Exercise Description — Data Science Take‑Home (Procurement & Supply Analytics)

Deliver in 1 week via GitHub, then present live on exam day

# 1. Overview

You are given a real‑world style dataset covering suppliers, products, price lists, purchase orders, and deliveries. Your goal is to (A) explore and assess data quality, (B) build and evaluate a late‑delivery prediction, (C) detect anomalous supplier price entries, and (D) answer a short SQL exercise. Focus on clarity, correct methodology (no leakage), and actionable insights.

# 2. Timeline & Submission

1. Deadline: Submit within 7 calendar days from receiving this exercise.
2. Submission format: A public or private GitHub repository. If private, invite all examiners (use the emails provided below).
3. Email the repository URL to the examiners when you submit.

* Examiners’ GitHub handles/emails:
* Abdalrahman.eweiwi@siegwerk.com
* Repository name suggestion: ds-takehome-[lastname]
* Branch: main (use feature branches if you like, keep commit history meaningful).

# 3. What to Deliver (Repository Contents)

* notebooks/EDA.ipynb — your exploratory data analysis and data‑quality checks (with narrative and visuals). Hence, you will find a starter code for your reference here.
* notebooks/Model\_Anomaly.ipynb — modeling, evaluation, calibration, threshold selection; price anomaly detection. Hence, you will find a starter code for your reference here.
* sql/sql\_exercise.sql — solutions to the SQL tasks (see Section 7).
* REPORT.md (max 2 pages) — executive summary, approach, metrics, key insights, and recommendations.
* README.md — environment setup and how to run your code/notebooks end‑to‑end.
* requirements.txt or environment.yml — exact package versions (reproducibility).
* Optional: scripts/ or src/ — helper functions (keep it simple and readable).

# 4. Live Presentation (on exam day)

* Total time: 30–45 minutes.
* Walk‑through (15–20 min): problem framing, EDA highlights, modeling choices, evaluation & calibration, anomaly method, recommendations.
* Q&A (10–15 min): methodology, assumptions, trade‑offs.
* Small live change (5–10 min): e.g., adjust threshold for a different operations capacity, add a simple feature, or re‑run slice analysis.

# 5. Dataset & Context

* The dataset includes: suppliers, products, price lists (with validity windows and min quantities), purchase orders (with logistics info), and deliveries (including late/partial/cancelled).
* Currencies are EUR and USD; you may normalize to a single currency with a clearly stated assumption.
* Target for classification: late\_delivery = 1 if actual\_delivery\_date > promised\_date. Exclude cancelled orders.

# 6. Tasks

## 6.A — EDA & Data Quality (recommended ~1.5–2h)

1. Join the data and verify keys (uniqueness, join coverage). Identify missing values and inconsistencies.
2. Seasonality: orders and late rates by month. Supplier/country patterns, ship modes, and distance bands.
3. Produce 3–5 clear visuals and a concise narrative.
4. List potential leakage risks and propose mitigations (e.g., time‑aware splits, causal features).

## 6.B — Predict Late Deliveries (recommended ~2–3h)

1. Construct features available at order time only (e.g., supplier rating/preferred, distance, ship mode, incoterm, payment terms, hazard, promised lead days, qty, urgent, normalized unit price).
2. Temporal split: Train on orders up to 2025‑03‑31; validate on 2025‑04‑01 to 2025‑06‑30 (to avoid leakage).
3. Metrics: Primary — PR‑AUC (average precision). Also report ROC‑AUC and F1.
4. Thresholding: report performance at (i) 0.5, (ii) best‑F1, and (iii) an operations‑capacity threshold (e.g., top‑k % you can action).
5. Calibration: include a reliability diagram and Brier score; comment on over/under‑confidence and fixes.
6. Slice analysis: compare performance by ship\_mode, supplier country, and distance bucket; comment on disparities and actions.

## 6.C — Price Anomaly Detection (recommended ~1–1.5h)

1. Normalize prices to EUR with a clear assumption.
2. Within each (supplier\_id, sku) series over time, flag anomalies using a robust and explainable method (e.g., z‑score on log price, rolling baseline bands, or Isolation Forest).
3. Provide a Top‑N anomalies table and include 2–3 small plots with flagged points.
4. Explain your threshold choice and how you would validate or review alerts operationally.

## 6.D — SQL Exercise (30–45 min)

1. Monthly late rate overall and by ship\_mode in 2025‑04 to 2025‑06.
2. Top 5 suppliers by volume with their late\_rate in the same window.
3. For each order: supplier trailing 90‑day late rate strictly before order\_date (windowed).
4. Detect overlapping price windows per (supplier\_id, sku).
5. Attach valid price at order date, normalize to EUR (assume USD→EUR = 0.92), compute order\_value\_eur.
6. Flag price anomalies via z on ln(price\_eur) per series; return top 10 |z|.
7. Incoterm × distance buckets: average delay\_days and count in validation window.
8. Bonus: with predictions(order\_id, p\_late), bucket top 10% as high risk and compare late\_rate vs. low risk.

*Deliver a single file: sql/sql\_exercise.sql with commented, readable queries (use CTEs).*

# 7. If You Already Have Predictions

1. Place predictions.csv (columns: order\_id, p\_late) at the repository root.
2. In notebooks/Model\_Anomaly.ipynb, merge predictions with the validation labels (exclude cancellations).
3. Compute PR‑AUC, ROC‑AUC; report F1 at 0.5, best‑F1, and capacity‑based thresholds; show confusion matrices.
4. Show a reliability diagram and Brier score; discuss calibration. **(Bonus)**
5. Run slice analysis by ship\_mode, supplier country, and distance buckets; discuss disparities and next steps. **(Bonus)**
6. Choose a business threshold aligned to an assumed operations capacity (e.g., top 15%). Justify your choice. **(Bonus)**

# 8. Technical Requirements

* Python 3.10+ with pandas, numpy, scikit‑learn, matplotlib/seaborn; optional: shap, xgboost/lightgbm.
* Notebook outputs should be visible (rendered plots, printed metrics).
* Random seeds set where appropriate to ensure stability.
* No external data is required; if you add any, document clearly and commit small samples only.

# 9. Evaluation Rubric (100 points)

* EDA & Data Quality — 25: joins, visuals, narrative, leakage awareness.
* Modeling — 30: temporal split, features, baselines → improvements, thresholding, calibration.
* Evaluation & Explainability — 20: correct metrics (PR‑AUC), validation, interpretation, business insight.
* Price Anomalies — 15: normalization, method, outputs & plots.
* Code & Repro — 10: clean notebooks/code, environment, README clarity.

# 10. Rules & Fair Usage

* You may consult documentation, tutorials, and public examples. The work and narrative must be your own.
* Cite any non‑trivial snippet you adapt. Avoid copying end‑to‑end solutions.
* We grade within the time cap. Focus on correctness, clarity, and trade‑offs over extra features.