Anticipating Caloric Expenditure With Machine Learning

Short Term Internship

PROJECT REPORT

Team Members:

1. Dhanaraju

2.Dileep

3. Kishore

4. Saleem

INTRODUCTION

- Predicting calorie consumption is a crucial aspect of nutrition and health management.

- By using data such as age, gender, weight, activity level and dietary choices, predictive models can estimate daily calorie needs

- These predictions aid individuals in making informed decisions about their deit and can be instrumental in acheiving health and fitness goals

- In this context, we provide data set and train our model to predict how many calories Burned during Exercise

OVERVIEW

- Data Collection:

Gather a diverse dataset that includes information like age, gender, weight, height. duration. heart rate and Body temperature during Exercise.

- Data Preprocessing:

→ clean the data by handling missing values and Outlines

→ Normalize or standardize Numerical features

Model Selection:

Choose appropriate machine learning algorithms for regression such as linear regression, decision trees, random forests or Neural Networks.

Training and Validation.

→ Split the dataset into training and testing Sets.

→ Train the Selected models on the training data and evalute their performance using metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), or Rsquared

5. Model Evaluation:

- Assess the Model's generalization on the testing data

6. Deployment:

- Once a satisfactory model is Achieved, deploy it as an application or Service where users can input their details and get calarie expenditure prediction

DESCRIPTION

This project focuses on developing a predictive model to estimate the No. of calaries a person burns during various physical activities.

→ The Primary goal of this project is to assist individuals in managing their fitness and health. The project could Lead to development of User friendly application or Service that allows users to input their information and get real time calorie expenditure Predictions

PURPOSE

1. health and fitness Management:

→ The project aims to assist individuals in managing their health and fitness more effectively by providing accurate estimates of the calories burned during various physical Activities. This information can help people make informed decisions about their exercise routines and dietary choices

2. Data-driven Decision Making:-

The project leverages machine learning to provide data driven insights. It emphasizes the importance of using technology to make informed choices regarding calorie intake and expenditure, aligning with modern trends in health and fitness

LITERATURE SURVEY

In a calarie expenditure prediction project. Several existing approaches and Methods can be used to address the problem effectively. Here are some common approache

1. Linear Regression.

Linear regression models can be employed to establish a linear relationships Between input features and output Variable (calorie expenditure)

2. Decision Trees and Random forests.

Decision trees and random forests are useful for capturing non-linear relationships Between input features and Output Variable

3. XG Boost Regressor.

XG Boost is a powerful gradient Boosting framework that is widely used in machine learning for regressive task.

Our Choice

Using XG Boost uegressor is an orobust choice, offering Both Accuracy and interpretability, making it a popular method for vregression tasks in Machine Learning.

1. High predictive Accuracy
2. Handling Non-Linearity Between input features and Output variable
3. Feature Importance
4. Scalability
5. Tuning Options of Hyper paranebers

Hardware and Software Requirements

Hardware Required: System or Laptop

Software Required: Anaconda Navigator, Flask, python
XgBoost Library, streamlit

System Required: Windows (7, 8, 9, 10, 11). 4GB Ram
256 GB Hard disk

Experimental Investigations

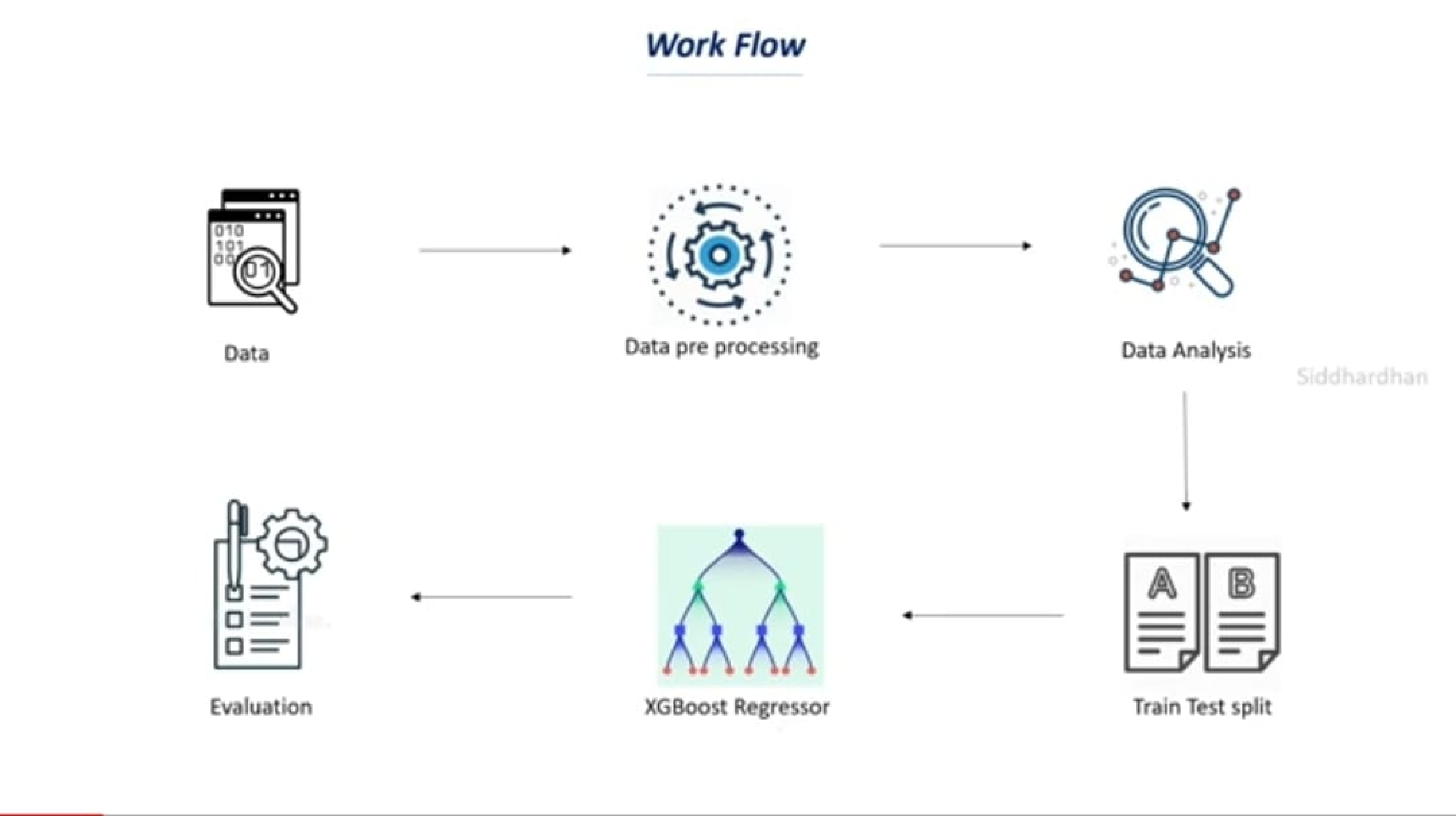
1. Data Splitting: we divide the given data set into two parts, typhically training and testing sets

2. Model training: We Used XG Boost regressor to train the model and data set.

3. Model Evaluation: We Evaluated the Model using appropriate regression metrics such as Mean Absolute Error (MAE) and R. Squared (R2) Error

4. Comparative Analysis: we have explored multiple machine learning algorithms. compare their performance to identify the most effective Approach

5. Model Visualization: we Visualized the models Predictions and compared them to actual Values. Visulaization helped in Understanding the models Behaviour and potential Shortcomings.



Work Flow

Data

Data pre processing

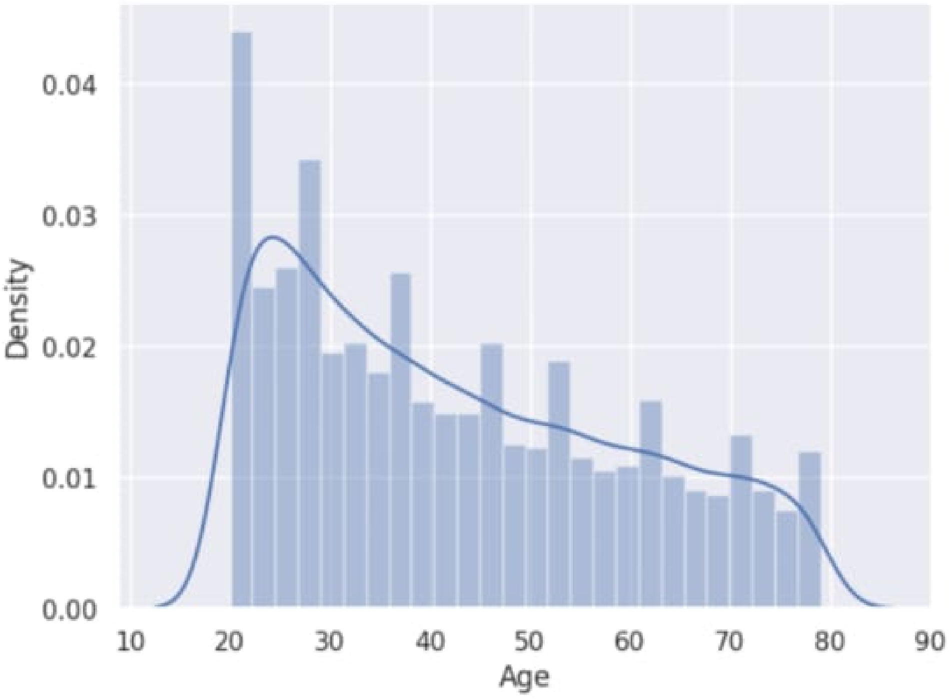
Data Analysis

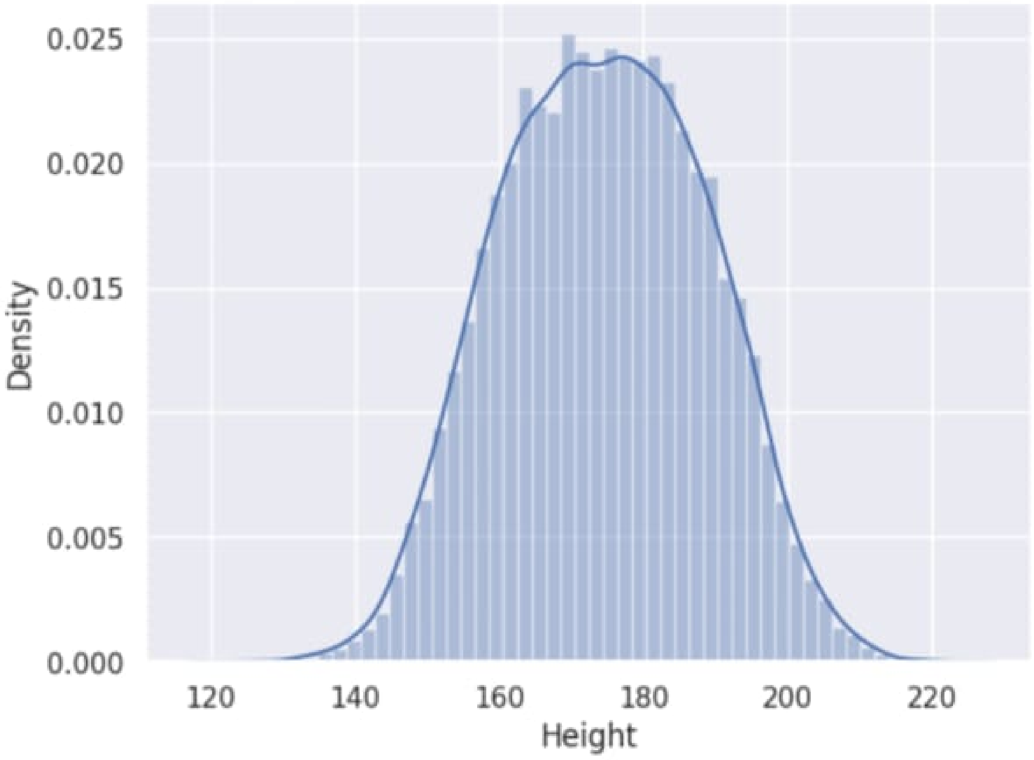
Train Test split

XGBoost Regressor

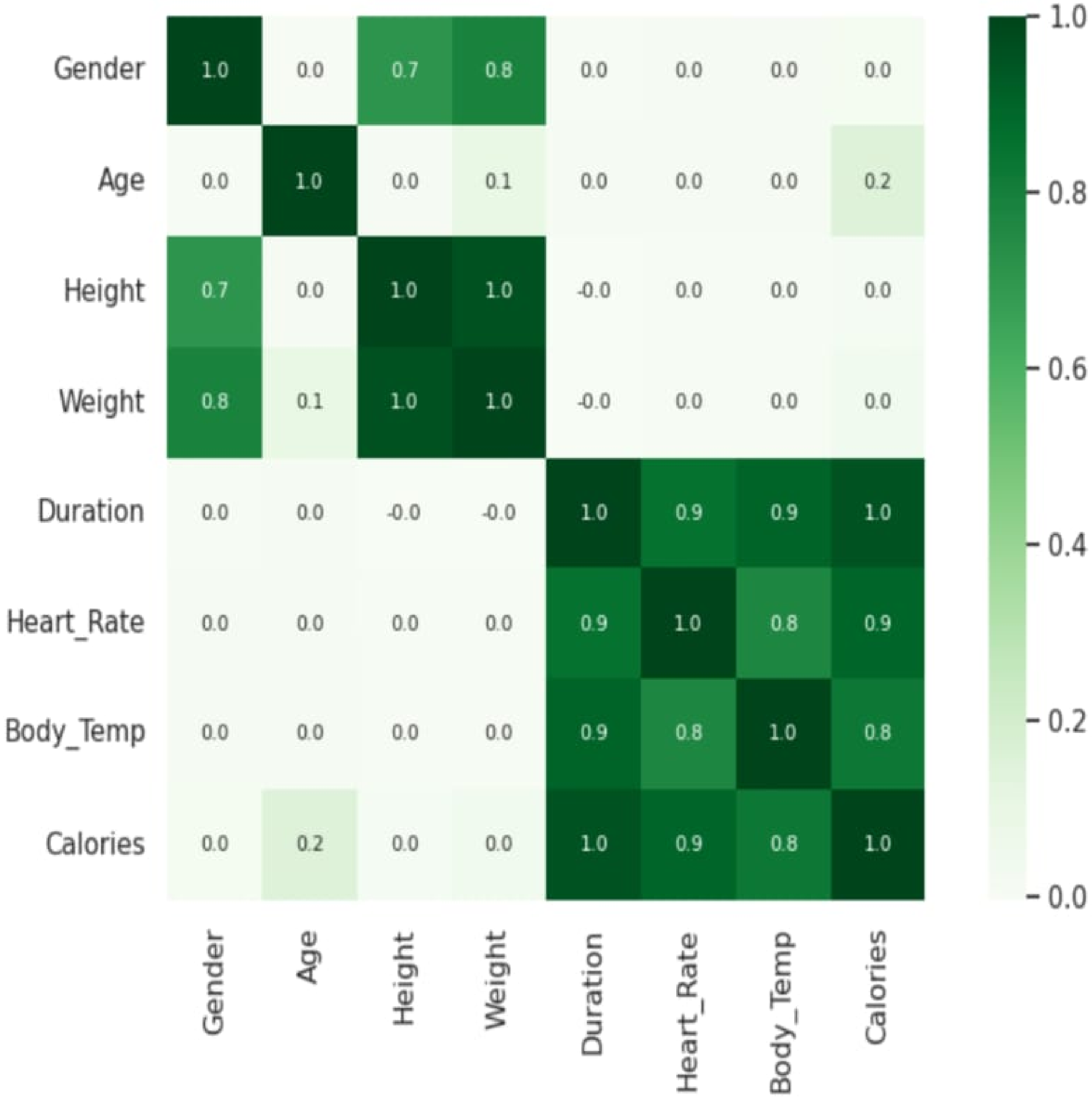
Evaluation

Graphs


0.04
0.03
0.02
0.01
0.00
Density
10
20
30
40
50
60
70
80
90
Age

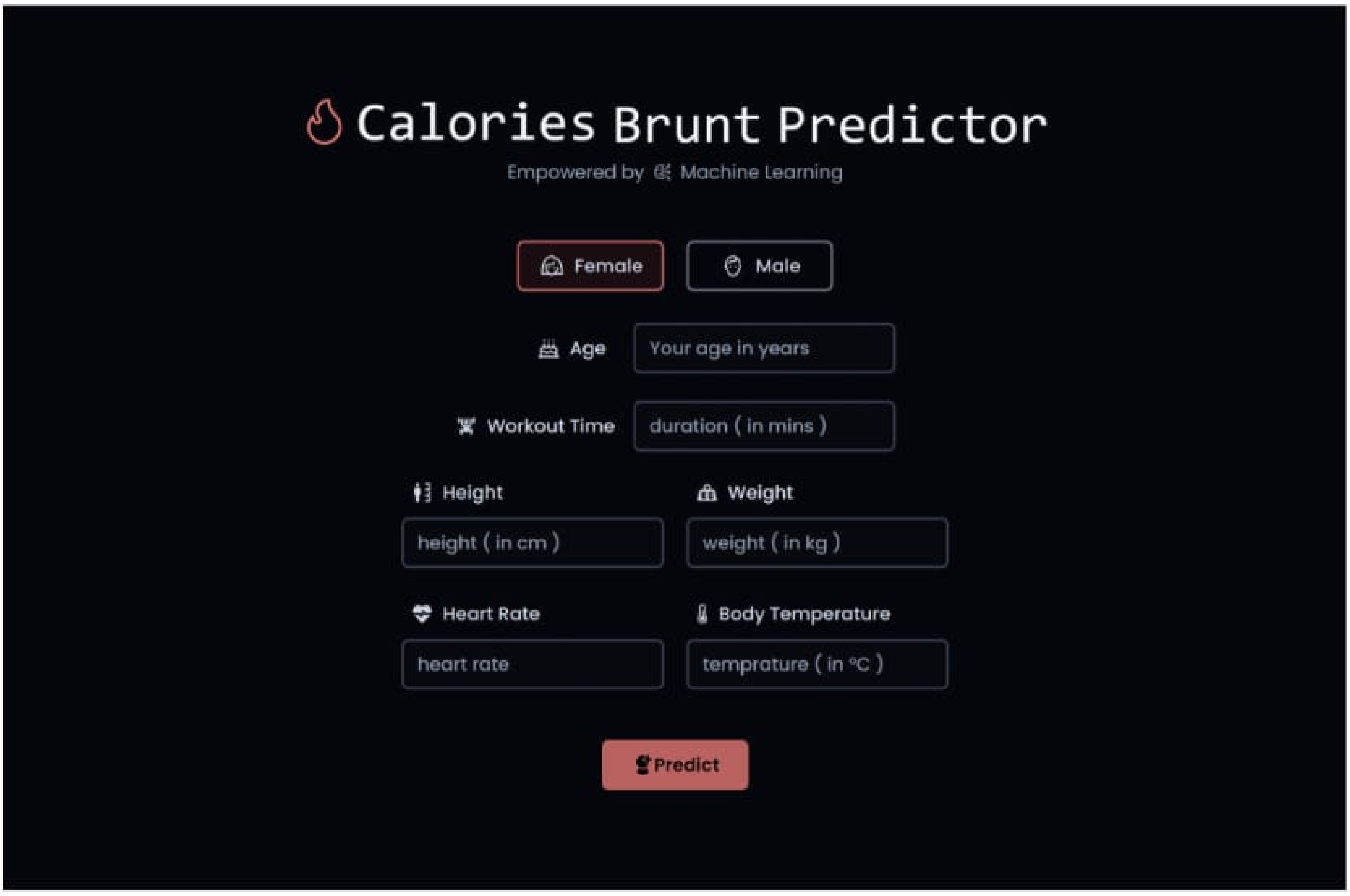

0.025
0.020
0.015
0.010
0.005
0.000
Density
120
140
160
180
200
220
Height

Graphs



Gender
Age
Height
Weight
Duration
Heart_Rate
Body_Temp
Calories
1.0 0.0 0.7 0.8 0.0 0.0 0.0 0.0
0.0 1.0 0.0 0.1 0.0 0.0 0.0 0.2
0.7 0.0 1.0 1.0 -0.0 0.0 0.0 0.0
0.8 0.1 1.0 1.0 -0.0 0.0 0.0 0.0
0.0 0.0 -0.0 -0.0 1.0 0.9 0.9 1.0
0.0 0.0 0.0 0.0 0.9 1.0 0.8 0.9
0.0 0.0 0.0 0.0 0.9 0.8 1.0 0.8
0.0 0.2 0.0 0.0 1.0 0.9 0.8 1.0
Gender
Age
Height
Weight
Duration
Heart_Rate
Body_Temp
Calories
1.0
0.8
0.6
0.4
0.2
0.0

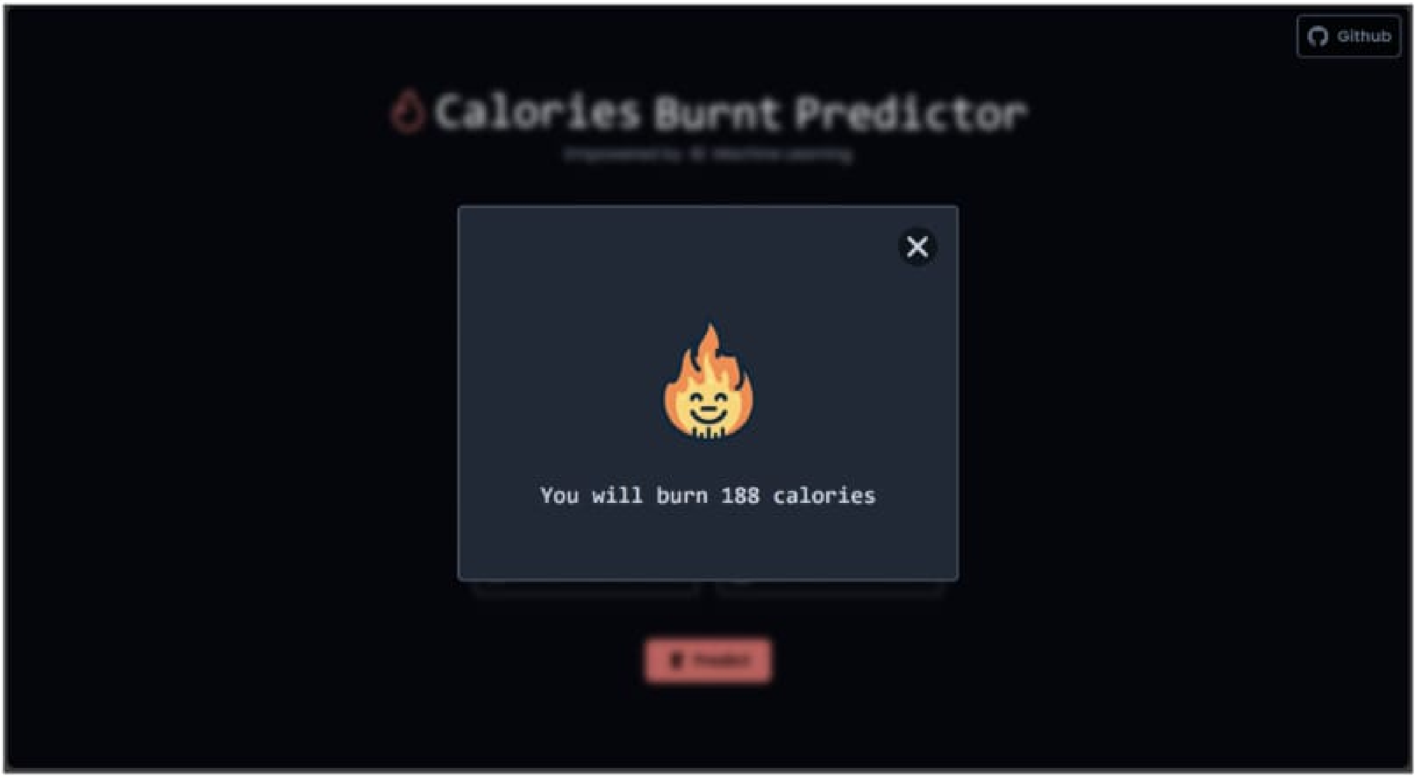
Before Input :



Calories Brunt Predictor

Empowered by Machine Learning

Result :



You will burn 188 calories

Advantages

1. Health and fitness Management- The Project can empower individuals to make informed decisions about health & fitness

2. Accessibility:- User friendly applications or Services make this information Accessible to a wide Audience

3. Continous Learning: The Project can continously improve its accuracy by incorporating new data and research findings

4. Predictive Accuracy:- Machine learning models can provide reasonably Accurate predictions of calorie expenditure

Disadvantages

1. Resource Intensive: Training and maintaing machine learning models can require Significant computational resources. particularly for deep learning models

2. Dependency on Data Updates:- The Accuracy of System relies on regulor Updates with new data and research findings

3. User Adoption: 'Users may be reluctant to adopt or trust the predictions, especially if they perceive inaccuracies in System

Applications

→ Public health Initiatives

→ Educational tools

→ Corporate Wellness program

→ Fitness apps and Service

→ Food and Beverage Industry etc

CONCLUSION

In a world where health and wellness are of paramount importance. a calorie expenditure prediction project stands at the intersection of data Science, technology. and personal well-Being.

It has the potential to improve lives. Support healthier choices and advance our understanding of Human health. The project will continue to play a Significant role in promoting healthier Life Styles

Future Scope

The future of calorie expenditure prediction projects is exciting. with opportunities for innovation and positive impacts on individual health and Well Being. Here are Some potential areas for future Scope

AI and Medical Diagnostics: Explore the integration of calorie expenditure data with AI-driven medical diagnostics for early detection of Health issues

Research Collaborations: Collaborate with Researchers and Universities to conduct large-scale Studies

Environmental Impact: Expand the Scope to assess the environmental Impact of physical activities, including carbon footprint Calculation

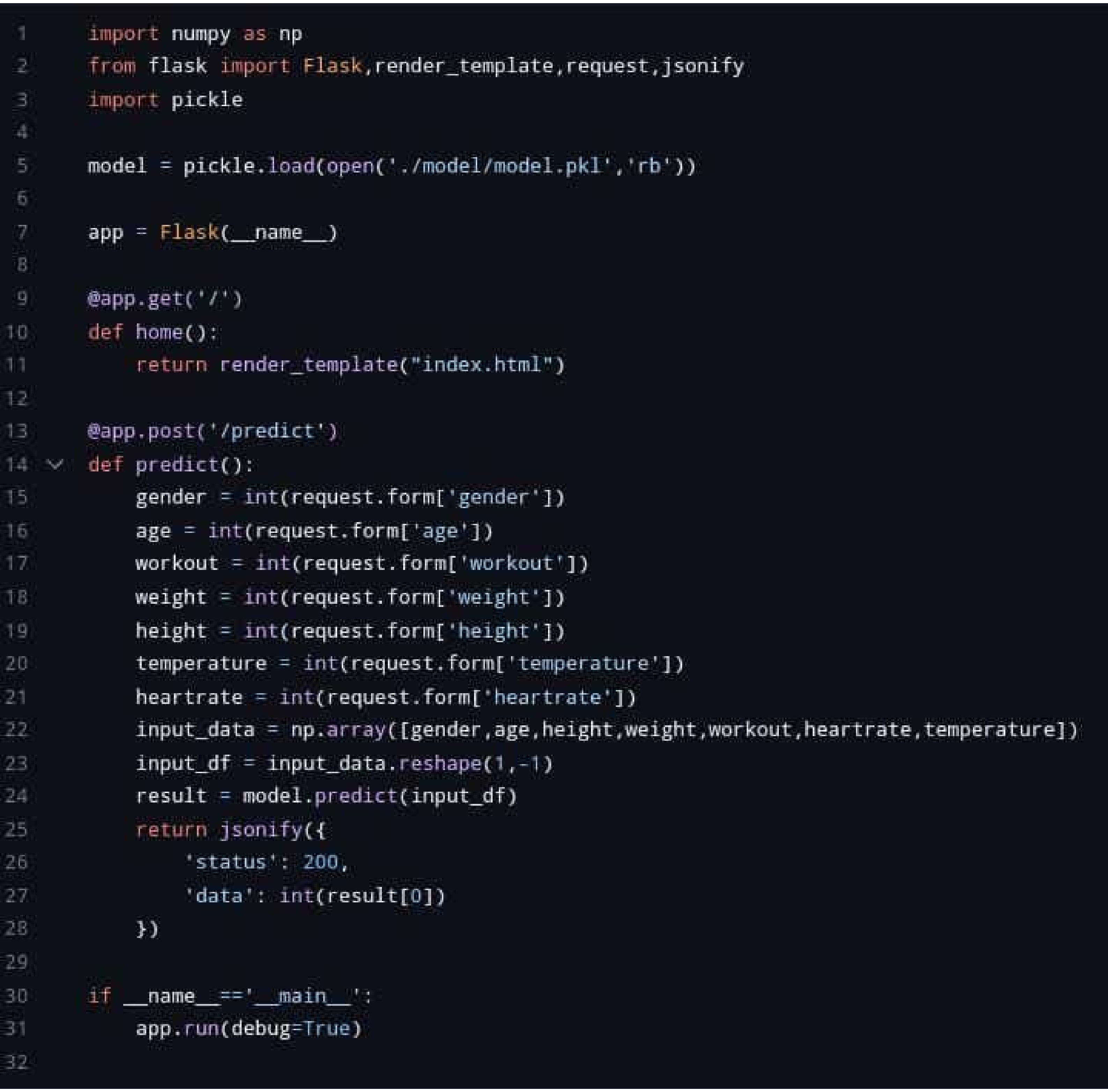
Machine Learning Advancements: Explore the Use of Advanced machine learning techniques, such as reinforcement learning, to optimize individualized exercise Routines

Reference Books and Websites

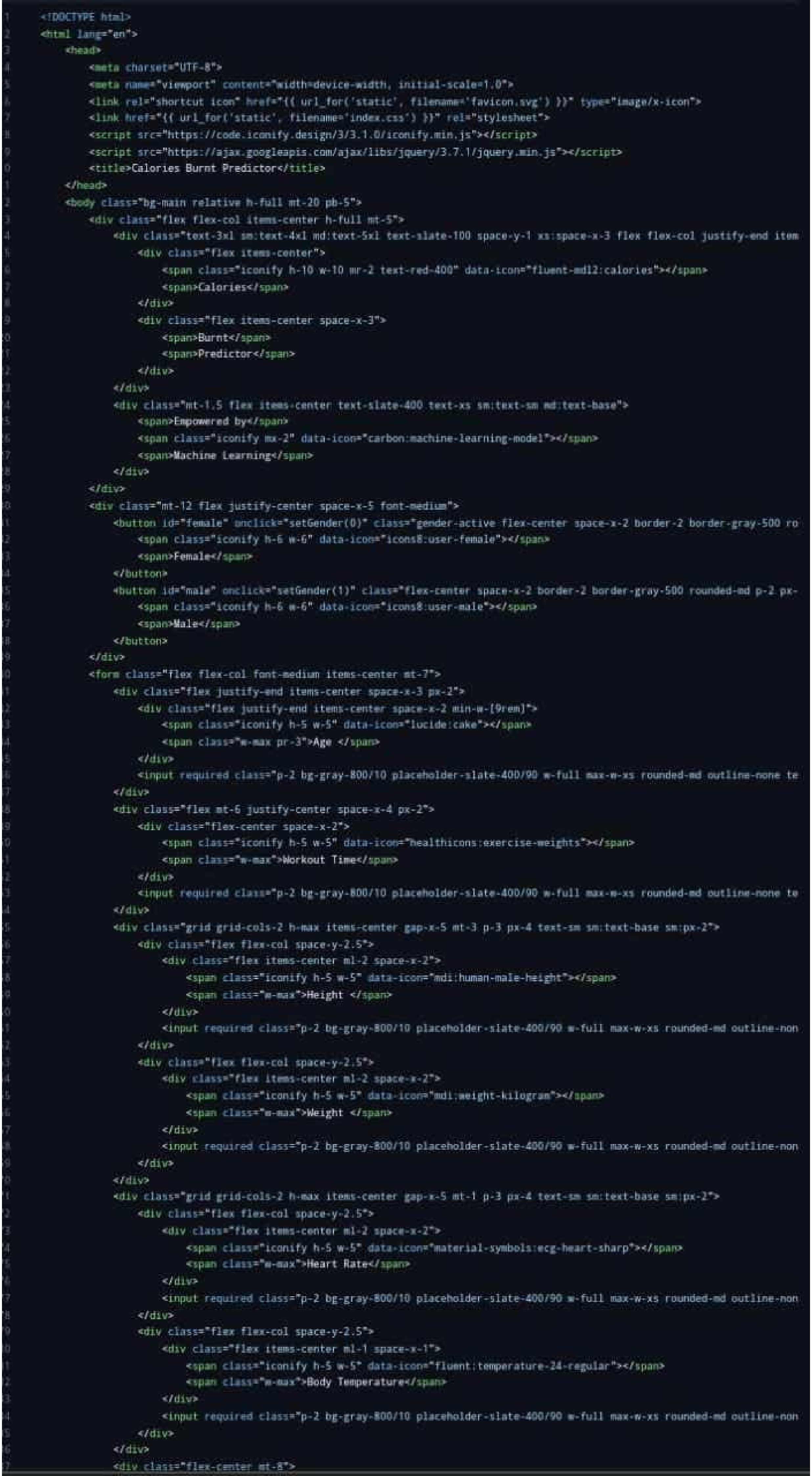
1. "Machine Leaning Yearning"

by Andrew Ng

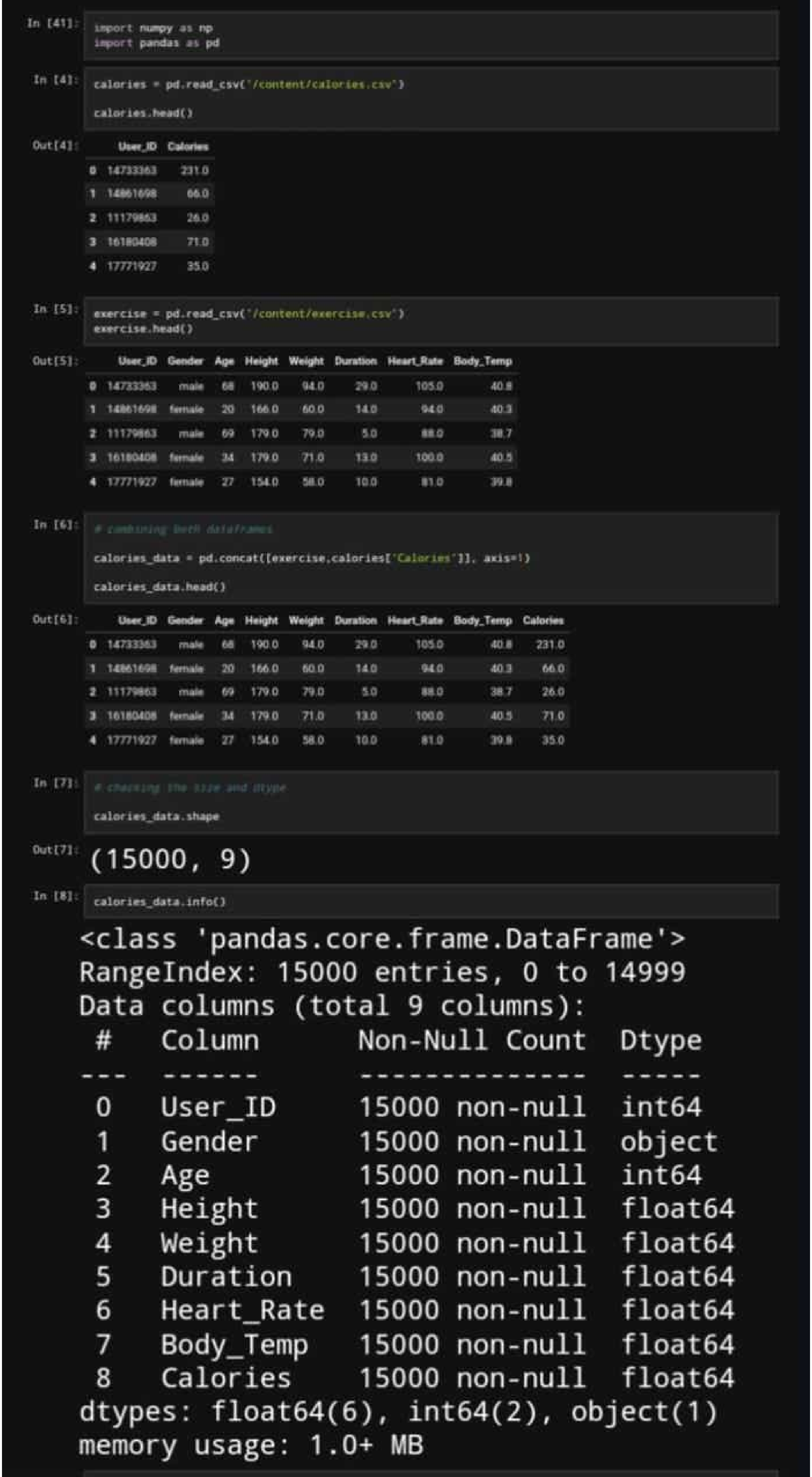
2. GeekforGeeks and Github (websites)



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

model = pickle.load(open('./model/model.pkl','rb'))

app = Flask(__name__)

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

@app.post('/predict')
def predict():
    gender = int(request.form['gender'])
    age = int(request.form['age'])
    workout = int(request.form['workout'])
    weight = int(request.form['weight'])
    height = int(request.form['height'])
    temperature = int(request.form['temperature'])
    heartrate = int(request.form['heartrate'])
    input_data = np.array([gender,age,height,weight,workout,heartrate,temperature])
    input_df = input_data.reshape(1,-1)
    result = model.predict(input_df)
    return jsonify({
        'status': 200,
        'data': int(result[0])
    })

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



```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <link rel="shortcut icon" href="{{ url_for('static', filename='favicon.svg') }}" type="image/x-icon">
        <link href="{{ url_for('static', filename='index.css') }}" rel="stylesheet">
        <script src="https://code.iconify.design/3/3.1.0/iconify.min.js"></script>
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
        <title>Calories Burnt Predictor</title>
    </head>
    <body class="bg-main relative h-full mt-20 pb-5">
        <div class="flex flex-col items-center h-full mt-5">
            <div class="text-3xl sm:text-4xl md:text-5xl text-slate-100 space-y-1 xs:space-x-3 flex flex-col justify-end item
                <div class="flex items-center">
                    <span class="iconify h-10 w-10 mr-2 text-red-400" data-icon="fluent-mdl2:calories"></span>
                    <span>Calories</span>
                </div>
                <div class="flex items-center space-x-3">
                    <span>Burnt</span>
                    <span>Predictor</span>
                </div>
            </div>
            <div class="mt-1.5 flex items-center text-slate-400 text-xs sm:text-sm md:text-base">
                <span>Empowered by</span>
                <span class="iconify mx-2" data-icon="carbon:machine-learning-model"></span>
                <span>Machine Learning</span>
            </div>
        </div>
        <div class="mt-12 flex justify-center space-x-5 font-medium">
            <button id="female" onclick="setGender(0)" class="gender-active flex-center space-x-2 border-2 border-gray-500 ro
                <span class="iconify h-6 w-6" data-icon="icons8:user-female"></span>
                <span>Female</span>
            </button>
            <button id="male" onclick="setGender(1)" class="flex-center space-x-2 border-2 border-gray-500 rounded-md p-2 px-
                <span class="iconify h-6 w-6" data-icon="icons8:user-male"></span>
                <span>Male</span>
            </button>
        </div>
        <form class="flex flex-col font-medium items-center mt-7">
            <div class="flex justify-end items-center space-x-3 px-2">
                <div class="flex justify-end items-center space-x-2 min-w-[9rem]">
                    <span class="iconify h-5 w-5" data-icon="lucide:cake"></span>
                    <span class="w-max pr-3">Age </span>
                </div>
                <input required class="p-2 bg-gray-800/10 placeholder-slate-400/90 w-full max-w-xs rounded-md outline-none te
            </div>
            <div class="flex mt-6 justify-center space-x-4 px-2">
                <div class="flex-center space-x-2">
                    <span class="iconify h-5 w-5" data-icon="healthicons:exercise-weights"></span>
                    <span class="w-max">Workout Time</span>
                </div>
                <input required class="p-2 bg-gray-800/10 placeholder-slate-400/90 w-full max-w-xs rounded-md outline-none te
            </div>
            <div class="grid grid-cols-2 h-max items-center gap-x-5 mt-3 p-3 px-4 text-sm sm:text-base sm:px-2">
                <div class="flex flex-col space-y-2.5">
                    <div class="flex items-center ml-2 space-x-2">
                        <span class="iconify h-5 w-5" data-icon="mdi:human-male-height"></span>
                        <span class="w-max">Height </span>
                    </div>
                    <input required class="p-2 bg-gray-800/10 placeholder-slate-400/90 w-full max-w-xs rounded-md outline-non
                </div>
                <div class="flex flex-col space-y-2.5">
                    <div class="flex items-center ml-2 space-x-2">
                        <span class="iconify h-5 w-5" data-icon="mdi:weight-kilogram"></span>
                        <span class="w-max">Weight </span>
                    </div>
                    <input required class="p-2 bg-gray-800/10 placeholder-slate-400/90 w-full max-w-xs rounded-md outline-non
                </div>
            </div>
            <div class="grid grid-cols-2 h-max items-center gap-x-5 mt-1 p-3 px-4 text-sm sm:text-base sm:px-2">
                <div class="flex flex-col space-y-2.5">
                    <div class="flex items-center ml-2 space-x-2">
                        <span class="iconify h-5 w-5" data-icon="material-symbols:ecg-heart-sharp"></span>
                        <span class="w-max">Heart Rate</span>
                    </div>
                    <input required class="p-2 bg-gray-800/10 placeholder-slate-400/90 w-full max-w-xs rounded-md outline-non
                </div>
                <div class="flex flex-col space-y-2.5">
                    <div class="flex items-center ml-1 space-x-1">
                        <span class="iconify h-5 w-5" data-icon="fluent:temperature-24-regular"></span>
                        <span class="w-max">Body Temperature</span>
                    </div>
                    <input required class="p-2 bg-gray-800/10 placeholder-slate-400/90 w-full max-w-xs rounded-md outline-non
                </div>
            </div>
            <div class="flex-center mt-8">
```



In [41]:

```
import numpy as np
import pandas as pd
```

In [4]:

```
calories = pd.read_csv('/content/calories.csv')

calories.head()
```

Out[4]:

	User_ID	Calories
0	14733363	231.0
1	14861698	66.0
2	11179863	26.0
3	16180408	71.0
4	17771927	35.0

In [5]:

```
exercise = pd.read_csv('/content/exercise.csv')
exercise.head()
```

Out[5]:

	User_ID	Gender	Age	Height	Weight	Duration	Heart_Rate	Body_Temp
0	14733363	male	68	190.0	94.0	29.0	105.0	40.8
1	14861698	female	20	166.0	60.0	14.0	94.0	40.3
2	11179863	male	69	179.0	79.0	5.0	88.0	38.7
3	16180408	female	34	179.0	71.0	13.0	100.0	40.5
4	17771927	female	27	154.0	58.0	10.0	81.0	39.8

In [6]:

```
# combining both dataframes

calories_data = pd.concat([exercise,calories['Calories']], axis=1)

calories_data.head()
```

Out[6]:

	User_ID	Gender	Age	Height	Weight	Duration	Heart_Rate	Body_Temp	Calories
0	14733363	male	68	190.0	94.0	29.0	105.0	40.8	231.0
1	14861698	female	20	166.0	60.0	14.0	94.0	40.3	66.0
2	11179863	male	69	179.0	79.0	5.0	88.0	38.7	26.0
3	16180408	female	34	179.0	71.0	13.0	100.0	40.5	71.0
4	17771927	female	27	154.0	58.0	10.0	81.0	39.8	35.0

In [7]:

```
# checking the size and dtype

calories_data.shape
```

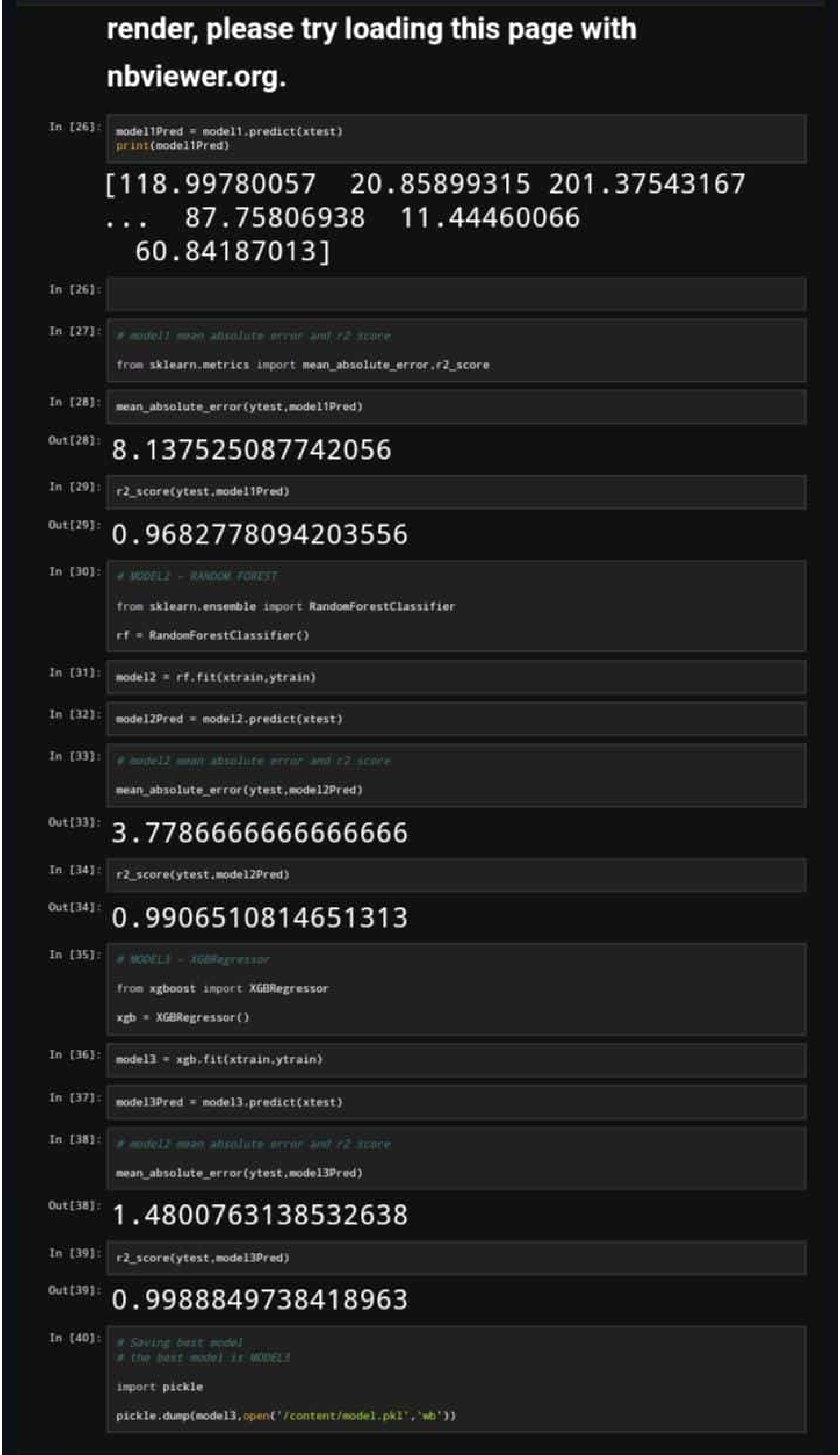
Out[7]:

```
(15000, 9)
```

In [8]:

```
calories_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15000 entries, 0 to 14999
Data columns (total 9 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   User_ID     15000 non-null  int64
 1   Gender      15000 non-null  object
 2   Age         15000 non-null  int64
 3   Height      15000 non-null  float64
 4   Weight      15000 non-null  float64
 5   Duration    15000 non-null  float64
 6   Heart_Rate  15000 non-null  float64
 7   Body_Temp   15000 non-null  float64
 8   Calories    15000 non-null  float64
dtypes: float64(6), int64(2), object(1)
memory usage: 1.0+ MB
```



render, please try loading this page with nbviewer.org.

```
In [26]: model1Pred = model1.predict(xtest)
         print(model1Pred)

[118.99780057  20.85899315 201.37543167
...  87.75806938  11.44460066
  60.84187013]

In [26]: 

In [27]: # model1 mean absolute error and r2 score

         from sklearn.metrics import mean_absolute_error,r2_score

In [28]: mean_absolute_error(ytest,model1Pred)

Out[28]: 8.137525087742056

In [29]: r2_score(ytest,model1Pred)

Out[29]: 0.9682778094203556

In [30]: # MODEL2 - RANDOM FOREST

         from sklearn.ensemble import RandomForestClassifier

         rf = RandomForestClassifier()

In [31]: model2 = rf.fit(xtrain,ytrain)

In [32]: model2Pred = model2.predict(xtest)

In [33]: # model2 mean absolute error and r2 score

         mean_absolute_error(ytest,model2Pred)

Out[33]: 3.7786666666666666

In [34]: r2_score(ytest,model2Pred)

Out[34]: 0.9906510814651313

In [35]: # MODEL3 - XGBRegressor

         from xgboost import XGBRegressor

         xgb = XGBRegressor()

In [36]: model3 = xgb.fit(xtrain,ytrain)

In [37]: model3Pred = model3.predict(xtest)

In [38]: # model2 mean absolute error and r2 score

         mean_absolute_error(ytest,model3Pred)

Out[38]: 1.4800763138532638

In [39]: r2_score(ytest,model3Pred)

Out[39]: 0.9988849738418963

In [40]: # Saving best model
         # the best model is MODEL3

         import pickle

         pickle.dump(model3,open('/content/model.pkl','wb'))
```