

PROJECT

2025년 01월 27일 월요일

영화 개봉 계절 예측기

스마트팩토리혁신을 위한 AI 솔루션 개발자 양성과정

오시윤

목차 LIST

01 서론

주제 선정 및 배경, 목표

사용데이터 출처

일정 및 개발환경

02 데이터 전처리

활용데이터

자료 정제 및 병합

상관분석 및 그룹화, 시각화

03 Deep Neural Network 분석

DNN

04 예측 시스템 구현

fastAPI

05 결론

연구의 결과 및 시사점

01 주제 선정 및 배경

영화 평점, 개봉월, 장르 데이터 분석하여 계절별, 관객수 예측

- 영화 산업은 관객의 선호와 트렌드 변화에 민감
- 계절과 영화 장르에 따른 관객수 간의 관계 증명
- 영화 제작 및 마케팅 전략 수립에 있어 데이터 필요

01 목표

- 영화 평점, 개봉월, 장르 데이터 분석하여 계절별, 관객수 예측영화 평점, 개봉일, 장르 데이터를 분석하여 트렌드를 시각화
- 딥러닝 모델을 통해 계절별 인기 장르를 예측
- 분석 결과를 통해 영화 제작 및 마케팅 전략 수립에 필요한 인사이트 제공

01 프로젝트 진행과정

WORK FLOW




01 프로젝트 진행과정

WORK FLOW

[illegible]

01 데이터 사용처

KOFIC 영화진흥위원회 : <https://www.kofic.or.kr/kofic/business/main/main.do>



KOFIC
KOBIS
영화산업진흥재단

[회원가입](#) | [로그인](#)

영화정보 ▼
검색어를 입력하세요.

영화정보검색
박스오피스
테마통계
공식통계
온라인상영관 박스오피스
고객센터

박스오피스

월별 박스오피스

[홈](#) > [박스오피스](#) > [박스오피스](#) > 월별

- [박스오피스]코너는 실시간 발권데이터를 전일기준까지 반영하여 일별/주간/주말/기간별 등 각종 통계정보를 제공합니다.
- 매일 24시 이후 전환/제공되는 [전일자 통계정보]는 상영마감 및 보정처리 등의 사유로 이일 오전까지 계속 업데이트 되며, **일마감 후 데이터보정 등의 사유로 통계정보는 변동 될 수 있음을 참고하시기 바랍니다.**
- 통계이용안내
 - ①[박스오피스], [테마통계]코너는 연도별 영화상영권 연동물에 따라 실시간 수집된 발권데이터를 전일기준까지 반영한 통계정보입니다.
 - ②[공식통계]코너는 영신위에서 매년 발표하는 "한국영화연감"의 영화별 유행기록을 참고한 것입니다.
 - 한국영화연감(1971~2010) 통계를 기준으로 정리한 것이며, 2011년부터는 통합선산상을 기준으로 일정한 주기(개월, 매년)로 마감 처리하여 산출되는 통계정보입니다.
 - 통계마감 주기(월별, 년별)에 따라 공식통계 수치는 후후 변동될 수 있습니다.
 - 스크린수 : 조화기간에 상영된 일별 스크린수의 합계중 최대값
(예 : 1일 10개 스크린, 2일 20개 스크린, 3일 30개 스크린일 경우 1~3일의 스크린수는 30개)

박스오피스

해외 박스오피스

에메울

좌석점유율

상영점유율

• 조회기간 ?

2024

▼

05

~

2024

▼

06

▼

• 지역

--전체--

• 영화구분

--전체--

• 지역

--전체--

조회

01 데이터 사용 출처

naver : <https://www.naver.com> (영화 평점)

N 월별개봉영화

블로그 카페 이미지 지식iN 인플루언서 동영상 쇼핑 뉴스 < > ...

이런 영화 어때요?

월별개봉영화 현재상영영화 개봉예정영화 박스오피스

< 월 10월 11월 12월 2025 1월 2월 3월 4월 5월 6월 >

더 퍼스트 슬램덩크
개요 애니메이션 · 124분
재개봉 2025.01.04. ★
9.25



보러가기

예고편

해리 포터와 죽음의 성물 2
개요 판타지 · 131분
재개봉 2025.01.15. ★
9.31
출연 다니엘 래드클리프,
엠마 왓슨, 루퍼트...



보러가기

예고편

러브레터
개요 멜로/로맨스 · 117분
재개봉 2025.01.01. ★
9.32
출연 나카야마 미호,
토요카와 에즈시, 한...



색, 계
개요 멜로/로맨스 · 157분
재개봉 2025.01.01. ★
8.88
출연 양조위, 탕웨이, 조안
첸, 왕리홍, 탁중화,...



웹 크롤링

```
def naver_crawling_grade(grade_non_list, file_path):  
    dv = webdriver.Chrome()  
    dv.get('http://www.naver.com')  
    time.sleep(3)  
    el = dv.find_element(By.CSS_SELECTOR, 'input#query')  
  
    try:  
        movie = pd.read_csv(file_path)  
    except :  
        movie = pd.DataFrame({'MOVIE_NM': movie_title})  
  
    for title in grade_non_list :  
        el.clear()  
        el.send_keys('영화 {} 평점'.format(title))  
        el.send_keys(Keys.ENTER)  
        time.sleep(3)  
  
        try :  
            grades = dv.find_element(By.CSS_SELECTOR, 'span.area_star_number')  
            grade = grades.text  
            grade = round(float(grade), 2)  
  
        except :  
            grade = np.nan  
  
        movie.loc[movie['MOVIE_NM']==title, '네이버_평점'] = grade  
  
        el = dv.find_element(By.CSS_SELECTOR, 'input#nx_query')  
  
    dv.close()  
    movie.to_csv(file_path, index=False, encoding='utf-8')  
    print(f"네이버 평점 업데이트 완료! {file_path}에 저장되었습니다.")
```


01 데이터 사용 출처

cine21 : <http://www.cine21.com> (영화 평점)

The screenshot shows the cine21 website interface. At the top, there's a navigation bar with links like '기사', '데일리뉴스', '영화', '랭킹', '멀티미디어', '이벤트&커뮤니티', '정기구독', '아카이브', and '영화인 리쿠르트'. Below this, there's a search bar. The main content area has tabs for '영화정보' and '영화별점'. Under '영화별점', there are four columns: '최근영화', '역대 박스오피스', '고별점 영화', and '필자별'. The '최근영화' column shows movie posters for '미드나잇 인 파리' (Midnight in Paris) with a rating of 7.86 and '반지의 정령' (The Lord of the Rings: The Fellowship Ring) with a rating of 9.00. The '고별점 영화' column shows a poster for 'TENORIO JR.' with a rating of 7.00. The '필자별' column shows a poster for '만나러 갈게' (I'll go meet you) with a rating of 5.33.

웹 크롤링

```
def cine21_crawling_grade(grade_non_list, file_path):  
    dv = webdriver.Chrome()  
    dv.get('http://www.cine21.com/')  
    time.sleep(1.5)  
    el = dv.find_element(By.CSS_SELECTOR, 'input.input_search')  
  
    try:  
        movie = pd.read_csv(file_path)  
    except :  
        movie = pd.DataFrame({'MOVIE_NM': movie_title})  
  
    for title in grade_non_list :  
        el.clear()  
        el.send_keys('{}'.format(title))  
        el.send_keys(Keys.ENTER)  
        time.sleep(1.5)  
  
        try :  
            grades = dv.find_element(By.CSS_SELECTOR, 'span.num')  
            grade = grades.text  
            grade = round(float(grade), 2)  
  
        except :  
            grade = np.nan  
  
        movie.loc[movie['MOVIE_NM']==title, '씨네21_평점'] = grade  
  
        el = dv.find_element(By.CSS_SELECTOR, 'input.input_search')  
  
    dv.close()  
    movie.to_csv(file_path, index=False, encoding='utf-8')  
    print(f"씨네21 평점 업데이트 완료! {file_path}에 저장되었습니다.")
```

01 개발환경

OS	Windows 10 Pro
Language	Python 3.10.9
IDE	Anacomda jupyter notebook(데이터정제 및 병합, 그룹화, ML&DL 분석), PyCharm Community 2024.3.1(ML&이 분석 및 웹 구현)
Open Source	Tensorflow 2.10, Pandas 1.5.3, Numpy 1.24.4, Seaborn 0.12.2, Selenium 4.27.1, Sklearn 1.2.1, Matplotlib 3.7.0,
Framework	fastAPI 0.115.7, Jinja2 3.1.5, Python-multipart 0.0.20, uvicorn 0.34.0,

02 자료 정제 및 통합

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6043 entries, 0 to 6042
Data columns (total 18 columns):
#   Column                Non-Null Count  Dtype
---  -
0   NO                     6043 non-null   int64
1   MOVIE_NM               6043 non-null   object
2   DRCTR_NM              5173 non-null   object
3   MAKR_NM               2364 non-null   object
4   INCME_CMPNY_NM        3197 non-null   object
5   DISTB_CMPNY_NM        6043 non-null   object
6   OPN_DE                 6041 non-null   object
7   MOVIE_TY_NM           6043 non-null   object
8   MOVIE_STLE_NM          6043 non-null   object
9   NLTY_NM               6043 non-null   object
10  TOT_SCRN_CO            5638 non-null   object
11  SALES_PRICE            2053 non-null   object
12  VIEWNG_NMPR_CO         4188 non-null   object
13  SEOUL_SALES_PRICE      2612 non-null   object
14  SEOUL_VIEWNG_NMPR_CO   4548 non-null   object
15  GENRE_NM              6029 non-null   object
16  GRAD_NM               6043 non-null   object
17  MOVIE_SDIV_NM          6043 non-null   object
dtypes: int64(1), object(17)
memory usage: 849.9+ KB
```

데이터 전처리 전 6043 rows

• 데이터 컬럼 삭제 및 컬럼명 변경

불필요한 컬럼 삭제

```
movies =
movie.drop(columns={'NO','DRCTR_NM','MAKR_NM','INCME_CMPNY_NM','MOVIE_TY_NM','MOVIE_STL
E_NM','NLTY_NM','SALES_PRICE',
'SEOUL_SALES_PRICE','SEOUL_VIEWNG_NMPR_CO','MOVIE_SDIV_NM'})
```

컬럼명 변경

```
movies = movies.rename(columns=
{'MOVIE_NM':'영화제목', 'DISTB_CMPNY_NM':'유통회사명', 'OPN_DE':'개봉일',
'TOT_SCRN_CO':'총스크린수', 'VIEWNG_NMPR_CO':'관람객수',
'GENRE_NM':'장르', 'GRAD_NM':'등급', '네이버_평점':'네이버_평점',
'씨네21_평점':'씨네21_평점'})
```


02 자료 정제 및 통합

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6043 entries, 0 to 6042
Data columns (total 9 columns):
#   Column      Non-Null Count  Dtype
---  -
0   영화제목    6043 non-null  object
1   유통회사명   6043 non-null  object
2   개봉일      6041 non-null  object
3   총스크린수   5638 non-null  object
4   관람객수     4188 non-null  object
5   장르        6041 non-null  object
6   등급        6043 non-null  object
7   네이버_평점  5479 non-null  float64
8   씨네21_평점  4731 non-null  float64
dtypes: float64(2), object(7)
memory usage: 425.0+ KB
```

- 데이터 결측치 처리

- 총스크린수 결측치 처리 (median)

```
movies['총스크린수'] = movies['총스크린수'].str.replace(',','').astype(np.float32)
median_screen_count = movies['총스크린수'].median()
```

- 관람객수, 네이버_평점, 씨네21_평점 결측치 처리 (mean)

```
movies_visitors = movies.groupby(['장르', '개봉월'])['관람객수'].mean().unstack()
naver_mean = movies.groupby(['장르', '개봉월'])['네이버_평점'].mean().unstack()
cine_mean = movies.groupby(['장르', '개봉월'])['씨네21_평점'].mean().unstack()
```

- 네이버_평점, 씨네21_평점 기준 평균 평점 생성

```
movies_mean['평균평점'] = (movies_mean['네이버_평점'] + movies_mean['씨네21_평점']) / 2
```

02 자료 정제 및 통합

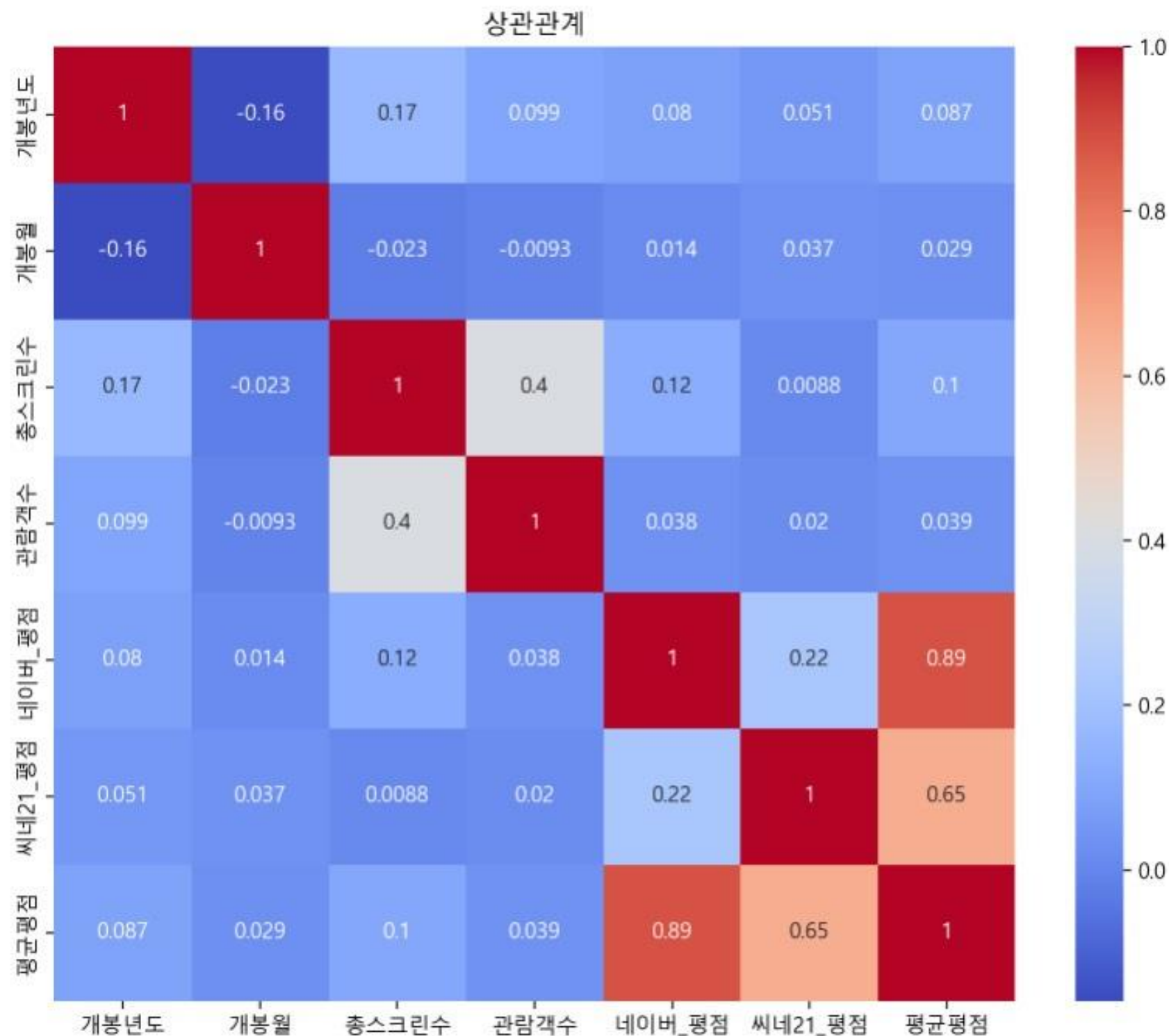
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5999 entries, 0 to 6042
Data columns (total 10 columns):
#   Column      Non-Null Count  Dtype
---  -
0   영화제목    5999 non-null  object
1   유통회사명   5999 non-null  object
2   개봉년도    5999 non-null  object
3   개봉월      5999 non-null  object
4   총스크린수  5999 non-null  float64
5   관람객수    5999 non-null  float64
6   장르        5999 non-null  object
7   등급        5999 non-null  object
8   네이버_평점 5999 non-null  float64
9   씨네21_평점 5999 non-null  float64
dtypes: float64(4), object(6)
memory usage: 515.5+ KB
```

	영화제목	유통회사명	개봉년 도	개봉 월	총스크린 수	관람객수	장르	등급	네이버_평 점	씨네21_평 점	평균평점
0	소울	월트디즈니컴퍼니코리아 유 한책임회사	2021	1	2018.0	875001.0	애니메이 션	전체관람가	9.320000	8.500000	8.910000
1	극장판 귀멸의 칼날: 무한열차편	위터홀컴퍼니(주)	2021	1	380.0	206309.0	애니메이 션	15세이상관 람가	8.392941	6.000000	7.196471
2	원더 우먼 1984	워너브러더스 코리아(주)	2020	12	2260.0	155562.0	액션	12세이상관 람가	7.540000	5.500000	6.520000
3	세자매	(주)리틀빅픽쳐스	2021	1	569.0	42290.0	드라마	15세이상관 람가	8.950000	5.000000	6.975000
4	명탐정 코난: 진홍의 수학여행	(주)씨제이이엔엠	2021	1	532.0	38131.0	애니메이 션	12세이상관 람가	8.020000	5.000000	6.510000
...
5994	뉴클레식 프로젝트 미안하다, 사랑 한다	씨제이 씨지브이(CJ CGV) (주)	2024	11	7.0	676.0	멜로/로 맨스	15세이상관 람가	8.330000	5.606905	6.968452
5995	극장판 블루 록 -에피소드 나가-	씨제이 씨지브이(CJ CGV) (주)	2024	8	276.0	674.0	애니메이 션	12세이상관 람가	10.000000	5.751522	7.875761
5996	우리는 천국에 갈 순 없지만 사랑 은 할 수 있겠지	(주)메리크리스마스	2024	10	87.0	652.0	드라마	15세이상관 람가	8.060000	6.000000	7.030000
5997	딸에 대하여	찬란,스튜디오 에이드	2024	9	106.0	609.0	드라마	12세이상관 람가	7.060000	7.000000	7.030000
5998	퍼펙트 데이즈	(주)티캐스트	2024	7	137.0	582.0	드라마	12세이상관 람가	8.470000	6.860000	7.665000

5999 rows × 11 columns

데이터 전처리 후 5999 rows

02 가중치 산출을 위한 상관분석 및 그룹화, 시각화

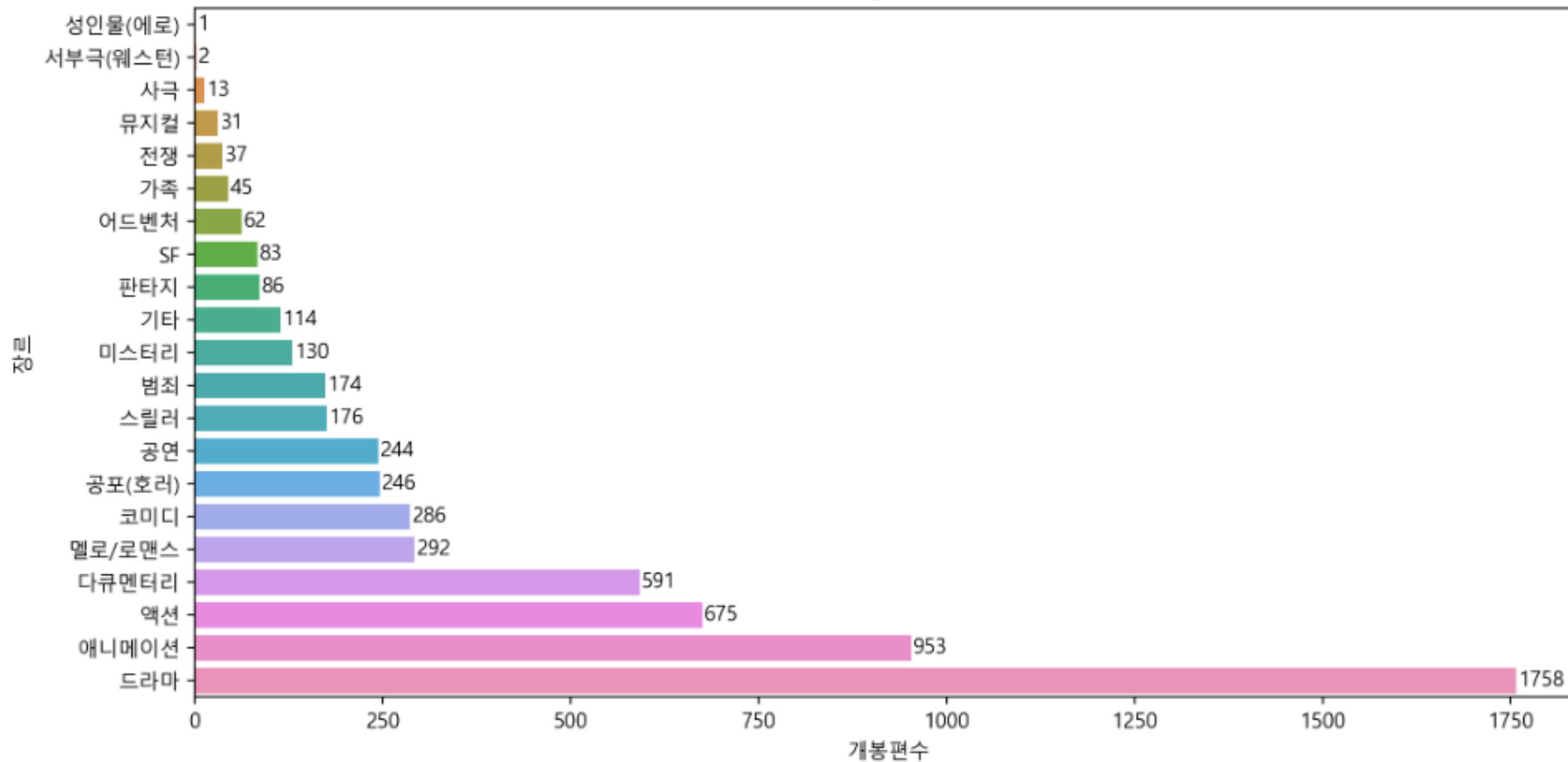


Heatmap 을 사용하여 상관관계 분석

스크린 수와 관객 수의 상관계수는 0.4로,
스크린 수 증가가 관객 수 증가에 기여함.

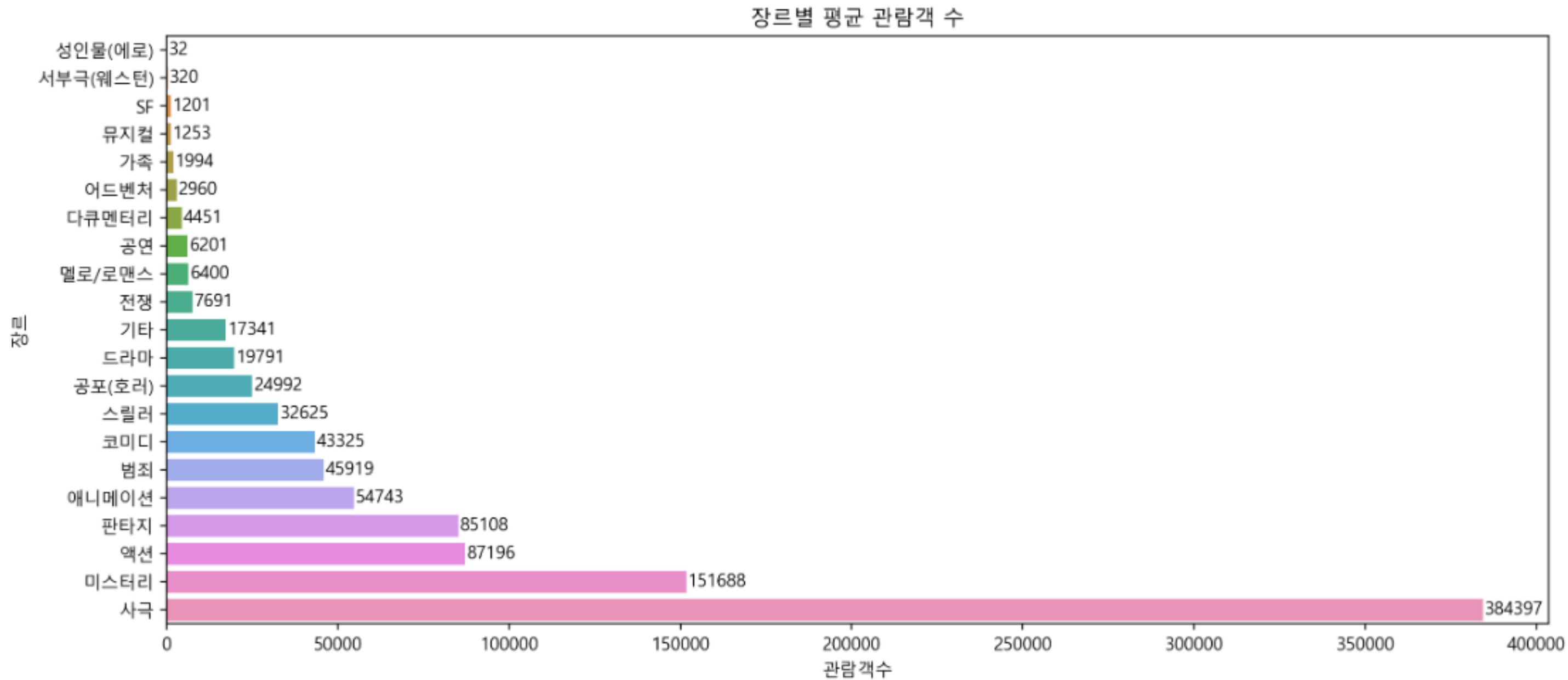
02 가중치 산출을 위한 상관분석 및 그룹화, 시각화

21~24년도 장르별 개봉편수



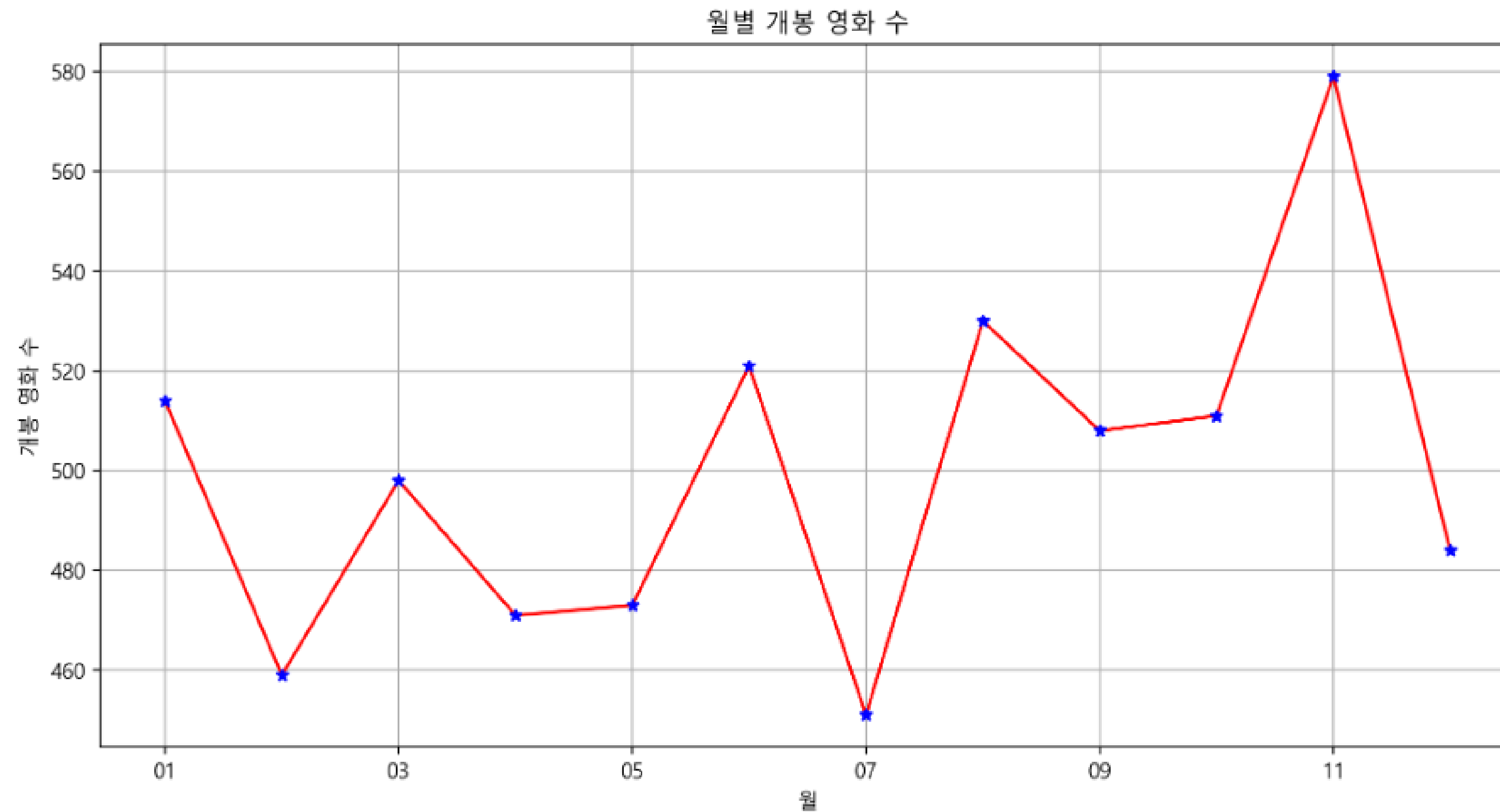
드라마와 애니메이션이 개봉작 수
상위를 차지하며,
성인영화와 서양영화는 가장 적음.

02 가중치 산출을 위한 상관분석 및 그룹화, 시각화



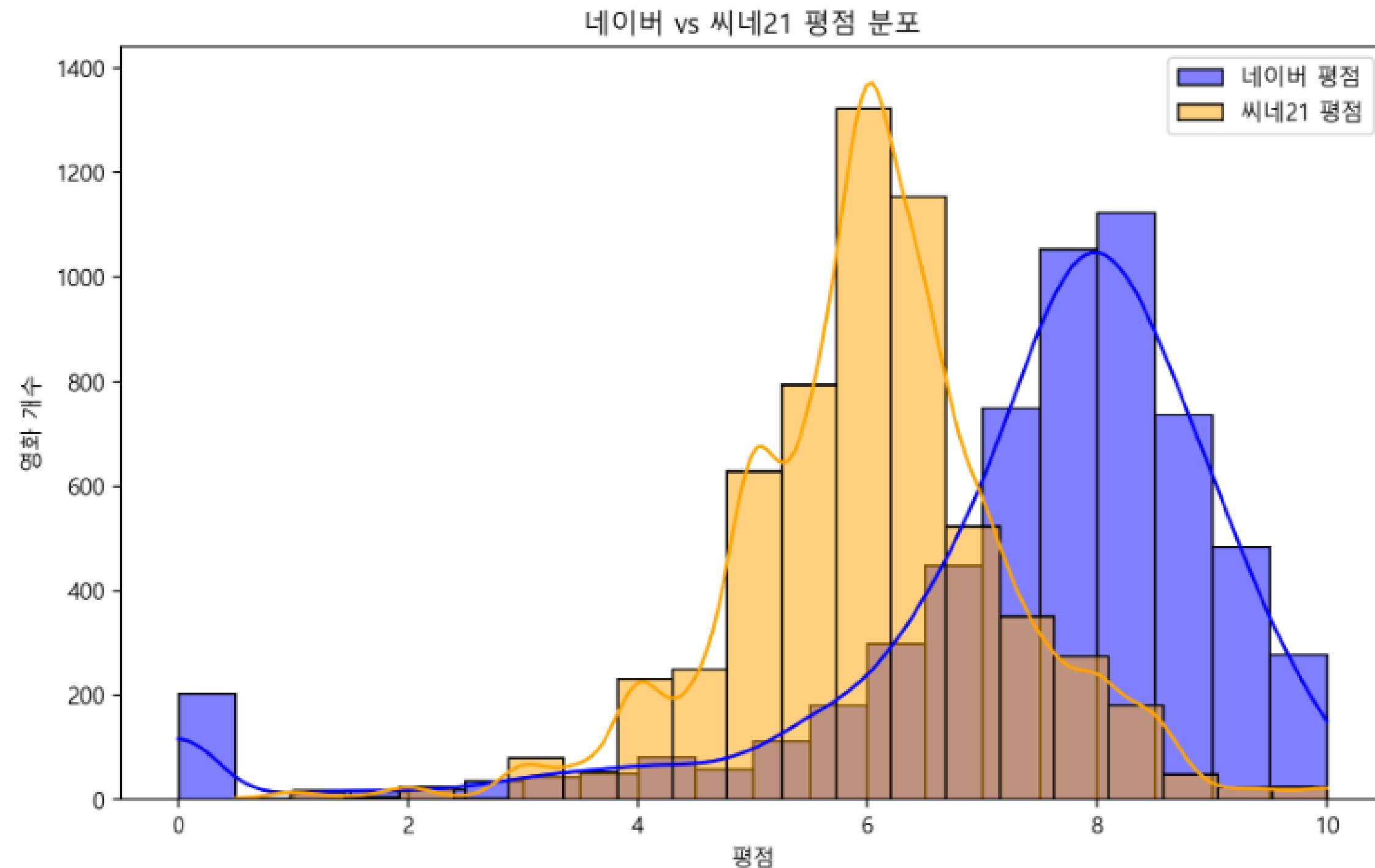
사극 장르는 평균 관객 수가 가장 높아 관객의 관심이 집중되는 장르임.

02 가중치 산출을 위한 상관분석 및 그룹화, 시각화



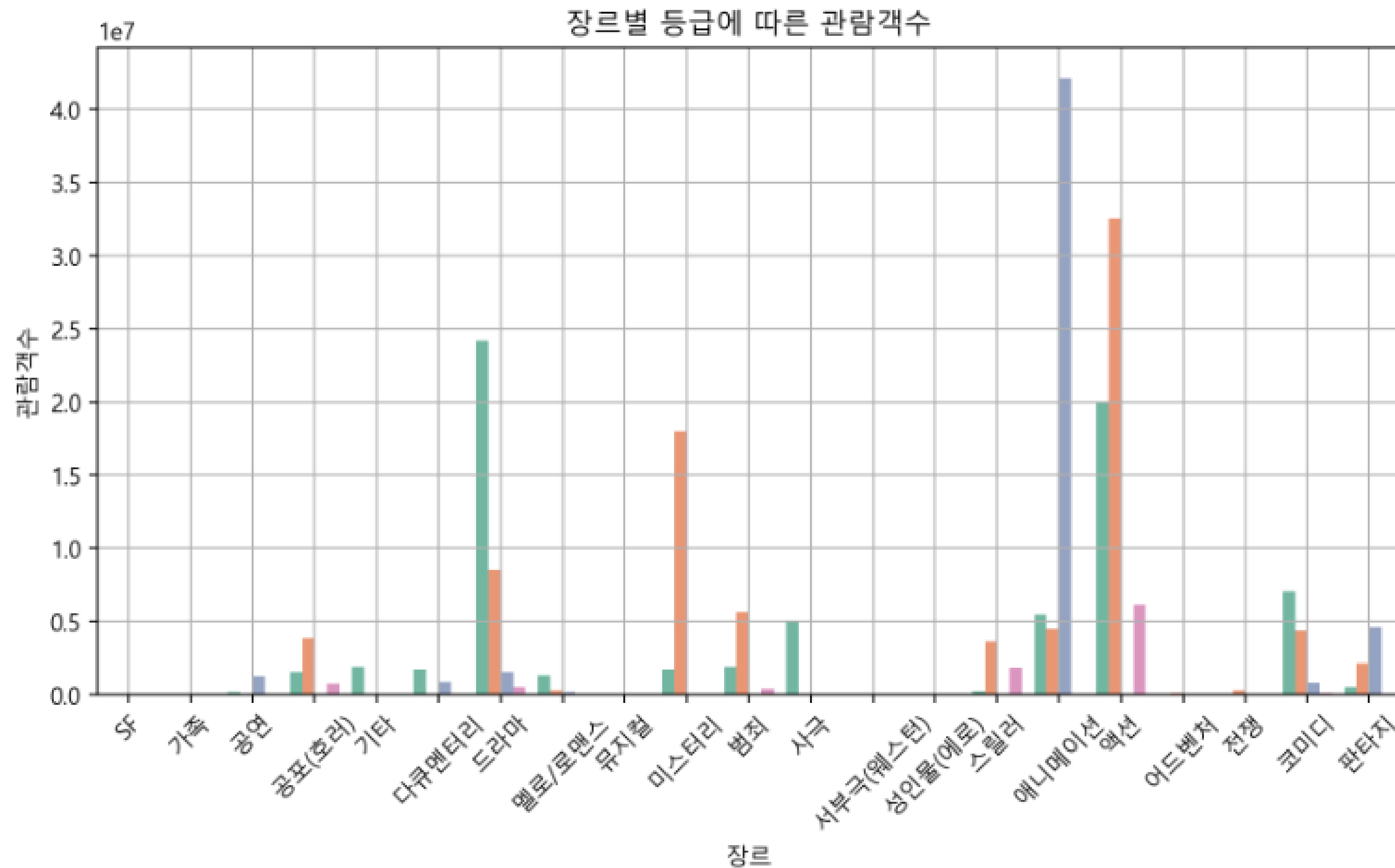
8월과 11월에 개봉 영화가 집중되어 계절적 영향이 보이는듯 함.

02 가중치 산출을 위한 상관분석 및 그룹화, 시각화



씨네21은 네이버보다
평균 평점이 낮으며,
평점은 관객 수 증가에 일부 영향을 미침

02 가중치 산출을 위한 상관분석 및 그룹화, 시각화

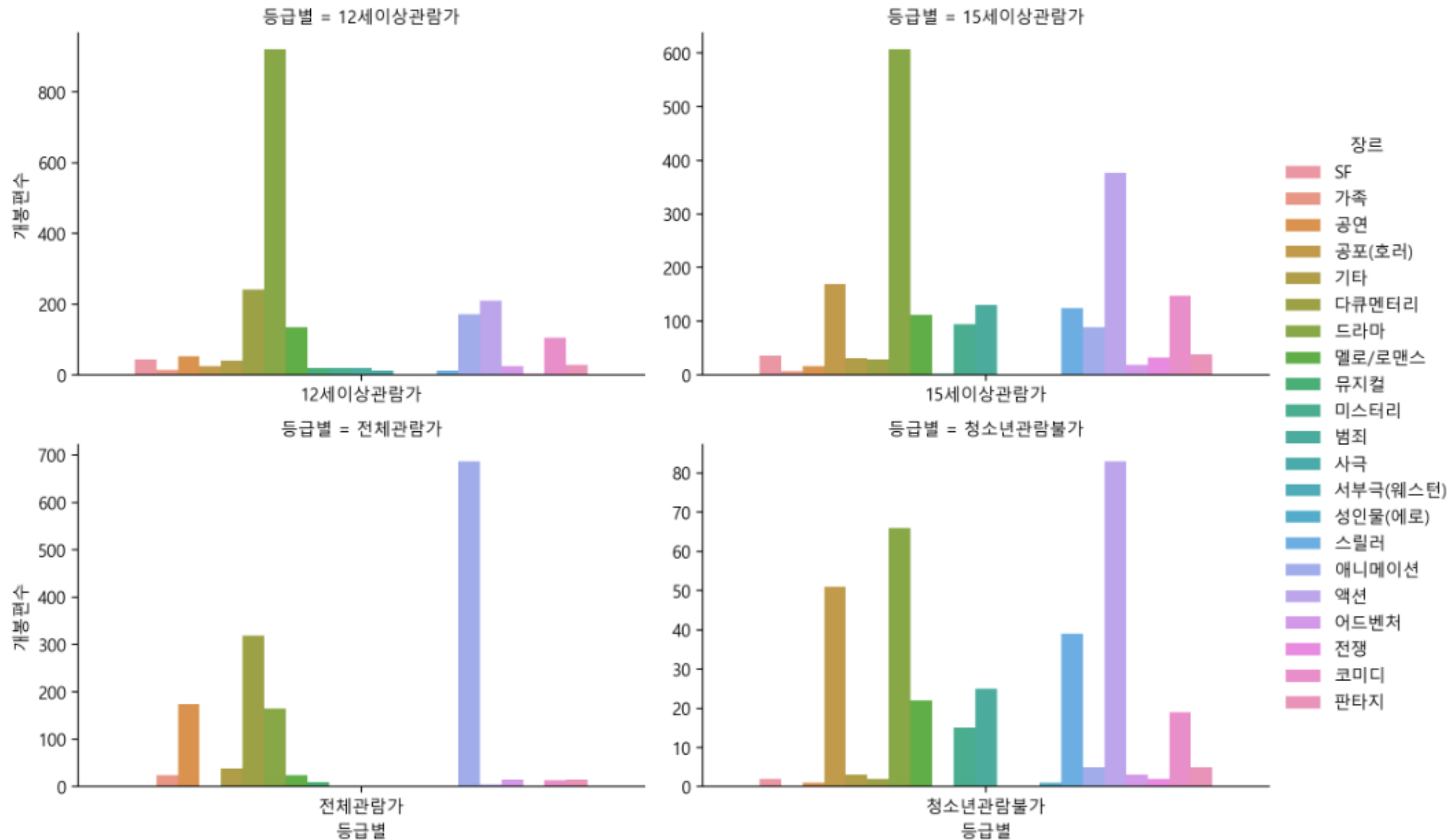


- `groupby(장르, 등급)[관람객]`

```
mog = movies_mean
.groupby(['장르', '등급'])
['관람객수'].sum().reset_index()
sns.barplot
(data=mog, x='장르', y='관람객수', hue='
등급', palette='Set2')
```

02 가중치 산출을 위한 상관분석 및 그룹화, 시각화

등급별 장르 개봉편수 그래프



`groupby(장르, 등급)[개봉월]`

```
movie_grade = movies_mean.groupby(['장르', '등급'])['개봉월']  
.count().reset_index()
```

```
sns.catplot(data=movie_grade,  
x='등급별', y='개봉편수', hue='장르',  
kind='bar', col='등급별',  
sharey=False, sharex=False,  
height=3.5, aspect=1.5,  
col_wrap=2)  
plt.show()
```

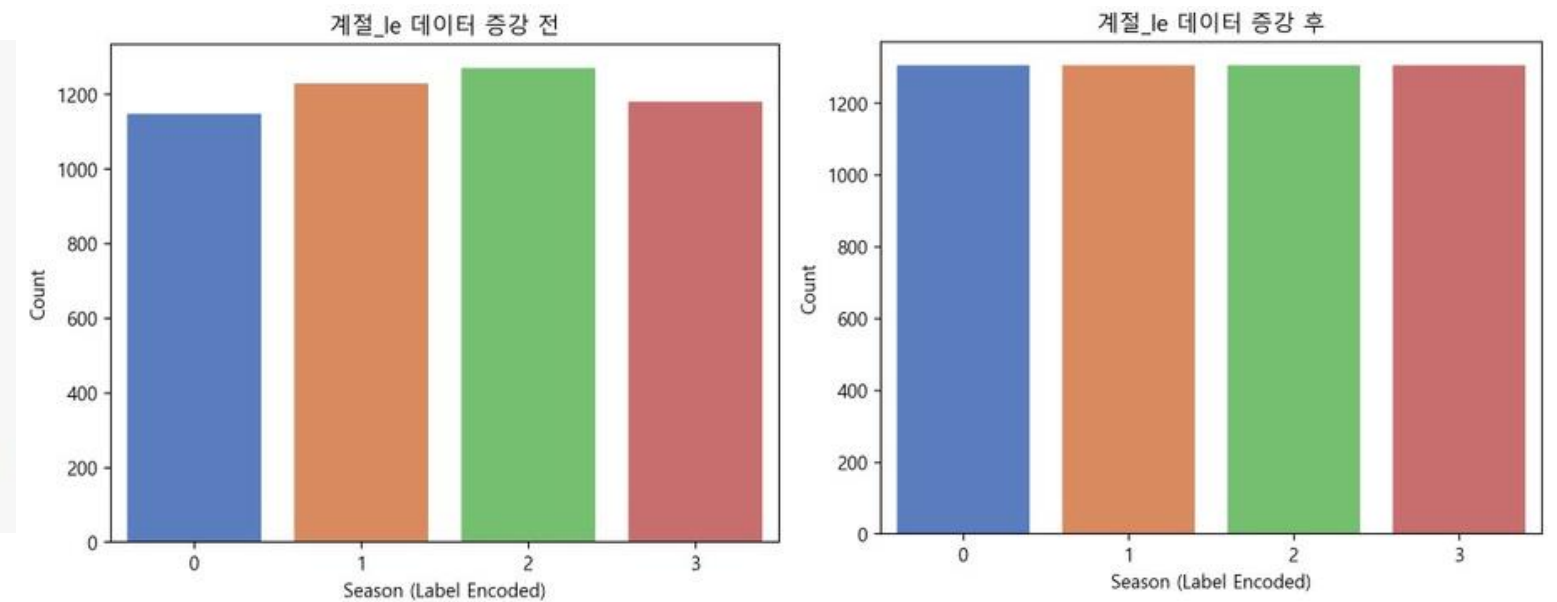
03 Deep Neural Network 분석

```
# 모델준비 (분류분석)
model = Sequential([
    Input(shape=(4,)), # 입력데이터의 column 수
    Dense(64, activation='relu'),
    Dropout(0.2),
    Dense(256, activation='relu'),
    Dropout(0.3),
    Dense(128, activation='relu'),
    Dense(32, activation='relu'),
    Dense(len(movies_mean['월_계절'].unique()), activation='softmax') # 계절의 개수만큼 출력층 생성
])
```

38/38 [=====] - 0s 4ms/step - loss: 1.3657 - accuracy: 0.3925
accuracy : 39.25 %

```
# 데이터 증강
smote = SMOTE(random_state=38, k_neighbors=2)
X_train_smote, y_train_smote = smote.fit_resample(X_train, y_train)
```

38/38 [=====] - 0s 3ms/step - loss: 1.3732 - accuracy: 0.4042
accuracy : 40.42 %



계절 예측 DNN

X = [장르, 총스크린수, 관람객수, 등급]
y = 계절

초기 데이터로 학습 시 정확도 : 39.25%

SMOTE를 활용해 데이터 증폭 후 : 40.42%

03 Deep Neural Network 분석

관객수 등급 예측 DNN

Model: "sequential_11"

Layer (type)	Output Shape	Param #
dense_55 (Dense)	(None, 128)	640
dropout_22 (Dropout)	(None, 128)	0
dense_56 (Dense)	(None, 256)	33024
dense_57 (Dense)	(None, 64)	16448
dropout_23 (Dropout)	(None, 64)	0
dense_58 (Dense)	(None, 64)	4160
dense_59 (Dense)	(None, 4)	260

=====
Total params: 54,532
Trainable params: 54,532
Non-trainable params: 0
=====

관객수 등급 예측 DNN

독립변수 = 장르, 총스크린수, 계절, 등급
종속변수 = 관객수_등급

전체 관객수 3등급으로 분할 학습 진행
전체 관객수 5등급으로 분할 학습 진행

03 Deep Neural Network 분석

관객수 3등급으로 분할 학습 결과

X = [장르, 총스크린수, 관람객수, 등급]
y =관객수_등급

3등급 데이터로 학습 시 정확도 : 56.33%

38/38 [=====] - 0s 3ms/step - loss: 0.9680 - accuracy: 0.5633
accuracy : 56.33 %

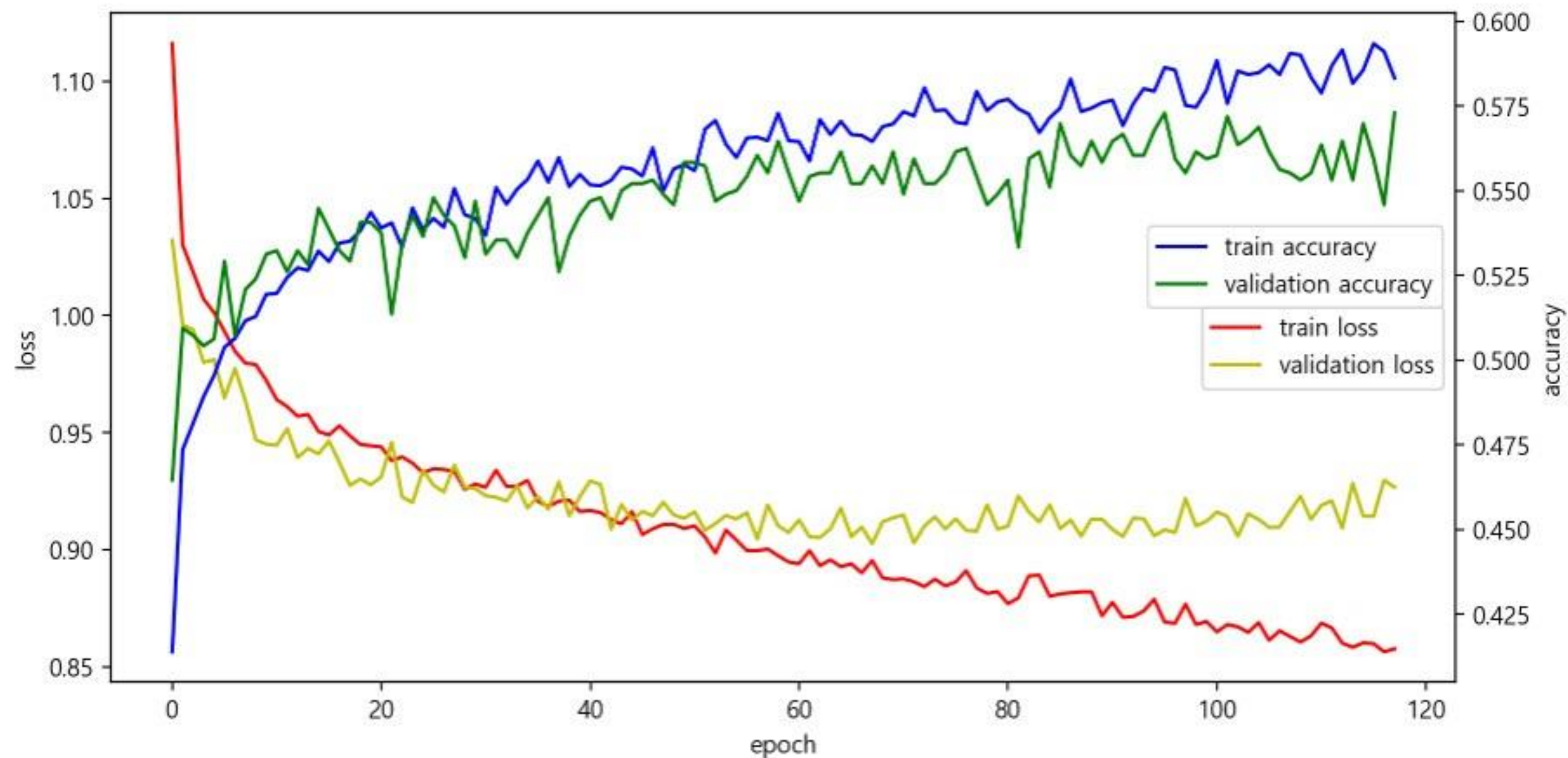
f1 score 결과 값	f1 score: 0.5578330983980071																				
분류분석 성능 지표	<table><tr><td>col_0</td><td>1</td><td>2</td><td>3</td></tr><tr><td>row_0</td><td></td><td></td><td></td></tr><tr><td>1</td><td>174</td><td>117</td><td>75</td></tr><tr><td>2</td><td>97</td><td>204</td><td>124</td></tr><tr><td>3</td><td>27</td><td>84</td><td>298</td></tr></table>	col_0	1	2	3	row_0				1	174	117	75	2	97	204	124	3	27	84	298
col_0	1	2	3																		
row_0																					
1	174	117	75																		
2	97	204	124																		
3	27	84	298																		

03 Deep Neural Network 분석

관객수 3등급으로 분할 학습 결과

X = [장르, 총스크린수, 관람객수, 등급]
 y = 관객수_등급

3등급 데이터로 학습 시 정확도 : **56.33%**



03 Deep Neural Network 분석

관객수 5등급으로 분할 학습 결과

X = [장르, 총스크린수, 관람객수, 등급]
y = 관객수_등급

5등급 데이터로 학습 시 정확도 : **41.50%**

38/38 [=====] - 0s 4ms/step - loss: 1.4269 - accuracy: 0.4150
accuracy : 41.50 %

f1 score 결과 값	f1 score: 0.3957703358858291																																										
분류분석 성능 지표	<table><tr><th>col_0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th></tr><tr><th>row_0</th><td></td><td></td><td></td><td></td><td></td></tr><tr><td>1</td><td>82</td><td>21</td><td>55</td><td>38</td><td>23</td></tr><tr><td>2</td><td>56</td><td>33</td><td>52</td><td>56</td><td>31</td></tr><tr><td>3</td><td>34</td><td>24</td><td>92</td><td>67</td><td>48</td></tr><tr><td>4</td><td>13</td><td>11</td><td>46</td><td>129</td><td>52</td></tr><tr><td>5</td><td>6</td><td>4</td><td>19</td><td>46</td><td>162</td></tr></table>	col_0	1	2	3	4	5	row_0						1	82	21	55	38	23	2	56	33	52	56	31	3	34	24	92	67	48	4	13	11	46	129	52	5	6	4	19	46	162
col_0	1	2	3	4	5																																						
row_0																																											
1	82	21	55	38	23																																						
2	56	33	52	56	31																																						
3	34	24	92	67	48																																						
4	13	11	46	129	52																																						
5	6	4	19	46	162																																						

04 fastAPI

①

계절 예측기

장르

애니메이션

오픈 스크린 수

예측 관람객 수

관람 등급

전체관람가

예측하기

②

예측 결과

입력된 값

장르

공포(호러)

스크린 수

500

관람객 수

50000

관람 등급

12세관람가

예측된 계절

2: 여름

다시 예측하기

① 독립변수

장르 : select option으로 20개의 장르가 나열

오픈 스크린 수 : 영화 개봉전 스크린수 (예상)

예측 관람객 수 : 손익분기점 기준으로 (예상)

관람 등급 : select option으로 4개의 등급이 나열

② 종속변수

예측된 계절 : 독립변수 입력 기준으로 영화 개봉
계절 예측

05 결론

연구 결과

관객 수와 스크린 수의 관계	스크린 수와 관객 수 사이의 상관계수 0.4 양의 상관관계
	스크린 수가 증가할수록 관객 수가 증가
특정 장르 분석	드라마와 애니메이션은 개봉작 수가 가장 많아 대중적인 장르로 보임
	사극 장르는 개봉작 수는 적지만 평균 관객 수가 높음
개봉 시기 트렌드	여름 (7~8월) 시즌에는 관객 수 증가 경향이 보임
	11월과 연말에 개봉작이 집중

05 결론

시사점 및 개선방안

데이터 기반 전략 수립	계절별 트렌드와 장르 선호도를 분석하여 맞춤형 영화 제작 및 마케팅 전략 마련
	관객 집중도가 높은 장르 (사극)와 시기(11월, 연말)를 활용한 배급 전략 최적화
딥러닝 결과 활용	딥러닝 예측 결과를 기반으로 개봉 시기와 장르 선정에 대한 의사 결정 강화
	데이터 증폭을 통해 모델의 정확도와 신뢰도 개선
추가 연구 및 개선 방안	데이터 양 확대 및 새로운 변수 생성으로 예측 정확도를 높이는 연구 필요
	트렌드 영화 ,지역별 영화 관람 패턴 등 외부 데이터를 결합하여 보다 풍부한 인사이트 도출 가능

THANK YOU

감사합니다