기계학습 Assignment Report

2022311917

최영조

Index

1. Overall analysis

A. Viewing

전반적인 데이터의 정보는 다음과 같다.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1411 entries, 0 to 1410
Data columns (total 13 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   movie_ID         1411 non-null   int64
 1   title            1411 non-null   object
 2   genre            1408 non-null   object
 3   running_time     1409 non-null   float64
 4   netizen_ratings  1411 non-null   float64
 5   num_reviews      1411 non-null   int64
 6   num_article      1411 non-null   int64
 7   country          1409 non-null   object
 8   rating           1410 non-null   object
 9   companyNm        1406 non-null   object
 10  released_year    1411 non-null   int64
 11  released_month   1411 non-null   int64
 12  num_viewers      1411 non-null   int64
dtypes: float64(2), int64(6), object(5)
memory usage: 143.4+ KB
```

이중 결측 데이터는 다음과 같다.

title	null column
작은 연못	[companyNm]
스테이트 오브 플레이	[companyNm]
코렐라인: 비밀의 문	[companyNm]
퍼블릭 에너미	[companyNm]
스타워즈: 라스트 제다이	[genre]
넛잡 2	[genre]
저수지 게임	[genre]
킬러의 보디가드	[running_time]
지오스톰	[running_time, country]
패터슨	[country]
로마의 휴일	[rating, companyNm]

해당 데이터셋에서 별도의 전처리 없이 결측 값이 포함된 행은 모두 제거하고 명목형 변수 역시 제거 후 수치형 변수인 'movie_id', 'running_time', 'netizen_ratings' ,'num_article', 'released_year', 'released_month'만을 이용해 'num_viewers'를 예측한 Linear Regression은 0.6692의 성능을 보였다. 성능은 r-squared를 사용했으며 학습 데이터셋과 평가 데이터셋의 불균형한 분리를 우려해 10겹 교차검증을 사용 후 평균값을 산출하였다. 이 0.6692에서 더 높은 성능을 얻는 것을 목표로 한다.

B. Strategy

결측 값의 분포를 보면 인터넷 검색을 통해 쉽게 얻을 수 있는 정보들이므로 크롤링을 이용하도록 한다. 추가적인 독립변수인 '감독'을 크롤링으로 수집하도록 한다. 그외 여러 번의 크롤링 코드로도 수집되지 않는 몇몇 케이스는 직접 수집하도록 한다.

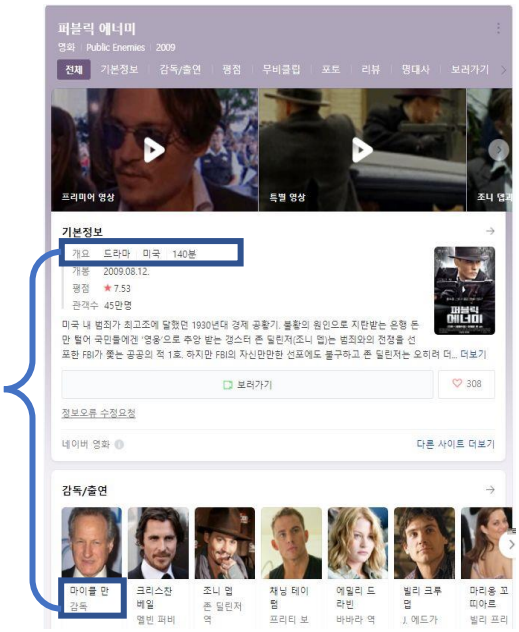
크롤링 이후 완성된 데이터셋에 대해 시각화, 통계적 검증 과정을 거쳐 종속변수와의 관계를 파악해 제거하거나 알맞게 변환해 사용하도록 한다. 얻은 변환 기준은 모델 파이프라인으로 구축해 학습과 검증을 완전히 분리하도록 한다.

'

2. Data Collection

A. Data Crawling

네이버 검색창에 '영화 + title'을 검색하면 나오는 정보는 다음과 같다.



해당 창에는 감독에 대한 정보 뿐 아니라 결측 값 중 'genre', 'country', 'running_time'에 대한 정보를 얻을 수 있다. 해당 창을 이용해도 'companyNm'과 'rating'정보를 얻을 수는 없으나 개수가 적기 때문에 따로 수집을 하도록 한다.

B. Transforming Crawled Data

감독데이터는 그대로 모델에 사용할 수 없으니 적절한 형태로 변환하도록 한다.

https://movie.naver.com/movie/sdb/rank/rpeople.naver?date=20220329&tg=2

위 사이트는 영화인의 검색랭킹을 집계한 사이트이다. 2005년 2월 7일 이후의 모든 날짜에 대해 日 단위로 집계 되어있으므로 모든 데이터에 있는 2008년부터 2018년까지의 모든 정보를 수집하도록 한다. (2005년은 2월 7일 이전 랭킹이 없는데다가 우리가 가진 데이터가 단 한 개 뿐이니 무시하고 진행)

미래에 인기 있는 감독이 과거 흥행에 영향을 끼치는 것은 이상하기 때문에 영화 개봉 시점(월 단위) 이전의 집계만을 이용하도록 한다. 해당 달과 그 이전 시점의 모든 데이터는 평균을 취해 순위를 계산한다.

추가로 집계되지 않은 감독 순위는 어떤 계수가 붙어도 종속변수에 영향을 미치지 않도록 0을 부여한다. 이때 집계된 순위는 숫자가 작을수록 좋은 값이었으므로 min과 max 값을 뒤바꿔 주기로 한다.

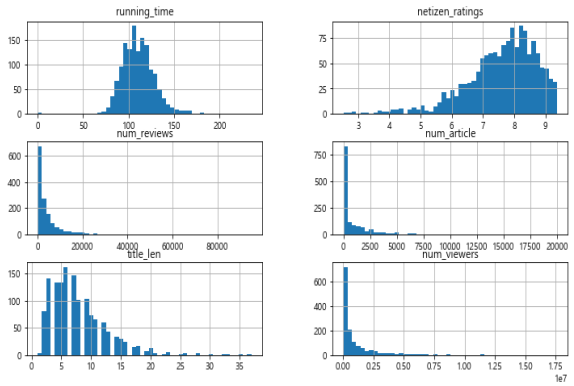
3. EDA

A. Numeric Variables

- movie_id : 영화의 고유 식별 번호 20168324와 같은 형식으로 앞의 4자리는 'released_year', 뒤의 네 자리는 고유 식별 번호이지만 별다른 특징 정보를 추출할 수 없으니 해당 열은 제거하기로 한다.

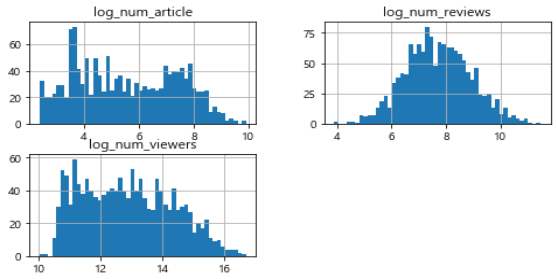
- title_len : title은 원래 명목형 변수이지만 짧고 강렬한 문구에 관객을 모을 수 있겠다는 생각이 들어 title의 길이를 수치형 변수로 추가하도록 한다.

-running_time, netizen_ratings, num_reviews, num_article, title_len(독립변수) + num_viewers(종속변수) : 수치형 변수의 전체적인 분포(히스토그램)는 다음과 같다.

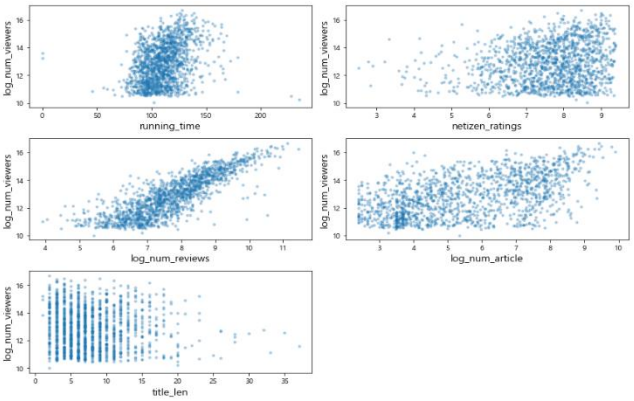


'num_article'이 0인 영화 목록을 보면 흥행했던 것들이 포함 되어있기 때문에 이를 이상치로 판단하여 다른 값으로 대치하기로 한다. 'num_article'이 0인 값들을 제거하면 'num_reviews'와 상관계수가 0.6 가량 나오기 때문에 'num_reviews'에 근거해 대치하도록 한다. 'num_reviews'를 20개 구간으로 나눠 이상치 데이터가 속하는 구간의 중앙값을 이상치에 채워넣었다.

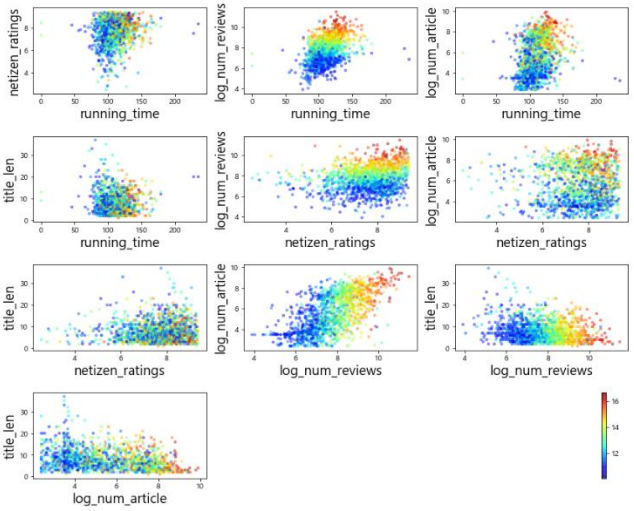
'num_reviews', 'num_article', 'num_viewers'가 심하게 skew되어있어 선형모델에 적합해보이지 않으므로 log를 취해주었다. 음수의 무한대로 빠지는 것을 방지하기 위해 1을 더한 뒤 log를 취한 분포는 다음과 같다. 원본 데이터보다 넓게 고루 분포하므로 log를 취한 값을 사용하도록 한다.



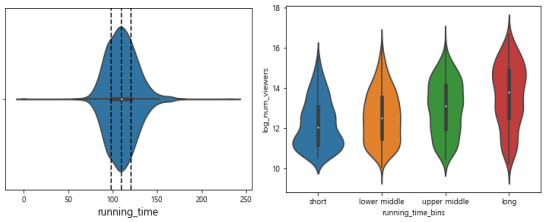
각 수치형 독립변수와 종속변수 간의 관계를 시각화하면 다음 그림들과 같다. x축은 각 독립변수 들이며 y축은 종속변수('log_num_viewers')이다. 'log_num_article'은 아주 좋은 분포를 보이지는 않으므로 'num_article'을 버리지 않고 계속 같이 비교해주기로 한다.



종속변수와 log_num_viewers는 상당한 선형성이 존재하는 것으로 보이며 상관계수 역시 0.83으로 매우 높은 수치를 보인다. 독립변수들의 상호작용을 파악하기 위해 scatter plot을 그리면 다음과 같다.



색이 빨간색에 가까울수록 종속변수의 값이 큰 것이므로 영역이 구분되는 모양을 보며 종속변수에 대한 영향력을 파악할 수 있다. 'log_num_reviews' 이외의 별다른 영향력은 포착되지 않으나 'running_time' 변수에서 색에 따라 구분되는 영역을 포착할 수 있으므로 구간을 나눠보았다.



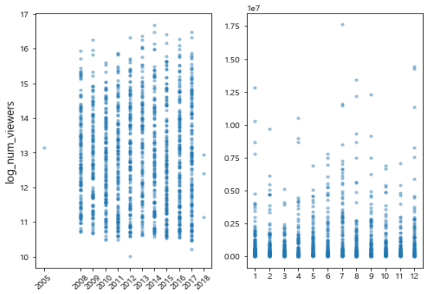
각 구간은 원소의 개수가 비슷하도록 나눴으며 98분, 110분, 121분을 경계 값으로 갖는다. 해당 구간에서 'log_num_viewers'의 분포는 오른쪽 violin plot과 같으며 각 구간 데이터들이 정규분포를 띄우지 않으므로 다음 비모수 검정을 이용하여 평균간 차이를 체크하였다.

```
KruskalResult(statistic=166.5
034661332048,
pvalue=7.235501554425752e-36)
```

검정 결과 구간별 차이가 있다는 가설을 취하고 4개로 나눠진 구간데이터 'running_time_bins' 변수를 추가하고 'running_time' 변수를 제거하였다.

B. Date Variables

날짜형 변수로는 'released_year'와 'released_month'가 존재한다.



해당 데이터는 수치형으로 제공되어 있으나 위의 종속변수와의 관계 그래프에서 보듯 어떤 순서를 찾기에는 부적절해 보이므로 명목형 변수로 취급하고 그룹간 종속변수의 차이가 있는지 체크하도록 한다. 두 날짜 범주에 대한 단순비교는 다음과 같다.

released_year	count	mean	min	max	var
2005	1	13.1355	13.1355	13.1355	NaN
2008	134	12.89041	10.72086	15.92518	1.750623
2009	126	12.88219	10.67549	16.25379	1.675356
2010	143	12.74562	10.48456	15.59502	1.627326
2011	141	12.75286	10.5029	15.86769	2.135518
2012	143	12.96664	10.02575	16.3267	2.208822
2013	124	13.16825	10.61683	16.36583	2.184397
2014	140	12.87634	10.57367	16.68419	2.304117
2015	151	12.70291	10.54149	16.41181	2.267148
2016	155	12.78445	10.55247	16.26354	2.364533
2017	150	12.93213	10.21979	16.4835	2.506884
2018	3	12.16352	11.15389	12.93927	0.837919

released_month	count	mean	min	max	var
1	129	12.99753	10.48799	16.36583	1.794428
2	127	12.78424	10.21979	16.08837	1.782302
3	117	12.56448	10.55336	15.45221	1.596601
4	126	12.62466	10.60068	16.16636	2.169741
5	96	13.125	10.63299	15.74412	1.906154
6	103	12.99392	10.61261	15.86769	2.327946
7	115	13.31251	10.62736	16.68419	2.866813
8	139	12.67682	10.02575	16.41181	2.249041
9	125	12.8381	10.47155	16.3267	2.278267
10	123	12.71502	10.59906	15.74411	1.778826
11	100	12.73789	10.54149	15.77166	1.840933
12	111	13.11064	10.47085	16.4835	2.381068

각 범주는 모두 정규분포를 띄지 않으므로 'running_time'과 마찬가지로 비모수 검정을 이용해 그룹간 평균 차이를 체크하였다.

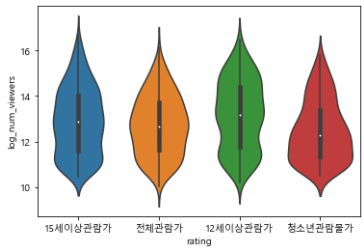
```
KruskalResult(statistic=11.0550881
16252954,
pvalue=0.2719518261862074)

KruskalResult(statistic=29.5490231
6949086,
pvalue=0.0018644350519613502)
```

위는 년도에 대해 (2005년과 2008년 제외), 아래는 월에 대해 검정을 실시한 것으로써 년도는 그룹간 차이를 발견하지 못했으나 월은 그룹간 차이가 존재한다는 가설을 받아들이기로 한다. 다만 종속변수에 영향을 미치는 순서로 존재하는 것은 아니므로 one-hot encoding을 사용하도록 한다.

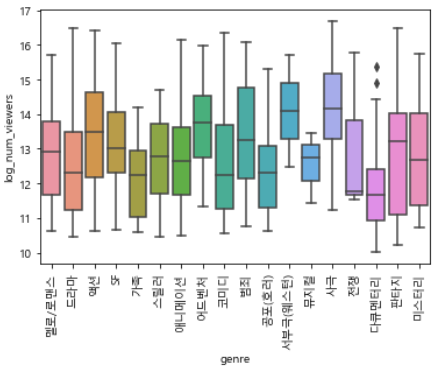
C. Categorical Variables

- 'rating'은 12세이상, 15세이상, 전체 관람, 청소년 관람 불가 네 범주로 나뉜다. 다음 violin plot은 각 범주 별 'log_num_viewers'의 분포를 나타낸 것이다.

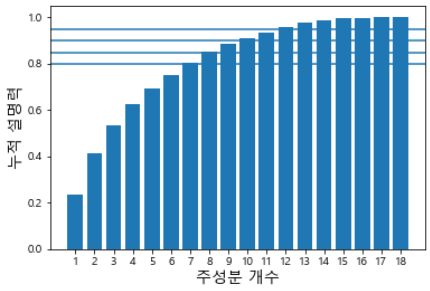


종속변수 분포에서 전체관람가->12세이상 관람가->15세이상 관람가->청소년관람불가로의 어떤 순서가 보이지는 않으므로 label encoding을 쓰기에는 무리가 있다. 동일한 비모수 검정을 통해 그룹간 차이가 존재한다는 가설을 받아들이고 one-hot encoding을 사용하였다. (사후검정을 실시하려 하였으나 해당 패키지가 존재하지 않아 미실시)

- 'genre'에 따른 종속변수 분포는 다음과 같다.



마찬가지로 one-hot encoding을 사용하지만 범주가 너무 많으므로 PCA를 통해 차원을 일정수준 축소하기로 한다.



주성분 개수에 따른 누적 설명력은 위 그림과 같다. 85%의 설명력을 확보할 수 있도록 주성분의 개수를 택하기로 한다.

- 'companyNm' 역시 수가 너무 많으므로 'genre'와 동일한 방법을 사용하기로 한다. 다만 배급사 수가 애초에 너무 많으므로 전체의 점유의 20% 미만을 차지하는 여러 배급사는 기타 범주로 통합 후 one-hot encoding과 PCA를 진행하도록 한다. 모든 배급사들은 총 132개이고, 전체 1411개의 영화의 20%인 282개 가량의 데이터를 기타로 통합한다. 해당 282개의 데이터는 배급 영화의 수가 11개 이하인 배급사의 영화 목록을 모두 합하면 얻을 수 있다. 이를 통해 남은 28개의 배급사에 대해 'genre' 변수와 동일한 전략을 취하기로 하였다.

- 'country'의 분포는 다음과 같다.

미국	675	이탈리아	4
한국	482	대만	2
일본	72	덴마크	2
영국	45	남아프리카공화국	2
프랑스	40	스웨덴	2
중국	23	뉴질랜드	2
독일	11	멕시코	1
스페인	11	러시아	1
홍콩	11	아이슬란드	1
벨기에	7	페루	1
캐나다	5	오스트리아	1
인도	5	아일랜드	1
호주	4		

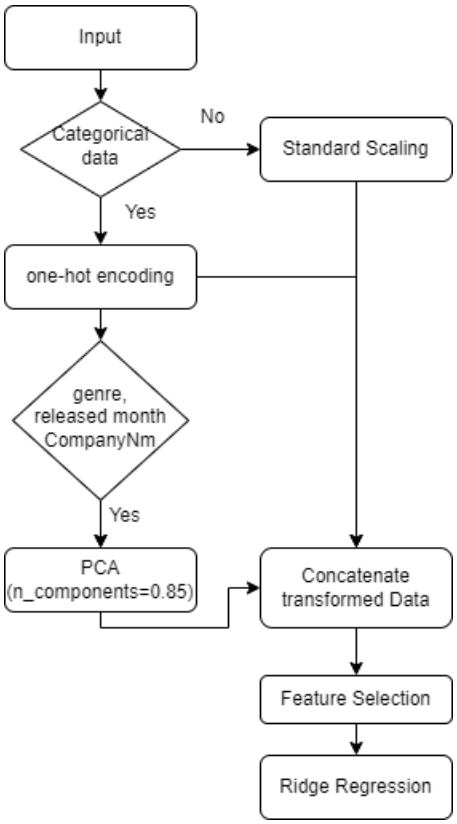
대부분의 영화가 미국과 한국이므로 '한국', '미국', '기타' 범주로 나눠 one-hot encoding을 사용하였다.

4. Modeling

A. Pipeline

One-hot encoding, Standard-Scaling, PCA등은 scikit-learn에서 제공하는 객체를 사용한다. 다만 train과 validation set를 동시에 사용해 각 변환 객체에 사용하면 데이터 유출의 가능성이 있기 때문에 scikit-learn에서 제공하는 pipeline 함수를 이용하도록 한다.

Pipeline의 순서는 다음과 같다.

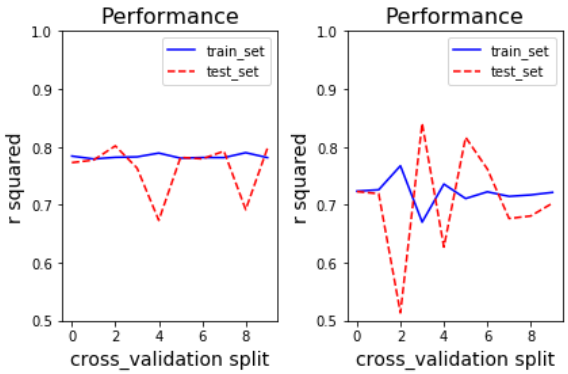


수치형 변수의 경우 Standardization을 사용한다. Categorical 변수는 one-hot을 사용하되 변수 종류에 따라 PCA를 사용하는 것도, 하지 않는 것도 있다. 모아진 데이터에 의미 없는 변수가 있다 판단하여, scikit learn의 Feature Selection모델 중 통계적 유의성을 검정해 변수를 선택하는 함수를 이용해 유의미한 80%의 변수들만 남기도록 했다. 해당 pipeline 통째로 cross validation을 수행하였다. Ridge Regression은 하이퍼파라미터 alpha에 대해 1을 사용하였다.

B. Result

해당 pipeline을 통해 얻어진 데이터는 분할되는 랜덤성에 의해 조금씩 다르지만 대략 37~39개 정도의 변수들로 이뤄진 것을 확인했다. Kfold는 10번 수행했으며 분할 때마다 랜덤성을 추가하였다. Cross-validation 결과 r-squared는 0.7645을 얻었다. 종속변수에 취한 로그변환을 풀고 exponential 함수를 취한 것에 대한 r-squared는 0.7125를 얻었다. (학습은 log 변환된 데이터에 fitting되었고 score 함수를 정의해 넣음)

각 cv별 결과는 다음과 같다. 좌측은 종속변수의 log를 풀지 않은 결과이고 우측은 종속변수의 log를 풀고 얻은 결과물이다.



5. Conclusion

전체적인 모델의 결과는 log변환시 0.7645, log 변환을 풀었을 때 0.7125의 결과를 얻었다. EDA를 통해 의미 없는 변수를 1차적으로 제거하는 과정을 거쳤고, 애매한 것들에 대해서는 scikit-learn함수를 통해 자동적으로 누락시킬 수 있도록 하였다. 명목형 변수에 대해서는 one-hot encoding에 이어 PCA를 수행했고, 수치형 변수에 대해서는 Standardization을 수행했다. 일괄적 적용이 아니기 때문에 scikit-learn의 ColumnTransformer를 이용했으며, 늘어난 변수들에 대해 통계적으로 유의한 변수 80%만 선택해서 사용하도록 하는 과정을 전체 pipeline에 넣었다. 추가로 수집한 변수는 감독 데이터로 검색순위에 맞게 수치형 변수로 변환하였으나 모델의 성능 향상에 크게 기여하지는 못한 듯싶다.

다음과 같이 아쉬운 점들이 있다.

- 감독 순위에 대한 기준을 크롤링할 때 현재와 가까운 데이터에 더 가중치를 주지 않고 아주 먼 과거와 일괄적으로 평균을 구했다.
- log변환으로 변수들의 좀 더 고른 분포를 얻었으나 다시 exponential을 취해 r-squared를 계산했을 때 높은 결과를 얻지 못했다. 모든 EDA가 log를 취한 종속변수에 맞추어 진행되어 그런 것이 아닐까 하는 생각이 드는데 종속변수가 불균형한 경우의 처리법에 대해 더 살필 필요가 있어보인다.
- 매 Cross-validation시 학습데이터로 분리된 데이터의 기준에 따라 ridge모델이 갖는 회귀계수의 개수가 일치하지 않아 변수와 회귀 계수를 매칭해 모델의 결과를 해석하는 것에 무리가 있었다.