# Chicken Stock

MultiCampus Final Project

# CONTENTS

01

# 프로젝트 배경

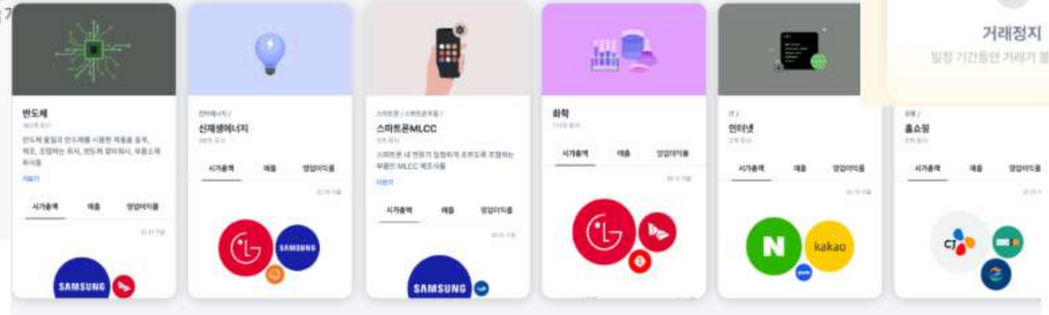
# 프로젝트 기획 배경


## 어려웠던 투자 시작부터 쉽게

투자를 시작하는 사람도
이해하기 쉬운 용어,
설명이 필요 없는 직관적 화면 구성.


## 주요 정보를 쉽게 만날 수 있도록

투자할 또는 투자한 회사의 최신 기사부터,
재무제표와 주요사업 현황을 보기 쉽게.





## 투자한 회사의 변화, 놓치지 않도록

투자 중인 회사의 주의해야할 정보,
놓치지 않도록 알려드려요.

# 프로젝트 기획 배경

## 쇼핑하듯 주식 사고, 게임하듯 적금 든다…쉬운 금융에 빠진

입력 2021-07-03 06:30:18 수정 2021.07.03 06:30:18 이태규 기자

[토요워치-MZ가 쏘아올린 '쉬운 금융' 바람]
토스증권, 제품 검색하면 제조사 주가정보 제공
카카오뱅크, 만기 1년 상식 깬 '26주 적금' 인기
하나銀 용돈관리앱 등 기존 금융권도 발빠른 대응
신한라이프-맥주·삼성생명-생수 제품 출시 등
소비재 기업과 협업 통해 친근한 이미지 구축

## 빚투에 빠지는 청년세대…증권사 신규대출액 38.7조원

증시 불안에 청년층 개인신용 우려↑

황대영 기자 | 입력 2021.10.04 16:00 | 댓글 0

## 인류학이 물었다, 개미 투자자는 왜 '개미지옥'에 빠져드나

양지호 기자

입력 2021.09.11 03:00

# PROBLEM

너무 복잡한 증권 웹 페이지
매일 쏟아지는 수많은 데이터
이해하기 어려운 시스템
내 위험 감내 수준이 반영되지 않음

# SOLUTION

투자자의 성향에 맞는
주식 포트폴리오 추천 시스템을
만들어보자 !
주요 정보를 쉽게 접할 수 있게
UI를 구축하자 !

# 프로젝트 주제


**" 주식 종가 예측 모델, 포트폴리오 추천 모델을 활용하여**
**고객의 투자 성향에 따른 투자 포트폴리오 추천 서비스 제공 "**

# 프로젝트 세부 목표

1. 사용자별 투자 성향 조사
2. 주식 종가 예측
3. 주식 포트폴리오 추천
4. 추천 기업 최근 일주일 뉴스기사 제공

# 팀 구성

| 팀원 명 | 역할 |
| --- | --- |
| 이재웅<br>(팀 리더) | LSTM 모델 구축, 포트폴리오 추천 모델 구축<br>데이터 전처리, 데이터 시각화 |
| 서미오<br>(팀원) | 웹 크롤링, SQLite3로 DB연동,<br>데이터 스케쥴링, 데이터 시각화 |
| 송건룡<br>(팀원) | LSTM 모델 구축, ARIMA 모델 구축, fbprophet 모델 구축 |
| 김병현<br>(팀원) | 웹 디자인, 웹 서버 구축 |
| 유준웅<br>(팀원) | 데이터 취합 |

02

# 프로젝트 시연

03

# 프로젝트 수행 방법

# 프로젝트 수행 일정


프로젝트 시작
- MindMap으로 주제 선정
- 프로젝트 산출물 제작
9/1
데이터 수집
- API
- 웹 크롤링
- 스케쥴링 설정
9/12
데이터 전처리
- 데이터 형식 통일
- 데이터 취합
- DB 구축
9/19
9/26
- 주식 종가 예측 모델링
- 포트폴리오 추천 모델링
데이터 분석 및 모델링
10/2
- pychart
- matplotlib
- Django
시각화 및 웹 구현
10/7
- 단위테스트
- 통합테스트
- 원격서버 연결
- AWS 구축
수정 및 보완

# 개발환경 및 도구

| 개발환경 및 도구 | |
|---|---|
| 언어 | Python, SQL |
| 통합 개발 환경 | Pycharm, Jupyter Notebook, Colab |
| 분석 라이브러리 | Pandas, Numpy, Sikit-Learn, ARIMA, FBprophet, Tensorflow, SciPy |
| 시각화 | Matplotlib, Plotly, Pychart |
| 웹 서버 | Django |
| 웹 클라이언트 | HTML5, JS, Ajax, CSS |
| 시스템 운영 환경 | AWS |
| 데이터베이스 | SQLite |
| 협업 도구 | Github, Zoom, Google Docs |

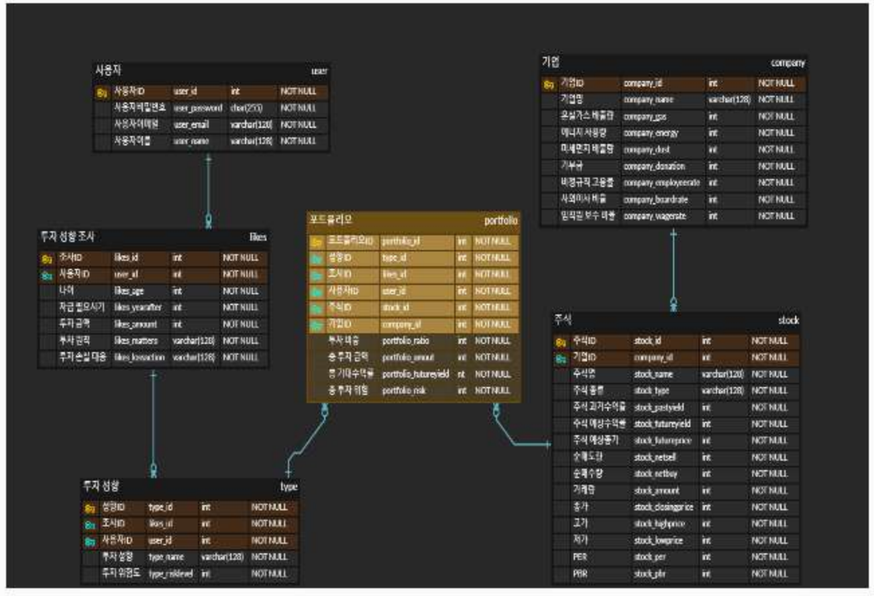
# 프로젝트 산출물

# WBS

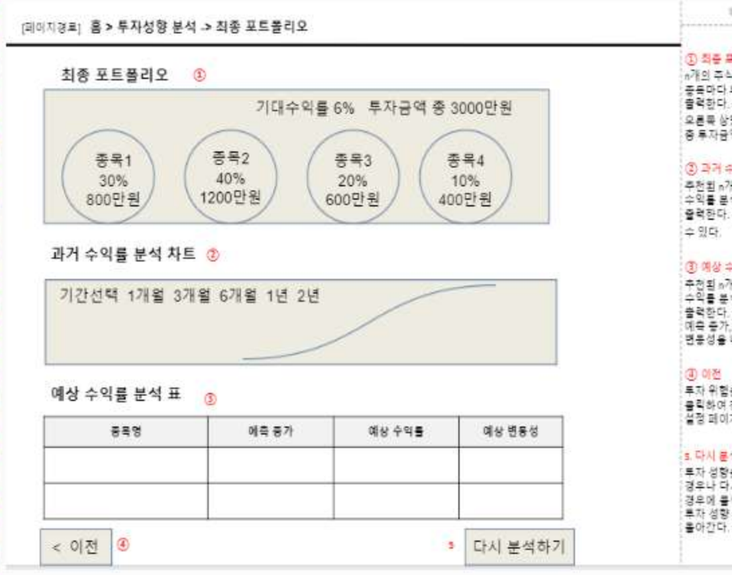
| PROJECT TITLE | 주식 포트폴리오 추천 서비스 | TEAM NAME | Chicken Stock |
|---|---|---|---|
| PROJECT DATE | 2021-08-31 ~ 2021-10-08 | TEAM MEMBER | 팀장 : 이재용<br>팀원 : 김명현, 서미오, 송 |

| WBS NUMBER | TASK TITLE | TASK OWNER | START DATE | DUE DATE | DURATION | PCT OF TASK COMPLETE | 2021-08-31 ~ 2021-09-03 (WEEK 1) M | T | W | R | F | 2021-09-06 ~ (WEE) M | T | W |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 프로젝트 기획 | | | | | | | | | | | | | |
| 1.1 | 주제 선정 | 공통 | 2021-08-29 | 2021-09-01 | 4 | 100% | | | | | | | | |
| 1.1.1 | 기획안 작성 | 재윤 | 2021-09-01 | 2021-09-02 | 2 | 100% | | | | | | | | |
| 1.2 | 요구사항 정의서 작성 | 공통 | 2021-09-02 | 2021-09-02 | 1 | 100% | | | | | | | | |
| 1.3 | 와이어프레임 작성 | 준윤 | 2021-09-02 | 2021-09-02 | 1 | 100% | | | | | | | | |
| 1.4 | 화면 정의서 작성 | 공통 | 2021-09-02 | 2021-09-02 | 1 | 100% | | | | | | | | |
| 1.5 | ERD 작성 | 미오 | 2021-09-02 | 2021-09-02 | 1 | 100% | | | | | | | | |
| 1.6 | 인터페이스 정의서 작성 | 미오 | 2021-09-02 | 2021-09-02 | 1 | 100% | | | | | | | | |
| | | | | | | | | | | | | | | |
| 2 | 데이터 수집 | | | | | | | | | | | | | |
| 2.1 | 데이터 수집 - API | 건률 | 2021-09-02 | 2021-09-03 | 5 | 95% | | | | | | | | |
| 2.2 | 데이터 수집 - Scraping | 미오 | 2021-09-20 | 2021-09-24 | 5 | 98% | | | | | | | | |
| 2.3 | 데이터 수집 스케줄링 | 미오 | 2021-09-27 | 2021-09-29 | 3 | 98% | | | | | | | | |
| | | | | | | | | | | | | | | |
| 3 | 데이터 분석 | | | | | | | | | | | | | |
| 3.1 | 데이터 전처리 | 재윤 | 2021-09-06 | 2021-09-06 | 1 | 100% | | | | | | | | |
| 3.2 | 추가 데이터 정제 및 적용 | 공통 | 2021-09-06 | 2021-09-06 | 2 | 100% | | | | | | | | |
| 3.3 | 주식 종가 예측 모델링 | 재윤 | 2021-09-07 | 2021-09-07 | 3 | 100% | | | | | | | | |
| 3.4 | 포트폴리오 모델링 | 재윤 | 2021-10-05 | 2021-10-06 | 5 | 90% | | | | | | | | |
| 3.5 | 데이터 시각화 | 미오 | 2021-10-05 | | 5 | 70% | | | | | | | | |
| 3.6 | | | | | | | | | | | | | | |
| 4 | 프론트 개발 | | | | | | | | | | | | | |
| 4.1 | 디자인 시안 컨펌 | 공통 | 2021-09-07 | 2021-09-07 | | 0% | | | | | | | | |
| 4.1.1 | 부트스트랩 적용 | 보현 | 2021-09-08 | 2021-09-08 | | 0% | | | | | | | | |

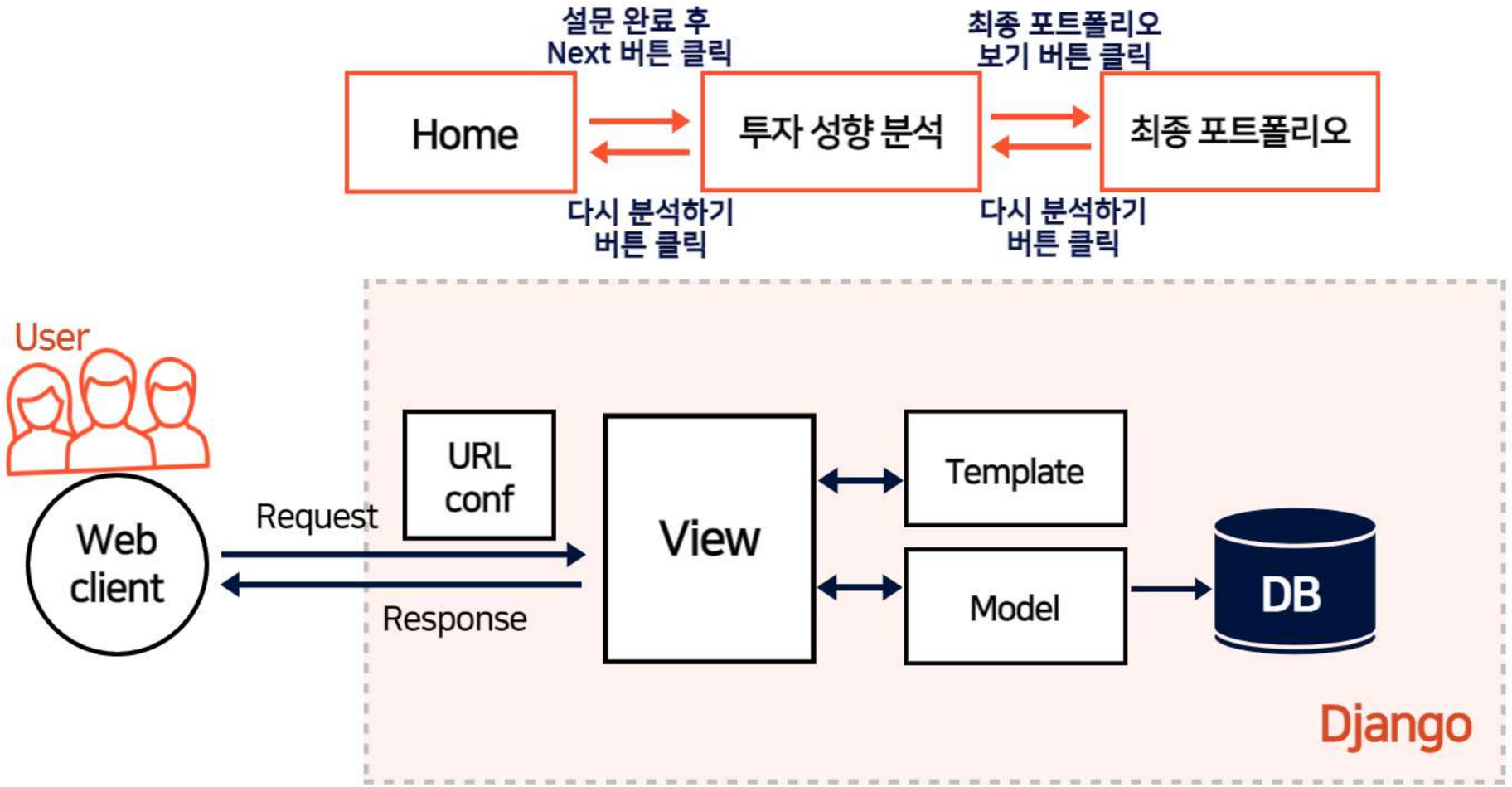
# ERD




# 화면 정의서



# MVT 패턴


설문 완료 후
Next 버튼 클릭
최종 포트폴리오
보기 버튼 클릭
Home
투자 성향 분석
최종 포트폴리오
다시 분석하기
버튼 클릭
다시 분석하기
버튼 클릭
User
Web
client
Request
Response
URL
conf
View
Template
Model
DB
Django

# 프로젝트 프로세스

## 데이터 수집

- KRX, INESTING.COM 에서 수집한 데이터
- 모델링 및 웹 구현을 위한 크롤링 데이터

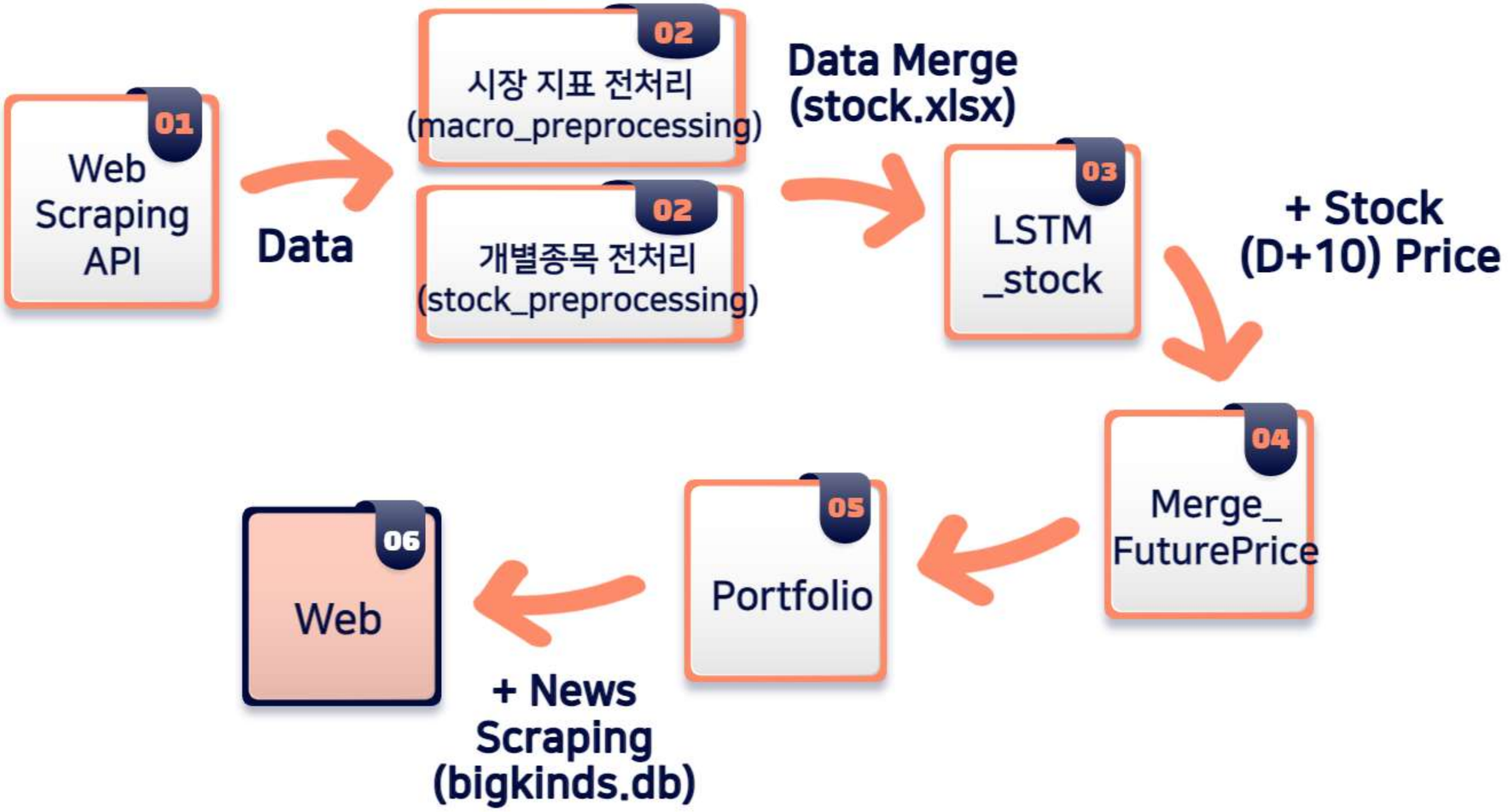
## 모델 선택 및 구축

- LSTM, ARIMA, fbprophet 등 다양한 알고리즘
- 각 모델별 결과 값 비교

## 웹 구현

- HTML, CSS, Javascript 등 다양한 언어
- Django 환경
- 원격서버, AWS 활용

# 데이터 수집 프로세스


01
Web Scraping API
Data
02
시장 지표 전처리 (macro_preprocessing)
02
개별종목 전처리 (stock_preprocessing)
Data Merge (stock.xlsx)
03
LSTM _stock
+ Stock (D+10) Price
04
Merge_ FuturePrice
05
Portfolio
06
Web
+ News Scraping (bigkinds.db)

# 데이터 수집 - 기간 및 대상 기업

## 데이터 수집기간

2011-09-01 ~ 2021-10-07 ~ schedule로 매일 업데이트

## 대상 기업

포트폴리오 분산을 위해 산업군별 대표주를 선정
-> 25개 산업군 50개 기업 선정

삼성전자, SK하이닉스, LG화학, LG전자, LG이노텍, 삼성에스디에스, 삼성전기, 삼성생명, 삼성화재, SK텔레콤,
KT, 현대건설, 삼성엔지니어링, 대한항공, 현대차, 기아, 오리온, CJ제일제당, 오뚜기, 미래에셋대우, 한국금융지주, NH투자증권,
LG생활건강, 아모레퍼시픽, 아모레G, 강원랜드, 호텔신라, KB금융, 신한지주, 하나금융지주, 롯데쇼핑, 이마트, 신세계, GS리테일,
NAVER, 카카오, CJ ENM, 스튜디오드래곤, 삼성바이오로직스, 셀트리온, 한미약품, 엔씨소프트, 넷마블, 한화솔루션, LS, POSCO,
고려아연, S-Oil, SK이노베이션, HMM

# 데이터 수집 - 지표 선정

## 거시 경제 지표

NASDAQ (세계 경제)
S&P 500 (세계 경제)
CBOE VIX (세계 경제 변동성)
원 달러 환율 (세계 경제와 한국 경제의 상관관계)
2년만기 미국채 선물 (금융정책 및 시장 전망)
10년만기 미국채 선물 (금융정책 및 시장 전망)

+

## 개별주 관련 지표

거래량 (시장 참여자의 반응, 모멘텀)
PER (주가수익비율, 재무정보)
PBR (주가순자산비율, 재무정보)
ATR(Average True Range, 변동성)
투자자별 순매수

# 데이터 수집 - API

## (1) FinanaceDataReader

```
class UpdateDB:
    def __init__(self):
        self.yday = (date.today() - timedelta(1)).strftime('%Y%m%d')  # 1일 전
        self.bfyday = (date.today() - timedelta(2)).strftime('%Y%m%d')  # 2일 전
        self.options = Options()
        self.options.add_argument('headless')
        self.contents = []

    def getINVEST(self):  # Investing.com 데이터
        self.sp500 = fdr.DataReader('US500', self.yday, self.yday)[['Close']]
        self.cboe = fdr.DataReader('VIX', self.yday, self.yday)[['Close']]
        self.rate = fdr.DataReader('USD/KRW', self.yday, self.yday)[['Close']]
        # 거시경제지표 중 S&P500은 데이터 업데이트가 늦음
        df_index = pd.concat([self.sp500, self.cboe, self.rate], axis=1).reset_index()
        df_index.columns = ['date', 'sp', 'cboe', 'exchangerate']
        self.df_index = df_index.set_index('date')
        print('getINVEST complete.')
        print(self.df_index)
        return self.df_index
```

## (2) pykrx

```
class UpdateDB:
    def __init__(self):
        self.yday = (date.today() - timedelta(1)).strftime('%Y%m%d')
        self.bfyday = (date.today() - timedelta(2)).strftime('%Y%m%d')
        self.bffyday = (date.today() - timedelta(3)).strftime('%Y%m%d')

    def getKRX(self, code):
        self.stockname = stock.get_market_ticker_name(code)
        def getATR():
            df_a = stock.get_market_ohlcv_by_date(self.bffyday, self.yday, code)
            df_atr = df_a.rename(index={'날짜': 'Date'}).drop(columns="시가", axis=1)
            df_atr.columns = ['high', 'low', 'close', 'volume']
            df_atr["atr"] = ""
            lst = []
            atr = []
            for i in range(len(df_atr) - 1):
                a = df_atr.iloc[i, 0] - df_atr.iloc[i, 1]  # 고가-저가
                b = df_atr.iloc[i, 0] - df_atr.iloc[i + 1, 2]  # 고가-전날종가
                c = df_atr.iloc[i, 1] - df_atr.iloc[i + 1, 2]  # 저가-전날종가
                lst = [abs(a), abs(b), abs(c)]
                atr.append(max(lst))
            df_atr.iloc[1, 4] = atr
            df_atr = df_atr[1:]
            return df_atr[['close','volume','atr']]
        self.ohlcv = getATR()
        self.funda = stock.get_market_fundamental_by_date(self.yday, self.yday, code)[['PER', '
        self.value = stock.get_market_trading_value_by_date(self.yday, self.yday, code).drop(['

        self.df = pd.concat([self.ohlcv, self.funda, self.value], axis=1)
        self.df_krx = self.df.reset_index()
        self.df_krx.columns=['date','y','volume', 'atr', 'per', 'pbr', 'institution', 'corp', '
        self.df_krx = self.df_krx.set_index('date')
        return self.df_krx
```

# 데이터 수집 - 크롤링

## (1) 인베스팅닷컴 주요 지표 크롤링

```
    def crawlINVEST(self):  # Investing.com 크롤링
        self.getINVEST()
        self.urls = ['https://kr.investing.com/indices/nasdaq-composite-historical-data','htt
-bonds/us-2-yr-t-note-historical-data', 'https://kr.investing.com/rates-bonds/us-10-yr-t-note
        for url in self.urls:
            self.driver = webdriver.Chrome('C:₩chromedriver.exe', options=self.options)
            self.driver.get(url)
            self.soup = BeautifulSoup(self.driver.page_source, 'html.parser')
            try:
                self.index = self.soup.select('#last_last')
            except:
                self.driver.execute_script("window.scrollTo(0, 700)")
                self.index = self.soup.select('#curr_table > tbody > tr:nth-child(1) > td:nth
            self.contents.append(self.index[0].text.strip())
            del self.soup
            self.driver.quit()
        self.dateindex = pd.DatetimeIndex([(date.today() - timedelta(1))])  #datetime.today(
        df_crawl = pd.DataFrame([self.contents], index= self.dateindex, columns=['nasdaq','fu
set_index().rename(columns={'index': 'date'})
        self.df_crawl = df_crawl.set_index('date')
        print('crawlINVEST complete.')
        print(self.df_crawl)
        return self.df_crawl

    def mergeINVEST(self):
        self.crawlINVEST()
        self.df_invest = pd.merge(self.df_index, self.df_crawl, on='date')
        print(self.df_invest)
        return self.df_invest
```

## (2) 빅카인즈 뉴스기사 크롤링

```
class CRAWL:
    def __init__(self):
        self.options = Options()
        self.options.add_argument('headless')
        self.driver = webdriver.Chrome('C:₩chromedriver.exe', options=self.options)
    def getURL(self,url):
        # 빅카인즈 사이트 이동
        self.driver.implicitly_wait(2)
        self.driver.set_window_size(1300,800)
        self.driver.get(url)
        # 검색기간 1주일로 설정
        self.driver.find_element_by_xpath('//*[@id="collapse-step-1-body"]/div[2]/div
        self.driver.find_element_by_xpath('//*[@id="srch-tab1"]
    def getPage(self, url, querytxt):
        self.getURL(url)
        searchbox = self.driver.find_element_by_id("total-searc
        searchbox.send_keys(querytxt)
        searchbox.send_keys("₩n")  # 검색버튼 엔터
        self.driver.find_element_by_xpath('//*[@id="filterTab03
        self.driver.find_element_by_xpath('//*[@id="select2"]/o
        soup = BeautifulSoup(self.driver.page_source, 'html.par
        pageNum = soup.select('div.lastNum')[0].text
        self.totalPage  = int(pageNum)
        del soup
    def crawling(self, url, querytxt):
        self.getPage(url, querytxt)
        print('Total Pages : ', self.totalPage)
        curPage = 1  # 현재 페이지
        self.contents = []
        self.texts = []
        while curPage <= self.totalPage:
            # bs4 초기화
            soup = BeautifulSoup(self.driver.page_source, 'html.parser')
            # 기사 리스트
            articles = soup.select('div.news-inner')
            # 페이지 번호 출력
            print('Current Page : {}'.format(curPage))
            # 세부 데이터
            for article in articles:
                title = article.select_one('span.title-ellipsis').text.strip()
                press =  article.select_one('div.info > div > a.provider').text.strip()
                category = article.select_one('div.info > div > span.bullet-keyword').text.strip
                date = article.select_one('div.info > p.name').text.strip()
                self.contents.append([title, press, category, date])
            # 기사 전문
            for i in range(len(articles)):
                self.driver.find_elements_by_css_selector('span.title-ellipsis')[i].click()
                time.sleep(2)
                text = self.driver.find_elements_by_css_selector('div.news-view-body')[0].text
                self.texts.append(text.replace('₩n', ''))
                self.driver.find_element_by_xpath("//div[@id='news-detail-modal']/div/div/butto
                time.sleep(2)
            # 페이지 수 증가
            curPage += 1
            if curPage > self.totalPage:
                print('Crawling succeed')
                break
            # 페이지 이동 클릭
            self.driver.implicitly_wait(3)
            nextbtn = self.driver.find_element_by_xpath('//*[@id="news-results-tab"]/div[6]/div
            self.driver.execute_script("arguments[0].click();", nextbtn)
            # bs4 인스턴스 삭제
            del soup
            time.sleep(2)
ef saving(self, querytxt):
    # 브라우저 종료
    self.driver.close()
    self.df = pd.DataFrame(data=self.contents, columns =['title', 'press', 'category', 'dat
    self.df['text'] = self.texts
```

# 데이터 수집 - 스케쥴러

" 매일 오전 5:30에 데이터 업데이트 "

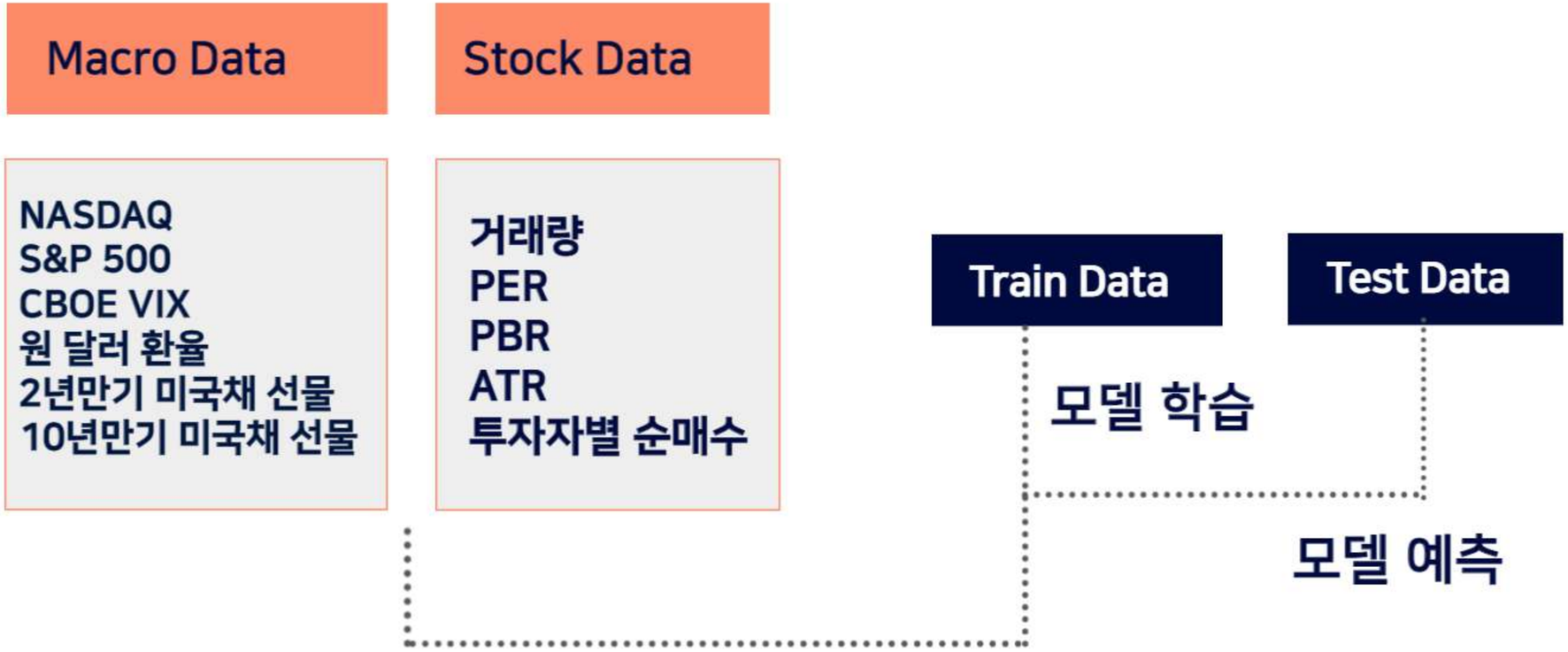
## (1) 주식 지표 데이터

```
    for code in codes:
        udb.getKRX(code)  # y, volume, per,
        udb.saving()  # db에 최종 저장
    udb.conn.close()


# 스크래핑 및 전처리 실행
schedule.every().day.at("05:32").do(stock)
# schedule.every(1).minutes.do(stock)
while True:
    schedule.run_pending()
    time.sleep(1)
```

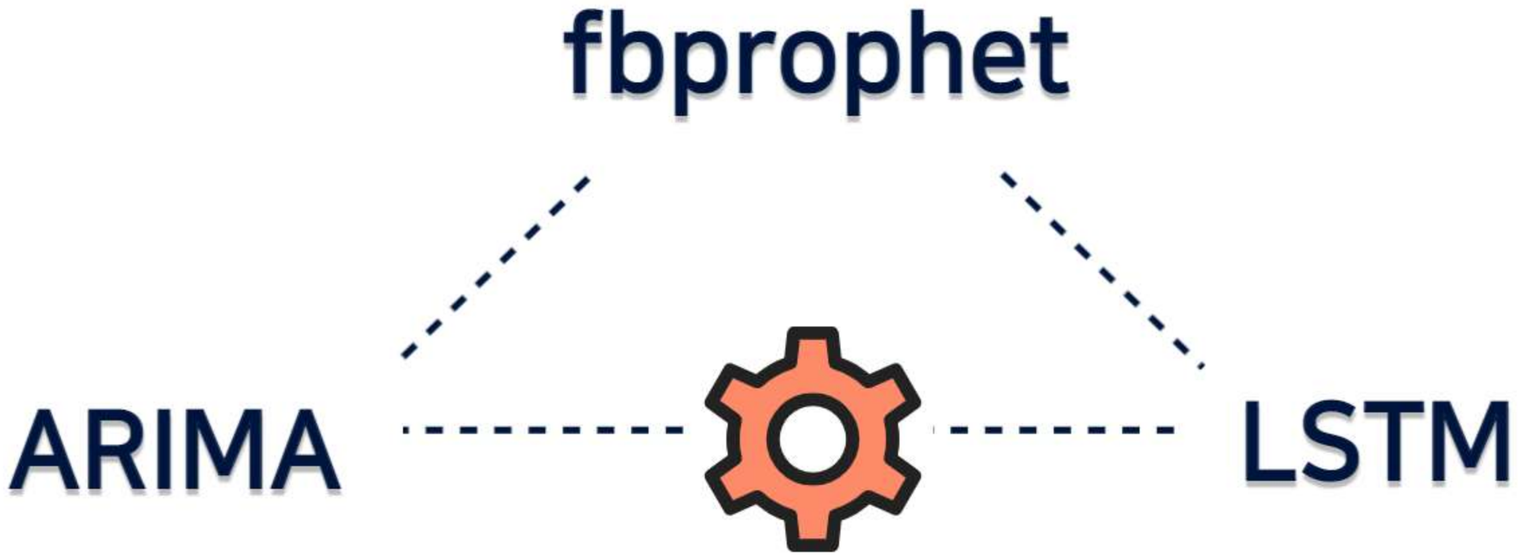
## (2) 빅카인즈 뉴스기사 데이터

```
    for stockname in stocknames:
        crawl = Bigkinds()
        crawl.crawling(url, stockname)
        crawl.saving(stockname)
    conn.close()


schedule.every().day.at("5:30").do(crawl)  # 매일 5:30에 동작
# schedule.every().monday.at("5:30").do(crawl)  # 매주 월요일
while True:
    schedule.run_pending()
    time.sleep(1)
```

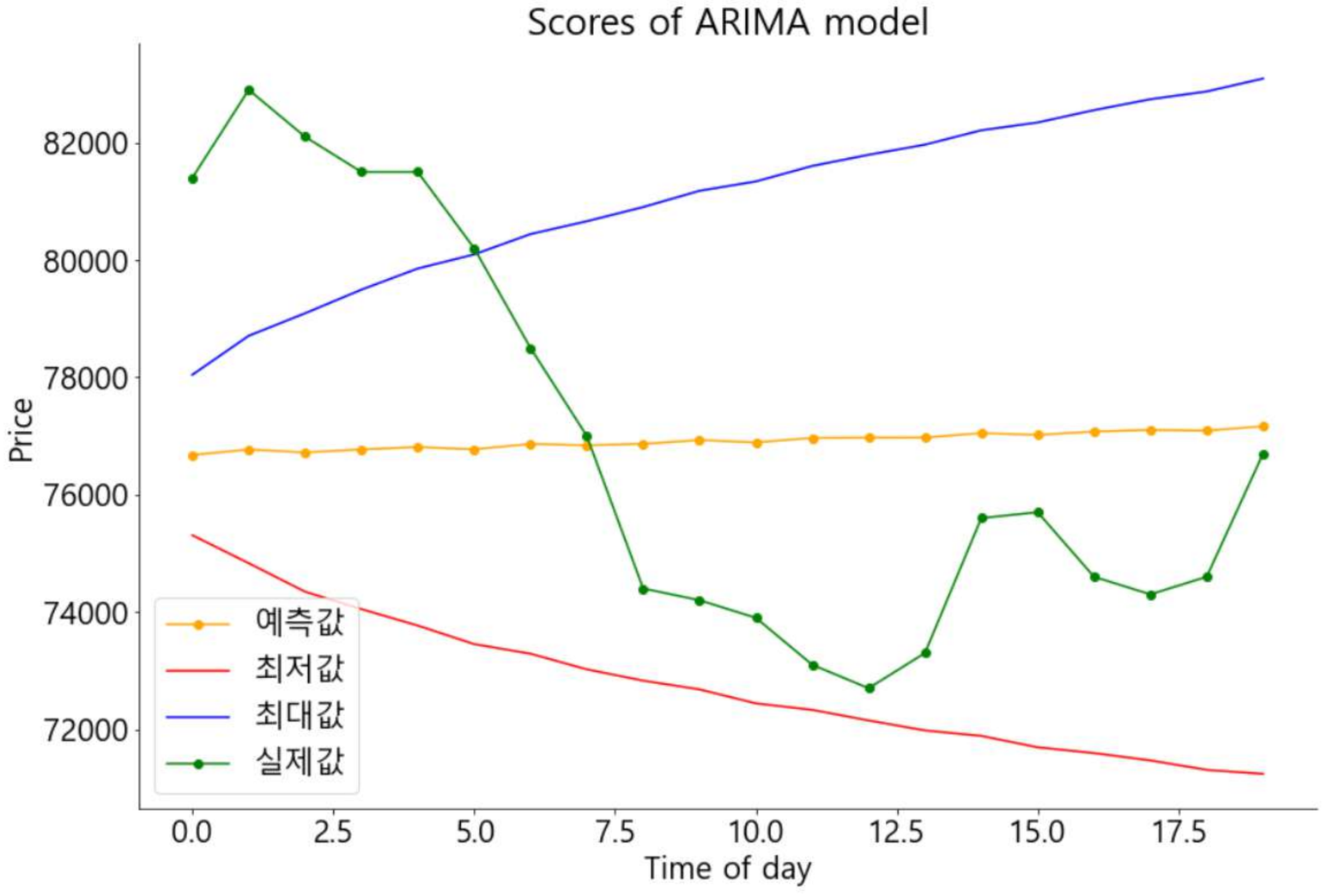
# 데이터 적용 프로세스


Macro Data
Stock Data
NASDAQ
S&P 500
CBOE VIX
원 달러 환율
2년만기 미국채 선물
10년만기 미국채 선물
거래량
PER
PBR
ATR
투자자별 순매수
Train Data
Test Data
모델 학습
모델 예측

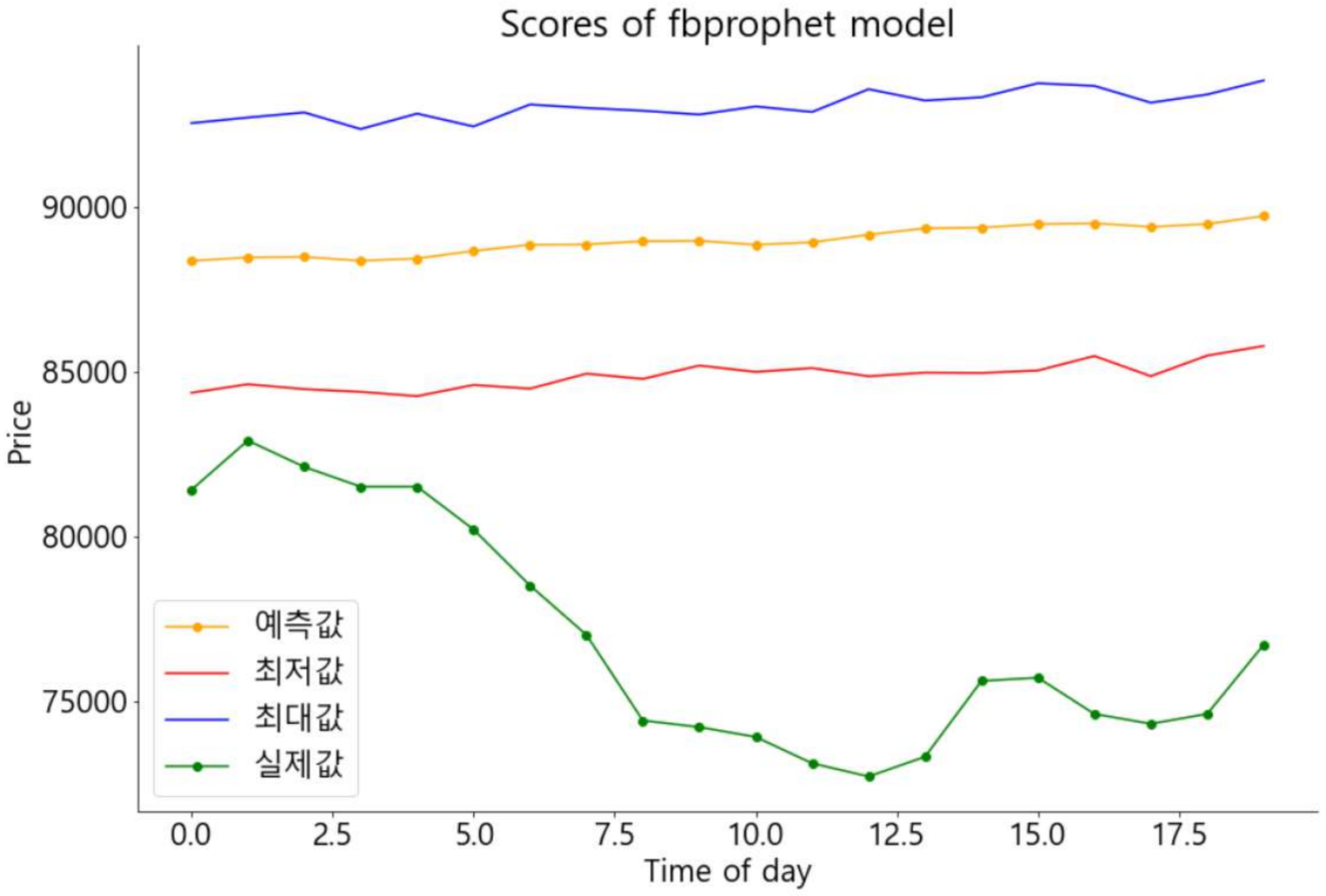
# 종가 예측 모델 선택



# 모델 선택 과정
# - ARIMA



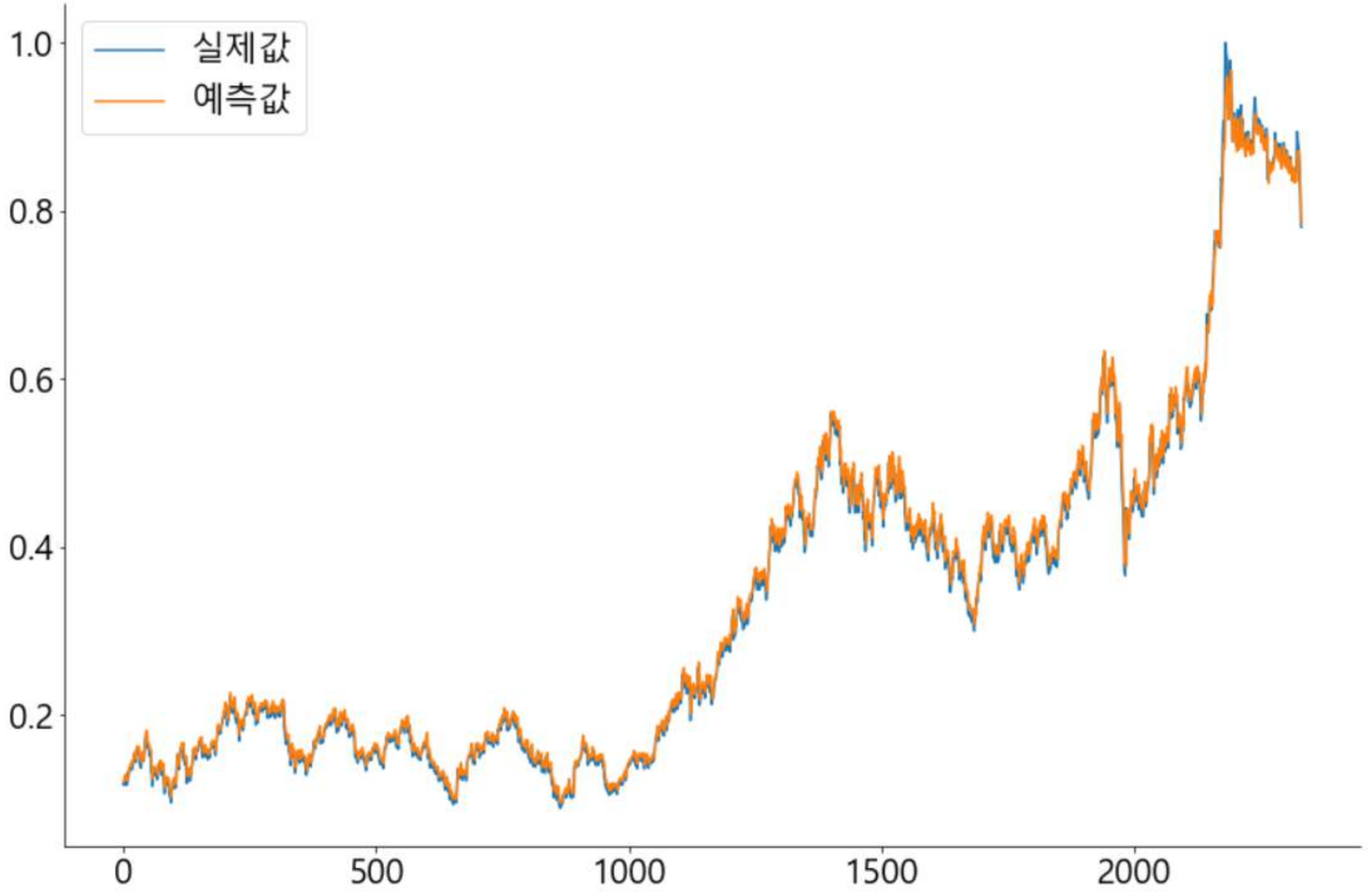
# 모델 선택 과정
# - fbprophet



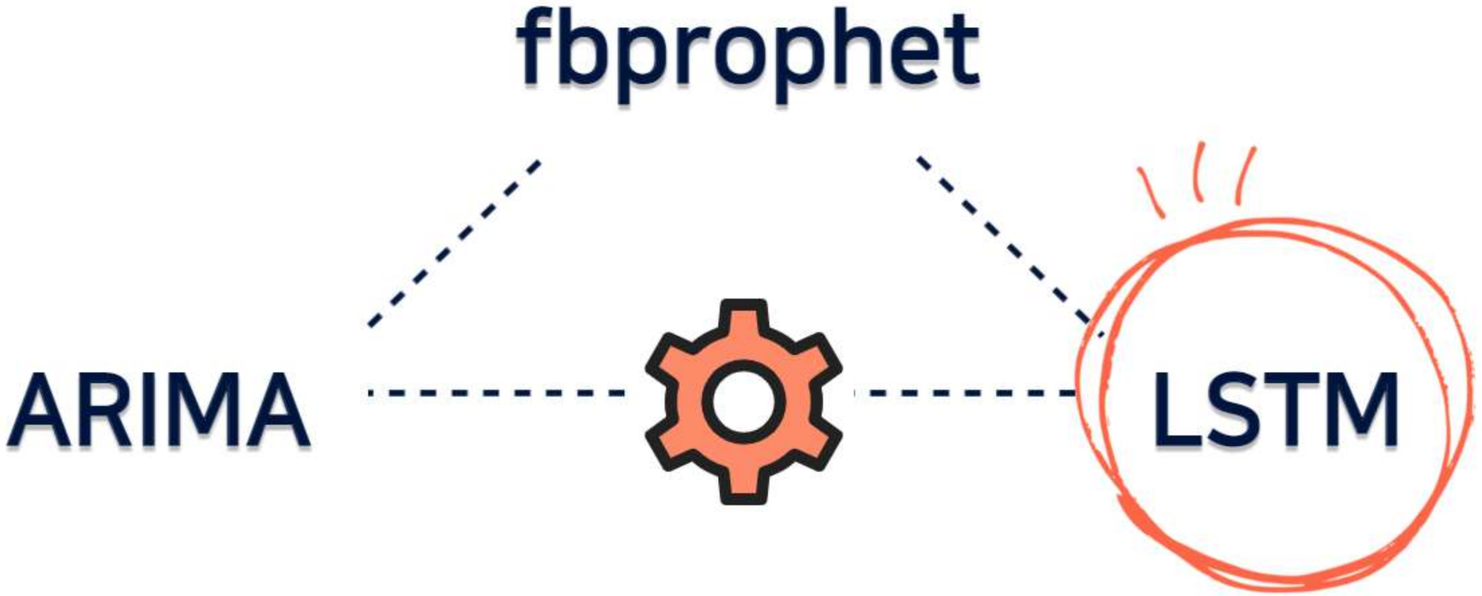
# 모델 선택 과정
# - LSTM

LSTM 모델 최대 스코어

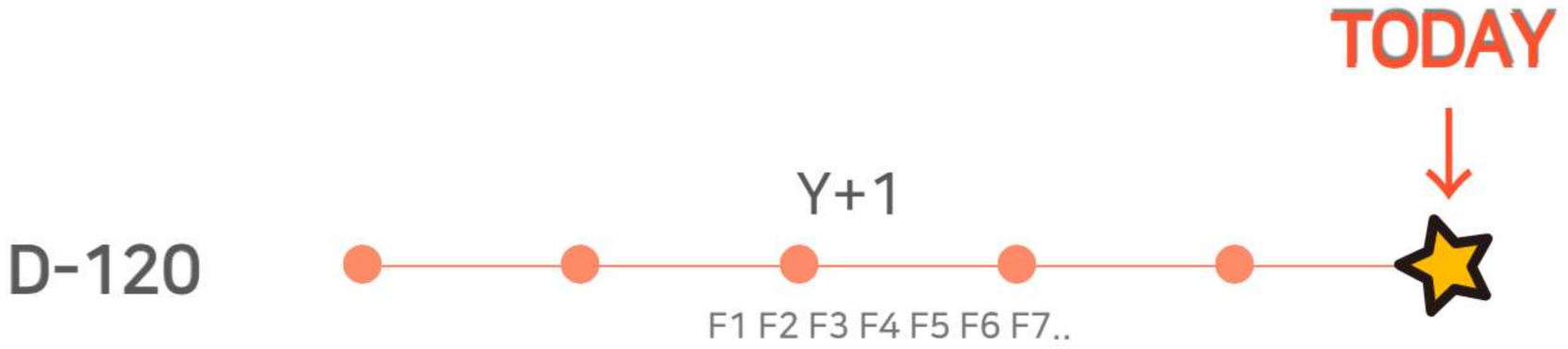
- RMSE : 0.015863938
- R2 : 0.9922155763



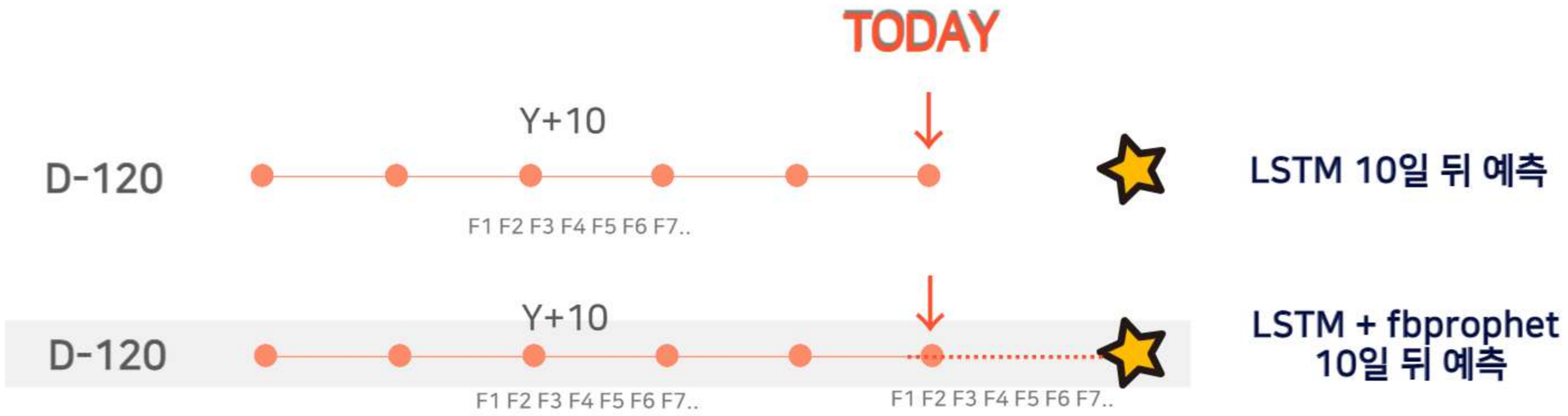
# 종가 예측 모델 선택



# LSTM 모델 설명

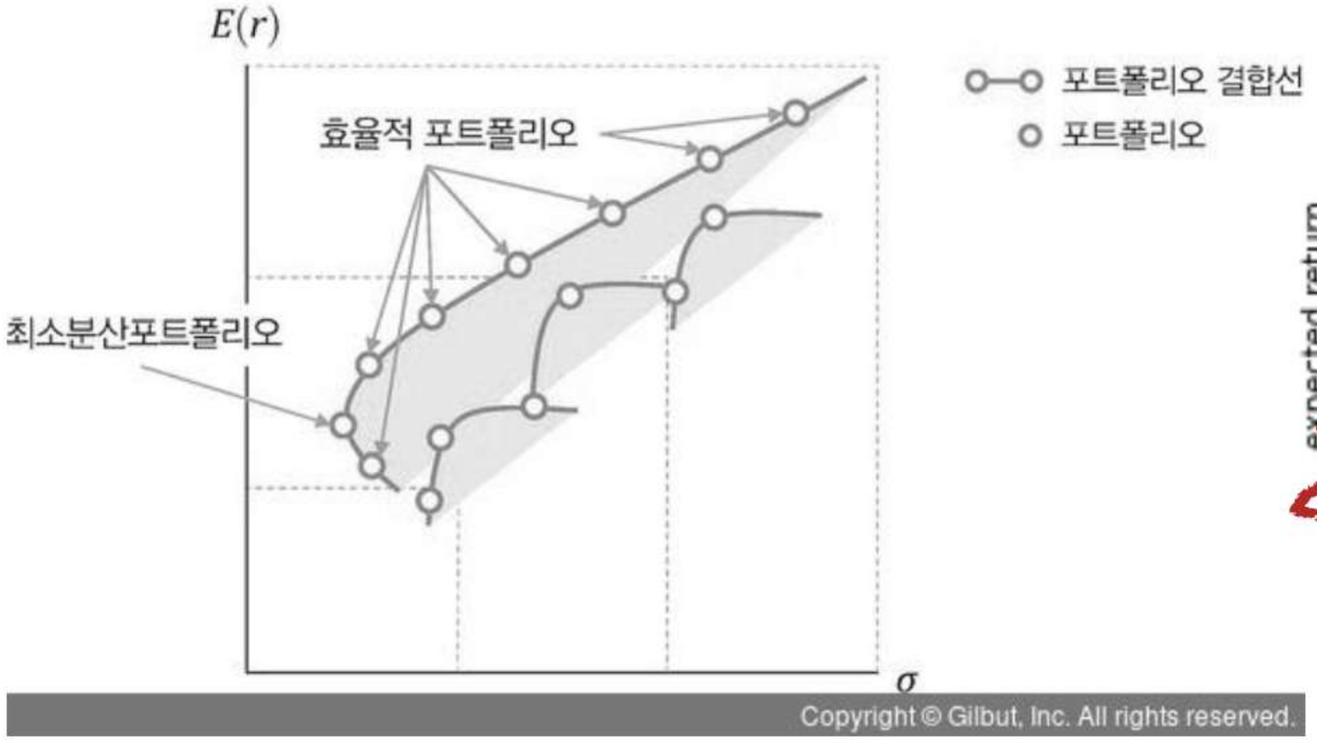


# LSTM 모델 설명

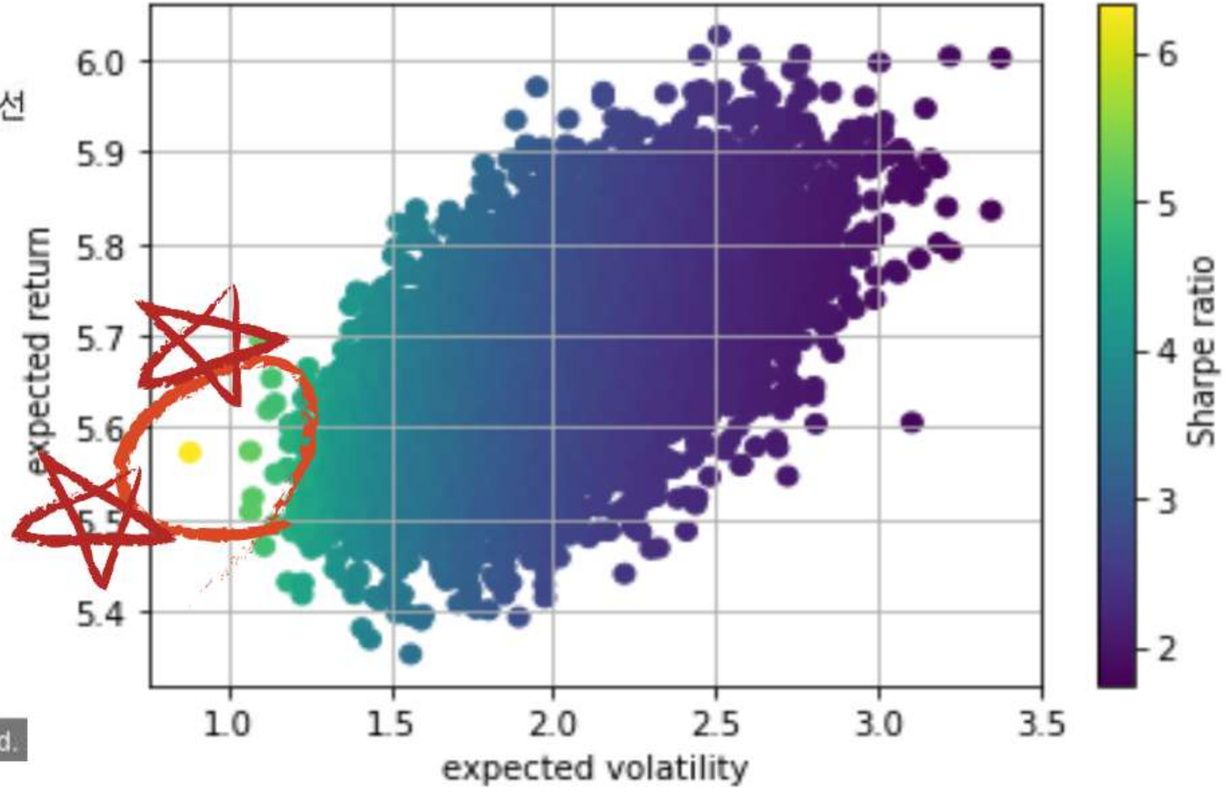



" 먼 미래를 예측할때 성능 감소가 적으며 feature를 더욱 정확하게 예측
하여 성능을 향상시킬 여지가 있다 "

# 포트폴리오 추천 - MVP 모델



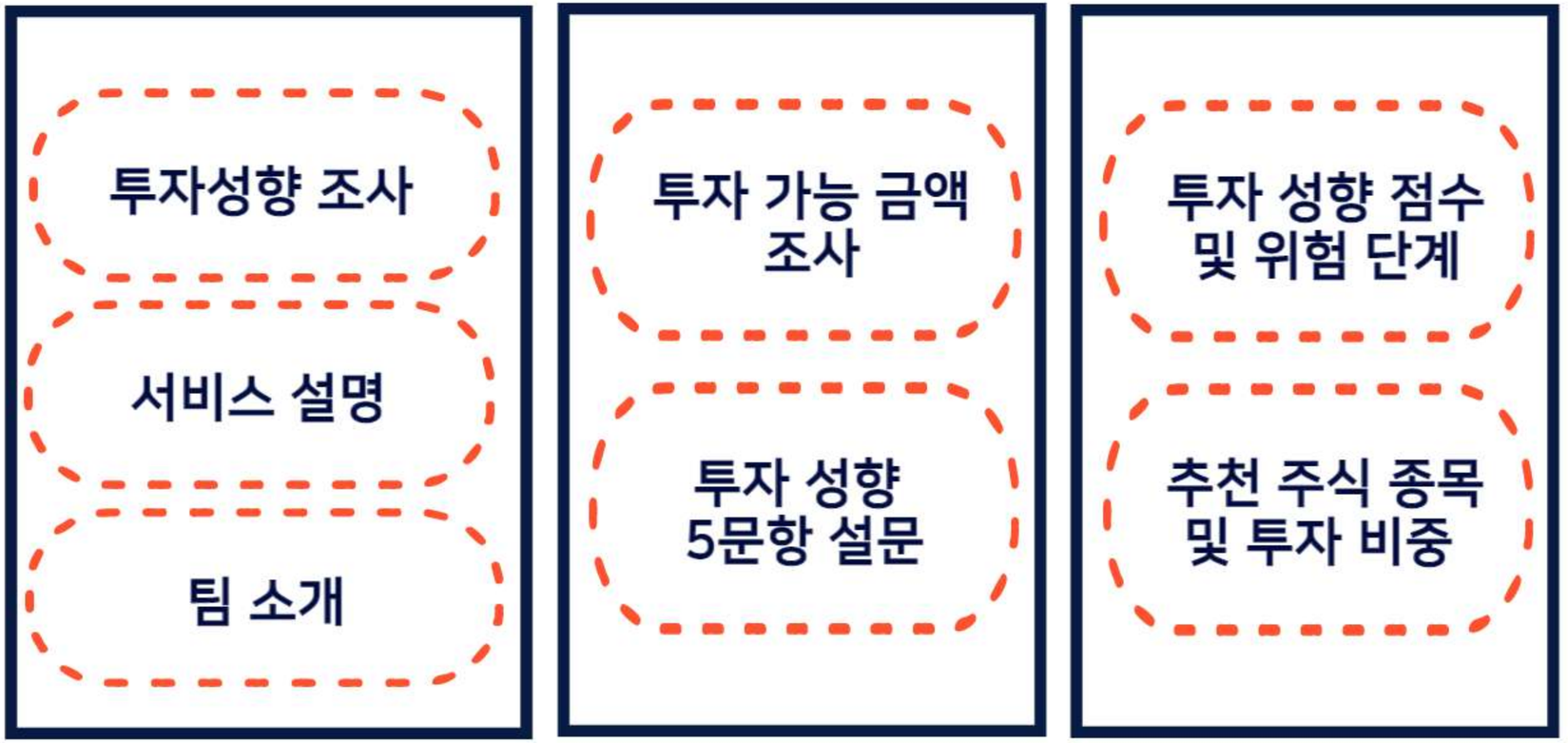


# Web Server


SERVER
- 점수 합산하여 사용자별 투자위험 수준 산출
- 투자위험 수준 포트폴리오 추천 모델에 전송
- 완성된 포트폴리오를 웹에 전송
WEB
설문1
설문2
설문3
설문4
설문5
설문6
포트폴리오
설문결과
투자금액,
투자위험
포트폴리오
PORTFOLIO
시각화 자료 제시

# 웹 구성


투자성향 조사
서비스 설명
팀 소개
투자 가능 금액 조사
투자 성향 5문항 설문
투자 성향 점수 및 위험 단계
추천 주식 종목 및 투자 비중

# 웹 연결
# - url 파라미터

```
var search = location.search;
var params = new URLSearchParams(search);
var getType = params.get('totals');
var total = parseInt(getType)

$(document).ready(function () {
    $('#radioButton').click(function () {
        var R1 = $('input[name="Q1"]:checked').val();
        var R2 = $('input[name="Q2"]:checked').val();
        //var R3 = $('input[name="Q3"]:checked').val();
        var R4 = $('input[name="Q4"]:checked').val();
        var R5 = $('input[name="Q5"]:checked').val();
        var R6 = $('input[name="Q6"]:checked').val();

        var Q1 = parseInt(R1);
        var Q2 = parseInt(R2);
        //var Q3 = parseInt(R3);
        var Q4 = parseInt(R4);
        var Q5 = parseInt(R5);
        var Q6 = parseInt(R6);

        var res = parseInt(Q1) + parseInt(Q2) + parseInt(Q4) + parseInt(Q5) + parseInt(Q6);


        window.location.href = "portpolio.html?res=" + res + "?total=" + total;
```

04

# 프로젝트 발전 방안 및 개발 후기

# 발전 방안

## 주가 예측 모델 성능 향상

유가, 광물 가격, 실리콘 가격 등 기타 원자재 가격이나
필라델피아 반도체 지수와 같은 산업군 관련 지표를
활용하여 예측 정확도 향상

## 장기 예측 모델 개발

현재 LSTM + fbprophet 모델에서 fbprophet을 활용
하여 시계열 예측으로 미래 features를 예측하는데,
이를 다른 방식으로 대체하여 더욱 정확하게 예측한다면
성능이 향상되고 장기 예측이 가능할 것으로 기대된다.

## 고객 최적화 포트폴리오 모델 개발

현재 고객의 위험 감수 성향으로만 고객을 분류하고
포트폴리오를 추천해 주지만, 포트폴리오 구성 과정에서
비재무적 지표 활용하여 투자자의 취향을 반영할 수 있다.
(ESG 등 특정 섹터나 테마에 대한 선호)

# 개발 후기

## 이재웅

PM 역할을 수행하면서 프로젝트에서 발생할 수 있는 다양한 문제를 경험한 값진 시간이었습니다. 병목현상을 비롯한 프로젝트 진행중 발생할 수 있는 문제의 해결책을 고민해보게 되었으며 WBS, 와이어프레임 등 프로젝트 산출물의 중요성을 알게 되었습니다. 또한 다양한 머신러닝, 딥러닝 기법을 응용하여 통합된 모델을 만드는 과정이 쉽지 않았지만 모델링에 대한 이해를 높일 수 있었습니다. 쉽지 않은 프로젝트 였지만 책임감 있게 맡은 일을 수행해준 팀원들 덕분에 만족할 만한 결과를 얻을 수 있었습니다. 감사합니다.

## 김병현

동일한 실력을 가지지 않은 다섯명이 팀을 이뤄서 같은 과제를 수행하며 다양한 문제를 맞닥뜨렸을 때 해결하는 과정에서 성취감을 느꼈습니다. 역할 분담 후 여러 과정들을 병합하는 과정에서 다른 사람들의 코드를 보고 이해하는데에 있어서 힘든점이 있었습니다

## 서미오

웹 스크래핑과 API로 데이터를 가져온 후 전처리를 하고, DB에 연동하는 과정에서 겪은  다양한 시행착오가 기억에 남습니다. 지속적인 서비스를 위해 스케쥴러도 적용하니 오류가 끝없이 나왔습니다. 시간이 부족해 모든 케이스를 고려한  코드를 구현하지 못한게 아쉽습니다. 또한 계속 담당했던 프론트가 아닌 백엔드를 해보니 시스템 프로세스와 다른 팀원들의 고충을 이해하게 되었습니다!

## 송건룡

마지막 프로젝트라서 다들 최선을 다한게 느껴졌습니다. 서로 모르거나 부족한 부분을 도와주기도 하고, PM분이 역할을 잘 해주셔서 팀 프로젝트가 원활하게 이루어 질 수 있었던거 같습니다.

## 유준웅

주식종목에 포토폴리오를 제공함에 있어서 많은 지수들이 얽히고설켜 있다는것을 알 수 있었고 그렇기 때문에 모델링 과정의 주식을 예측하는데 있어서 그 정확성을 높히는것이 상당히 어려운 작업임을 느낄 수 있었습니다. 비록 실력이 부족해서 많이 참여할수는 없었지만, 팀원들을 통해서 많은것을 배울 수 있어서 유익한 시간을 가질 수 있었습니다.

# THANK YOU