CSE422: Artificial Intelligence

Lab Project Report

Name: Md. Rafiul Islam

Student ID: 22201842

Submission Date: 12 May, 2025

Contents

Topics Page No.

Introduction

With the continually evolving nature of e-commerce, the secret to customer satisfaction is dependent on reliable logistics and rapid delivery. As online shopping gains greater usage, supply chain activity has been further complicated, hence more of data-driven methodology being adopted. The accurate forecasting of timeliness in shipment is one of the greatest challenges in this area. This project attempts to address this problem by employing machine learning algorithms on historical shipping records from online shopping websites in order to predict if the product arrived on time.

Developing accurate and repeatable classification models that predict on-time delivery depending on a series of parameters - such as shipment weight, delivery distance, mode of shipping, and customer feedback - is the overall objective of this study. The project leverages supervised learning techniques like Logistic Regression, Decision Tree, KNN, and Neural Networks to identify crucial information that could improve logistics decision-making. This automation of the prediction, the study not only minimizes the chances of human errors, but also enables more equitable operating policies in terms of clear-cut e-business supply chain systems.

Dataset Description

The dataset utilized by this research is the E-commerce Shipping Dataset, and it contains data on individual shipping orders, whose purpose is to forecast whether a product was delivered on time or not. The dataset contains 10,999 rows and has 12 original attributes, with 11 being input features and one being the binary target variable: `Reached.on.Time_Y.N` (0 = Delivered on Time, 1 = Not Delivered on Time).

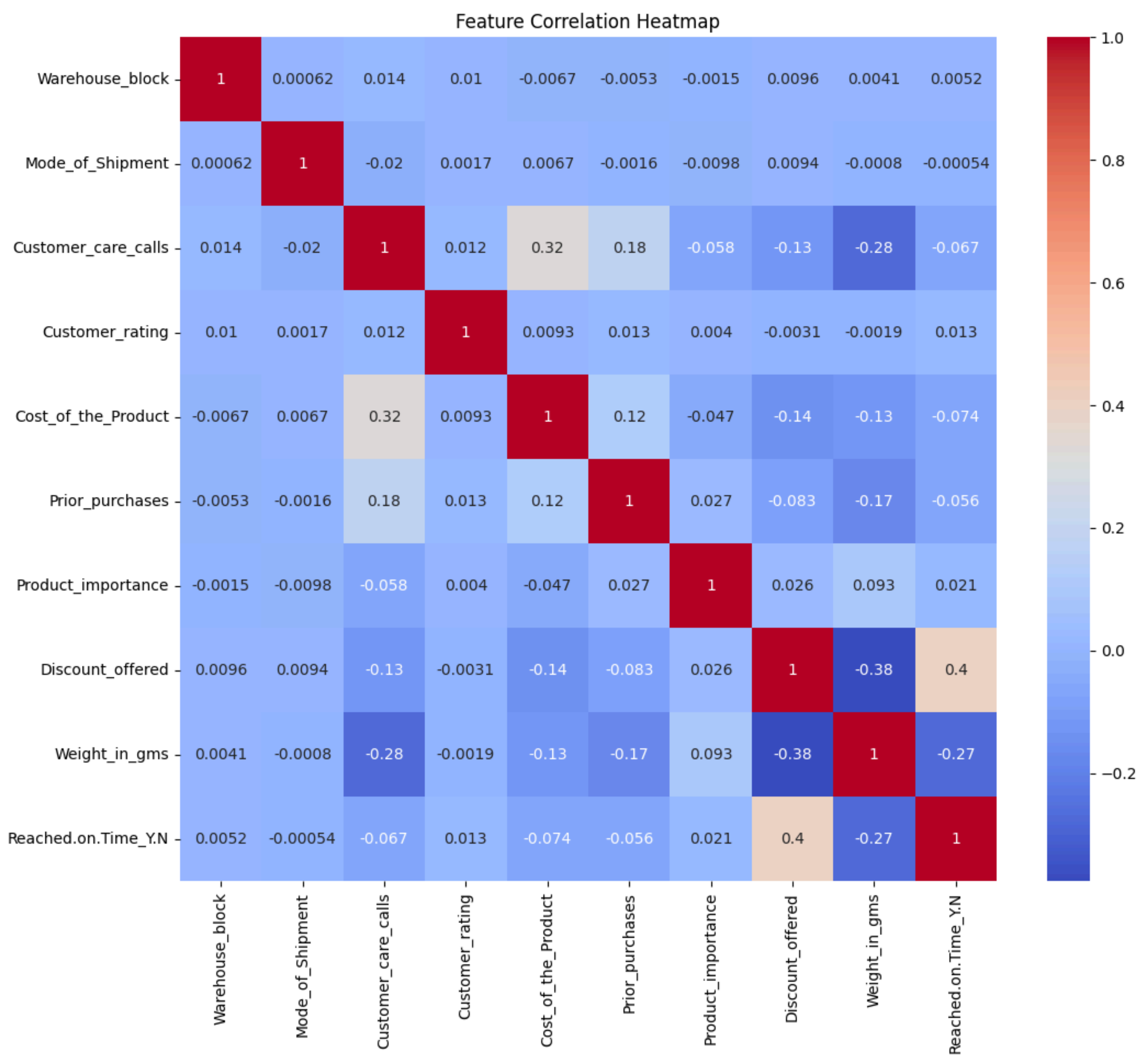
Two columns, ID and Gender, were dropped for feature correlation analysis and model training. The ID column is of no predictive value and is just a unique identifier. Gender was dropped to avoid any potential bias and since it has a very minimal impact on delivery outcome in preliminary analysis.

The final group of the features is a mix of quantitative and qualitative variables:

- **Quantitative Features:**
    - `Customer_care_calls`: Number of times the customer contacted customer service.
    - `Customer_rating`: Rating provided by the customer (1 to 5).
    - `Cost_of_the_Product`: Monetary cost of the product.
    - `Prior_purchases`: Number of previous purchases by the customer.
      `Discount_offered`: Discount applied to the product
    - `Weight_in_gms`: Weight of the shipment in grams.

- **Categorical Features (encoded for analysis):**
    - `Warehouse_block`: Encoded representation of the warehouse block where the product originated.
    - `Mode_of_Shipment`: Shipping method used (Flight, Ship, or Road).
    - `Product_importance`: Ordinal encoding for product priority (low = 0, medium = 1, high = 2).

Correlation matrix was obtained after performing suitable preprocessing, i.e., encoding categorical variables. It was found through the analysis that some features, like `Discount_offered` and `Weight_in_gms`, have a significant correlation with the target variable, indicating their possible predictive capability. Other less individually related characteristics may still contribute helpful interactions when combined with others in multivariate models.This filtered dataset forms the foundation for training and testing a variety of classification models that are engineered to predict e-commerce logistics delivery delays.



	Warehouse_block	Mode_of_Shipment	Customer_care_calls	Customer_rating	Cost_of_the_Product	Prior_purchases	Product_importance	Discount_offered	Weight_in_gms	Reached.on.Time_Y.N
Warehouse_block	1	0.00062	0.014	0.01	-0.0067	-0.0053	-0.0015	0.0096	0.0041	0.0052
Mode_of_Shipment	0.00062	1	-0.02	0.0017	0.0067	-0.0016	-0.0098	0.0094	-0.0008	-0.00054
Customer_care_calls	0.014	-0.02	1	0.012	0.32	0.18	-0.058	-0.13	-0.28	-0.067
Customer_rating	0.01	0.0017	0.012	1	0.0093	0.013	0.004	-0.0031	-0.0019	0.013
Cost_of_the_Product	-0.0067	0.0067	0.32	0.0093	1	0.12	-0.047	-0.14	-0.13	-0.074
Prior_purchases	-0.0053	-0.0016	0.18	0.013	0.12	1	0.027	-0.083	-0.17	-0.056
Product_importance	-0.0015	-0.0098	-0.058	0.004	-0.047	0.027	1	0.026	0.093	0.021
Discount_offered	0.0096	0.0094	-0.13	-0.0031	-0.14	-0.083	0.026	1	-0.38	0.4
Weight_in_gms	0.0041	-0.0008	-0.28	-0.0019	-0.13	-0.17	0.093	-0.38	1	-0.27
Reached.on.Time_Y.N	0.0052	-0.00054	-0.067	0.013	-0.074	-0.056	0.021	0.4	-0.27	1

Correlation Heatmap

Data Preprocessing

Proper preprocessing is an essential step to make the data ready for machine learning algorithms, particularly for mixed data types and different value ranges. The preprocessing steps applied to the *E-commerce Shipping Dataset* are as follows:

1. **Handling Missing Values**
   A preliminary verification with `df.isnull().sum()` ensured that the dataset has no missing values. Thus, neither imputation nor row removal was necessary, thereby enabling the maintenance of the original data structure.
2. **Dropping Irrelevant Features**
   The `ID` column was removed because it served the purpose of a unique identifier only and does not provide any predictive information. Also, in this implementation, the `Gender` column was excluded to keep the feature set simple and avoid any chance of data leakage or bias, as it wasn't adding much value to the model performance during exploratory analysis.
3. **Encoding Categorical Variables**
   Machine learning algorithms require numeric inputs, so all categorical variables were transformed into numeric format
   - Categorical features such as `Warehouse_block`, `Mode_of_Shipment`, and `Product_importance` were **one-hot encoded** using `pd.get_dummies()` with `drop_first=True` to prevent multicollinearity.
   - As the Gender feature was excluded, binary encoding was not applied in this implementation.
4. **Feature Scaling**
   Several numeric features, including `Cost_of_the_Product`, `Weight_in_gms`, and `Discount_offered`, span different ranges. This poses a challenge for algorithms like **K-Nearest Neighbors (KNN)** and **Neural Networks**, which are sensitive to feature scales. To address this, **Min-Max Scaling** was applied using `MinMaxScaler()` from `sklearn.preprocessing`, normalizing all features to a [0, 1] range.

5. **Final Dataset for Modeling**

   After preprocessing, the dataset was split into:

   - X: a DataFrame containing all scaled input features.
   - y: The dependent variable (Reached.on.Time_Y.N) is the on-time delivery indicator.

These preprocessing steps ensured that the dataset was clean, well-encoded, and well-scaled, and ready to be trained on a variety of classification algorithms.

Dataset Splitting

In order to ascertain that the machine learning model generalizes well to new and unseen data, it is important to divide the dataset into training and test sets. By doing so, the model learns from a part of the data and gets tested on a different part to evaluate its performance.

In this project, the 10,999 data point dataset was split in the following manner:

- **Training Set (70%)**: 7,699 data points that the model learns from to find patterns and relationships.
- **Testing Set (30%)**: 3,300 data points, reserved for testing the performance of the model in making predictions on new, unseen data.

The dataset was split using the `train_test_split()` function from the `sklearn.model_selection` module. The function ensures random splitting of the dataset while preserving the distribution of the target variable (`Reached.on.Time_Y.N`) in both training and test sets by utilizing the `stratify=y` parameter. The `random_state=20` was utilized to ensure reproducibility of the results.

The split resulted in:

- **Training samples**: 7,699
- **Testing samples**: 3,300

This partition ensures that the model is trained on a sufficient amount of data while being evaluated on a separate, unbiased set of data to gauge its performance.

Model Training & Testing :

Four separate machine learning models were trained and tested on the preprocessed dataset:

1. **Logistic Regression:** A linear model that calculates the probability of a binary event (on-time delivery or not) by applying the logistic (sigmoid) operation.

2. **Decision Tree:** A tree-based model in which at each node the dataset is split into branches of different feature values, where the split is decided based on measures such as entropy or Gini impurity.

3. **K-Nearest Neighbors (KNN):** A non-parametric method in which class assignment to a data point is given by the majority class of its neighbors nearby in the feature space.

4. **Neural Networks:** Suppose to capture more complex relationships due to its nonlinearity by passing data through multiple layers of transformation.

Each machine-learning model was trained on the **training set**, while being tested on the **testing set** for the predictive performance estimation. This thus allows seeing how well each model can generalize to new data.

Model Selection/Comparison Analysis

To evaluate the performance of the developed machine learning models, several key metrics were considered: **Accuracy**, **Precision**, **Recall**, **F1-Score**, and **ROC-AUC Score**. These metrics provide a comprehensive view of each model's effectiveness in predicting whether a shipment would be delayed or delivered on time.

Model Performance Summary:

Model	Accuracy	Precision	Recall	F1-score	AUC-score
K-Nearest Neighbour (KNN)	**0.64**	**0.71**	**0.67**	**0.69**	**0.71**
Decision Tree	**0.63**	**0.69**	**0.69**	**0.69**	**0.61**
Logistic Regression	**0.64**	**0.70**	**0.69**	**0.69**	**0.72**
Neural Network	**0.66**	**0.80**	**0.57**	**0.67**	**0.73**

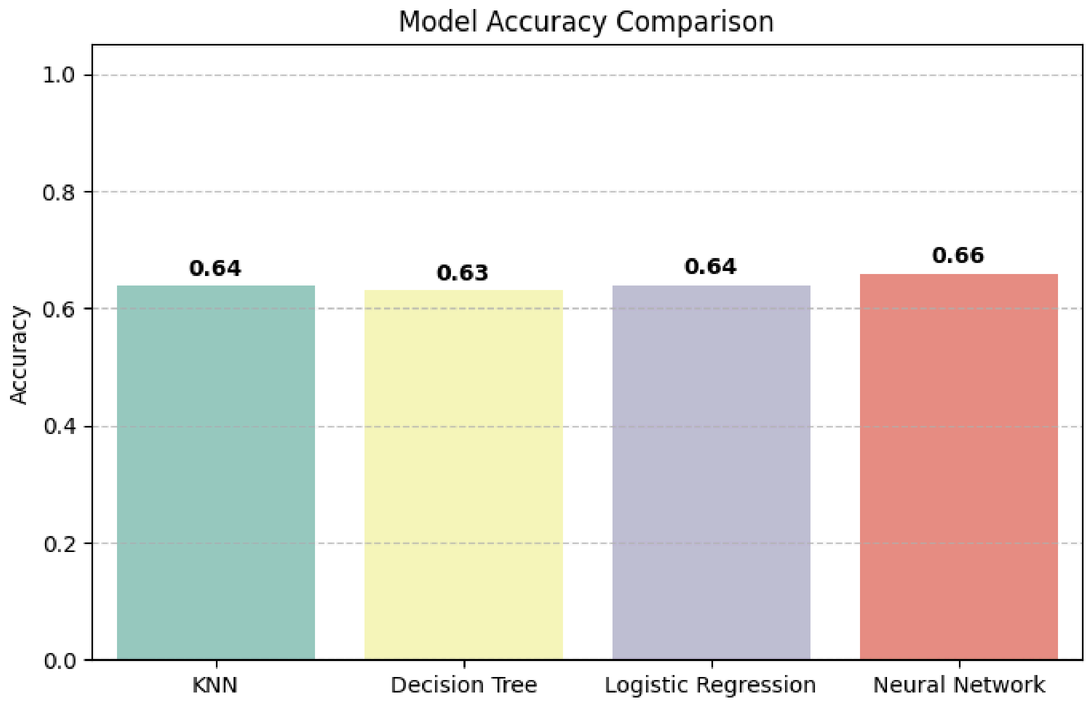
According to the comparison, Neural Network produced the highest AUC Score (0.73) and Precision (0.80), indicating high discriminant power and less false positives. However, it produced the lowest Recall (0.57), indicating that it missed many true delays.

Logistic Regression presented the best trade-off between Precision (0.70), Recall (0.69), and F1-Score (0.69), in addition to a high AUC Score (0.72), which makes it the prime candidate for stable and interpretable deployment.

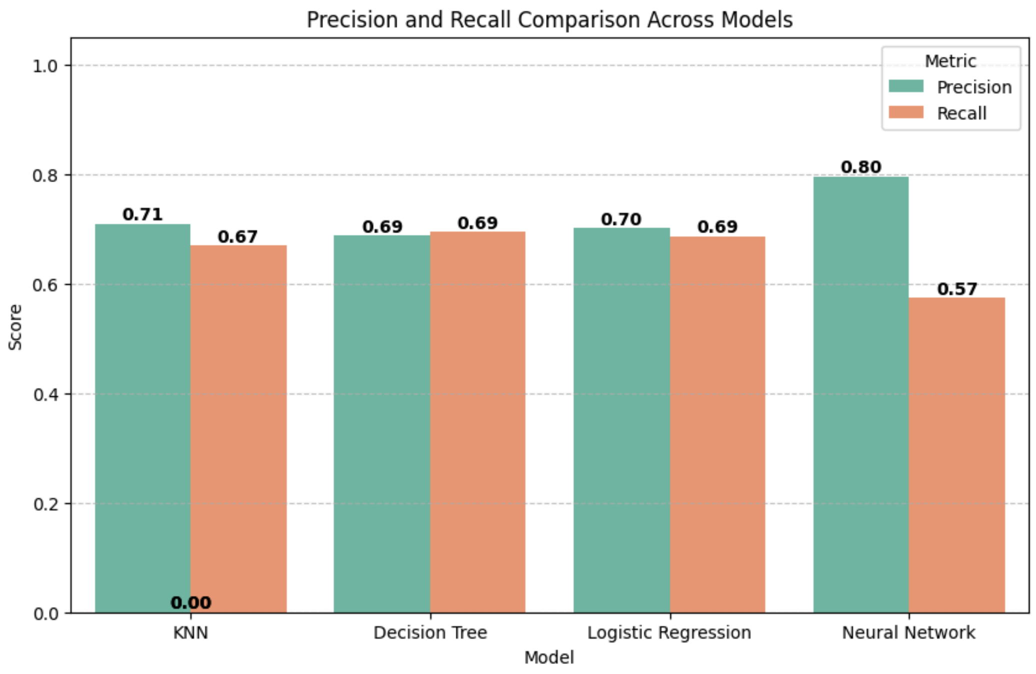
KNN performed like Logistic Regression with a bit better Precision but worse overall accuracy and computational efficiency. Decision Tree had the same values for Precision, Recall, and F1-Score but worse AUC Score, indicating worse class separation.

In conclusion, the optimal model choice is a question of business necessity - Neural Networks will be the optimal choice when reduction of false positives is the most important consideration, but Logistic Regression yields a more understandable and balanced solution for general deployment in predictive logistics solutions.

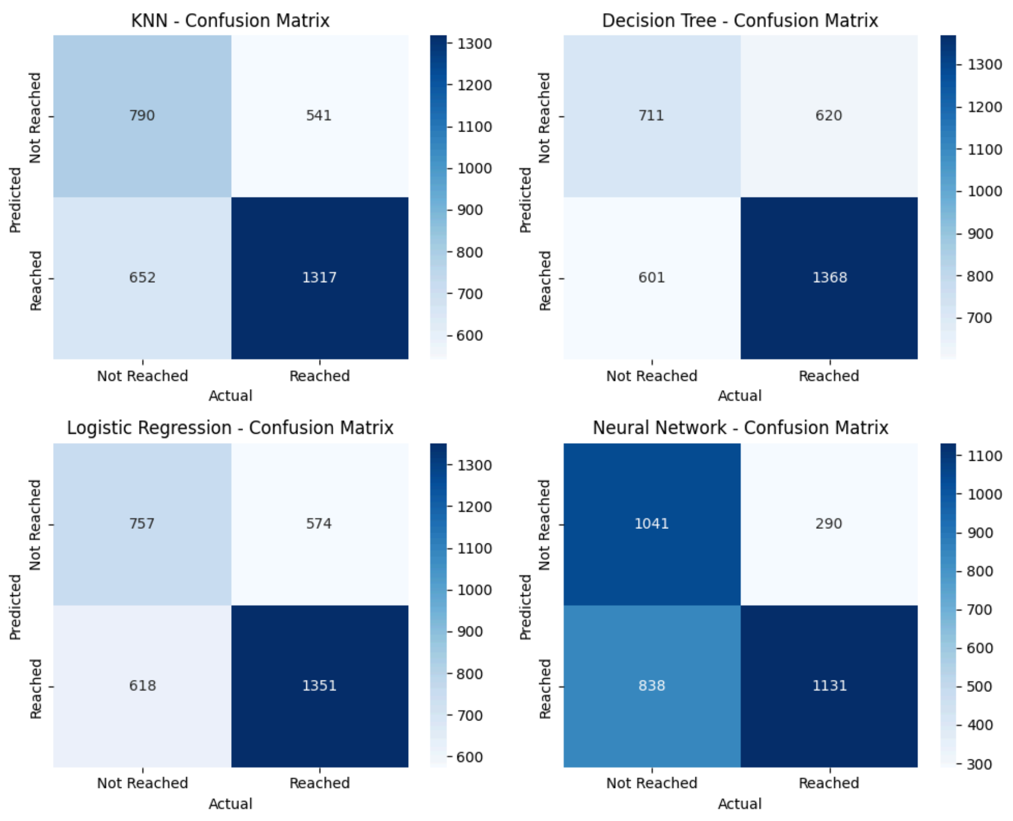
Outputs:



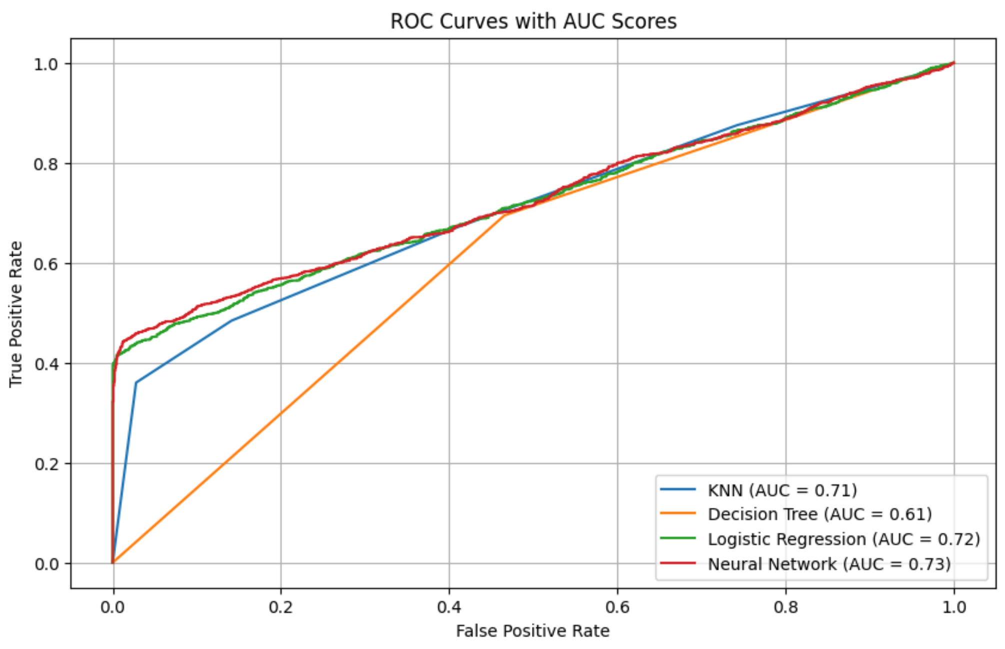
Model Accuracy Bar chart



Precision and Recall Comparison for each model



Confusion Matrix



ROC Curves and AUC Scores

Conclusion

From the results of this project, it is evident that machine learning models can effectively be applied to predict delivery delays in e-commerce logistics using historical shipping data. Among the models tested - **Logistic Regression**, **K-Nearest Neighbors (KNN)**, **Decision Tree**, and **Neural Network** - the **Neural Network** achieved the highest **Precision (0.80)** and **AUC Score (0.73)**, indicating strong potential for correctly identifying on-time deliveries. However, it had a relatively lower **Recall (0.57)**, suggesting it missed more actual delays than the other models.

Logistic Regression, on the other hand, demonstrated a more balanced performance across all evaluation metrics, with an **Accuracy of 0.64**, **F1-Score of 0.69**, and **AUC Score of 0.72**, making it the most consistent and interpretable model for general use. Models like **KNN** and **Decision Tree** also performed moderately well, but were either computationally less efficient (in the case of KNN) or less robust in terms of ROC-AUC (as seen with the Decision Tree).

These results may be influenced by a few key factors:

- The class distribution in the dataset is slightly imbalanced, which can affect model performance, especially in Recall.
- Some features may have limited predictive power individually but contribute more when combined, which benefits models capable of capturing feature interactions.
- Simpler models like Logistic Regression may perform comparably to complex models if the data is well-preprocessed and structured.

Challenges Faced

- **Preprocessing the Data**: Handling categorical variables and scaling numeric features took extra effort to make the data suitable for different models.
- **Choosing the Right Model**: Some models were easier to understand but less accurate, while others (like Neural Networks) performed better but were harder to interpret.
- **Evaluating the Models**: Just using accuracy wasn't enough. We had to look at other metrics like Precision, Recall, and AUC to get a complete picture.

- **Slight Class Imbalance**: Since the target variable wasn’t perfectly balanced, it made it harder for models to correctly identify delayed shipments.

In summary, the project highlights both the promise and complexity of applying machine learning to logistics problems, and demonstrates that with proper preprocessing and evaluation, predictive models can offer valuable operational insights.