### **Food Delivery Time Prediction**

**Objective** The goal is to predict food delivery times based on customer location, restaurant location, weather, traffic, and other factors. This involves both **data preprocessing** and building predictive models using **linear regression** and **logistic regression**.

#### **Phase 1: Data Collection and Exploratory Data Analysis (EDA)**

**Step 1 - Data Import and Preprocessing**

1. **Dataset**

Load the dataset ([Food\_Delivery\_Time\_Prediction.csv](https://drive.google.com/file/d/1gAJGXBob3zRpJkRWQhvf3a94PS6NhGjx/view?usp=sharing)).

1. **Handle Missing Values** Check for any missing or inconsistent values in columns such as Distance, Delivery\_Time, etc. and decide how to handle them, either through imputation or deletion.
2. **Data Transformation**
   * **Encode Categorical Variables**: Use one-hot encoding or label encoding for variables like Weather Conditions, Traffic Conditions, Vehicle Type.
   * **Normalize/Standardize Numeric Columns**: Normalize or standardize continuous features like Distance, Delivery\_sTime, and Order\_Cost for consistency.

**Step 2 - Exploratory Data Analysis (EDA)**

1. **Descriptive Statistics** Calculate the basic statistics for numerical features such as mean, median, mode, and variance.
2. **Correlation Analysis** Visualize correlations between features and the target variable (Delivery\_Time) to identify the most relevant predictors.
3. **Outlier Detection** Detect outliers in numerical features using boxplots and handle them appropriately.

**Step 3 - Feature Engineering**

1. **Distance Calculation** If the dataset doesn't contain an actual distance metric, calculate the distance between the customer and restaurant using latitudes and longitudes (Haversine formula).
2. **Time-Based Features** Create new features related to the time of day, such as Rush Hour vs Non-Rush Hour, to improve predictions.

#### **Phase 2: Predictive Modeling**

**Step 4 - Linear Regression Model**

1. **Train-Test Split** Split the dataset into training and testing sets (e.g., 80/20 split).
2. **Model Building** Use **Linear Regression** to predict the Delivery Time based on features like Distance, Traffic\_Conditions, and Order\_Priority.
3. **Evaluation Metrics** Evaluate the model using:  
   * **Mean Squared Error (MSE)**
   * **R-squared (R²)**
   * **Mean Absolute Error (MAE)**

**Step 5 - Logistic Regression Model (for Categorization)**

1. **Model Objective** Classify deliveries as "Fast" or "Delayed" based on binary features such as Traffic, Weather, Delivery\_Person\_Experience, etc.
2. **Model Implementation** Use **Logistic Regression** to predict the delivery status.
3. **Evaluation Metrics** Evaluate using metrics such as:  
   * **Accuracy**
   * **Precision**
   * **Recall**
   * **F1-score**
   * **Confusion Matrix**

#### **Phase 3: Reporting and Insights**

**Step 6 - Model Evaluation and Comparison**

* Compare the **Linear Regression** and **Logistic Regression** models based on their performance (e.g., accuracy, confusion matrix).
* Visualize the results using **confusion matrices** and **ROC curves**.

**Step 7 - Actionable Insights**

* Based on model predictions, suggest operational improvements such as:
  + Optimizing delivery routes.
  + Adjusting staffing during high-traffic periods.
  + Providing better training to delivery staff

### **Final Deliverables**

1. **Jupyter Notebook (.ipynb):**
   * Complete code for data preprocessing, model training, and evaluation.
2. **Data Visualizations:**
   * Visual representations such as **scatter plots**, **pair plots**, **confusion matrices**, and **ROC curves** to interpret the results.
3. **Final Report:**
   * A detailed summary of the project, including:
     + Description of the dataset and preprocessing steps.
     + Model evaluation and comparisons.
     + Actionable insights and recommendations for optimization.