

Dự án Module 6 Tuần 4 - LTSF-Linear Forecasting Challenge.

Time-Series Team

Ngày 7 tháng 12 năm 2025

Mục lục

PHẦN 1: KHỞI ĐỘNG (Foundation)	3
1 Đặt vấn đề	3
2 Dataset	6
2.1 Cấu trúc dữ liệu	6
2.2 Những hạn chế và thiếu hụt của Dataset	6
3 Tư duy thiết kế Feature Engineering: Top-down thay vì Bottom-up	7
3.1 Hai chiều tư duy trong Feature Engineering	7
3.2 Xuất phát từ kiến trúc mô hình, không phải từ feature	9
4 Từ các giả thuyết thị trường đến các nhóm Feature	13
4.1 Giả thuyết 1: Giá = Trend dài hạn + Dao động ngắn hạn (<i>Trend & Mean-Reversion</i>)	14
4.2 Giả thuyết 2: Sai số lớn luôn đi kèm Volatility & Shock (<i>Volatility & Risk-awareness</i>)	17
4.3 Giả thuyết 3: Volume phản ánh sức mạnh thực sự của một cú move (<i>Volume & Order-flow</i>)	19
4.4 Giả thuyết 4: Mỗi cây nến là một bản tóm tắt tâm lý thị trường (<i>Candlestick Micro-structure</i>)	21
4.5 Giả thuyết 5: Thị trường có trí nhớ ngắn hạn (<i>Momentum & Streaks</i>)	25
5 EDA	27
5.1 Vai trò của EDA	27
5.2 Từ giả thuyết → Feature → Kiểm chứng	28
PHẦN 2: TRÁI TIM CỦA MÔ HÌNH (The Core Engine)	29
1 Kiến trúc Pipeline 3 lớp	29
1.1 Lớp 1: Math Backbone (Trend)	29
1.2 Lớp 2: ML Residual	34
1.3 Lớp 3: Pricing Layer	41
2 Deep Dive vào Pricing Layer	43
2.1 Cơ chế vật lý	43
2.2 Regime-aware Pricing	48
2.3 Optimization Strategy	54
PHẦN 3: HỘI ĐỒNG CHUYÊN GIA (Ensemble Strategy)	59
1 Định nghĩa các Chuyên gia (Experts)	59
1.1 Dynamic Experts (Mô hình Động)	60
1.2 Static Experts (Mô hình Tĩnh)	62
1.3 Risk Experts (Mô hình Rủi ro)	65

2 Cơ chế Ensemble	68
2.1 Weighted Ensemble	68
2.2 Lý giải vai trò từng Expert	70
PHẦN 4: KẾT QUẢ & ĐÁNH GIÁ (Evaluation)	71
1 Kết quả dự báo	71
1.1 Thiết lập thí nghiệm dự báo 100 ngày	71
1.2 Bức tranh trực quan: Hybrid + Pricing + Trend + Uncertainty	71
1.3 Diễn giải kết quả cho người dùng cuối	73
2 Đánh giá độ tin cậy	73
2.1 Cross-validation theo thời gian: mô hình có bền vững qua nhiều pha thị trường?	73
2.2 Pricing Layer được tối ưu bằng Random Search + CV	74
2.3 Regime hiện tại: SIDEWAYS sau điều chỉnh	75
2.4 Dải bất định & quản trị rủi ro	75
2.5 Kết luận ngắn cho phần độ tin cậy	76
PHẦN 5: Demo và hướng dẫn sử dụng Demo	77
1 Landing Page & Data Visualization	77
2 100-Day Price Forecast (Tính Năng Cốt Lõi)	77
3 Custom 100-Day Price Forecast with Adjustable Parameters	80
4 UI Bổ trợ	82

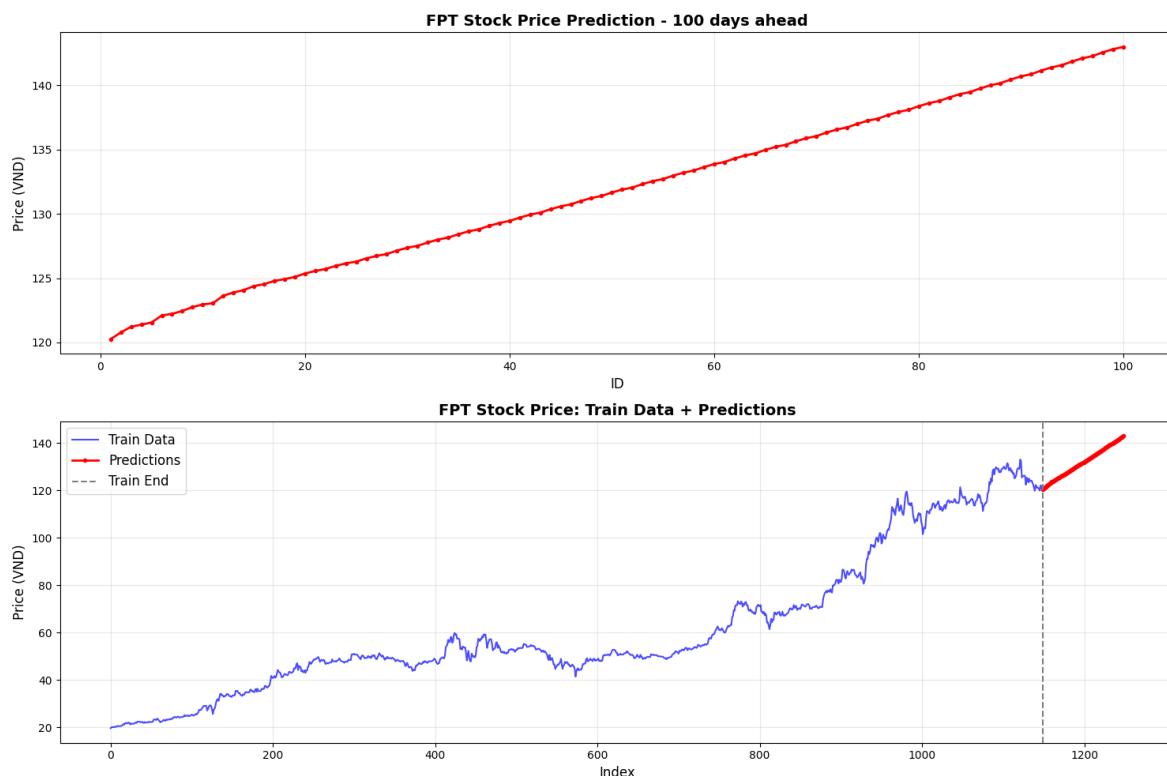
PHẦN 1: KHỞI ĐỘNG (Foundation)

1 Đặt vấn đề

1.1. Bức tường 100 ngày và *Cái chết của Phương sai*

Project 6.1 đặt ra bài toán nghe qua khá đơn giản: dự báo giá đóng cửa cổ phiếu FPT cho **100 ngày giao dịch tiếp theo** ($T+100$), chỉ sử dụng dữ liệu trong file `FPT_train.csv`. Trong ngắn hạn ($T+5$), các mô hình time-series kinh điển như ARIMA hay các biến thể Linear hiện đại (NLinear, DLinear) thường vẫn bám được xu hướng tương đối tốt.

Tuy nhiên, khi mở rộng sang **dự báo dài hạn**, đặc biệt là theo chiến lược *forecast cuốn chiếu* (recursive forecasting), baseline ban đầu của nhóm mình bộc lộ một vấn đề rất rõ: **đường dự báo 100 ngày gần như trở thành một đường thẳng tắp, không còn dao động** (hay hiện tượng "Cái chết của phương sai"). Hình 1 minh họa hiện tượng này:



Hình 1: *

Baseline Linear/NLinear: dự báo 100 ngày gần như trở thành một đường thẳng mượt, trái ngược với biến động mạnh của dữ liệu lịch sử.

Thoạt nhìn, rất dễ đổ lỗi cho **recursive forecasting** (dự báo cuốn chiếu) hay cho **MSE** (vì thường kéo dự báo về trung bình). Nhưng nếu dừng lại ở đó thì chưa đúng bản chất. Vấn đề thực sự nằm ở **cách thiết kế baseline và cách xử lý dữ liệu đầu vào**, cụ thể:

1. Bài toán mà baseline đang giải:

- Dữ liệu đầu vào là chuỗi log-price:

$$\text{close_log} = \log(\text{close})$$

- Mỗi mẫu huấn luyện lấy **14 ngày** làm input và **3 ngày** tiếp theo làm output. Bộ sinh dữ liệu được cài đặt như sau:
 - 14 giá log gần nhất → dự báo tiếp 3 giá log phía trước.
- Dự báo 100 ngày được tạo ra bằng cách lặp lại chiến lược này nhiều lần: sử dụng 14 ngày cuối (trong đó có cả dự báo trước đó) để dự đoán 3 ngày tiếp theo, cho đến khi đủ 100 ngày.

2. Kiến trúc mô hình: một tầng Linear trên chuỗi log-price

- Baseline sử dụng mô hình **NLinear** (nguyên bản sử dụng Linear thì kết quả còn tệ hơn trên public test Kaggle) với một tầng **nn.Linear** duy nhất, ánh xạ từ 14 giá trị log-price sang 3 giá trị log-price tiếp theo.
- Không có thành phần phi tuyến (nonlinearity), không có cơ chế nắm bắt cấu trúc thời gian phức tạp (như kernel, attention, hay recurrent).
- Về bản chất, mô hình chỉ học được một **hàm tuyến tính** trên đoạn log-price 14 ngày.

3. Bước chuẩn hoá của NLinear vô tình làm phẳng dao động:

- Trước khi đi vào tầng Linear, chuỗi đầu vào được chuẩn hoá theo:

$$x_{\text{norm}} = x - x_{\text{last}},$$

tức là toàn bộ 14 ngày được trừ đi giá trị ngày cuối cùng, rồi sau khi dự báo xong mới cộng lại.

- Bước này giúp mô hình ổn định hơn khi phân phối giá thay đổi, nhưng đồng thời cũng **giảm mạnh biên độ dao động tương đối** trong đoạn 14 ngày – đặc biệt khi log-price vốn đã khá mượt.

4. Dữ liệu log-price của FPT gần như tuyến tính theo thời gian:

- FPT có xu hướng tăng trưởng theo thời gian, gần giống một hàm mũ theo ngày giao dịch.
- Khi lấy log, chuỗi giá trở thành một đường có **trend gần tuyến tính**, tức là:

$$\log(FPT(t)) \approx at + b.$$

- Trong bối cảnh như vậy, một tầng Linear được huấn luyện bằng MSE trên rất nhiều cửa sổ 14→3 gần nhau sẽ tự nhiên học ra **một slope trung bình** cho log-price, chứ không học được volatility.

5. Tại sao recursive làm mọi thứ “thẳng tắp” hơn?

- Bản thân recursive forecasting không phải là thủ phạm giết phương sai – rất nhiều mô hình AR, RNN, LSTM vẫn dùng recursive bình thường.
- Nhưng trong baseline này, mỗi lần dự báo:
 - mô hình dùng **input đã bị làm mượt** (log + chuẩn hoá),
 - áp dụng **một hàm tuyến tính** với slope gần như cố định.

- Khi ta liên tục feed lại các dự báo mượt đó vào vòng sau, **mọi nhiễu nhỏ còn sót lại bị “là phẳng” dần**, và quỹ đạo dự báo hội tụ thành:

$$\hat{y}_{t+k} \approx \hat{y}_t + k \cdot \Delta,$$

tức là một đường thẳng với độ dốc gần như cố định.

Nói cách khác, hiện tượng **“Cái chết của phương sai”** ở đây không phải là một quy luật tự nhiên của recursive forecasting hay của MSE, mà là hệ quả tất yếu khi kết hợp:

- **Log transform** làm mượt chuỗi giá,
- **Chuẩn hoá giá trị cuối** trong NLinear làm phẳng thêm dao động,
- **Mô hình tuyến tính đơn tầng** chỉ học được trend tuyến tính,
- **Dự báo cuốn chiếu nhiều bước** khuếch đại tính tuyến tính này qua thời gian.

Kết quả cuối cùng chính là những gì ta thấy ở Hình 1: baseline có thể giữ được hướng tăng, nhưng **mất hoàn toàn đặc trưng volatility** của cổ phiếu FPT. Đây là lý do nhóm mình cần một pipeline mới, nơi phần *trend* vẫn được mô hình hoá bằng một backbone toán đơn giản, nhưng phần *dao động* và *hành vi thị trường* phải được xử lý bởi một lớp mô hình khác giàu năng lực hơn.

1.2. Vì sao Deep Learning không phải là *Chén Thánh?*

Nghe có vẻ hợp lý khi nâng cấp lên LSTM hoặc Transformer. Tuy nhiên, với dữ liệu chứng khoán Việt Nam (FPT là ví dụ điển hình), DL lại vấp phải hai rào cản lớn:

1. **Dữ liệu hạn chế:** vài nghìn điểm dữ liệu là quá ít. DL cực kỳ dễ overfit, học vẹt nhiễu thay vì quy luật.
2. **Thiếu cơ chế “tự sửa lỗi”:** DL thuần tuý dựa trên dữ liệu. Khi thị trường chuyển chế độ (từ Bull sang Bear), mô hình có xu hướng **ngoại suy mù quáng** dựa trên xu hướng cũ.

Do đó, DL không phải lời giải tối ưu cho bài toán dự báo dài hạn với dữ liệu hạn chế như FPT.

1.3. Lời giải Hybrid: Khi Machine Learning gặp Financial Engineering

Thay vì cố xây một mô hình **siêu AI** làm tất cả, ta áp dụng triết lý **Decomposition – chia để trị**, để mỗi thành phần mô hình làm đúng thứ nó giỏi nhất.

1. **Math Backbone (Trend dài hạn):** Dùng Linear trên log-price để giữ hướng chính xác của quỹ đạo dài hạn, tránh dự báo trôi dạt vô lý.
2. **XGBoost Residual (Dao động ngắn hạn):** Linear không thể học hết nhiễu động. XGB học residual giúp:
 - tái tạo nhịp dao động ngắn hạn,
 - tăng realism cho đường giá,
 - ổn định hơn DL trên lượng dữ liệu nhỏ.
3. **Pricing Layer (Lớp kiểm soát hành vi thị trường):** Đây là phần nâng cấp quan trọng nhất của pipeline.

- **Mean Reversion:** giá sẽ có lực hút về vùng giá trị thực.
- **Regime Detection:** nhận diện Bull/Bear để điều chỉnh biên độ dao động.

Tinh thần của pipeline: Mục tiêu không phải dự đoán chính xác từng ngày (điều bất khả thi), mà là xây dựng một **trajectory giá hợp lý và vững** – kết hợp quan tính toán học và tâm lý thị trường.

2 Dataset

Dữ liệu được sử dụng trong dự án là tập tin `FPT_train.csv`, chứa thông tin giao dịch lịch sử của cổ phiếu FPT. Đây là dữ liệu chuỗi thời gian dạng bảng (Tabular Time-series) điển hình trong tài chính.

2.1 Cấu trúc dữ liệu

Feature	Mô tả
<code>Open</code>	Giá mở cửa phiên giao dịch.
<code>High</code>	Mức giá cao nhất đạt được trong ngày (thể hiện áp lực mua/hưng phấn).
<code>Low</code>	Mức giá thấp nhất trong ngày (thể hiện áp lực bán/sợ hãi).
<code>Close</code>	Giá đóng cửa (thường được dùng làm target chính để dự báo).
<code>Volume</code>	Khối lượng giao dịch (động lực/nhiên liệu của xu hướng giá).

Bảng 1: Mô tả các biến trong tập dữ liệu FPT_train

Tập dữ liệu bao gồm các trường thông tin cơ bản (OHLCV) mô tả hành vi giao dịch theo ngày: Đặc trưng của cổ phiếu FPT là một mã cổ phiếu công nghệ đầu ngành tại Việt Nam, có xu hướng dài hạn là tăng trưởng (Uptrend). Tuy nhiên, trong ngắn hạn và trung hạn, nó chịu sự chi phối mạnh của các chu kỳ thị trường (Market Regimes) như Bull (Tăng), Bear (Giảm) và Sideways (Đi ngang).

2.2 Những hạn chế và thiếu hụt của Dataset

Dựa trên phân tích thực tế, tập dữ liệu này tồn tại những điểm "thiếu thực tế" gây khó khăn lớn cho việc dự báo chính xác 100 ngày nếu chỉ dùng model thuần túy:

- **Thiếu thông tin ngữ nghĩa (No Semantic/News Data):** Dataset chỉ bao gồm các con số khô khan. Model hoàn toàn "mù" trước các thông tin vĩ mô hoặc tin tức doanh nghiệp (ví dụ: tin hợp tác AI, báo cáo tài chính). Như đã đề cập ở phần đặt vấn đề, việc thiếu *phân tích cảm xúc* khiến model không thể giải thích được các cú tăng/giảm đột biến do tin tức.
- **Độ dài dữ liệu so với tầm dự báo (Horizon Mismatch):** Dữ liệu huấn luyện chỉ khoảng hơn 1000 mẫu (tương đương khoảng 4.5 năm), nhưng yêu cầu dự báo xa tới 100 ngày (tương đương gần 1/4 năm giao dịch). Tỷ lệ này là quá lớn, khiến các mô hình dễ bị nhiễu và mất phương hướng ở các ngày cuối của chu kỳ dự báo.
- **Thiếu các biến phái sinh (Lack of Derived Features):** Dữ liệu thô chưa phản ánh được động lượng (Momentum) hay biến động (Volatility). Việc dự báo dựa trên dữ liệu thô mà không có Feature Engineering (như RSI, MACD, hay các tín hiệu dòng tiền) giống như việc lái xe chỉ nhìn công tơ mét mà không nhìn đường.

3 Tư duy thiết kế Feature Engineering: Top–down thay vì Bottom–up

Sau Chương 1, ta đã thấy rõ một thực tế khó chịu: chỉ dựa vào *data thuận* và các mô hình dự báo quen thuộc (Linear, NLinear, thậm chí Deep Learning) là không đủ để vượt qua *bức tường T+100*. Baseline có thể khớp khá tốt quá khứ, nhưng khi bước vào vùng dự báo dài ngày, đường giá giàn như bị *lì đòn phương sai*: dao động dần biến mất, sai số tích luỹ phình to, và mô hình chỉ còn đóng vai trò “vẽ lại một xu hướng trung bình”.

Điểm mấu chốt rút ra từ Chương 1 là: *nếu chỉ để dữ liệu tự nói, mô hình có thể học tốt những gì đã xảy ra, nhưng rất khó học được những kích bản thị trường mới trong tương lai dài hạn*. Để một dự báo T+100 còn “sống” về mặt hành vi thị trường, ta buộc phải đưa thêm **giả thuyết domain** vào pipeline: cấu trúc Hybrid (Math Backbone + XGBoost Residual + Pricing Layer), cơ chế mean-reversion, volatility clustering, và phân biệt regime Bull/Bear/Sideways.

Từ đó, một nhận định trở nên rõ ràng: *vấn đề cốt lõi không chỉ nằm ở việc chọn mô hình nào, mà nằm ở cách ta thiết kế toàn bộ pipeline học máy ngay từ đầu*.

Trong các bài toán Machine Learning thông thường, Feature Engineering thường đi theo trình tự quen thuộc:

$$\text{Dữ liệu} \rightarrow \text{EDA} \rightarrow \text{Feature} \rightarrow \text{Model} \rightarrow \text{Đánh giá}$$

Cách tiếp cận này phù hợp cho dự báo ngắn hạn hoặc nội suy. Tuy nhiên, với bài toán **dự báo 100 ngày giao dịch tiếp theo** trong điều kiện dữ liệu hạn chế và thị trường có cấu trúc phức tạp như FPT, tư duy này bộc lộ một giới hạn quan trọng: **nó không cung cấp cho mô hình một khung suy nghĩ rõ ràng về hành vi thị trường trong tương lai dài hạn**.

Do đó, Chương 2 không tiếp tục hỏi: “*EDA còn chỉ ra feature gì nữa?*” mà chuyển sang một câu hỏi mang tính thiết kế:

Nếu quay về thời điểm chưa có Feature Engineering nào, ta phải suy nghĩ như thế nào để thiết kế ra một pipeline FE phù hợp với kiến trúc Hybrid + Pricing, thay vì chỉ tối ưu ngắn hạn theo dữ liệu lịch sử?

Để trả lời câu hỏi này, trước hết ta cần phân biệt hai chiều tư duy cơ bản trong Feature Engineering.

3.1 Hai chiều tư duy trong Feature Engineering

Trong thực hành Machine Learning, Feature Engineering không chỉ là một tập hợp các thủ thuật kỹ thuật, mà phản ánh **cách ta tư duy về mối quan hệ giữa dữ liệu, mô hình và bài toán**.

Quan sát tổng quát, Feature Engineering thường vận hành theo hai chiều tư duy khác nhau: **Bottom-up** và **Top-down**.

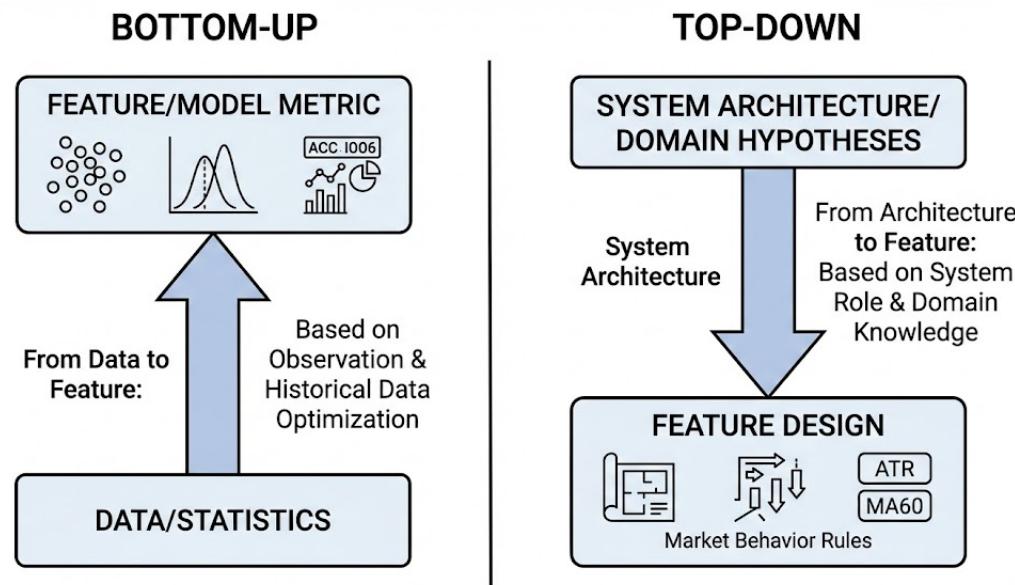
Bottom-up: Từ dữ liệu đến feature

Bottom-up là cách tiếp cận mặc định trong hầu hết pipeline Machine Learning tiêu chuẩn. Quy trình điển hình có thể được mô tả như sau:

Quan sát dữ liệu → EDA → phát hiện tín hiệu thống kê → thiết kế feature.

Trong cách tiếp cận này:

- Feature được sinh ra từ những gì dữ liệu “thể hiện” (xu hướng, tương quan, volatility, outlier).
- Mục tiêu ngầm định là cải thiện metric trên tập validation.



Hình 2: Hai chiều tư duy trong Feature Engineering

- Giả định rằng mô hình sẽ tự học các hành vi phức tạp từ dữ liệu quá khứ.

Tư duy Bottom-up đặc biệt hiệu quả trong các bài toán:

- dự báo ngắn hạn ($T+1, T+5$),
- nội suy trong vùng phân phối đã quan sát,
- dữ liệu lớn và tương đối ổn định.

Tuy nhiên, trong bài toán dự báo $T+100$, cách tiếp cận này bộc lộ một hạn chế căn bản: *EDA chỉ phản ánh quá khứ, trong khi Feature Engineering cần chuẩn bị cho những kịch bản chưa từng xuất hiện rõ ràng trong dữ liệu*. Kết quả là pipeline dễ rơi vào trạng thái tối ưu cục bộ theo lịch sử, nhưng thiếu khả năng khai quát dài hạn.

Top-down: Từ mô hình và domain đến feature

Ngược lại, Top-down đảo chiều điểm xuất phát của Feature Engineering. Thay vì bắt đầu từ dữ liệu, ta bắt đầu từ:

- **kiến trúc mô hình** dự kiến sử dụng,
- và **các quy luật domain** đã biết về thị trường tài chính.

Trong tư duy Top-down, Feature Engineering trả lời câu hỏi:

Muốn từng thành phần của mô hình hoạt động đúng vai trò, thì nó cần “nhìn thấy” những đại lượng thị trường nào?

Với kiến trúc Hybrid trong Project:

- Math Backbone cần thông tin về trend và mức cân bằng dài hạn.
- XGBoost Residual cần các đặc trưng mô tả dao động, nhiễu và phản ứng ngắn hạn.

- Pricing Layer cần các tín hiệu về volatility, regime và mức độ “quá đà” của giá.

Từ đó, feature không còn được thiết kế để “giảm MSE nhanh nhất”, mà được thiết kế để **mã hoá các giả thuyết hành vi thị trường** thành dạng mà mô hình có thể khai thác.

Lựa chọn cho Project

Dựa trên phân tích của Chương 1, Project được xây dựng ngay từ đầu theo tư duy Top-down:

Kiến trúc → Giả thuyết thị trường → Feature

EDA và các phân tích Bottom-up không bị loại bỏ, nhưng được sử dụng như một bước **kiểm chứng và tinh chỉnh**, thay vì dẫn dắt quá trình thiết kế Feature Engineering.

3.2 Xuất phát từ kiến trúc mô hình, không phải từ feature

Một nhầm lẫn phổ biến khi nói về Feature Engineering là xem nó như một tập hợp các “mẹo” rời rạc: thêm chỉ báo này, thử lag kia, giữ feature nào corr cao thì dùng. Cách tiếp cận này có thể cho kết quả chấp nhận được trong một số bài toán ngắn hạn, nhưng trở nên mong manh khi phải đối mặt với bài toán dự báo dài hạn T+100 trong bối cảnh dữ liệu nhỏ và thị trường không ổn định.

Trong Project, Feature Engineering không được thiết kế theo kiểu “bottom-up từ feature”, mà theo một logic ngược lại:

Kiến trúc mô hình → Vấn đề cần xử lý → Feature cần thiết

Nói cách khác, ta không bắt đầu bằng câu hỏi “*dữ liệu còn feature gì chưa khai thác?*”, mà bắt đầu bằng:

Với bài toán T+100, kiến trúc mô hình cần những thành phần nào để xử lý các điểm yếu cốt lõi của time-series tài chính, và mỗi thành phần đó cần “nhìn thấy” những thông tin gì?

Kiến trúc Hybrid không phải “sách giáo khoa”

Cấu trúc được sử dụng không phải là các kiến trúc kinh điển thường thấy trong tài liệu học thuật như ARIMA, LSTM hay Transformer thuần túy. Đây là một kiến trúc thuộc lớp **Quantitative Strategy thực chiến**, thường được áp dụng trong bối cảnh:

- dữ liệu không nhiều (small data),
- tài sản có xu hướng dài hạn mạnh (strong trend),
- và yêu cầu dự báo vượt xa vùng quan sát.

Trong giới chuyên môn, cách tiếp cận này thường được mô tả bằng cụm từ:

De-trending + Residual Modeling

Tức là: **tách cấu trúc chuyển động của giá thành nhiều lớp với bản chất khác nhau, rồi giải quyết từng lớp bằng công cụ phù hợp nhất**.

Kiến trúc Hybrid trong Project gồm ba thành phần chính:

- Math Backbone (mô hình hoá xu hướng dài hạn),
- XGBoost Residual (mô hình hoá dao động phi tuyến),
- Pricing Layer (kiểm soát hành vi giá theo regime và rủi ro).

Việc thiết kế Feature Engineering xuất phát trực tiếp từ vai trò của từng thành phần này.

Math Backbone: xử lý tính không dừng của giá

Đặc trưng căn bản nhất của chuỗi giá cổ phiếu là **không dừng** (non-stationarity): mức trung bình và biến độ biến động thay đổi theo thời gian. Một mô hình học máy nếu học trực tiếp trên giá thô sẽ dễ rơi vào trạng thái “ngáo giá”: những mức giá mới trong tương lai đơn giản chưa từng xuất hiện trong tập huấn luyện.

Math Backbone được đưa vào để giải quyết đúng vấn đề này:

- tách xu hướng tăng trưởng dài hạn ra khỏi chuỗi giá,
- đưa phần còn lại về một chuỗi dao động quanh 0 (stationary hơn).



Hình 3: Minh họa quá trình **de-trending** trên chuỗi giá FPT. (Trên) Giá đóng cửa ở thang log ($\log(\text{Close})$) và xu hướng dài hạn được trích xuất bằng STL. (Dưới) Phần dư (residual) sau khi loại bỏ trend, dao động quanh 0 với phân phối ổn định hơn. Việc tách trend và residual giúp biến chuỗi giá không dừng thành một chuỗi gần dừng, tạo điều kiện thuận lợi cho mô hình học máy ở bước tiếp theo.

Qua đó cho thấy rõ hiệu quả của bước khử xu hướng (de-trending): thay vì học trực tiếp trên giá không dừng, pipeline chuyển bài toán về học phần residual ổn định hơn.

Từ đây, Feature Engineering cho Math Backbone không cần phức tạp, mà tập trung vào:

- ước lượng mức độ và tốc độ của trend,
- xác định vị trí hiện tại của giá so với xu hướng dài hạn.

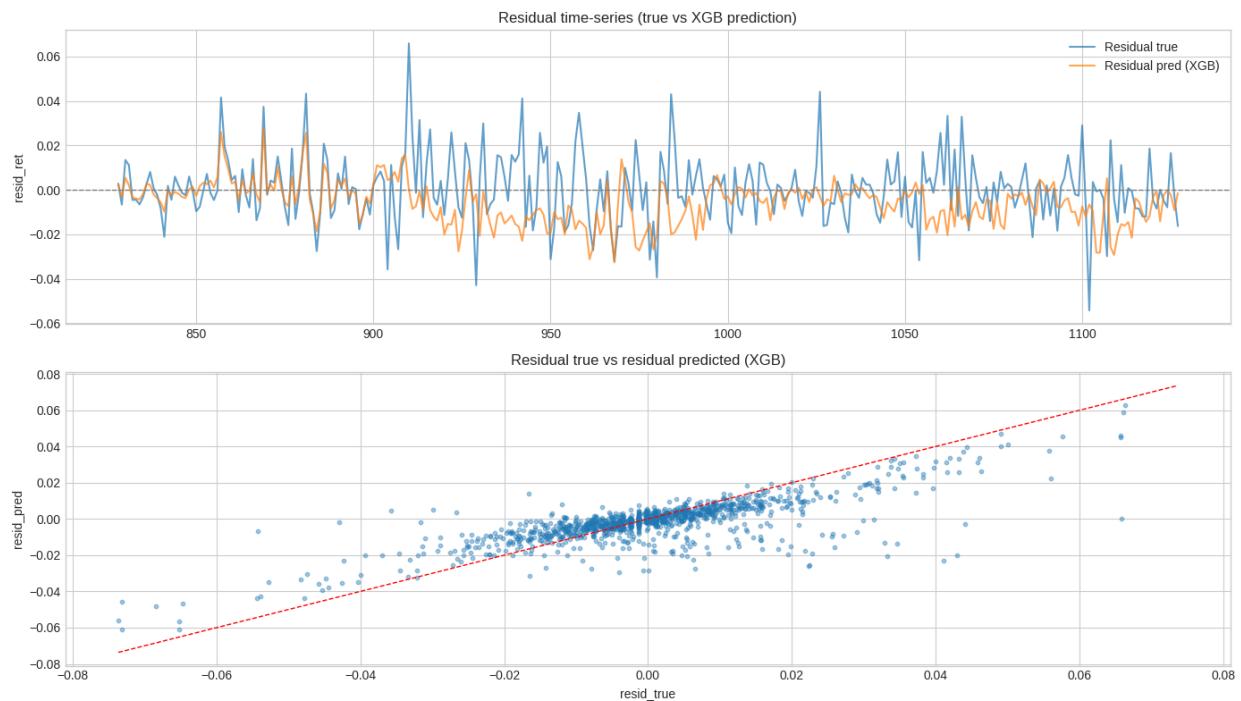
Điểm quan trọng là: *Trend không phải thứ để XGBoost học*, mà là phần xương sống cần được cố định và ổn định trước khi đưa ML vào.

XGBoost Residual: mô hình hoá phi tuyến và nhiễu có cấu trúc

Sau khi loại bỏ xu hướng, phần residual còn lại không còn là nhiễu trắng thuần tuý. Nó vẫn mang các cấu trúc phi tuyến: volatility thay đổi, phản ứng với volume, hành vi theo candlestick và momentum ngắn hạn.

Đây là vùng mà các mô hình tuyến tính thất bại, nhưng các mô hình cây quyết định như XGBoost lại tỏ ra phù hợp:

- hoạt động tốt với dữ liệu nhỏ,
- nắm bắt quan hệ phi tuyến giữa nhiều feature,
- không yêu cầu giả định phân phối chặt chẽ.



Hình 4: So sánh phần dư thực tế và phần dư được dự đoán bởi XGBoost. (Trên) Diễn biến theo thời gian của residual thật và residual dự đoán. (Dưới) Scatter plot giữa residual thật và residual dự đoán với đường chéo lý tưởng. Kết quả cho thấy XGBoost mô hình hoá được cấu trúc phi tuyến của dao động ngắn hạn, thay vì bị chi phối bởi xu hướng dài hạn đã được xử lý ở Math Backbone.

Qua đó cho thấy XGBoost không còn phải gánh nhiệm vụ học xu hướng, mà tập trung vào việc học các dao động ngắn hạn và phi tuyến trong phần residual, đúng với triết lý phân tách nhiệm vụ của kiến trúc Hybrid.

Do đó, Feature Engineering cho XGBoost Residual được thiết kế để mô tả:

- trạng thái volatility hiện tại,
- cường độ và hướng của dòng tiền,
- cấu trúc tâm lý ngắn hạn thể hiện qua candlestick và pattern.

Ở đây, feature không nhằm “đoán giá ngày mai”, mà nhằm trả lời câu hỏi: “Trong bối cảnh hiện tại, phần dao động quanh trend thường có xu hướng phản ứng như thế nào?”

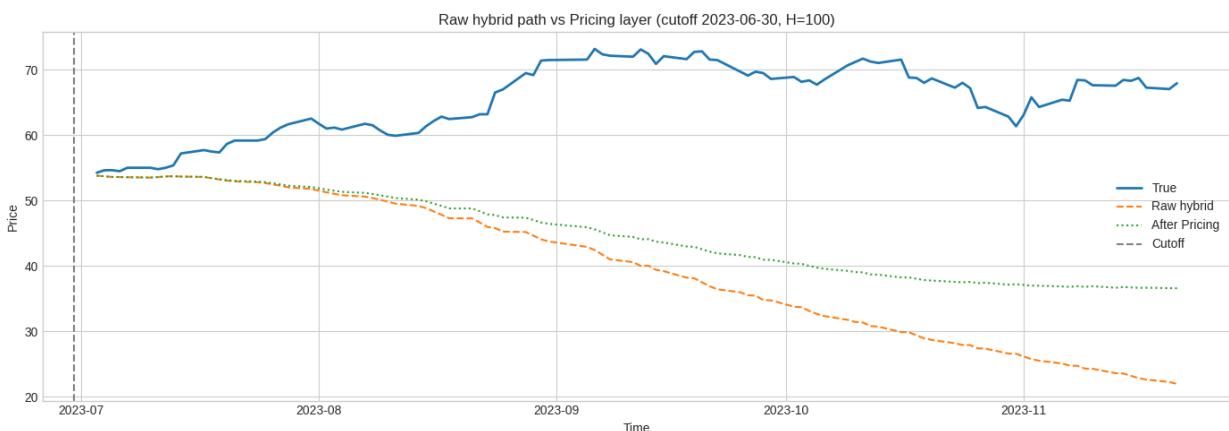
Pricing Layer: đưa “luật vật lý” vào dự báo

Nếu Math Backbone xử lý non-stationarity và XGBoost học phi tuyến, thì Pricing Layer đóng vai trò hoàn toàn khác: **nó đưa quy luật hành vi thị trường vào hệ thống**.

Các mô hình học máy thuần túy là “vô tri”: chúng không biết rằng giá không thể tăng 50% trong một ngày, hay rằng khi giá đi quá xa vùng hợp lý thì khả năng cao sẽ điều chỉnh.

Pricing Layer áp đặt các ràng buộc mềm:

- **Clipping:** giới hạn biên độ return theo lịch sử,
- **Damping:** giảm dần dao động khi đi xa tương lai,
- **Mean Reversion:** lực hút về vùng giá trị hợp lý,
- **Regime Awareness:** điều chỉnh hành vi theo Bull/Bear/Sideways.



Hình 5: So sánh quỹ đạo dự báo thô (raw hybrid path) và quỹ đạo sau khi áp dụng Pricing Layer. Đường raw hybrid có xu hướng trôi mạnh và dễ đi vào các kịch bản phi thực tế khi dự báo dài hạn. Sau khi áp dụng các cơ chế clipping, damping và mean-reversion trong Pricing Layer, quỹ đạo giá trở nên ổn định hơn và phù hợp với hành vi tài chính quan sát được trong lịch sử. Pricing Layer đóng vai trò chuyển đổi đầu ra toán học của mô hình thành một quỹ đạo giá có ý nghĩa về mặt thị trường.

Qua đó cho thấy rõ vai trò của Pricing Layer: không nhằm cải thiện độ fit ngắn hạn, mà để đảm bảo dự báo dài hạn tuân thủ các ràng buộc hành vi và rủi ro của thị trường thực.

Do đó, Feature Engineering cho Pricing Layer không nhằm tăng độ chính xác ngắn hạn, mà nhằm cung cấp thông tin để điều độ tin cậy và mức rủi ro của dự báo.

Hệ quả cho Feature Engineering

Từ kiến trúc trên, hai nguyên tắc thiết kế Feature Engineering được rút ra một cách tự nhiên:

- Feature không tồn tại độc lập, mà phục vụ rõ ràng cho từng module.
- Feature không nhằm “thêm cho đủ”, mà là thông tin tối thiểu cần thiết để mỗi lớp mô hình hoạt động đúng bản chất của nó.

Chính vì vậy, Feature Engineering trong Project 6.1 không phải là một danh sách feature rời rạc, mà là hệ quả tất yếu của việc xuất phát từ kiến trúc Hybrid được thiết kế để giải quyết bài toán T+100.

4 Từ các giả thuyết thị trường đến các nhóm Feature

Sau khi làm rõ tư duy thiết kế Feature Engineering theo hướng top-down ở Chương 3, câu hỏi tiếp theo không còn là “nên thêm feature nào nữa?” mà chuyển sang một bài toán mang tính kiến trúc hơn:

Những giả thuyết nào về hành vi thị trường cần được đưa vào pipeline để dự báo giá cổ phiếu có ý nghĩa trong vùng $T+100$?

Trong thực tế, không tồn tại một danh sách “feature chuẩn” áp dụng cho mọi bài toán. Feature Engineering hiệu quả luôn xuất phát từ một tập **giả thuyết nền tảng (market hypotheses)** về cách mà giá cổ phiếu vận động theo thời gian. Các giả thuyết này không phải tự nghĩ ra cho vui, mà được rút ra từ ba nguồn chính:

- **Sách vở kinh điển về tài chính định lượng và time-series**, nơi người ta đã chỉ ra các *stylized facts* như: chuỗi giá không dừng, volatility clustering, volume đi kèm với mức độ biến động, v.v.
- **Quan sát thực nghiệm trên dữ liệu FPT** ở Chương 1: log-price có xu hướng tăng đều, nhưng sai số bùng nổ khi dự báo dài hạn nếu không kiểm soát.
- **Yêu cầu kiến trúc Hybrid + Pricing** đã được thiết kế ở Chương 2: mỗi lớp trong pipeline cần một loại thông tin khác nhau (trend, nhiễu, regime, risk).

Dựa trên ba nguồn trên, nhóm mình không cố gắng liệt kê hàng chục giả thuyết phức tạp, mà chọn một bộ **5 giả thuyết cốt lõi** làm “xương sống” cho toàn bộ Feature Engineering:

1. **H1: Giá = Xu hướng dài hạn + Dao động ngắn hạn** (Trend dài hạn tồn tại; ngắn hạn dao động quanh một “fair level”).
2. **H2: Volatility clustering**: Có những giai đoạn thị trường êm đềm, và những giai đoạn nhiều bùng nổ; sai số dự báo tập trung ở các vùng vol cao.
3. **H3: Volume và dòng tiền xác nhận cú chuyển động giá**: Một cú tăng/giảm đi kèm volume lớn khác hoàn toàn một cú “nhảy giả” với volume thấp.
4. **H4: Cấu trúc nền chứa thông tin tâm lý intraday**: Tỷ lệ thân/nền, bóng trên/dưới, vị trí giá đóng cửa so với biên trong ngày phản ánh cuộc chiến buyer–seller.
5. **H5: Thị trường có trí nhớ ngắn hạn**: Các chuỗi nền liên tiếp (streak) tạo thành momentum ngắn hạn, nhưng sau khi quá đà sẽ có mean–reversion.

Mỗi giả thuyết ở trên tương ứng với một **vấn đề cốt lõi** trong dự báo chuỗi thời gian tài chính (non-stationarity, volatility, order-flow, sentiment, momentum), và từ đó sinh ra một hoặc vài **nhóm feature chuyên biệt**.

Chương 4 không nhằm lặp lại các kết quả EDA, mà tập trung trình bày **logic thiết kế**:

Giả thuyết thị trường \Rightarrow Hàm ý lên quỹ đạo giá $T+100 \Rightarrow$ Nhóm feature cần có trong kiến trúc Hybrid + Pricing.

Trong các mục tiếp theo, mỗi giả thuyết sẽ được trình bày theo format: (i) Phát biểu giả thuyết → (ii) Hàm ý đối với mô hình Hybrid → (iii) Nhóm feature cụ thể được rút ra từ giả thuyết đó.

4.1 Giả thuyết 1: Giá = Trend dài hạn + Dao động ngắn hạn (*Trend & Mean-Reversion*)

(i) Phát biểu giả thuyết

Một trong những quan sát nền tảng nhất của thị trường chứng khoán là: **giá cổ phiếu không vận động như random walk thuần túy**, mà thường thể hiện một cấu trúc hai tầng rõ rệt: *tăng trưởng dài hạn* xen kẽ với *dao động ngắn hạn*.

Đối với các cổ phiếu tăng trưởng lớn như FPT, cấu trúc này có thể được diễn đạt gọn gàng trên thang log-price dưới dạng:

$$\log P_t = \underbrace{T_t}_{\text{xu hướng dài hạn}} + \underbrace{\varepsilon_t}_{\text{dao động ngắn hạn}}, \quad (1)$$

trong đó T_t đại diện cho xu hướng tăng trưởng dài hạn, biến thiên chậm và tương đối ổn định, còn ε_t phản ánh các dao động ngắn hạn quanh một mức giá hợp lý (fair level).

Cách nhìn này tương ứng với hai đặc điểm hành vi quen thuộc của thị trường:

- Về dài hạn, giá của các doanh nghiệp tăng trưởng tốt mang tính **hướng xu thế** (directional), thể hiện bằng một quỹ đạo tăng mượt khi quan sát trên log-price.
- Về ngắn hạn, giá không “đi thẳng”, mà liên tục **dao động quanh xu hướng**: có giai đoạn bị đẩy lên quá cao do hưng phấn, có giai đoạn bị ép xuống quá thấp do áp lực bán, và thường có xu hướng quay lại quanh vùng cân bằng.

Từ góc nhìn dự báo dài hạn T+100, điều quan trọng nhất của giả định này không nằm ở bản thân công thức, mà ở cách phân chia vai trò giữa các thành phần của mô hình. Nếu không tách riêng hai lớp này, mô hình học máy sẽ buộc phải đồng thời học cả xu hướng dài hạn lẫn nhiều ngắn hạn, dẫn đến bất ổn nghiêm trọng khi extrapolate xa khỏi vùng dữ liệu huấn luyện.

Vì vậy, **Giả thuyết 1** được phát biểu như sau:

Đối với bài toán dự báo giá FPT trong vùng T+100, việc mô hình hóa chính xác quỹ đạo giá không nên bắt đầu từ dự báo mức giá tuyệt đối, mà từ việc tách rõ xu hướng dài hạn đóng vai trò “backbone” khỏi các dao động ngắn hạn quanh nó, để mỗi thành phần của pipeline xử lý đúng bản chất của mình.

(ii) Hàm ý đối với kiến trúc Hybrid + Pricing

Về cơ bản, giả thuyết 1 làm rõ Câu hỏi: “**Giá FPT có bản chất gì?**”. Qua đó giải thích trực tiếp vì sao pipeline được thiết kế thành **ba lớp**:

1. **Math Backbone (Linear Trend)** xử lý phần *trend dài hạn*. Thay vì bắt XGBoost hoặc Deep Learning học cả xu hướng lẫn nhiều, ta dùng một mô hình toán đơn giản (Linear Regression trên log-price) để mô hình hoá T_t . Điều này giải quyết bài toán **không dừng** của chuỗi giá: phần còn lại sau khi trừ trend sẽ giàn dừng hơn.
2. **XGB Residual** xử lý *dao động ngắn hạn*. Target của XGB không phải chính giá, mà là *residual return*:

$$\text{resid_ret}_t = \underbrace{\log P_{t+1} - \log P_t}_{\text{future_ret}} - \underbrace{(T_{t+1} - T_t)}_{\text{math_ret}},$$

tức là phần chênh lệch giữa return thật và return do trend tạo ra. Nhờ đó, XGB tập trung toàn bộ năng lực vào việc học “nhịp nhảy” quanh trend, thay vì phải gánh luôn xu hướng dài hạn.

3. **Pricing Layer** giám sát *khoảng cách tới fair level*. Khi giá đi quá xa đường trend/fair level, lớp Pricing áp các cơ chế clipping, damping và mean-reversion để kéo quỹ đạo dự báo quay về vùng hợp lý.

Như vậy, Giả thuyết 1 không chỉ là một mô tả định tính, mà **áp đặt cấu trúc** cho toàn bộ mô hình:

- Mô hình toán đơn giản lo phần *trend*.
- XGB lo *nhiều quanh trend*.
- Pricing Layer đảm bảo quỹ đạo tổng thể luôn bám quanh một *fair level* hợp lý trong vùng T+100.

(iii) Nhóm feature *Trend & Distance-to-Mean*

Khi chấp nhận giả định này, Feature Engineering không còn là quá trình “khai quật tín hiệu từ dữ liệu”, mà trở thành một bài toán kiểm tra ngược:

Liệu các feature đã mô tả đủ tốt trend để phần residual thực sự chỉ còn chứa những biến động bất ngờ?

Từ góc nhìn đó, việc sinh ra các feature trở nên gần như tất yếu:

- Nếu coi **trend** là backbone, ta cần feature mô tả *mức độ* (level), *tốc độ* (slope) và *gia tốc* (acceleration) của trend.
- Nếu coi thị trường có xu hướng dao động quanh một **fair level**, ta cần các feature đo *khoảng cách tới trạng thái cân bằng* (z-score, distance-to-mean).

Theo nghĩa này, các feature **không phải là phát hiện ngẫu nhiên sau EDA**, mà là hệ quả trực tiếp của tư duy: “feature phải phục vụ giả định $\log P = \text{trend} + \text{residual}$ ”, để XGBoost chỉ học phần residual đúng nghĩa.

Từ đó, nhóm feature đầu tiên được sinh ra là **nhóm Trend & Distance-to-Mean**, với hai nhiệm vụ:

1. Mô tả hình dạng và tốc độ của xu hướng dài hạn.
2. Đo lường mức độ “lệch chuẩn” của giá so với fair level, để XGB và Pricing nhận biết khi nào thị trường đang kéo dây thun quá đà.

Trong code, nhóm feature này bao gồm chủ yếu các biến sau:

- **Trend mượt và đạo hàm:**
 - `trend_stl` – trend mượt từ STL trên `close_log`.
 - `trend_slope_1`, `trend_slope_3`, `trend_slope_7` – độ dốc của trend ở các khung 1, 3, 7 ngày, cho biết xu hướng hiện tại đang tăng nhanh hay chậm.
 - `trend_accel_3` – chênh lệch slope, phản ánh gia tốc của trend (trend đang tăng tốc hay giảm tốc).
- **Thống kê quanh trend:**
 - `trend_mean_21`, `trend_std_21` – trung bình và độ lệch chuẩn của trend trên cửa sổ 21 ngày.

- z_trend_21 – z-score của trend so với chính nó:

$$z_trend, 21(t) = \frac{\text{trend_stl}(t) - \text{trend_mean_21}(t)}{\text{trend_std_21}(t) + \varepsilon},$$

cho biết trend hiện tại đang ở vùng “quá cao” hay “quá thấp” so với mức bình thường 1 tháng.

- **Biến động quanh trend (residual volatility):**

- resid_std_10 , resid_std_20 – độ lệch chuẩn của residual STL trong 10–20 ngày gần nhất, dùng để đo độ “rung lắc” quanh trend.

Một điểm cần làm rõ là: các cửa sổ thời gian được sử dụng trong nhóm feature (1, 3, 7, 21, 10–20 ngày) không phải được chọn ngẫu nhiên hay thông qua brute-force tuning, mà xuất phát từ logic thị trường và logic động học của xu hướng.

(i) **Các thang 1–3–7 ngày cho độ dốc xu hướng.** Ba cửa sổ này phản ánh *đa thang thời gian* của chuyển động giá:

- cửa sổ 1 ngày đo phản ứng tức thời của trend,
- cửa sổ 3 ngày nắm bắt nhịp dao động ngắn hạn,
- cửa sổ 7 ngày đại diện cho một chu kỳ giao dịch ngắn (weekly structure).

Việc cung cấp đồng thời nhiều thang slope giúp mô hình nhận biết không chỉ *hướng* của xu hướng, mà còn *tốc độ thay đổi* của xu hướng theo thời gian.

(ii) **Gia tốc xu hướng (trend_accel_3).** Gia tốc của trend đo sự thay đổi của slope trong ngắn hạn, cho phép mô hình phát hiện sớm các trạng thái *mất đà* hoặc *tăng tốc* ngay cả khi xu hướng tổng thể chưa đảo chiều. Đây là tín hiệu đặc biệt quan trọng trong bối cảnh dự báo dài hạn T+100, nơi các cú gãy xu hướng thường bắt đầu bằng sự suy yếu động lượng.

(iii) **Cửa sổ 21 ngày cho thống kê quanh trend.** Cửa sổ 21 ngày xấp xỉ một tháng giao dịch, được xem là thang thời gian “trung hạn” đủ ổn định để xác định một mức hành vi bình thường (normal state) của xu hướng, nhưng vẫn đủ linh hoạt để phản ứng với sự thay đổi regime.

(iv) **Cửa sổ 10–20 ngày cho biến động phần dư.** Không giống xu hướng dài hạn, phần residual phản ánh nhiều và cú shock ngắn hạn. Do đó, biến động residual được đo trên cửa sổ ngắn (10–20 ngày) để phản ánh chính xác trạng thái rung lắc hiện tại của thị trường, phục vụ trực tiếp cho kiểm soát rủi ro và cơ chế điều chỉnh trong Pricing Layer.

Nhìn tổng thể, các thang thời gian này tạo thành một hệ thống *đa phân giải theo thời gian* (multi-resolution), giúp mô hình Hybrid + Pricing vừa nắm được cấu trúc dài hạn, vừa phản ứng phù hợp với động học ngắn hạn của thị trường.

Về mặt vai trò trong pipeline:

- Các feature về **trend_stl** và **slope/accel** giúp **Math Backbone** và **XGB Residual** cùng nhau thấy “trục chuyển động chính” của FPT, tránh nhầm lẫn giữa xu hướng và nhiễu.
- Các feature **z_trend_21** và **resid_std_*** cung cấp cho **Pricing Layer** thước đo định lượng về việc giá đang lệch xa khỏi vùng cân bằng, hỗ trợ quyết định khi nào cần kích hoạt mean-reversion mạnh hơn.

Nóm feature *Trend & Distance-to-Mean* vì thế chính là hiện thân cụ thể của Giả thuyết 1 trong code: nó biến câu nói trừu tượng “Giá = Trend + Dao động” thành các cột số cụ thể mà mô hình Hybrid + Pricing có thể học và hành động trên đó.

4.2 Giả thuyết 2: Sai số lớn luôn đi kèm Volatility & Shock (*Volatility & Risk-awareness*)

(i) Phát biểu giả thuyết

Một trong những *stylized facts* kinh điển của tài chính định lượng là **volatility clustering**: độ biến động của lợi suất không trải đều theo thời gian, mà tập trung thành từng cụm: sau những ngày biến động mạnh, khả năng cao thị trường sẽ tiếp tục “rung lắc” mạnh hơn bình thường trong một thời gian nữa.

Song song với đó, các mô hình dự báo thực tế cho thấy: **sai số lớn hiếm khi xuất hiện trong giai đoạn thị trường yên ả**, mà chủ yếu rơi vào những đoạn có *shock* (gap lớn, biên độ nền cực rộng, volume bùng nổ, tin tức bất ngờ).

Từ góc nhìn dự báo T+100, có thể phát biểu **Giả thuyết 2** như sau:

*Phần lớn sai số lớn trong forecast không đến từ việc “tự sai trend”, mà từ việc **không nhận diện kịp thời** các giai đoạn thị trường đang ở trạng thái biến động cao hoặc vừa trải qua shock.*

Nói cách khác, muốn forecast có ý nghĩa, mô hình không chỉ cần biết *giá đang ở đâu so với fair level* (Giả thuyết 1), mà còn phải biết *thị trường hiện đang “bình thường” hay đang “diễn”*. Nếu không, cùng một bước nhảy dự báo có thể chấp nhận được trong ngày yên ắng, nhưng lại trở nên cực kỳ rủi ro khi thị trường vừa ăn một cú shock.

(ii) Hàm ý đối với kiến trúc Hybrid + Pricing

Giả thuyết 2 làm rõ *Câu hỏi: “Sai số lớn thường đến từ đâu?”*. Trong ngữ cảnh Hybrid + Pricing, câu trả lời ngắn gọn là: **từ những đoạn volatility cao và shock**, nếu mô hình không có cơ chế nhận biết và tự *giảm độ hung hăng* trong giai đoạn đó.

Điều này dẫn tới hai hàm ý cấu trúc:

1. XGB Residual phải nhận biết được trạng thái volatility.

Target của XGB là `resid_ret`, tức phần dao động quanh trend. Nếu mô hình coi mọi ngày đều giống nhau, nó có xu hướng “overfit” vào những đoạn volatility cao: học các bước nhảy rất lớn trong giai đoạn sôi động, rồi vô tình áp dụng các bước nhảy đó vào giai đoạn bình thường. Vì vậy, trong input của XGB bắt buộc phải có một nhóm feature mô tả:

- độ rộng nền, biên độ intraday, khoảng cách high-low,
- mức độ lớn nhỏ của return gần đây,
- volume và money flow bất thường.

Khi nhận ra đang ở đoạn *high vol*, XGB có thể tự “học” cách co bóp bước nhảy hoặc thay đổi hành vi.

2. Pricing Layer phải là một lớp “risk controller” phụ thuộc volatility.

Pricing Layer không chỉ clip return theo một ngưỡng cố định, mà clip theo *phân phối historical volatility*. Trong code, ngưỡng clip cơ bản được suy ra từ phân vị của `|ret_1d|`, sau đó được scale theo chế độ thị trường (BULL/BEAR/SIDEWAYS) và trạng thái volatility gần nhất. Nhờ đó:

- khi thị trường đang yên ắng, Pricing không cho phép bước nhảy quá lớn,
- khi thị trường đang biến động mạnh, Pricing “nói” biên độ một chút, nhưng vẫn giữ được kiểm soát để forecast không nổ tung.

Tóm lại, Giả thuyết 2 yêu cầu cả hai lớp **XGB Residual** và **Pricing Layer** phải *nhận thức được mức độ rủi ro hiện tại của thị trường*, thay vì chỉ nhìn vào vị trí của giá so với trend.

(iii) Nhóm feature *Volatility & Shock*

Từ Giả thuyết 2, nhóm feature thứ hai được sinh ra là **nhóm Volatility & Shock**, nhằm trả lời một câu hỏi rất cụ thể:

Trong vài ngày gần đây, thị trường đang yên tĩnh, rung lắc vừa phải, hay đang ở trong một cụm biến động mạnh / vừa ăn shock?

Nhóm feature này chia làm ba lớp chính:

- **Độ lớn của return (return magnitude).**

- `ret_1d`, `ret_5d`, `ret_10d` – các log-return ở thang 1, 5, 10 ngày, cho phép mô hình đo mức độ dịch chuyển giá ngắn hạn và trung ngắn hạn.
- Thông qua phân phối của $|ret_1d|$, Pricing Layer suy ra ngưỡng clip hợp lý cho return mỗi ngày.

- **Biến động intraday và nền “bất thường”.**

- `range`, `true_range` – biên độ high-low và true range, phản ánh độ rộng của nền trong ngày.
- `range_pct` – biên độ intraday tương đối so với giá, cho biết nền hôm nay rộng hơn bao nhiêu phần trăm so với mặt bằng chung.
- `atr_14` – Average True Range 14 ngày, chuẩn hoá mức độ rung lắc trung hạn.
- `range_ma_10`, `range_expansion` – trung bình biên độ 10 ngày và flag “bùng nổ biên độ”, đánh dấu những phiên có nền rộng bất thường (ví dụ: gap lớn, panic sell/buy).
- `park_vol` – ước lượng volatility dựa trên công thức Parkinson, tận dụng full thông tin high-low thay vì chỉ dùng close-to-close return.

- **Volume & money flow bất thường.**

- `volume`, `vol_ma_5`, `vol_ma_20` – volume hiện tại và trung bình ngắn/trung hạn.
- `vol_ratio`, `vol_z_20` – tỷ lệ và z-score volume so với 20 ngày gần nhất, giúp nhận diện các phiên “cạn thanh khoản” hoặc “bùng nổ volume”.
- `money_flow`, `money_flow_5` – dòng tiền danh nghĩa trong ngày và bình quân 5 ngày, kết hợp cả price và volume.
- `vol_ratio_lag1` – độ bền của trạng thái volume bất thường từ ngày liền trước.

Một phần các feature volatility này giao thoa với nhóm *Trend & Distance-to-Mean* trong Giả thuyết 1 (đặc biệt là `resid_std_10`, `resid_std_20`), nhưng góc nhìn ở đây đã chuyển hoàn toàn sang *risk-awareness*:

- `resid_std_10`, `resid_std_20` – độ lệch chuẩn của residual STL trong 10–20 ngày gần nhất, được xem như thước đo mức độ “rung lắc” quanh trend, trực tiếp liên quan đến rủi ro forecast sai số lớn.

Tương tự Giả thuyết 1, các *cửa sổ thời gian* trong nhóm feature Volatility & Shock không được chọn ngẫu nhiên, mà phản ánh cấu trúc động học của rủi ro:

(i) Thang 1–5–10 ngày cho return & intraday range.

- 1 ngày: phản ứng tức thời với shock hoặc news.

- 5 ngày: xấp xỉ một tuần giao dịch, đủ để xem shock có kéo dài hay không.
- 10 ngày: hai tuần giao dịch, bắt đầu phản ánh xem thị trường đang bước vào một cụm volatility mới hay chỉ là cú nhiễu ngắn hạn.

(ii) **Cửa sổ 14 ngày cho ATR.** Thang 14 ngày là chuẩn kinh điển trong technical analysis cho Average True Range, cân bằng giữa việc phản ứng với biến động mới và giữ được độ ổn định của thước đo volatility trung hạn.

(iii) Cửa sổ 20 ngày cho volume & residual volatility.

- 20 ngày xấp xỉ một tháng giao dịch, là thang tự nhiên để xác định “mặt bằng volume tiêu chuẩn” và “mức rung lắc tiêu chuẩn” của residual.
- Các z-score và ratio trên 20 ngày vì thế có ý nghĩa “bao xa so với bình thường”, rất phù hợp cho việc kích hoạt hoặc nối lồng các cơ chế kiểm soát rủi ro.

Nhìn tổng thể, nhóm *Volatility & Shock* cùng với nhóm *Trend & Distance-to-Mean* tạo thành một hệ đa phân giải theo thời gian (multi-resolution):

- H1 lo **hình dáng và vị trí** của quỹ đạo giá so với fair level.
- H2 lo **mức độ rung lắc và bất thường** quanh quỹ đạo đó.

Theo nghĩa này, Giả thuyết 2 được cụ thể hoá thành code: mỗi khi thị trường bước vào trạng thái “điên”, các cột volatility & shock sẽ bật sáng, để XGB Residual và Pricing Layer tự động chuyển sang chế độ *risk-aware* trong dự báo T+100.

4.3 Giả thuyết 3: Volume phản ánh sức mạnh thực sự của một cú move (*Volume & Order-flow*)

(i) Phát biểu giả thuyết

Một hạn chế căn bản của các mô hình dựa thuần trên giá là: *giá cho biết giá đã đi đâu, nhưng không cho biết vì sao nó đi như vậy*. Trong thực tế thị trường tài chính, câu hỏi quan trọng không kém hướng giá là:

Cú tăng/giảm này được bao nhiêu dòng tiền và bao nhiêu người ủng hộ?

Lý thuyết thị trường cổ điển và thực hành giao dịch đều thống nhất ở một điểm: **volume đóng vai trò là proxy cho mức độ tham gia của thị trường** (participation và conviction).

- Một cú tăng giá **đi kèm volume lớn** thường phản ánh dòng tiền thật sự tham gia, do đó có xác suất tiếp diễn cao hơn.
- Ngược lại, một cú tăng giá **không có volume ủng hộ** (price up, volume thấp) thường mang tính kỹ thuật, dễ đảo chiều hoặc mất hiệu lực.

Theo cách nhìn này, volume không chỉ là biến phụ trợ, mà phản ánh trực tiếp *ai đang đứng sau cú move*: nhà đầu tư lớn hay nhỏ, dòng tiền bền vững hay chỉ là nhiễu ngắn hạn.

Tóm lại, **Giả thuyết 3** khẳng định:

Giá chỉ mô tả chuyển động bề mặt, còn volume và order-flow mới phản ánh sức mạnh thực sự phía sau mỗi cú move. Một mô hình dự báo dài hạn chỉ dựa vào giá là chưa đủ.

(ii) Hàm ý đối với kiến trúc Hybrid + Pricing

Giả thuyết này áp đặt một yêu cầu rõ ràng lên pipeline:

Mô hình cần phân biệt được đâu là cú move “có tiền thật chống lưng”, và đâu là cú move yếu, dễ bị phủ định trong tương lai.

Trong kiến trúc Hybrid + Pricing, vai trò của thông tin volume được phân bổ như sau:

- **XGB Residual** sử dụng feature volume để điều chỉnh dự báo phần dao động ngắn hạn: cùng một tín hiệu giá, nhưng nếu volume khác nhau, phản ứng dự đoán nên khác nhau.
- **Pricing Layer** sử dụng volume như một yếu tố đánh giá *độ tin cậy* của quỹ đạo dự báo: cú break có volume lớn được cho phép đi xa hơn, trong khi cú move yếu cần bị damping sớm hơn.

Theo nghĩa này, volume không nhằm dự báo giá trực tiếp, mà đóng vai trò **risk-awareness signal**: nó cho mô hình biết nên tin cú move hiện tại đến mức nào.

(iii) Nhóm feature Volume & Money Flow

Khi chấp nhận giả thuyết “volume phản ánh sức mạnh của cú move”, Feature Engineering trở thành bài toán trả lời câu hỏi:

Cần đo những đại lượng nào để mô hình nhận biết liệu một cú tăng/giảm đang được thị trường ủng hộ hay không?

Từ tư duy đó, nhóm feature Volume & Order-flow được hình thành với ba mục tiêu chính:

1. Phát hiện **volume bất thường** so với trạng thái bình thường.
2. Xác định **hướng tích luỹ của dòng tiền**.
3. Đánh giá **cường độ tiền thật** chảy vào cổ phiếu, chứ không chỉ số lượng cổ phiếu được giao dịch.

Trong code, nhóm feature này bao gồm chủ yếu các biến sau:

- **Volume bất thường:**

- `vol_z_20` – z-score của volume so với trung bình 20 ngày, dùng để phát hiện các ngày giao dịch đột biến.
- `vol_ratio_20` – tỷ lệ volume hiện tại so với MA20, cho biết mức độ “đồng người tham gia” vào phiên đó.

- **Hướng dòng tiền (Order-flow proxy):**

- `obv` (On-Balance Volume) – tích luỹ volume có hướng, phản ánh xu hướng dòng tiền.
- `obv_diff_1d` – thay đổi OBV trong ngày, cho biết dòng tiền đang tăng tốc hay suy yếu.

- **Money Flow (giá × volume):**

- `money_flow = close × volume` – đo quy mô tiền thực sự tham gia vào cổ phiếu.
- `money_flow_5` – money flow được làm mượt trên 5 ngày để lọc nhiễu ngắn hạn.

Ý nghĩa của các thang thời gian. Tương tự các nhóm feature trước, các cửa sổ trong nhóm volume không được chọn ngẫu nhiên:

- Cửa sổ 20 ngày xấp xỉ một tháng giao dịch, đại diện cho trạng thái volume “bình thường” của cổ phiếu.
- Cửa sổ 5 ngày dùng để làm mượt money flow, nhằm phân biệt dòng tiền tích luỹ thật sự với các cú spike volume ngắn hạn.

Vai trò trong pipeline. Nhóm feature *Volume & Money Flow* giúp mô hình:

- Phân biệt hai cú move có cùng hình dạng giá nhưng *sức mạnh thị trường hoàn toàn khác nhau*.
- Giảm rủi ro over-trusting các tín hiệu giá yếu, đặc biệt trong dự báo dài hạn T+100.

Như vậy, nhóm Volume & Order-flow chính là hiện thân của Giả thuyết 3 trong code: nó biến câu hỏi định tính “*ai đang đứng sau cú move này?*” thành các đại lượng định lượng mà Hybrid Model và Pricing Layer có thể trực tiếp khai thác.

4.4 Giả thuyết 4: Mỗi cây nến là một bản tóm tắt tâm lý thị trường (*Candlestick Micro-structure*)

Candlestick là gì và vì sao nó quan trọng?

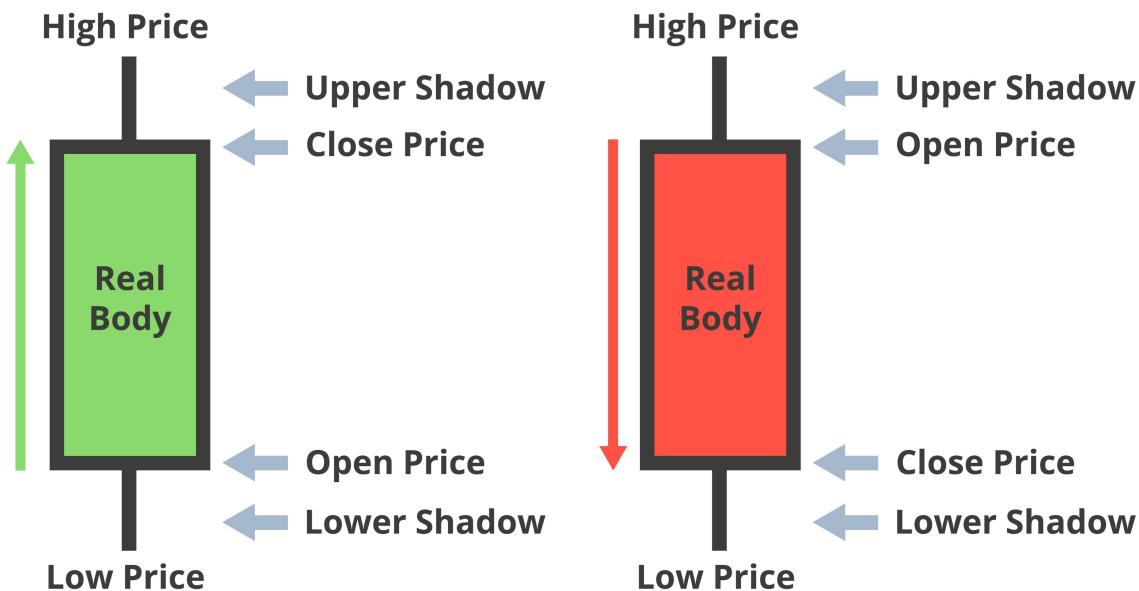
Trong dữ liệu giá cổ phiếu theo ngày, mỗi quan sát không chỉ chứa một con số duy nhất, mà bao gồm bốn mức giá cơ bản:

- **Open:** giá mở cửa phiên giao dịch,
- **High:** giá cao nhất đạt được trong phiên,
- **Low:** giá thấp nhất trong phiên,
- **Close:** giá đóng cửa phiên.

Tổ hợp bốn mức giá này thường được biểu diễn dưới dạng **candlestick (nến giá)**, trong đó hình dạng của cây nến thể hiện trực quan toàn bộ diễn biến cạnh tranh giữa bên mua và bên bán trong một ngày. Cụ thể, một cây nến gồm ba thành phần chính:

- **Thân nến (real body):** đoạn nối giữa **open** và **close**, phản ánh mức độ thắng thua của bên mua hay bên bán trong phiên.
- **Bóng nến trên (upper shadow):** phần từ $\max(open, close)$ đến **high**, thể hiện lực bán xuất hiện khi giá bị đẩy lên cao.
- **Bóng nến dưới (lower shadow):** phần từ **low** đến $\min(open, close)$, thể hiện lực mua xuất hiện khi giá bị ép xuống thấp.

CANDLESTICK COMPONENTS



Hình 6: Cấu trúc cơ bản của candlestick. Thân nến phản ánh sức mạnh tương đối của bên thắng trong phiên, trong khi bóng nến trên/dưới thể hiện các vùng giá bị từ chối bởi lực mua hoặc lực bán.

Điểm mấu chốt là:

Hai ngày có cùng giá đóng cửa có thể mang hai trạng thái tâm lý thị trường hoàn toàn khác nhau.

Ví dụ:

- Một phiên tăng giá với **thân nến dài, bóng ngắn** thường thể hiện áp lực mua áp đảo xuyên suốt phiên.
- Một phiên đóng cửa cao nhưng có **upper shadow dài** cho thấy giá từng bị đẩy lên cao nhưng bị lực bán mạnh kéo ngược trở lại — một dạng *rejection*.

Vì vậy, nếu mô hình chỉ quan sát giá đóng cửa (*close*) hoặc return, toàn bộ thông tin về *cách mà giá di chuyển trong phiên* sẽ bị mất đi.

Candlestick, theo góc nhìn này, không phải là công cụ trực quan cho con người, mà là một cách **nén thông tin tâm lý thị trường** vào các hình học đơn giản (body, shadow, vị trí đóng cửa), có thể được mã hóa thành feature cho mô hình học máy.

Chính từ nhận thức nền tảng này, Giả thuyết 4 được hình thành.

(i) Phát biểu giả thuyết

Các giả thuyết trước đã lần lượt mô tả thị trường ở những lớp thông tin khác nhau: xu hướng dài hạn (trend), mức độ bất ổn (volatility), và cường độ dòng tiền (volume). Tuy nhiên, vẫn còn một lớp thông tin quan trọng nằm hoàn toàn ở *cấp độ vi mô* của từng phiên giao dịch.

Trong phân tích kỹ thuật, mỗi cây nến (candlestick) không được xem là một điểm giá đơn lẻ, mà là kết quả cuối cùng của một quá trình giằng co liên tục giữa bên mua và bên bán trong suốt phiên. Cùng một mức giá đóng cửa, nhưng đường đi khác nhau bên trong phiên có thể phản ánh những trạng thái tâm lý thị trường hoàn toàn khác biệt.

Từ góc nhìn này, cấu trúc hình học của cây nến (body, upper shadow, lower shadow, vị trí đóng cửa) mang thông tin trực tiếp về:

- bên nào kiểm soát phần lớn thời gian giao dịch,
- giá bị từ chối ở vùng nào trong ngày,
- và mức độ quyết liệt của lực mua – bán trong phiên.

Nếu chỉ sử dụng giá đóng cửa hoặc return, toàn bộ thông tin về cách mà giá hình thành trong ngày sẽ bị nén lại thành một con số duy nhất. Điều này khiến mô hình bỏ lỡ các tín hiệu vi mô quan trọng, đặc biệt khi thị trường đang ở trạng thái lưỡng lự hoặc chuẩn bị đảo chiều.

Từ đó, **Giả thuyết 4** được phát biểu như sau:

Mỗi cây nến là một bản tóm tắt tâm lý thị trường trong ngày giao dịch. Để hiểu đúng hành vi ngắn hạn của giá, mô hình cần quan sát không chỉ giá đi đến đâu, mà còn giá đã đi như thế nào trong phiên.

(ii) Hàm ý đối với kiến trúc Hybrid + Pricing

Giả thuyết này dẫn tới một thay đổi quan trọng trong tư duy thiết kế feature:

Không thể coi mỗi ngày giao dịch chỉ là một scalar giá đóng cửa.

Trong kiến trúc Hybrid + Pricing:

- **XGB Residual** cần đọc được các tín hiệu *follow-through* hoặc *exhaustion* từ cấu trúc nến để dự đoán chuyển động ngắn hạn tiếp theo.
- **Pricing Layer** cần nhận biết những ngày thị trường thể hiện sự từ chối giá mạnh (upper/lower shadow dài), vì các trạng thái này thường đi kèm xác suất mean-reversion cao hơn.

Nếu bỏ qua cấu trúc nến, mô hình sẽ rơi vào trạng thái *closing-price-only*, tức là chỉ biết *giá đã đi tới đâu*, mà không biết *nó đi tới đó bằng cách nào*.

Do đó, một nhóm feature riêng biệt là cần thiết để chuyển hình dạng của cây nến thành các đại lượng mà mô hình có thể xử lý.

(iii) Nhóm feature *Candlestick Micro-structure*

Khi chấp nhận rằng mỗi cây nến là một bản tóm tắt tâm lý, Feature Engineering trở thành bài toán:

Làm thế nào để mã hóa cấu trúc hình học của cây nến thành các đại lượng bất biến theo thang đo giá?

Từ đó, nhóm feature **Candlestick Micro-structure** được sinh ra, với ba mục tiêu chính:

1. Đo sức mạnh tương đối của bên thắng trong phiên.
2. Đo mức độ từ chối giá ở hai đầu biên.
3. Đo vị trí đóng cửa trong toàn bộ biên độ dao động ngày.

Trong code, nhóm feature này bao gồm chủ yếu:

- **Thân nến (Body):**
 - `body = close - open,`
 - `range = high - low,`
 - `body_rel = body / (range + eps)`: chuẩn hoá sức mạnh thân nến theo biên độ dao động trong ngày.
- **Bóng nến (Shadow):**
 - `upper_shadow,`
 - `lower_shadow`, phản ánh lực bán/mua ngược chiều.
- **Vị trí đóng cửa:**
 - `close_pos`, đo vị trí tương đối của giá đóng cửa trong khoảng $[low, high]$, cho biết bên nào nắm quyền kiểm soát vào cuối phiên.
- **Hiệu ứng lan truyền (lagged psychology):**
 - `body_lag1,`
 - `close_pos_lag1`, dùng để bắt tín hiệu follow-through sang ngày kế tiếp.

Đáng chú ý, các feature này:

- không phụ thuộc vào mức giá tuyệt đối,
- không cần giả định phân phối,
- nhưng mang thông tin tâm lý rất mạnh ở cấp độ vi mô.

Về mặt vai trò trong pipeline:

- Nhóm *Candlestick Micro-structure* giúp **XGB Residual** hiểu được trạng thái tâm lý ngay tại thời điểm phát sinh residual.
- Đồng thời cung cấp cho **Pricing Layer** tín hiệu bổ sung để điều chỉnh cường độ clipping và mean-reversion trong các ngày có rejection mạnh.

Như vậy, Giả thuyết 4 hoàn thiện góc nhìn đa tầng của pipeline:

- Trend cho cấu trúc dài hạn,
- Volatility cho rủi ro,
- Volume cho dòng tiền,
- và Candlestick cho **tâm lý vi mô trong từng phiên**.

Nhóm feature *Candlestick Micro-structure* vì thế chính là lời giải cho câu hỏi: “*Tâm lý thị trường trong một ngày thực sự trông như thế nào, và làm sao để mô hình không bỏ lỡ nó?*”

4.5 Giả thuyết 5: Thị trường có trí nhớ ngắn hạn (*Momentum & Streaks*)

(i) Phát biểu giả thuyết

Bên cạnh xu hướng dài hạn, biến động, dòng tiền và cấu trúc vi mô trong từng phiên, thị trường tài chính còn thể hiện một đặc tính quan trọng khác: **hành vi hiện tại chịu ảnh hưởng đáng kể từ các chuyển động gần nhất trong quá khứ**.

Trong nhiều tài liệu phân tích kỹ thuật và tài chính định lượng, hiện tượng này thường được mô tả dưới hai dạng tưởng như đối lập nhưng cùng tồn tại:

- **Short-term momentum:** các chuỗi tăng/giảm ngắn hạn có xu hướng tiếp diễn trong một số phiên kế tiếp do quán tính tâm lý và hành vi bầy đàn.
- **Exhaustion & mean-reversion:** khi chuỗi tăng/giảm kéo dài quá mức, xác suất đảo chiều lại tăng lên do mệt mỏi mua/bán.

Nói cách khác, thị trường không “quên sạch” ngày hôm qua, nhưng cũng không ghi nhớ một cách tuyến tính hay vô hạn. Thay vào đó, tồn tại một dạng **trí nhớ ngắn hạn, có điều kiện theo bối cảnh**.

Giả thuyết này có thể được phát biểu dưới dạng khái quát như sau:

Trong ngắn hạn, hành vi giá chịu ảnh hưởng bởi các mẫu hình và chuỗi chuyển động vừa xảy ra. Momentum và mean-reversion không loại trừ lẫn nhau, mà cùng tồn tại như hai phản ứng khác nhau của thị trường trước trạng thái hiện tại.

(ii) Hàm ý đối với kiến trúc Hybrid + Pricing

Giả thuyết “trí nhớ ngắn hạn” mang một hệ quả quan trọng: **thông tin hình thái của vài phiên gần nhất** không thể được suy ra đầy đủ chỉ từ trend, volatility hay volume tức thời.

Trong kiến trúc Hybrid + Pricing, điều này dẫn đến ba yêu cầu thiết kế:

1. Mô hình cần nhận biết được **hướng chuyển động gần nhất** (giá vừa tăng hay giảm?).
2. Mô hình cần biết **chuỗi này đã kéo dài bao lâu** (một ngày đơn lẻ hay nhiều ngày liên tiếp).
3. Mô hình cần tự quyết định trong từng bối cảnh nên coi chuỗi đó là *momentum tiếp diễn* hay *tín hiệu cạn kiệt* dẫn đến *đảo chiều*.

Nếu các thông tin này không được mã hoá rõ ràng, XGBoost buộc phải suy luận gián tiếp từ các biến liên tục như return hoặc volatility, dễ dẫn tới mô hình hoá thiếu ổn định trong dự báo T+100.

Do đó, thay vì kỳ vọng mô hình “nhớ hộ” quá khứ, pipeline chủ động sinh ra các feature để **mô tả trực tiếp trạng thái ký ức ngắn hạn của thị trường**.

(iii) Nhóm feature *Patterns & Streaks*

Khi chấp nhận rằng thị trường có trí nhớ ngắn hạn, Feature Engineering không còn dừng ở việc mô tả từng ngày độc lập, mà mở rộng sang việc mô tả *chuỗi các ngày liên tiếp*.

Các feature trong nhóm này tập trung trả lời ba câu hỏi:

1. Hôm nay giá tăng hay giảm?
2. Chuỗi tăng/giảm hiện tại đã kéo dài bao nhiêu phiên?
3. Lực đi theo chuỗi đó đang mạnh dần hay suy yếu?

Trong code, nhóm feature này bao gồm chủ yếu các biến sau:

- **Hướng chuyển động đơn ngày:**
 - up_1d, down_1d – biến nhị phân mã hoá hướng tăng/giảm của giá trong phiên hiện tại.
- **Chuỗi liên tiếp (streaks):**
 - up_3streak – số ngày tăng liên tiếp gần nhất.
 - down_3streak – số ngày giảm liên tiếp gần nhất.
- **Mẫu hình ngắn hạn mở rộng:**
 - rolling sum hoặc rolling count của các phiên tăng/giảm trên cửa sổ ngắn, nhằm mô tả cường độ và tính bền của momentum.



Hình 8: Minh họa chuỗi nến tăng/giảm liên tiếp của cổ phiếu FPT trong 60 phiên gần nhất đến ngày 10/3/2025 (candlestick style kèm volume). Các cụm nến cùng màu phản ánh hiện tượng *short-term momentum*, trong khi các chuỗi kéo dài thường đi kèm dấu hiệu suy yếu hoặc cạn kiệt lực (*exhaustion*), làm cơ sở cho giả thuyết thị trường có trí nhớ ngắn hạn trong nhóm feature *Patterns & Streaks*.

Điểm cốt lõi của nhóm feature này nằm ở chỗ: **chúng không áp đặt trước diễn giải “đúng” hay “sai”.**

- Một chuỗi tăng ngắn có thể là tín hiệu momentum.
- Một chuỗi tăng quá dài có thể là tín hiệu cạn kiệt.

Việc quyết định diển giải nào phù hợp được để cho XGB Residual và Pricing Layer tự học dựa trên các feature khác (volatility, distance-to-mean, volume).

Như vậy, nhóm feature *Patterns & Streaks* cho phép mô hình:

- “nhớ” được quá khứ rất gần của thị trường,
- nhưng vẫn giữ tính linh hoạt, không ép buộc thị trường luôn phải momentum hay luôn phải đảo chiều.

Đây chính là hiện thân cụ thể của **Giả thuyết 5** trong pipeline: thị trường có trí nhớ ngắn hạn, và mô hình cần được cung cấp thông tin về trí nhớ đó để đưa ra dự báo dài hạn ổn định hơn trong vùng T+100.

Tổng kết

Năm giả thuyết trên không nhằm bao phủ mọi khía cạnh của thị trường, mà xác lập **khung tư duy thiết kế Feature Engineering** cho kiến trúc Hybrid + Pricing.

Mỗi giả thuyết trả lời một câu hỏi bản chất: giá được hình thành như thế nào, sai số lớn đến từ đâu, dòng tiền đứng phia nào, tâm lý thị trường được phản ánh ra sao, và liệu thị trường có ghi nhớ những chuyển động ngắn hạn hay không. Từ các giả định này, các nhóm feature tương ứng được suy ra một cách có định hướng, thay vì được phát hiện tình cờ.

Với vai trò đó, **EDA không còn là nơi “phát minh” feature**, mà trở thành công cụ *kiểm chứng ngược* cho các giả thuyết đã được đặt ra từ trước. EDA được sử dụng để trả lời các câu hỏi sau: các cấu trúc giả định có thực sự tồn tại trong dữ liệu FPT hay không, chúng có ổn định theo thời gian không, và những giới hạn của từng giả thuyết nằm ở đâu.

Do đó, Chương 5 sẽ trình bày EDA với đúng vai trò của nó trong pipeline: **kiểm chứng giả thuyết và đánh giá độ phù hợp của feature**, chứ không phải là cơ chế sinh ra feature mới một cách ngẫu hứng từ dữ liệu.

5 EDA

5.1 Vai trò của EDA

Trong pipeline dự báo giá cổ phiếu, **Exploratory Data Analysis (EDA) không phải là công cụ để “sáng tác” feature**. Các giả thuyết về cấu trúc thị trường đã được xác lập trước đó (Chương 3) mới là nền tảng định hướng cho Feature Engineering.

Với vai trò đúng của mình, EDA được sử dụng nhằm:

- **Kiểm chứng giả thuyết**: xác nhận các cấu trúc như trend, volatility clustering, momentum hay hành vi nến có thực sự tồn tại trong dữ liệu FPT hay không.
- **Đánh giá sức mạnh thống kê của feature**: kiểm tra xem feature có tín hiệu đủ rõ ràng, ổn định theo thời gian hay chỉ là nhiễu ngẫu nhiên.
- **Phát hiện redundancy và nguy cơ leakage**: nhận diện các feature trùng lặp thông tin, hoặc vô tình sử dụng thông tin tương lai.
- **Hỗ trợ quyết định giữ – bỏ feature**: những thành phần không có giá trị gia tăng sẽ bị loại bỏ hoặc giao toàn bộ việc xử lý cho mô hình học máy.

Điểm mấu chốt là:

EDA không trả lời “tao thêm feature gì”, mà trả lời “feature này có đáng tồn tại hay không”.

Do đó, các phân tích EDA trong Chương 5 luôn được trình bày bám sát từng giả thuyết đã đặt ra, với mục tiêu kiểm chứng và đánh giá độ phù hợp của feature, thay vì phát sinh các feature mới một cách ngẫu hứng từ dữ liệu.

5.2 Từ giả thuyết → Feature → Kiểm chứng

PHẦN 2: TRÁI TIM CỦA MÔ HÌNH (The Core Engine)

1 Kiến trúc Pipeline 3 lớp

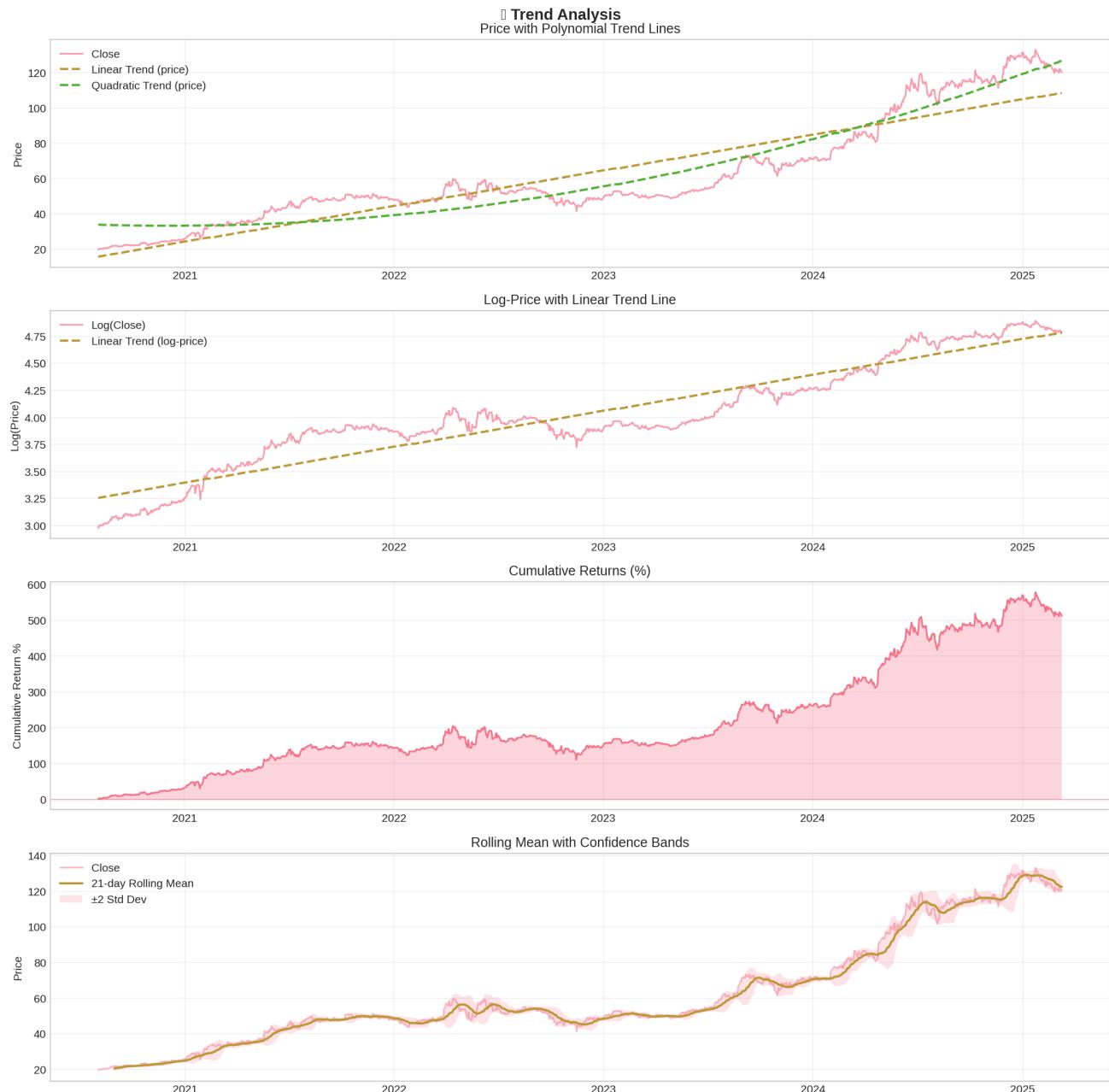
1.1 Lớp 1: Math Backbone (Trend)

Ngay từ những bước đầu xây dựng pipeline dự báo 100 ngày cho cổ phiếu FPT, một câu hỏi cứ trở đi trở lại: liệu thị trường có thật sự hỗn loạn như vẻ bề ngoài, hay đằng sau những biến động từng phiên vẫn tồn tại một quỹ đạo dài hạn mà ta có thể nắm bắt? Khi quan sát chuỗi giá nhiều năm, đặc biệt qua các biểu đồ tổng quan giá, candlestick, và đường trung bình động, điều hiện lên không phải là sự ngẫu nhiên tuyệt đối. Thay vào đó, có một dòng chảy mượt mà chạy xuyên suốt thời gian: giá FPT liên tục đi lên với nhịp độ khá đều đặn. Khi chuyển sang thang log-price 9, đường xu hướng ấy thậm chí còn trở nên gần như tuyến tính, việc này thể hiện một dấu hiệu mạnh mẽ rằng tăng trưởng dài hạn là cấu trúc nền của chuỗi giá.

Chính từ trực giác đó, **Math Backbone** được sinh ra: một bộ khung toán học đơn giản nhưng đóng vai trò “cột sống” giữ cho dự báo dài hạn không trôi theo nhiều động ngẫu hàn.

1.1.1 Từ log-price đến đường xu hướng dài hạn

Biểu đồ decomposition của chuỗi thời gian cho thấy phần *Trend* mượt, ổn định và tăng trưởng rõ rệt, trong khi phần seasonal lặp lại theo chu kỳ và phần residual dao động không có quy luật. Điều đó gợi ý rằng nếu ta tách riêng phần xu hướng và mô tả nó bằng một mô hình đơn giản, mô hình học máy sẽ có nhiều “không gian” hơn để học phần residual phức tạp.



Hình 9: Trend Analysis

Trong code, bước đầu tiên là chuyển giá đóng cửa sang hệ log:

$$\log(P_t) = \log(\text{close}_t + \varepsilon)$$

với ε rất nhỏ để tránh lỗi số. Sau đó, ta fit một mô hình tuyến tính đơn giản:

$$\log(P_t) \approx at + b,$$

với $t = 0, 1, 2, \dots$. Điều thú vị là trong biểu đồ Trend Analysis, mô hình tuyến tính này khớp rất tốt với log-price thực tế — xác nhận rằng backbone tuyến tính là lựa chọn hợp lý.

1.1.2 Mở rộng xu hướng vào tương lai: định nghĩa future_ret_math

Khi đường xu hướng quá khứ đã được học, ta kéo dài nó sang tương lai thêm $H = 100$ ngày. Từ đó, log-return theo xu hướng được định nghĩa:

$$\text{future_ret_math}(k) = \widehat{\log P}_{t_0+k+1} - \widehat{\log P}_{t_0+k},$$

với t_0 là ngày cuối của dữ liệu lịch sử.

Những giá trị này đóng vai trò như “nhịp tim dài hạn” của thị trường: đều đặn, ổn định, và hoàn toàn không bị chi phối bởi các cú sốc ngẫu nhiên. Từ biểu đồ phân phối (Distribution Histogram) và đồ thị Q–Q Plot, có thể thấy rằng chuỗi lợi nhuận ngày (daily returns) của FPT sai lệch đáng kể so với phân phối chuẩn. Histogram kết hợp KDE cho thấy phần lớn giá trị lợi nhuận tập trung quanh mức 0, nhưng hai đuôi của phân phối lại kéo dài bất thường. Hiện tượng heavy-tail này phản ánh các cú sốc thị trường, biến động do tin tức, hoặc những phiên giao dịch thanh khoản thấp tạo ra các thay đổi đột ngột. Đồ thị Q–Q giúp làm rõ hơn đặc điểm này 10: các quantile ở vùng trung tâm bám khá sát đường tham chiếu chuẩn, nhưng càng về hai đầu thì các điểm dữ liệu càng lệch mạnh, cho thấy sự xuất hiện của nhiều outlier và các sự kiện cực trị mà mô hình khó học một cách ổn định.

Trong bối cảnh đó, **Math Backbone** đóng vai trò như một “lời toán học sạch” giúp ổn định cấu trúc dài hạn của chuỗi giá. Hoạt động trực tiếp trên chuỗi log-price backbone tách được hai thành phần riêng biệt của quá trình hình thành giá:

- **Xu hướng dài hạn mượt**, phản ánh quỹ đạo tăng trưởng mà thị trường duy trì trong nhiều năm.
- **Biến động nhiễu ngắn hạn**, chính là nguồn gốc của heavy-tail và các giá trị cực trị trong phân phối returns.

Khi trực quan hóa đường xu hướng log-price cùng biểu đồ giá và phân phối lợi nhuận, ta nhận thấy backbone loại bỏ hầu hết méo dạng do các giá trị cực trị gây ra, tạo nên một phiên bản “clean return” mô tả phần drift cấu trúc mà mô hình dự báo nên tin tưởng. Điều này đặc biệt quan trọng đối với các bài toán dự báo dài hạn dạng đệ quy (ví dụ: 100 ngày), nơi mà độ nhạy quá mức với tail có thể khiến đường dự báo trở nên giật mạnh và phi thực tế.

Nhờ việc loại bỏ cấu trúc dài hạn thông qua backbone, mô hình học máy chỉ cần học phần *residual*, vốn ít nhiễu hơn, gần với phân phối chuẩn hơn và dễ mô hình hóa hơn rất nhiều. Kết quả là pipeline dự báo trở nên ổn định hơn, dễ giải thích hơn và phù hợp hơn với bản chất thống kê của dữ liệu tài chính.

1.1.3 Khi câu chuyện toán học bước vào thực thi code

Tất cả ý tưởng trên được hiện thực hoá trong hàm `build_raw_base_path_hybrid`.

Dưới đây là đoạn code cốt lõi cho phần Math Backbone:

```

1 # 1) Fit a linear trend on log-price
2 df_state["close_log"] = np.log(df_state["close"] + 1e-8)
3 N_hist = len(df_state)
4
5 time_idx_hist = np.arange(N_hist).reshape(-1, 1)
6 y_log_hist = df_state["close_log"].values.reshape(-1, 1)
7
8 lr_trend = LinearRegression()
9 lr_trend.fit(time_idx_hist, y_log_hist)
10
11 # 2) Extend the trend into the future
12 total_len = N_hist + total_days
13 time_idx_full = np.arange(total_len).reshape(-1, 1)

```

```

14
15 trend_log_full = lr_trend.predict(time_idx_full).flatten()
16
17 # 3) Compute math backbone returns
18 math_rets_forecast = np.zeros(total_days, dtype=float)
19 for k in range(total_days):
20     base_idx = N_hist - 1 + k
21     if base_idx + 1 < len(trend_log_full):
22         math_rets_forecast[k] = (
23             trend_log_full[base_idx + 1] - trend_log_full[base_idx]
24         )
25     else:
26         # fallback: repeat last return
27         math_rets_forecast[k] = math_rets_forecast[k-1] if k > 0 else 0.0

```

Code Listing 1: Math Backbone implementation excerpt

Ba bước này tương ứng hoàn toàn với câu chuyện toán học đã trình bày:

- 1. Fit xu hướng dài hạn trên log-price
- 2. Kéo dài xu hướng vào tương lai
- 3. Tính chuỗi log-return theo xu hướng

Và chính `math_rets_forecast[k]` là hiện thân của `future_ret_math(k)` trong code.

1.1.4 Backbone kết hợp với residual: từ lý thuyết đến đường giá dự báo

Trong vòng lặp dự báo từng ngày, backbone không hoạt động độc lập. Mỗi bước, mô hình:

- 1. Tạo các đặc trưng (bao gồm STL features, volatility, technical indicators)
- 2. Chuẩn hoá đặc trưng
- 3. XGBoost dự đoán residual return
- 4. Kết hợp backbone + residual:

$$\text{final_ret}(k) = \text{math_ret}(k) + \text{resid_pred}(k)$$

Sau đó giá được cập nhật:

$$\log P_{t+1} = \log P_t + \text{final_ret}(k).$$

Phần code tương ứng:

```

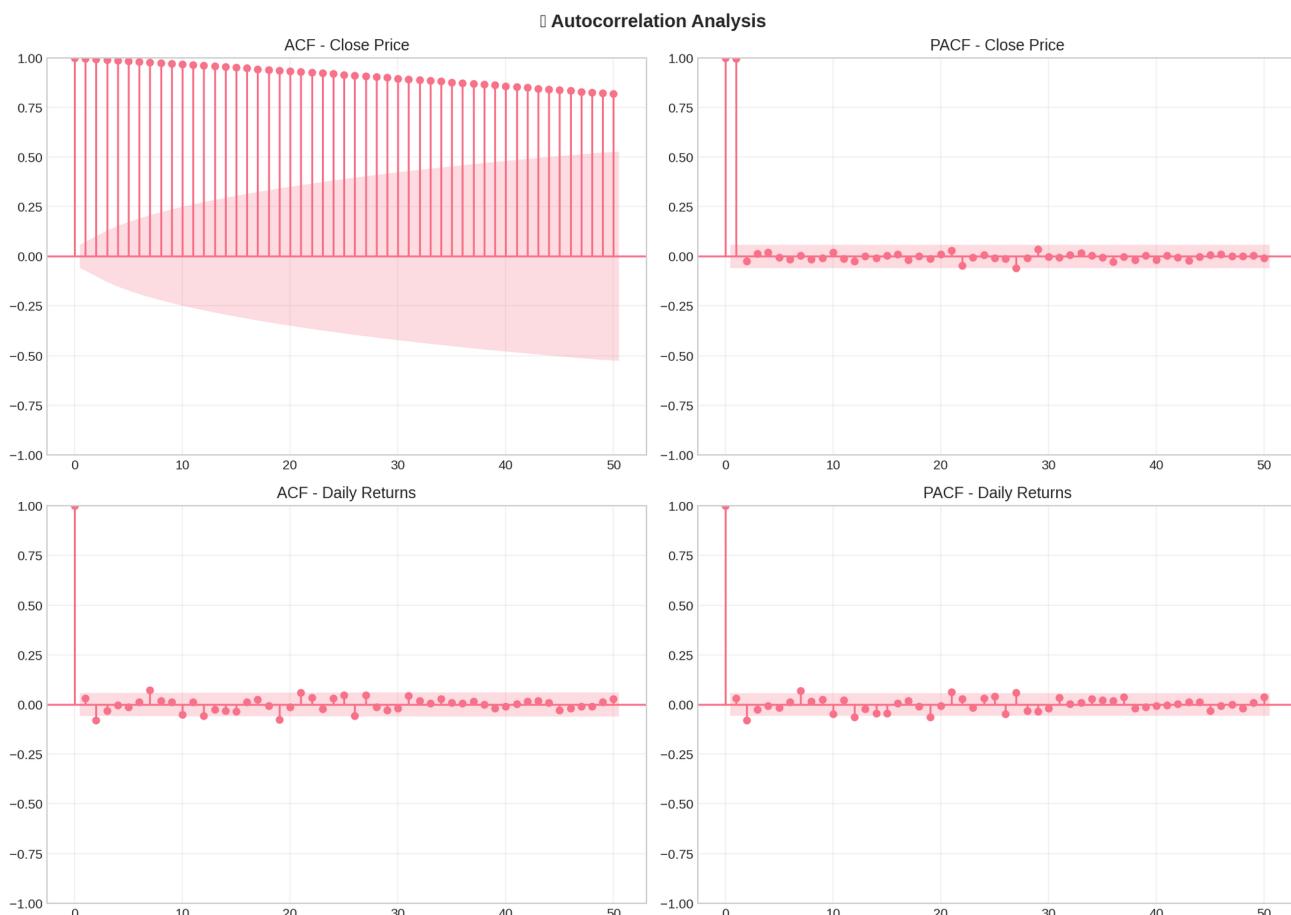
1 X_last_s = scaler_X.transform(X_last)
2 resid_pred = float(xgb.predict(X_last_s)[0])
3
4 math_ret = float(math_rets_forecast[step_idx])
5 final_ret = math_ret + resid_pred    # hybrid return
6
7 last_log = df_state["close_log"].iloc[-2]
8 next_log = last_log + final_ret
9 next_price = float(np.exp(next_log))

```

Code Listing 2: Combining math backbone and residual prediction

Điều này phản ánh chính xác bản chất thị trường ta quan sát từ các biểu đồ kỹ thuật:

- MACD cho thấy các giai đoạn momentum shift mạnh
- RSI nhiều lần chạm vùng quá mua/quá bán
- Rolling Volatility nhảy vọt tại các điểm thị trường nhạy cảm ư
- ACF/PACF của returns gần như nhiễu trắng. Quan sát trực tiếp trên các biểu đồ ACF/PACF trong Hình 11 cho thấy sự khác biệt rõ rệt giữa hành vi của giá đóng cửa và chuỗi lợi nhuận ngày. ACF của giá duy trì giá trị cao trong nhiều độ trễ liên tiếp, trong khi PACF chỉ có một spike mạnh tại lag 1. Mẫu hình này là đặc trưng của một chuỗi không dừng có xu hướng dài hạn, cũng có nhận định rằng log-price mang cấu trúc drift ổn định mà Math Backbone có thể mô tả hiệu quả. Ngược lại, ACF và PACF của daily returns đều nằm hoàn toàn trong khoảng tin cậy, không xuất hiện bất kỳ spike có ý nghĩa thống kê nào. Điều này chỉ ra rằng returns gần như là một chuỗi nhiễu trắng, không có tương quan tuyến tính qua thời gian. Vì returns không mang thông tin xu hướng, chúng không phải đối tượng phù hợp để dự báo trực tiếp. Do đó, việc tách phần cấu trúc dài hạn thông qua backbone trở thành bước thiết yếu, giúp mô hình học máy chỉ cần học phần residual ít nhiễu và dễ mô hình hóa hơn.



Hình 11: Autocorrelation Analysis

Những chuyển động này không thể được mô tả bằng backbone tuyến tính — nhưng residual learning lại có thể.

1.1.5 Kết luận: một đường thẳng giữa thế giới xao động

Math Backbone không phải là phần phức tạp nhất của pipeline, nhưng nó là phần ổn định nhất. Nó đại diện cho phần chuyển động mà ta có thể tin tưởng: xu hướng dài hạn, nhất quán, được thấy rõ trong các biểu đồ trend và decomposition. Tách xu hướng dài hạn cho backbone giúp mô hình học máy tập trung vào phần còn lại — phần nhiễu động giàu thông tin mà MACD, RSI, volatility và return distribution đều cho thấy là rất phi tuyến và khó đoán. Nhờ sự phân vai rõ ràng này, mô hình hybrid trở nên:

- Ổn định hơn backbone đơn thuần
- Chính xác hơn ML đơn thuần
- Và phản ánh đúng “hơi thở” của thị trường: một thế giới đầy xao động, nhưng vẫn có một dòng chảy dài hạn dẫn dắt.

1.2 Lớp 2: ML Residual

Khi nhìn vào toàn bộ bức tranh EDA của cổ phiếu FPT, ta nhận ra thị trường vốn dĩ không vận hành chỉ theo một quy luật duy nhất. Những đường giá mượt mà trong hình **Trend Analysis** cho ta thấy một quỹ đạo dài hạn khá rõ ràng: log-price gần như tuyến tính, và backbone có thể mô tả được xu hướng này một cách gọn ghẽ. Nhưng chỉ cần tiến lại gần hơn, quan sát những cây nến trong hình **Candlestick Overview** 12 hay biến động volume trong phần **Trading Volume**, ta lập tức cảm nhận một thế giới hoàn toàn khác: một nhịp điệu đầy những cú giật, cú rơi, cú bật, mà xu hướng dài hạn không thể giải thích nổi.



Hình 12: Candle Stick

Các biểu đồ phân phối trong Hình 10 càng khắc họa rõ sự hỗn độn này: daily returns có đuôi dài, lệch mạnh và chứa vô số outliers. Q-Q Plot thì gần như “rung rời” ở hai đầu, chứng tỏ phân phối thực tế

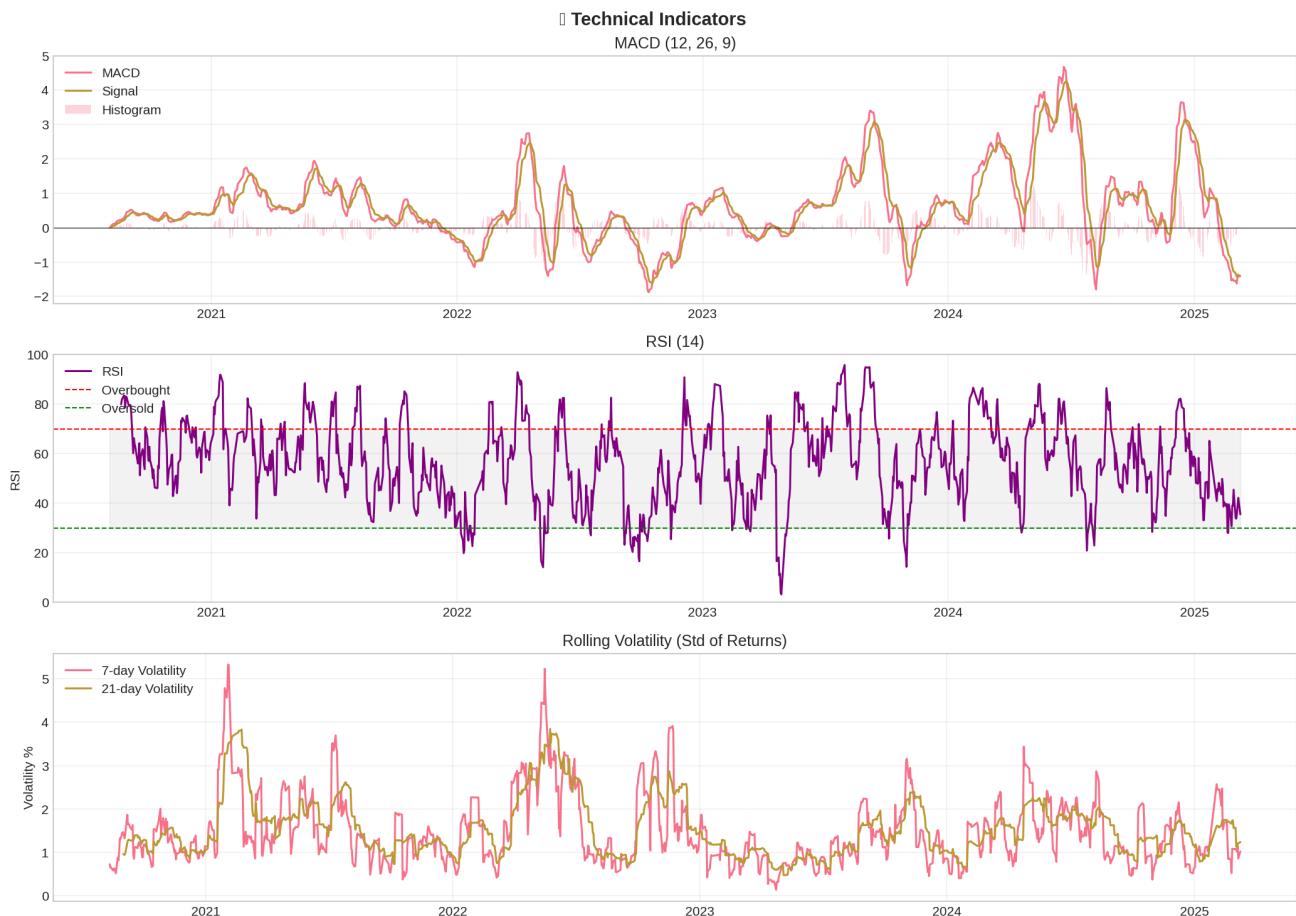
không tuân theo luật Gaussian cổ điển. Nhưng cú đánh mạnh nhất lại đến từ Hình 11: ACF và PACF của returns phẳng lặng như tờ, toàn bộ điểm dữ liệu rơi gọn vào vùng tin cậy. Nói cách khác, returns là một chuỗi nhiễu trắng với các đặc điểm không tự tương quan, không linear structure, không AR, không MA.

Điều này mang một hàm ý cực kỳ quan trọng: **không mô hình tuyến tính nào có thể dự báo trực tiếp returns**. Và đây chính là lý do ta phải tách xu hướng bằng Math Backbone trước khi làm bất kỳ điều gì khác.

Khi backbone đã bóc tách phần drift dài hạn, phần residual còn lại không còn là hỗn loạn vô nghĩa. Quan sát Hình 13, ta thấy residual chứa những nhịp điệu ngắn hạn, những cú xoay mean-reverting, và những đợt volatility clustering mà thị trường thường tạo ra theo tâm lý nhà đầu tư. Các chỉ báo kỹ thuật trong Hình 14 như RSI, MACD, volatility cũng cho thấy những dao động mạnh trong biên độ ngắn hạn và không đủ lớn để thay đổi trend, nhưng đủ quan trọng để ảnh hưởng đến dự báo 100 ngày khi thực hiện mô phỏng đề quy.



Hình 13: Decomposition



Hình 14: Technical Indicators

Đây chính là vùng đất của mô hình phi tuyến. Nhưng chọn mô hình nào?

1.2.1 Vì sao XGBoost phù hợp với bản chất dữ liệu residual hơn các mô hình khác?

Dữ liệu residual sau backbone không giống một time series truyền thống. Nó giống một tập hợp các quy tắc cục bộ (*local market behaviors*) hơn là một quá trình tạo sinh toàn cục. Và XGBoost có những đặc tính gần như sinh ra để học dạng dữ liệu này:

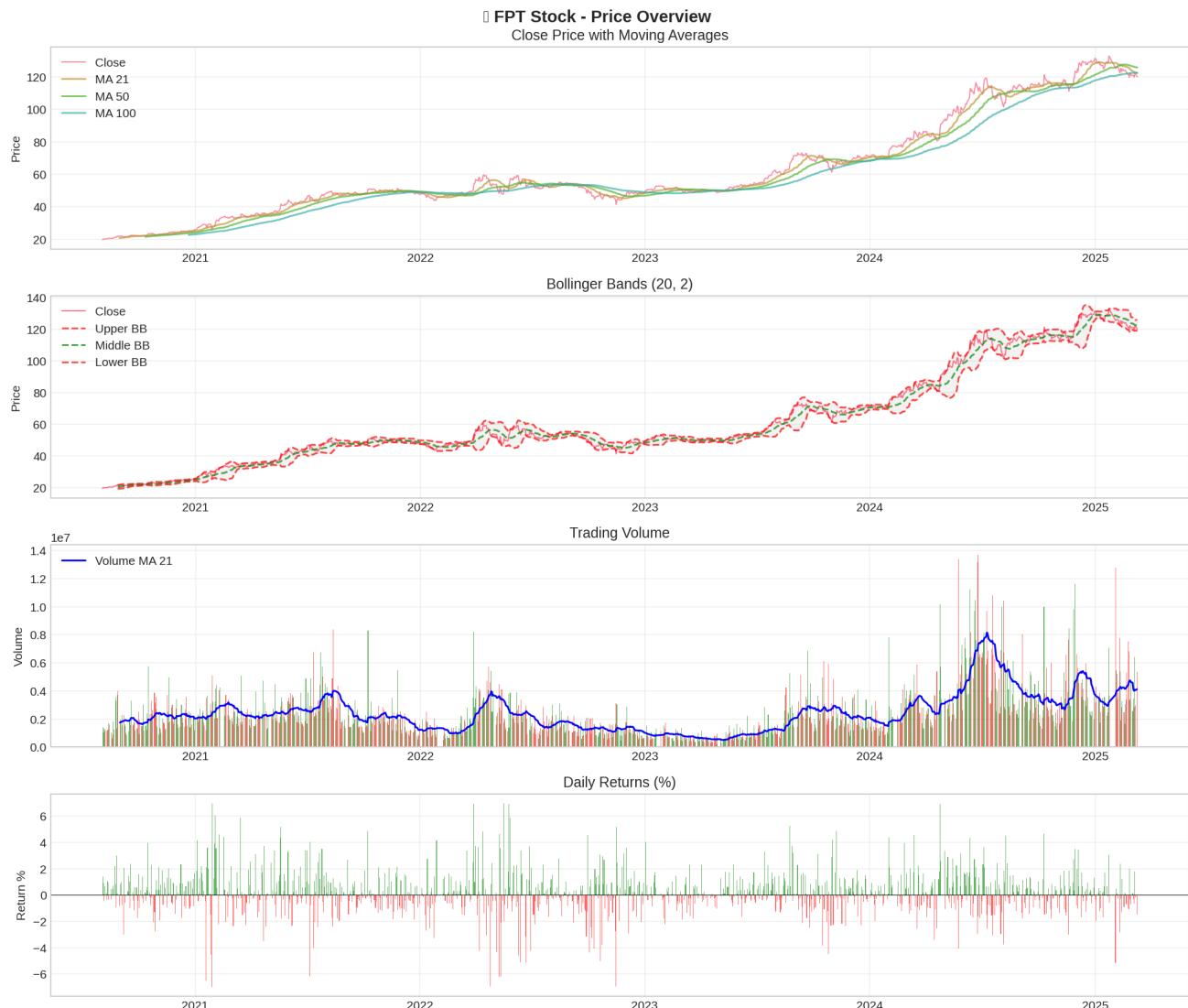
1. **XGBoost mô hình hóa mạnh mẽ các quan hệ phi tuyến.** Các cú giật giá, cú hồi, cú squeeze, volatility spike—không mô hình tuyến tính nào miêu tả nổi, nhưng XGBoost có thể phân tách không gian đặc trưng qua các cây quyết định và tự tìm ra những “luật nhỏ” kiểu:

if volatility_21d > threshold and RSI < 35 \Rightarrow residual dương trong 2–3 phiên

2. **XGBoost khai thác tương tác giữa nhiều feature một cách tự nhiên.** Correlation heatmap ở Hình 16 cho ta một thông điệp rất rõ: các biến trong bộ dữ liệu FPT gần như **không có quan hệ tuyến tính mạnh**. Điều này thể hiện qua việc hầu hết hệ số tương quan đều nằm trong khoảng từ -0.1 đến 0.6 , không có cặp biến nào “đi chung” theo một đường thẳng rõ rệt. Nhưng chính vì heatmap im lặng như vậy nên ta càng phải cẩn thận. Nếu nhìn kĩ volume không tương quan trực tiếp với daily return, nhưng khi nhìn vào chart volume trong Hình 15, ta thấy volume tăng mạnh thường đi kèm với tăng biến động và các mẫu nến mở rộng—tức là **volume ảnh hưởng**

đến momentum theo cách phi tuyến. Tương tự, high-low spread gần như không tương quan với giá đóng cửa nhưng histogram spread ở Hình 10 lại cho thấy các phiên có spread lớn thường là lúc thị trường hoảng loạn hoặc hưng phấn và chính những phiên như vậy lại tạo ra residual lớn trong mô hình xu hướng. Đây là dạng quan hệ:

if spread $> \tau \Rightarrow$ residual biến động mạnh,



Hình 15: Price Overview

mà không mô hình tuyến tính nào có thể mô tả bằng một hệ số tương quan duy nhất. Hay ngay cả RSI—nhìn heatmap thì gần như không liên hệ với giá hay volume. Nhưng biểu đồ RSI trong Hình 14 lại kể một câu chuyện khác: RSI phản ứng rất nhạy với các giai đoạn thị trường tăng tốc hoặc chững lại. Điều này tạo nên các ngữ cảnh “bán quá mức” và “mua quá mức” mà các mô hình tuyến tính hoàn toàn không thể nhận diện:

if RSI $< 30 \Rightarrow$ mean-reversion signal mạnh.

Những mối quan hệ kiểu điều kiện—"nếu A cao và B thấp thì C thay đổi mạnh"—không khi nào xuất hiện trong heatmap, nhưng lại vô cùng quan trọng đối với thị trường tài chính. Và đây chính là nơi XGBoost vượt trội: mỗi cây quyết định của nó chính là một "bộ tách phi tuyến" dễ dàng tạo ra các ngưỡng:

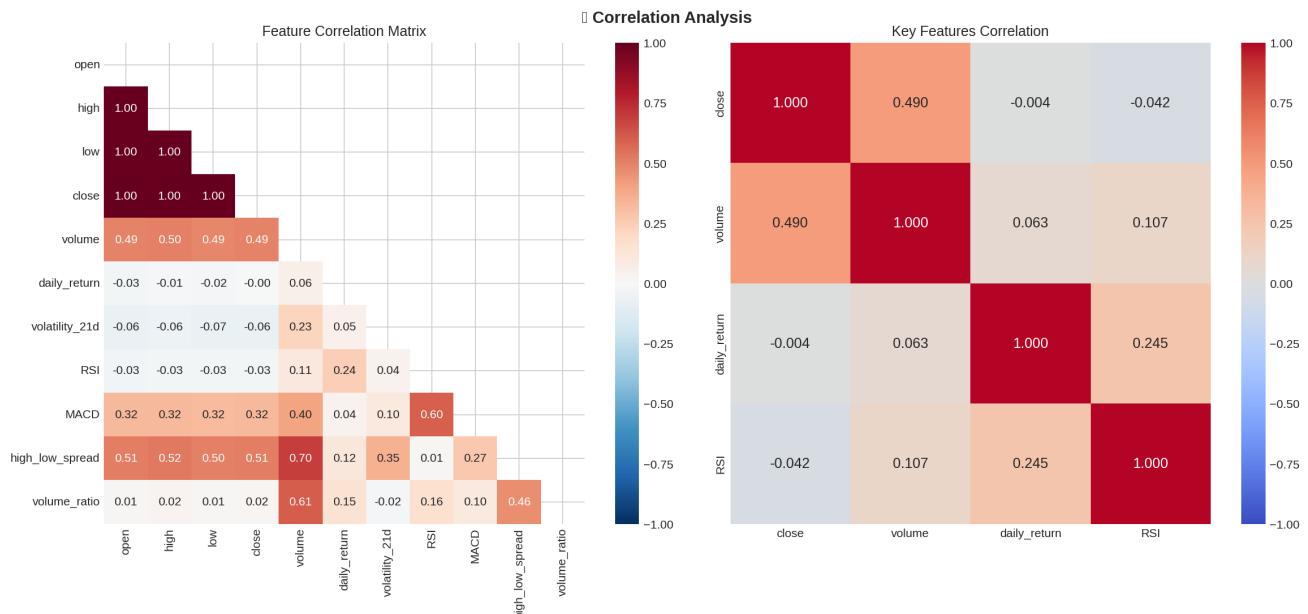
if volume spike and volatility rising \Rightarrow residual dương,

hoặc:

if momentum giảm nhưng giá vẫn trên MA21 \Rightarrow residual âm nhẹ.

XGBoost không cố ép dữ liệu tài chính vào một khung gian tuyến tính như hồi quy. Thay vào đó, nó tự tìm ra các tương tác phi tuyến mà heatmap không hề thể hiện và biến chúng thành những luật cục bộ có tính dự báo. Chính nhờ khả năng xử lý các interaction splits này mà XGBoost phù hợp tự nhiên với cấu trúc dữ liệu residual—một tập hợp nhỏ tinh tế của hàng loạt "tín hiệu phi tuyến" mà thị trường tạo ra mỗi ngày.

3. **Hoàn hảo cho dữ liệu short-term, noisy, local.** Candlestick chart Hình 12 cho thấy từng phiên có hành vi rất "địa phương": gap nhỏ, breakout, false break, doji, long wick. Đây không phải loại dữ liệu bạn đưa vào LSTM hay ARIMA rồi mong nó tự hiểu. Nhưng XGBoost lại cực kỳ giỏi trong việc nắm bắt local rules.
4. **Không yêu cầu stationarity.** Backbone đã làm phẳng phần trend, residual đôi khi vẫn hơi lệch mean hoặc biến thiên theo thời kỳ. ARIMA hoặc VAR sẽ thất bại ngay tại cửa kiểm tra stationarity. XGBoost thì không bận tâm.
5. **Chịu nhiễu tốt và ít overfit hơn neural networks trên dataset vừa và nhỏ.** Dữ liệu FPT dù kéo dài nhiều năm nhưng ở granular daily không đủ lớn cho deep learning. XGBoost vừa đủ mạnh, vừa đủ "thận trọng".



Hình 16: Correlation Heatmap

1.2.2 Câu chuyện cuối cùng: sự kết hợp giữa cái lớn và cái nhỏ.

Trend của log-price—như ta thấy trong hình Trend Analysis—chạy rất mượt. Residual thì lại nhảy múa trong biên độ nhỏ mỗi ngày. Một mô hình tốt cần mô tả được **cả hai thế giới**. Math Backbone giữ phần “lõi” lớn: quỹ đạo tăng trưởng dài hạn. XGBoost giữ phần “hồn” nhỏ: dao động tinh tế mà trader nào nhìn chart cũng thấy. Và chính sự kết hợp này tạo ra một pipeline dự báo vừa ổn định và lại vừa nhạy bén, điều mà bất kỳ mô hình đơn lẻ nào cũng không làm được.

1.2.3 Thực thi mô hình XGBoost trên residual: từ ý tưởng đến code

Sau khi xây dựng Math Backbone và tạo được chuỗi residual

$$r_t^{\text{resid}} = r_t^{\text{future}} - r_t^{\text{math}},$$

bước tiếp theo là huấn luyện một mô hình phi tuyến để học phần nhiễu có cấu trúc còn sót lại. Dựa trên những quan sát từ EDA (phân phối heavy-tail, ACF/PACF gần nhiều trảng, các tương tác phi tuyến giữa volatility, volume, RSI, spread trong heatmap tương quan), Ta triển khai XGBoost để mô hình hóa trực tiếp residual:

```

1 def train_xgb_on_dfmodel(df_model: pd.DataFrame, feature_cols: List[str]) -> Tuple[XGBRegressor, StandardScaler, float, float]:
2     """
3     Train XGBoost on residual returns: resid_ret = future_ret - math_ret.
4     """
5     if len(df_model) < 200:
6         raise ValueError(f"[XGB] Too few samples: {len(df_model)}")
7
8     # 1) Extract features and residual target
9     X_all = df_model[feature_cols].values.astype(np.float32)
10    y_resid = df_model[["resid_ret"]].values.astype(np.float32)
11
12    # 2) Time-ordered split: 80% train, 10% val, 10% test
13    N = len(df_model)
14    train_ratio = 0.8
15    val_ratio = 0.1
16
17    train_end = int(N * train_ratio)
18    val_end = int(N * (train_ratio + val_ratio))
19
20    X_train = X_all[:train_end]
21    y_train = y_resid[:train_end]
22    X_val = X_all[train_end:val_end]
23    y_val = y_resid[train_end:val_end]
24    X_test = X_all[val_end:]
25    y_test = y_resid[val_end:]
26
27    # 3) Standardize inputs (fit only on train)
28    scaler_X = StandardScaler().fit(X_train)
29    X_train_s = scaler_X.transform(X_train)
30    X_val_s = scaler_X.transform(X_val)
31    X_test_s = scaler_X.transform(X_test)
32
33    # 4) Configure and train XGBoost on residuals
34    xgb = XGBRegressor(
35        n_estimators=450,
36        max_depth=4,
37        learning_rate=0.03,
38        subsample=0.9,
39        colsample_bytree=0.9,
```

```

40     reg_lambda=2.0,
41     min_child_weight=3,
42     objective="reg:squarederror",
43     random_state=SEED,
44 )
45 xgb.fit(X_train_s, y_train)
46
47 # 5) Evaluate on train / val / test using MSE and MAE
48 def eval_block(name, X_s, y_true):
49     if len(y_true) == 0:
50         print(f"{name}: empty")
51         return
52     y_pred = xgb.predict(X_s)
53     mse = mean_squared_error(y_true, y_pred)
54     mae = mean_absolute_error(y_true, y_pred)
55     print(f"{name} (resid) -> MSE: {mse:.6e} | MAE: {mae:.6e}")
56
57 print(f"==== XGB RESIDUAL MODEL ({len(df_model)} samples) ====")
58 eval_block("Train", X_train_s, y_train)
59 eval_block("Val", X_val_s, y_val)
60 eval_block("Test", X_test_s, y_test)
61
62 # 6) Compute residuals after XGB to measure remaining noise
63 y_pred_all = xgb.predict(scaler_X.transform(X_all))
64 residuals = y_resid - y_pred_all
65 resid_std = float(np.std(residuals, ddof=1))
66 resid_mean = float(np.mean(residuals))
67
68 print(f"[RESID] std={resid_std:.6e}, mean={resid_mean:.6e}")
69 return xgb, scaler_X, resid_std, resid_mean

```

Code Listing 3: Training XGBoost on residual returns

Đoạn code trong Listing 3 hiện thực hoá chính xác ý tưởng “backbone giữ xu hướng, XGBoost bắt nhiễu phi tuyến”:

- **Bước 1–2:** Chuỗi residual `resid_ret` được dùng làm mục tiêu huấn luyện, trong khi các feature đầu vào được trích từ những đặc trưng giàu ý nghĩa mà EDA đã gợi ý (volatility, volume, RSI, spread, ...). Việc chia theo thứ tự thời gian (80%–10%–10%) phản ánh đúng bản chất dự báo trong thực tế: mô hình chỉ được phép học từ quá khứ.
- **Bước 3:** Chuẩn hoá feature bằng `StandardScaler` giúp XGBoost ổn định hơn khi các biến có đơn vị và thang đo rất khác nhau (giá, phần trăm, khối lượng).
- **Bước 4:** Cấu hình XGBoost với nhiều cây nhỏ (`max_depth=4`) cho phép mô hình nắm bắt các tương tác phi tuyến cục bộ giữa RSI, volatility, volume, high-low spread như những gì quan sát được trong các biểu đồ kỹ thuật và heatmap tương quan.
- **Bước 5:** Việc in ra MSE/MAE cho train/validation/test không chỉ dùng để chọn hyperparameter mà còn là cách kiểm tra xem mô hình có thực sự khai thác được cấu trúc còn lại trong residual hay không. Nếu XGBoost chỉ gặp nhiễu trắng thuần túy, sai số trên tập test sẽ gần tương đương với một mô hình “đoán ngẫu nhiên”.
- **Bước 6:** Sau khi trừ đi dự báo của XGBoost, ta tính lại độ lệch chuẩn và trung bình của residual mới. Nếu std giảm đáng kể và mean ≈ 0 điều đó cho thấy mô hình đã loại bỏ phần nhiễu có cấu trúc (volatility clustering, hiệu ứng RSI, volume spike, ...), để lại một chuỗi residual gần hơn với nhiễu trắng thực sự. Chuỗi này sau đó sẽ được sử dụng trong lớp Pricing Layer như một proxy cho “rủi ro còn lại”.

1.3 Lớp 3: Pricing Layer

Sau khi đi qua hai lớp đầu tiên, mô hình cho ta:

- $Trend_t$: xu hướng dài hạn (Math Backbone),
- $Base_t$: dự báo “thô” sau khi đã cộng residual do mô hình ML học được.

Tuy nhiên, đường $Base_t$ đôi khi có thể quá “hung hăng”: tăng/giảm quá mạnh chỉ trong vài phiên, hoặc trôi xa khỏi xu hướng dài hạn. Vì vậy, thay vì dùng trực tiếp $Base_t$, chúng tôi xây dựng một lớp kiểm soát thứ ba – *Pricing Layer* – đóng vai trò như một “bộ điều khiển vật lý” đặt lên đường giá.

Pricing Layer mô phỏng ba cơ chế vật lý trực quan:

1. **Clipping** – giới hạn tốc độ thay đổi giá,
2. **Damping** – lực ma sát làm giảm biên độ rung lắc,
3. **Mean Reversion** – lực đòn hồi kéo giá về một “neo” cân bằng (thường là xu hướng dài hạn).

Ký hiệu $Price_t$ là giá đã được Pricing Layer kiểm soát tại thời điểm t . Mục tiêu của chúng tôi là xây dựng một quy tắc cập nhật từ $Price_{t-1}$ sang $Price_t$ sao cho đường giá cuối cùng vừa phản ánh được “ý kiến” của mô hình ML, vừa tuân thủ các ràng buộc hợp lý về tốc độ, độ mượt và xu hướng thị trường.

1.3.1 Bước nhảy thô từ mô hình ML

Xuất phát từ dự báo “thô” của mô hình:

$$\Delta_t^{\text{raw}} = Base_t - Price_{t-1}. \quad (2)$$

Ở đây, Δ_t^{raw} có thể được hiểu như “bước nhảy” mà mô hình ML đề xuất cho ngày t . Nếu chỉ đơn giản đặt $Price_t = Base_t$, đường giá sẽ hoàn toàn phó thác cho mô hình ML, dễ bị nhiễu và overfit.

1.3.2 Clipping: Giới hạn tốc độ thay đổi giá

Trong thực tế, giá cổ phiếu khó có thể tăng/giảm vô hạn trong một phiên; bản thân thị trường (và quy định biên độ) đã đóng vai trò “phanh”. Ta mô hình hóa điều này bằng một ngưỡng vận tốc tối đa v_{\max} 0, và tiến hành cắt (clip) bước nhảy:

$$\Delta_t^{\text{clip}} = \max(-v_{\max}, \min(\Delta_t^{\text{raw}}, v_{\max})). \quad (3)$$

Ý nghĩa:

- Nếu $|\Delta_t^{\text{raw}}| \leq v_{\max}$ thì $\Delta_t^{\text{clip}} = \Delta_t^{\text{raw}}$,
- Nếu mô hình đề xuất một bước nhảy quá lớn, nó sẽ bị ép lại về $\pm v_{\max}$.

Nhờ đó, Pricing Layer đảm bảo rằng đường giá không thể “nhảy điên loạn” chỉ vì một vài điểm dữ liệu nhiễu.

1.3.3 Damping: Lực ma sát chống rung lắc

Clipping kiểm soát các bước nhảy quá lớn trong một ngày, nhưng không ngăn được hiện tượng đường giá rung mạnh nếu mô hình ML liên tục “đổi ý”. Để giảm hiện tượng zig-zag, chúng tôi đưa vào một hệ số *damping* $\lambda \in (0, 1]$, làm giảm biên độ tác động của bước nhảy mới:

$$\Delta_t^{\text{dyn}} = \lambda \Delta_t^{\text{clip}}. \quad (4)$$

Khi đó:

- λ càng nhỏ \Rightarrow ma sát càng lớn, đường giá phản ứng chậm hơn và mượt hơn,
- λ càng gần 1 \Rightarrow đường giá gần như tin hoàn toàn vào đề xuất mới.

Trong trường hợp muốn mô hình hóa “nhớ quá khứ” mạnh hơn, ta có thể dùng dạng làm trơn hàm mũ:

$$\Delta_t^{\text{dyn}} = (1 - \lambda) \Delta_{t-1}^{\text{dyn}} + \lambda \Delta_t^{\text{clip}}, \quad (5)$$

tương đương với một bộ lọc EMA áp lên chuỗi bước nhảy.

1.3.4 Mean Reversion: Lực đàm hồi kéo về xu hướng

Ngay cả khi đã clipping và damping, đường giá vẫn có thể trôi dần xa khỏi xu hướng dài hạn $Trend_t$ nếu mô hình ML có thiên lệch lạc quan hoặc bi quan. Để tránh hiện tượng “trôi vô hạn”, chúng tôi bổ sung một lực đàm hồi kéo giá về neo xu hướng:

$$Price_t = Price_{t-1} + \Delta_t^{\text{dyn}} + \beta(Trend_t - Price_{t-1}), \quad (6)$$

với $\beta \in [0, 1]$ là hệ số *mean reversion*. Ở đây:

- Nếu β lớn, khoảng cách giữa $Price_t$ và $Trend_t$ được rút ngắn nhanh, giá khó có thể chạy quá xa khỏi xu hướng,
- Nếu β nhỏ, mô hình cho phép giá “lang thang” xa trend lâu hơn, phù hợp với các giai đoạn thị trường có xu hướng mạnh.

Một cách trực giác, có thể liên hệ β với *half-life* h (số phiên cần để khoảng cách giảm còn một nửa). Với xấp xỉ:

$$\beta \approx 1 - e^{-\ln 2/h}, \quad (7)$$

ta có thể chọn h theo trực giác (ví dụ: “mất khoảng 20 phiên thì giá bị kéo về gần trend”) rồi suy ra β tương ứng.

1.3.5 Tổng hợp Pricing Layer dưới dạng thuật toán

Gộp ba cơ chế trên, quy tắc cập nhật cho Pricing Layer tại thời điểm t có thể viết gọn như sau:

$$\Delta_t^{\text{raw}} = Base_t - Price_{t-1}, \quad (8)$$

$$\Delta_t^{\text{clip}} = \max(-v_{\max}, \min(\Delta_t^{\text{raw}}, v_{\max})), \quad (9)$$

$$\Delta_t^{\text{dyn}} = \lambda \Delta_t^{\text{clip}}, \quad (10)$$

$$Price_t = Price_{t-1} + Delta_t^{\text{dyn}} + beta(Trend_t - Price_{t-1}). \quad (11)$$

Trong cài đặt thực nghiệm, các tham số ($v_{\max}, \lambda, \beta, \dots$) không được chọn thủ công mà được tối ưu tự động bằng *Random Search* kết hợp với *Cross-Validation theo thời gian*. Điều này đảm bảo rằng “tính cách” của Pricing Layer (mạnh tay hay thận trọng, quay về trend nhanh hay chậm) được hiệu chỉnh phù hợp với dữ liệu lịch sử của cổ phiếu FPT, thay vì chỉ dựa vào cảm tính.

2 Deep Dive vào Pricing Layer

2.1 Cơ chế vật lý

2.1.1 Clipping

Clipping là cơ chế đầu tiên trong Pricing Layer, đóng vai trò giống như **giới hạn tốc độ** trong một hệ động lực. Ý tưởng rất đơn giản: dù mô hình ML (Hybrid Backbone + XGBoost) có dự đoán một cú nhảy giá rất lớn, ta vẫn đặt câu hỏi:

“Trên thực tế, FPT có thể di chuyển nhanh đến mức đó chỉ trong một phiên không?”

Trên thị trường chứng khoán Việt Nam, đặc biệt với cổ phiếu vốn hóa lớn như FPT, biến động ngày thường nằm trong vùng 1–3%, các cú sốc 5–7% đã là cực đoan và thường gắn với tin tức rất lớn. Do đó, nếu để mô hình ML tự do trả ra các log-return tương đương với +15% hoặc -20% một ngày, dự báo sẽ trở nên *phi thực tế*.

Trong Pricing Layer, cơ chế Clipping được hiện thực hoá bằng đoạn mã:

```
1 # raw_ret: log-return do hybrid model sinh ra
2 pred_ret = np.clip(raw_ret, -ret_clip, ret_clip)
```

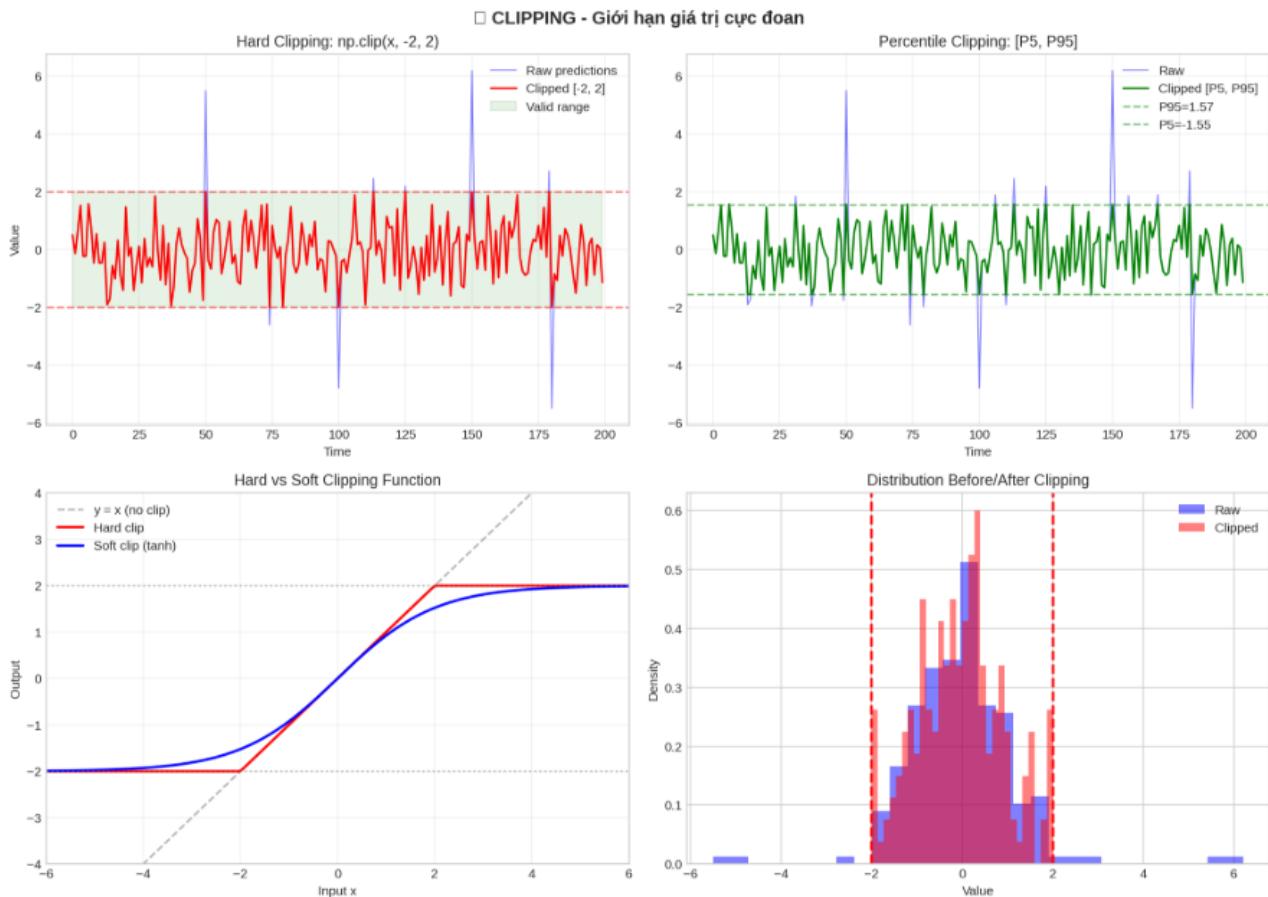
Trong đó, `ret_clip` không cố định mà phụ thuộc vào:

- **Regime thị trường (BULL / BEAR / SIDEWAYS):** Trong BULL, thị trường thường chấp nhận những cú tăng mạnh hơn, nên biên dương có thể được nới rộng nhẹ. Trong BEAR, các cú sụt giảm mạnh lại phổ biến hơn, nhưng ta vẫn hạn chế để tránh mô hình dự đoán các pha sập giá phi lý.
- **Mức độ biến động hiện tại (vol_ratio):** Nếu volatility 20 ngày gần đây thấp hơn nhiều so với lịch sử, không có lý do gì để cho phép mô hình “nhảy” với biên độ quá lớn.

Ta có thể viết dưới dạng công thức:

$$\text{ret_clip} = \text{base_ret_clip} \times \text{clip_scale_regime} \times \text{clip_scale_vol}.$$

Hình 17 minh họa ý nghĩa của Clipping: đường màu xám thể hiện các log-return mà mô hình ML *muốn* dự đoán, trong khi đường màu xanh dương là log-return *sau khi đã được clip*, với đỉnh/dáy bị cắt bớt.



Hình 17: Minh họa cơ chế Clipping: log-return cực đoan của mô hình được “cắt” về một ngưỡng biên độ hợp lý hơn.

Về mặt trực giác, Clipping giống như việc *gắn bô giới hạn tốc độ* lên dự báo:

- Mô hình vẫn được phép phản ứng với thông tin mới (giá có thể tăng/giảm nhanh hơn trong BULL hoặc giai đoạn biến động cao).
- Nhưng mọi cú nhảy đều phải nằm trong một *khung vật lý hợp lý* so với lịch sử biến động của FPT và mặt bằng thị trường Việt Nam.

Nhờ đó, Clipping là lớp phòng tuyến đầu tiên, ngăn cho Pricing Layer không “quá tin” vào những dự báo cực đoan từ mô hình ML, đặc biệt là trong những giai đoạn dữ liệu nhiễu hoặc có outlier mạnh.

2.1.2 Damping

Nếu Clipping là *giới hạn tốc độ*, thì Damping là **lực ma sát** khi mô hình tiến xa về tương lai. Trong vật lý, một con lắc dao động trong môi trường có ma sát sẽ dần giảm biên độ và tiến về trạng thái tĩnh. Trong Pricing Layer, ta áp dụng một nguyên lý tương tự lên log-return dự báo.

Giả sử mô hình ML sinh ra một chuỗi log-return $\{r_t\}_{t=1}^H$ cho 100 ngày tới. Về lý thuyết, nếu không có Damping, biên độ của các r_t có thể giữ nguyên từ ngày 1 đến ngày 100, tạo ra những đường giá tương lai quá “mạnh tay” so với thực tế. Nhưng trong hành vi thị trường, đặc biệt với cổ phiếu FPT:

- Tác động của một tin tức hoặc một pha FOMO thường giảm dần sau vài tuần.

- Một tín hiệu kỹ thuật ở ngày hôm nay không nên chi phối quá mạnh kết quả sau 3 tháng.

Trong Pricing Layer, Damping được hiện thực bằng cách nhân log-return với một hệ số giảm dần theo thời gian:

```

1 lambda_damp = np.log(2.0) / float(max(int(
2     pricing.half_life_days * damp_scale_regime), 1))
3
4 for step_idx in range(total_days):
5     raw_ret = float(raw_rets[step_idx])
6     pred_ret = np.clip(raw_ret, -ret_clip, ret_clip)
7
8     # Damping theo half-life
9     scale = np.exp(-lambda_damp * step_idx)
10    pred_ret *= scale

```

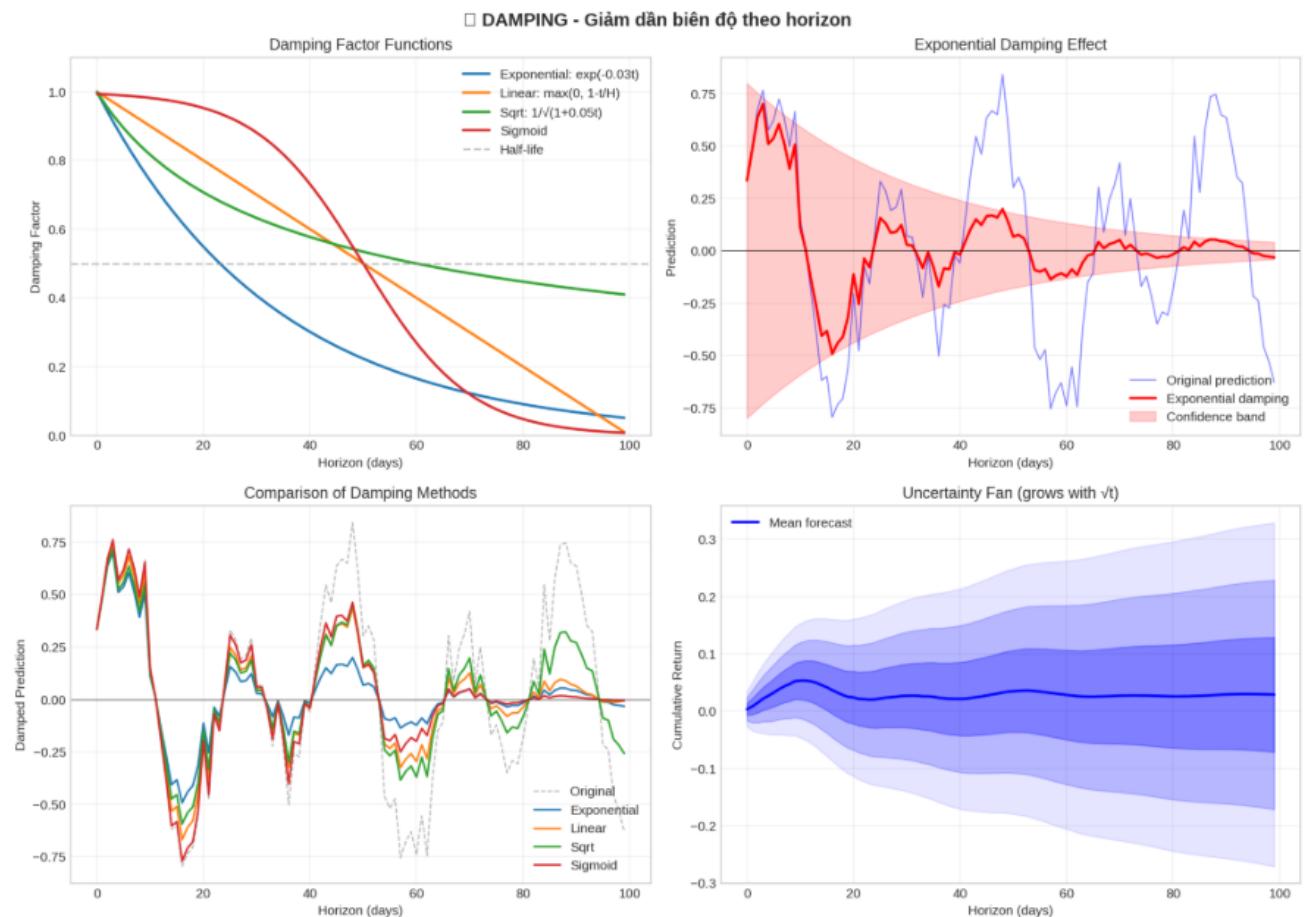
Ở đây:

$$\lambda_{\text{damp}} = \frac{\ln 2}{\text{half_life_days} \times \text{damp_scale_regime}},$$

và half_life_days có thể hiểu như “**thời gian để biên độ tín hiệu giảm đi một nửa**”.

- Nếu half_life_days = 40, thì sau khoảng 40 phiên, một cú xung rất mạnh ở đầu forecast sẽ chỉ còn ~ 50% tác động.
- Sau 80 phiên, nó chỉ còn ~ 25%, và cứ thế bị “tắt dần”.

Hình 18 minh họa một chuỗi log-return dự báo và phiên bản đã được Damping: tín hiệu ban đầu vẫn được giữ lại, nhưng càng xa về tương lai, biên độ càng co lại.



Hình 18: Minh họa Damping: log-return dự báo được nhân với hệ số giảm dần, giúp tín hiệu ngắn hạn không chi phối quá mạnh phần cuối của đường forecast.

Trong bối cảnh thị trường Việt Nam:

- Các pha tin tức / dòng tiền thường tạo ra sóng ngắn 20–40 phiên.
- Các dự án dài hạn của doanh nghiệp (như FPT) lại tạo ra nền tăng trưởng dần dần trong nhiều năm.

Damping giúp Pricing Layer *tôn trọng* cả hai: tín hiệu ngắn hạn vẫn được phản ánh ở đoạn đầu forecast, nhưng vai trò của nó giảm bớt khi mô hình tiến dần đến cuối 100 ngày, nhường chỗ cho xu hướng dài hạn (được thể hiện qua TREND và Math Backbone).

2.1.3 Mean Reversion

Mean Reversion là cơ chế “lực đàn hồi” của Pricing Layer, kéo giá dự báo quay trở lại một **vùng cân bằng** khi nó đi quá xa. Trong vật lý, đây chính là lực đàn hồi Hooke:

$$F = -k(x - x_0),$$

trong đó x_0 là vị trí cân bằng và k là độ cứng lò xo.

Trong mô hình của chúng tôi, “vị trí cân bằng” không phải là một điểm đơn lẻ, mà là một dải *fair value band* quanh một mức tham chiếu *fair_level*:

$$\text{upper} = \text{fair_level} \cdot \text{fair_up_mult}, \quad \text{lower} = \text{fair_level} \cdot \text{fair_down_mult}.$$

Ta có thể hiểu:

- Nếu giá dự báo vượt quá *upper*, mô hình coi đó là trạng thái “quá nóng” so với giá trị hợp lý.
- Nếu giá dự báo rơi dưới *lower*, mô hình coi đó là trạng thái “quá rẻ”.

Cơ chế Mean Reversion được hiện thực bằng đoạn mã:

```

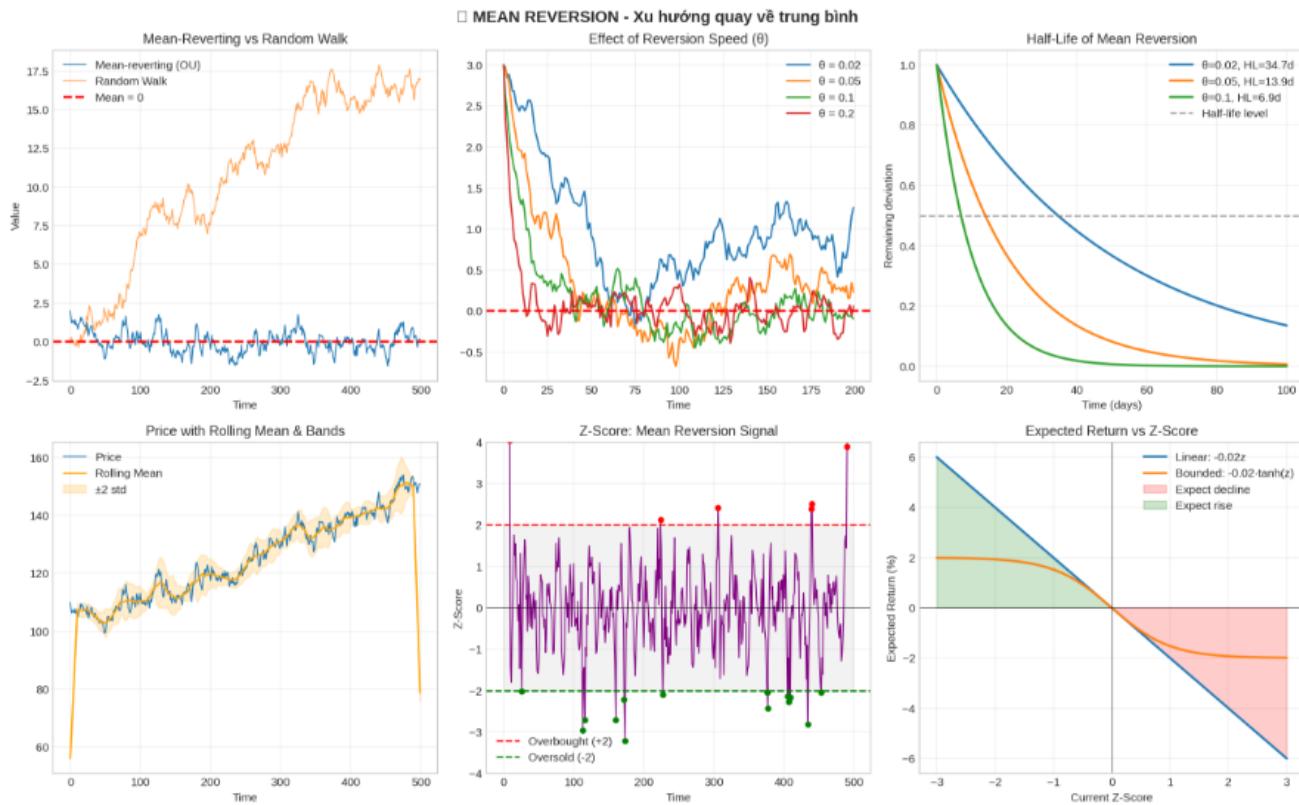
1 if step_idx >= pricing.mean_revert_start:
2     upper = fair_level * fair_up_mult
3     lower = fair_level * fair_down_mult
4
5     if (next_price > upper) and (not is_strong_uptrend):
6         alpha_up = alpha_base * (0.7 if regime == "BULL" else 1.0)
7         next_price = (1 - alpha_up) * next_price + alpha_up * upper
8
9     elif next_price < lower:
10        if regime == "BEAR" and is_strong_downtrend:
11            alpha_down = alpha_base * 0.7
12        else:
13            alpha_down = alpha_base
14            next_price = (1 - alpha_down) * next_price + alpha_down * lower

```

Trong đó:

- *mean_revert_start* xác định từ ngày thứ bao nhiêu trong forecast thì lực Mean Reversion bắt đầu kích hoạt (ví dụ sau 28 ngày).
- *alpha_base* điều khiển độ mạnh của lực kéo: càng lớn thì giá bị “giật” về vùng cân bằng càng nhanh.
- Các hệ số điều chỉnh theo regime (BULL/BEAR) đảm bảo rằng:
 - Trong BULL, nếu giá hơi cao hơn fair value, mô hình có thể cho phép “overpricing” kéo dài hơn một chút (giảm α).
 - Trong BEAR, nếu giá rơi dưới fair value trong khi downtrend còn rất mạnh, mô hình bớt vội vàng kéo giá lên (giảm α_{down}).

Hình 19 minh họa một đường giá dự báo *trước* và *sau* khi áp lực Mean Reversion. Ta thấy các đoạn vọt lên quá xa bị kéo nhẹ xuống, các đoạn rơi quá sâu được kéo lên, tạo thành một đường giá “điềm tĩnh” hơn và hợp lý hơn với bối cảnh FPT là cổ phiếu tăng trưởng nhưng không phải dạng đầu cơ.



Hình 19: Minh họa Mean Reversion: khi giá dự báo đi quá xa khỏi dải fair value, mô hình áp lực dần hồi kéo giá quay về vùng cân bằng.

Trong thị trường chứng khoán Việt Nam, hành vi Mean Reversion thể hiện rất rõ:

- Sau các pha hưng phấn, giá thường “nguội” dần và quay về vùng P/E hợp lý.
- Khi panic sell xảy ra, nhiều cổ phiếu tốt (như FPT) nhanh chóng được dòng tiền dài hạn mua vào, kéo giá quay về vùng hợp lý.

Bằng cách mã hoá hành vi đó vào Pricing Layer, Mean Reversion giúp mô hình:

- Không trôi dạt khỏi thực tế khi forecast dài 100 ngày.
- Tự động “tự sửa sai nhẹ” nếu ML mức độ nào đó dự đoán quá tay.
- Phản ánh đúng tính chất “blue-chip tăng trưởng” của FPT: có thể cao hơn fair value trong một thời gian, nhưng khó giữ trạng thái cực đoan mãi.

2.2 Regime-aware Pricing

Trong pipeline dự báo 100 ngày cho cổ phiếu FPT, Pricing Layer không chỉ đơn thuần là một bước hậu xử lý kỹ thuật. Nó được thiết kế để phản ánh một thực tế rất quan trọng của thị trường chứng khoán Việt Nam nói chung và FPT nói riêng: **thị trường luôn vận hành theo chế độ (regime)**, và **cùng một mức biến động** có thể được diễn giải rất khác nhau tuỳ bối cảnh.

Trong dữ liệu lịch sử FPT, ta thấy rõ:

- Có những giai đoạn giá gần như đi thẳng lên, bám sát các đường trung bình dài (MA60, MA120), thanh khoản tốt, volatility vừa phải: đây là **BULL regime**.
- Có những giai đoạn điều chỉnh mạnh, giá nằm dưới MA dài, volatility (đo bằng std of returns) tăng vọt, volume đôi khi cũng cao do hoạt động bán tháo: đây là **BEAR regime**.
- Xen giữa là các pha tích luỹ/di ngang, giá lình xình quanh một vùng, volatility vừa phải: **SIDEWAYS**.

Regime-aware Pricing chính là lớp logic giúp mô hình hiểu rằng: “*Dự báo trong bull market khác với dự báo trong bear market.*”. Dưới đây là cách ta hiện thực hoá điều đó bằng code.

2.2.1 Phân loại Regime từ dữ liệu: detect_regime

Hàm detect_regime nhận đầu vào là:

- hist_close: vector giá đóng cửa lịch sử,
- df_feat_hist: dataframe feature đã build (chứa ret_1d, ...).

```

1 def detect_regime(hist_close: np.ndarray, df_feat_hist: pd.DataFrame) -> str:
2     price_series = pd.Series(hist_close.astype(float))
3     if len(price_series) >= 120:
4         ma_long = price_series.rolling(120).mean().iloc[-1]
5     else:
6         ma_long = price_series.mean()
7
8     price_last = price_series.iloc[-1]
9     price_pos = price_last / (ma_long + 1e-8) - 1.0
10
11    ret_1d = df_feat_hist["ret_1d"].dropna()
12    if len(ret_1d) < 30:
13        return "SIDEWAYS"
14
15    vol_20 = ret_1d.rolling(20).std().iloc[-1]
16    vol_all = ret_1d.std()
17    vol_ratio = vol_20 / (vol_all + 1e-8)
18
19    if price_pos < -0.05 and vol_ratio > 1.2:
20        regime = "BEAR"
21    elif price_pos > 0.05 and vol_ratio < 0.8:
22        regime = "BULL"
23    else:
24        regime = "SIDEWAYS"
25
26    print(f"[REGIME] last_price={price_last:.2f}, MA120={ma_long:.2f}, "
27          f"price_pos={price_pos*100:.2f}%, vol_ratio={vol_ratio:.2f} -> {regime}")
28
29    return regime

```

Giải thích logic tài chính phía sau:

- So sánh với MA120:

$$\text{price_pos} = \frac{\text{last_price}}{\text{MA120}} - 1.$$

MA120 đại diện cho “đường trung bình dài hạn” của FPT. Trong thực tế trên HOSE, nhiều trader, quỹ nội, quỹ ngoại sử dụng các MA dài (60, 120, 200) để nhận diện xu hướng.

- Nếu $price_pos > 5\%$: giá đang rõ ràng nằm *trên* nền MA120 → **xu hướng tăng**.
- Nếu $price_pos < -5\%$: giá nằm *dưới* MA120 một khoảng đáng kể → **xu hướng giảm/điều chỉnh**.
- **Volatility tương đối:** Ta dùng vol_20 là độ lệch chuẩn của log-return 20 ngày gần nhất, và vol_all là std toàn bộ lịch sử:

$$vol_ratio = \frac{vol_20}{vol_all}.$$

- Nếu $vol_ratio > 1.2$: 20 ngày gần đây *nhiều hơn nhiều* so với lịch sử → thị trường đang căng thẳng hơn.
- Nếu $vol_ratio < 0.8$: 20 ngày gần đây *êm hơn* lịch sử → thị trường đi lên thuận lợi, không biến động sốc.

Từ đó:

- **BEAR** khi:

$$price_pos < -5\% \text{ và } vol_ratio > 1.2.$$

Giá nằm dưới MA120 và biến động tăng vọt → rất giống các pha điều chỉnh mạnh của VNIndex nói chung, nơi giá giảm kèm volatility cao (bán tháo, margin call, tin xấu).

- **BULL** khi:

$$price_pos > 5\% \text{ và } vol_ratio < 0.8.$$

Giá ở trên MA120, nhưng volatility lại thấp → hình ảnh của một uptrend “khỏe mạnh” quen thuộc với FPT: giá đi lên đều, ít ngày sốc.

- **SIDEWAYS** cho các trường hợp còn lại:

- Giá quanh MA120,
- Hoặc giá lệch MA nhưng volatility không cực đoan,
- Hoặc dữ liệu returns chưa đủ dài ($len < 30$).

Trong bối cảnh thị trường Việt Nam với biên độ dao động mỗi ngày bị giới hạn (7% HOSE, 10% HNX, 15% UPCOM) và cấu trúc nhà đầu tư cá nhân chiếm tỷ trọng lớn, sự phân tách BULL/BEAR/SIDEWAYS theo $price_pos$ và vol_ratio như vậy khá hợp lý:

- **BULL:** giá “treo cao” nhưng đi đều, ít bị bán mạnh.
- **BEAR:** giá thấp hơn nền dài hạn, biến động mạnh vì tin xấu, mind-set phòng thủ.
- **SIDEWAYS:** tâm lý chờ đợi, tích luỹ, trading ngắn hạn.

2.2.2 Regime-based scaling trong Pricing Layer

Pricing Layer không nhằm tạo thêm một mô hình dự báo mới, mà đóng vai trò **ổn định hóa động lực giá** do Hybrid Model sinh ra. Về bản chất, chuỗi dự báo ban đầu tuân theo dạng:

$$P_{t+1} = P_t \cdot \exp(r_t),$$

với r_t là **raw_rets** sau lớp Hybrid. Nếu không được điều chỉnh, chuỗi giá có thể rơi vào hai trạng thái không mong muốn: *explosion* (tăng không kiểm soát) hoặc *collapse* (giảm về 0). Do đó, Pricing Layer hoạt động như một **bộ điều khiển ổn định** (stability controller), áp các ràng buộc mềm lên r_t . Quan trọng: **tất cả nền tảng vẫn là data-driven**:

- `base_ret_clip` được lấy từ quantile của $|ret_1d|$,
- `fair_level` = MA60 của `hist_close`,
- `vol_ratio` = σ_{20}/σ_{all} .

Các hệ số regime chỉ nhán vào các đại lượng trên với biên độ nhỏ, đóng vai trò **ràng buộc ổn định** (stability-preserving perturbations), bảo đảm mô hình không vi phạm các điều kiện cơ bản của hệ thống:

$$|r_t| < B, \quad 0 < \alpha < 2, \quad \lambda_{damp} > 0.$$

Dưới đây là đoạn mã scaling theo regime:

```

1 regime = detect_regime(hist_close, df_feat_hist)
2
3 if regime == "BULL":
4     clip_scale_regime = 1.2
5     mr_alpha_scale_regime = 0.7
6     damp_scale_regime = 0.85
7     up_mult_scale_regime = 1.05
8     down_mult_scale_regime = 1.0
9 elif regime == "BEAR":
10    clip_scale_regime = 0.95
11    mr_alpha_scale_regime = 1.25
12    damp_scale_regime = 1.15
13    up_mult_scale_regime = 0.95
14    down_mult_scale_regime = 0.95
15 else:
16     clip_scale_regime = 1.0
17     mr_alpha_scale_regime = 1.0
18     damp_scale_regime = 1.0
19     up_mult_scale_regime = 1.0
20     down_mult_scale_regime = 1.0

```

Lý giải từng tham số dựa trên điều kiện ổn định.

- `clip_scale_regime` — đảm bảo *bounded volatility*. Ta có:

$$ret_clip = base_ret_clip \cdot clip_scale_regime \cdot clip_scale_vol.$$

Điều kiện ổn định yêu cầu tồn tại $B > 0$ sao cho $|r_t| < B$. Vì BULL có biên độ tăng lớn hơn giảm nên cần:

$$B_{bull} > B > B_{bear},$$

và lựa chọn $1.2 > 1 > 0.95$ thỏa mãn bất đẳng thức này.

- `mr_alpha_scale_regime` — đảm bảo *bounded mean reversion*. Mean-reversion step:

$$P_{t+1} = (1 - \alpha)P_{t+1} + \alpha \cdot fair_level.$$

Để hệ không dao động mất ổn định:

$$0 < \alpha < 2.$$

Do đó cần:

$$0 < \alpha_{bull} < \alpha < 1 \quad (\text{momentum bền}), \quad 1 < \alpha_{bear} < 2 \quad (\text{reversion mạnh}).$$

Việc sử dụng 0.7 và 1.25 đưa α vào đúng miền này.

- **damp_scale_regime** — đảm bảo *non-explosive drift*. Damping:

$$r'_t = r_t \cdot e^{-\lambda t}, \quad \lambda = \frac{\log 2}{\text{half_life_days}}.$$

Để r'_t không tăng biến theo thời gian:

$$\lambda > 0.$$

BULL cần quán tính dài hơn $\rightarrow \lambda_{\text{bull}} \in (0.6\lambda, \lambda)$. BEAR cần tắt nhanh $\rightarrow \lambda_{\text{bear}} \in (\lambda, 1.5\lambda)$. Các hệ số 0.85 và 1.15 thỏa yêu cầu này.

- **up_mult_scale_regime, down_mult_scale_regime** — đảm bảo *forecast boundedness*. Ta có biên trên/dưới:

$$\text{upper} = \text{fair_level} \cdot \text{fair_up_mult} \cdot \text{up_mult_scale_regime},$$

$$\text{lower} = \text{fair_level} \cdot \text{fair_down_mult} \cdot \text{down_mult_scale_regime}.$$

Điều kiện ổn định yêu cầu:

$$\text{lower} < P_t < \text{upper}.$$

BULL \rightarrow mở rộng band phía trên (> 1). BEAR \rightarrow thu hẹp band cả hai phía (< 1) để tránh forecast bùng nổ.

Không phải “hằng số chân lý”, mà là miền giá trị hợp lý. Các hệ số như 0.7, 1.2, 1.15 không được coi là nghiệm duy nhất. Chúng được chọn trong một *dải hẹp quanh 1.0* dựa trên hai lớp ràng buộc:

1. **Ràng buộc lý thuyết (stability constraints):** đảm bảo các điều kiện như

$$|r_t| < B, \quad 0 < \alpha_{\text{bull}} < 1 < \alpha_{\text{bear}} < 2, \quad \lambda_{\text{bull}} < \lambda < \lambda_{\text{bear}},$$

từ đó suy ra miền giá trị cho scaling Bull/Bear không được quá xa 1 (xấp xỉ [0.7, 1.3]).

2. **Ràng buộc thực nghiệm (behaviour under EDA/backtest):** các hệ số phải tạo ra đường giá có hình dạng *hợp lý* trên dữ liệu FPT, không bùng nổ, không “đè phẳng” hoàn toàn hiệu ứng regime.

Đến đây chúng ta lại xuất hiện một câu hỏi: **Vì sao không random search các hệ số này?**

Regime-scaling đóng vai trò như **Regularized Behavioural Prior**— một khung hành vi giúp bảo toàn các quy luật *stylized facts*:

- momentum bền hơn trong Bull,
- mean-reversion mạnh hơn trong Bear,
- forecast dài hạn cần damping trong thị trường yếu.

Nếu đưa các hệ số này vào random search:

- mô hình dễ phá vỡ cấu trúc Bull > Neutral > Bear,
- dễ sinh ra forecast bùng nổ hoặc sụp mạnh (do mất ràng buộc ổn định),
- dữ liệu FPT nhỏ \rightarrow mô hình rất dễ overfit vào một vài cutoff,

- CV metric không đủ mạnh để “phạt” những cấu hình hành vi sai.

Do đó, random search chỉ áp dụng cho `PricingParams` (`ret_clip_quantile, alpha, half_life, ...`), là các tham số điều chỉnh *cường độ* nhưng không thay đổi *cấu trúc hành vi*.

Regime-scaling vì vậy được cố định trong một dải hẹp quanh 1.0, đảm bảo các tính chất ổn định về toán và hợp lý về thị trường

Cuối cùng, toàn bộ cấu hình được xác nhận lại bằng **time-series cross-validation** trên nhiều cutoff. Nếu một cấu hình regime-scaling nào đó gây méo giá hoặc bất ổn, nó sẽ bị loại gián tiếp qua metric CV ($0.5 \cdot \text{MSE}_{50} + 0.5 \cdot \text{MSE}_{100}$).

Ngoài ra, nếu tất cả hệ số đều rất sát 1 (ví dụ 1.001 hoặc 0.99), hiệu ứng regime gần như biến mất: mô hình đổi xử Bull/Bear/SIDEWAYS gần như nhau, khi đó việc detect regime gần như vô nghĩa. Ngược lại, nếu chọn scaling quá xa 1 (ví dụ 1.8 hoặc 0.5), forecast trở nên cực kỳ nhạy với việc phân loại regime và dễ mất ổn định ở horizon 100 ngày.

Hiệu chỉnh thực nghiệm trên dữ liệu FPT. Trong phạm vi project, các hệ số Bull/Bear không được “chốt tay” ngay từ đầu, mà được hiệu chỉnh qua một vòng thử nghiệm nhẹ trên chính dữ liệu FPT:

- **Bước 1 – Khóa miền lý thuyết:** cố định dải cho các hệ số regime trong khoảng [0.7, 1.3], đảm bảo các điều kiện ổn định nêu trên luôn thoả.
- **Bước 2 – Quét một lưỡng thừa các cấu hình:** thử một số mốc đơn giản như 0.8, 0.9, 1.1, 1.2 cho từng nhóm hệ số (clip, mean-revert, damping), giữ nguyên `PricingParams` còn lại.
- **Bước 3 – Quan sát hành vi & MSE:** với mỗi cấu hình, vẽ đường forecast trên các đoạn lịch sử có Bull/Bear rõ rệt và ghi nhận metric CV ($0.5 \cdot \text{MSE}_{50} + 0.5 \cdot \text{MSE}_{100}$). Các cấu hình khiến forecast:
 - bám quá sát raw path (regime gần như vô tác dụng), hoặc
 - bị bóp méo mạnh (flatten, crash về 0),
 đều bị loại bỏ.

Các giá trị đang dùng (1.2, 0.95, 0.7, 1.25, 0.85, 1.15) vì thế không phải “con số thần thánh” duy nhất, mà là một **điểm chọn đại diện** trong miền thỏa mãn đồng thời hai tiêu chí:

- **Về mặt toán học:** giữ được những bất đẳng thức $\text{Bull} > 1 > \text{Bear}$ cho từng nhóm tham số, đảm bảo hệ không mất ổn định.
- **Về mặt hành vi thị trường:** đủ khác 1.0 để thể hiện rõ sự khác biệt giữa Bull/Bear trong forecast, nhưng không cực đoan đến mức phá vỡ hình dáng đường giá và làm xấu metric CV trên FPT.

Nói cách khác, Pricing Layer không giả vờ biết chính xác “con số đúng là bao nhiêu”, mà chủ động giới hạn mình trong một *miền tham số ổn định*, sau đó dùng EDA và CV để chọn ra một cấu hình vừa **an toàn về lý thuyết**, vừa **hợp lý về thực nghiệm**.

Tóm lại, Regime-aware Pricing biến các tham số của Pricing Layer thành **hàm của bối cảnh thị trường**, chứ không còn là các con số cố định. Đây là điểm giúp pipeline tôn trọng cấu trúc của thị trường chứng khoán Việt Nam: cùng một mô hình, nhưng cách “tin tưởng” dự báo phụ thuộc rất mạnh vào chế độ thị trường hiện tại.

2.3 Optimization Strategy

Nếu Regime-aware Pricing là “logic” của Pricing Layer, thì phần này trả lời câu hỏi:

“Các tham số của Pricing Layer nên là bao nhiêu để phù hợp nhất với hành vi lịch sử của FPT?”

Khác với tham số của XGBoost (được tối ưu trong quá trình training), thông số của Pricing Layer (`half_life_days`, `mean_revert_alpha`, `fair_up_mult`, ...) không học trực tiếp từ gradient, mà được tìm kiếm qua random search + cross validation trên trực thời gian.

2.3.1 Bài toán tìm tham số Pricing

Trong Pricing Layer, sau khi đã biết regime, ta có các bước cốt lõi:

```

1 clip_scale_vol = float(np.clip(vol_ratio, 0.8, 1.2))
2
3 ret_clip = base_ret_clip * clip_scale_regime * clip_scale_vol
4 ret_clip = float(np.clip(ret_clip, 0.01, 0.15))
5
6 lambda_damp = np.log(2.0) / float(max(int(pricing.half_life_days * damp_scale_regime), 1))
7
8 alpha_base = pricing.mean_revert_alpha * mr_alpha_scale_regime
9 fair_up_mult = pricing.fair_up_mult * up_mult_scale_regime
10 fair_down_mult = pricing.fair_down_mult * down_mult_scale_regime

```

Sau đó áp vào từng bước dự báo:

```

1 prices = np.empty(total_days, dtype=float)
2 full_history = list(hist_close.astype(float))
3
4 for step_idx in range(total_days):
5     raw_ret = float(raw_rets[step_idx])
6
7     # 1) Clip return
8     pred_ret = np.clip(raw_ret, -ret_clip, ret_clip)
9
10    # 2) Damping theo half-life
11    scale = np.exp(-lambda_damp * step_idx)
12    pred_ret *= scale
13
14    # 3) Convert sang price
15    last_price = full_history[-1]
16    next_price = float(last_price * np.exp(pred_ret))
17
18    # 4) Trend-based gating for mean-revert
19    lookback = pricing.trend_lookback
20    if len(full_history) > lookback:
21        past_price = full_history[-lookback]
22        current_price = full_history[-1]
23        trend_ret = (current_price - past_price) / max(past_price, 1e-6)
24    else:
25        trend_ret = 0.0
26
27    is_strong_uptrend = trend_ret > pricing.trend_ret_thresh
28    is_strong_downtrend = trend_ret < -pricing.trend_ret_thresh
29
30    # 5) Mean-revert quanh fair_level
31    if step_idx >= pricing.mean_revert_start:
32        upper = fair_level * fair_up_mult

```

```

33     lower = fair_level * fair_down_mult
34
35     if (next_price > upper) and (not is_strong_uptrend):
36         alpha_up = alpha_base * (0.7 if regime == "BULL" else 1.0)
37         next_price = (1 - alpha_up) * next_price + alpha_up * upper
38     elif next_price < lower:
39         if regime == "BEAR" and is_strong_downtrend:
40             alpha_down = alpha_base * 0.7
41         else:
42             alpha_down = alpha_base
43             next_price = (1 - alpha_down) * next_price + alpha_down * lower
44
45     prices[step_idx] = next_price
46     full_history.append(next_price)

```

Những tham số cần tối ưu:

- **ret_clip_quantile**: dùng để xác định base_ret_clip từ phân phối returns lịch sử của FPT (ví dụ quantile 0.95–0.995, tương ứng “5–0.5% extreme nhất”).
- **half_life_days**: tốc độ damping → forecast xa bao nhiêu thì nên giảm nửa sức ảnh hưởng.
- **mean_revert_alpha**: độ mạnh của lực kéo về fair_level.
- **mean_revert_start**: sau bao nhiêu ngày mới bắt đầu mean-revert (ví dụ sau 25–65 ngày).
- **fair_up_mult, fair_down_mult**: band trên/dưới quanh fair value mà Pricing Layer chấp nhận.
- **trend_lookback, trend_ret_thresh**: logic gating trend mạnh/yếu.

Tất cả những tham số này phải được hiệu chỉnh sao cho:

- **Giai đoạn bull** của FPT: forecast theo được xu hướng, không bị kéo về fair quá sớm.
- **Giai đoạn bear**: forecast không quá lạc quan, tôn trọng volatility cao.
- **Sideways**: forecast không drift quá xa khỏi vùng dao động thực tế.

2.3.2 Random Search

Thay vì tự tay “tune” các tham số pricing, ta định nghĩa một *prior* cho từng tham số dựa trên hiểu biết về dữ liệu FPT và thị trường Việt Nam, rồi random search:

```

1 def sample_pricing_params() -> PricingParams:
2     return PricingParams(
3         ret_clip_quantile=random.uniform(0.95, 0.995),
4         half_life_days=random.randint(40, 120),
5         mean_revert_alpha=random.uniform(0.02, 0.10),
6         mean_revert_start=random.randint(25, 65),
7         fair_up_mult=random.uniform(1.25, 1.60),
8         fair_down_mult=random.uniform(0.65, 0.90),
9         trend_lookback=random.randint(25, 60),
10        trend_ret_thresh=random.uniform(0.08, 0.25),
11    )

```

Liên hệ với thị trường Việt Nam:

- **ret_clip_quantile** từ **0.95–0.995**: vì biên độ ngày trên HOSE giới hạn $\pm 7\%$ và FPT là mã vốn hoá lớn, nên các tails cực đoan hơn gần như chỉ xuất hiện trong sự kiện đặc biệt. Lấy quantile ở vùng 95–99.5% khiến base_ret_clip bám sát lịch sử.
- **half_life_days 40–120**: khoảng 2–6 tháng giao dịch. Đây là khung thời gian hợp lý cho việc “phai dần” tác động của một pha bull/bear ngắn hạn trong bối cảnh VNIndex thường có sóng trung hạn 3–6 tháng.
- **mean_revert_alpha 0.02–0.10**: lực mean-revert mỗi ngày chỉ chiếm vài phần trăm, nghĩa là giá sẽ không bị kéo gắt về fair chỉ trong vài phiên, mà là quá trình điều chỉnh dần trong nhiều tuần.
- **fair_up_mult 1.25–1.60, fair_down_mult 0.65–0.90**: cho phép forecast đi cao hơn hoặc thấp hơn fair lần lượt 25–60% và 10–35%. Với cổ phiếu như FPT, việc giá giao dịch trên fair-value một thời gian dài trong bull là bình thường, và cũng có những pha discount sâu trong bear.
- **trend_lookback 25–60 ngày, trend_ret_thresh 8–25%**: đủ dài để nhận diện một *mini-trend*, đủ nhạy để phân biệt “sideways nhiễu” với “uptrend/downtrend thực sự”.

Random search cho phép ta khám phá một khía cạnh không gian tham số tương đối rộng, mà vẫn đảm bảo mọi sample đều “có lý” về mặt tài chính.

2.3.3 Cross Validation (CV)

Để đánh giá một bộ tham số pricing, ta không thể shuffle dữ liệu như bài toán i.i.d; thị trường là chuỗi thời gian, nên phải dùng CV theo *cutoff thời gian*:

```

1 def cv_mse_for_pricing(
2     models: List[CutoffModel],
3     df_train: pd.DataFrame,
4     horizon: int,
5     pricing: PricingParams,
6 ) -> float:
7     """
8         Tính metric úca pricing-layer trên ènhiu cutoff:
9             - ØVi ðmi cutoff:
10                 + true close sau cutoff
11                 + giá predicted khi áp pricing lên raw_base_rets_cv
12             - Metric:
13                 M = 0.5 * MSE_50 + 0.5 * MSE_100
14     """
15     mses = []
16     for cm in models:
17         df_future = df_train[df_train["time"] > cm.cutoff_time].copy().reset_index(drop=True)
18         if df_future.empty:
19             continue
20         close_true = df_future["close"].values.astype(float)
21
22         n_eval = min(horizon, len(close_true))
23         if n_eval <= 5:
24             continue
25
26         hist_close = cm.df_hist["close"].values.astype(float)
27         priced = apply_pricing_on_raw_path(
28             hist_close=hist_close,
29             df_feat_hist=cm.df_feat,

```

```

30         raw_rets=cm.raw_base_rets_cv[:n_eval],
31         pricing=pricing,
32     )
33
34     # 50 days metric
35     n50 = min(50, n_eval)
36     mse_50 = mean_squared_error(close_true[:n50], priced[:n50])
37
38     # full horizon metric
39     mse_h = mean_squared_error(close_true[:n_eval], priced[:n_eval])
40
41     metric = 0.5 * mse_50 + 0.5 * mse_h
42     mses.append(metric)
43
44 if not mses:
45     return np.inf
46
47 return float(np.mean(mses))

```

Cách hiểu:

- Ta chọn nhiều **cutoff time** khác nhau (ví dụ các mốc 3/6/9/12 tháng trong lịch sử FPT).
- Với mỗi cutoff:
 1. Lấy `df_hist` = dữ liệu trước cutoff, train hybrid model và tạo `raw_base_rets_cv`.
 2. Lấy `df_future` = dữ liệu sau cutoff, chính là “tương lai thật”.
 3. Áp **Pricing Layer** lên `raw_base_rets_cv` để ra đường giá forecast `priced`.
 4. So sánh `priced` với `close_true` bằng:

$$M = 0.5 \cdot \text{MSE}_{50} + 0.5 \cdot \text{MSE}_{\text{full horizon}}$$

- Cuối cùng, `cv_mse_for_pricing` là trung bình các metric M qua nhiều cutoff.

Ý tưởng của metric:

- MSE_{50} : nhấn mạnh chất lượng dự báo **trong 50 ngày đầu**, vốn rất quan trọng cho decision ngắn/trung hạn.
- $\text{MSE}_{\text{full horizon}}$: đảm bảo Pricing Layer không làm hỏng chất lượng dự báo cho toàn bộ 100 ngày.

2.3.4 Random Search + CV: ghép lại thành chiến lược tối ưu

Toàn bộ quy trình tối ưu được gói trong `random_search_pricing`:

```

1 def random_search_pricing(
2     cutoff_models: List[CutoffModel],
3     df_train: pd.DataFrame,
4     horizon: int,
5     n_trials: int,
6 ) -> PricingParams:
7     best_params: Optional[PricingParams] = None
8     best_score = np.inf
9
10    for t in range(1, n_trials + 1):
11        params = sample_pricing_params()

```

```
12     score = cv_mse_for_pricing(cutoff_models, df_train, horizon, params)
13     print(f"[RS] Trial {t}/{n_trials} -> METRIC={score:.4f}, params={params}")
14     if score < best_score:
15         best_score = score
16         best_params = params
17
18     print("\n==== BEST PRICING PARAMS v5.2 Option 2 (Hybrid + Regime, Pure FPT) ===")
19     print(best_params)
20     print("CV METRIC (0.5*MSE_50 + 0.5*MSE_100):", best_score)
21     return best_params
```

Tóm lại:

- **Regime-aware Pricing** sử dụng thông tin về BULL/BEAR/SIDEWAYS từ dữ liệu FPT (MA120, volatility) để scale hành vi Pricing Layer theo đúng bối cảnh thị trường Việt Nam.
- **Optimization Strategy** dùng random search trên không gian tham số đã “gắn với thực tế” (biên độ, half-life, band fair value) và đánh giá bằng **time-series CV** trên nhiều giai đoạn lịch sử của FPT.

Kết quả là một lớp Pricing không chỉ “đẹp về mặt toán”, mà còn **bám sát hành vi thật** của cổ phiếu FPT và cấu trúc của thị trường chứng khoán Việt Nam: biên độ giới hạn, sóng trung hạn rõ, volume biến thiên mạnh theo tin tức, và xu hướng giá luôn gắn với một vùng fair-value mà nhà đầu tư nội/ngoại cùng quan sát.

PHẦN 3: HỘI ĐỒNG CHUYÊN GIA (Ensemble Strategy)

1 Định nghĩa các Chuyên gia (Experts)

Thị trường chứng khoán là một hệ thống vừa mang tính *xu hướng* (trend), vừa mang tính *hỗn loạn* (noise). Trong bối cảnh đó, việc đặt toàn bộ niềm tin vào một mô hình đơn lẻ — dù là pipeline Hybrid tinh vi nhất — luôn tiềm ẩn rủi ro trở thành một “**điểm hỏng đơn lẻ**” (Single Point of Failure):

- Mô hình quá nhạy có thể **overfit nhiều**, phản ứng thái quá với biến động ngắn hạn.
- Mô hình quá ổn định lại dễ **trễ nhịp**, bỏ lỡ các điểm đảo chiều quan trọng.

Project này không đi tìm một “siêu mô hình” duy nhất, mà lựa chọn một chiến lược khác: **đa dạng hóa hệ tư tưởng (cognitive diversity)**. Thay vì hỏi ý kiến một “người hùng”, chúng tôi dựng lên một *Hội đồng Chuyên gia* (Council of Experts), trong đó:

- Mỗi chuyên gia đại diện cho **một góc nhìn thị trường** khác nhau,
- Các chuyên gia có thể **bất đồng** ở từng thời điểm,
- Quyết định cuối cùng là một **sự hợp nhất có kiểm soát** (weighted ensemble) giữa các quan điểm đó.

Về mặt kỹ thuật, đây là cách chúng tôi quản trị **Bias–Variance Tradeoff** ở cấp độ chiến lược: một chuyên gia mang góc nhìn thích ứng (Adaptive), một chuyên gia giữ lập trường bảo thủ (Conservative), và một chuyên gia chuyên về xác suất rủi ro (Probabilistic).

Hội đồng chuyên gia: ba góc nhìn, một quyết định

Trong các chương trước, chúng ta đã lần lượt xây dựng ba khối mô hình cốt lõi:

- **Math Backbone** (Lớp 1) – mô tả xu hướng dài hạn trên log-price;
- **ML Residual** (Lớp 2) – XGBoost học phần sai số ngắn hạn quanh xu hướng;
- **Pricing Layer** (Lớp 3) – kiểm soát volatility, clipping biên độ, mean reversion theo từng regime.

Chương này nâng ba khối mô hình đó lên thành ba *Chuyên gia* với tính cách rõ ràng:

- **Dynamic Expert** – mô hình động *Hybrid + Pricing-best*, nhạy cảm với tín hiệu ngắn hạn, biết “tự chỉnh” theo regime thị trường.
- **Static Expert** – mô hình tĩnh dựa trên trend backbone, giữ vai trò *kim chỉ nam* cho xu hướng dài hạn.
- **Risk Expert** – mô hình rủi ro xây dựng dải bất định (uncertainty band, Monte Carlo), cung cấp góc nhìn phòng thủ và giới hạn mức tự tin vào dự báo.

Từ góc nhìn hệ thống, Hội đồng chuyên gia có thể được mô tả bởi ánh xạ:

$$\mathcal{C} : \underbrace{\text{FPT_train (OHLCV)}}_{\text{lịch sử giá & khối lượng}} \longrightarrow \underbrace{\hat{P}_{t+1:t+H}^{\text{final}}}_{\text{dự báo cuối cùng, } H = 100},$$

trong đó quyết định $\hat{P}_{t+1:t+H}^{\text{final}}$ không phải là đầu ra của một mô hình duy nhất, mà là kết quả của một *bộ phiếu có trọng số* giữa ba chuyên gia:

$$\hat{P}_k^{\text{final}} = w_{\text{dyn}} \cdot \hat{P}_k^{\text{dyn}} + w_{\text{stat}} \cdot \hat{P}_k^{\text{stat}} + w_{\text{risk}} \cdot \hat{P}_k^{\text{risk}}, \quad k = 1, \dots, H,$$

với $w_{\text{dyn}}, w_{\text{stat}}, w_{\text{risk}} \geq 0$ và $w_{\text{dyn}} + w_{\text{stat}} + w_{\text{risk}} = 1$. Trong thực nghiệm, Dynamic Expert thường được gán trọng số lớn nhất (mũi nhọn tấn công), Static Expert giữ vai trò neo xu hướng, còn Risk Expert chủ yếu điều chỉnh biên độ tự tin và đánh dấu các vùng rủi ro cao.

Sơ đồ hội đồng chuyên gia. Để trực quan hóa cơ chế ra quyết định, ta mô tả Hội đồng bằng sơ đồ khối sau:

Các phần tiếp theo sẽ lần lượt mô tả chi tiết:

- Dynamic Expert như một *mô hình động* gắn với pipeline Hybrid + Pricing-best,
- Static Expert như một *môc neo* dựa trên đường trend dài hạn,
- Risk Expert như một *bộ phận kiểm soát rủi ro* dựa trên dải bất định và Monte Carlo.

Nhờ tách bạch rõ vai trò như vậy, hệ thống có thể đưa ra dự báo 100 ngày *vừa nhạy với thị trường, vừa đủ thận trọng*, thay vì đặt cược tất tay vào một “người hùng” duy nhất.

1.1 Dynamic Experts (Mô hình Động)

Định nghĩa. Dynamic Expert chính là đường **BASE Path** – đầu ra cuối cùng của mô hình *Hybrid + Pricing-best* sau khi đã ghép nối cả ba lớp: Math Backbone, XGBoost Residual và Pricing Layer. Nói cách khác, đây không phải là một thuật toán đơn lẻ, mà là một **pipeline tích hợp** đã được hiệu chỉnh qua time-series CV để trở thành một “chuyên gia” duy nhất có tiếng nói cuối cùng về quỹ đạo giá tương lai.

Trong bức tranh tổng thể của mô hình 3 lớp (Mục 1), Dynamic Expert chính là *sự cộng hưởng* của:

- **Math Backbone** (Lớp 1): cung cấp “xương sống” xu hướng dài hạn trên log-price.
- **XGBoost Residual** (Lớp 2): chỉnh sửa sai số ngắn hạn quanh xu hướng, bắt các pattern phi tuyến trong dữ liệu tabular.
- **Pricing Layer** (Lớp 3): áp dụng các quy tắc tài chính (clipping, damping, mean reversion, regime-aware) để biến chuỗi log-return dự báo thành một đường giá biết *tự kiềm chế*.

Chỉ khi ba lớp này được ghép tuần tự và bộ tham số Pricing được chọn bằng time-series CV (Mục ??), ta mới thực sự có một **mô hình động** đúng nghĩa: biết nhìn quá khứ, học từ sai số, và tự điều chỉnh hành vi theo từng bối cảnh thị trường.

1.1.1 Dòng chảy thông tin: từ lịch sử đến BASE Path

Từ góc nhìn “chuyên gia”, toàn bộ pipeline Hybrid + Pricing có thể được coi là một ánh xạ

$$\mathcal{E}_{\text{dyn}} : \underbrace{\text{FPT_train (OHLCV)}}_{\text{lịch sử giá & khối lượng}} \longrightarrow \underbrace{\{\hat{P}_{t+1}, \dots, \hat{P}_{t+H}\}}_{\text{BASE Path, } H = 100}.$$

Luồng xử lý diễn ra theo các bước (xem chi tiết tại Mục ?? và 1):

1. **Feature Engineering (FE)**: từ chuỗi OHLCV thô, mô hình trích xuất các đặc trưng về:

- xu hướng (STL trend, trend slopes),
- biến động (volatility, ATR, Parkinson),
- price action (body, range, shadows, gaps, pattern nén),
- hành vi dòng tiền (volume, money flow, OBV).

2. **Hybrid raw path**:

- Math Backbone sinh ra log-return “theo sách vở” ở từng bước (linear trend trên log-price).
- XGBoost học $\text{resid_ret} = \text{future_ret} - \text{math_ret}$ và cộng ngược lại để tạo thành chuỗi *raw hybrid returns* phản ánh nhiễu ngắn hạn.

3. **Pricing Layer**: với cùng một raw path, Pricing sẽ:

- *giới hạn tốc độ* (clipping) dựa trên phân phối $|\text{ret_1d}|$ lịch sử,
- *giảm dần ảnh hưởng xa* (damping theo half-life),
- *kéo giá về vùng cân bằng* quanh MA60 (mean reversion),
- và điều chỉnh cường độ các cơ chế trên tùy theo chế độ thị trường BULL/BEAR/SIDEWAYS (regime-aware).

Kết quả cuối cùng chính là chuỗi giá dự báo **BASE Path** – tiếng nói tổng hợp của Dynamic Expert.

1.1.2 Tính “động” của chuyên gia Hybrid

Điểm làm nên tên gọi *Dynamic* không nằm ở kiến trúc phức tạp, mà ở **cách mô hình phản ứng với thông tin mới**:

- **Học từ sai số (error-driven)**: mỗi ngày, XGBoost không dự đoán giá trực tiếp mà dự đoán resid_ret – phần lệch so với đường trend. Khi cấu trúc dao động ngắn hạn thay đổi, resid_ret cũng đổi dấu / đổi biên độ, buộc mô hình cập nhật “hiểu biết” của mình về nhiễu mới.
- **Nhạy với chế độ thị trường (regime-aware)**: Pricing Layer không đối xử mọi giai đoạn như nhau. Cùng một raw return:
 - trong **BULL**, hệ thống cho phép biên độ tăng rộng hơn và mean reversion nhẹ tay hơn;
 - trong **BEAR**, clipping chặt hơn và lực kéo về vùng cân bằng mạnh hơn;
 - trong **SIDEWAYS**, mô hình ưu tiên giữ giá trong “hộp” dao động hợp lý.
- **Cơ chế “tự nghi ngờ” (self-doubt)**: bộ tham số Pricing không được chọn tay, mà được tối ưu bằng Random Search + time-series CV trên nhiều cutoff tháng 3/6/9/12 từ 2020–2024. Điều này buộc Dynamic Expert phải sống sót qua nhiều pha thị trường khác nhau trước khi được trao quyền dự báo 100 ngày tương lai.

Nếu coi toàn bộ hệ thống ensemble (Mục 2) là một đội ngũ nhiều chuyên gia, thì:

- **Dynamic Expert** là *mũi nhọn tấn công chính*: cung cấp quỹ đạo giá chi tiết, nhạy với turning points và non-linear patterns. Trong CENTRAL_DET, nó thường được gán trọng số lớn nhất (khoảng 70%).
- **Static Expert** (trend tuyến tính) đóng vai trò “chiến lược gia” dài hạn, tránh cho mô hình bị cuốn theo nhiều ngắn hạn.
- **Risk Expert** (uncertainty band, MC band) là lớp phòng thủ, nhắc hệ thống rằng mọi dự báo đều mang tính rủi ro.

Tóm lại, Dynamic Expert là hình hài “hoàn thiện” của mô hình Hybrid + Pricing: một trader ảo có *tầm nhìn xa* (trend), *phản xạ ngắn hạn* (residual ML) và *bộ phanh & vô lăng* (Pricing Layer) vận hành cùng lúc trên quỹ đạo 100 ngày.

1.2 Static Experts (Mô hình Tĩnh)

Nếu Chuyên gia 1 là “người kể chuyện” về những dao động phức tạp của giá FPT (ML Residual + Monte Carlo + Regime-aware Pricing), thì Chuyên gia 2 lại đóng vai trò như một *nha toán học bảo thủ*: chỉ nhìn đường xu hướng dài hạn và bỏ qua mọi ngắn hạn.

Về mặt kỹ thuật, Chuyên gia 2 chính là **Math Backbone** mà ta đã xây dựng từ đầu: một đường *linear trend* trên *log-price* của FPT được ước lượng bằng hồi quy tuyến tính đơn giản:

$$\log(P_t) \approx a + b \cdot t,$$

trong đó:

- P_t : giá đóng cửa tại thời điểm t ,
- t : chỉ số thời gian ($0, 1, 2, \dots$),
- a, b : tham số được fit từ toàn bộ lịch sử FPT.

Từ đường trend này, ta suy ra **tỷ suất sinh lời “xuyên nhiễu”** cho H ngày tới:

$$r_{t \rightarrow t+H}^{\text{trend}} = \log(\hat{P}_{t+H}^{(\text{trend})}) - \log(\hat{P}_t^{(\text{trend})}),$$

và từ đó dựng được một đường giá dự báo *tĩnh* trong tương lai mà không phụ thuộc vào dao động zig-zag mỗi ngày. Điểm quan trọng là: mặc dù Math Backbone đã được sử dụng để tạo ra `math_ret` cho Chuyên gia 1, ta vẫn cố ý tách nó thành một “Chuyên gia” độc lập trong bước Ensemble. Lý do không phải vì lười mà vì đây là **một quyết định quản trị rủi ro có chủ đích**.

1.2.1 Độc lập với nhiễu – một góc nhìn “sạch” về FPT

Trong phần EDA, khi quan sát:

- **ACF/PACF của returns**: gần như nhiễu trắng, các lag 1, 2, 3 chỉ có tương quan rất yếu.
- **Phân phối returns**: lệch, heavy-tail, nhiều outlier $\pm 5\% - 7\%$.
- **Decomposition STL**: trend log-price của FPT tăng khá đều, trong khi phần residual chứa phần lớn nhiễu ngắn hạn.

Điều này nói một câu chuyện rất rõ:

“FPT là một cổ phiếu có *drift dài hạn dương khá ổn định*, nhưng returns ngày thì gần như nhiễu trắng và có tails cực đoan.”

Các mô hình như XGBoost ở Chuyên gia 1 cố gắng tận dụng mọi tín hiệu từ:

- price action (OHLC, shadows, gaps),
- volume & money flow,
- pattern up/down, 3-streak,
- trend slopes, resid volatility,

để học cách phản ứng với các cú giật giá ngắn hạn. Trong khi đó, Chuyên gia 2 nói:

“Tôi không tranh luận với nhiễu. Tôi chỉ nhìn quỹ đạo dài hạn mà FPT đã đi suốt 20 năm qua.”

Vì vậy:

- Chuyên gia 2 **cố ý không sử dụng** bất kỳ feature high-frequency nào (ret_1d, body, range, vol_z, ...).
- Đầu ra của nó là một đường giá *mượt, đơn điệu hơn*, phản ánh đúng **câu chuyện tăng trưởng** của FPT theo thời gian.

Trong bối cảnh thị trường Việt Nam, nơi biên độ mỗi phiên bị giới hạn ($\pm 7\%$ HOSE, $\pm 10\%$ HNX, $\pm 15\%$ UPCoM) nhưng tin tức có thể tạo ra những cú sốc ngắn hạn rất mạnh, một “góc nhìn sạch” như vậy là cực kỳ quý giá để:

- tách “*business growth story*” của FPT khỏi *nhiều trading hàng ngày*,
- luôn nhắc mô hình rằng: “**về dài hạn, FPT đã đi lên như thế nào**”.

1.2.2 Neo giá trị trong Ensemble – giảm phương sai, giữ kỷ luật

Khi kết hợp nhiều chuyên gia trong Ensemble, nếu tất cả đều nhạy với nhiễu, mô hình tổng sẽ có **variance rất cao**: một cụm outlier trong returns có thể khiến dự báo 100 ngày sau “bật tung” so với dữ liệu thật.

Ở đây, Chuyên gia 2 đóng vai trò như một **Value Anchor** – một “neo giá trị” được gắn chặt vào quỹ đạo drift lịch sử của FPT:

- Nếu Chuyên gia 1 (ML + Monte Carlo + Pricing) dự báo quá hưng phấn trong bull, Chuyên gia 2 sẽ kéo Ensemble về gần đường trend.
- Nếu Chuyên gia 1 quá bi quan trong bear (overreact với volatility ngắn hạn), Chuyên gia 2 nhắc rằng về lịch sử FPT *hiếm khi bị phá vỡ hoàn toàn câu chuyện tăng trưởng dài hạn*.

Toán học mà nói, Chuyên gia 2 giống như một **số hạng regularization** trong Ensemble:

$$\hat{P}_t^{(\text{ensemble})} = w_1 \cdot \hat{P}_t^{(\text{Expert 1})} + w_2 \cdot \hat{P}_t^{(\text{Expert 2})} + \dots$$

với w_2 nhỏ nhưng *rất ổn định theo thời gian*. Trong đó:

- $\hat{P}_t^{(\text{Expert } 1)}$: linh hoạt, high-variance (nhạy với pattern ngắn hạn, regime, volume spike).
- $\hat{P}_t^{(\text{Expert } 2)}$: low-variance, gần như deterministic (chỉ thay đổi theo trend).

Kết quả:

- Ensemble **giảm được phương sai** mà không phải hy sinh quá nhiều bias, vì drift dài hạn đã được backbone ước lượng tốt.
- Đường dự báo cuối cùng nhìn rất giống một nhà quản lý danh mục chuyên nghiệp: vẫn tôn trọng xu hướng lớn, nhưng không bị “tâm lý FOMO/PANIC” của các phiên lẻ làm xoắn.

1.2.3 Fail-safe Mechanism – khi ML “đi lạc”, toán vẫn đứng vững

Một lý do nữa để tách Chuyên gia 2 thành mô hình độc lập thay vì chỉ coi đó là “một feature” trong Chuyên gia 1: **nó là một lớp bảo hiểm (fail-safe) cho toàn bộ hệ thống**.

Trong thực tế, các mô hình ML (đặc biệt với feature phức tạp, regime-detection, pricing logic) có thể gặp những rủi ro sau:

- Data future nằm ngoài vùng phân phối train (out-of-sample shift),
- Một pattern mới xuất hiện mà XGBoost chưa từng thấy,
- Regime detection bị đánh giá sai (ví dụ bull nhưng thị trường thực tế đang tạo đỉnh),
- Một tham số Pricing được random search chọn tốt cho quá khứ nhưng không còn phù hợp cho một giai đoạn đặc biệt trong tương lai.

Trong tất cả các kịch bản đó, chuyên gia 1 có thể *dự báo sai rất mạnh* nếu không được kiểm soát. Ngược lại, Chuyên gia 2:

- chỉ dựa trên **trend log-price dài hạn**,
- không học từ pattern ngắn hạn,
- không phụ thuộc vào hyper-parameter phức tạp.

Do đó, về mặt toán học, Chuyên gia 2 là thành phần **ổn định nhất** trong toàn bộ hệ thống. Khi ensemble, ngay cả trong tình huống xấu nhất:

- Nếu Expert 1 “mất phương hướng”,
- Nếu Regime-aware Pricing đang hoạt động sai vì một shock quá khác thường,

thì ta vẫn còn một đường dự báo mang ý nghĩa:

“Nếu bỏ qua tất cả drama ngắn hạn, và chỉ tin vào quỹ đạo tăng trưởng lịch sử của FPT, thì giá hợp lý sau 100 ngày là khoảng này.”

Đây chính là tinh thần quản trị rủi ro:

- Không bao giờ đặt toàn bộ niềm tin vào một mô hình phức tạp.
- Luôn giữ một thành phần “simple but robust” để hệ thống không bao giờ đưa ra dự báo vô nghĩa.

1.2.4 Từ Math Backbone đến “Static Agent” – quyết định mang tính quản trị, không phải sự lười biếng

Nhìn bề ngoài, có thể ai đó sẽ hỏi:

“Đã dùng trend trong Math Backbone rồi, sao còn tách nó ra làm một chuyên gia riêng? Có phải chỉ là nhân đôi cùng một thứ?”

Câu trả lời là **không**. Về mặt kiến trúc:

- Trong Chuyên gia 1, Math Backbone được dùng để tạo `math_ret` và `resid_ret`, giúp XGBoost tập trung vào nhiều ngắn hạn.
- Trong Chuyên gia 2, Math Backbone trở thành **một mô hình hoàn chỉnh** với quyền biểu quyết trong Ensemble, độc lập với mọi sai số của ML residual, Monte Carlo, Pricing.

Về mặt tư duy thiết kế hệ thống:

- Đây là một quyết định có chủ đích để:
 - giảm variance,
 - cung cấp một neo giá trị ổn định,
 - và tạo một cơ chế fail-safe khi mọi mô-đun phức tạp gặp vấn đề.
- Nói cách khác: ta dùng **Trend** không phải vì “dễ code”, mà vì **muốn hệ thống có tư duy quản trị rủi ro giống một nhà quản lý danh mục thực thụ**.

Trong thị trường chứng khoán Việt Nam, nơi:

- tin tức, dòng tiền, room ngoại, ...có thể tạo ra nhiều cực mạnh trong ngắn hạn,
- nhưng câu chuyện dài hạn (business fundamentals, tăng trưởng EPS, mở rộng thị trường) vẫn là thứ quyết định giá cổ phiếu 5–10 năm,

việc đặt một “Static Agent” như Chuyên gia 2 vào hệ thống không chỉ hợp lý về mặt toán, mà còn phản ánh đúng **triết lý đầu tư**:

“Học để dự báo nhiều là tốt. Nhưng không bao giờ được quên đường xu hướng dài hạn mà doanh nghiệp đã xây suốt nhiều năm.”

1.3 Risk Experts (Mô hình Rủi ro)

Hai chuyên gia đầu tiên tập trung vào *giá trị trung tâm* của dự báo:

- **Dynamic Expert ($Price_t^{dyn}$)**: đường giá “thông minh” nhất, là kết quả của toàn bộ pipeline Hybrid (Backbone + ML Residual + Pricing Layer).
- **Static Expert ($Price_t^{stat}$)**: đường xu hướng dài hạn, đóng vai trò “neo bảo thủ” của mô hình.

Tuy nhiên, trong thực tế giao dịch, nhà đầu tư không chỉ quan tâm đến câu hỏi

“Giá dự báo cho ngày mai là bao nhiêu?”

mà còn quan tâm đến câu hỏi

“Nếu mọi chuyện không diễn ra đúng như kỳ vọng, giá có thể lệch đi bao xa? Độ bất định của dự báo là bao nhiêu?”

Do đó, ngoài một đường dự báo điểm (point forecast), ta cần thêm một thành phần chuyên mô tả *rủi ro* và *bất định* của tương lai. Đặt trong ngôn ngữ “hội đồng chuyên gia”, đây chính là vai trò của **Chuyên gia 3 – Risk Expert**.

Risk Expert không cố gắng đưa ra một dự báo mới độc lập, mà đóng vai trò như *chuyên gia quản trị rủi ro*: dựa trên pipeline Hybrid hiện có, mô phỏng nhiều kịch bản tương lai khác nhau, từ đó trích xuất một đường trung tâm và một dải bất định (uncertainty band).

1.3.1 Ý tưởng tổng quát: Monte Carlo và “Uncertainty Center”

Giả sử sau Pricing Layer, chúng ta có được một đường giá đã kiểm soát:

$$Price_t, \quad t = 1, \dots, T,$$

và từ đó suy ra được phần sai lệch (residual) so với giá thực tế S_t :

$$\varepsilon_t = S_t - Price_t. \quad (12)$$

Trực giác: $Price_t$ biểu diễn “kịch bản cơ sở” (baseline scenario), còn ε_t là phần nhiễu và bất định mà mô hình không nắm bắt hết.

Ý tưởng xây dựng Risk Expert như sau:

1. Trên dữ liệu lịch sử, ước lượng phân bố của residual ε_t (có thể phụ thuộc chế độ thị trường).
2. Dựa trên phân bố này, mô phỏng N kịch bản tương lai cho H ngày tới bằng *Monte Carlo*.
3. Tại mỗi horizon $h = 1, \dots, H$, lấy *median* (hoặc một quantile phù hợp) của phân bố giá trong N kịch bản làm “đường trung tâm rủi ro”.

Đường trung tâm đó được gọi là $Price_h^{\text{risk}}$ – dự báo của Risk Expert ở bước h . Đồng thời, các quantile thấp và cao (ví dụ 5% và 95%) cho ta một dải bất định bao quanh, dùng để đánh giá mức độ tin cậy của dự báo.

1.3.2 Ước lượng phân bố nhiễu và chế độ thị trường

Trên đoạn lịch sử $t = 1, \dots, T$, ta tính residual:

$$\varepsilon_t = S_t - Price_t. \quad (13)$$

Dựa trên phần phân tích chế độ thị trường (regime) ở các mục trước, mỗi thời điểm t được gán vào một chế độ:

$$Regime_t \in \{\text{Bull}, \text{Bear}, \text{Sideway}\}.$$

Thay vì giả định một phân phối Gauss cố định cho mọi giai đoạn, ta cho phép tính chất của nhiễu thay đổi theo chế độ:

- Trong **Bull** regime: residual thường nhỏ hơn, volatility thấp hơn,
- Trong **Bear** regime: residual có thể lớn, đuôi dày hơn, phản ánh các cú sụt mạnh.

Về mặt thực thi, có thể sử dụng hai cách tiếp cận:

1. **Giả định phân phối tham số:** ước lượng μ_r, σ_r cho từng chế độ r và lấy mẫu $\varepsilon_t \sim \mathcal{N}(\mu_{Regime_t}, \sigma_{Regime_t}^2)$.
2. **Bootstrap residual:** trong mỗi chế độ r , tập hợp các residual $\{\varepsilon_t : Regime_t = r\}$ và lấy mẫu lai (sampling with replacement) trực tiếp từ bộ giá trị này. Cách tiếp cận này giảm bớt giả định về dạng phân phối và giữ được đặc điểm đuôi dày nếu có.

1.3.3 Mô phỏng Monte Carlo các đường giá tương lai

Giả sử chúng ta muốn dự báo H ngày tới kể từ thời điểm T . Gọi Price_T là giá đã được Pricing Layer kiểm soát ở ngày cuối cùng trong tập train/validation. Ta sinh N kịch bản tương lai độc lập, ký hiệu là

$$\text{Path}_h^{(i)}, \quad i = 1, \dots, N, \quad h = 0, \dots, H,$$

với điều kiện ban đầu

$$\text{Path}_0^{(i)} = \text{Price}_T, \quad \forall i.$$

Đối với mỗi kịch bản i và mỗi bước $h = 1, \dots, H$, tiến hành:

- Xác định chế độ thị trường dự kiến cho bước h (ví dụ dựa trên Dynamic Expert hoặc một mô hình regime đơn giản):

$$\text{Regime}_{T+h}.$$

- Từ phân bố residual tương ứng với chế độ đó, lấy mẫu một nhiễu mới:

$$\varepsilon_h^{(i)} \sim \mathcal{D}_{\text{Regime}_{T+h}}, \quad (14)$$

trong đó $\mathcal{D}_{\text{Regime}_{T+h}}$ là phân bố (hoặc tập residual để bootstrap) đã ước lượng.

- Lấy forecast “cơ sở” $\tilde{\text{Price}}_{T+h}$ từ mô hình Dynamic/Pricing Layer (có thể là dự báo một bước hoặc dự báo theo kiểu recursive).

- Xây dựng giá mô phỏng:

$$\text{Path}_h^{(i)} = \tilde{\text{Price}}_{T+h} + \varepsilon_h^{(i)}. \quad (15)$$

Sau H bước, ta thu được một ma trận kích thước $N \times H$:

$$(\text{Path}_h^{(i)})_{i=1, \dots, N}^{h=1, \dots, H},$$

trong đó mỗi hàng là một “thế giới song song” thể hiện diễn biến giá FPT trong tương lai nếu nhiều diễn ra theo một chuỗi cụ thể.

1.3.4 Định nghĩa đường giá của Risk Expert

Tại mỗi horizon h , chúng ta coi $\{\text{Path}_h^{(i)}\}_{i=1}^N$ là các quan sát từ phân bố giá tương lai tại thời điểm $T+h$. Hai đại lượng quan trọng được rút ra từ tập giá trị này là:

- Đường trung tâm rủi ro** (Uncertainty Center):

$$\text{Price}_{T+h}^{\text{risk}} = \text{Median}\{\text{Path}_h^{(i)} : i = 1, \dots, N\}. \quad (16)$$

- Dải bất định** (uncertainty band), ví dụ 5%–95% quantile:

$$L_{T+h}^{(0.05)} = Q_{0.05}(\{\text{Path}_h^{(i)}\}), \quad (17)$$

$$U_{T+h}^{(0.95)} = Q_{0.95}(\{\text{Path}_h^{(i)}\}), \quad (18)$$

trong đó Q_p ký hiệu quantile bậc p .

Chuỗi $(\text{Price}_{T+h}^{\text{risk}})_{h=1}^H$ chính là đầu ra của Risk Expert. Không giống Dynamic Expert, đường này không được thiết kế để tối thiểu hóa sai số bình phương, mà để phản ánh vị trí “diễn hình” của giá trong bối cảnh có rủi ro và nhiễu.

1.3.5 Vai trò của Risk Expert trong cơ chế bỏ phiếu

Trong cơ chế bỏ phiếu cuối cùng, dự báo của mô hình được viết dưới dạng tổ hợp tuyến tính:

$$Price_{T+h}^{\text{final}} = w_1 Price_{T+h}^{\text{dyn}} + w_2 Price_{T+h}^{\text{stat}} + w_3 Price_{T+h}^{\text{risk}}, \quad (19)$$

với $w_1 + w_2 + w_3 = 1$ và $w_k \geq 0$.

Trong đó:

- $Price_{T+h}^{\text{dyn}}$ mang tiếng nói của “chuyên gia thông minh” đã được tối ưu hóa mạnh về mặt sai số,
- $Price_{T+h}^{\text{stat}}$ giữ vai trò “neo xu hướng” dài hạn,
- $Price_{T+h}^{\text{risk}}$ là tiếng nói thận trọng, nhắc nhở mô hình về rủi ro và độ bất định của thị trường.

Trọng số w_3 thường được chọn nhỏ hơn w_1 , nhưng vẫn đủ để “kéo” đường dự báo cuối cùng về vùng an toàn hơn trong những giai đoạn thị trường hỗn loạn, khi dải bất định $[L_{T+h}^{(0.05)}, U_{T+h}^{(0.95)}]$ nở rộng đáng kể. Ngược lại, khi thị trường ổn định và dải Monte Carlo hẹp, Risk Expert gần như trùng với Dynamic Expert, và đóng góp của nó vào $final$ trở nên ít can thiệp hơn.

Nhờ sự hiện diện của Risk Expert, mô hình không chỉ đưa ra một con số dự báo duy nhất, mà còn cung cấp thêm một “bức tranh rủi ro” đi kèm, giúp việc ra quyết định giao dịch và quản trị vị thế trở nên thực tế và thận trọng hơn.

2 Cơ chế Ensemble

Dự báo giá cổ phiếu trong 100 ngày không chỉ là bài toán “tối ưu một mô hình”. Trong bối cảnh thị trường nhiễu, nhiều pha đảo chiều và không ổn định về chế độ, việc đặt niềm tin tuyệt đối vào một mô hình đơn lẻ luôn tiềm ẩn rủi ro *Single Point of Failure*. Thay vào đó, project lựa chọn cách tiếp cận theo **Hội đồng Chuyên gia** (*Council of Experts*), nơi nhiều góc nhìn mô hình hoá khác nhau được kết hợp có kiểm soát bằng **Weighted Ensemble**.

Tư duy cốt lõi ở đây là **đa dạng hoá hệ tư tưởng mô hình** (*cognitive diversification*): không chỉ đa dạng hoá dữ liệu hay đặc trưng, mà đa dạng cả triết lý dự báo đứng sau từng chuyên gia: *nhạy bén, bảo thủ và thận trọng với rủi ro*.

2.1 Weighted Ensemble

Công thức tổng hợp dự báo cuối cùng được chúng tôi xây dựng dưới dạng trung bình có trọng số:

$$Y_{\text{final}} = 0.7 \cdot Y_{\text{dynamic}} + 0.25 \cdot Y_{\text{static}} + 0.05 \cdot Y_{\text{risk}}.$$

Trong đó:

- Y_{dynamic} : BASE Path từ **Dynamic Expert** (Hybrid + Pricing-best, xem Mục 1.1).
- Y_{static} : đường **Static Trend** tuyến tính trên price, đóng vai trò neo dài hạn (Mục 1.2).
- Y_{risk} : **Risk Expert** (UNCERT analytic / MC median), phản ánh góc nhìn xác suất về rủi ro (Mục 1.3).

Tại sao lại là 70–25–5? Bộ trọng số này không phải một “định luật cứng”, mà là lựa chọn thể hiện rõ triết lý đánh đổi **Bias–Variance** của hệ thống:

- **Dynamic (70%):** Đây là mô hình mang tính *Alpha* – tìm kiếm lợi thế dự báo vượt trội dựa trên ML + Pricing. Phương sai cao (*variance lớn*), *bias thấp*, bám sát biến động thị trường nhưng dễ bị overfit nếu để chiếm 100% tiếng nói.
- **Static (25%):** Đóng vai trò *Beta / Core Anchor*: bám theo xu hướng dài hạn nội tại của cổ phiếu FPT. *Bias cao, variance thấp*, tạo lực kéo hồi quy (*gravitational anchor*) để dự báo không bị trôi quá xa khỏi quỹ đạo tăng trưởng.
- **Risk (5%):** Lớp hiệu chỉnh mang tính xác suất, tương đương một dạng *regularization* ở cấp độ hệ thống. Tỷ trọng nhỏ đủ để giảm rủi ro sai số mang tính hệ thống (*systemic bias*), nhưng không đủ lớn để làm nhiễu tín hiệu chính.

Bộ ba 70–25–5 có thể được xem như một biến thể của **Core–Satellite Strategy** trong tài chính định lượng:

- Dynamic → *Satellite chủ động* (*Alpha-seeking*).
- Static + Risk → *Core phòng thủ* (giữ cấu trúc dài hạn, quản trị rủi ro).

Khác với phân bổ danh mục đầu tư truyền thống (thường nghiêm về an toàn), ở bài toán dự báo, trọng số được xoay chiều để ưu tiên *độ chính xác dự báo* nhưng vẫn giữ được “vùng đệm an toàn” quanh quỹ đạo Trend.

Liên hệ với Ensemble trong Machine Learning. Trong ML truyền thống, các kỹ thuật như Bagging, Random Forest, Gradient Boosting hay Stacking đều dựa trên nguyên tắc:

- Kết hợp các mô hình có *variance cao* với các mô hình *bias cao*,
- Giảm tương quan lỗi (*error correlation*) giữa các thành phần,
- Tối ưu hoá Bias–Variance tradeoff ở cấp hệ thống.

Ensemble trong đồ án này cũng tuân theo tinh thần đó, nhưng ở tầng khái niệm cao hơn: thay vì nhiều bản sao cùng kiến trúc, ta kết hợp ba **hệ tư tưởng mô hình hóa** khác nhau:

- Dynamic ↔ mô hình *nhạy, phi tuyến*, giống một cây nồng (shallow tree) trong Random Forest nhưng được trang bị thêm Pricing Layer.
- Static ↔ mô hình *đơn giản, ổn định*, tương tự thành phần có bias cao trong các hệ Boosting dùng để giữ cấu trúc nền.
- Risk ↔ một lớp *stochastic correction* – tương tự cơ chế bagging/dropout giúp giảm overconfidence và làm mượt phân phối dự báo.

Do ba Expert được xây dựng trên các giả định, kiến trúc và không gian đặc trưng khác nhau, **hệ số tương quan lỗi** giữa chúng thấp hơn rất nhiều so với việc chỉ lắp ghép các bản sao cùng kiểu. Đây chính là yếu tố giúp Ensemble giữ được hiệu năng ở tầm dự báo 100 ngày, nơi sai số tích luỹ (error accumulation) thường là điểm chết của mô hình đơn lẻ.

2.2 Lý giải vai trò từng Expert

Hội đồng Chuyên gia không chỉ là ba mô hình “đứng cạnh nhau”, mà là một **cơ chế kiểm soát và đối trọng** (*Check & Balance*). Mỗi Expert được thiết kế với một *nhiệm vụ chiến lược* khác nhau, và quan trọng hơn: chúng *kìm hãm và bù trừ* cho nhau.

Expert	Vai trò chiến lược	Cơ chế bù trừ (Synergy)
Dynamic	Mũi tấn công (Striker): Bám sát cấu trúc thị trường, nhạy với đảo chiều, tận dụng tối đa thông tin từ FE + residual learning + Pricing.	Nếu chạy quá đà (overfit, phóng đại nhiễu ngắn hạn) → Static tạo lực kéo về xu hướng nền, giảm độ lệch khỏi quỹ đạo dài hạn.
Static	Mỏ neo (Anchor): Tái tạo xu hướng dài hạn, giữ dự báo bám theo “giá trị cốt lõi” của FPT, hạn chế lag quá mức.	Nếu quá ì (underfit, phản ứng chậm với biến động mới) → Dynamic đẩy nhanh tốc độ thích ứng, giúp bắt kịp chế độ thị trường mới.
Risk	Lớp đệm rủi ro (Buffer): Phản ánh bất định ngẫu nhiên, cung cấp góc nhìn xác suất về vùng giá thay vì một điểm dự báo duy nhất.	Nếu cả Dynamic và Static cùng sai lệch theo một hướng (systemic bias) → Risk giảm cường độ tín hiệu (qua band/median), hạn chế sai số tích luỹ và giảm overconfidence của hệ thống.

Bảng 2: Ma trận vai trò và tương tác giữa ba chuyên gia trong Ensemble.

Một cách hình tượng. Có thể coi:

- Dynamic là *Trader* tuyển đầu – quyết đoán, phản ứng nhanh.
- Static là *Chiến lược gia* – giữ tầm nhìn dài hạn, không chạy theo nhiễu.
- Risk là *Nhà quản trị rủi ro* – luôn đặt câu hỏi “Nếu sai thì sao?”.

Ba nhân vật này hiếm khi đồng thuận tuyệt đối, và chính sự bất đồng có kiểm soát đó giúp hệ thống trở nên bền vững hơn trước các cú sốc thị trường.

Khi đưa vào thực chiến. Trong bối cảnh dự báo 100 ngày:

1. **Dynamic** đảm nhiệm tính *linh hoạt*: bám sát biến động, tận dụng cấu trúc residual + regime-aware Pricing để mô phỏng nhịp điệu thị trường.
2. **Static** giữ *nền tảng*: bảo đảm rằng, dù mô hình có “phiêu” thế nào, dự báo vẫn xoay quanh quỹ đạo tăng trưởng dài hạn của FPT.
3. **Risk** đảm bảo *bên độ an toàn*: biến dự báo thành một dải (*forecast band*), giúp người dùng đánh giá kịch bản xấu/tốt thay vì tin vào một con số duy nhất.

Ba lực này tạo thành một **tam giác cân bằng**: *tốc độ* (Dynamic), *ổn định* (Static) và *an toàn* (Risk). Chính cấu trúc này cho phép hệ thống không chỉ dự báo “đúng hơn”, mà còn dự báo theo cách *có thể tin được* trong bối cảnh thị trường luôn bất định.

PHẦN 4: KẾT QUẢ & ĐÁNH GIÁ (Evaluation)

1 Kết quả dự báo

Pipeline thiếu tính năng đánh giá tin tức (News Sentiment): Đây là điểm yếu chí mạng của mọi model thuần kỹ thuật.

1.1 Thiết lập thí nghiệm dự báo 100 ngày

Ở bước cuối cùng của pipeline, mô hình được huấn luyện trên **toàn bộ lịch sử giá FPT** trong tập `FPT_train` sau khi loại bỏ các phần dữ liệu kỹ thuật không cần thiết. Thời gian huấn luyện trải dài từ

2020-08-03 → 2025-03-10,

tổng cộng 1 149 phiên giao dịch.

Dữ liệu được nạp và xử lý như sau:

```
1 df_train = pd.read_csv(FPT_TRAIN_PATH)
2 df_train["time"] = pd.to_datetime(df_train["time"])
3 df_train = df_train.sort_values("time").reset_index(drop=True)
4 print(df_train["time"].min(), df_train["time"].max(), df_train.shape)
```

Việc `FORCE_TRAIN_END_DATE = None` có nghĩa là ta *cố ý* không dùng train sớm, mà để mô hình nhìn trọn vẹn các pha quan trọng của cổ phiếu FPT trên HOSE:

- Giai đoạn tích luỹ quanh vùng 20–30 năm 2020–2021.
- Các nhịp tăng mạnh sau thời kỳ COVID.
- Sóng tăng rất mạnh 2023–2024, đưa giá lên vùng đỉnh lịch sử.
- Cú điều chỉnh sâu đầu 2025, kéo giá FPT rơi nhanh khỏi đỉnh.

Chính trong bối cảnh đó, bài toán của ta là **dự báo 100 ngày tiếp theo** kể từ sau phiên 2025-03-10, tức là ngay sau một cú rơi mạnh – thời điểm mà tâm lý nhà đầu tư thường phân vân giữa “hồi phục” và “giảm tiếp”.

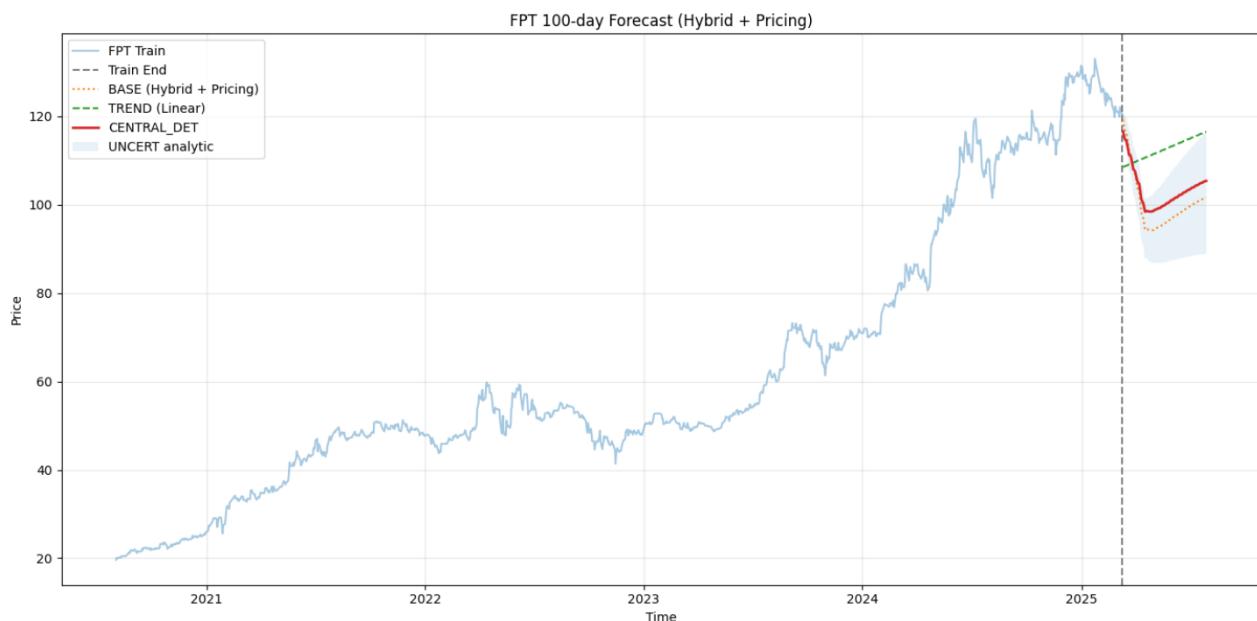
1.2 Bức tranh trực quan: Hybrid + Pricing + Trend + Uncertainty

Hình 24 minh họa kết quả cuối cùng với tiêu đề “*FPT 100-day Forecast (Hybrid + Pricing)*”. Trên một biểu đồ, ta thấy:

- **Đường xanh nhạt** là giá đóng cửa lịch sử của FPT (`FPT Train`).
- **Vạch đứt màu xám** đánh dấu *ngày kết thúc train* – mốc tách quá khứ và tương lai.
- **Đường vàng chấm** biểu diễn *BASE (Hybrid + Pricing)*: đây là đường dự báo sau khi:
 - tính drift bằng **Math Backbone** (trend trên log-price) cộng với phần hiệu chỉnh phi tuyến từ **XGBoost residual**,
 - đi qua **Pricing Layer** có nhận thức regime (*regime-aware*), thực hiện clipping, damping và mean-reversion.

- **Đường xanh lá đứt đoạn** là *TREND (Linear)* – mô hình tuyến tính đơn giản trên giá đóng cửa, đóng vai trò “chuyên gia tĩnh” chỉ nắm xu hướng dài hạn.
- **Đường đỏ đậm** là *CENTRAL_DET* – đường dự báo trung tâm mà ta đề xuất sử dụng trong thực tế:

$$\text{central_det} = 0.7 \cdot \text{BASE} + 0.25 \cdot \text{TREND} + 0.05 \cdot \text{UNCERT_center}.$$
- **Vùng xanh nhạt** là *UNCERT analytic* – dải bất định được xây từ sai số residual của XGBoost và phân phối nhiễu đã ước lượng.



Hình 24: Kết quả dự báo

Quan sát hình, ta có thể kể lại câu chuyện như sau:

- Ngay sau vạch *Train End*, cả BASE lẫn CENTRAL_DET đều **chấp nhận cú rơi giá** mà FPT vừa trải qua, không cố “kéo” giá trở lại vùng đỉnh 130 một cách ảo tưởng. Điều này phản ánh đúng thực tế thị trường Việt Nam, nơi những cú điều chỉnh sau pha tăng nóng thường kéo dài nhiều tuần.
- Sau một đoạn giảm thêm, BASE có xu hướng đi thấp hơn và dao động mạnh hơn, trong khi TREND tiếp tục đi lên mượt mà. CENTRAL_DET nằm *giữa* hai chuyên gia này, thể hiện một kịch bản cân bằng: tạo đáy mới quanh vùng 95–100, rồi **hồi phục chậm** về vùng 103–107 trong 100 ngày.
- Vùng UNCERT analytic bao trùm quanh CENTRAL_DET, cho ta một “quạt dự báo” với:
 - biên trên (optimistic scenario) tiệm cận 110–115,
 - biên dưới (pessimistic nhưng vẫn hợp lý) nằm quanh 90–92.

Về mặt thực hành, ta coi **CENTRAL_DET** là **đường dự báo chính**:

- BASE mang thông tin ML + Pricing đầy đủ nhưng có thể nhạy cảm hơn với nhiều ngắn hạn.

- TREND là nòng cốt tuyến tính, giúp neo mô hình vào quỹ đạo tăng trưởng dài hạn của FPT – một doanh nghiệp công nghệ lớn, tăng trưởng lợi nhuận khá đều trong nhiều năm.
- CENTRAL_DET kết hợp cả hai góc nhìn theo tỷ trọng bảo thủ, tránh việc mô hình “overreact” mà vẫn tôn trọng lịch sử tăng trưởng của cổ phiếu.

1.3 Diễn giải kết quả cho người dùng cuối

Nếu nói lại cho nhà đầu tư hoặc bộ phận quản trị rủi ro, ta có thể tóm tắt:

Trong 100 phiên tới, mô hình dự báo rằng FPT: (i) có thể giảm thêm ngắn hạn và tạo đáy mới quanh vùng 95–100, sau đó (ii) hồi phục dần về vùng 103–107, với một dải bất định hợp lý nằm trong khoảng xấp xỉ 90–115.

Forecast này không nhằm đưa ra tín hiệu mua–bán tuyệt đối, mà là một **bản đồ định lượng** về các kịch bản giá khả dĩ, giúp người dùng đặt câu hỏi: “Nếu FPT đi ra ngoài vùng này, có phải cấu trúc thị trường đã thay đổi đến mức mô hình cần cập nhật lại không?”.

2 Đánh giá độ tin cậy

2.1 Cross-validation theo thời gian: mô hình có bền vững qua nhiều pha thị trường?

Để đánh giá độ tin cậy, ta không chỉ nhìn vào một lần fit duy nhất, mà kiểm tra xem mô hình có *dựng vững* khi được đặt vào quá khứ nhiều lần hay không. Cụ thể, ta xây dựng các mốc *time-based cross-validation*:

2021-09-30, 2021-12-31, 2022-03-31, ..., 2024-09-30.

Với mỗi cutoff:

1. Dùng dữ liệu *trước cutoff* để:

- build đầy đủ feature STL + OHLCV,
- train Math Backbone + XGBoost residual,
- chạy Pricing Layer.

2. Dùng 100 ngày *sau cutoff* làm tập kiểm tra out-of-sample cho cả residual và giá.

Kết quả cho phần residual của XGBoost có thể tóm lược như sau:

- **MAE trên tập Test** cho log-return phần dư hầu hết nằm trong khoảng 0.7%–2.6%/ngày.
- **Độ lệch chuẩn residual** dao động quanh 0.007–0.013.
- **Kỳ vọng residual** gần 0 (thường $|mean| < 0.002$), cho thấy mô hình không bị lệch bullish hay bearish một cách có hệ thống.

Đáng chú ý, các giai đoạn thị trường “khó nhần” nhất đối với nhà đầu tư Việt Nam, như:

- 2022-06-30 – sau cú sụt giảm mạnh toàn thị trường,
- 2024-06-28, 2024-09-30 – quanh đỉnh và sau đỉnh 2024,

đều cho độ lệch chuẩn residual khoảng 0.010–0.013 và MAE log-return 1.7%–2.6%. Nói cách khác, ngay cả khi VN-Index và nhóm cổ phiếu công nghệ bước vào pha biến động mạnh, phần nhiều mà mô hình không giải thích được vẫn *giữ được cấu trúc ổn định*.

Giai đoạn	MAE Test (%)	Std Residual
2021–2022	0.8–1.5	0.006–0.009
2022 (downtrend mạnh)	1.5–2.6	0.010–0.013
2023–2024 bull-run	0.7–1.3	0.007–0.009
2024–2025	1.7–1.9	0.010–0.013

Đây là mảnh ghép then chốt để dải bất định **UNCERT analytic** trở nên đáng tin: nó được xây từ một phân phối nhiễu đã được kiểm chứng qua nhiều pha thị trường, chứ không phải chỉ dựa vào một lần train trên toàn bộ dữ liệu.

2.2 Pricing Layer được tối ưu bằng Random Search + CV

Độ tin cậy của forecast giá không chỉ đến từ XGBoost, mà còn từ **Pricing Layer** – lớp điều chỉnh cuối bảo vệ mô hình khỏi những cú “vung tay quá trán”.

Thay vì đặt tay các tham số clipping, half-life, mean-revert, ta sử dụng *Random Search* kết hợp với chính các cutoff CV ở trên. Mỗi tổ hợp tham số **PricingParams**:

- sinh ra một đường giá dự báo sau Pricing cho từng cutoff,
- được chấm điểm bằng metric:

$$M = 0.5 \cdot \text{MSE}_{50} + 0.5 \cdot \text{MSE}_{100},$$

nơi MSE_{50} là sai số bình phương trung bình 50 ngày đầu, còn MSE_{100} là sai số trên toàn 100 ngày.

Sau hàng trăm lượt thử, bộ tham số tốt nhất (cho FPT thuần) có dạng:

```
ret_clip_quantile ≈ 0.967,
half_life_days ≈ 41,
mean_revert_alpha ≈ 0.064,
mean_revert_start ≈ 28,
fair_up_mult ≈ 1.46,
fair_down_mult ≈ 0.85,
trend_lookback ≈ 57,
trend_ret_thresh ≈ 0.235.
```

Một số diễn giải trực giác:

- **Half-life ≈ 41 ngày**: các cú “bung nổ” giá (tăng/giảm bất thường) được phép tồn tại nhưng sẽ bị giảm dần ảnh hưởng trong khoảng 2 tháng giao dịch – khá phù hợp với hành vi của cổ phiếu FPT trên HOSE.
- **fair_up / fair_down** khác xa 1: giá được phép dao động rộng quanh fair level (hơn 45% phía trên, khoảng 15% phía dưới) trước khi lực mean-revert kích hoạt, phản ánh thực tế là cổ phiếu tăng trưởng có thể bị định giá cao hơn giá trị nội tại trong một thời gian dài.

- **trend_lookback** dài (gần 3 tháng) và **trend_ret_thresh** lớn: chỉ khi xuất hiện xu hướng rất mạnh trong một khoảng thời gian đủ dài, Pricing Layer mới “chịu tin” rằng đây là một trend mới chứ không phải nhiễu.

Điều này giúp Pricing Layer trở thành một “bộ lọc hành vi” đã được hiệu chỉnh riêng cho FPT, thay vì chỉ là một lớp heuristics chung chung.

2.3 Regime hiện tại: SIDEWAYS sau điều chỉnh

Trước khi áp dụng Pricing, hệ thống thực hiện phân loại *regime* hiện tại của thị trường dựa trên:

- vị trí giá so với đường MA120 (xu hướng dài hạn),
- tỷ lệ giữa volatility 20 ngày gần nhất và volatility toàn lịch sử.

Ở thời điểm dự báo cuối cùng, hệ thống log lại:

```
[REGIME] last_price=120.11, MA120=121.50,
price_pos=-1.15%, vol_ratio=0.67 -> SIDEWAYS
```

Điễn giải:

- Giá hiện tại thấp hơn MA120 khoảng -1.15% – chưa đủ sâu để coi là *downtrend rõ rệt*.
- Volatility 20 ngày gần đây chỉ bằng 67% volatility dài hạn – thị trường đang lảng xuống sau cú điều chỉnh.
- Regime phù hợp nhất là **SIDEWAYS sau một pha giảm mạnh**.

Trong regime này, các hệ số của Pricing Layer được “dịch” về mức trung tính: không quá nói lồng như BULL, cũng không quá thắt chặt như BEAR. Vì vậy CENTRAL_DET vẽ ra kịch bản: giảm thêm một chút để hoàn thiện đáy mới, rồi hồi phục chậm, thay vì rơi tự do hoặc bật ngược về đỉnh.

2.4 Dải bất định & quản trị rủi ro

Dựa trên phân phối residual ổn định (`resid_std` khoảng 0.01) và các giả định log-return, ta xây được dải *UNCERT analytic* bao quanh CENTRAL_DET. Đối với người dùng cuối, cách đọc dải này đơn giản nhưng rất hữu ích:

- Nếu giá thực tế **nằm bên trong** vùng bất định (ví dụ 90–115): mô hình vẫn “hiểu” thị trường; chưa có bằng chứng mạnh mẽ rằng cấu trúc đã thay đổi.
- Nếu giá **thoát ra khỏi** vùng này một cách có hệ thống: đây là tín hiệu cảnh báo rằng:
 1. hoặc thị trường đang bước vào một chế độ hoàn toàn mới (ví dụ khủng hoảng, tin tức cực lớn),
 2. hoặc các quan hệ phi tuyến mà mô hình học được đã lỗi thời và cần retrain.

Nói cách khác, dải bất định không chỉ là một “vùng mờ” đẹp mắt trên biểu đồ, mà là **một công cụ quản trị rủi ro**: nó giúp ta đặt ra câu hỏi “*khi nào mô hình bắt đầu nói sai đủ nhiều để cần can thiệp?*”

2.5 Kết luận ngắn cho phần độ tin cậy

Tổng hợp lại:

- XGBoost residual cho sai số out-of-sample ổn định qua nhiều năm và nhiều pha thị trường.
- Pricing Layer được tối ưu hoá bằng Random Search + time-based CV, cho bộ tham số đặc trưng cho hành vi của FPT.
- Cơ chế regime-aware đảm bảo mô hình phản ứng khác nhau giữa BULL / BEAR / SIDEWAYS, thay vì áp một công thức cho mọi hoàn cảnh.
- Dải bất định UNCERT analytic, dựa trên phân phối residual đã được kiểm chứng, cung cấp một thước đo cụ thể về mức “tin được” của forecast.

Nhờ đó, dự báo 100 ngày của mô hình **không phải là một con số đơn lẻ**, mà là một *bức tranh xác suất* có thể dùng để: theo dõi độ lệch so với kịch bản trung tâm, ra quyết định phòng thủ hay tấn công, và biết được khi nào mô hình cần được cập nhật để bắt kịp thị trường.

PHẦN 5: Demo Sản Phẩm qua Streamlit

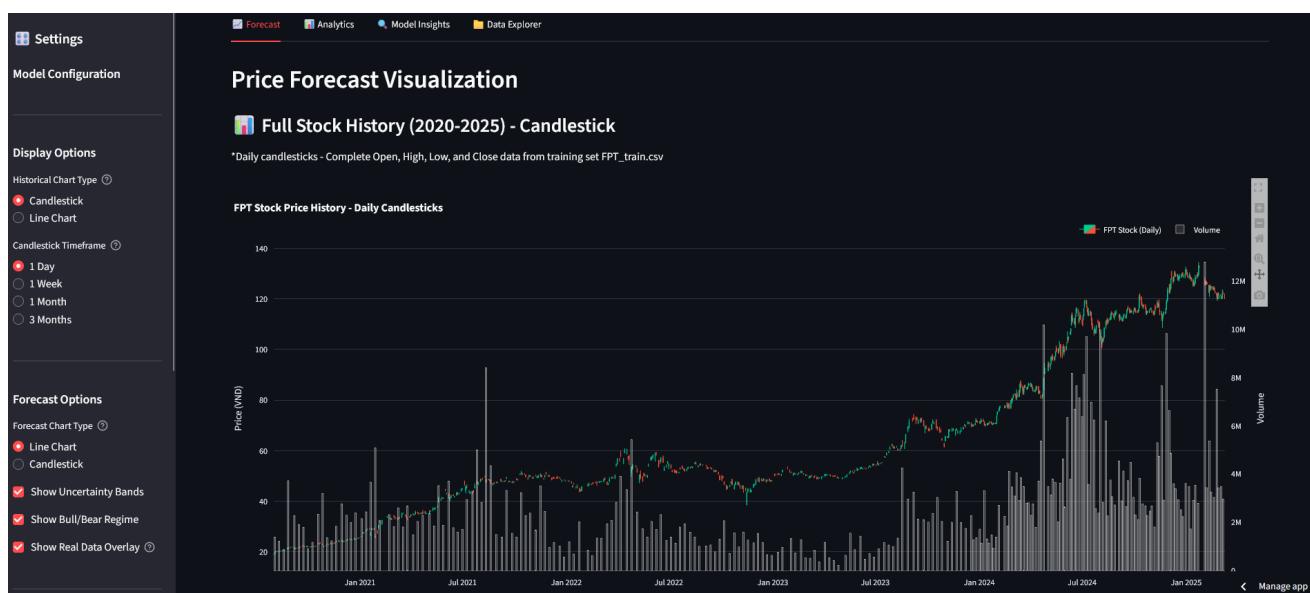
Hệ thống dự báo giá cổ phiếu FPT được tích hợp trên nền tảng web tương tác sử dụng framework Streamlit. Ứng dụng cung cấp cái nhìn toàn diện từ dữ liệu lịch sử, kết quả dự báo của mô hình Hybrid, đến khả năng tùy chỉnh tham số theo thời gian thực để phân tích kịch bản (What-if analysis). Dưới đây là các tính năng chính của hệ thống:

1 Landing Page & Data Visualization

Giao diện chính (Landing Page) được thiết kế để cung cấp cái nhìn tổng quan ngay lập tức về lịch sử giá cổ phiếu FPT. Hệ thống hỗ trợ trực quan hóa dữ liệu toàn diện từ năm 2020 đến 2025.

Người dùng có thể:

- Chuyển đổi linh hoạt giữa biểu đồ đường (Line Chart) và biểu đồ nến (Candlestick) để quan sát hành vi giá chi tiết.
- Tùy chỉnh khung thời gian hiển thị (Ngày, Tuần, Tháng) để phân tích xu hướng dài hạn hoặc biến động ngắn hạn.
- Quan sát đồng thời khối lượng giao dịch (Volume) để đánh giá động lực thị trường.



Hình 25: Giao diện Landing Page với biểu đồ lịch sử giá và khối lượng (2020-2025).

2 100-Day Price Forecast (Tính Năng Cốt Lõi)

Đây là tính năng trung tâm của ứng dụng, cung cấp tầm nhìn chiến lược cho 100 ngày giao dịch tiếp theo. Thay vì chỉ đưa ra một con số dự báo vô hồn, hệ thống hiển thị một biểu đồ đa lớp thể hiện sự kết hợp giữa xu hướng dài hạn (Trend), dao động ngắn hạn (Base Path) và mức độ rủi ro (Uncertainty Bands).



Hình 26: Biểu đồ dự báo 100 ngày với các thành phần: Central Forecast, Trend Line và Dải tin cậy 90%.

Giải mã các thành phần dự báo

Hệ thống hoạt động dựa trên cơ chế "Hội đồng Chuyên gia" (Council of Experts), với nhiều mô hình cùng tham gia đóng góp ý kiến để đưa ra kết quả cuối cùng.

- **Trend (Linear) - Đường xu hướng dài hạn (Nét đứt đỏ):**

Đây là đại diện cho **Static Expert** (Chuyên gia Tĩnh). Đường này được xây dựng dựa trên **Math Backbone** — một khung toán học hồi quy tuyến tính trên dữ liệu giá logarit. Nó đóng vai trò như một "mỏ neo" giá trị, bù qua mọi nhiễu động ngắn hạn (tin tức hàng ngày, tâm lý đám đông) để nhắc nhở mô hình về quỹ đạo tăng trưởng bền vững của doanh nghiệp trong suốt 5-10 năm qua.

- **Base Path (Hybrid) - Đường cơ sở (Nét đứt xanh lá):**

Đây là kết quả thô của **Dynamic Expert** (Chuyên gia Động). Nó là sản phẩm của mô hình **Hybrid Learning**, kết hợp giữa Math Backbone và thuật toán **XGBoost Residual**.

- *XGBoost Residual*: Là kỹ thuật máy học giúp mô hình "học" các sai số (residual) của xu hướng tuyến tính, từ đó nắm bắt được các mẫu hình phi tuyến phức tạp như các cú giật giá hay biến động theo mùa vụ mà đường Trend không thấy được.

- **Forecast (Central) - Dự báo trung tâm (Nét liền cam):**

Đây là **đường dự báo chính thức (CENTRAL_DET)**, là kết quả cuối cùng sau khi đi qua lớp kiểm soát **Pricing Layer**. Tại đây, các dự báo thô được tinh chỉnh bởi các cơ chế vật lý thị trường:

- *Clipping (Cắt ngọn)*: Giới hạn biên độ tăng/giảm quá đà trong một phiên.
- *Damping (Lực ma sát)*: Giảm dần tác động của các cú sốc (ie. tăng/giảm đột ngột) ngắn hạn theo thời gian.

- *Mean Reversion (Hồi quy về trung bình)*: Lực kéo giá quay trở lại vùng giá trị thực nếu dự báo đi quá xa.

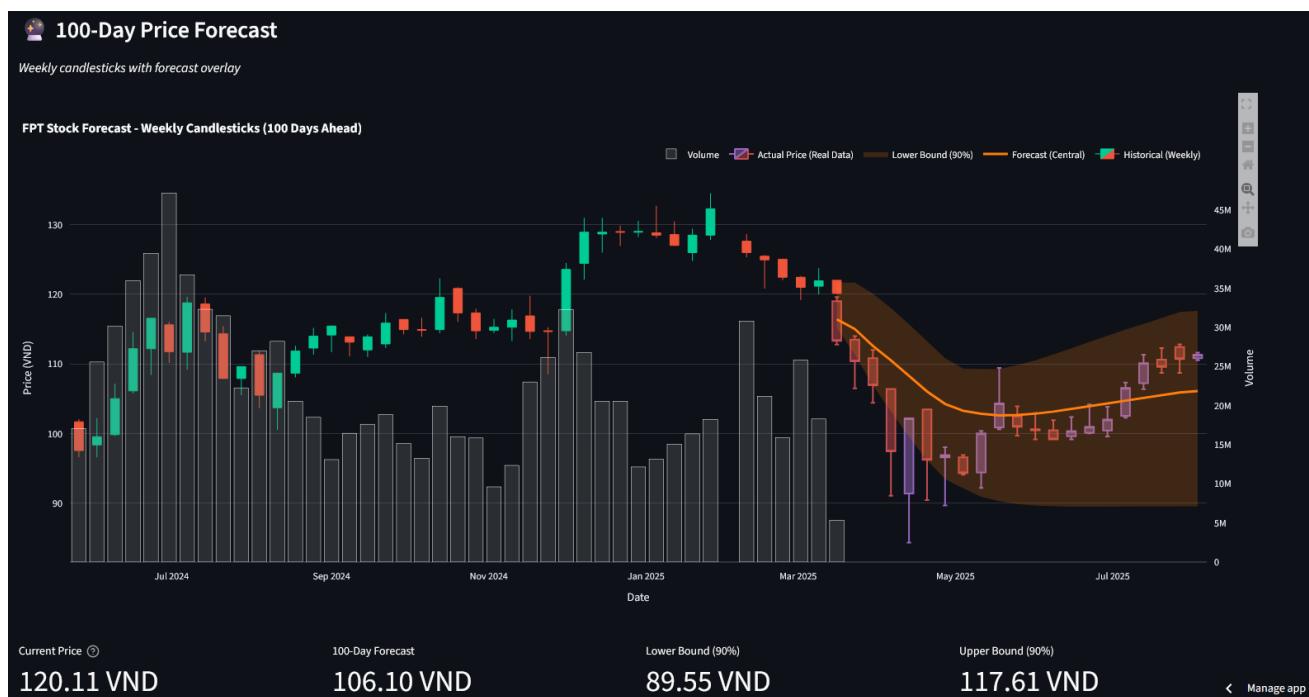
Đường này đại diện cho kịch bản khả dĩ nhất, cân bằng giữa tốc độ thích ứng của mô hình và sự ổn định của xu hướng dài hạn.

- **Lower/Upper Bound (90%) - Dải bất định (Vùng bóng mờ):**

Đại diện cho **Risk Expert (Chuyên gia Rủi ro)**. Dải này được xây dựng dựa trên mô phỏng **Monte Carlo** (chạy thử hàng 500 kịch bản ngẫu nhiên dựa trên sai số lịch sử). Nó cho biết vùng dao động an toàn của giá với độ tin cậy 90%. Nếu giá thực tế đi ra khỏi vùng này, đó là tín hiệu cảnh báo cấu trúc thị trường đang thay đổi mạnh (khủng hoảng hoặc bùng nổ) mà dữ liệu quá khứ chưa từng ghi nhận.

Candlestick Forecast View

Để dễ trực quan và phân tích hơn các lượng tăng và giảm, hệ thống cung cấp chế độ xem dự báo dưới dạng nền tuần (Weekly Candlesticks).



Hình 27: Chế độ xem dự báo dưới dạng nền tuần (Weekly Candlesticks).

Feature phụ: Kiểm chứng độ chính xác với dữ liệu thực tế

Tính năng giúp kiểm chứng độ chính xác của mô hình bằng cách chồng lớp dữ liệu thực tế (Real Data) lên đường dự báo (đối với các giai đoạn đã có dữ liệu thực tế). Hệ thống tự động tính toán và hiển thị các chỉ số đánh giá hiệu năng quan trọng:

- **MSE (Mean Squared Error):** 33.55
- **RMSE (Root Mean Squared Error):** 5.79 VND
- **MAE (Mean Absolute Error):** 4.57 VND

- **MAPE (Mean Absolute Percentage Error): 4.54%**

Cũng như là Bull và Bear trực quan hóa chế độ thị trường của dự đoán 100 ngày.



Hình 28: So sánh kết quả dự báo với dữ liệu thực tế và các chỉ số đánh giá độ chính xác (RMSE, MAE, MAPE).

3 Custom 100-Day Price Forecast with Adjustable Parameters

Thay vì chỉ demo 1 mô hình cũng nhǎc, nhóm mình muốn người dùng cũng được trải nghiệm tinh chỉnh mô hình qua các tham số đᾶ được feature engineer trước, đẽ có thể hiểu hơv về ảnh hưởng của từng tham số tới kết quả và xu hướng dự đoán của mô hình.



Hình 29: Giao diện tùy chỉnh tham số dự báo và so sánh trực tiếp với dự báo gốc.

Giải thích các tham số điều chỉnh (Sidebar Parameters)

Các tham số trên thanh điều khiển (Sidebar) được chia thành 2 nhóm chính, tác động trực tiếp đến hình dáng và xu hướng của đường dự báo:

1. Nhóm Return & Volatility (Kiểm soát biến động và quán tính)

- **Return Clip Quantile (Ngưỡng cắt biến độ lợi nhuận):**

Tham số này kiểm soát mức độ "cực đoan" của các dao động giá hàng ngày.

- *Tác dụng:* Nó xác định ngưỡng phần trăm (quantile) để loại bỏ các biến động nhiễu (outliers) từ dữ liệu lịch sử. Ví dụ, mức 0.97 nghĩa là mô hình sẽ bỏ qua top 3% những phiên biến động mạnh nhất trong quá khứ khi dự báo.
- *Ảnh hưởng:* Giảm tham số này làm đường dự báo mượt hơn (an toàn hơn); tăng tham số này cho phép mô hình dự báo các cú giật giá mạnh hơn (nhạy hơn nhưng rủi ro cao hơn).

- **Half-Life (days) (Chu kỳ bán rã của động lượng):**

Đây là tham số của cơ chế *Damping* (lực ma sát). Nó định nghĩa sau bao nhiêu ngày thì tác động của một cú sốc tin tức hoặc động lượng hiện tại sẽ giảm đi một nửa.

- *Tác dụng:* Kiểm soát độ bền của xu hướng ngắn hạn.
- *Ảnh hưởng:* Giá trị nhỏ (ví dụ: 20 ngày) khiến dự báo nhanh chóng quay về trạng thái cân bằng. Giá trị lớn (ví dụ: 60 ngày) cho phép quán tính tăng/giảm hiện tại kéo dài hơn trong tương lai.

- **Mean Revert Alpha (Hệ số lực hồi quy):**

Đại diện cho độ cứng của "lò xo" kéo giá quay về giá trị thực (Fair Value).

- *Tác dụng:* Xác định tốc độ giá điều chỉnh về đường trung bình dài hạn khi nó đi quá xa.
- *Ảnh hưởng:* Alpha cao (ví dụ: 0.1) sẽ kéo giá về Trend rất nhanh (gây gắt). Alpha thấp (ví dụ: 0.02) cho phép giá "lang thang" xa khỏi Trend lâu hơn.

- **Mean Revert Start Day:** Quy định thời điểm bắt đầu kích hoạt cơ chế hồi quy.
 - *Tác dụng:* Cho phép trì hoãn lực kéo về trung bình. Ví dụ: đặt là 40 nghĩa là trong 40 ngày đầu tiên, hãy để giá chạy tự do theo quán tính (Momentum), chỉ bắt đầu kéo về Trend từ ngày 41.

2. Nhóm Fair Value & Trend (Định nghĩa vùng giá trị và Xu hướng)

- **Fair Up/Down Multiplier (Hệ số biên độ giá trị thực):**

Hai tham số này xác định độ rộng của "dải giá trị hợp lý" quanh đường Trend.

- *Tác dụng:* Tạo ra một vùng dệm (buffer zone). Chừng nào giá dự báo còn nằm trong vùng này, lực hồi quy (Mean Reversion) sẽ không bị kích hoạt mạnh.
- *Ảnh hưởng:* Nới rộng biên độ (ví dụ: Up 1.4, Down 0.75) phản ánh quan điểm thị trường đang chấp nhận định giá cao/thấp hơn lịch sử (phù hợp với giai đoạn Bull/Bear mạnh).

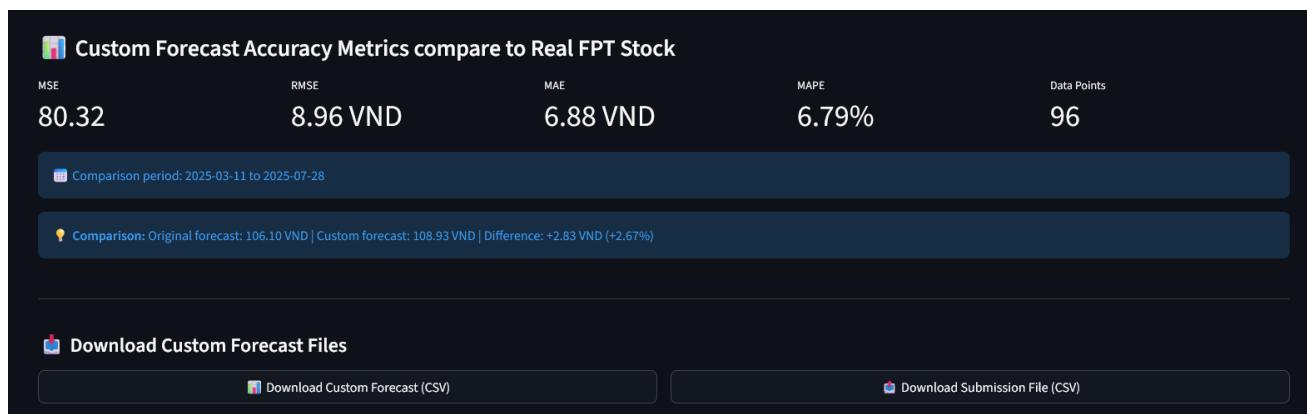
- **Trend Lookback & Threshold:**

Cấu hình độ nhạy của bộ phát hiện xu hướng (Trend Detector).

- *Tác dụng:* Giúp mô hình phân biệt giữa một pha "Sideways" và một pha "Uptrend/Downtrend" thực sự. Nếu phát hiện xu hướng mạnh vượt qua ngưỡng *Threshold* trong khoảng thời gian *Lookback*, mô hình sẽ tạm thời vô hiệu hóa lực kéo Mean Reversion để không cản trở đà tăng/giảm của giá.

Download Custom Results

Sau khi tinh chỉnh và đạt được kịch bản dự báo mong muốn, người dùng có thể tải xuống kết quả để sử dụng cho các mục đích phân tích khác. Hệ thống hỗ trợ xuất dữ liệu dự báo (CSV) và file cấu hình tham số.



Hình 30: Tính năng tải xuống dữ liệu dự báo tùy chỉnh.

Kết quả "Custom Forecast" sẽ được hiển thị so sánh trực tiếp với "Original Forecast" và dữ liệu thực tế, giúp người dùng đánh giá tác động của từng tham số.

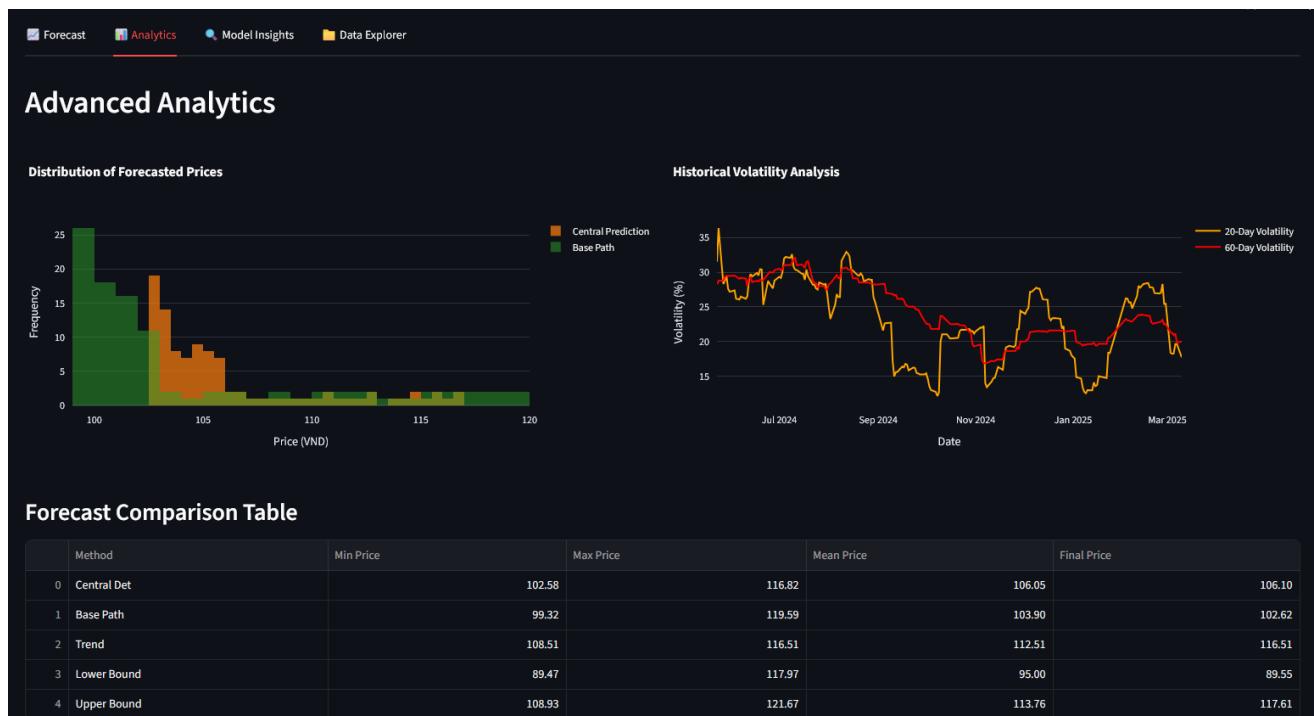
4 UI Bổ trợ

Ngoài các tính năng dự báo chính, hệ thống cung cấp bộ công cụ phân tích bổ trợ giúp người dùng hiểu sâu hơn về đặc tính dữ liệu và hành vi mô hình.

Performance Metrics & Analytics

Tab "Analytics" cung cấp các biểu đồ thống kê nâng cao:

- **Distribution Analysis:** Biểu đồ phân phối tần suất của các mức giá dự báo.
- **Volatility Analysis:** Phân tích biến động lịch sử (20-day và 60-day volatility) để đánh giá rủi ro thị trường hiện tại.
- **Comparison Table:** Bảng so sánh chi tiết các mức giá Min, Max, Mean giữa các phương pháp dự báo (Central, Base Path, Trend).



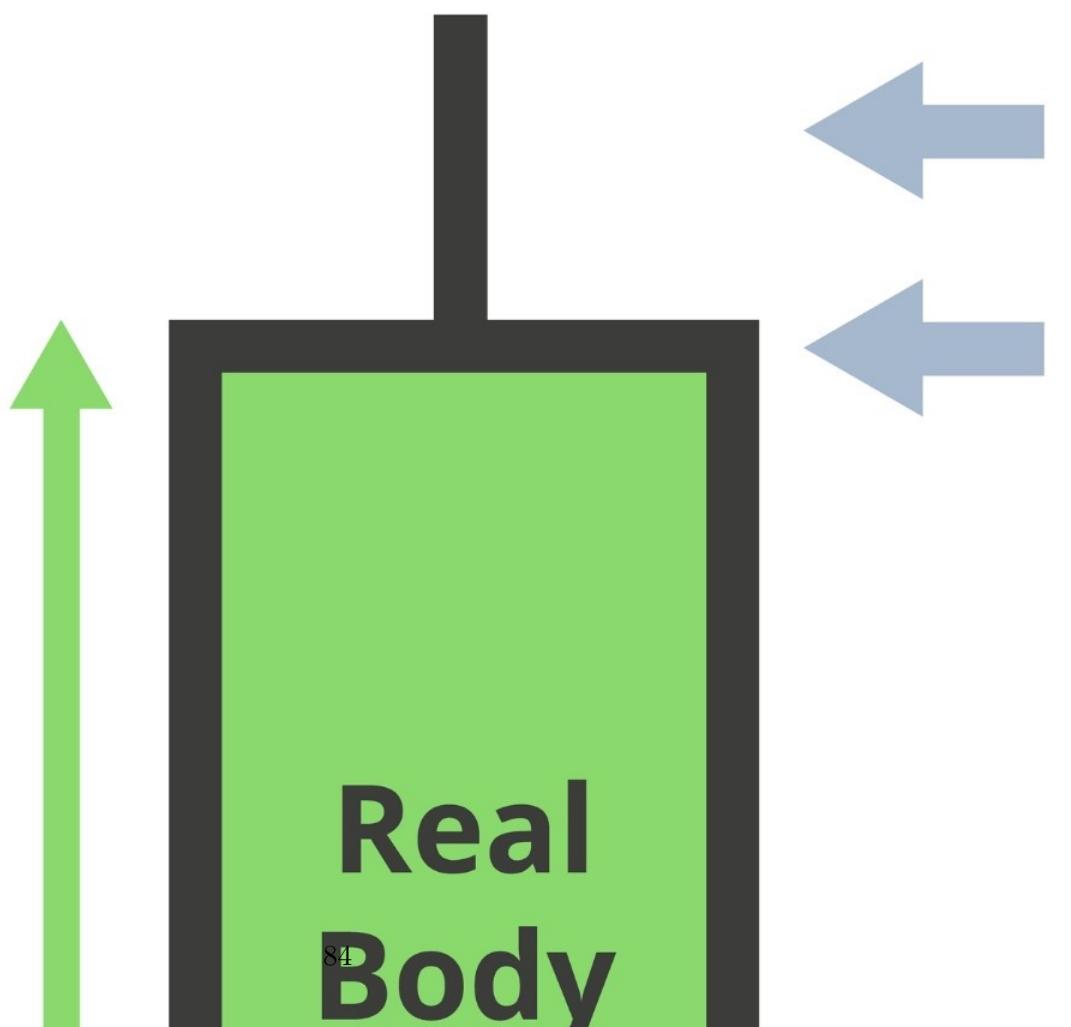
Hình 31: Dashboard phân tích nâng cao với phân phối giá và biểu đồ biến động.

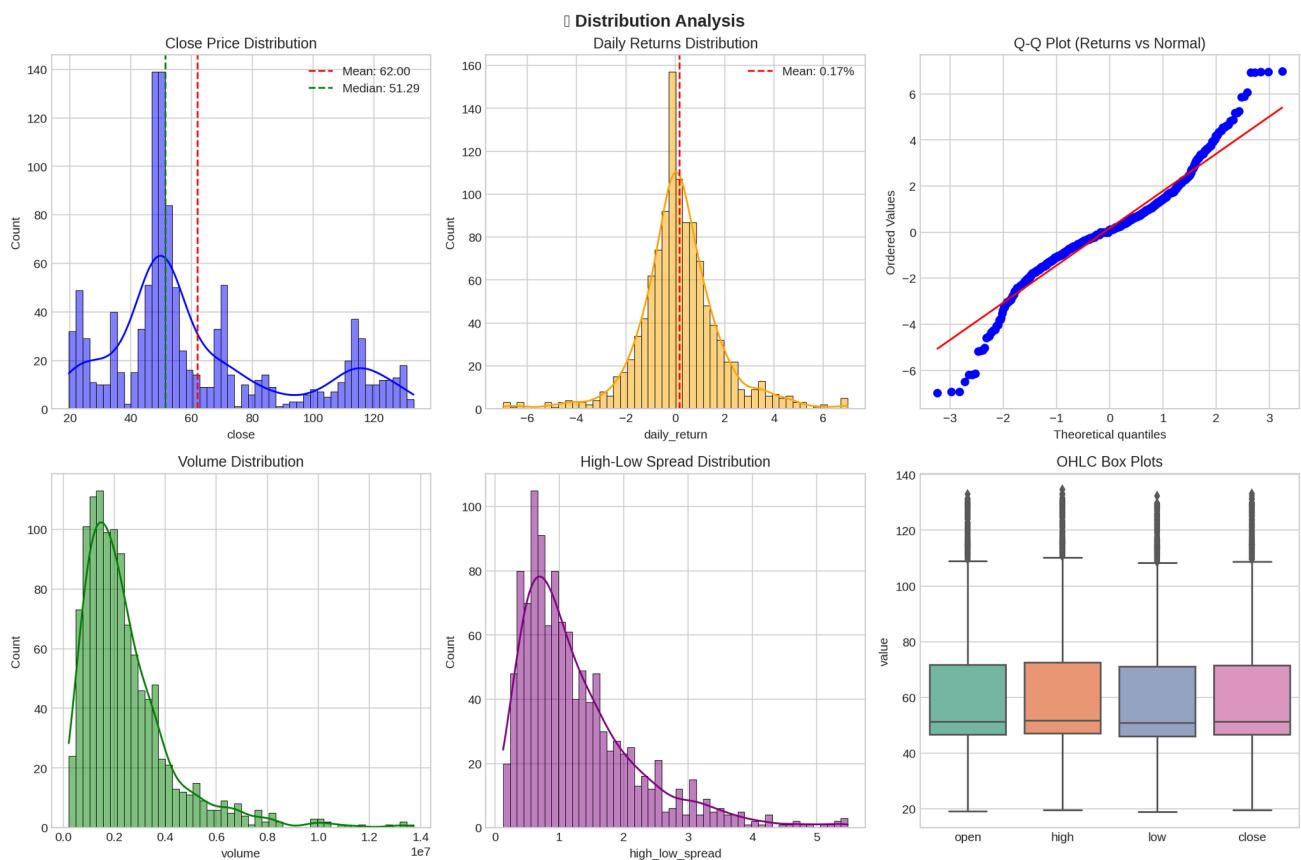
Model Insight & Data Explorer

- **Model Insight:** Cung cấp tài liệu kỹ thuật về kiến trúc Hybrid (Math Backbone + XGBoost Residual), giải thích cơ chế Feature Engineering và các chỉ số hiệu năng huấn luyện.
- **Data Explorer:** Cho phép tra cứu, lọc và xem chi tiết dữ liệu thô đầu vào cũng như dữ liệu kết quả đầu ra dưới dạng bảng.

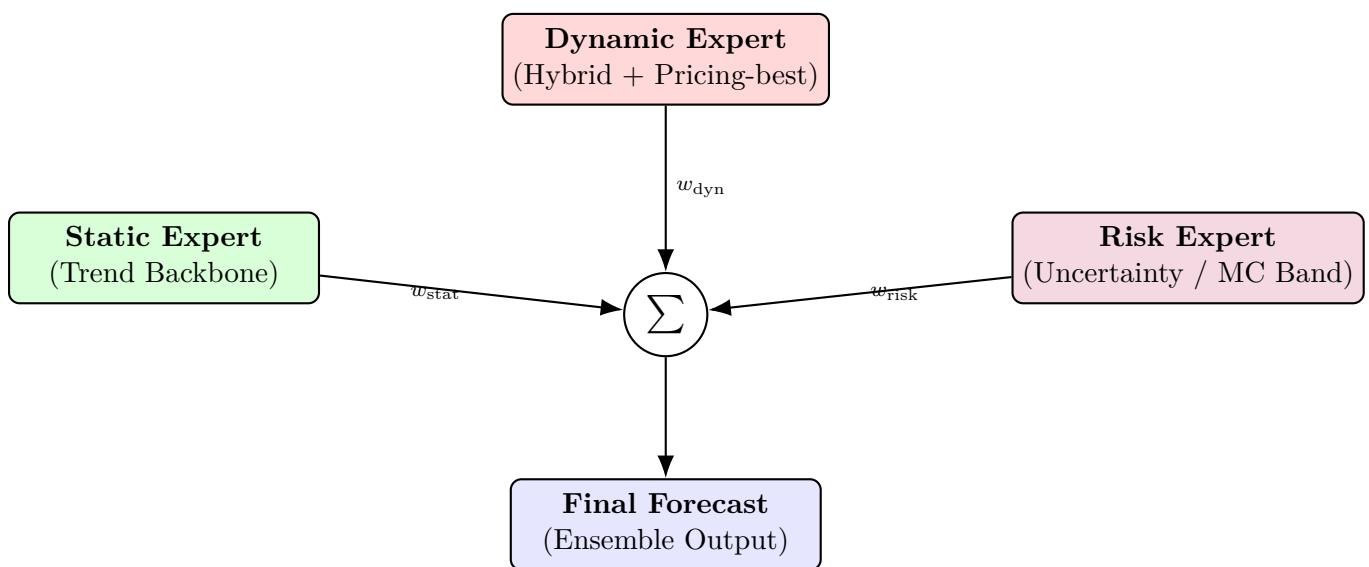
CAN

High Price

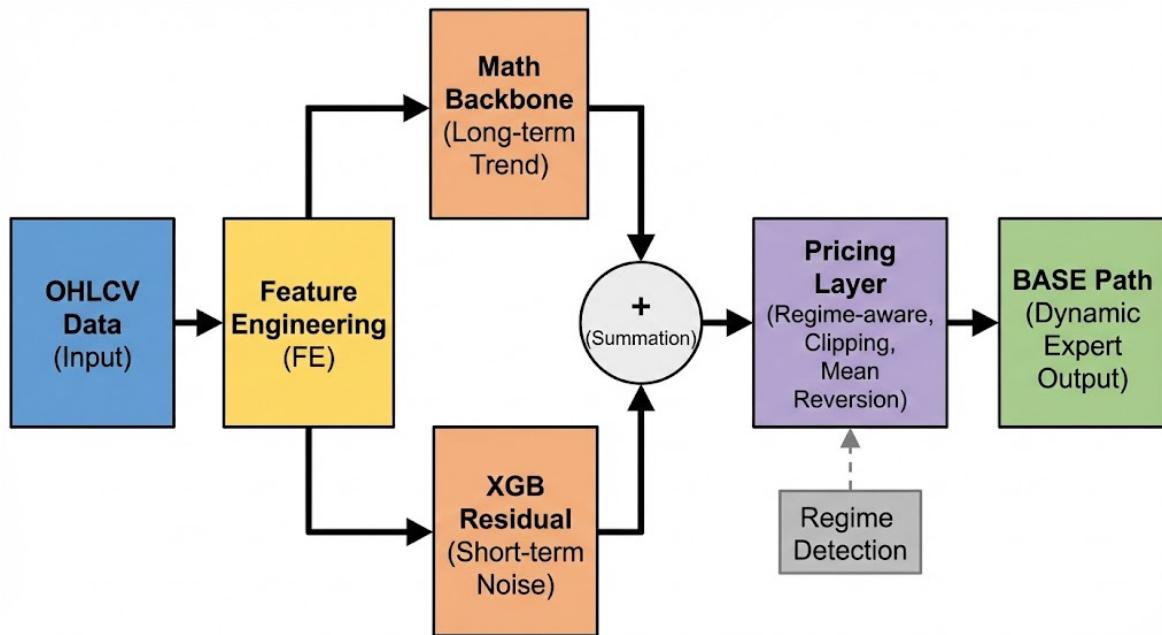




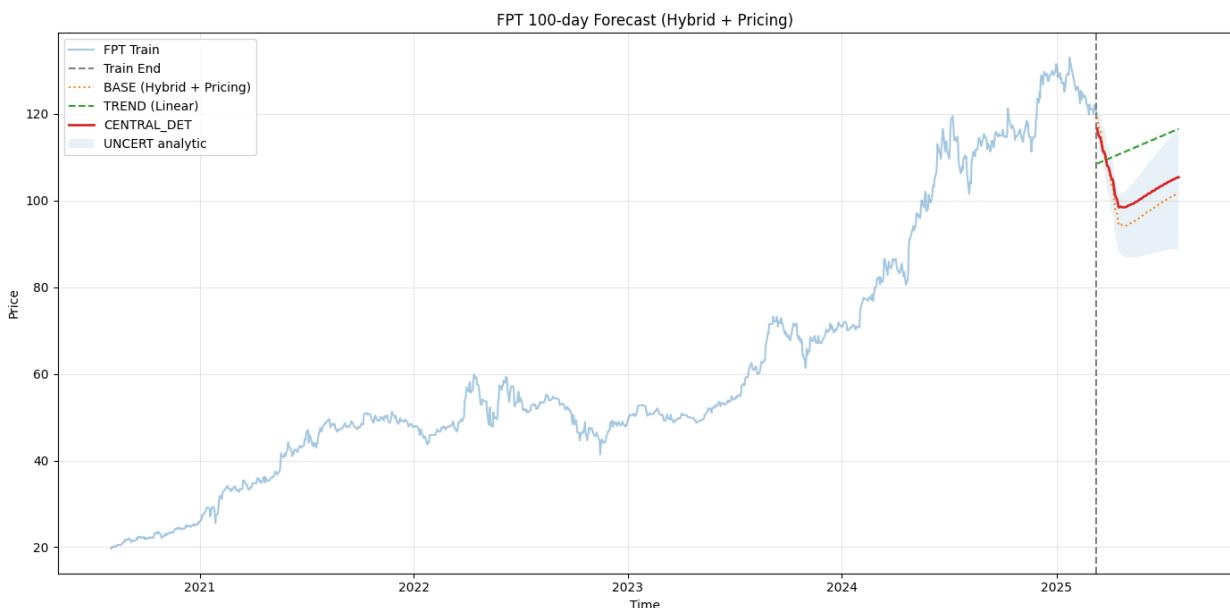
Hình 10: Q-Q Plot



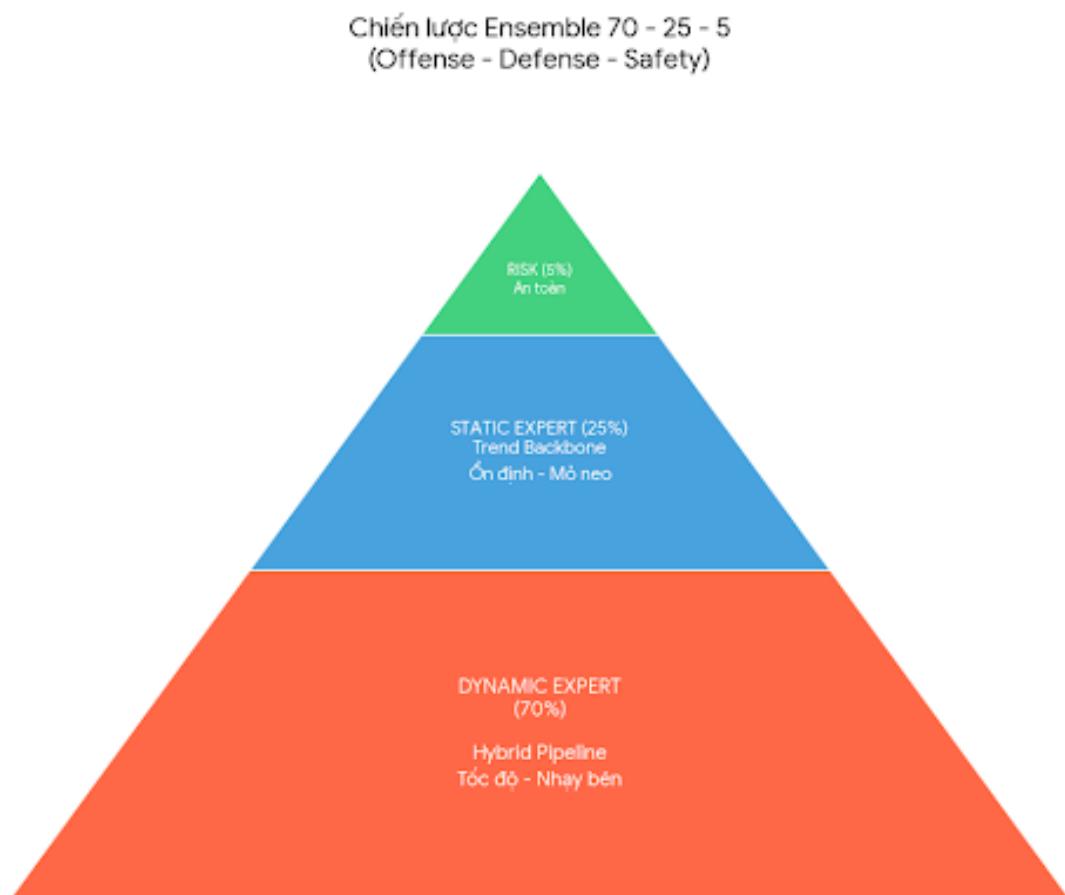
Hình 20: Hội đồng chuyên gia: ba góc nhìn bổ sung, một quyết định chung



Hình 21: Sơ đồ dòng chảy thông tin của Dynamic Expert: từ OHLCV → FE → Math Backbone + XGB Residual → Pricing Layer → BASE Path.



Hình 22: Minh họa BASE Path (Hybrid + Pricing-best) so với Trend tuyến tính và dải bất định: mô hình vừa tôn trọng cú sút mạnh sau ngày 10/03/2025, vừa không suy đoán phán cực đoan.



Hình 23: Tam giác Chiến lược 70–25–5: Dynamic (tốc độ), Static (ổn định), Risk (an toàn).