데이터 통계
단기 프로젝트

Netflix 데이터 통계

2020.12.18

목차

1. 주제 선택 및 주제 분석 복적
2. 분석 방법
3. 분석 내용
4. 결론 및 고찰

1. 주제 선택 및 주제 분석 목적

사전 조사 중 Netflix의 성공의 비결 중 하나가 데이터 분석이라는 기사를 보고 흥미가 생겨 이 주제를 선택하였다.

홈 > 기획

[왜 넷플릭스인가 ③] 빅데이터 기술로 국내 시장 파고들다

이승균 기자 | 입력 2019.02.21 13:01 | 수정 2019.02.21 13:02 | 댓글 0

2016년 첫 상륙한 넷플릭스는 유독 한국에서만큼은 맥을 못 추었다. 기존 OTT(Over The Top, 인터넷을 통해 볼 수 있는 TV서비스) 플랫폼들이 견고하게 자리를 잡고 있는 탓도 있었고, 넷플릭스에 볼만한 콘텐츠가 없기 때문이기도 했다.

그러나 봉준호 감독의 '옥자'를 시작으로 넷플릭스는 조금씩 천천히 한국형 콘텐츠를 통해 존재감을 넓히고, 이미 해외시장에서 인정받은 자체 제작 오리지널 콘텐츠의 입소문을 통해 상륙 2년 만에 100만 가입자를 확보하기에 이른다.

국내 지상파들은 작년부터 열차별을 주장하며 넷플릭스를 견제하고 있는데

최신뉴스

- '상법개정안 법사위 통과'..경
- [포토] BIG3 성과공유회에서
- [포토] 공수처법 법사위 통과
- 농협금융지주, 차기 회장 후보
- 하이트진로, 정기 임원인사 단

포토뉴스

넷플릭스(Netflix)는 어떻게 내 취향을 분석할까?

데이터로 연결하는 스마트한 세상 (주)비트나인 | 2020. 5. 27. 15:39

https://www.mediasr.co.kr/news/articleView.html?idxno=51354
https://bitnine.tistory.com/380

2. 분석 방법

- 주제에 맞는 데이터 셋 선정
- 데이터 구조 분석 및 파생 칼럼 생성
- 다양한 유형의 그래프나 표로 데이터 시각화

2. 분석 방법

항목	11. 20	21	22	23	24	25	26	27	28	29	30	12. 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Data set 분석																													
프로그래밍																													
결과 정리 및 보고서 작성																													
검토 및 제출																													

3. 분석 내용

Data Set :

All TV Shows and Movies meta data on Netflix. Updated every month.
Netflix의 모든 TV show들과 Movie들의 meta data로 매달 갱신됨.

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	81145628	Movie	Norm of the North: King Sized Adventure	Richard Finn, Tim Maltby	Alan Marriott, Andrew Toth, Brian Dobson, Cole...	United States, India, South Korea, China	September 9, 2019	2019	TV-PG	90 min	Children & Family Movies, Comedies	Before planning an awesome wedding for his gra...
1	80117401	Movie	Jandino: Whatever it Takes	NaN	Jandino Asporaat	United Kingdom	September 9, 2016	2016	TV-MA	94 min	Stand-Up Comedy	Jandino Asporaat riffs on the challenges of ra...
2	70234439	TV Show	Transformers Prime	NaN	Peter Cullen, Sumalee Montano, Frank Welker, J...	United States	September 8, 2018	2013	TV-Y7-FV	1 Season	Kids' TV	With the help of three human allies, the Autob...
3	80058654	TV Show	Transformers: Robots in Disguise	NaN	Will Friedle, Darren Criss, Constance Zimmer, ...	United States	September 8, 2018	2016	TV-Y7	1 Season	Kids' TV	When a prison ship crash unleashes hundreds of...
4	80125979	Movie	#realityhigh	Fernando Lebrija	Nesta Cooper, Kate Walsh, John Michael Higgins...	United States	September 8, 2017	2017	TV-14	99 min	Comedies	When nerdy high schooler Dani finally attracts...
...	...	...	...	...	...	...	...	...	...	...	...	...
6229	80000063	TV Show	Red vs. Blue	NaN	Burnie Burns, Jason Saldaña, Gustavo Sorola, G...	United States	NaN	2015	NR	13 Seasons	TV Action & Adventure, TV Comedies, TV Sci-Fi ...	This parody of first-person shooter games, mil...
6230	70286564	TV Show	Maron	NaN	Marc Maron, Judd Hirsch, Josh Brener, Nora Zeh...	United States	NaN	2016	TV-MA	4 Seasons	TV Comedies	Marc Maron stars as Marc Maron, who interviews...

feature의 의미

Feature	설명
type	타입, 'TV show'와 'Movie'로만 구성
date_added	Netflix에 추가된 날짜 (일-월-년)
year_added	date_added의 연도별 비교를 위해 date_added의 연도를 추출하여 파생 feature 생성
relase_year	실제 출시 년도
dif	출시된 년도와 추가된 년도의 차이를 비교하기 위해 release_year 값과 year_added 값의 차이를 계산하여 파생 feature 생성
listed_in	장르 (TV show, Movie별 장르 다름)
description	줄거리(요약 설명)

파생 Feature 생성 방법

- year_added

```
netflix_dataset['date_added'] = pd.to_datetime(netflix_dataset['date_added'])
netflix_dataset['year_added'] = netflix_dataset['date_added'].dt.year
```

- dif

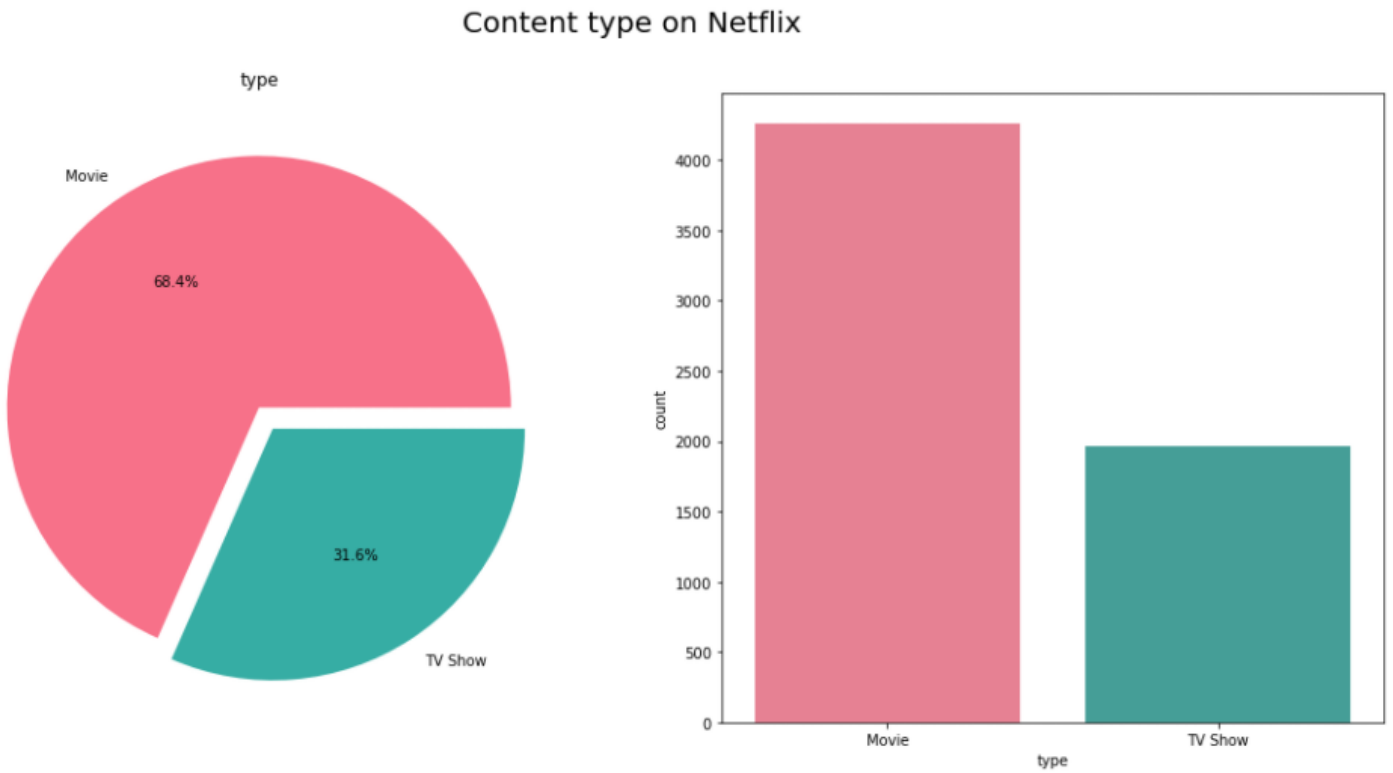
```
netflix_dataset['dif'] = netflix_dataset['year_added'] - netflix_dataset['release_year']
```

생성 결과

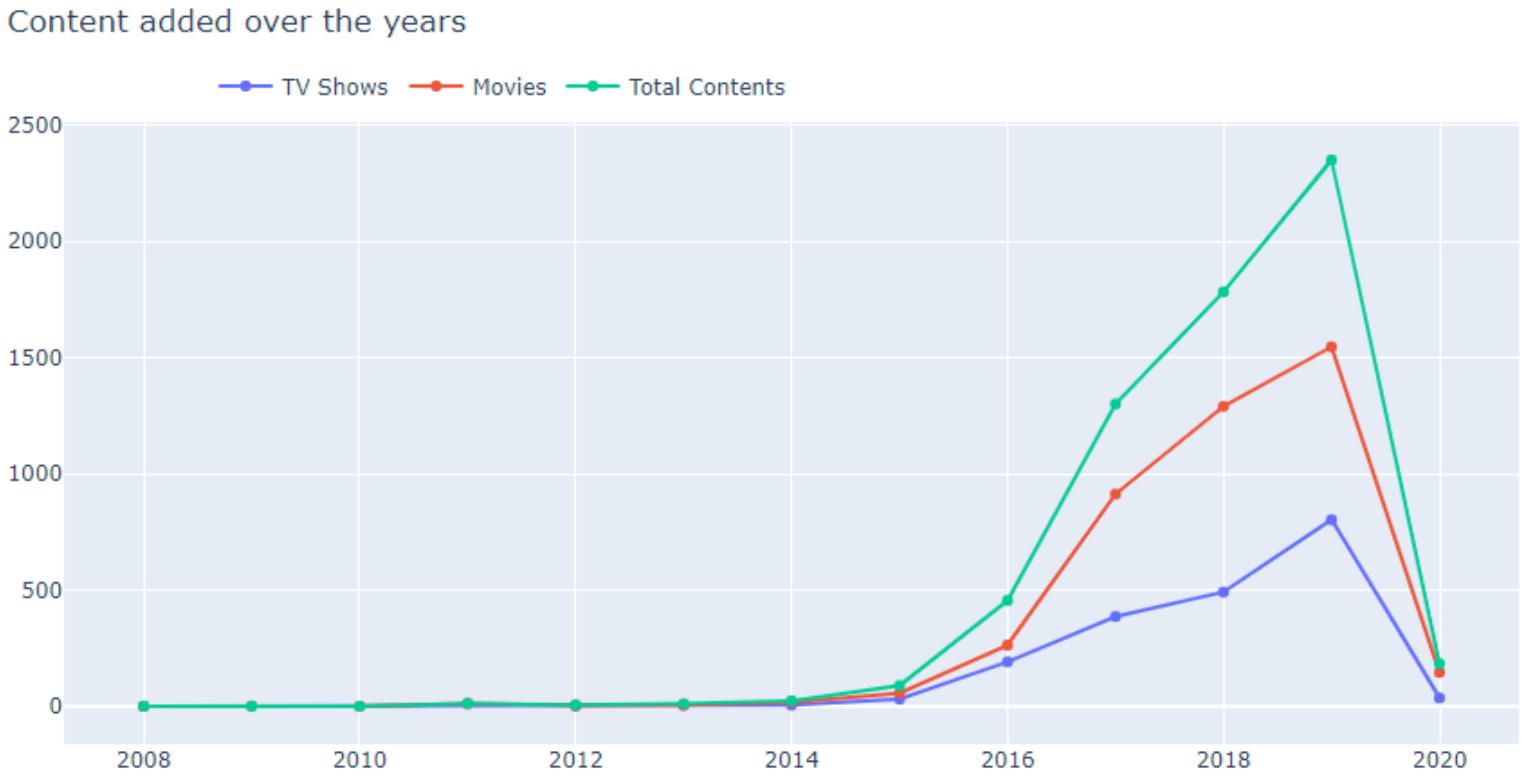
	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description	year_added	dif
0	81145628	Movie	Norm of the North: King Sized Adventure	Richard Finn, Tim Maltby	Alan Marriott, Andrew Toth, Brian Dobson, Cole...	United States, India, South Korea, China	2019-09-09	2019	TV-PG	90 min	Children & Family Movies, Comedies	Before planning an awesome wedding for his gra...	2019.0	0.0
1	80117401	Movie	Jandino: Whatever it Takes	NaN	Jandino Asporaat	United Kingdom	2016-09-09	2016	TV-MA	94 min	Stand-Up Comedy	Jandino Asporaat riffs on the challenges of ra...	2016.0	0.0
2	70234439	TV Show	Transformers Prime	NaN	Peter Cullen, Sumalee Montano, Frank Welker, J...	United States	2018-09-08	2013	TV-Y7-FV	1 Season	Kids' TV	With the help of three human allies, the Autob...	2018.0	5.0

① type - type별 비율, 개수 비교

```
f,ax = plt.subplots(1, 2, figsize=(18,8))
color = sns.color_palette('husl', 2)

netflix_dataset['type'].value_counts().plot.pie(explode=[0,0.1], autopct='%1.1f%%', ax=ax[0], colors=color)
ax[0].set_title('type')
ax[0].set_ylabel('')

sns.countplot(x='type', data=netflix_dataset, palette=color)

plt.suptitle('Content type on Netflix', fontsize=20)
plt.show()
```



② year_added - 연도별 year_added 비교



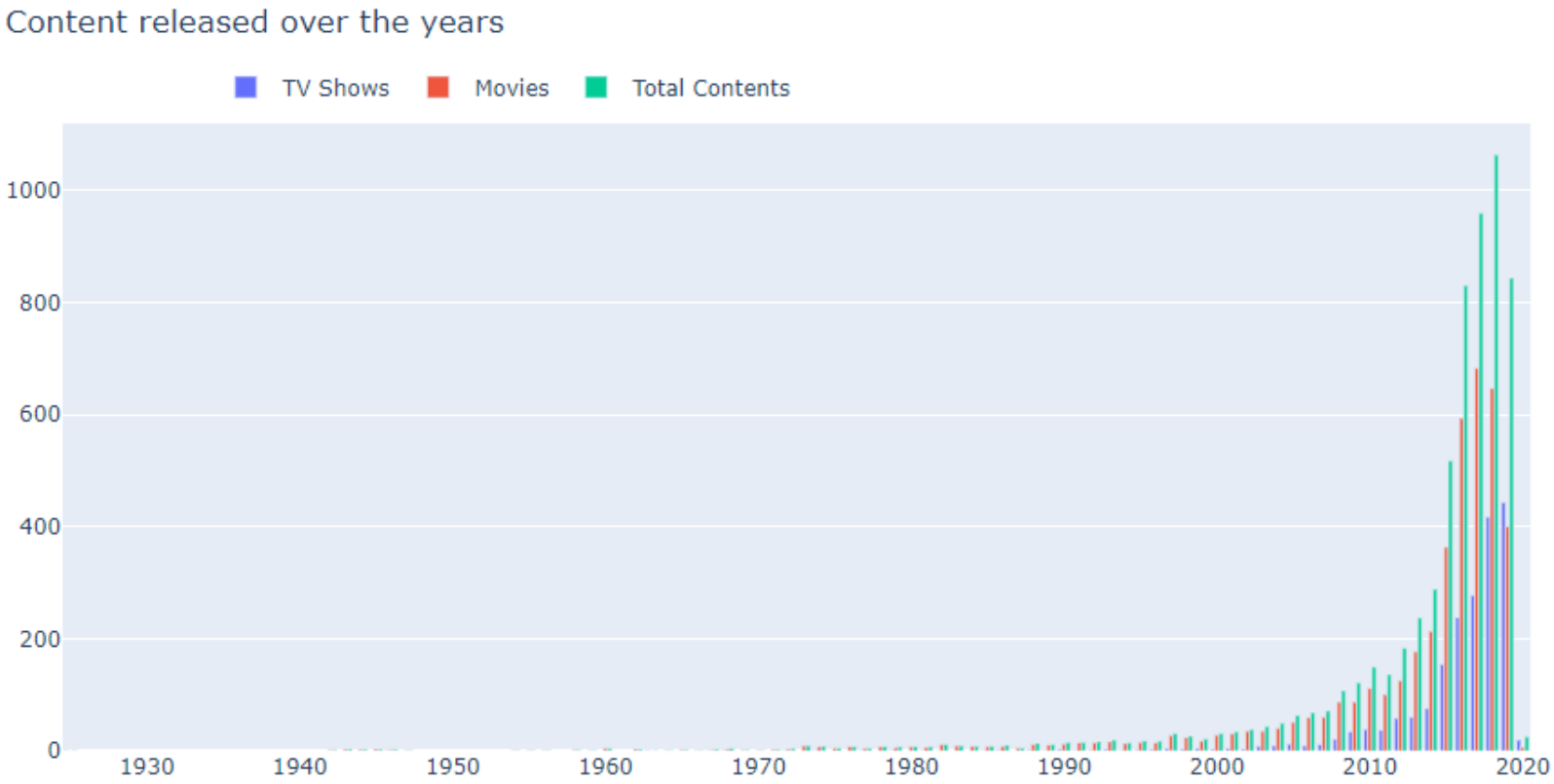
② year_added

```
tv_show = netflix_dataset[netflix_dataset['type']=='TV Show']
movies = netflix_dataset[netflix_dataset['type']=='Movie']

col = 'year_added'

content = netflix_dataset[col].value_counts().reset_index().rename(columns={col : 'count', 'index' : col}).sort_values(col)
content['percent'] = content['count'].apply(lambda x : 100*x/sum(content['count']))

tv_content = tv_show[col].value_counts().reset_index().rename(columns={col : 'count', 'index' : col}).sort_values(col)
tv_content['percent'] = tv_content['count'].apply(lambda x : 100*x/sum(tv_content['count']))

movies_content = movies[col].value_counts().reset_index().rename(columns={col : 'count', 'index' : col}).sort_values(col)
movies_content['percent'] = movies_content['count'].apply(lambda x : 100*x/sum(movies_content['count']))

movie = go.Scatter(x=movies_content[col], y=movies_content['count'], name='Movies')
tv = go.Scatter(x=tv_content[col], y=tv_content['count'], name='TV Shows')
total = go.Scatter(x=content[col], y=content['count'], name='Total Contents')

data = [tv, movie, total]

layout = go.Layout(title='Content added over the years', legend=dict(x=0.1, y=1.1, orientation='h'))
fig = go.Figure(data, layout=layout)
fig.show()
```

③ release_year - 연도별 release_year 비교

```
col = 'release_year'

tv_content = tv_show[col].value_counts().reset_index().rename(columns = {col : 'count', 'index' : col}).sort_values(col)
tv_content['percent'] = tv_content['count'].apply(lambda x : 100*x/sum(tv_content['count']))

movies_content = movies[col].value_counts().reset_index().rename(columns = {col : 'count', 'index' : col}).sort_values(col)
movies_content['percent'] = movies_content['count'].apply(lambda x : 100*x/sum(movies_content['count']))

content = netflix_dataset[col].value_counts().reset_index().rename(columns = {col : 'count', 'index' : col}).sort_values(col)
content['percent'] = content['count'].apply(lambda x : 100*x/sum(content['count']))

tv = go.Bar(x=tv_content[col], y=tv_content['count'], name='TV Shows')
movie = go.Bar(x=movies_content[col], y=movies_content['count'], name='Movies')
total = go.Bar(x=content[col], y=content['count'], name='Total Contents')

data = [tv, movie, total]

layout = go.Layout(title='Content released over the years', legend=dict(x=0.1, y=1.1, orientation='h'))
fig = go.Figure(data, layout=layout)
fig.show()
```

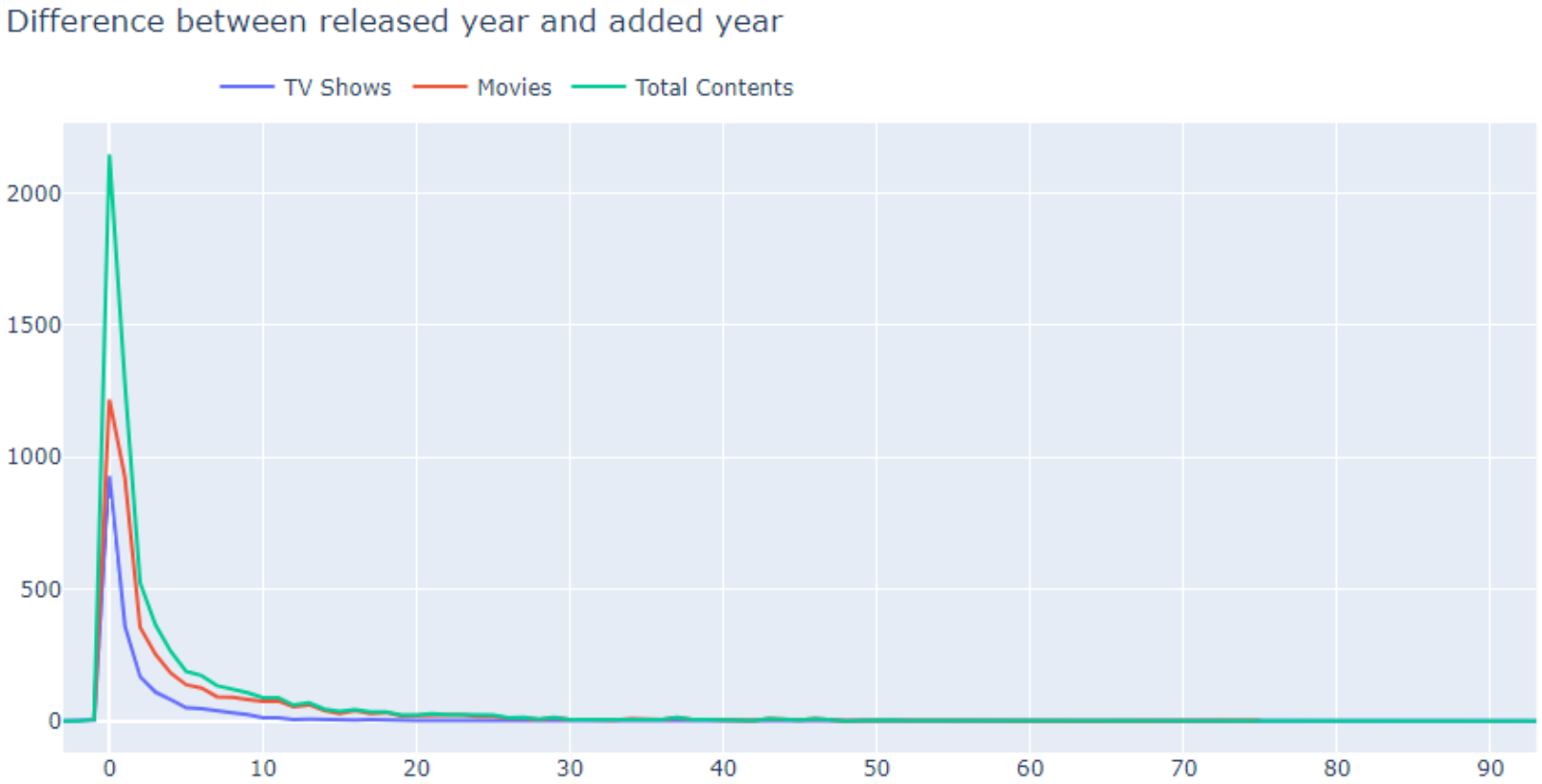
③ release_year



④ dif - release_year 와 year_added 의 차이 비교

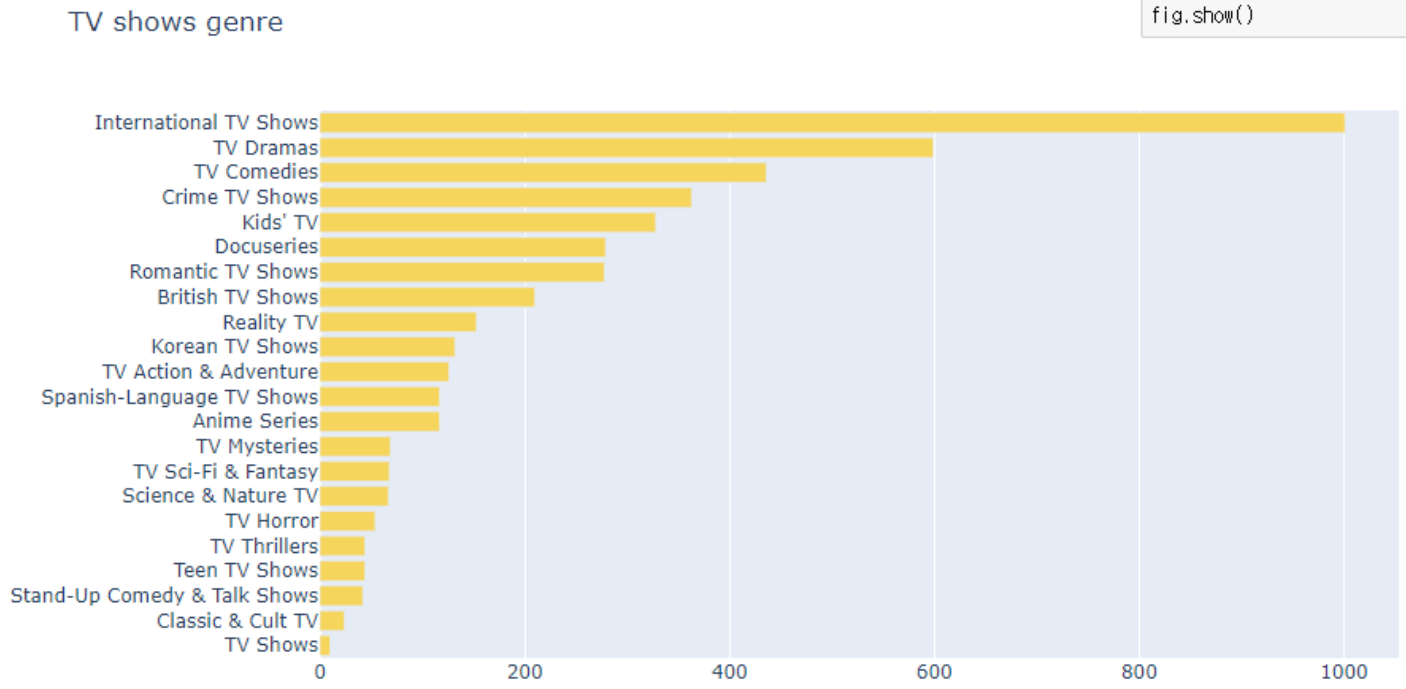
```
col = 'dif'

content = netflix_dataset[col].value_counts().reset_index().rename(columns={col : 'count', 'index' : col}).sort_values(col)
content['percent'] = content['count'].apply(lambda x : 100*x/sum(content['count']))

tv_content = tv_show[col].value_counts().reset_index().rename(columns={col : 'count', 'index' : col}).sort_values(col)
tv_content['percent'] = tv_content['count'].apply(lambda x : 100*x/sum(tv_content['count']))

movies_content = movies[col].value_counts().reset_index().rename(columns={col : 'count', 'index' : col}).sort_values(col)
movies_content['percent'] = movies_content['count'].apply(lambda x : 100*x/sum(movies_content['count']))

movie = go.Scatter(x=movies_content[col], y=movies_content['count'], name='Movies')
tv = go.Scatter(x=tv_content[col], y=tv_content['count'], name='TV Shows')
total = go.Scatter(x=content[col], y=content['count'], name='Total Contents')

data = [tv, movie, total]

layout = go.Layout(title='Difference between released year and added year', legend=dict(x=0.1, y=1.1, orientation='h'))
fig = go.Figure(data, layout=layout)
fig.show()
```

④ dif


Difference between released year and added year
TV Shows
Movies
Total Contents
2000
1500
1000
500
0
0
10
20
30
40
50
60
70
80
90

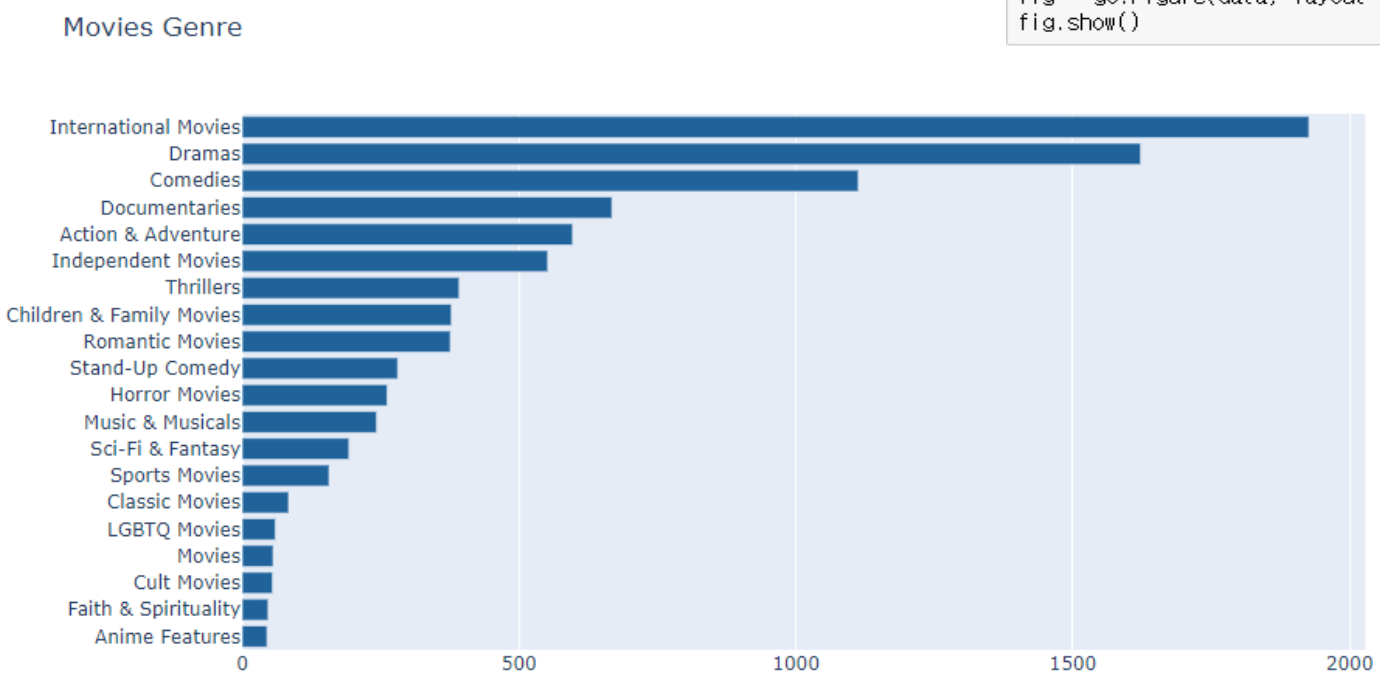
⑤ listed_in – tv show 개수

```
col = 'listed_in'

categories = ', '.join(tv_show['listed_in']).split(', ')
counter_list = Counter(categories).most_common(50)
labels = [_[0] for _ in counter_list][::-1]
values = [_[1] for _ in counter_list][::-1]
trace1 = go.Bar(y=labels, x=values, orientation='h', name='TV Shows', marker=dict(color='#F6D55C'))

data = [trace1]
layout = go.Layout(title='TV shows genre', legend=dict(x=0.1, y=1.1, orientation='h'))
fig = go.Figure(data, layout=layout)
fig.show()
```



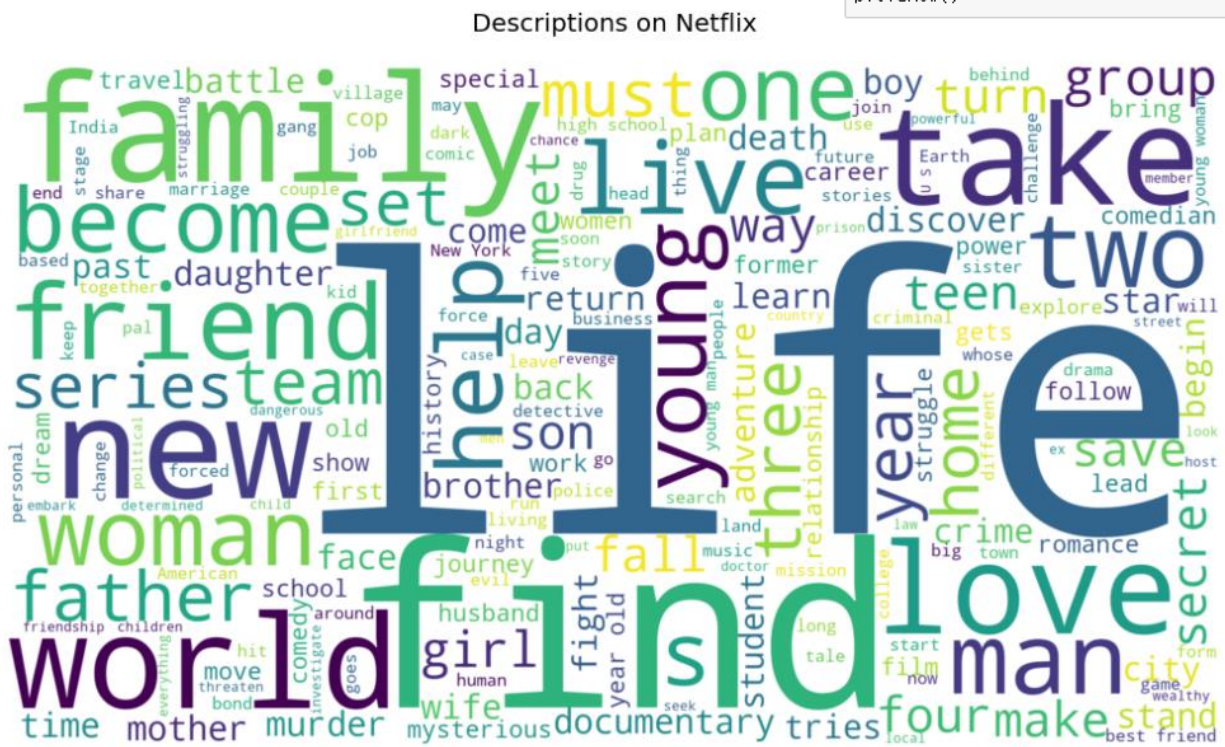
⑤ listed_in – movie 개수

```
col = 'listed_in'

categories = ', '.join(movies['listed_in']).split(', ')
counter_list = Counter(categories).most_common(50)
labels = [_[0] for _ in counter_list][::-1]
values = [_[1] for _ in counter_list][::-1]
trace1 = go.Bar(y=labels, x=values, orientation='h', name='Movies', marker=dict(color='#20639B'))

data = [trace1]
layout = go.Layout(title='Movies Genre', legend=dict(x=0.1, y=1.1, orientation='h'))
fig = go.Figure(data, layout=layout)
fig.show()
```

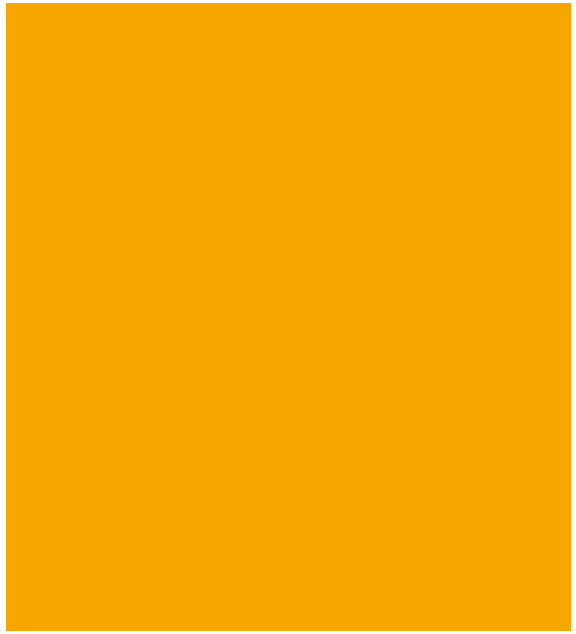


⑥ description - word cloud

```
wordcloud = WordCloud(background_color='white', width=1920, height=1080).generate(' '.join(netflix_dataset.description))

plt.figure(figsize=(18,10))
plt.imshow(wordcloud,interpolation='bilinear')

plt.axis('off')
plt.title('Descriptions on Netflix\n', fontsize=20)
plt.show()
```



4. 결론 및 고찰



결론 및 고찰

Feature	결론 및 고찰
type	Netflix에는 TV Show가 대략 2000개, Movie가 4500개 있고, TV Show보다 Movie가 더 많다. Netflix는 TV Show보다 Movie를 더 선호하는 것으로 보인다.
year_added	2008년에 처음 작품이 추가 되었고, 2019년에 가장 많이 추가 되었다. 또한, 추가되는 작품 수는 계속 증가하고 있다. Netflix에 추가되는 작품 수는 매년 증가할 것으로 보인다.
release_year	가장 먼저 출시된 연도가 1925년이며, 전체적으로는 2018년에 출시된 작품이 가장 많고, Movie의 경우 2017년, TV Show의 경우 2019년에 출시된 작품이 가장 많다. 즉, Netflix에는 최근에 출시된 작품이 많다.
dif	작품이 출시된 직후 Netflix에 추가된 경우 즉, 차이가 0인 경우가 가장 많고, 가장 많은 차이가 나는 경우는 TV Show로 93년 차이가 난다. Netflix는 최근에 출시된 작품을 추가하는 경우가 많으나, 오래된 작품도 주목하고 있는 것으로 생각된다.
listed_in	TV의 경우 International TV Shows, TV Dramas, TV Comedies 순으로 많고, Movie의 경우 International Movies, Dramas, Comedies 순으로 많다. type에 따라 장르는 다르지만, 선호 장르는 비슷한 것으로 보인다.
description	description에 'life', 'find', 'world', 'family' 등의 단어가 많이 사용된다. Netflix는 밝은 이미지의 작품이 많은 것으로 보인다.

결론 및 고찰

쥬피터 노트북 소스코드를 살펴보고, 출시 년도와 추가 년도의 차이를 비교한 경우가 없기에 추가로 분석해보았다. 그리고 다른 코드들을 참고하여 그래프를 작성하며 x값의 범위가 넓고 한 곳에만 집중된 경우 막대 그래프로 작성 시 소수의 값들을 파악하기 힘들다는 것을 알았지만, 이런 경우에 알맞은 그래프를 찾지 못해 아쉬웠다.

참고

- type

사용이유 및 목적: type별 개수 비교 뿐만 아니라 비율까지 비교한 점을 참고
https://www.kaggle.com/biphili/cinema-in-the-era-of-netflix

- date_added, release_year, listed_in

사용이유 및 목적: 그래프 작성 참고
https://www.kaggle.com/shikhnu/data-analysis-and-visualization-netflix-data
https://www.kaggle.com/shivamb/netflix-shows-and-movies-exploratory-analysis