Inspiring Excellence

Department of Computer Science and Engineering

CSE422: Artificial Intelligence
Lab Project Report

Title: Prediction of E-commerce Shipment Delivery Using Machine Learning

Submitted By
Md. Abrar Ahsan Purno - 24141162
Khandoker Md. Ragib Ahsan - 22301202
Section: 06

Submission Date:
3 January 2026

1. Introduction

This project aims to analyze and predict whether an e-commerce shipment will reach the customer on time using machine learning techniques. Timely delivery is a critical factor in customer satisfaction and overall business performance in the e-commerce sector. Delayed shipments can negatively impact customer trust, increase operational costs and reduce the reliability of logistics systems.

The objective of this project is to build and compare multiple machine learning models that can predict delivery outcomes based on shipment, product, and customer-related features. By analyzing historical delivery data, the project seeks to identify important patterns that influence delivery performance. Both supervised and unsupervised learning approaches are applied to gain deeper insights into the dataset.

This project follows a systematic machine learning workflow, including exploratory data analysis, data preprocessing, model training, evaluation and comparison. The final goal is to determine which model performs best for predicting on-time delivery while ensuring robustness and interpretability of results.

2. Dataset Description

2.1 Dataset Overview

The dataset used in this project contains 10,999 data points, where each data point represents an individual e-commerce shipment. The dataset consists of 12 features, including 11 input features and 1 output (target) feature.

2.2 Problem Type: Classification or Regression

This is a classification problem.

The target variable, Reached.on.Time_Y.N, has binary discrete values (0 or 1) where:

- 1 indicates the shipment reached on time
- 0 indicates the shipment did not reach on time

Since the output represents categorical class labels rather than continuous values, classification algorithms are appropriate for this task.

2.3 Number of Features

- Total features: 12
- Input features: 11
- Target feature: 1 (Reached.on.Time_Y.N)

2.4 Feature Types

The dataset contains both quantitative (numerical) and categorical features.

- Categorical features:
    - Warehouse_block
    - Mode_of_Shipment
    - Product_importance
    - Gender
- Numerical features:
    - ID
    - Customer_care_calls
    - Customer_rating
    - Cost_of_the_Product
    - Prior_purchases
    - Discount_offered
    - Weight_in_gms

2.5 Encoding of Categorical Variables

Yes, encoding of categorical variables is required.
Machine learning algorithms operate on numerical data and cannot directly process categorical text values. Therefore, categorical features were converted into numerical format using One-Hot Encoding, which avoids introducing any false ordinal relationships between categories.

2.6 Correlation Analysis

A correlation heatmap was generated using the seaborn library to analyze the linear relationships between numerical input features and the target variable.
The correlation values ranged between –1 and +1, where:

- Positive values indicate positive correlation
- Negative values indicate negative correlation
- Values close to zero indicate weak or no linear relationship

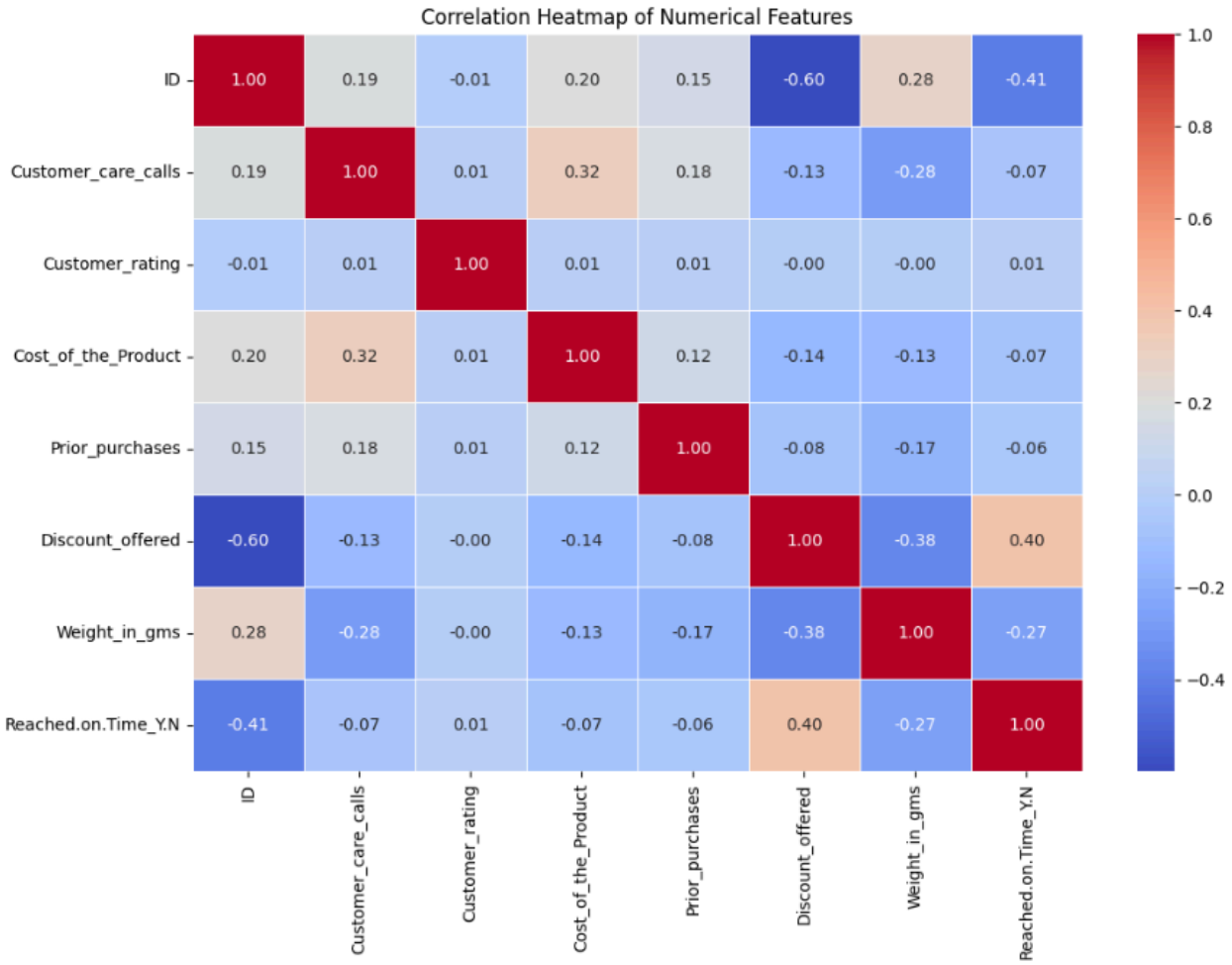


Figure 1: Correlation heatmap showing relationships between numerical features and the target variable

From the correlation analysis, it was observed that:

- No single feature has a very strong correlation with the target variable
- Some features, such as Discount_offered and Weight_in_gms, show moderate correlation with delivery outcome
- Most features exhibit weak linear relationships

This indicates that delivery performance depends on a combination of multiple features, justifying the use of machine learning models capable of learning complex and non-linear relationships.

3. Imbalanced Dataset Analysis

3.1 Class Distribution

The target variable of this dataset is Reached.on.Time_Y.N, which represents whether a shipment reached the customer on time or not. To analyze whether the dataset is balanced, the frequency of each class was examined.

A bar chart was used to visualize the distribution of the two classes:

- Class 1: Shipment reached on time
- Class 0: Shipment did not reach on time

From Figure 2, it is evident that the two classes do not contain an equal number of instances. The number of shipments that reached on time is higher than those that did not.

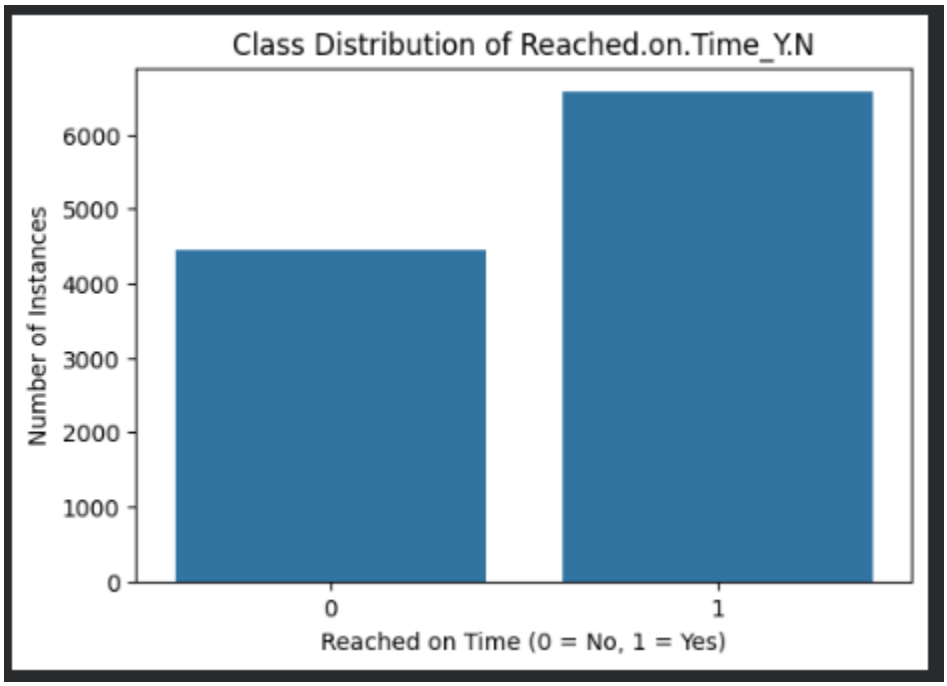


Figure 2: Class distribution of the target variable (Reached.on.Time_Y.N)

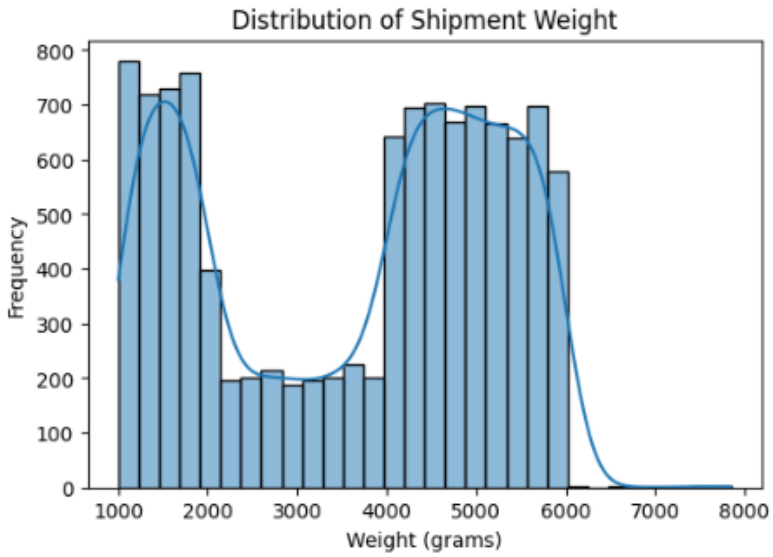
The unequal distribution of class labels indicates that the dataset is imbalanced. In such cases, relying solely on accuracy as an evaluation metric can be misleading, as a model may perform well by predicting only the majority class.
To address this issue, additional evaluation metrics such as precision, recall, F1-score, and ROC-AUC were used later in the project to obtain a more reliable assessment of model performance.

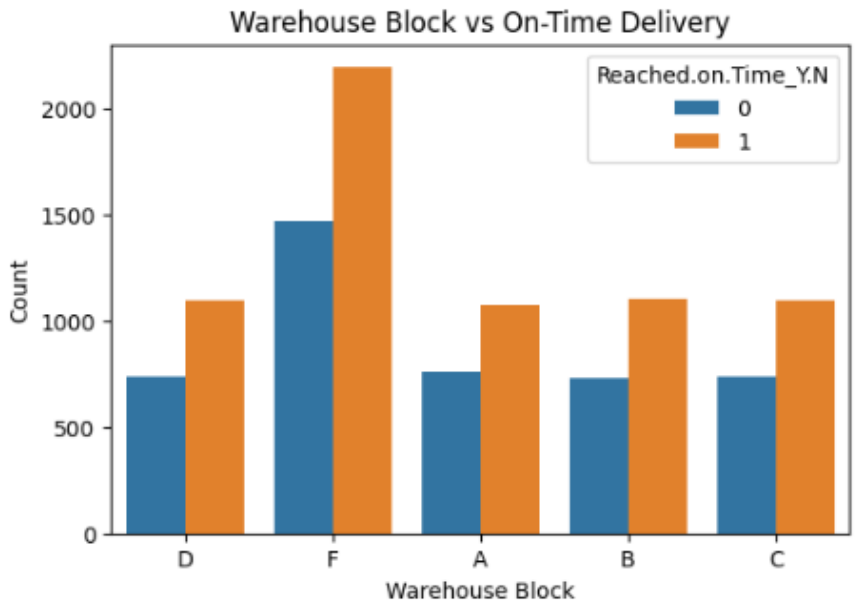
4. Exploratory Data Analysis (EDA)

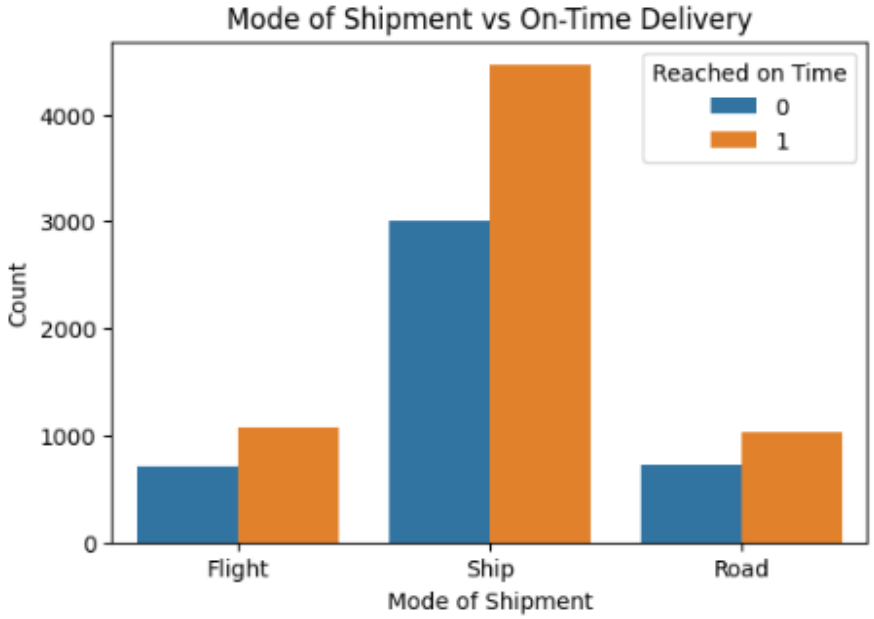
Exploratory Data Analysis (EDA) was performed to understand the underlying patterns, distributions and relationships within the dataset. This helps in gaining insights into feature behavior and identifying potential issues before model training.

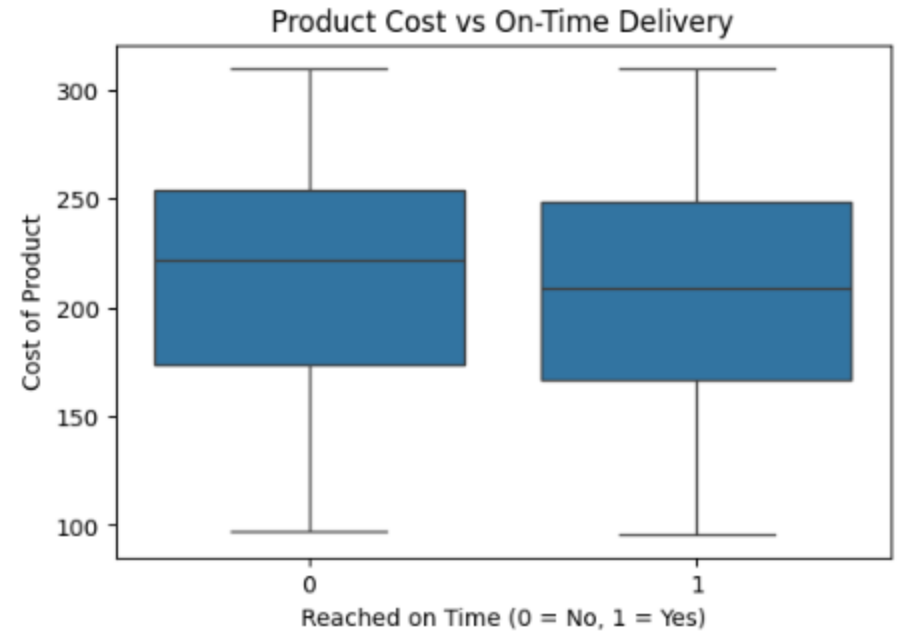
4.1 Distribution Analysis

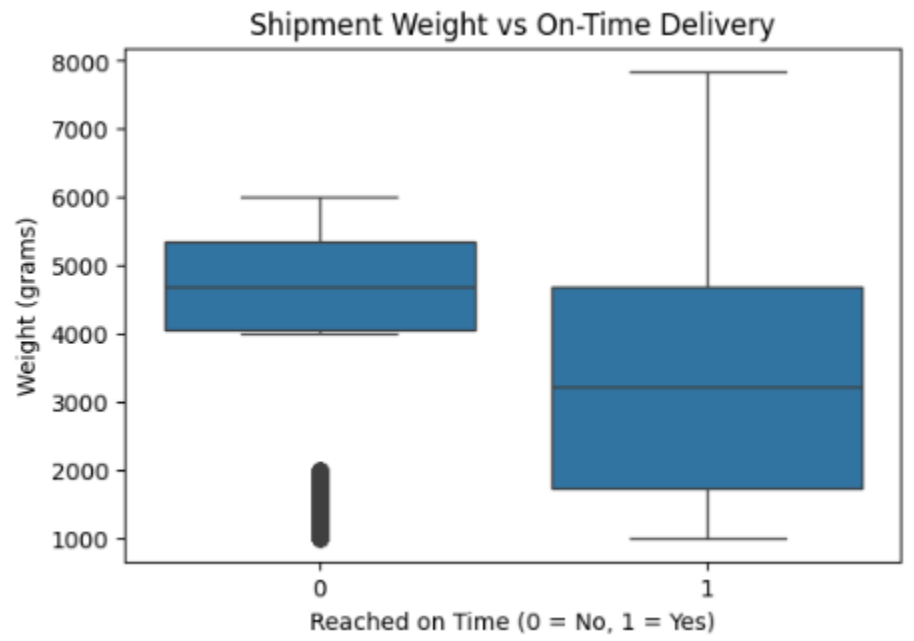
Boxplots were used to analyze the distribution of numerical features. The boxplots revealed the presence of outliers and varying spreads across different features. This information was important in deciding appropriate preprocessing techniques such as feature scaling.




Warehouse Block vs On-Time Delivery
Reached.on.Time_Y.N
0
1
Count
2000
1500
1000
500
0
D
F
A
B
C
Warehouse Block


Mode of Shipment vs On-Time Delivery
Reached on Time
0
1
Count
4000
3000
2000
1000
0
Flight
Ship
Road
Mode of Shipment





Correlation analysis and visualizations used to examine relationships between numerical features and the target variable. The analysis showed that:

- Some features exhibit moderate relationships with delivery outcomes
- No single feature alone can determine whether a shipment will be delivered on time

This reinforces the need for machine learning models that can learn from multiple interacting features rather than relying on simple rules.

From the exploratory analysis, the following insights were obtained:

- Shipment-related features such as weight and discount show noticeable influence on delivery performance
- The dataset contains a mix of numerical and categorical features, requiring proper preprocessing

Overall, EDA helped guide decisions related to feature encoding, scaling and model selection in later stages of the project.

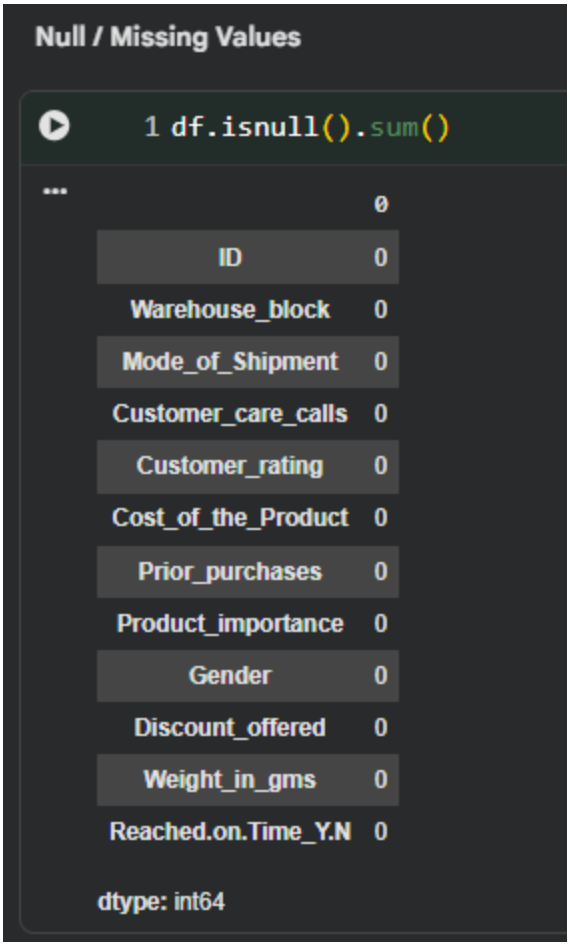
5. Dataset Pre-processing

Before training machine learning models, the dataset was carefully preprocessed to ensure data quality and compatibility with the algorithms. Each preprocessing step is discussed by first identifying the problem and then describing the solution applied.

5.1 Missing / Null Values

Problem: Missing or null values can negatively affect machine learning models, as most algorithms cannot handle incomplete data directly.

Solution: An initial inspection of the dataset using summary statistics and dataset information showed that no missing or null values were present in any of the features. Therefore, no rows or columns needed to be removed or imputed for this dataset.



5.2 Categorical Values

Problem: Several features in the dataset are categorical in nature, such as warehouse block, mode of shipment, product importance and gender. Machine learning models cannot directly process categorical text values.

Solution: Categorical variables were converted into numerical format using One-Hot Encoding. This encoding technique was chosen because it does not impose any ordinal relationship between categories and is suitable for models such as Logistic Regression, KNN and Neural Networks.

5.3 Feature Scaling

Problem: The numerical features in the dataset have different ranges and units. For example, product weight is measured in grams and has much larger values compared to customer rating. Models that rely on distance calculations or gradient-based optimization can be biased by such differences.

Solution: Feature scaling was applied to the numerical input features using standardization. This transforms features to have a mean of 0 and a standard deviation of 1. Scaling was applied only to the input features and not to the target variable, as the target represents class labels.
Feature scaling improved the performance and stability of models such as KNN, Logistic Regression and Neural Networks.

6. Dataset Splitting

After preprocessing, the dataset was divided into training and testing sets to evaluate model performance on unseen data.

6.1 Train -Test Split

The dataset was split using a 70% training and 30% testing ratio. This split provides sufficient data for model training while retaining a large enough test set for reliable evaluation. The training set was used to train the machine learning models, while the testing set was kept completely unseen during training. This approach allows for a fair and unbiased evaluation of model generalization performance.

6.2 Stratified Sampling

Since the dataset is imbalanced, stratified sampling was applied during the train-test split. This ensures that the class distribution of the target variable remains consistent across both training and testing sets.

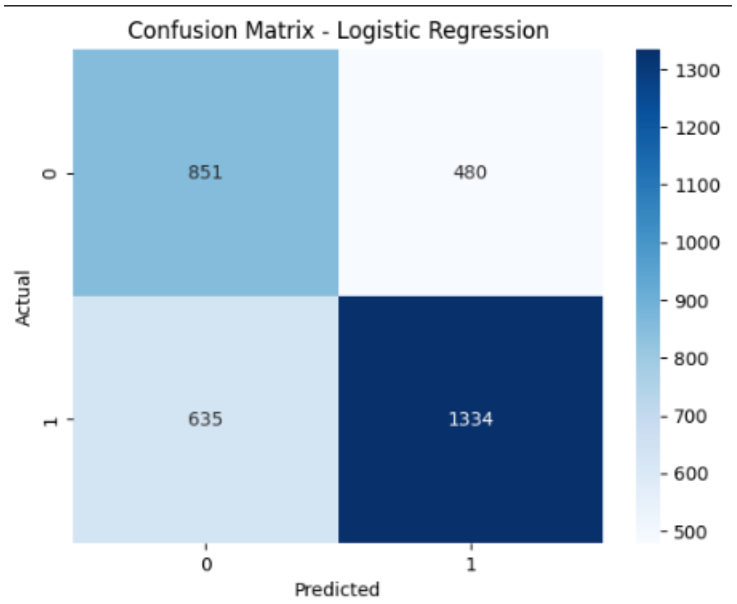
7. Model Training and Testing (Supervised Learning)

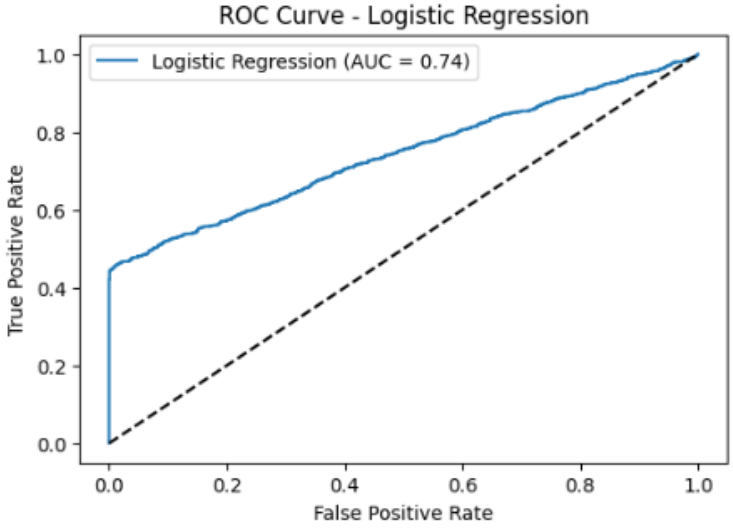
To solve the classification problem, multiple supervised machine learning models were trained and evaluated. Using different models allows for performance comparison and helps identify the most suitable approach for predicting on-time delivery.
Each model was trained using the training dataset and evaluated using the test dataset. Standard evaluation metrics were applied to ensure fair comparison.

7.1 Logistic Regression

Logistic Regression was used as a baseline classification model. The model predicts the probability of a shipment reaching on time by applying a sigmoid function to a linear combination of input features. Logistic Regression is computationally efficient and provides interpretable results.

After training, the model achieved moderate performance. Due to its linear nature, it struggled to capture more complex, non-linear relationships present in the dataset. However, it served as a useful reference point for comparing more advanced models.



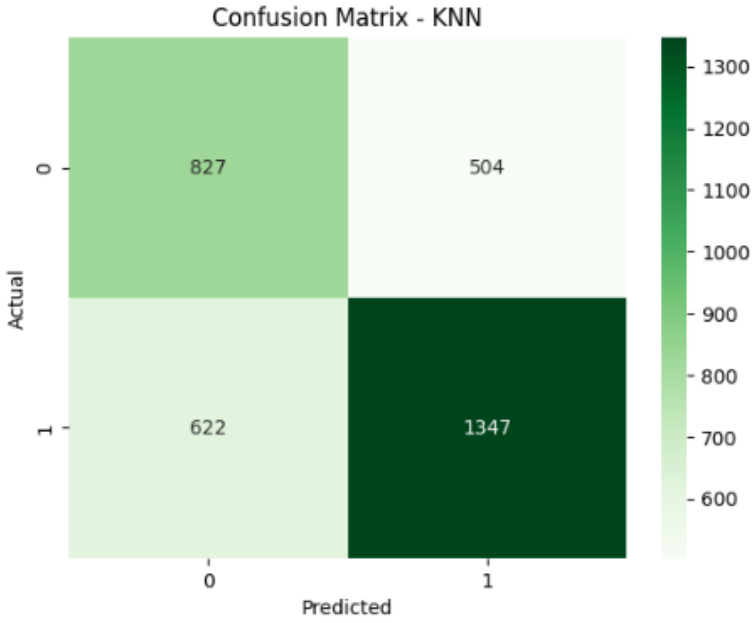


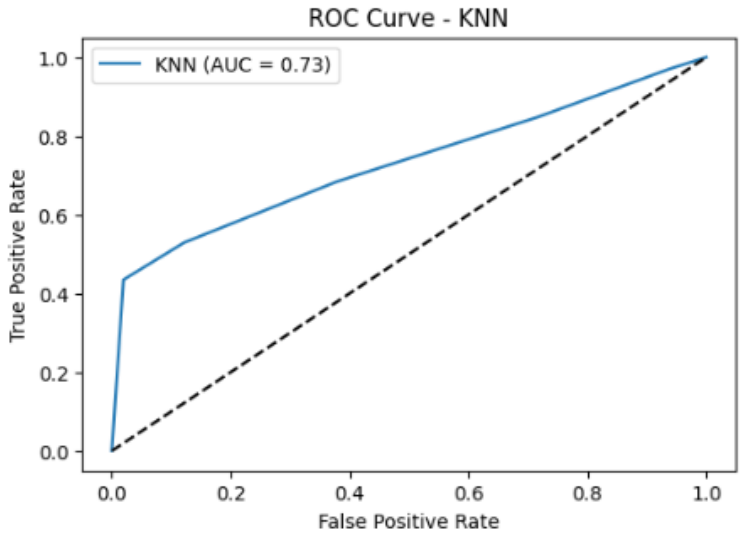
7.2 K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) is a distance-based classification algorithm that assigns a class to a data point based on the majority class of its nearest neighbors.

In this project, KNN classified shipments by measuring the distance between data points in the feature space. Feature scaling was particularly important for this model, as distance calculations are sensitive to differences in feature magnitude.

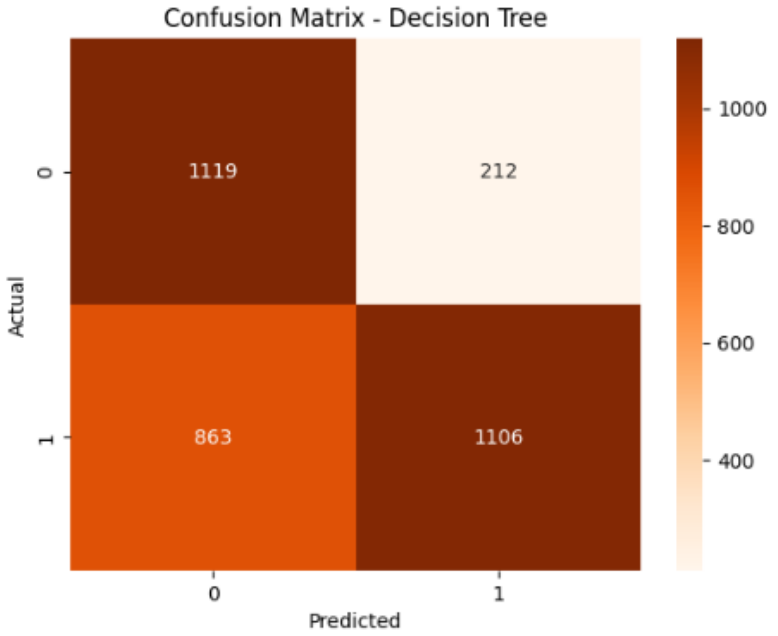
KNN demonstrated reasonable performance but was affected by noise and the choice of the number of neighbors. While simple and intuitive, its performance was slightly lower than more complex models.

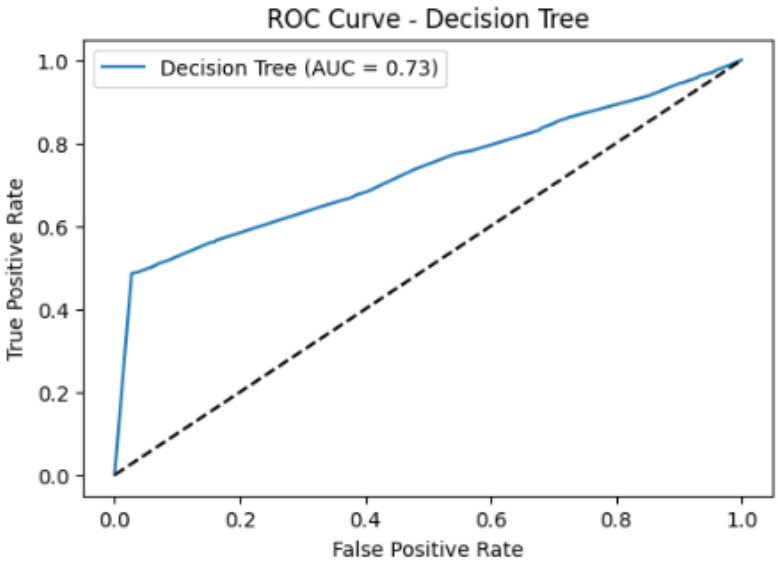




7.3 Decision Tree

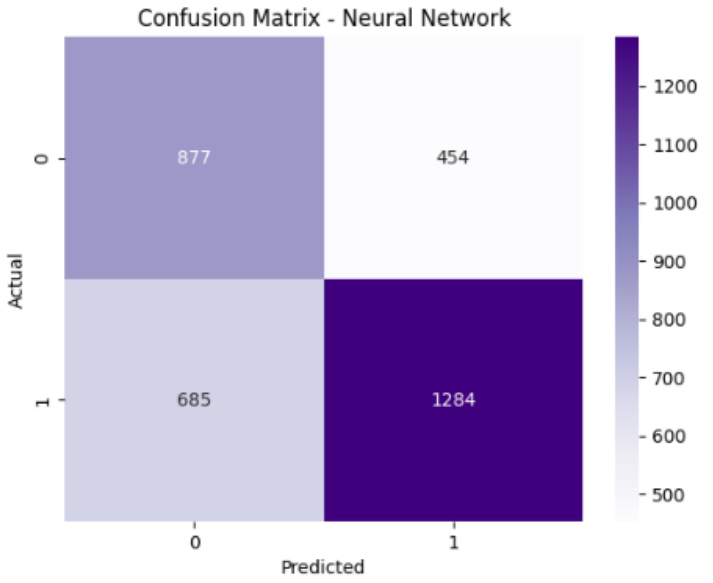
The Decision Tree model classifies data by recursively splitting the feature space based on conditions that maximize class separation. The Gini impurity criterion was used to determine the quality of each split. Decision Trees are capable of capturing non-linear relationships and feature interactions. In this project, the Decision Tree model performed slightly better than linear models in terms of accuracy.

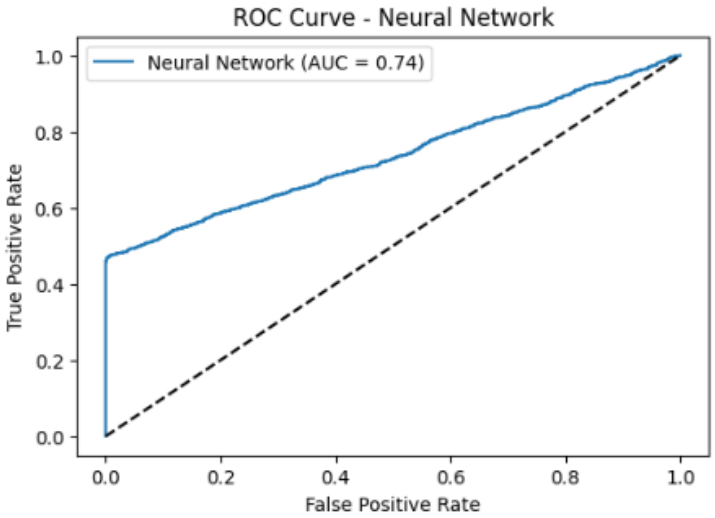




7.4 Neural Network

A Neural Network model was implemented to capture complex non-linear patterns in the data. The network consists of multiple layers of neurons that apply weighted transformations and activation functions to the input features. The Neural Network demonstrated strong performance compared to other models. Its ability to learn intricate feature interactions allowed it to generalize better on unseen data. Although less interpretable than simpler models, it achieved a more balanced performance across evaluation metrics.





All supervised models were trained using the same training data and evaluated on the same test data to ensure consistency. While simpler models provided baseline performance, more advanced models such as Decision Tree and Neural Network showed improved results by capturing non-linear relationships.

8. Unsupervised Learning: K-Means Clustering

In addition to supervised learning models, the dataset was also analyzed using an unsupervised learning approach. Unsupervised learning helps discover hidden patterns in the data without using labeled outputs.

8.1 Conversion to Unsupervised Dataset

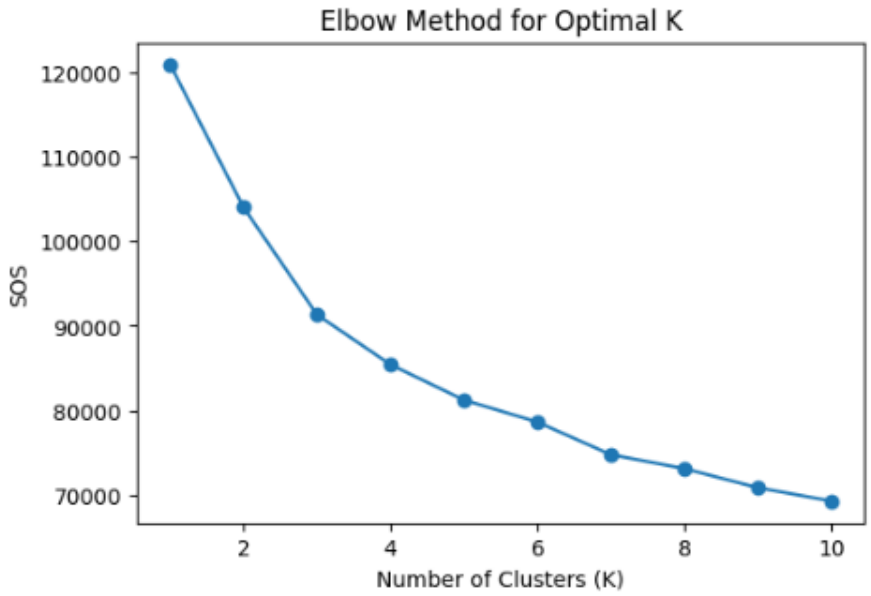
To apply unsupervised learning, the target variable Reached.on.Time_Y.N was removed from the dataset. Only the input features were used for clustering, as unsupervised algorithms must not rely on labeled data. All features used for clustering were numerical and scaled to ensure meaningful distance calculations.

8.2 K-Means Clustering

K-Means clustering was applied to group shipments into clusters based on feature similarity. This algorithm works by assigning data points to the nearest cluster centroid and updating centroids iteratively until convergence.

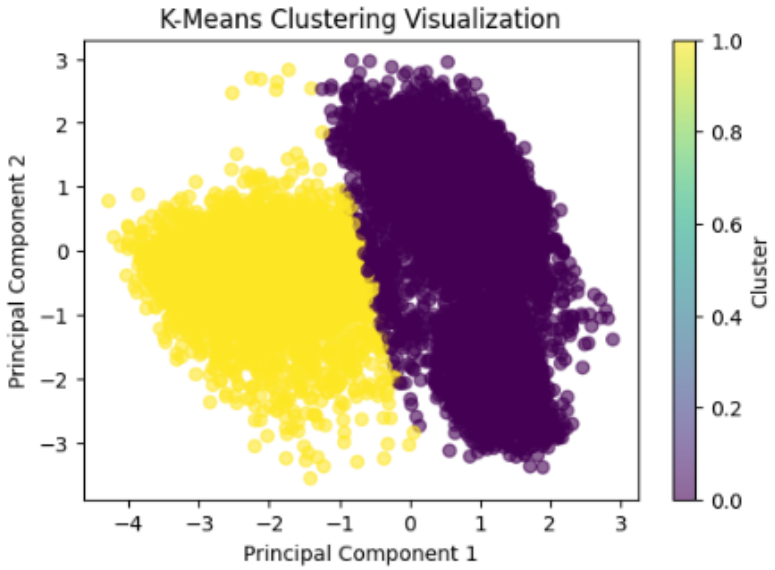
8.3 Selection of Number of Clusters

The Elbow Method was used to determine the optimal number of clusters. By plotting the sum of squared distances (inertia) against different values of K, a noticeable “elbow” point was observed at K = 2. Based on this observation, two clusters were selected for analysis, as increasing the number of clusters beyond this point resulted in only marginal improvements.



8.4 Interpretation of Clusters

The resulting clusters represent different shipment behavior patterns present in the data. Since K-Means is an unsupervised algorithm, the clusters do not directly correspond to the actual delivery labels. Instead, they provide exploratory insights into how shipments can be grouped based on shared characteristics. This unsupervised analysis complements the supervised learning models by offering additional understanding of the dataset structure.



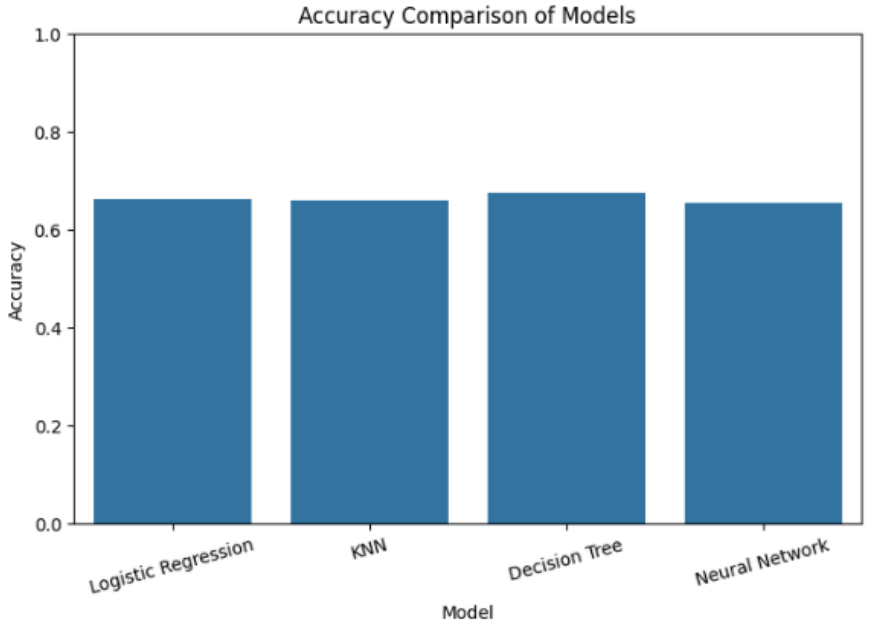
9. Model Selection and Comparison Analysis

After training all supervised learning models, their performances were compared using multiple evaluation metrics. Since the dataset is imbalanced, relying on a single metric such as accuracy is insufficient. Therefore, a combination of accuracy, precision, recall, F1-score, confusion matrix and ROC-AUC was used for comprehensive evaluation.

9.1 Accuracy Comparison

A bar chart was used to compare the prediction accuracy of all supervised models, including Logistic Regression, KNN, Decision Tree, and Neural Network.

The accuracy results showed that all models achieved similar performance, with values ranging between approximately 65% and 68%. While the Decision Tree model achieved slightly higher accuracy, the differences among models were not substantial. This observation highlights that accuracy alone is not sufficient to determine the best-performing model, especially in the presence of class imbalance.

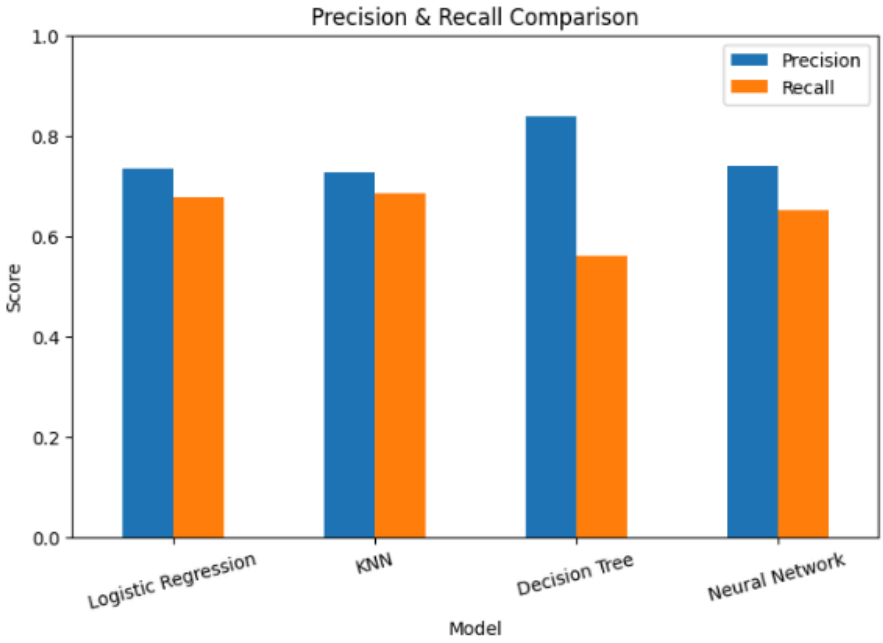


9.2 Precision and Recall Comparison

Precision and recall were compared across all models to better understand their behavior in identifying on-time and delayed shipments.

- Decision Tree showed higher precision but lower recall, indicating a more conservative prediction behavior.
- Neural Network achieved a more balanced precision-recall trade-off.
- Logistic Regression and KNN demonstrated moderate and comparable performance.

This comparison reveals how different models manage false positives and false negatives differently.

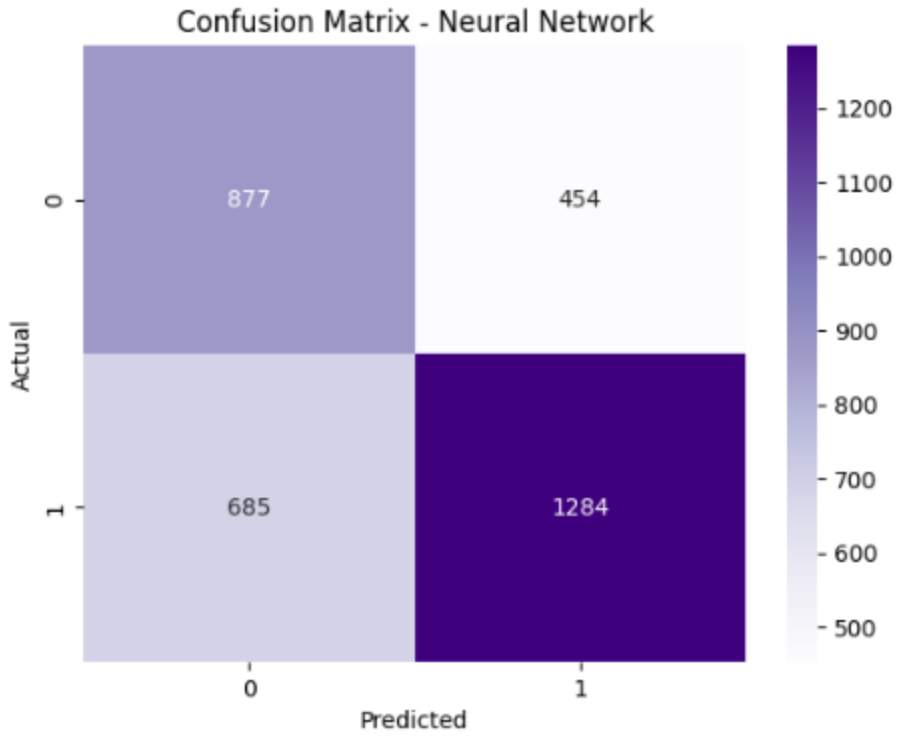


9.3 Confusion Matrix Analysis

Confusion matrices were generated for each model to visualize correct and incorrect predictions. The confusion matrix analysis showed that:

- Most models performed better on the majority class
- A noticeable number of misclassifications existed, particularly false negatives
- The Neural Network reduced misclassification more effectively than simpler models

Confusion matrices provided detailed insight into the types of errors made by each model.

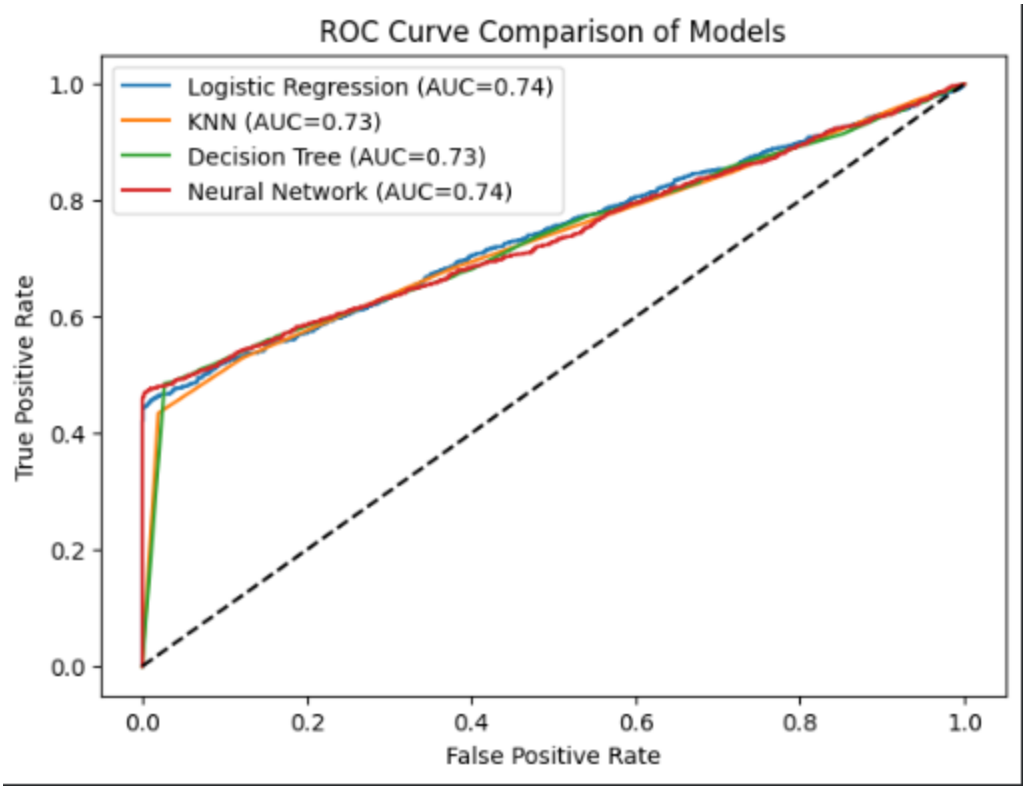


9.4 ROC Curve and AUC Score

Receiver Operating Characteristic (ROC) curves were plotted for all supervised models. The ROC curve illustrates the trade-off between true positive rate and false positive rate across different thresholds.

The Area Under the Curve (AUC) scores for all models were above 0.7, indicating performance better than random classification. Among the models, the Neural Network and Logistic Regression achieved the highest AUC values, demonstrating stronger discriminative ability.

ROC-AUC proved to be a reliable metric for comparing model performance on this imbalanced dataset.



Although accuracy values were similar across models, the Neural Network demonstrated the most balanced overall performance when considering precision, recall, F1-score and ROC–AUC together. Therefore, the Neural Network was selected as the most suitable model for predicting on-time delivery in this project.

10. Conclusion

In this project, multiple machine learning techniques were applied to predict whether an e-commerce shipment would reach the customer on time. The project followed a structured workflow that included data exploration, preprocessing, model training, evaluation and comparison using both supervised and unsupervised learning approaches.

Exploratory Data Analysis provided valuable insights into feature distributions, correlations, and class imbalance, which guided preprocessing decisions such as encoding and feature scaling. Several supervised models including Logistic Regression, KNN, Decision Tree and Neural Network were trained and evaluated using appropriate performance metrics.

The results showed that while all models achieved similar accuracy, accuracy alone was not sufficient due to class imbalance. Metrics such as precision, recall, F1-score and ROC-AUC provided a more meaningful evaluation. Among the supervised models, the Neural Network demonstrated the most balanced and reliable performance across these metrics, making it the most suitable model for this problem.

Unsupervised learning using K-Means clustering was also applied to explore hidden patterns in the dataset. Although the clusters did not directly correspond to delivery labels, the analysis offered additional insights into shipment behavior and data structure.

Several challenges were encountered during the project, including handling mixed feature types, managing class imbalance, and selecting appropriate evaluation metrics. These challenges were addressed through careful preprocessing, model selection, and comprehensive performance analysis.

Overall, this project highlights the importance of using multiple machine learning models and evaluation metrics to solve real-world classification problems. The systematic approach adopted in this work ensures robustness, interpretability and meaningful conclusions from the data.