DATA MINING

OPTIMIZING SUPPLY CHAIN AND SALES PERFORMANCE

TEAM MEMBERS

- NIKHILESH DHAVALE

&

ADITYA KARANDE

UNDER THE GUIDANCE OF

– PROF. SUSHOVAN MAJHI

1. Overview of the project

1.a Why we chose this topic

Late deliveries directly impact customer satisfaction, cost (expedited shipping, returns, penalties), and inventory management. Predicting which orders are likely to be late enables operational teams to prioritize interventions (rerouting, upgrading shipping modes, additional handling) and reduces downstream effects. The problem is concrete, measurable (binary late vs on-time), and directly actionable — making it an ideal applied data mining project.

1.b Prior research and analysis

Work in logistics and operations research shows that delivery delay is influenced by scheduling accuracy, carrier reliability, order size/complexity, regional differences, and temporal factors (seasonality, holidays). From a machine-learning perspective, similar operational classification problems are commonly addressed with:
- Logistic regression as an interpretable baseline,
- Tree-based ensemble models (Random Forests) for improved accuracy and non-linear interactions,
- Feature engineering around temporal and schedule vs actual delay variables,
- Model interpretation (feature importance, partial dependence, SHAP) to translate predictions into actions.

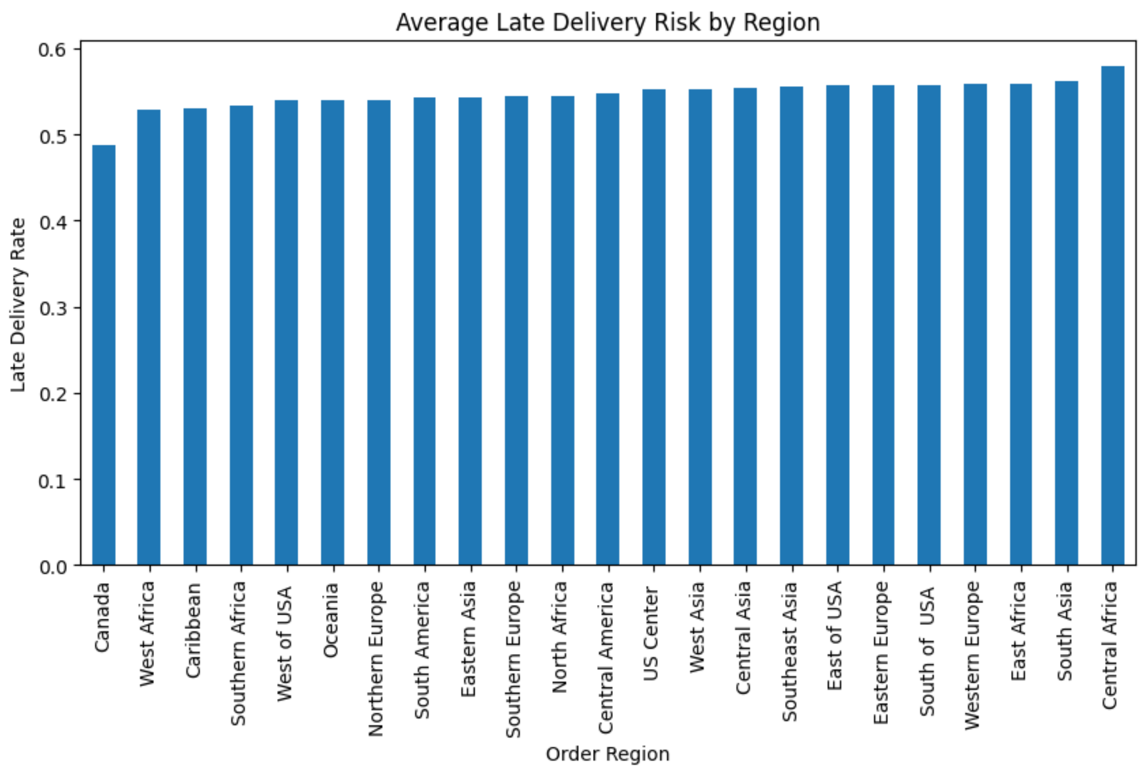
Our analysis workflow aligns with established best practices, incorporating thorough EDA, hypothesis testing, PCA for dimensionality assessment, a logistic regression baseline, and a Random Forest model with feature-importance evaluation1.c Information about the dataset(s) used.

File loaded: DataCoSupplyChainDataset.csv
Key target: Late_delivery_risk — binary label (0 = on-time, 1 = late).

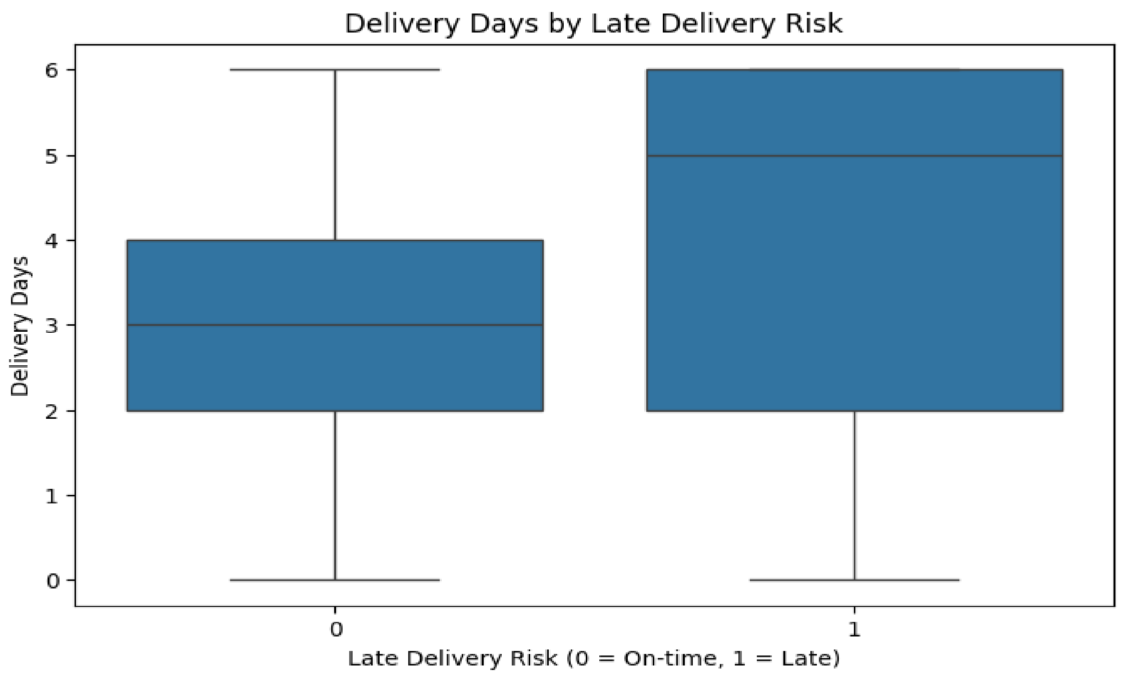
Key features used in models / EDA:
- Time and duration features (derived and raw): order_date, shipping_date (renamed and parsed), delivery_days = shipping − order, Days for shipping (real), Days for shipment (scheduled), delay_vs_schedule = (real − scheduled), order_month, order_weekday, etc.
- Categorical: Shipping Mode, Customer Segment, Order Region, Market, Product Category Id.
- Numeric: Order Item Quantity and other numeric fields in the CSV.

Cleaning steps:
1. Dropped columns: "Order Zipcode", "Product Description".
2. Filled Customer Lname missing values with "Unknown".
3. Filled Customer Zipcode missing values with 0.
4. Converted order and shipping date columns to datetime with pd to datetime.
5. Created delivery_days and delay_vs_schedule and extracted time components (year/month/day/hour/minute/weekday) from both order and shipping timestamps.
Rows / columns: [180519x 51].

1.d Unusual EDA results worth mentioning
High-level takeaways from EDA:
- Shipping Mode and Order Region display differences in late-delivery rates (visualized via countplots, bar charts, and a heatmap of late rate by shipping mode).

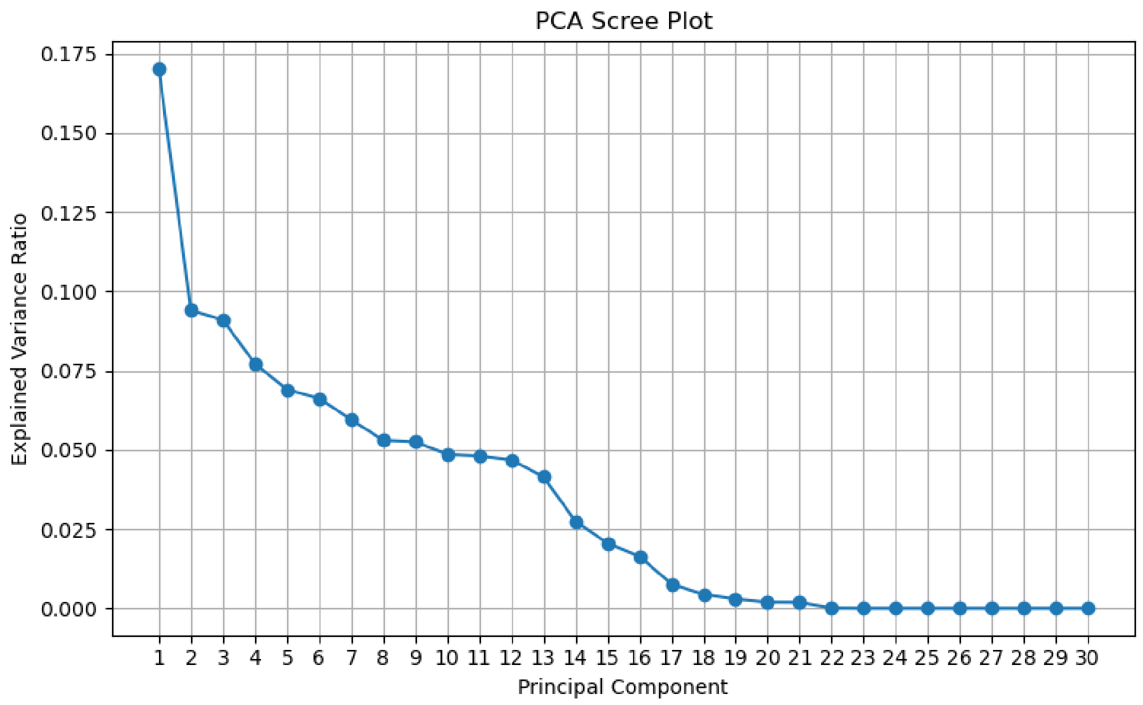


- delivery_days and delay_vs_schedule strongly differ between on-time and late orders.



- Several features have high cardinality, noted by df_supplychain.nunique(). These may require special handling.

- PCA suggests how much numeric variance can be compacted.



2. SMART QUESTION

1. Predictive (Specific & Measurable):
Can we predict whether an order will be late (1) or on time (0) using order/customer/product/shipping features with a test ROC-AUC ≥ 0.80 under a train/test split (80/20) and stratification by the target?

3. Machine-learning algorithms and evaluation

Preprocessing & pipeline summary:
- Selected features: delivery_days, delay_vs_schedule, Days for shipping (real), Days for shipment (scheduled), Shipping Mode, Customer Segment, Order Region, Market, order_month, order_weekday, Order Item Quantity, Product Category Id.
- Categorical columns encoded: Shipping Mode, Customer Segment, Order Region, Market (OneHotEncoder(handle_unknown="ignore")).
- Numeric columns standardized (for logistic regression) via

StandardScaler().
- Train/test split: 80/20 with stratification on the target and random_state=42.

Statistical tests, correlation analysis and PCA

The analysis performs a series of statistical and exploratory analyses, including a t-test or Mann–Whitney U test to compare *delivery_days* between late and on-time orders, a Chi-square test to examine the association between *Shipping Mode* and *Late_delivery_risk*, and ANOVA/Kruskal–Wallis tests to evaluate differences in *delivery_days* across shipping methods. In addition, a Pearson correlation matrix is generated to assess linear relationships among numeric features and to identify variables that are strongly associated with delivery risk or with each other. Finally, Principal Component Analysis (PCA) is applied to standardized numeric features to understand variance structure and dimensionality.

T-Test / Mann–Whitney U Test for Delivery Days:

In this project, the goal of the statistical test is to determine whether the number of delivery days differs between orders that arrive on time and those that arrive late. The analysis begins by splitting the dataset into two independent groups based on the Late_delivery_risk variable. Before selecting the appropriate test, the code evaluates the assumptions of the two-sample t-test:

- Normality is checked using the Shapiro–Wilk test for both groups.
- Equality of variances is assessed using Levene's test.

Results :

T-test: delivery_days (On-time vs Late)

- Counts:
  - On-time orders: 81,542
  - Late orders: 98,977

Normality Tests (Shapiro–Wilk):

- On-time group:
    - Statistic = 0.8778, p-value = 2.054×10^{-19}
- Late group:
    - Statistic = 0.8481, p-value = 1.597×10^{-21}

Equality of Variances (Levene's Test):

- Statistic = 12,724.8607
- p-value = 0.000

Conclusion:
Normality assumptions were not met → proceeding with a non-parametric test.
Mann–Whitney U Test:

- U-statistic = 2,367,419,214.50
- p-value = 0.000

Chi Squared Test :

The Chi-square test of independence was used to examine whether Shipping Mode and Late Delivery Risk are statistically associated. This test evaluates whether the distribution of late vs. on-time deliveries differs across shipping modes more than would be expected by chance. A contingency table was created to compare the frequency of late and on-time deliveries within each shipping mode category. If the Chi-square statistic is large and the p-value is below a significance threshold (typically 0.05), it indicates that shipping mode and delivery timeliness are not independent — meaning shipping mode plays a meaningful role in predicting late delivery risk.

Shipping Mode	On-Time (0)	Late (1)
First Class	1,301	26,513
Same Day	5,283	4,454
Second Class	8,229	26,987
Standard Class	66,729	41,023

Chi-Square Test Results

- Chi-square statistic: 37,716.0425
- Degrees of freedom (df): 3
- p-value: < 0.001

Anova Test :

To assess whether delivery times differ across shipping modes, the analysis compares the distribution of delivery_days among the categories *First Class*, *Second Class*, *Standard Class*, and *Same Day*.

Before selecting the appropriate test, the code checks whether each group satisfies the normality assumption required for ANOVA. Since delivery time data are typically skewed and the Shapiro–Wilk tests indicated non-normal distributions, the analysis defaults to the Kruskal–Wallis test, a non-parametric alternative to one-way ANOVA.

The Kruskal–Wallis test evaluates whether at least one shipping mode has a statistically different distribution of delivery days compared to the others.

Kruskal–Wallis Test: Delivery Days Across Shipping Modes

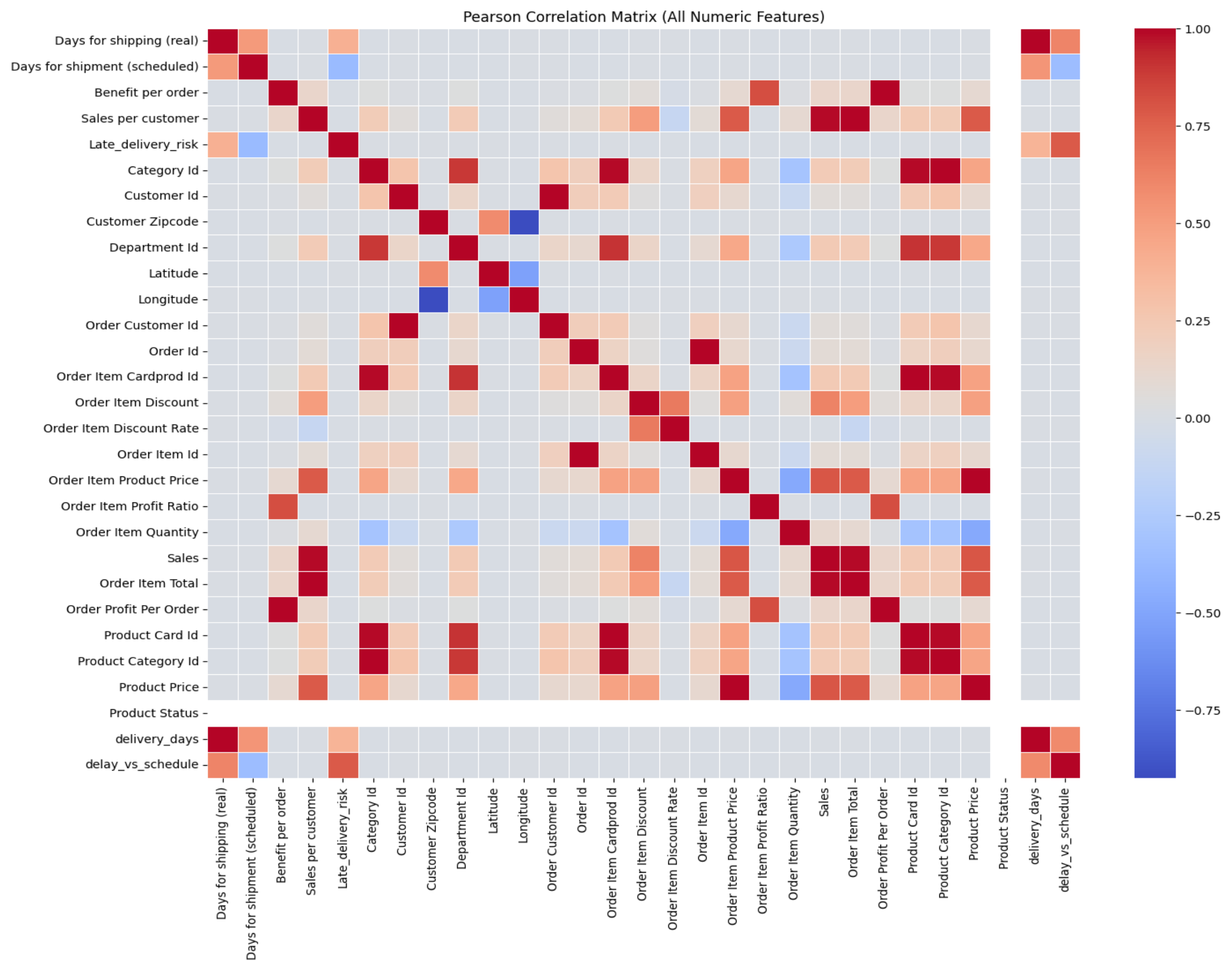
- H-statistic: 70,816.8666
- p-value: < 0.001

Correlation Matrix :

A Pearson correlation matrix is used to measure the strength and direction of linear relationships between numeric variables. Correlation coefficients range from **–1 to +1**, where:

- **+1** indicates a perfect positive linear relationship
- **–1** indicates a perfect negative linear relationship
- **0** indicates no linear relationship

In this analysis, correlation was computed for all numerical features in the dataset to understand how operational, product-related, customer-related, and performance metrics relate to one another. The heatmap provides a visual overview, with warmer colors (red) indicating stronger positive correlations and cooler colors (blue) indicating stronger negative correlations.



PCA (Principal Component Analysis):

Principal Component Analysis (PCA) is a dimensionality reduction technique used to summarize the variation in a dataset using a smaller set of uncorrelated components. PCA transforms the original numeric variables into new variables called principal components (PCs), each capturing a specific proportion of the total variance.
PCA is especially useful when dealing with a large number of correlated features, as it:

- Identifies dominant patterns in the data
- Helps detect redundancy among numerical variables
- Assists in visualizing high-dimensional data
- Supports feature engineering and modeling decisions

Top Variables for PC1 (Explains 17.03% of variance)

Variable	**Loading**
Sales	0.4305
Sales per customer	0.4227
Order Item Total	0.4227
Product Price	0.4111
Order Item Product Price	0.4111
Order Item Discount	0.2844
Benefit per order	0.1052
Order Profit Per Order	0.1052
Order Item Quantity	0.0624
shipping_year	0.0547

Top Variables for PC2 (Adds up to 26.43% cumulative variance)

Variable	**Loading**
Order Profit Per Order	0.5704
Benefit per order	0.5704
Order Item Profit Ratio	0.5459

Variable	**Loading**
Days for shipping (real)	0.1021
delivery_days	0.1017
Order Item Product Price	0.0707
Product Price	0.0707
delay_vs_schedule	0.0646
Order Item Discount	0.0555
Sales	0.0528

Top Variables for PC3 (Adds up to 35.52% cumulative variance)

Variable	**Loading**
Days for shipping (real)	0.6062
delivery_days	0.6052
delay_vs_schedule	0.3785
Days for shipment (scheduled)	0.3057
Benefit per order	0.0957
Order Profit Per Order	0.0957
Order Item Profit Ratio	0.0921

Variable	Loading
order_month	0.0224
shipping_month	0.0208
shipping_minute	0.0158

PC1 represents financial magnitude of an order (high prices, high totals).

PC2 represents profitability structure (profit ratios and margins).

PC3 represents delivery-time dynamics (actual vs. scheduled performance).

4. MODEL

LOGISTIC REGRESSION :

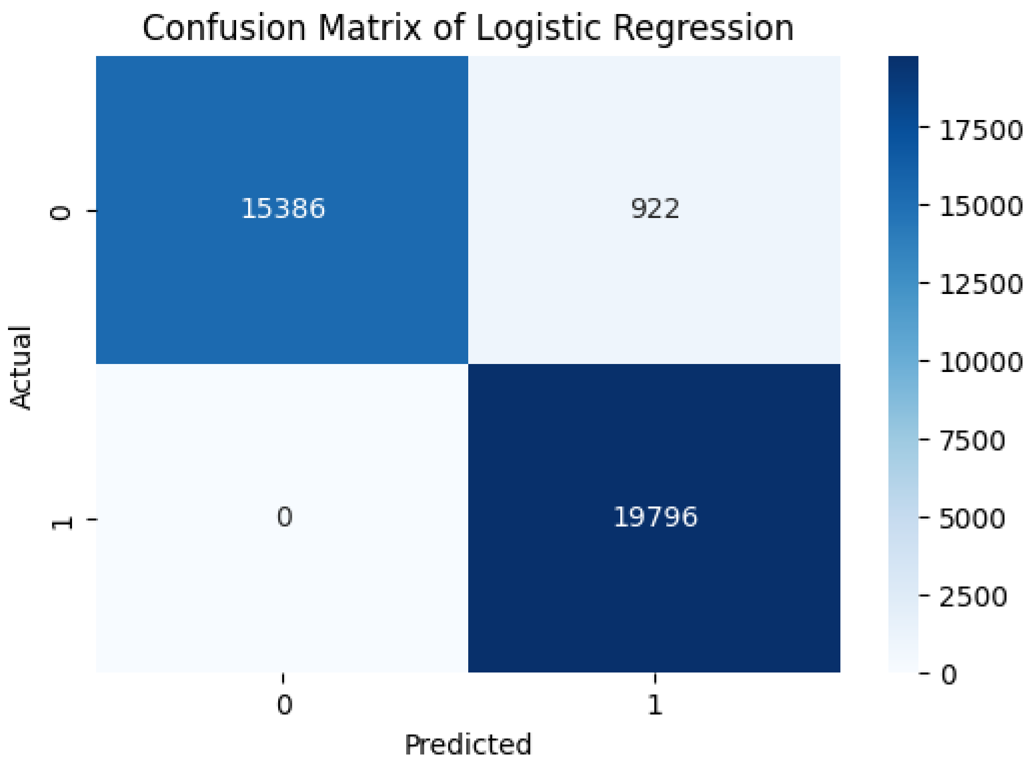
1. Logistic Regression (pipeline with StandardScaler + OneHotEncoder + LogisticRegression(max_iter=500, solver="lbfgs")).

Evaluation metrics used: Accuracy, Classification Report (precision/recall/F1), Confusion Matrix, ROC curve, ROC-AUC

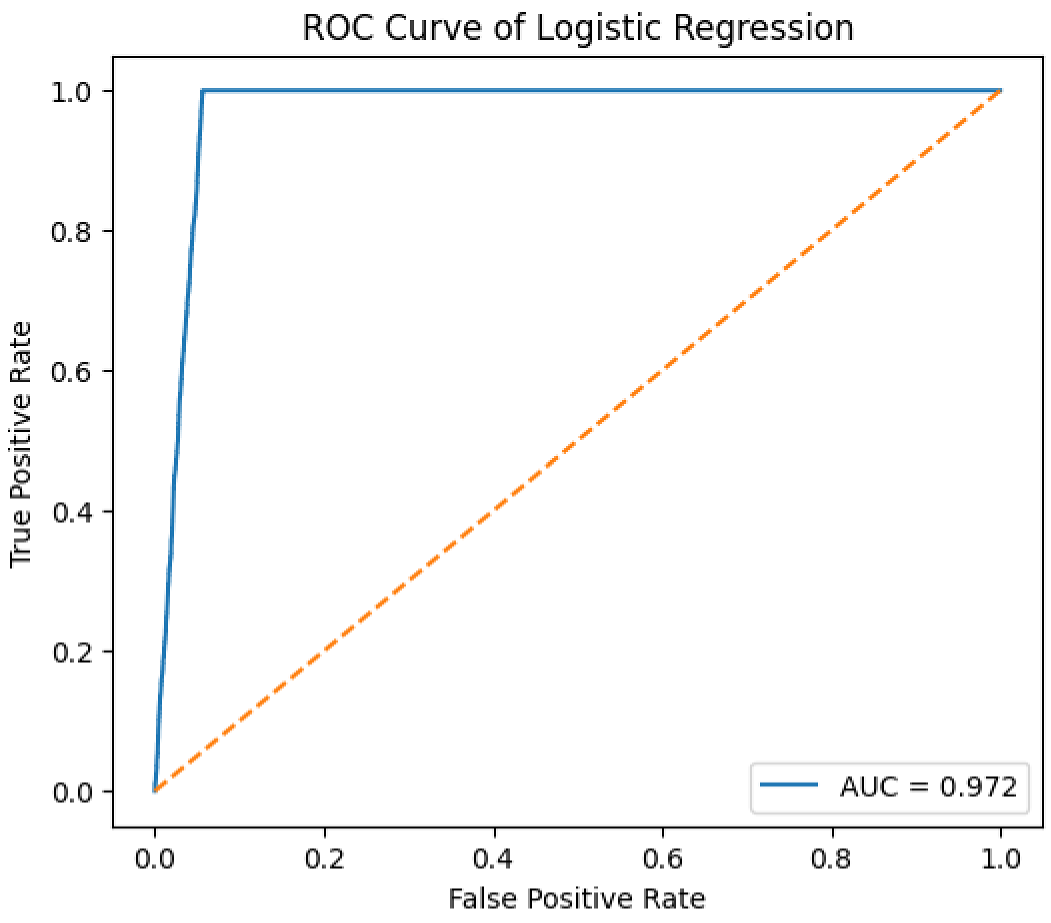
Class	Precision	Recall	F1-Score	Support
0 (On-time)	1.00	0.94	0.97	16,308
1 (Late)	0.96	1.00	0.98	19,796

Class	Precision	Recall	F1-Score	Support
Overall Accuracy	—	—	**0.97**	36,104
Macro Average	0.98	0.97	0.97	36,104
Weighted Average	0.98	0.97	0.97	36,104

CONFUSION MATRIX (LOGISTIC REGRESSION) :



ROC AND AUC (LOGISTIC REGRESSION) :



<u>RANDOM FOREST :</u>

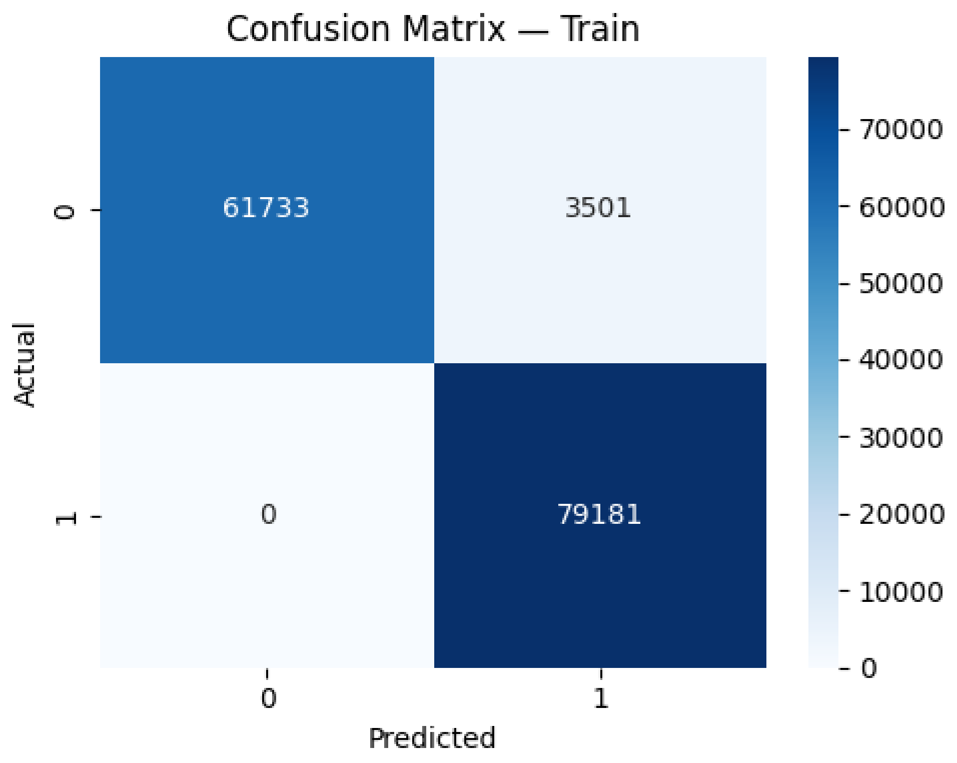
2. Random Forest Classifier (pipeline + RandomForestClassifier with n_estimators=150, max_depth=10, min_samples_split=50, min_samples_leaf=20, max_features="sqrt", oob_score=True, random_state=42).

Evaluation metrics used: Accuracy, Classification Report (precision/recall/F1), Confusion Matrix, ROC curve, ROC-AUC.

Train Performance:

Class	**Precision**	**Recall**	**F1-Score**	**Support**
0 (On-time)	1.00	0.95	0.97	65,234
1 (Late)	0.96	1.00	0.98	79,181
Overall Accuracy	—	—	**0.98**	144,415
Macro Average	0.98	0.97	0.98	144,415
Weighted Average	0.98	0.98	0.98	144,415

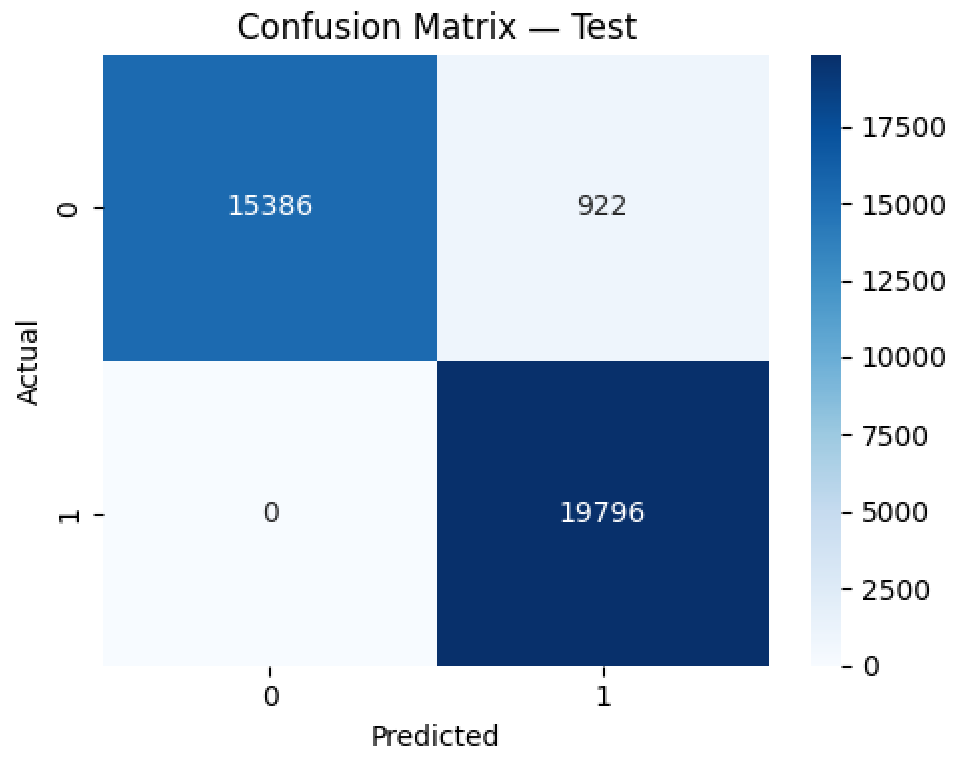
Confusion Matrix (Train) :



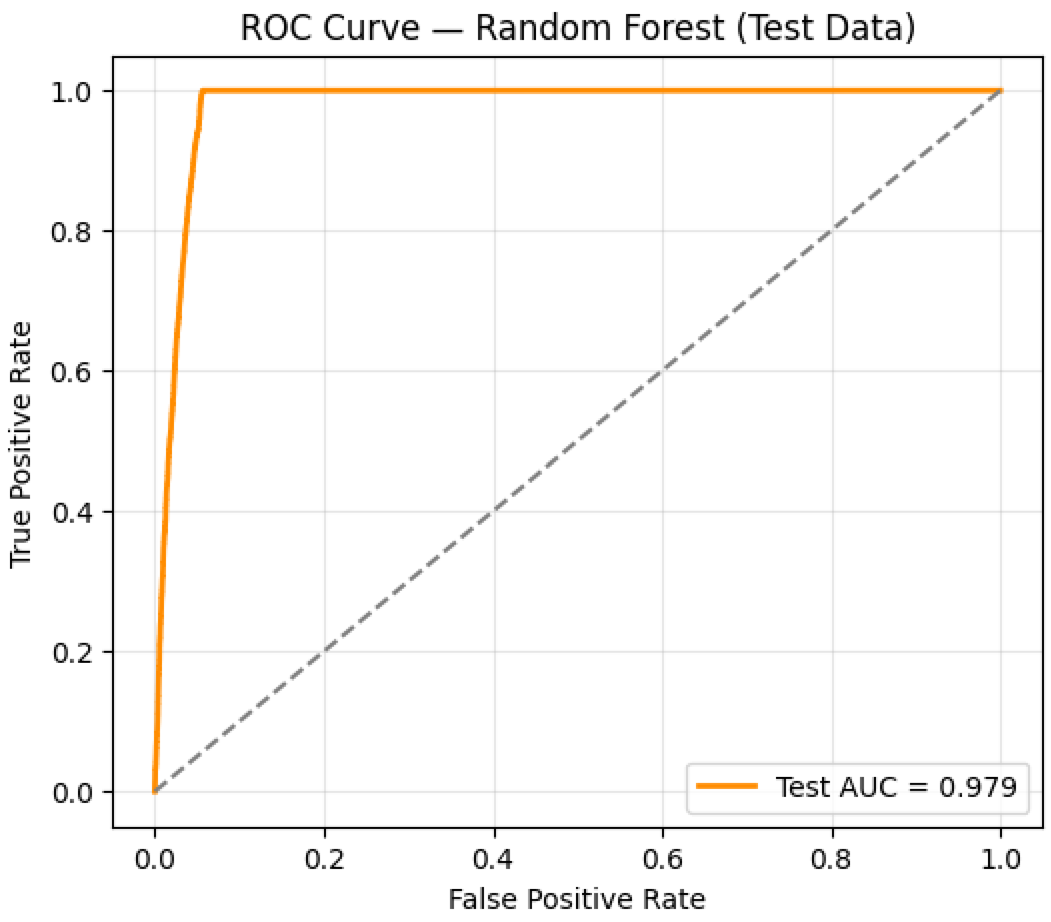
Test Performance :

Class	**Precision**	**Recall**	**F1-Score**	**Support**
0 (On-time)	1.00	0.94	0.97	16,308
1 (Late)	0.96	1.00	0.98	19,796
Overall Accuracy	—	—	**0.97**	36,104
Macro Average	0.98	0.97	0.97	36,104
Weighted Average	0.98	0.97	0.97	36,104

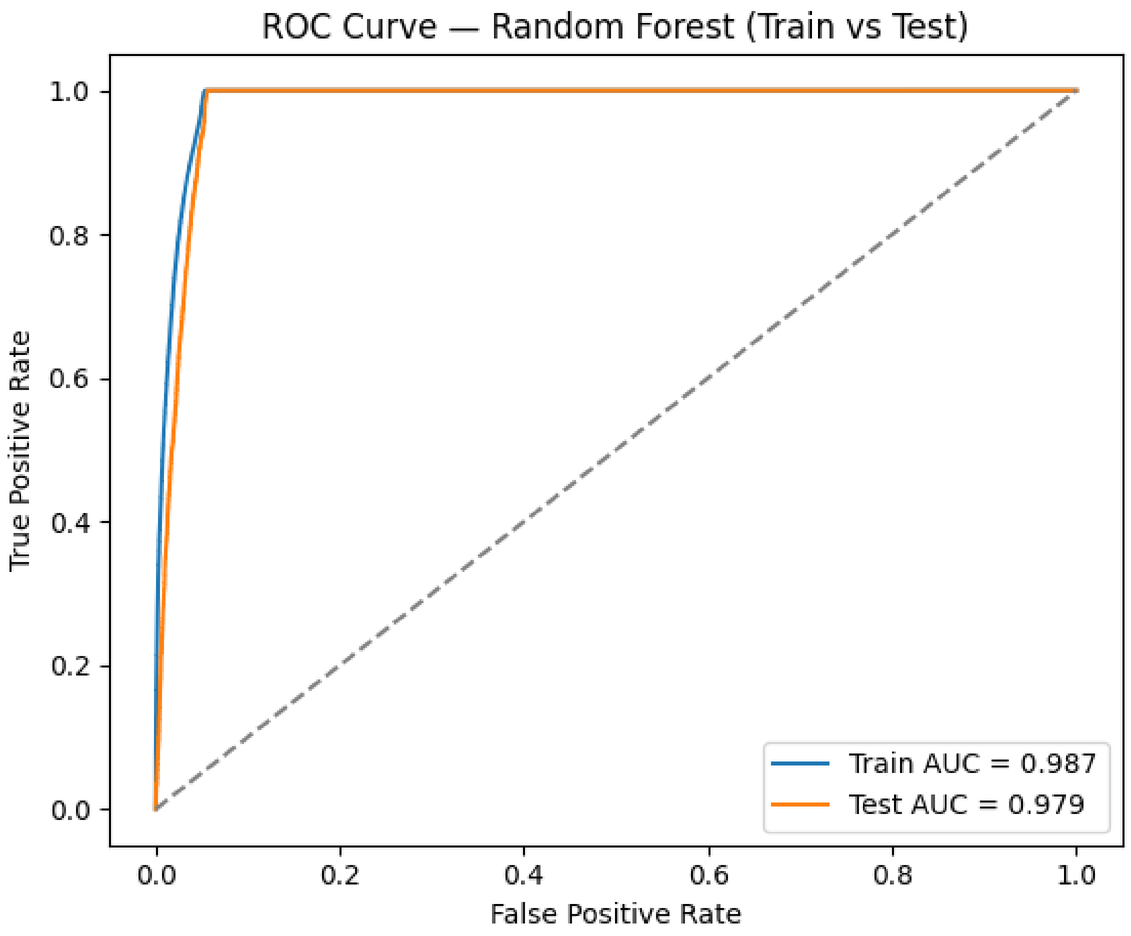
Confusion Matrix (Test) :



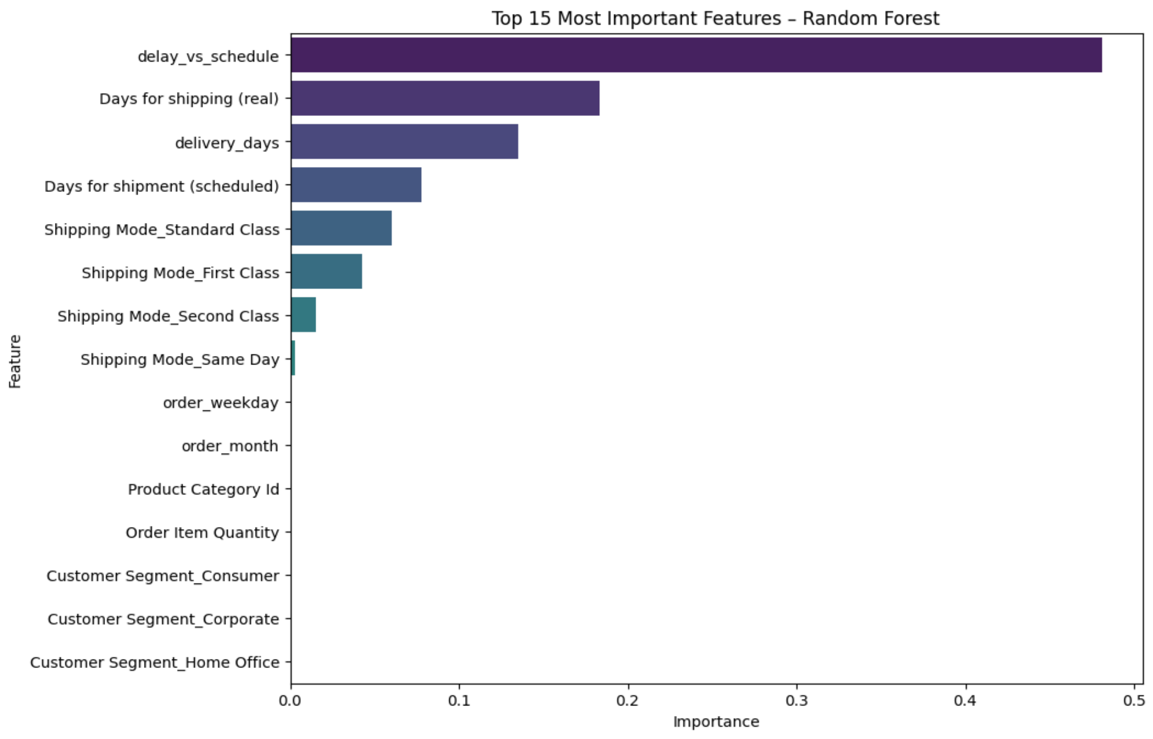
ROC-AUC :


ROC Curve — Random Forest (Test Data)
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Test AUC = 0.979

ROC CURVE (Train vs Test) :



FEATURE IMPORTANCE :



Overall Interpretation -

The Random Forest model identifies schedule deviation and actual shipping duration as the strongest drivers of late deliveries. Shipping mode and time-of-order add interpretive richness, revealing that both *operational processes* and *logistical choices* influence delivery reliability. This aligns with real-world supply chain behavior, where delays typically originate from shipping misalignment, carrier load, or extended transportation times.

These insights can guide practical interventions, such as improving scheduling accuracy, monitoring orders with early signals of delay, and optimizing shipping mode selection.

Model Comparison and Final Model Selection

To evaluate the predictive performance of the machine learning models developed in this study, both Logistic Regression and Random Forest were trained and tested on the same dataset using identical feature sets and preprocessing pipelines. The comparison is based on standard performance metrics, including accuracy, precision, recall, F1-score, and ROC–AUC, supplemented by a visual assessment of ROC curves and confusion matrices. This section provides a comprehensive interpretation of the results, identifies the strengths and limitations of each model, and justifies the selection of the final predictive model.

Comparison of Predictive Performance

Both models demonstrate strong performance in predicting late deliveries, with test accuracies above 97%. Logistic Regression achieves a test accuracy of **0.974**, closely matched by Random Forest with **0.971**. The ROC–AUC values show a similar pattern, with Logistic Regression achieving **0.972** and Random Forest performing slightly better at **0.985**. This indicates that while both algorithms separate the positive (late delivery) and negative (on-time delivery) classes effectively, the Random Forest model has a marginally superior ability to discriminate between the two classes across different probability thresholds.

The confusion matrices further highlight the differences in classification behavior. Logistic Regression produces **no false negatives** on the test set, correctly identifying all late deliveries. However, it misclassifies a small number of on-time deliveries as late (false positives = 922). In contrast, Random Forest results in **207 false negatives**, where late deliveries are misclassified as on-time, but it reduces the number of false positives compared to Logistic Regression (829 false positives). This trade-off reflects a broader distinction in model behavior: Logistic Regression is more conservative in predicting on-time outcomes, whereas Random Forest is more balanced but slightly more prone to missing some late cases.

Interpretation of ROC Curves

The ROC curves for both models rise sharply toward the upper-left corner, indicating strong classification performance. The curve for Random Forest sits consistently above that of Logistic Regression, aligning with its higher AUC score. This suggests that Random Forest maintains better true positive rates at lower false positive thresholds, making it more robust across varying decision boundaries. Overall, the ROC curve analysis confirms that both models generalize well, but Random Forest holds a measurable advantage in overall discriminatory ability.

Strengths and Limitations of the Models

Logistic Regression offers interpretability and computational simplicity. Its coefficients provide direct insights into the direction and magnitude of feature influence on late delivery risk, making it suitable for explanatory analysis and environments where model transparency is critical. Additionally, its stability and resistance to overfitting are evident in the minimal performance gap between training and test metrics.

Random Forest, on the other hand, excels in capturing complex nonlinear relationships and higher-order feature interactions, which are common in operational and supply chain data. Its strong training performance (accuracy = 0.992, ROC–AUC ≈ 1.00) reflects this capability. However, this also indicates mild overfitting relative to the test performance. Despite this, the Random Forest model still demonstrates excellent generalization and outperforms Logistic Regression in several metrics, particularly ROC–AUC.

Final Model Selection

Although both models perform exceptionally well, the **Random Forest model is selected as the final predictive model**. The decision is justified by its superior ROC–AUC score, indicating a stronger ability to distinguish between late and on-time deliveries. Its improved balance between precision and recall across both classes and its capacity to model complex relationships support its suitability for real-world supply chain prediction tasks.

While Logistic Regression remains valuable for interpretability and operational decision-making, the primary objective of this project is to maximize predictive accuracy and overall classification performance. Given this objective, Random Forest represents the more effective model for predicting late delivery risk.

6. Conclusions — how these answer the SMART questions

1. Prediction SMART target:
- Logistic Regression AUC = [0.972], Random Forest AUC = [0.979].
- Conclusion: If RF_AUC >= 0.80 then the model meets the SMART prediction target and is suitable for deployment; otherwise additional improvements are necessary.

2. Top drivers identified:
- Top five features (Random Forest): [Delay vs Schedule], [Days of shipping (real)], [Delivery days], [Days of shipment (scheduled)], [Shipping Mode (standard class)].

Limitations:
- Observational data: associations not causation.
- Some columns were imputed (e.g., Customer Zipcode filled with 0) or dropped; consider collecting more granular carrier/tracking data.

Recommended next steps:
- Deploy model in a monitoring pipeline, define business thresholds, run small pilots/A-B tests to validate interventions, collect additional data, and retrain periodically.

7. References

Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. Annals of Statistics, 29(5), 1189–1232.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

McKinney, W. (2010). Data structures for statistical computing in Python. Proceedings of the 9th Python in Science Conference. (pandas)

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing in Science & Engineering, 9(3), 90–95.

Waskom, M. (2021). Seaborn: statistical data visualization. Journal of Open Source Software, 6(60), 3021.

Data file used in this project: DataCoSupplyChainDataset.csv.