title.ipynb

```
from donggukUniversity import statistics
print(bigData)
```

비어플

```
print(topic)
```

주식 종료 가격 예측

```
df = data.about.us()
df.iloc[0, :]
```

9기 김기호

```
df.iloc[1, :]
```

10기 최은진

```
df.iloc[2, :]
```

10기 김평진

CONTENTS

01

데이터 소개 & 분석 목적

| 분석 목적 평가 산식, 규칙

- 분석 목적

  21.11.29 ~ 21.12.3일의 주식 종가를 예측하는 모델을 만들고 성능평가.

- 평가 산식: NMAE(Normalized Mean Absolute Error) *100

  스케일(데이터의 범위)가 다른 데이터 세트의 MAE에 관한 비교를 용이하게 하는데 사용.

  정규화의 수단으로, 측정된 데이터의 평균을 이용함.

  낮을수록 실제 값과 유사함

$$NMAE(\mathbf{y}, \hat{\mathbf{y}}) = \frac{MAE(\mathbf{y}, \hat{\mathbf{y}})}{\frac{1}{n}\sum_{i=1}^{n}|y_i|} = \frac{MAE(\mathbf{y}, \hat{\mathbf{y}})}{mean(|\mathbf{y}|)}$$

- 규칙:

  예측 전날까지의 데이터만 사용 가능.

  공공데이터와 같이 누구나 얻을 수 있고 법적 제약이 없는 외부 데이터 허용.

| 데이터 소개 | 평가 산식, 규칙

- Stock_list : 2021년 6월 기준 KOSPI-200내 주식
  + KOSDAQ-150내 주식 총 350종목
- Sample_submission : 결과 제출 파일, 종목코드별 기간 내 종가
- 데이터 사용기간 : 21.01.04 ~ 21.11.26
- 종가, 거래량 등 과거 데이터를 기반으로 추세를 파악할 수 있는 기술적 지표를 활용.

| 데이터 소개 기본 제공 데이터

	A	B	C
1	종목명	종목코드	상장시장
2	삼양홀딩스	70	KOSPI
3	하이트진로	80	KOSPI
4	유한양행	100	KOSPI
5	CJ대한통운	120	KOSPI
6	두산	150	KOSPI
7	DL	210	KOSPI
8	한국앤컴퍼니	240	KOSPI
9	기아	270	KOSPI
10	SK하이닉스	660	KOSPI

Stock_list : (350 obs X 3 variables)

변수명	변수 타입
종목명	str
종목코드	str
상장시장	str

	A	B	C	D	E	F	G	H
1	Day	60	80	100	120	150	240	250
2	2021-11-01	0	0	0	0	0	0	0
3	2021-11-02	0	0	0	0	0	0	0
4	2021-11-03	0	0	0	0	0	0	0
5	2021-11-04	0	0	0	0	0	0	0
6	2021-11-05	0	0	0	0	0	0	0
7	2021-11-29	0	0	0	0	0	0	0
8	2021-11-30	0	0	0	0	0	0	0
9	2021-12-01	0	0	0	0	0	0	0
10	2021-12-02	0	0	0	0	0	0	0
11	2021-12-03	0	0	0	0	0	0	0

Sample_submission : (10 obs X 350 variables)

변수명	변수 타입
Open	int
High	int
Low	int
Close	int
Volume	int
Chage	float

| 데이터 소개 외부 데이터

- Sample: 파이썬 내 finance data reader 패키지를 이용.
- 시작일부터 종료일까지 해당 주식의 시가(시작 가격), 고가(최고 가격), 저가(최저 가격),
- 종가(종료 가격), 거래량, 전일대비 수익률 자료.

	A	B	C	D	E	F	G
1	Date	Open	High	Low	Close	Volume	Change
2	2021-01-04	74400	74600	73100	73800	34561	-0.0094
3	2021-01-05	74000	74600	72800	74500	45447	0.009485
4	2021-01-06	74600	79200	73600	77100	97647	0.034899
5	2021-01-07	78000	79000	77700	78600	27677	0.019455
6	2021-01-08	77900	79100	77400	78300	35356	-0.00382
7	2021-01-11	78100	78100	75200	76200	46175	-0.02682
8	2021-01-12	76000	76400	73500	76300	40767	0.001312
9	2021-01-13	76500	77200	75700	77000	27986	0.009174
10	2021-01-14	77000	80800	77000	80300	50829	0.042857

Sample: (224 obs X 7 variables)

02

데이터 탐색(EDA) & 데이터 전처리

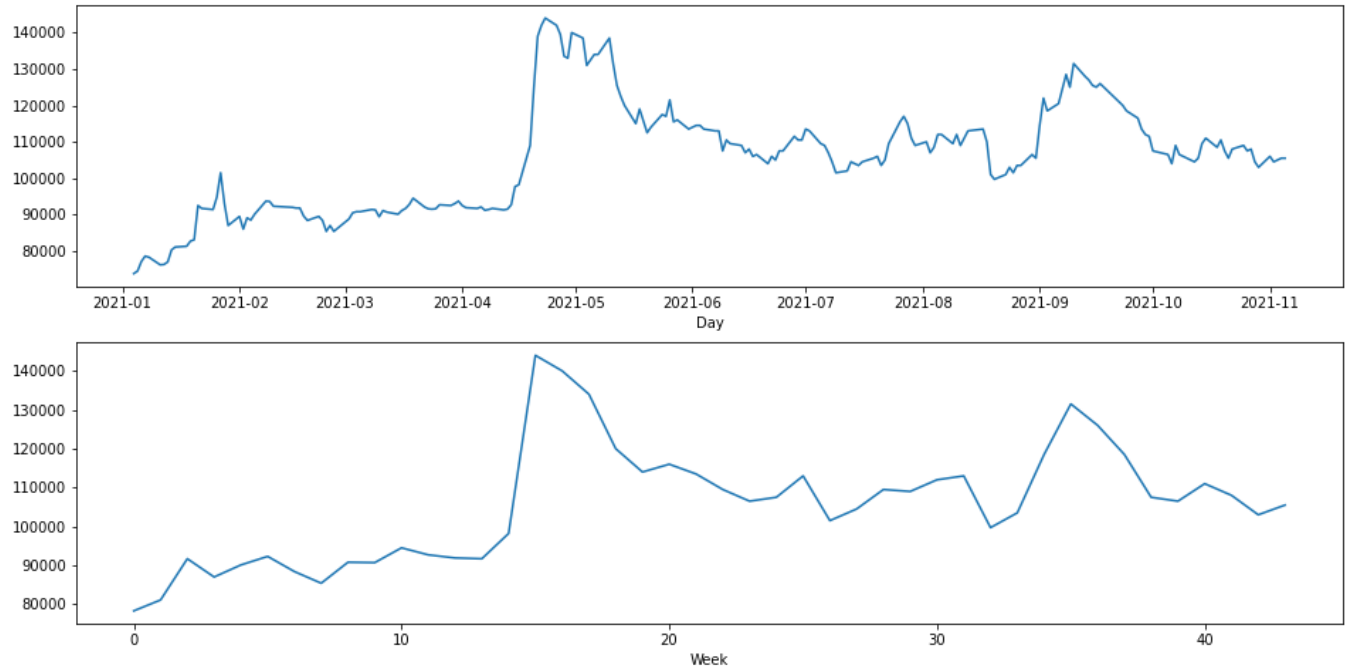
EDA 이상치 및 결측값 제거

- 주식 데이터는 상한선 및 하한선이 정해져 있으므로 따로 이상치로 처리하지 않는다.
- 21년에 상장된 주식은 데이터가 적으므로 제거한다
- DL이앤씨(11월) SK바이오사이언스(3월) LX홀딩스(5월)
- 21년도에 기존주식에 병합되거나 상장폐지된 주식은 제거한다
- GS홈쇼핑, SK머터리얼즈, 녹십자셀

EDA 주봉 데이터로 추세 파악

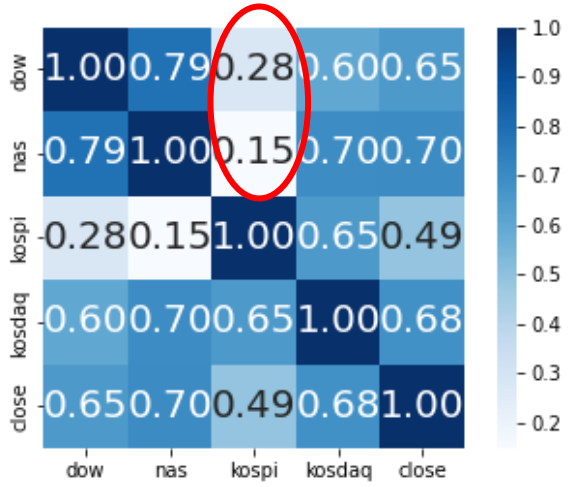
- 종가의 일별 데이터와 주별 데이터 시각화
- 주별 데이터가 더 완만하여 매일 발생되는 노이즈에 덜 예민하게 반응할 수 있다고 판단되어,

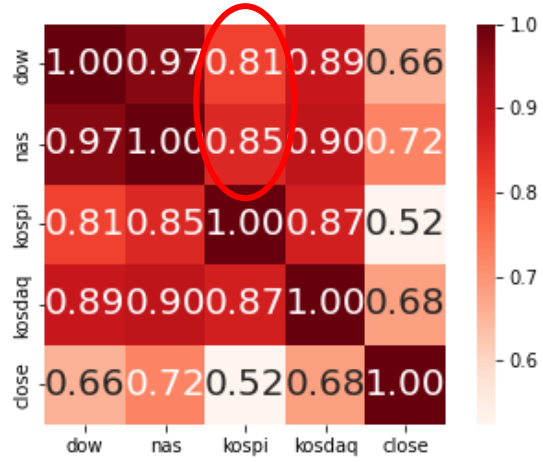
  일 단위 예측과 주 단위 예측을 병행

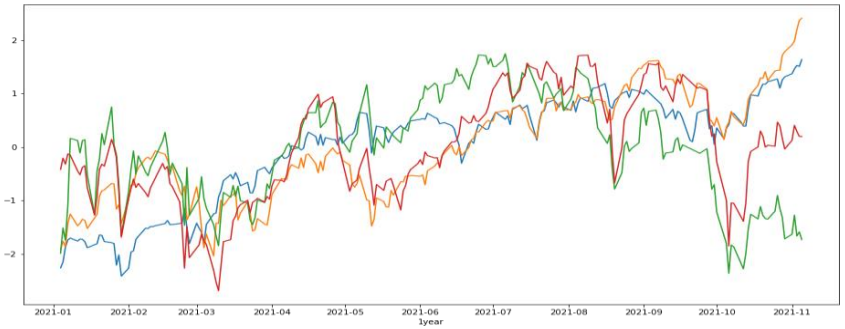


EDA 외부 지표 활용

- 나스닥, 다우지수를 가져와 코스피, 코스닥에 미치는 영향을 파악.
- 10년치 데이터와 다르게 1년치에서 코스피와 미장 지표들의 낮은 상관계수 파악
- 코스피, 코스닥, 나스닥 변수 추가

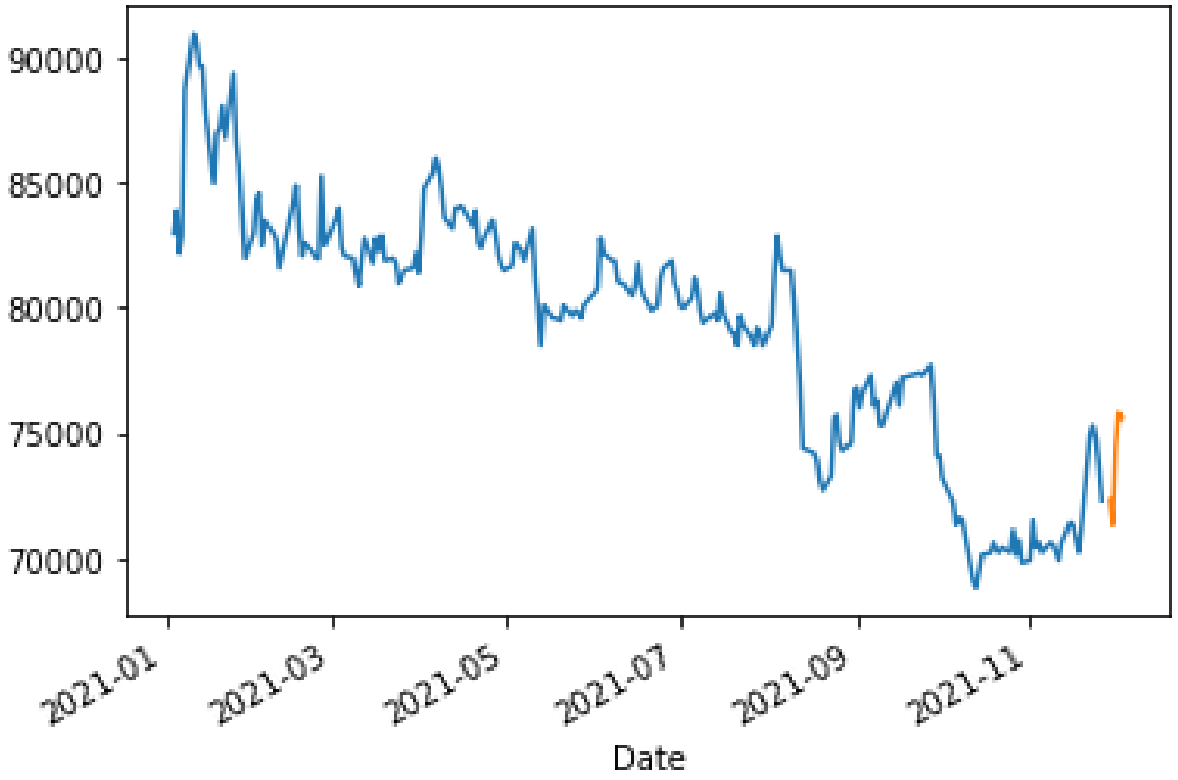






EDA 주식의 경향성

- 한 종목의 종가를 시각화 하여 추세, 계절성, 주기가 있는지 파악.

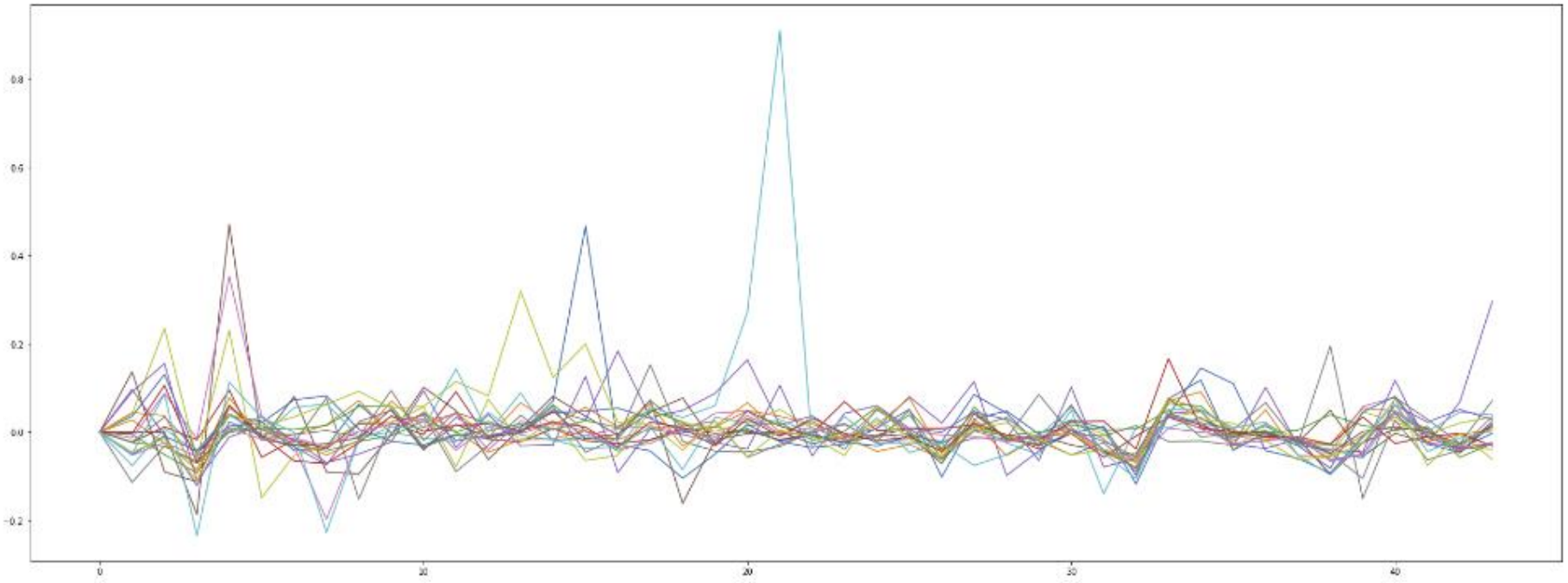


삼성전자의 2021년 1월 ~ 12월 첫째 주 시각화 모습

- 평균, 분산 등이 일정해 보이지 않음.
- 즉, 정상성은 만족하지 않음
- 하락 추세가 있지만, 상수항이 없는 임의 보행 모형
- 계절성, 주기성은 크게 보이지 않음

EDA 주식의 경향성

- 임의추출한 20개 주식의 주별 종가 시각화



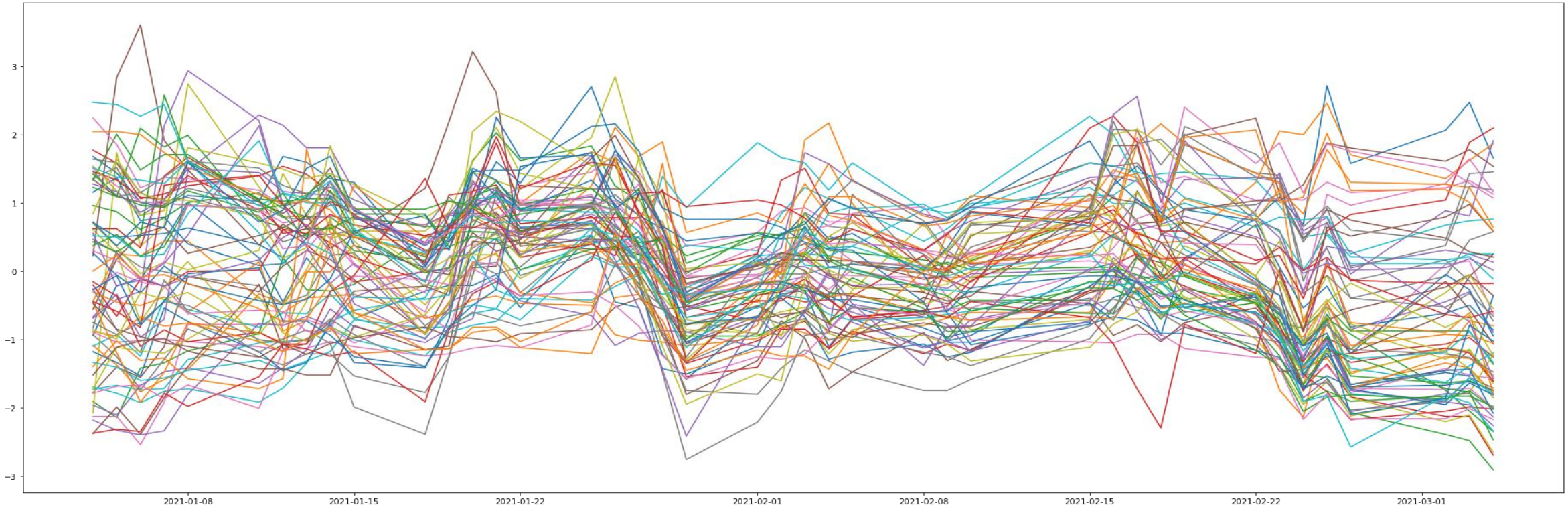
- 추세를 보면 주가는 독립적이지 않다
- 따라서 주식들 간에도 상관관계는 존재하며 이는 주식마다 테마 별로 나누면 두드러질 것이라 판단

전처리 주식별 테마 (섹터) 추출

- 기본 데이터프레임에 없는 섹터 정보 필요
- 공공데이터인 WICS의 섹터 분류지표를 웹크롤링을 이용하여 가져옴.
- 주식은 섹터별 추세를 따를 것이다. 즉 세트로 움직인다.
- Big은 대분류 Small는 소분류

	종목명	종목코드	상장시장	big	small
0	삼양홀딩스	000070	KOSPI	필수소비재	식품,음료,담배
1	하이트진로	000080	KOSPI	필수소비재	식품,음료,담배
2	유한양행	000100	KOSPI	건강관리	제약과생물공학
3	CJ대한통운	000120	KOSPI	산업재	운송
4	두산	000150	KOSPI	산업재	자본재

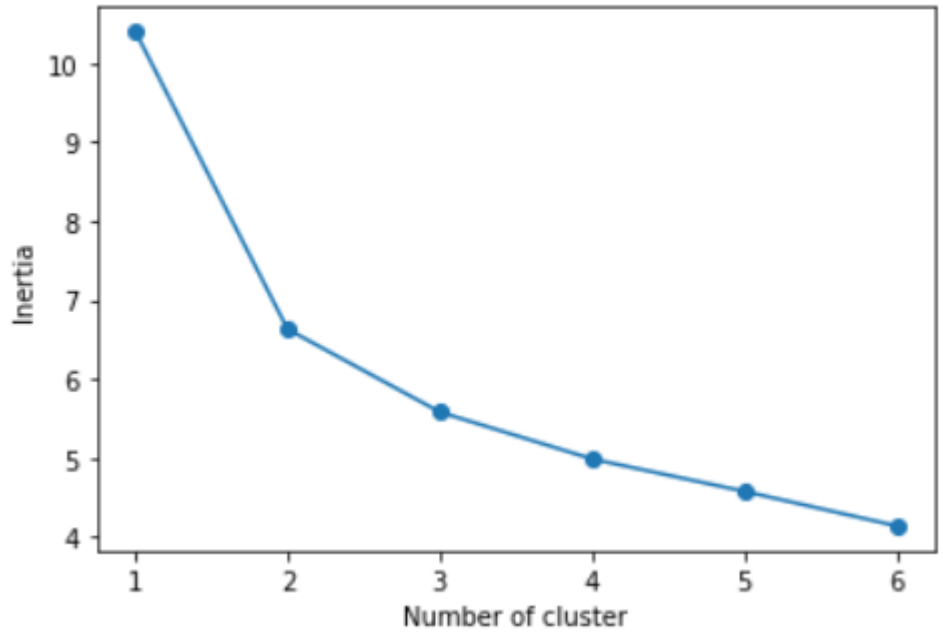
전처리 시계열 데이터 군집화



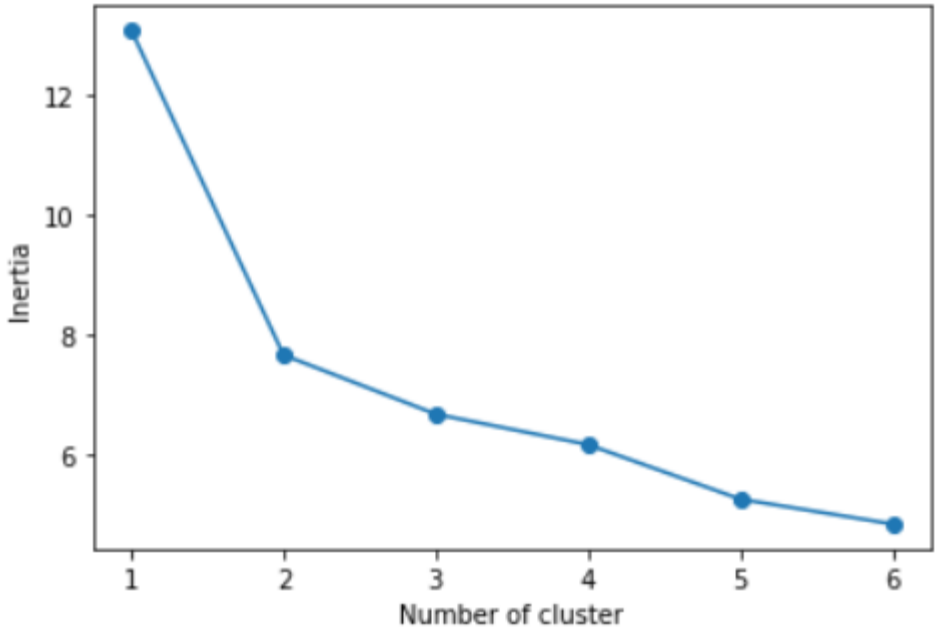
- IT 섹터 주식의 전체 종목 시각화
- 전체적으로 비슷한 경향성을 보이지만, 몇몇은 다른 움직임을 보인다. 같은 IT 대분류에 속할지라도 그 안의 70여가지의 종목에서 각기 다른 움직임을 가지는 것들끼리 묶을 수 있지 않을까?

- Times series k-means clustering
- Tslearn 패키지 사용

big 분류 루프: 건강관리

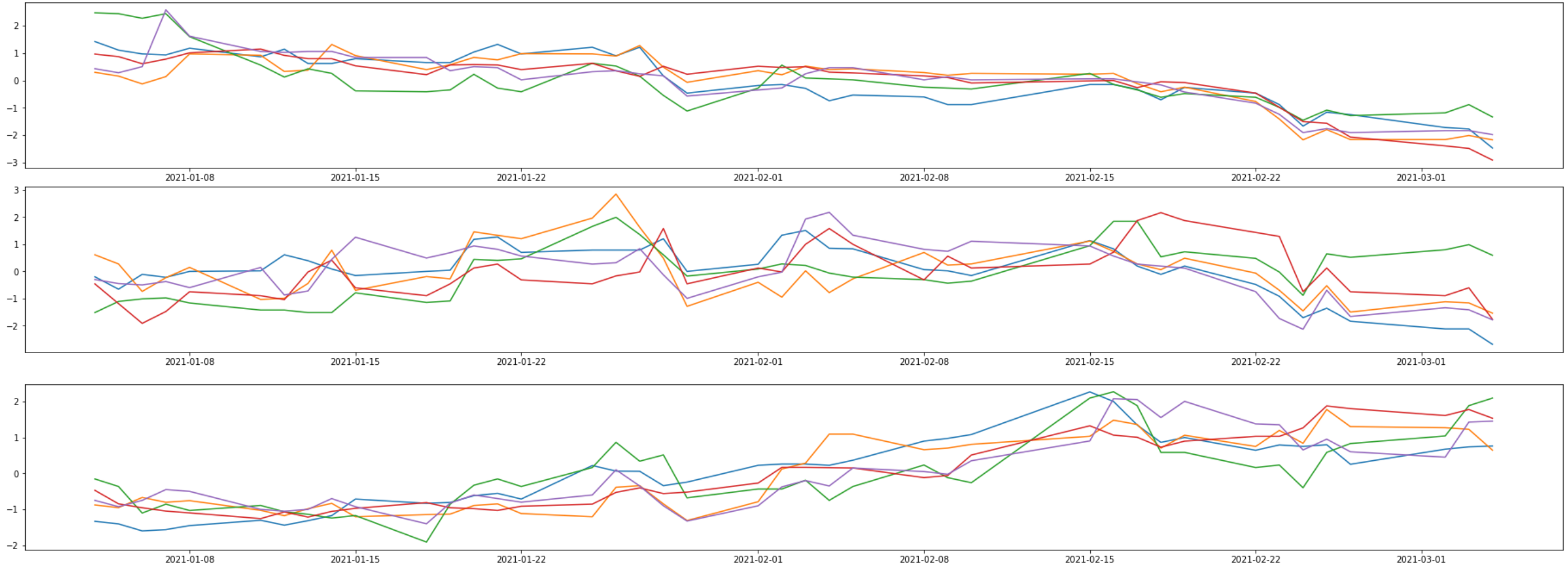


big 분류 루프: IT



- Inertia를 확인한 결과 섹터 당 종목수가 30개 넘는 종목들은 클러스터의 개수가 3개이면 적당하다고 판단

전처리 시계열 데이터 군집화



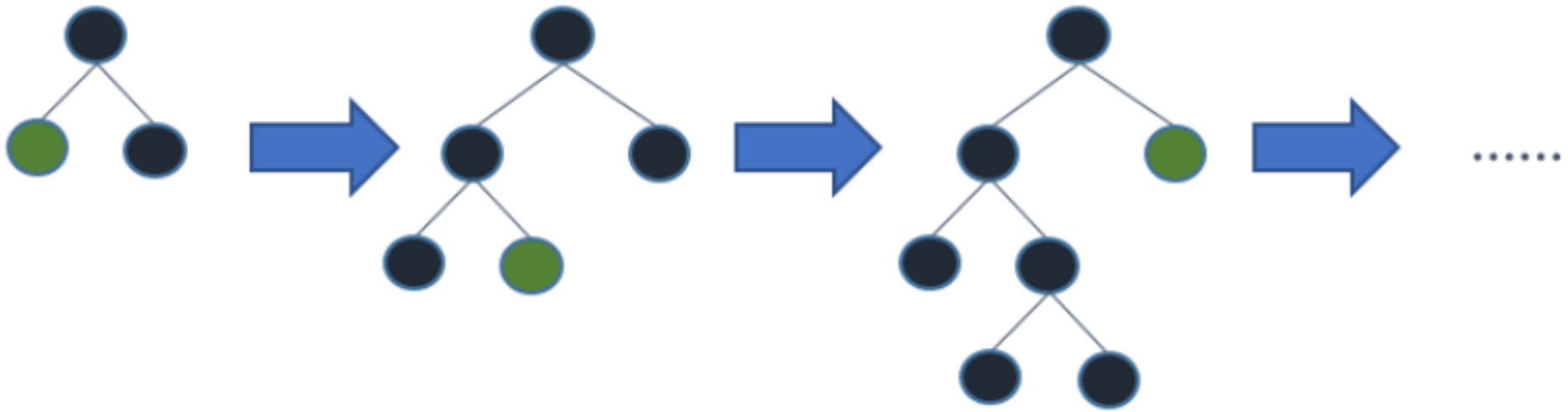
- IT 종목 군집 별 임의 5개 주식을 추출해서 시각화 한 결과 비슷한 흐름을 보이는 종목들끼리 군집화 된 것을 확인할 수 있다.

03

모형 구축 & 모형 설명

| 모형 소개 LightGBM regression

- 최대 손실값을 가지는 노드를 중심으로 계속해서 분할하는 '리프 중심 트리 분할(leaf-wise)' 방식을 사용하는 앙상블 기법.
- 모델을 학습하는데 걸리는 시간이 짧고, 메모리 사용량이 적고, 대용량 데이터 처리가 가능.



Leaf-wise tree growth

| 모델링 과정 데이터 전처리

- 예측변수 : 현재 날짜로부터 5일 뒤 종가
- 새로운 변수 추가 : after_5

사용한 변수	설명
after_5	예측변수, 현재 날짜로부터 5일 뒤 종가
Open	시가
High	고가
Low	저가
Close	종가
Volume	거래량
Change	등락률

(75336 obs X 8 variables)

- GridSearchCV로 최적 파라미터 선정 및 교차검증 진행
- parameters={‘min_data_in_leaf': [10,15,20,25,30], 'num_leaves': [20,31,35,40]'}
- 과적합 방지를 위해 min_data_in_leaf를 늘리고 num_leaves을 내리는 방향으로 접근
- CV=5로 개별 파라미터마다 train data 내 5번의 validation set을 만들어 평가

파라미터	설명
num_leaves	하나의 트리가 가질 수 있는 최대 리프 개수. default=31
learning_rate	부스팅 스탭을 반복할 때 학습률. default=0.1
bagging_fraction	트리가 커져서 과적합 되는 것을 제어하기 위해서 데이터를 샘플링 하는 비율. default=1.0
max_depth	트리의 최대 깊이 default=-1
min_data_in_leaf	한 리프의 최소 데이터 수 default=20

| 모델링 결과 모델 평가

- 예측 점수 확인
- Default 값과 임의로 파라미터를 조절한 두 모델의 평가 점수
- 거의 변화가 없지만 점수가 오히려 오른 것을 볼 수 있다.

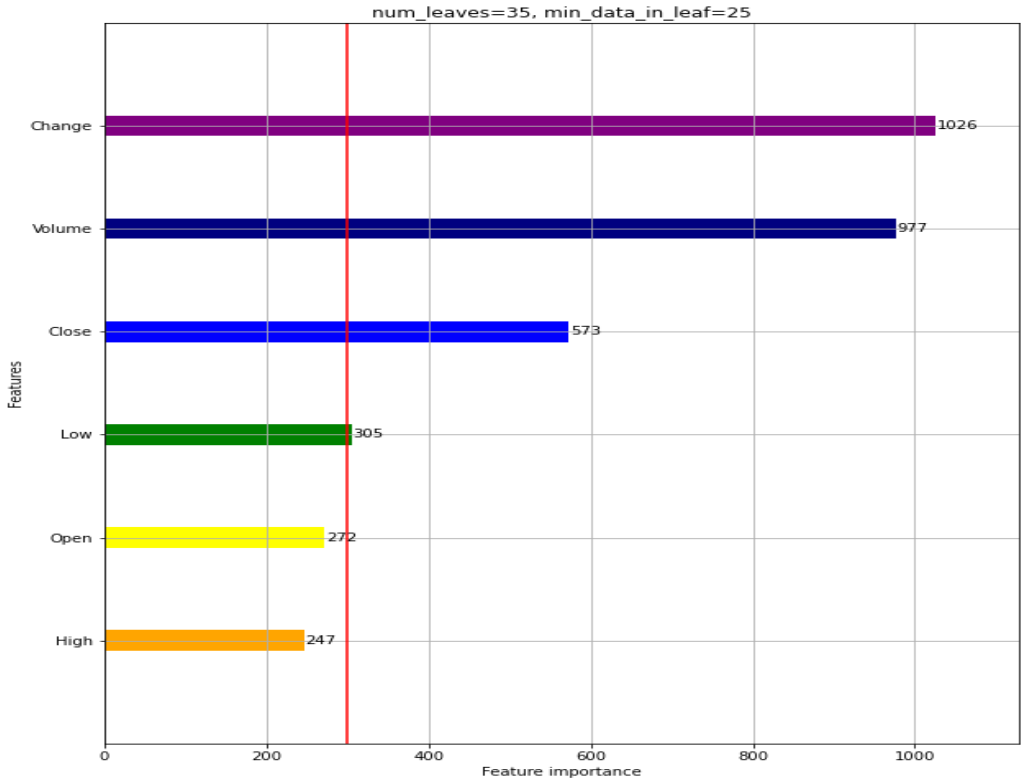
모델	NMAE(VAL)	NMAE(PRIVATE)
Default	4.61	6.76
num_leaves=31 Min_data_in_leaf=25	4.61	6.76
num_leaves=35 Min_data_in_leaf=25	4.60	6.75

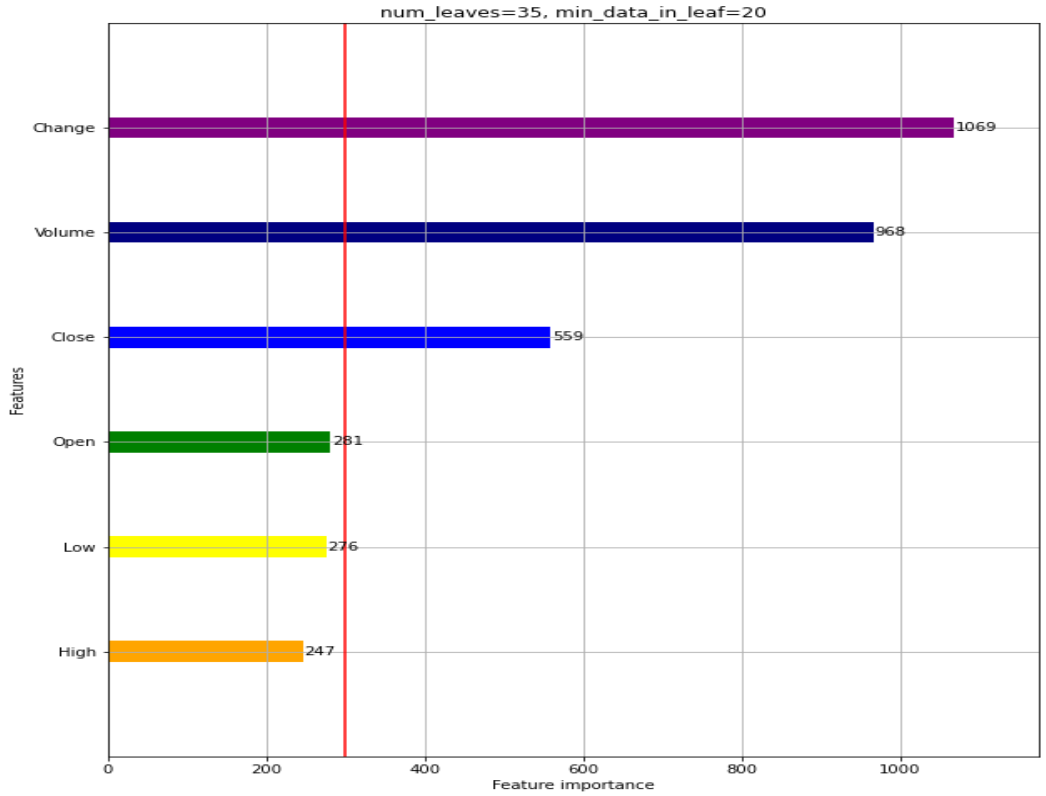
- 점수가 높아진 것을 볼 수 있다.

모델	NMAE(VAL)	NMAE(PRIVATE)
num_leaves=35 Min_data_in_leaf=30	4.61	6.72
num_leaves=35 Min_data_in_leaf=20	4.61	6.76

모델링 결과 | 변수 중요도 확인

- 임의로 파라미터를 조절한 두 모델의 변수 중요도 시각화.
- 파라미터의 값에 따라 변수 중요도가 달라진 것을 확인 할 수 있다.



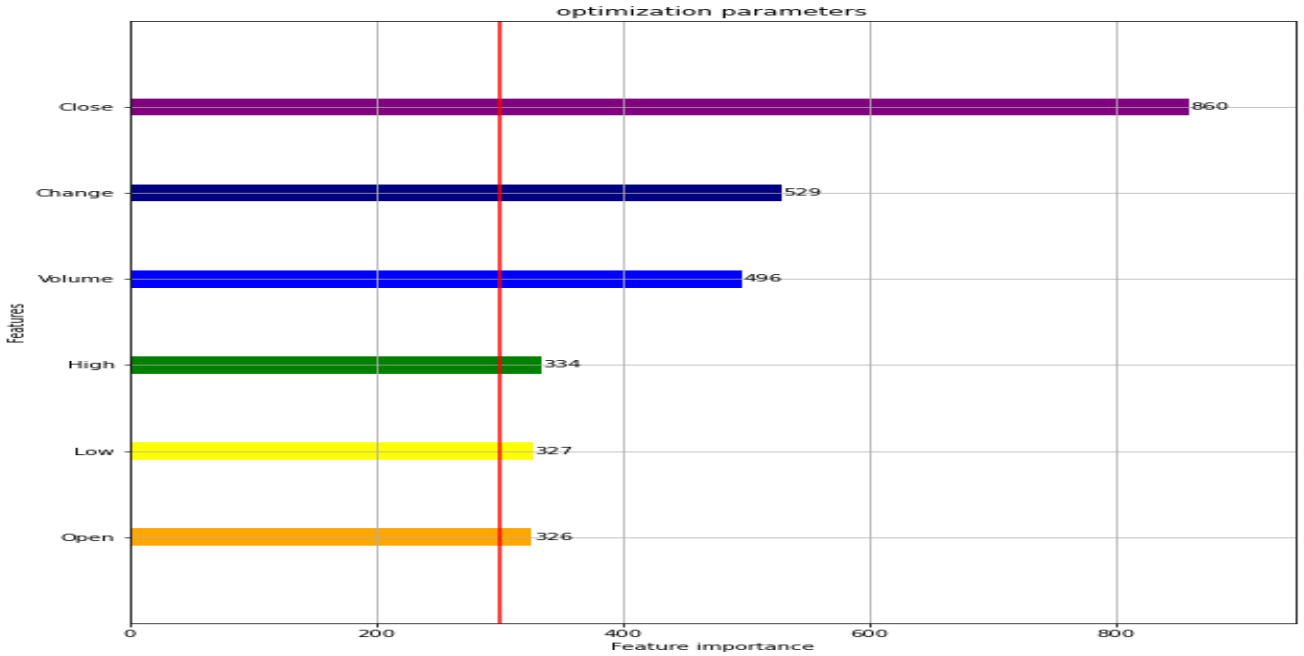


| 모델링 결과 모델 평가

- 최적화 된 하이퍼파라미터로 돌린 모델의 평가 점수
- 최적화 모델의 NMAE의 값이 가장 높게 나온 것으로 보아 모델의 과적합을 적절하게 방지하지 못한 것 같다.

모델	NMAE(VAL)	NMAE(PRIVATE)
Default	4.61	6.76
min_data_in_leaf=15 Learning_rate=0.05 num_leaves=35 Max_depth=5 Baggin_fraction=0.5	4.61	8.38

- 최적화 된 모델의 변수 중요도 시각화.



- 최적화 모델에서는 다른 모델들과 달리 Close(종가)의 변수 중요도가 가장 높게 나온다.

| 모델 소개 ARIMA 모델

- Autoregressive Integrated Moving Average Model (자기회귀누적이동평균 모형)
- AR(자기회귀) 모형 : 과거 p 시점의 값이 현재 자료에 영향을 줄 때
- MA(이동평균) 모형 : 과거 q 오차가 현재의 자료에 영향을 줄 때

- 주식은 비정상성 시계열 자료이다.
- 비정상시계열을 차분이나 변환을 통해 정상성을 만족하는 시계열로 바꿔줄 수 있다.
- 정상화란? – 일정하며 늘 한결같은 성질. 평균이 일정할 때, 분산이 일정할 때, 공분산도 단지 시차에만 의존하고 실제 특정 시점 T,S에 의존하지 않을 때 만족한다.
- 차분 : t시점과 t-1 시점의 차이
- 변환 : 로그변환을 통해 분산을 안정화 시키고, 지수적인 값을 가지는 시계열을 선형적으로 바꿔 줌.

- 차분이 필요한지, 몇 차 차분이 최선인지 파악하는 라이브러리 사용하여 변수로 입력
- Best 모델 찾기

파라미터	설명
y_train	예측할 마지막 5일 전 전체 학습 시킬 데이터
d	차분 차수. 위에서 찾은 최선의 차분 ndiffs 결과.
start_p ~ max_p	AR (p) 값을 찾을 범위
start_q ~ max_q	MA (q) 값을 찾을 범위
m	계절적 차분이 필요할 때. 필요하지 않으면 1
seasonal	계절성 arima 모형이 적합하지 않으면 false
stepwise	최적의 모수를 찾기 위한 힌드만 – 칸다카르 알고리즘을 사용할지?

```
Performing stepwise search to minimize aic
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=3422.515, Time=0.18 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=3423.523, Time=0.11 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=3425.906, Time=0.06 sec
 ARIMA(0,1,0)(0,0,0)[0]             : AIC=3421.229, Time=0.02 sec
 ARIMA(1,1,1)(0,0,0)[0] intercept   : AIC=3427.008, Time=0.19 sec

Best model:  ARIMA(0,1,0)(0,0,0)[0]
Total fit time: 0.579 seconds
```

- Minimum AIC 찾기
- AIC란? Akaike's information Criterion : ARIMA 차수 선택과 모수 추정의 기준이다.
- AIC=-2log(Likelihood)+2(p+q+k+1)
- Likelihood : 실제 관측된 데이터가 그 모형을 따를 확률 → 잘 적합 ↑ 가능도 ↑
- 복잡한 모형은 과적합하기 때문에 불필요한 변수가 들어가면 AIC값이 커지도록 패널티를 부여

| 모델링 과정 잔차 검정

SARIMAX Results

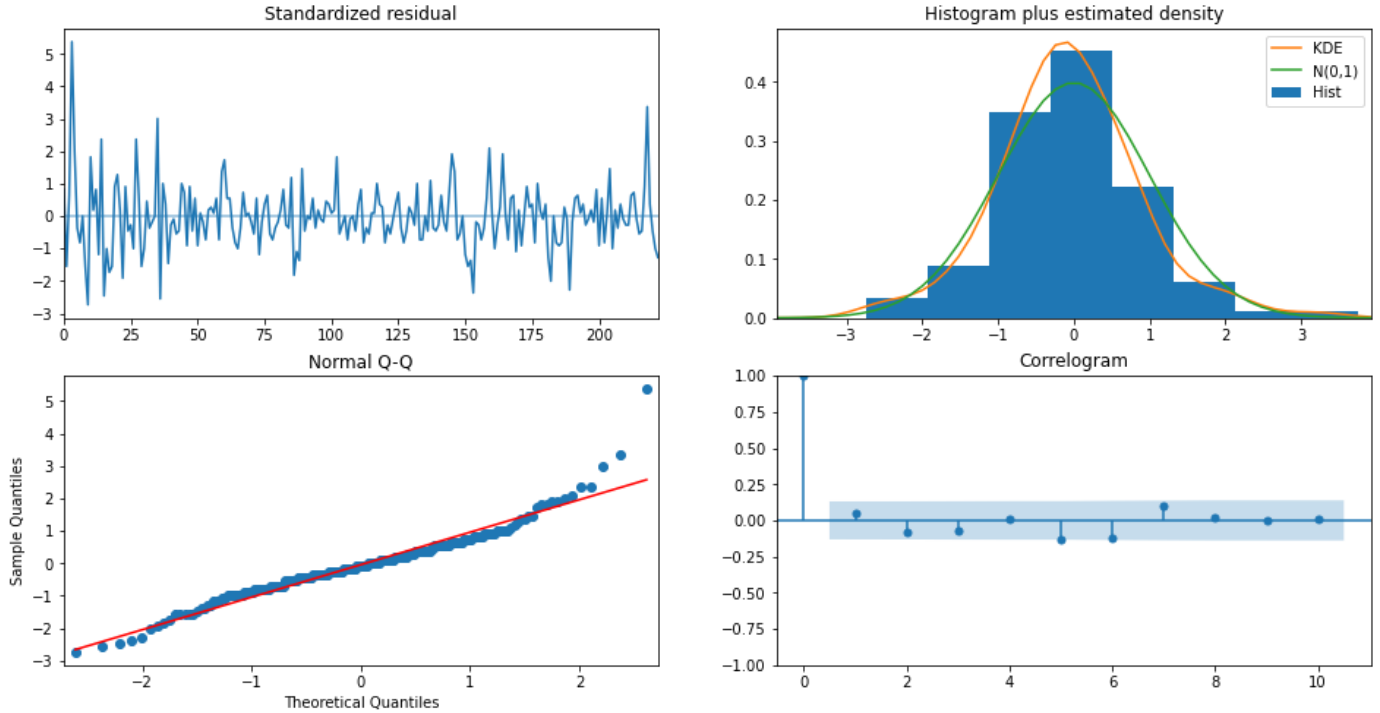
Dep. Variable:	y	No. Observations:	224
Model:	SARIMAX(0, 1, 0)	Log Likelihood	-1878.116
Date:	Sun, 27 Mar 2022	AIC	3758.233
Time:	21:05:49	BIC	3761.640
Sample:	0 - 224	HQIC	3759.608
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
sigma2	1.205e+06	6.61e+04	18.230	0.000	1.08e+06	1.33e+06

Ljung-Box (L1) (Q):	0.62	Jarque-Bera (JB):	182.53
Prob(Q):	0.43	Prob(JB):	0.00
Heteroskedasticity (H):	0.52	Skew:	0.87
Prob(H) (two-sided):	0.01	Kurtosis:	7.07

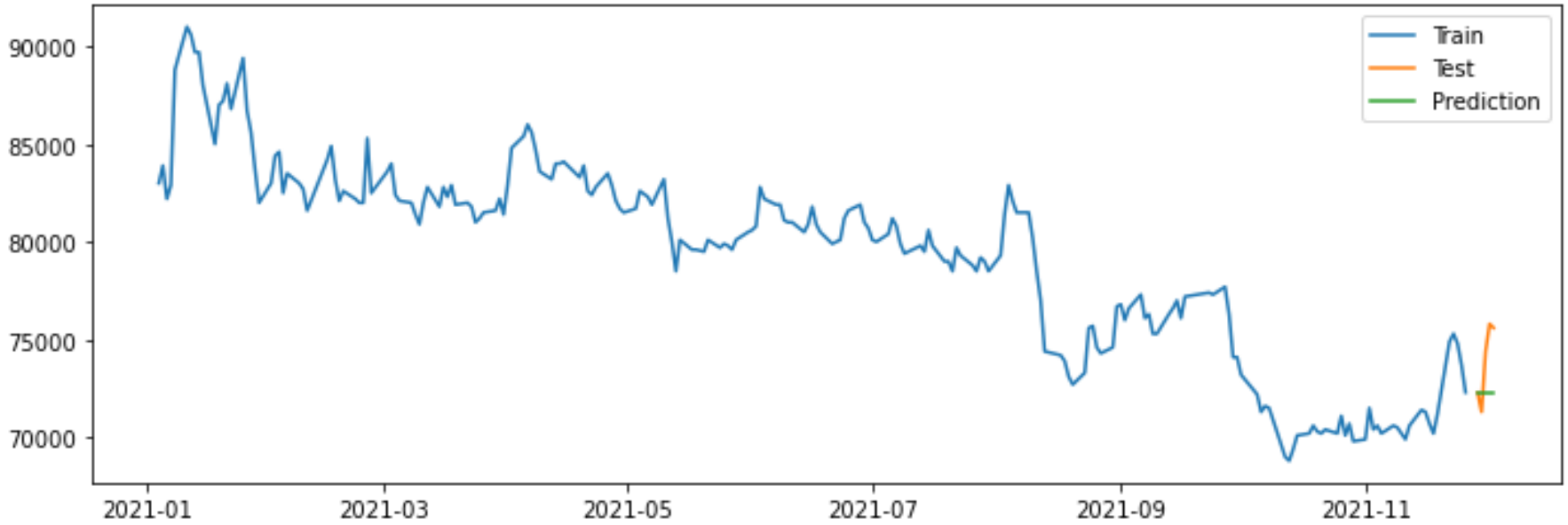
- Prob(Q)값을 보면 0.62으로 유의수준 0.05에서 귀무가설을 기각하지 않는다.
  남은 잔차는 더 이상 자기상관을 가지지 않는 백색 잡음이다.
- Prob(jb)는 0.00 이므로 유의수준 0.05에서 귀무가설을 기각한다.
  잔차가 정규성을 따르지 않음.

모델링 과정 | 잔차 검정



- Standardized residual을 확인하면 평균 0을 중심으로 무작위하게 움직임을 확인 가능하다.
- Correlogram은 허용 범위 안에 위치하며 자기상관이 없다.
- 잔차의 Histogram은 초록색 normal 분포와 비교하면 거의 대칭이지만 첨도가 더 뾰족하다.
- Normal Q-Q 그래프는 정규성을 만족한다면 빨간 일직선 위에 점들이 분포해야 한다.

| 모델링 과정 예측 결과



- 모델에 넣었을 때 예측 결과인 마지막 5일치가 일직선이다
- (0,1,0) 모델은 상수항이 없는 임의 보행 모형이므로, 가장 마지막 관측치를 넣어 모형을 계속 업데이트 해야 한다.

모델링 과정 | 예측 수정

- 약간의 시간 차는 있지만 비슷한 흐름으로 예측 성공했다.
- 이를 응용해서 전체 종목을 모두 Best 모델을 찾아 예측한다.

| 모델링 결과 예측 결과

- NMAE 결과 : 1.665

평균 예측 값과 실제 값이 1.6% 정도 차이 남을 알 수 있다.

차이가 적은 점을 보아 과적합의 가능성?

→ 개별 종목에 대한 모형은 매일 전 날의 실제 값을 넣어주면 단기 예측에 효과적이다.

	test	pred
0	106000	103000
1	104500	106000
2	105000	104500
3	105500	105000
4	105500	105500
5	35200	34739.5
6	35050	35247.7
7	34050	35032.2
8	33800	33930.2
9	33450	33769.8
10	57422	57327
11	59049	57422
12	58858	59049
13	58475	58858
14	57996	58475
15	146000	141905

- Random Forest : 다수의 결정 트리들을 학습하는 앙상블 기법.

  Bagging(Bootstrap aggregating) +random selection of features

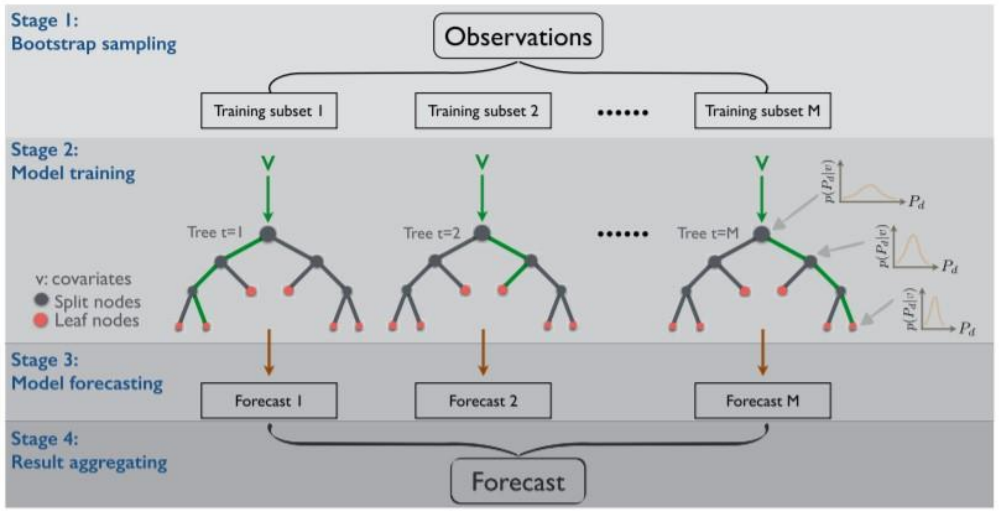
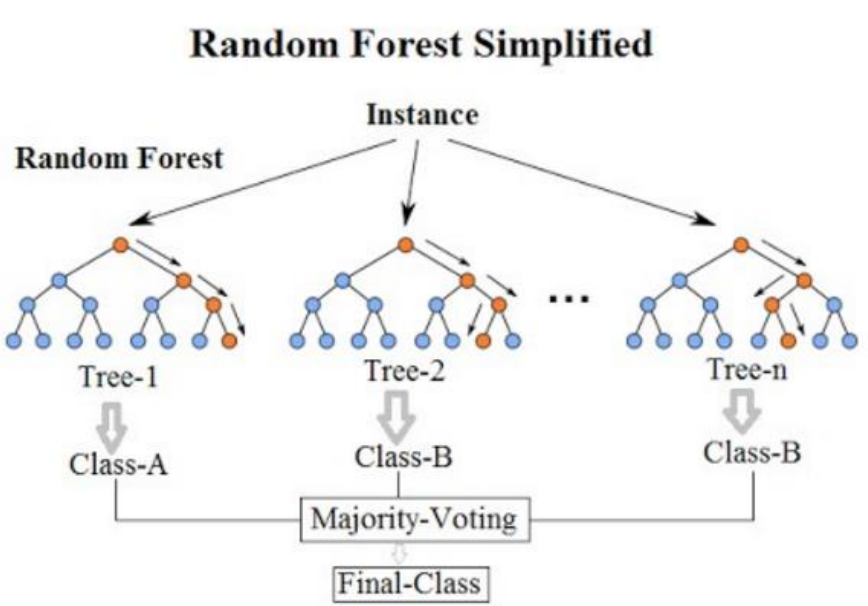


Figure 1. Schematic of the RF algorithm based on the Bagging (Bootstrap + Aggregating) method. (1) Stage 1: Use bootstrap method to sample *M* subsets from the original training data sets. (2) Stage 2: Build *M* independent decision trees for model training using input covariates (**v**). For each individual decision tree, the prediction confidence (posterior probability $p(P_d|\mathbf{v})$ increases from the root toward the leaves. (3) Stage 3: Obtain prediction from each bootstrap tree over *M* replications. (4) Stage 4: Decide the final result by average or majority voting.

출처 : www.researchgate.net/



출처 : https://medium.com/@williamkoehrsen/random-forest-simple-explanation-377895a60d2d

| 모델링 과정 데이터 전처리

- 앞서 진행한 두 모델과의 차이 : 예측값을 각 요일별 종가 → 예측할 주의 평균 종가
- 전체 시장과 코스피와 코스닥 두가지의 시장으로 나누어서 분석진행
- 새로운 변수 추가 : after5days, last5days, 5avg

사용한 변수	설명
after5days	예측변수, 예측일 기준 앞으로 5일간 평균종가
last5days	예측일 이전 5일의 평균수익률
5avg	예측일 이전 5일의 평균종가
Close	종가
Nasdaq	나스닥 지수
Kospi	코스피 지수
Kosdaq	코스닥 지수
Cluster dummies	앞서 크롤링한 섹터변수를 더미변수로 처리

(73402 obs X 26 variables)

- GridSearchCV로 최적 파라미터 선정 및 교차검증 진행
- parameters={'max_features': [2,4,5], 'n_estimators': [100,200,300]'}
- 과적합 방지를 위해 max_features를 줄이고 n_estimators를 늘리는 방향으로 접근
- CV=5로 개별 파라미터마다 train data 내 5번의 validation set을 만들어 평가

파라미터	설명
N_estimators	결정 트리의 개수. Default=10
Max_features	트리당 고려되는 최대 피처 개수. Default=auto (=sqrt)
Min_samples_split	노드를 분할하기 위한 최소한의 샘플 데이터수 Default=2
Max_depth	트리의 최대 깊이 Default=None
Min_sample_leafs	리프 노드가 되기 위해 필요한 최소한의 샘플 데이터 수
Max_leaf_nodes	리프 노드의 최대 개수

| 모델링 결과 모델 평가

- 예측 점수 확인
- Default 값과 임의로 파라미터를 조절한 두 모델의 평가 점수
- 눈에 띄는 변화는 보이지 않는다.

모델	NMAE(VAL)	NMAE(PRIVATE)
Default	3.38	3.68
Max_features=24 n_estimator=100	3.42	3.62
Max_features=24 n_estimator=300	3.42	3.60

- 최적화된 하이퍼파라미터로 돌린 모델의 평가점수
- 기본 모델보다 점수가 오히려 높아진 것을 볼 수 있다.
- 평균 점수가 높아진 것을 확인.

모델	NMAE(VAL)	NMAE(PRIVATE)
Max_features=5 n_estimator=300 (최적화)	8.01	6.03
Max_features=2 n_estimator=300	16.44	14.82

모델링 결과 | 모델 평가

- 그리드서치로 Cross Validation을 사용했을때 CV별 평가점수의 편차가 큰 것을 확인
- 이는 현재 Train set이 주식별 내림차순 정렬되어 있어 Validation set이 나뉘는 지점에 따른 영향이라 판단

Sampling	CV별 점수	평균
기존	[10.3, 15.4, 8.9, 5.7, 5.2]	8.01
임의추출	[2.99, 2.96, 3.0, 3.01, 2.93]	3.00

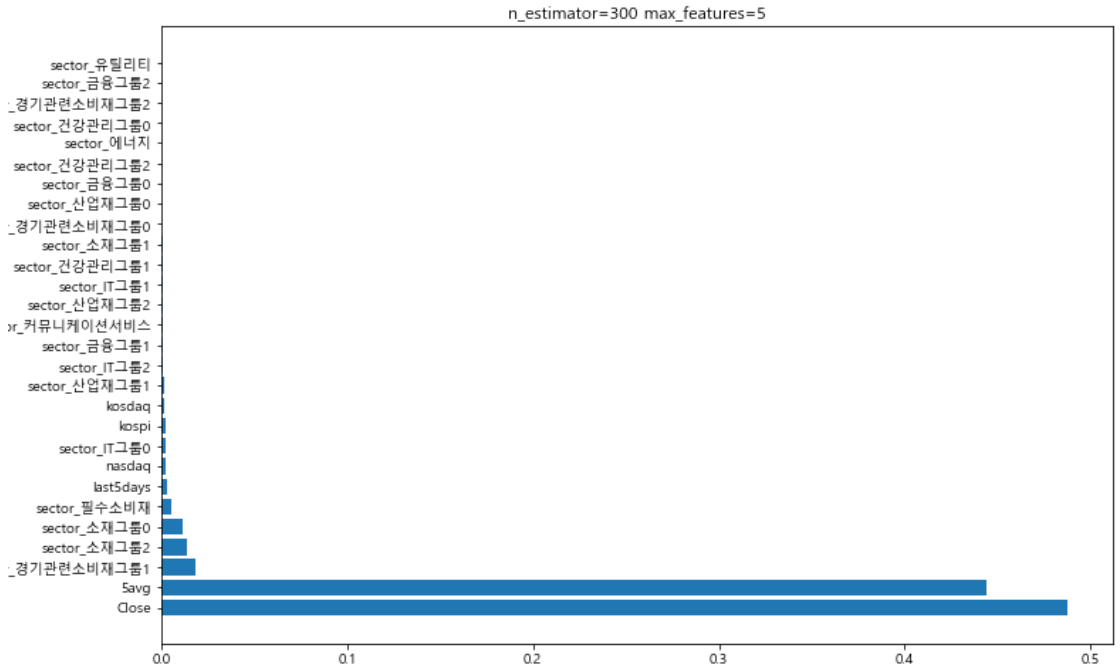
- 따라서 Validation set을 임의 추출하는 방식으로 변경하여 CV를 진행

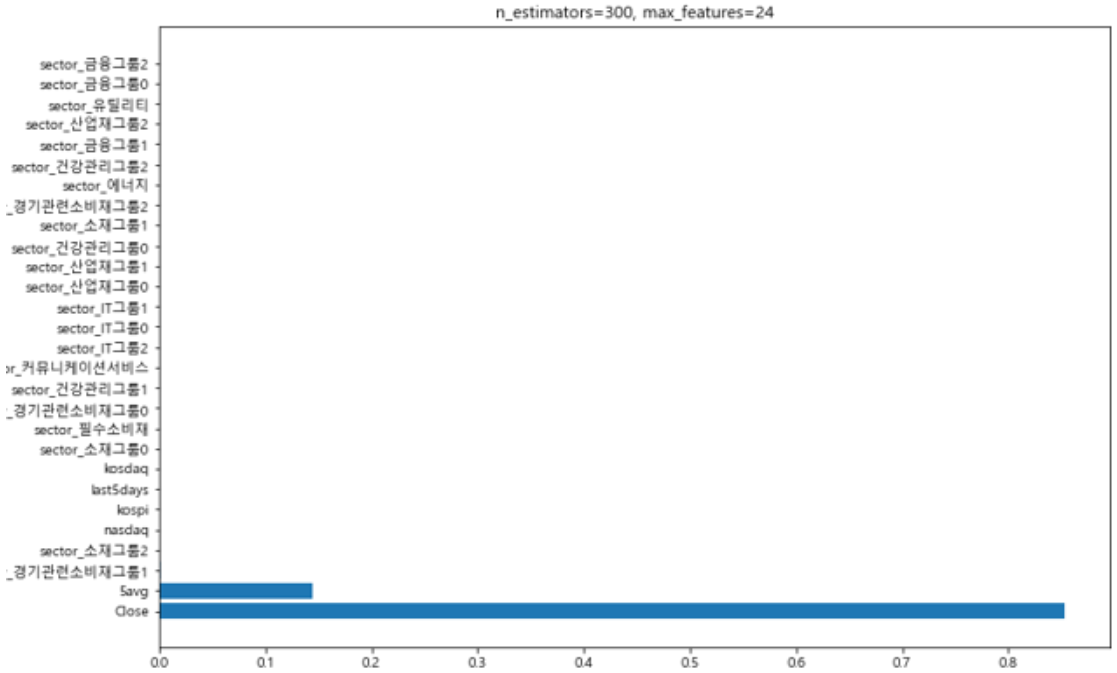
- 점수가 개선되었음을 확인
- 하지만 Default 모델과 비교했을 때 val점수와 private점수의 차이가 큰 것으로 보아 과적합이 발생한 것으로 보임

모델	NMAE(VAL)	NMAE(PRIVATE)
Max_features=5 n_estimator=300 (최적화)	3.00	6.03
Default	3.38	3.68

| 모델링 결과 변수 중요도 확인

- 최적화된 파라미터 값과 임의로 파라미터를 조절한 두 모델의 변수 중요도 시각화.
- 최적화된 파라미터 개수에 따라 변수 중요도가 달라짐.
- 종가의 영향력이 매우 큼.
- 더미변수를 제외하고 중요도가 제일 낮은 KOSPI, KOSDAQ 처리





모델링 결과 | 모델 평가

- 새로운 모델링 및 예측 점수 확인
- 앞서 EDA에서 확인한 미장 지표와 코스닥 코스피를 분리
- 코스피 시장 : 전체 모델에서 코스닥과 나스닥을 제외

모델	NMAE(VAL)	NMAE(PRIVATE)
Default	2.42	3.71
Max_features=5 n_estimator=300	2.44	4.94

평균 모델점수(가중치 고려)

Default = 3.86

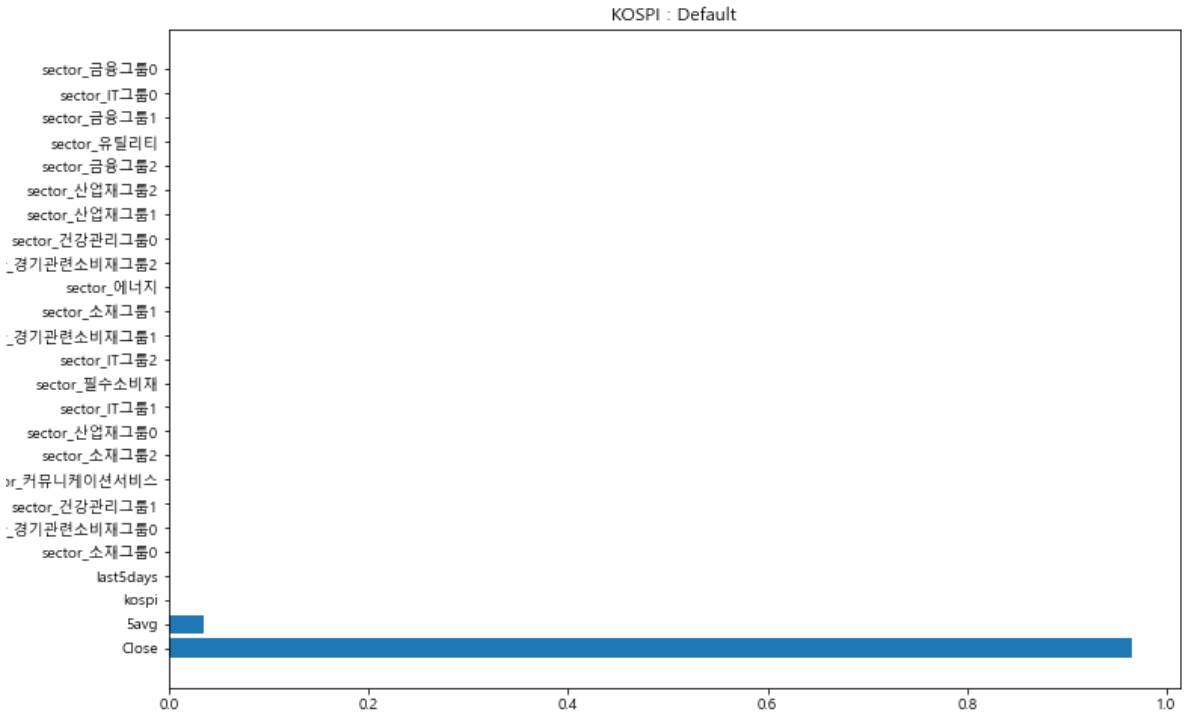
최적화 = 5.00

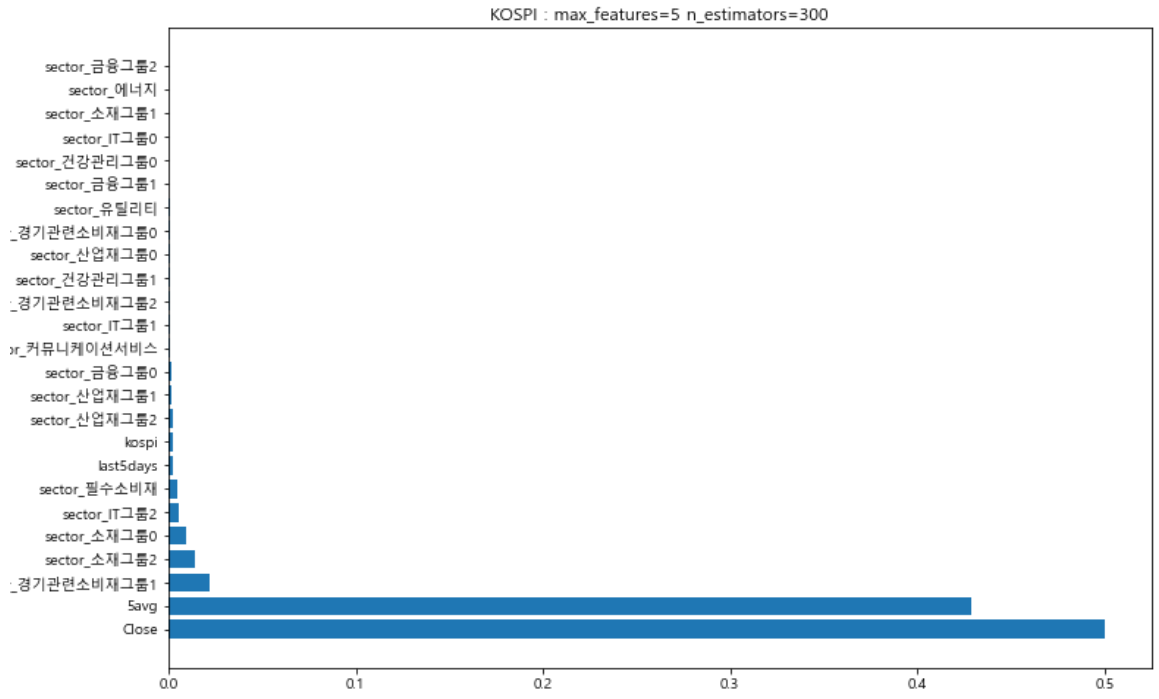
- 코스닥 시장 : 전체모델에서 코스피만 제외

모델	NMAE(VAL)	NMAE(PRIVATE)
Default	2.83	4.06
Max_features=5 n_estimator=300	2.68	5.09

| 모델링 결과 변수 중요도 확인

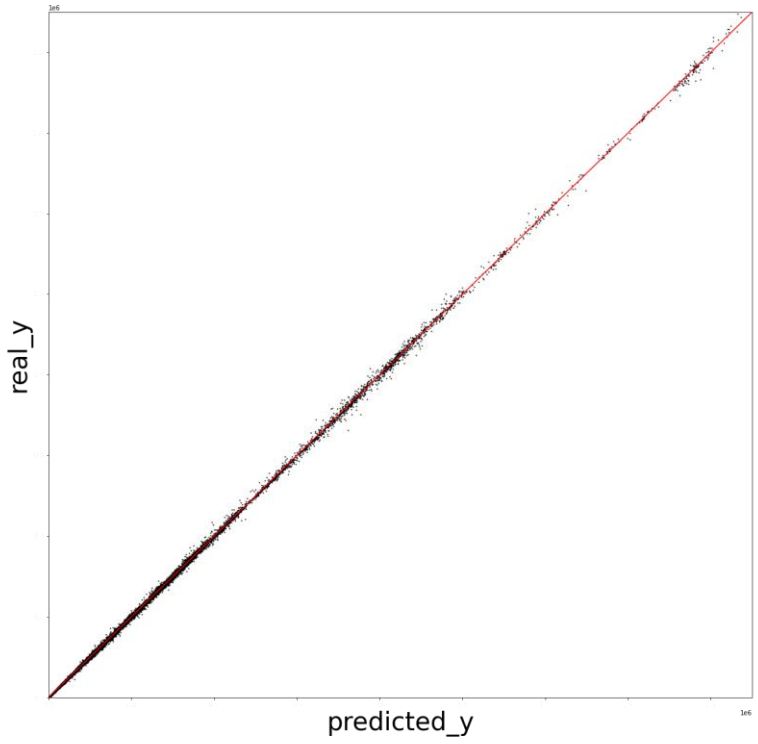
- KOSPI 예측 점수 확인
- Default 값과 최적화된 파라미터를 조절한 두 모델의 변수 중요도 시각화.
- 전체 모델링과 마찬가지로 피처 개수에 따라 변수 중요도가 달라진것을 확인할 수 있다.

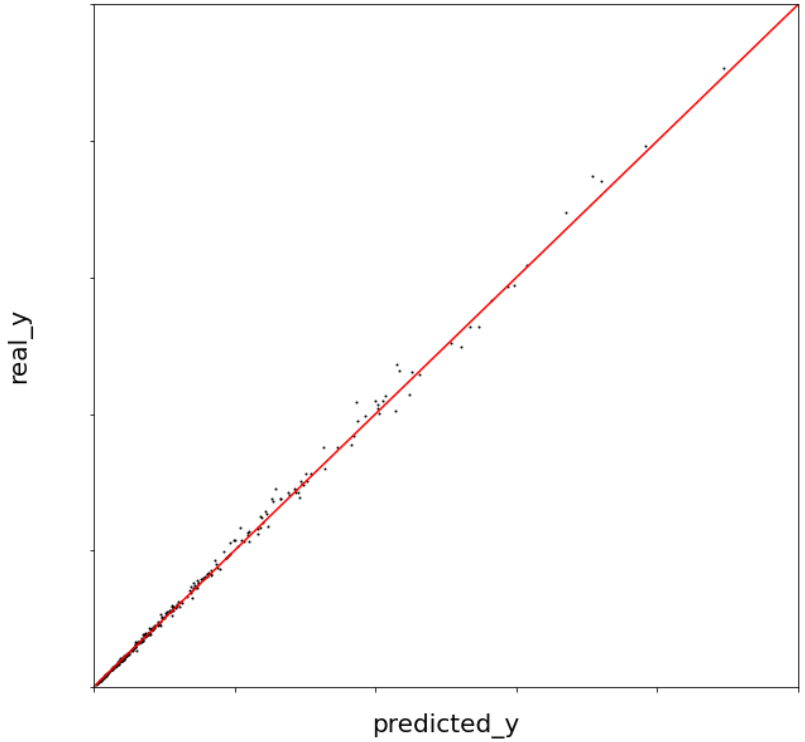




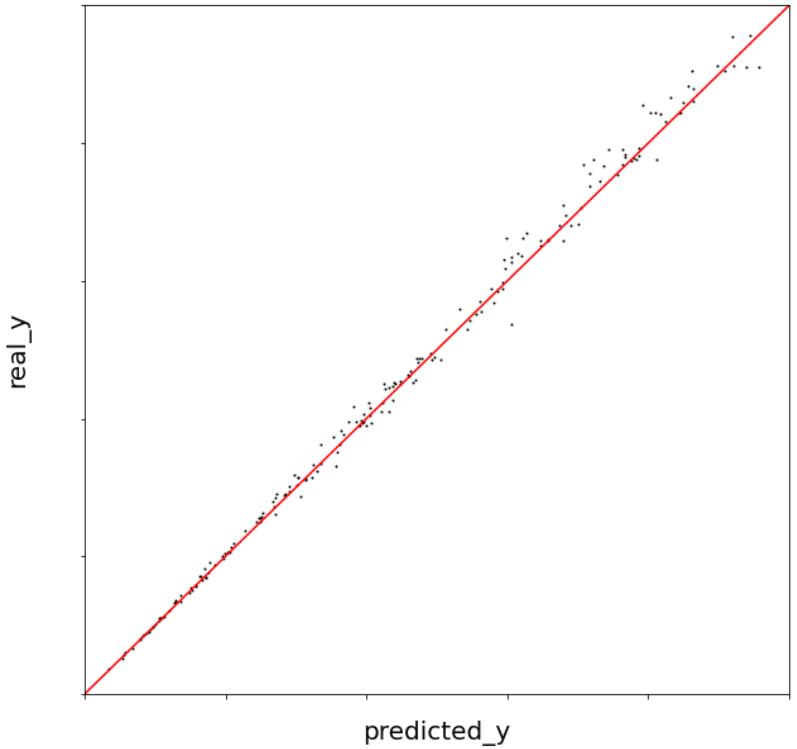
모델링 결과 | 예측 시각화

Train





Test



04

분석결과 & 결론

- 최종 모델은 랜덤포레스트 모델을 선택.
- 최종점수(NMAE) : 3.60
- 선정이유 :

  과적합 혹은 과소적합의 우려가 있지만, 범용 모델이 아니라 예측 전일까지의 데이터를 가지고 단기간의 가격을 예측하는 모델이기 때문

  validation set 와 test set 간 점수 차이가 가장 낮음

  ARIMA 모델이 점수는 제일 좋지만, 종가 한 가지의 데이터만 사용했을 때 예측 전일의 종가를 예측 당일 종가로 거의 그대로 가져오는 모습이 보여 예측에 큰 의미가 없다고 판단

| 결론 및 아쉬운 점

- 경진 대회 참여 버튼이 비활성화 되어 제출을 통해 우리의 결과를 직접 비교하지 못한 점

- 단순 종가만 가지고 다음 날의 가격을 예측하는것은 논리적 비약이 있으며, 주식 자체가 예측이 불가능하다는 랜덤워크 이론에 따르면 주식의 가격 자체를 예측하는 것보다 주식의 추세나 등락 여부를 예측 하는 것이 더 적합 했을것이라 생각

- 생성한 LightGBM 모델이 과적합이 심한 모델이여서 최적화 된 하이퍼 파라미터로 조정을 했음에도 불구하고 좋은 NMAE값을 얻지 못함

- 주식은 시계열 자료임에도 불구하고 시계열 분석 방법으로 더 깊이 들어가지 못한점

bye.ipynb

```
if __name__ ==“__end__”:
    say()
def say() :
    if page is last:
        print(“Thank you”)
```