

SW개발/HW제작 설계서

프로젝트 명 : 빅데이터 기반 주식 및 뉴스 분석 시스템 구축

2021. 08. 28

| 시장/기술 동향 분석

2030 토스 사용자 "주식투자 지속 혹은 확대할 것"

2030 토스 사용자 1093명 대상 설문조사 결과

현재 주식투자를 하고 계신가요?

1,093명이 이 질문에 답변했습니다



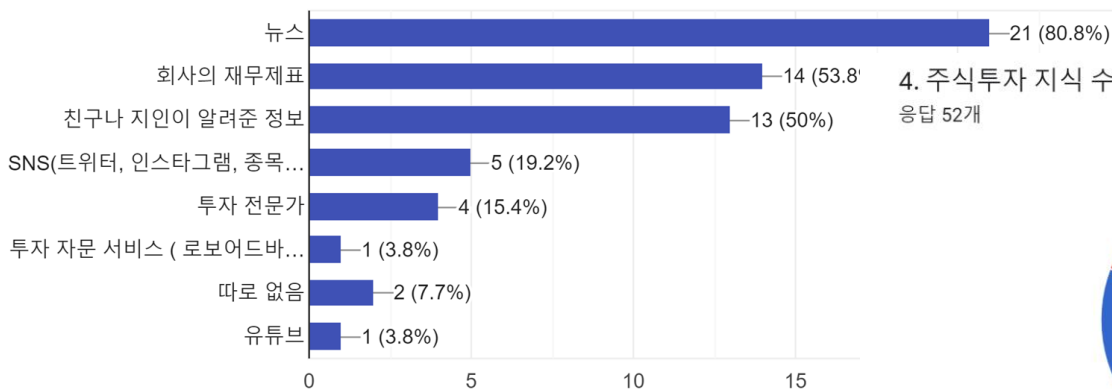
2030 세대 총 1093명을 설문 조사한 결과 970명이 주식 투자를 하고 있거나 할 계획이 있다는 생각을 가지고 있습니다. 최근 주식에 대한 엄청난 수요로 '동학개미' '서학개미'라는 신조어가 생길뿐만 아니라 대학생, 직장인들 사이에서 주식공부는 필수가 되어가고 있습니다.

이러한 상황 속에서 빅데이터를 이용해 더 빠르고, 더 정확한 주식 정보에 대한 수요 또한 높아지고 있습니다. 박재민 토스 증권 대표는 "밀레니얼 투자자들은 모바일로 모든 것을 해결하고 싶어하며 금융생활에 있어서도 간편하게 서비스를 이용하기를 원하는 성격이 강하다" 고 말했습니다.

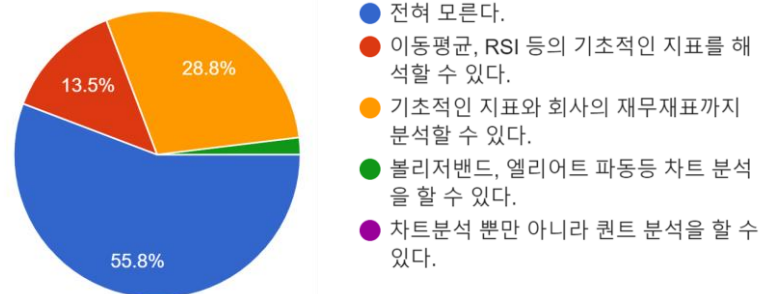
| 설문조사 분석

설문 이름	빅데이터 기반 주식 및 뉴스 분석에 대한 설문 조사
설문 시행 기간	2021년 8월 26일 ~ 2021년 8월 29일
설문 응답 인원	52명
표본 집단	10대~40대이상에 이르기까지 전 연령
표본 집단 선정 이유	주식에 대한 다양한 연령층의 인식과 투자 성향을 파악하고, 다양한 의견을 주식 및 뉴스 분석 시스템 개발에 반영하기 위함
설문 도구	Google 설문폼

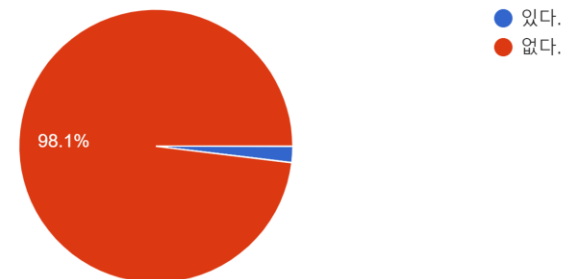
2-1. 주식 투자 시 참고하는 것 응답 26개



4. 주식투자 지식 수준 응답 52개



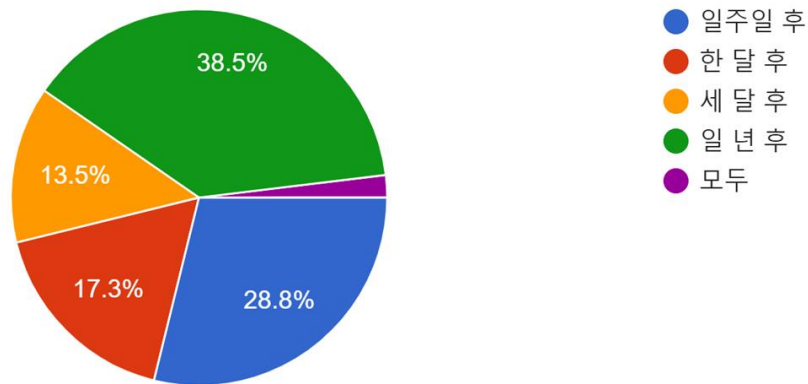
6. 로보어드바이저나 주가 예측 서비스를 사용해 본 경험이 있다. 응답 52개



| 설문조사 분석

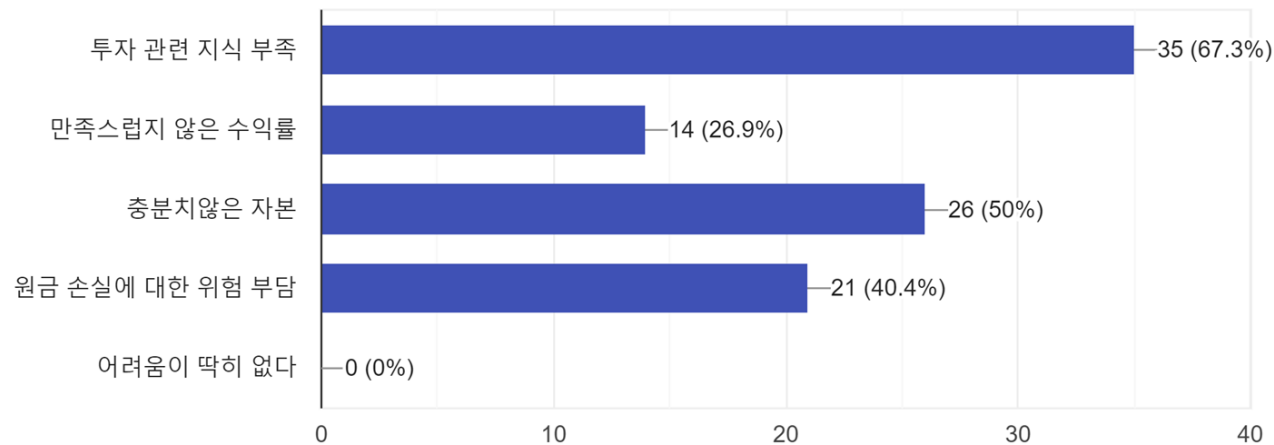
3. 특정 종목을 매수할 때, N일 후 주가가 오를지, 내릴지 알 수 있다면 가장 궁금한 것은?

응답 52개



5. 투자에 있어서 어려움

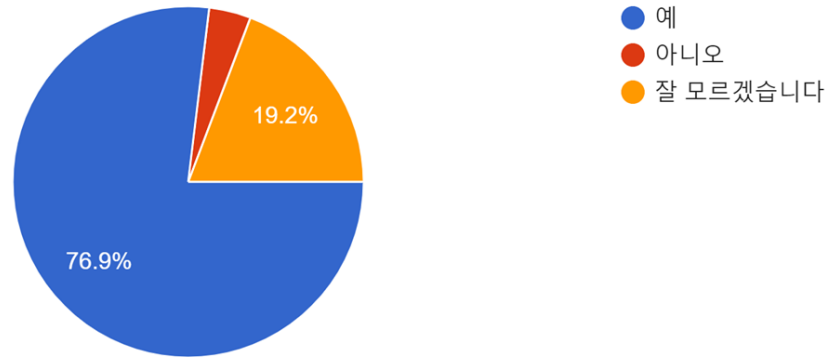
응답 52개



| 설문조사 분석

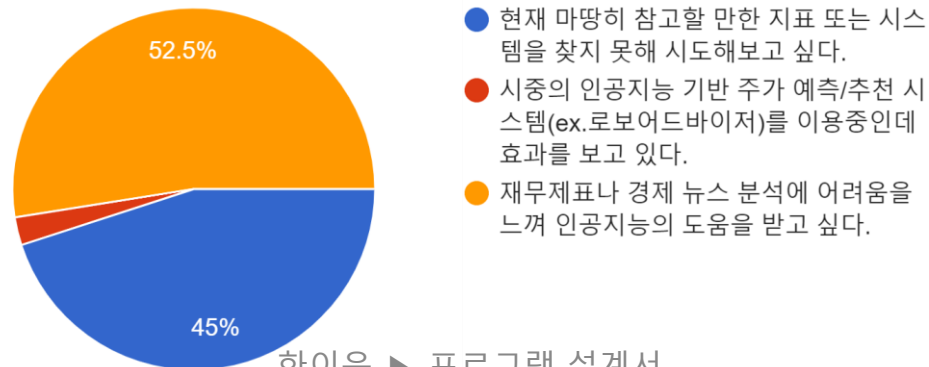
7. 빅데이터(주가, 시장지표, 거래량, 뉴스 등) 기반 뉴스 분석 및 주가 예측 시스템이 있다면 사용할 의향이 있습니까?

응답 52개



7번에 '예'라고 답변한 이유를 선택해주세요.

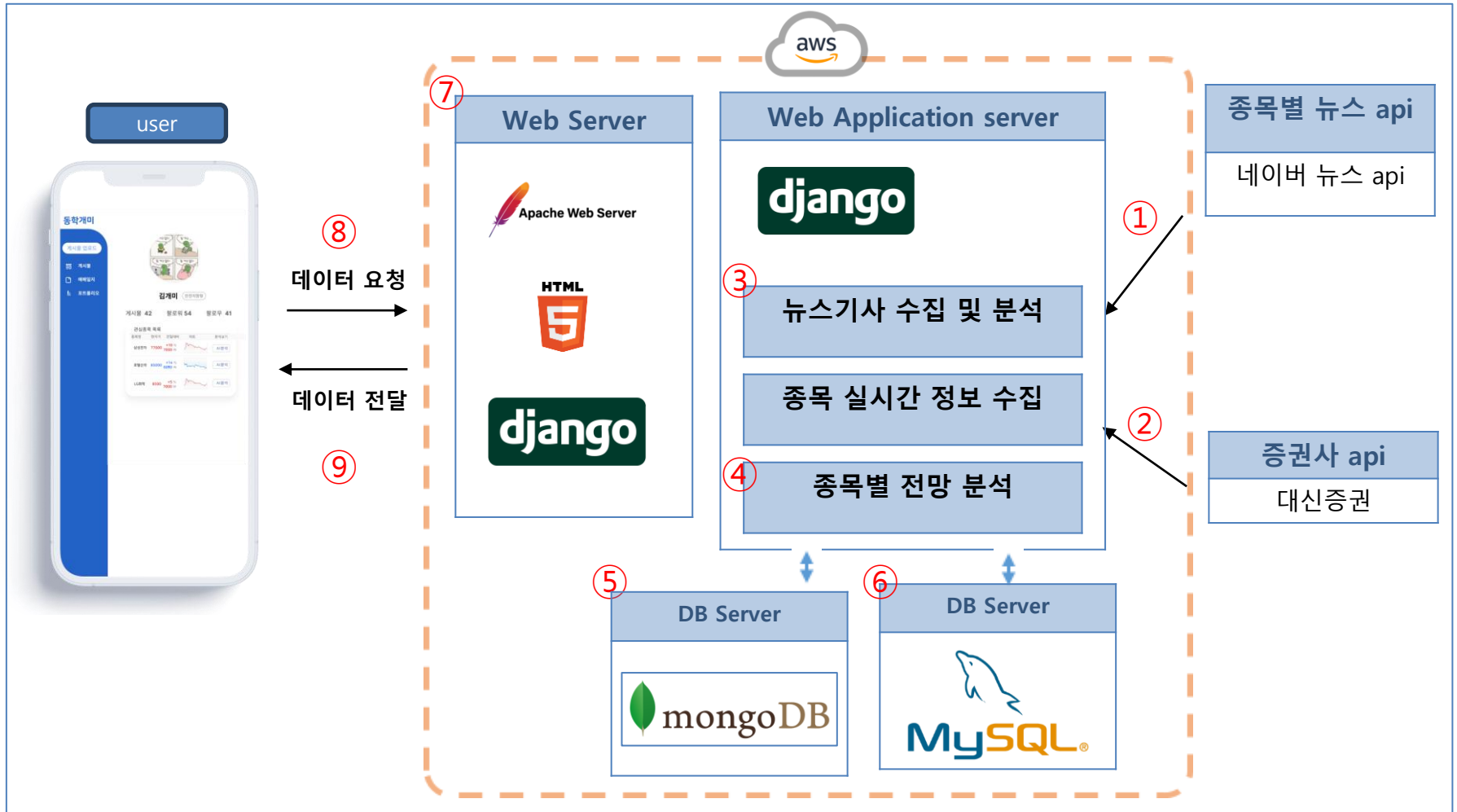
응답 40개



| 요구사항 정의서

ID		구분	서비스(메뉴)	기능	상세기능	개발형태	담당자	특이사항
1	1	종목분석	감성분석	뉴스 크롤링	실시간으로 뉴스를 크롤링해 최신뉴스를 상위에 보여준다.		김지현, 조설아	
	2			감성분석	종목과 관련있는 뉴스와 그에 대한 긍/부정의견을 보여준다.			
	3			뉴스링크 이동	감성분석 한 뉴스를 클릭하면 해당 뉴스 링크로 이동한다.			
	4		AI분석	종목 목록	종목명, 종목코드, 가격, 등락률, 상승/하락 예측, 예측 확률을 보여준다.		정수민, 이연수	
	5			AI분석	머신러닝을 활용해 종목별 가격전망을 예측해 상승 하락 여부와 그 확률을 보여준다.			
	6			실시간 주식 정보	일봉 차트, 일별 가격 정보를 보여준다.			
2	1	관리자	관리자 기능	이용자 통계	최근 1달간 접속자 수/가입자 수 현황			

| 서비스 구성도 - 서비스 시나리오



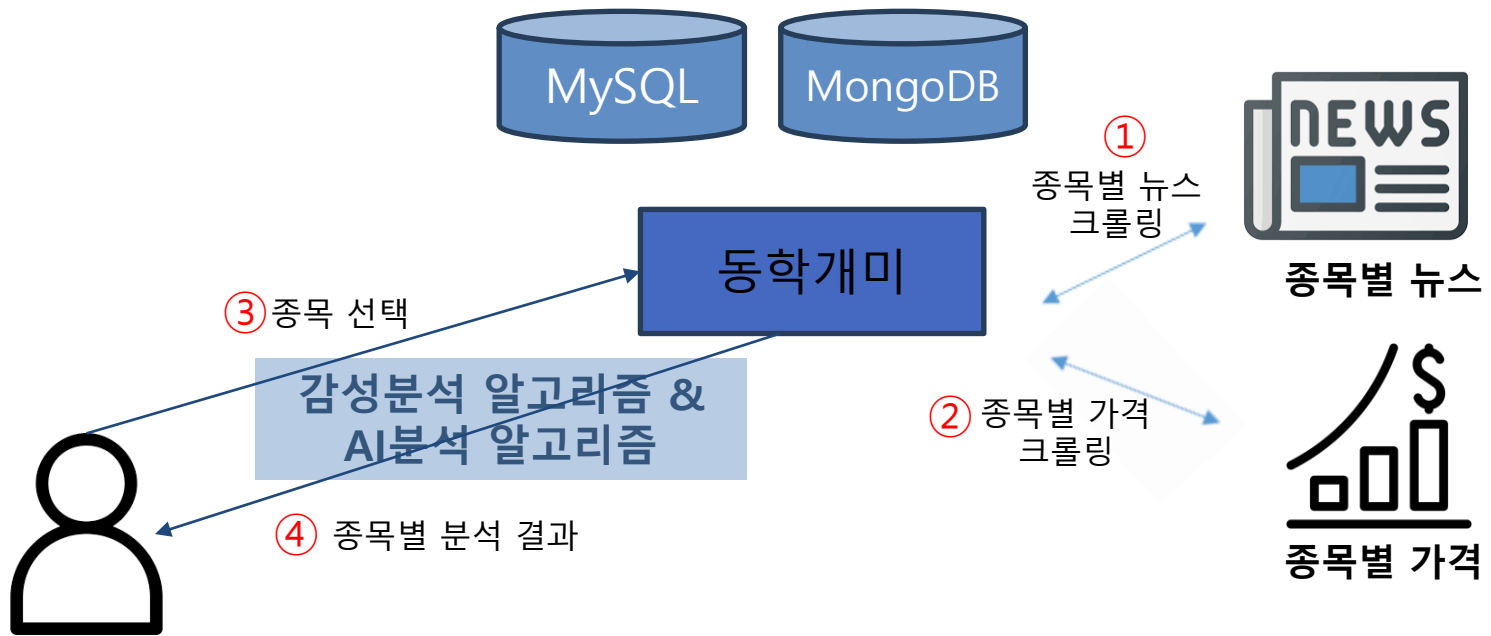
back-end

1. **종목별 뉴스 기사 수집:** 종목별 뉴스를 네이버 뉴스 api를 이용한 웹크롤링
2. **종목 실시간 정보 수집:** 각 종목에 대한 실시간 정보를 대신증권 api를 이용해 크롤링
3. **뉴스기사 분석:** 수집한 뉴스 텍스트에 대하여 자연어 처리를 한 후 감성분석을 하여 사용자에게 결과를 제공
4. **종목별 전망 분석:** 수집한 주가 정보에 대하여 feature engineering을 한 후 예측 알고리즘을 적용해 사용자에게 분석 결과를 제공
5. **MongoDB:** 수집한 뉴스 기사를 저장
6. **MySQL:** 주가 정보, 뉴스 정보, 종목별 전망 분석 결과 저장

front-end

- ⑦ **Web Server:** 반응형 웹을 통하여 사용자가 요청한 데이터를 시각화하여 보여줌
- ⑧ **사용자 데이터 요청:** 사용자가 원하는 분석을 선택하여 해당 분석 결과 요청
- ⑨ **데이터 전달:** 사용자가 요청한 분석 결과 전달

| 서비스 흐름도

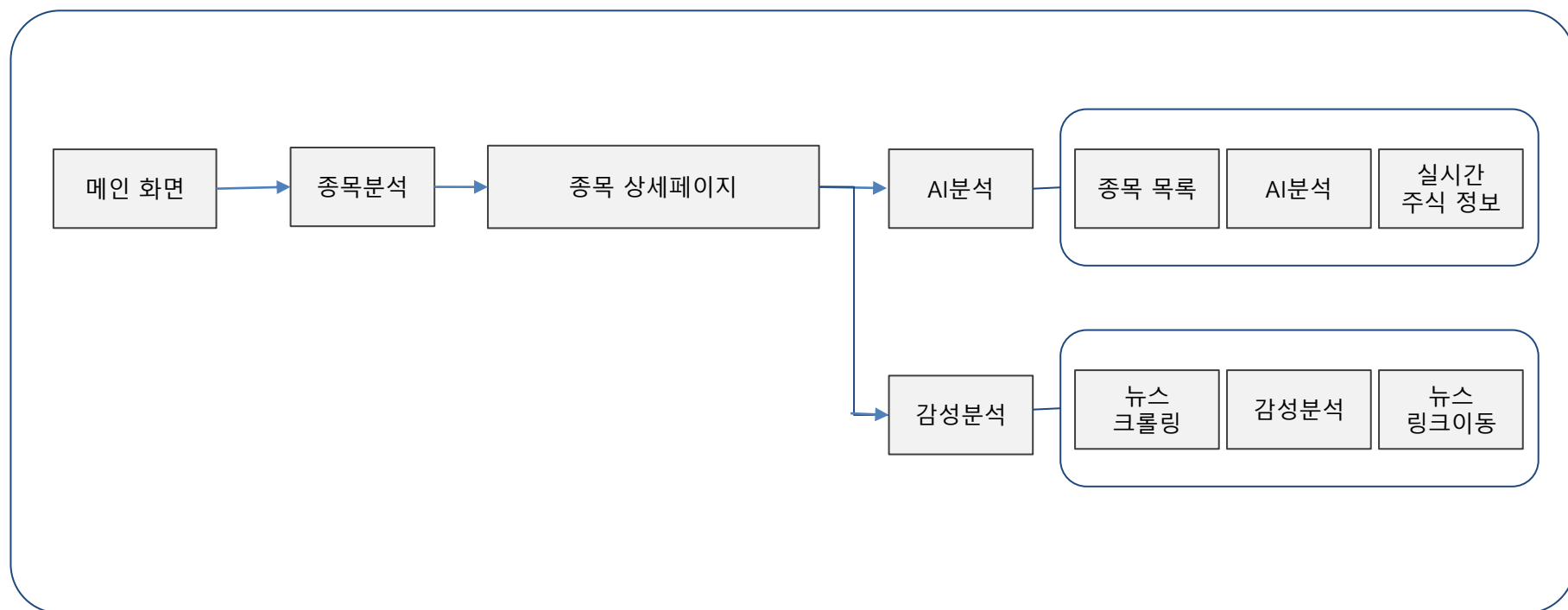


| 서비스 흐름도

데이터 흐름도 설명

1. 네이버 뉴스 api를 이용해 종목별 뉴스를 크롤링함
2. 증권사 api를 이용해 종목별 가격을 크롤링함
3. 사용자가 원하는 종목을 선택하여 서버에 전송.
4. 사용자가 선택한 종목에 대하여 감성분석과 AI분석결과를 제공함


| 메뉴 구성도



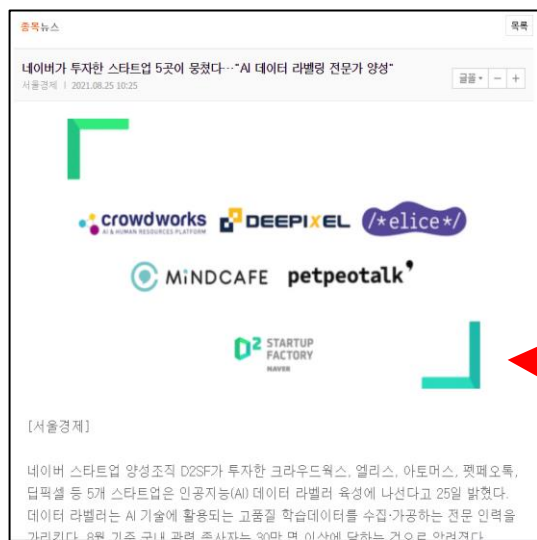
| 화면 설계서

<div><div>동학개미</div><div><div>종목분석</div><div>매매일지</div><div>포트폴리오</div><div>관심종목</div></div></div> <div><div>AI분석</div><div><table><thead><tr><th>종목명</th><th>종목코드</th><th>가격</th><th>등락률</th><th>상승/하락 예측</th><th>확률</th></tr></thead><tbody><tr><td>삼성전자</td><td>00539</td><td>80,000</td><td>+0.5%</td><td>상승</td><td>95%</td></tr><tr><td>LG</td><td>00246</td><td>80,000</td><td>+0.5%</td><td>하락</td><td>65%</td></tr><tr><td>LG</td><td>00246</td><td>80,000</td><td>+0.5%</td><td>상승</td><td>95%</td></tr><tr><td>LG</td><td>00246</td><td>80,000</td><td>+0.5%</td><td>유지</td><td>95%</td></tr><tr><td>LG</td><td>00246</td><td>80,000</td><td>+0.5%</td><td>유지</td><td>95%</td></tr><tr><td>LG</td><td>00246</td><td>80,000</td><td>+0.5%</td><td>하락</td><td>95%</td></tr><tr><td>LG</td><td>00246</td><td>80,000</td><td>+0.5%</td><td>하락</td><td>95%</td></tr><tr><td>LG</td><td>00246</td><td>80,000</td><td>+0.5%</td><td>하락</td><td>95%</td></tr><tr><td>LG</td><td>00246</td><td>80,000</td><td>+0.5%</td><td>하락</td><td>95%</td></tr><tr><td>LG</td><td>00246</td><td>80,000</td><td>+0.5%</td><td>하락</td><td>95%</td></tr></tbody></table></div></div>	종목명	종목코드	가격	등락률	상승/하락 예측	확률	삼성전자	00539	80,000	+0.5%	상승	95%	LG	00246	80,000	+0.5%	하락	65%	LG	00246	80,000	+0.5%	상승	95%	LG	00246	80,000	+0.5%	유지	95%	LG	00246	80,000	+0.5%	유지	95%	LG	00246	80,000	+0.5%	하락	95%	LG	00246	80,000	+0.5%	하락	95%	LG	00246	80,000	+0.5%	하락	95%	LG	00246	80,000	+0.5%	하락	95%	LG	00246	80,000	+0.5%	하락	95%	기능 번호	STOCK-01-01
종목명	종목코드	가격	등락률	상승/하락 예측	확률																																																															
삼성전자	00539	80,000	+0.5%	상승	95%																																																															
LG	00246	80,000	+0.5%	하락	65%																																																															
LG	00246	80,000	+0.5%	상승	95%																																																															
LG	00246	80,000	+0.5%	유지	95%																																																															
LG	00246	80,000	+0.5%	유지	95%																																																															
LG	00246	80,000	+0.5%	하락	95%																																																															
LG	00246	80,000	+0.5%	하락	95%																																																															
LG	00246	80,000	+0.5%	하락	95%																																																															
LG	00246	80,000	+0.5%	하락	95%																																																															
LG	00246	80,000	+0.5%	하락	95%																																																															
	기능 명	AI분석																																																																		
	기능 설명	종목명, 종목코드, 가격, 등락률,상승/하락 예측, 예측 확률을 보여준다.																																																																		
	처리 내용	<div>■ 상승/하락 예측</div> <p>머신러닝을 활용해 예측한 가격 전망을 상승/하락/유지로 보여준다.</p> <div>■ 상승/하락 예측 확률</div> <p>상승/하락/유지 예측 확률을 보여준다.</p> <div>■ 종목별 AI분석 상세 페이지</div> <p>일별차트, 일별 가격 정보를 보여준다.</p>																																																																		
<div><div>동학개미</div><div><div>종목분석</div><div>매매일지</div><div>포트폴리오</div><div>관심종목</div></div></div> <div><div>삼성전자 00539</div><div>80,000 ▲ 800 +0.5%</div><div><div>투자정보</div><div><div>상한가 100,000</div><div>하한가 60,000</div><div>52주 최고 90,000</div><div>52주 최저 60,000</div><div>일봉</div><div>월봉</div><div>분기별</div><div>연도별</div></div><div><div>외국인자본 13.35%(104%)</div><div>시가총액 4,796,705 (11%)</div><div>EPSPER 384(200) (30%)</div><div>EPSPER 384(200) (30%)</div><div>WCS 연도별과 연... 100%</div></div></div></div> <div><div>분석결과</div><div><div>감성분석</div><div>AI분석</div></div><div><div>10일 후</div><div>상승확률</div></div><div><div>상승</div><div>95%</div></div></div>	비고																																																																			
	요구사항 명	AI분석 - 종목목록																																																																		

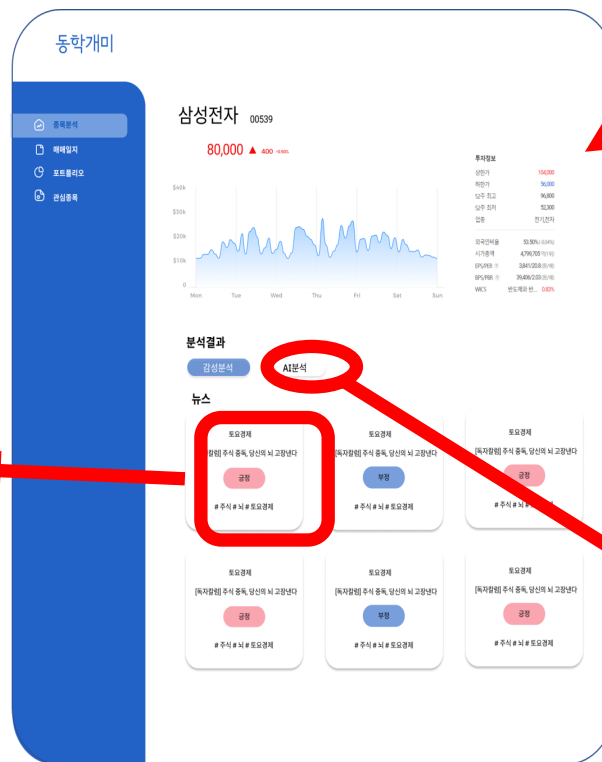
| 화면 설계서

	기능 번호	STOCK-01-02
기능 명	감성분석	
기능 설명	종목과 관련있는 뉴스에 대해 긍정/중립/부정 의견을 보여준다.	
처리 내용	<p>■ 감성분석</p> <p>종목에 대한 각 뉴스를 감성분석하여 종목을 어떻게 평가했는지 긍정/중립/부정 의견을 보여준다.</p> <p>■ 가격정보</p> <p>일별차트, 일별 가격 정보를 보여준다.</p>	
비고		
요구사항 명	종목분석 - 감성분석	

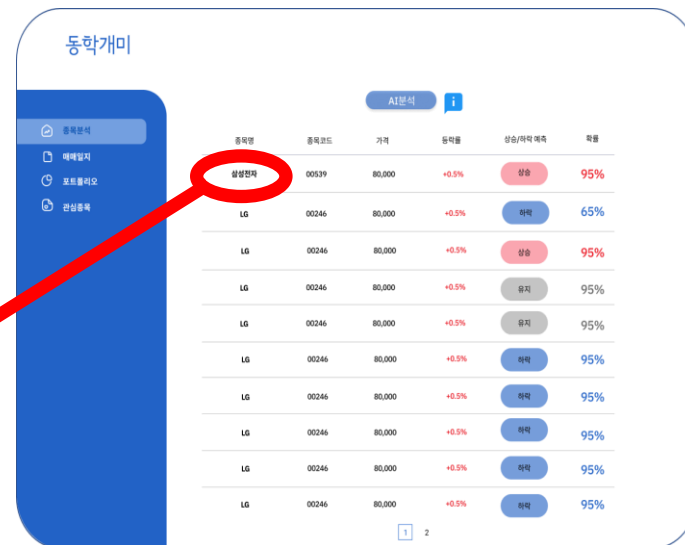
| 화면 설계서 - 사용자인터페이스(SW)



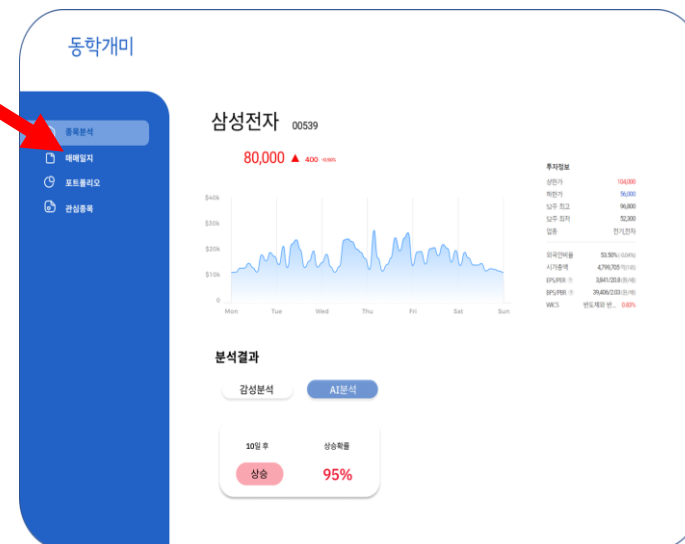
뉴스 본문으로 이동



종목 상세 페이지-감성분석

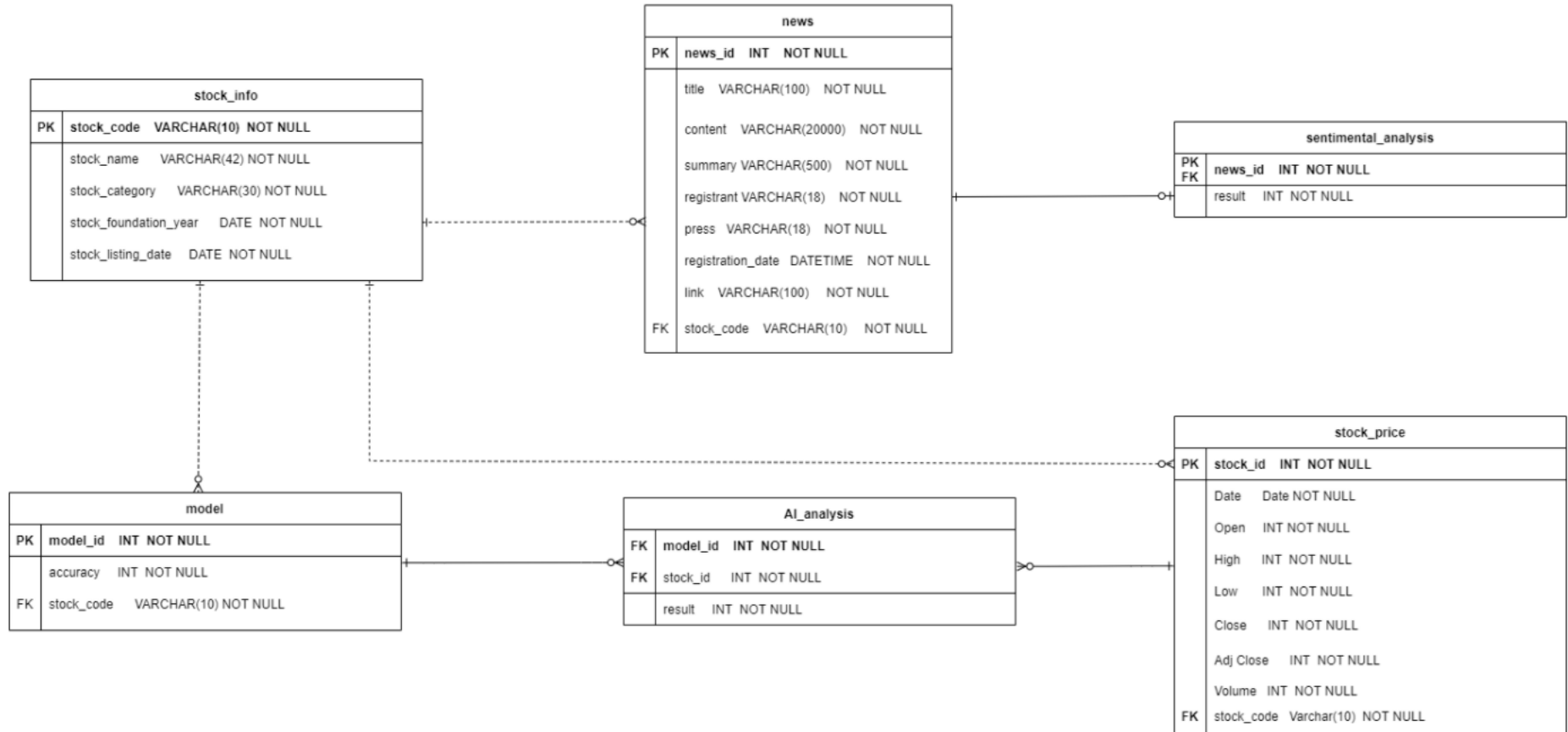


종목 목록 (메인 페이지)

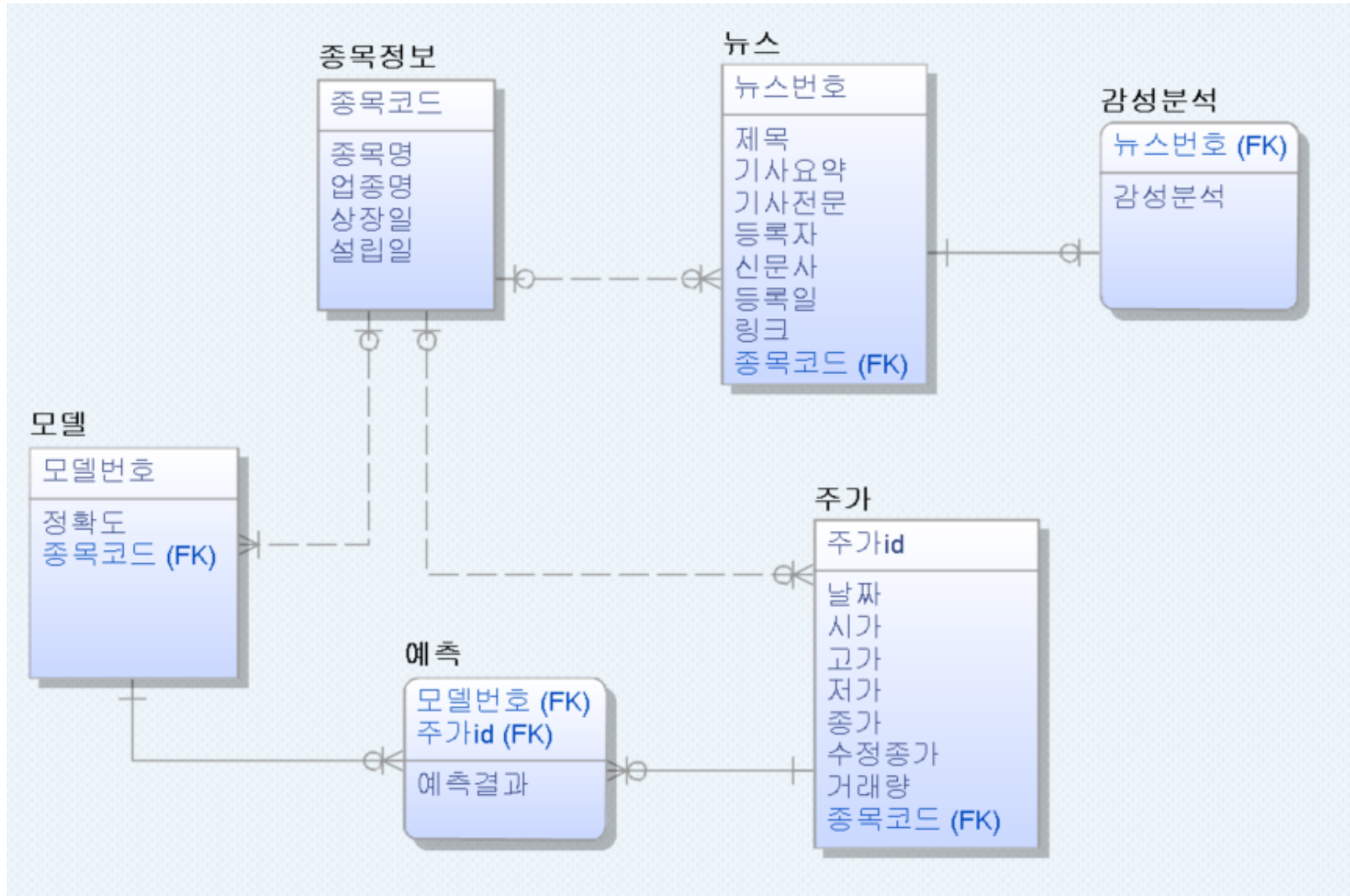


종목 상세 페이지-AI분석

| 엔티티관계도 - ERD



| 엔티티관계도 - ERD



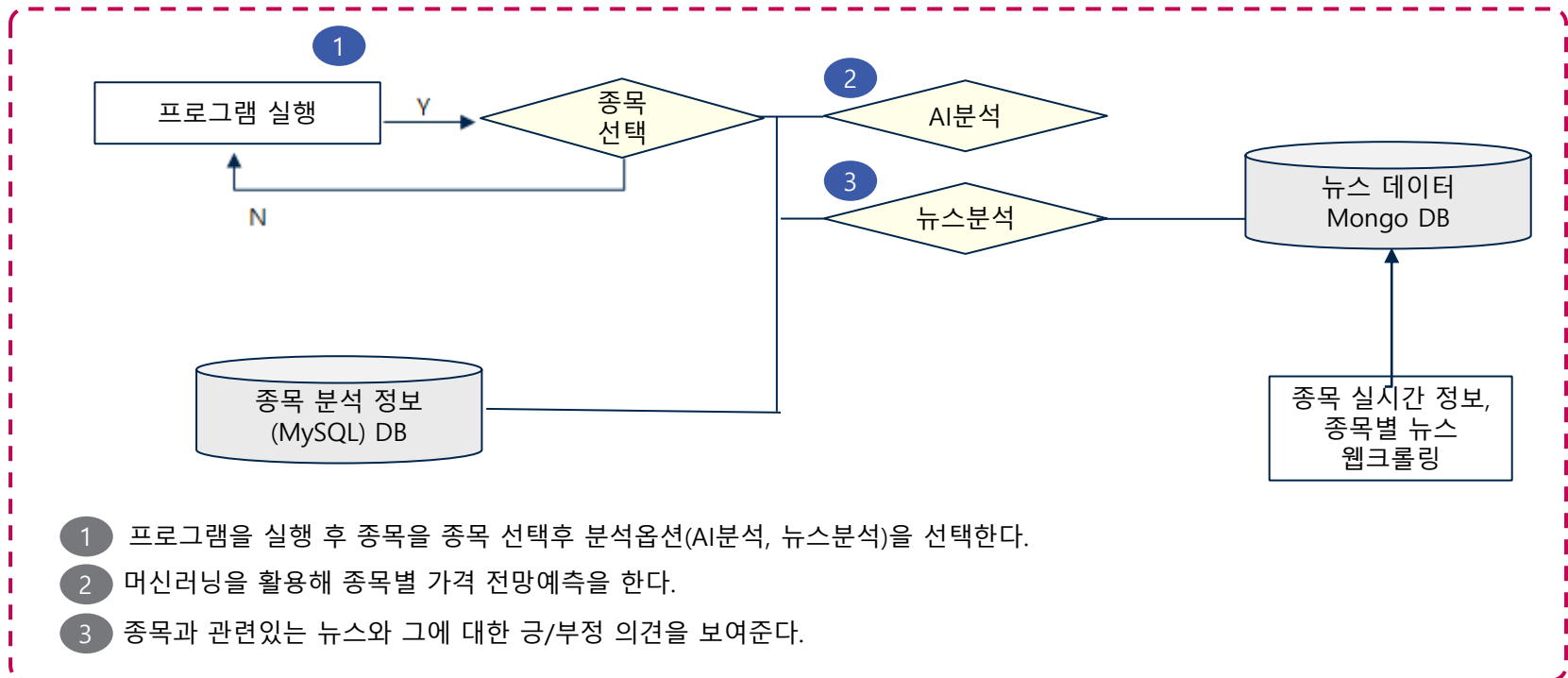
| 기능 처리도(기능 흐름도)

실무 산출물 형식

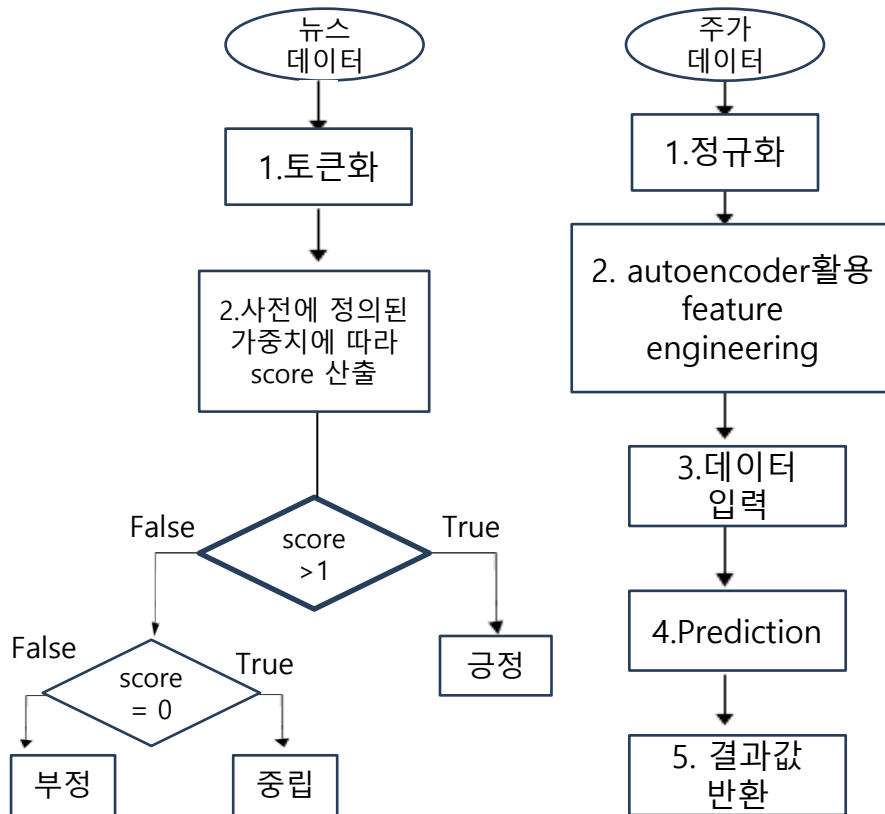
프로그램 ID	STOCK_00	프로그램 명	주식 종목 분석 프로그램	작성일	2021. 08. 26.	Page	1/1
개요	빅데이터를 AI분석, 감성분석에 활용하여 경제 및 주식 현황 정보, 종목별 주가 전망 예측을 제공하는 프로그램					작성자	동학개미

기능 흐름도

<빅데이터 기반 종목분석 연계도>



| 알고리즘 명세서



° 뉴스 감성분석 알고리즘 시나리오

1. 뉴스 데이터를 단어 토큰화한다.
2. 토큰화된 단어들에 대해 사전에 정의된 가중치따라 score값을 산출한다.
3. score 값이 1보다 크면 긍정, 0이면 부정이나 중립으로 판단한다.

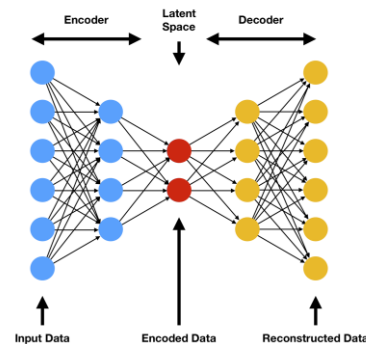
° 주가 전망 예측 알고리즘 시나리오

1. 주가 관련 데이터(시가, 종가, 고가, 저가)와 주가 지수(KOSPI)로 데이터셋을 구성하여 정규화한다.
2. autoencoder로 압축된 특성을 뽑아내 feature를 생성한다.
3. 예측할 데이터를 입력한다.
4. 학습된 모델로 예측을 진행한다.
5. 90일 후 주가의 상승,하락 여부를 결과값으로 반환한다.

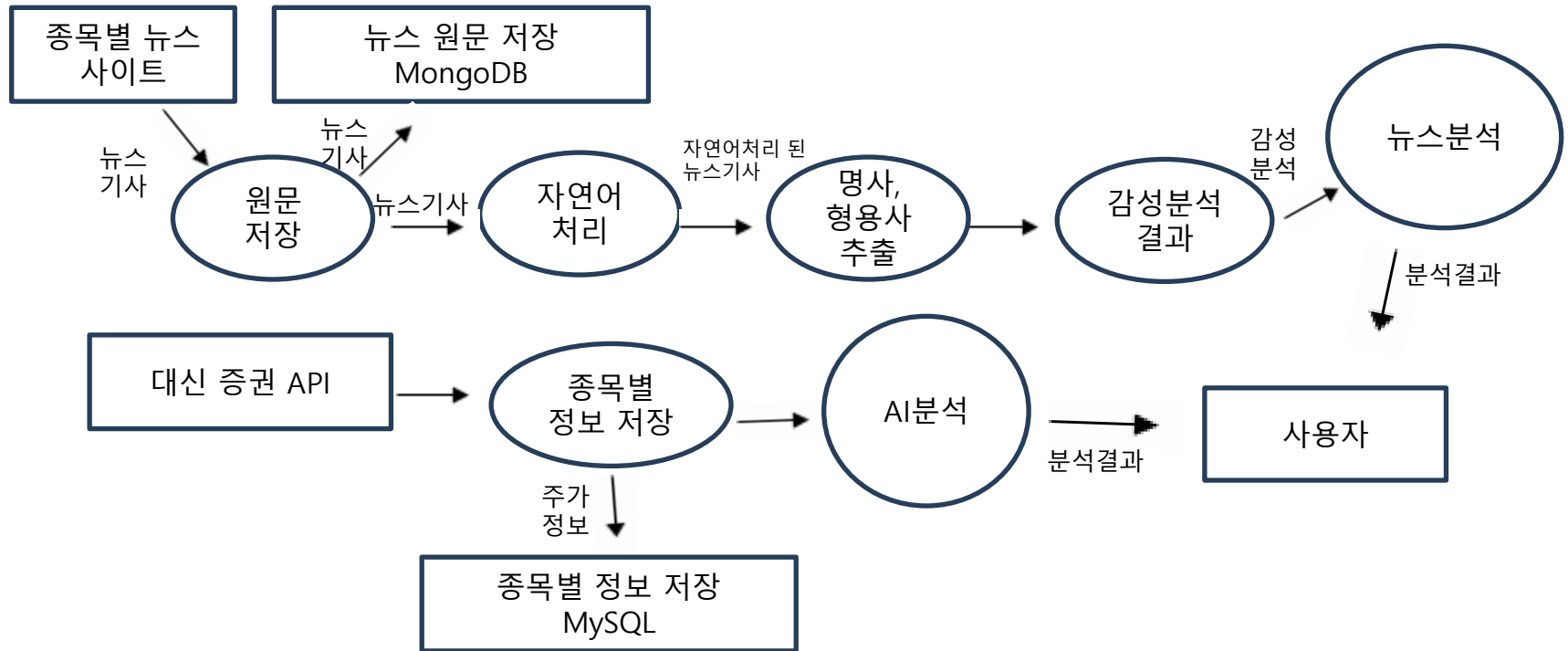
| 알고리즘 상세 설명서

◦ 주가 전망 예측 알고리즘

주가 전망 예측 알고리즘은 해당 일의 주식데이터를 앙상블 알고리즘인 RandomForest를 기반으로 학습된 모델에 입력해 전망을 예측하는 알고리즘이다. 주식 가격데이터, 기술지표 데이터, 시장 지표 데이터를 Min-Max 스케일링 한 후 autoencoder를 활용해 압축된 특성을 추출해 입력으로 사용한다. autoencoder는 입력값으로 전달받은 특성 데이터를 낮은 차원으로 변환하는 encoder와 낮은 차원에서 원본 입력값을 예측하는 decoder로 구성되어 새로운 데이터를 생성한다. 압축하는 과정에서 데이터의 특성이 추출되므로 이를 새로운 feature로 활용 할 수있다. 정규화 된 주가 데이터, autoencoder로 생성한 feature를 학습된 모델에 입력하여 예측을 진행하면 해당 종목의 90일 후 전망에 대해 이진(상승1,하락0) 결과값을 반환한다. 모델 학습에는 앙상블 학습 중 베깅 방법의 일종인 랜덤포레스트 분류기를 사용한다.



| 데이터 수집처리 정의서

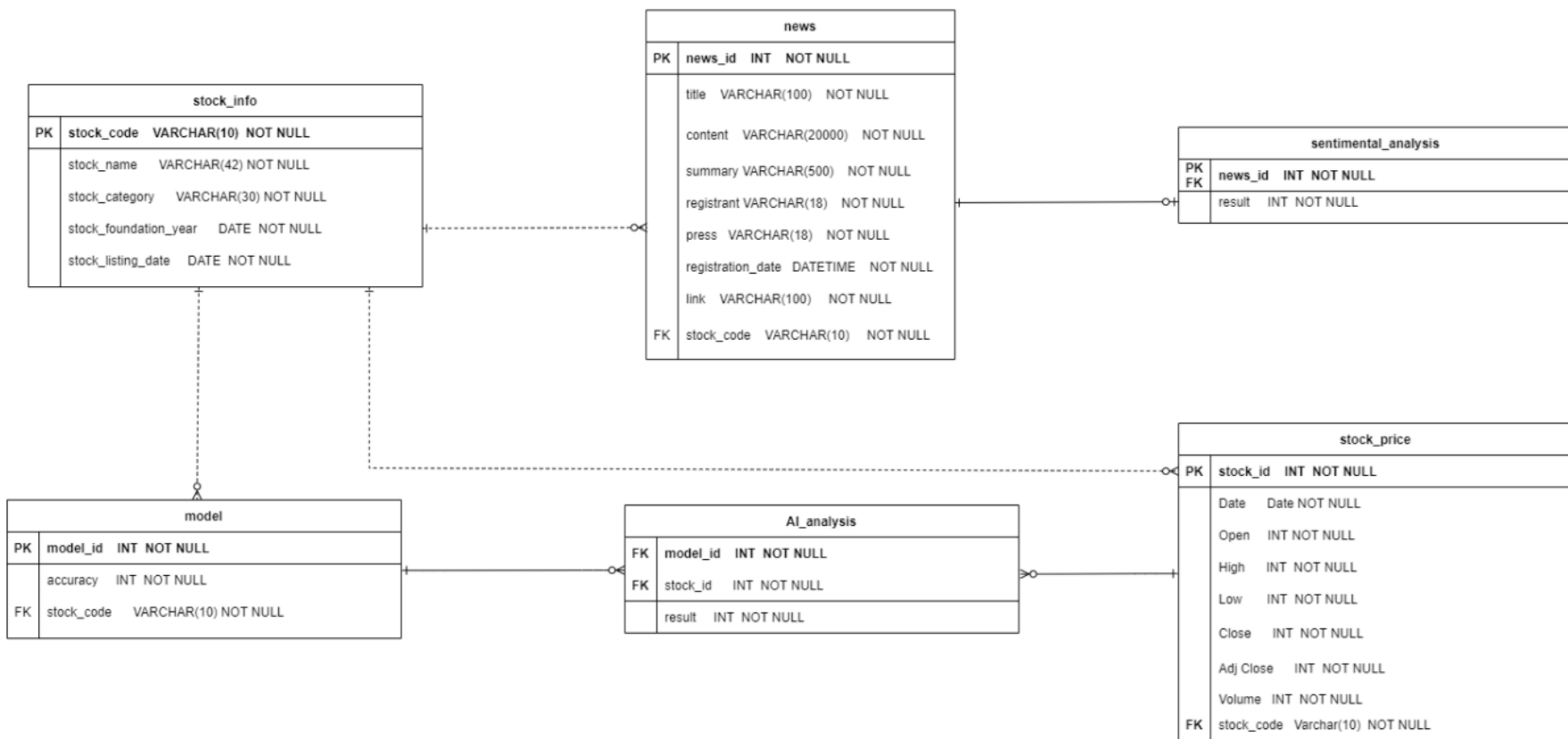


| 프로그램 - 목록

기능 분류	기능번호	기능 명	
STOCK	STOCK-01-01	AI분석	종목 목록
			AI분석
			실시간 주식 정보
	STOCK-01-02	감성분석	뉴스 크롤링
			감성분석
			뉴스 링크 이동
ADM	ADM-01-01	이용자 통계	

| 테이블 정의서 - ERD

<테이블 구성도>



| 테이블 정의서 - ERD

<주식 테이블>

항목명	TYPE	필수/선택	값 목록	설명
stock_code	VARCHAR(10)	필수		활성
stock_name	VARCHAR(42)	필수		활성
stock_category	VARCHAR(30)	필수		활성
stock_foundation_year	DATE	필수		활성
stock_listing_date	DATE	필수		활성

<예측모델 테이블>

항목명	TYPE	필수/선택	값 목록	설명
model_id	INT	필수		활성
stock_code(FK)	VARCHAR(10)	필수		활성
accuracy	INT	필수		활성

| 테이블 정의서 - ERD

<뉴스 테이블>

항목명	TYPE	필수/선택	값 목록	설명
news_id(PK)	INT	필수		활성
title	VARCHAR(100)	필수		활성
content	VARCHAR(20000)	필수		활성
summary	VARCHAR(500)	필수		활성
registrant	VARCHAR(18)	필수		활성
press	VARCHAR(18)	필수		활성
registration_date	DATE	필수		활성
link	VARCHAR(100)	필수		활성
stock_code	VARCHAR(10)	필수		활성

<감성분석 테이블>

항목명	TYPE	필수/선택	값 목록	설명
news_id(PK/FK)	INT	필수		활성
result	INT	필수		활성

| 테이블 정의서 - ERD

<주가 테이블>

항목명	TYPE	필수/선택	값 목록	설명
stock_id(PK)	INT	필수		활성
date	DATE	필수		활성
open	INT	필수		활성
high	INT	필수		활성
low	INT	필수		활성
close	INT	필수		활성
adj_close	INT	필수		활성
volume	INT	필수		활성
stock_code	VARCHAR(10)	필수		활성

<AI분석 테이블>

항목명	TYPE	필수/선택	값 목록	설명
model_id(PK/FK)	VARCHAR(10)	필수		활성
stock_id(PK/FK)	DATE	필수		활성
result	INT	필수		활성

| 핵심소스코드(1)

- autoencoder 주가 데이터 생성

```
x_train_sample = x_train.reshape((len(x_train), np.prod(x_train.shape[1:])))
x_test_sample = x_test.reshape((len(x_test), np.prod(x_test.shape[1:])))

input_window = Input(shape=(window_length,))

x = Dense(6, activation='relu')(input_window)
x = BatchNormalization()(x)
encoded = Dense(encoding_dim, activation='relu')(x)
# "decoded" is the lossy reconstruction of the input

x = Dense(6, activation='relu')(encoded)
x = BatchNormalization()(x)
decoded = Dense(window_length, activation='sigmoid')(x)

# this model maps an input to its reconstruction
autoencoder = Model(input_window, decoded)

# this model maps an input to its encoded representation
encoder = Model(input_window, encoded)

autoencoder.summary()

autoencoder.compile(optimizer='adam', loss='binary_crossentropy')
history = autoencoder.fit(x_train_sample, x_train_sample,
                          epochs=epochs,
                          batch_size=1024,
                          shuffle=True,
                          validation_data=(x_test_sample, x_test_sample))

decoded_stocks = autoencoder.predict(x_test_sample)
plot_history(history)
plot_examples(x_test_sample, decoded_stocks)
```

| 핵심소스코드(2)

- 앙상블

▼ 모델

```
[ ] logistic = LogisticRegression(solver='liblinear',penalty='l2',C= 0.1,random_state=1)
tree = DecisionTreeClassifier(max_depth=4,criterion='entropy',random_state=1)
knn = KNeighborsClassifier(n_neighbors=41,p=2,metric='minkowski')

voting_estimators = [('logistic', logistic), ('tree', tree), ('knn', knn)]

voting = VotingClassifier(estimators = voting_estimators,voting='soft')

forest = RandomForestClassifier(criterion='entropy',n_estimators=700,random_state=1)

lgb = LGBMClassifier(random_state=42)
xgb = XGBClassifier(tree_method='hist',random_state=41)

adaboost = AdaBoostClassifier(base_estimator=tree,
                             n_estimators=100,
                             learning_rate = 0.01,
                             random_state=42)

clf_labels = ['Random forest', 'Ada boost', 'XGB', 'LIGHTGBM']

all_clf = [ forest,adaboost,xgb,lgb]
```

▼ AUC 검정

```
[ ] for clf, label in zip(all_clf, clf_labels):
    scores = cross_val_score(estimator=clf,X=X_train,y=y_train,cv=10,scoring='roc_auc')
    print("ROC AUC: %0.3f (+/- %0.3f) [%s]"
          % (scores.mean(), scores.std(), label))
```

| 핵심소스코드(3)

- RNN&RandomForest 스택킹 앙상블

StratifiedKfold 기반 스택킹

```
def get_stacking_data11(model, X_train, Y_train, X_test, n_folds=5):
    stk = StratifiedKFold(n_splits=n_folds)

    #최종 모델에서 사용할 데이터 셋 세팅(0 값으로)
    #만약 shape가 (100, 10)이었으면 폴드의 검증 과정에서 저장할 데이터는 (100, 1)모양을 갖게 한다.
    train_fold_predict = np.zeros((X_train.shape[0], 1))
    #test는 X_test 값을 이용해서 매 폴드마다 예측을 하기 때문에 (100, fold개수) 만큼 shape을 갖게 된다.
    #그래서 해당 폴드마다 X_test의 예측 값을 해당 fold에 해당되는 열에 넣는다.
    test_predict = np.zeros((X_test.shape[0], n_folds))
    print('model: ', model.__class__.__name__)

    for cnt, (train_index, valid_index) in enumerate(stk.split(X_train, Y_train)):
        X_train_ = X_train[train_index]
        Y_train_ = Y_train[train_index]
        X_validation = X_train[valid_index]

        #학습
        model.fit(X_train_, Y_train_)
        #해당 폴드에서 학습된 모델에다가 검증 데이터(X_validation)로 예측 후 저장
        train_fold_predict[valid_index,:] = model.predict(X_validation).reshape(-1,1)
        #해당 폴드에서 생성된 모델에게 원본 테스트 데이터(X_test)를 이용해서 예측을 수행하고 저장
        test_predict[:,cnt] = model.predict(X_test)

    #for문이 끝나면 test_pred는 평균을 내서 하나로 합친다
    test_predict_mean = np.mean(test_predict, axis=1).reshape(-1, 1)
    return train_fold_predict, test_predict_mean
```

| 핵심소스코드(3)

- RNN&RandomForest 스택킹 앙상블

```
# 개별 ML 모델을 위한 Classifier 생성.
rf_clf = RandomForestClassifier(criterion='entropy', n_estimators=700, random_state=1) # 랜덤포레스트
dnn_clf = Sequential() # DNN
dnn_clf.add(Dense(20, input_dim=26, activation='relu'))
dnn_clf.add(Dense(20, activation='relu'))
dnn_clf.add(Dense(1, activation='sigmoid'))

# 모델을 컴파일합니다.
dnn_clf.compile(loss='binary_crossentropy',
                optimizer='adam',
                metrics=['accuracy'])

rf_train, rf_test = get_stacking_data11(rf_clf, X_train, Y_train, X_test)
dnn_train, dnn_test = get_stacking_data22(dnn_clf, X_train, Y_train, X_test)
```

```
new_X_train = np.concatenate((rf_train, dnn_train), axis=1)
new_X_test = np.concatenate((rf_test, dnn_test), axis=1)
```

```
print('원본 : ', X_train.shape, X_test.shape)
print('새로운 : ', new_X_train.shape, new_X_test.shape)
```

```
원본 : (1888, 26) (473, 26)
새로운 : (1888, 2) (473, 2)
```

StratifiedKfold 기반 스택킹 : 결과

```
# 최종 분류기 모델 생성
lr_final = LogisticRegression(C=10)
lr_final.fit(new_X_train, Y_train)
stack_pred = lr_final.predict(new_X_test)

print('최종 메타 모델의 예측 정확도: {0:.4f}'.format(accuracy_score(stack_pred, Y_test)))
```

최종 메타 모델의 예측 정확도: 0.8055

| 핵심소스코드(4)

- stacking

```
19 # 학습된 개별 모델들이 각자 반환하는 예측 데이터 셋을 생성하고 개별 모델의 정확도 측정.
20 rf_pred = rf_clf.predict(X_test)
21 ada_pred = ada_clf.predict(X_test)
22 lgb_pred = lgb_clf.predict(X_test)
23 xgb_pred = xgb_clf.predict(X_test)
24
25 print('랜덤 포레스트 정확도: {0:.4f}'.format(accuracy_score(y_test, rf_pred)))
26 print('에이다부스트 정확도: {0:.4f}'.format(accuracy_score(y_test, ada_pred)))
27 print('LGB 정확도: {0:.4f}'.format(accuracy_score(y_test, lgb_pred)))
28 print('XGB 정확도: {0:.4f}'.format(accuracy_score(y_test, xgb_pred)))
```

랜덤 포레스트 정확도: 0.9493
에이다부스트 정확도: 0.9154
LGB 정확도: 0.9366
XGB 정확도: 0.8985

▼ Stacking

```
1 # 시험데이터로 예측한 4가지 모델의 결과를 합침
2 pred = np.array([rf_pred, ada_pred, lgb_pred, xgb_pred])
3 print(pred.shape)
4
5 # transpose를 이용해 행과 열의 위치 교환. 컬럼 레벨로 각 알고리즘의 예측 결과를 피쳐로 만들.
6 pred = np.transpose(pred)
7 print(pred.shape)
8
9 # 최종 분류기 모델 생성
10 lr_final = LogisticRegression(C=10)
11
12 # 최종 분류기 학습 및 예측
13 lr_final.fit(pred, y_test)
14 final = lr_final.predict(pred)
15
16 print('최종 메타 모델의 예측 정확도: {0:.4f}'.format(accuracy_score(y_test, final)))
```

| 참조- 개발 환경 및 설명

구분		항목	적용내역
SW 개발 환경	웹 개발	Django	데이터베이스 기반 웹사이트를 개발하기 위해 사용
		GitLab	구성원 간 협업 효율 향상 및 프로젝트 형상 관리
	기능 개발	대신증권 API	종목 실시간 정보를 수집하고, 종목별 전망 분석에 사용
		네이버뉴스	증권 관련 뉴스를 크롤링하여 감성 분석에 이용
	서버 개발	MongoDB	감성분석을 위해 종목 관련 뉴스 텍스트를 수집
		mysql	주가정보, AI예측, 감성분석 결과를 저장
		Apache(1.7.1)	관리자 웹 페이지를 구동하는 웹 서버
		Amazon Web Service	컴퓨팅 클라우드 시스템

한이음 ▶ 프로그램 설계서

정수민

2021-08-20 04:31

pushed

branch develop

Modify urls file

김지현

2021-08-13 14:49

pushed to

pushed branch develop

add post template

김지현

2021-08-13 14:44

pushed to

pushed branch develop

add post template

김지현

2021-08-13 09:52

pushed to

pushed branch develop

add post.html

김지현

2021-08-13 09:47

pushed to

pushed branch develop

null

안정은

2021-08-13 09:39

pushed new

created branch account

null

김지현

2021-08-13 09:36

pushed to

pushed branch develop

add post.html

조설아

2021-08-13 09:04

pushed to

pushed branch develop

portfolio is first commit

한이음 ▶ 프로그램 설계서

MS Teams

모든 팀

동

동학개미

일반

공지사항

데이터수집

설계

주식

참고자료

Django

ELK

NLP

plan

참고자료

게시물

파일

+

2. 오드락 : 증감인증식도 아닌 증편식에 따른 시인할 인간에 편이될수 있는 오일 증편법. 관리하던 사용자들의 불편함을 해소하고 편리한 UX/UI로 사용자들에게 좋은 평가를 받

3. 더 보기

← 회신

2021년 7월 12일 월요일

김지현

월요일 오후 3:51

주식 예측 참고자료

<https://tbacking.com/2020/11/01/%EC%A3%BC%EC%8B%9D-%EC%98%88%EC%B8%A1-%EA%B4%80%EB%A0%A8%EB%85%BC%EB%AC%B8-%EC%A0%95%EB%A6%AC/>

Thank you

