Python을 활용한 데이터 수집 I

1. 목표

- Python 기본 문법 실습
- 파일 입출력에 대한 이해
- 데이터 구조에 대한 분석과 이해
- 데이터를 가공하고 JSON 형태로 저장

2. 준비사항

A. TMDB API

i. 평점 순 영화정보 API 서비스

ii. 장르 리스트 정보 API 서비스

iii. 영화 상세정보 API 서비스

B. 개발언어/프로그램

i. Python 3.7 이상

C. 필수 라이브러리

i. json

3. 요구사항

커뮤니티 서비스 개발을 위한 데이터 수집 단계로, 전체 데이터 중 필요한 데이터를 추출해 나가는 과정을 진행합니다. 아래 기술된 사항은 필수적으로 구현해야 하는 내용입니다.

A. **제공되는 영화 데이터의 주요내용 수집**

샘플 영화데이터가 주어집니다. 이중 서비스 구성에 필요한 정보만 뽑아 반환하는 함수를 완성합니다. 완성된 함수는 다음 문제의 기본기능으로 사용됩니다.

i. 데이터

1. 제공되는 movie.json 파일을 활용합니다.
2. movie.json은 '쇼생크 탈출' 영화 정보를 가지고 있습니다.

ii. 결과

1. 제공된 데이터에서 id, title, poster_path, vote_average, overview, genre_ids 키에 해당하는 정보만 가져옵니다.
2. 가져온 정보를 새로운 dictionary로 반환하는 함수 movie_info를 완성합니다.

iii. 예시

1. 입력예시

```
{
    'adult': False,
    'backdrop_path': '/avedvodAZUcwqevBfm8p4G2NziQ.jpg',
    'genre_ids': [80, 18],
    'id': 278,
    'original_language': 'en',
    'original_title': 'The Shawshank Redemption',
    'overview': '촉망받는 은행 간부 앤디 듀프레인(팀 로빈슨)은 아내와 그녀의.
                정부를 살해했다는 누명을 쓴다. ... 앤디는 이 돈을 세탁하여
                불려주면서 그의 돈을 관리하는데...',
    'popularity': 58.351,
    'poster_path': '/3hO6DIGRBaJQj2NLEYBMwpcz88D.jpg',
    'release_date': '1995-01-28',
    'title': '쇼생크 탈출',
    'video': False,
    'vote_average': 8.7,
    'vote_count': 16735
}
```

2. 출력예시

```
{
    'genre_ids': [80, 18],
    'id': 278,
    'overview': '촉망받는 은행 간부 앤디 듀프레인(팀 로빈슨)은 아내와 그녀의.
                정부를 살해했다는 누명을 쓴다. ... 앤디는 이 돈을 세탁하여
                불려주면서 그의 돈을 관리하는데...',
    'poster_path': '/3hO6DIGRBaJQj2NLEYBMwpcz88D.jpg',
    'title': '쇼생크 탈출',
    'vote_average': 8.7
}
```

B. 제공되는 영화 데이터의 주요내용 수정

이전단계에서 만들었던 데이터 중 genre_ids를 genre_names로 바꿔 반환하는 함수를 완성합니다. 완성된 함수는 다음 문제의 기본기능으로 사용됩니다.

i. 데이터

1. 제공되는 movie.json, genres.json 파일을 활용합니다.
2. movie.json은 '쇼생크 탈출' 영화 정보를 가지고 있습니다.
3. genres.json은 장르의 id, name 정보를 가지고 있습니다.

ii. 결과

1. 제공된 데이터에서 id, title, poster_path, vote_average, overview, genre_ids 키에 해당하는 정보만 가져옵니다.
2. genres.json파일을 이용하여 genre_ids를 genre_names로 변환하여 dictionary에 추가합니다.
3. 완성된 dictionary를 반환하는 함수 movie_info를 완성합니다.

iii. 예시

1. 입력예시

```
{
    'adult': False,
    'backdrop_path': '/avedvodAZUcwqevBfm8p4G2NziQ.jpg',
    'genre_ids': [80, 18],
    'id': 278,
    'original_language': 'en',
    'original_title': 'The Shawshank Redemption',
    'overview': '촉망받는 은행 간부 앤디 듀프레인(팀 로빈슨)은 아내와 그녀의.
                정부를 살해했다는 누명을 쓴다. ... 앤디는 이 돈을 세탁하여
                불려주면서 그의 돈을 관리하는데...',
    'popularity': 58.351,
    'poster_path': '/3hO6DIGRBaJQj2NLEYBMwpcz88D.jpg',
    'release_date': '1995-01-28',
    'title': '쇼생크 탈출',
    'video': False,
    'vote_average': 8.7,
    'vote_count': 16735
}
```

2. 출력예시

```
{
    'genre_names': ['Crime', 'Drama'],
    'id': 278,
    'overview': '촉망받는 은행 간부 앤디 듀프레인(팀 로빈슨)은 아내와 그녀의.
                정부를 살해했다는 누명을 쓴다. ... 앤디는 이 돈을 세탁하여
                불려주면서 그의 돈을 관리하는데...',
    'poster_path': '/3hO6DIGRBaJQj2NLEYBMwpcz88D.jpg',
    'title': '쇼생크 탈출',
    'vote_average': 8.7
}
```

C. 다중 데이터 분석 및 수정

TMDB기준 평점이 높은 20개의 영화데이터가 주어집니다. 이 중 서비스 구성에 필요한 정보만 뽑아 반환하는 함수를 완성합니다. 완성된 함수는 향후 커뮤니티 서비스에서 제공되는 영화 목록을 제공하기 위한 기능으로 사용됩니다

i. 데이터

1. 제공되는 movies.json, genres.json 파일을 사용합니다.
2. movies.json은 영화 전체 정보를 가지고 있습니다.
3. genres.json은 장르의 id, name 정보를 가지고 있습니다.

ii. 결과

1. 이전 단계의 함수 구조를 재사용합니다.
2. 영화 전체 정보를 수정하여 반환하는 함수 movie_info를 완성합니다.

iii. 예시

1. 출력예시

```
[
  {
    'genre_names': ['Crime', 'Drama'],
    'id': 278,
    'overview': '촉망받는 은행 간부 앤디 듀프레인(팀 로빈슨)은 아내와 그녀의.
                정부를 살해했다는 누명을 쓴다... 앤디는 이 돈을 세탁하여
                불려주면서 그의 돈을 관리하는데...',
    'poster_path': '/3hO6DIGRBaJQj2NLEYBMwpcz88D.jpg',
    'title': '쇼생크 탈출',
    'vote_average': 8.7
  },
  # 이하생략
]
```

D. 알고리즘을 통한 데이터 출력

세부적인 영화 정보 중 수익 정보를 이용하여 모든 영화 중 가장 높은 수익을 낸 영화를 출력하는 알고리즘을 작성합니다. 해당 데이터는 향후 커뮤니티 서비스에서 메인 페이지 기본정보로 사용됩니다.

i. 데이터

1. movies.json과 movies폴더 내부의 파일들을 사용합니다.
2. movies.json은 영화 전체 데이터를 가지고 있습니다.
3. movies 폴더 내부의 파일들은 각 영화의 상세정보를 가지고 있습니다.
4. movies 폴더의 파일의 이름은 영화의 id로 구성되어있습니다.
   아래는 13번 id를 가지고 있는 영화의 상세정보입니다.

```
// 13.json
{
  "adult": false,
  "backdrop_path": "/7c9UVPPiTPltouxRVY6N9uugaVA.jpg",
  "belongs_to_collection": null,
  "budget": 55000000,
  "genres": [
    {
      "id": 35,
      "name": "코미디"
    },
```

5. 수익정보는 상세정보 파일을 통해 확인 할 수 있습니다.

ii. 결과

1. 수익이 가장 높은 영화의 제목을 출력하는 함수 max_revenue를 완성합니다.

E. 수집데이터 저장

C단계에서 수집한 정보들을 json형식으로 저장합니다.

i. 결과

1. problem_c.py 파일을 수정합니다.
2. 각 함수들이 반환한 값을 json파일로 저장하는 코드를 작성합니다.
3. json파일의 이름은 각 문제의 이름과 같습니다.

4. 결과

반드시 활용하였던 데이터 정보 정리 및 저장한 파일에 대한 설명과 학습 내용을 README.md에 기록하여 제출합니다.
위에 명시된 사항은 최소 조건이며 추가적인 정보를 수정할 수 있습니다.

Gitlab에 제출하는 파일/폴더의 구조는 아래와 같습니다.

```
pjt01/
        data/
        README.md
        problem_a.py
        problem_b.py
        problem_c.py
        problem_c.json
        problem_d.py
```