© July 2025 | IJIRT | Volume 12 Issue 2 | ISSN: 2349-6002

Enhancing Supply Chain Decision-Making and Strategic Planning using Machine Learning and Neural Networks

Anubhav Kumar Tiwary[1], Prof. Vishnupant Potdar[2], and Prof. Shubhangi P. Tidake[3]
[1]M. Sc. (Data Science) Student, Symbiosis Skills & Professional University, Pune
[2,3]Assistant Professor, Symbiosis Skills & Professional University, Pune

Abstract—Accurately estimating delivery times is a key challenge in optimizing last-mile logistics within the broader scope of supply chain management. This study presents a structured machine learning pipeline to predict delivery time using a real-world dataset comprising order details, geolocation, weather, traffic, and time-based factors. Various regression models were evaluated, including Linear Regression, Decision Tree with Bagging, Random Forest, ElasticNet, SVM, XGBoost, CatBoost, LightGBM (with Optuna tuning), and Neural Networks. Model performance was assessed via 5-fold cross-validation using metrics such as MSE, RMSE, MAE, and R² Score. CatBoost emerged as the top-performing model, achieving the highest R² score of 0.8666 and the lowest error rates. While ensemble models dominated in performance, Neural Networks were also explored for their ability to learn complex feature patterns. The results demonstrate the practical value of advanced machine learning in handling high-dimensional, noisy delivery data, offering actionable insights for improving delivery time predictions and enhancing overall supply chain efficiency.

Index Terms—CatBoost, Cross-Validation, Delivery Time Prediction, Feature Engineering, Machine Learning, Neural Network, Regression Models, Supply Chain Management.

I. INTRODUCTION

In the era of rapid urbanization and booming e-commerce, timely and efficient delivery of goods has become a critical component for ensuring customer satisfaction and maintaining competitive advantage. Last-mile delivery, which refers to the final step of the logistics chain from the distribution hub to the customer's location, often involves unpredictable delays due to various real-world factors including weather, traffic, time of day, and geographical constraints. Traditional delivery systems largely rely on rule-based estimations or simplistic examinations that fail to capture the complexity and variability of real-world scenarios. In contrast, modern data-driven approaches, especially machine learning, offer significant potential to improve the accuracy of delivery time predictions by leveraging rich contextual features from historical data.

This study presents a machine learning pipeline to predict delivery times using real-world data, including order details, geolocation, weather, traffic, and time-based variables. Several regression algorithms, such as CatBoost, LightGBM, XGBoost, Random Forest, and Neural Networks, were compared using 5-fold cross-validation and key metrics like R^2, MAE, MSE, and RMSE. Beyond improving accuracy, the goal was to uncover key factors influencing delivery delays. The findings emphasize the value of predictive modeling in enhancing last-mile logistics, contributing to smarter and more efficient supply chain management.

II. LITERATURE REVIEW

Recent advancements highlight the growing application of machine learning in solving diverse supply chain challenges. Camur et al. (2023) focused on predicting product availability during disruptions, while Pietukhov et al. (2023) proposed hybrid models combining logistic regression and neural networks to improve KPI forecasting. Similarly, Rezki et al. (2024) used machine learning to proactively manage supplier delay risks. Across these works, ensemble-based techniques like Random Forest, Gradient Boosting, and Neural Networks consistently demonstrated high performance in complex prediction tasks. Badrinarayanan (2024) further applied AI in optimizing last-mile delivery using predictive analytics and route planning strategies.

Among relevant studies, Garg et al. (2025) applied machine learning to predict food delivery times in

Indian cities using contextual variables such as traffic, weather, and geospatial features. However, their model comparison was limited to a few traditional algorithms. This highlights a gap in exhaustive benchmarking of diverse regression techniques on real-world delivery datasets. Addressing this, the present study builds a robust pipeline integrating data preprocessing, feature engineering, and evaluation of nine algorithms. Importantly, it outperforms Garg et al. (2025) in R^2 score and Mean Square Error (MSE) using a CatBoost model, indicating a clear improvement in prediction accuracy and reliability. This positions the work as a practical contribution for improving delivery efficiency in real-world Indian logistics scenarios.

III. DATASET

A. About the Dataset

This study utilizes a real-world dataset of over 43,000 Amazon last-mile deliveries across Indian cities, sourced from Kaggle. It includes order and pickup timestamps, agent details (age, rating), geolocation data, product category, and contextual variables such as weather, traffic, vehicle type, and area type (urban, metropolitan). The target variable is Delivery Time, representing the duration from pickup to drop. This rich feature set supports effective machine learning modelling for identifying key factors that influence delivery times and enhancing decision-making in supply chain management.

B. Data Preprocessing

To prepare the dataset for machine learning modeling, several preprocessing steps were performed to ensure data quality, consistency, and suitability for regression analysis.

- Missing Values and Duplicates Handling: Missing values were identified in the *Agent_Rating* and *Weather* columns and handled by removing the affected rows to preserve data quality. Additionally, the dataset was checked for duplicate entries, and none were found, so no further action was needed.
- Data Type Standardization: All categorical columns including Order_ID, Weather, Traffic, Vehicle, Area, and Category were converted to string type for uniformity. Numeric columns such as Agent_Age, Delivery_Time, and coordinate columns were cast to their appropriate types (int or float) for processing consistency.
- Feature Extraction and Engineering: Date and time-related columns (Order_Date, Order_Time, Pickup_Time) were converted to datetime formats. Time-based features such as Order_Hour, Pickup_Hour, and their total minutes representations were extracted. Order Processing Time was computed by subtracting total order minutes from pickup minutes. A binary Is_Weekend column was also created using Order_Date to indicate weekend orders.
- Outlier Removal: Outliers in the Delivery_Time column were handled using the Interquartile Range (IQR) method. Additionally, unrealistic latitude values (<1) were removed based on domain knowledge (valid Indian coordinates). Geographic distances between pickup and drop locations were calculated using the Haversine formula. Deliveries with distances >50 km or processing times >120 minutes were excluded as extreme outliers.

After preprocessing, the final dataset consisted of 39,128 records with 24 well-defined features, ready for model training.

C. Exploratory Data Analysis (EDA)

Exploratory Data Analysis offered insights essential to predictive modeling. Several key findings appeared through our analysis:

- Influence of Agent Age on Delivery Time: Agents in the youngest age group, between 21 to 25 years, recorded the shortest average delivery times, around 110 minutes. In contrast, agents aged 36 to 40 took approximately 140 minutes on average. The data shows delivery time increases with age, possibly due to reduced agility or slower mobility in old age (see Fig.1).

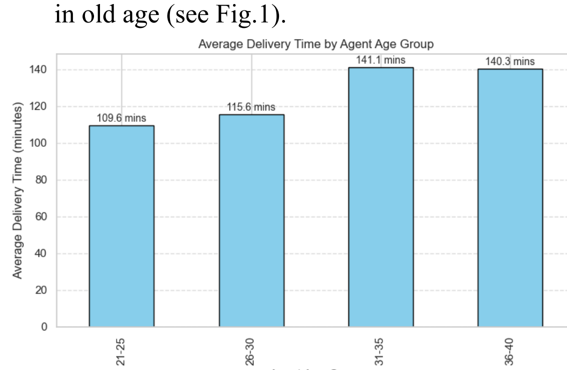

Fig.1 Average Delivery Time by Agent Age Group

- Distribution of Orders across Agent Age Groups: The distribution of delivery assignments across agent age groups reveals a balanced workload among younger agents. Age groups 21 to 25, 26 to 30, and 31 to 35 each handled approximately 26 percent of total orders. In contrast, the age group 36 to 40 managed only 21 percent of deliveries, suggesting either fewer assignments or lower operational involvement. This demographic insight may guide workforce allocation strategies for improved efficiency (see Fig.2).

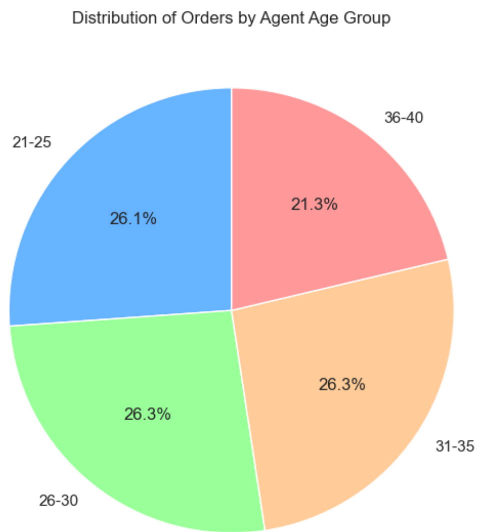

Fig.2 Distribution of Orders by Agent Age Group

- Relationship between Agent Ratings and Delivery Performance: A moderate negative correlation was observed between agent ratings and delivery time. Higher-rated agents tended to complete deliveries more quickly, reflecting the operational efficiency linked with better performance scores (see Fig.3).

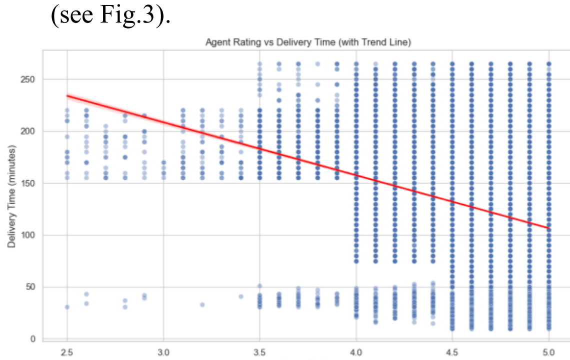

Fig.3 Agent Rating vs Delivery Time (with Trend Line)

- Variation in Delivery Time across Area Types: Delivery was faster in urban and other areas, averaging 105–110 minutes, while semi-urban areas had the longest average of about 235 minutes. This is likely due to variations in infrastructure and access (see Fig.4).

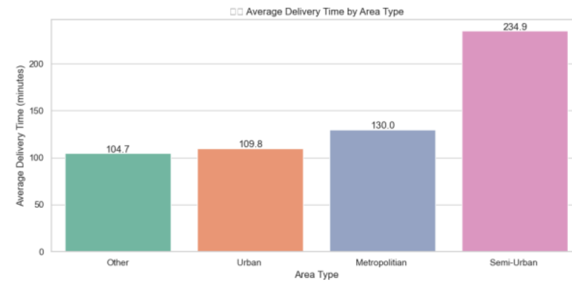

Fig.4 Average Delivery Time by Area Type

- Impact of Time of Day on Delivery Duration: Morning deliveries were the fastest, averaging around 100 minutes. Evening deliveries were slowest, taking about 138 minutes, likely due to traffic congestion. Afternoon and night slots showed moderate delivery durations (see Fig.5).

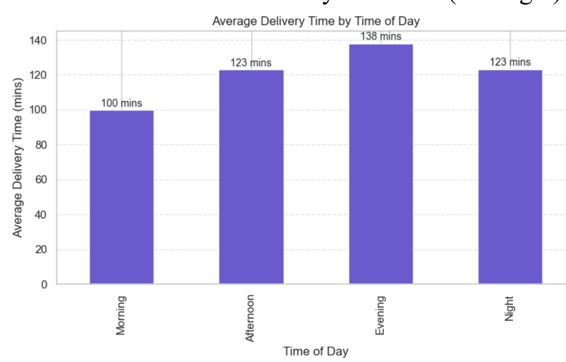

Fig.5 Average Delivery Time by Time of Day

- Comparison between Weekday and Weekend Deliveries: No significant difference in delivery times was observed, indicating consistent efficiency throughout the week (see Fig.6).

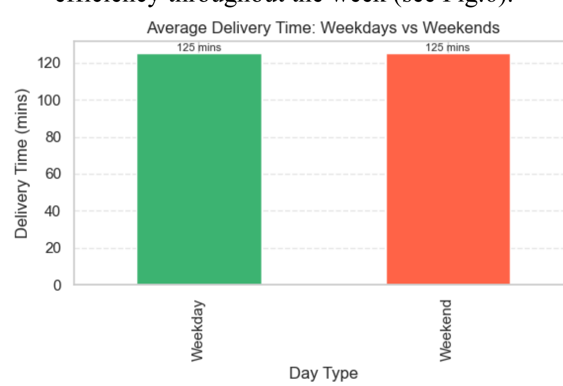

Fig.6 Average Delivery Time: Weekdays vs Weekends

- Effect of Weather Conditions on Delivery Time: Deliveries made under sunny weather were the quickest, with average times around 103 minutes. Cloudy and foggy conditions led to the longest delays, averaging over 137 minutes. These findings highlight how adverse weather can disrupt delivery timelines (see Fig.7).

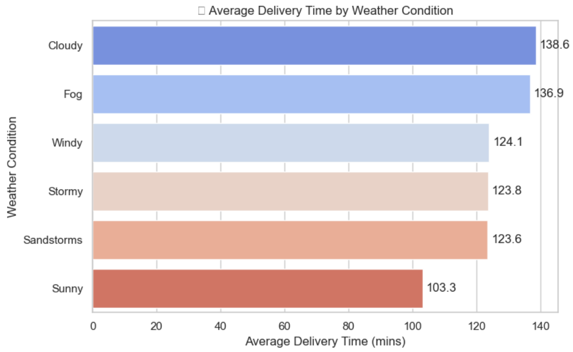

Fig.7 Average Delivery Time by Weather Condition

A heatmap was used to find features most correlated with delivery time, aiding in feature selection (see Fig. 8).

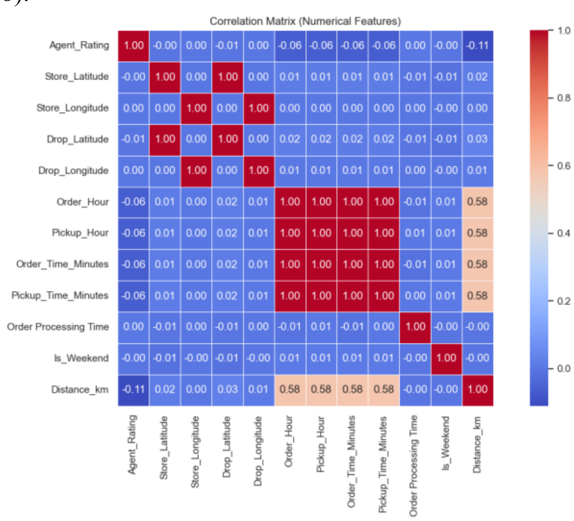

Fig.8 Correlation Matrix (Numerical Features)

IV. METHODOLOGY

A. *Overview*

To accurately forecast delivery times, we designed a structured machine learning pipeline involving thorough data preprocessing, feature selection, model training, and evaluation. The approach integrates dynamic contextual variables such as weather, traffic, and time-related features, which reflect real-time delivery conditions in urban Indian environments. This enhances the model's ability to capture practical delays and improve prediction reliability.

B. *Feature Selection*

To enable effective feature selection after EDA, additional preprocessing refined the dataset. Categorical columns like weather, traffic, vehicle type, area, and product category were converted using one-hot encoding. To handle multicollinearity, highly correlated features (correlation > 0.85) were identified via a correlation matrix and removed to enhance model stability. Numerical variables were scaled using MinMaxScaler to a uniform 0–1 range, improving convergence and ensuring fair algorithm treatment. Finally, Recursive Feature Elimination (RFE) with a linear regression estimator identified the most impactful variables for predicting delivery time. The top features selected for final model training included:

- Agent_Age
- Agent_Rating
- Order_Hour
- Distance_km
- Weather_Sandstorms
- Weather_Stormy
- Weather_Sunny
- Weather_Windy
- Traffic_Jam
- Traffic_Low
- Traffic_Medium
- Vehicle_scooter
- Vehicle_van
- Area_Other
- Area_Semi-Urban
- Area_Urban
- Category_Grocery
- Category_Toys

C. *Model Building*

To predict delivery time, nine diverse regression models were trained on the processed dataset, including traditional, ensemble, and deep learning approaches. The models were selected for their varying strengths and capabilities:

- Linear Regression: Used as a baseline for its simplicity, assuming a linear relationship between features and delivery time.
- Decision Tree with Bagging: Reduces overfitting by averaging predictions from multiple trees trained on random subsets, capturing non-linear patterns.
- Random Forest: An ensemble method that averages multiple trees, robust to overfitting, and handles categorical/numerical features well.

- XGBoost: A gradient boosting algorithm that sequentially builds trees to correct prior errors, with strong performance on tabular data.
- LightGBM: A fast, memory-efficient gradient booster tuned with Optuna for better R^2 scores, optimized for large datasets.
- CatBoost: CatBoost performs exceptionally well with categorical data while reducing overfitting and requiring minimal hyperparameter tuning, achieving the highest accuracy in our implementation.
- Elastic Net: A linear model combining L1 (lasso) and L2 (ridge) regularization, useful for multicollinearity and feature selection.
- Support Vector Machine (SVM): Captures complex patterns by mapping inputs to higher dimensions using kernels.
- Neural Network (MLP): A Multi-Layer Perceptron (MLP) neural network was used to learn complex non-linear patterns in the data. It is capable of automatically capturing feature interactions and delivering smooth, continuous predictions

All models were trained after log-transforming the target (delivery time) to stabilize variance and enhance performance.

V. EXPERIMENTAL RESULTS AND ANALYSIS

A. *Evaluation Metrics*

To evaluate the performance of all models, 5-fold cross-validation was used to ensure the results were stable and not biased due to data splits. In 5-fold cross-validation, the dataset is split into 5 parts: each time, 4 parts are used for training and 1 for testing, and this process is repeated five times. The average of all five results gives a reliable estimate of model performance. The following metrics were used to evaluate the models:

- Mean Absolute Error (MAE): It computes the average of absolute differences between estimated and observed values. Lower MAE indicates better accuracy.
- Mean Squared Error (MSE): This metric computes the average of squared differences between estimated and observed values. It penalizes larger errors more heavily.
- Root Mean Squared Error (RMSE): By taking the square root of MSE, RMSE expresses prediction errors in their original scale, making them more meaningful for real-world applications.
- R^2 Score (Coefficient of Determination): Indicates how well the model explains the variance in the target variable. Ranges from 0 to 1 in which higher values are better.

B. *Model Performance Comparison*

CatBoost attained the highest R^2 score of 0.8666, followed closely by LightGBM and XGBoost. These models demonstrated superior prediction accuracy and generalization performance, making them most suitable for this delivery time prediction task (see Table 1). A visual comparison of R^2 scores for all models can be seen in Fig.9.

Table 1 Performance comparison of various models.

Model	MSE	RMSE	MAE	R^2 Score
CatBoost	0.04	0.19	0.15	0.8666
LightGBM (Tuned)	0.04	0.19	0.15	0.8631
XGBoost	0.04	0.19	0.15	0.8627
Random Forest	0.04	0.20	0.16	0.8518
Decision Tree (Bagging)	0.04	0.21	0.16	0.8417
SVM	0.05	0.22	0.17	0.8274
Linear Regression	0.07	0.26	0.21	0.7559
Neural Network	0.09	0.30	0.24	0.6681

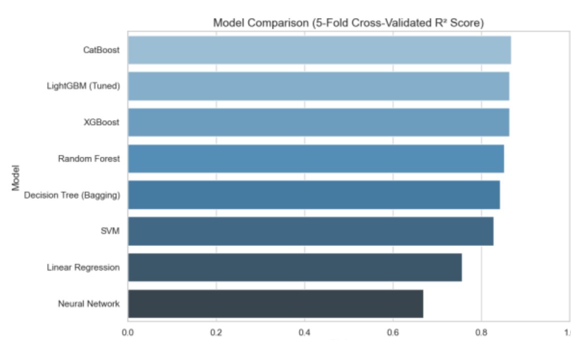

Fig.9 Visualization (5-Fold Cross-Validated R^2 Score)

C. *Overfitting Check*

To ensure the selected model is not overfitting, we compared performance on both training and testing datasets using the CatBoost model. Key evaluation metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R^2 Score were calculated separately for both sets. On the training set, the model attained an MAE of 16.34, RMSE of 21.19, and R^2 score of 0.8312. On the test set, the values were MAE of 17.41, RMSE of 22.63, and R^2 score of 0.8129, indicating minimal difference. The absence of substantial overfitting was observed, as the model maintained consistent performance on both training and validation datasets.

VI. FUTURE WORK

While the current study successfully demonstrated the effectiveness of various models, especially CatBoost, for delivery time prediction, future work can explore several promising directions. Incorporating real-time dynamic features such as live GPS tracking, real-time weather feeds, and traffic congestion APIs could further improve model accuracy and responsiveness. Furthermore, the learnt model can become adaptive to changing logistical patterns when it is used in a production-grade setting with continuous learning capabilities. Exploring advanced deep learning techniques like LSTM or Transformer architectures may also improve sequential understanding in time-based data. Collectively, these enhancements can drive more intelligent and adaptive solutions for real-time decision-making in supply chain management, especially in the context of last-mile logistics.

VII. CONCLUSION

This study presented a comprehensive and practical approach to predicting last-mile delivery time using real-world data. By incorporating contextual variables such as agent demographics, weather, traffic, and geolocation features, we developed an end-to-end machine learning pipeline tailored for Indian urban logistics. The models were rigorously evaluated using 5-fold cross-validation, where CatBoost emerged as the best performer with an R^2 score of 0.8666, indicating high predictive accuracy and generalization. Overfitting checks further validated the model's stability on unseen data. Importantly, the approach strengthens the predictive capabilities within supply chain management, specifically in the domain of last-mile delivery optimization. By combining advanced preprocessing, feature engineering, and robust algorithmic benchmarking, this work contributes meaningfully toward more efficient, reliable, and customer-centric supply chain systems in the delivery sector.

REFERENCES

[1] M. C. Camur, S. K. Ravi, and S. Saleh. Enhancing supply chain resilience: A machine learning approach for predicting product availability dates under disruption. Expert Systems with Applications, 247:123226, 2024.

[2] R. Pietukhov, M. Ahtamad, M. Faraji-Niri, and T. El-Said. A hybrid forecasting model with logistic regression and neural networks for improving key performance indicators in supply chains. Supply Chain Analytics, 4:100041, 2023.

[3] N. Rezki and M. Mansouri. Machine learning for proactive supply chain risk management: Predicting delays and enhancing operational efficiency. Management Systems in Production Engineering, 32(2):345-356, 2024.

[4] A. Badrinarayanan. AI-driven optimization of last-mile delivery. Massachusetts Institute of Technology, 2025.

[5] A. Garg, M. Ayaan, S. Parekh, and V. Udandarao. Food delivery time prediction in Indian cities using machine learning models. arXiv, 2025.

[6] E. Yalcinkaya and O. A. Hiziroglu. A comparative analysis of machine learning models for time prediction in food delivery operations. Advances in Artificial Intelligence and Applications, 2(1):34-45, 2024.

[7] H. Sahin and D. Icen. Application of random forest algorithm for the prediction of online food delivery service delay. Turkish Journal of Forecasting, 5(1):1-11, 2021.

[8] Sujal Suthar, Amazon Delivery Dataset, 2024. Link: https://www.kaggle.com/datasets/sujalsuthar/amazon-delivery-dataset

[9] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. An Introduction to Statistical Learning: with Applications in R. Springer, 2021.