AI for Surge Pricing Prediction in Ride-Sharing

Business Goal

The primary objective is to use AI to predict surge pricing in different areas of the city. This allows the company to - better balance **driver supply with rider demand**. - Improve **driver allocation and incentives**. - enhance **customer satisfaction** through fair pricing and availability. - Optimize **revenue and operational efficiency** in real-time.

Problem Classification

This is a **Regression Problem -** the output variable (surge price multiplier, e.g., 1.0x, 1.5x, 2.0x) is **Continuous** not categorical. - The goal is to **predict a numeric value** representing surge pricing for a specific time and location.

Data Requirements

To make accurate predictions, the company must collect **multimodal, real-time, and historical data** from various sources.

- **Demand & Supply** Data Number of ride requests per area per time slot, Number of available drivers in each zone, Ratio of demand and supply.
- **Temporal Data** Time of the day (peak, off-peak), Day of week (weekday vs weekend), Public holidays or special events.
- **Geospatial Data** Area /zonal identifiers (latitude, longitude, neighbourhood), Traffic congestion levels, Proximity to hot spots (airports, malls, offices).
- **Weather Data** Rain, temperature, and extreme conditions impact both demand and traffic.
- **Pricing History** Historical surge multipliers by zone and time, Base fares and promotions running.

Key Preprocessing Steps

Data Cleaning:

- Handle **missing values** in weather, traffic, or GPS data.
- Remove **outliers** such as unrealistically high or low surge multiplier.
- Ensure **timestamp synchronization** between different data streams (e.g., rides, traffic, weather)

Feature Engineering:

- Convert timestamp into useful features: **hour, weekday/weekend, peak/off-peak.**
- Create **demand-supply ratio** as a derived feature.

- Encode geospatial zones using **clustering (e.g., K-means)** to create zone IDs, Normalize or scale continuous variables (e.g., ride demand)

Modelling Approach

Start with **baseline regression model**, then move to more advanced models:

- **Baseline Models:** - Linear Regression, Decision Tree Regression
- **Advance Models:** - Random Forest Regression, Boosting (XGBoost, Gradient Boost, LightGBM)

Optional: Use **spatio-temporal models** or **deep learning (LSTM, TCNs)** for advanced use cases

Evaluation Metrics

The goal is to minimize prediction error and ensure business usefulness.

- **MAE (Mean Absolute Error):** Measures average magnitude of error
- **RMSE (Root Mean Squared Error):** Penalizes larger errors more
- **R^2 Score:** Explains variance captured by the model
- **Business Metric:** % improvement in **driver availability during peak** hours

Deployment

Once trained and validated, the model can be deployed into the mobile app ecosystem as follows:

- Integrated into the **pricing engine backend**
- For every new ride request, the model receives inputs (location, time, weather, etc.)
- Predicts the **surge multiplier** in real-time
- App updates the **fare shown to the user and driver** accordingly
- Feedback loop: store actual vs. predicted prices for continuous learning

Continuous Learning

- Monitor prediction drift due to changing city dynamics
- Periodically retrain using new data to adapt to seasonal trends, traffic patterns, and urban development