BODY FITNESS PREDICTION USING IBM WATSON

1. INTRODUCTION

1.1 Overview

An active life is important to stay healthy. Staying active helps to reduce your risk of both diseases and stress. Both physical and mental aspects contribute to a person's activeness. The number of steps they take in a day has a great influence on their vitality. Doing exercise is one of the crucial steps to improve activeness. Along with that, many other factors such as mood, sleep quality and weight also affect ones activeness. Sedentary lifestyle causes absence of exercise and inappropriate lifestyle.But what if we have a model that predicts ones activeness based on their step count.

This project is purely a web app that estimates whether exercise or workout improves a person's activeness. You create a model from a data set that includes the step count, hours of sleep, calories burned, mood and weight to predict one's activeness. The user creates IBM Watson Studio Service, IBM Cloud Object Storage Service on IBM Cloud. The user uploads the body fitness data file into Watson Studio. We create a machine learning model and deploy it to the user. The model will predict whether the person is active or not based on the information provided by the user.

1.2 Purpose

The purpose of the project is to predict the activeness of one person. For this we are building a model using a dataset consists of step count, hours of sleep, calories burned, mood and weight of an individual. Analysis made on these data helps to discover the effect of these aspects on ones activeness. By using this application one can know his or her status and their by they can increase their step count to improve their activeness. This application can be used in medical filed and their by analyze the condition of patients and doctor can suggest treatments.

2. LITERATURE SURVEY

2.1 Existing Problem

Today we are estimating someone's activeness by examining their physical or mental condition. We are considering different parameters to measure activeness. Timely changes in these parameters are difficult to record. So, It will be difficult to estimate their activeness by manual examination.
Also, people are very slothful in improving their activeness in this sedentary life. They are unaware of their condition, so visiting hospital and affecting disease is now in common.

2.1 Proposed Solution

This project aims at building a web app that automates the estimation of one's activeness**.** By using advanced machine learning model which is trained using verified dataset of both active and inactive people, the model is trained to predict one's activeness by providing necessary information. The model can predict the result in seconds and in a reliable manner. This model can predict someone's states at any moment. It will help them to take care of themselves to improve their fitness and their by they can get rid of diseases and hospital visits.

3. THEORETICAL ANALYSIS

3.1 Block Diagram

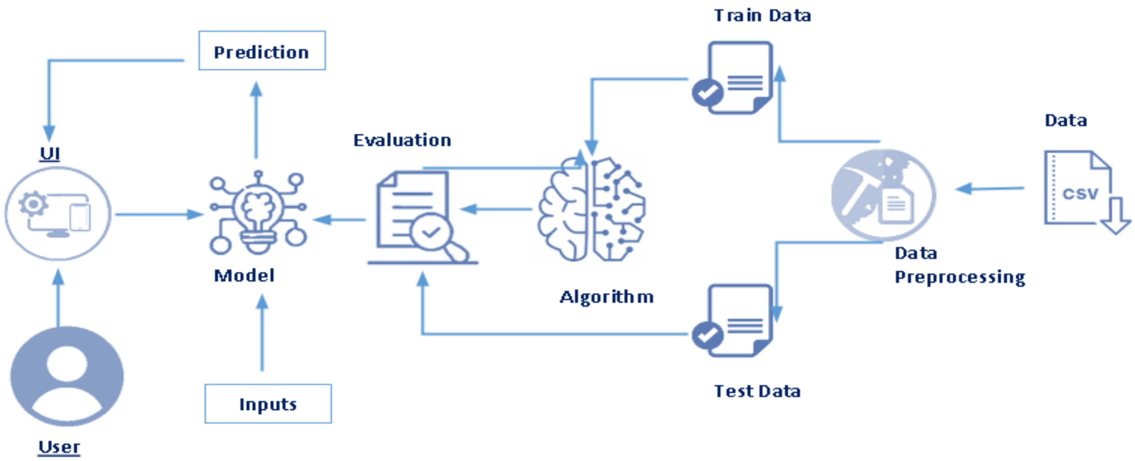


Fig1: Block Diagram

3.2 Hardware / Software Designing

- IBM Watson Studio

IBM Watson studio provides tools for data scientists, application developers and experts to collaboratively and easily work with data to build and train models at scale. It gives you the flexibility to build models where your data resides and deploy anywhere In a hybrid environment.

- IBM Watson Machine Learning

Watson Machine Learning provides a full range of tools and services so that you can build, train, and deploy Machine Learning models.

- IBM Cloud Object Storage

A service offered by IBM for storing and accessing unstructured data. It allows us to store limitless amounts of data, simply and cost effectively.

4. EXPERIMENTAL INVESTIGATIONS

Our aim is to build a machine learning model that estimates the activeness of an individual. For this purpose, we collect the dataset for training purposes. From the dataset we have chosen the necessary attributes that contribute to our prediction. Here we choose 5 parameters such as mood, step count, calories burned, hour of sleep and weight that affects our prediction. After performing some preprocessing on these data we are building our model.

A number of algorithms are available for building the model. We choose the algorithm by analyzing the accuracy rate on our date. In this project we are selecting Decision Tree model. Once the model is trained, it is ready to make predictions.

5. FLOWCHART

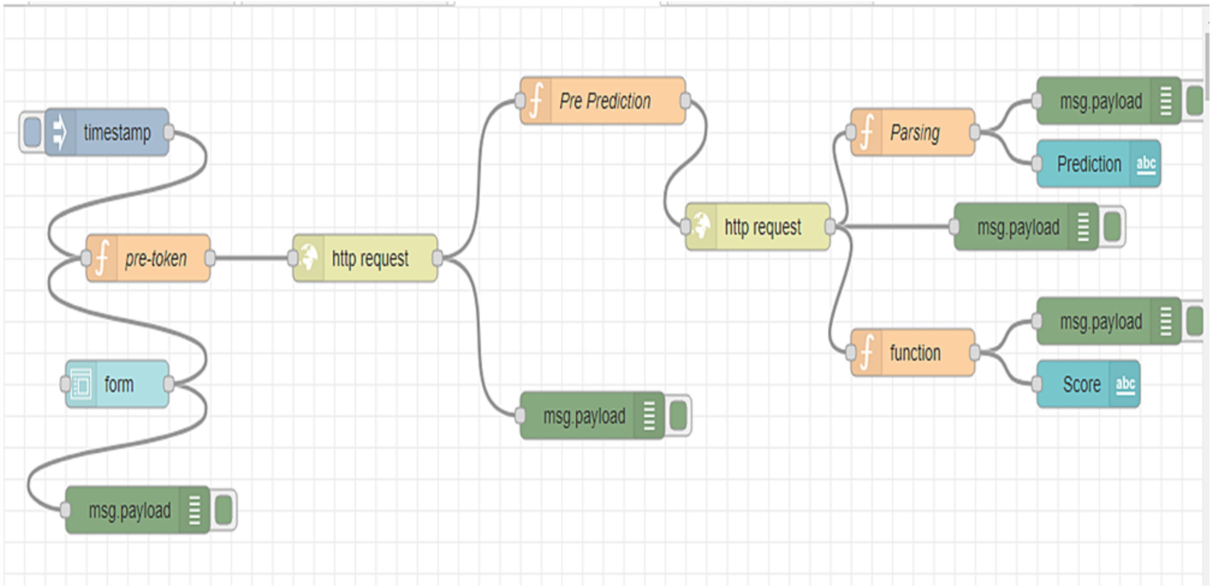
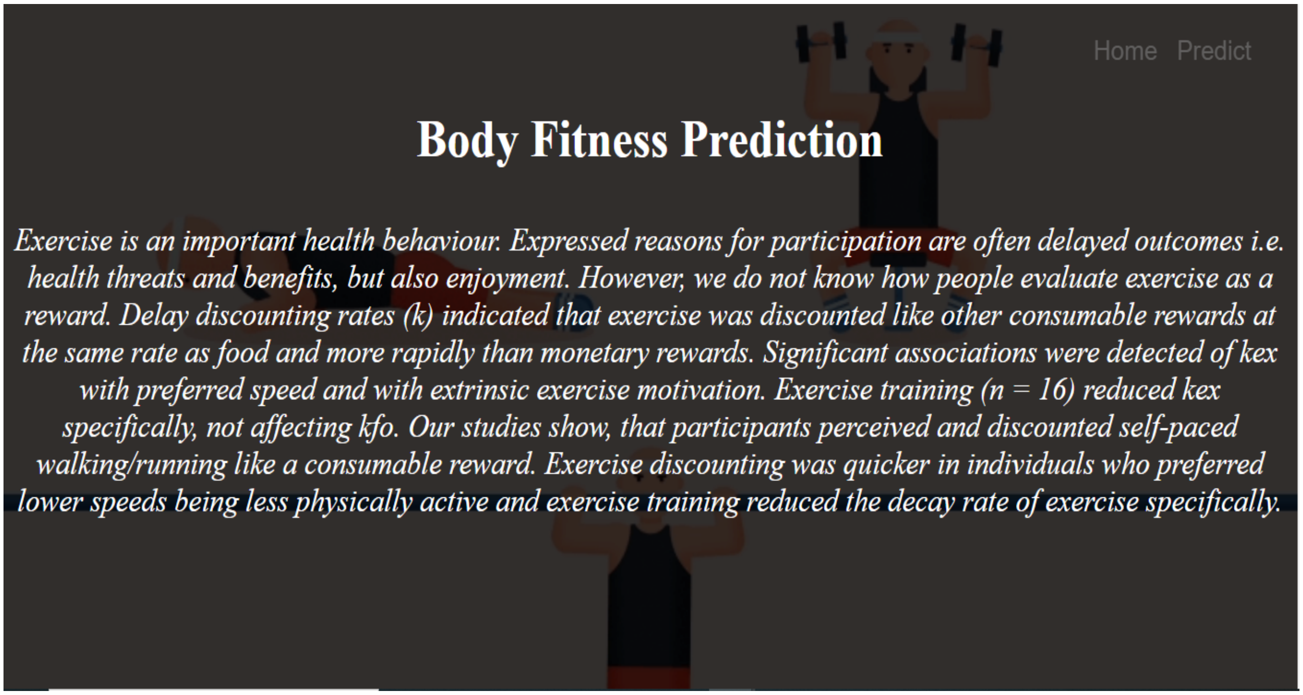


Fig2 : Flowchart

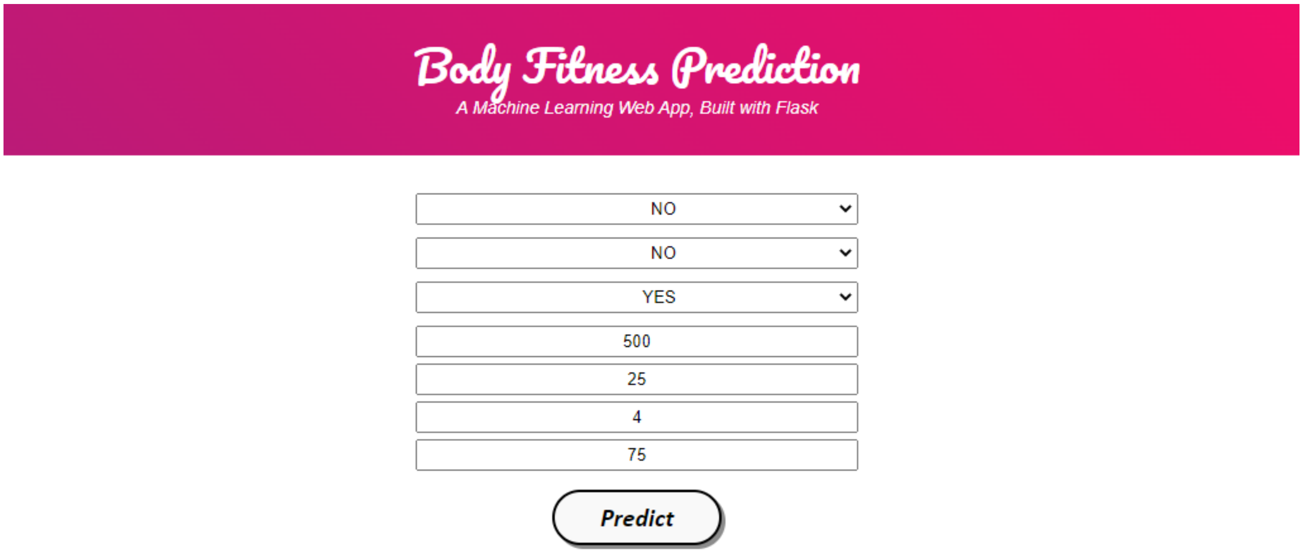
6. RESULT



Body Fitness Prediction

Exercise is an important health behaviour. Expressed reasons for participation are often delayed outcomes i.e. health threats and benefits, but also enjoyment. However, we do not know how people evaluate exercise as a reward. Delay discounting rates (k) indicated that exercise was discounted like other consumable rewards at the same rate as food and more rapidly than monetary rewards. Significant associations were detected of kex with preferred speed and with extrinsic exercise motivation. Exercise training (n = 16) reduced kex specifically, not affecting kfo. Our studies show, that participants perceived and discounted self-paced walking/running like a consumable reward. Exercise discounting was quicker in individuals who preferred lower speeds being less physically active and exercise training reduced the decay rate of exercise specifically.

Fig3 : Home page



Body Fitness Prediction

A Machine Learning Web App, Built with Flask

NO

NO

YES

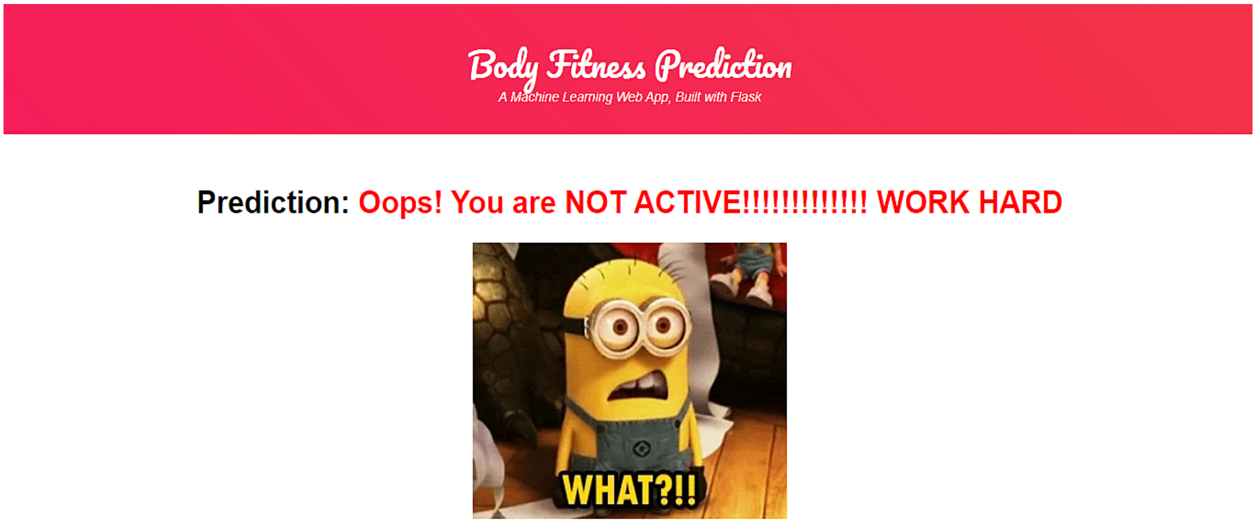
500

25

4

75

Fig4 : Predict page



Body Fitness Prediction

A Machine Learning Web App, Built with Flask

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

Fig5 : Result page

7. ADVANTAGES & DISADVANTAGES

7.1 Advantages

- Helps to track daily activities.
- This system helps us to improve our physical and mental health by setting goals and working towards achieving those goals.
- Cost effective.
- Save time.

7.2 Disadvantages

- Attributes and dataset collected should be correct. Otherwise it will affect the accuracy of our model.
- Require more dataset to increase the accuracy of model.

8. APPLICATIONS

An individual can use this application as a fitness tracker to improve their fitness. It can also be used in the medical industry to understand the condition of patients. It will reduce pressure on medical staff and save time, providing great convenience in medical treatments.

9. CONCLUSION

In the project we are building a machine learning model to predict the fitness of a person. A proper dataset is collected for our training purpose. Here we are using Decision Tree as the model. Various factors were used and their effect on the prediction was examined. Attributes that had no effect on the prediction were removed. After training, our model is tested with new data. Data that is provided should be significant then only our prediction will be accurate.

10. FUTURE SCOPE

When people are moving through a sedentary lifestyle, this prediction system will help them improve their both physical and mental health. Wearables with fitness tracker available on the market are one of the extensions of this system. It will help them to track their daily activities easily. An improved form of this application developed and used in the health sector for better treatments. Also, in future we can add more parameters that contribute to a better prediction.

11. BIBILIOGRAPHY

1. www.wikipedia.org
2. www.peerspot.com
3. https://www.healthline.com/health/womens-health-active-lifestyle
4. https://www.cavosmart.com/future-fitness-tracker-and-medical/

12. APPENDIX

Source Code

```python
# 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
import json
import requests
import os

# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "3QXKp5BxGwIzwS0TySElux6PSEZvCqbmnPkwZrkjz2c6"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey": API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]
header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}

# NOTE: manually define and pass the array(s) of values to be scored in the next line

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)
            payload_scoring = {"input_data": [{"fields": [['sad','neutral','happy','step_count',
                                               'calories_burned','hours_of_sleep','weight_kg']], "values": input_features}]}

      response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/51541a6f-cd48-45e7-
```

```python
86bb-6eb9ff91e676/predictions?version=2022-03-26', json=payload_scoring, headers={'Authorization': 'Bearer ' + mltoken})
    print("Scoring response")
    print(response_scoring.json())

if __name__ == '__main__':
    # running the app
    port=int(os.environ.get('PORT',5000))
    app.run(port=port,debug=False,use_reloader=False)
```