Impact Factor (SJIF): 6.126 E-ISSN: 2583-1615

International Journal of Research in Academic World

**Received**: 05/February/2025 **IJRAW:** 2025; 4(SP3):24-26 **Accepted:** 19/March/2025

# Predicting Delivery Delays in Logistics Using Machine Learning

**[*1]Dr. Krishnaveni Sakkarapani and [2]Sudharsana S**

[*1]Assistant Professor, Department of Data Analytics (PG), PSGR Krishnammal College for Women, Coimbatore, Tamil Nadu, India.

[2]PG Student, Department of Data Analytics (PG), PSGR Krishnammal College for Women, Coimbatore, Tamil Nadu, India.

**Abstract**

Using machine learning to predict logistics delays entails creating models that examine past data in order to predict possible delays in delivery procedures. The goal of this project is to improve supply chain management's operational efficiency by employing machine learning techniques to predict logistics delays. The machine learning model created for this project analyzes a large dataset with historical shipment data using sophisticated techniques like Random Forest and Decision Tree Classifier. To improve prediction accuracy, important factors affecting delays—such as supplier details, order volumes, and delivery methods—are carefully investigated. Stakeholders may readily obtain real-time predictions and explanations by deploying the model through an intuitive graphical user interface (GUI).

**Keywords:** Machine learning, Predicting, Random forest, Decision tree classifier, User friendly GUI, Logistics delay.

## 1. Introduction

In order to guarantee the efficient flow of goods from suppliers to customers, logistics is an essential part of supply chain management. Delays in logistics operations, however, can lead to inefficiencies, higher expenses, and unhappy customers. Businesses can take proactive steps to improve supply chain operations by anticipating these delays in advance. Through the analysis of historical data and the identification of important trends that contribute to delays, machine learning (ML) has become a potent technique to improve logistics delay prediction.

The Decision Tree algorithm, one of several machine learning algorithms, maps out multiple delay-causing factors in a hierarchical fashion to produce an interpretable model. An ensemble of several decision trees called the Random Forest model improves generalization and decreases overfitting to increase prediction accuracy. On the other hand, logistic regression can be used to categorize a shipment's likelihood of being delayed based on affecting factors.

Businesses may keep an eye on shipments and take remedial action before delays happen by integrating these models in real-time with logistics management systems and utilizing interactive visualization tools such as Streamlit.

## 2. Model Description

### i). Decision Tree Classifier

The decision tree operates in a tree-like structure in which each node denotes a choice made in response to a feature (for example, "Is distance > 500 km?"). The data is divided into branches until a final choice (such as "Delayed" or "On Time") is made. The model applies the rules it has learned from the dataset to fresh shipments.

**Advantages:** Easy to Interpret: It offers a clear decision path that is easy to comprehend. Handles Non-Linearity: It performs admirably even in cases when there is a complicated relationship between features and delays. No Scaling Requires: There is no need for preprocessing while handling numerical and categorical data. Quick Training: Uses small to medium-sized datasets effectively.

### ii). Random Forest

Multiple Decision Trees are combined in the ensemble learning technique known as Random Forest. A random subset of data is used to train each tree, and either majority voting (classification) or averaging (regression) is used to get the final prediction. It increases prediction accuracy and decreases overfitting.

**Advantages:** More Accurate than a Single Decision Tree-By averaging several forecasts, it lowers errors.

### iii). Logistic Regression

A mathematical technique called logistic regression forecasts the likelihood of a shipment delay. It converts output values into probabilities between 0 and 1 using a sigmoid function. It predicts an on-time delivery if the likelihood is below a threshold (e.g., 0.5); if it is over that threshold, it forecasts a delay.

**Advantages:** Efficient and straightforward, it performs best in issues where the relationships between variables are obvious. Gives probability scores, which aid in determining the degree

***Corresponding Author:** Dr. Krishnaveni Sakkarapani

of prediction confidence. Less prone to overfitting—performs better than sophisticated models on fewer datasets. Interpretable Model: Indicates how each feature affects the probability of a delay.

## 3. Methodology

i). **Data Collection:** 20,000 entries pertaining to logistics shipments make up this dataset, which includes information on cargo origins, destinations, schedules, vehicle types, distances, weather, traffic jams, and delays. It has eleven columns: Vehicle Type, Distance, Weather, Traffic, Origin, Destination, Shipment Date, Planned Delivery Date, Actual Delivery Date, and Delayed. Tracking a shipment's path from dispatch to delivery, each row represents a distinct shipment.

ii). **Data Preprocessing:** The Missing values are addressed by dropping rows with NaNs or filling missing entries with mode values for weather and traffic conditions. Date columns are converted from strings to datetime objects, enabling derivation of features such as delivery days and planned days that measure time differences. Categorical variables including origin, destination, vehicle type, weather, and traffic conditions are encoded with LabelEncoder. The code prints dataset shape, information, and missing counts to facilitate exploratory data analysis. Finally, the data is split into training and testing sets for modeling.

iii). **Exploratory Data Analysis:** Exploratory Data Analysis (EDA) is a crucial step in data science that involves examining and visualizing datasets to uncover patterns, detect anomalies, and gain insights before applying machine learning models. In the provided code, EDA is performed on shipment data, which includes attributes such as Shipment ID, Origin, Destination, Shipment_Date,Planned_Delivery_Date,Actual_Delivery _Date, Vehicle_Type, Distance, Weather_Conditions, Traffic_Conditions, and Delayed. The EDA process starts with loading and cleaning the dataset by handling missing values, such as filling null values in Weather_Conditions and Traffic_Conditions with their mode.

iv). **Model Training:** Model training training multiple machine learning models—Random Forest, Logistic Regression, and Decision Tree—to predict logistics delays. The dataset is first preprocessed using one-hot encoding to handle categorical features such as 'Origin,' 'Destination,"Vehicle_Type,','Weather_Conditions,' and 'Traffic_Conditions.' After encoding, the dataset is split into training and testing sets using an 80-20 split.
The DelayPredictor class initializes three models and iteratively trains them using the train_and_evaluate method. Each model is fitted with the training data, makes predictions on the test set, and evaluates accuracy using accuracy_score. The model with the highest accuracy is selected as the best model. The Logistic Regression model achieved the highest accuracy of 91.15%, outperforming Random Forest and Decision Tree. Once training is complete, predictions can be made using the predict method, which utilizes the best-performing model to classify new data points.

v). **Model Evaluation:** Accuracy is the main metric used to evaluate the model. Three machine learning models are implemented by the DelayPredictor class: Random Forest, Logistic Regression, and Decision Tree. Each model is trained using training data by the train_and_evaluate function, which then assesses how well it performs on the test set. Accuracy_score (y_test, predictions), which quantifies the percentage of correctly identified cases, is used to compute accuracy.

vi). **Deployment in GUI:** The shipment delay prediction model is deployed using Streamlit. The trained model, feature scaler, and selected features are saved as a .pkl file using joblib. In the Streamlit app, the model and scaler preprocess user input, including shipment details like origin, destination, vehicle type, weather, traffic, and distance. The model predicts and displays "Delayed" or "Not Delayed" with indicators. Deployment can be local (streamlit run model6.py) or cloud-based using Streamlit Community Cloud, Heroku, or AWS. For Heroku, a requirements.txt and Procfile are needed, while Streamlit Cloud requires linking a GitHub repository.

## Conclusion

This project develops a logistics delay prediction system using Decision Tree, Random Forest, and Logistic Regression to analyze factors affecting delivery delays. Decision Tree and Random Forest provided higher accuracy and interpretability, making them ideal for deployment. Logistic Regression performed well but had lower predictive accuracy. The trained model was deployed using Streamlit, enabling real-time delay predictions via a user-friendly interface. This project highlights how machine learning enhances logistics by predicting delays proactively. Future improvements include integrating real-time traffic and weather data to optimize accuracy for dynamic logistics environments.

## References


1. Tochkov K. “The efficiency of postal services in the age of market lib eralization and the internet: Evidence from central and eastern Europe,” Utilities Policy. 2015; 36:35–42.
2. Morganti E, Seidel S, Blanquart C, Dablanc L and Lenz B. “The impact of e-commerce on final deliveries: alternative parcel delivery services in france and Germany,” Transportation Research Procedia. 2014; 4:178–190.
3. Validi A, Polasek N, Alabi L, Leitner M and Olaverri-Monreal C. “Environmental impact of bundling transport deliveries using sumo: Analysis of a cooperative approach in austria,” in 2020 15th Iberian Conference on Information Systems and Technologies (CISTI). IEEE, 2020, 1–5.
4. Gonc J¸ Alves, Goncalves JS, Rossetti RJ and Olaverri-Monreal C. “Smartphone sensor platform to study traffic conditions and assess driv ing performance,” in 17th International IEEE Conference on Intelligent Transportation Systems (ITSC). IEEE, 2014, 2596–2601.
5. Kopica F, Morales W and Olaverri-Monreal C. “Automated delivery of shipments in urban areas,” in 6th International Physical Internet Conference, Church House, Westminster, London, United Kingdom., 2019.
6. Friedman J, Hastie T and Tibshirani R. The elements of statistical learning. Springer series in statistics New York, 2001, 1(10).
7. Nielsen D. “Tree boosting with xgboost-why does xgboost win” every” machine learning competition?” Master’s thesis, NTNU, 2016.
8. Taieb SB and Hyndman RJ. “A gradient boosting approach to the kaggle load forecasting competition,”

*International journal of forecast ing*. 2014; 30(2):382–394.
9. Schapire RE. "The strength of weak learnability," Machine learning. 1990; 5(2):197–227.
10. Drucker H. "Improving regressors using boosting techniques," in ICML. 1997; 97:107–115.
11. Abou Omar KB. "Xgboost and lgbm for porto seguros kaggle challenge: A comparison," Preprint Semester Project, 2018.
12. V Brandst¨ atter and Olaverri-Monreal C. "Efficient transport logistics: An approach for urban freight transport in austria," in 2020 15th Iberian Conference on Information Systems and Technologies (CISTI). IEEE, 2020, 1–6.
13. Mendes-Moreira J, Jorge AM, JF de Sousa, and Soares C, "Improv ing the accuracy of long-term travel time prediction using heterogeneous ensembles," Neurocomputing. 2015; 150:428–439.
14. Hassan SM, Moreira-Matias L, Khiari J and Cats O. "Feature selection issues in long-term travel time prediction," in International Symposium on Intelligent Data Analysis. Springer, 2016, 98–109.
15. Khiari J, Moreira-Matias L, Shaker A, Zenko B and Zeroski SD, "Metabags: Bagged meta-decision trees for regression," in Joint eu ropean conference on machine learning and knowledge discovery in databases. Springer, 2018, 637–652.
16. Hoch T. "An ensemble learning approach for the kaggle taxi travel time prediction challenge." in DC@ PKDD/ECML.