Sentiment Classification

데이터사이언스학과
19013230 홍예지
yeji980603@naver.com

Contents

1 About dataset

2 EDA & preprocessing

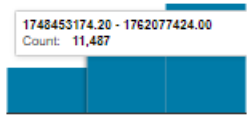
3 Modeling : DistilBert

4 Performance improvement

Dataset

Dataset

About dataset

tweet_id	sentiment	content
 1.69b 1.97b	neutral 22% worry 21% Other (22903) 57%	39827 unique values
1956967341	empty	@tiffanylue i know i was listenin to bad habit earlier and i started freakin at his part =[
1956967666	sadness	Layin n bed with a headache ughhhh...waitin on your call...
1956967696	sadness	Funeral ceremony...gloomy friday...
1956967789	enthusiasm	wants to hang out with friends SOON!
1956968416	neutral	@dannycastillo We want to trade with someone who has Houston tickets, but no one will.

✔ 트윗을 작성한 사람이 주석 형태로 작성한 감정 정보

- tweet_id
- sentiment : 13가지의 감정 범주
- content : 주석 형태로 작성된 텍스트

📒 데이터의 크기 : (40000, 3)

EDA & preprocessing

EDA & preprocessing

📂Remove unnecessary features

	0	1	2
0	tweet_id	sentiment	content
1	1956967341	empty	@tiffanylue i know i was listenin to bad habi...
2	1956967666	sadness	Layin n bed with a headache ughhhh...waitin o...
3	1956967696	sadness	Funeral ceremony...gloomy friday...
4	1956967789	enthusiasm	wants to hang out with friends SOON!
...	...	...	...
39996	1753918954	neutral	@JohnLloydTaylor
39997	1753919001	love	Happy Mothers Day All my love
39998	1753919005	love	Happy Mother's Day to all the mommies out ther...
39999	1753919043	happiness	@niariley WASSUP BEAUTIFUL!!! FOLLOW ME!! PEE...
40000	1753919049	love	@mopedronin bullet train from tokyo the gf ...

40001 rows × 3 columns

	1	2
1	empty	@tiffanylue i know i was listenin to bad habi...
2	sadness	Layin n bed with a headache ughhhh...waitin o...
3	sadness	Funeral ceremony...gloomy friday...
4	enthusiasm	wants to hang out with friends SOON!
5	neutral	@dannycastillo We want to trade with someone w...
...	...	...
39996	neutral	@JohnLloydTaylor
39997	love	Happy Mothers Day All my love
39998	love	Happy Mother's Day to all the mommies out ther...
39999	happiness	@niariley WASSUP BEAUTIFUL!!! FOLLOW ME!! PEE...
40000	love	@mopedronin bullet train from tokyo the gf ...

40000 rows × 2 columns

EDA & preprocessing

📂Change feature name

	1	2
1	empty	@tiffanylue i know i was listenin to bad habi...
2	sadness	Layin n bed with a headache ughhhh...waitin o...
3	sadness	Funeral ceremony...gloomy friday...
4	enthusiasm	wants to hang out with friends SOON!
5	neutral	@dannycastillo We want to trade with someone w...
...	...	...
39996	neutral	@JohnLloydTaylor
39997	love	Happy Mothers Day All my love
39998	love	Happy Mother's Day to all the mommies out ther...
39999	happiness	@niariley WASSUP BEAUTIFUL!!! FOLLOW ME!! PEE...
40000	love	@mopedronin bullet train from tokyo the gf ...

40000 rows × 2 columns

	sentiment	content
1	empty	@tiffanylue i know i was listenin to bad habi...
2	sadness	Layin n bed with a headache ughhhh...waitin o...
3	sadness	Funeral ceremony...gloomy friday...
4	enthusiasm	wants to hang out with friends SOON!
5	neutral	@dannycastillo We want to trade with someone w...
...	...	...
39996	neutral	@JohnLloydTaylor
39997	love	Happy Mothers Day All my love
39998	love	Happy Mother's Day to all the mommies out ther...
39999	happiness	@niariley WASSUP BEAUTIFUL!!! FOLLOW ME!! PEE...
40000	love	@mopedronin bullet train from tokyo the gf ...

40000 rows × 2 columns

EDA & preprocessing

Text preprocessing

사용자명 제거

HTML 엔티티 제거

@tiffanylue i know i was listenin to bad habit earlier and i started freakin at his part =[
Layin n bed with a headache ughhhh...waitin on your call...
Funeral ceremony...gloomy friday...
wants to hang out with friends SOON!
@dannycastillo We want to trade with someone who has Houston tickets, but no one will.
Re-pinging @ghostridah1: why didn't you go to prom? BC my bf didn't like my friends
I should be sleep, but im not! thinking about an old friend who I want. but he's married now. damn, & he wants me 2! scandalous!
Hmmm http://www.dlhero.com/ is down
@charviray Charlene my love. I miss you
@kelcouch I'm sorry at least it's Friday?

URL 제거

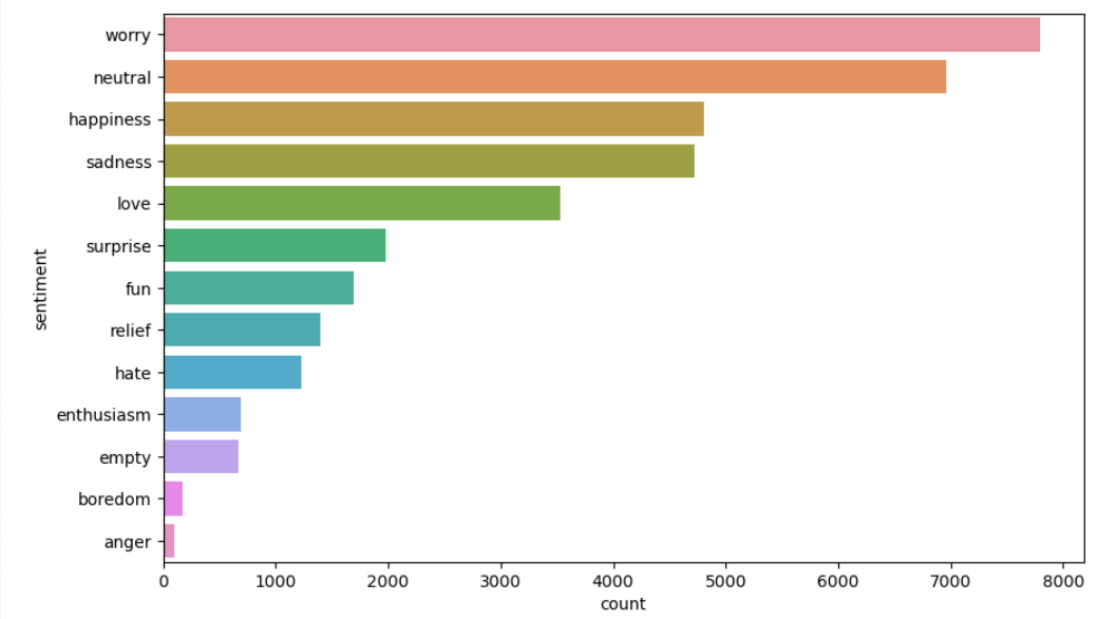
➕ 소문자 변환, 특수 문자 제거, 토큰화, 불용어 제거, 표제어 추출 등 진행

EDA & preprocessing

📂Check data structure

✔ 타입과 결측치 확인

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 35752 entries, 1 to 40000
Data columns (total 2 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   sentiment  35752 non-null  object
 1   content    35752 non-null  object
dtypes: object(2)
memory usage: 837.9+ KB
```



✔ 감정의 종류와 분포 확인

EDA & preprocessing

Encoding

	sentiment	content
1	empty	know listenin bad habit earlier started freaki...
2	sadness	layin n bed headache ughhhhwaitin call
3	sadness	funeral ceremonygloomy friday
4	enthusiasm	want hang friend soon
5	neutral	want trade someone houston ticket one
...	...	...
39995	happiness	succesfully following tayla
39997	love	happy mother day love
39998	love	happy mother day mommy woman man long youre mo...
39999	happiness	wassup beautiful follow peep new hit single ww...
40000	love	bullet train tokyo gf visiting japan since thu...

	sentiment	content
1	2	know listenin bad habit earlier started freaki...
2	10	layin n bed headache ughhhhwaitin call
3	10	funeral ceremonygloomy friday
4	3	want hang friend soon
5	8	want trade someone houston ticket one

Modeling

Modeling

📂Loading the pre-trained DistilBert model

```
import torch
import transformers as ppb
import warnings
warnings.filterwarnings('ignore')

model_class, tokenizer_class, pretrained_weights = (ppb.DistilBertModel, ppb.DistilBertTokenizer, 'distilbert-base-uncased')

tokenizer = tokenizer_class.from_pretrained(pretrained_weights)
model = model_class.from_pretrained(pretrained_weights)
```

📂Stratified sampling

```
from sklearn.model_selection import train_test_split

x = df["content"]
y = df["sentiment"]

x_train, _, y_train, _ = train_test_split(x, y, train_size=10000, stratify=y, random_state=42)

df_sampled = pd.DataFrame({'content':x_train, 'sentiment':y_train})
```

❓ 데이터가 40000개로 너무 많아서 deep learning을 진행하는데 무한 로딩 & 메모리 초과로 멈춤 발생
✔️ Stratified sampling을 통해 10000개의 데이터만 추출하여 데이터 크기 축소

Modeling

📂Check df_sample

✔ (10000,2)로 축소된 df_sample

	content	sentiment
31383	omg found thnx	5
33253	oops im watching mom son sleeping ing	8
14926	kno im sad evry leavin horrible im supposed b ...	10
1860	ever come across something reminds alot one pe...	10
37840	also try friendly fire havent already heard gr...	5
...	...	...
18914	aww dude fair thought point thing	6
38589	enjoying mommy day	4
2121	word counting hand hurt	12
21935	face mask hehe	4
24321	may th starwarsday via	8

10000 rows × 2 columns

✔ 축소된 sentiment categories

```
df_sampled['sentiment'].value_counts()
```

```
12    2180
8     1948
5     1346
10    1322
7      987
11     552
4      474
9      390
6      344
3      193
2      188
1       48
0       28
Name: sentiment, dtype: int64
```

Modeling

📂Preparing data for DistilBert

✔ Tokenization

```
tokenized = df_sampled['content'].apply((lambda x: tokenizer.encode(x, add_special_tokens=True)))
tokenized
```

```
31383          [101, 18168, 2290, 2179, 16215, 26807, 102]
33253    [101, 1051, 11923, 10047, 3666, 3566, 2365, 57...
14926    [101, 14161, 2080, 10047, 6517, 23408, 2854, 1...
1860     [101, 2412, 2272, 2408, 2242, 15537, 2632, 414...
37840    [101, 2036, 3046, 5379, 2543, 4033, 2102, 2525...
```

✔ Padding

```
max_len = 0
for i in tokenized.values:
    if len(i) > max_len:
        max_len = len(i)

#가장 긴 문장의 len을 구하고, 짧은 문장들은 뒤에 0을 추가해서 padding 진행
padded = np.array([i + [0]*(max_len-len(i)) for i in tokenized.values])
padded
```

```
array([[  101, 18168,  2290, ...,     0,     0,     0],
       [  101,  1051, 11923, ...,     0,     0,     0],
       [  101, 14161,  2080, ...,     0,     0,     0],
       ...,
       [  101,  2773, 10320, ...,     0,     0,     0],
       [  101,  2227,  7308, ...,     0,     0,     0],
       [  101,  2089, 16215, ...,     0,     0,     0]])
```

✔ Masking

```
attention_mask = np.where(padded != 0, 1, 0)
attention_mask.shape
```

```
(10000, 38)
```

Modeling

📂Deep learning

✔ 전처리된 입력을 사용해 모델 실행

```python
import time

start_time = time.time()

with torch.no_grad():
    last_hidden_states = model(input_ids, attention_mask=attention_mask)

end_time = time.time()

execution_time = end_time - start_time
print("실행시간 : {:.2f}초".format(execution_time))
```

```
실행시간 : 516.25초
```

```
BaseModelOutput(last_hidden_state=tensor([[[-0.2119, -0.0429,  0.0134,  ...,  0.0016,  0.2919,  0.3347],
         [-0.2285, -0.0771,  0.2305,  ..., -0.2790,  0.4037,  0.6434],
         [-0.2066, -0.2901,  0.3833,  ..., -0.2589,  0.0803,  0.5664],
         ...,
         [-0.1814, -0.3061,  0.0914,  ...,  0.4309, -0.2672,  0.2827],
         [-0.1648, -0.1960,  0.0909,  ...,  0.3828, -0.1762,  0.3305],
         [-0.1868, -0.1554,  0.1009,  ...,  0.3504, -0.1224,  0.3267]],

        [[-0.2241,  0.2844,  0.1016,  ..., -0.1495,  0.3913,  0.4542],
         [-0.6740,  0.8077,  0.4433,  ...,  0.0297,  0.6494,  0.2652],
         [-0.2993,  0.4834,  0.5944,  ..., -0.2159,  0.2872,  0.3982],
         ...,
         [ 0.0412,  0.0990,  0.5274,  ..., -0.2102,  0.0435,  0.1350],
         [-0.0976,  0.2543,  0.4701,  ..., -0.1508,  0.0746,  0.1171],
         [ 0.1394,  0.2586,  0.4294,  ..., -0.1421, -0.0988, -0.0767]],

        [[-0.3373,  0.0586,  0.1206,  ..., -0.1060,  0.2444,  0.5299],
         [-0.4679,  0.2288,  0.7018,  ...,  0.0090,  0.5424,  0.1029],
         [-0.8925,  0.2072,  0.4522,  ..., -0.4207, -0.2298,  0.3141],
         ...,
         [-0.2086,  0.0719,  0.3592,  ..., -0.0648,  0.1394,  0.1804],
         [-0.0529,  0.0275,  0.2426,  ..., -0.0929, -0.0309,  0.1999],
         [-0.0697, -0.0651,  0.1502,  ..., -0.0025, -0.0025,  0.2161]],

        ...,
```

✔ 입력 시퀀스에 대한 모델 처리 결과 (last_hidden_states)

Modeling

LogisticRegression

✔ train_test_split으로 분리 후
LogisticRegression에 입력을 넣어 예측 진행

```
from sklearn.linear_model import LogisticRegression

lr = LogisticRegression(solver='liblinear')
lr.fit(x_train, y_train)
```

```
LogisticRegression
LogisticRegression(solver='liblinear')
```

Evaluating

```
from sklearn.metrics import accuracy_score, confusion_matrix, precision_score, recall_score, f1_score

#accuracy
accuracy = accuracy_score(y_test, pred)
#precision
precision = precision_score(y_test, pred, average='macro')
#recall
recall = recall_score(y_test, pred, average='macro')
#f1_score
f1 = f1_score(y_test, pred, average='macro')

print(f"Accuracy: {accuracy}")
print(f"Precision: {precision}")
print(f"Recall: {recall}")
print(f"F1 Score: {f1}")
```

```
Accuracy: 0.322
Precision: 0.1647537115056242
Recall: 0.15962835318492952
F1 Score: 0.15301426150447464
```

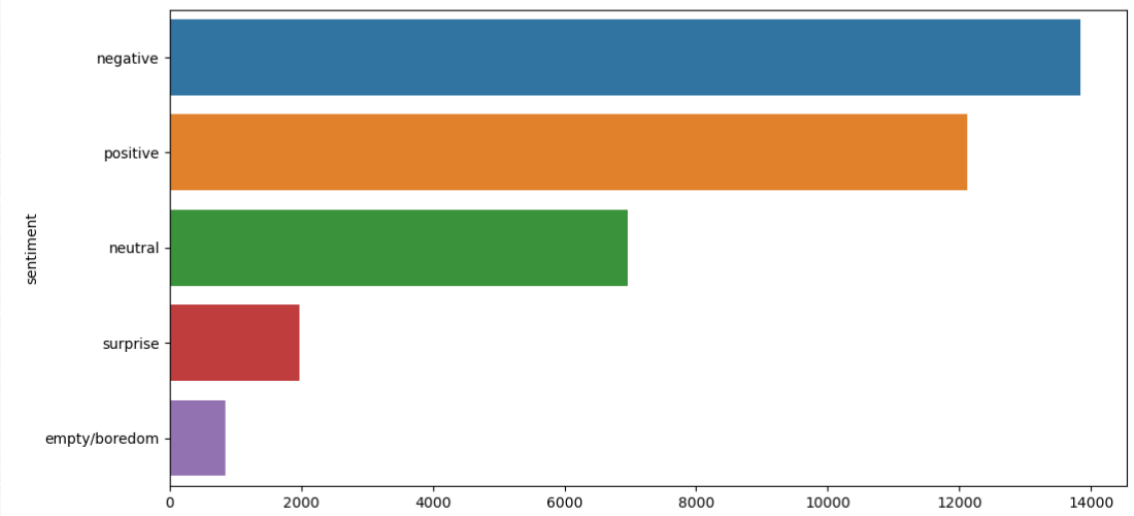
Performance Improvement

Performance improvement

📂 Reducing categories

```python
def categorization(sentiment):
    if sentiment in ['happiness', 'love', 'fun', 'relief', 'enthusiasm']:
        return 'positive'
    elif sentiment in ['worry', 'sadness', 'hate', 'anger']:
        return 'negative'
    elif sentiment == 'neutral':
        return 'neutral'
    elif sentiment in ['empty', 'boredom']:
        return 'empty/boredom' #두 감정 모두 감정적인 반응을 보이지 않음 (흥미가 없다거나 감정적으로 무감각)
    elif sentiment == 'surprise':
        return 'surprise' #surprise는 긍정이 될 수도, 부정이 될 수도 있으므로 단독
    else:
        return 'unknown'
```

✔ 감정의 종류와 분포 확인



Performance improvement

LogisticRegression

✔ train_test_split으로 분리 후
LogisticRegression에 입력을 넣어 예측 진행

```
from sklearn.linear_model import LogisticRegression

lr = LogisticRegression(solver='liblinear')
lr.fit(x_train, y_train)
```

```
▾ LogisticRegression
LogisticRegression(solver='liblinear')
```

```
pred = lr.predict(x_test)
pred
```

```
array(['positive', 'positive', 'neutral', ..., 'positive', 'positive',
       'negative'], dtype=object)
```

Evaluating

```
Accuracy: 0.322
Precision: 0.18475811505624 2
Recall: 0.15962835318492952
F1 Score: 0.15301426150447464
```

```
Accuracy: 0.5424
Precision: 0.23825139654445 81
Recall: 0.31795772639027414
F1 Score: 0.304735659483237
```

THANK YOU

데이터사이언스학과
19013230 홍예지
yeji980603@naver.com