TRƯỜNG ĐẠI HỌC CÔNG NGHỆ - ĐHQGHN
VIỆN TRÍ TUỆ NHÂN TẠO

Khai phá và phân tích dữ liệu
Báo cáo bài tập số 2

Phương pháp xác định vùng mua bán của tài sản số với tỷ suất lợi nhuận cao

Thành viên nhóm: Phạm Nhật Quang 23020413
Bùi Minh Quân 23020415
Phan Quang Trường 23020443
Hồ Lê Dương 22022641

Giáo viên hướng dẫn: GS. Nguyễn Phương Thái
ThS. Ngô Minh Hương

Mục lục

Chương 1

Giới thiệu

1.1 Bối cảnh và vấn đề nghiên cứu

Thị trường tiền mã hóa (Cryptocurrency), với đại diện tiêu biểu là Bitcoin, đã và đang trở thành một trong những kênh đầu tư tài chính sôi động và thu hút dòng tiền lớn nhất toàn cầu trong thập kỷ qua. Khác với thị trường chứng khoán truyền thống, thị trường tiền mã hóa hoạt động 24/7 với biên độ dao động giá cực lớn, tạo ra cơ hội sinh lời hấp dẫn nhưng đồng thời cũng tiềm ẩn rủi ro thanh lý tài sản rất cao. Trong bối cảnh đó, sự biến động của giá không chỉ được định hình bởi quy luật cung cầu đơn thuần mà còn chịu tác động mạnh mẽ từ tâm lý đám đông, các sự kiện kinh tế vĩ mô và luồng tin tức truyền thông liên tục. Điều này đặt ra một bài toán thách thức cho các nhà đầu tư: làm thế nào để tìm ra được "trật tự" trong sự "hỗn loạn" của dữ liệu giá và tin tức.

Vấn đề nghiên cứu của đề tài không nằm ở việc cố gắng dự đoán chính xác giá trị tương lai của Bitcoin — một nhiệm vụ được xem là bất khả thi trong lý thuyết bước đi ngẫu nhiên (Random Walk Theory). Thay vào đó, trọng tâm của nghiên cứu là tìm kiếm các mẫu hình (patterns) lặp lại trong lịch sử và lượng hóa chúng thành các luật giao dịch dựa trên xác suất thống kê. Bằng cách kết hợp dữ liệu lịch sử giá với dữ liệu phi cấu trúc từ tin tức, chúng em mong muốn xây dựng một hệ thống khai phá dữ liệu có khả năng nhận diện các điều kiện thị trường đặc thù. Khi các điều kiện này hội tụ, xác suất giá di chuyển theo một hướng nhất định sẽ cao hơn ngẫu nhiên, từ đó cung cấp cơ sở định lượng để ra quyết định mua hoặc bán một cách khoa học.

1.2 Lý do chọn đề tài

Lý do cốt lõi thúc đẩy nhóm lựa chọn đề tài này xuất phát từ chính niềm đam mê và trải nghiệm thực tế của các thành viên đối với thị trường tài chính. Với nền tảng kiến thức sẵn có về đầu tư chứng khoán và tiền mã hóa, nhóm nghiên cứu nhận thấy rằng phần lớn các quyết định giao dịch của nhà đầu tư cá nhân thường bị chi phối bởi cảm xúc hoặc những nhận định chủ quan thiếu cơ sở kiểm chứng. Trong khi đó, bản chất của giao dịch tài chính thành công là cuộc chơi của xác suất và quản trị rủi ro. Do đó, việc áp dụng các kỹ thuật Khai phá dữ liệu (Data Mining), cụ thể là các phương pháp "truyền thống" nhưng mạnh mẽ về mặt giải thích như Phân cụm (Clustering) và Khai phá luật kết hợp (Association Rules), là bước đi cần thiết để chuyển hóa các kinh nghiệm đầu tư cảm tính thành các chiến lược có tính kỷ luật và có thể đo lường được.

Hơn nữa, việc lựa chọn phương pháp tiếp cận dựa trên luật kết hợp thay vì các mô hình học sâu (Deep Learning) phức tạp là một quyết định có chủ đích. Trong tài chính, khả năng giải thích (interpretability) của mô hình quan trọng không kém độ chính xác. Chúng em muốn hiểu rõ **tại sao** một tín hiệu mua được đưa ra — liệu đó là do chỉ báo kỹ thuật chạm vùng quá bán hay do tác động từ một chuỗi tin tức tích cực — thay vì phó mặc tài sản cho một "hộp đen" thuật toán. Đề tài này, vì thế, không chỉ là một bài tập học thuật nhằm áp dụng các kỹ thuật xử lý dữ liệu lớn, mà còn là một nỗ lực nghiêm túc nhằm xây dựng một công cụ hỗ trợ ra quyết định thực tế, phục vụ trực tiếp cho hoạt động đầu tư của chính các thành viên trong nhóm trong tương lai.

Chương 2

Chuẩn bị dữ liệu

2.1 Dữ liệu giá Bitcoin (Numerical Data)

2.1.1 Nguồn dữ liệu và thu thập

Để đảm bảo tính chính xác và đồng nhất của dữ liệu thị trường, chúng em quyết định sử dụng nguồn dữ liệu từ Binance - sàn giao dịch tiền mã hóa có thanh khoản lớn nhất thế giới. Dữ liệu được thu thập thông qua thư viện Python `binance_historical_data`.

Chúng em tập trung thu thập dữ liệu nến (candlestick) ở thị trường giao ngay (Spot Market) với khung thời gian 1 phút (`1m timeframe`). Việc chọn khung thời gian nhỏ giúp mô hình nắm bắt được các biến động vi mô (micro-movements) mà dữ liệu theo ngày hoặc giờ có thể bỏ qua.

Dưới đây là đoạn mã thực hiện quá trình tải dữ liệu từ ngày 01/01/2023 đến hiện tại:

```
from binance_historical_data import BinanceDataDumper
import datetime

# Cấu hình bộ tải dữ liệu
data_dumper = BinanceDataDumper(
    path_dir_where_to_dump="Data",  # Thư mục lưu trữ
    asset_class="spot",             # Thị trường giao ngay
    data_type="klines",             # Dữ liệu nến (OHLCV)
    data_frequency='1m'             # Tần suất 1 phút
)

# Tải dữ liệu BTC/USDT từ 01/01/2023
data_dumper.dump_data(
    tickers=["BTCUSDT"],
    date_start=datetime.date(2023, 1, 1),
    date_end=None,                  # None: Tải đến hiện tại
    is_to_update_existing=False
)
```

2.1.2 Làm sạch và Tiền xử lý

Dữ liệu tải về được chia thành các file nhỏ theo tháng và ngày. Quy trình tiền xử lý bao gồm các bước: Tổng hợp dữ liệu (Aggregation), Chuẩn hóa thời gian (Timestamp Normalization) và Phân chia tập dữ liệu (Data Splitting).

Một vấn đề kỹ thuật phát sinh trong quá trình gộp dữ liệu là sự không đồng nhất về định dạng thời gian: một số bản ghi sử dụng đơn vị milliseconds (10^{13}), trong khi số khác sử dụng microseconds (10^{16}). Nhóm em đã xử lý vấn đề này bằng cách thiết lập ngưỡng lọc (threshold) để đưa tất cả về cùng một đơn vị chuẩn, sau đó chuyển đổi sang định dạng `datetime`. Cuối cùng, dữ liệu được chia thành tập Huấn luyện (Train) và Kiểm thử (Test).

Quy trình xử lý chi tiết được thể hiện qua đoạn mã sau:

```
# 1. Tải và gộp dữ liệu từ các thư mục con
```

```
df_list = load_data(monthly_path) + load_data(daily_path)
df_combined = pd.concat(df_list, ignore_index=True)

# 2. Xử lý định dạng thời gian hỗn hợp (Mixed Timestamps)
# Ngưỡng 10^13 dùng để phân biệt milliseconds và microseconds
df_combined['open_time'] = np.where(
    df_combined['open_time'] > 10_000_000_000_000,
    df_combined['open_time'] / 1000,    # Chuyển đổi về ms nếu quá lớn
    df_combined['open_time']            # Giữ nguyên nếu đã là ms
)

# Chuyển đổi sang đối tượng Datetime
df_combined['open_time'] = pd.to_datetime(df_combined['open_time'], unit='ms')

# 3. Làm sạch, sắp xếp và loại bỏ dữ liệu thừa
df_combined = df_combined.sort_values(by='open_time') \
                         .drop_duplicates(subset=['open_time']) \
                         .reset_index(drop=True)
df_combined = df_combined.drop(columns=['ignored', 'close_time'])

# 4. Phân chia tập dữ liệu (Train/Test Split)
# Train: 2023-01-01 -> 2024-12-31 | Test: 01-01-2025 trở đi
train_end_date = datetime.datetime(2024, 12, 31, 23, 59, 59)
df_train = df_combined[df_combined['open_time'] <= train_end_date]
df_test = df_combined[df_combined['open_time'] > train_end_date]

# Lưu trữ kết quả
df_train.to_csv("Data/bitcoin_data_train.csv", index=False)
df_test.to_csv("Data/bitcoin_data_test.csv", index=False)
```

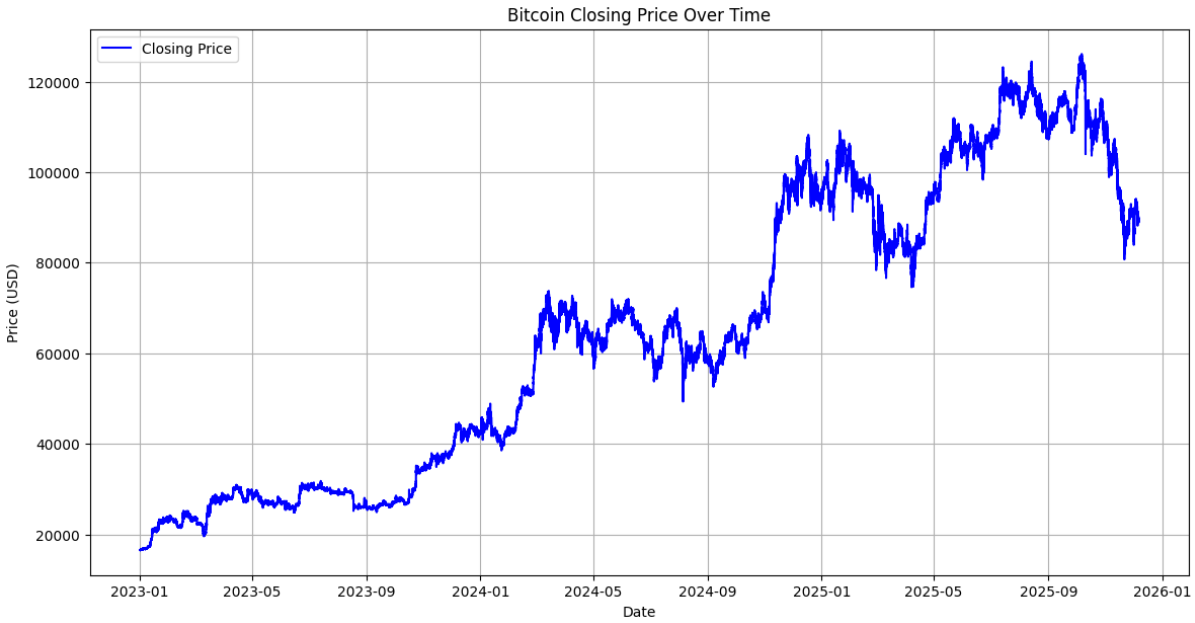
2.1.3 Đặc điểm dữ liệu (Data Understanding)

Bảng 2.1: Thống kê mô tả dữ liệu giá Bitcoin (2023-2025)

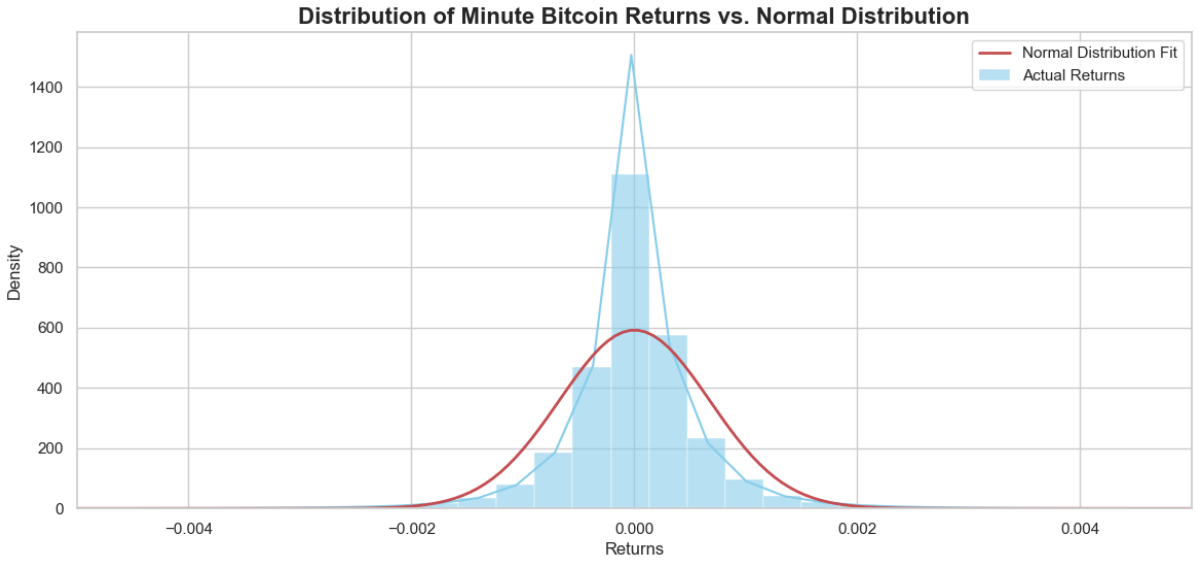
Statistic	Close Price ($)	Volume (USD)	Returns	Intra-Volatility
Mean	47,376.86	1,681,654	0.000002	0.0654
Std Dev	21,659.57	2,969,644	0.000673	0.0769
Min	16,505.87	0	-0.036800	0.0000
25%	27,728.17	346,496	-0.000253	0.0211
50%	42,696.65	771,385	0.000000	0.0462
75%	64,222.99	1,817,253	0.000257	0.0849
Max	108,258.39	256,885,789	0.031600	6.2983
Skewness	-	-	**-0.640**	**6.917**
Kurtosis	-	-	**82.958**	**171.868**

Nhận xét dữ liệu:

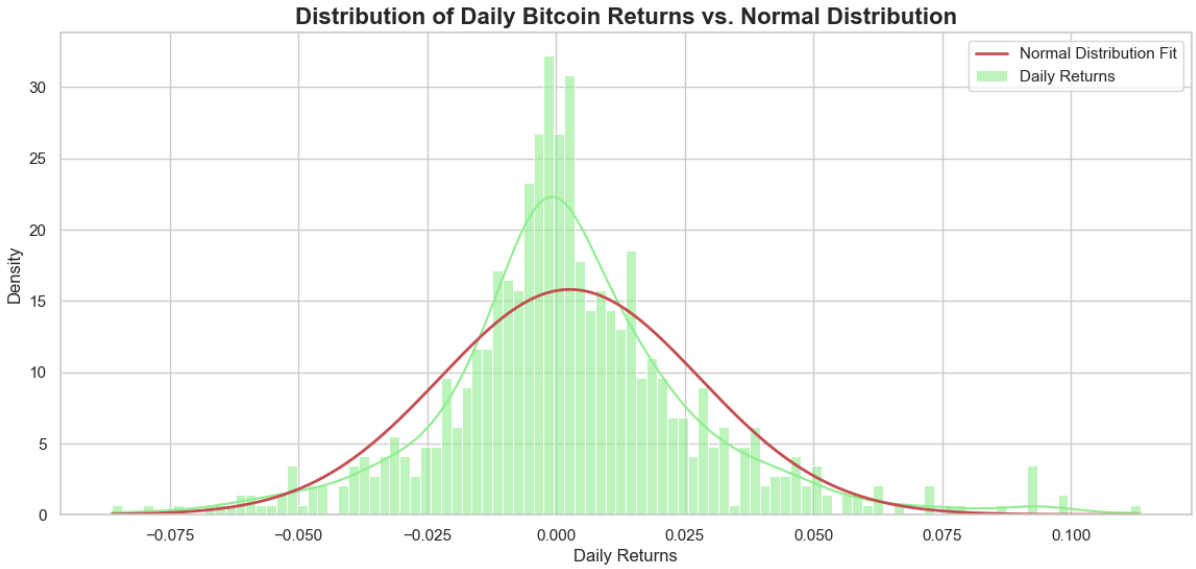
- **Biến động giá (Close Price):** Giá Bitcoin trải dài từ mức thấp nhất khoảng 16,500$ lên đến đỉnh điểm hơn 108,000$. Độ lệch chuẩn lớn (21,659$) cho thấy rủi ro biến động giá là rất cao.
- **Tính chất phân phối (Kurtosis & Skewness):** Chỉ số Kurtosis của lợi nhuận (*returns*) đạt mức cực đại (82.96), lớn hơn rất nhiều so với phân phối chuẩn (Kurtosis = 3). Điều này minh chứng cho hiệu ứng "đuôi dày" (Fat-tailed distribution), tức là thị trường thường xuyên xuất hiện các cú sốc giá (tăng/giảm đột ngột) nằm ngoài dự đoán thông thường. Chỉ số Skewness âm (-0.64) cho thấy các phiên giảm giá thường có cường độ mạnh hơn so với các phiên tăng giá.



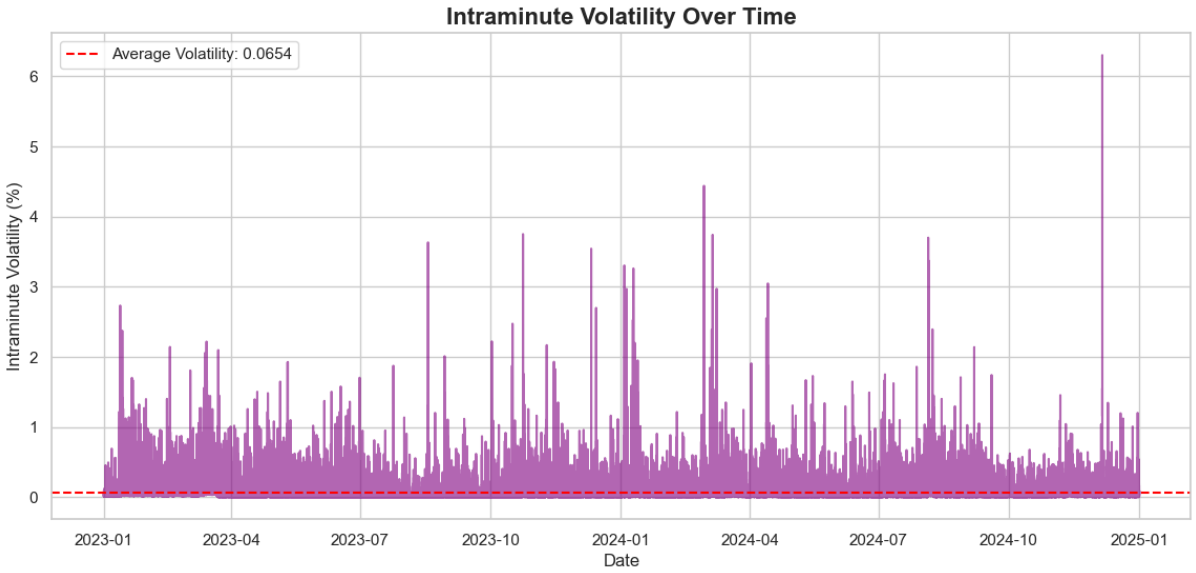
Hình 2.1: Giá đóng Bitcoin từ đầu năm 2023 đến nay



Hình 2.2: Biểu đồ Phân bố lợi nhuận theo phút của Bitcoin



Hình 2.3: Biểu đồ Phân bố lợi nhuận theo ngày của Bitcoin



Hình 2.4: Độ biến động nội tại theo phút của Bitcoin giai đoạn 2023-2025

2.2 Dữ liệu tin tức (Text Data)

2.2.1 Nguồn dữ liệu và Thu thập

Đối với dữ liệu văn bản, chúng em sử dụng bộ dữ liệu **Crypto News Dataset** được công bố trên GitHub[1]. Đây là tập hợp hơn 248,000 bản tin được thu thập từ *CryptoPanic* - nền tảng tổng hợp tin tức tiền mã hóa lớn nhất hiện nay.

Bộ dữ liệu bao gồm các thông tin liên quan đến 660 loại tiền mã hóa hàng đầu, với các trường dữ liệu:

- **Metadata:** `id`, `newsDatetime`, `sourceDomain`, `url`.
- **Nội dung:** `title` (tiêu đề), `description` (mô tả).

[1]Nguồn: https://github.com/soheilrahsaz/cryptoNewsDataset

- **Phản ứng người dùng (User Reactions):** Các chỉ số cảm xúc được dán nhãn bởi cộng đồng như `positive`, `negative`, `important`, `toxic`, `liked`, `saved`, v.v.

Để phù hợp với phạm vi đề tài, ta tiến hành lọc và chỉ giữ lại các bản tin có trường `currencies` chứa mã "BTC" và nằm trong khung thời gian nghiên cứu (2023-2025).

2.2.2 Làm sạch và Tiền xử lý văn bản

Dữ liệu văn bản thô từ web thường chứa nhiều nhiễu (HTML tags, URL, ký tự lạ) gây ảnh hưởng đến hiệu suất của thuật toán xử lý ngôn ngữ tự nhiên. Nhóm em đã xây dựng hàm `clean_text` để chuẩn hóa dữ liệu tiêu đề.

Quy trình làm sạch bao gồm: Giải mã HTML, loại bỏ đường dẫn, xóa tên người dùng/hashtag và các ký tự phi ASCII. Dưới đây là đoạn mã thực hiện:

```
import html
import re

def clean_text(text):
    if not isinstance(text, str):
        return ""

    # 1. Giải mã các ký tự HTML (ví dụ: &amp; -> &)
    text = html.unescape(text)

    # 2. Loại bỏ các đường dẫn URL (http/https/www)
    text = re.sub(r'http\S+|www\S+|https\S+', '', text, flags=re.MULTILINE)

    # 3. Loại bỏ User Handles (@user) và Hashtags (#btc)
    text = re.sub(r'@\w+|#\w+', '', text)

    # 4. Loại bỏ các ký tự không phải mã ASCII (emoji, ký tự lạ)
    text = re.sub(r'[^\x00-\x7F]+', ' ', text)

    # 5. Loại bỏ ký hiệu Retweet (RT)
    text = re.sub(r'\bRT\b', '', text)

    # 6. Xử lý khoảng trắng thừa
    text = re.sub(r'\s+', ' ', text).strip()

    return text
```

2.2.3 Xử lý vấn đề dữ liệu thưa (Sparsity Handling)

Vấn đề: Một thách thức lớn khi làm việc với dữ liệu tương tác người dùng là tính thưa thớt (Sparsity). Thống kê sơ bộ cho thấy hầu hết các trường dữ liệu phản ứng (Reactions) đều có giá trị bằng 0 chiếm tỷ trọng áp đảo (từ 82% đến 99.9%).

Bảng 2.2: Thống kê tỷ lệ dữ liệu thưa (Sparsity) trước khi xử lý

Trường dữ liệu	**Số lượng giá trị 0**	**Tỷ lệ (%)**
Toxic	17,319	99.92%
Disliked	16,025	92.45%
Negative	15,865	91.53%
Comments	15,787	91.08%
Lol	15,281	88.16%
Saved	15,312	88.34%
Important	14,805	85.42%
Liked	14,247	82.20%
Positive	14,243	82.17%

Trong bài toán Khai phá luật kết hợp (Association Rule Mining), dữ liệu quá thưa sẽ dẫn đến việc thuật toán sinh ra hàng loạt luật "rác" (trivial rules). Ví dụ: *Nếu Toxic = 0 thì Disliked = 0.* Những luật này có độ tin cậy (Confidence) cao nhưng không mang lại giá trị tri thức, làm lu mờ các luật quan trọng nhưng xuất hiện ít hơn.

Để khắc phục, ta áp dụng kỹ thuật **Gộp đặc trưng (Feature Aggregation)**. Thay vì giữ nguyên các cột rời rạc, tổng hợp chúng thành các nhóm chỉ số có ý nghĩa bao quát hơn:

- **Reaction_Positive:** Tổng hợp các phản ứng tích cực (Positive + Liked + Important + Saved).
- **Reaction_Negative:** Tổng hợp các phản ứng tiêu cực (Negative + Disliked + Toxic + Lol).
- **Total_Reactions:** Tổng lượng tương tác.

Đoạn mã xử lý như sau:

```
# Gộp các cột để giảm tính thưa của dữ liệu
df_news['Reaction_Positive'] = (df_news['positive'] + df_news['liked'] +
                                df_news['important'] + df_news['saved'])

df_news['Reaction_Negative'] = (df_news['negative'] + df_news['disliked'] +
                                df_news['toxic'] + df_news['lol'])

df_news['Total_Reactions'] = (df_news['Reaction_Positive'] +
                              df_news['Reaction_Negative'] +
                              df_news['comments'])

# Loại bỏ các cột gốc sau khi đã gộp
df_news = df_news.drop(columns=['positive', 'liked', 'important', 'saved',
                                'negative', 'disliked', 'toxic', 'lol', 'comments'])
```

Kết quả: Sau khi gộp, tỷ lệ dữ liệu thưa đã giảm đáng kể, giúp tập dữ liệu trở nên "đậm đặc" thông tin hơn (Dense), sẵn sàng cho việc xây dựng mô hình.

- **Reaction_Positive:** Tỷ lệ giá trị 0 giảm xuống còn **76.89%** (so với mức >82% của các cột thành phần).
- **Reaction_Negative:** Tỷ lệ giá trị 0 giảm xuống còn **85.94%** (cải thiện rõ rệt so với mức 99.92% của cột Toxic).
- **Total_Reactions:** Tỷ lệ giá trị 0 chỉ còn **72.53%**.

Chương 3

Kỹ thuật tạo đặc trưng dữ liệu

Mục tiêu cốt lõi của quá trình tạo đặc trưng trong nghiên cứu này là chuyển đổi dữ liệu thô từ nhiều nguồn khác nhau thành các tín hiệu định lượng giàu thông tin, giúp các thuật toán máy học có thể nhận diện được những mẫu hình tiềm ẩn. Đối với dữ liệu lịch sử giá (Numerical Data), nếu chỉ sử dụng các giá trị OHLCV nguyên bản, thuật toán phân cụm sẽ có xu hướng gom nhóm dựa trên độ lớn của giá thay vì bản chất vận động của thị trường. Do đó, dữ liệu cần được chuyển đổi thành các đặc trưng đại diện cho xu hướng, động lượng và dấu vết của dòng tiền lớn để phục vụ việc xác định cấu trúc thị trường (ví dụ: các pha tích lũy hay phân phối trong chu kỳ Wyckoff). Song song với đó, đối với dữ liệu tin tức (Text Data), thách thức nằm ở việc lượng hóa ngôn ngữ tự nhiên thành các chỉ số cảm xúc và mức độ quan tâm của cộng đồng, từ đó tích hợp vào mô hình dự báo tổng thể.

Chương này trình bày chi tiết quy trình xây dựng đặc trưng cho hai nguồn dữ liệu trên, bắt đầu với các chỉ báo kỹ thuật và khái niệm tiền thông minh áp dụng cho giá Bitcoin, tiếp theo là kỹ thuật xử lý và tổng hợp đặc trưng từ dữ liệu văn bản.

3.1 Tạo đặc trưng với dữ liệu lịch sử giá Bitcoin

3.1.1 Các chỉ báo kỹ thuật (Technical Indicators)

Nhóm các chỉ báo kỹ thuật được tính toán nhằm cung cấp cho mô hình góc nhìn định lượng về trạng thái hiện tại của thị trường. Nghiên cứu tập trung sử dụng bốn chỉ báo chính: Đường trung bình động lũy thừa (EMA), Chỉ số sức mạnh tương đối (RSI), Khoảng dao động trung bình thực tế (ATR) và Khối lượng đột biến (Shock Volume).

Đầu tiên, đường EMA (Exponential Moving Average) được sử dụng để xác định xu hướng chủ đạo. Khác với đường trung bình động đơn giản (SMA), EMA đặt trọng số lớn hơn vào các dữ liệu giá gần nhất, giúp chỉ báo phản ứng nhạy bén hơn với các biến động mới. Công thức tính EMA tại thời điểm t được biểu diễn như sau:

$$EMA_t = \alpha \cdot P_t + (1 - \alpha) \cdot EMA_{t-1} \tag{3.1}$$

Trong đó, P_t là giá đóng cửa tại thời điểm hiện tại, và hệ số làm mượt $\alpha = \frac{2}{N+1}$ với N là chu kỳ của đường trung bình. Hai đường EMA chu kỳ 50 ($N = 50$) và 200 ($N = 200$) được thiết lập; vị trí tương đối của giá so với hai đường này sẽ cho biết thị trường đang trong xu hướng tăng (Uptrend) hay giảm (Downtrend).

Tiếp theo, chỉ báo RSI (Relative Strength Index) chu kỳ 14 được tính toán để đo lường động lượng (Momentum). RSI so sánh độ lớn của các đợt tăng giá so với giảm giá nhằm nhận diện các vùng "Quá mua" hoặc "Quá bán". Chỉ số này được xác định dựa trên tỷ lệ sức mạnh tương đối (RS):

$$RSI = 100 - \frac{100}{1 + RS} \quad \text{với} \quad RS = \frac{\text{Average Gain}}{\text{Average Loss}} \tag{3.2}$$

Giá trị RS là tỷ số giữa mức tăng trung bình và mức giảm trung bình trong 14 phiên gần nhất. Khi $RSI > 70$, thị trường được xem là quá mua, và ngược lại khi $RSI < 30$ là vùng quá bán, nơi xác suất đảo chiều thường tăng cao.

Để đo lường mức độ rủi ro và biến động, chỉ báo ATR (Average True Range) được áp dụng. Trước hết, giá trị dao động thực (True Range - TR) của mỗi phiên được xác định là giá trị lớn nhất trong ba

đại lượng: biên độ phiên hiện tại ($High - Low$), chênh lệch giữa giá cao nhất và giá đóng cửa phiên trước ($|High - Close_{prev}|$), và chênh lệch giữa giá thấp nhất và giá đóng cửa phiên trước ($|Low - Close_{prev}|$).

$$TR_t = \max(H_t - L_t, |H_t - C_{t-1}|, |L_t - C_{t-1}|) \tag{3.3}$$

Chỉ báo ATR sau đó được tính bằng cách lấy trung bình trượt của chuỗi TR trong 14 phiên. Một giá trị ATR cao cho thấy thị trường đang biến động mạnh, thường xuất hiện ở các giai đoạn bùng nổ hoặc sập giá, trong khi ATR thấp báo hiệu giai đoạn tích lũy nén giá.

Cuối cùng, đặc trưng "Shock Volume" được tạo ra để phát hiện sự tham gia của các tổ chức tài chính lớn. Đây là một biến nhị phân, nhận giá trị 1 khi khối lượng giao dịch hiện tại (V_t) vượt quá hai lần mức trung bình của 20 phiên gần nhất ($MA_{20}(V)$), và nhận giá trị 0 trong các trường hợp còn lại:

$$ShockVol_t = \begin{cases} 1 & \text{nếu } V_t > 2 \times MA_{20}(V) \\ 0 & \text{nếu } V_t \leq 2 \times MA_{20}(V) \end{cases} \tag{3.4}$$

Sự đột biến về khối lượng này đóng vai trò như một tín hiệu xác nhận (confirmation signal), gia tăng độ tin cậy cho các xu hướng giá đang diễn ra.

Đoạn mã dưới đây thể hiện quy trình tính toán các chỉ báo trên:

```
# 1. Đường trung bình động lũy thừa (EMA)
# EMA 50 và 200 giúp xác định xu hướng trung và dài hạn
df_featured['ema_50'] = df_featured['close'].ewm(span=50, adjust=False).mean()
df_featured['ema_200'] = df_featured['close'].ewm(span=200, adjust=False).mean()

# 2. Chỉ số sức mạnh tương đối (RSI - 14 period)
# Đo lường động lượng và xác định vùng quá mua/quá bán
delta = df_featured['close'].diff()
gain = (delta.where(delta > 0, 0)).rolling(window=14).mean()
loss = (-delta.where(delta < 0, 0)).rolling(window=14).mean()
rs = gain / loss
df_featured['rsi_14'] = 100 - (100 / (1 + rs))

# 3. Khoảng dao động trung bình (ATR - 14 period)
# Đo lường mức độ biến động (Volatility) của thị trường
high_low = df_featured['high'] - df_featured['low']
high_close = np.abs(df_featured['high'] - df_featured['close'].shift())
low_close = np.abs(df_featured['low'] - df_featured['close'].shift())
ranges = pd.concat([high_low, high_close, low_close], axis=1)
true_range = ranges.max(axis=1)
df_featured['atr_14'] = true_range.rolling(window=14).mean()

# 4. Khối lượng đột biến (Shock Volume)
# Xác định dòng tiền lớn: Volume > 2 lần trung bình 20 phiên
df_featured['vol_ma20'] = df_featured['volume_usd'].rolling(window=20).mean()
df_featured['is_shock_vol'] = np.where(
    df_featured['volume_usd'] > 2 * df_featured['vol_ma20'], 1, 0
)
```

3.1.2 Khái niệm Tiền thông minh (Smart Money Concepts - SMC)

Bên cạnh các chỉ báo dao động truyền thống, nghiên cứu đi sâu vào việc trích xuất các đặc trưng hành vi dựa trên phương pháp Smart Money Concepts (SMC). Mục đích cốt lõi của phần này là tìm kiếm dấu vết của các nhà tạo lập thị trường (Market Makers) thông qua việc phân tích cấu trúc nến và các vùng mất cân bằng cung cầu.

Quy trình bắt đầu bằng việc giải phẫu cấu trúc nến (Candle Anatomy) để định lượng hành vi giá trong từng phiên giao dịch. Các thông số thành phần bao gồm kích thước thân nến ($Body$), độ dài râu nến trên ($Wick_{upper}$) và râu nến dưới ($Wick_{lower}$) được tính toán như sau:

$$
\begin{aligned}
Body &= |Close - Open| \\
Wick_{upper} &= High - \max(Open, Close) \\
Wick_{lower} &= \min(Open, Close) - Low
\end{aligned} \tag{3.5}
$$

Việc tách biệt này là tiền đề quan trọng, bởi lẽ râu nến dài thường thể hiện sự từ chối giá (Rejection) quyết liệt tại các vùng thanh khoản quan trọng.

Dựa trên các thông số này, Khối lệnh (Order Block - OB) được định nghĩa là vùng giá mà tại đó phe mua hoặc phe bán đã tham gia với khối lượng lớn, thường dẫn đến sự đảo chiều xu hướng. Một "Bullish Order Block" (Khối lệnh tăng) được xác định khi hội tụ đủ ba yếu tố: xuất hiện nến Pinbar với râu dưới chiếm ưu thế (gấp đôi thân nến), râu dưới dài hơn râu trên, và đi kèm với tín hiệu khối lượng đột biến ($ShockVol = 1$). Logic này được biểu diễn qua điều kiện:

$$
IsBullishOB = (Wick_{lower} > 2 \cdot Body) \wedge (Wick_{lower} > Wick_{upper}) \wedge (ShockVol == 1) \tag{3.6}
$$

Ngược lại, "Bearish Order Block" (Khối lệnh giảm) được xác định khi nến có râu trên chiếm ưu thế ($Wick_{upper} > 2 \cdot Body$) và $Wick_{upper} > Wick_{lower}$, báo hiệu áp lực bán tháo mạnh từ phe Gấu.

Một khái niệm quan trọng khác là Khoảng trống giá trị hợp lý (Fair Value Gap - FVG), đại diện cho sự mất cân bằng thanh khoản (Liquidity Imbalance). FVG xuất hiện khi giá di chuyển quá nhanh về một hướng, tạo ra khoảng hở giữa râu của cây nến thứ nhất ($t-2$) và cây nến thứ ba (t). Về mặt thuật toán, một Bullish FVG được xác nhận khi giá thấp nhất của nến hiện tại cao hơn giá cao nhất của nến cách đó 2 phiên, với độ lớn khoảng trống vượt qua một ngưỡng nhiễu (δ):

$$
(L_t > H_{t-2}) \wedge ((L_t - H_{t-2}) > \delta) \wedge (C_t > O_t) \tag{3.7}
$$

Trong đó L, H, C, O lần lượt là giá Low, High, Close, Open. Tương tự, Bearish FVG xuất hiện khi $H_t < L_{t-2}$. Những vùng FVG này đóng vai trò như các "nam châm" hút giá quay trở lại để tái cân bằng (Rebalancing) trong tương lai.

Dưới đây là mã nguồn chi tiết hiện thực hóa việc trích xuất các đặc trưng SMC:

```
def calculate_smart_money_features(df):
    df_smc = df.copy()

    # --- 1. Giải phẫu cấu trúc nến (Candle Anatomy) ---
    # Tính kích thước thân nến và độ dài râu nến
    body_size = np.abs(df_smc['close'] - df_smc['open'])
    upper_wick = df_smc['high'] - df_smc[['open', 'close']].max(axis=1)
    lower_wick = df_smc[['open', 'close']].min(axis=1) - df_smc['low']

    # Xác định điều kiện Volume đột biến (Whale Activity)
    avg_vol = df_smc['volume'].rolling(window=20).mean()
    is_shock_vol = df_smc['volume'] > (2 * avg_vol)

    # --- 2. Xác định Khối lệnh (Order Blocks) ---
    # Bullish OB: Râu dưới dài (từ chối giá giảm) + Volume lớn
    df_smc['is_bullish_ob'] = (
        (lower_wick > 2 * body_size) &  # Râu dài gấp đôi thân
        (lower_wick > upper_wick) &     # Râu dưới là chủ đạo
        (is_shock_vol)                  # Có dòng tiền lớn bảo trợ
    )

    # Bearish OB: Râu trên dài (từ chối giá tăng) + Volume lớn
    df_smc['is_bearish_ob'] = (
        (upper_wick > 2 * body_size) &
        (upper_wick > lower_wick) &
        (is_shock_vol)
    )

    # --- 3. Xác định Khoảng trống giá trị (Fair Value Gaps) ---
```

```
    # So sánh giá giữa nến hiện tại và nến trước đó 2 phiên
    prev_high = df_smc['high'].shift(2)
    prev_low = df_smc['low'].shift(2)
    threshold = df_smc['close'] * 0.0005 # Ngưỡng lọc nhiễu 0.05%

    # Bullish FVG: Giá thấp nhất nến hiện tại > Giá cao nhất nến t-2
    bullish_gap_size = df_smc['low'] - prev_high
    df_smc['is_fvg_bullish'] = (
        (df_smc['low'] > prev_high) &
        (bullish_gap_size > threshold) &
        (df_smc['close'] > df_smc['open'])
    )

    # Bearish FVG: Giá cao nhất nến hiện tại < Giá thấp nhất nến t-2
    bearish_gap_size = prev_low - df_smc['high']
    df_smc['is_fvg_bearish'] = (
        (df_smc['high'] < prev_low) &
        (bearish_gap_size > threshold) &
        (df_smc['close'] < df_smc['open'])
    )

    # Điền giá trị False cho các ô trống
    cols = ['is_bullish_ob', 'is_bearish_ob', 'is_fvg_bullish', 'is_fvg_bearish']
    df_smc[cols] = df_smc[cols].fillna(False)

    return df_smc
```

Việc kết hợp các chỉ báo kỹ thuật với các cấu trúc SMC giúp bộ dữ liệu không chỉ phản ánh được "giá đang ở đâu" mà còn gợi ý "tại sao giá lại di chuyển như vậy". Đây là tiền đề quan trọng để thuật toán phân cụm ở chương sau có thể gom nhóm các trạng thái thị trường dựa trên bản chất hành vi thay vì chỉ dựa trên biên độ giá đơn thuần.

3.2 Tạo đặc trưng với dữ liệu tin tức (News Features)

Sau khi xử lý dữ liệu giá, ta chuyển sang khai thác nguồn dữ liệu tin tức. Mặc dù dữ liệu tin tức thường được gắn liền với xử lý ngôn ngữ tự nhiên (sẽ được trình bày chi tiết trong chương Phân cụm), nhưng các siêu dữ liệu đi kèm như số lượng yêu thích (Like), bình luận (Comment) hay cảm xúc (Sentiment) lại mang tính định lượng rất cao. Đây là thước đo trực tiếp cho tâm lý đám đông (Crowd Sentiment) tại thời điểm tin tức được công bố.

Tuy nhiên, việc sử dụng trực tiếp số lượng tương tác thô (Raw count) chứa đựng nhiều rủi ro. Ví dụ: 100 lượt like trong giai đoạn thị trường sôi động (Uptrend) là con số bình thường, nhưng 100 lượt like trong giai đoạn thị trường ảm đạm (Downtrend) lại là một sự đột biến lớn. Nếu chỉ sử dụng giá trị tuyệt đối hoặc chuẩn hóa theo toàn bộ tập dữ liệu (Global Scaling), mô hình sẽ mắc lỗi "nhìn trước tương lai" (Look-ahead bias) do sử dụng thông tin của tương lai để tính toán cho quá khứ.

Để giải quyết vấn đề này, ta áp dụng kỹ thuật **Chuẩn hóa động (Dynamic Scaling)** sử dụng cửa sổ trượt (Rolling Window). Cụ thể, tại mỗi thời điểm t, mức độ "bất thường" của phản ứng người dùng được tính toán dựa trên độ lệch chuẩn so với trung bình của 24 giờ trước đó (Z-Score).

$$Z_t = \frac{X_t - \mu_{t-24h}}{\sigma_{t-24h}} \tag{3.8}$$

Phương pháp này giúp dữ liệu đầu vào của mô hình luôn phản ánh đúng bối cảnh thị trường tại thời điểm đó mà không vi phạm nguyên tắc nhân quả.

Dưới đây là đoạn mã thực hiện quy trình tính toán Z-Score cho cảm xúc và mức độ quan tâm (Hype):

```
def engineer_sentiment_volume(df, window_hours=24):
    df_feat = df.copy().set_index('newsDatetime').sort_index()

    # Định nghĩa cửa sổ trượt (Rolling Window)
```

```
    rolling_window = f'{window_hours}h'

    # --- 1. Tính Z-Score cho Cảm xúc (Sentiment) ---
    # Tính riêng cho Positive và Negative để tăng độ nhạy
    # Thay thế std=0 bằng 1 để tránh lỗi chia cho 0

    # Positive Component
    pos_mean = df_feat['Reaction_Positive'].rolling(rolling_window).mean()
    pos_std  = df_feat['Reaction_Positive'].rolling(rolling_window).std().replace(0, 1)
    df_feat['Z_Pos'] = (df_feat['Reaction_Positive'] - pos_mean) / pos_std

    # Negative Component
    neg_mean = df_feat['Reaction_Negative'].rolling(rolling_window).mean()
    neg_std  = df_feat['Reaction_Negative'].rolling(rolling_window).std().replace(0, 1)
    df_feat['Z_Neg'] = (df_feat['Reaction_Negative'] - neg_mean) / neg_std

    # Net Sentiment Z-Score = Độ nổi bật Tích cực - Độ nổi bật Tiêu cực
    df_feat['Sentiment_Z_Score'] = df_feat['Z_Pos'] - df_feat['Z_Neg']

    # --- 2. Tính Z-Score cho Mức độ quan tâm (Volume Hype) ---
    vol_mean = df_feat['Total_Reactions'].rolling(rolling_window).mean()
    vol_std  = df_feat['Total_Reactions'].rolling(rolling_window).std().replace(0, 1)
    df_feat['Volume_Z_Score'] = (df_feat['Total_Reactions'] - vol_mean) / vol_std

    # Xử lý trường hợp không có tương tác (Hard Neutral)
    df_feat.loc[df_feat['Total_Reactions'] == 0, 'Sentiment_Z_Score'] = 0

    return df_feat.reset_index()

# Thực thi tính toán
df_features = engineer_sentiment_volume(df_news)
```

Sau khi tính toán được các chỉ số Z-Score liên tục, để phục vụ cho bài toán Khai phá luật kết hợp (vốn yêu cầu dữ liệu dạng hạng mục/categorical), ta tiến hành rời rạc hóa (discretization) các giá trị này thành các nhãn.

Đối với chỉ số cảm xúc (`Sentiment_Z_Score`), dữ liệu được phân thành 3 nhóm: Tiêu cực (Negative), Trung tính (Neutral) và Tích cực (Positive). Kết quả phân phối cho thấy phần lớn các bản tin (khoảng 14,500 bản tin) rơi vào trạng thái trung tính, trong khi số lượng tin tức tích cực (1,577) nhỉnh hơn so với tiêu cực (1,242).

```
# Phân loại Cảm xúc (Sentiment Binning)
df_features['Sentiment_label'] = pd.cut(
    df_features['Sentiment_Z_Score'],
    bins=[-np.inf, -0.5, 0.5, np.inf],
    labels=['Negative', 'Neutral', 'Positive']
)
# Kết quả phân phối:
# Neutral: 14514 | Positive: 1577 | Negative: 1242
```

Tương tự, đối với mức độ quan tâm (`Volume_Z_Score`), dữ liệu được chia thành mức Thấp, Trung bình và Cao. Đáng chú ý có 953 bản tin được gán nhãn "High", tương ứng với những sự kiện gây chấn động cộng đồng (High Hype), đây là những điểm dữ liệu quan trọng mà mô hình cần tập trung khai thác.

```
# Phân loại Mức độ quan tâm (Volume Hype Binning)
df_features['Volume_label'] = pd.cut(
    df_features['Volume_Z_Score'],
    bins=[-np.inf, -0.5, 2, np.inf],
    labels=['Low', 'Medium', 'High']
```

```
)
# Kết quả phân phối:
# Medium: 14514 | Low: 1866 | High: 953
```

Việc chuyển đổi từ dữ liệu thô sang các nhãn định tính dựa trên thống kê động này giúp loại bỏ nhiễu và làm nổi bật các tín hiệu thị trường thực sự có ý nghĩa.

Chương 4

Phân cụm dữ liệu

Sau khi đã trích xuất được các đặc trưng từ dữ liệu thô, bước tiếp theo là áp dụng thuật toán học máy không giám sát để phân chia dữ liệu thành các nhóm có tính chất tương đồng. Chương này trình bày quy trình phân cụm dữ liệu nhằm hai mục tiêu chính: xác định các pha của thị trường (Market Regimes) dựa trên hành vi giá và gom nhóm các chủ đề tin tức dựa trên nội dung văn bản.

4.1 Phân cụm với dữ liệu giá Bitcoin

Mục tiêu của phần này là tự động gán nhãn cho từng phiên giao dịch vào một trong bốn pha của chu kỳ thị trường theo lý thuyết Wyckoff: Tích lũy (Accumulation), Tăng trưởng (Markup), Phân phối (Distribution), và Suy thoái (Markdown).

4.1.1 Chuẩn bị dữ liệu để phân cụm

Để thuật toán phân cụm hoạt động hiệu quả, dữ liệu đầu vào không thể là giá trị OHLCV đơn thuần mà phải là các chỉ số đại diện cho "trạng thái" (State) của thị trường. Nhóm đã xây dựng bộ 5 đặc trưng chuyên biệt (Wyckoff Features) để mô tả xu hướng, hiệu suất nến và cấu trúc sóng.

Đầu tiên là đặc trưng **Độ chín của xu hướng (Trend Maturity)**. Chỉ số này đo lường khoảng cách giữa xu hướng trung hạn (EMA 50) và dài hạn (EMA 200). Khoảng cách càng lớn chứng tỏ xu hướng đã diễn ra trong thời gian dài và có tính bền vững.

$$Trend_{maturity} = \frac{EMA_{50} - EMA_{200}}{Close} \times 100 \tag{4.1}$$

Thứ hai là **Vị thế chiến thuật (Trend Tactical)**, đo lường độ lệch của giá hiện tại so với đường trung bình EMA 50. Chỉ số này cho biết giá đang "quá nóng" (Overextended) hay đang ở vùng cân bằng.

$$Trend_{tactical} = \frac{Close - EMA_{50}}{EMA_{50}} \times 100 \tag{4.2}$$

Thứ ba là **Hiệu suất nến (Efficiency Regime)**. Trong lý thuyết Wyckoff, nỗ lực (biên độ nến) phải tương xứng với kết quả (thân nến). Một cây nến có thân lớn và râu ngắn thể hiện sự dứt khoát (Hiệu suất cao - thường thấy ở pha Markup/Markdown), ngược lại nến thân nhỏ râu dài thể hiện sự lưỡng lự (Hiệu suất thấp - thường thấy ở pha Tích lũy/Phân phối).

$$Efficiency = \frac{|Close - Open|}{High - Low + \epsilon} \tag{4.3}$$

(Trong đó ϵ là hằng số nhỏ để tránh lỗi chia cho 0).

Thứ tư là **Chế độ khối lượng (Volume Regime)**, tính bằng tỷ lệ giữa khối lượng hiện tại và trung bình 20 phiên. Khối lượng đột biến thường báo hiệu các điểm đảo chiều (Climax).

Cuối cùng và quan trọng nhất là **Điểm cấu trúc (Structure Score)**. Đây là một biến có tính "ghi nhớ" (memory), được tính cộng dồn theo thời gian với hệ số suy giảm (decay factor $\gamma = 0.98$). Điểm số sẽ tăng khi xuất hiện Bullish Order Block hoặc FVG, và giảm khi xuất hiện các tín hiệu Bearish.

$$Score_t = Score_{t-1} \times \gamma + \text{Signal}_t \tag{4.4}$$

Trong đó Signal_t nhận giá trị +1 nếu là Bullish OB, -1 nếu là Bearish OB. Đặc trưng này giúp mô hình nhận diện được phe nào đang kiểm soát thị trường về mặt cấu trúc.

4.1.2 Phân cụm và Diễn giải kết quả

Với giả thuyết thị trường vận động theo 4 pha chính, chúng em lựa chọn thuật toán **K-Means** với số cụm $k = 4$. Dữ liệu trước khi đưa vào mô hình được chuẩn hóa bằng `StandardScaler` để đưa về cùng một phân phối chuẩn, đảm bảo các đặc trưng có trọng số tương đương nhau.

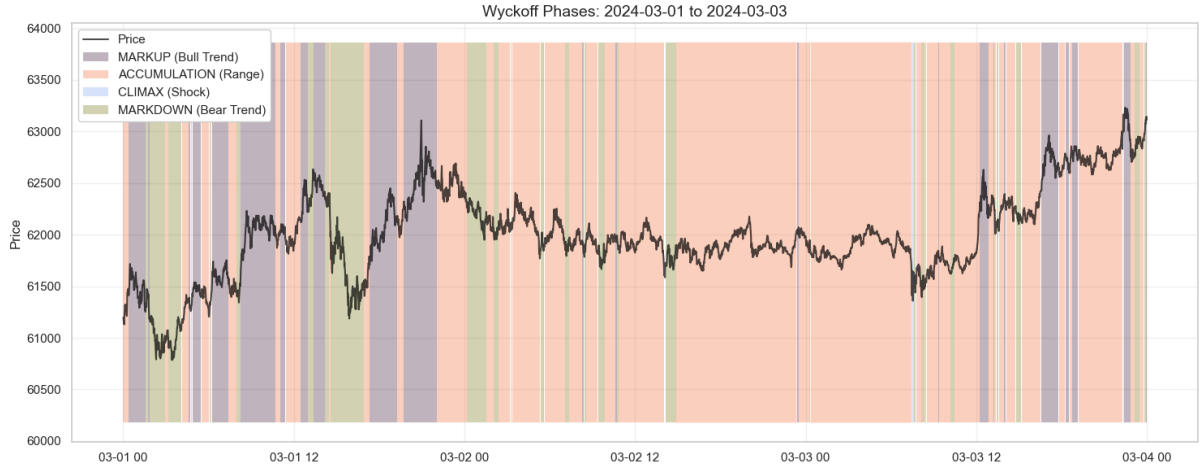
Sau khi thực hiện phân cụm, ta tiến hành phân tích giá trị trung bình (Centroids) của các đặc trưng trong từng cụm để gán nhãn ý nghĩa tài chính. Bảng dưới đây tóm tắt các đặc điểm chính:

Bảng 4.1: Đặc điểm trung bình của các cụm (Cluster Centroids)

Cluster	Tactical	Maturity	Efficiency	Volume	Structure	Số lượng mẫu
0	0.195	0.284	0.567	0.920	**0.781**	213,019
1	0.002	0.008	0.750	0.833	0.016	493,383
2	-0.011	0.004	0.699	**3.568**	0.049	59,253
3	-0.131	-0.188	0.529	0.896	**-0.295**	286,802

Dựa trên số liệu bảng 4.1, nhóm thực hiện biện luận và gán nhãn như sau:

- **Cụm 0 (Phase_Markup):** Đặc trưng bởi điểm cấu trúc rất cao (0.781) và độ chín xu hướng dương (0.284). Đây là giai đoạn phe Mua kiểm soát hoàn toàn, giá tăng trưởng ổn định.
- **Cụm 1 (Phase_Accumulation):** Các chỉ số xu hướng và cấu trúc đều tiệm cận 0, nhưng số lượng mẫu lại chiếm đa số (gần 500k mẫu). Đây là trạng thái thị trường đi ngang (Sideway), biên độ hẹp, tương ứng với giai đoạn tích lũy hoặc tái tích lũy.
- **Cụm 2 (Phase_Climax):** Điểm nổi bật nhất là khối lượng giao dịch cực lớn (gấp 3.5 lần trung bình). Đây là các điểm "cao trào" (Selling/Buying Climax), nơi diễn ra sự trao tay ồ ạt giữa dòng tiền thông minh và đám đông, thường báo hiệu sự đảo chiều sắp xảy ra.
- **Cụm 3 (Phase_Markdown):** Đặc trưng bởi điểm cấu trúc âm và xu hướng âm. Đây là giai đoạn giá giảm, phe Bán kiểm soát thị trường.



Hình 4.1: Kết quả phân cụm trạng thái thị trường theo lý thuyết Wyckoff trên biểu đồ giá Bitcoin (Giai đoạn mẫu: 01/03/2024 - 03/03/2024).

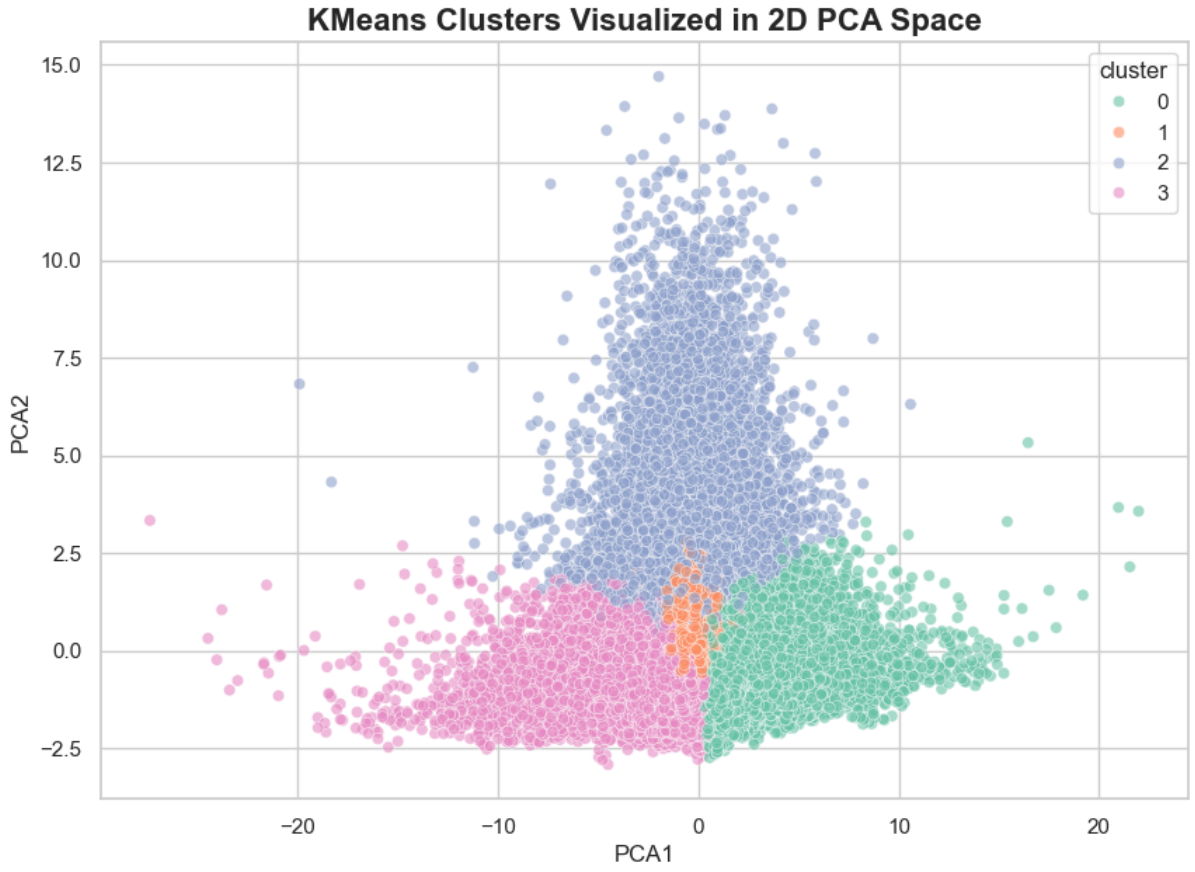
4.1.3 Đánh giá kết quả phân cụm

Để kiểm chứng chất lượng của các cụm được sinh ra, ta sử dụng hệ thống 4 chỉ số đánh giá nội tại (Internal Validation Metrics). Kết quả thu được như sau:

- **Inertia (3336295.15):** Là tổng bình phương khoảng cách từ mỗi điểm dữ liệu đến tâm cụm của nó. Giá trị này càng nhỏ càng tốt, tuy nhiên với kích thước dữ liệu lớn (hơn 1 triệu điểm), giá trị này ở mức chấp nhận được.
- **Silhouette Score (0.2191):** Đo lường mức độ tách biệt giữa các cụm, có giá trị từ -1 đến 1. Công thức:

$$S = \frac{b - a}{\max(a, b)} \tag{4.5}$$

  Trong đó a là khoảng cách trung bình nội bộ cụm, b là khoảng cách trung bình đến cụm lân cận gần nhất. Giá trị 0.22 cho thấy các cụm có sự phân tách nhưng vẫn tồn tại sự chồng lấn (overlap). Điều này hoàn toàn phù hợp với dữ liệu tài chính liên tục, nơi ranh giới giữa "Tích lũy" và "Tăng trưởng" thường không sắc nét mà có sự chuyển biến dần dần.
- **Davies-Bouldin Index (1.4146):** Tỷ lệ giữa độ phân tán nội bộ và khoảng cách giữa các cụm. Giá trị càng thấp càng tốt. Mức 1.41 là một kết quả khá tốt đối với bài toán phân cụm dữ liệu chuỗi thời gian nhiều nhiễu.
- **Calinski-Harabasz Index (202523.28):** Tỷ số giữa phương sai giữa các cụm và phương sai trong cụm. Giá trị càng cao cho thấy các cụm càng "đậm đặc" và tách biệt rõ ràng. Con số hơn 200,000 khẳng định cấu trúc phân cụm có ý nghĩa thống kê cao.



Hình 4.2: Trực quan hóa các cụm dữ liệu Wyckoff trên không gian 2 chiều sử dụng PCA.

Tổng hợp lại các phân tích trên, mô hình K-Means đã thành công trong việc tách dòng dữ liệu giá hỗn loạn thành 4 trạng thái thị trường riêng biệt.

Một điểm đáng chú ý là kết quả thực nghiệm có sự khác biệt nhỏ so với giả thuyết Wyckoff ban đầu. Thay vì tách bạch thành 4 pha chuẩn mực (Tích lũy - Tăng trưởng - Phân phối - Suy thoái), thuật toán đã nhận diện được pha "Climax" (Cao trào) thay cho pha "Phân phối".

Đây thực chất là một phát hiện giá trị về mặt dữ liệu:

- **Về mặt hành vi:** Pha Phân phối (Distribution) thường có đặc điểm đi ngang (Sideway) khá tương đồng với pha Tích lũy về mặt biến động giá, khiến thuật toán khó phân biệt rạch ròi.
- **Về mặt tín hiệu:** Ngược lại, pha Climax (Cao trào mua/bán) lại sở hữu các đặc trưng cực đoan về khối lượng và biên độ (như đã thấy ở Cụm 2 với Volume gấp 3.5 lần trung bình). Việc mô hình tách được cụm này có ý nghĩa thực tiễn lớn hơn nhiều trong giao dịch, bởi đây là những điểm đảo chiều "nóng" (Hot zones) mang lại cơ hội lợi nhuận cao nhất.

Như vậy, dù không rập khuôn theo lý thuyết, bộ nhãn cụm [**Accumulation, Markup, Markdown, Climax**] vẫn đảm bảo tính bao quát và thậm chí còn nhạy bén hơn trong việc phát hiện các điểm bất thường của thị trường. Đây sẽ là nguồn dữ liệu tiền đề (Antecedents) chất lượng cao cho thuật toán khai phá luật kết hợp ở chương sau.

4.2 Phân cụm với dữ liệu tin tức

Sau khi đã hoàn thành việc phân cụm dữ liệu giá, nhóm chuyển sang bài toán thách thức hơn: phân nhóm dữ liệu văn bản. Mục tiêu là tự động gom hơn 200,000 tiêu đề tin tức thành các chủ đề lớn (Topics) để phục vụ cho việc khai phá luật kết hợp ở chương sau.

4.2.1 Phương pháp và quá trình phân cụm

Thách thức và Quy trình xử lý

Đối với dữ liệu tin tức, nhóm nghiên cứu đối mặt với rào cản lớn về tính phi cấu trúc của ngôn ngữ tự nhiên và hiện tượng "lời nguyền số chiều" (curse of dimensionality). Các thuật toán phân cụm dựa trên khoảng cách như K-Means thường hoạt động kém hiệu quả trong không gian vector quá lớn, nơi khoảng cách Euclid giữa các điểm dữ liệu trở nên xấp xỉ nhau và mất đi ý nghĩa phân loại. Do đó, việc chuyển đổi trực tiếp văn bản thô sang các vector đặc trưng khổng lồ (như Bag-of-Words) sẽ khiến mô hình bị nhiễu và không thể hội tụ chính xác. Để giải quyết triệt để vấn đề này, nhóm đề xuất một quy trình xử lý tuần tự gồm ba bước chuyên sâu.

Bước đầu tiên là chuyển đổi không gian ngữ nghĩa thông qua kỹ thuật Nhúng (Embedding). Nhóm lựa chọn mô hình `all-MiniLM-L6-v2` thuộc thư viện Sentence-Transformers làm nòng cốt. Ưu điểm của mô hình này là khả năng mã hóa tiêu đề bài báo thành các vector 384 chiều mà vẫn bảo toàn được ngữ cảnh và ý nghĩa tương đồng của câu, thay vì chỉ so khớp từ khóa đơn thuần. Đây là nền tảng quan trọng để máy tính có thể "hiểu" được nội dung tin tức trước khi thực hiện các phép toán số học.

Tiếp theo, để khắc phục nhược điểm của K-Means với dữ liệu nhiều chiều, thuật toán PCA (Principal Component Analysis) được áp dụng để giảm chiều dữ liệu. Mục tiêu của bước này là nén các vector 384 chiều xuống một không gian thấp hơn (latent space), giúp loại bỏ các nhiễu tín hiệu (noise) trong khi vẫn giữ lại các thành phần phương sai quan trọng nhất. Cuối cùng, dữ liệu sau khi nén sẽ được đưa vào thuật toán K-Means kết hợp với kỹ thuật Tìm kiếm lưới (Grid Search). Quá trình này cho phép nhóm thử nghiệm và tìm ra bộ tham số tối ưu (số chiều PCA và số lượng cụm K) dựa trên độ tách biệt của các cụm, đảm bảo kết quả phân loại đạt độ tin cậy cao nhất.

Kết quả thực nghiệm Grid Search

Nhóm đã tiến hành thử nghiệm toàn diện với các tổ hợp tham số khác nhau: số chiều PCA dao động trong tập $\{3, 4, 5, 10, 15\}$ và số lượng cụm k từ 4 đến 12. Tiêu chí đánh giá được sử dụng là **Silhouette Score**, phản ánh độ tách biệt và cô đặc của các cụm dữ liệu. Kết quả chi tiết của quá trình thử nghiệm được trình bày tại Bảng 4.2.

Bảng 4.2: Kết quả Grid Search (Silhouette Score) theo số chiều PCA và số cụm K

Số cụm (K)	**Silhouette Score theo số chiều PCA**				
	PCA = 3	**PCA = 4**	**PCA = 5**	**PCA = 10**	**PCA = 15**
4	0.2841	0.2281	0.1941	0.1225	0.0957
5	**0.2883**	0.2532	0.2164	0.1349	0.1054
6	0.2811	0.2581	0.2227	0.1335	0.1016
7	0.2607	0.2453	0.2207	0.1368	0.1112
8	0.2608	0.2351	0.2114	0.1315	0.1060
9	0.2520	0.2237	0.1970	0.1355	0.1101
10	0.2484	0.2125	0.1938	0.1357	0.1159
11	0.2475	0.2113	0.1923	0.1294	0.1191
12	0.2492	0.2105	0.1864	0.1280	0.1176

Quan sát bảng số liệu cho thấy một xu hướng rõ rệt: hiệu suất phân cụm giảm dần khi số chiều PCA tăng lên. Cụ thể, khi giữ nguyên số chiều ở mức cao (10 hoặc 15), chỉ số Silhouette chỉ đạt mức rất thấp ($\approx 0.09 - 0.13$), cho thấy các cụm bị chồng lấn đáng kể do nhiễu. Ngược lại, việc giảm xuống không gian 3 chiều mang lại kết quả vượt trội nhất. Tại cấu hình **PCA = 3** và **Số cụm K = 5**, mô hình đạt chỉ số Silhouette cao nhất là **0.2883**. Do đó, nhóm quyết định lựa chọn bộ tham số này làm cấu hình chính thức cho việc phân loại chủ đề tin tức.

4.2.2 Diễn giải kết quả phân cụm

Sau khi thuật toán K-Means phân chia tập dữ liệu thành 5 nhóm riêng biệt, thách thức tiếp theo là "giải mã" ý nghĩa ngữ nghĩa của từng nhóm để gán các nhãn chủ đề (Topic Labeling) phù hợp. Bằng cách áp dụng kỹ thuật TF-IDF để trích xuất các từ khóa trọng tâm có tần suất xuất hiện cao, kết hợp với việc kiểm tra ngẫu nhiên nội dung các tiêu đề, nhóm nghiên cứu đã định danh được 5 luồng thông tin chính chi phối thị trường tiền mã hóa.

1. Hệ sinh thái và Hạ tầng (Topic_Exchange_DeFi - Cụm 0)
Nhóm chủ đề đầu tiên phản ánh sự vận động nội tại của công nghệ và hệ sinh thái blockchain. Các từ khóa nổi bật như *altcoins, web3, grayscale, predictions* cho thấy sự quan tâm của cộng đồng đối với sự phát triển dài hạn, các bản cập nhật công nghệ và sự cạnh tranh thị phần giữa Bitcoin và các Altcoins. Đây là nhóm tin tức mang tính nền tảng, thường xuất hiện các báo cáo phân tích cơ bản về dự án hoặc xu hướng dòng tiền dịch chuyển giữa các hệ sinh thái. Ví dụ tiêu biểu là các thảo luận về việc liệu Altcoins có "soán ngôi" Bitcoin hay các báo cáo danh mục đầu tư từ các quỹ lớn như Grayscale.

2. Dữ liệu định lượng thị trường (Topic_Market_Stats - Cụm 1)
Khác với các nhận định chủ quan, cụm chủ đề này tập trung hoàn toàn vào các "số liệu cứng" (Hard Data). Sự xuất hiện dày đặc của các từ khóa *etf, outflows, inflows, reserves, volume* chỉ ra rằng đây là nơi tập hợp các thông tin về dòng tiền thực tế. Đặc biệt, các tin tức về dòng vốn vào/ra khỏi các quỹ ETF (Spot ETF Flows) hay sự thay đổi dự trữ Bitcoin trên các sàn giao dịch được gom vào nhóm này. Đây là những tín hiệu khách quan nhất phản ánh sức khỏe cung cầu của thị trường, ví dụ như việc ghi nhận hàng trăm triệu USD rút ròng khỏi quỹ ETF trong một tuần.

3. Tâm lý đám đông và Đầu cơ (Topic_Altcoins_Memes - Cụm 2)
Nếu Cụm 1 đại diện cho sự lý trí của dòng tiền tổ chức, thì Cụm 2 lại phản chiếu sự sôi động và đôi khi là phi lý trí của nhà đầu tư cá nhân. Với các từ khóa như *doge, pepe, meme, pump, frenzy*, nhóm này bao gồm các tin tức liên quan đến Memecoins và các đợt tăng giá đột biến dựa trên hiệu ứng mạng xã hội (FOMO). Đây là nhóm tin tức có tính chất "nhiễu" cao đối với xu hướng dài hạn nhưng lại tác động cực mạnh đến biến động giá ngắn hạn và chỉ số cảm xúc toàn thị trường.

4. Kinh tế Vĩ mô và Chính trị (Topic_Macro_Politics - Cụm 3)
Nhóm chủ đề này bao gồm các yếu tố ngoại sinh (exogenous factors) nằm ngoài biểu đồ kỹ thuật nhưng có sức ảnh hưởng mang tính định hình xu hướng. Các từ khóa *trump, election, sec, regulation, fed* cho thấy sự nhạy cảm của thị trường tiền mã hóa đối với bối cảnh chính trị và pháp lý tại Hoa Kỳ. Các sự kiện như bầu cử Tổng thống, quyết định lãi suất của FED hay các vụ kiện tụng của SEC thường được

phân loại vào nhóm này. Tác động của nhóm tin này thường mang tính cấu trúc và dài hạn hơn so với các tin tức nội bộ ngành.

5. Dự báo và Kỳ vọng tương lai (Topic_Price_Prediction - Cụm 4)
Cuối cùng, thuật toán đã tách biệt thành công nhóm tin tức mang tính chất "Dự báo" khỏi nhóm "Dữ liệu thực tế" (Cụm 1). Với các từ khóa *outlook, predict, target, bull run, crash*, đây là tập hợp các nhận định, đồn đoán và phân tích kỹ thuật từ các chuyên gia hoặc KOLs về hướng đi tương lai của giá. Việc phân tách này rất quan trọng, giúp mô hình phân biệt được đâu là sự kiện đã xảy ra (Fact) và đâu là kỳ vọng (Expectation), từ đó đánh giá mức độ rủi ro của thị trường một cách chính xác hơn.

Tổng kết lại: Việc phân cụm thành công 5 chủ đề trên đóng vai trò then chốt cho giai đoạn khai phá luật kết hợp tiếp theo. Nó cho phép hệ thống không chỉ đánh giá tin tức dựa trên cảm xúc (Tốt/Xấu) mà còn đặt nó vào ngữ cảnh cụ thể. Chẳng hạn, một tin tức tích cực thuộc nhóm *Vĩ mô* (Cụm 3) có thể mang trọng số tín hiệu hoàn toàn khác biệt so với một tin tích cực thuộc nhóm *Memecoins* (Cụm 2), từ đó nâng cao độ chính xác của các quy tắc giao dịch được sinh ra.

Chương 5

Khai phá luật kết hợp

Sau khi đã hoàn tất công đoạn chuẩn bị dữ liệu và phân cụm, chương này tập trung vào việc áp dụng thuật toán Khai phá luật kết hợp để tìm kiếm các mẫu hình giao dịch tiềm năng. Mục tiêu là tìm ra các quy tắc có dạng *"Nếu điều kiện A, B, C xảy ra thì xác suất giá Pump/Dump là bao nhiêu"*.

Quy trình thực hiện được chia thành ba giai đoạn: (1) Định nghĩa biến mục tiêu và rời rạc hóa dữ liệu giá, (2) Khai phá luật trên dữ liệu giá thuần túy, và (3) Tích hợp dữ liệu tin tức để xây dựng hệ thống luật lai (Hybrid Rules).

5.1 Chuẩn bị dữ liệu và Định nghĩa biến mục tiêu

5.1.1 Định nghĩa vùng Mua, Bán và Giữ (Target Labeling)

Trong các bài toán dự báo tài chính truyền thống, mô hình thường được huấn luyện để dự đoán giá đóng cửa của phiên tiếp theo sẽ tăng hay giảm. Tuy nhiên, trong giao dịch thực tế, cách tiếp cận này bộc lộ hạn chế lớn vì nó bỏ qua yếu tố quản trị rủi ro và thời gian nắm giữ vị thế. Một nhà giao dịch không chỉ quan tâm giá có tăng hay không, mà quan trọng hơn là liệu giá có tăng đủ mạnh để chạm mức chốt lời (Take Profit) trước khi chạm mức cắt lỗ (Stop Loss) hay không.

Để giải quyết vấn đề này, nhóm áp dụng phương pháp **Triple-Barrier Method** (Phương pháp ba rào chắn). Đây là kỹ thuật gán nhãn dữ liệu tiên tiến, thiết lập ba giới hạn cho mỗi điểm dữ liệu dựa trên một khung thời gian cố định (Horizon). Đặc biệt, thay vì sử dụng các ngưỡng cố định (ví dụ: $\pm 1\%$), nhóm sử dụng chỉ báo **ATR (Average True Range)** để thiết lập các rào chắn động. Điều này đảm bảo rằng mục tiêu lợi nhuận sẽ tự động nới rộng trong các giai đoạn thị trường biến động mạnh và thu hẹp lại khi thị trường đi ngang, phản ánh đúng bản chất "co giãn" của rủi ro.

Ba rào chắn được thiết lập như sau:

1. **Rào chắn trên (Upper Barrier):** Đóng vai trò là mức Chốt lời kỳ vọng. Được tính bằng $Close_t + 3.0 \times ATR_{14}$.
2. **Rào chắn dưới (Lower Barrier):** Đóng vai trò là mức Cắt lỗ. Được tính bằng $Close_t - 3.0 \times ATR_{14}$.
3. **Rào chắn thời gian (Vertical Barrier):** Giới hạn thời gian nắm giữ vị thế, được thiết lập là 60 phiên (tương ứng 60 phút) kể từ thời điểm vào lệnh.

Cơ chế gán nhãn hoạt động dựa trên nguyên tắc "chạm trước": Nếu giá chạm rào chắn trên trước tiên trong vòng 60 phút, nhãn được gán là **Pump** (Tín hiệu Mua). Ngược lại, nếu giá chạm rào chắn dưới trước, nhãn là **Dump** (Tín hiệu Bán). Trường hợp giá không chạm rào chắn nào sau 60 phút, hoặc biến động quá mạnh chạm cả hai rào chắn (hiện tượng "quét hai đầu" hay Whipsaw), nhãn sẽ được gán là **Sideways**. Việc gán nhãn Sideways cho các trường hợp Whipsaw là một bước xử lý nhiễu quan trọng, giúp loại bỏ các tín hiệu giao dịch rủi ro cao khỏi tập dữ liệu huấn luyện.

Dưới đây là đoạn mã hiện thực hóa logic gán nhãn trên:

```
def prepare_target(df, horizon=60, atr_multiplier=3.0):
    data = df.copy()

    # 1. Nhìn về tương lai (Look-forward) để tìm giá cao nhất/thấp nhất
```

```
    indexer = pd.api.indexers.FixedForwardWindowIndexer(window_size=horizon)
    data['future_high'] = data['high'].rolling(window=indexer).max()
    data['future_low']  = data['low'].rolling(window=indexer).min()

    # 2. Thiết lập Rào chắn động dựa trên ATR (Dynamic Barriers)
    threshold = data['atr_14'] * atr_multiplier
    bull_barrier = data['close'] + threshold
    bear_barrier = data['close'] - threshold

    # 3. Kiểm tra điều kiện chạm rào chắn
    hit_tp = data['future_high'] >= bull_barrier # Chạm chốt lời
    hit_sl = data['future_low']  <= bear_barrier # Chạm cắt lỗ

    # 4. Gán nhãn theo thứ tự ưu tiên
    conditions = [
        (hit_tp & hit_sl), # Chạm cả hai (Whipsaw/Nhiễu) -> Coi là Sideways
        hit_sl,            # Chạm cắt lỗ -> Dump
        hit_tp             # Chạm chốt lời -> Pump
    ]

    choices = ['Sideways', 'Dump', 'Pump']

    data['target_label'] = np.select(conditions, choices, default='Sideways')

    # Loại bỏ các hàng không đủ dữ liệu ATR
    return data.dropna(subset=['target_label'])

# Áp dụng với khung thời gian 60 phút và biên độ 3 ATR
df_association = prepare_target(df_association, horizon=60, atr_multiplier=3.0)
```

5.1.2 Rời rạc hóa dữ liệu (Discretization)

Thuật toán khai phá luật kết hợp (như FP-Growth hay Apriori) được thiết kế để làm việc trên dữ liệu dạng giao dịch (transactional data), nơi mỗi bản ghi là tập hợp các "mặt hàng" (items) rời rạc. Trong khi đó, dữ liệu tài chính thu được từ các chương trước phần lớn là các biến số thực liên tục (Continuous variables). Do đó, để chuyển đổi dữ liệu này thành định dạng phù hợp cho mô hình, nhóm nghiên cứu thực hiện quá trình rời rạc hóa bằng cách phân chia dữ liệu vào các khoảng giá trị (bins) dựa trên các ngưỡng kỹ thuật tiêu chuẩn và đặc điểm phân phối thống kê.

Quy trình rời rạc hóa được áp dụng chi tiết cho từng nhóm đặc trưng như sau:

1. Chỉ báo động lượng (RSI): Thay vì chỉ sử dụng ngưỡng quá mua/quá bán kinh điển (70/30), nhóm chia nhỏ không gian giá trị thành 5 trạng thái để nắm bắt kỹ hơn các biến động trung gian. Cụ thể, vùng từ 45 đến 55 được tách riêng thành "RSI_Neutral" để lọc nhiễu khi thị trường không có xu hướng rõ ràng. Các giá trị trên 70 được gán nhãn "Overbought" (Quá mua) và dưới 30 là "Oversold" (Quá bán).

2. Chế độ khối lượng (Volume Regime): Được chia dựa trên bội số so với trung bình động 20 phiên. Mức "Normal" nằm trong khoảng 0.5 đến 1.5 lần trung bình. Đáng chú ý là nhãn "Vol_Shock" (trên 3.0 lần), đại diện cho các sự kiện đột biến thanh khoản (Climax) thường báo hiệu sự đảo chiều.

3. Vị thế xu hướng (Trend Tactical): Đo lường độ lệch của giá so với EMA50. Các trạng thái "Overextended" (Kéo dãn quá mức) được định nghĩa khi độ lệch vượt quá $\pm 0.6\%$, đây là vùng giá thường có xu hướng hồi quy về trung bình (Mean Reversion).

4. Điểm cấu trúc và Pha thị trường: Điểm cấu trúc (Structure Score) được đơn giản hóa thành 3 trạng thái: Bull (Bò), Bear (Gấu) và Neutral (Trung tính). Riêng đối với kết quả phân cụm từ Chương 4, các nhãn số (0, 1, 2, 3) được ánh xạ trực tiếp sang các tên gọi theo lý thuyết Wyckoff để tăng tính gợi nhớ (Markup, Accumulation, Climax, Markdown).

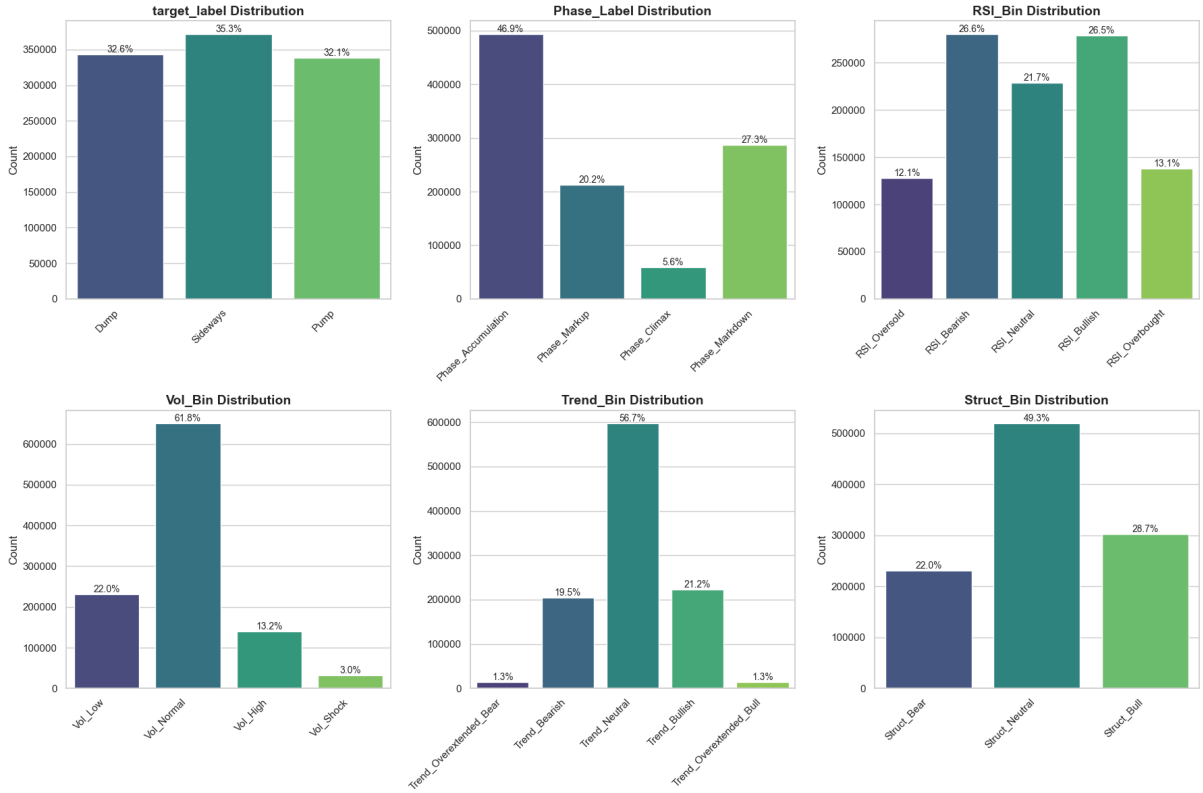
Đoạn mã dưới đây hiện thực hóa toàn bộ quy trình chuyển đổi từ dữ liệu số sang dữ liệu định danh (string labels):

```
def discretize_features(df):
    data = df.copy()
```

```
    # 1. Rời rạc hóa RSI (5 bins)
    # Thêm vùng đệm 45-55 để xác định trạng thái trung tính chính xác hơn
    data['RSI_Bin'] = pd.cut(data['rsi_14'],
                             bins=[-np.inf, 30, 45, 55, 70, np.inf],
                             labels=['RSI_Oversold', 'RSI_Bearish', 'RSI_Neutral',
                                     'RSI_Bullish', 'RSI_Overbought'])

    # 2. Rời rạc hóa Khối lượng (4 bins)
    # Tập trung phát hiện các điểm Shock Volume (> 3 lần trung bình)
    data['Vol_Bin'] = pd.cut(data['vol_regime'],
                             bins=[-np.inf, 0.5, 1.5, 3.0, np.inf],
                             labels=['Vol_Low', 'Vol_Normal', 'Vol_High', 'Vol_Shock'])

    # 3. Rời rạc hóa Vị thế xu hướng (5 bins)
    # Xác định khi giá đi quá xa đường trung bình (Overextended)
    data['Trend_Bin'] = pd.cut(data['trend_tactical'],
                               bins=[-np.inf, -0.6, -0.1, 0.1, 0.6, np.inf],
                               labels=['Trend_Overextended_Bear', 'Trend_Bearish',
                                       'Trend_Neutral', 'Trend_Bullish',
                                       'Trend_Overextended_Bull'])

    # 4. Rời rạc hóa Điểm cấu trúc (3 bins)
    data['Struct_Bin'] = pd.cut(data['structure_score'],
                                bins=[-np.inf, -0.5, 0.5, np.inf],
                                labels=['Struct_Bear', 'Struct_Neutral', 'Struct_Bull'])

    # 5. Ánh xạ Nhãn phân cụm (Mapping Cluster IDs)
    cluster_map = {
        0: 'Phase_Markup',
        1: 'Phase_Accumulation',
        2: 'Phase_Climax',
        3: 'Phase_Markdown'
    }
    data['Phase_Label'] = data['cluster'].map(cluster_map)

    # Lọc lấy các cột đã xử lý
    cols_to_keep = ['Phase_Label', 'RSI_Bin', 'Vol_Bin',
                    'Trend_Bin', 'Struct_Bin', 'target_label']

    return data[cols_to_keep].dropna().astype(str)

# Thực thi và xem trước kết quả
transactions = discretize_features(df_association)
```

Bảng 5.1: Tổng hợp quy tắc rời rạc hóa và gán nhãn đặc trưng

Đặc trưng (Feature)	**Ngưỡng phân chia (Thresholds)**	**Nhãn (Labels)**
RSI (14)	$< 30, 30\text{-}45, 45\text{-}55, 55\text{-}70, > 70$	Oversold, Bearish, Neutral, Bullish, Overbought
Volume Regime	$< 0.5, 0.5\text{-}1.5, 1.5\text{-}3.0, > 3.0$	Low, Normal, High, Shock
Trend Tactical	$< -0.6, -0.6\text{-} -0.1, \ldots, > 0.6$	Overextended_Bear, ..., Overextended_Bull
Structure Score	$< -0.5, -0.5\text{-}0.5, > 0.5$	Struct_Bear, Struct_Neutral, Struct_Bull
Cluster ID	Ánh xạ trực tiếp từ K-Means	Markup, Accumulation, Climax, Markdown



Hình 5.1: Phân bố tần suất các đặc trưng và nhãn mục tiêu trong tập dữ liệu giao dịch sau khi rời rạc hóa.

Sau bước này, dữ liệu giá đã được chuyển đổi hoàn toàn sang dạng tập giao dịch (Transaction Dataset), sẵn sàng để đưa vào thuật toán khai phá.

5.2 Khai phá luật trên dữ liệu giá

Trước khi tích hợp các yếu tố ngoại sinh từ tin tức, nhóm nghiên cứu tiến hành khai phá trên tập dữ liệu thuần kỹ thuật để thiết lập một mô hình cơ sở. Mục tiêu của bước này là trả lời câu hỏi: *"Liệu các chỉ báo kỹ thuật và cấu trúc thị trường đơn thuần có thể dự báo được các đợt biến động giá mạnh hay không?"*.

5.2.1 Thiết lập thuật toán và Tham số

Để xử lý lượng dữ liệu giao dịch lớn (hơn 1 triệu bản ghi), nhóm lựa chọn thuật toán **FP-Growth** (Frequent Pattern Growth). So với Apriori, FP-Growth hiệu quả hơn đáng kể về mặt bộ nhớ và tốc độ do nó nén dữ liệu vào cấu trúc cây FP-Tree và không cần sinh các tập ứng viên (candidate generation) lặp đi lặp lại.

Quy trình thực hiện bắt đầu bằng việc chuyển đổi dữ liệu dạng danh sách sang ma trận nhị phân (One-hot encoding) sử dụng `TransactionEncoder`. Sau đó, thuật toán được cấu hình với bộ tham số chiến lược như sau:

- **Độ hỗ trợ tối thiểu (Min Support = 0.001):** Nhóm quyết định hạ ngưỡng này xuống mức rất thấp (0.1%). Lý do là các cơ hội giao dịch lợi nhuận cao (Pump/Dump mạnh) thường là các sự kiện hiếm (Rare Events). Việc đặt ngưỡng quá cao (ví dụ 5%) sẽ vô tình loại bỏ các mẫu hình "Thiên nga đen" giá trị này.
- **Độ tin cậy tối thiểu (Min Confidence = 0.1):** Ngưỡng khởi điểm thấp để bao quát không gian tìm kiếm, sau đó sẽ lọc lại ở bước hậu xử lý.
- **Chiến lược lọc vùng Mua/Bán:** Để trích xuất các tín hiệu có giá trị thực chiến, nhóm áp dụng bộ lọc khắt khe hơn trên kết quả đầu ra:

- **Buy Zones (Pump):** Yêu cầu $Lift > 1.1$ và $Confidence > 0.35$.
- **Sell Zones (Dump):** Yêu cầu $Lift > 1.1$ và $Confidence > 0.40$ (Do tâm lý hoảng loạn thường dễ đoán hơn hưng phấn, nên ngưỡng tin cậy cho lệnh bán được đặt cao hơn).

Đoạn mã dưới đây minh họa quá trình huấn luyện và lọc luật trên tập dữ liệu giá:

```
# 1. Mã hóa dữ liệu giao dịch (One-hot Encoding)
if isinstance(transactions, pd.DataFrame):
    transactions = transactions.values.tolist()

te = TransactionEncoder()
te_ary = te.fit(transactions).transform(transactions)
df_encoded = pd.DataFrame(te_ary, columns=te.columns_)

# 2. Chạy thuật toán FP-Growth
# min_support thấp để bắt các mẫu hình hiếm (Rare patterns)
frequent_itemsets = fpgrowth(df_encoded, min_support=0.001, use_colnames=True)

# 3. Sinh luật kết hợp
rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.1)

# 4. Lọc luật theo chiến lược giao dịch
# Chiến lược Mua (Pump)
buy_zones = rules[
    (rules['consequents'] == frozenset({'Pump'})) &
    (rules['lift'] > 1.1) &
    (rules['confidence'] > 0.35)
].sort_values(by='confidence', ascending=False)

# Chiến lược Bán (Dump)
sell_zones = rules[
    (rules['consequents'] == frozenset({'Dump'})) &
    (rules['lift'] > 1.1) &
    (rules['confidence'] > 0.4)
].sort_values(by='confidence', ascending=False)
```

5.2.2 Kết quả thực nghiệm và Đánh giá

Kết quả khai phá trên dữ liệu giá Bitcoin cho thấy một lượng lớn các mẫu hình lặp lại, khẳng định tính hiệu quả của phân tích kỹ thuật. Bảng số liệu thống kê tổng hợp dưới đây tóm tắt hiệu năng của mô hình:

Bảng 5.2: Thống kê kết quả khai phá luật trên dữ liệu giá

Chỉ số (Metric)	**Giá trị (Value)**
Tổng số luật tìm thấy (Total Rules)	36,357
Độ nâng trung bình (Avg Lift)	2.389
Số lượng luật Mua (Pump Rules)	**2,949**
Max Lift (Pump)	50.55
Max Confidence (Pump)	36.39%
Số lượng luật Bán (Dump Rules)	**3,969**
Max Lift (Dump)	54.81
Max Confidence (Dump)	58.87%

Phân tích chuyên sâu:

1. **Sự vượt trội của phe Gấu (Bearish Bias):** Số lượng luật báo hiệu giảm giá (Dump: 3,969 luật) nhiều hơn đáng kể so với luật tăng giá (Pump: 2,949 luật). Đồng thời, độ tin cậy tối đa của

luật Dump lên tới **58.87%**, cao hơn hẳn so với mức 36.39% của luật Pump. Điều này phù hợp với đặc tính "Lên thang bộ, xuống thang máy" của Bitcoin: các đợt sụt giảm thường diễn ra nhanh, mạnh và tuân theo các cấu trúc kỹ thuật rõ ràng hơn (panic selling) so với các đợt tăng giá.

2. **Sức mạnh dự báo cực đại (Extreme Lift):** Mặc dù độ tin cậy trung bình chỉ ở mức khiêm tốn (29%), nhưng chỉ số **Max Lift** đạt tới con số ấn tượng: **50.55** cho chiều mua và **54.81** cho chiều bán. Điều này hàm ý rằng: Khi các tổ hợp điều kiện kỹ thuật đặc biệt này xuất hiện, xác suất giá biến động mạnh cao gấp 50 lần so với xác suất ngẫu nhiên. Đây chính là các "cơ hội vàng" (Golden Opportunities) mà hệ thống hướng tới, dù tần suất xuất hiện của chúng có thể không cao.

3. **Hạn chế của mô hình:** Độ tin cậy tối đa cho chiều Mua chỉ đạt **36.39%**. Trong giao dịch, con số này tuy có thể sinh lời (nếu tỷ lệ Rủi ro/Lợi nhuận tốt) nhưng vẫn tiềm ẩn nhiều tín hiệu giả (False Positives). Điều này đặt ra nhu cầu cấp thiết phải tích hợp thêm dữ liệu tin tức ở phần sau để lọc nhiễu và nâng cao độ chính xác cho các quyết định mua.

5.3 Khai phá luật lai (Hybrid Mining)

Sau khi đã thiết lập được mô hình cơ sở từ dữ liệu giá, bước tiến quan trọng tiếp theo là tích hợp luồng thông tin từ tin tức vào mô hình. Mục tiêu của phần này không chỉ là tìm ra các quy luật mới, mà là tìm ra các *ngữ cảnh thông tin* (Information Context) giúp giải thích tại sao giá lại biến động, từ đó nâng cao độ tin cậy của các quyết định giao dịch.

5.3.1 Chuẩn bị và Tích hợp dữ liệu đa nguồn

Thách thức lớn nhất trong việc kết hợp dữ liệu giá và tin tức nằm ở sự bất đồng bộ về thời gian (Asynchronous frequencies). Dữ liệu giá có tính liên tục và tần suất cao (1 phút/lần), trong khi tin tức xuất hiện ngẫu nhiên và rời rạc. Để giải quyết vấn đề này, nhóm xây dựng quy trình tích hợp gồm ba bước: Tổng hợp (Aggregation), Đồng bộ hóa (Synchronization) và Hợp nhất (Merging).

Bước 1: Tổng hợp và Rời rạc hóa tin tức theo giờ Thay vì xử lý từng bản tin riêng lẻ, nhóm tiến hành tổng hợp các chỉ số Z-Score của Cảm xúc (Sentiment) và Mức độ quan tâm (Volume Buzz) theo khung thời gian 1 giờ. Sau đó, các giá trị này được chuyển đổi sang dạng nhãn (labels) như *Sent_ Positive, Buzz_ High*. Đặc biệt, nhãn chủ đề (Topic) phổ biến nhất trong giờ đó cũng được trích xuất.

Bước 2: Xử lý độ trễ thông tin (Look-ahead Bias) Đây là bước quan trọng để đảm bảo tính thực tế. Dữ liệu tin tức được dịch chuyển (shift) lùi lại 1 giờ. Điều này mô phỏng đúng thực tế: tại thời điểm T, nhà đầu tư chỉ biết các tin tức đã xảy ra từ $T-1$ trở về trước.

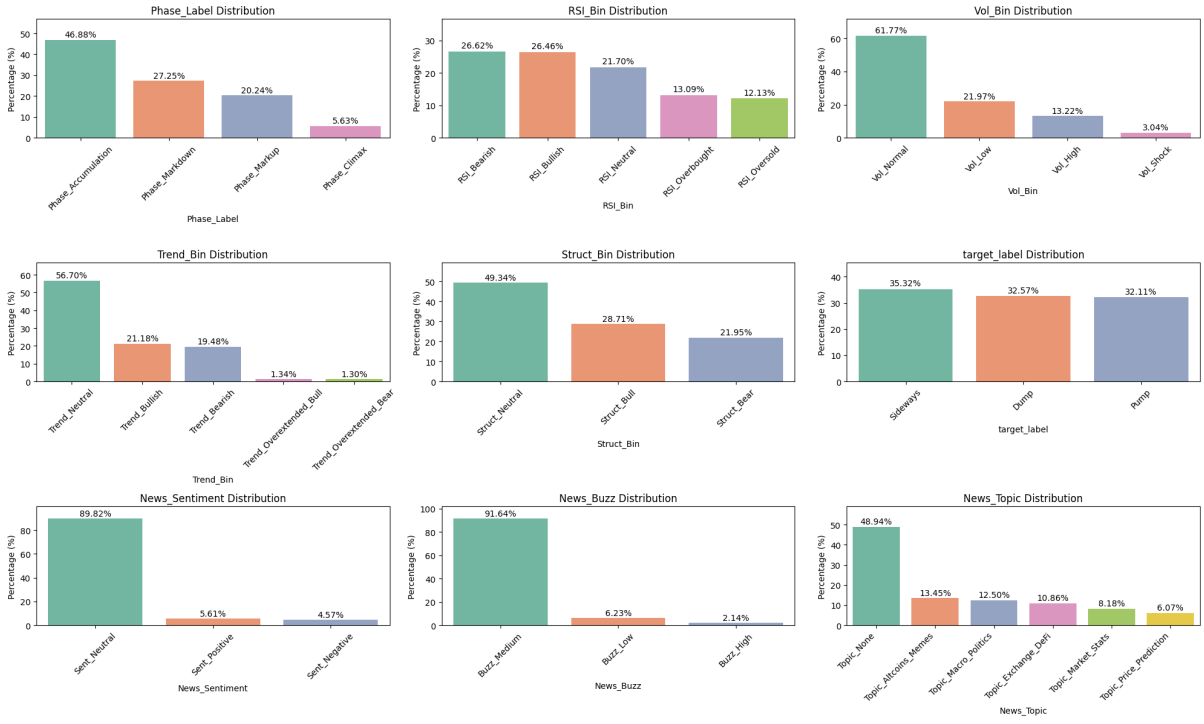
Bước 3: Hợp nhất dữ liệu (Merging) Nhóm sử dụng kỹ thuật `merge_asof` với hướng `backward`. Kỹ thuật này cho phép mỗi bản ghi giá tại thời điểm t tìm kiếm và kết hợp với bản ghi tin tức gần nhất trong quá khứ (trong phạm vi dung sai 2 giờ). Nếu không có tin tức nào trong khoảng thời gian này, các trạng thái mặc định (*Neutral, Medium, None*) sẽ được gán để lấp đầy dữ liệu.

Đoạn mã dưới đây mô tả chi tiết logic tích hợp này:

```
def integrate_data(df_price, df_news):
    # 1. Tổng hợp tin tức theo giờ (Hourly Aggregation)
    hourly_news = df_news.resample('1h').agg({
        'Sentiment_Z_Score': 'mean',
        'Volume_Z_Score': 'mean',
        'Topic_Label': lambda x: x.mode()[0] if not x.mode().empty else "Topic_None"
    })

    # 2. Xử lý Look-ahead Bias (Dịch chuyển 1 giờ)
    hourly_news = hourly_news.shift(1)

    # 3. Rời rạc hóa các chỉ số tin tức (Re-binning)
    # Sentiment > 0.5 -> Positive, < -0.5 -> Negative
    hourly_news['News_Sentiment'] = np.select(
        [hourly_news['Sentiment_Z_Score'] > 0.5, hourly_news['Sentiment_Z_Score'] < -0.5],
        ['Sent_Positive', 'Sent_Negative'], default='Sent_Neutral'
    )
```

```
# Buzz > 2.0 -> High (Tin nóng)
hourly_news['News_Buzz'] = np.select(
    [hourly_news['Volume_Z_Score'] > 2, hourly_news['Volume_Z_Score'] < -0.5],
    ['Buzz_High', 'Buzz_Low'], default='Buzz_Medium'
)

# 4. Hợp nhất với dữ liệu giá (Merge Asof)
merged_df = pd.merge_asof(
    df_price.sort_index(),
    hourly_news[['News_Sentiment', 'News_Buzz', 'Topic_Label']],
    left_index=True, right_index=True,
    direction='backward', tolerance=pd.Timedelta('2h')
)

# Gán giá trị mặc định cho các khoảng trống tin tức
merged_df.fillna({'News_Sentiment': 'Sent_Neutral',
                  'News_Buzz': 'Buzz_Medium',
                  'Topic_Label': 'Topic_None'}, inplace=True)

return merged_df
```

Kết quả của quá trình này là một bộ cơ sở dữ liệu giao dịch thống nhất, nơi mỗi biến động giá đều được đặt trong bối cảnh tin tức tương ứng.



Hình 5.2: Phân bố tần suất các đặc trưng trong tập dữ liệu tích hợp

5.3.2 Khai phá mẫu kết hợp lai

Trên tập dữ liệu đã tích hợp, nhóm tiếp tục áp dụng thuật toán FP-Growth. Tuy nhiên, để tối ưu hóa hiệu suất và tập trung vào các tín hiệu có giá trị, nhóm áp dụng chiến lược **"Lọc nhiễu chủ động"**:

- **Loại bỏ Stop-words dữ liệu:** Các giá trị mặc định như *Sent_ Neutral, Buzz_ Medium, Topic_ None* chiếm tỷ trọng lớn nhưng mang ít thông tin dự báo. Việc loại bỏ chúng khỏi tập giao dịch giúp thuật toán tập trung tìm kiếm các mối liên kết khi có sự kiện tin tức thực sự (Tích cực/Tiêu cực hoặc Tin nóng).

- **Giới hạn độ dài luật (Max Length = 6):** Để tránh sinh ra các luật quá phức tạp và khó giải thích (Overfitting), độ dài tối đa của tập phổ biến được giới hạn ở mức 6 phần tử.

Quá trình khai phá và sinh luật được thực hiện như sau:

```
def hybrid_mining(df_merged):
    # 1. Lọc nhiễu chủ động (Active Noise Filtering)
    ignore_items = ['Sent_Neutral', 'Buzz_Medium', "Topic_None", "nan", "None"]
    transactions = [
        [item for item in row if item not in ignore_items]
        for row in df_merged.values.tolist()
    ]
    transactions = [t for t in transactions if len(t) > 0]

    # 2. Chạy FP-Growth với giới hạn độ dài
    # Min Support = 0.5%
    te = TransactionEncoder()
    df_encoded = pd.DataFrame(te.fit(transactions).transform(transactions), columns=te.columns_)
    frequent_itemsets = fpgrowth(df_encoded, min_support=0.005, max_len=6, use_colnames=True)

    # 3. Sinh luật và Lọc chiến lược Hybrid
    rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.1)

    # Chỉ lấy các luật Mua có chứa yếu tố Tin tức ở vế trái
    buy_mask = (
        (rules['consequents'] == frozenset({'Pump'})) &
        (rules['lift'] > 1.1) &
        (rules['antecedents'].astype(str).str.contains('Sent_|Buzz_|Topic_'))
    )
    hybrid_buy = rules[buy_mask].sort_values(by='lift', ascending=False)

    return hybrid_buy
```

Kết quả thực nghiệm: Sau quá trình lọc nhiễu và khai phá, hệ thống thu được tổng cộng **24,188** luật kết hợp. Trong đó:

- **Số lượng luật Mua (Pump):** 1,821 luật. Max Lift đạt **17.15** và Max Confidence đạt **37.88%**.
- **Số lượng luật Bán (Dump):** 1,882 luật. Max Lift đạt **17.26** và Max Confidence đạt **39.69%**.

Mặc dù chỉ số Lift cực đại thấp hơn so với mô hình thuần giá (do điều kiện kết hợp khắt khe hơn), nhưng độ tin cậy trung bình (Avg Confidence) lại có sự cải thiện nhẹ. Quan trọng hơn, các luật sinh ra đều mang tính giải thích cao (Explainable AI), liên kết trực tiếp hành động giá với các sự kiện thực tế.

Chương 6

Thử nghiệm và Đánh giá

Mục tiêu của chương này là mô phỏng lại quá trình giao dịch thực tế trên tập dữ liệu kiểm thử (Out-of-Sample) từ 01/01/2025 đến 06/12/2025. Để đảm bảo tính khách quan và tránh hiện tượng quá khớp (overfitting), toàn bộ quy trình từ xử lý dữ liệu đến ra quyết định đều phải tuân thủ nghiêm ngặt nguyên tắc: "Tại thời điểm T, hệ thống chỉ được sử dụng thông tin có sẵn từ T trở về trước".

6.1 Thiết lập dòng dữ liệu giá kiểm thử

6.1.1 Trích xuất đặc trưng dữ liệu

Quy trình tạo đặc trưng trên tập kiểm thử (Test Set) được thực hiện hoàn toàn tương tự như trên tập huấn luyện (Train Set) để đảm bảo sự nhất quán cho mô hình. Các nhóm chỉ báo bao gồm: Chỉ báo kỹ thuật cơ bản (RSI, EMA), Các đặc trưng Tiền thông minh (Order Blocks, FVG) và Các đặc trưng cấu trúc Wyckoff.

Tuy nhiên, một điểm khác biệt cốt lõi nằm ở bước *Chuẩn hóa dữ liệu (Data Scaling)* trước khi đưa vào mô hình phân cụm K-Means. Thay vì tính toán trung bình (μ) và độ lệch chuẩn (σ) trực tiếp trên tập Test, nhóm sử dụng lại các tham số thống kê đã học được từ tập Train (giai đoạn 2023-2024).

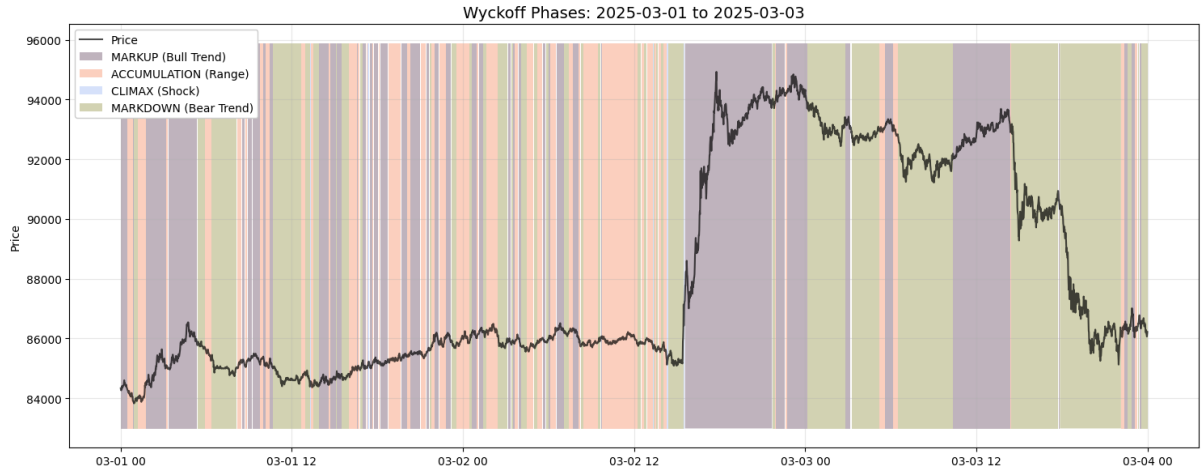
$$X_{test\_scaled} = \frac{X_{test} - \mu_{train}}{\sigma_{train}} \tag{6.1}$$

Vì sao ta phải làm như vậy? Nếu chúng ta chuẩn hóa lại tập Test bằng μ_{test}, đồng nghĩa với việc chúng ta đã "nhìn trộm" tương lai (biết trước giá trung bình của năm 2025). Việc tái sử dụng μ_{train} đảm bảo tính công bằng, mô phỏng đúng bối cảnh thực tế khi hệ thống vận hành (Live Trading). Mặc dù các đặc trưng thị trường có thể thay đổi theo thời gian (Concept Drift), nhưng do các biến đầu vào của chúng ta (như độ lệch EMA, RSI, Efficiency) đều là các đại lượng tương đối và có tính dừng (Stationary), nên việc sử dụng tham số từ quá khứ vẫn đảm bảo độ chính xác cao.

Dưới đây là đoạn mã thực hiện quy trình tính toán và chuẩn hóa:

```
# --- 1. TÍNH TOÁN CHỈ BÁO CƠ BẢN ---
def calculate_basic_indicators(df):
    data = df.copy()
    # Tính EMA 50 và 200
    data['ema_50'] = data['close'].ewm(span=50, adjust=False).mean()
    data['ema_200'] = data['close'].ewm(span=200, adjust=False).mean()

    # Tính RSI 14
    delta = data['close'].diff()
    gain = (delta.where(delta > 0, 0)).ewm(alpha=1/14, min_periods=14).mean()
    loss = (-delta.where(delta < 0, 0)).ewm(alpha=1/14, min_periods=14).mean()
    rs = gain / loss
    data['rsi_14'] = 100 - (100 / (1 + rs))

    # Tính Volume USD và Volume MA20
    if 'volume_usd' not in data.columns:
```

```
        data['volume_usd'] = data['close'] * data['volume']
    data['vol_ma20'] = data['volume_usd'].rolling(window=20).mean()
    return data

# --- 2. TÍNH TOÁN SMART MONEY CONCEPTS (SMC) ---
def calculate_smart_money_features(df):
    # (Đoạn mã tương tự chương trước, lược bỏ để tránh lặp lại)
    # Bao gồm: Order Blocks, FVG, Shock Volume
    ...
    return df_smc

# --- 3. TÍNH TOÁN ĐẶC TRƯNG WYCKOFF ---
def create_wyckoff_features(df):
    df_w = df.copy()
    # Trend Maturity & Tactical
    df_w['trend_maturity'] = (df_w['ema_50'] - df_w['ema_200']) / df_w['close'] * 100
    df_w['trend_tactical'] = (df_w['close'] - df_w['ema_50']) / df_w['ema_50'] * 100

    # Efficiency Regime
    epsilon = 1e-9
    range_len = df_w['high'] - df_w['low']
    body_len = abs(df_w['close'] - df_w['open'])
    df_w['efficiency_regime'] = (body_len / (range_len + epsilon)).rolling(window=20).mean()

    # Structure Score (Tích lũy điểm số cấu trúc)
    # ... (Logic cộng dồn điểm số theo OB/FVG)

    return df_w
```



Hình 6.1: Trực quan hoá kết quả của mô hình phân cụm (kmeans) với tập dữ liệu thử nghiệm (2025/03/01 đến 2025/03/03)

6.1.2 Dự báo trạng thái Wyckoff và Tạo tập giao dịch

Sau khi đã có đầy đủ các đặc trưng thô, bước tiếp theo là chuyển đổi chúng thành định dạng giao dịch (Transactional Database) để có thể khớp lệnh với các luật kết hợp đã tìm được. Quy trình này bao gồm 6 bước tuần tự:

1. **Tải dữ liệu:** Đọc dữ liệu Train (để lấy tham số chuẩn hóa) và dữ liệu Test.

2. **Tạo đặc trưng thô:** Áp dụng các hàm tính toán chỉ báo cho cả hai tập dữ liệu.

3. **Đồng bộ hóa dữ liệu:** Loại bỏ các hàng có giá trị NaN (do cửa sổ trượt ban đầu) để đảm bảo hai tập dữ liệu tương thích về cấu trúc.
4. **Huấn luyện bộ chuẩn hóa (Fitting Scaler):** Khởi tạo đối tượng `StandardScaler` và gọi hàm `.fit()` CHỈ trên tập Train. Điều này giúp bộ chuẩn hóa "học" được phân phối (Mean, Std) của quá khứ.
5. **Dự báo Pha thị trường (Predicting Clusters):** Sử dụng mô hình K-Means (đã lưu từ giai đoạn huấn luyện) để dự báo nhãn cụm (Cluster ID) cho tập Test dựa trên dữ liệu đã được chuẩn hóa.
6. **Rời rạc hóa (Discretization):** Chuyển đổi các giá trị số liên tục và nhãn cụm thành các nhãn văn bản (ví dụ: *RSI_ Oversold, Phase_ Markup*) để sẵn sàng cho việc tra cứu luật.

Đoạn mã thực thi quy trình này như sau:

```
def generate_test_transactions(train_path, test_path, model_path):
    print("1. Loading Data...")
    df_train = pd.read_csv(train_path)
    df_test = pd.read_csv(test_path)

    # Chuyển đổi định dạng thời gian
    df_train['open_time'] = pd.to_datetime(df_train['open_time'])
    df_test['open_time'] = pd.to_datetime(df_test['open_time'])

    print("2. Generating Raw Features...")
    # Tạo đặc trưng cho cả Train và Test
    df_train = calculate_smart_money_features(calculate_basic_indicators(df_train))
    df_train = create_wyckoff_features(df_train)

    df_test = calculate_smart_money_features(calculate_basic_indicators(df_test))
    df_test = create_wyckoff_features(df_test)

    # Xác định các cột đặc trưng đầu vào cho mô hình K-Means
    feature_cols = ["trend_tactical", "trend_maturity", "efficiency_regime",
                    "vol_regime", "structure_score"]

    df_train = df_train.dropna(subset=feature_cols)
    df_test = df_test.dropna(subset=feature_cols)

    print("3. Fitting Scaler on Training Data...")
    # QUAN TRỌNG: Chỉ học Mean/Std từ tập Train
    scaler = StandardScaler()
    scaler.fit(df_train[feature_cols])

    print("4. Scaling Test Data...")
    # Áp dụng tham số của Train lên Test
    X_test_scaled = scaler.transform(df_test[feature_cols])

    print("5. Loading Model and Predicting...")
    # Tải mô hình K-Means đã huấn luyện
    kmeans = joblib.load(model_path)
    df_test['cluster'] = kmeans.predict(X_test_scaled)

    # Ánh xạ ID cụm sang tên Pha Wyckoff
    cluster_map = {0: 'Phase_Markup', 1: 'Phase_Accumulation',
                   2: 'Phase_Climax', 3: 'Phase_Markdown'}

    print("6. Discretizing and Returning...")
    return discretize_features(df_test, cluster_map)
```

Kết quả trả về là một DataFrame chứa các giao dịch của năm 2025, với đầy đủ các nhãn trạng thái (Phase, RSI, Volume...) đã được đồng bộ hóa với hệ tri thức của mô hình.

6.2 Thiết lập dòng dữ liệu tin tức kiểm thử

Song song với việc tái tạo dữ liệu giá, dòng dữ liệu tin tức cho năm 2025 cũng cần được xử lý và chuẩn hóa để trích xuất các tín hiệu đầu vào cho mô hình. Quy trình này bao gồm hai công đoạn chính: xử lý các chỉ số phản ứng định lượng (Quantitative Reactions) và phân loại chủ đề dựa trên nội dung văn bản (Topic Classification).

6.2.1 Xử lý văn bản và Trích xuất đặc trưng phản ứng

Quy trình xử lý bắt đầu bằng việc làm sạch văn bản và lọc dữ liệu để chỉ giữ lại các tin tức liên quan trực tiếp đến Bitcoin trong giai đoạn kiểm thử (từ 01/01/2025 trở đi). Sau đó, các chỉ số tương tác thô (like, share, comment...) được tổng hợp và chuẩn hóa bằng kỹ thuật Z-Score trượt (Rolling Z-Score) với cửa sổ 24 giờ.

Điều quan trọng cần nhấn mạnh là việc sử dụng Rolling Window thay vì Global Mean giúp mô hình thích nghi với sự thay đổi trong hành vi người dùng theo thời gian. Một tin tức có 100 like trong giai đoạn ảm đạm (nền thấp) sẽ có Z-Score cao hơn nhiều so với tin tức tương tự trong giai đoạn sôi động (nền cao), qua đó phản ánh đúng mức độ "bất thường" của sự kiện.

Cuối cùng, các giá trị Z-Score liên tục được rời rạc hóa thành các nhãn định tính (*Sent_Positive, Buzz_High...*) để tương thích với bộ luật kết hợp.

```
import re
import html

# 1. Hàm làm sạch văn bản (Text Cleaning)
def clean_text(text):
    if not isinstance(text, str): return ""
    text = html.unescape(text)
    text = re.sub(r'http\S+|www\S+|https\S+', '', text, flags=re.MULTILINE) # Xóa URL
    text = re.sub(r'@\w+|#\w+', '', text)          # Xóa handle/hashtag
    text = re.sub(r'[^\x00-\x7F]+', ' ', text)     # Xóa ký tự lạ
    text = re.sub(r'\s+', ' ', text).strip()
    return text.lower() # Chuyển về chữ thường để khớp từ khóa

# 2. Pipeline xử lý và tạo đặc trưng phản ứng
def process_news_reaction_features(news_path):
    print("1. Loading & Filtering News Data (2025)...")
    df = pd.read_csv(news_path)
    df = df[df['currencies'].str.contains('BTC', na=False)] # Lọc tin BTC

    # Chuyển đổi thời gian và lọc dữ liệu năm 2025
    if 'newsDatetime' in df.columns:
        df['newsDatetime'] = pd.to_datetime(df['newsDatetime'])
        df = df.set_index('newsDatetime').sort_index()
    df = df[df.index >= '2025-01-01'].copy()

    # Làm sạch tiêu đề
    df['clean_title'] = df['title'].apply(clean_text)

    print("2. Aggregating Reactions...")
    # Tổng hợp các cột tương tác (Positive vs Negative)
    pos_cols = ['positive', 'liked', 'important', 'saved']
    neg_cols = ['negative', 'disliked', 'toxic', 'lol']
    # (Điền 0 nếu cột thiếu)
    for c in pos_cols + neg_cols + ['comments']:
        if c not in df.columns: df[c] = 0
```

```
    df['Reaction_Positive'] = df[pos_cols].sum(axis=1)
    df['Reaction_Negative'] = df[neg_cols].sum(axis=1)
    df['Total_Reactions'] = df['Reaction_Positive'] + df['Reaction_Negative'] + df['comments']

    print("3. Calculating Rolling Z-Scores (24h Window)...")
    window = '24h'
    # Tính Z-Score trượt để tránh look-ahead bias
    # (Code chi tiết tính mean/std và công thức Z-score như đã trình bày ở chương trước)
    # ...

    print("4. Discretizing to Match Rules...")
    # Rời rạc hóa Cảm xúc (Sentiment)
    df['News_Sentiment'] = pd.cut(df['Sentiment_Z_Score'],
                                  bins=[-np.inf, -0.5, 0.5, np.inf],
                                  labels=['Sent_Negative', 'Sent_Neutral', 'Sent_Positive'])

    # Rời rạc hóa Mức độ quan tâm (Buzz)
    df['News_Buzz'] = pd.cut(df['Volume_Z_Score'],
                             bins=[-np.inf, -0.5, 2, np.inf],
                             labels=['Buzz_Low', 'Buzz_Medium', 'Buzz_High'])

    return df.reset_index()
```

6.2.2 Phân loại chủ đề dựa trên Từ khóa

Đối với nhiệm vụ phân loại chủ đề cho tập kiểm thử, nhóm áp dụng một phương pháp tiếp cận khác biệt và hiệu quả hơn so với giai đoạn huấn luyện. Thay vì chạy lại mô hình SBERT và K-Means (vốn tốn kém tài nguyên và khó kiểm soát nhãn cụm mới sinh ra), nhóm sử dụng phương pháp **Túi từ (Bag-of-Words)** dựa trên bộ từ điển từ khóa đã được trích xuất từ tập Train.

Quy trình cụ thể như sau:

1. **Xây dựng từ điển chủ đề:** Từ kết quả phân cụm ở Chương 4, nhóm đã lưu lại danh sách 1000 từ khóa phổ biến nhất (Top Keywords) đại diện cho mỗi chủ đề (ví dụ: *"sec", "etf"* cho Topic_Macro_Politics) vào file `topic_keywords.json`.

2. **So khớp từ khóa (Matching):** Với mỗi tiêu đề tin tức mới trong năm 2025, hệ thống sẽ tách từ và đếm số lượng từ khóa trùng khớp (overlap) với từng bộ từ điển chủ đề.

3. **Gán nhãn (Assignment):** Tiêu đề sẽ được gán cho chủ đề có số lượng từ khóa trùng khớp nhiều nhất. Trong trường hợp có sự tranh chấp (số lượng trùng khớp bằng nhau), hệ thống ưu tiên các chủ đề nhạy cảm như *Topic_Price_Prediction* hoặc *Topic_Altcoins_Memes* để bắt tín hiệu tốt hơn.

Phương pháp này đảm bảo tính nhất quán của hệ thống nhãn chủ đề giữa quá khứ và tương lai, đồng thời cho tốc độ xử lý cực nhanh (Real-time capable).

```
import json

def assign_topic(df, json_path='Models/topic_keywords.json'):
    print(f"3. Classifying Topics using Keyword Matching...")
    data = df.copy()

    # 1. Tải bộ từ điển từ khóa đã học từ tập Train
    with open(json_path, 'r') as f:
        raw_map = json.load(f)
    # Chuyển đổi key sang int và value sang set để tối ưu tốc độ tra cứu
    topic_keywords = {int(k): set(v) for k, v in raw_map.items()}

    # Định nghĩa ánh xạ tên chủ đề
```

```
label_map = {
    0: 'Topic_Exchange_DeFi', 1: 'Topic_Market_Stats',
    2: 'Topic_Altcoins_Memes', 3: 'Topic_Macro_Politics',
    4: 'Topic_Price_Prediction'
}

# 2. Hàm phân loại dựa trên mức độ trùng khớp (Overlap)
def classify(text):
    if not isinstance(text, str): return 'Topic_None'
    words = set(text.lower().split()) # Tách từ đơn giản

    best_topic = None
    max_overlap = 0

    for topic_id, key_list in topic_keywords.items():
        overlap = len(words.intersection(key_list))

        # Chọn chủ đề có nhiều từ khóa trùng nhất
        if overlap > max_overlap:
            max_overlap = overlap
            best_topic = topic_id
        # Xử lý tranh chấp (Tie-breaking): Ưu tiên Topic 4 và 2
        elif overlap == max_overlap and overlap > 0:
            if topic_id in [4, 2]:
                best_topic = topic_id

    if best_topic is None: return 'Topic_None'
    return label_map.get(best_topic, 'Topic_None')

# Áp dụng phân loại
data['News_Topic'] = data['clean_title'].apply(classify)
return data
```

6.3 Xây dựng Mô hình Chấm điểm Tín hiệu

Sau khi đã có tập dữ liệu giao dịch (Transaction Database) và kho tri thức là các luật kết hợp (Association Rules), bước tiếp theo là xây dựng một cơ chế định lượng để chuyển đổi các mẫu hình rời rạc thành một con số tín hiệu duy nhất. Hệ thống này đóng vai trò như "bộ não" trung tâm, quyết định cường độ của tín hiệu Mua hoặc Bán tại từng thời điểm.

6.3.1 Bộ lọc và Trọng số hóa Quy tắc

Trong hàng nghìn luật được sinh ra từ thuật toán FP-Growth, không phải luật nào cũng có giá trị dự báo như nhau. Một số luật có độ tin cậy thấp hoặc độ nâng không đáng kể cần phải được loại bỏ để tránh gây nhiễu.

Quy trình lọc và gán trọng số được thực hiện như sau:

1. **Lọc luật hành động (Actionable Rules):** Chỉ giữ lại các luật mà vế phải (Consequents) là kết quả giao dịch mong muốn ("Pump" hoặc "Dump").

2. **Áp dụng ngưỡng chất lượng:** Các luật phải thỏa mãn ngưỡng $Confidence \geq 0.3$ và $Lift > 1.0$.

3. **Tính toán Sức mạnh quy tắc (Rule Power):** Đây là bước quan trọng nhất. Thay vì coi các luật là bình đẳng, nhóm gán cho mỗi luật một trọng số dựa trên chất lượng của nó:

$$Power(R) = Confidence(R) \times Lift(R) \tag{6.2}$$

Công thức này đảm bảo rằng một quy tắc vừa có độ chính xác cao, vừa có tính liên kết mạnh sẽ đóng góp tiếng nói lớn hơn trong quyết định cuối cùng.

```
def load_trading_rules(rules_path, min_conf=0.3, min_lift=1.0):
    print(f"1. Loading Rules from {rules_path}...")
    rules = pd.read_csv(rules_path)

    # Lọc luật hướng tới Pump/Dump và thỏa mãn ngưỡng
    actionable = rules[
        (rules['consequents'].astype(str).str.contains('Pump|Dump')) &
        (rules['confidence'] >= min_conf) &
        (rules['lift'] > min_lift)
    ].copy()

    # Tiền xử lý antecedent thành tập hợp (set) để so khớp nhanh
    actionable['itemset'] = actionable['antecedents'].apply(
        lambda x: set(x.replace("'", "").replace("frozenset({", "")
                       .replace("})", "").split(', '))
    )

    # Tính trọng số sức mạnh
    actionable['power'] = actionable['confidence'] * actionable['lift']

    return actionable
```

6.3.2 Thuật toán Tổng hợp Tín hiệu

Tại sao chúng ta cần một thuật toán tổng hợp? Trong thực tế, tại một thời điểm t, trạng thái thị trường (bao gồm hàng chục đặc trưng như RSI, Volume, News...) có thể kích hoạt đồng thời nhiều luật khác nhau, thậm chí là các luật đối nghịch.

- Ví dụ: Tập đặc trưng $\{A, B\}$ kích hoạt luật Mua (Pump).
- Nhưng đồng thời, tập đặc trưng $\{C, D\}$ trong cùng thời điểm đó lại kích hoạt luật Bán (Dump).

Nếu chỉ chọn một luật duy nhất, hệ thống sẽ bị phiến diện. Do đó, nhóm sử dụng cơ chế *Cộng hưởng Tín hiệu (Signal Resonance)*. Tại mỗi cây nến, thuật toán sẽ tính tổng sức mạnh của tất cả các luật Mua khớp lệnh ($Score_{Bull}$) và tất cả các luật Bán khớp lệnh ($Score_{Bear}$). Tín hiệu ròng (Net Signal) là hiệu số giữa hai lực lượng này:

$$Net\_Signal_t = \sum_{r \in Rules_{Buy}} Power(r) - \sum_{r \in Rules_{Sell}} Power(r) \tag{6.3}$$

Nếu $Net\_Signal > 0$, phe Mua đang áp đảo và ngược lại. Độ lớn của giá trị này thể hiện mức độ đồng thuận (Conviction) của hệ thống.

```
def generate_signals(df_transactions, rules_df):
    print("2. Scoring Signals...")
    data = df_transactions.copy()

    # Chuẩn bị dữ liệu so khớp
    feature_cols = [c for c in data.columns if c not in
                    ['open_time', 'close', 'target_label', 'cluster']]
    # Chuyển mỗi hàng thành một tập hợp (Set) các đặc trưng
    records = data[feature_cols].astype(str).values
    transaction_sets = [set(row) for row in records]

    bull_scores = np.zeros(len(data))
    bear_scores = np.zeros(len(data))

    # Tách tập luật
    pump_rules = rules_df[rules_df['consequents'].str.contains('Pump')]
    dump_rules = rules_df[rules_df['consequents'].str.contains("Dump")]
```

```
    # Quét từng giao dịch (Scoring Loop)
    for i, current_basket in enumerate(tqdm(transaction_sets)):
        # Cộng điểm các luật Mua khớp với trạng thái hiện tại
        bull_power = sum(r.power for r in pump_rules.itertuples()
                         if r.itemset.issubset(current_basket))

        # Cộng điểm các luật Bán khớp với trạng thái hiện tại
        bear_power = sum(r.power for r in dump_rules.itertuples()
                         if r.itemset.issubset(current_basket))

        bull_scores[i] = bull_power
        bear_scores[i] = bear_power

    data['Bull_Score'] = bull_scores
    data['Bear_Score'] = bear_scores
    data['Net_Signal'] = bull_scores - bear_scores

    return data
```

6.4 Thiết lập và Thực thi Kiểm thử Chiến lược

Giai đoạn cuối cùng và quan trọng nhất của nghiên cứu là đưa các mô hình vào môi trường giả lập để đánh giá hiệu quả đầu tư thực tế. Trong phần này, nhóm trình bày chi tiết về thiết kế kịch bản thử nghiệm, cơ chế vận hành của bộ máy mô phỏng (Backtest Engine) và các tham số chiến lược được áp dụng.

6.4.1 Thiết kế kịch bản thử nghiệm

Để kiểm chứng giả thuyết nghiên cứu về vai trò của thông tin trong việc dự báo biến động giá, nhóm áp dụng phương pháp kiểm thử đối chứng (Comparative Analysis) trên tập dữ liệu năm 2025. Mục tiêu cốt lõi của thiết kế này là cô lập tác động của biến số "tin tức" để đo lường chính xác đóng góp của nó vào hiệu quả giao dịch cuối cùng. Thí nghiệm được cấu trúc xoay quanh ba chiến lược hoạt động song song, đại diện cho ba tư duy đầu tư khác nhau trên thị trường.

Chiến lược đầu tiên, đóng vai trò là mức chuẩn cơ sở (Benchmark), là chiến lược Mua và Nắm giữ thụ động (Buy & Hold). Trong kịch bản này, hệ thống giả định thực hiện một lệnh mua Bitcoin tại thời điểm mở cửa của giai đoạn kiểm thử (01/01/2025) và duy trì vị thế bất chấp mọi biến động cho đến khi kết thúc kỳ. Đây là thước đo tiêu chuẩn vàng (Gold Standard) trong mọi bài toán tài chính nhằm đánh giá "Beta" của thị trường. Việc so sánh với chiến lược này là bắt buộc để xác định xem liệu các nỗ lực giao dịch chủ động, với độ phức tạp về thuật toán và chi phí giao dịch phát sinh, có thực sự mang lại hiệu quả vượt trội (Alpha) so với mức tăng trưởng tự nhiên của tài sản hay không.

Đối trọng tiếp theo là Chiến lược Kỹ thuật thuần túy (Technical Strategy), đại diện cho trường phái giao dịch định lượng truyền thống. Hệ thống ra quyết định hoàn toàn dựa trên các tập luật kết hợp được khai phá từ dữ liệu giá và khối lượng, bao gồm các chỉ báo động lượng (RSI), cấu trúc pha Wyckoff và các mô hình nến Smart Money. Chiến lược này hoạt động dựa trên giả định của Phân tích kỹ thuật rằng "giá phản ánh tất cả" và các mẫu hình hành vi trong quá khứ sẽ có xu hướng lặp lại. Tuy nhiên, điểm yếu cố hữu của phương pháp này là thiếu vắng sự nhạy bén với các sự kiện vĩ mô hoặc các tin tức thiên nga đen, dẫn đến nguy cơ sinh ra nhiều tín hiệu nhiễu trong các giai đoạn thị trường biến động phi kỹ thuật.

Cuối cùng và quan trọng nhất là Chiến lược AI Lai ghép (Hybrid Strategy), phương pháp đề xuất chính của nghiên cứu. Khác với hai hướng tiếp cận trên, chiến lược này tích hợp thêm chiều dữ liệu phi cấu trúc từ tin tức (Cảm xúc, Mức độ quan tâm, Chủ đề) vào bộ quy tắc ra quyết định. Tín hiệu giao dịch chỉ được kích hoạt khi và chỉ khi có sự "đồng thuận" (Confluence) giữa tín hiệu kỹ thuật và luồng thông tin. Ví dụ, một tín hiệu mua kỹ thuật sẽ chỉ được thực thi nếu đi kèm với tin tức tích cực hoặc sự bùng nổ về mức độ quan tâm của cộng đồng. Kỳ vọng đặt ra cho mô hình lai ghép này là khả năng lọc nhiễu vượt trội, giúp hệ thống hạn chế các giao dịch sai lầm (False Positives) và tối ưu hóa tỷ lệ lợi nhuận trên rủi ro.

6.4.2 Cơ chế mô phỏng giao dịch

Để đảm bảo tính trung thực của kết quả, nhóm đã xây dựng một lớp đối tượng `BacktestEngine` hoạt động theo cơ chế hướng sự kiện (Event-driven) trên từng cây nến phút. Bộ máy này mô phỏng các ràng buộc thực tế như phí giao dịch và quy tắc quản lý vốn.

Các quy tắc vận hành cốt lõi bao gồm:

1. **Vốn khởi điểm:** 10,000 USD.
2. **Phí giao dịch (Transaction Fee):** 0.1% trên tổng giá trị lệnh (tương đương mức phí taker tiêu chuẩn trên Binance). Việc tính phí là bắt buộc để trừng phạt các chiến lược giao dịch quá tần suất (Over-trading).
3. **Trạng thái vị thế (State Machine):** Tại mỗi thời điểm, tài khoản chỉ có thể ở một trong hai trạng thái:
   - **Full Cash:** Nắm giữ 100% tiền mặt. Khi có tín hiệu Mua (BUY), hệ thống sẽ dùng toàn bộ tiền để mua Bitcoin (sau khi trừ phí).
   - **Full Crypto:** Nắm giữ 100% Bitcoin. Khi có tín hiệu Bán (SELL), hệ thống sẽ bán toàn bộ lượng coin để chuyển về tiền mặt.
4. **Định giá tài sản (Mark-to-Market):** Tại mỗi bước thời gian, tổng giá trị danh mục (Portfolio Value) được cập nhật theo giá thị trường hiện hành để theo dõi biến động tài sản theo thời gian thực.

```
class BacktestEngine:
    def __init__(self, initial_capital=10000, fee_pct=0.001):
        self.initial_capital = initial_capital
        self.fee_pct = fee_pct

    def run_backtest(self, df, strategy_name="Strategy"):
        # Chuẩn bị dữ liệu và biến trạng thái
        data = df.copy().sort_values('open_time').reset_index(drop=True)
        cash = self.initial_capital
        btc_balance = 0
        is_invested = False

        portfolio_values = []
        trades = []

        # Vòng lặp mô phỏng từng phút (Bar-by-bar Simulation)
        for i, row in data.iterrows():
            price = row['close']
            action = row['Action']

            # Logic KHỚP LỆNH BÁN
            if is_invested and action == "SELL":
                revenue = btc_balance * price
                fee = revenue * self.fee_pct    # Trừ phí
                cash = revenue - fee
                btc_balance = 0                 # Đóng vị thế
                is_invested = False
                trades.append({'date': row['open_time'], 'type': 'SELL',
                               'price': price, 'value': cash})

            # Logic KHỚP LỆNH MUA
            elif not is_invested and action == 'BUY':
                fee = cash * self.fee_pct       # Trừ phí trước khi mua
                net_cash = cash - fee
                btc_balance = net_cash / price  # Mở vị thế
```

```
                cash = 0
                is_invested = True
                trades.append({'date': row['open_time'], 'type': 'BUY',
                               'price': price, 'value': net_cash})

            # Cập nhật giá trị danh mục (NAV)
            current_val = (btc_balance * price) if is_invested else cash
            portfolio_values.append(current_val)

        # Tổng hợp kết quả
        data['Portfolio_Value'] = portfolio_values
        total_return = ((portfolio_values[-1] - self.initial_capital) /
                        self.initial_capital * 100)

        return data, pd.DataFrame(trades), total_return
```

6.4.3 Thiết lập Ngưỡng hành động (Action Thresholds)

Biến tín hiệu đầu ra từ mô hình chấm điểm (`Net_Signal`) bản chất là một giá trị liên tục, đại diện cho cán cân cung cầu tại từng thời điểm. Để chuyển đổi đại lượng này thành các quyết định giao dịch nhị phân dứt khoát, nhóm nghiên cứu thiết lập một **ngưỡng cắt đối xứng (Symmetric Threshold) là** ± 5.0.

Cơ chế ra quyết định hoạt động dựa trên nguyên tắc "Sự đồng thuận cao" (High Conviction). Cụ thể, một lệnh **BUY** chỉ được kích hoạt khi `Net_Signal` vượt quá ngưỡng $+5.0$, đồng nghĩa với việc tổng sức mạnh (tích của Độ tin cậy và Độ nâng) của các luật Mua phải vượt trội hơn các luật Bán ít nhất 5 đơn vị. Ngược lại, lệnh **SELL** sẽ được thực thi khi áp lực bán áp đảo, đẩy tín hiệu xuống dưới mức -5.0. Khoảng giá trị nằm giữa biên độ này (từ -5.0 đến 5.0) được định nghĩa là vùng **HOLD**. Đây là vùng đệm chiến lược đóng vai trò như một *bộ lọc nhiễu (Noise Filtering)*, ngăn hệ thống thực hiện các giao dịch sai lầm khi xu hướng thị trường chưa thực sự rõ ràng.

Kết quả thống kê phân bố lệnh trên tập kiểm thử đã minh chứng cho tính kỷ luật của cơ chế này. Trong tổng số gần 490,000 phút giao dịch, cả Chiến lược Kỹ thuật và Chiến lược Hybrid chỉ thực hiện số lượng lệnh rất hạn chế (lần lượt là khoảng 1,439 và 1,464 lệnh tổng cộng). Đáng chú ý, tỷ lệ thời gian hệ thống ở trạng thái chờ (HOLD) chiếm tới hơn **99.7%**. Điều này khẳng định mô hình không hoạt động theo kiểu giao dịch tần suất cao (HFT) mà tuân theo phong cách *"Xạ thủ bắn tỉa" (Sniper approach)*: kiên nhẫn chờ đợi và chỉ bóp cò khi cơ hội có xác suất thắng cao nhất xuất hiện.

Dưới đây là đoạn mã hiện thực hóa logic chuyển đổi tín hiệu sang hành động:

```
def apply_trading_action(df, threshold_buy=5, threshold_sell=-5):
    data = df.copy()

    conditions = [
        data['Net_Signal'] > threshold_buy,   # Tín hiệu Mua mạnh (High Conviction)
        data['Net_Signal'] < -threshold_sell  # Tín hiệu Bán mạnh
    ]
    choices = ['BUY', 'SELL']

    # Mặc định là HOLD nếu tín hiệu nằm trong vùng nhiễu
    data['Action'] = np.select(conditions, choices, default='HOLD')
    return data

# Áp dụng ngưỡng thống nhất cho cả hai chiến lược để đảm bảo công bằng
df_scored = apply_trading_action(df_scored, threshold_sell=5, threshold_buy=5)
integrated_df = apply_trading_action(integrated_df, threshold_sell=5, threshold_buy=5)
```

6.4.4 Kết quả thực nghiệm và Đánh giá hiệu năng đa kịch bản

Trong giao dịch thuật toán, chi phí giao dịch đóng vai trò là biến số ngoại sinh then chốt, quyết định trực tiếp đến khả năng sinh lời thực tế của mô hình. Để kiểm chứng độ bền vững của các chiến lược đề xuất, nghiên cứu tiến hành phân tích độ nhạy trên 4 kịch bản phí giao dịch khác nhau: 1.0%, 0.5%,

0.1% (mức chuẩn) và 0.0% (lý thuyết). Kết quả được phân tích trên hai khía cạnh chính là hiệu suất lợi nhuận và mức độ rủi ro.

Phân tích hiệu suất lợi nhuận ròng

Bảng 6.1 trình bày chi tiết lợi nhuận ròng của ba chiến lược qua các kịch bản phí. Tại mức phí tiêu chuẩn 0.1%, chiến lược AI Hybrid thể hiện ưu thế vượt trội khi đạt mức lợi nhuận cao nhất +10.92%, đánh bại hoàn toàn chiến lược Kỹ thuật thuần túy (+8.39%) và thị trường chung đang suy giảm (Buy & Hold lỗ -4.87%). Kết quả này khẳng định rằng việc tích hợp dữ liệu tin tức giúp hệ thống nắm bắt được các cơ hội giao dịch chất lượng cao mà phân tích kỹ thuật đơn thuần thường bỏ lỡ.

Đáng chú ý, khi xét đến các kịch bản phí cao (0.5% - 1.0%), sự khác biệt về tính hiệu quả giữa hai mô hình thuật toán được bộc lộ rõ rệt. Chiến lược Kỹ thuật sụp đổ hoàn toàn với mức lỗ lên tới -21.69% do tần suất giao dịch quá dày đặc (361 lệnh), khiến chi phí bào mòn toàn bộ lợi nhuận. Ngược lại, chiến lược AI Hybrid, nhờ bộ lọc tin tức khắt khe, chỉ thực hiện 182 lệnh và duy trì mức lỗ thấp hơn đáng kể (-5.85%), gần tương đương với mức sụt giảm tự nhiên của thị trường. Điều này chứng minh khả năng kháng nhiễu và bảo toàn vốn tốt hơn của mô hình lai trong môi trường giao dịch khắc nghiệt.

Bảng 6.1: Tổng hợp Lợi nhuận ròng (%) của các chiến lược theo mức phí

Chiến lược	**Mức phí giao dịch**			
	1.0%	**0.5%**	**0.1% (Chuẩn)**	**0.0% (Lý thuyết)**
Buy & Hold	-4.96%	-4.91%	-4.87%	-4.86%
Technical Only	-21.69%	-21.69%	+8.39%	+12.37%
AI Hybrid	-5.85%	-5.85%	**+10.92%**	**+12.96%**

Phân tích rủi ro và Đặc điểm tài sản

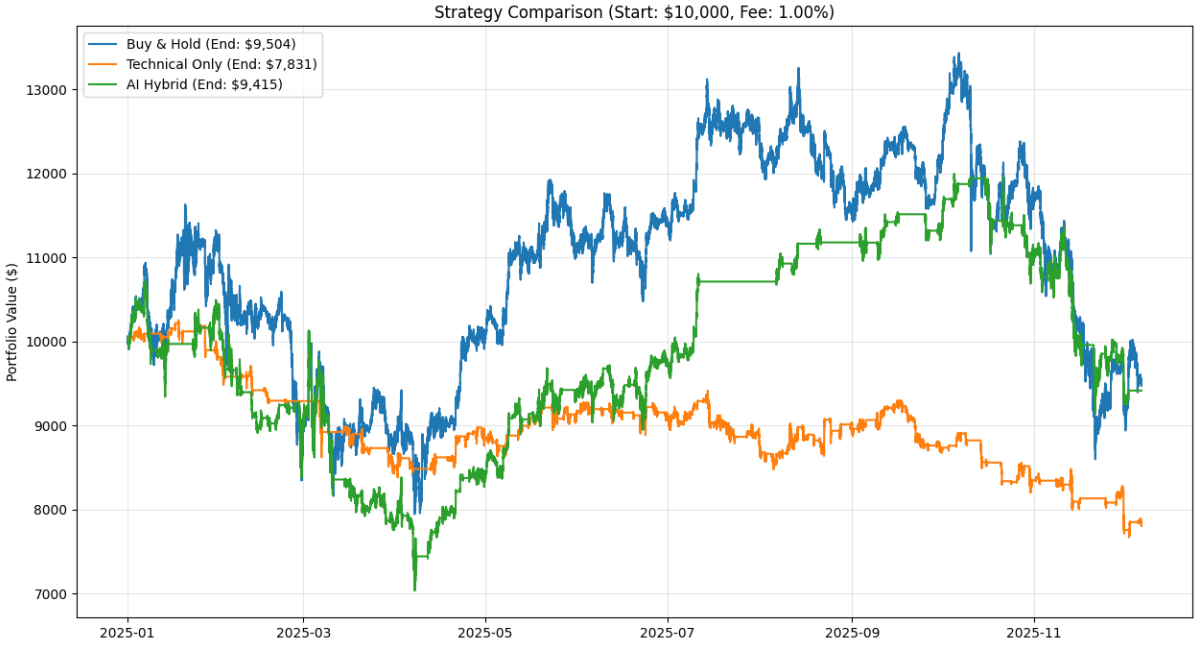
Bên cạnh lợi nhuận, các chỉ số về độ sụt giảm tài khoản (Max Drawdown) và giá trị tài sản đỉnh (Peak Equity) được tổng hợp tại Bảng 6.2 để đánh giá hành vi của chiến lược. Một quan sát quan trọng tại kịch bản chuẩn 0.1% là chỉ số Max Drawdown của chiến lược AI Hybrid (-30.54%) lại cao hơn so với Technical Only (-14.80%). Hiện tượng này phản ánh sự đánh đổi cốt lõi trong triết lý giao dịch của hai mô hình. Chiến lược Kỹ thuật hoạt động dựa trên các dao động ngắn hạn (Noise Trading), thường chốt lời hoặc cắt lỗ rất nhanh, giúp đường cong vốn mượt mà hơn nhưng lại hạn chế tiềm năng tăng trưởng, thể hiện qua mức đỉnh tài sản chỉ đạt $12,052.

Trái lại, chiến lược AI Hybrid mang đặc tính "Bám theo xu hướng" (Trend Following). Khi có sự xác nhận của tin tức, mô hình chấp nhận giữ vị thế lâu hơn để tận dụng tối đa đà tăng, đẩy giá trị tài sản lên mức đỉnh cao nhất là $13,721. Hệ quả tất yếu của việc "gồng lãi" là mô hình phải chấp nhận các đợt rung lắc mạnh hơn của thị trường trước khi tín hiệu đảo chiều được xác nhận, dẫn đến mức sụt giảm tạm thời lớn hơn. Tuy nhiên, xét trên hiệu quả tổng thể, AI Hybrid vẫn đảm bảo mức sụt giảm thấp hơn so với việc nắm giữ thụ động (Buy & Hold: -35.99%), đồng thời mang lại lợi nhuận cuối cùng cao nhất.

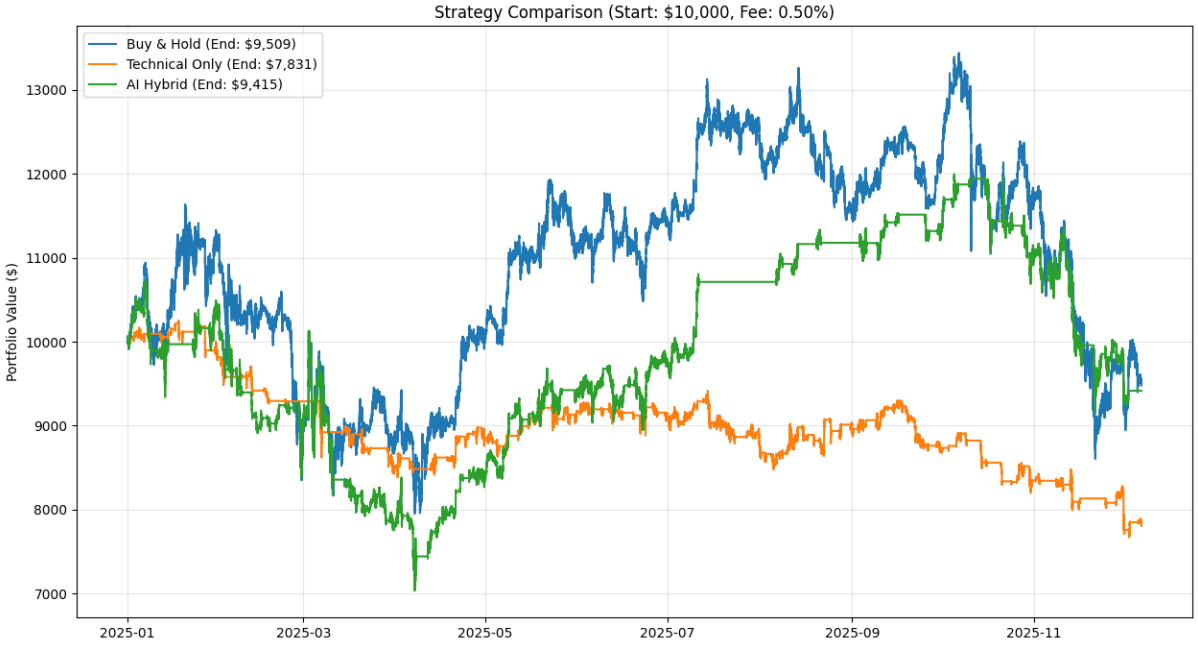
Bảng 6.2: Tổng hợp chỉ số Rủi ro và Giá trị tài sản đỉnh theo mức phí

Chiến lược	**Max Drawdown (%)**				**Giá trị tài sản cao nhất ($)**			
	1.0%	**0.5%**	**0.1%**	**0.0%**	**1.0%**	**0.5%**	**0.1%**	**0.0%**
Buy & Hold	-35.99	-35.99	-35.99	-35.99	13,432	13,438	13,444	13,445
Technical Only	-25.21	-25.21	-14.80	-14.41	10,253	10,253	12,052	12,405
AI Hybrid	-34.43	-34.43	-30.54	-30.09	11,995	11,995	**13,721**	**13,933**

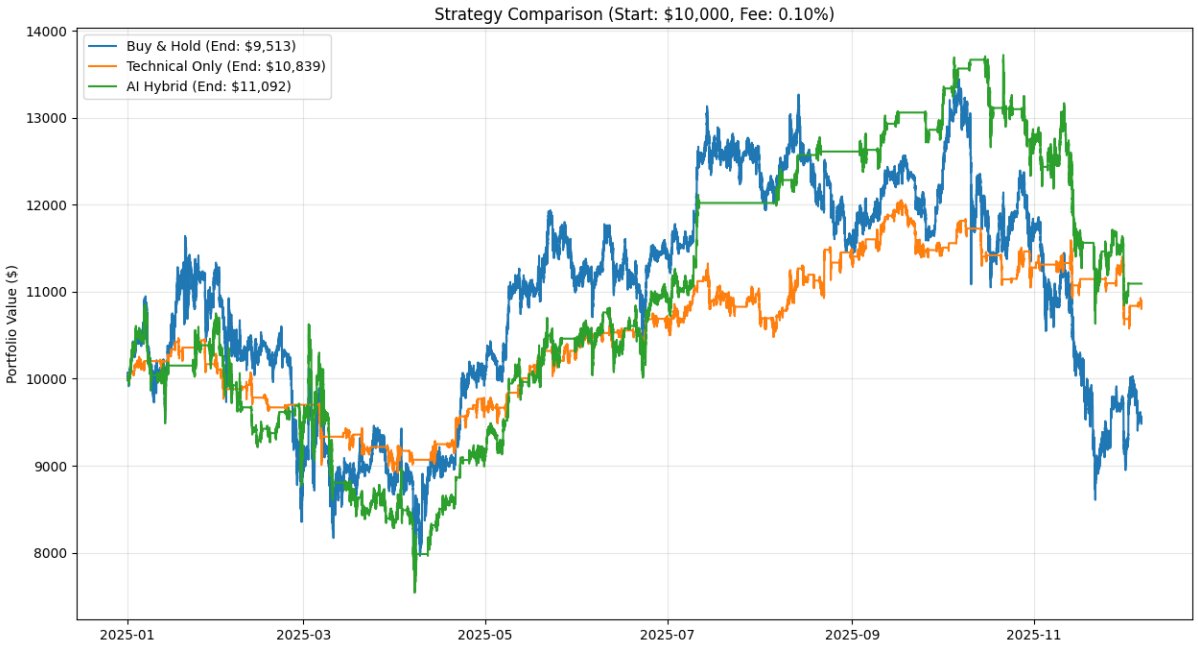
Tổng kết lại, kết quả thực nghiệm chứng minh rằng việc tích hợp dữ liệu tin tức vào mô hình giao dịch mang lại hiệu quả vượt trội về mặt lợi nhuận ròng trong điều kiện thị trường thực tế. Mặc dù phải chịu mức độ biến động tài sản lớn hơn trong ngắn hạn so với chiến lược thuần kỹ thuật, nhưng AI Hybrid tạo ra mức đỉnh tài sản cao hơn và tránh được rủi ro bào mòn vốn do chi phí giao dịch, đáp ứng tốt mục tiêu tối ưu hóa tăng trưởng tài sản dài hạn.



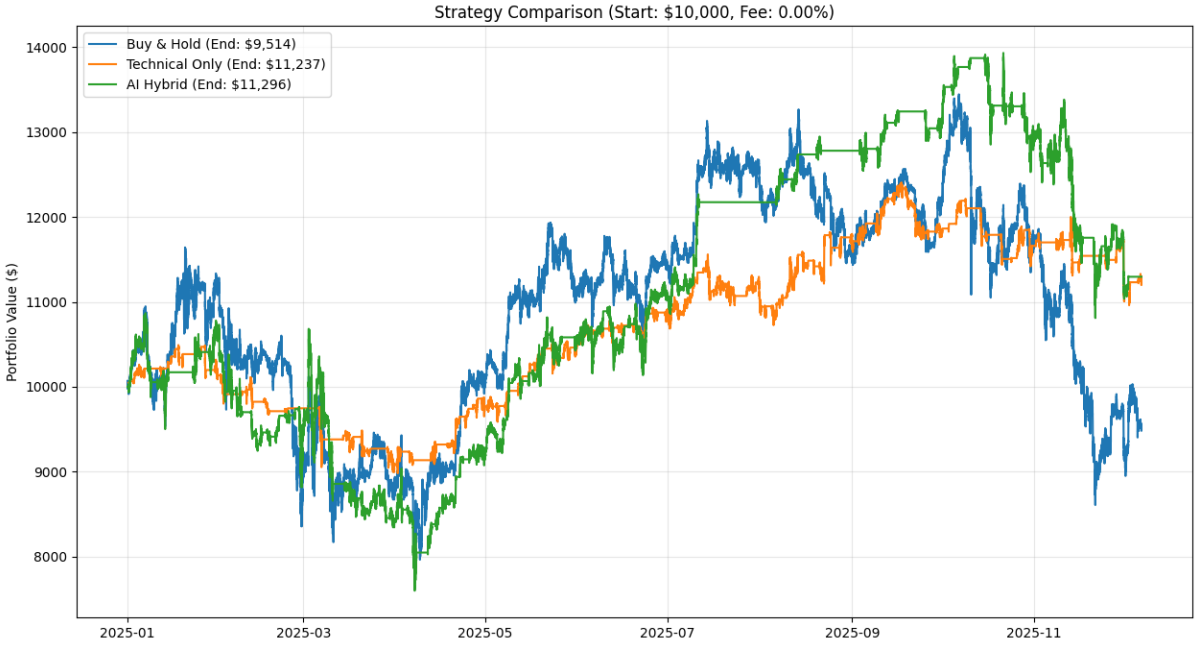
Hình 6.2: so sánh giá trị tài sản của các chiến lược theo thời gian (Mức phí giao dịch 1%)



Hình 6.3: so sánh giá trị tài sản của các chiến lược theo thời gian (Mức phí giao dịch 0.5%)



Hình 6.4: so sánh giá trị tài sản của các chiến lược theo thời gian (Mức phí giao dịch 0.1%)



Hình 6.5: so sánh giá trị tài sản của các chiến lược theo thời gian (Mức phí giao dịch 0%)

Chương 7

Tổng kết và phương hướng phát triển

7.1 Tổng kết

Nghiên cứu này đã hoàn thành mục tiêu xây dựng một hệ thống hỗ trợ ra quyết định giao dịch Bitcoin tự động dựa trên sự kết hợp giữa kỹ thuật khai phá dữ liệu truyền thống và xử lý ngôn ngữ tự nhiên. Kết quả thực nghiệm cho thấy mô hình lai (AI Hybrid) không chỉ vượt trội hơn so với chiến lược mua và nắm giữ thụ động mà còn khắc phục được nhược điểm giao dịch quá tần suất của các phương pháp phân tích kỹ thuật đơn thuần. Điểm mạnh cốt lõi của phương pháp đề xuất nằm ở khả năng tích hợp ngữ cảnh thông tin vào dữ liệu giá. Bằng cách sử dụng tin tức như một bộ lọc nhiễu, hệ thống đã chuyển đổi tư duy giao dịch từ việc phản ứng với mọi biến động giá sang chiến thuật "bắn tỉa", chỉ tham gia thị trường khi có sự đồng thuận cao giữa tín hiệu kỹ thuật và tâm lý đám đông. Bên cạnh đó, việc sử dụng các luật kết hợp (Association Rules) mang lại tính giải thích cao cho mô hình, cho phép người sử dụng hiểu rõ lý do đằng sau mỗi quyết định mua bán, khác biệt hoàn toàn với tính chất "hộp đen" của các mô hình học sâu hiện đại.

Tuy nhiên, nghiên cứu vẫn tồn tại những hạn chế nhất định cần nhìn nhận khách quan. Điểm yếu lớn nhất của mô hình hiện tại là độ trễ trong việc phản ứng với các tin tức thời gian thực do cơ chế tổng hợp dữ liệu theo khung giờ. Điều này khiến hệ thống có thể bỏ lỡ các cơ hội giao dịch chớp nhoáng hoặc phản ứng chậm trước các cú sập giá bất ngờ (Flash crash). Ngoài ra, chiến lược bám theo xu hướng dựa trên sự xác nhận của tin tức cũng buộc mô hình phải chấp nhận mức độ biến động tài sản (Drawdown) cao hơn trong ngắn hạn để đi hết con sóng lớn, đòi hỏi nhà đầu tư phải có tâm lý vững vàng và khả năng chịu đựng rủi ro tốt. Hơn nữa, độ phức tạp trong việc vận hành và bảo trì luồng dữ liệu đa nguồn cũng là một rào cản không nhỏ khi triển khai hệ thống vào thực tế.

7.2 Phương hướng phát triển

Để khắc phục các hạn chế nêu trên và nâng cao hiệu suất của hệ thống trong tương lai, nhóm nghiên cứu đề xuất một số hướng cải thiện tiềm năng. Trước hết, về mặt thuật toán, việc thay thế hoặc bổ sung các luật kết hợp tĩnh bằng các mô hình học sâu chuỗi thời gian như LSTM (Long Short-Term Memory) hay Transformer là một hướng đi hứa hẹn. Các mô hình này có khả năng nắm bắt tốt hơn sự phụ thuộc thời gian dài hạn và các mối quan hệ phi tuyến tính phức tạp mà thuật toán FP-Growth có thể bỏ sót. Đồng thời, cơ chế Học tăng cường (Reinforcement Learning) có thể được áp dụng để tối ưu hóa động các ngưỡng giao dịch (Thresholds) thay vì sử dụng các giá trị cố định, giúp mô hình thích nghi linh hoạt hơn với sự thay đổi của chế độ thị trường.

Về mặt dữ liệu, không gian đặc trưng cần được mở rộng thêm các nguồn thông tin định lượng khác bên cạnh giá và tin tức. Cụ thể, việc tích hợp dữ liệu On-chain (như dòng tiền vào ra các ví cá mập, tỷ lệ đòn bẩy trên sàn phái sinh) sẽ cung cấp cái nhìn sâu sắc hơn về hành vi của các nhà tạo lập thị trường. Song song với đó, hệ thống xử lý ngôn ngữ tự nhiên cần được nâng cấp lên cơ chế thời gian thực (Real-time Streaming) thay vì xử lý theo lô, cho phép đánh giá tác động của tin tức ngay khi chúng vừa được công bố. Cuối cùng, việc bổ sung thêm các quy tắc quản trị vốn nâng cao, như tự động điều chỉnh khối lượng lệnh dựa trên độ biến động (Volatility sizing), sẽ giúp giảm thiểu rủi ro sụt giảm tài khoản và làm mượt đường cong tăng trưởng vốn trong dài hạn.

Tài liệu tham khảo

[1] Charu C Aggarwal. *Data mining: the textbook*. Springer, 2015. Sách giáo khoa chuẩn về Data Mining để tham khảo lý thuyết chung.

[2] Elie Bouri, Rangan Gupta, and Seyed Mehdi Hosseini. Herding behaviour in cryptocurrencies. *Finance Research Letters*, 29:216–221, 2019. Cơ sở cho việc phân tích tâm lý đám đông (Crowd Sentiment).

[3] Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 1–12, 2000. Bài báo gốc đề xuất thuật toán FP-Growth mà bạn sử dụng.

[4] Ladislav Kristoufek. Bitcoin meets google trends and wikipedia: Quantifying the relationship between phenomena of the internet era. *Scientific reports*, 3(1):3415, 2013. Chứng minh mối liên hệ giữa thông tin internet và giá Bitcoin.

[5] Marcos Lopez de Prado. *Advances in financial machine learning*. John Wiley & Sons, 2018. Tác giả này đẻ ra khái niệm Triple-Barrier Method bạn dùng để gán nhãn.

[6] Tim Loughran and Bill McDonald. When is a liability not a liability? textual analysis, dictionaries, and 10-ks. *The Journal of Finance*, 66(1):35–65, 2011. Bài báo nền tảng về phân tích cảm xúc trong tài chính.

[7] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967. Tài liệu kinh điển về thuật toán K-Means.

[8] John J Murphy. *Technical analysis of the financial markets: A comprehensive guide to trading methods and applications*. Penguin, 1999. Sách gối đầu giường về RSI, EMA, Volume.

[9] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019. Mô hình SBERT bạn dùng để Embedding tiêu đề tin tức.

[10] Richard D Wyckoff. *The Richard D. Wyckoff Method of Trading and Investing in Stocks*. Wyckoff Associates, 1931. Nguồn gốc lý thuyết Wyckoff bạn dùng để phân cụm pha thị trường.