NexGen Logistics — Predictive Delivery Optimizer

Rahul Mishra

October 29, 2025

Abstract

Predict shipment delays before they happen using a tuned Random Forest ensemble model — and take proactive actions to reduce delays, costs, and customer churn.

1 Problem Statement

Delivery delays lead to:

- Penalties & operational losses
- Customer dissatisfaction
- Churn and missed revenue
- SLA violations We solve this by:
- Predicting high-risk shipments
- Prescribing targeted operational actions
- Preventing avoidable delays

2 Solution Overview

An end-to-end ML workflow including:

- Multi-source logistics data ingestion
- 30+ engineered features
- Tuned Random Forest delay classifier
- Probability-based risk scoring
- Actionable prescription engine
- Streamlit UI for user-friendly interaction

3 Machine Learning Approach — Random Forest

This project uses a RandomForestClassifier optimized via GridSearchCV.

Benefit	Impact

Bagging (Bootstrap sampling)	Lower variance, reduced overfitting
Random feature subset per split	Better generalization
Ensemble voting	Stability + accuracy
Probabilistic outputs	Risk-based prioritization

Table 1: Advantages of the Random Forest Model

Why Random Forest?

Final Tuned Model

Example

```
RandomForestClassifier(
n_estimators=100,
max_depth=None,
min_samples_split=2,
min_samples_leaf=1,
max_features='sqrt',
random_state=42 )
```

4 Comprehensive Model Evaluation Report

4.1 Overall Performance Metrics

Metric	Value
Accuracy	0.9737
Precision	1.0000
Recall	0.9444
F1-Score	0.9714
ROC-AUC	1.0000
Average Precision	1.0000

- Zero false alarms
- Nearly every real delay detected
- Perfect class separation (AUC = 1.0)

4.2 Detailed Classification Report

	precision	recall	f1-score	support

On-Time	0.95	1.00	0.98	20
Delayed	1.00	0.94	0.97	18

4.3 Confusion Matrix

```
[[20  0]
[ 1 17]]
```

Metric	Value
True Negatives	20
False Positives	0
False Negatives	1
True Positives	17

Only one delayed order was missed — can be improved via threshold tuning.

4.4 Business Interpretation

Outcome	Benefit
94% delays proactively detected	Reduced penalties & escalations
Zero unnecessary alerts	Saves operational time & costs
High trust & adoption	Better customer experience

Drives proactive logistics, boosts CSAT & operational efficiency.

5 Architecture

Data → Feature Engineering → Random Forest → Risk Scoring → Prescriptions → Business Impact → Streamlit Dashboard

6 Project Structure

```
nexgen_predictive_delivery_optimizer/
 app.py
 requirements.txt
 README.md
 innovation_brief_template.md src/
     data.py
     features.py
     model.py
    rules.py
```

```
utils.py models/
```

8 Prescriptive Actions

Automated recommendations include:

- Carrier Swap
- Route Optimization
- Priority Upgrade
- Vehicle Reassignment
- Warehouse Reroute
- Weekend Surge Planning • Proactive Customer Alerts Each includes:
- Priority
- Estimated delay reduction %
- Cost impact analysis

9 Testing with Synthetic Data

Sample data generation script included for demo & testing.

10 Future Enhancements

- SHAP Explainability
- FastAPI endpoint for real-time predictions
- Dockerized deployment
- Drift monitoring & auto-retraining
- Alerting integration (Slack / Email)
- Cost-aware optimization modeling

Author & License

Author: Rahul Mishra
GitHub Repository:
https://github.com/RahulMishra09/nexgen_predictive_delivery_ optimizer.git
APP Link:
https://nexgenpredictivedeliveryoptimizer-g7y7zpgu4uc8r3rq2dgtkp.streamlit.app/#200