**컴퓨터공학 종합설계**

|  |  |
| --- | --- |
| **팀명** | 주린이들 |
| **팀원** | 신보라 안광욱 정주은 |
| **제출일자** | 2020.12.20 |
|  |  |



목차

1. **주제1**
2. **배경2**
3. **개발내용3**
4. 플랫폼1
5. 개발언어2
6. 상세구조3

1. **개발방법4**
2. Data Crawler1
3. Machine Learning Model, Portfolio Builder2
4. Trader, UI3
5. 역할분담 4
6. **개발 결과물5**
7. Demo1
8. 기능설명2
9. **최종평가6**
10. 기능적 요구사항1
11. 비기능적 요구사항2
12. 역할 별 평가3
13. **추가사항7**
14. 개선 방안1
15. 활용 계획2
16. **참고문헌8**

인공지능 및 자동매매 알고리즘을 이용해 주식 단기매매를 통해 일반인들도 수익을 볼 수 있도록 하는 프로그램을 제안한다. 인공지능이 과거의 주식 데이터를 이용하여 종목을 선정하고, 사용자 입력 값에 기반하여 주식매매를 자동화하는 프로그램을 구현하고자 한다.

1. 주제

인공지능을 이용한 주식 단기매매 자동화 프로그램

1. 배경

* 동기

주식 단기 매매는 많은 노하우와 경험을 갖고 있는 전업투자자들이 아니면 일반인들이 수익을 보기 어려운 구조로, 일반인들에게는 진입 장벽이 높은 투자 방식이다. 이러한 노하우와 경험을 인공지능이 과거 주식 데이터를 이용하여 학습하게 만들고, 매매를 자동화하여 일반인들에게 주식 단타 매매의 진입장벽을 낮추고 좀 더 안정적인 수입원이 되도록 하고자 본 프로그램을 제안하게 되었다.

* 기대되는 효과
  + 바쁜 현대인들은 주식 시장 내내 주식 상황을 지켜보며 대처하기에는 무리가 있다. 일정시간 주식 차트를 보지 못할 상황의 경우, 적정 매도 타이밍을 놓쳐 손해를 볼 수도 있지만, 매도가 자동화된다면 이러한 문제는 해결될 수 있다.
  + 사람이 주식 단기 매매를 할 경우, 매도를 해야 할 상황이지만 감정에 치우쳐 매도를 하지 못해 결국 손해를 보게 되는 상황이 자주 발생한다. 하지만, 기계는 정해진 알고리즘대로 주식을 거래하므로 더 이성적인 판단을 할 것이라고 기대된다.
  + 단기 매매를 통해 그 날 바로 수익을 볼 수 있다.
* 관련 분야 survey
  + 삼성증권 ‘AI 주식선호분석’

인공지능이 고객의 투자 패턴을 분석하여 투자 시 주가 상승률이 가장 높을 것으로 추정되는 7개 종목을 선별해 제시한다. 자동화된 매매는 제공하지 않는다.

* + AIM

인공지능을 통해 자산관리 인사이트, 시장 전망 등의 정보 제공한다. 자동화된 매매는 제공하지 않는다.

* + 알파프로

AI를 이용한 자동 주식매매를 제공하지만, 단기 투자에 특화 되어있는 프로그램은 아니다.

* 기존 프로젝트와 제안 내용과의 차이점
  + 인공지능에 로직을 더한 자동매매 제안

인공지능으로 종목을 추천하고, 매수 매도는 로직에 의해 수행하는 알고리즘을 만들고자 한다. 로직에 의해 매수와 매도를 한다면, 이상적인 시점을 찾는 것은 아니지만, 리스크를 줄일 수 있는 방법이라고 판단하였다.

* + 사용자의 투자성향 반영

제안하는 프로젝트에서는 사용자가 종목 선정 조건식 및 목표수익 및 손절가와 같은 항목을 입력할 수 있다. 이는 완전히 프로그램에게 의존하는 것이 아닌, 사용자 개인의 투자성향을 반영하고 인공지능 투자의 리스크를 줄일 수 있다.

* + 단기매매에 특화된 알고리즘

기존의 프로젝트들 중에는, 단기매매에 특화된 프로젝트는 없었다. 단기 매매는 일반 투자자들에게 굉장히 어려운 투자 방법으로 알려져 있지만, 본 프로젝트를 통해 일반인들이 단기 매매를 통해 매일 수익을 낼 수 있는 알고리즘을 제시하고자 한다.

1. 개발 내용

* 개발 플랫폼: Window10
* 개발 언어: python
* 상세 구조

제안하는 프로그램은 다음과 같이 크게 5단계로 나누어 볼 수 있다.

1. **Data crawler**

키움증권으로부터 사용자 정보, 과거의 주식 데이터를 받아온다.

1. **ML Model**

과거의 주식 데이터를 입력으로 받아 각 종목별 다음날의 고가를 예측한다.

1. **Portfolio Builder**

ML model에서 분석된 자료를 토대로 종목을 선정한다.

1. **Trader**

Portfolio builder에서 선정된 종목을 바탕으로 실시간으로 주식 거래를 자동화하게 해주는 알고리즘이다.

1. **UI**

사용자가 프로그램을 이용할 수 있는 환경을 제공한다.

앉아있는, 측정기, 전화, 주차장이(가) 표시된 사진

자동 생성된 설명

[그림1. Data flow diagram]

* 상세 요구사항
  + 기능적 요구사항
    - 거래 설정: 매도 전략을 입력하여 해당 전략을 기반으로 자동매매를 실시한다.
    - AI 설정: Portfolio builder에서 인공지능이 선정한 종목 중에서 원하는 종목을 제거 혹은 추가할 수 있다. UI 참고.
* 비기능적 요구사항
  + - performance: 조건에 맞게 매수/매도 주문 전달부터 주문완료까지 걸리는 시간은 1초 이내로 한다. (키움 Open API의 성능에 따름)
    - Usability: 조작하기 쉬운, 간단한 UI 구성한다.

증권사 앱에서 이미 제공하고 있는 기능들보다는, 자동매매를 위한 세팅에 중점을 맞춘 기능들을 제공한다.

* + - Reliability: 전체 동작 중, 에러 없이 동작하는 비율은 95%을 목표로 한다.
    - Security, safety: 키움 open API를 이용한 인증서 로그인 제공한다.

1. 개발 방법

* Data Collection: 과거의 방대한 데이터를 가져오기 위해서는 키움 Open API 보다, Creon Plus API 가 더 많은 데이터를 제공했기 때문에, Creon Plus API를 사용했다. 코스피 200종목에 대한 종목별 주식데이터(시가, 고가, 저가, 종가, 거래량)을 받아와서 csv파일에 저장한다.
* Data Crawling: 매일 데이터 업데이트 시에는 키움 Open API 이용하여 추가로 데이터를 받아와 csv 파일에 저장한다.
* ML Model:
* 사용 라이브러리: keras, tensorflow, sklearn
* 데이터 전처리

1. 기존의 feature (시가, 고가, 저가, 종가, 거래량)에 추가로 종목 코드를 새로운 feature로 추가
2. 가격관련 지표와 거래량 관련지표를 나누어 각각 0~1값으로 scaling
3. Sliding window를 적용하여 일봉 데이터의 경우 3달 단위, 5분봉 데이터의 경우 3일 단위로 잘라 모델에 들어갈 데이터 생성

* 모델의 input:

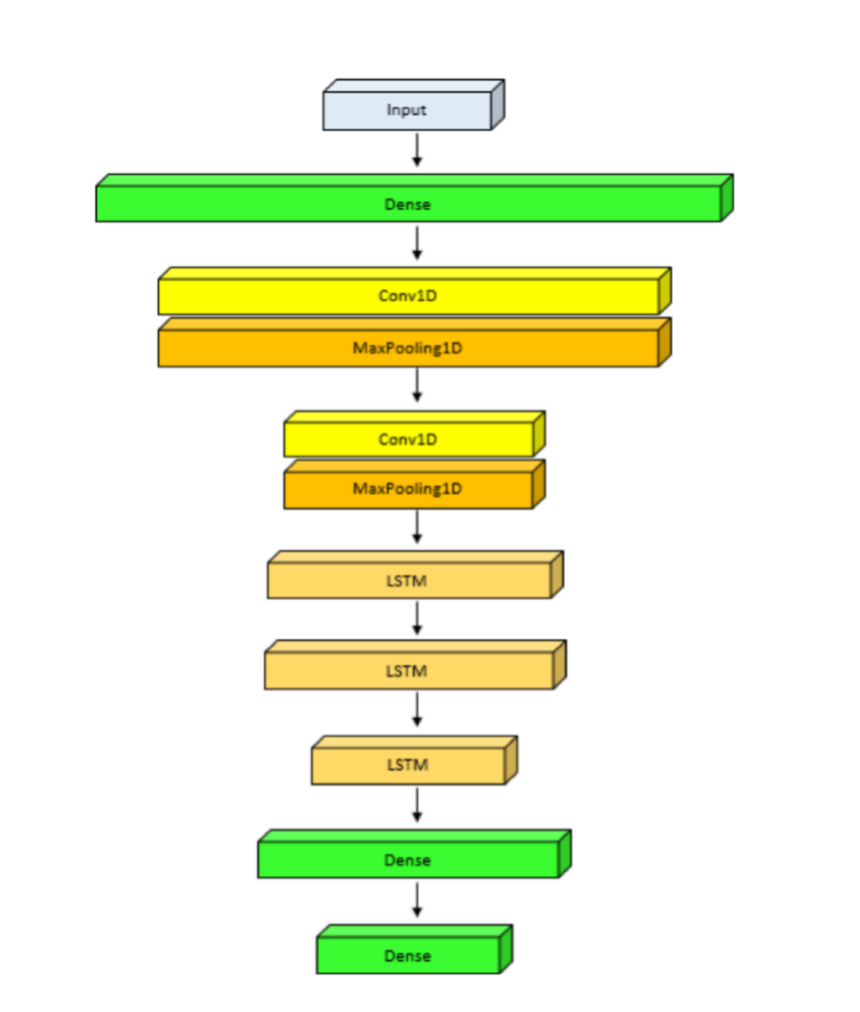
1. 3달 동안의 일별 주가 데이터
2. 3일 동안의 5분 단위의 주가 데이터

* 모델의 output: 다음날의 고가

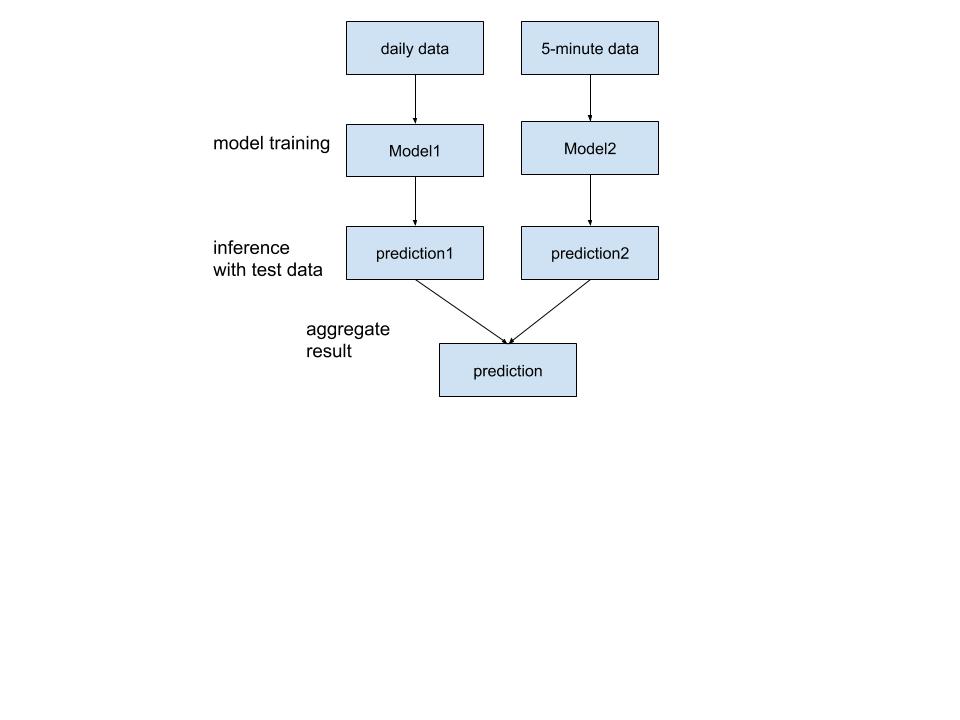
최종적으로, 전날 종가 대비 다음 날 고가의 상승률이 높은 종목들을 선정한다.

* 모델 구조 [1] [2] [3] [4]

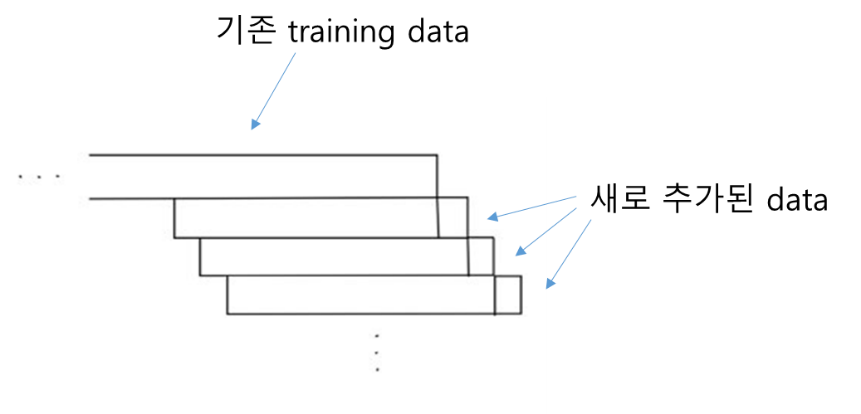
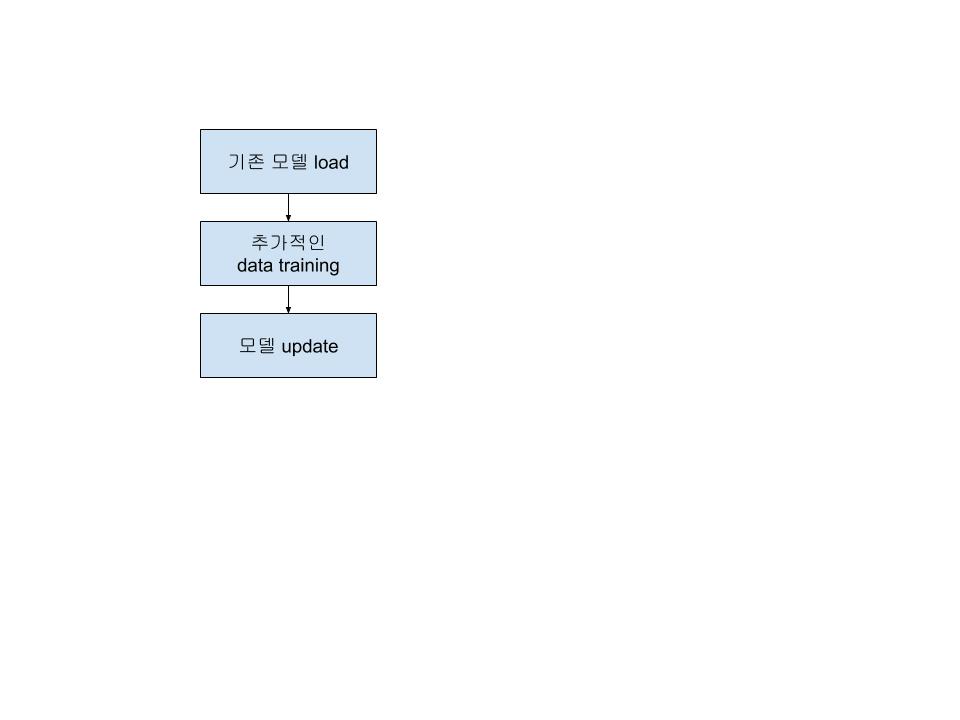
Conv1D레이어 2개와 LSTM 레이어 2개로 이루어진 모델



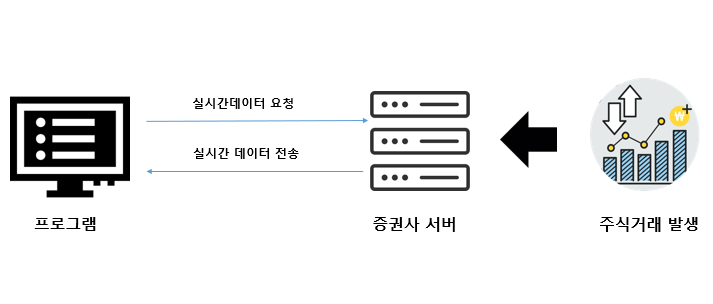
일별 데이터와 5분 단위 데이터는 따로 학습을 진행하고, 앙상블 학습을 이용하여 각 모델의 예측값의 평균을 이용하여 최종적으로 다음날의 고가 예측



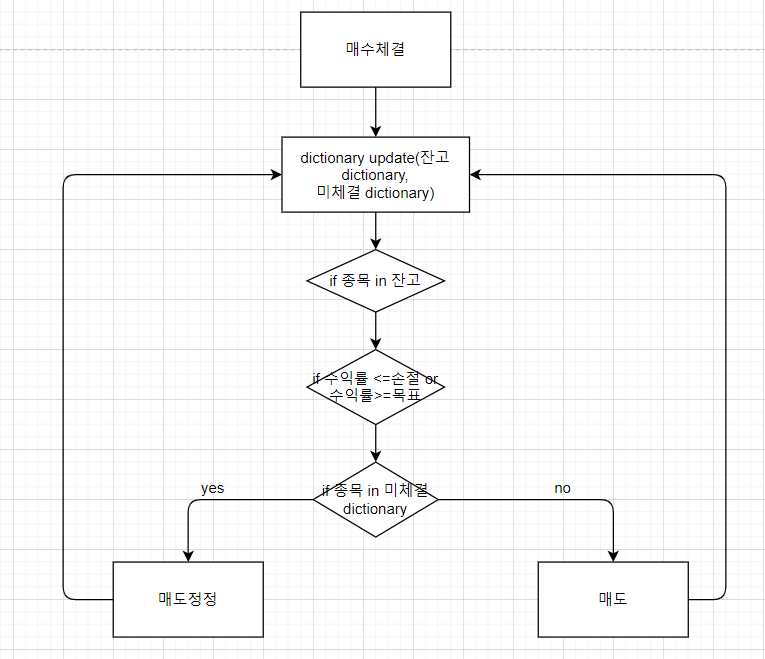
* Online learning
* 모델이 만들어진 시점까지의 데이터만을 활용하여 학습하면, 해당 모델을 계속해서 사용할 때 그 모델은 이전의 패턴만을 학습한 모델이 되어 새로운 경향을 반영하지 못하는 모델이 될 수 있다. 이를 보안하기 위해, 새로 생성되는 데이터에 대해서 계속해서 학습할 수 있도록 온라인 학습을 적용했다. 저장된 모델을 불러와서, 새로 생긴 200개 종목의 데이터를 추가로 학습시킨 뒤, 모델을 새로 업데이트 한다. 추가적인 데이터를 트레이닝 할 때는, 전체 데이터에 대해 학습하는 것이 아닌 기존 training data의 일부와 새로 추가된 data를 학습한다. 일 단위 데이터의 경우 6개월치 데이터, 5분 단위 데이터의 경우 10일치 데이터를 추가로 학습하게 된다. 일정한 크기의 데이터로 추가 트레이닝을 하는 방식이므로, 추가되는 데이터만큼 옛날 데이터는 배제되게 된다. 이를 통해 과거의 데이터보다는 현재 데이터의 패턴위주로 반영 되는 효과가 생기게 된다.



* **Trader 구현 :**
* 사용 API: 키움증권 Open Api (개발자들을 위한 Open Api 가이드북, KOA Studio 에서 제공하는 개발자들을 위한 api의 기본적인 함수 수도 코드 참고하였음)
* 주문할 계좌 : dynamicCall-GetLoginInfo 함수로 서버에 보유계좌 정보 요청 후 사용자에게 선택할 수 있게 구현했다.
* 매도 설정 : 사용자가 입력한 목표 수익률과 손절률을 float으로 변환, 손절률은 -를 적용하였음. Default 값은 손익비를 최대한 고려하여 구현했다.
* 추천종목 리스트 수정 : 사용자가 입력한 값을 입력하고 그에 대해 추가 혹은 제거를 누르게 되면 종목리스트에 있으면 추가 혹은 제거 되도록 설정했따
* 매수매도 : 매수매도는 틱봉별로 거래되도록 구현 하였음. 사용자가 수정한 종목리스트에 있는 종목들에 대해 dynamicCall\_SetRealReg 함수를 통해 증권사 서버에 실시간 정보를 요청하게 된다. 증권사 서버는 요청받은 종목들에 한해서 거래가 발생할 때마다 제공해준다. 그에 대한 과정은 다음 그림과 같다.

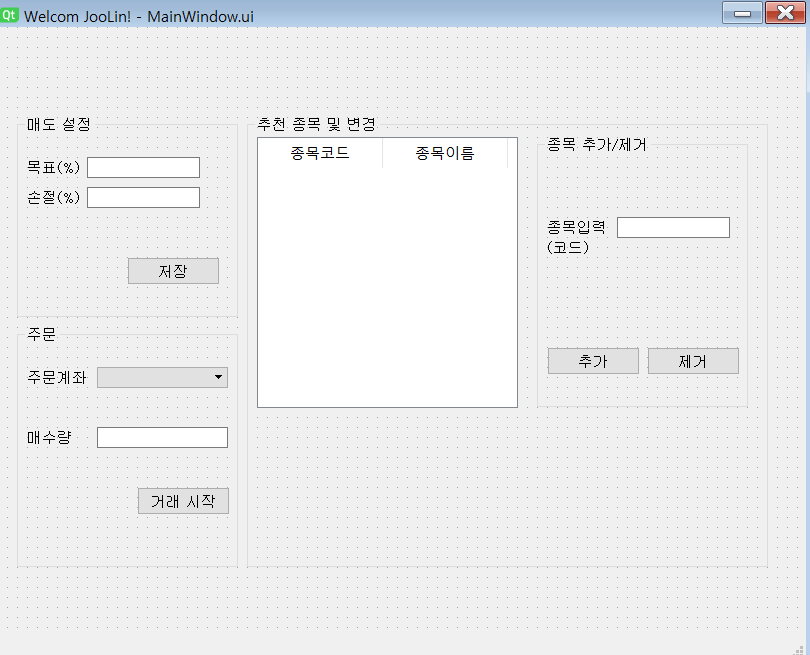


프로그램에서는 실시간 정보를 받아서 매수 혹은 매도를 진행하게 된다. 매수가 진행된 종목들에 대해서는 중복 매수를 방지하기 위해 done list를 생성해서 매수체결이 완료된 종목은 done에 update하도록 했다. 또한 매수매도의 설정은 Default 값으로 목표 수익률이 손절률의 2배가 되도록 설정하였다. 이는 단기매매에서 중요한 여러 종목을 거래할 시 중요한 손익비를 최대화하기 위함이다.

* 매수매도 logic은 다음그림과 같다.

[Logic of Trader]

* **UI**
* Tool : PyQt
* 매도 설정 : 목표 및 손절률을 설정할 수 있게 구현.
* 종목리스트 설정 : QTableWidget을 사용하여 종목리스트를 사용자에게 제공하고 추가 및 제거하면 그에 따라 다시 종목리스트를 업데이트하여 사용자에게 제공는 기능을 구현
* 주문계좌 선택 : ComboBox를 사용하여 사용자의 보유계좌중 거래하고 싶은 계좌를 선택할 수 있게 하였음.
* 거래 : 거래 시작 버튼을 눌러 사용자가 원하는 때에 거래하도록 거래시작버튼을 구현
* 매수량 : 매수할 양을 사용자가 선택할 수 있게 구현

[UI Screen Shot]

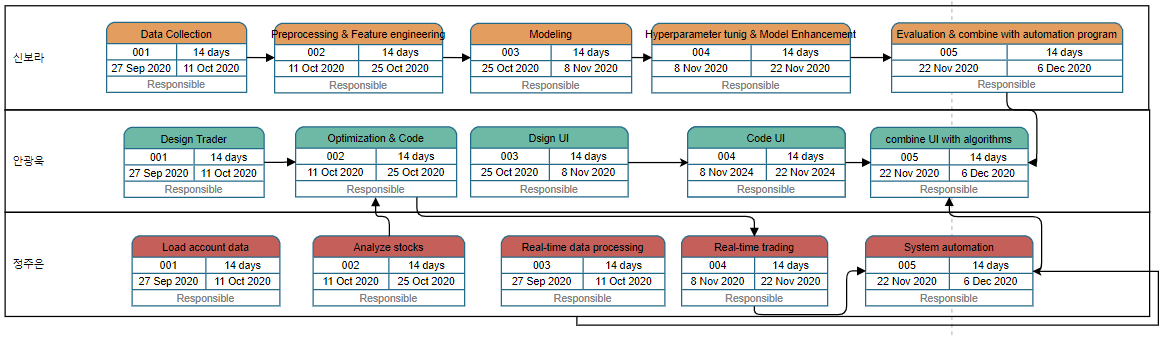
* 역할 분담

신보라: Data Collection, ML Model

안광욱: Trader, UI

정주은: Data Crawler, Open API

* PERT chart



* Gantt chart

스크린샷이(가) 표시된 사진

자동 생성된 설명

1. 개발 결과물

* 기능적 요구사항 : 사용자가 매수량, 주문계좌, 추천 종목리스트, 매도 설정 등을 하도록 UI를 구현 하였음.
* 데모링크 : <https://youtu.be/Uxdul4Z53ck>

1. 최종평가
   * 기능적 평가
     + 거래 설정: 매도 전략을 입력하여 해당 전략을 기반으로 자동매매를 실시 🡪 만족
     + AI 설정: Portfolio builder에서 인공지능이 선정한 종목 중에서 원하는 종목을 제거 혹은 추가 기능 🡪 만족
     + 잔고조회 🡪 자동매매에 있어 주요기능이 아니기에 구현하지 않았음.

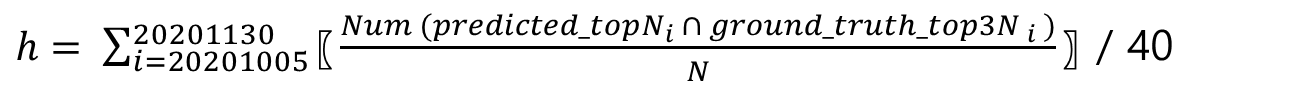
* 비기능적 평가
  + - performance: 조건에 맞게 매수/매도 주문 전달부터 주문완료까지 걸리는 시간: 평균 0.7s(100회 테스트)
    - Usability: 조작하기 쉬운, 간단한 UI 구성한다.

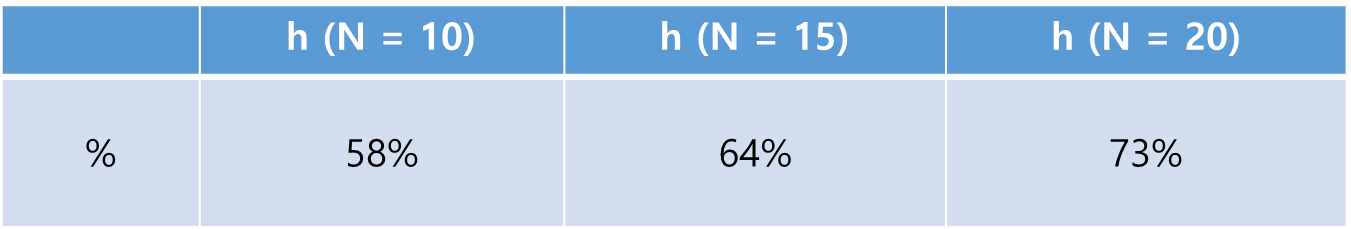
증권사 앱에서 이미 제공하고 있는 기능들보다는, 자동매매를 위한 세팅에 중점을 맞춘 기능들을 제공한다.

* + - Reliability: 전체 동작 중, 에러 없이 동작하는 비율: 100% (프로그램 테스트 중에러 경험 X)
    - Security, safety: 키움 open API를 이용한 인증서 로그인 제공한다.
  + Data Crawler : (담당: 정주은)
* 예수금 정보, 계좌평가잔고내역, 미체결 종목 등 계좌 정보 수집 🡪 만족
* 실시간 종목 데이터와 다음 날을 위한 종목 데이터 수집 🡪 만족
  + Machine Learning Model : (담당: 신보라)

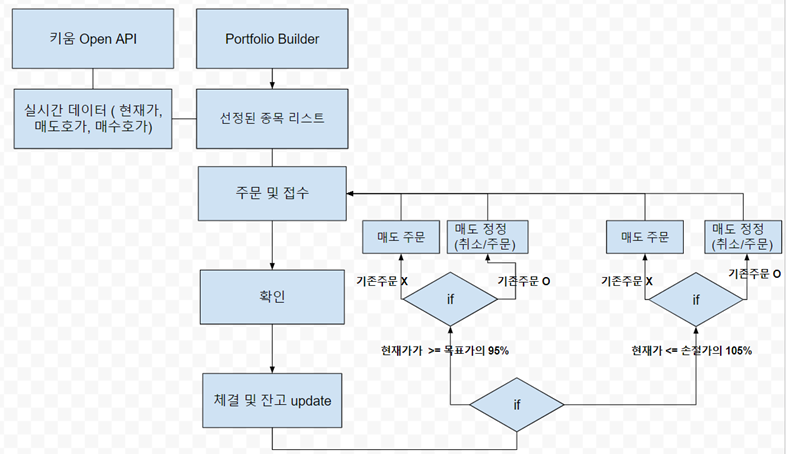
train data와 validation data는 2020년 10월02일까지의 데이터를 사용했고, 각각 랜덤으로 8:2로 나누어 사용하였다. Validation data를 특정 기간의 데이터로 할 경우에, 해당 기간에 bias가 존재할 수 있으므로, validation data를 랜덤으로 뽑아 전체 기간에 걸치게 하였다. 모델을 최종적으로 평가할 Test data는 2020/10/05 ~ 2020/11/30까지 40일 간의 데이터를 이용하였다.

모델이 선정한 상승률이 높은 topN개의 종목이 실제로 상승률이 높은 top 3N개의 종목에 들어있을 확률을 이용해 모델을 평가하였다. 40일간의 테스트 데이터를 이용하여 평가하였고, 그 평균치를 사용했다. (전체 종목 개수는 KOSPI 200 지수를 따르는 200개 종목이다.)





원래의 목표치는 h가 20%이상인 것인데, N이 20일 때 목표치를 만족하는 것을 알 수 있다. 본 모델은 적은 수의 종목을 추천할 때보다는, 20개 정도의 종목을 추천할 때 의미있는 결과를 얻을 수 있음을 알 수 있다.

* Trader: 손익비를 최대화하는 알고리즘 구현 🡪 만족 (담당: 안광욱)
* 목표 : 작지만 안전한 수익률을 목표로하여 다음 그림과 같은 알고리즘 구현

[그림. 목표 logic]

* 평가: 다음 그림과 같이 구현 성공. 변경된 사항은 원래 매도 정정 시에 기존의 주문을 취소하고 다시 매도를 주문하는 방식으로 했는데, 매도 정정이라는 기능이 있어서 취소->재주문의 과정을 매도정정 하나의 과정으로 더 효율적으로 구현
* UI: (담당: 안광욱)
  + 사용자가 매도 전략을 입력할 수 있는 기능을 구현 🡪 만족
  + Portfolio builder에서 인공지능이 선정한 종목 중에서 원하는 종목을 제거 혹은 추가할 수 있는 기능 구현 🡪 만족
  + 종목 선정을 위한 조건식을 설정할 수 있는 기능을 구현 🡪 만족
  + 자동매매 시작할 수 있는 기능 🡪 만족

1. 결론

* 향후 활용 방안: 주식 단기 매매시에 손절가와 목표가를 선택하고 거래를 진행하면, 원하는 가격에 매매를 진행할 수 있다.
* 추가 진행 및 개선 방안 :
* ML Model: 주가의 변화에는 과거의 주가 데이터뿐 아니라, 많은 요소들이 존재한다. 단순히 과거의 주가 데이터만을 이용하여 종목을 추천하는 것에 한계가 있으므로, 뉴스나 다른 지표를 활용하여 주가 분석을 하도록 개선이 필요하다.
* Trader : 차트를 수학적으로 분석하는 알고리즘을 추가해서 각 종목별 최대의 손익을 볼 수 있도록 할 것.

# 인용 자료

|  |  |
| --- | --- |
| [1] | D. Wei, “Prediction of Stock Price Based on LSTM Neural Network,” %1 *nternational Conference on Artificial Intelligence and Advanced Manufacturing (AIAM)*, 2019. |
| [2] | Y. C. a. Y. L. T. Gao, “Applying long short term momory neural networks for predicting stock closing price,” %1 *8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, 2017. |
| [3] | D. R. G. D. A. A. M. Sneh Jain, “Stock Price Prediction on Daily Stock Data using Deep Neural Networks,” 2018. |
| [4] | R. G. a. A. A. M. S. Jain, “Stock Price Prediction on Daily Stock Data using Deep Neural Networks,” %1 *International Conference on Advanced Computation and Telecommunication (ICACAT)*, 2018. |