Student_Marks_Predictor

```
# -*- coding: utf-8 -*-


import numpy as np
import pandas as pd
from flask import Flask, request, render_template
import joblib


app = Flask(__name__)
model = joblib.load(r'C:\Users\samua\Python pickle\student_mark_predictor.pkl')


# Use a global DataFrame for storing predictions
df = pd.DataFrame(columns=['Study Hours', 'Predicted Output'])


@app.route('/')
def home():
    return render_template('index.html')


@app.route('/predict', methods=['POST'])
def predict():
    global df


    try:
        input_features = [float(x) for x in request.form.values()]
    except ValueError:
        return render_template('index.html', prediction_text='Please enter numeric values only.')
    features_value = np.array(input_features).reshape(1, -1)

    # Validate study hours
    if input_features[0] < 0 or input_features[0] > 24:
        return render_template(
            'index.html',
            prediction_text='Please enter valid hours between 1 to 24 if you live on the Earth'
        )

    # Predict using the loaded model
    output = model.predict(features_value)
    if isinstance(output, np.ndarray):
        output = output.flatten()[0]
    output = round(float(output), 2)

    # Store inputs and prediction in the DataFrame
    new_row = pd.DataFrame({'Study Hours': [input_features[0]], 'Predicted Output': [output]})
    df = pd.concat([df, new_row], ignore_index=True)
    print(df)
    df.to_csv('smp_data_from_app.csv', index=False)

    return render_template(
        'index.html',
        prediction_text=f'You will get [{output}%] marks when you study [{int(input_features[0])}] hours per day.'
    )
```

```
if __name__ == "__main__":
    app.run(host='127.0.0.1', port=5000, debug=True)
```