**Forex Market Regime Prediction System**

Software Design & Flow Document

**1. Overview**

This system classifies whether a given 4-hour Forex timeframe is up trend, down trend or ranging and volatility level using technical indicators and a machine learning model LightGBM. It can be extended to real-time trading signals or batch analysis.

**2. System Architecture**

Modules:

- `extractor.py`: Fetches historical OHLCV data using MetaTrader5 API.

- `unify.py`: Concatenates all historical OHLCV data fetched using MetaTrader5 API.

- `eda\_1.py`: Univariate analysis.

- `eda\_2.py`: Advance analysis.

- `feature\_enginering.py`: Calculates more features from OHLCV data.

- `feature\_selection.py`: selects the most useful features out of the calculated features

- `market\_direction\_label.py`: Applies logic to label each row as down trend(-1) ,ranging (0), uptrend(1).

- `volatility.py`: Applies logic to label each row as low volatility(0) ,medium volatility (0), high volatility(1).

- `train\_model\_1.py`: Trains LightGBM model using final\_trend\_direction.csv and evaluates it using F1-score and ROC AUC

- `train\_model\_2.py`: Trains LightGBM model using final\_vol.csv and evaluates it using F1-score and ROC AUC

- `predict.py`: Loads the model and predicts trend/range on new incoming data

- `api.py`: REST API for model inference using Flask or FastAPI

**3. Data Flow**

1. Raw Data Source

- MetaTrader5 API pulls 4-hour OHLCV data

- Saved as CSV for offline access

2. Preprocessing & Feature Generation

- Data sorted, cleaned, and missing values removed

- Technical indicators computed (20+ features)

- Normalization not required for LightGBM

3. Labeling

- -1, 0, +1 for downtrend ranging and uptrend

- 0, 1, 2 for low volatility, medium volatility or high volatility

- Label is 0 (ranging) otherwise

4. Training & Evaluation

- Dataset split using time-based method

- Model trained on features, evaluated with F1 and ROC AUC

- Model saved as `xgb\_model.pkl` or `lgb\_model.pkl`

5. Prediction

- New data fed through the same preprocessing pipeline

- Model predicts label: 0 (range), 1 (trend)

- Can be served through an API or CLI

**4. Technologies**

- \*\*Python 3.9+\*\*

- `MetaTrader5` (market data)

- `pandas`, `numpy`, `ta`, `sklearn`

- `lightgbm`, `xgboost`

- `flask` or `fastapi` for deployment

**5. File Structure**

forex\_trend\_classifier/

1. data/ all data required
2. data pre processing/ all scripts for processing data
3. doc/ all project documents
4. EDA/ all eda scripts
5. features/ feature generation scripts
6. labeling/ all labeling scripts
7. models/ all scripts for building the two models
8. visualization/ all the project generated charts

**6. Future Enhancements**

- Add LSTM model to capture sequence patterns

- Stream live data for real-time prediction

- Add signal confidence scoring

- Create dashboard for visual monitoring