LineaRegression-studentGradesPrediction

March 7, 2022

[3]:
```
# Import Libraries
import numpy as np
import pandas as pd
from sklearn import linear_model
import sklearn
from sklearn.utils import shuffle
import matplotlib.pyplot as plt
from matplotlib import style
import pickle

"""Pickle in Python is primarily used in serializing and deserializing a Python␣
 ↪object structure.
 In other words, it's the process of converting a Python object into a byte␣
 ↪stream to store it in a file/database,
 maintain program state across sessions, or transport data over the network."""

style.use("ggplot")

data = pd.read_csv(r"C:
 ↪\Users\nomaniqbal\Downloads\Compressed\student\student-mat.csv", sep=";")

predict = "G3"

data = data[["G1", "G2", "absences", "failures", "studytime", "G3"]]
data = shuffle(data)  # Optional - shuffle the data

x = np.array(data.drop([predict], 1))
y = np.array(data[predict])
x_train, x_test, y_train, y_test = sklearn.model_selection.train_test_split(x,␣
 ↪y, test_size=0.1)

# TRAIN MODEL MULTIPLE TIMES FOR BEST SCORE
best = 0
for _ in range(20):
    x_train, x_test, y_train, y_test = sklearn.model_selection.
 ↪train_test_split(x, y, test_size=0.1)
```

```
    linear = linear_model.LinearRegression()

    linear.fit(x_train, y_train)
    acc = linear.score(x_test, y_test)
    print("Accuracy: " + str(acc))

    if acc > best:
        best = acc
        with open("studentgrades.pickle", "wb") as f:
            pickle.dump(linear, f)

# LOAD MODEL
pickle_in = open("studentgrades.pickle", "rb")
linear = pickle.load(pickle_in)

print("-------------------------")
print('Coefficient: \n', linear.coef_)
print('Intercept: \n', linear.intercept_)
print("-------------------------")

predicted = linear.predict(x_test)
for x in range(len(predicted)):
    print(predicted[x], x_test[x], y_test[x])

# Drawing and plotting model
plot = "failures"
plt.scatter(data[plot], data["G3"])
plt.xlabel(plot)
plt.ylabel("Final Grade")
plt.show()
```

```
Accuracy: 0.7132113378764056
Accuracy: 0.6733837156567337
Accuracy: 0.8539863423485813
Accuracy: 0.7112513381605943
Accuracy: 0.7590408635115957
Accuracy: 0.8369067448407239
Accuracy: 0.7784940822909842
Accuracy: 0.8102354449015575
Accuracy: 0.8572880596768806
Accuracy: 0.8844811271830666
Accuracy: 0.8427072666018267
Accuracy: 0.8336705188444369
Accuracy: 0.9038563162177965
Accuracy: 0.799614009775426
Accuracy: 0.7449890156527281
Accuracy: 0.8454741990415915
```

```
Accuracy: 0.8017085879007526
Accuracy: 0.7412524415508583
Accuracy: 0.803972864098609
Accuracy: 0.9350061561036642
-------------------------
Coefficient:
 [ 0.14722118  0.99274215  0.03684526 -0.28368356 -0.21596008]
Intercept:
 -1.5497199754613966
-------------------------
7.983587539795131 [7 9 0 0 2] 8
8.263341687914675 [ 9  9 15  2  2] 9
10.705337618621169 [11 11  4  0  2] 11
6.907954119193493 [10  8  3  3  1] 7
7.869540569049981 [10  8  6  0  1] 11
12.33259229241584 [11 13  6  0  4] 14
15.486262526438605 [15 15 10  0  2] 15
13.830625408805643 [13 14  0  0  2] 15
14.970588746093764 [14 15  0  0  2] 15
13.058794956212335 [14 13  2  0  2] 13
19.048422096346343 [18 18  8  0  1] 18
16.47373334355016 [16 16  0  0  1] 15
10.705337618621169 [11 11  4  0  2] 11
4.086469297017346 [6 5 6 0 2] 6
12.798006372517447 [13 12 20  0  1] 12
8.221384947070588 [9 9 8 2 1] 9
10.044362658440454 [10 10 17  0  2] 10
6.0032147114253105 [7 7 6 0 3] 7
18.24325757538821 [16 18  0  0  2] 18
15.775593410511835 [17 15  4  0  1] 16
10.679251167142164 [13 11  3  1  2] 11
6.853687308100142 [7 8 8 0 4] 8
17.613856537378336 [16 17  4  0  1] 18
13.762046370317252 [13 14  4  0  3] 14
-1.8806248107342047 [5 0 0 3 1] 0
10.558116436051256 [10 11  4  0  2] 9
13.353716890972336 [13 13 14  0  2] 14
6.92705811457501 [9 8 2 0 4] 8
5.884535829233743 [ 7  6 18  0  1] 6
7.948342569671562 [10  8 14  0  2] 9
11.923623386910693 [15 12  0  0  3] 14
13.132485475767322 [14 13  4  0  2] 13
9.088305906959683 [11  9 14  0  2] 9
8.425570800584987 [8 9 8 0 2] 10
10.347267839519485 [10 11 10  0  4] 11
-0.6566494875289046 [9 0 0 0 2] 0
11.635148062082878 [13 12  2  1  2] 12
11.919151332004331 [11 12 10  0  2] 13
```

```
14.130179255012063 [15 14  6  0  3] 14
10.075936580611295 [11 10  8  0  1] 10
```