



# **Data Science Intern at Data Glacier**

**Week 4:** Deployment on Flask

**Name:** Taiwo Akingbesote

**Batch Code:** LISUM22

**Date:** 28 June 2023

**Submitted to:** Data Glacier

## **Contents:**

1. Introduction
2. Data Information
3. Build Machine Learning Model
4. Turning Model into Flask Framework
5. Running the package

# 1. Introduction

In this project, we are going to deploy a machine learning model (SVM) using the Flask Framework. As a demonstration, our model help to recommend a minimum value for a soccer player based on their age and potential. After careful examination, this model is more accurate for players aged 18-28.

We will be building a machine learning model to determine recommended player value, then create an API for the model, using Flask, the Python micro-framework for building web applications. This API requests information (age + potential) from users and gives out the recommended value.

## 2. Data Information

This data, obtained from Kaggle.com contains the details of 52 professional soccer player and their attributes.

	ID	Name	Age	Photo	Nationality	Flag	Overall	Potential	Club	Club Logo	Value(E)	Wage(E)	Special	Preferred Foot	International Reputation
1	0	209058 L. Goretzka	27	https://cdn.sofifa.net/players/209058/23_80.png	Germany	https://cdn.sofifa.net/flags/de.png	87	7	88 FC Bayern München	https://cdn.sofifa.net/teams/21/20.png	9100000.0	115000.0	2312	Right	4.0
2	1	212198 Bruno Fernandes	27	https://cdn.sofifa.net/players/212198/23_80.png	Portugal	https://cdn.sofifa.net/flags/pt.png	86	9	87 Manchester United	https://cdn.sofifa.net/teams/11/30.png	7800000.0	180000.0	2325	Right	3.0
3	2	224334 M. Acuña	30	https://cdn.sofifa.net/players/224334/23_80.png	Argentina	https://cdn.sofifa.net/flags/ar.png	85	4	85 Sevilla FC	https://cdn.sofifa.net/teams/481/30.png	4600000.0	46000.0	2303	Left	2.0
4	3	192985 K. De Bruyne	31	https://cdn.sofifa.net/players/192985/23_80.png	Belgium	https://cdn.sofifa.net/flags/be.png	91	6	91 Manchester City	https://cdn.sofifa.net/teams/19/30.png	10700000.0	350000.0	2303	Left	4.0
5	4	224332 N. Barella	25	https://cdn.sofifa.net/players/224332/23_80.png	Italy	https://cdn.sofifa.net/flags/it.png	86	6	89 Inter	https://cdn.sofifa.net/teams/14/30.png	8900000.0	110000.0	2296	Right	3.0
6	5	212622 J. Kimmich	27	https://cdn.sofifa.net/players/212622/23_80.png	Germany	https://cdn.sofifa.net/flags/de.png	89	7	90 FC Bayern München	https://cdn.sofifa.net/teams/21/30.png	10500000.0	130000.0	2383	Right	4.0
7	6	197445 D. Alaba	30	https://cdn.sofifa.net/players/197445/23_80.png	Austria	https://cdn.sofifa.net/flags/at.png	86	9	86 Real Madrid CF	https://cdn.sofifa.net/teams/24/30.png	5500000.0	220000.0	2277	Left	4.0
8	7	187945 22 Paulinho	32	https://cdn.sofifa.net/players/187945/22_80.png	Brazil	https://cdn.sofifa.net/flags/br.png	83	9	83 Al Ahli	https://cdn.sofifa.net/teams/1238/70.png	2800000.0	81000.0	2273	Right	3.0
9	8	208333 E. Can	28	https://cdn.sofifa.net/players/208333/23_80.png	Germany	https://cdn.sofifa.net/flags/de.png	82	10	82 Borussia Dortmund	https://cdn.sofifa.net/teams/22/30.png	3000000.0	63000.0	2271	Right	3.0
10	9	210514 João Cancelo	28	https://cdn.sofifa.net/players/210514/23_80.png	Portugal	https://cdn.sofifa.net/flags/pt.png	88	11	88 Manchester City	https://cdn.sofifa.net/teams/19/30.png	8200000.0	250000.0	2262	Right	3.0
11	10	228251 L. Pellegrini	26	https://cdn.sofifa.net/players/228251/23_80.png	Italy	https://cdn.sofifa.net/flags/it.png	84	12	87 Roma	https://cdn.sofifa.net/teams/15/30.png	5900000.0	90000.0	2258	Right	2.0
12	11	177003 L. Modrić	36	https://cdn.sofifa.net/players/177003/23_80.png	Croatia	https://cdn.sofifa.net/flags/hr.png	88	13	88 Real Madrid CF	https://cdn.sofifa.net/teams/24/30.png	2900000.0	230000.0	2257	Right	4.0
13	12	223848 S. Milinković-Savić	27	https://cdn.sofifa.net/players/223848/23_80.png	Serbia	https://cdn.sofifa.net/flags/rs.png	86	14	87 Lazio	https://cdn.sofifa.net/teams/45/30.png	7700000.0	90000.0	2250	Right	3.0
14	13	225193 Merino	26	https://cdn.sofifa.net/players/225193/23_80.png	Spain	https://cdn.sofifa.net/flags/es.png	83	15	86 Real Sociedad	https://cdn.sofifa.net/teams/45/30.png	4700000.0	49000.0	2247	Left	1.0
15	14	226161 Marcos Llorente	27	https://cdn.sofifa.net/players/226161/23_80.png	Spain	https://cdn.sofifa.net/flags/es.png	84	16	85 Atlético de Madrid	https://cdn.sofifa.net/teams/240/30.png	4800000.0	81000.0	2244	Right	3.0
16	15	212616 R. De Paul	28	https://cdn.sofifa.net/players/212616/23_80.png	Argentina	https://cdn.sofifa.net/flags/ar.png	84	17	84 Atlético de Madrid	https://cdn.sofifa.net/teams/240/30.png	4200000.0	85000.0	2240	Right	1.0
17	16	181458 I. Perišić	33	https://cdn.sofifa.net/players/181458/23_80.png	Croatia	https://cdn.sofifa.net/flags/hr.png	84	18	84 Tottenham Hotspur	https://cdn.sofifa.net/teams/18/30.png	2600000.0	130000.0	2238	Right	3.0
18	17	228702 F. de Jong	25	https://cdn.sofifa.net/players/228702/23_80.png	Netherlands	https://cdn.sofifa.net/flags/nl.png	87	19	90 FC Barcelona	https://cdn.sofifa.net/teams/241/30.png	11600000.0	230000.0	2238	Right	3.0
19	18	194765 A. Griemmann	33	https://cdn.sofifa.net/players/194765/23_80.png	France	https://cdn.sofifa.net/flags/fr.png	83	20	83 Atlético de Madrid	https://cdn.sofifa.net/teams/240/30.png	3000000.0	180000.0	2235	Left	4.0
20	19	193982 J. Guadalupe	34	https://cdn.sofifa.net/players/193982/23_80.png	Colombia	https://cdn.sofifa.net/flags/co.png	83	21	83 Juventus	https://cdn.sofifa.net/teams/45/30.png	1300000.0	12000.0	2234	Right	3.0
21	20	208574 F. Kostić	29	https://cdn.sofifa.net/players/208574/23_80.png	Serbia	https://cdn.sofifa.net/flags/rs.png	85	22	85 Eintracht Frankfurt	https://cdn.sofifa.net/teams/1824/30.png	5300000.0	56000.0	2234	Left	3.0
22	21	231281 T. Alexander-Arnold	23	https://cdn.sofifa.net/players/231281/23_80.png	England	https://cdn.sofifa.net/flags/gb-eng.png	87	23	90 Liverpool	https://cdn.sofifa.net/teams/9/30.png	10000000.0	150000.0	2234	Right	3.0
23	22	170580 L. Suárez	35	https://cdn.sofifa.net/players/170580/23_80.png	Uruguay	https://cdn.sofifa.net/flags/uy.png	84	24	84 Club Nacional de Fútbol	https://cdn.sofifa.net/teams/11325/30.png	1800000.0	1000.0	2231	Right	4.0
24	23	230212 A. Hakimi	23	https://cdn.sofifa.net/players/230212/23_80.png	Morocco	https://cdn.sofifa.net/flags/ma.png	84	25	87 Paris Saint-Germain	https://cdn.sofifa.net/teams/73/30.png	5300000.0	90000.0	2229	Right	2.0
25	24	216502 M. Brozović	29	https://cdn.sofifa.net/players/216502/23_80.png	Croatia	https://cdn.sofifa.net/flags/hr.png	86	26	86 Inter	https://cdn.sofifa.net/teams/14/30.png	5800000.0	110000.0	2227	Right	3.0
26	25	205331 M. Salah	30	https://cdn.sofifa.net/players/205331/23_80.png	Egypt	https://cdn.sofifa.net/flags/eg.png	90	27	90 Liverpool	https://cdn.sofifa.net/teams/9/30.png	11500000.0	270000.0	2226	Left	4.0

Figure 2.1: Dataset Information

## 3. Building a Model

### 3.1.1 Importing Required Libraries and Dataset

In this part, we import libraires and dataset which contains information of the players

```
In [39]: import numpy as np
import pandas as pd
import pickle
from scipy.optimize import curve_fit
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

In [ ]: # Load the FIFA23 dataset
df = pd.read_csv('FIFA23.csv')
```

Figure 3.1.1

### 3.1.2 Data Manipulating

To ensure more accuracy for our model, we filter out players ranging from age 17 (the least age in the data) – 28. We also filtered out players with values < 0. These measures ensure a more accurate recommendation from the model.

```
In [130]: # Calculate the sum of Age and Potential to create a new feature
filtered_df['Age_potential'] = filtered_df['Age'] + filtered_df['Potential']

# Prepare the data for exponential curve fit
X = filtered_df['Age_potential'].values.reshape(-1,1) # x, the addition of both age and potential
y = filtered_df['Value (£)'].values                # y, the result, the value of the corresponding x
```

Figure 3.1.2

### 3.1.3 Building the Model

After data manipulation, we implemented linear regression on the new dataset from Scikit-learn. After this step, we generated the predicted values

```
In [ ]: # Perform Linear regression on the adjusted data
regressor = LinearRegression()
regressor.fit(X, y)

# Generate the predicted values
y_pred = regressor.predict(X)
```

Figure 3.1.3

### 3.1.4 Visualizing the Model

After successfully building the model, we formulated a graph to better understand this data and check for any visual trends.

```
In [132]: # Plotting the original data and the adjusted linear regression line
plt.scatter(X, y, label='Adjusted Data')
plt.plot(X, y_pred, color='r', label='Linear Regression')

# Adding labels and legend to the plot
plt.xlabel('Age')
plt.ylabel('Value (£)')
plt.legend()

# Display the plot
plt.show()
```

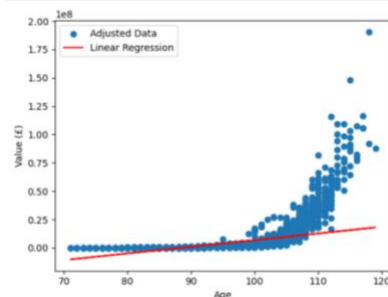


Figure3.1.4

Figure 3.1.1

### 3.1.5 Saving the Model

After that we save our model using pickle

```
In [ ]: # Save the model to a file
        pickle.dump(regressor, open('model.pkl', 'wb'))
        model = pickle.load(open('model.pkl', 'rb'))
```

Figure 3.1.5

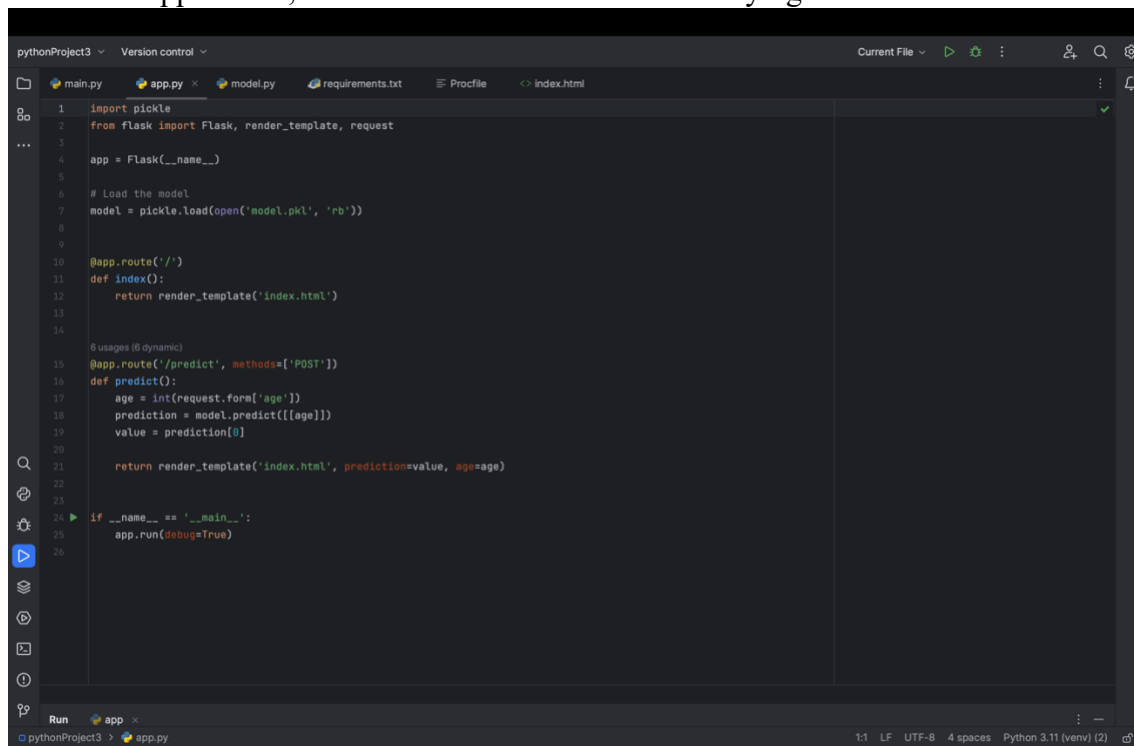
## 4. Turning Model into Flask Framework

To properly deploy this model, we've created a framework on pycharm which request users age + potential and recommends a value.

The files are as follows;

### 4.1 App.py

The app.py file contains the main code that will be executed by the Python interpreter to run the Flask web application, it included the ML code for classifying SD.



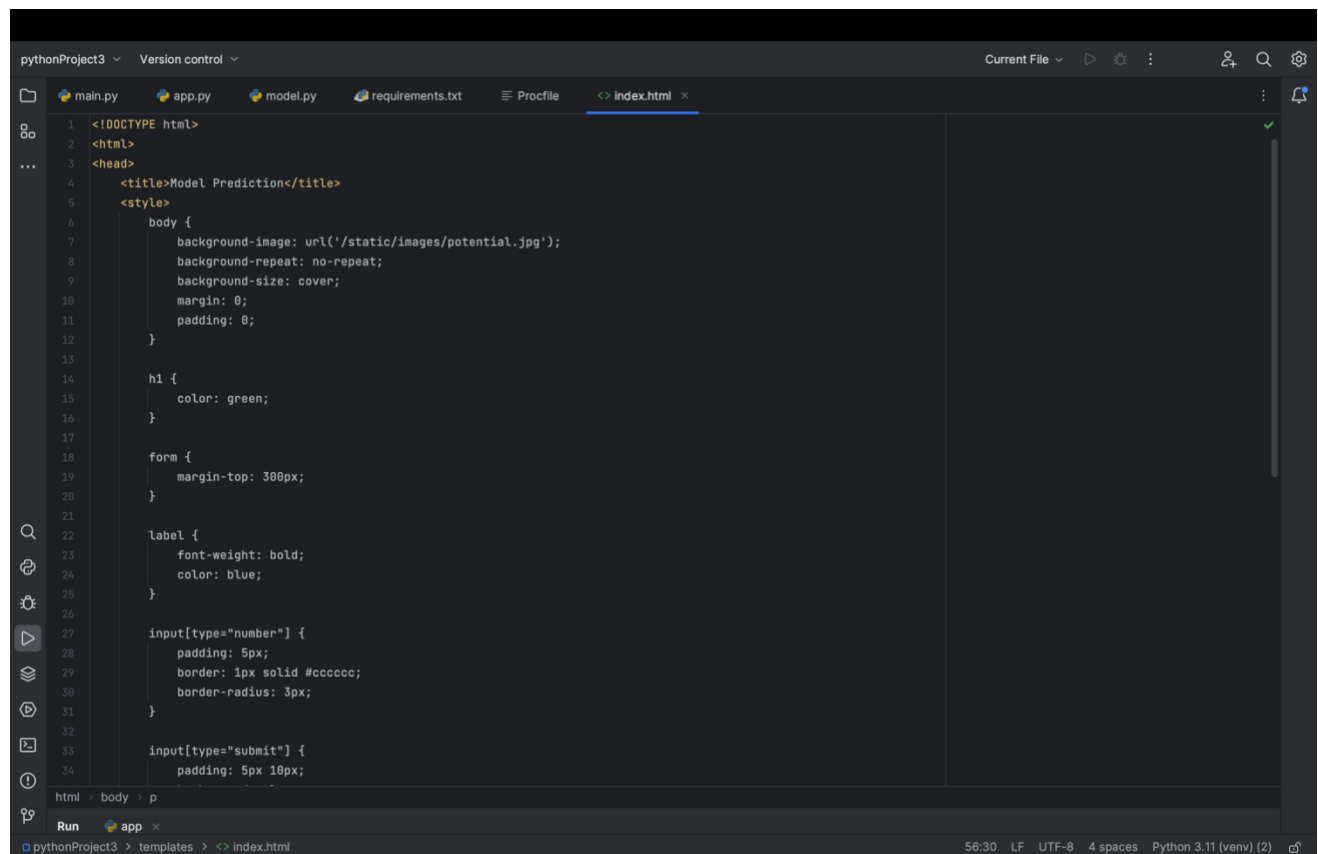
```
1 import pickle
2 from flask import Flask, render_template, request
3
4 app = Flask(__name__)
5
6 # Load the model
7 model = pickle.load(open('model.pkl', 'rb'))
8
9
10 @app.route('/')
11 def index():
12     return render_template('index.html')
13
14
15 @app.route('/predict', methods=['POST'])
16 def predict():
17     age = int(request.form['age'])
18     prediction = model.predict([[age]])
19     value = prediction[0]
20
21     return render_template('index.html', prediction=value, age=age)
22
23
24 if __name__ == '__main__':
25     app.run(debug=True)
```

Figure 4.1: App.py

- We ran our application as a single module; thus we initialized a new Flask instance with the argument `__name__` to let Flask know that it can find the `index.html` template folder (*templates*) in the same directory where it is located.
- Next, we used the route decorator (`@app.route('/')`) to specify the URL that should trigger the execution of the home function.
- Our *home* function simply rendered the *index.html* HTML file, which is in the *templates* folder.
- Inside the *predict* function, we request users age + potential the deploy the model to execute the collected data.
- Lastly, we used the *run* function to only run the application on the server when this script is directly executed by the Python interpreter, which we ensured using the *if* statement with `__name__ == '__main__'`.

## 4.2 Index.html

The following are the contents of the `index.html` file that will render a text form where a user can enter a message.



```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Model Prediction</title>
5 <style>
6     body {
7         background-image: url('/static/images/potential.jpg');
8         background-repeat: no-repeat;
9         background-size: cover;
10        margin: 0;
11        padding: 0;
12    }
13
14    h1 {
15        color: green;
16    }
17
18    form {
19        margin-top: 300px;
20    }
21
22    label {
23        font-weight: bold;
24        color: blue;
25    }
26
27    input[type="number"] {
28        padding: 5px;
29        border: 1px solid #cccccc;
30        border-radius: 3px;
31    }
32
33    input[type="submit"] {
34        padding: 5px 10px;

```

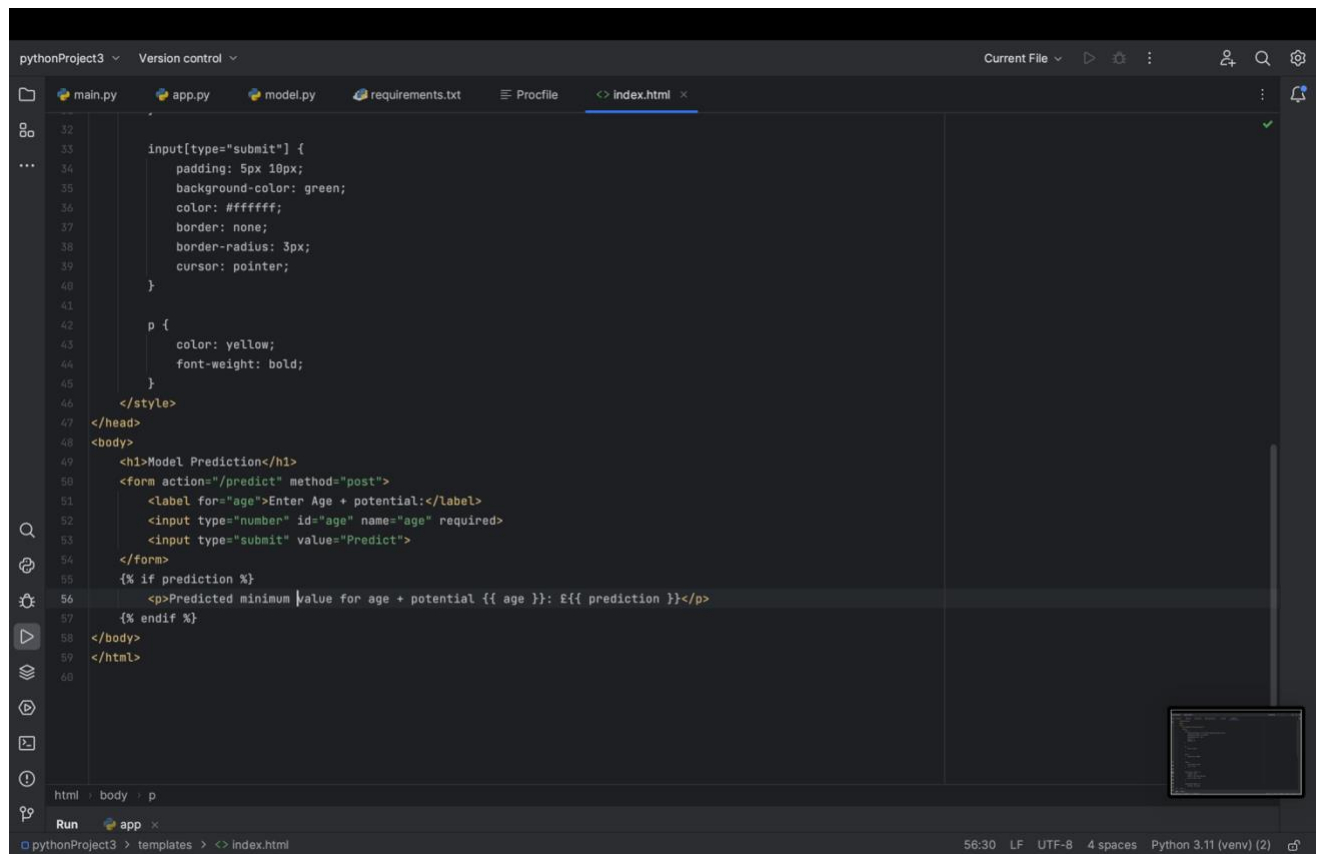
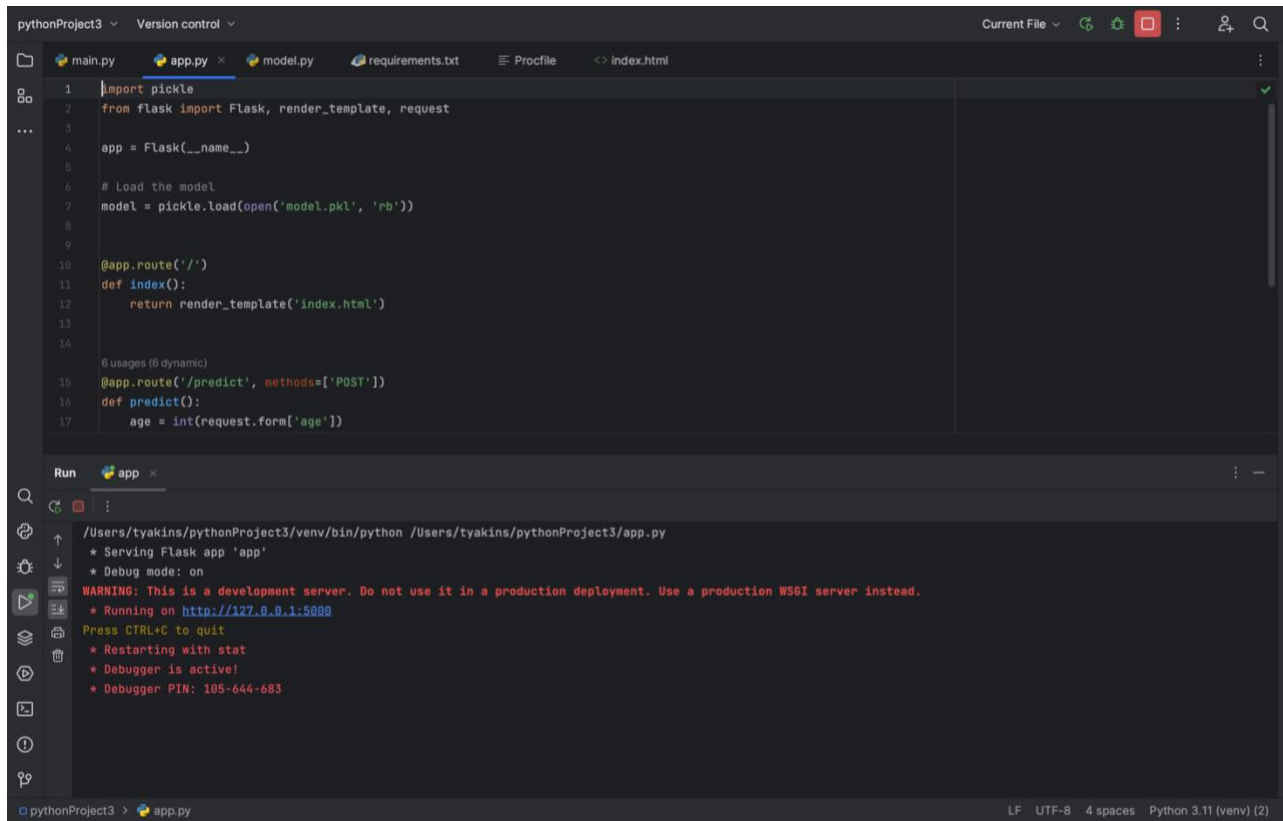


Figure 3.2: Home.html

- Background Image, pontential.jpg located in static/images contains an image that serves as the background instead of a plain color.

## 5 Running Procedure

Once we have done all the above, we can start running the API by either double clicking run.



The screenshot shows a code editor with a file named `app.py` open. The code is a Flask application that loads a model from a pickle file and has two routes: `/` and `/predict`. The `/predict` route takes an age as input and returns a prediction. The application is running in a terminal window, showing the command `python /Users/tyakins/pythonProject3/venv/bin/python /Users/tyakins/pythonProject3/app.py` and the output: `* Serving Flask app 'app'`, `* Debug mode: on`, `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`, `* Restarting with stat`, `* Debugger is active!`, and `* Debugger PIN: 105-644-683`.

Figure 5.1 Command Execution

Next, we head over to the link provided.

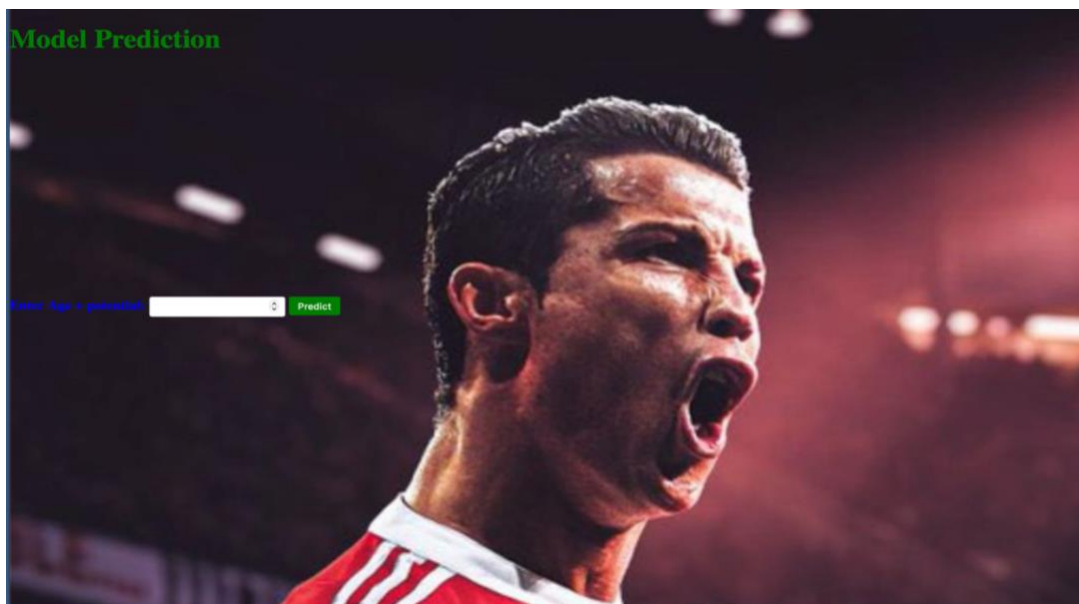


Figure 5.2: Homepage



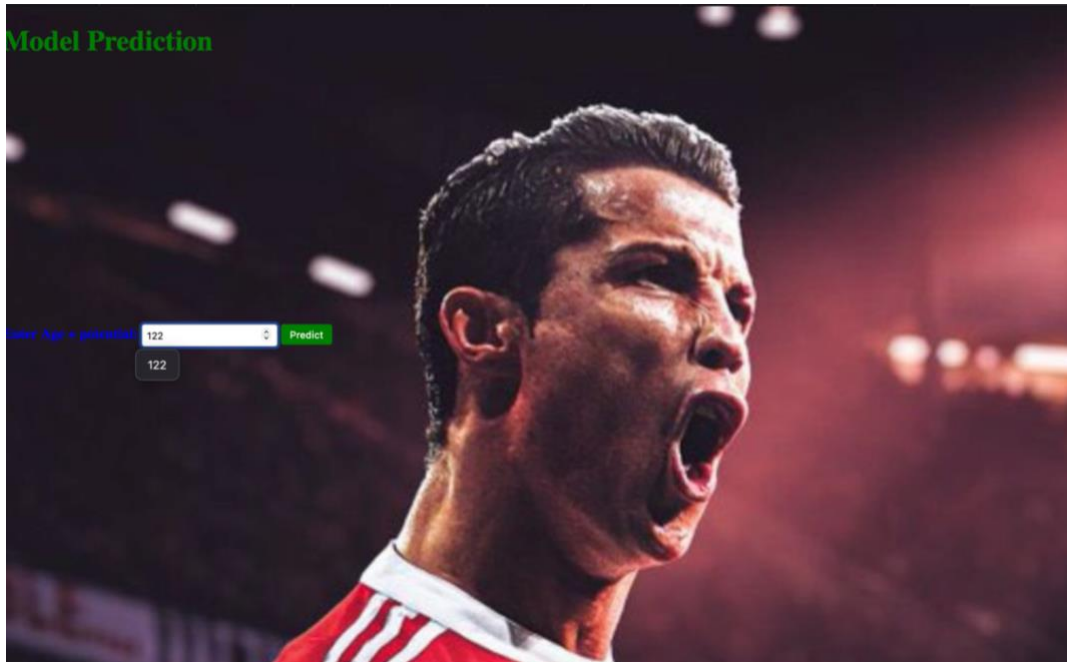


Figure 5.3: Input in The Comments Form

After entering the input click the predict button now, we can the result of our input.

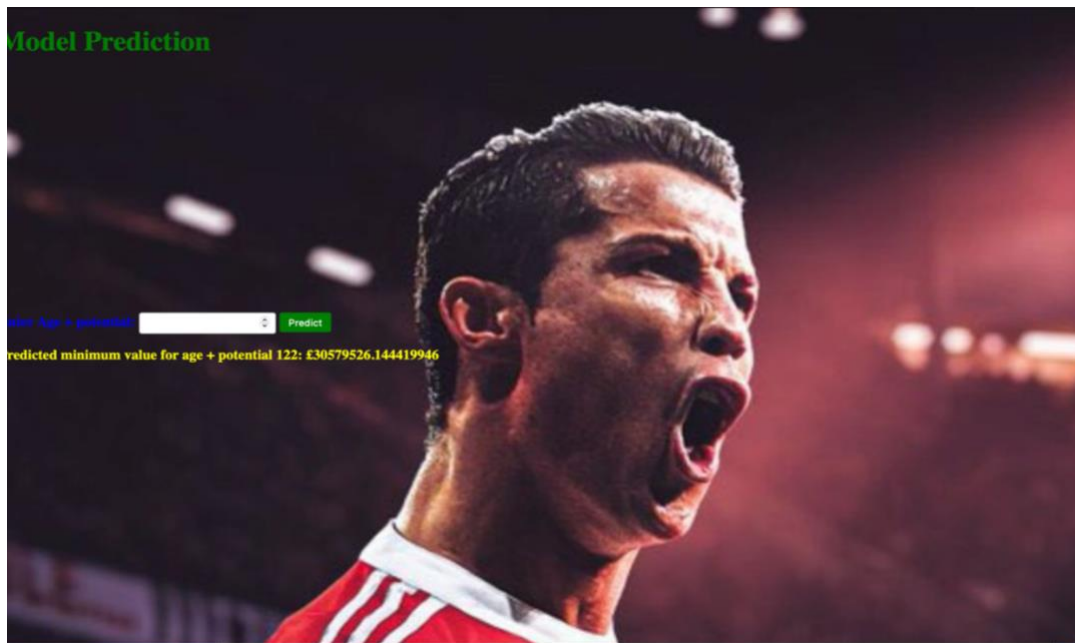


Figure 3.7: Result of Input