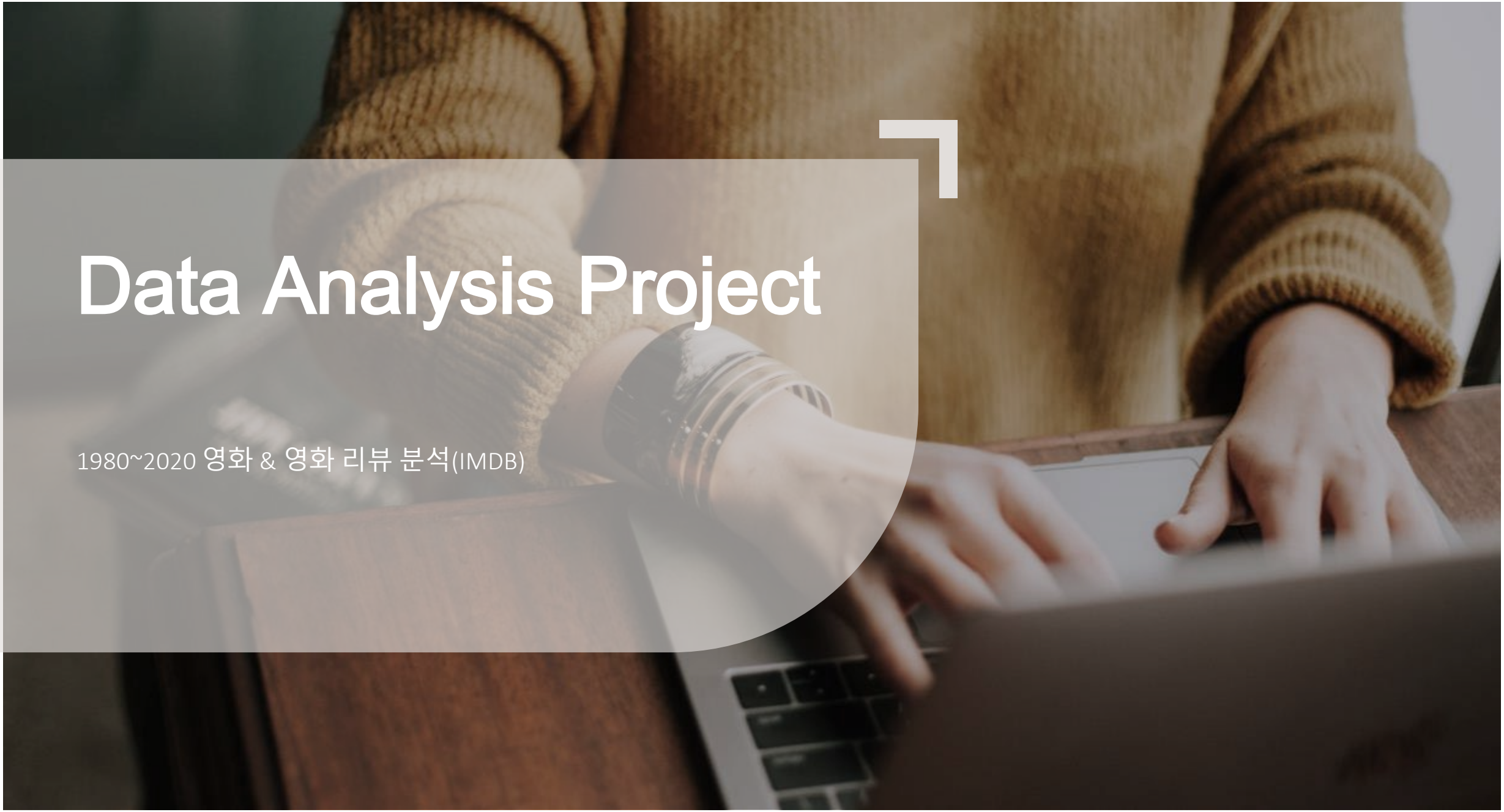


Data Analysis Project

1980~2020 영화 & 영화 리뷰 분석(IMDB)

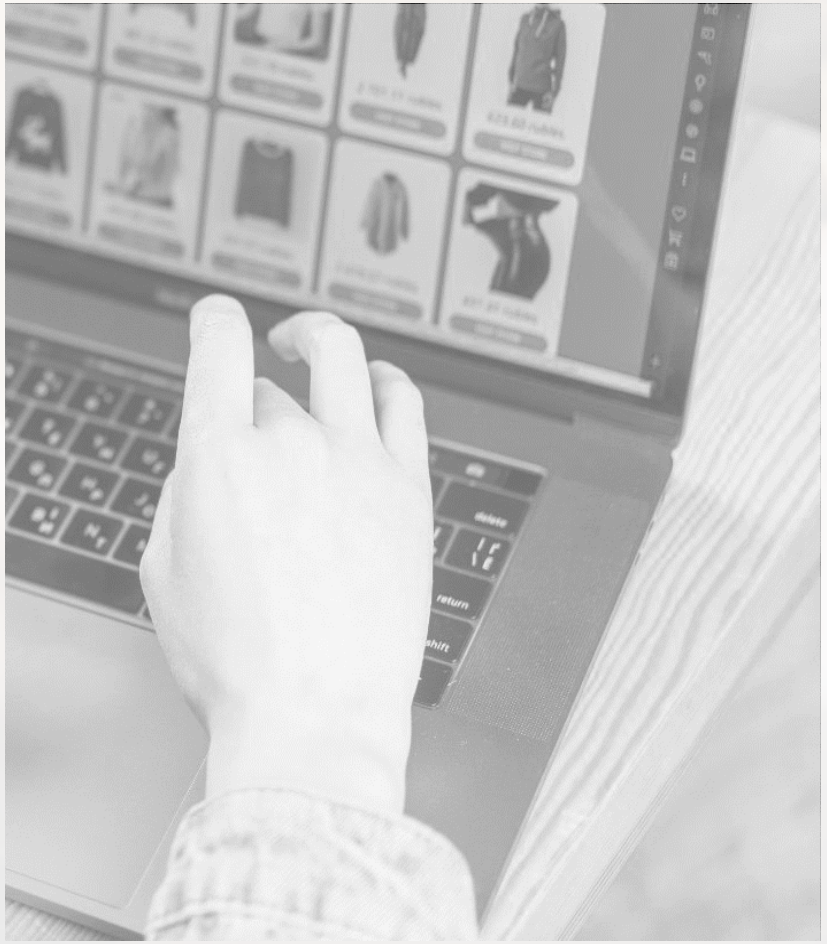
데이터 개요

정형데이터
❖ movies.csv – 변수 14개, 표본 7,652개
비정형데이터
❖ lmdb.csv – 변수 2개, 표본 5,000개
데이터 출처
◆ Kaggle.com
주요 사용 툴
● python

목차

전처리

1. 빈 값 검색해서 처리
2. Rating을 6가지로 분류
3. 장르를 8가지로 분류
4. 국가를 대륙으로 나눠서 7가지로 분류
5. 원핫인코딩
6. 문자로 된 필드 제거
7. 종속변수 0,1로 분류
8. 언더샘플링
9. 학습용:검증용=8:2

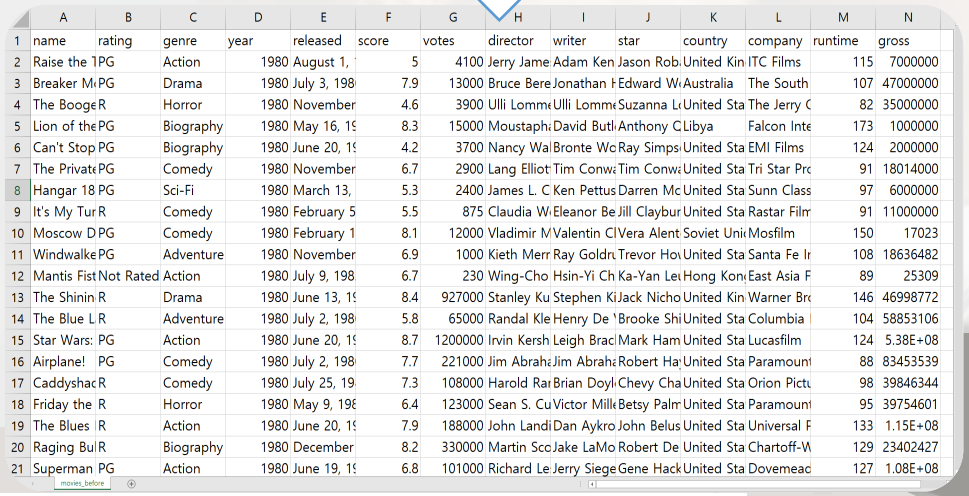


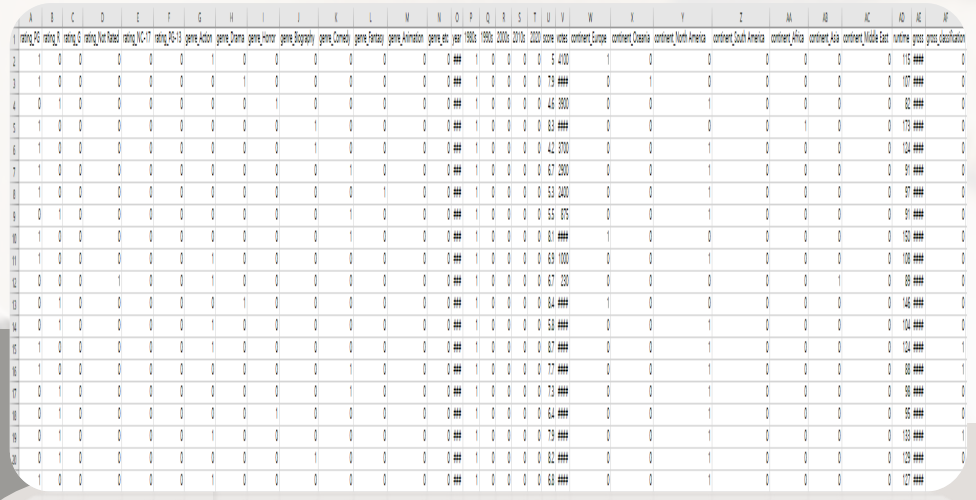
전처리

제목 | 관람등급 | 장르 | 개봉연도 | 개봉일자 | 평점 | 투표수 | 감독 | 작가 | 출연배우 | 제작국가 | 제작사 | 상영시간 | 총수입

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	name	rating	genre	year	released	score	votes	director	writer	star	country	company	runtime	gross

movies.csv





movies_y.csv

G: General Audiences: 모든 관객 (모든 연령층 적합)
PG: Parental Guidance Suggested: 부모 동반 (아동 관람 부적합)
PG-13: Parental Strongly Cautioned: 부모 주의(부분적 13세 미만 부적합)
R: Restricted: 제한 (17세 미만은 부모나 성인 보호자 동반 요망)
NC-17: No One 17 And Under Admitted: 18세 미만은 관람할 수 없는 영화
Not Rated: 등급이 메겨지지 않은 영화

관람등급별 원핫인코딩

장르별 원핫인코딩

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	rating_PG	rating_R	rating_G	rating_Not Rated	rating_NC-17	rating_PG-13	genre_Action	genre_Drama	genre_Horror	genre_Biography	genre_Comedy	genre_Fantasy	genre_Animation	genre_etc	year

P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF
1980s	1990s	2000s	2010s	2020	score	votes	continent_Europe	continent_Oceania	continent_North America	continent_South America	continent_Africa	continent_Asia	continent_Middle East	runtime	gross	gross_classification

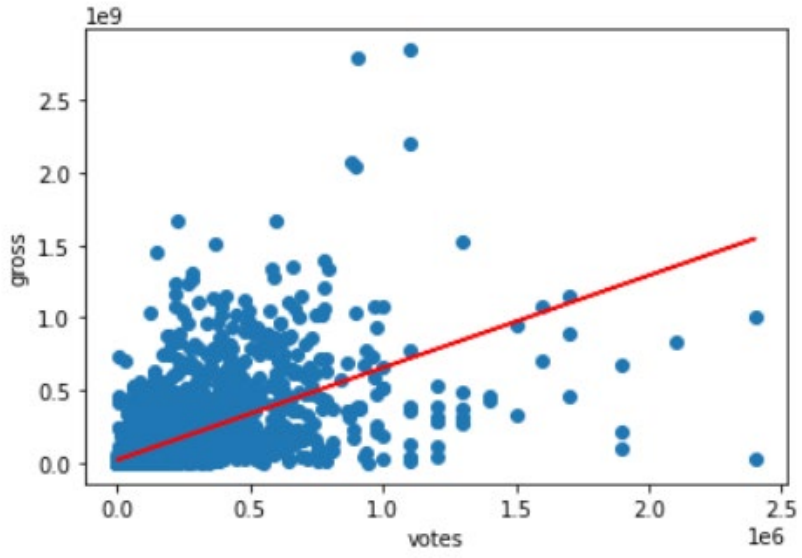
연도별 원핫인코딩

대륙별 원핫인코딩

회귀분석 - 단순회귀분석

독립변수: 투표수
종속변수: 총수입
귀무가설: 투표수와 총수입은 상관관계가 없다.
대립가설: 투표수와 총수입은 상관관계가 있다.

```
LinregressResult(slope=635.717924870875, intercept=20722621.796298243, rvalue=0.632618503879609, pvalue=0.0, stderr=8.898590718911574,
intercept_stderr=1652746.984876386)
Text(0, 0.5, 'gross')
```



사회과학에서는 어떤 현상을 100% 설명하는 것은 현실적으로 불가능해서 결정계수가 0.3이상이면 의미가 있다고 본다.

01 R-value

02 P-value

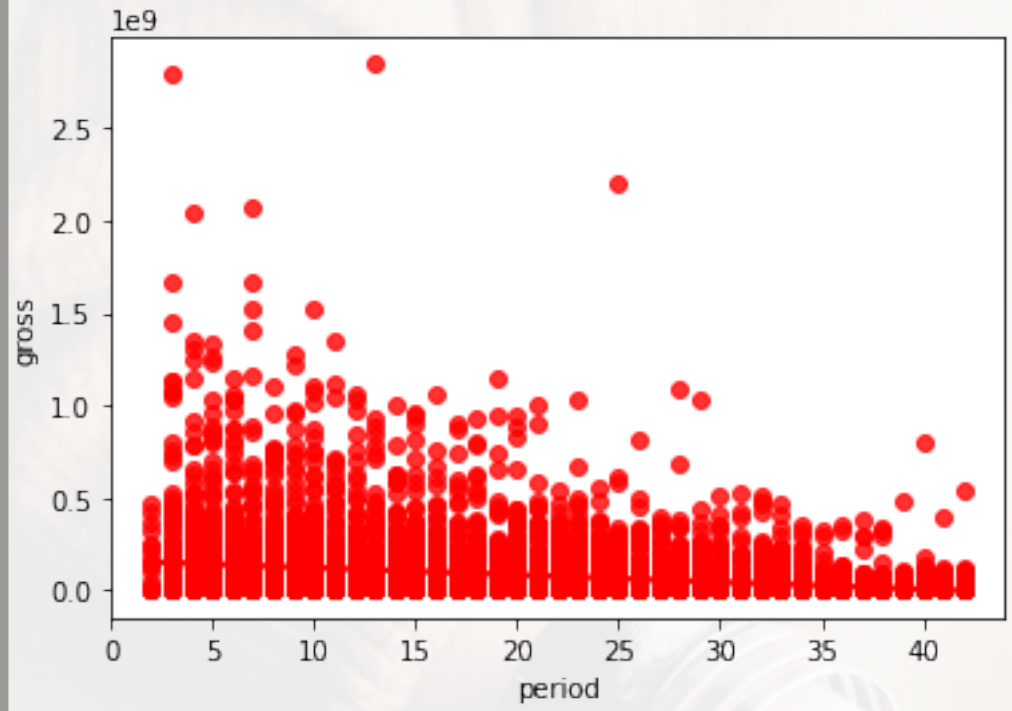
0.05보다 작으므로 통계적으로 유의하다.

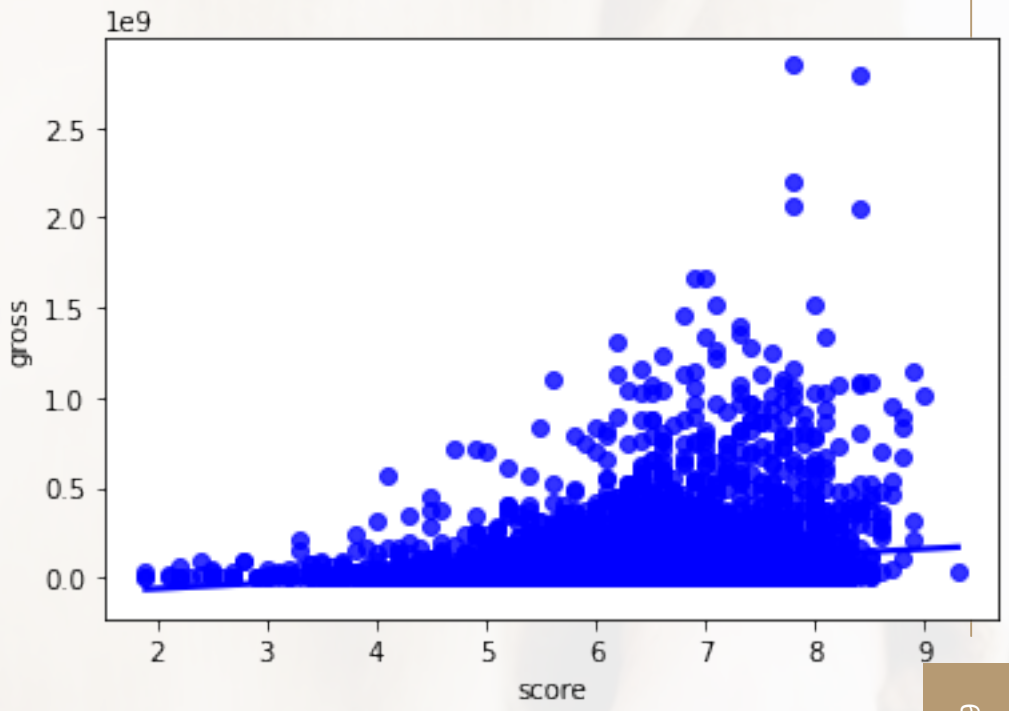
03 **결론**

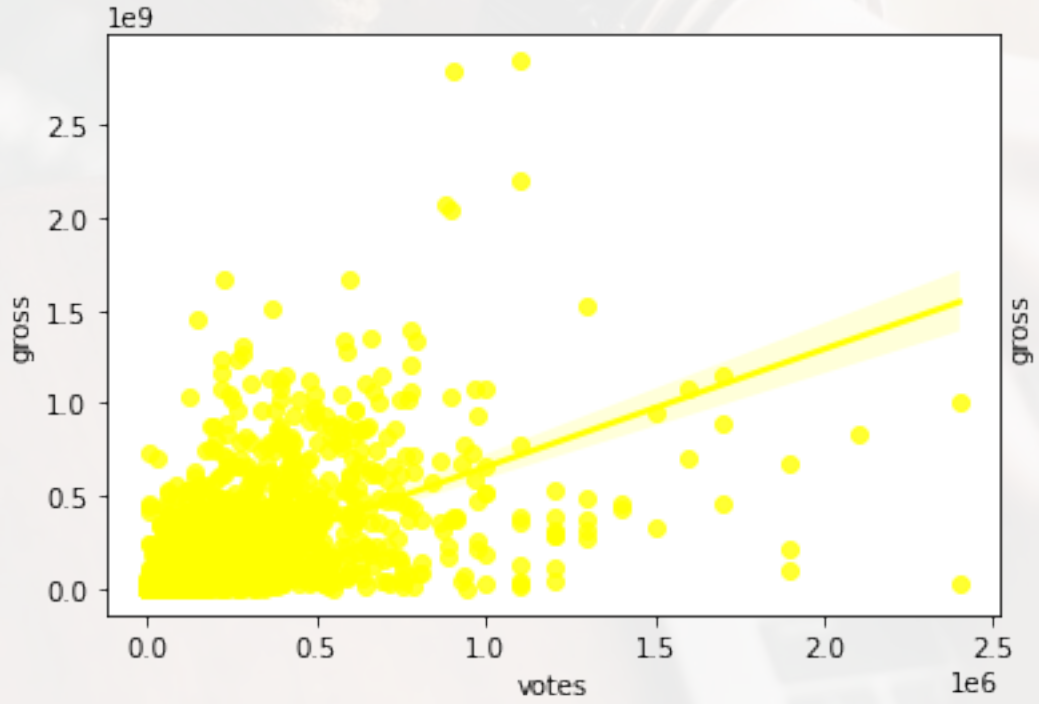
모형적합도가 63%의 설명력이 있다.

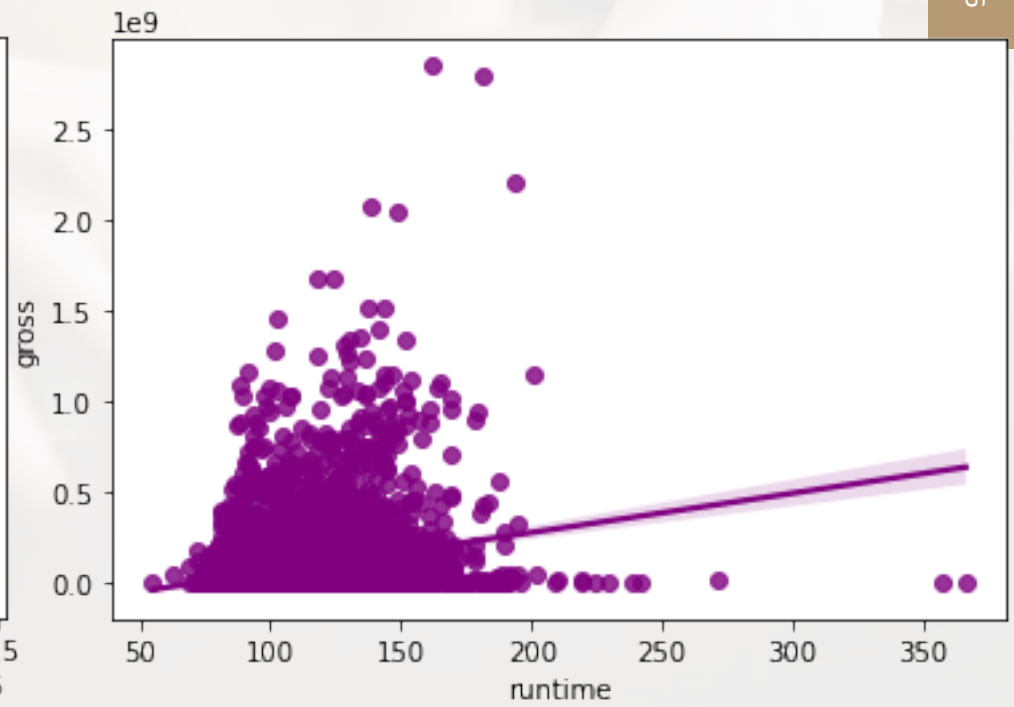
투표수와 총수입은 상관관계가 있다.

다중 회귀 분석


1e9
gross
period


1e9
gross
score


1e9
gross
votes
1e6


1e9
gross
runtime

다중회귀분석

R-squared:

0.428

OLS Regression Results

Dep. Variable:	gross	R-squared:	0.428
Model:	OLS	Adj. R-squared:	0.427
Method:	Least Squares	F-statistic:	1429.
Date:	Mon, 01 Aug 2022	Prob (F-statistic):	0.00
Time:	13:49:59	Log-Likelihood:	-1.5345e+05
No. Observations:	7651	AIC:	3.069e+05
Df Residuals:	7646	BIC:	3.070e+05
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	1.082e+08	1.2e+07	9.032	0.000	8.47e+07	1.32e+08
period	-1.828e+06	1.31e+05	-13.956	0.000	-2.09e+06	-1.57e+06
score	-1.967e+07	1.7e+06	-11.600	0.000	-2.3e+07	-1.63e+07
votes	629.9613	9.855	63.924	0.000	610.643	649.280
runtime	7.289e+05	8.48e+04	8.591	0.000	5.63e+05	8.95e+05

Omnibus:	7112.185	Durbin-Watson:	1.902
Prob(Omnibus):	0.000	Jarque-Bera (JB):	899317.821
Skew:	4.044	Prob(JB):	0.00
Kurtosis:	55.494	Cond. No.	1.57e+06

P-value:

0

현재와 기한이
짧을수록
총수입이 증가

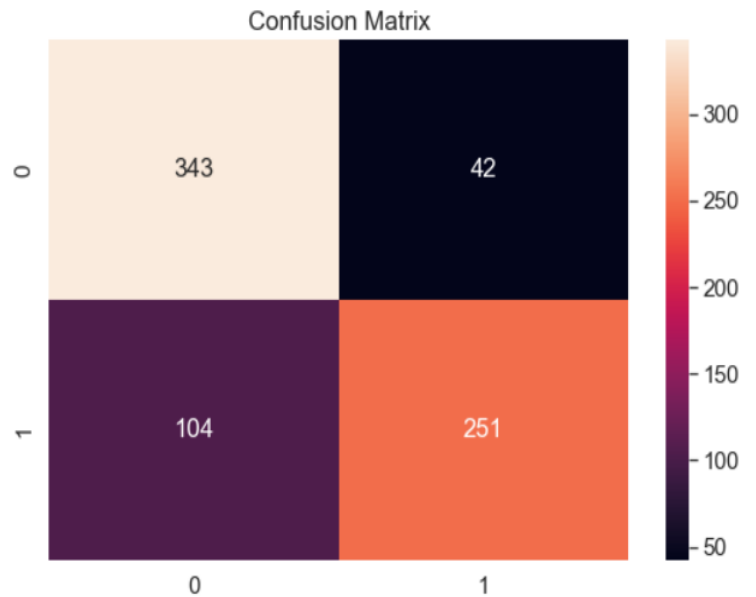
투표수가 높을수록
총수입이 증가

상영시간이 길수록
총수입이 증가

분류 - 로지스틱회귀분석

연도, 투표수, 평점, 상영시간이 총수입과 상관관계가 있다고 볼 수 있는 확률

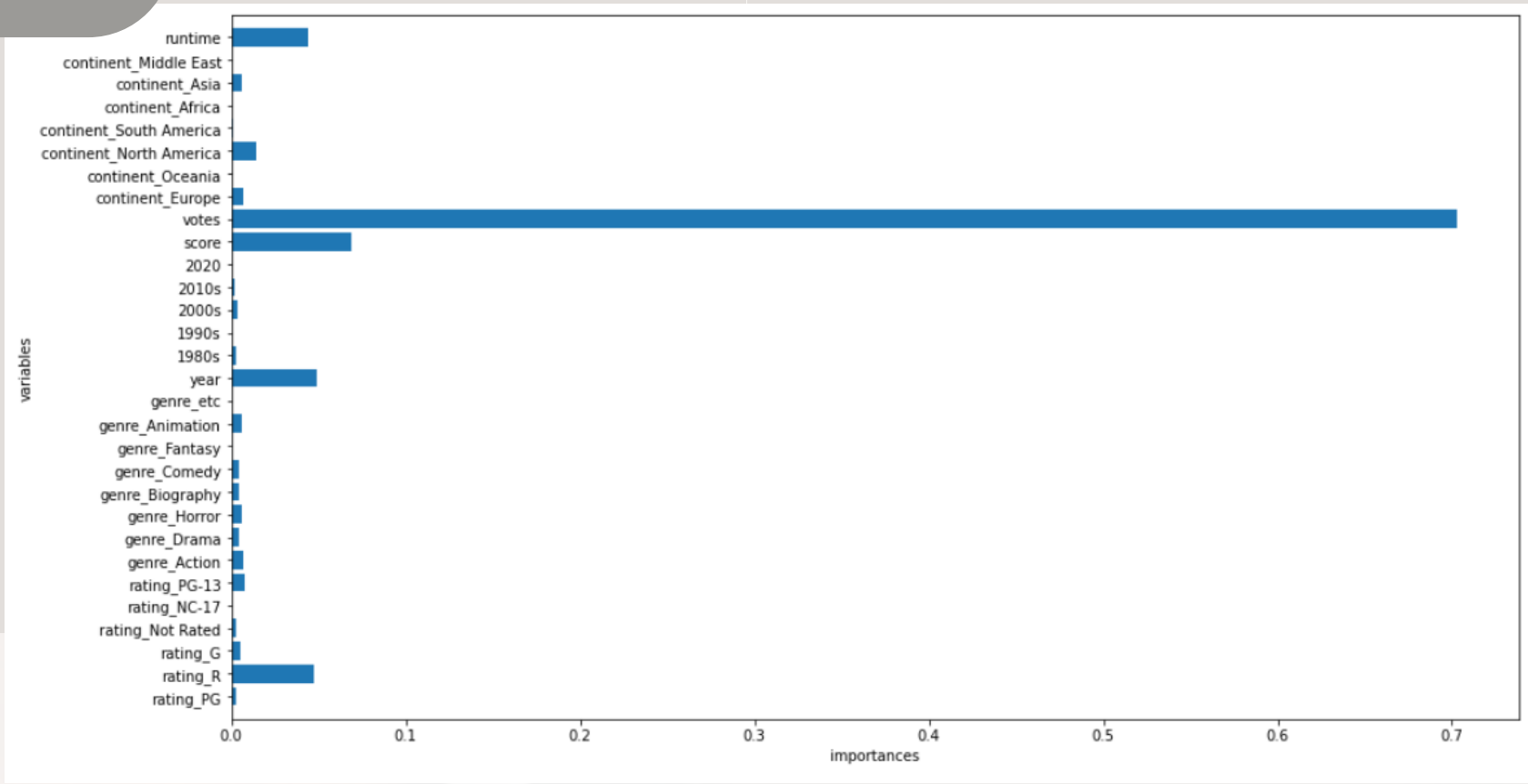
학습용: 0.8139377537212449
검증용: 0.8027027027027027



정확도: 80%

의사결정나무

```
model1 = DecisionTreeClassifier(random_state=0, max_depth=8)
DecisionTreeClassifier(max_depth=8, random_state=0)
 학습용: 0.9218538565629228
 검증용: 0.8486486486486486
```



정확도: 84%

Tree – splitter

노드분할전략 :
최선의 변수 선택 전략(기본 옵션)

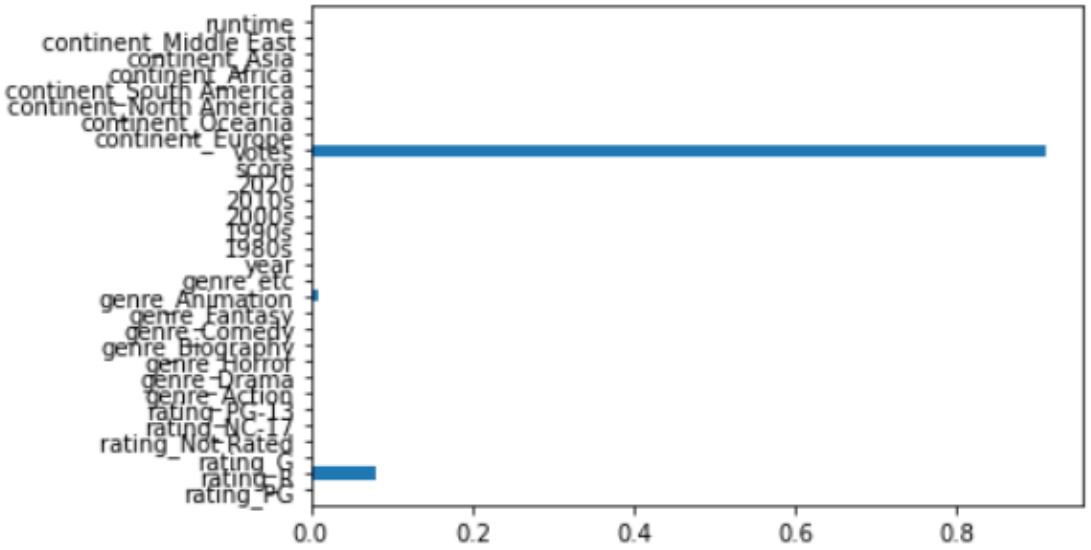
```
DecisionTreeClassifier
DecisionTreeClassifier(max_depth=3, random_state=0)
```

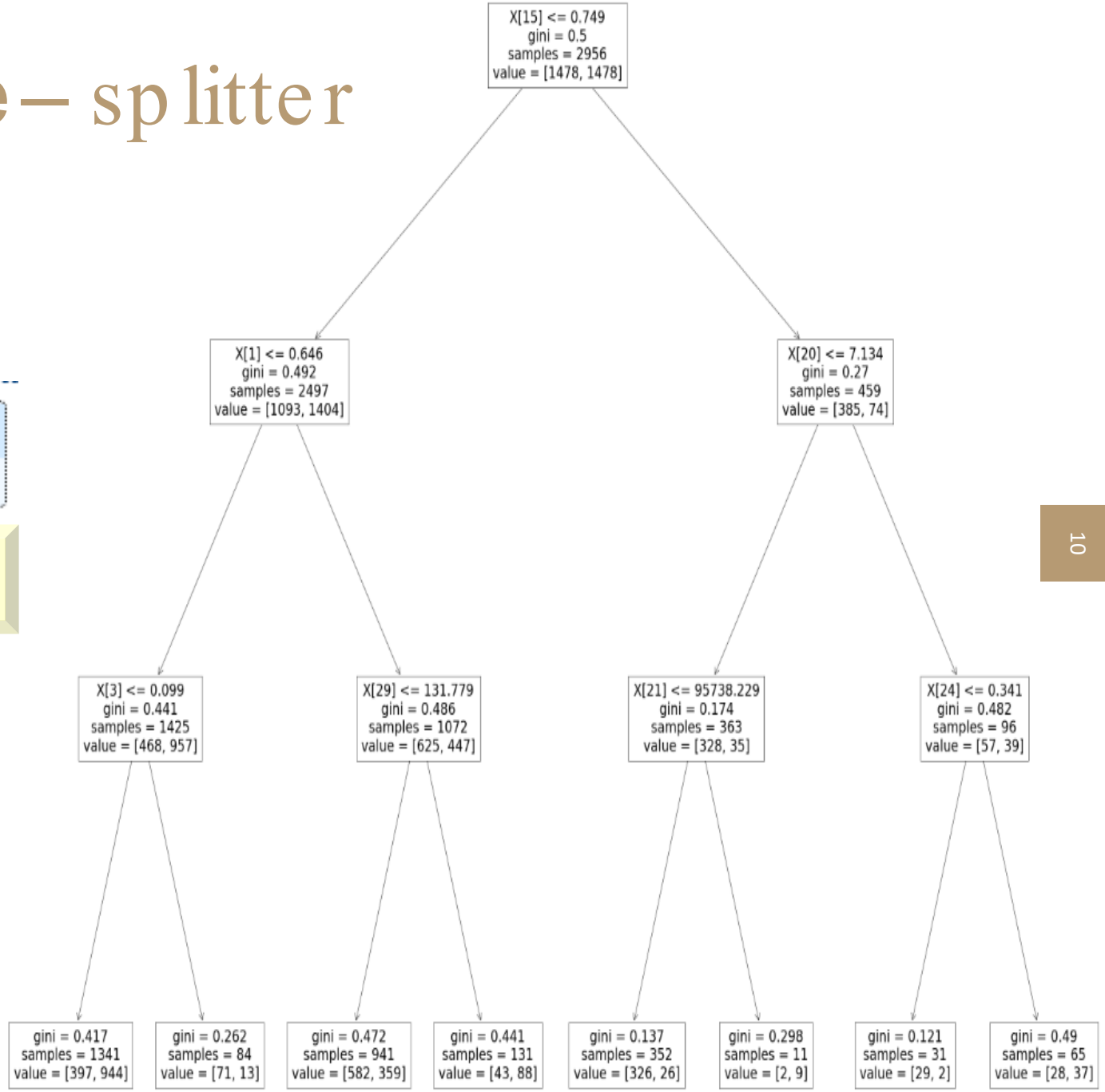
학습용: 0.8525033829499323

검증용: 0.8243243243243243

정확도: 82%

특성중요도





Tree – splitter

노드분할전략 :
랜덤 분할 전략

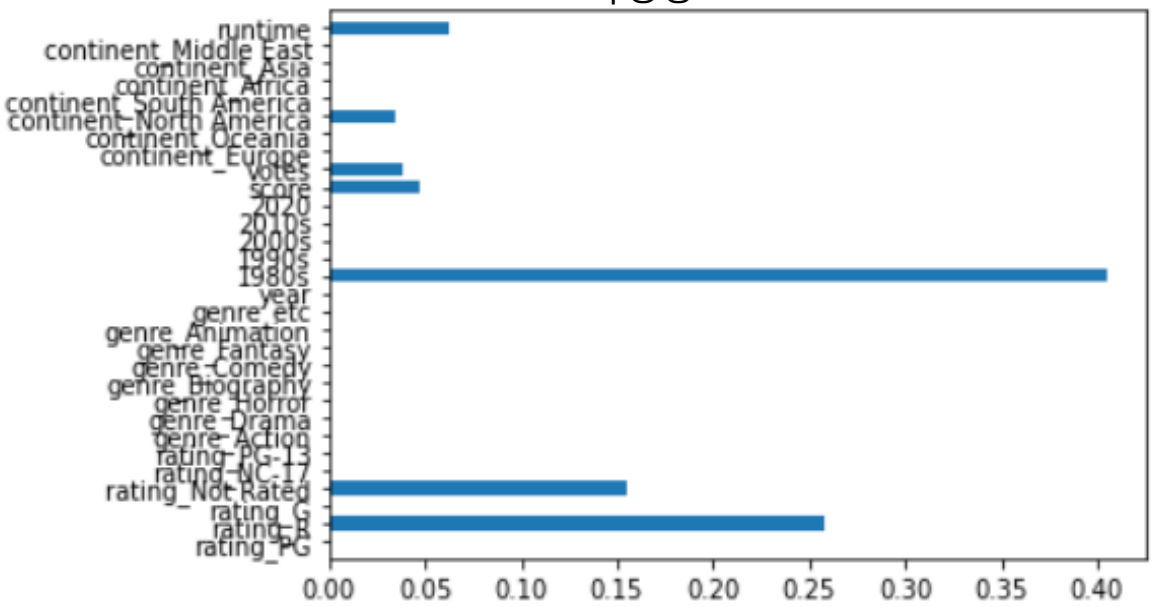
```
DecisionTreeClassifier
DecisionTreeClassifier(max_depth=3, random_state=0, splitter='random')
```

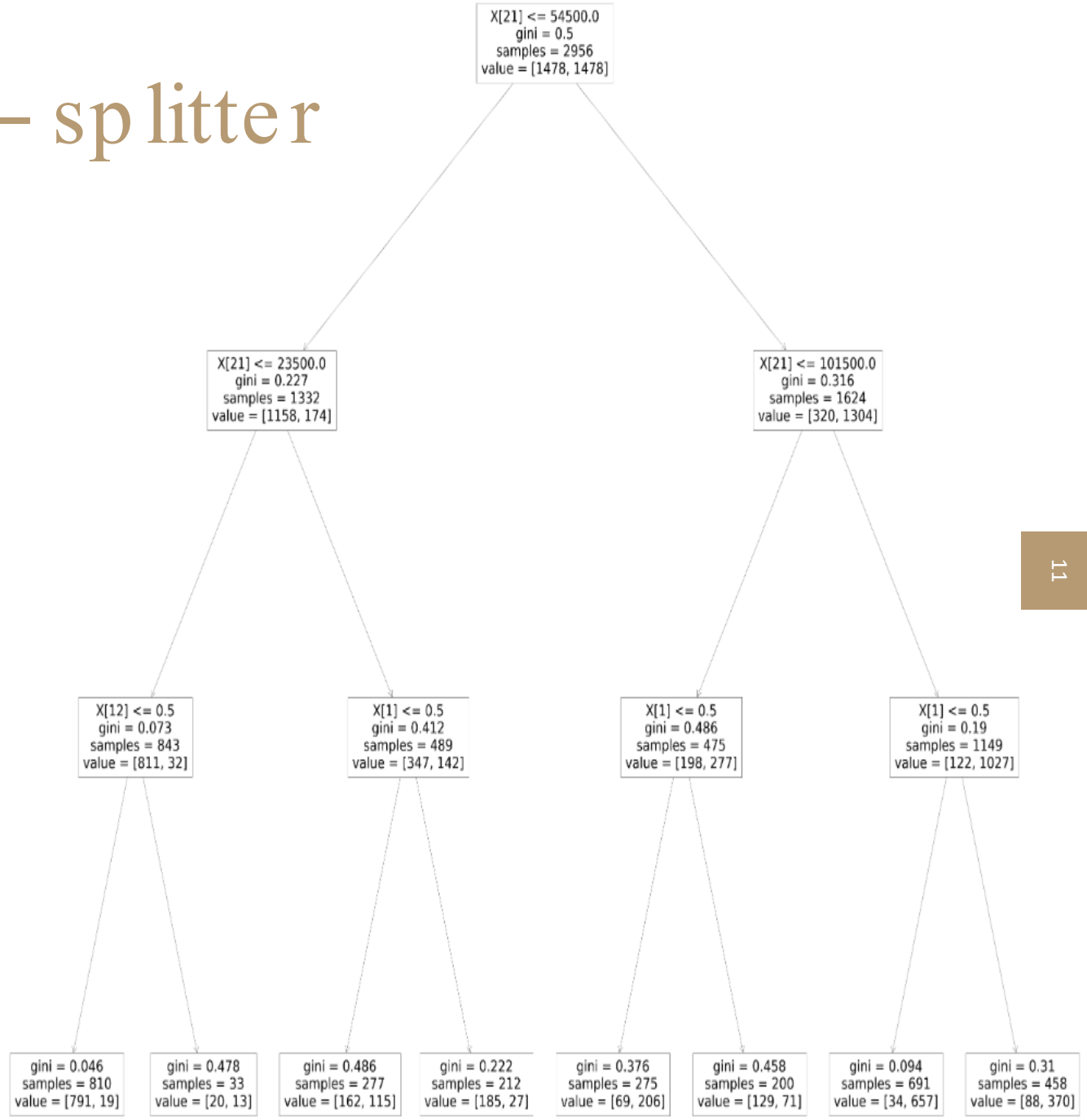
학습용: 0.7056833558863329

검증용: 0.6918918918918919

정확도: 69%

특성중요도

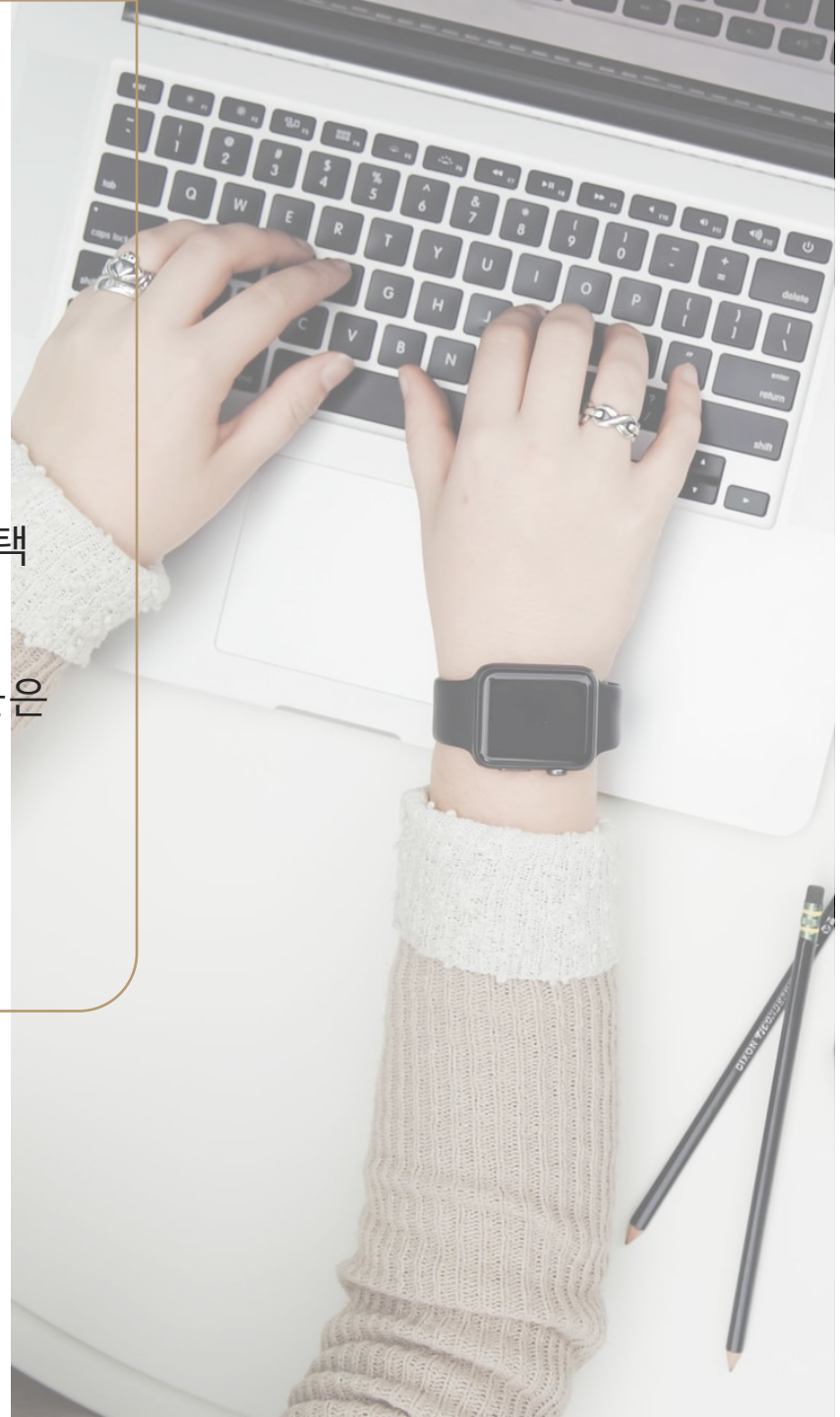




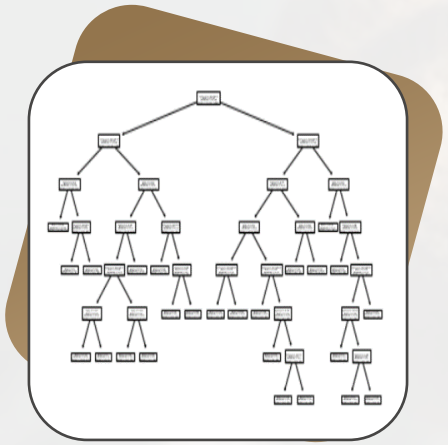
Tree – splitter

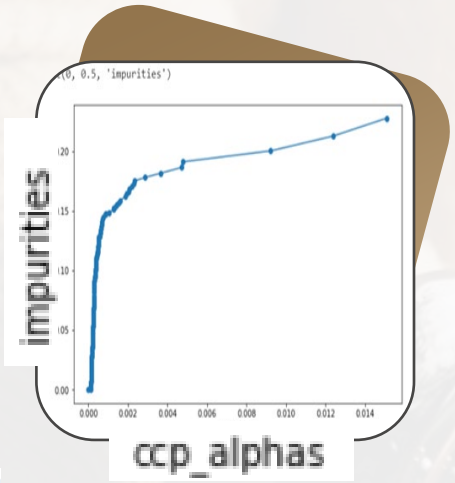
I. 랜덤분할전략보다 최선의 변수 선택 전략이 정확도가 더 높다.

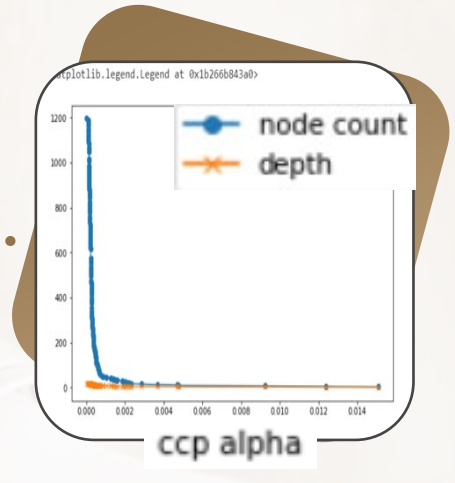
II. 최선의 변수 선택 전략에서 가장 높은 중요도를 보였던 votes가 랜덤분할전략에서는 그다지 높게 나오지 않았다.

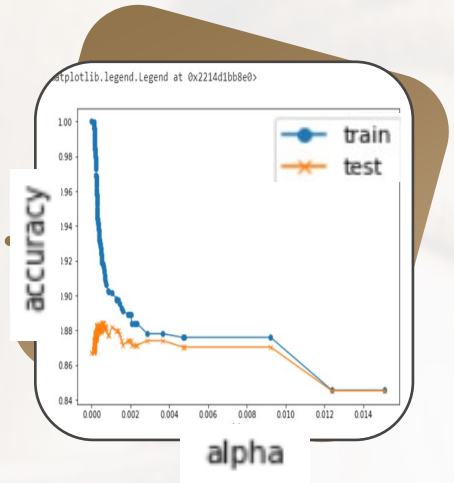


Tree-ccp-alpha










```python
import matplotlib.pyplot as plt
from sklearn import tree
clf = DecisionTreeClassifier(random_state=0, ccp_alpha=0.001)
clf.fit(X_train, y_train)
plt.Figure(figsize=(10,5))
tree.plot_tree(clf)
plt.show()
```

✓ 2.6s Python

ccp_alpha가 증가하면 가지치기된 노드수 증가
=> 불순도가 증가함
ccp_alpha가 감소하면 가지치기된 노드수 감소
=> 불순도가 감소함

ccp alpha가 증가하면
노드 개수가 감소한다.

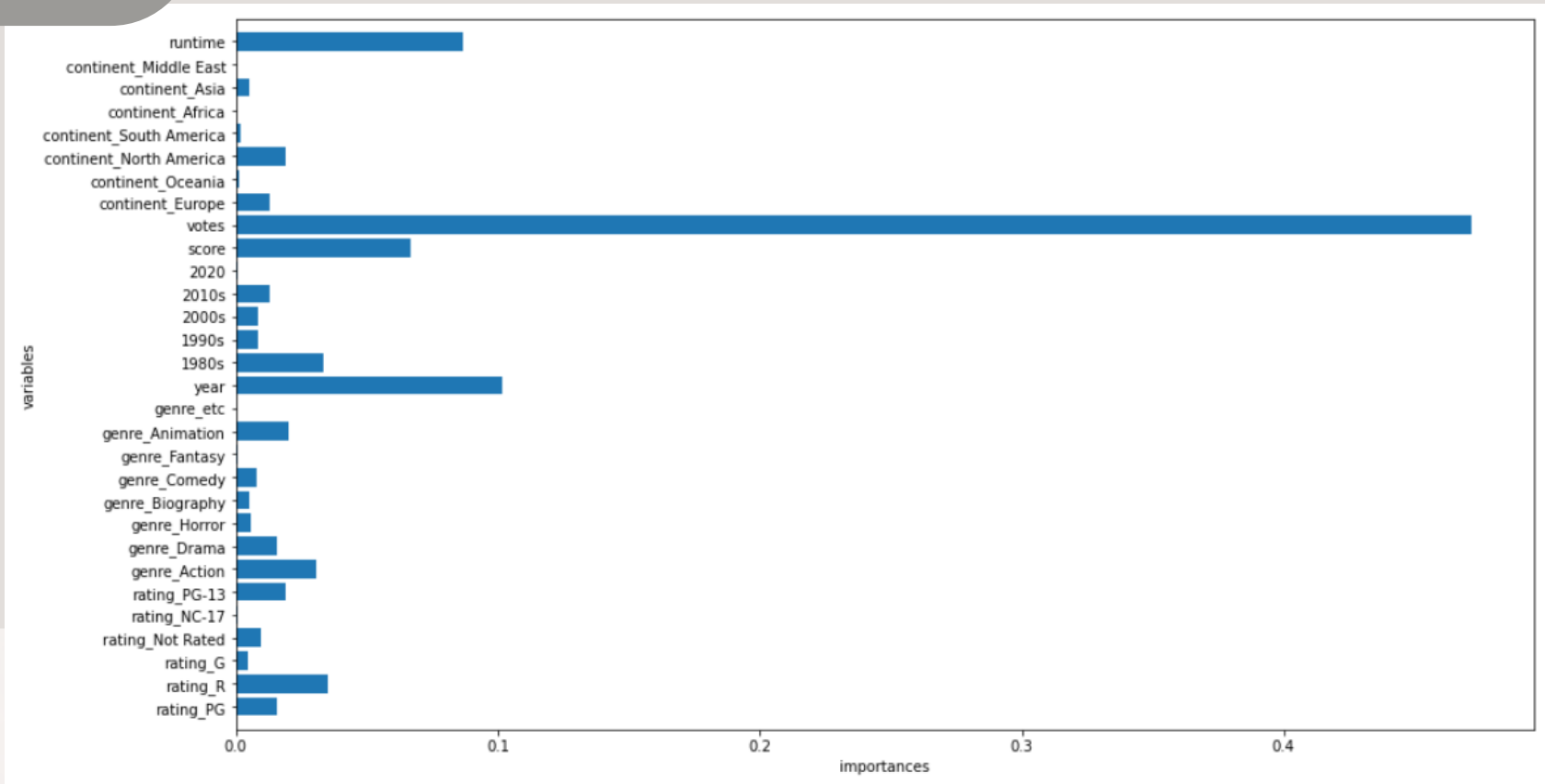
최적의 alpha: 0.0005228302544440571

최고 정확도:
88%

모형결합 - 랜덤포레스트

```
model2 = RandomForestClassifier(n_estimators=1000, random_state=0, max_depth=10)
```

```
RandomForestClassifier(max_depth=10, n_estimators=1000, random_state=0)
학습용: 0.9448579161028416
검증용: 0.8770270270270271
```



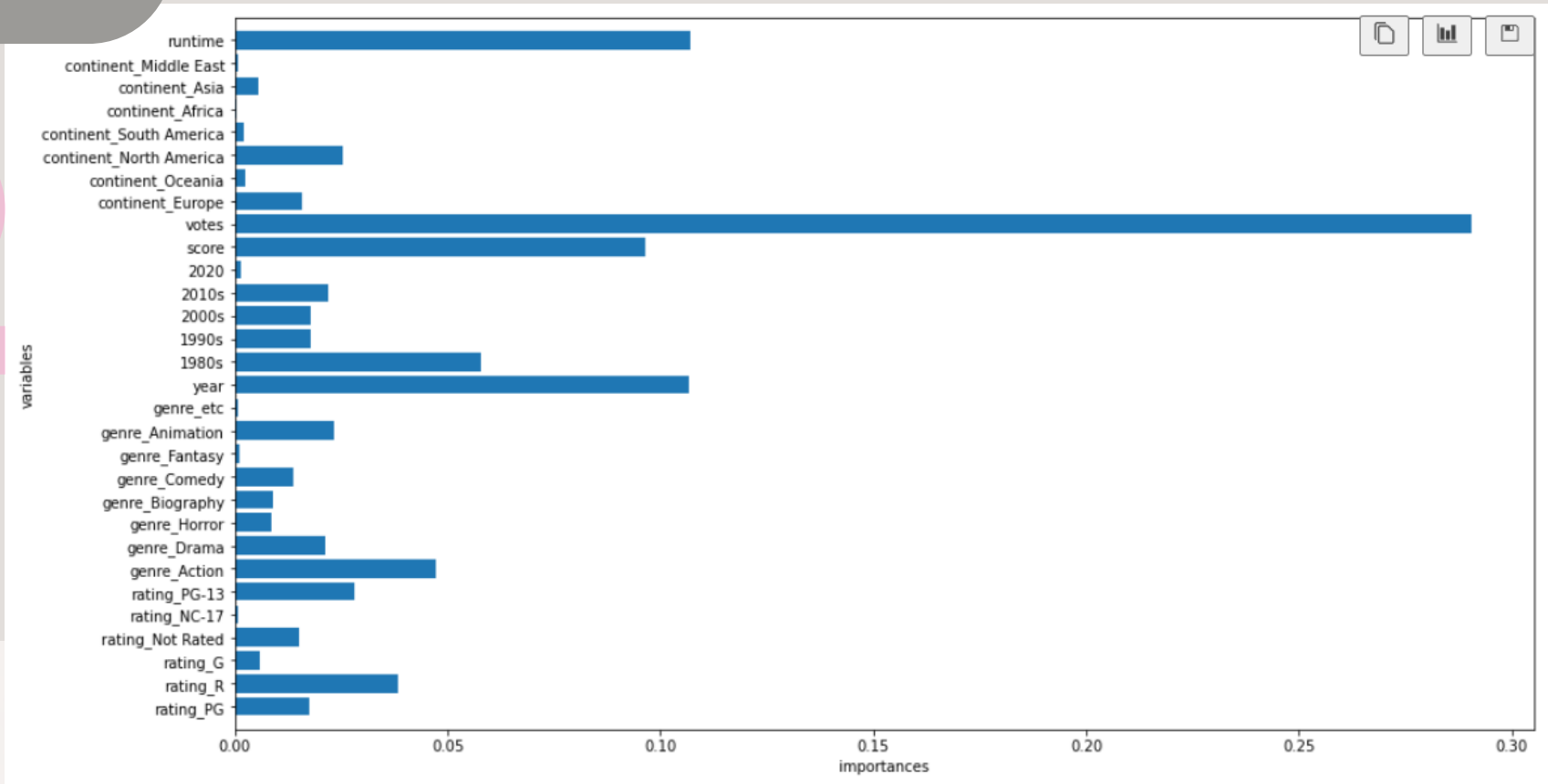
정확도: 87%

모형결합 - 엑스트라트리

```
model3 = ExtraTreesClassifier(n_estimators=1000, random_state=0, max_depth=16)
ExtraTreesClassifier(max_depth=16, n_estimators=1000, random_state=0)
학습용: 0.9935723951285521
검증용: 0.8432432432432433
```

엑스트라 트리는 포레스트 트리의 각 후보 특성을 무작위로 분할하는 식으로 무작위성을 증가 시킨다.

정확도: 84%



모형결합

정확도

01

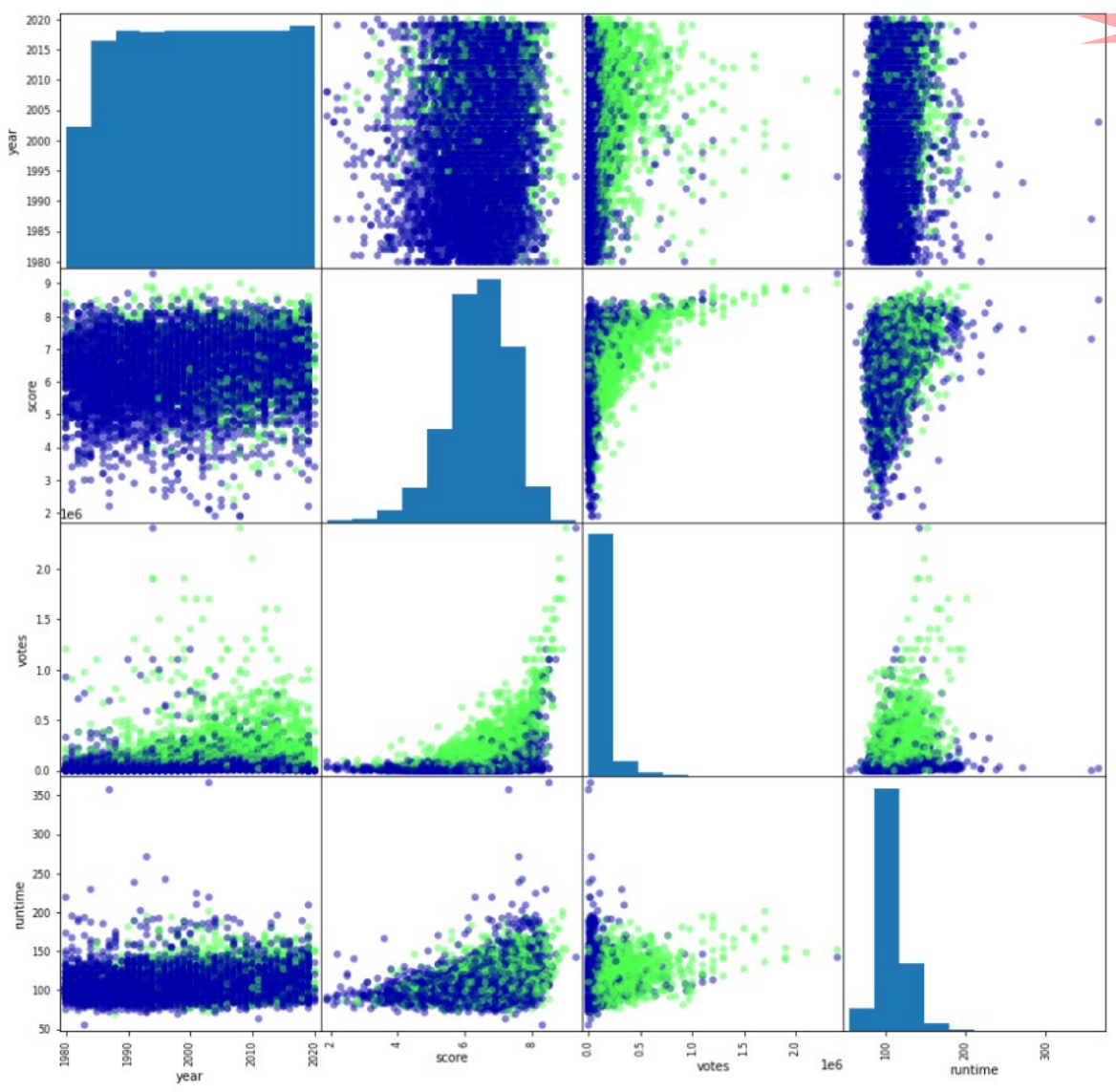
랜덤포레스트

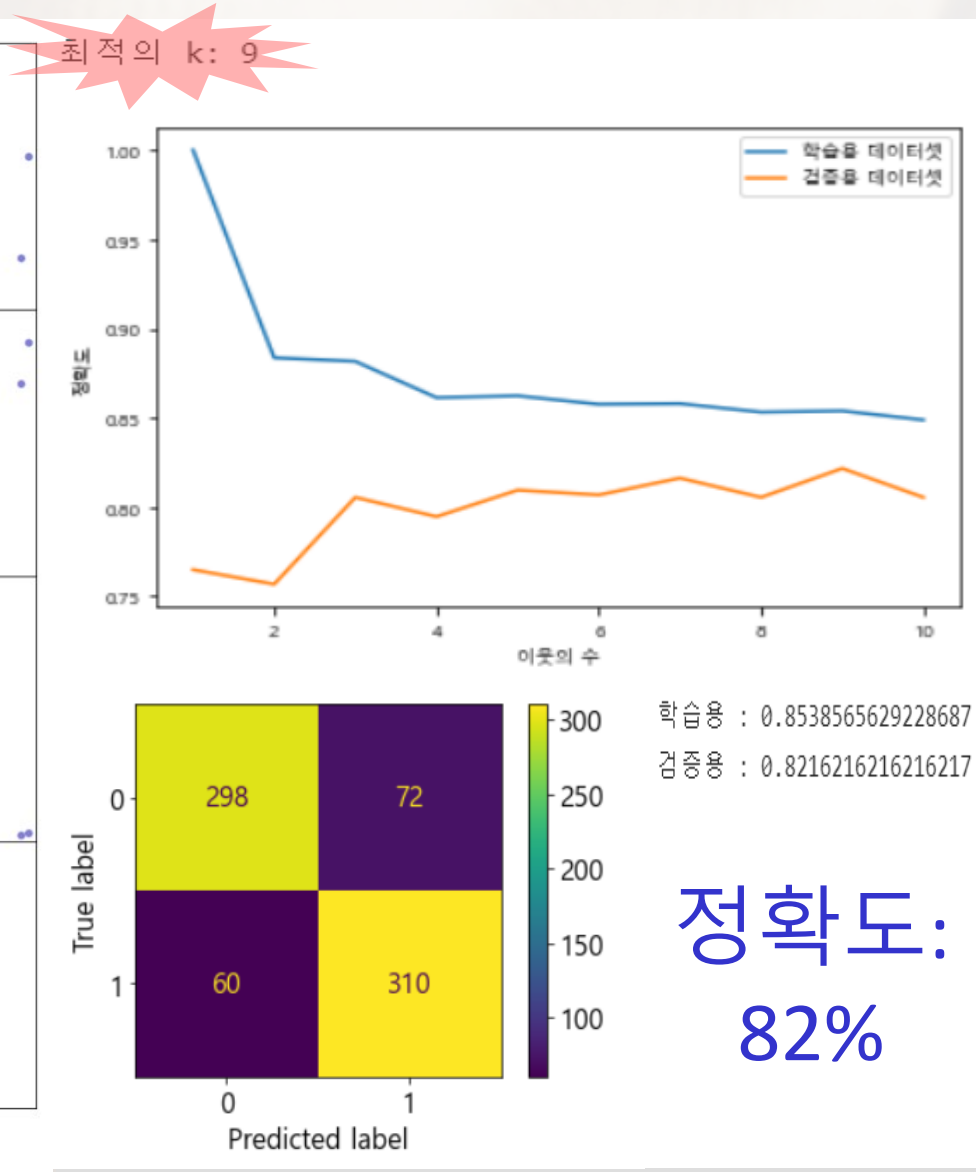
02

엑스트라트리

공통적으로 트리 만드는 결정에 votes의 중요도가 매우 높다

KNN





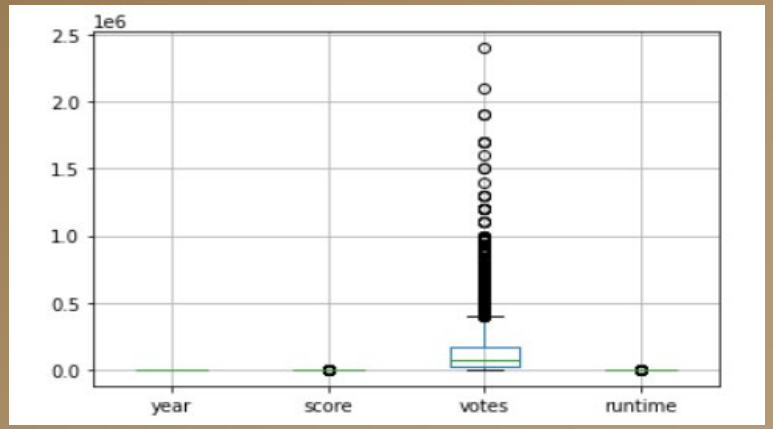
- 학습용 데이터셋의 경우 이웃이 1개일 때 100% 예측
- 이웃의 수가 늘어나면 학습용 데이터셋의 경우 모델이 단순해지고 정확도는 줄어든다.
- 실제값이 0, 예측값 0일 때 298개 일치
- 실제값1, 예측값1일 때 310개 일치

인공 신경망

스케일링 전

학습용 : 0.7547361299052774
검증용 : 0.7391891891891892

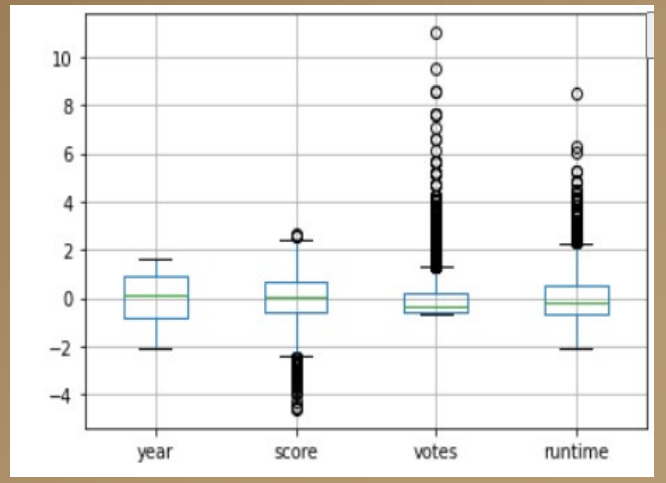
정확도: 73%



스케일링 후

학습용 : 0.8626522327469553
검증용 : 0.8540540540540541

정확도: 85%

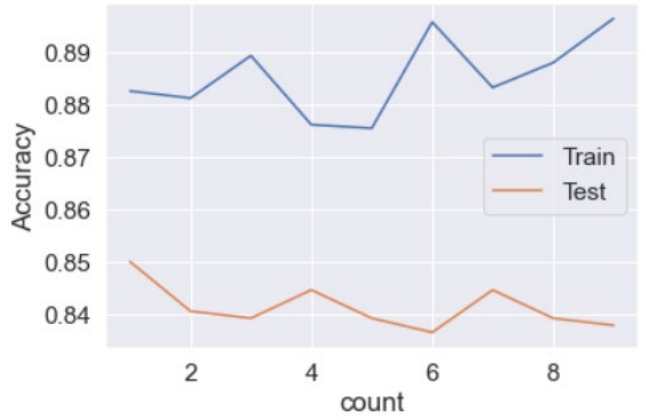


인공신경망

학습용 : 88.29%

검증용 : 83.38%

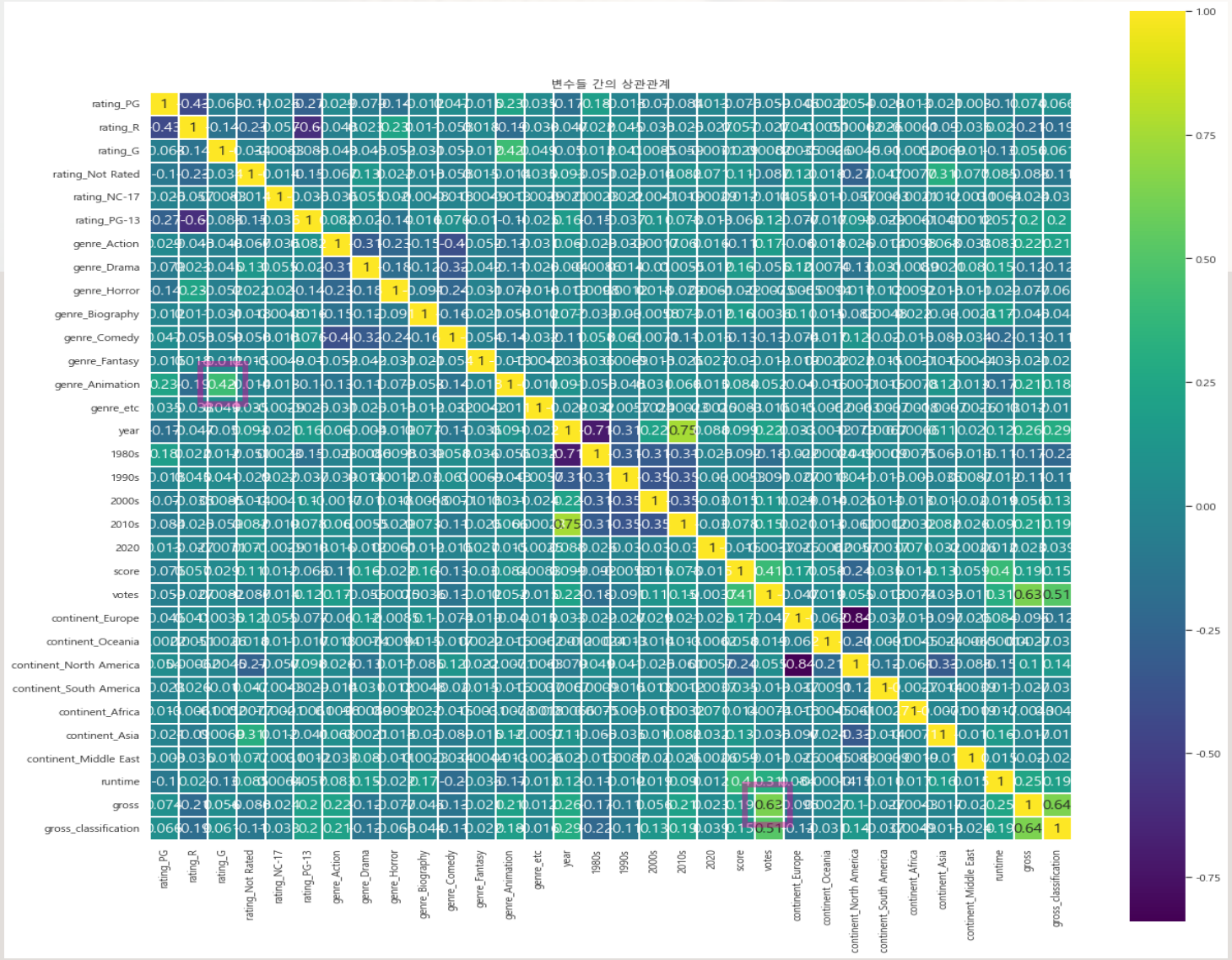
```
idx: 0
최고정확도: 0.85
100 100
```



최적의 은닉노드수로
만든 모형

Hidden layer 2개
Hidden layer1 – 100 nodes
Hidden layer2 – 100nodes

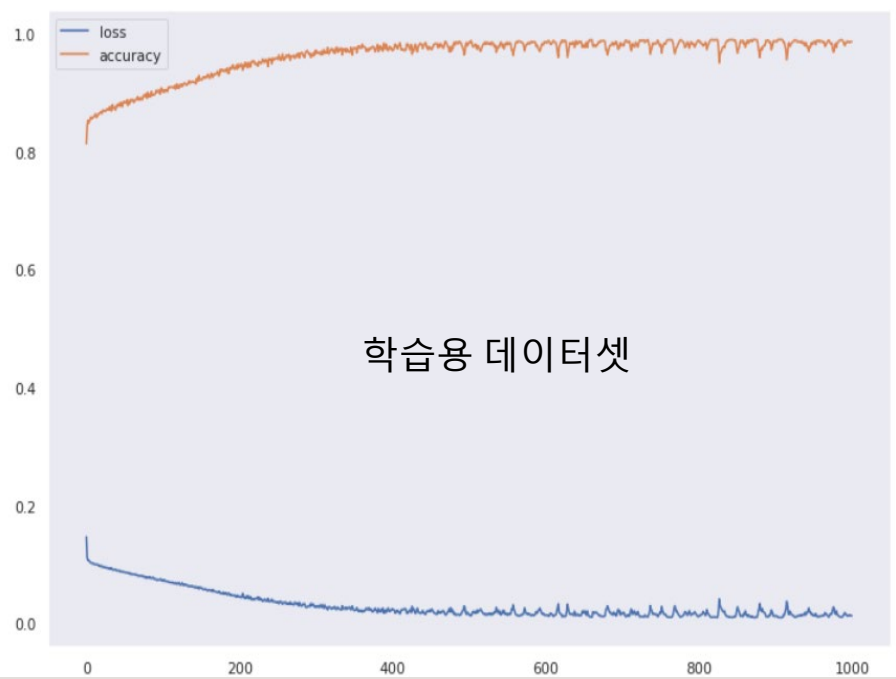
Keras- 인공신경망



상관계수 행렬 그래프

- ✓ Genre_Animation과 rating_G(전체관람가) 상관관계가 높다.
- ✓ votes와 gross 의 상관관계가 높다.

심층신경망



학습용 데이터셋

```
93/93 [==============================] - 1s 4ms/step - loss: 0.0118 - accuracy: 0.9882
[0.011780344881117344, 0.9881596565246582]
24/24 [==============================] - 0s 3ms/step - loss: 0.1763 - accuracy: 0.8095
[0.17633011937141418, 0.8094594478607178]
```

학습용
검증용

정확도: 80%

```
import numpy as np

test_set = np.array([7.5, 2004, 373000, 120]).reshape(1,4)
test_set=scaler.transform(test_set)
print(model.predict(test_set))
test_set = np.array([5, 2020, 90, 100]).reshape(1,4)
test_set=scaler.transform(test_set)
print(model.predict(test_set))
```

✓ 0.3s

np.array([score, year, votes, runtime])

실제 데이터와
동일한 값

```
1/1 [==============================] - 0s 113ms/step
[[1.]]

c:\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but RobustScaler was fitted with feature names
  warnings.warn(
c:\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but RobustScaler was fitted with feature names
  warnings.warn(


1/1 [==============================] - 0s 19ms/step
[[1.7129097e-21]]
```

심층신경망

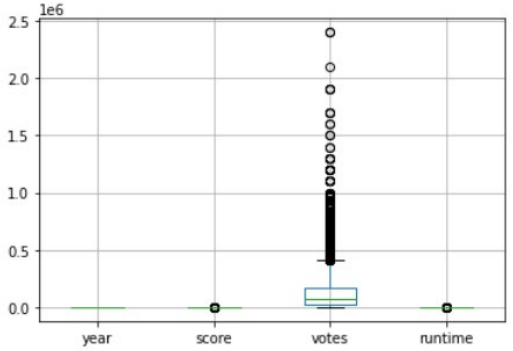
Hidden layer 1 :
128 nodes

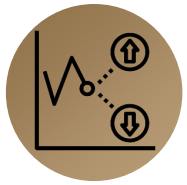
Hidden layer 2 :
64 nodes

Hidden layer 3 :
64 nodes

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 128)               640

 dense_1 (Dense)             (None, 64)                8256

 dense_2 (Dense)             (None, 64)                4160

 dense_3 (Dense)             (None, 1)                 65

=================================================================
Total params: 13,121
Trainable params: 13,121
Non-trainable params: 0
```

SVM

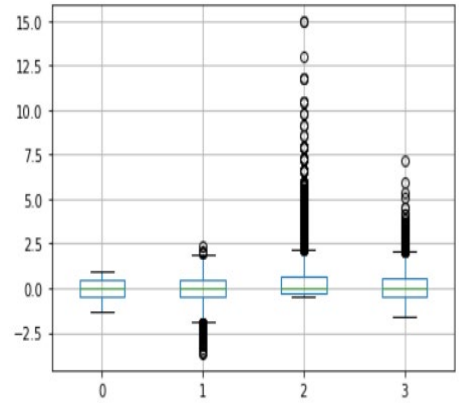


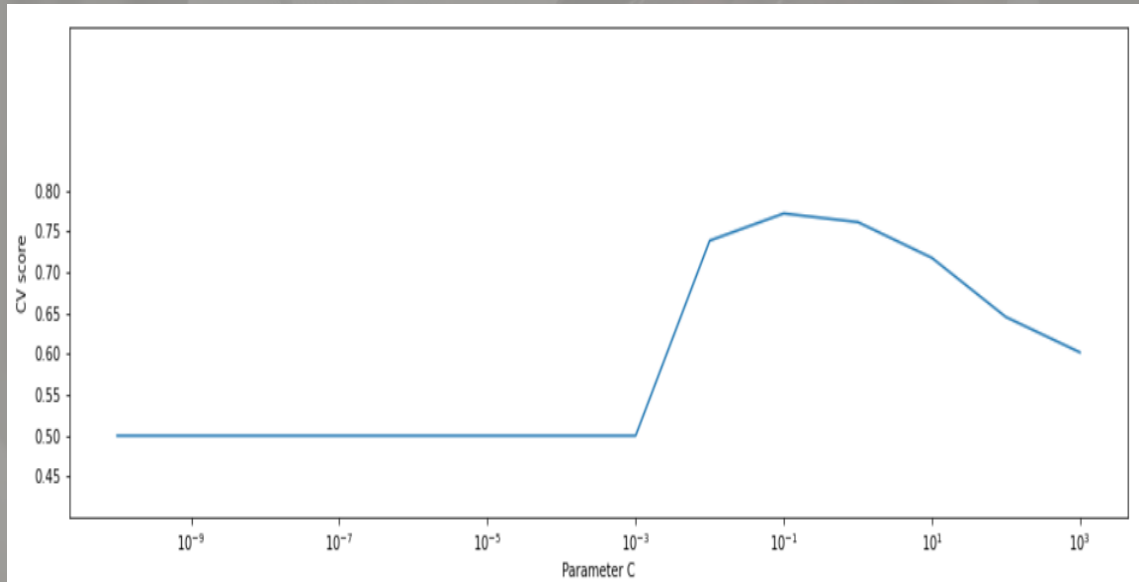


```
Best Parameters:
 {'C': 1, 'gamma': 1e-05}
Best Estimators:
 SVC(C=1, gamma=1e-05)
 학습용: 0.8487821380243572
 검증용: 0.8135135135135135
```

정확도: 81%

스케일링





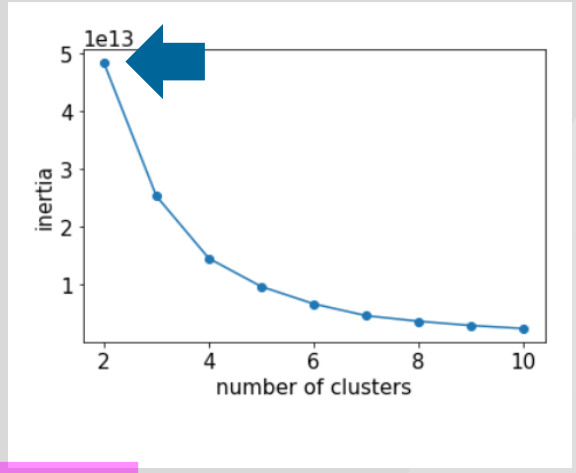
변수의 중요도

	0	1
0	year	0.0
1	score	0.0
2	votes	0.326569264069264
3	runtime	0.0

Clustering - 군집화

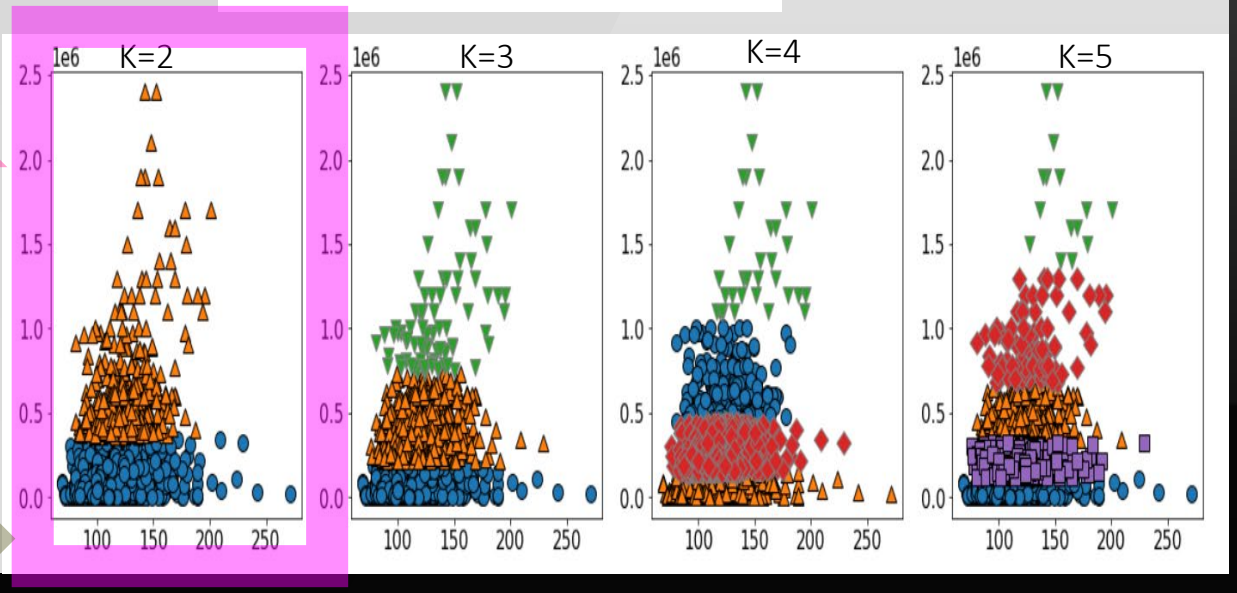
실루엣 계수 0.7748875581818264

- 군집화에서는 개별 라벨값보다는 군집이 중요하다.
- 실루엣 값은 한 클러스터 안의 데이터들이 다른 클러스터와 비교해서 얼마나 비슷한지 나타낸다.
- 실루엣계수가 클수록 좋은 모형
- 군집의 형상이 복잡하거나 크기의 차이가 많이 나면 비교가 어려운 단점이 있다.

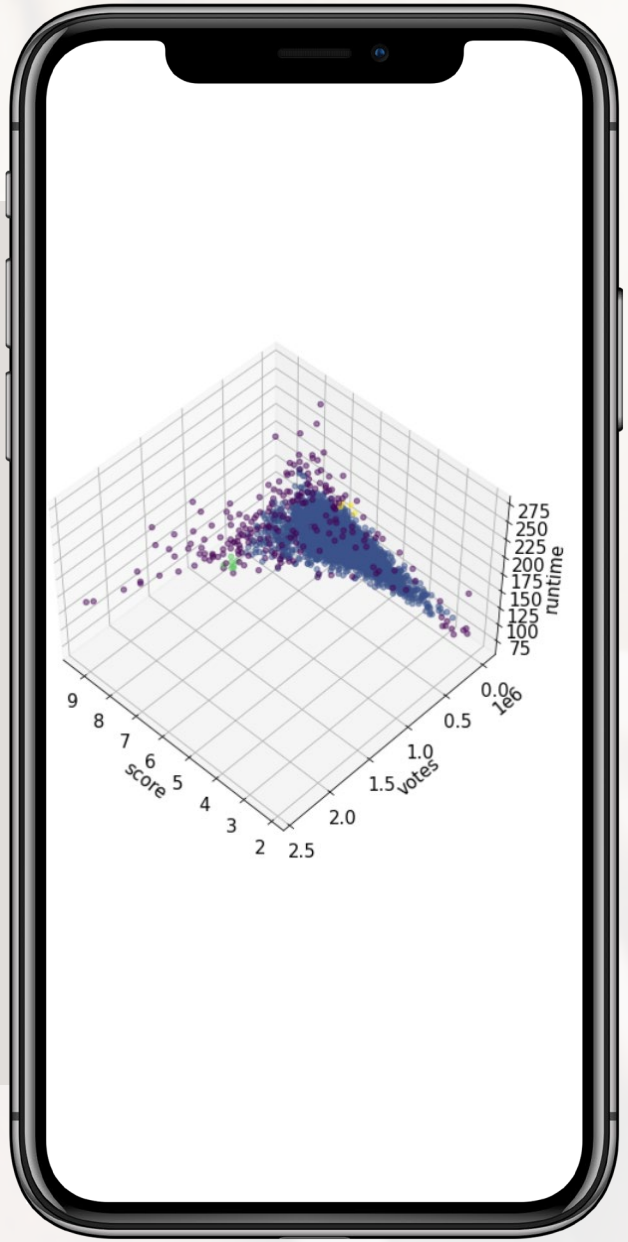


votes

runtime



Clustering – 밀도 기반



score

6~9

votes

0~0.5

runtime

0~125

일 때 밀도가 높음

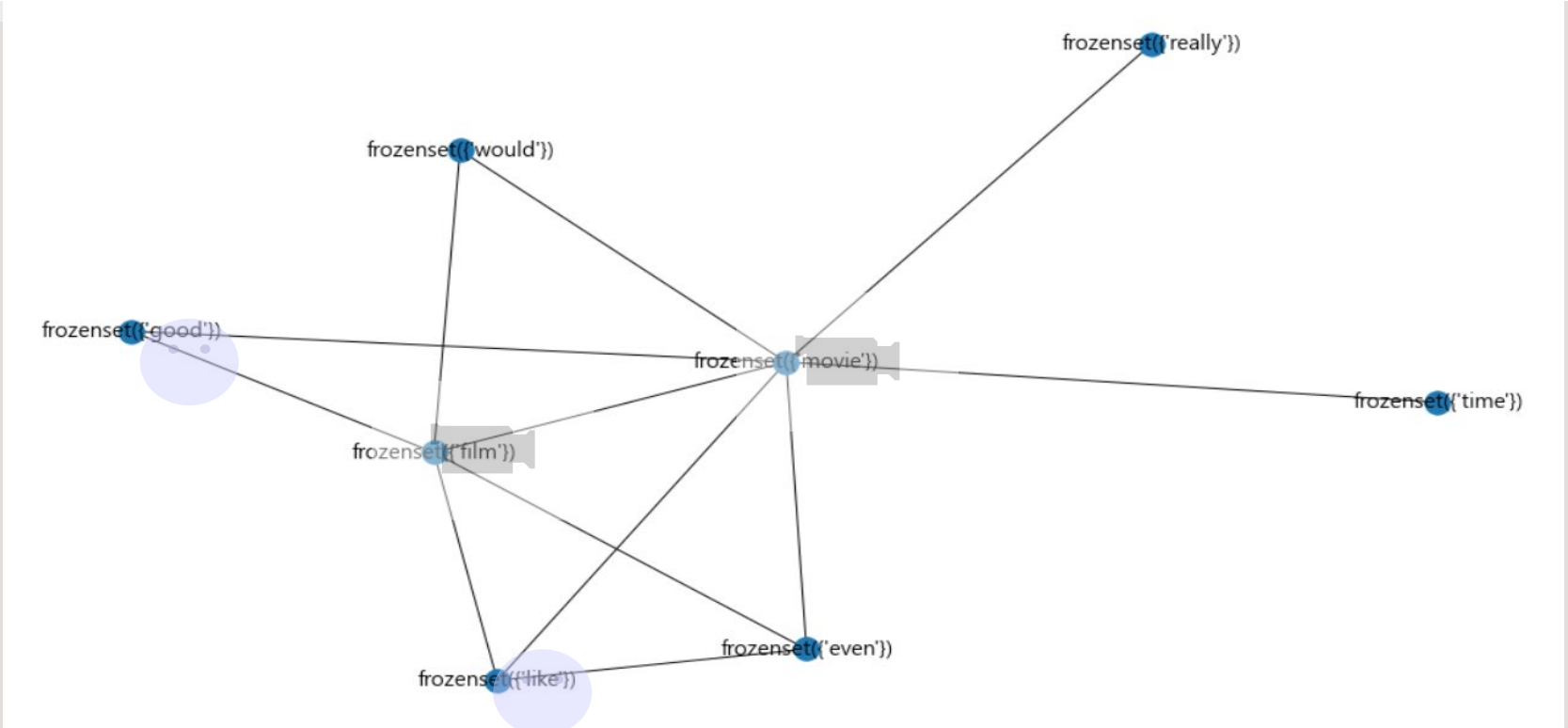
연관분석

연관도 높은 순서로 나열

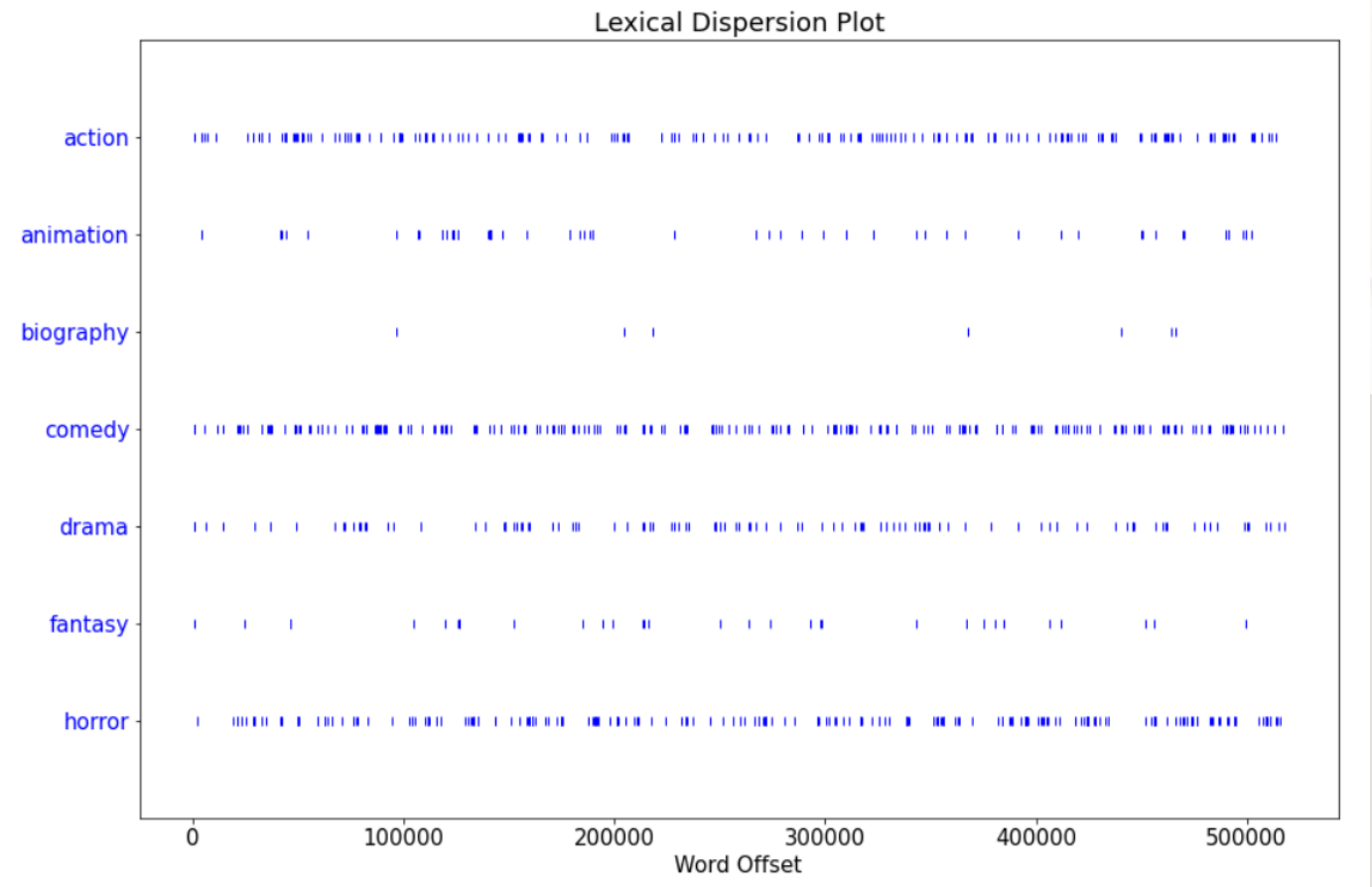
	support	itemsets
13	0.621	(movie)
5	0.550	(film)
9	0.471	(like)
7	0.353	(good)
4	0.349	(even)

연관분석

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(even)	(film)	0.349	0.550	0.208	0.595989	1.083616	0.016050	1.113830
1	(film)	(even)	0.550	0.349	0.208	0.378182	1.083616	0.016050	1.046930
2	(like)	(even)	0.471	0.349	0.204	0.433121	1.241034	0.039621	1.148393
3	(even)	(like)	0.349	0.471	0.204	0.584527	1.241034	0.039621	1.273248
4	(movie)	(even)	0.621	0.349	0.228	0.367150	1.052005	0.011271	1.028679



텍스트마이닝



Action, comedy, horror 단어 사용 빈도가 높고 biography 단어 사용 빈도가 낮음

결론

영화분석데이터

변수 중 votes가 대부분의
분석에서 가장 영향력이 크다.

영화리뷰데이터

① 영화 장르 중 comedy, action, horror가 선호도가 높다.
② 영화 관련 단어나 긍정적인 단어의 빈도수가 높다.

한계점&향후계획

1

투표수가 적으면 평점에 대한 표본수가 적어서 그 점수를 신뢰하기 어렵다는 한계가 있음

2

시계열분석 추가,
단어빈도분석 추가,
원핫인코딩 활용 추가

3

연관분석에서 특정 단어 불용어처리하기

THANK YOU

권영혜