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

Lakshman Raaj G,

lakshmanraajg@gmail.com

Under the Guidance of

Prof. Saomya Chaudhury

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to all individuals who supported and guided us throughout this project. Special thanks to our supervisor, Saomya Chaudury, whose valuable insights, encouragement, and mentorship helped shape this project into a meaningful endeavor. We also extend our appreciation to our peers and family members for their continuous support and motivation.

This project has been a great learning experience, allowing us to explore the integration of artificial intelligence with fitness tracking. The encouragement and support we received have helped us navigate challenges and refine our approach. We acknowledge the efforts of our teammates and contributors who have played a crucial role in different aspects of development, including coding, testing, and documentation. The collaborative spirit made this project successful and enjoyable.

ABSTRACT

This project presents the "Personal Fitness Tracker," an AI-powered application designed to track and predict calorie burn based on user input parameters such as age, BMI, heart rate, exercise duration, and body temperature. The system utilizes a Random Forest Regressor model trained on fitness data to provide accurate predictions. This tool helps users set fitness goals and track progress dynamically. The application is developed using Python and Streamlit for interactive user experience.

The motivation behind this project stems from the need to provide users with data-driven fitness insights without the requirement of expensive wearable devices. With a simple and user-friendly interface, users can input their details and receive personalized calorie burn estimates. This system not only promotes awareness about physical activity but also encourages users to make informed decisions regarding their exercise routines. Additionally, the application includes visualization tools to enhance engagement and understanding, making fitness tracking a more interactive experience.

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER 1

Introduction

1.1Problem Statement:

Maintaining a healthy lifestyle is challenging without proper tracking of exercise and calorie burn. Most people struggle to determine the effectiveness of their workouts due to a lack of real-time feedback. The need for a personalized fitness tracker that provides AI-driven insights is crucial.

Traditional fitness tracking solutions often rely on wearable devices, which can be expensive and inaccessible to many individuals. Moreover, generic fitness advice does not always align with individual physiological differences. Our project addresses these gaps by providing a machine-learning-based fitness tracker that personalizes calorie burn predictions. By leveraging AI, users receive tailored insights that help optimize their workouts, ultimately leading to better health outcomes.

1.2Motivation:

With the rise of AI and wearable health technology, fitness tracking has become a key focus area. This project aims to develop an accessible and AI-driven solution for users who want to monitor and improve their physical health without needing expensive devices.

Regular physical activity is essential for maintaining overall well-being. However, many individuals find it difficult to track their progress effectively. The lack of immediate feedback often results in demotivation and inconsistency in workouts. This project seeks to bridge this gap by using machine learning to provide users with insights into their exercise routines. By making fitness tracking accessible to everyone, we hope to encourage healthier lifestyles and increased participation in physical activities.

1.3Objective:

- Develop a machine learning model to predict calories burned.
- Create an interactive web application for real-time fitness tracking.
- Integrate visual analytics for enhanced user experience.
- Provide goal-setting and tracking features.

The overarching goal of this project is to empower individuals to make informed decisions about their fitness routines. By leveraging AI, we provide users with actionable insights that can improve their exercise habits. Additionally, by incorporating goal-setting features, users can stay motivated and monitor their progress over time. This project aims to make fitness tracking both intuitive and effective.

1.4Scope of the Project:

This project focuses on real-time tracking and prediction of calories burned based on user input. It leverages AI models for accuracy and visualization techniques to enhance user engagement. Future enhancements may include mobile app integration and extended fitness recommendations.

The application is designed to be scalable and adaptable. Future iterations may include real-time heart rate monitoring through integration with smart devices. Additionally, expanding the dataset with more diverse user inputs can further enhance the accuracy of predictions. The scope also includes integrating personalized workout recommendations based on user fitness levels and goals.

CHAPTER 2

Literature Survey

2.1 Previous work in this domain:

Several studies have explored AI in fitness tracking. Traditional fitness applications rely on heuristic-based estimations, whereas recent advancements utilize machine learning for improved accuracy. AI-driven models such as decision trees, support vector machines, and deep learning networks have been applied in various health monitoring applications. Research has demonstrated that AI models outperform traditional methods by adapting to individual user data, leading to more precise and personalized insights.

2.2 Existing models:

Existing fitness tracking models typically use either direct measurement devices such as smartwatches or indirect estimation techniques using mathematical formulas. Some popular models include:

- **Harris-Benedict Equation**: Estimates basal metabolic rate based on weight, height, age, and gender.
- **Mifflin-St Jeor Equation**: An improved version of the Harris-Benedict formula, often used in calorie estimation.
- **Machine Learning Models**: Random Forest, Neural Networks, and Linear Regression are commonly used for fitness prediction.
- **Wearable Technologies**: Devices like Fitbit and Apple Watch provide real-time monitoring but can be expensive.

This project utilizes a Random Forest Regressor due to its ability to handle complex datasets and provide reliable predictions.

2.3 Limitations in existing solutions:

While existing models provide valuable insights, they have several limitations:

- Many traditional methods lack personalization and rely on static formulas that do not consider unique physiological variations.
- Wearable devices, though accurate, are costly and inaccessible to many users.
- Existing AI models often require extensive datasets, making them impractical for personal use.

CHAPTER 3

Proposed Methodology

3.1 System Design

The system follows the following architecture:

- **User Input:** Age, BMI, Heart Rate, Exercise Duration, Body Temperature.
- **Preprocessing:** Data cleaning and feature engineering.
- **Model Training:** Random Forest Regressor trained on fitness datasets.
- **Prediction & Visualization:** Predicted calorie burn, interactive charts, and goal tracking.

3.2 Requirement Specification

Hardware Requirements

- Standard PC with Internet Access

Software Requirements

- Python
- Streamlit
- Scikit-learn
- Pandas, Matplotlib, Plotly

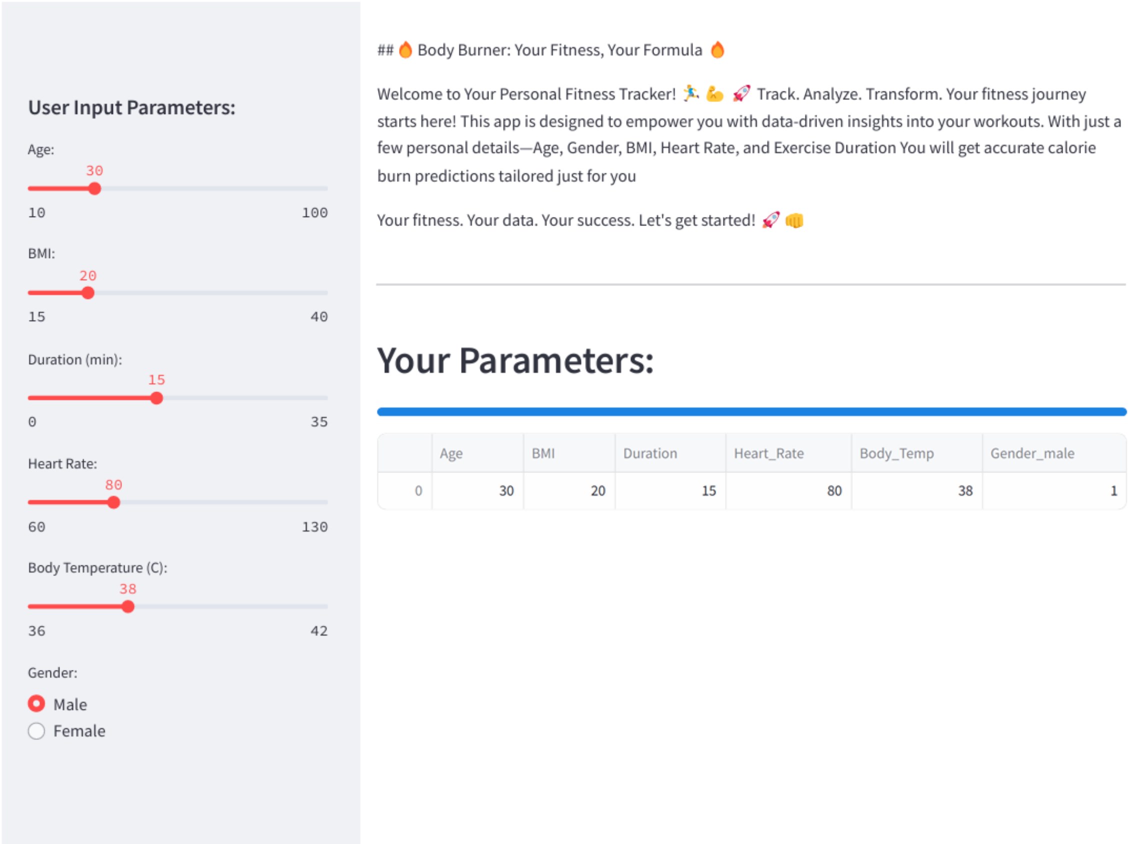
CHAPTER 4

Implementation and Result

4.1 Snap Shots of Result:

```python
import streamlit as st
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sn
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
import time
import plotly.express as px
import plotly.graph_objects as go
import plotly.figure_factory as ff

import warnings
warnings.filterwarnings('ignore')

# Title and description
st.write("## 🔥 Body Burner: Your Fitness, Your Formula 🔥")
st.write("""Welcome to Your Personal Fitness Tracker! 🏃 💪
🚀 Track. Analyze. Transform. Your fitness journey starts here!
This app is designed to empower you with data-driven insights into your workouts.
With just a few personal details—Age, Gender, BMI, Heart Rate, and Exercise Duration
You will get accurate calorie burn predictions tailored just for you""")
st.write("Your fitness. Your data. Your success. Let's get started! 🚀 👊")
st.sidebar.header("User Input Parameters: ")

# Function to collect user input
def user_input_features():
    age = st.sidebar.slider("Age: ", 10, 100, 30)
    bmi = st.sidebar.slider("BMI: ", 15, 40, 20)
    duration = st.sidebar.slider("Duration (min): ", 0, 35, 15)
    heart_rate = st.sidebar.slider("Heart Rate: ", 60, 130, 80)
    body_temp = st.sidebar.slider("Body Temperature (C): ", 36, 42, 38)
    gender_button = st.sidebar.radio("Gender: ", ("Male", "Female"))

    gender = 1 if gender_button == "Male" else 0

    data_model = {
        "Age": age,
        "BMI": bmi,
        "Duration": duration,
        "Heart_Rate": heart_rate,
        "Body_Temp": body_temp,
        "Gender_male": gender  # Gender is encoded as 1 for male, 0 for female
    }

    features = pd.DataFrame(data_model, index=[0])
    return features

df = user_input_features()

st.write("---")
st.header("Your Parameters: ")
latest_iteration = st.empty()
bar = st.progress(0)
for i in range(100):
    bar.progress(i + 1)
    time.sleep(0.01)
st.write(df)

# Load and preprocess data
calories = pd.read_csv("calories.csv")
exercise = pd.read_csv("exercise.csv")

# Calculate BMI before splitting
exercise["BMI"] = exercise["Weight"] / ((exercise["Height"] / 100) ** 2)
exercise["BMI"] = round(exercise["BMI"], 2)

exercise_df = exercise.merge(calories, on="User_ID")
exercise_df.drop(columns="User_ID", inplace=True)

exercise_train_data, exercise_test_data = train_test_split(exercise_df, test_size=0.2, random_state=1)

# Prepare the training and testing sets
exercise_train_data = exercise_train_data[["Gender", "Age", "BMI", "Duration", "Heart_Rate",
"Body_Temp", "Calories"]]
exercise_test_data = exercise_test_data[["Gender", "Age", "BMI", "Duration", "Heart_Rate", "Body_Temp",
"Calories"]]
exercise_train_data = pd.get_dummies(exercise_train_data, drop_first=True)
exercise_test_data = pd.get_dummies(exercise_test_data, drop_first=True)

# Separate features and labels
X_train = exercise_train_data.drop("Calories", axis=1)
y_train = exercise_train_data["Calories"]

X_test = exercise_test_data.drop("Calories", axis=1)
y_test = exercise_test_data["Calories"]

# ---- RANDOM FOREST REGRESSOR WITHOUT GRIDSEARCHCV ----
random_reg = RandomForestRegressor(n_estimators=1000)  # Default Random Forest Regressor without
additional hyperparameters
random_reg.fit(X_train, y_train)

# Align prediction data columns with training data
df = df.reindex(columns=X_train.columns, fill_value=0)

# Make prediction
prediction = random_reg.predict(df)

st.write("---")
st.header("Prediction: ")
latest_iteration = st.empty()
bar = st.progress(0)
for i in range(100):
    bar.progress(i + 1)
    time.sleep(0.01)

st.write(f"{round(prediction[0], 2)} **kilocalories**")

# ---- FEATURE IMPORTANCE VISUALIZATION ----
feature_importances = random_reg.feature_importances_
feature_df = pd.DataFrame({
    'Feature': X_train.columns,
    'Importance': feature_importances
}).sort_values(by='Importance', ascending=False)

fig, ax = plt.subplots(figsize=(10, 6))
feature_df.plot(kind='bar', x='Feature', y='Importance', ax=ax, color='skyblue')
plt.title('Feature Importance')
st.pyplot(fig)

# ---- ADDING GOAL SETTING AND TRACKING ----
def set_goals():
    goal_calories = st.sidebar.number_input("Set Your Target Calories Burned:", min_value=0,
max_value=1000, value=500)
    goal_duration = st.sidebar.number_input("Set Your Target Exercise Duration (min):", min_value=0,
max_value=120, value=30)
    return goal_calories, goal_duration

goal_calories, goal_duration = set_goals()

# Calculate progress
calories_burned = prediction[0]
calories_progress = (calories_burned / goal_calories) * 100
duration_progress = (df["Duration"].values[0] / goal_duration) * 100

# Display progress
st.header("Your Fitness Goal Progress")
st.write(f"**Calories Burned Today**: {calories_burned} kcal")
st.write(f"**Your Goal**: {goal_calories} kcal")
st.write(f"**Progress towards your goal**: {calories_progress:.2f}%")

st.write(f"**Exercise Duration Today**: {df['Duration'].values[0]} min")
st.write(f"**Your Goal**: {goal_duration} min")
st.write(f"**Progress towards your goal**: {duration_progress:.2f}%")
```

```python
# ---- INTERACTIVE SCATTER PLOTS WITH REGRESSION LINE ----
st.write("### Interactive Scatter Plots")
features_to_plot = ["Age", "BMI", "Duration", "Heart_Rate", "Body_Temp"]

for feature in features_to_plot:
    fig = px.scatter(exercise_df, x=feature, y="Calories", title=f"{feature} vs Calories Burned",
labels={feature: feature, 'Calories': 'Calories Burned'})
    fig.update_layout(showlegend=True)

    # Add a regression line
    x_vals = np.array(exercise_df[feature])
    y_vals = np.array(exercise_df['Calories'])
    coefficients = np.polyfit(x_vals, y_vals, 1)
    polynomial = np.poly1d(coefficients)
    line_x = np.linspace(x_vals.min(), x_vals.max(), 100)
    line_y = polynomial(line_x)

    fig.add_trace(go.Scatter(x=line_x, y=line_y, mode='lines', name='Regression Line',
line=dict(color='red')))
    st.plotly_chart(fig)

    st.write(f"""
        **Explanation of {feature} vs Calories Burned:**
        - A **positive correlation** suggests that as **{feature}** increases, calories burned
increases.
        - A **negative correlation** suggests that as **{feature}** increases, calories burned
decreases.
    """)

# ---- INTERACTIVE CORRELATION HEATMAP ----
st.write("### Correlation Heatmap of Features with Calories Burned")
corr = exercise_df[["Age", "BMI", "Duration", "Heart_Rate", "Body_Temp", "Calories"]].corr()

fig = ff.create_annotated_heatmap(
    z=corr.values,
    x=list(corr.columns),
    y=list(corr.index),
    colorscale='Viridis'
)
st.plotly_chart(fig)

# ---- GENERAL INFORMATION AND CONCLUSION BASED ON USER INPUT ----
st.write("### Conclusion: Your Personal Fitness Overview")

if df["BMI"].values[0] > 30:
    st.write("**You have a high BMI, which suggests that you might have some extra body fat.** Consider
focusing on cardiovascular exercises to burn more calories.")
else:
    st.write("**Your BMI is within a normal range, indicating a healthy body composition.** Keep
maintaining a balanced diet and regular exercise routine.")

if df["Heart_Rate"].values[0] > 100:
    st.write("**Your heart rate is relatively high during exercise.** This indicates that you might be
pushing yourself hard during physical activities.")
else:
    st.write("**Your heart rate during exercise is within a normal range.** This suggests you're
engaging in moderate exercise intensity.")

if df["Duration"].values[0] > 30:
    st.write("**You're exercising for more than 30 minutes, which is fantastic!** Consistent exercise
duration helps build endurance and aids in long-term calorie burning.")
else:
    st.write("**You might want to consider extending your exercise duration slightly.** Gradually
increasing your workout time will help you burn more calories.")

st.write("---")
st.write("Thank you for using our Personal Fitness Tracker! Stay consistent and keep track of your
progress. Remember, fitness is a journey, not a destination! 💪")
```



4.2 GitHub Link for Code:

https://github.com/lakshmanraajg/Personal-Fitness-Tracker

CHAPTER 5

Discussion and Conclusion

5.1 Future Work:

While the current implementation of the Personal Fitness Tracker provides a solid foundation, there is potential for further enhancements. Future iterations of the project could include integrating real-time sensor data from smartwatches or fitness trackers. This would improve the accuracy of calorie predictions by incorporating live heart rate and movement data. Additionally, incorporating a mobile version of the application would make fitness tracking more accessible and convenient for users.

Another improvement could be the implementation of personalized workout recommendations based on past exercise patterns and user preferences. By analyzing historical data, the system could suggest optimized workout plans tailored to individual fitness goals. Additionally, incorporating a social or community feature where users can compare progress and challenge each other could enhance engagement and motivation.

Furthermore, the integration of dietary tracking can complement the existing calorie burn predictions, providing users with a more holistic approach to fitness. Future enhancements may also include AI-driven coaching, where the system provides feedback on workout efficiency and adjustments based on real-time user performance. These additions will make the application more comprehensive and beneficial to users seeking to improve their fitness levels.

5.2 Conclusion:

This project successfully developed a personalized fitness tracker that provides real-time calorie burn predictions based on user input. By integrating AI and machine learning, the application enhances tracking accuracy, enabling users to make data-driven fitness decisions. The interactive visualizations further improve user engagement and motivation.

The project demonstrates the potential of AI in health and fitness applications. By providing a free and accessible solution, it encourages individuals to adopt a more data-driven approach to their fitness routines. The use of Random Forest Regression has proven effective in predicting calorie burn, and the integration of user-friendly visualizations enhances the overall experience.

Overall, the Personal Fitness Tracker showcases how AI can be leveraged to improve physical well-being. By refining and expanding its features, the application can continue to be a valuable tool for individuals seeking to enhance their fitness journey. The proposed future work will further elevate the effectiveness and user engagement, making the system an essential component in modern fitness tracking solutions.

REFERENCES

[1] [1].Ming-Hsuan Yang, David J. Kriegman, Narendra Ahuja, “Detecting Faces in Images: A Survey”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume. 24, No. 1, 2002.

[2] [2].Suleimenov IE, Vitulyova YS, Bakirov AS, Gabrielyan OA. Artificial Intelligence:what is it? Proc 2020 6th Int Conf Comput Technol Appl. 2020;22–5. https://doi.org/10.1145/3397125.3397141.

[3] [3].Davenport T, Kalakota R. The potential for artificial intelligence in Healthcare. Future Healthc J. 2019;6(2):94–8. https://doi.org/10.7861/futurehosp.6-2-94.

[4] [4].McCorduck P, Cfe C. Machines who think: a personal inquiry into the history and prospects of Artificial Intelligence. AK Peters; 2004.

[5] [5].Jordan MI, Mitchell TM. Machine learning: Trends, perspectives, and prospects. Science. 2015;349(6245):255–60. https://doi.org/10.1126/science.aaa8415.

[6] [6].VanLEHN K. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. Educational Psychol. 2011;46(4):197–221. https://doi.org/10.1080/00461520.2011.611369.

[7] [7].Topol EJ. High-performance medicine: the convergence of human and Artificial Intelligence. Nat Med. 2019;25(1):44–56. https://doi.org/10.1038/s41591-018-0300-7.

[8] [8].Esteva A, Kuprel B, Novoa RA, Ko J, Swetter SM, Blau HM, et al. Dermatologist-level classification of skin cancer with deep neural networks. Nature. 2017;542(7639):115–8. https://doi.org/10.1038/nature21056.

[9] [9].Myszczynska MA, Ojamies PN, Lacoste AM, Neil D, Saffari A, Mead R, et al. Applications of machine learning to diagnosis and treatment of neurodegenerative Diseases. Nat Reviews Neurol. 2020;16(8):440–56. https://doi.org/10.1038/s41582-020-0377-8.

[10] [10]. McKinney SM, Sieniek M, Godbole V, Godwin J, Antropova N, Ashrafian H, et al. International evaluation of an AI system for breast cancer screening. Nature. 2020;577(7788):89–94. https://doi.org/10.1038/s41586-019-1799-6.

[11] [11]. Kim H-E, Kim HH, Han B-K, Kim KH, Han K, Nam H, et al. Changes in cancer detection and false-positive recall in mammography using Artificial Intelligence: a retrospective, Multireader Study. Lancet Digit Health. 2020;2(3). https://doi.org/10.1016/s2589-7500(20)30003-0.

[12] [12]. Han SS, Park I, Eun Chang S, Lim W, Kim MS, Park GH, et al. Augmented Intelligence Dermatology: deep neural networks Empower Medical Professionals in diagnosing skin Cancer and Predicting Treatment Options for 134 skin Disorders. J Invest Dermatol. 2020;140(9):1753–61. https://doi.org/10.1016/j.jid.2020.01.019.