「최적의 주식 투자 전략」

최종보고서

2023. 12. 14

이 현 중

졸 업 작 품 최 종 보 고 서

학부/전공	데이터사이언스학과		
성 명	이현중	졸업예정일	24년 2월
구 분	단 독 ■ 공 동 □ ( )인		
휴대전화	01048869590	E-mail	lhj990502@naver.com
작품 제목	최적의 주식 투자 전략		
작품 개요	ㅇ 금융 시장은 현대 사회에서 매우 중요한 역할을 하지만 어렵고 복잡한 정보와 전문 용어들로 가득 차 있어서, 초보자나 소득이 적은 사람들은 개인 투자 서비스를 제대로 받거나 관련된 정보를 얻기 어렵거나 금융 위기 상황에서 신속하게 대응하기가 어려울 수 있다. ㅇ 본 프로젝트는 급변하는 금융 환경에서 능동적으로 리스크를 관리하기 위하여 금융 관련 지식(이동평균, MACD, RSI 등)과 인공지능, 추천시스템, 강화학습, 딥러닝 기술을 다양하게 적용하여 금융 트랜드를 분석하여 개인에게 최적화된 투자전략을 추천하기 위한 프로그램을 개발한다. ㅇ 오픈 API와 크롤링을 활용하여 금융 데이터를 수집하고 기본적 분석, 기술적 분석, 추천시스템, 강화학습 및 딥러닝 등 다양한 기술을 사용하여 종목 추천, 추천된 종목별 매도, 매수, 홀딩 타이밍을 추천하는 서비스를 개발한다. ㅇ 기본적, 기술적 분석을 위한 퀀트투자 이동평균, 볼린저밴드, MACD, RSI 등 금융 핀테크에 대한 지식 확보하며, FinanceDataReader , BeautifulSoup4 , OpenAPI, 크롤링 등의 데이터 수집 기술과 금융 예측 모형, 타이밍 및 상품 추천 시스템 기술 및 강화학습을 통한 추천 및 최적화 기술을 습득한다. ㅇ 본 프로젝트를 통해 향후, 금융 트랜드분석을 기반으로 최적화된 투자전략 추천 서비스에 활용 가능하며, 개인 투자 서비스를 받기 어렵거나 관련된 정보에 소외된 사회 저소득층 및 초년생들에게 관련 정보를 제공하여 금융 위기에 신속하게 대응가능하다.		

위와 같은 내용으로 최종보고서를 제출합니다.

2023년 12월 14일

제 출 자 : 이현중 이현중

인 공 지 능 융 합 공 학 부

※ 아래와 같은 내용의 **최종보고서를 작성하여 제출**하시오.

- 데이터 분석 기획 전 프로세스 내용 포함 (문제 사항 보완 결과 포함)
- 작품 제목, 필요성, 작품 개요(각 구성 요소별 block diagram 수준 기술), 분석 기술(플랫폼, 사용 프로그래밍 언어 및 기타 도구), 필요 장비/소프트웨어, 제작 과정, 작품 해석 등

※ 2인 이상의 공동 작품은 최종보고서를 각자의 담당 부분을 중심으로 기술하여 개별적으로 제출하되, 작품 전체에 대한 요약본을 첨부하시오.

승인여부	가(　)	부(　)	가(　)	부(　)	가(　)	부(　)	가(　)	부(　)
지도교수	(인)		(인)		(인)		(인)	

보 완 사 항 확 인 서

학부/전공	데이터사이언스		
성 명	이현중	졸업예정일	24년 2월
구 분	단 독 ■ 공 동 □ ( )인		
휴대전화	01048869590	E-mail	lhj990502@naver.com
작품 제목	최적의 주식 투자 전략		
요구사항	1. 최종목표가 포트폴리오 가치를 높이는 것인지, 아니면 개별 주식에 중점을 두어 각 주식에 대한 매수, 매도, 홀딩 타이밍을 예측하는 것인지를 정확히 하고 마무리해야 함 2. 수익을 낼 수 있는지 확인해야 함 3. 현재 주식 추천 서비스는 다양하게 존재하지만, 나만의 독특한 차별점이 무엇인지 고민해야 함		
최종 결과	1. 최종으로 포트폴리오 가치를 높이는 것을 목표로 함 2. 최종적으로 90,000,000원의 수익을 냄 3. 현재 주식 추천 서비스는 대부분 유료이지만, 본 서비스는 무료로 이용 가능. 특히, 돈이 부족한 초년생 등을 대상으로 한다는 점에서 차별점이 있다고 생각함		

위와 같이 졸업작품을 최종 보완하였음을 확인합니다.

2023년 12월 14일

교 수 : (인)

제 출 자 : 이현중 이현중

인 공 지 능 융 합 공 학 부

목 차

제 I 장 서론 ····· **2**
1 목적 및 필요성 ····· 2
2 활용 방안 ····· 3

제 II 장 분석 기획 ····· **4**
1 비즈니스 이해 및 범위 설정 ····· 4
2 프로젝트 정의 및 계획 설정 ····· 5
3 위험계획 수립 ····· 6

제 III 장 데이터 준비 ····· **7**
1 필요 데이터 정의 ····· 7
2 데이터 수집 및 전처리 ····· 8
3 품질 점검 ····· 12

제 IV 장 데이터 분석 ····· **13**
1 분석용 데이터 준비 ····· 13
2 탐색적 분석 ····· 17
3 모델링 ····· 22
4 모델평가 및 검증 ····· 31

제 V 장 평가 및 전개 ····· **33**
1 모델 발전 계획 수립 ····· 33
2 프로젝트 평가 및 향후 계획 ····· 33

참고문헌
참고문헌 ····· **34**

Ⅰ 서론

1. 목적 및 필요성

가. 목적

□ 금융 포트폴리오 최적화

○ 포트폴리오 최적화는 위험을 최소화하면서 수익률을 극대화하는 자산 포트폴리오를 구성하는 과정이다. 금융 포트폴리오 최적화 프로그램은 투자자가 위험 허용 범위, 투자 목표 및 시장 상황에 따라 투자 결정을 내릴 수 있도록 지원하는 데 사용된다. 기존의 포트폴리오 최적화 방법은 시간이 많이 걸리고 구현하기 어려운 통계 및 수학적 모델(MPT, CAPM, Markowitz 모형, Black-Litterman 모형 등)에 의존하여 최적의 투자 결정을 내린다. 때문에, 일부 투자자들에게는 이해하기 어려울 수 있고, 이러한 방법은 예측이 부정확하거나 예측할 수 없는 급격한 변화가 일어났을 때 제대로 작동하지 않을 수 있다.

○ 투자자들이 위험을 최소화하면서 수익률을 극대화하는 정보에 입각한 투자 결정을 내려야 한다. 기술의 발전으로 인공지능(딥러닝, 강화학습)을 활용하여 실시간으로 대량의 데이터를 분석, 해석할 수 있게 되어 보다 정확하고 최신의 투자 결정이 가능하다. 또한, AI 기반 포트폴리오 최적화 프로그램은 과거 투자 결정을 학습하고 변화하는 시장 상황에 적응하여 효율적이고 효과적인 투자 전략을 수립할 수 있다. 추천 기술과 인공지능을 기반으로 한 금융 포트폴리오 최적화 프로그램을 개발함으로써, 투자자들은 수익을 개선할 뿐만 아니라, 위험을 줄이는 정보에 입각한 투자 결정을 내릴 수 있다.

나. 필요성

□ 투자 트랜드의 변화

○ 국내에만 2000개가 넘는 주식 종목이 있으며 이들에 대해서 이러한 분석을 일일이 하는 것은 사실상 불가능에 가깝기에 많은 부분을 자동화 해야한다. 또한 투자자들이 가장 관심 있는 것은 위험을 최소화하면서 수익을 극대화하는 것이기에 현대 기술의 도움을 받아 투자 전략을 더욱 정확하게 세우고 효과적으로 수행할 수 있는 프로그램은 금융 시장에서 필수적이다. 그렇기에 인공지능을 이용한 주식 투자는 앞으로로의 투자 트랜드가 될 것이다.

□ 금융 위기 대응

○ 금융 분야에서 포트폴리오 최적화는 중요한 역할을 하며, 투자자에게 금융 시장의 불확실성을 극복하는데 도움을 준다.

○ 금융 시장에서는 예측할 수 없는 위기 상황이 발생할 수 있다. 이를 대비하기 위해

인공지능을 기반으로 한 포트폴리오 최적화 알고리즘을 사용하면 투자자들은 투자 포트폴리오를 신속하게 모니터링하고 조정할 수 있으며, 금융 위기 상황에서도 안정적으로 대응할 수 있다.

□ 개인 투자자의 현실

	2015	2016	2017	2018	2019	2020
개인	10	-25	11	-26	-12	12
기관	23	18	49	5	18	91
외인	40	10	47	5	30	124

그림1. 상위 50개 종목의 평균 수익률

○ 위 도표를 보면 개인 투자자들의 현실을 알 수 있다. 기관 및 외국인 투자자들에 비해 수익율이 현저히 낮은 것은 물론이며 수익률이 마이너스인 경우도 허다하다. 개인은 국내외 금융권 종사자에 비해 고급 정보를 접하기 어려울 수 밖에 없는 것이 현실인 것도 한 몫을 한다.

2. 활용 방안

□ 개인 투자자를 위한 포트폴리오 최적화

○ 개인투자자, 특히 초보자나 소득이 적은 사람들이 금융 시장에 적응하고 투자 포트폴리오를 구축하는 것은 어려운 일이다. 복잡한 금융 용어와 정보에 소외되는 경향이 있어서, 이들을 돕기 위한 혁신적인 서비스가 필요하다. 본 프로젝트를 통해 저소득층 및 초년생들에게 금융 시장의 정보와 서비스에 대한 더 나은 접근성을 제공할 수 있다.

□ 금융 위기 대응

○ 금융 위기 상황에서도 이 프로그램은 신속하게 대응할 수 있도록 포트폴리오 최적화 알고리즘을 적용하여 투자자의 포트폴리오를 모니터링하고 조정함으로써 금융 시장의 불확실성에 대응할 것이다.

Ⅱ 분석 기획

1. 비즈니스 이해 및 범위 설정

가. 비즈니스 이해

□ 본 프로젝트는 금융 시장에 어려움을 겪는 개인투자자, 특히 소득이 적거나 금융 분야에 대한 정보에 소외된 사회 저소득층 및 초년생들을 대상으로 한다. 이들은 복잡한 금융 용어와 정보에 어려움을 겪고 있으며, 이를 돕기 위한 서비스가 필요하다. 이 프로젝트는 고도의 기술과 정보 없이도 투자 포트폴리오를 구축하고 관리할 수 있는 서비스를 제공하여, 금융 시장의 정보와 서비스에 더 나은 접근성을 제공하고자 한다. 이를 위해 매도, 매수, 홀딩 타이밍 등의 투자 전략을 추천하는 것이 핵심 목표이다

나. 범위 설정

□ 종목 추천과 강화학습을 위한 데이터 수집

- ○ 기업의 재무 데이터, 실적 보고서, 경제 지표 등을 수집하여 주식의 잠재적 가치를 평가한다
- ○ 기업의 업종, 시가 총액, 성장성 등과 같은 특징을 고려하여 종목 추천에 활용할 데이터를 수집한다
- ○ 주식 가격, 거래량, 기술적 지표 등의 시계열 데이터를 수집하여 강화학습 모델을 훈련시키는 데 사용한다

□ 강화학습 모델 개발

- ○ 과거의 주가 움직임과 투자 전략의 성과를 학습하여 모델이 새로운 상황에 대응할 수 있도록 한다
- ○ 매도, 매수, 홀딩 타이밍 추천을 위한 강화학습 모델을 개발한다

□ 기간 및 일정

세부 업무	수행내용	추 진 일 정(월)				비중 (%)
		9	10	11	12	
주제 선정	주제 선정 및 사전 조사					10
데이터 수집	데이터 수집					20
데이터 정제	데이터 전처리					15
종목 추천	라벨링을 통해 데이터 가공					15
모델 개발	데이터 품질과 유효성 검증					25
결과 보고	졸업 발표 및 최종 보고서 작성					15

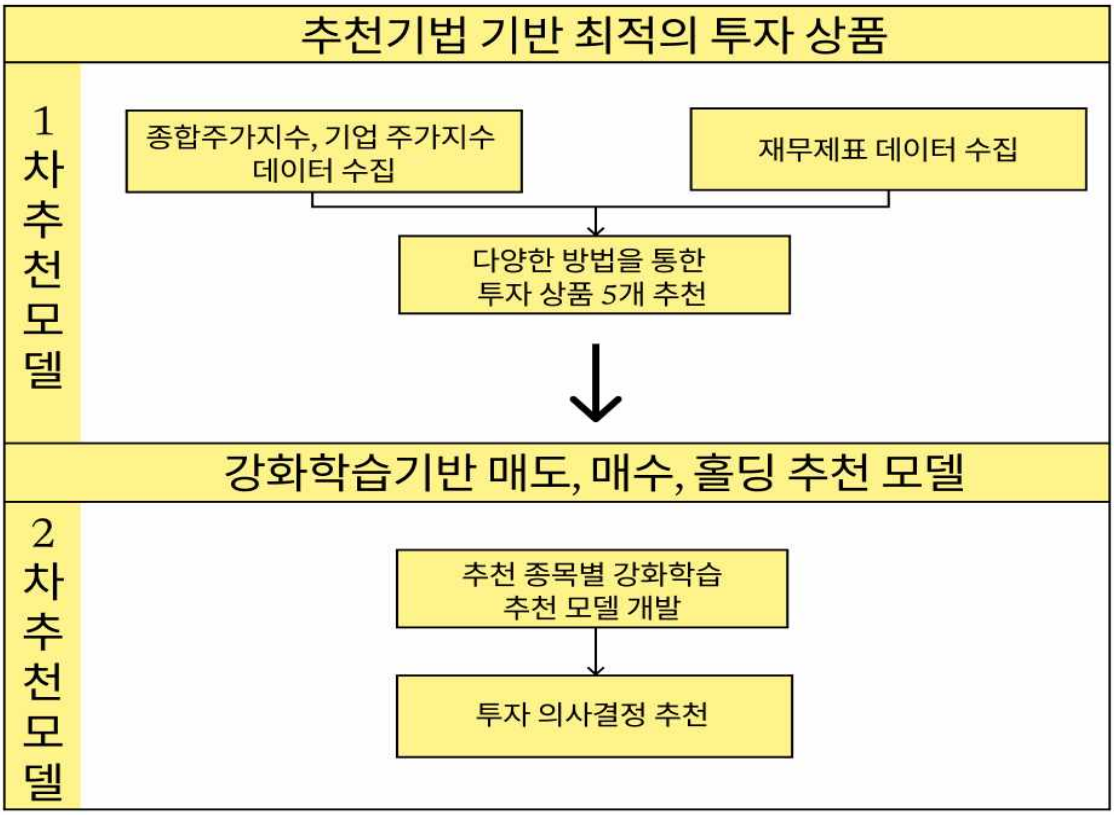
2. 프로젝트 정의 및 계획 설정

가. 프로젝트 정의

□ 본 프로젝트의 목표는 금융 데이터 분석과 투자자의 수익 극대화를 위한 종목 추천 및 강화학습 모델의 개발이다.

○ 금융 분야에서 다양한 분석 방법을 활용하여 투자자에게 필요한 정보를 제공한다. 금융 데이터 분석은 '기본적 분석', '기술적 분석', '정서 분석'이 있다. 기본적 분석은 기업의 가치, 산업, 경제를 평가하는 분석 방법이며, 기술적 분석은 흔히 차트 분석이라고도 하며 과거의 경험으로 미래를 예측하고자 하는 것이다. 마지막으로 정서분석은 주식 종목, 업종, 시장, 경기 등에 대한 투자자들의 정서를 파악하기 위한 분석 방법으로, 뉴스 분석, 토론방 분석, SNS 분석 등의 수많은 방법이 있다. 어느 하나가 좋다기보다는 성공적인 투자를 위해서는 이 모두를 활용해 투자를 결정을 내리는 것이 좋다.

나. 계획 설정



□ 본 프로젝트는 크게 2가지 절차로 진행된다. 첫 번째로 기본적 분석, 기술적 분석, 그리고 퀀트투자를 통해 하나의 종목을 선정하며, 최종으로 콘텐츠기반 필터링을 활용하여 5개의 종목을 추천받는다. 두 번째로, 이러한 추천된 종목 5개에 대해 강화학습 모델을 개발하여 매도, 매수, 홀딩 타이밍을 추천한다

3. 위험계획 수립

□ 데이터 수집 문제

- ○ 데이터 수집 과정에서 발생할 수 있는 문제를 고려. 데이터 누락, 오류, 형식 불일치와 같은 잠재적인 위험을 사전에 파악하고 대비책을 마련하여 원활한 데이터 수집을 보장한다

□ 모델 성능 부족

- ○ 모델의 성능이 예상에 미치지 못할 경우, 모델을 개선하기 위한 방안을 고려. 추가적인 피쳐 엔지니어링, 다른 모델 적용, 하이퍼파라미터 조정 등을 통해 모델 성능을 향상시키기 위한 계획 수립한다

Ⅲ 데이터 준비

1. 필요 데이터 정의

[데이터 수집 기간: 2017.01.01 - 2023.11.21]

□ 종합주가지수 (코스피 데이터)

- ○ 코스피의 일별 시가, 고가, 저가, 종가 데이터로 FinanceDataReader를 통해 수집 가능하다

□ 기업주가지수 (기업의 주식 가격 데이터)

- ○ 코스피 종목의 일별 시가, 고가, 저가, 종가 데이터로 FinanceDataReader를 통해 수집 가능하다

□ 등락률

- ○ 주식의 가격이 어제 대비 얼마나 증가했는지 또는 감소했는지 백분율로 표현함

□ 매출액 증가율 (Revenue Growth Rate):

- ○ 기업의 매출액 데이터가 필요하며, 최소 두 분기의 매출액 데이터가 필요합니다. 이 데이터는 해당 분기의 매출액이 전 분기 대비 얼마나 성장했는지 백분율로 나타낸다.

□ 영업이익 증가율 (Operating Income Growth Rate):

- ○ 기업의 영업이익 데이터가 필요하며, 최소 두 분기의 영업이익 데이터가 필요합니다. 이 데이터 역시 DART OPEN API를 통해 수집 가능

□ 부채비율 (Debt-to-Equity Ratio):

- ○ 기업의 부채총계와 자본총계 데이터가 필요하며, DART OPEN API를 통해 수집 가능하다

□ PER (Price-to-Earnings Ratio):

- ○ 현재 주가 / 주당순이익(EPS, Earning Per Share) 또는 시가총액 / 당기순이익을 뜻하며 투자한 만큼의 돈을 다시 회수하는데 걸리는 시간을 뜻하며 PER이 10이라면 10년이 걸린다는 뜻이다

□ PBR (Price-to-Book Ratio)

- ○ 주당순자산가치 대비 주가의 크기로 1주당 주당순자산가치의 몇 배로 거래되고 있는가를 말한다. PBR이 작을수록 주당순자산 대비 주가가 작다는 뜻으로 저평가되었다고 볼 수 있다

□ PSR (Price-to-Sales Ratio)

- ○ 낮은 PSR은 주식이 기업의 매출에 비해 저렴하다는 것을 나타내며 매출이 중요한 업종에서는 PSR이 높을 수 있으며, 업종 특성을 고려해야한다

□ POR (Price-to-Operating Income Ratio)

- ○ 낮은 POR은 주식이 운영 이익에 비해 저렴하다는 것을 나타내며 이익이 중요한 업

종에서는 POR이 높을 수 있으며, 업종과 기업의 재무 건강 상태를 고려해야 한다

□ PCR (Price-to-Cash Flow Ratio)

○ 주식이 현금 흐름에 비해 저렴하다는 것을 나타내며 현금 흐름은 기업의 현금 생성 능력을 반영한다

□ PFCR (Price-to-Free Cash Flow Ratio)

○ 낮은 PFCR은 주식이 무료 현금 흐름에 비해 저렴하다는 것을 나타내며 투자 가능한 현금을 나타낸다

□ NCAV/MK (Net Current Asset Value to Market Cap Ratio)

○ 주식의 안전 마진을 나타내는 지표로 1 미만이면 주식이 자산 가치에 비해 고평가되었을 수 있다

□ 기관 투자자, 외국인 투자자 순매수 거래량

○ 주식을 매수한 양에서 매도한 양을 뺀 값이며, 이는 해당 기관 또는 외국 투자자가 해당 기간 동안 주식에 대한 긍정적인 입장을 나타내며 한국거래소에서 수집 가능

□ 개인투자자, 기관투자자, 외국인 투자자 거래량

○ 시장 참여자들 간의 거래 활동을 종합적으로 파악하는 데 사용되며 한국거래소에서 수집 가능

□ 한국 국채 3년물

○ 국채는 정부가 자금을 조달하기 위해 발행하는 채권으로, 국채 금리는 시장 금리의 중요한 지표 중 하나이며 .investing.com에서 수집 가능

2. 데이터 수집 및 전처리

가. 데이터 수집

□ 증권사 API & FinanceDataReader

○ 종합주가지수(시가, 고가, 저가, 종가)

○ 기업주가지수(시가, 고가, 저가, 종가)

□ DART OPEN API

○ 재무제표 데이터

[유동자산, 부채총계, 자본총계, 매출액, 영업이익, 당기순이익, 영업활동현금, 잉여현금흐름, 시가총액]

□ 크롤링

○ 상장법인 목록

○ 한국 국채 3년물

○ 투자자별 매매 실적

나. 데이터 전처리

데이터 전처리			
수집 데이터		파생변수 생성	
데이터 설명	수집방법	데이터 설명	산출 방법
종합주가지수 시가, 고가, 저가, 종가	FinanceDataReader & 증권사 API	등락률	((오늘 주가 - 어제 주가) / 어제 주가) * 100
기업주가지수 시가, 고가, 저가, 종가	FinanceDataReader & 증권사 API	매출액 증가률	((당분기 매출액 - 전분기 매출액) / 전분기 매출액) * 100
기업주가지수 시가, 고가, 저가, 종가	FinanceDataReader & 증권사 API	전일 종가 대비 당일 시가, 종가 비율	(당일 시가, 종가 - 전일 종가) / 전일 종가
기업주가지수 시가, 고가, 저가, 종가	FinanceDataReader & 증권사 API	당일 종가 대비 당일 고가, 저가 비율	(당일 고가, 저가 - 당일 종가) / 당일 종가
기업주가지수 시가, 고가, 저가, 종가	FinanceDataReader & 증권사 API	전일 거래량 대비 당일 거래량 비율	(당일 거래량 - 전일 거래량) / 전일 거래량
기업주가지수 시가, 고가, 저가, 종가	FinanceDataReader & 증권사 API	5, 10, 20, 60, 120일 이동평균별 종가 대비 당일 종가	(당일 종가 - 이동평균별 종가) / 이동평균별 종가
기업주가지수 시가, 고가, 저가, 종가	FinanceDataReader & 증권사 API	5, 10, 20, 60, 120일 이동평균별 거래량 대비 당일 거래량	(당일 거래량 - 이동평균별 거래량) / 이동평균별 거래량
상장법인목록	한국거래소 크롤링	영업이익 증가율	((당분기 영업이익 - 전분기 영업이익) / 전분기 영업이익) * 100
유동자산	DART OPEN API	부채비율	(부채총계 / 자본총계) * 100
부채총계	DART OPEN API	PER	시가총액 / 최근 4분기 동안의 당기순이익 합
자본총계	DART OPEN API	PBR	시가총액 / 부채총계
매출액	DART OPEN API	ROE	PER/PBR

데이터 전처리			
수집 데이터		파생변수 생성	
데이터 설명	수집방법	데이터 설명	산출 방법
영업이익	DART OPEN API	PCR	시가총액 / 최근 4 분기 동안의 영업활동현금흐름 합
당기순이익	DART OPEN API	PSR	시가총액 / 최근 4 분기 동안의 매출액 합
영업활동현금흐름	DART OPEN API	POR	시가총액 / 최근 4 분기 동안의 영업이익 합
잉여현금흐름	DART OPEN API	PFCR	시가총액 / 최근 4 분기 동안의 잉여현금흐름 합
시가총액	DART OPEN API	NCAV/MK	(유동자산 - 부채총계) / 시가총액
투자자별 매매실적	한국거래소 크롤링	기관투자자 순매수 거래량의 전일 대비 당일 비율	(당일 기관투자자 순매수 거래량 / 전일 기관투자자 순매수 거래량) * 100
투자자별 매매실적	한국거래소 크롤링	기관투자자 순매수 거래량 5, 10, 20, 60, 120일 이동평균별 대비 당일 비율	(당일 거래량 / 5,10,20,60,120일 이동평균별 기관투자자 순매수 거래량) * 100
투자자별 매매실적	한국거래소 크롤링	외국인투자자 순매수 거래량의 전일 대비 당일 비율	(당일 외국인투자자 순매수 거래량 / 전일 기관투자자 순매수 거래량) * 100
투자자별 매매실적	한국거래소 크롤링	외국인투자자 순매수 거래량 5, 10, 20, 60, 120일 이동평균별 대비 당일 비율	(당일 거래량 / 5,10,20,60,120일 이동평균별 외국인투자자 순매수 거래량) * 100

데이터 전처리			
수집 데이터		파생변수 생성	
데이터 설명	수집방법	데이터 설명	산출 방법
투자자별 매매실적	한국거래소 크롤링	전일 개인투자자 거래량과 당일 개인 투자자 거래량 차이	당일 개인투자자 거래량 – 전일 개인투자자 거래량
투자자별 매매실적	한국거래소 크롤링	개인투자자 거래량 5, 10, 20, 60, 120일 이동평균별 전일 대비 비율	(당일 개인투자자 거래량 - 이동평균별 거래량) / 이동평균별 거래량
투자자별 매매실적	한국거래소 크롤링	전일 기관투자자 거래량과 당일 개인 투자자 거래량 차이	당일 기관투자자 거래량 - 전일 기관투자자 거래량
투자자별 매매실적	한국거래소 크롤링	기관투자자 거래량 5, 10, 20, 60, 120일 이동평균별 전일 대비 비율	(당일 기관투자자 거래량 - 이동평균별 거래량) / 이동평균별 거래량
투자자별 매매실적	한국거래소 크롤링	전일 외국인투자자 거래량과 당일 외국인투자자 거래량 차이	당일 외국인투자자 거래량 - 전일 외국인투자자 거래량
투자자별 매매실적	한국거래소 크롤링	외국인투자자 거래량 5, 10, 20, 60, 120일 이동평균별 전일 대비 비율	(당일 외국인투자자 거래량 - 이동평균별 거래량) / 이동평균별 거래량
한국 국채 3년물	investing 크롤링	5, 10, 20, 60, 120일 이동평균 한국 국채 3년물 대비 당일 한국 국채 3년물	(당일 한국 국채 3년물 – 이동평균별 한국 국채 3년물) / 이동평균별 한국 국채 3년물
종합주가지수 시가, 고가, 저가, 종가	FinanceDataReader & 증권사 API	5, 10, 20, 60, 120일 이동평균별 KOSPI 대비 당일 KOSPI	

3. 품질 점검

□ 결측값 확인

○ 이동평균별로 구한 데이터가 있기에 2017년에는 결측값이 생길 수밖에 없다. 2018년도 데이터부터 학습에 사용할 것이기에 2017년 데이터는 삭제

	date	market_kospi_ma5_ratio	market_kospi_ma20_ratio	market_kospi_ma60_ratio	market_kospi_ma120_ratio	bond_k3y_ma5_ratio	bond_k3y_ma20_ratio	bond_k3y_ma60_ratio	bond_k3y_ma120_ratio
0	20170102	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	20170103	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	20170104	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	20170105	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	20170106	1.003798	NaN	NaN	NaN	1.003939	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...
1685	20231115	1.022444	1.044242	1.007096	0.982494	0.974612	0.948600	0.946083	0.974483
1686	20231116	1.017950	1.043290	1.007923	0.983314	0.973309	0.944370	0.938054	0.965279
1687	20231117	1.005499	1.033550	1.000732	0.976384	0.976302	0.943500	0.933295	0.959333
1688	20231120	1.007021	1.039565	1.009503	0.985071	0.995767	0.955070	0.932640	0.955462
1689	20231121	1.008499	1.044819	1.017328	0.992816	0.992215	0.953876	0.927257	0.948891

□ 중복된 데이터 확인

○ 재무제표를 확인해본 결과 중복된 데이터가 있는 것을 확인함. 중복된 데이터 삭제 후 진행

```
stock_list[stock_list['stocks'] == '005930'].head(50)
```

	stocks	유동자산	부채총계	자본총계	매출액	영업이익	당기순이익	영업활동현금흐름	잉여현금흐름	시가총액
2017-03-31	005930	129284204000000	74399417000000	189817955000000	50547526000000	9898361000000	7684354000000	10597271000000	18762410000000	289799434220000
2017-03-31	005930	129284204000000	74399417000000	189817955000000	50547526000000	9898361000000	7684354000000	10597271000000	18762410000000	289799434220000
2017-06-30	005930	132172681000000	76883688000000	200705748000000	61000537000000	14066547000000	11053851000000	23023433000000	39126155000000	310599010238000
2017-03-31	005930	129284204000000	74399417000000	189817955000000	50547526000000	9898361000000	7684354000000	10597271000000	18762410000000	289799434220000
2017-06-30	005930	132172681000000	76883688000000	200705748000000	61000537000000	14066547000000	11053851000000	23023433000000	39126155000000	310599010238000
2017-09-30	005930	145322337000000	85887328000000	210691251000000	62048901000000	14533159000000	11193411000000	40470495000000	71527845000000	332726418616000
2017-03-31	005930	129284204000000	74399417000000	189817955000000	50547526000000	9898361000000	7684354000000	10597271000000	18762410000000	289799434220000
2017-06-30	005930	132172681000000	76883688000000	200705748000000	61000537000000	14066547000000	11053851000000	23023433000000	39126155000000	310599010238000

□ 데이터 형식 및 타입 확인

○ 데이터 형식 확인, 강화학습(딥러닝)에 사용하기 위해서는 int, float 형태만 가능하기에 int와 float형태로 변환

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1692 entries, 0 to 1691
Data columns (total 45 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   date                    1692 non-null   int64
 1   open                    1692 non-null   int64
 2   high                    1692 non-null   int64
 3   low                     1692 non-null   int64
 4   close                   1692 non-null   int64
 5   volume                  1692 non-null   int64
 6   per                     1692 non-null   float64
 7   pbr                     1692 non-null   float64
 8   roe                     1692 non-null   float64
 9   open_lastclose_ratio    1691 non-null   float64
 10  high_close_ratio        1692 non-null   float64
 11  low_close_ratio         1692 non-null   float64
 12  diffratio               1600 non-null   float64
 13  volume_lastvolume_ratio 1691 non-null   float64
 14  close_ma5_ratio         1688 non-null   float64
 15  volume_ma5_ratio        1688 non-null   float64
 16  close_ma10_ratio        1683 non-null   float64
 17  volume_ma10_ratio       1683 non-null   float64
 18  close_ma20_ratio        1673 non-null   float64
 19  volume_ma20_ratio       1673 non-null   float64
 20  close_ma60_ratio        1633 non-null   float64
 21  volume_ma60_ratio       1633 non-null   float64
 22  close_ma120_ratio       1573 non-null   float64
 23  volume_ma120_ratio      1573 non-null   float64
 24  ind                     1692 non-null   float64
 25  ind_diff                1692 non-null   float64
 26  ind_ma5                 1692 non-null   float64
 27  ind_ma10                1692 non-null   float64
 28  ind_ma20                1692 non-null   float64
 29  ind_ma60                1692 non-null   float64
 30  ind_ma120               1692 non-null   float64
 31  inst                    1692 non-null   float64
 32  inst_diff               1692 non-null   float64
 33  inst_ma5                1692 non-null   float64
 34  inst_ma10               1692 non-null   float64
 35  inst_ma20               1692 non-null   float64
 36  inst_ma60               1692 non-null   float64
 37  inst_ma120              1692 non-null   float64
 38  foreign                 1692 non-null   float64
 39  foreign_diff            1692 non-null   float64
 40  foreign_ma5             1692 non-null   float64
 41  foreign_ma10            1692 non-null   float64
 42  foreign_ma20            1692 non-null   float64
 43  foreign_ma60            1692 non-null   float64
 44  foreign_ma120           1692 non-null   float64
dtypes: float64(39), int64(6)
memory usage: 595.0 KB
```

Ⅳ 데이터 분석

1. 분석용 데이터 준비

가. 데이터 준비

□ KOSPI, 국채

KOSPI

	Date	Open	High	Low	Close	Adj Close	Volume
0	2017-01-02	2022.229980	2031.790039	2015.680054	2026.160034	2026.160034	229900
1	2017-01-03	2034.310059	2044.069946	2028.469971	2043.969971	2043.969971	268100
2	2017-01-04	2046.290039	2046.290039	2040.609985	2045.640015	2045.640015	371500
3	2017-01-05	2045.520020	2046.500000	2039.489990	2041.949951	2041.949951	541300
4	2017-01-06	2048.110107	2051.840088	2045.660034	2049.120117	2049.120117	455400
...	...	...	...	...	...	...	...
1685	2023-11-15	2482.209961	2487.419922	2468.429932	2486.669922	2486.669922	419800
1686	2023-11-16	2483.479980	2491.979980	2472.689941	2488.179932	2488.179932	407400
1687	2023-11-17	2477.429932	2481.100098	2463.590088	2469.850098	2469.850098	388200
1688	2023-11-20	2464.719971	2499.750000	2464.040039	2491.199951	2491.199951	323800
1689	2023-11-21	2504.699951	2517.739990	2500.909912	2510.419922	2510.419922	345055

1690 rows × 7 columns

국채

	date	bond_k3y_ma5_ratio	bond_k3y_ma20_ratio	bond_k3y_ma60_ratio	bond_k3y_ma120_ratio
0	20170102	NaN	NaN	NaN	NaN
1	20170103	NaN	NaN	NaN	NaN
2	20170104	NaN	NaN	NaN	NaN
3	20170105	NaN	NaN	NaN	NaN
4	20170106	1.003939	NaN	NaN	NaN
...	...	...	...	...	...
2409	20231117	0.976302	0.943500	0.933295	0.959333
2410	20231118	0.986824	0.948776	0.934611	0.959681
2411	20231119	0.996107	0.953811	0.935422	0.959508
2412	20231120	0.995767	0.955070	0.932640	0.955462
2413	20231121	0.992215	0.953876	0.927257	0.948891

2414 rows × 5 columns

□ 종목별 기업주가지수, 종목별 투자자별 매매실적

한솔PNS

	Date	Open	High	Low	Close	Volume	Change
0	2017-01-02	14275	14321	13953	14138	122844	-0.006465
1	2017-01-03	14137	15058	14137	14966	391866	0.058566
2	2017-01-04	14920	15012	14459	14690	259208	-0.018442
3	2017-01-05	14736	15242	14644	14782	311804	0.006263
4	2017-01-06	14828	14920	14598	14644	153264	-0.009336
...	...	...	...	...	...	...	...
1689	2023-11-15	6690	6690	6560	6610	91875	0.000000
1690	2023-11-16	6640	6700	6510	6650	57971	0.006051
1691	2023-11-17	6610	6680	6490	6550	65594	-0.015038
1692	2023-11-20	6570	6590	6480	6570	38915	0.003053
1693	2023-11-21	6580	6750	6580	6640	94175	0.010654

1694 rows × 7 columns

한솔PNS_2

	date	inst	corp_etc	ind	foreign	all	Code
0	2017-01-02	-42,702,000	0	207,063,400	-164,361,400	0	4710
1	2017-01-03	-107,333,700	-19,794,000	-903,820,050	1,030,947,750	0	4710
2	2017-01-04	270,817,900	4,680,000	603,485,850	-878,983,750	0	4710
3	2017-01-05	86,816,650	1,625,000	-113,023,200	24,581,550	0	4710
4	2017-01-06	-145,869,500	-1,517,800	970,185,400	-822,798,100	0	4710
...	...	...	...	...	...	...	...
1689	2023-11-15	11,829,740	2,631,150	53,198,580	-67,659,470	0	4710
1690	2023-11-16	2,519,760	5,950,500	3,633,070	-12,103,330	0	4710
1691	2023-11-17	-1,139,860	-1,644,000	62,993,850	-60,209,990	0	4710
1692	2023-11-20	4,617,050	-32,576,330	60,103,160	-32,143,880	0	4710
1693	2023-11-21	-14,088,470	-2,873,000	30,528,490	-13,567,020	0	4710

1694 rows × 7 columns

써니전자

	Date	Open	High	Low	Close	Volume	Change
0	2017-01-02	3900	3910	3750	3810	937803	-0.061576
1	2017-01-03	3820	3885	3755	3845	574701	0.009186
2	2017-01-04	3895	3950	3825	3860	411360	0.003901
3	2017-01-05	3875	3955	3870	3940	353649	0.020725
4	2017-01-06	3950	3950	3890	3900	297192	-0.010152
...	...	...	...	...	...	...	...
1689	2023-11-15	2410	2425	2375	2385	169977	0.000000
1690	2023-11-16	2390	2435	2380	2430	394755	0.018868
1691	2023-11-17	2415	2435	2400	2405	227240	-0.010288
1692	2023-11-20	2435	2435	2405	2410	205711	0.002079
1693	2023-11-21	2400	2430	2385	2425	195751	0.006224

1694 rows × 7 columns

써니전자_2

	date	inst	corp_etc	ind	foreign	all	Code
0	2017-01-02	354,680	-5,599,380	384,209,915	-378,965,215	0	4770
1	2017-01-03	0	8,833,500	-24,360,745	15,527,245	0	4770
2	2017-01-04	18,592,270	0	52,550,165	-71,142,435	0	4770
3	2017-01-05	0	39,427,205	-41,380,770	1,953,565	0	4770
4	2017-01-06	-94,080	-37,879,195	136,537,595	-98,564,320	0	4770
...	...	...	...	...	...	...	...
1689	2023-11-15	87,880	0	9,712,445	-9,800,325	0	4770
1690	2023-11-16	-14,976,380	-34,740,550	-43,306,080	93,023,010	0	4770
1691	2023-11-17	-414,480	0	36,371,315	-35,956,835	0	4770
1692	2023-11-20	1,582,420	0	-25,452,535	23,870,115	0	4770
1693	2023-11-21	3,892,690	0	-81,438,170	77,545,480	0	4770

1694 rows × 7 columns

경인전자

	Date	Open	High	Low	Close	Volume	Change
0	2017-01-02	22700	23000	22450	23000	760	0.019956
1	2017-01-03	23350	23400	22900	23400	3987	0.017391
2	2017-01-04	23400	23700	22900	23500	1336	0.004274
3	2017-01-05	23400	23400	22550	22550	1551	-0.040426
4	2017-01-06	22850	22850	22300	22300	350	-0.011086
...	...	...	...	...	...	...	...
1689	2023-11-15	21550	21750	21250	21650	4607	0.004640
1690	2023-11-16	21450	21700	21050	21350	4763	-0.013857
1691	2023-11-17	21350	21600	21050	21550	4324	0.009368
1692	2023-11-20	21550	21550	21050	21050	5697	-0.023202
1693	2023-11-21	21100	21500	21050	21500	2419	0.021378

1694 rows × 7 columns

롯데에너지머티리얼즈

	Date	Open	High	Low	Close	Volume	Change
0	2017-01-02	12925	13070	12392	12393	646882	-0.082408
1	2017-01-03	12537	12779	12392	12586	327433	0.015573
2	2017-01-04	12634	12779	12198	12296	308171	-0.023041
3	2017-01-05	12392	12586	12198	12586	166312	0.023585
4	2017-01-06	12634	12683	12440	12586	107246	0.000000
...	...	...	...	...	...	...	...
1689	2023-11-15	42950	43250	41400	41950	244158	0.008413
1690	2023-11-16	42000	42100	40900	41950	132767	0.000000
1691	2023-11-17	42100	42100	40900	41350	129501	-0.014303
1692	2023-11-20	41400	43800	41200	43050	180603	0.041112
1693	2023-11-21	43500	45450	42950	45000	297371	0.045296

1694 rows × 7 columns

포스코퓨처엠

	Date	Open	High	Low	Close	Volume	Change
0	2017-01-02	11309	11309	10981	11263	53734	0.000000
1	2017-01-03	11215	11215	10934	11028	98419	-0.020865
2	2017-01-04	11357	11544	11263	11357	135147	0.029833
3	2017-01-05	11357	11403	11216	11357	88358	0.000000
4	2017-01-06	11356	11544	11215	11544	96569	0.016466
...	...	...	...	...	...	...	...
1689	2023-11-15	317500	320000	293000	297500	1120565	0.006768
1690	2023-11-16	297000	303000	289000	293500	701953	-0.013445
1691	2023-11-17	288000	293000	283000	286000	552165	-0.025554
1692	2023-11-20	284500	305000	284000	300500	761493	0.050699
1693	2023-11-21	303000	306500	295500	303500	580537	0.009983

1694 rows × 7 columns

경인전자_2

	date	inst	corp_etc	ind	foreign	all	Code
0	2017-01-02	1,364,800	0	-1,364,800	0	0	9140
1	2017-01-03	0	0	34,561,200	-34,561,200	0	9140
2	2017-01-04	253,650	0	-253,650	0	0	9140
3	2017-01-05	0	0	8,847,500	-8,847,500	0	9140
4	2017-01-06	0	0	0	0	0	9140
...	...	...	...	...	...	...	...
1689	2023-11-15	43,450	0	2,929,200	-2,972,650	0	9140
1690	2023-11-16	-85,400	961,850	13,266,700	-14,143,150	0	9140
1691	2023-11-17	-21,150	0	10,517,550	-10,496,400	0	9140
1692	2023-11-20	35,200	42,650	-19,599,000	19,521,150	0	9140
1693	2023-11-21	-1,150	0	-12,491,750	12,492,900	0	9140

1694 rows × 7 columns

롯데에너지머티리얼즈_2

	date	inst	corp_etc	ind	foreign	all	Code
0	2017-01-02	-1,389,751,900	-72,346,000	1,394,537,950	67,559,950	0	20150
1	2017-01-03	-1,307,647,700	138,330,000	829,025,300	340,292,400	0	20150
2	2017-01-04	-456,035,550	132,541,100	284,512,000	38,982,450	0	20150
3	2017-01-05	21,848,100	0	-319,373,250	297,525,150	0	20150
4	2017-01-06	-6,049,350	-130,124,600	190,225,550	-54,051,600	0	20150
...	...	...	...	...	...	...	...
1689	2023-11-15	434,869,300	204,069,800	358,092,750	-997,031,850	0	20150
1690	2023-11-16	-1,066,061,500	56,453,150	215,944,150	793,664,200	0	20150
1691	2023-11-17	-747,983,250	4,909,350	579,284,450	163,789,450	0	20150
1692	2023-11-20	-395,744,300	394,428,200	-1,356,532,550	1,357,848,650	0	20150
1693	2023-11-21	2,287,040,200	102,729,000	-3,395,576,700	1,005,807,500	0	20150

1694 rows × 7 columns

포스코퓨처엠_2

	date	inst	corp_etc	ind	foreign	all	Code
0	2017-01-02	-106,656,400	51,443,500	93,198,550	-37,985,650	0	3670
1	2017-01-03	-528,033,600	0	377,161,750	150,871,850	0	3670
2	2017-01-04	-115,527,600	-2,430,000	-123,896,700	241,854,300	0	3670
3	2017-01-05	-58,413,000	0	61,325,750	-2,912,750	0	3670
4	2017-01-06	299,430,500	1,488,000	-332,216,600	31,298,100	0	3670
...	...	...	...	...	...	...	...
1689	2023-11-15	7,951,380,500	-1,234,747,000	42,333,066,500	-49,049,700,000	0	3670
1690	2023-11-16	1,808,552,000	-187,199,500	13,430,749,500	-15,052,102,000	0	3670
1691	2023-11-17	-2,403,186,000	-264,468,500	10,995,079,500	-8,327,425,000	0	3670
1692	2023-11-20	21,857,591,500	564,424,500	-27,705,837,500	5,283,821,500	0	3670
1693	2023-11-21	-2,416,624,000	-320,750,500	-10,962,410,000	13,699,784,500	0	3670

1694 rows × 7 columns

□ 종목별 재무제표

```
stock_list.drop_duplicates()
```

	stocks	유동자산	부채총계	자본총계	매출액	영업이익	당기순이익	영업활동현금흐름	잉여현금흐름	시가총액
2017-03-31	005930	129284204000000	74399417000000	189817955000000	50547526000000	9898361000000	7684354000000	10597271000000	18762410000000	289799434220000
2017-06-30	005930	132172681000000	76883688000000	200705748000000	61000537000000	14066547000000	11053851000000	23023433000000	39126155000000	310599010238000
2017-09-30	005930	145322337000000	85887328000000	210691251000000	62048901000000	14533159000000	11193411000000	40470495000000	71527845000000	332726418616000
2018-03-31	005930	154941953000000	89213233000000	223259880000000	60563714000000	15642170000000	11688544000000	15616352000000	22989688000000	315959161734000
2018-06-30	005930	156976839000000	85563535000000	233124845000000	58482658000000	14869035000000	11043412000000	29054126000000	48609201000000	299461497255000
...	...	...	...	...	...	...	...	...	...	...
2022-06-30	001210	56347835795	64517679122	17800950671	12526678787	-2452194181	-3681888462	-6344946020	-4179678124	32380432700
2022-09-30	001210	56080535008	65655762329	16342298401	10473977706	-2806981482	-5722573874	-7427497448	-5244694917	28656358640
2022-12-30	001210	36701210379	48974883827	12839330125	11556496277	-3599027516	-1589200854	-8782313605	-16326786145	25349855720
2023-03-31	001210	28806247318	39123366582	15565830100	15456646029	-1199239657	-1928085563	772860031	-2292529062	27817474242
2023-06-30	001210	24606553767	37774251610	14793697610	12942235121	-1353053566	-1433511859	633213487	-2763543278	42225841028

983 rows × 10 columns

나. 데이터 전처리 후 병합

□ 종목별 재무제표 전처리

```
df4 = pd.DataFrame(columns=['stocks','Date', 'PER','PBR','PSR','POR','PCR','PFCR','NCAV/MK'], index=['1900-01-01'])
```

```
for i in range(length):
    PER = df2.iloc[i]['시가총액'] / (df2.iloc[i-3]['당기순이익'] + df2.iloc[i-2]['당기순이익'] + \
          df2.iloc[i-1]['당기순이익'] + df2.iloc[i]['당기순이익'])
    PBR = df2.iloc[i]['시가총액'] / df2.iloc[i]['부채총계']
    PSR = df2.iloc[i]['시가총액'] / (df2.iloc[i-3]['매출액'] + df2.iloc[i-2]['매출액'] + \
          df2.iloc[i-1]['매출액'] + df2.iloc[i]['매출액'])
    POR = df2.iloc[i]['시가총액'] / (df2.iloc[i-3]['영업이익'] + df2.iloc[i-2]['영업이익'] + \
          df2.iloc[i-1]['영업이익'] + df2.iloc[i]['영업이익'])
    PCR = df2.iloc[i]['시가총액'] / (df2.iloc[i-3]['영업활동현금흐름'] + df2.iloc[i-2]['영업활동현금흐름'] + \
          df2.iloc[i-1]['영업활동현금흐름'] + df2.iloc[i]['영업활동현금흐름'])
    PFCR = df2.iloc[i]['시가총액'] / (df2.iloc[i-3]['잉여현금흐름'] + df2.iloc[i-2]['잉여현금흐름'] + \
          df2.iloc[i-1]['잉여현금흐름'] + df2.iloc[i]['잉여현금흐름'])
    NCAV_MK = (df2.iloc[i]['유동자산'] - df2.iloc[i]['부채총계']) / df2.iloc[i]['시가총액']

    # 종목 코드 (stocks) 열 추가
    stocks = df2.iloc[i]['stocks']
    Date = df2.iloc[i]['Unnamed: 0']
    df4.loc[i] = [stocks, Date, PER, PBR, PSR, POR, PCR, PFCR, NCAV_MK]


df4.reset_index(drop=True, inplace=True) # 인덱스 리셋
```

□ KOSPI, 기업주가지수 전처리

```
def preprocess(data):
    windows = [5, 10, 20, 60, 120]
    for window in windows:
        data[f'close_ma{window}'] = data['close'].rolling(window).mean()
        data[f'volume_ma{window}'] = data['volume'].rolling(window).mean()
        data[f'close_ma{window}_ratio'] = \
            (data['close'] - data[f'close_ma{window}']) / data[f'close_ma{window}']
        data[f'volume_ma{window}_ratio'] = \
            (data['volume'] - data[f'volume_ma{window}']) / data[f'volume_ma{window}']

    data['open_lastclose_ratio'] = np.zeros(len(data))
    data.loc[1:, 'open_lastclose_ratio'] = \
        (data['open'][1:].values - data['close'][:-1].values) / data['close'][:-1].values
    data['high_close_ratio'] = (data['high'].values - data['close'].values) / data['close'].values
    data['low_close_ratio'] = (data['low'].values - data['close'].values) / data['close'].values
    data['close_lastclose_ratio'] = np.zeros(len(data))
    data.loc[1:, 'close_lastclose_ratio'] = \
        (data['close'][1:].values - data['close'][:-1].values) / data['close'][:-1].values
    data['volume_lastvolume_ratio'] = np.zeros(len(data))
    data.loc[1:, 'volume_lastvolume_ratio'] = (
        (data['volume'][1:].values - data['volume'][:-1].values)
        / data['volume'][:-1].replace(to_replace=0, method='ffill')\
            .replace(to_replace=0, method='bfill').values
    )
    return data
```

```
def preprocess_3(data):
    investors = ['ind', 'inst', 'foreign']
    windows = [5, 10, 20, 60, 120]

    for investor in investors:
        for window in windows:
            # 이동평균 생성
            data[f'{investor}_ma{window}'] = data[investor].rolling(window).mean()

            # 이동평균 비율 생성
            data[f'{investor}_ma{window}_ratio'] = (
                data[investor] - data[f'{investor}_ma{window}']
            ) / data[f'{investor}_ma{window}']

            # 전일과 당일 거래량 차이 생성
            data[f'{investor}_diff'] = data[investor].diff()

    return data
```

□ 기업별로 병합

○ 한솔PNS

한솔PNS

	date	open	high	low	close	volume	per	pbr	roe	open_lastclose_ratio	...	inst_ma20	inst_ma60	inst_ma120	foreign	foreign_diff	foreign_ma5	foreign_ma10	foreign_ma20	foreign_ma60	foreign_ma120
0	20170102	14275	14321	13953	14137	122844	12.43	1.14	9.70	0.003233	...	-0.021736	-0.059990	-0.072447	-0.092499	-0.110061	-0.065051	0.039666	0.087289	0.033988	0.001916
1	20170103	14137	15058	14137	14966	391866	12.43	1.14	9.70	0.000000	...	-0.022088	-0.060647	-0.073346	0.166108	0.258607	0.006583	0.025773	0.094919	0.036032	0.001552
2	20170104	14920	15012	14459	14689	259208	12.43	1.14	9.70	-0.003074	...	-0.022038	-0.059150	-0.071418	-0.213296	-0.379404	-0.047086	-0.018249	0.066629	0.037467	0.001391
3	20170105	14736	15242	14643	14782	311804	12.43	1.14	9.70	0.003200	...	-0.017832	-0.055320	-0.069938	0.004346	0.217642	-0.023556	-0.032598	0.067449	0.039860	-0.000082
4	20170106	14828	14920	14597	14643	153264	12.43	1.14	9.70	0.003112	...	-0.027074	-0.052929	-0.071807	-0.336694	-0.341039	-0.094407	-0.086329	0.049881	0.035101	-0.003771
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1687	20231115	6690	6690	6560	6610	91875	11.13	0.52	5.04	0.012103	...	0.031005	0.017955	0.012457	-0.111097	-0.197202	-0.014268	-0.008367	-0.070974	-0.086732	-0.002084
1688	20231116	6640	6700	6510	6650	57971	11.13	0.52	5.04	0.004539	...	0.031186	0.017577	0.012516	-0.013697	0.097400	0.012003	-0.010859	-0.052806	-0.087145	-0.003989
1689	20231117	6610	6680	6490	6550	65594	11.13	0.52	5.04	-0.006015	...	0.031152	0.017632	0.012499	-0.138458	-0.124761	-0.067022	-0.033514	-0.037947	-0.087468	-0.007163
1690	20231120	6570	6590	6480	6570	38915	11.13	0.52	5.04	0.003053	...	0.025246	0.016602	0.012654	-0.126121	0.012337	-0.060653	-0.042181	-0.043184	-0.083730	-0.010048
1691	20231121	6580	6750	6580	6640	94142	11.13	0.52	5.04	0.001522	...	0.024599	0.015400	0.012484	-0.030603	0.095518	-0.083995	-0.025800	-0.041428	-0.082262	-0.012193

1692 rows × 45 columns

○ 써니전자

써니전자

	date	open	high	low	close	volume	per	pbr	roe	open_lastclose_ratio	...	inst_ma20	inst_ma60	inst_ma120	foreign	foreign_diff	foreign_ma5	foreign_ma10	foreign_ma20	foreign_ma60	foreign_ma120
0	20170102	3900	3910	3750	3810	937803	30.03	2.78	9.92	-0.039409	...	0.000054	-0.000916	-0.008795	-0.108642	-0.111989	0.027176	0.052921	0.033992	0.014366	0.007924
1	20170103	3820	3885	3755	3845	574701	30.03	2.78	9.92	0.002625	...	0.000056	-0.000665	-0.008795	0.008791	0.117433	0.047459	0.038581	0.030902	0.014261	0.008564
2	20170104	3895	3950	3825	3860	411360	30.03	2.78	9.92	0.013004	...	0.000719	-0.000216	-0.008698	-0.042970	-0.051760	0.029968	0.035675	0.020836	0.013512	0.008961
3	20170105	3875	3955	3870	3940	353649	30.03	2.78	9.92	0.003886	...	0.000778	-0.000011	-0.008698	0.000894	0.043863	-0.027716	0.031815	0.015473	0.013361	0.008119
4	20170106	3950	3950	3890	3900	297192	30.03	2.78	9.92	0.002538	...	0.000774	-0.000013	-0.008698	-0.095211	-0.096105	-0.047428	0.016869	0.016025	0.011790	0.006452
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1687	20231115	2410	2425	2375	2385	169977	17.95	1.23	7.12	0.010482	...	-0.000861	0.001077	0.001252	-0.026056	-0.478789	-0.009862	0.031790	0.036407	0.023806	0.023772
1688	20231116	2390	2435	2380	2430	394755	17.95	1.23	7.12	0.002096	...	-0.001646	0.000815	0.001123	0.123510	0.149566	-0.013491	0.025326	0.038089	0.021018	0.022660
1689	20231117	2415	2435	2400	2405	227240	17.95	1.23	7.12	-0.006173	...	-0.001688	0.000805	0.001122	-0.065380	-0.188890	0.006118	0.008944	0.022764	0.016284	0.021343
1690	20231120	2435	2435	2405	2410	205711	17.95	1.23	7.12	0.012474	...	-0.003845	0.000857	0.001149	0.047980	0.113360	0.106557	0.012803	0.029821	0.014697	0.022171
1691	20231121	2400	2430	2385	2425	194738	17.95	1.23	7.12	-0.004149	...	-0.003437	0.000996	0.001209	0.164433	0.116453	0.048897	0.045392	0.030998	0.019979	0.024443

1692 rows × 45 columns

○ 경인전자

경인전자

	date	open	high	low	close	volume	per	pbr	roe	open_lastclose_ratio	...	inst_ma20	inst_ma60	inst_ma120	foreign	foreign_diff	foreign_ma5	foreign_ma10	foreign_ma20	foreign_ma60	foreign_ma120
0	20170102	22700	23000	22450	23000	760	29.14	0.60	2.49	0.006652	...	0.105560	0.100172	0.071634	0.000000	0.000000	0.000000	0.000000	-0.124959	-0.081525	-0.066292
1	20170103	23350	23400	22900	23400	3987	29.14	0.60	2.49	0.015217	...	0.105560	0.100172	0.071634	-0.376223	-0.376223	-0.075245	-0.037622	-0.143770	-0.087796	-0.069427
2	20170104	23400	23700	22900	23500	1336	29.14	0.60	2.49	0.000000	...	0.105972	0.100309	0.071703	0.000000	0.376223	-0.075245	-0.037622	-0.143770	-0.087796	-0.065013
3	20170105	23400	23400	22550	22550	1551	29.14	0.60	2.49	-0.004255	...	0.079420	0.100309	0.071703	-0.247582	-0.247582	-0.124761	-0.062380	-0.156149	-0.091089	-0.067076
4	20170106	22850	22850	22300	22300	350	29.14	0.60	2.49	0.013304	...	0.079420	0.100309	0.071715	0.000000	0.247582	-0.124761	-0.062380	-0.156149	-0.091089	-0.067076
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1687	20231115	21550	21750	21250	21650	4607	14.04	0.49	4.38	0.000000	...	0.013031	0.014338	0.007035	-0.029737	-0.111655	-0.084148	-0.038907	-0.063313	-0.021647	-0.023133
1688	20231116	21450	21700	21050	21350	4763	14.04	0.49	4.38	-0.009238	...	0.013000	0.014313	0.007303	-0.139198	-0.109461	-0.064424	-0.095835	-0.063059	-0.024368	-0.023664
1689	20231117	21350	21600	21050	21550	4324	14.04	0.49	4.38	0.000000	...	0.013008	0.014293	0.007308	-0.114477	0.024721	-0.067869	-0.072891	-0.057534	-0.026332	-0.023737
1690	20231120	21550	21550	21050	21050	5697	14.04	0.49	4.38	0.000000	...	0.013008	0.014287	0.007311	0.161664	0.276141	-0.007966	-0.070997	-0.030733	-0.024470	-0.021949
1691	20231121	21100	21500	21050	21500	2419	14.04	0.49	4.38	0.002375	...	0.013000	0.014287	0.007304	0.243902	0.082238	0.024431	-0.030718	-0.014688	-0.020599	-0.020322

1692 rows × 45 columns

○ 롯데에너지머티리얼즈

롯데에너지머티리얼즈

	date	open	high	low	close	volume	per	pbr	roe	open_lastclose_ratio	...	inst_ma20	inst_ma60	inst_ma120	foreign	foreign_diff	foreign_ma5	foreign_ma10	foreign_ma20	foreign_ma60	foreign_ma120
0	20170102	12924	13069	12391	12391	646882	24.95	4.16	18.06	-0.042950	...	0.058178	0.018947	0.013157	0.010704	0.021172	0.010075	-0.002214	-0.041203	-0.020303	-0.031069
1	20170103	12536	12778	12391	12585	327433	24.95	4.16	18.06	0.011702	...	0.051714	0.013384	0.010620	0.078737	0.068033	0.008215	0.017658	-0.050431	-0.016656	-0.030568
2	20170104	12633	12778	12198	12294	308171	24.95	4.16	18.06	0.003814	...	0.041269	0.009421	0.009753	0.027589	-0.051148	0.005284	0.021511	-0.053130	-0.014944	-0.029907
3	20170105	12391	12585	12198	12585	166312	24.95	4.16	18.06	0.007890	...	0.039048	0.008536	0.010836	0.139238	0.111650	0.049160	0.031716	-0.036256	-0.010848	-0.029340
4	20170106	12633	12682	12440	12585	107246	24.95	4.16	18.06	0.003814	...	0.031241	0.009820	0.012350	-0.040719	-0.179958	0.043110	0.030203	-0.038693	-0.012099	-0.030128
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1687	20231115	42950	43250	41400	41950	244158	20.65	1.30	6.42	0.032452	...	-0.035697	-0.016337	-0.019537	-0.097674	-0.242236	0.116483	0.128626	0.088991	0.003952	0.008742
1688	20231116	42000	42100	40900	41950	132767	20.65	1.30	6.42	0.001192	...	-0.045473	-0.019849	-0.021124	0.139681	0.237355	0.092588	0.127478	0.087273	0.008083	0.009126
1689	20231117	42100	42100	40900	41350	129501	20.65	1.30	6.42	0.003576	...	-0.050273	-0.018729	-0.022831	0.028903	-0.110778	0.085104	0.137749	0.086948	0.010785	0.008498
1690	20231120	41400	43800	41200	43050	180603	20.65	1.30	6.42	0.001209	...	-0.051148	-0.020369	-0.021896	0.174632	0.145728	0.078021	0.141157	0.083765	0.012742	0.009582
1691	20231121	43500	45450	42950	45000	296896	20.65	1.30	6.42	0.010453	...	-0.041446	-0.016328	-0.018151	0.075283	-0.099349	0.064165	0.128299	0.079444	0.017436	0.008706

1692 rows × 45 columns

○ 포스코퓨처엠

포스코퓨처엠

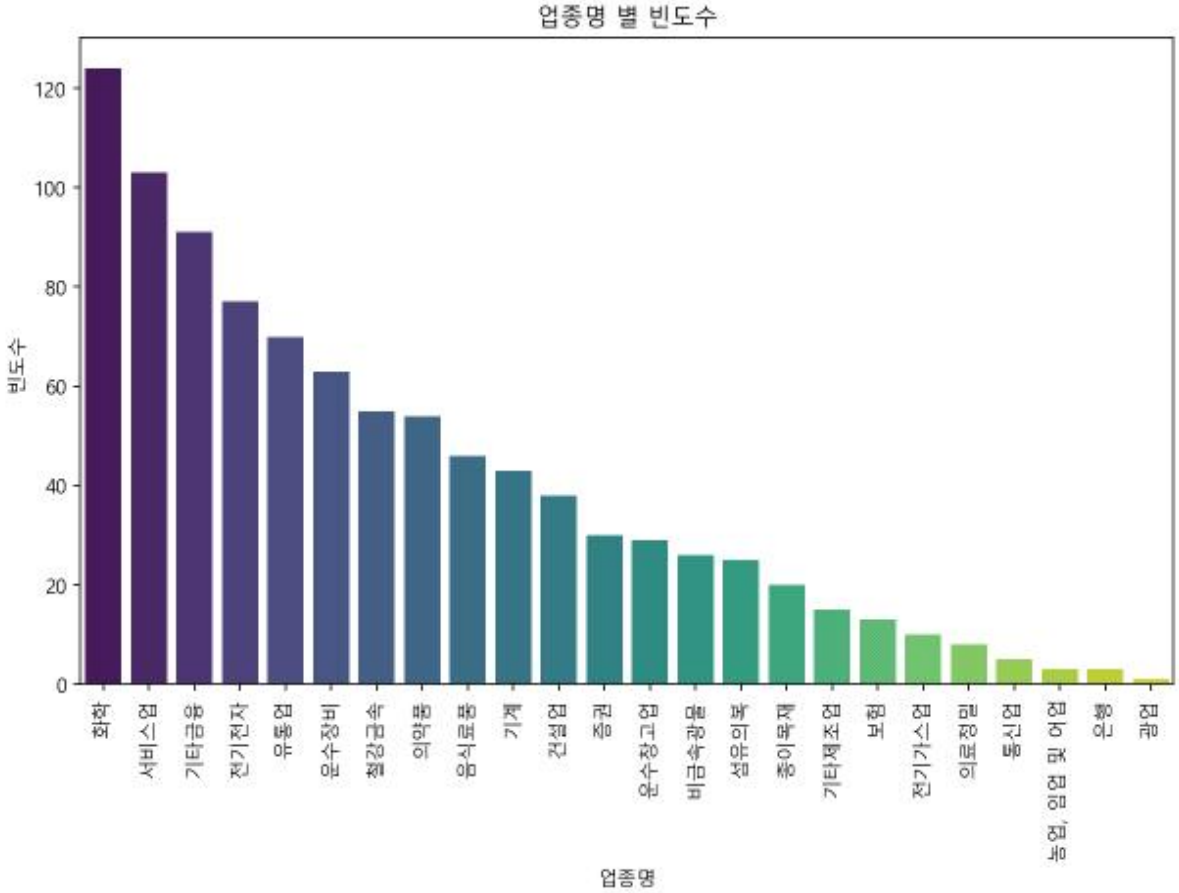
	date	open	high	low	close	volume	per	pbr	roe	open_lastclose_ratio	...	inst_ma20	inst_ma60	inst_ma120	foreign	foreign_diff	foreign_ma5	foreign_ma10	foreign_ma20	foreign_ma60	foreign_ma120
0	20170102	12050	12050	11700	12000	53734	53.2	4.76	9.32	0.004167	...	-0.083001	0.050975	0.037512	0.077772	-0.040962	0.139150	0.175896	0.143723	0.033026	0.027925
1	20170103	11950	11950	11650	11750	98419	53.2	4.76	9.32	-0.004167	...	-0.113076	0.041528	0.033451	0.134151	0.056379	0.162184	0.152667	0.149810	0.031897	0.028426
2	20170104	12100	12300	12000	12100	135147	53.2	4.76	9.32	0.029787	...	-0.117276	0.037624	0.032558	0.143015	0.008864	0.132059	0.149833	0.145940	0.033125	0.029733
3	20170105	12100	12150	11950	12100	88358	53.2	4.76	9.32	0.000000	...	-0.120928	0.037302	0.032398	0.003791	-0.139223	0.095493	0.132647	0.135351	0.036966	0.029280
4	20170106	12100	12300	11950	12300	96569	53.2	4.76	9.32	0.000000	...	-0.103586	0.038081	0.037046	0.027069	0.023277	0.077160	0.118305	0.121034	0.042258	0.028558
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1687	20231115	317500	320000	293000	297500	1120565	40.4	6.86	18.49	0.074450	...	-0.009190	-0.016696	-0.012652	-0.140738	-0.118711	-0.035333	-0.039107	-0.030956	-0.036361	-0.013098
1688	20231116	297000	303000	289000	293500	701953	40.4	6.86	18.49	-0.001681	...	-0.005952	-0.017263	-0.012798	-0.072961	0.067777	-0.035944	-0.051454	-0.030026	-0.036218	-0.014563
1689	20231117	288000	293000	283000	286000	552165	40.4	6.86	18.49	-0.018739	...	-0.005524	-0.017211	-0.013225	-0.053529	0.019431	-0.056122	-0.043781	-0.033261	-0.035829	-0.014398
1690	20231120	284500	305000	284000	300500	761493	40.4	6.86	18.49	-0.005245	...	-0.000719	-0.016503	-0.012428	0.025019	0.078549	-0.052847	-0.037600	-0.034459	-0.033010	-0.013142
1691	20231121	303000	306500	295500	303500	580088	40.4	6.86	18.49	0.008319	...	-0.001828	-0.017823	-0.012191	0.078692	0.053673	-0.032703	-0.019479	-0.021557	-0.029735	-0.012048

1692 rows × 45 columns

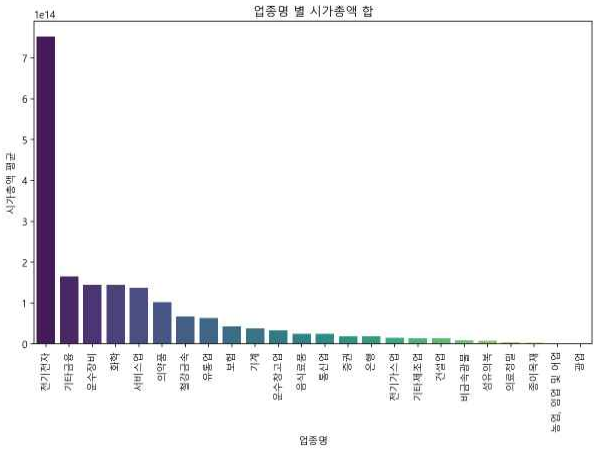
2. 탐색적 분석

가. EDA

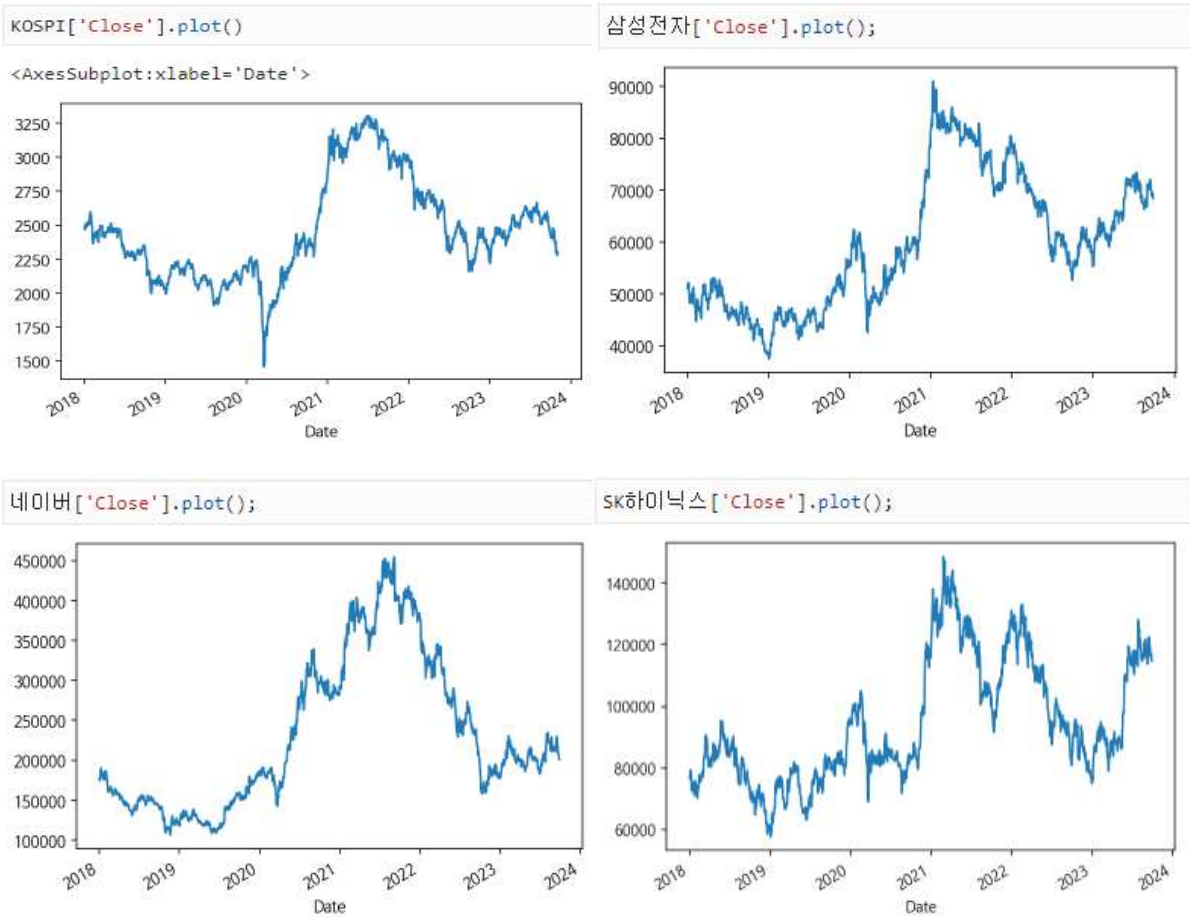
□ 주식 업종



○ 2000개가 넘는 주식중 KOSPI에 상장된 주식은 952개이다. 이 또한너무 많기에 범위를 좀 더 줄여줄 필요가 있으며 업종별 시가총액의 합과 평균을 확인해본 결과 전기전가 업종이압도적으로 높았기에 전기전자 업종을 선택하여 진행

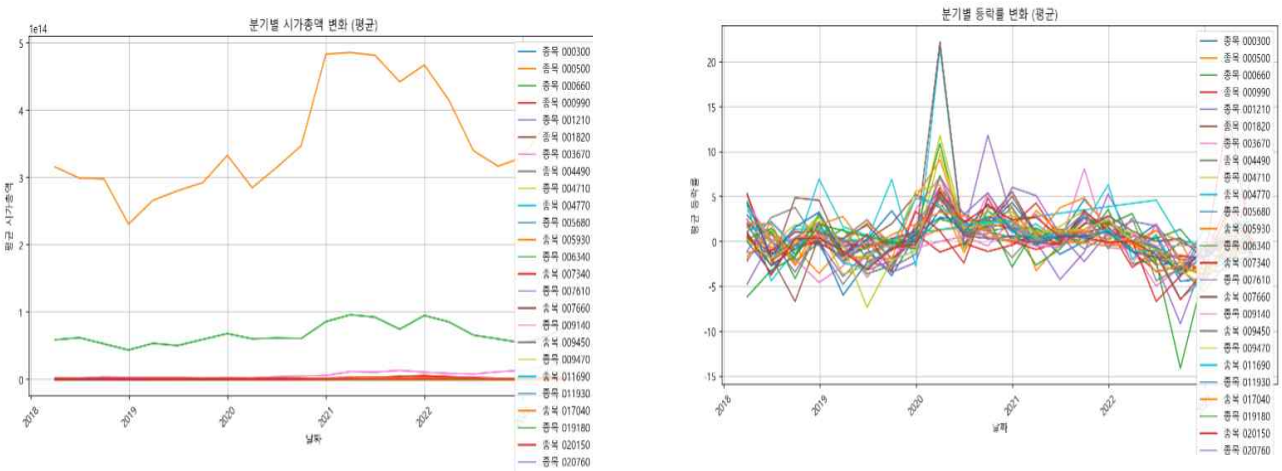


□ KOSPI지수와 기업주가지수



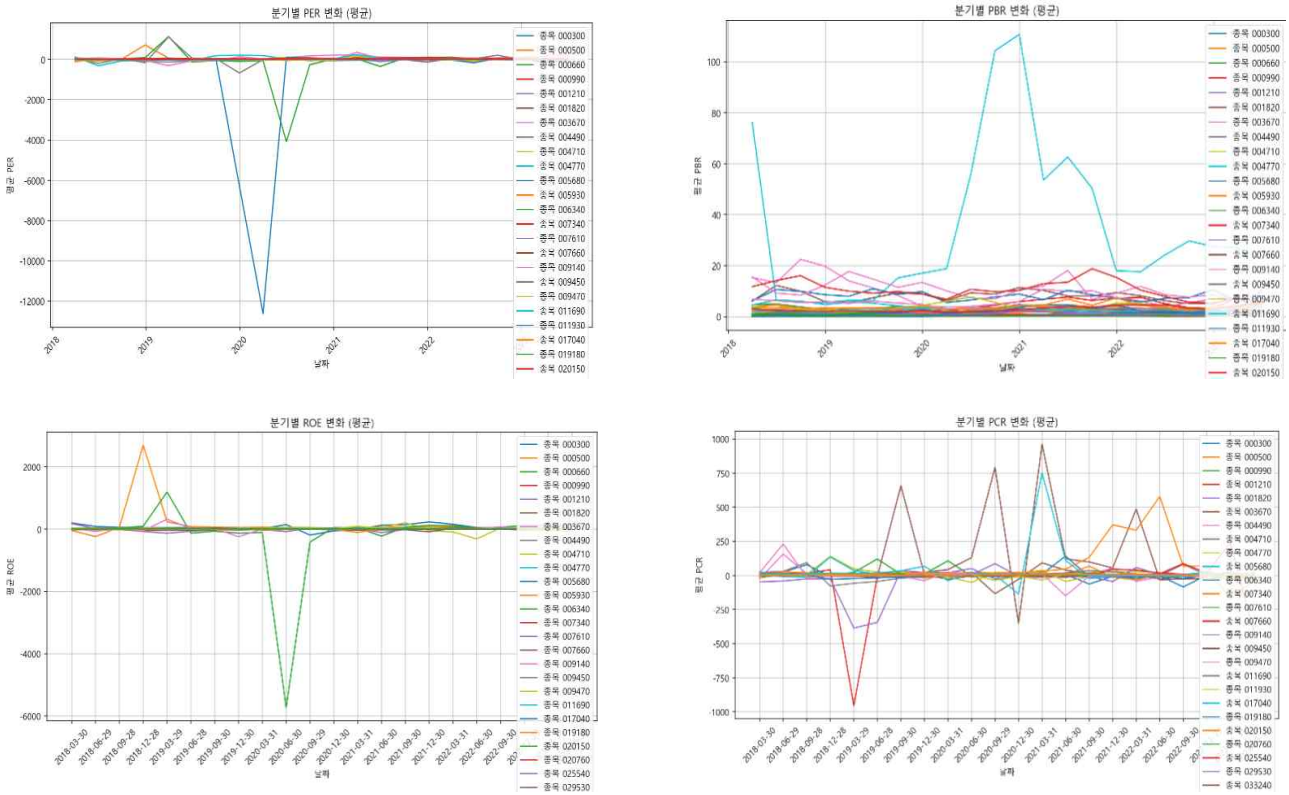
○ 기업주가지수는 KOSPI 지수와 비슷한 흐름을 가져간다는 경향이 있으며 이는 기업의 주가가 증권시장 전반의 움직임과 연관되어 있음을 나타낸다고 볼 수 있음

□ 시과총액과 등락률



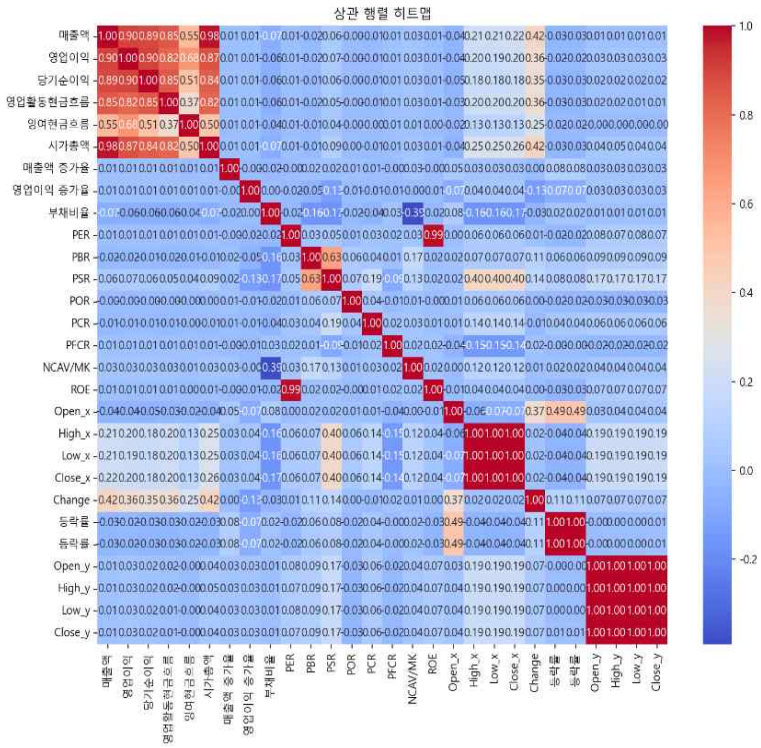
○ 시가총액이 높다고 해서 등락률이 높은 것은 아니며 등락률에는 더 많은 변수를 고려해야함

□ PER, PBR, ROE, PCR 등 재무제표 관련 변수 그래프의 한계



○ PER, PBR, ROE, PCR 등 재무제표 관련 변수의 그래프로만은 의미 있는 인사이트를 도출하기 어려운 한계가 있음

□ 상관관계 확인



- 매출액과 시가총액: 매출액과 시가총액 간에 매우 높은 양의 상관관계(약 0.98)가 있다. 이는 기업의 규모와 성과 간에 강력한 관련이 있을 수 있음을 나타냄
- 수익과 관련된 지표 간 상관관계: 영업이익, 당기순이익, 영업활동현금흐름, 잉여현금흐름은 서로 강한 상관관계를 가지고 있습니다. 이는 기업의 수익 구조가 서로 연관되어 있다는 것을 나타냄
- 부채비율과 수익성 지표 간 상관관계: 부채비율은 매출액, 영업이익, 당기순이익과 음의 상관관계를 가진다. 즉, 부채비율이 낮을수록 기업의 수익성이 높다는 것을 나타냄
- 시장 지표와 주가 지수 간 상관관계: PER, PBR, PSR 등의 시장 지표는 시가총액과 양의 상관관계를 가지고 있습니다. 이는 이러한 지표가 기업의 시장 가치에 영향을 미칠 수 있다는 것을 나타냄
- 등락률과 주가 지수 간 상관관계: 등락률은 기술적 지표와도 관련이 있으며, 특히 Change와 강한 양의 상관관계를 보인다. 이는 주식 가격의 변동이나 등락이 크다면, 주가 지수도 함께 크게 움직인다는 것을 의미한다. 투자자들은 등락률을 통해 특정 기간 동안의 주식 시장의 활동을 파악하고, 이를 기반으로 투자 전략을 수립

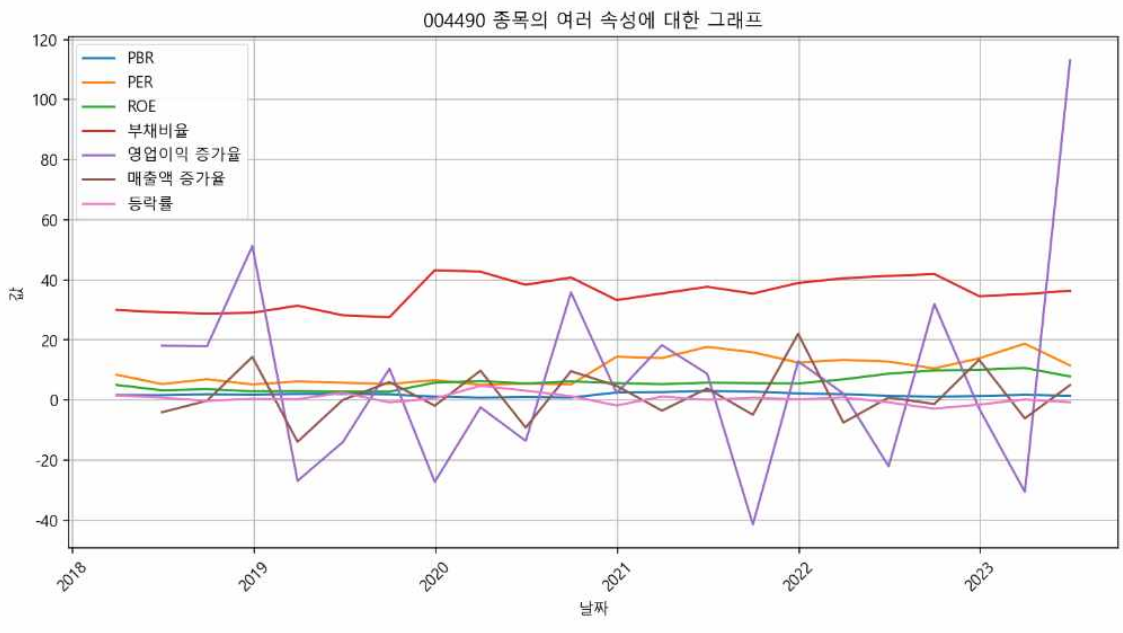
나. 종목 추천

□ 종목 선택을 위한 필터링

```
# 필터 조건 설정
filter_condition = (
    (merged_df_최근['시가총액'] >= 50000000000) &  # 시가총액 500억 이상
    (merged_df_최근['PBR'] <= 2) &  # PBR 2 이하
    (merged_df_최근['PER'] <= 12) &  # PER 12 이하
    (merged_df_최근['ROE'] >= 5) &  # ROE (+)10% 이상
    (merged_df_최근['매출액 증가율'] >= 5) &  # 매출액 증가율 (+)10% 이상
    (merged_df_최근['영업이익 증가율'] >= 5) &  # 영업이익 증가율 (+)10% 이상
    (merged_df_최근['부채비율'] <= 100)  # 부채비율 100% 이하
```

	Code	Close_x	등락률	PER	PBR	ROE	Close_y	매출액 증가율	영업이익 증가율	부채비율
175	004490	47350.0	-0.733753	11.501688	1.463222	7.860520	2564.280029	5.002913	113.123901	36.387621
549	025540	58700.0	-0.170068	10.159181	1.522263	6.673736	2564.280029	5.434708	75.698308	45.729403

- PBR이 작을수록 주당순자산 대비 주가가 작다는 뜻으로 저평가되었다고 볼 수 있다
- PER이 작을수록 주가 대비 순이익이 높다는 뜻으로 저평가되었다고 볼 수 있다
- 자기 자본으로 얼마만큼의 당기순이익을 만드는가로, 단순하게 이익이 많을수록 ROE가 커지기 때문에 클수록 좋다고 할 수 있다
- 004490, 025540둘다 괜찮은 상태의 종목이며, 영업이익 증가율이 더 높은 004490 선택



□ Min-Max 스케일러를 사용한 정규화

$$x_{new} = \frac{x - min(X)}{max(X) - min(X)}$$

○ 독립변수들의 값이 범위를 0~1 사이로 변경시켜 주는 피처 스케일링 방법이다

□ 콘텐츠 기반 필터링

○ 사용자에게 상품, 콘텐츠, 또는 아이템을 추천하는 데 활용하며, 이 방법은 아이템의 내용(콘텐츠)과 사용자의 선호도를 기반으로 추천을 수행하는 방식이다

○ 코사인 유사도 (Cosine Similarity)를 사용하여 아이템(주식) 간의 유사도를 계산한 후, 이를 기반으로 추천 목록을 생성하는 방식을 사용함

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

○ 협업 필터링과는 달리 콘텐츠 기반 필터링은 사용자의 이전 평가나 행동을 사용하지 않고, 아이템의 내용(특성)에만 의존한다. 따라서 사용자-아이템 평가 데이터나 사용자 행동 로그가 없어도 콘텐츠 기반 필터링을 사용할 수 있었음

```
# 사용자가 선택한 주식과 유사한 주식 추천
user_selected_stock = '004710'  # 사용자가 선택한 주식
num_recommendations = 4  # 004710 더불어 추천할 주식 수
recommendations = get_stock_recommendations(user_selected_stock, similarity_matrix, num_recommendations)
print(recommendations)
```

```
['011930', '011690', '009470', '009450']
```

004710: 한솔PNS
004770: 써니전자
003670: 포스코켐텍
009140: 삼성중공업
020150: 일진머티리얼즈

3. 모델링

가. 강화학습

□ 마르코프 가정

○ 상태가 연속적인 시간에 따라 이어질 때 어떠한 시점의 상태는 그 시점 바로 이전의 상태에만 영향을 받는다는 가정

○ 현재의 상태는 그 직전의 상태에 가장 큰 여향을 받으며 직전의 상태는 그 이전 상태들을 이미 반영

○ 마르코프 가정으로 어려운 문제들을 단순화할 수 있으며, 단순화를 하여도 만족스러운 결과를 얻는 경우가 많음

□ 마르코프 의사결정 과정 - MDP

○ 마르코프과정을 기반으로 한 의사결정 모델이며 구성요소는 다음과 같다

○ 상태 집합 (State Space) - 가능한 모든 상태의 집합
EX) 주식 거래에서의 상태는 현재 주가, 보유 중인 주식 수, 보유 중인 현금 등

○ 행동 집합 (Action Space) - 에이전트가 가능한 모든 행동의 집합

EX) 주식 거래에서의 행동은 매수, 매도, 홀딩 등

○ 상태 전이 확률 (State Transition Probability) - 현재 상태에서 어떤 행동을 취했을 때, 다음 상태로 전이할 확률

EX) 주식 거래에서 현재 주가와 보유 중인 주식 수에 대한 특정 행동이 취해졌을 때, 다음 시점의 주가와 주식 수의 변화

○ 보상 함수 (Reward Function) - 에이전트가 어떤 상태에서 어떤 행동을 취했을 때 받는 보상의 크기를 정의한 함수로 에이전트가 학습하고자 하는 목표를 반영하며, 최적의 의사결정을 돕는 역할을 함

EX) 주식 거래에서 매수로 인한 수익이나 매도로 인한 손실 등

○ 할인 요인 (Discount Factor) - 미래의 보상을 현재 가치로 할인하여 고려하는 요인으로 미래의 보상이 현재로부터 먼 미래에 받을 때, 그 가치를 낮춰주어 현재의 의사결정에 영향을 미침

EX) 할인 요인이 0.9일 경우, 1년 후의 보상은 현재 가치의 0.9배로 감소하여 반영

□ 강화학습

○ 강화학습은 MDP의 개념을 기반으로 하지만, 아래와 같은 몇가지 차이점이 있음

○ 무한한 상태 집합 : 일반적인 MDP에서는 상태 집합이 유한지만 강화학습에서는 상태가 무한할 수 있으며 예로 주식 거래에서 가능한 주가 패턴은 무한하다는 것

○ 신경망을 통한 상태 전이 확률 모델링 : MDP에서는 상태 전이 확률이 명시적으로 주어지만, 강화학습에서는 이러한 확률을 인공신경망(neural network)을 사용하여 모델링함. 특히, Deep Q-Network (DQN), Policy Gradient (PG), Asynchronous Advantage Actor-Critic (A3C) 등과 같은 알고리즘에서 신경망을 사용함

○ 신경망을 사용한 가치 함수와 정책 함수: 강화학습에서는 상태 가치 함수나 상태-행동 가치 함수를 인공신경망으로 근사화하여 학습하며 이를 통해 에이전트는 보상을 최대화하는 최적의 정책을 학습함

○ 강화학습은 더 복잡한 환경에서도 적용 가능하도록 인공신경망과 학습 알고리즘을 사용하여 MDP를 발전시킨 것

□ 강화학습 기법 종류

○ 몬테카를로 학습 (MC) : 샘플링을 통해 상태 가치 함수를 학습하며 충분한 에피소드를 거치면 상태 가치 함수가 정답에 근접해짐

○ 시간차 학습 (TD) : 시간차 예측을 기반으로 하며 On-policy(행동을 결정하는 정책과 학습할 정책이 같음) 시간차 제어인 SARSA가 이에 속함

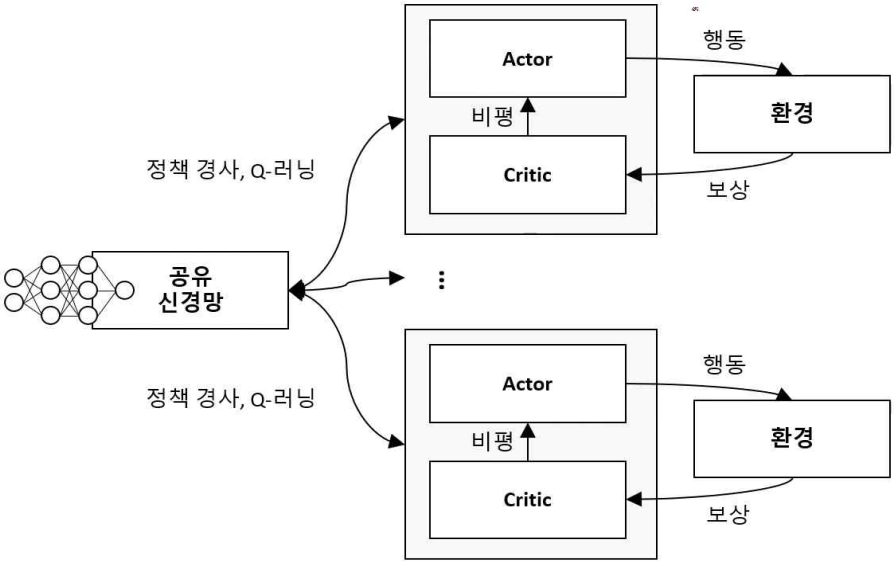
○ Q-learning : Off-policy(행동하는 정책과 학습하는 정책이 다름) 시간차 제어로, 다음 상태에서 가장 큰 Q값을 이용해 Q 함수를 갱신함. 상태-행동 가치가 가장 높은 행동을 선택함

○ DQN (Deep Q-Network) : 상태-행동 가치를 심층 신경망으로 모델링하며 복잡한 문제에서 상태 공간이 큰 경우 유용함. 회귀 문제를 푸는 강화학습에 적합함

○ 정책 경사 (Policy Gradient, PG) : 분류 문제를 푸는 강화학습으로, 상태에서 어떤 행

동을 취할지를 결정함. 신경망 모델을 사용하며, 손실 함수에 소프트맥스 등을 사용하여 분류함

- ○ 액터-크리틱 (Actor-Critic, AC) : DQN과 PG를 혼합한 모델로, 액터는 행동을 결정하고 크리틱은 비평하며 두 가지 강화학습 기법을 융합하여 사용함
- ○ A2C (Advantage Actor-Critic) : 어드밴티지를 적용한 AC 모델로, 상태-행동 가치에서 상태 가치를 뺀 값을 사용함. AC와 유사하지만 어드밴티지를 적용한 손실 함수를 사용함
- ○ A3C (Asynchronous Advantage Actor-Critic) : A2C에 비동기 개념을 추가한 방법으로, 여러 환경에서 동시에 행동을 수행하고 신경망을 학습함. 다양한 경험을 통해 빠르고 효과적으로 학습함



□ 강화학습에서의 딥러닝

- ○ 강화학습과 딥러닝은 상호보완적인 분야로, 딥러닝 기술은 강화학습의 성능을 향상시키고 확장하는 데 기여함. 딥러닝은 다양한 신경망 구조와 학습 알고리즘을 통해 강화학습의 핵심 요소들을 강화하고 확장하는 데 사용됨
- ○ 에이전트의 "두뇌"로서 딥러닝은 주식 데이터의 정보를 처리하여 매수, 매도, 홀딩과 같은 행동을 결정하는 데에 활용됨. 포트폴리오 가치(PV)를 최대화하기 위해 주식 잔고, 현재 주가, 현금 잔고 등 다양한 변수를 고려하여 효과적인 투자 전략을 학습함
- ○ 신경망 모델링

1. 심층 신경망 (DNN) : 대부분의 특징이 연속 값을 가지는 경우, DNN은 이들 특징의 무한한 조합을 효과적으로 학습할 수 있는 강력한 구조를 제공함. 다양한 계층과 많은 수의 퍼셉트론으로 구성되어 있어, 복잡한 비선형성과 추상적인 패턴을 학습하는 데 적합함

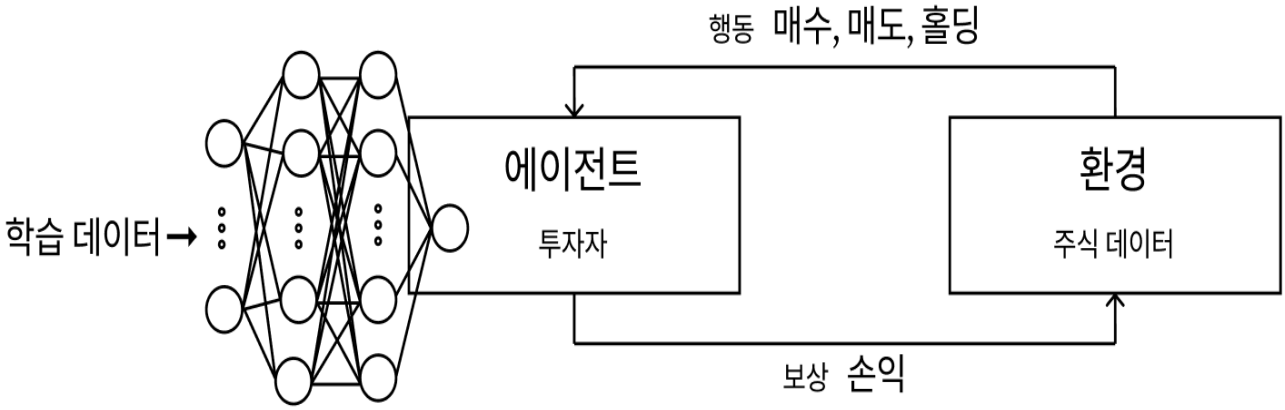
2. 합성곱 신경망 (CNN) : 주로 이미지 기반의 상태 정보를 다루는 데에 사용되며 특히, 특정 기간 동안의 상태 정보를 한꺼번에 고려할 수 있는 구조를 제공함. 예를 들어, 5일, 10일 등의 일련의 주가 데이터를 종합적으로 고려하여 패턴을 학습할 수 있으며 공유 가

중치와 피처 맵을 통해 이전 상태들의 흐름을 종합적으로 고려하므로, 높은 학습 품질을 기대할 수 있음

3. 순환 신경망 (LSTM) : 현재의 상태뿐만 아니라 이전에 에이전트가 결정한 행동의 영향을 함께 고려할 수 있는 구조. 예를 들어, [매수->매도->매수->매도]와 같은 변화가 많은 투자 패턴을 피하고 어느 정도의 투자 일관성을 유지할 수 있도록 도움을 줌

□ 주식에서의 강화학습

○ 주식 투자의 목표는 초기 자본금을 최대한 증가시키는 것이며, 손익을 최소화하는 것



○ 강화학습에서는 매 행동마다의 손익을 기준으로 긍정적인 보상과 부정적인 보상을 정의함

○ 상태 공간

시장 및 종목의 상태 : 코스피, 코스닥 지수 및 기초 자산의 상태, 각 종목의 차트와 같은 주가 데이터 등을 고려

에이전트(투자자)의 상태 : 초기 자본금, 보유 종목 수, 현금 잔고, 평가 금액, 손익 금액 및 손익률 등을 고려

○ 행동 공간

주식 투자에서 가능한 행동들을 정의함. 예로 매수, 매도, 관망과 같은 행동들이 가능하며, 어떤 비율로 주식을 매수할지 등의 세부적인 행동도 정의할 수 있음

○ 행동 결정

정책 신경망 : 투자에 대한 의사 결정은 주로 정책 신경망을 활용하여 수행. 이 신경망은 학습 데이터와 발생한 보상을 기반으로 갱신.

규칙 적용 : 추가로 특정 규칙을 적용하여 투자 전략을 더 세밀하게 조절함

○ 무작위 행동 결정

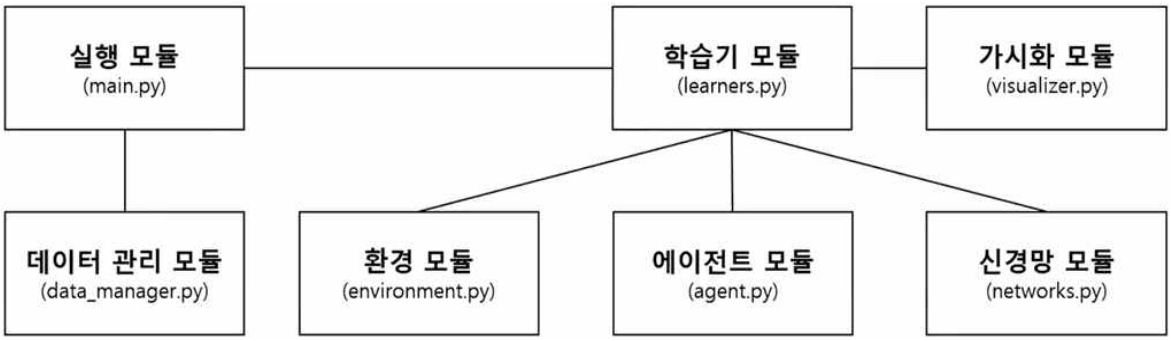
무작위 행동 : 학습 초기에는 무작위로 행동을 결정하여 다양한 경험을 쌓도록 함

무작위 행동 결정 비율(엡실론) : 무작위로 행동을 결정하는 비율을 조절하여 학습 초기에는 높게 설정하고, 학습이 진행됨에 따라 줄여나감

○ 주식 투자에서의 강화학습은 동적이고 불확실한 환경에서 효과적으로 학습하며, 지속적인 수정과 최적화를 통해 최적의 투자 전략을 찾음

나. 강화학습 설계

□ 구조



□ 실행 모듈

○ 강화학습의 변수를 선택하고 조절할 수 있는 옵션을 제공

```python
if __name__ == '__main__':
    parser = argparse.ArgumentParser()
    parser.add_argument('--mode', choices=['train', 'test', 'update', 'predict'], default='train')
    parser.add_argument('--ver', choices=['v1', 'v2', 'v3', 'v4', 'v4.1', 'v4.2'], default='v4.1')
    parser.add_argument('--name', default=utils.get_time_str())
    parser.add_argument('--stock_code', nargs='+')
    parser.add_argument('--rl_method', choices=['dqn', 'pg', 'ac', 'a2c', 'a3c', 'monkey'], default='a2c') #monkey - 무작위
    parser.add_argument('--net', choices=['dnn', 'lstm', 'cnn', 'monkey'], default='dnn') # monkey - 무작위는 신경망 생성 X
    parser.add_argument('--backend', choices=['pytorch', 'tensorflow', 'plaidml'], default='pytorch')
    parser.add_argument('--start_date', default='20200101')
    parser.add_argument('--end_date', default='20201231')
    parser.add_argument('--lr', type=float, default=0.001)
    parser.add_argument('--discount_factor', type=float, default=0.9) # 할인 요인
    parser.add_argument('--balance', type=int, default=100000000)
    args = parser.parse_args()

    # 학습기 파라미터 설정
    output_name = f'{args.mode}_{args.name}_{args.rl_method}_{args.net}'
    learning = args.mode in ['train', 'update']
    reuse_models = args.mode in ['test', 'update', 'predict']
    value_network_name = f'{args.name}_{args.rl_method}_{args.net}_value.mdl'
    policy_network_name = f'{args.name}_{args.rl_method}_{args.net}_policy.mdl'
    start_epsilon = 1 if args.mode in ['train', 'update'] else 0
    num_epoches = 1000 if args.mode in ['train', 'update'] else 1 # 에포크 1000으로 설정
    num_steps = 5 if args.net in ['lstm', 'cnn'] else 1
```

□ 데이터 관리 모듈

○ 강화학습을 위한 차트 데이터를 불러오며 데이터 전처리 함수 포함

```python
def load_data_v3_v4(code, date_from, date_to, ver):
    columns = None
    if ver == 'v3':
        columns = COLUMNS_TRAINING_DATA_V3
    elif ver == 'v4':
        columns = COLUMNS_TRAINING_DATA_V4

    # 시장 데이터
    df_marketfeatures = pd.read_csv(
        os.path.join(settings.BASE_DIR,'RLTRADER', 'data', ver, 'marketfeatures.csv'),
        thousands=',', header=0, converters={'date': lambda x: str(x)})

    # 종목 데이터
    df_stockfeatures = None
    for filename in os.listdir(os.path.join(settings.BASE_DIR,'RLTRADER', 'data', ver)):
        if filename.startswith(code):
            df_stockfeatures = pd.read_csv(
                os.path.join(settings.BASE_DIR,'RLTRADER', 'data', ver, filename),
                thousands=',', header=0, converters={'date': lambda x: str(x)})
            break

    # 시장 데이터와 종목 데이터 합치기
    df = pd.merge(df_stockfeatures, df_marketfeatures, on='date', how='left', suffixes=('', '_dup'))
    df = df.drop(df.filter(regex='_dup$').columns.tolist(), axis=1)

    # 날짜 오름차순 정렬
    df = df.sort_values(by='date').reset_index(drop=True)

    # NaN 처리
    df = df.fillna(method='ffill').fillna(method='bfill').reset_index(drop=True)
    df = df.fillna(0)
```

□ 학습기 모듈

○ DQN, PC, A2C, A3C 등의 강화학습 알고리즘을 구현하고 실행

```
class ReinforcementLearner:
    __metaclass__ = abc.ABCMeta
    lock = threading.Lock() # a3C 때문에설정
    reward_instance = Reward()
    def __init__(self, rl_method='rl', stock_code=None,
                 chart_data=None, training_data=None,
                 min_trading_price=100000, #주식 최소 거래 가격 (10만원)
                 max_trading_price=10000000, # 주식 최대 거래 가격 (1000만원)
                 net='dnn', num_steps=1,
                 lr=0.0005, #학습률 (강화 학습 모델이 가중치를 업데이트)
                 discount_factor=0.9, # 할인 계수로 (미래 보상의 가치를 줄이는 데 사용됩니다)
                 num_epoches=1000, #학습횟수
                 balance=100000000, # 초기 자본금 (1억)
                 start_epsilon=1, # 입실론의 초기 값으로(0과 1 사이의 확률 값), 탐험(exploration)과 활용(exploitation) 사이에 균형이 중요, 처음에는 1로 시작해서 줄여나감
                 value_network=None, policy_network=None,
                 output_path='', reuse_models=True, gen_output=True):
        # 인자 확인
        assert min_trading_price > 0
        assert max_trading_price > 0
        assert max_trading_price >= min_trading_price
        assert num_steps > 0
        assert lr > 0
        # 강화학습 설정
        self.rl_method = rl_method
        self.discount_factor = discount_factor
        self.num_epoches = num_epoches
        self.start_epsilon = start_epsilon
        # 환경 설정
        self.stock_code = stock_code
```

□ 환경모듈

○ 에이전트가 투자할 종목의 차트 데이터를 관리함

○ 과거 시점부터 최근 시점까지의 데이터를 순차적으로 제공하며, 미래의 차트 데이터는 알 수 없음

```
# 차트데이터 환경 모듈
class Environment:
    PRICE_IDX = 4  # 종가의 위치 date, o, h, l, c 고정 (차트 바꿀시 변경)

    def __init__(self, chart_data=None):
        self.chart_data = chart_data
        self.observation = None
        self.idx = -1

    def reset(self):
        self.observation = None
        self.idx = -1

    def observe(self):
        if len(self.chart_data) > self.idx + 1:
            self.idx += 1
            self.observation = self.chart_data.iloc[self.idx]
            return self.observation
        return None

    def get_price(self):
        if self.observation is not None:
            return self.observation[self.PRICE_IDX]
        return None
```

□ 에이전트 모듈

○ 주식을 매수하거나 매도하는 투자자 역할을 수행함

○ 초기 자본금, 현금 잔고, 주식 잔고 상태를 관리하며 포트폴리오 가치(PV)는 현금 잔고와 주식 잔고의 평가액을 합한 값

○ 투자 목표는 PV를 높이는 것이며, 매수 시 주식 잔고는 늘어나고 현금 잔고는 감소하며, 매도 시 주식 잔고는 감소하고 현금 잔고는 늘어남

```python
def __init__(self, environment, initial_balance, min_trading_price, max_trading_price):
    # 현재 주식 가격을 가져오기 위해 환경 참조
    self.environment = environment
    self.initial_balance = initial_balance  # 초기 자본금

    # 최소 단일 매매 금액, 최대 단일 매매 금액
    self.min_trading_price = min_trading_price
    self.max_trading_price = max_trading_price

    # Agent 클래스의 속성
    self.balance = initial_balance  # 현재 현금 잔고
    self.num_stocks = 0  # 보유 주식 수
    # 포트폴리오 가치: balance + num_stocks * {현재 주식 가격}
    self.portfolio_value = 0
    self.num_buy = 0  # 매수 횟수
    self.num_sell = 0  # 매도 횟수
    self.num_hold = 0  # 관망 횟수

    # Agent 클래스의 상태 - (비율로 (0~1, -1~1 등)넣어야함, 변수로 사용하기 때문에)
    self.ratio_hold = 0  # 주식 보유 비율
    self.profitloss = 0  # 손익률
    self.avg_buy_price = 0  # 주당 매수 단가
```

□ 신경망 모듈

- ○ DNN, CNN, LSTM 신경망을 구현하여 주식 데이터가 주어졌을 때 매수, 매도, 홀딩을 결정함
- ○ 신경망 종류에 따라 데이터 학습 방법이 달라지므로 매수와 매도 행위 결정은 다를 수 있음
- ○ 학습 목표는 매수, 매도, 홀딩 행위에 대해 향후 PV를 최대한 높이는 것

```python
class LSTMNetwork(Network):
    def __init__(self, *args, num_steps=1, **kwargs):
        self.num_steps = num_steps
        super().__init__(*args, **kwargs)

    @staticmethod
    def get_network_head(inp, output_dim):
        return torch.nn.Sequential(
            torch.nn.BatchNorm1d(inp[0]),
            LSTMModule(inp[1], 128, batch_first=True, use_last_only=True),
            torch.nn.BatchNorm1d(128),
            torch.nn.Dropout(p=0.1),
            torch.nn.Linear(128, 64),
            torch.nn.BatchNorm1d(64),
            torch.nn.Dropout(p=0.1),
            torch.nn.Linear(64, 32),
            torch.nn.BatchNorm1d(32),
            torch.nn.Dropout(p=0.1),
            torch.nn.Linear(32, output_dim),
        )

    def train_on_batch(self, x, y):
        x = np.array(x).reshape((-1, self.num_steps, self.input_dim))
        return super().train_on_batch(x, y)

    def predict(self, sample):
        sample = np.array(sample).reshape((-1, self.num_steps, self.input_dim))
        return super().predict(sample)
```

□ 가시화 모듈

- ○ 에이전트의 상태(보유 주식 수), 가치 그래프, 정책 그래프, 포트폴리오 가치(PV) 등을 시각적으로 나타냄
- ○ 에포크 진행에 따라 정책 신경망이 결정하는 행동 확률이 포트폴리오 가치를 높이는 방향으로 변화함. 이에 따라 포트폴리오 가치가 변하는 것을 확인할 수 있음
- ○ 과소적합, 과대적합 등 학습을 진행함에 따라 이상이 있는지 확인하여 학습이 효과적으로 이루어지고 있는지를 판단함

```python
def plot(self, epoch_str=None, num_epoches=None, epsilon=None,
        action_list=None, actions=None, num_stocks=None,
        outvals_value=[], outvals_policy=[], exps=None,
        initial_balance=None, pvs=None, cumulative_rewards=None):
    with lock:
        actions = np.array(actions)  # 에이전트의 행동 배열
        # 가치 신경망의 출력 배열
        outvals_value = np.array(outvals_value)
        # 정책 신경망의 출력 배열
        outvals_policy = np.array(outvals_policy)
        # 초기 자본금 배열
        pvs_base = np.zeros(len(actions)) + initial_balance

        # 차트 2. 에이전트 상태 (행동, 보유 주식 수)
        for action, color in zip(action_list, self.COLORS):
            for i in self.x[actions == action]:
                # 배경 색으로 행동 표시
                self.axes[1].axvline(i, color=color, alpha=0.1)
        self.axes[1].plot(self.x, num_stocks, '-k')  # 보유 주식 수 그리기

        # 차트 3. 가치 신경망
        if len(outvals_value) > 0:
            max_actions = np.argmax(outvals_value, axis=1)
            for action, color in zip(action_list, self.COLORS):
                # 배경 그리기
                for idx in self.x:
                    if max_actions[idx] == action:
                        self.axes[2].axvline(idx, color=color, alpha=0.1)
                # 가치 신경망 출력 그리기
                self.axes[2].plot(self.x, outvals_value[:, action],
                    color=color, linestyle='-')
```

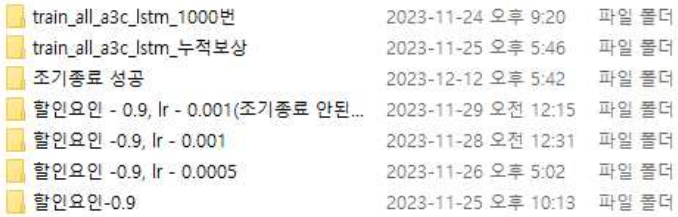
다. 학습 과정

□ 학습 과정

- ○ 강화학습 모델 : DQN, PG, A3C와 같은 다양한 강화학습 모델 중에서 가장 성능이 우수한 모델을 선택
- ○ 신경망 모듈 : DNN, CNN, LSTM 신경망 구조를 사용하여 성능을 비교하고 선택
- ○ 최적화 함수 선택 : RMSprop, NAdam 중 가장 효과적인 함수를 선택
- ○ 하이퍼파라미터 설정 :

학습률(learning rate)을 0.0005에서 0.001까지 조절하면서 안정적인 수렴을 확인합니다.
탐험률(exploration rate)은 입실론 값을 사용하며, 초기 값은 1로 설정하여 초기에는 탐험 위주로 학습을 진행하고, 학습이 진행될수록 탐험 비율을 감소시킵니다.
discount factor는 0과 1 사이의 값으로, 작을수록 미래의 보상을 덜 중요시합니다. 0.7부터 0.9까지의 값을 확인하여 적절한 값을 선택합니다.

- ○ 학습 진행 : 선택한 강화학습 모델과 신경망 구조에 대해 학습을 진행하며 학습 과정을 지속적으로 모니터링하며 모델의 성능 체크, 하이퍼파라미터 및 구조의 조정이 필요한 경우 체크
- ○ 결과 평가 : 학습이 완료된 모델을 테스트 데이터에 적용하여 성능을 평가함. 에이전트의 행동이 목표인 포트폴리오 가치를 최대화하는지 확인하고 모델의 효과를 평가함
- ○ 최적화 : 성능이나 학습 속도 등을 고려하여 하이퍼파라미터와 모델 구조를 조정하고 추가적인 실험을 통해 최적화 진행
- ○ 반복 : 위 단계를 반복하여 최적의 학습 과정을 찾고 모델을 개선함



train_all_a3c_lstm_1000번	2023-11-24 오후 9:20	파일 폴더
train_all_a3c_lstm_누적보상	2023-11-25 오후 5:46	파일 폴더
조기종료 성공	2023-12-12 오후 5:42	파일 폴더
할인요인 - 0.9, lr - 0.001(조기종료 안된...	2023-11-29 오전 12:15	파일 폴더
할인요인 -0.9, lr - 0.001	2023-11-28 오전 12:31	파일 폴더
할인요인 -0.9, lr - 0.0005	2023-11-26 오후 5:02	파일 폴더
할인요인-0.9	2023-11-25 오후 10:13	파일 폴더

□ 학습 과정 모니터링

□ 모델 선정

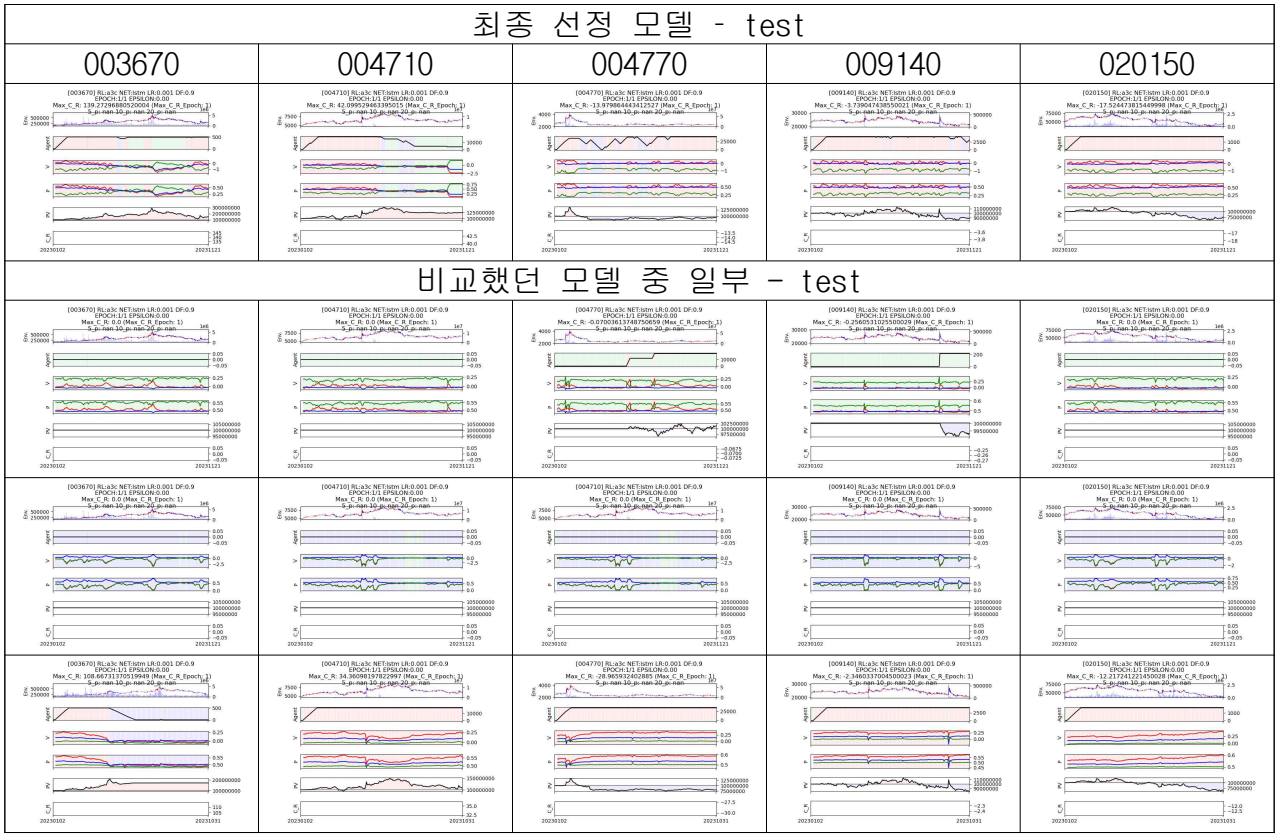
- ○ 누적보상이 높은 모델을 좋은 모델이라는 기준으로 학습된 모델을 테스트한 결과, 주식 5개에 대한 포트폴리오 수익이 가장 높은 모델을 선정
- ○ 종목 5개의 수익을 종합하여 가장 뛰어난 성과를 보인 모델을 최종적으로 선택

□ 최종 모델 선정

- ○ 강화학습 모델 : A3C
- ○ 신경망 : LSTM
- ○ 최적화 함수 : RMSprop
- ○ 학습률 : 0.001
- ○ 에포크수 : 1000

[과적합 방지 : 누적 보상 비율이 상위 5% 평균보다 5번 이상 높을 경우 조기 종료]

- ○ 탐험률 : 100%에서 10%씩 감소
- ○ 할인 비율 : 0.9



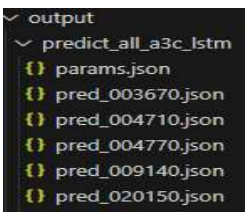
4. 모델평가 및 검증

□ 모델 평가

- ○ 테스트 결과, 각 주식별로 초기 자본금 100,000,000원에서 시작하여 다음과 같은 포트폴리오 가치를 얻음
- ○ 003670 : 200,000,000
- ○ 004710 : 125,000,000
- ○ 004770 : 0
- ○ 009140 : 90,000,000
- ○ 020150 : 75,000,000
- ○ 총 90,000,000의 수익을 냄

□ 매수, 매도, 홀딩 타이밍 추천

- ○ 2023년 11월에 대한 매수, 매도, 홀딩 추천과 확률 결과를 확인함
- ○ 학습된 모델로부터 얻은 추천 결과는 JSON 파일로 저장됨



○ 첫 번째 리스트는 에이전트가 매수, 매도, 홀딩에서 취할 수 있는 행동에 대한 기대되는 미래 보상, 즉 가치 값이며 값이 클수록 해당 행동의 가치가 높음

○ 두 번째 리스트는 에이전트가 매수, 매도, 홀딩에 대해 각 행동을 선택할 정책으로 그 확률을 보여줌

○ 003670

```
[["20231107", [-0.055992256850004196, -0.029559601098299026, 0.005038700997829437], [0.4858199954032898, 0.4925579130649566 7, 0.5013256072998047]],
["20231108", [0.022160910069942474, 0.010195571929216385, -0.015583120286464691], [0.5055067539215088, 0.5025491714477539, 0.4960661232471466]],
["20231109", [0.04689495265483856, 0.02531236782670021, -0.035760797560214996], [0.5117064714431763, 0.506303608417511, 0.4910205006599426]],
["20231110", [0.06306716799736023, 0.03460973873734474, -0.051496557891368866], [0.5157368779182434, 0.5085986852645874, 0.4870681464672088 6]],
["20231113", [0.03896792232990265, 0.025832820683717728, -0.07333993911743164], [0.5096511840820312, 0.5063653588294983, 0.4816080033779144 3]],
["20231114", [-0.01915295422077179, 0.0025225020945072174, -0.08334670215845108], [0.4950670301914215, 0.5005506277084351, 0.4790713191032409 7]],
["20231115", [-0.02236158400774002, 0.0025405213236808777, -0.07842397689819336], [0.4942745268344879, 0.5005776882171631, 0.4802555441856384]],
["20231116", [-0.00900818407535553, 0.0046699680387973785, -0.07746325433254242], [0.4976046979427337 6, 0.5010918378829956, 0.4805051982402801 5]],
["20231117", [0.010073177516460419, 0.01146121695637703, -0.08021395653486252], [0.5023801922798157, 0.5027689337730408, 0.4798451066017151]],
```

○ 004710

```
[["20231107", [-0.10515448451042175, -0.03733513504266739, -0.0703054890036583], [0.4734981060028076, 0.4906321167945862, 0.482211172580719]],
["20231108", [-0.07884889841079712, -0.03504997491836548, -0.07128456234931946], [0.48003897070884705, 0.4911497533321380 6, 0.4820375740528106 7]],
["20231109", [-0.08680638670921326, -0.03431629762053 49, -0.08672884106636047], [0.4780078232288360 6, 0.4912939369678497 3, 0.4781842231750488 3]],
["20231110", [-0.04716153815388679 5, -0.02414191514253616 3, -0.07474635541439056], [0.48800063133239746, 0.4938730597496032 7, 0.4812212288379669]],
["20231113", [-0.09203274548053741, -0.03075415268540382 4, -0.0844898372888565], [0.4767546653747558 6, 0.4922417104244232, 0.4786970019340515]],
["20231114", [-0.08075430989265442, -0.03116037696599 9603, -0.06091279909014702], [0.47966280579566956, 0.4922209084033966, 0.4845675230026245]],
["20231115", [-0.06872499734163284, -0.02063750848174095, -0.05631449818611145], [0.4826984703540802, 0.4948752224445343, 0.485681205987930 3]],
["20231116", [-0.06102824956178665, -0.02101237699389457 7, -0.0662834569811821], [0.48457714915275574, 0.4947267174720764, 0.483225494623184 2]],
["20231117", [-0.05720842257142067, -0.02025152742862701 4, -0.0678727924823761], [0.48551827669143677, 0.4948968291282654, 0.482853263616561 9]],
```

○ 004770

```
[["20231107", [-0.03092515468597412, -0.005215562880039215, -0.05719825625419617], [0.49222010374069214, 0.4987456202507019, 0.4854770302772522]],
["20231108", [0.016851887106895447, 0.0108569897711277, -0.03890055418014526 4], [0.504288911819458, 0.5028182864189148, 0.4900813102722168]],
["20231109", [0.011030644178390503, 0.013961326330900192, -0.05301558971405029], [0.5027961730957031, 0.5035675168037415, 0.486531615257263 2]],
["20231110", [-0.006867691874504089, 0.006763976067304611, -0.05315143615007400 5], [0.4982803165912628, 0.501750648021698, 0.486508667469024 66]],
["20231113", [-0.0025739073753356934, 0.009796049445867538, -0.04182411730289459], [0.49936601519584656, 0.5025231242179871, 0.489333212375640 87]],
["20231114", [0.006901867687702179, 0.007181864231824875, -0.06858133524656296], [0.5016884207725525, 0.5018035769462585, 0.482681244611740 1]],
["20231115", [-0.04067355766892433, -0.008104942739009857, -0.09056530892848969], [0.4896715879440307 6, 0.4979121088981628 4, 0.477192997932434 1]],
["20231116", [-0.03859798927394485, -0.01433564350008964 5, -0.067983232438564 3], [0.49025288224220276, 0.4964072704315185 5, 0.482853502035141]],
["20231117", [-0.0358126200735569, -0.006295118480920792, -0.0727766677737236], [0.49094462394714355, 0.4984205961227417, 0.481605023145675 66]],
```

○ 009140

```
[["20231107", [-0.15871980786323547, -0.08137483149766922, -0.08386670053005219], [0.4598674178123474, 0.4794005155563354 5, 0.4790123403072357]],
["20231108", [-0.14808650314807892, -0.06446200609207153, -0.12499640882015228], [0.4625369906425476, 0.4836223721504211 4, 0.468688428401947]],
["20231109", [-0.1588975191116333, -0.06846404075622559, -0.12517857551574707], [0.459838330745697, 0.4826269745826721, 0.4686232805252075]],
["20231110", [-0.1219482272863388, -0.06151529774069786, -0.1014382392168045], [0.46917590498924255, 0.4844596087932586 7, 0.4745522141456604]],
["20231113", [-0.1412358283996582, -0.05763990432024002, -0.10942432284355164], [0.46432217955589294, 0.4854103922843933, 0.472498089075088 5]],
["20231114", [-0.1276833862066269, -0.05913111940026283, -0.10544633865356445], [0.46767979860305786, 0.485007584095012, 0.473549216985702 5]],
["20231115", [-0.12190023064613342, -0.05464678087123184, -0.0822722315788269], [0.46923941373825073, 0.486244142055511 5, 0.479274362325668 33]],
["20231116", [-0.09967581182718277, -0.03776013106107712, -0.06487404555082321], [0.4748623073101043 7, 0.4905329644680023, 0.483560651540756 2]],
["20231117", [-0.10156077891588211, -0.03959131613337564, -0.06309057027101517], [0.4743785262107849, 0.49007168412208557, 0.484000533819198 6]],
```

○ 020150

```
[["20231107", [-0.08021457493305206, -0.02854570560157299, -0.05621209740638733], [0.4797733128070831 3, 0.492850661277771, 0.485759496688842 8]],
["20231108", [-0.03092847764492035, -0.014368195086717606, -0.06622830033302307], [0.4920781552791595 5, 0.4963168203830719, 0.483353286981582 64]],
["20231109", [-0.01287986338138580 3, -0.005432870239019394, -0.0748303011059761], [0.49660658836364746, 0.49853515625, 0.481233209371566 8]],
["20231110", [-0.00970455259084701 5, -0.0005145706236362457, -0.08365631103515625], [0.4973801672458648 7, 0.4997389614582062, 0.479033857583999 63]],
["20231113", [-0.04617936164140701, -0.01776915043592453, -0.09883539378643036], [0.4881555736064911, 0.4953683912754059, 0.475247561931610 1]],
["20231114", [-0.06106817722320556 6, -0.02526928856968879 7, -0.10764573514461517], [0.4844413101673126, 0.4935092031955719, 0.473048895597457 9]],
["20231115", [-0.07363861799240112, -0.02564158290624618 5, -0.09260565042495728], [0.4813638329505920 4, 0.4934965372085571 3, 0.476724803447723 4]],
["20231116", [-0.07367254793643951, -0.02835154719650745 4, -0.08793295174837112], [0.481319189071655 3, 0.492796927690506, 0.477881729602813 7]],
["20231117", [-0.05454176291823387, -0.02236214280128479, -0.09015848487615585], [0.48610037565231323, 0.4942696988582611, 0.477369397878646 85]],
```

V 평가 및 전개

1. 모델 발전계획 수립

□ 하이퍼파라미터 조정 및 모델 최적화

- ○ 하이퍼파라미터 조정 : 현재 사용 중인 강화학습 모델의 하이퍼파라미터를 검토하고, 성능 향상을 위해 조정하며 최적화 알고리즘을 변경하거나 다양한 최적화 알고리즘을 실험하여 모델 성능을 향상
- ○ 모델 아키텍처 변경 : 현재 사용 중인 신경망 아키텍처(DNN, CNN, LSTM 등)를 검토하고, 모델의 복잡성을 조절하며 추가적인 레이어를 추가하거나 제거하여 모델의 표현 능력을 향상시키거나 과적합을 방지
- ○ 다양한 아키텍처를 실험하여 어떤 구조가 주식 시장 데이터에 더 적합한지 확인

□ 데이터 추가

- ○ 외부 변수 추가 : 기존에 사용 중인 주식 데이터 외에도 다양한 외부 변수를 추가. 이는 경제 지표, 기업 실적, 시장 동향 등을 포함할 수 있으며 외부 변수의 다양성은 모델의 학습을 더욱 일반화시킬 수 있도록 도움을 줄 수 있음
- ○ 뉴스 및 이슈 데이터 활용 : 뉴스와 같은 이벤트 데이터를 모델에 통합하여 주식 시장의 감성 및 이벤트에 민감한 반응을 모델에 반영합니다.
- ○ 감정 분석을 통해 뉴스의 긍정적 또는 부정적인 영향을 모델이 학습할 수 있도록 데이터를 구성합니다.

2. 프로젝트 평가 및 향후 계획

가. 성과 및 한계

- ○ 2018년부터 2023년까지의 데이터를 활용하여 강화학습을 진행하였고, 그 결과 차트 데이터에 따른 주식의 매도, 매수, 홀딩 전략을 추천받을 수 있음
- ○ A3C 모델은 특정 종목에서 양호한 성과를 보이지만, 다른 종목에서는 과적합 현상이 발생한 케이스도 있었으며, 이를 극복하기 위해 하이퍼파라미터 조정과 모델의 일반화 능력을 향상시키는 노력을 하였으며 추가적인 테스트가 필요함
- ○ 주식 시장은 이슈에 민감하게 반응하는 특성이 있어, 이를 고려하기 위해기사를 수집하여 감정분석을 진행함, 이 과정에서 아직 모델 학습에 녹여내지 못한 부분이 아쉬움

나. 향후 계획

- ○ 거래 수수료와 거래세를 고려하여 모델을 학습
- ○ 데이터의 다양성을 확보하기 위해 기사 데이터와 더불어 다양한 변수들을 추가할 계획
- ○ 향후 최적의 주식 투자전략을 위해 만들어진 이 모델을 활용하여 자동주식 매매 시스템을 구축 예정

참 고 문 헌

1. 강화학습을 이용한 주가 예측 Stock price prediciton using reinforcement learing

2. 딥러닝기반 강화학습 모델 성능비교 -국내 주식시장 사례연구

3. 딥러닝을 이용한 주가 예측 모델 = Stock price prediction model using deep learning

4. 빅데이터를 활용한 인공지능 주식 예측 분석

5. 파이썬을 이용한 딥러닝/강화학습 주식투자

6. 인공지능 모델을 활용한 주가 예측성과 분석 : 국제시장 비교 = Prediction Performance of Stock Price using Artificial Intelligence Models : Global Market Approach

7. 주식시장 매매시점 및 포트폴리오 결정을 위한 인공지능시스템의 설계에 관한 연구 = (A) Study for the Design of Fuzzy-Neuro Intelligent Stock Portfolio Management System for Stock Investment Timing and Optimal Portfolio Selection

8. 주식 매매를 위한 강화학습에서의 대조적 표현학습 연구 = Reinforcement Learning for Stock Trading based on Contrastive Representation Learning of Market States

9. 강화학습을 활용한 주식 포트폴리오 구성 및 트레이딩 시뮬레이션 모형에 대한 연구 = A study on the portfolio selection methodology and trading simulation model using reinforcement learning

10. 강화학습을 활용한 주식시장에서의 초단기 매매 수익률 연구 = Short term trading profitability with reinforcement learning

11. 양방향 LSTM 및 강화학습을 이용한 주식 예측 모델 = The stock forecasting model using bidirectional LSTM and reinforcement learning

12. CNN과 LSTM이 결합된 멀티모달 강화학습 기반의 주식 거래 시스템

13. 강화된 다중 예측 네트워크 기반 지능형 주식 트레이드에 관한 연구 : A study on intelligent stock trading based on an enhanced multi-prediction network