



오픈소스SW기초 6분반

분산 투자 전략 지원, SABU

#주식 #분산 투자

사고8조

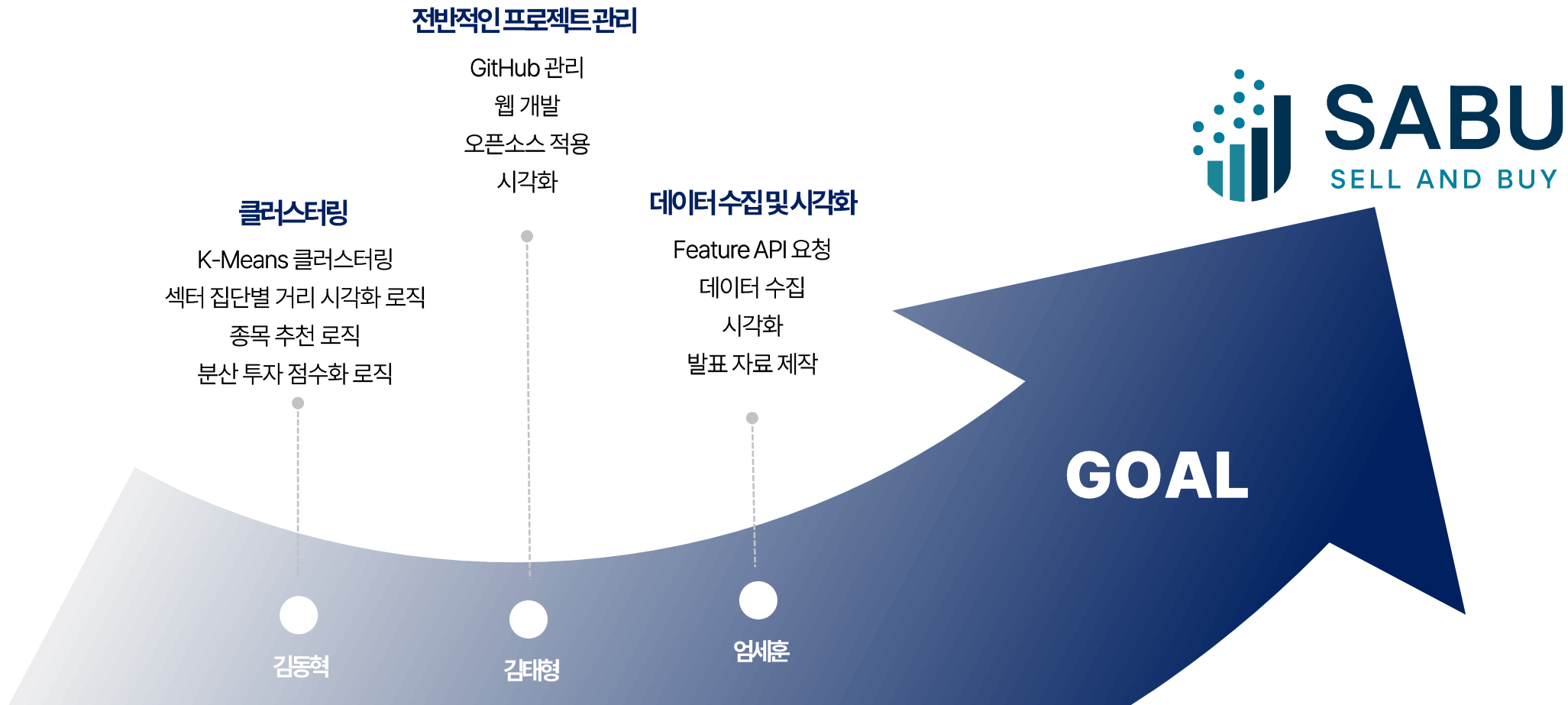
정보통계학과	32200472	김동혁
컴퓨터공학과	32211228	김태형
소프트웨어학과	32232597	엄세훈

목차

- 0 역할
- 1 문서화
- 2 클러스터링
- 3 백테스팅
- 4 Testing
- 5 추가 기능 및 시연

0. 역할

역할 분담



1. 문서화

소제목




- 발표에서 필수적인 것만 소개
- 옆에 코멘트로 깃허브에 다 올려놨고 어디에 뭐가 있다 설명 달기

프로젝트 설계 문서 모음

이 레포지토리는 프로젝트의 기획, 설계, 구조, 아키텍처 등의 **비개발 산출물**을 정리하기 위한 공간입니다.
프론트엔드/백엔드 개발 이전의 **기획·디자인·요구사항 정의서** 등을 모두 포함합니다.

문서 목록

-  [기획 개요](#)
-  [디자인 문서](#)
-  [유스케이스 / UML](#)
-  [API 명세](#)
-  [WBS 및 일정표](#)
-  [주차별 회의록](#)
-  [발표자료 및 포스터](#)
-  [향후 개선 계획](#)
-  [의사결정 기록](#)
-  [회고 및 문제 해결](#)

폴더 구조

```

docs/
├── design/      # 디자인 설계 문서
├── uml/         # 유스케이스, 클래스 다이어그램 등
├── api-spec/    # API 명세
├── schedule/    # 일정표, WBS 등
├── presentation/ # 발표용 자료
├── meeting/     # 회의록
├── project-overview.md
├── future-work.md
└── README.md
    
```

Frontend (React)

이 디렉토리는 사용자 인터페이스(UI)를 담당하는 **프론트엔드 애플리케이션**입니다.
React 기반으로 개발되었으며, REST API를 통해 백엔드와 통신합니다.

주요 기능

- 사용자 인증 (로그인/회원가입)
- 프로젝트/서비스 대시보드
- 시각화 컴포넌트 (차트, 노드 기반 UI 등)
- 반응형 인터페이스

기술 스택

- React + Vite
- useContext 상태 관리
- TailwindCSS
- Axios

폴더 구조

```
frontend/
├── docs/                # 화면 설계 문서, 컨벤션 등
│   ├── Code-Convention.md
│   ├── Commit-Message-Convention.md
│   └── Issue-Convention.md
├── src/
│   ├── src/
│   │   ├── api/        # API 호출 관련 함수
│   │   ├── components/ # 재사용 가능한 컴포넌트
│   │   ├── contexts/   # 전역 상태 관리 컨텍스트
│   │   └── pages/      # 페이지 컴포넌트
```

컴포넌트 가이드

개요

이 문서는 프로젝트에 사용된 주요 **공통 컴포넌트**들의 역할과 props 인터페이스, 내부 구조를 설명합니다.

1. 공통 컴포넌트 목록

컴포넌트 이름	위치	설명
Button	src/components/Button.jsx	기본 버튼
Card	src/components/Card.jsx	대시보드 카드 UI
Modal	src/components/Modal.jsx	팝업 레이어

2. 컴포넌트 상세 설명

Button

```
<Button variant="primary" onClick={handleClick}>
  확인
</Button>
```


상태 흐름 구조 (Zustand)

개요

이 문서는 Zustand 기반 상태 관리 흐름과 store 정의, 주요 전역 상태 변수, 사용 위치 등을 설명합니다.

1. 상태 관리 라이브러리

- 사용: [Zustand](#)
- 도입 이유:
 - Redux 대비 가볍고 보일러플레이트 적음
 - 전역 상태와 로컬 상태 분리가 쉬움

2. 전역 상태 구조

Store 파일: `src/store/projectStore.js`

```
import { create } from 'zustand'

export const useProjectStore = create((set) => ({
  projects: [],
  setProjects: (projects) => set({ projects }),
}))
```


UI 디자인 문서

개요

이 문서는 전체 화면의 흐름과 레이아웃 구조를 설명하기 위해 작성되었습니다.
와이어프레임 이미지, 페이지 전환 흐름, 사용자 인터랙션 등을 포함합니다.

1. 전체 와이어프레임

`./wireframes/main-ui.png`

 [Main Wireframe](#)

2. 페이지별 설명

2-1. 로그인 페이지

- 구성 요소: 이메일 입력, 비밀번호 입력, 로그인 버튼
- UX 고려 사항: 잘못된 입력 시 실시간 에러 메시지 표시

2-2. 대시보드

- 주요 기능: 프로젝트 리스트, 생성 버튼, 최근 알림 표시
- 사용자 흐름: 로그인 → 대시보드 → 프로젝트 상세

3. 사용자 플로우



Login

2. 클러스터링

초기 논의사항

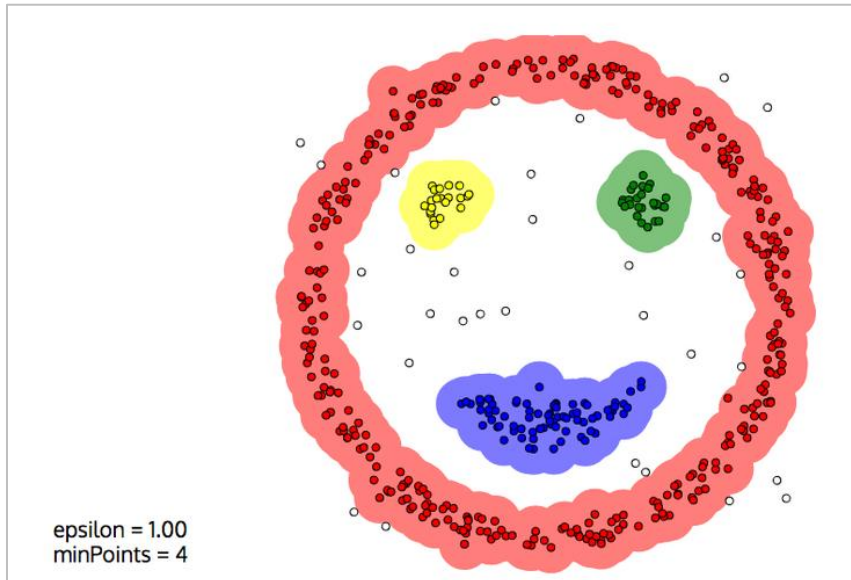
2. 클러스터링

- ✓ 기업분석은 정형화 하기 어려움 → 차트분석으로 진행
- ✓ 다양한 클러스터링 알고리즘을 이용하여 성능이 좋은 클러스터링 모델 이용
(K-Means, 계층형 클러스터링, DBSCAN)
- ✓ 또는 3가지를 모두 구현하여 사용자가 목적에 맞는 클러스터링을 이용할 수 있도록 생각

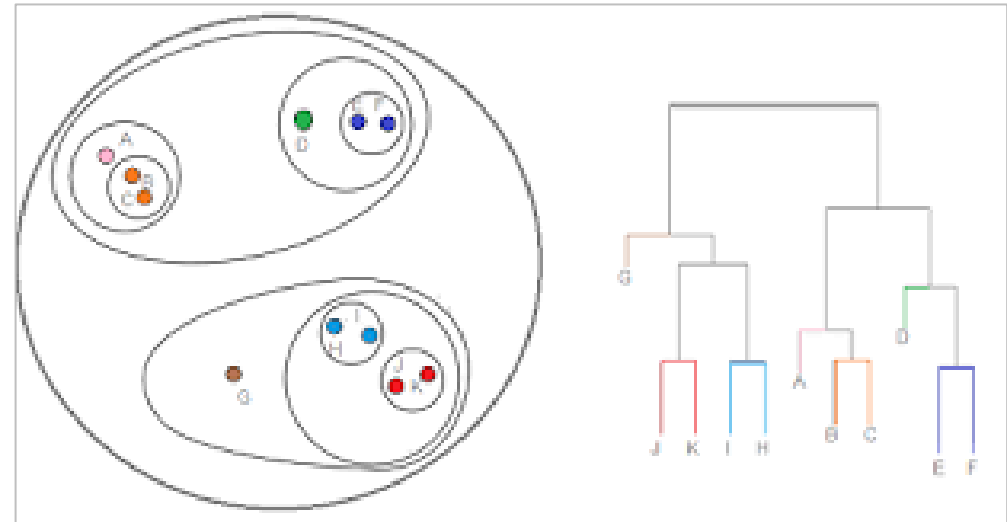


DBSCAN, 계층형 클러스터링 한계

- ✓ 코사인 유사도 행렬이 필요
(코사인 유사도: feature 벡터 간의 방향 유사성)
- ✓ DBSCAN의 경우 파라미터(이웃 반경 ϵ , 최소 이웃 수 MinPts)에 따라 결과가 크게 변할 수 있음 / 또한 직접 클러스터 개수를 설정할 수 없어 시각화에 어려움이 생길 수 있음 → 제외
- ✓ 계층형 클러스터링의 경우 대용량 데이터나 고차원 데이터에는 비효율적 → 제외



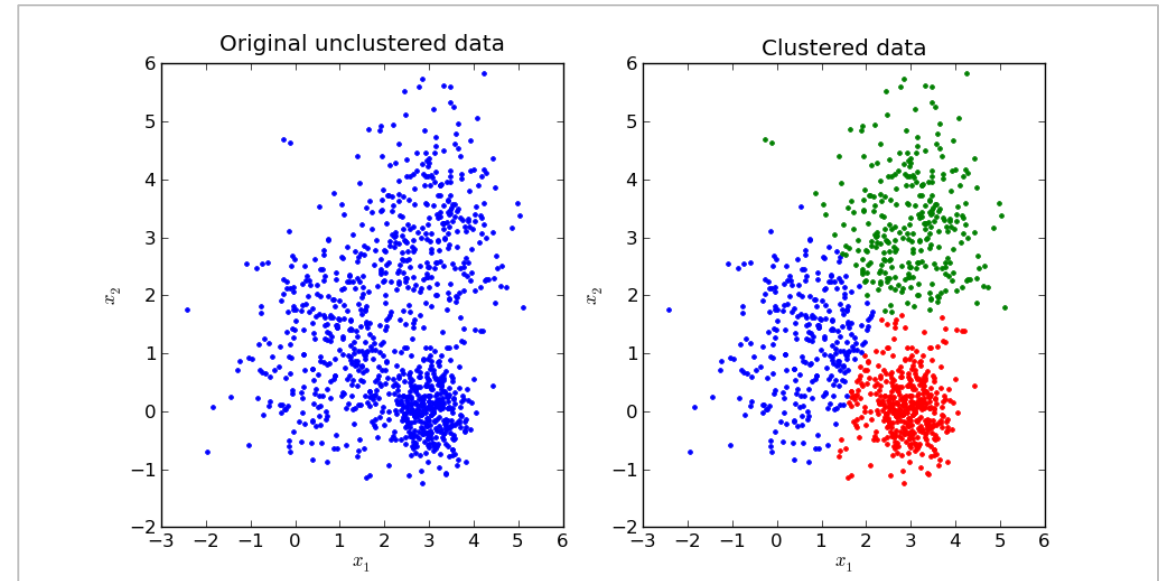
DBSCAN



계층형 클러스터링

K-Means 클러스터링

- ✓ 대중적인 클러스터링 알고리즘
- ✓ K-Means 클러스터링은 각 주식 종목의 feature 벡터 간의 유클리드 거리로 군집을 형성
- ✓ feature를 표준화 한 뒤 이 벡터 자체를 K-Means에 입력하면 됨
- ✓ 시각화 했을 때 비교적 쉽게 사용자가 직관적으로 이해할 수 있음



→ 따라서 K-Means 클러스터링을 우선으로 구현하기로 결정

초기 클러스터링 계획

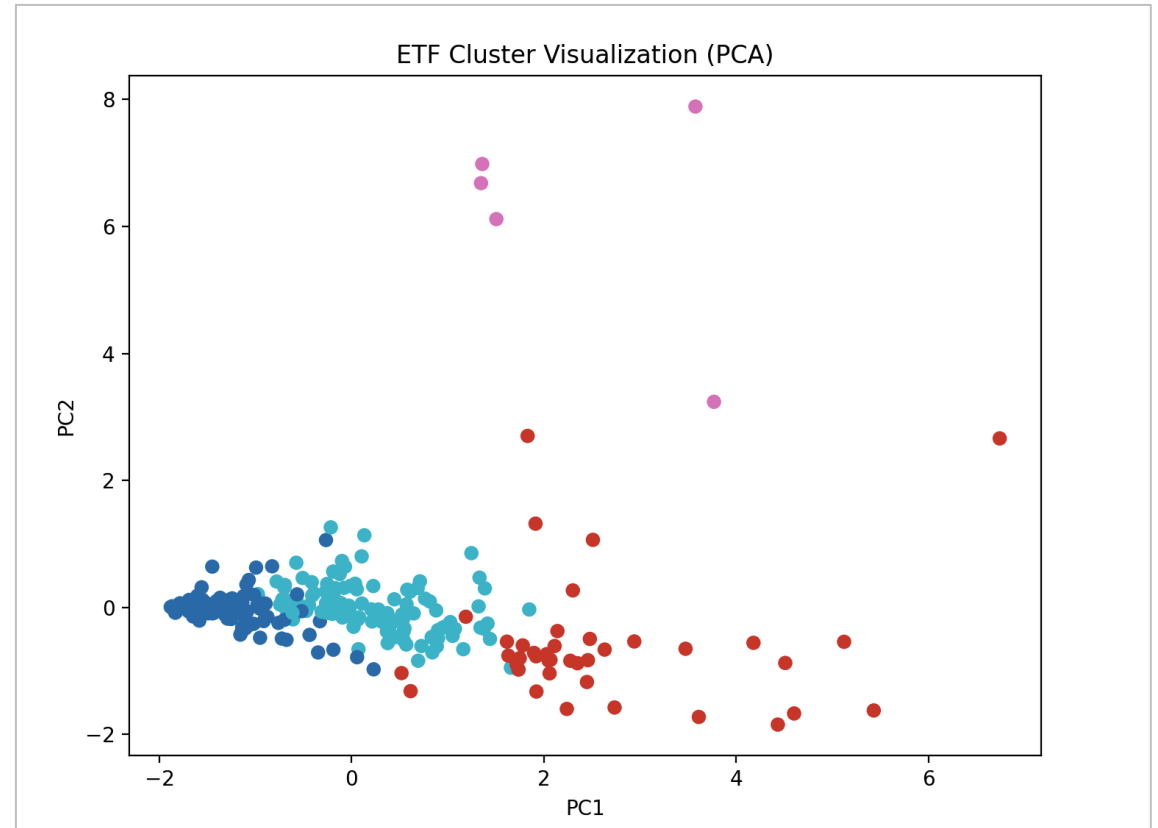
2. 클러스터링

- ✓ 차트 분석에 활용하는 주식 데이터 수집
- ✓ 처음에 주식 종목만을 가지고 진행하려 계획, But 요즘 같은 변동성이 큰 시기에 안전 자산(ex. 금)을 많이 찾게 되는데, 주식이 이 모든 것을 반영하기 어렵다는 피드백 → 따라서 ETF 데이터도 함께 수집
- ✓ 이동평균선, 수익률, 변동성, RSI(상대 강도 지수), MDD(최대 손실률, 최대 낙폭), 거래량 등
→ 시계열 데이터로만 하다 보니 상장 기간의 차이로 인해 가격 지표를 이용한 클러스터링이 적합하지 않을 수 있음
→ 동등하게 지표를 가진 재무제표 같은 지표를 이용하는 것도 검토해보았지만, 추가로 또 다른 데이터를 수집해야 하므로 후순위로 두었음



초기 클러스터링 계획

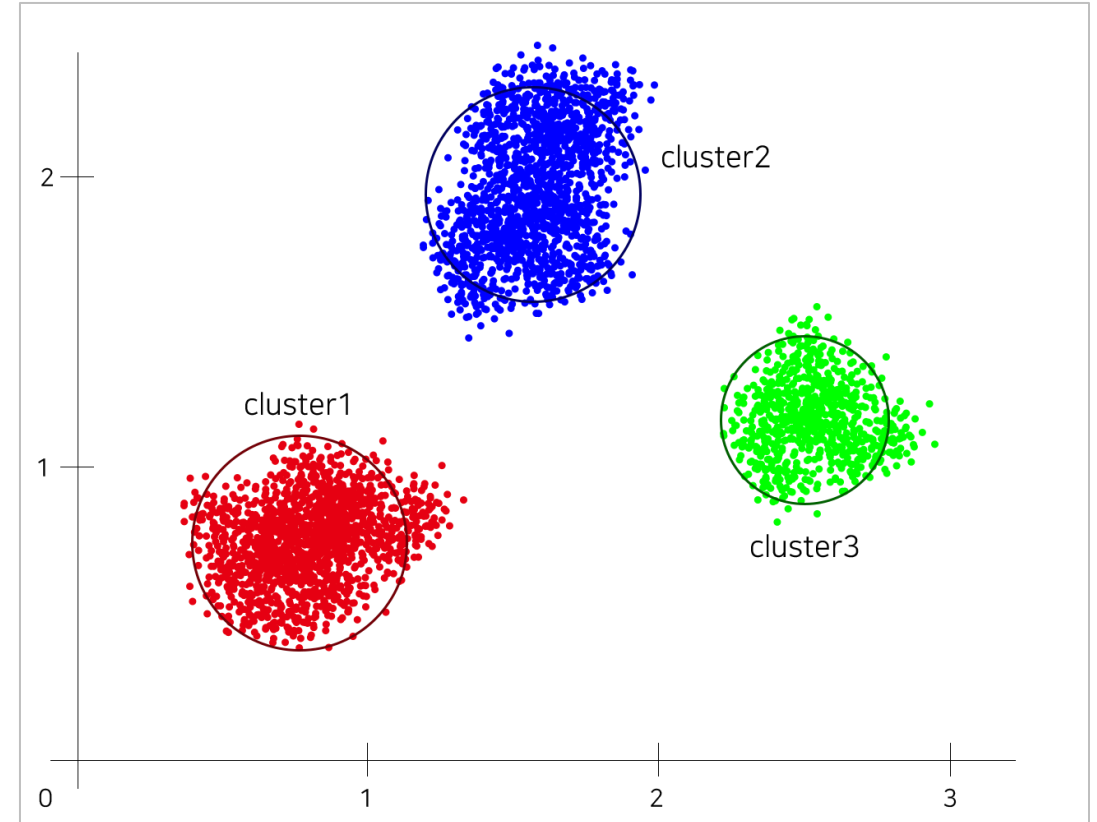
- ✓ 데이터 전처리 (결측값 이상치 처리, feature 표준화)
- ✓ K-Means 클러스터링 수행
- ✓ PCA 차원 축소를 이용한 2차원 좌표 시각화
- ✓ 일단 클러스터링 할 수 있게 아무거나 상관없이 하나 기준으로 만들기
→ 그 후 모델 최적화
- ✓ 초기 수집된 소수의 데이터로 진행한 클러스터
- 종가, 거래량, 현금배당, 주식분할 조정계수



성능 향상 고민

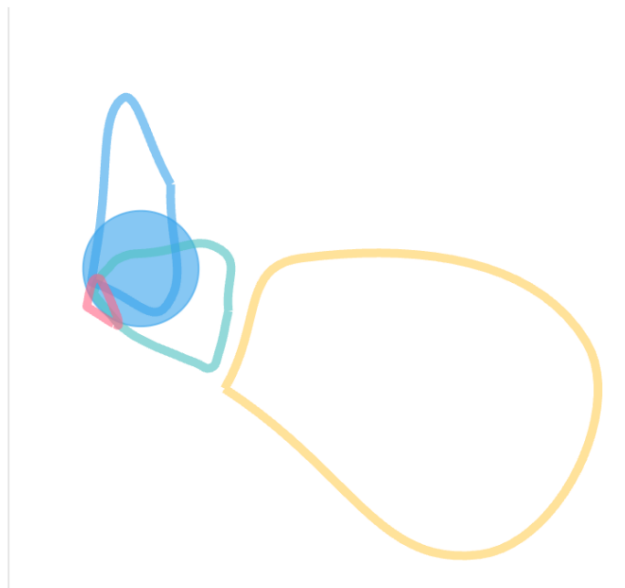
2. 클러스터링

- ✓ 클러스터링 평가 지표를 활용하여 모델 성능 측정
 - 실루엣 계수: 응집도 + 분리도 종합
 - 데이비스-볼딘 지수: 분산 / 거리 비율
 - 칼린스키-하라바츠 지수: 분리도 대비 응집도 비율
- * 응집도: 클러스터 내 분산
- * 분리도: 클러스터 간 분산
- ✓ 클러스터링이 결국 비지도 학습이기 때문에
3가지 평가 지표를 종합하여 성능을 높이는 방향으로 개선



클러스터링

Cluster 0 Cluster 1 Cluster 3
Cluster 2 Cluster 1



데이터 전처리

- ✓ 10,210개의 데이터를 가지고 시작
- ✓ 데이터가 없는 종목 삭제
- ✓ 상장 폐지되어 최신 정보가 없는 데이터 삭제

초기 구현 클러스터링 모델에 약 10,000개의 데이터를 넣었을 때 결과

데이터 전처리 (기간 필터링)

- ✓ 종목별로 상장일자가 달라 feature 계산이 어려운 상황
- ✓ 기간을 동일하게 맞춰서 구성해야 클러스터링 성능이 높아질 것으로 예상
→ 데이터 수집 개수가 최소 1년 이상은 되어야 할 것으로 판단
- ✓ 그 중에서 데이터 보유 기간이 가장 짧은(start date가 제일 최근인 것) 기간 기준으로 모두 기간을 동일하게 처리

```
print(start_date)  
print(end_date)
```

```
2024-05-08 00:00:00  
2025-05-09 00:00:00
```

Feature 생성 및 성능 비교

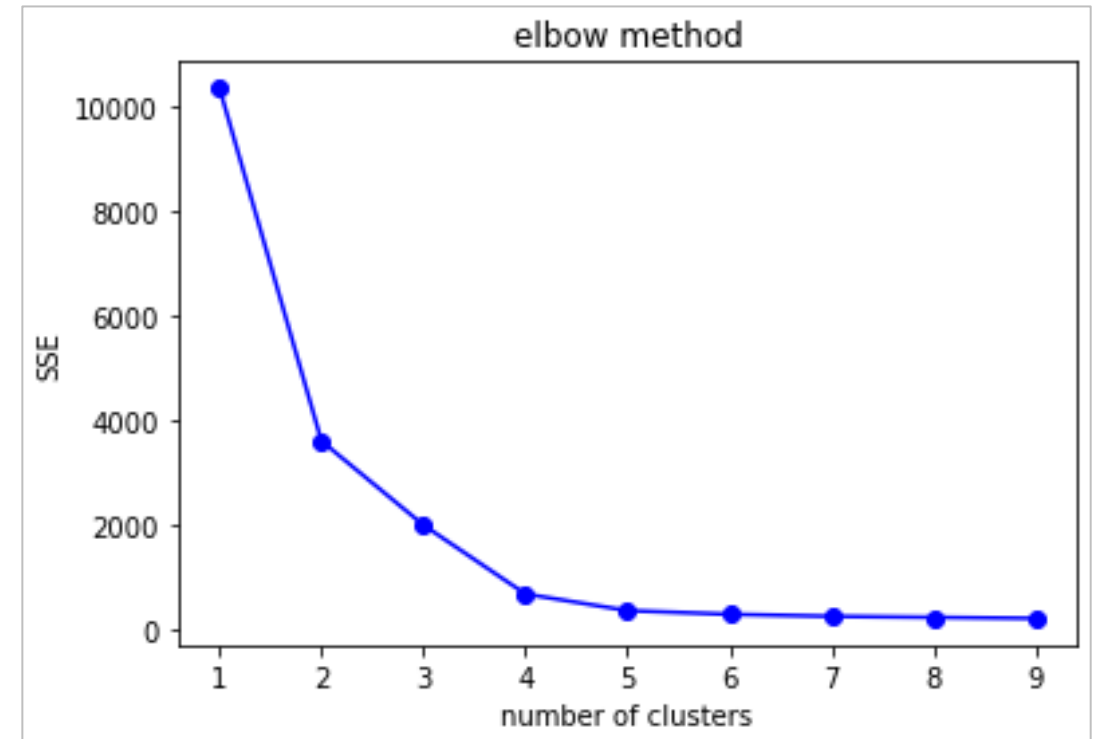
- ✓ 종가와 거래량 지표를 가지고 이동평균, 수익률, 변동성, RSI, MDD, 평균 거래량 계산
- ✓ 주식은 최신 정보의 영향을 많이 받기 때문에 최신 정보를 반영할 수 있는 추가 지표가 필요하다고 판단
→ 6개월 이동평균, 6개월 수익률, 6개월 변동성 지표 추가 계산
- ✓ 총 9개의 feature 생성
- ✓ 1년치 이상 데이터 보유, 6개월 계산 지표 / 3년치 이상, 6개월 계산 지표 / 3년치 이상, 1년 계산 지표 모두 활용해 보았지만 1년치 이상 데이터 보유, 6개월 계산 지표가 클러스터링 평가 지표가 가장 높았음
- ✓ StandardScaler를 이용해 스케일링 (평균 0, 분산 1)
- ✓ 모델 성능을 높이기 위해 PCA로 주성분만 남겨놓은 후 클러스터링을 진행하는 방안도 고민
→ 주성분 분석을 해서 feature를 줄이면 결국 모든 feature 정보를 담지는 못하는 것이므로 기각

성능 향상 고민

엘보우 기법

- ✓ 엘보우 기법 이용하여 최적의 클러스터 수(K) 탐색(SSE)
- ✓ 찾은 K를 Default로 지정하여 이후 사용자가 직접 K를 지정하는 방식 고민
→ 일반 사용자가 K-Means 클러스터링을 모를 수도 있다는 판단 하에
파라미터(K)를 직접 지정하는 방식은 삭제
- ✓ 주식은 투자 가격이 있다 보니 K가 너무 커지면 기초 자산의 크기가
너무 커지는 것이 문제
→ 따라서 범위를 지정해놓고 엘보우 기법 적용
(최대한 4~5의 값이 나올 수 있도록)

→ K=4로 지정

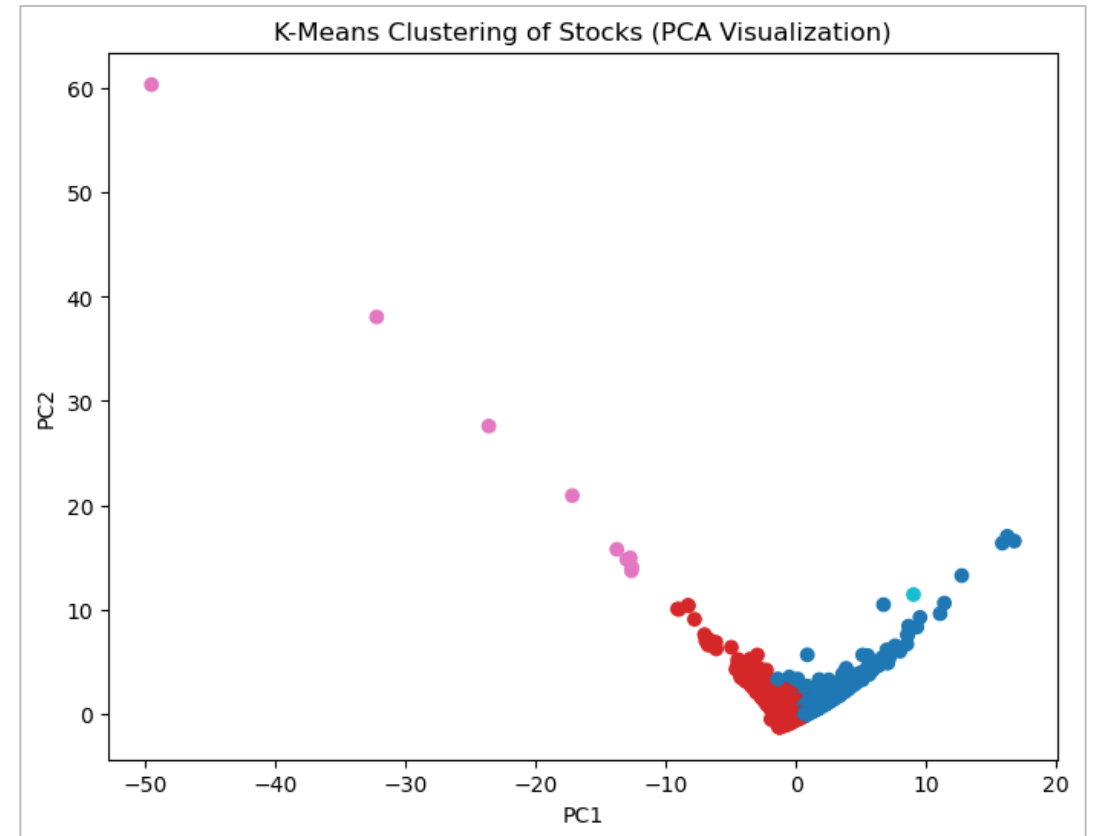


성능 향상 고민

2. 클러스터링

Issue

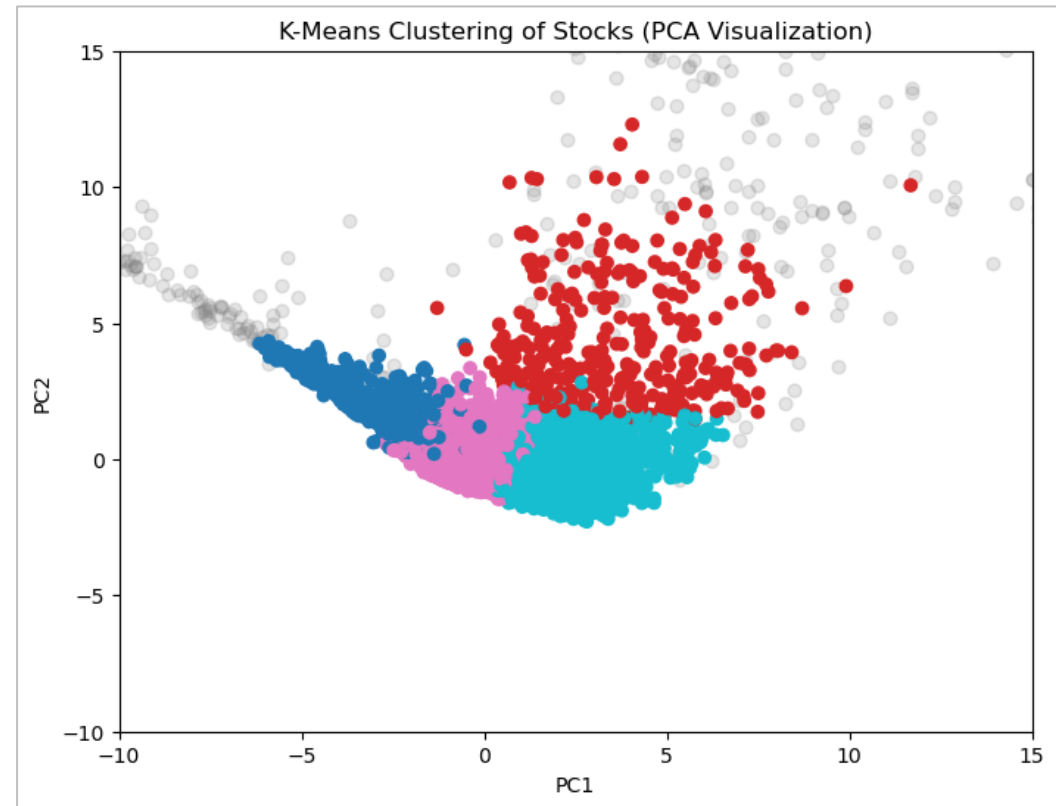
- ✓ 특정 클러스터에 데이터가 쏠리고, 1~2개의 데이터만 포함되어 있는 클러스터 발생
- ✓ 해결을 위해 Isolation Forest 사용 (다변량 이상치 처리 기법)
- ✓ 각 데이터 포인트가 여러 개의 랜덤 결정 트리에서 고립(isolated)되는 데 필요한 경로 길이(depth)를 기반으로 전체의 5% 처리



성능 향상 고민

Issue

- ✓ 이상치 데이터를 완전히 배제하고 진행하면 안 된다는 판단
- ✓ 이상치 데이터를 제외시키기보다 이상치 데이터만의 클러스터를 따로 분류 시켜 시각화 할 필요성 제기



이상치 데이터 회색으로 표시

성능 향상 고민

2. 클러스터링

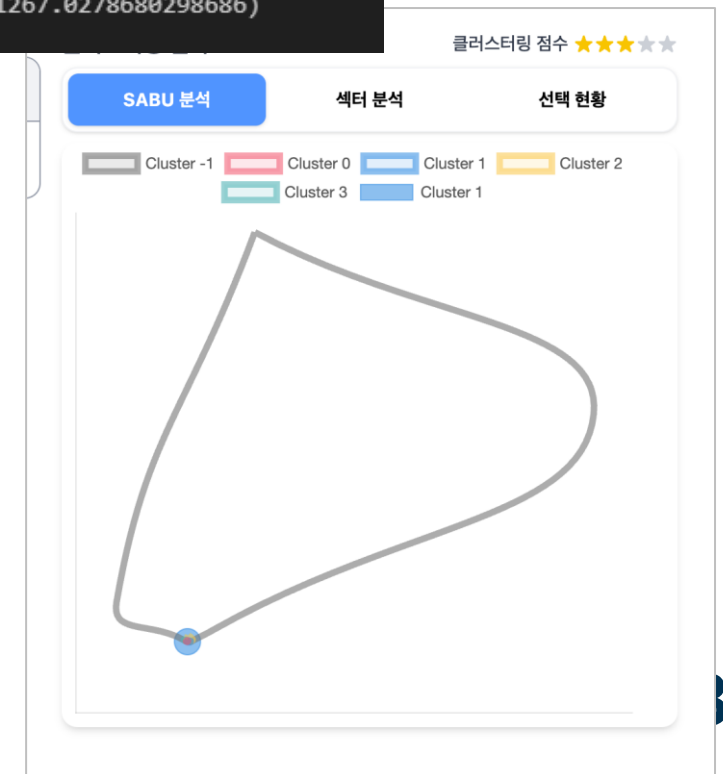
Issue

- ✓ But, 이상치 데이터의 PCA 2차원 축소 값이 정상치 데이터와 차이가 너무 커서 클러스터링 시각화가 잘 되지 않는 문제 발생
- ✓ 처음 해결 방안으로 시각화 축에서 많이 벗어나는 데이터를 제거하는 것으로 결정
→ 개인적인 주관이 들어가 데이터를 제거하면 논리적으로 옳지 않다는 판단

```

np.min(X_pca), np.max(X_pca)
[89] ✓ 0.0s
... (-6.15544036102779, 12.29131458522084)

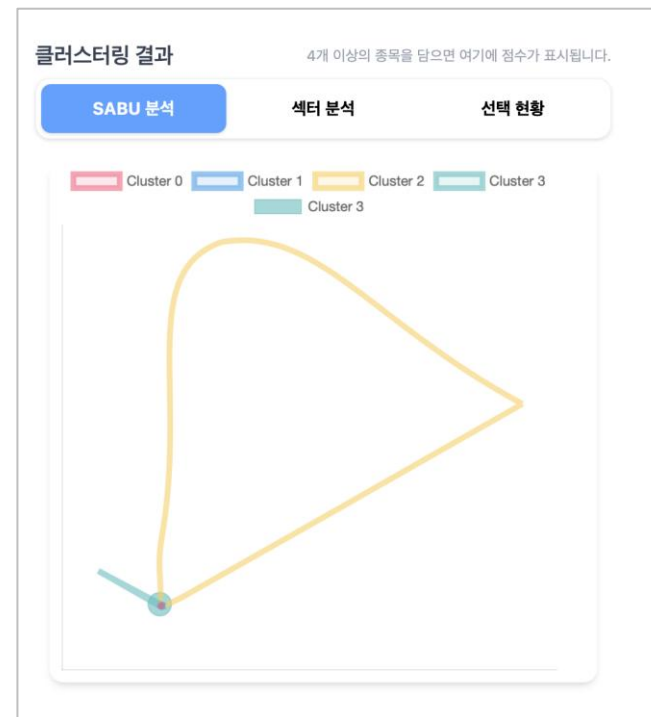
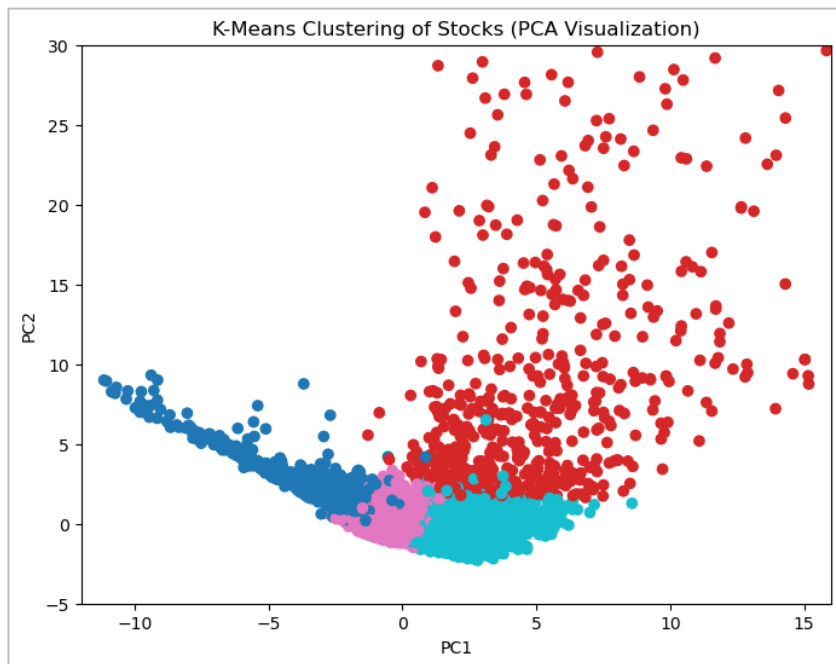
np.min(outlier_pca), np.max(outlier_pca)
[90] ✓ 0.0s
... (-125.78068040888974, 1267.0278680298686)
    
```



성능 향상 고민

Solution

- ✓ 다른 방안 - 정상치 데이터로 학습된 클러스터링 모델에 이상치 데이터를 넣어 각각의 이상치 데이터를 클러스터에 분류

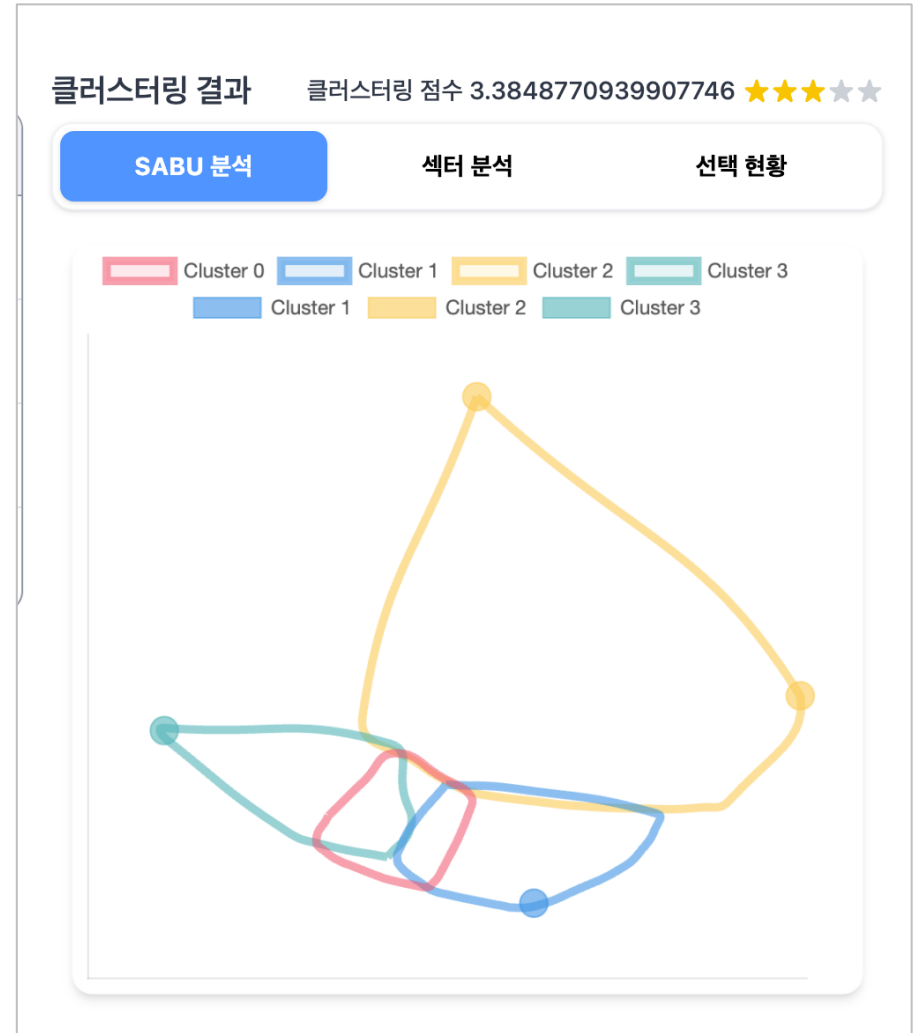


성능 향상 고민

2. 클러스터링

Solution

- ✓ 그 후 정상치의 각 클러스터에서 PC1, PC2 평균을 구해 클러스터에 맞춰 이상치 데이터의 PC1, PC2 값을 대체
- ✓ 이상치 데이터 보존, 만족스러운 시각화 결과 도출



3. 백테스팅

- ✓
 - bt는 살짝 설명
 - 어떻게 돌아가는지랑 전략에 대해 알려줘야 할 듯
 - RSI + monthly / RSI + quarterly / RSI + none
 - buy & hold + monthly / buy & hold + quarterly / buy & hold + none
 - sma cross + monthly / sma cross + quarterly / sma cross + none
 - 구현 내용은 최대한 줄이고
 - 코드 설명할거면 다이어그램으로
 - 에러를 적을거면 (이 기능이 필요한데 오픈소스에 이 기능이 없어서 만들어서 이걸 해결하고 등)

Open Source - bt



내용을 입력해주세요.

작동 플로우



내용을 입력해주세요.

백테스팅 전략 소개



- RSI + monthly / RSI + quarterly / RSI + none
- buy & hold + monthly / buy & hold + quarterly / buy & hold + none
- sma cross + monthly / sma cross + quarterly / sma cross + none

백테스팅 전략 소개

투자 전략 선택 ①

rsi + monthly 🏆

rsi + quarterly

buy-and-hold + quarterly

rsi + none

buy-and-hold + monthly

sma cross + none

sma cr

초기 자산

\$10,000

최종 자산

\$22,377.762

총 수익률 ①

123.77761535234211%

CAGR ①

0%

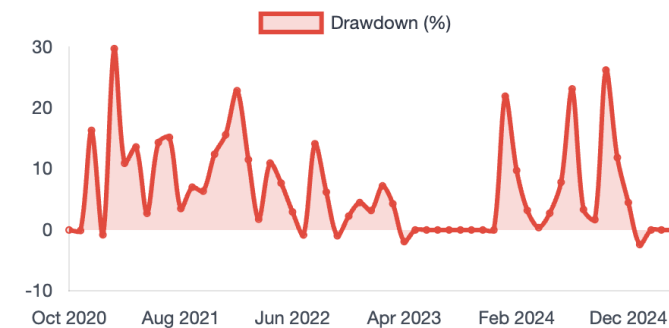
최대 낙폭 ①

76.43%

포트폴리오 성장



최대 낙폭 (Drawdown)



RSI 전략

3. 백테스팅

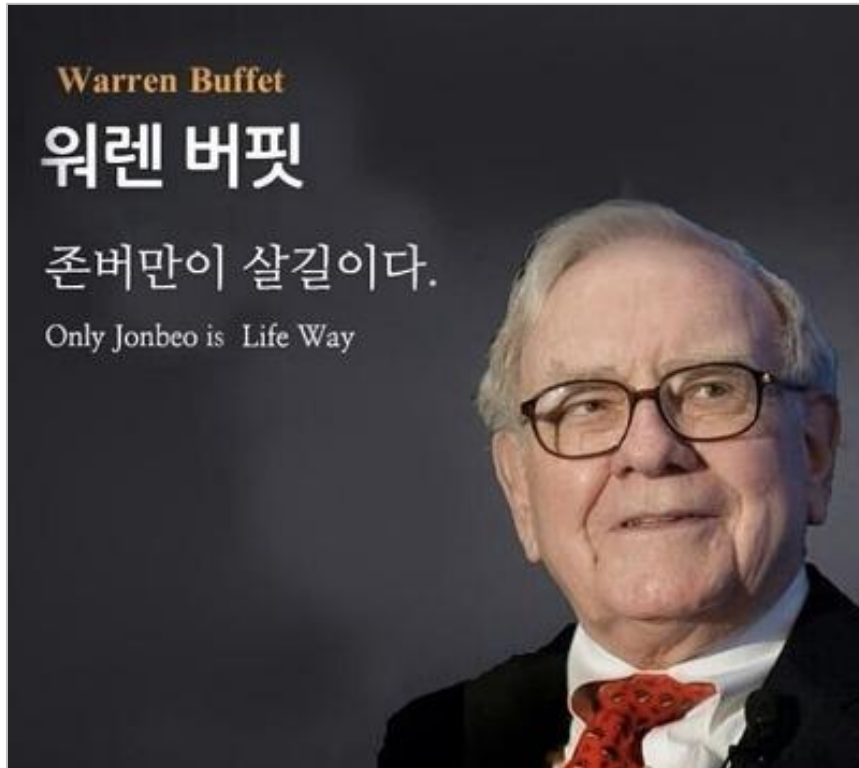
- ✓ **RSI(상대강도지수):** 일정 기간 동안의 상승 변화량과 하락 변화량의 비율
- ✓ 과매수 또는 과매도 상태를 파악하고 매매 시점을 결정

$$RS = \frac{\text{평균 상승}}{\text{평균 하락}}$$

$$RSI = 100 \frac{100}{1+RS}$$



Buy and hold 전략



- ✓ **buy and hold**(매수 및 보유): 주식을 매수한 후 장기 보유하는 전략
- ✓ 주가는 장기적으로 볼 때 지속적으로 상승하는 경향이 있으므로 장기 투자 시 투자 성공 확률이 높다는 것을 이용하는 전략

SMA 전략

- ✓ **SMA(단순이동평균):** 특정 기간 동안 주식의 평균 증가
- ✓ 단기 SMA가 장기 SMA를 상향 돌파할 때 매수 신호, 하향 돌파할 때 매도 신호로 판단
- ✓ 주가의 추세 변화를 예측하고 수익을 얻기 위해 활용



4. Testing

소제목



- gunicorn (서버)
- k6.js (테스팅)



내용을 입력해주세요.



내용을 입력해주세요.



#63 백테스팅 및 API 성능 향상 내용 작성

5. 추가 기능 및 시연

- ✓ - 리더보드
- 감정분석
- ✓ 종목 추천, 섹터 집단 시각화, 분산투자 점수화 등을 여기에 넣을지,
클러스터링에 넣을지

SABU 종목 추천



사용자가 선택한 종목들을 기준으로 분산 투자에 도움이 되는 종목
5가지 추천

전체

Cluster 0

Cluster 1

Cluster 2

Cluster 3

섹터	클러스터	티커	이름	현재가	등락률
🔥 SABU 추천 종목					
Industrials	Cluster 2	AZ	A2Z Cust2Mate Solutions Corp. Common Shares	\$9.31	-0.21%
Consumer Disc...	Cluster 2	BQ	Boqii Holding Limited American Depositary Shares (...	\$2.35	8.29%
Health Care	Cluster 2	CCM	Concord Medical Services Holdings Limited Ameri...	\$8.45	0.48%
Finance	Cluster 2	DISTW	Distoken Acquisition Corporation Warrant	\$0.09	-13.46%
Consumer Sta...	Cluster 2	FAMI	Farmmi Inc. Ordinary Shares	\$1.77	-0.56%

SABU 종목 추천



사용자가 선택한 종목들을 기준으로 분산 투자에 도움이 되는 종목
5가지 추천



추천 종목: ['TSLA', 'AAPL', 'ZWS', 'YANG', 'XXII']

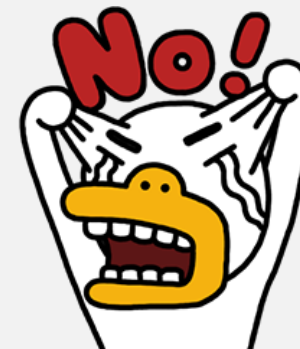
추천 종목: ['TSLA', 'AAPL', 'ZWS', 'YANG', 'XXII']

추천 종목: ['TSLA', 'AAPL', 'ZWS', 'YANG', 'XXII']

추천 종목: ['TSLA', 'AAPL', 'ZWS', 'YANG', 'XXII']

추천 종목: ['TSLA', 'AAPL', 'ZWS', 'YANG', 'XXII']

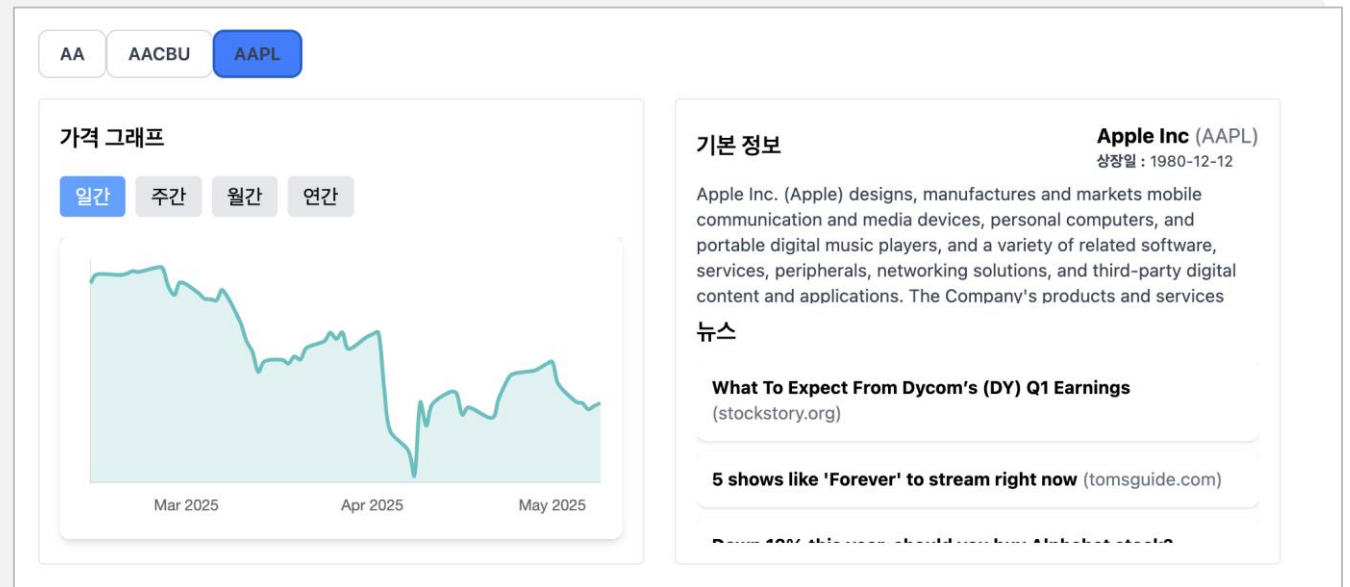
추천 종목: ['TSLA', 'AAPL', 'ZWS', 'YANG', 'XXII']



주가 그래프 및 뉴스 정보 제공

6.추가기능및시연

✓ 내용을 입력하세요.





finBERT API

뉴스

Now or wait? (finance.yahoo.com)

☐ 중립적인 의견이에요.

South32 gains funding for Worsley Alumina

(australianmining.com.au)

☒ 긍정적인 뉴스예요!

✓ 내용을 입력하세요.

클러스터링 결과

클러스터링 점수 3.3848770939907746 ★★☆☆☆

SABU 분석

섹터 분석

선택 현황

Cluster 0 Cluster 1 Cluster 2 Cluster 3
Cluster 1 Cluster 2 Cluster 3

섹터별 거리 시각화

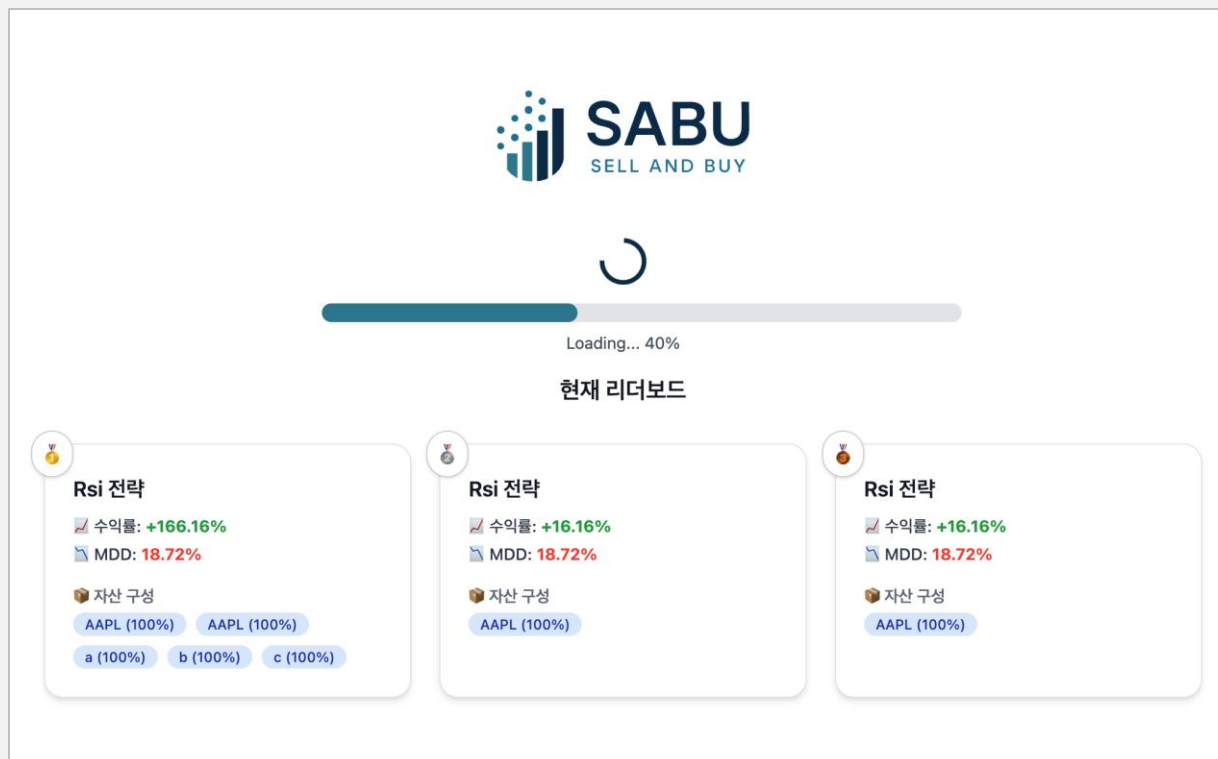
6.추가기능및시연

- ✓ Nasdaq에서 수집된, 섹터 정보가 있는 주식만 가지고 진행
- ✓ 섹터별로 IQR을 이용하여 이상치 처리
- ✓ 섹터를 그룹화 하여 평균 지점의 위치를 2차원으로 시각화



백테스팅 실시간 리더보드

✓ 내용을 입력하세요.



시연

6.추가기능및시연

Q & A