


2024 Fall
20190741 김지수



인공지능과 마케팅 코딩 팀 과제 7
: Recommendation System using movie datasets

Preparing the Dataset

Project 기초 정보

1) Datasets: <https://www.kaggle.com/datasets/tmdb/tmdb-movie-metadata>

- 5,000여개의 영화 데이터로 제목, 출연 배우, 장르, 키워드, 언어, 인기도, 간단한 개요(Overview) 등의 정보를 포함
- The movie Database(TMDb)에서 수집된 영화 정보를 포함하며 Kaggle이 제공하는 데이터셋

2) 최신 딥 러닝 기반 추천 시스템 이전에 전통적인 추천 시스템 모델을 활용하여 프로젝트를 진행

Coding 과제 7

추천시스템에 적용할 수 있는 자료(구매자료, 영화선택, 기사선택)를 구해서 추천시스템에 적용해보시오.

Preparing the Dataset

성검학원의 마검사와 비슷한 콘텐츠 >



드라마



오늘 대한민국의 TOP 10 영화



Recommendation Systems

1) Content-Based Filtering (콘텐츠 기반 필터링)

- 추천 대상의 Domain 특징을 활용하는 방법
- 사용자가 관심 있는 아이템의 속성을 분석해 새로운 아이템을 추천함
- 다른 유저의 정보는 사용되지 않음
- 추천 대상의 특징(Feature)을 추출하기 위한 TF-IDF나 Word2Vec과 같은 특징 추출 방법론이 사용됨

Ex) 내가 시청한 영화와 비슷한 특징을 가진 영화를 추천

2) Collaborative Filtering (협업 필터링)

- 유저-아이템의 관계로부터 도출
- 최근접 이웃 기반, 잠재 요인 협업 필터링 방법 등
- 유저 간의 선호도나 구매 이력을 비교하여 비슷한 케이스를 찾아 추천하는 User-based 방법
- 아이템을 구매/사용한 유저 목록을 비교하여 추천하는 Item-based 방법
- User-Item 관계를 머신 러닝이나 딥러닝 모델을 사용하여 학습하기도 한다

3) Hybrid Filtering (하이브리드 필터링)

- 위의 두 방법을 같이 활용하여 단점을 보완한 방법

Preparing the Dataset

Recommendation Systems

4) Demographic Filtering

- 가장 간단하지만 효과적인 방법일 수 있다.
- 모든 사용자에게 제공되는 일반적인 추천으로 인기도, 장르 등에 기반
- Top 10 Movies in U.S Today, Top 10 TV shows in Korea Today, etc..

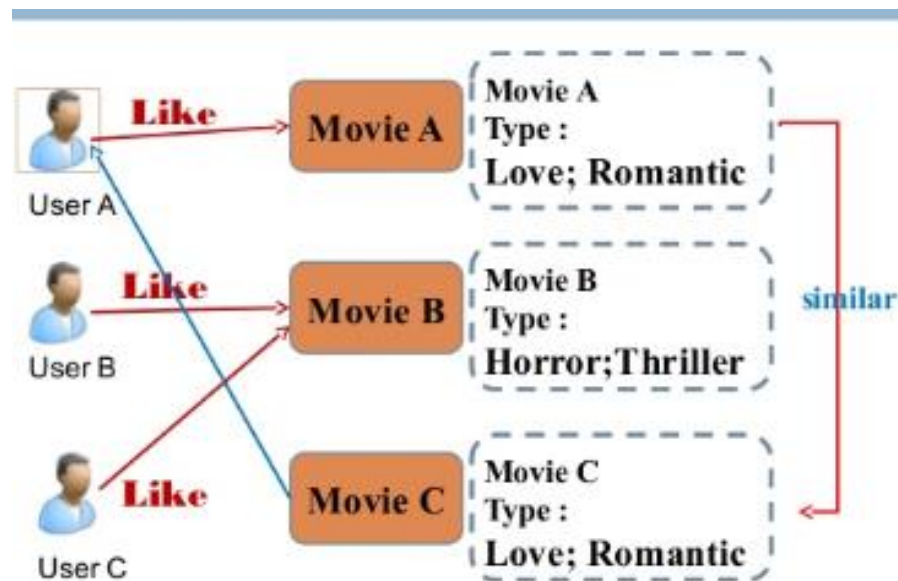
오늘 대한민국의 TOP 10 영화



Preparing the Dataset

Content-based Filtering

- 특정 아이템에 기반하여 유사한 아이템을 추천
- 아이템에 대한 정보인 메타 데이터 (장르, 감독, 배우, 설명 등)를 활용
- 다니엘 크레이그 주연의 액션 영화를 사용자가 좋아했다면 해당 배우의 다른 액션 영화도 좋아할 것이라는 생각



Preparing the Dataset

Content-based Filtering

- 사용된 데이터셋의 메타데이터 (주요 데이터를 중심으로 정리)

Feature	설명	Feature	설명
movie_id	각 영화의 고유 번호(id)	title	영화의 제목
cast	주연 및 조연 배우 목록	overview	개요 (간단한 영화 설명)
crew	영화 감독, 편집자, 작가 등	popularity	영화의 인기도 (수치)
budget	제작 예산	production_comp	제작 회사
genre	영화 장르 (Action, Comedy 등)	release_date	출시일
homepage	영화 홈페이지	vote_count	평점 리뷰 개수
keyword	영화 관련 키워드	vote_average	평점 평균

Preparing the Dataset

Import library

```
import pandas as pd
import numpy as np
df1=pd.read_csv('../input/tmdb-movie-metadata/tmdb_5000_credits.csv')
df2=pd.read_csv('../input/tmdb-movie-metadata/tmdb_5000_movies.csv')
```

```
df1.columns = ['id', 'tittle', 'cast', 'crew']
df2= df2.merge(df1,on='id')
df2.head(5)
```

*일부 속성은 잘림

	budget	genres	homepage	id	keywords	original_language	original_title	overview	popularity	production_companies	production_countries	release_date	revenue	runtime
0	237000000	[[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://www.avatarmovie.com/	19995	[[{"id": 1463, "name": "culture clash"}, {"id": ...	en	Avatar	In the 22nd century, a paraplegic Marine is di...	150.437577	[[{"name": "Ingenious Film Partners", "id": 289...	[[{"iso_3166_1": "US", "name": "United States o...	2009-12-10	2787965087	162.0
1	300000000	[[{"id": 12, "name": "Adventure"}, {"id": 14, "...	http://disney.go.com/disneypictures/pirates/	285	[[{"id": 270, "name": "ocean"}, {"id": 726, "na...	en	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	139.082615	[[{"name": "Walt Disney Pictures", "id": 2}, {""...	[[{"iso_3166_1": "US", "name": "United States o...	2007-05-19	961000000	169.0

Plot description-based Recommendation model

- 1) 영화의 줄거리 설명(Overview) 텍스트를 기반으로 영화 간 유사성을 계산하여 추천하는 모델
- 2) 텍스트 데이터 처리
 - 줄거리 설명을 Word Vector(단어 벡터)로 변환
 - TF-IDF(Term Frequency-Inverse Document Frequency) 방법을 사용
 - TF: 특정 단어가 문서에서 등장하는 비율, IDF: 특정 단어가 전체 문서에서 얼마나 흔하지 않은지를 측정
 - $TF-IDF = TF * IDF$ 로 단어의 중요도를 평가
- 3) 유사성 점수를 계산
 - 코사인 유사도를 사용 (두 벡터의 방향을 비교), Scikit-learn의 TfidfVectorizer를 사용하면 간단하게 구현 가능
- 4) 결과적으로 줄거리 텍스트 데이터를 사용하여 비슷한 내용의 영화를 추천하는 방식

Preparing the Dataset

Plot description-based Recommendation model

```
df2['overview'].head(5)
```

```
0 In the 22nd century, a paraplegic Marine is di...
1 Captain Barbossa, long believed to be dead, ha...
2 A cryptic message from Bond's past sends him o...
3 Following the death of District Attorney Harve...
4 John Carter is a war-weary, former military ca...
Name: overview, dtype: object
```

007 스카이폴
영화 · SKYFALL · 2012

전체 기본정보 감독/출연 관람평 무비클럽 포토 리뷰 명대사 >

개봉	2012.10.26.	
등급	15세 이상 관람가	
장르	액션	
국가	영국, 미국	
러닝타임	143분	
배급	소니 픽처스 릴리징 브에나 비스타 영화㈜	
원작	소설	

[바로보기](#) ♡ 2,178

정보오류 수정요청

네이버 영화 ⓘ [다른 사이트 더보기](#)

소개

상관 M의 지시에 따라 현장 요원 이브와 함께 임무를 수행하던 제임스 본드는 달리는 열차 위에서 적과 치열한 결투를 벌이다 M의 명령으로 이브가 쏜 총에 맞고 추락하여 실종된다. 이에 임무가 실패로 끝나자 전세계에서 테러단체에 잠입해 임무를 수행 중이던 비밀요원들의 정보가 분실되고 MI6는 사상 최대의 위기에 빠진다. 설상가상으로 M의 과거에 얽힌 비밀로 인해 미스터리한 적 '실바'에게 공격을 받은 MI6는 붕괴 위험에 처하게 되고, 이 사건으로 인해 M은 책임 추궁을 당하며 퇴출 위기에 놓인다. 이때, 죽음의 고비에서 부활한 제임스 본드가 M의 걸어로 다시 돌아온다. 절체절명의 위기에 놓인 MI6와 M을 구하기 위해 제임스 본드는 비밀스러운 여인 서버린을 통해 '실바'를 찾아간다. 그리고 마침내 사상 최강의 적 '실바'와 피할 수 없는 대결을 시작하게 되는데...

Preparing the Dataset

Plot description-based Recommendation model

```
#Import TfidfVectorizer from scikit-learn
from sklearn.feature_extraction.text import TfidfVectorizer

tfidf = TfidfVectorizer(stop_words='english')
df2['overview'] = df2['overview'].fillna('')
tfidf_matrix = tfidf.fit_transform(df2['overview'])
tfidf_matrix.shape
```

(4803, 20978)

- 20,000여개의 다른 단어들이 4800여개의 영화 줄거리 설명에 사용되었음
- 코사인 유사도 방법을 사용하여 줄거리를 기준으로 영화의 유사도를 계산

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}},$$

Preparing the Dataset

Plot description-based Recommendation model

```
from sklearn.metrics.pairwise import linear_kernel

# Compute the cosine similarity matrix
cosine_sim = linear_kernel(tfidf_matrix, tfidf_matrix)
indices = pd.Series(df2.index, index=df2['title']).drop_duplicates()

# Function that takes in movie title as input and outputs most similar movies
def get_recommendations(title, cosine_sim=cosine_sim):

    idx = indices[title]
    sim_scores = list(enumerate(cosine_sim[idx]))
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
    sim_scores = sim_scores[1:11]
    movie_indices = [i[0] for i in sim_scores]
    return df2['title'].iloc[movie_indices]
```

Preparing the Dataset

Plot description-based Recommendation model

```
get_recommendations('The Dark Knight Rises')
```

```
65          The Dark Knight
299         Batman Forever
428         Batman Returns
1359                Batman
3854  Batman: The Dark Knight Returns, Part 2
119         Batman Begins
2507                Slow Burn
9          Batman v Superman: Dawn of Justice
1181                JFK
210         Batman & Robin
Name: title, dtype: object
```

```
get_recommendations('The Avengers')
```

```
7          Avengers: Age of Ultron
3144                Plastic
1715                Timecop
4124                This Thing of Ours
3311                Thank You for Smoking
3033                The Corruptor
588   Wall Street: Money Never Sleeps
2136                Team America: World Police
1468                The Fountain
1286                Snowpiercer
Name: title, dtype: object
```


Preparing the Dataset

Credits, Genres and Keywords based Recommendation model

- 1) 더 나은 메타 데이터 사용을 통한 추천 시스템의 품질 향상
- 2) Credits, Genres, Keywords를 추천 시스템에 활용
 - 상위 3명의 배우, 감독, 키워드를 추출
 - 문자열로 변환된 리스트 형태이므로 사용할 수 있는 구조로 변환

```
# Parse the stringified features into their corresponding python objects  
from ast import literal_eval
```

```
features = ['cast', 'crew', 'keywords', 'genres']  
for feature in features:  
    df2[feature] = df2[feature].apply(literal_eval)
```

Preparing the Dataset

Credits, Genres and Keywords based Recommendation model

- 각 특성에서 필요한 정보를 추출하는 함수를 작성

```
def get_director(x):
    for i in x:
        if i['job'] == 'Director':
            return i['name']
    return np.nan

def get_list(x):
    if isinstance(x, list):
        names = [i['name'] for i in x]
        if len(names) > 3:
            names = names[:3]
        return names

    return []
```

Preparing the Dataset

Credits, Genres and Keywords based Recommendation model

```
df2['director'] = df2['crew'].apply(get_director)

features = ['cast', 'keywords', 'genres']
for feature in features:
    df2[feature] = df2[feature].apply(get_list)
df2[['title', 'cast', 'director', 'keywords', 'genres']].head(3)
```

	title	cast	director	keywords	genres
0	Avatar	[Sam Worthington, Zoe Saldana, Sigourney Weaver]	James Cameron	[culture clash, future, space war]	[Action, Adventure, Fantasy]
1	Pirates of the Caribbean: At World's End	[Johnny Depp, Orlando Bloom, Keira Knightley]	Gore Verbinski	[ocean, drug abuse, exotic island]	[Adventure, Fantasy, Action]
2	Spectre	[Daniel Craig, Christoph Waltz, Léa Seydoux]	Sam Mendes	[spy, based on novel, secret agent]	[Action, Adventure, Crime]

Preparing the Dataset

Credits, Genres and Keywords based Recommendation model

```
# Convert all strings to lower case and strip names of spaces
def clean_data(x):
    if isinstance(x, list):
        return [str.lower(i.replace(" ", "")) for i in x]
    else:
        #Check if director exists. If not, return empty string
        if isinstance(x, str):
            return str.lower(x.replace(" ", ""))
        else:
            return ''

features = ['cast', 'keywords', 'director', 'genres']

for feature in features:
    df2[feature] = df2[feature].apply(clean_data)
```

- Johnny Depp과 Johnny Galecki가 있을 때 서로 다른 사람으로 확인하도록 스페이스를 지우고 소문자로 변환

Preparing the Dataset

Credits, Genres and Keywords based Recommendation model

```
from sklearn.feature_extraction.text import CountVectorizer
```

```
count = CountVectorizer(stop_words='english')  
count_matrix = count.fit_transform(df2['soup'])
```

```
from sklearn.metrics.pairwise import cosine_similarity
```

```
cosine_sim2 = cosine_similarity(count_matrix, count_matrix)  
df2 = df2.reset_index()  
indices = pd.Series(df2.index, index=df2['title'])
```

- CountVectorizer를 사용하여 텍스트 데이터를 수치화
: 단어의 등장 횟수를 기반으로 단어 벡터를 생성
- stop_words='english': a, the 등을 제거
- Count.fit_transform()으로 텍스트 데이터를 수치화하여 희소 행렬로 변환
- Cosine_similarity로 두 벡터 간의 유사성을 측정
- 영화 제목을 기반으로 인덱스를 생성하여 제목으로 검색 가능하도록 설정

Preparing the Dataset

```
get_recommendations('The Dark Knight Rises', cosine_sim2)
```

```
65          The Dark Knight
119         Batman Begins
4638  Amidst the Devil's Wings
1196         The Prestige
3073      Romeo Is Bleeding
3326      Black November
1503          Takers
1986          Faster
303          Catwoman
747      Gangster Squad
Name: title, dtype: object
```

```
get_recommendations('The Godfather', cosine_sim2)
```

```
867      The Godfather: Part III
2731      The Godfather: Part II
4638  Amidst the Devil's Wings
2649      The Son of No One
1525      Apocalypse Now
1018      The Cotton Club
1170  The Talented Mr. Ripley
1209      The Rainmaker
1394      Donnie Brasco
1850      Scarface
Name: title, dtype: object
```

Conclusion

1. 추천 시스템을 줄거리 기반이나 키워드 기반과 같이 특정 알고리즘으로 구현하는 것은 실제로 소비자 들이 해당 추천에 얼마나 만족했는지 정보를 얻기가 어려워 성능 측정이 어려울 수도 있을 것이라고 생각하였다.
2. 추천 시스템을 자주 다루지 않아 몰랐었는데 생각보다 최근에 나타난 인공지능 모델보다는 알고리즘 에 가까운 예전의 기법들이 사용되었던 것을 알게 되었다.
3. 보다 중요한 것은 넷플릭스 화면에도 나타나듯이 장르의 TOP10과 같은 정말 기본적인 추천이 소비자 들에게 추천하는 효과가 좋을 수 있기 때문에 상황에 맞게 적절하면서 안정적인 모델을 활용하는 것이 중요할 것이라고 생각하였다.

The End

2024. 11. 03. 월