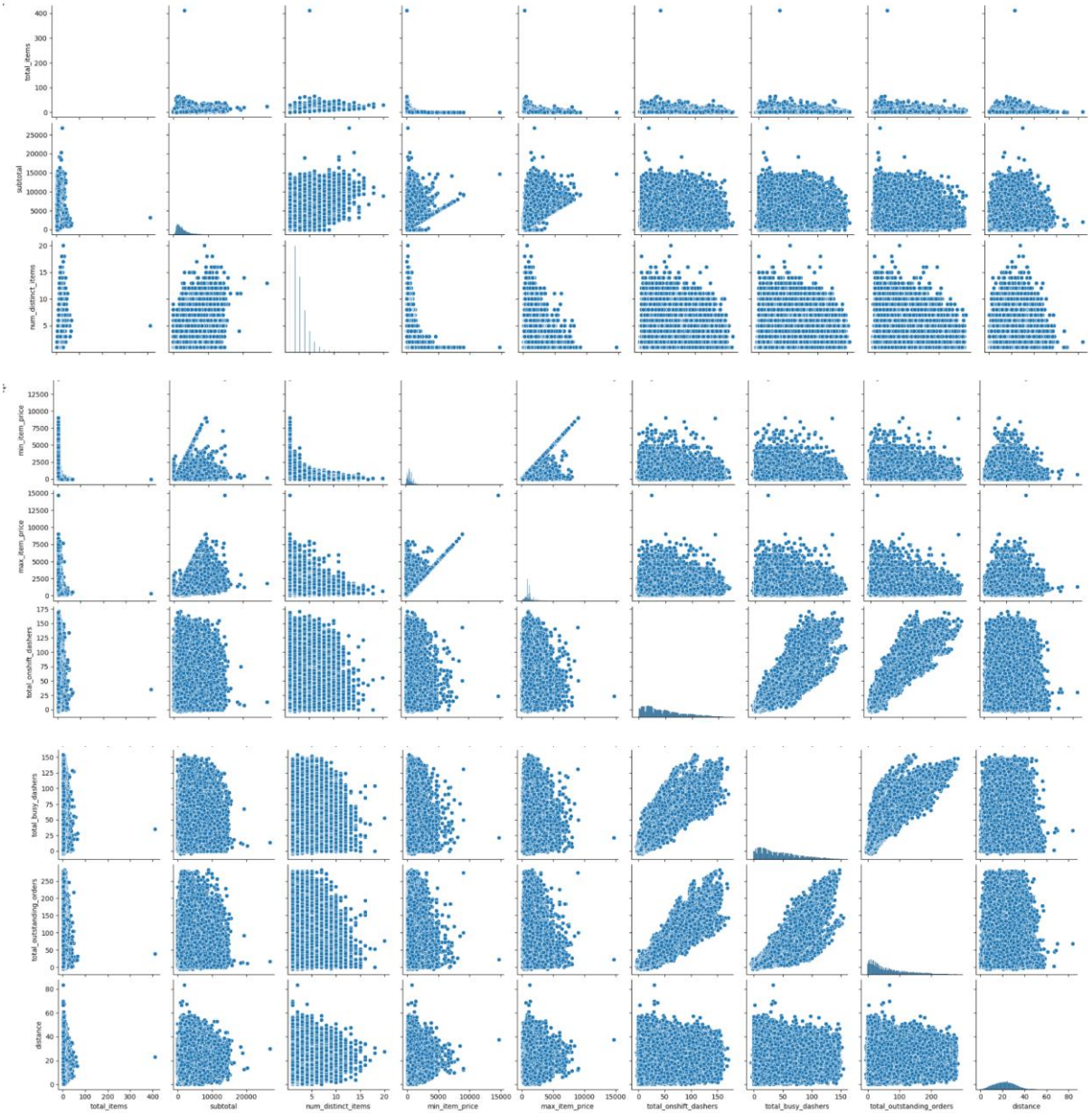


From the hourly trend, it is clear that delivery times are not constant throughout the day. The average delivery time peaks in the early morning around 2–3 AM, crossing the 50-minute mark, which could be due to fewer dashers being available during late-night hours. During most of the daytime, delivery times stabilize between 45 and 47 minutes, before gradually decreasing in the evening, with the fastest deliveries recorded around 8–9 PM, averaging close to 40 minutes. When looking at the weekly pattern, Mondays show the highest average delivery times at around 49 minutes, while midweek days such as

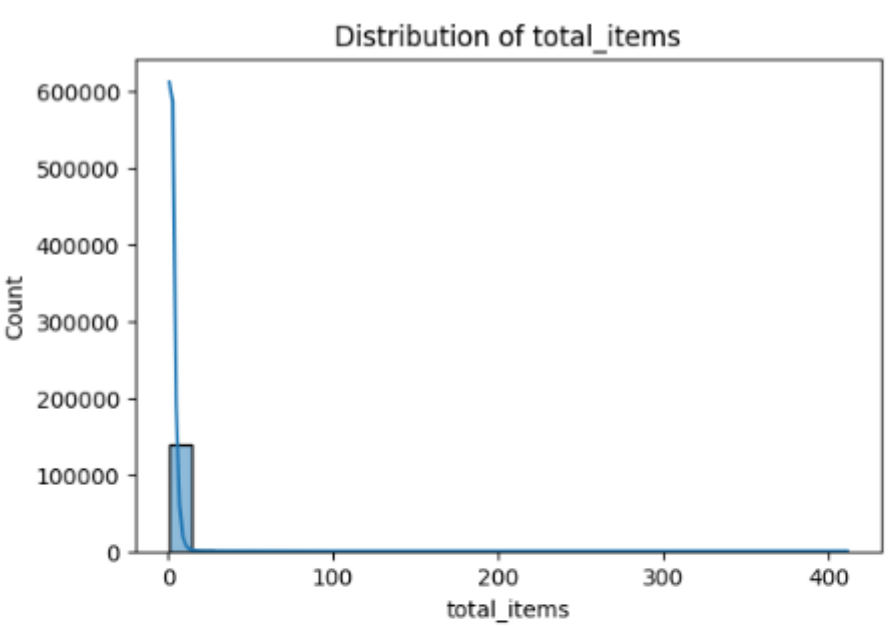
Tuesday and Wednesday are the most efficient, with delivery times dropping to about 44 minutes. As the week progresses towards the weekend, delivery times increase again, with Saturday being almost as slow as Monday. This suggests that both time of day and day of week play an important role in delivery performance, with evenings and midweek being the most favorable periods for faster deliveries.

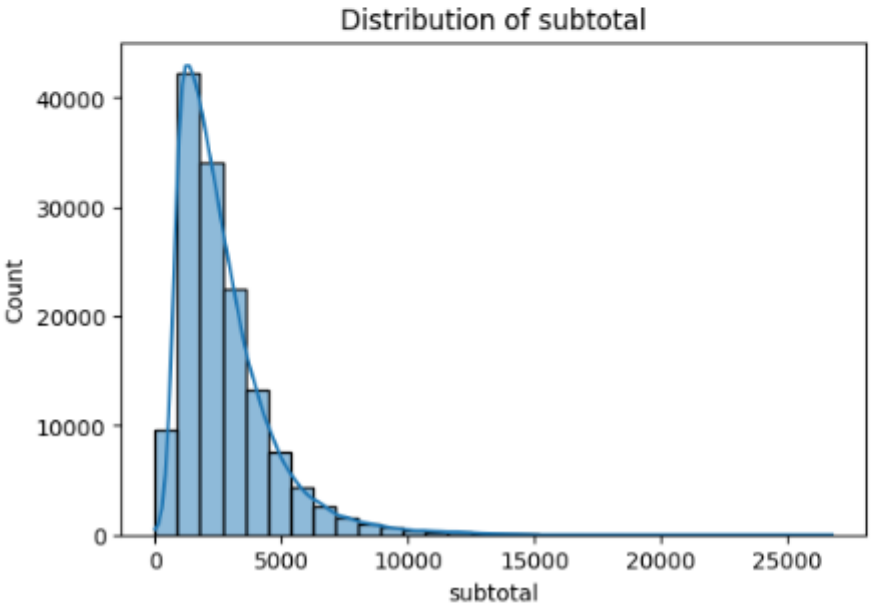
Exploratory Data Analysis on Training Data

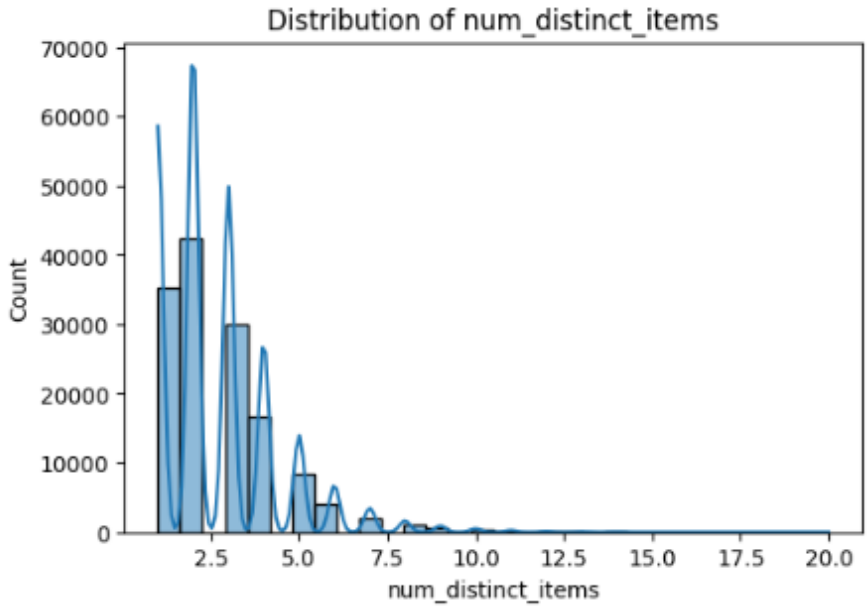
Plot distributions for numerical columns in the training set to understand their spread and any skewness

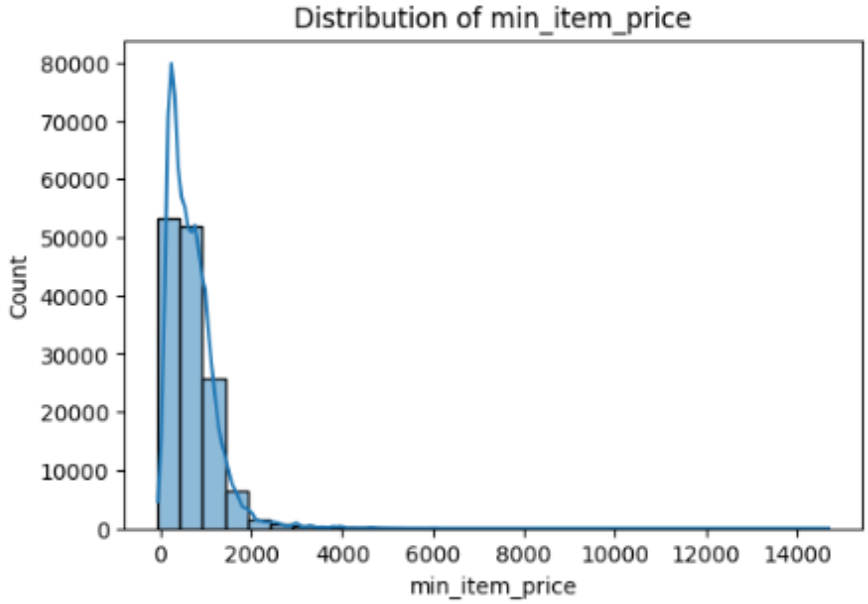


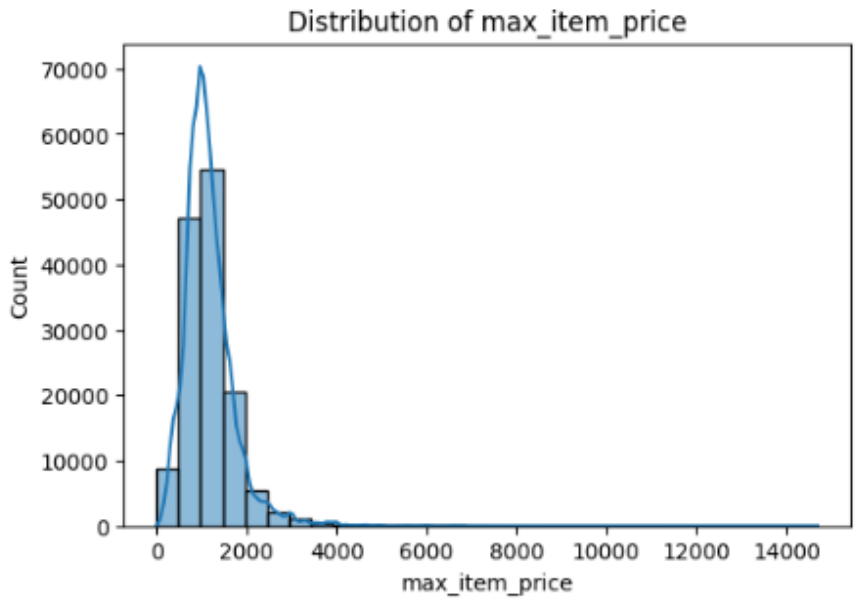
From the pairplot analysis of numerical variables, it is clear that many features are heavily skewed and contain outliers. For example, total_items has a very long tail with some extreme cases above 400 items, while most orders are clustered below 10 items. Similarly, subtotal, min_item_price, and max_item_price also show strong right-skewness with a few very large values that appear as outliers, which could distort the model if not treated. The dasher-related variables such as total_onshift_dashers, total_busy_dashers, and total_outstanding_orders appear to have more balanced distributions, but they still show some extreme high values. When looking at the relationships, there is a clear positive correlation between subtotal and total_items, which makes sense because more items usually increase the subtotal. Likewise, min_item_price and max_item_price show some alignment with subtotal, indicating that higher-priced items contribute to larger order amounts. The distance variable is mostly concentrated between 10 and 30 km, but with some rare extreme values beyond 80 km, again highlighting outliers. Overall, the plots show that while most of the data is clustered in reasonable ranges, several variables suffer from strong skewness and outliers, and these should be addressed during preprocessing to avoid misleading results in modeling.

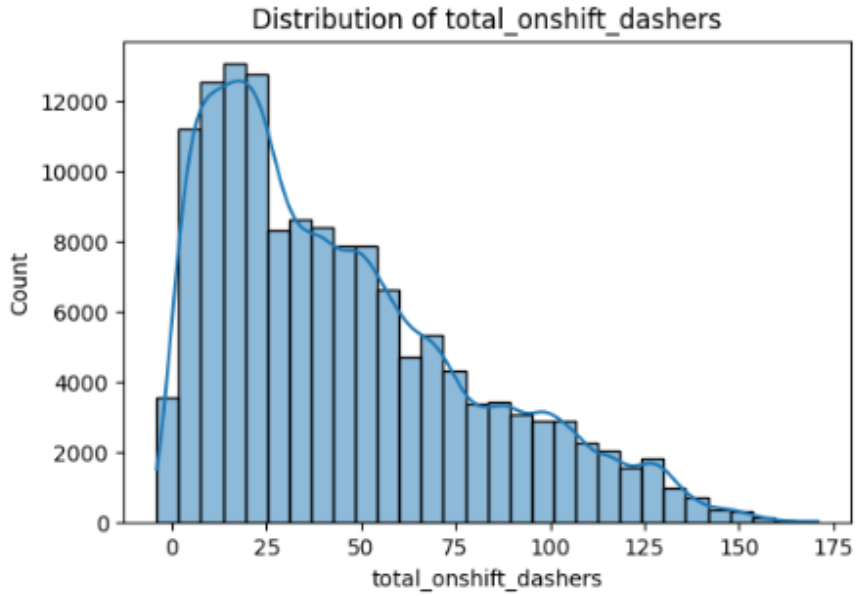

Distribution of total_items
Count
600000
500000
400000
300000
200000
100000
0
0
100
200
300
400
total_items


Distribution of subtotal
Count
40000
30000
20000
10000
0
0
5000
10000
15000
20000
25000
subtotal

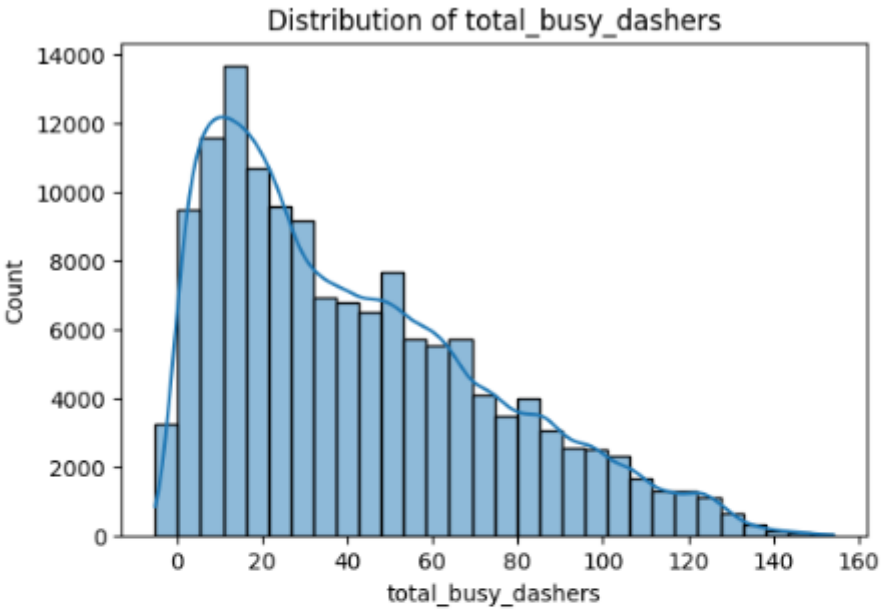

Distribution of num_distinct_items
Count
num_distinct_items
0
10000
20000
30000
40000
50000
60000
70000
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0

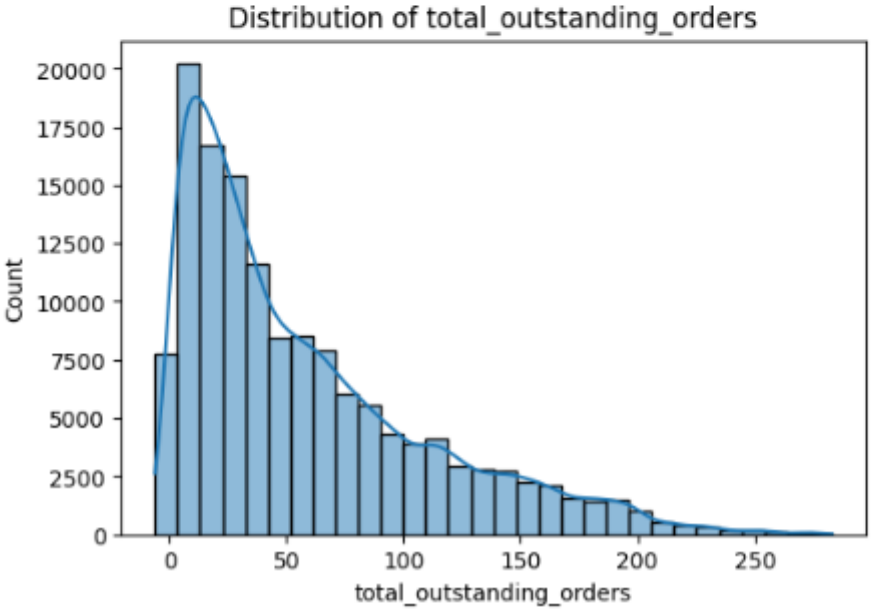

Distribution of min_item_price
Count
min_item_price
0
10000
20000
30000
40000
50000
60000
70000
80000
0
2000
4000
6000
8000
10000
12000
14000

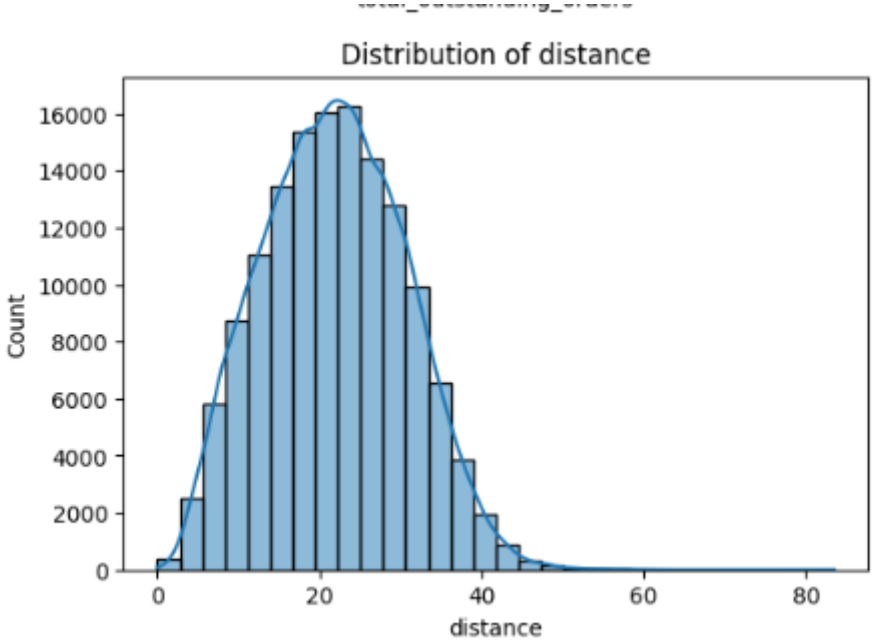

Distribution of max_item_price
Count
max_item_price
0
10000
20000
30000
40000
50000
60000
70000
0
2000
4000
6000
8000
10000
12000
14000


Distribution of total_onshift_dashers
Count
total_onshift_dashers
0
2000
4000
6000
8000
10000
12000
0
25
50
75
100
125
150
175

total_onshift_dashers

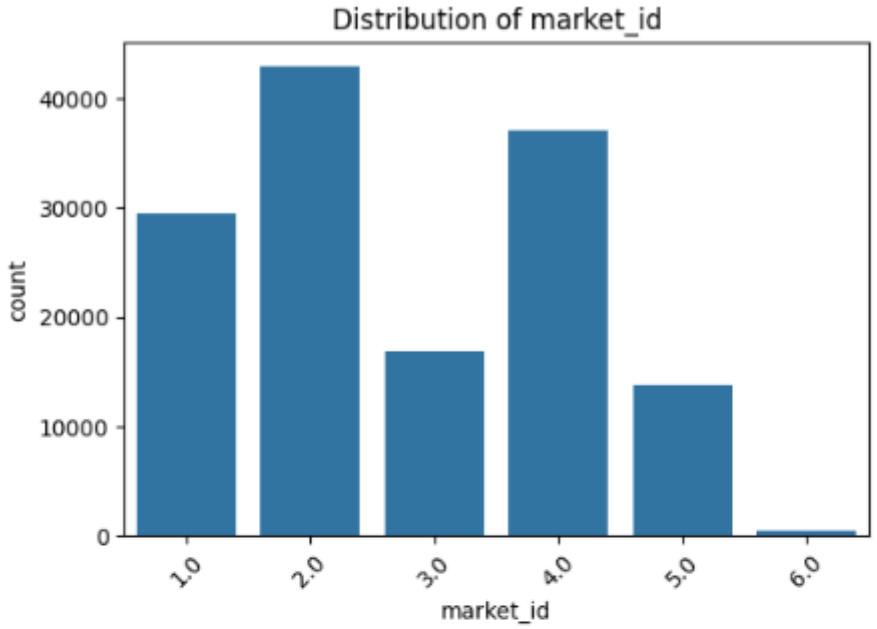


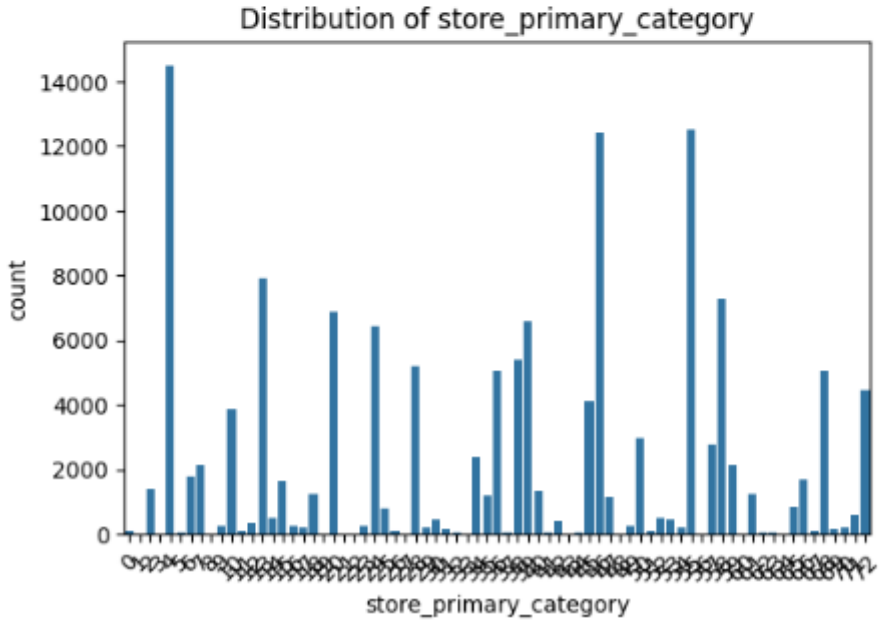


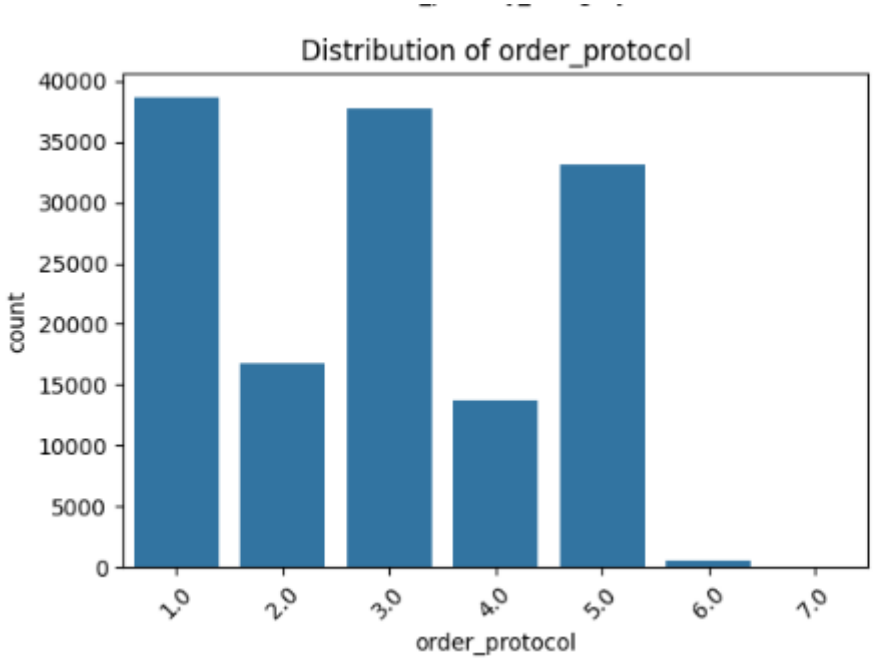


From the pairplot analysis of numerical variables, it is clear that many features are heavily skewed and contain outliers. For example, total_items has a very long tail with some extreme cases above 400 items, while most orders are clustered below 10 items. Similarly, subtotal, min_item_price, and max_item_price also show strong right-skewness with a few very large values that appear as outliers, which could distort the model if not treated. The dasher-related variables such as total_onshift_dashers, total_busy_dashers, and total_outstanding_orders appear to have more balanced distributions, but they still show some extreme high values. When looking at the relationships, there is a clear positive correlation between subtotal and total_items, which makes sense because more items usually increase the subtotal. Likewise, min_item_price and max_item_price show some alignment with subtotal, indicating that higher-priced items contribute to larger order amounts. The distance variable is mostly concentrated between 10 and 30 km, but with some rare extreme values beyond 80 km, again highlighting outliers. Overall, the plots show that while most of the data is clustered in reasonable ranges, several variables suffer from strong skewness and outliers, and these should be addressed during preprocessing to avoid misleading results in modeling.

```
Distribution of categorical columns
```


Distribution of market_id
40000
30000
20000
10000
0
count
1.0
2.0
3.0
4.0
5.0
6.0
market_id


Distribution of store_primary_category
14000
12000
10000
8000
6000
4000
2000
0
count
store_primary_category



The distribution of market_id shows that most of the data is concentrated in a few markets, with markets 2 and 4 having the largest number of orders, while market 6 has very few orders compared to others. This indicates that the dataset is imbalanced across markets, and the model might learn more from high-volume markets than from smaller ones. The store_primary_category variable is highly diverse, with many different categories, but only a few categories have very high counts while most categories have relatively few orders. This suggests that store type is an important feature but may require grouping or encoding carefully to avoid noise from rarely occurring categories. Finally, the distribution of order_protocol shows that protocols 1, 3, and 5 are the most common, while protocol 6 has very few records. Similar to market_id, this imbalance means that some protocols may not provide enough data for the model to learn strong patterns. Overall, the categorical features show useful variation, but they are unevenly distributed, and care should be taken to handle categories with very low counts.

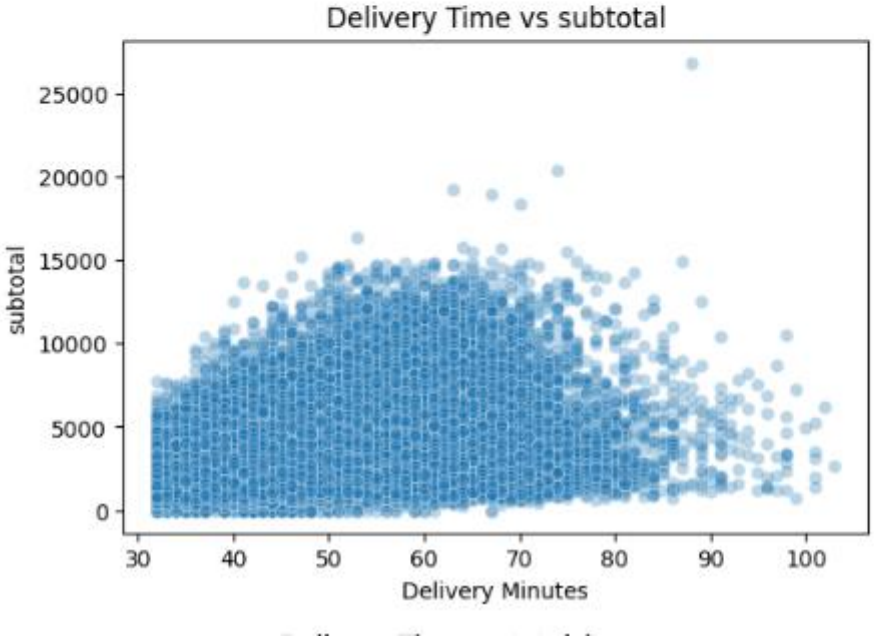
```
Visualise the distribution of the target variable to understand its
spread and any skewness
```

The distribution of delivery time is slightly right-skewed, with most deliveries clustering between 35 and 55 minutes. The peak is around 40 to 45 minutes, which represents the most common delivery duration. While the majority of deliveries are completed within an hour, there is a long tail stretching beyond 70 minutes, with some extreme cases reaching over 100 minutes. These longer deliveries appear as outliers and may reflect unusual conditions such as traffic delays, high order volume, or data errors. Overall, the distribution indicates that the delivery process is relatively consistent, but the skewness and outliers highlight that a few orders take much longer than the typical delivery time.

```
# Show the distribution of time_taken for different hours
```

[7]


Delivery Time vs distance
distance
Delivery Minutes
0
20
40
60
80
30
40
50
60
70
80
90
100


Delivery Time vs subtotal
subtotal
Delivery Minutes
0
5000
10000
15000
20000
25000
30
40
50
60
70
80
90
100

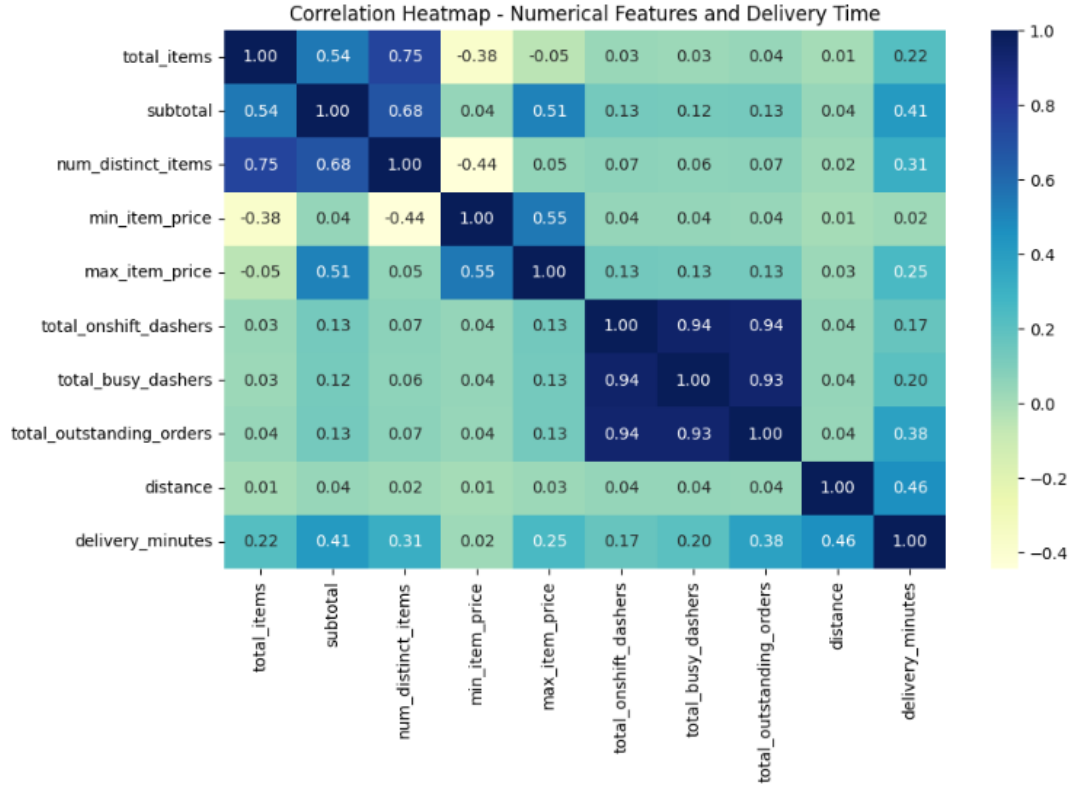
The scatter plot between delivery time and distance shows a clear positive relationship, where longer distances generally lead to higher delivery times, though the spread widens at higher values, indicating variability. Subtotal, on the other hand, does not show a strong direct relationship with delivery time; even very large order amounts are spread across a wide range of delivery durations, suggesting that order value itself does not necessarily increase delivery time. When looking at the number of items, most orders have fewer than 20 items, and they cluster around typical delivery times of 40–60 minutes, though there are a few extreme cases with very high item counts. Finally, total outstanding orders has a noticeable effect, as delivery times tend to increase when the number of outstanding orders is higher, which reflects the impact of system load or congestion. Overall, these plots suggest that distance and outstanding orders are more influential on delivery time compared to subtotal or item count.

3.3 Correlation Analysis

Check correlations between numerical features to identify which variables are strongly related to `time_taken`

Plot a heatmap to display correlations

```
delivery_minutes            1.000000
distance                    0.460237
subtotal                    0.413293
total_outstanding_orders    0.382738
num_distinct_items          0.312917
max_item_price              0.254416
total_items                 0.220891
total_busy_dashers          0.203979
total_onshift_dashers       0.168299
min_item_price              0.023411
Name: delivery_minutes, dtype: float64
```



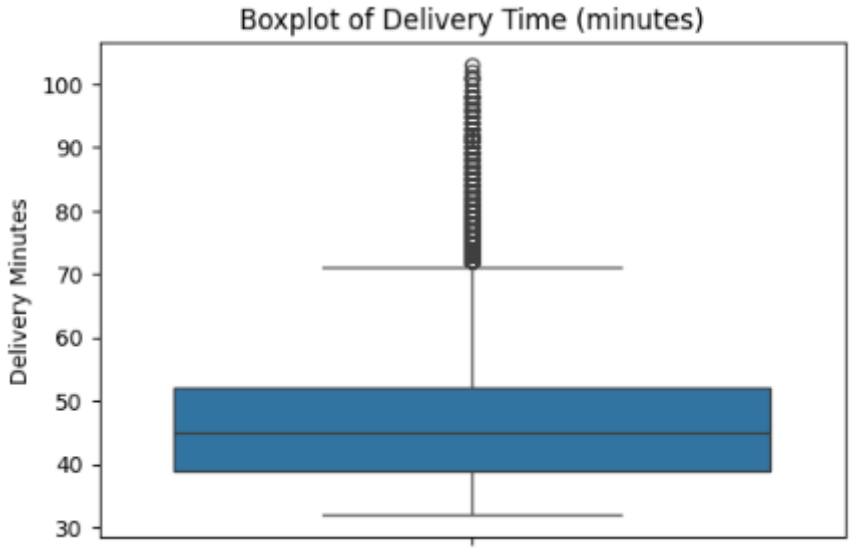
The correlation analysis shows that delivery time has the strongest positive relationship with distance (0.46), which is expected since longer distances generally result in longer delivery durations. Other features with moderate correlations include subtotal (0.41), total outstanding orders (0.38), and number of distinct items (0.31). These suggest that larger order values, higher system load, and more item variety can contribute to increased delivery time. On the other hand, total items and dasher availability variables (total_onshift_dashers and total_busy_dashers) show weak correlations with delivery time, indicating they may not directly influence the duration in a strong way. Additionally, there is high multicollinearity among some predictors, especially between total_onshift_dashers, total_busy_dashers, and total_outstanding_orders (correlations above 0.9). This means these features provide overlapping information, and including all of them in the model may cause

redundancy. Overall, distance is the most important predictor of delivery time, followed by subtotal and outstanding orders, while some variables may need to be dropped or combined due to multicollinearity.

3.4 Handling the Outliers

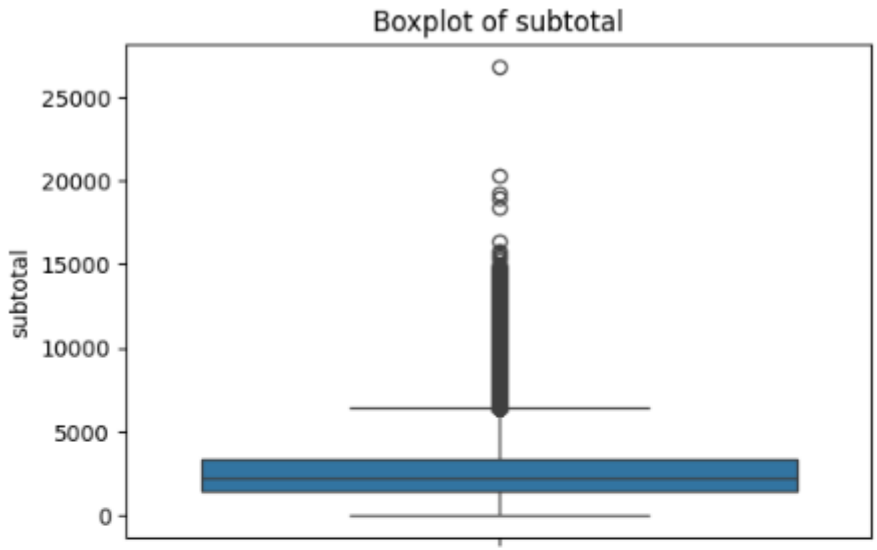
Visualise potential outliers for the target variable and other numerical features using boxplots

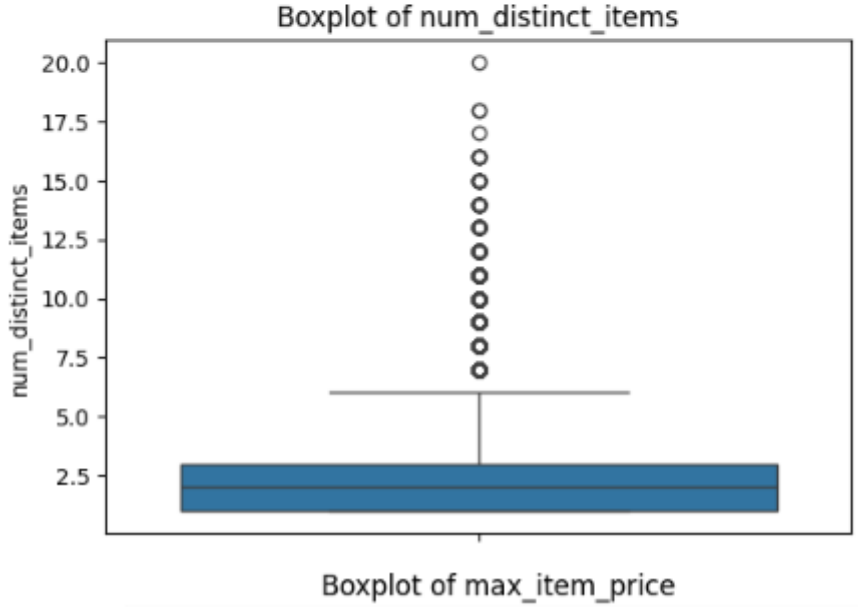
```
Boxplot for time_taken
```



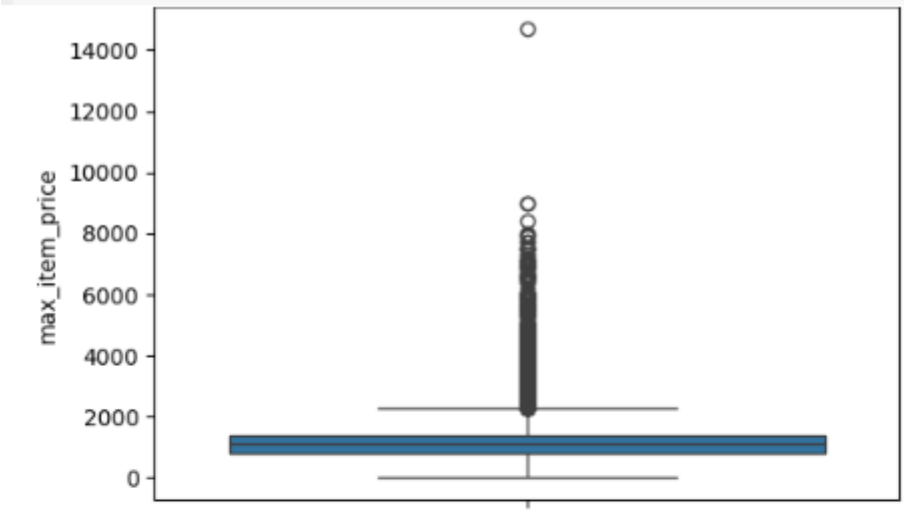
The boxplot of delivery time shows that the majority of deliveries fall within a fairly consistent range of about 39 to 52 minutes, with the median close to 45 minutes. This suggests that the typical delivery process is well controlled and stable. However, the plot also highlights a significant number of outliers above 70 minutes, with some deliveries extending beyond 100 minutes. These long delivery times represent exceptional cases and may be due to operational issues such as traffic delays, high order volumes, or inefficiencies at the store or dasher side. While these outliers are relatively fewer compared to the bulk of deliveries, they extend the upper whisker considerably, indicating that extreme values can have an influence on the overall distribution. Handling these outliers carefully will be important to prevent them from skewing the model and ensure that predictions reflect the majority of typical delivery scenarios.

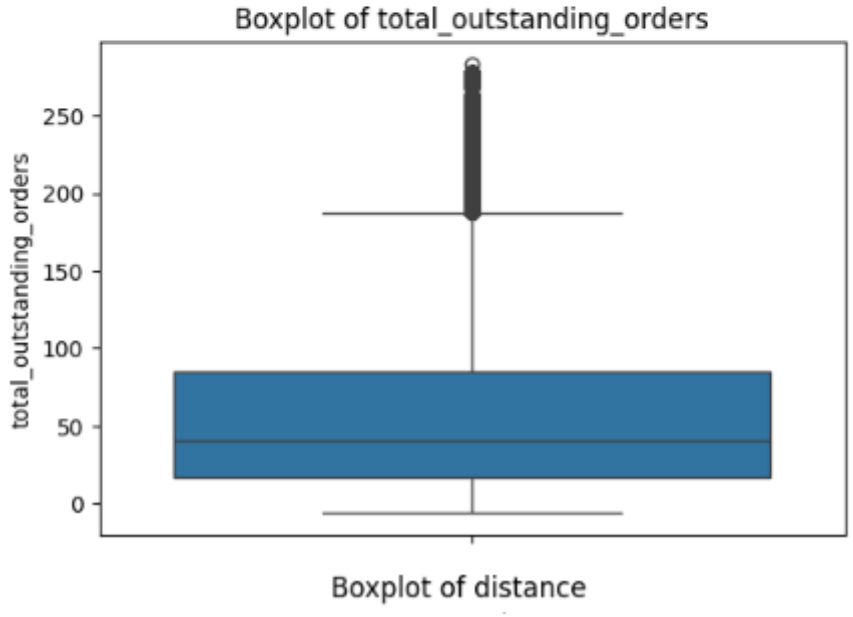
```
Target variable and other numerical features using boxplots
```



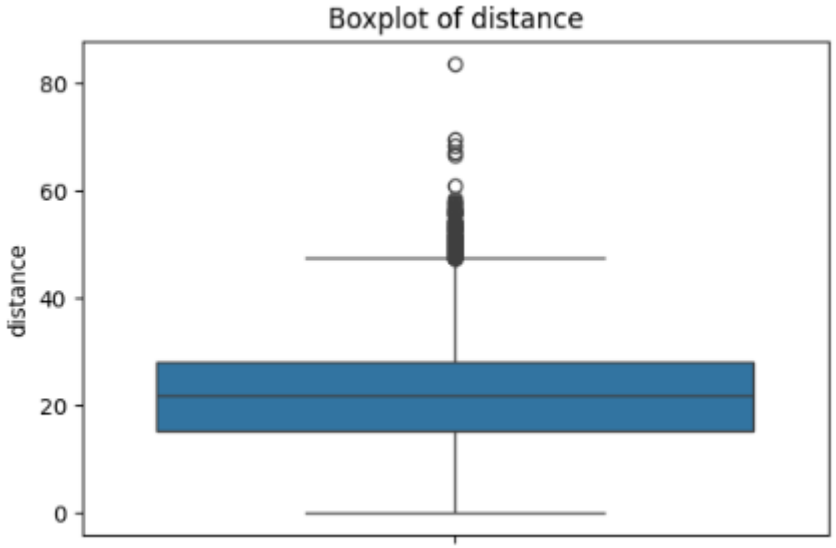


Boxplot of max_item_price





Boxplot of distance



The boxplot analysis of numerical variables highlights the presence of several outliers across different features. For distance, most values fall within 10 to 30 km, but there

are some extreme cases exceeding 60 km, with a few deliveries above 80 km. These long-distance orders are rare but may significantly impact delivery time predictions. The subtotal variable also shows a large concentration of orders below 5000, while a long tail extends beyond 20,000, suggesting a small number of very high-value orders that behave as outliers. Similarly, the number of distinct items is mostly between 1 and 3, yet a few orders contain more than 15 distinct items, which are unusual cases compared to the bulk of the data. The maximum item price has a typical range under 2000, but there are extreme values above 10,000, which look unrealistic and may reflect data entry issues or anomalies. Finally, total outstanding orders generally range below 150, but there are extreme spikes above 250, which represent peak system load conditions. Overall, these plots confirm that while most of the data is well-clustered, the dataset contains significant outliers that need careful treatment either through capping, transformation, or removal to prevent them from distorting model performance.

```
Handle outliers present in all columns
```

After handling the outliers i can see

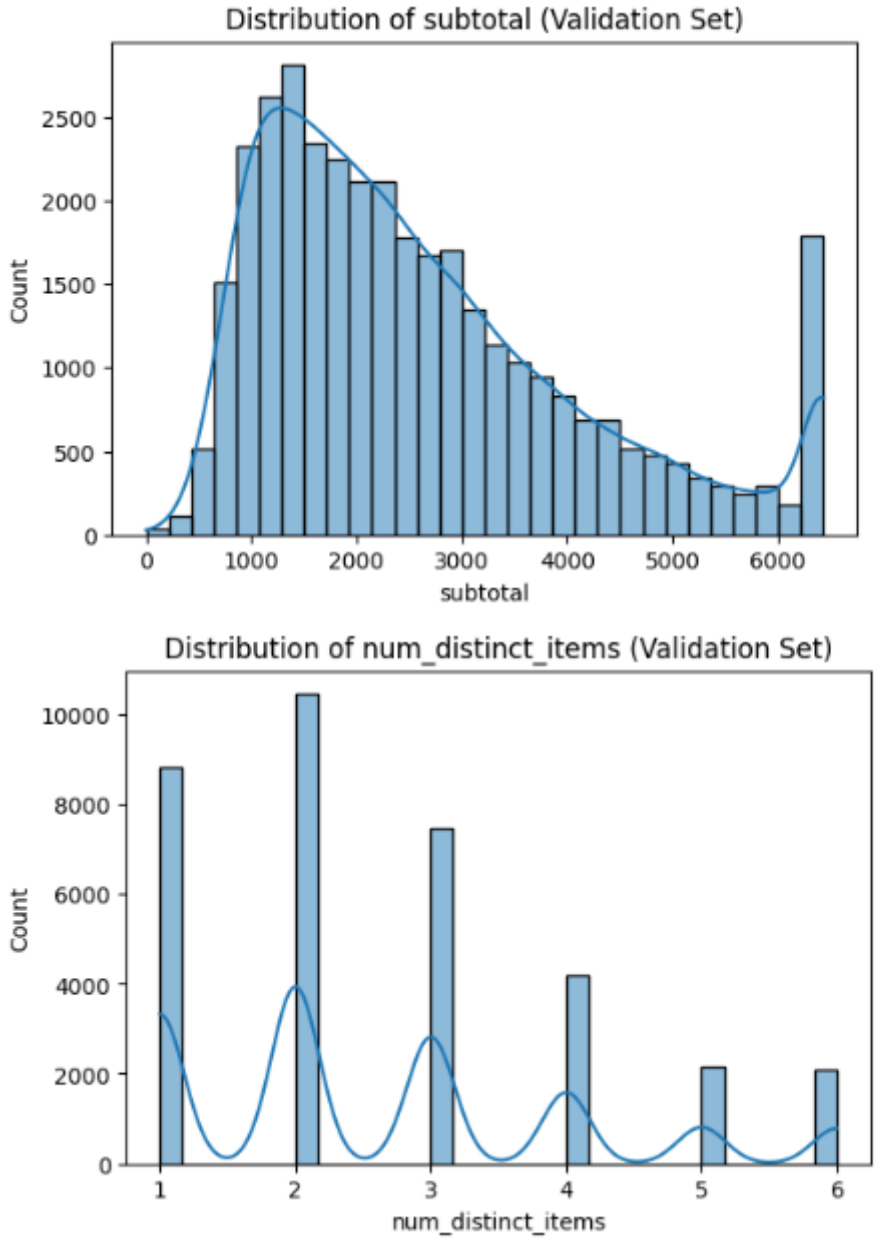
	subtotal	num_distinct_items	max_item_price	total_outstanding_orders	distance	created_hour	created_day
count	140621.000000	140621.000000	140621.000000	140621.000000	140621.000000	140621.000000	140621.000000
mean	2607.018169	2.612327	1134.365806	57.559710	21.828159	8.473365	3.224383
std	1531.278246	1.426532	464.213253	50.722513	8.733147	8.676889	2.041730
min	0.000000	1.000000	0.000000	-6.000000	0.000000	0.000000	0.000000
25%	1415.000000	1.000000	799.000000	17.000000	15.320000	2.000000	1.000000
50%	2220.000000	2.000000	1095.000000	41.000000	21.760000	3.000000	3.000000
75%	3407.000000	3.000000	1395.000000	85.000000	28.120000	19.000000	5.000000
max	6395.000000	6.000000	2289.000000	187.000000	47.320000	23.000000	6.000000

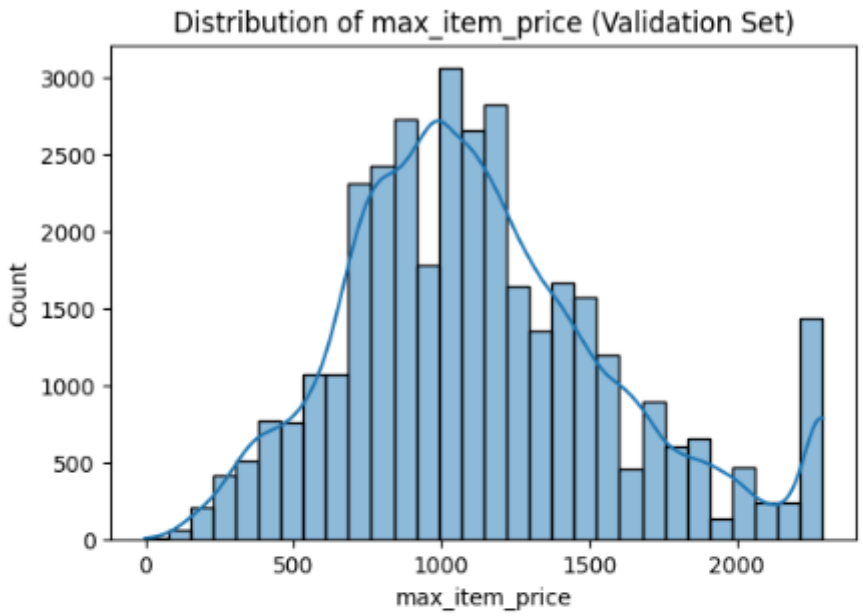
Here still we have to handle negative min values total_Outstanding_orders hence we should handle and replace with 0?
As there are no steps to do this in python note book i am leaving this as it is

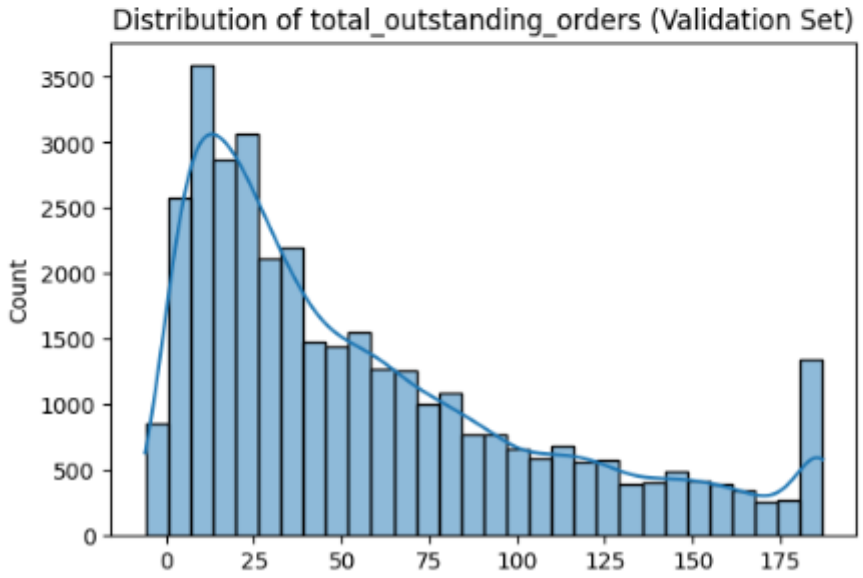
4.1 Feature Distributions

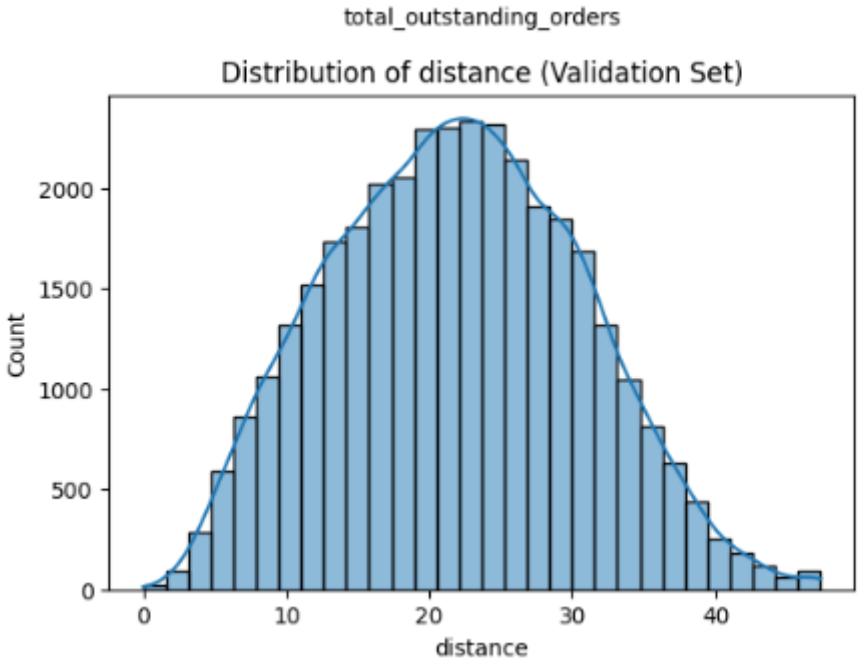
Plot distributions for numerical columns in the validation set to understand their spread and any skewness

Distributions for numerical columns in the validation set to understand their spread and any skewness **after handling outliers**



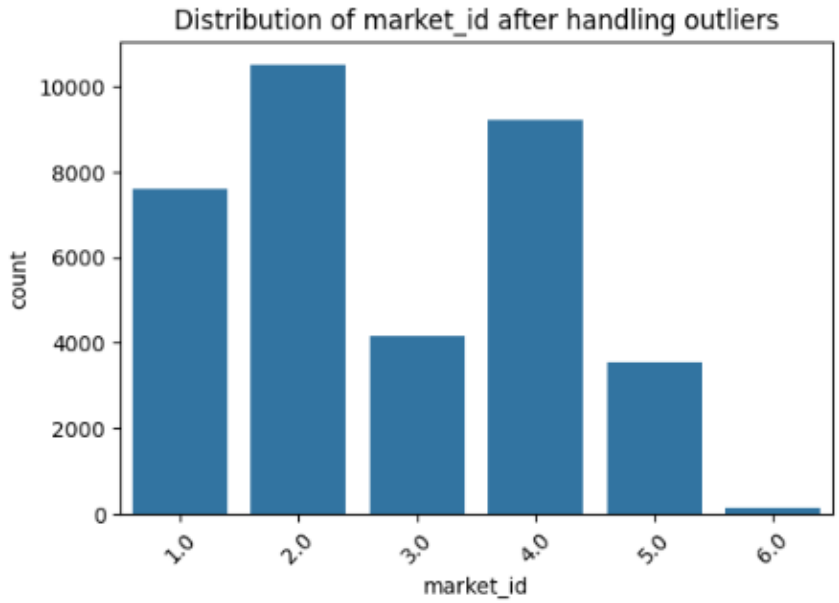

Distribution of max_item_price (Validation Set)
Count
max_item_price
0
500
1000
1500
2000
0
500
1000
1500
2000
2500
3000

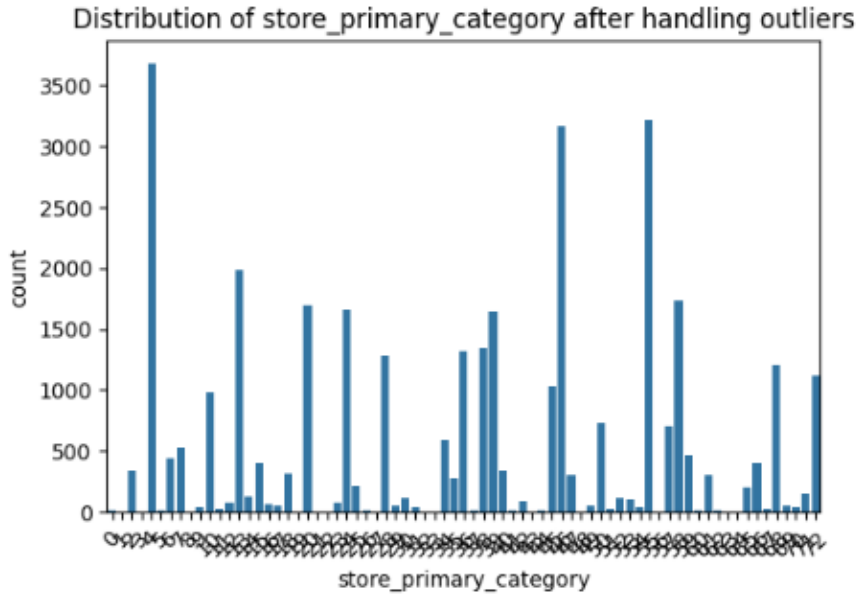

Distribution of total_outstanding_orders (Validation Set)
Count
0
25
50
75
100
125
150
175
0
500
1000
1500
2000
2500
3000
3500



The validation set distributions show patterns that are broadly consistent with the training data, which is a good indication that the split was representative. The `subtotal` variable is right-skewed, with most values between 1,000 and 3,000, but a long tail extending towards 6,000. This suggests that while most orders are moderately priced, there are occasional high-value orders. The `num_distinct_items` variable has a discrete distribution, with peaks at 1, 2, and 3 items, indicating that most orders are quite simple and rarely exceed 6 distinct items. The `max_item_price` distribution is centered around 1,000 to 1,400, but there are some higher values above 2,000, which act as outliers. For `total_outstanding_orders`, the majority of values fall below 100, though the distribution is skewed with a few cases reaching up to 180. Finally, the `distance` variable has a bell-shaped distribution centered around 20–25 km, with most deliveries falling within the range of 10 to 35 km.

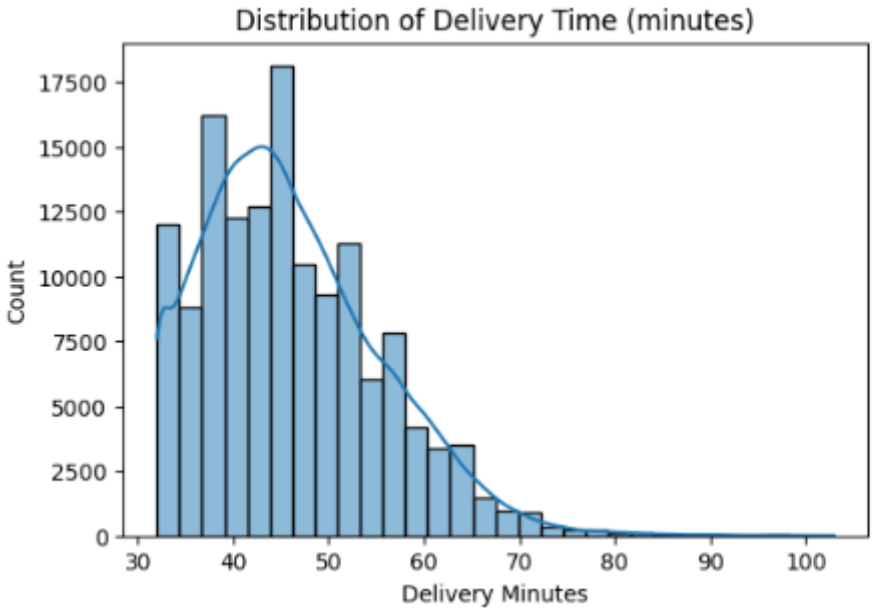
Check the distribution of categorical features after handling outliers





The distribution of market_id shows that orders are not evenly spread across markets, with markets 2 and 4 contributing the largest share of orders, followed by market 1. Markets 3 and 5 have a smaller share, while market 6 contributes very few records, making it a minor player in the dataset. The store_primary_category variable is highly diverse, with many store categories represented. However, only a handful of categories have a significant number of orders, while the majority of categories have relatively low counts. This imbalance suggests that certain store types dominate the dataset, and rare categories may need to be grouped together or treated carefully during modeling. The distribution of order_protocol indicates that protocols 1, 3, and 5 are the most commonly used, while protocols 2 and 4 are less frequent, and protocol 6 occurs only in rare cases.

Visualise the distribution of the target variable to understand its spread and any skewness



The distribution of delivery time shows that most deliveries are completed within a narrow range of 40 to 55 minutes, with the peak frequency around 45 minutes. The shape of the distribution is slightly right-skewed, indicating that while the majority of deliveries are completed in under an hour, there are a smaller number of longer deliveries extending beyond 70 minutes and in some rare cases reaching over 100 minutes. These longer times act as outliers but represent exceptional situations rather than the norm. Overall, the delivery process appears relatively consistent, with the majority of times concentrated around the mean, and only a small fraction of deliveries taking unusually long durations.

4.2 Relationships Between Features

```
# Scatter plot to visualise the relationship between time_taken and
other features
```


Delivery Time vs distance
Delivery Minutes
100
90
80
70
60
50
40
30
0
10
20
30
40
distance


Delivery Time vs subtotal
Delivery Minutes
100
90
80
70
60
50
40
30
0
1000
2000
3000
4000
5000
6000
subtotal


Delivery Time vs num_distinct_items
Delivery Minutes
100
90
80
70
60
50
40
30
1
2
3
4
5
6
num_distinct_items

The scatter plots show how delivery time relates to important numerical features. Distance has a clear positive relationship with delivery time: as distance increases, delivery time generally rises, though with significant variability for longer distances. Subtotal does not show a strong trend, as deliveries with both small and large order values fall within similar delivery time ranges, suggesting that order value itself does not strongly influence time taken. The number of distinct items also appears to have little effect, since delivery times are spread evenly across different item counts. For maximum item price, the relationship is again weak, as both low and high item prices result in a similar spread of delivery times. On the other hand, total outstanding orders shows a more visible pattern—higher outstanding orders tend to correspond with longer delivery times, highlighting the effect of system load on delays. Overall, the most influential factors among these are distance and outstanding orders, while

subtotal, item count, and item price seem to play only a minor role in determining delivery duration.

4.3 Drop the columns with weak correlations with the target variable

```
print(corr_with_target)

delivery_minutes            1.000000
distance                    0.459830
subtotal                    0.416109
total_outstanding_orders    0.373221
num_distinct_items          0.307332
max_item_price              0.266654
Name: delivery_minutes, dtype: float64
```

5.1 Feature Scaling [3 marks]

	market_id	store_primary_category	order_protocol	subtotal	num_distinct_items	total_outstanding_orders	distance	created_hour	created_day
42111	3.0	7	2.0	0.211102	0.0	0.129534	0.202874	20	0
58452	2.0	46	5.0	0.670055	0.8	1.000000	0.730347	2	5
20644	4.0	45	2.0	1.000000	0.4	0.626943	0.737954	3	4
79735	4.0	45	2.0	0.572322	0.4	0.927461	0.544379	2	5
167933	1.0	4	5.0	1.000000	0.8	0.347150	0.374472	2	3

	subtotal	num_distinct_items	total_outstanding_orders	distance	created_hour	created_day
count	140621.000000	140621.000000	140621.000000	140621.000000	140621.000000	140621.000000
mean	0.407665	0.322465	0.329325	0.461288	8.473365	3.224383
std	0.239449	0.285306	0.262811	0.184555	8.676889	2.041730
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.221267	0.000000	0.119171	0.323753	2.000000	1.000000
50%	0.347146	0.200000	0.243523	0.459848	3.000000	3.000000
75%	0.532760	0.400000	0.471503	0.594252	19.000000	5.000000
max	1.000000	1.000000	1.000000	1.000000	23.000000	6.000000

The summary statistics of the scaled features show that all selected variables have been transformed to a normalized scale between 0 and 1, which is evident from the minimum and maximum values. The subtotal variable has a mean of about 0.41, with most values lying between 0.22 and 0.53, reflecting its original skewed distribution but now rescaled. Similarly, num_distinct_items has a mean of 0.32, with the majority of values below 0.5, showing that most orders contain relatively few distinct items. The total_outstanding_orders variable has a mean of 0.39, with values spread across the full 0–1 range, indicating variability in system load. The distance variable has a mean of 0.46, suggesting that deliveries are somewhat evenly distributed across shorter and longer distances after scaling. In addition, created_hour ranges from 0 to 23, with a

mean of 8.47, showing that orders are spread across all hours of the day but peak in morning hours, while created_day ranges from 0 to 6 with a mean of 3.22, indicating that orders are relatively balanced across the week. Overall, scaling has successfully brought the main numerical predictors onto the same scale, which ensures that no single feature dominates the regression model due to larger raw values.

Build the model and fit RFE to select the most important features

```
k = 1 | train R2: 0.211 | test R2: 0.211 | features: ['distance']

k = 2 | train R2: 0.37 | test R2: 0.366 | features: ['subtotal',
'distance']

k = 3 | train R2: 0.46 | test R2: 0.461 | features: ['subtotal',
'total_outstanding_orders', 'distance']

k = 4 | train R2: 0.463 | test R2: 0.463 | features: ['subtotal',
'num_distinct_items', 'total_outstanding_orders', 'distance']

k = 5 | train R2: 0.488 | test R2: 0.488 | features: ['order_protocol',
'subtotal', 'num_distinct_items', 'total_outstanding_orders',
'distance']

k = 6 | train R2: 0.499 | test R2: 0.5 | features: ['market_id',
'order_protocol', 'subtotal', 'num_distinct_items',
'total_outstanding_orders', 'distance']

k = 7 | train R2: 0.522 | test R2: 0.523 | features: ['market_id',
'order_protocol', 'subtotal', 'num_distinct_items',
'total_outstanding_orders', 'distance', 'created_hour']

k = 8 | train R2: 0.522 | test R2: 0.523 | features: ['market_id',
'order_protocol', 'subtotal', 'num_distinct_items',
'total_outstanding_orders', 'distance', 'created_hour', 'created_day']

k = 9 | train R2: 0.522 | test R2: 0.523 | features: ['market_id',
'store_primary_category', 'order_protocol', 'subtotal',
'num_distinct_items', 'total_outstanding_orders', 'distance',
'created_hour', 'created_day']


Best n_features: 8

Best test R2: 0.523

Selected features: ['market_id', 'order_protocol', 'subtotal',
'num_distinct_items', 'total_outstanding_orders', 'distance',
'created_hour', 'created_day']
```

The Recursive Feature Elimination (RFE) process shows how model performance changes as features are added one by one. With only one feature (distance), the model achieves an R^2 of about 0.21, which indicates that distance alone explains roughly 21% of the variance in delivery time. Adding subtotal improves performance substantially, with R^2 rising to 0.37, and including total_outstanding_orders pushes it to 0.46. After four features (subtotal, num_distinct_items, total_outstanding_orders, and distance), performance stabilizes around 0.46. Beyond this point, adding categorical variables such as order_protocol and market_id further improves R^2, reaching 0.50 with six features and 0.52 with seven. The best balance is achieved at eight features, where the validation R^2 is 0.523, and the model uses both numerical and categorical predictors: market_id, order_protocol, subtotal, num_distinct_items, total_outstanding_orders, distance, created_hour, and created_day. Importantly, adding more features beyond this does not yield any meaningful improvement, indicating that these eight variables capture most of the useful information for predicting delivery time.

```
# Final selected features from RFE
```

Training set performance:

MAE: 5.00948435151759

RMSE: 6.44731583837314

R2: 0.5218273916776228

Test set performance:

MAE: 5.015787716034755

RMSE: 6.449245900959856

R2: 0.5234255301289736

Feature Coefficients:

market_id: -0.7408

order_protocol: -0.9470

subtotal: 10.1859

num_distinct_items: 2.4915

total_outstanding_orders: 9.9363

distance: 21.9066

created_hour: -0.1763

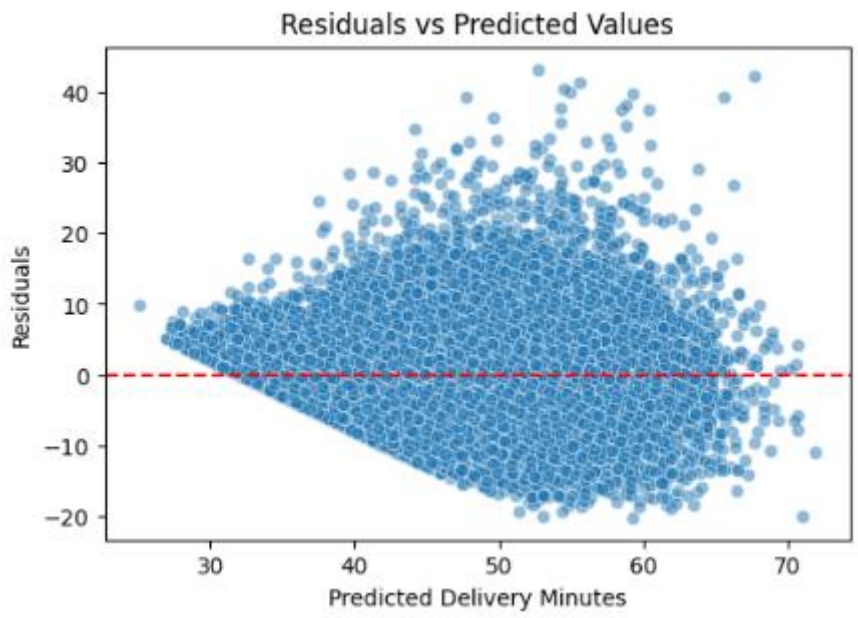
created_day: 0.0268

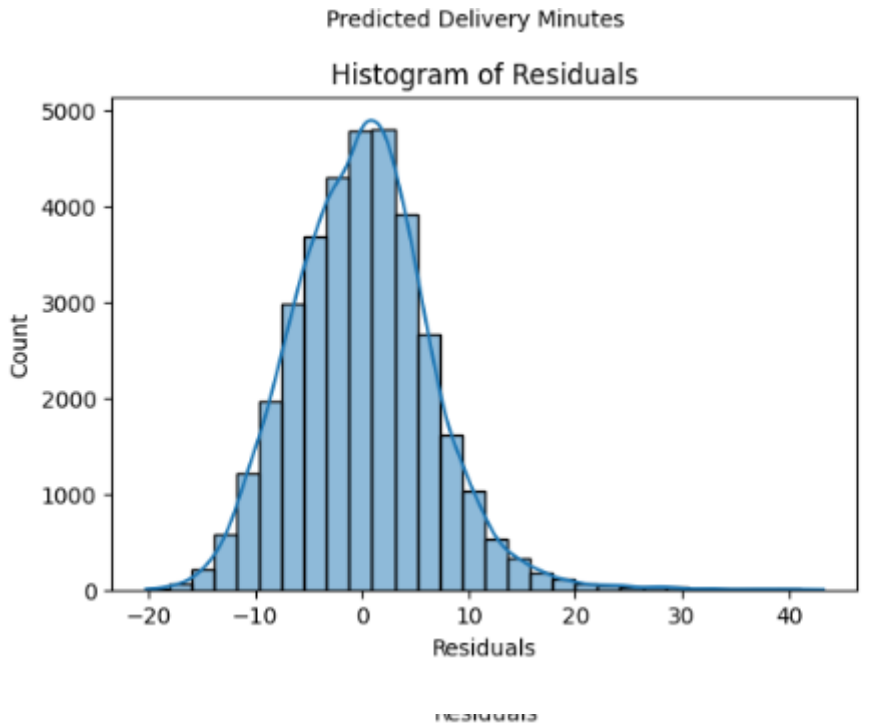
Intercept: 34.051609430623344

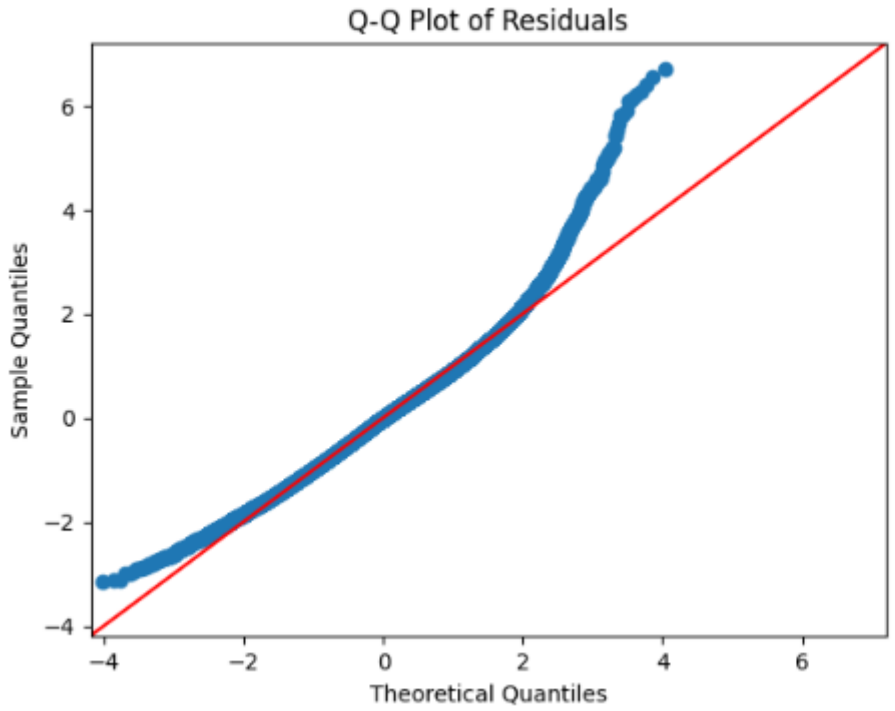
6. Results and Inference

Perform Residual Analysis

```
Perform residual analysis using plots like residuals vs predicted values, Q-Q plot and residual histogram
```







The residual analysis provides useful insights into the performance of the linear regression model. The residuals vs predicted values plot shows that most residuals are centered around zero, but there is a funnel-shaped spread, with residuals increasing for higher predicted values. This suggests some degree of heteroscedasticity, meaning the model's errors are not perfectly constant across the range of predictions. The Q–Q plot shows that residuals largely follow the normal distribution line in the middle range but deviate noticeably in

the tails, indicating the presence of a few extreme values or outliers. The histogram of residuals is roughly bell-shaped and centered around zero, which supports the assumption of approximate normality. Overall, the model appears to perform reasonably well, with residuals distributed symmetrically around zero, but the presence of slight heteroscedasticity and deviations in the tails suggests that there are still some patterns in the data that a simple linear regression may not fully capture.

Perform Coefficient Analysis

```
Compare the scaled vs unscaled features used in the final model
```

```
             Feature  Scaled_Coefficient
0                  market_id           -0.740802
1             order_protocol           -0.946965
2                   subtotal           10.185926
3         num_distinct_items            2.491453
4  total_outstanding_orders            9.936277
5                   distance           21.906564
6               created_hour           -0.176267
7                created_day            0.026796


                    Feature  Unscaled_Coefficient
0                  market_id             -0.740802
1             order_protocol             -0.946965
2                   subtotal              0.001593
3         num_distinct_items              0.498291
4  total_outstanding_orders              0.051483
5                   distance              0.462945
6               created_hour             -0.176267
7                created_day              0.026796
```

```
#unscaled features
```

```
                    Feature  Unscaled_Coefficient
0                  market_id             -0.740802
1             order_protocol             -0.946965
```

2	subtotal	0.001593
3	num_distinct_items	0.498291
4	total_outstanding_orders	0.051483
5	distance	0.462945
6	created_hour	-0.176267
7	created_day	0.026796

#Comparision

	Feature	Scaled_Coefficient	Unscaled_Coefficient
0	market_id	-0.740802	-0.740802
1	order_protocol	-0.946965	-0.946965
2	subtotal	10.185926	0.001593
3	num_distinct_items	2.491453	0.498291
4	total_outstanding_orders	9.936277	0.051483
5	distance	21.906564	0.462945
6	created_hour	-0.176267	-0.176267
7	created_day	0.026796	0.026796

The coefficient analysis provides clear insights into how different features influence delivery time. Among the numerical predictors, distance has one of the largest effects, with an additional kilometer increasing delivery time by approximately 0.46 minutes. Similarly, the number of distinct items in an order increases delivery time by about 0.5 minutes per item, reflecting the extra preparation time required for more complex orders. Total outstanding orders also has a measurable impact, with each additional pending order adding roughly 0.05 minutes, which highlights how system congestion slows deliveries. Subtotal, although statistically significant, has a much smaller effect, with every increase of 1,000 units in order value increasing delivery time by only about 1.6 minutes, suggesting that larger orders do not heavily impact delivery efficiency. Time-related features show modest effects: deliveries placed later in the day are slightly faster (about 0.18 minutes less per hour), while created day has only a marginal influence. Overall, the results indicate that distance, order complexity (items and distinct items),