

This source code created by IndianAIProduction.com team

<https://indianaiproduction.com/ml-project-student-mark-prediction/>

Project Tutorial: https://youtu.be/U_oJqcyc0eI

Anaconda Navigator, Jupyter Notebook, Spyder Installation: <https://youtu.be/HlvGIPr4X3Q>

For video tutorial visit our youtube channel

www.youtube.com/IndianAIProduction

Business Probelm

```
In [3]: # importing libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

loading the dataset

```
In [5]: df=pd.read_csv(r"C:\Users\Jan Saida\OneDrive\Documents\Student Marks Predictor\Student mark Predictor Deployment\student_info.csv")
df
```

Out[5]:

	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
...
195	7.53	81.67
196	8.56	84.68
197	8.94	86.75
198	6.60	78.05
199	8.35	83.50

200 rows × 2 columns

In [6]: `df.head()`

Out[6]:

	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

In [7]: `df.tail()`

```
Out[7]:
```

	study_hours	student_marks
195	7.53	81.67
196	8.56	84.68
197	8.94	86.75
198	6.60	78.05
199	8.35	83.50

```
In [8]: df.shape
```

```
Out[8]: (200, 2)
```

Discover and Visualize the data to gain Insights

```
In [10]: df.describe()
```

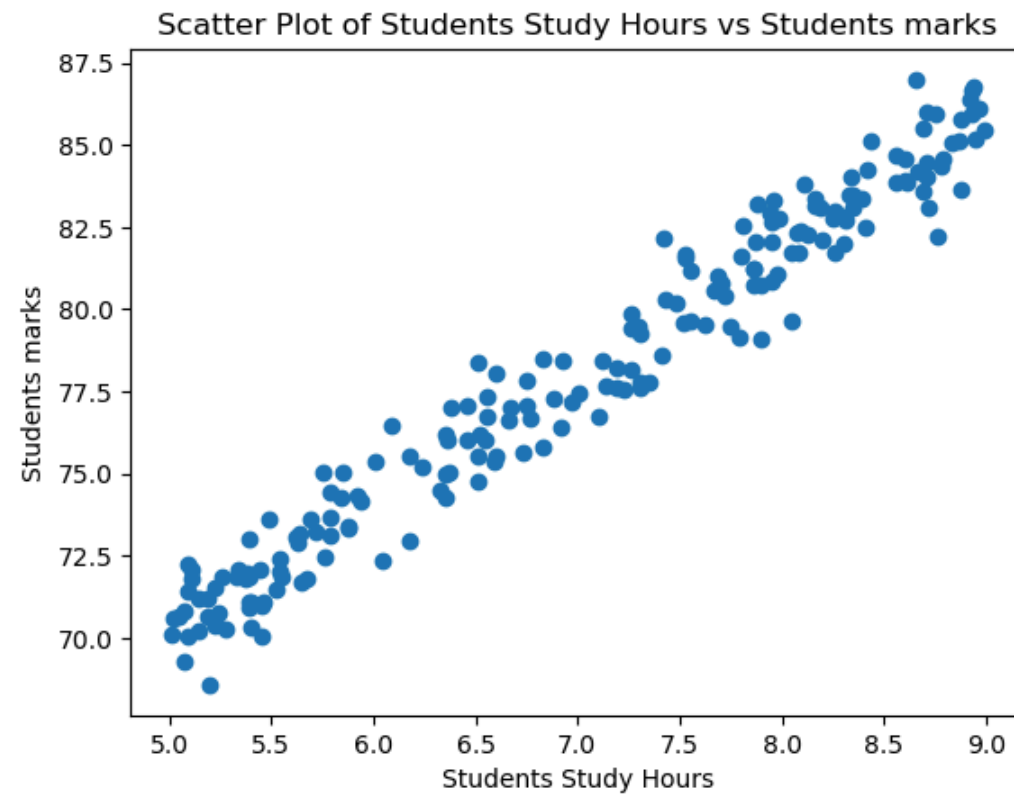
```
Out[10]:
```

	study_hours	student_marks
count	195.000000	200.00000
mean	6.995949	77.93375
std	1.253060	4.92570
min	5.010000	68.57000
25%	5.775000	73.38500
50%	7.120000	77.71000
75%	8.085000	82.32000
max	8.990000	86.99000

```
In [11]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 200 entries, 0 to 199  
Data columns (total 2 columns):  
#   Column      Non-Null Count  Dtype  
---  ---  
0    study_hours  195 non-null    float64  
1    student_marks 200 non-null    float64  
dtypes: float64(2)  
memory usage: 3.3 KB
```

```
In [12]: plt.scatter(x =df.study_hours, y = df.student_marks)  
plt.xlabel("Students Study Hours")  
plt.ylabel("Students marks")  
plt.title("Scatter Plot of Students Study Hours vs Students marks")  
plt.show()
```



Prepare the Data for Machine Learning Algorithms

Data Cleaning

```
In [15]: df.isnull().sum()
```

```
Out[15]: study_hours      5  
student_marks      0  
dtype: int64
```

```
In [16]: df.mean()
```

```
Out[16]: study_hours      6.995949  
student_marks      77.933750  
dtype: float64
```

```
In [17]: df2=df.fillna(df.mean())  
df2
```

```
Out[17]:
```

	study_hours	student_marks
0	6.830000	78.50
1	6.560000	76.74
2	6.995949	78.68
3	5.670000	71.82
4	8.670000	84.19
...
195	7.530000	81.67
196	8.560000	84.68
197	8.940000	86.75
198	6.600000	78.05
199	8.350000	83.50

200 rows × 2 columns

```
In [18]: df2.head()
```

```
Out[18]:
```

	study_hours	student_marks
0	6.830000	78.50
1	6.560000	76.74
2	6.995949	78.68
3	5.670000	71.82
4	8.670000	84.19

```
In [19]: df2.isnull().sum()
```

```
Out[19]: study_hours    0  
student_marks    0  
dtype: int64
```

Splitting the dataset

```
In [21]: X=df2.drop('student_marks',axis='columns')  
X
```

Out[21]:

	study_hours
0	6.830000
1	6.560000
2	6.995949
3	5.670000
4	8.670000
...	...
195	7.530000
196	8.560000
197	8.940000
198	6.600000
199	8.350000

200 rows × 1 columns

```
In [22]: y=df2.drop('study_hours',axis='columns')  
y
```

Out[22]:

student_marks	
0	78.50
1	76.74
2	78.68
3	71.82
4	84.19
...	...
195	81.67
196	84.68
197	86.75
198	78.05
199	83.50

200 rows × 1 columns

```
In [23]: print("shape of X = ", X.shape)
         print("shape of y = ", y.shape)
```

```
shape of X = (200, 1)
shape of y = (200, 1)
```

```
In [24]: from sklearn.model_selection import train_test_split
         X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=51)
```

```
In [25]: X_train
```


Out[25]:

	study_hours
8	5.19
146	7.99
63	7.72
43	8.30
64	8.88
...	...
180	6.97
149	8.97
16	6.36
197	8.94
57	8.93

160 rows × 1 columns

In [26]: X_test

Out[26]:

	study_hours
148	8.300000
104	7.230000
4	8.670000
7	8.990000
192	8.710000
160	7.700000
118	5.690000
58	5.390000
190	5.790000
174	5.390000
23	5.850000
10	6.590000
115	5.790000
86	5.880000
67	8.260000
68	5.070000
177	5.790000
171	7.190000
128	6.380000
14	8.190000
82	6.660000
50	5.090000
45	6.180000
31	6.995949

study_hours	
176	8.930000
21	8.160000
198	6.600000
89	8.790000
35	7.100000
36	7.860000
113	7.950000
121	8.310000
99	8.070000
162	7.790000
79	5.880000
131	5.260000
65	5.450000
13	7.900000
85	5.630000
42	5.460000

In [27]: `y_train`

Out[27]:

student_marks	
8	70.66
146	82.75
63	80.43
43	82.84
64	85.78
...	...
180	77.19
149	86.12
16	76.04
197	86.75
57	86.65

160 rows × 1 columns

In [28]:

```
y_test
```

Out[28]:

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

student_marks	
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

```
In [29]: print("shape of X_train = ", X_train.shape)
print("shape of y_train = ", y_train.shape)
print("shape of X_test = ", X_test.shape)
print("shape of y_test = ", y_test.shape)
```

```
shape of X_train = (160, 1)
shape of y_train = (160, 1)
shape of X_test = (40, 1)
shape of y_test = (40, 1)
```

Select a Model and Train it

```
In [31]: # y = m * x + c

from sklearn.linear_model import LinearRegression
lr = LinearRegression()
```

```
In [32]: lr.fit(X_train,y_train)
```

```
Out[32]: ▼ LinearRegression ⓘ ?
LinearRegression()
```

```
In [33]: lr.coef_
```

```
Out[33]: array([[3.93571802]])
```

```
In [34]: lr.intercept_
```

```
Out[34]: array([50.44735504])
```

```
In [35]: m=3.93
c=50.44
y=m*4+c
y
```

```
Out[35]: 66.16
```

```
In [36]: lr.predict([[4]])[0][0].round(2)
```

```
C:\Users\Jan Saida\anaconda3\Lib\site-packages\sklearn\base.py:493: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

```
Out[36]: 66.19
```

```
In [37]: y_pred=lr.predict(X_test)
y_pred
```

```
Out[37]: 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]])
```

```
In [38]: pd.DataFrame(np.c_[X_test, y_test, y_pred], columns = ["study_hours", "student_marks_original", "student_marks_predicted"])
```


Out[38]:

	study_hours	student_marks_original	student_marks_predicted
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

	study_hours	student_marks_original	student_marks_predicted
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

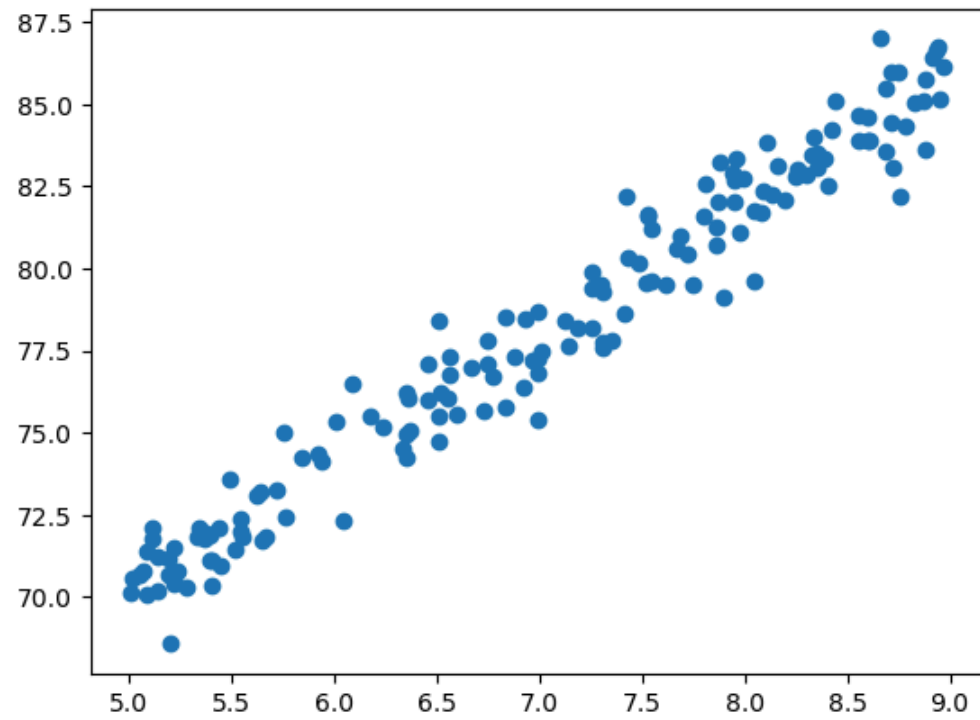
Fine-Tune your model

```
In [40]: lr.score(X_test,y_test)
```

```
Out[40]: 0.9514124242154464
```

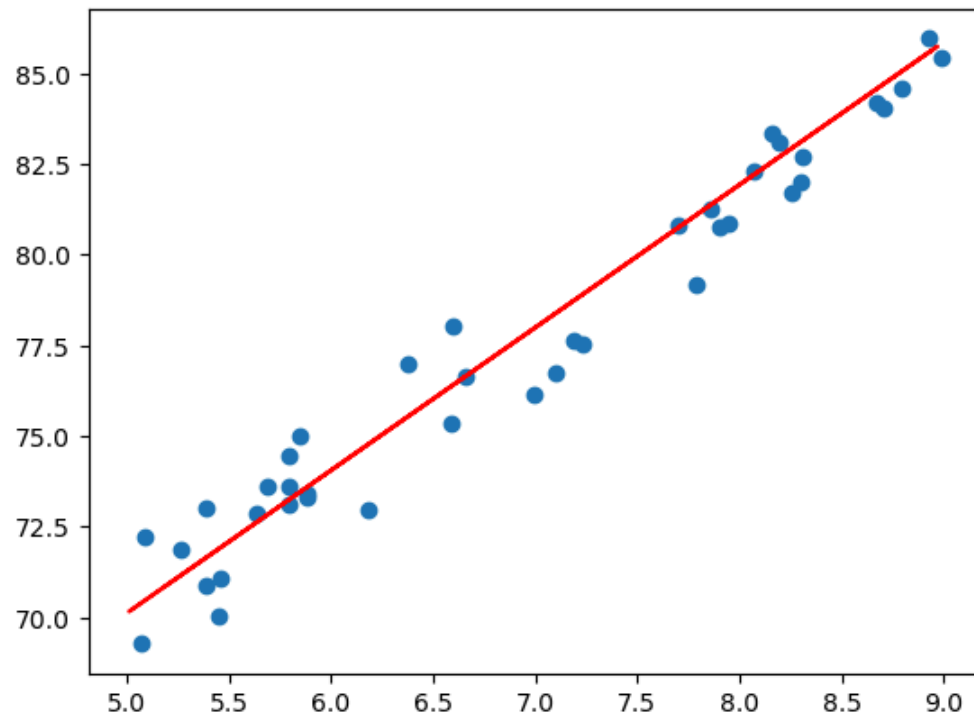
```
In [41]: plt.scatter(X_train,y_train)
```

```
Out[41]: <matplotlib.collections.PathCollection at 0x1b5177bd3a0>
```



```
In [42]: plt.scatter(X_test, y_test)
plt.plot(X_train, lr.predict(X_train), color = "r")
```

```
Out[42]: [<matplotlib.lines.Line2D at 0x1b51a22dac0>]
```



Present your Model

Save ML Model

```
In [45]: import joblib
joblib.dump(lr, 'student_mark_predictor.pkl')
```

```
Out[45]: ['student_mark_predictor.pkl']
```

```
In [49]: model = joblib.load("student_mark_predictor.pkl")
```

```
In [51]: model.predict([[5]])[0][0]
```

```
C:\Users\Jan Saida\anaconda3\Lib\site-packages\sklearn\base.py:493: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

```
Out[51]: 70.12594512018406
```

```
In [53]: import os  
os.getcwd()
```

```
Out[53]: 'C:\\Users\\Jan Saida'
```

```
In [ ]:
```

Flask\Student_Marks_Prediction_app.py

```
1 import numpy as np
2 import pandas as pd
3 from flask import Flask, request, render_template
4 import joblib
5 import os
6
7 # Set up the Flask app and template folder
8 app = Flask(__name__, template_folder='C:/Users/Jan Saida/OneDrive/Documents/Students Marks Prediction Project/templates')
9
10 # Load the trained model
11 model = joblib.load(r"C:/Users/Jan Saida/OneDrive/Documents/Students Marks Prediction Project/student_mark_predictor.pkl")
12
13 # Create an empty DataFrame for saving prediction data
14 df = pd.DataFrame()
15
16 # Define the home route
17 @app.route('/')
18 def home():
19     return render_template("index.html")
20
21 # Define the predict route
22 @app.route('/predict', methods=['POST'])
23 def predict():
24     global df
25
26     # Collect input data from the form
27     input_features = [int(x) for x in request.form.values()]
28     features_value = np.array(input_features)
29
30     # Validate the input hours
31     if input_features[0] < 0 or input_features[0] > 24:
32         return render_template('index.html', prediction_text='Please enter valid hours between 1 and 24 if you live on the Earth')
33
34     # Predict the output based on the input features
35     output = model.predict([features_value])[0][0].round(2)
36
37     # Save the input data and predicted output into the DataFrame
38     df = pd.concat([df, pd.DataFrame({'Study Hours': input_features, 'Predicted Output': [output]})], ignore_index=True)
39
```

```
40     # Print and save the DataFrame to a CSV file
41     print(df)
42     df.to_csv('smp_data_from_app.csv')
43
44     # Render the result back to the user
45     return render_template("index.html", prediction_text=f'You will get [{output}%] marks, when you study [{input_features[0]}] hours per day')
46
47 # Run the Flask app
48 if __name__ == '__main__':
49     app.run(host='127.0.0.1', port=5000)
50
```

~\AppData\Roaming\Code\logs\20250205T190658\window1\exthost\output_logging_20250205T190659\13-Code.log

```
1 [Running] python -u "c:\VS Code\Flask\Student_Marks_Prediction_app.py"
2 C:\Users\Jan Saida\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\base.py:380: InconsistentVersionWarning: Trying to unpickle estimator
  LinearRegression from version 1.4.2 when using version 1.6.0. This might lead to breaking code or invalid results. Use at your own risk. For more info
  please refer to:
3 https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
4 warnings.warn(
5 * Serving Flask app 'Student_Marks_Prediction_app'
6 * Debug mode: off
7 WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
8 * Running on http://127.0.0.1:5000
9 Press CTRL+C to quit
10 127.0.0.1 - - [05/Feb/2025 19:07:32] "GET / HTTP/1.1" 200 -
11 127.0.0.1 - - [05/Feb/2025 19:07:32] "GET /static/images/college_banner2.png HTTP/1.1" 404 -
12 127.0.0.1 - - [05/Feb/2025 19:07:32] "GET /favicon.ico HTTP/1.1" 404 -
13 C:\Users\Jan Saida\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\utils\validation.py:2739: UserWarning: X does not have valid feature
  names, but LinearRegression was fitted with feature names
14 warnings.warn(
15 Study Hours Predicted Output
16 0 5 70.13
17 127.0.0.1 - - [05/Feb/2025 19:07:40] "POST /predict HTTP/1.1" 200 -
18 127.0.0.1 - - [05/Feb/2025 19:07:40] "GET /static/images/college_banner2.png HTTP/1.1" 404 -
19 C:\Users\Jan Saida\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\utils\validation.py:2739: UserWarning: X does not have valid feature
  names, but LinearRegression was fitted with feature names
20 warnings.warn(
21 Study Hours Predicted Output
22 0 5 70.13
23 1 6 74.06
24 127.0.0.1 - - [05/Feb/2025 19:07:59] "POST /predict HTTP/1.1" 200 -
25 127.0.0.1 - - [05/Feb/2025 19:07:59] "GET /static/images/college_banner2.png HTTP/1.1" 404 -
26 C:\Users\Jan Saida\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\utils\validation.py:2739: UserWarning: X does not have valid feature
  names, but LinearRegression was fitted with feature names
27 warnings.warn(
28 Study Hours Predicted Output
29 0 5 70.13
30 1 6 74.06
31 2 7 78.00
32 127.0.0.1 - - [05/Feb/2025 19:08:11] "POST /predict HTTP/1.1" 200 -
33 127.0.0.1 - - [05/Feb/2025 19:08:11] "GET /static/images/college_banner2.png HTTP/1.1" 404 -
```




```
34 C:\Users\Jan Saida\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\utils\validation.py:2739: UserWarning: X does not have valid feature
names, but LinearRegression was fitted with feature names
35     warnings.warn(
36         Study Hours    Predicted Output
37 0          5          70.13
38 1          6          74.06
39 2          7          78.00
40 3          8          81.93
41 127.0.0.1 - - [05/Feb/2025 19:08:26] "POST /predict HTTP/1.1" 200 -
42 127.0.0.1 - - [05/Feb/2025 19:08:26] "GET /static/images/college_banner2.png HTTP/1.1" 404 -
43 C:\Users\Jan Saida\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\utils\validation.py:2739: UserWarning: X does not have valid feature
names, but LinearRegression was fitted with feature names
44     warnings.warn(
45         Study Hours    Predicted Output
46 0          5          70.13
47 1          6          74.06
48 2          7          78.00
49 3          8          81.93
50 4          9          85.87
51 127.0.0.1 - - [05/Feb/2025 19:08:31] "POST /predict HTTP/1.1" 200 -
52 127.0.0.1 - - [05/Feb/2025 19:08:31] "GET /static/images/college_banner2.png HTTP/1.1" 404 -
53 C:\Users\Jan Saida\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\utils\validation.py:2739: UserWarning: X does not have valid feature
names, but LinearRegression was fitted with feature names
54     warnings.warn(
55         Study Hours    Predicted Output
56 0          5          70.13
57 1          6          74.06
58 2          7          78.00
59 3          8          81.93
60 4          9          85.87
61 5          10         89.80
62 127.0.0.1 - - [05/Feb/2025 19:08:35] "POST /predict HTTP/1.1" 200 -
63 127.0.0.1 - - [05/Feb/2025 19:08:35] "GET /static/images/college_banner2.png HTTP/1.1" 404 -
64 C:\Users\Jan Saida\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\utils\validation.py:2739: UserWarning: X does not have valid feature
names, but LinearRegression was fitted with feature names
65     warnings.warn(
66         Study Hours    Predicted Output
67 0          5          70.13
68 1          6          74.06
69 2          7          78.00
70 3          8          81.93
```

```
71 4          9          85.87
72 5          10         89.80
73 6          11         93.74
74 127.0.0.1 - - [05/Feb/2025 19:08:39] "POST /predict HTTP/1.1" 200 -
75 127.0.0.1 - - [05/Feb/2025 19:08:39] "GET /static/images/college_banner2.png HTTP/1.1" 404 -
76 C:\Users\Jan Saida\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\utils\validation.py:2739: UserWarning: X does not have valid feature
names, but LinearRegression was fitted with feature names
77     warnings.warn(
78         Study Hours   Predicted Output
79 0          5          70.13
80 1          6          74.06
81 2          7          78.00
82 3          8          81.93
83 4          9          85.87
84 5          10         89.80
85 6          11         93.74
86 7          12         97.68
87 127.0.0.1 - - [05/Feb/2025 19:08:43] "POST /predict HTTP/1.1" 200 -
88 127.0.0.1 - - [05/Feb/2025 19:08:43] "GET /static/images/college_banner2.png HTTP/1.1" 404 -
89
```


This is How My Frontend App will run

Student Mark Predictor ML App

127.0.0.1:5000/predict



Oxstandhard University



*** Enter Your Study Hours to Predict Marks ***


study hours

Predict Marks


You will get [78.0%] marks, when you study [7] hours per day

Student Mark Predictor ML App

127.0.0.1:5000/predict



Oxstandhard University



*** Enter Your Study Hours to Predict Marks ***

study hours

Predict Marks

You will get [89.8%] marks, when you study [10] hours per day

This is how the Excel/CSV sheet will be created in the Backend

df - DataFrame

Index	Study Hours	Predicted Output
0	5	70.13
1	6	74.06
2	7	78
3	8	81.93
4	9	85.87
5	10	89.8
6	11	93.74
7	12	97.68