

HanS:hop

Online shopping malls
구디 아카데미 빅데이터 2기 조한서

Chapter. 1
소 개

Chapter. 2
개 발 환 경

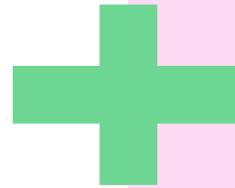
Chapter. 3
기 능

Chapter. 4
느 낀 점

HanS:hop 프로젝트 소개



Online Shopping



설명

- 빅데이터를 활용한 웹 쇼핑몰 기획 / 제작
- 데이터에 따른 유동적인 웹 운영
- 게시판, 댓글, 리뷰 구현
- REST API 적용

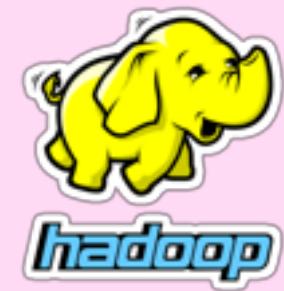
요구 사항

- 하둡을 이용, 데이터 분석 및 시각화할 것
- 분석할 데이터는 1,000건 이상일 것
- 개발 언어는 자바를 사용할 것

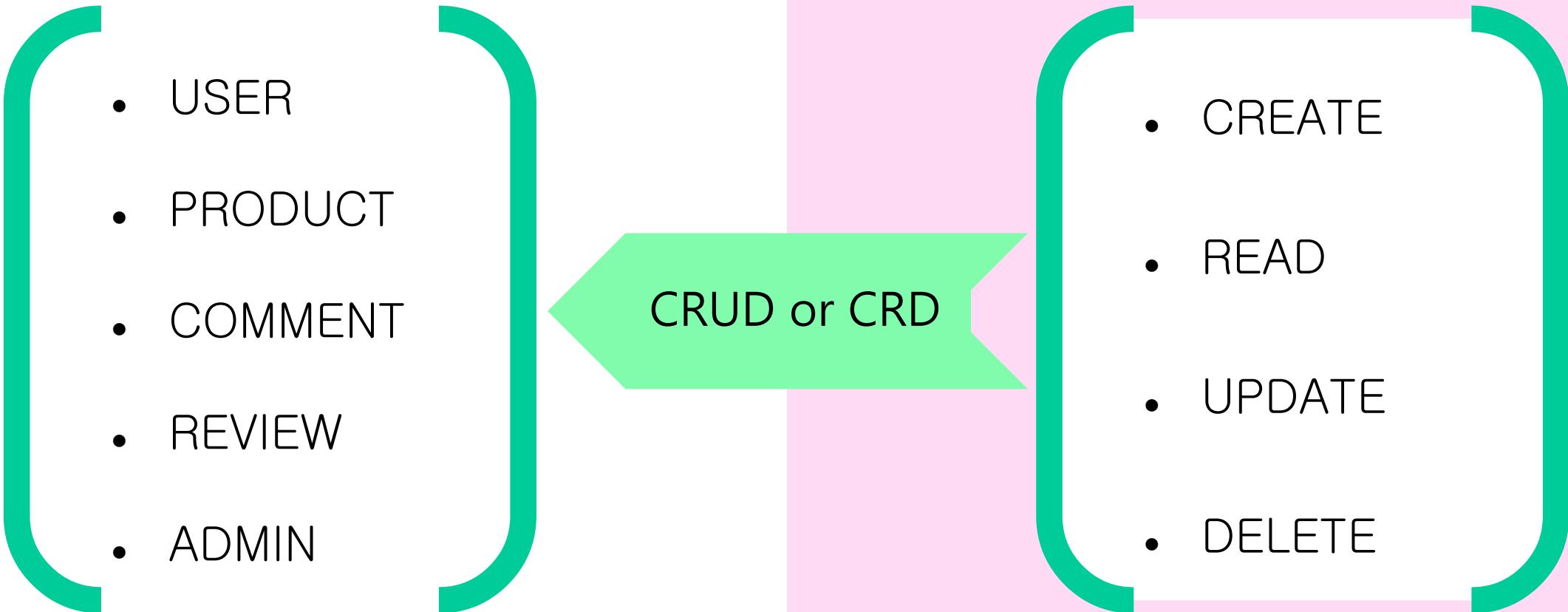
HanS:hop 프로젝트 일 정

SUN	MON	TUE	WED	THR	FRI	SAT
		1	2	3	4	5
6	7	8	9	10	11	12
				분석 / 기획	DB 분석 / 설계	
13	14	15	16	17	18	19
	UI 설계			UI 디자인		
20	21	22	23	24	25	26
		UI 디자인		기능 구현		
27	28	29	30	31
		기능 구현		점검 및 수정		

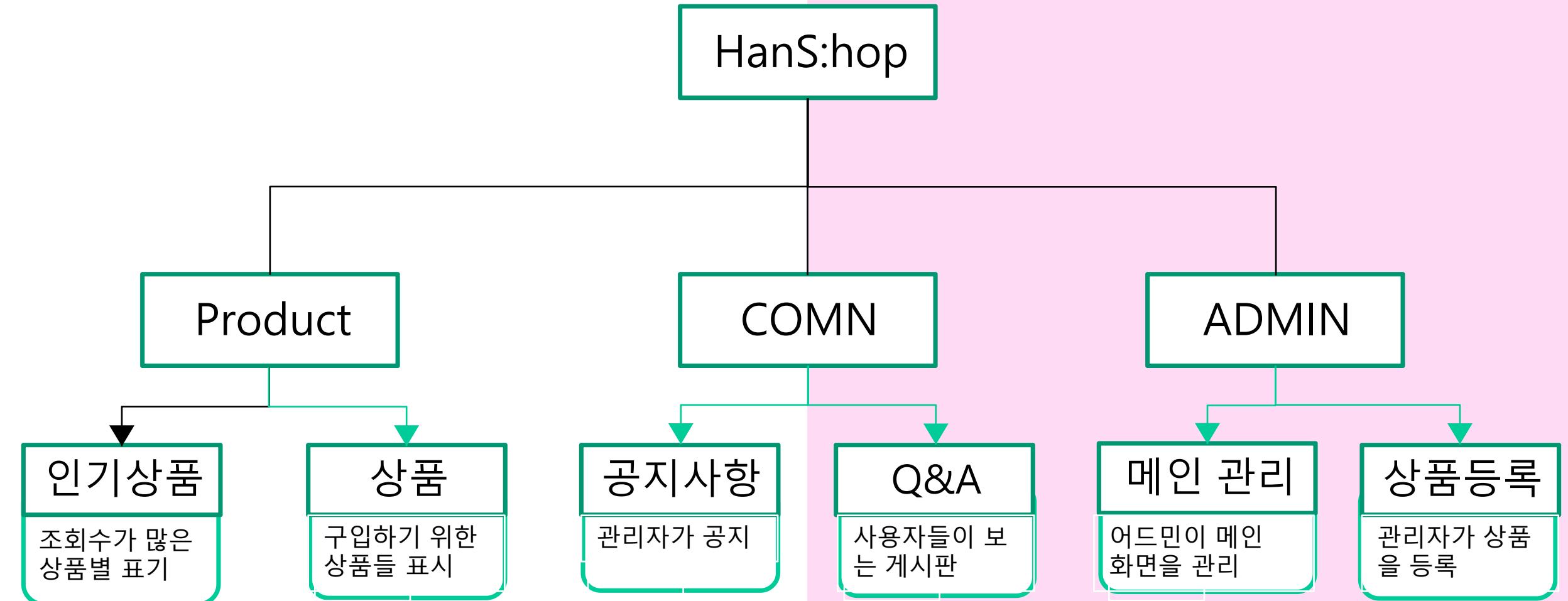
HanS:hop
개발환경



버전
JAVA:1.8
SpringFrameWork:4.3.17
Aspecj : 1.8.13
Slf4j : 1.6.6



HanS:hop 시나리오



HanS:hop USER / 이메일 인증

```
@Override
public ResponseEntity<String> insert(UserVO user) {
    user.insertDb(); //로컬된 값을 합쳐주기, 이메일 풀 생김
    try {
        if(idPattern(user)==false) {
            return new ResponseEntity<String>("아이디를 규칙에 맞게 다시 입력해주세요", HttpStatus.ACCEPTED);
        }
        if(!selectid(user).equals("비밀복")) {
            return new ResponseEntity<String>("비밀복은 아이디입니다.", HttpStatus.ACCEPTED);
        };
        if(user.checkPw()==false) {
            return new ResponseEntity<String>("비밀번호를 확인해주세요", HttpStatus.ACCEPTED);
        }
        Map<String, Object> param = MapUtil.makeParam("insert", "user.insert");
        user.setKey();
        di.call(param, user);

        MailHandler sendMail = new MailHandler(mailSender);
        sendMail.setSubject("[HanS:hop 이메일 인증]");
        sendMail.setText(
            new StringBuffer().append("<div style='border:1px solid black; text-align:center;'><h1>메일인증</h1><h3>안녕하세요\nHanS:hop에 계속 가입하실려면 "
                + "아래 링크를 눌러주세요.</h3>").append("<a href='http://gudi.iptime.org:11100/email/" + user.getId() + "/key/" +
                user.getKey() + "' target='_blank'>이메일 인증 확인</a></div>").toString());
        sendMail.setFrom("trullo393@gmail.com", "HanS:hop");
        sendMail.setTo(user.getEmail());
        sendMail.send();
        return new ResponseEntity<String>("이메일발송", HttpStatus.OK);
    }catch (Exception e) {
        e.printStackTrace();
        return new ResponseEntity<String>("가입실패", HttpStatus.ACCEPTED);
    }
}
```



```
//이메일 인증
@RequestMapping(value="/email/{id}/key/{key}",method=RequestMethod.PUT)
public ResponseEntity<HashMap<String, Object>> keyMatch(UserVO user) throws Throwable{
    return usi.matchKey(user);
}
```

HanS:hop **PRODUCT** / 조회수

no	productNo	userType	userNo	date
1	1	회원	1	2018-08-30 14:37:19
2	1	회원	1	2018-08-30 14:37:19
3	1	회원	1	2018-08-30 14:37:19
4	13	회원	1	2018-08-30 14:37:19
5	6	회원	1	2018-08-30 14:37:19
6	15	회원	1	2018-08-30 14:37:19
7	15	회원	1	2018-08-30 14:37:19
8	15	회원	1	2018-08-30 14:37:19
9	15	회원	1	2018-08-30 14:37:19
10	15	회원	1	2018-08-30 14:37:19
11	15	회원	1	2018-08-30 14:37:19

HanS:hop

ADMIN / 이벤트박스 관리

이벤트박스 예시

1 2 3
4 5 6

1 ▼ 번째

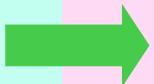
이동할 URL주소

변경할 이미지

파일 선택 선택된 파일 없음

수정

미리보기



HanS:hop USER / 이메일 인증

```
@Override
public ResponseEntity<String> insert(UserVO user) {
    user.insertDb(); //로컬된 값을 합쳐주기, 이메일 풀 생김
    try {
        if(idPattern(user)==false) {
            return new ResponseEntity<String>("아이디를 규칙에 맞게 다시 입력해주세요", HttpStatus.ACCEPTED);
        }
        if(!selectid(user).equals("비밀번호")) {
            return new ResponseEntity<String>("비밀번호가 틀렸습니다.", HttpStatus.ACCEPTED);
        };
        if(user.checkPw()==false) {
            return new ResponseEntity<String>("비밀번호를 확인해주세요", HttpStatus.ACCEPTED);
        }
        Map<String, Object> param = MapUtil.makeParam("insert", "user.insert");
        user.setKey();
        di.call(param, user);

        MailHandler sendMail = new MailHandler(mailSender);
        sendMail.setSubject("[HanS:hop 이메일 인증]");
        sendMail.setText(
            new StringBuffer().append("<div style='border:1px solid black; text-align:center;'><h1>메일인증</h1><h3>안녕하세요!\nHanS:hop에 계속 가입하실려면 "
                + "아래 링크를 눌러주세요.</h3>").append("<a href='http://gudi.iptime.org:11100/email/" + user.getId() + "/key/" +
                user.getKey() + "' target='_blank'>이메일 인증 확인</a></div>").toString());
        sendMail.setFrom("trullo393@gmail.com", "HanS:hop");
        sendMail.setTo(user.getEmail());
        sendMail.send();
        return new ResponseEntity<String>("이메일발송", HttpStatus.OK);
    }catch (Exception e) {
        e.printStackTrace();
        return new ResponseEntity<String>("가입실패", HttpStatus.ACCEPTED);
    }
}
```



```
//이메일 인증
@RequestMapping(value="/email/{id}/key/{key}",method=RequestMethod.PUT)
public ResponseEntity<HashMap<String, Object>> keyMatch(UserVO user) throws Throwable{
    return usi.matchKey(user);
}
```

HanS:hop

ADMIN / 유저 관리

아이디	이름	가입일	차단여부
user	테스터	2018-09-04	차단하기
user1	테스터	2018-09-04	차단하기
user2	테스터	2018-09-04	차단해제
user1234	test	2018-09-04	차단하기

gudi.iptime.org:11100 내용:

운영자에 의해 차단되었습니다

확인

Login or Create

user2

.....

아이디찾기 비밀번호찾기

LOGIN

HanS:hop REVIEW / 리뷰 등록

REVIEW | 문의글 혹은 악의적인 비방글은 무통보 삭제됩니다 :)

종류

★★★★★ 아주좋아요!!

리뷰 등록

5 Stars

(0)

4 Stars

(0)

3 Stars

(0)

2 Stars

(0)

1 Stars

(0)

Nan

0개 리뷰 평점

리뷰목록

HOME SHOP ABOUT BOARD

REVIEW | 문의글 혹은 악의적인 비방글은 무통보 삭제됩니다 :)
리뷰를 남겨주세요(로그인시 이용가능)

☆☆☆☆☆ 별점을 주세요!!

리뷰 등록

5.0

1개 리뷰 평점

5 Stars

(1)

4 Stars

(0)

3 Stars

(0)

2 Stars

(0)

1 Stars

(0)

리뷰목록

★★★★★ 아주좋아요!!
등록!!

작성자
관리자
작성일
2018.09.05
작성자 등급
회원



상품종류관리

대분류

상의
하의

중분류

티셔츠
셔츠
스웨터
데님
치노팬츠

소분류

단가라티셔츠
맨투맨티셔츠
박스티셔츠
드레스셔츠
옥스포드셔츠
데님셔츠
카디건
풀오버
스키니
스트레이트

삭제

추가할 상품종류

종류 ▼

상세조건1 ▼

상세조건2 ▼

추가

중분류로 상품게시판이 변동

HanS:hop

ADMIN / 상품 추가

상품추가

상의 ▼ 셔츠 ▼ 옥스포드셔츠 ▼

상품명 옥스포드셔츠

가격 20000

색깔 Black

사이즈
S M L

details 좋은옷

viewImage 파일 선택 선택된 파일 없음
ViewImage는 3개까지 가능합니다

moreInfo 파일 선택 선택된 파일 없음
moreInfo이미지는 1개만 가능합니다

상품등록

리셋

HanS:hop **A D M I N** / 데이터 분석

```

public ResponseEntity<HashMap<String, Object>> saveData(String inDate, String inTime) throws Throwable{
    Map<String, Object> param = MapUtil.makeParam("selectList", "productviews.allList");
    List<HashMap<String, Object>> list = (List<HashMap<String, Object>>) di.call(param);
    for(int i=0;i<list.size()/600;i++) {
        List<HashMap<String, Object>> list2 = new ArrayList<HashMap<String, Object>>();
        for(int j=(i*600);j<((6*(i+1))*100);j++) {
            list2.add(list.get(j));
        }
        Job job = Job.getInstance(conf, "test");
        FileSystem file = FileSystem.get(conf);
        FSDataOutputStream fsos = file.create(new Path("/insert/" + inDate + "/" + inTime + "/data" + i + ".csv")); // 출력 객체 생성
        String result = "";
        List<String> textlist = new ArrayList<String>();
        //System.out.println( String.format("키 : %s, 값 : %s", key, map.get(key)) ); 참고 : http://stove99.tistory.com/96
        for(int k=0;k<list2.size();k++) {
            for(String key : list2.get(k).keySet()) {
                if(key.equals("date")) {
                    SimpleDateFormat d = new SimpleDateFormat("yyyy/MM");
                    result += d.format(list2.get(k).get(key));
                }
                else {
                    result += String.format("%s", list2.get(k).get(key)).replace(", ", "/");
                }
                result += ",";
            }
            result = result.substring(0, result.length()-1);
            System.out.println(result.length());
            result+="\n";
        }
        System.out.println(textList.size());
        fsos.writeUTF(result); // 내용작성
        fsos.close(); // 출력 객체 종료
        file.close();
    }
    System.out.println("■");
    return null;
}

```

전체분석

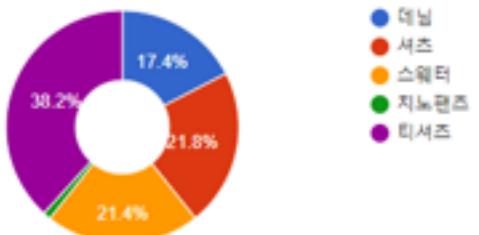
쇼핑몰에 상품조회로 인한 데이터를 전부 분석합니다.
분석할 데이터가 많아 오래걸릴수 있습니다.

세부분석

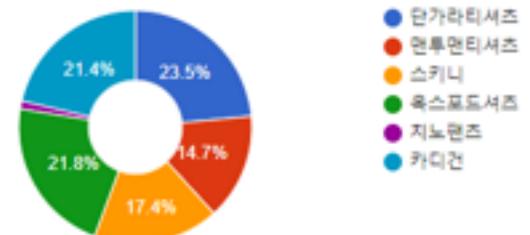
연도와 월을 지정해 상품조회로 인한 데이터를 분석합니다.

----년 --월 검색

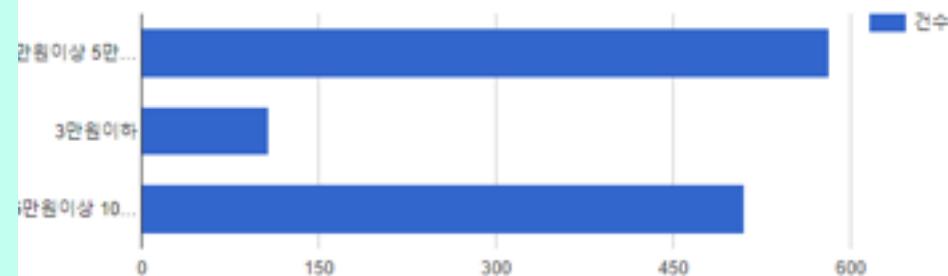
중분류별 분석



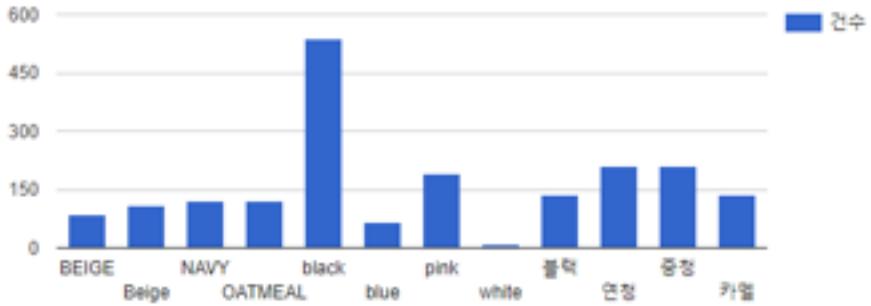
소분류별 분석



가격대별 분석



인기색상 분석



감사합니다

Git으로

<https://github.com/trullo/project.git>