Python을 활용한 데이터 분석 전문가 양성과정 교육 후기

글. 이정우 CBS 플랫폼개발부 엔지니어

들어가며

방송기술교육원에서 지난 11월 9일부터 11일까지 3일간 진행한 'Python을 활용한 데이터 분석 전문가 양성과정' 교육에 참석했다. 개발자로서 현재는 'Client side'의 개발을 맡고 있지만, 향후 'Server side'까지 지식과 기술을 쌓을 때 도움이 되리라는 기대를 가지고 있었다.

다양한 미디어의 등장과 함께 미디어를 이용한 콘텐츠 이용 행태가 변화하고 있고, 유저들이 어떠한 의도로 우리 콘텐츠를 찾아왔는지, 유입 경로는 어떠했는지, 어디서 감동했고 왜 이탈했는지 분석하거나 또는 우리가 생산한 콘텐츠는 어떠한 속성들을 가지고 있는지 등 데이터를 분석하는 방법을 아는 것이 대단히 중요해졌기 때문이다.

어디선가에서 듣기로는 머신러닝과 인공지능을 도입할 때, 그리고 대규모 데이터를 처리할 때 가장 많이 고려되는 언어가 Python이라는 이야기도 있었기 때문에 언젠가는 필요하겠구나 싶어서 이번 기회를 잡게 되었다.

일정

강사는 AI Education Lab의 오영제 교수 한 분으로 3일에 걸친 전 일정을 맡아주었다.

1일 차 : 파이썬 소개 및 특징, 개발환경 세팅, 기초문법
2일 차 : 기초 문법, 라이브러리 사용
3일 차 : 실습 프로젝트 : 웹 크롤링, 비트코인 실시간 가격조회

교육은 기본적인 툴 교육도 포함되어 실습 위주로 진행되었다. 기초 문법의 경우에는 강사가 기능과 문법을 소개해주며 코드를 작성하면, 교육생들이 따라 해 보는 방식이었으며, 한 주제가 끝날 때쯤 연습문제를 제시해 교육생 개인의 힘으로 주어진 시간 동안 풀었다.

왜 파이썬인가?

교육을 시작하면서 가장 먼저 언어에 대한 소개를 들었다. 프로그래밍을 위해 여러 언어들이 소개된 와중에, 파이썬은 91년에 네덜란드의 프로그래머 'Guido Van Rossum'이 만든 언어이다. 오픈소스이며, Compiler 언어가 아닌 Interpreter 언어이다. 기존에 Interpreter 언어를 접해보지 않아 생소한 개념이어서 개인적으로 더 많이 다뤄보고 여러 문제를 해결하며 경험을 쌓아 봐야 이해할 수 있을 것 같다. 다만 강사님이 다른 유명한 언어들인 Java, C언어와 함께 비교해주어서 감을 조금 잡을 수 있었다. 바로 파이썬이 학습 속도와 개발 생산성, 유지보수성 면에서 Java나 C 같은 Compiler 언어에 비해 유리하다는 점이었다.

재미있는 통계가 있다. 신입직원 채용 시 코딩 테스트를 하는 카카오의 경우, 2017년에 지원자의 11%만이 파이썬을 주 언어로서 시험을 보았으나, 2019년에 파이썬을 택하여 응시한 지원자가 60%에 육박했다는 사실이다. 같은 문제를 해결하려고 할 때, 파이썬이 더 간결한 코드로 문제가 해결되기 때문에 시간이 적게 들어 시험 치르는 입장에서 더 많은 문제를 해결할 수 있어서 파이썬이 인기를 얻어가고 있다는 설명이었다.

파이썬이 장점만 존재하는 언어는 아니다. 컴파일 언어가 아니기 때문에 연산의 수행 속도가 (다른 언어에 비해) 느리다는 단점이 있다. 하지만 구글의 연구에 의하면, 상용 서비스에 있어서 CPU의 연산속도보다는 파일 입출력과 특히 네트워크상에서의 병목현상에서 속도 문제가 발생하기 때문에 그다지 심각한 약점은 아니라고 한다. 속도 문제가 심각하게 일어난다면 그 부분만 컴파일 언어를 적용하여 해결하는 방법을 쓴다.
파이썬의 장점 중 개인적으로 가장 매력적인 부분은 바로 풍부한 외부 라이브러리이다. 인공지능 개발을 위한 오픈소스 프레임워크인 텐서플로우를 쓸 수 있다. 텐서플로우가 아니더라도, 오래 인기를 끄는 언어라 많은 라이브러리가 완숙한 수준일 것으로 예상되어 매력적이라고 느꼈다.

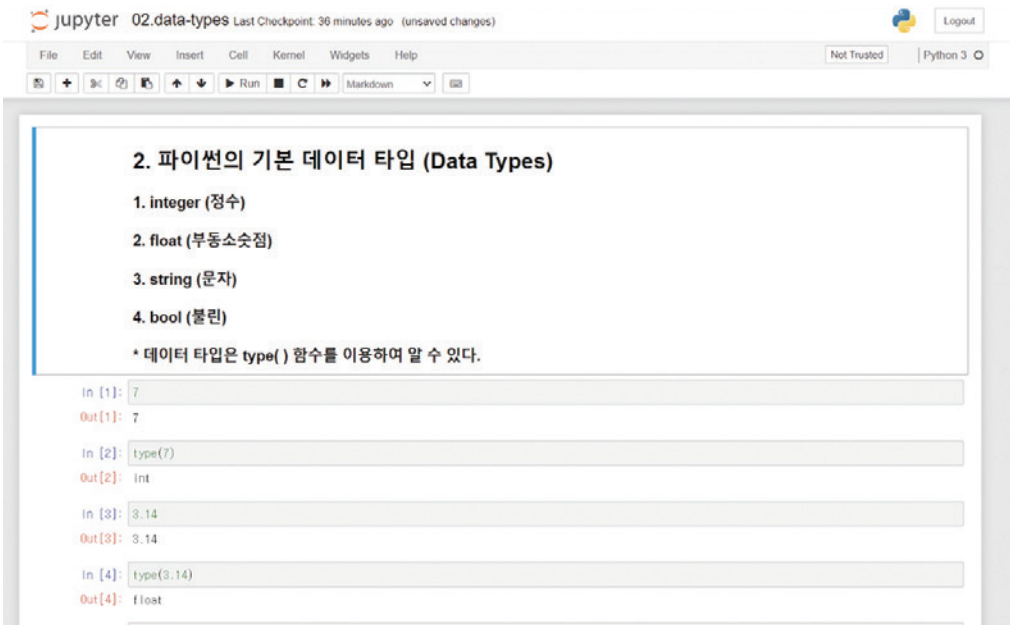
특별했던 개발환경 세팅

Compiler 언어만을 다뤄왔던 터라 Interpreter 언어인 파이썬 개발환경을 구축하는 과정 또한 특별하게 다가왔다. Client 사이드의 개발을 위해 타겟 OS 제조사가 제공하는 통합개발환경(IDE)과 해당 OS 상의 응용 애플리케이션 개발용 Framework를 사용해왔다. 이번 교육에서 우리는 Jupyter Notebook이라는 무료 개발툴을 사용했다. 이를 실행하자, 웹브라우저가 열렸다. 충격이었다. 웹브라우저에서 코드를 실행시킨다고? 나중에 본인이 가진 'IDE라면 이래야지!'라는 고정관념에 더 부합하는 VSCode라는 툴을 사용하여 프로그래밍하는 시간도 있었지만, 두 가지를 다 써본 뒤 드는 생각은 Jupyter Notebook이 더 편하고 좋았다는 것이다. '셀' 단위로 코드를 실행할 수 있으며, 셀의 추가/제거, 셀 코드 실행, 커널 끄기/재시작 등 생소하고 편리한 기능들이 많았다.

이번 실습에는 사용하지 않았지만, 구글의 Colaboratory 또한 Jupyter Notebook을 내장한 클라우드 기반 개발환경 또한 소개받았다. 머신러닝과 인공지능을 개발할 때 유용하다고 하니, 나중에 사용할 일이 있을 것 같다.

특징 1. 자료형

파이썬의 특징 중 하나는 바로 변수 선언할 때 자료형(type)을 명시하지 않고, 할당하는 데이터의 종류에 따라 dynamic 하게 정해진다는 점이다.



대부분의 언어		파이썬	
String name = "김이병" ;		name = "김병장"	# type(name) : string
int age = 21;		age = 22	# type(age) : int
float height = 175.5;		height = 180.4	# type(height) : float

위와 같은 차이다. 자료형을 명시하지 않으니, 얼마나 편한가. 생산성이 높은 면모를 여실히 보여준다고 할 수 있다.

특징 2. 자료구조의 처리

데이터셋을 연산하기 위해서 프로그램 안에서 데이터 모음을 어떻게 구조화하고 처리할 것인지 고민하게 된다. 이때, iterable(반복형태) 자료형을 사용하게 되는데, 파이썬에서는 list와 tuple, dictionary라는 자료형이 제공된다. 다른 언어들도 명칭은 조금씩 다르지만 비슷한 자료형이 있다. 리스트의 경우에는 데이터셋 내부의 특정 요소를 지시하고 싶을 때, 0번지부터 시작하는 '인덱스' 정수를 사용한다.

```
hello = ['H', 'e', 'l', 'l', 'o', '!']
```

위에서 hello[0] = 'H' 이고 hello[4] = 'o' 이다. 여기까지는 본인이 기존에 다루어봤던 대부분의 언어와 성격이 같았다. 파이썬이 특이했던 부분은, 번지수를 음수로 지시했을 때 리스트의 마지막 요소로부터 역으로 카운트해서 지시할 수 있다는 점이었다.

index	0	1	2	3	4	5
hello =	['H'	'e'	'l'	'l'	'o'	'!']
index	-6	-5	-4	-3	-2	-1

인덱스를 이용한 리스트 slicing도 간편했다. 본인이 기존에 다루어봤던 언어들에서 slicing 할 때보다 대단히 간편해서 아주 감격스러웠다.

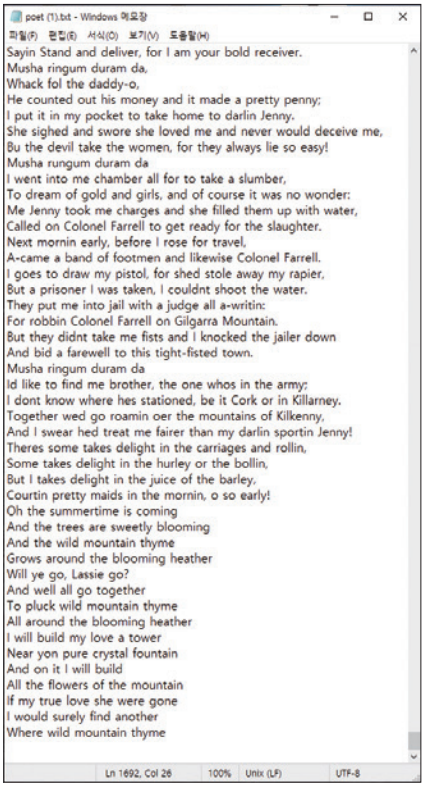
명령	리턴 값(결과물)	예시	
list[start : end]	start ~ end-1	hello[1:4]	['e','l','l']
list[start :]	start ~ 끝까지	hello[1:]	['e','l','l','o','!']
list[: end]	처음 ~ end-1	hello[:4]	['H','e','l','l']
list[:]	list 전체	hello[:]	['H','e','l','l','o','!']
list[start:end:increment]	증가분	hello[:-2:2]	['H','l']

심지어 문자열에도 위를 적용할 수 있는데, 예를 들어 one_sentence = "Python is powerful"로 선언하고 one_sentence[::-1] 명령하면 결과물이 "lufrewop si nohtyP"이 나온다.

간단한 응용

기본적인 문법을 배우면서 조금 더 실전에 가까운 알고리즘을 짜는 시간을 가지게 되었다. 먼저 영어로 작성된 시들을 모아놓은 텍스트 파일(1,692줄)을 읽은 뒤 많이 쓰인 단어 순으로 데이터를 추출해내는 과제였다. 파일 입출력 함수를 통해 텍스트 파일을 읽어 들인 뒤, line-by-line 연산을 위해 for 루프를 만든다. 내부에서 해당 line을 단어로 쪼개는 for 루프를 만들고, 총괄 dictionary에 단어별 count를 증가시키는(해당 단어가 없다면 dictionary에 해당 항목을 신설) 코드를 작성한다. 이후 총괄 딕셔너리를 정렬하고 출력하는 코드를 작성했다. 이때, 전치사처럼 관심 없는 단어들을 지정하여 출력하지 않도록 작성할 수 있었다.

방송국에서 생산한 콘텐츠에서 메타데이터를 추출하려고 한다면 어떻게 구현해야 할까? 이번 과제는 raw data가 완전무결하다는 점, 그리고 영어라는 언어의 문법적 특성상 단어 구별이 쉽다는 점을 고려해야 한다. 실제 업무에 적용하려면 한국어에서 많이 쓰이는 합성어에 대한 처리, 단어마다 붙어있는 조사에 대한 처리 등등 이번 과제보다 훨씬



복잡한 문제에 대한 대응이 필요하다.

그리고 웹사이트를 Crawling 하는 프로그램을 작성해보았다. 보통 웹 서핑을 할 때는 대략적으로 이와 같은 과정을 거친다 : 웹브라우저 켜기 → 서비스 제공자의 주소를 입력 → 서버가 제공하는 html 문서를 response로 수신 → 웹브라우저가 html 문서 파싱 및 렌더링 → 이용자가 예쁘게 꾸며진 웹페이지를 보게 된다. 여기에서 html 문서의 원본을 보면서 우리가 원하는 정보만을 추출해내는 것이다. 정보를 갱신할 주기를 정해서 추출 알고리즘을 while문 안에서 주기적으로 돌려주면 크롤러를 작성할 수 있다. html 문서로부터 우리가 원하는 정보를 추출할 때, 정규식(regular expression)을 사용하거나 또는 더 간편한 라이브러리인 BeautifulSoup을 사용하여 추출하였다. 우측의 이미지는 방송기술교육원 홈페이지의 공지사항 제목 목록을 추출하는 프로그램이다.

여기에서 한발 더 나아가, 주식이나 암호화폐 가격정보를 제공해주는 야후파이낸스, 인베스팅닷컴에서 삼성전자, 테슬라, 비트코인의 가격을 크롤링해서 그래프로 보여주는 과제도 수행했다. 윈도우 창을 만들어서 가격을 5초마다 갱신해서 보여주는 프로그램도 만들어보았다.

다음 file 을 읽어서 가장 빈번하게 나타나는 top 10 단어들을 출력

1. "poet.txt" file 을 open
2. count dictionary 생성
3. count.items() 를 이용하여 (key, value) list 생성
4. value 의 reverse 순으로 정렬 -> sorted(list, reverse=True)

In [7]:

```
f = open('poet.txt')
counts = dict()
for line in f:
    for word in line.split():
        word = word.lower()
        if not word in counts:
            counts[word] = 1
        else:
            counts[word] +=1

# sorted(counts.items(), key = lambda kv:kv[1], reverse = True)
word_list = []
for k, v in counts.items():
    word_list.append((v,k))

stop_word = ['the', 'and', 'a', 'to', 'i', 'of', 'in', 'for', 'is', 'was', 'am', 'were', 'that', 'on', 'but', 'are', 'be']
for v, k in sorted(word_list, reverse = True):
    if k in stop_word:
        continue
    else:
        print((v,k))
```

```
(240, 'my')
(146, 'me')
(114, 'all')
(108, 'you')
(107, 'she')
(102, 'with')
(96, 'her')
(89, 'as')
(81, 'when')
```

실전 연습

방송기술 교육원 게시판 제목 검색

In [37]:
```
url = 'http://edu.kobeta.com/community/' #방송기술교육원 '공지사항'

titles = []

res = req.urlopen(url)
soup = BeautifulSoup(res, 'html.parser')
divs = soup.find_all(class_ = "notice-list-middle")

for div in divs:
    for dd in div.find_all('dd'):
        if re.search('[가-힣a-zA-Z ]', dd.text):
            titles.append(dd.text)
```

In [38]:
```
titles
```

Out[38]:
```
['2021 콘텐츠 QC 세미나 개최',
 '위탁교육 홍보 및 안내',
 '지역 방송기술 세미나 희망 지역사 의견 접수',
 '2021년도 컬러 그레이딩 전문가 양성과정 선발 결과 안내',
 '방송 네트워크 파일기반 제작 시스템 교육 일정 연기 안내',
 '5G Broadcast World Forum',
 '[외부교육 안내] 2020 저널리즘아카데미 위드 코로나, 변화의 키워드 …',
 '2020 Total Production Control System 전문가 양성과정 신청 안내',
 '2019 글로벌 IP Technical 전문가 양성과정 신청 안내',
 '2019 글로벌 IP Technical 전문가 양성과정 선발 결과 안내',
 '2019 글로벌 Broadcast ATSC 3.0 전문가 양성과정 선발 결과 안내',
 '2019 글로벌 Broadcast ATSC 3.0 전문가 양성과정 신청 안내',
 'KOC 2018 "연결이 가치를 만들어낸다" 개최 안내']
```

마치며

처음 C언어를 배우던 그 시절이 생각났다. 당시 들은 수업은 2학점짜리 과목이었지만 다른 3학점짜리 전공과목들보다 더욱 오랜 시간을 들여 공부해야 했다. 이번 교육 일정이 다소 짧아서 아쉬운 점이 있었지만, 학생시절 C언어를 배울 때 그러했듯 문법이 익숙해질 때까지 혼자서 수련하고 본인이 필요한 프로그램을 직접 만들어보는 시간을 가진다면 이번 교육을 온전히 소화할 수 있을 것이다.

회사 선배가 어떤 인기 많은 상품을 온라인으로 예약해야 했는데, 예약해주는 프로그램을 파이썬으로 직접 짰다는 이야기를 듣고 그 능력을 부러워했던 적이 있다. KOBA 때 어떤 회사에서 타 언론사의 뉴스 헤드라인들을 크롤링해서 보여주는 프로그램을 선보였던 바 있다.

이번 교육을 받은 이후에, 본인도 조금만 더 공부하고 익히면 이러한 프로그램을 혼자 힘으로 만들 수 있겠다는 자신이 생겼다. 나아가 인공지능, 자연어 처리와 같은 라이브러리들도 사용하게 되면 우리 회사 콘텐츠로부터 메타데이터를 추출하는 등 유의미한 데이터를 추출 할 수 있는 역량을 가질 수 있겠다.

3일간의 파이썬 교육이었지만, 당장 실생활에도 적용할 수 있는 기술을 갖추게 된 좋은 결과를 얻었다. 끝으로 3일 내내 고생하며 강의해주신 오영제 교수님께 다시 한번 감사드린다.