

Team : CAFÉ_IN

Café recommendation by keyword

김우석 김정문 임민우 허환욱

Contents

01. Purpose

02. Data crawling

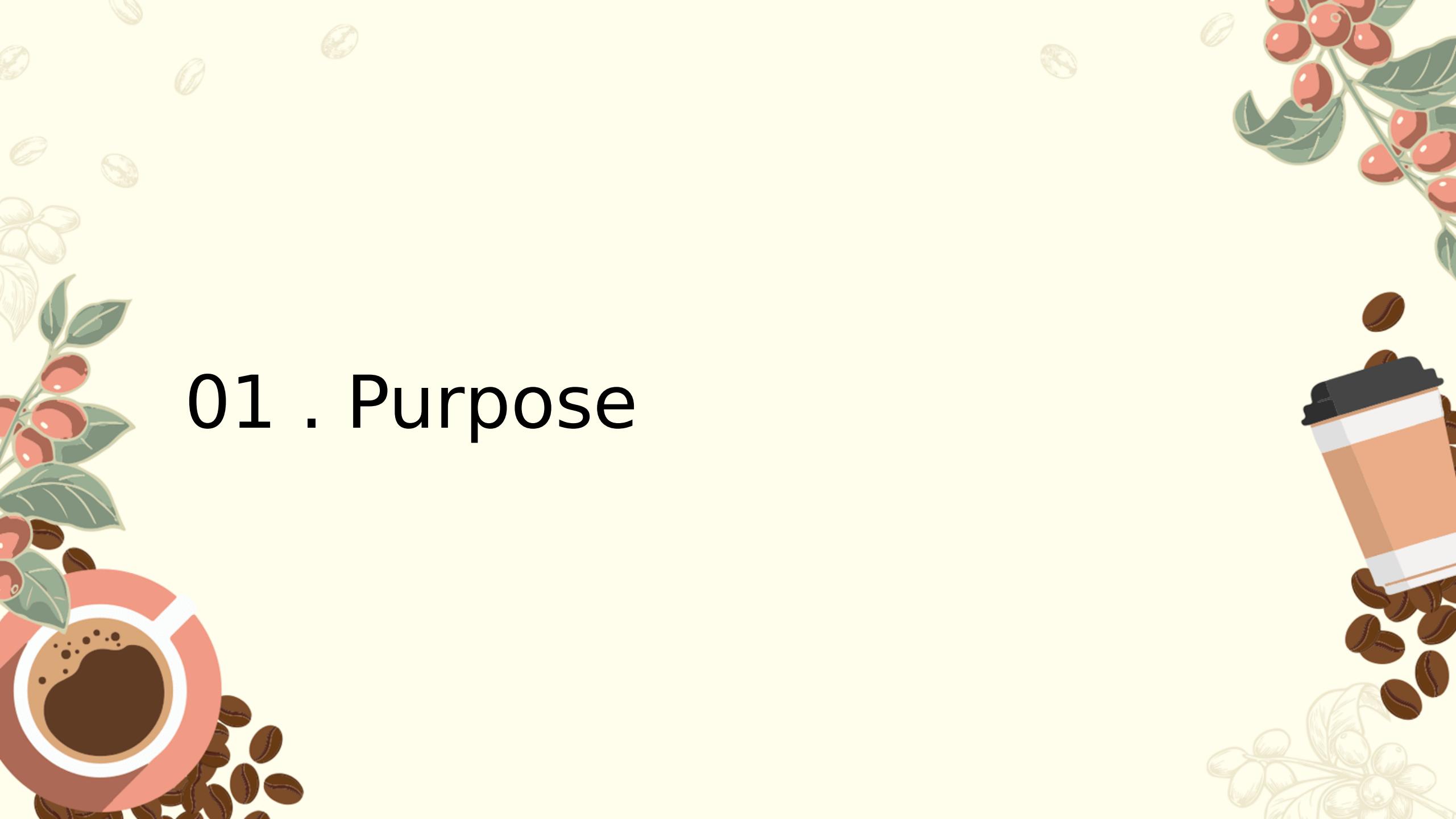
03. Preprocessing

04. TF-IDF and Word2vec

05. App demonstration
video

06. Improvements

01 . Purpose



01. Purpose



Coffee



Sandwich
, Bread

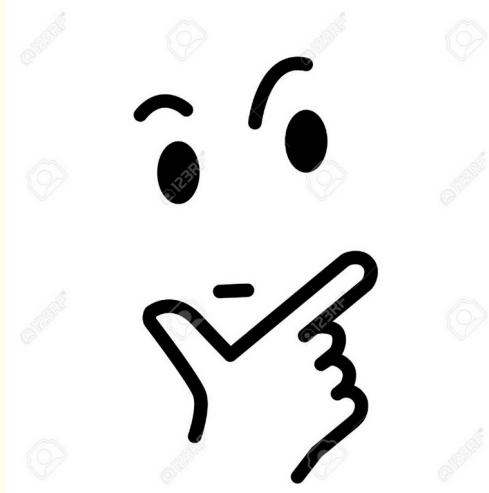


Space
,
mood



01. Purpose

Other cafes...?
Other regions...?



02 . Data crawling



02. Data crawling

검색어: 강남역 카페

지도 화면에서 표시되는 카페 목록:

- 썸띵어바웃커피 카페, 디저트 (영업 중, ★4.42, 리뷰 999+)
- 트리오드 카페, 디저트 (영업 중, ★4.56, 리뷰 999+, 식스센스3)
- 베이커스트 브라운 카페, 디저트 (영업 중, 리뷰 999+, 서울 강남구 역삼동)

지도 상의 카페 위치:

- 썸띵어바웃커피 (★4.42)
- 정월 (★4.46)
- 노티드 강남 카카오 (★4.53)
- 고양이라 좋은날 (★4.79)
- 필메이트 강남점 (★4.56)
- 트리오드 (★4.56)
- 타르타트 강남역점 (★4.45)
- 장고방 (★4.45)
- 스타벅스 강남R점 (★4.51)
- 리퍼크 (★4.42)
- 던킨 라이브 강남 (★4.46)
- 셀렉티드마롱 (★4.94)
- 시티갤러리카페 본점 (점포)
- 비아살라리아 (점포)
- 노블라 (아이스크림)
- 브레멘코 강남역점 (점포)
- 터家装 (점포)
- 리체힐신한은행 오피스텔
- 노보텔앰배서더 서울강남
- 삼성호텔
- 더체플렛 녹현
- 교회
- KT 강남지사
- 서울집
- GS칼텍스
- GS타워
- 한국은행
- 브리티ッシュ아메리칸 토바코코리아
- LG역삼 애플라트
- 신동궁김자탕
- 신동아 1차아파트
- 신동아 2차아파트
- 래미안서초 에스티지아파트
- 래미안서초 에스티지S아파트
- 한국 컨퍼런스센터
- 고깃집열
- 광일프라자
- 대륭서초타워
- 성우스타우스 오피스텔
- 유니온센타
- 신동아 1차아파트
- 신동아 2차아파트
- 래미안리더스원 아파트
- 현대백화점
- HJ빌딩
- SK엔크린 LPG
- 문암미술관
- 역삼역
- 역삼1동
- 역삼2동
- 역삼3동
- 역삼4동
- 역삼5동
- 역삼6동
- 역삼7동
- 역삼8동
- 역삼9동
- 역삼10동
- 역삼11동
- 역삼12동
- 역삼13동
- 역삼14동
- 역삼15동
- 역삼16동
- 역삼17동
- 역삼18동
- 역삼19동
- 역삼20동
- 역삼21동
- 역삼22동
- 역삼23동
- 역삼24동
- 역삼25동
- 역삼26동
- 역삼27동
- 역삼28동
- 역삼29동
- 역삼30동
- 역삼31동
- 역삼32동
- 역삼33동
- 역삼34동
- 역삼35동
- 역삼36동
- 역삼37동
- 역삼38동
- 역삼39동
- 역삼40동
- 역삼41동
- 역삼42동
- 역삼43동
- 역삼44동
- 역삼45동
- 역삼46동
- 역삼47동
- 역삼48동
- 역삼49동
- 역삼50동
- 역삼51동
- 역삼52동
- 역삼53동
- 역삼54동
- 역삼55동
- 역삼56동
- 역삼57동
- 역삼58동
- 역삼59동
- 역삼60동
- 역삼61동
- 역삼62동
- 역삼63동
- 역삼64동
- 역삼65동
- 역삼66동
- 역삼67동
- 역삼68동
- 역삼69동
- 역삼70동
- 역삼71동
- 역삼72동
- 역삼73동
- 역삼74동
- 역삼75동
- 역삼76동
- 역삼77동
- 역삼78동
- 역삼79동
- 역삼80동
- 역삼81동
- 역삼82동
- 역삼83동
- 역삼84동
- 역삼85동
- 역삼86동
- 역삼87동
- 역삼88동
- 역삼89동
- 역삼90동
- 역삼91동
- 역삼92동
- 역삼93동
- 역삼94동
- 역삼95동
- 역삼96동
- 역삼97동
- 역삼98동
- 역삼99동
- 역삼100동

Naver
map

02. Data crawling

Seoul

Gangnam,
Itaewon...etc

Suwon

Ingye-dong,
Haenggung-dong...
etc

Daejeon

Dunsan-dong,
Daeheung-dong...
etc



Daegu

Myeongdeok ,
Jungang-ro...etc

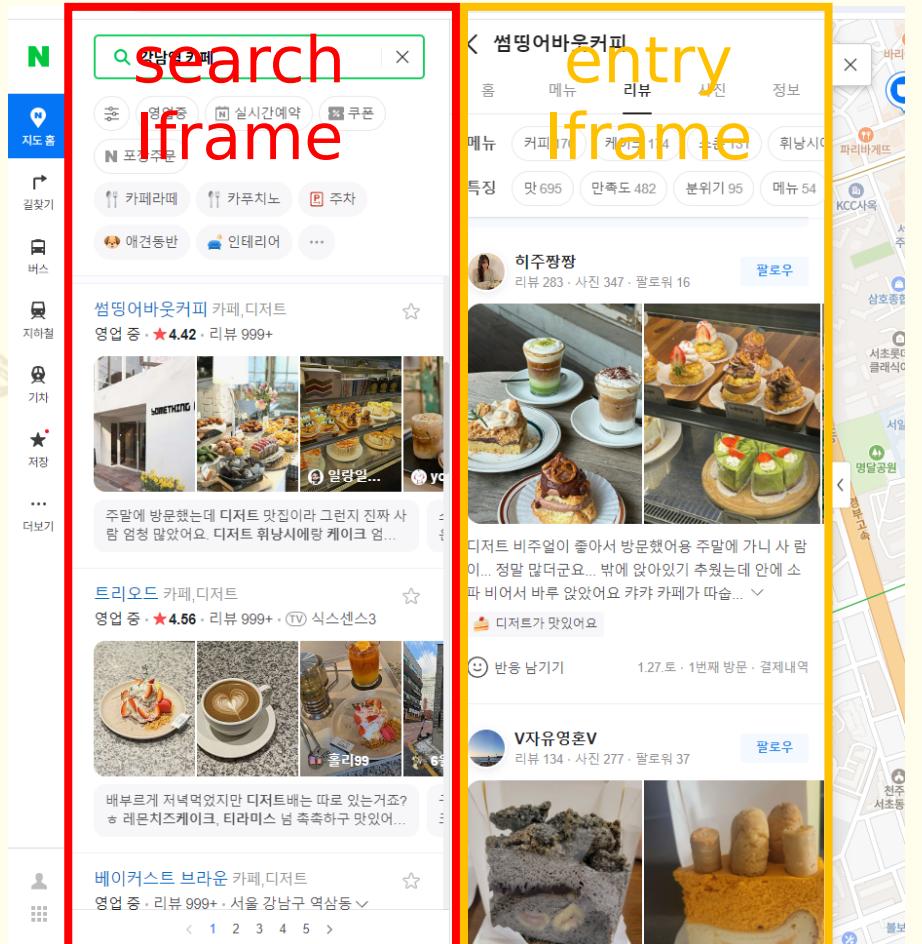
Busan

Seomyeon,
Haeundae...etc

40 regions, 4000 cafes, Total 400k re-views

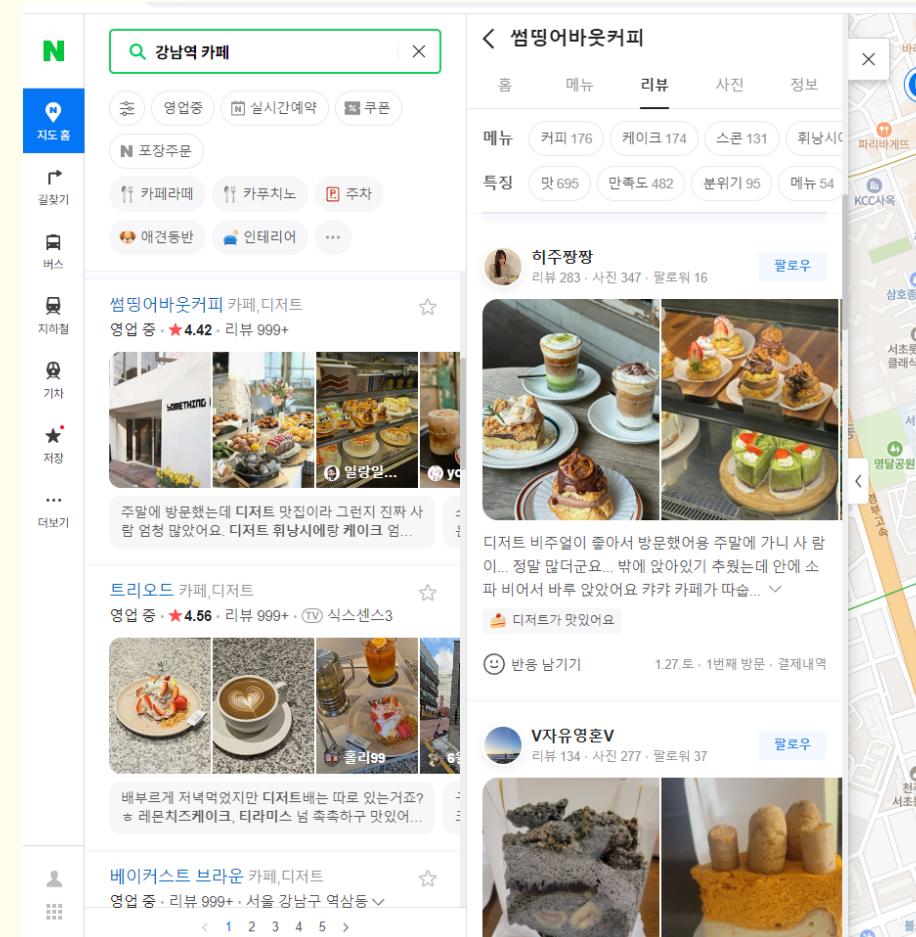
02. Data crawling - Difficulties and solutions

Frame issue



```
driver.switch_to.default_content() #프레임 초기화  
driver.switch_to.frame('searchIframe') #프레임 변경
```

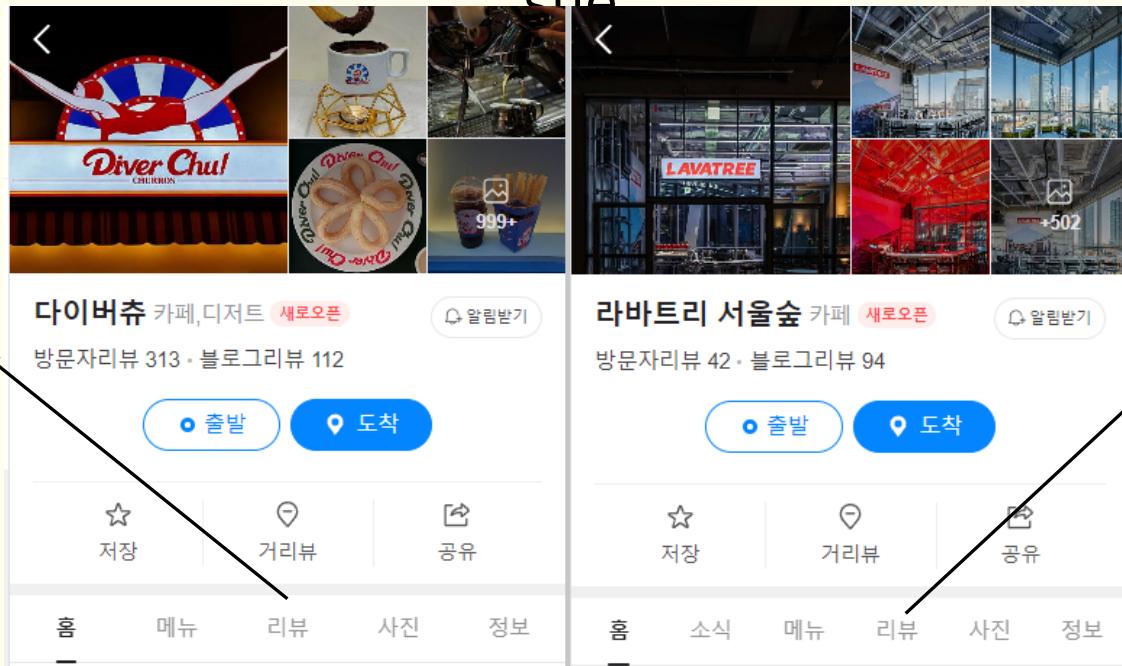
Scroll issue



```
pyautogui.keyDown('pgdn')  
pyautogui.keyUp('pgdn')
```

02. Data crawling - Difficulties and solutions

Review menu button location is-



`//*[@id="app-root"]/div/div/div[4]/div/div/div/div/a[2]`

`//*[@id="app-root"]/div/div/div[4]/div/div/div/div/a[3]`

```
try:  
    btn_lists = driver.find_elements(By.CLASS_NAME, value: 'veBoZ')  
    for btn_list in btn_lists:  
        if btn_list.text == '리뷰':  
            btn_list.click()  
            time.sleep(1)  
except:  
    print('리뷰메뉴 클릭 오류')
```

03 . Preprocessing



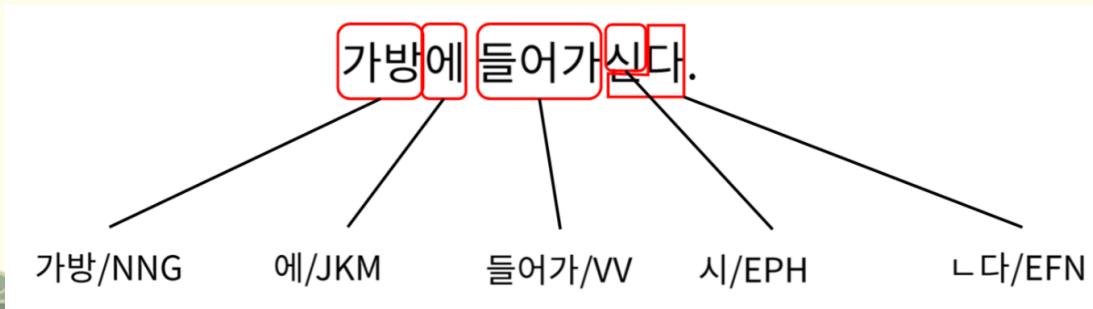
03. Preprocessing

Morpheme seperation



Remove stop words

| | |
|---|-----------|
| 1 | ,stopword |
| 2 | 0,아 |
| 3 | 1,휴 |
| 4 | 2,아이구 |
| 5 | 3,아이쿠 |
| 6 | 4,아이고 |
| 7 | 5,어 |
| 8 | 6,나 |



03. Preprocessing



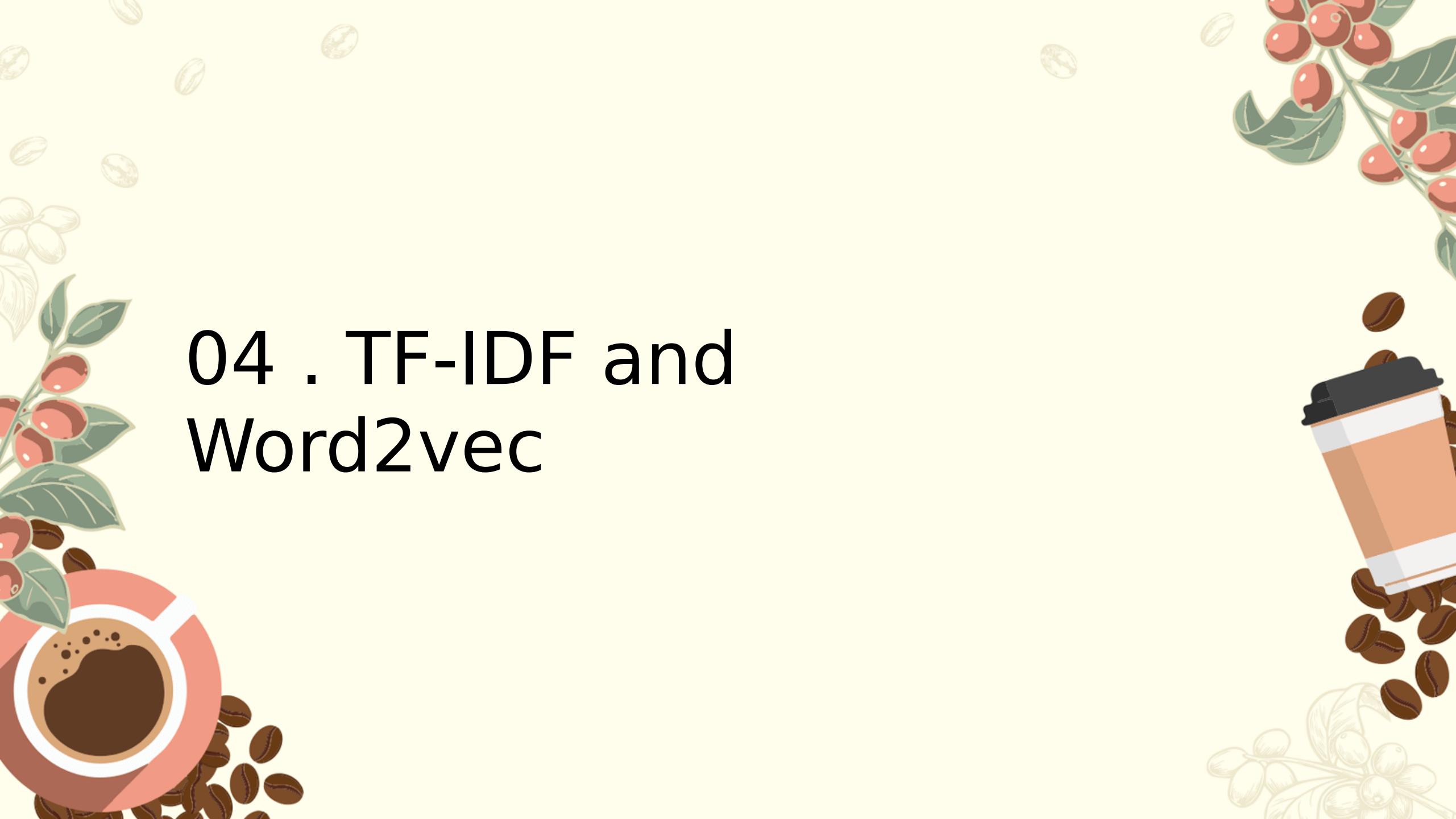
Unnecessary words appearing in Word cloud

Remove stop words

```
df_stopwords = pd.read_csv('./stopwords.csv')
stopwords = list(df_stopwords['stopword'])
stopwords = stopwords + ['맛있다', '좋다', '먹다']
```

Combining morphemes And Tokenization

04 . TF-IDF and Word2vec



04. TF-IDF and Word2vec - TF-IDF

TF-IDF?

- TF(Term Frequency)
 - $tf(d, t)$: 특정 문서 d 에서의 특정 단어 t 의 등장 횟수
 - 값이 높을수록 해당 단어가 문서에서 중요하다는 것을 의미함
- DF(Document Frequency)
 - $df(t)$: 특정 단어 t 가 등장한 문서의 수
 - 값이 높을수록 해당 단어가 흔하게 등장한다는 것을 의미함

- IDF(Inverse Document Frequency)
 - $idf(d, t) : df(t)$ 에 반비례하는 수
 - 모든 문서에 자주 등장하는 단어일수록 낮은 가중치를 줌

$$idf(d, t) = \log\left(\frac{n}{1 + df(t)}\right)$$

- log를 사용하는 이유
 - log를 사용하지 않으면 총 문서의 수 n 이 커질수록, IDF의 값이 기하급수적으로 커지게 됨
- 분모에 1을 더하는 이유
 - 특정 단어가 전체 문서에서 등장하지 않을 경우 분모가 0이 되지 않도록 하기 위함

04. TF-IDF and Word2vec - TF-IDF

TF-IDF Example

- 문서 1 : 나는 학교에 갔다
- 문서 2 : 나는 집에 갔다
- 문서 3 : 나는 회사에 갔다

TF

| 문서 | 갔다 | 나는 | 집에 | 학교에 | 회사에 |
|------|----|----|----|-----|-----|
| 문서 1 | 1 | 1 | 0 | 1 | 0 |
| 문서 2 | 1 | 1 | 1 | 0 | 0 |
| 문서 3 | 1 | 1 | 0 | 0 | 1 |
| 총 개수 | 3 | 3 | 1 | 1 | 1 |

IDF

| 단어 | IDF |
|-----|--|
| 갔다 | $\log\left(\frac{3}{1+3}\right) = -0.287682$ |
| 나는 | $\log\left(\frac{3}{1+3}\right) = -0.287682$ |
| 집에 | $\log\left(\frac{3}{1+1}\right) = 0.405465$ |
| 학교에 | $\log\left(\frac{3}{1+1}\right) = 0.405465$ |
| 회사에 | $\log\left(\frac{3}{1+1}\right) = 0.405465$ |

| 문서 | 갔다 | 나는 | 집에 | 학교에 | 회사에 |
|------|-----------|-----------|----------|----------|----------|
| 문서 1 | -0.287682 | -0.287682 | 0 | 0.405465 | 0 |
| 문서 2 | -0.287682 | -0.287682 | 0.405465 | 0 | 0 |
| 문서 3 | -0.287682 | -0.287682 | 0 | 0 | 0.405465 |

TF-IDF(TF x IDF)

04. TF-IDF and Word2vec - TF-IDF

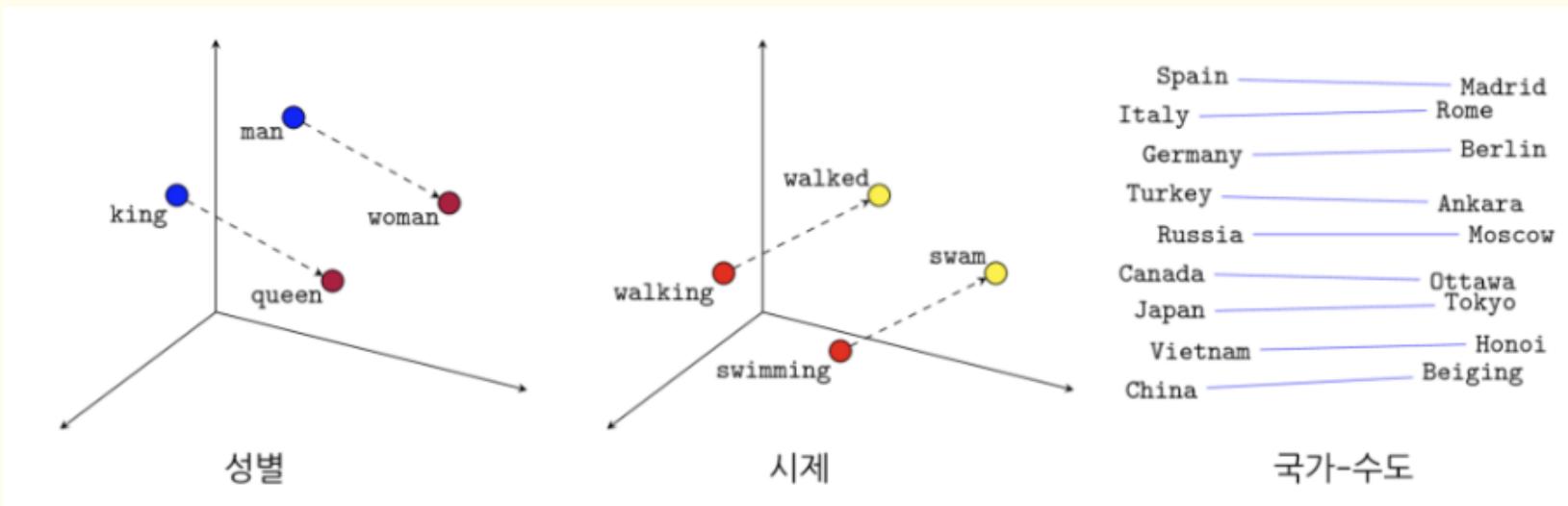
| 문서 | 갔다 | 나는 | 집에 | 학교에 | 회사에 |
|------|-----------|-----------|----------|----------|----------|
| 문서 1 | -0.287682 | -0.287682 | 0 | 0.405465 | 0 |
| 문서 2 | -0.287682 | -0.287682 | 0.405465 | 0 | 0 |
| 문서 3 | -0.287682 | -0.287682 | 0 | 0 | 0.405465 |

TF-IDF matrix

```
Tfidf_matrix = Tfidf.fit_transform(df_reviews['reviews'])
```

```
mmwrite('./models/Tfidf_cafe_review mtx', Tfidf_matrix)
```

04. TF-IDF and Word2vec - Word2vec

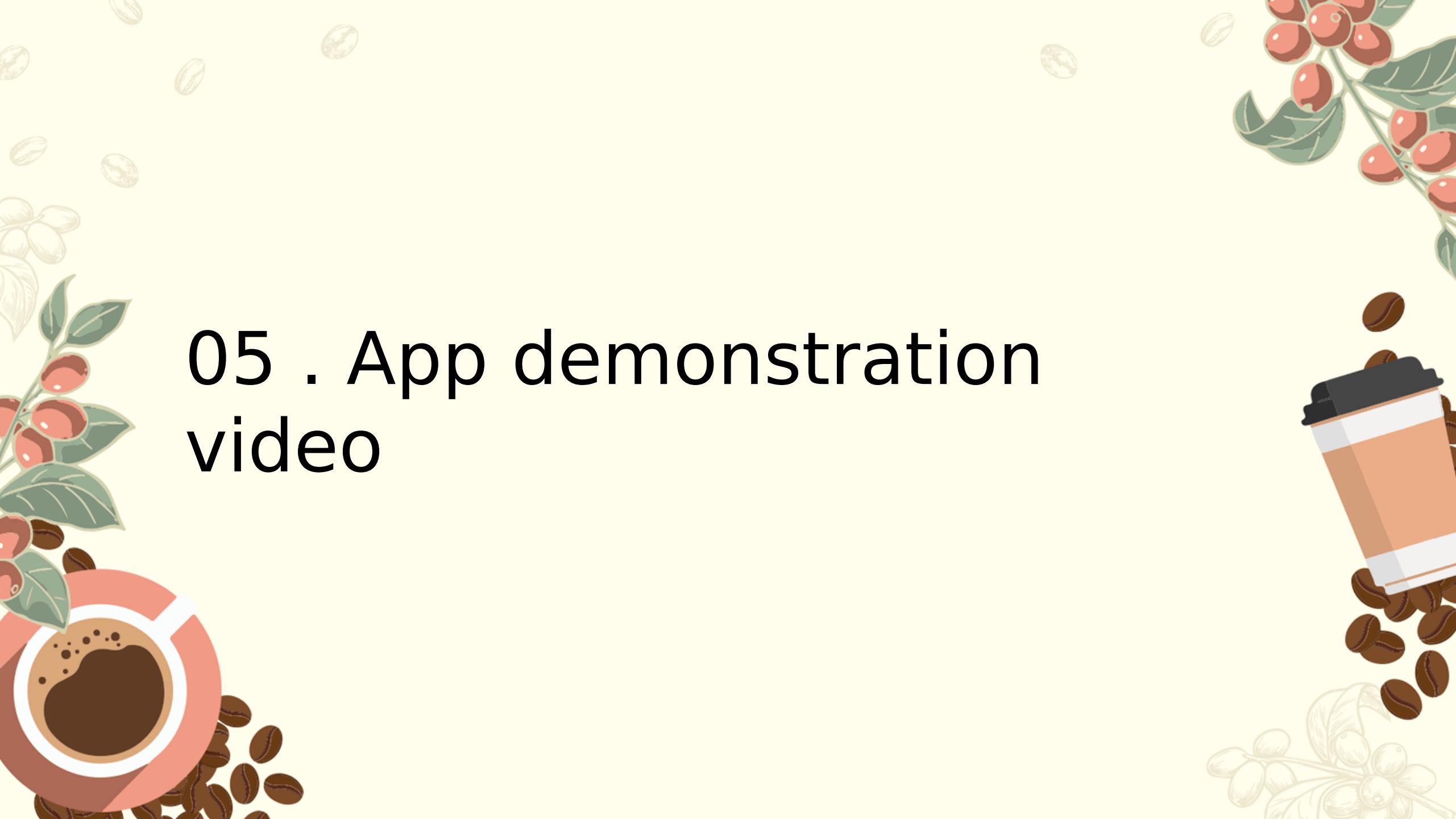


- 1) word2vec은 word를 다차원 벡터(vector)공간에 표현하여 벡터간의 유사도를 계산할 수 있게함
- 2) 앞뒤 단어를 고려하여 임베딩을 하기 때문에 단어의 문맥상의 의미까지 정량화된 벡터로 표현 가능

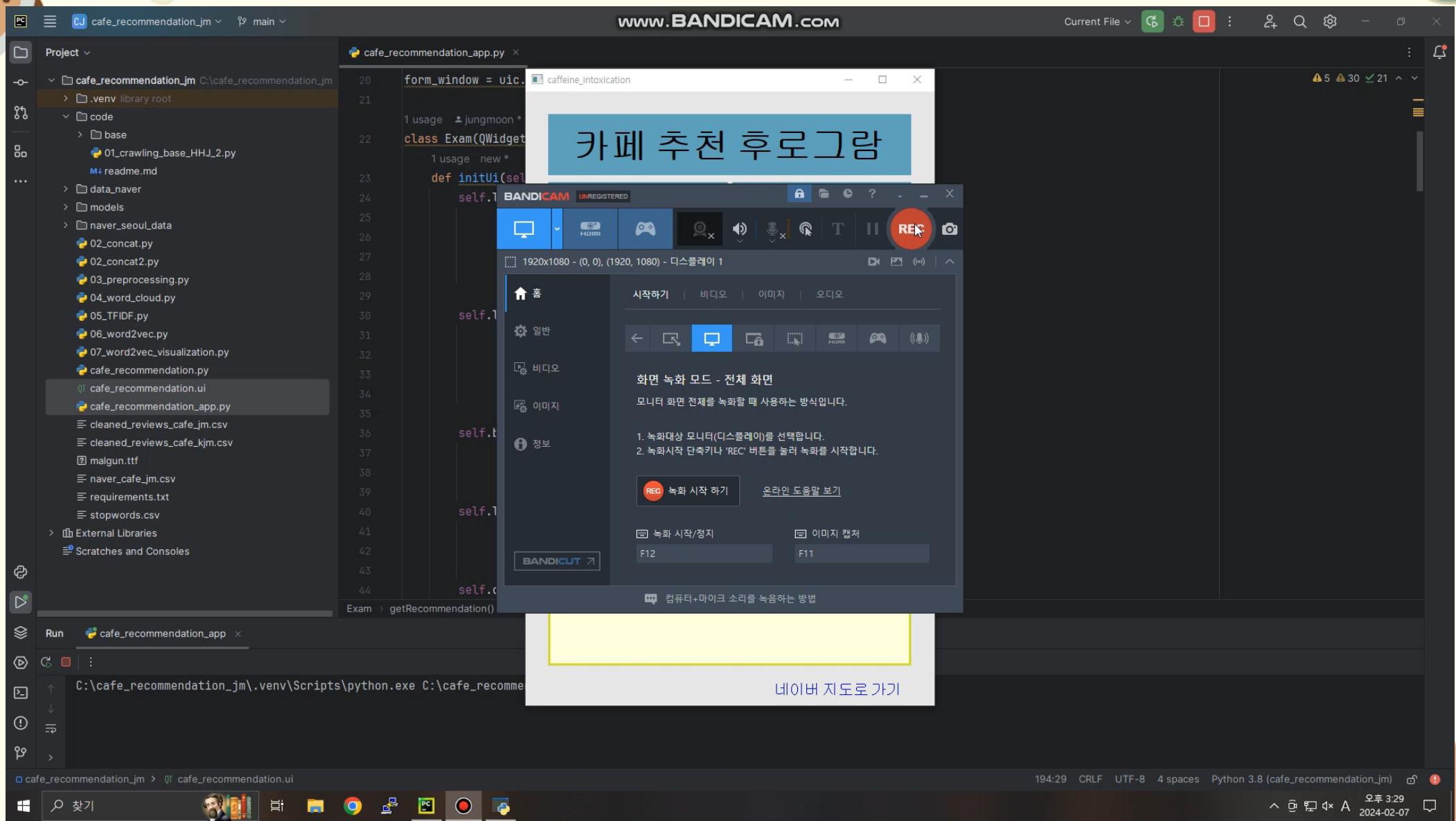
- 유사의미, 연상단어를 찾기위해 word2vec 사용.

```
embedding_model = Word2Vec(tokens, vector_size=100, window=4,  
                           min_count=20, workers=4, epochs=100, sg=1)
```

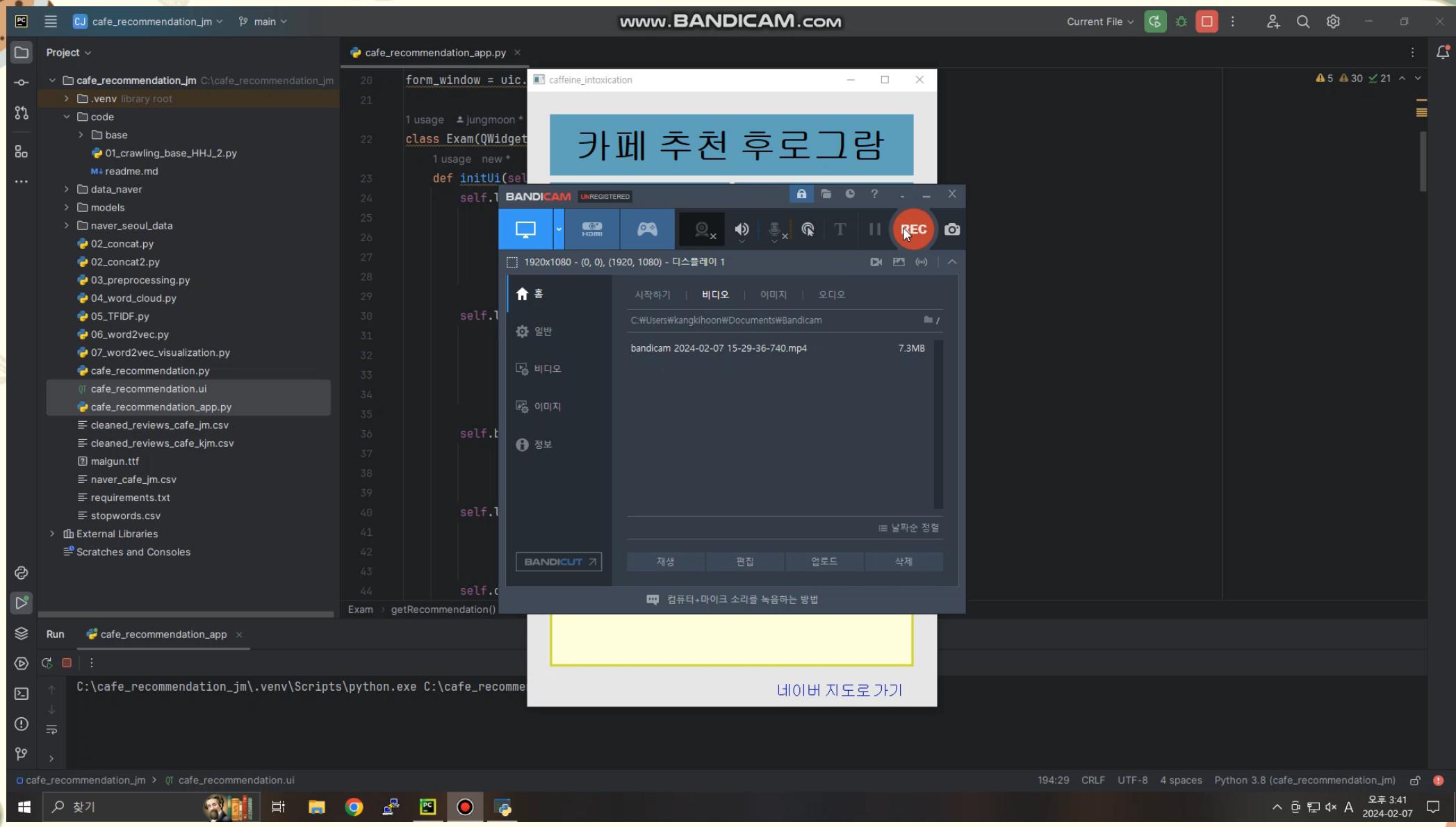
05 . App demonstration video



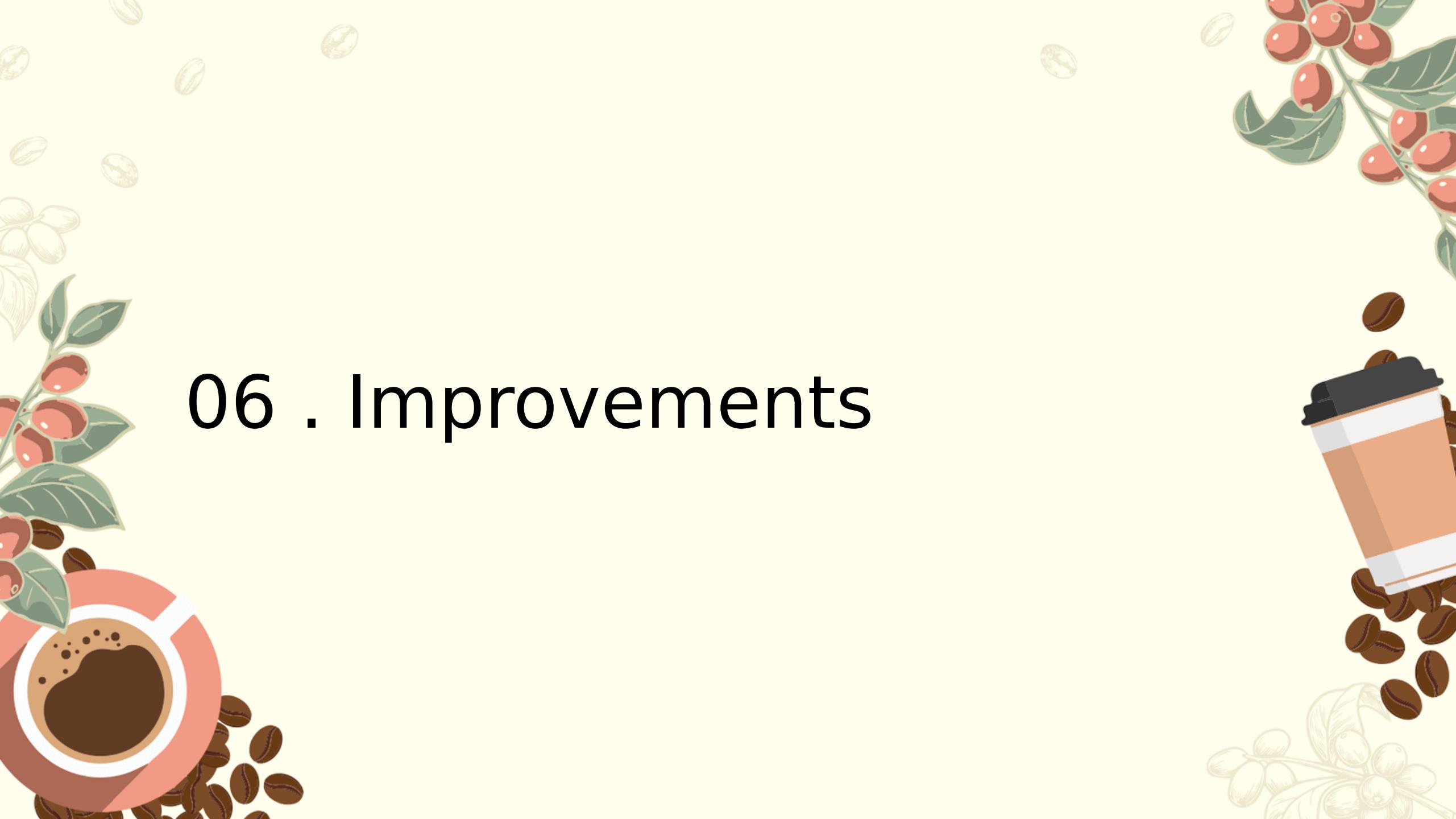
05. App demonstration video



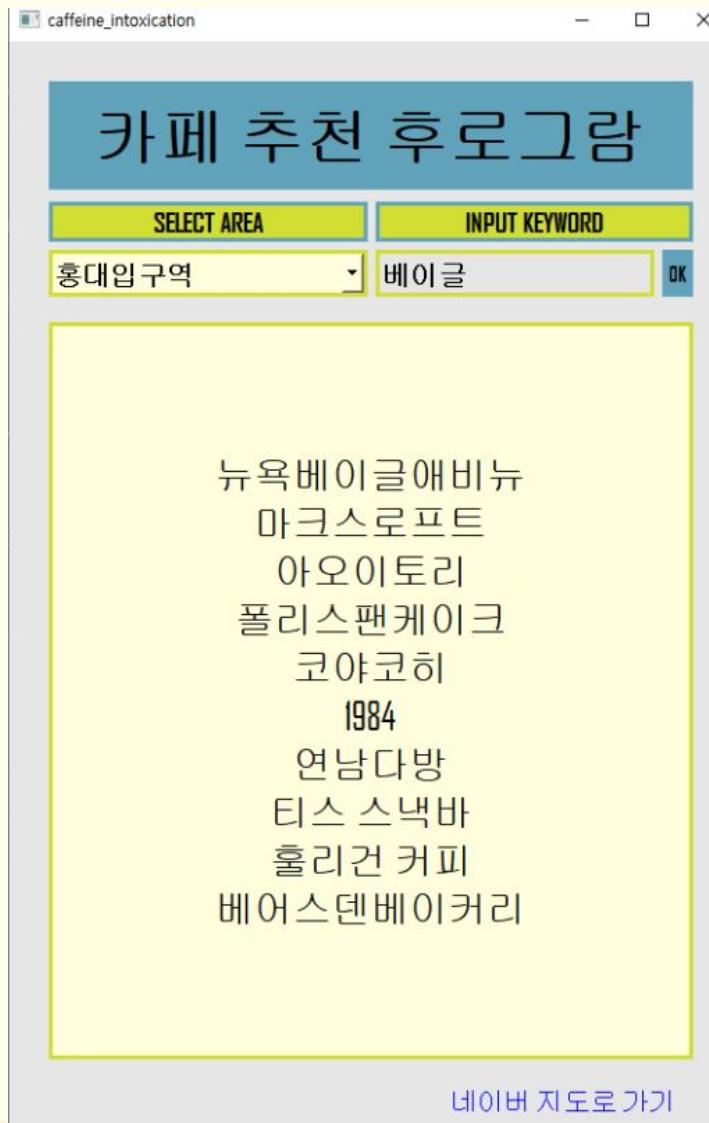
05. App demonstration video



06 . Improvements



06. Improvements



- Display café summary information
- Add café link

Thanks !



Q & A

