



오픈소스SW기초 6분반

분산 투자 전략 지원, SABU

#주식 #분산 투자

사고8조

정보통계학과	32200472	김동혁
컴퓨터공학과	32211228	김태형
소프트웨어학과	32232597	엄세훈

목차

1

문서화

2

클러스터링

3

백테스팅

4

Testing

5

프론트엔드

6

추가 기능 및 시연

1. 문서화

소제목



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

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

소제목

2. 클러스터링

- ✓ 구현 과정 상세히 설명
- ✓ 고민 내용, 실패 경험, 처리 방법 등

초기 고민 내용

- ✓ 기업분석을 하는 것은 정형화를 하기 어렵기 때문에 차트분석으로 진행하는 것이 좋아 보임
 - ✓ - 지표 기반 유사도 계산 (코사인 유사도: feature 벡터 간의 방향 유사성) / 피어슨 상관계수: 시계열 간의 선형 상관관계
 - 상관계수는 본래 두 변수 간의 선형관계를 측정 -> feature "벡터"로 묶어 비교하면 새로운 유사도(or 거리) 매트릭스가 필요
 - 피어슨 상관계수 = 두 벡터를 평균 0으로 중앙정렬한 뒤 코사인 유사도 계산
 - Feature 벡터 생성: 종목별 이동평균, 변동성, RSI, MACD 등 여러 지표 계산해 하나의 벡터에 담음
 - StandardScaler 등 이용해 스케일링
 - 그 사이에 PCA 등을 이용하여 주성분을 선택해 noise를 줄이는 방법도 있음
 - K-means, 계층적 클러스터링, DBSCAN 등 다양한 클러스터링을 진행해보는 방법
- > 코사인 유사도 행렬 or 피어슨 상관계수 행렬을 거리로 변환하여 클러스터에 적용

초기 고민 내용

- ✓ 또는, 차트분석에 필요한 지표를 고를 수 있게 하여 지표 선택 지표별 클러스터링 결과를 보여주는 방법도 있음
또는, 상관계수를 이용해 분산 투자를 잘 하고 있는지 확인 판단 가능
- ✓ 상위 100개 종목을 클러스터링 하여 back 단에 미리 구축
 - 기술적 지표 추출
 - 데이터 전처리 (표준화)
 - 차원 축소(PCA, 시각화 용도)
 - 엘보우 기법 이용하여 최적의 K값 탐색(SSE)
 - K-means 클러스터링 수행
 - 학습된 클러스터링 모델에 새로운 종목 넣으면 군집 레이블 예측 가능
- ✓ StandardScaler로 정규화하여 단위를 맞춰주기(평균 0, 분산 1)
- ✓ PCA(주성분 분석)를 이용해 먼저 변수를 줄이고 시작하는 방법도 고려

소제목

고민 내용

- ✓ 각 클러스터의 평균 Feature 값 비교해 클러스터를 고수익·고변동 클러스터 / 안정형·저변동 클러스터 등으로 분류?
-> 이런 성향을 띤 클러스터를 가지고 분산 투자를 추천
- ✓ - 일단 클러스터링 할 수 있게 아무거나 상관없이 하나 기준으로 만들기
- 그 후 모델 최적화
- ✓ [클러스터링 구성 전략]
- 기간 데이터 필터링
-> 기간을 맞춰서 구성해야 클러스터링 성능이 높아질 듯
-> 데이터 수집 개수가 최소 6개월 이상은 되어야 할 듯 (or 1년 이상)
-> 그 중에서 처음 날짜가 제일 최근인 것 기준으로 모두 기간을 동일하게 맞춰야 할 듯(전처리)
- ✓ - 이동평균, 수익률, 변동성, RSI(상대 강도 지수), MDD(최대 손실률, 최대 낙폭), 거래량 등 지표를 사용하기로 결정

소제목

고민 내용

- ✓ 클러스터링 평가지표:
 - 실루엣 계수 (응집도 + 분리도 종합, 1에 가까울 수록 좋음)
 - 데이비스-볼딘 지수 (분산 / 거리 비율)
 - 칼린스키-하라바츠 지수 (분리도 대비 응집도 비율)
 - > 비지도 학습이므로 3가지 지표를 모두 종합해서 보아야 할 것 같음
- ✓ - 초기 클러스터를 구현해보니 클러스터가 쏠리는 현상이 발생해서 이상치 or 특이값 처리가 필요해 보인다
 - Isolation Forest의 이상치 판단 기준은 각 데이터 포인트가 여러 개의 랜덤 결정 트리에서 고립(isolated)되는 데 필요한 경로 길이(depth)를 기반
- ✓ - 가격 지표만 이용하는 것은 한계가 있을 수 있으므로 재무제표 같은 지표를 이용하는 것이 어떨지 검토
- ✓ 데이터가 들어있지 않은 파일과 상장 폐지된 종목 등에 대한 전처리도 필요

소제목

2. 클러스터링

고민 내용



1. 1년치 이상 / 6개월치 계산 지표(이동평균, 수익률, 변동성)

- 1년치 이상 데이터 보유 종목: 9277개

- $k = 4$

- 정상치: 8798 / 이상치: 464

- 클러스터링 평가 지표

- 실루엣: 0.35934526367893854

- 데이비스: 1.2018531854639207

- 칼린스키: 2420.7878505083745

2. 3년치 이상 / 1년치 계산 지표(이동평균, 수익률, 변동성)

- 3년치 이상 데이터 보유 종목: 7797개

- $k = 4$

- 정상치: 7403 / 이상치: 390

- 클러스터링 평가 지표

- 실루엣: 0.30196390869338263

- 데이비스: 1.292719271867659

- 칼린스키: 1970.553476542335

이외에 3년치 이상 / 6개월치 계산 지표 등 진행해보았지만 1년치 이상 / 6개월치 계산 지표를 이용하는 것이 제일 좋았음.
최신성을 반영할 수 있어서 인것으로 판단

고민 내용

- ✓ 이상치 데이터를 따로 표현하지 말고
학습 시킨 클러스터에 넣어서 예측 분류를 시키자
그렇게 해서 일단 시각화
PCA로 시각화 했을 때 y축 너무 많이 벗어나는 종목들은 제외시켜야 할 듯

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

Open Source - bt



내용을 입력해주세요.

작동 플로우



내용을 입력해주세요.

백테스팅 전략 소개



내용을 입력해주세요.

백테스팅 전략 소개

투자 전략 선택 ⓘ

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 ⓘ

123.77761535234211%

CAGR ⓘ

0%

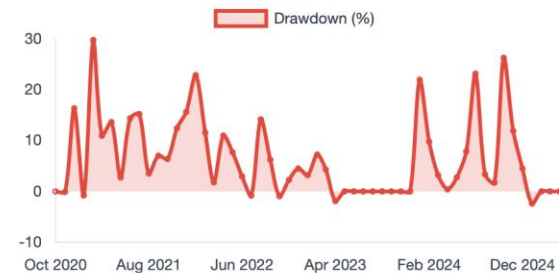
최대 낙폭 ⓘ

76.43%

포트폴리오 성장



최대 낙폭 (Drawdown)



4. Testing

소제목

4. Testing



- gunicorn (서버)
- k6.js (테스팅)



내용을 입력해주세요.



내용을 입력해주세요.

5. 프론트엔드

기능 수정이 가능하다면 작성
안 된다면 제외

6. 추가 기능 및 시연

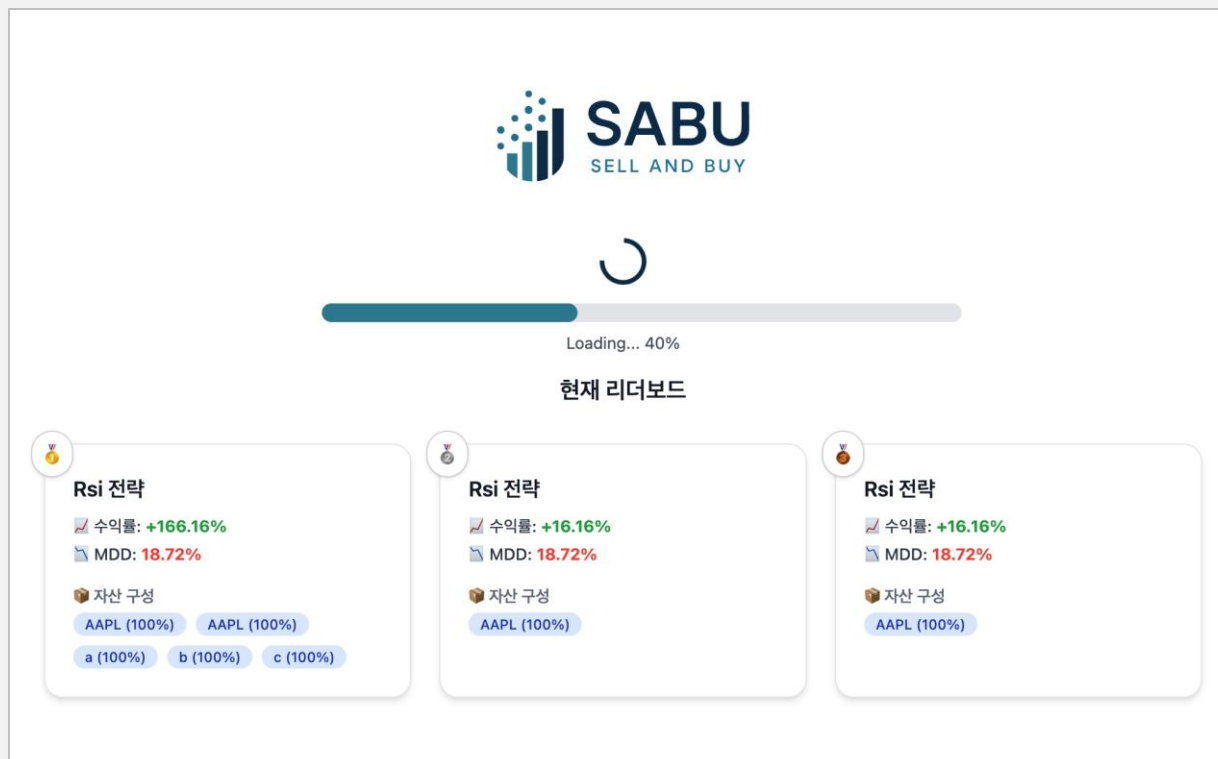
소제목

6.추가기능및시연

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

백테스팅 리더보드

✓ 내용을 입력하세요.



뉴스 감정 분석

✓ 내용을 입력하세요.

뉴스

Now or wait? (finance.yahoo.com)

☐ 중립적인 의견이에요.

South32 gains funding for Worsley Alumina
(australianmining.com.au)

☒ 긍정적인 뉴스예요!

시연

6.추가기능및시연

Q & A