

**Project:** Delivery Time Estimation

**Author:** Payal Chakraborty

**Assignment:** Linear Regression Assignment

**Dataset:** Porter Delivery Dataset (porter\_data\_1.csv)

## 1. Introduction

We are predicting delivery time (in minutes) for orders using a linear regression model. This problem is important because accurate delivery time estimates help operations plan more effectively and allow customers to have realistic expectations. Our approach involves splitting the dataset, performing exploratory data analysis (EDA) on the training data, applying simple preprocessing, building a baseline linear regression model, refining it with recursive feature elimination (RFE), and finally running diagnostics to validate the model.

## 2. Data Overview

### Target Variable:

- `delivery_minutes = actual_delivery_time - created_at` (Created in Section 2.2.1 of the notebook.)

### Key Features Used:

- **Numeric:** `distance`, `total_outstanding_orders`, `total_busy_dashers`, `subtotal`, `total_items`
- **Categorical:** `market_id`, `store_primary_category`, `order_protocol`, `isWeekend`
- **Time-based:** `order_hour`, `order_dayofweek` (Created in Section 2.2.2 of the notebook.)

**Notes:** `market_id` is treated as a categorical feature (even if it appears numeric in the raw data). `isWeekend` is derived from `order_dayofweek` (Saturday/Sunday = 1, Monday–Friday = 0).

### Feature Definition Step:

- The final definition of X (features) and y (target) is shown in Section 2.3.1 of the notebook.

## 3. Methodology

Roadmap (tied to notebook sections):

- **Split (Section 2.3.2):** 80/20 train/validation with fixed `random_state` for reproducibility.
- **EDA on train only (Section 3.):** Distributions of target/features, pairwise relationships, correlations, and outlier checks performed only on the training split to avoid leakage.
- **Preprocessing for modeling (Section 5.1):** `StandardScaler` on numeric features and `OneHotEncoder` on categorical features, applied via a `ColumnTransformer` (or equivalent) so transformations are learned only from train data.
- **Baseline model (Section 5.2):** Linear Regression inside a single Pipeline that chains preprocessing + model, ensuring clean, leakage-safe evaluation.
- **Feature selection (Section 5.3):** Recursive Feature Elimination (RFE) testing different feature counts, choose the configuration with the best validation MAE/RMSE.
- **Diagnostics (Section 6.1 & 6.2):** Residual plots (residuals vs fitted/feature) and coefficient interpretation (sign/magnitude) to validate assumptions and extract insights.

## 4. Exploratory Data Analysis (EDA) & Visualizations

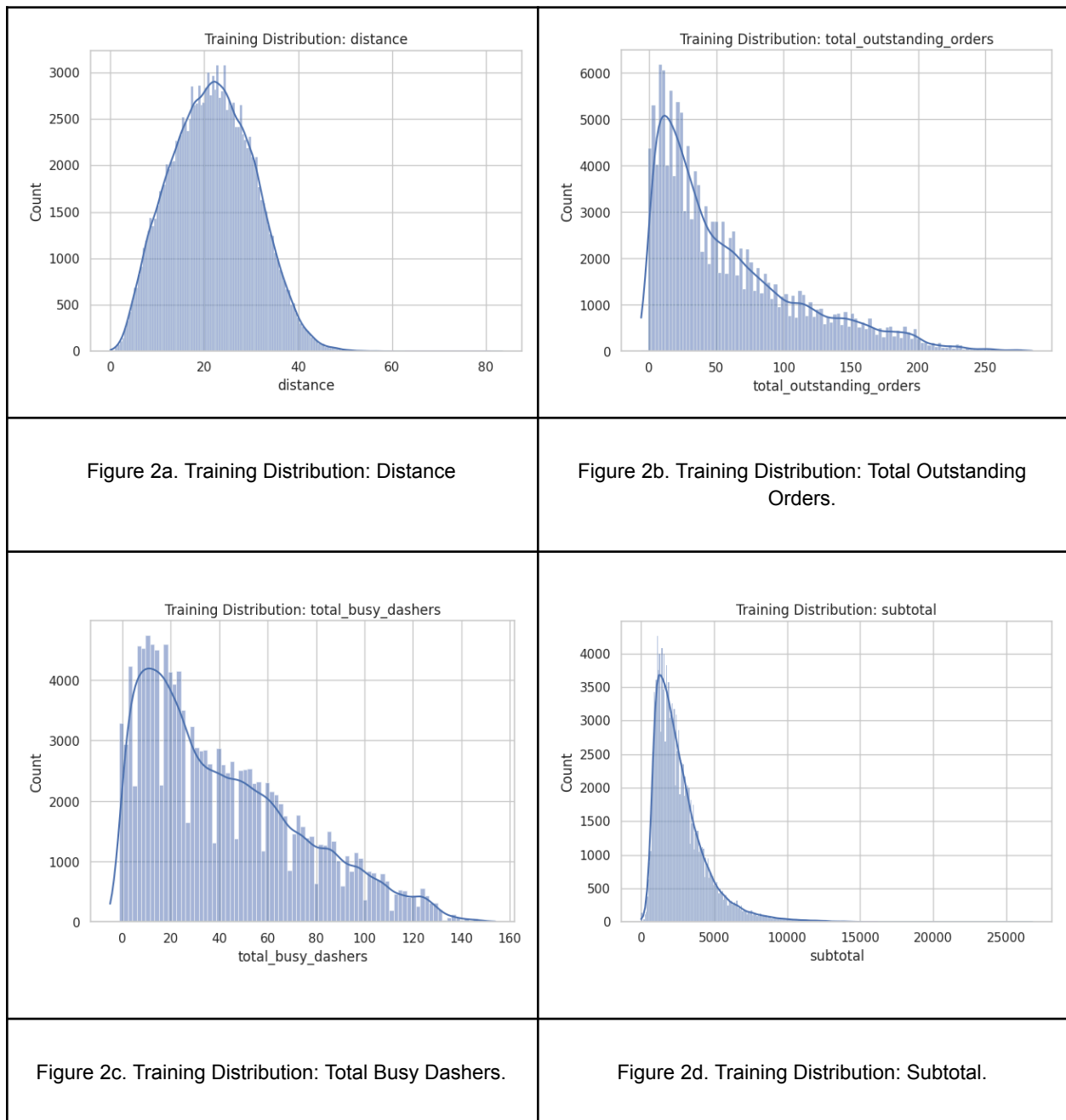
### A. Target Distribution (Training Only)



Training Distribution: Delivery Time (minutes)

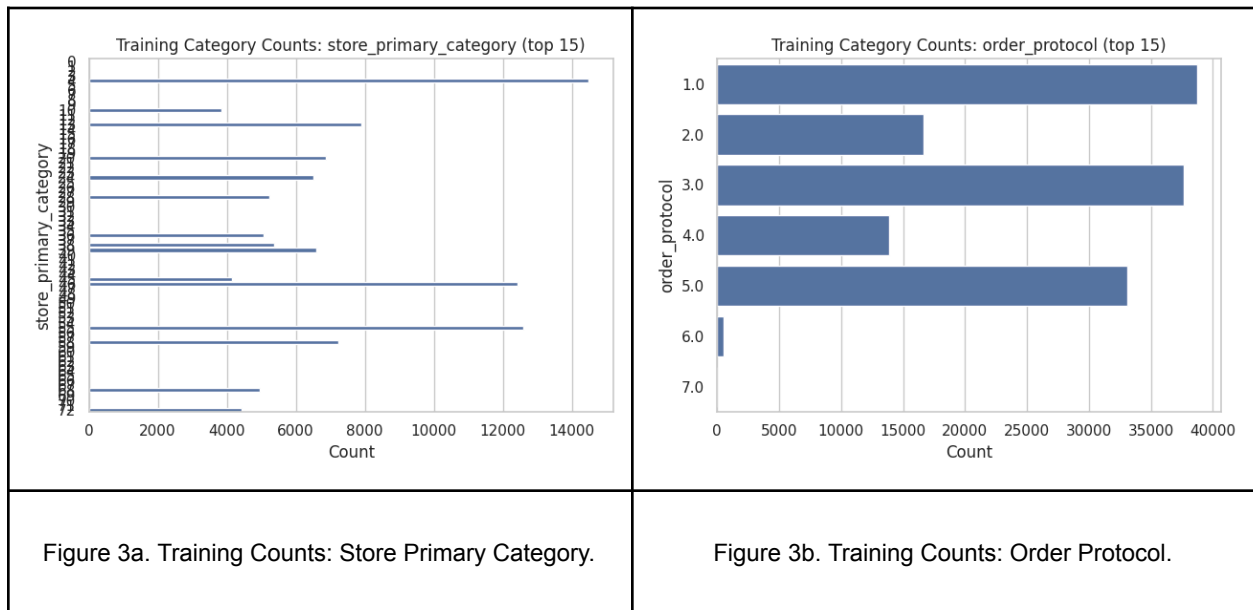
- **What it shows:** How delivery times are spread across orders, highlights any skew or long tail.
- **How we describe:** We see delivery times are slightly right-skewed. A few longer deliveries appear as the right tail.

## B. Numeric Features - Distributions (Training Only)



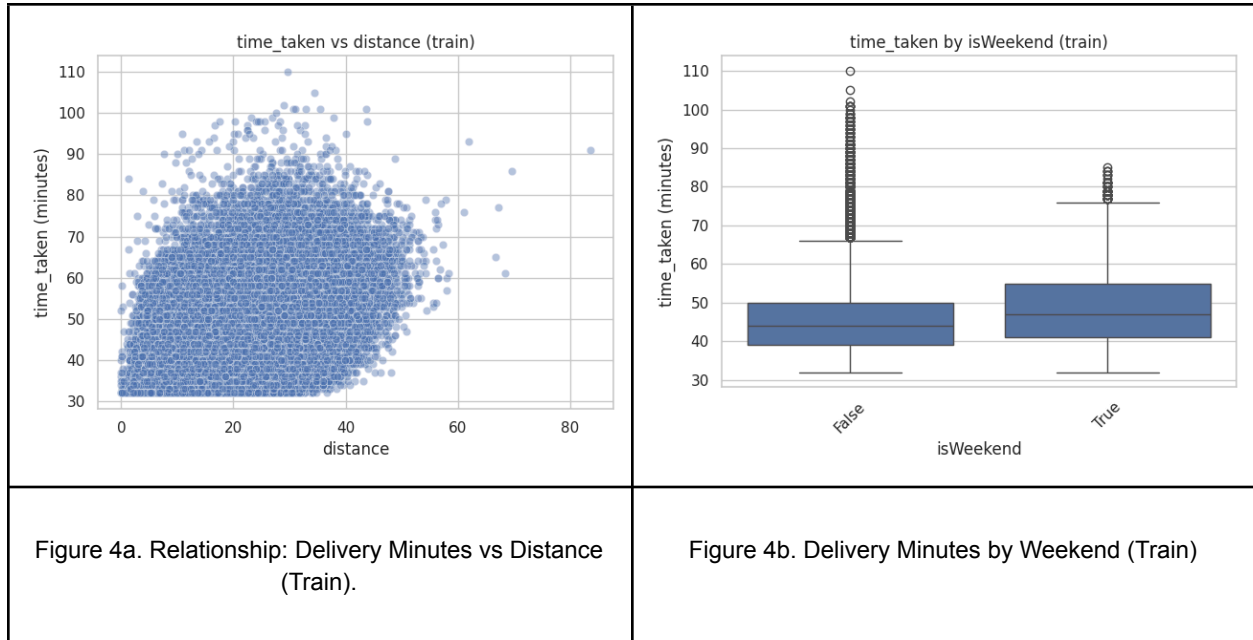
- **What they show:** Overall spread and skew of core numeric drivers, potential tails/outliers.
- **How we describe:** distance is concentrated at lower values with a tail, operational counts show moderate spread, suggesting different load conditions.

## C. Categorical Features - Counts (Training Only)



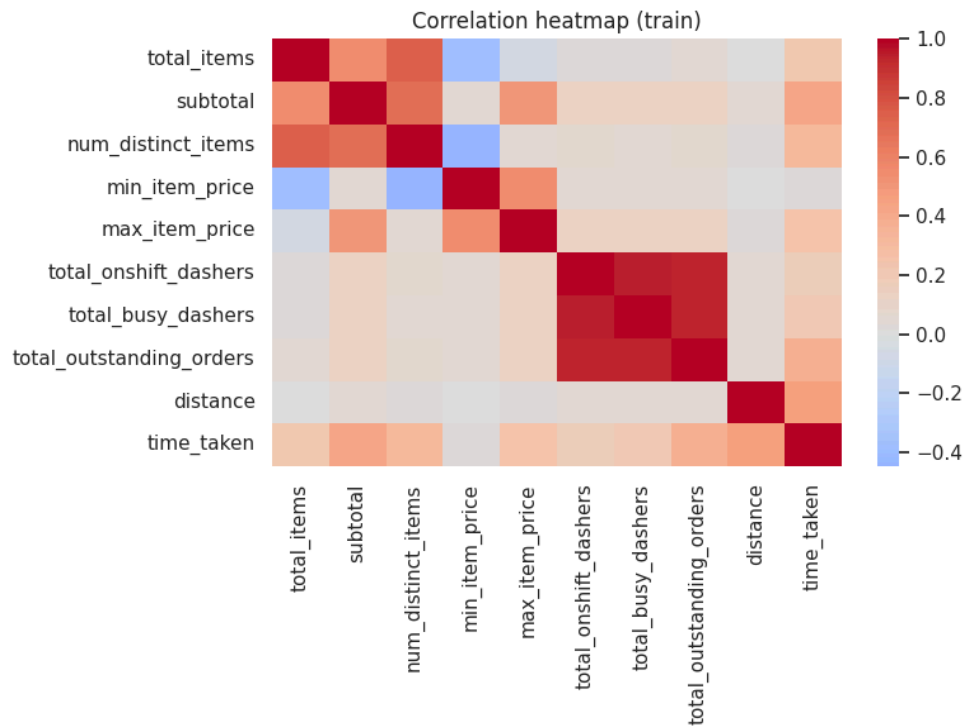
- **What they show:** Prevalence of store types and order channels, highlights any imbalance.
- **How we describe:** Some store types are more common than others, order channel distribution is slightly imbalanced.

## D. Relationships with the Target (Training Only)



- **What they show:** Strength and form of association between distance and delivery time; group-level effects from time-of-day or weekend.
- **How we describe:** distance increases with delivery time (clear upward trend). Weekend effects show slightly higher times during weekends (higher median and upper whisker), likely due to demand.

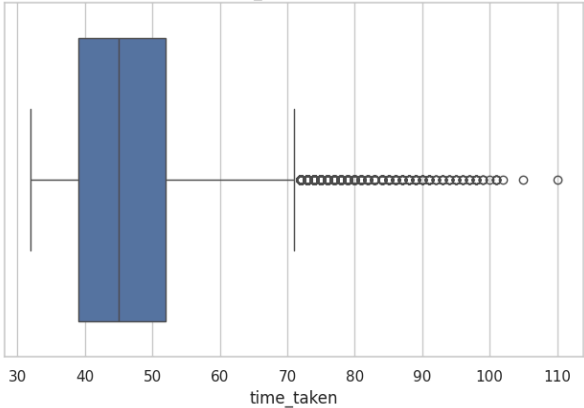
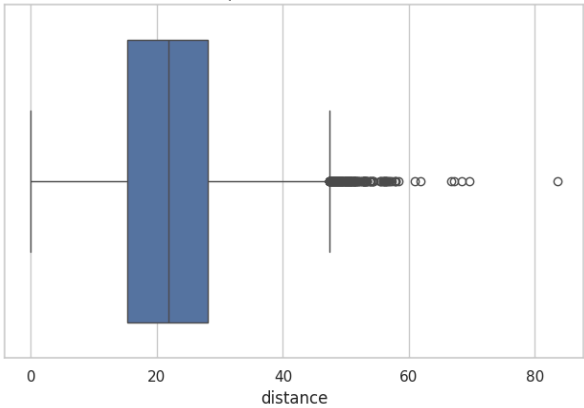
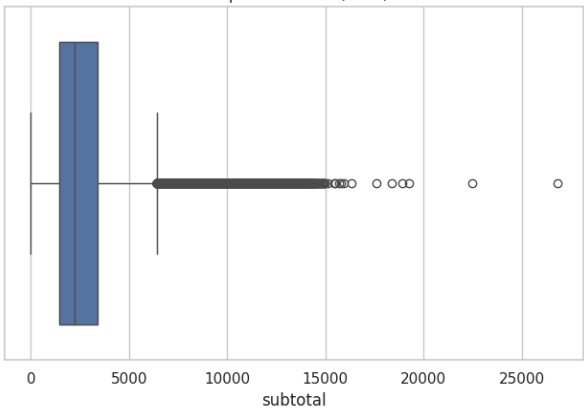
## E. Correlation Heatmap — Numeric Features (Training Only)



Correlation Heatmap (Train - Numeric Features)

- **What it shows:** Pairwise linear correlations among numeric inputs and the target.
- **How we describe:** distance shows the strongest positive correlation with delivery time (time\_taken), followed by load features (total\_outstanding\_orders, total\_busy\_dashers). Price/subtotal variables have weak correlation. The load features are also highly inter-correlated with each other, so feature selection/regularization is recommended.

F. Outliers (Training Only)

<div><p>Boxplot: time_taken (minutes) - train</p><p>time_taken</p></div>	<p>Figure 6a. Boxplot: Delivery Minutes (Train).</p>
<div><p>Boxplot: distance (train)</p><p>distance</p></div>	<p>Figure 6b. Boxplot: Distance (Train).</p>
<div><p>Boxplot: subtotal (train)</p><p>subtotal</p></div>	<p>Figure 6c. Boxplot: Subtotal (Train).</p>

- **One line on handling (from Section 3.4.2):** We capped outliers using a simple IQR rule on the training set and applied the same caps to validation for consistency.
- **What it shows / why it matters:** The boxplots for `delivery_minutes`, `distance`, and `subtotal` show prominent upper-tail outliers (many points beyond  $Q3 + 1.5 * IQR$ ). This confirms the right-tail behavior seen in the target distribution and indicates that a few large orders/distances can dominate error in linear regression. Capping on the training set reduces the influence of these extremes while preserving genuine variability; we then apply the same caps to validation for consistency.

## 5. Modeling Results

### A. Baseline Linear Regression (Validation)

- Baseline Linear Regression gives  $R^2 = 0.5989$ , MAE = **4.57** min, RMSE = **5.92** min on validation.
- **Source:** Section 5.2
- **What it shows:** MAE is the typical error in minutes, RMSE penalises bigger mistakes,  $R^2$  is the fraction of variance explained.

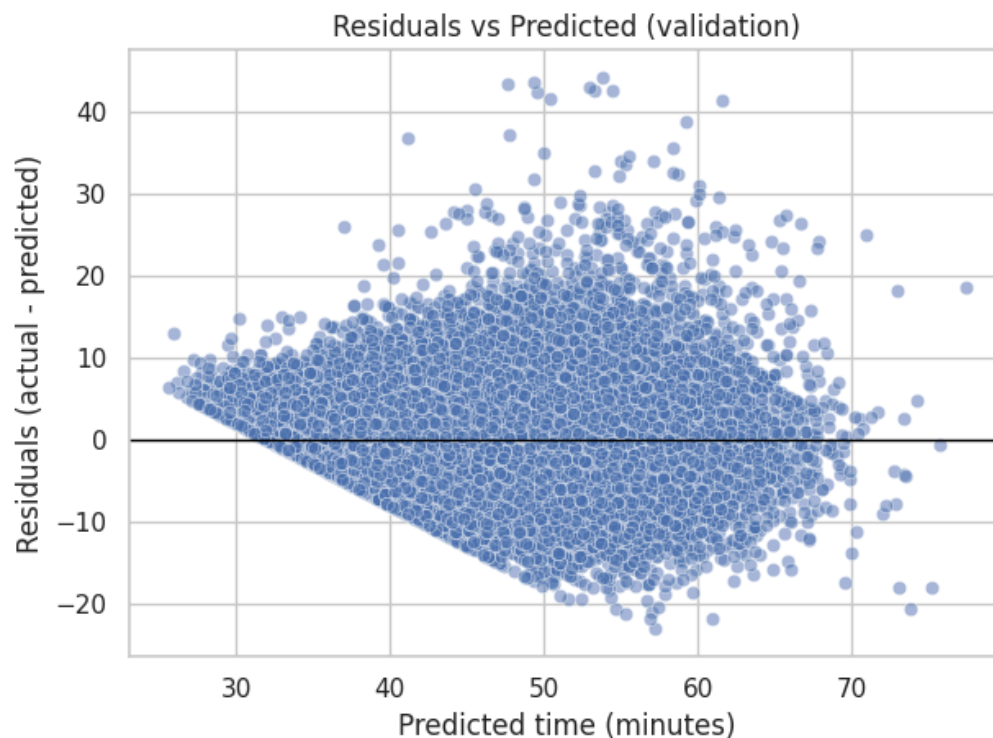
### B. Final Model (after RFE)

- We tried  $k \in \{5, 8, 10, 15, \dots\}$  and selected  $k = 15$  (lowest RMSE). Final model:  $R^2 = 0.5740$ , MAE = **4.72** min, RMSE = **6.11** min.
- **Selected features:** `num__subtotal`, `num__total_outstanding_orders`, `num__distance`, `cat__market_id_1.0`, `cat__market_id_2.0`, `cat__market_id_4.0`, `cat__store_primary_category_3`, `cat__store_primary_category_5`, `cat__store_primary_category_8`, `cat__store_primary_category_26`, `cat__store_primary_category_29`, `cat__store_primary_category_37`, `cat__store_primary_category_60`, `cat__store_primary_category_64`, `cat__store_primary_category_67`.
- **What it shows:** Compared to the baseline pipeline, the RFE model ( $k = 15$ ) shows **worse** validation performance ( $R^2$  0.5740 vs 0.5989; MAE 4.72 vs 4.57; RMSE 6.11 vs 5.92). The baseline pipeline remains the best performer on validation.



## 6. Error Analysis

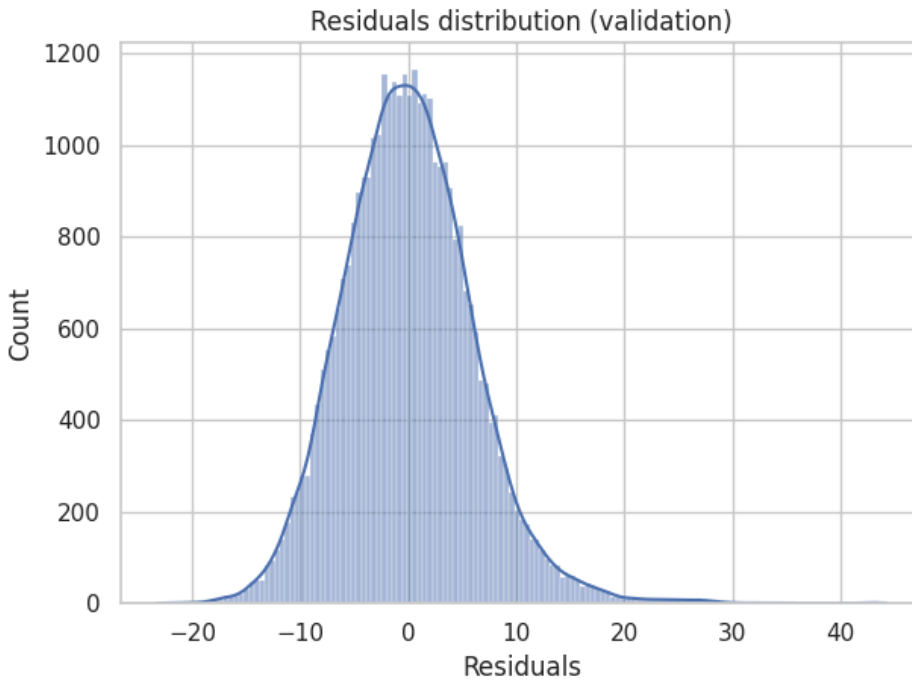
### A. Residuals vs Predicted (Validation)



Residuals vs Predicted (Validation)

- **What it shows:** Whether errors are centered around 0 with no systematic pattern.
- **How we describe:** Points are roughly centered around zero with no strong curve. However, the funnel shape shows increasing spread at higher predicted times (heteroscedasticity). There is a slight tilt toward negative residuals at the highest predictions, indicating mild over-prediction for the longest ETAs, acceptable for a baseline but worth addressing in future iterations.
- **Where does the model underperform?** Segment by route, distance bucket, hour, courier, etc.
- **Residual Plots:** residuals vs predicted/feature.
- **Actionable Fixes:** data collection, new features.

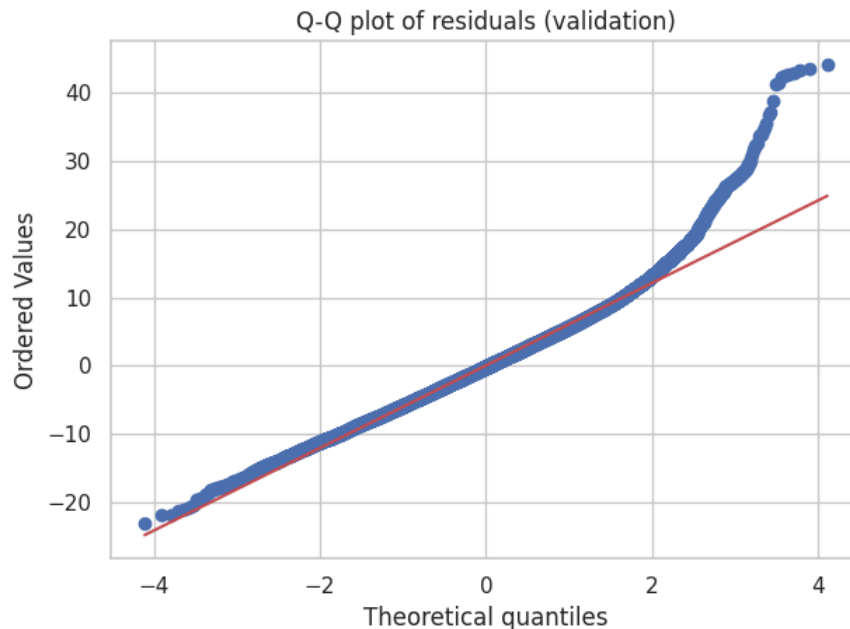
## B. Residual Histogram (Validation)



Residual Histogram (Validation)

- **What it shows:** Whether errors are symmetric, big tails = outliers.
- **How we describe:** Residuals look fairly symmetric with a light right tail, consistent with a few longer deliveries.

### C. Q-Q Plot (Validation)



Q-Q Plot of Residuals (Validation)

- **What it shows:** Normality of errors (points on the line).
- **How we describe:** Most points follow the line through the mid-quantiles. The upper tail bends above the line (heavier right tail) and the lower tail deviates slightly, typical for delivery-time data.

### D. Coefficient Analysis (Section 6.2)

- **Source:** Section 6.2 coefficient comparison DataFrame (`coef_compare`) showing processed feature names, original features, coefficient in scaled space, and (for numeric features) `minutes_per_+1_original_unit`.
- **What we report (text-only):**

**Top drivers (by `|coef|` in processed space, RFE final model):**

store\_primary\_category\_3 (+), market\_id\_2.0 (-),  
store\_primary\_category\_64 (+), total\_outstanding\_orders (+), distance  
(+), market\_id\_1.0 (+), store\_primary\_category\_29 (+),  
store\_primary\_category\_8 (+), store\_primary\_category\_37 (+),  
store\_primary\_category\_67 (+).

**Numeric per-unit effects (from `minutes_per_+1_original_unit`):**

- distance  $\approx$  **+0.49 min** per +1 unit
- total\_outstanding\_orders  $\approx$  **+0.09 min** per +1
- subtotal  $\approx$  **+0.002 min** per +1

For **one-hot categories**, coefficients are minutes **relative to the omitted baseline category** (positive = slower than baseline, negative = faster).

- **How we describe:** Among numeric features, **distance** has the largest positive per-unit effect, higher queue/workload features (e.g., **total\_outstanding\_orders**) add minutes, several **store/market categories** nudge ETAs up or down relative to their baselines (e.g., market\_id\_2.0 negative vs baseline, market\_id\_1.0 positive).
- **Note:** Although these effects are informative, recall that the **baseline pipeline** performed better than the RFE final model on validation ( $R^2$  0.5989 vs 0.5740). Use the coefficient insights for interpretation, not to claim superiority of the RFE model.

## 7. Insights

- Distance matters most, longer trips predictably increase delivery time.
- Operational load pushes times up, more outstanding orders or busy riders - slower assignment/fulfillment.
- Timing & categories matter, weekends and certain store types run slightly slower.
- The model is simple & explainable, easy for ops teams to interpret and trust.

## 8. Limitations & Next Steps

- Linear models may miss non-linear patterns.
- Regularised variants (Ridge/Lasso) and interaction terms might help.
- Extreme scenarios produce larger errors, more context (e.g., traffic, weather) could improve accuracy.

## 9. Conclusion

This project built an explainable baseline to predict delivery time in minutes using a linear regression pipeline (scaling + one-hot encoding). EDA confirmed distance as the strongest driver, with workload and timing adding smaller effects. On validation, the baseline model performed best ( $R^2$  **0.5989**, MAE **4.57 min**, RMSE **5.92 min**), and diagnostics showed centered residuals with mild heteroscedasticity and slight tail deviations, reasonable for a first pass. These results provide a practical, trustworthy starting point for operations, accuracy can likely improve with regularized models (Ridge/Lasso), interaction terms, and richer context features such as traffic and weather.