Name: Meghna Ijjapureddy
Batch Code: LISUM20
Submission Date: 28th April 2023
Submitted to: Data Glacier

Data set contains PCOS data (KOTTARATHIL, n.d.)

Sample data:

	Sl. No	Patient File No.	PCOS (Y/N)	I beta-HCG(mIU/mL)	II beta-HCG(mIU/mL)	AMH(ng/mL)
0	1	10001	0	1.99	1.99	2.07
1	2	10002	0	60.80	1.99	1.53
2	3	10003	1	494.08	494.08	6.63
3	4	10004	0	1.99	1.99	1.22
4	5	10005	0	801.45	801.45	2.26

Data frame after data cleaning:

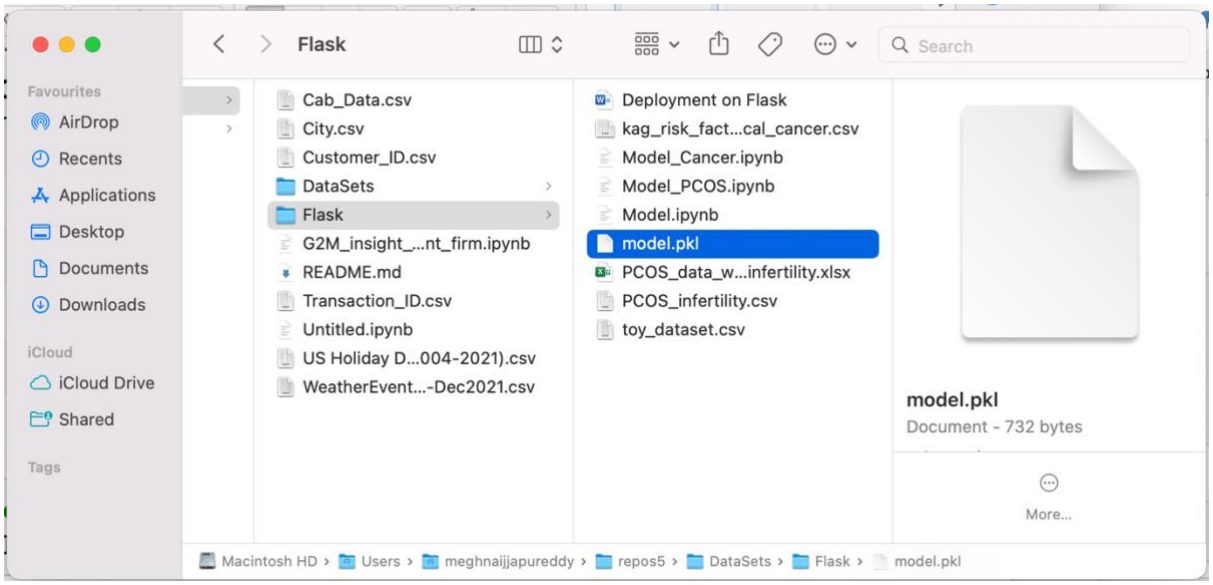
	PCOS (Y/N)	I beta-HCG(mIU/mL)	II beta-HCG(mIU/mL)	AMH(ng/mL)
0	0	1.99	1.99	2.07
1	0	60.80	1.99	1.53
2	1	494.08	494.08	6.63
3	0	1.99	1.99	1.22
4	0	801.45	801.45	2.26

Saving and loading the model:

```
import pickle
pickle.dump(PCOS_model, open('model.pkl','wb'))
```

```
# Loading model to compare the results
PCOS_model = pickle.load(open('model.pkl','rb'))
print(PCOS_model.predict([[60.80, 1.99, 1.53]]))
```

```
[0]
```



Flask Deployment:

Users > meghnaijjapureddy > Desktop > Python > TestFlask.py > ...

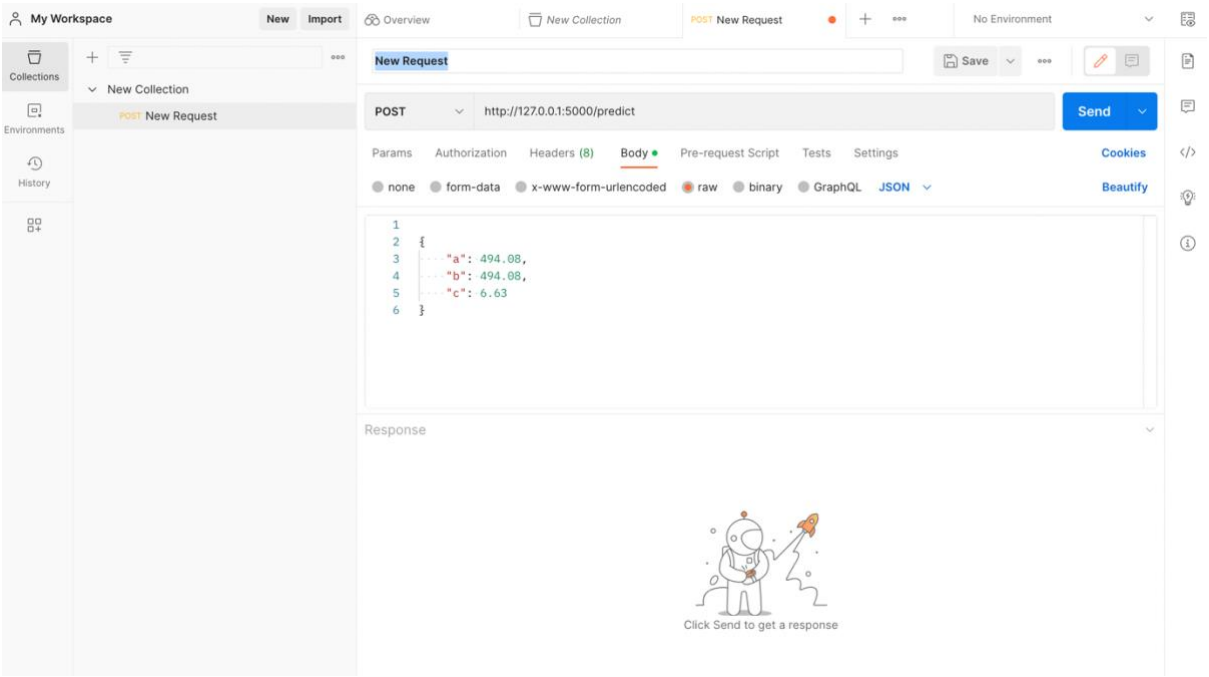
```python
from flask import Flask, request, json
import pickle
import sklearn
from sklearn.linear_model import LogisticRegression
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn.impute import SimpleImputer
from flask_cors import CORS

app = Flask(__name__)
CORS(app)
#app = Flask(__name__)

@app.route('/predict', methods=['POST']) #http:

def predict():
    PCOS_model = pickle.load(open('/Users/meghnaijjapureddy/repos5/DataSets/Flask/model.pkl','rb'))
    input = []
    for key,value in json.loads(request.data).items():
        input.append(value)
    return (PCOS_model.predict([input]).tolist())


```

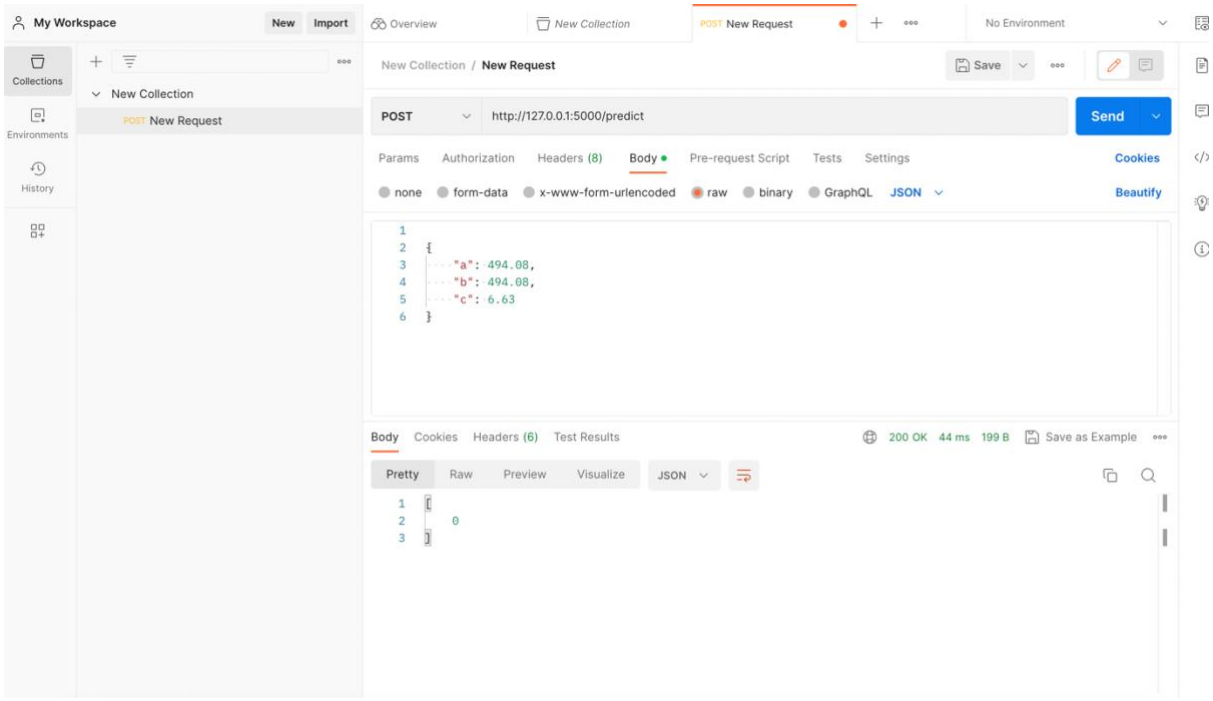
```
Users > meghnaijjapureddy > Desktop > Python > TestFlask.py > ...
1   from flask import Flask, request, json
2   import pickle
3   import sklearn
4   from sklearn.linear_model import LogisticRegression
5   from sklearn import preprocessing
6   from sklearn.model_selection import train_test_split
7   from sklearn.impute import SimpleImputer
8   from flask_cors import CORS
9
10  app = Flask(__name__)
11  CORS(app)
12  #app = Flask(__name__)
13
14  @app.route('/predict', methods=['POST']) #http:
15
16  def predict():
17      PCOS_model = pickle.load(open('/Users/meghnaijjapureddy/repos5/DataSets/Flask/model.pkl','rb'))
18      input = []
19      for key,value in json.loads(request.data).items():
20          input.append(value)
21      return (PCOS_model.predict([input]).tolist())
22
23
24
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL Python

```
(base) meghnaijjapureddy@Meghnas-MacBook-Air repos2 % flask --app /Users/meghnaijjapureddy/Desktop/Python/TestFlask.py run
 * Serving Flask app '/Users/meghnaijjapureddy/Desktop/Python/TestFlask.py'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

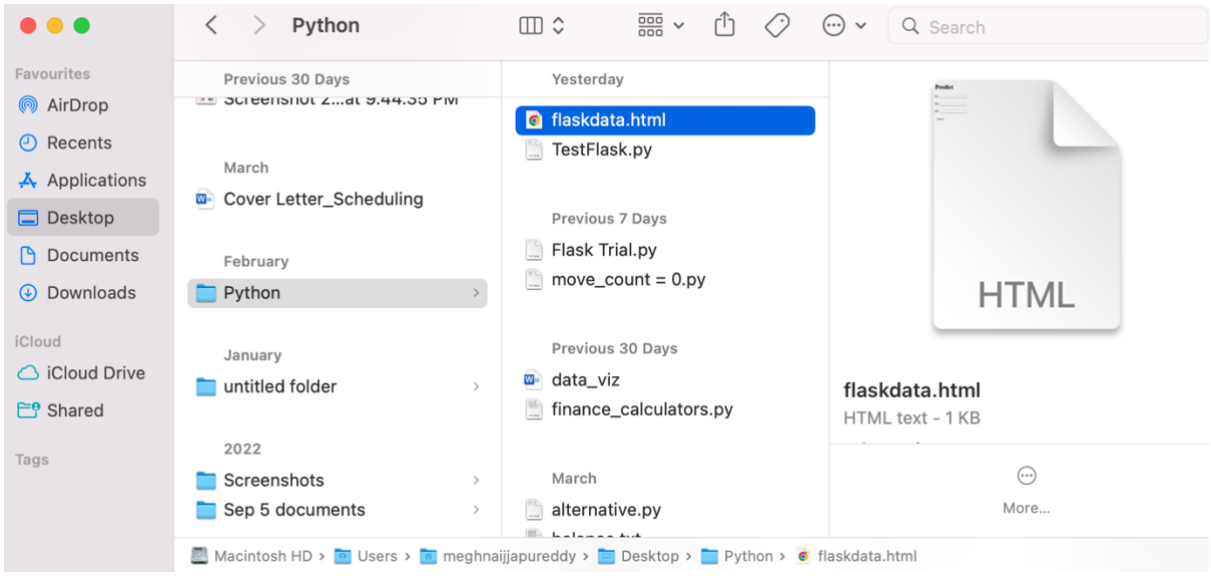
Providing the input in the Postman:

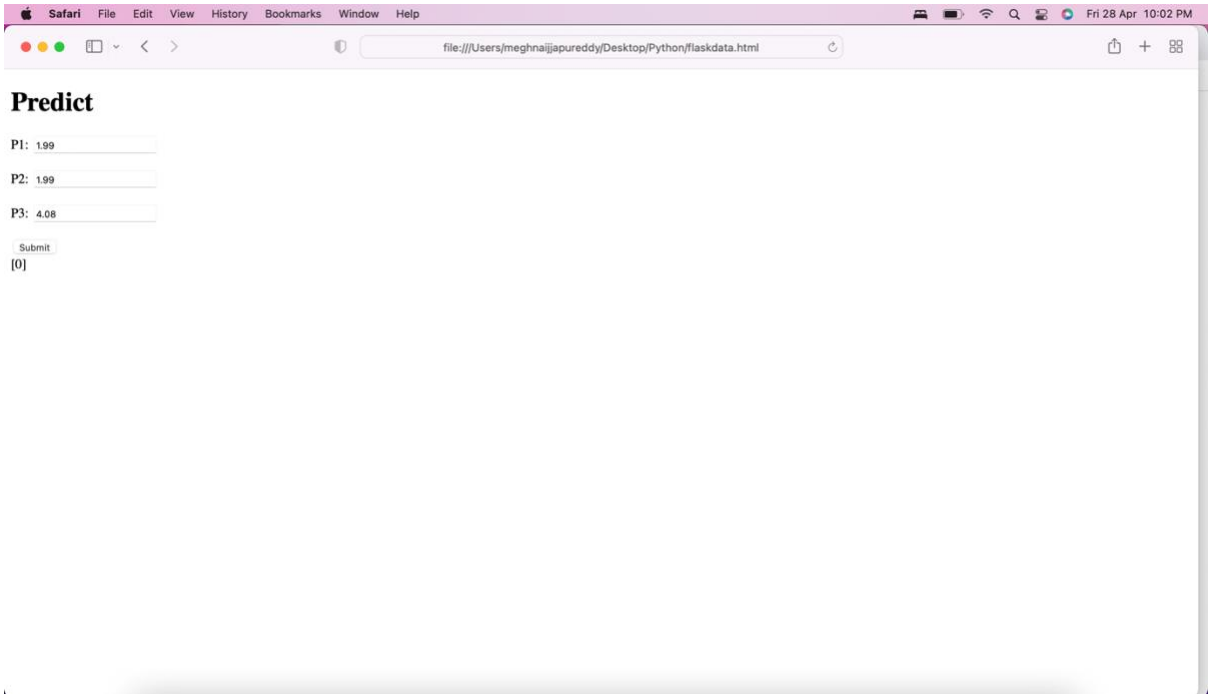




In the web:

```html
<!DOCTYPE html>
<html>
<head>
    <title>Project</title>
</head>
<body>
    <h1>Predict</h1>
    <form>
        <label for="p1">P1:</label>
        <input type="text" id="p1" name="p1"><br><br>
        <label for="p2">P2:</label>
        <input type="text" id="p2" name="p2"><br><br>
        <label for="p3">P3:</label>
        <input type="text" id="p3" name="p3"><br><br>
        <input type="button" value="Submit" onclick="sendData()">
    </form>
    <div id="response"></div>
    <script>
        function sendData() {
            const p1 = document.getElementById("p1").value;
            const p2 = document.getElementById("p2").value;
            const p3 = document.getElementById("p3").value;

            const data = JSON.stringify({ "a": parseFloat(p1), "b": parseFloat(p2), "c": parseFloat(p3) });
            console.log(data)

            fetch('http://127.0.0.1:5000/predict', {
                method: 'POST',
                body: data,
                headers: {
                    'Content-Type': 'application/json'
                }
            })
            .then(response => response.json())
            .then(data => {
                console.log('Success:', data);
                const responseDiv = document.getElementById("response");
                responseDiv.innerHTML = JSON.stringify(data);
            })
            .catch((error) => {
                console.error('Error:', error);
            });
        }
    </script>
</body>
</html>
```


Python
Search
Favourites
AirDrop
Recents
Applications
Desktop
Documents
Downloads
iCloud
iCloud Drive
Shared
Tags
Previous 30 Days
March
Cover Letter_Scheduling
February
Python
January
untitled folder
2022
Screenshots
Sep 5 documents
Yesterday
flaskdata.html
TestFlask.py
Previous 7 Days
Flask Trial.py
move_count = 0.py
Previous 30 Days
data_viz
finance_calculators.py
March
alternative.py
HTML
flaskdata.html
HTML text - 1 KB
More...
Macintosh HD › Users › meghnaijjapureddy › Desktop › Python › flaskdata.html


Safari File Edit View History Bookmarks Window Help
Fri 28 Apr 10:02 PM
file:///Users/meghnaijjapureddy/Desktop/Python/flaskdata.html
Predict
P1: 1.99
P2: 1.99
P3: 4.08
Submit
[0]