

단타충

단타도 타율이 충분하다면



단타충?

주식 투자자들을 위한 실시간 이슈 종목 알림 어플리케이션



팀원 소개

어플팀

모델팀



이범석 (팀장)
Front-end, Git 관리
Firebase 데이터베이스 구축



이건민
Front-end, UI/UX 디자인



이승준
ai 모델 제작, 크롤링 모델 구축



이원형
ai 모델 제작, 학습 데이터 구축

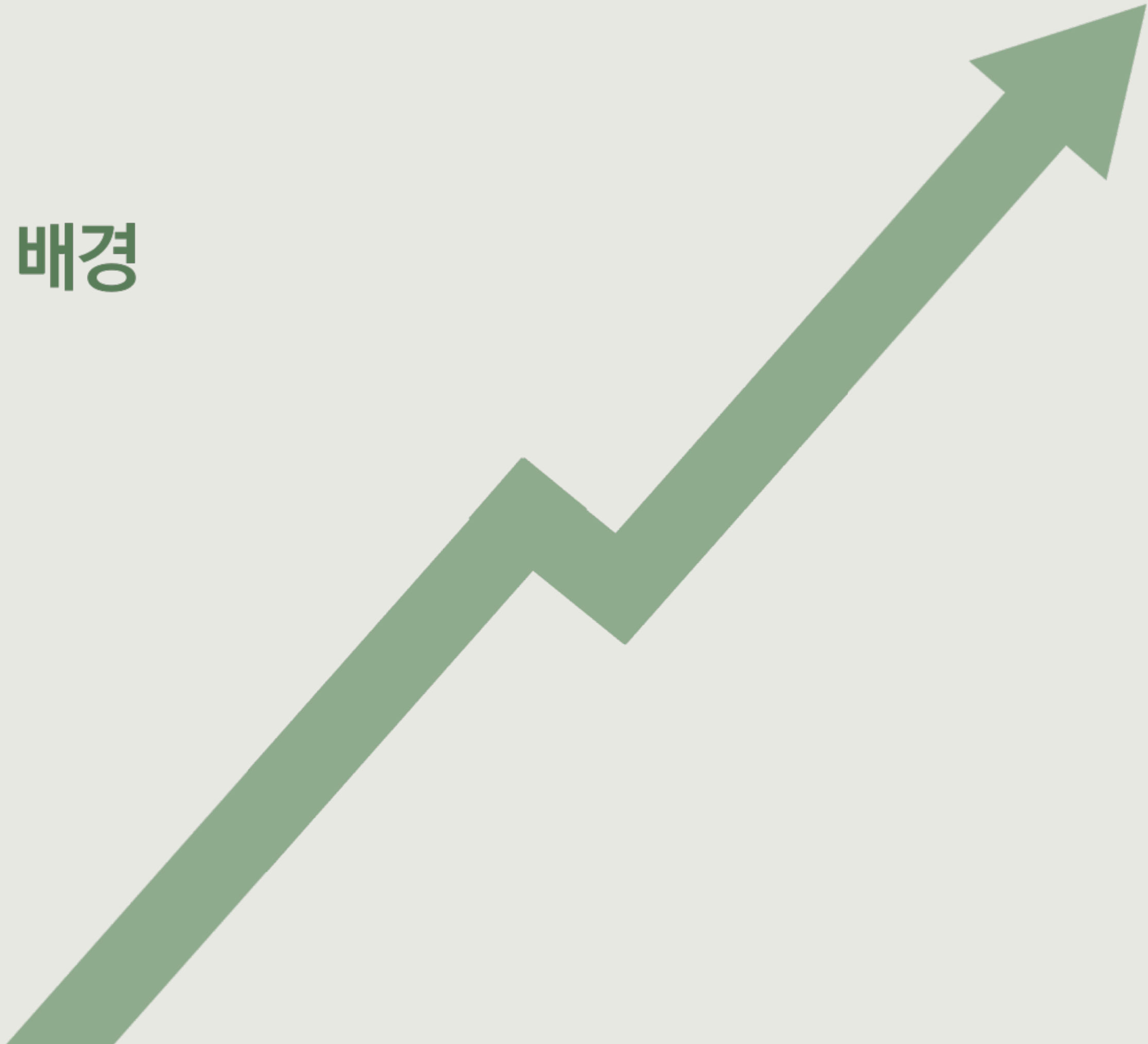


목차

- 01 개발 배경
- 02 시연 영상
- 03 주요 기능
- 04 시스템 구성도
- 05 프로젝트 보완점 및 제한사항
- 06 기대 효과



01 개발 배경





개발 배경



시간 부족



투자 실패





개발 배경

시간을 많이 들이지 않더라도
단기 투자할만한 종목을 알 수 있다면?

게다가 종목의 정보를 알려주고,
기사 분석까지 대신해준다면?





프로젝트 소개 - 개발 배경

그래서 직접 만들었습니다.



02 시연 영상

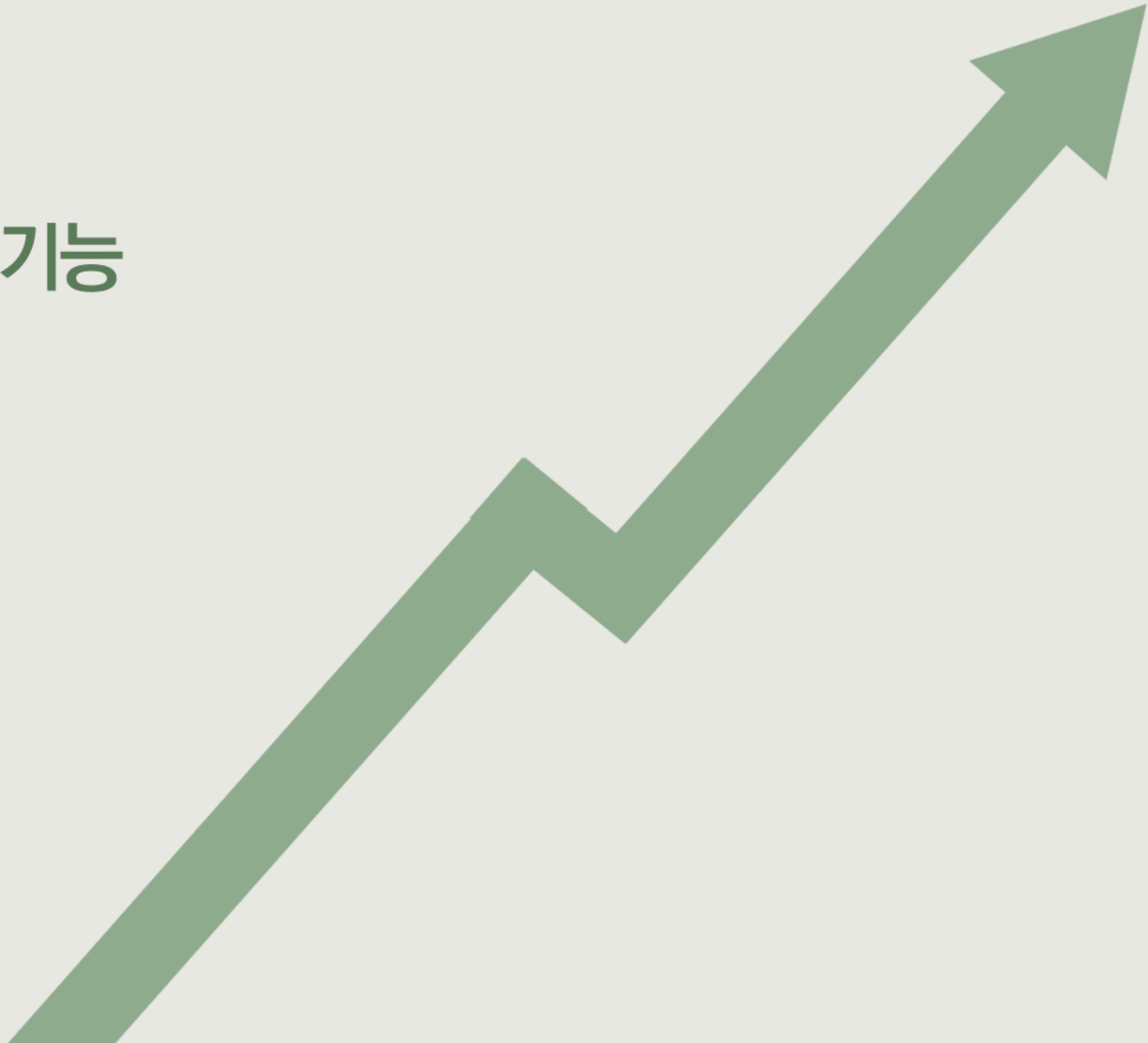




시연 영상

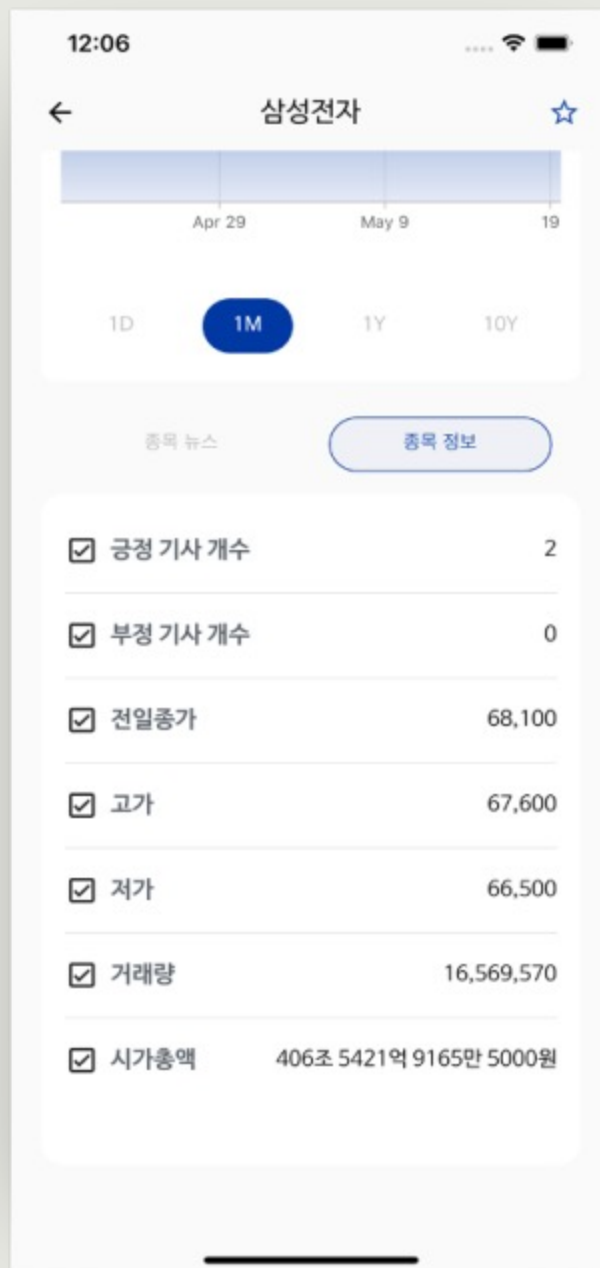


02 주요 기능

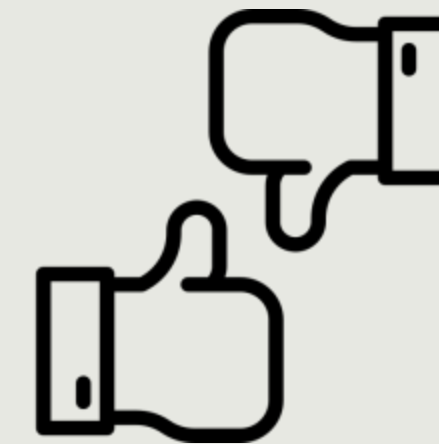




주요 기능



주식 종목에 대한
기본 정보 및 차트 제공





기사 분석을 통한
호/악재 판별





구현 방법 - 모델 학습

11:49

****  

단타도 타율이 충분하다면

시세 순위

상승 종목

하락 종목

해인

9,800

▲ 2,260

29.97%

174개

HLB글로벌

11,700

▲ 1,970

20.25%

272개

수산중공업

4,450

▲ 520

13.23%

0개

다스코

6,870

▲ 770

12.62%

0개

현대에너지솔루션

27,150

▲ 2,900

11.96%

0개

뉴스 순위

기사 개수 순위

호재 순위

악재 순위

CJ

81,000

▼ -2,100

-2.53%

523개

기아

82,100

▼ -2,400

-2.84%

486개

현대건설

40,200

▼ 3,000

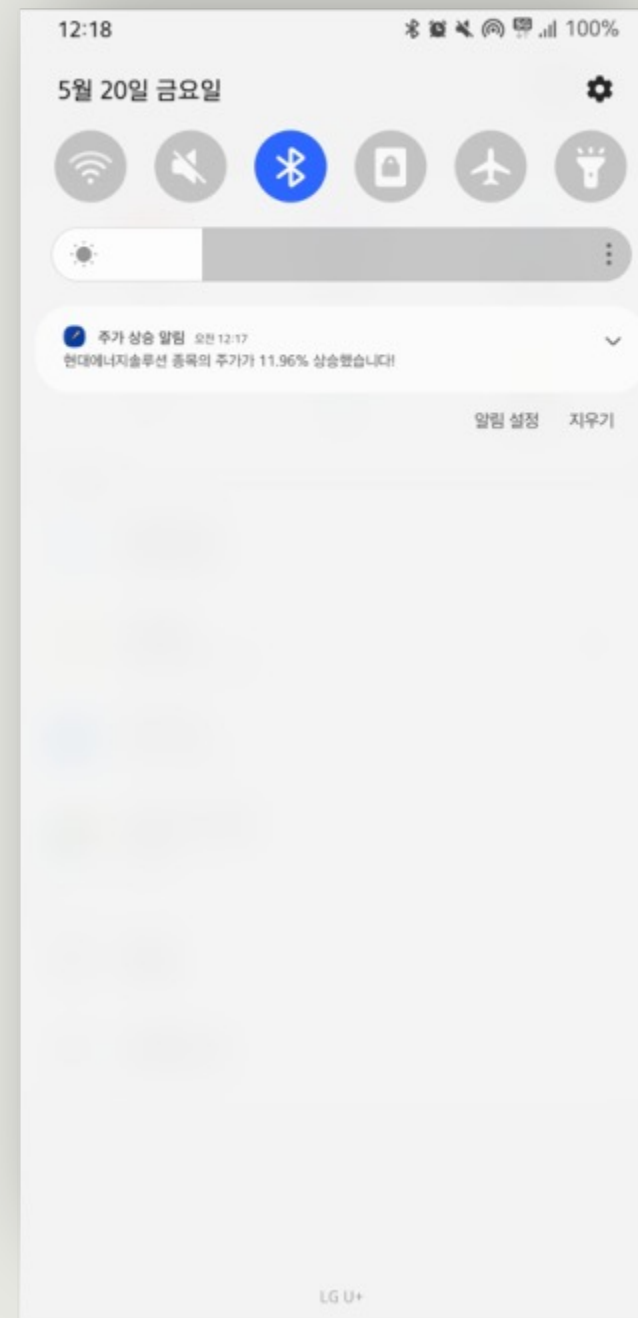
-3.37%

390개

홈



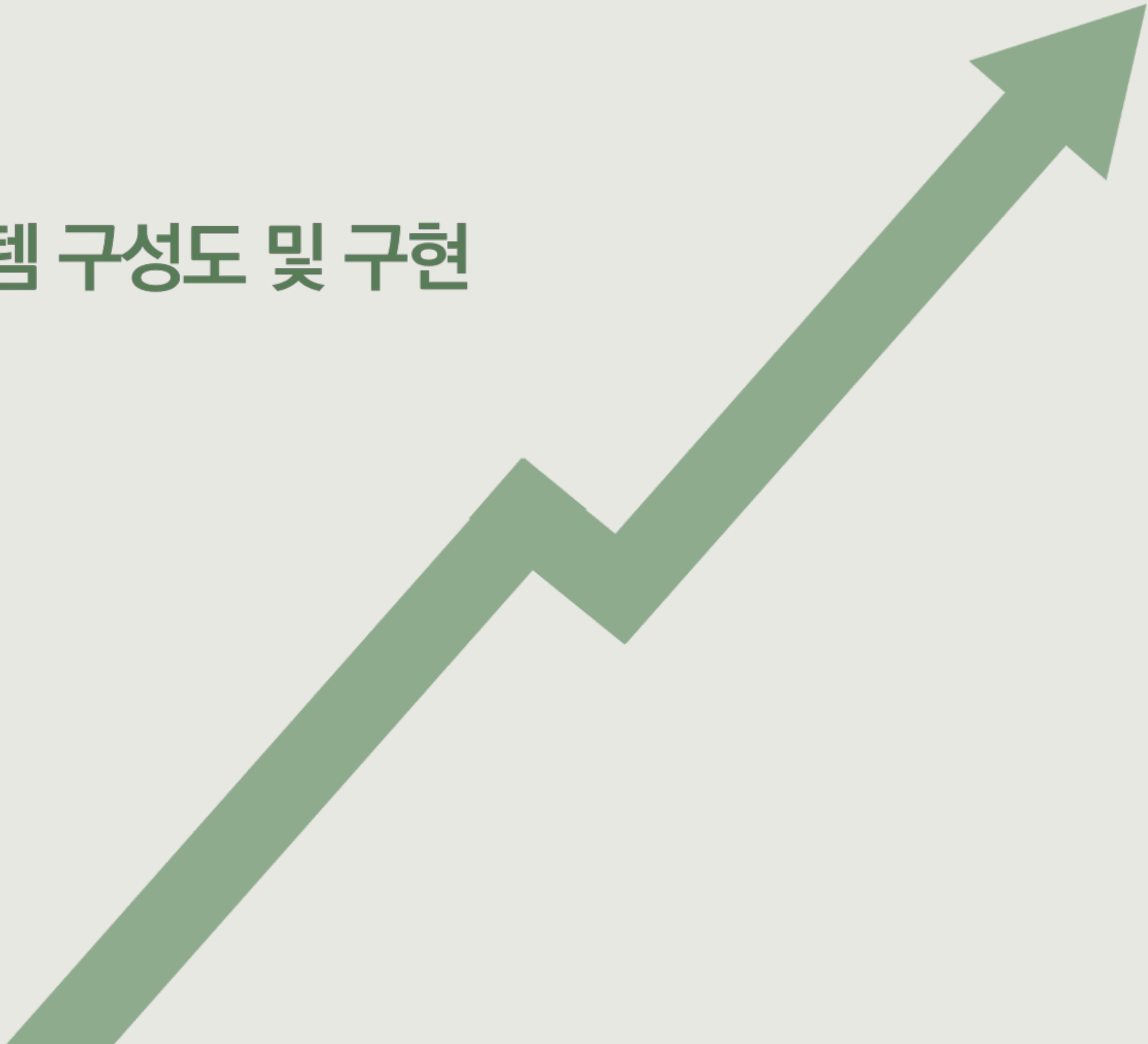
이슈 종목에 대한
종목 랭킹 제공



주가 변화가 큰 종목과
이슈 종목의 알림 제공

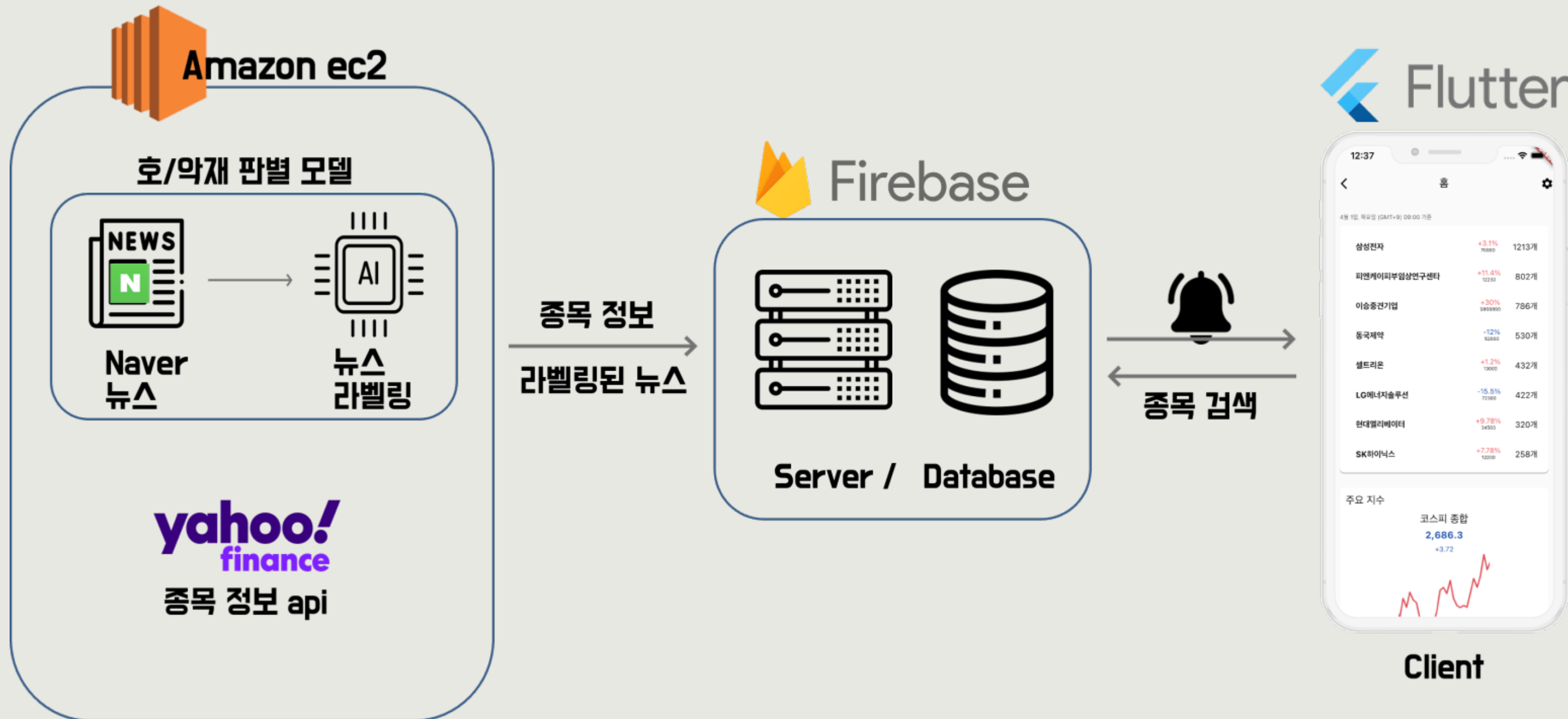


03 시스템 구성도 및 구현





시스템 구성도





구현 방법 - 학습 데이터 수집

N 삼성전자

인사이트 3분 전
뒷차 시야 가리니까 '대형 모니터'로 앞 도로 상황 보여주는 트럭 운전자 (영...
한편 지난 2016년 삼성전자는 위 영상과 비슷한 '삼성 세이프티 트럭'을 공개한 바 있다.당시 삼
성은 트럭 뒷면에 삼성 스마트 사인지 4대를 통해 트럭 앞면의 카메라로 촬영되는 도로 상...

아주경제 5분 전
[한·미 정상회담] 尹 "전략산업 분야 협력" 바...
그러면서 전날 평택 삼성전자 반...
과 동행한 첨단 반도체 산업현장...

MBN 13분 전 네이버...
홍준표 "이재용 안쓰러워...
홍준표 국민의힘 대구시장 후보가
성전자 부회장과 이명박 전 대통...

위키리크스한국 13분 전
[바이든 방한] 이재용·최태원...
이 자리에는 이재용 삼성전자 부...
장, 신동빈 롯데그룹 회장, 김동원...

서울경제 20분 전 네이...
[속보] 尹대통령 "한미동맹...
윤 대통령은 전날 저녁 바이든 대...
실을 거론하고 "반도체 산업현장...

title
뒷차 시야 가리니까 '대형 모니터'로 앞 도로
[한·미 정상회담] 尹 "전략산업 분야 협력" 바
홍준표 "이재용 안쓰러워...사면·복권 안 돼
[바이든 방한] 이재용·최태원·정의선...한 자
[속보] 尹대통령 "한미동맹, 경제안보 시대 5
김건희 여사, 오늘 바이든 만나 인사...공식
尹 "전략산업 분야 협력"...바이든 "한미동맹
尹 "한미동맹, 경제안보로 진화해야" 바이든
미, 바이든 첫 아시아 순방..."북 위협 대응...
한미 '공급망·산업대화' 장관급 격상...경제동

크롤링을 통한
15만건의 뉴스 데이터



positive_words.txt

↑ 협력
↓ 굶는
↑ 급등
↑ 신고가
↑ 최고가
↑ 돌파
↑ 행진
↑ 경신
↑ 역전
↑ 상승
↑ 출시
↑ 공급
↑ 선정
↑ 지원
↑ 증가
↑ 촉진
↑ 품귀
↑ 흑자
↑ 흑자전환
↑ 착공
↑ 기대
↑ 체결
↑ 결실
↑ 승인
↑ 임상
↑ 개발
↑ 급증

negative_words.txt

↓ 피소
↓ 처분
↓ 하락
↓ 타격
↓ 추락
↓ 지연
↓ 감질
↓ 잡음
↓ 적자
↓ 적전
↓ 위기
↓ 최저
↓ 압수수색
↓ 감소
↓ 금감
↓ 영업손실
↓ 폭락
↓ 패소
↓ 약세
↓ 사망
↓ 악재
↓ 배임
↓ 하한가
↓ 하한가
↓ 피해
↓ 축소
↓ 실패
↓ 논란
↓ 허위
↓ 위반

tf-idf를 활용한 단어장 제작



title	label
출시 3주년 진로 누적 판매 10억병 돌파	1
스타트업 투자 확대	1
이번엔 테라 레깅스 선보인다	0
맥주보다 인기 있는 병따개 스푼너 품귀 왜	1
지난해 소주 수출 역대 최대	1
브랜드 진로 누적판매 10억병 돌파 소주 대표 브랜드로 성장	1
17대 한중현 사장 취임	0
잇치 가글액 출시	1
제약 활명수 124주년 기념판 수익금 기부 외	1
사진 2016 서울모터사이클쇼 바이크와 미녀	0
사진 레이싱모델 이다희 바이크와 함께 매력발산 중	0
창원시 내년 악취관리 어떻게 하나	0
창원산단 11개 기업 악취감축 자율동참	0
대구 중구 보험 업무협약 체결	1
경남정보대 가족회사 협약식	1
남동구 와 업무협약 체결	1
보험계약자 보호의무 위반으로 금감원에 제재 당해	-1
1분기 영업이익 1517억 전년비 67.9%↑	1
손보사 2019년 3분기 실적 분석 코리안리 자산운용이익 뚝쑥 전년비	1
메리츠금융 에 700억 출자	0
더벨 투심 위축은 남의 일 첫 오버부킹	1

긍/부정 단어 여부를
활용한 데이터 셋





구현 방법 - 모델 학습



```
del_list = [
    "장애",
    "모친상",
    "동향",
    "부친상",
    "부고",
    "소외층",
    "오늘의",
    "소식",
    "증권투자정보",
    "시황정보",
    "주식정보",
    "주식시황",
    "뉴스",
    "급락주",
    "마감",
    "주요",
    "급등주",
    "증시일정",
    "캐리더",
    "안타",
    "홈런",
    "가드",
    "포워드",
    "손흥민",
    "골프",
    "유니폼",
    "창단",
    "핸드볼",
    "관중",
    "배구",
    "세터",
    "스파이크",
    "득점",
    "도움",
    "리베로",
    "스포츠",
    "훈련",
]
```

스포츠, 연예 뉴스 등
불필요 데이터 제거



Lstm

Lstm으로 모델 학습 진행





구현 방법 - 모델 선정 과정

신경망 모델 비교 과정

```
1 print("test data : ",len(X_test))
2 print("\n 테스트 정확도 : {:.2f}%".format(model.evaluate(X_test, y_test)[1]*100))
```

```
test data : 1001
32/32 [=====] - 0s 10ms/step - loss: 2.7387 - accuracy: 0.8322
```

테스트 정확도 : 83.22%

Lstm + softmax

```
print("test data : ",len(X_test))
print("\n 테스트 정확도 : {:.2f}%".format(model.evaluate(X_test, y_test)[1]*100))
```

```
test data : 1001
32/32 [=====] - 0s 6ms/step - loss: 2.9084 - accuracy: 0.8232
```

테스트 정확도 : 82.32%

GRU + softmax

```
print("test data : ",len(X_test))
print("\n 테스트 정확도 : {:.2f}%".format(model.evaluate(X_test, y_test)[1]*100))
```

```
test data : 1001
32/32 [=====] - 0s 4ms/step - loss: 1.4712 - accuracy: 0.8272
```

테스트 정확도 : 82.72%

RNN + softmax

```
print("test data : ",len(X_test))
print("\n 테스트 정확도 : {:.2f}%".format(model.evaluate(X_test, y_test)[1]*100))
```

```
test data : 1001
32/32 [=====] - 0s 6ms/step - loss: 1.7687 - accuracy: 0.8192
```

테스트 정확도 : 81.92%

RMSprop

```
print("test data : ",len(X_test))
print("\n 테스트 정확도 : {:.2f}%".format(model.evaluate(X_test, y_test)[1]*100))
```

```
test data : 1001
32/32 [=====] - 0s 12ms/step - loss: 3.0934 - accuracy: 0.8252
```

테스트 정확도 : 82.52%

sigmoid

```
1 print("test data : ",len(X_test))
2 print("\n 테스트 정확도 : {:.2f}%".format(model.evaluate(X_test, y_test)[1]*100))
```

```
test data : 1001
32/32 [=====] - 0s 11ms/step - loss: nan - accuracy: 0.0989
```

테스트 정확도 : 9.89%

ReLu





구현 방법 - 모델 학습 결과

```
Epoch 1/10
7880/7880 [=====] - 201s 25ms/step - loss: 0.1278 - accuracy: 0.9595 - val_loss: 0.0568 - val_accuracy: 0.9858
Epoch 2/10
7880/7880 [=====] - 200s 25ms/step - loss: 0.0325 - accuracy: 0.9907 - val_loss: 0.0500 - val_accuracy: 0.9869
Epoch 3/10
7880/7880 [=====] - 199s 25ms/step - loss: 0.0146 - accuracy: 0.9956 - val_loss: 0.0635 - val_accuracy: 0.9824
Epoch 4/10
7880/7880 [=====] - 194s 25ms/step - loss: 0.0065 - accuracy: 0.9979 - val_loss: 0.0790 - val_accuracy: 0.9805
Epoch 5/10
7880/7880 [=====] - 193s 24ms/step - loss: 0.0027 - accuracy: 0.9991 - val_loss: 0.0946 - val_accuracy: 0.9781
Epoch 6/10
7880/7880 [=====] - 195s 25ms/step - loss: 0.0022 - accuracy: 0.9993 - val_loss: 0.0809 - val_accuracy: 0.9836
Epoch 7/10
7880/7880 [=====] - 196s 25ms/step - loss: 0.0015 - accuracy: 0.9996 - val_loss: 0.0975 - val_accuracy: 0.9828
Epoch 8/10
7880/7880 [=====] - 196s 25ms/step - loss: 7.9995e-04 - accuracy: 0.9998 - val_loss: 0.1090 - val_accuracy: 0.9585
Epoch 9/10
7880/7880 [=====] - 191s 24ms/step - loss: 8.6455e-04 - accuracy: 0.9997 - val_loss: 0.1069 - val_accuracy: 0.9804
Epoch 10/10
7880/7880 [=====] - 195s 25ms/step - loss: 8.2566e-04 - accuracy: 0.9997 - val_loss: 0.1179 - val_accuracy: 0.9767
```

학습 정확도 : 97%

```
1 print("test data : ",len(X_test))
2 print("\n 테스트 정확도 : {:.2f}%".format(model.evaluate(X_test, y_test)[1]*100))
```

```
test data : 1001
32/32 [=====] - 0s 5ms/step - loss: 2.4759 - accuracy: 0.8352

테스트 정확도 : 83.52%
```

test 데이터 : 83%





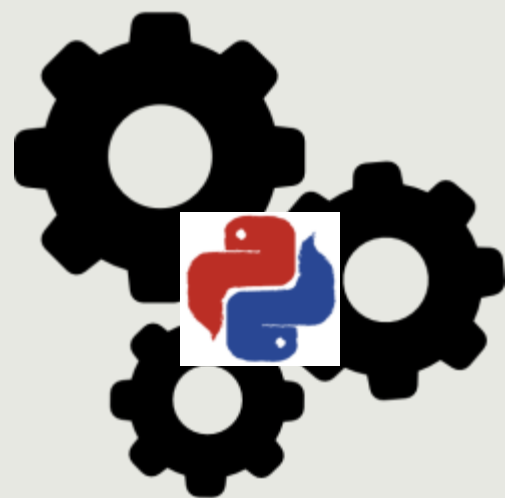
구현 방법 - 시스템 시나리오

NAVER OpenAPI

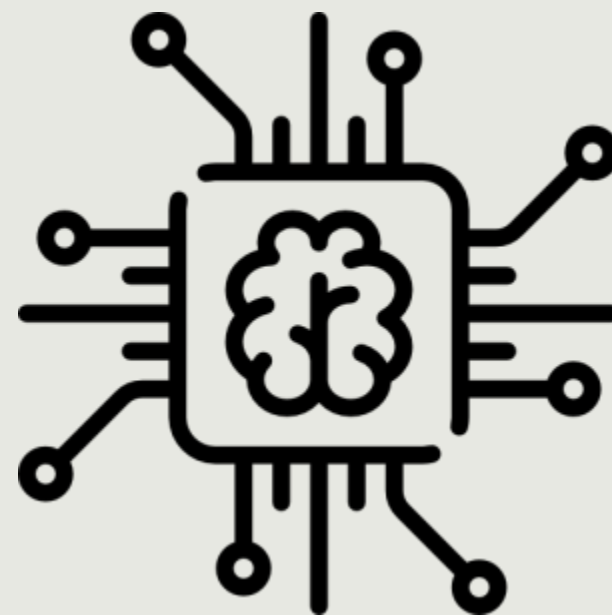
yahoo!
finance

DATA 공공데이터포털
.GO.KR

네이버 api를 통한
종목별 뉴스 기사 및
종목 정보 실시간 수집



데이터 전처리 및
형태소 분석



딥러닝을 통한
호악재 판별



서버로 쿼링



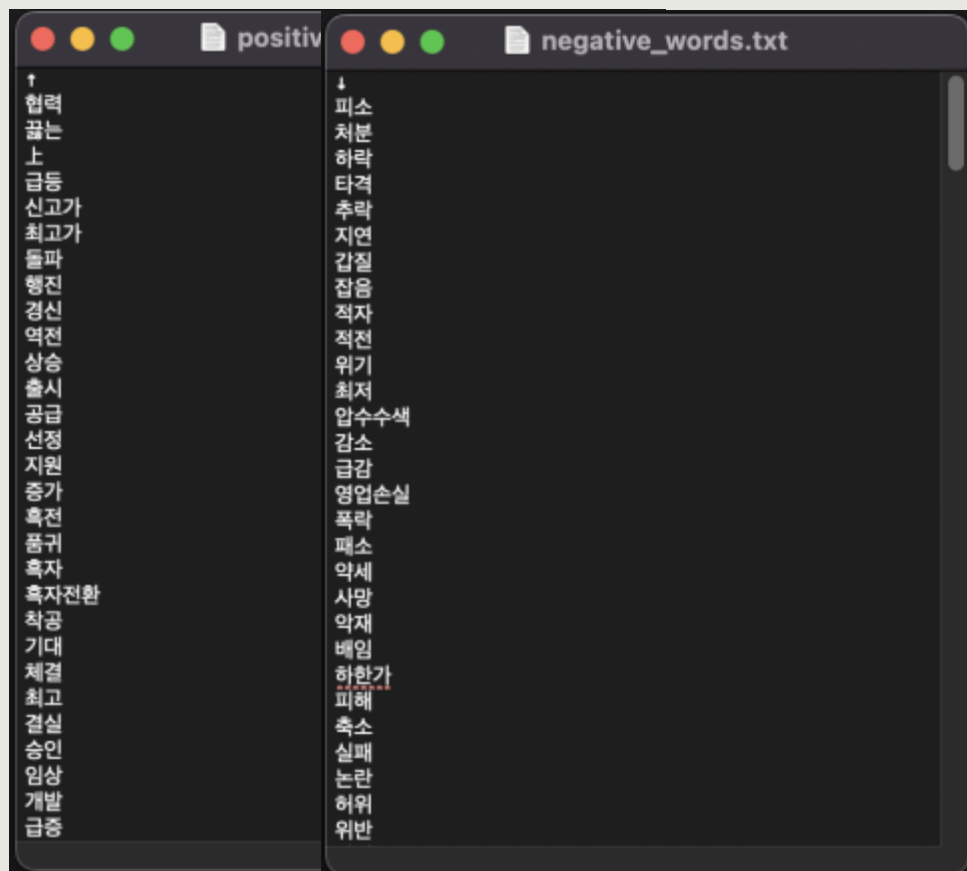
05

프로젝트 보완점 및 제한사항

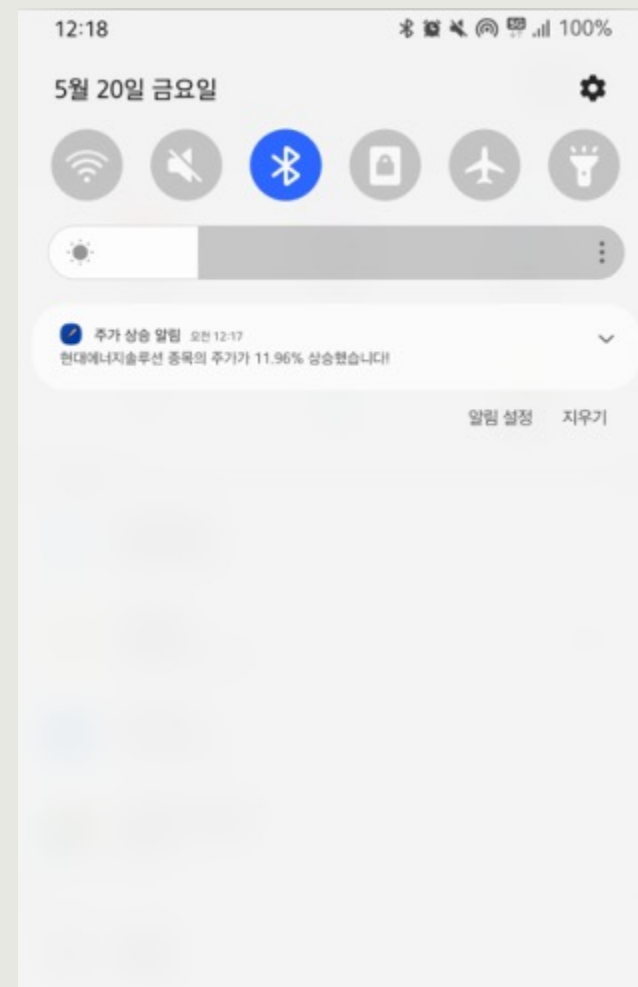




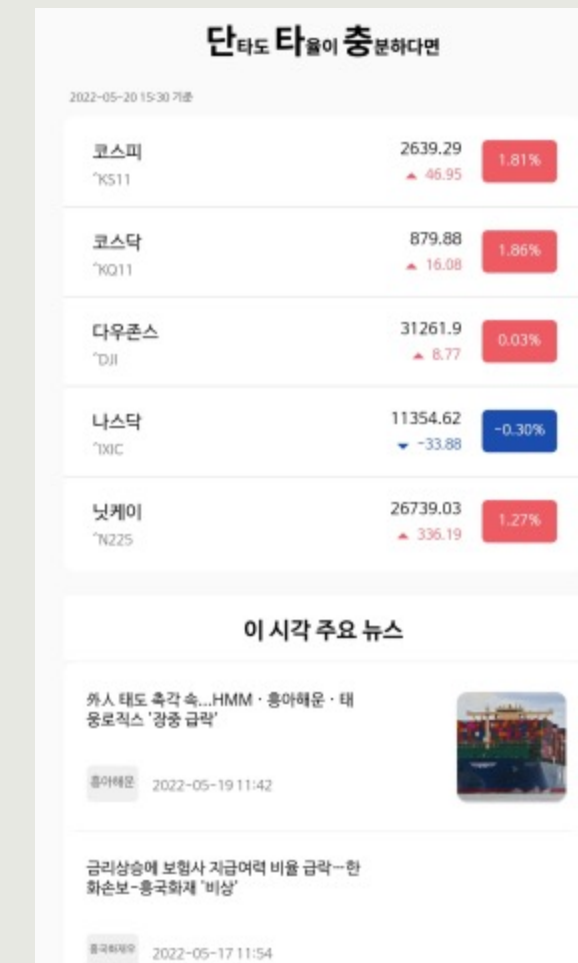
보완한 점



기사의 호악재 판별 기준 및
단어의 가중치



관심종목 알림기능



세계 주요 지수





보완할 점



계열사

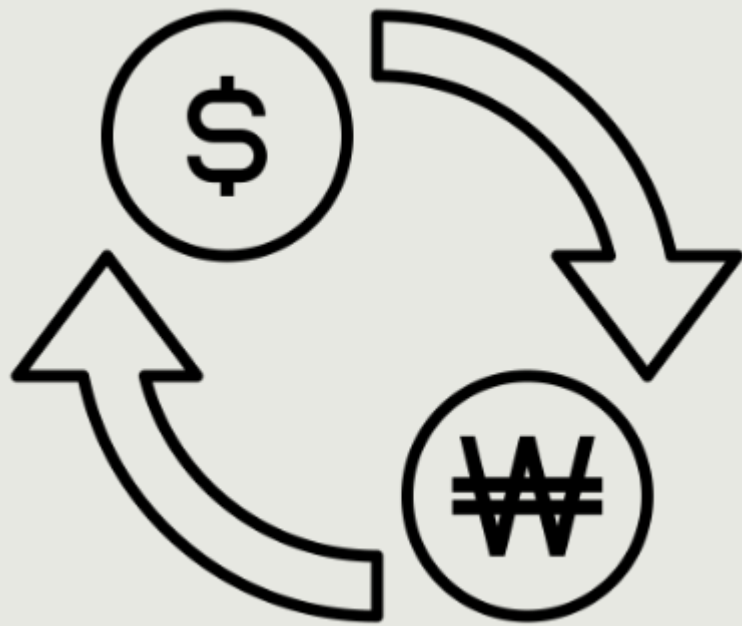
테마주

종목 카테고리





프로젝트 제한사항



환율 미제공

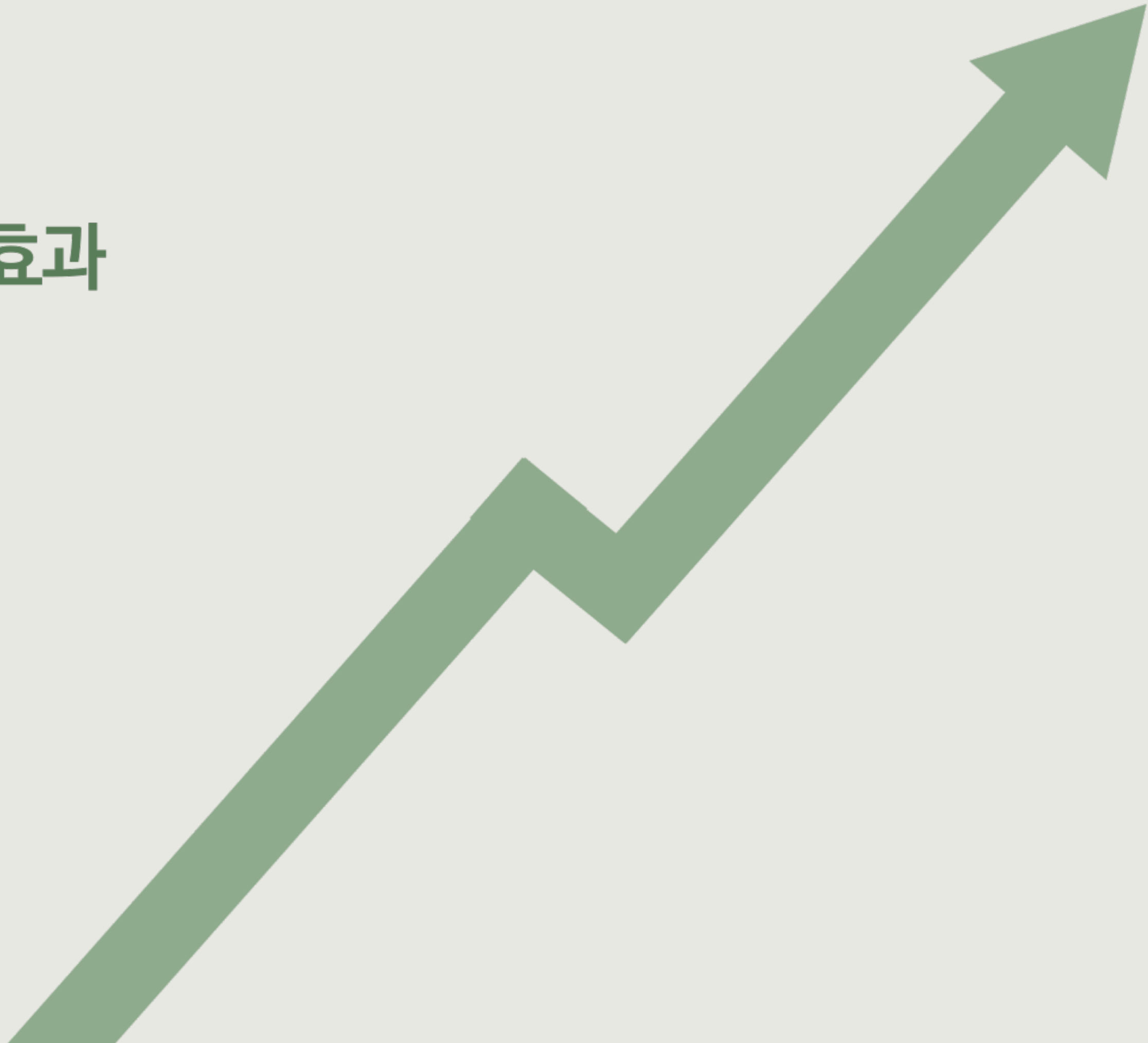
NAVER OpenAPI

api 횟수 제한



06

기대효과





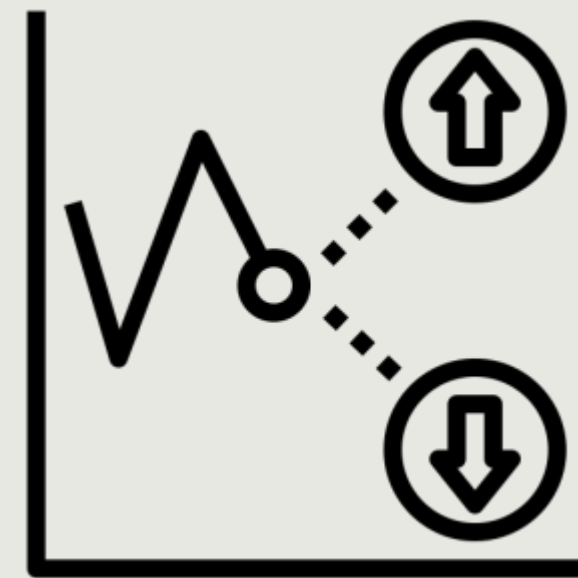
기대효과



편리성



시간 절약



흐름 파악





기대효과





기대효과



단타도 타율이 충분하다면

실시간 이슈 종목 알림 애플리케이션

플레이스토어에서 뵈겠습니다



Q & A

