Predicting Late Deliveries Using Machine Learning & Operational Diagnostics

Author: Sarah Hirani

Tools: Python, Pandas, Scikit-Learn, Seaborn

Executive Summary

Delivery performance is one of the most important indicators of supply chain health. Delays increase operational costs, disrupt inventory cycles, destabilize capacity planning, and erode customer trust.

This project leverages operational data to build a predictive model that identifies orders likely to be delayed *before* the delay occurs.

Beyond the algorithm itself, a combination of exploratory analysis, feature engineering, and domain understanding was used to uncover systemic root causes.

This approach reflects how **predictive analytics** connects with **supply chain strategy**, **capacity management**, and **continuous improvement initiatives** in real-world organizations.

Context & Real-World Problem

Late deliveries typically occur because of compounding operational risks, such as:

- inaccurate lead-time assumptions
- batching delays in fulfillment
- carrier performance inconsistencies
- SKU-level production bottlenecks
- seasonal spikes that overwhelm capacity
- lack of real-time visibility
- demand surges outpacing labor or carrier availability

In industry, these issues lead to:

- higher expedited freight spend
- reduced OTIF (on-time-in-full) scores
- inventory imbalances
- order backlogs
- lost sales opportunities
- lower customer satisfaction

Organizations often address these issues reactively.
This model aims to shift the business toward **proactive, predictive intervention**.

Dataset & Operational Relevance

The DataCo Supply Chain dataset mirrors a real-world transactional system, containing:

- order-level timestamps
- customer geography
- fulfillment scheduling windows
- product-level attributes
- pricing, discounts, and profitability
- delivery status
- shipping modes (Standard, Second Class, Same Day, etc.)

These represent the same variables tracked in WMS, OMS, ERP, and TMS systems in real supply chain operations.

The model uses these variables to replicate what a real logistics command center would monitor.

Data Preparation & Engineering

To model delivery risk accurately, it was critical to transform operational timestamps into meaningful features.

Key engineered features:

1. Actual Shipping Lead Time

Formula: Actual Shipping Lead Time=Shipping Date−Order Date

Python Code: `df["actual_lead_time"] = (df["shipping_date"] - df["order_date"]).dt.days`

Business relevance:
Indicates internal handling speed, warehouse throughput, and order processing delays.

2. Scheduled vs Actual Variance
Formula: Lead Time Variance=Days Scheduled−Days Actual

Python Code: `df["lead_time_variance"] = df["Days for shipment (scheduled)"] - df["Days for shipping (actual)"]`

Business relevance:
Highlights where planning buffers fail — a major problem in high-SKU environments.

3. Shipping Mode Categorization

Categorical → encoded numerically
Business relevance:
Carrier/service levels have direct correlation with SLA adherence.

4. Price- and Margin-Based Features

- discount rates
  - Discount Ratio = Order Item Discount / Order Item Product Price
  - Python Code: df["discount_ratio"] = df["Order Item Discount"] / df["Order Item Product Price"]
- order profitability
- unit economics
  - Price Per Unit = Order Item Total / Order Item Quantity
  - Python code: df["price_per_unit"] = df["Order Item Total"] / df["Order Item Quantity"]

Business relevance:
High-value orders may receive prioritization, but inconsistencies can reveal scheduling misalignment.

5. Regional & Route Patterns

Encoding of:

- order city/country
- customer region
- order state

Business relevance:
Regional cost-to-serve and route reliability issues become quantifiable.

These engineered features allow the model to identify operational inefficiencies that would not be obvious from raw timestamps alone.

Machine Learning Model & Evaluation

Python: df["late_delivery_flag"] = df["Delivery Status"].apply(lambda x: 1 if x == "Late" else 0)

Interpretation: df["late_delivery_flag"] = df["Delivery Status"].apply(lambda x: 1 if x == "Late" else 0)

A Random Forest Classifier was selected due to its ability to:

- handle nonlinear relationships
- learn complex patterns across mixed data
- maintain accuracy even with noisy variables
- provide meaningful feature importance insight

Performance Metrics:

- **97% accuracy**
- **0.96–0.98 precision**
- **0.94–0.98 recall**
- **0.97 ROC-AUC**

These numbers indicate the model can reliably distinguish late vs on-time orders — especially valuable for capacity planning teams.

Feature Importance & Interpretation

This is where the model provides operational value, not just predictions.

Top predictive factors:

1. Shipping Mode

Standard and Second Class correlated strongly with delays due to:

- longer transit times
- batching behavior by carriers
- lower prioritization in mixed-load routes

2. Lead Time Variability

Orders with inconsistent lead times signal:

- unstable warehouse workload
- lack of labor planning
- ineffective slotting or picking efficiency

2. Lead Time Variability

Orders with inconsistent lead times signal:

- unstable warehouse workload
- lack of labor planning
- ineffective slotting or picking efficiency

4. High-Value / Low-Quantity SKUs

These SKUs often suffer because:

- they require specialized handling
- they may be stored separately
- prioritization rules can be inconsistent

Operational Impact & Scenario Analysis

Scenario 1: Reducing Late Deliveries by 10%

Using the model to identify high-risk orders early might result in:

- **reallocating labor** to high-risk shipments
- **adjusting carrier selection**
- **adding buffer time** to Standard/Second Class orders
- **prioritizing bottleneck SKUs earlier in the day**

Impact:
Could reduce reships and support calls while boosting OTIF.

Scenario 2: Improving Carrier Strategy

If Second Class shipments have the highest late-delivery correlation:

- negotiate SLA improvements
- reassign sensitive shipments to Express modes
- diversify carriers in weak regions

Outcome:
Better performance without expanding fleet or labor.

Scenario 2: Improving Carrier Strategy

If Second Class shipments have the highest late-delivery correlation:

- negotiate SLA improvements
- reassign sensitive shipments to Express modes
- diversify carriers in weak regions

Outcome:
Better performance without expanding fleet or labor.

Real World Applications

This model can be integrated into:

✓ TMS Systems

Highlight orders requiring prioritization.

✓ WMS Dashboards

Flag orders exceeding internal SLA thresholds.

✓ ERP Order Prioritization

Rank orders before printing pick lists.

✓ Carrier Performance Scorecards

Measure actual vs expected SLA by mode or route.

✓ Customer Service Alerts

Trigger warnings for orders likely to miss promised delivery windows.

Strategic Benefits

Implementing this model could lead to:

- better SLA compliance
- reduced expedited freight spend
- higher customer satisfaction (NPS, CSAT)
- improved warehouse flow
- smarter labor planning
- more predictable cash flow
- enhanced transparency across teams

Conclusion

This project demonstrates not just a high-performing predictive model, but an analysis that connects technical results with real-world supply chain decision-making.

The ability to translate data into operational strategy is what sets apart strong candidates in supply chain analytics and this project exemplifies that.