

CH16 – EX16~18

LLM 뉴스 요약, CNN 패턴 인식, Chronos 시계열 예측



LLM 기반
뉴스 감성분석



CNN 기반
패턴 인식



Chronos
시계열 예측

Example 16: LLM 뉴스 요약 트레이딩 – 요약 및 모티베이션

🎯 요약

목표: 뉴스 데이터 기반 감성 분석으로 트레이딩 우위 확보

- OpenAI GPT-4 모델로 테크 기업 관련 뉴스의 감성 지수 추출
- TSLA 관련 뉴스 데이터의 감성 트렌드를 트레이딩에 활용
- 롤링 방식의 시간별 감성 점수 계산 및 모니터링
- 실시간 감성 변화에 따른 포지션 자동 조정

🚀 핵심 기술 요소

🔧 기술: LLM (GPT-4)

📊 자산군: 미국 주식(TSLA)

📈 난이도: 중상

💡 모티베이션

왜 비전통적 데이터 소스를 활용하는가?

- 경쟁 우위 확보: 전통적 금융 지표를 넘어선 새로운 통찰력 발굴
- 시장 감정 포착: 이벤트와 발표에 대한 시장의 감정적 반응을 감지하여 투자 의사결정에 활용
- 선행 지표 역할: 향후 시장 트렌드나 하락에 대한 조기 감지 기능 제공
- 심리적 영향 정량화: 투자자들이 뉴스에 보이는 심리적 반응을 수치화하여 객관적 분석 가능

📈 기대 효과

- ✓ 시장 트렌드 변화 조기 감지 능력 향상
- ✓ 감정 기반 시장 반응에 선제적 대응 가능
- ✓ 정성적 정보의 정량적 활용으로 투자 프로세스 개선

Example 16: LLM 뉴스 요약 트레이딩 - 모델 및 포트폴리오 구성

LLM 모델 상세

OpenAI GPT-4 활용

- 대량의 비정형 텍스트 데이터 처리 가능
- 감성 분석에 특화된 프롬프트 설계
- JSON 형식 응답으로 일관성 확보

데이터 소스

Tiingo News API: TSLA 관련 뉴스 기사(2023.11.01 ~ 2024.03.01)

전처리: 날짜별 그룹화, 중복 기사 제거, 시간대별 집계

시간 단위 그룹화: 안정적 점수 생성 및 API 호출 최적화

GPT-4 프롬프트 구조

실제 사용된 프롬프트:

Article 1 title: {title}

Article 1 description: {description}

...

Review the news titles and descriptions above and then create an aggregated sentiment score which represents the emotional positivity towards TSLA after seeing all of the news articles. -10 represents extreme negative sentiment, +10 represents extreme positive sentiment, and 0 represents neutral sentiment. Reply ONLY with the numerical value in JSON format. For example, `{ "sentiment-score": 0 }`

포트폴리오 구성

모델 학습 타이밍:

리서치 노트북에서 필요시 온디맨드 방식으로 재계산

포트폴리오 리밸런싱:

시간별(Hourly) 감성 점수 업데이트마다 포지션 재조정

포트폴리오 비중 설정:

- 감성 점수 상승/플랫 + 미보유 → TSLA 100% 롱 포지션
- 감성 점수 부정적 + 하락세 + 미보유 → TSLA 100% 숏 포지션

```
if self._roc.current.value >= 0 and not self._tsla.holdings.is_long:
    self.set_holdings(self._tsla.symbol, 1)
elif (sentiment < 0 and self._roc.current.value < 0 and not self._tsla.holdings.is_short):
    self.set_holdings(self._tsla.symbol, -1)
```

CSV 데이터 구조

CSV 파일 구조:

- hour - 해당 시간(인덱스)
- sentiment - 시간별 집계 감성 점수 (-10~+10)
- volume - 해당 시간대 검토된 뉴스 기사 수

감성 변화율 계산:

RateOfChange(ROC) 지표로 감성 점수 방향성 파악

* 트렌드가 중요: 점수 자체보다 점수의 변화 방향이 더 중요한 신호

Example 16: LLM 뉴스 요약 트레이딩 – 트레이딩 로직

📈 감성 점수 판별 방식

트렌드 기반 트레이딩 규칙:

- 감성 점수가 평평하거나(flat) 최근 2개 점수가 증가 → **롱 포지션**
- 감성 점수가 음수이며 하락 추세 → **숏 포지션**
- 감성 추이 계산에 Rate of Change(RoC) 지표 활용

📊 데이터 구조

CSV 파일 구조(날짜별 시간당):

- **hour** - 인덱스 컬럼, 특정 시간 표시
- **sentiment** - 해당 시간의 뉴스 감성 점수(-10~+10)
- **volume** - 해당 시간 분석된 뉴스 기사 수

✎ 백테스트 설계

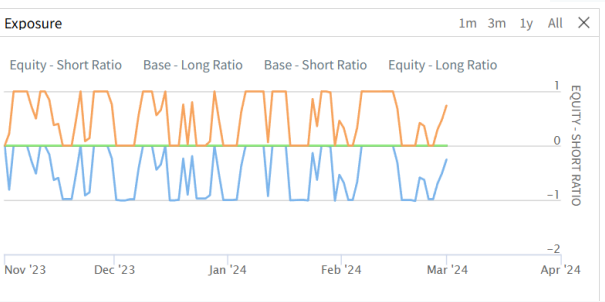
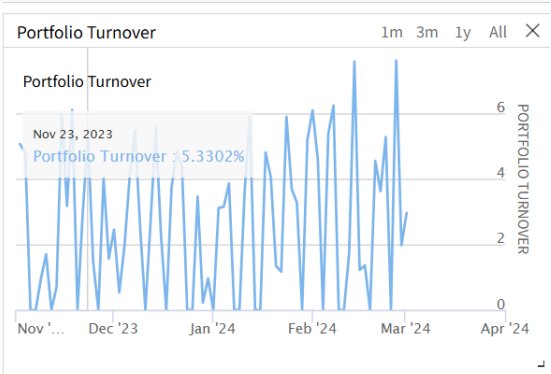
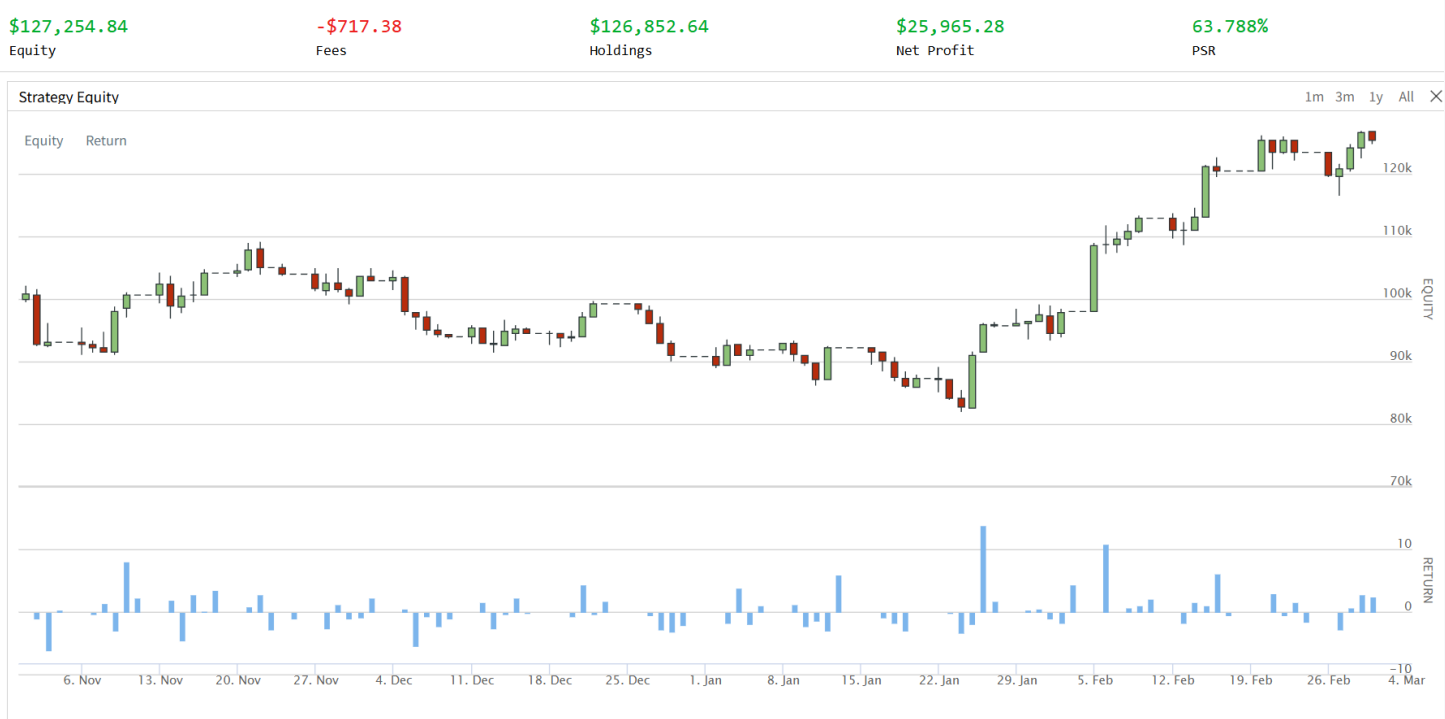
실거래 환경 시뮬레이션:

- 시장 개장 후 매 시간마다 감성 신호 재확인
- Object Store에서 해당 일자의 CSV 파일 자동 로딩
- 업데이트된 감성 데이터에 따라 트레이딩 결정
- 감성 점수와 주가 사이 잠재적 시차 고려

🏆 성과 측정 및 비교

- 📊 알고리즘 **Sharpe** 비율: 1.695 (위험 조정 수익률 우수)
- 💰 벤치마크(**TSLA buy&hold**): -0.06 Sharpe 비율
- 📈 월별 수익 분석: 뉴스 변동성 높은 달에 더 높은 수익률
- 🔗 감성 모니터링: 실시간 감성 데이터로 포지션 적시 조정 가능

Example 16: LLM 뉴스 요약 트레이딩 – 트레이딩 결과



핵심 수익/위험 지표		
항목	값	설명
PSR	63.788%	Profit-to-Stress Ratio. 수익 대비 리스크 수준 (높을수록 좋음)
Sharpe Ratio	1.883	리스크 대비 초과 수익. 일반적으로 1.0 이상이면 우수
Sortino Ratio	3.419	하방 리스크만 고려한 샤프비율. 매우 우수
Treynor Ratio	-1.835	배타 기반 위험 조정 수익률. 음수는 시장보다 못했음을 의미
Information Ratio	1.244	벤치마크 초과 수익률 대비 변동성. 1 이상이면 우수
Alpha	0.77	시장 대비 초과 수익률. 양수면 전략의 우수함 의미
Beta	1.19	시장 민감도. 1보다 크면 시장 변동성보다 큼

Example 17 — CNN 기반 헤드&숄더 패턴 트레이딩

🎯 요약

1차원 CNN(합성곱 신경망)을 활용해 헤드앤숄더(H&S) 트레이딩 패턴을 감지합니다. 합성 데이터로 모델을 훈련시키고 외환 시장(USD/CAD)에서 실제 패턴을 식별하여 매매 신호를 생성합니다.

💡 모티브이션

- 기술적 분석을 통한 시장 심리와 집단 행동 예측
- 규칙 기반 매매 결정으로 감정적 판단 배제
- 패턴 기반 반복적 매매 기회 포착
- 외환 시장의 높은 연속성과 글로벌 이벤트 반응성 활용

🤖 패턴 감지 모델

입력: 25개 종가데이터 포인트

출력: 패턴 존재 확률 (0~1)

- class 0: 패턴 없음
- class 1: 헤드앤숄더 패턴 존재

사용 모델: **1D-CNN**

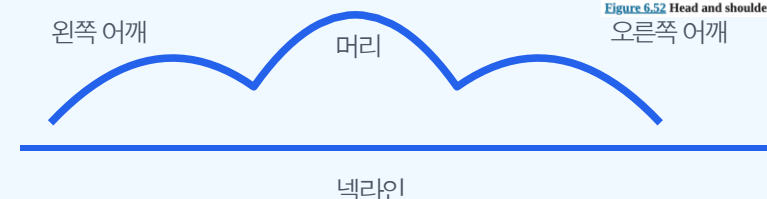
📈 헤드앤숄더 패턴 구조

상승 추세에서 하락 추세로 전환을 예측하는 차트 패턴:

- 왼쪽 어깨: 가격 상승 후 하락
- 머리(헤드): 더 높은 정점 형성 후 하락
- 오른쪽 어깨: 머리보다 낮은 상승 후 하락
- 넥라인: 어깨들의 저점을 연결한 지지선



Figure 6.52 Head and shoulder pattern and the typical entry timing.



</> 트레이딩 접근법

📊 CNN으로 패턴 자동 감지하여 주관적 해석 배제

📅 다양한 시간대 가격 데이터 다운샘플링 후 패턴 탐색

📷 패턴 감지 시 10,000 USD 숏 포지션 진입

📅 포지션 10일 유지 후 자동 청산

Example 17: CNN 기반 헤드&숄더 패턴 트레이딩 - 모델 아키텍처

1D-CNN 모델 구조

순차적 모델 레이어 구성:

1. 입력층: 25개 데이터 포인트 (각 특성은 독립 채널)
2. 1D 합성곱 층: 32개 필터, 커널 크기 5, ReLU 활성화
3. 최대 풀링 층: 풀 크기 2, 복잡도 감소 및 스케일 불변성
4. 플래튼 층: 합성곱 출력을 1차원으로 변환
5. 출력층: 시그모이드 활성화 함수 (이진 분류)

Keras 구현 코드

```
model = Sequential()
model.add(Input(shape=(X_train.shape[1], 1)))
model.add(Conv1D(filters=32, kernel_size=5, activation='relu'))
model.add(MaxPooling1D(pool_size=2))
model.add(Flatten())
model.add(Dense(1, activation='sigmoid'))
```

예측 결과 해석

출력: 0~1 확률값

- 0: 패턴 없음 (no pattern)
- 1: 헤드&숄더 패턴 존재 (pattern present)
- 0.5 이상일 경우 트레이딩 신호로 사용

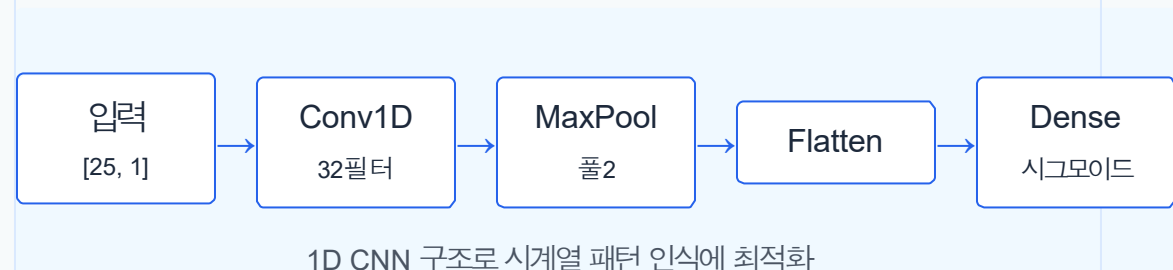
훈련 데이터 생성

- + 긍정 샘플: 100,000개 합성 H&S 패턴
랜덤 노이즈 포함 25 데이터 포인트로 구성된 패턴
- 부정 샘플: 100,000개 랜덤워크 무작위 시계열 데이터로 비패턴 학습
- 표준화: 각 샘플별 데이터 평균 0, 표준편차 1로 정규화

모델 성능

테스트 정확도: 99.9%
25개 데이터 포인트만으로 매우 높은 인식률
80/20 분할 (훈련/테스트 데이터셋)

아키텍처 시각화



Example 17: CNN 기반 헤드&숄더 패턴 트레이딩 – 트레이딩 로직 상세

🔍 패턴 탐지 과정

1. 일별 USD/CAD 종가데이터 수집(2019-2024년)
2. 다양한 윈도우 크기 설정 (25~100일, 10일 단위 증가)
3. 각 윈도우를 25 데이터 포인트로 다운샘플링
4. 표준화(정규화) 과정을 통한 CNN 입력 형태 변환
5. CNN 모델을 통한 패턴 존재 확률 예측

⚙️ 다운샘플링 원리

헤드&숄더 패턴은 다양한 기간에 걸쳐 형성되지만, CNN 입력은 항상 25개 포인트가 필요함

📊 성과 인사이트

- 📈 Holding period 증가 → Sharpe ratio 상승
- 📉 Confidence threshold 증가 → 거래 횟수 감소
- ✅ 모든 파라미터 조합에서 수익성 확인

📈 트레이딩 실행 로직

진입 조건:
패턴 감지 확률 $\geq 50\%$ → USD 10,000 숏 포지션 진입

청산 조건:
포지션 진입 후 10일 경과시 자동 청산

⚙️ 백테스트 파라미터

- 📄 max_size: 100 (최대 윈도우 크기)
- ↔ step_size: 10 (윈도우 크기 증가 단위)
- 📊 confidence_threshold: 0.8 (80% 신뢰도)
- 📅 holding_period: 10 (포지션 유지 일수)

Example 17: CNN 기반 헤드&숄더 패턴 트레이딩 - 구현 인사이트 및 합성 데이터

합성 데이터 생성

합성 H&S 패턴 10만개를 생성하여 CNN 학습 데이터로 활용

- v1, p1, v2, p2, v3, p3, v4와 같은 핵심 포인트 생성
- 각 포인트 간 3개의 내삽점 추가(총 25 포인트) 무
- 작위 노이즈를 가미하여 패턴 다양성 확보
- 랜덤 워크 기반 부정 샘플 10만개 생성



Figure 6.52 Head and shoulder pattern and the typical entry timing.

</> 합성데이터 구현 코드

```
np.random.seed(1)
ref_count = 100_000
v1 = np.array([0] * ref_count) + 0.02 * norm.rvs(size=(ref_count,))
p1 = np.array([1] * ref_count) + 0.2 * norm.rvs(size=(ref_count,))
v2 = v1 + 0.2 * norm.rvs(size=(ref_count,))
v3 = v1 + 0.2 * norm.rvs(size=(ref_count,))
p3 = p1 + 0.02 * norm.rvs(size=(ref_count,))
p2 = 1.5 * np.maximum(p1, p3) + abs(uniform.rvs(size=(ref_count,)))
v4 = v1 + 0.02 * norm.rvs(size=(ref_count,))
```

구현 단계

- 1 합성 입력 데이터셋 생성 (scipy.stats 활용 정규분포 노이즈)
- 2 데이터셋 표준화 및 훈련테스트셋 분할 (80%/20%)
- 3 Sequential CNN 모델 구축 및 훈련 (10 에포크, 배치 크기 16)
- 4 모델 평가 및 Object Store에 저장 후 백테스트 환경에서 활용

💡 핵심 인사이트

다운샘플링 기법

다양한 윈도우 크기(25~100일)를 25 포인트로 다운샘플링하여 H&S 패턴 탐지 - 길이가 다른 패턴도 인식 가능

표준화 전처리

입력 데이터의 표준화(평균 0, 표준편차 1)로 모델 성능 안정화 및 학습 효율성 증대

파라미터 최적화

confidence_threshold(0.3~0.9), holding_period(2~10일) 민감도 테스트를 통한 최적 파라미터 도출

실증 결과

모든 파라미터 조합에서 수익성 확인, 홀딩 기간 증가시 샤프 비율 향상, 높은 confidence_threshold에 서 거래 수 감소

Example 17: CNN 기반 헤드&숄더 패턴 트레이딩 - 결과



지표명	값	지표 의미 및 해석
Net Profit	1.166%	최종 수익률로, 초기 자본 10만 → 약 101,166원. 수익은 긍정적이나 미미함.
Sharpe Ratio	-15.053	위험 대비 초과 수익률. 음수는 변동성에 비해 성과가 매우 나쁨을 의미. 전략 위험 대비 성과 부정적.
Sortino Ratio	-26.96	하방 리스크만 고려한 성과. 음수이므로 손실 위험이 수익보다 큼.
Expectancy	2.227	1회 거래당 기대 수익. 2.227은 긍정적이며, 기대값이 양수.
Win Rate	78%	전체 거래 중 수익 실현 비율. 매우 높은 수치로 전략 신뢰도 높음.
Loss Rate	22%	손실 비율. Win Rate와 합쳐서 해석 시 전략 일관성 우수.
Profit-Loss Ratio	3.15	이익:손실 비율. 이익이 손실보다 3배 이상 커서 리스크-리워드 비율 양호.
Drawdown	0.200%	최대 낙폭. 매우 작음, 전략의 안정성 높음.
Drawdown Recovery	304	낙폭 회복에 걸린 시간. 상대적으로 길다면 복원력 낮을 수 있음.
Alpha	-0.022	시장 초과 수익. 음수로 시장 대비 부진.
Tracking Error	0.171	벤치마크 대비 변동성 차이. 중간 정도 수준.
Information Ratio	-0.754	추적오차 대비 초과 수익률. 음수는 일관되지 못한 전략.
Compounding Annual Return	0.221%	연복리 수익률. 실질 장기 투자 수익이 거의 없음.

Example 18: Amazon Chronos 모델 - 요약 및 모티브이션

🎯 요약

사전학습된 HuggingFace "amazon/chronos-t5-tiny" 모델을 활용해 시장 내 가장 유동성 높은 5개 자산의 미래 성과를 예측하고, 이를 기반으로 Sharpe ratio를 최대화하는 포트폴리오 최적화를 수행합니다. 3개월마다 최신 예측과 시장 상황에 맞춰 포트폴리오를 리밸런싱합니다.

💡 모티브이션

- Amazon Chronos 모델의 시계열 예측 능력과 Sharpe ratio 기반 포트폴리오 최적화 결합
- 사전학습 모델 활용으로 직접 훈련 시간 단축 및 자원 절약
- GPU 등 강력한 하드웨어에서 확장성 보장, 대규모 데이터셋 처리 가능
- ARIMA나 LSTM과 같은 전통적 모델보다 금융 시장 예측에 특화된 성능

🔧 활용 기술

주요 구성 요소:

- 시계열 예측: Amazon Chronos T5-tiny
- 포트폴리오 최적화: SciPy minimize 기반 Sharpe ratio 최대화
- 모델 구현: HuggingFace Transformers 파이프라인

📈 모델 특징

입력 데이터: 일별 종가데이터

출력 데이터: 향후 3개월(63영업일) 동안의 일별 종가예측값

대안 모델과의 비교:

- ✓ ARIMA: 선형 패턴 데이터에 효과적이나 복잡한 금융 시계열에 한계
- ✓ Prophet(Facebook): 계절성과 트렌드에 강하나 금융특화기능 부족
- ✓ LSTM/SVM: 특성 추출 능력은 우수하나 학습 복잡성과 자원 요구량 높음

</> 전략 구현

본 예제는 두 가지 전략으로 구현:

- ⚙ 기본 모델: 사전학습된 Chronos 모델을 그대로 활용
- ⚙ 파인튜닝 모델: 매월 첫 거래일에 모델을 재학습하여 예측 정확도 향상

Example 18: Amazon Chronos 모델 설명 및 타 모델 비교

Amazon Chronos 모델

모델 특성

- 입력: 1년간 일별 종가데이터 출
- 력: 63일(3개월) 미래 가격 예측
- 모델 유형: HuggingFace "amazon/chronos-t5-tiny"

모델 구성 방식

- Base 모델: 사전학습된 기본 모델 사용
- Fine-tuned 모델: 실제 데이터로 재학습(월별)
- context_length: 126(6개월) 최적화

주요 장점

- ⚡ 금융 시장 예측에 특화된 최신 시계열 모델
- 🕒 신속한 배포 및 확장성 (학습 불필요)
- 💻 GPU 활용한 대규모 데이터셋 처리 가능

타 모델과의 비교

모델	특징	강점	약점
ARIMA	시계열 예측 모델	선형 패턴 포착 우수	정상성 가정 필요
Prophet	Facebook 개발	계절성/추세 처리	복잡한 패턴 한계
LSTM	순환신경망	장기 의존성 포착	학습시간 길고 자원 많 이 소모
SVM	지도학습 모델	고차원 데이터 처리	대규모 데이터 처리 어려 움
XGBoost	그래디언트 부스 팅	테이블형 데이터 처리 우수	시계열 특화 부족
Amazon Chronos	사전학습 특화 모 델	금융 시계열 특화, GPU 확장성	데이터 누락시 성능 저 하

모델 선택 고려사항

최적의 예측 모델 선택 기준:

- 데이터 특성: 정상성/비정상성, 계절성 여부
- 예측 정확도 요구 수준
- 계산리소스 가용성 (GPU 등)
- 배포 및 실시간 업데이트 필요성
- 다양한 모델 테스트 및 성능 비교 권장

Example 18: Amazon Chronos 모델 – 포트폴리오 최적화

🕒 포트폴리오 최적화 접근법

목표: 샤프 비율(Sharpe Ratio) 최대화

미래 수익과 리스크 간 최적 균형을 찾는 최적화 문제

최대화: $(R_p - R_f) / \sigma_p$

R_p : 포트폴리오 수익률 R_f : 무위험 수익률 σ_p : 포트폴리오 리스크

⚖️ 제약 조건

- 모든 자산비중 합계 = 1 (100%)
- 각 자산별 비중 ≥ 0 (공매도 없음)
- 3개월(분기)마다 리밸런싱

최적화는 SciPy의 *minimize* 함수 사용, 목적함수에 -1을 곱해 변환

</> 최적화 코드 구조

```
def _sharpe_ratio(self, weights, returns, risk_free_rate): # 연간화된 수익률과 공분산 계산
    mean_returns = returns.mean() * 252
    cov_matrix = returns.cov() * 252
```

```
    # 샤프 비율 계산
    portfolio_return = np.sum(mean_returns*weights)
    portfolio_std = np.sqrt(np.dot(weights.T, np.dot(cov_matrix, weights)))
    sharpe = (portfolio_return-risk_free_rate)/portfolio_std
```

```
    # 최소화를 위해 음수 반환
    return -sharpe
```

```
def _optimize_portfolio(self, equity_curves):
    returns = equity_curves.pct_change().dropna()
    num_assets = returns.shape[1]
    initial_guess = num_assets * [1. / num_assets]
```

```
    # 포트폴리오 최적화 (샤프 비율 최대화)
    result = minimize(
        self._sharpe_ratio, initial_guess,
        args=(returns, self.risk_free_rate), method='SLSQP',
        bounds=tuple((0, 1) for _ in range(num_assets)), constraints=({'type': 'eq',
```

Example 18: Amazon Chronos – 백테스팅 결과 (01 – base model)



지표명	값	지표 의미 + 값 해석
Net Profit	237.263%	총 수익률 초기 10만 → 337,263.17원. 수익성 매우 우수.
Sharpe Ratio	0.726	변동성 대비 초과 수익률. 양호한 수준, 효율적인 전략.
Sortino Ratio	0.788	하방 리스크 대비 수익률. 양수이며, 위험 대비 수익 구조 안정적.
Win Rate	77%	거래 중 수익 실현 비율. 정확도 높음, 전략 신뢰 가능.
Loss Rate	23%	손실 발생 비율. 낮은 손실 빈도와 높은 Win Rate는 좋은 조합.
Profit-Loss Ratio	0.80	평균 이익/손실 비율. 손실 대비 이익이 적지만 Win Rate로 보완.
Drawdown	47.700%	최대 낙폭. 크게 하락한 적 있음, 리스크 관리 필요.
Compounding Annual Return	26.031%	연복리 수익률. 장기 운용 시 고수익 기대 가능.
Alpha	0.059	시장 초과 수익률. 긍정적, 시장 대비 우수한 성과.
Expectancy	0.380	거래당 기대 수익. 양수이며, 일관된 수익 구조 가능성 있음.