📝 Final Project Report: Food Delivery Time Prediction

◆ 1. Dataset Description & Preprocessing Steps

Dataset Used: `Food_Delivery_Time_Prediction.csv`
Target Variable:

- For regression: `Delivery_Time` (continuous)
- For classification: `Delivery_Status` (binary — whether delivery time is above median)

Features Considered:

- Distance (calculated via Haversine formula from lat/lon)
- Weather conditions
- Traffic conditions
- Vehicle type
- Order priority
- Order cost

Preprocessing Steps:

- Categorical variables (`Weather_Conditions`, `Traffic_Conditions`, `Vehicle_Type`, `Order_Priority`) were encoded using `LabelEncoder`.
- Numerical features (`Distance`, `Order_Cost`) were standardized using `StandardScaler`.
- Target variable `Delivery_Time` was normalized using `MinMaxScaler`.
- Latitude and longitude were extracted from string fields and converted into Haversine distance for better spatial representation.

◆ 2. Model Evaluation & Comparison

📌 Linear Regression (Predicting `Delivery_Time` as a continuous variable)

Metrics:

- Mean Squared Error (MSE): `0.0834`
- Mean Absolute Error (MAE): `0.2446`
- R^2 Score: `0.0161`

➞ **Interpretation**:

- The **low R^2** indicates that the model explains only ~1.6% of the variance in delivery time.

- Prediction accuracy is limited, suggesting linear regression is not capturing complex relationships.

📌 Logistic Regression (Predicting if delivery time is above median)

Metrics:

- **Accuracy**: `0.525`
- **Precision**: `0.4762`
- **Recall**: `0.5556`
- **F1-Score**: `0.5128`
- **Confusion Matrix**:

```
lua
CopyEdit
[[11 11]
 [ 8 10]]
```

➡ **Interpretation**:

- **Accuracy (~52.5%)** is only marginally better than random guessing.
- **Recall (0.5556)** indicates the model catches ~55.6% of the delayed deliveries.
- The confusion matrix shows a balance of false positives and false negatives, indicating no strong class imbalance.
- **ROC AUC Score** (not numerically printed but present in the code): Helps determine threshold tuning (was plotted using `roc_curve` and `auc`).

◆ 3. Actionable Insights & Recommendations

🔍 Observations:

- The **linear model underperforms**, likely due to:
  - Lack of temporal features (e.g., delivery hour/day)
  - Potential nonlinear patterns in traffic or weather impacts
- Logistic regression performs slightly better but still lacks robustness.

✅ Recommendations for Optimization:

1. **Feature Engineering**:
   - Include time-based features: `Order_Hour`, `Day_of_Week`
   - Add route-based complexity scores or average speed estimates
   - Consider one-hot encoding for categorical data instead of label encoding
2. **Model Improvements**:
   - Use **tree-based models** (e.g., Random Forest, XGBoost) for better handling nonlinearity

    - Try **Polynomial Regression** to capture curvature in continuous targets
3. **Evaluation Enhancements**:
    - Use **cross-validation** to validate model consistency
    - Use **stratified sampling** for better class balance in logistic regression
4. **Additional Metrics**:
    - Track **AUC-ROC**, **Precision-Recall Curves**, and **Cost-sensitive learning** in case of skewed delivery penalties

---

✓ Final Summary:

This project implemented both regression and classification techniques on a food delivery dataset. While preprocessing was done carefully, both models show room for improvement. Future work should involve better feature extraction, model complexity tuning, and alternative algorithms like ensemble methods for real-world applicability.