IMPLEMENTATION OF PERSONAL FITNESS TRACKER USING PYTHON

A Project Report

submitted in partial fulfillment of the requirements

of

AICTE Internship on AI: Transformative Learning

With

TechSaksham – A joint CSR initiative of Microsoft & SAP

By

Dudam Amulya,

dudamamulya09@gmail.com

Under the Guidance of

Saomya Chaudhary

ACKNOWLEDGEMENT

I would like to take this opportunity to express my heartfelt thanks to everyone who supported me throughout the completion of my project, "**Implementation of Personal Fitness Tracker using Python."**

Firstly, I want to thank my project mentor, Saomya chaudhary, for their constant guidance, support, and helpful suggestions throughout this project. Their advice has been a key factor in making this project a success.

I am also grateful to edunet Foundation and TechSaksham for providing all the resources and tools needed to work on this project. The access to different software and platforms really helped me improve the project.

I would like to thank all the mentors, for their continuous support and feedback. They always offered great ideas and kept me motivated throughout the process.

A special thanks to the creators of the Python libraries and tools like NumPy, Matplotlib etc. These open-source tools were essential in building the fitness tracker.

Lastly, I want to thank my family for their patience, encouragement, and understanding during this time. Their support has always kept me going.

Thank you all for your help and belief in me.

ABSTRACT

Regarding the growing significance of fitness and health in modern times, having tools for monitoring physical activity is crucial for helping people achieve their wellness goals. The main objective of this project is to implement Python to develop an online personal fitness tracker that can forecast a user's calories burned based on a range of input parameters.

Users may provide important personal information which involves age, gender, body temperature, heart rate, BMI (body mass index), activity duration, and steps taken using this web-based program. The tracker uses this data to estimate the number of kilograms of calories burned during activity through predictive modelling. The Random Forest Regressor, which is one of the machine learning algorithms utilized in the application's backend, trains the model using historical data from a dataset of exercise activities with associated calorie expenditure.

Several Python libraries are utilized by the program to boost its functionality. Pandas is used to process and organize input data for data analysis and modification, and matplotlib and seaborn are utilized to generate attractive and easily searchable graphs that show users' fitness progress over time. Streamlit is used in the front-end interface's design to facilitate smooth user-application interaction. Using the simple and user-friendly sidebar, users can adjust the different parameters and get immediate input on what number of calories they will burn.

In addition, the application has real-time progress proofs, boosting the interactive and engaging user experience. A progress bar that indicates the app's processing status when users input data improves user satisfaction and involvement overall. After the calculation is finished, the tracker displays a predicted number of calories burned, which is determined by comparing the user's actual metrics with previous information from other users and provide graphs that useful to better understand their conditions based on inputs and give suggestions and ideas to improve their fitness. This app allows user to set the target to burn calories burned.

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER 1

Introduction

1.1 Problem Statement:

The need for a simple to use data-driven fitness tracking application that provides users accurate predictions of the number of calories used during exercise based on their own inputs is the problem this project aims to solve. In order provide useful information into how their fitness levels compare to others, the solution should not only allow users submit important parameters like age, gender, heart rate, and exercise duration, but also compare their data with a larger dataset.

In addition, current fitness tracking apps frequently come short in providing interacting real-time feedback or predictions with a simple user interface. Without clear insights into their progress, users may find it hard to understand raw data analysis or get motivated.

In order to address these problems, this project uses Python and machine learning to create a personal fitness tracker that enables users to:

- Enter personal information like body temperature, heart rate, gender, age, BMI, and exercise history.
- Use a trained machine learning model (Random Forest Regressor) to forecast the number of calories burned depending on these parameters.
- To offer unique insights, compare a person's data with those of others in the same demographic.

1.2 Motivation:

The increasing demand for personal fitness monitors that provide clear, data-driven insights motivated selecting of this project. Customized feedback based on personal health elements can be missing from current fitness apps. This project aims to help users better track their fitness progress by predicting calories burned based on parameters like age, gender, heart rate, and exercise time using Python and machine learning.

Potential Applications and Impacts:

- Personalized fitness tracking helps users to maintain an active lifestyle and reach their goals for fitness by tracking calories burned based on their own health data.
- Weight management: By providing light on the connection between exercise and calorie expenditure, this tool helps users control their weight.
- Health monitoring: Medical practitioners could use it to keep track on patients' physical activities or to follow their recovery.
- Corporate Wellness: implemented as wellness programs at work to support workers' mental and physical health.
- Enhanced Health knowledge: Enables healthier, more educated lifestyle choices and raises information of personal fitness measures.

1.3 Objective:

The primary objective of this project is to build a personal fitness tracker that allows users predict how many calories they will burn while exercising based on their own health parameters, such as body temperature, age, gender, BMI, heart rate, and duration of activity.

The software uses machine learning (more particularly, a Random Forest Regressor) to give users data-driven, accurate predictions and insights into how well they are doing in terms of fitness. The app will additionally offer users with context and motivation by comparing their data to that of other users, allowing them to see how their fitness levels compare to others.

The ultimate objective is to develop a simple to use interactive web tool that enables people in tracking their physical activity, creating fitness goals and making wise choices to improve their general health.

1.4 Scope and Limitations of the Project

1.4.1 Scope of the Project

The scope of the project is to offer a Personal Fitness Tracker program that uses a variety of inputs, including age, gender, body temperature, heart rate, BMI, and exercise duration, to predict how many calories a user would burn while exercising. Users of the app may establish exercise objectives, monitor their progress, and get data-driven individualized feedback. Along with visualizations like feature significance, interactive scatter plots, and correlation heatmaps, it uses a Random Forest Regressor model to forecast calories burnt. The software also provides a progress tracking indicator that indicates how near users are to reaching their fitness objectives in terms of exercise duration and calories burned. Below are the key functionalities:

a. User Input for customization
b. Prediction Model
c. Goal setting and Tracking
d. Data visualization
e. User specific fitness insights
f. Responsive user interface

1.4.2 Limitations of the Project

The Project's simplicity and dedication to providing an approachable, user-friendly experience are its good limitations. Because it uses a simple Random Forest Regressor without complex hyperparameter tuning or complicated algorithms, the software is still lightweight and user-friendly, making it perfect for anyone who are new to machine learning or fitness tracking. Because of its simplicity, people without expensive hardware can use the application because it is quick and uses little processing power. Furthermore, users can tailor their data to meet their unique requirements using the manual input approach, which always produces precise predictions based on the features that are accessible.

By allowing users to monitor their fitness progress in real-time, the goal-setting tool further increases the app's utility. Users may easily and interestingly understand the connections between their parameters and calories burned with the use of visualizations such as correlation heatmaps and scatter plots. Although the app does not yet have complex features like customized suggestions or real-time data integration, these drawbacks present chances for future improvement.

CHAPTER 2

Literature Survey

2.1 What is python?

Python contains dynamic semantics as well as is an a high-level, object-oriented, interpreted programming language. It is particularly attractive for quick application creation and for usage as a language of scripting or glue for connecting existing elements due to its high-level built-in data structures, dynamic typing, and dynamic binding. Python's easy-to-learn syntax encourages readability, which decreases software maintenance costs. Python improves code reuse and software architecture by allowing modules and packages. For all major platforms, the Python interpreter and the large standard library are openly accessible and available in binary or source format.[1]

Python Libraries:

Machine learning is growing as a key component in many areas, helping businesses to automate processes, analyse data, and make predictions. Python is known for its ease of use and flexibility due to the number of libraries it provides for machine learning positions. Developers and data scientists may fast and effectively create complex algorithms with the help of these libraries. Users can tackle machine learning projects more effectively and provide better results by utilizing Python's tools.[2]

BEST
PYTHON
LIBRARIES
FOR MACHINE LEARNING
01 NUMPY
02 SCIPY
03 SCIKITLEARN
04 THEANO
05 TENSORFLOW
06 KERAS
07 PYTORCH
08 PANDAS
09 MATPLOTLIB

Figure-2.1: Python Libraries

1. **NumPy**: NumPy is a well-known Python library for processing huge multi-dimensional arrays and matrices. For basic scientific calculations in machine learning, it is very useful. It is particularly useful for capabilities related to random numbers, linear algebra, and the Fourier transform. NumPy is used internally by high-end libraries such as TensorFlow to deal with tensors. [2]
2. **Pandas:** Pandas is a well-known Python data analysis toolbox. It has nothing to do with machine learning specifically. As is well known, the dataset needs to be ready before training starts. Pandas is useful in this situation because it was created especially for data preprocessing and extraction. It gives a large range of data analysis capabilities as well as high-level data structures. It has several built-in techniques for filtering, merging, and organizing data.[2]
3. **Matplotlib**: One of the most widely used Python libraries for data visualization is called Matplotlib. It has nothing to do with machine learning, just like pandas. When a programmer wishes to see any trends in the data, it is particularly useful. This 2D plotting library is used to generate 2D plots and graphs.[2]
4. **Scikit-Learn**: Scikit-Learn is most widely used ML libraries for traditional ML algorithms is Scikit-learn. Many supervised and unsupervised learning algorithms are provided by Scikit-learn. Scikit-learn is an excellent tool for those who are new to machine learning because it can be used for both data mining and data analysis.[2]

Random Forest Regressor: A random forest is a meta estimator that employs averaging to increase predictive accuracy and manage over-fitting by fitting multiple decision tree regressors on different dataset subsamples. [7]

2.2 Existing Models:

How To Use Python to Analyze Fitness Tracker Market: Step by Step EDA

Author: Devashree

About:

Essentially, a wearable device is one that the user may wear and that uses smart sensors for collecting important data about the user's activities and health. One important group of wearable technology is fitness trackers. To put it simply, a fitness tracker can be worn by the user to record and track information about their level of fitness through a variety of activities, such as daily steps or the total distance they walk each day, the number of hours they sleep and the stages of their sleep, the type of activity they perform and the number of calories they burn, cardiac monitoring, etc. These devices with internet access could send recorded data to a cloud.[3]

Track Your Fitness with Python: Analysing Google Fit Data

Author: Kaustubh Gupta

Visualizing and communicating information findings is an essential talent in the big data era. We may effectively explain data-driven results and understand complex data sets with the help of data visualization, an effective tool. Information and data are represented graphically in data visualization. It makes use of visual components such as maps, graphs, and charts to give users an easy method to observe and comprehend patterns, trends, and outliers in data.[4]

Live gym tracker using artificial intelligence

Author: Ayman Hayat Babu

Our physical and mental well-being must do to our posture. Many techniques have been used for determining different human positions. For instance, in the medical field, posture analysis can be used to determine a patient's resting position. OpenCV and the mediapipe Python package are used in image processing to estimate human posture. An image-processing-based method facilitates studying of sitting and standing postures. Fitness activities have amazing health advantages, but if done badly, they can be harmful and unsuccessful. [5]

AI Powered gym tracker using AI Pose estimation

Author: Guttikonda Gowrish

Through AI pose analysis, there is a great chance that an AI gym tracker can help individuals improve their training and prevent errors. In order accurately determine the pose, this study introduces a novel method that expertly combines Mediapipe and OpenCV features. While 33 Points for Human Pose Estimation uses pre-trained models to measure human conditions, the popular computer vision library OpenCV offers a wide variety of techniques for deciding images and videos. This strategy combines the benefits of both approaches to deliver an AI positioning algorithm. [6]

2.3 Highlight the gaps or limitations in existing solutions and how this project will address them.

Existing fitness tracking devices frequently have several significant limitations that reduce their usefulness. Many apps don't completely account for individual characteristics like age, body composition, heart rate, and exercise habits because they rely on generalized data or believe a one-size-fits-all approach. Poor projections and advise that doesn't suit the user's particular needs may result from this. Additionally, a lot of well-known fitness applications need external devices to offer real-time data, such as fitness trackers (like the Fitbit, Apple Watch, etc.). Users who don't have access to these devices or who would rather not use them may find this dependency to be a barrier. Furthermore, although some applications offer simple tracking of steps taken or calories burnt, they frequently lack complex progress tracking capabilities or thorough, informative feedback. Additionally, they might not have long-term goal tracking, which would restrict users' capacity to track their development over time.

These drawbacks are addressed in multiple ways by our efforts. First, it offers a highly customized fitness tracker that allows users to directly enter personal information like age, gender, BMI, heart rate, and amount of time spent exercising. More precise predictions and insights dedicated to each person are made possible by these modifications. Our solution is available to a wider range of customers regardless of their fitness equipment because, unlike many other apps, it doesn't require external tools.

Furthermore, our app provides a thorough method for defining and monitoring goals. The system allows users to create customized objectives for the number of calories burned and the amount of time spent exercising, and it tracks their progress in real time. Users can better understand their exercise routines and how various parameters (such heart rate or BMI) affect calorie burn by using interactive visualizations like scatter plots, correlation heatmaps, and progress gauges. Compared to simpler programs that offer static or basics tracking, this is a major advance.

CHAPTER 3

Proposed Methodology

3.1 System Design

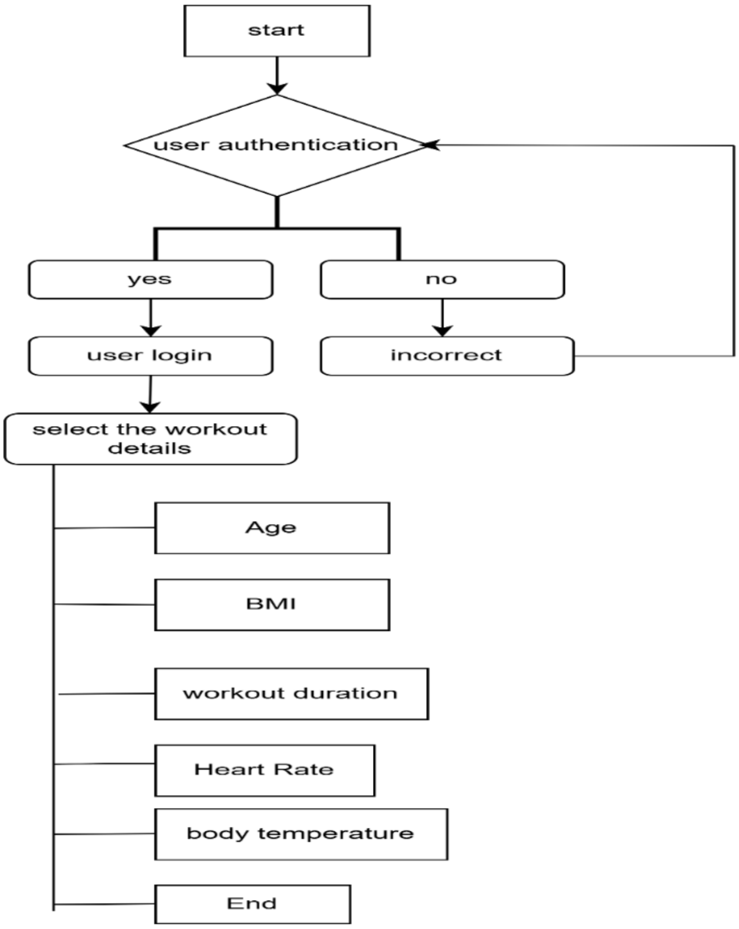


Figure-3.1: System Diagram for Implementation of personal fitness Tracker

The picture displays a flowchart in the Proposed Methodology section that shows a System Design. The flowchart can be explained as follows:

- Start: The "Start" point is where the process starts.
- User Authentication: The system confirms the validity of the user's authentication.
- The user is logged in if the answer is yes.

- If not, the system goes back to user authentication and produces an "Incorrect" notice.
- Choose the Workout Details: Following a successful login, the user has the option to choose work out details like: Age, Body Mass Index or BMI, Duration of Workout, Heart Rate, Temperature of the Body.
- End: After collecting all required workout information, the procedure comes to an end.

This flowchart, which is possibly for a fitness or health-related application, shows an appropriate process for user identification and workout detail selection.

3.2 Requirement Specification

3.2.1 **Hardware Requirements:** The following hardware specifications are necessary to properly install and operate the personal fitness tracker solution.

- CPU: For smooth operation, a multi-core CPU (such as an Intel i5/i7 or similar) is required.
- RAM: For fast calculation handling and visualization presentation, a minimum of 8 GB of RAM is required.
- Storage: The application itself, model storage, and dataset files require a minimum of 10 GB of free storage.
- Graphics: A simple GPU is suggested but optional for displaying complex graphs and visualizations more quickly, particularly when using Matplotlib or Plotly. • Any Internet-connected desktop, laptop, tablet, or smartphone that can execute web-based applications (using Streamlit for interactive UIs).
- Web Browser: To access the app online, use any popular web browser (such as Microsoft Edge, Mozilla Firefox, or Google Chrome).
- You must have an accurate internet connection.

3.2.2 Software Requirements:

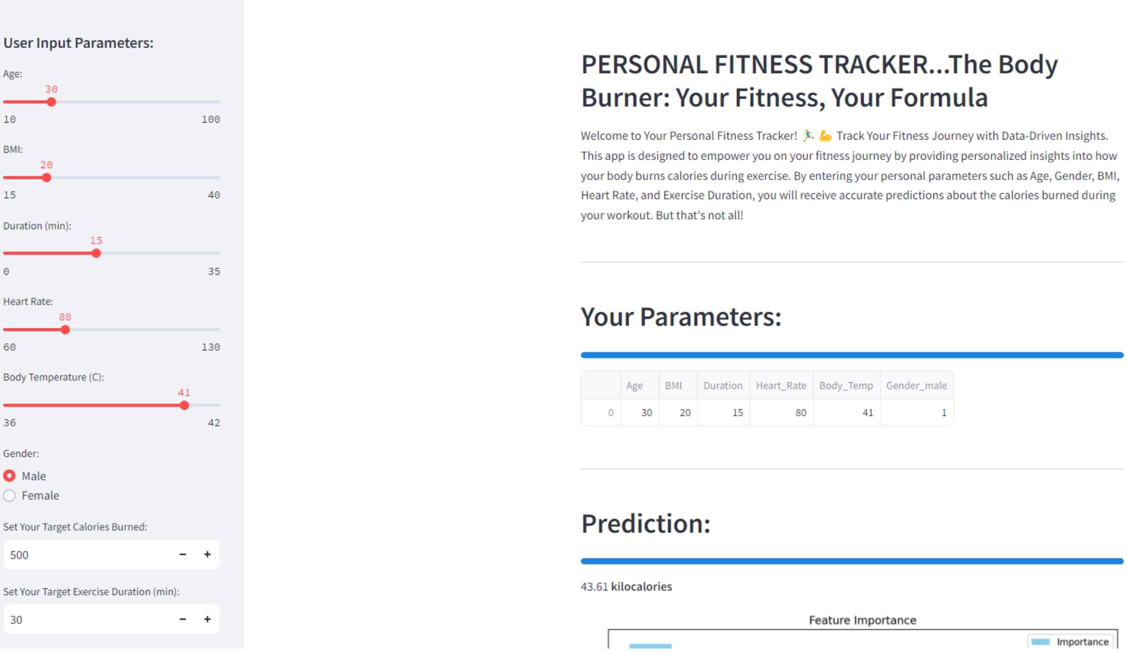
The following tools and software are needed to create, operate, and maintain the Personal Fitness Tracker:

- Programming Language: Python 3.x: The primary programming language for data processing, model training, application development, and visualization.
- Libraries and Frameworks: Streamlit, Numpy, Pandas, Scikit-learn, Matplotlib & Seaborn, Plotly, Time.
- Development Environment: IDE/Code editor like VScode, PyCharm, or jupyter Notebooks and git version like github.
- Local Storage: Used to hold datasets such as calories.csv and exercise.csv.
- Cloud Deployment: If the program must be deployed on the cloud, it can be hosted on Heroku, AWS, Google Cloud, or Streamlit Cloud.
- For better managing the python packages and dependencies install pip or Conda.

CHAPTER 4

Implementation and Result

4.1 Snap Shots of Result:



User Input Parameters:

Age: 30 (10 – 100)

BMI: 20 (15 – 40)

Duration (min): 15 (0 – 35)

Heart Rate: 80 (60 – 130)

Body Temperature (C): 41 (36 – 42)

Gender:

● Male

○ Female

Set Your Target Calories Burned: 500

Set Your Target Exercise Duration (min): 30

PERSONAL FITNESS TRACKER...The Body Burner: Your Fitness, Your Formula

Welcome to Your Personal Fitness Tracker! 🏃 💪 Track Your Fitness Journey with Data-Driven Insights. This app is designed to empower you on your fitness journey by providing personalized insights into how your body burns calories during exercise. By entering your personal parameters such as Age, Gender, BMI, Heart Rate, and Exercise Duration, you will receive accurate predictions about the calories burned during your workout. But that's not all!

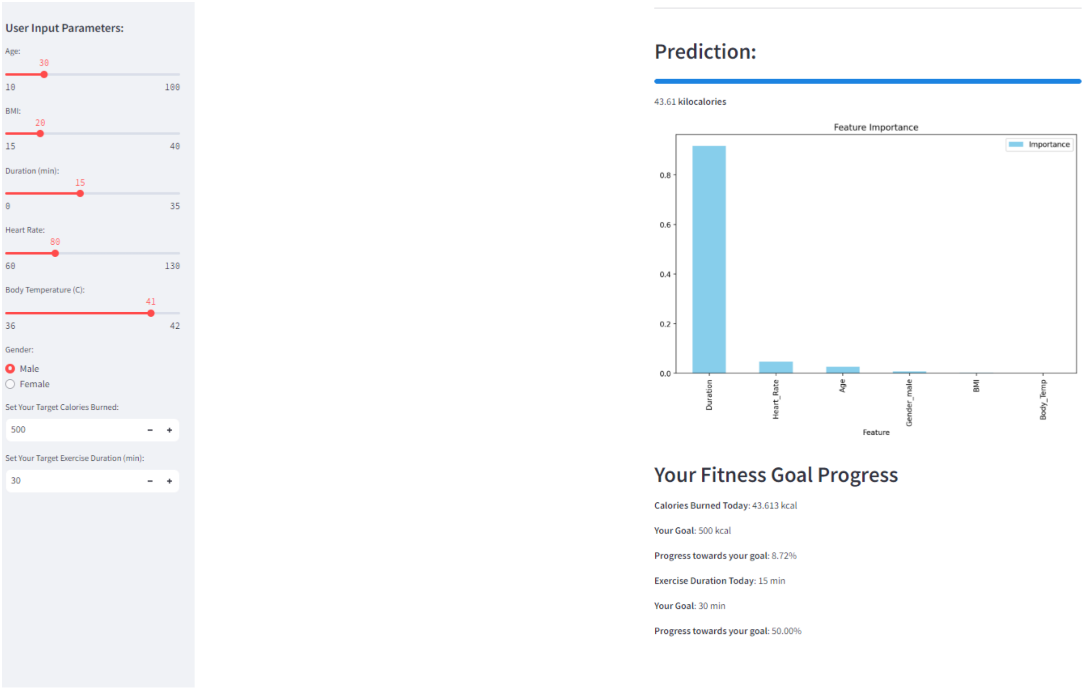
Your Parameters:

	Age	BMI	Duration	Heart_Rate	Body_Temp	Gender_male
0	30	20	15	80	41	1

Prediction:

43.61 **kilocalories**

Figure-4.1: It is the first output generated when the code is run, which allows users to enter their input in user input parameters and set the target calories burned which can be used to predict the calories burned.


User Input Parameters:

Age: 30

BMI: 20

Duration (min): 15

Heart Rate: 80

Body Temperature (C): 41

Gender:

● Male
○ Female

Set Your Target Calories Burned: 500

Set Your Target Exercise Duration (min): 30

Prediction:

43.61 kilocalories

Your Fitness Goal Progress

Calories Burned Today: 43.613 kcal

Your Goal: 500 kcal

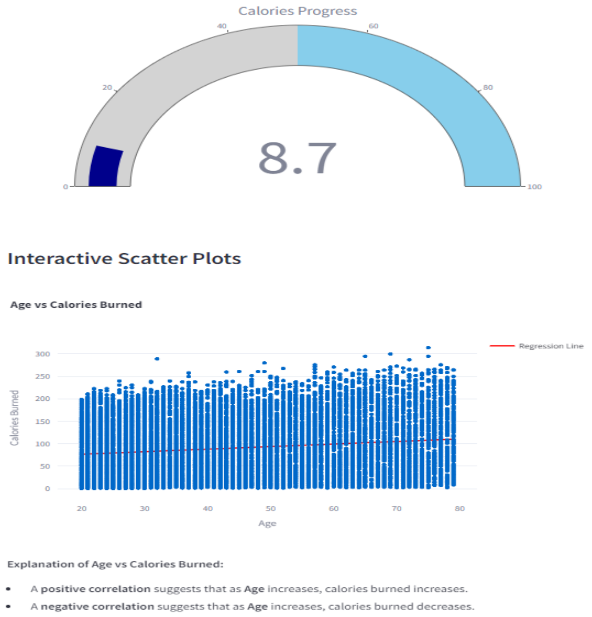
Progress towards your goal: 8.72%

Exercise Duration Today: 15 min

Your Goal: 30 min

Progress towards your goal: 50.00%

Figure-4.2: It is the output for predicting the calories that are burned in kilograms and a graph that shows the importance of selected features along with the user fitness goal progress.

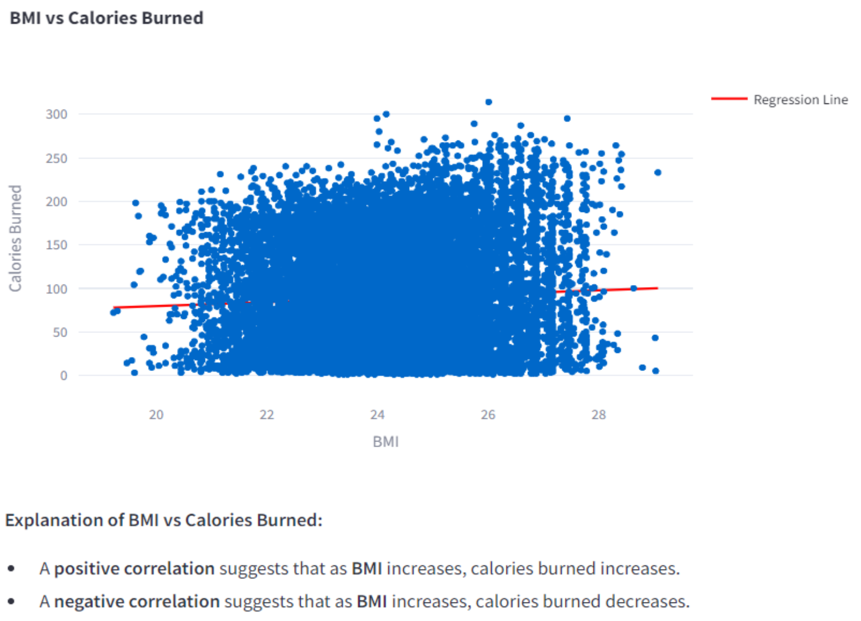


Interactive Scatter Plots

Explanation of Age vs Calories Burned:

- A **positive correlation** suggests that as **Age** increases, calories burned increases.
- A **negative correlation** suggests that as **Age** increases, calories burned decreases.

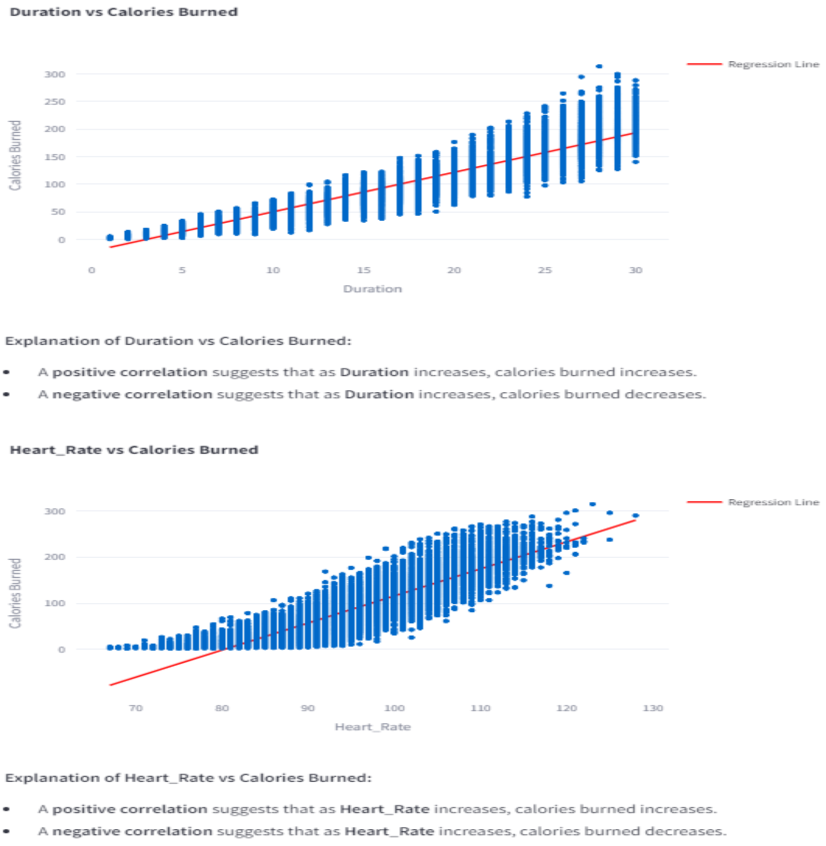
Figure-4.3: Generated Scatter plot based on age and calorie burned and given context to make better understand for the user.



Explanation of BMI vs Calories Burned:

- A **positive correlation** suggests that as **BMI** increases, calories burned increases.
- A **negative correlation** suggests that as **BMI** increases, calories burned decreases.

Figure-4.4: generated graph based on the user input BMI and calories burned with the explanation for better user understanding.



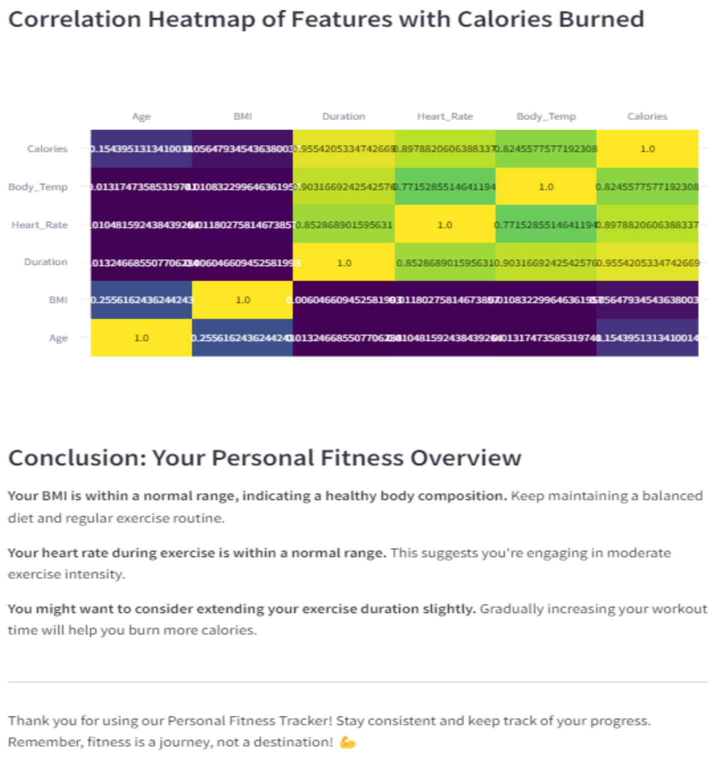
Explanation of Duration vs Calories Burned:

- A **positive correlation** suggests that as **Duration** increases, calories burned increases.
- A **negative correlation** suggests that as **Duration** increases, calories burned decreases.

Explanation of Heart_Rate vs Calories Burned:

- A **positive correlation** suggests that as **Heart_Rate** increases, calories burned increases.
- A **negative correlation** suggests that as **Heart_Rate** increases, calories burned decreases.

Figure- 4.5: generated graph based on the user inputs Duration, Heart rate with burned calories and its explanation for better user understanding.



Correlation Heatmap of Features with Calories Burned

Conclusion: Your Personal Fitness Overview

Your BMI is within a normal range, indicating a healthy body composition. Keep maintaining a balanced diet and regular exercise routine.

Your heart rate during exercise is within a normal range. This suggests you're engaging in moderate exercise intensity.

You might want to consider extending your exercise duration slightly. Gradually increasing your workout time will help you burn more calories.

Thank you for using our Personal Fitness Tracker! Stay consistent and keep track of your progress. Remember, fitness is a journey, not a destination! 💪

Figure-4.6: Generated correlation Heatmaps of features with burned calories and provided conclusion along with the overview and suggestions to improve their health fitness based on the given user inputs.

4.2GitHub Link for Code:

https://github.com/AmulyaDudam/Implementation-of-personal-Fitness-Tracker-using-python

4.3Video of the Project

https://github.com/AmulyaDudam/Implementation-of-personal-Fitness-Tracker-using-python/blob/main/video%20of%20the%20project.mp4

CHAPTER 5

Discussion and Conclusion

5.1 Future Work:

Ideas for Enhancing the Model or Solving Unresolved Problems in Future Research

- Advanced Methods for Machine Learning: Explore more complex models, such as XGBoost, LightGBM, or Neural Networks, to enhance prediction accuracy and capture complex structures.
- Hyperparameter Optimization: To fine-tune model parameters and prevent overfitting, extend the existing grid search using methods like Bayesian Optimization or Random Search.
- Personalized Models: To improve prediction accuracy and relevance, create customized models for various user groups according to age, gender, and fitness level.
- Connect the app to wearable technology (such as the Fitbit or Apple Watch) to gather data in real time (heart rate, steps, etc.) for more precise and automated fitness tracking.
- Mobile Integration: To automatically import user data and streamline the tracking process, sync the app with health platforms such as Apple Health or Google Fit.
- Nutritional Tracking: Include a nutritional tracker in the app so users may keep tabs on their caloric intake and get a more complete picture of their fitness development.
- Exercise Type Specificity: For more precise calorie burn estimates customized to the activity, let consumers choose their favorite type of exercise (cycling, jogging, etc.).
- Personalized Fitness Plans: Create workout routines according to the user's objectives (e.g., muscle growth, weight loss) and monitor their development over time.

5.2 Conclusion:

By offering data-driven insights into calorie burn during activity, the Personal Fitness Tracker app greatly enhances how people monitor and manage their fitness journeys. The software helps users make educated decisions about their health habits by using machine learning algorithms to provide precise predictions based on user inputs like age, gender, BMI, heart rate, and activity duration.

A smooth, customizable user experience is facilitated by its integration with goal tracking tools, individualized exercise advice, and real-time data from wearables. Furthermore, having the capacity to establish and track fitness objectives (such as the number of calories burnt or the length of an exercise session) not only inspires users but also offers a thorough picture of their development, promoting consistency and long-term fitness success.

The project offers a useful option for people looking for individualized, precise, and actionable insights by addressing the gap between fitness tracking and machine learning. Its potential for growth—including the addition of nutritional tracking, enhanced predictive models, and the integration of more varied user data—further establishes it as a useful resource for anybody dedicated to enhancing their well-being.

REFERENCES

[1] “What is python?” http://python.org/doc/essays/blurb/

[2] https://www.geeksforgeeks.org/best-python-libraries-for-machine-learning/ “python libraries”

[3] How to Use Python to Analyze Fitness Tracker Market: Step by Step EDA. Author-Devashree: https://www.analyticsvidhya.com/blog/2021/11/how-to-use-python-to-analyse-fitness-tracker-market-step-by-step-eda/

[4] Live gym tracker using artificial intelligence Author-Ayman Hayath Babu

https://pubs.aip.org/aip/acp/article-abstract/3075/1/020255/3305187/Live-gym-tracker-using-artificial-intelligence

[5] Track Your Fitness with Python: Analysing Google Fit Data. Author: Kaustubh Gupta https://www.analyticsvidhya.com/blog/2021/04/track-your-fitness-with-python-analyzing-google-fit-data/

[6] AI Powered gym tracker using AI Pose estimation. Author-Guttikonda Gowrish https://ieeexplore.ieee.org/abstract/document/10537312

[7] Random Forest Regressor

https://scikitlearn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

Thank You