Big Contest 2017 Report

"Elastic Net을 이용한 개봉예정 영화 관객수 예측"

팀명 : DNA
팀원 : 소민걸 이진규 전민재 홍동준

목 차

1. 개요

(1) 분석 주제

대회 주제는 개봉 예정 및 개봉 영화 3개의 2017년 10월 10일까지 누적관객수를 예측하는 것이다. 주어진 영화는 '킹스맨2 : 골든서클'(2017년 9월 27일 개봉), '넛잡2'(2017년 10월 3일 개봉), '남한산성'(2017년 10월 3일 개봉)이다. 세 영화 모두 '추석 명절'이 상영 기간에 포함되어 있으므로 이를 고려한 세 영화의 누적관객수 예측을 분석 주제로 선정하였다.

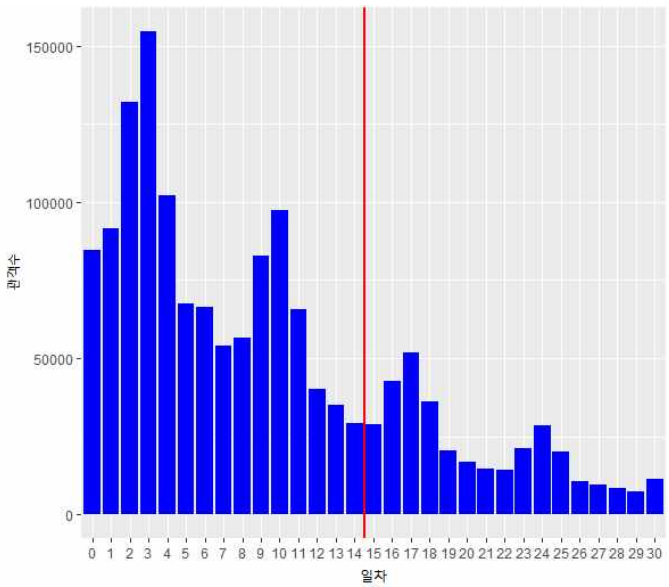
(2) 분석 방향

본 대회의 목표는 개봉 및 개봉 예정인 영화의 누적 관객수 예측이다. 영화 예측에서는 영화 내적 요소와 외적 요소, 영화에 대한 평가 부문으로 나누어 볼 수 있다. 영화진흥위원회(KOFIC)에서 수집한 데이터의 경우 영화 내적(장르, 등급, 배우 등)요소와 영화 외적(배급사, 스크린, 제작사 등) 요소를 포함하고 있다. 이를 바탕으로 데이터 탐색을 통해 요소별 특징을 파악하고, 변수를 생성했다. 또한, 영화의 평가 부문에서는 관람객들의 구전 효과를 반영하기 위해 웹 크롤링을 통해 각 영화별 평점 및 코멘트 데이터를 수집하고, 데이터 탐색을 바탕으로 파생변수를 생성했다. 또한, 잠재 관객들의 영화 관심도를 증가시킬 수 있는 영화 관련 기사 수를 변수로 생성했다. 이에 더해, 예측하고자 하는 10월 10일 경우 9월 30일부터 약 10일간의 연휴를 포함하고 있기 때문에 이를 반영할 수 있는 변수가 필요하다고 판단했다. 그 때문에 데이터 탐색을 통해 평일, 주말, 공휴일등의 특징을 파악하고, 이를 바탕으로 파생변수를 생성했다. 전처리와 데이터 탐색은 R과 Python을 이용하고, 모형 수립은 R을 이용했다.

2. 데이터 전처리

(1) 데이터 정제

대회 측에서 제공한 데이터는 1주 단위로 영화정보가 묶여있기 때문에 일일관객수를 예측하기에 부적합하다고 판단하였다. 따라서 데이터는 영화진흥위원회(KOFIC) 일별 박스오피스 정보를 사용하였다. 최근 영화 동향을 반영하기 위해 상영일이 2013년 1월 1일부터 2017년 9월 28일까지의 데이터로 정제했다. 개봉 전 시사회와 재개봉 영화 데이터는 상영 일차 기준의 명확성을 위해 제거하였다. 최종 누적관객수가 10만 미만인 영화는 전체 영화 매출액의 약 2%를 차지하고 있다. 이는 모형수립에 왜곡을 줄 수 있다고 판단하여 제거하였다. 상영일차별 관객수를 분석한 결과 14일 이후부터 급격하게 감소하는 패턴을 보였다. 따라서 상영일차가 14일 이후인 데이터를 제거하였다. 추가 데이터는 네이버 뉴스 사이트와 다음 영화 사이트에서 웹 크롤링을 통해 가져왔다.



<상영 일차별 관객수>

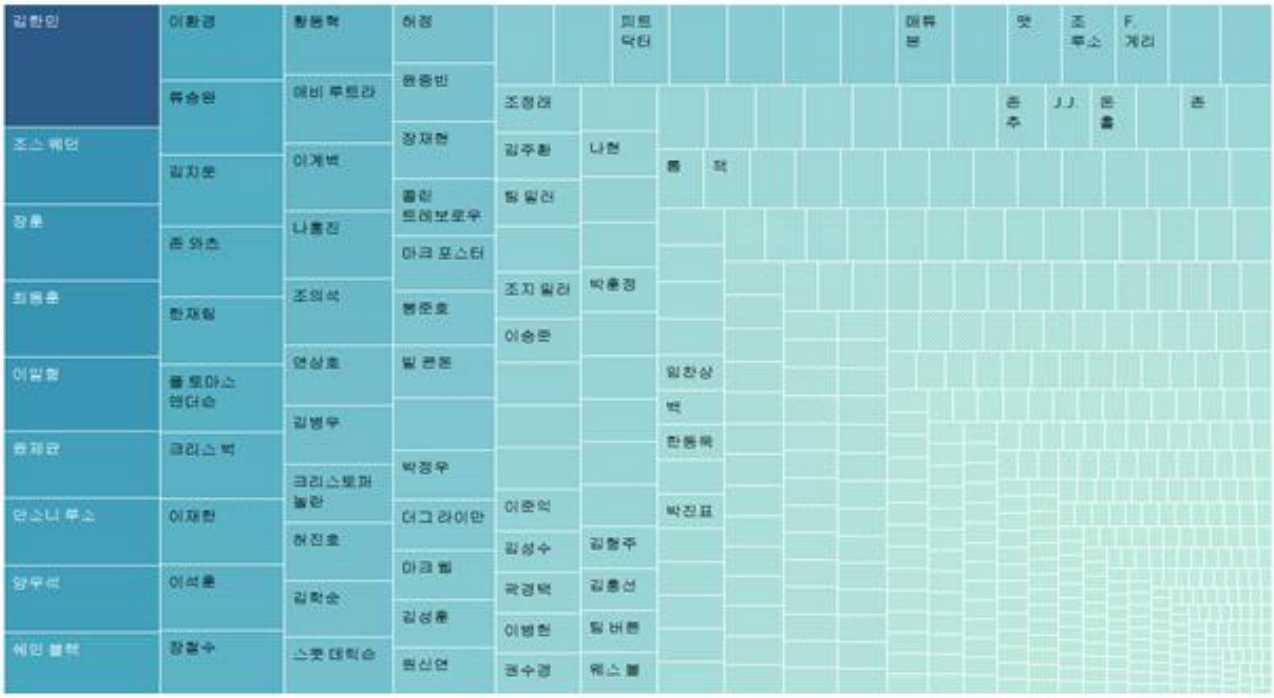
(2) 변수 생성

예측의 정확도를 높이기 위해 종속변수인 누적관객수에 영향을 준다고 판단되는 변수를 생성하였다. 대부분 변수는 관객 수를 이용하여 만들었으며 Numeric 형식을 하고 있다. 일자 구분은 데이터 탐색을 위해 변수를 생성하였다.

각각의 변수 설명은 다음과 같다.

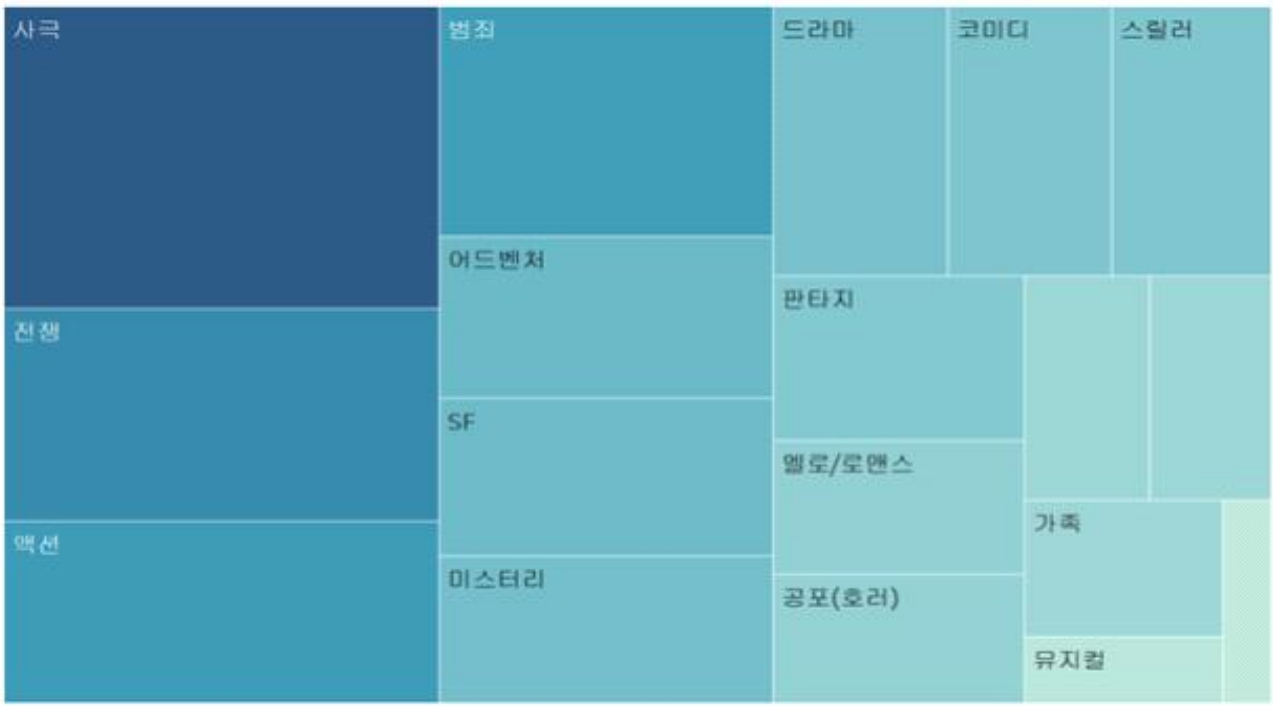
변수명	변수 유형	변수 설명
영화코드	Factor	각 영화 구별 번호 * 동일한 영화제목을 구별하기 위해서 사용
날짜	Date	해당 영화 상영날짜
일자구분	Character	평일, 주말, 공휴일, 추석, 설로 구분
배우파워	Numeric	배우별 누적관객수의 합 * 배우는 각 영화당 대표배우 3명을 추출하여 사용
평점		각 영화의 평균 평점
코멘트수		각 영화의 코멘트수
일차		각 영화별 개봉 후 경과 일수(날짜-개봉일)
일차2		일차 변수의 제곱값
일차3		일차 변수의 세제곱값
감독평균최대관객수		감독별 누적관객수 평균 * 각 영화당 대표감독 1명을 추출하여 사용
장르평균최대관객수		장르별 누적관객수 평균 * 각 영화당 대표장르 1개를 추출하여 사용
배급사평균최대관객수		배급사별 누적관객수 평균 * 각 영화당 배급사 1곳을 추출하여 사용
등급평균최대관객수		등급별 누적관객수 평균
공휴일평균관객수		각 영화의 일자구분별 평균 일일관객수
공휴일누적평균관객수		영화의 상영일차별 공휴일평균관객수 누적 값
공휴일총합		각 영화의 공휴일평균관객수 최대값
평점계		각 영화별 평점 * 코멘트수
기사수		각 영화의 기사수

1) 감독평균최대관객수



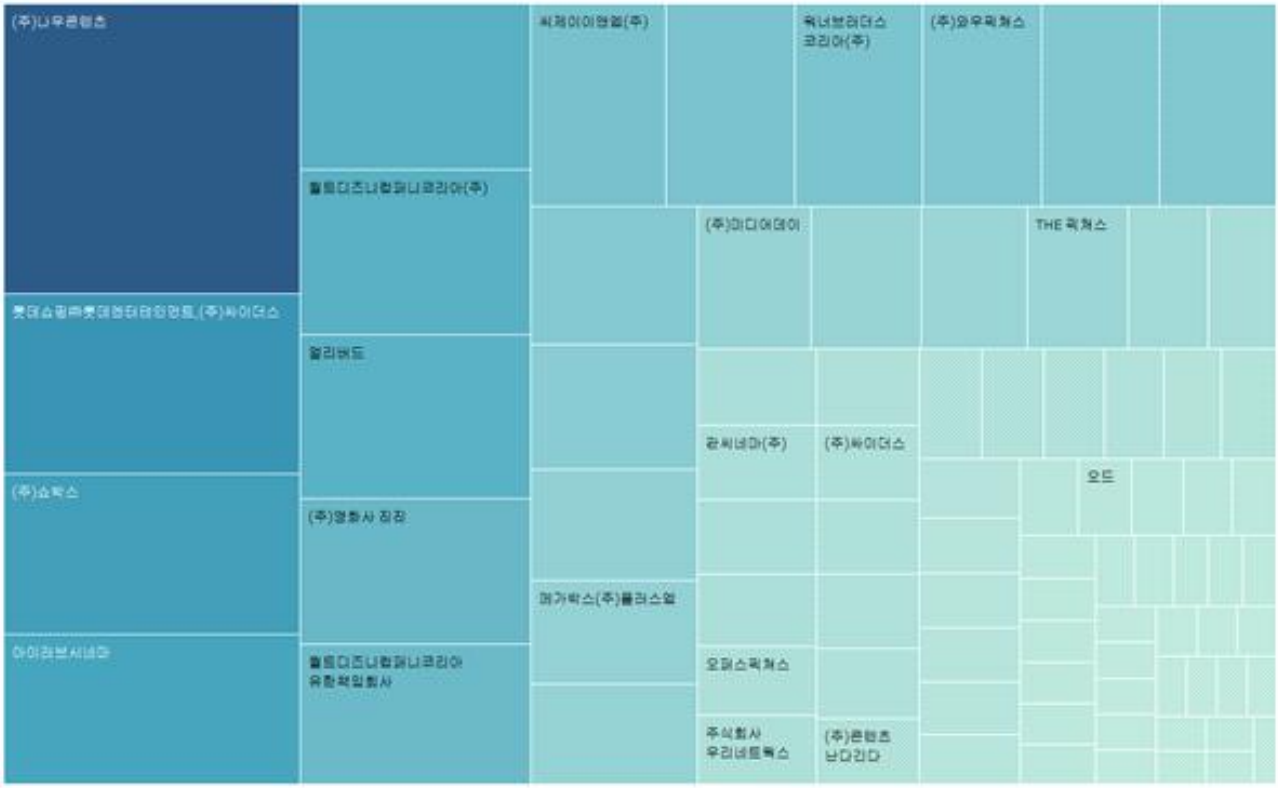
설명 : 그래프에서 네모 칸의 크기는 감독별 누적최대관객수(누적관객수의 최댓값)의 평균값을 나타낸다. 네모 칸이 클수록, 색이 진할수록 감독평균최대관객수가 큰 값을 나타낸다.

2) 장르평균최대관객수



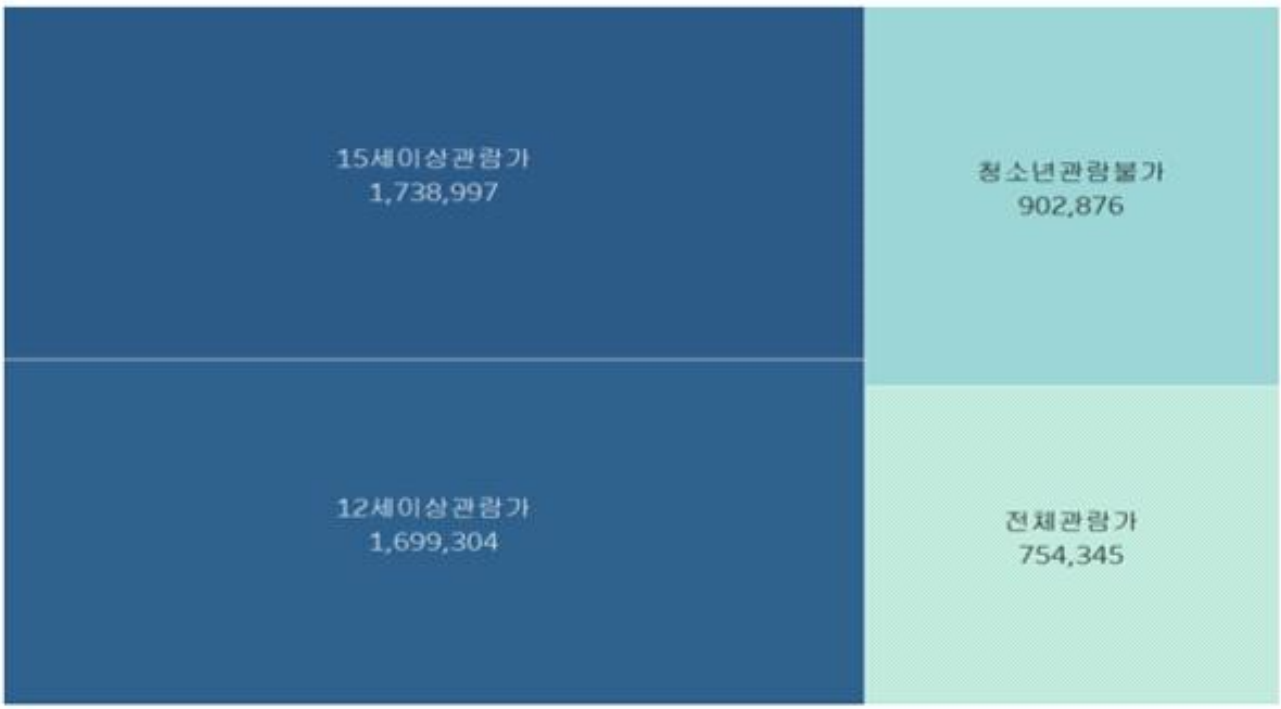
설명 : 그래프에서 네모 칸의 크기는 장르별 누적최대관객수(누적관객수의 최댓값)의 평균값을 나타낸다. 네모 칸이 클수록, 색이 진할수록 장르평균최대관객수가 큰 값을 나타낸다.

3) 배급사평균최대관객수



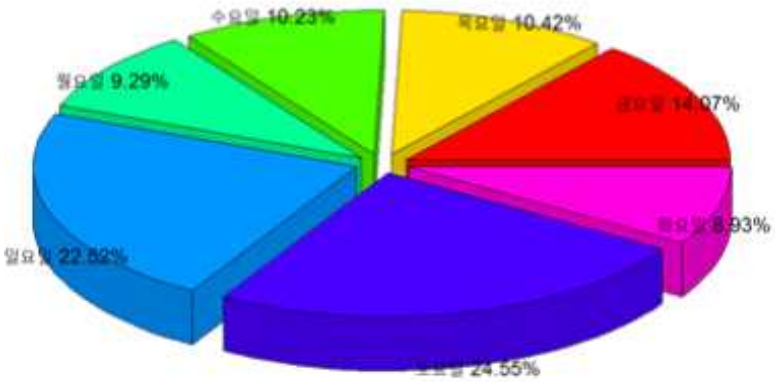
설명 : 그래프에서 네모 칸의 크기는 배급사별 누적최대관객수(누적관객수의 최댓값)의 평균값을 나타낸다. 네모 칸이 클수록, 색이 진할수록 배급사평균최대관객수가 큰 값을 나타낸다.

4) 등급평균최대관객수



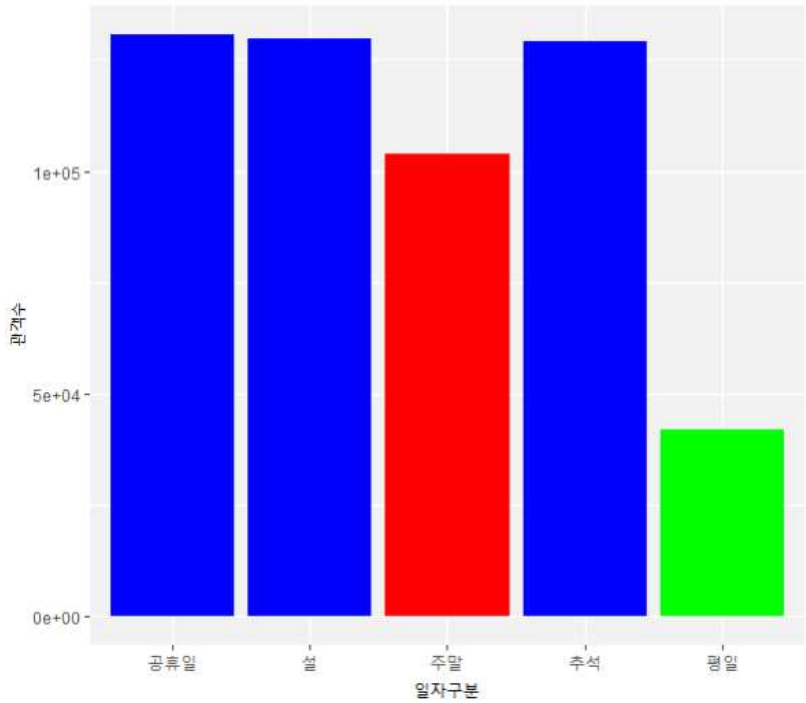
설명 : 그래프에서 네모 칸의 크기는 등급별 누적최대관객수(누적관객수의 최댓값)의 평균값을 나타낸다. 네모 칸이 클수록, 색이 진할수록 등급평균최대관객수가 큰 값을 나타낸다.

5) 요일별 일일 평균 관객수 비율



설명 : 요일별 일일 관객 수 비율을 보면, 월요일, 화요일, 수요일, 목요일이 약 10%를 차지하고 있으며, 금요일이 약 15% 토요일과 일요일이 각각 25%, 23%를 차지하는 것으로 보였다. 따라서 평일과 주말 간 관객 수의 차이는 유의한 것으로 판단하였다.

6) 일자구분, 공휴일평균관객수, 공휴일누적평균관객수, 공휴일총합



설명 : 일자구분별 일일관객수의 평균값을 보면 평일의 평균 일일관객수는 42032, 주말의 평균 일일관객수는 103860, 공휴일, 추석, 설날의 평균일일관객수는 129807이 나온다. 영화의 관객수 예측일은 9월 27일과 10월 10일이다. 두 날짜 사이에는 추석 연휴로 인해 공휴일이 많이 존재하기 때문에 연휴의 특수성을 모형에 반영하기 위해서 일자 구분별로 가중치를 부여하고 공휴일평균관객수, 공휴일누적평균관객수, 공휴일총합이라는 파생변수를 생성했다.

7) 평점, 코멘트수

```
art=[]
for n in range(1,120000) :
    try :
        url ="http://movie.daum.net/moviedb/grade?movieId={}".format(n)
        webpage=urlopen(url)
        source = BeautifulSoup(webpage,'html5lib')
        title =source.find("h2",{'class':'tit_rel'})
        count =source.find('span',{'class','score_rating'})
        article = source.find('span',{'class':'num_review'})
        art.append(title.get_text() + '₩t' + count.get_text()+'₩t'+article.get_text().strip())
    except AttributeError :
        pass
with open('daum_comment.txt', 'w', encoding = 'utf-8') as f :
        for i in info :
            f.write(i + '₩n')
        f.close()
```

설명 : 해당 영화의 평점 평균 및 코멘트수를 나타낸다. 다음 사이트의 영화 평점 및 코멘트를 python을 통해 크롤링하여 사용했다.

8) 기사수

```
art=[]
for n in info : # info는 영화명이 들어있는 리스트
    try :
        k = urllib.parse.quote(n)
        url='https://search.naver.com/search.naver?ie=utf8&where=news&query=%EC%98%81%
            ED%99%94%20{}&sm=tab_pge&sort=0&photo=0&field=0&reporter_article=&pd=0
            &ds=&de=&docid=&nso=so:r,p:all,a:all&mynews=0&cluster_rank=84&start=1&refre
            sh_start=0'.format(k)
        webpage=urlopen(url)
        source = BeautifulSoup(webpage,'html5lib')
        article = source.find('div',{'class':'title_desc all_my'})
        art.append(n + '₩t' + article.get_text().strip())
    except AttributeError :
        pass
with open('naver_article.txt', 'w', encoding = 'utf-8') as f :
        for i in art:
            f.write(i + '₩n')
        f.close()
```

설명 : 해당 영화의 기사수를 나타낸다. 네이버 뉴스에서 python으로 크롤링하여 각 영화 관련 기사수를 생성하였다.

3. 모형

(1) Elastic Net Model

선형회귀 모형에서 설명변수의 수가 상당히 큰 경우, 또는 몇 개의 설명변수들이 상당한 상관관계에 있는 경우 등의 상황에서 최소제곱추정(least squares estimate)은 통계적으로 불안정한 행태를 보인다. 이런 문제를 피하고자 ridge와 lasso를 대안으로 고려하게 된다. ridge(α=0)와 lasso(α=1)는 최소제곱추정에 비해 안정적인 모형을 제공하지만 반대급부로 통계적 편향(bias)을 갖는다. 페널티 파라미터 λ가 커짐에 따라, 제곱 편향이 증가하지만, 모형의 불안정성(unstability)은 감소한다. 반면, λ가 작아짐에 따라 제곱 편향은 감소하고 불안정성이 증가한다. 즉, 편향의 제거와 안정성 제고라는 2개의 목표가 상반된 상황이므로, λ의 선택은 일종의 "거래 문제(trade-off problem)"가 된다.

일련의 변수들이 양의 상관관계가 되어 변수들의 합이 하나의 잠재변인으로 반응 y에 영향을 준다고 하자. 그런 상황에서 최소제곱추정 또는 최대가능도추정은 불안정성을 보인다. 변수들에 붙는 계수에 큰 변동이 나타난다는 말이다. 이는 다중공선성의 상황이기 때문이다. 이런 상황에서 ridge는 모든변수들의 합을 적합모형에 반영시키고 lasso는 양 상관 변수 중 1개를 선별하여 적합모형에 포함한다. 결과적으로 ridge와 lasso는 그런대로 좋은 해를 제시하게 되는데, 모형의 안정성 측면에서는 ridge가 낫고 간결성 측면에서는 lasso가 낫다. elastic net은 ridge와 lasso를 보완하여 긴밀한 설명변수의 그룹을 유지하면서 변수를 선택할 수 있는 방법론을 일컫는다. elastic net의 목적식은 다음과 같다.

$$\hat{\beta} = \mathrm{argmin}_{\beta} |\mathbf{Y} - \mathbf{X}\beta|^2 + \lambda_2 |\beta|^1 + \lambda_1 |\beta|^2$$

이 목적식은 L1-norm과 L2-norm의 합인 제약 조건을 가지며 선형 계수 값의 크기를 줄이고 그룹이 유지되도록 변수를 선택하게 만든다. Elastic net은 ridge와 lasso의 결합형태의 모형으로 가중치의 절대값의 합과 제곱합을 동시에 제약 조건으로 가진다. 값을 α를 0부터 1까지로 조정함으로써 필요에 따라 rigde와 lasso을 사용할 수도 있다.

(2) All Subset Feature Selection

일반적으로 변수의 개수가 20개 이하일 경우 계산량이 비교적 부담스럽지 않아 모든 변수의 조합 수 2^n개를 확인하여 최적의 변수조합을 확인한다. 본 팀은 변수선택을 해주는 elastic net 모형을 사용하였지만, 변수의 개수가 많아지면서 생기는 변수들 사이에 상관관계에 따라 모형의 성능이 달라지는 것을 확인하였다. 그 때문에 14개 변수를 전부 사용해 모형을 생성하기보다는 종속변수인 누적관객수를 보다 잘 예측할 수 있는 변수들의 조합을 찾기 위해 all-subset feature selection 방법을 하였다.

4. 예측

(1) 모형수립

1) Glmnet 패키지

R의 "glmnet" 패키지를 활용하여 elastic net 모형을 수립하였다. 모형의 parameter인 α값을 조정함으로써 ridge, lasso 또한 고려할 수 있었다.

2) 변수 및 α의 선택

모형의 성능확인을 위해 target 영화("킹스맨 : 골든 서클", "남한산성", "넛잡2")와 비슷하다고 판단한 47개의 영화를 test set으로 사용하였다. test set를 제외한 나머지 576개의 영화를 train set으로 설정하여 14개 변수의 조합과 α값을 훈련시켜 성능을 비교하였다. 비교 지표로는 test set의 RMSE를 사용하였다. 변수와 α의 조합에 대해 모든 영화의 RMSE 평균을 구하고, 그 평균이 최소가 되는 모형을 선정하였다.

3) Cross validation을 사용한 best λ선택

Glmnet의 파라미터인 λ를 추정하기 위해 Cross validation(CV)을 진행하였다. Cross validation은 근사모형의 정확도 평가 방법 중 하나이다. 모형 생성에 필요한 전체 데이터 중 test set를 제거하고 남은 train set를 n분할하여 모형을 추정한다. 추정된 모형을 통해 test 데이터에 대한 예측값을 구한 후 이 값의 차이를 비교하는 방법으로 성능을 검사한다. Glmnet 패키지의 Cross validation은 이 과정을 통해 RMSE를 최소로 하는 best λ를 선택한다. Cross validation에서 RMSE 값이 가장 작게 생성되는 λ값을 파악할 수 있었다. 본 팀의 경우 10-fold cross validation을 사용했다.

(2) 최종모형

1) 킹스맨 : 골든서클

① 사용변수

```
log(누적관객수) ~ 일차 + sqrt(배우파워) + log(감독평균최대관객수) + log(장르평균최대관객수) + log(배급사평균최대관객수) + log(등급평균최대관객수) + log(공휴일평균관객수) + sqrt(공휴일누적평균관객수) + log(공휴일총합) + 평점 + 코멘트수 + 기사수
```

② 최종모형

```
set.seed(1)
cv.elastic = cv.glmnet(data.matrix(x_train), y_train, type.measure="mse", alpha =0.1,
                       family="gaussian")
bestlam = cv.elastic$lambda.min  # best λ=0.002382049
elastic.fit = glmnet(data.matrix(x_train), y_train, alpha =0.1, lambda=bestlam,
                     family="gaussian")
result_elastic = predict(elastic.fit, data.matrix(x_test))
result_elastic = round(exp(result_elastic),2)
result_elastic
```

2) 남한산성

① 사용변수

```
log(누적관객수) ~ 일차 + 일차2 + 일차3 + log(감독평균최대관객수) + log(배급사평균최대관객수) + log(등급평균최대관객수) + log(공휴일평균관객수) + sqrt(공휴일누적평균관객수) + log(공휴일총합) + 평점 + 코멘트수 + 기사수
```

② 최종모형

```
set.seed(1)
cv.elastic = cv.glmnet(data.matrix(x_train), y_train, type.measure="mse", alpha=0.9,
                       family="gaussian")
bestlam = cv.elastic$lambda.min  # best λ=0.001059422
elastic.fit = glmnet(data.matrix(x_train), y_train, alpha =0.9, lambda=bestlam,
                     family="gaussian")
result_elastic = predict(elastic.fit, data.matrix(x_test))
result_elastic = round(exp(result_elastic),2)
result_elastic
```

3) 넛잡2

① 사용변수

```
log(누적관객수) ~ 일차 + 일차2 + 일차3 + log(감독평균최대관객수) + log(배급사평균최대관객수) + log(등급평균최대관객수) + log(공휴일평균관객수) + sqrt(공휴일누적평균관객수) + log(공휴일총합) + 평점 + 코멘트수 + 기사수
```

② 최종모형

```
set.seed(1)
cv.elastic = cv.glmnet(data.matrix(x_train), y_train, type.measure="mse", alpha=0.9,
                       family="gaussian")
bestlam = cv.elastic$lambda.min  # best λ=0.001059422
elastic.fit = glmnet(data.matrix(x_train), y_train, alpha =0.9, lambda=bestlam,
                     family="gaussian")
result_elastic = predict(elastic.fit, data.matrix(x_test))
result_elastic = round(exp(result_elastic),2)
result_elastic
```

(3) 최종결과

영화명	개봉일	개봉 후 경과일	**10월 10일 관객수**
킹스맨 : 골든서클	9월 27일	13일	**4,772,115**
남한산성	10월 3일	7일	**3,702,712**
넛잡2	10월 3일	7일	**350,956**

5. 결과 분석 및 후기

대회의 목표인 누적관객수 예측을 위해 기존 변수 및 파생 변수들의 데이터 탐색을 통해 데이터의 특징을 파악하고자 노력했다. 이 과정에서 스크린 수와 누적 관객 수를 변수로 생성하는 것이 영화 흥행에 있어 중요한 역할을 한다는 것을 확인했지만, 본 팀이 예측하고자 하는 영화의 경우 개봉 초기이거나 미개봉 영화이기 때문에 스크린 수와 누적 관객수를 반영하는데 한계가 있었다. 이에 스크린 수의 영향력을 반영하기 위해 배급사라는 변수를 이용했다. 그리고 실제로 배급사를 이용해 만든 파생변수를 통해 성능이 향상되었다. 또한 영향력을 크다고 판단한 연휴의 경우, 보유한 데이터에서는 이번 10일간의 연휴만큼 긴 데이터가 없었다. 이에 보유한 데이터에서 최대한 긴 연휴 데이터를 학습하려고 했지만 대부분의 연휴가 3~5일로 10일의 연휴를 예측하는데 어려움이 있었다. 또한 연휴를 평일이나 일반적인 공휴일과 다르게 가중치를 주는 과정에서 단순히 연휴 일자에만 가중치를 주지 않고, 연휴의 연속되는 일자에 따라 가중치를 다르게 주도록 했다.

주어진 변수와 데이터를 어떤 모형에 적용할지에 대해도 고민이 많았다. 종속변수 '누적관객수'가 연속형 데이터이기 때문에 회귀모형을 사용하기로 했고, 여러 회귀모형 중 제약조건을 통해 정규화를 하며 변수선택도 가능하고 안정성이 뛰어난 여러 장점을 가진 elastic net 모형의 성능이 가장 좋았다. 모형을 비교적 빨리 결정해서 추가적인 파생변수에 시간을 더 많이 투자할 수 있었다.