2022 2R KUBIG COMPETITION

심리 성향 예측 AI 경진대회

16기 머신러닝 1팀 김진서 임정준 최규빈 하예은

"투표 여부 예측"

CONTENTS

01 EDA

(1) 데이터 구조 파악

```
RangeIndex: 45532 entries, 0 to 45531
Data columns (total 78 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   index       45532 non-null  int64
 1   QaA         45532 non-null  float64
 2   QaE         45532 non-null  int64
 3   QbA         45532 non-null  float64
 4   QbE         45532 non-null  int64
 5   QcA         45532 non-null  float64
 6   QcE         45532 non-null  int64
 7   QdA         45532 non-null  float64
 8   QdE         45532 non-null  int64
 9   QeA         45532 non-null  float64
 10  QeE         45532 non-null  int64
 11  QfA         45532 non-null  float64
 12  QfE         45532 non-null  int64
 13  QgA         45532 non-null  float64
 14  QgE         45532 non-null  int64
 15  QhA         45532 non-null  float64
 16  QhE         45532 non-null  int64
 17  QiA         45532 non-null  float64
 18  QiE         45532 non-null  int64
 19  QjA         45532 non-null  float64
 20  QjE         45532 non-null  int64
 21  QkA         45532 non-null  float64
 22  QkE         45532 non-null  int64
 23  QlA         45532 non-null  float64
 24  QlE         45532 non-null  int64
 25  QmA         45532 non-null  float64
 26  QmE         45532 non-null  int64
 27  QnA         45532 non-null  float64
 28  QnE         45532 non-null  int64
 29  QoA         45532 non-null  float64
 30  QoE         45532 non-null  int64
 31  QpA         45532 non-null  float64
 32  QpE         45532 non-null  int64
 33  QqA         45532 non-null  float64
 34  QqE         45532 non-null  int64
 35  QrA         45532 non-null  float64
 36  QrE         45532 non-null  int64
 37  QsA         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
 55  tp06        45532 non-null  int64
 56  tp07        45532 non-null  int64
 57  tp08        45532 non-null  int64
 58  tp09        45532 non-null  int64
 59  tp10        45532 non-null  int64
 60  urban       45532 non-null  int64
 61  voted       45532 non-null  int64
 62  wf_01       45532 non-null  int64
 63  wf_02       45532 non-null  int64
 64  wf_03       45532 non-null  int64
 65  wr_01       45532 non-null  int64
 66  wr_02       45532 non-null  int64
 67  wr_03       45532 non-null  int64
 68  wr_04       45532 non-null  int64
 69  wr_05       45532 non-null  int64
 70  wr_06       45532 non-null  int64
 71  wr_07       45532 non-null  int64
 72  wr_08       45532 non-null  int64
 73  wr_09       45532 non-null  int64
 74  wr_10       45532 non-null  int64
 75  wr_11       45532 non-null  int64
 76  wr_12       45532 non-null  int64
 77  wr_13       45532 non-null  int64
```

Feature (77개)

Q_A : 질문에 대한 대답 (1 ~ 5)

Q_E : 대답을 하기까지 걸린 시간

tp__ : 자신에 대한 평가 (1 ~ 7)

wf__ : 실존하지 않는 단어를 아는지

wr__ : 실존하는 단어를 아는지

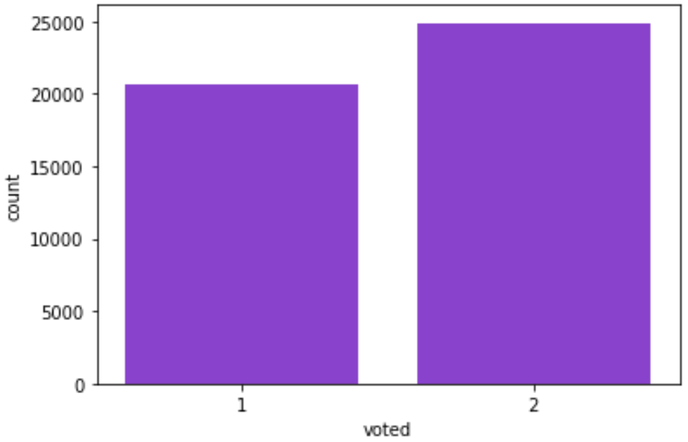
age_group, education, engnat(영어가 모국어),urban, familysize, gender, hand, married, race, religion,

Target (voted)

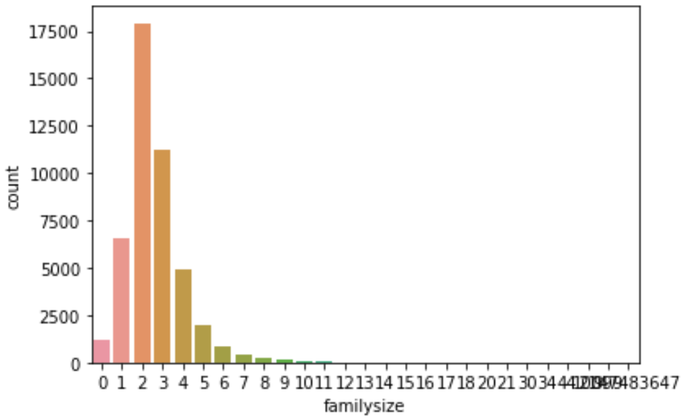
Train Size : 45532, Test Size : 11383

(2) 데이터 시각화

target(voted)

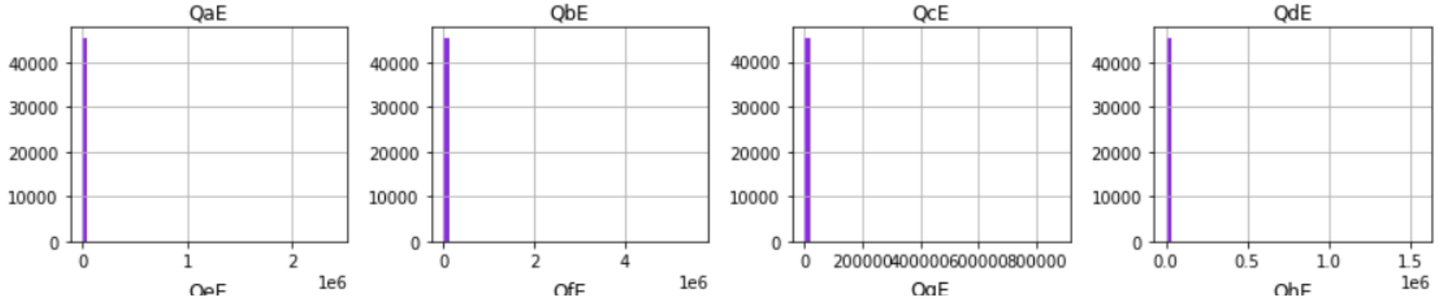


familysize



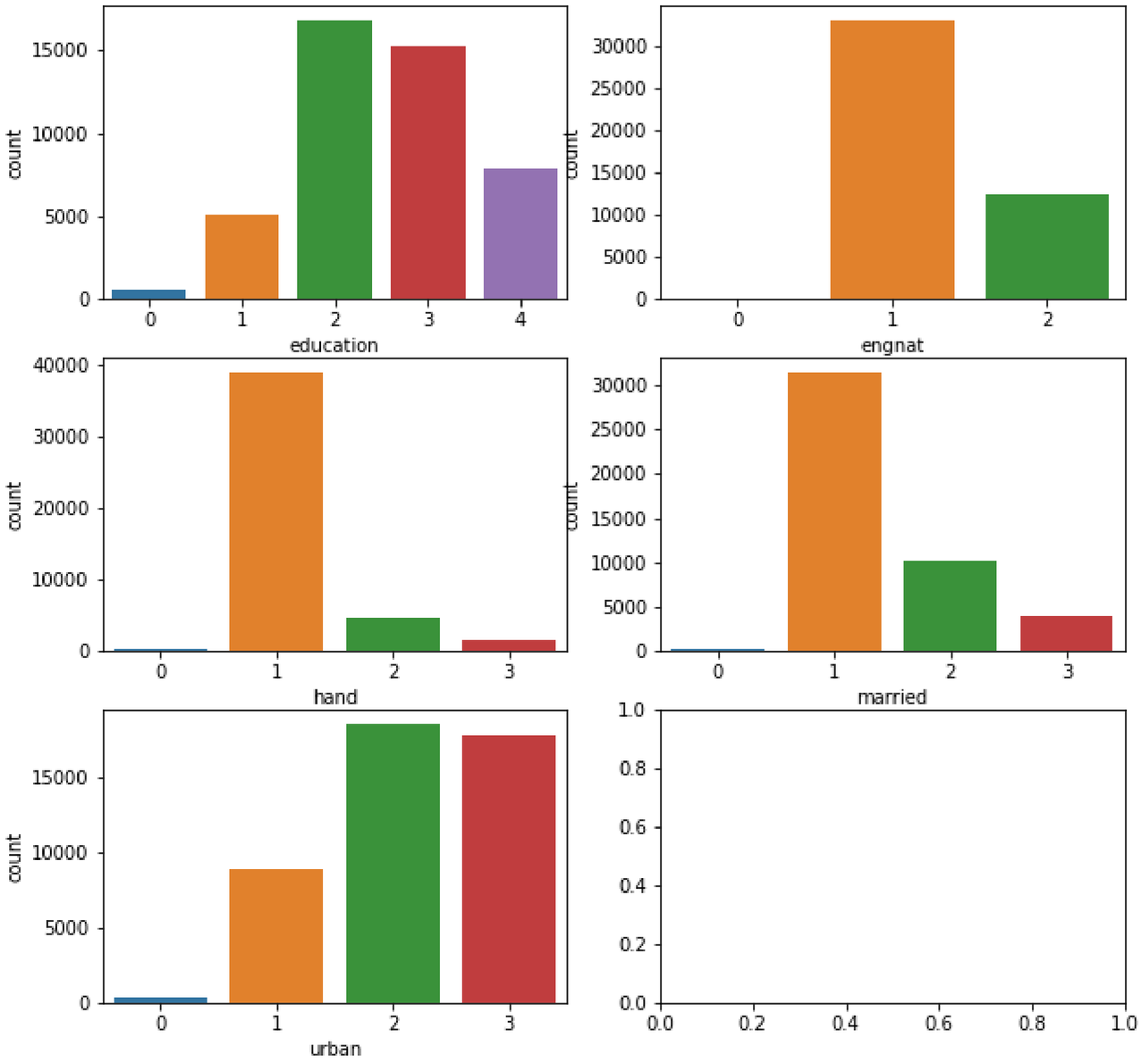
Q_E

	count	mean	std	min	25%	50%	75%	max
QaE	45532.0	945.357046	13075.648143	25.0	404.0	557.0	827.0	2413960.0
QbE	45532.0	2189.588575	33510.265924	25.0	875.0	1218.0	1838.0	5580395.0
QcE	45532.0	1484.294518	8977.664318	25.0	651.0	899.0	1335.0	871557.0
QdE	45532.0	1490.672231	10922.600860	26.0	679.0	931.0	1355.0	1552821.0
QeE	45532.0	1899.292278	16707.654162	25.0	834.0	1154.0	1656.0	1919926.0



매우 큰 분산 & min - q1, q3 - max 간 큰 차이

(3) 결측치 파악



	무응답 개수	무응답 비율
education	528	0.01160
engnat	77	0.00169
hand	161	0.00354
married	93	0.00204
urban	322	0.00707

02 데이터 전처리

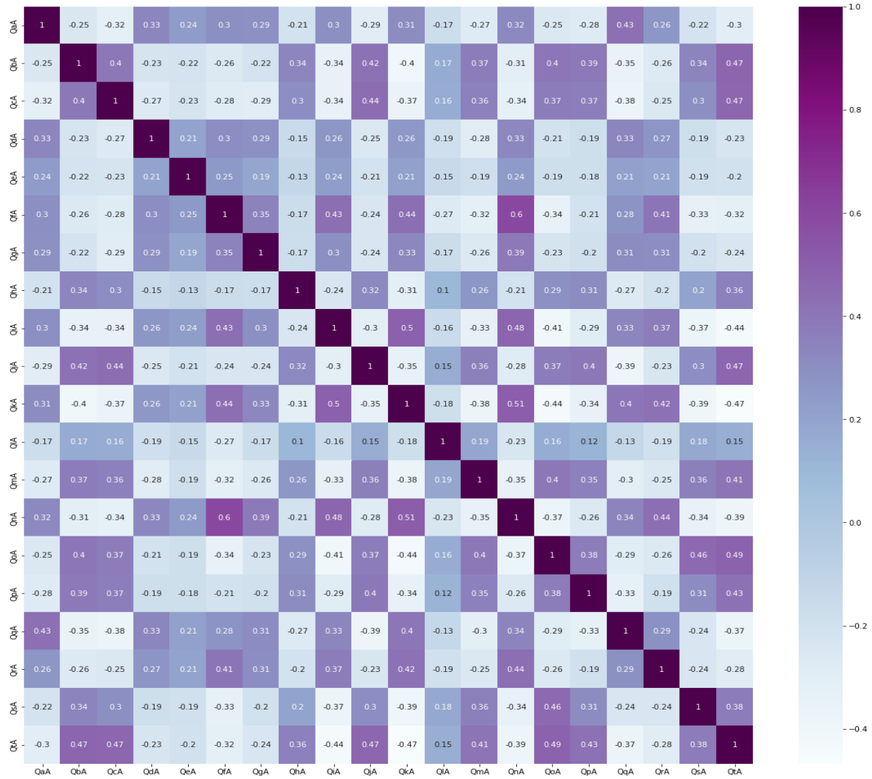
(1) 변수별 전처리 1. Q_A : scoring

Q_A : 마키아벨리즘 성향 테스트 문항

→ 상관계수 부호로 그룹화

성향 강함 : a, d, e, f, g, l, k, n, q, r

성향 약함 : b, c, h, j, l, m, o, p, s, t



성향 약함 문항 답변 reverse 후

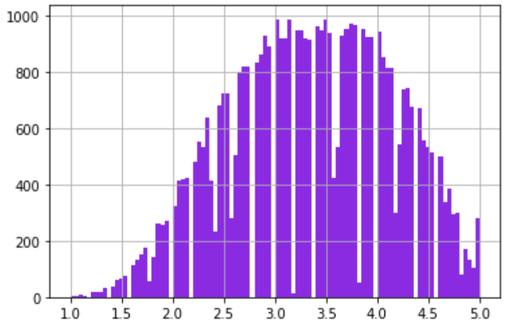
'Mach_score' 변수 생성

&

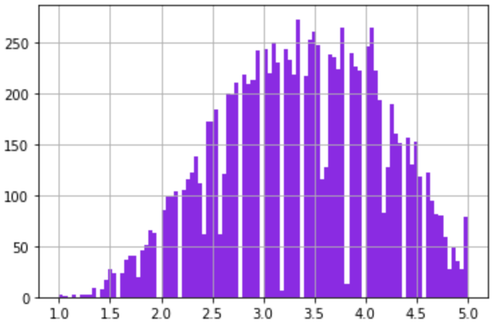
'tactics', 'view', 'morality' 변수 생성하여

검사 유형 구분

<Mach_score>



train



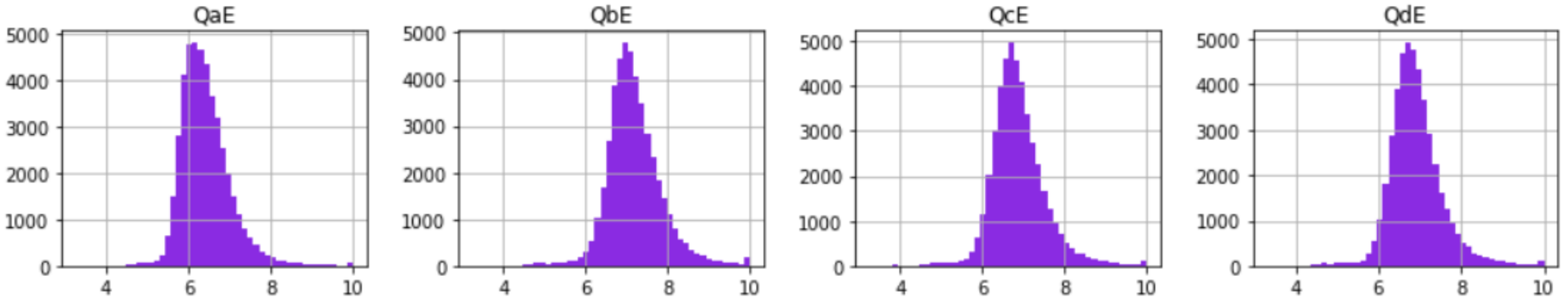
test

(1) 변수별 전처리 2. Q_E : 로그변환 후 값 병합

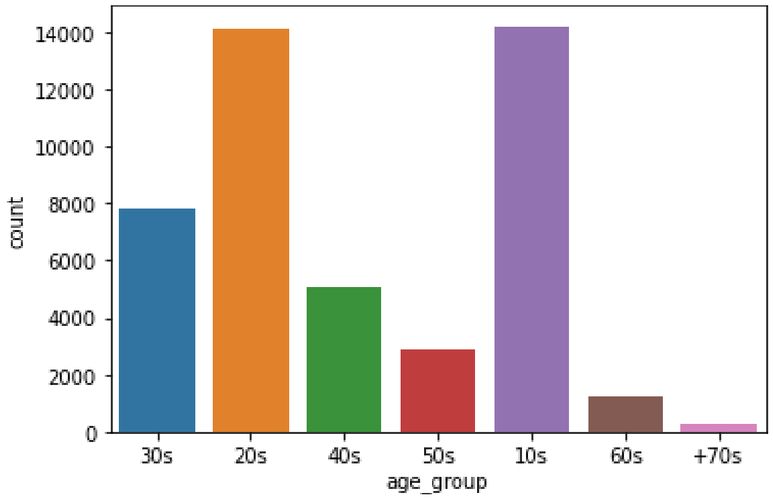
	count	mean	std	min	25%	50%	75%	max
QaE	45532.0	945.357046	13075.648143	25.0	404.0	557.0	827.0	2413960.0
QbE	45532.0	2189.588575	33510.265924	25.0	875.0	1218.0	1838.0	5580395.0
QcE	45532.0	1484.294518	8977.664318	25.0	651.0	899.0	1335.0	871557.0
QdE	45532.0	1490.672231	10922.600860	26.0	679.0	931.0	1355.0	1552821.0
QeE	45532.0	1899.292278	16707.654162	25.0	834.0	1154.0	1656.0	1919926.0

mean	std	min	25%	50%	75%	max
6.407732	0.614277	3.258097	6.003887	6.324359	6.719013	10.0
7.178677	0.678994	3.258097	6.775366	7.105786	7.516977	10.0
6.880644	0.655525	3.258097	6.480045	6.802395	7.197435	10.0
6.905539	0.639446	3.295837	6.522093	6.837333	7.212294	10.0
7.101968	0.676615	3.258097	6.727432	7.051856	7.412764	10.0

로그변환 후 10 이상인 값을 10으로 대체

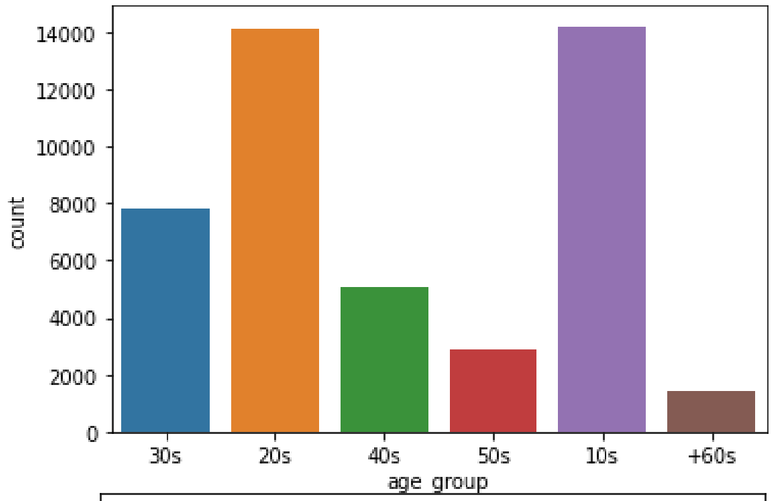


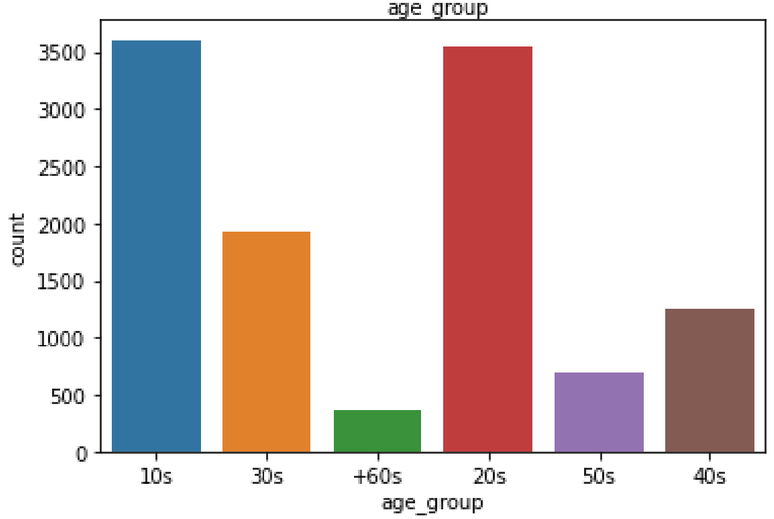
(1) 변수별 전처리 3. age_group : 범주 병합



	age_group	voted
0	+70s	1.263830
1	10s	1.837214
2	20s	1.469671
3	30s	1.411179
4	40s	1.357949
5	50s	1.329872
6	60s	1.271357

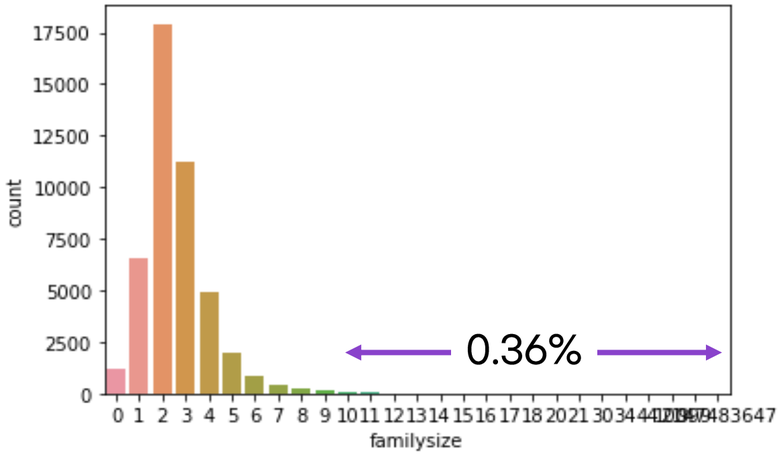
60s & +70s
→ +60s

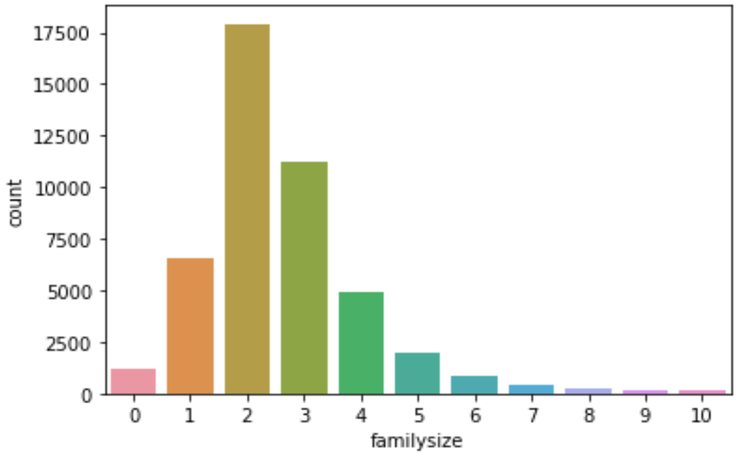




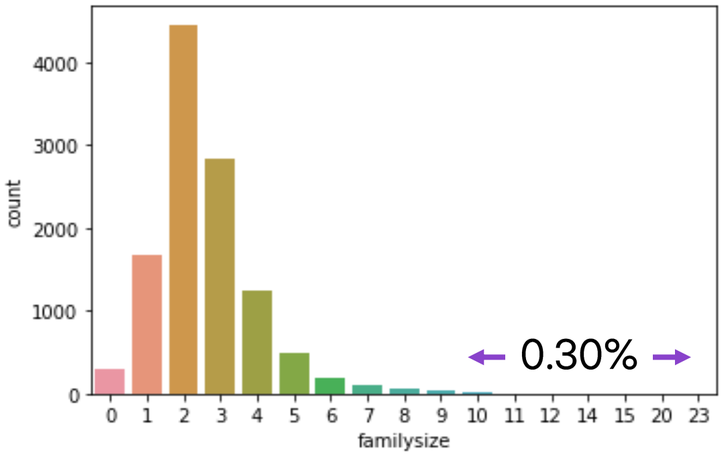
(1) 변수별 전처리 4. familysize : 값 병합

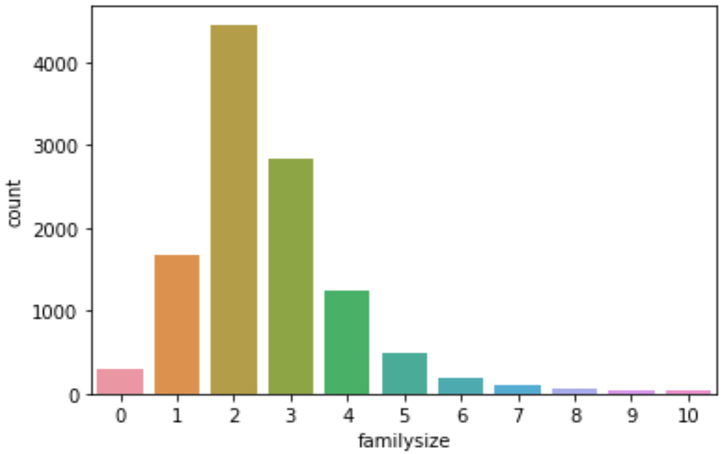
train



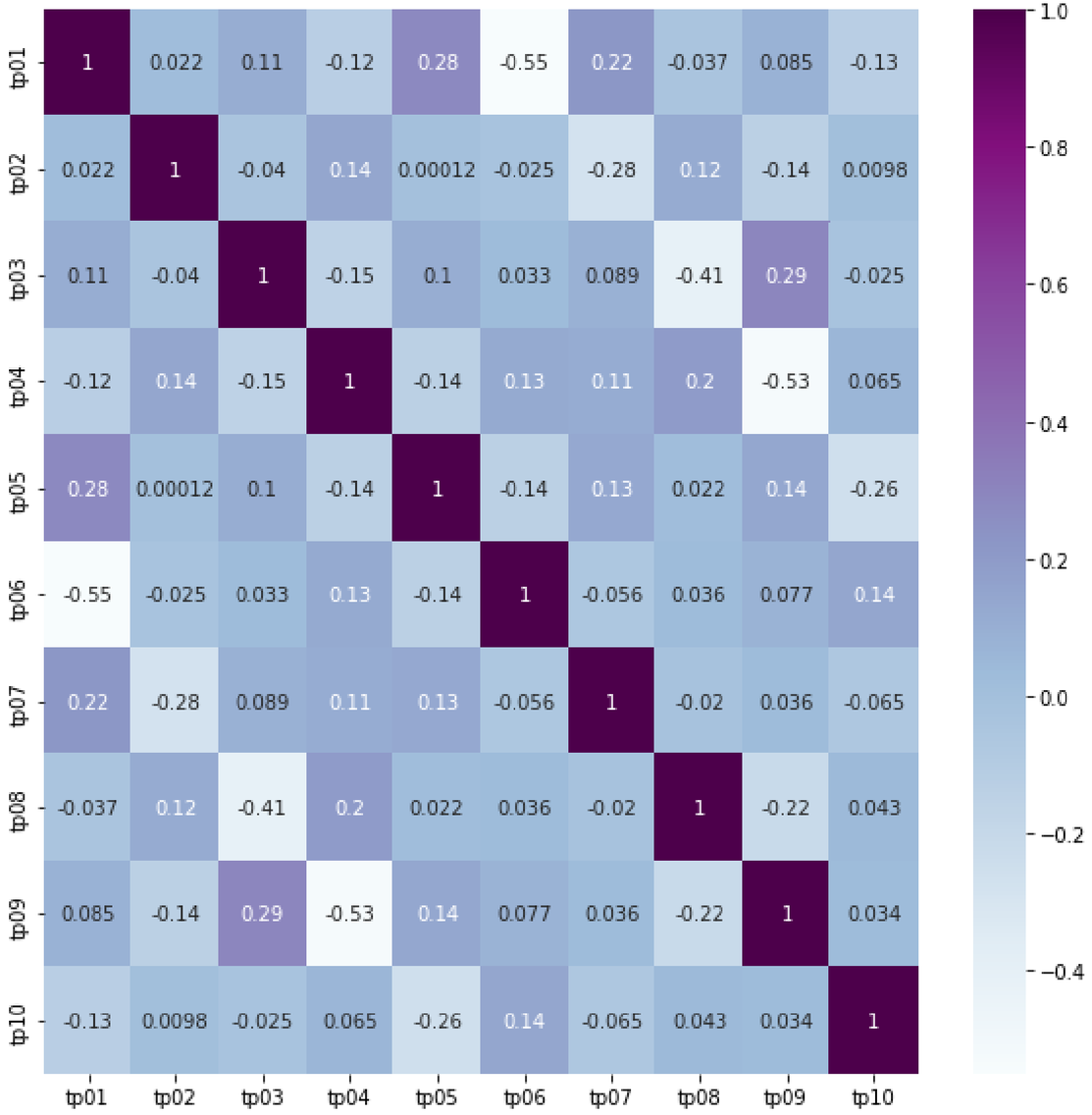


test





(1) 변수별 전처리 5. tp__ : 범주 병합



01 외향적인, 열정적인
02 비판적인, 싸우기 좋아하는
03 믿을 만한, 자기 훈련이 된
04 걱정하는, 쉽게 맘이 상하는
05 새로운 경험을 하는, 복잡한

↔

06 내성적인, 조용한
07 동정적인, 따뜻한
08 체계적이지 못한, 부주의한
09 차분한, 감정적으로 안정된
10 관습적인, 창의적이지 못한

01 & 06R > extraversion
02R & 07 > agreeableness
03 & 08R > conscientiousness
04R & 09 > stability
05 & 10R > openness

→ 5개의 파생변수 생성

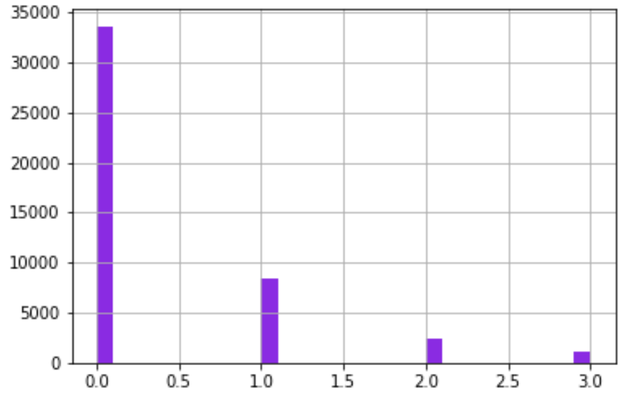
(1) 변수별 전처리

6. wf__ & wr __ : total 변수 생성

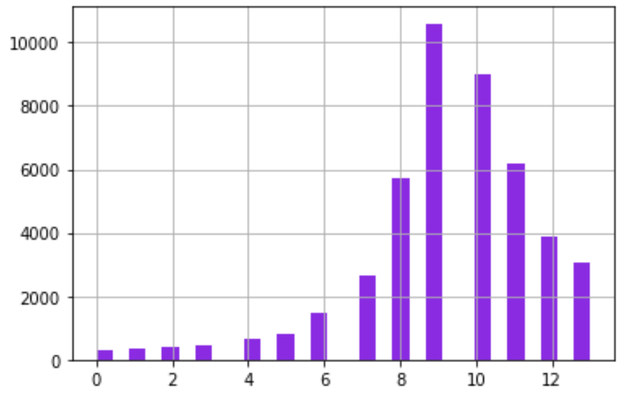
<wf__>
cuivocal
florted
verdid

<wr__>
boat
incoherent
pallid
robot
audible
paucity
epistemology
decide
pastiche
abysmal
lucid
betray
funny

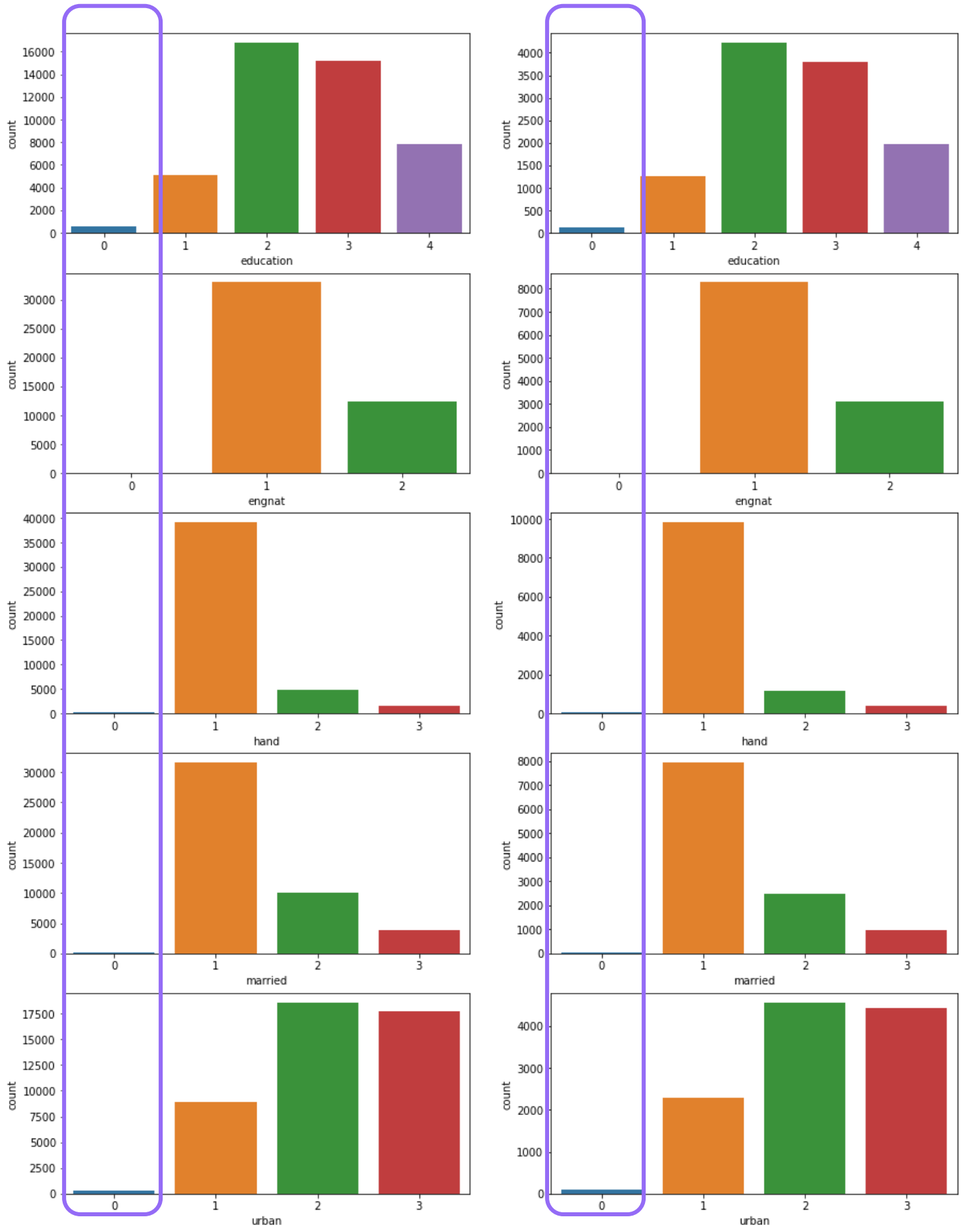
wf__ 변수 총합으로
total_wf 변수 생성



wr__ 변수 총합으로
total_wr 변수 생성



(2) 결측치 처리



education, engnat, hand, married, urban

: 0 = 무응답 = 결측치

1. 0 값 NaN으로 변환

2. Imputation

- KNN Imputation
- DNN Imputation

(3) 범주형 변수 encoding

type='object'인 범주형 변수에 대해 label encoding

: age_group, gender, race, religion

	age_group	gender	race	religion
0	3	0	6	10
1	2	0	1	7
2	3	1	6	10
3	2	0	1	7
4	2	1	6	0

(4) 수치형 변수 정규화

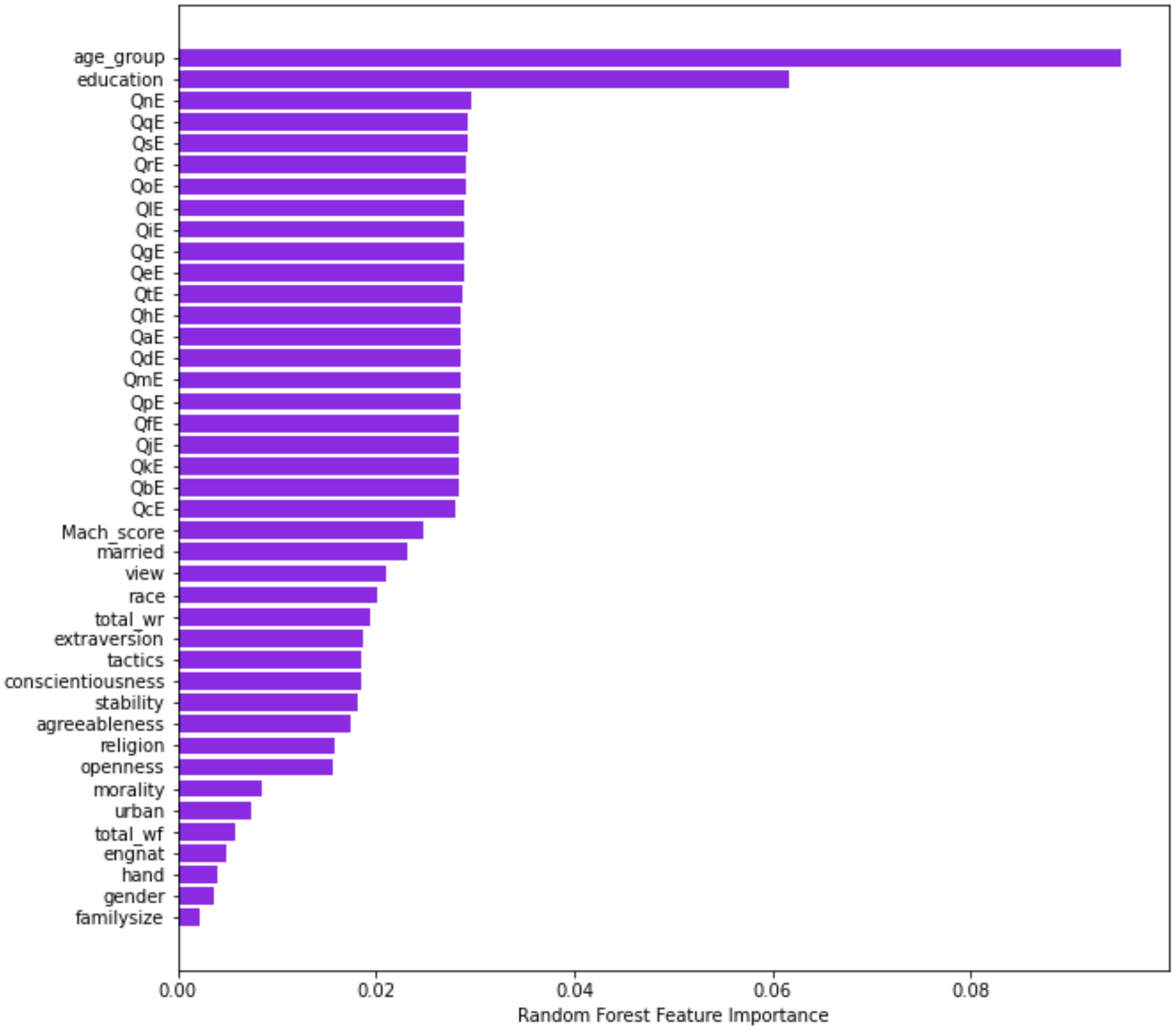
: Mach_score, familysize, extraversion, agreeableness, conscientiousness, stability, openness, total_wf, total_wr

	Mach_score	familysize	extraversion	agreeableness	conscientiousness	stability	openness	total_wr	total_wf
0	-0.506478	-0.004686	0.129628	1.916629	-0.085904	1.118125	0.801315	-0.962165	-0.523797
1	-0.946923	-0.004686	-0.440193	0.888043	-0.742946	0.226893	-0.007245	-0.539503	-0.523797
2	-1.827812	-0.004686	-0.725104	-0.140542	-1.071468	-0.961418	1.205595	0.305820	0.924702
3	-0.003113	-0.004687	-0.440193	-0.826266	-0.085904	0.226893	0.397035	-1.807488	-0.523797
4	-0.443558	-0.004687	-0.440193	-0.140542	-1.071468	-0.961418	-1.220085	0.728481	0.924702

	Mach_score	familysize	extraversion	agreeableness	conscientiousness	stability	openness	total_wr	total_wf
0	-1.135685	-0.004686	-0.155283	0.545181	-0.085904	0.226893	-0.007245	-1.384826	-0.523797
1	-0.695240	-0.004687	-0.155283	0.202320	0.571139	0.226893	1.205595	-3.920795	-0.523797
2	0.248569	-0.004686	1.269269	0.202320	0.899660	1.415203	3.226995	-0.962165	-0.523797
3	-0.003113	-0.004686	-1.864746	-1.511990	-0.742946	-1.258495	-1.220085	1.573804	-0.523797
4	-0.569399	-0.004686	1.554180	1.573767	0.571139	0.821048	1.205595	0.305820	-0.523797

(4) Feature Selection

Random Forest - Feature Importance



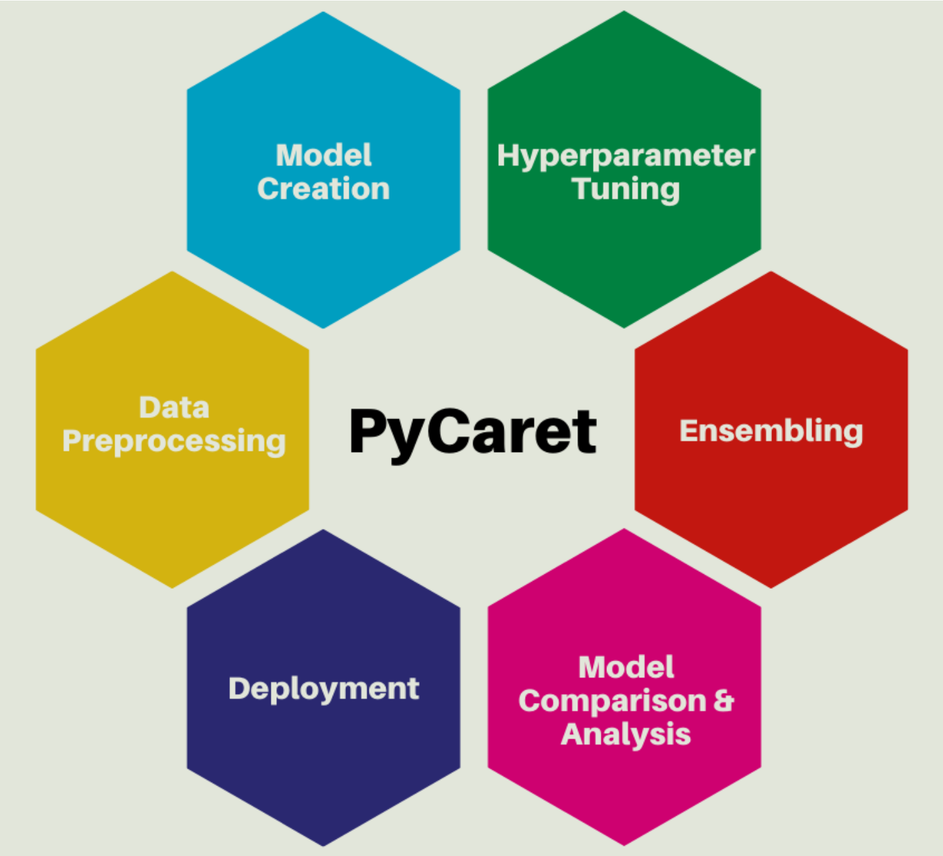
☆ age_group, education, Q_E, Mach_score, married ☆

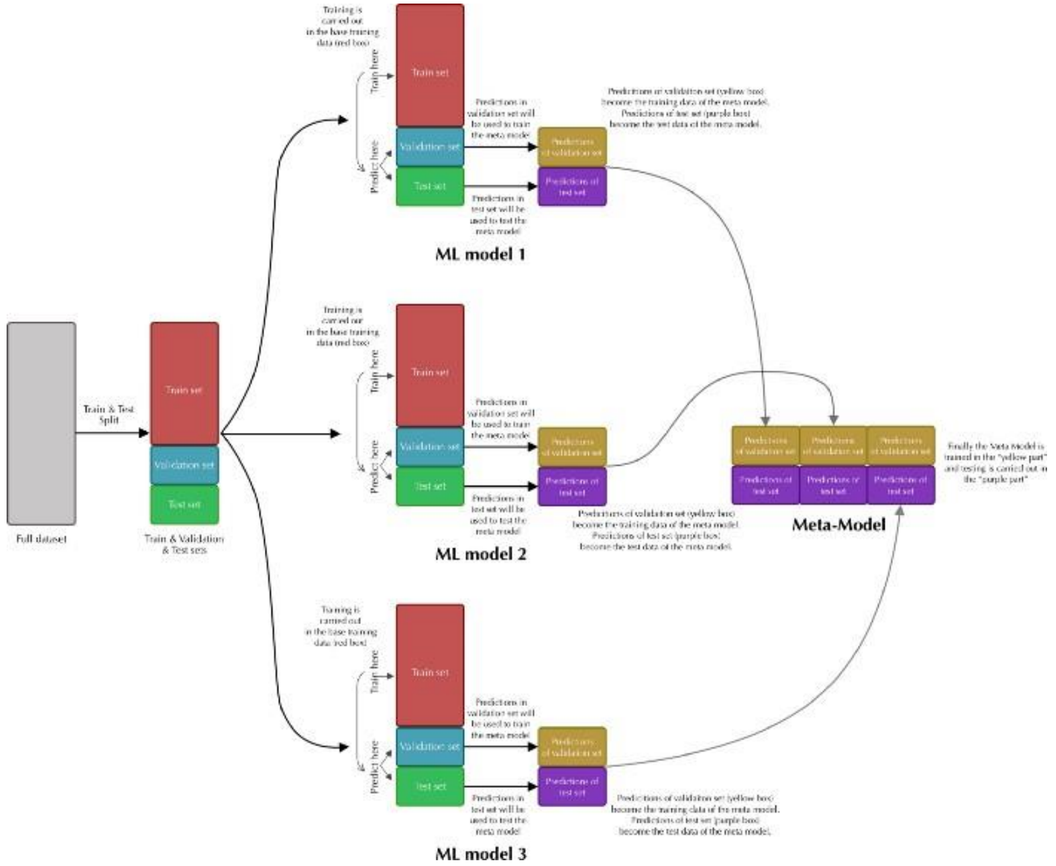
urban, total_wf, engnat, hand, gender, familysize drop

03 모델링

(1) 학습 방식

Auto ML & Blending





(2) Auto ML

1. Install & model comparison

```
[ ]  !pip install --pre pycaret
```

```
[ ]  #Data Setup
     from pycaret.classification import *
     clf1 = setup(data = train,
                  target = 'voted',
                  session_id = 1)
```

0	Session id	1
1	Target	voted
2	Target type	Binary
3	Target mapping	1: 0, 2: 1
4	Original data shape	(45532, 42)
5	Transformed data shape	(45532, 42)
6	Transformed train set shape	(31872, 42)
7	Transformed test set shape	(13660, 42)
8	Ordinal features	1
9	Numeric features	37
10	Categorical features	4
11	Preprocess	True
12	Imputation type	simple
13	Numeric imputation	mean
14	Categorical imputation	constant
15	Maximum one-hot encoding	5
16	Encoding method	None
17	Low variance threshold	0
18	Fold Generator	StratifiedKFold
19	Fold Number	10

```
[ ]  #Best Model Comparison
     best = compare_models(sort='AUC',fold=5)
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
gbc	Gradient Boosting Classifier	0.6941	0.7638	0.6543	0.7537	0.7005	0.3910	0.3951	11.0220
lightgbm	Light Gradient Boosting Machine	0.6925	0.7633	0.6480	0.7549	0.6973	0.3884	0.3932	0.9620
catboost	CatBoost Classifier	0.6888	0.7612	0.6560	0.7445	0.6974	0.3798	0.3830	13.6400
et	Extra Trees Classifier	0.6882	0.7590	0.6594	0.7417	0.6981	0.3780	0.3808	2.6440
ada	Ada Boost Classifier	0.6886	0.7555	0.6430	0.7516	0.6930	0.3810	0.3858	2.3120
rf	Random Forest Classifier	0.6846	0.7545	0.6371	0.7487	0.6883	0.3732	0.3783	6.0760
lda	Linear Discriminant Analysis	0.6690	0.7411	0.7180	0.6895	0.7035	0.3293	0.3297	0.2060
lr	Logistic Regression	0.6589	0.7283	0.7621	0.6651	0.7095	0.3009	0.3059	1.9900
qda	Quadratic Discriminant Analysis	0.4990	0.7133	0.1439	0.6455	0.1437	0.0715	0.0749	0.1520
nb	Naive Bayes	0.4944	0.6376	0.1420	0.1362	0.1390	0.0620	0.0620	0.1300
knn	K Neighbors Classifier	0.6018	0.6288	0.6499	0.6323	0.6409	0.1943	0.1944	6.2400
dt	Decision Tree Classifier	0.6130	0.6098	0.6443	0.6467	0.6455	0.2196	0.2196	0.8680
dummy	Dummy Classifier	0.5468	0.5000	1.0000	0.5468	0.7070	0.0000	0.0000	0.1160
svm	SVM - Linear Kernel	0.5453	0.0000	0.8766	0.5528	0.6718	0.0237	0.0280	1.0660
ridge	Ridge Classifier	0.6691	0.0000	0.7184	0.6894	0.7036	0.3294	0.3298	0.1220

(2) Auto ML

2. Creating & tuning

```
[ ]  #Creating Model
     gbc = create_model('gbc')
     cat = create_model('catboost')
     lgbm = create_model('lightgbm')
```

```
[ ]  #Hyperparameter tuning
     tuned_gbc=tune_model(gbc, search_library='optuna',n_iter=5)
     tuned_cat=tune_model(cat,search_library='optuna',n_iter=5)
     tuned_lgbm=tune_model(lgbm, search_library='optuna')
```

Gradient boosting

Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.7001	0.7749	0.6873	0.7446	0.7148	0.3997	0.4011
1	0.6857	0.7568	0.6764	0.7291	0.7018	0.3705	0.3717
2	0.6903	0.7626	0.6816	0.7333	0.7065	0.3796	0.3808
3	0.6878	0.7664	0.6902	0.7256	0.7074	0.3732	0.3737
4	0.6944	0.7616	0.6764	0.7420	0.7077	0.3889	0.3908
5	0.6771	0.7468	0.6546	0.7277	0.6892	0.3552	0.3574
6	0.6919	0.7648	0.6644	0.7447	0.7022	0.3851	0.3878
7	0.6890	0.7638	0.6563	0.7448	0.6978	0.3802	0.3834
8	0.6963	0.7711	0.6843	0.7404	0.7112	0.3919	0.3933
9	0.6994	0.7703	0.6780	0.7484	0.7114	0.3994	0.4015
Mean	0.6912	0.7639	0.6749	0.7381	0.7050	0.3824	0.3842
Std	0.0065	0.0076	0.0118	0.0079	0.0071	0.0131	0.0131

Catboost

Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.6998	0.7741	0.6368	0.7741	0.6988	0.4052	0.4129
1	0.6920	0.7567	0.6265	0.7674	0.6898	0.3901	0.3980
2	0.6966	0.7649	0.6242	0.7771	0.6923	0.4000	0.4094
3	0.6975	0.7668	0.6420	0.7670	0.6989	0.3998	0.4062
4	0.6869	0.7605	0.6139	0.7670	0.6820	0.3809	0.3901
5	0.6793	0.7463	0.6047	0.7599	0.6735	0.3663	0.3756
6	0.6900	0.7612	0.6271	0.7638	0.6887	0.3859	0.3933
7	0.6909	0.7583	0.6116	0.7758	0.6840	0.3898	0.4004
8	0.7069	0.7729	0.6320	0.7898	0.7022	0.4206	0.4308
9	0.6978	0.7674	0.6246	0.7788	0.6932	0.4025	0.4121
Mean	0.6938	0.7629	0.6243	0.7721	0.6903	0.3941	0.4029
Std	0.0073	0.0078	0.0109	0.0083	0.0083	0.0142	0.0143

Light GBM

Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.7083	0.7772	0.6569	0.7752	0.7112	0.4205	0.4264
1	0.6904	0.7597	0.6483	0.7513	0.6960	0.3841	0.3884
2	0.6922	0.7648	0.6437	0.7571	0.6958	0.3884	0.3936
3	0.6981	0.7705	0.6649	0.7541	0.7067	0.3983	0.4016
4	0.6909	0.7660	0.6368	0.7592	0.6927	0.3866	0.3927
5	0.6806	0.7499	0.6162	0.7547	0.6785	0.3675	0.3750
6	0.6931	0.7653	0.6403	0.7607	0.6953	0.3908	0.3967
7	0.6947	0.7627	0.6334	0.7677	0.6941	0.3949	0.4022
8	0.6997	0.7736	0.6538	0.7629	0.7042	0.4029	0.4079
9	0.7022	0.7721	0.6510	0.7688	0.7050	0.4086	0.4143
Mean	0.6950	0.7662	0.6445	0.7612	0.6979	0.3943	0.3999
Std	0.0072	0.0074	0.0131	0.0071	0.0088	0.0138	0.0136

(3) Blending

```
[ ]  #Blending Model('Soft Voting')
     blend = blend_models([tuned_gbc, tuned_cat, tuned_lgbm],method = 'soft',optimize='AUC')
```

Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.7061	0.7771	0.6512	0.7753	0.7078	0.4166	0.4230
1	0.6885	0.7590	0.6368	0.7551	0.6909	0.3815	0.3872
2	0.6925	0.7649	0.6408	0.7593	0.6951	0.3894	0.3951
3	0.6969	0.7701	0.6638	0.7528	0.7055	0.3958	0.3991
4	0.6922	0.7654	0.6368	0.7613	0.6935	0.3892	0.3955
5	0.6815	0.7484	0.6219	0.7528	0.6811	0.3687	0.3755
6	0.6894	0.7644	0.6403	0.7546	0.6927	0.3829	0.3882
7	0.6931	0.7621	0.6299	0.7673	0.6919	0.3921	0.3997
8	0.7029	0.7744	0.6515	0.7695	0.7056	0.4098	0.4156
9	0.6934	0.7715	0.6406	0.7607	0.6955	0.3914	0.3973
Mean	0.6936	0.7657	0.6414	0.7609	0.6960	0.3917	0.3976
Std	0.0067	0.0079	0.0112	0.0073	0.0078	0.0130	0.0129

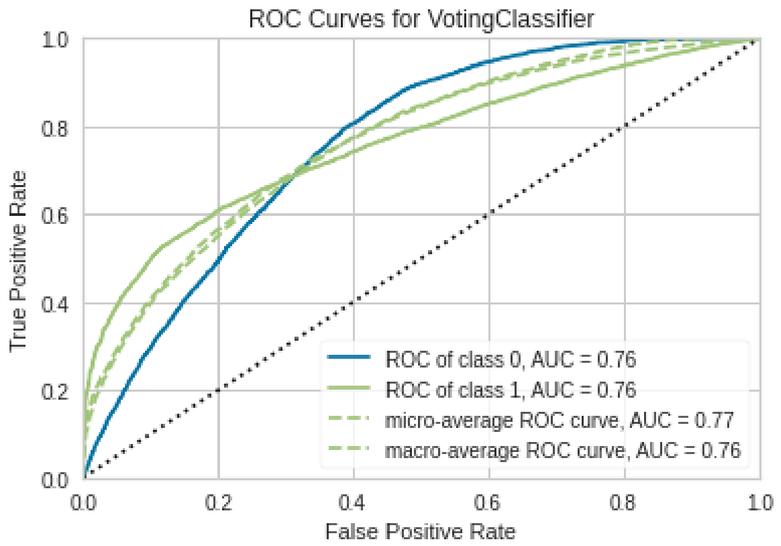
Processing: 100% 6/6 [03:05<00:00, 30.34s/it]

04 Conclusion

(1) Predict model

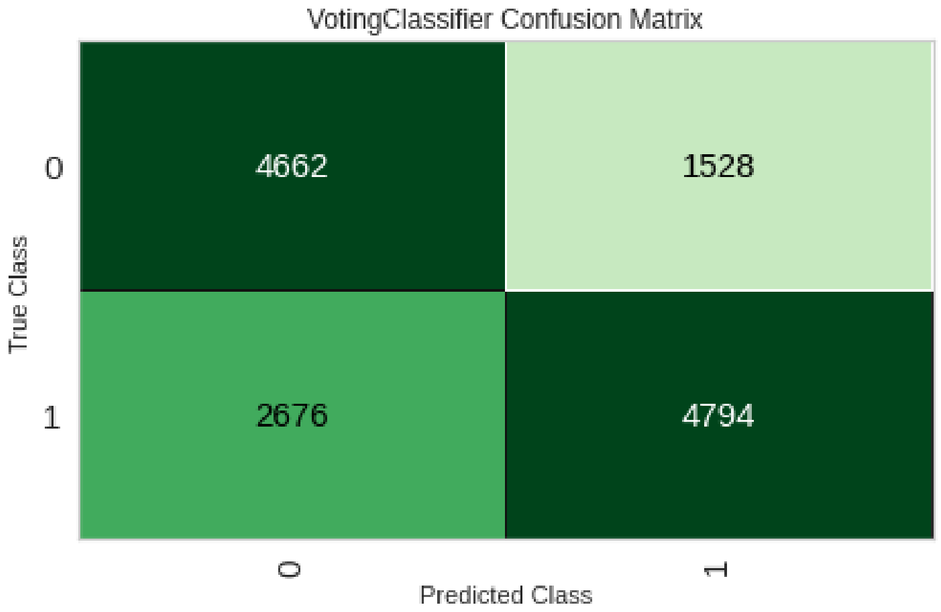
```
[ ]  #Prediction
     predict_model(blend)
```

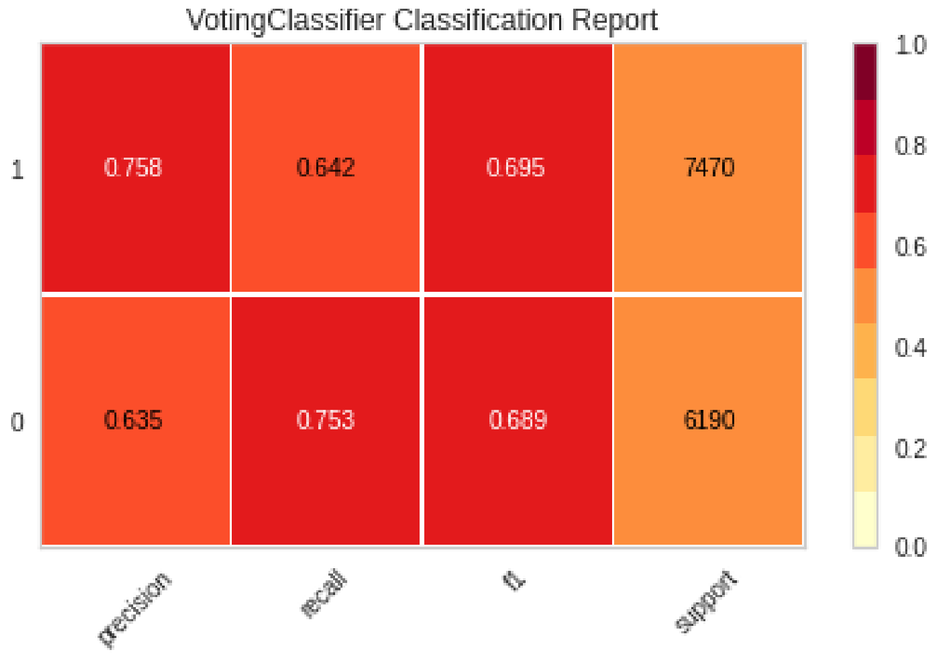
	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	Voting Classifier	0.6922	0.7611	0	0	0	0.3886	0.3941



(2) Model evaluation

Confusion matrix & class report





(3) 제출 결과

722263	submission1 (2).csv sample3 edit		2022-08-30 03:39:06	0.7037467918 0.704640265

데이콘 최종 제출 결과

Public score : 0.7037

Private score : 0.7046

(4) 소감 및 의의

데이터 전처리

전처리 과정은 한정된 데이터를 더 효율적으로 사용하기 위한 필수적인 과정이지만 과도한 전처리는 오히려 데이터가 가지고 있는 정보를 훼손할 수 있기에 데이터에 맞는 기법을 적절히 사용하여야 함

모델링 과정

단순히 하나의 과정에 국한되지 않고 다양한 알고리즘을 알아볼 수 있는 기회였고, 하나의 강한 알고리즘보다 앙상블을 통해 여러 알고리즘을 결합했을 때 더 좋은 결과를 얻을 수 있다는 것을 배움

감사합니다