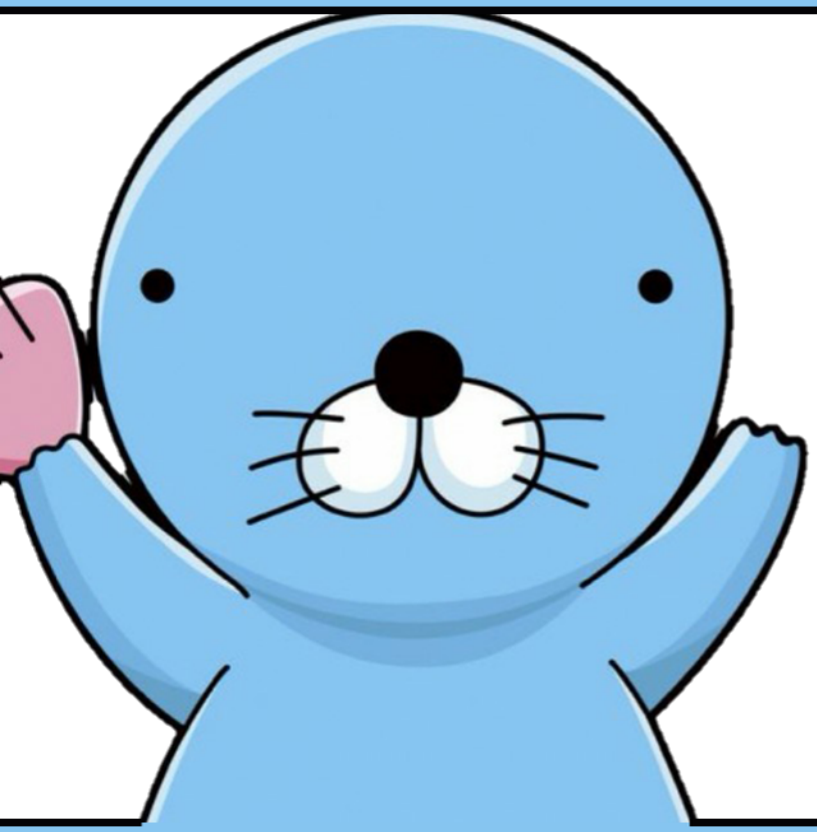


# Ants Together Strong(3)

## ML/DL을 이용한 주가 향방 예측

김동수 안태용 이찬영 최선빈

# 목차

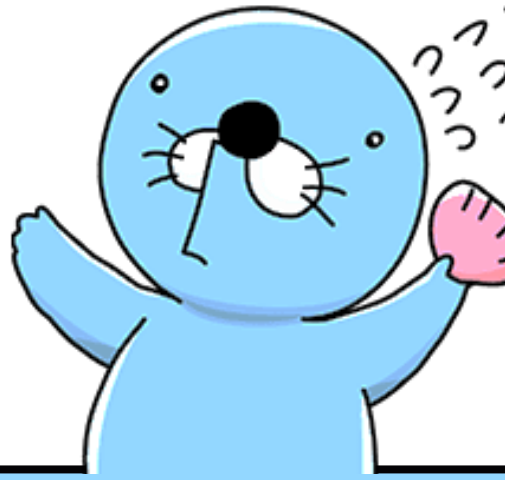


# 개요

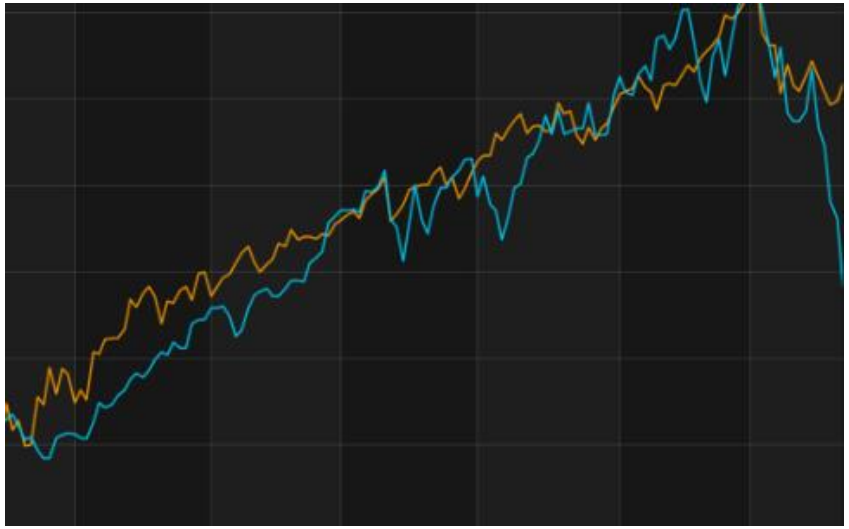

## 주식이란?

개인 / 단체가 특정 회사에 일정 금액을 투자
1. 투자금에 걸맞은 이익을 배당받음
2. 회사 경영권에 행사 : 주주총회 참여 및 투표권을 보장하는 증서

즉, 배당 / 경영참여권



# 개요



## 증시란?

국가에서 증권을 거래하는 시장을 운영
이 거래소를 증권시장, 즉 증시라고 함
국내에는 유가증권시장, 코스닥
해외에는 NYSE / NASDAQ 등이 있음
증시의 과거 대비 현 위치를 나타내는 숫자 값을
‘지수’라고 함
KOSPI / KOSDAQ / S&P500 등이 있음

# 개요

## 선물(futures)이란?

특정 기초자산을, 특정일에, 특정 가격으로 거래하겠다는 계약을 선도계약(forwards)라고 함. 이를 표준화하여 거래소에서 거래 할 수 있게 만든 계약을 '선물계약' 이라고 함

**WTI유 선물 - 2022년 7월물 $ 116.09**

-> 2022년 7월 계약 만기일에, 배럴 당 $116.09에, WTI유를 사겠다 / 팔겠다는 "계약"

# 개요


## ETF란?

상장 지수 펀드. 주가 지수나 채권 지수 등 특정 지수를 추종하여 거래소에 상장되어 거래되는 “펀드”
선물 지수 또한 추종 대상이 될 수 있고,
대표적으로 KODEX 200은 선물 지수 KOSPI 200을 추종
선물 거래는 규모가 크기 때문에, 개인이 지수에 투자하기에 가장 좋은 수단

# 워렌 버핏, 옥시덴탈 페트롤리움 90만주 추가 매수

## "초단타매매로 개미 털었다" 금감원도 칼 뽑아...시타델 "공정 거래"

환율 내리자 돌아온 외국인...'안도랠리 기대' - 서울경제

## '피엔티' 52주 신고가 경신, 단기·중기 이평선 정배열로 상승세

---

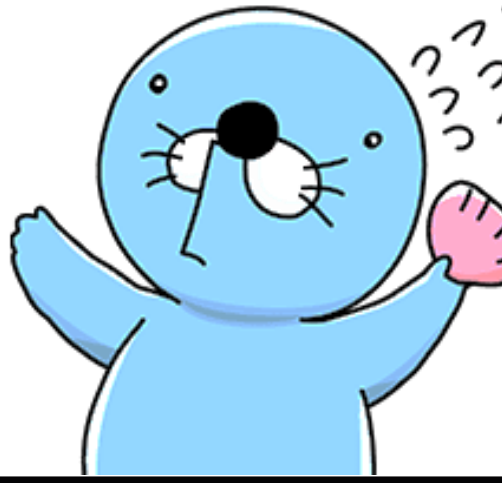
**주식의 향방을 예측하는 지표는 너무나도 많다**

↓

**각자의 지표는 의미를 가질 것이라는 기대**
**컴퓨터를 활용해 각자의 지표 학습 / 합치면 수익 기대?**

↓

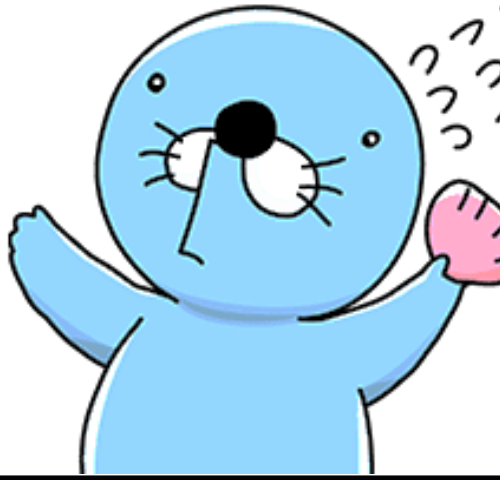
**ML / DL을 통한 주가 향방 분석**




각자 주가 분석에 있어 '유의'하다고 생각되는 부분을 선택

- 차트 지표
- 캔들 차트
- 수급
- 시계열
- 자연어 / 거시경제

1. KODEX 200의 상승 / 하락 예측
2. 개별주 차원으로 확장

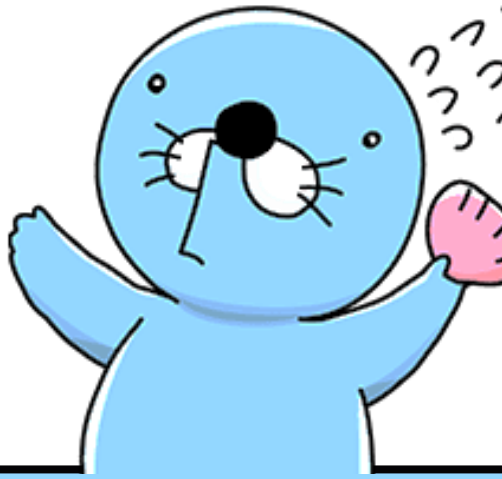


# 데이터

| Date | Close | Open | High | Low | Volume | Change |
|---|---|---|---|---|---|---|
| **2018-01-01** | 93500 | 93500 | 93500 | 93500 | 0.0 | 0.0000 |
| **2018-01-02** | 102500 | 95900 | 104000 | 93300 | 6760000.0 | 0.0963 |
| **2018-01-03** | 103000 | 102600 | 104900 | 99500 | 4720000.0 | 0.0049 |
| **2018-01-04** | 92200 | 102600 | 104000 | 92200 | 6390000.0 | -0.1049 |
| **2018-01-05** | 100000 | 85800 | 101200 | 85700 | 8250000.0 | 0.0846 |
| **2018-01-07** | 100000 | 100000 | 100000 | 100000 | 0.0 | 0.0000 |
| **2018-01-08** | 93800 | 98000 | 98400 | 92500 | 6280000.0 | -0.0620 |
| **2018-01-09** | 109000 | 96500 | 119200 | 93800 | 12290000.0 | 0.1620 |
| **2018-01-10** | 98000 | 105000 | 107300 | 97800 | 6510000.0 | -0.1009 |
| **2018-01-11** | 96700 | 97600 | 100600 | 95200 | 4040000.0 | -0.0133 |

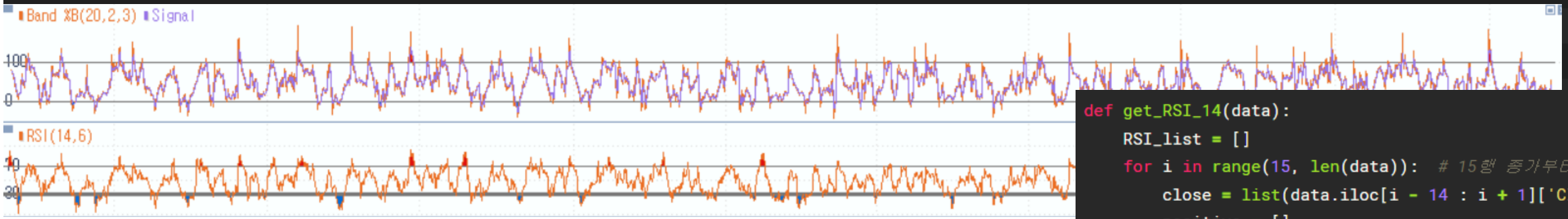
## 데이터 출처

주로 FinanceDataReader 라이브러리 사용
가격, 거래량, 종목 구성 등의 정보를 바로 가져올 수 있음
추가로 한국거래소, pykrx 라이브러리 등을 사용



# 1. 차트 지표

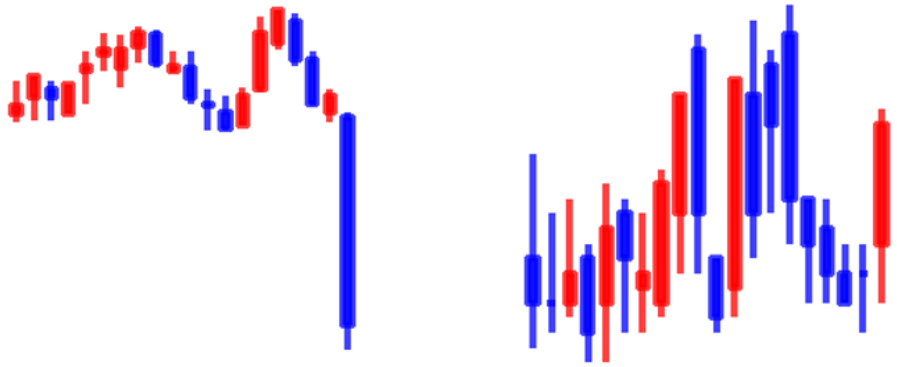
| RSI | STOCASTIC_K | STOCASTIC_D | Bollinger | MACD | MACD_cat | MACD_SIGNAL | RSI_delta | D_delta | sto_diff | B_delta | MACD_delta | MA5 | MA20 | MA5_adj | MA20_adj |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.448341 | 0.528217 | 0.417295 | 0.552657 | 55.980216 | 0 | 74.979173 | -0.030153 | 0.055461 | 0.110922 | 0.043872 | 1.053686 | 6248.0 | 6297.45 | -0.011236 | -0.003410 |
| 0.368376 | 0.173815 | 0.336135 | 0.165915 | 43.888930 | 0 | 68.761036 | -0.079965 | -0.081160 | -0.162320 | -0.386742 | -12.091285 | 6226.4 | 6297.90 | 0.010451 | 0.022055 |




```python
def get_RSI_14(data):
    RSI_list = []
    for i in range(15, len(data)):  # 15행 종가부터 시작
        close = list(data.iloc[i - 14 : i + 1]['Close'])
        positive = []
        negative = []
        for j in range(14):
            diff = close[j + 1] - close[j]
            if diff >= 0:
                positive.append(diff)
            else:
                negative.append(diff)

        AU = np.sum(positive) / 13
        AD = abs(np.sum(negative) / 13)
        RSI = AU / (AU + AD)
        RSI_list.append(RSI)
```

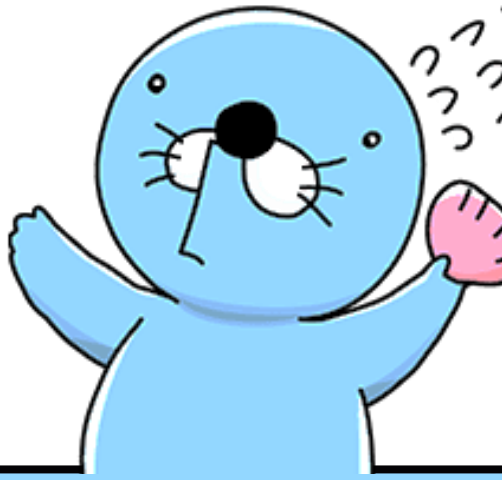
주가 흐름 상 현재 주가가 어디에 있는지를 나타내는 지표들을 사용
RSI / STOCASTIC / Bollinger Band(%b) / MACD / MA5, MA20
열이 많아서 과최적화 + 학습이 느림
개별주 차원으로 학습량 추가(1.5m) + 파라미터 조정 없이도
어느 정도 성능이 보장되는 Catboost 사용

# 2. 캔들 차트



```
label = (data.iloc[i]['Change'] > 0).astype(int)

fig = plt.figure(figsize=(1, 1), dpi = 300)

ax = fig.add_subplot(111)

candlestick2_ohlc(ax, local_data['Open'], local_data['High'],

plt.axis('off')

  plt.savefig('drive/My Drive/datasets/datasets/{}_{}_{}.png'.f
if label == 1:
    plt.savefig('datasets/1/{}.png'.format(variable))
else:
    plt.savefig('datasets/0/{}.png'.format(variable))
```

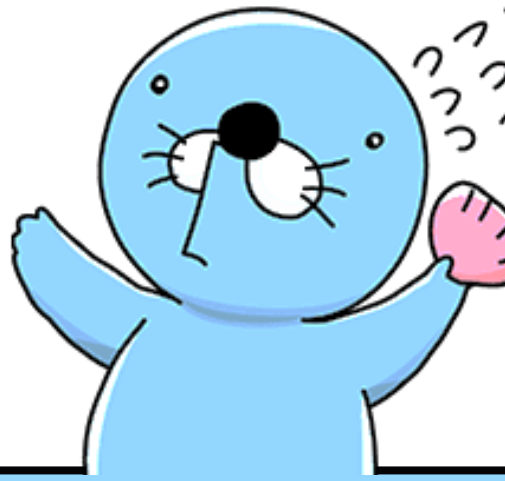
CNN을 활용한 관련 연구 많이 있었음
: CNN을 통해 N일 전까지의 주가 봉 차트를 학습 -> 주가 상 / 하방 예측
Mplfinance 라이브러리 사용 : 봉 차트를 생성(20일)
-> CNN 모델 생성 후 학습(50,000)



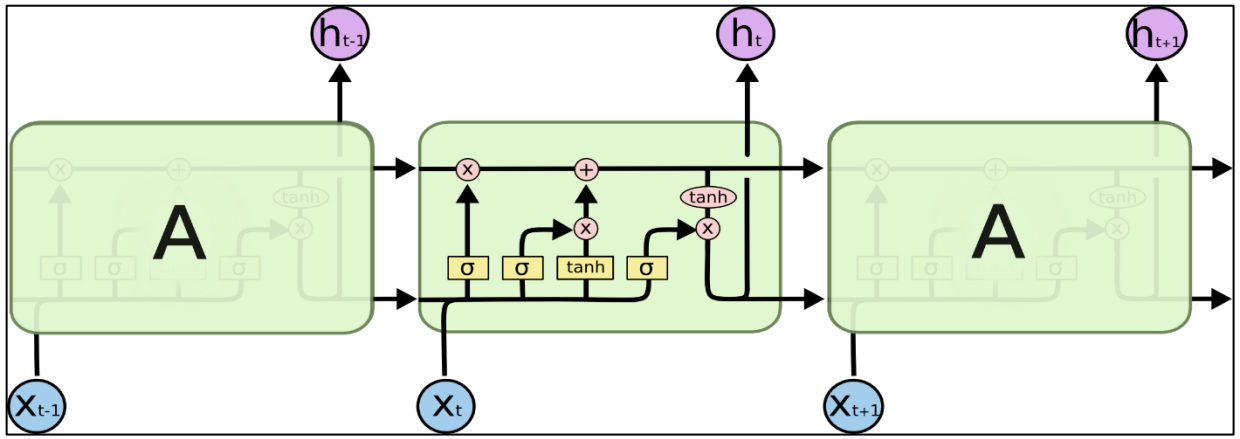
# 3. 수급

| 일자 | 지수 | 등락폭 | 개인 | 외국인 | 기관 |
|---|---|---|---|---|---|
| 2022/05/31 | 2,666.10 ▼ | 3.56 | 283 | 275 | -598 |
| 2022/05/30 | 2,669.66 ▲ | 31.61 | -8,021 | 3,540 | 4,203 |
| 2022/05/27 | 2,638.05 ▲ | 25.60 | -7,447 | 2,759 | 5,543 |
| 2022/05/26 | 2,612.45 ▼ | 4.77 | 930 | 376 | -1,546 |
| 2022/05/25 | 2,617.22 ▲ | 11.35 | -3,581 | -1,654 | 5,112 |
| 2022/05/24 | 2,605.87 ▼ | 41.51 | 5,786 | -3,254 | -2,820 |
| 2022/05/23 | 2,647.38 ▲ | 8.09 | -1,652 | -303 | 1,697 |
| 2022/05/20 | 2,639.29 ▲ | 46.95 | -10,412 | 1,966 | 8,364 |

국내 주식시장은 투자주체를 크게 세 가지로 구분함 **개인 / 외국인 / 기관**
지수를 견인하는건 '외국인'
개별주를 견인하는건 '투신' -> 투자주체와 주가 간 유의미성을 예측
SVC를 활용하여 모델링 : 크게 유의하지 않은 것으로 나타남
**예측 모델 및 추후 보팅에서 제외**



# 4. 시계열

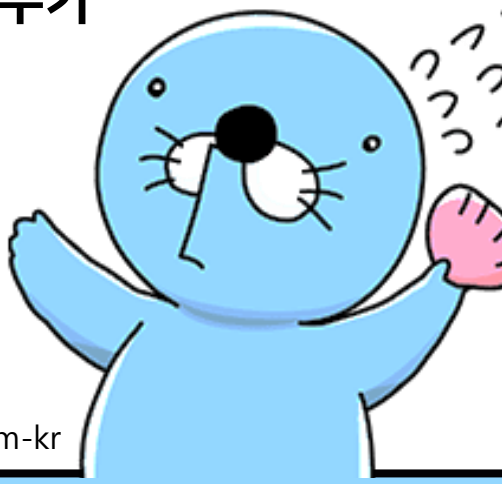



```
#Build the LSTM model
model = Sequential()
model.add(LSTM(50,return_sequences=True, input_shape=(x_train.shape[1],6)))
model.add(LSTM(50, return_sequences=False))
model.add(Dense(25))
model.add(Dense(1))
#Compile the model
model.compile(optimizer='adam', loss='binary_crossentropy',metrics=['accuracy'])
```

## LSTM

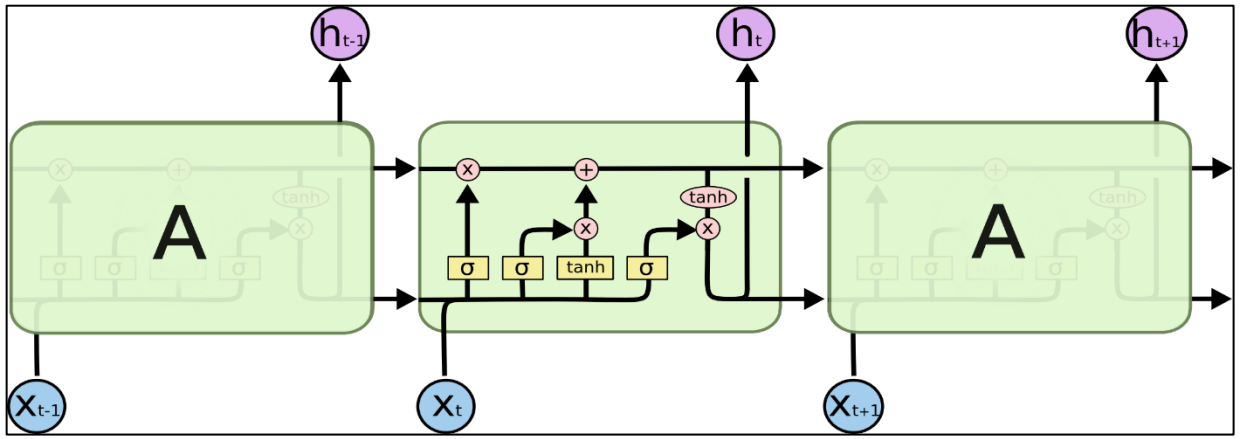
RNN의 한 종류이며 긴 기간의 의존성(long-term dependencies)이 필요한 문제를 해결하기 위해 고안된 모델.
다양한 시계열 데이터 해석 문제에 사용됨.

간단한 LSTM 모델을 사용하여 주가 차트 내용을 바탕으로 분석.



그림출처 : https://dgkim5360.tistory.com/entry/understanding-long-short-term-memory-lstm-kr

# 4. 시계열




```
for i in range(60, len(train_data)):
    x_train.append(train_data.iloc[i-60:i,0:6])
    y_train.append(train_data.iloc[i,6])
```

```
#Convert the train data to numpy arrrays
x_train, y_train = np.array(x_train), np.array(y_train)
```

ed in 104ms, finished 01:08:31 2022-05-27

```
#Reshape the data
x_train = np.reshape(x_train, (x_train.shape[0], x_train.shape[1],6))
x_train.shape
```

## LSTM

RNN의 한 종류이며 긴 기간의 의존성(long-term dependencies)이 필요한 문제를 해결하기 위해 고안된 모델.
다양한 시계열 데이터 해석 문제에 사용됨.

시계열 데이터이므로 X_data에 시간열을 넣어주기 위해 다음과 같은 전처리 작업을 거쳐 (N,60,6) 형태로 변형

그림출처 : https://dgkim5360.tistory.com/entry/understanding-long-short-term-memory-lstm-kr

# 4. 시계열

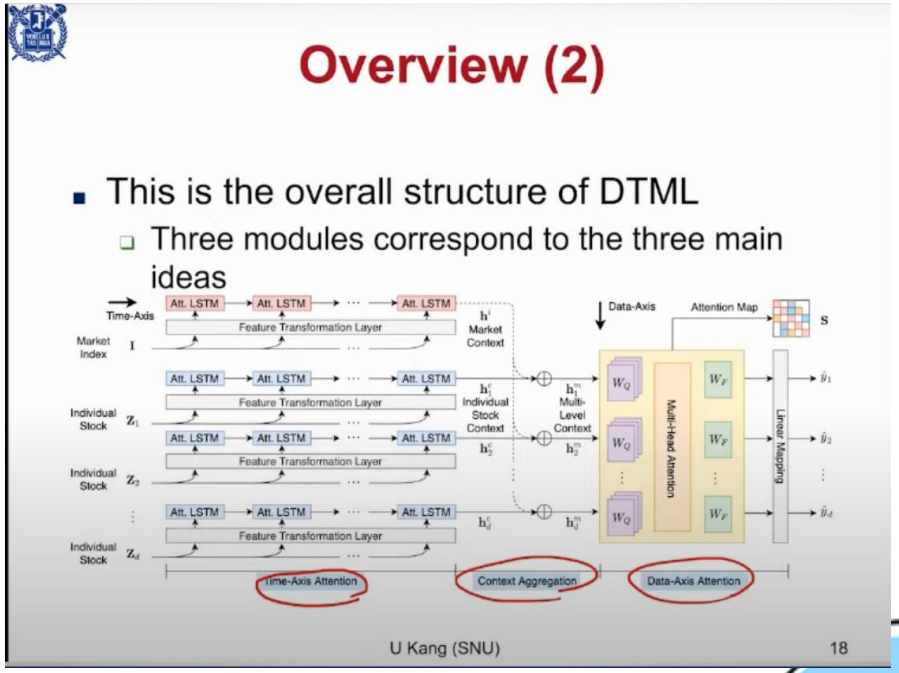
```
result = model.evaluate(x_test, y_test, batch_size=1)
```

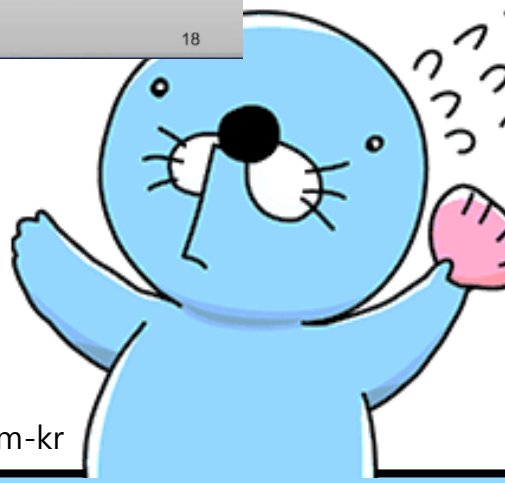
executed in 6.96s, finished 01:17:56 2022-05-27

```
100/100 [==============================] - 7s 6ms/step - loss: 6.7870 - accuracy: 0.5600
```

간단하게 작성된 모델이라 정확도가 0.56~0.6 정도로 나옴
데이터 측면 : OHLCV만 사용된 데이터라 장의 분위기 정도 밖에 읽지 못하였다.
모델적 측면 : TCN(Temporal Convolutional Network), Bi-LSTM(bidirectional-LSTM), attention mechanism,DTML(Data-axis Transformer with Multi-Level contexts) 등 다양한 모델을 적용한다면 성능이 올라 갈 것이라 생각.

# LSTM



그림출처 : https://dgkim5360.tistory.com/entry/understanding-long-short-term-memory-lstm-kr

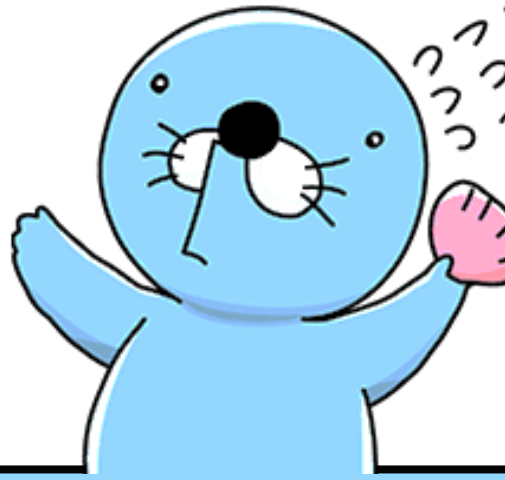
# 5. 자연어 / 거시경제

| | Open | High | Low | Close | Volume | Change | Target | NASDAQ | DJI | GOLD | KRW | EUR | CNY | sentiment |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2007-01-03 | 14451 | 14455 | 14195 | 14232 | 1106268 | -0.014268 | 0 | 2423.16 | 12474.52 | 627.1 | 926.15 | 0.7597 | 7.8160 | 0.000000 |
| 2007-01-05 | 14058 | 14065 | 13818 | 13929 | 2322080 | -0.007128 | 0 | 2434.25 | 12398.01 | 604.9 | 934.45 | 0.7689 | 7.8090 | 1.000000 |
| 2007-01-08 | 13907 | 13907 | 13727 | 13783 | 1869083 | -0.010482 | 0 | 2438.20 | 12423.49 | 607.5 | 938.10 | 0.7680 | 7.8127 | 1.000000 |
| 2007-02-02 | 13916 | 14214 | 13916 | 14208 | 1357483 | 0.021350 | 1 | 2475.88 | 12653.49 | 646.2 | 937.25 | 0.7712 | 7.7560 | -1.000000 |
| 2007-02-05 | 14195 | 14284 | 14113 | 14300 | 1572882 | 0.006475 | 1 | 2470.60 | 12661.74 | 650.9 | 935.80 | 0.7733 | 7.7585 | 0.000000 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 2022-05-16 | 34850 | 34925 | 34395 | 34470 | 7376786 | -0.003901 | 0 | 11662.79 | 32224.01 | 1814.0 | 1280.38 | 0.9583 | 6.7822 | -0.454545 |
| 2022-05-17 | 34525 | 34850 | 34520 | 34805 | 3999458 | 0.009719 | 1 | 11984.52 | 32655.05 | 1818.9 | 1266.50 | 0.9478 | 6.7337 | -0.400000 |
| 2022-05-18 | 34990 | 35145 | 34770 | 34870 | 6376438 | 0.001868 | 1 | 11418.15 | 31493.56 | 1815.9 | 1275.33 | 0.9551 | 6.7540 | 0.071429 |
| 2022-05-19 | 34125 | 34450 | 34060 | 34370 | 6054919 | -0.014339 | 0 | 11388.50 | 31253.26 | 1841.2 | 1262.61 | 0.9442 | 6.7085 | -0.800000 |
| 2022-05-20 | 34515 | 35025 | 34485 | 34985 | 4661990 | 0.017894 | 1 | 11354.62 | 31260.58 | 1842.1 | 1273.59 | 0.9466 | 6.6921 | -0.333333 |

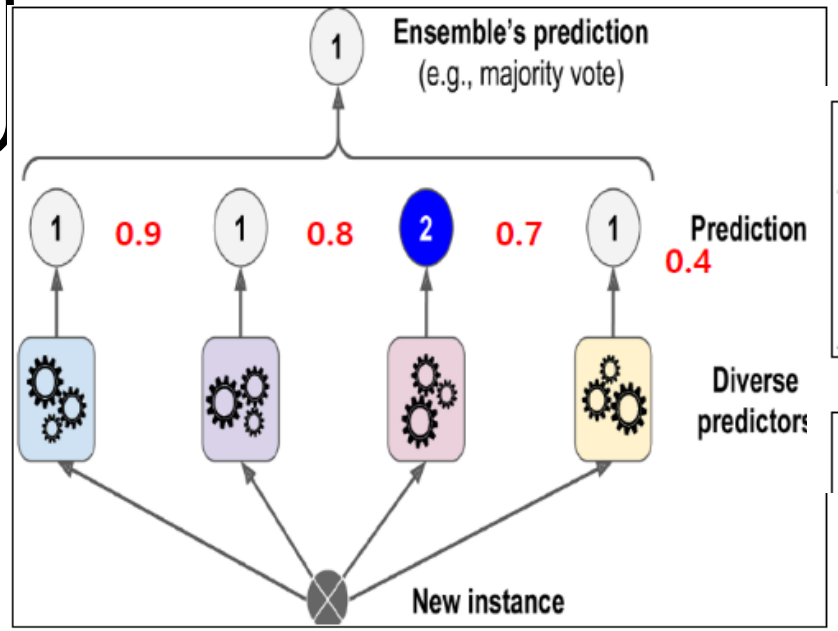
1. 나스닥 / 다우 / 금 / 유로 / 위안화 등 거시경제 지표 사용
2. KoBERT를 사용하여 그 날 네이버 뉴스 감성 분석 진행
   -> Sentiment 지수로 나타내어 변수로 사용

⇒ LSTM 모델을 활용해 주가 향방 학습 ( Target 열)

결이 조금 다르다고 판단, 개별 모델로써는 사용하지만
보팅 모델에는 포함하지 않았음



# 주요 성능



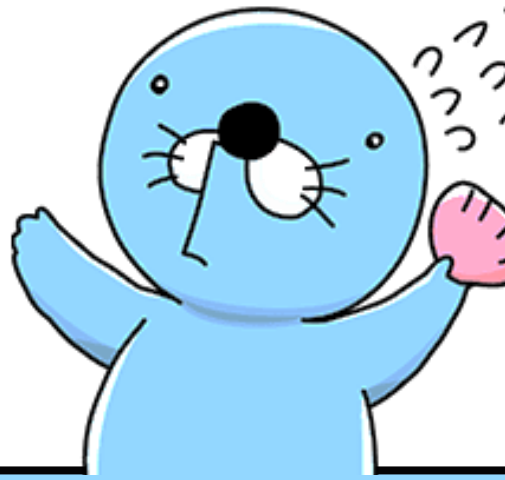

$$p(i_1 \mid \mathbf{x}) = \frac{0.9+0.8+0.3+0.4}{4} = 0.6$$

$$p(i_2 \mid \mathbf{x}) = \frac{0.1+0.2+0.7+0.6}{4} = 0.4$$

$$\hat{y} = \arg\max_i [p(i_1 \mid \mathbf{x}), p(i_2 \mid \mathbf{x})] = 1$$

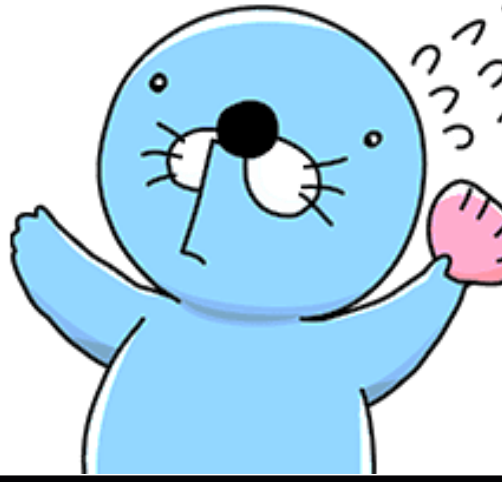
1. 차트 지표 모델
2. CNN을 통한 캔들 차트 분석 모델
3. LSTM을 통한 시계열 분석 모델

-> Soft Voting을 통한 예측 모델 생성

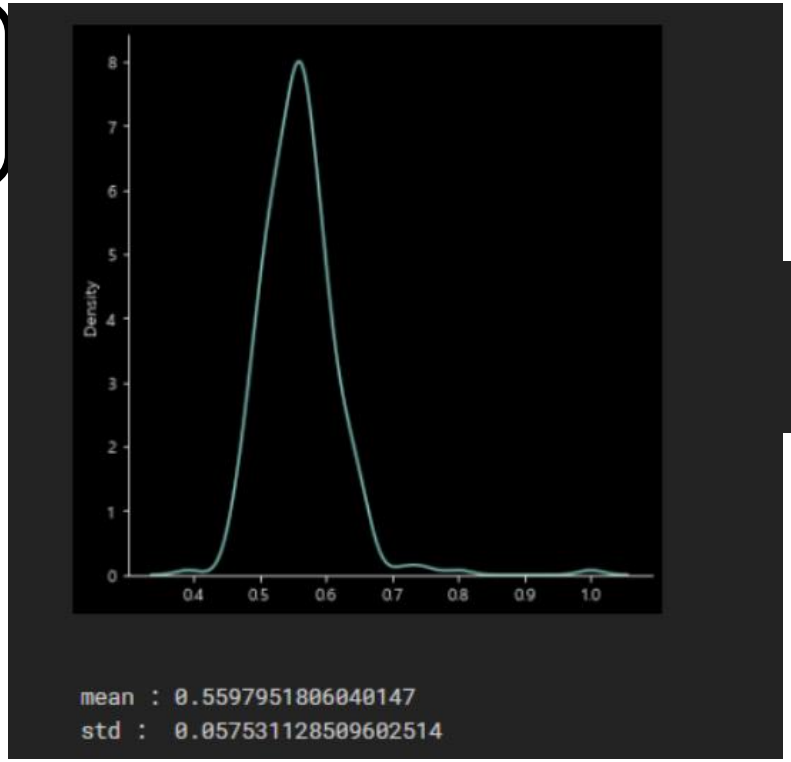



1. KODEX 200의 상승 / 하락 예측
2. 개별주 차원으로 확장

오늘의 지수 예측을 '투자 날씨'로써 표현
+ 특정 모델을 과거 데이터를 바탕으로 실험하는 백테스팅 기능으로 구현

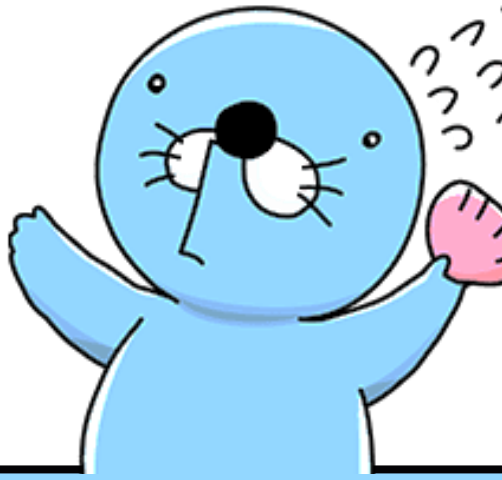


# 주요 성능

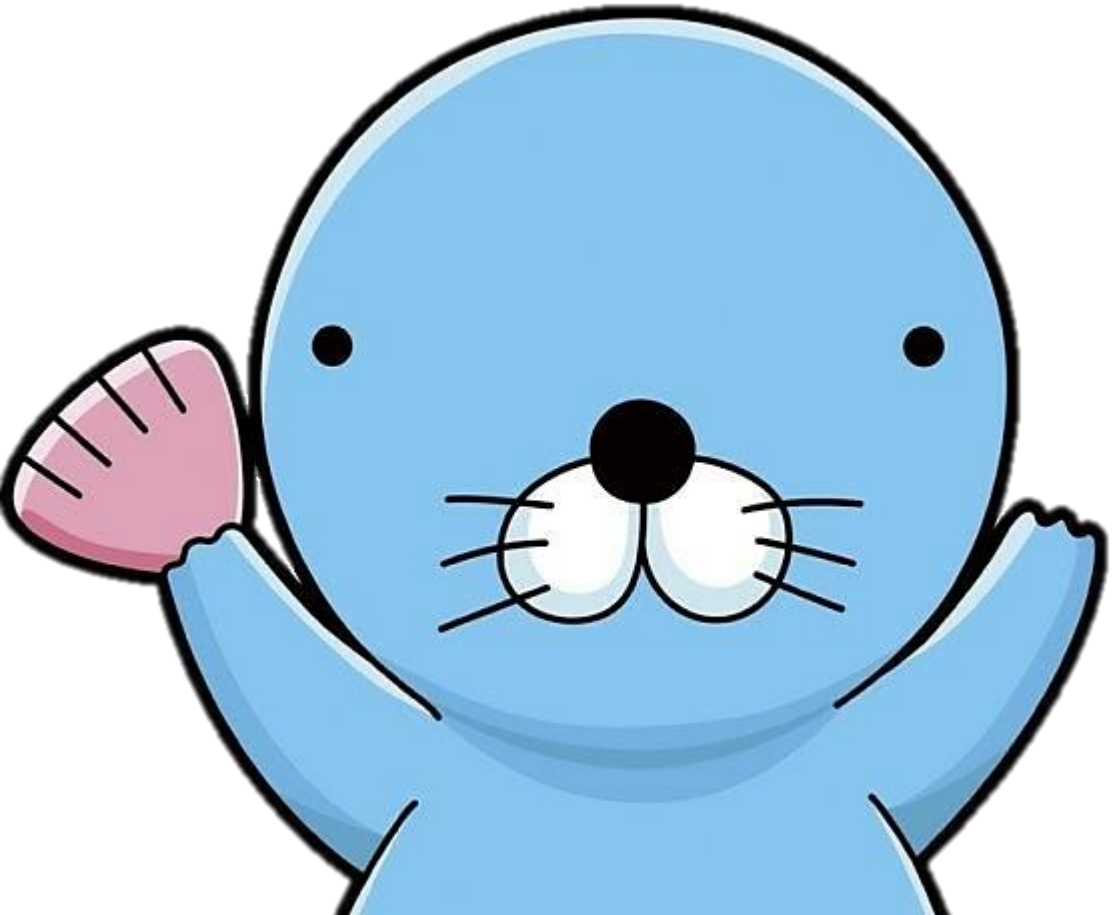



캔들 차트 모델의 2022년 코스피 예측률은 : 0.5463917525773195
차트 지표 모델의 2022년 코스피 예측률은 : 0.5360824742268041
LSTM 모델의 2022년 코스피 예측률은 : 0.5
보팅 모델의 2022년 코스피 예측률은 : 0.5463917525773195

- 백테스팅 기능을 구현해 놓아서, 사용자가 직접 인터랙티브하게 성능 측정 가능
- 대표격으로 KODEX 200의 2022년 이후 accuracy를 테스트함
- Cross-entropy / accuracy 모두 보팅 모델이 우수한 결과를 보임
- 그래서 보팅 모델을 대상으로 2022년 이후 국내 개별주 500개에 대한 성능을 테스트 : MEAN 0.56 / STD : 0.057



# 아쉬운 점



- 성능이 덜 나오는 모델 성능 향상 집착

- 생각보다 많은 모델을 구현하지 못함

- 구현한 모델도 성능 평가에 있어
시계열적 흐름 / 인풋 추가 옵션을 반영하지 못함

- 퀄리티 나쁜 논문들 / 학회지들에 속았음

- 결국 시간부족