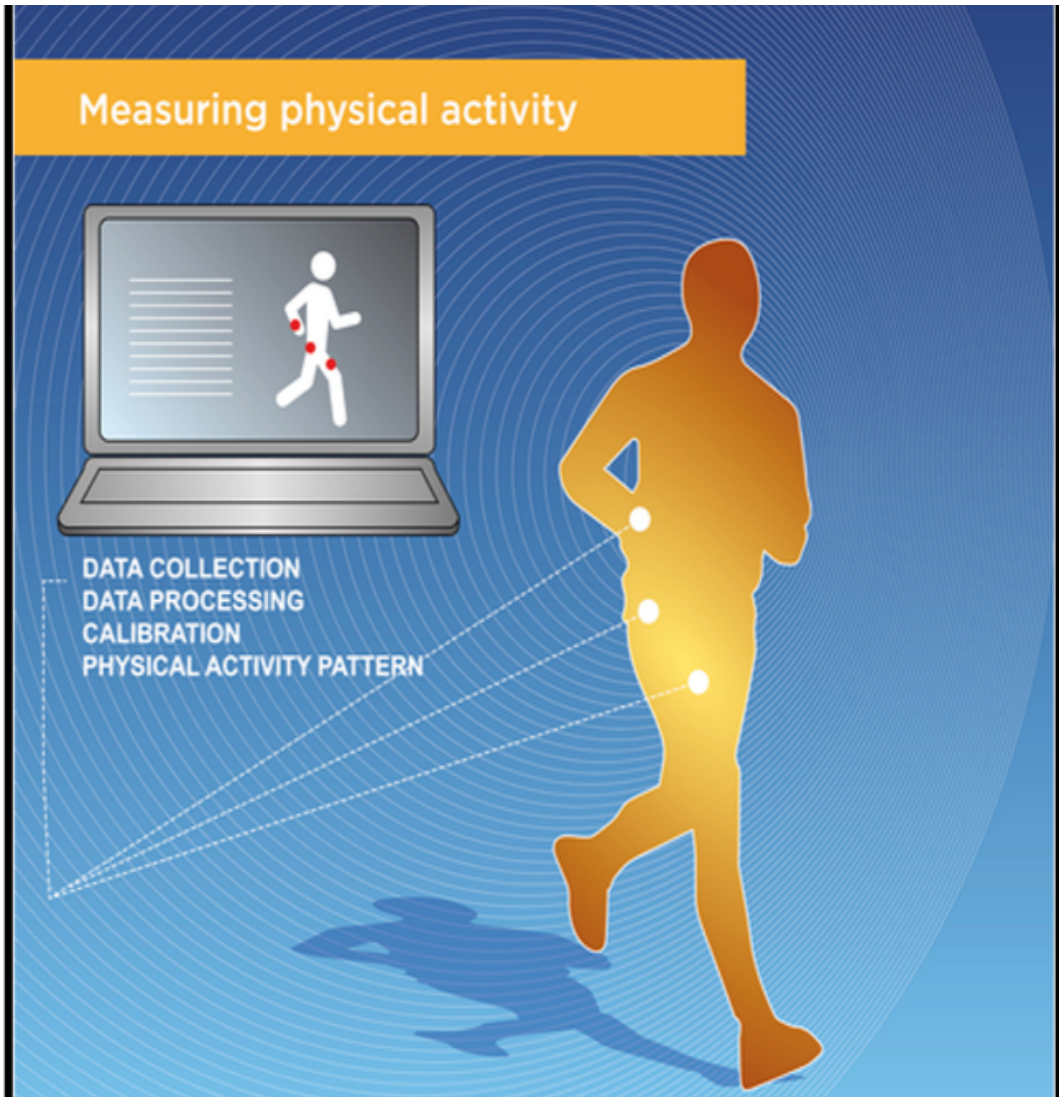
PHYSICAL ACTIVITY FITNESS PREDICTION



1. INTRODUCTION

1.1 Overview

The use of mobile fitness apps has been on the rise for the last decade and especially during the worldwide SARS-CoV-2 pandemic, which led to the closure of gyms and to reduced outdoor mobility. Fitness apps constitute a promising means for promoting more active lifestyles, although their attrition rates are remarkable and adherence to their training plans remains a challenge for developers.

1.2 Purpose

Sedentary lifestyle is defined by the absence of physical activity practices throughout the day and causes a decrease in caloric expenditure. This behavior is explained by the inappropriate lifestyle, for example, too much time sitting or lying down and still eating unhealthy foods during this time of immobilization. Currently, a third of the adult world population is physically inactive and this generates 5 million deaths per year (The LAncet, 2012). In addition to contributing to several chronic diseases, physical inactivity also influences mood, sleep quality and body weight

The main purpose of this project is to analyze the parameters of various classification algorithms and compare their predictive accuracies to discover the best classifier for determining the physical fitness.

2.LITERATURE SURVEY

2.1 Existing Problem (OR) Problem Statement

Given a dataset containing various attributes of 96 Adults, use the features available in the dataset and define a supervised classification algorithm which can identify whether a person is physically fit or not.

2.2 Proposed Solution

This a classic example of supervised learning. We have been provided with a fixed number of features for each data point, and our aim will be to train a variety of Supervised Learning algorithms on this data, so that, when a new data point arises, our best performing classifier can be used to categorize the data point as a positive example or negative. Exact details of the number and types of algorithms used for training is included in the 'Algorithms and Techniques' sub-section of the 'Analysis' part.

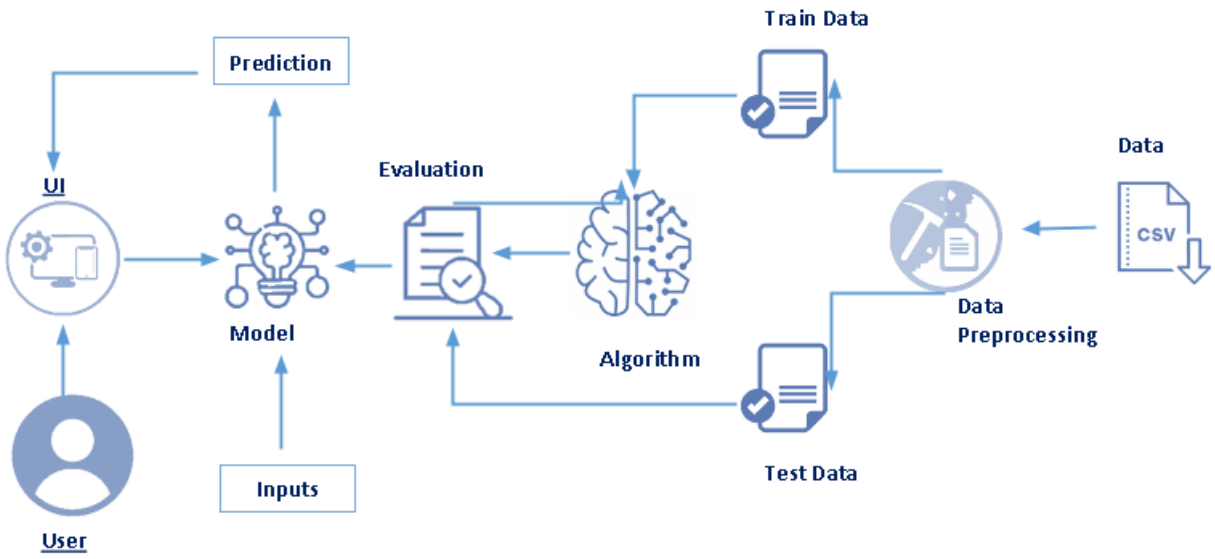
This project focuses on the related works of various authors on liver disease such that algorithms were implemented using Jupyter that is a machine learning software written in Python. Various attributes that are essential in the prediction of physical fitness were examined and the dataset of the adults were also evaluated. This project compares various classification algorithms such as Linear Regression, Logistic Regression. Random Forest Regression / Classification, Decision Tree Regression / Classification with an aim to identify the best technique.

Based on this study, Decision Tree with the highest accuracy outperformed the other algorithms and can be further utilized in the prediction of physical fitness recommended to the user.

Later by using Flask app create html files and create an user interface to display whether the patient has liver problem or not.

3.THEORITICAL ANALYSIS

3.1 Block Diagram



3.2 Hardware / Software designing

The following is the Hardware required to complete this project:

- Internet connection to download and activate
- Administration access to install and run Anaconda Navigator
- Minimum 10GB free disk space
- Windows 8.1 or 10 (64-bit or 32-bit version) OR Cloud: Get started free, *Cloud account required.

Minimum System Requirements To run Office Excel 2013, your computer needs to meet the following minimum hardware requirements:

- 500 megahertz (MHz)
- 256 megabytes (MB) RAM
- 1.5 gigabytes (GB) available space
- 1024x768 or higher resolution monitor

The following are the software s required for the project:

1. Jupyter Notebook
2. Spyder
3. Microsoft Excel 2013

4.EXPERIMENTAL INVESTIGATIONS

Coming to analysis or investigations three supervised learning approaches are selected for this problem. Care is taken that all these approaches are fundamentally different from each other, so that we can cover as wide an umbrella as possible in term of possible approaches.

For each algorithm, we will try out different values of a few hyper parameters to arrive at the best possible classifier. This will be carried out with the help of grid search cross validation technique.

There are several Machine learning algorithms to be used depending on the data you are going to process such images, sound, text, and numerical values. The algorithms that you can choose according to the objective that you might have may be classification algorithms and Regression algorithms.

1. Decision Tree Algorithm

A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.

(2) Build the model with the Decision Tree Classifier.

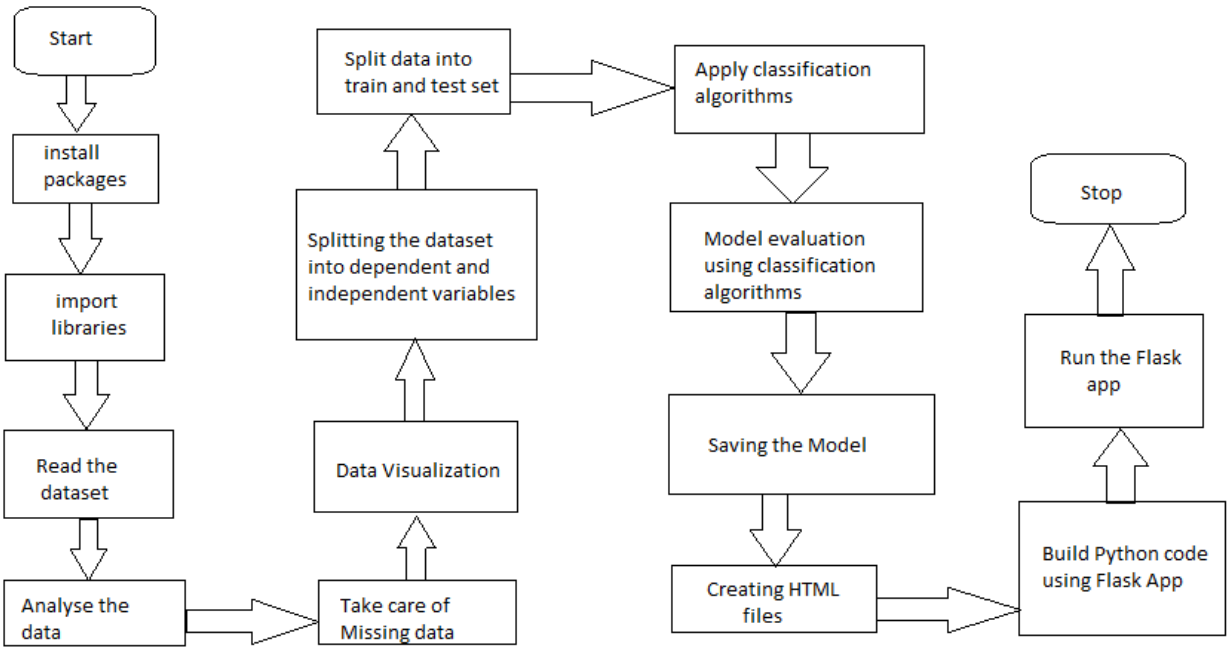
We're going to use x_train and y_train obtained above in train_test_split section to train our decision tree classifier model. We're using the fit method and passing the parameters as shown below.

(3) Predict the values

Once the model is trained, it's ready to make predictions. We can use the predict method on the model and pass x_test as a parameter to get the output as dt_y_train.

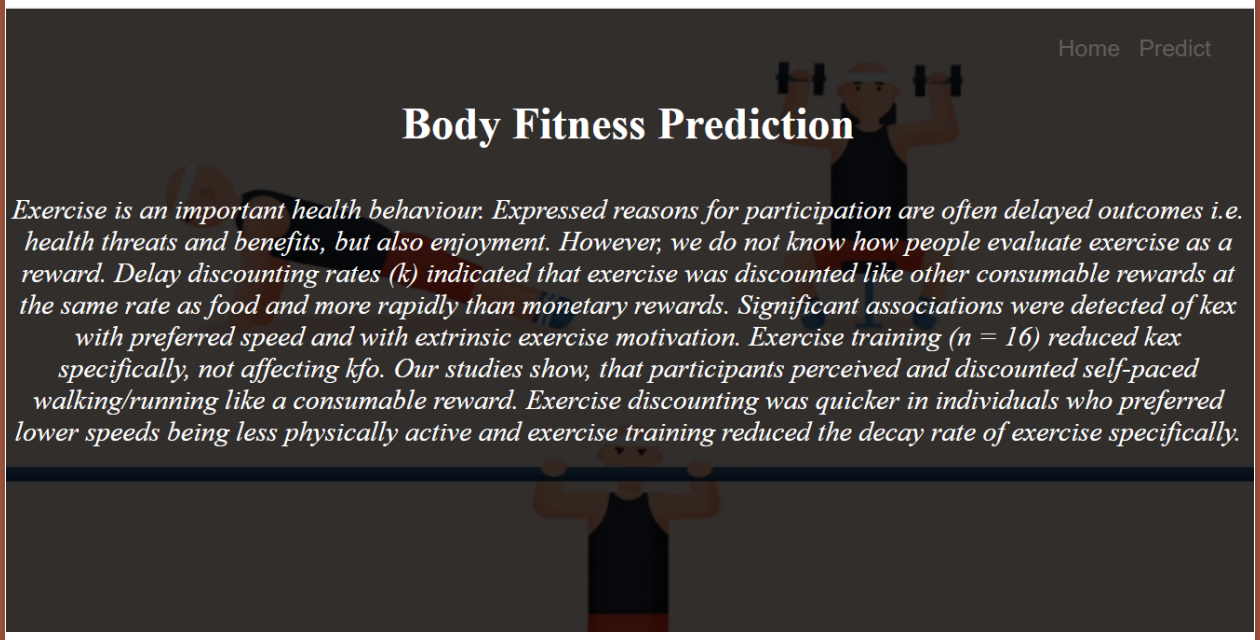
Notice that the prediction output is an array of real numbers corresponding to the input array.

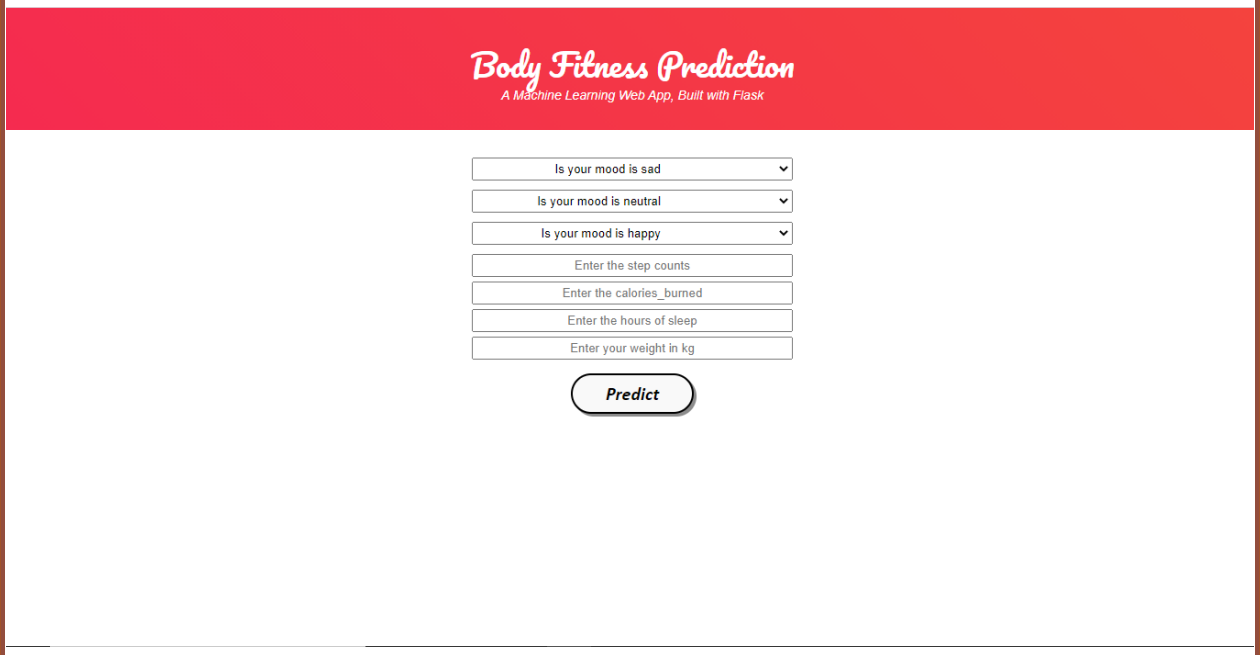
5.FLOW CHART



6.RESULT

This is our home page where we get to know the summary of the project.





Body Fitness Prediction

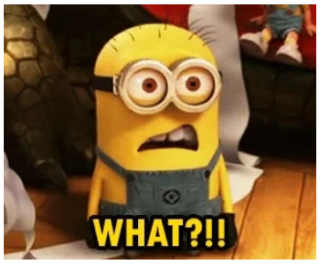
A Machine Learning Web App, Built with Flask

Prediction: Great! You are ACTIVE.....

Body Fitness Prediction

A Machine Learning Web App, Built with Flask

Prediction: Oops! You are NOT ACTIVE!!!!!!!!!!!! WORK HARD



As we see the predicted output is displayed on the User Interface

7.ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

1. **Efficiency in workflow:** One of the first desires that probably comes to mind is efficiency. When building your website, you want to be able to reach as many people as you can. You also want to reach them on a consistent basis in a way that doesn't involve you spending all the your time waiting at the hospital for the long time.

2. **Reduce costs:** You don't need a large time to wait for the results for the entire day. They can approach hospital when they're ready to get about their condition and you don't need additional teams to get to know about results. Paying for a large team to constantly contact prospects isn't needed.

3. It's commonly known that people have taken fitness to their apartments and houses. Interactive home gym fitness apps are a more viable option for those who are not able to visit a fitness center, especially in pandemic times. Talking about the big players on the home fitness software market, needless to say that AI features are utilized by the majority of them.

DISADVANTAGES:

1. Any single error in data set can change the entire data.

2. Correct accuracy must be needed while doing the project using supervised machine learning algorithms.

3. Python code should be correct without any error.

8. APPLICATIONS:

This application can further be developed with more idea and implementation and by using different algorithms. The accuracy score of the model can be further improved by using Random Forest and also by increasing the data set, K-Nearest Neighbors algorithm is also one of the pertinent methods which can be used to predict accurately. It proposes to improve the accuracy further.

9.CONCLUSION

The purpose of this study is to review research papers about machine learning techniques in predicting physical activities from fitness data. Currently, there exist so many techniques for predictions in machine learning, but we observed three models that most appropriate for predicting physical activities using fitness data. We also identified few parameters and features that most appropriate to be used in predicting the suitable physical activity based on personal context. We found two models that considered important parameters in their process for predicting physical activities not only using fitness data but also using personal context data. We believe there is a need to develop an application for predicting suitable physical activities using fitness data and personal context data.

The framework describes the process to predict suitable physical activity based on fitness data and personal context collected from the users. We are currently working on developing fitness personalization application for predicting physical activities that suitable based on features of individual. The fitness personalization application will predict based on personal context such as age, and fitness data are collected from wearable devices, such as number of walking steps and number of calories burnt. It is hoped that this application will encourage people to continue maintaining their health with suitable physical activities according to their personal health context.

10.FUTURE SCOPE

In the future, the local interpretable model-agnostic explanation (LIME) method will be used to understand the model's interpretability. Instead of binary classification, one may use multinomial classification by separating the types of fitness they are lacking. In this way, each model's performance can be compared. The described ML methods can assist health sectors to achieve a better fitness providing effective results in identifying groups. Moreover, ML methods are data driven, and they directly use diagnostic variables from patients' medical tests. Thus, it i is a more reliable process. The applied ML methods in this project can save time, costs, and potentially lives for the betterment of physical fitness.

11. BIBILOGRAPHY

We referred some books and surfed the internet for the better outcome of the project

1. W. Raghupathi, V. Raghupathi, Big data analytics in healthcare: promise and potential, Health information science and systems (2014), 2-3.

2. A. Charleonnan, T. Fufaung, T. Niyomwong, W Chokchueypattanakit, S. Suwannawach, N. Ninchawee “Predictive Analytics for Chronic Kidney Disease Using Machine Learning Techniques” MITiCON2016.

3. Youtube videos by simplilearn, Datacamp, Codebasics, eduraka!.

4. Smartinternz tutorial classes help me for completion of project.

APPENDIX

A. SOURCE CODE:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
import pickle
data=pd.read_csv("C:/Users/Sunny/OneDrive/Desktop/smart bridge/25.csv")
data.shape
data.head()
data.tail()
data.columns
data.info()
data.describe()
np.unique(data.dtypes,return_counts=True)
def category(data):
```

```
    for i in data.columns:
        print(i)
        print(data[i].unique())
        print("x"*90)
category(data)
data["bool_of_active"].unique() # 0 = inactive , 500 = active
data["mood"].unique() # 100 = sad , 200 = neutral  , 300 = happy
data['weight_kg'].unique()
data['hours_of_sleep'].unique()
data.isnull().any()
data.isnull().sum()
sns.heatmap(data.isnull(),cbar=False)
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
data['bool_of_active']=le.fit_transform(data['bool_of_active'])
data
data.boxplot(column="step_count")
data.boxplot(column="calories_burned")
x=pd.DataFrame(data.iloc[:,[1,2,3,4,6]])
#Dependent Variable
y=pd.DataFrame(data.iloc[:,5])
columnTransformer = ColumnTransformer([('encoder',
                                  OneHotEncoder(),
                                  [1])],
                                 remainder='passthrough')

x = pd.DataFrame(columnTransformer.fit_transform(x),dtype =
np.str,columns=['sad','neutral','happy','step_count','calories_burned','hours_of_sleep','weight_kg'])
x
x.shape
from sklearn import model_selection, neighbors
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.1,random_state=0)
from sklearn.tree import DecisionTreeClassifier
dtc = DecisionTreeClassifier(max_depth=12)   #max_leaf_nodes=26  max_depth=12
dtc.fit(x_train,y_train)
dt_y_train=dtc.predict([['0.0', '0.0', '1.0', '4435.0', '141.0', '5.0', '64.0']])
dt_y_train
dtc.score(x_train,y_train)
import pickle
pickle.dump(dtc, open('fitness.pkl','wb'))
```

```
model=pickle.load(open("fitness.pkl","rb"))
```

FLASK APP CODE:

```
# importing the necessary dependencies
import numpy as np #used for numerical analysis
import pandas as pd # used for data manipulation
from flask import Flask, render_template, request
# Flask-It is our framework which we are going to use to run/serve our application.
#request-for accessing file which was uploaded by the user on our application.
import pickle


app = Flask(__name__) # initializing a flask app
model = pickle.load(open('fitness.pkl', 'rb')) #loading the model

@app.route('/')# route to display the home page
def home():
    return render_template('home.html') #rendering the home page
@app.route('/Prediction',methods=['POST','GET'])
def prediction():
    return render_template('indexnew.html')
@app.route('/Home',methods=['POST','GET'])
def my_home():
    return render_template('home.html')

@app.route('/predict',methods=['POST'])# route to show the predictions in a web UI
def predict():

    #reading the inputs given by the user
    input_features = [float(x) for x in request.form.values()]
    features_value = [np.array(input_features)]

    features_name = ['sad','neutral','happy','step_count',
                     'calories_burned','hours_of_sleep','weight_kg']

    df = pd.DataFrame(features_value, columns=features_name)

    # predictions using the loaded model file
```

```
    output = model.predict(df)

    # showing the prediction results in a UI# showing the prediction results in a UI
    return render_template('result.html', prediction_text=output)

if __name__ == '__main__':
    # running the app
    app.run(debug=False)

runfile('C:/Users/Sunny/OneDrive/Desktop/Flask App/app.py',
wdir='C:/Users/Sunny/OneDrive/Desktop/Flask App')
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   Use a production WSGI server instead.
 * Debug mode: off
C:\Users\Sunny\anaconda3\lib\site-packages\sklearn\base.py:310: UserWarning: Trying to unpickle
estimator DecisionTreeClassifier from version 0.22.1 when using version 0.24.2. This might lead to
breaking code or invalid results. Use at your own risk.
  warnings.warn(
 * Running on http://127.0.0.1:5000/
```

Submitted by,
Vallamreddy Saketh Reddy