

머신러닝

학생 결석 여부 확인 데이터

/kaggle/input/adp-p8/problem1.csv

성별(sex) 바이너리 : 'F' - 여성 또는 'M' - 남성

나이(age) 숫자: 15 - 22

부모님동거여부 (Pstatus) 바이너리: T: 동거 또는 'A': 별거

엄마학력(Medu) 숫자 : 0 : 없음, 1 : 초등 교육, 2 : 5-9학년, 3 - 중등 교육 또는 4 - 고등 교육

아빠학력(Fedu) 숫자 : 0 : 없음, 1 : 초등 교육, 2 : 5-9학년, 3 - 중등 교육 또는 4 - 고등 교육

주보호자(guardian) 명목형 : '어머니', '아버지' 또는 '기타'

등하교시간(travelttime) 숫자 : 1 : 15분이하, 2 : 15 ~ 30분, 3 : 30분 ~ 1시간, 4 : 1시간 이상

학습시간(studytime) 숫자 : 1 : 2시간이하, 2 : 2~5시간, 3 : 5~10시간, 4 : 10시간이상

학교횃수(failures) 숫자 : 1, 2, 3 else 4

자유시간(freetime) 숫자 : 1(매우 낮음), 2, 3, 4, 5(매우 높음)

가족관계(famrel) 숫자 : 1(매우 나쁨), 2, 3, 4, 5(우수)

1-1 데이터 EDA 및 시각화

In [1]:

```
import pandas as pd
```

In [3]:

```
df= pd.read_csv('data/problem1.csv')
```

In [4]:

```
df.head()
```

Out[4]:

	sex	age	Pstatus	Fedu	Medu	guardian	studytime	travelttime	failures	famrel	freetime
0	F	18.0	A	4	4	mother	2	2.0	0	4	3.0
1	F	17.0	T	1	1	father	2	1.0	0	5	3.0
2	F	15.0	T	1	1	mother	2	1.0	3	4	3.0
3	F	15.0	T	2	4	mother	3	1.0	0	3	2.0
4	F	NaN	T	3	3	father	2	1.0	0	4	3.0

In [9]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 395 entries, 0 to 394
Data columns (total 12 columns):
 #   Column      Non-Null Count  Dtype
---  ---
 0   sex         395 non-null    object
 1   age         392 non-null    float64
 2   Pstatus     395 non-null    object
 3   Fedu        395 non-null    int64
 4   Medu        395 non-null    int64
 5   guardian    395 non-null    object
 6   studytime   395 non-null    int64
 7   traveltime  392 non-null    float64
 8   failures    395 non-null    int64
 9   famrel      395 non-null    int64
10  freetime    390 non-null    float64
11  absences    395 non-null    int64
dtypes: float64(3), int64(6), object(3)
memory usage: 37.2+ KB
```

In [11]:

```
df.isnull().sum()
```

Out[11]:

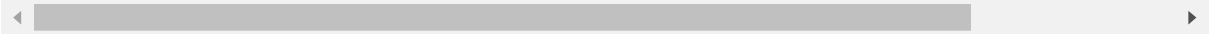
```
sex         0
age         3
Pstatus     0
Fedu        0
Medu        0
guardian    0
studytime   0
traveltime  3
failures    0
famrel      0
freetime    5
absences    0
dtype: int64
```

In [12]:

```
df.describe()
```

Out[12]:

	age	Fedu	Medu	studytime	travelttime	failures	famrel	
count	392.000000	395.000000	395.000000	395.000000	392.000000	395.000000	395.000000	39
mean	16.698980	2.521519	2.749367	2.035443	1.446429	0.334177	3.944304	
std	1.279865	1.088201	1.094735	0.839240	0.695022	0.743651	0.896659	
min	15.000000	0.000000	0.000000	1.000000	1.000000	0.000000	1.000000	
25%	16.000000	2.000000	2.000000	1.000000	1.000000	0.000000	4.000000	
50%	17.000000	2.000000	3.000000	2.000000	1.000000	0.000000	4.000000	
75%	18.000000	3.000000	4.000000	2.000000	2.000000	0.000000	5.000000	
max	22.000000	4.000000	4.000000	4.000000	4.000000	3.000000	5.000000	



In [17]:

```
import warnings
warnings.simplefilter(action = 'ignore', category = FutureWarning)
warnings.simplefilter(action = 'ignore', category = UserWarning)

import seaborn as sns
import matplotlib.pyplot as plt

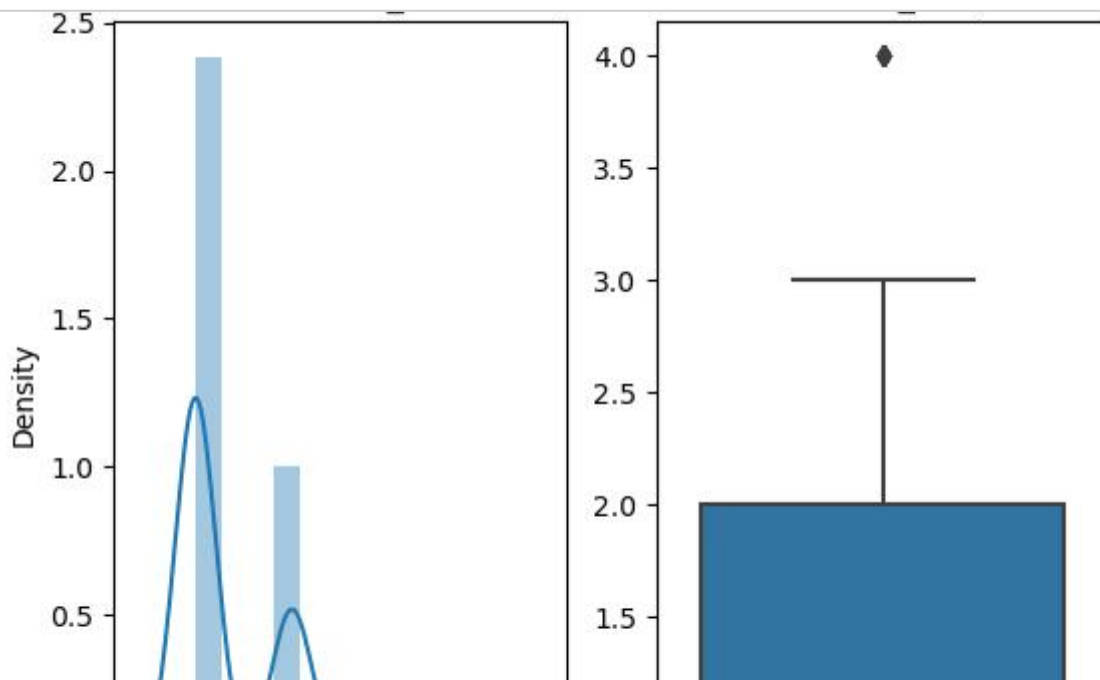
def edaplot(df):
    for v in df.select_dtypes(exclude = 'object'):
        fig = plt.figure()
        ax1 = fig.add_subplot(1,2,1)
        ax2 = fig.add_subplot(1,2,2)
        target = df[v].dropna()

        sns.distplot(target, ax = ax1)
        sns.boxplot(target, ax = ax2)

        ax1.set_title(v+"_dist")
        ax2.set_title(v+"_boxplot")

    plt.show()
```

edaplot(df)

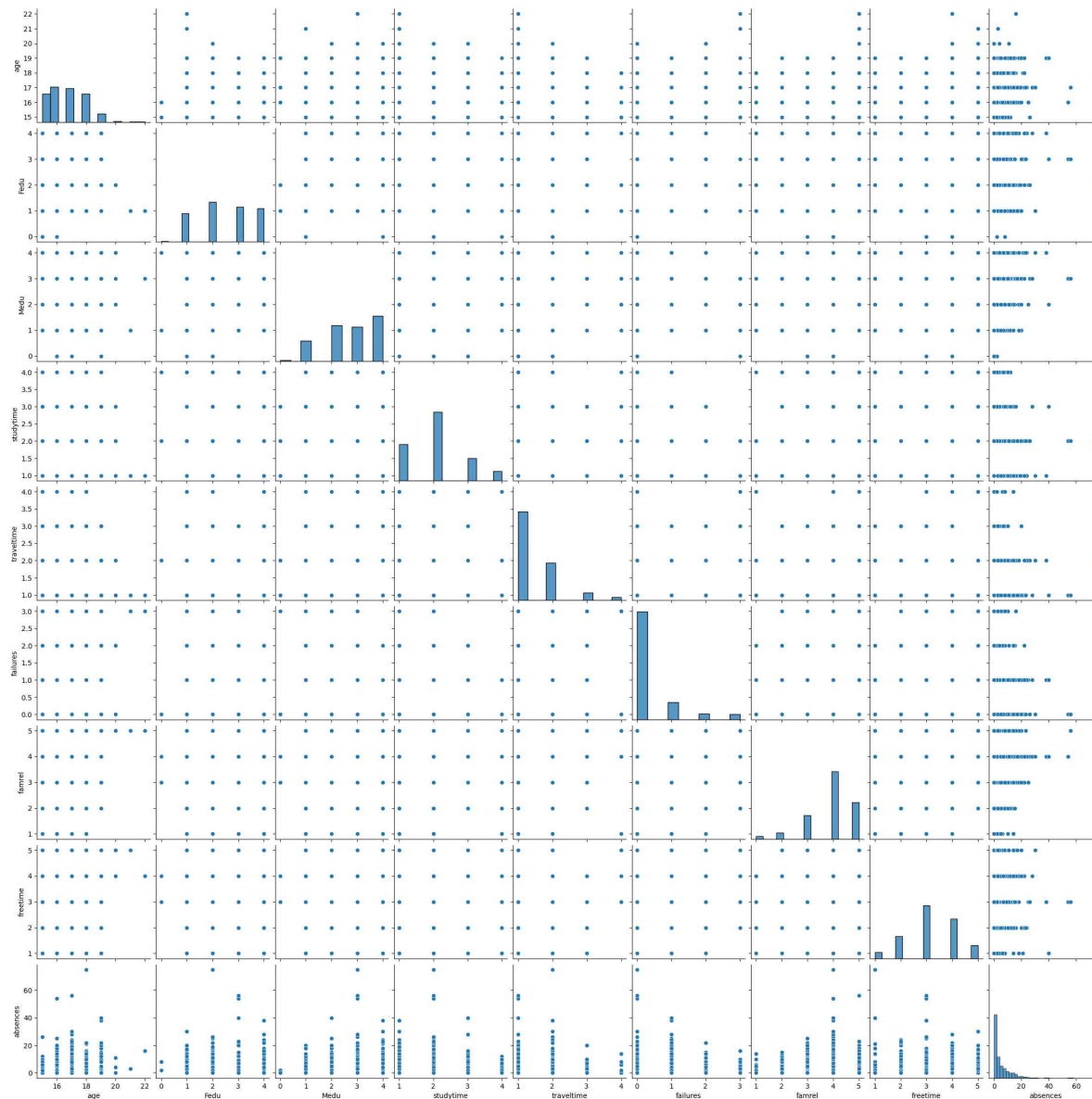


In [8]:

sns.pairplot(df)

Out[8]:

<seaborn.axisgrid.PairGrid at 0x114318fcf40>



1-2 결측치 처리 및 변화 시각화, 추가 전처리가 필요하다면 이유와 기대효과를 설명하

라

In [18]:

```
df.loc[df.age.isnull()]
```

Out[18]:

	sex	age	Pstatus	Fedu	Medu	guardian	studytime	traveltime	failures	famrel	freetime
4	F	NaN	T	3	3	father	2	1.0	0	4	3.0
96	M	NaN	T	3	4	mother	1	2.0	0	3	3.0
321	F	NaN	T	2	2	mother	2	1.0	0	4	2.0

In [19]:

```
df.loc[df.traveltime.isnull()]
```

Out[19]:

	sex	age	Pstatus	Fedu	Medu	guardian	studytime	traveltime	failures	famrel	freetime
66	M	15.0	A	4	4	mother	4	NaN	0	1	3.0
284	F	17.0	T	2	2	mother	2	NaN	0	5	4.0
360	F	18.0	A	4	1	mother	2	NaN	0	4	3.0

In [20]:

```
df.loc[df.freetime.isnull()]
```

Out[20]:

	sex	age	Pstatus	Fedu	Medu	guardian	studytime	traveltime	failures	famrel	freetime
31	M	15.0	T	4	4	mother	2	2.0	0	4	NaN
33	M	15.0	T	3	3	mother	2	1.0	0	5	NaN
97	F	16.0	T	1	2	mother	2	1.0	0	4	NaN
153	M	19.0	T	2	3	mother	1	1.0	3	4	NaN
316	F	18.0	T	1	2	mother	2	2.0	0	5	NaN

In [21]:

```
df_na = df.dropna()
```

In [23]:

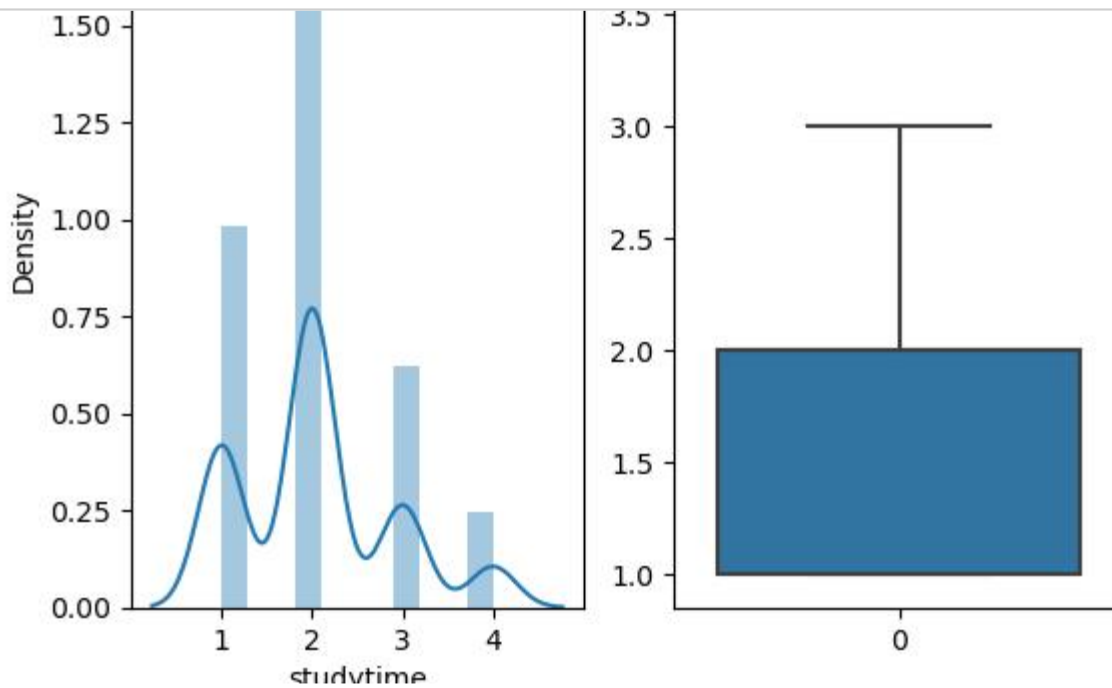
```
df_na.isnull().sum()
```

Out[23]:

```
sex          0
age          0
Pstatus      0
Fedu         0
Medu         0
guardian     0
studytime    0
travelttime  0
failures     0
famrel       0
freetime     0
absences     0
dtype: int64
```

In [24]:

```
edaplot(df_na)
```



In [27]:

```
df.describe().T
```

Out[27]:

	count	mean	std	min	25%	50%	75%	max
age	392.0	16.698980	1.279865	15.0	16.0	17.0	18.0	22.0
Fedu	395.0	2.521519	1.088201	0.0	2.0	2.0	3.0	4.0
Medu	395.0	2.749367	1.094735	0.0	2.0	3.0	4.0	4.0
studytime	395.0	2.035443	0.839240	1.0	1.0	2.0	2.0	4.0
traveltime	392.0	1.446429	0.695022	1.0	1.0	1.0	2.0	4.0
failures	395.0	0.334177	0.743651	0.0	0.0	0.0	0.0	3.0
famrel	395.0	3.944304	0.896659	1.0	4.0	4.0	5.0	5.0
freetime	390.0	3.233333	1.000985	1.0	3.0	3.0	4.0	5.0
absences	395.0	5.708861	8.003096	0.0	0.0	4.0	8.0	75.0

In [28]:

```
df_na.describe().T
```

Out[28]:

	count	mean	std	min	25%	50%	75%	max
age	384.0	16.700521	1.274944	15.0	16.0	17.0	18.0	22.0
Fedu	384.0	2.518229	1.088598	0.0	2.0	2.0	3.0	4.0
Medu	384.0	2.750000	1.098302	0.0	2.0	3.0	4.0	4.0
studytime	384.0	2.036458	0.841932	1.0	1.0	2.0	2.0	4.0
traveltime	384.0	1.447917	0.698671	1.0	1.0	1.0	2.0	4.0
failures	384.0	0.335938	0.739882	0.0	0.0	0.0	0.0	3.0
famrel	384.0	3.945312	0.890700	1.0	4.0	4.0	5.0	5.0
freetime	384.0	3.236979	1.005773	1.0	3.0	3.0	4.0	5.0
absences	384.0	5.799479	8.078201	0.0	0.0	4.0	8.0	75.0

In [32]:

```
#이상치 제거
def IQR_rule(df):
    Q1 = df.quantile(0.25)
    Q3 = df.quantile(0.75)
    IQR = Q3 - Q1

    not_outlier_condition = (Q3 + 1.5*IQR > df) & (Q1 - 1.5*IQR < df)
    return not_outlier_condition

col_list = ['absences']
conditions = df_na[col_list].apply(IQR_rule)
total_condition = conditions.sum(axis=1) == len(col_list)
df_new = df_na.loc[total_condition, :]
```

In [33]:

```
df_new.describe().T
```

Out[33]:

	count	mean	std	min	25%	50%	75%	max
age	365.0	16.663014	1.268328	15.0	16.0	17.0	18.0	22.0
Fedu	365.0	2.509589	1.093458	0.0	2.0	2.0	3.0	4.0
Medu	365.0	2.734247	1.105955	0.0	2.0	3.0	4.0	4.0
studytime	365.0	2.046575	0.851966	1.0	1.0	2.0	2.0	4.0
traveltime	365.0	1.438356	0.699024	1.0	1.0	1.0	2.0	4.0
failures	365.0	0.326027	0.745320	0.0	0.0	0.0	0.0	3.0
famrel	365.0	3.947945	0.901785	1.0	4.0	4.0	5.0	5.0
freetime	365.0	3.257534	0.988661	1.0	3.0	3.0	4.0	5.0
absences	365.0	4.495890	4.754190	0.0	0.0	3.0	7.0	19.0

In [34]:

```
#가변수 처리
df_new = pd.get_dummies(df_new)
```

In [35]:

```
df_new.head()
```

Out[35]:

	age	Fedu	Medu	studytime	travelttime	failures	famrel	freetime	absences	sex_F	sex_M
0	18.0	4	4	2	2.0	0	4	3.0	6	1	0
1	17.0	1	1	2	1.0	0	5	3.0	4	1	0
2	15.0	1	1	2	1.0	3	4	3.0	10	1	0
3	15.0	2	4	3	1.0	0	3	2.0	2	1	0
5	16.0	3	4	2	1.0	0	5	4.0	10	0	1

In [36]:

```
df_new.corr()['absences']
```

Out[36]:

```

age          0.155815
Fedu         0.000932
Medu         0.101418
studytime    -0.069475
travelttime  -0.067243
failures     0.082950
famrel       -0.092645
freetime     0.073871
absences     1.000000
sex_F        -0.034902
sex_M        0.034902
Pstatus_A    0.135764
Pstatus_T   -0.135764
guardian_father -0.105834
guardian_mother 0.044243
guardian_other 0.093951
Name: absences, dtype: float64

```

'absences'와 다른 변수들 간 상관계수가 전체적으로 낮아 예측 성능이 좋지 않을 것으로 생각됨.

1-3 결석일수 예측모델을 2개 제시하고 선택한 근거 설명

1. 선형 회귀 : 해석력이 좋기 때문에 추후 모델을 발전 시키는 데 도움
2. 랜덤 포레스트 : 강력한 일반화 성능

1-4 선정한 모델 2가지 생성 및 모델의 평가 기준을 선정하고 선정 이유 설명

0) 데이터 전처리 및 훈련 / 테스트 데이터 분리

In [37]:

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
df_new_sc = pd.DataFrame(sc.fit_transform(df_new), columns = df_new.columns)
```

In [38]:

```
X = df_new_sc.drop(['absences'], axis = 1)
y = df_new_sc.absences
```

In [39]:

```
from sklearn.model_selection import train_test_split
```

In [40]:

```
Xtrain, Xtest, Ytrain, Ytest = train_test_split(X, y, test_size = 0.3, random_state = 42)
```

1) 선형 회귀

In [41]:

```
import statsmodels.api as sm
```

In [42]:

```
linearmodel = sm.OLS(Ytrain, Xtrain)
lmdlfit = linearmodel.fit()
```

In [43]:

```
lmdlfit.summary()
```

Out[43]:

OLS Regression Results

Dep. Variable:	absences	R-squared (uncentered):	0.073
Model:	OLS	Adj. R-squared (uncentered):	0.027
Method:	Least Squares	F-statistic:	1.590
Date:	Sun, 20 Nov 2022	Prob (F-statistic):	0.0951
Time:	15:42:40	Log-Likelihood:	-352.46
No. Observations:	255	AIC:	728.9
Df Residuals:	243	BIC:	771.4
Df Model:	12		
Covariance Type:	nonrobust		

	coef	std err	t	P> t	[0.025	0.975]
age	0.1388	0.066	2.096	0.037	0.008	0.269
Fedu	-0.1144	0.082	-1.391	0.166	-0.276	0.048
Medu	0.1653	0.082	2.004	0.046	0.003	0.328
studytime	-0.0538	0.066	-0.821	0.413	-0.183	0.075
traveltime	-0.0135	0.067	-0.200	0.842	-0.146	0.119
failures	0.0013	0.071	0.019	0.985	-0.138	0.141
famrel	-0.1216	0.070	-1.748	0.082	-0.259	0.015
freetime	0.0298	0.065	0.461	0.645	-0.097	0.157
sex_F	-0.0024	0.034	-0.072	0.942	-0.069	0.064
sex_M	0.0024	0.034	0.072	0.942	-0.064	0.069
Pstatus_A	0.0683	0.033	2.058	0.041	0.003	0.134
Pstatus_T	-0.0683	0.033	-2.058	0.041	-0.134	-0.003
guardian_father	-0.0418	0.039	-1.066	0.287	-0.119	0.035
guardian_mother	0.0179	0.034	0.522	0.602	-0.050	0.085
guardian_other	0.0364	0.060	0.608	0.544	-0.081	0.154
Omnibus:	29.177	Durbin-Watson:	1.805			
Prob(Omnibus):	0.000	Jarque-Bera (JB):	36.061			
Skew:	0.913	Prob(JB):	1.48e-08			
Kurtosis:	3.251	Cond. No.	1.48e+16			

Notes:

[1] R² is computed without centering (uncentered) since the model does not contain a constant.

[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[3] The smallest eigenvalue is 2.74e-30. This might indicate that there are strong multicollinearity problems or that the design matrix is singular.

설명 계수가 0.05 밑인 변수 활용해서 다시 피팅 수행

In [44]:

```
linearmodel2 = sm.OLS(Ytrain, Xtrain[['age', 'Medu', 'Pstatus_A', 'Pstatus_T']])
Imdlfit2 = linearmodel2.fit()
```

In [45]:

```
Imdlfit2.summary()
```

Out[45]:

OLS Regression Results

Dep. Variable:	absences		R-squared (uncentered):			0.045
Model:	OLS		Adj. R-squared (uncentered):			0.034
Method:	Least Squares			F-statistic:		3.958
Date:	Sun, 20 Nov 2022			Prob (F-statistic):		0.00878
Time:	15:45:11			Log-Likelihood:		-356.22
No. Observations:	255			AIC:		718.4
Df Residuals:	252			BIC:		729.1
Df Model:	3					
Covariance Type:	nonrobust					
	coef	std err	t	P> t	[0.025	0.975]
age	0.1519	0.063	2.411	0.017	0.028	0.276
Medu	0.0852	0.062	1.370	0.172	-0.037	0.208
Pstatus_A	0.0688	0.032	2.126	0.035	0.005	0.133
Pstatus_T	-0.0688	0.032	-2.126	0.035	-0.133	-0.005
Omnibus:	32.036	Durbin-Watson:		1.870		
Prob(Omnibus):	0.000	Jarque-Bera (JB):		40.599		
Skew:	0.967	Prob(JB):		1.53e-09		
Kurtosis:	3.291	Cond. No.		4.43e+15		

Notes:

- [1] R^2 is computed without centering (uncentered) since the model does not contain a constant.
- [2] Standard Errors assume that the covariance matrix of the errors is correctly specified.
- [3] The smallest eigenvalue is 2.46e-29. This might indicate that there are strong multicollinearity problems or that the design matrix is singular.

In [48]:

```
predYtestLR = Imdlfit2.predict(Xtest[['age', 'Medu', 'Pstatus_A', 'Pstatus_T']])
```

In [49]:

```
from sklearn.metrics import r2_score
```

In [50]:

```
r2_score(Ytest, predYtestLR)
```

Out[50]:

0.08096654734642161

2)랜덤 포레스트 회귀

In [51]:

```
from sklearn.ensemble import RandomForestRegressor
```

In [52]:

```
rf = RandomForestRegressor(random_state = 42)
```

In [53]:

```
rf.fit(Xtrain, Ytrain)
```

Out[53]:

RandomForestRegressor(random_state=42)

In [54]:

```
predYtest = rf.predict(Xtest)
```

In [55]:

```
r2_score(Ytest, predYtest)
```

Out[55]:

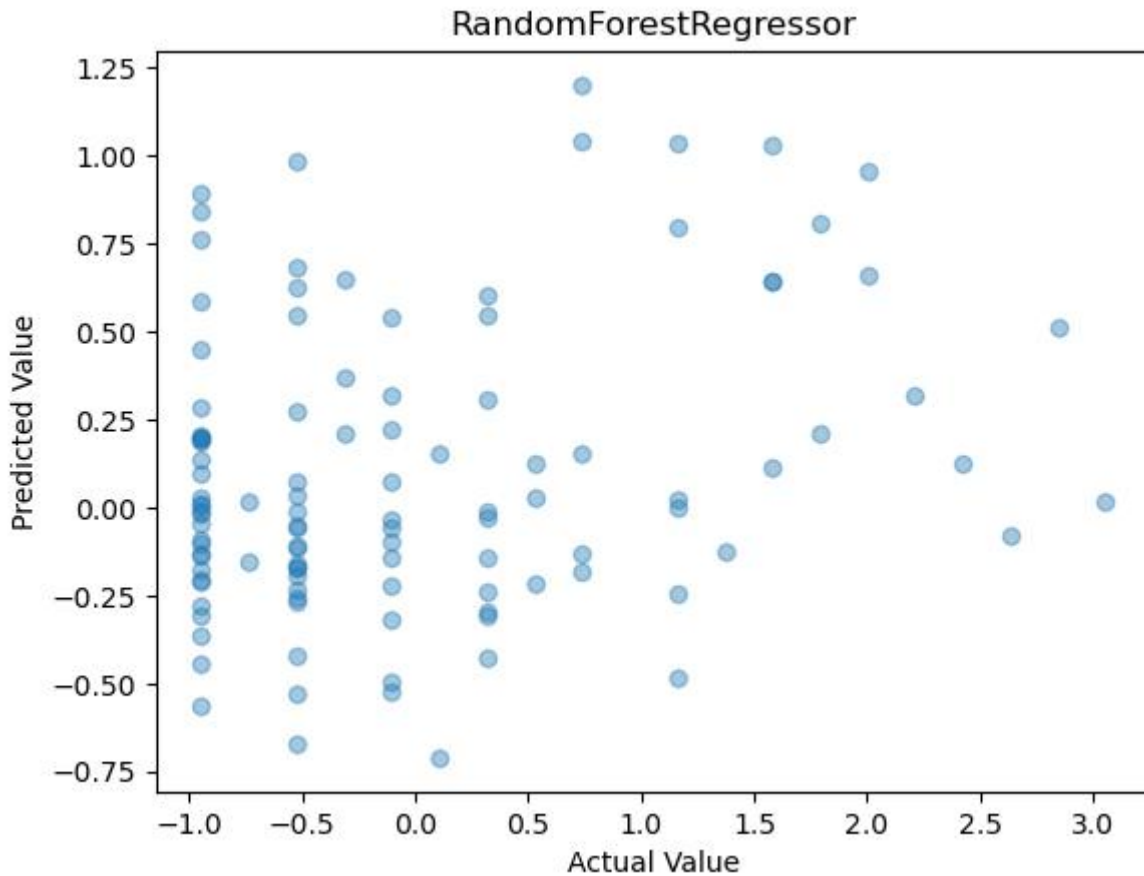
0.03356641465499399

1-5 모델이 다양한 일상 상황에서도 잘 동작한다는 것을 설명하고 시각화 하라

두 모델링 모두 앞선 설명과 같이 종속 변수인 absences와 상관성이 떨어지는 것에 따라 r2 score가 점수가 낮다. 랜덤 포레스트 회귀는 일반성이 강점으로 상관성이 있는 독립변수들을 활용하면 일상 상황에 활용 가능하다.

In [56]:

```
import matplotlib.pyplot as plt
plt.scatter(Ytest, predYtest, alpha = 0.4)
plt.xlabel('Actual Value')
plt.ylabel('Predicted Value')
plt.title('RandomForestRegressor')
plt.show()
```



1-6 모델 최적화 방안에 대해 구체적으로 설명하라

종속 변수인 `absences`를 로그 변환 등으로 변환을 하거나 데이터들을 좀 더 많이 수집하여 상관성을 더 확보할 필요가 있다고 생각된다.

통계분석

2

광고횟수와 광고비에 따른 매출액의 데이터이다.

/kaggle/input/adp-p8/problem2.csv

2-1 광고비 변수를 가변수 처리후 다중회귀를 수행하여 회귀계수가 유의한지 검정

In [58]:

```
df = pd.read_csv('data/problem2.csv', encoding = 'cp949')
```

In [59]:

```
df.head()
```

Out[59]:

	광고횟수	광고비	매출액
0	2	낮음	15
1	3	낮음	16
2	4	낮음	17
3	4	높음	18
4	5	높음	20

In [60]:

```
df['광고비'].unique()
```

Out[60]:

```
array(['낮음', '높음'], dtype=object)
```

In [61]:

```
df.isnull().sum()
```

Out[61]:

```
광고횟수    0
광고비      0
매출액      0
dtype: int64
```

In [62]:

```
newdf = pd.get_dummies(df)
```


In [63]:

```
newdf.head()
```

Out[63]:

	광고횟수	매출액	광고비_낮음	광고비_높음
0	2	15	1	0
1	3	16	1	0
2	4	17	1	0
3	4	18	0	1
4	5	20	0	1

In [65]:

```
import statsmodels.api as sm
```

In [66]:

```
Xdata = newdf.drop(['매출액'], axis = 1)  
target = newdf['매출액']
```

In [67]:

```
# 상수항 추가  
Xdata1 = sm.add_constant(Xdata, has_constant = 'add')
```

In [68]:

```
# OLS 검정
lmdl = sm.OLS(target, Xdata1)
lmdlfit = lmdl.fit()
lmdlfit.summary()
```

Out [68]:

OLS Regression Results

Dep. Variable:	매출액	R-squared:	0.982
Model:	OLS	Adj. R-squared:	0.978
Method:	Least Squares	F-statistic:	221.2
Date:	Sun, 20 Nov 2022	Prob (F-statistic):	9.96e-08
Time:	16:00:04	Log-Likelihood:	-11.883
No. Observations:	11	AIC:	29.77
Df Residuals:	8	BIC:	30.96
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P> t	[0.025	0.975]
const	8.2836	0.374	22.175	0.000	7.422	9.145
광고횟수	1.4350	0.074	19.518	0.000	1.265	1.605
광고비_낮음	3.8805	0.285	13.621	0.000	3.223	4.537
광고비_높음	4.4032	0.364	12.109	0.000	3.565	5.242

Omnibus:	7.665	Durbin-Watson:	0.919
Prob(Omnibus):	0.022	Jarque-Bera (JB):	3.407
Skew:	1.265	Prob(JB):	0.182
Kurtosis:	4.015	Cond. No.	9.91e+16

Notes:

- [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
 [2] The smallest eigenvalue is 6.96e-32. This might indicate that there are strong multicollinearity problems or that the design matrix is singular.

모든 변수에 대한 회계계수의 p value가 0.05 밑으로 모두 유의미하다고 판단된다.

2-2 회귀식이 유의한지 판단

1. F-statistic의 p value 값은 Prob (F-statistic)으로 표현되는데, 이는 9.96e-08으로 0.05보다 작기에 이 회귀식은 회귀 분석 모델 전체에 대해 통계적으로 의미가 있다고 볼 수 있다.
2. Adj. R-squared가 표기 되어 있는데, 값이 0.978이며 97.8% 만큼의 설명력을 가진다고 판단된다.

3

A생산라인의 제품 평균은 5.7mm이고 표준편차는 0.03, B생산라인의 제품 평균은 5.6mm이고 표준편차는 0.04라면 5%유의수준으로 두 제품의 평균이 차이가 있는지 여부를 검정하기 $Z(0.05) = 1.65$

3-1 귀무가설과 대립가설을 세워라

귀무가설 : 두 제품의 평균 차이가 없다.

연구가설 : 두 제품의 평균 차이가 존재한다.

3-2 두 평균이 차이가 있는지 검정하라

In [69]:

```
from scipy import stats
```

In [71]:

```
muA = 5.7
stdA = 0.03
muB = 5.6
stdB = 0.04
n = 12

rvsA = stats.norm.rvs(loc = muA, scale = stdA, size = n, random_state = 1234)
rvsB = stats.norm.rvs(loc = muB, scale = stdB, size = n, random_state = 1234)

stats.ttest_ind(rvsA, rvsB, equal_var = False)
```

Out[71]:

```
Ttest_indResult(statistic=6.208769010979042, pvalue=4.203409249564616e-06)
```

p 값이 0.05보다 작으므로 연구가설이 채택되어 두 제품의 평균차이가 존재한다.

4

바이러스 감염 분류표를 보고 베이지안 분류 방법을 사용해 양성으로 예측된 사람이 실제로 양성일 확률을 구하라

	양성(실제)	음성(실제)
양성(예측)	370	10
음성(예측)	15	690

In [73]:

```
import numpy as np
```

In [74]:

```
PredPos = 370 + 10
PredNeg = 15 + 690
RealPos = 370 + 15
RealNeg = 10 + 690
TotalNum = PredPos + PredNeg

CrossPos = 370
Prob = CrossPos / PredPos * 0.01
print(np.round(Prob, 4))
```

0.0097

5 주어진 데이터에서 신뢰구간을 구하려고한다

정규분포에서 표본을 추출함 $Z(0.05) = -1.65$, $Z(0.025) = -1.96$, $T(0.05, 8) = 1.860$, $T(0.025, 8) = 2.306$
 데이터(9개): [3.1, 3.3, 3.5, 3.7, 3.9, 4.1, 4.3, 4.4, 4.7]

5-1 모분산을 모르는 경우 주어진 데이터의 95% 신뢰구간을 구하라

In [75]:

```
data = [3.1, 3.3, 3.5, 3.7, 3.9, 4.1, 4.3, 4.4, 4.7]
```

In [78]:

```
sample_mean = np.round(np.mean(data), 2)
sample_std = np.round(np.std(data), 2)
t_025 = 2.306
Lt = np.round(sample_mean - t_025*sample_std / np.sqrt(9), 2)
Ut = np.round(sample_mean + t_025*sample_std / np.sqrt(9), 2)
print(Lt, Ut)
```

3.51 4.27

sigma = 0.04인걸 알고있을때의 95% 신뢰구간을 구하라

In [80]:

```
z_025 = 1.96
sigma = 0.04
Lnor = np.round(sample_mean - z_025*sigma / np.sqrt(9), 2 )
Unor = np.round(sample_mean + z_025*sigma / np.sqrt(9), 2 )
print(Lnor, Unor)
```

3.86 3.92