

인공지능 기반의 주식데이터 분석 및 주가예측

Analysis and prediction of
stock data based on artificial intelligence



Work Team Name & Members

파닥팀

장형욱(팀장), 김도현, 이동한, 장성호, 장연식

Work Schedule

2021-08-24 : 데이터 전처리 구현하기 위한 자료조사

2021-08-31 : 전처리 및 레퍼런스 수집공부

2021-09-11 : 유망주 주식현재가 선별 및 시스템 구현

2021-09-20: 발표준비

Work Rule

Data Set : Training, FinanceDataReader

인공지능 기반기술

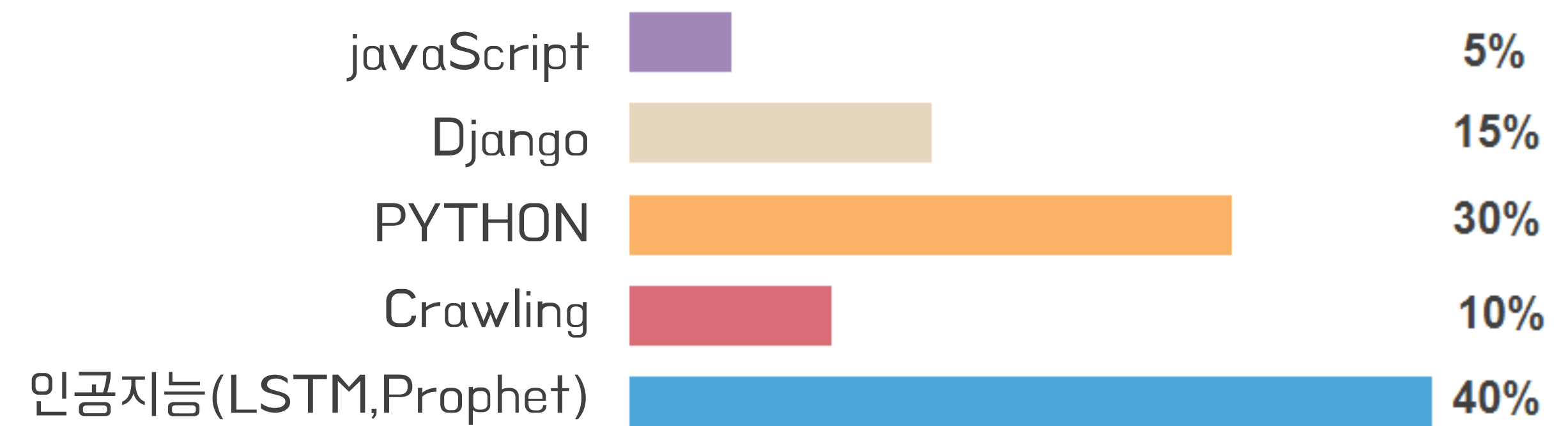


Skills

인공지능 기반의 주식데이터 분석 및 주가 예측

■ COVID - 19 발생 기점으로 테마별 주식종목 선정 및 분석

■ LSTM, Prophet 모델을 적용한 주가예측



1.개요

1.개요

2.분석 프로세스

3.개발

4.서비스

5.마무리

개요

- 1.팀원 소개
- 2.주제 선정 배경 및 목적
- 3.개발 환경

분석 프로세스

- 1.데이터 수집

개발

- 1.LSTM
- 2.Prophet
- 3.감성분석

서비스

- 1.FrameWork
- 2.부트스트랩

마무리

- 1.기대효과 & 아쉬운 점
- 2.데모 시연

1.개요

1.팀원 소개

1.개요

2.분석 프로세스

3.개발

4.서비스

5.마무리

Team member 1



DA / 장형욱

Team member 2



DA / 김도현

Team member 3



AA / 이동한

Team member 4



AA / 장성호

Team member 5



TA / 장연식

1.개요

2.주제 선정
배경 및 목적

1.개요

2.분석 프로세스

3.개발

4.서비스

5.마무리

(시리즈) 경제 비즈니스

주식투자자 '세대교체' 온다...MZ세대·미성 년자 비중 확대

뉴스 > 경제 > 금융

취준생도... 직장인도... 2030 주식투자 열풍

세종=구특교 기자, 김자현 기자 입력 2020-07-18 03:00 수정 2020-07-18 03:41

코로나 이후 '동학개미운동'
넘쳐나는 시중 유동성도 영향
청년층 '재테크 스테디' 부쩍 늘어

<https://www.donga.com> > news > Economy > article > all ▼

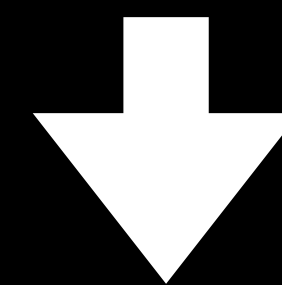
국민 5명중 1명, 코로나 이후 주식투자 시작...절반은 '2030'

뉴스핌 - "부자 2명 중 1명, 코로나 이후 주식투자 늘렸다"

2021. 3. 8. — 부자의 53%, 대중부유층의 48%가 코로나19 이후 주식 비중을 늘렸다.

투자자 중 절반(48%)은 “투자지식 낮은 상태”

주식 관심도 증가
+
무지성 투자



주식 동향 파악 정보 제공

1.개요

2.분석 프로세스

3.개발

4.서비스

5.마무리

1.개요
3.개발 환경

django

API



TensorFlow



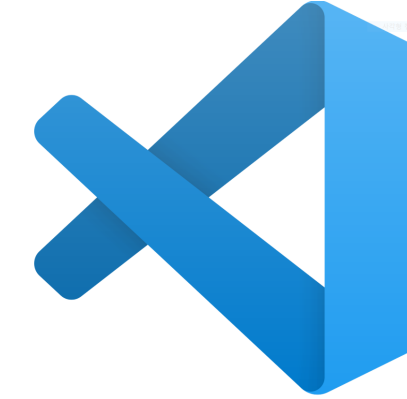
FinanceDataReader



pandas



Front-End



plotly

2.분석 프로세스

1.데이터

1.개요

2.분석 프로세스

3.개발

4.서비스

5.마무리



FinanceDataReader

한국 주식 가격, 미국 주식 가격, 지수, 환율, 암호화폐 가격, 종목 리스팅 등
금융 데이터 수집 라이브러리

2.분석 프로세스

1.데이터

1.개요

2.분석 프로세스

3.개발

4.서비스

5.마무리

```
21 def start_predict(self, theme, company, modelString, predict_term):
22     code = ''
23     from_date = '2020-01-20'
24
25     if theme == 'game':
26         if company == '엔씨소프트':
27             code = '036570'
28         if company == '넷마블':
29             code = '251270'
```

```
62 df = fdr.DataReader(code, from_date)
```

-직관적

-사용이 간편

2.분석 프로세스

1.데이터

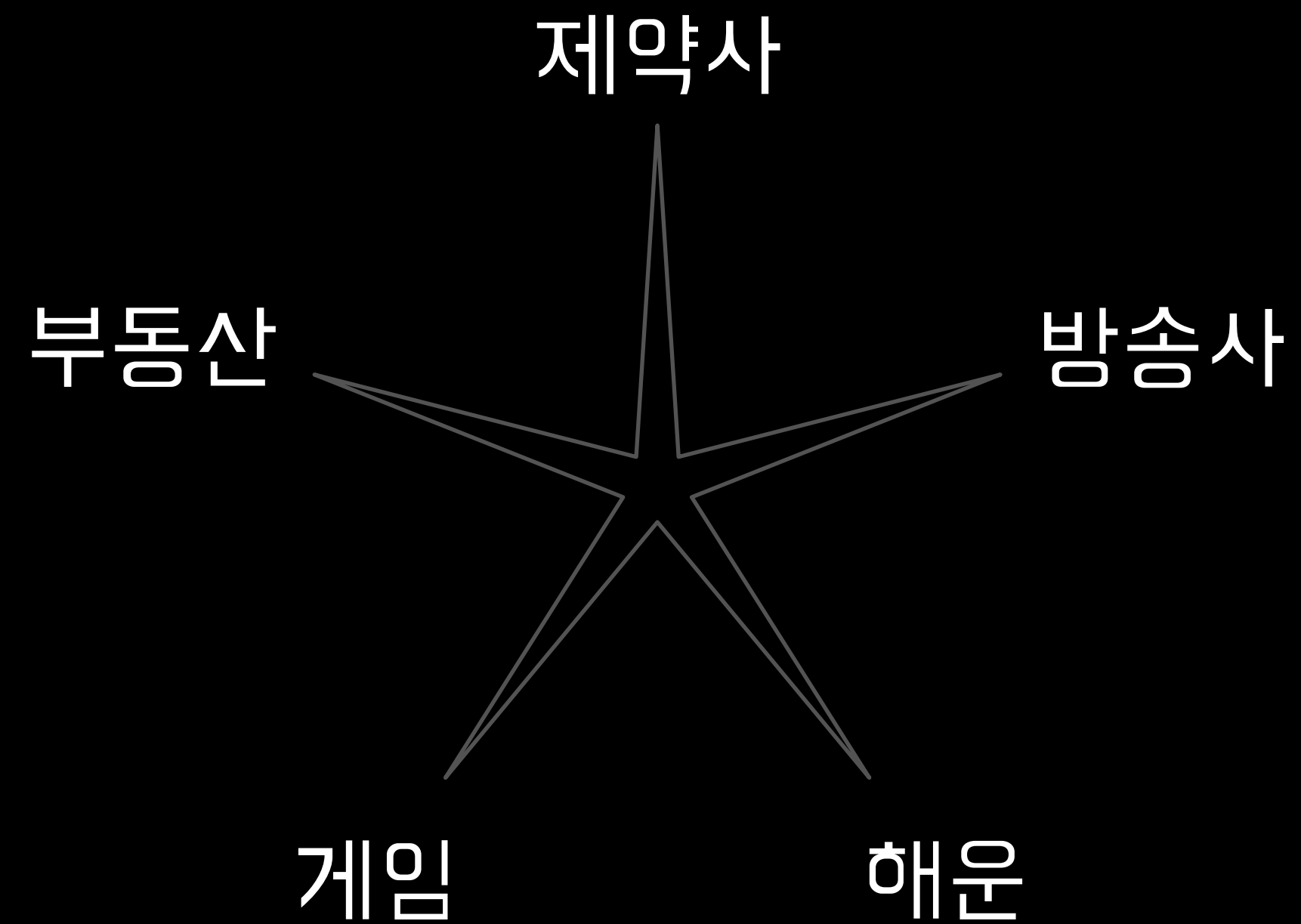
1.개요

2.분석 프로세스

3.개발

4.서비스

5.마무리



데이터 수집기간
2020.01.20 - 2021.09.15

3.개발

1.LSTM

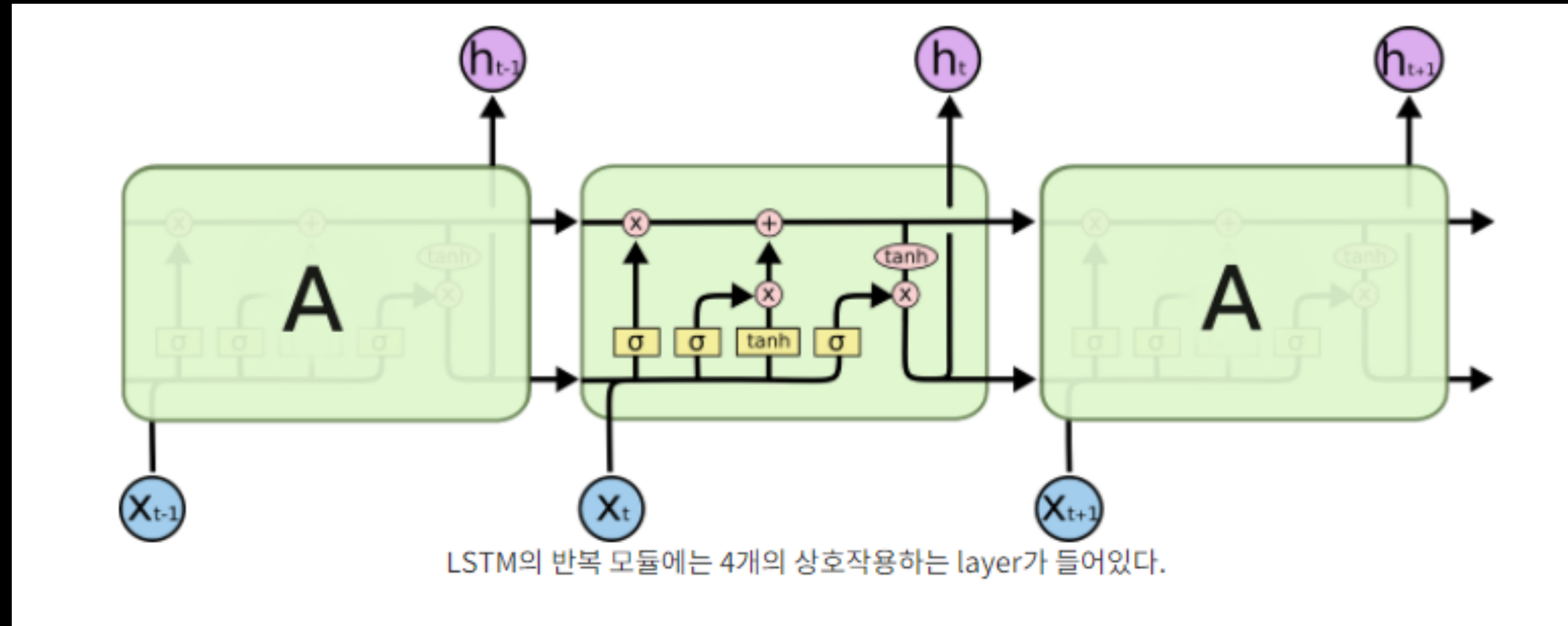
1.개요

2.분석 프로세스

3.개발

4.서비스

5.마무리



LSTM (Long Short Term Memory) : 장/단기 기억을 가능하게 설계한 신경망의 구조
시계열 처리나 자연어 처리에 사용

3.개발

1.LSTM

1.개요

2.분석 프로세스

3.개발

4.서비스

5.마무리

model.py ×

main > predict_model > model.py > PredictModel > lstm

```
96     model = Sequential()
97     model.add(
98         LSTM(50,
99             activation='elu',
100             return_sequences=True,
101             input_shape=(look_back, 1))
102     )
103     model.add(Dropout(0.2))
104     model.add(LSTM(50, activation='tanh'))
105     model.add(Dropout(0.2))
106     model.add(Dense(1))
107     sgd = tf.keras.optimizers.SGD(learning_rate=0.01,
108                                   decay=1e-7,
109                                   momentum=0.9,
110                                   nesterov=False)
111     model.compile(optimizer=sgd, loss='mean_squared_error')
```

3.개발

1.LSTM

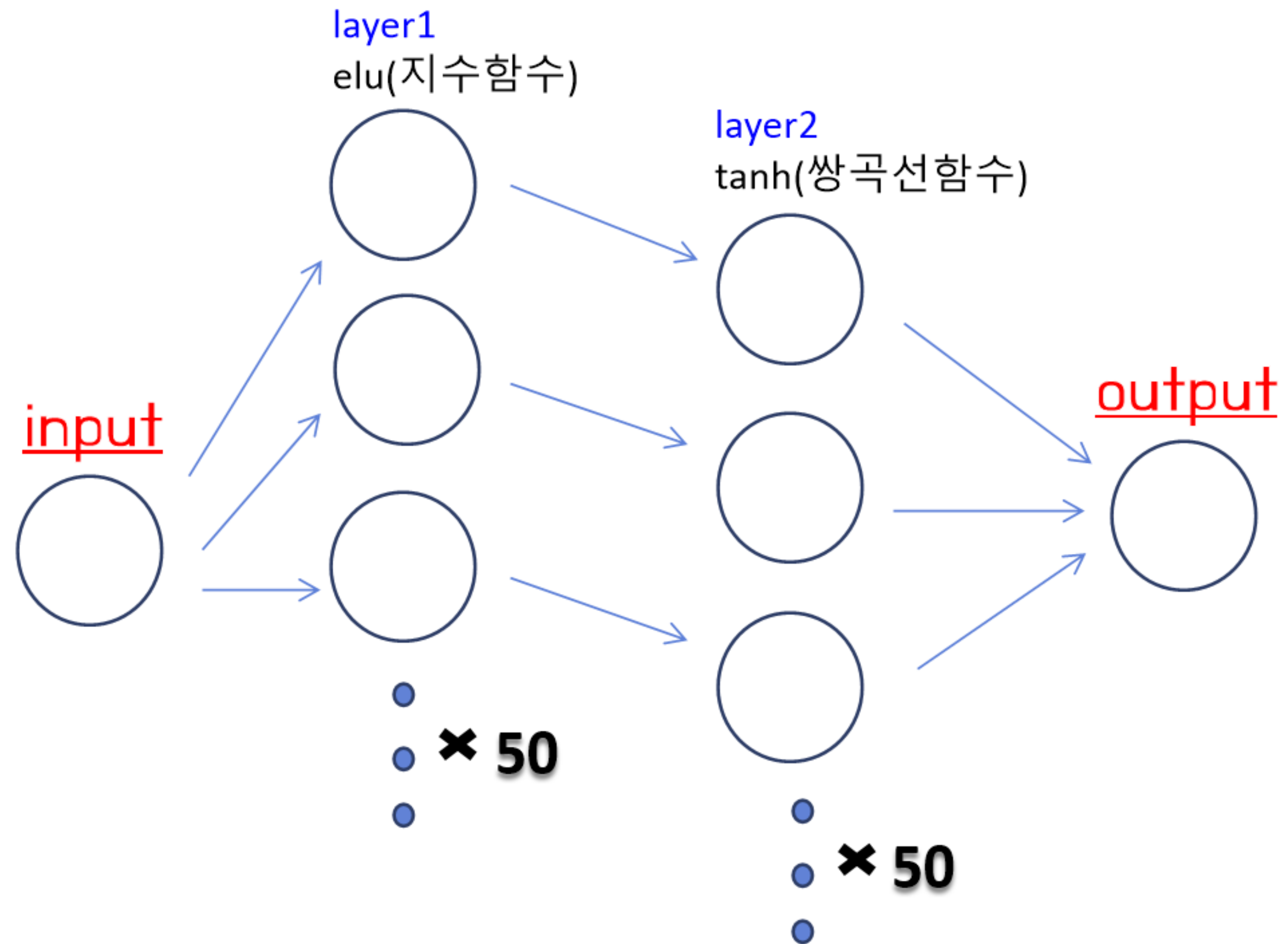
1.개요

2.분석 프로세스

3.개발

4.서비스

5.마무리



3.개발
2.Prophet

1.개요

2.분석 프로세스

3.개발

4.서비스

5.마무리

PROPHET

-Facebook 오픈소스

-일변량(변수1개)의 시계열

-추세, 계절성, 휴일

-속도, 접근성

1.개요

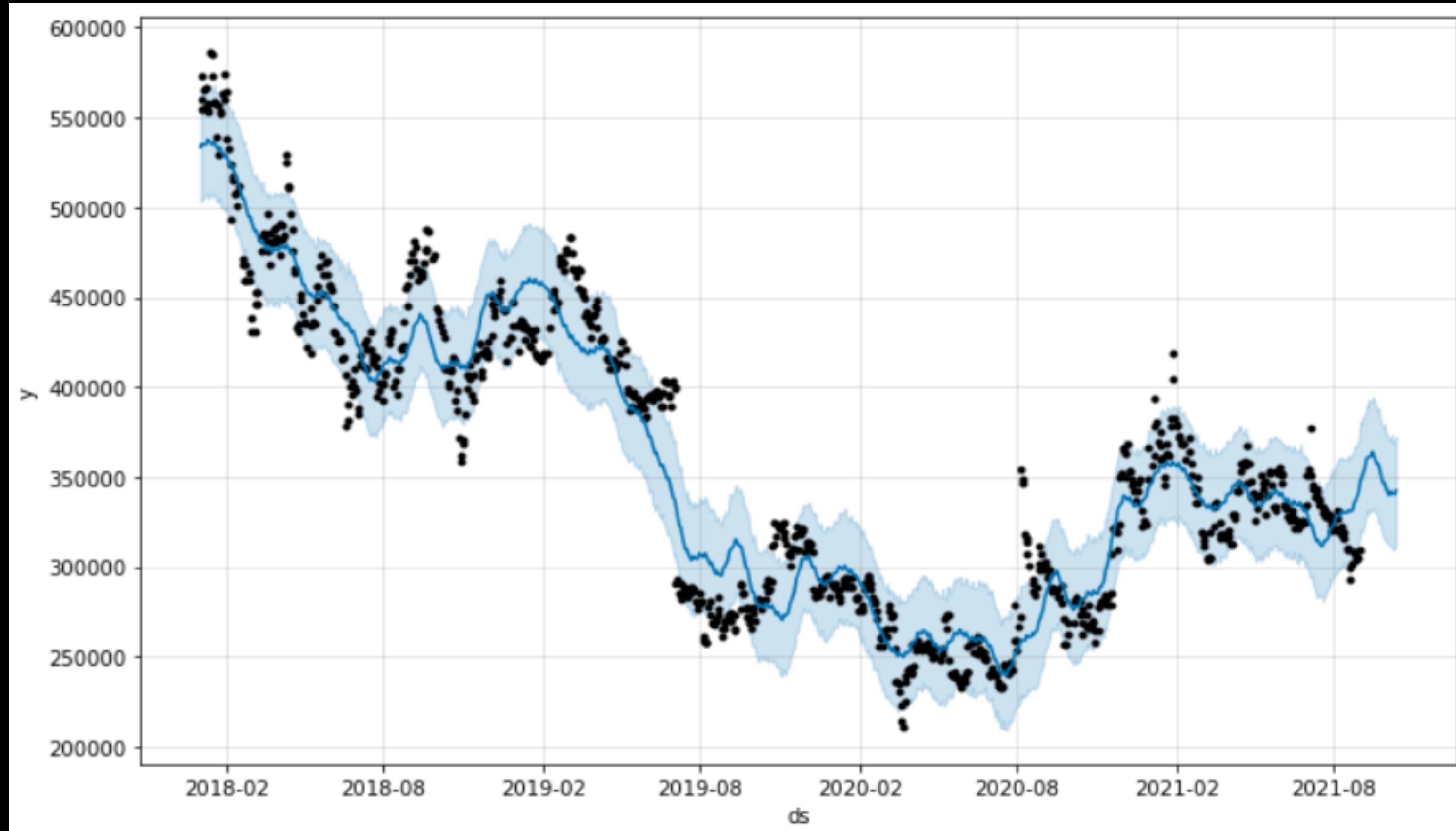
2.분석 프로세스

3.개발

4.서비스

5.마무리

3.개발
2.Prophet



3.개발
3.감성분석

1.개요

2.분석 프로세스

3.개발

4.서비스

5.마무리

감성분석

텍스트에 들어있는 의견이나 감성, 평가, 태도 등의 **주관적인** 정보를
컴퓨터를 통해 분석하는 과정

네이버 증권 기사 제목과 주가의 증감을 활용하여 미래 주가 예측

TF-IDF
Vectorizer

Logistic
Regression

Pipeline

3.개발

3.감성분석

1.개요

2.분석 프로세스

3.개발

4.서비스

5.마무리

```
# 새로운 칼럼 생성
# (Price : 당일 대비 다음날 주가가 상승했으면 1, 하락했으면 0 표시)
df['Price'] = 0
for i in range(df.shape[0]-1):
    if df['Close'][i] < df['Close'][i+1]:
        df['Price'][i] = 1
    else:
        df['Price'][i] = 0

# 파일 저장
df.to_csv('Analysis/{_주가데이터}.csv'.format(company))
```

```
# 수집날짜 리스트 생성
price_data = pd.read_csv('Analysis/{_주가데이터}.csv'.format(company))
df_0 = price_data[price_data['Price']==0]['Date']
date_0 = []
for i in range(0, len(df_0)):
    date_0.append(str(df_0.tolist()[i]))

df_1 = price_data[price_data['Price']==1]['Date']
date_1 = []
for i in range(0, len(df_1)):
    date_1.append(str(df_1.tolist()[i]))
```

-주가 데이터 불러오기

-뉴스 제목 크롤링

-Concat -> csv save

1.개요

2.분석 프로세스

3.개발

4.서비스

5.마무리

3.개발

3.감성분석

뉴스제목	주가변동
정부 세탁기 세이프가드 공청회서 반대 입장 피력	0
정부 삼성 LG 세탁기 세이프가드서 제외해야	0
이총리 최저임금 인상 노동시간 단축 정책 연착륙 노력	0
박용만 끊임없는 혁신이 미래 성장 담보하는 유일한 길	0
이재용 삼성전자 부회장 보아오 포럼 이사 물러날듯	0
삼성전자 썬더볼트 지원 QLED 커브드 모니터 선봬	0
금융혁신으로 경제성장 뒷받침 금융사 경쟁력 위해 규제혁파 필요	0
머니S토리 달아오른 M A시장 돌격 앞으로	0
법제처로 간 이견희 차명계좌	0
박용만 기업이 많은 일 벌일 수 있게 제도 정책 설계를	0
브리핑 현대차 국내 최초 보증수리 조건 선택제 도입	0
이주열 한계기업 구조조정 적기 박용만 기업들 일 벌일 수 있게	0
코스닥 공시 슈프리마 원익IPS 선익시스템	0
이낙연 총리 성장 막는 규제 없앨 것 경제계 신년인사회 열려	0
원달러 하락에 비상 걸린 수출기업들 수입업계도 불안불안	0
서울경제TV 증시 월 효과 기대 쑥쑥 정책 수혜주에 주목	0
삼성전자 SK하이닉스 실적 발표 임박 연일 강세	0
비즈 삼성 QLED 커브드 모니터 CES서 공개	0
검 경 황창규 KT 회장 정조준 내부 위기감 확산	0
수출주 환율 사드 부담덜고 최대 실적으로 강세장 준비	0
포토 전면에 듀얼캠 갤럭시A 출시	0

-주가 데이터 불러오기

-뉴스 제목 크롤링

-Concat -> csv save

3.개발

3.감성분석

1.개요

2.분석 프로세스

3.개발

4.서비스

5.마무리

```
sentiment.py ×
main > predict_model > sentiment.py > SentimentModel > start_sentiment
21 def start_sentiment(self, company):
22     headers = {
23         'X-Naver-Client-Id' : 
24         'X-Naver-Client-Secret' : 
25     }
26
27     query = company
28     display = 10
29     params = {
30         'query': query,
31         'display': display,
32         'start': 1,
33         'sort': 'sim',
34     }
35
36     naver_news_url = 'https://openapi.naver.com/v1/search/news.json'
37
38     res = requests.get(naver_news_url, headers=headers, params=params)
39     if res.status_code == 200:
40         news = res.json().get('items')
```

1 전처리



2 형태소 분석



3 딕셔너리화



HIGHCHARTS

1.개요

2.분석 프로세스

3.개발

4.서비스

5.마무리

3.개발

3.감성분석

```
sentiment.py ×  
main > predict_model > sentiment.py > SentimentModel > start_sentiment  
72 stop_words = ['\n', '키로', '부터']  
73 tokens_ko = [each_word for each_word in tokens_ko if each_w  
74 ko = nltk.Text(tokens_ko)  
75  
76 data = ko.vocab().most_common(20)  
77 data = [each_word for each_word in data if len(each_word[0]  
78 cloud_data = []  
79 for i in data:  
80     cloud_data.append(dict(x=i[0],value=i[1]))  
81  
82 using_text = []  
83 for i in ko:  
84     using_text.append(i)
```



HIGHCHARTS

1.개요

2.분석 프로세스

3.개발

4.서비스

5.마무리

3.개발

3.감성분석

```
loc_csv = 'data/crawling_data/{company}_뉴스타이틀.csv'.format(company)

def tokenizer(text):
    okt = Okt()
    return okt.morphs(text)

def data_preprocessing(csv):
    news_df = pd.read_csv(csv)
    title_list = news_df['뉴스제목'].tolist()
    price_list = news_df['주가변동'].tolist()
    title_train, title_test, price_train, price_test = train_test_split(title_list, price_list, test_size=0.2, random_state=0)
    return title_train, title_test, price_train, price_test

def learning(x_train, y_train, x_test, y_test):
    tfidf = TfidfVectorizer(lowercase=False, tokenizer=tokenizer)
    logistic = LogisticRegression(C=3, penalty='l2', random_state=0, solver='lbfgs', max_iter=1000)
    pipe = Pipeline([('vect',tfidf), ('clf',logistic)])
    pipe.fit(x_train, y_train)
    y_pred = pipe.predict(x_test)
    joblib.dump(pipe, 'data/sentiment_model/{company}_pipe.pkl'.format(company))

def model_learning(csv):
    title_train, title_test, price_train, price_test = data_preprocessing(csv)
    learning(title_train, price_train, title_test, price_test)

model_learning(loc_csv)
```

-csv파일 dataframe형식

-리스트화

-훈련/검증 split

-Sklearn 모델링

-pickle파일 저장

1.개요

2.분석 프로세스

3.개발

4.서비스

5.마무리

3.개발

3.감성분석

```
def using(using_text):  
    pipe = joblib.load('data/sentiment_model/{}_pipe.pkl'.format(company))  
    text = using_text  
    # 예측 정확도  
    r1 = np.max(pipe.predict_proba(text))*100  
    # 예측 결과  
    r2 = pipe.predict(text)[0]  
    if r2 == 1:  
        senti_result = '주가는 상승할 것으로 예상됩니다'  
    else:  
        senti_result = '주가는 하락할 것으로 예상됩니다'  
    senti_accuracy = '정확도 : ' + str(round(r1, 2)) + '%'  
    return senti_result, senti_accuracy
```

```
senti_result, senti_accuracy = using(using_text)
```

```
return cloud_data, senti_result, senti_accuracy
```

-load

-예측

감성분석 결과

주가는 하락할 것으로 예상됩니다

정확도 : 90.48%

4.서비스
1.Framework

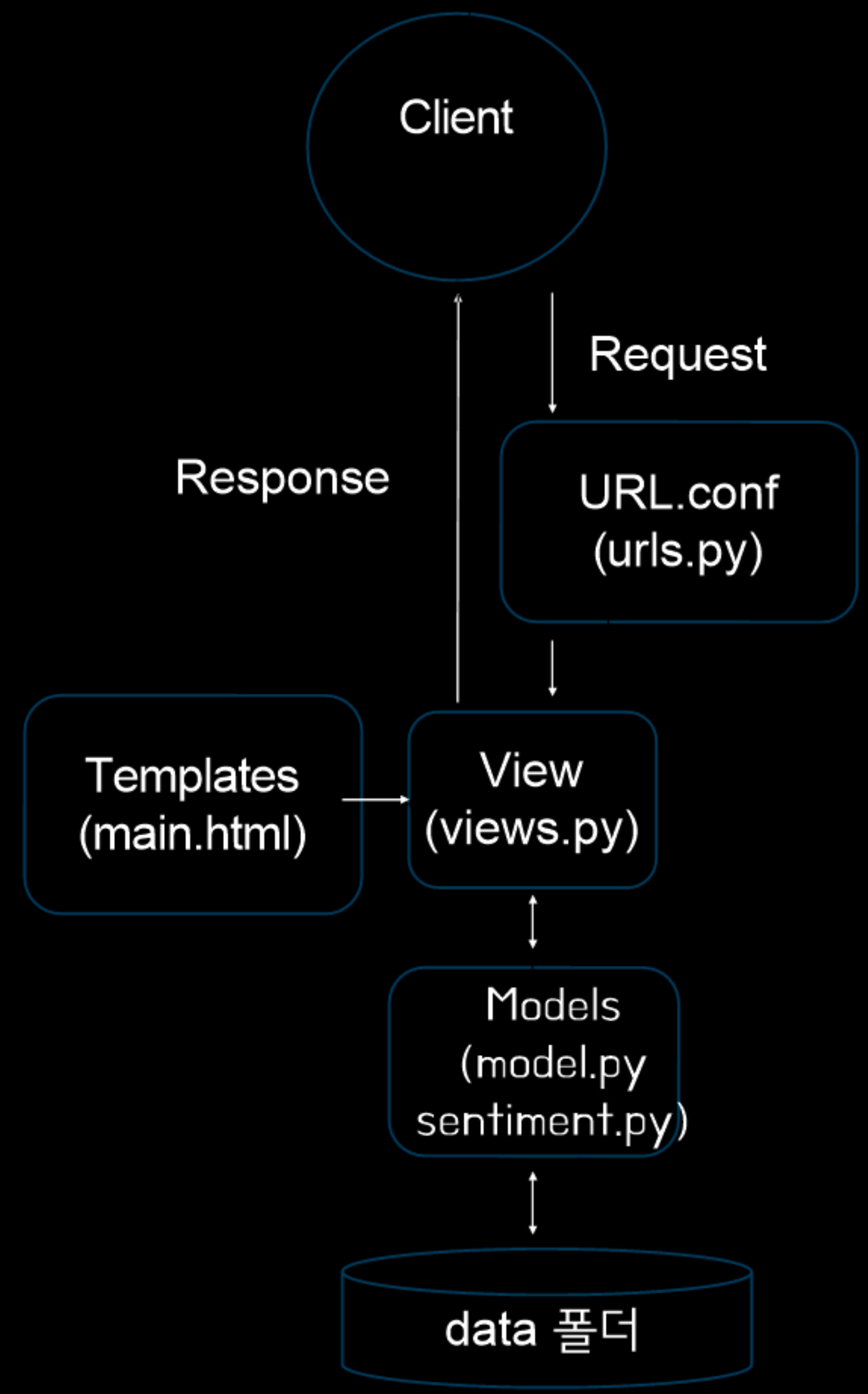
1.개요

2.분석 프로세스

3.개발

4.서비스

5.마무리



4.서비스

1.Framework

1.개요

2.분석 프로세스

3.개발

4.서비스

5.마무리

```
views.py x
main > views.py > main
1 from django.shortcuts import render
2 from finance_doctor.settings import BASE_DIR
3 import os
4 import json
5 from main.predict_model.model import PredictModel
6 from main.predict_model.sentiment import SentimentModel
7
8 def main(request):
9     theme = request.GET.get('theme')
10    company = request.GET.get('company')
11    model_string = request.GET.get('model')
12    predict_term = request.GET.get('predictTerm')
13    request_status = request.GET.get('requestStatus')
```

```
views.py x
main > views.py > ...
15 if request_status == 'true':
16     model = PredictModel()
17     accuracy, cost = model.start_predict(theme, company, model_string, predict_term)
18
19     c = os.path.join(BASE_DIR, 'chart.json')
20     with open(c) as f:
21         data = json.load(f)
22     json3 = data
23
24     sentiment = SentimentModel()
25     cloud_data, senti_result, senti_accuracy = sentiment.start_sentiment(company)
26
27     context = {
28         'company_name': company,
29         'chart': json3,
30         'testState': '1',
31         'accuracy': accuracy,
32         'cost' : cost,
33         'wordcloud': cloud_data,
34         'sentiment': {
35             'result': senti_result,
36             'accuracy': senti_accuracy
37         }
38     }
39
40     return render(
41         request,
42         'main.html',
43         context
44     )
45 else:
46     return render(request, 'main.html')
```

4.서비스

1.Framework

1.개요

2.분석 프로세스

3.개발

4.서비스

5.마무리

model.py 2 ×

main > predict_model > model.py > PredictModel > start_predict

```
17 class PredictModel:
18     def __init__(self):
19         pass
20
21     def start_predict(self, theme, company, modelString, predict_term):
22         code = ''
23         from_date = '2020-01-20'
24
25         # 여기서 테마, 회사 를 조건으로 종목코드 넣기
26         if theme == 'game':
27             if company == '엔씨소프트':
28                 code = '036570'
29             if company == '넷마블':
30                 code = '251270'
31             if company == '펄어비스':
32                 code = '263750'
33         if theme == 'broadCast':
34             if company == 'CJ ENM':
35                 code = '035760'
36             if company == '스튜디오드래곤':
37                 code = '253450'
38             if company == 'SBS':
39                 code = '034120'
```


4.서비스

1.Framework

1.개요

2.분석 프로세스

3.개발

4.서비스

5.마무리

model.py 2 x

main > predict_model > model.py > PredictModel > start_predict

```
73 def lstm(self, df, predict_term, company):
74     stock = df.copy()
#전처리 75     stock['Volume'] = stock['Volume'].replace(0, np.nan)
76     stock = stock.dropna()
77
78     scaler = MinMaxScaler()
79     scaled_stock = scaler.fit_transform(stock['Close'].values.reshape(-1,1))
#정규화 80
81     close_data = scaled_stock
82
83     split_percent = 0.70
84     split = int(split_percent * len(close_data))
85
#split 86     close_train = close_data[:split]
87     close_test = close_data[split:]
88
89     date_train = stock.index[:split]
90     date_test = stock.index[split:]
91
92     look_back = 5
93
94     train_generator = TimeseriesGenerator(close_train, close_train, length=look_back, batch_size=32)
95     test_generator = TimeseriesGenerator(close_test, close_test, length=look_back, batch_size=1)
```


4.서비스

1.Framework

1.개요

2.분석 프로세스

3.개발

4.서비스

5.마무리

model.py 2 x

main > predict_model > model.py > PredictModel > lstm

```
114 ##### save #####
115 h5 = 'data/lstm_model/LSTM_{}.h5'.format(company)
116 checkpoint = ModelCheckpoint(h5,
117                               monitor='loss',
118                               verbose=1,
119                               save_best_only=True,
120                               save_weights_only=True,
121                               mode='auto')
#save
122 early_stop = EarlyStopping(monitor='loss',
123                             patience=20)
124 json = 'data/lstm_model/LSTM_{}.json'.format(company)
125 model_json = model.to_json()
126 with open(json, 'w') as json_file:
127     json_file.write(model_json)
128
129 num_epochs = 100
130 model.fit(train_generator, validation_data=test_generator, epochs=num_epochs, verbose=1,
131           callbacks=[checkpoint, early_stop])
132 ##### load #####
133 def load_lstm_model(json, h5):
134     json_file = open(json, 'r')
135     loaded_model_json = json_file.read()
136     json_file.close()
137     model = model_from_json(loaded_model_json)
138     model.load_weights(h5)
139     return model
#load
140 model = load_lstm_model('data/lstm_model/LSTM_{}.json'.format(company),
141                        'data/lstm_model/LSTM_{}.h5'.format(company))
142 #####
```

▼ crawling_data

- 넷마블_뉴스타이틀.csv
- 넷마블_주가데이터.csv
- 롯데리츠_뉴스타이틀.csv
- 롯데리츠_주가데이터.csv
- 삼성바이오로직스_뉴스...
- 삼성바이오로직스_주가...
- 셀트리온_뉴스타이틀.csv
- 셀트리온_주가데이터.csv
- 스튜디오드래곤_뉴스타...
- 스튜디오드래곤_주가데...
- 신한알파리츠_뉴스타이...
- 신한알파리츠_주가데이...
- 엔씨소프트_뉴스타이틀...
- 엔씨소프트_주가데이터...
- 와이엔텍_뉴스타이틀.csv
- 와이엔텍_주가데이터.csv
- 풀어비스_뉴스타이틀.csv
- 풀어비스_주가데이터.csv
- 한미약품_뉴스타이틀.csv
- 한미약품_주가데이터.csv
- CJ ENM_뉴스타이틀.csv
- CJ ENM_주가데이터.csv
- HMM_뉴스타이틀.csv
- HMM_주가데이터.csv
- KSS해운_뉴스타이틀.csv
- KSS해운_주가데이터.csv
- SBS_뉴스타이틀.csv
- SBS_주가데이터.csv
- SK디앤디_뉴스타이틀.csv
- SK디앤디_주가데이터.csv

▼ lstm_model

- LSTM_넷마블.h5
- { } LSTM_넷마블.json
- LSTM_롯데리츠.h5
- { } LSTM_롯데리츠.json
- LSTM_삼성바이오로직...
- { } LSTM_삼성바이오로직...
- LSTM_셀트리온.h5
- { } LSTM_셀트리온.json
- LSTM_스튜디오드래곤.h5
- { } LSTM_스튜디오드래곤.js...
- LSTM_신한알파리츠.h5
- { } LSTM_신한알파리츠.json
- LSTM_엔씨소프트.h5
- { } LSTM_엔씨소프트.json
- LSTM_와이엔텍.h5
- { } LSTM_와이엔텍.json
- LSTM_풀어비스.h5
- { } LSTM_풀어비스.json
- LSTM_한미약품.h5
- { } LSTM_한미약품.json
- LSTM_CJ ENM.h5
- { } LSTM_CJ ENM.json
- LSTM_HMM.h5
- { } LSTM_HMM.json
- LSTM_KSS해운.h5
- { } LSTM_KSS해운.json
- LSTM_SBS.h5
- { } LSTM_SBS.json
- LSTM_SK디앤디.h5
- { } LSTM_SK디앤디.json

▼ sentiment_model

- 넷마블_pipe.pkl
- 롯데리츠_pipe.pkl
- 삼성바이오로직스_pipe....
- 셀트리온_pipe.pkl
- 스튜디오드래곤_pipe.pkl
- 신한알파리츠_pipe.pkl
- 엔씨소프트_pipe.pkl
- 와이엔텍_pipe.pkl
- 풀어비스_pipe.pkl
- 한미약품_pipe.pkl
- CJ ENM_pipe.pkl
- HMM_pipe.pkl
- KSS해운_pipe.pkl
- SBS_pipe.pkl
- SK디앤디_pipe.pkl

4.서비스

1.Framework

1.개요

2.분석 프로세스

3.개발

4.서비스

5.마무리

model.py ×

main > predict_model > model.py > ...

```
150     # 평균절대값백분율오차계산 (MAPE)
151     mape = mean_absolute_percentage_error(prediction, close_test[look_back:])
152     accuracy = (1 - mape) * 100
153     accur_print = '정확도: ' + str(round(accuracy, 2)) + '%'
```


1.개요

2.분석 프로세스

3.개발

4.서비스

5.마무리

4.서비스

1.Framework

```
model.py 2 x
main > predict_model > model.py > PredictModel > lstm

156 def predict(num_prediction, model):
157     prediction_list = close_data[-look_back:]
158
159     for _ in range(num_prediction):
160         x = prediction_list[-look_back:]
161         x = x.reshape((1, look_back, 1))
162         out = model.predict(x)[0][0]
163         prediction_list = np.append(prediction_list, out)
164     prediction_list = prediction_list[look_back - 1:]
165
166     return prediction_list
167
168 def predict_dates(num_prediction):
169     last_date = df.index.values[-1]
170     prediction_dates = pd.date_range(last_date, periods=num_prediction + 1).tolist()
171     return prediction_dates
172
173 num_prediction = 20
174
175 if predict_term == '5':
176     num_prediction = 5
177 if predict_term == '30':
178     num_prediction = 30
179
180 forecast = predict(num_prediction, model)
181 forecast_dates = predict_dates(num_prediction)
```

4.서비스

1.Framework

1.개요

2.분석 프로세스

3.개발

4.서비스

5.마무리

```
model.py 2 x
main > predict_model > model.py > PredictModel > lstm
192     trace1 = go.Scatter(
193         x=date_train,
194         y=chart_close_train,
195         mode='lines',
196         name='TrainData'
197     )
198     trace2 = go.Scatter(
199         x=stock.index[split-1:],
200         y=chart_close_data[split-1:],
201         mode='lines',
202         name='TestData'
203     )
204     trace3 = go.Scatter(
205         x=date_test[look_back:],
206         y=chart_prediction,
207         mode='lines',
208         name='Prediction'
209     )
210     trace4 = go.Scatter(
211         x=forecast_dates,
212         y=chart_forecast,
213         mode='lines',
214         name='Forecast'
215     )
216     layout = go.Layout(
217         # title="Stock Prediction",
218         margin=dict(l=20, r=20, t=10, b=50),
219         legend=dict(orientation='h', yanchor='bottom', y=1.02, xanchor='right', x=1),
220         xaxis={'title': "날짜"},
221         yaxis={'title': "종가"}
222     )
223     fig = go.Figure(data=[trace1, trace2, trace3, trace4], layout=layout)
224     fig.write_json('chart.json')
225
226     tomorrow_cost = scaler.inverse_transform(forecast.reshape(-1,1))
227     cost print = '(익일)예측가: ' + '{:.1}'.format(int(tomorrow_cost[1.0])) + '원'
```

4.서비스

1.Framework


1.개요

2.분석 프로세스

3.개발

4.서비스

5.마무리

 model.py ×

main > predict_model >  model.py >  PredictModel >  prophet

```
291 tomorrow_cost = forecast['yhat'][stock.shape[0]]
292 cost_print = '(익일)예측가: ' + '{:,}'.format(int(tomorrow_cost)) + '원'
293 return accur_print, cost_print
```


4.서비스

1.Framework



시계열분석 결과

(익일)예측가: 5,500원

정확도: 94.39%

1.개요

2.분석 프로세스

3.개발

4.서비스

5.마무리

5.마무리

1.기대효과 & 아쉬운점

기대효과 & 아쉬운점

1.개요

2.분석 프로세스

3.개발

4.서비스

5.마무리

5.마무리

2.데모 시연

데모 시연