pre_processing

April 22, 2023

1 Set-Up

```
[ ]: from google.colab import drive
     drive.mount('/content/drive')

     !pip install ko-ww-stopwords
     !pip install kr-sentence
     !python -m spacy download ko_core_news_md
     !pip install konlpy
```

```
Mounted at /content/drive
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Collecting ko-ww-stopwords
  Downloading ko_ww_stopwords-0.0.1-py3-none-any.whl (4.0 kB)
Installing collected packages: ko-ww-stopwords
Successfully installed ko-ww-stopwords-0.0.1
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Collecting kr-sentence
  Downloading kr_sentence-0.0.3-py3-none-any.whl (3.5 kB)
Installing collected packages: kr-sentence
Successfully installed kr-sentence-0.0.3
2023-04-17 14:11:37.160443: I tensorflow/core/util/port.cc:110] oneDNN custom
operations are on. You may see slightly different numerical results due to
floating-point round-off errors from different computation orders. To turn them
off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2023-04-17 14:11:37.218757: I tensorflow/core/platform/cpu_feature_guard.cc:182]
This TensorFlow binary is optimized to use available CPU instructions in
performance-critical operations.
To enable the following instructions: AVX2 AVX512F AVX512_VNNI FMA, in other
operations, rebuild TensorFlow with the appropriate compiler flags.
2023-04-17 14:11:38.251097: W
tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not
find TensorRT
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Collecting ko-core-news-md==3.5.0
```

```
  Downloading https://github.com/explosion/spacy-
models/releases/download/ko_core_news_md-3.5.0/ko_core_news_md-3.5.0-py3-none-
any.whl (69.0 MB)
                           69.0/69.0 MB
8.3 MB/s eta 0:00:00
Requirement already satisfied: spacy<3.6.0,>=3.5.0 in
/usr/local/lib/python3.9/dist-packages (from ko-core-news-md==3.5.0) (3.5.1)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in
/usr/local/lib/python3.9/dist-packages (from spacy<3.6.0,>=3.5.0->ko-core-news-
md==3.5.0) (3.0.8)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in
/usr/local/lib/python3.9/dist-packages (from spacy<3.6.0,>=3.5.0->ko-core-news-
md==3.5.0) (2.4.6)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in
/usr/local/lib/python3.9/dist-packages (from spacy<3.6.0,>=3.5.0->ko-core-news-
md==3.5.0) (1.0.4)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in
/usr/local/lib/python3.9/dist-packages (from spacy<3.6.0,>=3.5.0->ko-core-news-
md==3.5.0) (1.0.9)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.9/dist-
packages (from spacy<3.6.0,>=3.5.0->ko-core-news-md==3.5.0) (23.0)
Requirement already satisfied: smart-open<7.0.0,>=5.2.1 in
/usr/local/lib/python3.9/dist-packages (from spacy<3.6.0,>=3.5.0->ko-core-news-
md==3.5.0) (6.3.0)
Requirement already satisfied: pathy>=0.10.0 in /usr/local/lib/python3.9/dist-
packages (from spacy<3.6.0,>=3.5.0->ko-core-news-md==3.5.0) (0.10.1)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in
/usr/local/lib/python3.9/dist-packages (from spacy<3.6.0,>=3.5.0->ko-core-news-
md==3.5.0) (2.0.7)
Requirement already satisfied: typer<0.8.0,>=0.3.0 in
/usr/local/lib/python3.9/dist-packages (from spacy<3.6.0,>=3.5.0->ko-core-news-
md==3.5.0) (0.7.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.9/dist-
packages (from spacy<3.6.0,>=3.5.0->ko-core-news-md==3.5.0) (67.6.1)
Requirement already satisfied: langcodes<4.0.0,>=3.2.0 in
/usr/local/lib/python3.9/dist-packages (from spacy<3.6.0,>=3.5.0->ko-core-news-
md==3.5.0) (3.3.0)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in
/usr/local/lib/python3.9/dist-packages (from spacy<3.6.0,>=3.5.0->ko-core-news-
md==3.5.0) (1.1.1)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in
/usr/local/lib/python3.9/dist-packages (from spacy<3.6.0,>=3.5.0->ko-core-news-
md==3.5.0) (4.65.0)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in
/usr/local/lib/python3.9/dist-packages (from spacy<3.6.0,>=3.5.0->ko-core-news-
md==3.5.0) (2.27.1)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in
/usr/local/lib/python3.9/dist-packages (from spacy<3.6.0,>=3.5.0->ko-core-news-
md==3.5.0) (3.0.12)
Requirement already satisfied: numpy>=1.15.0 in /usr/local/lib/python3.9/dist-
packages (from spacy<3.6.0,>=3.5.0->ko-core-news-md==3.5.0) (1.22.4)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.9/dist-packages
(from spacy<3.6.0,>=3.5.0->ko-core-news-md==3.5.0) (3.1.2)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in
/usr/local/lib/python3.9/dist-packages (from spacy<3.6.0,>=3.5.0->ko-core-news-
md==3.5.0) (2.0.8)
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<1.11.0,>=1.7.4 in
/usr/local/lib/python3.9/dist-packages (from spacy<3.6.0,>=3.5.0->ko-core-news-
md==3.5.0) (1.10.7)
Requirement already satisfied: thinc<8.2.0,>=8.1.8 in
/usr/local/lib/python3.9/dist-packages (from spacy<3.6.0,>=3.5.0->ko-core-news-
md==3.5.0) (8.1.9)
Requirement already satisfied: typing-extensions>=4.2.0 in
/usr/local/lib/python3.9/dist-packages (from
pydantic!=1.8,!=1.8.1,<1.11.0,>=1.7.4->spacy<3.6.0,>=3.5.0->ko-core-news-
md==3.5.0) (4.5.0)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/usr/local/lib/python3.9/dist-packages (from
requests<3.0.0,>=2.13.0->spacy<3.6.0,>=3.5.0->ko-core-news-md==3.5.0) (1.26.15)
Requirement already satisfied: charset-normalizer~=2.0.0 in
/usr/local/lib/python3.9/dist-packages (from
requests<3.0.0,>=2.13.0->spacy<3.6.0,>=3.5.0->ko-core-news-md==3.5.0) (2.0.12)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.9/dist-packages (from
requests<3.0.0,>=2.13.0->spacy<3.6.0,>=3.5.0->ko-core-news-md==3.5.0)
(2022.12.7)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.9/dist-
packages (from requests<3.0.0,>=2.13.0->spacy<3.6.0,>=3.5.0->ko-core-news-
md==3.5.0) (3.4)
Requirement already satisfied: blis<0.8.0,>=0.7.8 in
/usr/local/lib/python3.9/dist-packages (from
thinc<8.2.0,>=8.1.8->spacy<3.6.0,>=3.5.0->ko-core-news-md==3.5.0) (0.7.9)
Requirement already satisfied: confection<1.0.0,>=0.0.1 in
/usr/local/lib/python3.9/dist-packages (from
thinc<8.2.0,>=8.1.8->spacy<3.6.0,>=3.5.0->ko-core-news-md==3.5.0) (0.0.4)
Requirement already satisfied: click<9.0.0,>=7.1.1 in
/usr/local/lib/python3.9/dist-packages (from
typer<0.8.0,>=0.3.0->spacy<3.6.0,>=3.5.0->ko-core-news-md==3.5.0) (8.1.3)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.9/dist-
packages (from jinja2->spacy<3.6.0,>=3.5.0->ko-core-news-md==3.5.0) (2.1.2)
Installing collected packages: ko-core-news-md
Successfully installed ko-core-news-md-3.5.0
  Download and installation successful
You can now load the package via spacy.load('ko_core_news_md')
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
```

```
Collecting konlpy
  Downloading konlpy-0.6.0-py2.py3-none-any.whl (19.4 MB)
                          19.4/19.4 MB
64.6 MB/s eta 0:00:00
Requirement already satisfied: lxml>=4.1.0 in
/usr/local/lib/python3.9/dist-packages (from konlpy) (4.9.2)
Requirement already satisfied: numpy>=1.6 in /usr/local/lib/python3.9/dist-
packages (from konlpy) (1.22.4)
Collecting JPype1>=0.7.0
  Downloading
JPype1-1.4.1-cp39-cp39-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (465 kB)
                          465.3/465.3 kB
49.0 MB/s eta 0:00:00
Requirement already satisfied: packaging in /usr/local/lib/python3.9/dist-
packages (from JPype1>=0.7.0->konlpy) (23.0)
Installing collected packages: JPype1, konlpy
Successfully installed JPype1-1.4.1 konlpy-0.6.0
```

[ ]:
```python
from ko_ww_stopwords.stop_words import ko_ww_stop_words
from ko_ww_stopwords.tools import is_stop_word, strip_outer_punct

print(ko_ww_stop_words)
```

```
{'  ', '  ', ' ', '  ', '   ', ' ', ' ', '  ', '   ', '   ', ' ', '  ', '  ',
'  ', '  ', ' ', ' ', '   ', '   ', '   ', ' ', ' ', ' ', '  ', '  ', ' ', '  ',
' ', '  ', '  ', '   ', '  ', ' ', '   ', '   ', ' ', '  ', '  ', '   ', '  ',
'  ', '  ', ' ', '   ', '  ', ' ', '   ', ' ', '  ', '  ', '  ', '  ', ' ',
'  ', '   ', '  ', '   ', '  ', ' ', '  ', '  ', '  ', '  ', '  ', '  ', '  ',
'   ', ' ', ' ', '  ', '  ', '  ', '   ', ' ', ' ', '   ', '   ', '  ', '  ',
' ', ' ', '  ', ' ', ' ', '  ', '  ', '  ', ' ', '  ', '  ', ' ', '  ',
'   ', '  ', '  ', '  ', ' ', '  ', ' ', '   ', '    ', ' ', '  ', '  ', '  ',
'  ', ' ', '  ', ' ', '   ', ' ', '  ', '   ', '   ', '  ', '  ', '   ', '  ',
'   ', '   ', '  ', ' ', ' ', '  ', '  ', '   ', '  ', '    ', ' ', ' ', ' ',
' ', ' ', '  ', '  ', ' ', ' ', '  ', '   ', '  ', '  ', '   '}
```

#Data & Pre-processing

[ ]:
```python
import pandas as pd

data = pd.read_table('/content/drive/MyDrive/korean-food-data/kr3.tsv')
data.head()
```

[ ]:
```
   Rating                                             Review
0       1           .                     …
1       1          !                      …
2       1        ,                        …
3       1                                 …
4       1                    Top5      .      …
```

[ ]:
```
data.shape
```

[ ]:
```
(641762, 2)
```

I had to manually perform data pre-processing on the korean text.

Since there are no “casings” in korean, I didn’t have to lower() everything.

I removed extra characters and white space.

I did removed stop words in the korean language using the konlpy (Ko-NLP-py) library to tokenize the korean text, remove the stop words, and put the tokens back into a string.

[ ]:
```
import spacy
from konlpy.tag import Okt
from nltk.stem import WordNetLemmatizer
import re

# create a spacy nlp object
nlp = spacy.load("ko_core_news_md")

# create a WordNetLemmatizer object
lemmatizer = WordNetLemmatizer()

# create a list of stop words
stop_words = set(ko_ww_stop_words)

# define a function to preprocess text
def preprocess_text(text):
    # convert text to lowercase

    # remove non-alphanumeric characters and extra whitespaces
    # [^a-zA-Z\s] doesnt apply to korean
    # Remove special characters
    text = re.sub(r'[^\w\s]', '', text)
    # Remove excess whitespace
    text = re.sub(r'\s+', ' ', text)
    # apply spacy nlp to tokenize and lemmatize the text
    doc = nlp(text)

    # tokenize korean sentence
    okt = Okt()
    tokens = okt.morphs(text, stem=True)

    # filter out stop words
    filtered_tokens = [token for token in tokens if token not in stop_words]

    # join the tokens back into a string
    processed_text = ' '.join(tokens)
```

```
         return processed_text
```

I remove data with ambiguous reviews, so we are only left with positive and negative reviews. This still leaves us with 459,000 samples.

```
[ ]: # remove data with ambiguous reviews
     data = data[data.Rating != 2]
```

```
[ ]: data.shape
```

```
[ ]: (459021, 2)
```

Next, I can see that there are only 70,000 negative reviews, and 388,000 positive reviews. In order to balance out the dataset, I pick 25,000 positive samples, 25,000 negative samples, merge them, and shuffle the data.

```
[ ]: equal_zero = data.loc[data['Rating'] == 0]
     equal_zero_trunc = equal_zero.iloc[:2500]
```

```
[ ]: equal_one = data.loc[data['Rating'] == 1]
     equal_one_trunc = equal_one.iloc[:2500]
```

```
[ ]: equal_zero_trunc.shape
```

```
[ ]: (2500, 2)
```

```
[ ]: frames = [equal_zero_trunc, equal_one_trunc]

     equal_unsorted = pd.concat(frames)
```

```
[ ]: equal_unsorted.shape
```

```
[ ]: (5000, 2)
```

```
[ ]: equal_sorted = equal_unsorted.sample(frac=1).reset_index(drop=True)
```

```
[ ]: # apply the preprocess_text function to the text column of the dataframe
     # took 15 minutes
     equal_sorted['Review'] = equal_sorted['Review'].apply(preprocess_text)
```

```
[ ]: equal_sorted.to_csv('/content/drive/MyDrive/korean-food-data/
       ↪equal-pre-processed_kr3_5k.csv', index=False)
```

Code below is the raw distribution, which has an overwhelming ratio of positive to negative reviews, which could skew the training

```
[ ]: # dataset = data.iloc[:50000]
```

[ ]:
```
# dataset.to_csv('/content/drive/MyDrive/korean-food-data/kr3_50k.csv',␣
↪index=False)
```

[ ]:
```
# df = pd.read_csv('/content/drive/MyDrive/korean-food-data/kr3_50k.csv')
```

[ ]:
```
# df.head()
```

[ ]:
```
# apply the preprocess_text function to the text column of the dataframe
# took 15 minutes
# df['Review'] = df['Review'].apply(preprocess_text)
```

References:

sentence tokenizer: https://github.com/Rairye/kr-sentence

spacy korean Korean language support: https://spacy.io/usage/models

spacy korean pretrained model: https://spacy.io/models/ko

korean regex: https://stackoverflow.com/questions/38156300/regex-how-do-you-match-korean-hangul-letters-in-javascript-es6

[ ]:
```
# df.to_csv('/content/drive/MyDrive/korean-food-data/pre-processed_kr3_50k.
↪csv', index=False)
```

[ ]:
```
df2 = pd.read_csv('/content/drive/MyDrive/korean-food-data/
↪pre-processed_kr3_50k.csv')
df2.head()
```

[ ]:
```
   Rating                                              Review
0       1           .                     …
1       1          !                      …
2       1        ,                        …
3       1                                 …
4       1                  Top5      .      …
```

[ ]:
```
X = df2.Review
y = df2.Rating
```

[ ]:
```
X.head()
```

[ ]:
```
0             .                     …
1            !                      …
2          ,                        …
3                                   …
4                    Top5      .      …
Name: Review, dtype: object
```

[ ]:
```
y[:10]
```

[ ]:
```
0    1
1    1
2    1
3    1
4    1
5    1
6    1
7    1
8    1
9    1
Name: Rating, dtype: int64
```

[ ]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,stratify=y,␣
 ↪test_size=0.2, train_size=0.8,random_state=1234)

X_train.shape
```

[ ]:
```
(40000,)
```

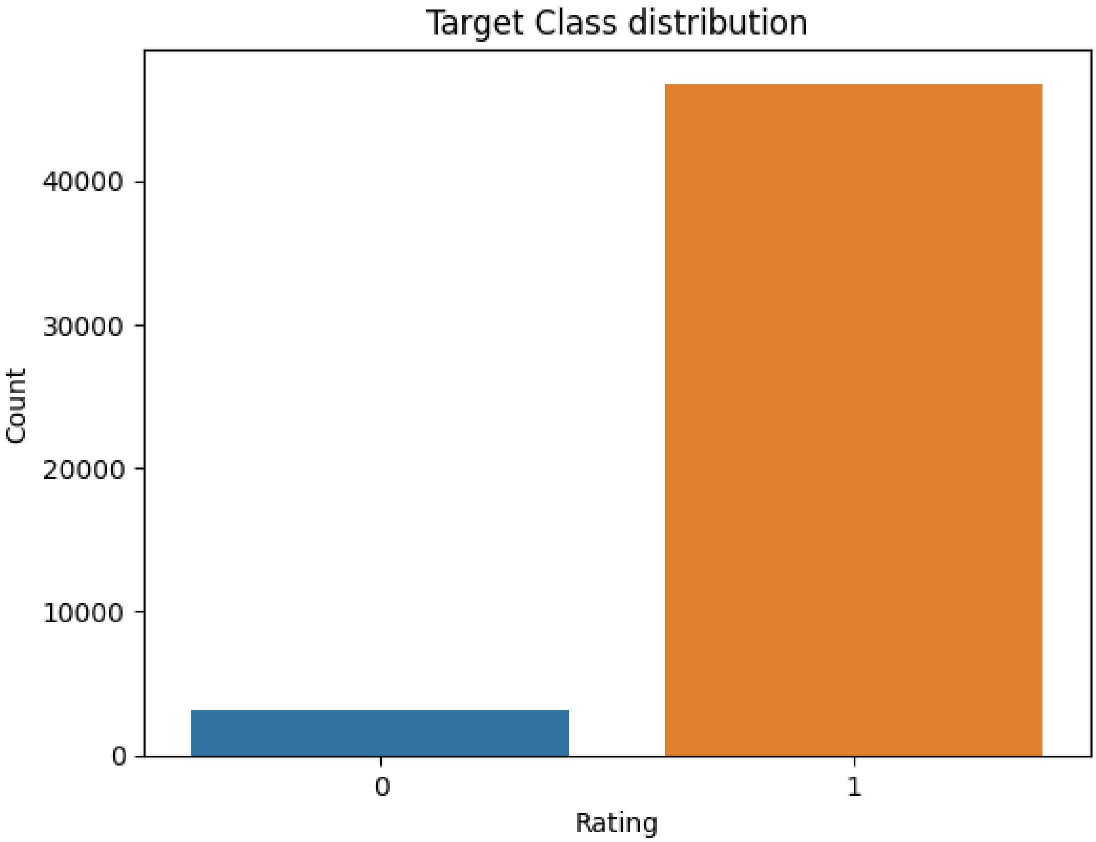
[ ]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# count the number of occurrences of each string in the DataFrame
counts = df2['Rating'].value_counts()

# create a bar plot of the counts using seaborn
sns.barplot(x=counts.index, y=counts.values)

# add a title and labels to the plot
plt.title('Target Class distribution')
plt.xlabel('Rating')
plt.ylabel('Count')
```

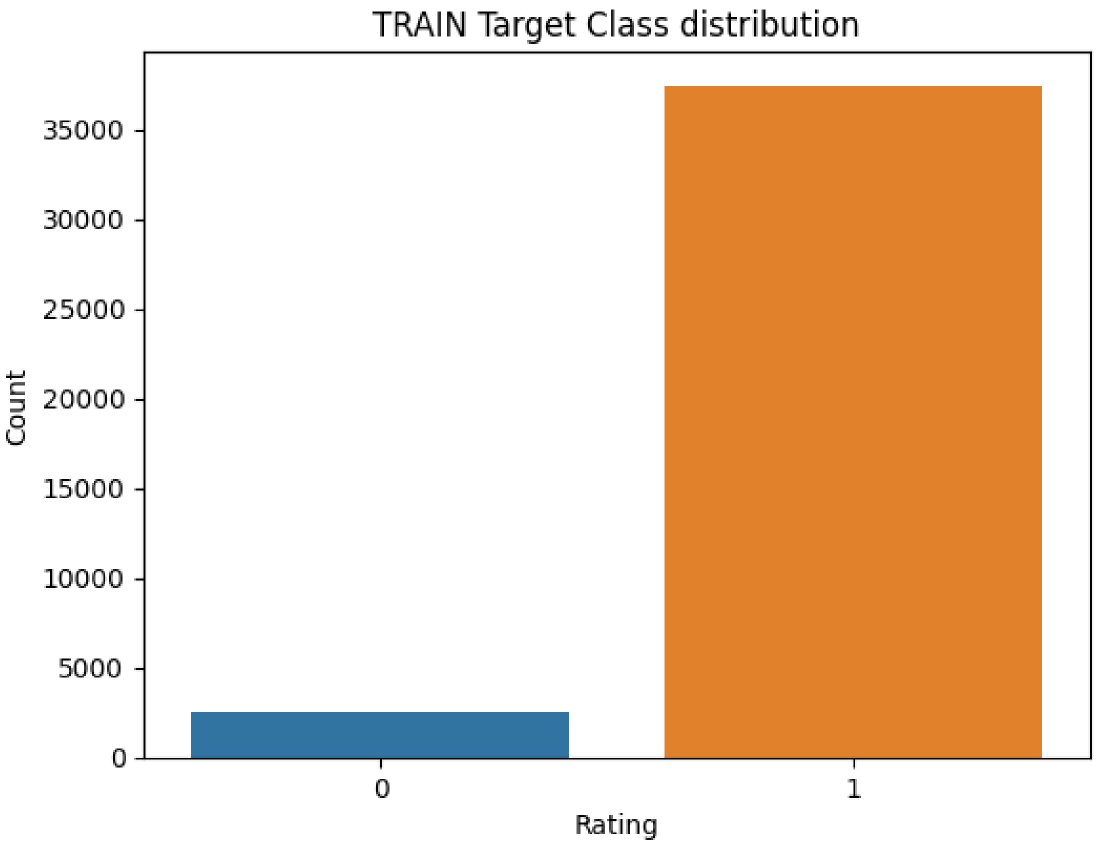
[ ]:
```
Text(0, 0.5, 'Count')
```



```
[ ]: train_counts = y_train.value_counts()
     # create a bar plot of the counts using seaborn
     sns.barplot(x=train_counts.index, y=train_counts.values)

     # add a title and labels to the plot
     plt.title('TRAIN Target Class distribution')
     plt.xlabel('Rating')
     plt.ylabel('Count')
```

```
[ ]: Text(0, 0.5, 'Count')
```



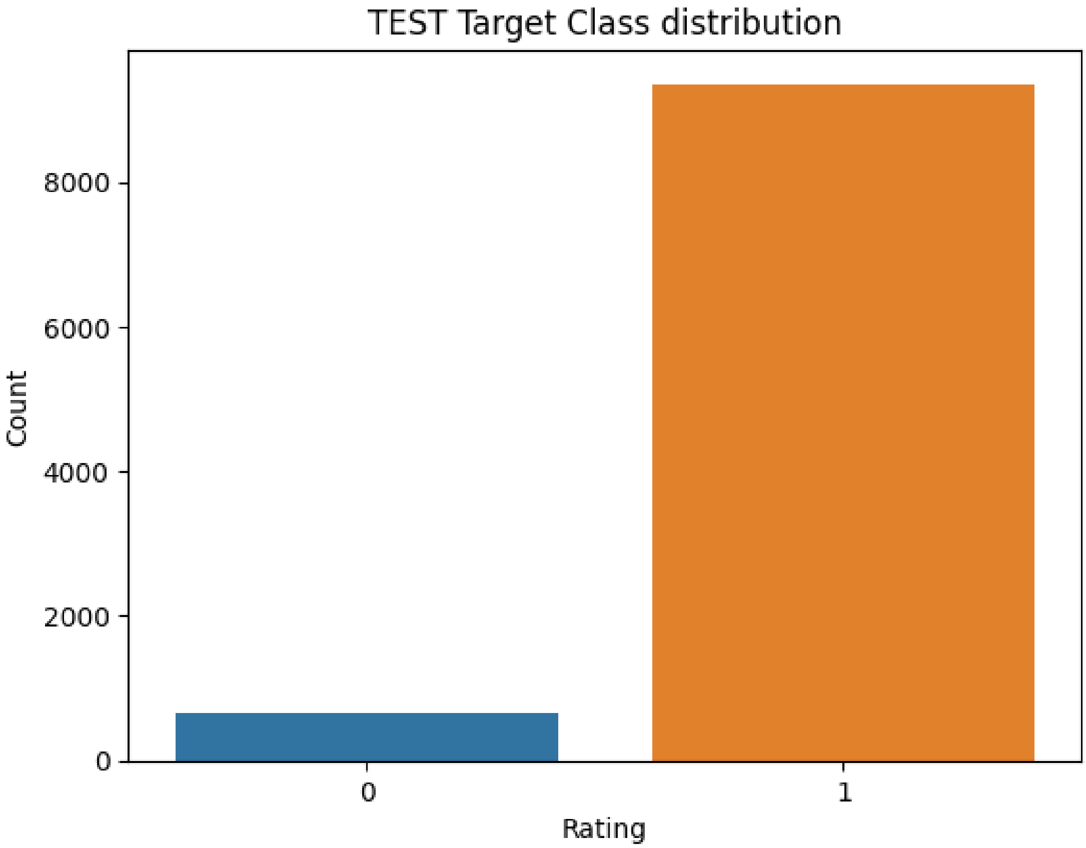
[ ]:

```
test_counts = y_test.value_counts()
# create a bar plot of the counts using seaborn
sns.barplot(x=test_counts.index, y=test_counts.values)

# add a title and labels to the plot
plt.title('TEST Target Class distribution')
plt.xlabel('Rating')
plt.ylabel('Count')
```

[ ]:

```
Text(0, 0.5, 'Count')
```



1.1 Describe the data set and what the model should be able to predict:

The dataset has 2 columns, a column containing restaurant reviews in Korean and a column containing the corresponding Rating of the restaurant (0 = negative, 1 = positive). Once trained, a model should be able to take in a review of a restaurant in korean, and predict weather the reviewer liked or disliked the restaurant.