영화 관객 수 예측하기

경영학과 2019111139 안은정

문제정의

- 감독, 이름, 상영등급, 스태프 수 등의 정보로 영화 관객 수를 예측하는 모델을 만들고자 함
- Y: box_off_num

executed in 51ms, finished 12:07:52 2023-04-07

[3]:

```
1 df.info()
```

executed in 42ms, finished 12:07:58 2023-04-07

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 600 entries, 0 to 599
Data columns (total 12 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   title           600 non-null    object 
 1   distributor     600 non-null    object 
 2   genre           600 non-null    object 
 3   release_time    600 non-null    object 
 4   time            600 non-null    int64  
 5   screening_rat   600 non-null    object 
 6   director        600 non-null    object 
 7   dir_prev_bfnum  270 non-null    float64
 8   dir_prev_num    600 non-null    int64  
 9   num_staff       600 non-null    int64  
 10  num_actor       600 non-null    int64  
 11  box_off_num     600 non-null    int64  
dtypes: float64(1), int64(5), object(6)
memory usage: 56.4+ KB
```

데이터 전처리

```
In [6]:
1  ## 결측치를 찾음
2  df.isnull().sum()
3  ## dir_prev_bfnum 에 결측치가 있음. 감독이 이전
```

executed in 20ms, finished 12:09:08 2023-04-07

Out[6]:

title	0
distributor	0
genre	0
release_time	0
time	0
screening_rat	0
director	0
dir_prev_bfnum	330
dir_prev_num	0
num_staff	0
num_actor	0
box_off_num	0
dtype: int64	

```
In [9]:
1  df['dir_prev_bfnum'] = df['dir_prev_bfnum'].fillna(0)
```

executed in 20ms, finished 12:10:15 2023-04-07

```
In [11]:
1  df.isnull().sum()
```

executed in 22ms, finished 12:10:27 2023-04-07

Out[11]:

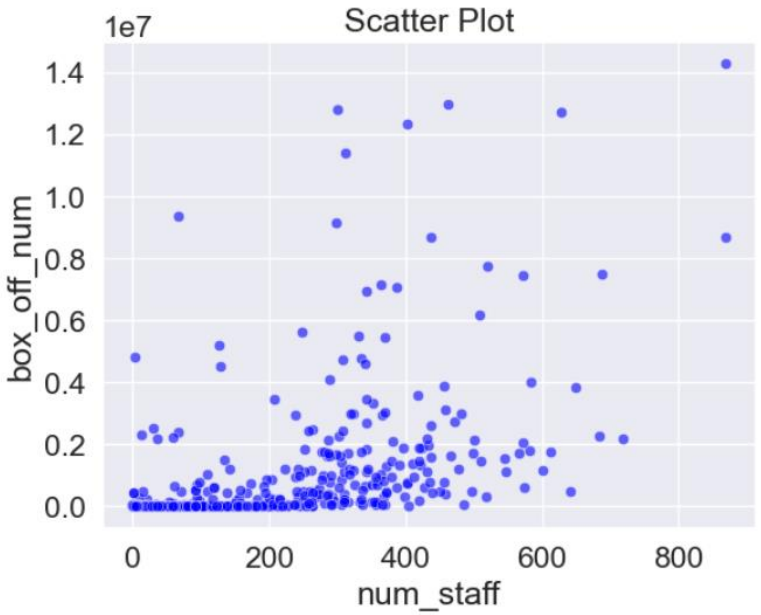
title	0
distributor	0
genre	0
release_time	0
time	0
screening_rat	0
director	0
dir_prev_bfnum	0
dir_prev_num	0
num_staff	0
num_actor	0
box_off_num	0
dtype: int64	

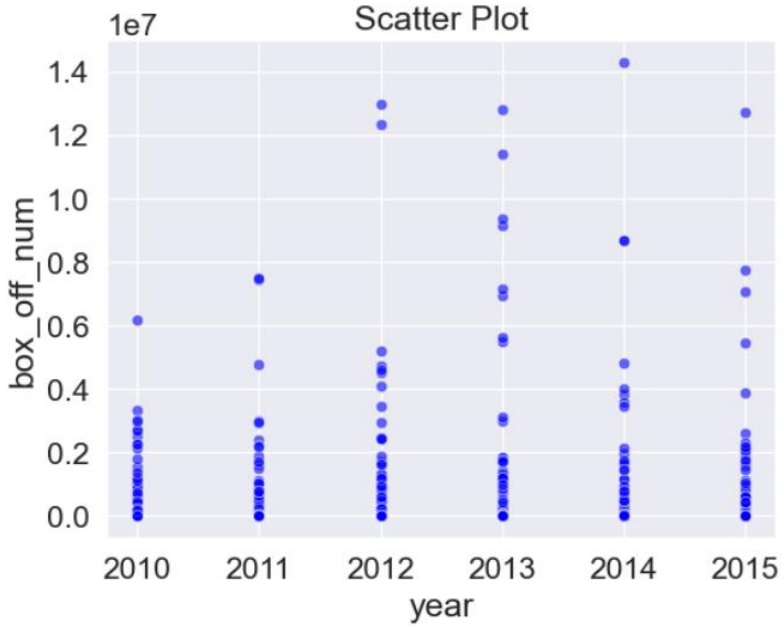
	title	distributor	genre	release_time	time	screening_rat	director	dir_prev_bfnum	dir_prev_num	num_staff	num_actor	box_off_num
0	개들의 전쟁	롯데엔터테인먼트	액션	2012-11-22	96	청소년 관람불가	조병옥	NaN	0	91	2	23398
1	내부자들	(주)쇼박스	느와르	2015-11-19	130	청소년 관람불가	우민호	1161602.50	2	387	3	7072501
2	은밀하게 위대하게	(주)쇼박스	액션	2013-06-05	123	15세 관람가	장철수	220775.25	4	343	4	6959083

year	month
2012	11
2015	11
2013	6

종속, 독립변수 탐색

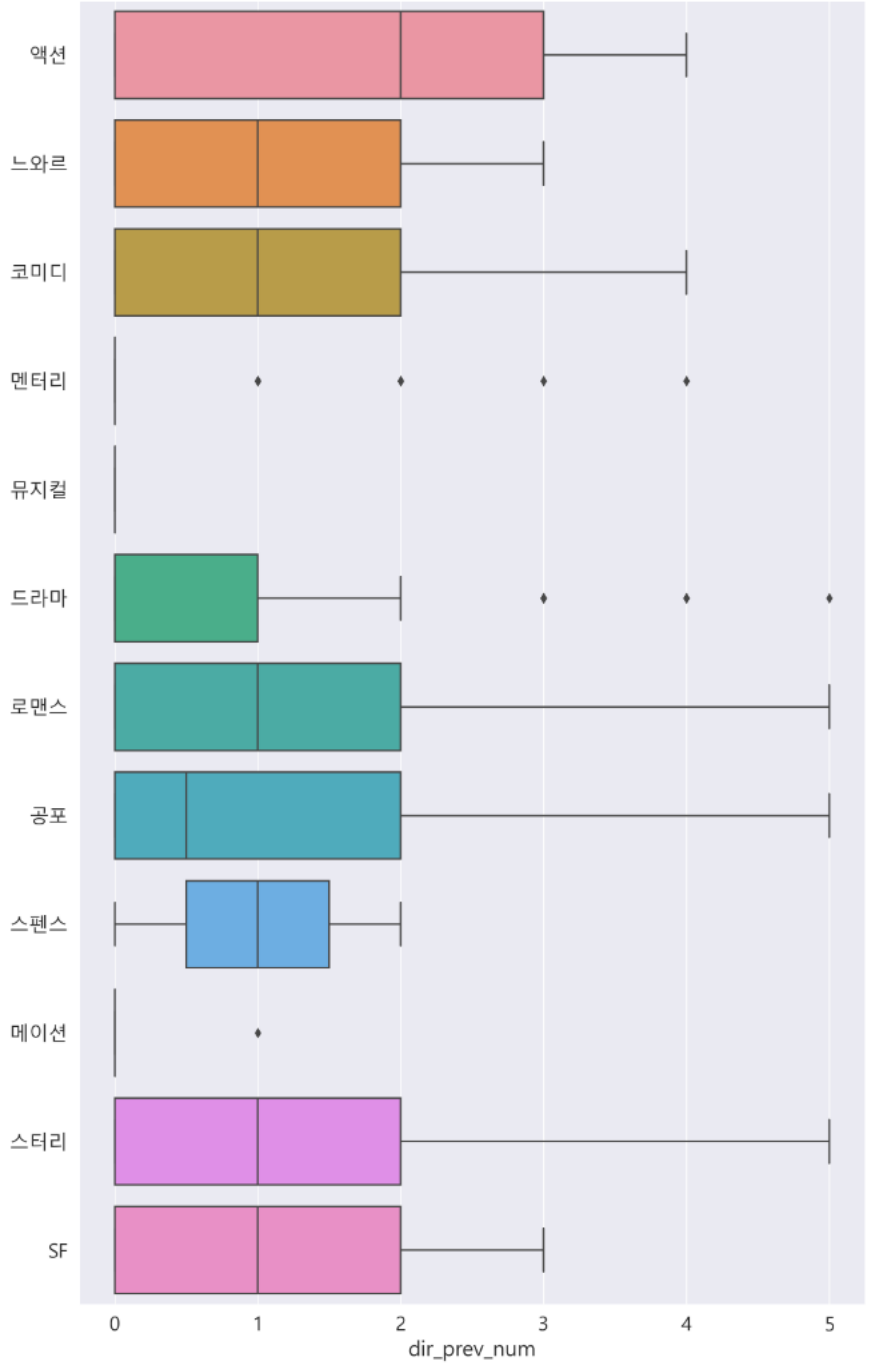
종속변수





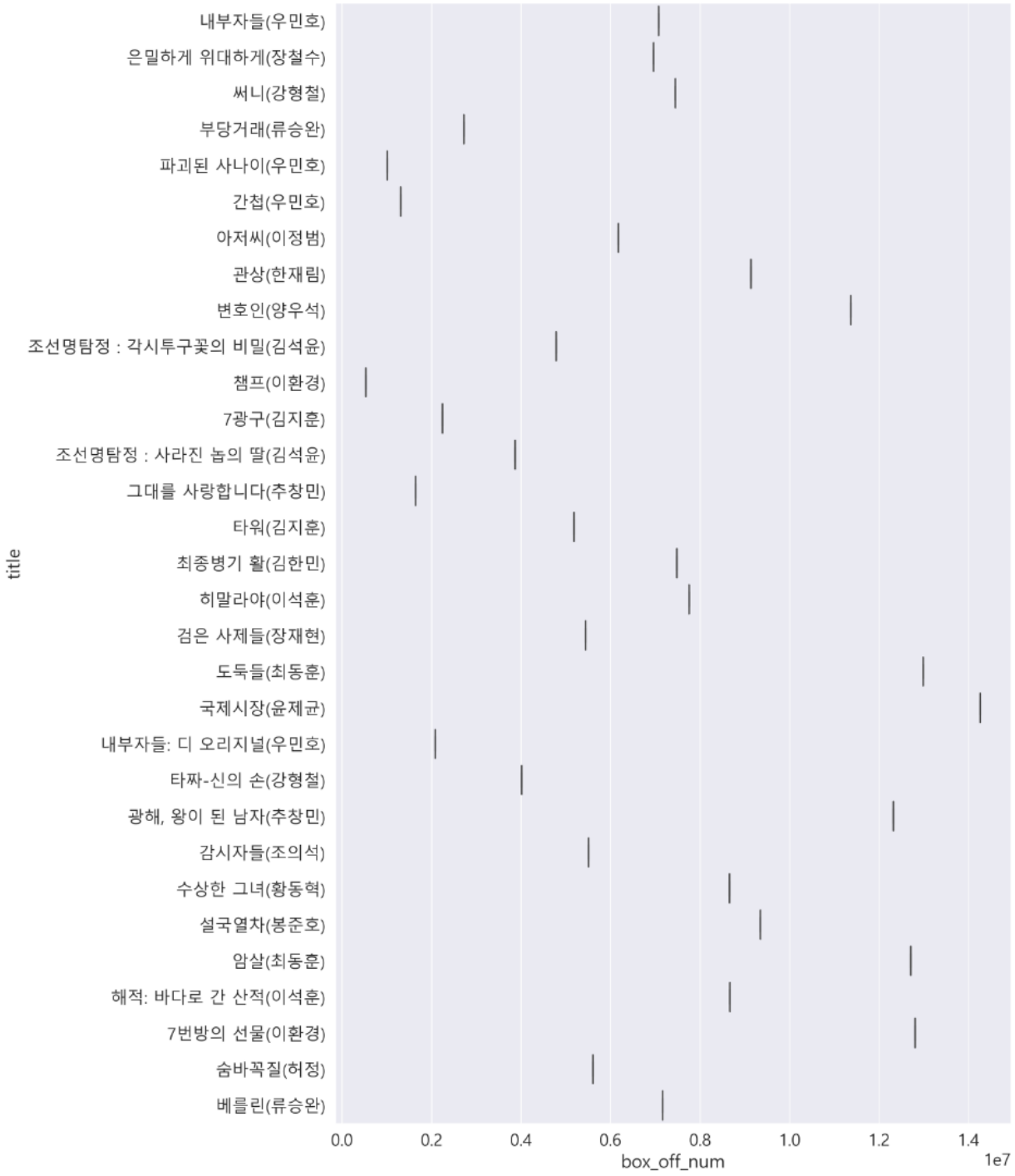
문자형인 장르 차이 탐색

장르가 무엇인지에 따라
영화 관객의 수가 변한다고
생각하여 탐색 진행



문자형인 감독 차이 탐색

감독이 누구인지에 따라
영화 관객의 수가 변한다고
생각하여 탐색 진행



모델링 전 전처리

In [69]:
```
1 df_dummies
```
executed in 20ms, finished 13:01:26 2023-04-07

Out [69]:

	time	dir_prev_bfnum	dir_prev_num	num_staff	num_actor	box_off_num	year	month	SF	공포	느와르	다큐멘터리	드라마	멜로/로맨스	뮤지컬	미스터리	서스펜스	애니메이션	액션	코미디
0	96	0.00000	0	91	2	23398	2012	11	0	0	0	0	0	0	0	0	0	0	1	0
1	130	1161602.50000	2	387	3	7072501	2015	11	0	0	1	0	0	0	0	0	0	0	0	0
2	123	220775.25000	4	343	4	6959083	2013	6	0	0	0	0	0	0	0	0	0	0	1	0
3	101	23894.00000	2	20	6	217866	2012	7	0	0	0	0	0	0	0	0	0	0	0	1
4	108	1.00000	1	251	2	483387	2010	11	0	0	0	0	0	0	0	0	0	0	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
595	111	3833.00000	1	510	7	1475091	2014	8	0	0	0	0	1	0	0	0	0	0	0	0
596	127	496061.00000	1	286	6	1716438	2013	3	0	0	0	0	1	0	0	0	0	0	0	0
597	99	0.00000	0	123	4	2475	2010	9	0	1	0	0	0	0	0	0	0	0	0	0
598	102	0.00000	0	431	4	2192525	2015	5	0	0	1	0	0	0	0	0	0	0	0	0
599	120	0.00000	0	363	5	7166532	2013	1	0	0	0	0	0	0	0	0	0	0	1	0

600 rows × 20 columns

다중공선성 판단

In [78]:

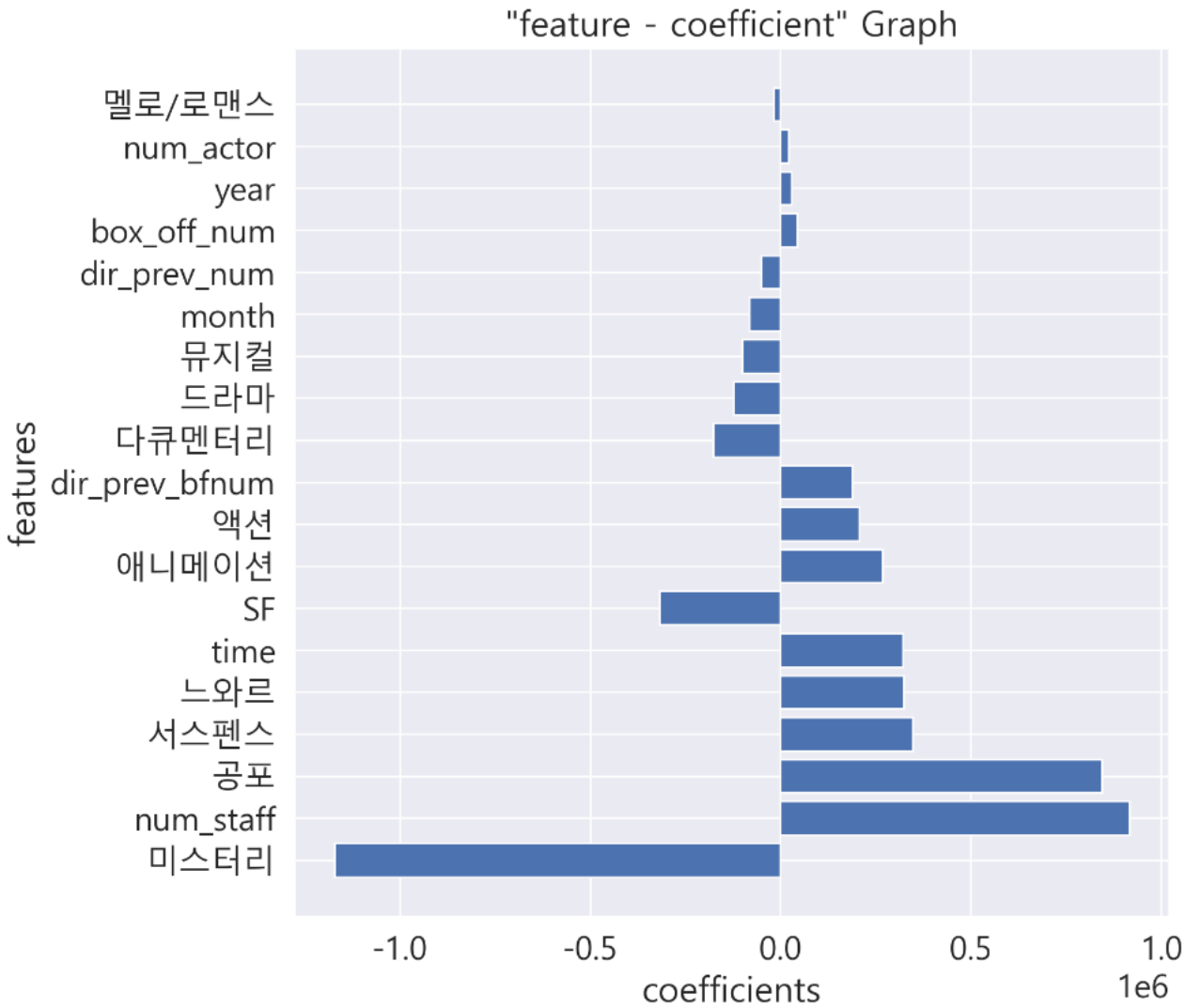
```
from statsmodels.stats.outliers_influence import variance_inflation_factor

vif = pd.DataFrame()
vif['features'] = X_train.columns
vif["VIF Factor"] = [variance_inflation_factor(X_train.values, i) for i in range(X_train.shape[1])]

vif.round(1)
```

executed in 153ms, finished 13:07:36 2023-04-07

Out[78]:

	features	VIF Factor
0	time	1.80000
1	dir_prev_bfnum	1.30000
2	dir_prev_num	1.40000
3	num_staff	2.20000
4	num_actor	1.10000
5	year	1.10000
6	month	1.00000
7	SF	1.00000
8	공포	1.00000
9	느와르	1.00000
10	다큐멘터리	1.20000
11	드라마	1.00000
12	멜로/로맨스	1.00000
13	뮤지컬	1.00000
14	미스터리	1.00000
15	서스펜스	1.00000
16	애니메이션	1.10000
17	액션	1.10000
18	코미디	1.00000

회귀모델링


"feature - coefficient" Graph
features
멜로/로맨스
num_actor
year
box_off_num
dir_prev_num
month
뮤지컬
드라마
다큐멘터리
dir_prev_bfnum
액션
애니메이션
SF
time
느와르
서스펜스
공포
num_staff
미스터리
-1.0
-0.5
0.0
0.5
1.0
coefficients
1e6

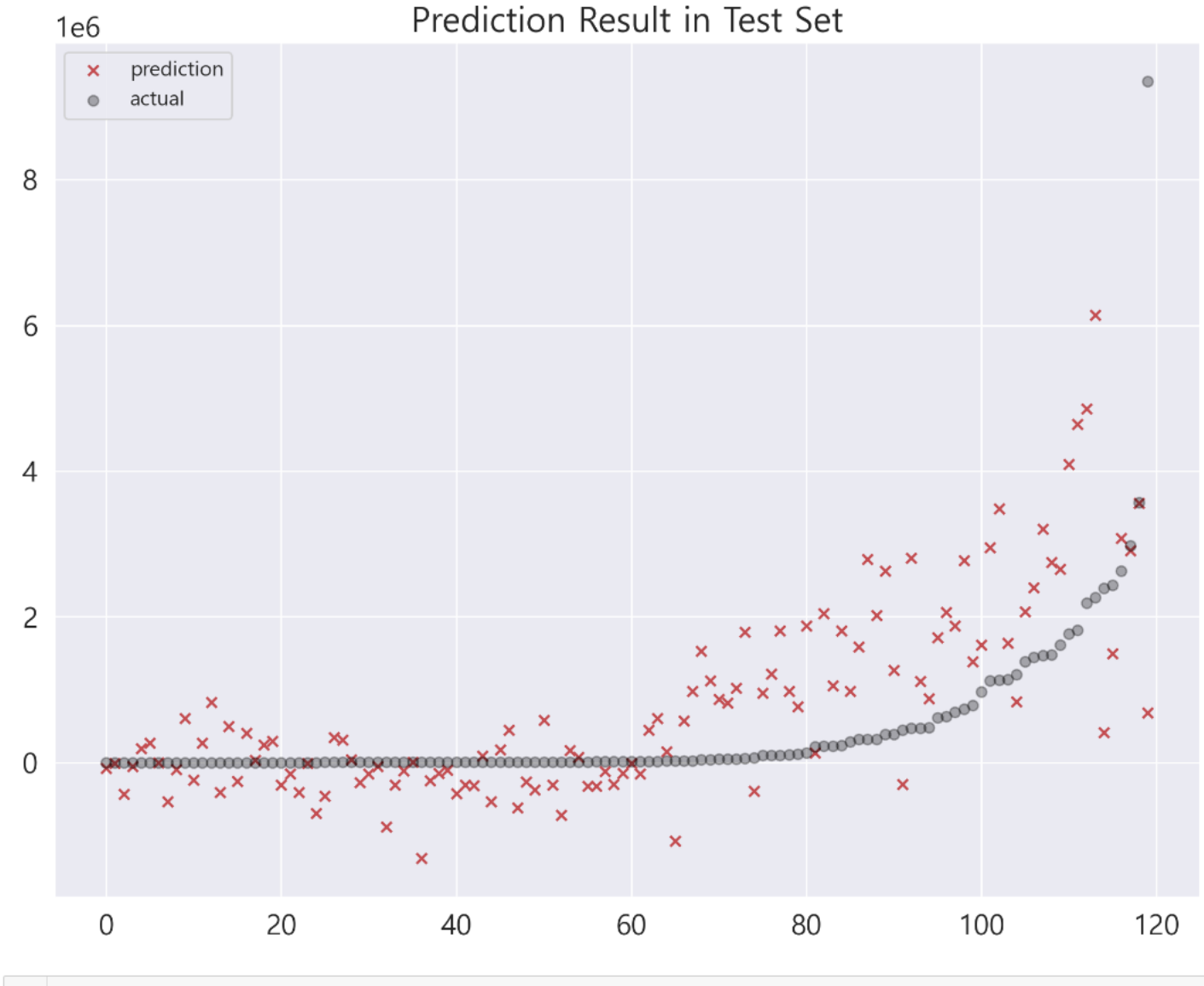
유의성 검정

OLS Regression Results

Dep. Variable:	box_off_num	**R-squared:**	0.381
Model:	OLS	**Adj. R-squared:**	0.357
Method:	Least Squares	**F-statistic:**	15.76
Date:	Fri, 07 Apr 2023	**Prob (F-statistic):**	1.09e-37
Time:	13:14:29	**Log-Likelihood:**	-7521.3
No. Observations:	480	**AIC:**	1.508e+04
Df Residuals:	461	**BIC:**	1.516e+04
Df Model:	18		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	6.991e+05	1.55e+05	4.508	0.000	3.94e+05	1e+06
time	3.213e+05	9.69e+04	3.317	0.001	1.31e+05	5.12e+05
dir_prev_bfnum	1.888e+05	7.68e+04	2.459	0.014	3.79e+04	3.4e+05
dir_prev_num	-5.141e+04	8.66e+04	-0.594	0.553	-2.22e+05	1.19e+05
num_staff	9.162e+05	1.07e+05	8.532	0.000	7.05e+05	1.13e+06
num_actor	2.224e+04	7.05e+04	0.316	0.753	-1.16e+05	1.61e+05
year	4.439e+04	7.3e+04	0.608	0.543	-9.9e+04	1.88e+05
month	2.853e+04	7.38e+04	0.387	0.699	-1.17e+05	1.74e+05
SF	-2.364e+04	5.14e+05	-0.046	0.963	-1.03e+06	9.87e+05
공포	-2.597e+05	2.89e+05	-0.899	0.369	-8.27e+05	3.08e+05
느와르	9.018e+05	3.71e+05	2.429	0.016	1.72e+05	1.63e+06
다큐멘터리	3.813e+05	2.39e+05	1.593	0.112	-8.9e+04	8.52e+05
드라마	-1.181e+05	1.91e+05	-0.619	0.536	-4.93e+05	2.57e+05
멜로/로맨스	-6.468e+04	2.44e+05	-0.265	0.791	-5.45e+05	4.15e+05
뮤지컬	4.136e+04	6.79e+05	0.061	0.951	-1.29e+06	1.38e+06
미스터리	-4.31e+04	4.2e+05	-0.103	0.918	-8.69e+05	7.83e+05
서스펜스	-1.114e+06	1.46e+06	-0.761	0.447	-3.99e+06	1.76e+06
애니메이션	4.062e+05	4e+05	1.015	0.311	-3.81e+05	1.19e+06
액션	3.261e+05	3.43e+05	0.951	0.342	-3.48e+05	1e+06
코미디	2.659e+05	2.78e+05	0.955	0.340	-2.81e+05	8.13e+05

Omnibus:	372.924	**Durbin-Watson:**	1.775
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	6455.753
Skew:	3.282	**Prob(JB):**	0.00
Kurtosis:	19.724	**Cond. No.**	8.75e+15



모델의 시각화 및 성능평가

In [91]:

```
print(model.score(X_train, y_train))
print(model.score(X_test, y_test))
```

executed in 12ms, finished 13:20:57 2023-04-07

```
0.38089505880785524
-0.4318416495644859
```

In [93]:

```
# RMSE
# RMSE
from sklearn.metrics import mean_squared_error
from math import sqrt

# training set
pred_train = lr.predict(X_train)
print(sqrt(mean_squared_error(y_train, pred_train)))
# train error 구함

# test set
print(sqrt(mean_squared_error(y_test, pred_test)))
# test error 구함
```

executed in 19ms, finished 13:21:13 2023-04-07

```
1544906.5426182372
1297485.689910567
```

감사합니다.