STATFLOW STRATEGY

PRESENTED BY: KPO MEMBERS

DOMAIN 2: QUANTITATIVE TRADING

Problem Statement

Problem 1: Traditional Momentum Strategies Are Easily Fooled by Market Noise

Problem 2: High Volatility Periods Cause Unpredictable Losses

Problem 3: Momentum Signals Are Prone to False Breakouts


Euro / U.S. Dollar, 1D, IDC
EMA (200, close)
Negative Momentum
1.21000
1.20000
1.19000
1.18000
1.17000
1.16000
1.15000
1.14000
1.13000
1.12000
1.11000
1.10239
1.09000
Jul
Sep
Nov
2019
Mar
May
Jul
Sep

What is the **StatFlow Strategy?**

Volatility-Aware Momentum (VAM) is a trading strategy that combines momentum signals with a volatility filter. It enters trades only when the price is trending strongly and market volatility is within a safe range this help to reduce false signals and manage risk effectively.


BTC vs Strategy Returns
Sharpe Ratio: 0.9334
BTC Returns
Strategy Returns
Cumulative Return
2.6
2.4
2.2
2.0
1.8
1.6
1.4
1.2
1.0
2024-01
2024-03
2024-05
2024-07
2024-09
2024-11
2025-01
Date

How the **StatFlow Strategy Works**

Buy Signal Formula:

$$\text{If} \quad \text{Momentum}_t > M_{threshold} \quad \text{AND} \quad \text{Volatility}_t < V_{threshold}, \quad \text{then BUY}$$

Sell Signal Formula:

$$\text{If} \quad \text{Momentum}_t < -M_{threshold} \quad \text{AND} \quad \text{Volatility}_t < V_{threshold}, \quad \text{then SELL}$$

How the **StatFlow Strategy Works**

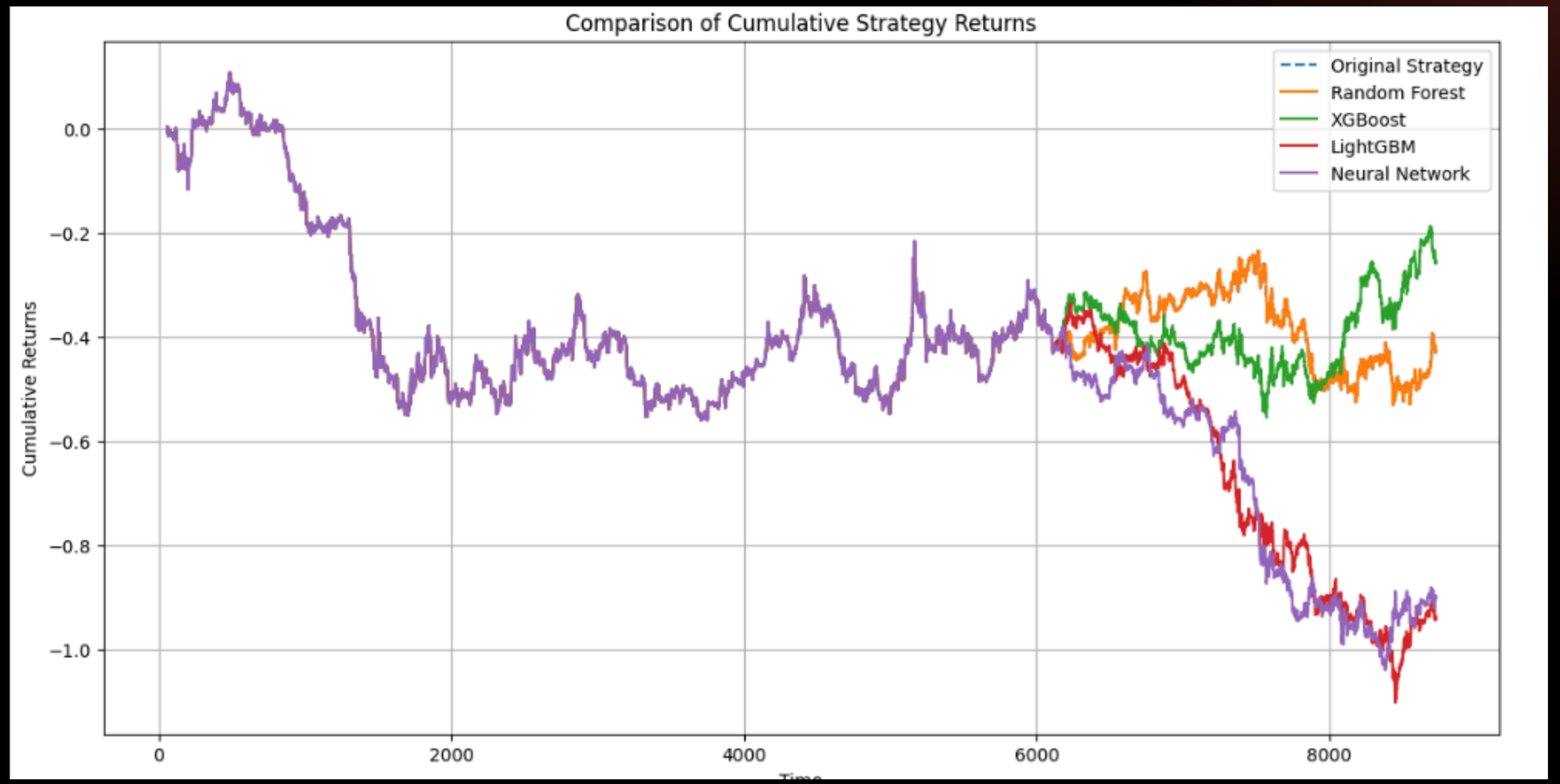
```
threshold = merged_df['combo_signal'].quantile(0.80)
```

Use a dynamic cutoff — only trade the top 20% strongest signals:,it reacts only to high-confidence setups.

Business Value

- combines momentum and volatility filters to avoid false signals
- Proactively avoids trades during high volatility, reducing drawdowns
- More Predictable Returns: Reduces surprises from sudden market swings.
- Scalable Automation: Easily expanded to multiple assets or strategies.

Machine Learning




```
RandomForestClassifier Accuracy: 0.5186753946861764
XGBoost Accuracy: 0.5097963887821744
LightGBM Accuracy: 0.4978870533999232
Neural Network Accuracy: 0.4802151363810987 5
```

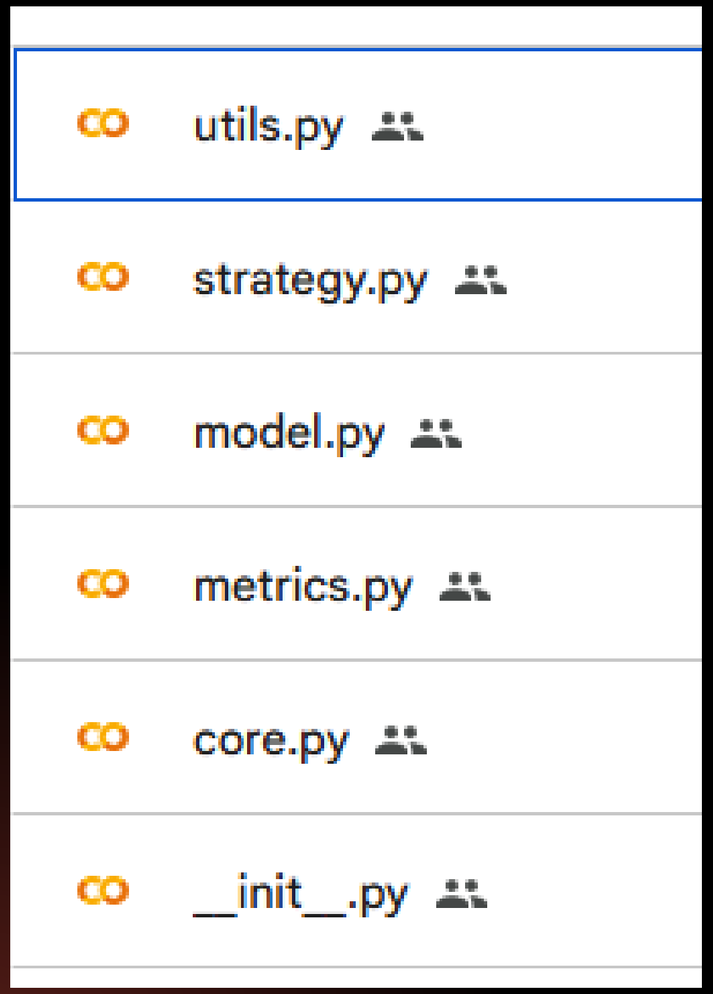
Choosen Model

LSTM Accuracy: 0.5196153846153846

What Our Model do?

- Learns from past 10 time steps of flow + momentum features.
- Predicts whether BTC price will rise next time step.
- Generates binary trading signals.
- Simulates a basic long-short strategy.
- Plots cumulative P&L to evaluate performance.

Architectural Design of Backtest Library



1. Core.py—This is likely where the main engine lives — stuff like data handling, orchestration of strategy logic, or managing the pipeline (e.g., loading prices → calculating metrics → deciding trades).

2. metrics.py - it's the calculator. It computes things like momentum, volatility, returns, maybe drawdowns. Used to filter stocks or generate signals.

3. model.py - define a machine learning model, or a rule-based decision system. Could be what actually makes the buy/sell call after seeing metrics.

4. strategy.py - "If momentum > X and volatility < Y → buy stock"

5. utils.py - Smoothing data, Converting dates, Normalizing values

6. __init__.py - This makes your folder a Python package. It's like the "hey, I'm ready to be imported" file. Might also expose key functions or classes from the library.


THANK
YOU