



개인 투자 성향을 반영한 **Warning System**

신한AI, 보다 나은 금융 생활을 위한 AI 서비스 아이디어 경진대회

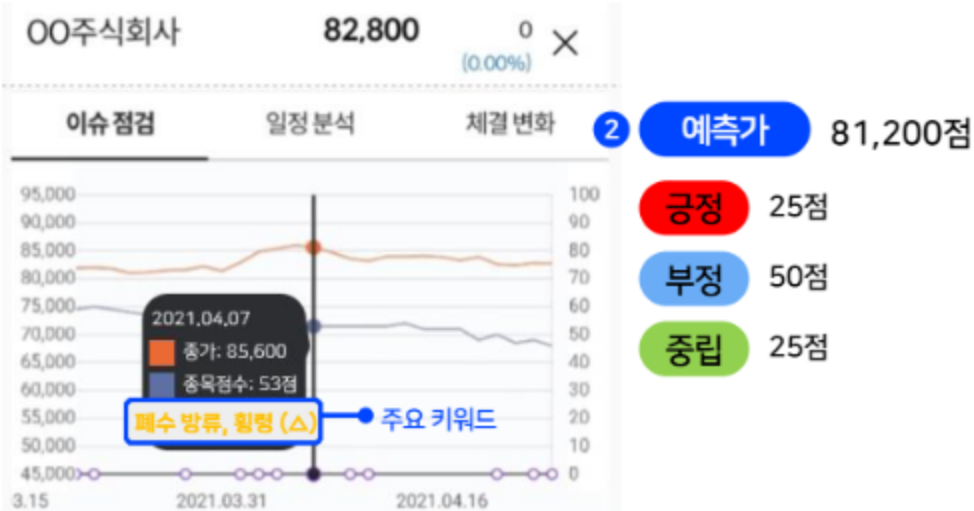
제2의 사춘기

프로젝트 요약

개인 투자 성향을 반영한 Warning System

적용 모델: Retain 모델

- 1 기업의 시점별 "주가 영향 ESG 지표별" 주요 뉴스 및 키워드 제시
- 2 주가 예측 (신한알파 주식 기능에서 확장)



- 1 시점 1 주요 키워드
 - 적대적 인수 합병
 - 금리 인상
- 2 시점 2 주요 키워드
 - 사회공헌 활동
 - 경영진 교체
- 3 시점 3 주요 키워드
 - 신사업 추진
 - 액면 분할

- 3 Index 라벨링 된 세부 뉴스 참조 가능

[신사업 추진] 긍정 2021.09.13

"포드-LG엔솔 튀르키예 공장 짓기로" 버핏, 비야디 지분 추가 매각 外
 미국 포드 자동차가 SK온 대신 LG에너지솔루션과 함께 튀르키예에 전기차 배터리 공장을 짓는다고 블룸버그통신이 현지시간 9일 보도했습니다.

[환경오염 예방] 긍정 2021.10.01

'ESG 경영대상' LG엔솔
 LG에너지솔루션이 '2022 대한민국 ESG 경영대상'에서 종합 대상을 받았다. 대한민국 ESG 경영포럼은 올해 ESG 경영대상 수상 기업을 19일 공개했다. LG에너지솔루션은 환경(E)과 사회(S), 지배구조(G) 등 세 부문 모두에서 'S등급'을 받았다. ESG 경영대상 평가 대상 기업 중 '올 S등급'을 받은 유일한 사례다. 대한민국 ESG 경영포럼은 한국경제신문사와 연세대 동반경영연구센터, IBS컨설팅이 설립한 ESG 플랫폼이다.

Index



문제 정의

문제 인식
솔루션 도출

데이터 및 모델링

모델 개요
모델 플로우

서비스 소개

주요 기능
이용 시나리오
아이디어 차별점

기대 효과

앱 내 활용방안
: 신한알파 투자플러스

타겟 고객 측면의 문제 인식



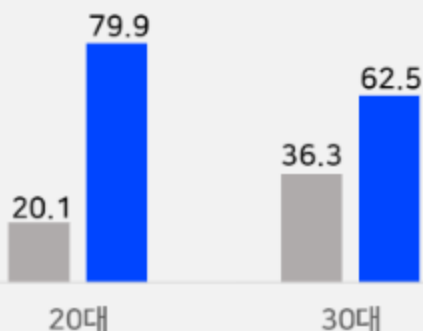
초보 투자자인 MZ세대는 주식 정보를 습득하는데 있어 '편리함'을 가장 중시하며
금융 지식 부족에 따른 '물타기식' 투자로 피해를 보고 있습니다.

● 금융 지식이 부족한 초보 투자자

- MZ세대는 금융지식이 부족하며, 초보자일수록 위험하고 단기적인 수익을 추구하는 특징을 가짐

[2030세대 진단한 금융지식 수준 (2022.07)]

■ 충분함 ■ 충분하지 않음



● 빠르고 간편한 투자 방식 선호

- MZ세대는 쉽고, 빠르고, 편리한 주식을 포함한 재테크 방식을 선호함

[재테크 관련 감성 키워드(2022.10)]



주식 정보 습득 "채널"에서의 Pain Point

- ✓ 본인만의 명확한 기준 없는 무조건적 투자

"어떤 구조로 수익이 나는지 모르지만
유튜브에서 매월 배당금을 준다고 하니 믿고 구매했어요"

"커뮤니티에서 주식을 다들 하길래
8만원대에 산 삼오 주식이 5만원이 됐어요"

- ✓ 신뢰할 수 없는 경로를 통해 정보를 습득



포털 검색
(49.1%)



온라인 커뮤니티
(34.1%)



동영상 플랫폼
(29.9%)



오픈채팅방
(11.7%)



시장 측면의 문제 인식

이에 대응하여 최근 증권 업계 내에서 **개인 투자 성향 관련 서비스**들이 등장하고 있으나,
그 중 **개인이 '보유한' 주식 관리**를 위한 **맞춤형 서비스**는 부재한 상황입니다.

Phase 1 투자 성향 파악

NH투자증권: '투자가 문화로'

- 투자 성향 유형별 전문가 조언 제공

삼성증권: '연금 MBTI'

- 투자성향, 소득, 연령으로 연금투자 성향 분석

한국투자증권: '투자 MBTI'

- 최근 6개월 주식 매매 내역 분석으로
투자 유형 제시

Phase 2 성향별 상품/주식 포트폴리오 추천

신한투자증권: '마이 AI 솔루션'

- AI 데이터 분석 기술로 개인별 예금, 펀드 등
최적 비율 산출해 적합한 상품 추천

키움증권: '키우GO 자산배분랩'

- 투자 목표, 기간, 금액, 성향 종합 분석으로
맞춤형 자산 배분 포트폴리오 제공

What's next? 보유한 개별 주식 관리

?



"개인별 투자 성향 기반 보유 주식 Warning 서비스"

보유한 주식에 과거 패턴과 유사한 Risk 이슈 발생이 예측되면,
본인의 투자 성향에 따라 Risk Warning을 알려 투자 결정을 돕습니다.

휩쓸리지 않는, 과거로 배우는 주식 관리

산발적인 정보에 휩쓸리는 MZ세대에게,

본인 투자 성향에 맞는 투자 결정 추천으로
본인만의 투자 패턴을 형성하는데 도움을 줍니다.

핵심 이슈를 간편하게 확인하여 투자 결정 도움

어떤 정보부터 봐야 할 지 모르는 MZ세대에게,

과거 데이터에 기반한 주가에 영향력을 주는 이슈를
빠르고 편리하게 파악하도록 하여
정확한 투자 결정을 도움

Retain 모델 개요

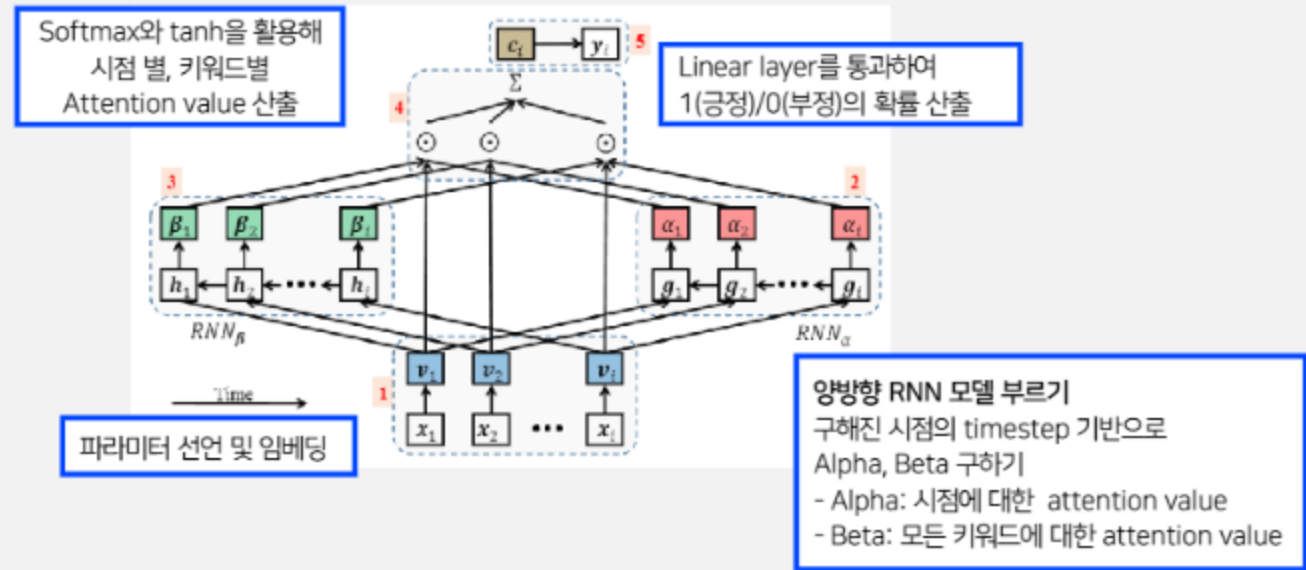
등장 배경

- 환자들의 순차적인 진료기록을 통해
1) 환자 별 질병 발생 확률 2) 시점/진료 별 영향력 확인을 위해 제안
- RNN 모델의 단점 보완
: Sequential 한 데이터에 강하고 예측력이 높은 RNN 모델의 해석이 어렵다는 단점을 극복하고자 Attention 접근방식을 이용하여 해석 가능함

접근 방식

- Seq2Seq 모델을 활용하여 시간적으로 축적되는 정보(뉴스)를 통해 해당 기업의 주가 등락을 예측
- 각 뉴스 키워드의 시점 별, Attention Weight를 통해 예측하고, 구해진 Attention Weight를 활용하여 각 시점 별 뉴스의 영향력 파악 가능

모델 흐름



Sequential 하고 High dimensional 한 데이터에 대해

- 1) Seq2Seq(RNN)을 활용하여 등락에 대한 예측 제공
- 2) Attention을 활용함으로써 예측에 대한 해석을 제공

ESG 지표 별 예측과 해석에 사용

Data & Feature Engineering

조건 설정

모델 구현 및 훈련

결과

① 뉴스 크롤링

- 분석을 진행할 기업 뉴스 크롤링

: 네이버 뉴스에서 관련도 순으로
필터링이 된 기사를 크롤링하여
기사 제목, 날짜, URL, 내용 수집



- 크롤링한 기사 분류

: 뉴스 기반 ESG 지표 감성 분류

→ Fine Tuning된 Bert 모델을 활용

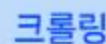
- 1) ESG 카테고리 분류
- 2) 긍정/부정 분류

Bert

② 주가 크롤링

- 해당 기업의 주가를 네이버 증권에서 크롤링

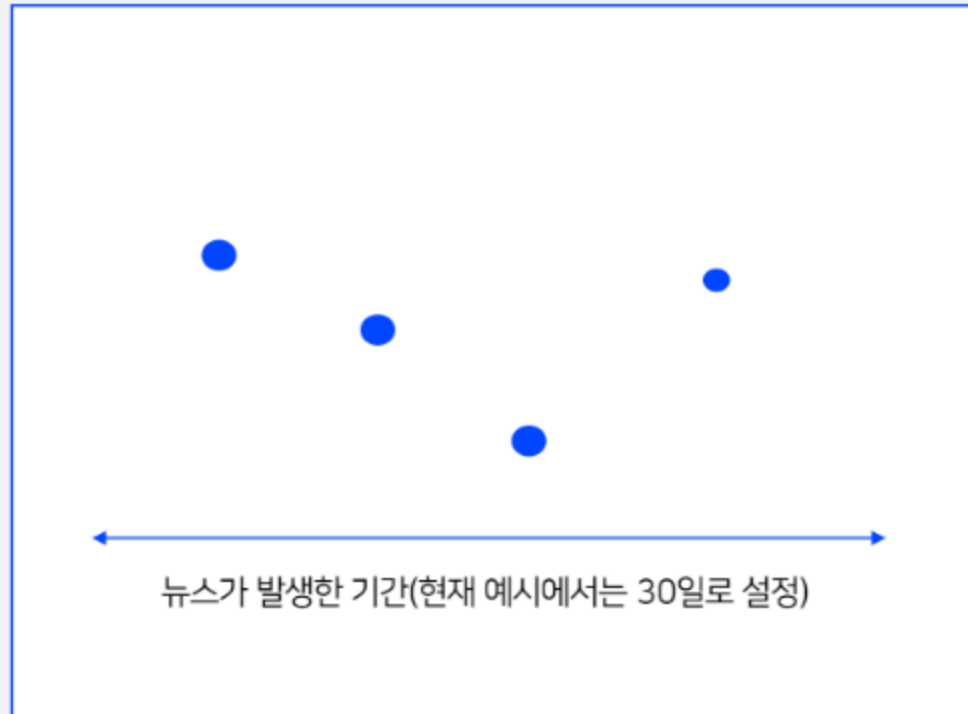
: 뉴스 크롤링 데이터를 날짜순으로
정렬 했을 때, 가장 최근 날짜부터
가장 오래된 날짜까지를 기간으로 잡고
날짜 별 종가, 전일 비, 시가, 고가,
전가 크롤링

[illegible]

모델 플로우: <Appendix> Retain 모델의 적용

Retain 모델 원리 간단 시각화

*임의 설정 및 세분화 설정 가능



예측

긍정

중립

부정

현재 모델의 긍정/중립/부정의 기준
: $\pm 5\%$ *

모델 플로우: 2) 조건 설정



Data & Feature Engineering

조건 설정

모델 구현 및 훈련

결과

Train/Test Data 설명

에코프로비엠의 2021~2023년 뉴스 및 주가 데이터 활용, 그 중 약 80%를 Train Set, 20%를 Test Set에 활용

*30일 간의 뉴스데이터를 기반으로 100일 후의 주가의 등락을 예측.

이 때, 30일이나 100일 같은 기간에 대한 설정은 임의 설정 가능

*전체적인 흐름을 보고 싶다면, 100일 이상의 뉴스 데이터를 기반으로 조금 더 먼 미래를 예측하는 형태로도 Dataset 변경 가능

- E.G.) 2021년 1월 1일 부터 2021년 2월1일까지의 기간 동안의 뉴스 데이터를 통해서 2월 1일로부터 100일 후의 주가의 등락 예측

```
for i in range(30):
    alpha[:,1,:]
    valueimpact=(beta[:,i,:])
    for k in range(52):
        if x[301,i,k]==0:
            continue
        weight=(model.embedding.weight[:,k])
        weight*valueimpact
        ans=model.Linear(weight*valueimpact)*alpha[:,i,:]
        print(ans.shape)
        print("effect of timestep {} and feature {} for obtaining Negative Result: {}".format(i+1,k,ans[:,1]))
```

해당 코드는 임의의 Test Set에 대하여
30일간의 뉴스 데이터 속에서,
결과 (Negative)에 유의미하게
영향을 미친 요소를 찾는 코드



모델 플로우: 3) 모델 구현 및 훈련

Data & Feature Engineering

조건 설정

모델 구현 및 훈련

결과

Pytorch를 활용한 Retain 모델 구현 (1)

```
import torch.nn as nn
import torch
class RETAIN(nn.Module):
    def __init__(self, input_dim, embedding_dim, hidden_dim) -> None:
        super().__init__()
        ## 1. 기본적인 parameter 선언
        self.hidden_dim=hidden_dim ## dimension of hidden state, i.e) 128,256, ...
        self.input_dim=input_dim ## dimension of input time series. 기간을 얼마나 잘라낼 것인지에 대한 정보를 담음.
        self.embedding_dim=embedding_dim
        ## embedded dimension. 단순히 binary encoding이 경우 정보를 많이 지상할 수 있을 수 있으므로, 심보를 조금 더 다채롭게 표현하기 위해서(빈수등)

        ## 2. 임베딩
        self.embedding=nn.Linear(input_dim,embedding_dim) ## 'Linearly' embedding input dimension.

        ## 3. 양방향으로 RNN 기반의 모델 부르기
        self.GRUalpha=nn.GRU(embedding_dim,hidden_dim,batch_first=True) ## GRU, for obtaining alphas
        self.GRUBeta=nn.GRU(embedding_dim,hidden_dim,batch_first=True) ## GRU, for obtaining Betas.

        ## 4. 구해진 시점에서의 timestep 기반으로 alpha, beta 구하기
        self.LinearAlpha=nn.Linear(hidden_dim,1,bias=True) ## Linear Alpha, making hidden state alpha for every timestep (timestep* hiddendim) into (tin
        self.LinearBeta=nn.Linear(hidden_dim,embedding_dim,bias=True) ## Linear Beta, making hidden state beta for every timestep( timestep*hiddendir
```



모델 플로우: 3) 모델 구현 및 훈련

Data & Feature Engineering

조건 설정

모델 구현 및 훈련

결과

Pytorch를 활용한 Retain 모델 구현 (2)

```
## 5. softmax와 tanh를 통해서 각시점에서의 attention value를 구하기 위한 장치 마련.
self.AlphaSoftmax=nn.Softmax(dim=1) ##(size of timestep)
self.Betatanh=nn.Tanh() ##(size of timestep)

## 6. 마지막 Linear layer을 통과시켜, 1(긍정)의 확률 과 0(부정)의 확률을 구해줌.
self.lastLinear=nn.Linear(embedding_dim,3) ## 마지막에 걸리기 긍정(1)의 확률과 0의 확률을 구해줘야하므로, 2개의 dimension을 출력하도록 하였음
self.lastSoftmax=nn.Softmax(dim=1)

def forward(self,x):
    h0 = torch.zeros(1,x.shape[0],self.hidden_dim)# Initialize hidden state with zeros
    #c0 = torch.zeros(1, x.shape[0], self.hidden_dim) for LSTM, in GRU there is no cell state.

    x=self.embedding(x) ## input 임베딩

    outalpha, _ = self.GRUalpha(x, (h0)) ## outputsize= timestep* hiddensize

    ## 각각의 timestep에 대하여, 1에 대한 hidden state출력해줌. (1번째 시퀀스에서의 hidden state, 2번째 시퀀스에서의 hidden state... 50번째 시퀀스에서의 hidden state)
    ## 논문에서는 alpha 는 조금더 visit자체에 focus를 둔 attention vector
```

모델 플로우: 3) 모델 구현 및 훈련

Data & Feature Engineering

조건 설정

모델 구현 및 훈련

결과

Pytorch를 활용한 Retain 모델 구현 (3)

```
outbeta,_=self.GRUbeta(x,(h0)) ## outputsize= timestep* hiddensize

## 각각의 timestep에 대하여, 그에 대한 hidden state출력에줌. (1번째 시점에서의 hidden state, 2번째 시점에서의 hidden state... 50번째 시점에서의
## 논문에서는 beta는 벡터(feature)들에 조짐을 맞춘 attention vector.

attentionalpha=self.LinearAlpha(outalpha) ##timestep *hiddensize-> timestep* 1
attentionbeta=self.LinearBeta(outbeta) ##timestep *hiddensize-> timestep* embedding_dim

attentionalpha=self.Alphasoftmax(attentionalpha) ##dimension:softmax통과시키는 이유? 가중치를 정규화(normalize해서 )해서 표현하기 위해
attentionbeta=self.Betaatanh(attentionbeta) ##dimension : tanh를 사용하는 이유? softmax와는 다른 어떤 의미를 capture하기 위해서.

alphabeta=torch.mul(attentionalpha[:,:],attentionbeta[:,:]) ## dimension: timestep.
totalsum=torch.sum(alphabeta*x,dim=1) ## sum up attention value to get the probability ,
```



모델 플로우: 3) 모델 구현 및 훈련

Data & Feature Engineering

조건 설정

모델 구현 및 훈련

결과

Pytorch를 활용한 Retain 모델 구현 (4)

```
#print(totalsum.shape)
a=self.lastLinear(totalsum)

b=self.lastSoftmax(a)

return attentionalpha,attentionbeta,b
```

모델 플로우: 3) 모델 구현 및 훈련

Data & Feature Engineering

조건 설정

모델 구현 및 훈련

결과

Retain 모델 훈련

```
epochs=100
for epoch in range(epochs):
    for batch_idx, samples in enumerate(train):
        size=len(train)
        x_train,y_train=samples
        _prediction=model(x_train)
        loss = loss_fn(prediction, y_train.long())

        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
        #loss, current = loss.item(), (batch_idx + 1) * len(X)
    if batch_idx % 5 == 0:
        loss, current = loss.item(), (batch_idx + 1) * len(X)
        print(f"loss: {loss:>7f} [{current:>5d}/{size:>5d}]")
```

Data & Feature Engineering

조건 설정

모델 구현 및 훈련

결과

1 Positive/Negative/Neutral 별 확률과 확률에 대한 변수의 영향력 확인 가능

Probability of Positive Result : 0.00027898585540242493

Probability of Negative Result : 0.9030212759971619

Probability of Neutral Result : 0.09669972211122513

2 Negative Risk 가 큰 경우, 모든 투자자들에게 Warning 알람

```

for i in range(30):
    alpha[i,:]=
    valueimpact=(beta[i,:])
    for k in range(52):
        if X[301,i,k]==0:
            continue
        weight = (model.embedding.weight[:,k])
        weight*valueimpact
        ans=model.lastLinear(weight*valueimpact*alpha[i,:])
        print(ans.shape)
        print("effect of timestep {} and feature {} for obtaining Negative Result: {}".format(i+1,k+1,ans[:,1]))

```

```

ans=torch.tensor([ 0.])
effect of timestep 1 and feature 28 for obtaining Negative Result: tensor([ 0.3856e+01]) grad_fn=
<SelectBackward0>
ans=torch.tensor([ 0.])
effect of timestep 25 and feature 51 for obtaining Negative Result: tensor([ 0.7516e+01]) grad_fn=
<SelectBackward0>
ans=torch.tensor([ 0.])
effect of timestep 29 and feature 46 for obtaining Negative Result: tensor([ 0.4670e+01]) grad_fn=
<SelectBackward0>
torch.Size([1, 1])
effect of timestep 26 and feature 41 for obtaining Negative Result: tensor([ 0.3479e+01]) grad_fn=
<SelectBackward0>
torch.Size([1, 1])
effect of timestep 30 and feature 41 for obtaining Negative Result: tensor([ 0.4056e+01]) grad_fn=
<SelectBackward0>

```

14원자의 negative_Management_Of_Legal_And_Regulatory_Framework 기 상대적으로 수익이 벌어지는 것에 영향을 줄 수 있음을 확인 할 수 있음

개인 투자 성향을 반영한 Warning System



주요 기능

1 긍정/부정/중립 점수 산정 및 주가 예측

- 날짜 별 "ESG 지표"와 "주가 등락률" 기반 향후 주가 예측
- 긍정/부정/중립 별 점수 기재

→ 1) 종목의 주가 예측치 제시,
2) 분산 분석을 통한 Risk 판단 가능

2 개인 맞춤형 페르소나 제시

- 사용자 간편 설문조사 통한 직접 기입
 - 투자목적, 재산 및 소득 상황, 투자 경험, 금융 상품에 관한 지식, 위험선호도 등을 감안
 - 신한투자증권 내 '투자성향진단' 서비스 이용
- 1) 안정형/ 2) 안정추구형/ 3) 위험중립형/ 4) 적극투자형/ 5) 공격투자형으로 구분

→ 페르소나 유형 별 다른 기준에 따라 개인 맞춤화 된 Warning Sign 결정

3 날짜 별 세부 영향력 有 키워드 및 뉴스 정렬

- 실제 영향력 있는 키워드 및 뉴스 확인 가능
- 주가에 대한 해당 뉴스의 Impact Score 기재

※ 실제 관련 뉴스: 크롤링 통한 탐색 가능 (구현 완료)



WHY US?

1) 과거 주가 분석 ▶ 예상 주가 제시

- Retain/Bert 모델을 활용
- ESG 지표 및 주가 등락율을 변수로 주가 예측

2) 보유 주식 개인화 서비스 無 ▶ 개인 맞춤형 투자 Warning Sign

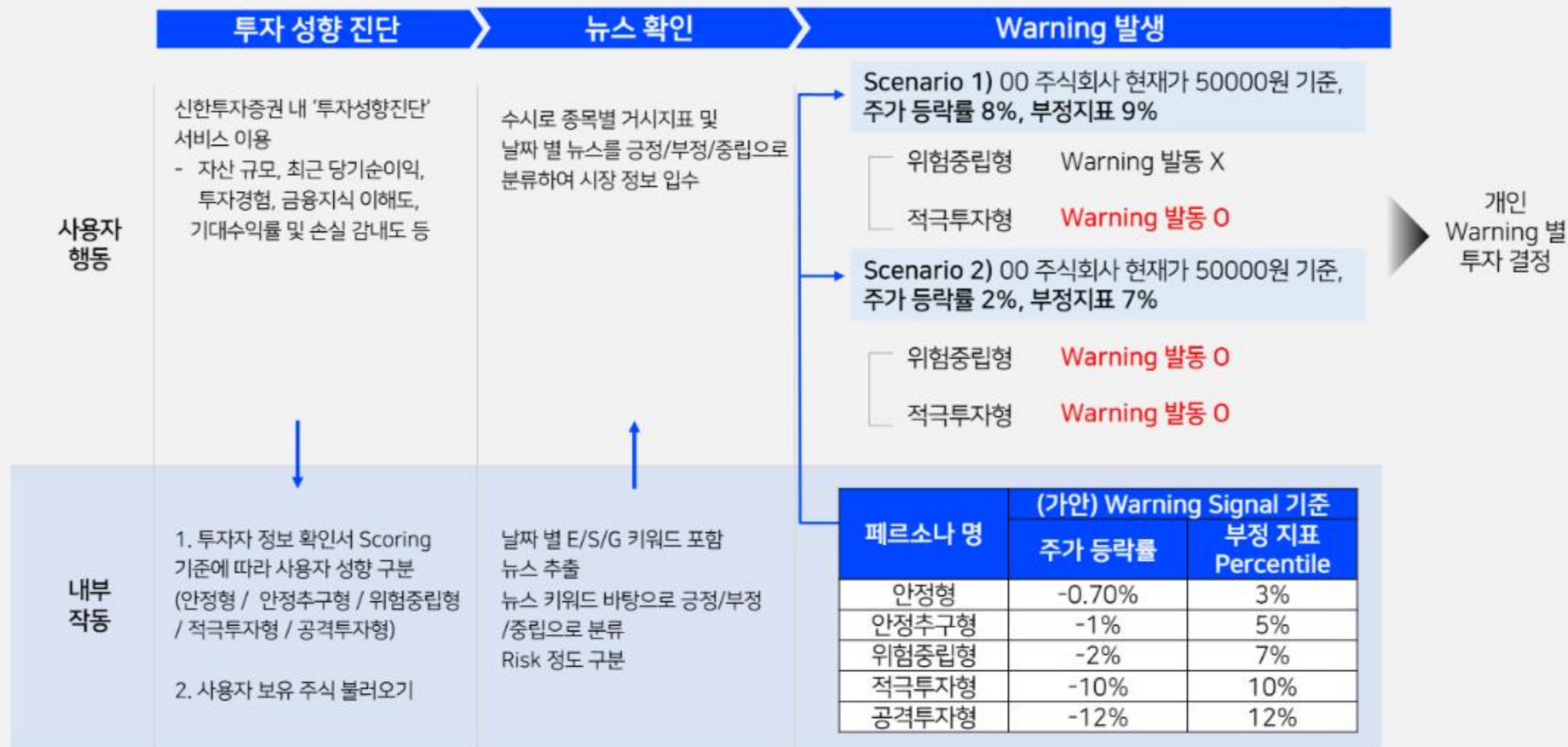
- 개인 페르소나 별 기준에 부합 시 Warning Sign 작동
- 앱 내/ 디바이스 내 푸쉬 알림 작동

+ 소비자들의 이해를 돕기 위한 관련 거시지표 키워드 및 뉴스 제시

- 주가 예측에 영향을 주는 거시지표(E/S/G Index) 키워드를 긍정/부정/중립 분류 및 제시
- 날짜 별 주요 키워드 해당 뉴스 살펴보기 기능 제공



이용 시나리오





As-Is : 현 신한 투자증권 서비스

1



종목 별 최근 이슈 점검 가능

- 향후 주가 예측 無
- 현 주가 원인 분석 無
 - 타사 주가 분석 지표들만 존재
 - 현 주가에 영향을 주는 지표 도출 無

2



보유종목/관심종목의 새로운 이슈 확인

- 시간에 따른 지표 영향력 분석 불가
 - 시간 순의 뉴스 배치 無
 - 지표의 카테고리화 必
- 기업의 포괄적인 단순 이슈만 언급하여
 - 1) 개인별 투자 동향, 2) 카테고리 별 기업 상황에 대한 객관적 판단 불가

To-Be : Retain을 활용한 맞춤형 보유 주식 Risk Warning 서비스

1

종목 별 최근 이슈 점검 기능

- 1일~1달 사이 주가 예측치 제공
 - 주가/ 긍정·부정 Score/ Index 별 Impact Score 제공 통한 투자 근거 제시
- 상세 날짜 별 현재 • 미래 주가에 영향주는 키워드 제공 통한 정확한 원인 분석을 편리하고 가시적으로 확인 가능
 - E/S/G에 해당하는 키워드를 제시하기 때문에 한 눈에 확인 가능
- 개인 맞춤형된 페르소나 별 Risk 알람 서비스 제공

2

보유종목/관심종목의 새로운 이슈 확인

- 정밀 분석이 불가능한 기존의 통합 이슈가 아닌 E/S/G 지표 내 카테고리 별 뉴스 취합
 - 뉴스 별 주가 Impact Score를 제시해 주기 때문에 주가 예측에 대한 객관적인 판단 가능
 - E/S/G 지표 내 카테고리 별 뉴스로 구분해 보여 주어, 경제 지식이 얕은 소비자도 효과적으로 거시적 상황 판단 가능



앱 내 활용방안: 신한알파 투자플러스

1 (기존) 보유종목 브리핑



뉴스 키워드에
긍정/부정/중립 분류 표시

- 보유 주식 관련 최근 이슈, 주요 뉴스 제공을 넘어, 해당 뉴스가 주가에 어떤 영향력을 끼칠 수 있을지 직관적으로 파악 가능

▶ 신한알파 '투자플러스' > '보유종목 브리핑'
에 뉴스 키워드 추가한 모습



2 (기존) 종목 이슈 체크



이슈에 따른
Warning Signal 제공

- 종목별로 이슈를 확인해주는 것을 넘어, 개인 투자 성향별로 투자 행동을 추천해주어 본인 상황에 맞는 투자 결정 용이

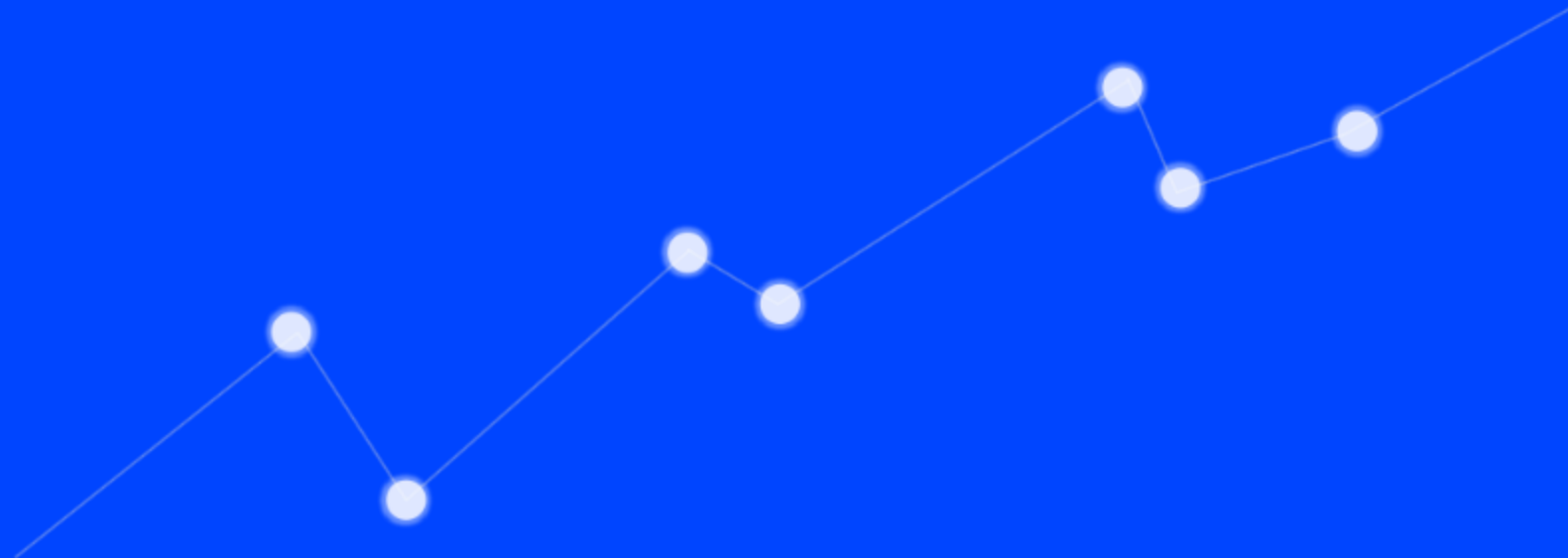
▶ 신한알파 '투자플러스' > '종목이슈체크'
에 Warning Signal 추가한 모습



- 1) 주식을 처음 하는 초보 이용자들 유입 및 Lock-In 효과,
- 2) 투자플러스 내 '프리미엄' 콘텐츠와 결합 시, 부가적 수익 창출 효과 기대 가능



내 주식을 나에게 맞게
스스로 바꿔갈 힘을 기를 수 있도록,



감사합니다