Student Marks Prediction using Data Science

April 15, 2023

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

[2]:
```
data = pd.read_csv('student_info.csv')
data.head()
```

[2]:

	study_hours	student_marks
0	6.83	78.50
1	6.56	76.74
2	NaN	78.68
3	5.67	71.82
4	8.67	84.19

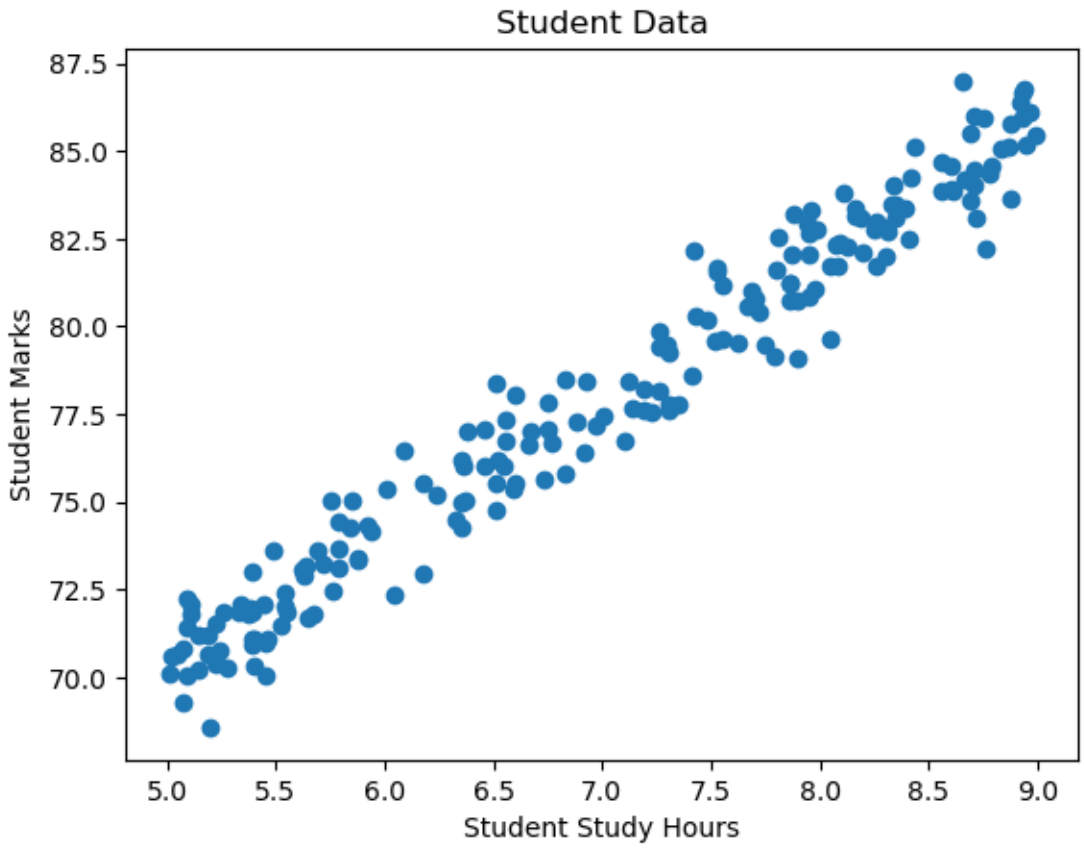
[3]:
```
data.shape
```

[3]:
```
(200, 2)
```

[4]:
```
data.isnull().sum()
```

[4]:
```
study_hours      5
student_marks    0
dtype: int64
```

[5]:
```
plt.scatter(x = data.study_hours , y=data.student_marks)
plt.title("Student Data")
plt.xlabel("Student Study Hours")
plt.ylabel("Student Marks")
plt.show()
```



[6]:
```
data.mean()
```

[6]:
```
study_hours       6.995949
student_marks    77.933750
dtype: float64
```

[7]:
```
data = data.fillna(data.mean())
data.isnull().sum()
```

[7]:
```
study_hours      0
student_marks    0
dtype: int64
```

[8]:
```
X = data.drop(columns = 'student_marks')
y = data.drop(columns = 'study_hours')
X.shape , y.shape
```

[8]:
```
((200, 1), (200, 1))
```

[9]:
```
from sklearn.model_selection import train_test_split
```

[10]:
```
X_train , X_test , y_train , y_test = train_test_split(X, y , random_state=51 ,␣
↪test_size=0.2)
X_train.shape , y_train.shape , X_test.shape , y_test.shape
```

[10]:
```
((160, 1), (160, 1), (40, 1), (40, 1))
```

[11]:
```
from sklearn.linear_model import LinearRegression
```

[12]:
```
lr = LinearRegression()
lr.fit(X_train , y_train)
```

[12]:
```
LinearRegression()
```

[13]:
```
lr.score(X_test , y_test)
```

[13]:
```
0.9514124242154464
```

[14]:
```
lr.intercept_
```

[14]:
```
array([50.44735504])
```

[15]:
```
pred = lr.predict(X_test)
pred
```

[15]:
```
array([[83.11381458],
       [78.9025963 ],
       [84.57003024],
       [85.82946001],
       [84.72745896],
       [80.75238377],
       [72.84159055],
       [71.66087515],
       [73.23516235],
       [71.66087515],
       [73.47130543],
       [76.38373677],
       [73.23516235],
       [73.58937697],
       [82.95638585],
       [70.40144538],
       [73.23516235],
       [78.74516758],
       [75.55723598],
       [82.68088559],
       [76.65923703],
       [70.48015974],
       [74.77009238],
       [77.98143645],
```

```
       [85.59331693],
       [82.56281405],
       [76.42309395],
       [85.0423164 ],
       [78.39095296],
       [81.38209865],
       [81.73631327],
       [83.15317176],
       [82.20859943],
       [81.10659839],
       [73.58937697],
       [71.1492318 ],
       [71.89701823],
       [81.53952737],
       [72.60544747],
       [71.93637541]])
```

[16]:
```
y_test
```

[16]:

	student_marks
148	82.02
104	77.55
4	84.19
7	85.46
192	84.03
160	80.81
118	73.61
58	70.90
190	73.14
174	73.02
23	75.02
10	75.37
115	74.44
86	73.40
67	81.70
68	69.27
177	73.64
171	77.63
128	77.01
14	83.08
82	76.63
50	72.22
45	72.96
31	76.14
176	85.96
21	83.36
198	78.05

```
89          84.60
35          76.76
36          81.24
113         80.86
121         82.69
99          82.30
162         79.17
79          73.34
131         71.86
65          70.06
13          80.76
85          72.87
42          71.10
```

[17]:
```
pd.DataFrame(np.c_[X_test , y_test , pred] ,columns =[ 'Study hours' ,
 ↪'Original Marks' , 'Predicted Marks'])
```

[17]:

	Study hours	Original Marks	Predicted Marks
0	8.300000	82.02	83.113815
1	7.230000	77.55	78.902596
2	8.670000	84.19	84.570030
3	8.990000	85.46	85.829460
4	8.710000	84.03	84.727459
5	7.700000	80.81	80.752384
6	5.690000	73.61	72.841591
7	5.390000	70.90	71.660875
8	5.790000	73.14	73.235162
9	5.390000	73.02	71.660875
10	5.850000	75.02	73.471305
11	6.590000	75.37	76.383737
12	5.790000	74.44	73.235162
13	5.880000	73.40	73.589377
14	8.260000	81.70	82.956386
15	5.070000	69.27	70.401445
16	5.790000	73.64	73.235162
17	7.190000	77.63	78.745168
18	6.380000	77.01	75.557236
19	8.190000	83.08	82.680886
20	6.660000	76.63	76.659237
21	5.090000	72.22	70.480160
22	6.180000	72.96	74.770092
23	6.995949	76.14	77.981436
24	8.930000	85.96	85.593317
25	8.160000	83.36	82.562814
26	6.600000	78.05	76.423094
27	8.790000	84.60	85.042316
28	7.100000	76.76	78.390953

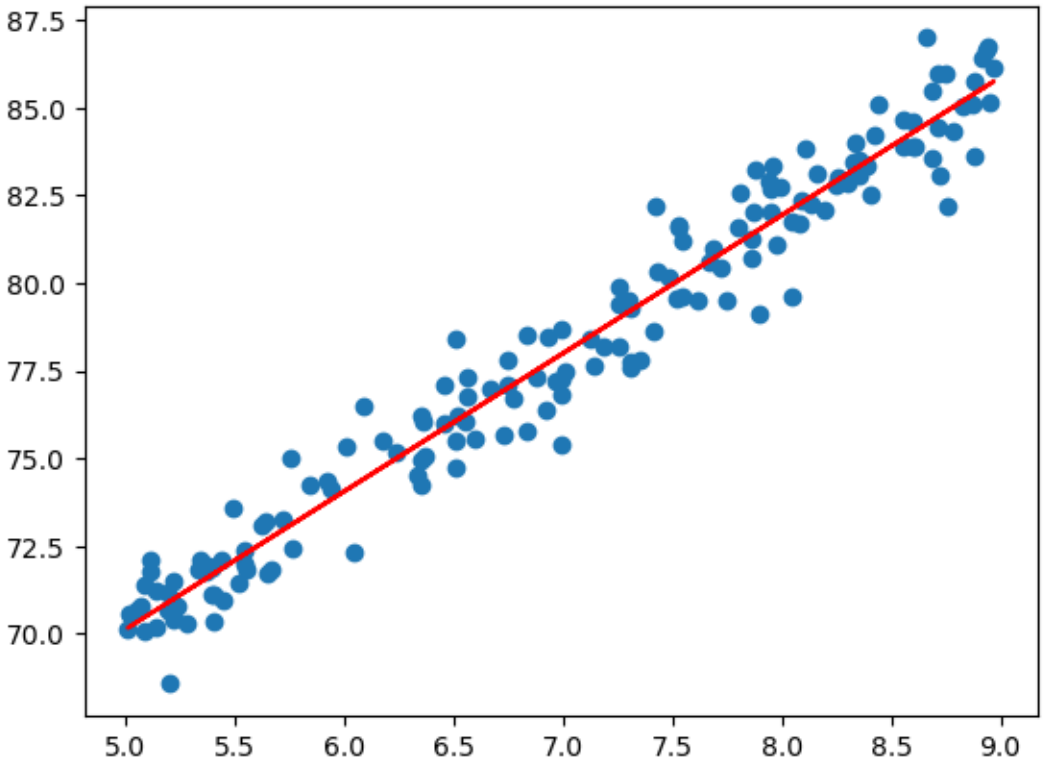
29	7.860000	81.24	81.382099
30	7.950000	80.86	81.736313
31	8.310000	82.69	83.153172
32	8.070000	82.30	82.208599
33	7.790000	79.17	81.106598
34	5.880000	73.34	73.589377
35	5.260000	71.86	71.149232
36	5.450000	70.06	71.897018
37	7.900000	80.76	81.539527
38	5.630000	72.87	72.605447
39	5.460000	71.10	71.936375

[18]:
```
# Fine Tune Model
lr.score(X_test,y_test)
```

[18]: 0.9514124242154464

[19]:
```
plt.scatter(X_train, y_train)
plt.plot(X_train ,lr.predict(X_train) , color='r')
```

[19]: [<matplotlib.lines.Line2D at 0x1c4358f9100>]



[20]:
```
import joblib
```

[21]:
```
joblib.dump(lr , 'Student_Marks_Prediction_Model.pkl')
```

[21]:
```
['Student_Marks_Prediction_Model.pkl']
```

[25]:
```
model = joblib.load('Student_Marks_Prediction_Model.pkl')
```

[26]:
```
model.predict([[ 1 ]])[0][0]
```

```
C:\Users\HP\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does
not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

[26]:
```
54.38307305359076
```

[27]:
```
data.describe()
```

[27]:

	study_hours	student_marks
count	200.000000	200.00000
mean	6.995949	77.93375
std	1.237218	4.92570
min	5.010000	68.57000
25%	5.790000	73.38500
50%	7.002974	77.71000
75%	8.072500	82.32000
max	8.990000	86.99000

[ ]:
```

```