

E-commerce Delivery Delay Risk Prediction



Recap ...

Business Problem

Olist: e-commerce platform at scale (~9k partner retailers)

Faces **inconsistent** delivery performance, leading to a late-delivery rate that fluctuates based on operational conditions.

Solution

Late delivery prediction

Operationalize an end-to-end machine learning pipeline to **predict late deliveries**.

Key Users

Data Science Team (DS)

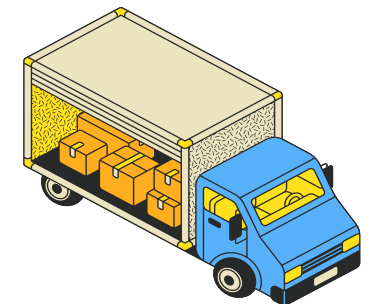
Owns and monitor the model and the entire pipeline

Logistics Team

Results of the model help them expedite orders that are at high risk of being delivered late

Customer Service Team

Results of the model help them determine which customers to proactively notify in terms of order delivery



Recap ...

Machine Learning Task

Supervised Binary Classification

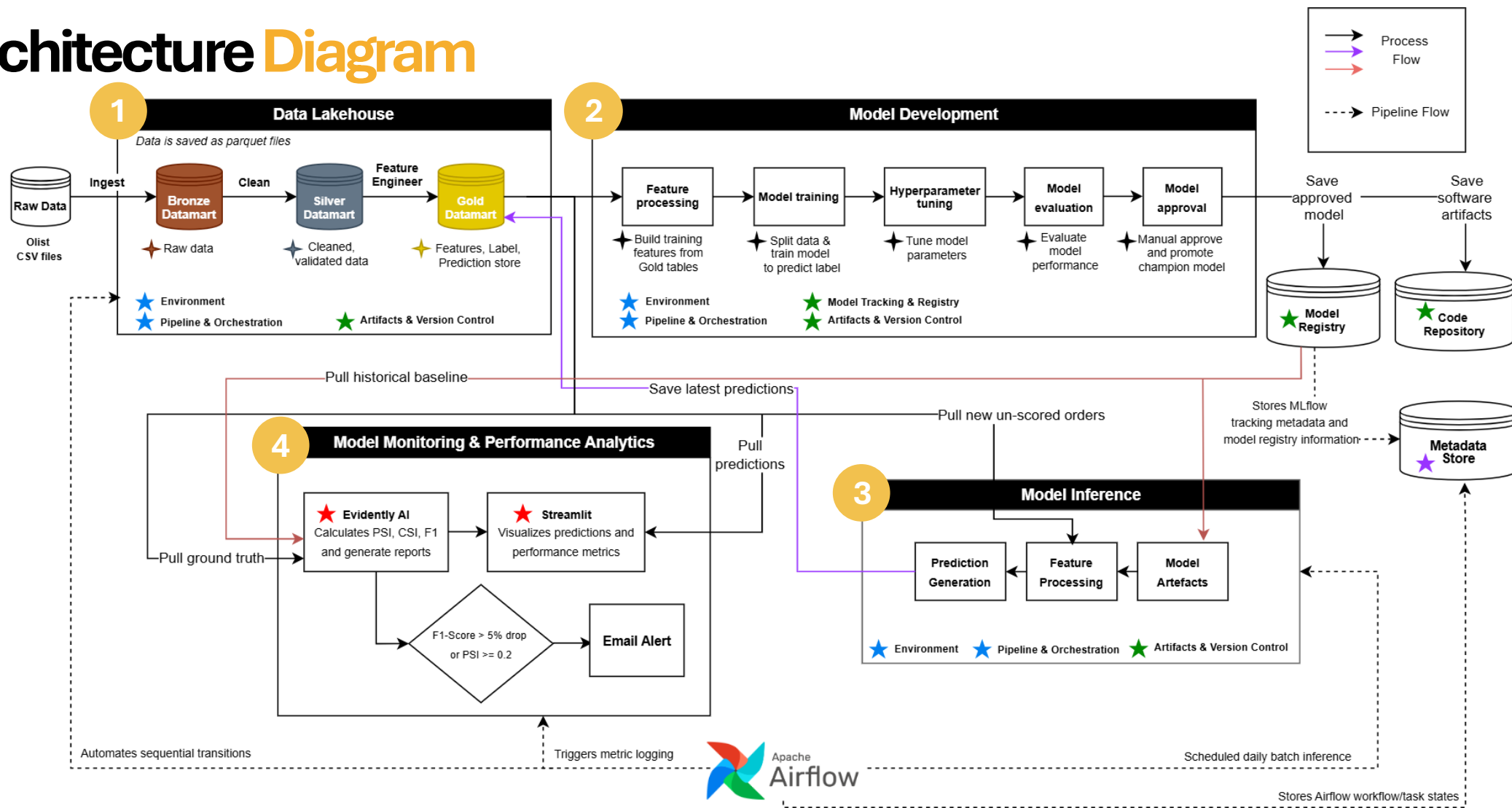
Given vector of features (X), predict late delivery status ($y \in \{0, 1\}$) where 1 corresponds to late delivery while 0 corresponds to on-time delivery



Dataset

- Sourced from Kaggle and covers 100k orders from 2017 to 2018
- Consists of 8 different files:
 - orders
 - order_items
 - order_payments
 - order_reviews
 - products
 - customers
 - sellers
 - geolocation
- Features cover the end-to-end order lifecycle (order details, shipment details, reviews, etc.)

Architecture Diagram



<p>★ Environment</p> <p>Package scripts and dependencies into containers</p>	<p>★ Pipeline & Orchestration</p> <p>Setup & trigger the end-to-end pipeline</p>	<p>★ Model Tracking & Registry</p> <p>Store experiment results and save model artifacts</p>	<p>★ Metadata Store</p> <p>Store metadata for Apache Airflow and mlflow</p>	<p>★ Artifacts & Version Control</p> <p>Manage project code, DAG files, and documentation</p>	<p>★ Model Monitoring</p> <p>Calculate PSI, CSI, F1 and generate reports</p>	<p>★ Observability Dashboard</p> <p>Centralize monitoring of performance metrics and predictions</p>
--	--	---	---	---	--	--

Data Lakehouse: Medallion Architecture

Bronze Layer

- **Creates immutable copy** of raw data
- Converts CSV files to compressed Parquet for efficient storage
- Append-only and idempotent, with lineage fields tracking when, where, and which run each record came from

Silver Layer

- Enforces schema, removes duplicates, validates keys, and flags every record with a data quality pass/fail
- **Handles domain-specific fixes** (e.g. conflicting geolocation data) and uses Delta Lake for tables with changing statuses
- Generates quality report for each table

Gold Layer

- **Builds training and inference tables** from a shared 29-feature pipeline with safeguards against target leakage
- Stores model predictions and joins them with features into an operational monitoring table that powers the fulfilment dashboard

Sample Quality Report

table	batch_date	metric	column	value
customers	6/3/2026	table		customers
customers	6/3/2026	row_count		96096
customers	6/3/2026	duplicate_rows		0
customers	6/3/2026	missing_values	customer_id	0
customers	6/3/2026	missing_values	customer_unique_id	0
customers	6/3/2026	missing_values	customer_zip_code_pr	0
customers	6/3/2026	missing_values	customer_city	0
customers	6/3/2026	missing_values	customer_state	0
customers	6/3/2026	missing_values	_ingest_date	0
customers	6/3/2026	missing_values	_batch_id	0
customers	6/3/2026	missing_values	_source_layer	0
customers	6/3/2026	missing_values	_source_file	0
customers	6/3/2026	missing_values	_ingested_at	0
customers	6/3/2026	missing_values	data_quality_remark	0
customers	6/3/2026	missing_values	data_quality_pass	0
customers	6/3/2026	data_types	customer_id	string
customers	6/3/2026	data_types	customer_unique_id	string
customers	6/3/2026	data_types	customer_zip_code_pr	string
customers	6/3/2026	data_types	customer_city	string
customers	6/3/2026	data_types	customer_state	string
customers	6/3/2026	data_types	_ingest_date	string
customers	6/3/2026	data_types	_batch_id	string
customers	6/3/2026	data_types	_source_layer	string
customers	6/3/2026	data_types	_source_file	string
customers	6/3/2026	data_types	_ingested_at	timestamp
customers	6/3/2026	data_types	data_quality_remark	string
customers	6/3/2026	data_types	data_quality_pass	boolean
customers	6/3/2026	distinct_counts	customer_id	96096
customers	6/3/2026	distinct_counts	customer_unique_id	96096
customers	6/3/2026	distinct_counts	customer_zip_code_pr	14982
customers	6/3/2026	distinct_counts	customer_city	4118
customers	6/3/2026	distinct_counts	customer_state	27
customers	6/3/2026	distinct_counts	_ingest_date	1
customers	6/3/2026	distinct_counts	_batch_id	1
customers	6/3/2026	distinct_counts	_source_layer	1
customers	6/3/2026	distinct_counts	_source_file	1
customers	6/3/2026	distinct_counts	_ingested_at	1
customers	6/3/2026	distinct_counts	data_quality_remark	1
customers	6/3/2026	distinct_counts	data_quality_pass	1

Model Development: Human-Governed Building & Deployment

Model Building & Calibration

Aim: a fair, comparable set of calibrated model candidates, which is reproducible and ready for human review.

- **3 model families:** Logistic Regression, Random Forest, XGBoost; covering linear, ensemble and boosting algorithm.
- **Chronological split preventing data leakage:** train / validation / test / OOT
- **Feature Selection:** Boruta - fit once, reused across all tuning trials
- **Class imbalance handling:** balance class weights per model family
- **Hyperparameter tuning:** 20-iteration random search per model
- **Probability calibration:** sigmoid for LR, isotonic for tree models so predicted probabilities reflect **real world likelihood**.

Champion Selection & Deployment

Aim: automatic shortlisting of models but the promotion into production is based on human-decision.

- **Metric-based shortlisting:** best per family by validation F1 (balances precision/recall for imbalanced class), then compared across model families to identify best-overall
- **Human review in MLflow:** DS team inspects calibration curves, feature importance, and edge-case behaviour
- **Champion sign-off & deployment:** DS team manually promotes the approved model to the MLflow Model Registry (no DAG)
- **Human guardrail by design**

The top screenshot shows the MLflow Experiments interface for the experiment 'olist-late-delivery-prediction'. It features a search bar, filters for 'Time created', 'State: Active', and 'Datasets', and a table of runs. The table has columns for 'Run Name', 'Created', and 'Dataset'. Three runs are listed, all created '22 hours ago' and associated with a 'RandomSearch' model.

Run Name	Created	Dataset
RandomSearch-ran...	22 hours ago	-
RandomSearch-logi...	22 hours ago	-
RandomSearch-xgb...	22 hours ago	-

The bottom screenshot shows the MLflow Registered Models interface. It features a search bar and a table of registered models. The table has columns for 'Name', 'Latest version', 'Aliased versions', 'Created by', 'Last modified', and 'Tags'. Four models are listed, all created on '2026-06-21 15:3...'. The 'olist-best-overall' model is marked as '@ champion'.

Name	Latest version	Aliased versions	Created by	Last modified	Tags
olist-best-logistic_regression	Version 1			2026-06-21 15:3...	-
olist-best-overall	Version 5	@ champion . Version		2026-06-21 15:3...	-
olist-best-random_forest	Version 1			2026-06-21 15:3...	-
olist-best-xgboost	Version 1			2026-06-21 15:3...	-

MLOps Workflow: Automated Orchestration

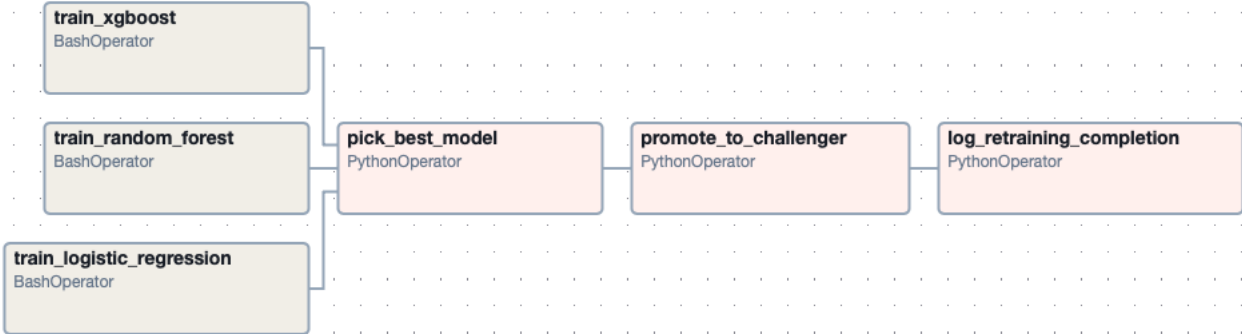
Orchestration

- **Apache Airflow DAG:** Data pipeline (Bronze → Silver → Gold) → Batch Inference → Monitoring
- **Failure isolation:** each stage can be re-run independently.
- **Champion-only inference:** same preprocessing as training, no training-serving skew.

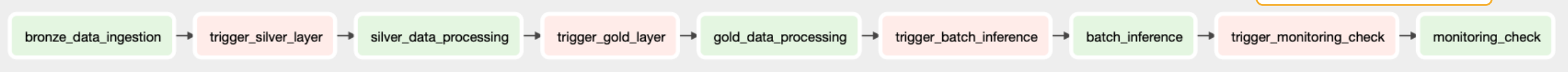
DAG	Owner	Runs	Schedule	Last Run	Next Run	Recent Tasks	Actions	Links
batch_inference	group_8	5	None	2026-06-22, 13:47:56		0	[Play] [Stop] [More]	
bronze_data_ingestion	group_8	4	@daily	2026-06-21, 15:43:04	2026-06-22, 08:00:00	12	[Play] [Stop] [More]	
fulfillment	group_8	1	None	2026-06-22, 13:58:39		3	[Play] [Stop] [More]	
gold_data_processing	group_8	4	None	2026-06-22, 13:46:50		5	[Play] [Stop] [More]	
monitoring_check	group_8	5	None	2026-06-22, 13:48:19		5	[Play] [Stop] [More]	
retrain_volume_trigger	group_8	3	0 6 ***	2026-06-22, 13:57:44	2026-06-22, 14:00:00	1	[Play] [Stop] [More]	
retraining_dag	group_8	1	None	2026-06-21, 15:32:47			[Play] [Stop] [More]	
silver_data_processing	group_8	4	None	2026-06-22, 13:45:51			[Play] [Stop] [More]	

Sub-DAGs

Sample tasks per sub-DAG



Overall DAG



MLOps Workflow: Human-Governed Monitoring and Retraining

Monitoring

What is it for? Model and data monitoring so degradation is caught before it affect downstream users.

Who is it for? Data Science team to investigate and make decision.

How does it work? Evidently report run automatically after each inference cycle and are surfaced in a Streamlit dashboard. Four signals are tracked, each with a defined threshold:

- **F1 7-day moving average drops > 5%** below validation baseline
- **Prediction Drift (PSI) ≥ 0.2**
- **Feature Drift (CSI) ≥ 0.2**
- **Data freshness check: $\geq 10k$ new Gold rows since last retrain**

If any threshold is breached, **alert email will be sent to Data Science team for further investigation.**

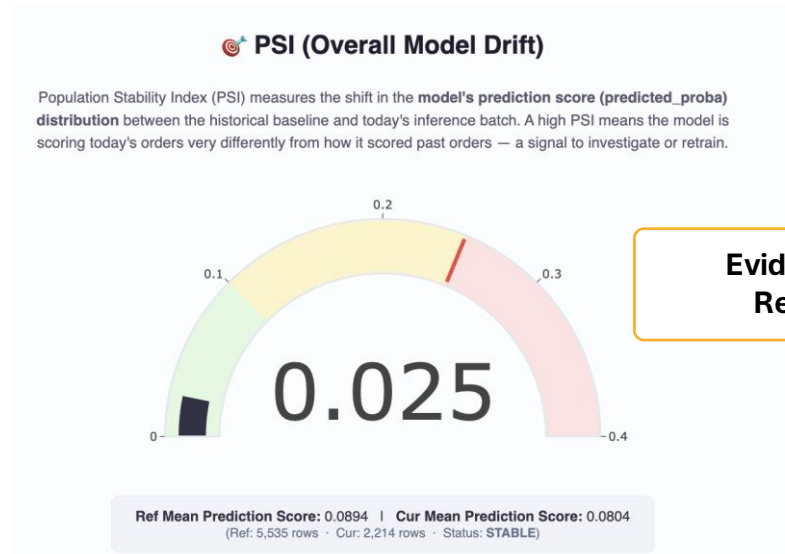
Retraining

Principle: investigate first, resolve root cause. Retraining is the last resort, not the default response to every alert.

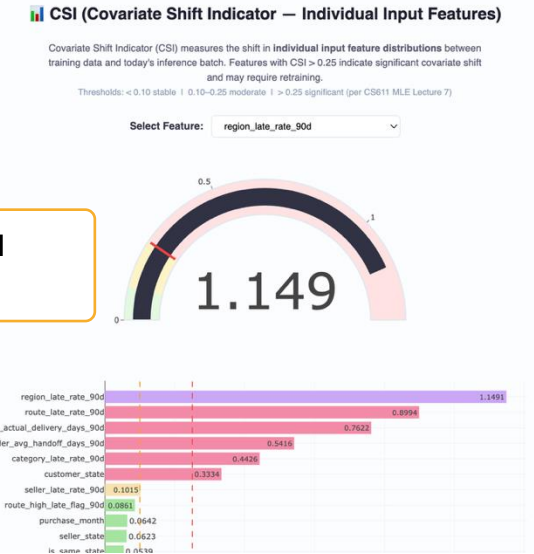
Why human-governed? Not every alert is real drift (e.g. temporary data quality). Automating retraining on every alert would be noisy and costly.

Retraining trigger: DS team initiated via Airflow, which will run all 3 models in parallel for retraining.

After retraining: champion swap still requires manual approval, same governance as initial deployment.



Evidently AI Reports



Streamlit Dashboard

Email Alert

Olist MLE - Model Monitoring Dashboard

Evidently AI · Data Quality · Feature Drift · PSI / CSI Stability

Data Quality · Feature Drift · PSI / CSI · Full Report

Total Rows	Features Monitored	Avg Null %	Quality Score
2,307	31	1.1%	0.989

Per-Feature Quality

Feature	Type	Null Count	Null %	Status	Unique	Mean	Std	Outliers (±5σ)
seller_late_rate_90d	Numeric	276	12.0%	Watch	109	0.0775	0.0646	3
seller_avg_handoff_days_90d	Numeric	276	12.0%	Watch	395	2.9757	1.8304	9
route_late_rate_90d	Numeric	96	4.2%	OK	59	0.0789	0.0257	2
route_avg_actual_delivery_days_90d	Numeric	96	4.2%	OK	78	13.0832	3.9366	0
category_late_rate_90d	Numeric	52	2.2%	OK	53	0.0783	0.0168	9
region_late_rate_90d	Numeric	6	0.3%	OK	24	0.0773	0.0124	13
min_order_weight	Numeric	0	0.0%	OK	521	2151.7200	3906.9598	18

cs611.mle.grp8@gmail.com
to me

Olist Late-Delivery Model - Drift Alert

Monitoring thresholds were breached. Manual investigation is recommended before triggering retraining_dag.

Alert reasons

- PSI ≥ 0.2 for 3 features: route_high_late_flag_90d (test alert)

Metrics

Metric	Value
Validation baseline F1	0.3500
F1 7-day moving average	0.2800
F1 drop %	20.0%
Max PSI	2.100
Features above PSI threshold	route_high_late_flag_90d

This alert was generated automatically by the monitoring_check DAG. Retraining will NOT start without a manual trigger.

Downstream Usage: Fulfillment Dashboard



What is it for?

Forward-looking view of late-delivery risk at dispatch time, enabling proactive intervention before customer impact.



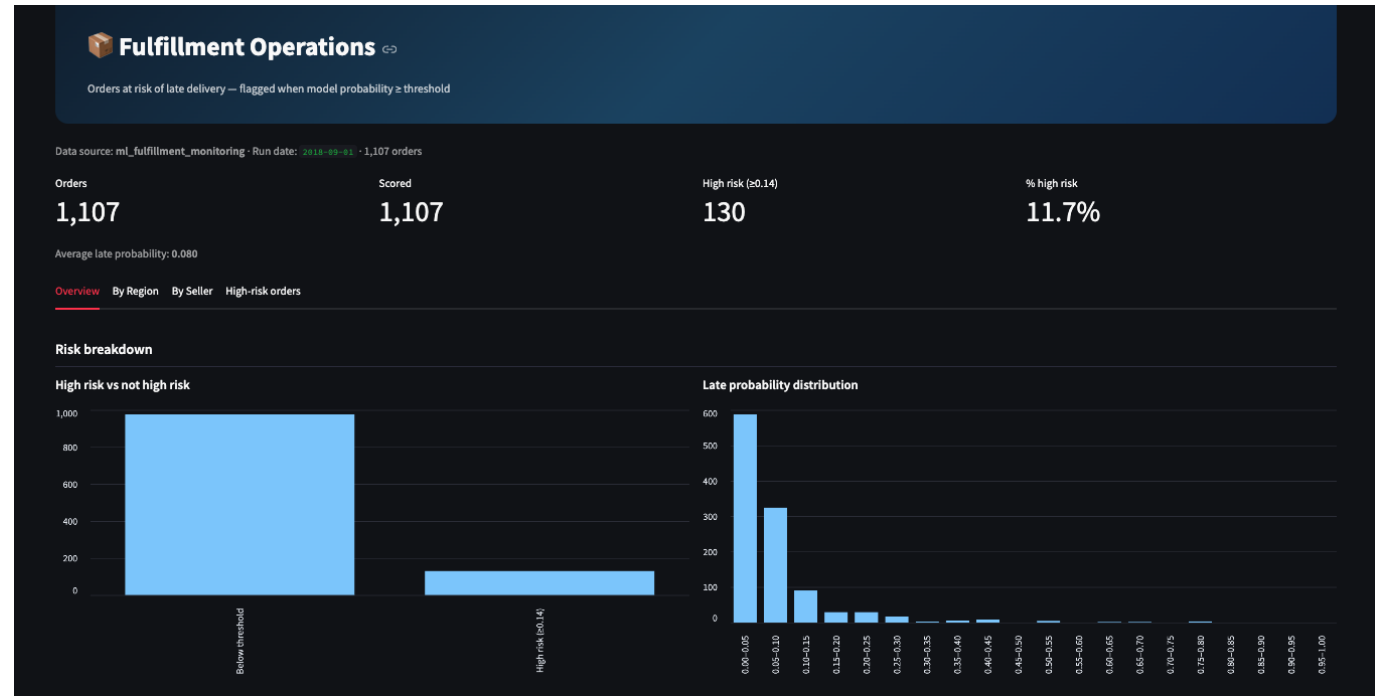
Who is it for?

Logistics and Customer Service Teams to expedite and reroute orders, and notify customers.



How it works?

Powered by Gold ml_fulfillment_monitoring datamart. It is auto-refreshed during each inference cycle.



Overview | By Region | By Seller | **High-risk orders**

Average late probability: 0.080

Overview | By Region | By Seller | High-risk orders

Customer region summary

One row per delivery state. High-risk = late probability \geq threshold.

Filter regions

Choose an option

order_id	customer_state	seller_state	primary_seller_id	ml_late_probability	high_late_probability_flag	estimated_delivery_days	seller_late
0c635f2fc897dc04b56643f997f25d33	MA	SP	cc419e0650a3c5ba77189a1882b7556a	0.2279	1	30.3034	
7e63ea5a0a34684140ae0e51ebe53035	PI	SP	ea8482cd71df3c1969d7b9473ff13abc	0.2279	1	28.5563	
38ff29b6f54b264f8992081bcd3f3c6e	ES	SP	966cb4760537b1404caedd472cc610a5	0.2279	1	24.256	
60d3a610dc0f2e4967d8d1fc1d5313c8	CE	SP	ea8482cd71df3c1969d7b9473ff13abc	0.2279	1	24.2279	
0538bda829ac14e64a201dc84fb1ba3b	PI	SP	7c67e1448b00f6e969d365cea6b010ab	0.2279	1	37.0268	

Customer region	Orders	High-risk orders	Avg late probab	Avg seller late rate (500)	Avg region late rate (500)	Avg est. delivery days	% high-risk
SP	328	37	0.089	0.052	0.076	18.7	11.3
BA	68	21	0.129	0.053	0.048	28.1	30.9
RJ	289	12	0.056	0.043	0.041	26.4	4.2
CE	38	11	0.160	0.052	0.005	29.0	28.9
MA	17	10	0.193	0.051	0.148	30.3	58.8
AL	9	5	0.170	0.037	0.051	26.8	55.6

Limitations & Future Works

1. Manual mode retrain and swap loop

Today:

Retraining and Champion model swap require human triggers, but these functions have already been implemented.

Limitation:

Slower response as relying on DS team investigation.

Future:

Steady state: alert → auto-retrain → challenger → human approval → champion swap.

2. On-premise architecture

Today:

Pipeline runs on a single Docker host.

Limitation:

No central store for predictions, MLflow runs, or monitoring artefacts.

Future:

Provision cloud infrastructure and container registry.

3. Model enhancement

Today:

Orders scored once, at "shipped" status.

Limitation:

No re-prediction as the order progresses.

Future:

Re-score at each stage of the fulfilment lifecycle, progressively refining risk estimates as more signal arrives over time.