Analysis and Prediction of Delivery Success for Online Furniture Orders

Group 12

Group members

16231 – Hemal Mewantha

16349 – Maheesha Sewmini

16145 – Sanjana Fernando

Table of Contents

List of Figures

List of Tables

Abstract

This study looks at what makes online furniture orders get delivered (or not). Using a 1,938-row Kaggle dataset of orders and delivery attributes, we examine how product type, brand tier, prices and fees, requested assembly, payment method/timing, and promised delivery window affect whether an order is Delivered, Failed, or still On Going.

After cleaning and engineering features (dropping redundant totals, grouping subcategories and brands, imputing missing values, and encoding the target), descriptive analysis revealed a strong class imbalance: failed deliveries are the largest group. Multicollinearity checks led us to remove the highly redundant total_amount. We tested several classifiers (Decision Tree, Random Forest, SVM, XGBoost, AdaBoost) under baseline, SMOTE, and tuned conditions.

AdaBoost was chosen as the final model because it balanced recall on the Delivered class with interpretability. Feature importance analyse point to brand tier, product category, assembly requests, and payment attributes as top predictors.

1. Introduction

With the rise of online furniture retailing, ensuring timely and successful deliveries is increasingly critical. Factors influencing delivery success include product category, brand, pricing, shipping and assembly costs, delivery window, and assembly service requests. By examining historical data, this report aims to identify key drivers of delivery success and provide insights to improve logistics efficiency.

2. Description of the Question

With the growing popularity of online furniture shopping, ensuring successful and timely deliveries has become a critical area of focus for retailers and logistics providers. Various factors, such as product category, brand, price, shipping cost, requested assembly service, and delivery window, play a significant role in determining whether an order is delivered successfully and on time.

By analyzing these factors, we can gain valuable insights into how different order attributes contribute to delivery success. This study aims to develop a predictive model that can assess and forecast delivery outcomes based on product, order, and logistics characteristics. By leveraging data-driven techniques, the model will analyze historical order data to identify patterns and relationships between order attributes and delivery performance.

The goal is to create a reliable system that can provide early risk flags for at-risk deliveries, helping businesses take proactive action (e.g., route optimization, carrier prioritization, better inventory allocation) and ultimately improve customer satisfaction and reduce operational costs.

Thus, the key objectives of this project are to:

1. Analyze how various product and logistics-related attributes influence delivery success rates.

2. Develop a predictive model to estimate the probability of successful delivery based on order features such as product category, shipping cost, delivery window, and assembly service request.

3. Description of the Dataset

The dataset, sourced from Kaggle ('Online Furniture Orders – Delivery and Assembly 2025'), contains 1,938 observations and 14 variables. Six are categorical while eight are numerical.. The target variable is delivery_status

The table below provides a detailed description of each variable in the dataset.

Variable	Description	Type
Product_category	High-level category (Living Room, Bedroom, Dining Room, Office, Kitchen, Outdoor).	Categorical
Product_subcategory	Specific item type (e.g., Sofa, Bed Frame, Desk, Patio Set).	Categorical
brand	Furniture brand (e.g., IKEA, West Elm, Pottery Barn).	Categorical
delivery_status	Fulfillment status (Pending, In Transit, Delivered, Failed Delivery, Rescheduled, Cancelled).	Categorical

assembly_service_requested	True/False flag indicating if assembly was requested.	Boolean
payment_method	Payment mode (Credit/Debit Card, PayPal, Apple/Google Pay, Bank Transfer, COD).	Categorical
order_id	Unique order identifier per transaction.	Numerical
customer_id	Unique customer identifier for cohort and LTV analysis.	Numerical
product_price	Base product price in USD, ranges tuned by category.	Numerical
shipping_cost	Shipping fee; free or reduced for higher price points.	Numerical
assembly_cost	Assembly fee driven by complexity tiers; zero when not requested.	Numerical
total_amount	All-in order value: product_price + shipping_cost + assembly_cost.	Numerical
delivery_window_days	Promised delivery window in days (1–14).	Numerical
customer_rating	Post-delivery rating (1.0–5.0) with natural missingness.	Numerical

Table 1 Description of the Dataset

4. Pre-Processing

- Convert the assembly_service_requested variable from boolean to object data type.
- Drop order id and customer id variables
- Split data into two sets as training (80%) and testing (20%). Descriptive analysis was conducted using the training set.
- The variables brand, shipping_cost, assembly_cost, and customer_rating contain missing values. Imputed them appropriately.

5. Feature Engineering

- Simplified the product_subcategory variable by mapping detailed categories into broader groups such as Seating, Table, Storage, Sleeping, Patio furniture, and Work.
- Simplified the brand variable by grouping brands into broader tiers as Premium, Mid Range, Affordable, and Other (for unknown/unclassified brands).
- Standardized the payment_method variable by grouping similar options into Card, Digital Wallet, Bank Transfer, and Cash.
- Created a new feature payment_timing to indicate when the payment occurred (*Prepaid* for online methods and *Postpaid* for cash payments).
- Encoded the target variable (delivery_status) by grouping similar statuses into broader categories (Delivered, Failed Delivery, and On Going).

6. Important Results of Descriptive Analysis

6.1 Target Variable-Delivery Status

The chart shows the distribution of the target variable (Delivery Status). Most orders fall under Failed Delivery (767), followed by On Going (508), while Delivered orders (275) make up the smallest share. This indicates an imbalanced distribution, with a higher proportion of failed deliveries compared to successful ones.

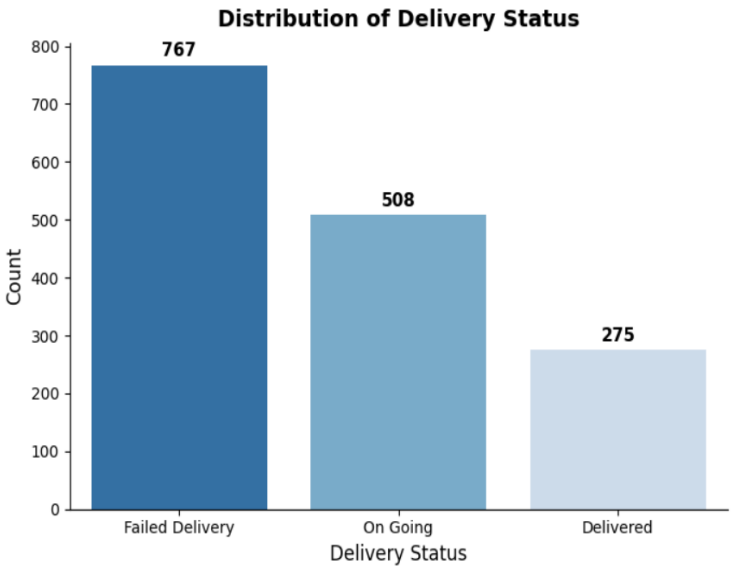

Figure 1 Bar chart of Delivery Status

6.2 Multicollinearity Assessment Using Correlation and VIF Analysis

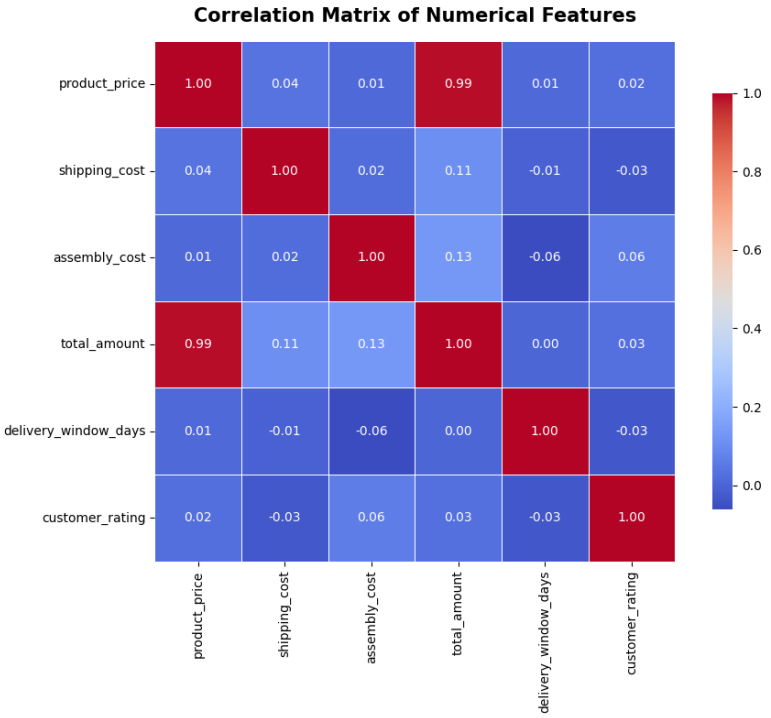

Figure 2 correlation Matrix

Feature	VIF
total_amount	6247.620856
product_price	5258.092782
assembly_cost	32.182423
shipping_cost	24.361044
customer_rating	5.143943
delivery_window_days	3.632479

Table 2 VIF values

The correlation matrix of numerical features shows a very high positive correlation (0.99) between total_amount and product_price, indicating redundancy between these two variables.

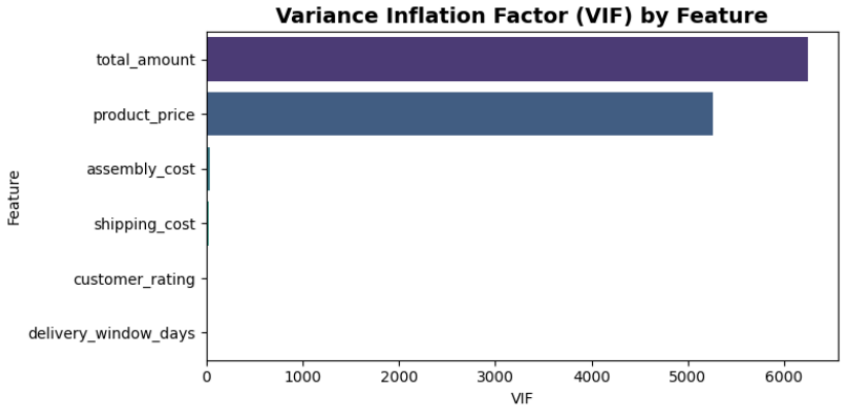

Figure 3 VIF by Feature

Other features exhibit low correlations, suggesting limited linear relationships among them. The Variance Inflation Factor (VIF) table confirms this: total_amount (VIF ≈ 6247) and product_price (VIF ≈ 5258) have extremely high multicollinearity, while assembly_cost, shipping_cost, customer_rating, and delivery_window_days have much lower VIF values, indicating acceptable levels of multicollinearity. These results suggest that total_amount and product_price cannot be used together in regression models without causing instability.

So the total_amount variable was dropped.

Table 3 contain the VIF values after dropped the total_amount variable. Now features are within acceptable limits.

Feature	VIF
customer_rating	5.141784
delivery_window_days	3.632404
product_price	3.464187
shipping_cost	2.857258
assembly_cost	1.521409

Table 3 VIF values after dropped total_amount

7. Fisher's Discriminant Analysis (FDA) Decision Boundary

By using Figure 3, we can say that although LDA is a supervised technique designed to maximize class separability, the significant overlap among the three color-coded delivery statuses (Failed Delivery, On Going, and Delivered) suggests that these categories share many common feature characteristics. This overlap indicates that the model struggles to distinguish between the classes using linear boundaries, implying that the factors influencing delivery outcomes may not be strongly separable within the chosen feature space.

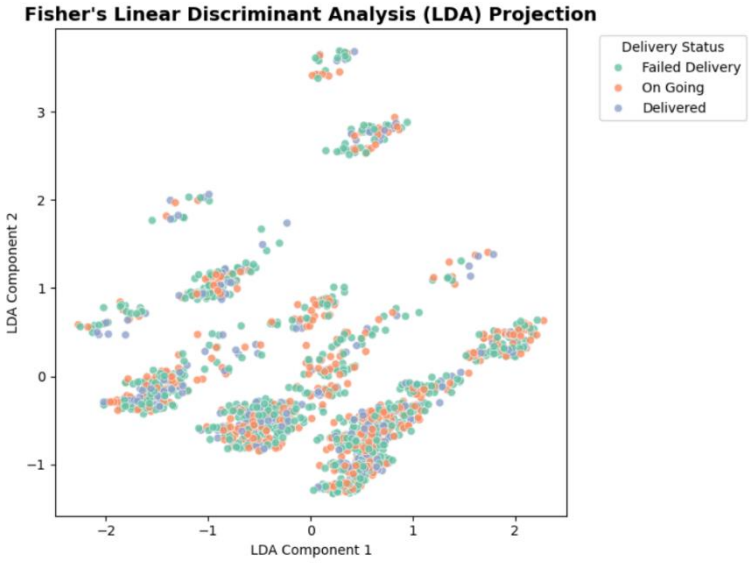

Figure 4 LDA projection plot

8. Cluster Analysis

We conducted K-Prototypes clustering (K=3) to validate our dataset's structure before modeling, achieving a silhouette score of 0.543 that indicates moderate cluster quality. Cross-tabulation analysis revealed strong alignment between unsupervised clusters and supervised class labels (>85% purity per cluster), with no evidence of heterogeneous subgroups within classes. The FAMD dimensionality reduction plot (Figure 5) visually confirms this separation, showing three

distinct groupings across the first two components (explaining 6.47% and 5.98% of variance). While some overlap exists between groups—consistent with the moderate silhouette score—the clear spatial separation validates that our features can meaningfully distinguish between the three target classes and supports the use of standard supervised classification methods rather than cluster-based modeling approaches.

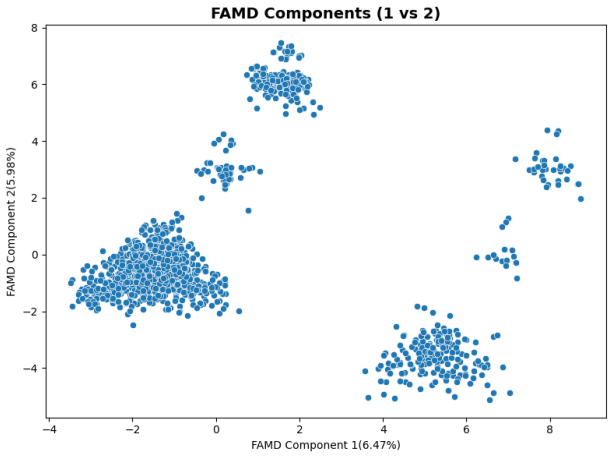

Figure 5 FAMD dimensionality reduction plot

9. Important Results of Advanced Analysis

9.1 Best Model

The model performance was evaluated under four different conditions to assess the impact of data balancing and hyperparameter tuning. Four separate tables were created to present the accuracy, precision, and F1-score for both the training and testing datasets. The table 4 represents the baseline performance without SMOTE, while the table 5 shows results without SMOTE but with hyperparameter tuning to optimize model parameters. The table 6 presents results with SMOTE applied, addressing class imbalance, and the table 7 shows performance with both SMOTE and hyperparameter tuning. This comparison provides a clear understanding of how balancing the dataset and tuning parameters affect the model's accuracy and overall classification performance.

Model	Training Set			Testing Set		
	Accuracy	Precision	F1-score	Accuracy	Precision	F1-score
AdaBoost	0.59	0.58	0.56	0.41	0.37	0.38
DecisionTree	0.59	0.58	0.57	0.41	0.38	0.39
Random Forest	0.59	0.58	0.57	0.42	0.38	0.39
SVM	0.53	0.51	0.39	0.49	0.35	0.34
XGBoost	0.59	0.58	0.57	0.42	0.39	0.40

Table 4 Baseline performance without SMOTE

Model	Training Set			Testing Set		
	Accuracy	Precision	F1-score	Accuracy	Precision	F1-score
AdaBoost	0.49	0.24	0.32	0.49	0.24	0.32
DecisionTree	0.58	0.56	0.54	0.43	0.40	0.40
Random Forest	0.55	0.54	0.47	0.47	0.40	0.38
SVM	0.49	0.24	0.32	0.49	0.24	0.32
XGBoost	0.51	0.47	0.36	0.48	0.30	0.32

Table 5 Performance without SMOTE with hyperparameter tuning

Model	Training Set			Testing Set		
	Accuracy	Precision	F1-score	Accuracy	Precision	F1-score
AdaBoost	0.58	0.57	0.57	0.40	0.38	0.38
DecisionTree	0.58	0.57	0.57	0.39	0.39	0.38
Random Forest	0.58	0.57	0.57	0.41	0.39	0.40
SVM	0.62	0.64	0.62	0.35	0.37	0.35
XGBoost	0.50	0.51	0.50	0.37	0.37	0.37

Table 6 Baseline performance with SMOTE

Model	Training Set			Testing Set		
	Accuracy	Precision	F1-score	Accuracy	Precision	F1-score
AdaBoost	0.42	0.42	0.41	0.39	0.39	0.38
DecisionTree	0.51	0.51	0.51	0.40	0.40	0.40
Random Forest	0.58	0.57	0.57	0.41	0.40	0.40
SVM	0.99	0.98	0.98	0.35	0.36	0.36
XGBoost	0.48	0.47	0.44	0.40	0.38	0.39

Table 7 Performance with both SMOTE and hyperparameter tuning

9.2 Final Model

Although several models demonstrated competitive results, AdaBoost stands out as the most suitable choice for this dataset due to its strong generalization ability and adaptive learning mechanism. Despite slightly lower overall accuracy compared to Random Forest, AdaBoost consistently achieved comparable F1-scores while maintaining better recall for the "Delivered" class—an important factor when minimizing false negatives in critical delivery predictions. Its iterative boosting process effectively combines multiple weak learners, enabling it to capture complex relationships within the data and perform well even with limited parameter tuning. Moreover, AdaBoost offers a more interpretable framework for understanding feature contributions, making it advantageous for explaining classification outcomes in practical applications. Considering its balanced performance, adaptability, and interpretability, AdaBoost is selected as the final model for this study.

Model	Training Set			Testing Set		
	Accuracy	Precision	F1-score	Accuracy	Precision	F1-score
AdaBoost	0.42	0.42	0.41	0.39	0.39	0.38

Table 8 Final Model

9.3 Feature Importance Plot

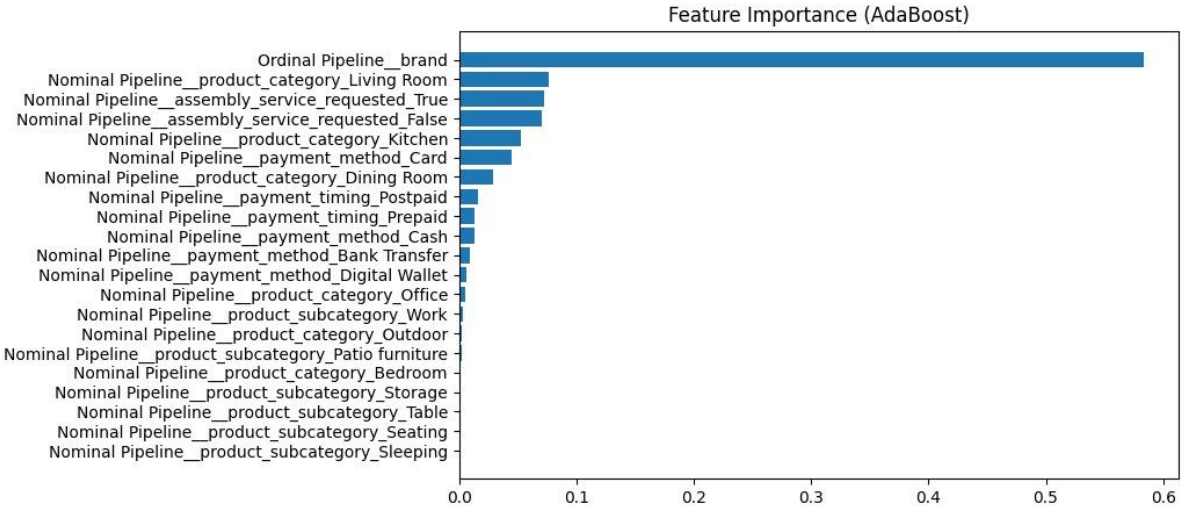

Figure 6 Feature Importance plot

This bar chart shows the feature importance from an AdaBoost model. It highlights which factors have the greatest impact on the model's predictions. The most important feature by far is "Ordinal Pipeline__brand", which contributes significantly more than any other feature. Other features like product category, assembly service request, and payment method have smaller but still notable influences, while the rest contribute minimally to the model's performance.

10. Discussion and Conclusions

This project transformed complex delivery data into actionable insights, revealing that a few key operational factors (particularly brand tier, product category, assembly requests, and payment type) drive much of the variation in delivery outcomes. Despite high class imbalance and dataset limitations such as missing values and multicollinearity, the AdaBoost model emerged as the most balanced performer, offering useful recall on Delivered orders and interpretability for operational use. The findings suggest that while data quality and missing process variables limit overall accuracy, the model still provides valuable early warnings that can guide proactive delivery management.

In practice, the model can serve as an early risk-scoring tool to flag orders likely to fail or delay, helping operations prioritize premium-brand or assembly-required deliveries for closer monitoring. To enhance performance, future efforts should focus on enriching data with route, carrier, and timing details, applying cost-sensitive or threshold-tuned learning to better align with business goals, and maintaining continuous retraining as delivery patterns evolve.

11. Appendix including python code and technical details

https://drive.google.com/drive/folders/1JigCaUz2QcQJYIYZ50__3E3zLhl-9NSt?usp=sharing