





22-2 Machine Learning

# EDUCATION

## : Student Performance Dataset



 16기 김상옥 2021140655

 16기 노연수 2020150447

 16기 임정준 2019190035



# CONTENTS

INTRO

EDA

Data  
Preprocessing

Modeling

Statistical  
Reasoning

Conclusion



# Domain Knowledge

- ✓ '지능(IQ)', '거주지 주택 가격(부동산)', '조부모의 자산', '부모의 학력' 변수들이 성적에 대하여 95% 설명력을 가지고 있다고 알려져 있으며, 다른 국가에서도 이러한 변수 설명력이 크게 다르지 않음
- ✓ 지능은 보통 유전으로 결정되는 경우가 많으며, 부모의 부동산 및 학력은 학생이 노출될 수 있는 교육 환경과 주변 상황들을 내포하고 있음
- ✓ [Reference]: [학업성적 결정구조(중/고등학생의)].이해명.교육과학사.1988



# ABOUT

Student Performance Data was obtained in a survey of students' math course in secondary school:

- ✓ School ID
- ✓ Health
- ✓ Occupation & Education of Mother/Father
- ✓ Alcohol Consumption
- ✓ Family size & relationship
- ✓ G1 / G2 / G3 – tested on each three time



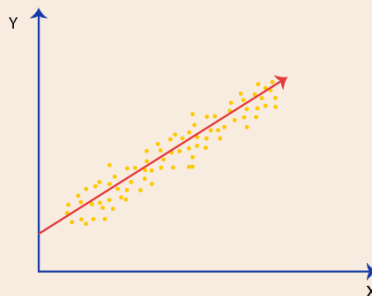
직관적으로 성적과 연관되어 있는 변수는 **아니지만**,  
다양한 Modeling을 통하여 **최적화된 Model Selection**

# MISSION

**Regression** about 'Grade':

주어진 Feature Data

바탕으로 학생들의 성적 예측





## EDA

[Raw Data]



## Imported Library

From dataprep.eda import create\_report

From seaborn import pairplot, heatmap, catplot, countplot

From matplotlib.pyplot import plot, show



### School

GP/MS – 두 학교 중 한 곳



### Sex

Female/Male



### Health

현재 건강 상태: from 1 to 5



### Address

도시 / 시골



### Family Size

Greater than 3 / Less or Equal to 3



### Pstatus

가족과 같이 거주 / 따로 거주



### Reason

집과 가까움 / 평판 / 수업 / 기타



### Guardian

Mother / Father / Other



### Internet

Internet 사용 유무 (yes or no)



### Romantic

현재 연애 상태 (yes or no)



### Family Relationship

관계의 질 수준(from 1 to 5)



### Family/School support

가족/학교의 교육적 지원 여부(yes or no)



### Traveltime

통학 시간(from 1 to 4)



### Activities/go out

방과후 학교(yes or no) / 친구들과 노는 정도(1-5)



### Nursery

보건 학교 출석 여부 (yes or no)



### Higher/Paid

고등 교육 희망 여부 / 사교육 여부 (yes or no)



## EDA [Detailed Features]



### Expected Prime Variable

- 부모의 직업과 교육 수준은 학생에게 있어 교육 환경 노출 정도를 결정할 것이다.
- 알코올 소비량은 궁극적으로 학업에 대한 집중력과 공부 시간에 영향이 있을 것이다.



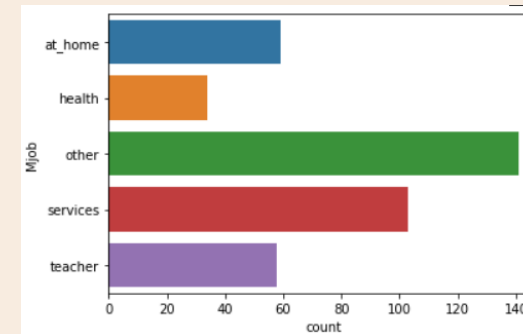
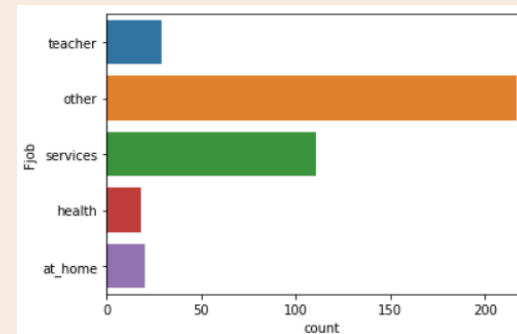
#### Mjob/Fjob

부모의 직업

(교사 / 보건 관련 직종 / 서비스업 / 기타)

기타 비율이 비교적 높으나,

상식적으로 성적에 유의하다고 판단

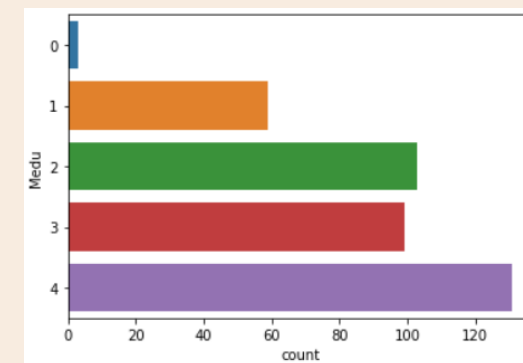
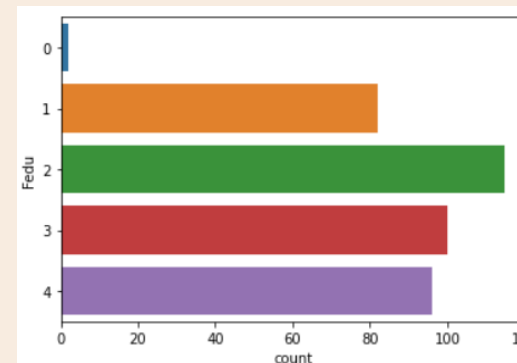


#### Medu/Fedu

부모의 학력 수준

0 (= None) ~ 4 (= higher than Secondary)

타겟 변수와의 **Correlation = 0.2**



#### Dalc/Walc/Studytime

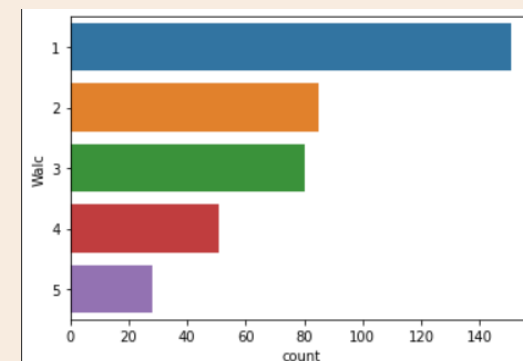
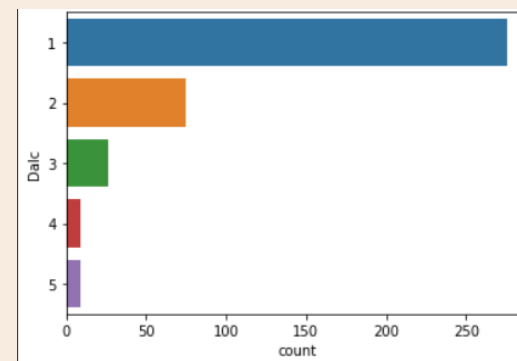
Dalc, Walc = 주중, 휴일 중 알코올 소비량

Studytime = 공부시간(from 1 to 4)

두 변수 간 상관관계 비교적 높음 (약 0.65)

Studytime과의 상관관계 = - 0.3

타겟 변수와의 **Correlation = 0.1**





## EDA [Detailed Features]



### Expected Prime Variable

- 누적되는 수업에 대한 낙제는 자신의 점수에 대한 부정적인 영향을 끼칠 것이다.
- 결석 횟수는 간접적으로 수업을 따라가는 데 있어서 부정적인 영향을 끼칠 것이다.

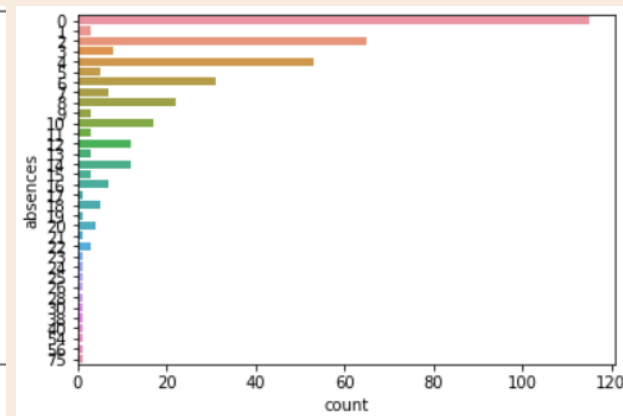
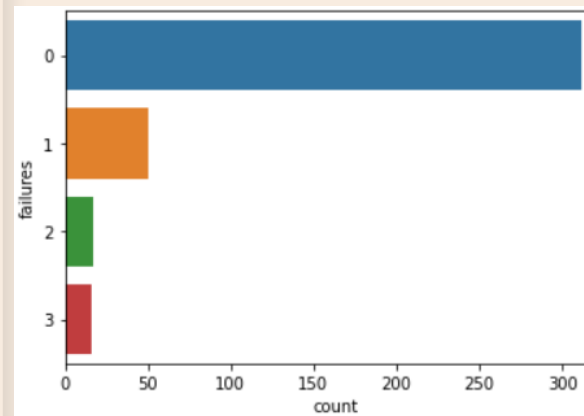


#### Failures

Class Failure(F 학점) 횟수

0 ~ 3 (4 이상은 3으로 처리)

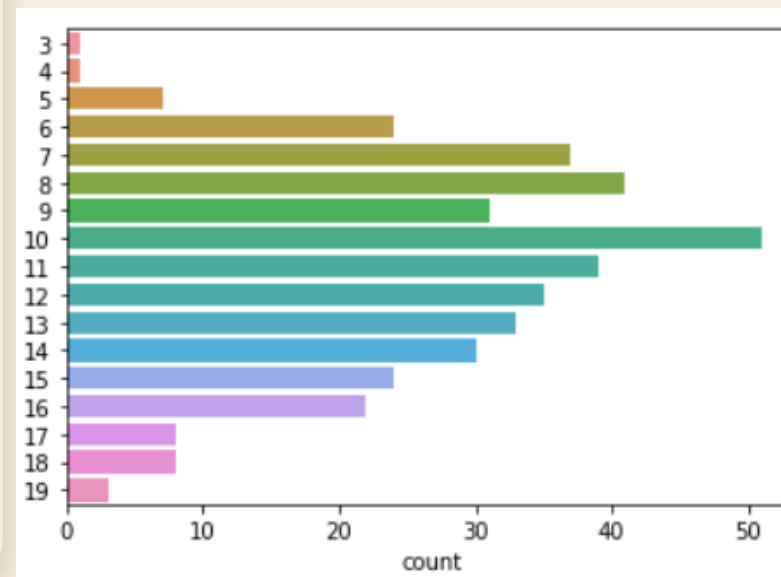
타겟 변수와의 **Correlation = - 0.35**



#### Absences

결석 횟수(Numerical Variable), 0 ~ 93

타겟 변수와의 **Correlation = -0.03**



#### G1/G2/G3 [Target Variable]

0 ~ 20, 정규분포와 유사

점수 간 상관관계 비교적 높음 (약 0.85)

**:  $G\_mean = (G1+G2+G3) / 3$**

# Assumption



## Data Preprocessing



### ✓ Overfitting Problem:

Dataset shape(395, 33)를 고려하면, Overfitting에 Sensitive

### ✓ Correlation Problem:

Feature와 Target 간 Correlation이 높지 않아 Non-linear Modeling이 필요

### ✓ Categorical Variable Encoding:

Overfitting이 쉽게 일어날 수 있고, Non-linear Model에 적합한 Encoding 필요

### ✓ Multicollinearity Problem:

부모 학력 및 직업, 알코올 소비량 간 높은 상관관계 문제 해결 필요



### Derived Variable / Dropping

```
#파생변수 생성
df['Overall alc']=(df['Dalc']+df['Walc'])/2
df['Parent Job/Edu']=(df['Medu']+df['Fedu'])/2
df=df.drop(['Dalc','Walc','Medu','Fedu'],axis=1)
```



### One-hot Encoding

```
#One-hot Encoding
df_1 = pd.get_dummies(df1)
```



### Log Transformation

```
skewed_list=['internet','romantic','higher','nursery','scho
for i in df2.columns:
    if i in skewed_list:
        df2[i]=np.log1p(df2[i]) #Log Transformation
```



### Standard Scaling

```
for i in scaled_cols:
    scaler = StandardScaler()
    A_n = scaler.fit_transform(train[i].values.reshape(-1,1)) #train: fit_transform()
    train[i]=A_n
    B_n = scaler.transform(test[i].values.reshape(-1,1)) #test: transform()
    test[i]=B_n
```



# PREPROCESSING DIAGRAM



## One-hot Encoding

Label Encoding / Mean Encoding / M-estimator Encoding

Overfitting 문제로 **One-hot encoding** 활용

## Derived Variable/ Column Drop

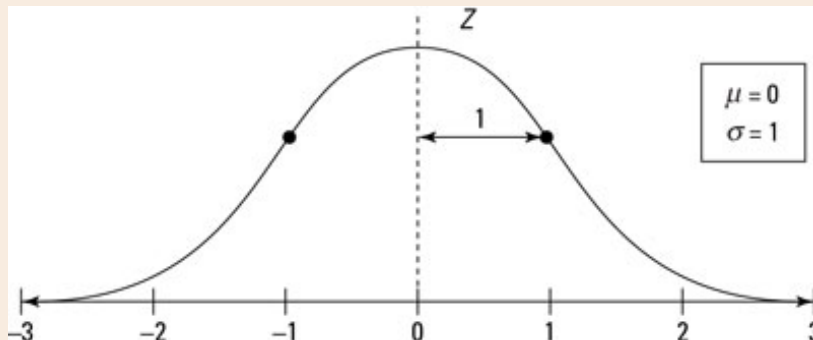
Multicollinearity Problem 해결 위해 변수 병합:

('Dalc' + 'Walc')/2, ('Medu' + 'Fedu')/2

'Pstatus'/'Address'/'School' **Column Drop**

## Standard Scaling

각 Categorical 변수마다 범주가 다르므로 표준화 진행



## New Derived Variable?

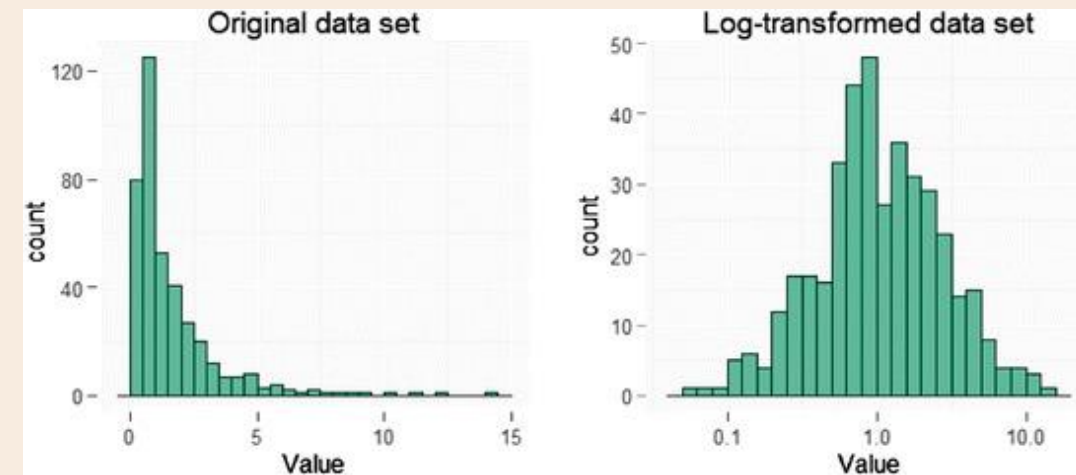
몇몇 파생변수는 모델 성능에 유의한 효과를 미치지 못함,  
과도한 Column Drop은 오히려 성능 저하 요인

## Log Transformation

Skewed Distribution에서

넓은 부분은 좁혀주고, 좁은 부분은 넓힘

→ **Skewed Dataset**을 **Normal Distribution**으로 변형





Data  
Preprocessing



# Preprocessing Summary



**Original Dataset**

- ✓ One-hot Encoding
- ✓ Derived Variable
- ✓ Log Transformation
- ✓ StandardScaler

**Unnecessary Column:**

- ✓ Pstatus
- ✓ School
- ✓ Address



Modeling



## AutoML: Pycaret

적은 코드로 Machine Learning Workflows를  
자동화하는 오픈소스 라이브러리

```
!pip install pycaret[full]  
from pycaret.utils import enable_colab  
enable_colab()  
from pycaret.regression import *
```

The Pycaret logo is displayed in a stylized font, with 'PYCARET' in blue and black. It is positioned within a large, circular graphic that has a blue and white gradient, resembling a CD or a stylized 'P'.



## Modeling



## Setup(dataset , target)

```
from pycaret.regression import *  
reg=setup(data=train, target = 'G_mean',session_id=14,silent=True) #Feature/Target setup
```

	Description	Value
0	session_id	14
1	Target	G_mean
2	Original Data	(276, 40)
3	Missing Values	False
4	Numeric Features	30
5	Categorical Features	9
	Ordinal Features	False
	High Cardinality Features	False
	High Cardinality Method	None
9	Transformed Train Set	(193, 38)
10	Transformed Test Set	(83, 38)
11	Shuffle Train-Test	True
12	Stratify Train-Test	False
13	Fold Generator	KFold



## Model Comparison



```
best = compare_models(sort = 'R2') #Model 간 Performance 비교
```

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
rf	Random Forest Regressor	2.6628	11.5052	3.3265	0.1456	0.3329	0.3603	0.449
br	Bayesian Ridge	2.6876	11.3955	3.3487	0.1423	0.3390	0.3690	0.016
lightgbm	Light Gradient Boosting Machine	2.7079	11.9202	3.4179	0.1103	0.3454	0.3734	0.029
catboost	CatBoost Regressor	2.7432	11.9588	3.4171	0.1077	0.3456	0.3814	1.191
ada	AdaBoost Regressor	2.7109	11.9848	3.4173	0.1072	0.3399	0.3667	0.084
ridge	Ridge Regression	2.7322	11.9417	3.4254	0.0975	0.3519	0.3698	0.015
lr	Linear Regression	2.7423	12.1815	3.4580	0.0769	0.3585	0.3713	0.016
gbr	Gradient Boosting Regressor	2.7963	12.8303	3.5044	0.0538	0.3526	0.3773	0.062
huber	Huber Regressor	2.7654	12.5885	3.5003	0.0451	0.3601	0.3759	0.041
omp	Orthogonal Matching Pursuit	2.8499	12.7984	3.5402	0.0395	0.3582	0.3935	0.015
xgboost	Extreme Gradient Boosting	2.8682	13.1970	3.5911	-0.0122	0.3590	0.3713	0.345
en	Elastic Net	2.9831	13.5419	3.6600	-0.0188	0.3709	0.4223	0.014
llar	Lasso Least Angle Regression	3.0706	14.2402	3.7539	-0.0711	0.3782	0.4329	0.015
dummy	Dummy Regressor	3.0706	14.2402	3.7539	-0.0711	0.3782	0.4329	0.011
lasso	Lasso Regression	3.0804	14.3152	3.7641	-0.0771	0.3793	0.4343	0.015
knn	K Neighbors Regressor	3.0266	14.4614	3.7626	-0.0963	0.3825	0.4261	0.062
et	Extra Trees Regressor	3.1757	16.0971	3.9572	-0.2230	0.3834	0.4181	0.408



## Modeling



## Model Selection

### Blending/Stacking

데이터셋이 작기 때문에 Generalization에서 Overfitting Problem 문제가 커지고, 성능 저하 문제 발생

### Ultimately, Single Model

Single Model의 Hyperparameter Tuning이 다른 Modeling보다 성능 좋음  
가장 우수한 **RandomForestRegressor** 선정  
→ Dataset의 특수성에 기인

```
predict_model(tuned_rf,data=test) #Test data
```

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	Random Forest Regressor	2.2504	8.5356	2.9216	0.2648	0.2943	0.282



## Tuning Hyperparameter

Overfitting Problem Solution

#Kfold = 10 #max\_depth Control #n\_estimators >=250

```
params = {"max_depth": [5,8,11],  
         "max_features": [1,2,3],  
         "n_estimators": [250, 500, 1000]}  
tuned_rf=tune_model(rf,optimize = 'R2', custom_grid = params) #Hyperparameter tuning
```

	MAE	MSE	RMSE	R2	RMSLE	MAPE
Fold						
0	2.6854	14.0389	3.7469	0.1430	0.4657	0.5599
1	2.7617	12.0331	3.4689	-0.0463	0.3513	0.3612
2	2.7946	11.4950	3.3904	-0.1558	0.2771	0.2326
3	2.8004	16.4480	4.0556	0.0397	0.4459	0.5051
4	3.4465	16.4593	4.0570	0.1777	0.4591	0.5890
5	2.6249	9.0307	3.0051	0.4250	0.3345	0.3774
6	2.8737	11.6953	3.4198	0.3360	0.4045	0.4836
7	2.2161	6.6321	2.5753	0.2985	0.2091	0.2037
8	2.8659	11.0383	3.3224	0.2910	0.4045	0.5115
9	2.9287	12.4982	3.5353	-0.1875	0.3319	0.3519
Mean	2.7998	12.1369	3.4577	0.1321	0.3684	0.4176
Std	0.2882	2.8774	0.4259	0.2020	0.0792	0.1263

# RandomForestRegressor

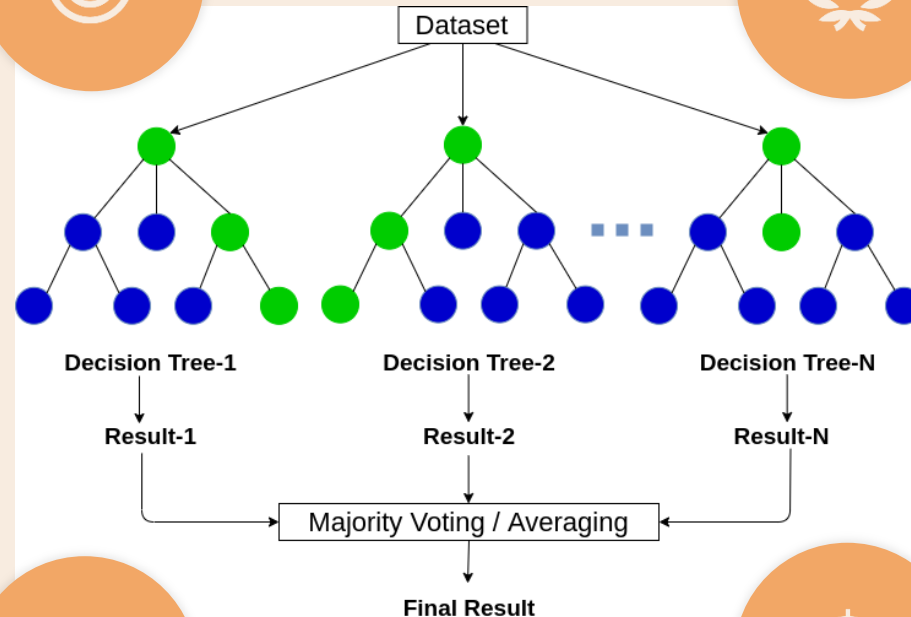


## Definition



여러 **Decision Tree**를 **Ensemble**,

평균/MSE를 사용해 예측 정확도 높이고  
Overfitting을 제어하는 **Meta Estimator**



## Hyperparameter



#**n\_estimators**: Tree의 개수

#**max\_depth**: Tree의 최대 깊이

#**min\_samples\_leaf**: 내부 Node를  
분할하는 데 필요한 최소 샘플 수

#**max\_features**: 최상의 분할을 찾는데  
고려하는 Feature 수

#**bootstrap**: 복원 추출 제어

## Advantages



- ✓ 대부분의 Dataset에서 평균 이상의  
Performance 발휘
- ✓ Feature Importance Estimating 가능

## Modeling



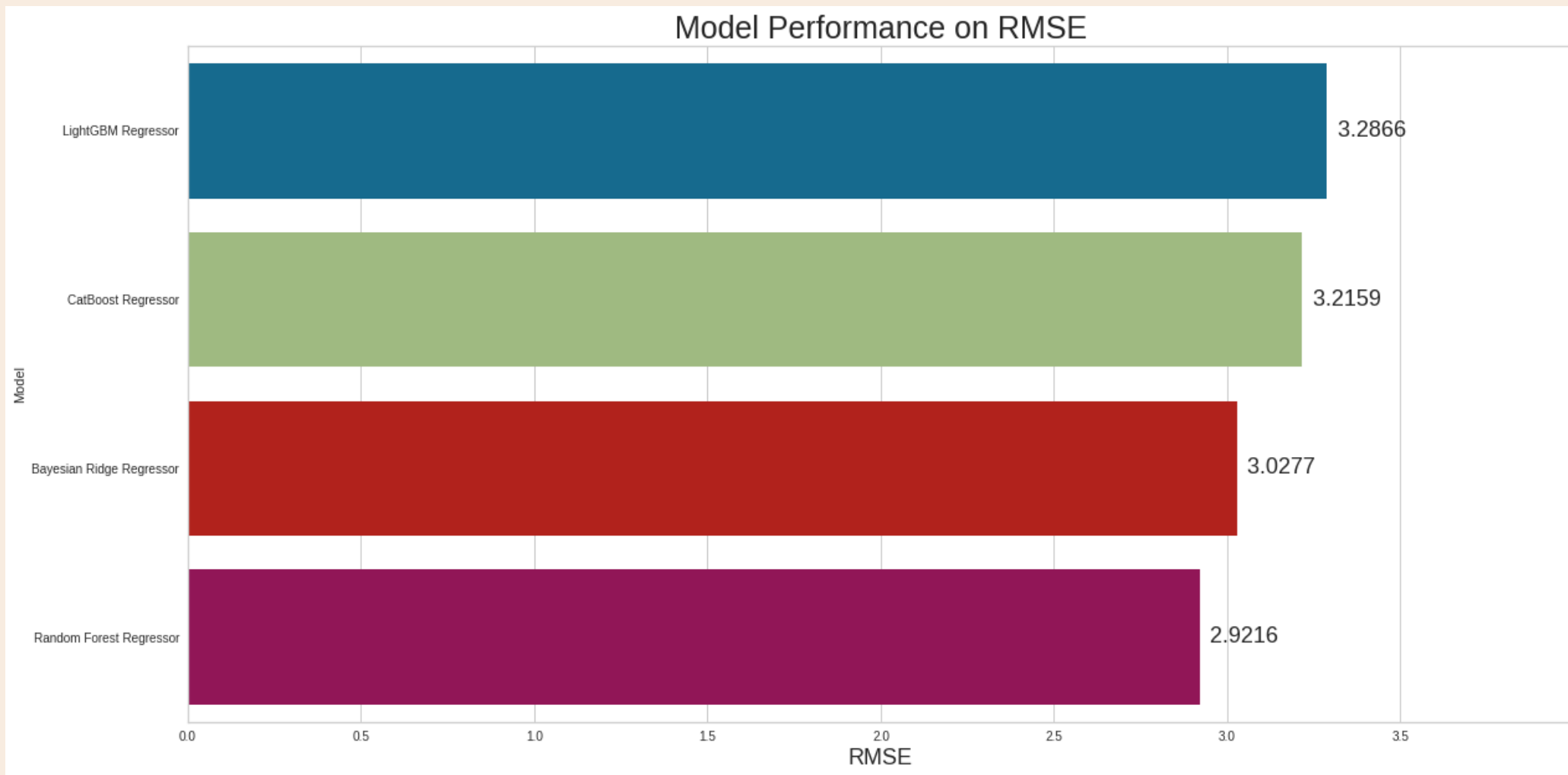
- ✓ Overfitting 조절 위해 Hyperparameter  
조절 (max\_depth, n\_estimators)
- ✓ RandomForest 활용, Feature Engineering  
전 Feature Importance 확인



Modeling



## Final Model Performance



# Statistical Reasoning

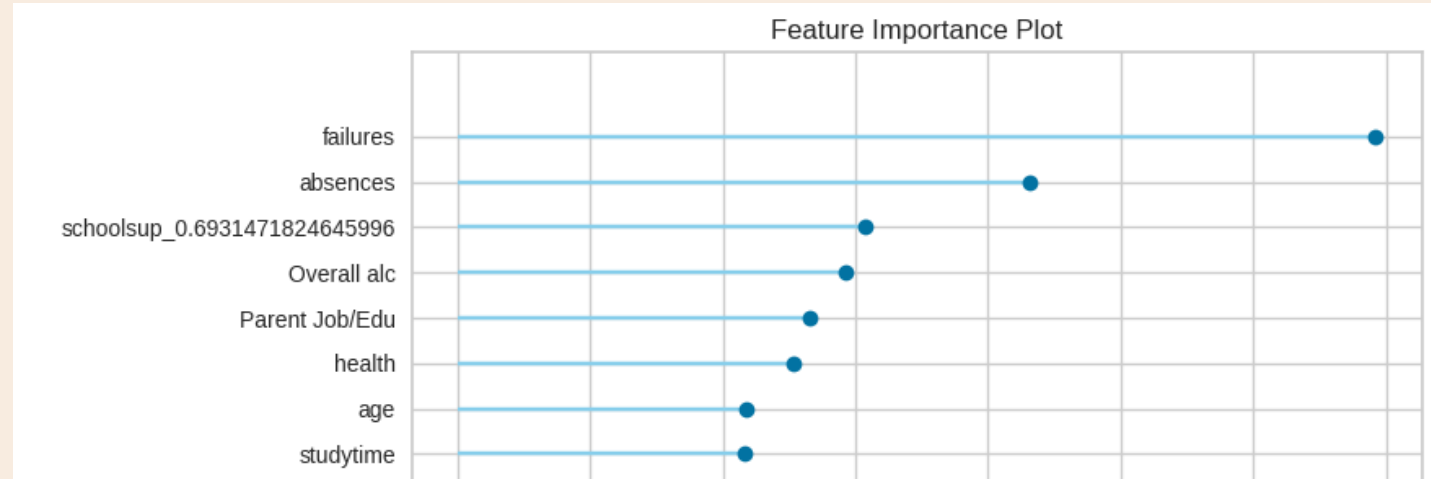


Statistical  
Reasoning



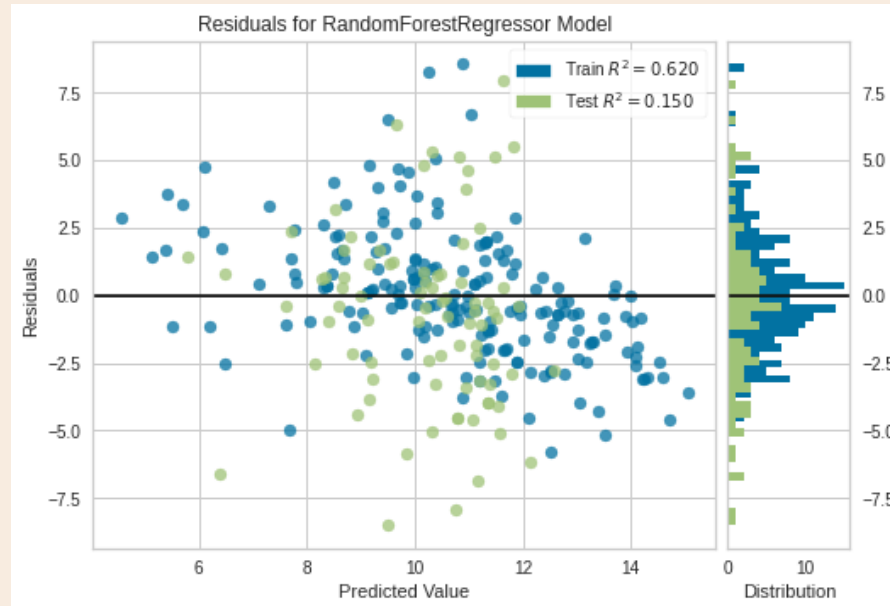
## Feature Importance Analysis

- ✓ Failures
  - ✓ Absences
  - ✓ School Support
  - ✓ Overall Alc
  - ✓ Parent Edu
  - ✓ Health
  - ✓ Age
  - ✓ Studytime
- ✓ 낙제를 덜 할수록
  - ✓ 결석을 덜 할수록
  - ✓ 학교 지원 있으면
  - ✓ 술을 덜 마시면
  - ✓ 부모 교육 수준 ↑
  - ✓ 더욱 건강하면
  - ✓ 나이가 많으면
  - ✓ 공부시간이 많으면



## Residuals Plot

- ✓ Residual Distribution은 대체로 Normal distribution Shape
- ✓ Regression의 Quality 대체로 준수
- ✓ Bias-Variance 간 Balance를 이루었다고 할 수 있음







Modeling'



# 번외: Classification



- ✓ Regression으로 정확히 성적을 예측하기에는 Dataset의 한계 존재
- ✓ 만점 20점 기준으로 A, B, C 구간 3등분, 타겟 변수를 범주화
- ✓ Logistic Regression, Naive Bayes, Random Forest, Support Vector Machine, Linear Kernel 단순 적용
- ✓ Naive Bayes 제외 60% 수준의 정확도



## Target Simplification

```
# 등급 분류 문제로 치환하는 대안(절대평가, ABC)
# 평균 20점 만점에 2/3 이상 A, 1/3 이상 B, 그 미만 C
grades = []
for row in df['G_mean'] :
    if row >= 20 * 2/3:
        grades.append('A')
    elif row >= 20 * 1/3:
        grades.append('B')
    else :
        grades.append('C')
y1 = pd.DataFrame(grades)
print("Grade: %n", y1.astype('category').value_counts())
y1.head()
```



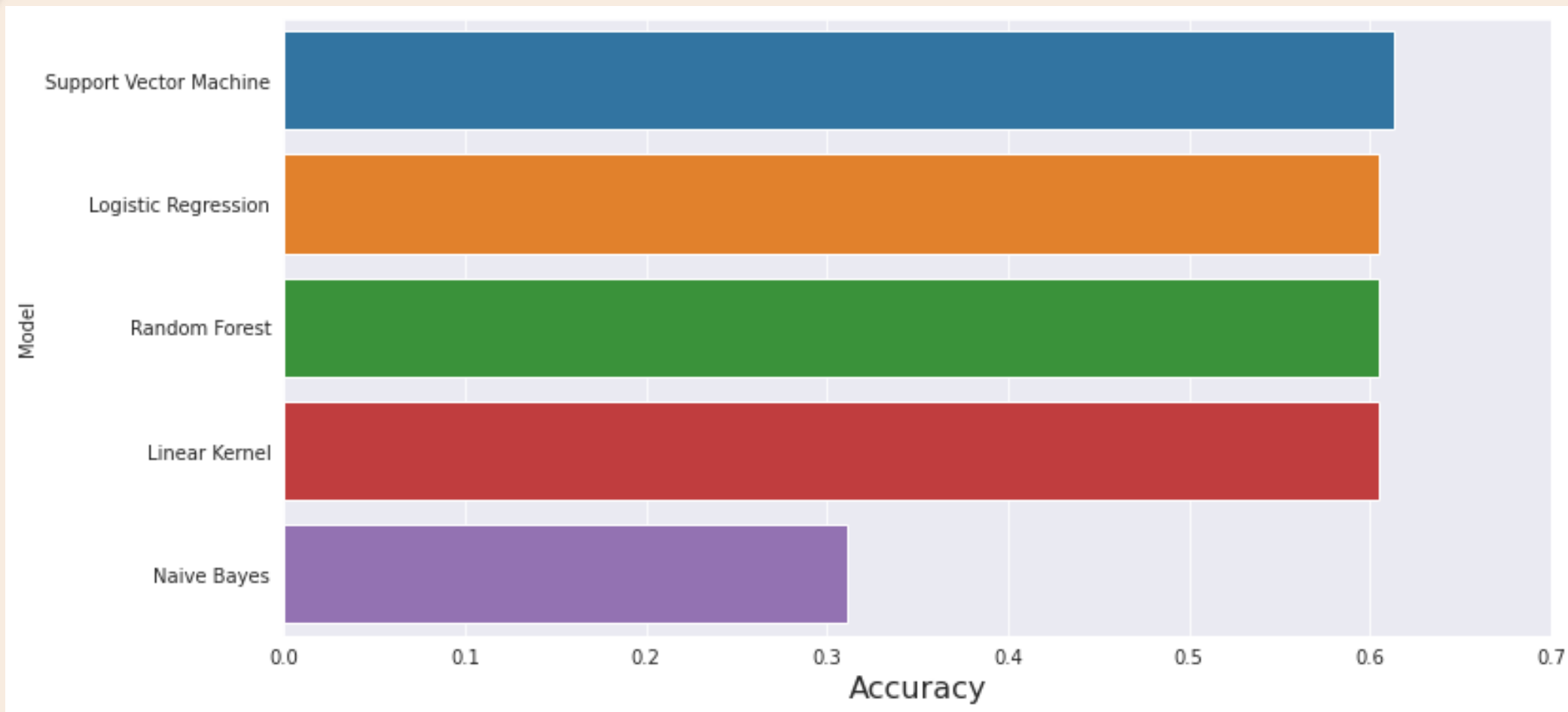
## Model Comparison

	Model	Accuracy
0	Logistic Regression	0.605042
1	Naive Bayes	0.310924
2	Random Forest	0.605042
3	Support Vector Machine	0.613445
4	Linear Kernel	0.605042

# 번외: Classification



## Classification Model Performance



Modeling'



# Conclusion



## #도메인 지식의 중요성

“순수한 공부가 성적을 올리는 절대적인 요소가 아니다!”  
어떠한 주변 환경과 요소들이 중요한지 알았다면 더 좋은 Feature Engineering이 가능했을 것

## #Beginner~Intermediate ML-er에게 유용한 AutoML

Models Comparison과 Model Visualization이 빠른 시간 내에 가능하다는 것은  
ML-er들에게 있어 Resource를 크게 덜어줄 수 있을 것

## #Data Preprocessing

Modeling보다 더 중요하게 느껴진 Data Preprocessing  
창의적인 EDA를 바탕으로 보다 좋은 Modeling을 이끌어내는 것이 진정한 Data Engineer

## #Basic Statistics

Statistics을 기반으로 DataSet을 바라보는 Insight Level 'UP'



# THANKS

16기 김상욱  
16기 노연수  
16기 임정준

