# KUBIG CONTEST

## 머신러닝 분반 < 심리 성향 예측 >

이은찬 이수찬 이영노 임채명 정은미

# 목차 contents

Data analysis 9~13 page

EDA

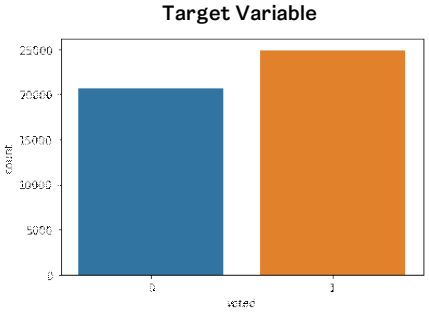
Data Preprocessing

Modeling Ensemble

심리 성향 테스트를 활용하여 설문자의 투표 여부를 맞추는 알고리즘

```
1 train.shape
```

```
(45532, 78)
```




voted - 지난 해 국가 선거 투표 여부 (1=Yes, 2=No) => **(0=Yes, 1=No)**

## Q_A (a~t) : 비식별화를 위해 일부 질문은 Secret 처리

Qa : Secret
Qb : The biggest difference between most criminals and other people is that the criminals are stupid enough to get caught.
Qc : Anyone who completely trusts anyone else is asking for trouble.
Qd : Secret
Qe : P.T. Barnum was wrong when he said that there's a sucker born every minute.
Qf : There is no excuse for lying to someone else.
Qg : Secret
Qh : Most people forget more easily the death of their parents than the loss of their property.
Qi : Secret
Qj : It is safest to assume that all people have a vicious streak and it will come out when they are given a chance.
Qk : All in all, it is better to be humble and honest than to be important and dishonest.
Ql : Secret
Qm : It is hard to get ahead without cutting corners here and there.
Qn : Secret
Qo : The best way to handle people is to tell them what they want to hear.
Qp : Secret
Qq : Most people are basically good and kind.
Qr : One should take action only when sure it is morally right.
Qs : It is wise to flatter important people.
Qt : Secret
1=Disagree, 2=Slightly disagree, 3=Neutral, 4=Slightly agree, 5=Agree.

```
1 train['QaA'].head()
```

```
0    3.0
1    5.0
2    4.0
3    3.0
4    1.0
Name: QaA, dtype: float64
```

=> 마키아벨리즘 성향을 파악

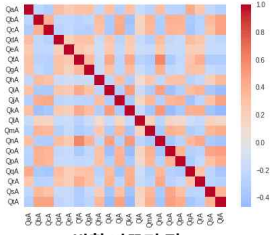
## Q_A (a~t) : 비식별화를 위해 일부 질문은 Secret 처리

Qm : It is hard to get ahead without cutting corners here and there.

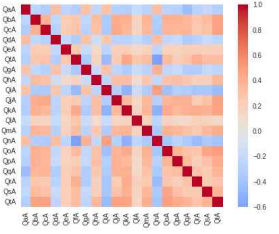
Qq : Most people are basically good and kind.

1=Disagree, 2=Slightly disagree, 3=Neutral, 4=Slightly agree, 5=Agree.

**=> 방향이 반대**



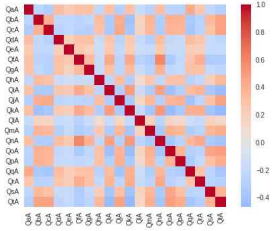
방향 바꾸기 전



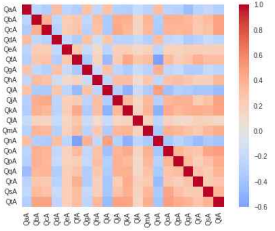
방향 바꾸기 후

Secret 문항 변수도
방향 전환이 필요함을 확인

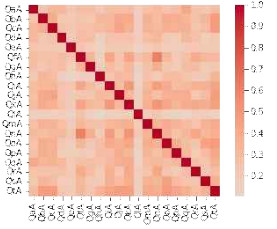
Q_A (a~t) : 비식별화를 위해 일부 질문은 Secret 처리



방향 바꾸기 전



방향 바꾸기 후



Secret 변수까지 방향 바꾼 후

Q_A (a~t) : 비식별화를 위해 일부 질문은 Secret 처리

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


flipping_secret_columns = ["QaA", "QdA", "QgA", "QiA", "QnA"]
for data in dataset:
  for flip in flipping_secret_columns:
    data[flip] = 6 - data[flip]
```

이를 바탕으로

1. 모든 변수 모델링 포함
2. 마키아벨리즘 성향을 나타내는 하나의 파생 변수 생성
3. 각각 Tactic/ Morality/ View를 나타내는 세 개의 파생 변수 생성

tp__(01~07) : items were rated "I see myself as:" _____ such that

tp01 : Extraverted, enthusiastic.
tp02 : Critical, quarrelsome.
tp03 : Dependable, self-disciplined.
tp04 : Anxious, easily upset.
tp05 : Open to new experiences, complex.
tp06 : Reserved, quiet.
tp07 : Sympathetic, warm.
tp08 : Disorganized, careless.
tp09 : Calm, emotionally stable.
tp10 : Conventional, uncreative.

```
flipping_columns = ["tp06", "tp02", "tp08", "tp04", "tp10"]
for data in dataset:
  for flip in flipping_columns:
      data[flip] = 7 - data[flip]
```

```
for data in dataset:
  data['Ex'] = (data['tp01']+data['tp06'])/2
  data['Ag'] = (data['tp07']+data['tp02'])/2
  data['Con'] = (data['tp03']+data['tp08'])/2
  data['Es'] =(data['tp09']+data['tp04'])/2
  data['Op'] =(data['tp05']+data['tp10'])/2
```

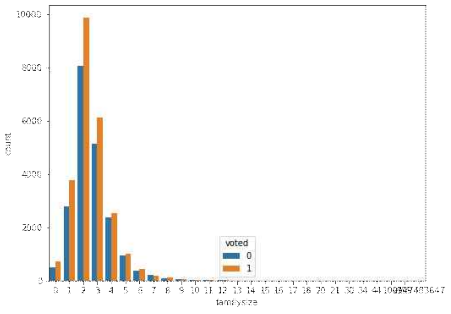
상반되는 질문들을 방향을 맞춰주고 성향에 따라 새로운 5개의 파생변수 생성

wr_(01~13) : 실존하는 해당 단어의 정의를 앎
wf_(01~03) : 허구인 단어의 정의를 앎

1. 모든 변수 포함하여 모델링
2. Wr_ 변수 13개가 나타내는 정보가 동일, 모든 변수가 필요하지 않다고 판단하여 임의로 5개 선택
3. 존재하지 않는 3개의 단어를 선택한 응답자는 신뢰도가 낮다고 평가하여

이를 위해 허구의 단어를 안다고 거짓말한 사람과 진실로 답한 사람을 구별하는 파생변수 생성

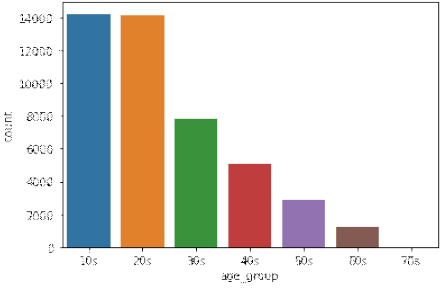
## familysize : 형제자매 수

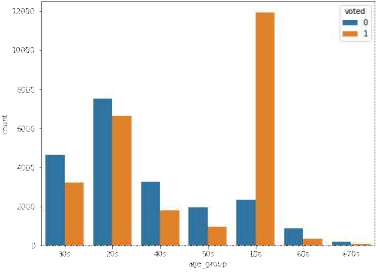



| | familysize | voted |
|---|---|---|
| 0 | 1217 | 0.413311 |
| 1 | 6535 | 0.424331 |
| 2 | 17918 | 0.450106 |
| 3 | 11256 | 0.456912 |
| 4 | 4907 | 0.469391 |
| 5 | 1962 | 0.481651 |
| 6 | 838 | 0.464200 |
| 7 | 387 | 0.519380 |
| 8 | 221 | 0.438914 |
| 9 | 126 | 0.476190 |
| 10 | 59 | 0.474576 |
| 11 | 39 | 0.564103 |
| 12 | 21 | 0.523810 |
| 13 | 11 | 0.636364 |
| 14 | 9 | 0.555556 |
| 15 | 8 | 0.625000 |
| 16 | 2 | 0.000000 |
| 17 | 3 | 1.000000 |
| 18 | 1 | 0.000000 |
| 20 | 2 | 0.000000 |
| 21 | 2 | 0.500000 |
| 30 | 1 | 1.000000 |
| 34 | 1 | 0.000000 |
| 44 | 3 | 0.333333 |
| 100 | 1 | 1.000000 |
| 999 | 1 | 1.000000 |
| 2147483647 | 1 | 0.000000 |

이상치 존재 – 22 이상 제거

## age_group: 연령




- 10대인지 아닌지가 투표 여부에 큰 영향 => 새로운 파생 변수

## 그 외 변수

| | |
|---|---|
| hand : 필기하는 손 | 결과에 큰 영향을 끼치지 않을 것으로 판단하여 제거 |
| index | 결과에 큰 영향을 끼치지 않을 것으로 판단하여 제거 |
| Q_E(a~t) : 질문을 답할 때까지의 시간 | 모두 합하여 새로운 하나의 Delay 변수 생성<br>결과에 큰 영향을 끼치지 않을 것으로 판단하여 제거 |

train.info()

```
37  usA          45532 non-null  float64
38  QsE          45532 non-null  int64
39  QtA          45532 non-null  float64
40  QtE          45532 non-null  int64
41  age_group    45532 non-null  object
42  education    45532 non-null  int64
43  engnat       45532 non-null  int64
44  familysize   45532 non-null  int64
45  gender       45532 non-null  object
46  hand         45532 non-null  int64
47  married      45532 non-null  int64
48  race         45532 non-null  object
49  religion     45532 non-null  object
50  tp01         45532 non-null  int64
51  tp02         45532 non-null  int64
52  tp03         45532 non-null  int64
53  tp04         45532 non-null  int64
54  tp05         45532 non-null  int64
```

범주형 변수 -> 수치형 변수

1. Label encoding
2. one-hot encoding
3. autoML자체처리

## Best AUC 전처리 set

Index Column 제거
방향 전환한 Q_A Score 변수 모두 사용
Categorical 변수는 그대로 유지
=> autoML자체처리

### Train size 할당

pycaret의 setup함수 사용(train_size = 0.8)하여 모델의 AUC 비교 - Catboost의 AUC가 0.7657로 최고 성능

### Train size 할당X

GradientBoostingClassifier의 AUC가 0.7669로 0.12%p 소폭 상승

## AUTOML

| | Model | Accuracy | AUC | Recall | Prec. | F1 | Kappa | MCC | TT (Sec) |
|---|---|---|---|---|---|---|---|---|---|
| gbc | Gradient Boosting Classifier | 0.6959 | 0.7669 | 0.7584 | 0.6378 | 0.6929 | 0.3962 | 0.4021 | 25.256 |
| lightgbm | Light Gradient Boosting Machine | 0.6963 | 0.7663 | 0.7561 | 0.6389 | 0.6925 | 0.3967 | 0.4022 | 1.090 |
| lda | Linear Discriminant Analysis | 0.6931 | 0.7629 | 0.7277 | 0.6418 | 0.6820 | 0.3877 | 0.3906 | 0.851 |
| et | Extra Trees Classifier | 0.6925 | 0.7623 | 0.7495 | 0.6358 | 0.6880 | 0.3889 | 0.3940 | 6.600 |
| ada | Ada Boost Classifier | 0.6887 | 0.7575 | 0.7319 | 0.6354 | 0.6802 | 0.3800 | 0.3836 | 4.833 |
| rf | Random Forest Classifier | 0.6908 | 0.7566 | 0.7494 | 0.6339 | 0.6868 | 0.3857 | 0.3910 | 8.430 |
| dt | Decision Tree Classifier | 0.6084 | 0.6050 | 0.5692 | 0.5670 | 0.5680 | 0.2099 | 0.2100 | 1.502 |
| nb | Naive Bayes | 0.5084 | 0.5321 | 0.4226 | 0.4487 | 0.3081 | 0.0017 | 0.0029 | 0.065 |
| lr | Logistic Regression | 0.5471 | 0.5267 | 0.0050 | 0.4837 | 0.0098 | -0.0002 | 0.0003 | 2.538 |
| knn | K Neighbors Classifier | 0.5117 | 0.5103 | 0.4606 | 0.4602 | 0.4603 | 0.0145 | 0.0145 | 36.296 |
| qda | Quadratic Discriminant Analysis | 0.4528 | 0.5004 | 0.9993 | 0.4525 | 0.6229 | 0.0007 | 0.0131 | 0.486 |
| dummy | Dummy Classifier | 0.5477 | 0.5000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.020 |
| svm | SVM - Linear Kernel | 0.4986 | 0.0000 | 0.4941 | 0.4507 | 0.4702 | -0.0036 | -0.0036 | 0.315 |
| ridge | Ridge Classifier | 0.6931 | 0.0000 | 0.7275 | 0.6419 | 0.6820 | 0.3877 | 0.3906 | 0.121 |

**AUTOML**

**+**

**Blending**

: validation set에 대한 예측값을 학습에 이용

**Soft Voting**

**최종**

**0.778의 점수**

# 소감 및 발전방향



1. 성능을 높이기 위해 복잡한 전처리를 하는게 무조건 능사는 아님

2. validation set 설정 시 split값에 따라 결과가 다르게 나타날 수 있음

3. 머신러닝에는 생각보다 많은 randomness가 존재하기 때문에

안정된 모델을 만들기 위해서는 이를 해결해야 함