ETH Implied Volatility Prediction

1. Technical Implementation

The solution applies **proper handling of time series data** by ensuring temporal order is strictly preserved in both training and validation phases.
Key steps include:

- **Feature Engineering:** Extraction of lag-based, rolling window statistics and volatility measures from raw market microstructure data.
- **Validation Strategy:** Use of **time-based cross-validation** (walk-forward split) to simulate real-world scenarios and prevent data leakage.
- **Modeling:** Gradient boosting (LightGBM) with optimized hyperparameters to capture nonlinear relationships in market features.
- **Prediction Post-Processing:** Ensured submission format aligns with competition requirements (timestamp starting from 1, correct ID handling).

---

2. Business Understanding

The project is designed to predict **short-term market volatility** using order book and trade data, simulating high-frequency trading environments.

- **Market Microstructure Insights:** Features reflect liquidity depth, price impact, bid-ask spread dynamics, and short-term imbalance signals.
- **Volatility Focus:** Model aims to anticipate rapid fluctuations, enabling informed trading and risk management decisions.
- **Practical Relevance:** Such models can be integrated into execution algorithms to reduce slippage and optimize order placement.

---

3. Documentation and Presentation

The methodology and results are **clearly documented** within the Kaggle notebook:

- Step-by-step workflow with explanatory markdown cells.
- Visualization of **OOF (Out-of-Fold) predictions** for sanity checks.
- Code comments explaining implementation decisions.
- Submission file validation to avoid format errors.

---

4. Platform Proficiency

The solution demonstrates **effective use of Kaggle's environment**:

- GPU/CPU runtime selection for efficient computation.
- `/kaggle/input` for reading competition datasets and `/kaggle/working` for outputting submission files.
- Use of Kaggle's built-in plotting tools (Matplotlib, Seaborn) for quick data exploration and result verification.
- Compliance with competition rules and formatting requirements.

Final Outcome:
A robust, reproducible pipeline capable of high leaderboard performance while maintaining business relevance and clarity in presentation.