평균회귀(Mean reversion)이론을 사용한 주식동향예측 및 모의 투자

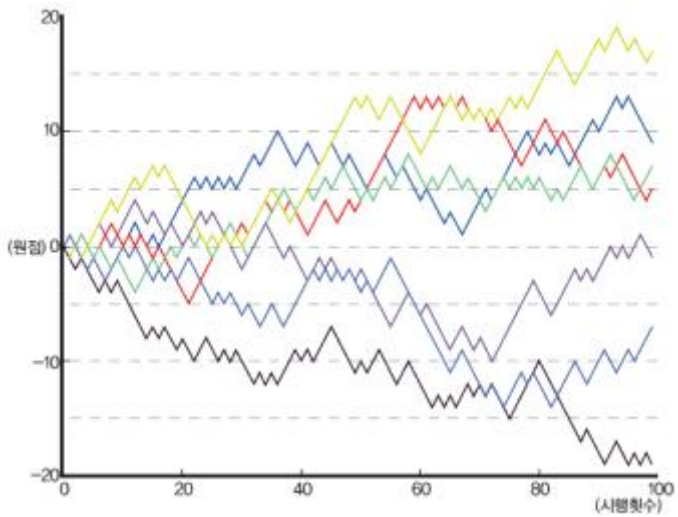
컴퓨터 공학과 2011104027 안재성

요약

평균회귀(Mean reversion)이론에 부합하는 주식 데이터를 간추려내 해당 주식의 동향 예측 결과를 손 보이고, 모의 투자를 진행한 뒤 그 결과를 분석하여 활용 가능성을 파악한다.

I. 도입: 연구 배경 및 목적과 기대 효과

I.1. 연구 흐름



- 연구 배경

일반적으로 학계에서는 주가데이터를 위 그림처럼 랜덤워크(Random walk)라고 가정한다. 주가의 움직임이 불규칙하다는 것은 이제 상식이 되었지만, 사실 주가의 움직임이 무작위적이라는 것을 알게 된 것은 여러 수학자와 경제학자들의 오랜 연구의 결과 덕분이었다. 하지만 이러한 데이터의 흐름을 분석해보면 적정한 수준에서 유지되던 주가가 별다른 이유 없이 올랐다가는 떨어지고 떨어졌다가는 다시 제자리를 찾는 현상이 관찰된다. 이러한 현상을 평균회귀(Mean reversion)라 하며, 이 현상을 띄는 주식을 파악하여 수익을 낼 수 있는 지 알아낸다.

- 연구 목적

평균회귀(Mean reversion)은 통계학에서 말하는 평균으로의 회귀와는 약간의 차이가 있다. 평균으로의 회귀는 정의로부터 유도되는 논리적인 수학적 법칙이고, 평균회귀(Mean reversion)는 이런 법칙에 더해 주가, 상승률 등의 지표가 과거 평균, 혹은 전체평균에 수렴

하는 경향이 있다는 가설이다.

이러한 가설을 바탕으로 평균회귀(Mean reversion) 성향도를 측정하는 알고리즘을 사용하여 성향도의 차이에 따른 주식그래프를 비교해본다. 이후 평균회귀(Mean reversion) 성향이 높은 주식의 동향을 예측한다. 예측한 동향 값으로 주식모의투자를 진행한 뒤 수익률을 파악하여, 해당 자료를 근거로 평균회귀(Mean reversion)이론을 주식투자에 응용하는 것이 긍정적일지 알아낸다.

I.2 관련 연구 조사

-평균회귀(Mean reversion) 성향 파악

평균회귀(Mean reversion) 성향을 파악하기 위한 알고리즘은 다양하게 존재하므로 본인은 관련 알고리즘에 대한 논문과 서적을 다양하게 조사하였으며, 주가 데이터의 평균회귀(Mean reversion) 성향을 파악하기 위한 알고리즘을 동시에 3가지를 활용하기로 결정하였다.

-Augmented Dickey-Fuller test(ADF)

어떤 시계열 데이터가 랜덤워크(Random walk)를 따른다는 가설을 세운 뒤, 이 가설을 검증하여 랜덤워크(Random walk)가 아닌지 판단한다. 해당 테스트는 "몬테 카를로 실험에 의한 Augmented Dickey-Fuller 단근 검정법의 검정력에 한 연구"논문에서 검정력을 보였듯이 사용할 만하다.

이제, ADF 테스트를 진행해보자. 아래와 같은 시계열데이터가 있다고 가정한다.

$$y_t = \alpha + \beta t + \gamma y_{t-1} + \sum_{i=1}^{p} \nabla y_{t-i} + e_t$$

여기서 는 상수, β는 트랜드 계수이다. 랜덤워크(Random walk)는 정의에 따라 현재의 값이 이전 값에 영향을 받지 않는 특성을 지닌다. 따라서 y 와 y_{t-1}간에는 상관관계가 없어야 하므로 γ계수는 0이 되어야한다. 해당 시계열데이터가 랜덤워크(Random walk)라면 γ=0, α=0, β=0이라고 가정하고, 가정검정을 거친다. γ =0 이라는 가설검정이 실패하면 시계열데이터는 랜덤워크(Random walk)가 아니라고 증명된다. 여기서 얻은 결과 값으로 평균회귀(Mean reversion) 성향을 파악한다.

-Hurst-Exponent(허스트지수)

주가 데이터는 시간에 따라 주식 값이 확산하게 된다. 이 확산하는 속도를 수학적으로 계산이 가능하다. Hurst-Exponent는 여기서 분산을 확산속도로 치환하여 GBM의 속도를 기준으로 비교한다. 여기서 GBM(기하적 브라운 운동)은 표류하는 랜덤워크(Random walk)이다.

$$X(t) = \sum_{1}^{t} (\xi(t) - \langle \xi \rangle_\tau)$$

$$R/S = \frac{R(\tau)}{S(\tau)} \qquad S(\tau) = \sqrt{\frac{1}{\tau}\sum_{t=1}^{\tau} \{\xi(t) - \langle \xi \rangle_\tau\}^2}$$

$$E\left[\frac{R(n)}{S(n)}\right] = Cn^H \quad \text{as } n \to \infty$$

$$E\left[\frac{R(n)}{S(n)}\right] = Cn^{0.5} \quad \text{as } n \to \infty$$

$$ret_{t-\Delta,t} = \log(P_t) - \log(P_{t-\Delta})$$

$$ret_{t-\Delta,t} = \frac{(P_t - P_{t-\Delta})}{P_{t-\Delta}}$$

$$P_j = \frac{1}{2^j}\sum_{i=0}^{2^j-1} C_i^2 \qquad H = \left|\frac{(slope-1)}{2}\right| \qquad H = 1 - \frac{\alpha}{2}$$

여기서 알아낸 허스트지수 H는 GBM의 경우 0.5이며, 0에 가까울수록 평균회귀(Mean reversion), 1에 근접할수록 발산하는 추세경향을 띈다.

-Half-Life

평균으로 회귀하는데 걸리는 시간이다.

$$dx_t = \lambda(\mu - x_t)dt + \sigma dW_t$$

$$f(t) = y_0 e^{-\lambda t}$$

$$f\left(t_{1/2}\right) = \frac{f(t)}{2}$$

$$Half - life, t_{1/2} = -\frac{ln2}{\lambda}$$

첫 번째 식은 오른스타인-우렌벡 과정이며, 는 평균회귀(Mean reversion)의 속도, μ는 평균, d 는 에러텀이다. 여기서 평균회귀(Mean reversion) 성향이 있는 랜덤과정을 오른스타인-우렌벡 과정이라고 한다.

half-life를 간단히 구하기위해 오른스타인-우렌벡과정으로 f(t)를 만들었다. 이때, y_0는 초기치, λ는 평균회귀(Mean reversion)의 속도이다. 평균회귀(Mean reversion)를 따르는 어떤 시계열데이터는 f(t)를 만족해야한다. 따라서 $f(\frac{t}{2}) = y_0 e^{-\lambda}(t/2) = \frac{f(t)}{2}$ 를 성립한다. 이를 토대로 Half-life가 평균회귀(Mean reversion) 속도에 반비례함을 알 수 있다.

I.3 기존 연구의 문제점 및 해결 방안

-문제점

주식투자자들은 주식거래시간 외 시간에도 투자전략(investment strategy)을 고심해야한다. 여기서 평균회귀(Mean reversion) 성향을 띄는 주식을 개인이 판단해 내는 것은 툴을 통하지 않고서는 불가능에 가깝다. 개인이 매일 새로운 데이터를 바탕으로 평균회귀(Mean reversion)성향의 정도를 판단하는 것은 무리며, 다양한 기법을 사용하기도 무리이다.

-해결방안

a. 주식거래시간 외 시간동안에도 주가동향 분석을 지속적으로 해야 하는 전문가들에게는 자동화된 분석 툴이 필연적으로 필요하다. 이러한 전문가들을 위해 자동으로 매일 Yahoo에서 지원하는 일일주식거래데이터를 지원받아 평균회귀(Mean reversion)성향과 동향예측을 계산해준다. 따라서 주식투자자의 시간을 줄여줘 주식투자전략(investment strategy)을 짜는데 도움을 줄 것이다.

b. 평균회귀(Mean reversion) 성향을 판단함에 ADF, Hurst, HalfLife 알고리즘을 동시에 사용하여, 단일 알고리즘을 사용함으로서 가지는 판단오류 가능성을 최소화 할 수 있었다.

II.본론 : 연구 내용 및 세부 계획

II.1. 실제 시나리오

실제 구현 시나리오는 Python으로 돌린 TensorFlow에서 Python코드를 동작시켜서 결과물을 만들어낸다.

a. runDownloadStockData

DataReadWriter를 이용해 Koscom에서 상장회사 정보를, yahoo에서 일일 주가데이터를 가져온다.

b. runCheckMeanReversion

평균회귀(Mean reversion) 성향 테스트(MeanReversionTest)를 진행한다.

$1-hurst) + (adf\ /adf\ + adf_S/adf_5 + adf_S/adf_{10})/4 +\ \ L$ 식을 통해 rank_score값을 계산해낸다. 해당 식은 ADF,hurst,half_life를 1:1:1비율로 적용시키는 식이다. 해당 식에서 HL는 half_life값을 백분위수로 어느 부분에 위치해있는지에 대한 값이다. HL은 half_life값이 작을수록 값이 높다. 이 결과인 rank_score 값으로 정렬한 상위 10%의 데이터만으로 내일 주가동향(direction)을 예측한다. 평균회귀(Mean reversion) 성향 테스트(MeanReversionTest)의 계산과정에 50분정도 소요되고, 동향(direction)예측에 10분정도 소요된다.

c. runCreateStockPrediction
주가데이터와 미리 예측한 동향(direction)을 비교해 예측적중률(correct_ratio)과 모의투자 순이익(money_diff)을 계산한다. 예측적중과 순이익(money_diff)은 HalfLife 평균인 한달 뒤의 결과를 낸다.. 전체가(6+day/10)분정도 소요된다.

d. runShowStockPrediction
예측적중률(correct_ratio)과 순이익(money_diff)을 불러와 사용자가 원하는 대로 전체 통계와 원하는 날짜의 추천 주식을 보여준다.

e. 매일 a,b,c,d순서대로 진행하는 것이 원칙이긴 하나, 순서가 뒤섞여도 동작하는데 문제가 없다.

II. 2. 구현을 위한 SYSTEM요구사항

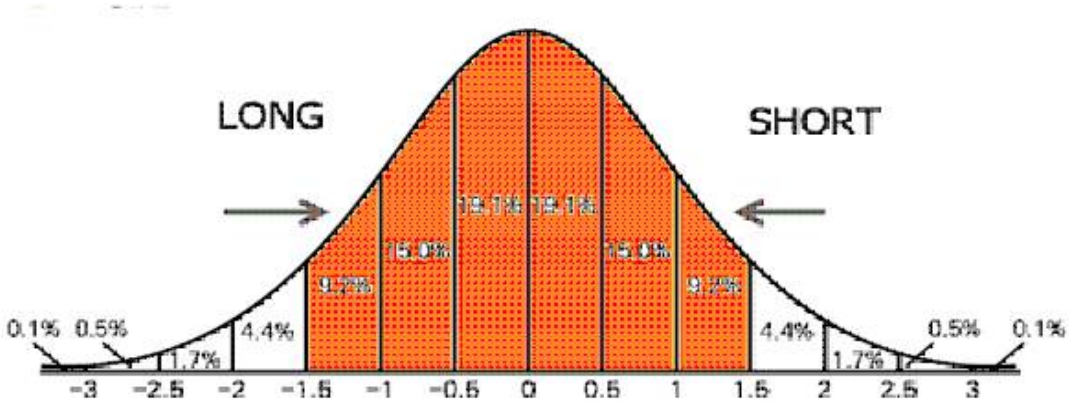
- 구현기반 : python2.7 Application
platform : python2.7 console
IDE : vim
Database : MySql 5.6.25
Library : statsmodels, numpy, pandas, datetime, requests, os, sys, pandas_datareader, BeautifulSoup, logging, MySQLdb, tensorflow

II. 3. 구현 기능 및 세부 계획

- 기본 주식관련 데이터 획득
Koscom에서 상장회사정보와 Yahoo Finance에서 주식데이터를 가져와 DB에 저장.

- 평균회귀(Mean reversion) 성향 파악
Augmented Dickey-Fuller test, Hurst Exponent, Half-life를 통해 평균회귀(Mean reversion) 성향 순위를 파악.

- 동향 예측
평균회귀(Mean reversion) 성향 순위의 상위권의 주식데이터의 평균과 표준편차를 이용해 상승/하강을 예측한다. [fig]의 주황지점 외각에 위치한 13.4%의 주가는 상/하에 따른 방향성을 지닌다고 했다.



[fig] 표준편차를 이용해 $\pm\alpha*1.5$ 범위 바깥에 있는 주가의 방향을 예측

- 예측적중률(correct_ratio)과 모의투자 순이익(money_diff) 분석

HalfLife 평균치의 반인 35일 뒤의 주식데이터를 바탕으로 적중률(correct_ratio)과 순이익(money_diff)을 파악한다.

투자는 LONG(상승예측)판단 1회당 100,000원 어치의 주식을 구입한다고 가정한다.

- 전체 예측 통계를 파악

- 원하는 날짜의 추천 주식을 파악

[시스템에 사용된 Database Tables]

```
mysql> describe codes;
+-------------+-----------+------+-----+---------+-------+
| Field       | Type      | Null | Key | Default | Extra |
+-------------+-----------+------+-----+---------+-------+
| last_update | datetime  | NO   |     | NULL    |       |
| code        | char(20)  | NO   | PRI | NULL    |       |
| full_code   | char(200) | NO   |     | NULL    |       |
| market_type | int(1)    | YES  |     | 0       |       |
| company     | char(100) | NO   |     | NULL    |       |
+-------------+-----------+------+-----+---------+-------+
5 rows in set (0.01 sec)
```

```
mysql> describe prices;
+----------------+----------+------+-----+---------+-------+
| Field          | Type     | Null | Key | Default | Extra |
+----------------+----------+------+-----+---------+-------+
| last_update    | datetime | NO   |     | NULL    |       |
| price_date     | datetime | NO   | PRI | NULL    |       |
| code           | char(20) | NO   | PRI | NULL    |       |
| price_open     | double   | YES  |     | NULL    |       |
| price_close    | double   | YES  |     | NULL    |       |
| price_low      | double   | YES  |     | NULL    |       |
| price_high     | double   | YES  |     | NULL    |       |
| price_adj_close | double  | YES  |     | NULL    |       |
| volume         | int(1)   | YES  |     | NULL    |       |
+----------------+----------+------+-----+---------+-------+
9 rows in set (0.00 sec)
```

```
mysql> describe directions;
+---------------+----------+------+-----+---------+-------+
| Field         | Type     | Null | Key | Default | Extra |
+---------------+----------+------+-----+---------+-------+
| last_update   | datetime | NO   |     | NULL    |       |
| price_date    | datetime | NO   | PRI | NULL    |       |
| code          | char(200) | NO  | PRI | NULL    |       |
| company       | char(200) | NO  |     | NULL    |       |
| target_column | char(30) | NO   | PRI | NULL    |       |
| direction     | char(10) | NO   |     | NULL    |       |
| rank_score    | double   | YES  |     | NULL    |       |
+---------------+----------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

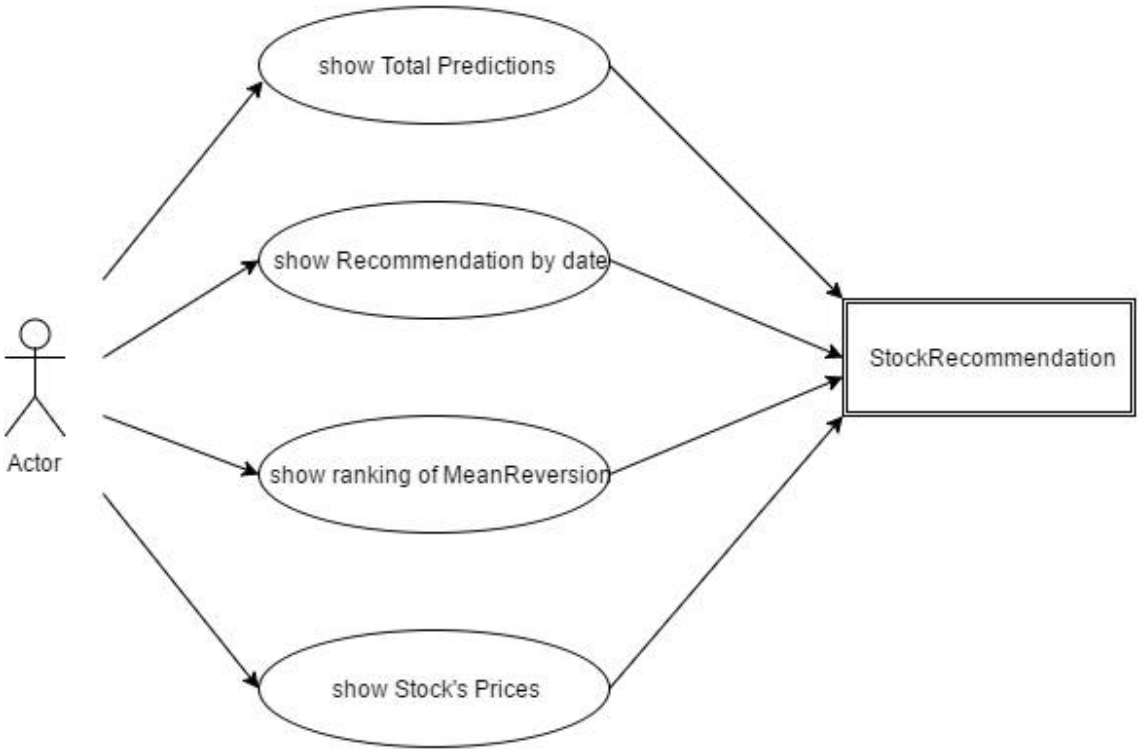
```
mysql> describe countPrediction;
+---------------+----------+------+-----+---------+-------+
| Field         | Type     | Null | Key | Default | Extra |
+---------------+----------+------+-----+---------+-------+
| last_update   | datetime | NO   |     | NULL    |       |
| code          | char(20) | NO   | PRI | NULL    |       |
| company       | char(100) | NO  |     | NULL    |       |
| target_column | char(30) | NO   | PRI | NULL    |       |
| count_true    | int(1)   | YES  |     | NULL    |       |
| count_false   | int(1)   | YES  |     | NULL    |       |
| count_all     | int(1)   | YES  |     | NULL    |       |
| money_diff    | int(1)   | YES  |     | NULL    |       |
+---------------+----------+------+-----+---------+-------+
8 rows in set (0.00 sec)
```

Ⅲ. 시스템 설계

Ⅲ. 1. UML Diagram을 통한 시스템 모델링

a. Use Case Diagram

- Use Case Diagram 표기

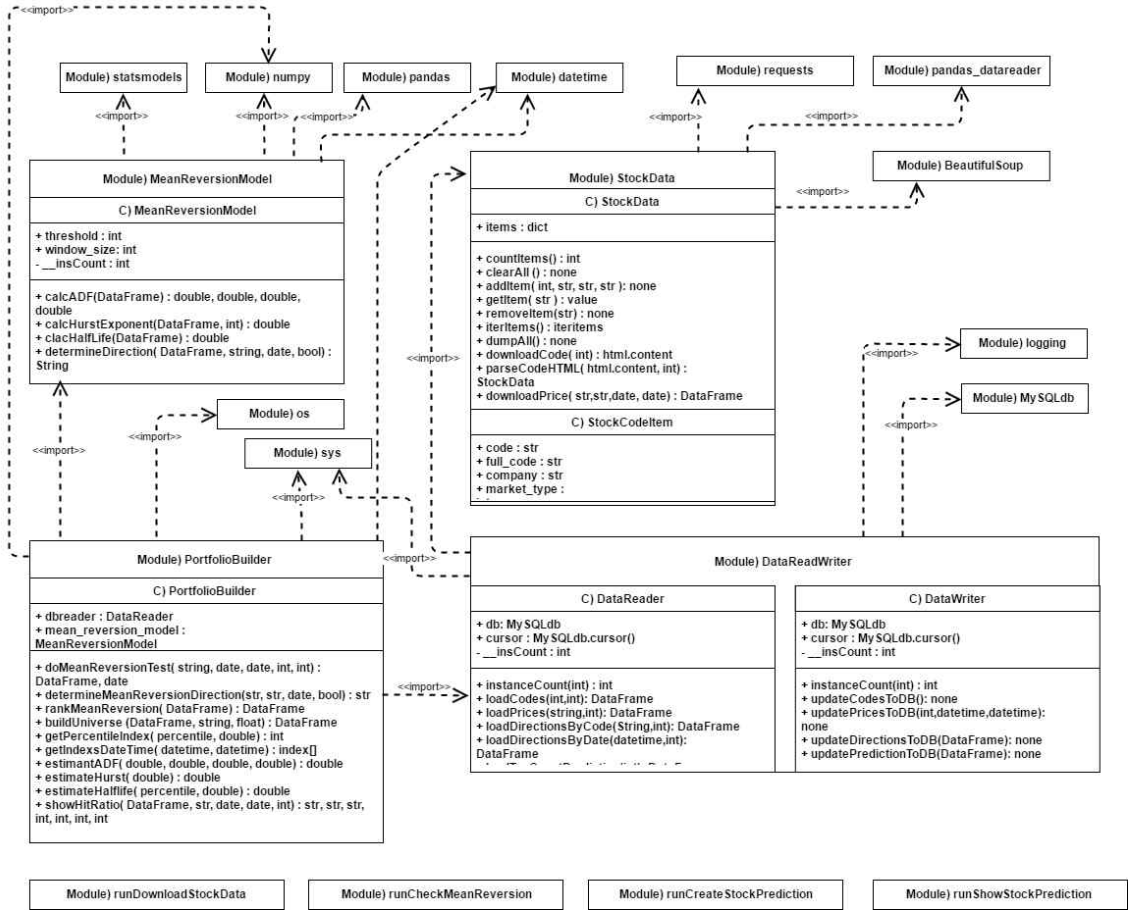


[Fig a.1] Actor 사용자가 주식 추천 시스템의 여러 기능을 동작시키는 Use Case Diagram

해당 그림대로 사용자는 StockRecommendation을 이용해 주식거래 정보부터 평균회귀(Mean reversion)성향 랭킹, 특정날짜의 추천 주식, 주가예측 통계 등을 알아볼 수 있다.

b. Class Diagram

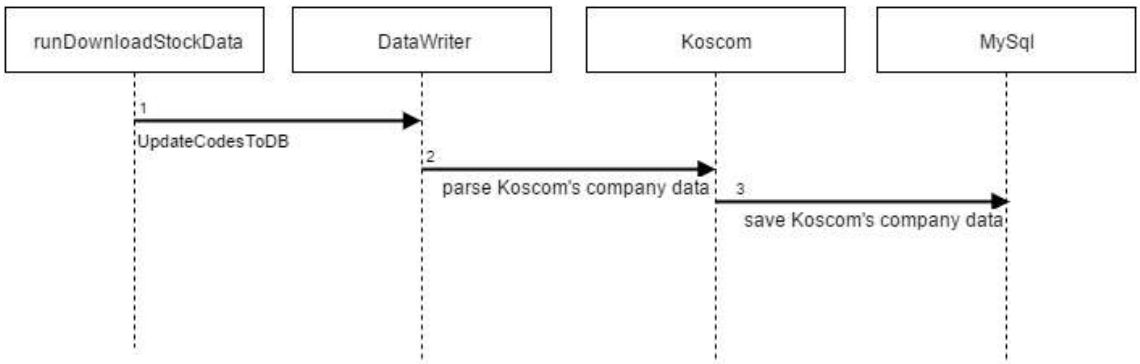
- Class Diagram 표기



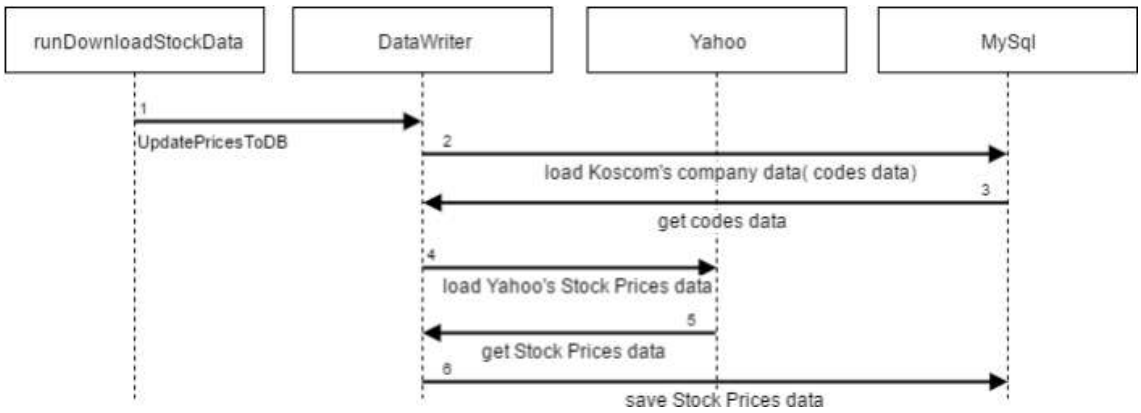
[fig b.1] 전체 System의 Class Diagram (확대 시 잘 보임)

c. Sequence Diagram

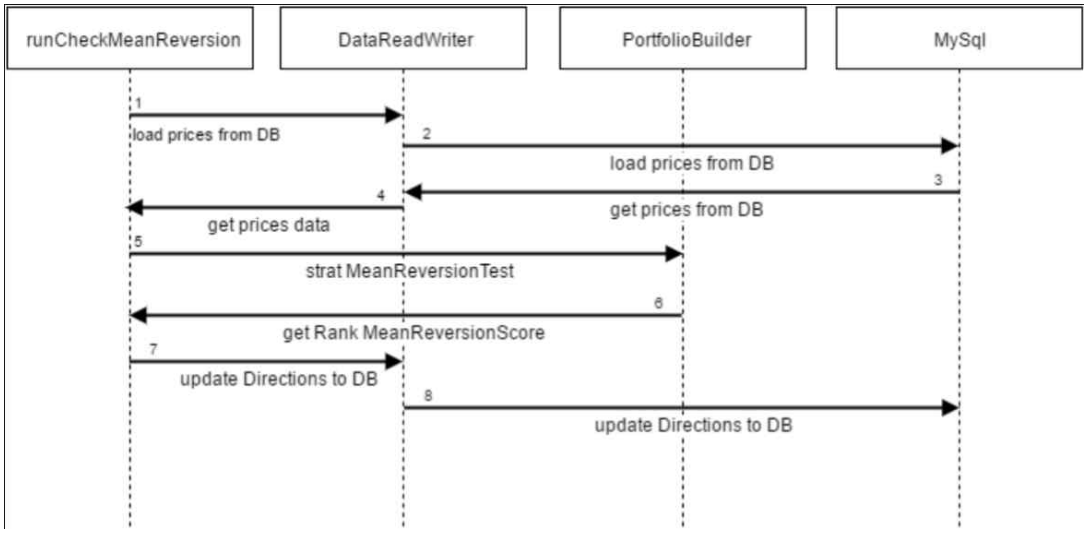
-Sequence Diagram 표기



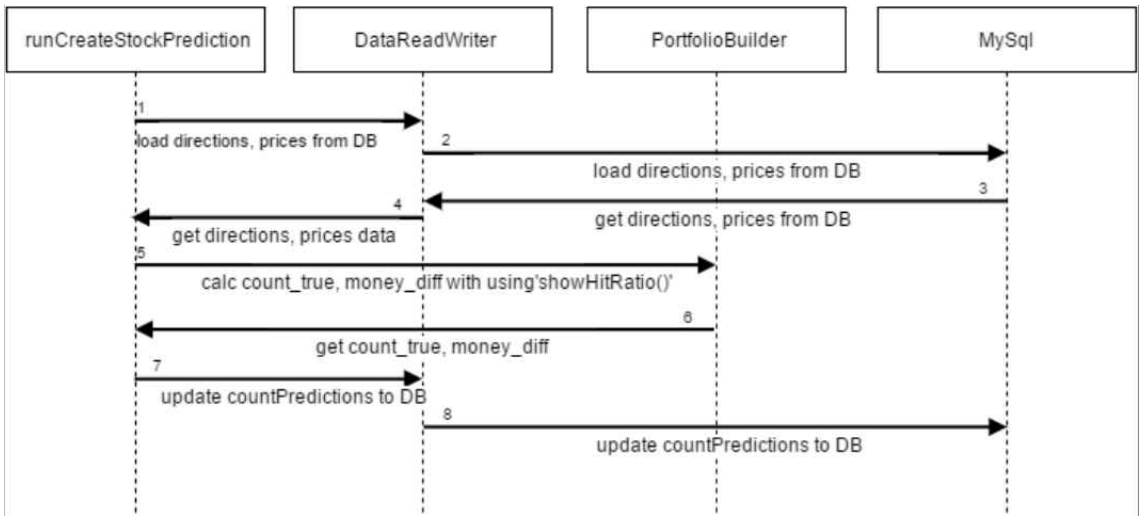
[fig c.1] 상장회사의 회사정보를 Koscom사이트에서 가져와 DB에 저장



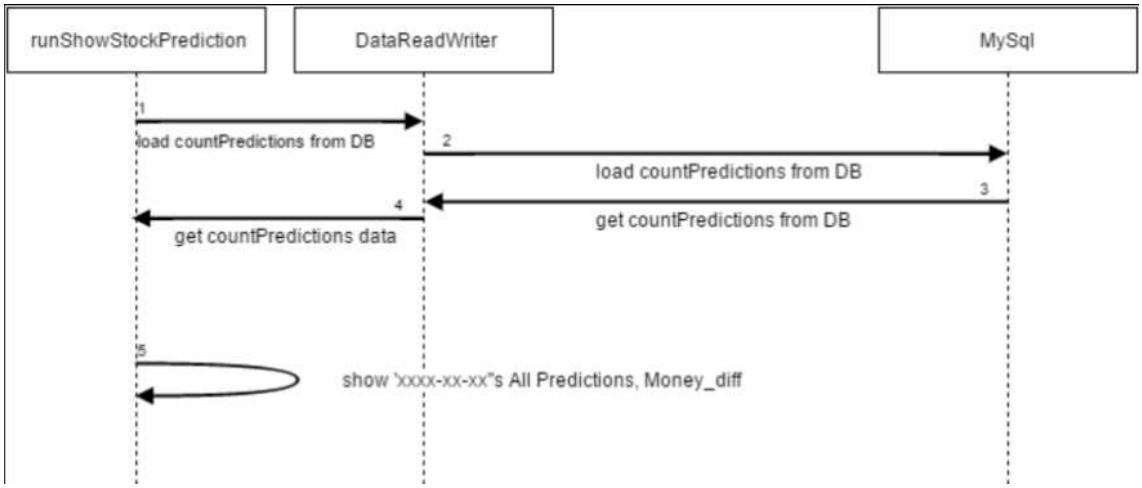
[fig c.2] 주식데이터를 Yahoo Finance사이트에서 가져와 DB에 저장



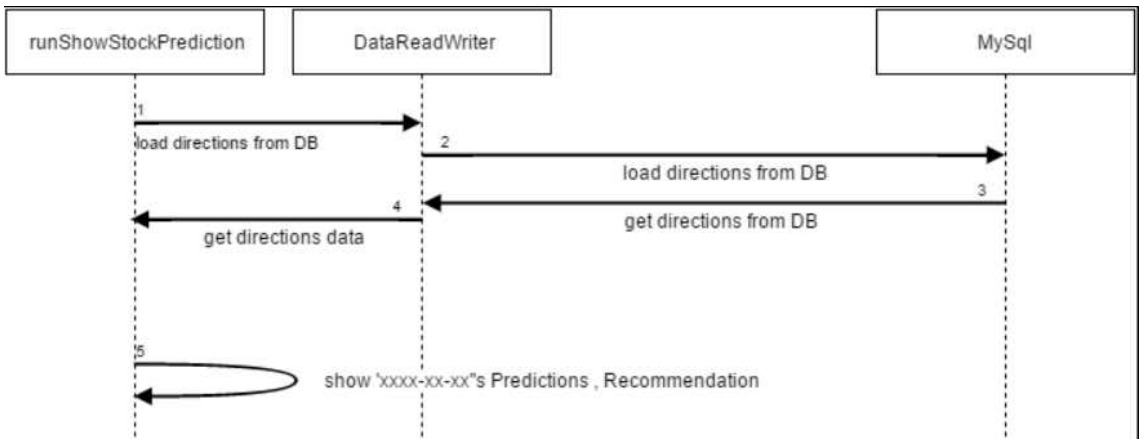
[fig c.3] 주식데이터의 평균회귀(Mean reversion) 성향을 파악하여 순위를 매긴 뒤, 순위 상위권 주식데이터의 동향만 DB에 저장



[fig c.4] 동향과 주식데이터를 이용해 HalfLife 평균치의 반인 3주 뒤로 예측적중횟수와 모의투자수익을 계산해서 결과를 DB에 저장.



[fig c.5] 예측결과를 가져와 통계를 보여준다.



[fig c.6] 특정 날짜의 동향을 보여주고, 주식을 추천해준다.

IV. 연구 결과 및 가능성 모색

IV. 1. 연구 결과 및 평가

- 사용자가 짠 투자전략(investment strategy)
평균회귀(Mean reversion) 성향이 강한(상위 10%)의 주식만 투자진행
동향예측 1회당 100,000원 일괄 투자
사용자가 입력한 기간(35일, 9일) 뒤의 주식데이터로 예측적중률(correct_ratio), 순 이익률을 평가

- 모의 투자 결과 표

16-01-04~	correct_ ratio	count_ true	count_ all	benefit_ ratio	money_ diff	money_ used
16-01-22 (35d)	55.814%	120	215	101.0%	216,655	21,500,000
16-02-16 (35d)	56.757%	231	407	102.2%	907,326	40,700,000
16-03-08 (35d)	52.264%	277	530	101.3%	677,690	53,000,000
16-01-22 (9d)	53.814%	127	236	101.1%	267,335	23,600,000
16-02-16 (9d)	57.385%	237	413	101.6%	660,788	41,300,000
16-03-08 (9d)	52.857%	296	560	100.9%	505,997	56,000,000

실제 주가 방향 적중률(correct_ratio)이 55% 수준으로 볼 때 이득을 볼 수 있을 것이라 예상이 가지만, 실제 투자금 반환률이 101% 수준으로 나타나므로 현재의 투자전략(investment strategy)으로 이득을 볼 수 없다.

IV. 2. 가능성 모색

연구 결과를 보듯이 평균회귀(Mean reversion)이론을 통한 미래예측은 가능할 지라도, 동향예측 1회당 동일한 금액 100,000원을 투자했을 때의 총 이익률은 0%에 가깝다. 시간관계상 이익을 내지 못하는 이유로 추측만 가능하였다. 추측하는 것은 총 3가지이다.

첫째, 프로그램 구현상 동향예측은 하되, 동향의 상승/하강 정도를 예상하고 있진 않기 때문에, 정도가 기입되지 않은 단순한 투자전략이었으므로 이익을 내지 못했다고 추측한다.

둘째, 모의투자에서 결과를 파악하는 기간설정을 일괄적으로 적용시켰기 때문에, 개별적 주식 특성을 반영하지 못했다. 주식데이터 고유의 Half_life값을 활용해 자동적으로 기간설정을 해줬다면 더 나은 결과를 냈을 것이라 추측된다.

마지막으로, 학계에서 말하길 주식투자에서 평균회귀(Mean reversion)와 역방향 접근을 너무 장기적인 관점에서 접근하면 그 효율성이 떨어지는데, 그 이유는 장기적으로는 추세(trend) 라는 것이 생길 수 있어서, 평균회귀(Mean reversion)와 역방향 접근에 필요한 통계학적 요소의 일부가 깨어지기 때문이라고 한다. 따라서 평균회귀(Mean reversion) 이론을 바탕으로 한 투자전략(investment strategy)에서는 매수 후 매도까지의 기간설정을 너무 길게 설정해서는 안 된다고 보인다.

따라서 평균회귀(Mean reversion)성향을 토대로 동향을 예측하는 것은 가능하므로, 프로그램 사용자가 추측대로 문제점을 해결한 주식 투자 전략(investment strategy)을 잘 설정한다면 이윤창출이 가능할 것이라 생각된다.

V. 결론 : 기대 효과

IV. 1. 평균회귀 성향으로 동향 예측은 가능하나 이를 통한 이윤창출을 위해선 더 깊은 연구가 필요

- 앞서 IV. 2. 가능성 모색에서 말했듯이, 본 연구에서 사용한 ADF, Hurst Exponent, Half-Life 값들로 주식의 동향 예측은 55%정도의 성취로 가능함을 보이고 있다. 하지만 투자금 반환율 결과 값으로 인해 실제 투자에 사용할 만큼의 성취도를 얻어내진 못했음이 결과에서 여실히 드러났다. 따라서 연구를 진행하면서 간과했던 모의투자의 기간설정 부분과 동향 예측의 강도 반영이 결과 값에 영향을 미칠 것이라 추측되었다. 하지만 반환률을 101%로 보여줬더라도 동향 예측률은 55%을 보여준 것은 명백한 사실이므로, 이 연구의 의의는 평균회기 이론으로 특정 주식데이터의 동향을 예측하는 것이 가능함을 보인데 있다.

VI. 참조 문헌

- MeanReversionTest

Michael Halls-Moore. Basics of Statistical Mean Reversion Testing. October 21st, 2013
https://www.quantstart.com/articles/Basics-of-Statistical-Mean-Reversion-Testing

- Augmented Dickey-Fuller Test

anonymous
https://en.wikipedia.org/wiki/Augmented_Dickey%E2%80%93Fuller_test
조성일, 최종수, "몬테 카를로 실험에 의한 Augmented Dickey-Fuller 단위근 검정법의 검정력에 관한 연구"
http://kostat.go.kr/attach/journal/10-1-8.pdf

- Hurst exponent

Ian Kaplan. May 2003 Revised: May 2013
http://www.bearcave.com/misl/misl_tech/wavelets/hurst/
Peter Ponzo, "This tutorial written and reproduced with permission". May 29th, 2012
http://www.stator-afm.com/hurst-exponent/

- Half-life

http://www.investinganswers.com/financial-dictionary/debt-bankruptcy/half-life-6199

- Ornstein–Uhlenbeck_process

Ross A. Maller, Gernot M¨uller, and Alex Szimayer "Ornstein-Uhlenbeck Processes and Extensions"
http://mediatum.ub.tum.de/doc/1072652/1072652.pdf

- 내 프로젝트 Git

https://github.com/AnJaeSeongS2/StockRecommendationML