

# WIKI-MACH

# 투표 할까?

## 투표 안할까?





## INDEX

1. 분석 목적
2. 변수 소개
3. 변수 연구
4. 모델 소개
5. 결론
6. 한계점



## 1. 분석 목적

## 1) 대회 설 명

### 심리 성향 예측 AI 경진대회

월간 데이콘 8 | 심리 테스트 분석 | AUC | 분류

💰 상금 : 100만원+애플워치

🕒 2020.09.28 ~ 2020.11.16 17:59

[+ Google Calendar](#)

👥 1,043팀 📅 D-10



참여중

1. 주제 : 심리학 테스트 분석 알고리즘 개발

2. 대회설명

- 마키아벨리즘 심리테스트를 활용하여 테스트 참가자의 국가 선거 투표 여부 예측
- 주어진 데이터만을 활용 (외부 데이터셋 사용 불가)

3. 주최/주관 : DACON



## 2. 변수 소개

1) 데이터 설명

◆ 본 공모전에서 제공받은 데이터 셋은 총 76개의 변수로 구성되어 있습니다.

변수명	설명	범주
QaA ~ QtA	마키아벨리즘 테스트 문항	1~5 (리커트 척도)
QaE ~ QtE	테스트 문항별 상대적 소요 시간	숫자 (연속형)
tp01 ~ tp10	TIPI 테스트 문항	1~5 (리커트 척도)
wr_01 ~ 13	단어 테스트 문항 (존재 o)	0 또는 1 (명목척도)
wf_01 ~ 03	단어 테스트 문항 (존재 x)	0 또는 1 (명목척도)
age_group	연령	10s ~ 40s (명목척도)
education	교육 수준	0~4 (순서형)
engnat	모국어가 영어	0, 1, 2 (명목척도)

변수명	설명	범주
familysize	형제자매 수	숫자 (연속형)
gender	성별	1, 2 (명목척도)
hand	필기하는 손	0~3 (명목척도)
married	혼인 상태	0~3 (명목척도)
race	인종	문자형 (명목척도)
religion	종교	문자형 (명목척도)
urban	유년기의 거주 구역	0~3 (명목척도)
voted	작년 국가 투표 여부	1 또는 2 (명목척도)



3. 변수 연구

1) | 데이터 설

명  
마키아벨리즘이? 사회심리학 등에서는  
란

“ 개인적인 욕구의 충족을 위해 남을 속이거나 조종하려는 욕구를 가리키는 용어 ” 로 사용됨

마키아벨리즘 테스트문항		전 혀x	약 간x	중 립	약 간o	매 우o
secret +	1. 당신이 어떤 일을 한 이유를 필요한 때를 제외하고는 누구에게도 절대 말하지 않는다.					
	2. 사람을 다루는 가장 좋은 방법은 그들이 듣기 원하는 말을 해주는 것이다					
	3. 도덕적으로 올바른 일 일때만, 그 행동을 해야 한다.					
secret +	4. 대부분의 사람들은 기본적으로 선하며 착하다.					
	5. 모든 사람들에게는 악한 경향이 있고 기회만 주어진다면 그 성향이 표현된다고 가정하는 것이 가장 안					
secret -	6. 모든 경우에 있어 정직이 최선이다.					
secret -	7. 누군가에게 거짓말하는 것은 변명의 여지가 없다.					
secret -	8. 일반적으로 사람들을 강제하지 않는 한, 열심히 일하지 않는다.					
	9. 대체로 검손하고 정직한 것이 중요하고 부정직한 것보다 낫다.					
secret +	10. 어떤 일을 해왔다고 누군가에게 자랑할때, 내가 그 일을 했다는 실제 이유를 말해주는 것이 더 중요해					
secret +	보이는 듯한 이유를					
	알려주는 것보다 좋은 방법이다.					
secret +	11. 세상을 앞에서 이끌어가는 사람들은 깨끗하고 도덕적인 삶을 산다.					
	12. 누군가를 완전히 믿는 사람은 고생을 지치하는 것이다.					
secret +	13. 범죄자들과 다른 일반 사람들의 가장 큰 차이점은, 범죄자들이 잡힐만큼 멍청하다는 것이다.					
secret -	14. 대부분의 사람들은 용감하다.					
secret +	15. 중요한 사람들에게 아첨하는 것이 현명하다.					
	16. 모든면에서 좋은 것이 가능하다.					
secret +	17. F.T Barnum이 "매 분마다 신천적으로 잘 속는 사람이 태어난다"라고 말한 것은 틀리다.					
secret +	18. 요령이 없다면 앞으로 나아가기 힘들다					
	19. 불치병에 걸린 사람들에게 안락사를 선택할 권리를 갖게 해야 한다.					
secret +	20. 대부분의 사람들은 그들의 재산은 잃은 것보다 부모의 죽음을 더 빨리 잊는다					

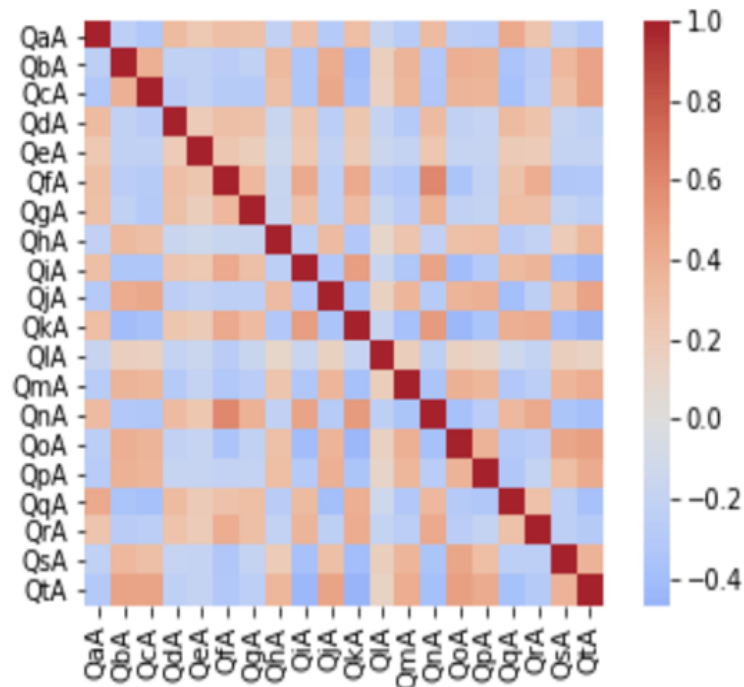


## 2) | 마키아벨리즘 테스트 점수

```
In [22]: import seaborn as sns
          from IPython.display import Image

          correlations = df1[Q_Ques].corr(method = 'spearman')
          sns.heatmap(correlations, cmap="coolwarm", square=True, center=0)
```

Out[22]: <AxesSubplot:>



부호가 같은 문항끼리는 (QbA와 QcA)  
상관관계 값이 **양수**이고,

부호가 다른 문항끼리는 (QbA와 QeA)  
상관관계 값이 **음수**이다.

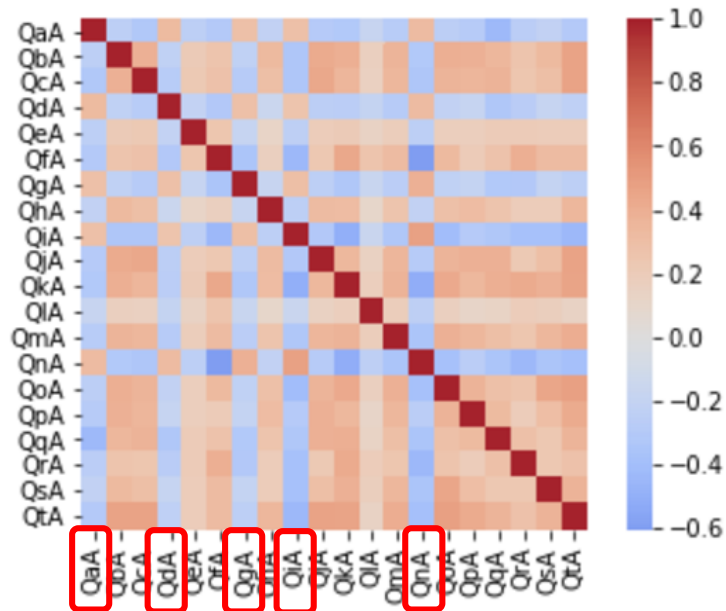
## 2) | 마키아벨리즘 테스트 점수

```
In [25]: import seaborn as sns
from IPython.display import Image

flipping_columns = ["QeA", "QfA", "QkA", "QqA", "QrA"]
for flip in flipping_columns:
    df1[flip] = 6 - df1[flip]

correlations = df1[Q_Ques].corr(method = 'spearman')
sns.heatmap(correlations, cmap="coolwarm", square=True, center=0)
```

Out[25]: <AxesSubplot:>



이미 알고 있는 (-) 부호 문항을  
(+)로 reverse

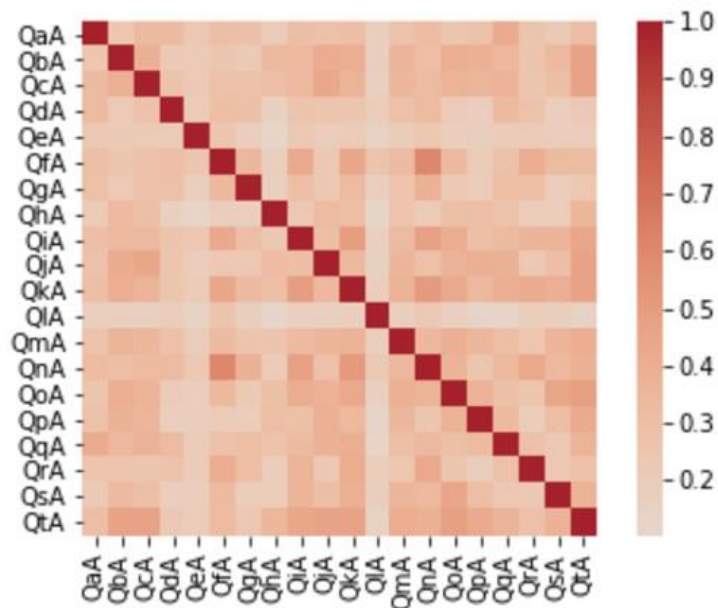
QaA, QdA, QgA, QiA, QnA는  
(-)부호

## 2) | 마키아벨리즘 테스트 점수

```
In [26]: flipping_secret_columns = ["QaA", "QdA", "QgA", "QiA", "QnA"]
         for flip in flipping_secret_columns:
             df1[flip] = 6 - df1[flip]

         correlations = df1[Q_Ques].corr(method = 'spearman')
         sns.heatmap(correlations, cmap="coolwarm", square=True, center=0)
```

Out[26]: <AxesSubplot:>



(-) 부호인 secret 문항을 (+)로 reverse  
=> 모두 (+)로 변경

## 2) | 마키아벨리즘 테스트 점수

```
In [27]: # 컬럼에 추가
df1['Mach_score'] = df1[Q_Ques].sum(axis = 1)
df2['Mach_score'] = df2[Q_Ques].sum(axis = 1)
```

```
In [28]: df1['Mach_score']
```

```
Out[28]: index
0      59.0
1      52.0
2      38.0
3      67.0
4      60.0
...
45527   83.0
45528   76.0
45529   30.0
45530   58.0
45531   68.0
Name: Mach_score, Length: 45532, dtype: float64
```



0~100 사이의 마키아벨리즘 스코어

Mach\_Score가 높을수록 계산적, 신중  
Mach\_Score가 낮을수록 정직, 공감능력  
높음

## 1) | 데이터 설명 : 변수그룹2. TIPI


# 10문항 =&gt; 5요인으로 구성

1. \_\_\_\_ 나는 활발하고 열심히 하는 사람이다.
2. \_\_\_\_ 나는 따지기를 좋아하고 다투기를 좋아하는 사람이다.
3. \_\_\_\_ 나는 믿음직스럽고 자기관리가 가능한 사람이다.
4. \_\_\_\_ 나는 불안하고 화를 잘 내는 사람이다.
5. \_\_\_\_ 나는 새로운 경험을 마다하지 않으며 여러 가지로 생각해보는 사람이다.
6. \_\_\_\_ 나는 내향적이고 조용한 사람이다.
7. \_\_\_\_ 나는 동정심이 많고 다정한 사람이다.
8. \_\_\_\_ 나는 계획적이지 않고 조심성 없는 사람이다.
9. \_\_\_\_ 나는 침착하고 기분이 안정된 사람이다.
10. \_\_\_\_ 나는 변화를 싫어하며 창의적이지 않은 사람이다.

## 1) | 데이터 설명 : 변수그룹2. TIPI

# 10문항 =&gt; 5요인으로 구성

1. \_\_\_\_ 나는 활발하고 열심히 하는 사람이다.
2. \_\_\_\_ 나는 따지기를 좋아하고 다투기를 좋아하는 사람이다.
3. \_\_\_\_ 나는 믿음직스럽고 자기관리가 가능한 사람이다.
4. \_\_\_\_ 나는 불안하고 화를 잘 내는 사람이다.
5. \_\_\_\_ 나는 새로운 경험을 마다하지 않으며 여러 가지로 생각해보는 사람이다.
6. \_\_\_\_ 나는 내향적이고 조용한 사람이다.
7. \_\_\_\_ 나는 동정심이 많고 다정한 사람이다.
8. \_\_\_\_ 나는 계획적이지 않고 조심성 없는 사람이다.
9. \_\_\_\_ 나는 침착하고 기분이 안정된 사람이다.
10. \_\_\_\_ 나는 변화를 싫어하며 창의적이지 않은 사람이다.


$$\text{외향성} = (1\text{번} + 6\text{번})/2$$

## 1) | 데이터 설명 : 변수그룹2. TIPI

# 10문항 =&gt; 5요인으로 구성

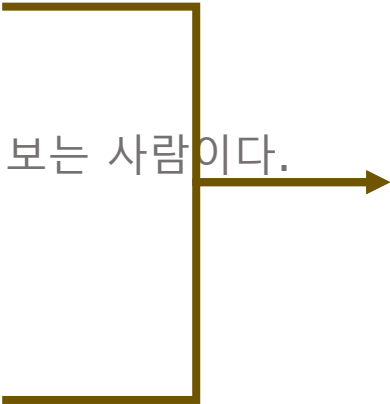
1. \_\_\_\_ 나는 활발하고 열심히 하는 사람이다.
2. \_\_\_\_ 나는 따지기를 좋아하고 다투기를 좋아하는 사람이다.
3. \_\_\_\_ 나는 믿음직스럽고 자기관리가 가능한 사람이다.
4. \_\_\_\_ 나는 불안하고 화를 잘 내는 사람이다.
5. \_\_\_\_ 나는 새로운 경험을 마다하지 않으며 여러 가지로 생각해보는 사람이다.
6. \_\_\_\_ 나는 내향적이고 조용한 사람이다.
7. \_\_\_\_ 나는 동정심이 많고 다정한 사람이다.
8. \_\_\_\_ 나는 계획적이지 않고 조심성 없는 사람이다.
9. \_\_\_\_ 나는 침착하고 기분이 안정된 사람이다.
10. \_\_\_\_ 나는 변화를 싫어하며 창의적이지 않은 사람이다.


$$\text{친화성} = (7\text{번} + 2\text{번})/2$$

## 1) | 데이터 설명 : 변수그룹2. TIPI

# 10문항 =&gt; 5요인으로 구성

1. \_\_\_\_ 나는 활발하고 열심히 하는 사람이다.
2. \_\_\_\_ 나는 따지기를 좋아하고 다투기를 좋아하는 사람이다.
3. \_\_\_\_ 나는 믿음직스럽고 자기관리가 가능한 사람이다.
4. \_\_\_\_ 나는 불안하고 화를 잘 내는 사람이다.
5. \_\_\_\_ 나는 새로운 경험을 마다하지 않으며 여러 가지로 생각해보는 사람이다.
6. \_\_\_\_ 나는 내향적이고 조용한 사람이다.
7. \_\_\_\_ 나는 동정심이 많고 다정한 사람이다.
8. \_\_\_\_ 나는 계획적이지 않고 조심성 없는 사람이다.
9. \_\_\_\_ 나는 침착하고 기분이 안정된 사람이다.
10. \_\_\_\_ 나는 변화를 싫어하며 창의적이지 않은 사람이다.



$$\text{성실성} = (3\text{번} + 8\text{번})/2$$



## 1) | 데이터 설명 : 변수그룹2. TIPI

# 10문항 =&gt; 5요인으로 구성

1. \_\_\_\_ 나는 활발하고 열심히 하는 사람이다.
2. \_\_\_\_ 나는 따지기를 좋아하고 다투기를 좋아하는 사람이다.
3. \_\_\_\_ 나는 믿음직스럽고 자기관리가 가능한 사람이다.
4. \_\_\_\_ 나는 불안하고 화를 잘 내는 사람이다.
5. \_\_\_\_ 나는 새로운 경험을 마다하지 않으며 여러 가지로 생각해보는 사람이다.
6. \_\_\_\_ 나는 내향적이고 조용한 사람이다.
7. \_\_\_\_ 나는 동정심이 많고 다정한 사람이다.
8. \_\_\_\_ 나는 계획적이지 않고 조심성 없는 사람이다.
9. \_\_\_\_ 나는 침착하고 기분이 안정된 사람이다.
10. \_\_\_\_ 나는 변화를 싫어하며 창의적이지 않은 사람이다.

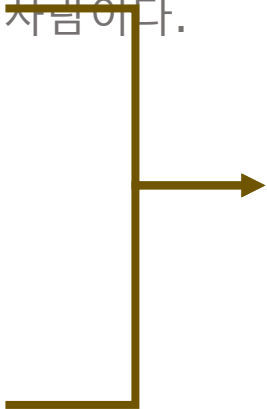


정서적 안정성 = (9번 + 4번)/2

## 1) | 데이터 설명 : 변수그룹2. TIPI

# 10문항 =&gt; 5요인으로 구성

1. \_\_\_\_ 나는 활발하고 열심히 하는 사람이다.
2. \_\_\_\_ 나는 따지기를 좋아하고 다투기를 좋아하는 사람이다.
3. \_\_\_\_ 나는 믿음직스럽고 자기관리가 가능한 사람이다.
4. \_\_\_\_ 나는 불안하고 화를 잘 내는 사람이다.
5. \_\_\_\_ 나는 새로운 경험을 마다하지 않으며 여러 가지로 생각해보는 사람이다.
6. \_\_\_\_ 나는 내향적이고 조용한 사람이다.
7. \_\_\_\_ 나는 동정심이 많고 다정한 사람이다.
8. \_\_\_\_ 나는 계획적이지 않고 조심성 없는 사람이다.
9. \_\_\_\_ 나는 침착하고 기분이 안정된 사람이다.
10. \_\_\_\_ 나는 변화를 싫어하며 창의적이지 않은 사람이다.



경험개방성 = (5번 + 10번)/2

## 2) | TIPI 점수

```
In [29]: fea2 = ['tp01', 'tp02', 'tp03', 'tp04', 'tp05', 'tp06', 'tp07', 'tp08', 'tp09', 'tp10']
df1.loc[:, fea2] = df1.loc[:, fea2].applymap(lambda x: 7 - x)
df2.loc[:, fea2] = df2.loc[:, fea2].applymap(lambda x: 7 - x)

fea3 = ['tp02', 'tp04', 'tp06', 'tp08', 'tp10']
df1.loc[:, fea3] = df1.loc[:, fea3].applymap(lambda x: 0 if x == 0 else 8 - x)
df2.loc[:, fea3] = df2.loc[:, fea3].applymap(lambda x: 0 if x == 0 else 8 - x)

df1['sung'] = (df1.tp03 + df1.tp08)/2
df1['chin'] = (df1.tp07 + df1.tp02)/2
df1['jung'] = (df1.tp09 + df1.tp04)/2
df1['kyung'] = (df1.tp05 + df1.tp10)/2
df1['why'] = (df1.tp01 + df1.tp06)/2

# 기존 tp변수 빼주기
df1 = df1.drop(fea2, axis=1)
df2 = df2.drop(fea2, axis=1)
```

```
In [30]: df1['sung']
```

```
Out[30]: index
0         5.0
1         6.0
2         6.5
3         5.0
4         6.5
...
45527      1.0
45528      5.0
45529      4.0
45530      5.0
45531      5.0
Name: sung, Length: 45532, dtype: float64
```

10개의 질문은 2개씩  
총 5개의 요인  
으로 구성

## 1) 데이터 설명 : 변수그룹3. 기타 변수

In the grid below, check all the words whose definitions you are sure you know.

<input type="checkbox"/> boat	<input type="checkbox"/> incoherent	<input type="checkbox"/> pallid	<input type="checkbox"/> robot
<input type="checkbox"/> audible	<input type="checkbox"/> equivocal	<input type="checkbox"/> paucity	<input type="checkbox"/> epistemology
<input type="checkbox"/> flouted	<input type="checkbox"/> decide	<input type="checkbox"/> pastiche	<input type="checkbox"/> verdid
<input type="checkbox"/> abysmal	<input type="checkbox"/> lucid	<input type="checkbox"/> betray	<input type="checkbox"/> funny

## 1) 데이터 설명 : 변수그룹3. 기타 변수

In the grid below, check all the words whose definitions you are sure you know.

<input type="checkbox"/> boat	<input type="checkbox"/> incoherent	<input type="checkbox"/> pallid	<input type="checkbox"/> robot
<input type="checkbox"/> audible	<input type="checkbox"/> cuivocal	<input type="checkbox"/> paucity	<input type="checkbox"/> epistemology
<input type="checkbox"/> florted	<input type="checkbox"/> decide	<input type="checkbox"/> pastiche	<input type="checkbox"/> verdid
<input type="checkbox"/> abysmal	<input type="checkbox"/> lucid	<input type="checkbox"/> betray	<input type="checkbox"/> funny

=> " 존재 하지 않는 "  
단어

“ 설문조사의 신뢰도를 판단할 수도 있는 척도라는 부분을 파악  
했지만  
특성중요도가 낮아 제거하기로 결정 ”

4) 가설  
설정

Socioeconomic factors are significantly associated with whether individuals develop the habit of voting. The most important socioeconomic factor affecting voter turnout is education. The more educated a person is, the more likely they are to vote, even controlling for other factors that are closely associated with education level, such as income and class.

-> 교육 수준이 투표율과 가장 큰 연관이 있다.

differences in turnout between such groups in many societies. Other demographic factors have an important influence: young people are far less likely to vote than the elderly. -> 젊은 사람들일수록 투표율이 떨어진다



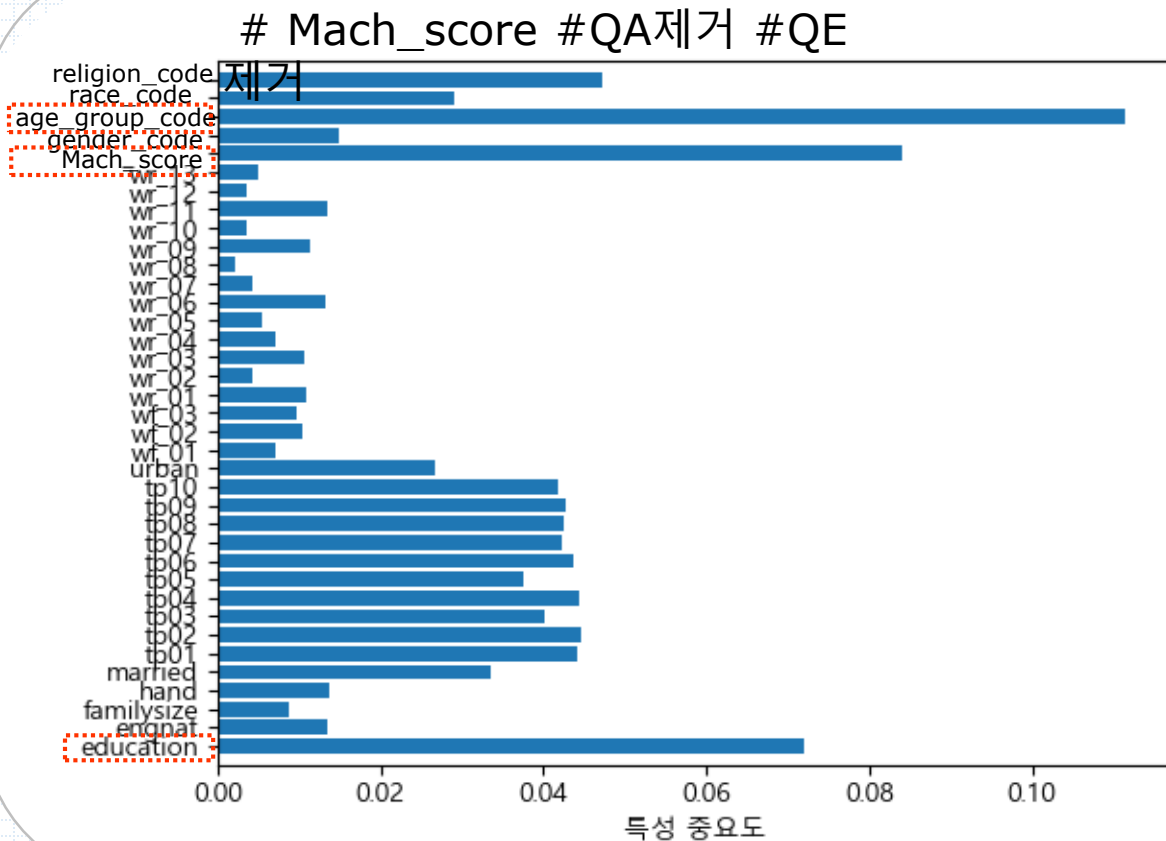
## 마키아벨리즘

마키아벨리즘 성향 높음 마키아벨리즘 성향 높음 (이하 높은 성향) 분류의 사람들은 다른 사람들과 소통하는데 있어 보다 계산적이고 신중하게 접근하는 경향이 있다. 5가지 성격 특성 요소에서, 이 사람들은 친화성 수치가 낮고 성실성 수치가 높게 나오는 경향이 있다. -> Mach스코어가 높을수록 성실성이 높고, 친화성이 낮을 것

가설 : 마키아벨리즘 테스트 결과와 투표여부의 상관관계가 있다 !

## 3) EDA

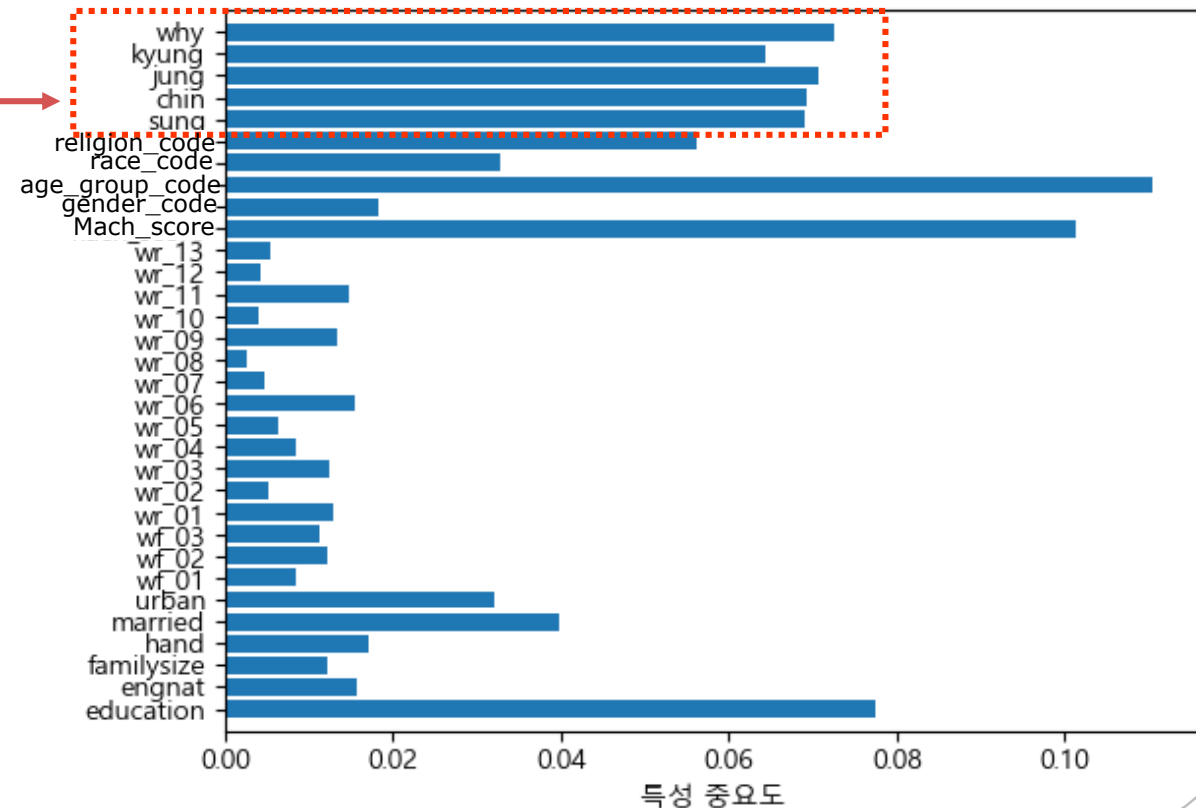
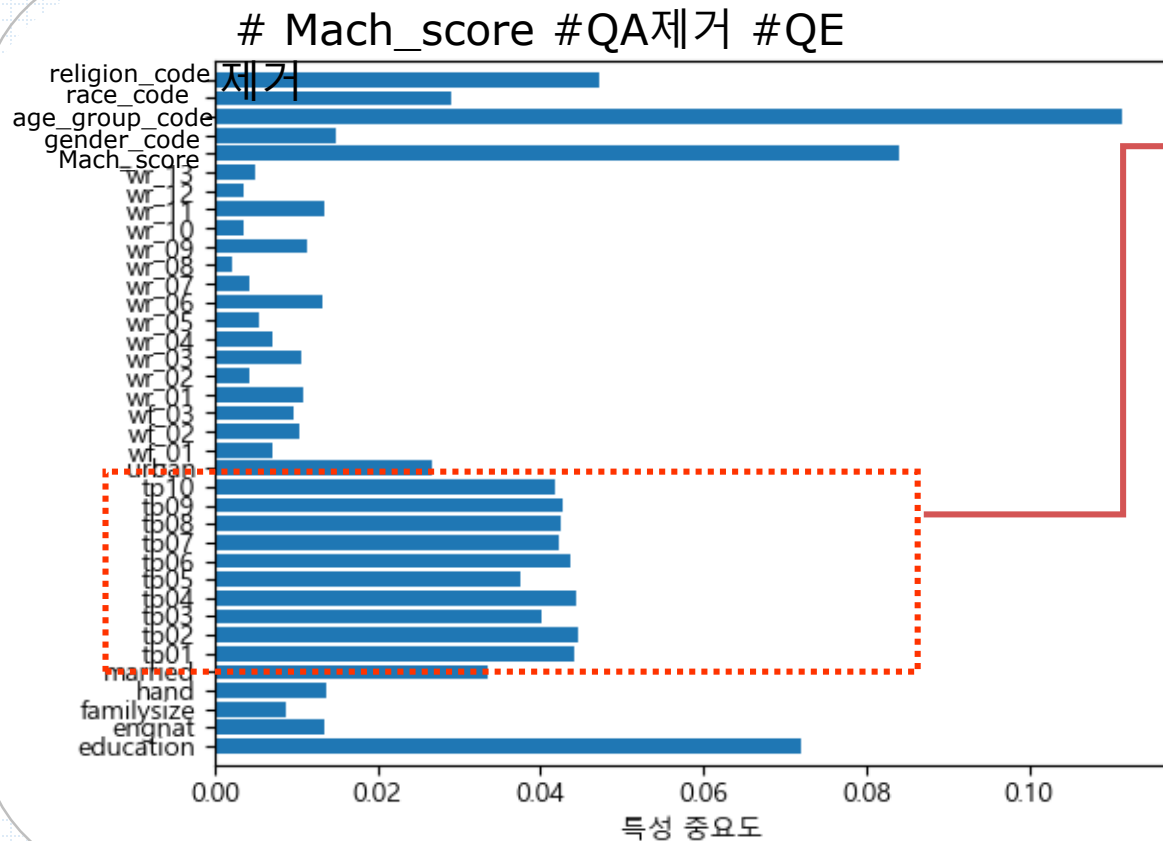
## (1) 특성중요도



▶ age\_group, Mach\_score, education 컬럼의 특성 중요도가 높다는 것을 알 수 있다

## 3) EDA

## (1) 특성중요도



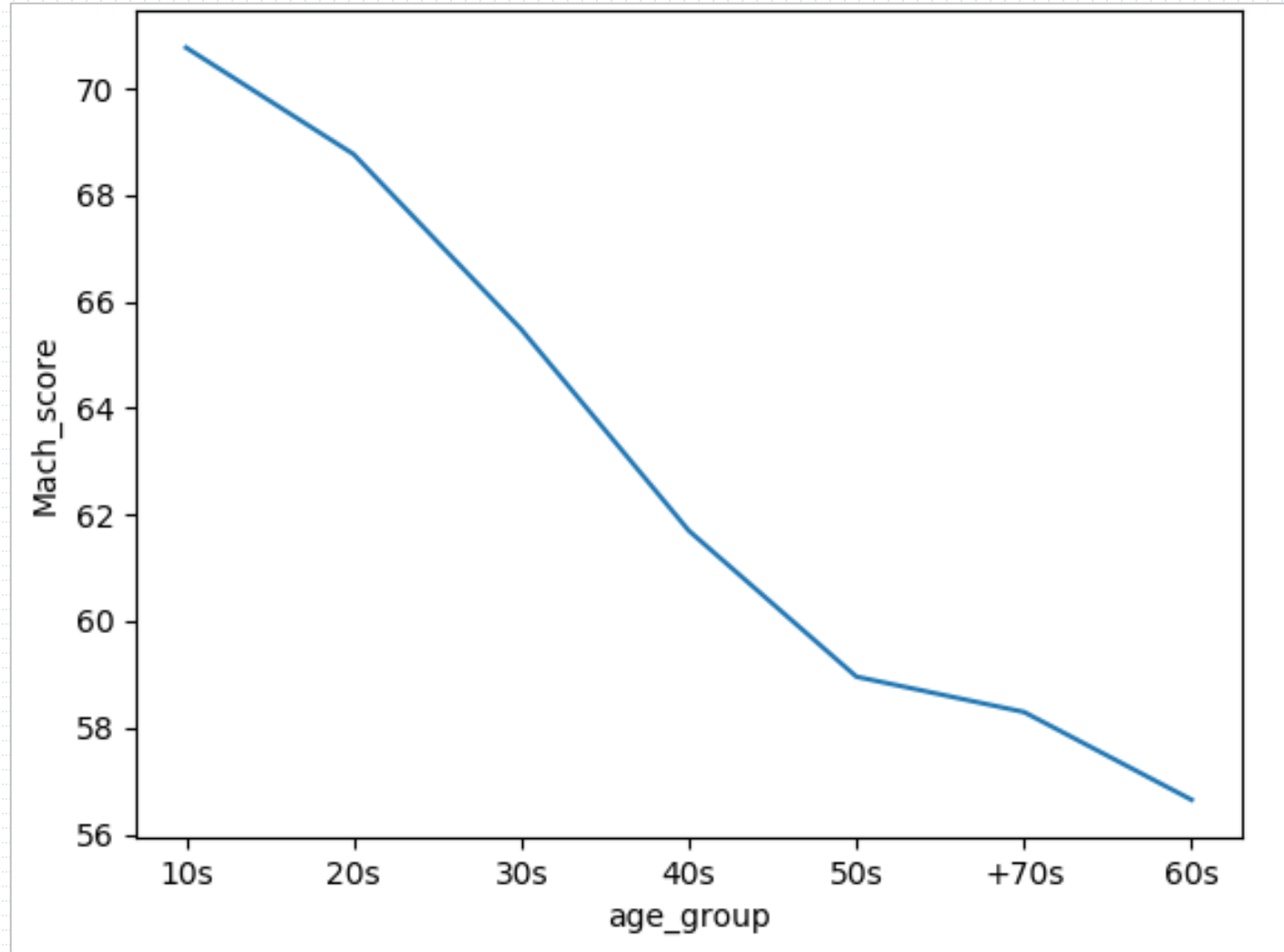
▶ 10개의 TIPI 컬럼을 5가지로 점수화한 결과, 특성중요도가 높아진 것을 알 수 있다.



## 3) EDA - age\_group

(2) 나이가 어릴수록 Mach\_score가 높다는 것을 알 수 있다.

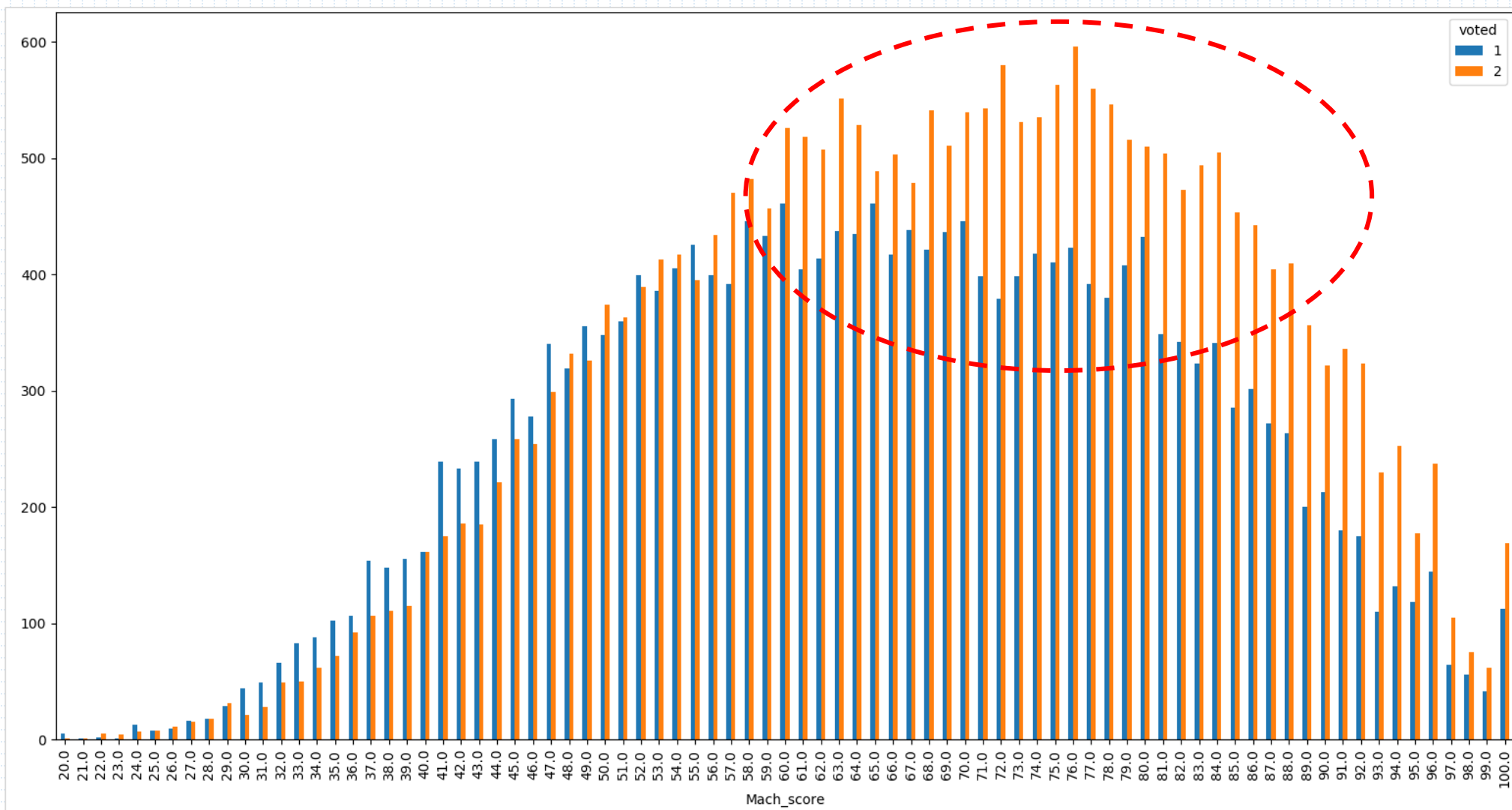
#age\_group+ Mach\_score 상관 관계



## 3) EDA - age\_group

- Mach\_score가 높아질수록, 투표율이 낮아지는 경향

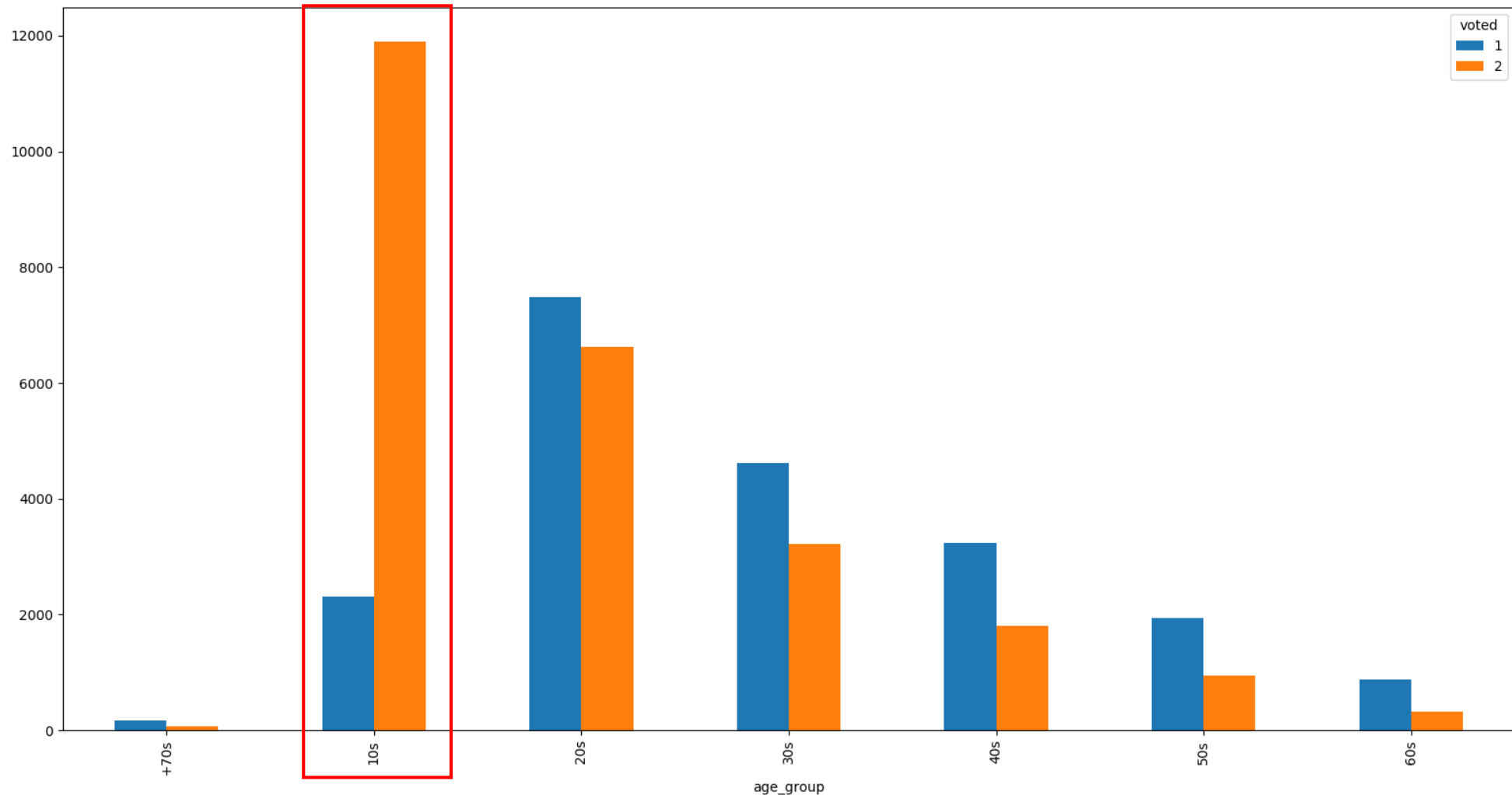
# Mach\_score와 voted 관계



## 3) EDA - age\_group

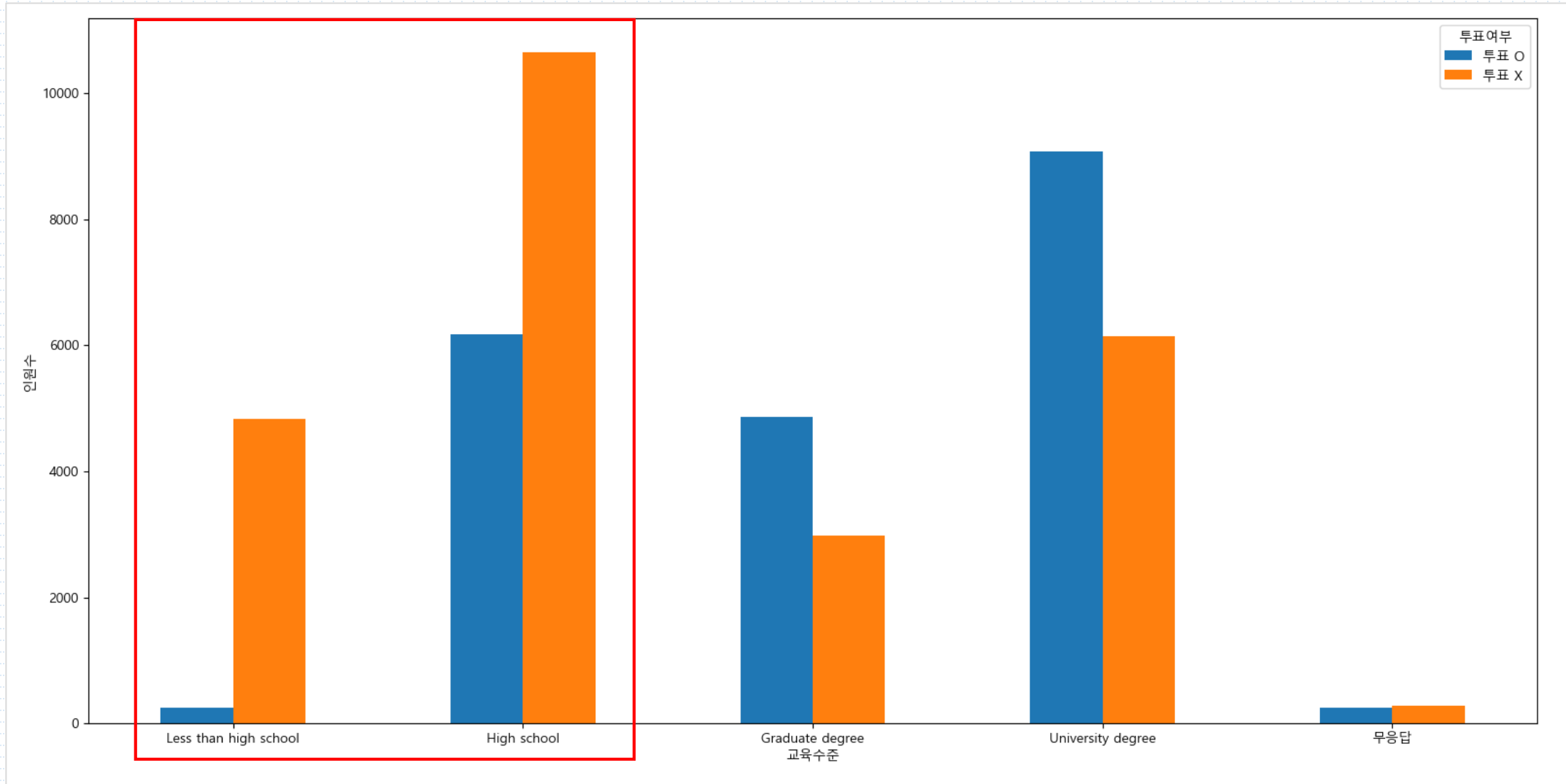
=> 앞선 그래프들을 통해, 나이가 어릴수록 투표율이 낮은 것을 알 수 있다.

# age\_group + voted 상관관계



## 3) EDA - education

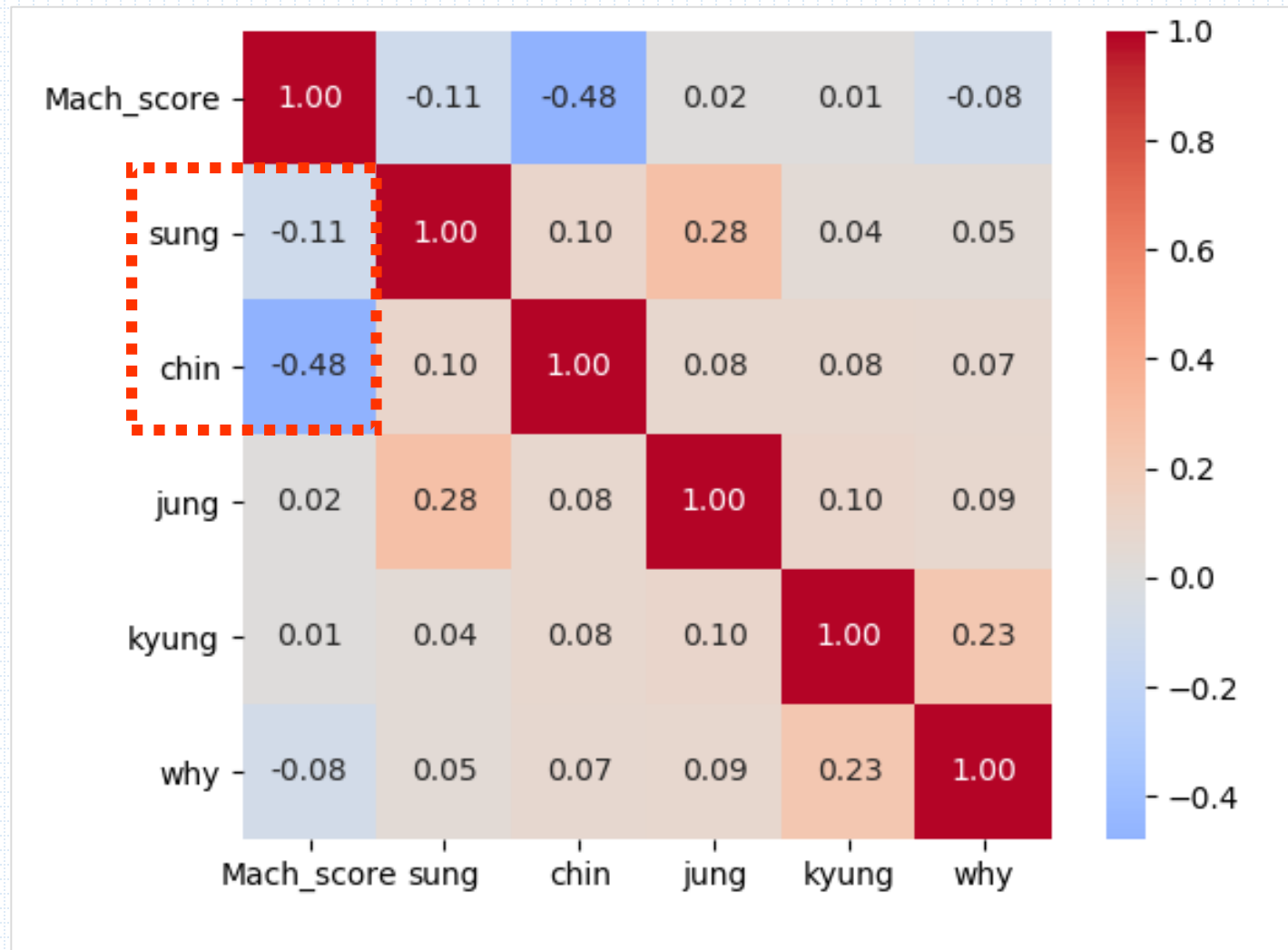
(3) 교육수준이 낮을수록, 투표율이 낮은 것을 알 수 있다.



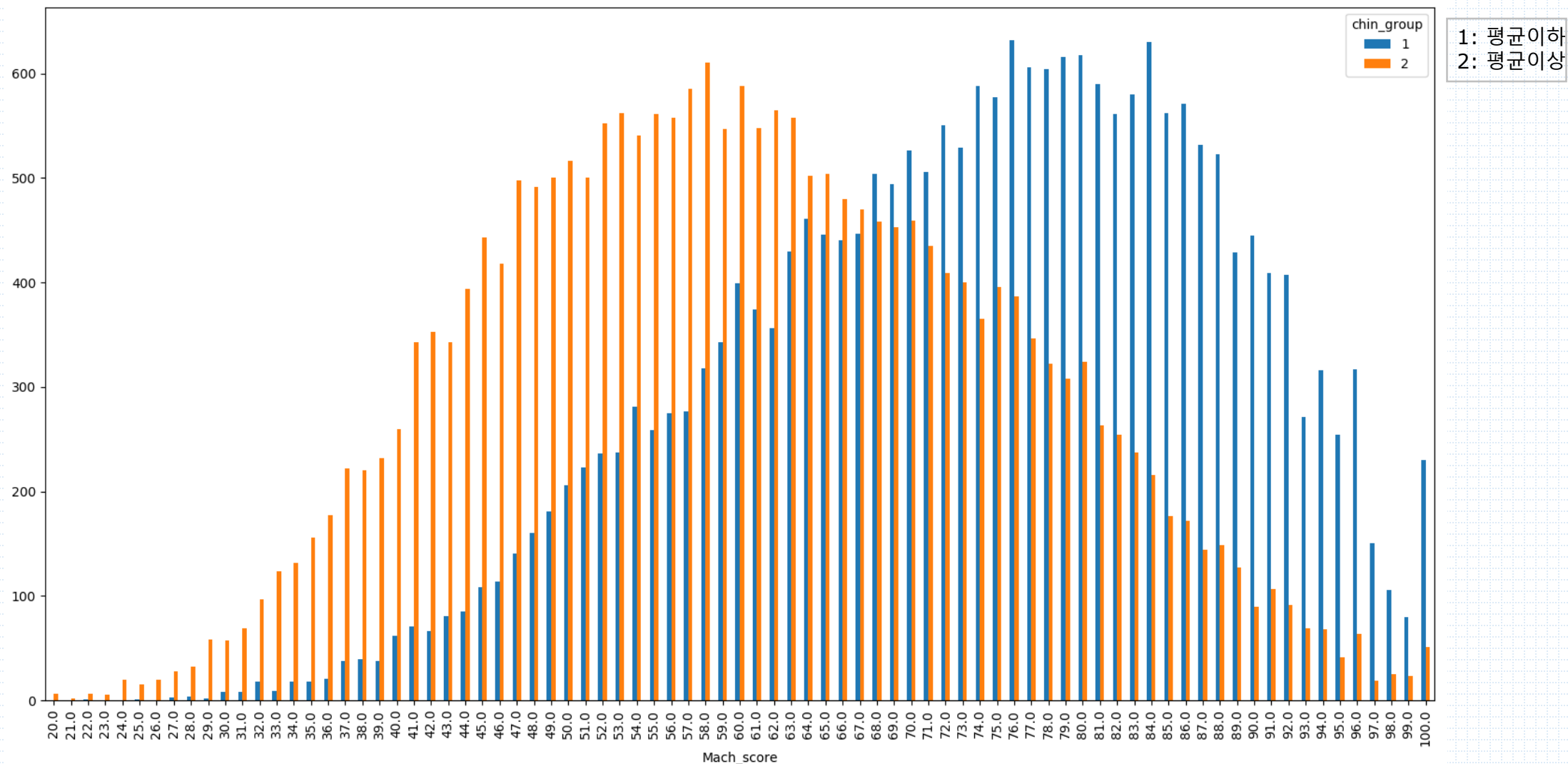
### 3) EDA - Mach\_Score

- ◆ Mach\_score와 친화성은 뚜렷한 음(-)의 상관관계가 있는 것으로 보이나, 성실성에서는 뚜렷한 상관관계를 찾지 못함

# Mach\_score + tipi 히트맵



- 3) EDA -  
Mach\_Score  
- Mach\_score 가 높을 수록 친화성이 낮다  
# Mach\_score + 친화성 상관관계

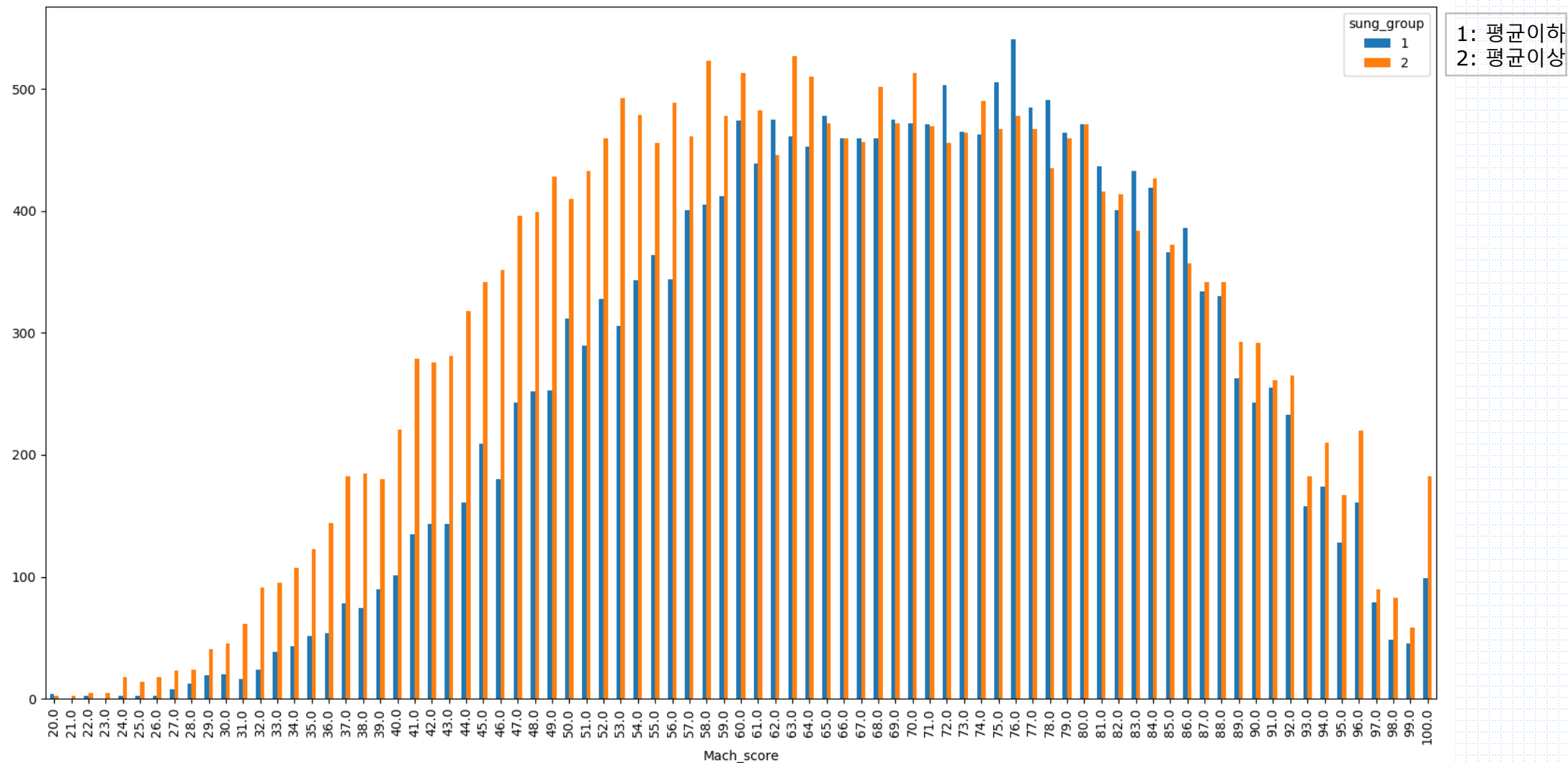


## 3) EDA -

Mach\_Score

- Mach\_score 가 높을수록 성실성이 높아진다는 기존의 논리와 맞지 않음

# Mach\_score + 성실성 상관관계



## 3) EDA

## 특성중요도

- W컬럼의 중요도가 낮게 나와 컬럼 제외
- TIPI테스트를 점수화한 결과, 특성 중요도가 높아짐

## 상관관계(1)

- 나이가 어릴수록, Mach\_Score가 높다
  - Mach\_Score가 높을수록, 투표율이 낮다
- => 나이가 어릴수록, 투표율이 낮다

## 상관관계(2)

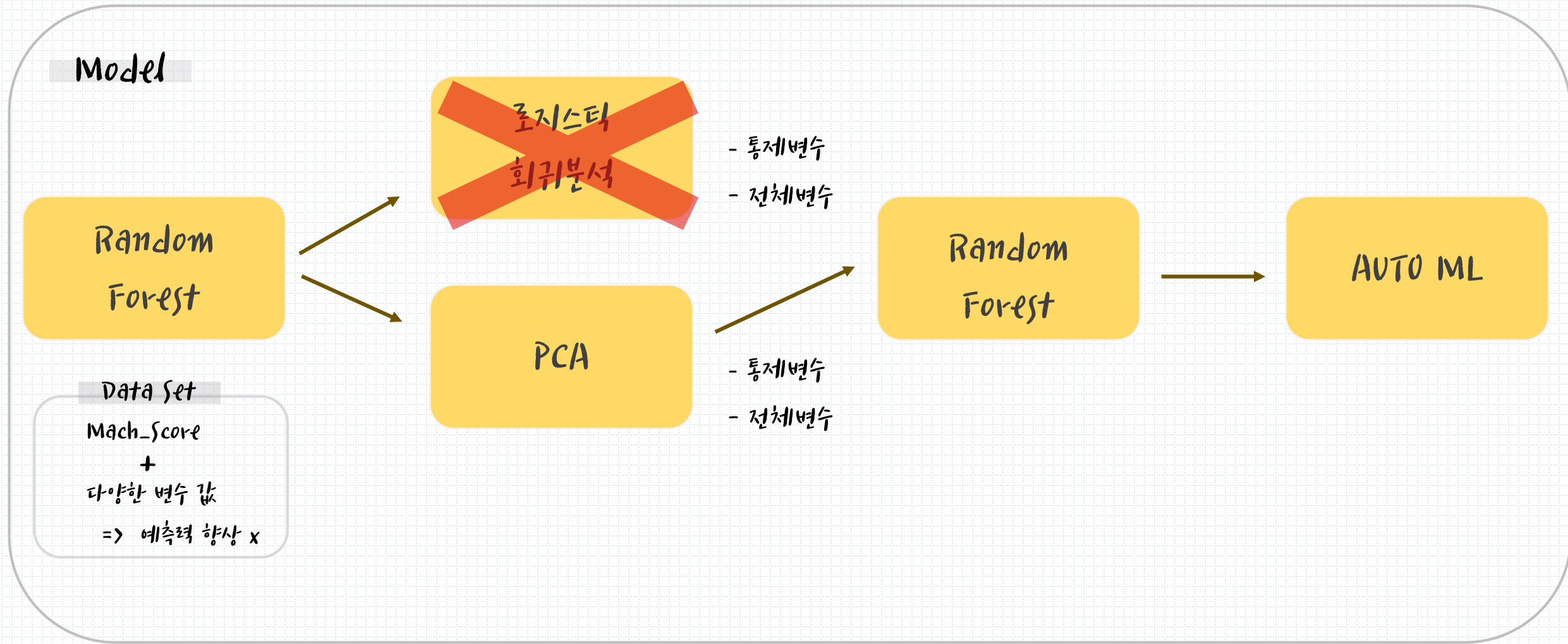
- 교육 수준이 낮을수록, 투표율이 낮다

## 상관관계(3)

- Mach\_Score성향과 TIPI테스트의 약한 상관관계



## 3) EDA





#### 4. 모델 소개

## 1) PCA(통제변수)

- ◆ 본 모델은 마키아벨리즘 변수들의 가중치를 구하기 위해 사용한 모델로서  
마키아벨리즘 변수들을 제외한 다른 변수들을 통제하여 유의미한 회귀계수를 구하기 위해 사용하였습니다.

## 1. 모델 학습 : 통제변수

```
In [56]: x_h_control
```

```
Out[56]:
```

통제변수 선정 :

나이, 학력, 종교, 인종 등

인구통계학적 컬럼의 값이 완벽하게

같은 행을 찾아서 선정

...	urban	gender_code	age_group_code	race_code	religion_code	sung	chin	jung	kyung
...	2	1	1	6	1	5.0	4.0	7.0	4.5
...	2	1	1	6	1	5.5	3.5	5.5	5.0
...	2	1	1	6	1	6.0	5.0	6.5	6.0
...	2	1	1	6	1	5.0	2.0	4.0	5.0
...	2	1	1	6	1	2.0	3.5	2.0	5.5
...	...	...	...	...	...	...	...	...	...
...	2	1	1	6	1	2.0	2.0	7.0	6.5
...	2	1	1	6	1	5.5	6.0	7.0	7.0
...	2	1	1	6	1	6.0	4.0	7.0	5.0
...	2	1	1	6	1	6.5	3.5	5.5	4.0
...	2	1	1	6	1	7.0	4.0	6.0	6.0

499 rows x 33 columns

## 1) PCA(통제변수)

1. 모델 학습 : 1) 변수통제 후 Q\_Ques 컬럼만 뽑아 인공변수 생성 및 변수제거 후 RF (최적의 인공변수 개수 결정)

```
In [15]: # 3. PCA
# 1) 인공변수 생성
vscore = []
for i in [0.7, 0.75, 0.8, 0.85, 0.9, 0.95]:
    from sklearn.decomposition import PCA
    m_pca = PCA(n_components = i)
    m_pca.fit(x_h_control)
    x_pca = m_pca.transform(x)
    df2_pca = m_pca.transform(df2_1)

    # 2) 인공변수 대입
    # 인공변수만을 가지는 데이터프레임 d1, d2 생성
    s1_columns = np.arange(1, len(x_pca[1]) + 1)
    d1 = DataFrame(x_pca, columns = s1_columns)
    d2 = DataFrame(df2_pca, columns = s1_columns)

    # tpscore, human(engnat, familysize, hand 제외) 컬럼을 가지는 데이터프레임 col1, col2 생성
    col1 = df1.drop(['voted'], axis = 1).drop(Q_Ques, axis = 1).drop('engnat', axis = 1)
    col2 = df2.drop(Q_Ques, axis = 1).drop('engnat', axis = 1)

    c1 = d1.columns.tolist()
    c2 = col1.columns.tolist()
    c3 = c1 + c2

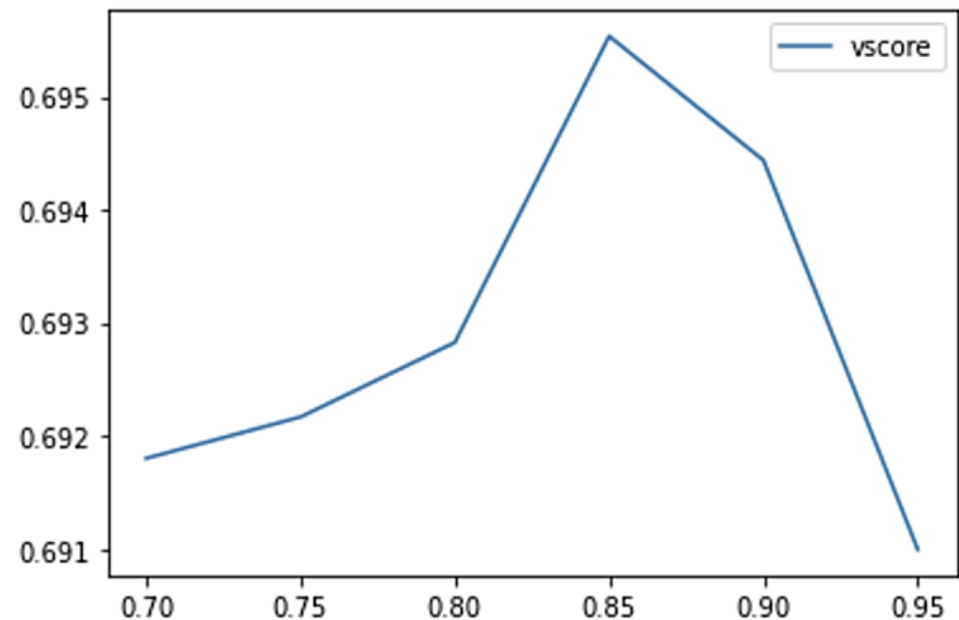
    # d1, d2에 나머지 컬럼데이터 추가
    df1_new = DataFrame(np.hstack([d1, col1]), columns = c3)
    # df1_new['y'] = y => automl용

    df2_new = DataFrame(np.hstack([d2, col2]), columns = c3)

# 4. RF 모델적용
# 1) test, train split
from sklearn.model_selection import train_test_split
train_x, test_x, train_y, test_y = train_test_split(df1_new,
                                                    y,
                                                    train_size = 0.7,
                                                    random_state = 0)

# 2) 모델링
m_rf = rf(random_state = 0)
m_rf.fit(train_x, train_y)
vscore.append(m_rf.score(test_x, test_y))    # 0.6955 (i = 0.85, 13개 차원)
```

13개 인공변수일 때  
예측력이 가장 높음



## 1) PCA(통제변수)

## 1. 모델 학습 : RF 매개변수 튜닝

```
# train_test_split
train_x, test_x, train_y, test_y = train_test_split(df1_new, y, random_state= 0)

v_score_te = [] ; v_score_tr = []
for i in range(1, 101) :
    m_rf = rf(random_state = 0, n_estimators = i)
    m_rf.fit(train_x, train_y)
    v_score_tr.append(m_rf.score(train_x, train_y))
    v_score_te.append(m_rf.score(test_x, test_y))

max(v_score_te)      # 0.698059
Series(v_score_te).sort_values(ascending = False)
```

```
97      0.698059
99      0.697971
98      0.697619
91      0.696477
96      0.696038
...
4       0.639989
3       0.632346
2       0.625494
1       0.607485
0       0.599578
Length: 100, dtype: float64
```

n\_estimators가 98일  
때 가장 높은 예측력 보  
임

## 1) PCA(통제변수)

## 1. 모델 학습 제출 및 결과

```
In [37]: # RF
m_rf = rf(random_state = 0, n_estimators = 98)
m_rf.fit(df1_new, y)

pred_y = m_rf.predict(df2_new)
submission['voted'] = pred_y
```

```
In [38]: sum(submission['voted'] == 1)
```

```
Out[38]: 45
```

```
In [39]: sum(submission['voted'] == 2)
```

```
Out[39]: 11338
```

```
In [ ]: submission.to_csv('sample_submission_PCA2.csv')    # 0.52
```

## 04. 모델 소개

### 1) PCA(전체, 통제변수)

- ◆ 본 모델은 마키아벨리즘 변수들의 가중치를 구하기 위해 사용한 모델로서 마키아벨리즘 변수들을 제외한 다른 변수들을 통제하여 유의미한

회귀계수를 구하기 위해 사용하였습니다.

1. 모델 학습 : 같은 방식으로 통제변수 없이 전체 데이터셋으로 다시 시도

```
In [28]: sum(submission['voted'] == 1)
```

```
Out[28]: 6045
```

```
In [29]: sum(submission['voted'] == 2)
```

```
Out[29]: 5338
```

```
In [30]: submission.to_csv('sample_submission_전체_PCA3.csv')    # 0.6998
```

## 2) | AUTO M

- ◆ 본 모델은 머신 러닝을 실제 문제에 적용하는 프로세스를 자동화 하는 프로세스로서,  
어떤 모델이 최적인지, 사용된 모델의 최적의 매개 변수 값을 찾아주는 과정을 자동화 해주는 모델

목적 : 자동화를 통한 생산성 & 효율성 증가

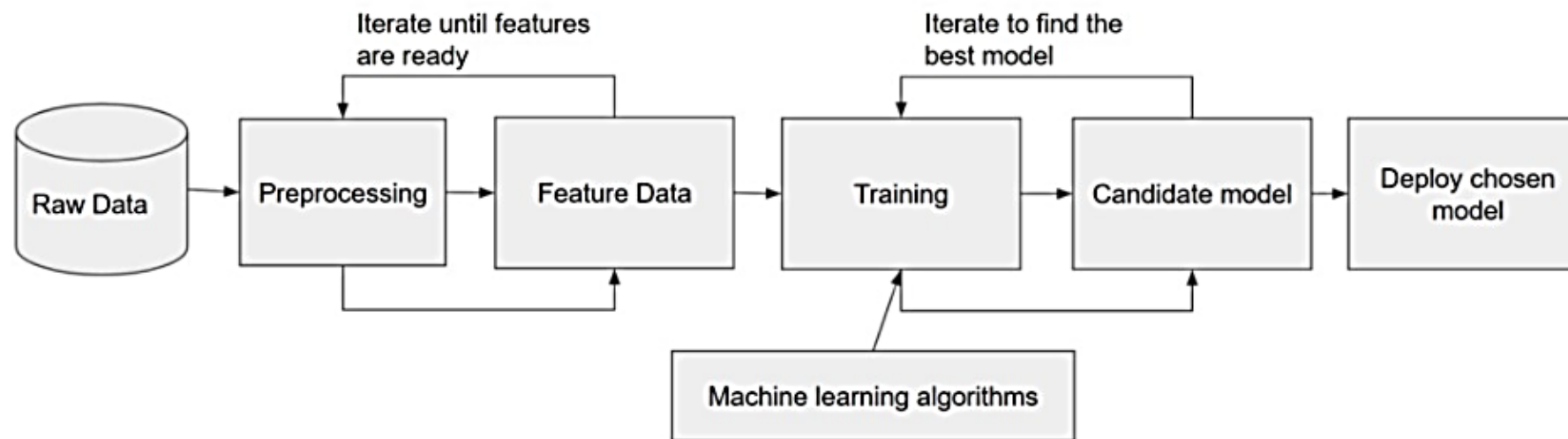


Figure 1. Machine learning process



## 2) | AUTO ML - 모델 학습 및 비교

- ◆ 분석을 위해 최종 선발된 모델로서 15개의 기본 모델을 학습하고 모델들의 성능을 비교 & 분석하여, 최적의 성능을 가지는 3개의 모델들을 앙상블을 통해 구현하는 방법을 사용하였습니다.

## 실험 환경 구축

- PyCaret 활용 시 모델 학습 전 실험환경을 구축 필요
- Setup 단계를 통해 자동으로 컬럼 형태 인식

```

1  pip install pycaret
2  from pycaret.classification import *
3
4  clf = setup(data = df1_new, target = 'voted')
5
6  best_3 = compare_models(sort = 'AUC', n_select = 3)
7
8  blended = blend_models(estimator_list = best_3, fold = 5, method = 'soft')
9
10 pred_holdout = predict_model(blended)
11
12 final_model = finalize_model(blended)
13
14 predictions = predict_model(final_model, data = df2_new)
15
16 submission['voted'] = predictions['Score']
17
18 submission.to_csv('auto_pca.csv', index = True)

```



	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
0	Gradient Boosting Classifier	0.6924	0.7642	0.6461	0.7560	0.6967	0.3886	0.3936	8.5047
1	Light Gradient Boosting Machine	0.6911	0.7619	0.6440	0.7552	0.6951	0.3860	0.3911	0.2375
2	CatBoost Classifier	0.6911	0.7614	0.6574	0.7474	0.6995	0.3845	0.3879	9.0749
3	Extra Trees Classifier	0.6866	0.7581	0.6606	0.7388	0.6974	0.3745	0.3771	0.5922
4	Ada Boost Classifier	0.6866	0.7577	0.6523	0.7431	0.6947	0.3755	0.3789	1.9552
5	Extreme Gradient Boosting	0.6769	0.7457	0.6595	0.7248	0.6906	0.3540	0.3558	0.9434
6	Linear Discriminant Analysis	0.6697	0.7436	0.7245	0.6881	0.7058	0.3299	0.3305	0.0969
7	Logistic Regression	0.6698	0.7434	0.7244	0.6883	0.7058	0.3301	0.3307	0.0447
8	Naive Bayes	0.4749	0.7258	0.0723	0.0690	0.0706	0.0333	0.0333	0.0219
9	Quadratic Discriminant Analysis	0.4708	0.7216	0.0429	0.5866	0.0572	0.0286	0.0329	0.0250
10	Random Forest Classifier	0.6624	0.7193	0.6109	0.7279	0.6642	0.3299	0.3351	0.1391
11	K Neighbors Classifier	0.6313	0.6673	0.6657	0.6619	0.6638	0.2557	0.2557	0.3714
12	Decision Tree Classifier	0.6137	0.6105	0.6447	0.6474	0.6460	0.2210	0.2210	0.4515
13	SVM - Linear Kernel	0.6604	0.0000	0.7273	0.6769	0.7005	0.3093	0.3113	0.1968
14	Ridge Classifier	0.6696	0.0000	0.7246	0.6879	0.7058	0.3296	0.3302	0.0234

## 2) | AUTO ML - 모델 앙상블

- ◆ 분석을 위해 최종 선발된 모델로서 15개의 기본 모델을 학습하고 모델들의 성능을 비교 & 분석하여, 최적의 성능을 가지는 3개의 모델들을 앙상블을 통해 구현하는 방법을 사용하였습니다.

## 모델 선발

- AUC 기준 성능이 가장 좋은 3개의 모델 선발
- Score 최적화를 위해 soft vote ensemble 사용

```

1 pip install pycaret
2 from pycaret.classification import *
3
4 clf = setup(data = df1_new, target = 'voted')
5
6 best_3 = compare_models(sort = 'AUC', n_select = 3)
7
8 blended = blend_models(estimator_list = best_3, fold = 5, method = 'soft')
9
10 pred_holdout = predict_model(blended)
11
12 final_model = finalize_model(blended)
13
14 predictions = predict_model(final_model, data = df2_new)
15
16 submission['voted'] = predictions['Score']
17
18 submission.to_csv('auto_pca.csv', index = True)

```



	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.6965	0.7661	0.6535	0.7581	0.7019	0.3962	0.4007
1	0.6880	0.7577	0.6483	0.7476	0.6944	0.3790	0.3830
2	0.6963	0.7711	0.6480	0.7611	0.7000	0.3964	0.4017
3	0.6936	0.7619	0.6479	0.7567	0.6981	0.3908	0.3957
4	0.6944	0.7631	0.6496	0.7569	0.6992	0.3923	0.3970
Mean	0.6937	0.7640	0.6495	0.7561	0.6987	0.3909	0.3956
SD	0.0031	0.0045	0.0021	0.0045	0.0025	0.0064	0.0067

## 2) | AUTO ML - 모델 예측

- ◆ 분석을 위해 최종 선발된 모델로서 15개의 기본 모델을 학습하고 모델들의 성능을 비교 & 분석하여, 최적의 성능을 가지는 3개의 모델들을 앙상블을 통해 구현하는 방법을 사용하였습니다.

## 제출

- Train dataset을 통해 예측률 확인
- 최적의 성능을 위해 전체 데이터 재학습 실시
- Test dataset을 통한 최종 확률 값 추출하여 제출

```
1 pip install pycaret
2 from pycaret.classification import *
3
4 clf = setup(data = df1_new, target = 'voted')
5
6 best_3 = compare_models(sort = 'AUC', n_select = 3)
7
8 blended = blend_models(estimator_list = best_3, fold = 5, method = 'soft')
9
10 pred_holdout = predict_model(blended)
11
12 final_model = finalize_model(blended)
13
14 predictions = predict_model(final_model, data = df2_new)
15
16 submission['voted'] = predictions['Score']
17
18 submission.to_csv('auto_pca.csv', index = True)
```



	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	Voting Classifier	0.6985	0.7684	0.6487	0.7644	0.7018	0.4011	0.4066



5. 결론

## 2) | AUTO ML - 최종 제출

- ◆ 분석을 위해 최종 선발된 모델로서 15개의 기본 모델을 학습하고 모델들의 성능을 비교 & 분석하여, 최적의 성능을 가지는 3개의 모델들을 앙상블을 통해 구현하는 방법을 사용하였습니다.

1	1996	💡 🐼 🐼	0.78601	129	18시간 전
2	정재환	🐼	0.78348	99	12시간 전
3	harryjo97	🐼 🐼	0.78348	118	13시간 전
30	andys	🌐	0.78012	67	8일 전
31	zeorjin	🌈	0.78011	27	2일 전
32	okso6441	🐼	0.78008	77	10일 전
33	자카종신	🐼	0.78	11	12일 전
34	둘루	🐼 🐼 🌱	0.77996	51	19시간 전
35	schbd_lms	🐼	0.77996	15	하루 전
36	최정명	👤	0.77992	5	한 달 전
37	Choi_0605	🐼	0.7799	21	하루 전
38	nunnunanna	🐼	0.77989	25	7일 전
39	Pura Vida	👉	0.77979	23	10시간 전
40	숯	🐼	0.77962	12	4일 전

Auto ML을 활용한  
최종 예측률  
0.77996





## 6. 한계점

### 1. 마키아벨리즘 심리테스트 척도의 신뢰도 문제점

- 인구통계학적인 변인에 따라서 신뢰도의 차이가 심한 편  
=> 남자는 신뢰도가 0.7이지만, 여자는 0.4로 상당히 낮은 편

### 2. 외부 데이터 활용 제한

- 외부 데이터 사용이 불가하여 마키아벨리즘 테스트만을 이용하여 예측을 하는데 무리가 있음  
=> 소득 분위와 같은 외부 자료를 사용할 수 있었다면 도움이 됐을 것

### 3. 주최사측의 데이터 임의 수정 및 삭제

- 기존 설문지에는 전공선택 및 성적취향 문항도 있었으나 주최측에서 임의로 삭제함  
=> 위의 변수도 제공되었다면 예측력을 높이는 데 도움이 될 것



## 포털 사이트

- 위키피디아 [https://en.wikipedia.org/wiki/Automated\\_machine\\_learning](https://en.wikipedia.org/wiki/Automated_machine_learning) (머신러닝 정의)
- 지식백과 <https://terms.naver.com/entry.nhn?docId=1091092&cid=40942&categoryId=31645> (마키아벨리즘)
- 위키피디아 [https://en.m.wikipedia.org/wiki/Voter\\_turnout](https://en.m.wikipedia.org/wiki/Voter_turnout) (voter turnout 문서)
- 브런치 <https://brunch.co.kr/@a376100/45> (TIPI)
- <https://gosling.psy.utexas.edu/scales-weve-developed/ten-item-personality-measure-tipi/> (TIPI)

## 논문

- 김희송, 홍현기, 현명호, 한국판 마키아벨리즘 척도(MPS)의 타당화 및 신뢰도 연구(국립과학수사연구원, 2011), 2
- 이내영, 유권자 투표참여에 영향을 미치는 요인에 관한 연구(동아시아연구원), 2010, 12



질문 있나요?



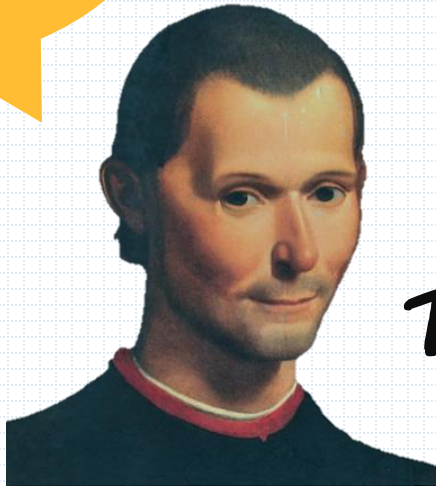
Q&A



# WIKI-MACH

투표 할까?

투표 안할까?



감사합니다

