

# 배달 어플리케이션 리뷰 감성 분석 및 답글 서비스

---

빅데이터경영통계전공 원동욱

# Index

---

01

서론

02

데이터 수집 및 전처리

03

단어 빈도

04

감성 분석

05

최종 모델링

# 서론

## 코로나 바이러스로 인한 국내 음식 배달 시장의 급격한 활성화

지난해부터 코로나 바이러스 감염증 사태에 따른 비대면 문화 확산으로  
지난 1년간 국내 음식 배달 시장이 급속도로 성장했다.

실제로 국내 음식 배달 거래액은 2019년 14조36억원에서  
2020년 20조1005억원으로 **43.5%** 뛰었다.

나 또한 1인 가구일 뿐만 아니라 거리두기로 인해 사람들과의 만남 및 외출이 줄어들면서  
대부분의 식사를 배달 음식으로 해결하고 있다.

## 배달 어플 이용 급증으로 인한 각 배달 음식 업체의 고충 파악

급격한 배달 어플 사용 증가와 텍스트데이터분석 강의 내용을 연관 지어 보았고,  
어플 사용 증가로 인한 리뷰 개수 증가로 각 업체의 리뷰 관리에 대한 고충이 예상 되었다.  
따라서 각 리뷰의 텍스트를 통해 감정을 분석하고 이를 기반으로 답글을 다는 서비스를 도입  
하고자 한다. 이를 통해 각 업체의 업무 효율성 증대에 기여하고자 한다.



# 데이터 수집

배달 어플리케이션은 웹 스크랩을 위해 웹페이지로도 이용 가능한 **요기요**로 선정하였다.

데이터 수집 대상 업체는 내가 직접 이용해 본 정릉 내 배달 업체 중

리뷰 이벤트를 진행하지 않으며, 사장님이 답글을 남기지 않는 **‘열혈분식 성북점’**으로 선정하였다.

열혈분식-성북점

메뉴 65	클린리뷰 1394	정보
-------	-----------	----

4.6  
★★★★★

맛 ★★★★★ 4.6  
양 ★★★★★ 4.7  
배달 ★★★★★ 4.7 ?

리뷰 1394개·사장님댓글 1개

사진리뷰만



cc\*\*님 어제

★★★★★ | 맛 ★ 5 양 ★ 5 배달 ★ 5

참치김밥/1,고구마튀김 ( 2P ) /1,제육덮밥/1

너무맛있어요 담에도먹을개요

신고



# 데이터 수집

열혈분식-성북점

메뉴 65	클린리뷰 1394	정보
-------	-----------	----

4.6  
★★★★☆

맛

★★★★☆ 4.6  

양

★★★★☆ 4.7  

배달

★★★★☆ 4.7 ?

리뷰 1394개·사장님댓글 1개

사진리뷰만 ☐

cc\*\*님 어제

신고

★★★★★ | 맛 ★ 5 양 ★ 5 배달 ★ 5

참치김밥/1,고구마튀김 ( 2P ) /1,제육덮밥/1

너무맛있어요 담에도먹을게요

리뷰 데이터는 요기요 리뷰 페이지 내 리뷰 내용, 리뷰 점수, 주문한 음식을 수집하였고  
리뷰 점수는 별점 스크랩이 어려워 맛, 양, 배달 점수 숫자의 평균으로 계산하였다.

# 데이터 수집

수집한 리뷰 데이터는 Positive Review와 Negative Review로 구분하였다.

**Positive Review** = 리뷰 점수 4점 이상, **Negative Review** = 리뷰 점수 3점 이하

감성		리뷰
0	1	너무맛있어요 담에도먹을게요
1	1	고기짬뽕은언제나맛있습니다
2	1	항상 맛있게 먹어요. 근데 오늘은 김밥 밥이 좀 아쉬웠어요.
3	1	너무낫있어요~~~~~
5	1	김밥이 쉬어서 왔어요ㅠ
...	...	...
1383	1	GOOD GOOD GOOD
1385	1	대패삼겹김밥 진짜맛있게 먹었어요ㅠㅠ 떡볶이 양도 많구 같이 넣어주신 캔커피도 감사합...
1387	1	맛있어요@@@@@@@@
1389	1	별 기대없이 시켰는데 괜찬네요
1390	1	맛있었고 생각보다 배달이 빨리 왔어요!

→ Positive Review DataFrame

# 데이터 수집

수집한 리뷰 데이터는 Positive Review와 Negative Review로 구분하였다.

**Positive Review** = 리뷰 점수 4점 이상, **Negative Review** = 리뷰 점수 3점 이하

감성		리뷰
4	0	쫄면 별로..특히 제육덮밥은 고기냄새나고ㅜ
51	0	비빔국수 김치군내가 아쉬웠어요. 드셔보세용 오므라이수 맛나요
152	0	김밥은 맛있는데 치즈김밥인데 참치김밥으로 주셨어요 열혈냉면/김치말이국수 점점 맛이 ...
231	0	김밥이 좀 말라있어서 아쉬웠는데 쫄면은 아주 맛있었어요!!
262	0	많이 시켜먹어서 기대치가 있었는데 고기짬뽕 평 좋아서 시켰다가 너무 후회했습니다; ...
...	...	...
1347	0	국수맛은 그냥 시원한것빠곤 모르겠고 김치찌국은 고추들어가서 약간 매콤하고 맛있네요!
1352	0	김밥도 괜찮고떡볶이도 배달음식치고 맛있더라고요 근데먹다가 떡에이상한 반점있는거보고....
1384	0	아 기분나빠서 못먹겠네요 동생이 시킨 제육덮밥은 달고 그냥 달다 정도 아니고 못먹을...
1386	0	맛과 양은 괜찮은데 배달시간이 너무 아쉽네요. 65분 걸린다고 해서 기다렸는데 80...
1388	0	포장이나 배달등이 좀 잘 됬음 좋을것 같아염!!!

→ Negative Review DataFrame

# 데이터 전처리

## Kiwi를 통한 형태소 분석

형태소 분석을 통해 감성 분석에 불필요한 형태소를 제외하고 체언과 용언만 남김

## 결측값 처리

## 내용이 없는 리뷰는 삭제 처리함

```
In [14]: positive_df = positive_df.loc[(positive_df['리뷰'] != ' ') & (positive_df['리뷰'] != ' ') &
                                         (positive_df['리뷰'] != ' ')].reset_index(drop = True)
```



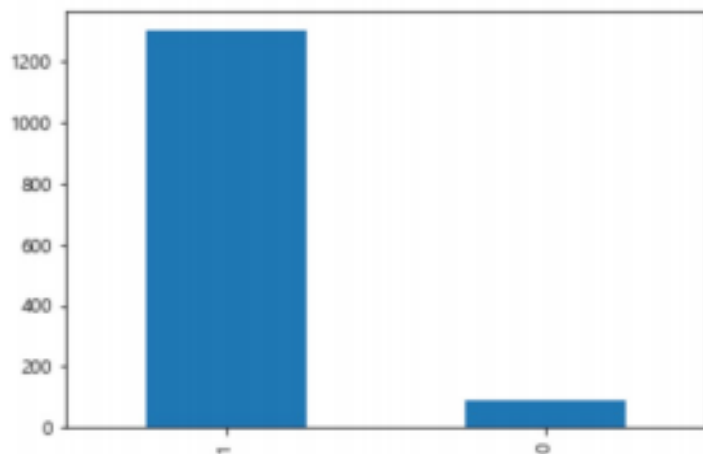
# 데이터 전처리

Positive Review 1300개, Negative Review 가 90개로 밸런스가 맞지 않는 문제 발생  
정상적인 지도학습을 위해 오버 샘플링을 통해 데이터 균형을 맞춤

```
yogiyo = pd.read_csv('열혈분식.csv')
```

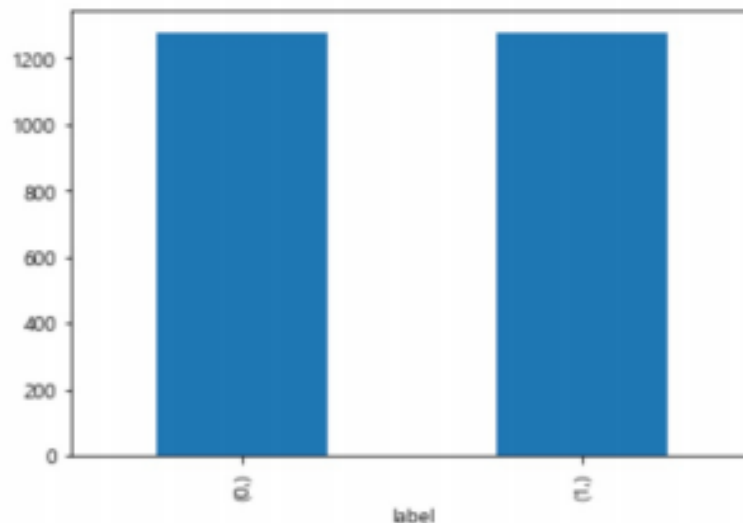
```
yogiyo['label'].value_counts().plot.bar()
```

<AxesSubplot:>



```
: pd.DataFrame(y_over).value_counts().plot.bar()
```

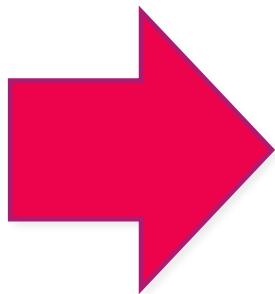
```
: <AxesSubplot:xlabel='label'>
```



# 단어 빈도

TDM과 단어 구름을 활용하여 Positive Review에서 빈도수가 높은 단어를 분석했다.

단어	빈도
맛있어요	232
맛있게	166
너무	107
항상	106
먹었습니다	96
양도	83
진짜	74
배달도	64
맛있네요	63



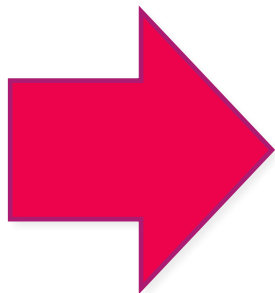
→ Positive Review Wordcloud

[맛있어요, 맛있게, 너무, 항상, 먹었습니다, 양도, 진짜, 배달도, 맛있네요] 등 긍정적 단어 파악

# 단어 빈도

TDM과 단어 구름을 활용하여 Negative Review에서 빈도수가 높은 단어를 분석했다.

단어	빈도
너무	13
그냥	10
맛은	8
쫄면은	6
배달은	6
양이	5
시켰는데	5
배달이	5
조금	5



→ Negative Review Wordcloud

[너무, 그냥, 맛은, 쫄면은, 배달은, 양이, 시켰는데, 배달이, 조금] 등 부정적 단어 파악

# 감성 분석

train\_test\_split, 로지스틱 회귀분석과 Early stopping 을 활용하여 감성 분석을 진행하였다.

```
In [186]: lb_review.label.values
```

```
Out[186]: array([1, 1, 1, ..., 0, 0, 0])
```

```
In [208]: x = dtm
```

```
In [209]: dtm.shape
```

```
Out[209]: (1301, 1000)
```

```
In [211]: y = lb_review.label.values[:1301]
```

```
In [212]: from sklearn.model_selection import train_test_split
```

```
In [214]: x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=1300)
```

```
In [108]: cv = CountVectorizer(max_features=2000, stop_words=stopwords_ko)
tdm = cv.fit_transform(review['review'])
X = tdm
Y = review['sentiment']
x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=42)
```

```
In [109]: import tensorflow as tf
model = tf.keras.models.Sequential()
model.add(
    tf.keras.layers.Dense(
        1,
        input_shape=(2000,),
        activation='sigmoid',
        kernel_regularizer=tf.keras.regularizers.l2(0.00001)))
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

```
In [110]: model.fit(x_train.toarray(), y_train.values, epochs=300, batch_size=100, validation_split=0.1,
                    callbacks=[tf.keras.callbacks.EarlyStopping()])
```

# 최종 모델링

작성자가 자유롭게 Review를 입력하면,  
이를 감성 분석해 Positive Review에는 자동으로 감사 인사를,  
Negative Review에는 자동으로 사과말을 답글로 남기는 모델을 구현하였다.

```
def review_auto() :  
  
    new_review = (input('리뷰를 남겨주세요: '))  
    new_sentence = list(extract_keywords(new_review))  
    new_sentence = [word for word in new_sentence if not word in stopwords['불용어']].tolist()  
    new_sentence = ' '.join(new_sentence)  
  
    vector = CountVectorizer(max_features=1135)  
    tdm = vector.fit_transform(total_df['corpus']).toarray()  
    array_value = vector.transform([new_sentence]).toarray()  
    x = model.predict(array_value)  
    print(x[0][0])  
  
    if x>0.7:  
  
        print('안녕하세요. 배달의 민주입니다. 맛있게 드셨다니 정말 다행입니다.) 다음에도 꼭 주문해주세요! '  
    else:  
        print('고객님 안녕하세요, 불편을 끼쳐드려 진심으로 죄송합니다. 앞으로 더욱 노력하는 배달의 민주 되겠습니다.'
```

```
review_auto()
```

리뷰를 남겨주세요: 김밥이 맛있고 사장님이 친절해요

0.92384684

안녕하세요. 배달의 민주입니다. 맛있게 드셨다니 정말 다행입니다.) 다음에도 꼭 주문해주세요!

```
review_auto()
```

리뷰를 남겨주세요: 맛이 별로예요 줄면이 다 불어서 왔어요

0.25309855

고객님 안녕하세요, 불편을 끼쳐드려 진심으로 죄송합니다. 앞으로 더욱 노력하는 배달의 민주 되겠습니다.

**Thank you  
for watching!**