**Python Machine Learning**

**Jinwoo Seol**

**All of these studies are referenced from youtuber Teck with Tim's Python Machine Learning Tutorial.**

<https://www.youtube.com/watch?v=1BYu65vLKdA&ab_channel=TechWithTim>

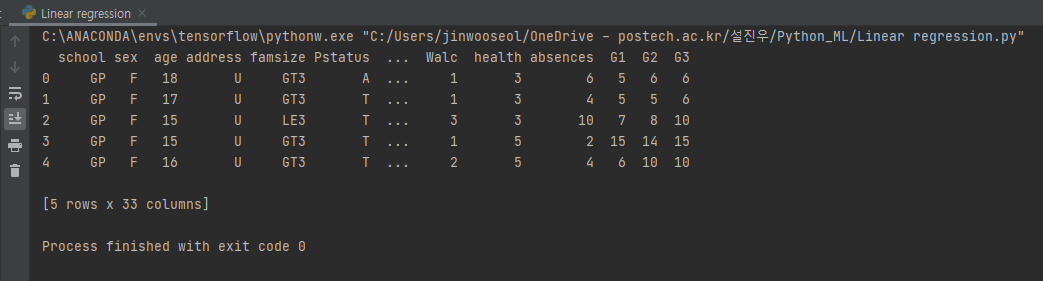
<https://www.techwithtim.net/tutorials/machine-learning-python/introduction/>

**06/29 Setup**

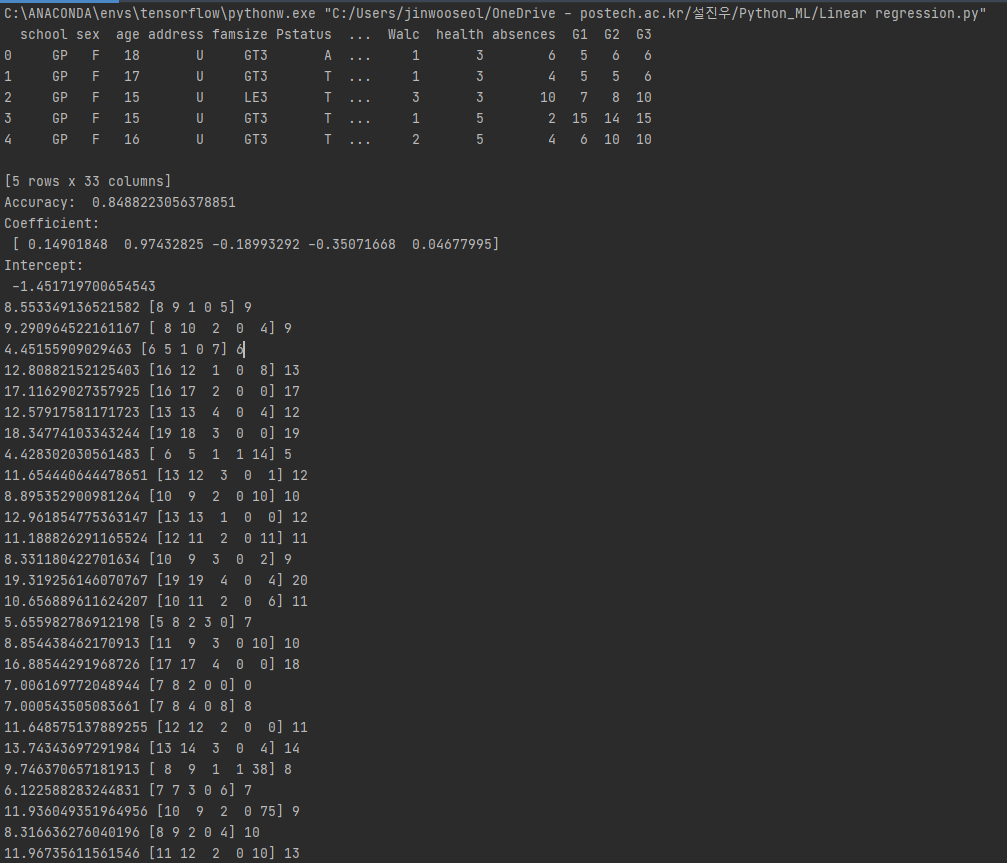
**06/29 Linear regression**

https://www.techwithtim.net/tutorials/machine-learning-python/linear-regression/

import pandas as pd  
import numpy as np  
import sklearn  
from sklearn.linear\_model import LinearRegression  
from sklearn.utils import shuffle  
  
data = pd.read\_csv("student-mat.csv", sep=";")  
print(data.head()) #show data  
data = data[["G1", "G2", "G3", "studytime", "failures", "absences"]]  
  
predict = "G3"  
  
x = np.array(data.drop([predict],1))  
y = np.array(data[predict]) # labels  
  
# x\_train -> x의 data set, y\_train -> y의 data set, x\_test -> x의 accuracy, y\_test same  
x\_train, y\_train, x\_test, y\_test = sklearn.model\_selection.train\_test\_split(x, y, test\_size = 0.1) # test\_size means accuracy percentage 10%



import pandas as pd  
import numpy as np  
import sklearn  
from sklearn.linear\_model import LinearRegression  
from sklearn.utils import shuffle  
  
data = pd.read\_csv("student-mat.csv", sep=";")  
print(data.head()) #show data  
data = data[["G1", "G2", "G3", "studytime", "failures", "absences"]]  
  
predict = "G3"  
  
x = np.array(data.drop([predict],1))  
y = np.array(data[predict]) # labels  
  
# x\_train -> x axis, y\_train -> y axis, x\_test -> x의 accuracy, y\_test same  
x\_train, x\_test, y\_train, y\_test = sklearn.model\_selection.train\_test\_split(x, y, test\_size = 0.1) # test\_size means accuracy percentage 10%  
  
linear = sklearn.linear\_model.LinearRegression()  
  
linear.fit(x\_train, y\_train) # x\_train, y\_train data에 가장 잘 맞는 best linear line  
acc = linear.score(x\_test, y\_test) # accuracy of linear regression  
  
print('Accuracy: ',acc)  
  
print('Coefficient: \n', linear.coef\_)  
print('Intercept: \n', linear.intercept\_)  
  
predictions = linear.predict(x\_test)  
  
for x in range(len(predictions)):  
 print(predictions[x], x\_test[x], y\_test[x])



**Saving Models & Plotting Data**

import pandas as pd  
import numpy as np  
import sklearn  
from sklearn.linear\_model import LinearRegression  
from sklearn.utils import shuffle  
import matplotlib.pyplot as pyplot  
import pickle  
from matplotlib import style  
  
data = pd.read\_csv("student-mat.csv", sep=";")  
print(data.head()) #show data  
data = data[["G1", "G2", "G3", "studytime", "failures", "absences"]]  
  
predict = "G3"  
  
x = np.array(data.drop([predict],1))  
y = np.array(data[predict]) # labels  
  
# x\_train -> x axis, y\_train -> y axis, x\_test -> x의 accuracy, y\_test same  
x\_train, x\_test, y\_train, y\_test = sklearn.model\_selection.train\_test\_split(x, y, test\_size = 0.1) # test\_size means accuracy percentage 10%  
  
best = 0  
for \_ in range(30):  
 x\_train, x\_test, y\_train, y\_test = sklearn.model\_selection.train\_test\_split(x, y, test\_size=0.1) # test\_size means accuracy percentage 10%  
 linear = sklearn.linear\_model.LinearRegression()  
  
 linear.fit(x\_train, y\_train) # x\_train, y\_train data에 가장 잘 맞는 best linear line  
 acc = linear.score(x\_test, y\_test) # accuracy of linear regression  
  
 print('Accuracy: ',acc)  
  
 if acc > best:  
 best = acc  
 with open("studentmodel.pickle", "wb") as f: # Save the model using pickle  
 pickle.dump(linear, f)  
  
pickle\_in = open("studentmodel.pickle", "rb") # Load the saved model  
linear = pickle.load(pickle\_in)  
  
print('Coefficient: \n', linear.coef\_)  
print('Intercept: \n', linear.intercept\_)  
  
predictions = linear.predict(x\_test)  
  
for x in range(len(predictions)):  
 print(predictions[x], x\_test[x], y\_test[x])  
  
p = 'absences' # x axis (using dynamically)  
style.use("ggplot")  
pyplot.scatter(data[p],data["G3"]) # scatter(x axis, y axis)  
pyplot.xlabel(p) # x axis label under plot  
pyplot.ylabel("Final grade") # y axis label  
pyplot.show()

C:\ANACONDA\envs\tensorflow\pythonw.exe "C:/Users/jinwooseol/OneDrive - postech.ac.kr/설진우/Python\_ML/Linear regression.py"

school sex age address famsize Pstatus ... Walc health absences G1 G2 G3

0 GP F 18 U GT3 A ... 1 3 6 5 6 6

1 GP F 17 U GT3 T ... 1 3 4 5 5 6

2 GP F 15 U LE3 T ... 3 3 10 7 8 10

3 GP F 15 U GT3 T ... 1 5 2 15 14 15

4 GP F 16 U GT3 T ... 2 5 4 6 10 10

[5 rows x 33 columns]

Accuracy: 0.807894436200824

Accuracy: 0.7843205554842542

Accuracy: 0.7615494621833754

Accuracy: 0.9101225089939594

Accuracy: 0.8714551615436411

Accuracy: 0.8653852107035681

Accuracy: 0.8349915685650224

Accuracy: 0.841409089128509

Accuracy: 0.9466926767964293

Accuracy: 0.7923260256427213

Accuracy: 0.7361075579696237

Accuracy: 0.8558147605601817

Accuracy: 0.7675834275907143

Accuracy: 0.857240708713933

Accuracy: 0.6780497269727894

Accuracy: 0.8577519975816047

Accuracy: 0.8412165568579743

Accuracy: 0.7328685721089407

Accuracy: 0.8151096109723879

Accuracy: 0.7830524848196634

Accuracy: 0.7696506373443368

Accuracy: 0.9294720217737228

Accuracy: 0.8681272186306919

Accuracy: 0.6867722510277963

Accuracy: 0.8073510083817882

Accuracy: 0.8101190703360209

Accuracy: 0.7516824015851136

Accuracy: 0.7904657086265761

Accuracy: 0.7667822647996144

Accuracy: 0.7423781297829057

Coefficient:

[ 0.1567994 0.98967861 -0.16919597 -0.26212299 0.03765503]

Intercept:

-1.713100360757963

7.158490109119649 [ 8 8 4 0 10] 8

5.341646258579213 [7 6 1 0 5] 7

6.965243067218061 [ 9 7 2 0 18] 6

18.817221295249716 [19 18 2 0 2] 18

15.8093610523579 [14 16 3 0 0] 16

13.021917768518911 [13 13 1 0 0] 12

15.377787272691545 [16 15 2 0 2] 15

20.120536696374135 [18 19 1 0 10] 19

3.9883172542223004 [6 5 2 0 4] 6

10.441124730793424 [10 11 4 0 10] 11

12.785339729130468 [13 12 1 0 20] 12

15.679027499590166 [16 15 2 0 10] 15

9.151164789101728 [10 10 2 1 0] 0

9.24409180901035 [10 10 3 0 0] 9

12.928031856768369 [13 13 2 0 2] 13

12.02605987444883 [12 12 1 0 4] 13

16.36746587932654 [16 16 2 0 2] 17

14.646976787310233 [15 14 2 1 20] 13

10.72420668347689 [13 11 2 1 3] 11

12.928031856768369 [13 13 2 0 2] 11

8.085217233900158 [10 9 4 0 0] 0

11.248242157461943 [ 8 12 1 0 0] 12

8.185320425806417 [8 9 2 0 2] 10

10.028126806152954 [12 10 2 0 8] 11

13.479957396301925 [14 13 1 0 8] 13

8.266809769966155 [9 9 2 0 0] 10

8.41768470622281 [ 7 10 4 1 2] 10

9.759824530334054 [11 10 2 1 12] 10

9.268884944191953 [ 8 10 1 0 0] 11

10.559765785004934 [11 11 2 0 0] 10

7.784014999722171 [7 9 3 0 0] 8

7.492837372405507 [8 9 1 3 0] 10

16.36746587932654 [16 16 2 0 2] 16

12.158104132872255 [12 12 2 0 12] 11

12.671129263977718 [14 13 4 0 0] 14

8.34211982669081 [9 9 2 0 2] 10

11.488241608161765 [12 11 1 0 16] 11

14.988878414198098 [14 15 2 0 0] 16

16.041470509966953 [15 16 3 0 2] 18

-1.1106958924019885 [6 0 2 0 0] 0

Process finished with exit code 0

