

RED

금융 1팀

김경현 김반석 김홍엽 이규복 이형빈



투자, 왜 해야할까?

금리

2020년 12월 27일 기준

한국은행 금리 연 0.5%
미국 연준 금리 연 0.25%

물가

한국 물가상승률 연 0.6%(2020년 11월)

미국 물가상승률 연 1.8%(2019년)

적금

자유적금 이자율 연 1.01%

은행 정기적금(24개월) 최고우대금리
연 2.55%

자산

KODEX 200
2020년 3월부터 현재까지 31.85% 성장

비트코인
지난 1년동안 약 340% 상승

로보어드바이저란?



증권사 등에서 머신러닝 기술 및 퀀트 트레이딩 알고리즘을 활용하여 만든 “주식추천시스템”.

종목과 투자전략을 제시하거나 대신 투자해주면서 수익률의 일부를 수수료로 가져감



투자자들로 하여금 쉽게 투자를 유도할 수 있음



사용자 정보와 수익률이 모두 기록되기 때문에 관리와 분석이 용이

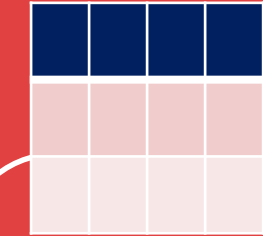


최근에 이용 고객수가 늘어나고 있어 매우 유망함

SW 아키텍처

주가 데이터 크롤링

네이버 금융
DART
한국거래소



투자 지표 생성

Sigma, RSI 등
기술적 분석에
필요한 지표 생성
및 데이터 전처리

인터페이스

Ipywidgets과
이미지처리 모듈로
편리한 인터페이스
및 사용환경 구축

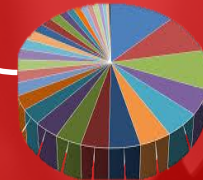


전략알고리즘

기술적 분석을
통한 매매기법을
백테스팅 후 적용

포트폴리오

주식과 ETF에 분배된
포트폴리오 제공.
산업분석 서비스 및
고객정보 반영



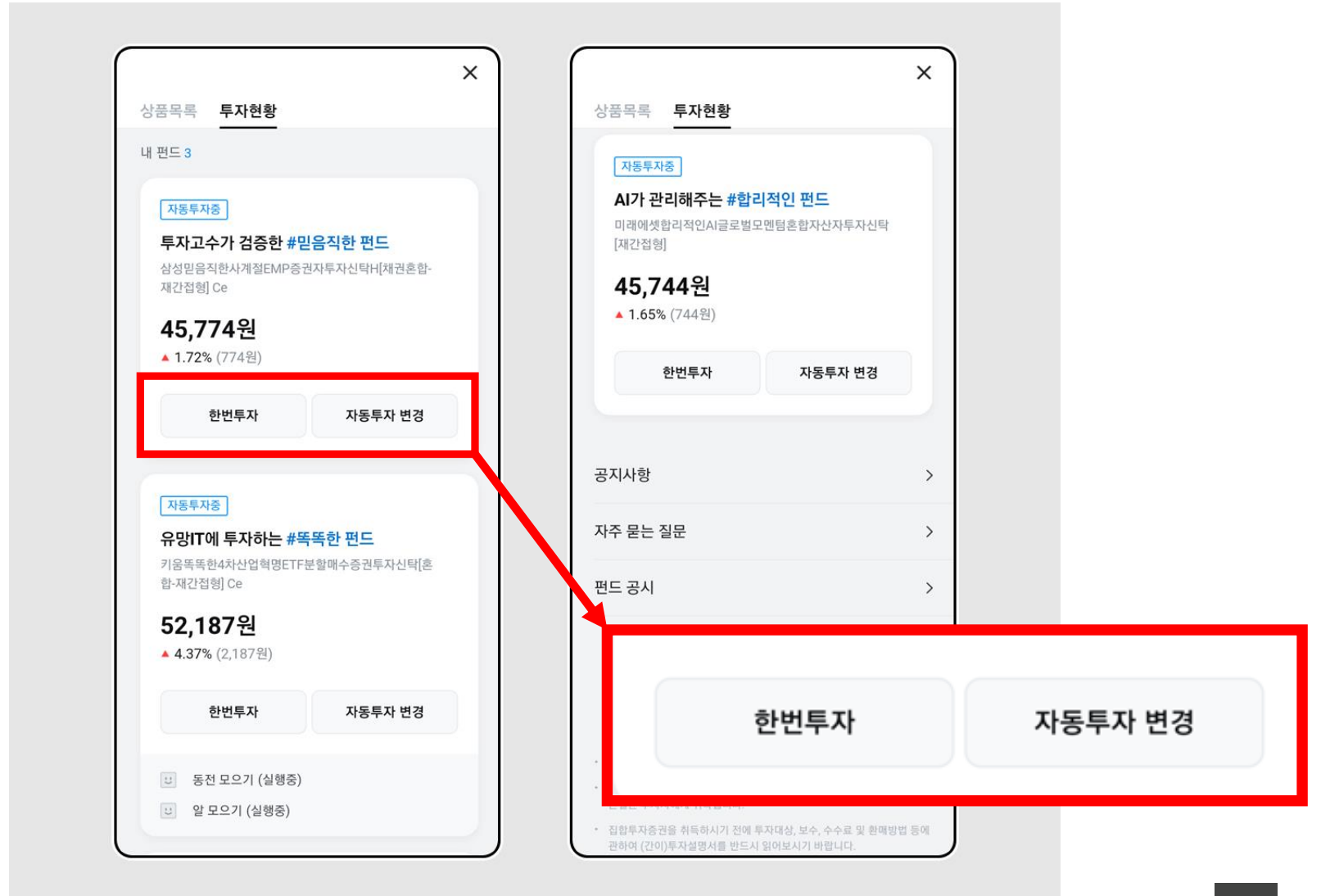
사용자 DB

사용자 정보 저장
및 지속적 관리.
이후 피드백과
수익률 추적 가능

INTERFACE 란?

INTERFACE

사용자가 기기를 쉽게 동작시키는데 도움을 주는 시스템을 의미한다.



EXAMPLE #1

■ 투자정보

01. 고객님의 연령대는 어떻게 되십니까?

- ☐ 19세 이하 ☐ 20세~40세 ☐ 41세~50세 ☐ 51세~60세 ☐ 61세 이상

02. 고객님의 투자하고자 하는 자금의 투자가능기간은 얼마나 되십니까?

- ☐ 6개월 이내 ☐ 6개월 이상~1년 이내 ☐ 1년 이상~2년 이내 ☐ 2년 이상~3년 이내 ☐ 3년 이상

03. 다음 중 고객님의 투자경험과 가장 가까운 금융상품은 어느 것입니까? (중복응답 가능)

- ☐ 은행 예·적금, 국채, 지방채, 보증채, MMF, CMA 등
☐ 금융채, 신용도가 높은 회사채, 채권형 펀드, 원금보장형 ELS 등
☐ 신용도 중간 등급의 회사채, 원금의 일부만 보장되는 ELS, 혼합형 펀드 등
☐ 신용도가 낮은 회사채, 주식, 원금이 보장되지 않는 ELS, 시장수익률 수준의 수익률 추구형 펀드 등
☐ ELW, 선물옵션, 시장수익률 이상의 수익을 추구하는 주식형펀드, 파생상품펀드, 등

04. 고객님의께서는 금융상품 투자에 대한 본인의 지식수준이 어느 정도라고 생각하십니까?

- ☐ 매우 낮은 수준 (투자의사결정을 스스로 내려본 경험이 없는 정도)
☐ 낮은 수준 (주식과 채권의 차이를 구별할 수 있는 정도)
☐ 높은 수준 (투자할 수 있는 대부분의 금융상품의 차이를 구별할 수 있는 정도)
☐ 매우 높은 수준 (금융상품을 비롯하여 모든 투자대상 상품의 차이를 이해할 수 있는 정도)

05. 고객님의 투자하고자 하는 자금은 고객님의 전체 금융자산(부동산 등을 제외) 중 어느 정도의 비중을 차지하십니까?

- ☐ 10% 이내 ☐ 10% 초과~20% 이내 ☐ 20% 초과~30% 이내 ☐ 30% 초과~40% 이내 ☐ 40% 이상

06. 다음 중 고객님의 수입원을 가장 잘 나타내는 것은 어느 것입니까?

- ☐ 현재 일정한 수입이 발생하고 있으며, 향후 현재 수준을 유지하거나 증가할 것으로 예상
☐ 현재 일정한 수입이 발생하고 있으나, 향후 감소하거나 불안정할 것으로 예상
☐ 현재 일정한 수입이 없으며, 연금이 주 수입원임

EXAMPLE #2

I 투자자 기초정보

01 [연령구분] * 현재 고객님의 나이는 어떻게 되십니까?

- ☐ 만65세 미만 ☐ 만 65세 이상

02 [소득상태] * 향후 고객님의 연간수입원에 대한 예상은 어떻게 되십니까?

- ☐ 현재 일정한 수입이 없으며, 연금이 주 수입원임
☐ 현재 일정한 수입이 발생하고 있으나, 향후 감소하거나 불안정할
☐ 현재 일정한 수입이 발생하고 있으며, 향후 현재 수준을 유지하게

03 [투자자금의 비중] 기존에 보유하고 계신 총자산 대비 금융자산의 비중은 어느 정도입니까?

- ☐ 5% 이하 ☐ 10% 이하 ☐ 20% 이하 ☐ 30% 이하 ☐ 30% 초과

04 [투자경험] 투자해 보신 경험이 있는 것을 모두 선택해 주십시오(중복체크 가능)

- ☐ 주식신용거래 ☐ 선물/옵션 ☐ ELW ☐ 원금비보장형ELS/DLS/EL

- ☐ 1년 미만 ☐ 1년 이상 3년 미만 ☐ 3년 이상

- ☐ 주식 ☐ 주식형펀드 ☐ 해외펀드 ☐ 원금보장형ELS/DLS/ELF ☐ 투자자문/일임(Wrap) ☐ 외화증권

- ☐ 1년 미만 ☐ 1년 이상 3년 미만 ☐ 3년 이상

- ☐ 채권/혼합형 펀드 ☐ 신탁 ☐ 채권

- ☐ 1년 미만 ☐ 1년 이상 3년 미만 ☐ 3년 이상

RED INTERFACE

USER INFO

1. 투자금액

2. 투자기간

3. 나 이

4. 성 별

5. 정기수입

6. 관심사업

7. 지식수준

8. 위험추구

PROGRAM

데이터 업데이트

투자시작

뒤로가기

결과저장

· 사용자 정보를 받는 이유?

1. 사용자 정보에 따라 투자 가능한 종목이 달라짐
2. 사용자의 관심사업 위주로 종목 추천이 가능
3. 사용자의 성향에 맞추어 포트폴리오의 추천 비중을 다르게 하기 위해

VISUAL

USER INFO

투자 금액(만 원)

투자 기간

나이 (만)

성별

월 정기 수입(만 원)

관심산업분야

금융지식수준

위험추구성향

PROGRAM

데이터 업데이트

투자 시작

뒤로 가기

데이터 업데이트를 하신 후, 입력하신 정보를 확인하시고 투자 시작을 눌러주세요.

데이터 수집 : 크롤링



*모든 정보는 네이버 금융을 활용

코스피 200	코스닥 150	ETF	재무정보
<ul style="list-style-type: none">• 코스피 시장에 등록되어 있는 주식 중 시가총액 상위 200위의 기업들 선정.• 시가총액이 상위권이고 거래량이 많을 수록 위험성이 덜하다고 판단	<ul style="list-style-type: none">• 코스닥 시장에 상장된 주식 중 시가총액 상위 150위의 기업들 선정• 코스피에 상장된 주식에 비해 크기가 작지만 수익률이 크기 때문에 데이터에 포함하되 150개만 포함	<ul style="list-style-type: none">• 네이버 금융 API에서 제공하는 ETF 데이터 활용• 시장지수• 채권• 일반ETF	<ul style="list-style-type: none">• 네이버 금융에서 제공하는 영업이익률과 PER 활용



1. 종목 이름과 종목 코드 저장 :
KOSPI 200
KOSDAK 150
ETF



2. 종목별 과거 주가 데이터 기록 :
최근 10년



3. 종목별 재무정보 추가 (ETF 제외)

데이터 수집 : 크롤링

최종목표 : 포트폴리오 구성에 활용 가능한 CSV파일 저장

Step 1 종목이름과 종목코드

KOSPI 200

편집종목상위

종목별 ▾	현재가 ▾	전일비 ▾	등락률 ▾	거래량 ▾	거래대금(백만) ▾	시가총액(억) ▾
삼성전자	77,800	▲ 3,900	+5.28%	32,317,535	2,472,545	464,491
SK하이닉스	118,000	▲ 2,000	+1.72%	3,478,433	408,805	859,043
LG화학	818,000	▲ 12,000	+1.49%	270,386	220,534	577,445
삼성바이오로직스	794,000	▼ 2,000	-0.25%	112,920	89,923	525,350
셀트리온	347,500	▼ 7,500	-2.11%	2,022,142	706,635	469,117
NAVER	282,000	▼ 2,000	-0.70%	452,158	127,590	463,223
현대차	187,000	▲ 2,000	+1.08%	1,052,729	197,207	399,560
삼성SDI	563,000	▲ 9,000	+1.62%	359,924	202,677	387,144
카카오	374,000	▼ 3,500	-0.93%	461,503	173,174	330,280
삼성물산	139,000	▲ 2,000	+1.46%	1,172,992	162,249	259,773

1 2 3 4 5 6 7 8 9 10 다음 맨뒤 »

```
def get_kospi200_list():
    stock_list = []
    base_url = "https://finance.naver.com/sise/entryJongmok.nhn?&page="

    for i in range(1, 21):
        try:
            url = base_url + str(i)
            r = requests.get(url)
            soup = BeautifulSoup(r.text, "lxml")
            items = soup.find_all("td", {"class": "ctg"})

            for item in items:
                txt = item.a.get("href")
                k = re.search("#d+", txt)

                if k:
                    code = k.group()
                    name = item.text
                    data = code, name
                    stock_list.append(data)

        except:
            pass

    finally:
        kospi200 = pd.DataFrame(stock_list, columns=["종목코드", "종목명"])

    return kospi200
```

<https://finance.naver.com/item/main.nhn?code=005930>

편입종목상위

종목별 ▾	현재가 ▾	전일비 ▾	등락률 ▾	거래량 ▾	거래대금(백만) ▾	시가총액(억) ▲
삼성전자	77,800	▲ 3,900	+5.28%	32,317,535	2,472,545	4,644,491
SK하이닉스	118,000	▲ 2,000	+1.72%	3,478,433	408,805	859,043
LG화학	818,000	▲ 12,000	+1.49%	270,386	220,534	577,445
삼성바이오로직스	794,000	▼ 2,000	-0.25%	112,920	89,923	525,350
셀트리온	347,500	▼ 7,500	-2.11%	2,022,142	706,635	469,117
NAVER	282,000	▼ 2,000	-0.70%	452,158	127,590	463,223
현대차	187,000	▲ 2,000	+1.08%	1,052,729	197,207	399,560
삼성SDI	563,000	▲ 9,000	+1.62%	359,924	202,677	387,144
카카오	374,000	▼ 3,500	-0.93%	461,503	173,174	330,280
삼성물산	139,000	▲ 2,000	+1.46%	1,172,992	162,249	259,773

1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 다음 ▶ | 맨뒤 ▶▶

데이터 수집 : 크롤링

최종목표 : 포트폴리오 구성에 활용 가능한 CSV파일 저장

Step 1 종목이름과 종목코드

KOSDAQ 150

N	종목명	현재가	전일비	등락률	시가총액	상장주식수	외국인비율	거래량	PER	ROE	토론실
1	삼성전자	168,000	▼ 2,200	-1.29%	1,000	255,040	151,810	17,02	3,855.518	102.00	3.94
2	SK하이닉스	240,000	▲ 4,800	+2.04%	500	85,946	35,811	9.00	1,575.253	597.01	3.13
3	LG화학	187,700	▲ 1,100	+0.59%	500	49,241	26,234	17.62	813.471	15.32	19.01
4	삼성바이오	89,100	▼ 700	-0.78%	500	47,196	52,970	11.74	863.875	-106.32	-24.69
5	삼성물산	156,600	▲ 6,200	+4.12%	500	43,978	28,083	19.41	1,138.905	283.18	-1.03
6	SK이노비오	333,500	0	0.00%	500	35,176	10,548	15.77	75.856	28.79	30.22
7	삼성바이오	268,500	▲ 3,700	+1.40%	500	35,119	13,080	23.29	36.929	23.71	31.62
8	삼성바이오	164,600	▲ 4,600	+2.88%	500	34,624	21,035	6.48	289.456	91.90	12.99
9	삼성바이오	45,250	▲ 350	+0.78%	100	33,508	74,052	0.00	451.108	229.70	2.64
10	삼성바이오	77,400	▲ 400	+0.52%	500	30,821	30,821	9.11	708.562	221.74	67.76

```
def get_kosdak150_list():  
    """'https://finance.naver.com/'에서 kosdak150의 종목코드와 종목명을 스크래핑합니다.  
    Returns: kospi200의 종목코드와 종목명 dataframe  
    """  
  
    stock_list = []  
    base_url = "https://finance.naver.com/sise/sise_market_sum.nhn?sosok=1&page="
```

for i in range(1, 4):
 url = base_url + str(i)
 r = requests.get(url)
 soup = BeautifulSoup(r.text, "lxml")
 items = soup.find("table", {"class": "type_2"}).find_all("a")

for j in range(1, 51):
 txt = items[2 * j - 2].get("href")
 k = re.search("[\w\d]+", txt)
 if k:
 code = k.group()
 name = items[2 * j - 2].text
 data = code, name
 stock_list.append(data)
 else:
 pass

kosdak150 = pd.DataFrame(stock_list, columns=["종목코드", "종목명"])

return kosdak150

https://finance.naver.com/item/main.nhn?code=091990

금융 홈 **국내증시** 해외증시 시장지표 펀드 리서치 뉴스 MY

국내증시

주요시세정보

코스피 | 코스닥 | 선물

코스피200 | 코넥스

시가총액 | 배당

업종 | 테마 | 그룹사

ETF | ETN

상승 | 보합 | 하락

상한가 | 하한가

급등 | 급락

거래상위 | 급증 | 급감

투자자별매매동향

외국인매매 | 기관매매

프로그램매매동향

증시자금동향

신규상장

외국인보유

장외시세

IPO

투자자보호

관리종목

거래정지종목

시장경보종목

금융홈 > 국내증시 > 시가총액

시가총액

항목을 자유롭게 변경하실 수 있습니다. | 최대 6개까지 설정 가능합니다.

<input checked="" type="checkbox"/> 거래량	<input type="checkbox"/> 매수호가	<input type="checkbox"/> 거래대금(백만)	<input checked="" type="checkbox"/> 시가총액(억)	<input type="checkbox"/> 영업이익(억)	<input checked="" type="checkbox"/> PER(배)
<input type="checkbox"/> 시가	<input type="checkbox"/> 매도호가	<input type="checkbox"/> 전일거래량	<input type="checkbox"/> 자산총계(억)	<input type="checkbox"/> 영업이익증가율	<input checked="" type="checkbox"/> ROE(%)
<input type="checkbox"/> 고가	<input type="checkbox"/> 매수총잔량	<input checked="" type="checkbox"/> 외국인비율	<input type="checkbox"/> 부채총계(억)	<input type="checkbox"/> 당기순이익(억)	<input type="checkbox"/> ROA(%)
<input type="checkbox"/> 저가	<input type="checkbox"/> 매도총잔량	<input checked="" type="checkbox"/> 상장주식수(천주)	<input type="checkbox"/> 매출액(억)	<input type="checkbox"/> 주당순이익(원)	<input type="checkbox"/> PBR(배)
			<input type="checkbox"/> 매출액증가율	<input type="checkbox"/> 보통주배당금(원)	<input type="checkbox"/> 유보율(%)

[적용하기](#) [초기 항목으로](#)

코스피		코스닥										
N	종목명	현재가	전일비	등락률	액면가	시가총액	상장주식수	외국인비율	거래량	PER	ROE	토론실
1	셀트리온헬스케어	168,000	▼ 2,200	-1.29%	1,000	255,040	151,810	17.02	3,855,518	102.00	3.94	
2	셀트리온제약	240,000	▲ 4,800	+2.04%	500	85,946	35,811	9.00	1,575,253	597.01	3.13	
3	씨젠	187,700	▲ 1,100	+0.59%	500	49,241	26,234	17.62	813,471	15.32	19.01	
4	에이치엘비	89,100	▼ 700	-0.78%	500	47,196	52,970	11.74	863,875	-106.32	-24.69	
5	알테오젠	156,600	▲ 6,200	+4.12%	500	43,978	28,083	19.41	1,138,905	283.18	-1.03	
6	SK머티리얼즈	333,500	0	0.00%	500	35,176	10,548	15.77	75,856	28.79	30.22	
7	펄어비스	268,500	▲ 3,700	+1.40%	500	35,119	13,080	23.29	36,929	23.71	31.62	
8	에코프로비엠	164,600	▲ 4,600	+2.88%	500	34,624	21,035	6.48	289,456	91.90	12.99	
9	카카오게임즈	45,250	▲ 350	+0.78%	100	33,508	74,052	0.00	451,108	229.70	2.64	
10	케이엠더블유	77,400	▲ 400	+0.52%	500	30,821	39,821	9.11	708,562	231.74	67.76	

데이터 수집 : 크롤링

최종목표 : 포트폴리오 구성에 활용 가능한 CSV파일 저장

Step 1 종목이름과 종목코드

ETF

```
← → ↺ ↻ 🔍 ☆ 📄 👤 ...
https://finance.naver.com/api/sise/...

{"resultCode":"success","result":{"etfItemList":[{"itemcode":"069500","etfTabCode":1,"itemname":"KODEX 200","nowVal":37850,"riseFall":2,"changeVal":740,"changeRate":1.93,"nav":37963.0,"threeMonthEarnRate":25.4682,"quantity":8705601,"amout":327439,"marketSum":507383}, {"itemcode":"102110","etfTabCode":1,"itemname":"TIGER 200","nowVal":37845,"riseFall":2,"changeVal":770,"changeRate":2.08,"nav":37943.0,"threeMonthEarnRate":25.5392,"quantity":1315297,"amout":49471,"marketSum":26813}, {"itemcode":"252670","etfTabCode":3,"itemname":"KODEX 200선물인버스 2X","nowVal":2750,"riseFall":5,"changeVal":120,"changeRate":4.18,"nav":2737.0,"threeMonthEarnRate":37.9393,"quantity":251089384,"amout":700002,"marketSum":21172}, {"itemcode":"153130","etfTabCode":6,"itemname":"KODEX 단기채권","nowVal":102620,"riseFall":2,"changeVal":5,"changeRate":0.0,"nav":102616.0,"threeMonthEarnRate":0.1317,"quantity":50869,"amout":5219,"marketSum":18960}, {"itemcode":"310370","etfTabCode":1,"itemname":"TIGER MSCI Korea TR","nowVal":15435,"riseFall":2,"changeVal":370,"changeRate":2.46,"nav":15440.0,"threeMonthEarnRate":26.9325,"quantity":260100,"amout":3982,"marketSum":16531}, {"itemcode":"278540","etfTabCode":1,"itemname":"KODEX MSCI Korea TR","nowVal":12285,"riseFall":2,"changeVal":265,"changeRate":2.2,"nav":12296.0,"threeMonthEarnRate":27.3066,"quantity":715948,"amout":8701,"marketSum":16044}, {"itemcode":"273130","etfTabCode":6,"itemname":"KODEX 종합채권(AA-이상)액티브","nowVal":110085,"riseFall":5,"changeVal":155,"changeRate":0.14,"nav":110047.0,"threeMonthEarnRate":0.7439,"quantity":6514,"amout":717,"marketSum":14673}, {"itemcode":"122630","etfTabCode":3,"itemname":"KODEX 레버리지","nowVal":21545,"riseFall":2,"changeVal":770,"changeRate":3.71,"nav":21830.0,"threeMonthEarnRate":54.8329,"quantity":61296374,"amout":1305633,"marketSum":14090}, {"itemcode":"278530","etfTabCode":1,"itemname":"KODEX 200TR","nowVal":12160,"riseFall":2,"changeVal":210,"changeRate":1.76,"nav":12191.0,"threeMonthEarnRate":25.5549,"quantity":257262,"amout":3116,"marketSum":14027}, {"itemcode":"102780","etfTabCode":2,"itemname":"KODEX 삼성그룹","nowVal":9380,"riseFall":2,"changeVal":165,"changeRate":1.79,"nav":9433.0,"threeMonthEarnRate":28.0646,"quantity":793455,"amout":7428,"marketSum":13742}, {"itemcode":"233740","etfTabCode":1,"itemname":"KODEX 코스닥150 레버리지","nowVal":15850,"riseFall":2,"changeVal":345,"changeRate":2.23,"nav":15972.0,"threeMonthEarnRate":38.4884,"quantity":37120544,"amout":566102,"marketSum":11773}, {"itemcode":"148020","etfTabCode":1,"itemname":"KBS1AR 200","nowVal":37910,"riseFall":2,"changeVal":770,"changeRate":2.07,"nav":38003.0,"threeMonthEarnRate":25.5398,"quantity":333958,"amout":12565,"marketSum":11411}, {"itemcode":"157450","etfTabCode":6,"itemname":"TIGER 단기통안채","nowVal":100795,"riseFall":3,"changeVal":0,"changeRate":0.0,"nav":100794.0,"threeMonthEarnRate":0.1241,"quantity":3623,"amout":365,"marketSum":9866}, {"itemcode":"114600","etfTabCode":3,"itemname":"KODEX 인버스"}]}
```

```
def get_etf_list():
    """'https://finance.naver.com/'에서 ETF 리스트를 스크래핑합니다.
    Returns: etf list를 dataframe으로 반환합니다.
    """

    # 네이버 금융 API로부터 ETF 데이터 갖고오기
    url = "https://finance.naver.com/api/sise/etfItemList.nhn"
    json_data = json.loads(requests.get(url).text)
    df = json_normalize(json_data["result"]["etfItemList"])

    # etf 데이터프레임 정제
    # etfTabCode = {1: 국내 시장지수, 4: 해외 주식, 6: 채권}
    etf_df = df[["itemcode", "etfTabCode", "itemname"]]
    etf_df = etf_df[etf_df["etfTabCode"].isin([1, 4, 6])]

    etf_df["etfTabCode"].loc[etf_df["etfTabCode"] == 1] = "국내시장지수"
    etf_df["etfTabCode"].loc[etf_df["etfTabCode"] == 4] = "해외주식"
    etf_df["etfTabCode"].loc[etf_df["etfTabCode"] == 6] = "채권"

    return etf_df
```

```
{
  "resultCode": "success",
  "result": {
    "etfItemList": [
      {
        "itemcode": "069500",
        "etfTabCode": 1,
        "itemname": "KODEX 200",
        "nowVal": 37850,
        "risefall": "2",
        "changeVal": 740,
        "changeRate": 1.99,
        "nav": 37963.0,
        "threeMonthEarnRate": 25.4682,
        "quant": 8705601,
        "amonut": 327439,
        "marketSum": 50738,
        "itemcode": "102110",
        "etfTabCode": 1,
        "itemname": "TIGER 200",
        "nowVal": 37845,
        "risefall": "2",
        "changeVal": 770,
        "changeRate": 2.08,
        "nav": 37943.0,
        "threeMonthEarnRate": 25.5932,
        "quant": 1315297,
        "amonut": 49471,
        "marketSum": 26813,
        "itemcode": "252670",
        "etfTabCode": 3,
        "itemname": "KODEX 200선물인버스 2X",
        "nowVal": 2750,
        "risefall": "5",
        "changeVal": -120,
        "changeRate": -4.18,
        "nav": 2737.0,
        "threeMonthEarnRate": -37.9933,
        "quant": 251088384,
        "amonut": 700002,
        "marketSum": 21172,
        "itemcode": "153130",
        "etfTabCode": 6,
        "itemname": "KODEX 단기채권",
        "nowVal": 102620,
        "risefall": "2",
        "changeVal": 5,
        "changeRate": 0.0,
        "nav": 102616.0,
        "threeMonthEarnRate": 0.1317,
        "quant": 50869,
        "amonut": 5219,
        "marketSum": 18960,
        "itemcode": "310970",
        "etfTabCode": 1,
        "itemname": "TIGER MSCI Korea TR",
        "nowVal": 15435,
        "risefall": "2",
        "changeVal": 370,
        "changeRate": 2.46,
        "nav": 15440.0,
        "threeMonthEarnRate": 26.9325,
        "quant": 260100,
        "amonut": 3982,
        "marketSum": 16531,
        "itemcode": "278540",
        "etfTabCode": 1,
        "itemname": "KODEX MSCI Korea TR",
        "nowVal": 12285,
        "risefall": "2",
        "changeVal": 265,
        "changeRate": 2.2,
        "nav": 12296.0,
        "threeMonthEarnRate": 27.3056,
        "quant": 715949,
        "amonut": 8701,
        "marketSum": 16044,
        "itemcode": "273130",
        "etfTabCode": 6,
        "itemname": "KODEX 종합채권(AA-이상)액티브",
        "nowVal": 110085,
        "risefall": "5",
        "changeVal": -155,
        "changeRate": -0.14,
        "nav": 110047.0,
        "threeMonthEarnRate": -0.7439,
        "quant": 6514,
        "amonut": 717,
        "marketSum": 14673,
        "itemcode": "122630",
        "etfTabCode": 3,
        "itemname": "KODEX 레버리지",
        "nowVal": 21545,
        "risefall": "2",
        "changeVal": 770,
        "changeRate": 3.71,
        "nav": 21830.0,
        "threeMonthEarnRate": 54.8329,
        "quant": 61296374,
        "amonut": 1305633,
        "marketSum": 14090,
        "itemcode": "278530",
        "etfTabCode": 1,
        "itemname": "KODEX 200TR",
        "nowVal": 12160,
        "risefall": "2",
        "changeVal": 210,
        "changeRate": 1.76,
        "nav": 12191.0,
        "threeMonthEarnRate": 25.5549,
        "quant": 257262,
        "amonut": 3116,
        "marketSum": 14027,
        "itemcode": "102780",
        "etfTabCode": 2,
        "itemname": "KODEX 삼성그룹",
        "nowVal": 9380,
        "risefall": "2",
        "changeVal": 165,
        "changeRate": 1.79,
        "nav": 9433.0,
        "threeMonthEarnRate": 28.0546,
        "quant": 793455,
        "amonut": 7428,
        "marketSum": 13742,
        "itemcode": "233740",
        "etfTabCode": 1,
        "itemname": "KODEX 코스닥150 레버리지",
        "nowVal": 15850,
        "risefall": "2",
        "changeVal": 345,
        "changeRate": 2.23,
        "nav": 15972.0,
        "threeMonthEarnRate": 38.4884,
        "quant": 37120544,
        "amonut": 586102,
        "marketSum": 11777,
        "itemcode": "148020",
        "etfTabCode": 1,
        "itemname": "KBSTAR 200",
        "nowVal": 37910,
        "risefall": "2",
        "changeVal": 770,
        "changeRate": 2.07,
        "nav": 38003.0,
        "threeMonthEarnRate": 25.5838,
        "quant": 333958,
        "amonut": 12565,
        "marketSum": 11411,
        "itemcode": "157450",
        "etfTabCode": 6,
        "itemname": "TIGER 단기통안채",
        "nowVal": 100795,
        "risefall": "3",
        "changeVal": 0,
        "changeRate": 0.0,
        "nav": 100794.0,
        "threeMonthEarnRate": 0.1241,
        "quant": 3623,
        "amonut": 365,
        "marketSum": 9866,
        "itemcode": "114800",
        "etfTabCode": 3,
        "itemname": "KODEX 인버스 2X",
        "nowVal": 2750,
        "risefall": "5",
        "changeVal": -120,
        "changeRate": -4.18,
        "nav": 2737.0,
        "threeMonthEarnRate": -37.9933,
        "quant": 251088384,
        "amonut": 700002,
        "marketSum": 21172
      }
    ]
  }
}
```

데이터 수집 : 크롤링

KOSPI 200

KOSDAQ 150

ETF

A	B	C	D
	종목코드	종목명	분류
0	5930	삼성전자	kospi200
1	660	SK하이닉스	kospi200
2	51910	LG화학	kospi200
3	207940	삼성바이오로직스	kospi200
4	68270	셀트리온	kospi200
5	35420	NAVER	kospi200
6	5380	현대차	kospi200
7	6400	삼성SDI	kospi200
8	35720	카카오	kospi200
9	28260	삼성물산	kospi200
10	51900	LG생활건강	kospi200

200	91990	셀트리온헬스케어	kosdak150
201	68760	셀트리온제약	kosdak150
202	96530	씨젠	kosdak150
203	28300	에이치엘비	kosdak150
204	196170	알테오젠	kosdak150
205	36490	SK머티리얼즈	kosdak150
206	263750	펄어비스	kosdak150
207	247540	에코프로비엠	kosdak150
208	293490	카카오게임즈	kosdak150
209	32500	케이엠더블유	kosdak150
210	35760	CJ ENM	kosdak150
211	95700	제넥신	kosdak150
212	253450	스튜디오드래곤	kosdak150

A	B	C
itemcode	etfTabCode	itemname
102110	국내시장지수	TIGER 200
273130	채권	KODEX 종합채권(AA-이상)액티브
157450	채권	TIGER 단기통안채
214980	채권	KODEX 단기채권PLUS
272580	채권	TIGER 단기채권액티브
133690	해외주식	TIGER 미국나스닥100
226490	국내시장지수	KODEX 코스피
251350	해외주식	KODEX 선진국MSCI World
148070	채권	KOSEF 국고채10년
273140	채권	KODEX 단기변동금리부채권액티브
219480	해외주식	KODEX 미국S&P500선물(H)
292190	국내시장지수	KODEX KRX300
360750	해외주식	TIGER 미국S&P500

데이터 수집 : 크롤링

최종목표 : 포트폴리오 구성에 활용 가능한 CSV파일 저장

Step 2 종목별 과거 주가 데이터 저장

종목별로 최근 10년 간의 주가 데이터를 저장한다.

```
def get_history(itemcode, years):  
    today = datetime.datetime.now()  
    duration = float(years) * 365  
    real_duration = float(duration) / 1.47  
    url = f"https://fchart.stock.naver.com/sise.nhn?symbol={itemcode}"  
    result = requests.get(url)  
    bs_obj = BeautifulSoup(result.content, "html.parser")  
    price_data = bs_obj.select("item")  
    data_dict = {}  
    date_list = []  
    start_price_list = []  
    high_price_list = []  
    low_price_list = []  
    close_price_list = []  
    amount_list = []  
    for i in range(len(price_data)):  
        temp = str(price_data[i])[12:-9]  
        temp_list = temp.split("|")  
        date = temp_list[0]  
        start_price = float(temp_list[1])  
        high_price = float(temp_list[2])  
        low_price = float(temp_list[3])  
        close_price = float(temp_list[4])  
        amount = float(temp_list[5])  
        date_list.append(date)  
        start_price_list.append(start_price)  
        high_price_list.append(high_price)  
        low_price_list.append(low_price)  
        close_price_list.append(close_price)  
        amount_list.append(amount)  
    data_dict["start price"] = start_price_list  
    data_dict["high price"] = high_price_list  
    data_dict["low price"] = low_price_list  
    data_dict["close price"] = close_price_list  
    data_dict["amount"] = amount_list  
    stock_data = pd.DataFrame(data_dict, index=date_list)  
    return stock_data
```

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<protocol>  
  <chartdata symbol="005930" name="삼성전자" count="3" timeframe="day" precision="0" origintime="19900103">  
    <item data="20201222|72500|73200|72100|72300|16304910"/>  
    <item data="20201223|72400|74000|72300|73900|19411326"/>  
    <item data="20201224|74100|78800|74000|77800|32317535"/>  
  </chartdata>  
</protocol>
```

삼성전자

A	B	C	D	E
20201211	73500	73800	73100	73400
20201214	73600	74500	73200	73800
20201215	73800	74100	73300	73800
20201216	74100	74500	73400	73800
20201217	73400	73700	72600	73300
20201218	73300	73700	73000	73000
20201221	73100	73400	72000	73000
20201222	72500	73200	72100	72300
20201223	72400	74000	72300	73900
20201224	74100	78800	74000	77800

데이터 수집 : 크롤링

최종목표 : 포트폴리오 구성에 활용 가능한 CSV파일 저장

Step 3 재무정보 저장

ETF를 제외한 코스피, 코스닥 데이터에 재무정보를 추가한다.

기업실적분석										
주요재무정보	최근 연간 실적				최근 분기 실적					
	2017.12	2018.12	2019.12	2020.12 (E)	2019.09	2019.12	2020.03	2020.06	2020.09	2020.12 (E)
	IFRS 연결	IFRS 연결	IFRS 연결	IFRS 연결	IFRS 연결	IFRS 연결	IFRS 연결	IFRS 연결	IFRS 연결	IFRS 연결
매출액(억원)	2,395,754	2,437,714	2,304,009	2,377,337	620,035	598,848	553,252	529,661	669,642	624,795
영업이익(억원)	536,450	588,867	277,685	368,941	77,779	71,603	64,473	81,463	123,532	99,512
당기순이익(억원)	421,867	443,449	217,389	277,163	62,877	52,270	48,849	55,551	93,607	78,142
영업이익률(%)	22.39	24.16	12.05	15.52	12.54	11.96	11.65	15.38	18.45	15.93
순이익률(%)	17.61	18.19	9.44	11.66	10.14	8.73	8.83	10.49	13.98	12.51
ROE(%)	21.01	19.63	8.69	10.39	10.05	8.69	8.45	8.49	9.51	
부채비율(%)	40.68	36.97	34.12		34.14	34.12	34.19	32.67	36.09	
당좌비율(%)	181.61	204.12	233.57		235.80	233.57	237.80	250.04	229.69	
유보율(%)	24,536.12	27,531.92	28,856.02		28,541.64	28,856.02	29,134.12	29,477.97	30,242.29	
EPS(원)	5,421	6,024	3,166	4,046	899	770	720	808	1,364	1,090
PER(배)	9.40	6.42	17.63	19.23	13.73	17.63	15.24	16.52	15.89	71.36
BPS(원)	30,427	35,342	37,528	40,330	37,600	37,528	38,053	38,534	39,446	40,330
PBR(배)	1.67	1.09	1.49	1.93	1.30	1.49	1.25	1.37	1.48	1.93

```
def get_finance(itemcode):  
    URL = "https://finance.naver.com/item/main.nhn?code=" + itemcode  
    stock_info = requests.get(URL)  
    html = stock_info.text  
    soup = BeautifulSoup(html, "html.parser")  
  
    finance_html = soup.select("div.section.cop_analysis div.sub_section")[0]  
  
    th_data = [item.get_text().strip() for item in finance_html.select("thead th")]  
    annual_date = th_data[3:7] # 연 데이터  
    quarter_date = th_data[7:13] # 분기 데이터  
  
    finance_index = [  
        item.get_text().strip() for item in finance_html.select("th.h_th2")  
    ][3:]  
    finance_data = [item.get_text().strip() for item in finance_html.select("td")]  
    finance_data = np.array(finance_data)  
    finance_data.resize(len(finance_index), 10)  
    finance_date = annual_date + quarter_date  
    finance = pd.DataFrame(  
        data=finance_data[0:, 0:], index=finance_index, columns=finance_date  
    )  
    return finance.T # 날짜 데이터를 가로로, 항목들을 세로로 두기 위해 설정
```

종목별 재무 정보 저장

KOSPI 200

A	B	C	D	E	F	G
	종목코드	종목명	분류	영업이익률	PER	
0	5930	삼성전자	kospi200	['18.45', '15.93']	['15.89', '71.36']	
1	660	SK하이닉스	kospi200	['15.99', '11.68']	['21.31', '117.68']	
2	51910	LG화학	kospi200	['12.02', '9.44']	['61.10', '121.79']	
3	207940	삼성바이오로직스	kospi200	['34.14', '23.75']	['141.20', '661.05']	
4	68270	셀트리온	kospi200	['44.70', '46.46']	['68.24', '265.83']	
5	35420	NAVER	kospi200	['21.44', '20.62']	['59.03', '193.36']	
6	5380	현대차	kospi200	['-1.14', '5.91']	['42.65', '35.52']	
7	6400	삼성SDI	kospi200	['8.66', '8.99']	['147.91', '123.48']	
8	35720	카카오	kospi200	['10.92', '11.85']	['-404.46', '270.82']	
9	28260	삼성물산	kospi200	['2.74', '3.49']	['17.59', '104.41']	
10	51900	LG생활건강	kospi200	['15.82', '12.46']	['32.44', '178.90']	
11	270	기아차	kospi200	['1.20', '5.81']	['21.79', '28.52']	
12	12330	현대모비스	kospi200	['5.99', '6.24']	['14.03', '32.50']	
13	5490	POSCO	kospi200	['4.67', '5.48']	['17.95', '46.69']	
14	17670	SK텔레콤	kospi200	['7.64', '5.48']	['17.53', '78.99']	
15	36570	엔씨소프트	kospi200	['37.20', '32.82']	['31.54', '144.27']	
16	105560	KB금융	kospi200	['12.97', '13.45']	['4.44', '4.58']	

KOSDAQ 150

A	B	C	D	E	F	G
206	263750	펄어비스	kosdak150	['34.27', '33.69']	['17.86', '102.76']	
207	247540	에코프로비엠	kosdak150	['1.11', '7.15']	['30.37', '209.84']	
208	293490	카카오게임즈	kosdak150	['13.64', '12.37']	['35.98', '186.87']	
209	32500	케이엠더블유	kosdak150	['8.12', '15.33']	['235.40', '316.66']	
210	35760	CJ ENM	kosdak150	['8.90', '8.11']	['52.94', '77.84']	
211	95700	제넥신	kosdak150	['-120.78', '-356.78']	['-462.55', '-87.05']	
212	253450	스튜디오드래곤	kosdak150	['15.07', '10.52']	['76.67', '237.18']	
213	145020	휴젤	kosdak150	['39.14', '38.43']	['59.48', '142.62']	
214	235980	메드팩토	kosdak150	[]	[]	
215	240810	원익IPS	kosdak150	['27.23', '-1.21']	['11.06', '-919.79']	
216	357780	솔브레인	kosdak150	[]	[]	
217	58470	리노공업	kosdak150	['38.57', '37.40']	['30.43', '130.44']	
218	66970	엘앤에프	kosdak150	['-1.57', '1.09']	['-316.73', '1,775.35']	
219	78340	컴투스	kosdak150	['20.55', '23.46']	['16.33', '68.02']	
220	278280	천보	kosdak150	['18.47', '19.41']	['74.67', '302.11']	
221	237690	에스티팜	kosdak150	['-14.43', '-34.85']	['-63.78', '-74.75']	
222	5290	동진세미캠	kosdak150	['13.04', '14.26']	['18.46', '17.86']	

보완해야할 점

ETF 섹터 구분

ETF 크롤링 과정에서 섹터 정보를 가져오지 못하여 임시로 수작업으로 섹터를 구분함
ETF의 경우 짧은 주기로 종목이 변하기 때문에 수작업은 프로그램의 효율을 낮춤

102780	섹터	KODEX 삼성그룹	기타
139260	섹터	TIGER 200 IT	IT
305720	섹터	KODEX 2차전지산업	화학
364980	섹터	TIGER KRX2차전지K-뉴딜	화학
91170	섹터	KODEX 은행	금융
143860	섹터	TIGER 헬스케어	의약
91180	섹터	KODEX 자동차	운수장비
368190	섹터	HANARO Fn K-뉴딜디지털플러스	IT
364970	섹터	TIGER KRX바이오K-뉴딜	의약
91160	섹터	KODEX 반도체	전기전자
102970	섹터	KODEX 증권	금융
329200	섹터	TIGER 부동산인프라고배당	건설
157490	섹터	TIGER 소프트웨어	IT
365000	섹터	TIGER KRX인터넷K-뉴딜	IT
326240	섹터	KBSTAR IT플러스	IT
322400	섹터	HANARO e커머스	IT
91230	섹터	TIGER 반도체	전기전자

전체	국내 시장지수	국내 업종/테마		국내 파생	해외 주식	원자재
종목명	현재가 ▲	전일비 ▲	등락률 ▲	NAV ▲	3개월수익률 ▲	거래량 ▲
KODEX 삼성그룹	9,380	▲ 165	+1.79%	9,433	+28.05%	793,455
TIGER TOP10	13,000	▲ 205	+1.60%	13,014	+25.30%	561,228
TIGER 200 IT	33,255	▲ 365	+1.11%	33,264	+30.23%	709,267
TIGER KRX BBIG K-뉴딜	11,530	▲ 30	+0.26%	11,536	N/A	417,971
TIGER 2차전지테마	14,050	▲ 135	+0.97%	14,060	+28.72%	1,030,150
KODEX Top5PlusTR	16,965	▲ 375	+2.26%	16,981	+24.24%	16,206
KODEX 2차전지산업	15,040	▲ 195	+1.31%	15,078	+33.57%	1,807,882
TIGER KRX 2차전지K-뉴딜	13,100	▲ 120	+0.92%	13,098	N/A	430,958
KODEX 은행	6,215	▲ 95	+1.55%	6,246	+24.18%	510,437
ARIRANG 고배당주	10,755	▲ 140	+1.32%	10,809	+23.76%	79,126



주식 선별



기술적 분석

- 과거의 주가나 거래량 등을 계량화 또는 도표화하여, 일정한 추세나 패턴을 찾아 시세를 예측하는 방법
- 차트 등을 이용하여 투자심리, 매매시점, 주가동향을 예측



기본적 분석

- 전통적인 증권 분석 방법으로, 기업의 재무요인 등에 따라 결정된 주식의 내재적 가치를 분석하여 미래의 주가를 예측하는 방법

주식 선정 투자 지표 (기술적 분석)

캔들 (양봉 / 음봉)	이동평균 (Moving Average)	Sigma
<ul style="list-style-type: none"> 시가보다 종가가 높은 경우, 양봉 시가보다 종가가 낮은 경우 음봉 <div> <div> <div>캔들 (양봉)</div> <div> <div>고가</div> <div>종가</div> <div>시가</div> <div>저가</div> </div> </div> <div> <div>캔들 (음봉)</div> <div> <div>고가</div> <div>시가</div> <div>종가</div> <div>저가</div> </div> </div> </div>	<ul style="list-style-type: none"> 일정기간 동안의 주가를 산술 평균한 값 일반적으로, 5일(일주일), 20일(1개월), 60일(3개월), 120일(6개월)지표가 쓰임 <div> </div>	<ul style="list-style-type: none"> 변동성 지표 특정 기간 주가의 표준편차와 당일 주가의 차를 비교한 지표 주가의 침체도 과열도를 판단하는 지표 <div> </div>

주식 선정 투자 지표 (기술적 분석)

RSI	MFI	Stochastic	Williams %R
<ul style="list-style-type: none"> Relative Strength Index 가격의 상승 압력과 하락압력간의 상대적인 강도를 나타내는 지표 일정기간(14일)동안의 주가상승폭을 일정기간(14일)동안의 주가하락폭으로 나눈 값 	<ul style="list-style-type: none"> Money Flow Index RSI와 유사한 지표 돈의 흐름인 (주가) * (거래량)을 기준으로 함 RSI는 주가만을 고려, MFI는 거래량까지 반영하여 좀 더 입체적임 	<ul style="list-style-type: none"> 일정 구간의 주가 등락범위 고가와 저가의 현재 주가를 찾기 위함 $Fast\%K = 100 * \frac{(\text{현재 종가} - \text{최저가})}{(\text{최고가} - \text{최저가})}$ Slow 지표는 Fast지표의 이동평균 	<ul style="list-style-type: none"> 스토캐스틱 지표와 유사한 계산법 일정 기간 동안 최저가와 최고가를 중심으로 계산 주가추세전환 예측, 고점, 바닥권 판단하는데 도움

주식 추천 알고리즘



```
def long_candle(stock_name, stock_data, lst):  
    temp = stock_data.iloc[-1, :]  
    s = abs(stock_data.iloc[1:-1, :]['std'].mean()) / stock_data.iloc[-1, :]['close price']  
    cond1 = temp["close price"] > temp["pre_close_price"] * (1 + (5 / 100))  
    cond2 = temp["high price"] > temp["low price"] * (1 + (5 / 100))  
    cond3 = temp["amount"] > temp["amount_ma5"]  
  
    if cond1 and cond2 and cond3:  
        lst.append([stock_name, temp["close price"], "long_candle", 2.16,s])
```

Long Candle

- 저가보다 고가가 5% 이상 큼
- 전날 증가보다 현재 증가가 5% 이상 상승
- 거래량이 5일 평균 거래량보다 많은 경우
▷ 이 세가지를 모두 만족할 경우 추천

```
def gold_cross(stock_name, stock_data, lst):  
    temp = stock_data.iloc[-2, :]  
    s = abs(stock_data.iloc[1:-1, :]['std'].mean()) / stock_data.iloc[-1, :]['close price']  
    if (temp["ma5"] < temp["ma20"]) & (temp["ma5"] > temp["ma60"]) & (temp["ma5"] > temp["ma120"]):  
        if (stock_data.iloc[-1, :]["ma5"] >= stock_data.iloc[-1, :]["ma20"]) & (  
            stock_data.iloc[-1, :]["일수익률"] >= 2  
        ):  
            lst.append([stock_name, stock_data.iloc[-1, :]["close price"], "g", 2.1,s])
```

Gold Cross

- 5일 이평선이 60일, 120일 이평선 이상이고,
20일 이평선을 상향 돌파
- 일 수익률이 2% 이상일 경우
▷ 이 두 가지를 모두 만족할 경우 추천

주식 추천 알고리즘



```
def r_sigma(stock_name, stock_data, lst):  
    temp = stock_data.iloc[-2, :]  
    s = abs(stock_data.iloc[1:-1, :]['std'].mean()) / stock_data.iloc[-1, :]['close price']  
    if (temp["williams"] < -80) & (temp["sigma"] < -2) & (temp["ma20"] < temp["ma60"]):  
        if stock_data.iloc[-1, :]['일수익률'] >= 2.5:  
            lst.append([stock_name, stock_data.iloc[-1, :]['close price'], "r", 1.95,s])
```

R SIGMA

- Williams 지표가 -80 미만, Sigma 값이 -2 미만 (과매도 구간), 20일 이평선이 60일 이평선 미만
- 일 수익률이 2.5 % 이상인 경우
 - ▷ 이 조건을 모두 만족할 경우 추천

```
def mfi_checker(stock_name, stock_data, lst):  
    temp = stock_data.iloc[-1, :]  
    s = abs(stock_data.iloc[1:-1, :]['std'].mean()) / stock_data.iloc[-1, :]['close price']  
    if temp["MFI"] <= 0.2:  
        lst.append([stock_name, stock_data.iloc[-1, :]['close price'], "mfi", 1.65,s])
```

MFI

- MFI 지표가 0.2이하
 - ▷ (과매도 구간인 경우) 추천

주식 추천 알고리즘



```
def rsi_sto_cross(stock_name, stock_data, lst):  
    temp1 = stock_data.iloc[-1, :]  
    temp2 = stock_data.iloc[-2, :]  
    s = abs(stock_data.iloc[1:-1, :]['std'].mean()) / stock_data.iloc[-1, :]['close price']  
    cond1 = temp2["fast_k"] < temp1["fast_k"]  
    cond2 = temp1["fast_k"] > 20  
  
    cond3 = temp2["fast_k"] < temp2["slow_d"]  
    cond4 = temp1["fast_k"] > temp1["slow_d"]  
  
    cond5 = temp2["rsi"] < 0.3  
  
    if (cond1 and cond2 and cond5) or (cond3 and cond4 and cond5):  
        lst.append([stock_name, temp1["close price"], "rsi_sto", 3.67, s])
```

RSI + Stochastic

- Fast_k지표가 전일 대비 상승하고, 20이상
- Fast_k지표가 slow_d 지표를 상향 돌파
 - ▷ 이 조건 중 하나를 만족하면서,
RSI 지표가 0.3미만일 때 추천

주식 선정 투자 지표 (기본적 분석)



PER (Price-Earning Ratio)

- 주가수익비율(Price-Earning Ratio)
- (현재 주가, Price) / (주당 순이익, EPS)
- 주식의 고평가, 저평가를 판단하는 유용한 지표로, 상대적으로 낮을수록 저평가된 주식이라고 할 수 있음

ROE (Return on equity)

- 자기자본이익률(Return on Equity)
- (당기 순이익 ÷ 자기자본) × 100
- 대표적인 수익성 지표, 경영의 효율성을 표시하며, 상대적으로 높은 종목일수록 투자 매력도가 높은 주식

- 그러나 이 지표(분기, 연도지표)는 대부분 주가에 선 반영 되어있는 경우가 많으며, 업종별로 다른 경우가 많기 때문에, 절대적인 지표로 활용하기에 부적합

주식 선정 투자 지표 (기본적 분석)



PER (Price-Earning Ratio)

ROE (Return on equity)

```
recommender = Recommender(self.path, self.stock_path, self.etf_path, self.user_info[5])  
  
recommender.cal_weight()  
rec_stock_lst = recommender.rec_stock()  
  
df = pd.read_csv(self.path + "/data/stock_list2.csv", encoding = 'cp949')  
names = [i[0] for i in rec_stock_lst]  
a = list(df[df['종목명'].isin(names)][['종목명', '가중치']].sort_values(by="가중치", ascending = False).종목명.values)  
rec_stock_lst.sort(key=lambda x: a.index(x[0]))
```

- 따라서, 기술적 분석으로 추천된 주식에서 이러한 재무정보들에 가중치를 부여하여,
추천 우선순위를 정하고, 필터링 하는 방식으로 활용



ETF



상장 지수 펀드

- Exchange Traded Fund: 여러 개의 기초 자산 (주식, 채권 등)을 담은 펀드를 거래소에서 상장시켜, 주식처럼 투자자들이 편리하게 거래할 수 있도록 만든 상품
- EX) KODEX 200: KOSPI 시가총액이 높은 200개의 기업을 추종



활용

- 소액으로도 분산투자를 할 수 있기 때문에, 개별 주식이 가지는 비체계적 위험을 헤지(Hedge) 할 수 있음
- 관심 분야 (섹터)의 ETF 추천
- 위험 성향에 따라, 채권 ETF와 주식형 ETF 비율을 조절

ETF 추천 알고리즘



```
def momentum(etf_name, etf_data, lst1, lst2, bonds):
```

```
    try:
```

```
        profit6 = round(
            (etf_data.iloc[-1, :]["close price"] - etf_data.iloc[-60, :]["close price"])
            / etf_data.iloc[-60, :]["close price"],
            2,
        )
```

```
        profit12 = round(
            (etf_data.iloc[-1, :]["close price"] - etf_data.iloc[-120, :]["close price"])
            / etf_data.iloc[-120, :]["close price"],
            2,
        )
```

```
        if etf_name[:-4] in bonds:
```

```
            profit = (etf_data.iloc[-1, :]["close price"] - etf_data.iloc[-20, :]["close price"]) / etf_data.iloc[-20, :]["close price"]
            s = abs(etf_data.iloc[1:-1, :]['std'].mean()) / etf_data.iloc[-1, :]["close price"]
            lst1.append(etf_name[:-4], etf_data.iloc[-1, :]["close price"], (profit6 + profit12), profit, s)
```

```
        else:
```

```
            etf_data["slope1"] = (etf_data["ma20"] - etf_data["ma20"].shift(10)) / etf_data["ma20"]
```

```
            if (etf_data.iloc[-1, :]["slope1"] > 0.02) & (
                etf_data.iloc[-1, :]["ma20"] > etf_data.iloc[-1, :]["ma60"]
            ):
```

```
                # 중기적으로 상향세일때
```

```
                s = abs(etf_data.iloc[1:-1, :]['std'].mean()) / etf_data.iloc[-1, :]["close price"]
```

```
                lst2.append(
                    [etf_name[:-4], etf_data.iloc[-1, :]["close price"], (profit6 + profit12), 1.21, s]
                )
```

```
    except IndexError:
```

```
        return
```

듀얼 모멘텀 전략



- 투자자산 가운데 강세를 보이는 곳에 투자하는 방식

- 20일 이동평균선, 60일 이동평균선을 활용하여

중기적으로 상향 추세를 가지며,

3개월 / 6개월 간 수익률이 높은 ETF를 추천하는 방식

ETF 추천 알고리즘



```
def rec_etf(self):
    lst1 = [] # 채권 etf
    lst2 = [] # 그외 etf
    print("추천 ETF 찾는 중...")
    etfs = pd.read_csv(self.path + "/data/etf_list.csv", encoding="cp949")
    bonds = etfs[etfs["etfTabCode"] == "채권"]["itemname"].values
    sector_etfs = etfs[etfs["섹터"] == self.sector]["itemname"].values

    for i in os.listdir(self.etf_path):
        etf_data = pd.read_csv(self.etf_path + "/" + i, encoding="cp949")
        etf.momentum(i, etf_data, lst1, lst2, bonds)

    lst1.sort(key=lambda x: x[1])
    lst2.sort(key=lambda x: x[1])
    lst1.sort(key=lambda x: x[2], reverse=True)
    lst2.sort(key=lambda x: x[2], reverse=True)

    for y in lst2:
        if y[0] in sector_etfs:
            lst2.remove(y)
            lst2.insert(0, y) # 맨 앞으로 이동 최우선 추출
            break

    return lst1[:20], lst2[:20]
```

최우선 추천



추천된 ETF를 수익률 순으로 정렬하고,
최초에 입력 받은 관심분야의 ETF를 최우선으로 추천

백 테스트 (Back Test)

“내가 만든 투자 전략을 과거에 똑같이 적용했다면 결과가 어땠을까”에 대한 대답



< SK케미칼 주가 차트 >

나의 투자 전략은

- ✓ 어떤 매수 시점을 제안할까?
- ✓ 선정된 매수 시점은 적절한가?
- ✓ 평균 수익률이 어떻게 될까?
- ✓ 평균 위험률이 어떻게 될까?

백 테스트 (Back Test)

“내가 만든 투자 전략을 과거에 똑같이 적용했다면 결과가 어땠을까”에 대한 대답



< SK케미칼 주가 차트 >

나의 투자 전략은

- ✓ 어떤 매수 시점을 제안할까? 10월 22일
- ✓ 선정된 매수 시점은 적절한가? 94점
- ✓ 평균 수익률이 어떻게 될까? 34점
- ✓ 평균 위험률이 어떻게 될까? 90점

백 테스트 (Back Test)

어느 시점에 매수 하는 것이 좋을까?



< SK케미칼 주가 차트 >

백 테스트 (Back Test)

정답 : 1번, 3번, 4번



< SK케미칼 주가 차트 >

R Sigma 전략

William's R + Sigma

- ✓ 전체 주식종목 대비 기간 수익률 : 1.95%
- ✓ 매수 신호 횟수 : 2,602회
- ✓ 실제 매수 추천 날짜 : 10월 23일
- ✓ 실제 매수 결과 합격



R Sigma 전략

William's R + Sigma

- ✓ 전체 주식종목 대비 기간 수익률 : 1.95%
- ✓ 매수 신호 횟수 : 2,602회
- ✓ 실제 매수 추천 날짜 : 10월 23일
- ✓ 실제 매수 결과 합격



R Sigma 전략

William's R + Sigma

- ✓ 전체 주식종목 대비 기간 수익률 : 1.95%
- ✓ 매수 신호 횟수 : 2,602회
- ✓ 실제 매수 추천 날짜 : 10월 23일
- ✓ 실제 매수 결과 합격



R Sigma 전략

William's R + Sigma

- ✓ 전체 주식종목 대비 기간 수익률 : 1.95%
- ✓ 매수 신호 횟수 : 2,602회
- ✓ 실제 매수 추천 날짜 : 10월 23일
- ✓ 실제 매수 결과 **합격**



RSI Stochastic 전략

Fast K + Slow D + RSI



✓ 전체 주식종목 대비 기간 수익률 : 4.19%

✓ 매수 신호 횟수 : 6,608회

✓ 실제 매수 추천 날짜 : 9월 24일

✓ 실제 매수 결과 합격

RSI Stochastic 전략

Fast K + Slow D + RSI



- ✓ 전체 주식종목 대비 기간 수익률 : 4.19%
- ✓ 매수 신호 횟수 : 6,608회
- ✓ 실제 매수 추천 날짜 : 9월 24일
- ✓ 실제 매수 결과 합격

RSI Stochastic 전략

Fast K + Slow D + RSI



- ✓ 전체 주식종목 대비 기간 수익률 : 4.19%
- ✓ 매수 신호 횟수 : 6,608회
- ✓ 실제 매수 추천 날짜 : 9월 24일
- ✓ 실제 매수 결과 합격

RSI Stochastic 전략

Fast K + Slow D + RSI



✓ 전체 주식종목 대비 기간 수익률 : 4.19%

✓ 매수 신호 횟수 : 6,608회

✓ 실제 매수 추천 날짜 : 9월 24일

✓ 실제 매수 결과

합격

RSI Stochastic 전략



R sigma



Golden
Cross



Long
Candle



MFI

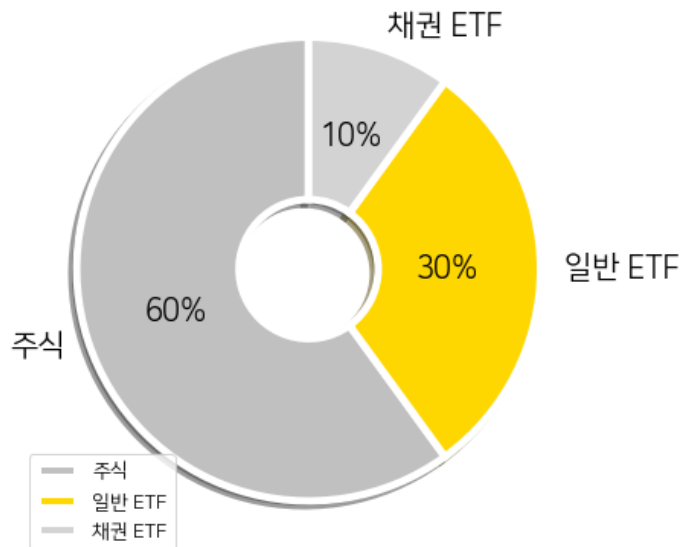


RSI
Stochastic

포트폴리오 (Portfolio)

투자자의 위험성향과 투자 기간 그리고 관심산업을 반영하여, 포트폴리오 구성

Step 1 위험성향별 구성 비율 산정



< 적극 투자형 비율 예 >

Step 2 관심산업 기준 후보군 정렬



Step 3 구성 비율별 포트폴리오 생성

1 주식 종목 : 177,600원

- DB하이텍 : 1개 매입, Long Candle 전략
현재가 : 42,950원
- 삼성전자 : 2개 매입, MFI 전략, 현재가 : 42,950원
- . . .

2 채권 종목 : 29,600원

- TIGER 단기선진하이일드 : 1개 매입, 기간 내 보유 권장,
현재가 12,045원
- KODEX 미국채울트라30년 : 1개 매입, 기간 내 보유 권장,
현재가: 17,555원

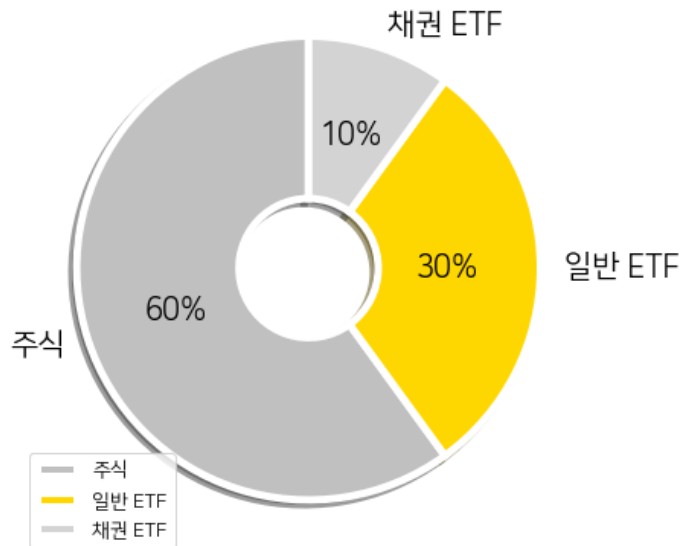
3 일반 종목 : 88,800원

- KODEX 자동차 : 1개 매입, 20일 후 리밸런싱 권장
현재가 18,790원
- KBSTAR 헬스케어 : 1개 매입, 20일 후 리밸런싱 권장
현재가 18,395원

포트폴리오 (Portfolio)

투자자의 5가지 위험성향별로 주식, 채권 ETF, 일반 ETF의 비율을 조절

Step 1 위험성향별 구성 비율 산정



< 적극 투자형 비율 예 >

- 투자 기간이 짧을수록, 채권 ETF 비율 감소
- 위험 성향이 클수록, 주식 투자 비율 증가

고객님의 투자성향
안정형



[고객 정보]

- 나이 : 20세
- 성별 : 여자
- 투자 기간 : 1개월 ~ 6개월
- 투자 금액 : 53만원
- 관심 산업 분야 : 의약

※ 투자 성향은 총 5개 유형으로 공격형, 적극투자형, 중립 추구형, 안정 추구형, 안정형 입니다

고객님의 투자성향
적극 투자형



[고객 정보]

- 나이 : 20세
- 성별 : 여자
- 투자 기간 : 6개월 ~ 1년
- 투자 금액 : 30만원
- 관심 산업 분야 : 의약

※ 투자 성향은 총 5개 유형으로 공격형, 적극투자형, 중립 추구형, 안정 추구형, 안정형 입니다

포트폴리오 (Portfolio)

투자자의 투자 기간과 관심산업을 반영하여, 포트폴리오 내 종목별 후보군 도출

Step 2 관심산업 기준 후보군 정렬

- 우선 순위 정렬 기준 : 수익률, 관심 산업
- 수익률 기준 후보군 정렬 → 관심 산업에 속할 경우, 우선 순위 Rank Up



```
def mk_portfolio(self):  
    """포트폴리오 만드는 함수, r1: ETF비율, r2 : 채권 비율"""  
    capital = self.user_info[0] * 10000  
    if self.user_info[7] == self.risk_list[0]:  
        r1 = 1  
        r2 = 0.67  
    elif self.user_info[7] == self.risk_list[1]:  
        r1 = 0.8  
        r2 = 0.4  
    elif self.user_info[7] == self.risk_list[2]:  
        r1 = 0.6  
        r2 = 0.3  
    elif self.user_info[7] == self.risk_list[3]:  
        r1 = 0.4  
        r2 = 0.1  
    elif self.user_info[7] == self.risk_list[4]:  
        r1 = 0.2  
        r2 = 0  
  
    if self.user_info[1] == self.term_list[0] or self.user_info[1] == self.term_list[1]:  
        r2 = 0 # 투자 기간이 짧으면 채권 제외
```

포트폴리오 (Portfolio)

투자자의 투자 기간과 관심산업을 반영하여, 포트폴리오 내 종목별 후보군 도출

Step 2 관심산업 기준 후보군 정렬

- 우선 순위 정렬 기준 : 수익률, 관심 산업
- 수익률 기준 후보군 정렬 → 관심 산업에 속할 경우, 우선 순위 Rank Up



```
def rec_etf(self):  
    lst1 = [] # 채권 etf  
    lst2 = [] # 그외 etf  
    print("추천 ETF 찾는 중...")  
    etfs = pd.read_csv(self.path + "/data/etf_list.csv", encoding="cp949")  
    bonds = etfs[etfs["etfTabCode"] == "채권"] ["itemname"].values  
    sector_etfs = etfs[etfs["섹터"] == self.sector] ["itemname"].values  
  
    for i in os.listdir(self.etf_path):  
        etf_data = pd.read_csv(self.etf_path + "/" + i, encoding="cp949")  
        etf.momentum(i, etf_data, lst1, lst2, bonds)  
  
    lst1.sort(key=lambda x: x[1])  
    lst2.sort(key=lambda x: x[1])  
    lst1.sort(key=lambda x: x[2], reverse=True)  
    lst2.sort(key=lambda x: x[2], reverse=True)  
  
    for y in lst2:  
        if y[0] in sector_etfs:  
            lst2.remove(y)  
            lst2.insert(0, y) #맨 앞으로 이동 최우선 추출  
            break
```

포트폴리오 (Portfolio)

종목 후보군 내에서 Best Portfolio를 구성하여 제안 / 시각화

Step 3 구성 비율별 포트폴리오 생성

- 앞선 알고리즘을 고려한 포트폴리오 구성 결과
- 자동 시각화 알고리즘 추가

1 주식 종목 : 177,600원

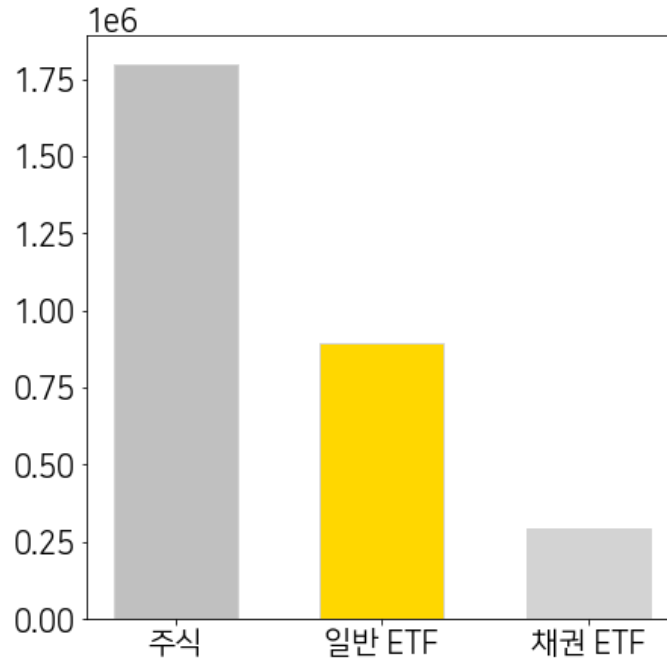
- DB하이텍 : 1개 매입, Long Candle 전략
현재가 : 42,950원
- 삼성전자 : 2개 매입, MFI 전략, 현재가 : 42,950원
- ...

2 채권 종목 : 29,600원

- TIGER 단기선진하이일드 : 1개 매입, 기간 내 보유 권장,
현재가 12,045원
- KODEX 미국채울트라30년 : 1개 매입, 기간 내 보유 권장,
현재가: 17,555원

3 일반 종목 : 88,800원

- KODEX 자동차 : 1개 매입, 20일 후 리밸런싱 권장
현재가 18,790원
- KBSTAR 헬스케어 : 1개 매입, 20일 후 리밸런싱 권장
현재가 18,395원



추천 포트폴리오 구성

DB하이텍
삼성전자
이오테크닉스
빅히트
TIGER 단기선진하이일드(합성 H)
KBSTAR 중기우량회사채
KBSTAR 헬스케어
...

사용자 DB

사용자의 투자 관련 개인 정보와 RED의 추천 포트폴리오를 기록 및 관리

User Interface → 사용자의 투자 관련 개인 정보/ 포트폴리오 추천 시 반영될 투자 성향 정보

RED().run()

"userDB.csv" 파일을 찾을 수 없습니다. 먼저 userDB를 생성하겠습니다.

투자 금액(만 원)

100

투자 기간

1개월 ~ 6개월

나이 (만)

24

성별

남

월 정기 수입(만 원)

0

관심산업분야

IT

금융지식수준

널리 알려진 금융투자 상품(주식, 채권 및 펀드 등)의 구조 및 위험을 깊이 있게 이해하고 있음

위험추구성향

투자원금의 보전보다 수익을 추구 / 투자자금의 대부분을 주식, 옵션 등의 위험자산에 투자할 의향이 있음

데이터 업데이트

투자 시작

뒤로 가기



Class RED

```
def save_info(self, portfolios):
    df = pd.DataFrame(
        np.array(self.user_info).reshape(1, 8),
        columns=[
            "투자 금액(만 원)",
            "투자 기간",
            "나이",
            "성별",
            "월 정기 수입(만 원)",
            "관심산업분야",
            "금융지식수준",
            "위험추구성향",
        ],
    )

    df["주식"] = [list(self.portfolios[0].items())]
    df["채권"] = [list(self.portfolios[1].items())]
    df["ETF"] = [list(self.portfolios[2].items())]
    df["날짜"] = date.today()
    df = pd.concat([self.total_info, df], ignore_index=True)
    df.to_csv(self.path + "/data/users/userDB.csv", encoding="cp949")
    print("저장되었습니다.")
```

사용자 DB

현재 주가를 반영한 피드백과 수익률 추적 가능

CSV userDB.csv

투자 금액 (만 원)	투자 기 간	나 이	성 별	월 정기 수 입(만 원)	관심산 업분야	금융지식수준	위험추구성향	주식	채권	ETF	날짜
0	100	1주 ~ 1개월	24	남	0	IT	널리 알려진 금융투자 상품(주식, 채권 및 펀드 등)의 구조 및 위험을 깊이 있게 ...	투자원금의 보전보다 수익을 추구 / 투자자금의 대부분을 주식, 옵션 등의 위험자산에...	[('DB하이텍', [1, [11950.0, 'long_candle', 2.16, ...	[[('KBSTAR 200IT', [5, [825.0, 0.67, 1.21, 0.01... 2020.12.26
1	50	1개월 ~ 6개월	24	남	0	IT	널리 알려진 금융투자 상품(주식, 채권 및 펀드 등)의 구조 및 위험을 일정 부분 ...	투자원금의 보전보다 수익을 추구 / 투자자금의 대부분을 주식, 옵션 등의 위험자산에...	[('DB하이텍', [1, [22950.0, 'long_candle', 2.16, ...	[[('KBSTAR 200IT', [3, [14825.0, 0.67, 1.21, 0.... 2020.12.26
2	30	1개월 ~ 6개월	24	남	0	유통	널리 알려진 금융투자 상품(주식, 채권 및 펀드 등)의 구조 및 위험을 일정 부분 ...	투자원금의 손실 위험을 최소화하고, 안정적인 투자를 목표 / 예금·적금보다 높은 수...	[('DB하이텍', [1, [42950.0, 'long_candle', 2.16, ...	[[('KBSTAR 200경기소비재', [4, [10015.0, 0.59, 1.21,... 2020.12.26
3	30	1개월 ~ 6개월	26	남	65	의약	금융투자상품에 투자해 본 경험이 없음	투자원금의 보전보다 수익을 추구 / 투자자금의 대부분을 주식, 옵션 등의 위험자산에...	[('DB하이텍', [1, [15950.0, 'long_candle', 2.16, ...	[[('KBSTAR 헬스케어', [2, [18395.0, 0.56, 1.21, 0.0... 2020.12.26
4	500	6개월 ~ 1년	25	남	0	화학	파생상품을 포함한 대부분의 금융투자상품의 구조 및 위험을 이해하고 있음	투자원금의 보전보다 수익을 추구 / 투자자금의 대부분을 주식, 옵션 등의 위험자산에...	[('DB하이텍', [5, [4250.0, 'long_candle', 2.16, 0... [('TIGER 단기선진하이일드(합성 H)', [3, [12045.0, 0.1100... [('KODEX 에너지화학', [18, [16255.0, 0.81, 1.21, 0.... 2020.12.26		

Columns: 투자 금액(만 원)/ 투자 기간 / 나이/ 성별/ 월 정기 수입(만 원)/ 금융지식수준/ 위험추구성향/ 주식/ 채권/ ETF

현재 주가를 반영한 피드백과 수익률 추적 가능

CSV userDB.csv

투자 금액 (만 원)	투자 기 간	나 이	성 별	월 입(만 원)	정기 수 입	관심산 업분야	금융자산수준	위험추구성향	주식	채권	ETF	날짜	현재 수익률 (%)	수익률 기준 날짜
0	100	1주 ~ 1개월	24	남	0	IT	널리 알려진 금융투자 상품(주식, 채권 및 펀드 등)의 구조 및 위험을 깊이 있게 ...	투자원금의 보전보다 수익을 추구 / 투자자금의 대부분을 주식, 옵션 등의 위험자산에...	[('DB하이텍', [1, [11950.0, 'long_candle', 2.16, ...	[]	[('KBSTAR 200IT', [5, [825.0, 0.67, 1.21, 0.01... 2020.12.26	0.51	2020-12-24	
1	50	1개월 ~ 6개월	24	남	0	IT	널리 알려진 금융투자 상품(주식, 채권 및 펀드 등)의 구조 및 위험을 일정 부분 ...	투자원금의 보전보다 수익을 추구 / 투자자금의 대부분을 주식, 옵션 등의 위험자산에...	[('DB하이텍', [1, [22950.0, 'long_candle', 2.16, ...	[]	[('KBSTAR 200IT', [3, [14825.0, 0.67, 1.21, 0.... 2020.12.26	3.93	2020-12-24	
2	30	1개월 ~ 6개월	24	남	0	유통	널리 알려진 금융투자 상품(주식, 채권 및 펀드 등)의 구조 및 위험을 일정 부분 ...	투자원금의 손실 위험을 최소화하고, 안정적인 투자를 목표 / 예금·적금보다 높은 수...	[('DB하이텍', [1, [42950.0, 'long_candle', 2.16, ...	[]	[('KBSTAR 200경기소비재', [4, [10015.0, 0.59, 1.21, ... 2020.12.26	10.85	2020-12-24	
3	30	1개월 ~ 6개월	26	남	65	의약	금융투자상품에 투자해 본 경험이 없음	투자원금의 보전보다 수익을 추구 / 투자자금의 대부분을 주식, 옵션 등의 위험자산에...	[('DB하이텍', [1, [15950.0, 'long_candle', 2.16, ...	[]	[('KBSTAR 헬스케어', [2, [18395.0, 0.5, 1.21, 0.0... 2020.12.26	2.51	2020-12-24	
4	500	6개월 ~ 1년	25	남	0	화학	파생상품을 포함한 대부분의 금융투자상품의 구조 및 위험을 이해하고 있음	투자원금의 보전보다 수익을 추구 / 투자자금의 대부분을 주식, 옵션 등의 위험자산에...	[('DB하이텍', [5, [4250.0, 'long_candle', 2.16, 0... [('TIGER 단기선진하이일드(합성 H)', [3, [12045.0, 0.1100... [('KODEX 에너지화학', [18, [16255.0, 0.11, 1.21, 0.... 2020.12.26	0.89	2020-12-24			



Class Portfolio

def calculate_return()

def stock_return()
def etf_return()
def bond_return()

def get_latest_price()

현재 주가를 반영한 피드백과 수익률 추적 가능



Class Portfolio

```
def  
calculate_return()  
{  
    def stock_return()  
    def etf_return()  
    def bond_return()  
  
    {  
        def  
        get_latest_price()
```

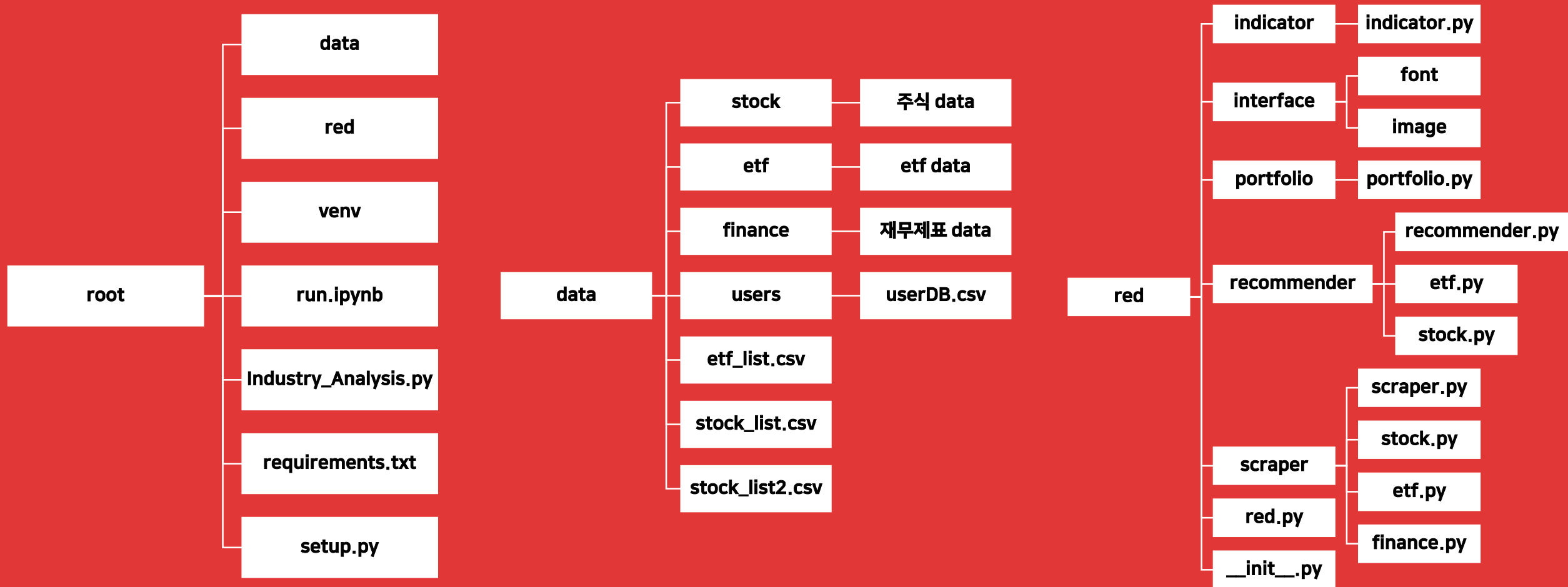
```
class Portfolio:  
    def __init__(self, path=os.getcwd()):  
        self.user_path = path + "/data/users/userDB.csv"  
        self.data_path = path + "/data"  
        self.latest_date = datetime.today()  
        if not os.path.isfile(self.user_path):  
            print('"userDB.csv"이 없습니다. ')  
            return  
  
    def calculate_return(self):  
        userDB = pd.read_csv(self.user_path, encoding="cp949", index_col=0)  
  
        for index, user in userDB.iterrows():  
            buy_price_lst = []  
            sell_price_lst = []  
            self.stock_return(user, buy_price_lst, sell_price_lst)  
            self.etf_return(user, buy_price_lst, sell_price_lst)  
            self.bond_return(user, buy_price_lst, sell_price_lst)  
            current_return = (sum(sell_price_lst) / sum(buy_price_lst) * 100 - 100).round(2)  
            userDB.loc[index, "현재 수익률(%)"] = current_return  
            latest_date = datetime.strptime(str(self.latest_date), "%Y%m%d").strftime("%Y-%m-%d")  
            userDB.loc[index, "수익률 기준 날짜"] = latest_date  
            userDB.to_csv(self.user_path, encoding="cp949")  
  
    def stock_return(self, user, buy_price_lst, sell_price_lst):  
        stock_lst = ast.literal_eval(user.주식)  
        if len(stock_lst) == 0:  
            return  
        for i in range(len(stock_lst)):  
            stock_name = stock_lst[i][0]  
            latest_price = self.get_latest_price("stock", stock_name)  
            sell_price = latest_price * stock_lst[i][1][0]  
            buy_price = stock_lst[i][1][1][0] * stock_lst[i][1][0]  
            sell_price_lst.append(sell_price)  
            buy_price_lst.append(buy_price)
```

```
    def etf_return(self, user, buy_price_lst, sell_price_lst):  
        etf_lst = ast.literal_eval(user.ETF)  
        if len(etf_lst) == 0:  
            return  
        for i in range(len(etf_lst)):  
            etf_name = etf_lst[i][0]  
            latest_price = self.get_latest_price("etf", etf_name)  
            sell_price = latest_price * etf_lst[i][1][0]  
            buy_price = etf_lst[i][1][1][0] * etf_lst[i][1][0]  
            sell_price_lst.append(sell_price)  
            buy_price_lst.append(buy_price)  
  
    def bond_return(self, user, buy_price_lst, sell_price_lst):  
        bond_lst = ast.literal_eval(user.채권)  
        if len(bond_lst) == 0:  
            return  
        for i in range(len(bond_lst)):  
            bond_name = bond_lst[i][0]  
            latest_price = self.get_latest_price("etf", bond_name)  
            sell_price = latest_price * bond_lst[i][1][0]  
            buy_price = bond_lst[i][1][1][0] * bond_lst[i][1][0]  
            sell_price_lst.append(sell_price)  
            buy_price_lst.append(buy_price)  
  
    def get_latest_price(self, category, name):  
        file_path = os.path.join(self.data_path, category, name)  
        file_path = file_path + ".csv"  
        df = pd.read_csv(file_path, encoding="cp949", index_col=0)  
        latest_price = df.iloc[-1, :]["close price"]  
        self.latest_date = df.index[-1]  
        return latest_price
```


Project Structure

패키지 구조를 활용한 프로젝트 진행

*패키지: 모듈을 디렉토리 형식으로 구조화한 것



RED의 추천 종목은?



RED().run()

"userDB.csv" 파일을 찾을 수 없습니다. 먼저 userDB를 생성하겠습니다.

투자 금액(만 원)

투자 기간

나이 (만)

성별

월 정기 수입(만 원)

관심산업분야

금융지식수준

위험추구성향

데이터 업데이트

투자 시작

뒤로 가기