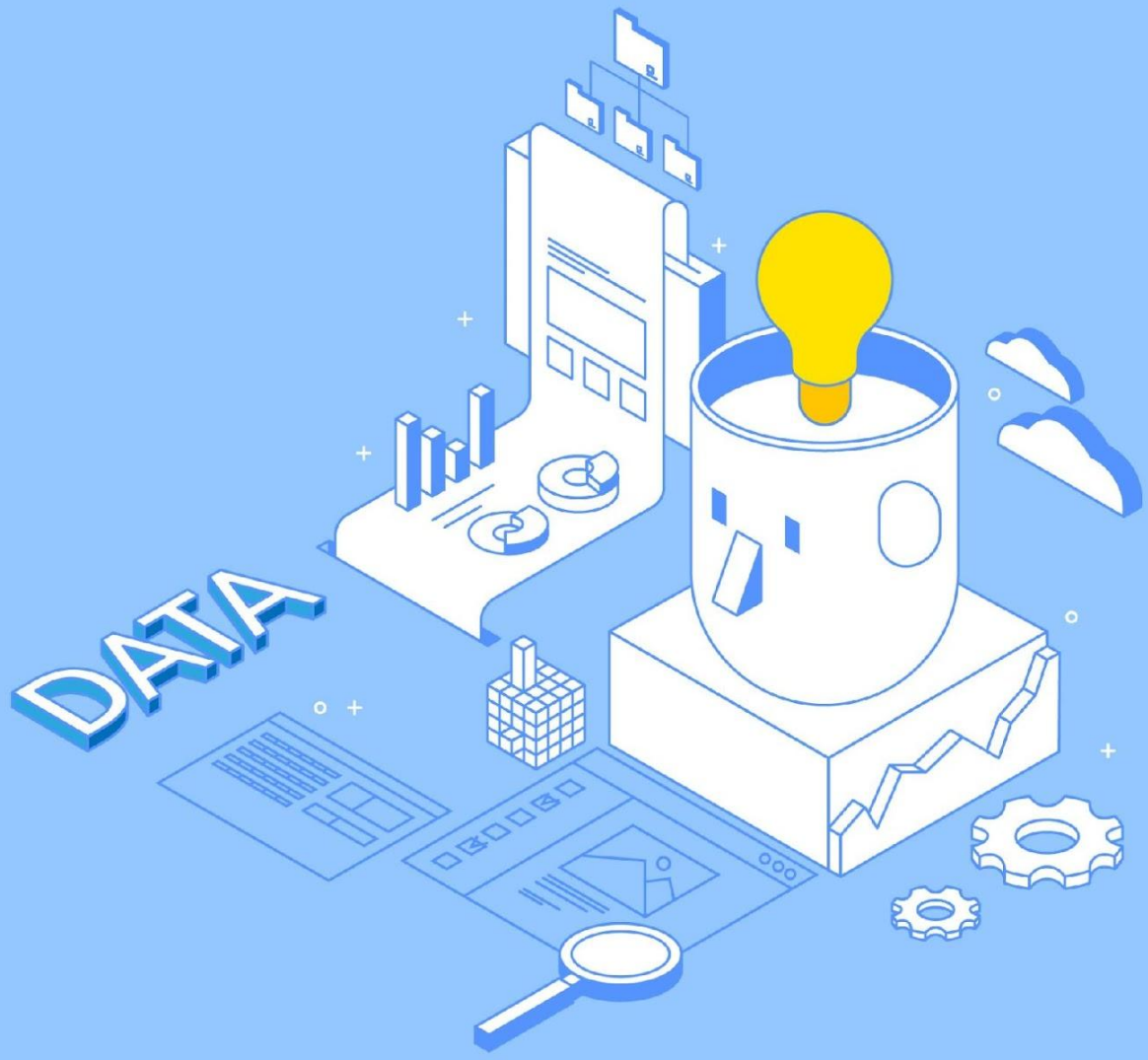
2021 - 2 Big Data

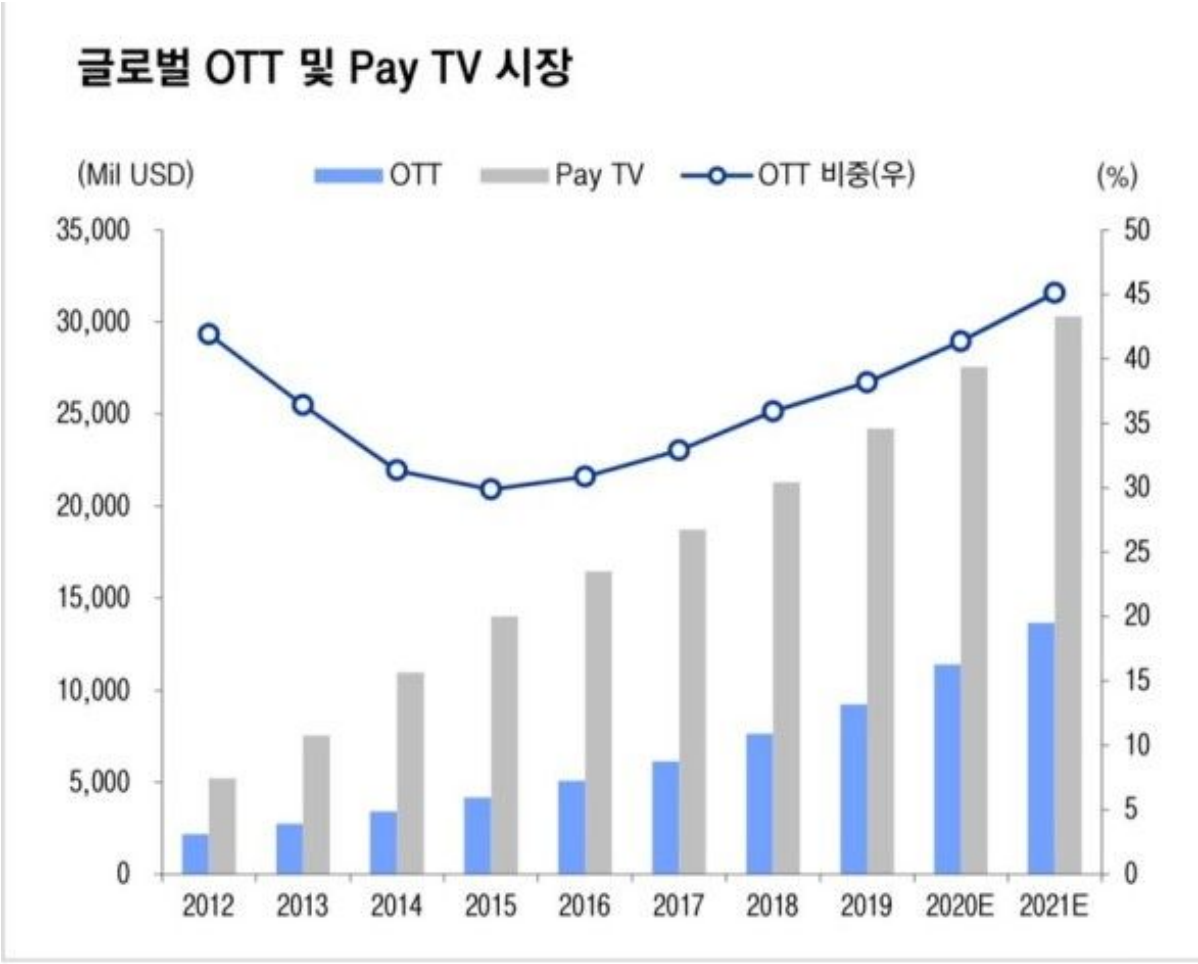
Movies on OTT Platforms & MBTI



Contents

Introduction

COVID-19 & OTT Platforms



자료: Bloomberg, Netflix, Global Entertainment&Media Outlook, 이베스트투자증권 리서치센

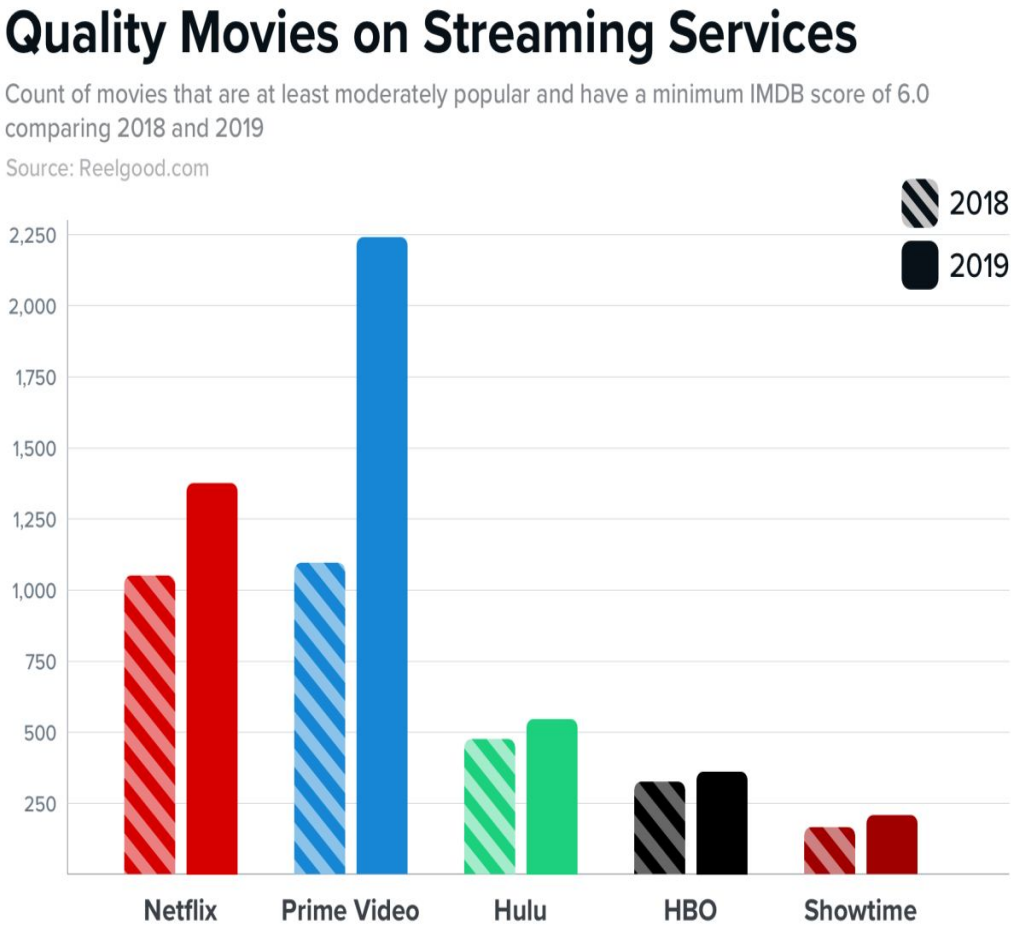
자료: 넷플릭스

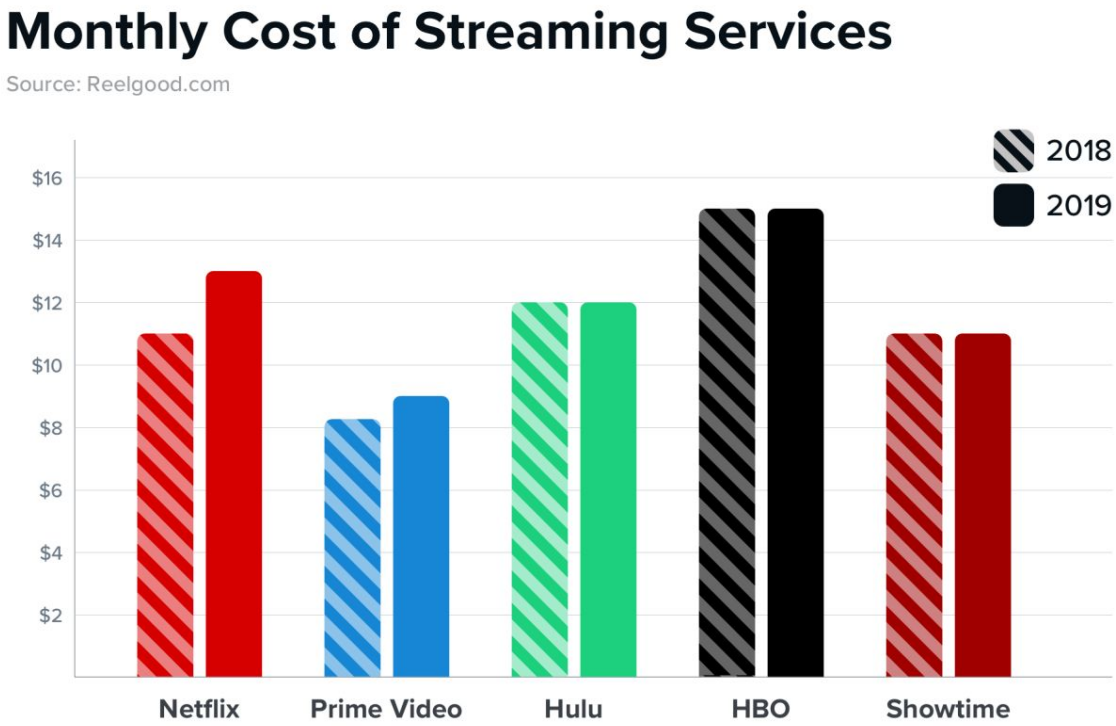
The JoongAng

NETFLIX amazon Prime hulu Disney+

Problem Definition

Importance of selecting OTT Platform





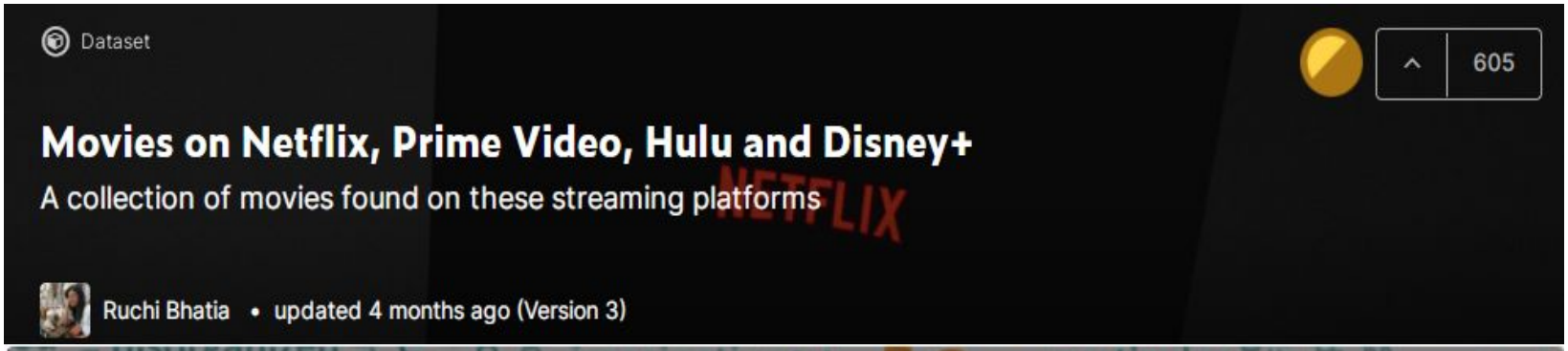
Previous approach

Lack Of Reflecting User's Propensity

Movie Recommendation System

- **Demographic Filtering**
  -Not sensitive to a particular user
  -based on movie popularity
- **Contents-Based Filtering**
  -Most likely to be similar
- **Collaborative Filtering**
  -based on persons matching

Data Description



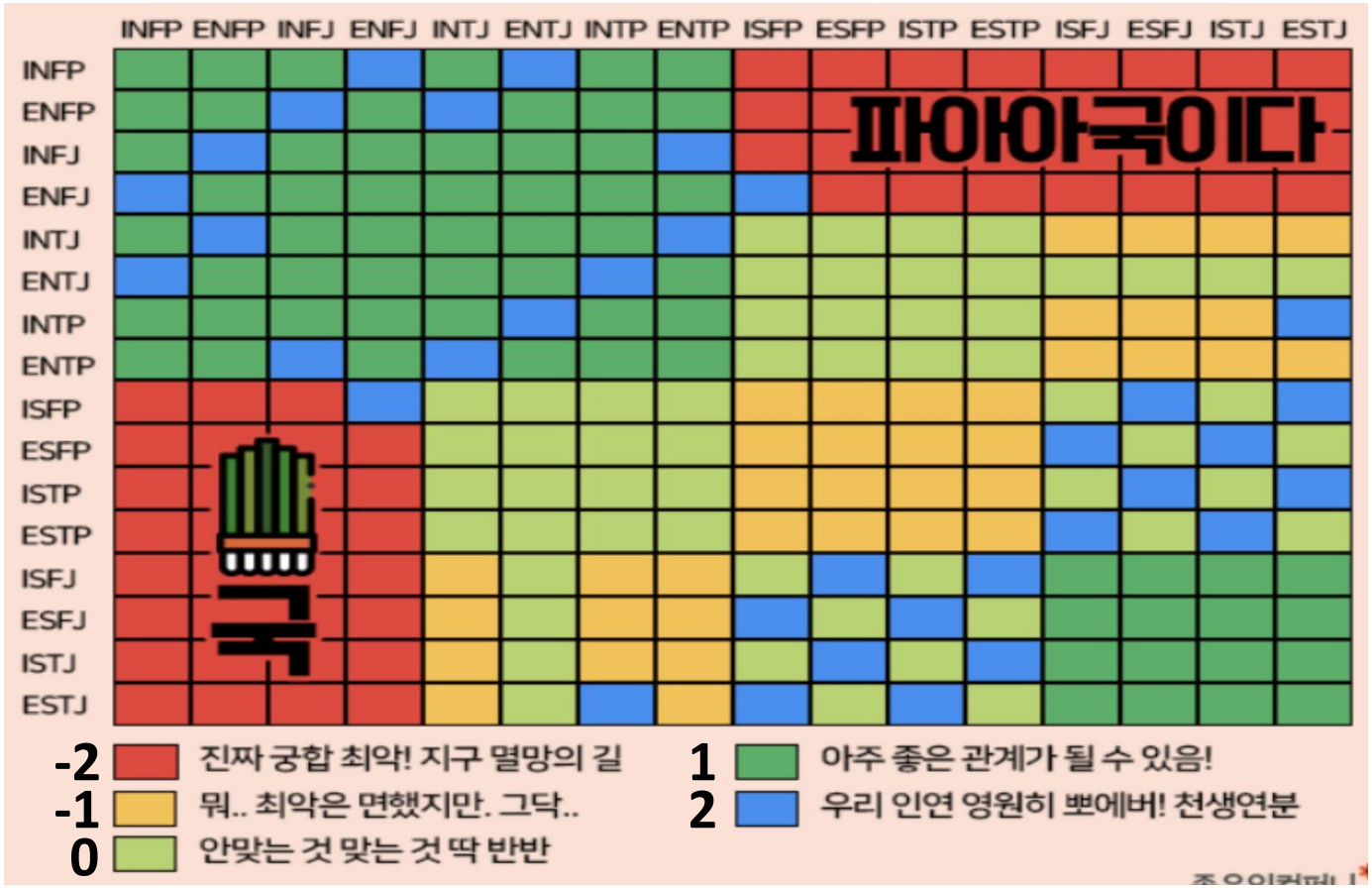
Title, IMDb, Rotten Tomatoes, Netflix, Hulu, Prime Video, Disney+, Genres

mbti, role, movie

Data Preprocessing

1. Missing Value & Data Processing
   -Delete NaN, Duplication, MBTI “XXXX”
2. Data Transformation
   -Change ‘ /10’, ‘ /100’ into %
3. Dataset Merge
   -Change MBTI ‘movie’ into ‘Title’

Modeling Methodology



+Personal Taste

1~5 X
(Affected by number of characters)

-2~2 O
(unique MBTI)

ISTJ:	11-14%
ISFJ:	9-14%
ESFJ:	9-13%
ESTJ:	8-12%
ISFP:	5-9%
ESFP:	4-9%
ENFP:	6-8%
ISTP:	2-4%
ESTP:	4-5%
INFP:	4-5%
INTP:	3-5%
ENTJ:	2-5%
ENTP:	2-5%
ENFJ:	2-5%
INTJ:	2-4%
INFJ:	1-3%

Compatibility between User & Character -> Fit well with Character's Propensity -> Understanding & Interest of Character -> Immersion in the movie

Type Genre & User MBTI -> Sum of Compatibility + IMDb & Rotten Tomatoes -> Top 5 movie & OTT platform

Results - Analytic

1) MBTI Compatibility Matrix Formation

```
mbti_score = np.matrix([
    [1, 1, 1, 2, 1, 2, 1, 1, -2, -2, -2, -2, -2, -2, -2, -2],
    [1, 1, 2, 1, 2, 1, 1, 1, -2, -2, -2, -2, -2, -2, -2, -2],
    [1, 2, 1, 1, 1, 1, 1, 2, -2, -2, -2, -2, -2, -2, -2, -2],
    [2, 1, 1, 1, 1, 1, 1, 1, 2, -2, -2, -2, -2, -2, -2, -2],
    [1, 2, 1, 1, 1, 1, 1, 2, 0, 0, 0, 0, -1, -1, -1, -1],
    [2, 1, 1, 1, 1, 1, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0],
    [1, 1, 1, 1, 1, 2, 1, 1, 0, 0, 0, 0, -1, -1, -1, 2],
    [1, 1, 2, 1, 2, 1, 1, 1, 0, 0, 0, 0, -1, -1, -1, -1],
    [-2, -2, -2, 2, 0, 0, 0, 0, -1, -1, -1, -1, 0, 2, 0, 2],
    [-2, -2, -2, -2, 0, 0, 0, 0, -1, -1, -1, -1, 2, 0, 2, 0],
    [-2, -2, -2, -2, 0, 0, 0, 0, -1, -1, -1, -1, 0, 2, 0, 2],
    [-2, -2, -2, -2, 0, 0, 0, 0, -1, -1, -1, -1, 2, 0, 2, 0],
    [-2, -2, -2, -2, -1, 0, -1, -1, 0, 2, 0, 2, 1, 1, 1, 1],
    [-2, -2, -2, -2, -1, 0, -1, -1, 2, 0, 2, 0, 1, 1, 1, 1],
    [-2, -2, -2, -2, -1, 0, -1, -1, 0, 2, 0, 2, 1, 1, 1, 1],
    [-2, -2, -2, -2, -1, 0, 2, -1, 2, 0, 2, 0, 1, 1, 1, 1]
])
```

```
mbti_all = dict(
    INFP=0,
    ENFP=1,
    INFJ=2,
    ENFJ=3,
    INTJ=4,
    ENTJ=5,
    INTP=6,
    ENTP=7,
    ISFP=8,
    ESFP=9,
    ISTP=10,
    ESTP=11,
    ISFJ=12,
    ESFJ=13,
    ISTJ=14,
    ESTJ=15
)
```

Results - Analytic

2) MBTI Compatibility Function Definition

```
def movie_list(user_mbti, genre):
  movies = list(set(data.Title[data.mbti == user_mbti]) & set(data.Title[data.Genres == genre]))
  if movies == [] : return None
  else : return movies
```

```
def sum_score(user_mbti, movies_list):
  L = []
  a = mbti_all.get(user_mbti)
  for i in range(len(movies_list)):
    s = 0
    n = data.mbti[data.Title == movies_list[i]].unique()
    for j in range(len(n)):
      b = mbti_all.get(n[j])
      s += mbti_score.A[a][b]
    L.append(s)
  return L
```

Results - Analytic

3) Recommendation Function Definition

```
def recommended_movie(user_mbti, movie_list) :
  scores = sum_score(user_mbti, movie_list)

  for i in range(len(movie_list)) :
    scores[i] += data.IMDb[data.Title == movie_list[i]].unique()
    scores[i] += data.Rotten_Tomatoes[data.Title == movie_list[i]].unique()

  idx = np.argmax(scores)
  ott = data.OTT[data.Title==movie_list[idx]].unique()
  ott_name = ""

  for i in ott :
    ott_name += i
    ott_name += " "

  print("Recommended Movie : {} , Serviced by : {}".format(movie_list[idx], ott_name))
```

Results - Analytic

4) Final Result

```
M = input("What is your mbti?> ")
```

```
What is your mbti?> ESTJ
```

```
g = input("Which genre do you want to watch?> ")
```

```
Which genre do you want to watch?> Drama
```

```
mlist = movie_list(M, g)
print(mlist)
```

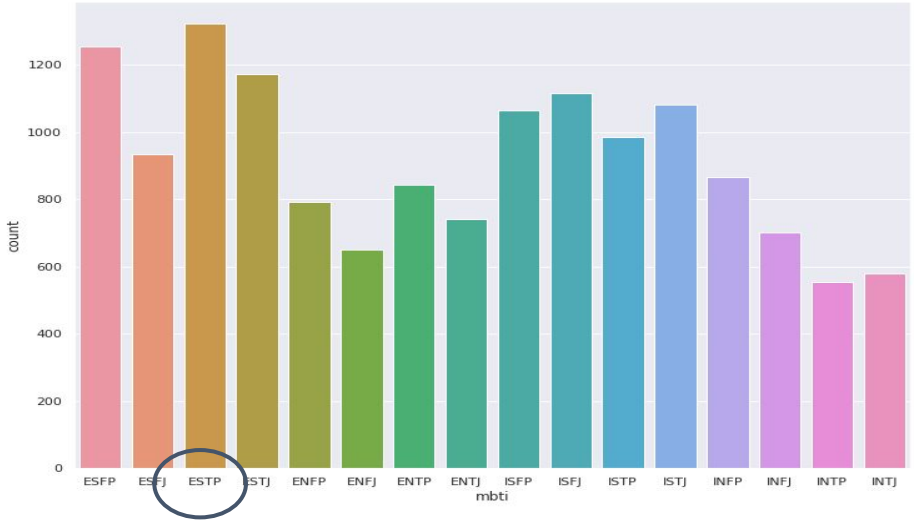
```
['On the Waterfront', 'My Fair Lady', 'The Abyss', 'A Christmas Princess', 'Twilight', 'Trumbo', 'Ip Man', 'Panipat', 'Diary of a Wim
```

```
recommended_movie(M, mlist)
```

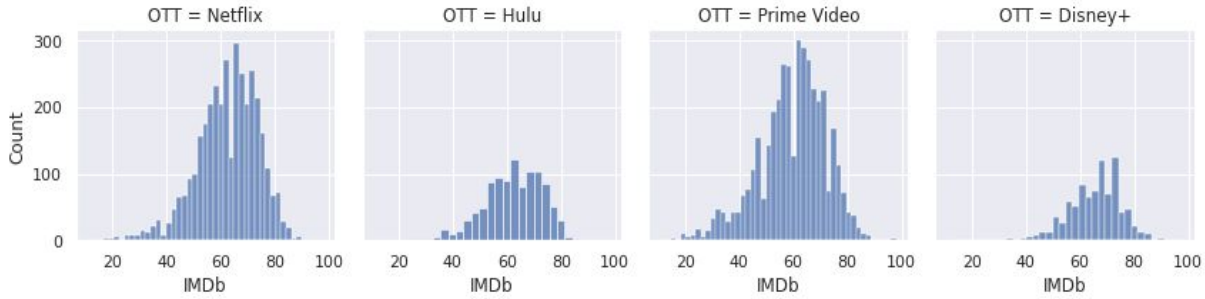
```
Recommended Movie : Ip Man , Serviced by : Netflix
```

Results - Visualization

MBTI counts

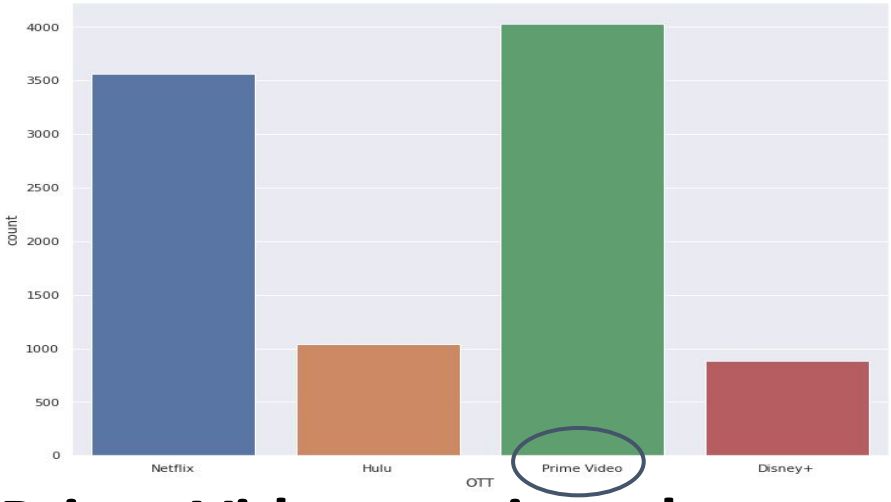


IMDB score distribution

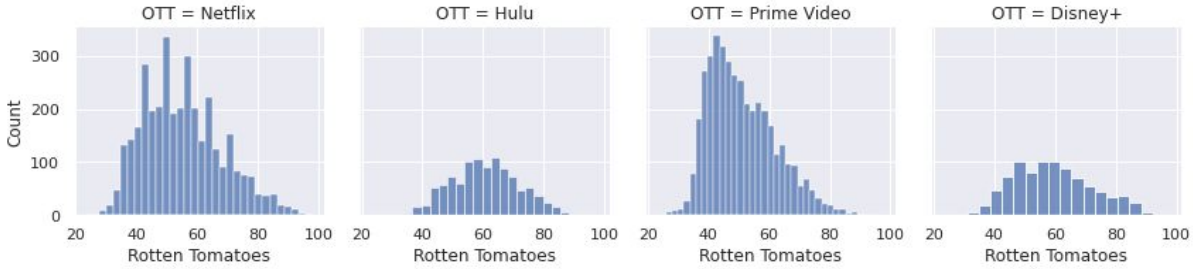


most common personality type in movie characters：**ESTP**

Movie Counts



Rotten tomatoes distribution



Prime Video services the most movies

Conclusion

Evaluations

- 사용자의 성향에 적합한 영화 및 해당영화를 서비스화하는 OTT 플랫폼을 추천

- 콜드 스타트 문제가 발생하지 않는다.

- MBTI는 최근에 주목받는 심리검사 지표이므로, 많은 사람들의 흥미를 기반으로 더욱 넓은 영역으로 확장되어 적용할 수 있을 것으로 기대된다.

- MBTI 궁합이 좋은 영화캐릭터에 대한 영화를 통해서 자신에 대해 더욱 알아갈 수 있는 기회를 갖을 수 있다.

Limitations

- MBTI 궁합이 심리유형론을 기반으로 하므로, MBTI를 이용한 추천시스템의 과학적 근거 부족

- 전처리 과정에서 데이터 손실

Conclusion

Developments & Further Applications

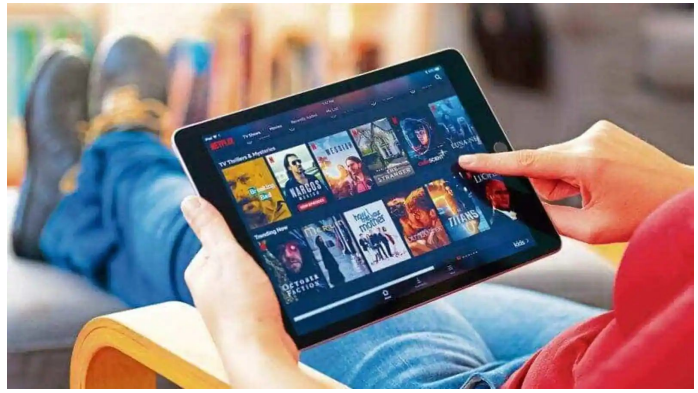
– 북미의 대중적인 OTT 플랫폼에 국한, 다양한 국가의 OTT플랫폼으로의 확장

– MBTI 궁합도와 장르 기반 영화 줄거리에 대한 자연어 처리 과정을 추가 및 영화에 대한 비평을 예측 –> 합리적인 OTT 플랫폼 결정

– 각 MBTI의 댓글을 모은 데이터셋 + 영화 줄거리에 대한 데이터셋–> 코사인 유사도를 통한 자연어 처리

– 정신적인 지표를 수학 및 과학적인 방법에서 접근 – 객관적인 지표화

– 메인 캐릭터의 데이터셋에 한정 한 분석 –> 효과적인 분석 기대



Thank you for listening!