

# 당신의 똑똑한 AI투자친구

## Heymoon!😊



인공지능을 활용하여 주식투자 리스크 관리하기  
“Heymoon” : 관리종목 편입가능성을 예측하는 AI

진아람

# 목차

- I. 프로젝트 진행배경
- II. Time Table
- III. 데이터
- IV. 모델링
- V. 결과
- VI. Q & A



## 프로젝트 진행배경

올해 개인 국내외 주식 순매수 첫 100만

입력 2021/12/26 06:16 | 수정 2021/12/26 06:16

지라시에 호해

증시 퇴출 기업 8년 만에 최다...관리종목 106개도 유의해야

횡령·배임 등 상장 적격성 실질심사 대상 늘어  
거래소 “심사 기준에는 변화 없어”  
“기업 실적 양극화로 도덕적 해이 만연”

노자운 기자

입력 2022.01.03 16:01

증시는 올해 말부터 본격 상승...코스닥 20년 만에 '천스닥'  
증시 개인투자자 급증...하반기 들어 증시 이탈 현상 가속  
증권사 '1조 클럽' 시대...미래·한투 등 연간 영업익 1조원 넘겨

홍승빈 기자 입력 2021-12-28 07:00 | 수정 2021-12-28 07:00

# Time Table

**Data수집 및 전처리**

2022.01.03~2022.01.09

**분석 및  
발표준비**

2022.01.09~2022.01.10

2022.01.09~2022.01.10

**모델구성**

# 사용언어



TensorFlow

BeautifulSoup



python<sup>TM</sup>

matplotlib

XGBoost



seaborn



pandas



NumPy



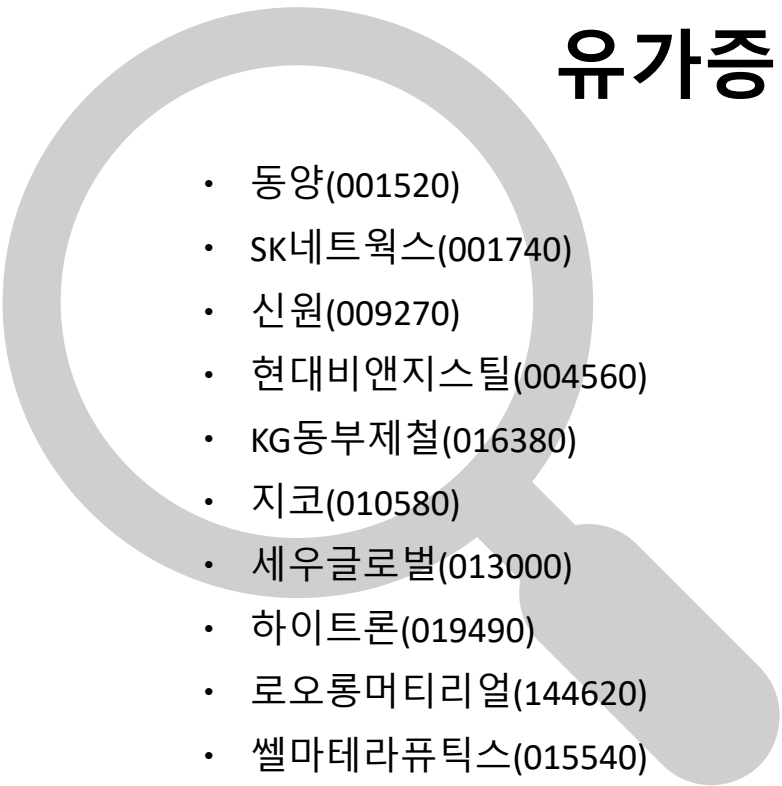
# 데이터 수집

- **KRX** 한국거래소 홈페이지 <https://www.krx.co.kr/main/main.jsp>
- **KIND** 한국거래소 전자공시 홈페이지 <https://kind.krx.co.kr/>
- **Fn-Guide** <https://www.fnguide.com/>
- **DART** 금융감독원 전자공시 홈페이지 <https://dart.fss.or.kr/>
- **네이버금융** [https://finance.naver.com/sise/sise\\_market\\_sum.naver](https://finance.naver.com/sise/sise_market_sum.naver)



# 유가증권시장 **관리종목** 지정 리스트(2022.01.03 기준)

2021년 기준으로 최근 1년 안에 지정된 종목들 총 18개

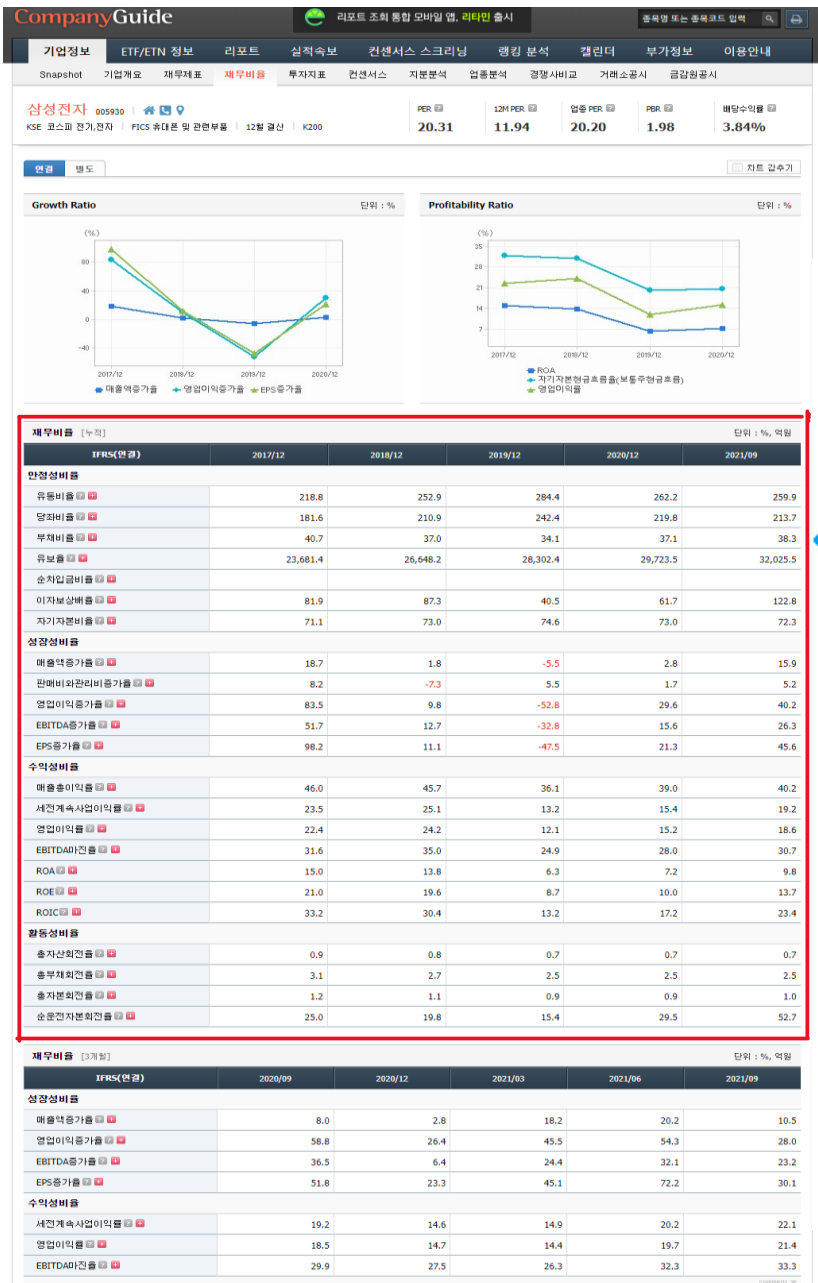
- 
- 동양(001520)
  - SK네트웍스(001740)
  - 신원(009270)
  - 현대비앤지스틸(004560)
  - KG동부제철(016380)
  - 지코(010580)
  - 세우글로벌(013000)
  - 하이트론(019490)
  - 로오롱머티리얼(144620)
  - 쉐마테라퓨틱스(015540)
  - 오리엔트바이오(002630)
  - 센트럴인사이트(012600)
  - 쌍용차(003620)
  - 성안(011300)
  - 와이투솔루션(011690)
  - JW홀딩스(096760)
  - JW생명과학(234080)
  - 세기상사(002420)

“ Target = 1 ”  
투자위험 존재

# 유가증권시장 시가총액 상위 종목 리스트 (2022.01.03 기준) 총 120개

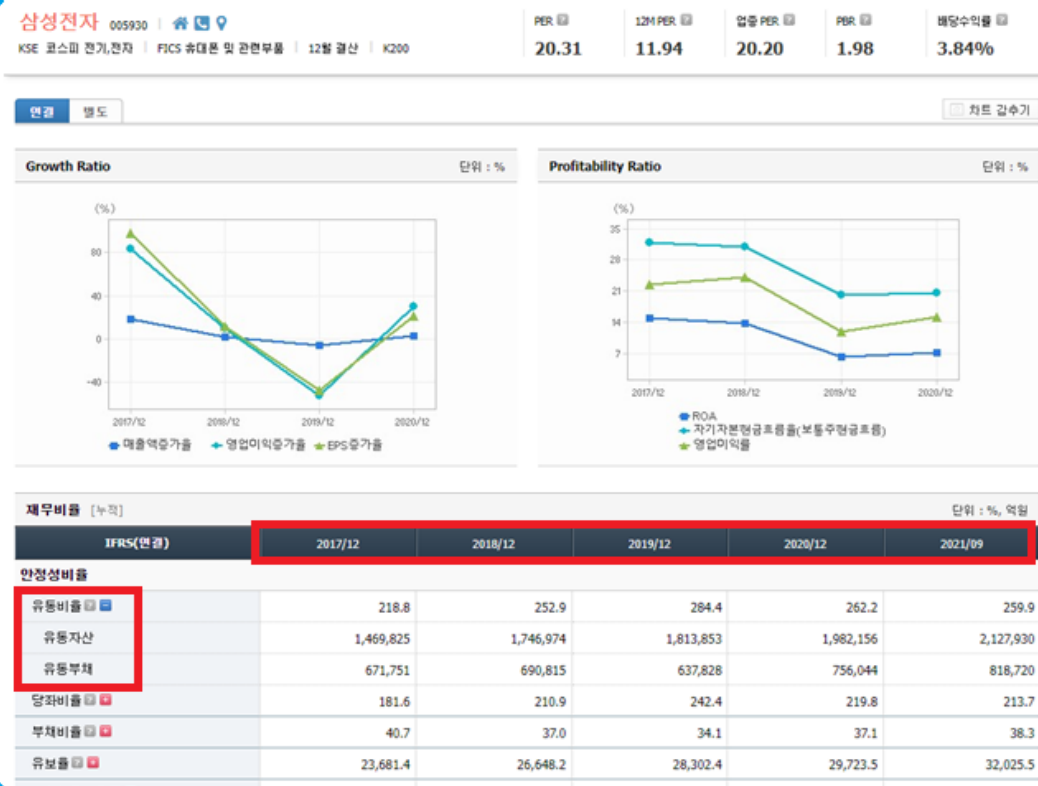
• 삼성전자(005930)	• LG(003550)	• SKC	• GS	• 만도	• 한샘
• SK하이닉스(000660)	• HMM(011200)	• 현대제철	• GS건설	• 포스코인터내셔널	• 영원무역
• 삼성바이오로직스(207940)	• SK텔레콤(017670)	• LG유플러스	• DB하이텍	• 에스원	• 힐라홀딩스
• NAVER(035420)	• 삼성에스디에스(018260)	• CJ제일제당	• 현대오토에버	• SK케미칼	• 더존비즈온
• LG화학(051910)	• 두산중공업(034020)	• 에스디바이오센서	• 한국가스공사	<div>“Target = 0”</div> <div>안정적 투자가능</div>	진하이솔루스
• 삼성SDI(006400)	• KT&G(033780)	• 코웨이	• 아모레G		심
• 현대차(005380)	• S-Oil(010950)	• 강원랜드	• 한미		화학
• 카카오(035720)	• 포스코케미칼(003670)	• 일진머티리얼즈	• KCC		
• 기아(000270)	• 고려아연(010130)	• 금호석유	• 한국		
• 셀트리온(068270)	• 대한항공(003490)	• 삼성중공업	• 한미약품	• OCI	• 코오롱인더
• POSCO(005490)	• 넷마블(251270)	• 현대건설	• 현대미포조선	• 롯데쇼핑	• LS
• 현대모비스(012330)	• 아모레퍼시픽(090430)	• 한국타이어앤테크놀로지	• 한화시스템	• 녹십자	• 대웅
• SK이노베이션(096770)	• LG디스플레이(034220)	• 삼성엔지니어링	• 한솔케미칼	• 신세계	• 한미반도체
• LG전자(066570)	• LG이노텍(011070)	• 유한양행	• GS리테일	• CJ	• 현대백화점
• 삼성물산(028260)	• KT(030200)	• 두산밥캣	• 롯데지주	• BGF리테일	• 동국제강
• SK(034730)	• 롯데케미칼(011170)	• 현대중공업지주	• 한전기술	• 현대로템	• 오뚜기
• LG생활건강(051900)	• 한화솔루션(009830)	• 오리온	• 팬오션	• 한화	• LS ELECTRIC
• 한국전력(015760)	• 한국조선해양(009540)	• 이마트	• 동서	• 후성	• 현대엘리베이
• 삼성전기(009150)	• 한온시스템(018880)	• 쌍용C&E	• 호텔신라	• 하이트진로	• 대웅제약
• 엔씨소프트(036570)	• 현대글로비스(086280)	• 한진칼	• CJ대한통운	• 현대위아	• 한전KPS





# Fn-Guide 홈페이지 살펴보기

## 기업 재무비율(누적)

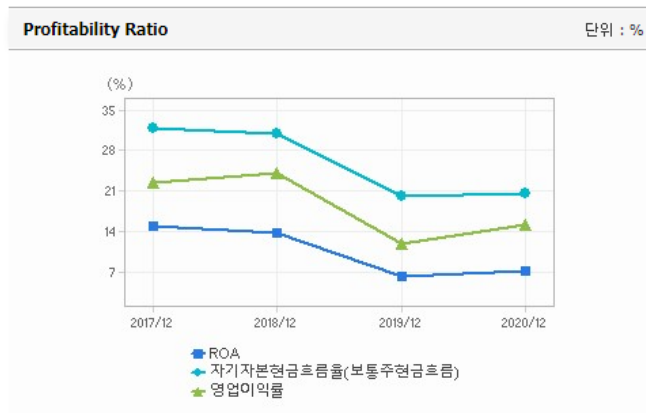
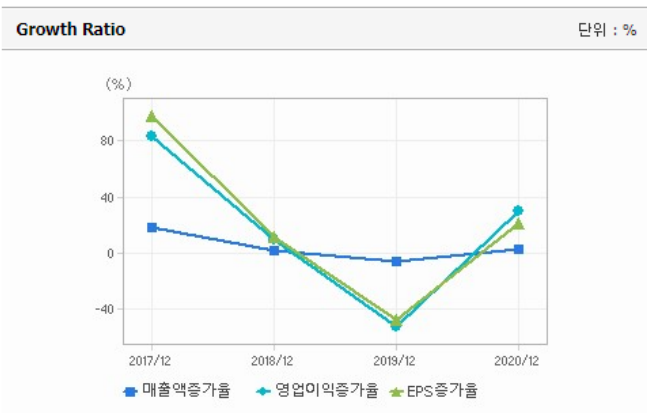


```
aside, figure, video, audio {
margin: > 0;
padding: > 0;
}
```



# Fn-Guide 홈페이지 살펴보기

## 기업 재무비율(누적)



**재무비율 [누적]** 단위 : %, 억원

IFRS(연결)	2017/12	2018/12	2019/12	2020/12	2021/09
<b>안정성비율</b>					
유동비율	218.8	252.9	284.4	262.2	259.9
당좌비율	181.6	210.9	242.4	219.8	213.7
부채비율	40.7	37.0	34.1	37.1	38.3
유보율	23,681.4	26,648.2	28,302.4	29,723.5	32,025.5
순차입금비율					
이자보상배율	81.9	87.3	40.5	61.7	122.8

DevTools is now available in Korean! Always match Chrome's language Switch DevTools to Korean

Elements Console Sources Network Performance Memory Application

```
<!-- Growth Ratio , Profitability Ratio -->
<div class="ul_col2wrap">...</div>
<!-- 재무비율 [누적] -->
<div class="ul_col2wrap pd_t25">
  <div class="um_topbar bshadow">...</div>
  <div class="um_table">
    <table class="us_table_ty1 h_fix zigbg_no">
      <caption class="cphidden">재무비율</caption>
      <thead>...</thead>
      <tbody>
        <tr class="tbody_tit">...</tr>
        <tr id="p_grid1_1" class="rwf acd_dep_start_close"> == $0
          <th scope="row" class="l_cif">...</th>
          <td class="r">218.8</td>
          <td class="r">252.9</td>
          <td class="r">284.4</td>
          <td class="r">262.2</td>
          <td class="r_cle">259.9</td>
        </tr>
        <tr class="c_grid1_1 rwf acd_dep2_sub" style="display:none;">...</tr>
        <tr class="c_grid1_1 rwf acd_dep2_sub" style="display:none;">...</tr>
        <tr id="p_grid1_2" class="rwf acd_dep_start_close">...</tr>
        <tr class="c_grid1_2 rwf acd_dep2_sub" style="display:none;">...</tr>
        <tr class="c_grid1_2 rwf acd_dep2_sub" style="display:none;">...</tr>
        <tr id="p_grid1_3" class="rwf acd_dep_start_close">...</tr>
        <tr class="c_grid1_3 rwf acd_dep2_sub" style="display:none;">...</tr>
        <tr class="c_grid1_3 rwf acd_dep2_sub" style="display:none;">...</tr>
        <tr id="p_grid1_4" class="rwf acd_dep_start_close">...</tr>
        <tr class="c_grid1_4 rwf acd_dep2_sub" style="display:none;">...</tr>
        <tr class="c_grid1_4 rwf acd_dep2_sub" style="display:none;">...</tr>
      </tbody>
    </table>
  </div>
</div>
```

BeautifulSoup

```

keras > _project > 3_pandas_final.py > ...
1 import pandas as pd
2 import requests
3 from bs4 import BeautifulSoup
4 from tabulate import tabulate
5
6 res = requests.get('http://comp.fnguide.com/SV02/ASP/SVD_FinanceRatio.asp?pGB=1&gicode=A005930&cID=&MenuYn=Y&ReportGB=&NewMenuID=104&stkGb=701')
7 soup = BeautifulSoup(res.content, 'lxml')
8 table = soup.find_all('table')
9 df = pd.read_html(str(table))
10 print(tabulate(df[0], headers='keys', tablefmt='psql'))

```

BeautifulSoup

pandas

	IFRS(연결)	2017/12	2018/12	2019/12	2020/12	2021/09
0	안정성비율	안정성비율	안정성비율	안정성비율	안정성비율	안정성비율
1	유동비율계산에 참여한 계정 펼치기	218.8	252.9	284.4	262.2	259.9
2	당좌비율계산에 참여한 계정 펼치기	181.6	210.9	242.4	219.8	213.7
3	부채비율계산에 참여한 계정 펼치기	40.7	37.0	34.1	37.1	38.3
4	유보율계산에 참여한 계정 펼치기	23681.4	26648.2	28302.4	29723.5	32025.5
5	순차입금비율계산에 참여한 계정 펼치기	nan	nan	nan	nan	nan
6	이자보상배율계산에 참여한 계정 펼치기	81.9	87.3	40.5	61.7	122.8
7	자기자본비율계산에 참여한 계정 펼치기	71.1	73.0	74.6	73.0	72.3
8	성장성비율	성장성비율	성장성비율	성장성비율	성장성비율	성장성비율
9	매출액증가율계산에 참여한 계정 펼치기	18.7	1.8	-5.5	2.8	15.9
10	판매비와관리비증가율계산에 참여한 계정 펼치기	8.2	-7.3	5.5	1.7	5.2
11	영업이익증가율계산에 참여한 계정 펼치기	83.5	9.8	-52.8	29.6	40.2
12	EBITDA증가율계산에 참여한 계정 펼치기	51.7	12.7	-32.8	15.6	26.3
13	EPS증가율계산에 참여한 계정 펼치기	98.2	11.1	-47.5	21.3	45.6
14	수익성비율	수익성비율	수익성비율	수익성비율	수익성비율	수익성비율
15	매출총이익율계산에 참여한 계정 펼치기	46.0	45.7	36.1	39.0	40.2
16	제조비용차감이익율계산에 참여한 계정 펼치기	23.5	25.1	13.2	15.4	19.2
17	영업이익율계산에 참여한 계정 펼치기	22.4	24.2	12.1	15.2	18.6
18	EBITDA마진율계산에 참여한 계정 펼치기	31.6	35.0	24.9	28.0	30.7
19	ROA계산에 참여한 계정 펼치기	15.0	13.8	6.3	7.2	9.8
20	ROE계산에 참여한 계정 펼치기	21.0	19.6	8.7	10.0	13.7
21	ROIC계산에 참여한 계정 펼치기	33.2	30.4	13.2	17.2	23.4
22	활동성비율	활동성비율	활동성비율	활동성비율	활동성비율	활동성비율
23	총자산회전율계산에 참여한 계정 펼치기	0.9	0.8	0.7	0.7	0.7
24	총부채회전율계산에 참여한 계정 펼치기	3.1	2.7	2.5	2.5	2.5
25	총자본회전율계산에 참여한 계정 펼치기	1.2	1.1	0.9	0.9	1.0
26	순운전자본회전율계산에 참여한 계정 펼치기	25.0	19.8	15.4	29.5	52.7

# (1) DATA생성 - 관리종목

```
1 import numpy as np
2 import pandas as pd
3 import requests
4 from pandas.core.frame import DataFrame
5 from pandas import Series, DataFrame
6
7 # [ 1: 관리종목 , 0: 안전종목 ] - 유가증권
8
9 #1-1) 관리종목 데이터 생성
10 codes = ['001520', '001740', '009270', '004560', '016380', '010580', '013000', '019490', '144620', '015540', '002630', '012600', '003620', '011300', '011690', '096760', '234080', '002420']
11
12 def fs_data(code):
13     url = f'http://comp.fnguide.com/SV02/ASP/SVD_FinanceRatio.asp?pGB=1&gicode=A{code}'
14     res = requests.get(url)
15     df = pd.read_html(res.text)
16     temp_df = df[0]
17     temp_df = temp_df.set_index(temp_df.columns[0])
18     temp_df = temp_df[temp_df.columns[:12]]
19     temp_df = temp_df.loc[['유동비율계산에 참여한 계정 펼치기', '당좌비율계산에 참여한 계정 펼치기', '부채비율계산에 참여한 계정 펼치기', '유보율계산에 참여한 계정 펼치기',
20                          '순차입금비율계산에 참여한 계정 펼치기', '이자보상배율계산에 참여한 계정 펼치기', '매출액증가율계산에 참여한 계정 펼치기', '판매비와관리비증가율계산에 참여한 계정 펼치기',
21                          'EBITDA증가율계산에 참여한 계정 펼치기', '매출총이익율계산에 참여한 계정 펼치기', '영업이익률계산에 참여한 계정 펼치기']]
22     return temp_df
```

def fs\_data(code) 함수 지정



# (1) DATA 생성 - 관리종목

```
24 dataset_all = pd.DataFrame()
25 for code in codes:
26     dataframe = fs_data(code)
27     dataframe = dataframe.T
28     dataframe = dataframe.reset_index(drop=True)
29     dataframe = dataframe.values
30     datalist = []
31     datalist.append(dataframe)
32     data_np = np.array(datalist)
33     dataset = data_np.reshape(1,55)
34     dataset = pd.DataFrame(dataset)
35     dataset.columns = ['CR1','QR1','DR1','RR1','NDR1','ICR1','SGR1','SAEGR1','EBITDA1','GPM1','OPP1',
36                       'CR2','QR2','DR2','RR2','NDR2','ICR2','SGR2','SAEGR2','EBITDA2','GPM2','OPP2',
37                       'CR3','QR3','DR3','RR3','NDR3','ICR3','SGR3','SAEGR3','EBITDA3','GPM3','OPP3',
38                       'CR4','QR4','DR4','RR4','NDR4','ICR4','SGR4','SAEGR4','EBITDA4','GPM4','OPP4',
39                       'CR5','QR5','DR5','RR5','NDR5','ICR5','SGR5','SAEGR5','EBITDA5','GPM5','OPP5']
40     dataset_all = dataset_all.append(dataset)
41
42 dataset_all["Target"] = 1
43 print(dataset_all)
```

## Column명 변경

유동비율 : CR  
당좌비율 : QR  
부채비율 : DR  
유보율 : RR  
순차입금비율 : NDR  
이자보상배율 : ICR  
매출액증가율 : SGR  
판매비와관리비증가율 : SAEGR  
EBITDA증가율 : EBITDA  
매출총이익율 : GPM  
영업이익률 : OPP

for code in codes 반복문 선언



## (2) 결측치 처리



	CR1	QR1	DR1	RR1	NDR1	ICR1	SGR1	SAEGR1	EBITDA1	GPM1	OPP1	CR2	QR2	...	OPP4	CR5	QR5	DR5	RR5	NDR5	ICR5	SGR5	SAEGR5	EBITDA5	GPM5	OPP5	Target
0	448.5	389.1	15.1	712.8	NaN	16.7	9.8	7.8	16.3	11.7	1.4	335.3	271.7	...	-2.5	229.8	183.2	25.8	671.8	NaN	6.9	15.1	5.4	52.7	11.6	1.4	1
0	81.4	61.6	207.7	273.8	78.4	2.3	17.8	23.0	16.3	7.2	0.9	87.0	60.0	...	1.2	80.0	67.9	305.0	261.8	153.5	1.3	3.1	7.6	0.8	10.3	1.2	1
0	154.6	57.8	149.3	465.3	84.1	0.1	0.0	0.8	-57.4	29.9	0.1	134.8	48.5	...	0.0	187.4	82.8	142.0	399.4	63.3	3.6	18.4	12.1	347.5	21.6	2.4	1
0	204.3	122.7	60.1	380.7	13.1	9.5	9.8	4.9	-11.3	9.3	4.2	249.4	132.6	...	4.9	281.9	187.5	55.6	534.1	NaN	NaN	25.5	19.2	232.2	15.9	10.8	1
0	36.8	23.7	5842.4	NaN	3726.7	NaN	11.5	17.1	-61.7	6.4	-0.5	86.9	54.3	...	4.7	139.7	61.5	142.1	102.0	88.8	8.7	37.1	-4.2	173.7	13.3	10.3	1
0	52.7	35.0	172.2	NaN	96.2	NaN	-6.6	-6.0	NaN	-0.7	-4.0	72.0	39.8	...	-5.6	102.4	78.0	35.8	29.1	NaN	NaN	6.3	-6.1	NaN	-2.7	-12.2	1
0	406.4	289.7	20.0	209.8	NaN	51.8	-17.0	-11.8	-3.8	11.6	5.5	336.8	264.2	...	6.6	4459.4	3845.0	2.0	209.5	NaN	315418.7	31.5	54.4	17.8	13.8	5.1	1
0	163.5	94.8	77.1	221.7	16.6	NaN	-8.2	17.3	NaN	-0.8	-29.4	155.6	70.3	...	-71.2	124.5	82.3	259.7	NaN	74.3	NaN	-5.4	-13.4	NaN	0.4	-28.2	1
0	96.1	50.0	97.9	186.5	36.1	NaN	4.6	5.2	NaN	-5.0	-12.6	77.4	37.8	...	-20.6	85.8	69.2	95.2	NaN	32.6	NaN	-0.2	-9.9	NaN	4.1	-11.5	1
0	83.0	74.2	133.9	241.8	56.6	NaN	-4.3	38.7	NaN	0.9	-8.8	70.4	43.9	...	-100.3	88.2	87.1	NaN	NaN	NaN	NaN	-4.6	-24.1	NaN	0.8	-55.0	1
0	106.0	71.4	137.8	NaN	33.9	NaN	-76.5	-44.0	NaN	8.7	-30.9	216.8	149.7	...	4.4	258.1	184.1	37.3	NaN	1.7	1.3	16.7	9.7	41.7	35.7	1.0	1
0	46.4	19.8	170.0	NaN	106.1	NaN	-39.9	-24.1	NaN	14.3	-16.2	89.9	68.9	...	-20.7	119.7	60.3	40.5	245.4	10.4	NaN	-49.4	23.6	NaN	21.6	-41.9	1
0	65.1	42.4	190.0	12.4	2.4	NaN	-3.7	3.1	-38.2	14.8	-1.9	57.4	34.4	...	-15.2	29.7	18.6	NaN	NaN	NaN	NaN	-13.8	-20.8	NaN	1.7	-13.4	1
0	93.2	44.3	172.2	209.8	99.1	NaN	3.8	9.7	40.7	6.8	-2.4	103.8	51.4	...	-13.6	82.3	42.2	174.0	108.2	93.1	NaN	-36.2	-8.8	NaN	-4.1	-22.6	1
0	119.7	90.2	60.0	382.5	16.4	NaN	14.5	-1.3	NaN	5.9	-2.9	98.1	78.0	...	-3.9	339.7	252.4	29.4	26.8	NaN	NaN	-5.3	1.0	NaN	5.6	-4.3	1
0	109.5	82.8	263.5	437.7	181.4	1.3	-0.5	6.2	12.5	43.3	6.1	128.4	94.7	...	4.4	88.1	59.2	225.5	327.1	111.7	3.7	-3.9	2.5	26.9	43.9	7.5	1
0	74.0	57.7	94.5	157.9	58.8	10.1	8.6	6.0	0.3	25.9	16.6	79.2	59.2	...	21.5	84.7	57.6	66.4	251.9	31.5	23.4	-9.7	-0.1	-17.0	29.4	17.4	1
0	15.2	15.0	81.8	1061.9	44.4	NaN	3.0	-20.4	NaN	6.5	-6.6	32.1	31.9	...	-53.8	26.7	18.8	142.2	843.1	96.5	NaN	505.4	93.4	NaN	-2.2	-10.2	1

[18 rows x 56 columns]

NaN값이  
'0'으로  
대체

	CR1	QR1	DR1	RR1	NDR1	ICR1	SGR1	SAEGR1	EBITDA1	GPM1	OPP1	CR2	QR2	...	OPP4	CR5	QR5	DR5	RR5	NDR5	ICR5	SGR5	SAEGR5	EBITDA5	GPM5	OPP5	Target
0	448.5	389.1	15.1	712.8	0	16.7	9.8	7.8	16.3	11.7	1.4	335.3	271.7	...	-2.5	229.8	183.2	25.8	671.8	0	6.9	15.1	5.4	52.7	11.6	1.4	1
0	81.4	61.6	207.7	273.8	78.4	2.3	17.8	23.0	16.3	7.2	0.9	87.0	60.0	...	1.2	80.0	67.9	305.0	261.8	153.5	1.3	3.1	7.6	0.8	10.3	1.2	1
0	154.6	57.8	149.3	465.3	84.1	0.1	0.0	0.8	-57.4	29.9	0.1	134.8	48.5	...	0.0	187.4	82.8	142.0	399.4	63.3	3.6	18.4	12.1	347.5	21.6	2.4	1
0	204.3	122.7	60.1	380.7	13.1	9.5	9.8	4.9	-11.3	9.3	4.2	249.4	132.6	...	4.9	281.9	187.5	55.6	534.1	0	0	25.5	19.2	232.2	15.9	10.8	1
0	36.8	23.7	5842.4	0	3726.7	0	11.5	17.1	-61.7	6.4	-0.5	86.9	54.3	...	4.7	139.7	61.5	142.1	102.0	88.8	8.7	37.1	-4.2	173.7	13.3	10.3	1
0	52.7	35.0	172.2	0	96.2	0	-6.6	-6.0	0	-0.7	-4.0	72.0	39.8	...	-5.6	102.4	78.0	35.8	29.1	0	0	6.3	-6.1	0	-2.7	-12.2	1
0	406.4	289.7	20.0	209.8	0	51.8	-17.0	-11.8	-3.8	11.6	5.5	336.8	264.2	...	6.6	4459.4	3845.0	2.0	209.5	0	315418.7	31.5	54.4	17.8	13.8	5.1	1
0	163.5	94.8	77.1	221.7	16.6	0	-8.2	17.3	0	-0.8	-29.4	155.6	70.3	...	-71.2	124.5	82.3	259.7	0	74.3	0	-5.4	-13.4	0	0.4	-28.2	1
0	96.1	50.0	97.9	186.5	36.1	0	4.6	5.2	0	-5.0	-12.6	77.4	37.8	...	-20.6	85.8	69.2	95.2	0	32.6	0	-0.2	-9.9	0	4.1	-11.5	1
0	83.0	74.2	133.9	241.8	56.6	0	-4.3	38.7	0	0.9	-8.8	70.4	43.9	...	-100.3	88.2	87.1	0	0	0	0	-4.6	-24.1	0	0.8	-55.0	1
0	106.0	71.4	137.8	0	33.9	0	-76.5	-44.0	0	8.7	-30.9	216.8	149.7	...	4.4	258.1	184.1	37.3	0	1.7	1.3	16.7	9.7	41.7	35.7	1.0	1
0	46.4	19.8	170.0	0	106.1	0	-39.9	-24.1	0	14.3	-16.2	89.9	68.9	...	-20.7	119.7	60.3	40.5	245.4	10.4	0	-49.4	23.6	0	21.6	-41.9	1
0	65.1	42.4	190.0	12.4	2.4	0	-3.7	3.1	-38.2	14.8	-1.9	57.4	34.4	...	-15.2	29.7	18.6	0	0	0	0	-13.8	-20.8	0	1.7	-13.4	1
0	93.2	44.3	172.2	209.8	99.1	0	3.8	9.7	40.7	6.8	-2.4	103.8	51.4	...	-13.6	82.3	42.2	174.0	108.2	93.1	0	-36.2	-8.8	0	-4.1	-22.6	1
0	119.7	90.2	60.0	382.5	16.4	0	14.5	-1.3	0	5.9	-2.9	98.1	78.0	...	-3.9	339.7	252.4	29.4	26.8	0	0	-5.3	1.0	0	5.6	-4.3	1
0	109.5	82.8	263.5	437.7	181.4	1.3	-0.5	6.2	12.5	43.3	6.1	128.4	94.7	...	4.4	88.1	59.2	225.5	327.1	111.7	3.7	-3.9	2.5	26.9	43.9	7.5	1
0	74.0	57.7	94.5	157.9	58.8	10.1	8.6	6.0	0.3	25.9	16.6	79.2	59.2	...	21.5	84.7	57.6	66.4	251.9	31.5	23.4	-9.7	-0.1	-17.0	29.4	17.4	1
0	15.2	15.0	81.8	1061.9	44.4	0	3.0	-20.4	0	6.5	-6.6	32.1	31.9	...	-53.8	26.7	18.8	142.2	843.1	96.5	0	505.4	93.4	0	-2.2	-10.2	1

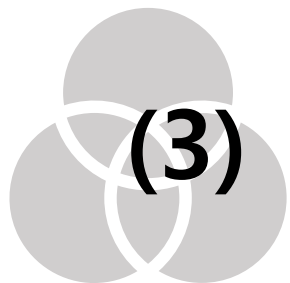


## (2) 결측치 처리

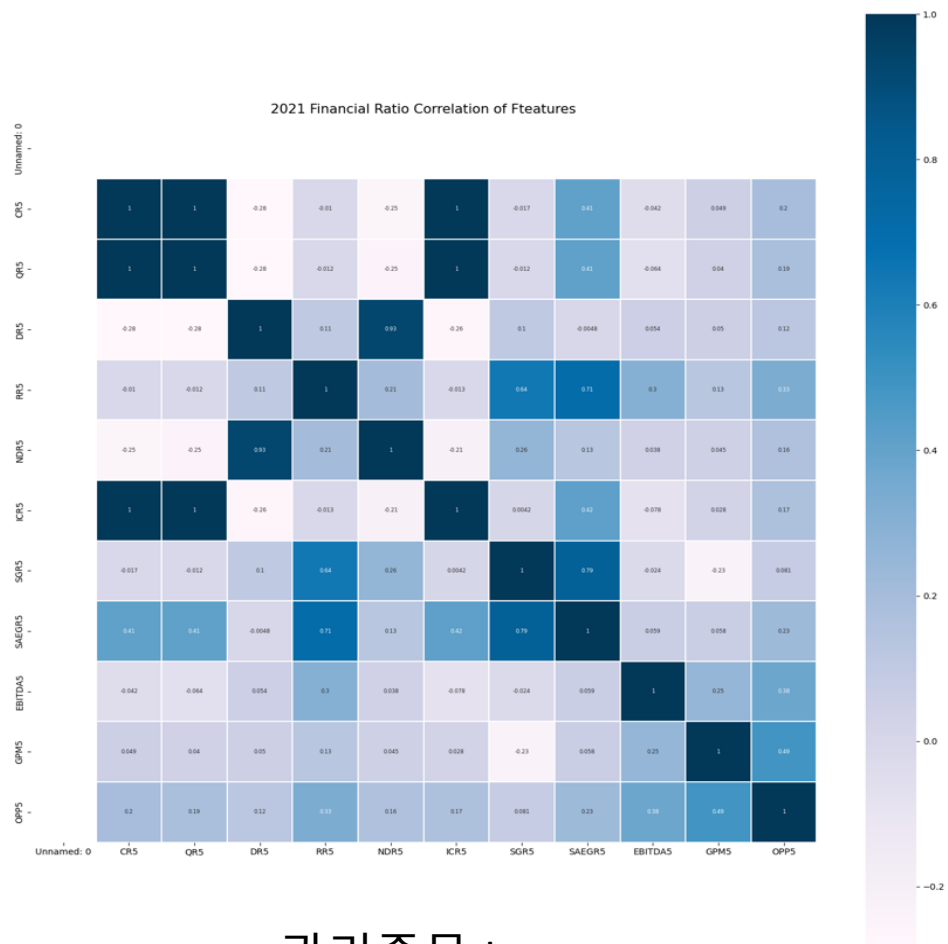


```
24 dataset_all = pd.DataFrame()
25 for code in codes:
26     dataframe = fs_data(code)
27     dataframe = dataframe.T
28     dataframe = dataframe.reset_index(drop=True)
29     dataframe = dataframe.values
30     datalist = []
31     datalist.append(dataframe)
32     data_np = np.array(datalist)
33     dataset = data_np.reshape(1,55)
34     dataset = pd.DataFrame(dataset)
35     dataset.columns = ['CR1', 'QR1', 'DR1', 'RR1', 'NDR1', 'ICR1', 'SGR1', 'SAEGR1', 'EBITDA1', 'GPM1', 'OPP1',
36                       'CR2', 'QR2', 'DR2', 'RR2', 'NDR2', 'ICR2', 'SGR2', 'SAEGR2', 'EBITDA2', 'GPM2', 'OPP2',
37                       'CR3',
38                       'CR4',
39                       'CR5']
40     dataset_all = dataset_all.append(dataset)
41
42 dataset_all = dataset_all.where(pd.notnull(dataset_all), '0')
43 dataset_all["Target"] = 1
44 print(dataset_all)
45
46 dataset_all.to_csv('관리종목data.csv', index=True, encoding='utf-8-sig')
```

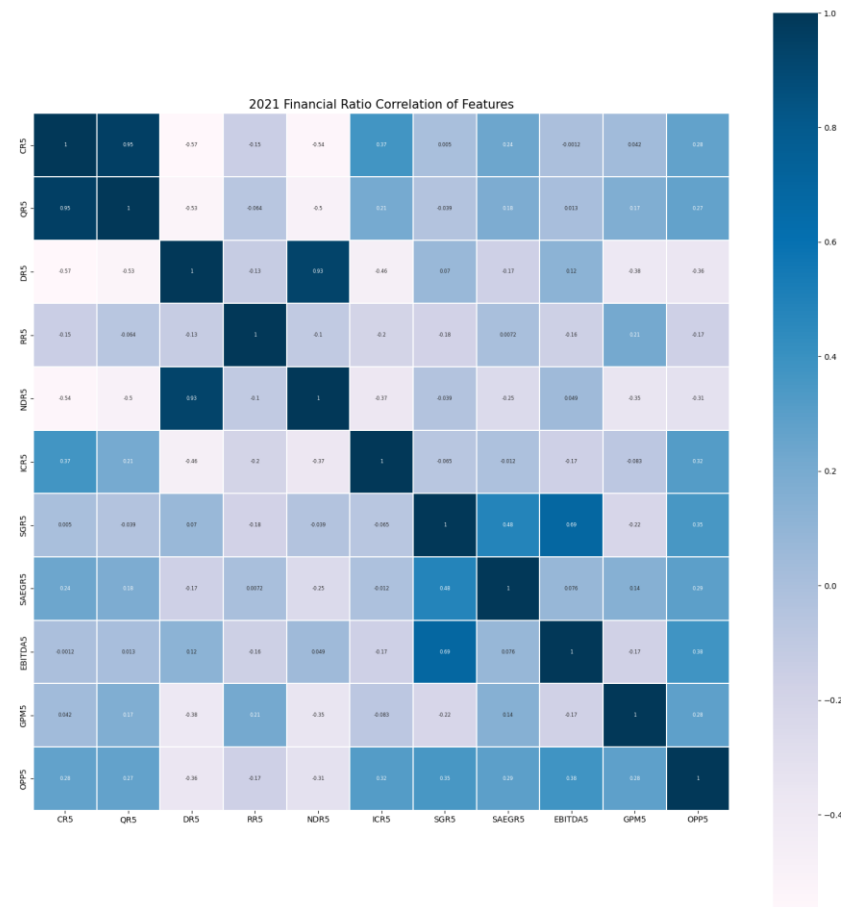
```
dataset_all = dataset_all.where(pd.notnull(dataset_all), '0')
```



### (3) Feature간 상관관계 분석



관리종목data



안전종목data



# DATA

```
data1 = pd.read_csv("./관리종목data.csv")
data2 = pd.read_csv("./안전종목data.csv")

dataset = pd.concat([data1,data2],ignore_index=True)
del dataset['Unnamed: 0']
```

	A	B	C	D	E	BA	BB	BC	BD	BE
1		CR1	QR1	DR1	RR1	SAEGR5	EBITDA5	GPM5	OPP5	Target
2	0	448.5	389.1	15.1	712.8	5.4	52.7	11.6	1.4	1
3	0	81.4	61.6	207.7	273.8	7.6	0.8	10.3	1.2	1
4	0	154.6	57.8	149.3	465.3	12.1	347.5	21.6	2.4	1
5	0	204.3	122.7	60.1	380.7	19.2	232.2	15.9	10.8	1
6	0	36.8	23.7	5842.4	0	-4.2	173.7	13.3	10.3	1
7	0	52.7	35	172.2	0	-6.1	0	-2.7	-12.2	1
8	0	406.4	289.7	20	209.8	54.4	17.8	13.8	5.1	1
9	0	163.5	94.8	77.1	221.7	-13.4	0	0.4	-28.2	1
10	0	96.1	50	97.9	186.5	-9.9	0	4.1	-11.5	1
11	0	83	74.2	133.9	241.8	-24.1	0	0.8	-55	1
12	0	106	71.4	137.8	0	9.7	41.7	35.7	1	1
13	0	46.4	19.8	170	0	23.6	0	21.6	-41.9	1
14	0	65.1	42.4	190	12.4	-20.8	0	1.7	-13.4	1
15	0	93.2	44.3	172.2	209.8	-8.8	0	-4.1	-22.6	1
16	0	119.7	90.2	60	382.5	1	0	5.6	-4.3	1
17	0	109.5	82.8	263.5	437.7	2.5	26.9	43.9	7.5	1
18	0	74	57.7	94.5	157.9	-0.1	-17	29.4	17.4	1
19	0	15.2	15	81.8	1061.9	93.4	0	-2.2	-10.2	1

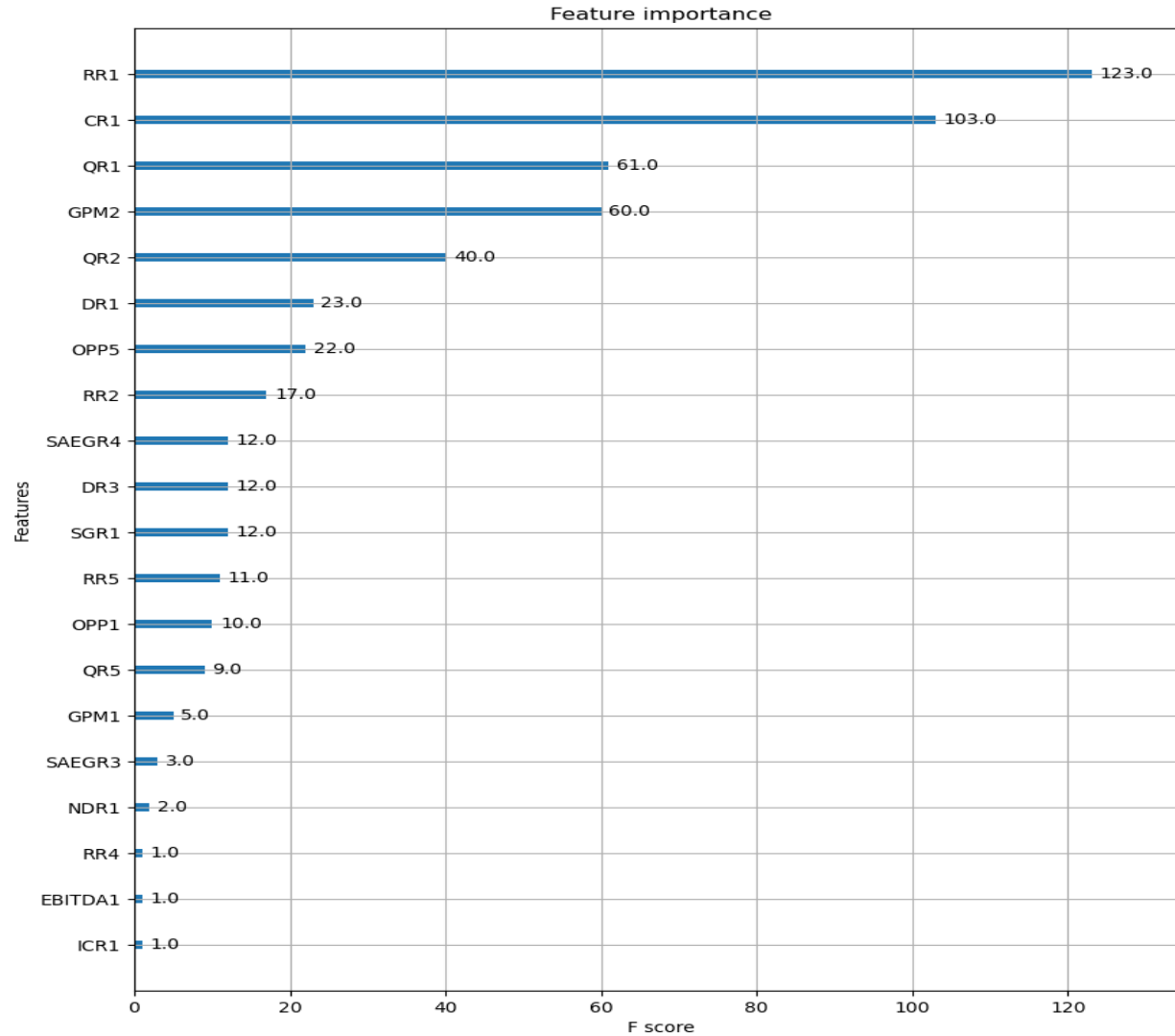
“ 관리종목 data ”

concat  
+

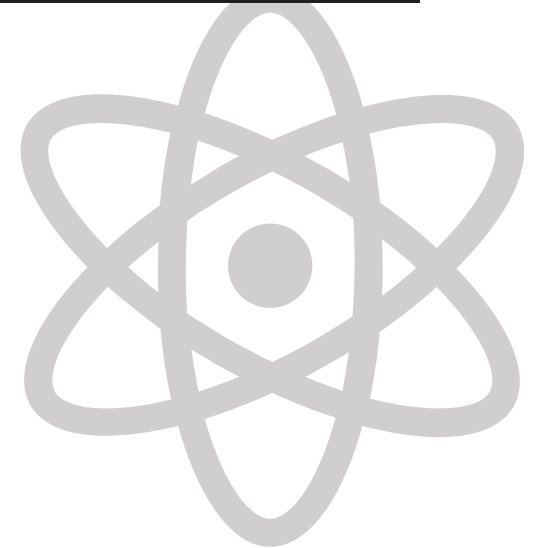
	A	B	C	D	E	BA	BB	BC	BD	BE
1		CR1	QR1	DR1	RR1	SAEGR5	EBITDA5	GPM5	OPP5	Target
2	0	218.8	181.6	40.7	23681.4	5.2	26.3	40.2	18.6	0
3	0	213.3	180.8	34.3	845.6	8.7	42.1	42.1	26.8	0
4	0	27.2	18	80.6	2304	57.1	72.3	47.5	36.4	0
5	0	207.8	206.2	51.2	35907.9	35	9.3	100	19.9	0
6	0	168.6	118.2	53.3	4121.6	70.8	88.6	28.2	13.5	0
7	0	112.2	88.4	138.4	4651.1	-9.3	98.7	18.5	6	0
8	0	191.5	189.4	42.3	11751.3	47.7	49.7	100	11.2	0
9	0	134.6	98.3	37.5	3152.6	14.8	41.6	23	8.2	0
26	0	149.1	87.7	120.5	2247.9	-2.1	0	11.3	9.1	0
27	0	362.6	235	27.5	751.4	5.2	-5	53.9	25.7	0
28	0	78.6	63.8	280.2	481.8	-21.6	222.5	16.7	8.2	0
29	0	294.7	232.7	28.2	2082	40.4	92.5	12.8	7	0
30	0	54	43.7	557.1	656.5	-4.4	36	19.4	11.7	0
31	0	582.2	581.8	19.8	50821.9	-0.8	-33.7	100	5.5	0
32	0	605.1	381.6	15.3	6274	13.8	21	13.2	11.6	0
33	0	160.3	124.5	28.8	11941.1	1.9	108.8	72.2	9	0
34	0	116.7	90.5	94.6	703.4	13.7	118.2	18.9	8.3	0
35	0	109.5	83.8	201.3	1548.8	33.5	69.8	14.9	9.1	0
36	0	217	176.5	59.5	6604.3	13	178.9	16.6	11.6	0
37	0	102.1	97.3	125.1	710	1.7	6.6	100	7.1	0

“ 안전종목 data ”

# XGBoost



유동비율 : CR  
당좌비율 : QR  
부채비율 : DR  
유보율 : RR  
순차입금비율 : NDR  
이자보상배율 : ICR  
매출액증가율 : SGR  
판매비와관리비증가율 : SAEGR  
EBITDA증가율 : EBITDA  
매출총이익율 : GPM  
영업이익률 : OPP



# 모델 예측 및 결과비교

```
1 import requests
2 import numpy as np
3 import pandas as pd
4 from pandas.core.frame import DataFrame
5
6 # 아센디오(012170)
7 # 신풍제약(019170), 대한전선(001440)
8 pre_code = ['012170', '019170', '001440']
9
10 def fs_data(code):
11     url = f'http://comp.fnguide.com/SV02/AS
12     res = requests.get(url)
13     df = pd.read_html(res.text)
14     temp_df = df[0]
15     temp_df = temp_df.set_index(temp_df.co
16     temp_df = temp_df[temp_df.columns[:12]]
17     temp_df = temp_df.loc[['유동비율계산에
18     '순차입금비율계산
19     '판매비와관리비증
20     '영업이익률계산에
21     return temp_df
```

위험종목으로 편입될 가능성이 높은

“아센디오”

안전한 투자종목일 가능성 높은

“신풍제약, 대한전선”

# 1. LSTM 모델구성

```
73 model = Sequ
74 model.add(LS
75 model.add(De
76 model.add(Dr
77 model.add(De
78 model.add(De
79 model.add(Dr
80 model.add(De
81 model.add(Dr
82 model.add(De
83 model.add(De
```

```
Epoch 00120: early
1/1 [=====
loss : 0.31515434
accuracy : 0.9285
[69.83464]
[4.8694215]
[3.148824]
```

이 종목은 위험률 [69.83] % 으로 관리종목으로 지정될 가능성이 있는 위험한 종목입니다.  
이 종목은 투자해도 좋습니다. 위험률 [4.87] %  
이 종목은 투자해도 좋습니다. 위험률 [3.15] %

```
pre_data = np.array(pre_data)
pre_data = pre_data.reshape(len(pre_data),11,5)
result = model.predict(pre_data)
result = result*100
print(result[0])
print(result[1])
print(result[2])

# {'safe': 0, 'danger': 1}
for i in result:
    if( i <=50):
        safe = i
        print(f"이 종목은 투자해도 좋습니다. 위험률 {np.round(safe,2)} % ")
    elif( i >50):
        danger = i
        print(f"이 종목은 위험률 {np.round(danger,2)} % 으로 관리종목으로 지정될 가능성이 있는 위험한 종목입니다.")
    else:
        print("ERROR")
```

이엔터코(012170)  
신풍제약(019180)  
대한전선(001440)

## 2. GRU 모델구성

```
73 model = Sequential()
74 model.add(GRU(32, activation='tanh', input_shape=(11,5)))
75 model.add(Dropout(0.2))
76 model.add(Dense(48, activation='tanh'))
77 model.add(Dropout(0.2))
78 model.add(Dense(32, activation='relu'))
79 model.add(Dropout(0.2))
80 model.add(Dense(16))
81 model.add(Dropout(0.1))
82 model.add(Dense(8, activation='relu'))
83 model.add(Dropout(0.1))
84 model.add(Dense(4))
85 model.add(Dropout(0.1))
86 model.add(Dense(1, activation='sigmoid'))
```

StandardScalar

```
Epoch 00107: early stopping
1/1 [=====] - 0s 16ms/step - loss: 0.3110 - accuracy: 0.9286
loss : 0.3109632134437561
accuracy : 0.9285714030265808
[44.382957]
[1.321261]
[0.37930158]
이 종목은 투자해도 좋습니다. 위험률 [44.38] %
이 종목은 투자해도 좋습니다. 위험률 [1.32] %
이 종목은 투자해도 좋습니다. 위험률 [0.38] %
```

아센디오(012170)  
신풍제약(019180)  
대한전선(001440)

### 3. DNN 모델구성

```
73 model = Sequential()
74 model.add(Dense(32, activation='tanh', input_dim=55))
75 model.add(Dropout(0.2))
76 model.add(Dense(48))
77 model.add(Dropout(0.2))
78 model.add(Dense(32, activation='relu'))
79 model.add(Dense(16))
80 model.add(Dropout(0.2))
81 model.add(Dense(8, activation='relu'))
82 model.add(Dropout(0.2))
83 model.add(Dense(4))
84 model.add(Dense(1, activation='sigmoid'))
```

```
Epoch 00468: early stopping
1/1 [=====] - 0s 18ms/step - loss: 2.2735 - accuracy: 0.9286
loss : 2.2735307216644287
accuracy : 0.9285714030265808
[8.997829e-28]
[3.169619e-33]
[4.170392e-11]
이 종목은 투자해도 좋습니다. 위험률 [0.] %
이 종목은 투자해도 좋습니다. 위험률 [0.] %
이 종목은 투자해도 좋습니다. 위험률 [0.] %
```

StandardScaler

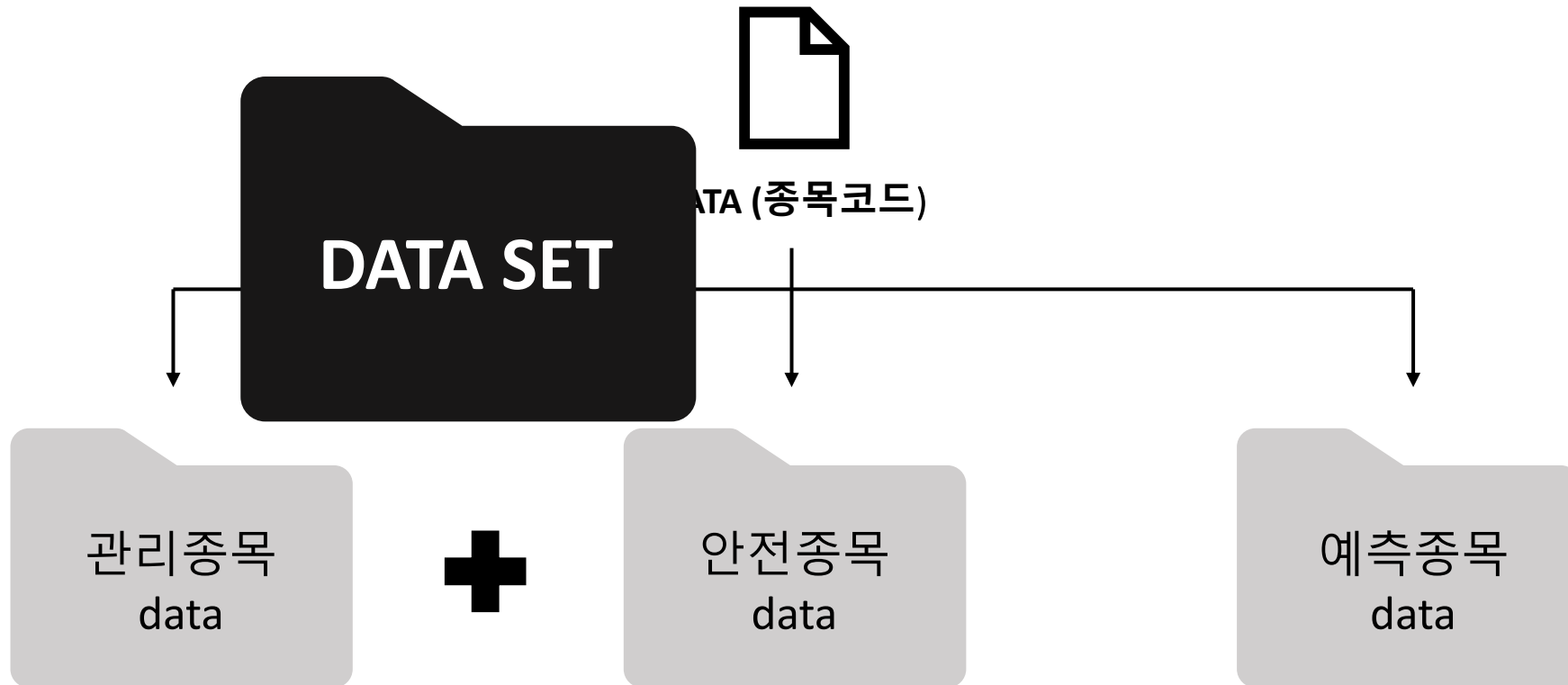
아센디오(012170)  
신풍제약(019180)  
대한전선(001440)



# Heymoon

LSTM 기반 인공지능 투자 FRIEND

# Heymoon Tool





# “Heymoon” 성능 테스트

## 위험종목 예상

- 아센디오(012170)  
2021.03 관리종목에서 해지

## 안전종목 예상

- 신풍제약(019170)
- 대한전선(001440)  
시총 상위 150위 내 종목

```
# 아센디오(012170)  
# 신풍제약(019170), 대한전선(001440)  
pre_code = ['012170', '019170', '001440']
```

```
Epoch 00120: early stopping  
1/1 [=====] - 0s 18ms/step - loss: 0.3152 - accuracy: 0.9286  
loss : 0.3151543438434601  
accuracy : 0.9285714030265808  
[69.83464]  
[4.8694215]  
[3.148824]  
이 종목은 위험률 [69.83] % 으로 관리종목으로 지정될 가능성이 있는 위험한 종목입니다.  
이 종목은 투자해도 좋습니다. 위험률 [4.87] %  
이 종목은 투자해도 좋습니다. 위험률 [3.15] %
```

Wow!!

