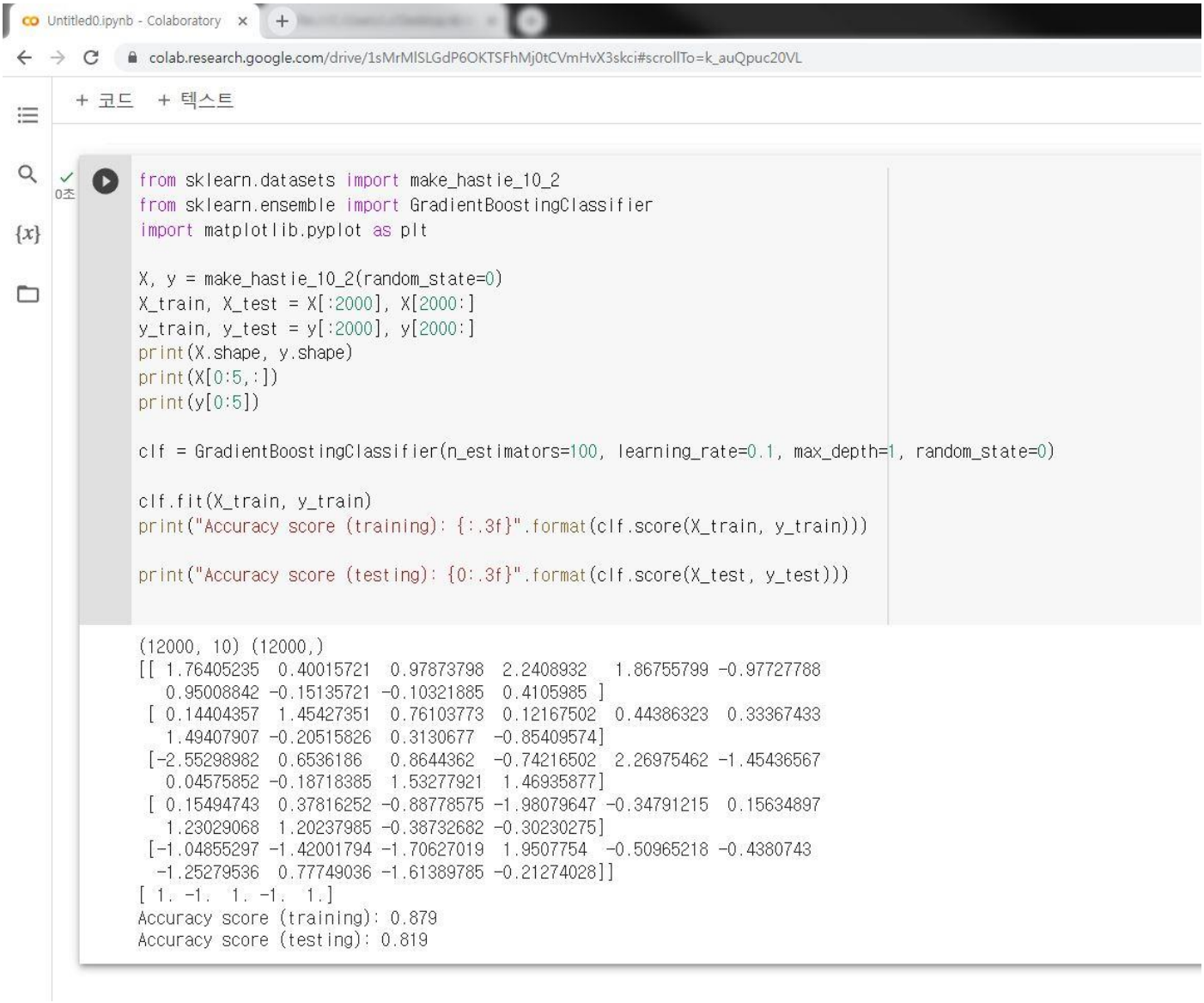
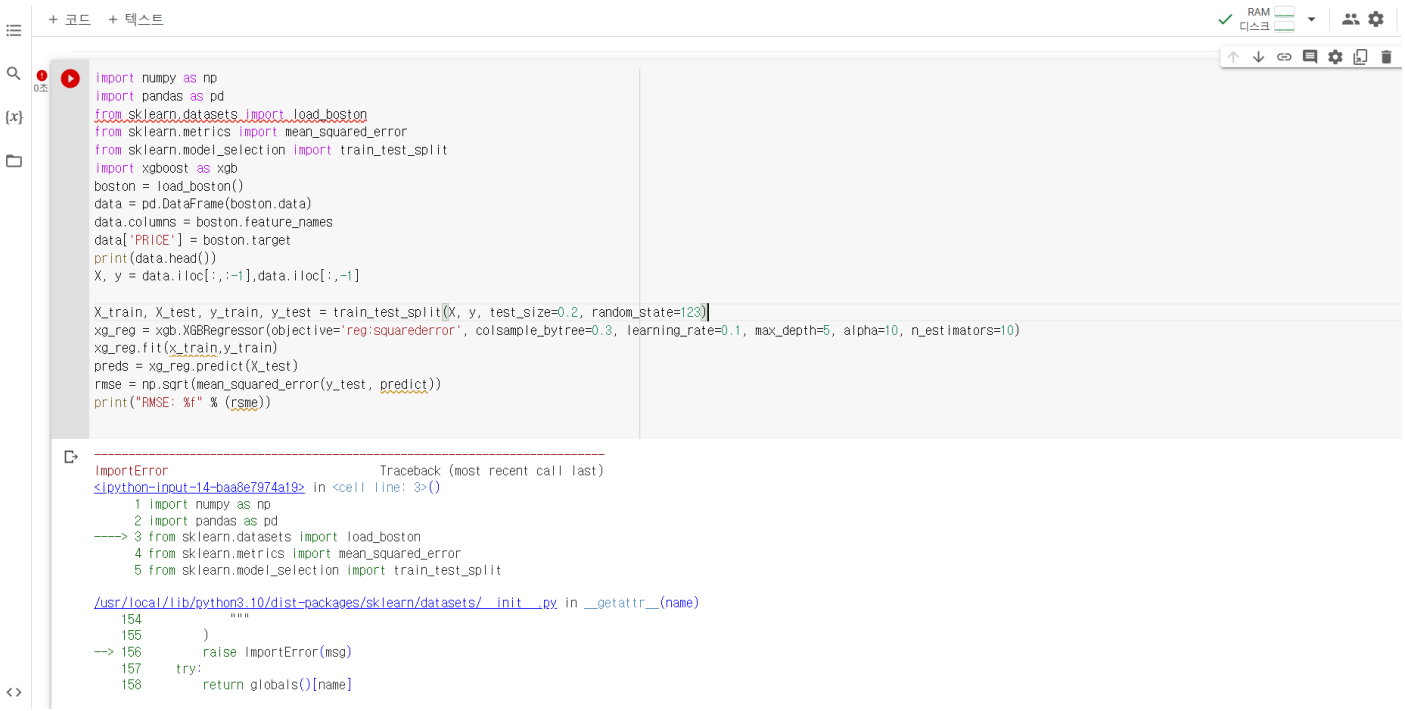
# XGBoost 실습


Untitled0.ipynb - Colaboratory
colab.research.google.com/drive/1sMrMlSLGdP6OKTSFhMj0tCVmHvX3skci#scrollTo=k_auQpuc20VL
+ 코드
+ 텍스트
0초
from sklearn.datasets import make_hastie_10_2
from sklearn.ensemble import GradientBoostingClassifier
import matplotlib.pyplot as plt
X, y = make_hastie_10_2(random_state=0)
X_train, X_test = X[:2000], X[2000:]
y_train, y_test = y[:2000], y[2000:]
print(X.shape, y.shape)
print(X[0:5,:])
print(y[0:5])
clf = GradientBoostingClassifier(n_estimators=100, learning_rate=0.1, max_depth=1, random_state=0)
clf.fit(X_train, y_train)
print("Accuracy score (training): {:.3f}".format(clf.score(X_train, y_train)))
print("Accuracy score (testing): {0:.3f}".format(clf.score(X_test, y_test)))
(12000, 10) (12000,)
[[ 1.76405235 0.40015721 0.97873798 2.2408932 1.86755799 -0.97727788
0.95008842 -0.15135721 -0.10321885 0.4105985 ]
[ 0.14404357 1.45427351 0.76103773 0.12167502 0.44386323 0.33367433
1.49407907 -0.20515826 0.3130677 -0.85409574]
[-2.55298982 0.6536186 0.8644362 -0.74216502 2.26975462 -1.45436567
0.04575852 -0.18718385 1.53277921 1.46935877]
[ 0.15494743 0.37816252 -0.88778575 -1.98079647 -0.34791215 0.15634897
1.23029068 1.20237985 -0.38732682 -0.30230275]
[-1.04855297 -1.42001794 -1.70627019 1.9507754 -0.50965218 -0.4380743
-1.25279536 0.77749036 -1.61389785 -0.21274028]]
[ 1. -1. 1. -1. 1.]
Accuracy score (training): 0.879
Accuracy score (testing): 0.819

# LightGBM 실습



```
import numpy as np
import pandas as pd
from sklearn.datasets import load_boston
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split
import xgboost as xgb
boston = load_boston()
data = pd.DataFrame(boston.data)
data.columns = boston.feature_names
data['PRICE'] = boston.target
print(data.head())
X, y = data.iloc[:,:-1],data.iloc[:,-1]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=123)
xg_reg = xgb.XGBRegressor(objective='reg:squarederror', colsample_bytree=0.3, learning_rate=0.1, max_depth=5, alpha=10, n_estimators=10)
xg_reg.fit(x_train,y_train)
preds = xg_reg.predict(X_test)
rmse = np.sqrt(mean_squared_error(y_test, predict))
print("RMSE: %f" % (rsme))
```

```
---------------------------------------------------------------------------
ImportError                               Traceback (most recent call last)
<ipython-input-14-baa8e7974a19> in <cell line: 3>()
      1 import numpy as np
      2 import pandas as pd
----> 3 from sklearn.datasets import load_boston
      4 from sklearn.metrics import mean_squared_error
      5 from sklearn.model_selection import train_test_split

/usr/local/lib/python3.10/dist-packages/sklearn/datasets/__init__.py in __getattr__(name)
    154             """
    155         )
--> 156         raise ImportError(msg)
    157     try:
    158         return globals()[name]
```

실행 결과 에러 발생

에러내용을 종합하자면 해당 데이터는 인종적인 윤리문제로 인하여 1.2버젼 이상에서 삭제되었음

따라서 교육 용도로 해당 데이터 불러오기 실시

```
import numpy as np
import pandas as pd
import xgboost as xgb
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split

data_url = "http://lib.stat.cmu.edu/datasets/boston"
raw_df = pd.read_csv(data_url, sep="\s+", skiprows=22, header=None)
data = np.hstack([raw_df.values[::2, :], raw_df.values[1::2, :2]])
target = raw_df.values[1::2, 2]

data = pd.DataFrame(data)
data.columns = ['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS', 'RAD', 'TAX', 'PTRATIO', 'B', 'LSTAT']
data['PRICE'] = target

X, y = data.iloc[:,:-1], data.iloc[:,-1]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=123)

xg_reg = xgb.XGBRegressor(objective='reg:squarederror', colsample_bytree=0.3, learning_rate=0.1, max_depth=5, alpha=10, n_estimators=10)
xg_reg.fit(X_train, y_train)

preds = xg_reg.predict(X_test)
rmse = np.sqrt(mean_squared_error(y_test, preds))
print("RMSE: %f" % (rmse))
```

```
RMSE: 10.915755
```