LLM 기반 투자 어드바이저 프로젝트

빅테크 기업 주가 데이터 분석 (2015-2025)

공공공일
송채원 & 정다희

목차

데이터셋 소개

🖥️ 빅테크 10년간 일별 주가 데이터 (AAPL, MSFT, GOOGL 등)

- **데이터 출처**: Yahoo Finance API (yfinance)
- **시간 범위**: 2015.01.02 ~ 2025.03.21 (약 2,570 거래일)
- **기본 변수**:
  - (1) 기본 주가 데이터: Open, High, Low, Close, Volume
  - (2) 시간적 특성: Year, Month, Quarter 등

목표

💡 투자 어드바이저 엔진

(1) **기술적 지표 신호 분석**: 모멘텀, 이동평균, RSI, MACD, 볼린저 밴드, 변동성 지표를 종합 분석

(2) **시장 추세 분석**: 상승/하락/횡보장 식별 및 추세 안정성 측정

(3) **지표 분석**: 급락 이벤트, 거래량 및 변동성 비정상 관계 분석

(4) **투자 점수 계산**: 투자자 성향(위험 성향, 투자 기간 등)에 따른 가중치 조정

(5) **위험 및 기회 평가**: 투자자 프로필에 맞춘 진입/퇴출 전략 생성

💡 LLM 기반 Agent

- **데이터 처리 레이어**: 주가 데이터 수집 및 전처리 / 파생지표 생성
- **분석 엔진 레이어**: 투자 신호 계산 / 포트폴리오 최적화 및 보고서 생성
- **LLM 통합 레이어 (인터페이스 연동)**: 분석 결과 자연어로 변환 / 투자 인사이트 생성

Data Preprocessing

1. 기술적 지표 신호, 시장 추세 및 지표 분석을 위한 파생변수

```
# 기본 파생변수 계산
data['Daily_Return'] = data['Close'].pct_change() * 100
data['Volatility_20d'] = data['Daily_Return'].rolling(window=20).std()
data['Volume_Change'] = data['Volume'].pct_change() * 100
data['MA20'] = data['Close'].rolling(window=20).mean()
data['MA50'] = data['Close'].rolling(window=50).mean()
data['MA200'] = data['Close'].rolling(window=200).mean()
data['Price_Momentum'] = (data['Close'] / data['MA20'] - 1) * 100
```

```
# 시계열 특성 추가
data['Year'] = data.index.year
data['Month'] = data.index.month
data['Quarter'] = data.index.quarter
data['Month_Name'] = data.index.strftime('%B')
data['Day_of_Week'] = data.index.dayofweek
```

2. 종가, 수익률, 변동성 등의 비교를 위한 정규화 및 표준화

```
# Min-Max 정규화 (종가)
data['Close_Norm'] = (data['Close'] - data['Close'].min()) / (data['Close'].max() - data['Close'].min())

# Z-점수 표준화
data['Daily_Return_Z'] = (data['Daily_Return'] - data['Daily_Return'].mean()) / data['Daily_Return'].std()
data['Volatility_20d_Z'] = (data['Volatility_20d'] - data['Volatility_20d'].mean()) / data['Volatility_20d'].std()
```

Feature Engineering

1. RSI, MACD, 볼린저 밴드 등 파생변수를 사용한 기술적 지표 신호 분석

```
# RSI (Relative Strength Index) - 상대강도지수
delta = data['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
data['RSI'] = 100 - (100 / (1 + rs))

# 볼린저 밴드 (Bollinger Bands)
data['BB_Middle'] = data['Close'].rolling(window=20).mean()
std_dev = data['Close'].rolling(window=20).std()
data['BB_Upper'] = data['BB_Middle'] + (std_dev * 2)
data['BB_Lower'] = data['BB_Middle'] - (std_dev * 2)
data['BB_Width'] = (data['BB_Upper'] - data['BB_Lower']) / data['BB_Middle']

# MACD (Moving Average Convergence Divergence)
data['EMA_12'] = data['Close'].ewm(span=12, adjust=False).mean()
data['EMA_26'] = data['Close'].ewm(span=26, adjust=False).mean()
data['MACD'] = data['EMA_12'] - data['EMA_26']
data['MACD_Signal'] = data['MACD'].ewm(span=9, adjust=False).mean()
data['MACD_Histogram'] = data['MACD'] - data['MACD_Signal']
```

Feature Engineering

2. 이동평균선 기반 시장 추세 분석

```
# 이동평균선 기반 시장 상황 판단
data['Market_Trend'] = 'Sideways'  # 기본값: 횡보장

# 상승장 조건: 현재가 > MA50 > MA200 & 단기 추세 상승
up_trend = (data['Close'] > data['MA50']) & (data['MA50'] > data['MA200']) & (data['MA20'].diff() > 0)

# 하락장 조건: 현재가 < MA50 < MA200 & 단기 추세 하락
down_trend = (data['Close'] < data['MA50']) & (data['MA50'] < data['MA200']) & (data['MA20'].diff() < 0)

data.loc[up_trend, 'Market_Trend'] = 'Bullish'  # 상승장
data.loc[down_trend, 'Market_Trend'] = 'Bearish'  # 하락장
```

Feature Engineering

3. 급락 이벤트 등 지표 분석

📌EERR

```
# EERR (Extreme Event Recovery Rate) - 급락 이벤트 후 회복률
data['Large_Drop'] = data['Daily_Return'] <= -3  # 3% 이상 하락한 날
data['Recovery_Rate_5d'] = data['Close'].pct_change(5) * 100  # 5일 후 가격 변화율
data['Recovery_Rate_10d'] = data['Close'].pct_change(10) * 100  # 10일 후 가격 변화율

# 급락 이벤트 후 회복률 계산 (5일, 10일)
data['EERR_5d'] = np.where(data['Large_Drop'], data['Recovery_Rate_5d'].shift(-5), np.nan)
data['EERR_10d'] = np.where(data['Large_Drop'], data['Recovery_Rate_10d'].shift(-10), np.nan)
```

```
# 연도별 이벤트 빈도
yearly_events = extreme_drops.groupby('Year').size()

# 월별 이벤트 빈도
monthly_events = extreme_drops.groupby('Month').size()

# 요일별 이벤트 빈도
day_of_week_events = extreme_drops.groupby('Day_of_Week').size()

# 급락 이후 추가 하락 vs 회복 확률
further_decline_5d = (extreme_drops['EERR_5d'] <= threshold).mean() * 100
strong_recovery_5d = (extreme_drops['EERR_5d'] >= abs(threshold)).mean() * 100
```

📌VVAS

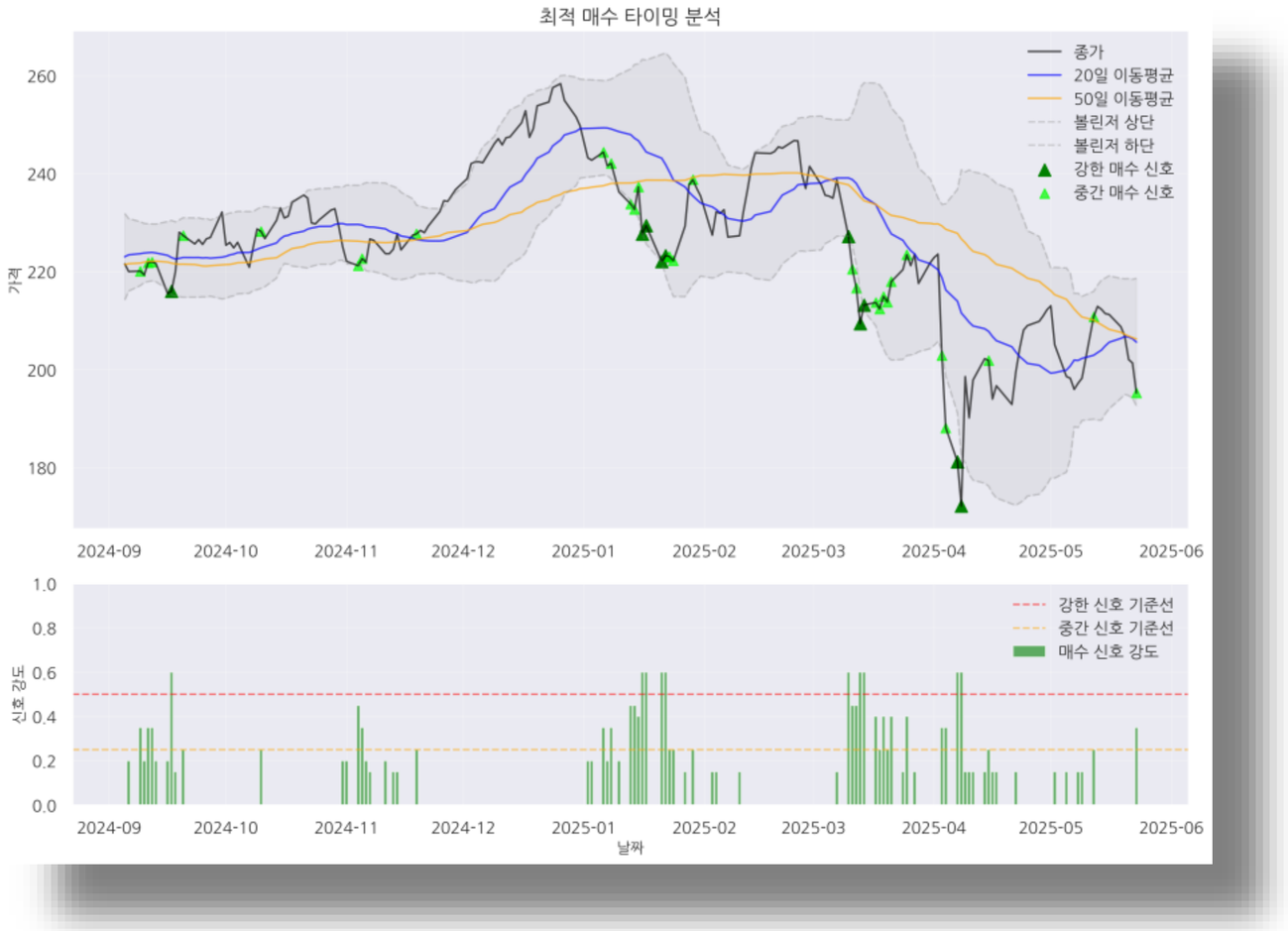
```
# VVAS (Volume-Volatility Anomaly Score) - 거래량-변동성 이상점수
# 1년(252 거래일) 기준 거래량과 변동성의 Z점수 차이
data['Volume_Z'] = (data['Volume'] - data['Volume'].rolling(252).mean()) / data['Volume'].rolling(252).std()
data['Volatility_Z'] = (data['Volatility_20d'] - data['Volatility_20d'].rolling(252).mean()) / data['Volatility_20d'].rolling(252).std()
data['VVAS'] = data['Volume_Z'] - data['Volatility_Z']
```

Exploratory Analysis of the Data

1. RSI, MACD, 볼린저 밴드 등 파생변수를 사용한 기술적 지표 신호 분석

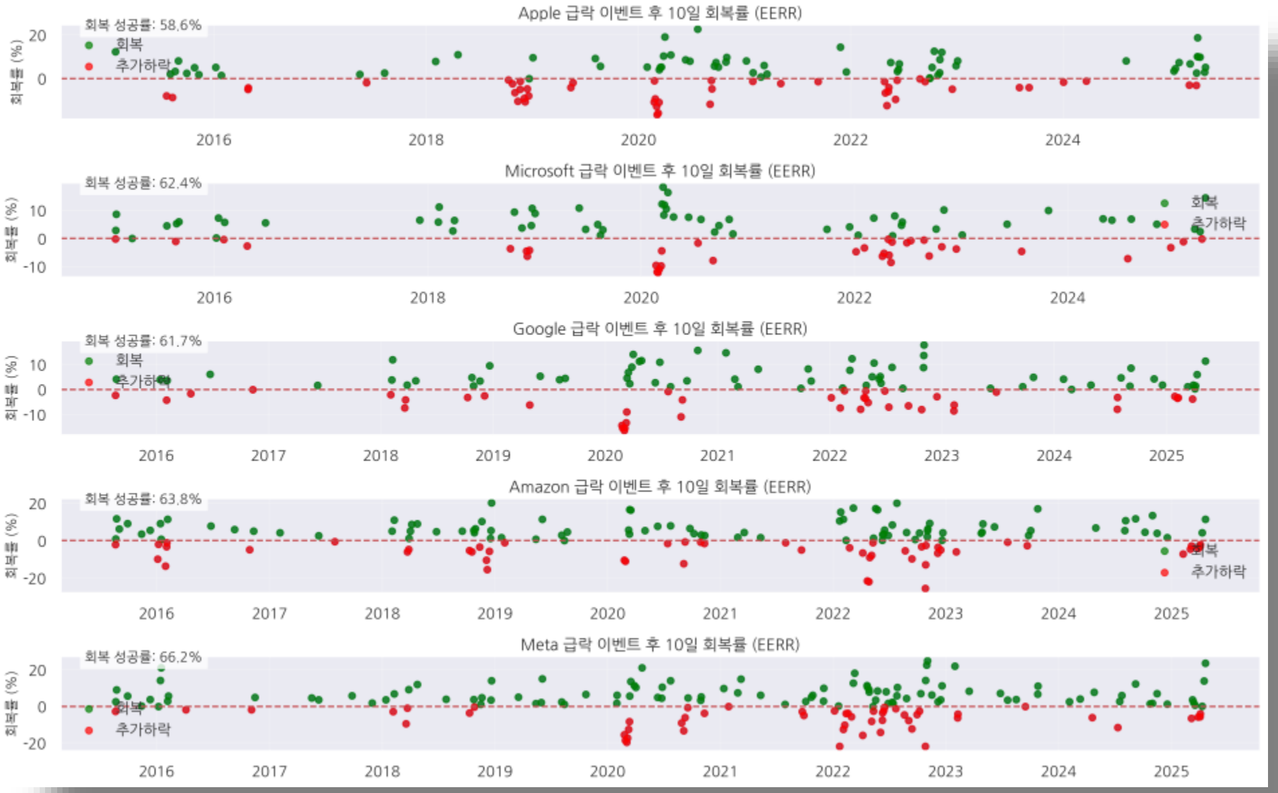
Exploratory Analysis of the Data

2. 이동평균선 기반 시장 추세 분석



Exploratory Analysis of the Data

3. 급락 이벤트 등 지표 분석



Methods: 투자 어드바이저 엔진

코드

```
# 기술적 지표 신호 분석
technical_signals = self._analyze_technical_signals(data)

# 시장 추세 분석
market_trend = self._analyze_market_trend(data)

# EERR 및 VVAS 지표 분석
advanced_signals = self._analyze_advanced_indicators(data)

# 투자 신호 계산
investment_score, signal_details = self._calculate_investment_score(
    technical_signals, market_trend, advanced_signals
)

# 위험 평가
risk_assessment = self._assess_risk(data)

# 투자 기회 평가
opportunity_score = self._assess_opportunity(data, investment_score, risk_assessment)
```

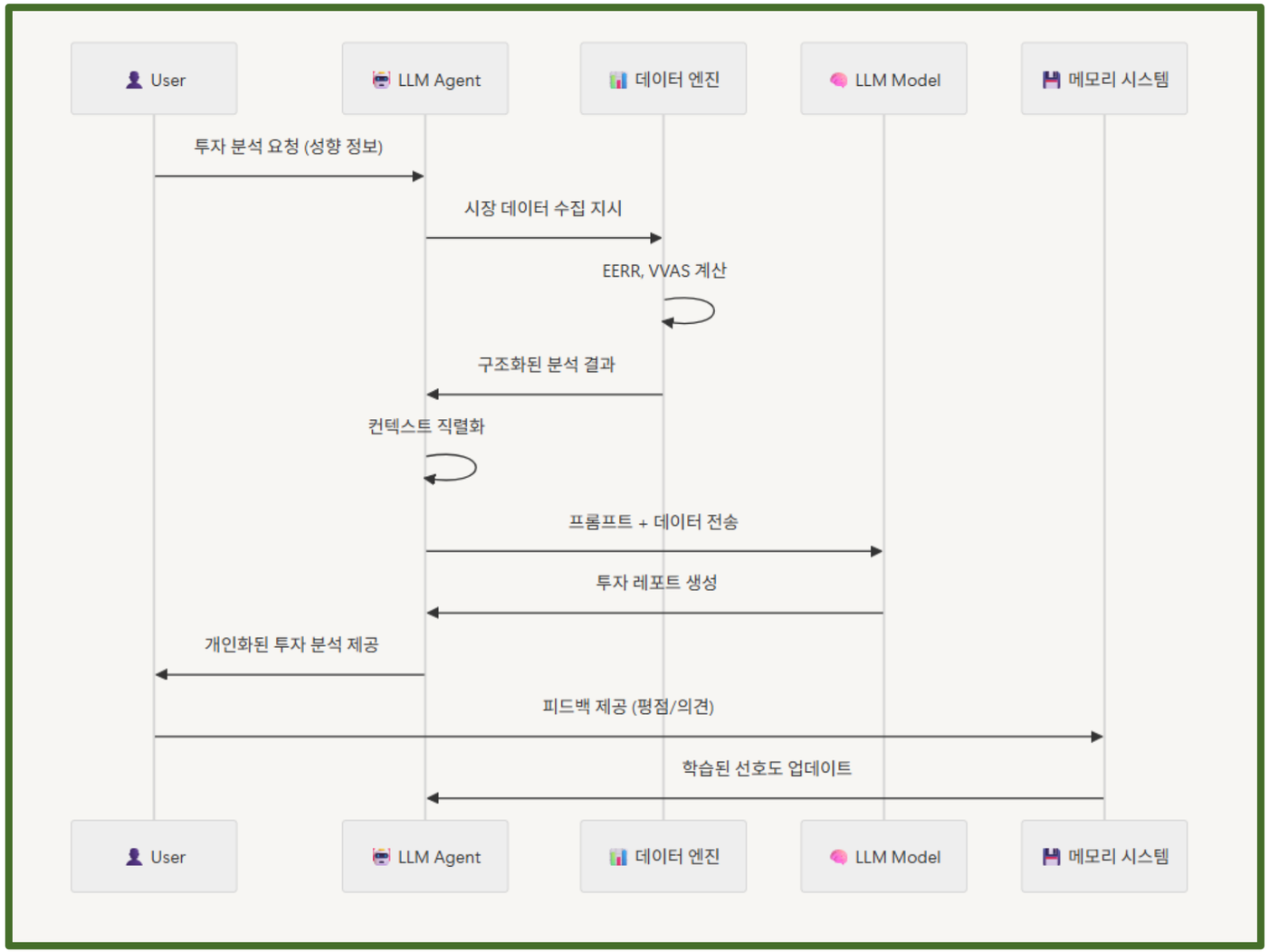
```
"""포트폴리오 자금 배분 계산"""
self.portfolio_allocation = AdvisorUtils.calculate_portfolio_allocation(
    self.recommendations,
    total_funds=total_funds,
    max_stocks=self.investor_profile['portfolio_size']
)
return self.portfolio_allocation
```

결과

```
===== 투자 어드바이저 요약 보고서 =====

최상위 투자 기회:
- Apple: 기회 점수 5.9/10, 추천: reduce
- Amazon: 기회 점수 5.8/10, 추천: hold
- Google: 기회 점수 5.1/10, 추천: hold

투자 행동별 종목 분류:
- 보유 (4개): Google, Microsoft, Meta, Amazon
- 비중 축소 (1개): Appl

추천 포트폴리오 자산 배
- Amazon: 27.7% ($2773.
- Google: 24.1% ($2414.
- Microsoft: 24.1% ($2
- Meta: 24.0% ($2403.4
```

```
===== 종목별 상세 추천 =====

투자 권고안: Apple

추천: ⚠️ 비중 축소
투자 점수: 4.8/10 ★★☆☆☆
투자 기회: 5.9/10
위험 수준: 🔴 높음

메시지:
Apple 비중 축소 고려. 일부 기술적 지표가 약세를 보이고 있습니다.

퇴출 전략:
현재 매도 압력이 높으니 호가창을 주시하며 단계적으로 매도하세요.

생성일시: 2025-05-25 18:52:34
```

Methods: LLM 기반 Agent



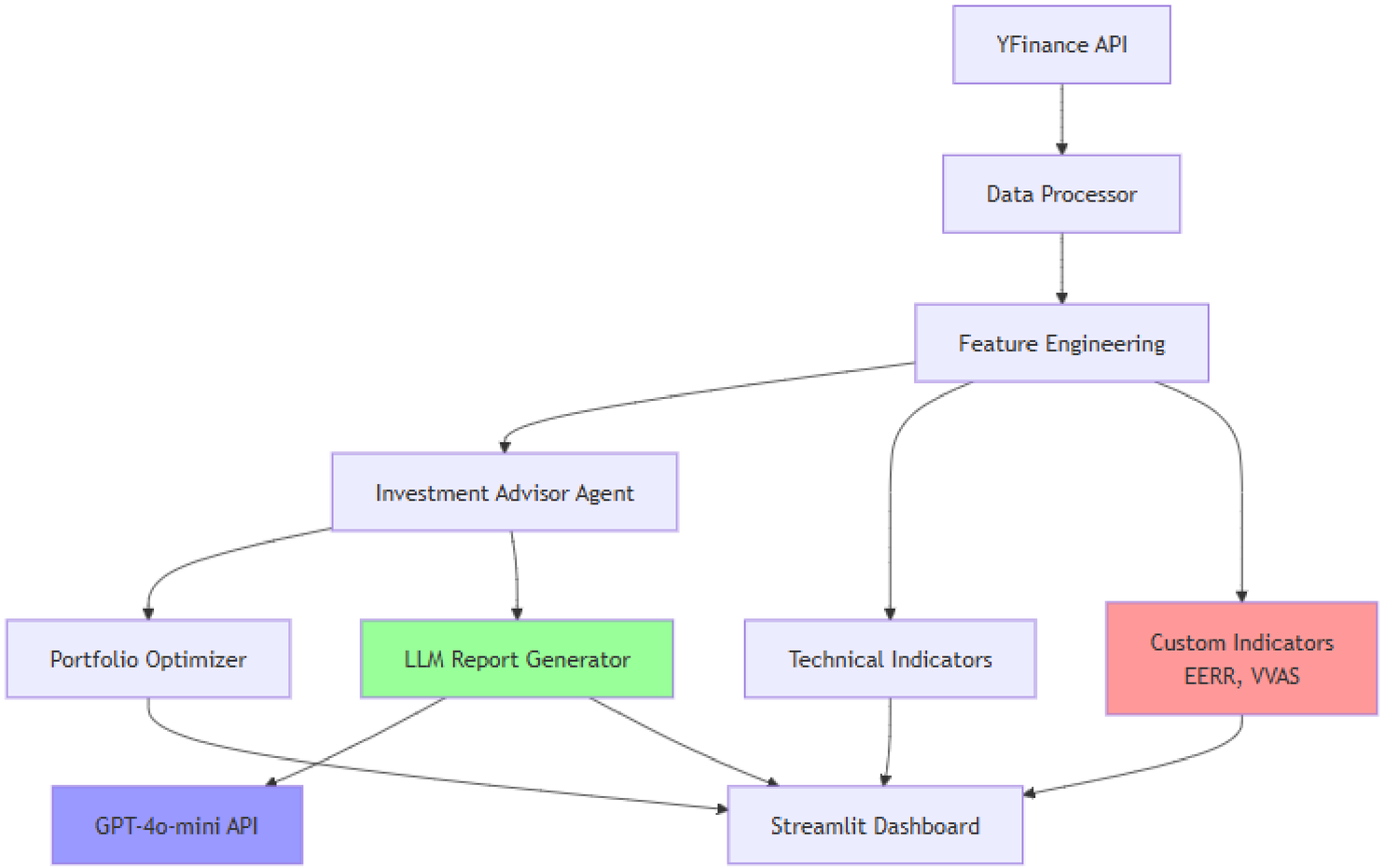
[Agent 시퀀스 다이어그램]

LLM-Based Agent 아키텍쳐

- 인식 : yfinance 데이터 수집 + 실시간 파싱
- 해석 : 멀티시그널 가중 평균 알고리즘
- 계획 : 포트폴리오 최적화 + 전략 수립
- 실행 : LLM 기반 동적 프롬프트 엔지니어링
- 학습 : 사용자 평점 + 백테스팅 결과

```python
class InvestmentAdvisorAgent:
    def __init__(self, stocks_data=None, feature_engineer=None):
        """
        투자 어드바이저 에이전트 초기화

        Args:
            stocks_data (dict): 종목별 데이터 딕셔너리
            feature_engineer (StockFeatureEngineer): 특성 엔지니어링 객체
        """
        self.stocks_data = stocks_data or {}
        self.feature_engineer = feature_engineer

        # 투자자 프로필/설정
        self.investor_profile = {
            'risk_tolerance': 'moderate',  # low, moderate, high
            'investment_horizon': 'medium',  # short, medium, long
            'portfolio_size': 5,  # 포트폴리오 종목 수
            'diversification_weight': 0.3,  # 분산투자 가중치
            'technical_weight': 0.4,  # 기술적 분석 가중치
            'market_trend_weight': 0.3,  # 시장 추세 가중치
        }
```

Methods: LLM 기반 Agent



[데이터 플로우]

Web Demo

Validation and Testing

⚙️ 통계적 검증 결과

```
validation_statistics = {
    "검증 기간": "2024-2025 (353일)",
    "급락 이벤트": "총 9회 식별",
    "EERR 성공 예측": "6.5회 (72.2%)",
    "데이터 신뢰성": "5년 히스토리컬",
    "종목별 균등성": "16-20개 데이터포인트"
}

performance_highlights = {
    "Microsoft": "100% 예측 (11회 중 11회 성공)",
    "Apple": "66.7% 양호 예측 (16회 중 11회 성공)",
    "Google": "50% 기준선 달성 (20회 중 10회 성공)"
}
```

```
============================================================
📊 로보어드바이저 성과 검증 리포트
============================================================

- 핵심 정확도 지표:
   EERR 예측 정확도: 72.2%
   VVAS 신호 정확도: 40.0%

- 종목별 상세 결과:
   Apple:
      - EERR: 66.7%
      - VVAS: 0.0%
      - 데이터 포인트: 16개
   Microsoft:
      - EERR: 100.0%
      - VVAS: 0.0%
      - 데이터 포인트: 11개
   Google:
      - EERR: 50.0%
      - VVAS: 40.0%
      - 데이터 포인트: 20개
```

Discussion & Conclusions

💡 **이론적 의의:**

├── EERR 지표 개발: Event-driven 투자 방법론의 새로운 패러다임

├── VVAS 지표 개발: 다변량 이상치 탐지를 통한 변동성 예측 기법

├── LLM-Finance 통합: 금융 도메인 특화 AI Agent 아키텍처

└── 적응적 신호 융합: 사용자 맞춤형 다중 신호 가중치 시스템

💡 **방법론적 기여:**

├── 히스토리컬 백테스팅 프레임워크 설계

├── 구조화된 동적 프롬프트 엔지니어링

└── 설명 가능한(XAI) LLM 기반 레포트 생성

➡️ 개인 투자자도 **기관 투자자 수준의 AI 기반 분석 도구** 활용할 수 있는 플랫폼 제안 !

Q & A