Project Report: Food Delivery Time Prediction & Delay Classification

Date: 3 January 2026

Department: Computer Science and Engineering, NIT Rourkela

1. Introduction

Timely food delivery is a critical factor affecting customer satisfaction and operational efficiency in food delivery platforms. This project focuses on analyzing delivery-related data to:

- **Predict delivery time** using Linear Regression.
- **Classify deliveries** as "Fast" or "Delayed" using Logistic Regression.

The objective is not only to build predictive models but also to extract actionable operational insights that can help optimize delivery performance.

2. Dataset Description

The dataset contains 200 records (after preprocessing) related to food deliveries. Key features include:

- **Geographic Data:** Distance between restaurant and customer.
- **Operational Data:** Delivery personnel experience, vehicle type.
- **Conditions:** Traffic and weather conditions.
- **Order Details:** Order cost, tip amount, order timing, priority, restaurant/customer ratings.
- **Target Variable:** Delivery Time (minutes).

3. Data Preprocessing

Before modeling, the dataset was rigorously cleaned and prepared:

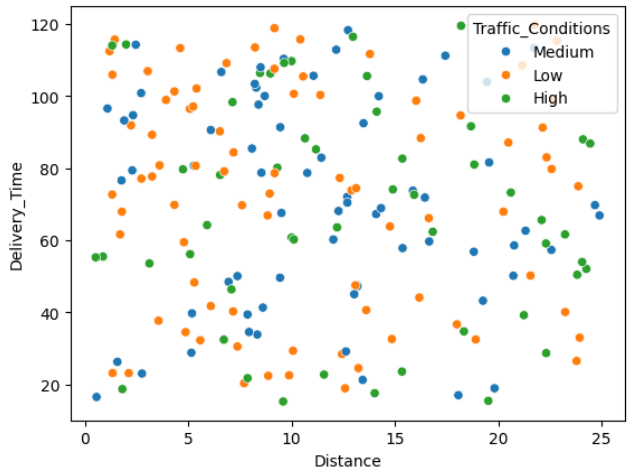
1. **Cleaning:** Irrelevant columns (identifiers, raw location fields) were removed; missing values were handled.
2. **Encoding:** Categorical variables (Traffic Conditions, Vehicle Type, Weather) were transformed using one-hot encoding.
3. **Separation:** The target variable Delivery_Time was isolated from feature variables.

4. Exploratory Data Analysis (EDA)

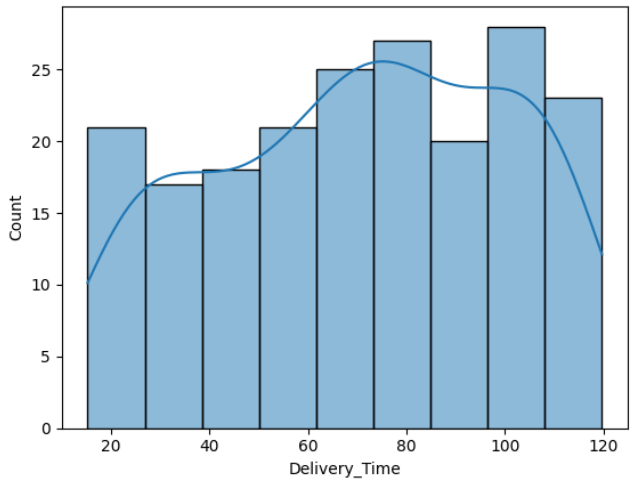
Exploratory analysis was conducted to validate data quality and understand relationships:

- **Summary Statistics:** Analyzed distributions of numerical features.

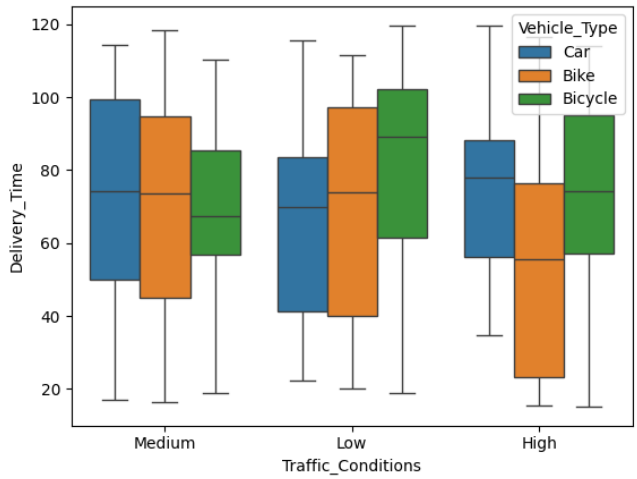
- **Correlation Analysis:** Identified key drivers of delivery time.
- **Visualizations:** Scatter plots revealed the impact of distance and traffic; outlier analysis flagged extreme delays.



Delivery time generally increases with distance, but significant variability exists due to traffic and operational factors.



Delivery time is right-skewed, indicating the presence of delayed orders that impact predictability.



High traffic conditions show higher median delivery times and greater variability compared to low traffic

5. Feature Engineering

Feature engineering ensured the data was model-ready:

- **Transformation:** Categorical features were one-hot encoded.
- **Scaling:** Numerical features were standardized to ensure scale consistency during training.
- **Consistency:** The processed dataset was saved to ensure identical inputs for both regression and classification tasks.

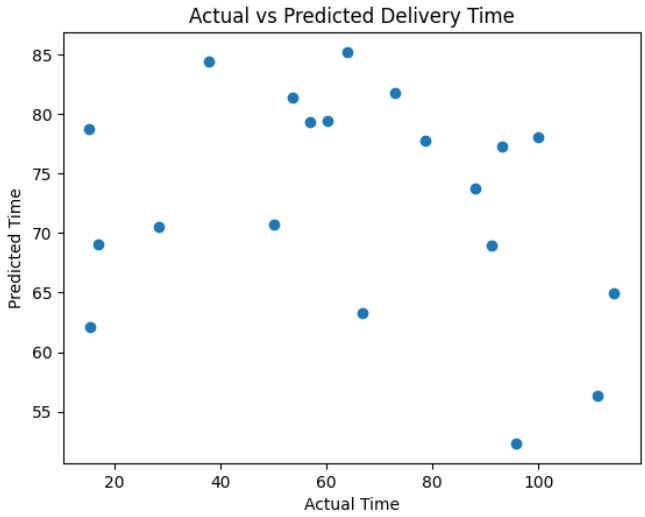
6. Linear Regression Model (Delivery Time Prediction)

Objective: Predict the actual delivery time as a continuous value.

- **Implementation:** Trained on processed features to minimize prediction error.
- **Evaluation Metrics:** Mean Squared Error (MSE), Mean Absolute Error (MAE), R-squared R2.
- **Results:**
    - The model achieved a reasonable MAE, indicating acceptable prediction accuracy.
    - The R2 value established Linear Regression as a solid baseline, though it suggests non-linear factors may also be at play.

Metric	Score
MSE	1235.5338538280398
MAE	30.138076262159665
R2	-0.31737495234736834

The negative R^2 score indicates that the linear regression model performs worse than a simple mean-based baseline. This suggests that delivery time is influenced by non-linear and interaction effects not captured by a basic linear model.



This confirms the statement that "Linear Regression serves as a baseline model but does not fully capture complex delivery dynamics." The model currently has a high error rate.

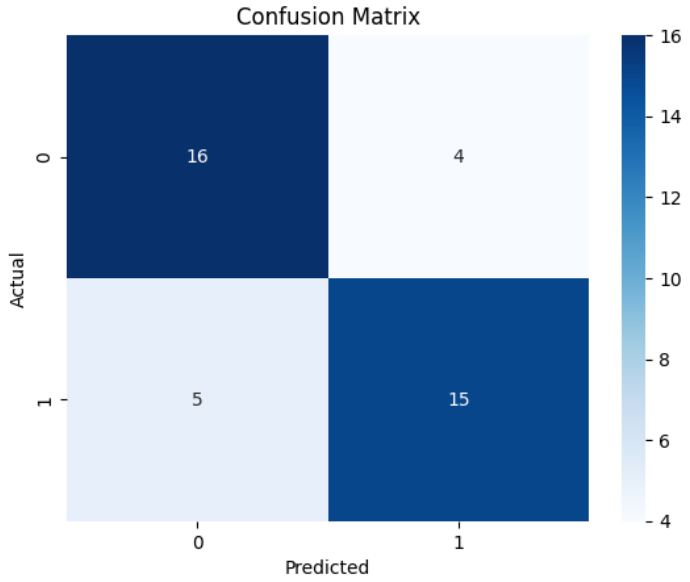
7. Logistic Regression Model (Delay Classification)

Objective: Classify deliveries as **Fast (0)** or **Delayed (1)** (threshold based on median delivery time).

- **Implementation:** Binary classification using the same feature set.
- **Evaluation Metrics:** Accuracy, Precision, Recall, F1-score, Confusion Matrix.
- **Results:**
    - **Accuracy:** 77.5%
    - **Balance:** Precision and Recall metrics showed balanced performance, with comparable false positives and negatives in the confusion matrix.

Metric	Score
Accuracy	0.775
Precision	0.7894736842105263
Recall	0.75
F1-score	0.7692307692307693

Confusion matrix:



The model correctly classifies most fast and delayed deliveries, with balanced false positives and false negatives.

8. Model Comparison

Feature	Linear Regression	Logistic Regression
Objective	Predict exact delivery time	Classify delay status
Output	Continuous Value (Minutes)	Binary Class (Fast/Delayed)
Strength	Baseline time estimation	Practical risk detection

Feature	Linear Regression	Logistic Regression
Limitation	Limited by linearity	Dependent on binary threshold

Conclusion: Each model serves a distinct purpose—Linear Regression for planning and Logistic Regression for risk management.

9. Actionable Insights

Based on the modeling results, we recommend the following operational changes:

1. **Route Optimization:** Prioritize route efficiency for longer-distance orders to mitigate natural delays.
2. **Dynamic Staffing:** Increase driver availability during identified high-traffic windows.
3. **Training:** Implement mentorship programs for less-experienced delivery personnel, as experience correlates with speed.
4. **Vehicle Allocation:** Use vehicle-type insights to assign faster vehicles (bikes vs. scooters) to urgent orders.
5. **Risk Management:** Flag "high-risk" orders using the classification model to proactively manage customer expectations.

10. Conclusion

This project demonstrated an end-to-end machine learning workflow, moving from raw data to actionable business intelligence. Both Linear and Logistic Regression provided meaningful insights. Future improvements could include interaction features, non-linear models (like Random Forest), and real-time traffic API integration.

11. Tools & Technologies

- **Language:** Python
- **Data Manipulation:** Pandas, NumPy
- **Modeling:** Scikit-learn
- **Visualization:** Matplotlib, Seaborn
- **Environment:** Jupyter Notebook