# Traffic Prediction Second Draft

**1. Data Preparation**

* **Data Cleaning**: Ensure all data is complete, handle any missing or erroneous values, and remove duplicates.
* **Feature Engineering**:
  + Transform categorical variables like weather, holidays, and road names into numerical representations.
  + Normalize/standardize numerical features like traffic volume, pedestrian count, average speed, etc.
* **Time Series Transformation**: If your data is aggregated at the daily level, consider creating hourly time-series data to capture short-term patterns.
* **Adjacency Matrix (Optional)**: If using Graph Neural Networks (GNNs), create an adjacency matrix for the road network to capture spatial dependencies.
* **Label Creation**: Decide on the output variable (e.g., traffic volume) and prepare the dataset accordingly for prediction.

**2. Exploratory Data Analysis (EDA)**

* Analyze patterns in traffic volume over time (daily, weekly, and seasonal trends).
* Visualize the relationship between traffic and key features like weather, holidays, public transport usage, etc.
* Understand feature correlations to identify the most important predictors.

**3. Model Selection**

* **Multimodal Learning Model**:
  + Select a model that can handle multiple data modalities, such as weather, incidents, and spatial/temporal data. For example:
    - **LSTM/GRU**: For temporal dependencies.
    - **Graph Neural Networks (GNN)**: If spatial dependencies (roads) are crucial.
    - **Multimodal Neural Networks**: Combine features from different data sources (weather, holidays, etc.) in a unified model.
* If necessary, perform model benchmarking to compare multiple approaches.

**4. Model Training**

* Split the dataset into training, validation, and testing sets.
* Train the model using appropriate evaluation metrics like MAE, MSE, or RMSE for regression tasks.
* Tune hyperparameters to optimize model performance.

**5. Model Evaluation**

* Evaluate the model on the test set to check its generalization ability.
* Visualize model predictions versus actual traffic data.
* Analyze model performance by feature importance to understand what’s driving predictions.

**6. User Input Interface**

* **Input Handling**: Allow users to input dates, locations (roads), and other factors (e.g., weather).
* **Prediction Output**: The model should return traffic volume and related insights based on the user's input.
* Create a user-friendly interface using Flask or another web framework.

**7. Deployment**

* Deploy the model to a cloud platform or local server.
* Ensure the prediction system is fast enough to handle real-time inputs.

**8. Monitoring & Feedback**

* Monitor the model's performance in production.
* Collect user feedback to further refine predictions or improve features like weather data or event handling.

**Next Steps:**

1. Start a new session to focus on **data preparation**.
2. Move step-by-step through feature engineering, cleaning, and data transformation.
3. Proceed to modeling and finally to the deployment phase.