Tài liệu Kỹ thuật Tổng hợp: Forecasting Module

1. Mục tiêu và Vai trò

Forecasting Module là một thành phần AI chuyên biệt, có trách nhiệm trả lời các câu hỏi về **diễn biến tương lai** của một tài sản. Khác với Technical Analysis Module (phân tích quá khứ và hiện tại), module này cung cấp các dự báo định lượng để hỗ trợ Decision Maker Agent.

Module này được thiết kế để tạo ra hai loại dự báo riêng biệt, phục vụ cho hai mục đích khác nhau:

1. **Dự báo Hồi quy Phân phối (Distribution Regression):** Trả lời câu hỏi "Phạm vi giá có khả năng xảy ra là gì và mức độ rủi ro biến động ra sao?".
2. **Dự báo Phân loại Giao dịch (Trade Classification):** Trả lời câu hỏi "Nếu mở một vị thế giao dịch bây giờ, xác suất thành công là bao nhiêu?".

2. Kiến trúc Mô hình: 39 Mô hình Chuyên biệt

Để đạt được độ chính xác và tính thích ứng cao, hệ thống không sử dụng một mô hình chung duy nhất. Thay vào đó, chúng ta sẽ huấn luyện và triển khai một bộ các mô hình chuyên biệt:

- **13 Nhóm Đối tượng:**
  - 1 nhóm "University" (chung cho tất cả, làm baseline hoặc fallback).
  - 12 nhóm chuyên biệt cho 12 lĩnh vực kinh tế chính (Công nghệ, Tài chính, Y tế, v.v.).
- **3 Loại Mô hình cho mỗi Nhóm:**

  1. **Distribution Regression (5-day horizon):** Một mô hình hồi quy đa đầu ra, dự báo phân phối giá trong 5 ngày tới.
  2. **Distribution Regression (20-day horizon):** Một mô hình hồi quy đa đầu ra, dự báo phân phối giá trong 20 ngày tới.
  3. **Triple-Barrier Classification (5-10 day horizon):** Một mô hình phân loại ba lớp, dự báo kết quả của một giao dịch.

=> Tổng cộng: 13 nhóm * 3 loại mô hình = 39 mô hình đã được huấn luyện sẽ được quản lý và triển khai.

3. Thuật toán và Tính giải thích

- **Lựa chọn Chính:** Ưu tiên các mô hình dựa trên cây quyết định mạnh mẽ như **LightGBM (LGBMRegressor, LGBMClassifier)**, XGBoost, hoặc RandomForest.
- **Lý do:** Các mô hình này cho hiệu suất rất cao trên dữ liệu dạng bảng, huấn luyện nhanh, và quan trọng nhất là có **tính giải thích (Explainability)** cao.

- **Cung cấp "Bằng chứng" (Evidence):** Mọi dự đoán sẽ được đi kèm với bằng chứng được tạo ra bởi **shap.TreeExplainer**. Nó sẽ chỉ ra các đặc trưng (features) có ảnh hưởng lớn nhất (cả tích cực và tiêu cực) đến dự đoán đó, làm cho mô hình trở nên minh bạch.

4. Quy trình Chuẩn bị Dữ liệu và Huấn luyện (MLOps Pipeline)

Đây là một quy trình hai giai đoạn phức tạp, được thiết kế để chạy offline (ví dụ: trên Kaggle/Colab).

Giai đoạn 1: Tối ưu hóa Bài toán (Problem Definition Optimization)

- **Mục tiêu:** Giai đoạn này không nhằm mục đích tìm ra mô hình tốt nhất, mà là để tìm ra **cách định nghĩa "thành công" và "thất bại" một cách hợp lý nhất** cho mỗi nhóm ngành. Chúng ta cần trả lời câu hỏi: "Với đặc tính biến động của nhóm ngành này, một mục tiêu giao dịch thực tế là gì?". Giai đoạn này chỉ áp dụng cho mô hình **phân loại Triple-Barrier**.
- **Các Tham số Bài toán cần Tối ưu:**
    1. **horizon (Chân trời Thời gian):**
        - **Ý nghĩa:** Số ngày tối đa mà một giao dịch được phép diễn ra trước khi bị coi là "hết giờ".
        - **Tại sao cần tối ưu:** Một horizon quá ngắn (ví dụ: 3 ngày) có thể không đủ thời gian để giá chạm đến các mục tiêu. Một horizon quá dài (ví dụ: 15 ngày) sẽ làm tín hiệu ban đầu bị nhiễu bởi các sự kiện mới. Chúng ta cần tìm một khoảng thời gian "vàng" (thường từ 5-10 ngày cho giao dịch trung hạn).
    2. **tp_pct (Take Profit Percentage - Tỷ lệ Chốt lời):**
        - **Ý nghĩa:** Ngưỡng lợi nhuận mục tiêu. Ví dụ, 0.05 tương ứng với rào cản trên ở mức giá hiện tại * 1.05.
        - **Tại sao cần tối ưu:** Các ngành có biến động thấp (như Tiện ích) có thể chỉ đạt được mức lợi nhuận 2-3% một cách thực tế, trong khi các ngành biến động cao (như Công nghệ) có thể nhắm đến 7-8%. Việc đặt một mục tiêu không thực tế sẽ tạo ra một bộ dữ liệu có rất ít nhãn "thành công".
    3. **sl_pct (Stop Loss Percentage - Tỷ lệ Cắt lỗ):**
        - **Ý nghĩa:** Ngưỡng rủi ro chấp nhận được. Ví dụ, 0.03 tương ứng với rào cản dưới ở mức giá hiện tại * 0.97.
        - **Tại sao cần tối ưu:** Một mức cắt lỗ quá chặt trên một cổ phiếu biến động cao sẽ khiến giao dịch bị đóng quá sớm. Một mức cắt lỗ quá rộng sẽ làm tăng rủi ro không cần thiết. Ngưỡng này phải phản ánh đúng "tính cách" biến động của ngành.
- **Hành động và Quy trình Gán nhãn (Labeling Process):**
    1. **Chạy Vòng lặp Grid Search:** Lặp qua các bộ ba tham số (horizon, tp_pct, sl_pct) đã được định nghĩa trước.
    2. **Đối với mỗi điểm dữ liệu i trong bộ huấn luyện:**
    a. **Xác định các Rào cản (Barriers):** Dựa trên giá đóng cửa tại ngày i (close_i), tính toán:
    * Upper Barrier (Take Profit) = close_i * (1 + tp_pct)

* Lower Barrier (Stop Loss) = close_i * (1 - sl_pct)
* Vertical Barrier (Time Out) = ngày i + horizon
b. **Quét Tương lai:** Nhìn vào chuỗi giá từ ngày i+1 đến ngày i+horizon.
c. **Xác định sự kiện Chạm Rào cản đầu tiên:**
* Tìm ngày đầu tiên mà giá high chạm hoặc vượt Upper Barrier.
* Tìm ngày đầu tiên mà giá low chạm hoặc xuống dưới Lower Barrier.
d. **Gán Nhãn:**
* Nếu **Upper Barrier** được chạm trước (hoặc chỉ có Upper Barrier được chạm), gán nhãn **1** (Thành công).
* Nếu **Lower Barrier** được chạm trước (hoặc chỉ có Lower Barrier được chạm), gán nhãn **-1** (Thất bại).
* Nếu sau horizon ngày mà không có rào cản nào được chạm, gán nhãn **0** (Hết giờ).

3. **Đánh giá Bộ dữ liệu:** Sau khi đã gán nhãn cho toàn bộ dữ liệu với bộ tham số hiện tại, tính toán tỷ lệ phân phối của các lớp 1, 0, và -1.
4. **Lựa chọn:** Chọn bộ tham số (horizon, tp_pct, sl_pct) nào tạo ra một bộ dữ liệu có tỷ lệ các lớp cân bằng và hợp lý nhất (ví dụ: lớp 1 và -1 đều chiếm ít nhất 5-10%).

- **Kết quả:** Một bộ tham số (horizon, tp_pct, sl_pct) tối ưu cho mỗi nhóm ngành, sẵn sàng cho Giai đoạn 2.

Giai đoạn 2: Huấn luyện và Tối ưu hóa Siêu tham số của Mô hình Lõi
(Sau khi đã có bộ dữ liệu được gán nhãn tối ưu từ Giai đoạn 1)

- **Mục tiêu:** Với bài toán đã được định nghĩa rõ ràng, tìm ra cấu hình mô hình (siêu tham số) sẽ cho hiệu suất dự đoán tốt nhất.
- **Hành động:**
  1. **Walk-Forward Validation:** Chia dữ liệu theo các mốc thời gian để mô phỏng điều kiện giao dịch thực tế.
  2. **Tối ưu hóa Siêu tham số:** Sử dụng RandomizedSearchCV hoặc Optuna để tìm kiếm không gian các siêu tham số của LightGBM (n_estimators, learning_rate...).
  3. **Huấn luyện Mô hình Cuối cùng:** Huấn luyện lại mô hình với bộ siêu tham số tốt nhất trên một tập dữ liệu lớn hơn.
  4. **Quản lý Phiên bản:** Lưu lại cả mô hình sản xuất cuối cùng và các mô hình tại các mốc quá khứ để phục vụ backtest.

5. Giao tiếp và Tích hợp

- **Phụ thuộc:** Forecasting Module **không** gọi trực tiếp Technical Module.
- **Luồng hoạt động:**
  1. Nó định nghĩa một **"yêu cầu đặc trưng"** (ví dụ: qua file config).
  2. AIServiceQuickOrchestrator sẽ đọc yêu cầu này, gọi TechnicalOrchestrator để lấy enriched_df.
  3. AIServiceQuickOrchestrator sau đó truyền enriched_df vào ForecastingEngine (lớp Facade của module này).

4. ForecastingEngine sẽ tải các mô hình .pkl phù hợp cho ticker được yêu cầu và trả về một báo cáo dự báo có cấu trúc, bao gồm cả **dự đoán** và **bằng chứng (evidence)**.