Project Report

Name: 500_Person_Gender_Height_Weight_Index
Report date: June 28, 2023
Internship Batch: LISUM22
Version: 1.0
Project by: Pavan Walvekar
Data intake reviewer:
Data storage location: https://www.kaggle.com/yersever/500-person-gender-height-weight-bodymassindex
Project Location: https://github.com/walvekarpavan

Context

The body mass index (BMI) is a numerical value that takes into account a person's weight and height. It is calculated by dividing the weight of the body in kilograms by the square of the height in meters. The BMI is expressed in units of kilograms per square meter (kg/m^2).

Content

The dataset contains information about gender, height, weight and BMI index of individuals

Gender : Male / Female

Height : Number (cm)

Weight : Number (Kg)

Index

0 - Extremely Weak

1 - Weak

2 - Normal

3 - Overweight

4 - Obesity

5 - Extreme Obesity

1. Modeling the dataset “500_Person_Gender_Height_Weight_Index.csv”

/Users/pavanwalvekar/Documents/DataGlacier/WEEK_4_Flask/model.py

model.py* | app.py | index.html

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

bmi_prediction = pd.read_csv("/Users/pavanwalvekar/Documents/Data Glacier Intern/WEEK 4 Fla


X = bmi_prediction[['Height', 'Weight']]
y = bmi_prediction['Index']

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.4, random_state=3)

logistic_reg = LogisticRegression(solver='liblinear')
logistic_reg.fit(X_train, y_train)
print(logistic_reg.predict(X_test))
pickle.dump(logistic_reg, open('model.pkl', 'wb'))
```

2. HTML codes (index.html)

/Users/pavanwalvekar/Documents/DataGlacier/WEEK_4_Flask/index.html

model.py* | app.py | index.html

```html
 <!DOCTYPE html>
<html >
<!--From https://codepen.io/frytyler/pen/EGdtg-->
<head>
  <meta charset="UTF-8">
  <title>Body Mass Index (BMI) Prediction</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
<link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
 </head>

<body>
 <div class="login">
    <h1> Body Mass Index (BMI) Prediction </h1>
    <h3> Please Enter Your Height and Weight</h3>

<!-- Main Input For Receiving Query to our ML -->
    <form action="{{ url_for('predict')}}"method="post">
    <input type="text" name="Height" placeholder="height in cm" required="required" />
    <input type="text" name="Weight" placeholder="weight in kg" required="required" />

    <button type="submit" class="btn btn-primary btn-block btn-large">BMI Predict</button>
    </form>
<br>
<br>

   {{prediction_text}}

 </div>
</body>
</html>
```

3. app.py

/Users/pavanwalvekar/Documents/DataGlacier/WEEK_4_Flask/app.py

model.py* | app.py | index.html

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

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

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


@app.route('/predict',methods=["POST"])
def predict():
    '''
    For rendering results on HTML GUI
    '''
    int_features = [int(x) for x in request.form.values()]
    final_features = [np.array(int_features)]
    prediction = model.predict(final_features)

    if prediction == 0:
        output = "Extremely Weak. Please go to a doctor to check."
    elif prediction == 1:
        output = "Weak"
    elif prediction == 2:
        output = "Normal"
    elif prediction == 3:
        output = "Overweight"
    elif prediction == 4:
        output = "Obesity. Please go to a doctor to check."
    elif prediction == 5:
        output = "Extreme Obesity. Please go to a doctor to check."
    return render_template("index.html", prediction_text = output)


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

4. Converting notebook to .py file and running python code

WEEK_4_Flask — python ◂ python app.py — 80×24

```
ls: Intern: No such file or directory
(base) pavanwalvekar@Pavans-MacBook-Pro ~ % cd Documents/
(base) pavanwalvekar@Pavans-MacBook-Pro Documents % Data Glacier Intern
zsh: command not found: Data
(base) pavanwalvekar@Pavans-MacBook-Pro Documents % cd Data Glacier Intern
cd: too many arguments
(base) pavanwalvekar@Pavans-MacBook-Pro Documents % cd DataGlacier
(base) pavanwalvekar@Pavans-MacBook-Pro DataGlacier % cd WEEK_4_Flask
(base) pavanwalvekar@Pavans-MacBook-Pro WEEK_4_Flask % app.py
zsh: command not found: app.py
(base) pavanwalvekar@Pavans-MacBook-Pro WEEK_4_Flask % python app.py
/opt/anaconda3/lib/python3.8/site-packages/sklearn/base.py:318: UserWarning: Try
ing to unpickle estimator LogisticRegression from version 0.24.2 when using vers
ion 1.2.2. This might lead to breaking code or invalid results. Use at your own
risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-
limitations
  warnings.warn(
 * 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
```

5. Examples of the model