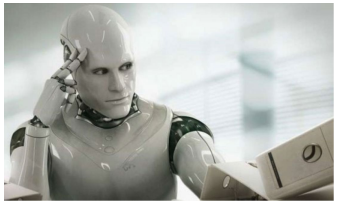
TensorFlow를 활용한 콘텐츠 분석

김태훈

BUZZ BEATERZ

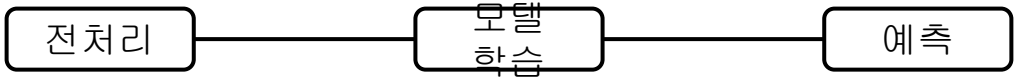
개요

데일리하면서 과하지않은 애교살 메이크업💜

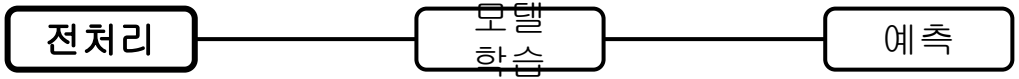


해당 회원이 콘텐츠를 좋아할 확률은 **85.9%** 입니다.

과정



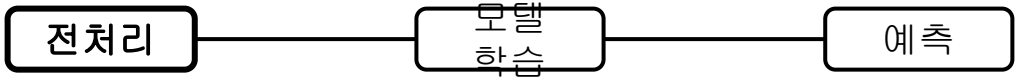
과정



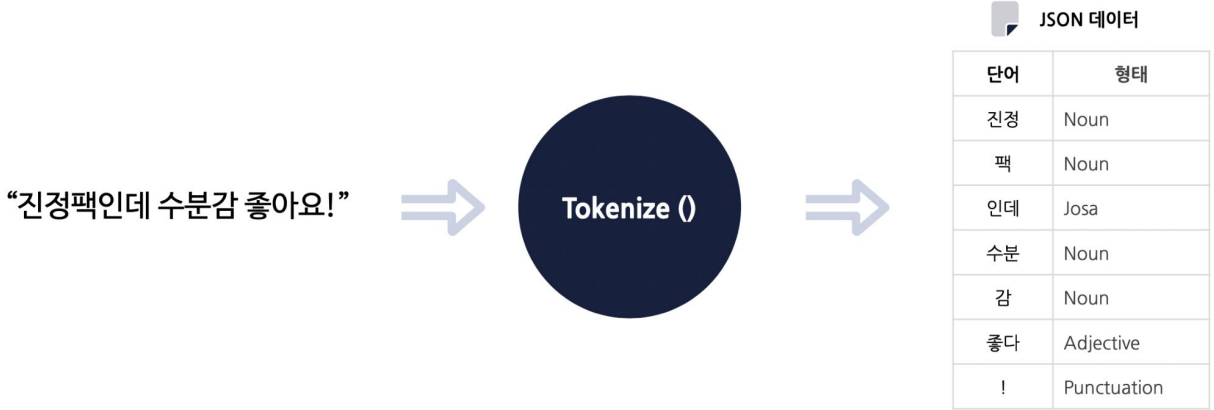
DB에서 회원이 좋아요 누른 콘텐츠 데이터 가져오기

id	la	content
00071	0	요거에요! 톤28 세정젤이네용
00073	1	진정팩인데 수분감 좋아요! 거즈마스크인 것두 좋아요!
00074	1	롬앤 더스티로즈 강추여! 처음엔 붉고 웜하게 올라오는데 갈수록 살짝 안개낀 핑크느낌 들면서 붉은데 너무 웜하지 않은 느낌! 쿨톤인데 완전 잘쓰고 있어요!
00079	0	사실 저 키보드 삼,, 키보드 소리 넘 귀여워 소듕해 다들 내 키보드 소리 들어줘쓰면 조케써,,☆
00112	1	문샷틴트핏블러 발림성 완전 버터 같아요! 지속력이 재금 아쉽지만 겉보속촉 넘 부드럽게 발려서 계속 바르고 싶은 고론느낌 #문샷 #틴트핏블러 #틴트추천
00115	1	#클리오#킬커버워터프루프마스카라이거 진짜 찐탱이에요..벌써 다써감.. 올영 세일할 때 1+1으로 꼭 겟하세요!!!!
00119	0	화일못이에요 저

과정



모든 콘텐츠 형태소 분석



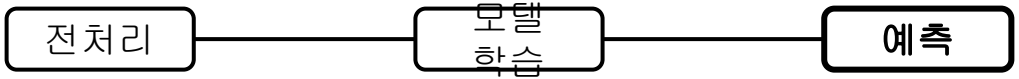
단어	형태
진정	Noun
팩	Noun
인데	Josa
수분	Noun
감	Noun
좋다	Adjective
!	Punctuation

과정

전처리가 완료된 데이터를
텐서플로우로 학습

모델
저장!

과정



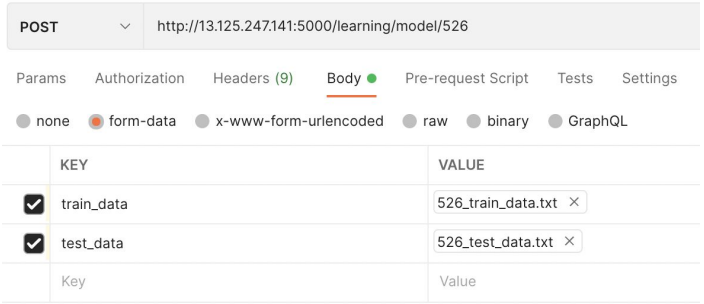
저장된 모델 불러와 결과 예측

데일리하면서과하지않은애교살 메이크업💜

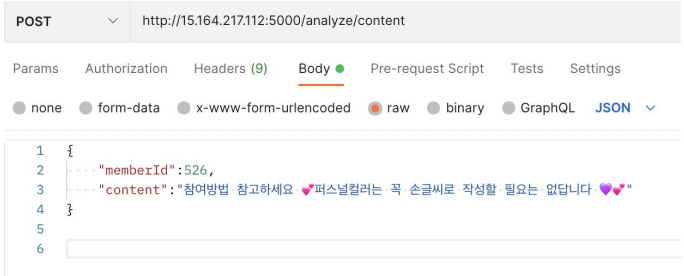
해당 회원이 콘텐츠를 좋아할 확률은 85.9% 입니다.

API 화

- 모델 학습 API



- 예측 API



개발환경 세팅, 실행, 구현 방법

김태훈

BUZZ BEATERZ

개발환경 세팅

os: mac Ventura 13.0 M1

Anaconda : 4.12.0

python : 3.9.0

라이브러리

nltk : 3.7
numpy : 1.22.3
konlpy : 0.6.0
tensorflow : 2.9.1
sklearn : 1.1.1
flask : 2.1.2

라이브러리 설정

```python
import json
import os
import time
from pathlib import Path

import nltk
import numpy as np
from konlpy.tag import Okt
okt = Okt()

from tensorflow.keras import models
from tensorflow.keras import layers
from tensorflow.keras import optimizers
from tensorflow.keras import losses
from keras.wrappers.scikit_learn import KerasClassifier
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score

from flask import Flask, request
from werkzeug.utils import secure_filename
app = Flask(__name__)
```

```
>>> import nltk
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ModuleNotFoundError: No module named 'nltk'
```

설치되지 않은 라이브러리는
(base)% conda activate tf (가상환경 실행)
(tf)% conda install nltk (라이브러리 설치)
명령어로 설치

라이브러리
설정

Linux or mac os 이슈!

라이브러리 설치하면서 설치가 안 되는 애들이 있는데
java jdk 가 필요한 것들이 있다.

근데 또 버전이 안 맞으면 설치가 안 된다.

```
openjdk version "15.0.7" 2022-04-19
OpenJDK Runtime Environment Zulu15.40+19-CA (build 15.0.7+4-MTS)
OpenJDK 64-Bit Server VM Zulu15.40+19-CA (build 15.0.7+4-MTS, mixed mode)
```

이 버전으로 하면 설치가 될 것이다.

설치 URL
https://www.azul.com/downloads/?version=java-15-mts&os=macos&architecture=x86-64-bit&package=jdk

본인 OS 환경에 맞는 jdk 설치

모델 학습 API

회원 id, 학습데이터, 테스트데이터를
받는다

```
def read_data(filename):
    with open(filename, 'r') as f:
        data = [line.split('  ') for line in f.read().splitlines()]
        data = data[1:]
    return data
```

```
@app.route("/learning/model/<int:memberId>", methods=['POST'])
def learning_model(memberId):

    if (not memberId):
        return {"error": "memberId is missing"}, 400

    file = request.files['train_data']
    filename = secure_filename(file.filename)
    file.save(filename)
    train_data = read_data(str(memberId) + '_train_data.txt')

    file = request.files['test_data']
    filename = secure_filename(file.filename)
    file.save(filename)
    test_data = read_data(str(memberId) + '_test_data.txt')
```

```
id    la  content
00071  0  요거에요! 톤28 세정젤이네용
00073  1  진정팩인데 수분감 좋아요! 거즈마스크인 것두 좋아요!
00074  1  롬앤 더스티로즈 강추여! 처음엔 붉고 웜하게 올라오는데 갈수록 살짝 안개낀 핑
00079  0  사실 저 키보드 삼,, 키보드 소리 넘 귀여워 소듕해 다들 내 키보드 소리 들어
00112  1  문샷틴트핏블러 발림성 완전 버터 같아요! 지속력이 재금 아쉽지만 겉보속촉 💕
00115  1  #클리오#킬커버워터프루프마스카라이거 진짜 찐탱이에요..벌써 다써감.. 올영 세
```

train_data.txt

```
print(train_data[0])
print(train_data[1])
```

```
['00071', '0', '요거에요! 톤28 세정젤이네용']
['00073', '1', '진정팩인데 수분감 좋아요! 거즈마스크인 것두 좋아요!']
```

모델 학습 API

모든 콘텐츠를 형태소 분석 하여 토큰화 후 저장 합니다.

```
def tokenize(doc):
    # norm은 정규화, stem은 근어로 표시하기를 나타냄
    return ['/'.join(t) for t in okt.pos(doc, norm=True, stem=True)]
```

```
if os.path.isfile(str(memberId) + '_rating_data.json'):
    with open(str(memberId) + '_rating_data.json') as f:
        train_docs = json.load(f)
    with open(str(memberId) + '_test_data.json') as f:
        test_docs = json.load(f)
else:
    train_docs = [(tokenize(row[2]), row[1]) for row in train_data]
    test_docs = [(tokenize(row[2]), row[1]) for row in test_data]
    # JSON 파일로 저장
    with open(str(memberId) + '_rating_data.json', 'w', encoding="utf-8") as make_file:
        json.dump(train_docs, make_file, ensure_ascii=False, indent="\t")
    with open(str(memberId) + '_test_data.json', 'w', encoding="utf-8") as make_file:
        json.dump(test_docs, make_file, ensure_ascii=False, indent="\t")

tokens = [t for d in train_docs for t in d[0]]
```

모델 학습 API

데이터
전처리

```
text = nltk.Text(tokens, name='NMSC')
selected_words = [f[0] for f in text.vocab().most_common(10000)]

def term_frequency(doc):
    return [doc.count(word) for word in selected_words]

train_x = [term_frequency(d) for d, _ in train_docs]
test_x = [term_frequency(d) for d, _ in test_docs]
train_y = [c for _, c in train_docs]
test_y = [c for _, c in test_docs]

x_train = np.asarray(train_x).astype('float32')
x_test = np.asarray(test_x).astype('float32')

y_train = np.asarray(train_y).astype('float32')
y_test = np.asarray(test_y).astype('float32')
```

가장 빈번한 단어 10000개만 사용

각 단어 별 등장 횟수를 리턴하는 함수

x에는 등장 횟수, y에는 단어

다차원 배열을 빠르게 처리하기 위한 마지막 전처리

모델 학습 API

모델 정의 및
생성

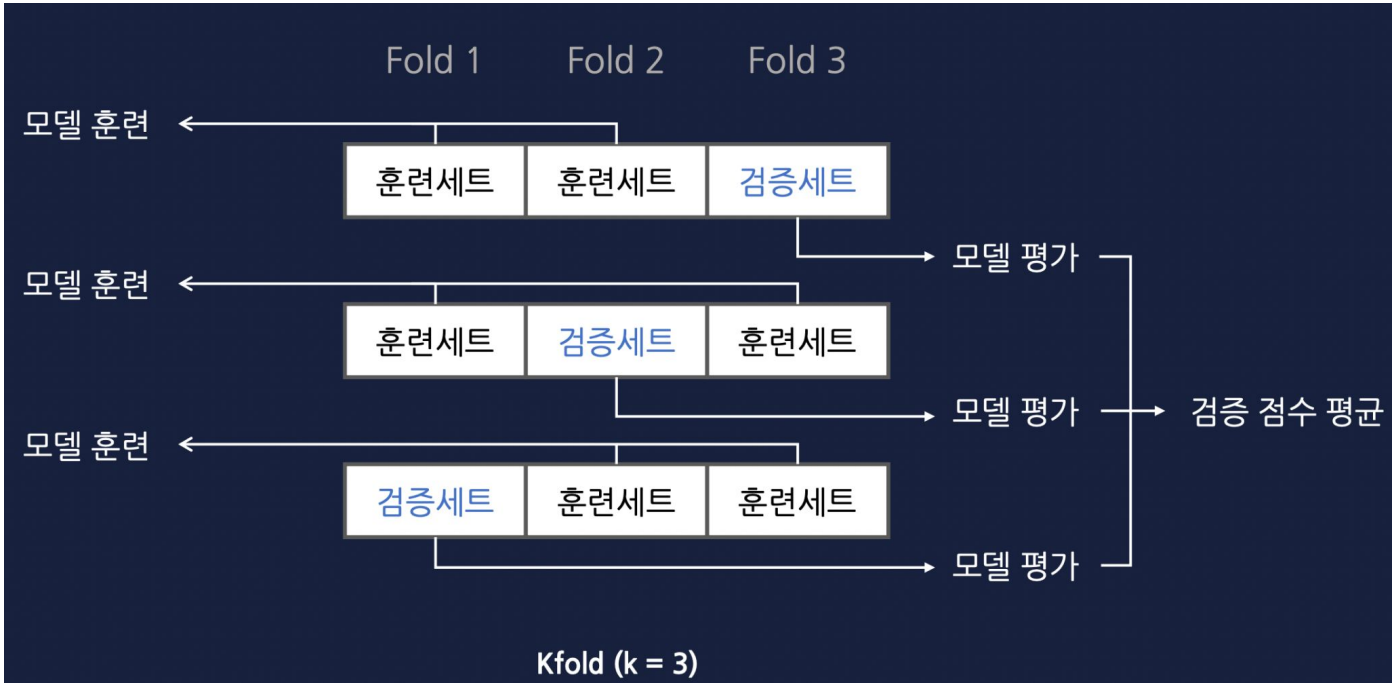
```
def create_model():
    model = models.Sequential()
    model.add(layers.Dense(64, activation='relu', input_shape=(10000,)))
    model.add(layers.Dense(64, activation='relu'))
    model.add(layers.Dense(1, activation='sigmoid'))
    model.compile(optimizer=optimizers.RMSprop(lr=0.001),
                  loss=losses.binary_crossentropy,
                  metrics=["acc"])
    return model
```

3개의 레이어로 구성된 모델 정의

모델에 추가적인 지표 설정
- optimizer: 값 보정 함수
- loss: 손실 함수
- metrics: 평가 지표 설정

모델 학습 API

K-Fold로 교차
검증



모델 학습 API

K-Fold로 교차
검증

```
seed = 7
np.random.seed(seed)

model = KerasClassifier(build_fn=create_model, epochs=10, batch_size=512, verbose = 0)
kfold = KFold(n_splits=10, shuffle=True, random_state=seed)
results = cross_val_score(model, x_train, y_train, cv=kfold)
```

```
model = create_model()
model.fit(x_train, y_train, epochs=10, batch_size=512, validation_data=(x_test, y_test))
model.save("saved_model/" + str(memberId) + "_model")

response = {
    "kfold_average": "{:.2f}".format(results.mean()),
}

return response, 200
```

콘텐츠 예측 API

회원 id, 콘텐츠를 받는다.

```python
@app.route("/analyze/content", methods=['POST'])
def analyze_content():
    params = request.get_json()

    memberId = params['memberId']
    if (not memberId):
        return {"error": "memberId is missing"}, 400

    content = params['content']
    if (not content):
        return {"error": "content is missing"}, 400
```

콘텐츠 예측 API

학습 모델을 생성하면서 전처리 된 json 데이터를 가져옵니다.

```
if os.path.isfile(str(memberId) + '_rating_data.json'):
    with open(str(memberId) + '_rating_data.json') as f:
        train_docs = json.load(f)
else:
    return {"error": "memberId: " + str(memberId) + " No learning model found"}, 400

tokens = [t for d in train_docs for t in d[0]]
text = nltk.Text(tokens, name='NMSC')
selected_words = [f[0] for f in text.vocab().most_common(10000)]
```

콘텐츠 예측

API

model.predict 함수로 콘텐츠 예측

```
def term_frequency(doc):
    return [doc.count(word) for word in selected_words]

def predict_pos_neg(model, content):
    token = tokenize(content)
    tf = term_frequency(token)
    data = np.expand_dims(np.asarray(tf).astype('float32'), axis=0)
    score = float(model.predict(data))
    return score
```

콘텐츠 예측 API

모델 불러오기, 콘텐츠 예측하기

```
model_dir = Path("saved_model/" + str(memberId) + "_model")
if not model_dir.is_dir():
    return {"error": "memberId: " + str(memberId) + " No learning model found"}, 400

model = models.load_model("saved_model/" + str(memberId) + "_model")
score = predict_pos_neg(model, content)

response = {
    "ai_feedback": round(score, 2)
}

return response, 200
```