



인스타그램 데이터 분석

음식 인스타그램 계정 분석



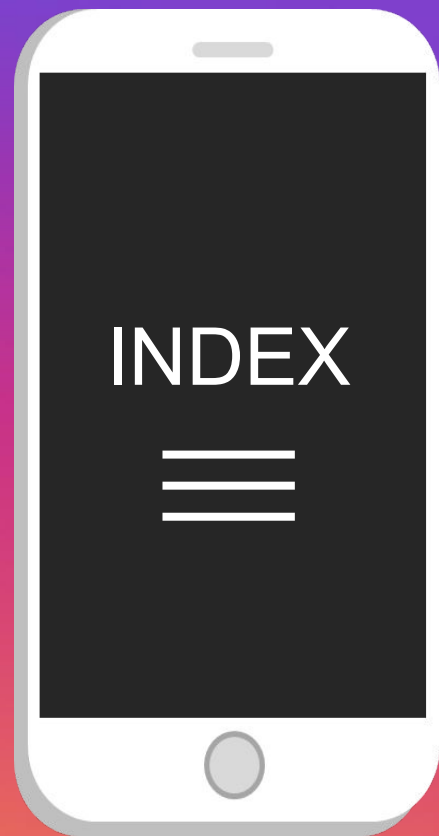
이 광평



김 도희



이 현민



1 데이터 수집
크롤링

2 데이터 전처리
Pandas
Okt

3 시각화 및 탐색
Bar chart
Word Cloud

4 결론 및 제언



데이터 수집

크롤링



atozzang_g

133k
Share

1.3m
Reaction

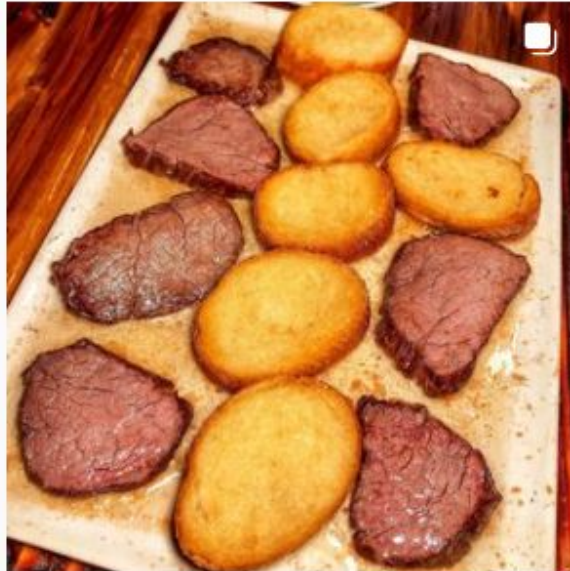
#아토짱 9년째 먹스타그램 中

📖 게시물

📺 릴스

🎥 동영상

🏠 태그됨





데이터 수집 크롤링

```
# 크롤 시작
driver = webdriver.Chrome('chromedriver.exe')
driver.get('https://www.instagram.com')
time.sleep(4)

#인스타그램 로그인
email = ' '
input_id = driver.find_elements_by_css_selector('input._2hvTZ.pexuQ.zyHYP')[0]
input_id.clear()
input_id.send_keys(email)

password = ' '
input_pw = driver.find_elements_by_css_selector('input._2hvTZ.pexuQ.zyHYP')[1]
input_pw.clear()
input_pw.send_keys(password)
input_pw.submit()
time.sleep(5)

#게시물 조회할 검색어
word = input('검색어 : ')

word = str(word)
url = insta_searching(word)

# 검색 결과 페이지 열기
driver.get(url)
time.sleep(np.random.uniform(3,8))

#첫 번째 게시물 클릭
select_first(driver)

# 데이터 수집 시작
resu = []
target = 900

for i in range(target):
    try:
        data = get_content(driver)
        resu.append(data)
        move_next(driver)
    except:
        time.sleep(np.random.uniform(3,11))
        move_next(driver)

print(resu)
```

크롤링

Beautiful Soup를 사용해 원하는 검색어로 데이터 추출

num(게시물), content(게시물 내용), date(날짜), like(좋아요),
place(장소), tags(해시태그) 순으로 크롤링한 데이터를 엑셀
파일로 저장

	A	B	C	D	E	F	G	H	I	
1		num	content	date	like	place	tags			
2	0	0	☹ 코로나	2021-07-2	882	망원동	['#코끝집', '#망원역', '#해진뒤', '#망원			
3	1	1	☹ 코로나	2021-07-2	2,419	면뽕는선상	['#코끝집', '#제주', '#한림', '#아길라호			
4	2	2	#광고 전	2021-07-2	2,441	Jeju	['#광고', '#제주도', '#회심', '#고등어참			
5	3	3	☹ 코로나	2021-07-2	4,028	제주시 이	['#코끝집', '#마구로쇼쿠도', '#성수동			
6	4	4	☹ 코로나	2021-07-2	3,362	월정리해변	['#코끝집', '#제주도', '#월정리해변', '#			

2

데이터 전처리



데이터 전처리 'content' 전처리

```
#content 분석

#content 전처리
#1
for i in dfs:
    prepro_content(i)
#2
sentences_pos = []
vocab = []
for i in dfs:
    tempS, tempV = make_vocab(i)
    sentences_pos.append(tempS)
    vocab.append(tempV)
#3
vocab_value_counts = []
for i in vocab:
    vocab_value_counts.append(value_counts_num(i, 15))
```

main 문

```
def make_vocab(dataframe):
    okt = Okt()

    sentences_pos = []
    for line in dataframe['content']:
        sentences_pos.append(okt.nouns(line))

    max_len = max([len(i) for i in sentences_pos])

    vocab = []
    for line in sentences_pos:
        for word in line:
            vocab.append(word)

    vocab_size = len(vocab) + 1
    vocab = sorted(list(vocab))

    vocab = [item for item in vocab if len(item) != 1]

    return sentences_pos, vocab
```

함수 구문

1. 결측값을 'nan'으로 바꿔준다.
2. okt모듈을 사용해서 'content'컬럼의 명사를 추출해준다.
3. 추출한 명사의 빈도수를 구해준다.



데이터 전처리
'content' 전처리

	0	1		0	1		0	1		0	1		0	1
0	맛집	531	0	정말	1314	0	맛집	363	0	오늘	124	0	구매	316
1	주문	334	1	소스	414	1	주문	264	1	느낌	69	1	진짜	274
2	메뉴	276	2	맛집	395	2	역시	200	2	소개	62	2	고기	114
3	서울	257	3	치즈	372	3	치킨	197	3	살짝	57	3	떡볶이	111
4	정도	252	4	추천	281	4	느낌	178	4	달달	52	4	맛집	104
5	소스	247	5	새우	252	5	오늘	177	5	제품	51	5	여기	104
6	위치	243	6	고기	246	6	가격	160	6	생각	50	6	치즈	103
7	추천	235	7	달달	243	7	고고	158	7	음식	45	7	바삭	101
8	출구	210	8	매콤	235	8	존맛	147	8	매콤	44	8	양념	101
9	매일	209	9	후기	233	9	바로	145	9	진짜	41	9	달달	100
10	거리	203	10	양념	224	10	가게	141	10	소스	40	10	하나	100
11	국물	187	11	식감	217	11	메뉴	140	11	아주	38	11	조합	98
12	도보	182	12	떡볶이	214	12	진짜	132	12	고기	37	12	먹기	96
13	바로	177	13	진짜	202	13	제대로	129	13	크림	36	13	서울	78
14	고기	159	14	솔직	200	14	고기	120	14	연어	33	14	정말	78

ato

bbo

haewon

twomuk

mukcong

인스타 내용에 어떤 단어들을 자주 사용하는지 알 수 있다. 보통 맛집, 고기 등의 단어가 공통적으로 쓰이는걸 볼 수 있다.



#태그분석

#태그 전처리

```
#1
for i in dfs:
    rename_Unnamed(i)
```

#태그수 보기

```
tag_value_counts = []
for i in dfs:
    tag_value_counts.append(value_counts_num(i['tags'], 15))
```

#워드클라우드 용 text파일

```
text_list = []
for i in dfs:
    text_list.append(make_all_text(i, 'tags'))
```

main 문

```
def value_counts_num(listOrSeries, num):
    if type(listOrSeries) == list:
        result = pd.DataFrame(Counter(listOrSeries).most_common(num))
    else:
        temp = make_tags_list(listOrSeries)
        result = pd.DataFrame(Counter(temp).most_common(num))
    return result
```

```
def make_all_text(dataFrame, columnName):
    dataFrame.reset_index(drop=True, inplace=True)
    result = ""

    if dataFrame[columnName][0][0] == '[':
        for i in dataFrame.index:
            text = " ".join(dataFrame[columnName][i][1:-1].replace(' ', '').replace('`', '').split(','))
            result = result + " " + text

    else:
        for i in dataFrame.index:
            text = " ".join(dataFrame[columnName][i])
            result = result + " " + text

    return result
```

함수 구문

1. Unnamed 컬럼을 num으로 바꿔준다.
2. 태그 빈도수 별로 빈도수를 구해준다
3. word cloud를 만들기 위해 태그들을 string으로 바꿔준다.



데이터 전처리 'tags' 전처리

0 1		0 1		0 1		0 1		0 1	
0	#광고 60	0	#먹스타그램 46	0	#원주맛집 80	0	#협찬 92	0	#협찬 200
1	#떡볶이맛집 54	1	#간편식 46	1	#yummy 78	1	#협찬 64	1	#먹스타맛팔 194
2	#떡볶이 53	2	#고기맛집 44	2	#foodie 78	2	#fff 59	2	#먹콩_택배 117
3	#코골집 51	3	#간편식추천 42	3	#food 78	3	#daily 58	3	#먹스타맛팔 111
4	#홍대맛집 46	4	#홍대맛집 41	4	#foodstagram 78	4	#food 54	4	#먹스타그램 78
5	#존맛 45	5	#연남동맛집 38	5	#instafood 78	5	#cu덕후 48	5	#먹콩_간식 76
6	#연남동맛집 41	6	#홍국 38	6	#foodpics 78	6	#일상 39	6	#먹팔 53
7	#먹스타그램 41	7	#먹팔로우 37	7	#foodgasm 77	7	#foodstagram 34	7	#광고 48
8	#제주도맛집 37	8	#안주맛집 35	8	#foodporn 67	8	#음식 34	8	#협찬 46
9	#아토짱_제주 35	9	#아이들간식 33	9	#mukbang 64	9	#먹스타그램 33	9	#먹스타그램 44
10	#노포맛집 32	10	#요리레시피 30	10	#먹스타 53	10	#먹팔 33	10	#먹콩_택배 42
11	#제주도 30	11	#홍국스타그램 28	11	#먹스타그램 53	11	#선팔 33	11	#먹콩픽 🍌 40
12	#koreanfood 29	12	#택배맛집 28	12	#맛스타 53	12	#맛팔 33	12	#땡큐파머스 33
13	#신촌맛집 28	13	#디저트맛집 27	13	#맛스타그램 53	13	#선팔하면맛팔 33	13	#줄반 33
14	#노포투어 28	14	#반찬추천 26	14	#먹부림 53	14	#먹스타맛팔 32	14	#먹콩_떡볶이 32

ato bbo haewon twomuk mukcong

#먹스타그램 이란 태그를 다들 자주 사용하고 있다. 그리고 주의깊게 봐야할 태그가 #광고와 #협찬이다. 딱 봐도 많이 사용하는것으로 보인다.

3

시각화 및 탐색

Bar Chart

Word Cloud



데이터 시각화 Bar chart

#content 시각화
#barplot

```
fig = plt.figure(figsize = (20,20))
fig.add_subplot(2, 3, 1)
barh(vocab_value_counts[0][0], vocab_value_counts[0][1])
fig.add_subplot(2, 3, 2)
barh(vocab_value_counts[1][0], vocab_value_counts[1][1])
fig.add_subplot(2,3,3)
barh(vocab_value_counts[2][0], vocab_value_counts[2][1])
fig.add_subplot(2,3,4)
barh(vocab_value_counts[3][0], vocab_value_counts[3][1])
fig.add_subplot(2,3,5)
barh(vocab_value_counts[4][0], vocab_value_counts[4][1])
plt.show()
```

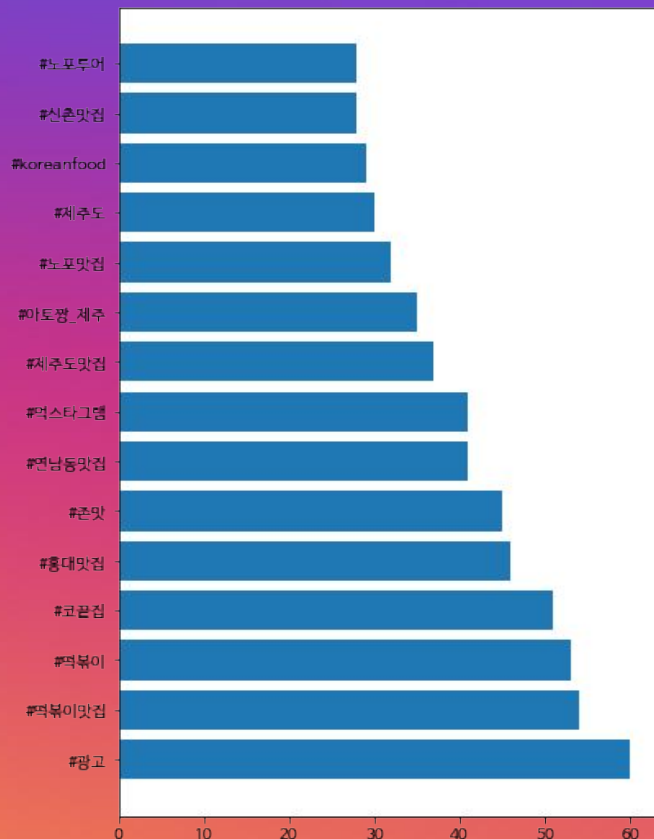
#tag 시각화
#barplot

```
fig = plt.figure(figsize = (20,20))
fig.add_subplot(2, 3, 1)
barh(tag_value_counts[0][0], tag_value_counts[0][1])
fig.add_subplot(2, 3, 2)
barh(tag_value_counts[1][0], tag_value_counts[1][1])
fig.add_subplot(2,3,3)
barh(tag_value_counts[2][0], tag_value_counts[2][1])
fig.add_subplot(2,3,4)
barh(tag_value_counts[3][0], tag_value_counts[3][1])
fig.add_subplot(2,3,5)
barh(tag_value_counts[4][0], tag_value_counts[4][1])
plt.show()
```

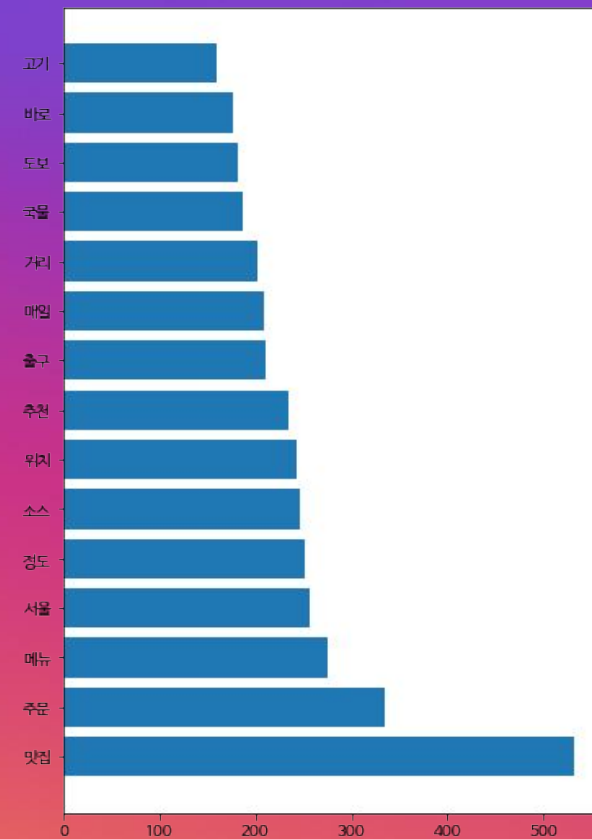
content, tags 컬럼의 단어들의 사용 빈도수를 이용하여 bar차트를 만들어준다.



데이터 시각화 Bar chart



Ato의 tags



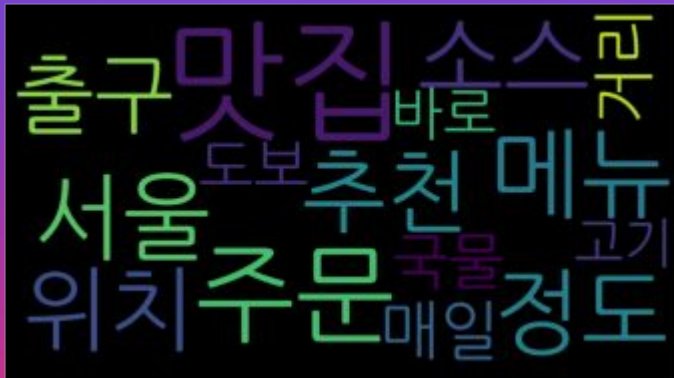
Ato의 content

ato의 계정의 특징은 맛집, 주문, 메뉴, 위치, 추천 등의 단어를 자주 사용한다.
가게의 정보를 중심으로 게시글을 올리는 것을 확인 할 수 있다.

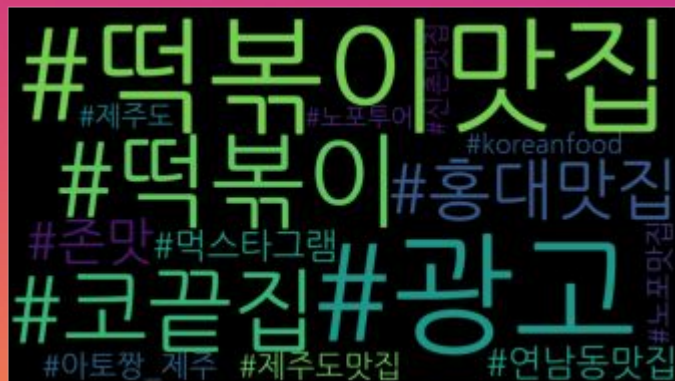
또한 #광고가 전체 400개의 게시글 중 60개로 약 15%의 광고게시글을 업로드했다.



데이터 시각화 Word Cloud



Ato의 content

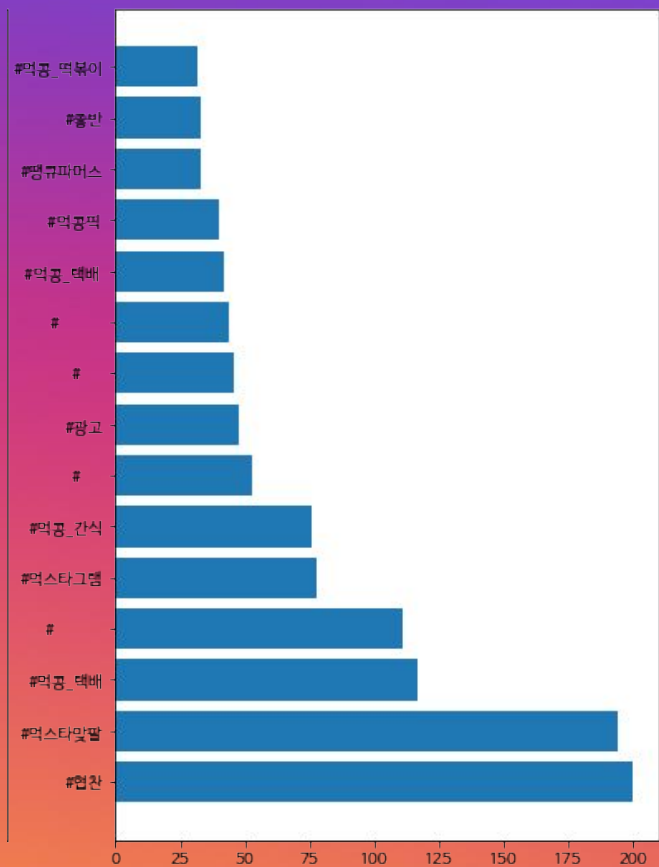


Ato의 tags

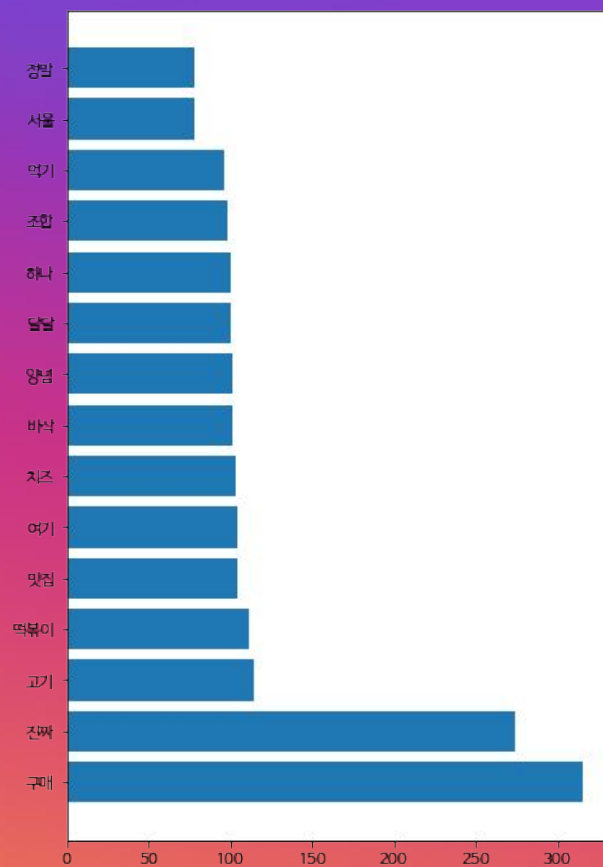
Bar Chart에서도 볼 수 있었듯이 광고의 비중이 높다는 것을 확인 할 수 있고 맛집, 주문, 위치의 빈도수도 높다는 것을 확인 할 수 있다.



데이터 시각화 Bar chart



먹콩이의 tags

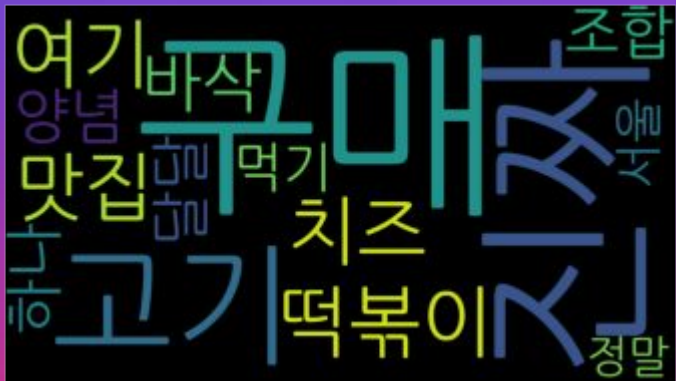


먹콩이의 content

Mukong의 계정의 특징은 구매, 진짜, 고기, 떡볶이 등의 단어를 자주 사용한다. 음식의 설명을 중점으로 게시글로 올리는 것을 확인 할 수 있다.



데이터 시각화 Word Cloud



먹콩이의 content



먹콩이의 tags

Bar Chart에서도 볼 수 있었듯이 음식 설명에 관한 단어들의 빈도수가 높게 나오는 것을 확인할 수 있다.

4

결론 및 제언



결론 :

1. 일반 게시글과 광고성 게시글의 비율을 확인.
2. 음식 인스타그램 게시글의 신뢰성에 의문. (광고, 협찬)
3. 작성자 성향에 따라 자주 쓰이는 단어가 다르다.

Thank you :)