Collaborative Filtering, Pearson correlation coefficient 를 통한 영화 장르 rating 예측 및 유저 추천 시스템

팀 명 고상원 | 팀 원 고상원 | 지도교수 이 슬 | 멘 토 이 슬

개발 동기 및 목적

현재 Netflix나 YouTube 등 영상 컨텐츠를 제공하는 여러 플랫폼에서 사용자가 시청할만한 영상을 추천해주는 시스템을 실생활에서 잘 관찰할 수 있다.

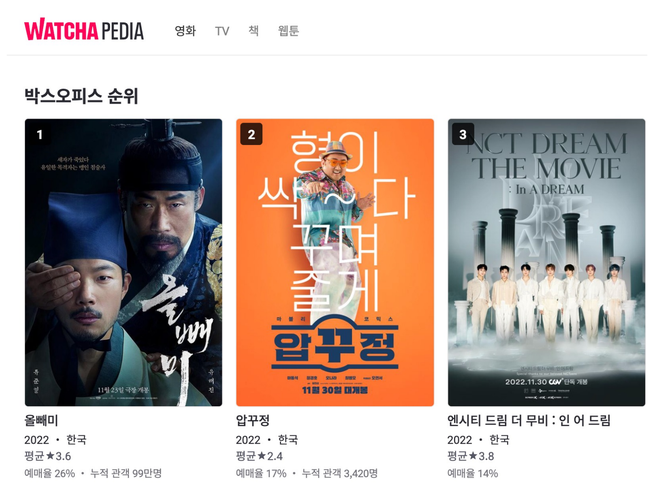
그러나 자신이 보는 영상에 대한 성향이 비슷한 유저를 매칭시켜주는 시스템은 존재하지 않는다.

따라서 이 점에 착안하여 WATCHA PEDIA에서 약 2800명의 유저 정보를 얻은 후, 그 정보를 바탕으로 1:N의 추천 시스템을 구현한다.

추가로 유저가 아직 보지 않은 영화 장르에 대해 예상되는 평점을 계산하는 시스템도 함께 구현한다.

따라서 유저가 이 시스템을 활용하면 자신이 보지 않은 장르를 보았을 때의 기대치와 자신과 비슷한 사람이 보는 영화에는 어떤 것들이 있을 지 알 수 있을 것으로 본다.

주요기술

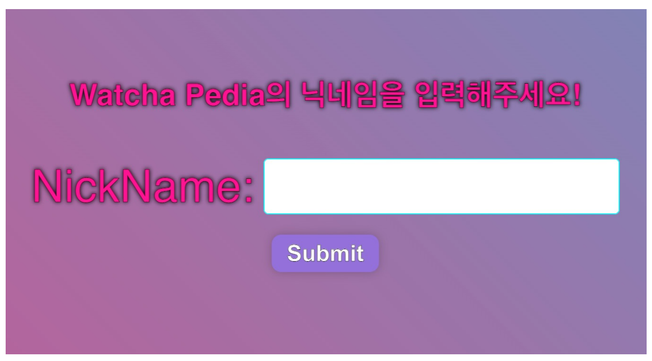


WATCH PEDIA user crawling

```
def user_similarity(user1, user2):
    both_rated = {}
    for item in df.keys():
        try:
            if df.loc[user1][item] != 'NaN' and df.loc[use
                both_rated[item] = [df.loc[user1][item], d
        except:
            continue

    number_of_ratings = len(both_rated)
    if number_of_ratings == 0:
```

weighted_average 및 similarity 계산



Flask를 통한 웹페이지 구현

개발 내용

WATCHA PEDIA에 있는 유저들을 crawl한다. 그와 동시에 유저들이 실제로 본 영화들의 정보 또한 crawl한다.

이후, 유저가 시스템을 동작시키면 crawl한 data를 바탕으로 찾고자 하는 유저가 평가하지 않은 장르에 대해 평점 예측을 실시하며 입력된 유저와 유사한 유저 5명을 구한다.

진행 방법은 다음과 같다.
1) 선택된 유저와 다른 모든 유저간의 pair를 만든다.
2) 두 유저 사이의 similarity를 측정한다.
3) Similarity가 높게 나온 상위 50명을 기준으로weighted_average를 계산하여 NaN으로 되어있는 장르의 평점을 예측한다.
4) 유저에게 상위 5명의 유사한 취향을 가지고 있는 유저와 시청하지 않은 장르에 대해 예측이 완료된 평점을 제공한다.

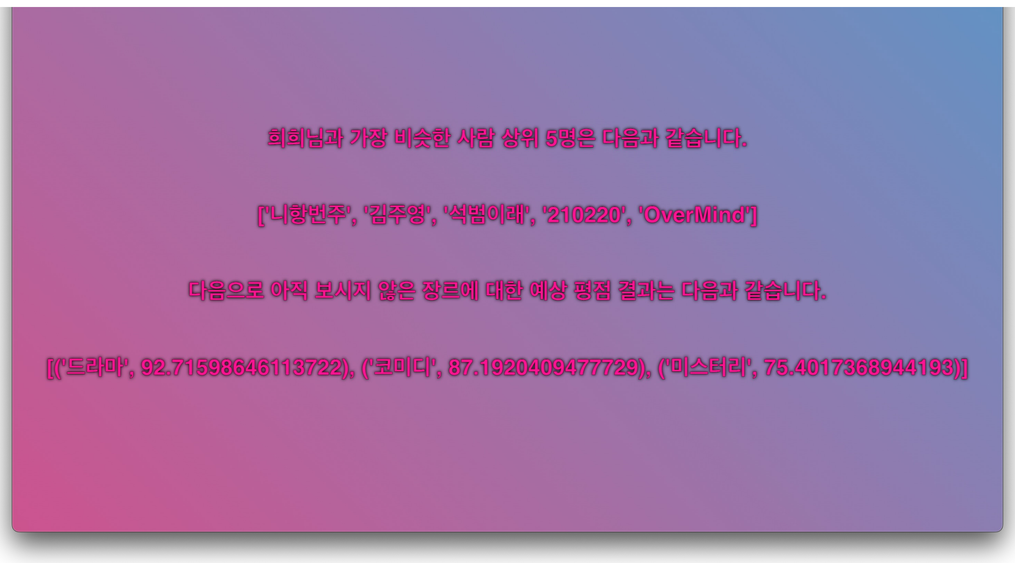
```
@app.route("/result", methods=['GET'])
def result():
    global similarity
    similarity = {}

    user1 = request.args.get('name')

    if user1 not in df.index:
        return "찾고자 하는 사람이 DB에 존재하지 않습니다."
    for user2 in tqdm.tqdm(df.index):
        if user1 != user2:
            similarity[user2] = user_similarity(user1, user2)
    similarity = sorted(similarity.items(), key=itemgetter(1), reverse=True)
    similarity_n = []
    for i in range(5):
        similarity_n.append(similarity[i][0])

    unseen_genres = get_unseen_genres(user1)
    result = get_weighted_average(unseen_genres, n)
    result = sorted(result.items(), key=itemgetter(1), reverse=True)
    return render_template("res.html", similarity = similarity_n, name = user1, result = result, n = n)
```

결과 및 분석



다음은 임의의 유저 '희희'와 가장 유사한 유저 5명, 보지 않은 장르에 대한 예상 평점이 출력된 결과이다.

현재는 유저의 입력이 들어옴과 동시에 실시간으로 similarity를 계산하여 시간이 걸리게 되는데, 사전 작업을 거치면 속도를 높일 수 있을 것으로 예상된다.