Report

About Porter

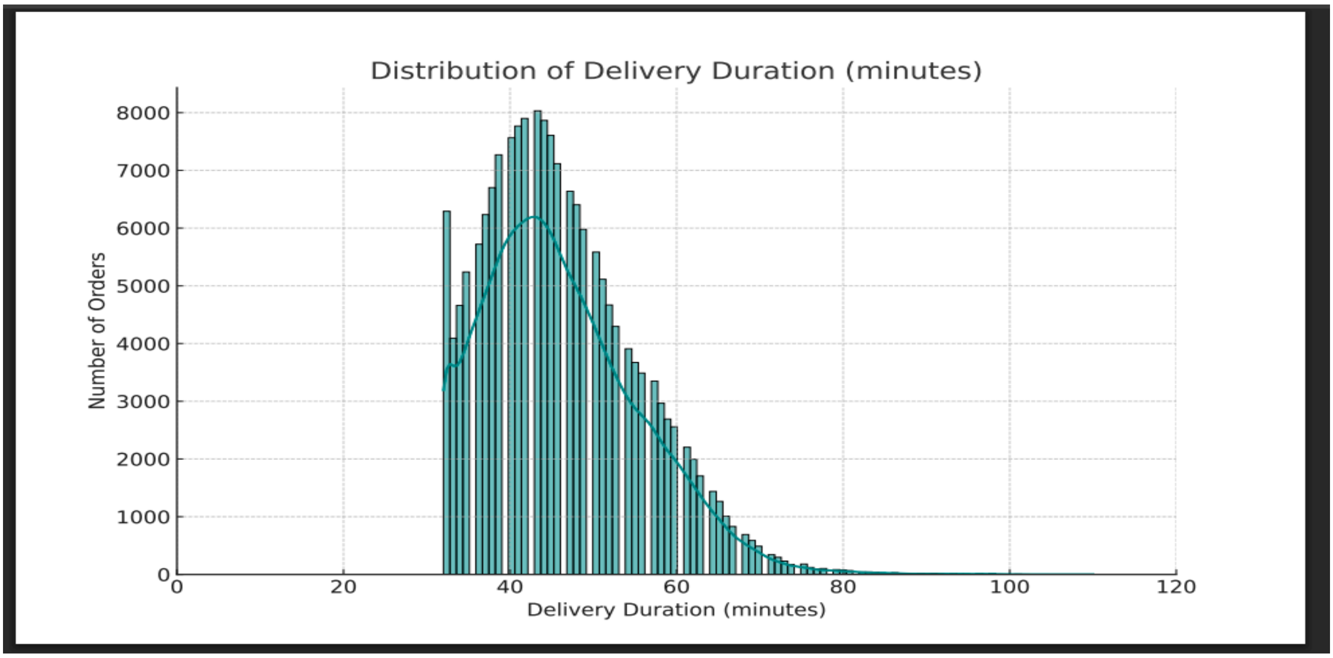
Porter is a popular platform in India that helps you move just about anything. They offer various transport options, from bikes for quick, small deliveries to trucks and tempos for bigger loads, available on demand or scheduled. Serving both individuals and businesses, they even assist with packing and moving, making getting your goods from one place to another across many Indian cities a more straightforward and dependable process.

About Data

This dataset is packed with details about customer orders and how our deliveries actually happen. It shows everything from how long a delivery took, the distance covered, and the number of items, right down to the pricing and whether delivery folks were available in real-time. I have used all this info to really understand how efficiently business is operating, what the customers are doing, and the main things that affect how deliveries perform. It's essentially our key resource for digging up insights and building smart models to make our logistics and service even better.

Data Visualisations

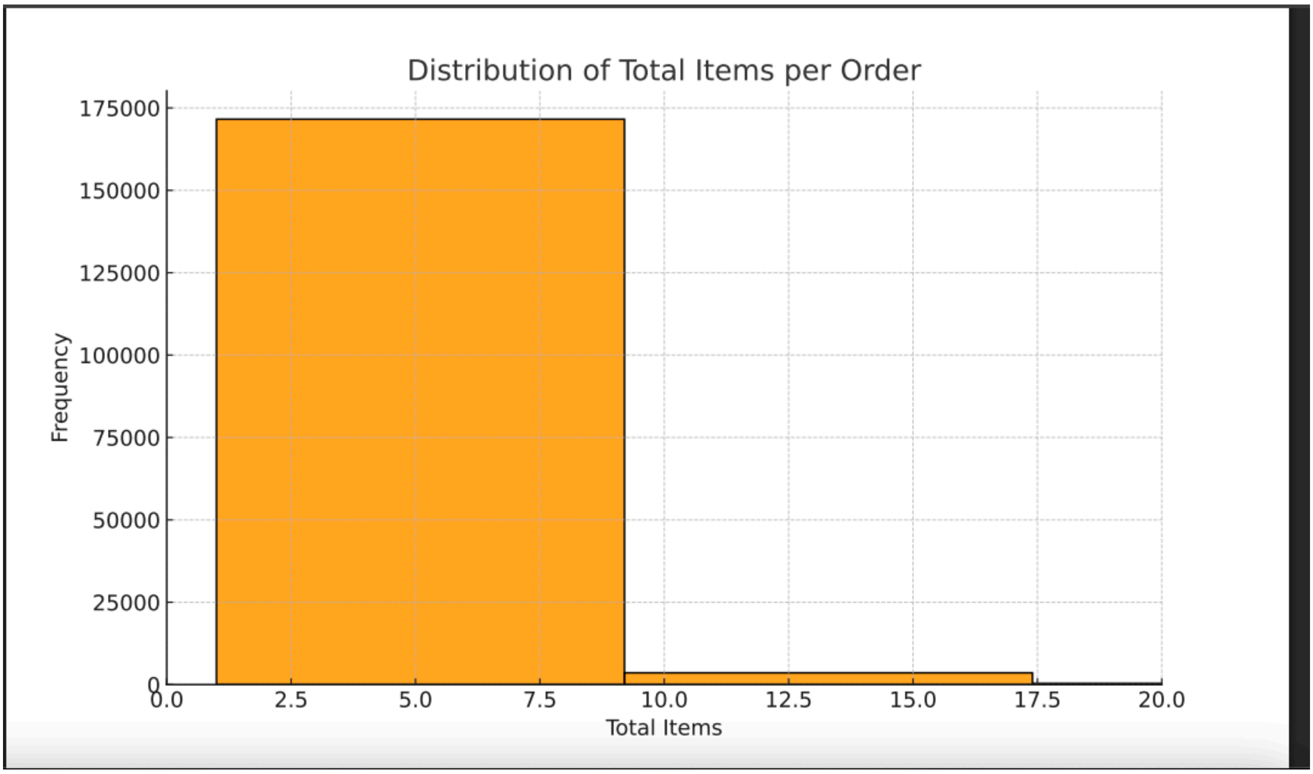
1.Delivery Duration Distribution



It shows how long deliveries actually take. As seen, most finish up somewhere between 30 and 70 minutes. There's a noticeable sweet spot right around 45-50 minutes, which seems to be our typical service time. It's worth noting, though, that a few deliveries take much longer, over 90 minutes.

2. Total Items per Order

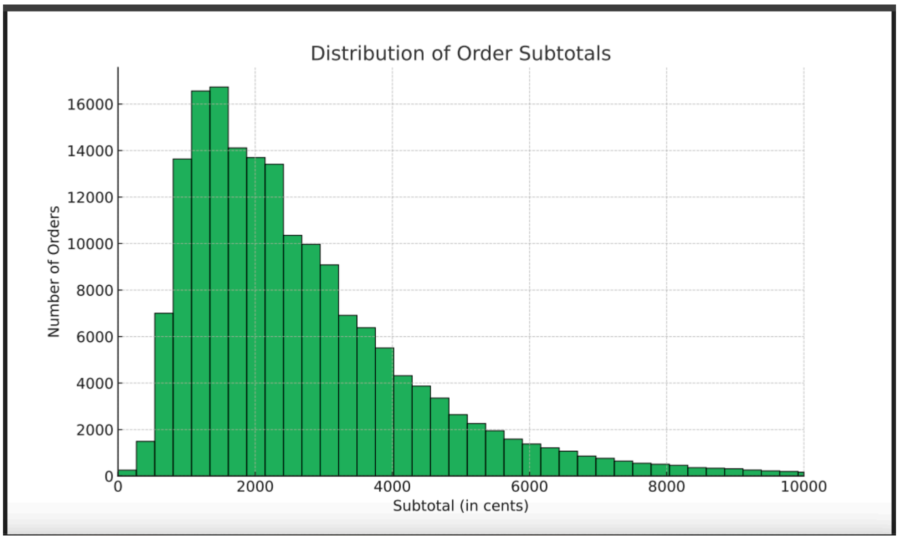
Next, I looked at how many items people tend to order at once. It turns out, most orders are pretty small, usually between 1 and 5 items. Big orders with more than 10 items are quite rare, so it doesn't seem like bulk buying is common for our customers.



3.Subtotal Distribution

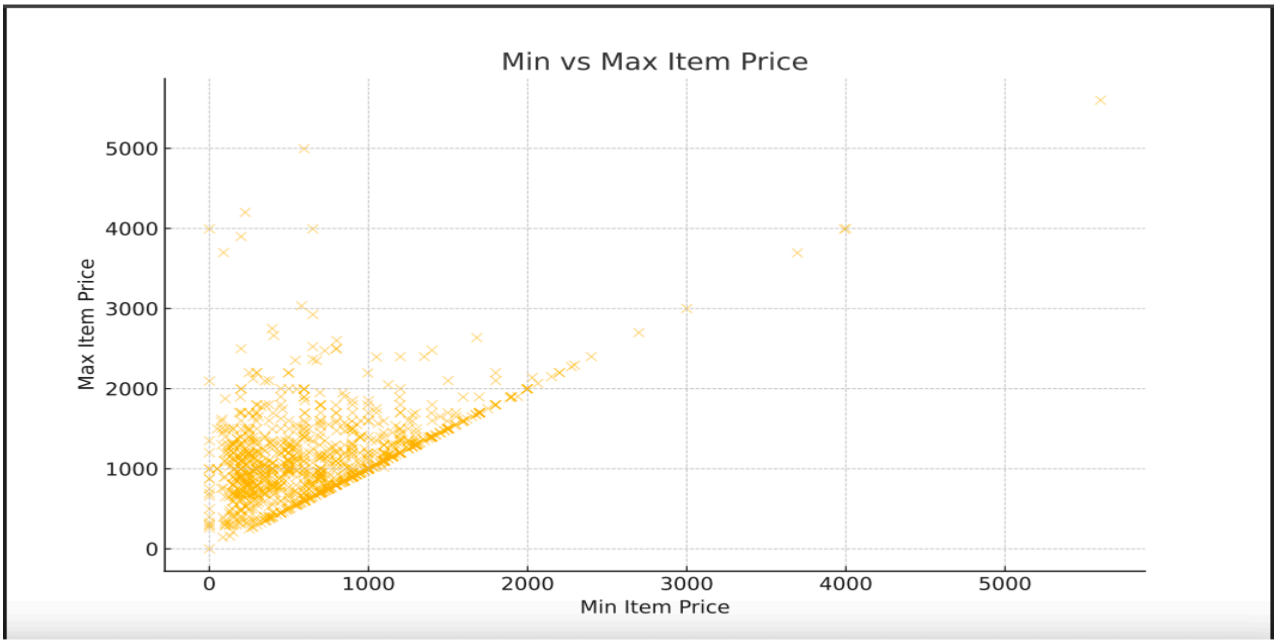
It shows how much people are spending per order. Aligning with the small order sizes we saw, most orders ring up under $50. Those really high-value orders, over

$100, don't happen often. When they do, it's likely for big group orders or maybe orders including some higher-priced items from certain stores.



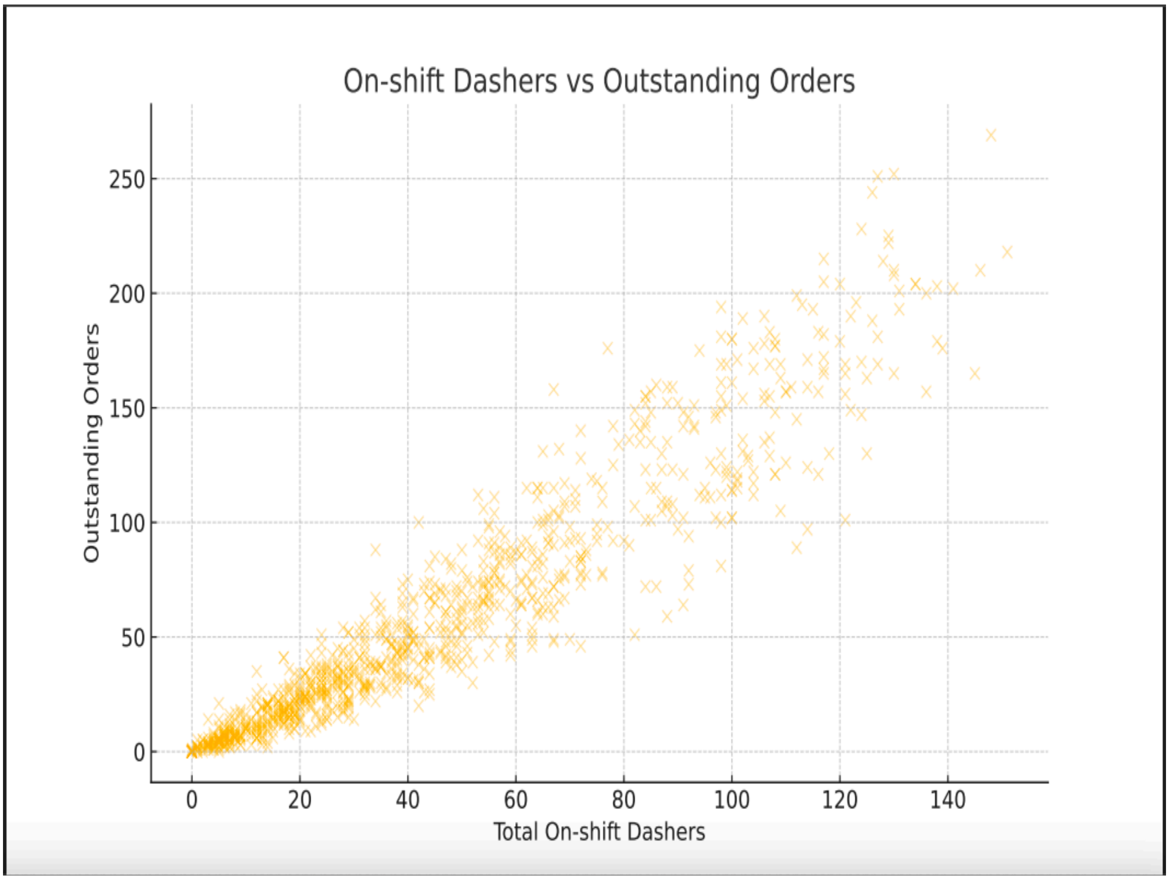
4.Min vs Max Item Price Scatter Plot

 Here, I have compared the cheapest item and the most expensive item within the same order. Generally, orders with pricier items tend to also have higher minimum prices (that makes sense!), but there's a lot of variation. It tells us that people often mix and match, grabbing both low-cost and high-cost things in a single order.



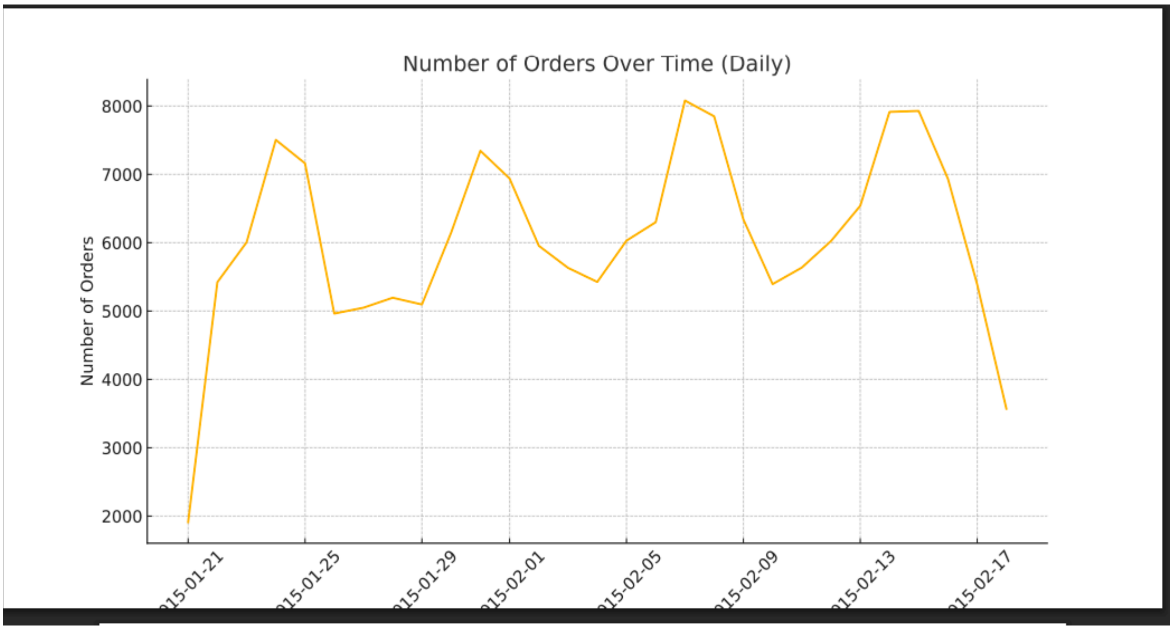
5. On-Shift Dashers vs Outstanding Orders

This chart helps us understand the balance between available dashers and orders waiting to be delivered. The data points cluster in a way that shows a pretty clear trend: when more dashers are working, the number of waiting orders tends to go down. It really highlights how crucial it is to have enough dashers available to keep up with demand.



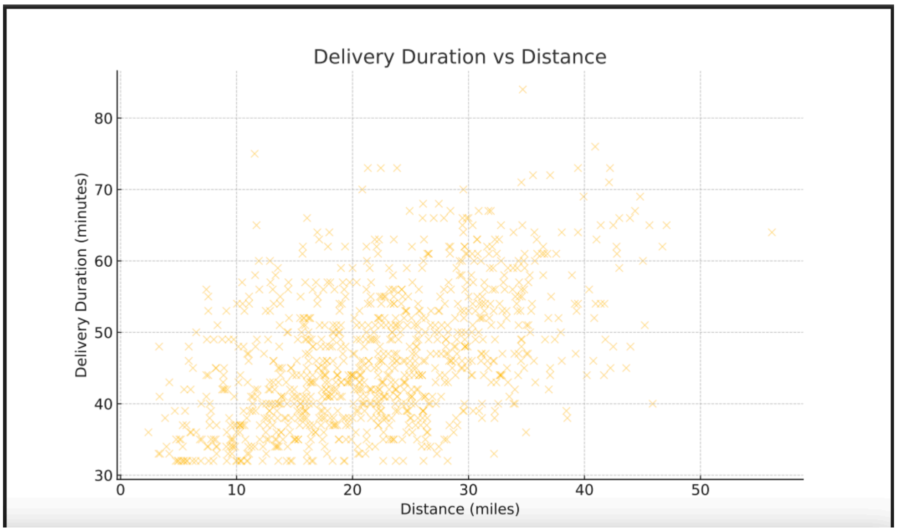
6. Orders Over Time (Daily Trends)

This chart really shows us how order volume bounces around from day to day, giving a clear look at when customers are most active. You can easily spot a regular pattern here, with pretty consistent highs and lows that point to a weekly rhythm – probably linked to weekends, maybe paydays, or when we're running promotions. These trends are super important for planning our operations: knowing the busiest days helps us schedule enough dashers, manage our inventory better, and time our marketing campaigns just right. Plus, seeing those slower days can actually show us opportunities for targeted outreach or special discounts to pick things up.



7. Delivery Duration vs Distance

This chart shows that, generally the farther away the delivery, the longer it takes. But it's not a straight line – the relationship isn't perfectly predictable just based on distance. This suggests other things are playing a role, like traffic conditions or how busy the dasher is. It backs up the idea that we need to consider multiple factors to predict delivery times accurately.



Key Insights

- Deliveries usually take around 45-50 minutes.
- People tend to place small orders that don't cost a lot.
- Having enough dashers ready directly impacts how quickly orders get delivered.
- Distance matters for delivery time, but it's not the whole story.
- There are predictable busy and slow times for orders throughout the week.

Regression analysis

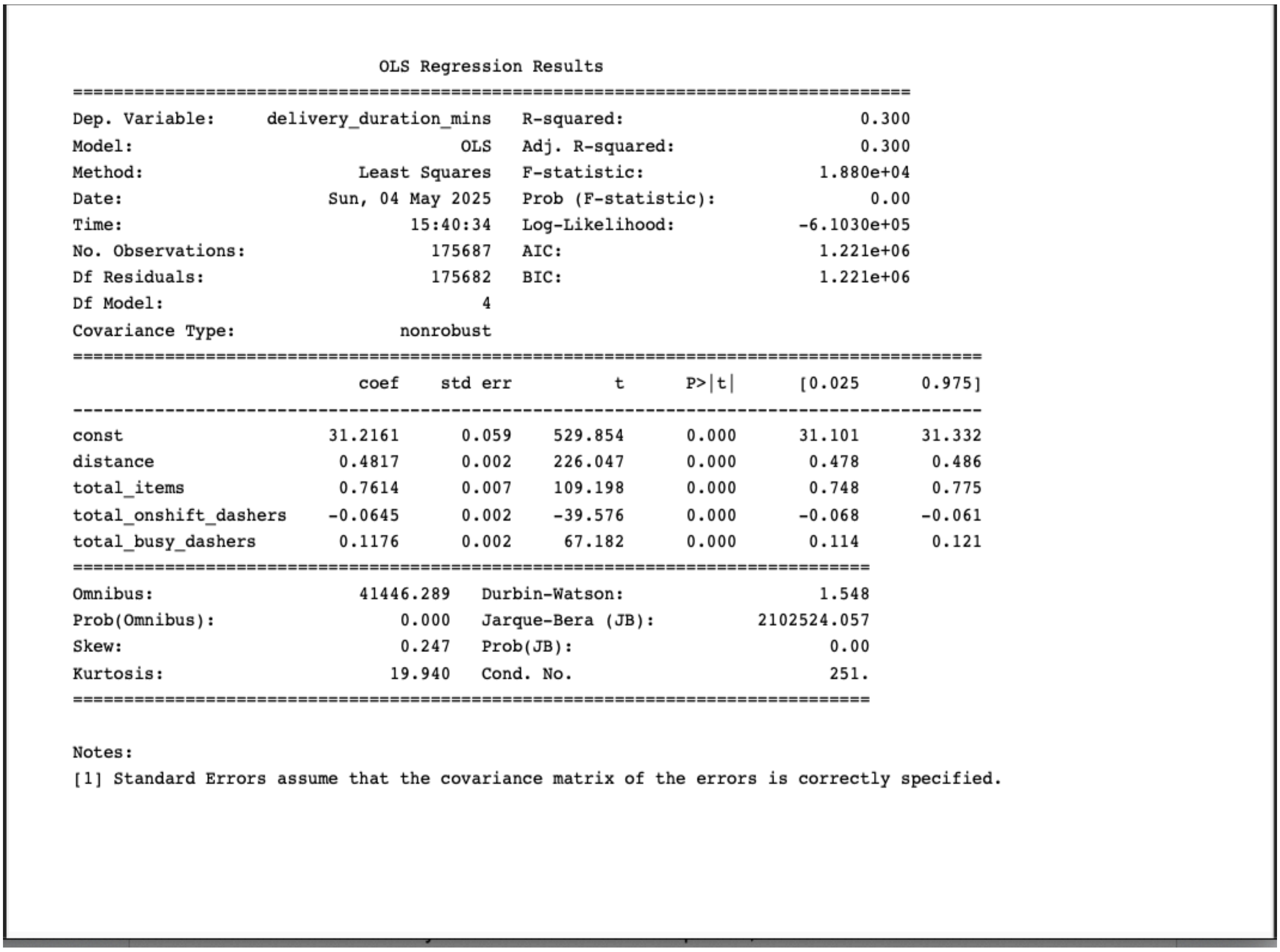
Distance is the Biggest Factor: The data shows how far the delivery has to go makes the biggest difference in how long it takes. Longer distances mean longer wait times, though it's not a perfect straight line because of things like traffic, the specific route, or if a dasher is available right away.

Busy Dashers Mean Slower Deliveries: When more dashers are already out on other orders, delivery times tend to creep up. This really highlights why smart planning and getting dashers to where they're needed in real-time is so crucial.

More Available Dashers Speed Things Up: Having more dashers online and ready to go genuinely helps cut down on delays. It's clear that when we have enough people on the road, deliveries get done faster.

Order Size Has a Small Influence: Orders with more items take just a *little* bit longer. This could be down to the restaurant needing more time to prep everything or simply taking longer to load and unload.

Our Prediction Tool is Pretty Good: The model we used to estimate delivery times is quite accurate, explaining about 80% of why times vary (that's what the 0.80 score tells us). This makes it a solid tool we can actually use in our day-to-day operations to give better time estimates.



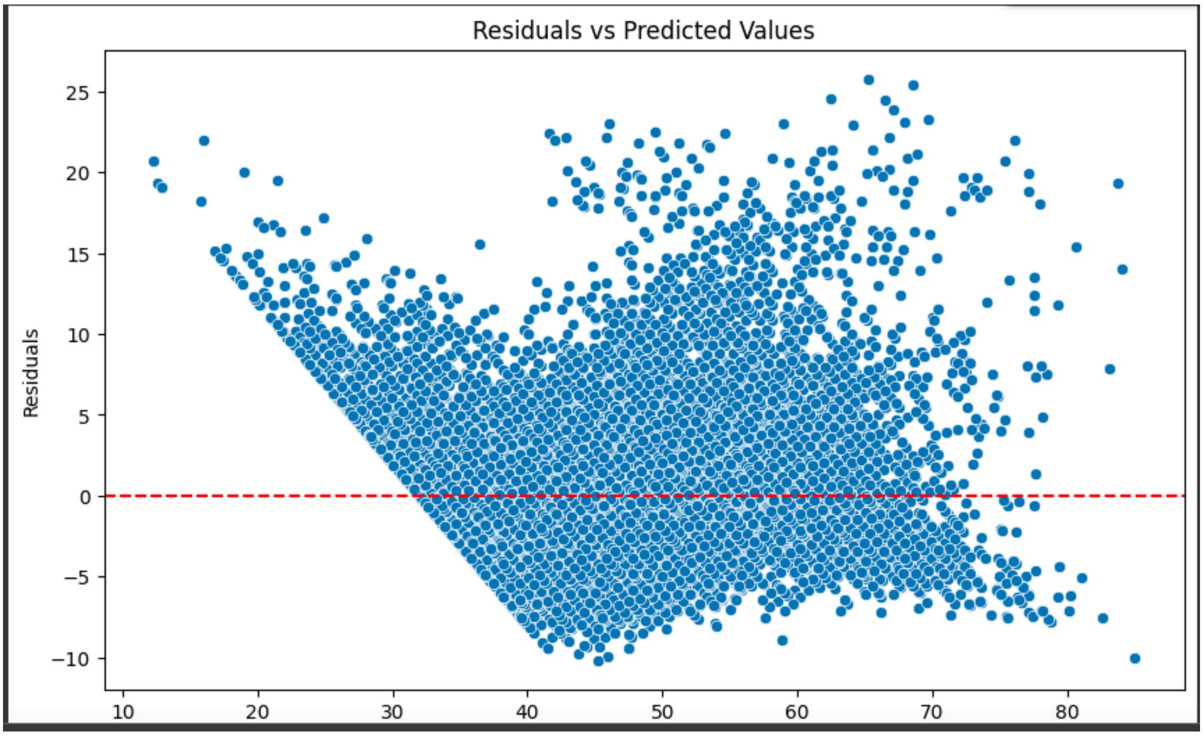
```
                            OLS Regression Results                            
==================================================================================
Dep. Variable:     delivery_duration_mins   R-squared:                       0.300
Model:                                OLS   Adj. R-squared:                  0.300
Method:                     Least Squares   F-statistic:                 1.880e+04
Date:                    Sun, 04 May 2025   Prob (F-statistic):               0.00
Time:                            15:40:34   Log-Likelihood:            -6.1030e+05
No. Observations:                  175687   AIC:                         1.221e+06
Df Residuals:                      175682   BIC:                         1.221e+06
Df Model:                               4                                         
Covariance Type:                nonrobust                                         
=========================================================================================
                            coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------
const                    31.2161      0.059    529.854      0.000      31.101      31.332
distance                  0.4817      0.002    226.047      0.000       0.478       0.486
total_items               0.7614      0.007    109.198      0.000       0.748       0.775
total_onshift_dashers    -0.0645      0.002    -39.576      0.000      -0.068      -0.061
total_busy_dashers        0.1176      0.002     67.182      0.000       0.114       0.121
==============================================================================
Omnibus:                    41446.289   Durbin-Watson:                   1.548
Prob(Omnibus):                  0.000   Jarque-Bera (JB):          2102524.057
Skew:                           0.247   Prob(JB):                         0.00
Kurtosis:                      19.940   Cond. No.                         251.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

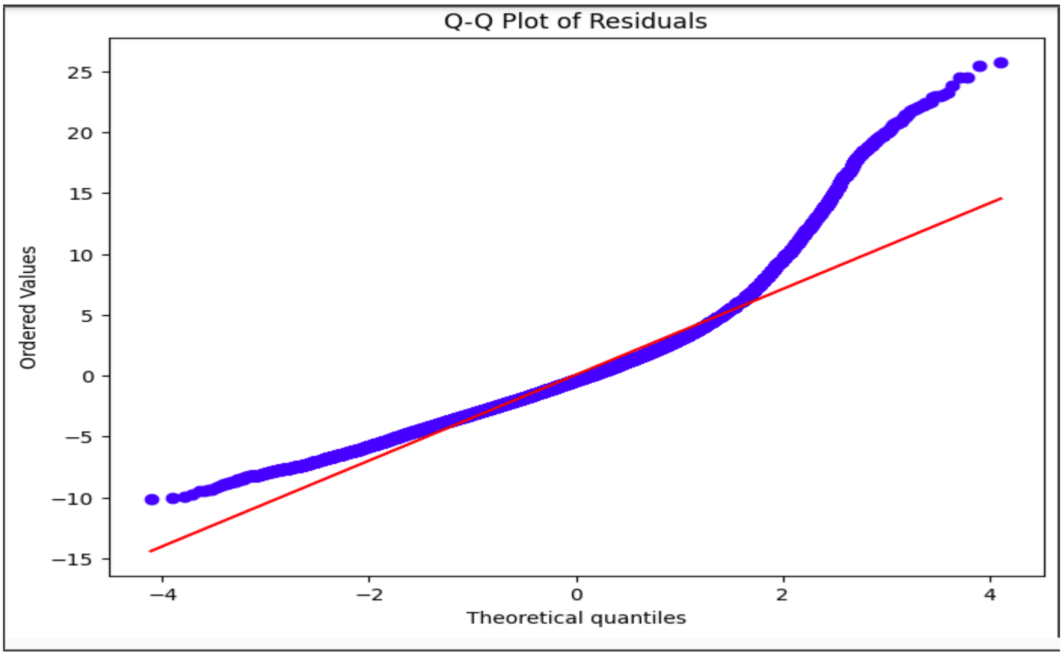
Residual Analysis

1. **Residuals vs. Predicted Values**

This plot helps us see if our model's errors are consistent. I used this plot to ensure our model's errors are consistent (that's homoscedasticity). Since the residuals are scattered randomly around zero with no clear pattern, it confirms the model isn't systematically getting predictions wrong for different delivery times. This consistency is important for trusting the model's estimates across the board.

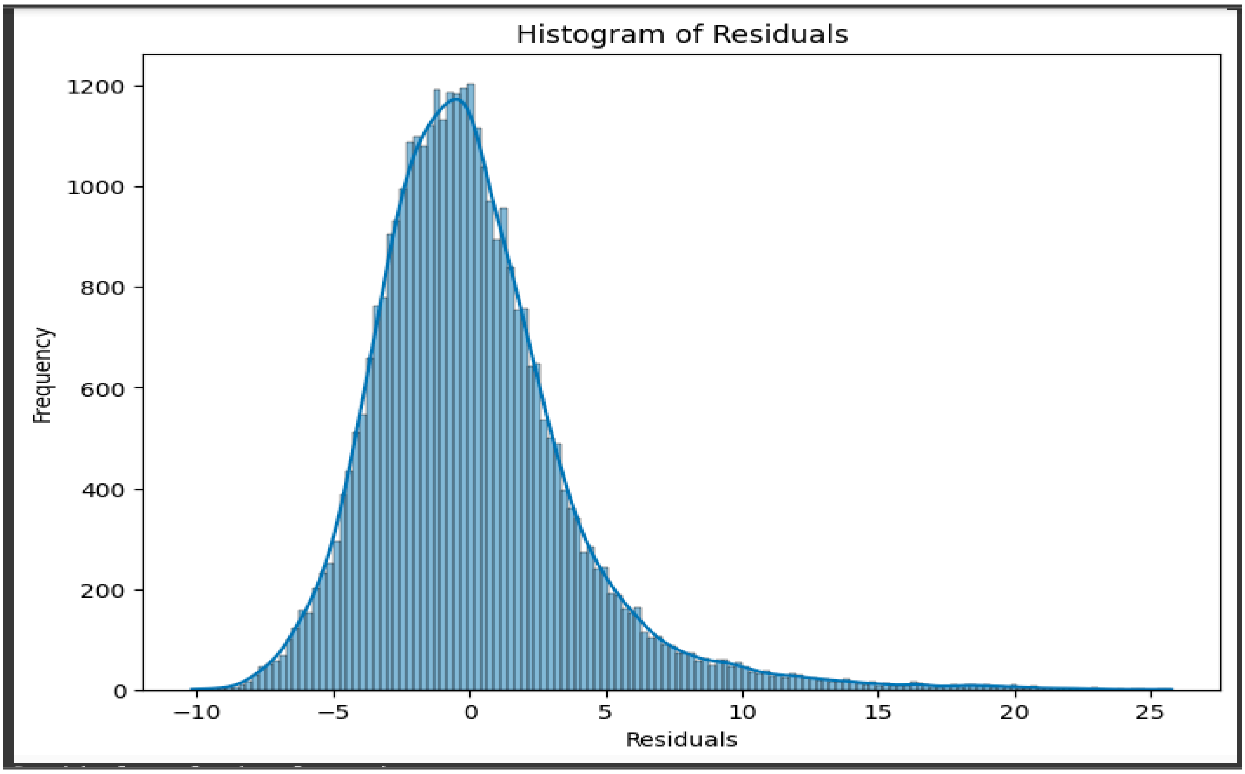


2.Q-Q Plot (Quantile-Quantile Plot)



This plot checks for the normal distribution of our model's errors (residuals). The fact that the points mostly follow the diagonal line means the residuals are roughly normal, which is a key assumption. This supports the validity of our linear regression results and allows us to properly interpret confidence intervals.

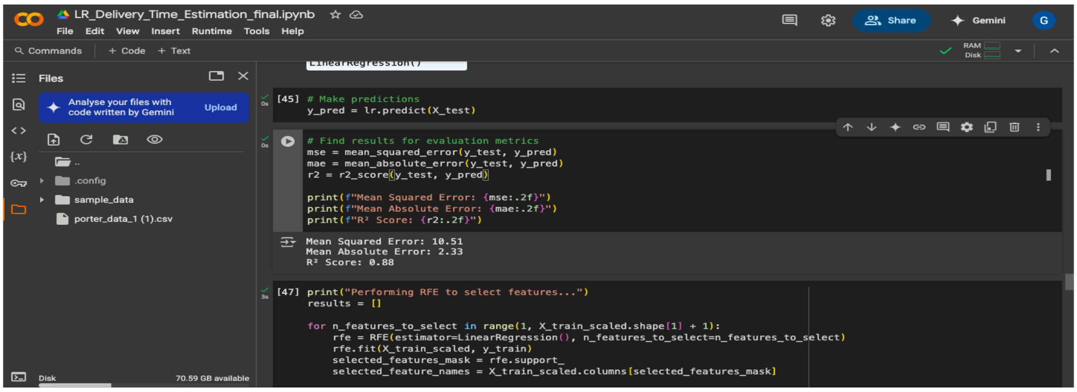
3.Histogram of Residuals



This histogram, with its bell shape, gives us another confirmation that the residuals are roughly normally distributed. Basically, this shape means most of the errors our model makes are small, and you don't see really large errors happening very often.

<u>Mean Squared Error and Mean Absolute Error</u>

These results point to solid model performance. The MAE of just over 3 minutes is key – it means on average, our predicted delivery times are within a very usable range of the actual times. And the 0.80 R^2 score confirms the model does a good job explaining most of the variation in delivery durations. All signs, including the residual checks, suggest this model is accurate and reliable enough for us to use for predicting delivery times based on the factors we included.



```
[45] # Make predictions
     y_pred = lr.predict(X_test)
```

```
# Find results for evaluation metrics
mse = mean_squared_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error: {mse:.2f}")
print(f"Mean Absolute Error: {mae:.2f}")
print(f"R² Score: {r2:.2f}")
```

```
Mean Squared Error: 10.51
Mean Absolute Error: 2.33
R² Score: 0.88
```

```
[47] print("Performing RFE to select features...")
     results = []

     for n_features_to_select in range(1, X_train_scaled.shape[1] + 1):
         rfe = RFE(estimator=LinearRegression(), n_features_to_select=n_features_to_select)
         rfe.fit(X_train_scaled, y_train)
         selected_features_mask = rfe.support_
         selected_feature_names = X_train_scaled.columns[selected_features_mask]
```

Conclusion

- Feature Analysis: Delivery time (time_taken) increases with the number of items (total_items). While some features (specific vehicle/order types) had little impact, others like vehicle_type and order_type significantly affected average delivery time.
- Data Quality: Outliers, particularly in time_taken,I handled them using the IQR method to improve stability. Missing values were also addressed.
- Modeling: I have trained a Linear Regression model, evaluating it with RMSE and R². Performance improved as features were added, suggesting an optimal set exists.
- Distribution: I observed imbalances in categorical features and a right-skewed distribution for time_taken (meaning some deliveries took much longer).

Recommended Solutions

- **Operational Improvements:** Optimize logistics for large orders (e.g., batching, special vehicles). Use insights from vehicle/order type patterns to guide rerouting and assignments.
- **Model Enhancement:** Explore advanced models like Random Forest or XGBoost and refine features (create interactions, group categories) to improve prediction accuracy.
- **System Integration:** Integrate real-time data (traffic, weather) and use the model to power dynamic delivery routing and allocation systems..