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(traveltime) 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
학고횟수(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

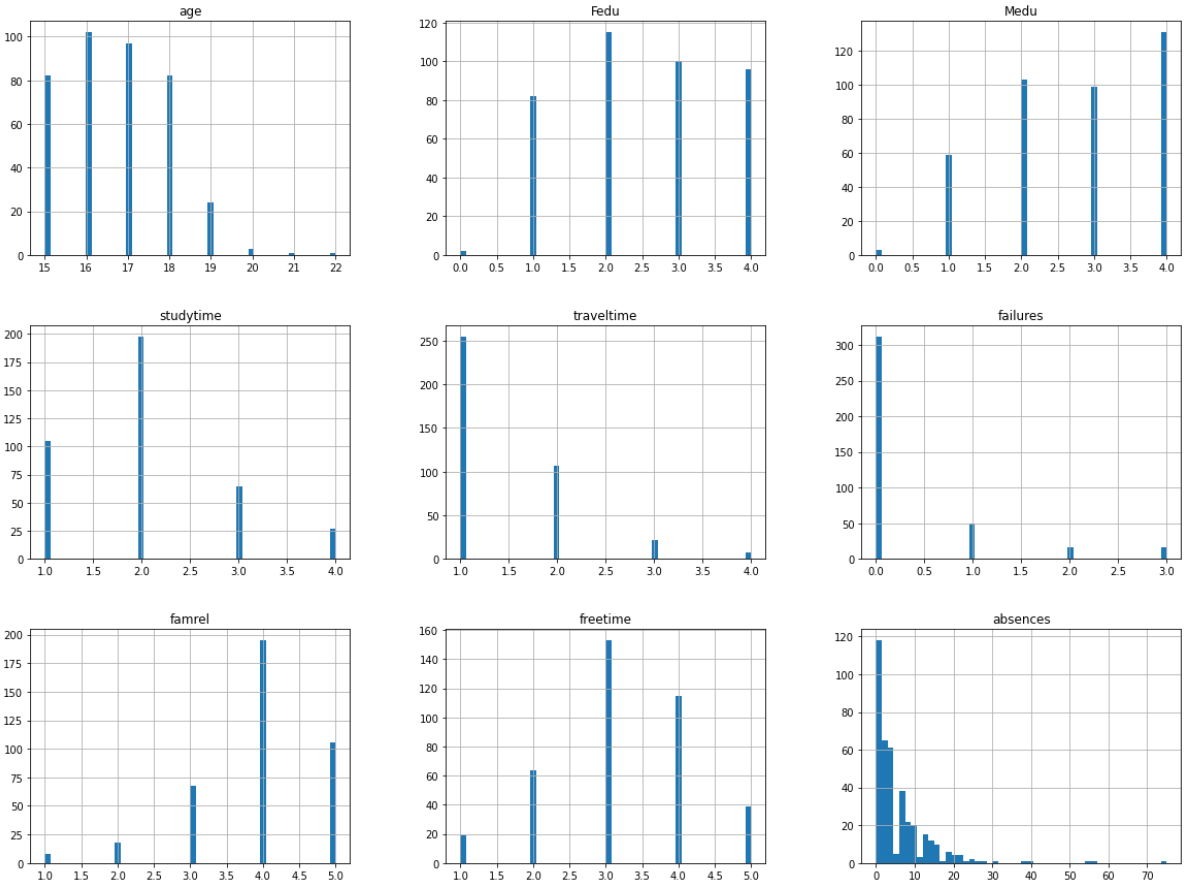
In [124…

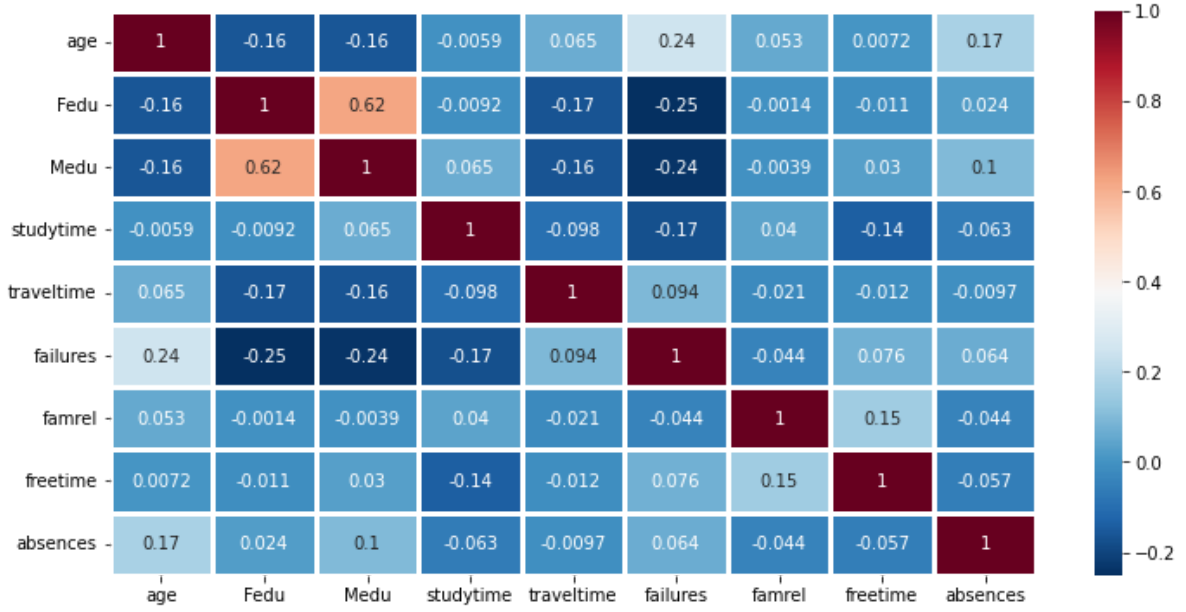
```
import pandas as pd
import numpy as np

data = pd.read_csv('problem1.csv')

print(data.info())
print(data.isnull().sum())
print(data.describe())

obj = [col for col in data.columns if data[col].dtype == 'object']
numeric = [col for col in data.columns if data[col].dtype != 'object']

for col in obj :
    print('=========', col, '=========')
    print(data[col].value_counts())


# 마법의 matplotlib 명령
%matplotlib inline

# 수치형 변수 히스토그램 그려보기
import matplotlib.pyplot as plt
data.hist(bins = 50, figsize = (20,15))
plt.show()

# 상관관계 살펴보기
data_corr = data.corr()

# 히트맵 그리기
import seaborn as sns
plt.figure(figsize = (12,6))
sns.heatmap(data_corr,
            xticklabels = data_corr.columns,
```

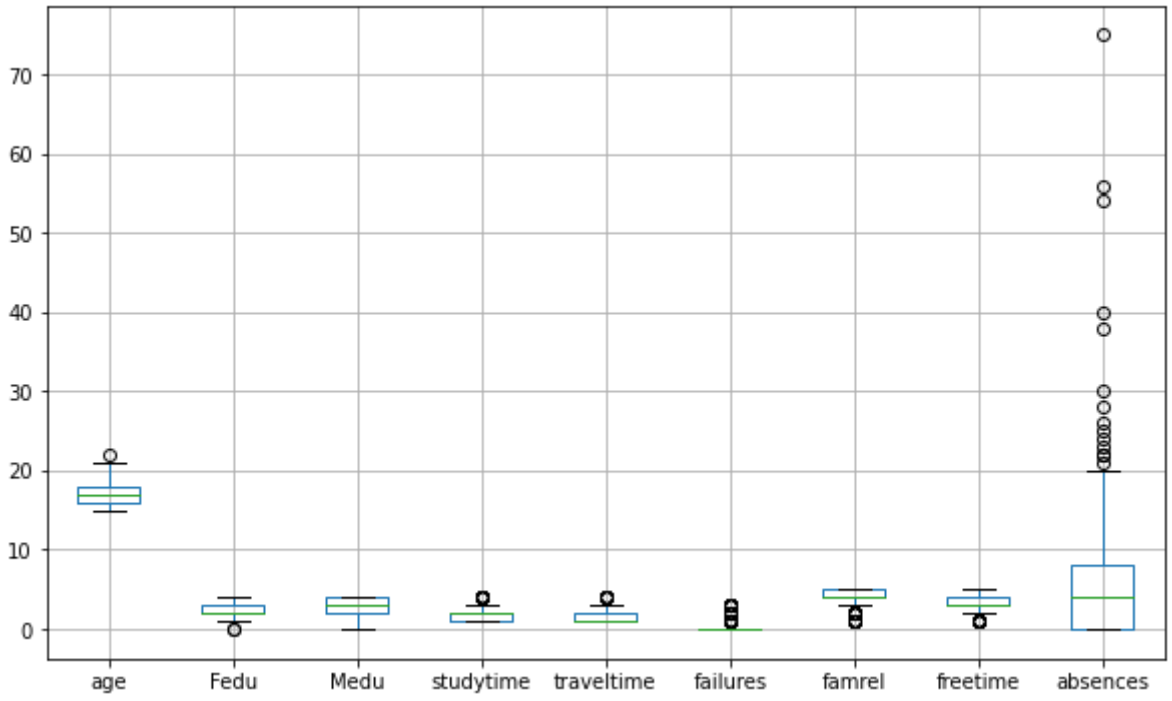
```
            yticklabels = data_corr.columns,
            annot = True,
            cmap = 'RdBu_r',
            linewidth = 3)
plt.show()

# boxplot 그리기
data.boxplot(figsize = (10,6)) # numeric 변수만 그려짐
plt.show()

data.boxplot(column='failures')
plt.show()
```

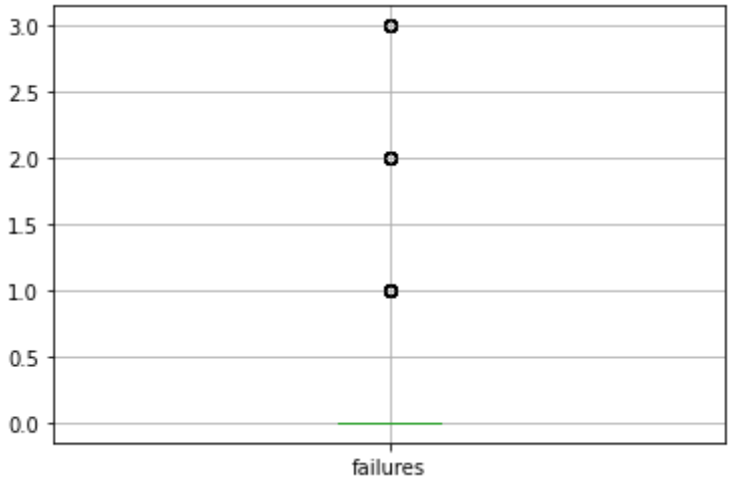
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 395 entries, 0 to 394
Data columns (total 12 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   sex         395 non-null    object
 1   age         392 non-null    float64
 2   Pstatus     395 non-null    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
None
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
                age        Fedu        Medu   studytime  traveltime    failures  W
count  392.000000  395.000000  395.000000  395.000000  392.000000  395.000000
mean    16.698980    2.521519    2.749367    2.035443    1.446429    0.334177
std      1.279865    1.088201    1.094735    0.839240    0.695022    0.743651
min     15.000000    0.000000    0.000000    1.000000    1.000000    0.000000
25%     16.000000    2.000000    2.000000    1.000000    1.000000    0.000000
50%     17.000000    2.000000    3.000000    2.000000    1.000000    0.000000
75%     18.000000    3.000000    4.000000    2.000000    2.000000    0.000000
max     22.000000    4.000000    4.000000    4.000000    4.000000    3.000000

           famrel    freetime    absences
count  395.000000  390.000000  395.000000
mean     3.944304    3.233333    5.708861
std      0.896659    1.000985    8.003096
min      1.000000    1.000000    0.000000
25%      4.000000    3.000000    0.000000
50%      4.000000    3.000000    4.000000
75%      5.000000    4.000000    8.000000
max      5.000000    5.000000   75.000000
========= sex =========
F    208
M    187
Name: sex, dtype: int64
========= Pstatus =========
T    354
A     41
Name: Pstatus, dtype: int64
========= guardian =========
mother    273
father     90
```

```
other       32
Name: guardian, dtype: int64
```









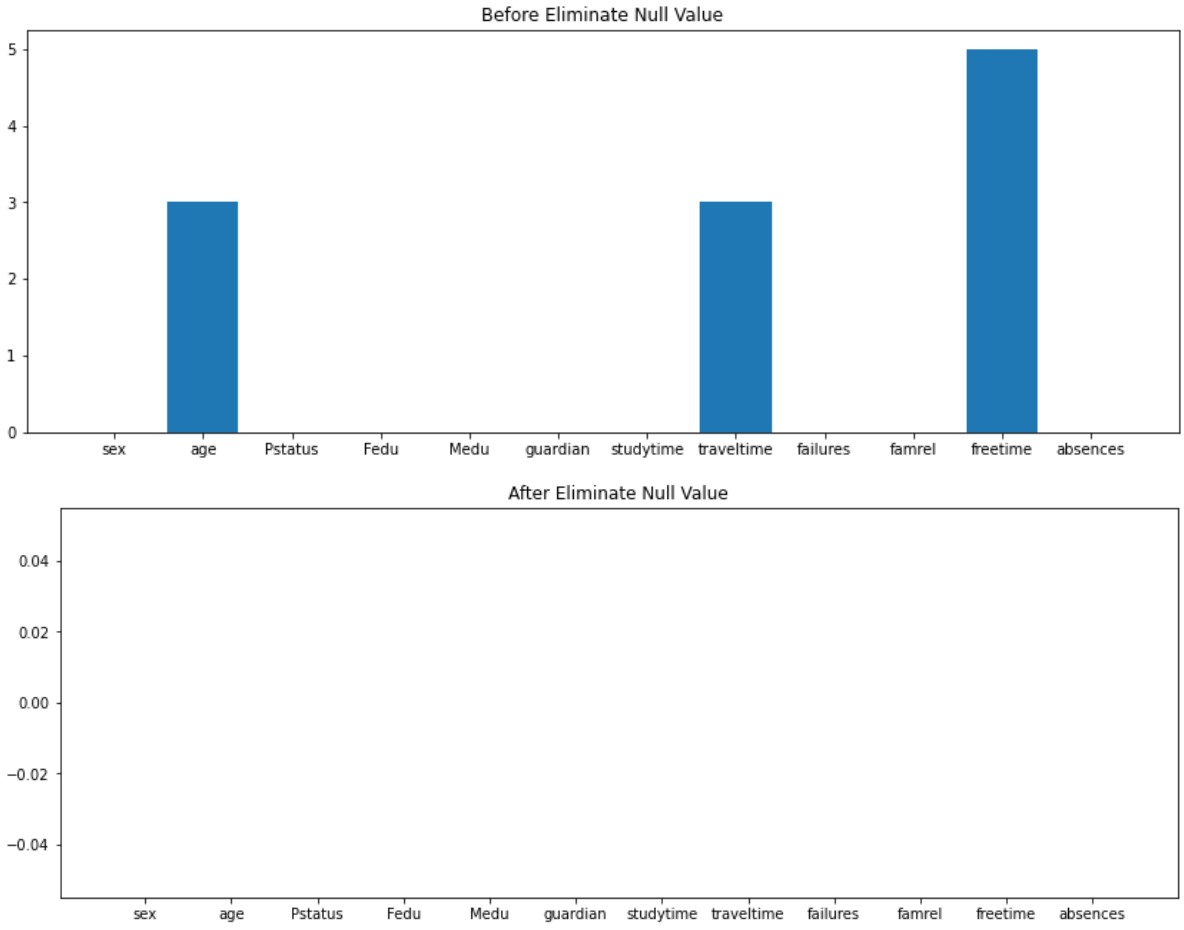
답안 : EDA 결과

- 데이터는 총 12개 컬럼 및 395행으로 이루어져 있으며, y값은 absences이다.
- 총 3개의 컬럼에서 null값이 확인된다 : age, traveltime, freetime
- describe()를 통해 15세에서 22세까지의 학생을 대상으로 조사한 데이터임을 유추할 수 있다.
- absences의 max값 75는 이상치로 확인되나, 따로 보정은 하지 않겠다.
- data.info()를 통해 수치형 컬럼 9개, object 컬럼 3개가 확인된다.
- 수치형 변수들에 대해 histogram을 그린 결과 형식은 수치형이나 일종의 범주형으로 봐도 좋을 데이터로 판단된다.
- 대부분 정규분포를 띄지 않고 있기에 추후 log1p등의 변환이 필요할 것으로 판단된다.
- y값인 absences와 가장 연관이 있는 컬럼은 age로 확인된다.

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

In [125…
```
null_before = pd.DataFrame(data.isnull().sum(), columns=['num'])
plt.figure(figsize = (14,5))
plt.bar(null_before.index, null_before.num.values)
```

```
plt.title('Before Eliminate Null Value')
plt.show()

cols = ['age', 'traveltime', 'freetime']

for col in cols :
    data[col] = data[col].fillna(data[col].median())

null_after = pd.DataFrame(data.isnull().sum(), columns=['num'])

plt.figure(figsize = (14,5))
plt.bar(null_after.index, null_after.num.values)
plt.title('After Eliminate Null Value')
plt.show()
```



답안 : 결측치 처리 및 변화 시각화, 추가 전처리가 필요하다면 이유와 기대효과

- 결측치는 age, traveltime, freetime에서 확인된다.
- 결측치를 각 컬럼의 평균값으로 대치할 경우 데이터 형식이 깨질 수 있다.
- 따라서 각 컬럼의 중위수로 대치하고자 한다.
- 그래프를 통해 Null값 제거 전과 제거 후의 변화를 확인할 수 있다.
- 추가 전처리로 y값 absences를 제외한 나머지 컬럼의 이상치 제거를 제안한다.
- 이상치 제거를 통해 모델 적용시 정확도를 높일 수 있을 것으로 판단된다.
- 그리고 object컬럼에 대한 수치형으로의 변환 역시 필요하다.

1-3 결석일수 예측모델을 2개 제시하고 선택한 근거 설명

답안

- 결석일수 예측에 사용할 모델 : LinearRegression, RandomForesetRegression
- 선택 이유</br> . LinearRegression : 선형 모델이 학습이 빠르다는 특성을 활용하기 위해 선택하였다. 또한 전통적인 회귀 기법을 통해 해당 데이터를 얼마나 설명할 수 있을지 궁금하다.</br> .RandomForestRegressor : 대표적인 앙상블 모델로서 많은 사람들에게 사랑받는 모델이다. 일반적으로 배깅 방법을 사용한 결정트리 앙상블 모델로서 무작위로 선택한 특성들 중 최선의 특성을 찾는 방식으로 속도도 빠르고 정확도도 매우 높은 편이다.
- 이에 위 2개 모델을 활용하여 모델링을 진행하고자 한다.

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

In [135…

```
def detect_outlier(df = None, column = None, weight = 1.5) :
    Q1 = data[column].quantile(0.25)
    Q3 = data[column].quantile(0.75)

    IQR = Q3 - Q1
    IQR_weight = IQR * weight

    out_idx = data[(data[column] < Q1 - IQR_weight) | data[column] > Q3 + IQR_weigh

    # print('==========', column, '==========')
    # print(out_idx)

    return out_idx


out_col = [ 'age', 'Fedu', 'Medu', 'studytime', 'traveltime', 'failures', 'famrel',

for col in out_col :
    detect_outlier(data, col)

detect_outlier(data, 'failures')
data.loc[detect_outlier(data, 'failures'), 'failures'] = data.failures.median()

from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()

for col in obj :
    data[col] = le.fit_transform(data[col])


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

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size = 0.3, random_sta

from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()

scaler.fit(X_train)

X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)

from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
```

```
lr = LinearRegression()
rf = RandomForestRegressor()

lr.fit(X_train, y_train)
rf.fit(X_train, y_train)

y_pred_lr = lr.predict(X_test)
y_pred_rf = rf.predict(X_test)

from sklearn.metrics import mean_squared_error
mse_lr = mean_squared_error(y_test, y_pred_lr)
mse_rf = mean_squared_error(y_test, y_pred_rf)

rmse_lr = np.sqrt(mse_lr)
rmse_rf = np.sqrt(mse_rf)

print("LinearRegression의 RMSE : ", rmse_lr)
print("RandomforestRegressor RMSE : ", rmse_rf)
```

```
LinearRegression의 RMSE :  6.171419339729821
RandomforestRegressor RMSE :  7.688032807519674
```

답안

- 위에서 주어진 데이터 셋은 분류(Classifier)가 아닌 회귀(Regressor)를 해야 하는 데이터이다.
- 따라서 회귀에서 사용할 수 있는 여러 평가지표들 중 대표적인 RMSE를 활용하고자 한다.
- RMSE는 MSE의 값에 루트를 씌운 값으로서 Mean Squared Error에 루트를 씌움으로써 에러가 크면 클수록 그에 따른 가중치가 높이 반영된다.
- 에러에 따른 손실이 기하급수적으로 올라가는 상황에서 쓰기 적합하며
- MAE와 마찬가지로 예측값과 결과값의 Scale이 같기에 직관적이다.

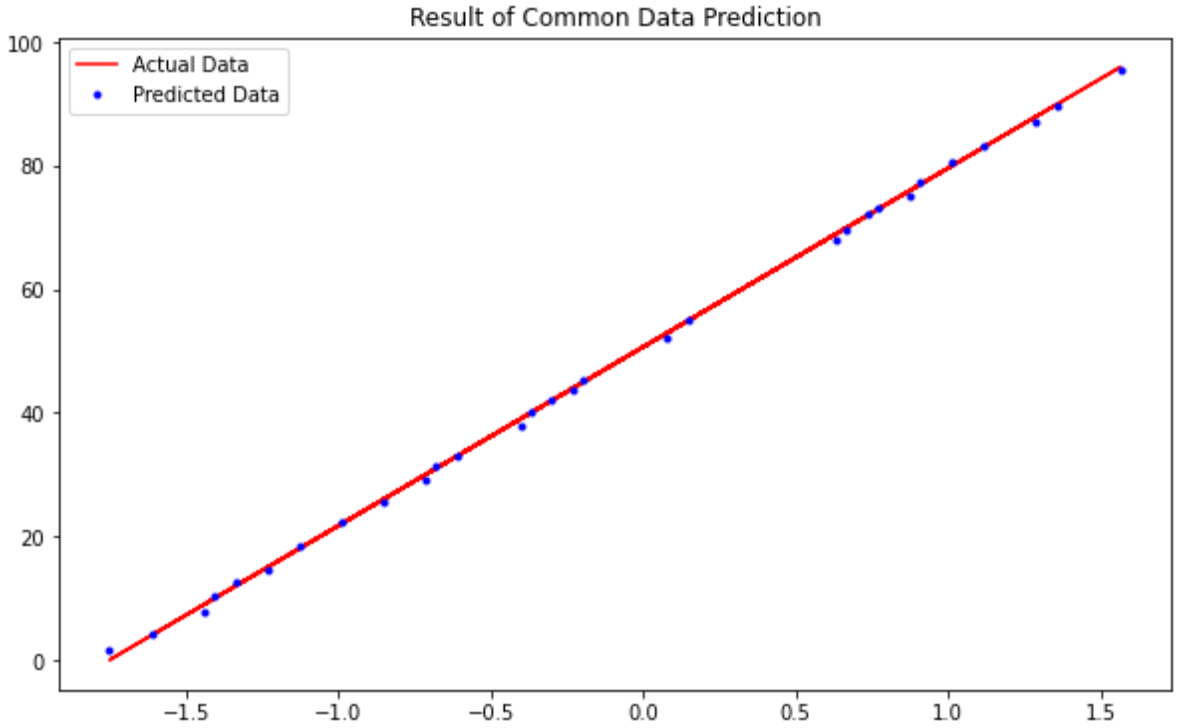
1-5 모델이 다양한 일상 상황에서도 잘 동작한다는 것을 설명하고 시각화 하라

In [127…

```
# 일상생활에서 흔히 발생하는 선형성을 가진 임의의 데이터 생성
train_source = {'x1' : range(0,100),
                'x2' : np.arange(0,1,0.01),
                'y' : range(0,100)}
train = pd.DataFrame(train_source)


X = train.drop('y', axis = 1)
y = train['y']

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size = 0.3, random_sta

from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()

scaler.fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)


from sklearn.ensemble import RandomForestRegressor
rf = RandomForestRegressor()
```

```
rf.fit(X_train, y_train)

y_pred = rf.predict(X_test)

from sklearn.metrics import mean_squared_error
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)

print("RMSE : ", rmse)

plt.figure(figsize = (10,6))
plt.plot(X_test[:,0], y_test,'r-' , label = 'Actual Data')
plt.plot(X_test[:,0], y_pred,'b.' ,label = 'Predicted Data')
plt.title('Result of Common Data Prediction')
plt.legend()

plt.show()
```

```
RMSE :  0.6393043093863829
```



답안

- 일상생활에서 흔히 접할 수 있는 선형성을 가진 데이터를 임의로 생성하였다.
- x1, x2 변수의 증가는 같되 스케일을 다르게 설정하였다.
- 이후 위의 Process와 똑같이 데이터를 나누고 스케일링을 진행한 다음 RandomForestRegressor를 사용하여 모델링 진행
- 30%의 테스트 데이터에 대해 predict를 진행하였고, RMSE는 약 0.6으로 확인된다.
- 이후 Actual 값과 Predicted 값을 도식화 했을 때 Actual값(직선)에 Predicted값(Dot)이 거의 근접하게 위치해 있음이 확인된다.
- 원래는 주어진 데이터를 통해 학습한 모델을 활용하여 일상생활 데이터를 Predict만 하려 했으나,
- 주어진 데이터의 컬럼이 11개나 되기에 다시 fit 과정을 거쳐 모델에 학습을 시켰음.
- 이를 통해 위에서 생성한 모델이 일상생활의 다양한 데이터에서도 잘 동작함을 확인할 수 있다.

1-6 모델 최적화 방안에 대해 구체적으로 설명하라

답안

- 현재 위에서 사용한 LinearRegression과 RandomForestRegressor는 HyperParameter 튜닝을 전혀 하지 않은 상황이다.
- GridSearchCV를 통해 HyperParameter별 최적값을 도출해 낼 수 있다.
- 또한 주어진 데이터의 전처리 과정에서 파생변수는 사용하지 않은 상황이다.
- Corelation을 확인하긴 했지만 다중공선성 역시 확인하지 않았다.
- RandomForestRegressor의 Feature Importance 역시 확인하지 않았다.
- 또한 PCA를 통한 차원축소 기법 역시 한번 도전해보고 싶다.
- 위에서 언급한 남아있는 여러가지 Action을 수행할 경우 모델의 RMSE값은 더욱 작아질 것으로 판단된다.

In [129…
```
formula = 'absences ~ sex + age + Pstatus + Fedu + Medu + guardian + studytime + tra
```

In [130…
```
import numpy as np
import statsmodels.api as sm
import statsmodels.formula.api as smf


model = smf.ols(formula = formula, data = data)
result = model.fit()
result.summary()
```

Out[130]:

OLS Regression Results

Dep. Variable:	absences	**R-squared:**	0.091
Model:	OLS	**Adj. R-squared:**	0.067
Method:	Least Squares	**F-statistic:**	3.826
Date:	Wed, 16 Nov 2022	**Prob (F-statistic):**	6.02e-05
Time:	10:53:48	**Log-Likelihood:**	-1362.8
No. Observations:	395	**AIC:**	2748.
Df Residuals:	384	**BIC:**	2791.
Df Model:	10		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-7.6337	6.085	-1.254	0.210	-19.598	4.331
sex	-1.2791	0.846	-1.512	0.131	-2.943	0.384
age	1.0404	0.324	3.209	0.001	0.403	1.678
Pstatus	-2.7818	1.299	-2.141	0.033	-5.336	-0.228
Fedu	-0.2959	0.467	-0.633	0.527	-1.215	0.623
Medu	1.1363	0.466	2.437	0.015	0.220	2.053
guardian	1.5220	0.775	1.965	0.050	-0.001	3.045
studytime	-0.9603	0.496	-1.937	0.054	-1.935	0.015
traveltime	-0.0091	0.576	-0.016	0.987	-1.142	1.124
failures	-2.536e-16	2.06e-16	-1.228	0.220	-6.6e-16	1.52e-16
famrel	-0.3352	0.442	-0.759	0.448	-1.204	0.533
freetime	-0.4090	0.410	-0.998	0.319	-1.214	0.396

Omnibus:	327.332	**Durbin-Watson:**	1.899
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	7297.888
Skew:	3.399	**Prob(JB):**	0.00
Kurtosis:	22.930	**Cond. No.**	1.70e+18

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 4.49e-32. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

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 [120…

```
data = pd.read_csv('problem2.csv', encoding='cp949')
data = pd.get_dummies(data, columns=['광고비'])
data.columns = ['ad_num', 'income', 'price_high', 'price_low']

import statsmodels.api as sm
import statsmodels.formula.api as smf


model = smf.ols(formula = 'income ~ ad_num + price_high + price_low', data = data)
result = model.fit()
result.summary()

#data
```

```
C:₩Users₩storm₩anaconda3₩envs₩myworkspace₩lib₩site-packages₩scipy₩stats₩stats.py:160
4: UserWarning: kurtosistest only valid for n>=20 ... continuing anyway, n=11
  "anyway, n=%i" % int(n))
```

Out[120]:

OLS Regression Results

Dep. Variable:	income	**R-squared:**	0.982
Model:	OLS	**Adj. R-squared:**	0.978
Method:	Least Squares	**F-statistic:**	221.2
Date:	Wed, 16 Nov 2022	**Prob (F-statistic):**	9.96e-08
Time:	00:18:42	**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]
Intercept	8.2836	0.374	22.175	0.000	7.422	9.145
ad_num	1.4350	0.074	19.518	0.000	1.265	1.605
price_high	3.8805	0.285	13.621	0.000	3.223	4.537
price_low	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.**	1.84e+17

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 2.02e-32. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

답안

- 컬럼명이 한글로 되어 있기에 임의로 영어로 변경하였다.
- 광고비 변수를 pd.get_dummies를 활용하여 가변수 처리한 결과 추가로 2개의 컬럼이 생성되었다.
- 데이터를 활용하여 다중회귀를 수행한 결과 모델이 전체 데이터의 98.2%를 설명할 수 있음이 확인된다.(R square)
- 종속변수 income에 대한 세개의 feature의 회귀계수와 유의성을 판단할 수 있는 p-value는 다음과 같다.
- [회귀계수] 광고횟수 : 1.435, 광고비 높음 : 3.8805, 광고비 낮음 : 4.4032
- [p-value] 3개의 모든 feature에 대한 p-value값이 유의수준 5%인 0.05보다 작기에 모두 유의하다 판단할 수 있다.
- 다라서 회귀식은 다음과 같다. y = 8.2836 + 1.4350*광고횟수* + *3.8805*광고비_높음 + 4.4032*광고비 낮음

- y값에 가장 영향을 미치는 변수는 광고비 낮음 변수로서 광고 횟수보다 광고비가 낮을 때 income에 더 영향을 주는 것으로 판단할 수 있다.

2-2 회귀식이 유의한지 판단

In [121…
```
result.summary()
```

```
C:₩Users₩storm₩anaconda3₩envs₩myworkspace₩lib₩site-packages₩scipy₩stats₩stats.py:1604: UserWarning: kurtosistest only valid for n>=20 ... continuing anyway, n=11
  "anyway, n=%i" % int(n))
```

Out[121]:

OLS Regression Results

Dep. Variable:	income	**R-squared:**	0.982
Model:	OLS	**Adj. R-squared:**	0.978
Method:	Least Squares	**F-statistic:**	221.2
Date:	Wed, 16 Nov 2022	**Prob (F-statistic):**	9.96e-08
Time:	00:25:13	**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]
Intercept	8.2836	0.374	22.175	0.000	7.422	9.145
ad_num	1.4350	0.074	19.518	0.000	1.265	1.605
price_high	3.8805	0.285	13.621	0.000	3.223	4.537
price_low	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.**	1.84e+17

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 2.02e-32. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

답안

- 회귀식은 위에서 언급한대로 y = 8.2836 + 1.4350*광고횟수* + *3.8805*광고비_높음 + 4.4032*광고비 낮음 이며,
- 해당 회귀식의 F통계량은 221.2로 확인된다. 통계량에 따른 유의확률 p-value는 9.96e-08 로서

- 유의수준 5%인 0.05보다 매우 작은 곳에 위치하기에 최종적으로 해당 회귀식은 유의하다 판단할 수 있다.

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

답안

- 귀무가설 : 두 생산라인에서 생산된 제품의 평균에는 차이가 없다. mu_a = mu_b
- 대립가설 : 두 생상라인에서 생산된 제품의 평균에는 차이가 있다. mu_a > mu_b(단측검정)

3-2 두 평균이 차이가 있는지 검정하라

In [123…

```
# 두 집단의 평균의 차이를 계산하는 공식
# (mu_a - mu_b) / ((var_a / n) + (var_b / n))**(1/2) # Z통계량값 계산

n = 12
mu_a = 5.7
mu_b = 5.6
var_a = (0.03)**2
var_b = (0.04)**2

Z = (mu_a - mu_b) / ((var_a / n) + (var_b / n))**(1/2)
print('두 집단의 평균의 차이에 대한 Z통계량 : ', Z)
print('두 집단의 평균 차이에 대한 Z통계량이 5%유의수준보다 크므로,')
print('귀무가설을 기각하고 대립가설을 채택한다. 즉, A생산라인의 평균이 B생산라인의 평
# 원래 각 집단 표본의 수가 30 미만이기에 t통계량을 사용해야 하나, 문제에서 Z값이 주어
# 또한 문제에서 5% 유의수준에 대해 Z(0.05) = 1.65로 주어졌으므로 양측검정이 아닌 단측
```

```
두 집단의 평균의 차이에 대한 Z통계량 :  6.928203230275546
두 집단의 평균 차이에 대한 Z통계량이 5%유의수준보다 크므로,
귀무가설을 기각하고 대립가설을 채택한다. 즉, A생산라인의 평균이 B생산라인의 평균보다 크다는 결론을 내릴 수 있다.
```

4

바이러스 감염 분류표를 보고 베이지안 분류 방법을 사용해 양성으로 예측된 사람이 실제로 양성일 확률을 구하라. 유병률은 0.01

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

In [140…
```
# 실제 양성일 사건 : A
# 유병률 사건 : B
# P(A|B) =P(B|A) P(A) / P(B)

result = ((370 / (370+15))*0.01) / (370/(370+15)*0.01 + 10/(690+10)*0.99)
print("양성으로 예측된 사람이 실제로 양성일 확률 : ", result)
```

양성으로 예측된 사람이 실제로 양성일 확률 : 0.4045926735921268

5 주어진 데이터에서 신뢰구간을 구하려한다

정규분포에서 표폰을 추출함[Z(0.05) = -1.65 , Z(0.025) = -1.96, T(0.05, 8) = 1.860 , T0.025(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 [142…
```
import pandas as pd
import numpy as np

source = {'num' : [3.1, 3.3, 3.5, 3.7, 3.9, 4.1, 4.3, 4.4, 4.7]}
data = pd.DataFrame(source)

mean = data.num.mean()
n = 9
std = data.num.std()

under = mean - 1.96 * (std / np.sqrt(n))
upper = mean + 1.96 * (std / np.sqrt(n))

print('주어진 데이터의 95% 신뢰구간 : ', under, '<= mu <=', upper)
```

주어진 데이터의 95% 신뢰구간 : 3.539425102107476 <= mu <= 4.238352675670302

sigma = 0.04인걸 알고있을때의 95% 신뢰구간을 구하라

In [143…
```
mean = data.num.mean()
n = 9
std = 0.4

under = mean - 1.96 * (std / np.sqrt(n))
upper = mean + 1.96 * (std / np.sqrt(n))

print('주어진 데이터의 95% 신뢰구간 : ', under, '<= mu <=', upper)
```

```
주어진 데이터의 95% 신뢰구간 :  3.6275555555555554 <= mu <= 4.150222222222222
```