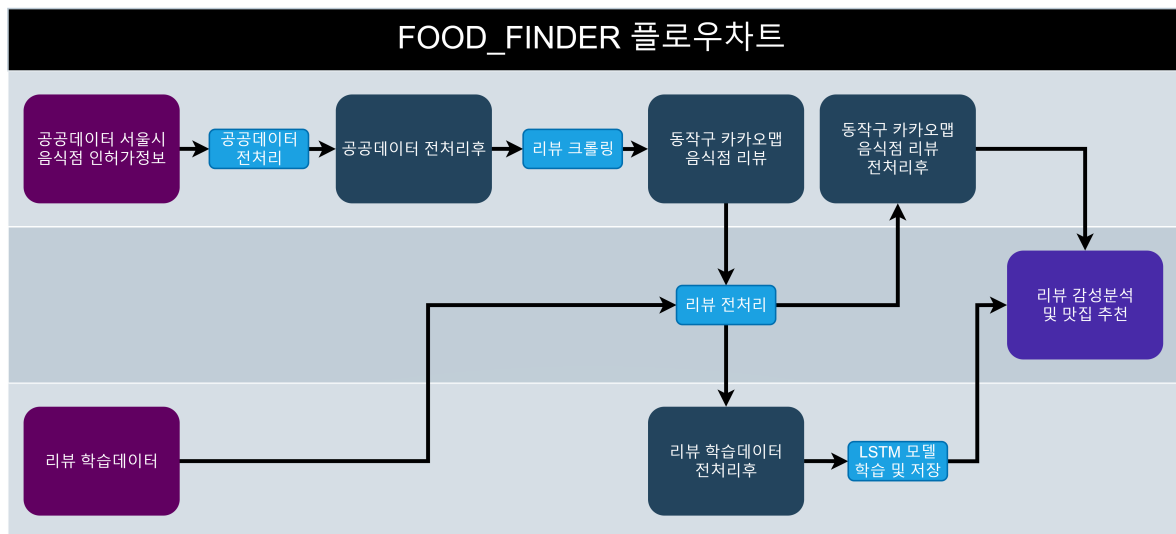


FOOD_FINDER(동작구)

동작구 음식점의 카카오맵 리뷰를 감성분석해서 맛집을 자동 추천해주는 시스템

0. 플로우차트 및 목차

플로우차트



목차

1. [공공데이터 및 학습데이터 다운로드](#)
 - [공공데이터 다운로드](#)
 - [리뷰 학습데이터 다운로드](#)
2. [공공데이터 전처리](#)
 - [사용할 컬럼만 남기고 나머지 제거](#)
 - [결측치 제거](#)
 - [폐업 음식점 제거](#)
 - [동작구 음식점 추출 및 결측치 제거](#)
 - [위도 경도 계산](#)
3. [카카오맵 음식점 리뷰 크롤링](#)
 - [리뷰 크롤링](#)
 - [결측치 제거](#)
4. [리뷰 학습 데이터로 LSTM 모델 학습](#)
 - [리뷰학습데이터 전처리](#)
 - [리뷰학습데이터 형태소 분리](#)
 - [리뷰학습데이터 토큰화 및 토큰나이저 저장](#)
 - [리뷰 LSTM학습 및 모델 저장](#)
5. [리뷰 감성분석 및 맛집 추천](#)
 - [동작구 음식점 리뷰 전처리](#)
 - [LSTM모델로 감성분석](#)

- [최종 맛집 추천!](#)

6. [참고문헌](#)

1. 공공데이터 및 학습데이터 다운로드

- [서울시 일반음식점 인허가 정보\(서울열린데이터광장\)](#)

서울시에 존재하는 모든 음식점 인허가 정보 데이터파일 다운로드

```
df = pd.read_csv("./src/서울특별시 일반음식점 인허가 정보.csv",
                  encoding='cp949')
print('공공데이터 크기:', df.shape)
df.head()
# 공공데이터 크기: (474939, 44)
```

	개방자치 단체코드	관리번호	인허가일 자	인허 가위 소일 자	영 업 상 태 코 드	영 업 상 태 명	상 세 영 업 상 태 코 드	상 세 영 업 상 태 명	폐업 일자	휴업 시작 일자	...	공장판 매직종 업원수	공장생 산직종 업원수	건물 소유 구분 명	보증 액	월세 액	다중 이용 업소 여부	시설총 규모	전통 업소 지정 번호	전통 업소 주된 음식	홈 이지
0	3000000	3000000-101-2001-11784	20010214	NaN	1	영업/정상	1	영업	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	N	34.56	NaN	NaN	NaN
1	3000000	3000000-101-2001-11802	20010214	NaN	1	영업/정상	1	영업	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	N	130.00	NaN	NaN	NaN
2	3200000	3200000-101-2020-00255	20200602	NaN	1	영업/정상	1	영업	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	3160000	3160000-101-2020-00141	20200602	NaN	1	영업/정상	1	영업	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	3220000	3220000-101-2020-00571	20200602	NaN	1	영업/정상	1	영업	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

- 리뷰 감성분석을 위한 학습데이터 다운로드

```
df = pd.read_csv('./src/train.csv')
print('리뷰 학습 데이터 크기:', df.shape)
df.head()
# 리뷰 학습 데이터 크기: (50000, 2)
```

		review	score
0	오늘 오니기리와 이규동 가서 돈가스랑 계란규동 먹었는데 진짜 합리적인 가격에 호강했...		5.0
1	여기는 미스터파파 임실치즈피자 인가 그렇습니다 저도 냅뎀네요 피자는 가격 저렴한 만...		2.0
2	숯불닭갈비가 그런맛이었다니 처음 먹어본 닭갈비 진짜맛있었어요 추천도가보고싶네요 정말강추		5.0
3	바하커피샵 전체적인 분위기는 작고 부담이 없다고 할까 하지만 커피맛이 좋은집 카페인...		3.0
4		폐북에서 봤는데 가관 장사 망	1.0

2. 공공데이터 전처리

1. 사용할 컬럼만 남기고 나머지 제거

```
df = df[['상세영업상태코드', '도로명주소', '사업장명', '좌표정보(X)', '좌표정보(Y)']]
df = df.reset_index(drop=True)
print('공공데이터 크기:', df.shape)
# 공공데이터 크기: (474939, 5)
```

2. 결측치 제거

```
df = df.dropna()
df = df.reset_index(drop=True)
print('공공데이터 크기:', df.shape)
# 공공데이터 크기: (222827, 5)
```

3. '상세영업상태코드' 1: 영업중 2: 폐업 -> 영업중인 음식점만 남기고 '상세영업상태코드' 컬럼 제거

```
df = df[df["상세영업상태코드"] == 1]
df = df[['도로명주소', '사업장명', '좌표정보(X)', '좌표정보(Y)']]
df = df.reset_index(drop=True)
print('공공데이터 크기:', df.shape)
# 공공데이터 크기: (121773, 4)
```

4. 도로명 주소에서 자치구를 분리한 후, 동작구만 추출

```
df['자치구'] = [i.split()[1] for i in df['도로명주소']]
df = df[df['자치구'] == '동작구']
df = df.reset_index(drop=True)
print('공공데이터 크기:', df.shape)
# 공공데이터 크기: (3168, 5)
```

5. 도로명 주소에서 자치동을 분리한 후, 결측치 제거

```
import re
dong_list = []
for address in df['도로명주소']:
    try:
        dong = re.findall('사당동|신대방동|흑석동|상도동|상도1동|대방동|노량진동', address)[0]
        if dong == '상도1동':
            dong = '상도동'
        dong_list.append(dong)
    except:
        dong_list.append('error')
df['자치동'] = dong_list
df = df[df['자치동'] != 'error']
df = df[['좌표정보(X)', '좌표정보(Y)', '자치구', '자치동', '사업장명']]
df = df.reset_index(drop=True)
print('공공데이터 크기:', df.shape)
# 공공데이터 크기: (3140, 5)
```

6. XY좌표계를 경도, 위도 좌표계로 변경

```
import pyproj
xy = np.array([[df['좌표정보(X)'][i], df['좌표정보(Y)'][i]] for i in range(df.shape[0])])
def project_array(coord, p1_type, p2_type):
    p1 = pyproj.Proj(init=p1_type)
    p2 = pyproj.Proj(init=p2_type)
    fx, fy = pyproj.transform(p1, p2, coord[:, 0], coord[:, 1])
    return np.dstack([fx, fy])[0]
p1_type = "epsg:2097"
p2_type = "epsg:4326"
```

```

result = project_array(xy, p1_type, p2_type)
df['경도'] = result[:, 0]
df['위도'] = result[:, 1]
df = df[['자치구', '자치동', '사업장명', '경도', '위도']]
print('공공데이터 크기:', df.shape)
df
# 공공데이터 크기: (3140, 5)

```

7. 최종 데이터

	자치구	자치동	사업장명	경도	위도
0	동작구	사당동	역전우동0410 & 뽕다방 동작남성점	126.971410	37.483504
1	동작구	사당동	한강치킨(수제전문점)	126.978409	37.480073
2	동작구	사당동	조가네갑오징어 사당점	126.977314	37.477970
3	동작구	상도동	태능솔밭갈비	126.945938	37.502914
4	동작구	사당동	면장의 밥상	126.975286	37.477573
...
3135	동작구	대방동	웰컴댕댕	126.929445	37.509096
3136	동작구	사당동	쉴(수)포차	126.978007	37.480737
3137	동작구	사당동	프레퍼스 다이어트 푸드 사당점	126.978444	37.480348
3138	동작구	신대방동	케이(K)호프	126.920478	37.491435
3139	동작구	신대방동	우동짜장&야식포차	126.904450	37.488106

3140 rows × 5 columns

3. 카카오맵 음식점 리뷰 크롤링

1. '자치구+자치동+사업장명'으로 검색하여 리뷰코멘트 주소 크롤링

```

import requests
import bs4
from bs4 import BeautifulSoup
id_list = []
for i in range(df.shape[0]):
    keyword = df['자치구'][i]+' '+df['자치동'][i]+' '+df['사업장명'][i]
    url = f'https://search.map.kakao.com/mapsearch/map.daum?
callback=jQuery18105963491453995977_1649742498249&q=
{keyword}&msFlag=A&sort=0'
    header = {
        'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/100.0.4896.75 Safari/537.36',
        'referer': 'https://map.kakao.com/'
    }
    response = requests.get(url, headers=header)
    try:
        confirmid = re.findall('confirmid':"([0-9]+)"', response.text)[0]
        review_address = f'https://place.map.kakao.com/{confirmid}'

```

```

id_list.append(review_address)
except:
    id_list.append('error')
print(f'\r{i+1}개 진행완료',end='')
# 3140개 진행완료

```

2. 카카오맵에 존재하지 않는 음식점 제거

```

df['리뷰주소'] = id_list
df = df[df['리뷰주소'] != 'error']
df = df.reset_index(drop=True)
print('공공데이터 크기:',df.shape)
# 공공데이터 크기: (2637, 6)

```

3. 카카오맵 리뷰주소

	자치구	자치동	사업장명	경도	위도	리뷰주소
0	동작구	사당동	역전우동0410 & 백다방 동작남성점	126.971410	37.483504	https://place.map.kakao.com/25116156
1	동작구	사당동	한강치킨(수제전문점)	126.978409	37.480073	https://place.map.kakao.com/11515257
2	동작구	상도동	태능술발갈비	126.945938	37.502914	https://place.map.kakao.com/11222981
3	동작구	사당동	면장의 밥상	126.975286	37.477573	https://place.map.kakao.com/1623686424
4	동작구	사당동	클로이스 칠하우스	126.964916	37.488247	https://place.map.kakao.com/505721130
...
2632	동작구	신대방동	마라홀릭 신대방	126.923730	37.498262	https://place.map.kakao.com/919062136
2633	동작구	상도동	모녀당	126.946396	37.503721	https://place.map.kakao.com/839496033
2634	동작구	상도동	양계장집아들 상도점	126.940679	37.498569	https://place.map.kakao.com/48395912
2635	동작구	대방동	웰컴댕댕	126.929445	37.509096	https://place.map.kakao.com/1591541759
2636	동작구	신대방동	우동짜장&야식포차	126.904450	37.488106	https://place.map.kakao.com/20604687

2637 rows × 6 columns

4. 카카오맵 리뷰

	name	score	review
0	역전우동0410 동작남성점	1	포장주문 했는데 집에 와서 확인하니 상품이 누락되어있어 연락하니 와서 찾아가라고 그...
1	역전우동0410 동작남성점	1	1년전쯤인가 일하시는 이모님 한국분에서 중국분으로 바뀌셨는데 맛이 없어짐.. 전 차...
2	역전우동0410 동작남성점	1	역전우동 수입곳을 다녀봤는데 여기가 제일 맛있어요..하필 집근처지점이라 더 화나네요
3	역전우동0410 동작남성점	1	오자마자 가위를 좌우로 휘저으면서 자리 옮기라고 하는건 아니죠...못 앉을 곳 앉았나요...
4	역전우동0410 동작남성점	1	맛없어요,,, 배고파서 먹었다
...
13645	홍부왕족발부홍보쌈	4	존맛임~ 이집 항상 사람들 바글바글 다 이유가 있음
13646	홍부왕족발부홍보쌈	1	아니 ㅋㅋㅋㅋ 족발 왜이래요??? 지방 90프로에 살 10프로??? 이곳은 추천하...
13647	홍부왕족발부홍보쌈	3	족발은 뽀뽀함. 맛있음. 근데 같이 나오는 국물이 예술임. 그것때문에 먹음. ...
13648	홍부왕족발부홍보쌈	1	여자들은 가면 안될듯요. 이렇게 얹게 썬 족발은 살다살다 처음 봐요. 특대 대 중...
13649	마망베베키즈카페	3	NaN

13650 rows × 3 columns

4. 리뷰 학습 데이터로 LSTM 모델 학습

1. 리뷰 학습 데이터 전처리

- 리뷰 학습 데이터 불러오기

```
df = pd.read_csv('./src/train.csv')
print('리뷰 학습 데이터 크기:', df.shape)
# 리뷰 학습 데이터 크기: (50000, 2)
```

- 별점을 1과 0으로 라벨링

```
def score_evaluate(score):
    if score >= 3.5:
        return 1
    else:
        return 0
df['score'] = df['score'].apply(score_evaluate)
```

- 한글을 제외한 초성, 특수문자, 영어 제거

```
df['review'] = df['review'].str.replace('[^가-힣]', ' ').str.replace(' ', '+', ' ')
```

	review	score
0	오늘 오니기리와 이규동 가서 돈가스랑 계란규동 먹었는데 진짜 합리적인 가격에 호강했...	1
1	여기는 미스터파파 임실치즈피자 인가 그렇습니다 저도 낚였네요 피자는 가격 저렴한 만...	0
2	숯불닭갈비가 그런맛이었다니 처음 먹어본 닭갈비 진짜맛있었어요 춘천또가고싶네요 정말강추	1
3	바하커피샵 전체적인 분위기는 작고 부담이 없다고 할까 하지만 커피맛이 좋은집 카페인...	0
4	페북에서 봤는데 가관 장사 망	0

2. 리뷰 학습 데이터 형태소 분리

- konlpy okt 모듈을 이용해서 형태소 분리

```
from konlpy.tag import Okt

tag_list = ['Noun', 'Verb', 'Adjective', 'VerbPrefix']
okt = Okt()
tokenized_data = []
for i in range(df.shape[0]):
    tokenized_sentence = okt.pos(df['review'][i], stem=True) # 토큰화
    tag_checked_sentence = []
    for j in tokenized_sentence:
        x, y = j
        if y in tag_list:
            tag_checked_sentence.append(x)
    print(f'\r{i+1}개 형태소분석 완료', end='')
    tokenized_data.append(tag_checked_sentence)
df['토큰화뿔글'] = tokenized_data
# 50000개 형태소분석 완료
```

- 단어 갯수가 3개 이하의 리뷰 제거


```
df = df[df['토큰화댓글'].str.len() > 3]
df = df.reset_index(drop=True)
print('리뷰 학습 데이터 크기:', df.shape)
# 리뷰 학습 데이터 크기: (48817, 3)
```

	review	score	토큰화댓글
0	오늘 오니기리와 이규동 가서 돈가스랑 계란규동 먹었는데 진짜 합리적인 가격에 호강했...	1	[오늘, 오니기리, 이규동, 가다, 돈가스, 계란, 규동, 먹다, 진짜, 합리, 가...
1	여기는 미스터파파 임실치즈피자 인가 그렇습니다 저도 낫였네요 피자는 가격 저렴한 만...	0	[여기다, 미스터, 파파, 임실, 치즈, 피자, 그렇다, 저, 낫였네, 피자, 가...
2	숯불닭갈비가 그런맛이었다니 처음 먹어본 닭갈비 진짜맛있었어요 춘천또가고싶네요 정말강추	1	[숯불, 닭갈비, 맛, 이다, 처음, 먹다, 보다, 닭갈비, 진짜, 맛있다, 춘천, ...
3	바하커피샵 전체적인 분위기는 작고 부담이 없다고 할까 하지만 커피맛이 좋은집 카페인...	0	[바하, 커피샵, 전체, 분위기, 작고, 부담, 없다, 하다, 커피, 맛, 좋다, ...
4	페북에서 봤는데 가관 장사 망	0	[페북, 보다, 가관, 장사, 망]

3. 리뷰 학습 데이터 토큰화 및 토큰나이저 저장

- 리뷰 학습 데이터 토큰화

```
from tensorflow.keras.preprocessing.text import Tokenizer

tokenizer = Tokenizer(num_words=40000, oov_token = True)
tokenizer.fit_on_texts(df['토큰화댓글'])
df['토큰'] = tokenizer.texts_to_sequences(df['토큰화댓글'])
```

	review	score	토큰화댓글	토큰
0	오늘 오니기리와 이규동 가서 돈가스랑 계란규동 먹었는데 진짜 합리적인 가격에 호강했...	1	['오늘', '오니기리', '이규동', '가다', '돈가스', '계란', '규동', ...]	[135, 7564, 16042, 7, 770, 1450, 16043, 3, 19, ...]
1	여기는 미스터파파 임실치즈피자 인가 그렇습니다 저도 낫였네요 피자는 가격 저렴한 만...	0	['여기다', '미스터', '파파', '임실', '치즈', '피자', '그렇다', ...]	[127, 3365, 9170, 5242, 218, 118, 37, 80, 8224, ...]
2	숯불닭갈비가 그런맛이었다니 처음 먹어본 닭갈비 진짜맛있었어요 춘천또가고싶네요 정말강추	1	['숯불', '닭갈비', '맛', '이다', '처음', '먹다', '보다', '닭갈...', ...]	[717, 595, 4, 14, 75, 3, 15, 595, 19, 5, 976, ...]
3	바하커피샵 전체적인 분위기는 작고 부담이 없다고 할까 하지만 커피맛이 좋은집 카페인...	0	['바하', '커피샵', '전체', '분위기', '작고', '부담', '없다', '...', ...]	[8225, 16044, 741, 91, 1173, 721, 8, 2, 156, 4, ...]
4	페북에서 봤는데 가관 장사 망	0	['페북', '보다', '가관', '장사', '망']	[4017, 15, 2136, 72, 2523]

- 토큰나이저 저장

```
import pickle

with open('./src/tokenizer.pickle', 'wb') as handle:
    pickle.dump(tokenizer, handle, protocol=pickle.HIGHEST_PROTOCOL)
```

4. 리뷰 학습 데이터 LSTM 학습 및 모델 저장

```
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import load_model
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense, Embedding, LSTM
from sklearn.preprocessing import LabelEncoder

train = pad_sequences(df['토큰'], maxlen=20)
label = df['score']
encoder = LabelEncoder()
batch_size = label.shape[0]
input_dim = 1
label = encoder.fit_transform(label)
label = np.reshape(label, (batch_size, input_dim))
model = Sequential()
model.add(Embedding(40000, 128))
model.add(LSTM(128, dropout=0.2, recurrent_dropout=0.2))
```

```

model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=
['accuracy'])
hist = model.fit(train, label, batch_size=32, epochs=5)
model.save('./src/model.h5')

```

```

Epoch 1/5
1526/1526 [=====] - 222s 143ms/step - loss: 0.3000 - accuracy: 0.8829
Epoch 2/5
1526/1526 [=====] - 193s 127ms/step - loss: 0.2244 - accuracy: 0.9195
Epoch 3/5
1526/1526 [=====] - 225s 147ms/step - loss: 0.1828 - accuracy: 0.9356
Epoch 4/5
1526/1526 [=====] - 209s 137ms/step - loss: 0.1473 - accuracy: 0.9480
Epoch 5/5
1526/1526 [=====] - 204s 133ms/step - loss: 0.1154 - accuracy: 0.9594

```

5. 리뷰 감성분석 및 맛집 추천

1. 동작구 카카오맵 음식점 리뷰 전처리

- 동작구 카카오맵 리뷰 데이터 불러오기

```

test_df = pd.read_csv('./src/dongjack_review.csv')
test_df = test_df[['name', 'review']]
print('리뷰 학습 데이터 크기:', test_df.shape)
# 리뷰 학습 데이터 크기: (13650, 2)

```

- 리뷰 데이터 한글제외 초성, 특수문자, 영어 제거

```

test_df['review'] = test_df['review'].str.replace('[^가-힣]', ' ')
test_df['review'] = test_df['review'].str.replace(' +', ' ')

```

- 리뷰 데이터 형태소 분리

```

from konlpy.tag import Okt

tag_list = ['Noun', 'Verb', 'Adjective', 'VerbPrefix']
okt = Okt()
tokenized_data = []
for i in range(test_df.shape[0]):
    tokenized_sentence = okt.pos(str(test_df['review'][i]), stem=True) # 토큰
    화
    tag_checked_sentence = []
    for j in tokenized_sentence:
        x, y = j
        if y in tag_list:
            tag_checked_sentence.append(x)
    print(f'\r{i+1}개 형태소분석 완료', end='')
    tokenized_data.append(tag_checked_sentence)
for i in tokenized_data:
    for j in range(len(i)):
        i[j] = ""+i[j]+" "
test_df['토큰화맷글'] = tokenized_data
print('리뷰 학습 데이터 크기:', test_df.shape)
test_df.head()
# 13650개 형태소분석 완료
# 리뷰 학습 데이터 크기: (13650, 3)

```


- 리뷰 길이가 3이하의 리뷰 제거

```
test_df = test_df[test_df['토큰화맷글'].str.len() > 3]
test_df = test_df.reset_index(drop=True)
print('리뷰 학습 데이터 크기:', test_df.shape)
# 리뷰 학습 데이터 크기: (9662, 3)
```

	name	review	토큰화맷글
0	역전우동0410 동작남성점	포장주문 했는데 집에 와서 확인하니 상품이 누락되어있어 연락하니 와서 찾아가라고 그...	['포장', '주문', '하다', '집', '오다', '확인', '하다', '상품', ...]
1	역전우동0410 동작남성점	년전쯤인가 일하시는 이모님 한국분에서 중국분으로 바뀌셨는데 맛이 없어짐 전 차별하...	['년', '쯤', '일', '하다', '이모', '한국', '중국', '바뀌다', ...]
2	역전우동0410 동작남성점	역전우동 수십곳을 다녀봤는데 여기가 제일 맛있어요 하필 집근처지점이라 더 화나네요	['역전', '우동', '곳', '다니다', '보다', '여기', '제일', '맛없', ...]
3	역전우동0410 동작남성점	오자마자 가위를 좌우로 휘저으면서 자리 옮기라고 하는건 아니죠 못 앉을 곳 앉았나요...	['자마자', '가위', '좌우', '휘저', '면서', '자리', '옮기다', '못', ...]
4	한강치킨수제전문점 본점	튀김옷 얇고 쫄깃함	['튀김', '옷', '얇다', '쫄깃함']

2. LSTM모델로 감성분석

- 토큰나이저 불러오기 및 토큰화

```
import pickle

with open('./src/tokenizer.pickle', 'rb') as handle:
    tokenizer = pickle.load(handle)
tokenized_data = test_df['토큰화맷글']
test = tokenizer.texts_to_sequences(tokenized_data)
test = pad_sequences(test, maxlen=20)
test
```

```
array([[4190, 198, 1035, ..., 1404, 2, 29],
       [ 123, 2, 638, ..., 540, 2, 666],
       [ 0, 0, 0, ..., 471, 46, 959],
       ...,
       [ 0, 0, 0, ..., 58, 2, 21],
       [ 0, 0, 0, ..., 5, 94, 5],
       [ 150, 529, 1805, ..., 1092, 1183, 289]])
```

- 모델 불러오기 및 분석

```
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing.sequence import pad_sequences

model = load_model('./src/model.h5')
pred = model.predict(test)
test_df['score'] = pred
df = test_df[['name', 'score']]
df
```

	name	score
0	역전우동0410 동작남성점	0.010379
1	역전우동0410 동작남성점	0.038175
2	역전우동0410 동작남성점	0.005100
3	역전우동0410 동작남성점	0.123762
4	한강치킨수제전문점 본점	0.002647

3. 최종 맛집 추천

- 가게별 리뷰 갯수 및 리뷰 스코어 평균 측정

```
df = df.groupby('name').agg(review_count=('score', 'size'), score_mean=
('score', 'mean')).reset_index()
print('최종 리뷰데이터 크기:', test_df.shape)
df.head()
# 최종 리뷰데이터 크기: (1490, 3)
```

	name	review_count	score_mean
0	1994안나	1	0.997245
1	20세기청년	1	0.992970
2	28청춘호프포차	2	0.964859
3	37.5 동작이수점	20	0.591296
4	60계 서울이수점	10	0.302207

- 리뷰 갯수가 5개 이상인 가게 및 리뷰스코어 0.9점 이상 내림차순 정렬

```
df2 = df[df['review_count'] > 4][df['score_mean'] >
0.9].sort_values(by='score_mean', ascending=False).reset_index(drop=True)
df2
```

	name	review_count	score_mean
0	청와대소금구이	8	0.998295
1	죽발의발견	8	0.998156
2	동화장	8	0.997453
3	정정아식당 중앙대직영점	6	0.997325
4	커피가지고요	8	0.996954
...
126	쌈반	12	0.909257
127	나드리분식	8	0.908545
128	참숯옛맛서울불고기	15	0.903804
129	리얼후라이 송실대점	8	0.903009
130	명갈비 상도본점	6	0.902476

131 rows × 3 columns

- 최종 동작구 리뷰 스코어 상위 10개 맛집 리스트!

	name	review_count	score_mean
0	청와대소금구이	8	0.998295
1	죽발의발견	8	0.998156
2	동화장	8	0.997453
3	정정아식당 중앙대직영점	6	0.997325
4	커피가지고요	8	0.996954
5	꼬멧 익스프레스	6	0.996615
6	마루	8	0.996611
7	프리미엄만화카페 잼툰	6	0.996508
8	곱창의전설	8	0.996428
9	흥부네아구찜	8	0.996108

6. 참고문헌

- 맛집지도 시각화 - 카카오맵 크롤링 방법 : <https://nostalgiaa.tistory.com/36>

- 맛집프로젝트 - 카카오맵 크롤링 방법 : <https://cordingdoah.tistory.com/97>
- 맛집 분석 : <https://cottonwood-moa.tistory.com/79>
- 구글 api를 사용하여 서울시 맛집리스트 시각화하기 : <https://dacon.io/codeshare/2638>
- 녹차라떼 맛집 베스트30 크롤링/시각화 : <https://velog.io/@hangnii/greenteabest30>
- 성북구 음식점 평점 리뷰 크롤링하기 : <https://eochodevlog.tistory.com/17>
- 음식점 검색 프로그램 (1): 데이터셋&환경구축 : <https://blog.naver.com/kyy0810/221494376698>
- 엘라스틱 서치 : <https://blog.naver.com/kyy0810/221495323064>
- 검색/결과 화면 구성 : <https://blog.naver.com/kyy0810/221495979970>
- OpenAPI 활용 - 날씨 정보를 사용한 맛집 추천 프로젝트 : <https://ai-creator.tistory.com/31>
- 강남역 맛집 리뷰 감성 분류 : <https://speedanddirection.tistory.com/m/94-> <https://www.youtube.com/watch?v=Kzlc3TDN9JY>
- 맛집 리뷰를 통한 감성분석(긍정/부정 언어 판별): <https://moojuksublime.tistory.com/14>
- 고객 방문 후기 평점 감성분석 : <http://aispiration.com/text/nlp-text-sentiment-yelp.html4>.
- 우리 동네 맛집추천(크롤링과 코사인 유사도) : <https://data101.oopy.io/recommendation-engine-cosine-similarity>
- 요기요 음식점 크롤링(코드 오류 있음) : <https://scareddev.tistory.com/3>
- 카카오 맵 리뷰 크롤링 (깃허브) : https://github.com/wlgh325/python_crawling
- 세미프로젝트 공유 드라이브 : <https://drive.google.com/drive/folders/1MeadyLQRxVak7XgxYtBEE6JMIo5WTPwr?usp=sharing>