

Executive Summary: Last Mile Simulation + ML Dispatch POC

This project is a compact proof-of-concept that mirrors a common last-mile analytics workflow: simulate operations under uncertainty, learn delay risk from operational features, and improve dispatch decisions using that learned signal. The goal is measurable customer impact (on-time delivery) while tracking cost and utilization trade-offs.

What's included

1) Discrete event simulation (orders, stations, drivers, traffic). 2) Delay risk model (tree-based classifier). 3) Dispatch policies: FIFO baseline vs ML-aware prioritization. 4) A/B experiment framework with per-day paired comparisons and KPIs.

KPIs

Primary: On-time delivery rate (SLA). Secondary: Average and P90 lead time, cost per package, assignment rate, overtime cost proxy.

How to run

```
python run_experiment.py --days 14 --drivers 60 --stations 6 --seed 7
```

Outputs are written to reports/: results_summary.json, results_runs.csv, and plots. In a real deployment, the same structure can be wired to production telemetry, with online feature updates and a more detailed routing model.

Architecture + Extension Ideas

Architecture (logical)

Synthetic demand → Simulator → Labeled outcomes → Model training → Predicted risk → Dispatch policy → Simulator (treatment) → KPI comparison + significance checks.

Extensions that map to real Amazon work

- Add station-specific capacity constraints and sortation cutoffs.
- Replace heuristic dispatch with a small optimization model (assignment / MILP).
- Add experiment layers: traffic scenarios, staffing scenarios, seasonal demand spikes.
- Benchmark multiple models with latency/quality trade-offs and build a lightweight model registry.