**Capstone Design 1 Final Report**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **과제명** | Capstone Design 1 | | | |
| **프로젝트 설명** | 데이터분석 &코디 추천 웹사이트 | | | |
| **팀명** | 패피(패션피플) | | | |
| **수행기간** | 2019.03. ~ 2019.06 | | | |
| **지도교수** | **성명** | **소속** | **연락처** | **이메일** |
| 정현숙 | IT융합대학 | 010-2508-9231 |  |
| **팀원** | **학과** | **학번** | **성명** | **이메일** |
| 컴퓨터공학과 | 20144800 | 천승현 | ehwjsx@gmail.com |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**캡스톤 디자인 최종 보고서**

서론

빅데이터에 대한 이해

빅데이터의 개념

빅데이터의 적용

추천 시스템

본론

개발동기

기존 시스템 소개,분석

-내옷장(어플)

-스타일닷컴(웹사이트)

개선방안

개발 제목

프로젝트 소개

프로젝트 세부내용

-일정표

-개발 환경 설정

-개발 현황

-코드 해석

-실행 화면

결론

피드백

부록

전체코드

개발환경

참조

github

**서론**

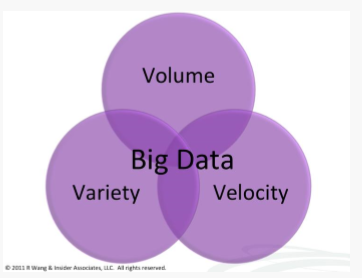
빅데이터에 대한 이해

빅데이터는 현대 4차산업혁명에서 가장 주목받는 중요한 IT산업 키워드이다. 빅데이터는 여러가지 관점에 따라 다양한 정의를 내릴 수 있는데, IDC에서는 2011년 6월 빅데이터가 다양한 종류의 대규모 데이터로부터 저렴한 비용으로 가치를 추출하고, 데이터의 빠른 수집, 발굴, 분석을 지원하도록 고안된 차세대 기술 및 아키텍처라고 업무수행 방식에 초점을 맞춘 정의를 내렸다. 맥킨지에서는 2011년 5월 빅데이터는 데이터의 규모에 초점을 맞췄을 때 기존 데이터베이스 관리도구의 데이터 수집, 저장, 관리, 분석 역량을 넘어서는 데이터라고 정의 했다. 종합하여 평가하자면 디지털 환경에서 생성되는 데이터로 그 규모가 방대하고, 생명주기가 짧고, 형태도 수치 데이터 뿐만 아니라 문자와 영상 데이터를 포함하는 대규모 데이터라고 정의 할 수 있다.

디지털 정보량의 기하급수적인 증가에 따라 데이터의 종류가 다양해져 도로, 건축물 등에 내장된 센서 및 임베디드 시스템, 정형, 비정형, 소설, 실시간 데이터 등의 복합적인 구성을 통해 생각과 의견까지 분석 및 예측 가능한 환경이 구성 되고 있다.

빅데이터의 개념

빅데이터의 개념에서 가장 중요한 특징은 3V이다. 3V는 빅데이터의 3가지 가장 주요한 특징의 단어들을 뜻한다. BI/DW 리서치 기관인 TDWI가 빅데이터의 3대 요소를 아래와 같이 그림으로 표현 했다.

  
Volume(용량)

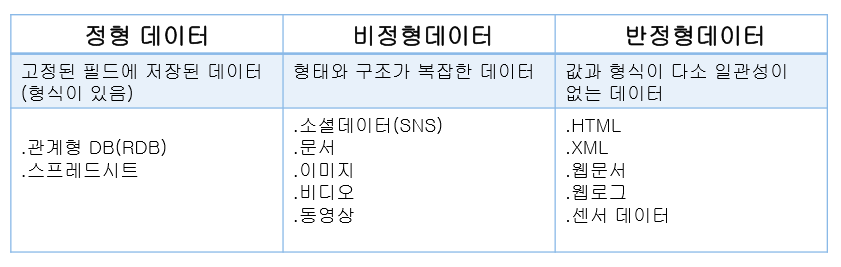
빅데이터는 대용량의 크기를 가진다.일반적으로 수십 테라바이트(terabyte) 혹인 수십 페타바이트(petabyte) 이상이 빅데이터의 범위에 해당한다. 1 페타바이트는 5기가바이트 DVD 영화를 약 17만 4천편 담을 수 있는 용량이다. 빅데이터는 기존 파일 시스템에 저장하기 어려울 뿐만 아니라 데이터 분석을 위해 사용하는 기존 DW(데이터 웨어하우스) 같은 솔루션에서 소화하기 어려울 정도로 급격하게 데이터의 양이 증가하고 있다. 이러한 문제를 극복하기 위해 확장 가능한 방식으로 데이터를 저장하고 분석하는 분산 컴퓨터 방식이 필요하다. 현재 사용 되는 분산 컴퓨팅 방식에는 아파치의 하둡이 대표적이다.

Velocity(속도)

속도는 두가지의 해석이 가능하다. 첫번째는 데이터의 빠른 생성 속도를 뜻하고 두번째는 데이터의 처리 속도의 개념이다. 사물 데이터, 웹 검색 데이터, 실시간 전송 데이터, 모바일에서 생성되는 데이터들은 지속적이고 빠른 속도로 생성되는 경우가 많다. 이러한 빅데이터들은 빠르게 분석되고 처리 되어야 한다. 오늘날 디지털 데이터는 매우 빠른 속도로 생성 되기 때문에 데이터의 수집, 저장, 분석등이 실시간으로 처리돼야 한다.

Variety(다양성)

빅데이터의 3V중 마지막 개념은 데이터의 다양성이다. 데이터는 그 특징에 따라 정형, 반정형, 비정형으로 구분 된다. 졍형 데이터는 고정된 필드에 저장되는 데이터를 의미하며 일정한 형식을 갖추고 있다. 반정형 데이터는 고정된 필드는 아니지만 메타데이터나 스키마등을 포함한다. 비정형 데이터는 고정된 필드에 저장되지 않는 데이터를 뜻한다. 사진, 동영상, 위치 정보, 통화 내용 등이 해당된다.

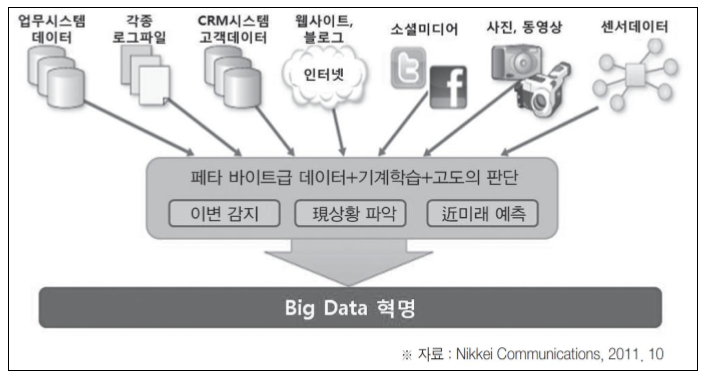


이외에도 Value(가치)라는 새로운 속성이 더해져서 4V라고 불리기도한다.

빅데이터의 적용

빅데이터의 기술에는 크게 4가지의 단계가 있다. 수집, 저장 및 관리, 처리 및 분석, 시각화 및 활용단계이다. 먼저 새로운 데이터를 생성 및 외부 데이터를 수집 하고 데이터를 통합 하거나 데이터의 형태와 소재에 관련없이 데이터를 확보 한다. 다음으로 거대한 형식의 데이터를 실시간 저장, 관리할 수 있는 분산 컴퓨팅 저장 기술이 필요하다. 저장 된 데이터의 가치를 분석하기 위해 대규모 통계, 예측 처리, 데이터 마이닝등의 분석 방법이 필요하고 머신러닝 및 인공지능을 활용한 심층 분석기술이 필요하다. 마지막으로 비전문가가 데이터 분석을 수행할 수 있게 환경을 제공하는 분석 도구 모듈을 개발하거나, 분석 결과를 표시하고 직관적인 정보를 제공하기 위해 그래픽적 기술로 구성하는 시각화가 필요하다.

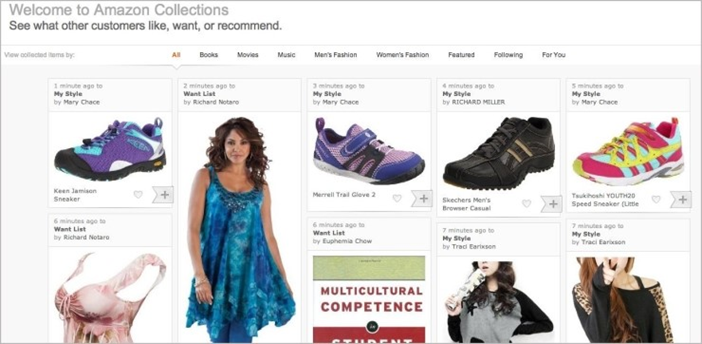
이렇게 빅데이터가 출현하고 갑자기 주목받는 배경에는 여러가지 이유가 있다. 가장 큰 영향은 스마트폰과 같은 모바일 기기의 보급과 소셜네트워크 서비스(SNS)의 유행으로 엄청난 양의 데이터가 생산되고 있다. 빅데이터는 기업,정부,교육,문화 분야에서 그 가치가 입증되고 있으며 빅데이터를 적극적으로 활용하는 기업은 이미 비즈니스적 성과를 보이며 빅데이터의 효과를 입증하고 있다.



빅데이터 안에서 소비자의 의식이나 행동을 깊이 있게 관찰하고 소비자의 스스로 자각하지 못하는 속마음, 경향, 트렌드를 꿰뚫어보아서 의미 있는 해석을 도출하는 것이 본 프로젝트에서 가장 주목하는 기술이다.

추천 시스템

추천 시스템(recommend system)은 현대 상업시장에서 가장 강력하고 필수적인 전략 중 하나이다. 사용자(User)는 모든 아이템들을 다 둘러볼 시간이 없고, 자신에게 가장 필요로 하는 상품을 추천 해 주길 원한다. 추천 시스템을 적용한 웹 쇼핑사이트들은 자신들의 알고리즘을 이용하여 사용자들에게 적합한 아이템을 추천해준다. 사용자들이 기존에 구매했던 상품들의 속성이나, 평점을 분석하여 관련 있는 상품들을 추천 리스트로 작성해 보여준다.



추천시스템을 가진 회사중 가장 유명한 회사는 단연 아마존이다. 아마존은 회원들의 소비 패턴을 분석해 구매 가능한 상품을 추천하는데, 아마존 성장의 일등 공신으로 아마존 매출의 약35%가 추천 상품에서 발생한다. 아마존은 고유의 아이템기반 협업필터링 알고리즘을 ‘A9’이라 부르고 특허를 등록하였다.

이러한 추천 시스템을 적용한 웹시스템을 개발하기 위해 빅데이터 분야의 요소를 접목시킨다. 빅데이터 수집단계를 통해 데이터를 수집하고, 분석단계를 통해 데이터를 파싱, 정규화 하며 활용단계에서 수집한 데이터들을 이용하기로 한다. 이렇게 빅데이터의 요소를 웹 사이트와 알고리즘을 구현하며 직접 학습해보기로 한다.

**본론**

개발동기

온라인 컨텐츠 시장이 성장함에 따라 소비자들은 더욱 많은 선택권을 가지게 되었다. 하지만 그만큼 모든 컨텐츠를 소모하기엔 시간적 여유가 따르지 않기 때문에 소비자들은 제한된 시간을 효율적으로 소모하기 위해 제한적으로 컨텐츠를 고를 수 밖에 없다. 대부분의 멀티미디어 서비스 웹사이트들은 이런 소비심리를 전략적으로 이용하기 위해 대부분 컨텐츠 추천 알고리즘을 사용하고 있다.

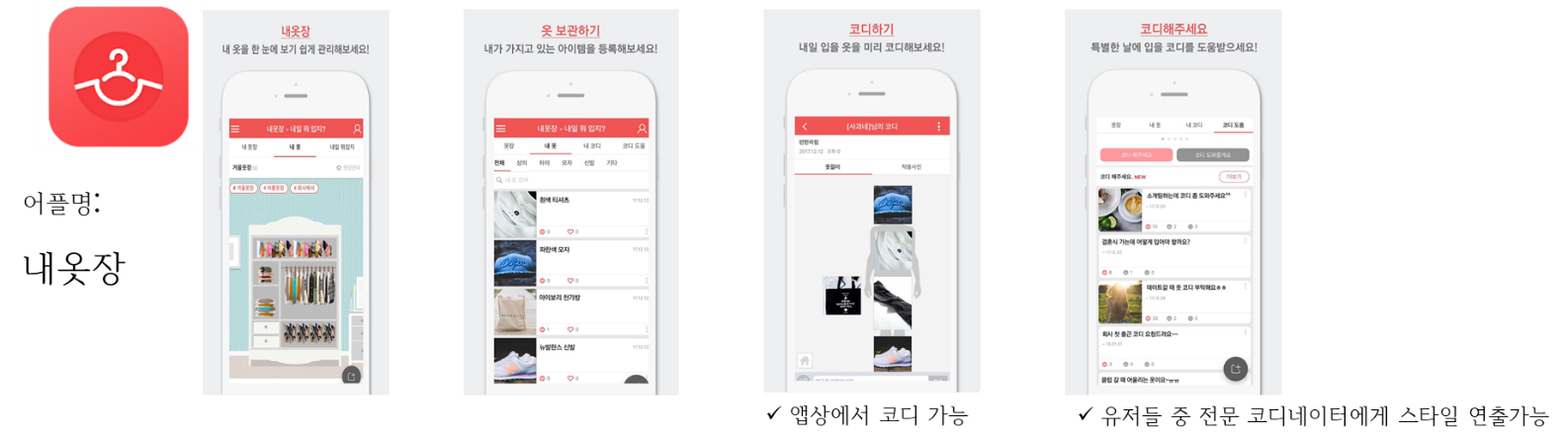
최근 가장 높은 성장률을 보이고 있는 넷플릭스(Netflix)는 대표적인 멀티미디어 컨텐츠 서비스 사이트이다. 넷플릭스는 강력한 컨텐츠 추천알고리즘을 사용하여 사용자(User)가 시청한 영상물들을 분석하여 트렌드를 파악하고, 취향에 맞는 컨텐츠를 추천해준다. 또는 다른 유저들이 많이 즐긴 컨텐츠들을 리스트로 상단에 노출시키는 마케팅을 하기도 한다

* **Netflix : 대여되는 영화의 2/3가 추천으로부터 발생**
* **Google News : 38% 이상의 조회가 추천에 의해 발생**
* **Amazon : 판매의 35% 가 추천으로부터 발생**
* **Netflix Prize (~2009) Netflix에서 주관하는 경연대회로, 영화 선호도를 가장 잘 예측하는 협업 필터링 알고리즘에 수상 (US$1,000,000)**

컨텐츠 추천 알고리즘은 소비산업에서 가장 필수적이면서 기본적인 마케팅 기법이 되었다. 미디어 산업뿐만 아니라 쇼핑, 스포츠, 부동산등 여러 분야에서 활발히 사용되고 있고 소비산업 뿐만 아니라 사용자들의 니즈를 분석하여 새로운 데이터를 수집할 수 있다는 점에서 다양하게 사용될 수 있다. 앞으로 이런 추천 알고리즘을 개량 시키고 발전시킴으로써 궁극적으로 소비자의 마음을 읽어 완벽하게 필요로 하는 요소들만 추천하는 인공지능 분야에도 사용될 수 있다.

기존 시스템 소개,분석

-내옷장(어플)



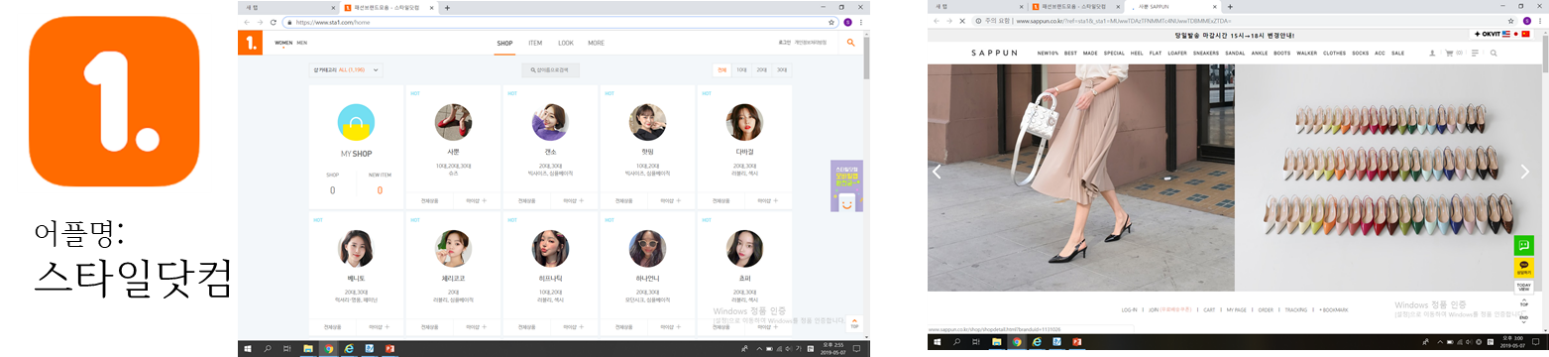
단점

1.상의 중에서도 아우터, 이너 이런 것으로 분류되는 것이 아닌 상의 옷이면 그대로 상의 옷으로 분류되어 레이어드 등 코디 불가

2.앱 자체에서의 오류로 인한 user 등의 불편 증가

<https://play.google.com/store/apps/details?id=com.cubelab.owncloset>

-스타일닷컴(웹사이트)



장점

1.유사 앱 중에서 가장 많은 의류 브랜드가 있다.

2.검색 수 기준으로 가장 브랜드 순위, 남자 쇼핑몰 인기 순위, 여자 쇼핑몰 인기 순위 등을 제공함으로써 user가 가장 대중들이 선호하는 의류 분석 가능

단점

1.자신이 선호하는 의류들을 의류의 모양 등이 아닌 색상 위주로 추천

<https://www.sta1.com/home>

개선방안

프로젝트에서 핵심으로 주목한 부분은 추천 시스템의 개선이다. 현재 개선해야 할 문제점은 크게 2가지로 볼 수 있다.

Cold Start Problem

첫 방문한 사용자는 유사한 사용자가 없기 때문에 추천이 되질 않음

->따라서, 웹페이지 첫 방문 시, 일정 개수 이상의 코디에 평점을 남기는 것을 유도

User-Rating sparsity Problem

대부분의 사용자는 평점을 잘 남기지 않는 경향이 있음

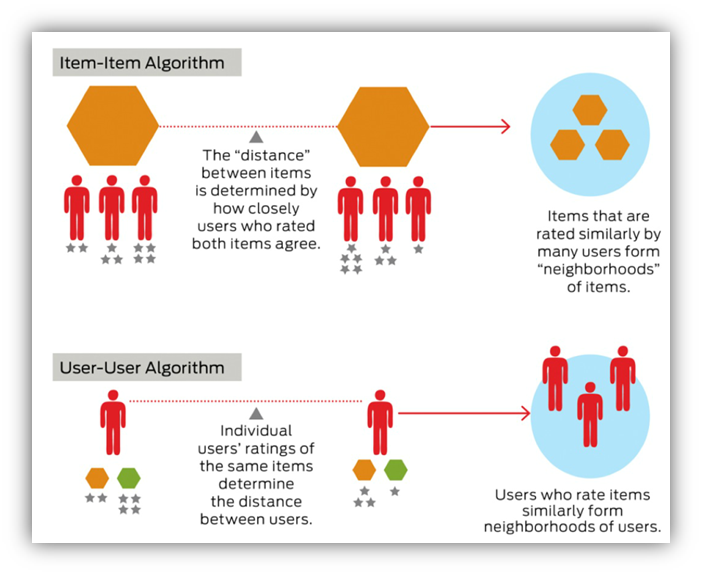
->이들에게는 개인화가 어려움, Popularity based 추천

추천 알고리즘의 단계는 3가지로 나누어 진다. 첫번째는 가장 단순 단계(Popularity)이다. 가장 쉬운 알고리즘으로, 제일 높은 인기도, 즉 평점을 갖는 아이템을 추천해준다. 사용자 모두에게 동일한 item을 추천한다. 그 다음으로 중간단계(Collaborative Filtering)는 사용자와 아이템간의 평가를 이용하여 사용자끼리의 유사도를 찾는다.특정 사용자와 유사한 사용자들이 남긴 평점을 기반으로 해당 사용자의 평점을 예측한다. 해당 사용자가 해당 아이템을 얼마나 좋아할 것인지 수치적으로 예측하는 것을 의미한다.

가장 마지막 단계는 Personalization(개인화)단계로 모두에게 동일한 아이템이 아닌 각 개인의 성향에 맞는 아이템을 추천한다. user-item평가, 협업 필터링 알고리즘, SVD등이 사용된다.

현재 가장 많이 사용되고 있는 추천 알고리즘은 협업 필터링이다. 협업 필터링은 사용자들로부터 정보를 입력 받아 관심사들을 자동으로 예측하여 추천하는 필터링 기법이다. 협업 필터링은 유저 기반 협업 필터링(User-based-CF), 아이템 기반 필터링(Item-based-CF)로 구분된다. 유저 기반 협업 필터링은 다시 능동적, 수동적 필터링으로 분류된다. 이러한 필터링 알고리즘들은 데이터를 분석하여 유사도 계산을 통해 아이템을 추천하게 된다.

유저 기반 필터링(UBCF)는 사용자들간 공통된 아이템을 기준으로 얼마나 선호도가 일치하는지를 수치화 하여 그 점수를 사용한다. 예를 들어 A가 Z라는 블록버스터 영화에 평점5점을 주고, B가 5점을 준다면 둘의 선호도도 점수의 거리는 0이라고 할 수 있다. 만약 B가 3점을 줬다면, 두 사용자의 거리는 증가하게 된다. UBCF는 이것을 수치화 하여 더 가까운 유사도를 활용한 추천 방식이라고 볼 수 있다. 그러나 UBCF는 단점을 가지고 있는데, 만약 사용자가 최초로 시스템을 이용하는 거라면, 필요한 초기 데이터가 없기 때문에 추천을 하기가 어렵고, 또한 이런 데이터를 수집하는데 많은 시간이 소요 될 수 있다. 이러한 단점을 보완하는 것이 아이템 기반 필터링 이다.



아이템 기반 필터링(IBCF)는 추천 알고리즘에 사용자 데이터 대신 평가된 아이템 객체들이 데이터로 사용된다. 오늘날 대부분의 추천 시스템은 IBCF를 사용하고 있는데, 가장 대표적인 곳이 넷플릭스와 아마존이다. 아마존에서는 사용자가 구매한 상품들과 유사한 상품들을 선호할 것이라는 예측을 기반으로 하고 있다. IBCF는 UBCF와 달리 초기에 등록한 객체들의 데이터에 의존하기 때문에, 처음 시스템을 접하는 사용자라도 원하는 상품과 유사한 상품들을 쉽게 추천 받을 수 있다. 그러나 IBCF역시 상품들과의 유사도를 고려하였지만, 사용자들간의 선호도가 전혀 고려되지 않았기 때문에 특정 고객과 선호도가 매우 다르다면 상품들간의 추천 정확도가 떨어지고, 추천 시스템의 추천 능력이 떨어질 수 있다

|  |  |  |
| --- | --- | --- |
|  | 장점 | 단점 |
| 유저 기반 필터링 | 알고리즘이 간단함, 아이템 정보 없이 추천이 가능 | 데이터가 많아질수록 연산이 복잡해짐, 신규 사용자의 데이터 공백 |
| 아이템 기반 필터링 | 아이템 자체의 정보 없이 추천 가능, 신규 사용자 추천 가능 | 데이터가 커질수록 연산이 복잡해짐, 초기 서비스 추천 정확도 떨어짐 |
| 능동적 필터링 | 신뢰성이 높음 | 편협 적인 유사도, 적은 피드백 |
| 수동적 필터링 | 능동적 필터링의 변수 제거 | 사용자 의존적 |

이외에도 최근에 인기가 증가하고 있는 능동적 필터링이나, 미래에 가장 잠재력이 있다고 여겨지는 수동적 필터링 방법들도 존재한다. 본프로젝트에서는 이러한 협업 필터링을 직접 구현하는 방식으로 추천 시스템을 개선시키기로 한다.

-개발 제목

**Fashion recommend system using BigData**

빅데이터를 이용한 패션 아이템 추천 시스템

빅데이터의 여러가지 적용분야를 접목하여 패션 아이템을 추천해주는 시스템을 개발한다는 뜻을 가지고 있다.

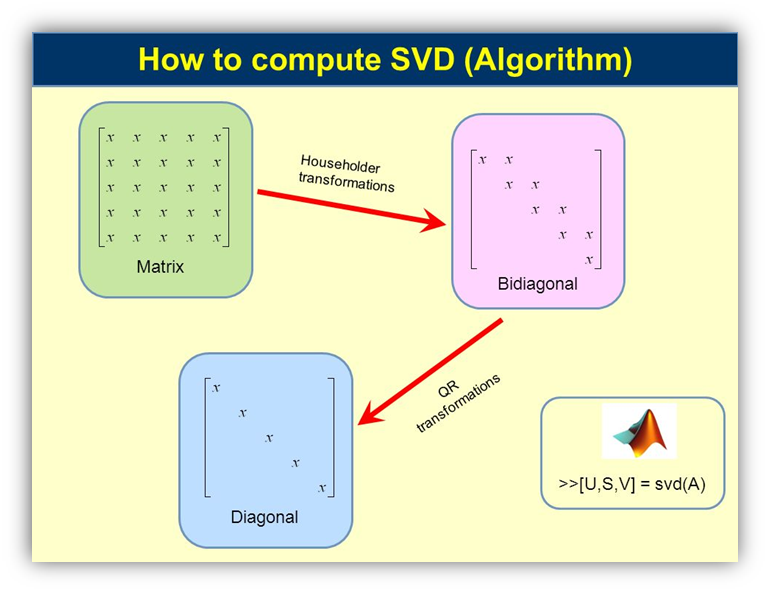
프로젝트 소개

프로젝트 세부내용

-이론

협업 필터링들의 단점들을 보완 하는 것에는 차원 축소 알고리즘이 가장 주목 받고 있다. 협업 필터링의 주목할 단점은 2가지가 존재한다. 첫번째로 거대한 데이터 즉, 빅데이터의 성질에 대해 어떻게 대응하냐이다. 거대한 데이터를 최대한 빨리 분석하고 예측 하기 위해서는 데이터를 축소할 필요가 있다. 두번째로는 결측치이다. 협업필터링을 구성하면서 매트릭스에는 사용자가 아직 평가 못한 아이템이 존재할 수 있다. 이러한 결측값들이 존재하면 데이터를 사용하여 예측하는데 오차가 발생할 수 있다.

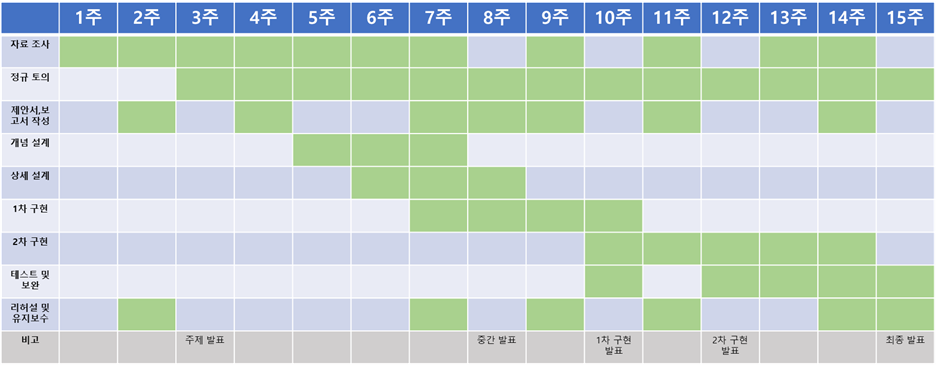
그렇다면 이러한 협업 필터링들의 단점들을 어떻게 보완할 수 있을까? 차원 축소 알고리즘들은 여러가지가 있지만 가장 대표적으로 SVD(특이값분해)라는 알고리즘을 사용하여 추천 시스템을 보완하도록 한다. SVD는 추천 시스템에 쓰이는 가장 대표적인 차원 축소 알고리즘으로, 매트릭스를 이용하여 차원 축소를 수행한다.



먼저 사용자와 상품 간의 선호도를 수많은 데이터셋으로 정리하고 매트릭스로 표현한다. 이때 SVD를 사용하여 대각 행렬을 축소하여 처리해야 하는 전체 데이터의 양을 줄일 수 있고, 아직 평가하지 않은 데이터에 대해서 평균 값을 이용하여 예상점수를 구할 수 있다. 추천 시스템은 이 점수를 이용하여 추천 점수가 높은 아이템을 사용자에게 추천하게 된다. 차원 축소 알고리즘은 다양한 상품에 대해 차원 축소를 적용함으로써 일반화 현상이 발생되고, 차원 축소가 많이 적용될수록 추천시스템은 더 효율적으로 작동한다.

FRUB프로젝트에서는 위 시스템을 프로토타입으로 구현하는 것을 목표로 한다. 사용자의 성별, 각 아이템에 대해 평가한 점수를 가지고 있는 샘플 CSV파일을 생성하고, 파이썬의 서프라이즈패키지에서 지원하는 SVD를 사용하여 예측치를 계산한다. 마지막으로 구글 웹 크롤링 오픈소스를 추가하여 해당 아이템을 검색하여 이미지를 출력한다.

-일정표

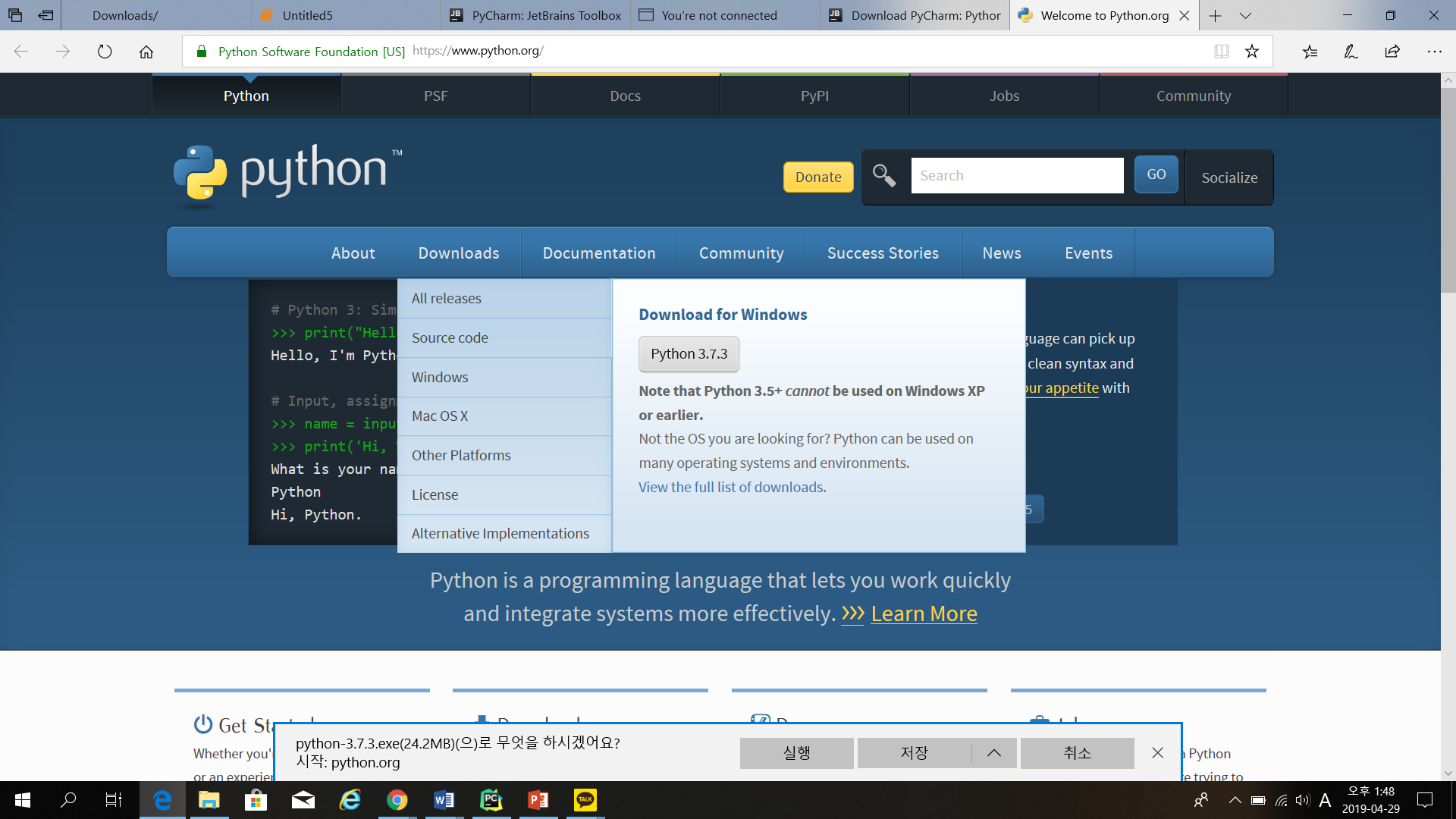


-개발 환경



파이썬 -> 플랫폼이 독립적이고 인터프리터식으로 만들어져 있으며, 객체지향적, 동적 타이핑 대화형 언어로 사용된다.데이터분석에 특화된 언어이며 ,장고를 분석해 웹사이트를 만들 때도 파이썬을 사용하기 때문에 설치를 해준다.

설치 방법은 파이썬 다운로드링크는 <https://www.python.org/> 이고,링크에 들어가

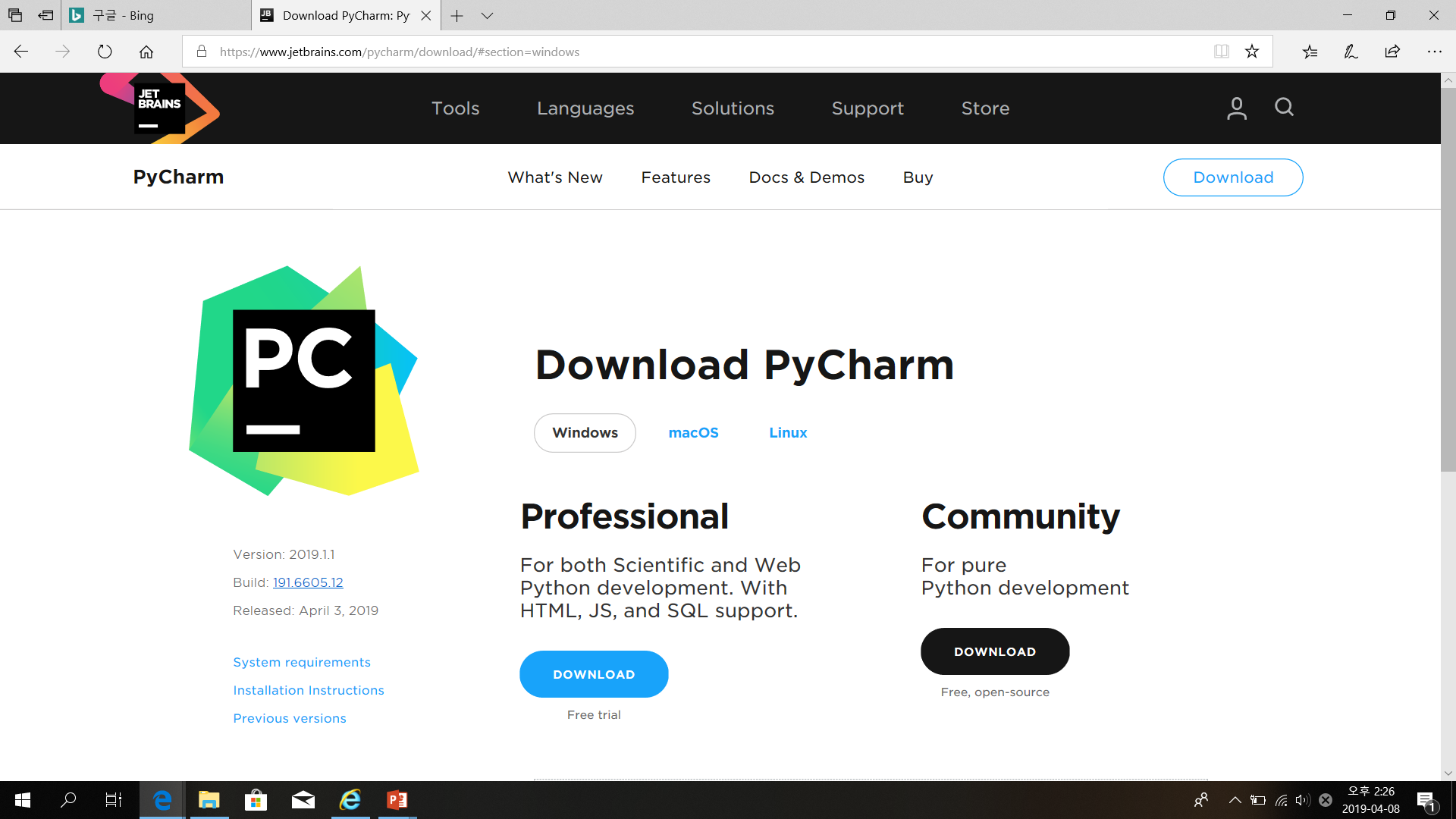


위 그림과 같은 페이지에 접속된다.접속후 downloads카테고리를 선택한후 자신의 환경을 선택하여 python 3.7.3 을 클릭하여 파이썬을 실행시켜준다.

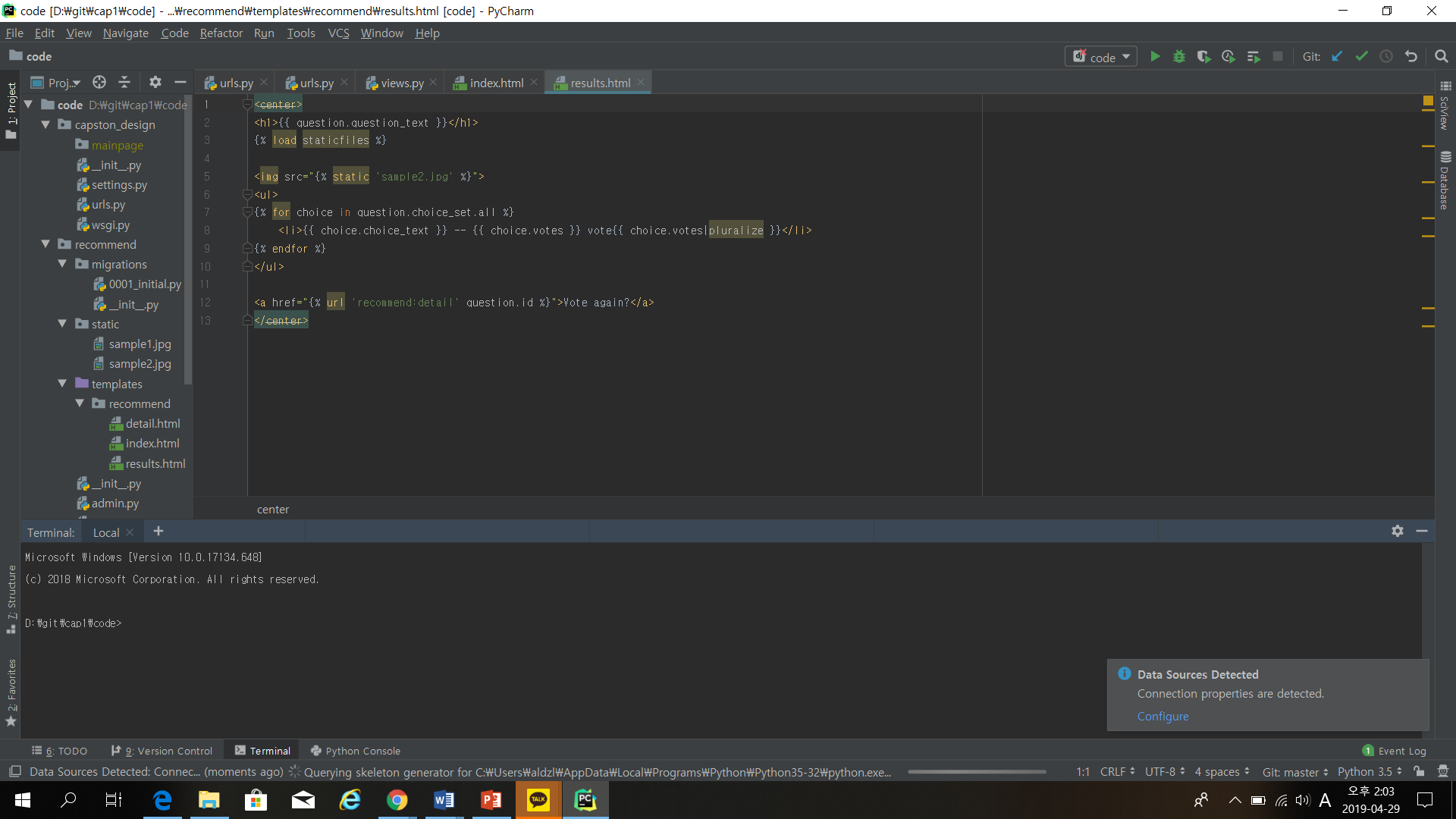
- 파이참 설치

파이참은 Python 개발에 필요한 모든 도구를 제공해 주는 역할을 한다.파이참은 다운로드시 별도의 제약 없으며 다운로드후 ->power shell 에서 장고 설치하기위해 pip install Django 를 입력하며 개발환경을 만들어준다.

다운로드 방법은 [https://www.jetbrains.com/pycharm/download/#section=windows](https://www.jetbrains.com/pycharm/download/) 링크에 접속하여 화면에 보이는Download를 클릭하여 Community Download를 눌러준다. Professional Download를 선택시 시간이 흐른 뒤 유료화 되기 때문에 community를 다운받아준다.

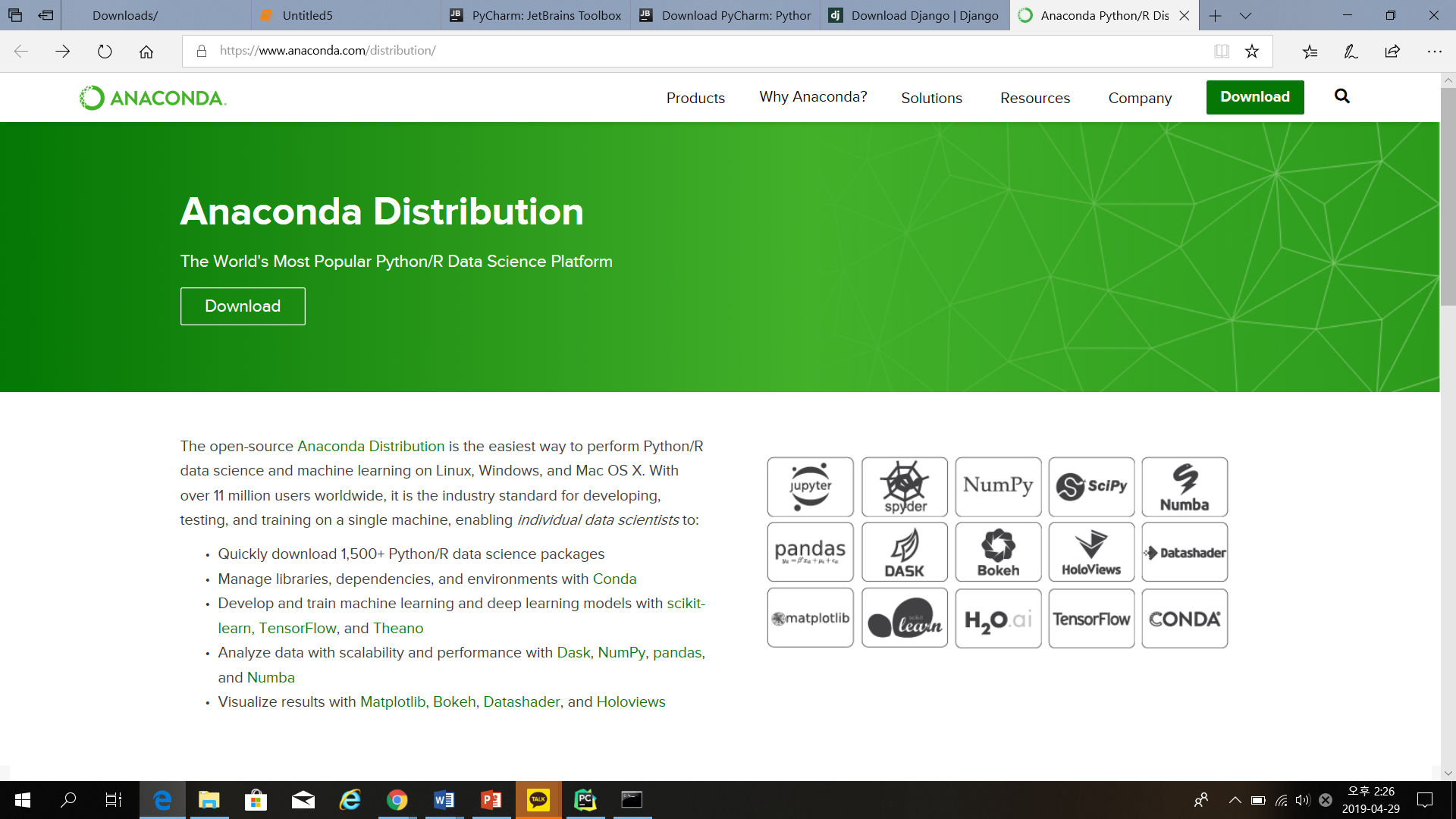


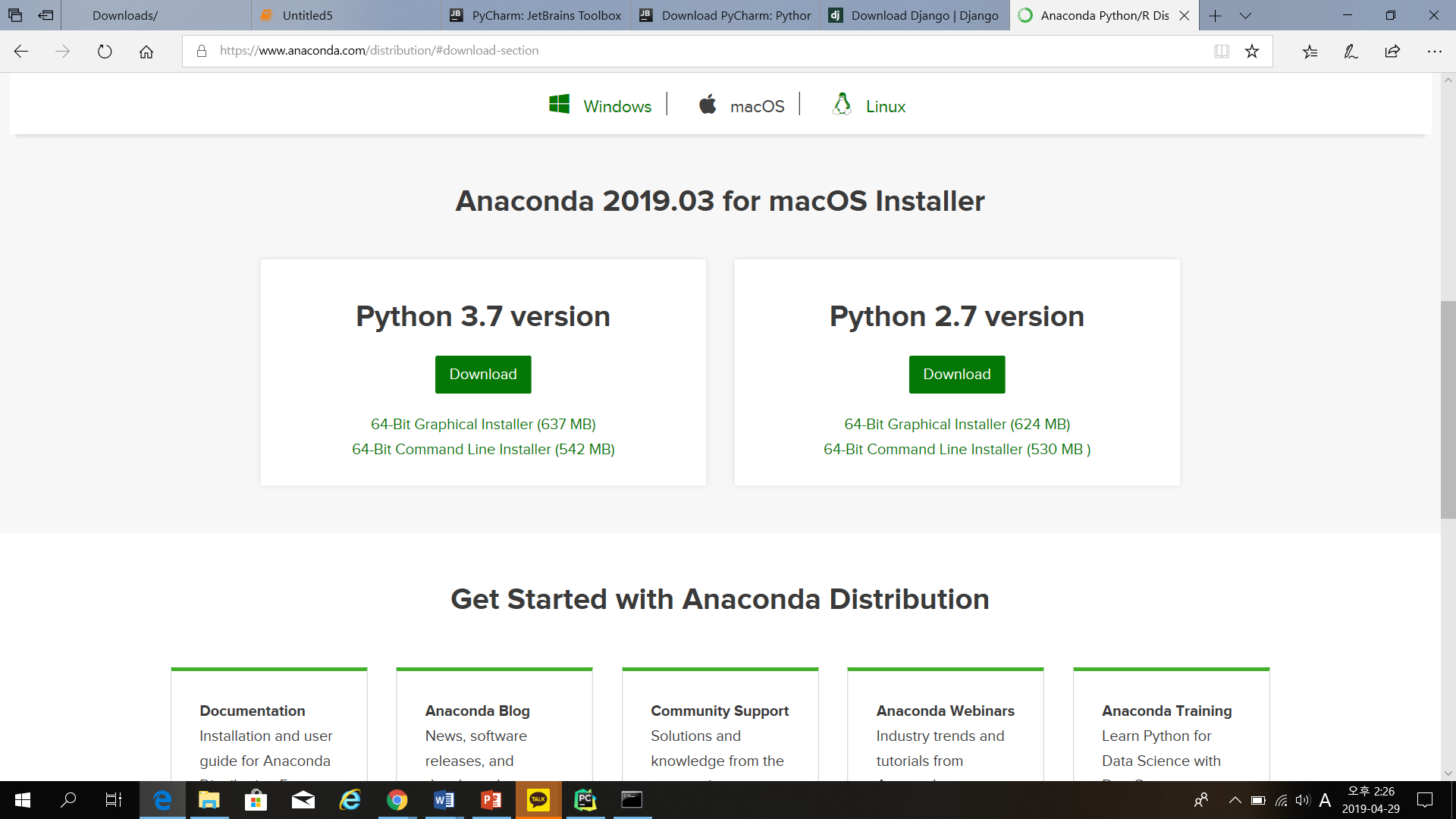
다운로드가 완료되고 실행시키면 pycharm 실행화면이 나온다.



- 아나콘다 네비게이터 ,

**Anaconda** 는 데이터 분석용으로 사용되며, 패키지관리를 단순화하고 과학 컴퓨팅을 위한 Python및 R 프로그래밍 언어를 위한 무료 및 오픈소스 배포판이다. **아나콘다 설치를 위해** [**https://www.anaconda.com/distribution/**](https://www.anaconda.com/distribution/) **링크에 클릭해주면,** 위와 같은 페이지에 접속이 된다.

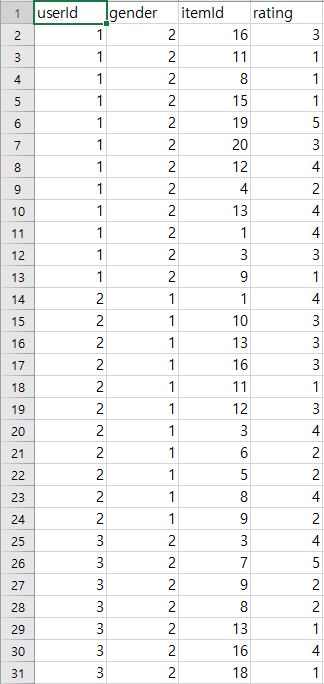
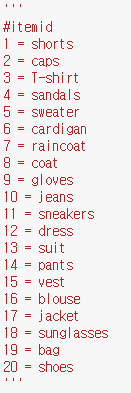


여기서 Download버튼을 클릭 하고 각자 필요한 version을 클릭하여 다운로드하여 설치해준다

-개발 현황

프로젝트를 진행하기 위해 필요한 프로그램을 설치하고 환경을 구성한다. 이후에 데이터를 수집하고 활용하기 위해 파이썬을 선택한다. 파이썬은 데이터를 분석하는데 특화된 프로그래밍 언어로 다양한 패키지를 제공하고 있다. 우선 아나콘다 네비게이터에서 제공하는 주피터 노트북으로 데이터를 관리하고 필요한 패키지를 설치해서 함수를 작성한다. 코디 사진을 수집하기 위한 웹 크롤러 패키지를 설치하고, SVD를 사용하기위해 서프라이즈 패키지를 import 한다.

-코드 해석

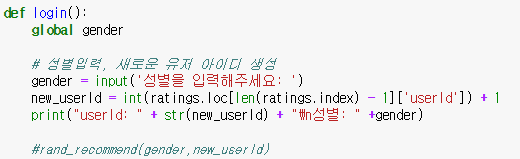


사용되는 데이터는 미리 준비한 사용자-아이템 평가 데이터셋이다. 데이터는 userId, gender, itemId, rating속성을 가지고 있고 csv형식으로 저장 되어있다. 각 아이템 아이디는 각자 정해진 정수형 코드를 가지고 있고 사용자들은 그 아이템에 대하여 평가를 최소 1점에서 최대 5점까지 줄 수 있다. 또한 사용자들의 성별이 남자일경우 1, 여자일경우 2를 가지게 된다. 프로그램이 실행됨에 따라 파일에 새로운 사용자와 아이템에 대한 평가데이터가 추가된다.

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

필요한 모듈들을 import한다. imte\_keyword변수에 기존에 정한 아이템들의 키워드들을 리스트형식으로 순서대로 저장한다. pandas에서 제공하는 read\_csv() 함수로 기존 csv파일을 rating변수에 바인딩한다.



login 함수는 새로운 사용자의 정보를 저장한다. 성별을 입력받고 기존 csv파일의 맨 아래의 인덱스의 userId의 값에 +1 하여 새로운 userId를 부여한다. (ex: 맨 마지막 인덱스의 UserID가 30이면 30+1=31을 부여한다.)

스크린샷이(가) 표시된 사진

자동 생성된 설명

rand\_recommend 함수는 무작위로 사용자에게 아이템을 추천해준다. 사용자가 원하는 추천 횟수를 입력 받는다. 위코드에서 입력했던 아이템 리스트에서 무작위로 하나를 추출하여 키워드를 만들고 검색하여 이미지 url을 사용자에게 보여준다. 사용자는 해당 링크에 대해 점수를 입력하고 평가점수를 csv파일에 새로운 레코드로 저장한다.그리고 반복문을 종료하고 루프 처음으로 돌아간다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

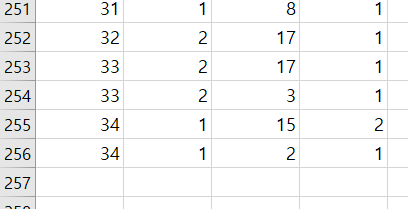
svd\_ratings함수는 svd라이브러리에서 제공하는 예측 계산 함수를 사용한다. 우선 ratings변수를 학습시키고 svd메서드에 fit시킨다.기존에 설정한 아이템리스트에서 사용자가 평가한 항목들을 빼면 사용자가 평가 안한 아이템의 리스트만 남게 된다. 그리고 그 리스트를 svd.predict함수로 예측값을 구성하고, 각각 아이템과 예측값을 key와 value로 묶어 dic자료형 변수로 생성하고 max()로 최댓값을 추출한다. 즉 ‘사용자가 평가를 아직 안내린(결측치) 아이템에 대한 예측값중에 가장 큰값을 가진 아이템 코드’가 recommend\_itemId에 담기게 된다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

최종적으로 사용자가 아직 평가를 안내린 아이템들의 예측값들 중에 가장 큰 값의 키워드를 검색한 결과를 사용자에게 추천해주게 된다.

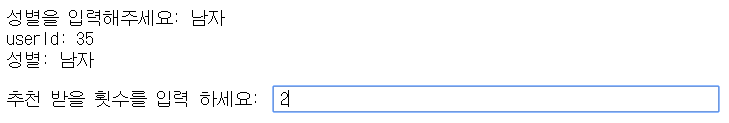
-실행 화면



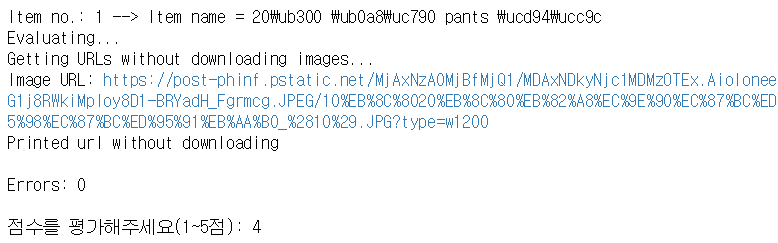
프로그램을 실행하기전 CSV파일의 마지막 레코드의 정보. 유저34가 15번과 2번 아이템에 대해 평가를 내린 점수를 보여주고 있다.

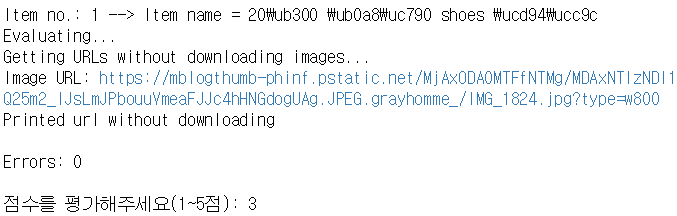


새로운 사용자를 추가해보도록 하자. 먼저 성별은 남자로 설정한다.



그러면 새로운 userID 35를 부여해주고 추천 받을 횟수를 요구한다. 2회를 입력 하게 되면 2번아이템을 무작위로 추천해준다.

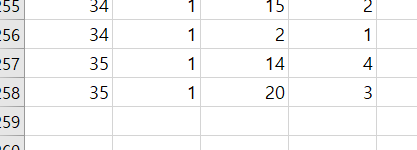




각각 pants와 shoes아이템에 대해 4점과 3점을 평가한다.



그러면 프로그램은 최종적으로 사용자의 평점을 분석해서 shorts아이템을 추천해주게 된다.



CSV파일(DB)을 확인해본 결과 정상적으로 유저35가 각 아이템에 4점과 3점 평점을 준 것이 저장 된 것을 확인할 수 있다. 이제 이 데이터를 이용하여 새로운 유저가 추가되면, 유저35가 아이템에 내린 평가가 SVD 평점 예측계산에 쓰이게 되고 또 새로운 유저가 데이터에 추가되는 식으로 빅데이터의 저장과 처리 분석이 서로 영향을 주며 작동하게 된다.

**결론**

프로젝트를 진행하며 빅데이터의 개념과 활용 순서를 적용할 수 있었다. 빅데이터는 저장, 처리, 분석, 시각화의 순서로 처리되며,각 단계에서 사용되는 기술이 있다. 추천 시스템은 빅데이터의 기술의 집합체로서 다양한 기술이 적재적소에 활용된다. FRUB프로젝트에서는 빅데이터에서 쓰이는 DB를 샘플로 생성한 csv파일로 예시를 들어 사용하고 SVD기술을 사용하여 처리하고, 결측치를 분석하였다. 또한 데이터의 이미지를 시각화 하여 사용자에게 추천했다. 기존 추천 시스템의 단점이었던 단순추천을 협업 필터링을 사용하여 사용자들의 취향을 예측하여 점수를 부여 하는 방법으로 보완했다. 개선된 시스템을 파이썬으로 프로토타입 코딩으로 작성하여 실제로 동작하는 샘플시스템을 구현해볼 수 있었다.

피드백

FRUB프로젝트에서는 사용자의 취향을 단순 평점(1~5점)으로 분석 하고 있다. 실제 상업에서 쓰이는 추천알고리즘은 더욱 다양한 속성을 사용한다. 본 프로젝트의 패션추천을 예시로 든다면 사용하는 아이템의 색상, 가격, 브랜드, 디자인등의 다양한 속성을 테이블에 추가 함으로서 추천 시스템의 알고리즘을 더 고도화 할 수 있을 것이다.

소감문

**부록**

전체코드

# 전체 코드

from google\_images\_download import google\_images\_download

from imageai.Prediction import ImagePrediction

from surprise import Reader, Dataset, SVD, evaluate

from PIL import Image

from io import BytesIO

import pandas as pd

import csv

import random

import requests

import warnings

warnings.filterwarnings(action='ignore')

item\_keyword = ['shorts','caps','T-shirt','sandals','sweater',

'cardigan','raincoat','coat','gloves','jeans',

'sneakers','dress','suit','pants','vest',

'blouse','jacket','sunglasses','bag','shoes']

ratings = pd.read\_csv('item\_rating.csv')

def login():

global gender

gender = input('성별을 입력해주세요: ')

new\_userId = int(ratings.loc[len(ratings.index) - 1]['userId']) + 1

print("userId: " + str(new\_userId) + "\n성별: " +gender)

rand\_recommend(gender,new\_userId)

def rand\_recommend(gender,new\_userId):

global ratings

r\_num = int(input('추천 받을 횟수를 입력 하세요: '))

for a in range(1,r\_num+1):

item\_Id = random.randint(1,20)

keywords = str("20대 " + gender + " " + item\_keyword[(item\_Id)-1] + " 추천")

response = google\_images\_download.googleimagesdownload()

arguments = {"keywords":keywords,"limit":1,"output\_directory":'image',"no\_download":True, "safe\_search":True}

paths = response.download(arguments)

while True:

itemscore = int(input('점수를 평가해주세요(1~5점): '))

if 1 <= itemscore and itemscore <= 5:

break

else:

print('warning: 1점에서 5점사이의 값을 입력하세요')

if gender == '남자':

new\_gender = 1

else: new\_gender = 2

new\_itemId = item\_Id

new\_rating = itemscore

ratings.loc[len(ratings.index)] = [new\_userId, new\_gender, new\_itemId, new\_rating]

ratings = ratings.set\_index("userId")

ratings.to\_csv('item\_rating.csv')

ratings = pd.read\_csv('item\_rating.csv')

svd\_ratings(ratings,new\_userId)

def svd\_ratings(ratings,new\_userId):

reader = Reader()

svd = SVD()

data = Dataset.load\_from\_df(ratings[['userId', 'itemId', 'rating']], reader)

trainset = data.build\_full\_trainset()

svd.fit(trainset)

lst\_itemId = list(range(1,21))

lst\_ratings = list(ratings[ratings.userId == new\_userId]['itemId'].values)

notrating\_itemId =list(set(lst\_itemId)-set(lst\_ratings))

rand\_itemId = random.sample(notrating\_itemId,3)

a = 0

svd\_rating = []

for i in range(1,4):

svd\_rating.append(round((svd.predict(new\_userId,rand\_itemId[a]).est), 1))

a += 1

rating\_dic = dict(zip(rand\_itemId,svd\_rating))

recommend\_itemId = max(rating\_dic, key=rating\_dic.get)

final\_recommend(recommend\_itemId)

def final\_recommend(recommend\_itemId):

keywords = str("20대 " + gender + " " + item\_keyword[(recommend\_itemId)-1] + " 추천")

response = google\_images\_download.googleimagesdownload()

arguments = {"keywords":keywords,"limit":1,"output\_directory":'image',"no\_download":True, "safe\_search":True}

print('사용자에게 추천는 아이템입니다')

paths = response.download(arguments)

'''

url = paths[keywords][0]

resp = requests.get(url)

img = Image.open(BytesIO(resp.content))

'''

if \_\_name\_\_ == '\_\_main\_\_':

login()

print('프로그램을 종료합니다')

개발환경

참조

github