FedEx Product Ops — Decision Support Report

Executive Brief :

Purpose: Convert daily network ops data into an auditable KPI summary + decision signals with recommended actions.

Data: Portfolio-safe synthetic dataset shaped like an ops feed. Metrics are computed directly from source columns (see Data Lineage).

Current performance snapshot: Capacity utilization = 93.1%; on-time rate = 94.9%; exceptions = 188.4/day; labor efficiency = 83.0 pkgs/hr; cost per package = $5.00.

What’s driving risk: Disruption scenarios are typically the main driver of service degradation; confirm using the Top Drivers tables and KPI trend charts.

Immediate actions (7 days): Execute capacity relief + investigate the worst on-time days (hub staffing, sort constraints, last-mile coverage).

Next actions (30 days): Add hub/region drilldowns and alerting so leadership can see where issues originate, not just that they exist.

Sriyan S

System Architecture — How This Tool Works

This report is generated by a small decision-support pipeline designed to be auditable and easy to extend.

Layer / Module	What it does
1) Data Layer	Input dataset: C:/schoolwork/Python portfolio v1/Fedex Product Ops Support/data/fedex_product_ops.csv (Synthetic dataset structured like an ops feed)
2) Metrics Engine (metrics.py)	Aggregates KPIs (volume, utilization, on-time, exceptions, labor efficiency, cost)
3) Rules Engine (rules.py)	Evaluates KPIs vs configurable thresholds and produces decision signals (category, severity, issue, rationale, action)
4) Report Generator (export_pdf.py)	Builds the PDF: context + scenario mix + KPI summary + data lineage + trends + drivers + signals + actions
5) Outputs	fedex_product_ops_report.pdf + charts in C:/schoolwork/Python portfolio v1/Fedex Product Ops Support/outputs/

Flow (high-level)

```
C:/schoolwork/Python portfolio v1/Fedex Product Ops Support/data/fedex_product_ops.csv
↓
metrics.py (compute_metrics)
↓
rules.py (generate_recommendations)
↓
export_pdf.py (assemble report + charts)
↓
fedex_product_ops_report.pdf
```

Design choice: rules stay deterministic and auditable (threshold policy), while the system can later add an optional AI summary layer without changing the underlying KPI calculations.

Report Context

Field	Value
Data Source	C:/schoolwork/Python portfolio v1/Fedex Product Ops Support/data/fedex_product_ops.csv
Days Analyzed	60
Date Range	2025-10-31 → 2025-12-29

Scenario Mix (Counts + Avg Metrics)

Scenario Definitions

Scenario	Definition
NORMAL	Baseline operating conditions with expected demand and staffing.
PEAK	Elevated demand periods such as seasonal or promotional volume spikes.
DISRUPTION	Network disturbances (e.g., weather, facility outages, transport delays).

Scenario	Definition
LABOR SHORTAGE	Reduced labor availability impacting throughput or service levels.

Scenario	Days	Avg Volume	Avg Util
NORMAL	42	124,643	88.8%
PEAK	7	147,740	105.4%
DISRUPTION	7	127,730	91.3%
LABOR SHORTAGE	4	129,012	92.7%

Scenario	Avg On-Time	Avg Exceptions	Avg Labor Eff (pkgs/hr)	Avg Cost
NORMAL	96.5%	159	83.2	$4.66
PEAK	95.7%	181	89.8	$4.91
DISRUPTION	91.0%	298	75.4	$5.81
LABOR SHORTAGE	93.4%	218	93.6	$5.08

KPI Summary (recent 14 days)

Metric	Value
Avg Package Volume	130,328
Capacity Utilization	93.1%
On-Time Delivery Rate	94.9%
Avg Daily Exceptions	188
Labor Efficiency (pkgs/hr)	83.0
Avg Cost per Package	$5.00

Data Lineage — How Metrics Are Calculated

Metric	Formula	Source Columns (fedex_product_ops.csv)
Avg Package Volume	mean(package_volume)	package_volume
Capacity Utilization	mean(package_volume / network_capacity)	package_volume, network_capacity
On-Time Delivery Rate	mean(on_time_rate)	on_time_rate
Avg Daily Exceptions	mean(exceptions)	exceptions
Labor Efficiency (pkgs/hr)	mean(package_volume / labor_hours)	package_volume, labor_hours
Avg Cost per Package	mean(cost_per_package)	cost_per_package

Decision Signals

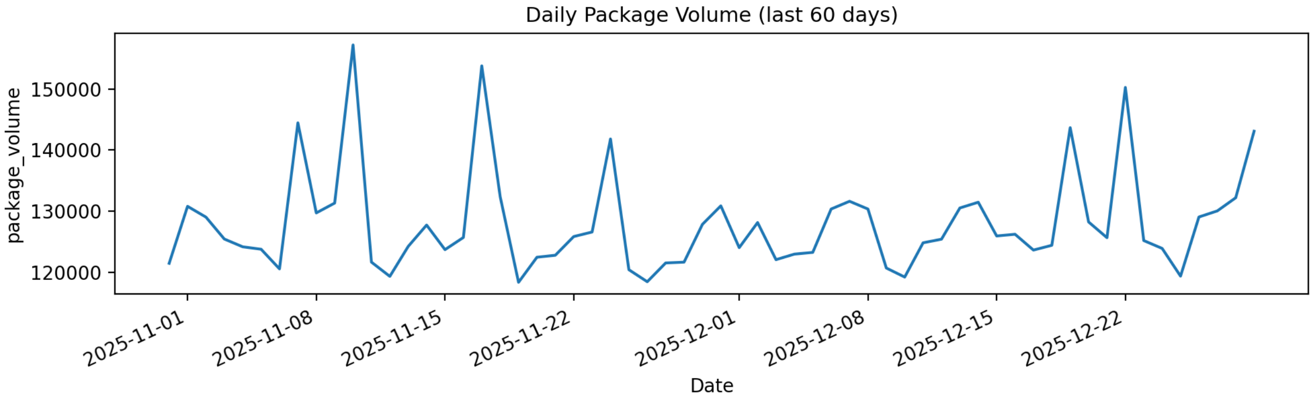
Category	Severity	Issue
SERVICE	MEDIUM	Capacity utilization trending toward saturation.
SERVICE	HIGH	On-time delivery performance below target.

Rationale	Action
Capacity utilization = 93.1% in warning band (90.0%–95.0%).	Monitor volume growth and prepare contingency capacity plans.
On-time rate = 94.9% < 96.0% target.	Investigate hub bottlenecks and last-mile staffing coverage.

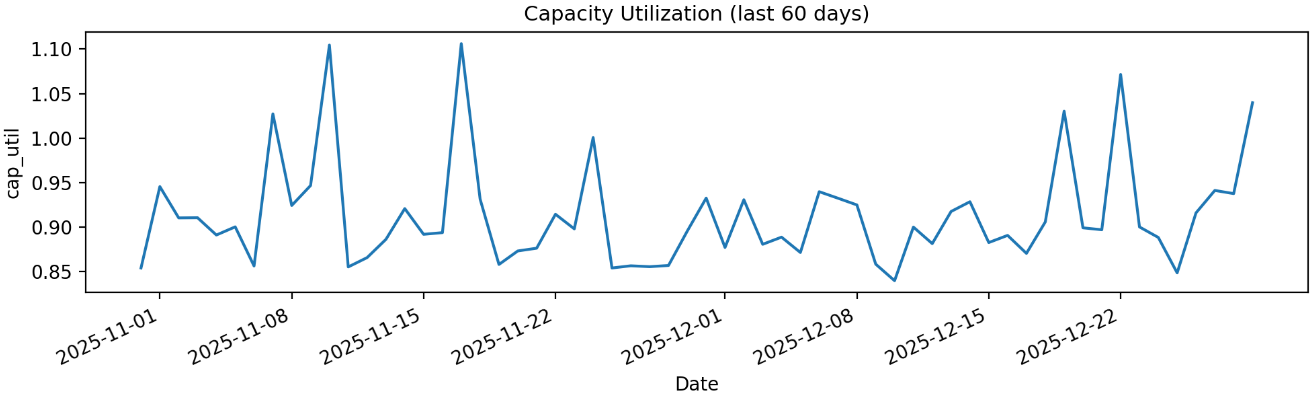
KPI Trends (last 60 days)

These charts visualize the same source columns used in the KPI calculations.

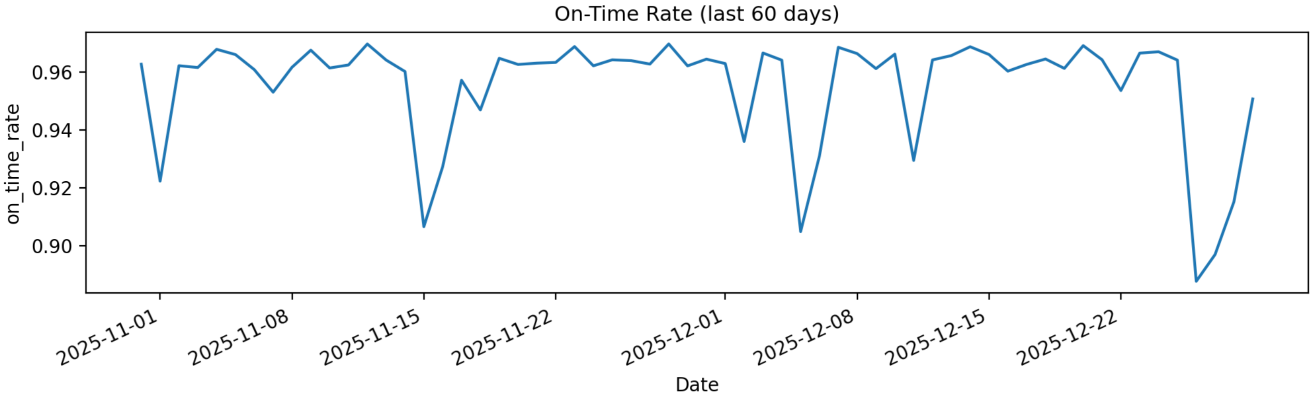
Daily Package Volume



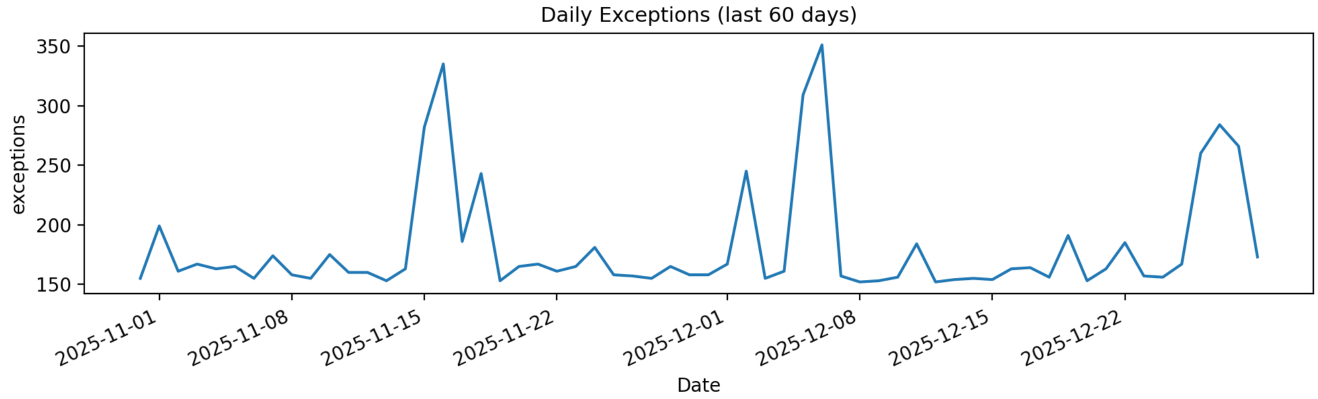
Capacity Utilization



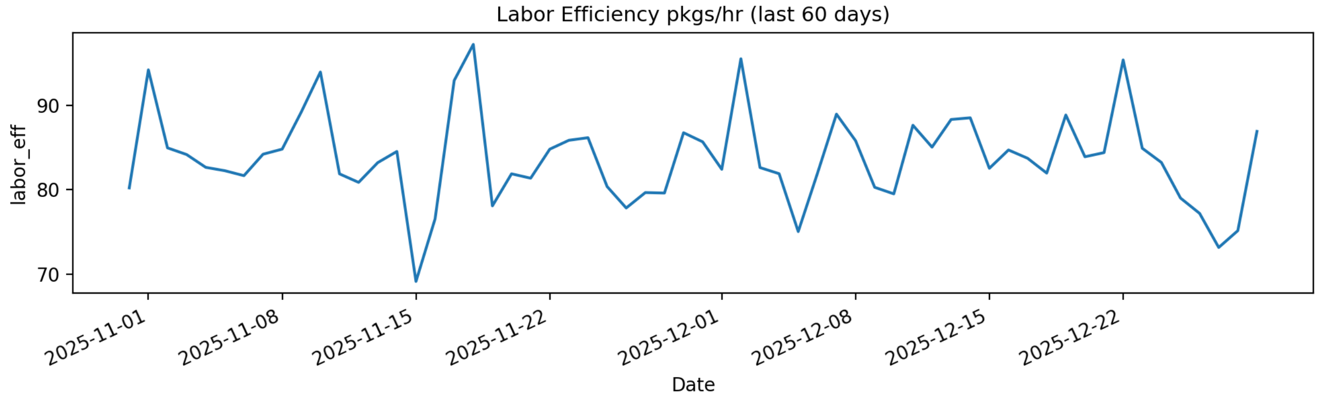
On-Time Rate



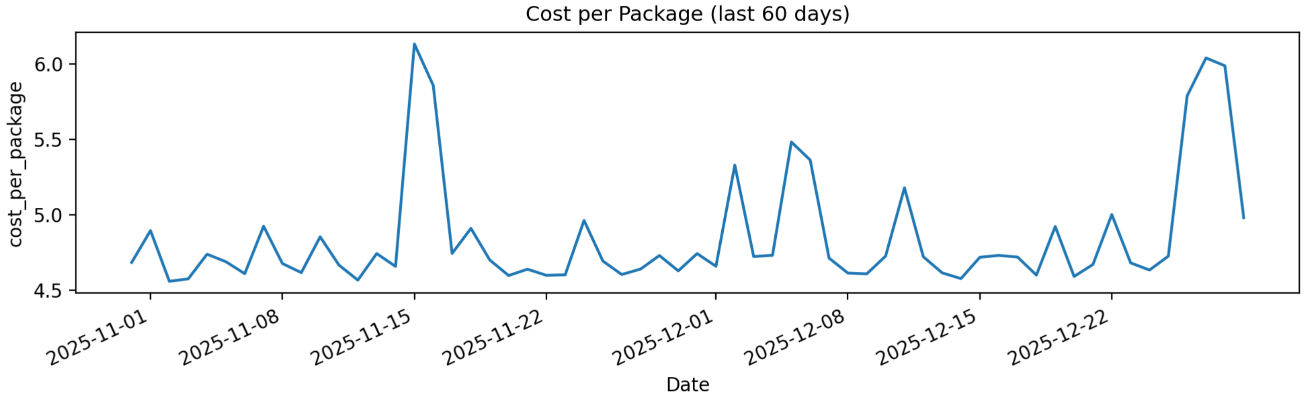
Daily Exceptions



Labor Efficiency



Cost per Package



Top Drivers — Dates to Investigate

Specific dates most responsible for poor performance or cost.

Worst On-Time Days (bottom 5)

Date	Scenario	On-Time	Volume	Exceptions
2025-12-26	DISRUPTION	88.8%	129,023	260
2025-12-27	DISRUPTION	89.7%	130,020	284
2025-12-05	DISRUPTION	90.5%	123,225	309
2025-11-15	DISRUPTION	90.7%	123,675	282
2025-12-28	DISRUPTION	91.5%	132,170	266

Highest Exception Days (top 5)

Date	Scenario	Exceptions	On-Time	Volume
2025-12-06	DISRUPTION	351	93.1%	130,329
2025-11-16	DISRUPTION	335	92.7%	125,670
2025-12-05	DISRUPTION	309	90.5%	123,225
2025-12-27	DISRUPTION	284	89.7%	130,020
2025-11-15	DISRUPTION	282	90.7%	123,675

Highest Capacity Utilization Days (top 5)

Date	Scenario	Cap Util	Capacity	Volume
2025-11-17	PEAK	110.6%	139,015	153,772
2025-11-10	PEAK	110.5%	142,321	157,199
2025-12-22	PEAK	107.2%	140,215	150,241
2025-12-29	PEAK	104.0%	137,607	143,072
2025-12-19	PEAK	103.0%	139,438	143,654

Next Actions (7 Days / 30 Days)

7-Day

- Validate the bottom-5 on-time days with lane/hub notes and confirm root-cause category (hub constraint vs last-mile staffing vs weather).
- Run a 1-week capacity relief plan: reroute volume to underutilized hubs or add temporary linehaul capacity during peak windows.
- Create an exception driver Pareto (top reason codes) and deploy 1–2 process controls to reduce the highest-frequency exception types.
- Confirm if cost/pack spikes correlate with disruption days (overtime, re-handling, re-routes).

30-Day

- Add feature-level breakdowns (by region/hub/lane) to pinpoint where service degradation is concentrated.
- Introduce an alerting threshold policy (targets per KPI) and log signal history to prove the system catches issues early.
- Build a simple scenario impact estimator: predicted on-time/cost deltas under PEAK vs DISRUPTION conditions.
- Optionally add an AI summary layer: generate a short exec brief from metrics + drivers + recommendations (keep deterministic logic intact).

Roadmap — What This Tool Would Do Next (Product Ops Enhancements)

- Drilldowns (hub/region/lane) to localize root cause.
- Driver attribution for exceptions and service misses.
- Early-warning alerts (trend-to-threshold) instead of threshold-only.
- Policy tuning via config profiles (PEAK vs NORMAL).
- Optional AI narrative layer that does not alter KPI calculations.