Torchline Solutions

Supply Chain Optimization AI

Reduced manual reporting steps by 40 percent using custom AI agents built with AI, automation, and Python.

User	Problem	Outcome
Supply chain operations, planning, and analytics teams needing faster weekly reporting and exception visibility.	Reporting required manual extraction, cleanup, and narrative writing across multiple sources, creating delays and inconsistent results.	Manual reporting steps reduced by 40 percent through automated data preparation and AI drafted insights with a review workflow.

Context and constraints

- Weekly reporting cadence with frequent ad hoc requests
- Data spread across spreadsheets, exports, and operational systems
- Traceability needed from insight back to source rows
- Output must be consistent in format and easy to review
- Adoption depends on human control with easy edits

Objective

Create an AI assisted workflow that converts raw supply chain data into a reviewable weekly report with clear exceptions, drivers, and recommended actions.

Hypothesis and success metrics

Hypothesis

If data collection, cleaning, and first draft analysis are automated, the team spends less time assembling reports and more time acting on exceptions.

Success metrics

- Manual reporting steps reduced by 40 percent
- Report cycle time decreases week over week
- Exception coverage improves with fewer missed issues
- Stakeholder satisfaction improves through consistent output
- Review edits decrease over time as rules and prompts improve

Solution design

A lightweight pipeline with three layers.

Data prep layer	Agent layer	Delivery layer
Python scripts ingest exports, standardize fields, and validate data quality.	Custom AI agents generate insights, detect exceptions, and draft narratives with citations back to source fields.	Automation publishes the report package and notifies stakeholders on a predictable schedule.

What we built

AI agents

- **Data quality and anomaly scan** identifies missing values, outliers, and suspicious changes in lead time, demand, inventory, and supplier performance.
- **Exception summarizer** groups issues by severity and category such as stockout risk, demand spikes, and late shipments.
- **Driver analysis drafter** explains what changed and why in plain language, referencing underlying fields.
- **Action recommendation generator** suggests next actions with confidence notes and required human validation.

Automation workflow

- Scheduled run each week plus manual runs on demand
- Pull newest exports and run Python validation and metrics
- Generate an exception set with priority scoring
- Draft narratives via agents constrained to approved categories
- Compile the report package and notify stakeholders

Python components

- Data ingestion, cleaning, and schema validation
- Metric calculations such as service level, fill rate, and inventory turns
- Delta detection week over week and anomaly flags
- Exception scoring logic to prioritize issues

Results

- Manual reporting steps reduced by 40 percent
- Improved consistency of weekly reporting structure
- Faster time to first draft report
- Higher visibility into exceptions through standard categories
- Clear review workflow that preserved human control

What made it work

- Human in the loop review for AI drafted narratives
- Deterministic validation in Python before AI touched the data

- Standardized output format that stakeholders recognized
- Continuous improvement through feedback on agent outputs

Risks and mitigations

Risk
AI may hallucinate or overstate causes.

- Require the agent to cite source fields in the dataset
- Constrain outputs to approved categories
- Use confidence labels and require review for low confidence items
- Log each run for audit and iteration

Artifacts to include

- Workflow diagram showing data sources, Python processing, AI agents, and report output
- Screenshot of the exception report view
- Before and after reporting checklist showing reduced steps

Next iterations

- Add a feedback capture step to improve rules and prompts
- Add automated root cause suggestions tied to known constraints
- Expand to daily exception alerts for high severity issues
- Integrate with a ticket workflow so actions are tracked to closure

Role and contribution

- Defined the problem, success metrics, and workflow
- Designed agent responsibilities and review checkpoints
- Built the Python data pipeline and validation rules
- Implemented automation triggers and delivery
- Iterated prompts and rules based on stakeholder feedback