SBA온보딩 내용(파이썬 활용)

제목주제 : 파이썬을 활용한 네이버쇼핑 크롤링 프로젝트

작성일자 : 2022. 12. 28

작성자 : 연구소 데이터분석팀 엄태영PM

파이썬을 활용한 네이버쇼핑 크롤링 프로젝트

배경

네이버 쇼핑의 검색 결과 페이지에서 구매건수에 영향을 미치는 항목을 분석하기 위하여 크롤링을 진행해야하는 과정에서 파이썬을 이용하여 해당 프로젝트를 진행하였습니다.

크롤링이란

우선 크롤링이란 인터넷에서 데이터를 검색해 필요한 정보를 색인하는 것을 의미합니다.

사용자가 키워드를 하나씩 검색하여 정보를 얻고 저장 및 가공 과정을 대신 해주는 기술이라고 할 수 있습니다.

파이썬이란

우선, 파이썬은 크롤링을 할 때 가장 많이 사용하는 언어입니다.

파이썬 선택이유

이번 프로젝트에서 파이썬을 사용한 이유는 문법이 간결하고 특정 기능이 들어있는 소스코드의 묶음인 라이브러리의 활용성이 뛰어나 쉽고 빠른 개발이 가능한 언어이기 때문입니다.

파이썬 특징

파이썬의 특징으로는 컴파일 과정 없이 인터프리터(Interpreter, 해석기)가 소스 코드를 한 줄씩 읽어 들여 곧바로 실행하는 스크립트 언어(Script language)점 과

자료형 변환 시 번거로운 과정을 거치지 않아도 되는 동적타입 언어라는 점 입니다.

또한, 리눅스(Linux), 유닉스(Unix), 윈도우즈(Windows), 맥(Mac) 등 대부분의 운영체제(Operating System, OS)에서 모두 동작합니다.

운영체제별로 컴파일할 필요가 없기 때문에 한 번 소스 코드를 작성하면 어떤 운영체제에서든 활용이 가능합니다

파이썬은 간결하고 쉬운문법으로 빠른 개발속도를 만들어 높은 생산성을 만들 수 있습니다.

또한 대표적인 글루(Glue)언어이기 때문에 다른 언어나 라이브러리에 쉽게 접근해 연동할 수 있어서 높은 확장성과 이식성을 가지고 있습니다.

파이썬은은 구글, 인스타그램, 넷플릭스, 스포티파이, 드롭박스 등 많은 기업에 사용하고 있습니다.

파이썬 라이브러리

파이썬에서는 라이브러리를 사용하기 위해서는 상단에 from 과 import를 사용하는데,

해당 프로젝트를 진행하기 위해서는 requets와 데이터를 가공하기 위하여 json, urllib등 라이브러리를 사용하였고, 엑셀을 사용하기 위하여 WorkBook에 openpyxl를 사용하기도 하였습니다.

크롤링을 위한 라이브러리는 대표적으로 request와 selenium이 있습니다.

이 프로젝트에서는 request 라이브러리를 사용했습니다.

selenium을 사용하지 않은 이유는 상대적으로 동적수집 형태로 브라우저를 직접 조작하고 브라우저가 실행될때까지 기다려주기도 해야해서 그 속도가 느리다는 특징이있습니다.

때문에 크롤링을 빠르게 진행하기 위해서 request 라이브러리를 사용하였습니다.

requests는 복잡한 HTTP 요청과 쿠키 헤더를 잘 저리하며 그외에도 많은 기능이 존재합니다.

장점으로는 endpoint 만 알고 있으면 빠르게 요청하고 스크레핑이 가능 하고 단점으로는 여기저기 모두 사용할 수 있는 범용성은 떨어집니다.

소스코드 작동순서

1. input을 이용하여 크롤링을 진행할 검색어를 입력한다. (한국어의 경우 decode가 필요하므로 parse.qoute함수를 사용)
2. 데이터를 가져와야하기 때문에 search.shopping.naver.com 에 request를 보내서 header변수에 json방식으로 authority, user-agent, referer 등 정보를 작성한다 (referer 내용 중 query부분에 디코딩시킨 변수를 넣어준다.)
3. 추가로, 네이버쇼핑에 request하기 위해 params 정보를 작성하고, requests.get 함수를 사용하여 데이터를 전달받는다.
4. 전달받는 모든 데이터를 엑셀파일로 저장하기 위하여 Workbook 객체를 ws로 생성하고 시트와 컬럼을 추가한다.
5. request.get함수를 page만큼 반복 요청하고, response.text를 json.loads로 json화 시켜서 요청받은 데이터를 배열로 만든다. (반복문이 돌면서 전달받는 데이터를 ws에 append시킨다)
6. 중복되는 데이터에 대해여 중복처리를 진행하고, 작업이 마무리 되면 ws.save를 사용하여 엑셀파일로 저장하여 다운로드한다.

위 내용에서 input을 통해 검색어를 변수처리하는데 한국어 경우 인코딩되어 변수에 저장되기 때문에 디코딩을 진행한다.

작업한 소스코드

```
#from ast import keyword
from asyncio.windows_events import NULL
import requests
import json
from dateutil.parser import parse
from openpyxl import Workbook
from datetime import datetime
from urllib import parse

inputValue = input('크롤링을 진행할 검색어를 입력하세요 : ')
decode_Value = parse.quote(inputValue)
headers = {
    'authority': 'search.shopping.naver.com',
    'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.0.0 Safari/537.36',
    'referer': 'https://search.shopping.naver.com/search/all?query='+decode_Value+'&cat_id=&frm=NVSHATC',
    'accept-language': 'ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7',
}
wb = Workbook(write_only=True)
ws = wb.create_sheet('네이버쇼핑 크롤링 데이터')
columns = ['순위', '상품이름', '상품가격', '카테고리', '리뷰평점', '리뷰개수', '구매건수', '찜 개수', '톡톡사용여부', '상품등록일', '키워드']
ws.append(columns)ㅍ

def isRepeat(previousItemList, itemList) : #데이터 중복확인함수
    if previousItemList['shoppingResult']['products'][0]['productName'] == itemList['shoppingResult']['products'][0]['productName']:
        print('----중복되어서 끝 혹은 10페이지 미만이여서 끝-----')
        return True
    return False

def printData(itemList) : # 데이터 변수처리 함수
    for i in itemList['shoppingResult']['products']:
      title = i['productName']                                          # 상품명
      price = i['price']
      category1 = i['category1Name']
      category2 = i['category2Name']
      category3 = i['category3Name']
      category = category1+' > '+category2+' > '+category3              # 카테고리
      scoreInfo = i['scoreInfo']                                        # 리뷰평점
      rank = i['rank']                                                  # 상품랭킹
      reviewCnt = i['reviewCount']                                      # 리뷰개수
      purchaseCnt = i['purchaseCnt']                                    # 구매건수
      openDate = i['openDate']
      resultDate = openDate[:9]
      openDate =resultDate[0:4]+'-'+resultDate[4:6]+'-'+resultDate[6:8]     # 상품등록일
      keepCnt = i['keepCnt']                                            # 찜개수
      try:
        istalktalk = i['channelInfoCache']['talkAccountId']
```

```
        istalktalk = 'O'                                                    # 톡톡사용여부
      except:
        istalktalk ='X'
      keywords = i['characterValue']                                        # keyword
      row = [rank, title, price, category, scoreInfo, reviewCnt, purchaseCnt, keepCnt, istalktalk, openDate, keywords]
      ws.append(row)

def makeRequestAndGetResponse(number) : # request보내기 함수
    pageingIndex = number
    params = (
      ('sort', 'rel'),
      ('pagingIndex', pageingIndex),
      ('pagingSize', '40'),
      ('viewType', 'list'),
      ('productSet', 'total'),
      ('deliveryFee', ''),
      ('deliveryTypeValue', ''),
      ('frm', 'NVSHATC'),
      ('query', inputValue),
      ('origQuery', inputValue),
      ('iq', ''),
      ('eq', ''),
      ('xq', ''),
     )
    response = requests.get('https://search.shopping.naver.com/api/search/all', headers=headers, params=params)
    return response

previousItemList = []
number = 1
while number <  11:                                 # 10페이지 설정 후 10번 반복
  print('-----네이버쇼핑 크롤링 작업중 ',number,'/10 -----')
  response = makeRequestAndGetResponse(number)      # request보내기 함수
  itemList = json.loads(response.text)
  if number == 1 :
      number = number + 1
      previousItemList = itemList
      printData(itemList)                           # 데이터 변수처리 함수
      continue
  if isRepeat(previousItemList, itemList) : #데이터 중복확인함수
      break
  previousItemList = itemList
  printData(itemList)
  number = number + 1

now=datetime.now()
excel_title = str(now.date())
wb.save('D:\crawling/files/naver_crawling_'+inputValue+'_'+excel_title+'.xlsx') # 엑셀파일제목설정
print('----------네이버쇼핑 크롤링 작업 완료 ----------')
```