



Predicting stock prices
based on people's reactions.

201600779 김영민

Context



01 Motivation

02 Introduce My Project

03 Flow Chart

04 Method of Data Collection

05 Semantic Analysis

06 Result(Samsung, Naver, Celltrion)

07 Conclusion

Motivation

어론 속의 여론

주식 투자자 43% "코로나 이
후 시작"... 92% "계속할 것"

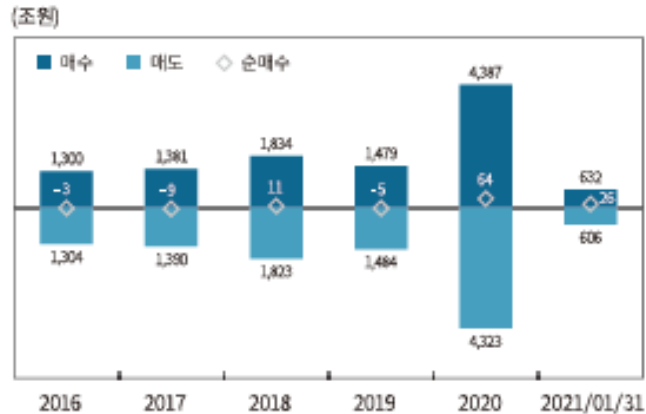
♡ 1 ○ 0

주식 직접투자 행태 및 투자자산 인식 조사

업력 2021.05.06 04:30



〈그림 1〉 개인투자자 거래규모



코로나 이후 개인 투자자 비율

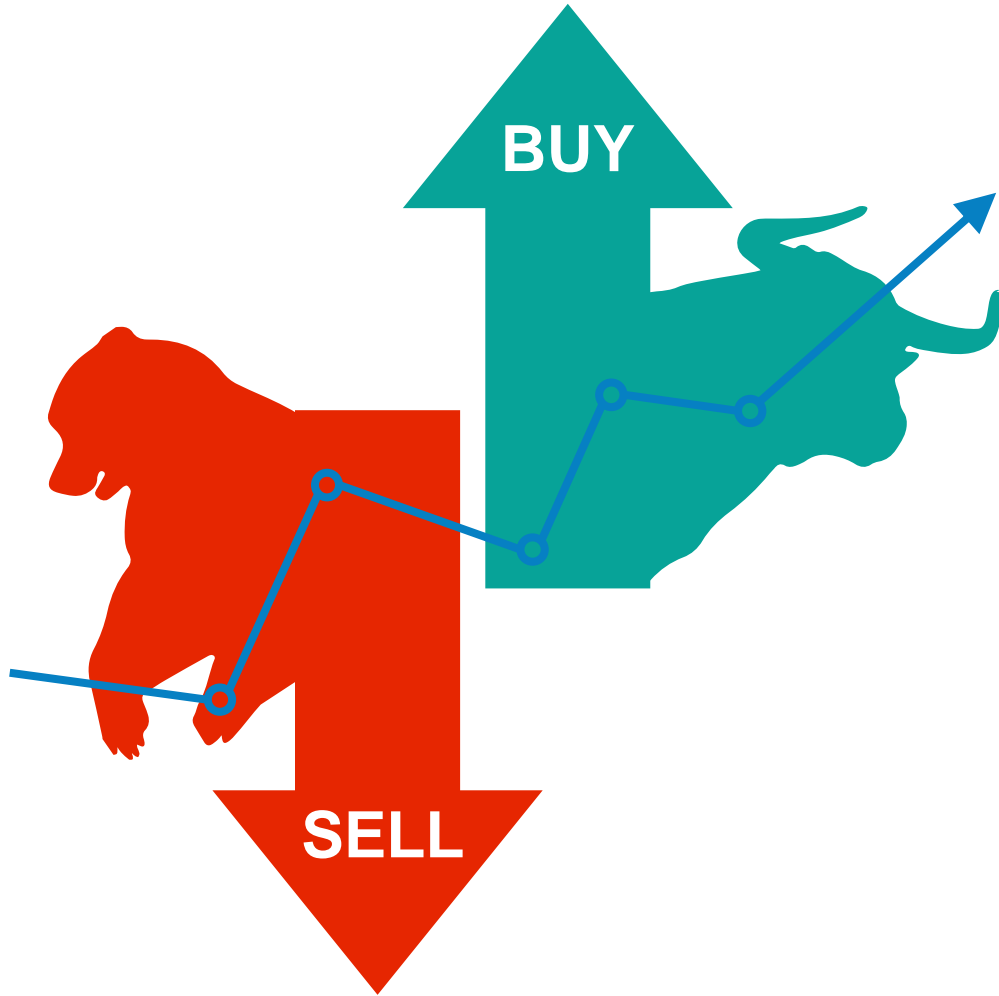
Then



사람들의 기대 심리가
주가에 영향을 미칠까???

개인들이 가장 많이 접하게 되는
회사 소식인 기사들에 의해
주가 변동에 영향이 있을까?

Introduce My Project



Is **Individual reaction** related to **stock prices**?

종목

- ❖ 삼성전자
- ❖ 네이버
- ❖ 셀트리온

기간

- ❖ Train : 2021.01~2021.10.30
- ❖ Test : 2021.11.1~2021.11.15

수집 데이터

- ❖ Investing.com의 여론 데이터
- ❖ 네이버 기사 데이터
- ❖ 시작가와 종가 데이터

Prediction Model

- ❖ RandomForest Classification
- ❖ LGBM Classification
- ❖ LSTM or GRU

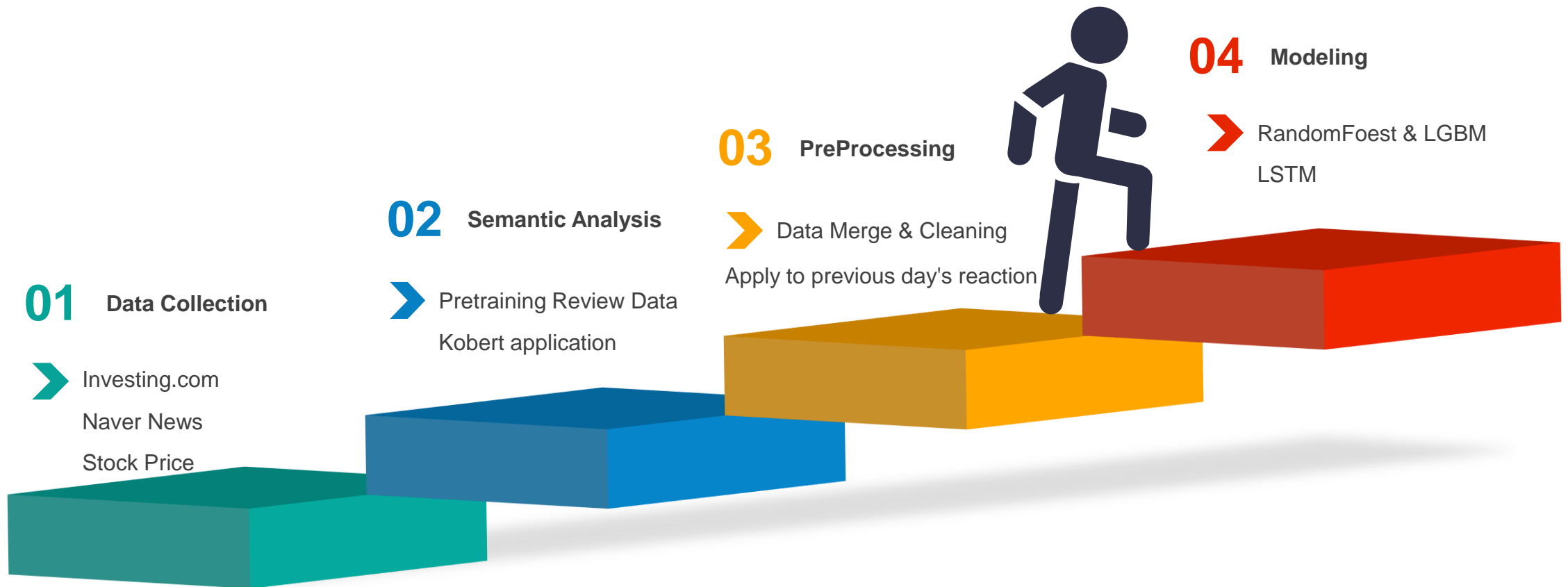
Semantic Analysis

- ❖ KoBERT

발전 방향

- ❖ 코로나 전후 1년 비교
- ❖ 딥러닝을 활용한 감정 분석
- ❖ 여러 사이트를 활용한 감정 분석

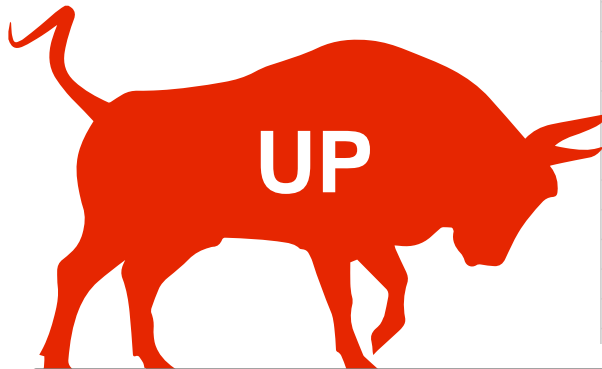
Flow Chart



Investing.com Opinion



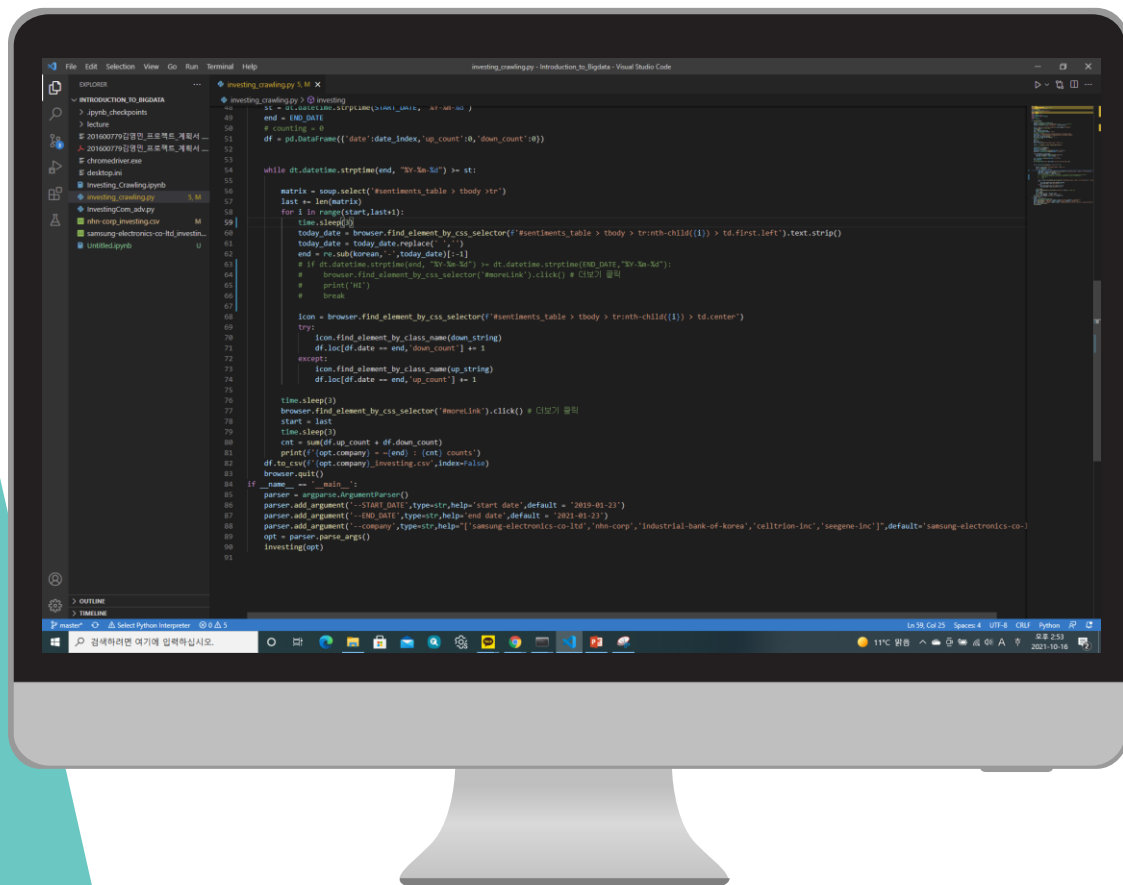
전망에 빨간 소가
있으면 상승 예측



전망에 **파란** **곰**이
있으면 하락 예측



Crawling



[https://github.com/winston1214/INU/blob/master/Introduction to Bigdata/investing_crawling.py](https://github.com/winston1214/INU/blob/master/Introduction%20to%20Bigdata/investing_crawling.py)

일반 BeautifulSoup 로는 접근이 불가능 하기 때문에
Selenium을 이용하여 사이트 접근

BeautifulSoup

BeautifulSoup를 사용하여 원하는 정보 추출

	date	up_count	down_count
0	2021-10-31	0	0
1	2021-10-30	0	0
2	2021-10-29	1	3
3	2021-10-28	7	0
4	2021-10-27	1	2
...
299	2021-01-05	22	5
300	2021-01-04	19	10
301	2021-01-03	0	0
302	2021-01-02	0	0
303	2021-01-01	1	0

Stock Price

주가 데이터 다운로드

005930 역사적 데이터

기간:

일간

데이터 다운로드

2021/09/16 - 2021/10/16

i

날짜	종가	오픈	고가	저가	거래량	변동 %
2021년 10월 15일	70,100	70,200	71,000	70,000	17.53M	1.01%
2021년 10월 14일	69,400	69,100	69,800	68,800	0.23K	0.87%
2021년 10월 13일	68,800	68,700	69,600	68,400	0.57K	-0.29%
2021년 10월 12일	69,000	70,800	70,900	68,700	18.42K	-3.50%
2021년 10월 11일	71,500	71,500	71,500	71,500	-	0.00%
2021년 10월 08일	71,500	72,300	72,400	71,500	13.97M	-0.14%
2021년 10월 07일	71,600	71,800	72,100	71,300	0.17K	0.42%
2021년 10월 06일	71,300	72,700	72,800	71,200	8.34K	-1.25%
2021년 10월 05일	72,200	72,900	73,000	71,400	1.12K	-1.37%
2021년 10월 04일	73,200	73,200	73,200	73,200	-	0.00%
2021년 10월 01일	73,200	73,900	74,000	72,900	15.70M	-1.21%
2021년 09월 30일	74,100	74,300	74,800	73,800	6.44K	0.00%
2021년 09월 29일	74,100	74,800	75,300	73,800	15.27K	-2.88%
2021년 09월 28일	76,300	77,600	77,600	76,200	0.94K	-1.80%
2021년 09월 27일	77,700	77,300	77,700	77,000	11.63M	0.52%
2021년 09월 26일	77,300	77,300	77,300	77,300	-	0.00%
2021년 09월 24일	77,300	77,600	77,700	77,100	11.85M	-0.13%
2021년 09월 23일	77,400	77,500	77,600	76,900	0.49K	0.26%
2021년 09월 22일	77,200	77,200	77,200	77,200	-	0.00%
2021년 09월 17일	77,200	76,300	77,200	75,900	15.66M	1.45%
2021년 09월 16일	76,100	77,300	77,400	76,100	12.88M	-1.17%
최고: 77,700	최저: 68,400	차이: 9,300	평균: 73,643	변동 %: -9		

데이터 다운로드가 한 달
단위로 제공되기 때문에
데이터 결합 필요



```
import pandas as pd
import os
ls = os.listdir('samsung')
df_list = []
for i in ls:
    data = pd.read_csv('samsung/'+i)
    df_list.append(data)
df = pd.concat(df_list).reset_index(drop=True)
```

	날짜	종가	오픈	고가	저가	거래량	변동 %
0	2021년 02월 26일	82,500	82,800	83,400	82,000	36.72M	-3.28%
1	2021년 02월 25일	85,300	84,000	85,400	83,000	6.35K	4.02%
2	2021년 02월 24일	82,000	81,900	83,600	81,400	0.98K	0.00%
3	2021년 02월 23일	82,000	81,300	82,900	81,100	7.03K	-0.24%
4	2021년 02월 22일	82,200	83,800	84,200	82,200	17.48K	-0.48%
...
230	2021년 09월 06일	77,300	76,800	77,600	76,600	3.42K	0.91%
231	2021년 09월 05일	76,600	76,600	76,600	76,600	-	0.00%
232	2021년 09월 03일	76,600	76,400	76,700	76,000	11.89M	0.79%
233	2021년 09월 02일	76,000	76,800	76,800	75,700	1.76K	-1.04%
234	2021년 09월 01일	76,800	76,600	77,100	75,900	0.84K	0.13%

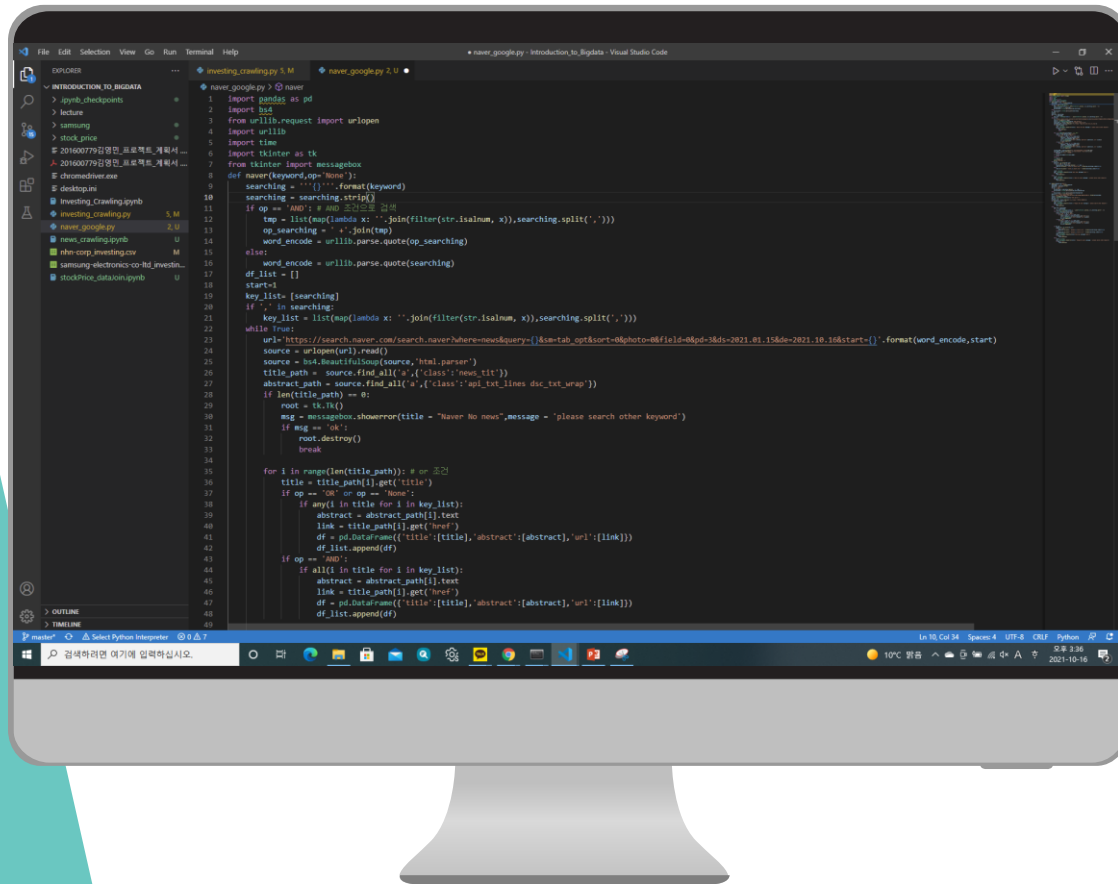
235 rows × 7 columns

By investing.com

Naver News Crawling

BeautifulSoup

BeautifulSoup를 사용하여 원하는 정보 추출



https://github.com/winston1214/project/tree/master/Keyword_Search_News_Crawling_Machine

title	abstract	url	date	engine
"20일 삼성	삼성전자의	http://mon	2021.10.15	naver
"삼성전자	개미(개인	https://www	2021.10.16	naver
떨어지는	추가, 연고	http://yna	2021.10.14	naver
삼성전자	가조 바이든	http://new	2021.10.16	naver
삼성전자,	삼성전자는	http://www	2021.10.15	naver
삼성전자,	기사내용	http://www	2021.10.15	naver
삼성전자,	삼성전자는	http://www	2021.10.15	naver
[특징주]	삼성전자	http://yna	2021.10.15	naver
삼성전자,	반도체	https://biz	2021.10.15	naver
삼성전자	국내 코스	http://new	2021.10.15	naver
[특징주]	삼성전자	http://yna	2021.10.15	naver
[속보]	삼성전자	http://www	2021.10.15	naver
7만전자	삼성전자	http://www	2021.10.15	naver
위기의 삼,	최근 6만	https://pre	2021.10.16	naver
반동장 개	최근 코스	https://vie	2021.10.15	naver
삼성전자	[삼성전자	http://yna	2021.10.14	naver
삼성전자,	기사내용	http://www	2021.10.12	naver
삼성전자	삼성전자의	http://www	2021.10.14	naver
출매만 5	분삼성전자	https://www	2021.10.16	naver
대장주 삼,	코스피 시	https://www	2021.10.15	naver
삼성전자,	목표가 연	http://www	2021.10.14	naver
"국민주	여삼성전자의	http://new	2021.10.13	naver
삼성전자	삼성전자의	http://new	2021.10.15	naver
삼성전자,	삼성전자	https://vie	2021.10.13	naver
삼성전자	!기사내용	http://www	2021.10.12	naver
삼성전자,	삼성전자의	http://new	2021.10.15	naver
"동학개미	15일 오후	http://www	2021.10.15	naver
삼성전자	삼성전자	http://www	2021.10.12	naver
미국발	훈:6만전자	https://www	2021.10.15	naver
코스피 30	국내 증시	https://www	2021.10.15	naver
"6만전자	대장주 삼	http://www	2021.10.13	naver
코스피 3,0	2,900선으	https://lvr	2021.10.15	naver
미워도 삼,	코스피 시	https://www	2021.10.15	naver
코스피 0.8	구스피 대	https://www	2021.10.15	naver

crawling machine

Engine

☐ Naver☐ Google

Operator

☐ AND☐ OR☐ None

If you choice AND or OR
Please separate the two words by ','(comma)

Search

Naver News Crawling

보도자료 + 지면기사 중심으로

Trouble



날짜가 아닌 다른 값이
추출 되는 상황 발생

Solution

```
13 return date
14 def all_site_crawling(url_ls):
15     df = pd.DataFrame(columns = ['url', 'date'])
16     df['url'] = url_ls
17     fix_date = []
18     for idx, i in enumerate(tqdm(url_ls)):
19         site = i.split('/')[2]
20         news = i
21         try:
22             source = urlopen(i).read()
23             source = bs4.BeautifulSoup(source, 'html.parser')
24         except:
25             pass
26         korean = re.compile('[\u3131-\u3163\uac00-\ud7a3]*')
27         if site == 'news.mk.co.kr': # 크롬링만아 않
28             time.sleep(4)
29             fix_date.append(source.find_all('li', {'class': 'lasttime'})[0].text.split()[-2])
30         elif site == 'www.hankyung.com': # 고화
31             date = number2date(news)
32             fix_date.append(date)
33         elif site == 'www.sedaily.com': # 크롬링만아 않
34             time.sleep(4)
35             da = source.find_all('span', {'class': 'url_txt'})[0].text
36             date = re.sub(korean, '', da.split(' ')[0])
37             fix_date.append(date.replace('.', ''))
38         elif site == 'www.etnews.com': # 고화
39             date = number2date(news)
40             fix_date.append(date)
41         elif site == 'news.heraldcorp.com': # 고화
42             date = number2date(news)
43             fix_date.append(date)
44         elif site == 'view.asiae.co.kr': # 고화
45             date = number2date(news)
46             fix_date.append(date)
47         elif site == 'www.fnnews.com': # 고화
48             number = news.split('/')[1]
49             date = number[:4]+'-'+number[4:6]+'-'+number[6:8]
50             fix_date.append(date)
51         elif site == 'news.kmib.co.kr': # 크롬링만아 않
```

뉴스 사이트별 날짜 추출
코드 개발

Semantic Analysis

Naver Review Pretrained Weight 사용 (Environment : Ubuntu 18.04.5 LTS, GPU : V100 32GB)

	id	document	label
0	9976970	아 더빙.. 진짜 짜증나네요 목소리	0
1	3819312	흠...포스터보고 초딩영화줄...오버연기조차 가법지 않구나	1
2	10265843	너무재밌었다그래서보는것을추천한다	0
3	9045019	교도소 이야기구먼..솔직히 재미는 없다.평점 조정	0
4	6483659	사이몬페그의 익살스런 연기가 돋보였던 영화스파이더맨에서 킬어보이만 했던 커스틴 ...	1

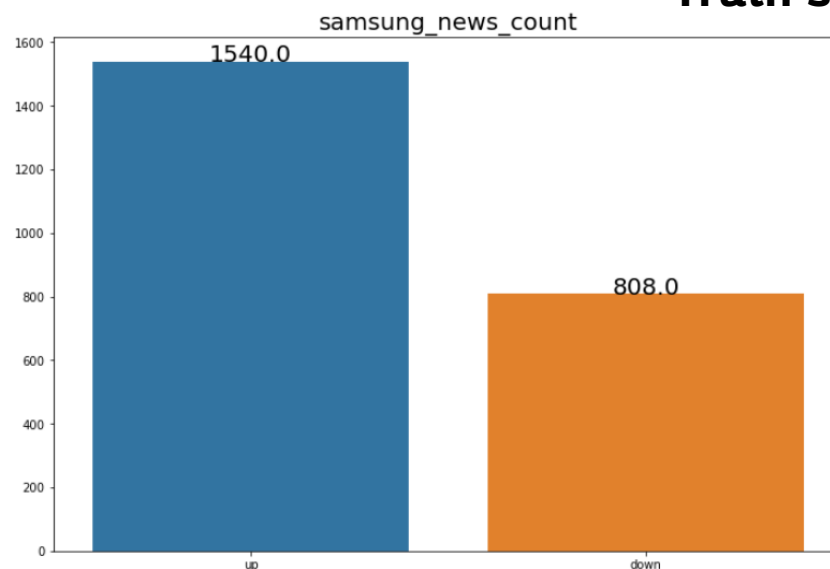
Label : 0 = 부정
Label : 1 = 긍정

```
epoch 4 batch id 1 loss 0.4208475351333618 train acc 0.859375
epoch 4 batch id 201 loss 0.027972815558314323 train acc 0.955146144278607
epoch 4 batch id 401 loss 0.07473891228437424 train acc 0.9570604738154613
epoch 4 batch id 601 loss 0.19486865401268005 train acc 0.9588446339434277
epoch 4 batch id 801 loss 0.14141656458377838 train acc 0.9608107053682896
epoch 4 batch id 1001 loss 0.1097184345126152 train acc 0.9620223526473527
epoch 4 batch id 1201 loss 0.05550093576312065 train acc 0.9632727935054122
epoch 4 batch id 1401 loss 0.02658044546842575 train acc 0.9644115810135617
epoch 4 batch id 1601 loss 0.22273029386997223 train acc 0.9650804184884447
epoch 4 batch id 1801 loss 0.08296814560890198 train acc 0.9658002498611882
epoch 4 batch id 2001 loss 0.04506051912903786 train acc 0.9664386556721639
epoch 4 batch id 2201 loss 0.05467686802148819 train acc 0.9670320308950477
epoch 4 train acc 0.9674701365187713
0%|          | 0/782 [00:00<?, ?it/s]
epoch 4 test acc 0.8984574808184144
0%|          | 0/2344 [00:00<?, ?it/s]
```

Epoch : 4 훈련 결과

	title	date	label
0	삼성전자, 국내 최대 20kg 용량 비스포크 그랑데 건조기 AI 선배	2021.10.28.	1.0
1	SK매직, '삼성전자'와 가전 렌탈서비스 전국 SKT 매장 확대 판매,	2021.10.28.	1.0
2	삼성전자, '삼성 개발자 콘퍼런스 2021' 개최	2021.10.27.	1.0
3	삼성전자, 세계 3분기 스마트폰 판매량 1위..."4분기 애플이 선두로"	2021.10.30.	1.0
4	SKT, 삼성전자와 손잡고 생활 가전 '렌탈 구독 서비스' 시행	2021.10.28.	1.0
...
349	삼성전자, 유럽서 베티올 행사 개최... '비스포크 홈' 제품 소개	2021.10.12.	1.0
350	삼성전자, 원스톱 AI 가전 관리 서비스 '홈케어 매니저' 확대	2021.10.21.	1.0
351	삼성전자, 원스톱 AI 가전 관리 서비스 '홈케어 매니저' 확대	2021.10.21.	1.0
352	삼성전자서비스, 다문화가정 청소년 자립 위한 후원 실시	2021.10.19.	1.0
353	삼성전자, 15년 간 우수 기능인력 양성 산실로 '우뚝'	2021.10.05.	1.0

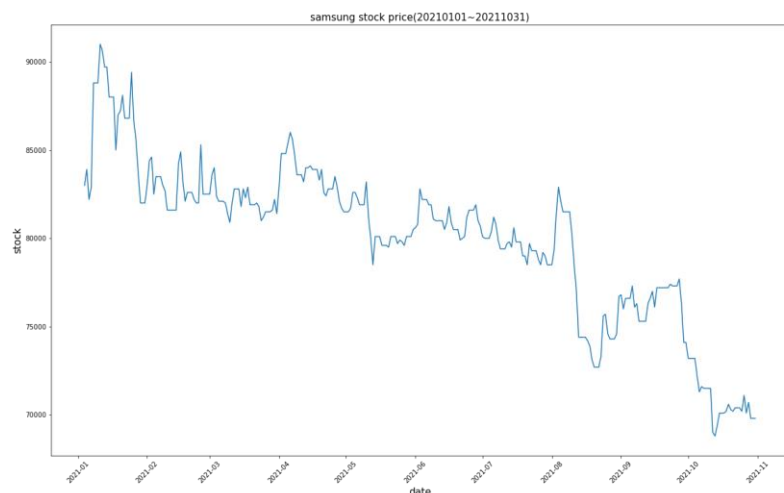
Train score : 0.96



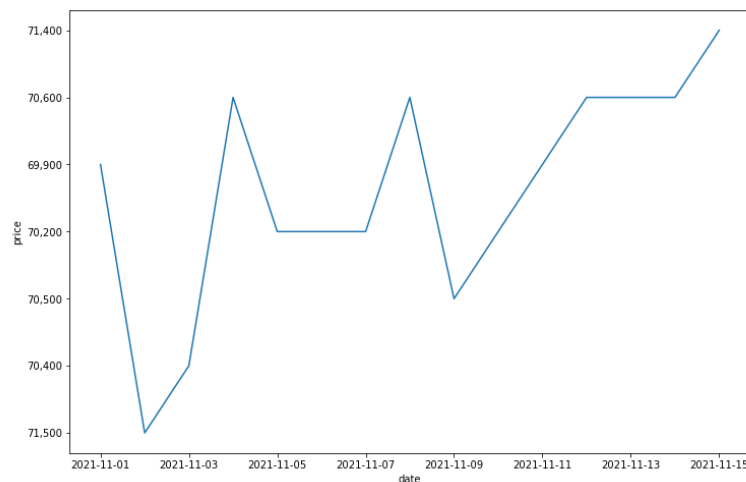
Samsung Price Predict

RandomForest & LGBM (추가만) Label : 0 = 하락, Label : 1 = 상승

Train set



Test set

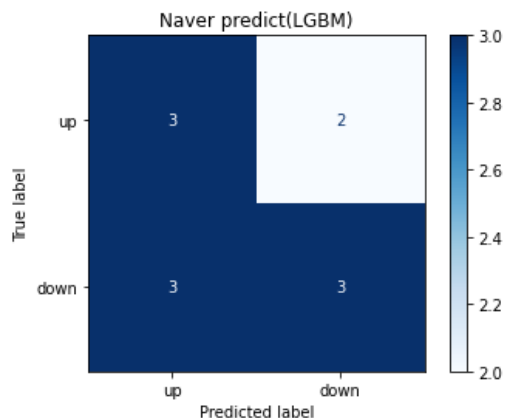
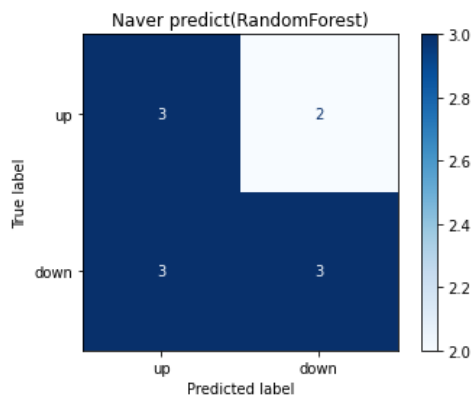


```
: from sklearn.ensemble import RandomForestClassifier  
rf = RandomForestClassifier(random_state = 42)  
rf.fit(train_x.values.reshape(-1,1),train_y)  
naver_pred = rf.predict(test_x.values.reshape(-1,1))  
accuracy_score(test_y,naver_pred)
```

: 0.5454545454545454

```
: lgbm = LGBMClassifier(random_state = 42)  
lgbm.fit(train_x.values.reshape(-1,1),train_y)  
naver_pred = lgbm.predict(test_x.values.reshape(-1,1))  
accuracy_score(test_y,naver_pred)
```

: 0.5454545454545454

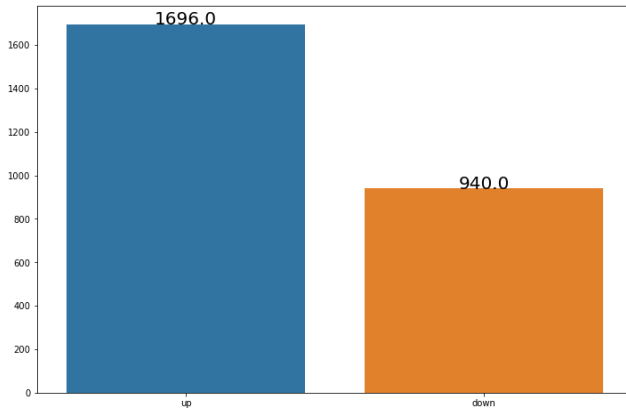


정확도 : 54.54%

Samsung Price Predict

RandomForest & LGBM (Investing.com 반응 추가)

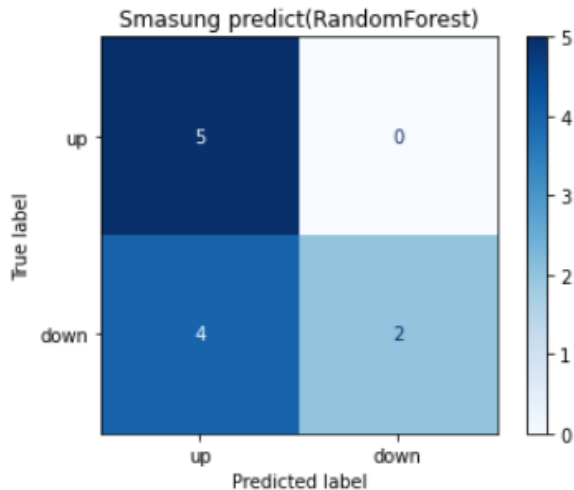
긍정 - 부정 반응개수



	samsung_price	up_count	down_count
0	69800	1	-3
1	70700	7	0
2	70100	1	-2
3	71100	0	-1
4	70200	5	-6
...
243	88800	20	-9
244	82900	13	-13
245	82200	15	-9
246	83900	22	-5
247	83000	19	-10

긍정 = 긍정 반응 개수

부정 = - 부정 반응 개수

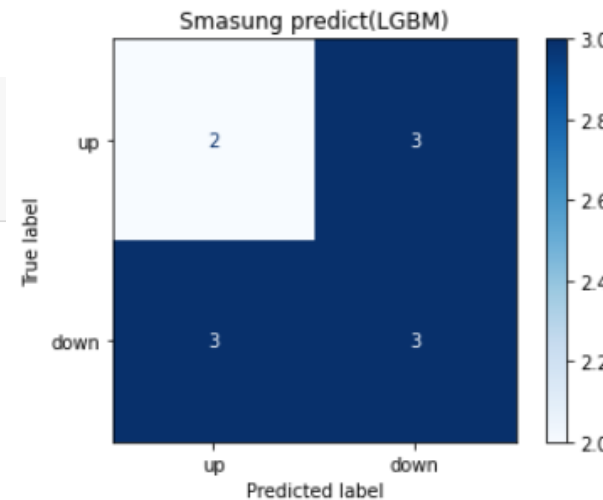


```
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(random_state = 42)
rf.fit(train_x, train_y)
pred = rf.predict(test_x)
print(accuracy_score(test_y, pred))
```

0.6363636363636364



Random Forest
54.54 -> 63.6



```
from lightgbm import LGBMClassifier
lgbm = LGBMClassifier(random_state = 42)
lgbm.fit(train_x, train_y)
pred = lgbm.predict(test_x)
print(accuracy_score(test_y, pred))
```

0.45454545454545453

LGBM
54.54 -> 45.45%

Samsung Price Predict

RandomForest & LGBM (Preprocessing 추가)

모델	전처리	변수	정확도
RandomForest	Score = up-down	Price, score	24.24%
LGBM			54.55%
RandomForest	Standard Scaling(Price)	Price, up_count, -down_count	72.73%
LGBM			54.55%
RandomForest		Price	36.36%
LGBM			72.73%
RandomFoest	Standard Scaling(Price) Score = up-down	Price, up_count, -down_count	72.73%
LGBM			63.64%

Samsung Price Predict

RandomForest & LGBM (Naver news 반응 추가)

1. News 데이터와 Investing 데이터 컬럼 구분

RandomForest : **45.45%** LGBM: **45.45%**

	date	samsung_price	samsung_change	up_count	down_count	news_up	news_down
0	2021-10-29	69800	-1.27	1	-3	5.0	-2.0
1	2021-10-28	70700	0.86	7	0	12.0	-10.0
2	2021-10-27	70100	-1.41	1	-2	8.0	-2.0
3	2021-10-26	71100	1.28	0	-1	4.0	-5.0
4	2021-10-25	70200	-0.28	5	-6	2.0	-7.0
...
234	2021-01-11	91000	2.48	73	-18	3.0	-0.0
236	2021-01-08	88800	7.12	20	-9	0.0	-1.0
237	2021-01-06	82200	-2.03	15	-9	0.0	-1.0
238	2021-01-05	83900	1.08	22	-5	2.0	-0.0
239	2021-01-04	83000	2.47	19	-10	2.0	-0.0

2. News 데이터와 Investing 데이터 컬럼 구분 X

RandomForest : **36.36%** LGBM: **45.45%**

	date	samsung_price	samsung_change	up	down
0	2021-10-29	69800	-1.27	6.0	-5.0
1	2021-10-28	70700	0.86	19.0	-10.0
2	2021-10-27	70100	-1.41	9.0	-4.0
3	2021-10-26	71100	1.28	4.0	-6.0
4	2021-10-25	70200	-0.28	7.0	-13.0
...
234	2021-01-11	91000	2.48	76.0	-18.0
236	2021-01-08	88800	7.12	20.0	-10.0
237	2021-01-06	82200	-2.03	15.0	-10.0
238	2021-01-05	83900	1.08	24.0	-5.0
239	2021-01-04	83000	2.47	21.0	-10.0

3. 컬럼 구분 + 가격 정규화

RandomForest : **100%** LGBM: **86.1%**

4. 컬럼 구분 X + 가격 정규화

RandomForest : **100%** LGBM: **86.1%**

주식의 종가를 정규화한 후 뉴스의 반응을 추가하였을 때,
더욱 유의미한 결과 도출

Samsung Price Predict

LSTM(Investing 반응 + News 반응)

```
from keras.models import Sequential
from keras.layers import Dense
from keras.callbacks import EarlyStopping, ModelCheckpoint
from keras.layers import LSTM

model = Sequential()
model.add(LSTM(16,
              input_shape=(train_feature.shape[1], train_feature.shape[2]),
              activation='relu',
              return_sequences=False)
          )
model.add(Dense(1))
```

Loss : RMSE, Optimizer = Adam, Epoch = 100

1. News 데이터와 Investing 데이터 컬럼 구분
(추가 MinMax)

Accuracy : 60%

2. News 데이터와 Investing 데이터 컬럼 구분 X
(추가 MinMax)

Accuracy : 40%

3. 추가만 (추가 Standard Scaling) Accuracy : 70%

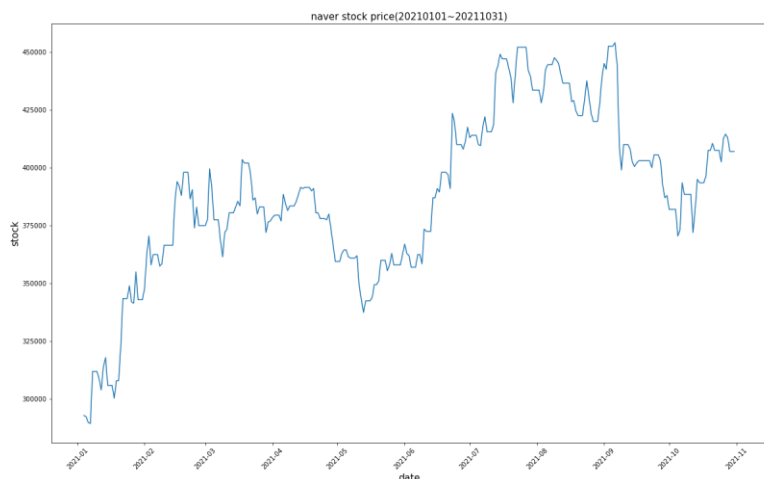
4. 추가만 (추가 MinMax) Accuracy : 70%

기존의 최고 성능을 뛰어넘지 않음

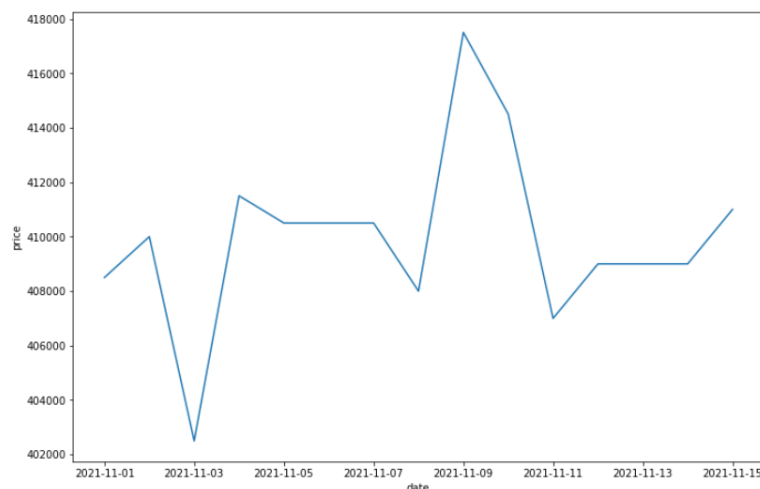
Naver Price Predict

RandomForest & LGBM (추가만)

Train set



Test set



Preprocessing X

```
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(random_state = 42)
rf.fit(train_x.values.reshape(-1,1),train_y)
naver_pred = rf.predict(test_x.values.reshape(-1,1))
accuracy_score(test_y,naver_pred)
```

0.5454545454545454

```
lgbm = LGBMClassifier(random_state = 42)
lgbm.fit(train_x.values.reshape(-1,1),train_y)
naver_pred = lgbm.predict(test_x.values.reshape(-1,1))
accuracy_score(test_y,naver_pred)
```

0.5454545454545454

After, StandardScaling

```
rf = RandomForestClassifier(random_state = 42)
rf.fit(scale_train_x,train_y)
accuracy_score(test_y,rf.predict(scale_test_x))
```

0.7272727272727273

→ 72.7%

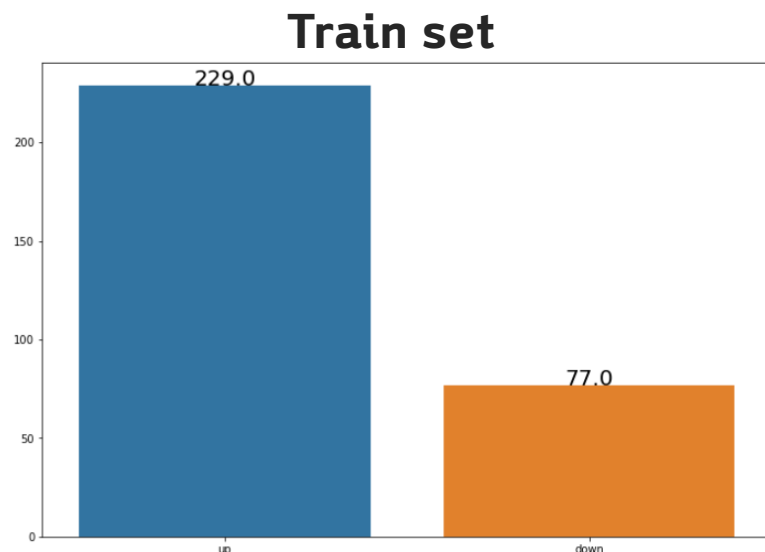
```
lgbm = LGBMClassifier(random_state = 42)
lgbm.fit(scale_train_x,train_y)
accuracy_score(test_y,lgbm.predict(scale_test_x))
```

0.9090909090909091

→ 90.9%

Naver Price Predict

RandomForest & LGBM (Investing 반응 추가)



Test set

해당 기간 내 반응이 적음
(기간 총 반응 : 긍정2, 부정0)

• • • 유의미한 변수가 되지 못함

RandomForest 63.6%

LGBM 54.5%

↓
변화 X

After, StandardScaling

```
rf.fit(scale_train_x,train_y)
accuracy_score(test_y,rf.predict(scale_test_x))
```

0.5454545454545454

```
lgbm.fit(scale_train_x,train_y)
accuracy_score(test_y,lgbm.predict(scale_test_x))
```

0.5454545454545454

→ 무의미

Naver Price Predict

RandomForest & LGBM (Naver news 반응 추가)

1. 컬럼 구분 + 가격 정규화(Standard)

2. 컬럼 구분 X + 가격 정규화(Standard)

RandomForest : 54.54% LGBM: 63.63% LSTM: 50% RandomForest : 36.36% LGBM: 54.54% LSTM: 50%

3. 컬럼 구분 + 가격 정규화(MinMax)

4. 컬럼 구분 X + 가격 정규화(MinMax)

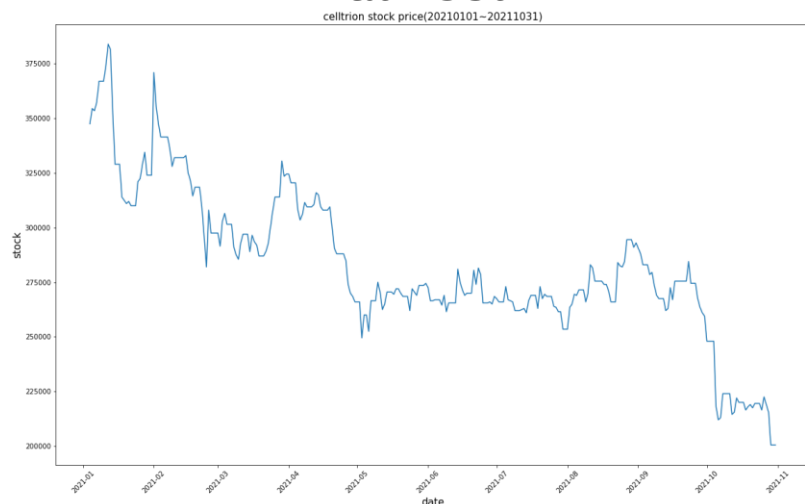
RandomForest : 54.54% LGBM: 63.63% LSTM: 60% RandomForest : 72.72% LGBM: 81.81% LSTM: 50%

주식의 종가를 정규화한 후 **종가만으로 예측했을 때**
가장 좋은 예측값을 냄(반응은 유의미하지 않음)

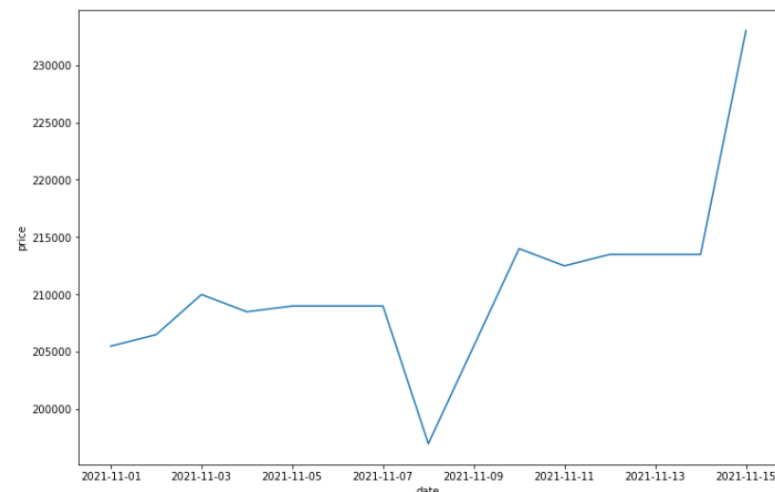
Celltrion Price Predict

RandomForest & LGBM (추가만)

Train set



Test set



After, StandardScaling

RandomForest	➔	18.18%
LGBM	➔	27.27%
LSTM	➔	30%

Celltrion Price Predict

RandomForest & LGBM (investing 반응 + news 반응)

1. Investing 반응만(정규화)

RandomForest : **36.36%** LGBM: **27.27%** LSTM: **30%**

2. Investing + 뉴스 반응 컬럼 구분(정규화)

RandomForest : **36.36%** LGBM: **36.36%** LSTM: **30%**

3. Investing + 뉴스 반응 컬럼 구분 X(정규화)

RandomForest : **36.36%** LGBM: **45.45%** LSTM: **30%**

Test 시점이 **기존과 다른 양상**이기 때문에 예측의 **정확도 떨어짐**으로 추정

Conclusion & Future Work

✓ Conclusion

1. 주가와 사람들 기대 심리의 관계성

- 주식 종목마다 다르지만 예측력을 높이는데 영향을 줌
- 뉴스 데이터로 인한 긍-부정적 사실이 추가되었을 때, 예측력이 높아짐

2. 몇몇 종목의 예측 실패 이유

- 해당 주식에 대한 사람들의 반응 데이터가 부족
- Test 시기가 짧음(11일)

3. 수치값의 정규화

- 대부분의 경우 종가의 정규화 이후, 모델의 정확도가 향상됨



✓ Future Work

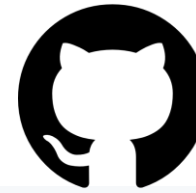
1. 시계열 모델의 고도화

- ARIMA, FAVAR 등의 통계적 시계열 모델을 적용 비교
- Bi-LSTM, SCI-Net(SOTA 2021), Transformer 등의 시계열에 고도화된 모델 적용

2. 더 많은 데이터의 수집

- 수치화된 기대심리가 아닌 사용자의 텍스트 기반 의견을 수집하여 더 많은 데이터를 수집하여 고도화
- Naver News 뿐만 아니라 증권 뉴스 등을 더 많이 수집

Github



[winston1214 / Stock-Prediction](#) Public

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

master 1 branch 0 tags

[Go to file](#) [Add file](#) [Code](#)

[About](#)

winston1214	Create README.md	dc79faa	1 minute ago	7 commits
PT	data		23 minutes ago	
bert_data	data		23 minutes ago	
final_data	data		23 minutes ago	
investing	data		23 minutes ago	
lecture	data		23 minutes ago	
merge_data	data		23 minutes ago	
naver_news	data		23 minutes ago	
picture	data		23 minutes ago	
stock_price	data		23 minutes ago	
test_data	data		23 minutes ago	
Final.ipynb	Jupyter file		22 minutes ago	
Investing_Crawling.ipynb	Jupyter file		22 minutes ago	
README.md	Create README.md		1 minute ago	
all_news_data_crawling.py	Crawling.py		21 minutes ago	
bert_test.py	bert commit		19 minutes ago	
bert_train.py	Bert train		19 minutes ago	
investing_crawling.py	Crawling.py		21 minutes ago	
naver_google.py	Crawling.py		21 minutes ago	
news_crawling.ipynb	Jupyter file		22 minutes ago	
preprocessing.ipynb	Jupyter file		22 minutes ago	
stockPrice_dataJoin.ipynb	Jupyter file		22 minutes ago	

README.md

Stock-Prediction

Predicting stock prices based on people's reactions

Description

Based on the data from January 1st to October 31st, 2021, I predicted the fluctuation of stock prices from November 1st to November 15th, 2021.

I collected people's reaction and news. And I conducted an semantic analysis(using KoBERT) with that.

[Readme](#)

Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

Languages

Jupyter Notebook 88.0%

Python 12.0%

<https://github.com/winston1214/Stock-Prediction>

