Delivery Time Estimation — Final Report

Date: 19th Aug 2025
Dataset: `porter_data_1.csv` (175,777 rows, 14 columns)
Target: `delivery_time` (minutes)
Author : Arvind Krishna

---

1) Executive Summary

We built a linear regression model to estimate delivery time using order details, market identifiers, and operational load indicators. After exploratory analysis, feature engineering, and multicollinearity control, the **final OLS model** retains eight predictors (plus intercept) and generalizes well:

- **Model 3 (Final):** $R^2 \approx$ **0.60** on train, **0.59** on test (your notebook's evaluation).

- **Key drivers (↑ delivery time):** `distance`, `subtotal`, `total_outstanding_orders`.

- **Key drivers (↓ delivery time):** `order_hour` (later hours reduce time modestly), and certain markets vs. baseline market (markets 2, 3, 4, 5 have negative adjustments).

- **Operational takeaway:** Queue load (`total_outstanding_orders`) and dispatch distance dominate service time; time-of-day and market effects meaningfully shift the baseline.

---

2) Problem, Data & Scope

Business goal. Predict a realistic delivery time estimate at order creation to improve ETA accuracy, customer satisfaction, and fleet allocation.

Data. Single table with order timestamps, item/price details, market and protocol identifiers, as well as live-load signals (on-shift / busy dashers, outstanding orders), and courier **`distance`**.

Outcome variable. `delivery_time` (minutes), computed as the difference between `actual_delivery_time` and `created_at`.

Train/test split. 80/20 with `random_state=100` → **140,621** training and **35,156** test records.

---

3) Data Preparation

3.1 Fixing Data Types (Notebook §2.1)

- Parsed timestamps: `created_at`, `actual_delivery_time` → `datetime`.
- Derived categorical and integer fields:
    - `order_hour` (0–23, `int32`), `order_day_of_week` (categorical), `isWeekend` (0/1).

- Dropped raw timestamps post-derivation.

3.2 Feature Engineering

- **Target:** `delivery_time` in minutes.

- **Calendar/time features:** `order_hour`, `order_day_of_week`, `isWeekend`.

- **Operational load:** `total_onshift_dashers`, `total_busy_dashers`, `total_outstanding_orders`.

- **Commercials & items:** `subtotal`, `max_item_price`, `num_distinct_items`, `total_items`.

- **Geography:** `distance`.

3.3 Encoding & Split

- One-hot encoded: `market_id`, `order_protocol`, `order_day_of_week` (`drop_first=True`; baseline market is **1.0**).

- Defined X/y, then 80/20 split (as above).

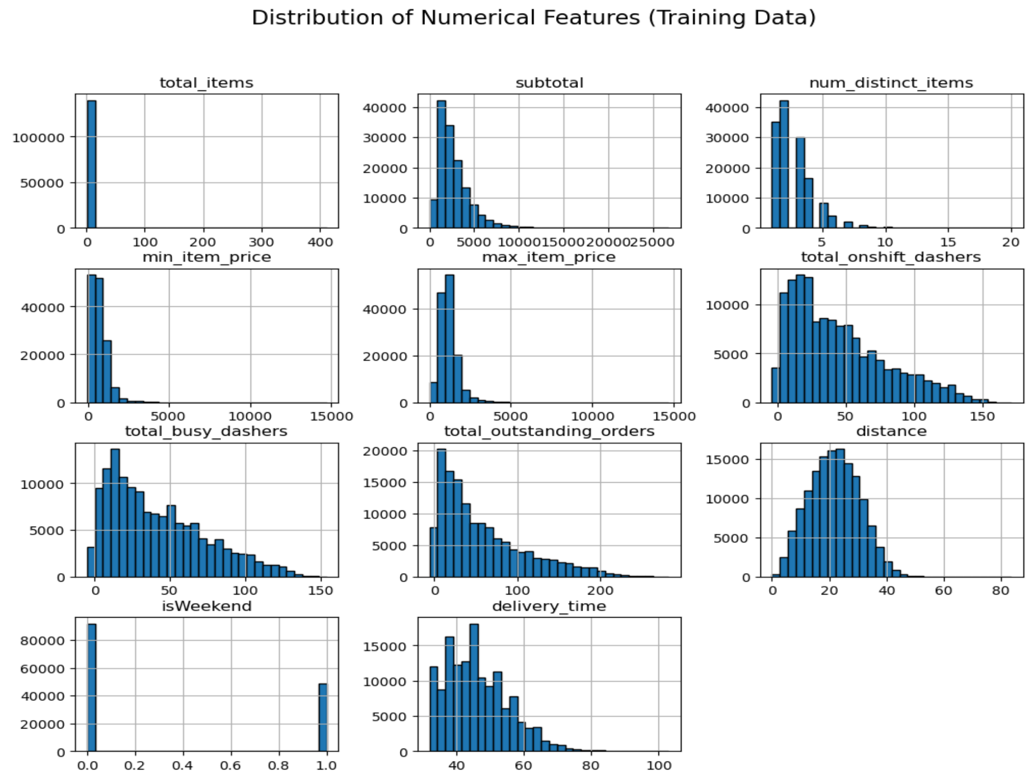
3.4 Outliers

- Capped **1st/99th percentiles** for selected numeric columns on train & test; also capped the target in train (per notebook).

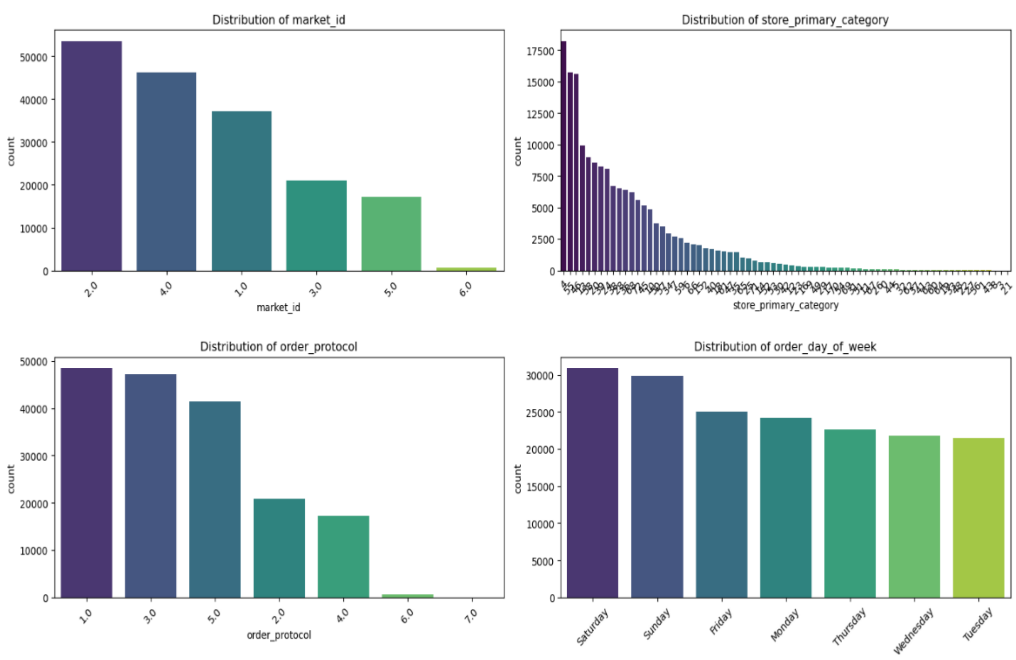
3.5 Scaling

- Standardized features for the scikit-learn run (to compare coefficient magnitudes).

- Final OLS model (Statsmodels) used **unscaled** features for interpretability.



Histograms for numerical features (training set).



Categorical distributions and `delivery_time` histogram.

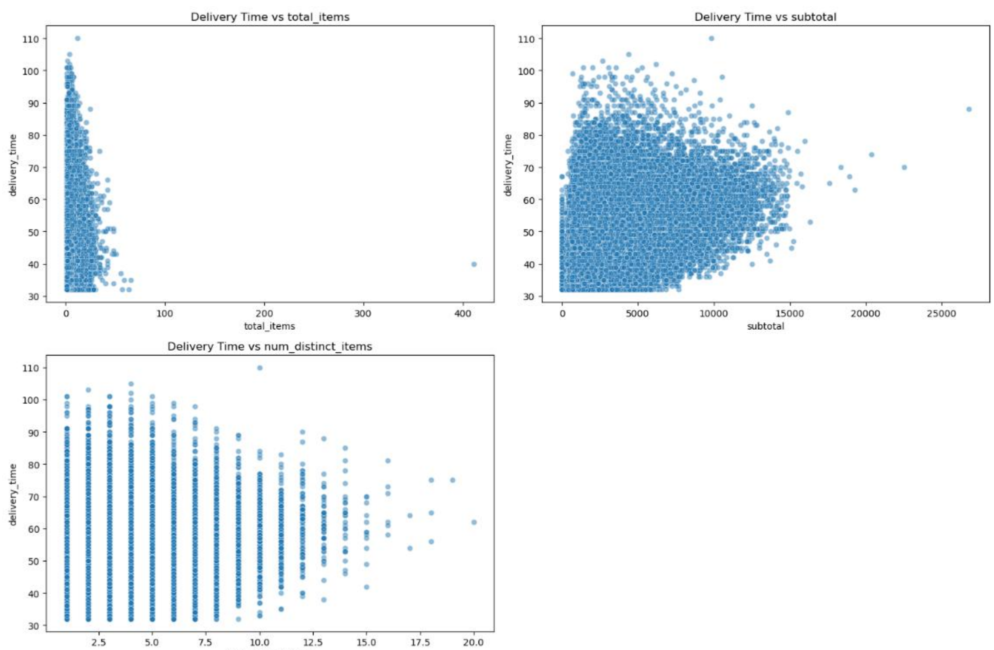
---

4) Exploratory Data Analysis (Highlights)

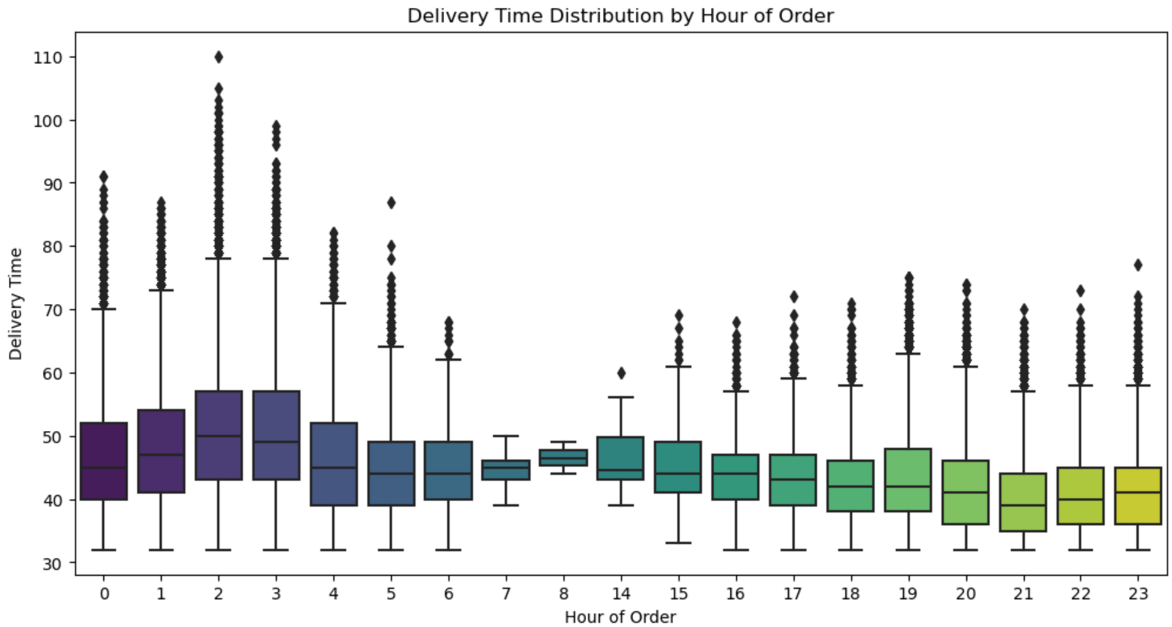
Univariate signals. `delivery_time` has a right-tailed distribution (long tail orders). `distance` and `subtotal` also exhibit long tails (capped later).
Categoricals. Market sizes and protocols are imbalanced (encode with `drop_first`). Weekends show slightly higher central tendency.

Bivariate relationships (from notebook correlation ranking): - **Top positive correlations with `delivery_time`:**
`distance` (**0.46**), `subtotal` (**0.41**), `total_outstanding_orders` (**0.38**), `num_distinct_items` (**0.31**), `max_item_price` (**0.25**), `total_items` (**0.22**).
- **Top negative correlations:**
`order_hour` (**~–0.34**), some `market_id` and `order_protocol` dummies (weak–moderate).

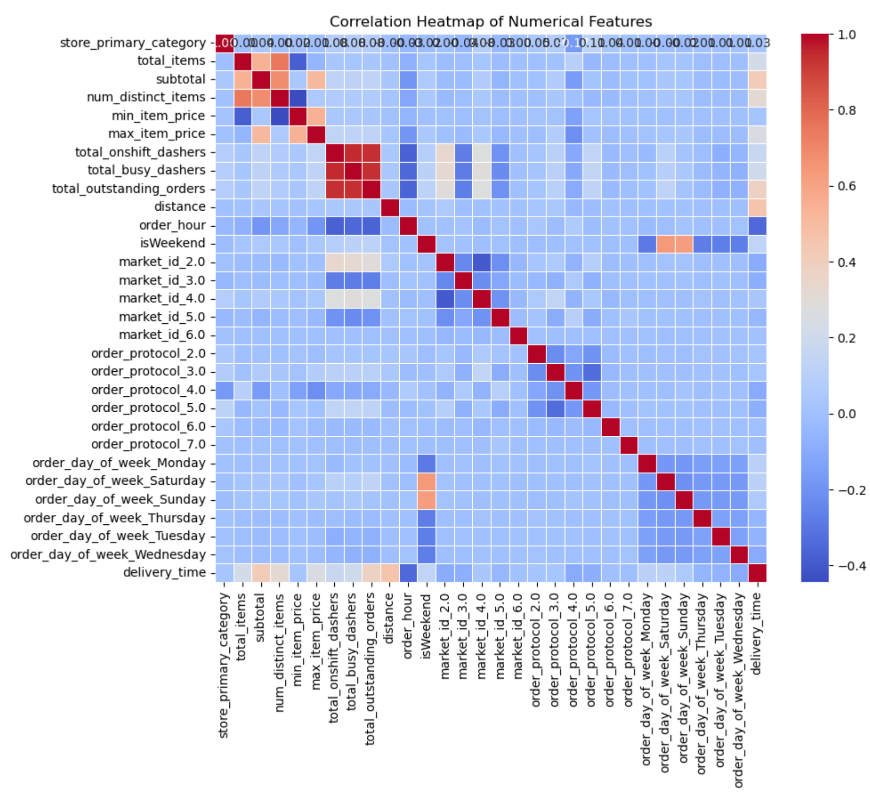
Multicollinearity (VIF). - Initial VIFs show high collinearity among load variables: `total_onshift_dashers` (**12.72**), `total_busy_dashers` (**11.89**), `total_outstanding_orders` (**10.38**).
- After pruning, the VIFs are acceptable (e.g., `total_busy_dashers` falls to **8.58**; final model removes both `onshift` and `busy` for stability).



Scatter plots — `delivery_time` vs. selected numeric features (e.g., `distance`, `subtotal`, `num_distinct_items`).



`delivery_time` by `order_hour` (strip/violin/box).



Correlation heatmap (numerics).

5) Modeling Approach

We iterated from a rich baseline to a lean, generalizable final model:

- **Model 1 — Full OLS (all engineered features & dummies).**
    - Train **R^2 ≈ 0.888**; high multicollinearity (see VIF table).

- **Model 2 — Reduced OLS (post correlation-based pruning).**
    - Dropped weak features: `order_protocol_6.0`, `order_protocol_7.0`, `market_id_6.0`, `order_protocol_2.0`, `min_item_price`.

    - Train **R^2 ≈ 0.752**; collinearity still present among load variables.
- **Model 3 — Final OLS (post-VIF refinement).**
    - Removed both `total_onshift_dashers` and `total_busy_dashers`.

    - **Train R^2 ≈ 0.602**; **Test R^2 ≈ 0.59** (per notebook evaluation).

    - Best bias–variance trade-off among OLS variants; coefficients are stable and interpretable.

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.602
Model:	OLS	**Adj. R-squared:**	0.602
Method:	Least Squares	**F-statistic:**	2.654e+04
Date:	Tue, 19 Aug 2025	**Prob (F-statistic):**	0.00
Time:	20:45:09	**Log-Likelihood:**	-4.4588e+05
No. Observations:	140621	**AIC:**	8.918e+05
Df Residuals:	140612	**BIC:**	8.919e+05
Df Model:	8		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	46.1328	0.015	3000.587	0.000	46.103	46.163
subtotal	2.9372	0.016	186.254	0.000	2.906	2.968
total_outstanding_orders	4.0279	0.020	201.490	0.000	3.989	4.067
distance	4.1786	0.015	270.819	0.000	4.148	4.209
order_hour	-1.0550	0.017	-61.565	0.000	-1.089	-1.021
market_id_2.0	-4.0558	0.022	-184.728	0.000	-4.099	-4.013
market_id_3.0	-1.3156	0.018	-72.749	0.000	-1.351	-1.280
market_id_4.0	-3.1136	0.022	-144.601	0.000	-3.156	-3.071
market_id_5.0	-1.0648	0.018	-60.281	0.000	-1.099	-1.030

Omnibus:	1712.811	**Durbin-Watson:**	2.003
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	2199.214
Skew:	0.187	**Prob(JB):**	0.00
Kurtosis:	3.485	**Cond. No.**	2.79

6) Results & Interpretation

6.1 Final model specification (Model 3 — OLS)

Predictors kept: `subtotal`, `total_outstanding_orders`, `distance`, `order_hour`, and market dummies (`market_id_2.0`, `market_id_3.0`, `market_id_4.0`, `market_id_5.0`), with **market 1.0** as the baseline.

Estimated equation (minutes):

```
Delivery Time = 46.1328
                + 2.9372·subtotal
                + 4.0279·total_outstanding_orders
                + 4.1786·distance
                − 1.0550·order_hour
                − 4.0558·market_id_2.0
                − 1.3156·market_id_3.0
                − 3.1136·market_id_4.0
                − 1.0648·market_id_5.0
```

> Coefficients reproduced from your notebook's Statsmodels summary (Model 3). All are statistically significant at $p < 0.001$.

6.2 How to read these effects

- **Distance (+4.18 min per unit):** Largest structural driver—longer trips lengthen service time near-linearly within the observed range.
- **Queue load — `total_outstanding_orders` (+4.03):** Each additional outstanding order adds meaningful latency, capturing batching/queuing delays.
- **Commercial size — `subtotal` (+2.94):** Larger baskets (proxy for prep/hand-off complexity) increase time.
- **Time of day — `order_hour` (–1.06):** Later hours tend to be faster, possibly due to lower kitchen/road congestion.
- **Market effects (negative vs. Market 1 baseline):** Markets 2/3/4/5 are faster on average by ~1–4 minutes, reflecting localized operations/traffic.

6.3 Performance snapshot

- **Train $R^2 \approx 0.60$; Test $R^2 \approx 0.59$.**

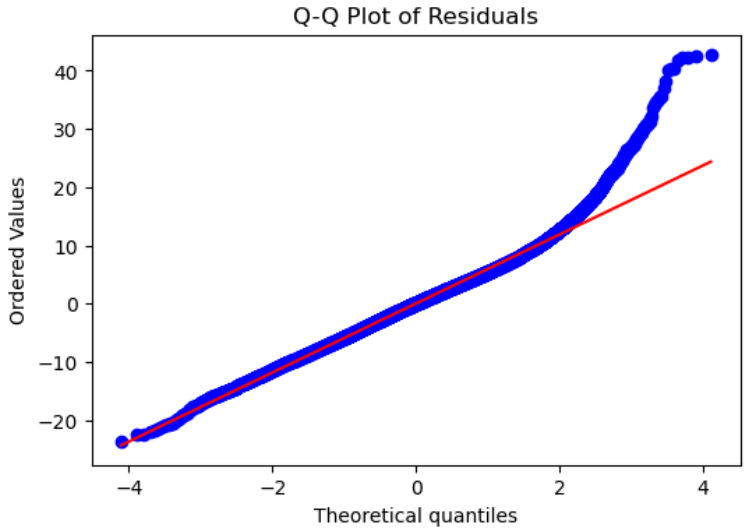
- **Diagnostics (Section 7)** show well-behaved residuals overall with mild right-tail under-prediction on very long deliveries (as expected for linear models).

---

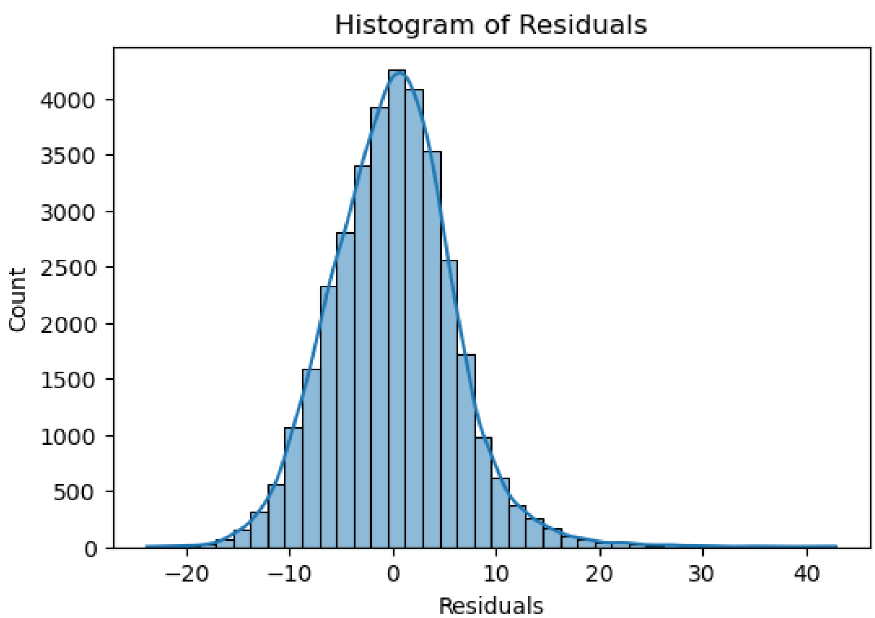
7) Model Diagnostics

- **Residuals vs. fitted:** No strong structure; slight funneling at high predictions suggests mild heteroscedasticity due to long-tail deliveries.

- **Q–Q plot:** Deviations in upper quantiles (heavy tail), acceptable elsewhere.



- **Residual histogram:** Roughly symmetric with right tail.



---

8) Business Insights & Recommendations

1. **Manage queue load proactively.** `total_outstanding_orders` materially lifts delivery time.
    - Trigger *pre-dispatch* or *micro-batching* only below a dynamic threshold of outstanding orders per zone.

    - Consider surge/slot controls when the queue exceeds threshold (ETA protection).
2. **Distance-aware routing.** Prioritize closer courier assignment to reduce the strongest driver (`distance`).
    - Introduce a hard cap (or surcharge) beyond a distance band to preserve SLA.

3. **Kitchen prep orchestration for large orders.** `subtotal` indicates prep complexity; fast-track packaging, pre-prep cues for high-value carts.
4. **Time-window messaging.** Use the `order_hour` effect to set customer-facing delivery windows (later hours can allow tighter ranges).

5. **Market playbooks.** Markets 2/3/4/5 outperform the baseline—harvest best practices (station placement, vendor SLAs, courier mix) and replicate.

---

9) Limitations & Next Steps

- **Linear form.** Effects are assumed additive and linear; interactions (e.g., `distance × hour`) and non-linearities are not modeled.

- **Omitted drivers.** Weather, traffic incidents, vendor prep times, courier skill are not captured.

- **Heteroscedasticity.** Long-tail variance suggests considering robust or transformed targets (e.g., log-minutes) for stability.

Appendix A — Final Coefficients (Model 3)

Feature	Coefficient (minutes)
Intercept	46.1328
`distance`	4.1786
`total_outstanding_orders`	4.0279
`subtotal`	2.9372
`order_hour`	−1.0550
`market_id_2.0`	−4.0558
`market_id_4.0`	−3.1136
`market_id_3.0`	−1.3156
`market_id_5.0`	−1.0648

Baseline category for markets is **`market_id_1.0`** (dummy-encoding with `drop_first=True`).

Appendix B — Reproducibility Notes

- Train/test split: `test_size=0.2`, `random_state=100`.
- Outlier capping: 1st/99th percentiles on selected numerics and on train target.
- Encoding: one-hot for `market_id`, `order_protocol`, `order_day_of_week` with first level dropped.
- Final model: Statsmodels OLS on unscaled features; removed `total_onshift_dashers` and `total_busy_dashers`.