

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 ▼

🔍

영화정보검색
박스오피스
테마통계
공식통계
온라인상영관 박스오피스
고객센터

박스오피스

월별 박스오피스

☰
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 통계정보**입니다.
 - ②**[공식통계]**코너는 영신위에서 매년 발표하는 "한국영화연감"의 영화별 유행기록을 참고한 것입니다.
 - 한국영화연감(1971~2010) 통계를 기준으로 정리한 것이며, 2011년부터는 통합선산상을 기준으로 일정한 주기(매월, 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)

• 조화기간 ?

2024

▼

05

~

2024

▼

06

▼

• **국적**

--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 two tabs: '영화정보' and '영화별점'. Under '영화별점', there are four columns: '최근영화', '역대 박스오피스', '고별점 영화', and '필자별'. Each column displays movie posters and their corresponding ratings. For example, '미드나잇 인 파리' has a rating of 7.86, and '네티즌' has a rating of 7.73.

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

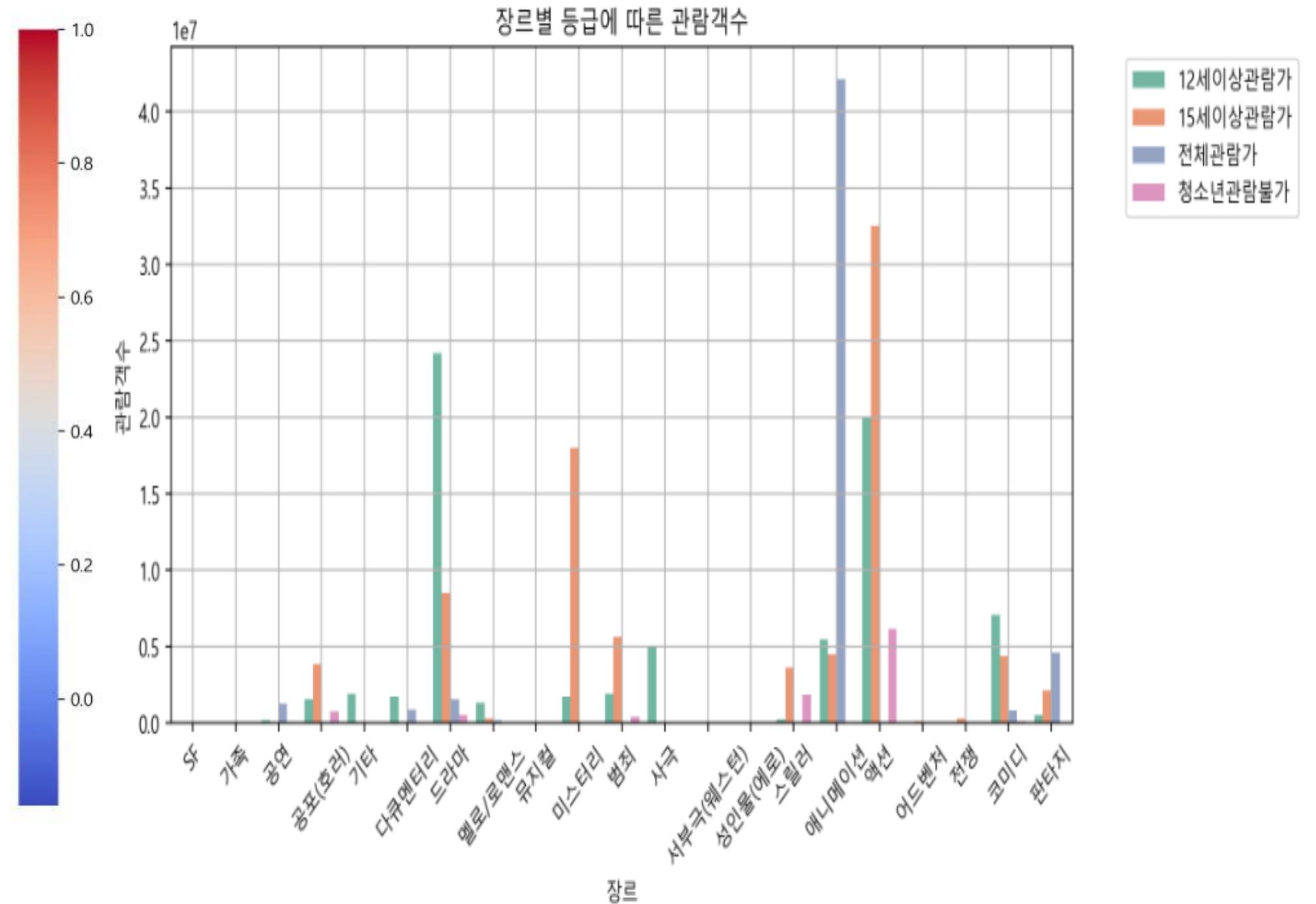
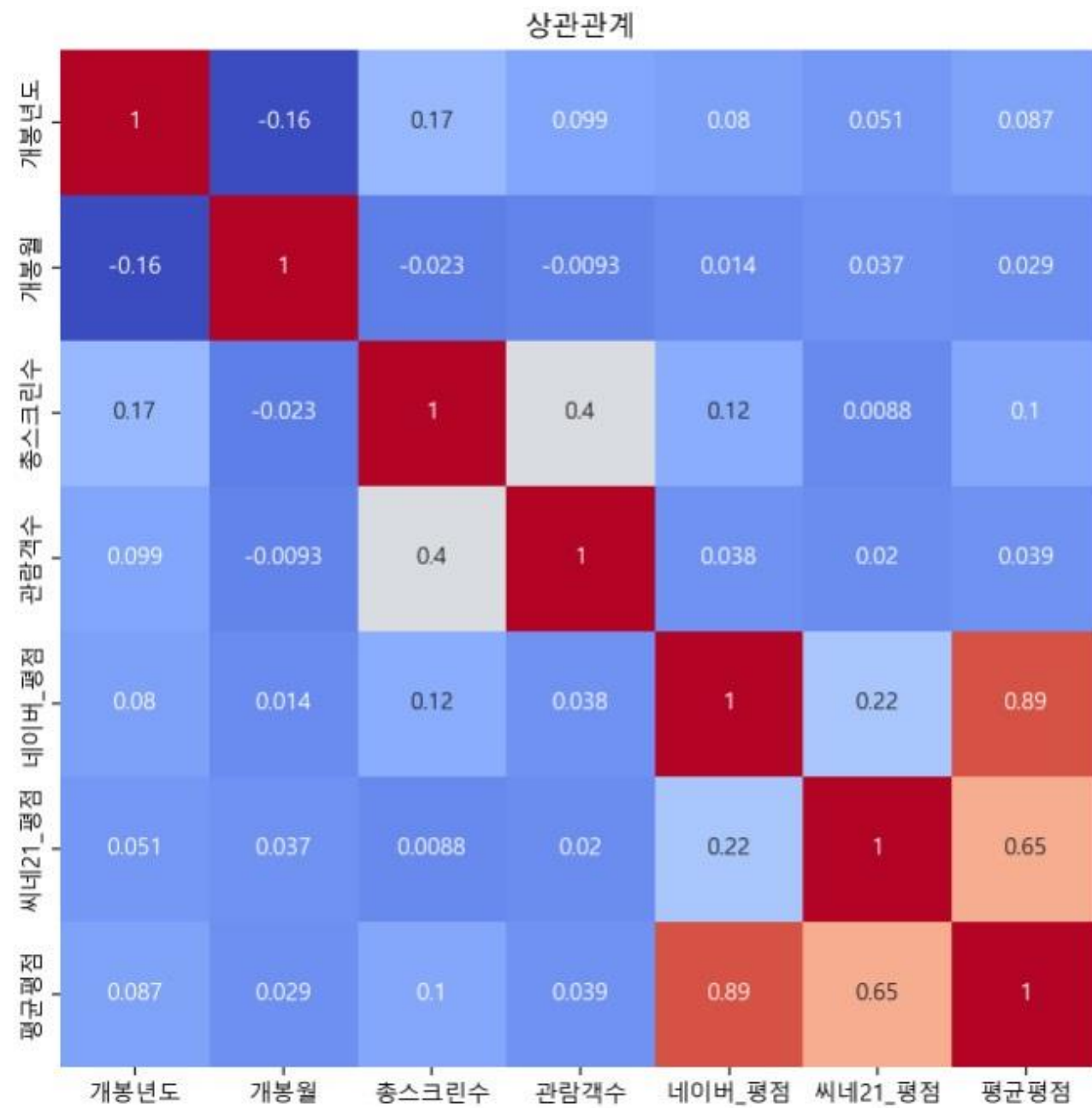
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6043 entries, 0 to 6042
Data columns (total 18 columns):
#   Column                Non-Null Count  Dtype
---  -
0   NO                     6043 non-null   int64
1   MOVIE_NM               6043 non-null   object
2   DRCTR_NM               5173 non-null   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

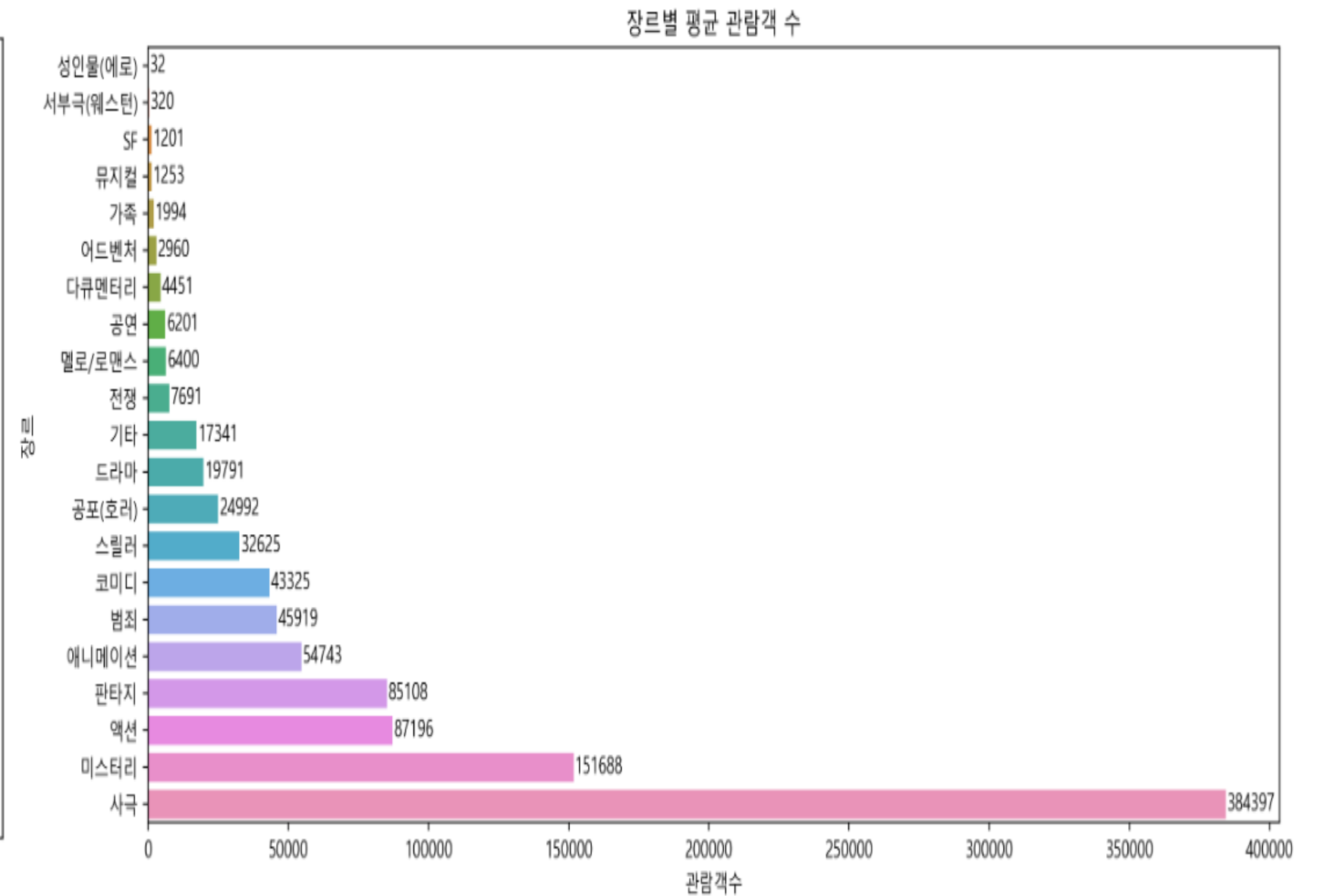
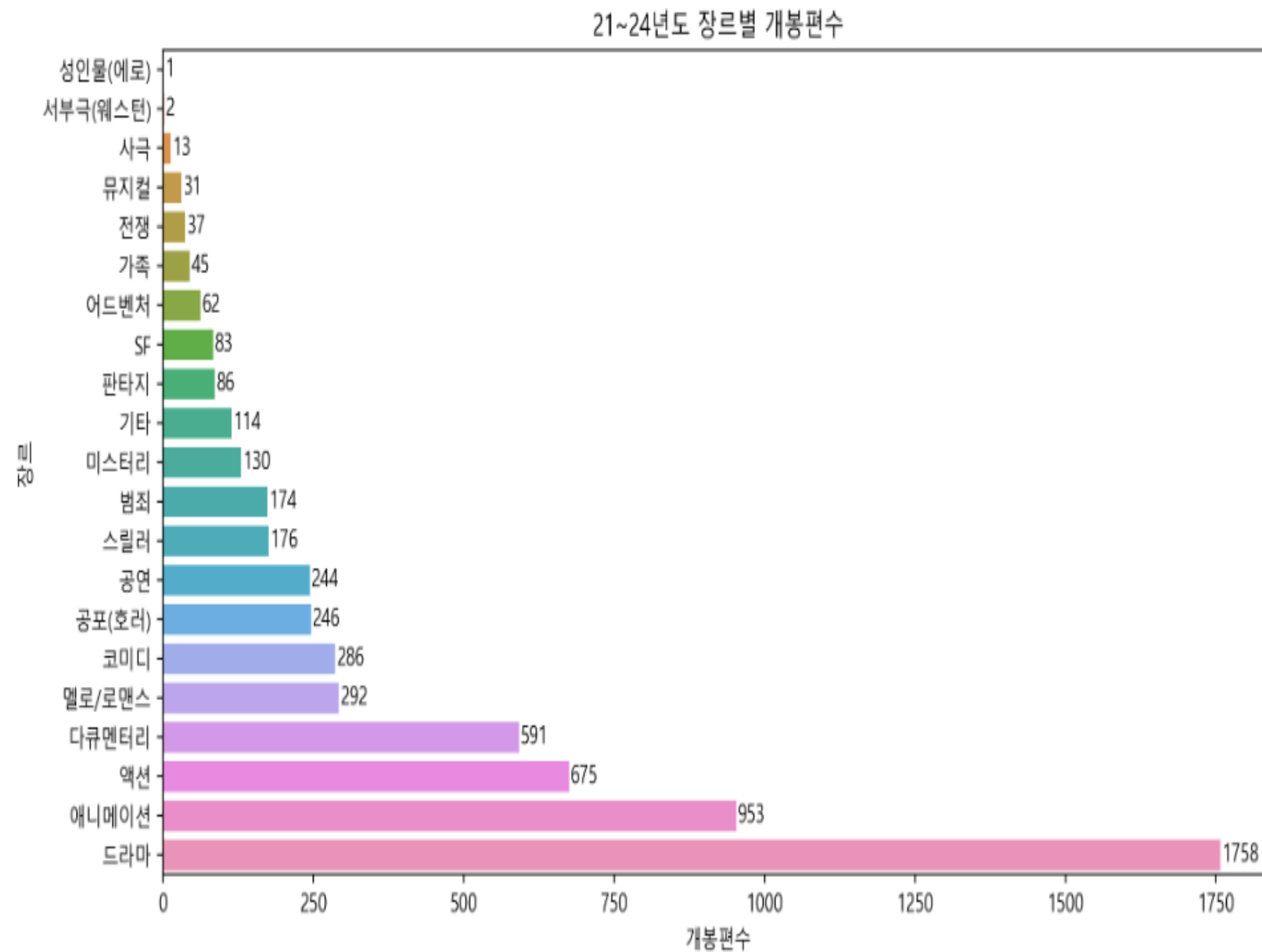
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5999 entries, 0 to 6042
Data columns (total 10 columns):
#   Column                Non-Null Count  Dtype
---  -
0   영화제목              5999 non-null   object
1   유통회사명             5999 non-null   object
2   개봉년도              5999 non-null   object
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
```

데이터 전처리 후 5999 rows

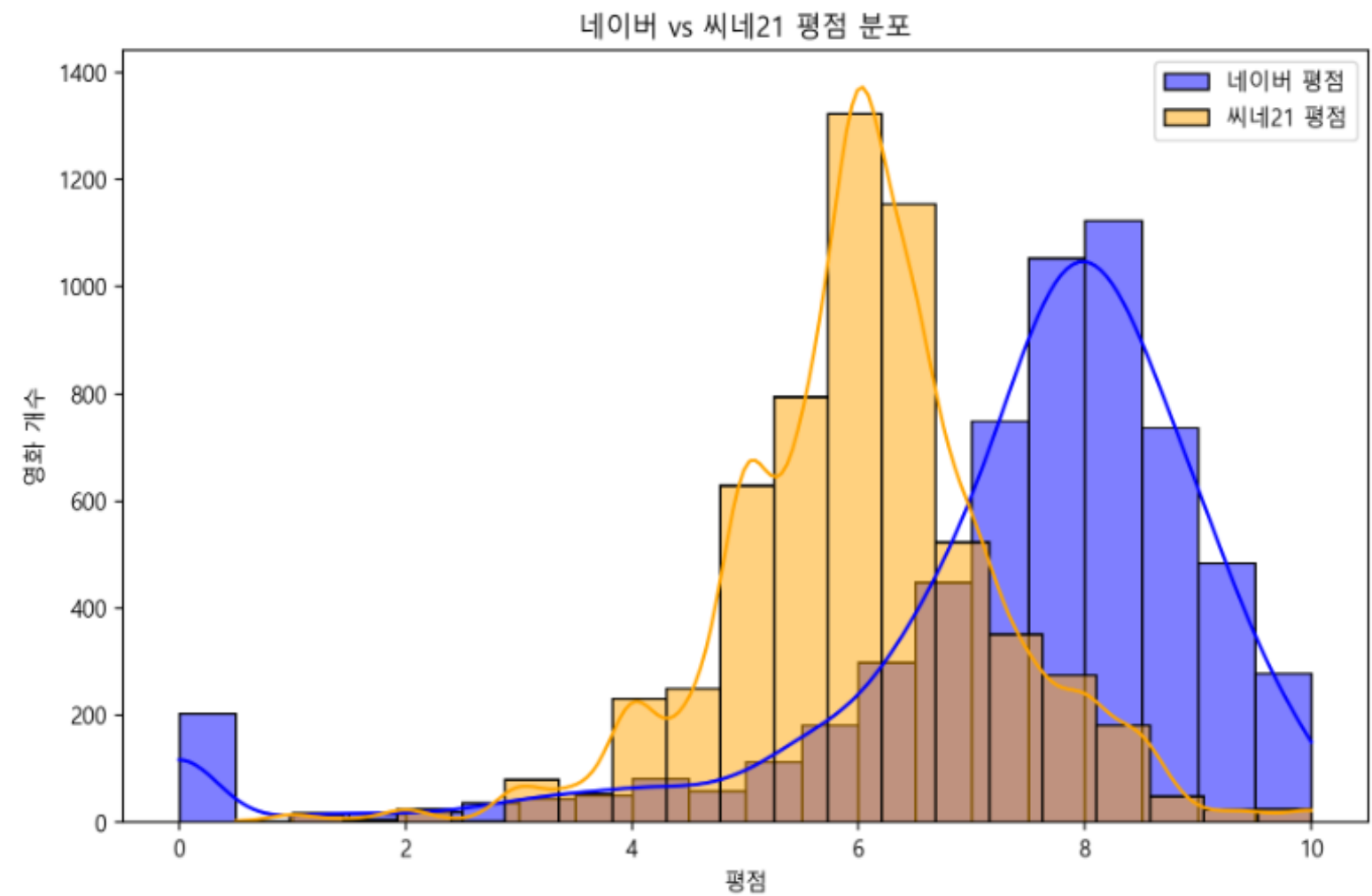
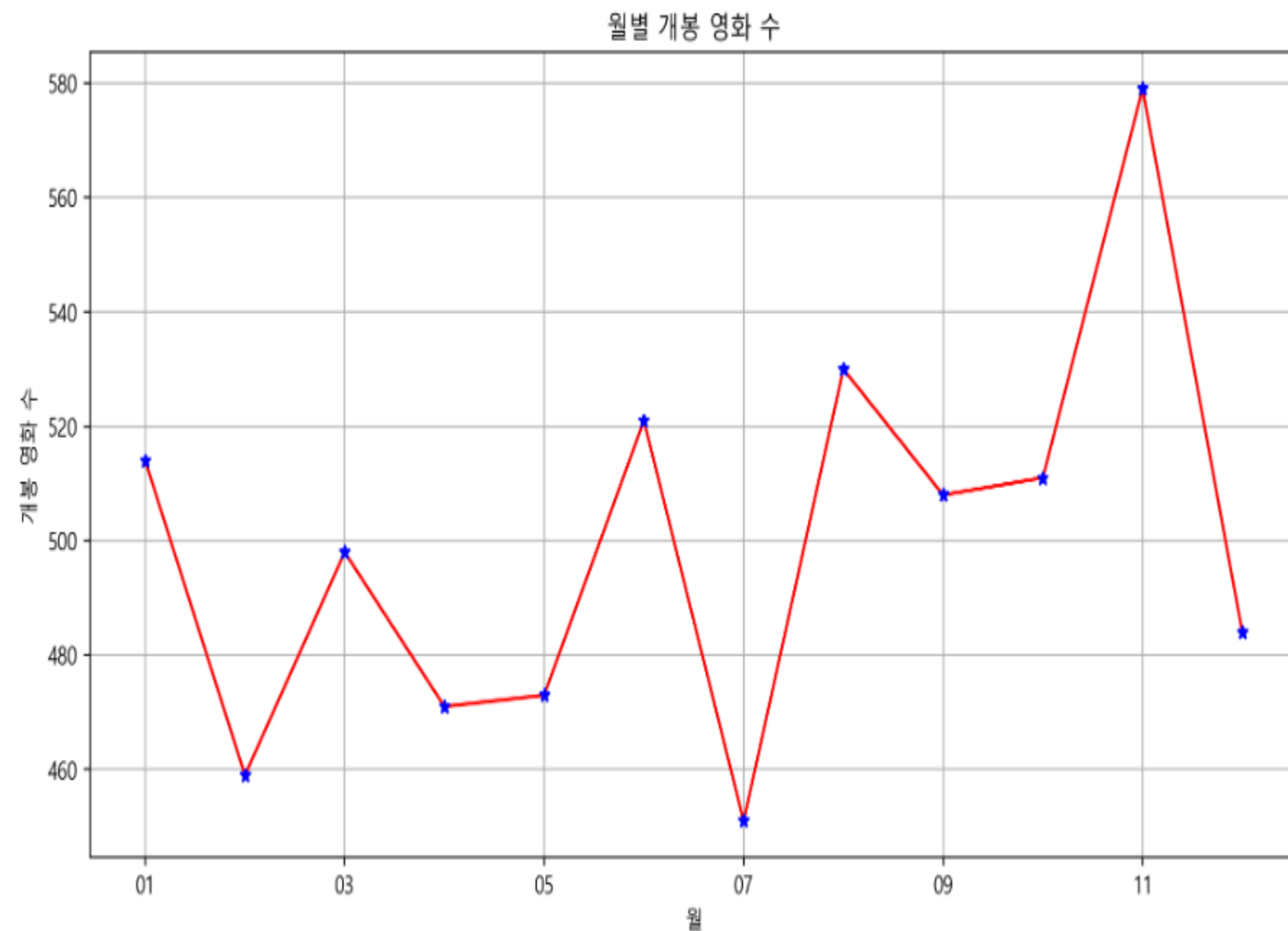
02 가중치 산출을 위한 상관분석 및 그룹화, 시각화



02 가중치 산출을 위한 상관분석 및 그룹화, 시각화

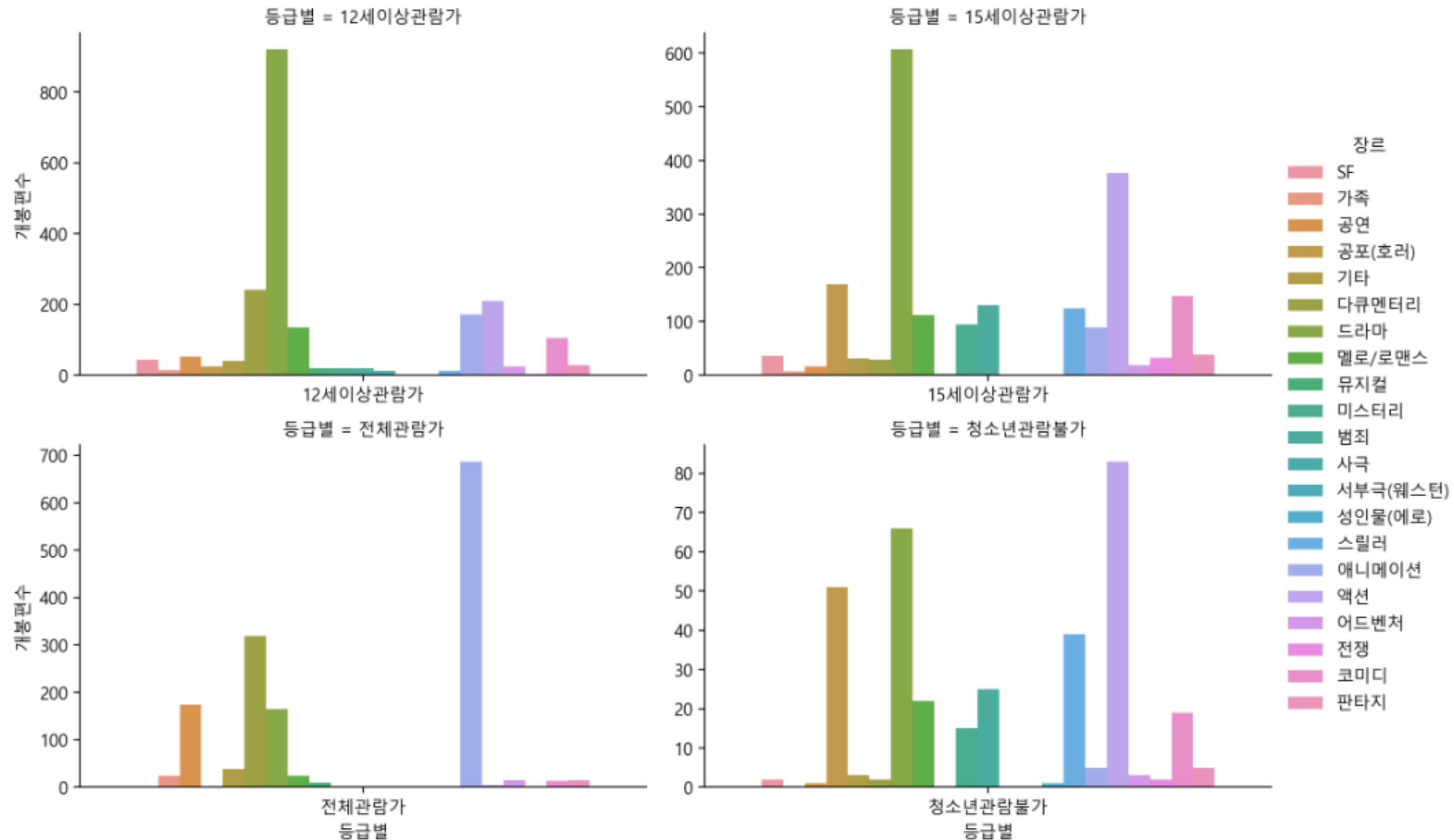


02 가중치 산출을 위한 상관분석 및 그룹화, 시각화

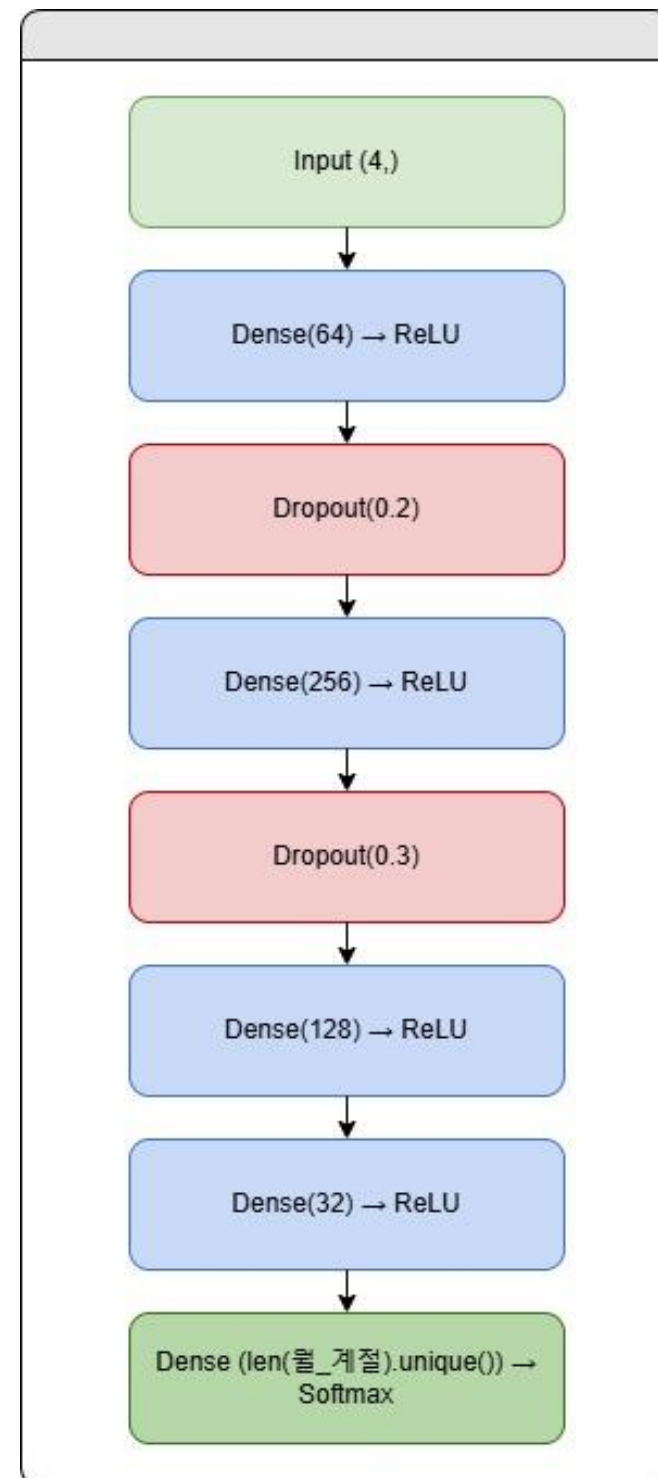


02 가중치 산출을 위한 상관분석 및 그룹화, 시각화

등급별 장르 개봉편수 그래프



03 Deep Neural Network 분석



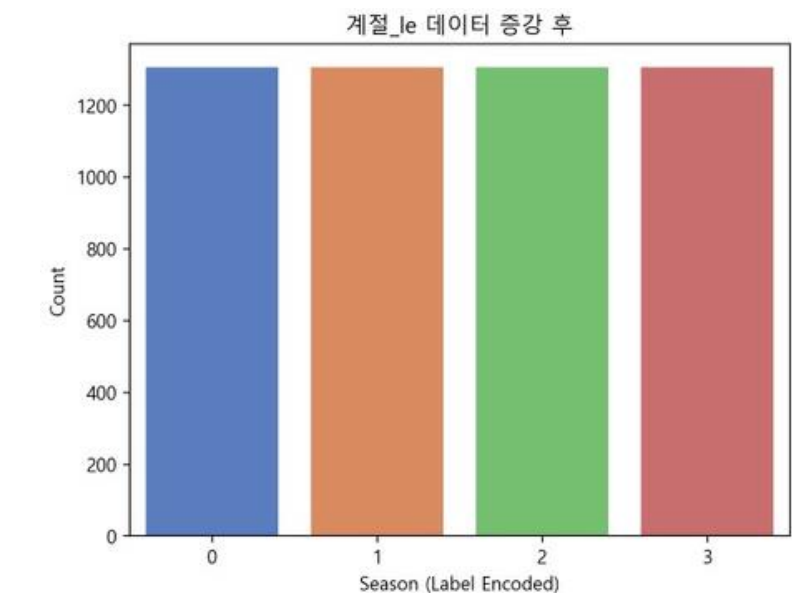
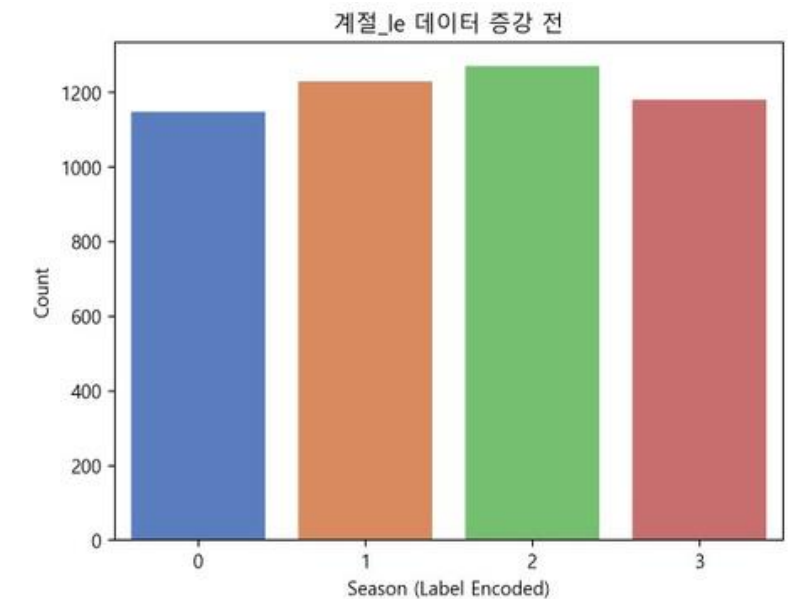
데이터 증강(SMOTE) 전

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)
```

데이터 증강(SMOTE) 후

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



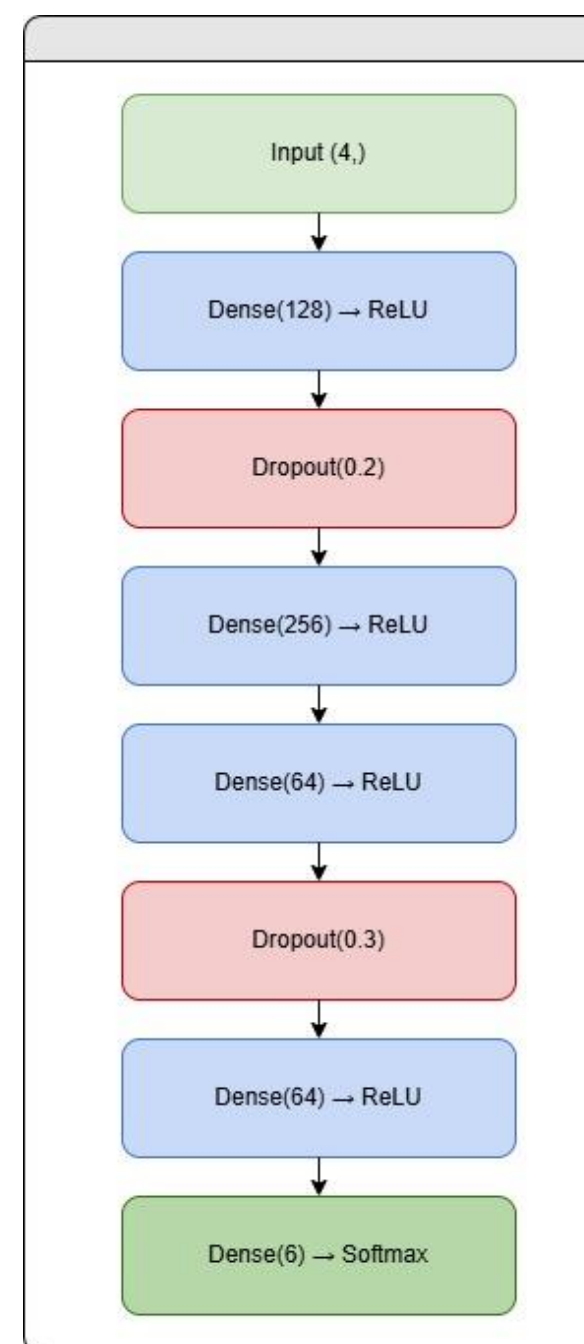
03 Deep Neural Network 분석

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
=====



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 %

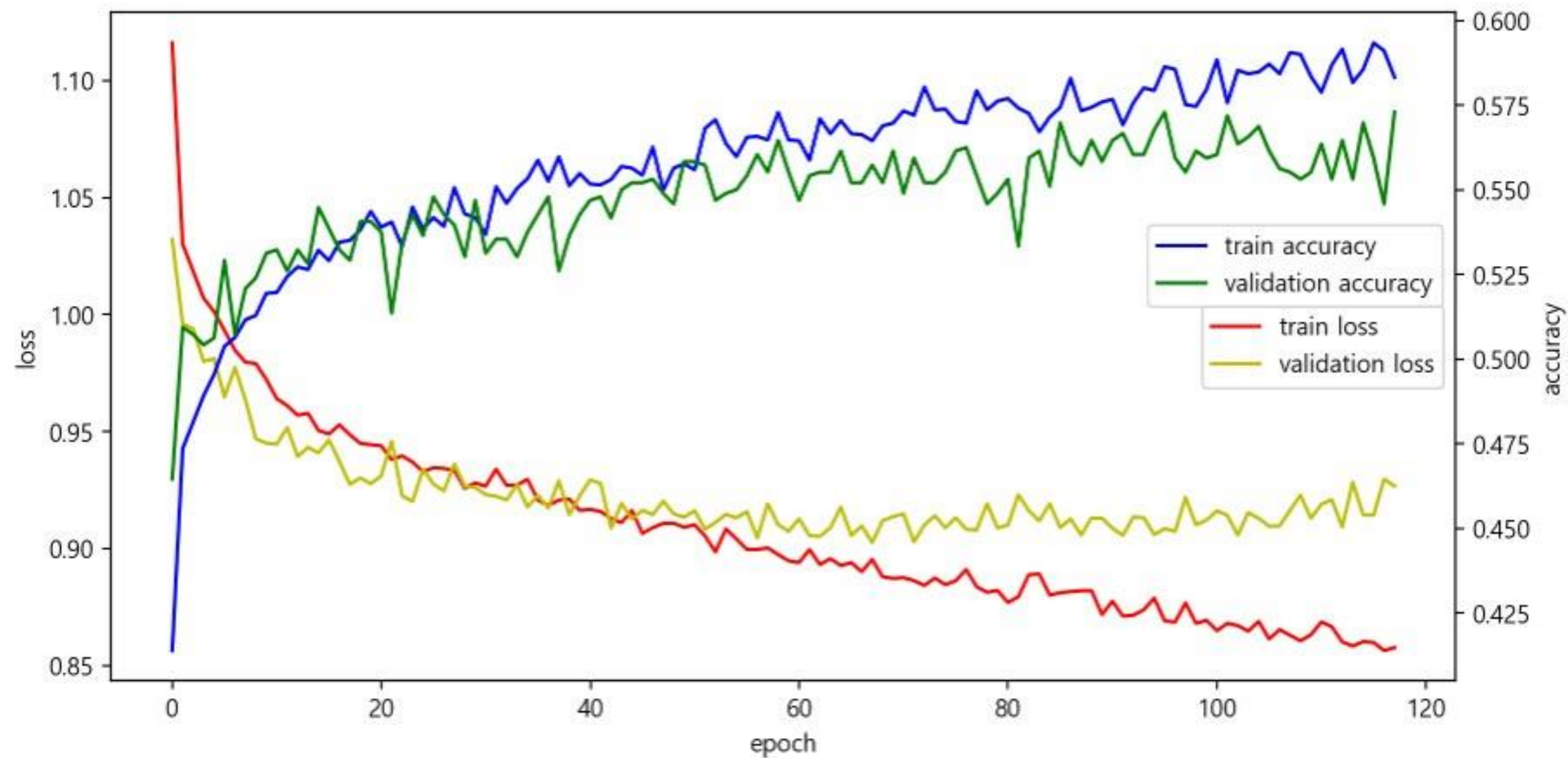
| | | | | | | | | | | | | | | | | | |
|---------------|--|-------|------------|-------|------------|------------|-----|----|-----|-------|----|-----|-----|------------|----|----|-----|
| f1 score 결과 값 | f1 score: 0.5578330983980071 | | | | | | | | | | | | | | | | |
| 분류분석 성능 지표 | <table><tr><td></td><td>낮은 관객 수 그룹</td><td>중간 그룹</td><td>높은 관객 수 그룹</td></tr><tr><td>낮은 관객 수 그룹</td><td>175</td><td>95</td><td>101</td></tr><tr><td>중간 그룹</td><td>84</td><td>203</td><td>140</td></tr><tr><td>높은 관객 수 그룹</td><td>23</td><td>81</td><td>303</td></tr></table> | | 낮은 관객 수 그룹 | 중간 그룹 | 높은 관객 수 그룹 | 낮은 관객 수 그룹 | 175 | 95 | 101 | 중간 그룹 | 84 | 203 | 140 | 높은 관객 수 그룹 | 23 | 81 | 303 |
| | 낮은 관객 수 그룹 | 중간 그룹 | 높은 관객 수 그룹 | | | | | | | | | | | | | | |
| 낮은 관객 수 그룹 | 175 | 95 | 101 | | | | | | | | | | | | | | |
| 중간 그룹 | 84 | 203 | 140 | | | | | | | | | | | | | | |
| 높은 관객 수 그룹 | 23 | 81 | 303 | | | | | | | | | | | | | | |

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></th><th>낮은 관객 수 그룹</th><th>중간 하위 그룹</th><th>중간 그룹</th><th>중간 상위 그룹</th><th>높은 관객 수 그룹</th></tr><tr><td>낮은 관객 수 그룹</td><td>93</td><td>15</td><td>35</td><td>49</td><td>29</td></tr><tr><td>중간 하위 그룹</td><td>65</td><td>33</td><td>37</td><td>53</td><td>53</td></tr><tr><td>중간 그룹</td><td>29</td><td>23</td><td>73</td><td>72</td><td>54</td></tr><tr><td>중간 상위 그룹</td><td>11</td><td>10</td><td>33</td><td>153</td><td>65</td></tr><tr><td>높은 관객 수 그룹</td><td>7</td><td>9</td><td>14</td><td>35</td><td>155</td></tr></table> | | 낮은 관객 수 그룹 | 중간 하위 그룹 | 중간 그룹 | 중간 상위 그룹 | 높은 관객 수 그룹 | 낮은 관객 수 그룹 | 93 | 15 | 35 | 49 | 29 | 중간 하위 그룹 | 65 | 33 | 37 | 53 | 53 | 중간 그룹 | 29 | 23 | 73 | 72 | 54 | 중간 상위 그룹 | 11 | 10 | 33 | 153 | 65 | 높은 관객 수 그룹 | 7 | 9 | 14 | 35 | 155 |
| | 낮은 관객 수 그룹 | 중간 하위 그룹 | 중간 그룹 | 중간 상위 그룹 | 높은 관객 수 그룹 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 낮은 관객 수 그룹 | 93 | 15 | 35 | 49 | 29 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 중간 하위 그룹 | 65 | 33 | 37 | 53 | 53 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 중간 그룹 | 29 | 23 | 73 | 72 | 54 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 중간 상위 그룹 | 11 | 10 | 33 | 153 | 65 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 높은 관객 수 그룹 | 7 | 9 | 14 | 35 | 155 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

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