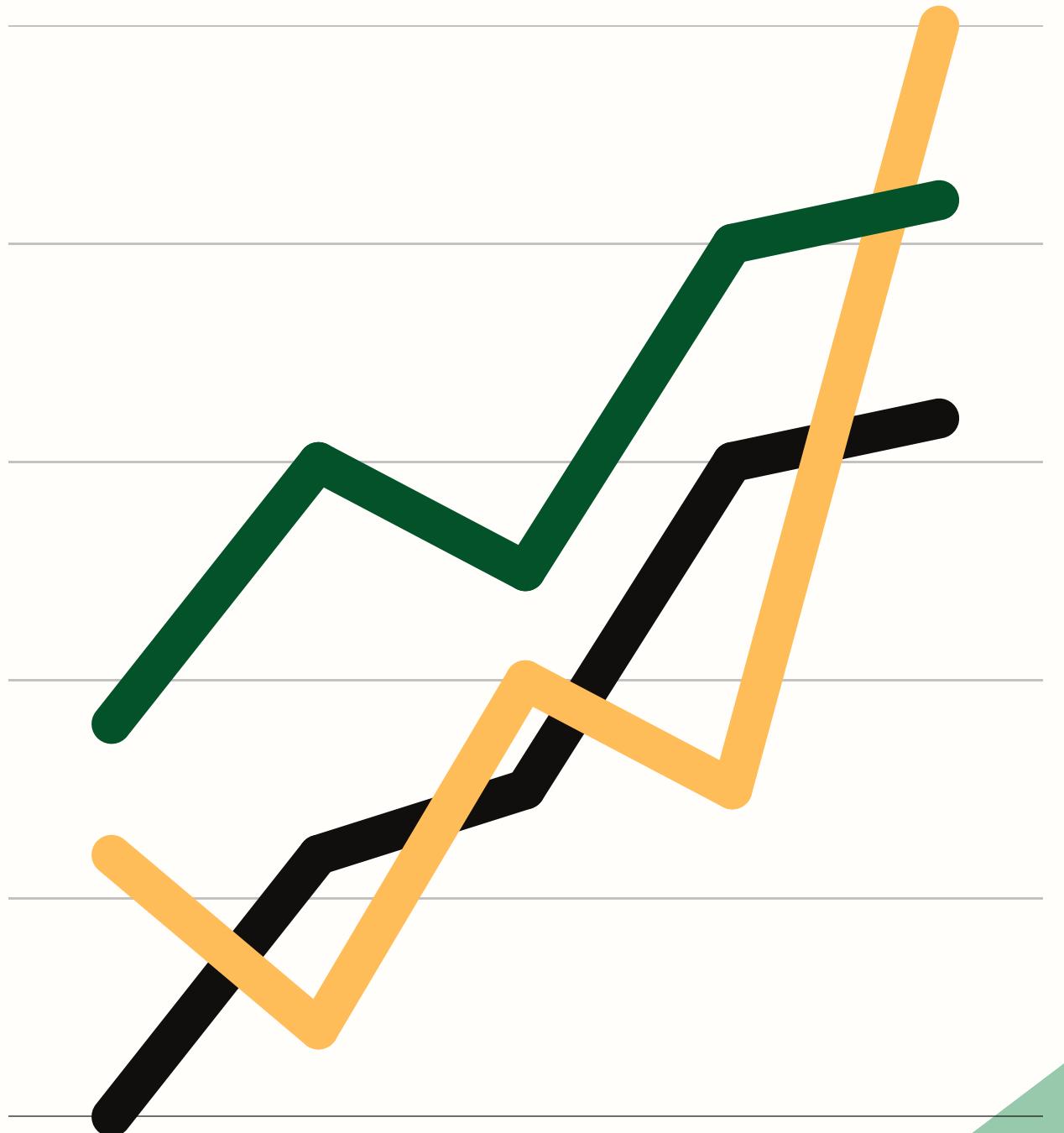


# 머신러닝을 활용한 고객성향별 주식 포트폴리오 구축

“Creating Customer-Specific Stock Portfolios  
Using Machine Learning”

#TeamBR : Beat the Recession



# Contents

- 01 Introduction
- 02 EDA / Data Preprocessing
- 03 Modeling / Portfolio 구성
- 04 Insight / Contribution / 한계

# Introduction

1. TeamBR
2. Topic
3. Workflow
4. Timetable

유가증권시장본부  
Stock Market  
코스닥시장본부  
KOSDAQ Market



# Introduction: TeamBR

## TeamBR: Beat the Recession의 조원을 소개합니다

### 서선우

- Main Coder
- 기획

### 안종진

- 회계/금융아이디어제시
- 발표자료준비

### 임대혁

- 자료조사
- 기술적 아이디어제시

### 박영선

- Main Domain: 금융 아이디어제시
- Coder

## 주제

**재무제표 항목을 이용해 만든 Feature를 머신러닝  
에 적용하여 고객성향별 주식 포트폴리오 구축**

## 부연설명

**1년 동안 재무제표를 관측하고 다음 1년 동안 보유하는 접근법**

# Introduction: Topic

What is the topic?

## 주제 선정배경



### 주제 선정배경1

주식에 대한 높은 관심:

- 2020년 1월 기준 경제경영 부문 베스트셀러 TOP 10의 키워드:
  - 트랜드, 자기계발, 성공, 부자가 되는 법 등
- 2021~2023년의 경제경영 부문 TOP10:
  - 제무제표 모르면 주식투자 절대로 하지마라, 구루들의 투자법, 현명한 투자자, 한국형 가치투자 등
  - 특히 기본적 분석에 관한 서적들이 TOP 10 안에 포함됨

# Introduction: Topic

What is the topic?

## 주제 선정배경

재무상태표		
제 48 기 제 47 기말 제 46 기말	1분기말 2016.03.31 현재 2015.12.31 현재 2014.12.31 현재	
제 48 기 1분기말	제 47 기말	(단위 : 백만원)
67,079,811	67,002,055	62,054,773
6,067,702	3,062,960	1,643,318
25,163,554	27,763,589	26,454,093
2,364,249	3,021,210	2,137,533
21,022,927	20,251,464	18,940,786
1,309,355	2,314,823	3,218,973
952,281	1,105,216	977,114
2,479,717	1,980,305	1,846,360
7,128,481	6,578,112	5,553,834
591,545	847,303	821,079
98,829,335	77,073	461,683
2,904,160	101,967,575	102,005,810
42,720,014	3,205,283	7,106,234
44,258,354	44,107,398	41,857,431
3,443,079	45,148,629	43,744,259
4,286,995	3,407,229	3,051,564
128,452	3,845,119	4,415,935
1,088,281	56,174	1
165,909,1	865,903	

## 주제 선정배경2

대부분의 주식 관련 논문들은 거래량을 기반으로 예측을 수행하는 경우가 많았고 재무제표 항목을 Feature로 이용한 경우는 적었음. 또한 PEG와 같은 투자 대가들이 이용한 재무 비율을 Feature로 사용한 경우는 거의 없었음

## 주제 선정배경3

머신러닝 사용 시 상장 기업의 재무 지표를 일일히 분석해야 되는 시간과 수고를 줄이는 필터링 효과가 기대됨

## 위험선호도

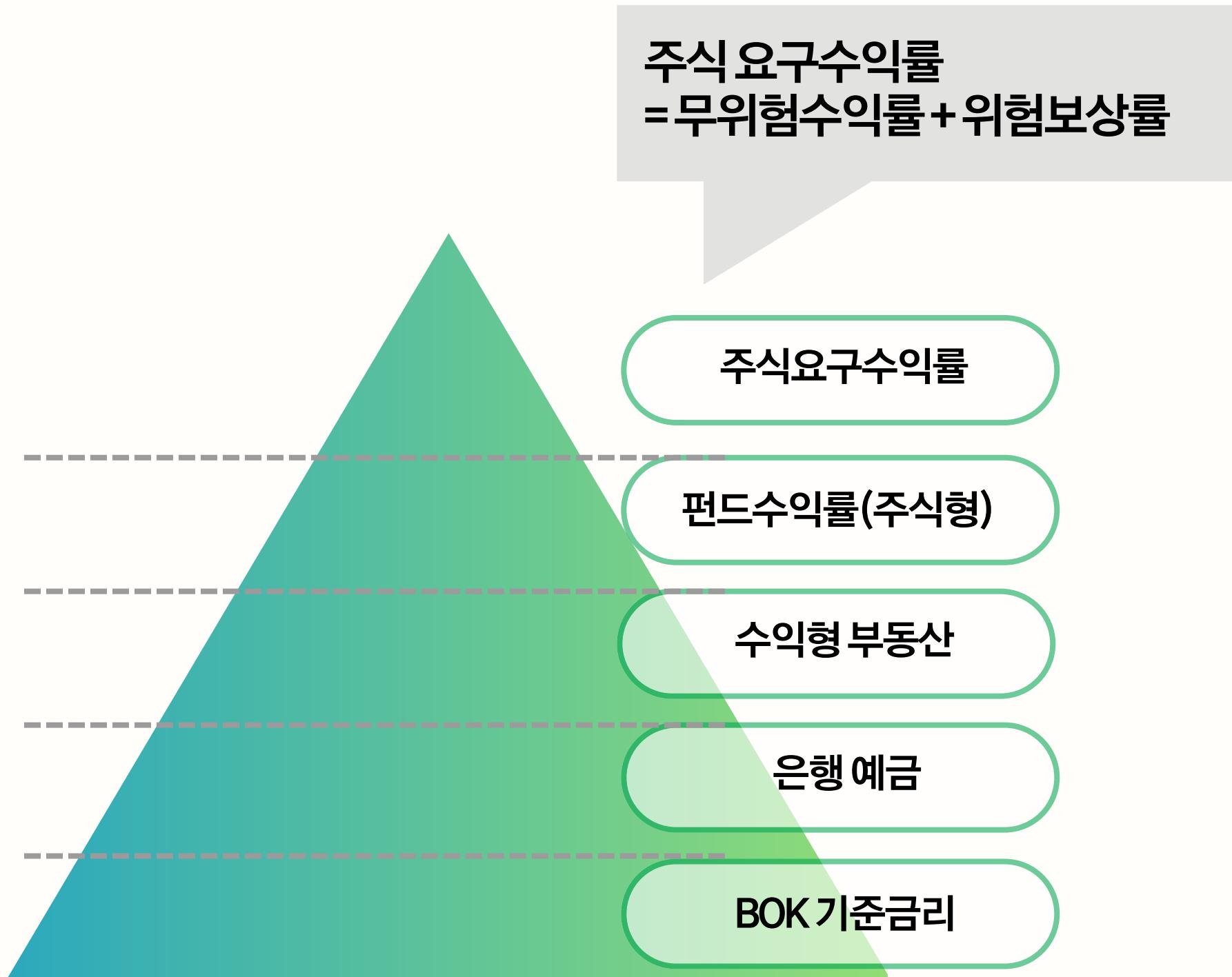
적극투자형에 해당하는 주식투자자들의 포트폴리오에 포함될 종목을 머신러닝을 이용하여 발굴하기



- 주식은 고위험자산으로 분류되고 고위험자산에 80% 이상 투자하는 경우 2등급 (높은 위험), 고위험자산에 투자하면서 레버리지까지 사용시 1등급 (매우 높은 위험)을 부여함  
(출처 - 금융소비자의 보호에 관한 법률에서 제시한 가이드라인)
- 적극투자형은 투자자금의 상당 부분을 주식, 주식형펀드 등의 위험자산에 투자할 의향이 있음  
(출처 - 기획재정부 KDI 경제정보센터)

# Introduction: Topic

Goal: Outperform the benchmark and achieve the target return



## 무위험수익률

- 본 프로젝트에서는 2011년 ~ 2019년도 시중은행 예금 금리가 1~2%였던 점을 감안하여 1.5%를 무위험수익률로 산정 (출처 - 은행연합회)

## 위험보상률

- 본 프로젝트에서는 위험보상률을 5%로 산정

## 프로젝트 목표

- 1차 목표: 벤치마크인 KOSPI 상회
- 2차 목표: 목표수익률 달성
  - 주식의 요구수익률(6.5%) 상회

## 데이터 수집 및 전처리

2011년~2019년  
9년치 데이터 수집, 전처리 및 EDA

Feature 후보군 96개

상관분석

피처선정

Feature

PER

기업규모코드

PBR

자기자본회전률

PEG

현금흐름스코어

매출액증가율

성장비용

매출액순이익률

영업손익

부채자본비율

Target

1년 수익률

## 분석 및 모델 선정

모델 Dataset

Train/Validation Set

Test Set

2011-2018

2018-2019

Modeling

Random Forest

Logistic Regression

Decision Tree

XGboost

Precision, Accuracy,  
F1-score 등

## 포트폴리오 구축

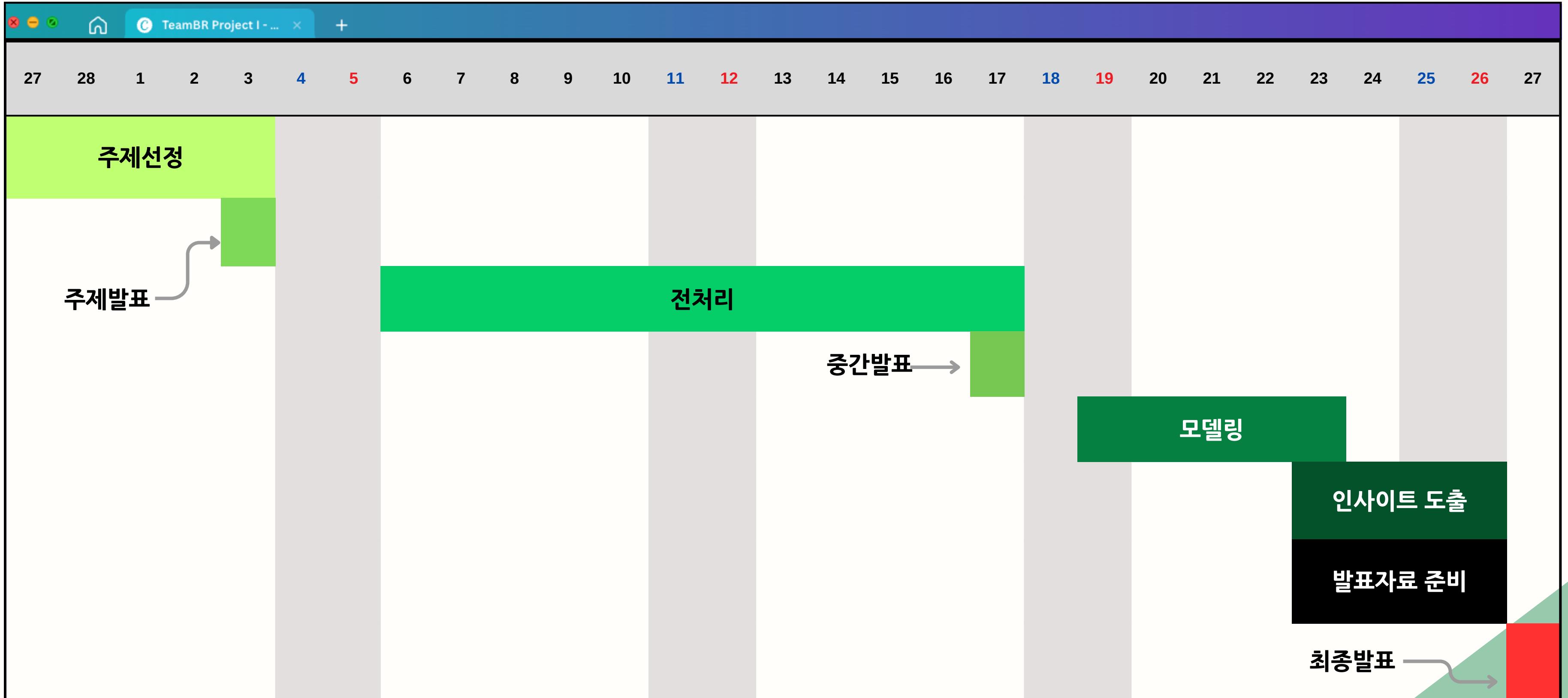
머신러닝을 활용한  
종목 선정 및 포트폴리오 구축

후보 포트폴리오 6개  
벤치마크와 비교 (백테스팅)

최종 포트폴리오 1개 선정

인사이트 도출  
결론

# Introduction: Timetable





# EDA/Preprocessing

1. 자료수집
2. 전처리
3. Feature Selection
4. 파생변수 생성

# EDA:데이터수집

## Data Collection

Data Collecting

		일반사항_기업규모.csv					일반사항_목적사업.csv		일반사항_발행주식총수.csv		일반사항_시장점유율(연결).csv		일반사항_외국인주식.csv	
#	재무데이터.csv													
자본(*)(Ifrs)(천원)		2,727,815,936	#	재무데이터.csv										
부채(*)(Ifrs)(천원)		522,090,155	#	재무데이터.csv										
Abc			#	재무데이터.csv										
일반사항_외국인주식.csv			매출액(영업수익)(*)(연결)(...)											
회계년도			3,057,369,527											
2011/12			3,048,983,304											
			3,162,829,387											
			4,583,363,182											
#	거래데이터.csv		56.9200											
최고가(원)		110,072,149	#	거래데이터.csv										
11,950		10,300	#	거래데이터.csv										
13,900		11,900	#	거래데이터.csv										
16,500		14,950	#	거래데이터.csv										
15,900		14,250	#	거래데이터.csv										
9,700		8,180	#	거래데이터.csv										
9,240		8,620	#	거래데이터.csv										
9,740		9,210	#	거래데이터.csv										
7,760		7,290	#	거래데이터.csv										
900			50.7300											
			16,295,021											

- 출처 : TS2000
- 기간 : 2011년 - 2019년까지의 재무데이터
- 대상기업 : KOSPI+KOSDAQ 전 종목
- 재무, 경영, 트레이딩 관련 다수의 데이터 수집

# 전처리: 데이터수집

## Preprocessing : Data Collection

### 데이터 전처리

- 출처: TS2000
- 기업규모, 외국인의 주식 분포, 주가, 투자, 재무데이터
- CAPEX 데이터프레임을 별도로 생성하여 결합

```
df_foreign.columns = ['회사명', '거래소코드', '회계년도', '기업규모코드', '기업규모명']
df_g = pd.concat([df_size, df_foreign], ignore_index=True)
```

df_size				
회사명	거래소코드	회계년도	기업규모코드	기업규모명
0 (주)BNK금융지주	138930	2011/12	90.0	기타
1 (주)BNK금융지주	138930	2012/12	90.0	기타
2 (주)BNK금융지주	138930	2013/12	90.0	기타
3 (주)BNK금융지주	138930	2014/12	90.0	기타
4 (주)BNK금융지주	138930	2015/12	90.0	기타
...	...	...	...	...
18459	흥아해운(주)	3280	2015/12	30.0
18460	흥아해운(주)	3280	2016/12	30.0
18461	흥아해운(주)	3280	2017/12	30.0
18462	흥아해운(주)	3280	2018/12	30.0
18463	흥아해운(주)	3280	2019/12	30.0
18464 rows × 5 columns				

df_foreign				
회사명	거래소코드	회계년도	외국인_주식수(주)	외국인_주식
0 (주)BNK금융지주	138930	2011/12	110072149.0	
1 (주)BNK금융지주	138930	2012/12	119906064.0	
2 (주)BNK금융지주	138930	2013/12	114353856.0	
3 (주)BNK금융지주	138930	2014/12	122094107.0	
4 (주)BNK금융지주	138930	2015/12	117256395.0	
...	...	...	...	...
18459	흥아해운(주)	3280	2015/12	28729650.0
18460	흥아해운(주)	3280	2016/12	27238474.0
18461	흥아해운(주)	3280	2017/12	33714404.0
18462	흥아해운(주)	3280	2018/12	48009214.0
18463	흥아해운(주)	3280	2019/12	25232008.0
18464 rows × 5 columns				

df_stock						
회사명	거래소코드	회계년도	거래년도(*)	거래월(*)	거래일수	
0 (주)BNK금융지주	138930	2011/12	2011.0	12.0	21.0	
1 (주)BNK금융지주	138930	2012/12	2012.0	12.0	18.0	
2 (주)BNK금융지주	138930	2013/12	2013.0	12.0	20.0	
3 (주)BNK금융지주	138930	2014/12	2014.0	12.0	21.0	
4 (주)BNK금융지주	138930	2015/12	2015.0	12.0	21.0	
...	...	...	...	...	...	...
18459	흥아해운(주)	3280	2015/12	2015.0	12.0	21.0
18460	흥아해운(주)	3280	2016/12	2016.0	12.0	21.0
18461	흥아해운(주)	3280	2017/12	2017.0	12.0	19.0
18462	흥아해운(주)	3280	2018/12	2018.0	12.0	19.0
18463	흥아해운(주)	3280	2019/12	2019.0	12.0	20.0
18464 rows × 10 columns						

회사명	거래소코드	회계년도	(전월) 영업이익(백만원)	(전월) 영업이익률(%)	(전월) 영업이익증감률(%)	(전월) 영업이익증감액(백만원)
0 (주)BNK금융지주	138930	2011/12	3.24995e+09	2.727816e+09	5.220902e+08	1.057370e+09
1 (주)BNK금융지주	138930	2012/12	3.403623e+09	2.740560e+09	6.630553e+08	1.049893e+09
2 (주)BNK금융지주	138930	2013/12	3.507652e+09	2.737860e+09	7.697917e+08	1.162829e+09
3 (주)BNK금융지주	138930	2014/12	4.539500e+09	3.550847e+09	9.886534e+08	1.481363e+09
4 (주)BNK금융지주	138930	2015/12	5.244360e+09	4.086109e+09	1.158250e+09	5.174026e+09
...	...	...	...	...	...	...
18459	흥아해운(주)	3280	2015/12	7.712121e+08	1.618245e+08	6.093875e+08
18460	흥아해운(주)	3280	2016/12	9.425680e+08	1.838293e+08	7.265924e+08
18461	흥아해운(주)	3280	2017/12	1.076150e+09	7.254924e+08	8.364275e+08
18462	흥아해운(주)	3280	2018/12	7.827870e+08	5.673155e+07	7.260555e+08
18463	흥아해운(주)	3280	2019/12	3.865508e+08	1.763223e+07	3.689186e+08
18464 rows × 7 columns						



df_c = pd.read_csv('./CAPEX.csv')	df_c
회사명	거래소코드
0 (주)BNK금융지주	138930
1 (주)BNK금융지주	138930
2 (주)BNK금융지주	138930
3 (주)BNK금융지주	138930
4 (주)BNK금융지주	138930
...	...
18459	흥아해운(주)
18460	흥아해운(주)
18461	흥아해운(주)
18462	흥아해운(주)
18463	흥아해운(주)
18464 rows × 7 columns	

CAPEX

# 전처리: 데이터 수집

## Preprocessing : Data Collection

### 데이터 전처리

- 각 데이터프레임의 결측치와 데이터 유형을 확인
- 결측치는 매출액 등 연결손익계산서 항목에서 많이 발생함

```
df_invest.isnull().sum()
```

회사명	0
거래소코드	0
회계년도	0
[공통]PER(최고)(IFRS)	1762
[공통]PER(최저)(IFRS)	1762
[공통]PBR(최고)(IFRS)	1762
[공통]PBR(최저)(IFRS)	1762
[공통]PCR(최고)(IFRS)	1762
[공통]PCR(최저)(IFRS)	1762
[공통]PSR(최고)(IFRS)	1762
[공통]PSR(최저)(IFRS)	1762
dtype: int64	

```
df_stock.isnull().sum()
```

회사명	0
거래소코드	0
회계년도	0
거래년도(*)	2898
거래월(*)	2898
거래일수	2898
최고가(원)	2898
최저가(원)	2898
종가(원)	2898
거래량(주)	2898
dtype: int64	

```
df_invest.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18464 entries, 0 to 18463
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   회사명            18464 non-null   object 
 1   거래소코드        18464 non-null   int64  
 2   회계년도          18464 non-null   object 
 3   [공통]PER(최고)(IFRS) 16702 non-null   float64
 4   [공통]PER(최저)(IFRS) 16702 non-null   float64
 5   [공통]PBR(최고)(IFRS) 16702 non-null   float64
 6   [공통]PBR(최저)(IFRS) 16702 non-null   float64
 7   [공통]PCR(최고)(IFRS) 16702 non-null   float64
 8   [공통]PCR(최저)(IFRS) 16702 non-null   float64
 9   [공통]PSR(최고)(IFRS) 16702 non-null   float64
 10  [공통]PSR(최저)(IFRS) 16702 non-null   float64
dtypes: float64(8), int64(1), object(2)
memory usage: 1.5+ MB
```

```
df_stock.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18464 entries, 0 to 18463
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   회사명            18464 non-null   object 
 1   거래소코드        18464 non-null   int64  
 2   회계년도          18464 non-null   object 
 3   거래년도(*)       15566 non-null   float64
 4   거래월(*)         15566 non-null   float64
 5   거래일수          15566 non-null   float64
 6   최고가(원)         15566 non-null   float64
 7   최저가(원)         15566 non-null   float64
 8   종가(원)           15566 non-null   float64
 9   거래량(주)         15566 non-null   float64
dtypes: float64(7), int64(1), object(2)
```

# 전처리: 데이터수집

## Preprocessing : Data Collection

### 데이터 전처리

#### 데이터의 특이점 조정

- 거래소코드를 2개 가지고 있는 기업을 찾음
- SK오션플랜트(주)사가 삼강엠앤티라는 회사를 2020년 이후 인수함
- 2011-2019년 기간에는 양사가 각자 회사로서 존재하였기 때문에 분리하여 분석

```
check = df_stock.groupby(['회사명'])['거래소코드'].nunique().reset_index()
check.rename(columns={'거래소코드': '거래소코드_개수'}, inplace=True)
check = check[check['거래소코드_개수'] > 1]
check
```

회사명	거래소코드_개수
에스케이오션플랜트(주)	2

```
check_df = df_stock[df_stock['회사명'].isin(check['회사명'].tolist())]
check_df = check_df.reset_index()
check_df
```

index	회사명	거래소코드	회계년도	거래년도(*)	거래월(*)	거래일수	최고가(원)	최저가(원)	종가(원)	거래량(주)
0	에스케이오션플랜트(주)	100090	2011/12	2011.0	12.0	21.0	11600.0	7680.0	11250.0	1096070.0
1	에스케이오션플랜트(주)	25440	2011/12	2011.0	12.0	21.0	820.0	701.0	779.0	394866.0
2	에스케이오션플랜트(주)	100090	2012/12	2012.0	12.0	18.0	4990.0	4420.0	4950.0	893416.0
3	에스케이오션플랜트(주)	25440	2012/12	2012.0	12.0	18.0	894.0	753.0	793.0	2514085.0
4	에스케이오션플랜트(주)	25440	2013/12	2013.0	12.0	20.0	790.0	648.0	718.0	939171.0
5	에스케이오션플랜트(주)	100090	2013/12	2013.0	12.0	20.0	4265.0	3335.0	3990.0	622946.0
6	에스케이오션플랜트(주)	100090	2014/12	2014.0	12.0	21.0	4080.0	3600.0	3880.0	102085.0
7	에스케이오션플랜트(주)	25440	2014/12	2014.0	12.0	21.0	898.0	675.0	800.0	2438687.0
8	에스케이오션플랜트(주)	25440	2015/12	2015.0	12.0	21.0	1040.0	761.0	893.0	10910231.0
9	에스케이오션플랜트(주)	100090	2015/12	2015.0	12.0	21.0	4735.0	4020.0	4340.0	421786.0
10	에스케이오션플랜트(주)	100090	2016/12	2016.0	12.0	21.0	6490.0	5490.0	5560.0	2348745.0
11	에스케이오션플랜트(주)	25440	2016/12	2016.0	12.0	21.0	1120.0	1000.0	1110.0	5224001.0
12	에스케이오션플랜트(주)	25440	2017/12	2017.0	12.0	19.0	1970.0	1605.0	1685.0	16702041.0
13	에스케이오션플랜트(주)	100090	2017/12	2017.0	12.0	19.0	6060.0	5280.0	5480.0	1191041.0
14	에스케이오션플랜트(주)	25440	2018/12	2018.0	12.0	19.0	1010.0	688.0	847.0	13359454.0
15	에스케이오션플랜트(주)	100090	2018/12	2018.0	12.0	19.0	5660.0	4225.0	4545.0	4208533.0
16	에스케이오션플랜트(주)	25440	2019/12	2019.0	12.0	20.0	1155.0	904.0	995.0	23502255.0
17	에스케이오션플랜트(주)	100090	2019/12	2019.0	12.0	20.0	4710.0	4270.0	4505.0	1019564.0

```
df_stock.loc[df_stock['거래소코드'] == 25440, '회사명'] = '삼강엠앤티'
```

```
df_stock[df_stock['회사명'] == '삼강엠앤티']
```

회사명	거래소코드	회계년도	거래년도(*)	거래월(*)	거래일수	최고가(원)	최저가(원)	종가(원)	거래량(주)
삼강엠앤티	25440	2011/12	2011.0	12.0	21.0	820.0	701.0	779.0	394866.0
삼강엠앤티	25440	2012/12	2012.0	12.0	18.0	894.0	753.0	793.0	2514085.0
삼강엠앤티	25440	2013/12	2013.0	12.0	20.0	790.0	648.0	718.0	939171.0
삼강엠앤티	25440	2014/12	2014.0	12.0	21.0	898.0	675.0	800.0	2438687.0
삼강엠앤티	25440	2015/12	2015.0	12.0	21.0	1040.0	761.0	893.0	10910231.0
삼강엠앤티	25440	2016/12	2016.0	12.0	21.0	6490.0	5490.0	5560.0	2348745.0
삼강엠앤티	25440	2017/12	2017.0	12.0	19.0	1970.0	1605.0	1685.0	16702041.0
삼강엠앤티	25440	2018/12	2018.0	12.0	19.0	1010.0	688.0	847.0	13359454.0
삼강엠앤티	25440	2019/12	2019.0	12.0	20.0	1155.0	904.0	995.0	23502255.0
삼강엠앤티	25440	2019/12	2019.0	12.0	20.0	4710.0	4270.0	4505.0	1019564.0

# 전처리: 데이터수집

## Preprocessing : Data Collection

### 데이터 전처리

#### 데이터의 특이점 조정

- 기업규모코드 컬럼과 기업규모명 컬럼이 겹치는 항목일 수 있기 때문에 value\_counts()로 서로 값들을 비교해 보았는데 서로 맞지 않음
- 코드(5개 범주) : 0, 10, 20, 30, 90
- 규모명(4개 범주) : 중소기업, 중견기업, 대기업, 기타
  - '규모명=대기업'과 '코드=10'의 value\_counts 수가 같은 것을 볼 때  
코드 10은 대기업과 같은 항목이라고 볼 수 있다.
  - 코드 컬럼의 0은 대기업을 제외한 중소기업 일부, 기타 일부와 중견기업 일부를 나타낸다.

```
# 기업규모명에 맞춰 기업규모코드 입력하기
df.loc[df['기업규모명'] == '중소기업', '기업규모코드'] = 20.0
df.loc[df['기업규모명'] == '중견기업', '기업규모코드'] = 30.0
df.loc[df['기업규모명'] == '기타', '기업규모코드'] = 90.0
df.loc[df['기업규모명'] == '대기업', '기업규모코드'] = 10.0
```

# 전처리: 데이터 수집

## Preprocessing : Data Collection

```
# combine_first() 명령어로 개별재무제표의 값으로 연결재무제표의 동일한 컬럼의 결측치 채우기
df_filled = df1.combine_first(df)
```

df\_filled

	PBR	PER	거래소 코드	기업 규모 코드	매출액	매출액 순이익 률	매출액 증가율	부채	성장비용	영업손익	영업활동 현금흐 름	자기 자본 회전 률	자본	자산
0	2.210	0.000	58820	20.0	17751704.0	-8.50	56.40	9257624.0	376118.0	640283.0	-2560438.0	0.97	18290915.0	27548539.0
1	2.555	0.000	58820	20.0	16255389.0	-40.97	-8.43	25386064.0	401085.0	-4205503.0	-1572581.0	0.66	31602540.0	56988604.0
2	2.945	375.055	58820	20.0	20402140.0	1.45	25.51	6913407.0	243898.0	791250.0	-3939013.0	0.51	49048213.0	55961620.0
3	2.855	646.030	58820	20.0	22752584.0	0.69	11.52	9060059.0	215128.0	442322.0	724893.0	0.46	49067039.0	58127098.0
4	5.915	0.000	58820	20.0	27041894.0	-13.50	18.85	9719267.0	195671.0	-2572168.0	2410205.0	0.56	46872705.0	56591972.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
13369	1.525	27.710	3280	30.0	845115359.0	1.24	2.42	609387512.0	0.0	21244278.0	33913401.0	4.96	161824549.0	771212062.0
13370	1.050	0.000	3280	30.0	831746081.0	-2.06	-1.58	758757483.0	0.0	5895873.0	-1389623.0	4.55	183829287.0	942586770.0
13371	1.825	0.000	3280	30.0	836427496.0	-8.79	0.56	726492437.0	0.0	-13098624.0	-6892102.0	5.41	107614965.0	834107402.0
13372	2.030	0.000	3280	30.0	753865569.0	-11.48	-9.87	726055487.0	0.0	-37595967.0	-33794586.0	7.93	56731554.0	782787041.0
13373	4.890	0.000	3280	30.0	102166838.0	-50.27	-86.45	368918610.0	0.0	-12364293.0	-12048291.0	1.96	17632229.0	386550839.0

13374 rows × 20 columns

```
from scipy.stats.mstats import winsorize
df['PEG'] = winsorize(df['PEG'], limits=[0.01, 0.01])
```

✓ 0.0s

## 연결재무제표에 존재하는 결측치 값

- 개별재무제표를 확인한 결과 연결재무제표에서 결측치가 있었던 동일한 인덱스에 값이 존재함을 확인
- 개별-연결재무제표간 주요 값들의 유의미한 값의 차이가 없음을 확인
- 결측치를 개별재무제표 값으로 보충

## 이상치 조정: Winsorising

공시 데이터이므로 데이터를 이상치라고 여겨서 무분별하게 삭제하지 않고 Winsorising을 통해 이상치를 조정

# 전처리: 데이터 수집

Preprocessing : Data Collection

## 데이터 전처리

- 2011년부터 2019년까지의 재무제표 자료가 모두 존재하는 기업들만 남김:
  - 2011년 이후 상장한 기업이나 2019년 이전 상장폐지된 기업 등을 삭제
- 그 결과 1079개 종목만 잔존
- 금융업 관련 기업 제거:
  - 재무기준이 일반 업종과 다르므로 비교를 위해 제거

```
# 금융 관련 기업 제거
finance_list = [138930, 175330, 105560, 139130, 388790, 19570,
                 1290, 55550, 21080, 316140, 6220, 323410, 355150, 2
                 7330, 86790, 34830, 16610, 5830, 397880, 421800, 4
                 30610, 293580, 298870, 30210, 27830, 387310, 3540, 8
                 404950, 367340, 367360, 241520, 400, 277070, 8560, 35
                 100790, 412930, 85620, 6800, 1270, 377630, 380320, 32
                 16360, 29780, 810, 415580, 330730, 26890, 419270, 430
                 445970, 366330, 393360, 405640, 23460, 405350, 27360, 19
                 1510, 353070, 353060, 380440, 391060, 396770, 5940, 195
                 41190, 10050, 435380, 367480, 430700, 3470, 373340, 3888
                 413630, 1200, 3460, 78020, 24110, 307930, 436530, 44020
                 3690, 16600, 413600, 39490, 446750, 388220, 400560, 40676
                 418170, 427950, 430230, 435620, 372290, 377400, 400840, 123890
                 409570, 436610, 23760, 71050, 1750, 88350, 370, 3530
                 386580, 1500, 1450, 540]

df[df['거래소코드'] == 540]
df.drop(df[df['거래소코드'] == 1450].index, inplace=True)
```

# 전처리: 변수삭제

Preprocessing : Data Collection

## 변수삭제

```
df_f.drop(['재무활동으로 인한 현금흐름(*) (IFRS연결) (천원)',  
          '정상영업이익증가율(IFRS연결)',  
          '순이익증가율(IFRS연결)',  
          '총자본증가율(IFRS연결)',  
          '자기자본증가율(IFRS연결)',  
          '매출액정상영업이익률(IFRS연결)',  
          '총자본사업이익률(IFRS연결)',  
          '총자본정상영업이익률(IFRS연결)',  
          '총자본순이익률(IFRS연결)',  
          '자기자본정상영업이익률(IFRS연결)',  
          '자기자본순이익률(IFRS연결)',  
          '자본금정상영업이익률(IFRS연결)',  
          '자본금순이익률(IFRS연결)',  
          '영업비용 대 영업수익비율(IFRS연결)',  
          '수지비율(관계기업투자손의 제외) (IFRS연결)',  
          '세금과공과 대 총비용비율(IFRS연결)',  
          '유보율(IFRS연결)',  
          '사내유보율(IFRS연결)',  
          '사내유보 대 자기자본비율(IFRS연결)',  
          '적립금비율(재정비율)(IFRS연결)',  
          '평균배당률(IFRS연결)',  
          '자기자본배당률(IFRS연결)',  
          '배당성향(IFRS연결)',  
          '1주당매출액(IFRS연결)(원)',  
          '1주당순이익(IFRS연결)(원)',  
          '1주당 CASH FLOW(IFRS연결)(원)',  
          '1주당순자산(IFRS연결)(원)',  
          '1주당정상영업이익(IFRS연결)(원)',  
          '자기자본구성비율(IFRS연결)',  
          '타인자본구성비율(IFRS연결)',  
          '차입금의존도(IFRS연결)',  
          '차입금비율(IFRS연결)'], axis=1)
```

- 최종 데이터프레임을 구성하기 전에 재무데이터에서 유사/중복되는 컬럼을 삭제 \*
- 외국인 주식 수 및 외국인 주식 분포비율, 감사의견, 종업원 1인당 매출액 등 삭제, 기업의 연도별 시장점유율: 다수 결측치 존재하여 삭제
- PCR: 다른 현금흐름 관련 변수를 사용하기로 결정하여 삭제
- PSR: 다른 매출액 관련 변수를 사용하기로 결정하여 삭제
- 파생변수 생성에 사용한 변수 삭제
- 이 외의 55개 변수들을 삭제

\* 재무데이터 항목에서 유사/중복 컬럼을 삭제

# 전처리: 파생변수 생성

## Preprocessing : Making Derived Variables

### 파생변수 생성

- 현금흐름스코어 : 투자현금흐름과 영업현금흐름 활용하여 생성
- 성장비용 : 경상연구개발비, 광고 및 판매촉진비, 임차료 리스료 활용하여 생성
- 부채자본비율 : 부채와 자본 활용하여 생성
- PEG : 주당순손익과 PER을 활용하여 생성

...	현금흐름스코어	성장비용	부채자본비율	PEG
...	0	104837223	0.3421	0.1837
...	2	384720498	4.2995	1.3984
...	1	-194832	1.2847	0.4827

# 저친리·파생변수상성

```
## '성장 비용' 컬럼 생성
df_sfs_rn['성장비용'] = df_sfs_rn[['[공통] 경상연구개발비(IFRS)']] +
    df_sfs_rn[['[공통] 임차료 및 리스료(IFRS)']] +
    df_sfs_rn[['[공통] 광고 및 판매촉진비(IFRS)']]
```

# 파생변수 생성

```
## 'PER'와 'PBR' 컬럼 생성  
df_sfs_rn['PER'] = (df_sfs_rn[['[공통]PER(최고)(IFRS)']]  
+ df_sfs_rn[['[공통]PER(최저)(IFRS)']])/2
```

```
df_sfs_rn['PBR'] = (df_sfs_rn[['[공통]PBR(최고)(IFRS)']] + df_sfs_rn[['[공통]PBR(최저)(IFRS)']]) / 2
```

```
## '경상연구개발비', '임차료 및 리스료', '광고 및 판매촉진비' 컬럼들 삭제  
df_sfs_rn = df_sfs_rn.drop(['[공통] 경상연구개발비(IFRS)'  
, '[공통] 임차료 및 리스료(IFRS)'  
, '[공통] 광고 및 판매촉진비(IFRS)'], axis=1)
```

# 활용하여 생성 |, 임차료 리스로

# 전처리: 파생변수 생성

Preprocessing : Making Derived Variables

## 파생변수 생성

### 현금흐름스코어

$$\begin{array}{l} \text{투자} \\ \text{---} \\ \text{+} = 0 \\ \text{-} = 1 \end{array}$$

$$\begin{array}{l} \text{영업} \\ \text{---} \\ \text{+} = 1 \\ \text{-} = 0 \end{array}$$

- 현금흐름스코어: 투자현금흐름과 영업현금흐름 활용하여 생성
- 현금흐름스코어는 0, 1, 2의 가지고 높을수록 좋은 값

$$\text{투자} \begin{array}{c} \text{---} \\ \text{+} \end{array} = 1 + \text{영업} \begin{array}{c} \text{---} \\ \text{+} \end{array} = 1$$

현금흐름스코어 = 2

- 현금흐름스코어는 투자현금흐름스코어와 영업현금흐름스코어의 합
- (-) 투자현금흐름 - 기업이 투자를 위해 현금을 사용했다는 의미: 1
- (+) 투자현금흐름 - 기업이 투자를 회수했다는 의미: 0
- (-) 영업현금흐름 - 기업의 영업현금이 유출되었다는 의미: 0
- (+) 영업현금흐름 - 기업이 영업현금흐름을 창출했다는 의미: 1

# 전처리: 파생변수 생성

Preprocessing : Making Derived Variables

## 파생변수 생성

### 성장비용



- 성장비용 : 경상연구개발비, 광고 및 판매촉진비, 임차료 및 리스료의 합
- 기업이 미래 수익성을 높이기 위한 동력을 확보하는 신호가 다음에 해당
  - 연구 및 개발에의 투자
  - 회사의 이미지 향상과 홍보를 위한 마케팅에 투자
  - 건물 및 부지를 임대
- 성장비용이 높을수록 기업의 미래 수익성이 성장할 것이라고 판단

# 전처리: 파생변수 생성

Preprocessing : Making Derived Variables

## 파생변수 생성

### 부채자본비율

### 재무상태표(F/S)

자산

부채

자본

- 부채자본비율: 재무상태표 상 부채와 자본 활용하여 생성

부채자본비율 ( $B/E$ ) =



$$B/E > 1$$



부채자본비율이 클수록 자본 대비  
많은 부채를 가짐을 의미



$$B/E < 1$$

부채자본비율이 작을수록 자본 대비  
적은 부채를 가짐을 의미

# 전처리: 파생변수 생성

## Preprocessing : Making Derived Variables

### 파생변수 생성

#### PEG Ratio

- PEG Ratio: 피터 린치가 성장을 대비 PER를 판단하기 위해 고안해낸 지표
- 저평가된 고성장 기업을 찾기 위해 사용하는 비율

$$PEG = \frac{PER}{3\text{년 연평균 EPS 증가율}}$$

$$EPS = \frac{NI}{\text{유통주식수}}$$

\*EPS는 Earning per Share의 약자로 주당순이익을 의미

$$PEG \downarrow = \frac{PER}{3\text{년 연평균 EPS 증가율}} \uparrow$$

- PEG 해석: PEG가 낮다는 것은 다음의 의미를 지닌다.
  - 분자인 PER가 낮아서 주가가 저평가되어 있거나
  - 분모인 EPS 성장률이 높아서 기업의 이익이 높은 성장성을 보인다

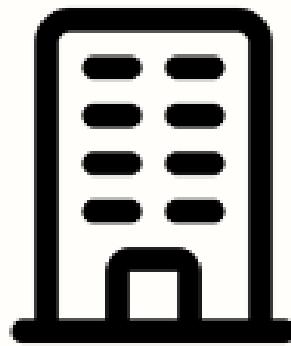
# 전처리: 파생변수 생성

Preprocessing : Making Derived Variables

## 예시

### 파생변수 생성

PEG Ratio

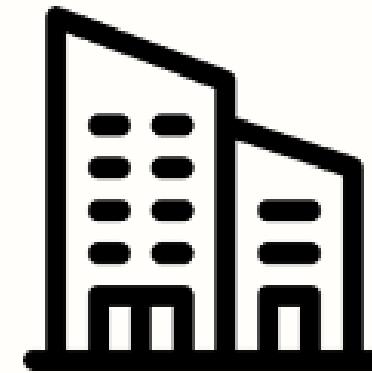


PEG = 5

회사 A

PER=10

순이익 연 2% 성장



PEG = 0.5

회사 B

PER=10

순이익 연 20% 성장

“PER가 같지만 B가 더 저평가”

# 전처리: Feature 선정

Preprocessing : Feature Selection

## <재무비율 관련 변수>

매출액증가율, 영업손익, 자기자본회전율, 주당순손익, PBR, PER, 매출액순이익률

## <파생변수>

현금흐름스코어, 성장비용, 부채자본비율, PEG

## <기타>

기업규모코드

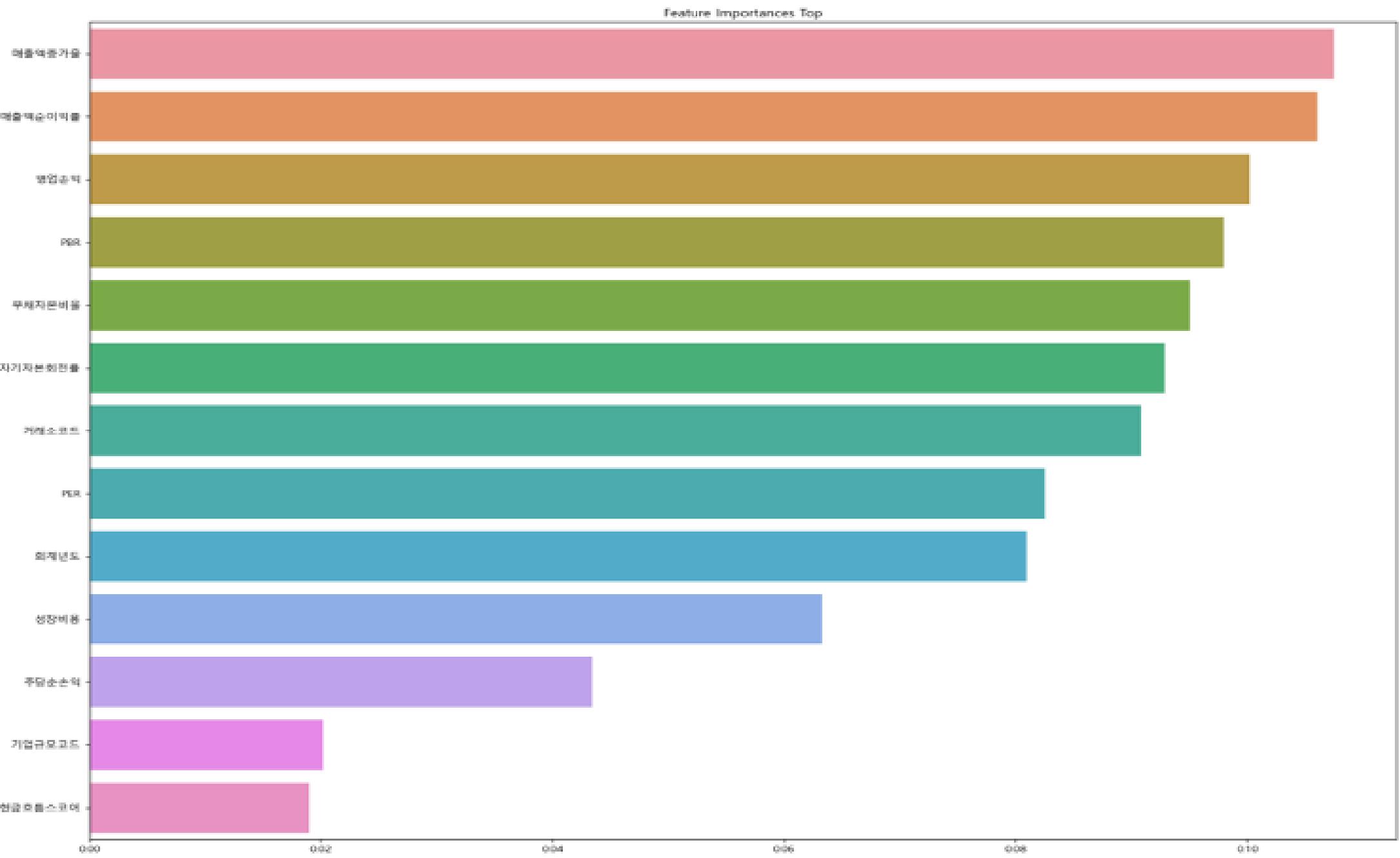
- 선정된 Feature 간의 상관관계 확인

# 전처리: Feature 선정

Preprocessing : Feature Selection

## Feature 선정

RandomForest를 이용,  
피처 중요도를 확인해본 결과  
매출액 관련 변수, 영업손익,  
PBR, PER, 부채자본비율, 성  
장비용, 회전율 관련 지표의  
중요성이 높은 것 확인



# 전처리: Target 설정

Preprocessing : Setting Target

## TARGET

'투자신호'

```
df['투자신호'] = ''  
  
for i in df.index:  
    if df.loc[i, '수익률'] < 6.5:  
        df.loc[i, '투자신호'] = 0  
    else:  
        df.loc[i, '투자신호'] = 1
```

- 타겟을 '투자신호'라고 명명
- 기준이 되는 목표수익률 6.5%로 설정 (무위험수익률 : 1.5%)
- 6.5% 미만 수익률 - 투자 부적격 주식으로 간주 : 0,
- 6.5% 이상 수익률 - 투자 적격 주식으로 간주 : 1

기준	수익률	투자신호
목표수익률 6.5%	▲	1
	▼	0

**재무기간**  
(차기 3월말까지 공시)

**보유기간**  
(04.01~03.31)

**Train/Test**

# 전처리: Train/Test data

## Train/Test Data Overview

2011년	2012년 - 2013년	Train
2012년	2013년 - 2014년	Train
2013년	2014년 - 2015년	Train
2014년	2015년 - 2016년	Train
2015년	2016년 - 2017년	Train
2016년	2017년 - 2018년	Train
2017년	2018년 - 2019년	Train
2018년	2019년 - 2020년	Test

년도	회사명	매출액 증가율	부채 비율	수익률	투자 신호
2011	A	25	0.5	8%	1

### 최종 데이터 구성

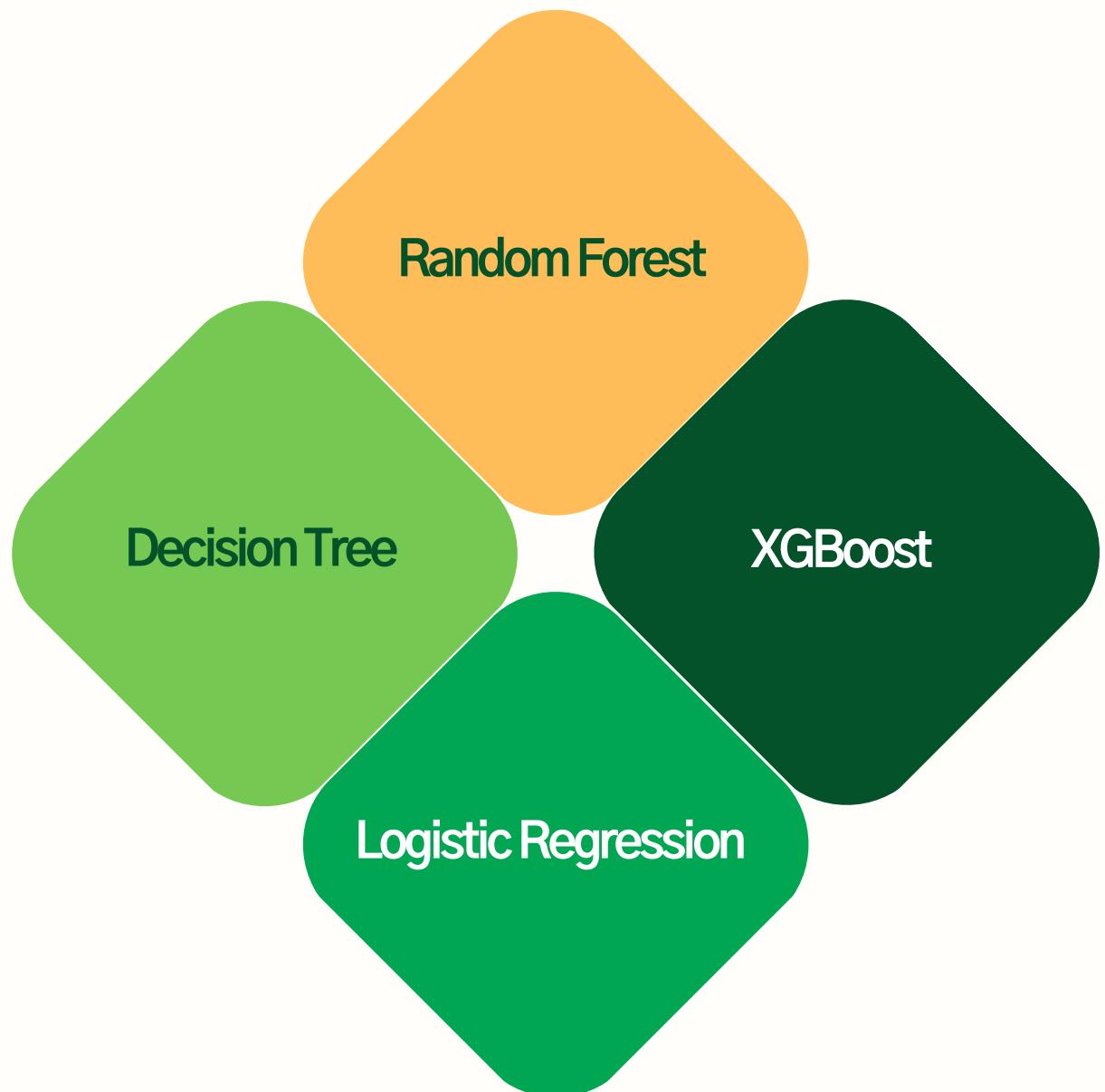
- 재무기간: 선정된 피처 중
  - 손익계산서 항목은 연초 시점에서 연말 시점까지의 변화율
  - 재무상태표 항목은 연말 시점의 항목을 이용
- 우리는 2011년의 재무데이터를 2012년 4월에 볼 수 있음
- 보유기간: 주식을 보유한 기간(1년)
- Train Data는 2011~2017년, Test Data는 2018년
- Scaling : MinMaxScaler 채택

# Modeling / Backtesting

1. 알고리즘 선정 및 Modeling
2. Backtesting

# 알고리즘선정 및 모델링

Algorithm Selection & Modeling



## 알고리즘 후보 선정 이유

- 분류 알고리즘을 사용하므로 분류 모델 중에서 고려
- DecisionTree는 분류모델의 기본 트리계열 모델로 선정
- 성능향상을 위해 Boosting, Ensemble 계열의 XGBoost, RandomForest 모델을 선정
- Logistic Regression은 선형으로 분류 작업을 시행할 수 있어 고려

# 알고리즘선정 및 모델링

## Algorithm Selection & Modeling

최종선택 모델 : RandomForest

모델 선정 이유

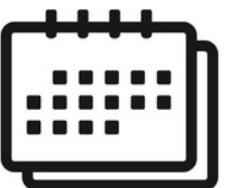
	precision	recall	f1-score	support
0	0.59	0.97	0.73	876
1	0.65	0.08	0.14	635
accuracy			0.59	1511

연구 목표상 1을 예측하는게 중요  
그래서 1의 Precision 값에 더 주목했는데,  
RandomForest 알고리즘의 값이 가장 높았다.

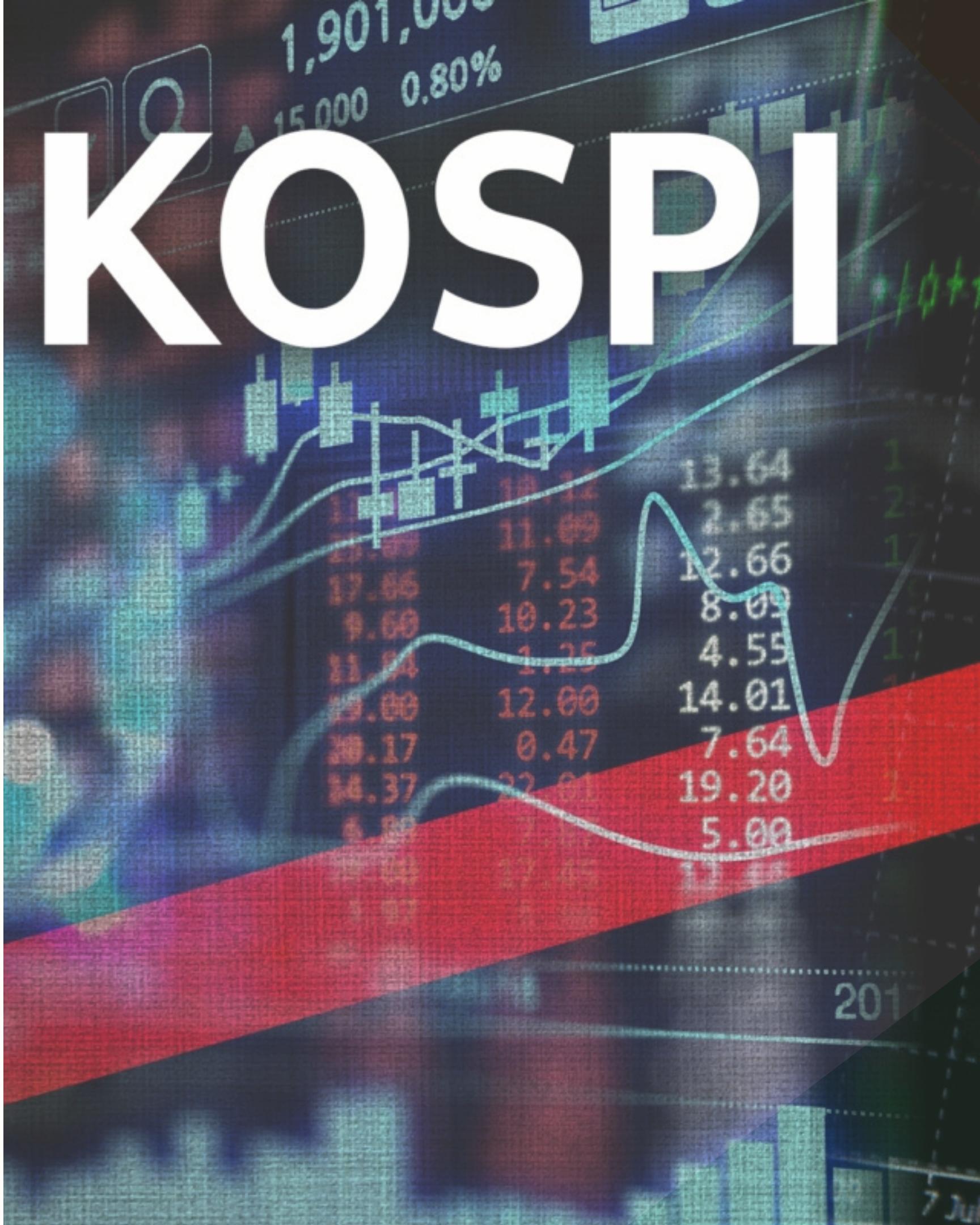
# 백테스팅

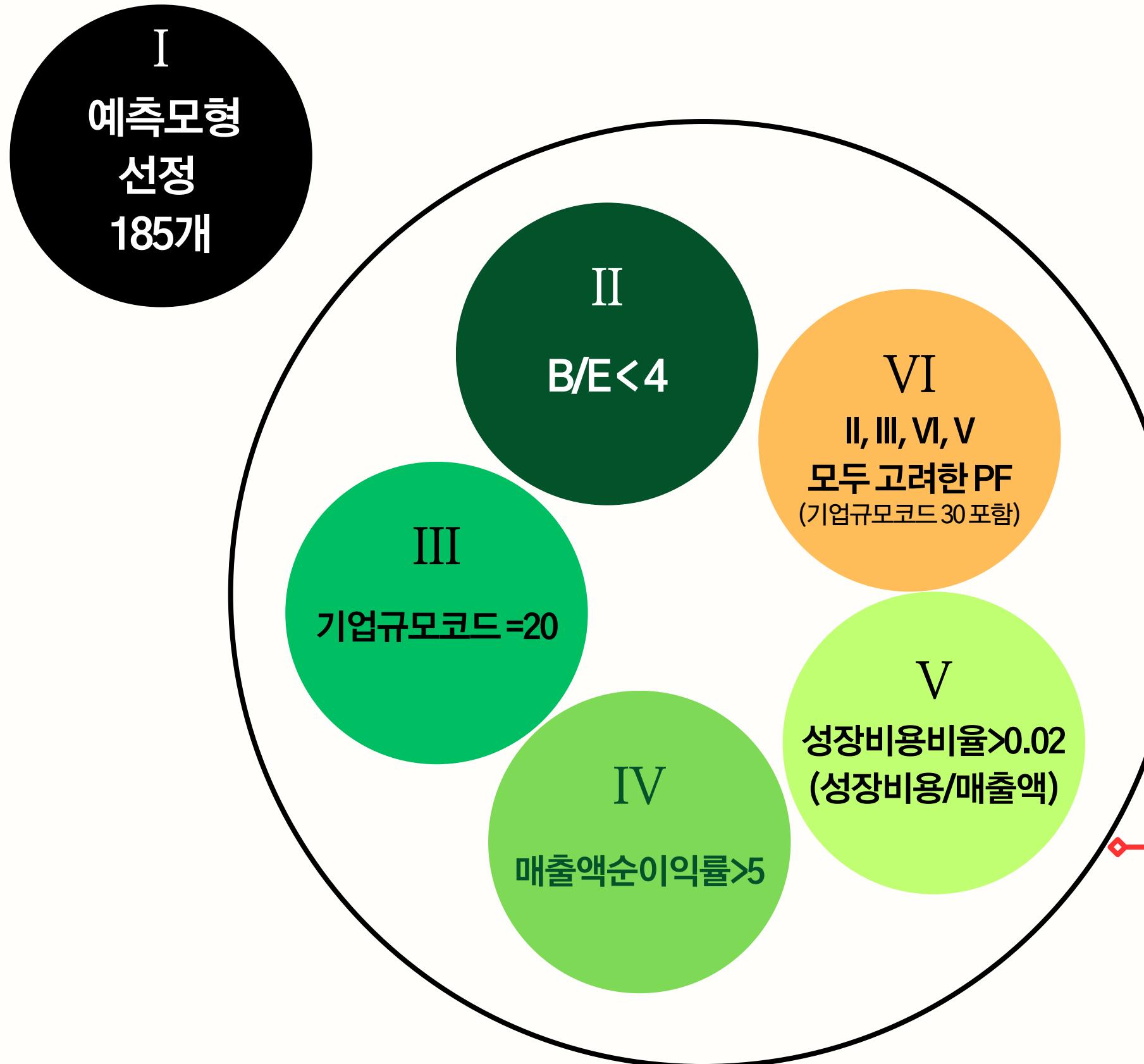
Backtesting

- 비교군(Benchmark) :  
KOSPI 지수



(2019.04.01 – 2020.03.31)





### 대조군 분류 조건

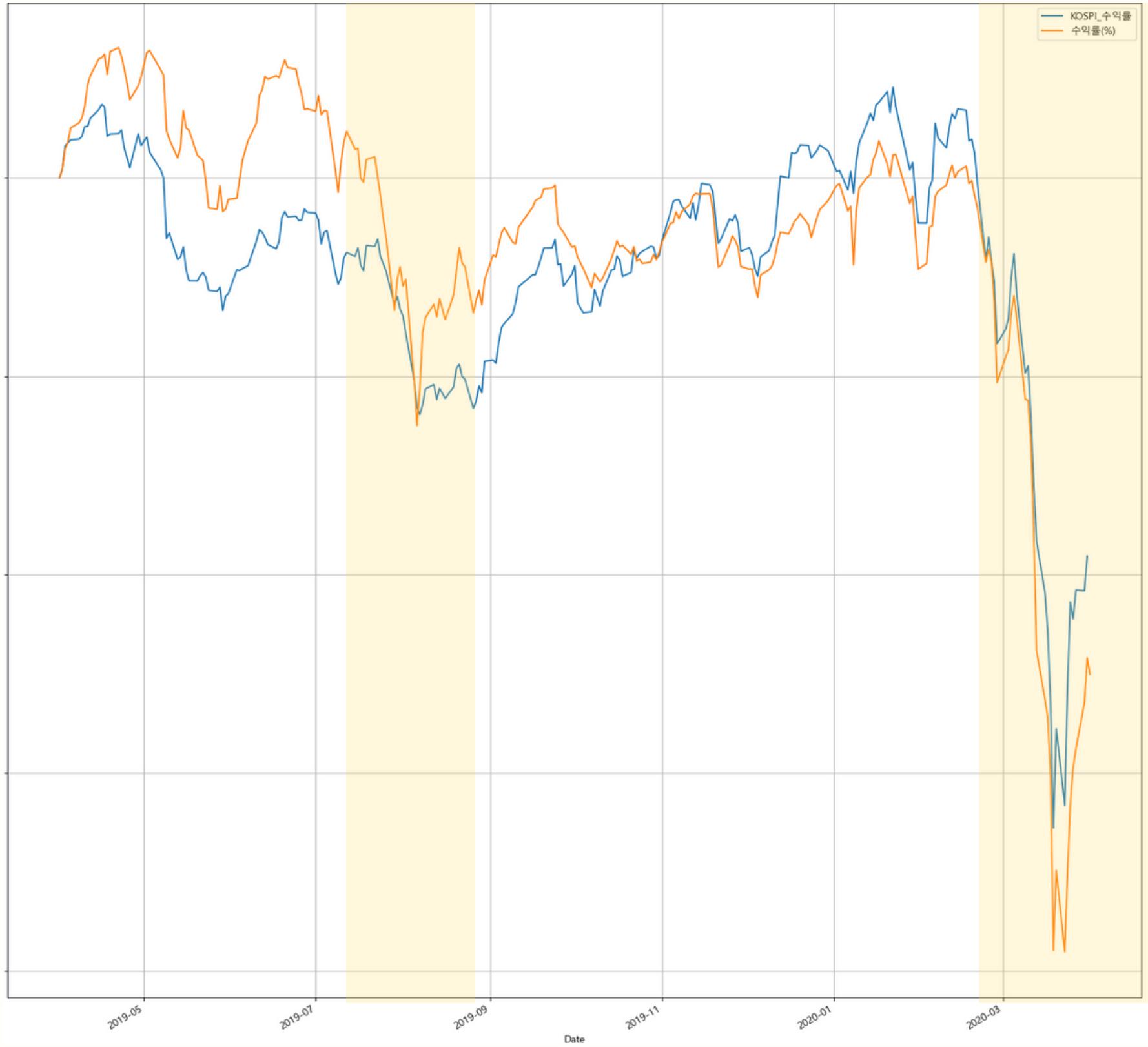
- $0 < PEG \leq 1.5$
- 각 포트폴리오 별 제약조건

위 두 기준으로 필터링

PEG 낮은 순으로 10개 기업 선정

# 대조군 vs KOSPI

Backtesting using KOSPI Index



## 대조군 분류 조건

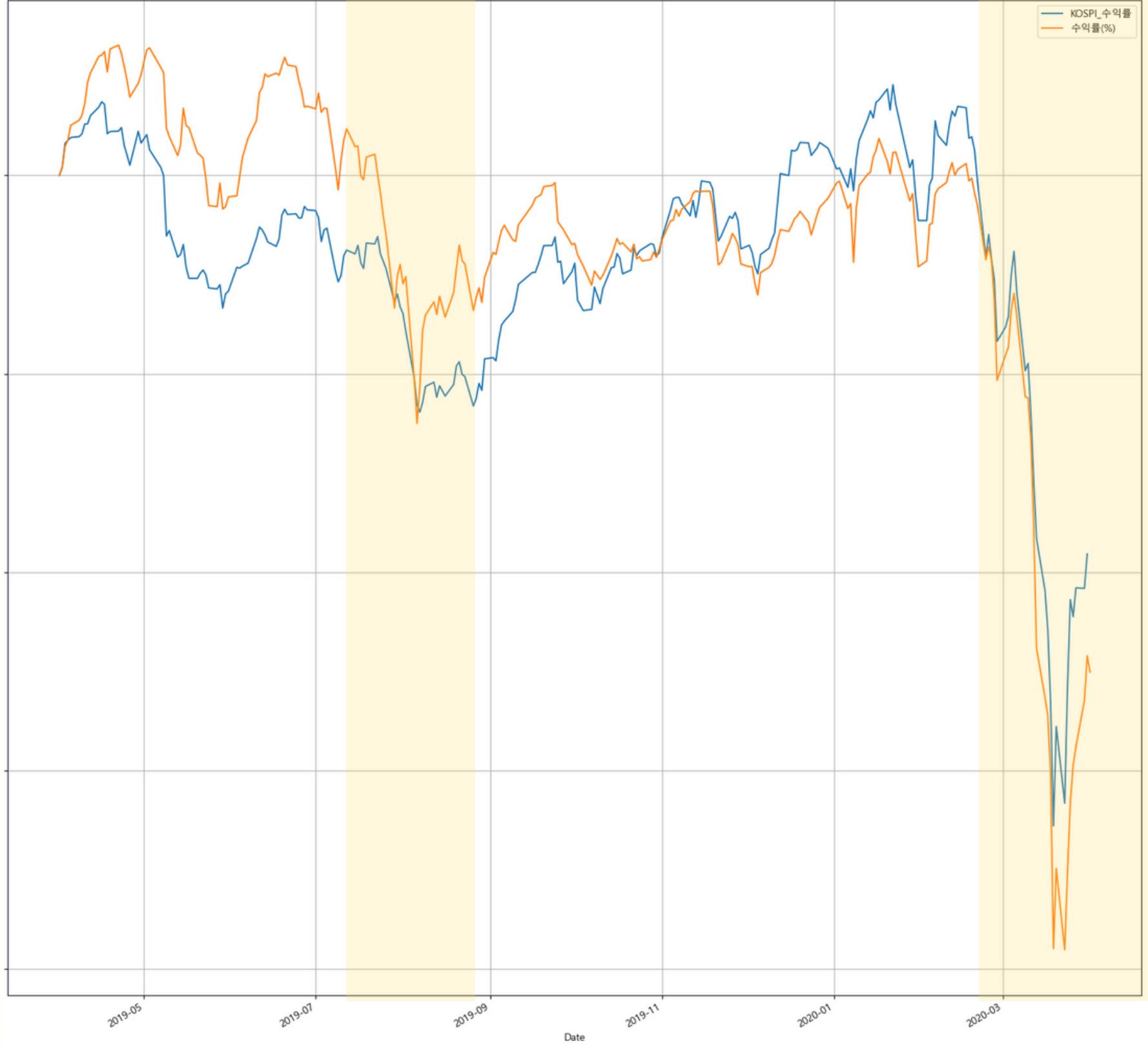
실제 투자가능 지수와 상관없이 예측모델이 분류한 투자 가능 주식 185개를 모두 백테스팅함.

**대조군I**  
Sharpe Ratio  
**-0.60**

**KOSPI**  
Sharpe Ratio  
**-0.7696**

# 대조군 vs KOSPI

## Backtesting using KOSPI Index



### Insight

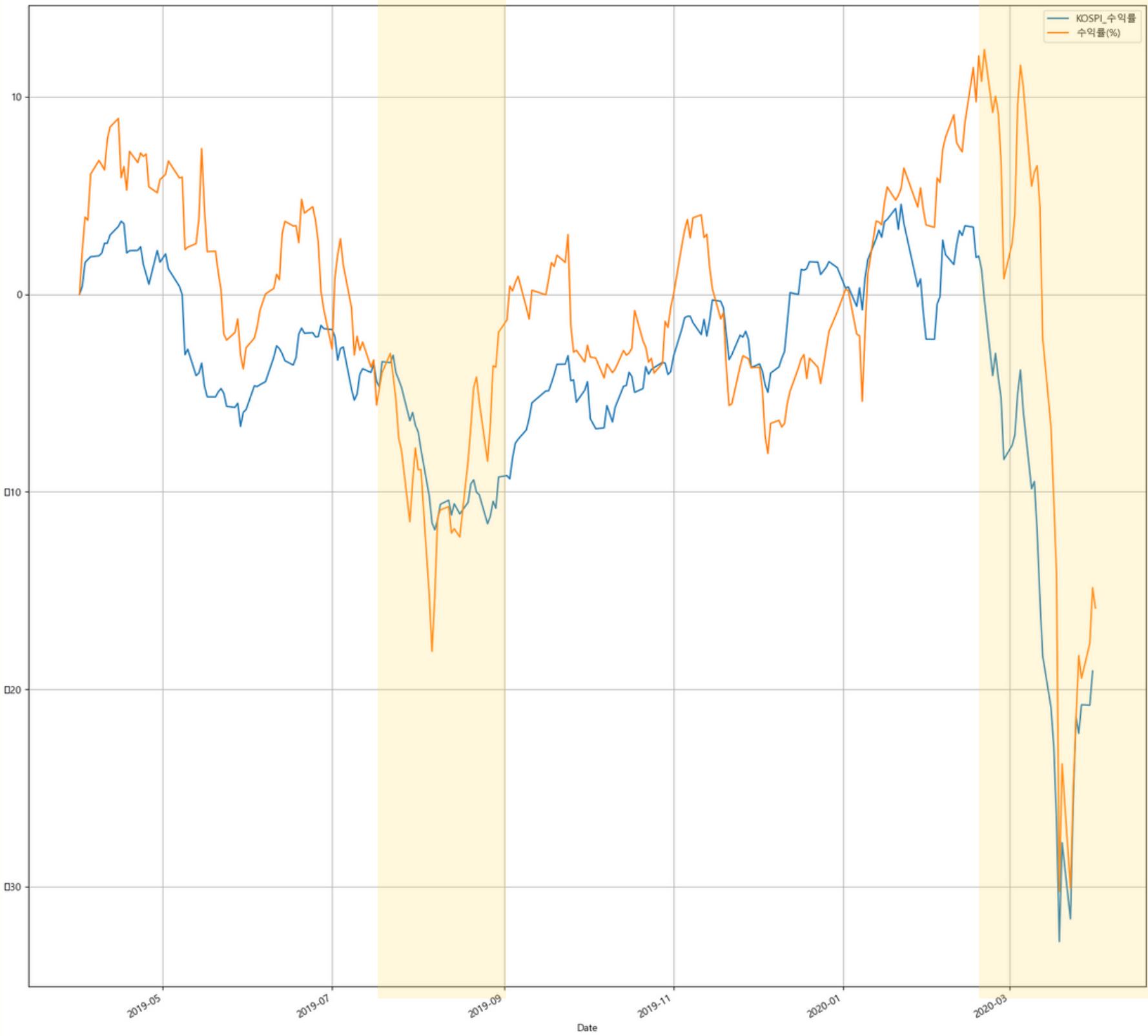
- 2019년 10월 이전까지는 KOSPI 지수를 오히려 능가함
- 이후 KOSPI와 비슷하다가 COVID가 본격화된 2020년 1월 이후 떨어지기 시작

# 모델의 예측값이 KOSPI 지수에 크게 떨어지지 않는 이유는 모델의 성능이 떨어지더라도 1차 필터 역할은 한것으로 판단

	전체기업 (1079개 기업)	예측기업 (185개 기업)
영업손익/매출액 평균	0.030846	0.356300
$\Sigma$ PEG/기업수	217.816577	161.932883
$\Sigma$ PER/기업수	34.555194	21.855837

# 대조군|| vs KOSPI

Backtesting using KOSPI Index



## 대조군 분류 조건

- $0 < \text{PEG} \leq 1.5$
- 기업규모코드 = 20 (중소기업)

## PEG 낮은 순으로 10개 기업 선정

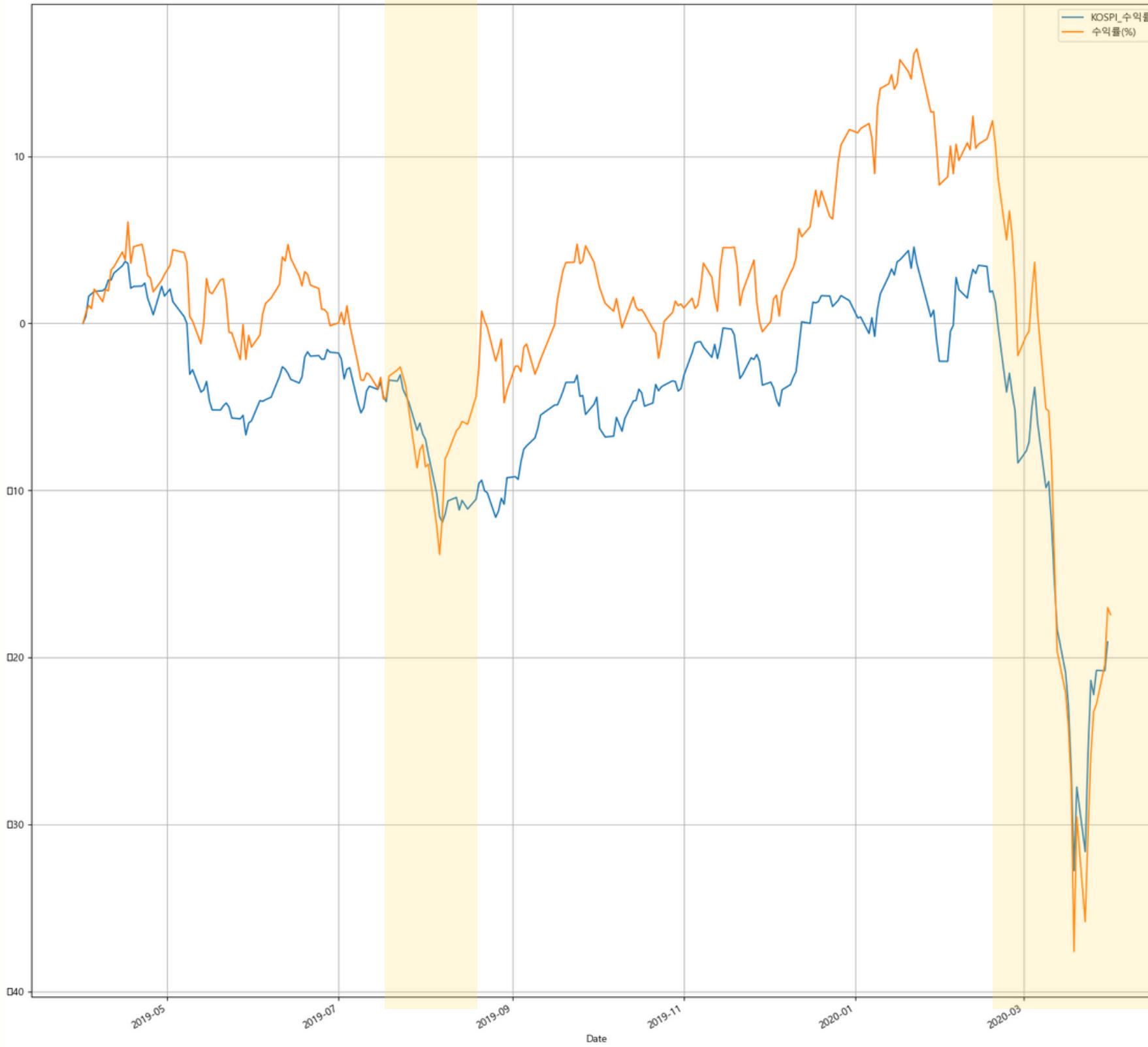
- PEG 기반
- 기업규모에 따라 수익률의 차이가 있는지 보기 위한 포트폴리오
- 변동성이 큰 만큼 오를 때는 지수보다 더 오르나 떨어질 때는 하락폭이 더 큼

대조군||  
Sharpe Ratio  
-0.40

KOSPI  
Sharpe Ratio  
-0.7696

# 대조군Ⅲ vs KOSPI

Backtesting using KOSPI Index



## 대조군 분류 조건

- $0 < \text{PEG} \leq 1.5$
- 매출액순이익률 > 5

## PEG 낮은 순으로 10개 기업 선정

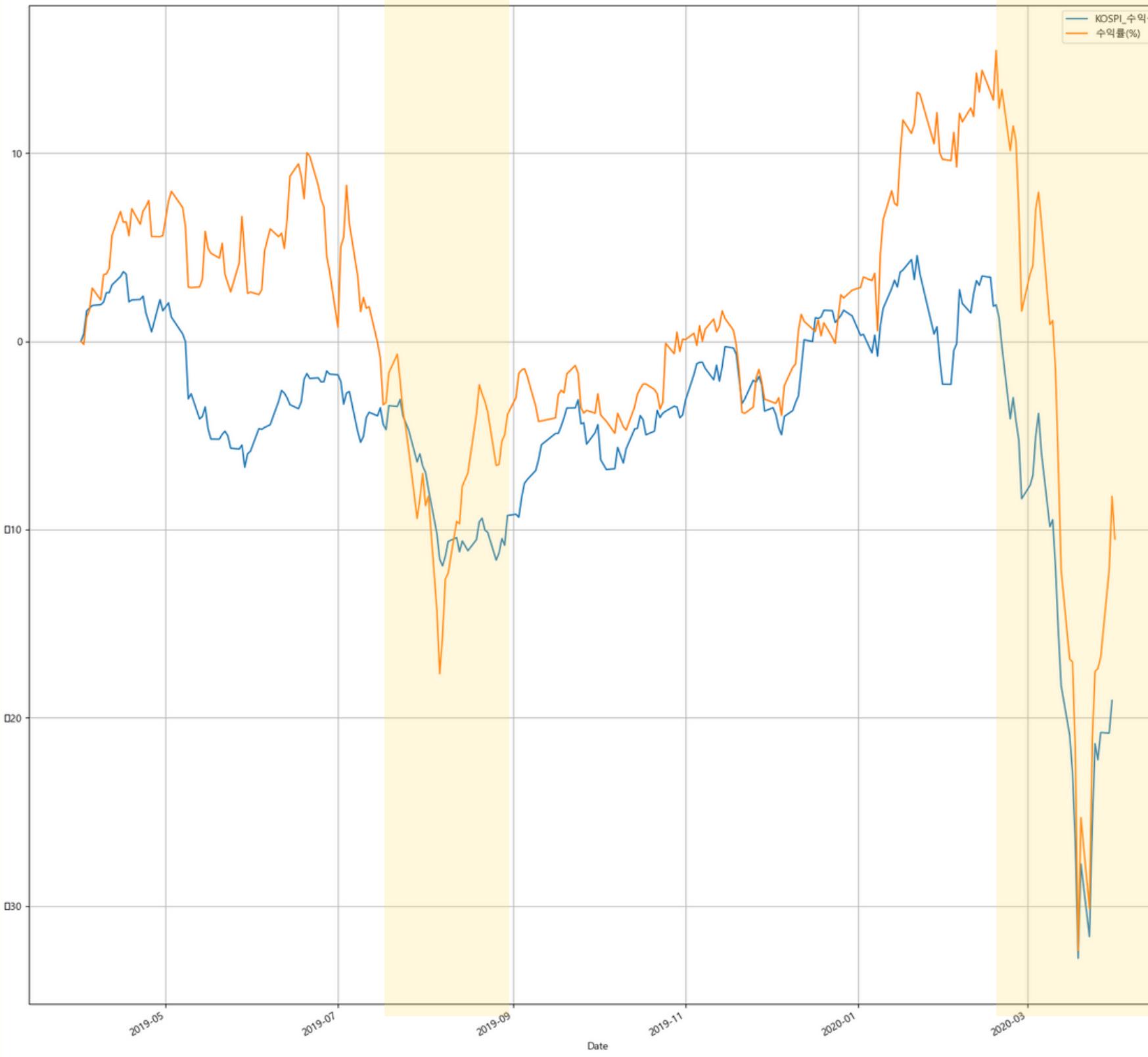
- 최종 포트폴리오로 선정한 포트폴리오와 유사하나 샤프지수가 떨어짐
- 투자 위한 주식 선정 시 매출액순이익률을 필터로 고려시
  - 상승폭은 적을 수 있고 변동성은 낮게 유지할 수 있을 것으로 보임
  - 저위험 선호 투자자들의 고려 사항

대조군Ⅲ  
Sharpe Ratio  
-0.22

KOSPI  
Sharpe Ratio  
-0.7696

# 대조군IV vs KOSPI

Backtesting using KOSPI Index



## 대조군 분류 조건

- $0 < \text{PEG} \leq 1.5$
- 성장비용비율  $> 0.02$

(성장비용비율 = 성장비용 / 매출액)

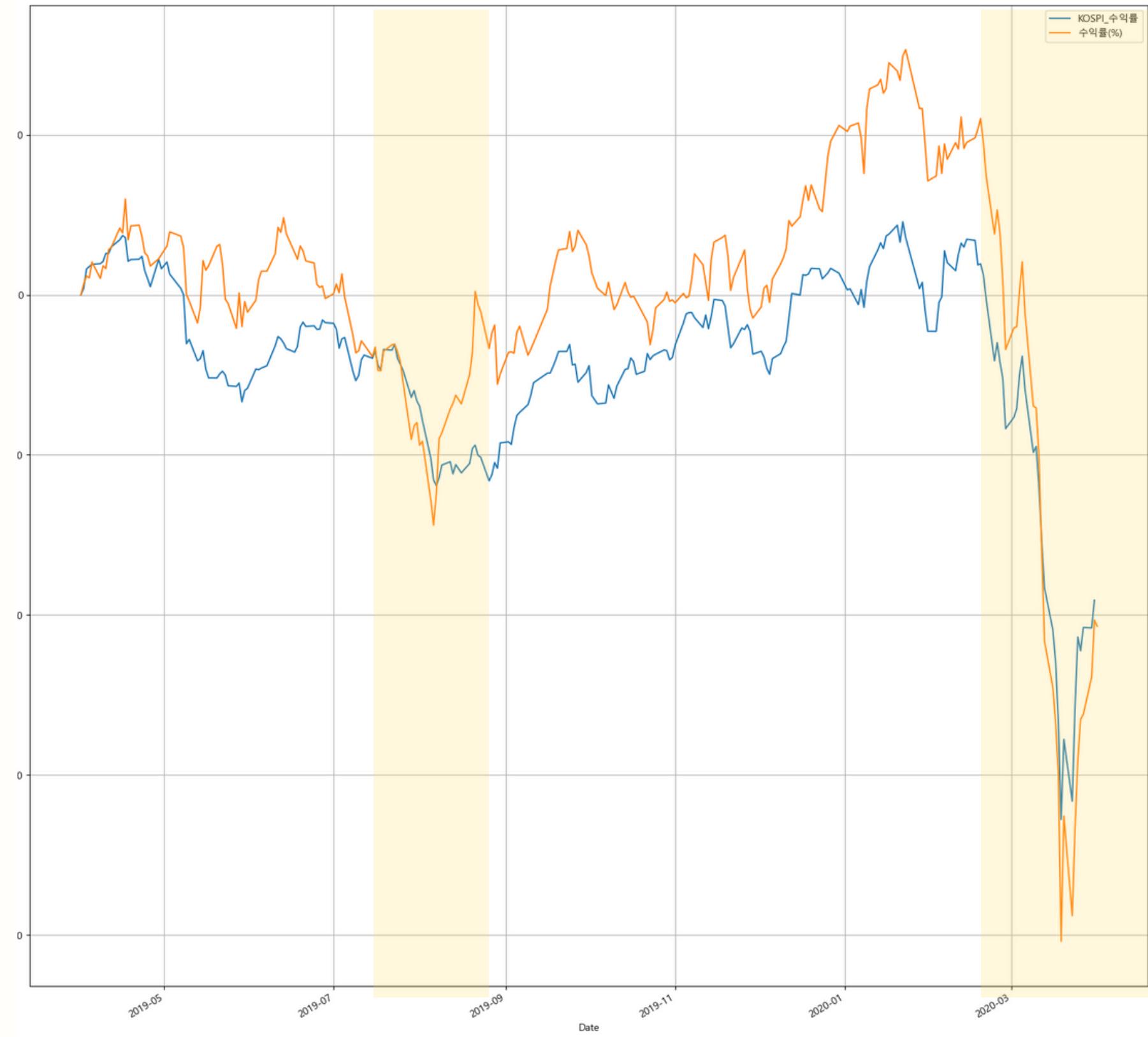
## PEG 낮은 순으로 10개 기업 선정

- 성장 비용이 비교적 높은 기업 주식들로 구성된 포트폴리오
- 기업수명주기 이론상 성장기 기업들로 파악
- 변동성이 비교적 큰 것을 확인



# 대조군V vs KOSPI

## Backtesting using KOSPI Index



### 대조군 분류 조건

- $0 < \text{PEG} \leq 1.5$
- 부채자본비율 < 4

### PEG 낮은 순으로 10개 기업 선정

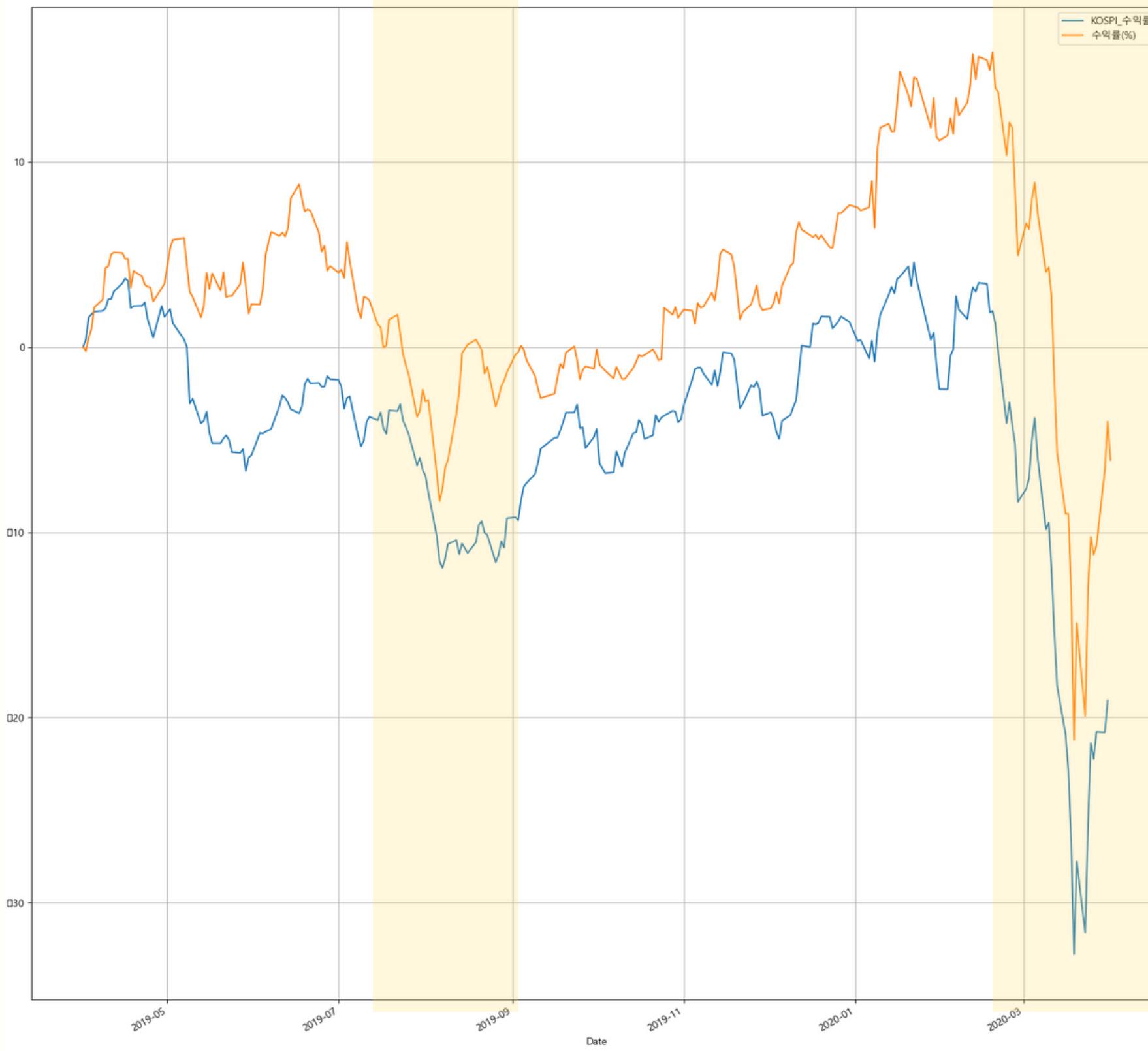
- PEG 기준으로 주식에 투자해서 KOSPI지수를 능가하는 샤프지수와 수익률을 기록할 수 있음을 보여줌
- 부채자본비율 역시 저변동성을 선호하는 투자자들이 투자주식 선정 시 고려 해야 할 항목으로 확인

대조군V  
Sharpe Ratio  
-0.18

KOSPI  
Sharp eRatio  
-0.7696

# 대조군VI vs KOSPI

Backtesting using KOSPI Index



## 대조군 분류 조건

- $0 < \text{PEG} \leq 1.5$
- 부채자본비율 < 4
- 기업규모코드 = 20,30
- 매출액순이익률 > 5
- 성장비용비율 (=성장비용/매출액) > 0.02

(중소+중견기업) → 중소만 뽑으면 5개 종목 뿐이라 중견도 포함

## PEG 낮은 순으로 10개 기업 선정

- 다른 포트폴리오의 선정 기준들을 종합해서 구성한 포트폴리오
- PEG 뿐 아니라 수익성을 고려해서 주식 선정

대조군VI

Sharpe  
Ratio **-0.23**

KOSPI

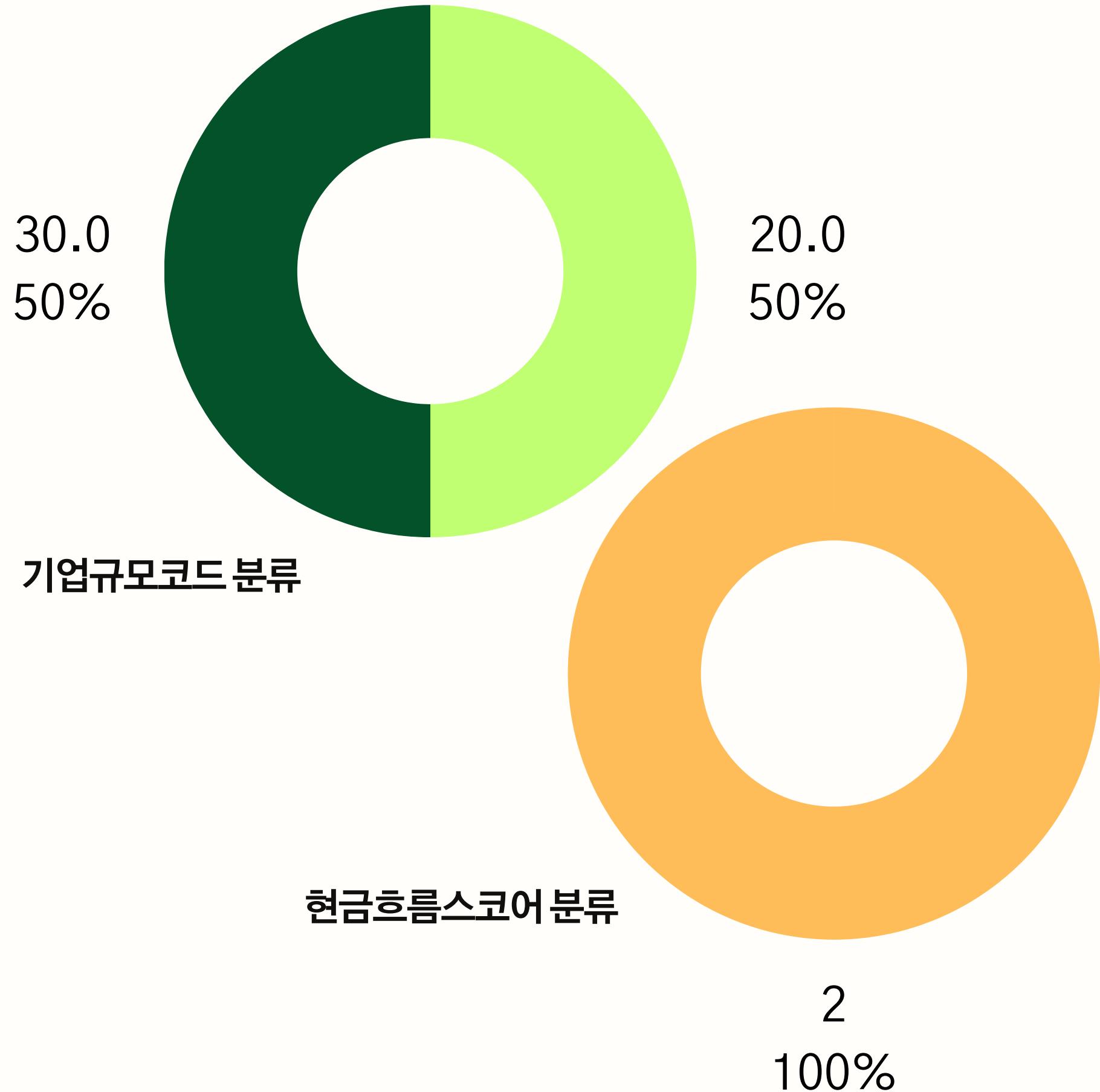
Sharpe  
Ratio  
**-0.7696**

# Conclusion

1. Insight
2. 보완점
3. 기여
4. 참고 문헌

# 인사이트: 최종 포트폴리오 주식의 특성

Insight: Features of Final Portfolios



포트폴리오에 선정된 기업:  
기업규모코드에 따른 분류를 보면 중소기업과  
중견기업이 절반씩 분포

영업, 투자현금흐름을 통합,  
현금흐름스코어 피쳐 생성

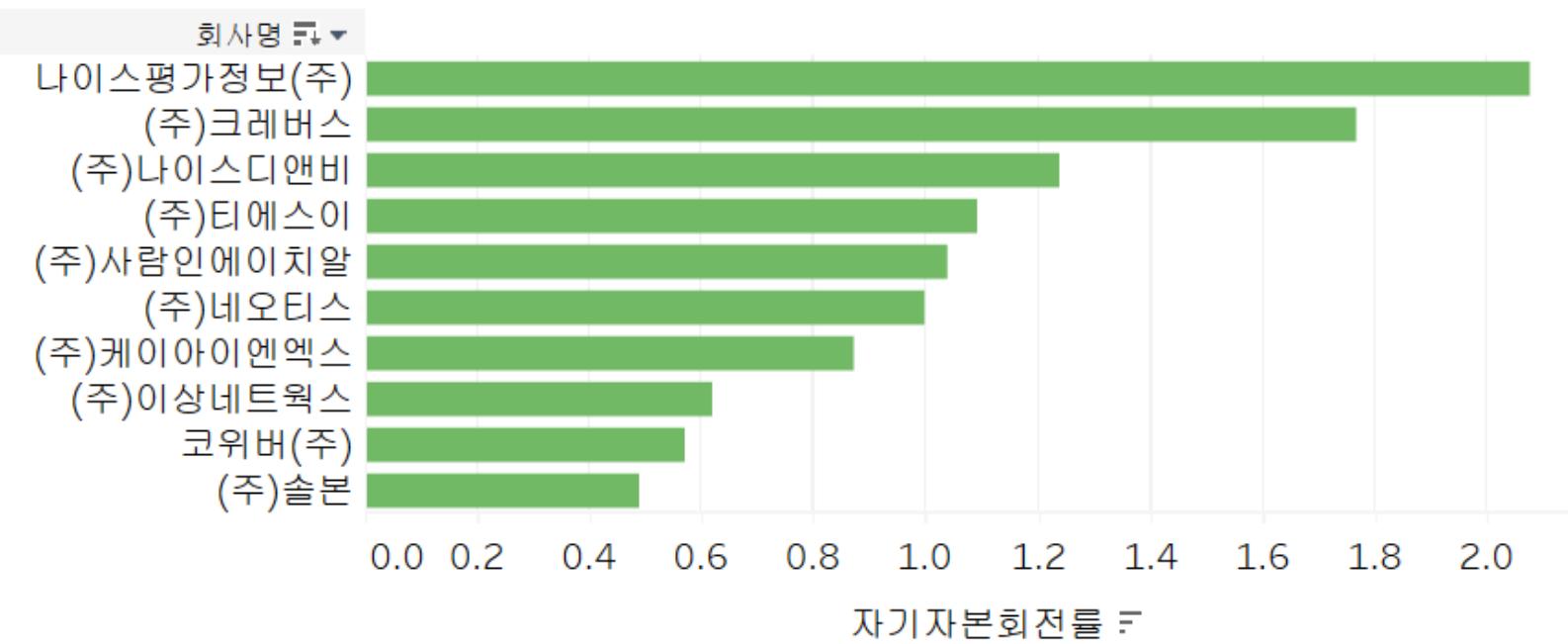
- 현금흐름이 마이너스 값은 좋지 않은 수치
- 현금흐름을 0을 기준으로 컬럼의 값을 1, 2  
로 현금흐름스코어를 분류

분류된 10곳 모두 현금흐름 점수가 좋음

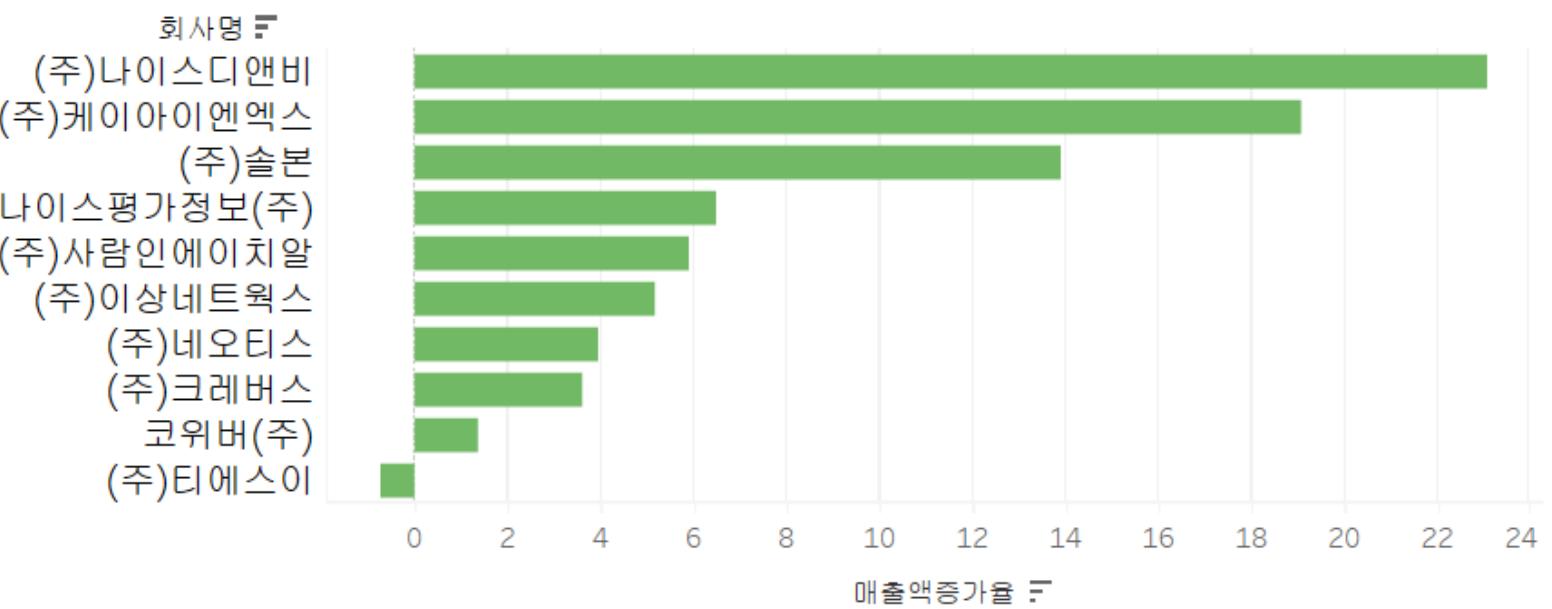
# 인사이트: 최종 포트폴리오 주식의 특성

Insight: Features of Final Portfolios

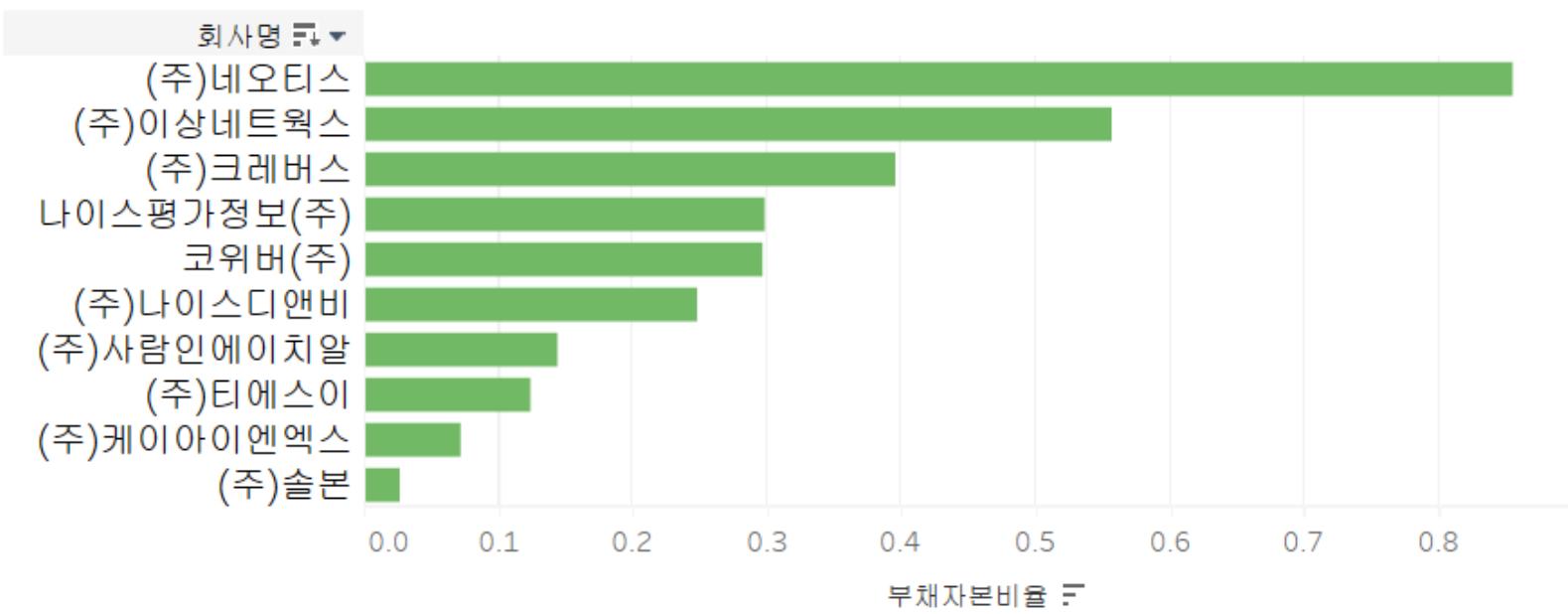
기업별 자기자본회전률



기업별 매출액증가율

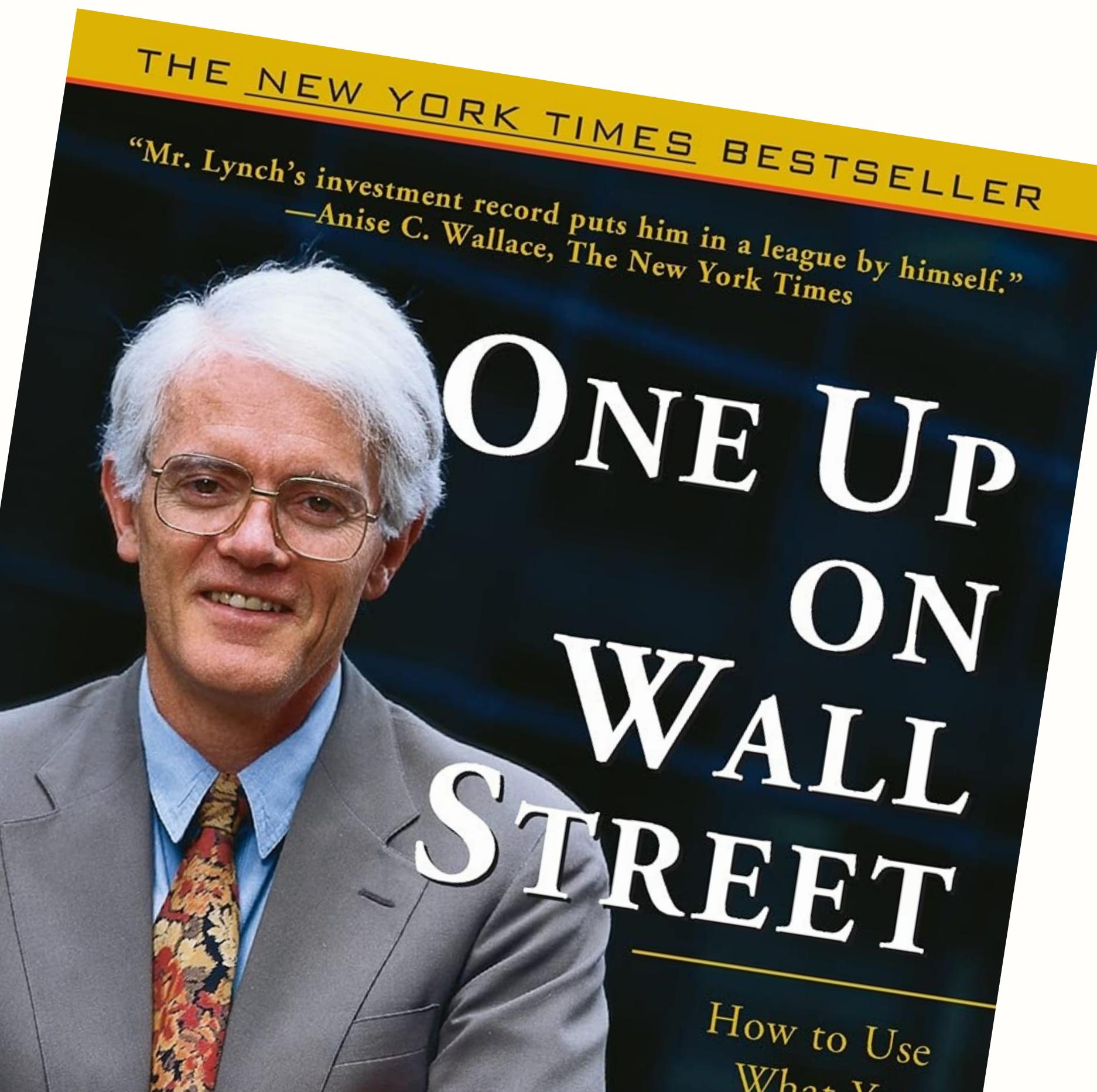


기업별 부채자본비율



# 인사이트: PEG비율

Insight: PEG(Price/Earnings to Growth) Ratio



- Peter Lynch의 메트릭을 기반으로 PEG를 계산
- 모델의 성능에 미치는 영향력 확인 및 포트폴리오를 만드는 10개의 주식 분류 사용
- PEG 기준으로 만든 포트폴리오가 백테스트 기간 동안 코스피 지수를 능가함

# 인사이트: 머신러닝 모델에서 재무제표의 역할

Insight: Roles of Financial Statements on Machine Learning Model

Weight	Feature
0.0408 ± 0.0154	PBR
0.0130 ± 0.0189	영업손익
0.0119 ± 0.0122	매출액
0.0116 ± 0.0098	매출액증가율
0.0107 ± 0.0068	PER
0.0093 ± 0.0113	PEG
0.0066 ± 0.0067	매출액순이익률
0.0065 ± 0.0050	성장비용
0.0030 ± 0.0095	총자본회전률
0.0017 ± 0.0058	기업규모코드_10.0
0.0009 ± 0.0053	현금흐름스코어
0.0001 ± 0.0103	자기자본회전률
0 ± 0.0000	기업규모코드_90.0
-0.0001 ± 0.0031	기업규모코드_20.0
-0.0013 ± 0.0014	기업규모코드_30.0
-0.0026 ± 0.0044	부채자본비율

Machine Learning 모델 성능에 중요한 역할을 하는 재무제표 항목들 확인

- Positive : PBR, 영업손익, 매출액 관련 항목들, PER, PEG, 성장비용
- Negative : 총자본회전률, 기업규모코드, 현금흐름스코어, 자기자본회전률, 부채자본비율

### 1. 모델의 성능- 데이터 처리와 모델링 설정 숙련도의 부족

### 2. 포트폴리오

- 투자 기간 내 Buy & Hold 전략 수행 시 목표로 했던 수익률 달성 실패
  - 투자 기간 설정의 어려움
  - 목표 수익률 달성을 시 매도 전략으로 전환 가능
- 다양한 고객의 위험선호도에 따른 포트폴리오를 구축하지 못함

### 3. 모델링을 통한 절차의 자동화 구현 못함

- 재무제표를 활용하여 포트폴리오 구축에 필요한 주식을 선별하는 데 머신러닝 알고리즘을 활용했다.
- 모델의 성능이 떨어지더라도 모델의 예측 결과로 구축한 포트폴리오의 수익률이 설정한 투자 기간 내 KOSPI 지수를 이길 수 있었다.
- 모델링 과정에서 PEG 계산, Feature로 활용
- 투자자의 위험 감수 성향을 감안하여 주식 포트폴리오 구성

### Feature 및 알고리즘 참고

- Troy J Strader; John J Rozycki; Thomas H Root; Yu-Hsiang John Huang. "Machine Learning Stock Market Prediction Studies: Review and Research Direction", pages 5–15  
Published By: Journal of International Technology and Information Management, 2020
- Matthias X. Hanauer; Marina Kononova; Marc Steffen Rapp. "Boosting Agnostic Fundamental Analysis: Using Machine Learning to Identify Mispricing in European Stock Markets", pages 1–22  
Published By: Journal of International Technology and Information Management, 2022
- Oscar Azrak; Alperen Kinalli; Kristian Makadsi. "Applying machine learning to automate stock portfolio management", pages 40–54  
Published By: HKUST, 2021
- You Haifeng; Cao, Kai. "Fundamental Analysis via Machine Learning", pages 2–10  
Published By: KTH Royal Institute of Technology, 2022
- Yuxuan Huang; Luiz Fernando Capretz; Danny Ho. "Machine Learning for Stock Prediction Based on Fundamental Analysis", pages 2–9  
Published By: IEEE, 2021

### Feature 및 알고리즘 참고

- Sima Siami-Namini; Neda Tavokoli; Akbar Siami Namin. "A Comparative Analysis of Forecasting Financial Time Series Using Arima, LSTM, and BiLSTM", pages 2–8  
Published By: Texas Tech University, Georgia Institute of Technology, 2019
- 이해영, 김형규. "기본적 변수가 주식수익률에 미치는 영향 – 패널자료로부터의 근거", pages 21–24  
Published By: 국토연구, 2021
- 장근혁. "2019년 국내 주식시장 동향과 시사점", pages 1–6  
Published By: 자본시장포커스, 2019

### 재무 데이터 관련 참고

- Easton Peter D. "PE Ratios, PEG Ratios, and Estimating the Implied Expected Rate of Return on Equity Capital", The Accounting Review Vol. 79, No. 1, pages 73–95  
Published By: American Accounting Association, 2004
- 왕현선. "K-IFRS 도입에 따른 재무비율이 신용평가에 미치는 영향", pages 27–56  
Published By: 대한경영정보학회, 2016

# Thank you