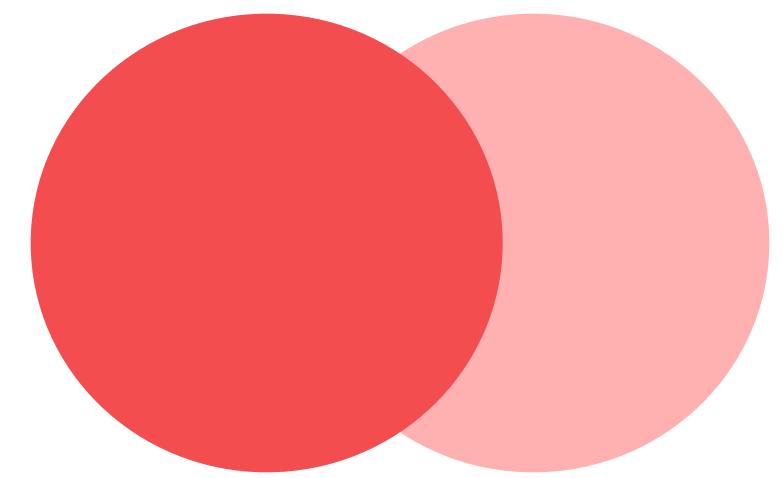


서울시 대학 근처 업종 추천 및 과잉 분석 시스템

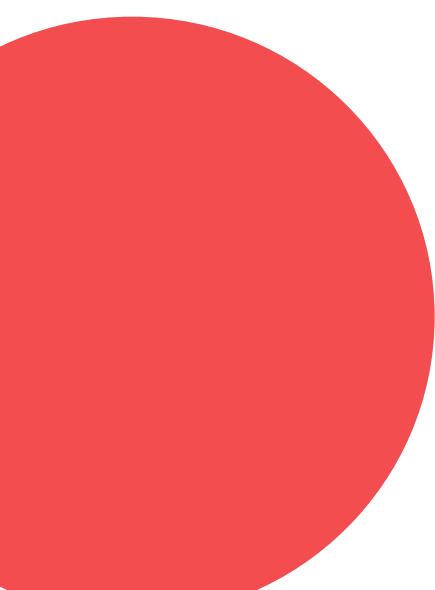
3조

고도희 60221114
김지웅 60222083
노종욱 60221349
문장훈 60191652
민영은 60211571



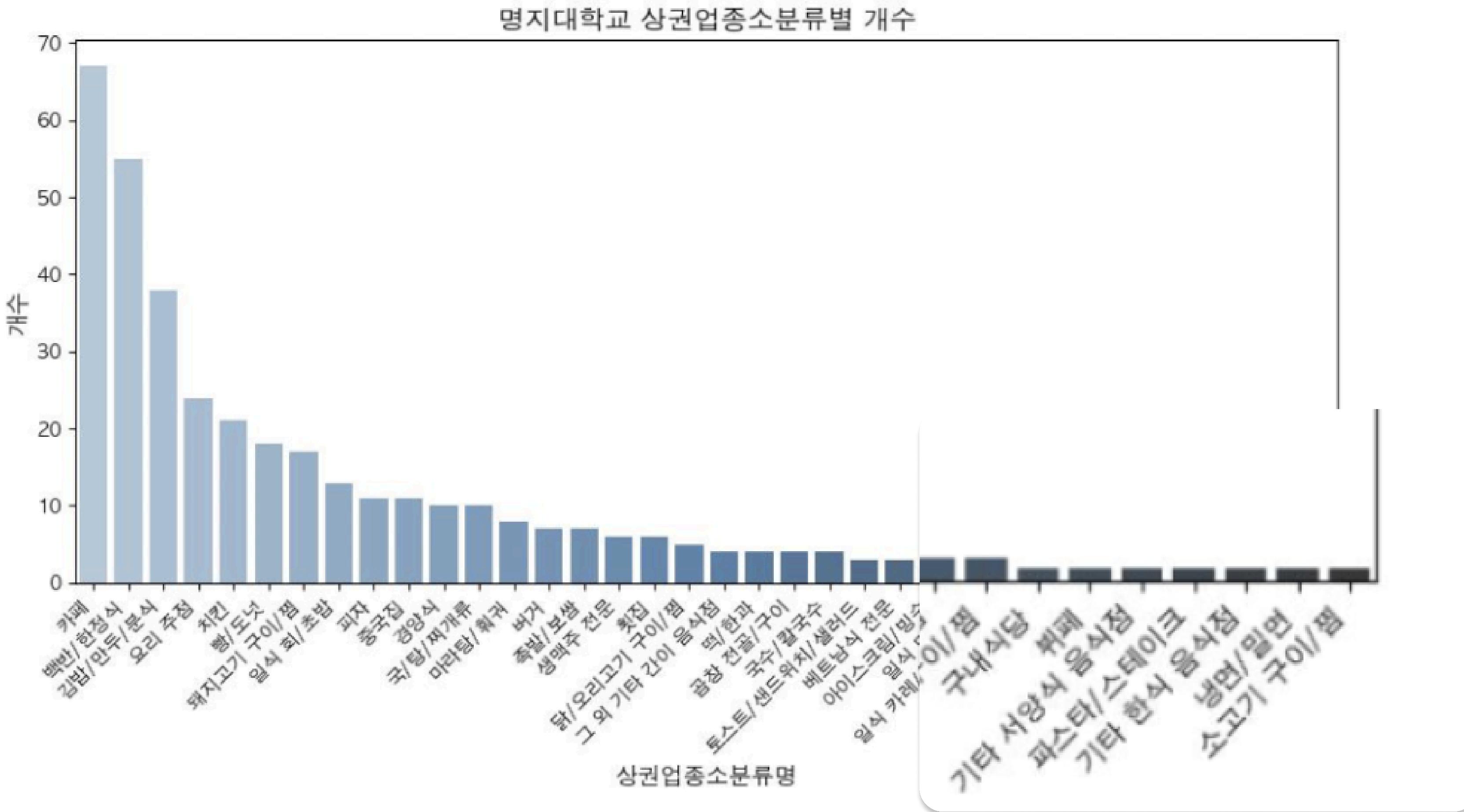
목차

- section 1 주제 소개
- section 2 데이터 수집
- section 3 데이터 전처리
- section 4 서울시 대학 업종 추천 및 과잉 분석



1. 주제 소개

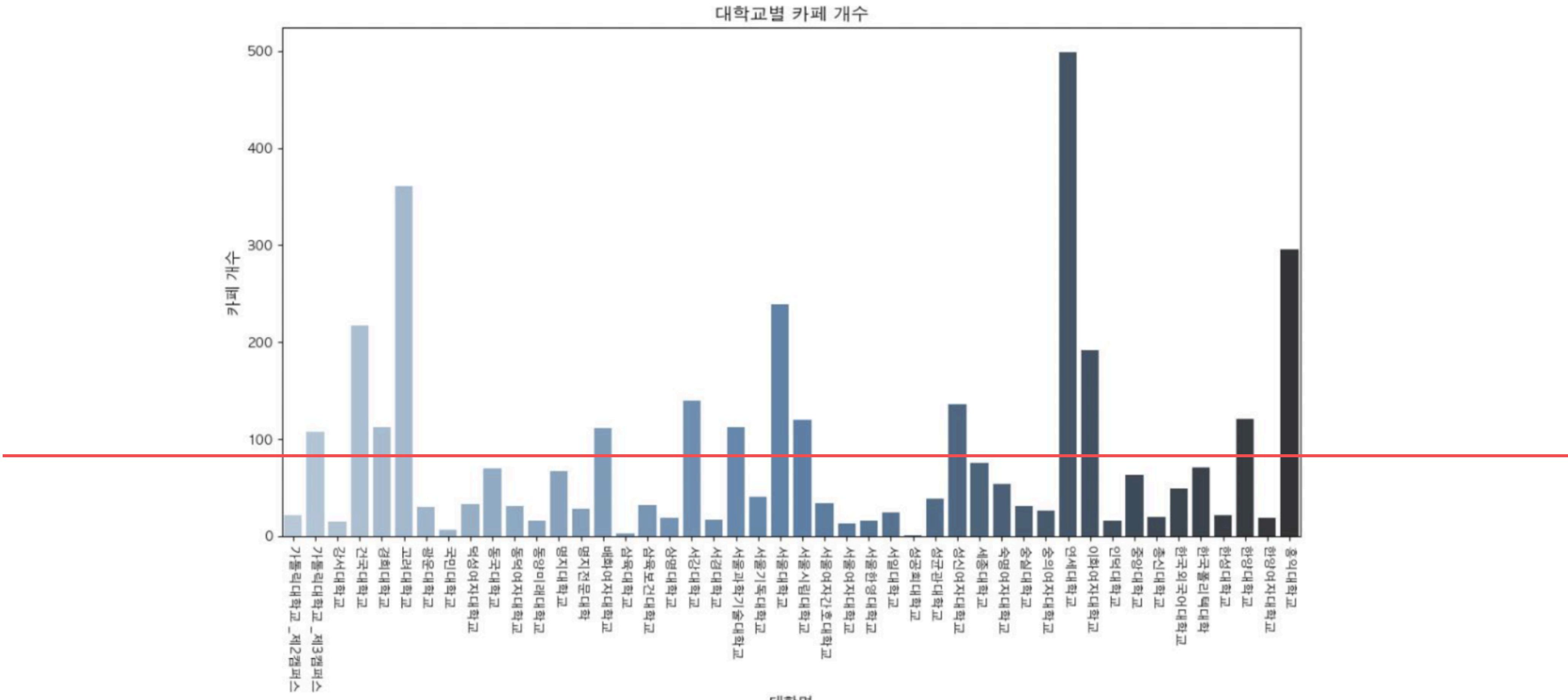
BDP Term Project



소고기/구이/찜을 추천해주고 싶은게 아님

1. 주제 소개

BDP Term Project

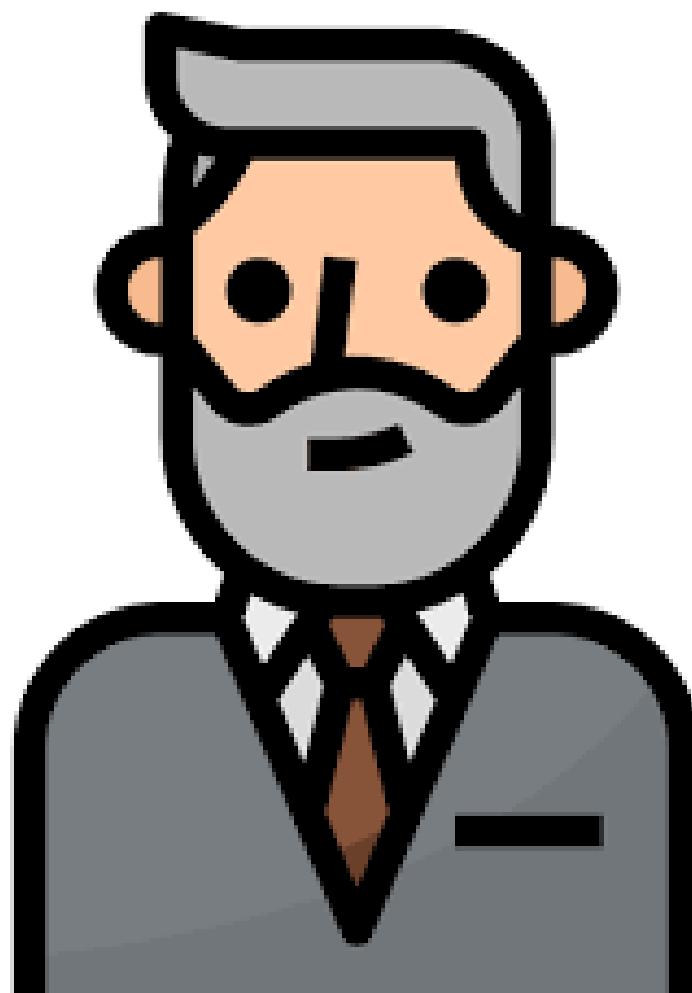


대학별 카페 개수의 평균을 계산해서 평균보다 낮다면 추천

1. 주제 소개

BDP Term Project

명지대학교 근처에 대학생들이
선호하는 음식점을 차리고 싶어



명지대학교를 입력하면, 명지대학교의 카페개수가 평균보다 작기 때문에 카페는 추천대상이된다.

대학생들이 선호하는 음식점은 어떤 유형인가?

학생들의 선호도가 빠르게 변화

상권 트렌드 급격히 변화

다른 대학 상권에서 인기 있는 유형은 무엇인가?

대학가 근처에 어떤 음식점을 차려야 할까?

대학가 상권 분석의 필요성

경험, 직감, 주변 사례를 통한 창업은 실패 확률이 높음

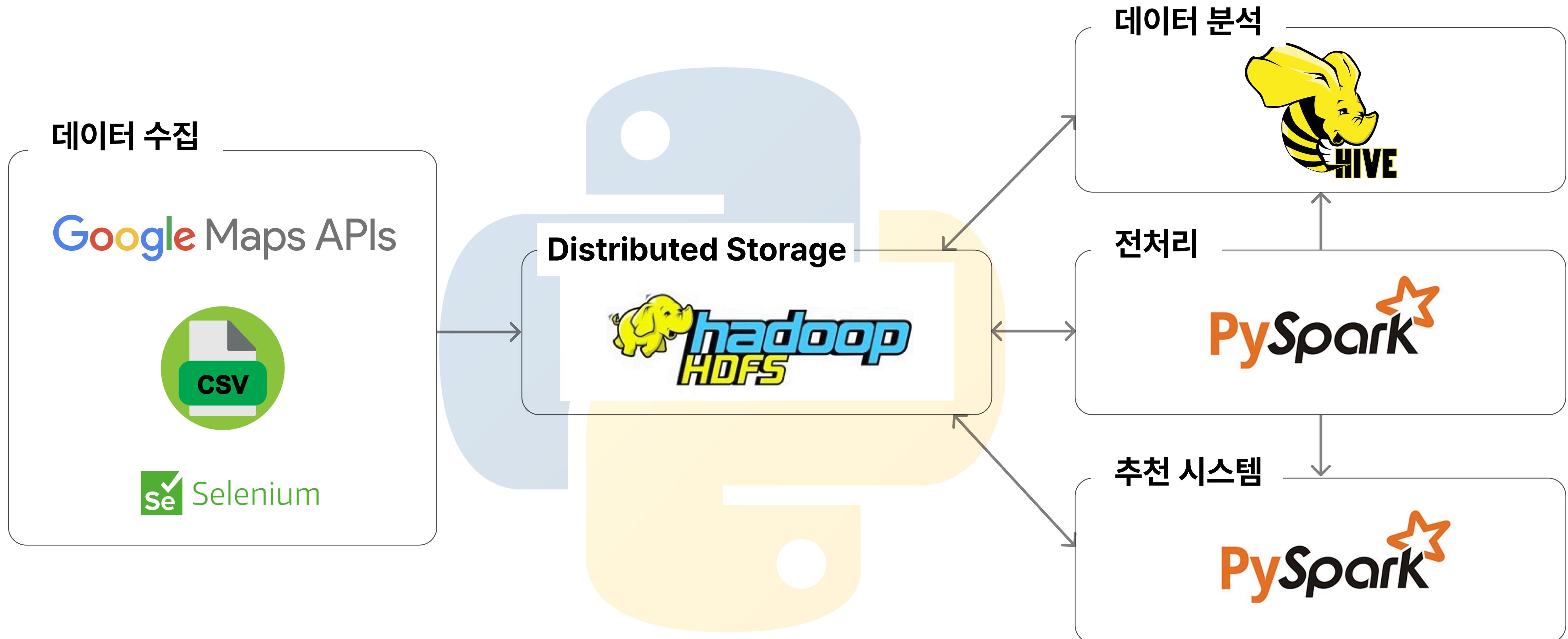
데이터 분석을 통해 신뢰성 높은 창업 정보 도출

데이터 분석을 활용한 창업 증가

서울시 대학생들의 선호도에 맞는 업종을 한눈에 제공하는
업종 추천 및 과잉 분석 시스템

시스템 아키텍쳐

BDP Term Project



2. 데이터 수집

BDP Term Project

서울소재 대학 별 재학생 수



대학알리미

재적_학생_현황_(대학)
_2024-12-04079849.csv

서울시 상권 정보

DATA 공공데이터포털 . GO . KR

소상공인시장진흥공단_
상권정보_서울_202409.csv

서울 소재 대학별 면적, 위도, 경도

Google Maps APIs



Selenium

고려대학교	성북구	946,177m ²	대학알리미
이화여자대학교	서대문구	554,927m ²	대학알리미
서울과학기술대학교	노원구	504,922m ²	대학알리미

university_locations.csv



논의점

1

대학교마다 면적이 다른데, 반경을 어떻게 잡을 것인가?

면적 $100,000\text{m}^2$ 이하 학교는 반경 500m

면적 $100,000\text{m}^2$ 초과시, $100,000\text{m}^2$ 당 반경 100m 추가

면적 $1,000,000\text{m}^2$ 이상 학교의 반경은 2000m로 고정

Why?

- 왜 이렇게 기준을 잡았는가?

2. 전처리

BDP Term Project

명지대학교

2. 전처리

BDP Term Project

서울대학교

지도 홈

길찾기

버스

지하철

기차

저장

...

더보기

N

대중교통 자동차 도보 자전거

서울 관악구 관악로 1
서울 관악구 남부순환로237길 60

다시입력 경유지 길찾기 >

1종 휘발유 차량 기준

차종/연료 설정 >

① 실시간 추천
6분 | 2.0km
택시비 5,000원 | 통행료 무료 | 연료비 282원
서행 낙성대로 1.2km → 미확인 남부순환로237길 302m
상세보기 >

② 짧은 거리
6분 | 2.0km
택시비 5,000원 | 통행료 무료 | 연료비 282원
서행 낙성대로 1.2km → 미확인 남부순환로237길 302m
상세보기 >

③ 무료우선
6분 | 2.0km
택시비 5,000원 | 통행료 무료 | 연료비 282원
서울 관악구 관악로 1 → 서울 관악구 남부순환...

2호선 봉천역 2호선 서울대입구역 행운동
봉천두산 1단지아파트 중앙동 까치산공원
동작고등학교 일반지도 위성지도 지형지도
사당4동 사당 테마 저장
지적편집도 거리부 반경 면적 거리 다운로드
인쇄 공유

도착

1 6분

출발

전체경로 보기

© NAVER 300m

2. 데이터 전처리

BDP Term Project

따라서, 면적마다 반경을 추가해줌

대학명	면적	위도	경도	반경
서울대학교	4317000	37.4648267	126.9571988	2000
연세대학교	952750	37.5663937	126.9387066	1400
고려대학교	946177	37.5893875999999	127.0324773	1400
이화여자대학교	5549271	37.5618588	126.9468339	1000
서울과학기술대학교	5049221	37.6316684	127.0774813	1000
건국대학교	4563921	37.5423265	127.0759204	900
서울시립대학교	4340041	37.5838657	127.0587771	900
경희대학교	4073761	37.59685	127.05181	900
한양대학교	4017291	37.5571759	127.0454092	900
삼육대학교	3195371	37.6435824	127.1063459	800
서강대학교	2420911	37.5509442	126.9410023	700
서울여자대학교	2250201	37.6281126	127.0904568	700
국민대학교	1917361	37.6108694	126.9972889	600

university_locations.csv

3. 데이터 전처리

BDP Term Project



논의점

2

대학교(위도, 경도)와 음식점(위도, 경도)의 거리를
어떻게 구할 것인가?

3. 전처리

BDP Term Project



보통 두 점(위도, 경도) 사이의 거리를 구하지만

$$\overline{AB} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

3. 전처리

BDP Term Project



지구는 둥글다.

3. 전처리

BDP Term Project

하버사인 공식 (Haversine Formula)

둥근 지구를 고려하여
두 위도, 경도 좌표 사이의 거리를 구할 때
사용하는 공식

$$\Theta = \frac{d}{r}$$

Θ 는 두 점을 잇는 호의 중심각입니다 (라디안 단위)

$$\text{hav}(\Theta) = \text{hav}(\varphi_2 - \varphi_1) + \cos(\varphi_1) \cos(\varphi_2) \text{hav}(\lambda_2 - \lambda_1)$$

- φ_1, φ_2 : 1지점과 2지점의 위도 (라디안 단위),
- λ_1, λ_2 : 1지점과 2지점의 경도 (라디안 단위).

$\text{hav}(\Theta)$ 은 하버사인 함수로 다음과 같이 표현됩니다.

$$\text{hav}(\theta) = \sin^2\left(\frac{\theta}{2}\right) = \frac{1 - \cos(\theta)}{2}$$

거리를 구하기 위해서 역함수인 아크하버사인을 곱해줍니다.

$$d = r \text{archav}(h) = 2r \arcsin(\sqrt{h})$$

$$\begin{aligned} d &= 2r \arcsin\left(\sqrt{\text{hav}(\varphi_2 - \varphi_1) + \cos(\varphi_1) \cos(\varphi_2) \text{hav}(\lambda_2 - \lambda_1)}\right) \\ &= 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos(\varphi_1) \cos(\varphi_2) \sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right) \end{aligned}$$

3. 전처리

BDP Term Project

일단 restaurant.csv 전처리

10개
G2 : 소매업
I1 : 숙박업
I2 : 음식점업
L1 : 부동산업
M1 : 전문, 과학 및 기술 서비스업
N1 : 사업시설 관리, 사업 지원 및 임대 서비스업
P1 : 교육 서비스업
Q1 : 보건의료업
R1 : 예술, 스포츠 및 여가관련 서비스업
S2 : 수리 및 개인 서비스업

Point 1

'상호명', '상권업종대분류코드', '상권업종대분류명', '상권업종중분류코드', '상권업종중분류명', '상권업종소분류코드', '상권업종소분류명', '경도', '위도' column을 제외한 나머지 삭제

Point 2

상권업종대분류코드 'I2'를 제외한 나머지 삭제 ('I2' = 음식점)

3. 전처리

BDP Term Project

	상호명	상권업종대분류코드	상권업종대분류명	상권업종중분류코드	상권업종중분류명	상권업종소분류코드	상권업종소분류명	경도	위도
0	매머드커피외대	I2	음식	I212	비알코올	I21201	카페	127.056962	37.598493
1	썬더치킨	I2	음식	I210	기타 간이	I21006	치킨	127.073598	37.553662
2	멜팅그릴	I2	음식	I210	기타 간이	I21004	버거	126.955913	37.600976
3	제이엔제이슨R	I2	음식	I211	주점	I21104	요리 주점	126.924515	37.562797
4	미스홍대게스트하우스 2	I2	음식	I212	비알코올	I21201	카페	126.923175	37.552909

restaurant.csv

3. 전처리

BDP Term Project

```
from math import radians, sin, cos, sqrt, atan2 # 필요한 수학 함수들을 임포트

def haversine(lat1, lon1, lat2, lon2):
    R = 6371000 # 지구의 평균 반지름을 미터 단위로 정의

    # 위도와 경도를 라디안으로 변환
    lat1, lon1, lat2, lon2 = map(radians, [lat1, lon1, lat2, lon2])
    dlat = lat2 - lat1 # 위도 차이
    dlon = lon2 - lon1 # 경도 차이

    # 하버사인 공식을 사용하여 두 지점 간의 중심각을 계산
    a = sin(dlat / 2)**2 + cos(lat1) * cos(lat2) * sin(dlon / 2)**2
    c = 2 * atan2(sqrt(a), sqrt(1 - a))

    # 두 지점 간의 거리를 반환합니다.
    return R * c # 반환 값: 두 지점 간의 거리
```

하버사인(Haversine) 공식을 사용하여, 두 지점 간의 거리를 반환 함수

3. 전처리

BDP Term Project

각 음식점이
어느 대학교의 반경 안에 있는지 추가

```
def find_universities_for_restaurant(lat, lon):

    universities_within_radius = [] # 대학 목록을 저장할 리스트 초기화

    for row in broadcast_universities.value:
        uni_lat = row['위도']
        uni_lon = row['경도']
        radius = row['반경']

        # 하버사인 공식을 사용해 음식점과 대학사이의 거리 계산
        distance = haversine(lat, lon, uni_lat, uni_lon)

        # 계산된 거리가 대학의 반경(radius) 이내라면, 대학명을 리스트에 추가
        if distance <= radius:
            universities_within_radius.append(row['대학명'])

    # 반경 내에 있는 대학명을 문자열로 반환
    return ', '.join(universities_within_radius)

# restaurant_final_df 데이터프레임에 "대학교"라는 새로운 열을 추가
restaurant_final_df = restaurant_final_df.withColumn("대학교",
                                                       find_universities_udf(F.col("위도"), F.col("경도")))

# "대학교" 열이 비어 있지 않은 레스토랑만 필터링
restaurant_final_df = restaurant_final_df.filter(F.col("대학교") != "")
```

3. 전처리

상호명	상권업종대분류코드	상권업종대분류명	상권업종중분류코드	상권업종중분류명	상권업종소분류코드	상권업종소분류명	경도	대학교
								경희대학교, 한국외국어대학교
0 매머드커피외대	I2	음식	I212	비알코올	I21201	카페	127.056962	3 경희대학교, 한국외국어대학교
1 썬더치킨	I2	음식	I210	기타 간이	I21006	치킨	127.073598	3 세종대학교
2 멜팅그릴	I2	음식	I210	기타 간이	I21004	버거	126.955913	3 상명대학교
3 제이앤제이슨R	I2	음식	I211	주점	I21104	요리 주점	126.924515	3 연세대학교
4 미스홍대게스트하우스 2	I2	음식	I212	비알코올	I21201	카페	126.923175	3 홍익대학교

restaurant_with_universities.csv

4. 추천 시스템

논의점 3

대학 비교 지표는 어떻게 설정할 것인가?

└ count_per_density

고려 항목

- 유동 인구 수(재학생 수)
- 대학교의 면적에 따라 결정한 반경의 면적(반경 * 반경 * pi)
- 대학의 업종 카테고리별 count 수

1

인구밀도(재학생수/면적)를 정규화

대학명, 반경(m), 총재학생수, 밀도(학생/km²)

서울대학교, 2000, 17010, 0.06326242189824456

연세대학교, 1400, 20066, 0.1523022181202145

고려대학교, 1400, 21029, 0.15961144945928388

건국대학교, 900, 15491, 0.2845089461175943

서울시립대학교, 900, 8796, 0.16154804015559743

경희대학교, 900, 25575, 0.4697124973828335

한양대학교, 900, 16469, 0.3024709724104745

삼육대학교, 800, 5214, 0.12119718833680451

서강대학교, 700, 8193, 0.24874156743923398

서울여자대학교, 700, 7207, 0.21880635622294145

국민대학교, 600, 14486, 0.5986148302423221

덕성여자대학교, 600, 5357, 0.2213709544117161

중앙대학교, 600, 18688, 0.7722569341135245

상명대학교, 600, 6152, 0.2542232801084333

동국대학교, 600, 13559, 0.5603077787695461

성신여자대학교, 600, 9254, 0.38240933584581305

성균관대학교, 600, 19351, 0.7996545340341829

...

4. 추천 시스템

논의점 3

대학 비교 지표는 어떻게 설정할 것인가?

└ count_per_density

고려 항목

● 유동 인구 수(재학생 수)

● 대학교의 면적에 따라 결정한 반경의 면적

● 대학의 업종 카테고리별 count 수

2

업종 카테고리별 count 수 / 인구밀도(재학생수/면적) 정규화

대학교	상권업종중분류코드	상권업종소분류코드	count	count_per_density
가톨릭대학교 _제2캠퍼스 I201	I20101	19	443.8507042253521	
가톨릭대학교 _제2캠퍼스 I201	I20102	1	23.36056338028169	
가톨릭대학교 _제2캠퍼스 I201	I20103	2	46.72112676056338	
가톨릭대학교 _제2캠퍼스 I201	I20105	1	23.36056338028169	
가톨릭대학교 _제2캠퍼스 I201	I20106	1	23.36056338028169	
가톨릭대학교 _제2캠퍼스 I201	I20107	1	23.36056338028169	
가톨릭대학교 _제2캠퍼스 I201	I20111	1	23.36056338028169	
가톨릭대학교 _제2캠퍼스 I201	I20112	1	23.36056338028169	
가톨릭대학교 _제2캠퍼스 I202	I20201	4	93.44225352112676	
가톨릭대학교 _제2캠퍼스 I202	I20202	1	23.36056338028169	
가톨릭대학교 _제2캠퍼스 I203	I20301	8	186.88450704225352	
가톨릭대학교 _제2캠퍼스 I203	I20302	2	46.72112676056338	
가톨릭대학교 _제2캠퍼스 I204	I20401	6	140.16338028169014	
가톨릭대학교 _제2캠퍼스 I204	I20402	1	23.36056338028169	
가톨릭대학교 _제2캠퍼스 I205	I20501	2	46.72112676056338	
가톨릭대학교 _제2캠퍼스 I207	I20701	11	256.9661971830986	
가톨릭대학교 _제2캠퍼스 I210	I21001	11	256.9661971830986	
가톨릭대학교 _제2캠퍼스 I210	I21002	1	23.36056338028169	
가톨릭대학교 _제2캠퍼스 I210	I21003	4	93.44225352112676	

only showing top 20 rows

4. 추천 시스템 Recommendation - 2 (GroupBy)

서울시 모든 대학들의 카테고리별 평균 밀도값(avg_count_per_density) 계산

업종 중분류, 소분류별로 그룹화하고 `count_per_density`의 평균을 계산

상권업종중분류코드	상권업종소분류코드	avg_count_per_density
I210	I21001	130.94743983736024
I210	I21002	35.54037796497919
I206	I20601	7.801505174035747
I203	I20399	9.830987370885103
I203	I20302	23.40838890595438
I207	I20702	9.017991388296176
I210	I21006	124.25655180339008
I201	I20101	466.20388760858316
I204	I20403	8.06683946896279
I205	I20501	46.22047559597508
I202	I20201	93.69836522191731
I211	I21104	249.54751874991572
I207	I20701	27.115657698361
I201	I20113	5.548612621963348
I210	I21007	193.72820694640816

grouped_df

4. 추천 시스템 Recommendation - 2 (Recommendation Calculate)

$$\text{difference_from_avg} = \frac{\text{count_per_density} - (\text{명지대학교에서 카테리별 평균 밀도 값} - \text{모든 대학들의 카테고리별 평균 밀도 값})}{\frac{\text{모든 대학들의 카테고리별 평균 밀도 값}}{\text{avg_count_per_density}}}$$

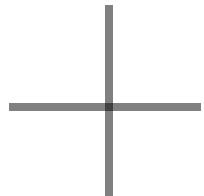
상권업종중분류코드	상권업종소분류코드	difference_from_avg	current_count
I204	I20402	-0.9190250492546234	1
I207	I20701	-0.9149504853692133	1
I201	I20108	-0.9128362125899546	1
I211	I21101	-0.9119706479386412	2

값이 평균에서 얼마나 떨어져 있는지를 비율로 정규화

4. 추천 시스템 Recommendation - 2 (Recommendation ready to output)

추천 결과를 업종 명칭 데이터와 매칭하여 사람이 읽기 쉽게 변환

상권업종중분류코드	상권업종소분류코드	difference_from_avg	current_count
I204	I20402	-0.9190250492546234	1
I207	I20701	-0.9149504853692133	1
I201	I20108	-0.9128362125899546	1
I211	I21101	-0.9119706479386412	2



|상권업종대분류코드, 상권업종대분류명, 상권업종중분류코드, 상권업종중분류명, 상권업종소분류코드, 상권업종소분류명
I2, 음식, I201, 한식, I20101, 백반/한정식
I2, 음식, I201, 한식, I20102, 국/탕/찌개류
I2, 음식, I201, 한식, I20103, 족발/보쌈
I2, 음식, I201, 한식, I20104, 전/부침개
I2, 음식, I201, 한식, I20105, 국수/칼국수
I2, 음식, I201, 한식, I20106, 냉면/밀면

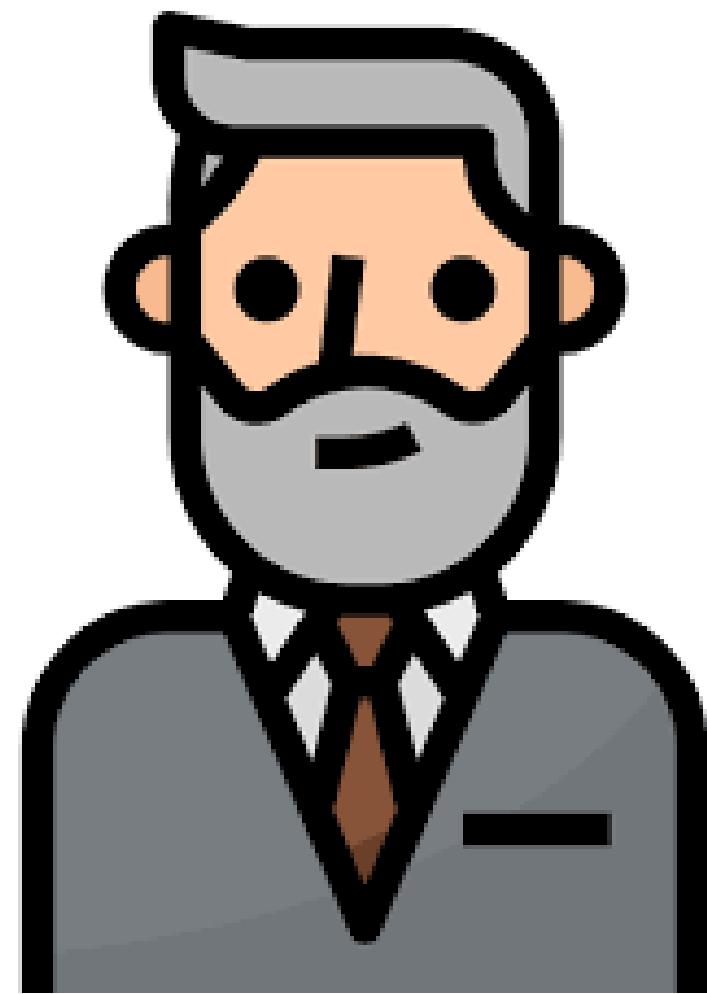
4. 추천 시스템 Recommendation - 2 (Recommendation ready to output)

상권업종중분류명	상권업종소분류명	percentage	current_count
서양식	파스타/스테이크	-91.90250492546234 1	
구내식당·뷔페	구내식당	-91.49504853692133 1	
한식	소고기 구이/찜	-91.28362125899547 1	
주점	일반 유흥 주점	-91.19706479386413 2	
서양식	기타 서양식 음식점	-89.45898013600593 1	
한식	냉면/밀면	-88.86818854351515 1	
한식	해산물 구이/찜	-88.1567622936747 2	
한식	기타 한식 음식점	-86.46870381358126 1	
동남아시아	베트남식 전문	-85.03148120132511 3	
한식	국수/칼국수	-83.79906614551484 4	
서양식	경양식	-83.19305197658198 10	
주점	생맥주 전문	-81.62327819633913 6	
일식	일식 면 요리	-81.47091568352045 2	
일식	일식 카레/돈가스/덮밥	-80.2961789860426 2	
기타 간이	그 외 기타 간이 음식점	-79.14346395456579 4	
주점	요리 주점	-77.82059108236177 24	
한식	돼지고기 구이/찜	-76.47819254135851 17	
기타 간이	아이스크림/빙수	-76.31895955616235 2	
일식	일식 회/초밥	-76.28953278971318 13	
기타 간이	피자	-75.00356599515106 11	

only showing top 20 rows

추천 결과를 업종 명칭 데이터와 매칭하여 사람이 읽기 쉽게 변환

4. 추천 시스템 Recommendation - 2 (Recommendation Calculate)



사장

명지대학교 근처에 대학생들이
선호하는 음식점을 차리고 싶어

4. 추천 시스템 Recommendation - 2 (Recommend)

→ BDP spark-submit recommend.py 명지대학교

서울시 대학 근처 음식점 평균 개수를 기반으로
- 타겟 대학에서 창업을 추천할 만한 카테고리를 추천
- 과잉 분포하고 있는 카테고리도 제공

필요한 CSV 파일 및 위치:

1. university_info.csv
위치: hdfs://user/maria_dev/term_project/output/
내용: 대학교 정보 (대학명, 밀도(학생/km²) 등)
2. university_business_count.csv
위치: hdfs://user/maria_dev/term_project/output/
내용: 대학별 사업체 수 데이터
3. classification_codes.csv
위치: hdfs://user/maria_dev/term_project/output/
내용: 상권 업종 분류 코드 및 명칭

추천/과잉 기준: 0.5

step 1

spark-submit을 통해 recommend 코드를 수행

step 2

타겟 대학을 입력하기 위해 코드 내에 sys.argv를 사용하여 입력받도록 함

step 3

명령어를 입력하면 코드에 대한 description과 추천/과잉 기준 값이 출력됨

4. 추천 시스템 Recommendation - 2 (Recommend)

===== [추천 업종] =====
서양식 > 파스타/스테이크
현재 1개 운영 중
평균보다 91.9% 적음

구내식당·뷔페 > 구내식당
현재 1개 운영 중
평균보다 91.5% 적음

한식 > 소고기 구이/찜
현재 1개 운영 중
평균보다 91.3% 적음

주점 > 일반 유통 주점
현재 2개 운영 중
평균보다 91.2% 적음

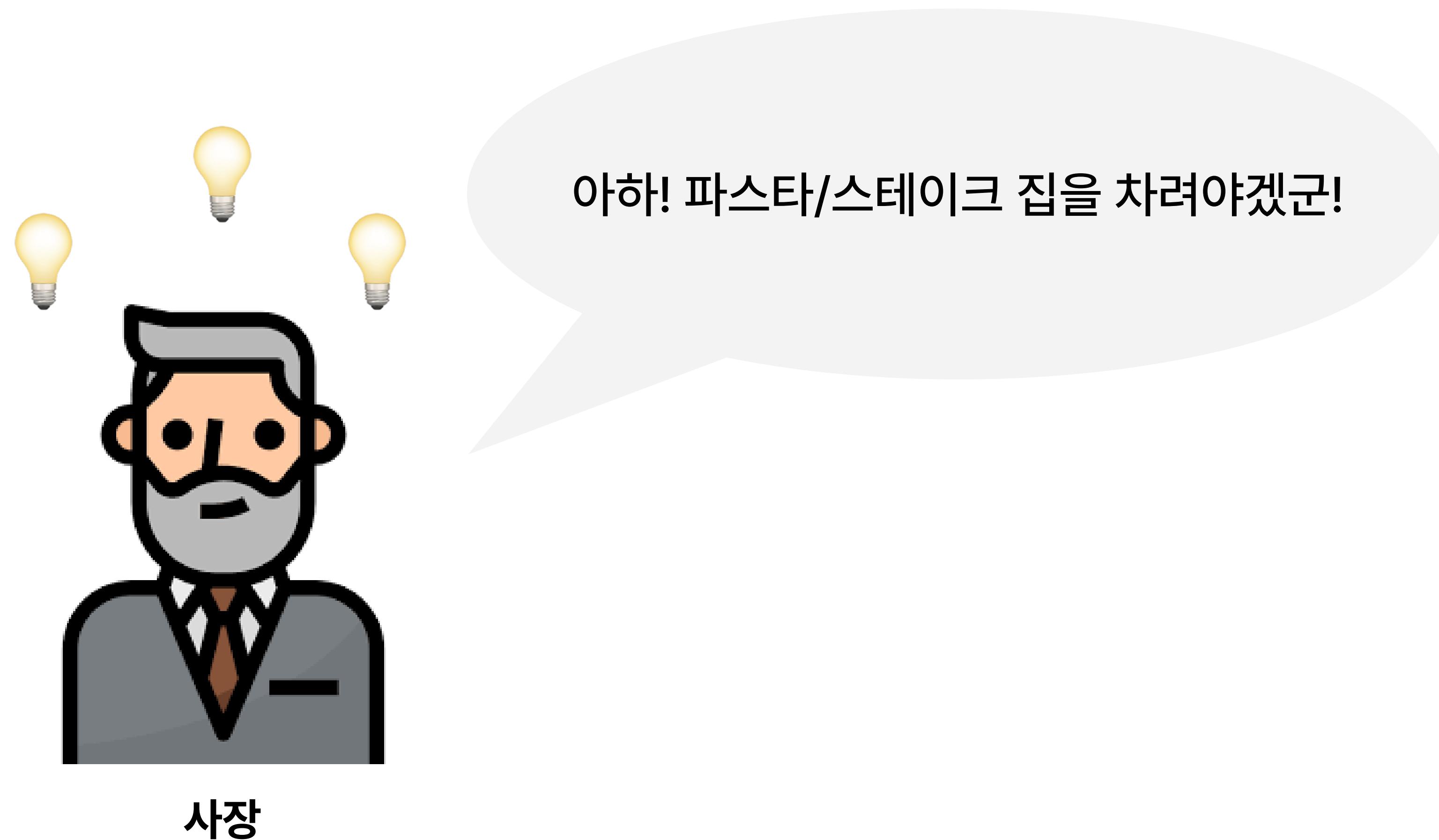
서양식 > 기타 서양식 음식점
현재 1개 운영 중
평균보다 89.5% 적음

한식 > 냉면/밀면
현재 1개 운영 중
평균보다 88.9% 적음

===== [과잉 업종] =====
과잉 업종이 없습니다.

difference_from_avg 값을 100을 곱해 퍼센트로 계산되어,
50% 이상은 "과잉", -50% 이하는 "추천"으로 분류

4. 추천 시스템 Recommendation - 2 (Recommendation Calculate)



출처

최단거리 구하기, 하버사인 공식(Haversine Formula)

<https://kayuse88.github.io/haversine/>