Model Deployment on Flask

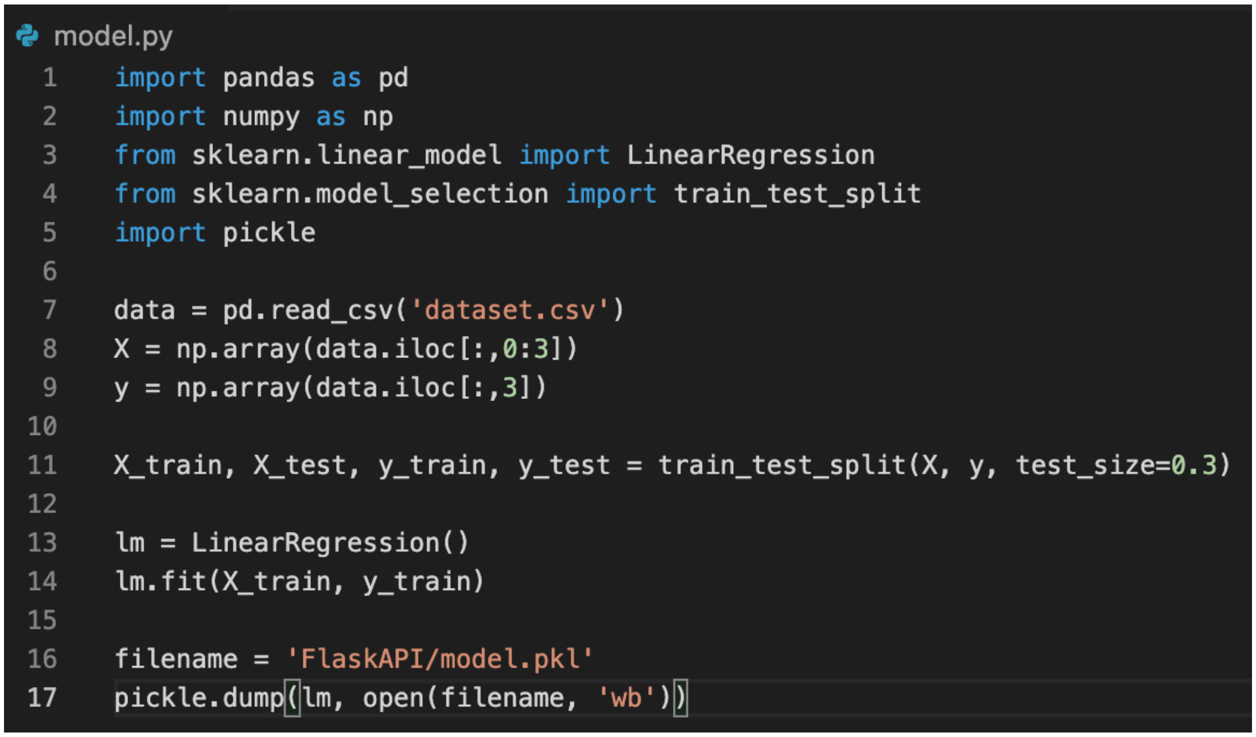
Udbhav Balaji
LISUM01

The Dataset:
The dataset used for this deliverable can be found at the URL
https://www.kaggle.com/anubhabswain/brain-weight-in-humans.

It gives us information regarding a person's brain weight given other features like gender, age and head size (in cm^3).

The Model:
Since this deliverable focussed more on deployment of the model rather than performance of the model, I have used a simple Linear Regression Model to predict the weight of the brain given user-entered attributes.

Below is the code used to train and serialize the model into a pickle file. Pickle was used as it is quite simple to use and understand.

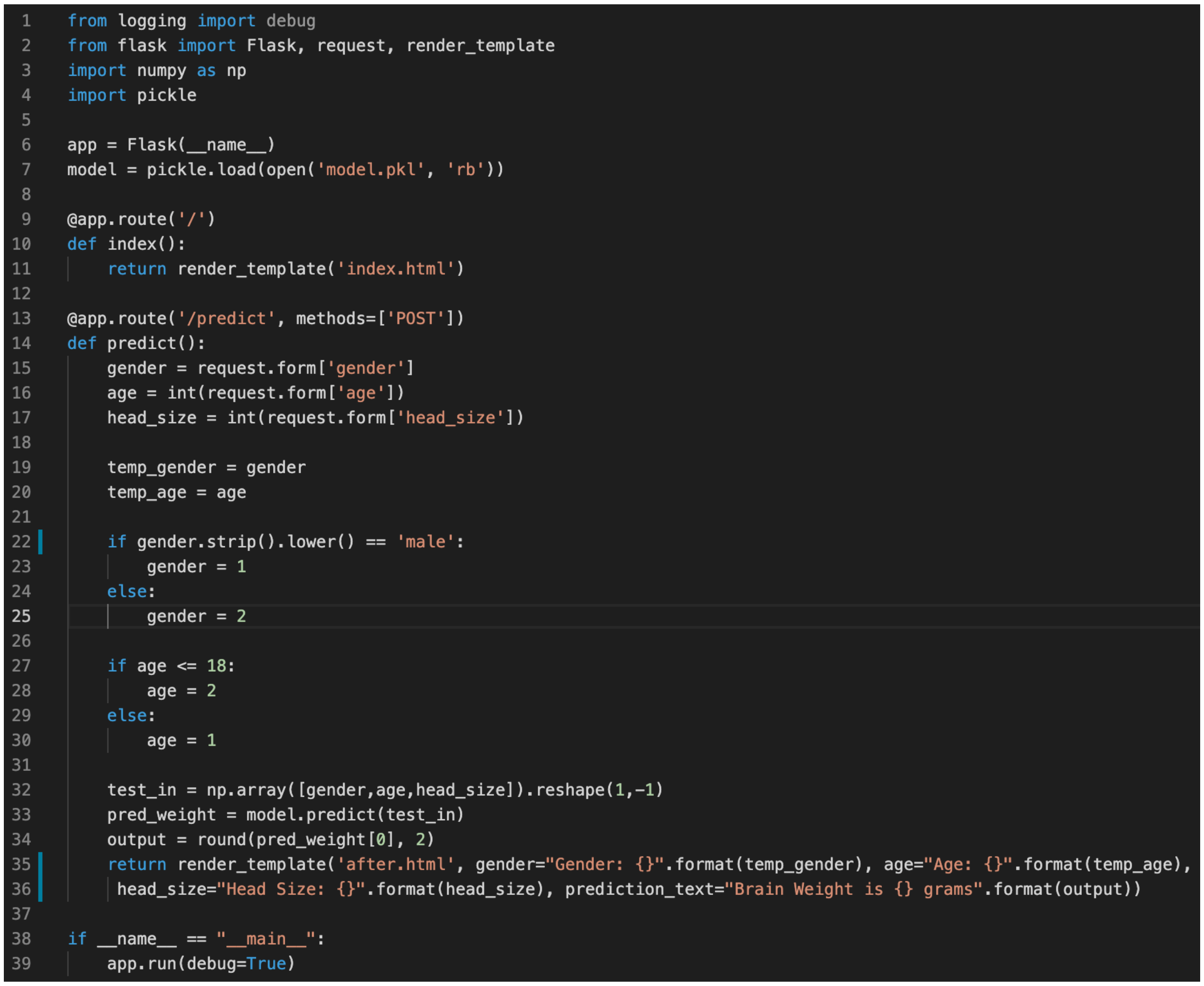


model.py

```python
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
import pickle

data = pd.read_csv('dataset.csv')
X = np.array(data.iloc[:,0:3])
y = np.array(data.iloc[:,3])

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)

lm = LinearRegression()
lm.fit(X_train, y_train)

filename = 'FlaskAPI/model.pkl'
pickle.dump(lm, open(filename, 'wb'))
```

Model Deployment:
Once we have our model stored in the pickle file, we can now go on and deploy the model using Flask. It is in this step where we de-serialize the model back into a python object so that we can send some unseen data into our model, through our webpage, and predict an output.

The python script for the deployment of our model is shown below.

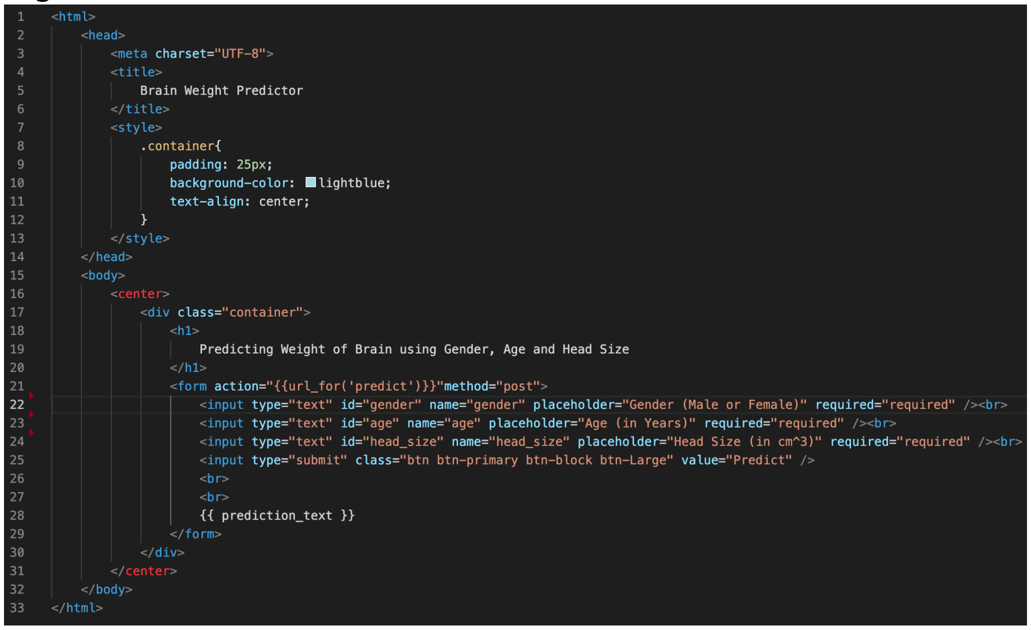


```python
from logging import debug
from flask import Flask, request, render_template
import numpy as np
import pickle

app = Flask(__name__)
model = pickle.load(open('model.pkl', 'rb'))

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

@app.route('/predict', methods=['POST'])
def predict():
    gender = request.form['gender']
    age = int(request.form['age'])
    head_size = int(request.form['head_size'])

    temp_gender = gender
    temp_age = age

    if gender.strip().lower() == 'male':
        gender = 1
    else:
        gender = 2

    if age <= 18:
        age = 2
    else:
        age = 1

    test_in = np.array([gender,age,head_size]).reshape(1,-1)
    pred_weight = model.predict(test_in)
    output = round(pred_weight[0], 2)
    return render_template('after.html', gender="Gender: {}".format(temp_gender), age="Age: {}".format(temp_age),
     head_size="Head Size: {}".format(head_size), prediction_text="Brain Weight is {} grams".format(output))

if __name__ == "__main__":
    app.run(debug=True)
```

Here, since our training data has classified 'Male' as 1 and 'Female' as 2, it isn't intuitive that the user must enter these values. Instead, the webpage takes in the data as Male/Female and the script transforms it into the required form that our model needs to give an output. Similarly, for age, all ages > 18 are categorized as 1 while ages <= 18 are categorized as 2. A similar step was done to prepare the user-inputted data for our model.

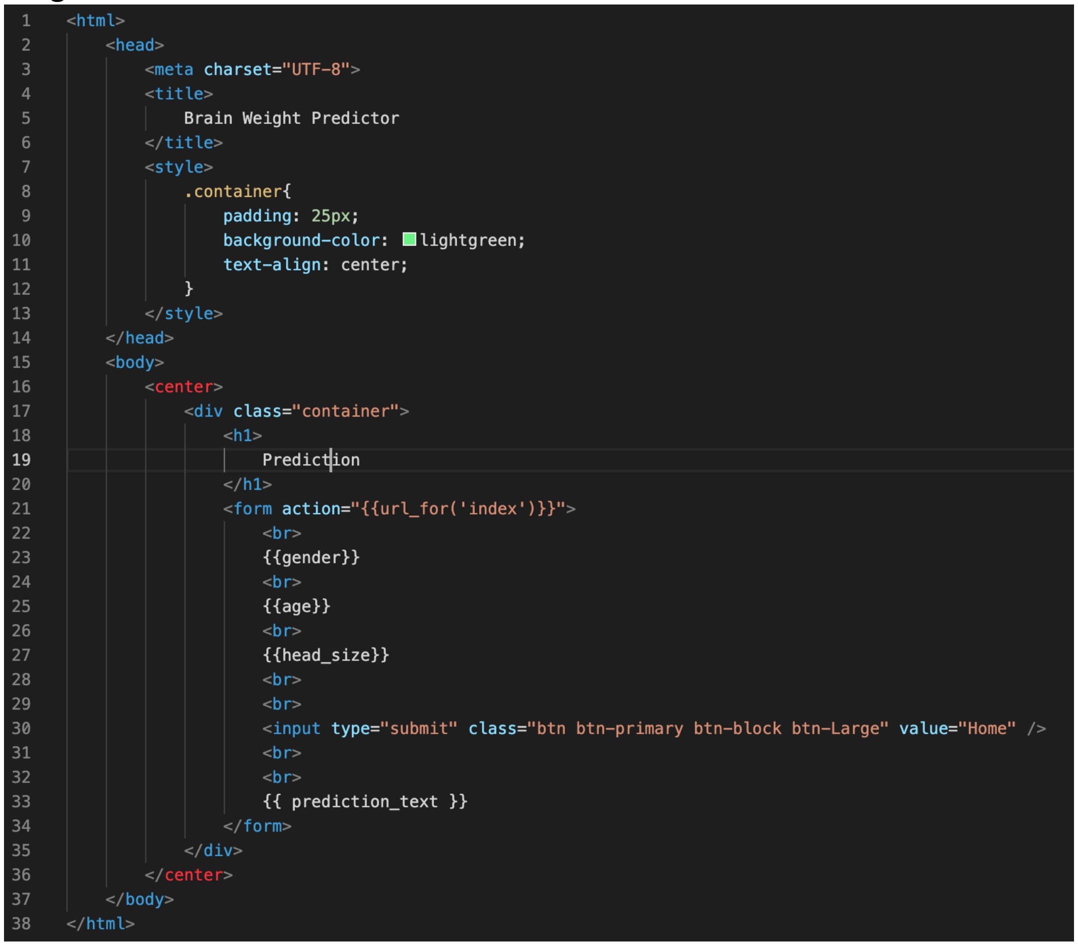
HTML Webpage:
Now that our Flask app is ready, we needed an interface to interact with the user and get the data needed to perform predictions. For this, an HTML was used. Below is the HTML code written to generate the page.

Landing Page Source Code:



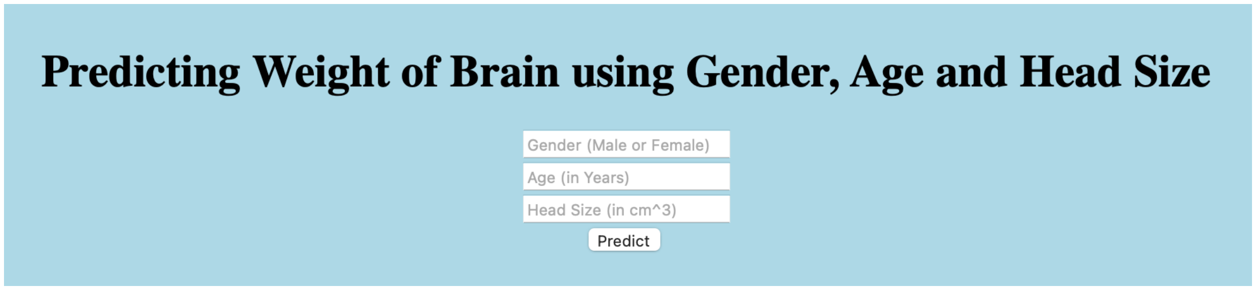
```html
<html>
    <head>
        <meta charset="UTF-8">
        <title>
            Brain Weight Predictor
        </title>
        <style>
            .container{
                padding: 25px;
                background-color: lightblue;
                text-align: center;
            }
        </style>
    </head>
    <body>
        <center>
            <div class="container">
                <h1>
                    Predicting Weight of Brain using Gender, Age and Head Size
                </h1>
                <form action="{{url_for('predict')}}"method="post">
                    <input type="text" id="gender" name="gender" placeholder="Gender (Male or Female)" required="required" /><br>
                    <input type="text" id="age" name="age" placeholder="Age (in Years)" required="required" /><br>
                    <input type="text" id="head_size" name="head_size" placeholder="Head Size (in cm^3)" required="required" /><br>
                    <input type="submit" class="btn btn-primary btn-block btn-Large" value="Predict" />
                    <br>
                    <br>
                    {{ prediction_text }}
                </form>
            </div>
        </center>
    </body>
</html>
```

Output Page Source Code:



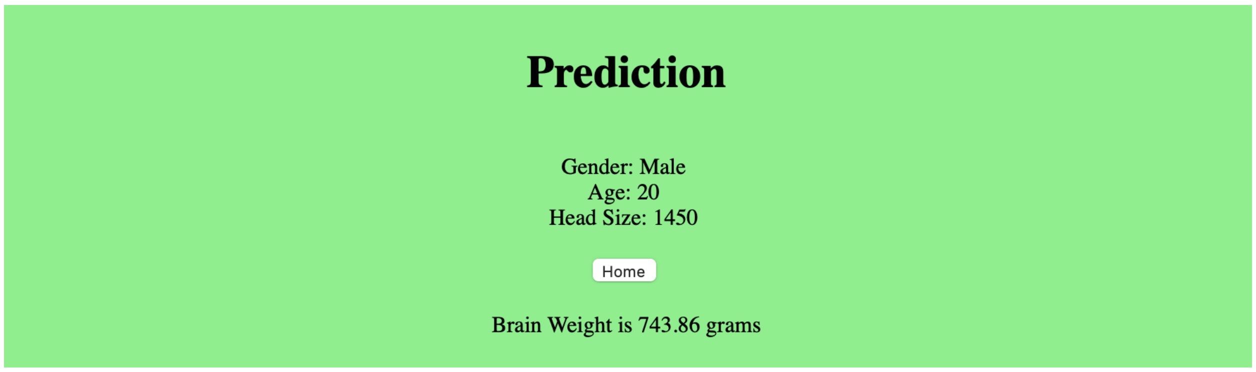
```html
<html>
    <head>
        <meta charset="UTF-8">
        <title>
            Brain Weight Predictor
        </title>
        <style>
            .container{
                padding: 25px;
                background-color: lightgreen;
                text-align: center;
            }
        </style>
    </head>
    <body>
        <center>
            <div class="container">
                <h1>
                    Prediction
                </h1>
                <form action="{{url_for('index')}}">
                    <br>
                    {{gender}}
                    <br>
                    {{age}}
                    <br>
                    {{head_size}}
                    <br>
                    <br>
                    <input type="submit" class="btn btn-primary btn-block btn-Large" value="Home" />
                    <br>
                    <br>
                    {{ prediction_text }}
                </form>
            </div>
        </center>
    </body>
</html>
```

Landing Page:



Predicting Weight of Brain using Gender, Age and Head Size

Output Page (with Sample Output):



Prediction

Gender: Male
Age: 20
Head Size: 1450

Brain Weight is 743.86 grams