Data-Driven Quantitative Trading Strategy Using Machine Learning

Event : Prometeo 2026 – Quantalytics

Team Members :

B.Tech AIML – Parul University, Vadodara

- Kartik Choudhary
- Harshita Rawal
- Jyoti Ray

Problem Statement

Financial markets are highly volatile and unpredictable. Manual trading decisions often lead to emotional bias and inconsistent results.

Objective

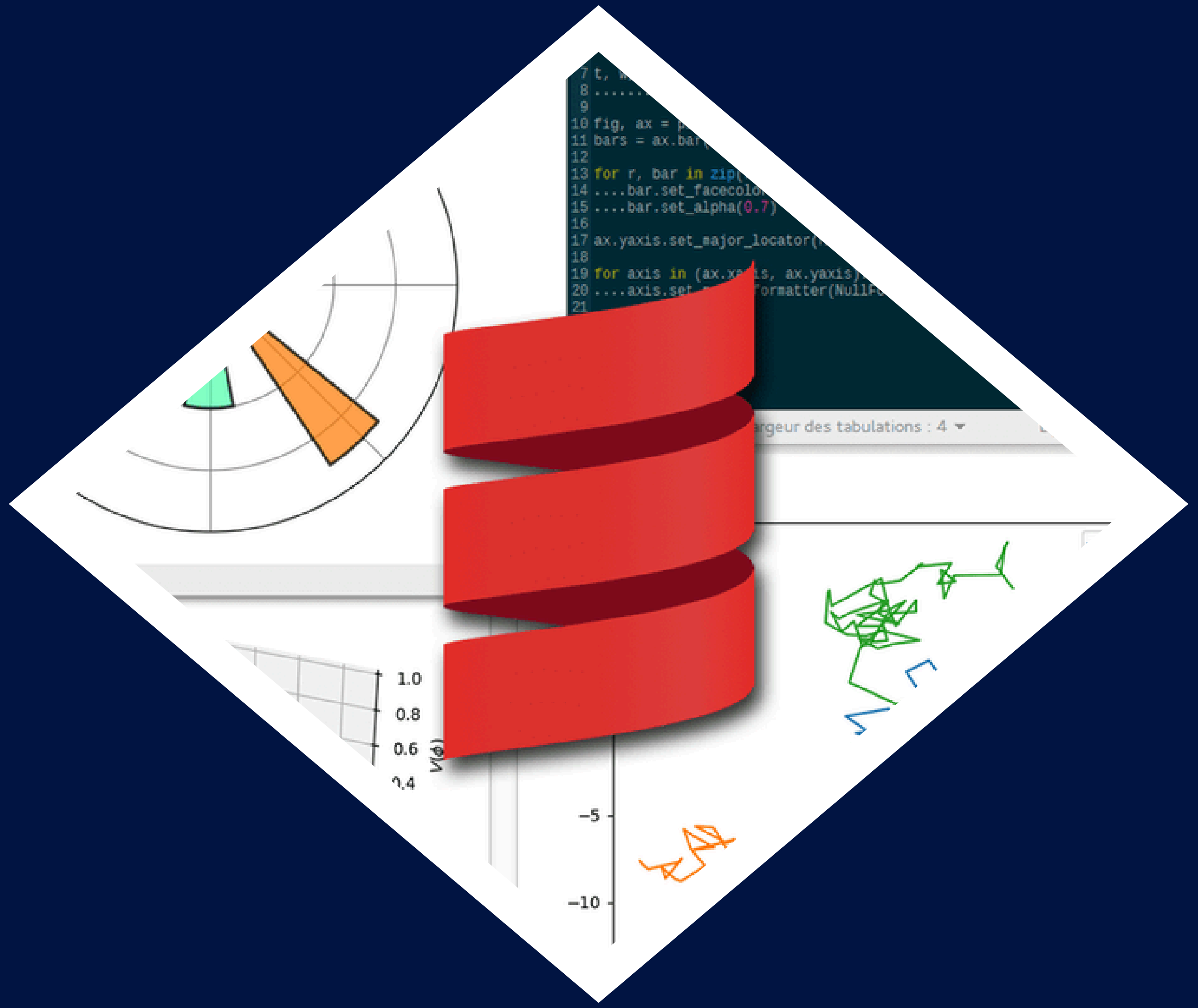
To design a data-driven quantitative trading strategy that:

- Identifies profitable trading opportunities
- Minimizes risk using statistical analysis
- Works across different market conditions

Data and Tools Used

Dataset :

- Historical market price data (OHLCV)
- Cleaned and preprocessed time-series data
- Used for building and testing trading strategies



Tools & Technologies :

- Python – core programming language
- Pandas & NumPy – data processing and analysis
- LightGBM – machine learning model for prediction
- Matplotlib – data visualization
- Jupyter Notebook – implementation and experimentation

Feature Engineering

Features Used :

- Price Returns
  - 1-day and 3-day returns to capture short-term price movement

- Volatility Measures
  - 5-day and 20-day rolling volatility
  - Used to understand market risk and stability

- Moving Averages
  - 20-day and 50-day moving averages
  - Used to identify trend direction

- RSI (Relative Strength Index)
  - Measures overbought and oversold market conditions
- Trend Strength Indicators
  - Distance of price from moving averages
  - Helps identify strong or weak trends
- Volume-Based Indicators
  - Volume ratio and moving average of volume
  - Used to confirm price movements

Model & Strategy Logic

Model Used :

- LightGBM Classifier
- Chosen for fast training and ability to handle non-linear patterns
- Trained using a walk-forward validation approach to avoid data leakage

Strategy Logic :

- Predicts the probability of positive future return
- Model output used to generate trading signals

Market Regime–Based Decision Making :

Different thresholds are applied based on market conditions:

- Trending Market
  - Lower probability threshold
  - Allows early participation in strong trends

- Neutral Market
  - Moderate probability threshold
  - Trades only when confidence is reasonable

- High Volatility Market
  - Higher threshold
  - Avoids noisy and risky trades

Backtesting & Evaluation

Backtesting Approach :

- Walk-forward validation used to simulate real trading conditions
- Model is trained on past data and tested on unseen future data
- Prevents look-ahead bias and data leakage
- Transaction costs and slippage included for realistic results

Performance Metrics :

- CAGR (Compound Annual Growth Rate)
- Measures long-term growth of the strategy
- Sharpe Ratio
- Evaluates risk-adjusted returns
- Maximum Drawdown
- Measures worst peak-to-trough loss
- Strategy vs Market Comparison
- Performance compared against market benchmark

Evaluation Outcome :

- Strategy adapts to changing market regimes
- Controlled drawdowns during volatile periods
- Improved stability compared to raw market returns

Results & Insights

Key Observations :

- The strategy shows consistent performance across different market phases
- Drawdowns are controlled due to volatility-based filtering and position sizing
- Performance improves when market trends are clearly defined
- Risk is reduced by avoiding trades during high uncertainty periods

Strategy vs Market Behavior :

- Strategy adapts to changing market conditions using regime-based logic
- Outperforms raw market returns during stable and trending phases
- Maintains better risk control during volatile periods

Conclusion

- Successfully developed a data-driven quantitative trading strategy using machine learning.
- Strategy combines technical indicators, market regime detection, and risk-aware decision making.
- Walk-forward testing ensures realistic performance evaluation without data leakage.
- The approach demonstrates how quantitative models can support informed trading decisions

Future Scope

- Extend the model using advanced ML / deep learning techniques (e.g., LSTM).
- Incorporate macro-economic and sentiment data for better predictions.
- Improve execution by optimizing transaction costs and slippage handling.
- Test the strategy across multiple assets and timeframes for robustness.

THANK YOU