

Django와 AWS를 이용한 카테고리 예측과 후기 긍정/부정 분석

01. Project Info



“

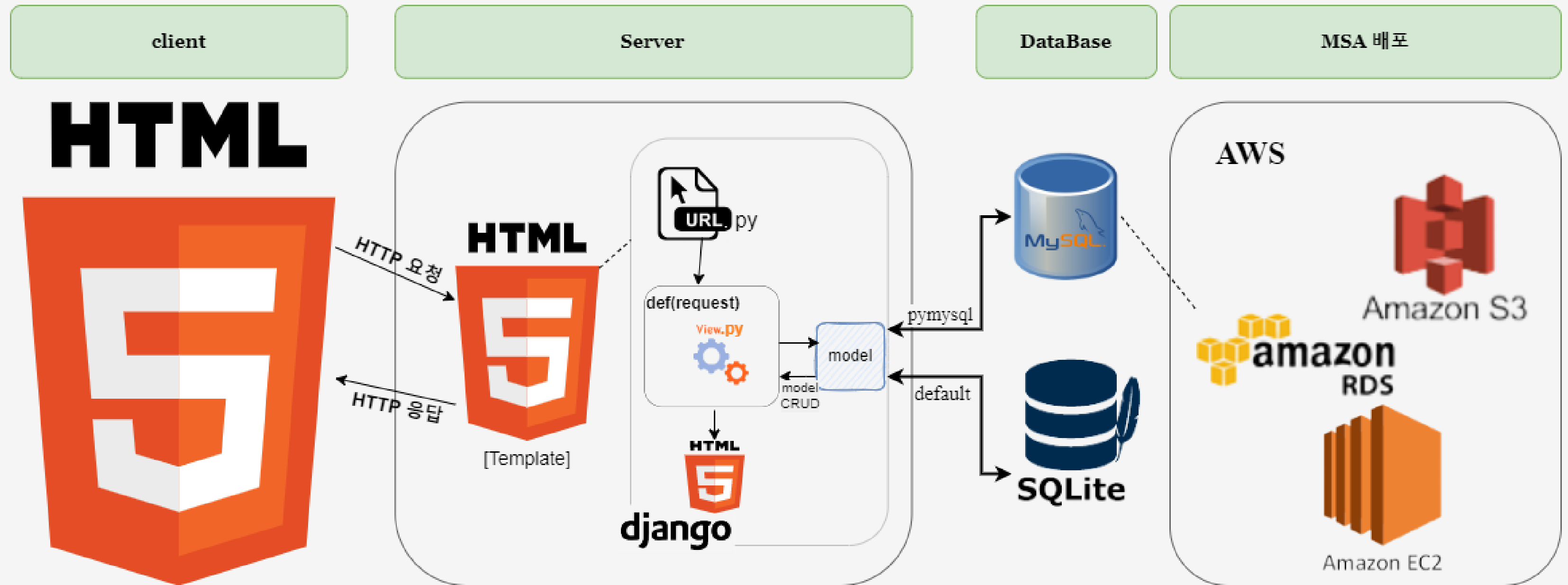
Decision Tree와 Joblib 라이브러리, SQLite를 사용하여 입력한 정보에 따른 카테고리 예측과 선형 회귀, TfidfVectorizer, wordcloud, pymysql을 사용하여 작성한 후기에 대한 긍정 여부를 분석하여 서버에 배포

”

- 1) 참여 인원: 3명
- 2) 개발 기간: 2022.05.26 ~ 2022.06.07
- 2) 담당 업무: django 서버 구축, DB 설계, 카테고리 예측 구현, 후기 긍정/부정 분석 구현
- 3) 개발 환경: Windows 10, Aws(Ubuntu)
- 4) 사용 도구: django, MySQL, Pycharm, Apache Tomcat, SQLite, DBeaver, draw.io 등
- 5) 사용 기술: Python, Spring, JSP, JavaScript, CSS, wordcoud, tfidfVectorizer, Decision Tree 등

+

02. 시스템 아키텍처



+

03. 서버 구축

콘솔 홈

RD

데이터베이스

그룹 리소스

↺

수정

작업 ▼

S3에서 복원

데이터베이스 생성

인스턴스

검색

데이터베이스을(를) 기준으로 필터링

< 1 > ⚙

<input type="checkbox"/>	Na	DB 식별자	역할 ▼	엔진 ▼	리전 및 AZ ▼	크기
<input type="checkbox"/>	ub	database-1	인스턴스	MySQL Community	ap-northeast-2b	db.t3
<input type="checkbox"/>	ub					
<input type="checkbox"/>	ai2	i-01a831601c7b4de4f	실행 중	t2.micro	2/2개 검사 통과	경보 없음

보안 그룹 (5) 정보

↺

작업 ▼

보안 그룹을 CSV로 내보내기 ▼

보안 그룹 생성

인바운드 규칙 (4)

↺

태그 관리

인바운드 규칙 편집

보안 그룹 필터링

< 1 > ⚙

<input type="checkbox"/>	Name ▼	보안 그룹 ID ▼	보안 그룹 이름 ▼	VPC
<input type="checkbox"/>	-	sg-0316fcd8fde06a58c	launch-wizard-1	vpc-
<input type="checkbox"/>	-	sg-05632c2f10124dee3	default	vpc-
<input type="checkbox"/>	-	sg-057910fae5fa53a3d	launch-wizard-3	vpc-
<input type="checkbox"/>	-	sg-0c819265d9da5cffa	launch-wizard-2	vpc-
<input type="checkbox"/>	ai2	sg-0c8c484516d7e11d7	launch-wizard-4	vpc-

보안 그룹 규칙 필터

< 1 > ⚙

<input type="checkbox"/>	보안 그룹 규칙 ID ▼	IP 버전 ▼	유형
<input type="checkbox"/>	sgr-0c4ad59086cf746ee	IPv4	HTTPS
<input type="checkbox"/>	sgr-0cb79b079d86edfa0	IPv4	모든 TCP
<input type="checkbox"/>	sgr-0cba2bf1964cc0dc8	IPv4	HTTP
<input type="checkbox"/>	sgr-04e1a327edbec8242	IPv4	SSH

+

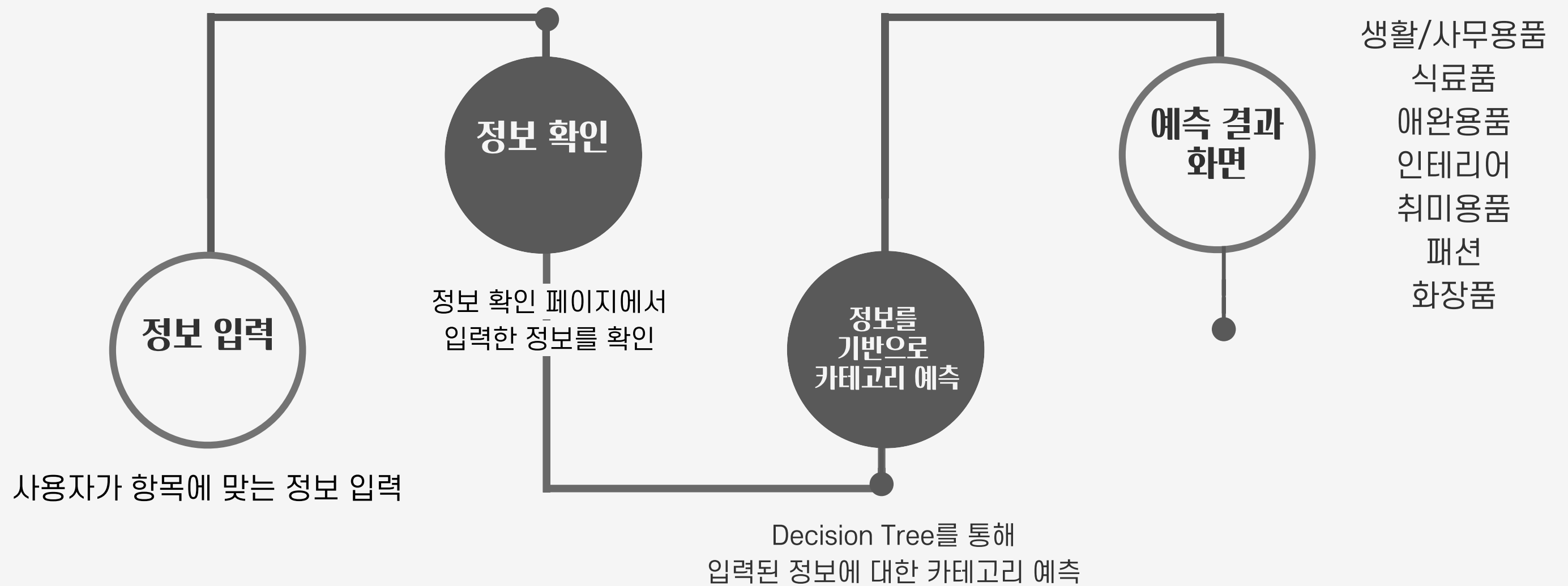
04. Urls.py

```
from django.contrib import admin
from django.urls import path

import ML.views

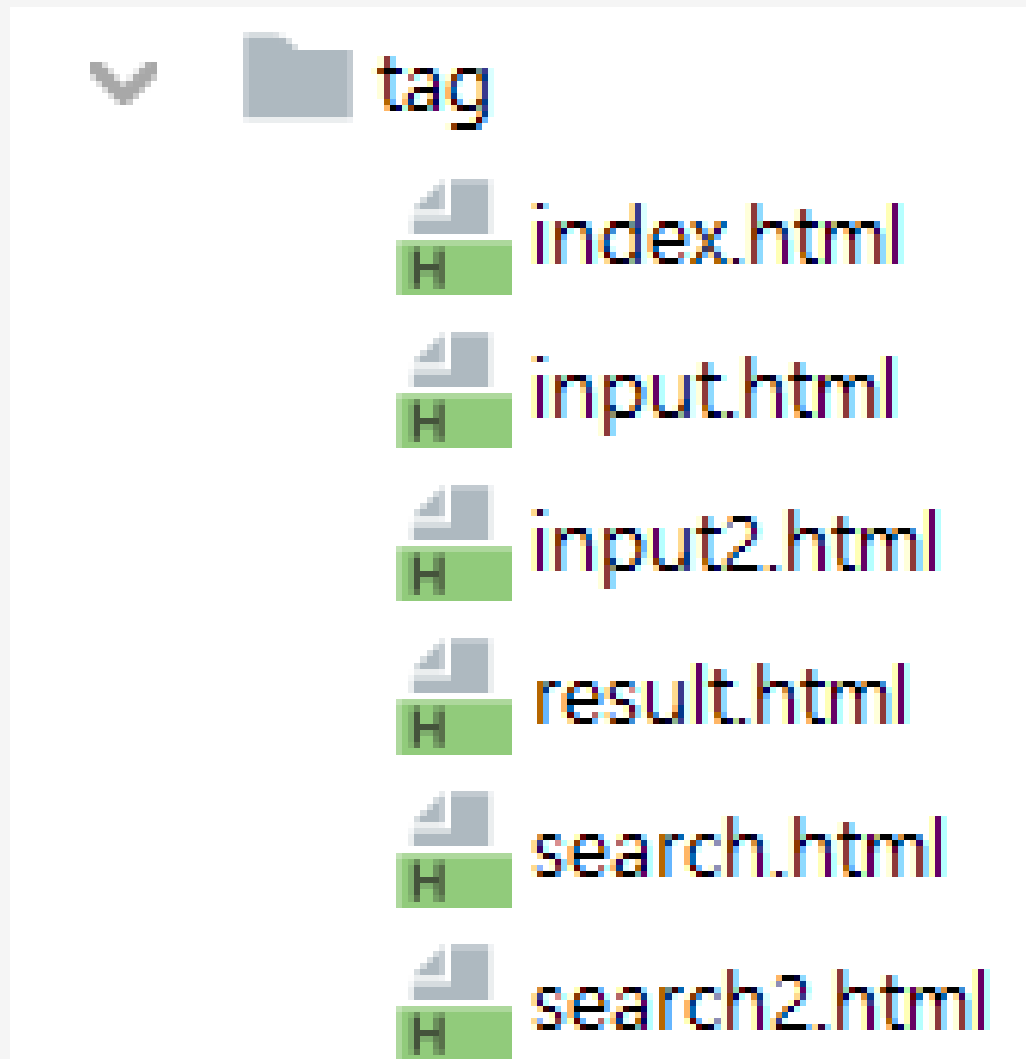
urlpatterns = [
    path('admin/', admin.site.urls),
    path('tag/', ML.views.index1),
    path('tag/input', ML.views.input),
    path('tag/input2', ML.views.input2),
    path('tag/search', ML.views.search),
    path('tag/search2', ML.views.search2),
    path('tag/output/<id>', ML.views.output),
    path('review/', ML.views.read_all),
    path('review/pn_review', ML.views.pn_review),
    path('review/wordcloud_pn', ML.views.pn_wordcloud),
    path('chart/', ML.views.index2),
    path('chart/chart1', ML.views.chart1),
    path('chart/chart2', ML.views.chart2)
]
```

05. 카테고리 예측 분석 과정



06. 카테고리 예측

tag.py



```
import warnings
warnings.filterwarnings(action='ignore')

import pandas as pd
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.tree import DecisionTreeClassifier
import joblib

def load_pkl(input_data):
    pred_module = joblib.load('/home/ubuntu/projects/ShoppingMallML/ai/tag_pred/tag_predict.pkl')
    result = pred_module.predict_proba([input_data])
    if result[0][0] == 1:
        tag = '생활/사무용품'
    elif result[0][1] == 1:
        tag = '식료품'
    elif result[0][2] == 1:
        tag = '애완용품'
    elif result[0][3] == 1:
        tag = '인테리어'
    elif result[0][4] == 1:
        tag = '취미용품'
    elif result[0][5] == 1:
        tag = '패션'
    elif result[0][6] == 1:
        tag = '화장품'
    return tag
```

06. 카테고리 예측

```
# import library
from django.shortcuts import render, redirect
from ML.models import Tag
from ai.review_pred.readall import readAll
from ai.review_pred.wordcloudneg import wordcloud_neg
from ai.review_pred.wordcloudpos import wordcloud_pos
from ai.tag_pred.tag_predict import load_pkl
from ai.review_pred import updatedb, pos_neg, readall

def index1(req):
    return render(req, 'tag/index.html')

def input(req):
    return render(req, 'tag/input.html')

def input2(req):
    data = req.POST
    print('입력한 정보 >> ', data)
    one = Tag(week1=data['week1'], week2=data['week2'], hour=data['hour'],
              gender=data['gender'], age=data['age'], size=data['size'],
              tag_click=data['tag_click'])
    one.save()
    return render(req, 'tag/input2.html')

def search(req):
    return render(req, 'tag/search.html')
```

```
def search2(req):
    data = req.POST
    idValue = data['id']
    one = Tag.objects.get(id=idValue)
    print('검색 결과 >> ', one)
    result = {'one': one}
    return render(req, 'tag/search2.html', context=result)

def output(req, id):
    one = Tag.objects.get(id=id)
    input_data = [one.week1, one.week2, one.hour,
                  one.gender, one.age, one.size,
                  one.tag_click]
    tag_pred = load_pkl(input_data)
    result = {'tag': tag_pred}
    return render(req, 'tag/result.html', context=result)
```

views.py

06. 카테고리 예측

```
<div id = "total">
  <p style="float:right">
    <button type="button" class="btn btn-warning" onClick="location.href='http://ec2-54-67-6-12.us-west-1.compute.am
    <button type="button" class="btn btn-warning" onClick="location.href='http://ec2-54-67-6-12.us-west-1.compute.am
  </p>
```

질문 정보 사이트

주의 요함 | .us-west-1.compute.amazonaws.com:9797/tag/

cafe ppt 사전 N D w3 수업 wiki git git-doc aws-applealicia

tag 예측페이지입니다

정보 작성 페이지에서 정보 작성 후
태그 예측 페이지로 이동해주세요

항목	입력
정보 작성	정보 작성 go!
정보 검색	정보 검색 go!

정보 입력

[예측 시작 페이지로](#)

	입력
평일/휴일	0
요일	3
시간대	0
성별	0
연령대	1
건수합계	5
태그 클릭량	100
정보 저장	

정보 작성 결과

정보 작성이 정상적으로 완료되었습니다!!

[태그 예측 페이지로..](#)

정보 작성 예시

평일/휴일	0(평일), 1(휴일)
요일	3(월요일), 6(화요일), 2(수요일), 1(목요일), 0(금요일), 5(토요일), 4(일요일)
시간대	0(02-06시), 1(07-10시), 2(11-14시), 3(15-18시), 4(19-22시), 5(23-02시)
성별	0(여성), 1(남성)
연령대	0(20대), 1(30대), 2(40대), 3(50대), 4(60대 이상)
건수합계	건수합계를 입력해주세요
태그 클릭량	태그 클릭량을 입력해주세요

+

06. 카테고리 예측

tag 예측페이지입니다

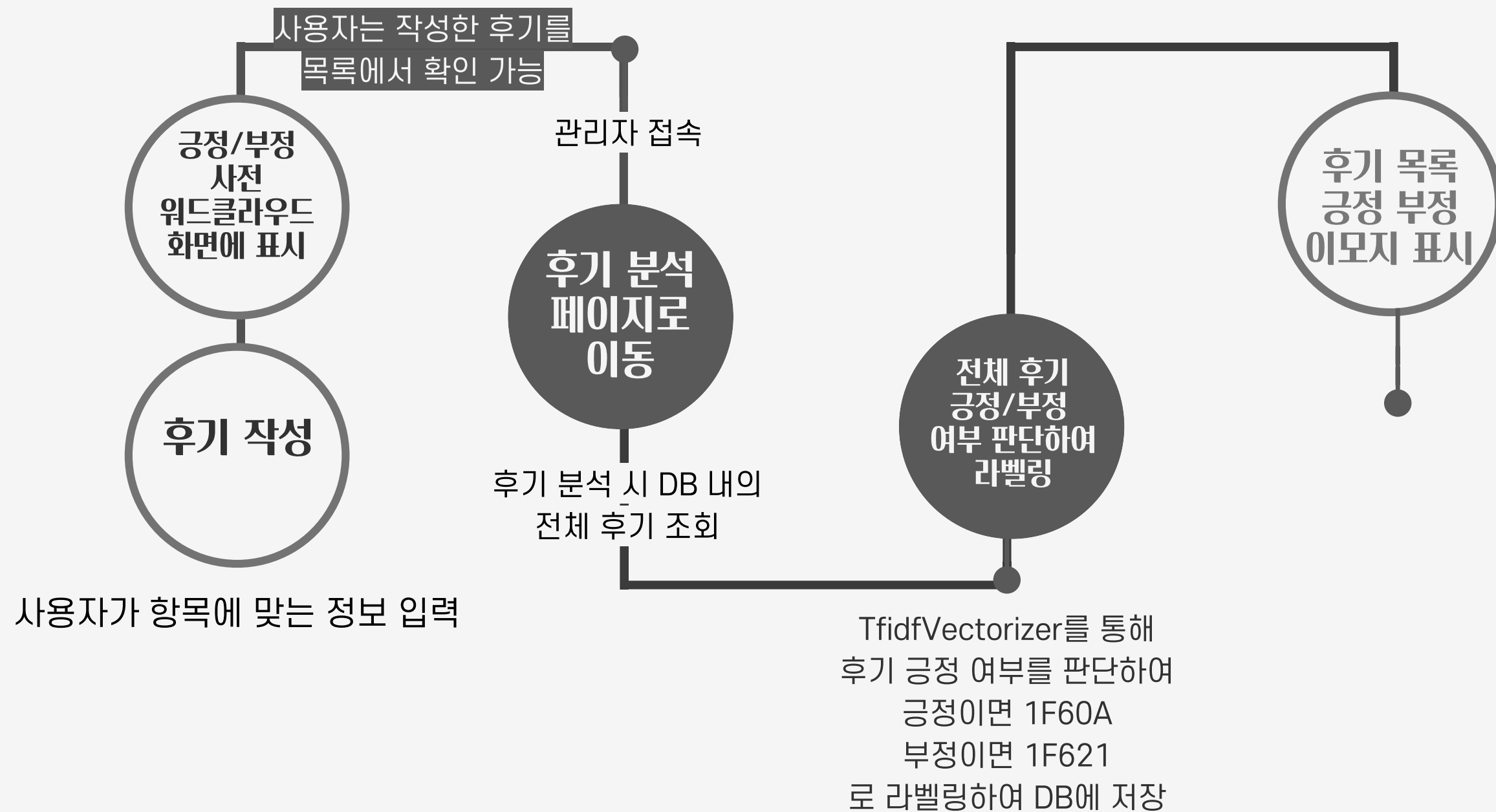
정보 작성 페이지에서 정보 작성 후
태그 예측 페이지로 이동해주세요

항목	입력
정보 작성	정보 작성 go!
정보 검색	정보 검색 go!

저장한 정보 검색 후 카테고리 예측을 누르면

예측 카테고리는 애완용품입니다.

07. 후기 긍정 여부 분석 과정



+

08. 부정 사전 워드 클라우드로 생성

```
from wordcloud import WordCloud
import matplotlib.pyplot as plt
from collections import Counter
import joblib

def wordcloud_neg():
    neg_dict = joblib.load('/home/ubuntu/projects/ShoppingMallML/ai/review_pred/neg_dict.pkl')
    words = [n for n in neg_dict if len(n) > 1]
    neg_count = Counter(words)
    wc = WordCloud(font_path='malgun', width=800, height=800,
                   scale=2.0, max_font_size=200,
                   background_color='white', colormap='summer')
    gen = wc.generate_from_frequencies(neg_count)
    plt.figure()
    plt.imshow(gen, interpolation="bilinear")
    plt.axis('off')
    wc.to_file('static/wordcloud/neg_wordcloud.png')
```

+

09. 긍정 사전 워드 클라우드로 표시

```
from wordcloud import WordCloud
import matplotlib.pyplot as plt
from collections import Counter
import joblib

def wordcloud_pos():
    pos_dict = joblib.load('/home/ubuntu/projects/ShoppingMallML/ai/review_pred/pos_dict.pkl')
    words = [n for n in pos_dict if len(n) > 1]
    pos_count = Counter(words)
    wc = WordCloud(font_path='malgun', width=800, height=800,
                   scale=2.0, max_font_size=200,
                   background_color='white', colormap='seismic')
    gen = wc.generate_from_frequencies(pos_count)
    plt.figure()
    plt.imshow(gen, interpolation="bilinear")
    plt.axis('off')
    wc.to_file('static/wordcloud/pos_wordcloud.png')
```

+

10. CORS 허용

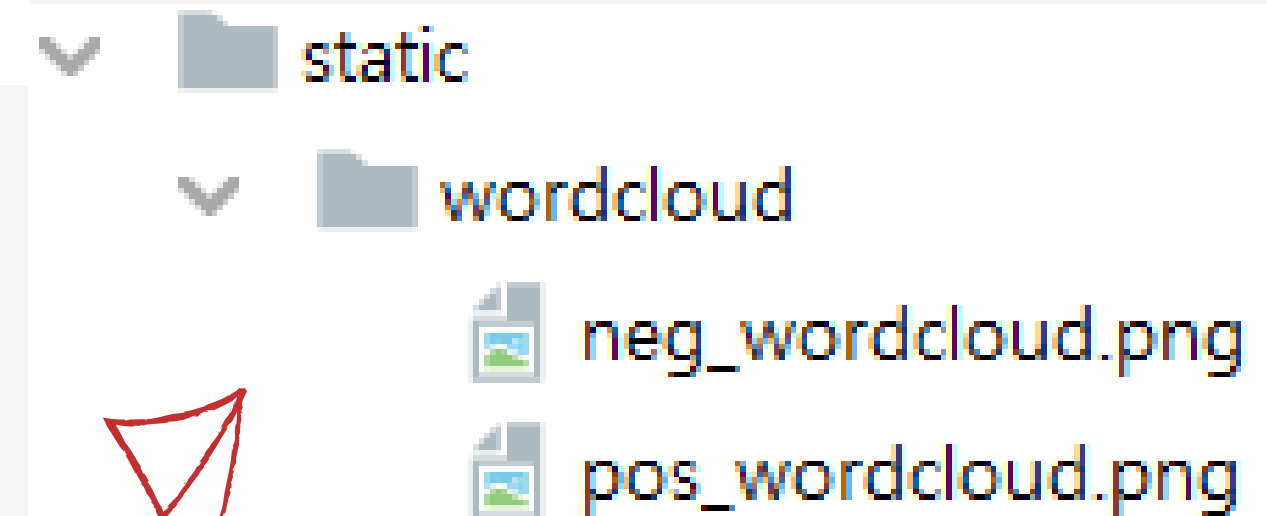
django 서버에서 java 서버로
워드클라우드 이미지를 보내기 위해
CORS Http 허용을 위해 설정

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'ML.apps.MLConfig',  
    'corsheaders',  
]  
  
MIDDLEWARE = [  
    'django.middleware.security.SecurityMiddleware',  
    'django.contrib.sessions.middleware.SessionMiddleware',  
    'django.middleware.common.CommonMiddleware',  
    'django.middleware.csrf.CsrfViewMiddleware',  
    'django.contrib.auth.middleware.AuthenticationMiddleware',  
    'django.contrib.messages.middleware.MessageMiddleware',  
    'django.middleware.clickjacking.XFrameOptionsMiddleware',  
    'corsheaders.middleware.CorsMiddleware',  
]  
  
CORS_ORIGIN_ALLOW_ALL = True
```

+

11. 긍정 부정 사전 워드 클라우드 표시

```
def pn_wordcloud(req):  
    wordcloud_pos()  
    wordcloud_neg()  
    return render(req, 'review/wordcloud_pn.html')
```



```
<div>  
    
```

11. 긍정 부정 사전 워드 클라우드 표시

```
$(function() {
  /* 긍정 사전, 부정 사전 워드클라우드 */
  $.ajax({
    url: 'http://ec2-54-67-6-12.us-west-1.compute.amazonaws.com:9797/review/wordcloud_pn',
    success: function(result) {
      alert(result)
      $('#div#pic').append(result)
    }
  })
})
```

로그아웃

회원관리

상품정보

좋아요

장바구니

고객센터

통계

총 평점:

3.5점



평점은 수정이 불가능하니 정확히 선택해주세요!!

<!-- 긍정 사전, 부정 사전에 관한 워드클라우드 보여주기 -->

<td>

<div id='pic' style="width:300px; height:300px;"></div>

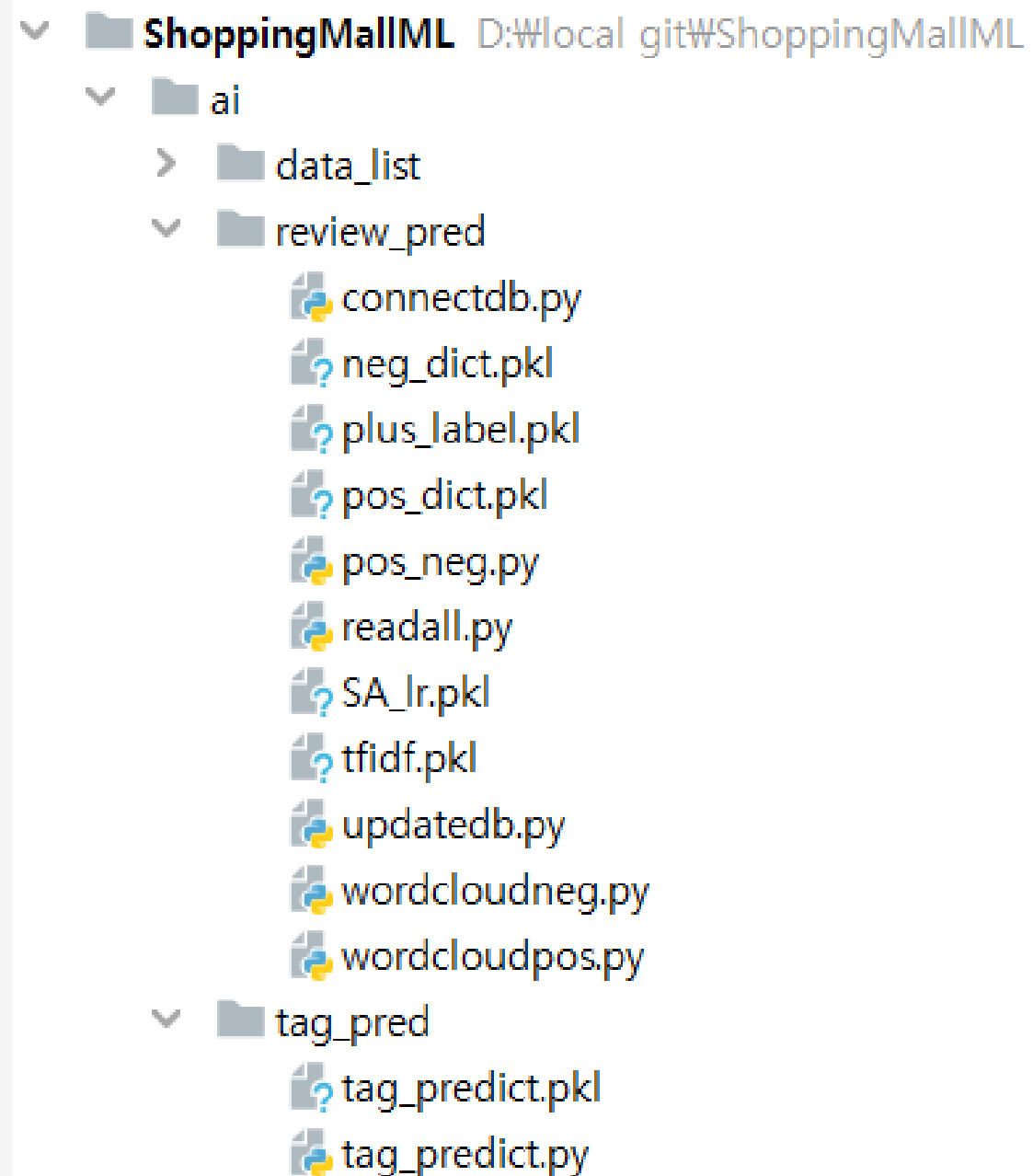
</td>

파일 선택 nistory.txt

후기 작성



12. 후기 긍정 부정 분석



connectdb.py

```

import pymysql

# 데이터 베이스 연결 및 데이터 불러오기
def db_read(review_idx):
    try:
        conn = pymysql.connect(
            host='database-1.cgind9azzrj.ap-northeast-2.rds.amazonaws.com',
            port=3306,
            user='root',
            password='db_password', # blind db_password
            db='shop2',
            charset='utf8'
        )
        # 연결된 통로(stream)을 통해, SQL문을 보내보자.
        # 2. 연결된 통로를 지정(접근할 수 있는) 커서 객체를 획득
        cur = conn.cursor()
        # 3. sql문을 보내보자
        sql = 'select * from review where review_idx = %s order by review_rgstdate limit 1'
        # 커서로 sql문을 보냄.
        result = cur.execute(sql, review_idx)
        # read인 경우, 커서로 연결통로(스트림)에 검색결과를 꺼내주어야 한다
        row = cur.fetchone() # row 하나만 꺼내
        conn.close()
        return row
    except Exception as e:
        print("db 연결 중 에러 발생!!")
        print('에러 정보>> ', e)
  
```

12. 후기 긍정 부정 분석

```
import pymysql
def readAll():
    try:
        conn = pymysql.connect(
            host='database-1.cgind9azzzrj.ap-northeast-2.amazonaws.com',
            port=3306,
            user='root',
            password='db_password', # blind db_password
            db='shop2',
            charset='utf8'
        )

        # 연결된 통로(stream)을 통해, SQL문을 보내보자.
        # 2. 연결된 통로를 지정
        cur = conn.cursor(pymysql.cursors.DictCursor)

        # 3. sql문을 보내보자
        sql = 'select * from review'
        # 커서로 sql문을 보냄.
        result = cur.execute(sql)
        # read인 경우, 커서로 연결 통로에 검색 결과를 꺼.
        row = cur.fetchall()
        # print(row)
        conn.close()
        return row, result
    except Exception as e:
        print("db 연결 중 에러발생!!")
        print('에러정보>> ', e)
```

readall.py

```
import pymysql
from ai.review_pred.pos_neg import labeling
def update_db(review_idx):
    review_label = labeling(review_idx)
    print(review_label)
    print(review_idx)
    try:
        conn = pymysql.connect(
            host='database-1.cgind9azzzrj.ap-northeast-2.rds.amazonaws.com',
            port=3306,
            user='root',
            password='db_password', # blind db_password
            db='shop2',
            charset='utf8'
        )
        print('연결 성공!!', conn.host_info)
        cur = conn.cursor()
        sql = 'update review set review_label = "' + review_label + '" where review_idx = ' + str(review_idx)
        result = cur.execute(sql)
        conn.commit() # update한 것 반영
        # print('sql문 전송 결과>', result)
        conn.close()
        return result # 0: update 실패, 1: update 성공
    except Exception as e:
        print("db 연결 중 에러 발생!!")
        print('에러 정보>> ', e)
```

updatedb.py

12. 후기 긍정 부정 분석

posneg.py

```
# 라이브러리 추가
import ...
warnings.filterwarnings(action='ignore')
import re
import pandas as pd
import numpy as np
import urllib.request
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
import joblib
# 모듈 불러오기
pos_dict = joblib.load('/home/ubuntu/projects/ShoppingMallML/ai/review_pred/pos_dict.pkl')
neg_dict = joblib.load('/home/ubuntu/projects/ShoppingMallML/ai/review_pred/neg_dict.pkl')
plus_label = joblib.load('/home/ubuntu/projects/ShoppingMallML/ai/review_pred/plus_label.pkl')
tfidf = joblib.load('/home/ubuntu/projects/ShoppingMallML/ai/review_pred/tfidf.pkl')
SA_lr = joblib.load('/home/ubuntu/projects/ShoppingMallML/ai/review_pred/SA_lr.pkl')
```

```
# 하나의 내용에 대한 라벨(긍정/부정에 따른) 붙이는 함수
def labeling(i):
    tfidf_list = tfidf
    tfidf_module = tfidf_list[0]
    SA_module = SA_lr
    row = db_read(i)
    p_row = list(row)
    data = p_row[3]
    data = re.sub("[^ㄱ-ㅎㅌ-ㅣ가-힣 ]", "", data)
    pn_tfidf = tfidf_module.transform([data])
    predict = SA_module.predict(pn_tfidf)
    if(predict[0] == 1):
        p_row[7] = '1F60A'
        return p_row[7]
    else:
        p_row[7] = '1F621'
        return p_row[7]
```

+

12. 후기 긍정 부정 분석

review_all.html

views.py

```
def pn_review(req):
    data = req.POST
    review_idx = data['id']
    result = updatedb.update_db(review_idx)
    print(result)
    return redirect('/review/')

def read_all(req):
    review_all = readAll()[0]
    total_count = readAll()[1]
    context = {
        'review_all': review_all,
        'total_count': total_count
    }
    # print(review_all)
    return render(req, 'review/review_all.html', context)
```

```
<h3>구매 후기 조회(전체)</h3>
<form action="/review/pn_review" method="post">
    {% csrf_token %}
    <table border="1">
        <tr>
            <td class="c1">후기 판별(긍정/부정)</td>
            <td width="250">
                <input type="text" name="id" placeholder="review_idx를 입력하세요">
                <button>update</button>
            </td>
        </tr>
    </table>
</form>
<table border="1">
    <tr>
        <td class="c1">돌아가기</td>
        <td width="250">
            <button onclick="location.href='http://ec2-3-39-249-247.ap-northeast-2'">
        </td>
    </tr>
</table>
```

+ 12. 후기 긍정 부정 분석

```
<hr color="snow">
<p>전체 구매 후기 수: {{ total_count }}</p>
<table border="1">
  <tr>
    <td class="c1">후기 번호</td>
    <td class="c1">상품 번호</td>
    <td class="c1">고객 아이디</td>
    <td class="c1">후기 내용</td>
    <td class="c1">후기 이미지</td>
    <td class="c1">후기 평점</td>
    <td class="c1">후기 등록 시간</td>
    <td class="c1">후기 라벨</td>
  </tr>
  {% for one in review_all %}
  <tr>
    <td width="200">
      {{ one.review_idx }}
    </td>
    <td width="200">
      {{ one.product_idx }}
    </td>
    <td width="200">
      {{ one.user_id }}
    </td>
    <td width="200">
      {{ one.review_content }}
    </td>
    <td width="200">
      {{ one.review_img }}
```

```
</td>
<td width="200">
  {{ one.review_idx }}
</td>
<td width="200">
  {{ one.product_idx }}
</td>
<td width="200">
  {{ one.user_id }}
</td>
<td width="200">
  {{ one.review_content }}
</td>
<td width="200">
  {{ one.review_img }}
```

인스턴스에 연결 | EC2 Management Console

여행일정 게시판 상세보기

구매 후기 조회(전체)

구매 후기 조회(전체)

후기 판별(긍정/부정)

review_idx를 입력하세요

update

돌아가기

상품 전체 목록으로 이동

전체 구매 후기 수: 12

후기 번호	상품 번호	고객 아이디	후기 내용	후기 이미지	후기 평점	후기 등록 시간	후기 라벨
1	1	admin	색감이 곱네요	good!.png	4.5	May 12, 2022, 9:10 p.m.	1F621
3	1	aaaaaa1234	색이 오묘한게 정말 예뻐요 여름에 진짜 잘 입을 것 같아요 케 키케	good!.png	5.0	May 12, 2022, 9:20 p.m.	1F60A
4	1	use06048	우선 옷은 되게 얇아요! 그리고 제가 키가 큰 편이라 너무 짧고 작을까봐 걱정했는데 입고다 널만 합니다! 다만 하이웨스트 바지와 함께 입어야 할 듯!	good!.png	4.5	May 12, 2022, 9:20 p.m.	1F621
5	1	qwer	색감이 너무 마음에 들어요! 사이즈도 괜찮고	good!.png	4.0	May 12, 2022, 9:20 p.m.	1F60A

review_all.html

12. 후기 긍정 부정 분석

admin 2022. 6. 2. 오전 3:14:09

[3점] 좋아요 ㅎㅎ



수정

삭제

후기 분석 표시

후기 분석 전

후기 분석 후



분석 결과 예시

구매 후기

중의적인 표현을 사용할 경우 부정으로 판단될 수 있으니 유의바랍니다

김들리 2022. 5. 13. 오전 6:20:53

[5점] 색이 오묘한게 정말 예뻐요 여름에 진짜 잘 입을 것 같아요 케키케



수정

삭제

