Test model UCI

January 10, 2022

[84]:
```
# Import the necessary libraries
import pandas as pd
import numpy as np
import sklearn
from sklearn import linear_model
from sklearn.model_selection import train_test_split
from sklearn.utils import shuffle
import matplotlib.pyplot as plt
from matplotlib import style
```

[85]:
```
# Loading the dataset in Jupyter Notebooks
data = pd.read_csv("student-mat.csv", sep=';')
```

[86]:
```
print (data.head())
```

```
  school sex  age address famsize Pstatus  Medu  Fedu     Mjob      Fjob  … \
0     GP   F   18       U     GT3       A     4     4  at_home   teacher  …
1     GP   F   17       U     GT3       T     1     1  at_home     other  …
2     GP   F   15       U     LE3       T     1     1  at_home     other  …
3     GP   F   15       U     GT3       T     4     2   health  services  …
4     GP   F   16       U     GT3       T     3     3    other     other  …

  famrel freetime  goout  Dalc  Walc health absences  G1  G2  G3
0      4        3      4     1     1      3        6   5   6   6
1      5        3      3     1     1      3        4   5   5   6
2      4        3      2     2     3      3       10   7   8  10
3      3        2      2     1     1      5        2  15  14  15
4      4        3      2     1     2      5        4   6  10  10

[5 rows x 33 columns]
```

[87]:
```
# Selecting multiple columns of the dataset
dataset=data[['G1', 'G2', 'G3', 'studytime', 'failures', 'absences']]
```

[88]:
```
print (dataset.head())
```

```
   G1  G2  G3  studytime  failures  absences
0   5   6   6          2         0         6
```

```
1   5   5   6          2          0          4
2   7   8  10          2          3         10
3  15  14  15          3          0          2
4   6  10  10          2          0          4
```

[89]:
```
# Creating numpy arrays to predict the value G3
predict = 'G3'
```

[90]:
```
X = np.array(dataset.drop([predict], 1))
```

[91]:
```
y = np.array(dataset[predict])
```

[92]:
```
# Splitting the arrays in training and test samples
x_train, x_test, y_train, y_test = sklearn.model_selection.train_test_split(X,␣
↪y, test_size = 0.1)
```

[93]:
```
# Implementing a linear regression
linear = linear_model.LinearRegression()
```

[94]:
```
# Fitting the training data
linear.fit(x_train, y_train)
```

[94]:
```
LinearRegression()
```

[95]:
```
# Predict the accuracy of the model
acc = linear.score(x_test, y_test)
```

[96]:
```
print (acc)
```

```
0.8508688175776227
```

[97]:
```
# Printing coefficients of the 5 variables of 'G3'
print ('Coefficient \n', linear.coef_)
```

```
Coefficient
 [ 0.13445604  0.97916752 -0.28293296 -0.24565735  0.03199837]
```

[98]:
```
# Printing the intercept of the linear regression of the value of 'G3'
print ('Intercept \n', linear.intercept_)
```

```
Intercept
 -1.0365502126319193
```

[99]:
```
# Making predictions based on the value of G3
predictions = linear.predict(x_test)
```

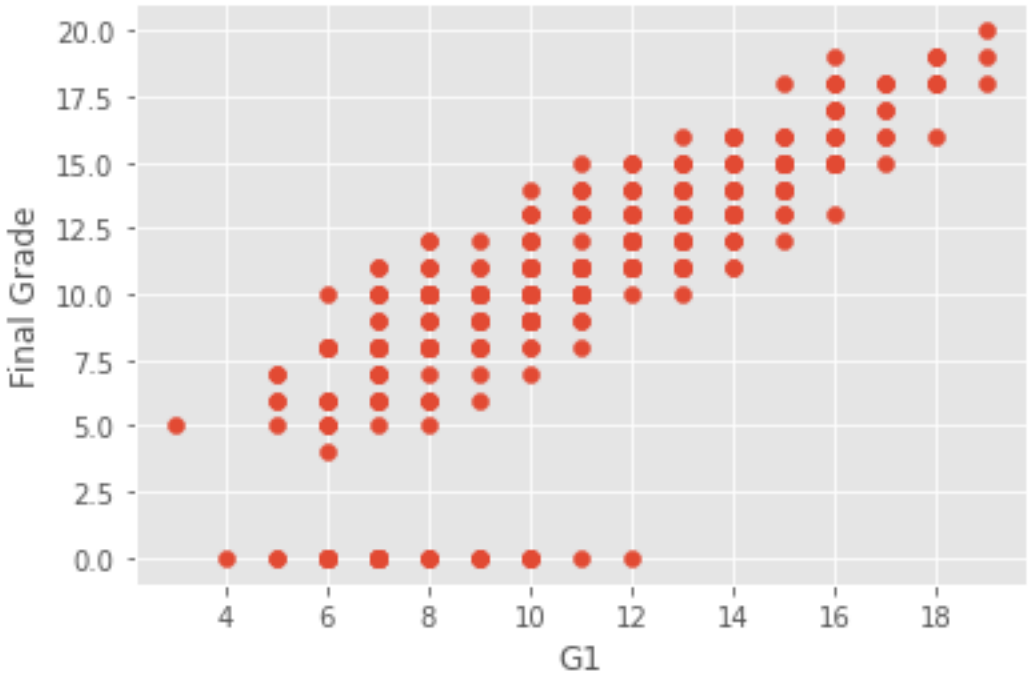
[100]:
```
for x in range(len(predictions)):
    print(predictions[x],  x_test[x], y_test[x])
```

```
7.851959371264607 [7 9 2 2 6] 8
13.883350114359905 [16 14  4  0  6] 15
9.597816222720999 [10 10  2  0  2] 10
16.763318485365822 [17 17  4  0  0] 18
-0.7956798904012308 [6 0 2 0 0] 0
14.517079857757132 [15 14  2  1 20] 13
14.953460365876358 [16 15  3  0  0] 15
7.2174404739242135 [10  8  1  3  3] 7
12.804230866499532 [12 13  2  0  2] 12
11.953056833755095 [12 12  2  0  6] 12
11.337215044810039 [10 12  3  0  4] 12
3.6461186318941685 [6 5 1 3 0] 0
14.897021948604062 [13 15  2  0  2] 16
5.005399541058942 [7 6 1 2 0] 0
15.031477989847534 [14 15  2  0  2] 15
13.903833548045405 [15 14  3  0  2] 15
7.583047288777665 [5 9 2 2 6] 7
8.49570759928302 [10  9  3  0  7] 9
14.136790646945553 [13 14  1  0  0] 13
14.894516006711958 [16 15  4  0  7] 17
7.060690149515054 [ 8  8  4  0 10] 8
11.899389717793467 [12 12  2  1 12] 13
11.542130383453932 [12 12  3  0  2] 11
12.171993052527117 [12 12  1  0  4] 13
6.072684930627917 [ 7  6  1  0 18] 6
15.300390072334476 [16 15  2  0  2] 15
17.74516324557577 [17 17  2  0 13] 17
10.775436528238048 [11 11  2  0  4] 11
8.56867283857555 [8 9 1 0 0] 8
12.299986540213212 [12 12  1  0  8] 12
11.784321337222122 [14 12  2  1  0] 12
7.087636136757931 [8 8 3 0 2] 10
11.825063346069 [12 12  2  0  2] 11
8.87463567766266 [10  9  2  0 10] 10
6.058492752612479 [6 7 2 0 0] 0
5.4967142360404875 [7 6 1 0 0] 0
10.916355123038894 [13 11  2  0  0] 10
11.839084226197128 [10 12  1  0  2] 12
9.01555879665353 [10 10  3  2  8] 10
12.010591024040767 [11 12  2  0 12] 11
```

[101]:
```
# Plotting the correlation between the first grade and the final grade
p = 'G1'
```

[102]:
```
style.use('ggplot')
```

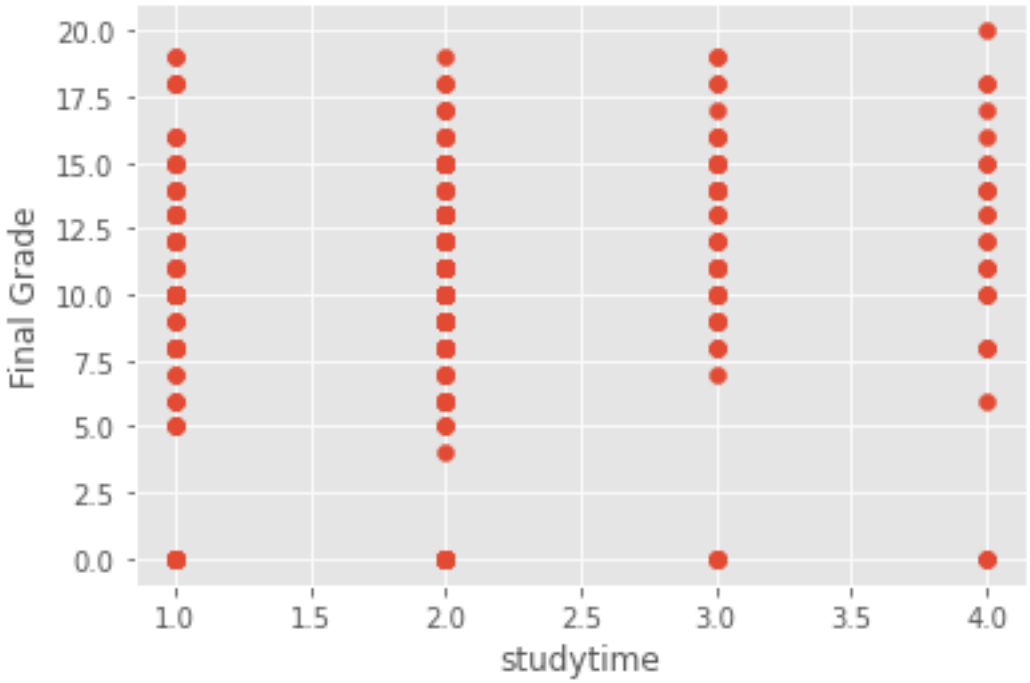
[107]:
```
plt.scatter(data[p], data['G3'])
plt.xlabel(p)
plt.ylabel('Final Grade')
plt.show()
```



[111]:
```
# Plotting the correlation between the studytime and the final grade
p = 'studytime'
```

[112]:
```
style.use('ggplot')
```

[113]:
```
plt.scatter(data[p], data['G3'])
plt.xlabel(p)
plt.ylabel('Final Grade')
plt.show()
```


Final Grade
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
studytime