머신러닝 데이터 수집과 분석 시각화

비트캠프 **KDT5**기 수강생 장원종

1. 과제 개요

시카고 샌드위치 맛집 분석

- 시카고 매거진 홈페이지에 접속해서 샌드위치 가게 정보를 수집
- **Top50** 개의 샌드위치 가게를 전처리를 통하여 추출한다.
- **Google map**에 샌드위치 가게 데이터를 바탕으로 위치정보를 표시해준다

1. 라이브러리 패키지 설정

```
import re
from urllib.parse import urljoin
from urllib.request import urlopen, Request

import numpy as np
import pandas as pd
from bs4 import BeautifulSoup
import folium
from tqdm import tqdm

from context.domains import Reader, File
```

2. 이닛 및 훅 처리**(OOP)**

```
class Solution(Reader):
    def __init__(self):
        self.url_base = 'http://www.chicagomag.com'
        self.url_sub = '/Chicago-Magazine/November-2012/Best-Sandwiches-Chicago/'
        self.url = self.url_base + self.url_sub
        self.file = File()
        self.file.context = './data/'

    def hook(self):
        def print_menu():
            print('0. Exit')
            print('1. preprocessing.')
            print('2. filter.')
            print('3. map.')
            return input('메뉴 선택 \n')

        while 1:
            menu = print_menu()
            if menu == '0':
                break
            if menu == '1':
                self.preprocessing()
            elif menu == '2':
                self.filter()
            elif menu == '3':
                self.map()
            elif menu == '0':
                break
```

3. 데이터 전처리

```
def preprocessing(self):
    req = Request(self.url, headers={'User-Agent': 'Mozilla/5.0'})
    html = urlopen(req)
    soup = BeautifulSoup(html, "html.parser")
    rank = []
    main_menu = []
    cafe_name = []
    url_add = []
    list_soup = soup.find_all('div', 'sammy')

    for item in list_soup:
        rank.append(item.find(class_='sammyRank').get_text())
        tmp_string = item.find(class_='sammyListing').get_text()
        main_menu.append(re.split(('\n|\r\n'), tmp_string)[0])
        cafe_name.append(re.split(('\n|\r\n'), tmp_string)[1])
        url_add.append(urljoin(self.url_base, item.find('a')['href']))
    data = {'Rank': rank, 'Menu': main_menu, 'Cafe': cafe_name, 'URL': url_add}
    df = pd.DataFrame(data, columns=['Rank', 'Cafe', 'Menu', 'URL'])
    df.head()
    df.to_csv("./data/best_sandwiches_list_chicago.csv", sep=',', encoding='UTF-8')
```

4. 데이터 필터링

```
def filter(self):
    file = self.file
    file.fname = 'best_sandwiches_list_chicago'
    df = self.csv(file)

    price = []
    address = []
    for i in df['URL']:
        req = Request(i, headers={'User-Agent': 'Mozilla/5.0'})
        html = urlopen(req)
        soup_tmp = BeautifulSoup(html, 'lxml')
        gettings = soup_tmp.find('p', 'addy').get_text()
        price.append(gettings.split()[0][:-1])
        address.append(' '.join(gettings.split()[1:-2]))

    df['Price'] = price
    df['Address'] = address
    df = df.loc[:, ['Rank', 'Cafe', 'Menu', 'Price', 'Address']]
    df.set_index('Rank', inplace=True)
    print(df.head())
    df.to_csv('./data/best_sandwiches_list_chicago2.csv', sep=',', encoding='UTF-8')
```

5. 구글 맵 출력 및 메인 메소드

```
    def map(self):
        file = self.file
        file.fname = 'best_sandwiches_list_chicago2'
        df = self.csv(file)
        gmaps = self.gmaps()
        lat = []
        lng = []

        for i in tqdm(df.index):
            if df['Address'][i] != 'Multiple':
                target_name = df['Address'][i] + ', ' + 'Cicago'
                gmaps_output = gmaps.geocode(target_name)
                location_output = gmaps_output[0].get('geometry')
                lat.append(location_output['location']['lat'])
                lng.append(location_output['location']['lng'])
            else:
                lat.append(np.nan)
                lng.append(np.nan)
        df['lat'] = lat
        df['lng'] = lng

        mapping = folium.Map(location=[df['lat'].mean(), df['lng'].mean()], zoom_start=11)
        for i in df.index:
            if df['Address'][i] != 'Multiple':
                folium.Marker([df['lat'][i], df['lng'][i]], popup=df['Cafe'][i]).add_to(mapping)
        mapping.save('./data/map.html')

if __name__ == '__main__':
    Solution().hook()
```

6. 시행 결과