| **Graduation Project Progress Report 1** |
| --- |

| **Title** | Personalized Invest AI |
| --- | --- |
| **Summary**  **(in 150 words)** | Minerva is an AI-driven personal asset management platform that leverages large language models (LLMs) and time-series forecasting to deliver personalized investment advice. In response to the growing number of individual investors, Minerva aims to provide a Private Banker-like experience using advanced AI technologies. The system begins by analyzing a user’s financial goals and investment preferences through a natural language survey processed via NAVER Clova Studio. Based on the extracted investment profile, it recommends optimized portfolios using vector-based similarity matching between user and stock embeddings. Time-series models such as LSTM and GRU are integrated to forecast future stock performance, allowing the AI to offer proactive insights. A multi-agent framework ensures reliable and specialized responses by dividing tasks between a user-centric agent, a market expert agent, and a coordinating agent. The platform is delivered via a web interface, providing intuitive, real-time financial guidance for everyday investors. |
| **Advisor** | 김현준 교수님 |
| **Period** | 2025. 03. 01 ~ 2025. 12. 20 |

| **Name** | **Department** | **Student ID** | **Hanyang Email** | **Phone** |
| --- | --- | --- | --- | --- |
| 황병웅 | 심리뇌과학과 | 2022054084 | quddnd1214@hanyang.ac.kr | 010-2766-1743 |
| 김선권 | 심리뇌과학과 | 2021071394 | asdf4596@hanyang.ac.kr | 010-5537-4280 |

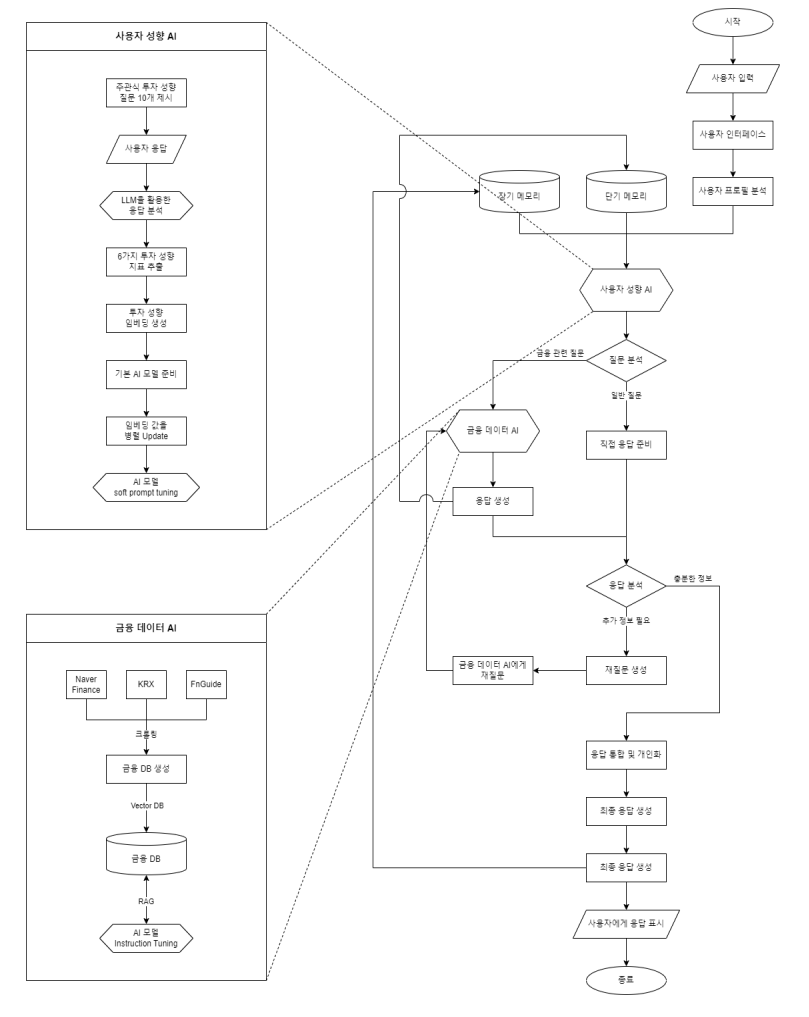
**College of Data Science**

**Hanyang University**

**Contents**

1. Progress to Date

1-1. 서비스 알고리즘 흐름도 작성



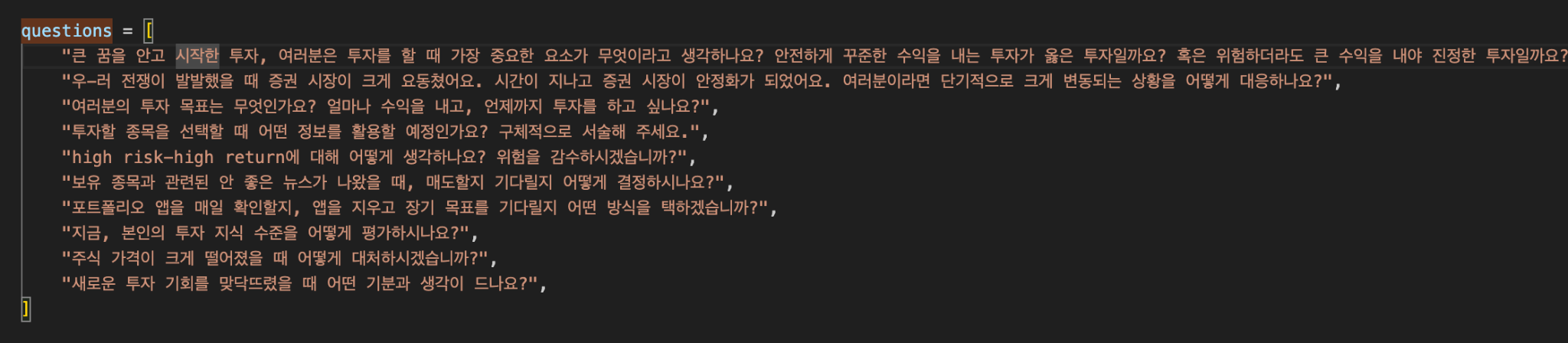
1-2 Progress coding

1. **사용자 성향 AI**

시스템 시뮬레이션은 PyQt5 기반의 데스크탑 GUI 애플리케이션으로 구현되었으며, Clova Studio API를 활용하여 사용자의 자유 서술형 응답을 자동으로 분석하고 정량화하는 기능을 포함하고 있습니다. 이번 실행은 본격적인 서비스 개발에 앞서, 전체 기능이 정상적으로 동작하는지를 사전에 점검해보기 위한 테스트 목적입니다.

설문 문항 구성 및 투자 성향 파악

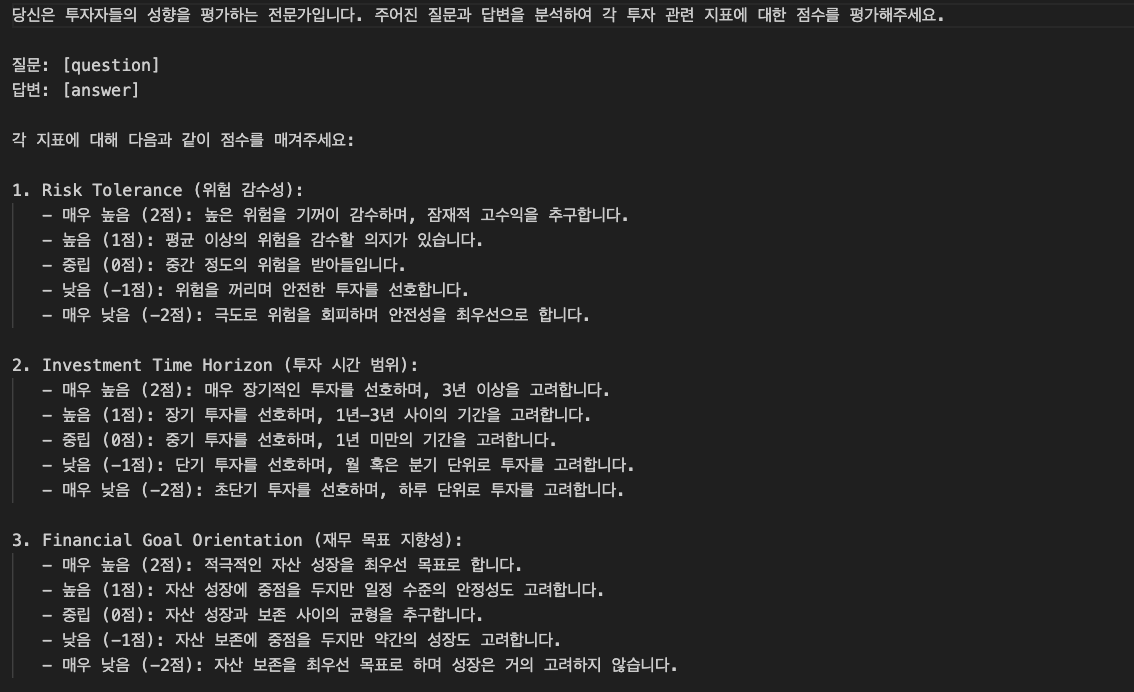
* 총 10개의 서술형 질문을 기반으로 사용자의 투자 성향을 심층적으로 파악할 수 있는 구조 설계.
* 각 질문은 투자 철학, 정보 수용 태도, 위험 선호도, 감정 반응 등을 유도하도록 설계되었으며, 객관식 설문 대비 자유 응답 기반, 높은 심리 분석을 가능하게 함.

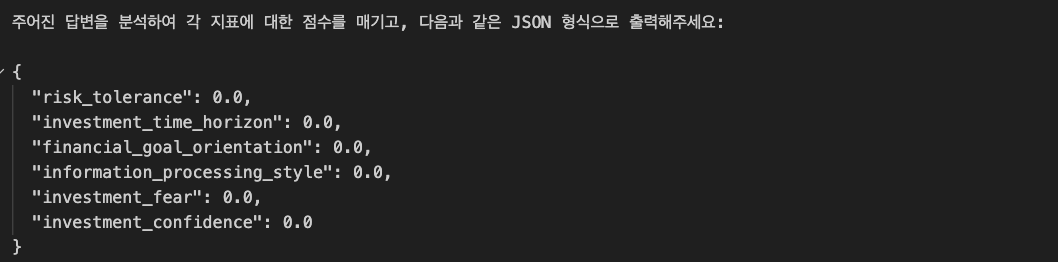


Clova Studio API를 활용한 자동 점수화

* 사용자의 각 응답에 대해, 프롬프트 템플릿(prompt\_survey-score.txt)을 기반으로 Clova API에 메시지를 보내고, JSON 형식으로 6가지 항목의 점수를 반환받음.
* 해당 점수는 항목별로 누적되어 전체 투자 성향을 정량적으로 평가하는 데 활용됨.

**prompt\_survey-score.txt :**

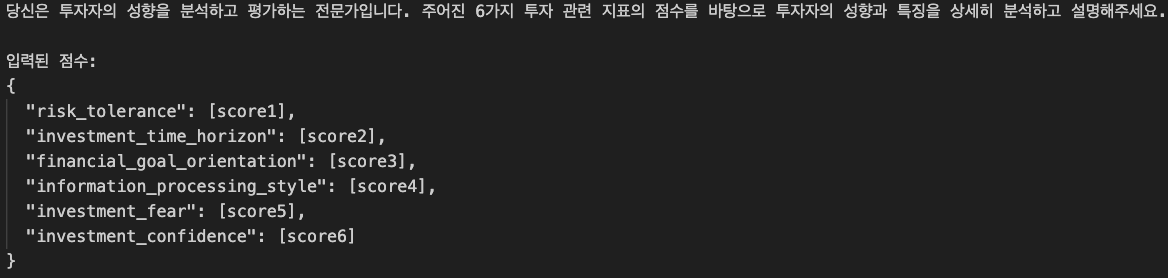


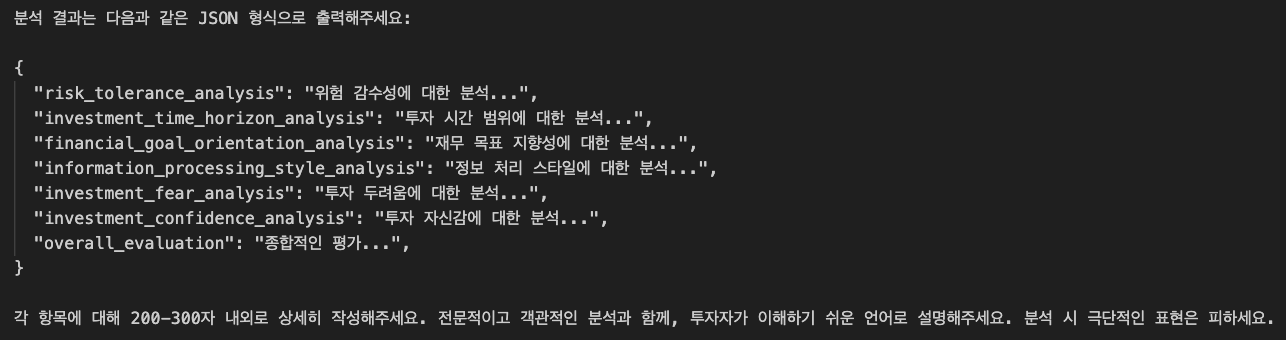


최종 분석 결과 도출 과정

* 모든 문항 응답이 완료되면 누적된 점수를 분석 프롬프트(prompt\_survey-analysis.txt)에 삽입하여, Clova API로부터 종합적인 투자 성향 분석 결과를 생성.
* 항목별 해석과 함께, 전반적인 평가가 포함된 사용자 맞춤형 결과 리포트를 생성하여 GUI에 출력.

**prompt\_survey-analysis.tx**t :



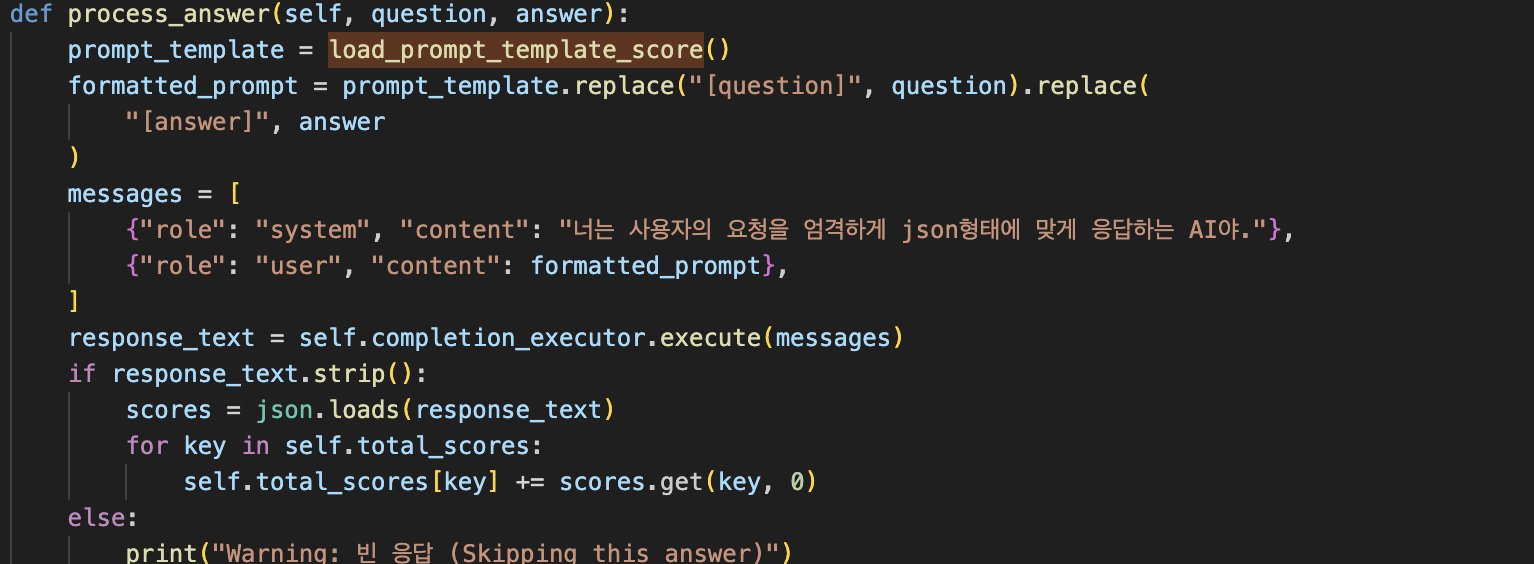


**user\_survey.py :**



**def process\_answer** : 사용자의 응답을 기반으로 clova llm에 점수 요청 함수

* prompt\_survey-score.txt 템플릿을 이용해 사용자 답변을 llm에 전달.
* llm은 json 으로 6개 항목에 대한 점수를 응답 → total\_scores 에 누적시킴.



**def analyze\_results** : 누적된 투자 성향 점수 기반 clova llm 호출, 종합 분석 결과 생성 및 출력

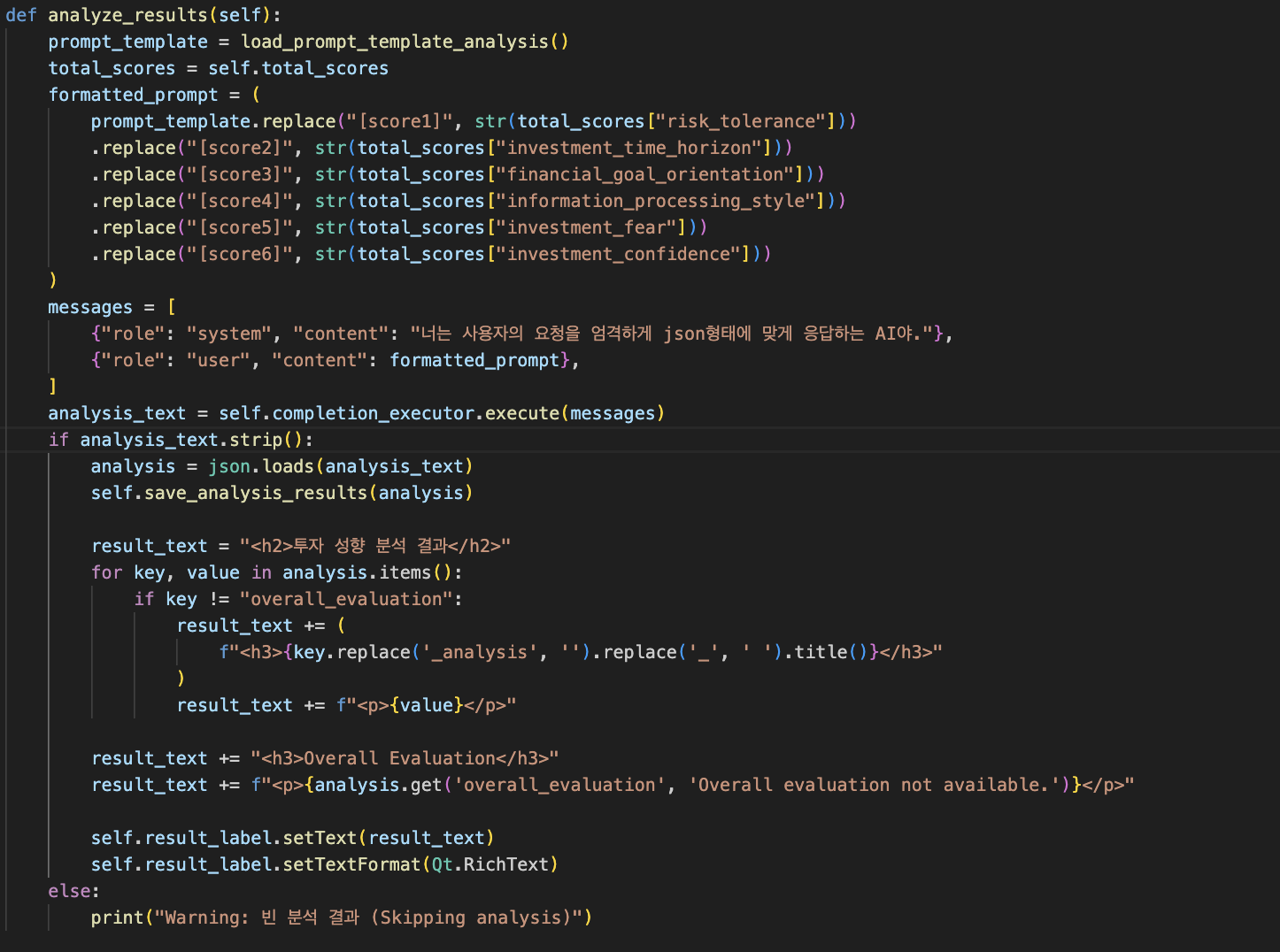
- def process\_answer 로 부터 나온 total\_scores를 prompt\_survey-analysis.txt 템플릿에 채워 넣음.

- clova studio api에 사용자 점수 기반 메세지 전달, 종합 분석 요청

- messages 리스트를 구성하여 execute() 함수에 전달, 첫번째 message는 응답 포맷을, 두번째 message는 실제 요청 내용을 명시함.

- execute() 는 clova studio api에 post 요청을 보내고 analysis\_text를 반환함.

- llm으로 부터 받은 json 형식 분석 결과(analysis\_text) 파싱 (json 결과를 딕셔너리 형태로 변환하여 이후 GUI 출력 및 파일 저장에 활용 가능하도록 준비)

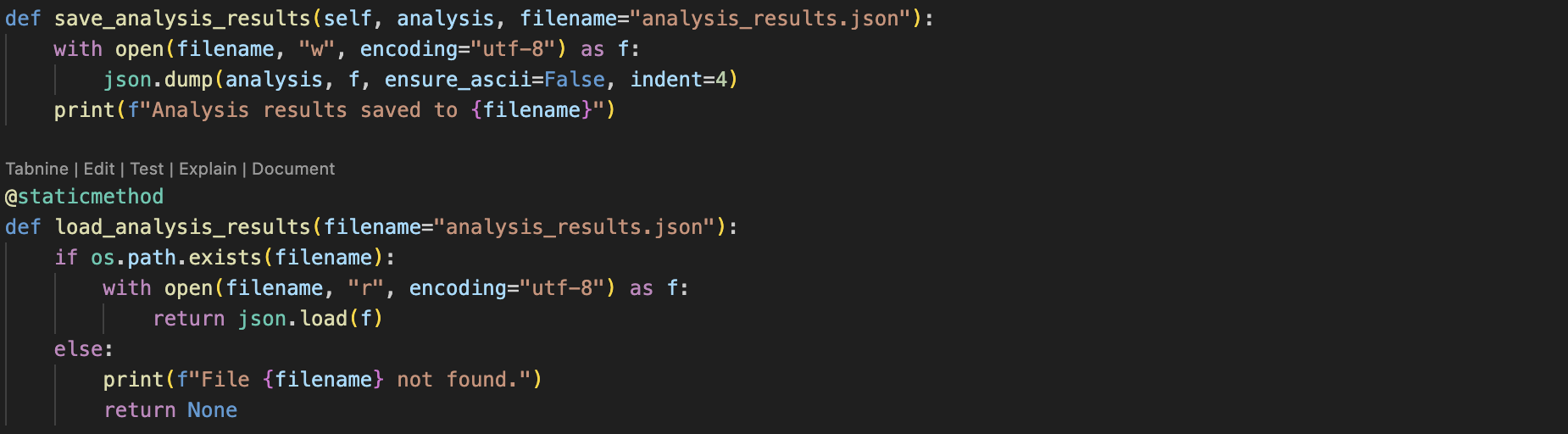


: 결과 저장 및 활용

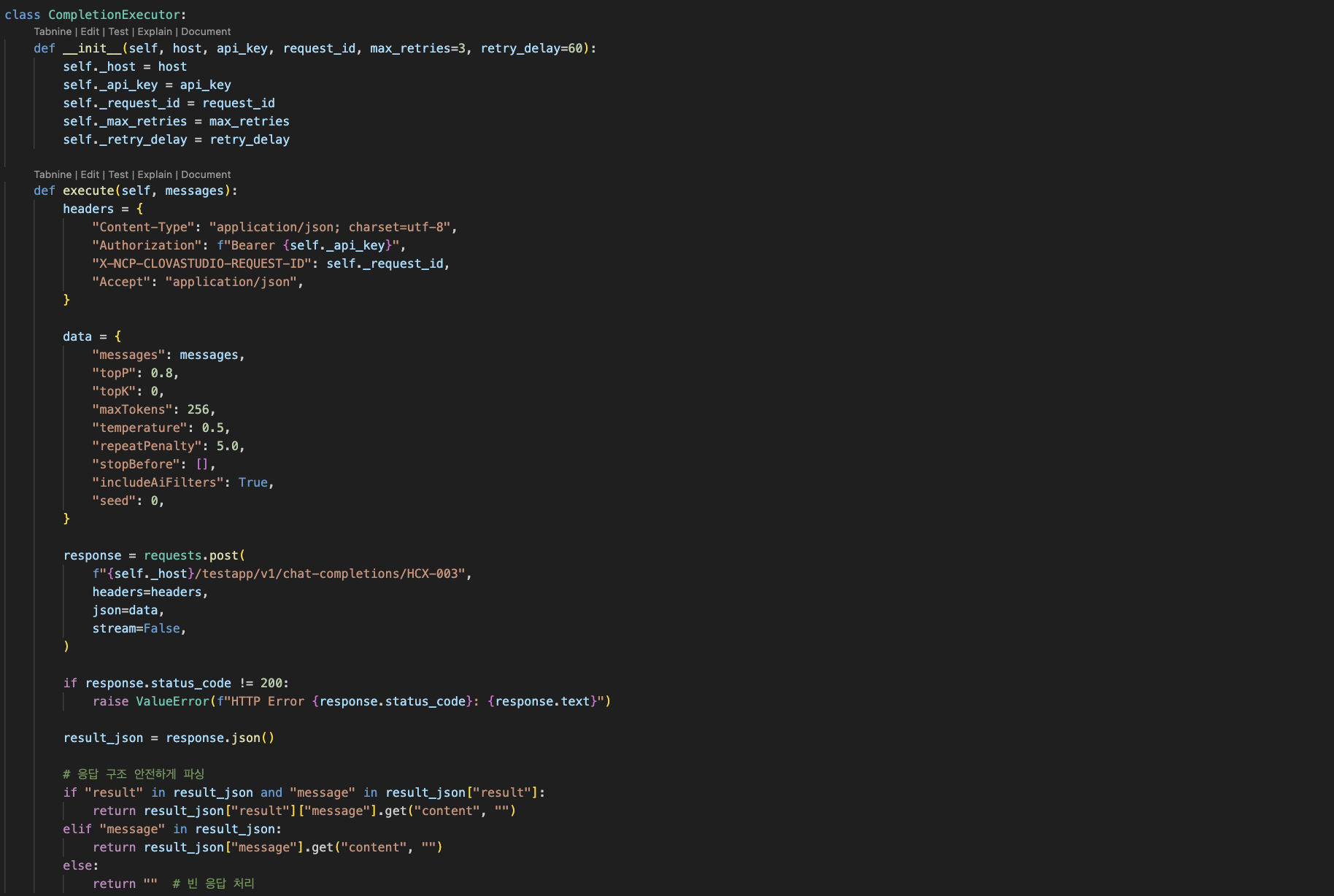
- 분석 결과는 analysis\_results.json 파일로 저장되며, 이후 사용자 맞춤형 투자 조언 생성에 활용됨.

- 저장된 결과는 다음 단계의 AI 시스템 ([main.py](http://main.py)) 에서 프롬프트 자동 완성 시 사용

- 사용자의 성향 기반 투자 조언 시스템과 자연스럽게 연동될 수 있는 구조를 확보.



**CompletionExecutor** : clova studio의 gpt 기반 api (HCX-003) 에 프롬프트 전송, 응답 처리

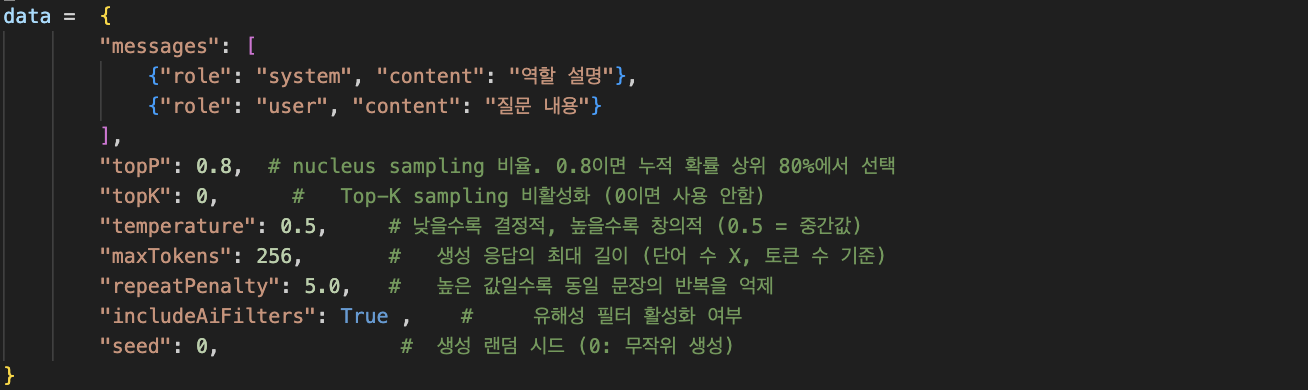


clova studio api : naver cloud에서 제공하는 gpt 기반 언어 모델 api 서비스, HCX-003은 clova의 고성능 챗 모델로, 대화형 프롬프트를 기반으로 텍스트 생성을 수행함.

authorization : bearer 방식의 api 키 제공

X-NCP-CLOVASTUDIO-REQUEST-ID : : 요청 추적용 고유 ID 지정 (UUID 권장)

요청 데이터 형식 :



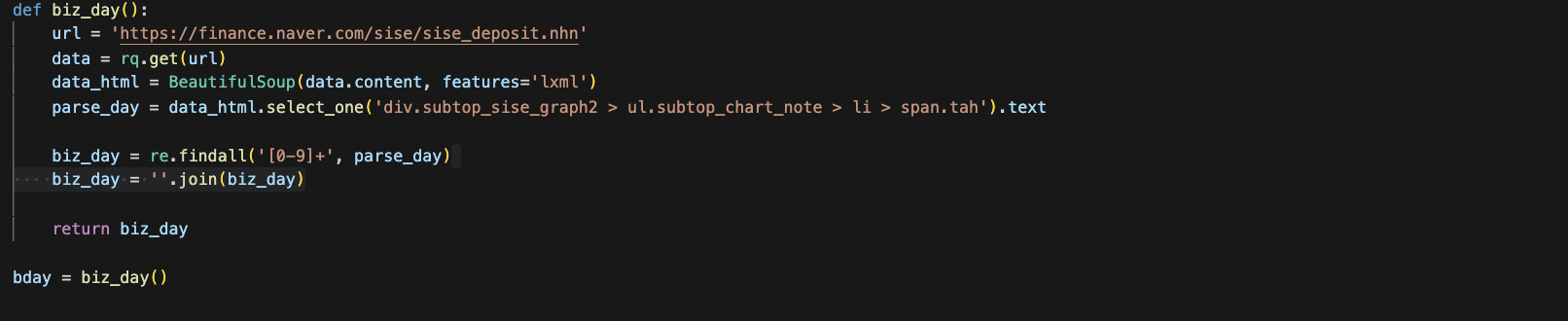
**1-2) Data Crawling**

1. **Libray import**



1. **영업일 가져오기**

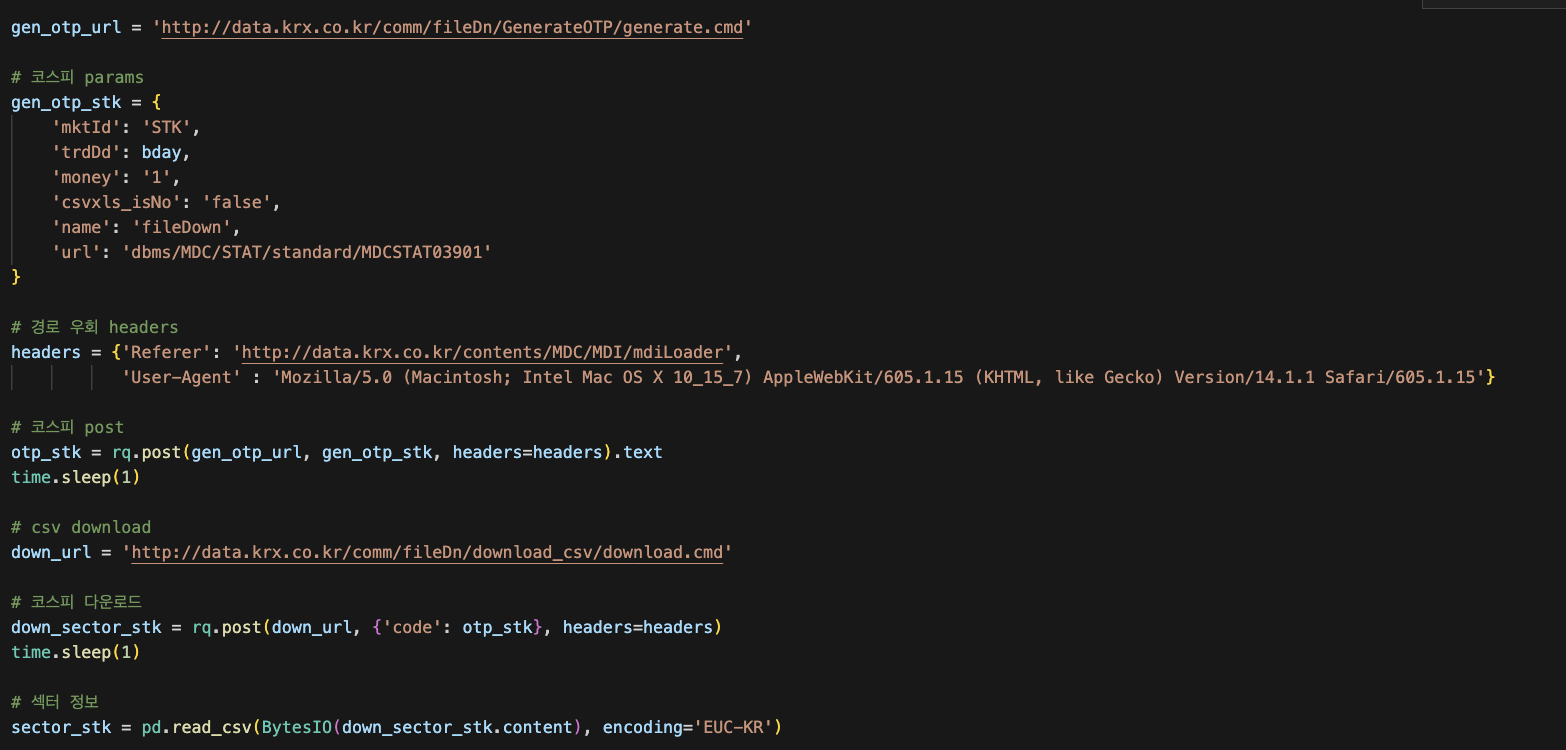
:네이버 금융의 증시자금동향 페이지에서 최신 영업일 날짜 이용



1. requests 모듈 사용 - HTML 가져옴
2. HTML 데이터를 BeautifulSoup으로 파싱 HTML 구조로 변환 (paser = lxml)
3. CSS 선택자를 이용해 기준일을 포함하는 요소 찾고 text 추출

**3. Ticker List**

: 한국거래소(KRX)의 OTP 기반 API를 활용하여 KOSPI와 KOSDAQ의 종목별 주요 지표 및 산업, 섹터 정보를 수집

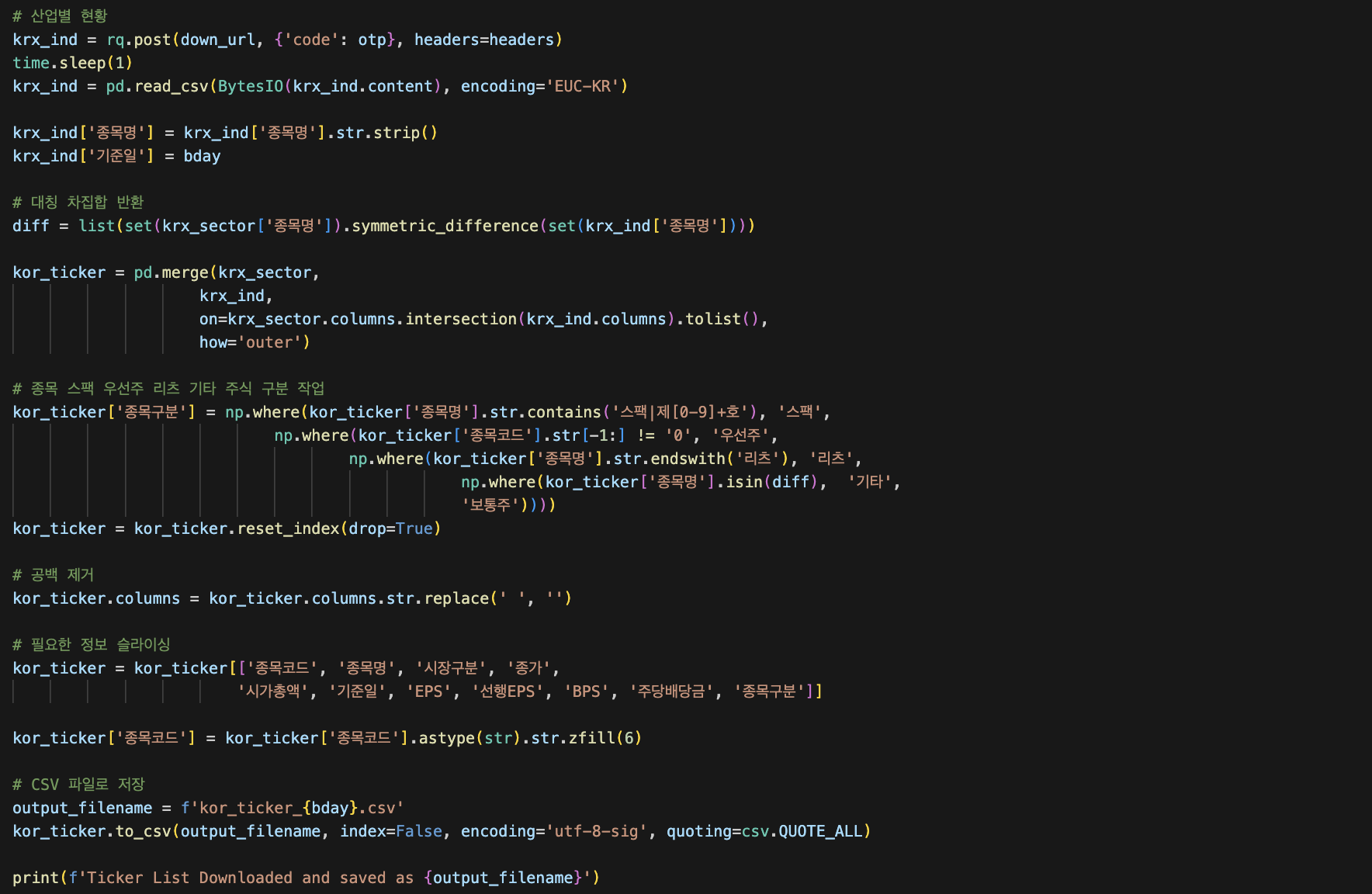


1) KRX 서버에 OTP를 먼저 요청 (generateOTP) , csv파일로 저장

2) 크롤링 방지 피하기 위해 경로 우회용 헤더 삽입 (요청 헤더에 Referer 포함)

3) KOSPI, KOSDAQ 데이터 수집 후 merge

4) 종목명 중복제거 및 기준일(bday) 삽입



1. 산업별 현황 데이터( 예: EPS, BPS 등) 따로 요청
2. KOSPI, KOSDAQ 종목 + 산업별 정보 - merge
3. 두 dataset 간의 종목 불일치 목록 (diff) 도출 - outer join 기준으로 kor\_ticker 생성
4. 종목명 분류 ( 스팩 종목, 우선주 , 리츠 , 기타 ,보통주)
5. 컬럼 정리 및 정규화
6. csv 저장 - ‘kor\_ticker\_{bday}.csv’

**4. Sector List**

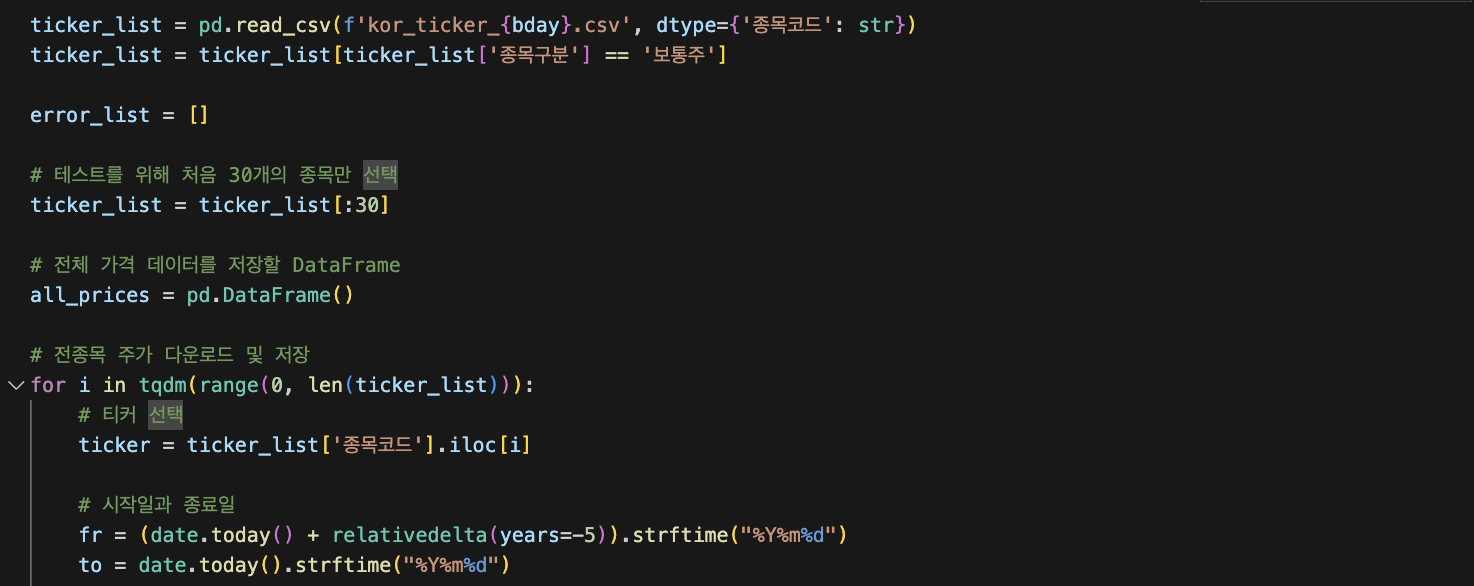
: WiseIndex 제공하는 섹터별 종목 구성 데이터를 활용하여, KOSPI/KOSDAQ 주요 섹터에 포함된 종목 리스트를 수집



1. WiseIndex 에서 정의한 10개 주요 섹터 코드 저장 ( IT, 바이오, 금융, 소재, 소비재 등 다양한 산업군 포함)
2. Sector 별 종목 데이터 요청 및 수집 (requests module 활용) , DataFrame으로 변환
3. 수집된 sector 별 종목 데이터를 하나의 DataFrame으로 병합 (지수코드 , 종목코드, 종목명, 섹터명만 선택 정리)
4. 기준일(bday)를 각 행에 삽입, 종목 코드 문자열화 + 6자리 맞춤
5. csv 저장 - 'kor\_sector\_{bday}.csv’

**5. price list**

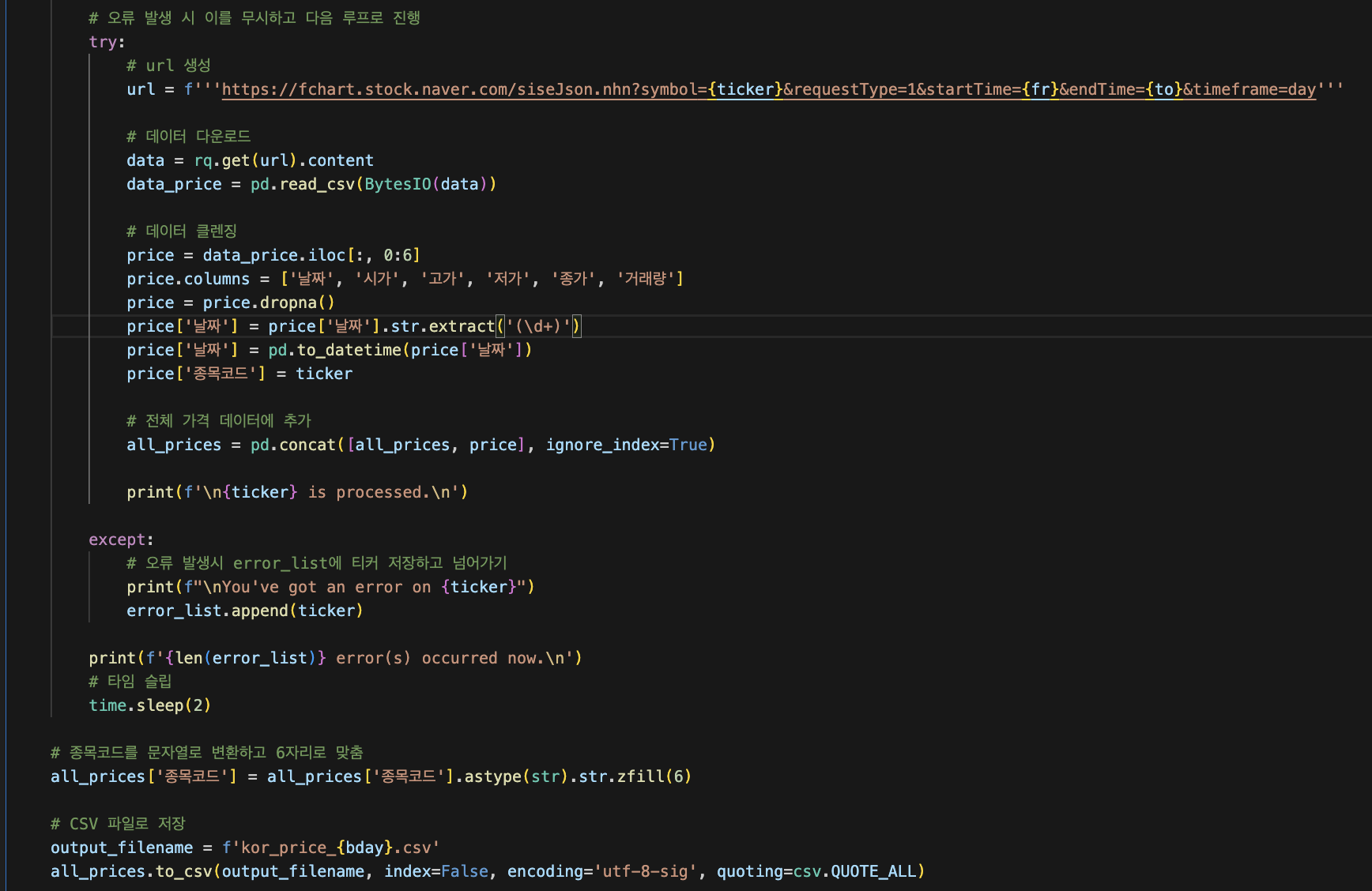
: 한국 보통주 종목들의 최근 5년간 일별 주가 데이터를 수집, 각 종목의 일자별 시가,고가,저가, 종가 거래량 데이터를 수집



1) kor\_ticker\_{bday}.csv 에서 보통주만 필터링 하여 사용

2) 종목 리스트를 순회하며 개별 종목 코드 선택

3) 시작일 : 현재일 기준-5년 , 종료일 : 오늘 날짜

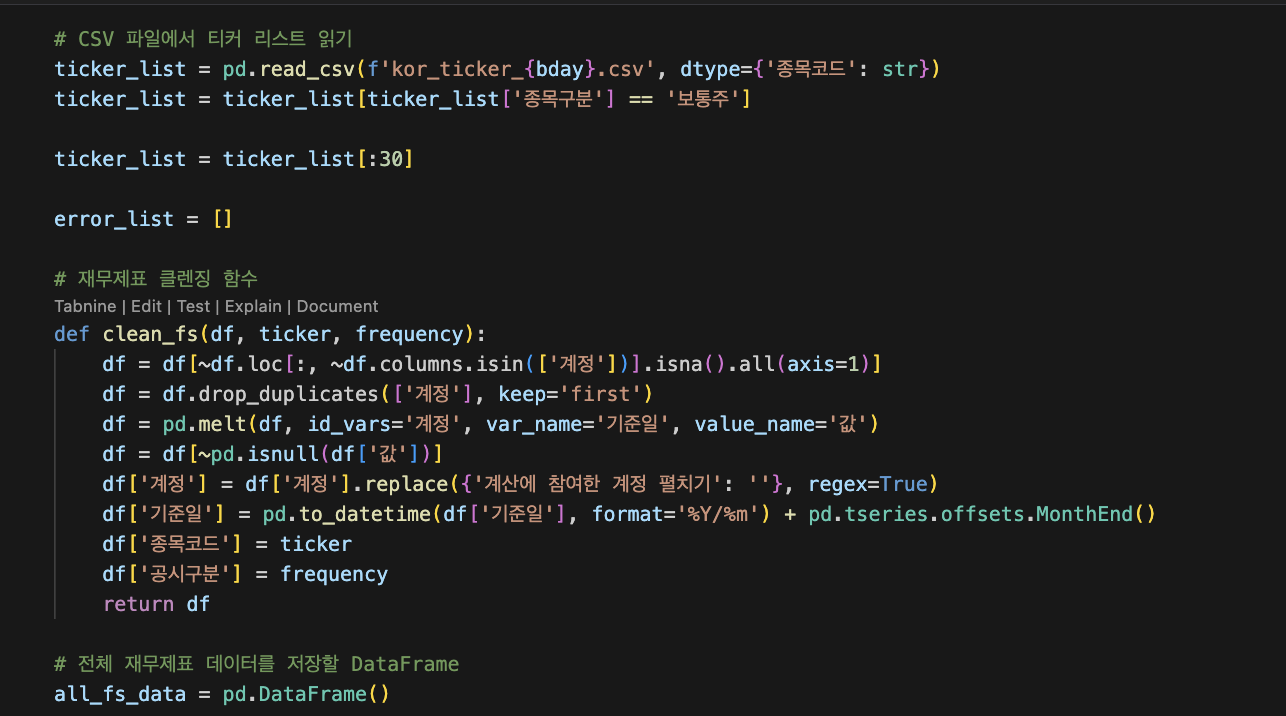


4) 네이버 fchart api를 통해 해당 종목의 일별 데이터 요청

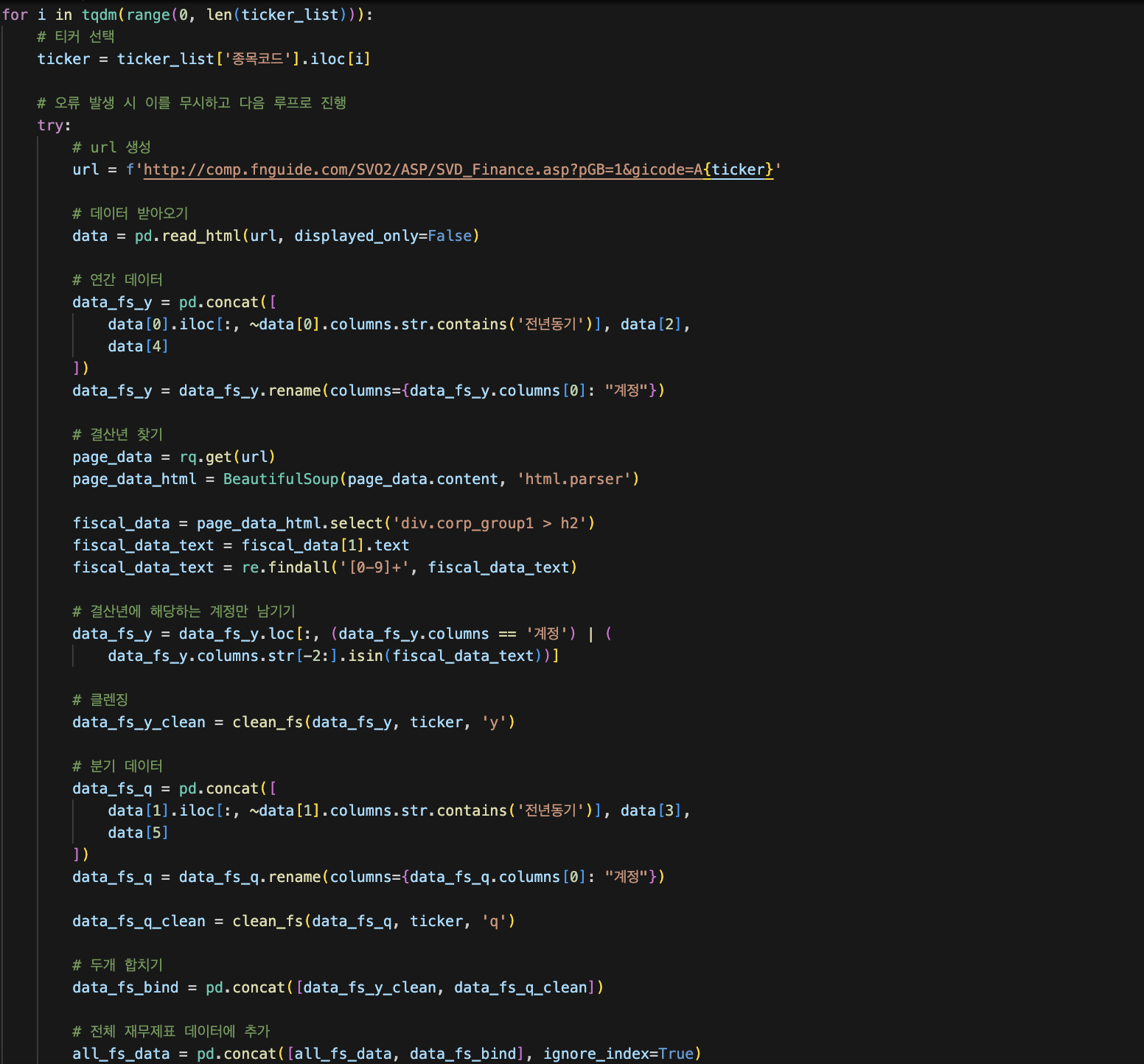
5) csv 저장 - ‘kor\_price\_{bday}.csv

**6.Financial Statement List**

**:** 한국 상장기업의 연간 및 분기 재무제표 데이터를 수집

****

1. 영업일 기준으로 보통주 종목 필터링.
2. clean\_fs() : 입력 데이터의 결측 행 제거 → 중복 계정명 제거 → wide **→** long format 변환 → 날짜 파싱 및 보정 → 종목코드 및 공시주기 컬럼 추가

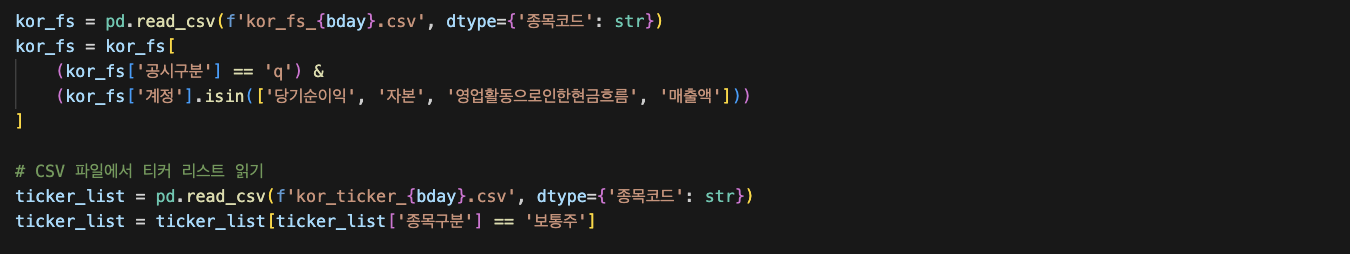


1. FnGuide의 연간 재무제표는 3개 테이블로 분할되어 있으므로 이를 수평 결합
2. 연간 제무제표 - 결산 연도 필터링을 위해 HTML의 corp\_group > h2 태그에서 텍스트를 추출하여 해당 연도에 해당하는 column만 유지
3. 분기 제무제표 - 마찬가지로 3개 테이블에서 분기 데이터를 수집 및 병합
4. 클렌징 및 merge
5. csv 저장 - ‘kor\_fs\_{bday}.csv’

**7. Value list**

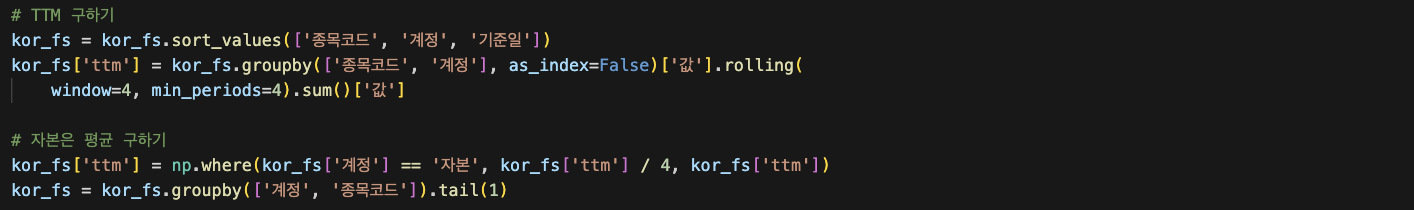
**:** 국내 상장기업의 분기 재무제표 데이터를 기반으로 주요 밸류에이션 지표(PER, PBR, PSR, PCR, DY)를 수집

1) 분기 공시 데이터만 선택, 밸류에이션 지표에 필요한 핵심 계정 항목만 추출

****

2) 최근 4개 분기의 데이터를 기반으로 TTM 합산 계산, 자본 계정은 평균값을 사용

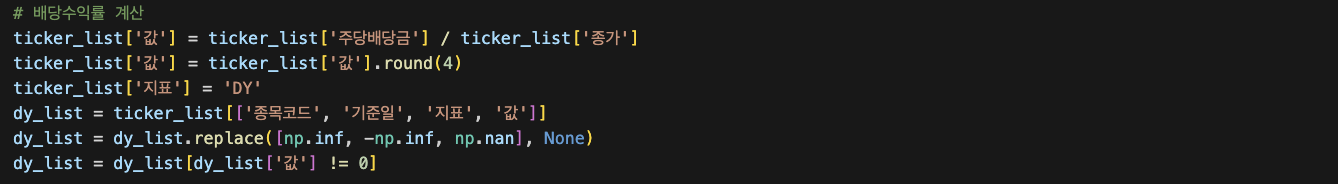
3) 각 계정/종목코드 별로 최신 영업일의 데이터만 남김



4) tiker list 에서 시가총액 병합 , 지표 계산:value = 시가총액 / TTM 값

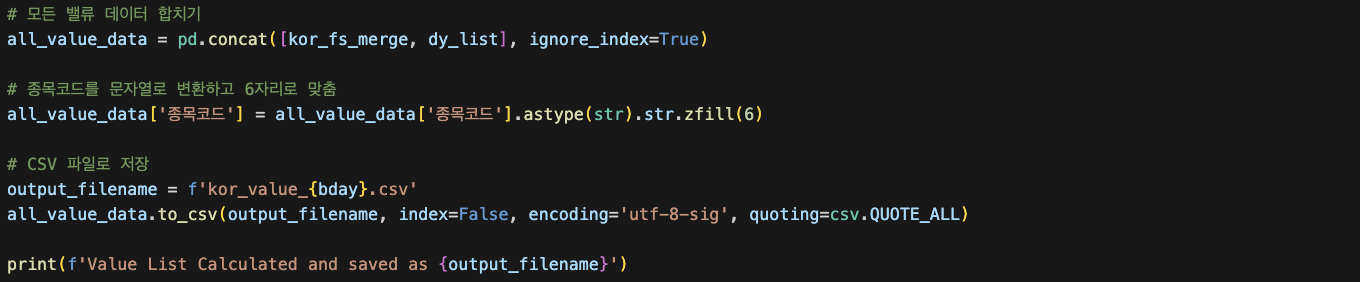


5) DY = 주당배당금 ÷ 종가**,** 계산 결과를 dy\_list 로 별도 추출 및 클렌징



6) PER, PBR, PSR, PCR, DY 지표를 모두 합쳐서 최종 결과 생성

7) csv 저장 - ‘kor\_value\_{bday}.csv’

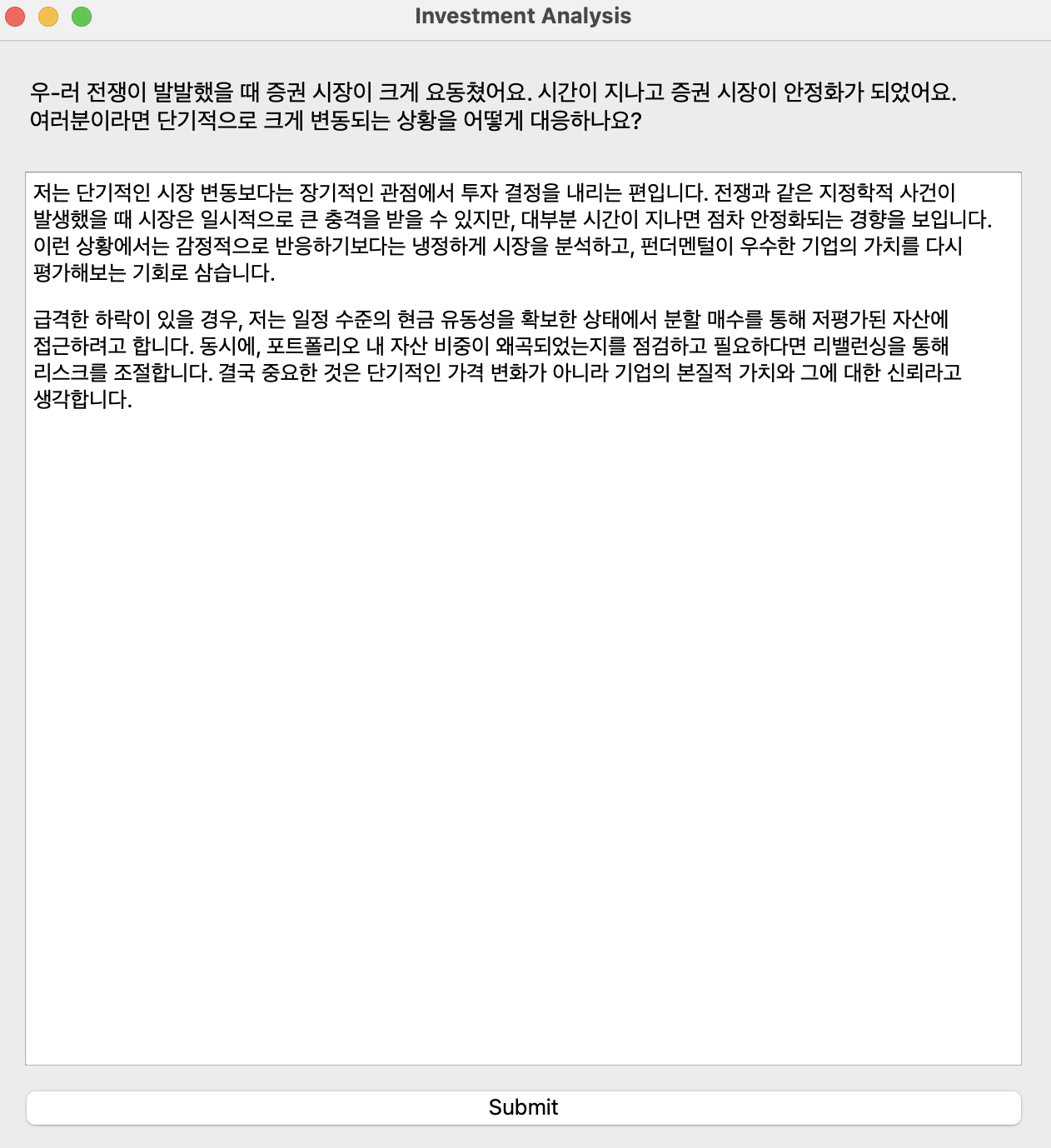
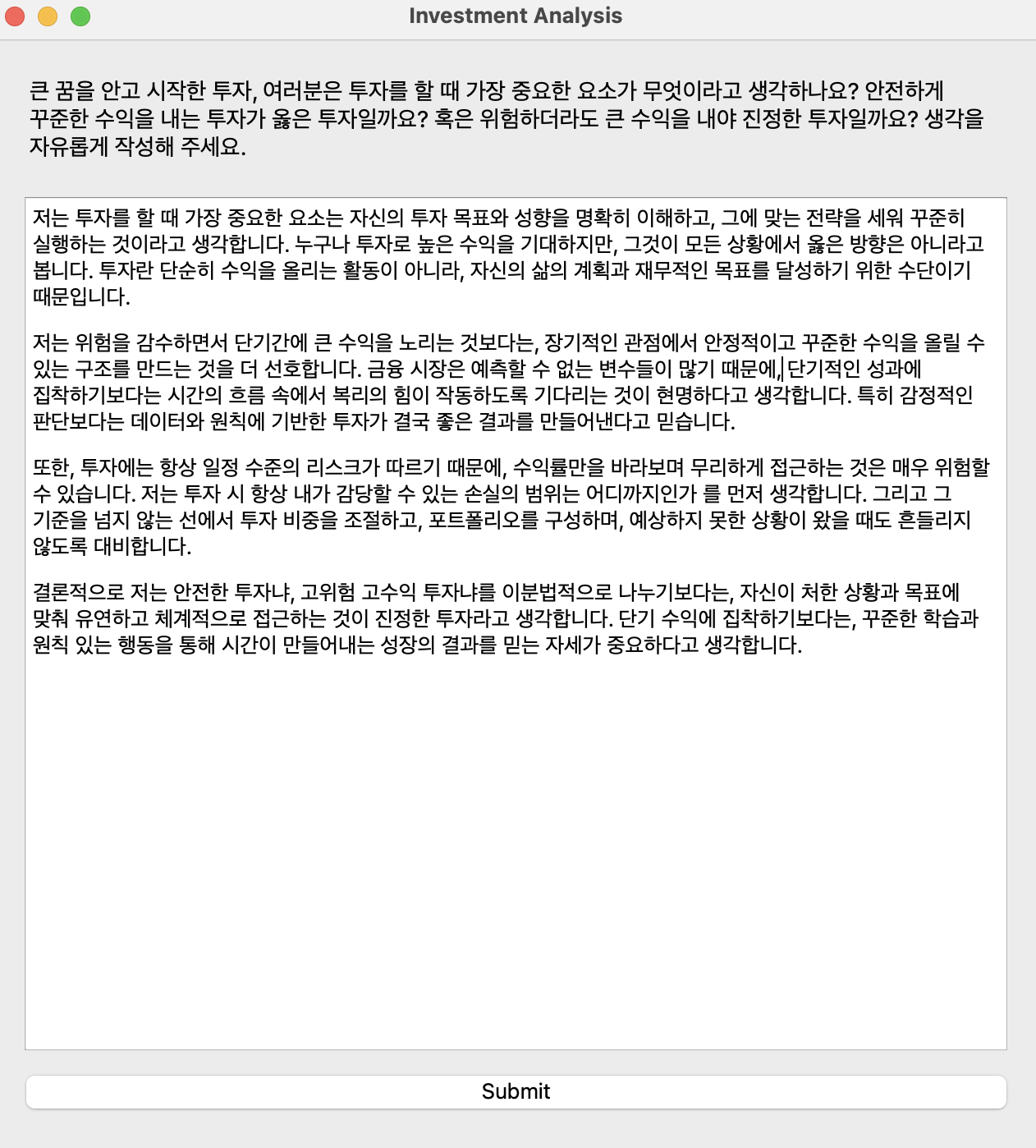


2. Preliminary Results

**2-1. 사용자 성향 분석 결과**

- user\_survey.py의 process\_answer 및 analyze\_results 함수, 그리고 Clova Studio API 연동을 통해 실제 사용자 응답 데이터를 수집 및 분석.

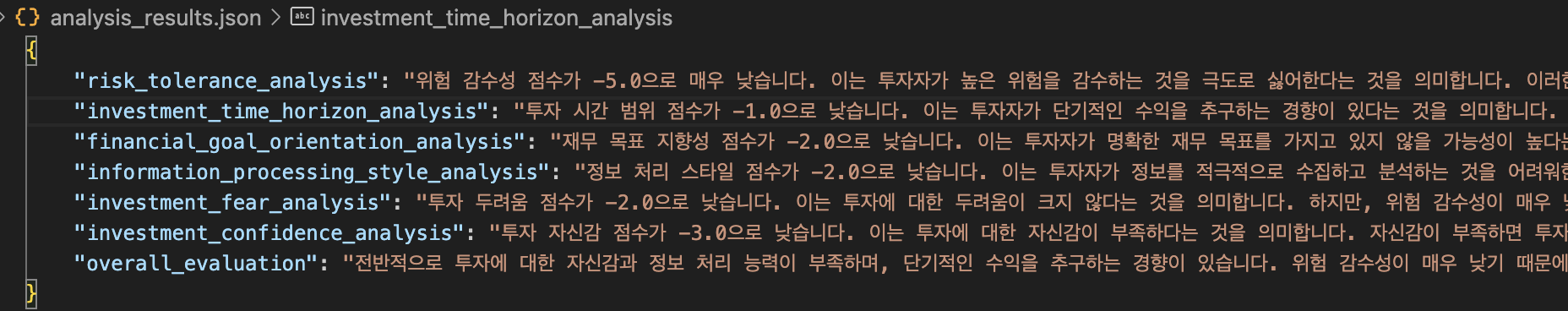
ex) user\_survey.py



- 10개 서술형 문항에 대한 응답을 Clova LLM에 전달한 결과, 각 항목별( 위험 선호, 정보 탐색, 감정 반응 등) 점수가 JSON 형태로 반환되어 total\_scores에 누적됨을 확인.

- 누적 점수를 기반으로 analyze\_results 함수가 종합 분석 프롬프트를 생성, Clova API로부터 상세한 투자 성향 리포트(“위험 중립적, 정보 탐색 적극적, 감정 기복 낮음” 등)를 성공적으로 수신하였고, 결과는 analysis\_results.json에 저장됨을 확인.

ex) analysis\_results.json



- 사용자 시뮬레이션 및 내부 테스트

실제 GUI 상에서 사용자별로 상이한 성향 프로파일이 시각적으로 출력되는 것을 검증하였으며, 이는 이후 투자 전략 추천에 활용 가능한 구조임을 확인.

Clova Studio API를 이용한 실제 사용자 성향 분석 시나리오를 3회 이상 반복 수행, 다양한 응답에 따른 점수 및 리포트 분석 결과의 일관성 및 유효성 검토 완료.

성향 분석 결과가 추천 종목 및 LLM 조언에 어떤 식으로 영향을 미치는지 확인하기 위해, 샘플 사용자 A/B 테스팅 구조를 기획 및 테스트 중.

ex) 사용자 A(고위험/공격형) vs. 사용자 B(저위험/장기투자형)에 대해 서로 다른 종목이 추천되는지 확인.

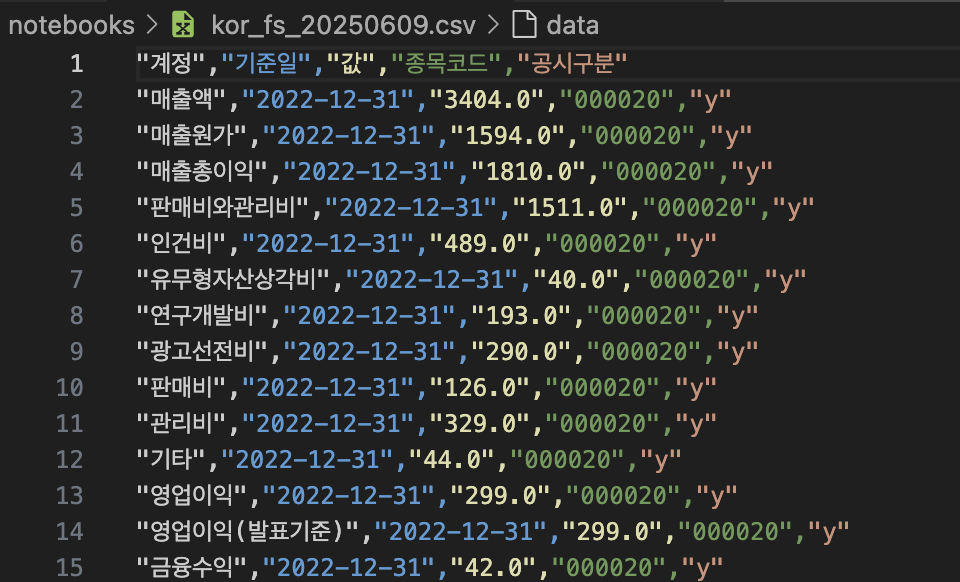
결과는 analysis\_results.json 및 recommendation\_log.json으로 저장되어, 추후 설명 가능성(Explainability)을 위한 로깅 자료로 활용 가능.

**2-2. 데이터 크롤링 및 전처리**

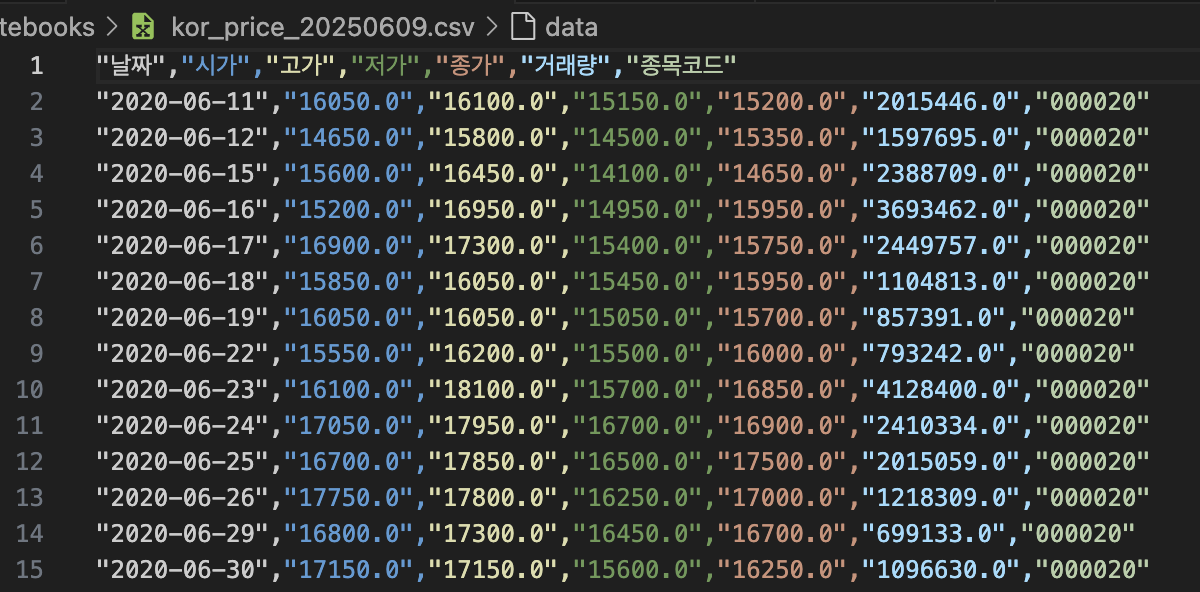
- data\_collector.py와 data\_processor.py를 통해 네이버 금융, KRX, WISEINDEX 등에서 실시간으로 영업일, 종목, 섹터 데이터를 수집하였고, pandas 기반 전처리 로직이 정상적으로 동작함을 확인.

- OTP 기반 KRX 종목 리스트 요청, 산업별 현황 데이터 병합, 종목 구분(스팩, 우선주, 리츠 등) 분류, 컬럼 정규화 등 일련의 데이터 파이프라인이 자동화되어, kor\_ticker\_{bday}.csv, kor\_sector\_{bday}.csv 파일이 생성됨을 검증.

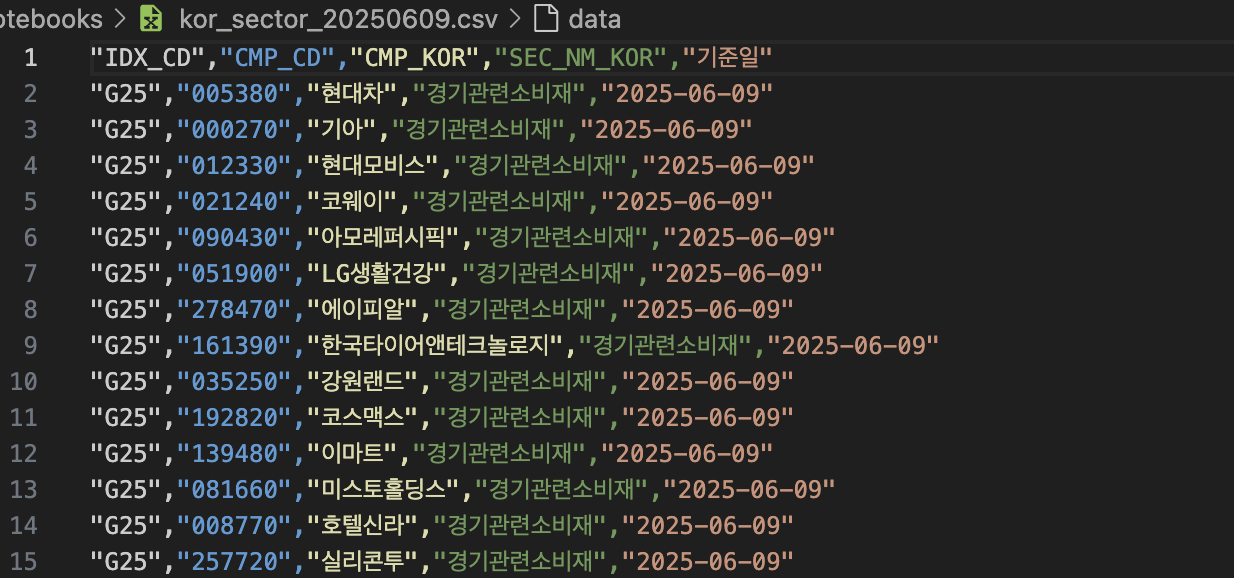
ex) kor\_fs\_{bday}.csv



ex) kor\_price\_{bday}.csv



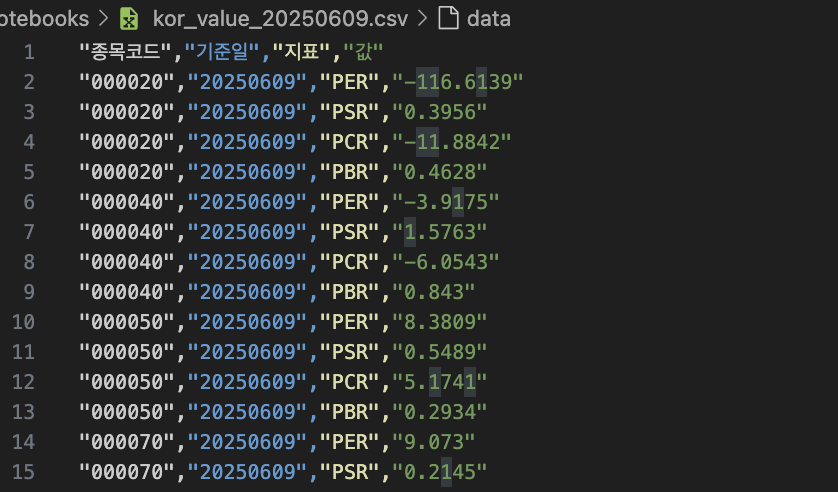
ex) kor\_sector\_{bday}.csv



ex) kor\_ticker\_{bday}.csv



ex) kor\_value\_{bday}.csv



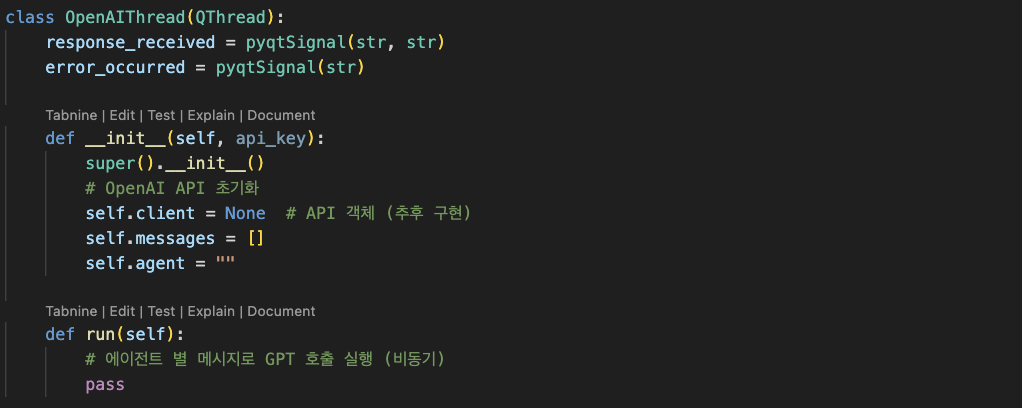
- 데이터 수집 및 전처리 과정은 MinervaSystem 클래스의 run\_immediate\_data\_update() 및 start\_data\_update\_scheduler() 메서드를 통해 명령행에서 직접 실행 및 스케줄링이 가능함을 확인.

3. Plan for Final Submission  
Outline the remaining tasks and strategies, including a timeline that leads to the final submission.

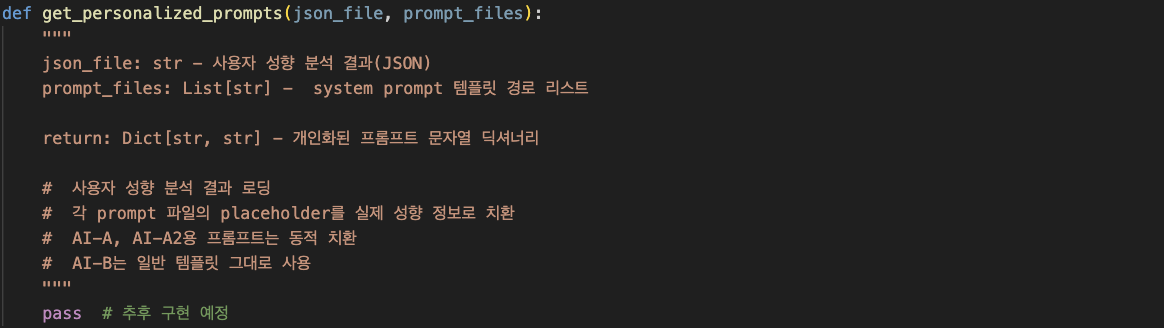
3-1. 남은 개발 과제 및 전략

- minerva.py 구현

1. OpenAITread : 비동기 응답 처리 및 에러 전파를 위한 GPT 스레드 클래스

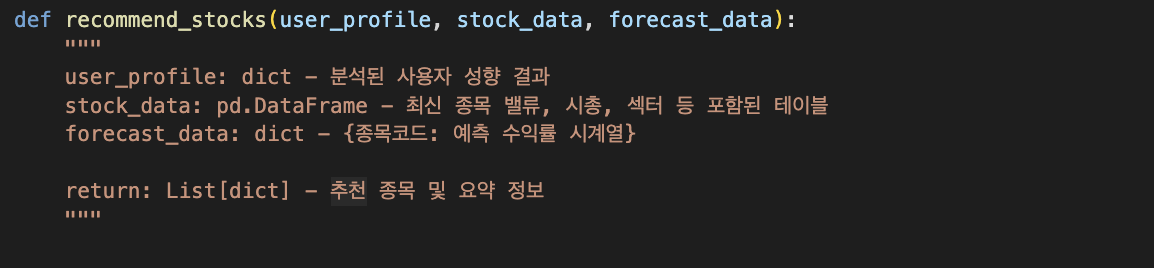


1. FinancialChatbot: 각 에이전트의 응답 생성 로직(AI-A → AI-A2 → AI-B → Final)을 구현
2. PyQt5 기반의 사용자-챗봇 인터페이스 및 실시간 상태 표시
3. get\_personalized\_prompts() :설문 결과를 기반으로 프롬프트를 동적으로 생성 및 삽입



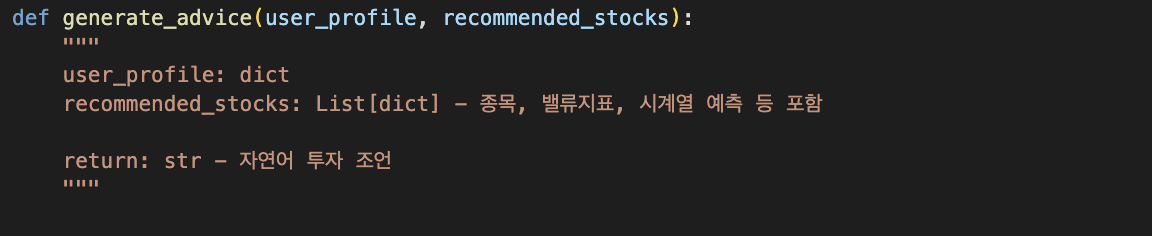
- LLM 기반 투자 전략 추천 엔진 고도화

- investment\_advisor.py, llm\_service.py 내에서 사용자 성향(analysis\_results.json)과 실시간 금융 데이터(데이터 크롤링 결과)를 통합하여, 벡터 임베딩 기반 유사도 매칭 및 시계열 예측(LSTM, GRU 등) 결과를 반영한 맞춤형 포트폴리오 추천 로직을 구현할 예정입니다.



사용자 성향 + 종목 데이터 -> 필터링 및 랭킹 -> 추천 종목 리스트 생성

- MinervaSystem의 프롬프트 자동 완성 및 LLM 질의 함수(llm\_service.generate\_advice() 등)와 연동하여, 실제 투자 조언이 자연어로 생성되도록 개선합니다.



investment\_advisor.py 의 추천결과 -> 자연어 프롬프트 생성

- 시계열 예측 모델 통합

- financial\_data\_processor.py에 LSTM/GRU 기반 주가 예측 모델을 추가하고, 예측 결과를 투자 어드바이저 모듈과 연동하여 미래지향적 투자 전략을 제안할 수 있도록 구현합니다.

- 웹 인터페이스 개발 및 통합

- 현재 PyQt5 기반 GUI를 웹 프론트엔드(React, Flask 등)로 확장하여, 실시간 사용자 입력, 결과 시각화, 투자 조언 제공이 가능하도록 할 예정입니다.

- API 서버(api\_service.py)를 통해 프론트엔드와 백엔드가 RESTful 방식으로 통신하도록 설계합니다.

멀티에이전트 프레임워크 적용

* 사용자 중심 에이전트, 시장 전문가 에이전트, 조정 에이전트 등으로 역할을 분리하여, 각 에이전트가 독립적으로 동작하면서도 협력적으로 결과를 도출하는 구조를 구현합니다.
* 각 에이전트는 별도의 클래스로 구현(user\_agent.py, market\_agent.py, coordinator\_agent.py 등)되며, MinervaSystem에서 통합 관리합니다.

- 사용자 중심 에이전트, 시장 전문가 에이전트, 조정 에이전트 등으로 역할을 분리하여, 각 에이전트가 독립적으로 동작하면서도 협력적으로 결과를 도출하는 구조를 구현합니다.

- 각 에이전트는 별도의 클래스로 구현(user\_agent.py, market\_agent.py, coordinator\_agent.py 등)되며, MinervaSystem에서 통합 관리합니다.

리스크 요인 & 대응 계획:

* Clova Studio API의 비용 및 호출 제한 문제 → 캐싱 및 결과 저장 방식 적용

- yfinance API 호출 실패 가능성 → 백업 소스로 네이버 금융 웹크롤링 구성

3-2. 일정

2025.06 ~ 2025.08 LLM 기반 투자 전략 추천 로직 및 시계열 예측 모델 개발, 코드 통합 테스트

2025.09 ~ 2025.10 | 웹 인터페이스 개발, API 서버 구축, 프론트-백엔드 연동

2025.11 멀티에이전트 프레임워크 적용, 통합 시스템 시나리오 테스트

2025.12 최종 리포트 작성, 시연 준비, 결과물 제출 및 발표

3-3. 웹페이지 구현 상세

1) 프레임워크 및 구조

- 프론트엔드는 React.js 기반으로 개발할 예정이며, 사용자 친화적인 UI/UX를 위해 Material-UI, Chart.js 등 최신 라이브러리를 활용할 계획입니다.

- 백엔드는 Flask 기반 RESTful API 서버로 구현, 프론트엔드와의 데이터 통신을 담당

- 전체 웹 시스템은 /web 디렉토리에 위치

- /web/frontend/ : React 소스코드(컴포넌트, 라우팅, 스타일 등)

- /web/backend/ : Flask API 서버 코드

- /web/static/ : 정적 파일(이미지, CSS 등)

2) 주요 기능 및 컴포넌트

- 로그인 및 사용자 인증

- LoginPage.jsx에서 이메일/비밀번호 기반 로그인 구현, JWT 토큰 방식으로 인증 처리

- 회원가입, 비밀번호 찾기 등 부가 기능 포함

- 투자 성향 설문 페이지

- SurveyPage.jsx에서 10개 서술형 문항을 순차적으로 입력받는 폼 구현

- 각 답변은 실시간으로 상태(state)에 저장되며, 제출 시 /api/survey 엔드포인트로 POST 요청

- 제출된 답변은 백엔드에서 Clova Studio API와 연동되어 점수화 및 분석 결과 반환

- 분석 결과 및 투자 조언 시각화

- ResultPage.jsx에서 투자 성향 분석 결과(예: 위험 선호, 정보 탐색 등)를 레이더 차트, 바 차트 등으로 시각화

- 맞춤형 투자 조언은 자연어 텍스트와 함께, 추천 포트폴리오(종목 리스트, 비중 등)를 표와 그래프로 제공

- Chart.js, Material-UI Table 등 시각화 라이브러리 활용

- 실시간 금융 데이터 및 예측 결과 제공

- MarketPage.jsx에서 주요 지수, 관심 종목의 실시간 시세, LSTM/GRU 기반 예측 결과를 차트로 표시

- 데이터는 /api/market 등 백엔드 API에서 주기적으로 받아옴

- 사용자 맞춤 대시보드

- Dashboard.jsx에서 전체 투자 현황, 최근 조언, 알림, 투자 목표 달성률 등을 한눈에 확인 가능

- 각 위젯은 독립 컴포넌트로 설계되어 유지보수 용이

3) 백엔드 연동 및 API 설계

- Flask 기반 API 서버 (/web/backend/app.py)

- /api/survey : 설문 응답 수신 → Clova Studio API 호출 → 점수/분석 결과 반환

- /api/advice : 사용자 성향, 금융 데이터 기반 투자 조언 생성 및 반환

- /api/market : 실시간 시세, 예측 결과 제공

- /api/user : 회원 정보, 투자 이력 등 관리

- CORS, JWT 인증, 에러 핸들링 등을 적용하여 보안 및 신뢰성 강화

4) 사용자 경험(UX) 및 디자인

- 반응형 웹으로 설계하여 PC, 태블릿, 모바일 모두에서 최적화된 화면 제공

- 직관적인 네비게이션 바와 단계별 진행 표시(Stepper)로 사용자가 설문, 결과 확인, 투자 조언까지 자연스럽게 이동 가능

- 로딩, 에러, 성공 메시지 등 피드백 요소를 강화하여 사용자의 혼란 최소화

- 접근성(Accessibility)을 고려한 색상, 폰트, 버튼 크기 등 UI 요소 설계

5) src/main.py와의 연동

- 웹 백엔드(Flask)는 내부적으로 src/main.py의 MinervaSystem 클래스를 임포트하여,

투자 성향 분석, 데이터 업데이트, 투자 조언 생성 등 핵심 로직을 직접 호출

(예: /api/advice 요청 시, MinervaSystem.investment\_advisor.recommend\_portfolio() 실행)

- 이를 통해 웹페이지에서 입력된 데이터가 곧바로 AI 엔진에 반영되어, 실시간 맞춤형 결과 제공이 가능

4. References

List all sources referenced, ensuring proper citation to support the research foundation and demonstrate depth of review.

1. Kim, H., & Park, J. (2024). *FinCon: A Synthesized LLM Multi-Agent System with Conceptual Verbal Reinforcement for Enhanced Financial Decision Making*. In *Proceedings of the 38th Conference on Neural Information Processing Systems (NeurIPS 2024)*.
2. Lee, J., Kim, D., & Park, S. (2020). *MAPS: Multi-agent Reinforcement Learning-based Portfolio Management System*. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI 2020)*.
3. Chen, M., Zhao, Y., Xu, J., & Li, X. (2023).  
    *LLM4Finance: Leveraging Large Language Models for Financial Decision Making*. *arXiv preprint arXiv:2305.14328*.<https://arxiv.org/abs/2305.14328>
4. Liu, X., Zhang, X., & Wang, C. (2023). *FinGPT: Large Language Model for Finance*. *arXiv preprint arXiv:2306.03048*.<https://arxiv.org/abs/2306.03048>
5. Li, Z., Wang, J., Liu, Y., & Wu, C. (2024). *Developing a Multi-Agent and Self-Adaptive Framework with Deep Reinforcement Learning for Dynamic Portfolio Risk Management*. In *Proceedings of AAMAS 2024*.
6. Choi, E., Lee, K., & Jung, M. (2024). *TradingAgents: Multi-Agent LLM Financial Trading Framework*. *arXiv preprint arXiv:2402.10555*.<https://arxiv.org/abs/2402.10555>
7. Fang, Y., Zhao, H., & Zhou, Y. (2023). *Learning Multi-Agent Intention-Aware Communication for Optimal Multi-Order Execution in Finance*. In *EMNLP 2023*.

Dataset & API

1. OpenAI. (2023). *GPT-4 Technical Report*.<https://openai.com/research/gpt-4>
2. NAVER Cloud Platform. (2024). *Clova Studio API Documentation*.

<https://api.ncloud-docs.com/docs>

1. Yahoo Finance. (2024). *yfinance Python Library Documentation*.<https://pypi.org/project/yfinance/>
2. FnGuide. (2024). *재무제표 및 가치지표 데이터 제공 가이드라인*.<https://www.fnguide.com/>
3. 한국거래소 (KRX). (2024). *한국거래소 데이터포털*. <https://data.krx.co.kr>
4. WiseIndex. (2024). *섹터별 지수 및 구성종목 데이터*.<http://www.wiseindex.com>