**Implementation of AI 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

**Irlanki Leela Sai Abhiram**

[**lirlanki@gitam.in**](mailto:lirlanki@gitam.in)

Under the Guidance of

**Saomya Chaudhury**

**ACKNOWLEDGEMENT**

I would like to take this opportunity to express our deep sense of gratitude to all individuals who helped us directly or indirectly during this thesis work.

I would like to thank my supervisor, Saomya Chaudhury for being a great mentor and the best adviser I could ever have. I extend my heartfelt thanks to AICTE and TechSaksham, a joint CSR initiative by Microsoft and SAP, for providing me with the opportunity to undertake this internship and gain practical experience in implementing machine learning models and building interactive applications.

I am deeply grateful for the valuable knowledge and insights imparted through the structured learning modules and hands-on sessions during the internship. The exposure to real-world problem-solving and the opportunity to apply theoretical concepts to practical applications have greatly enhanced my technical skills and analytical abilities.

I also wish to acknowledge the continuous support and encouragement from my peers and colleagues, whose discussions and feedback helped refine the project. Their constructive suggestions and shared knowledge were instrumental in addressing challenges and ensuring the successful deployment of the solution.

Finally, I am thankful to the organizers and facilitators of this program for fostering a collaborative and stimulating learning environment that enabled me to develop a deeper understanding of artificial intelligence and its transformative potential. This experience has been invaluable in strengthening my skills and preparing me for future challenges in the field of AI and technology.

#### **ABSTRACT**

The project titled **“Implementation of Personal Fitness Tracker using Python”** aims to address the growing need for accurate and personalized fitness tracking by developing a machine learning-based application that predicts calories burned based on user-input parameters. Traditional fitness tracking methods often rely on generalized formulas that fail to account for individual differences, leading to inaccurate calorie estimations. This project leverages machine learning techniques to provide a more accurate and dynamic prediction system.

The primary objective of this project is to design and implement a user-friendly application that predicts the number of calories burned during physical activity by considering key factors such as age, BMI, duration of activity, heart rate, body temperature, and gender. The application was developed using Python and Streamlit, with a **Gradient Boosting Regressor (GBR)** as the core prediction model. Data from exercise.csv and calories.csv datasets were preprocessed, merged, and used to train the model. The dataset was split into training and testing sets, with 80% of the data used for training and 20% for evaluation.

The model achieved high accuracy, with an R-squared score of approximately 98%, indicating its effectiveness in predicting calorie expenditure. The application interface, built with Streamlit, allows users to input their personal information and view real-time predictions of calories burned. Additionally, the application provides visual insights into the data and displays performance metrics to evaluate the model’s accuracy.

In conclusion, this project successfully demonstrates the potential of machine learning in enhancing the accuracy of fitness tracking applications. The model's high performance and interactive interface offer users a reliable and personalized tool to monitor their fitness journeys. Future enhancements may include expanding the model to predict other health parameters and integrating real-time data from wearable devices.

**TABLE OF CONTENT**

**Abstract .................................................................................................................. i**

**Chapter 1: Introduction ...................................................................................... 1**

* **1.1 Problem Statement ......................................................................................... 1**
* **1.2 Motivation ...................................................................................................... 1**
* **1.3 Objectives ...................................................................................................... 2**
* **1.4 Scope of the Project ....................................................................................... 3**

**Chapter 2: Literature Survey ............................................................................... 4**

* **2.1 Introduction ................................................................................................... 4**
* **2.2 Review of Existing Models ............................................................................ 4**
* **2.3 Gaps and Limitations in Existing Solutions ...................................................6**
* **2.4 Proposed Solution and Justification ............................................................... 6**

**Chapter 3: Proposed Methodology ...................................................................... 8**

* **3.1 System Design ............................................................................................... 8**
* **3.2 System Architecture ....................................................................................... 9**
* **3.3 Requirement Specification ............................................................................ 9**
  + **3.3.1 Hardware Requirements ................................................................ 9**
  + **3.3.2 Software Requirements .................................................................. 9**
* **3.4 Data Collection and Preprocessing .............................................................. 10**
* **3.5 Model Selection and Training ....................................................................... 10**
* **3.6 Application Development ............................................................................ 11**

**Chapter 4: Implementation and Results ............................................................. 12**

* **4.1 Implementation Details ................................................................................ 12**
* **4.2 Snapshots of Results .................................................................................... 13**
* **4.3 GitHub Link for Code ................................................................................... 16**

**Chapter 5: Discussion and Conclusion .............................................................. 17**

* **5.1 Future Work .................................................................................................. 17**
* **5.2 Conclusion .................................................................................................... 18**

**References .......................................................................................................... 19**

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Figure Caption** | **Page No.** |
|  | Figure 4.1 User Input Interface in Streamlit Application | **9** |
|  | Figure 4.2 Predicted Calories and Similar Results Display | **9** |
|  | Figure 4.3 R-squared Score Graph from Model Evaluation | **10** |
|  | Figure 4.4 Mean Absolute Error (MAE) Graph from Model Evaluation | **10** |

**CHAPTER 1**

**Introduction**

**1.1 Problem Statement**

Accurately tracking the number of calories burned during physical activity is essential for individuals seeking to maintain or improve their health and fitness levels. Most conventional fitness tracking methods use generalized equations that consider only a limited number of parameters such as age, gender, and activity type. These methods often overlook individual differences, leading to inconsistent and inaccurate predictions. Additionally, existing fitness applications lack personalized predictions based on real-time data, limiting their effectiveness for users with unique physiological characteristics.

To address these challenges, this project aims to develop a machine learning-based Personal Fitness Tracker that predicts calories burned by considering multiple factors, including age, height, weight, BMI, duration of activity, heart rate, body temperature, and gender. The model ensures personalized predictions by utilizing real-world data and enhancing accuracy through advanced regression techniques.

**1.2 Motivation**

With the increasing awareness of the importance of maintaining an active and healthy lifestyle, there is a growing demand for reliable fitness tracking systems that provide personalized insights. Existing applications often rely on static formulas that do not account for variations in individual physiological attributes, resulting in imprecise calorie predictions. This gap in accuracy and personalization motivates the need for a data-driven approach that dynamically adapts to individual user profiles.

Furthermore, advancements in machine learning provide an opportunity to improve prediction accuracy by training models on real-world datasets. A personalized fitness tracker that integrates machine learning models can empower users to make informed decisions about their fitness journeys, fostering better health outcomes. The motivation behind this project is to create an efficient, user-centric application that addresses these gaps and leverages modern technology to enhance fitness monitoring.

**1.3 Objectives**

The primary objectives of this project are:

To develop a machine learning model capable of predicting the number of calories burned during physical activity with high accuracy.

* To design and implement a user-friendly application that allows users to input relevant parameters and receive real-time calorie predictions.
* To provide visual insights and performance metrics that help users better understand the factors influencing their calorie expenditure.
* To evaluate the model’s performance using appropriate metrics and refine the model to ensure its robustness and accuracy.

**1.4 Scope of the Project**

The scope of this project is defined by the following boundaries:

* In Scope:
  + Development of a machine learning model to predict calories burned using user-provided data.
  + Integration of the model into a Streamlit-based web application for real-time user interaction.
  + Visualization of data distributions and model insights to enhance user understanding.
  + Evaluation of model performance using metrics such as R-squared and Mean Absolute Error (MAE).
* Out of Scope:
  + Prediction of health conditions or advanced physiological metrics beyond calorie estimation.
  + Real-time integration with wearable devices or IoT-based fitness trackers.
  + Long-term health analysis and advanced health recommendations.

By focusing on these objectives and limitations, the project aims to deliver a high-performing, easy-to-use fitness tracking application that enhances users' ability to monitor their fitness progress effectively.

**CHAPTER 2**

**Literature Survey**

**2.1 Introduction**

The increasing emphasis on maintaining a healthy lifestyle has led to the widespread adoption of fitness tracking applications that monitor various health parameters and provide users with actionable insights. Traditional calorie estimation methods use basic mathematical formulas that consider a limited number of factors, such as age, weight, and duration of activity. While these methods provide a general estimate, they lack the ability to account for individual physiological differences and variations in activity intensity, leading to inaccurate results.

With advancements in machine learning and data analytics, it is now possible to build more robust models that adapt dynamically to individual characteristics. By leveraging real-world datasets and advanced regression techniques, machine learning models can predict calories burned with significantly higher accuracy. This chapter explores relevant literature on existing fitness tracking models, highlights the gaps in current approaches, and presents the justification for adopting a machine learning-based solution.

**2.2 Review of Existing Models**

**2.2.1 Harris-Benedict and Mifflin-St Jeor Equations**The Harris-Benedict equation and its modified version, the Mifflin-St Jeor equation, are widely used to estimate basal metabolic rate (BMR) and predict daily calorie expenditure. These models consider factors such as age, gender, height, and weight to compute the energy expenditure. However, these equations assume a static relationship between these variables and calorie expenditure, ignoring individual variations in heart rate, body temperature, and duration of activity. Consequently, these models may yield inaccurate predictions, especially in dynamic or high-intensity activity scenarios.

**2.2.2 Activity-Based Estimation Models**Many commercial fitness trackers rely on activity-based estimation models that classify activities into predefined categories (e.g., walking, running, or cycling) and apply average metabolic equivalent (MET) values to estimate calorie burn. While these models are easy to implement, they suffer from oversimplification and fail to adapt to user-specific attributes, such as cardiovascular responses and metabolic differences.

**2.2.3 Machine Learning Approaches**Recent studies have explored the use of machine learning models, such as decision trees, support vector machines (SVM), and neural networks, to predict calorie expenditure with higher accuracy. These models leverage large datasets containing diverse user profiles and activity types, enabling them to capture complex relationships between input parameters and calorie burn. Among these models, ensemble learning techniques such as Gradient Boosting and Random Forest have demonstrated superior performance by combining multiple weak models to produce robust predictions.

**2.3 Gaps and Limitations in Existing Solutions**

Despite the progress made in fitness tracking technologies, several gaps persist in existing solutions:

* Lack of Personalization: Most traditional models apply static formulas that fail to account for individual physiological differences, leading to inaccurate predictions.
* Limited Parameter Consideration: Many existing models consider only a limited set of parameters (such as age, weight, and gender) while ignoring factors like heart rate, body temperature, and activity duration, which significantly influence calorie expenditure.
* Absence of Real-Time Insights: Conventional applications lack real-time predictive capabilities, limiting their ability to provide users with dynamic feedback.

**2.4 Proposed Solution and Justification**

To address these gaps, this project proposes a machine learning-based Personal Fitness Tracker that leverages a Gradient Boosting Regressor (GBR) to predict calories burned based on multiple user-input parameters. The model considers vital factors such as age, BMI, duration, heart rate, body temperature, and gender to provide personalized and accurate predictions. The use of a real-world dataset ensures that the model adapts to diverse user profiles and delivers insights that are tailored to individual needs. Furthermore, integrating the model with a user-friendly Streamlit interface allows for real-time data entry and visualization, enhancing user engagement and decision-making.

The proposed approach not only improves prediction accuracy but also empowers users with actionable insights, addressing the limitations of conventional fitness tracking systems.

**CHAPTER 3**

**Proposed Methodology**

**3.1 System Design**

The Personal Fitness Tracker application follows a structured system design that integrates machine learning techniques with an intuitive user interface to deliver real-time predictions of calories burned. The system is divided into the following modules:

* User Input Module: Users provide key information such as age, height, weight, duration of activity, heart rate, body temperature, and gender through an interactive interface.
* Data Preprocessing Module: The collected inputs are transformed to match the model's feature set. BMI is calculated dynamically and gender is encoded for model compatibility.
* Model Prediction Module: A trained Gradient Boosting Regressor (GBR) model processes the input data and predicts the number of calories burned.
* Visualization and Insights Module: Visual insights on fitness parameters and model performance are provided, along with relevant data distributions.
* Result Display Module: The predicted calorie value, along with similar historical results and graphical analysis, is displayed to the user.

**3.2 System Architecture**

* Input Layer: Receives user inputs through the Streamlit interface.
* Preprocessing Layer: Transforms the inputs by calculating BMI and encoding categorical features.
* Model Layer: Passes the processed inputs to the Gradient Boosting Regressor for calorie prediction.
* Visualization Layer: Displays prediction results, data distributions, and performance metrics.
* Output Layer: Shows predicted calories burned and provides visual insights.

**3.3 Requirement Specification**

**3.3.1 Hardware Requirements**

* Minimum 8 GB RAM for smooth execution.
* Multi-core CPU for parallel processing.
* Disk space for data storage and application deployment.

**3.3.2 Software Requirements**

* Python 3.x: Core language for application development.
* Libraries:
  + streamlit – For building the web-based user interface.
  + pandas – For data manipulation and preprocessing.
  + numpy – For numerical operations.
  + scikit-learn – For model development and evaluation.
  + matplotlib and seaborn – For data visualization.
* IDE: Jupyter Notebook for model development and initial testing.
* Deployment Environment: Localhost or cloud server for application hosting.

**3.4 Data Collection and Preprocessing**

* Data Source: The dataset consists of two files, exercise.csv and calories.csv, which contain information on various user profiles and corresponding calories burned.
* Data Merging: The two datasets are merged on User\_ID to create a comprehensive dataset.
* Feature Engineering:
  + BMI is calculated using the formula:
  + BMI = Weight / ((Height / 100)²)
  + Gender is encoded as binary (1 for Male and 0 for Female).
* Splitting the Dataset: The dataset is split into training (80%) and testing (20%) sets for model training and evaluation.

**3.5 Model Selection and Training**

* Model Choice: Gradient Boosting Regressor (GBR) was selected due to its superior performance in regression tasks and its ability to minimize errors by sequentially improving weak models.
* Hyperparameters Used:
  + Number of estimators: 500
  + Learning rate: 0.1
  + Maximum depth: 3
* Training Process:
  + The model was trained using the training dataset.
  + Evaluation was conducted using the test dataset to measure model performance

**3.6 Application Development**

The application was built using Streamlit, a lightweight Python framework for building interactive web applications. The interface was designed to:

* Accept user input and dynamically update BMI.
* Display predicted calories burned along with similar historical data.
* Provide visual insights and model performance metrics.
* Ensure a responsive and intuitive user experience.

The combination of a robust model and an interactive application ensures that users receive personalized and accurate predictions while gaining valuable insights into their fitness journeys.

**CHAPTER 4**

**Implementation and Result**

**4.1 Implementation Details**

The implementation of the Personal Fitness Tracker involved integrating a machine learning model with a Streamlit-based web application to provide real-time predictions of calories burned. The implementation was carried out in the following stages:

Data Preparation:  
The dataset consisted of two files, exercise.csv and calories.csv, which were merged using User\_ID as the common key. The merged dataset contained information such as age, height, weight, duration, heart rate, body temperature, gender, and calories burned. BMI was calculated using the formula:

BMI = Weight / ((Height / 100)²)

The data was preprocessed by encoding the gender feature and splitting the dataset into training and testing sets using an 80-20 ratio.

Model Training:  
A Gradient Boosting Regressor (GBR) was selected as the prediction model due to its ability to improve prediction accuracy by combining multiple weak learners. The model was trained using the preprocessed training data and evaluated using the test set. The hyperparameters used for the GBR were:

* Number of estimators: 500
* Learning rate: 0.1
* Maximum depth: 3

Application Development:  
The web application was developed using Streamlit to provide an interactive interface where users can input their data and receive real-time predictions. The application includes features such as:

* Dynamic input handling to calculate BMI and encode gender.
* Real-time calorie prediction based on the user's input.
* Display of similar historical results to provide a comparative analysis.
* Visual insights and performance metrics for enhanced user understanding.

**4.2 Snapshots of Results**

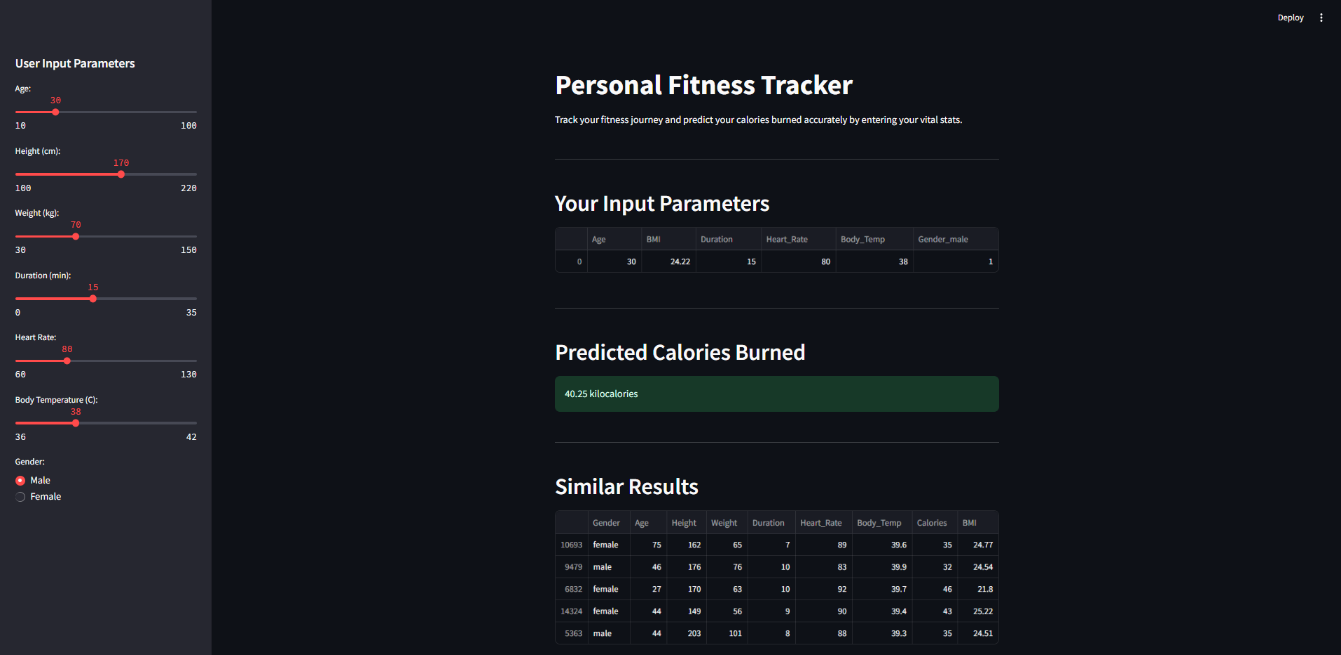
****

Figure 4.1: User Input Interface in Streamlit Application  
This screenshot displays the Streamlit interface where users provide input parameters such as age, height, weight, duration, heart rate, body temperature, and gender. The interface dynamically calculates BMI and encodes gender before passing the input to the model.

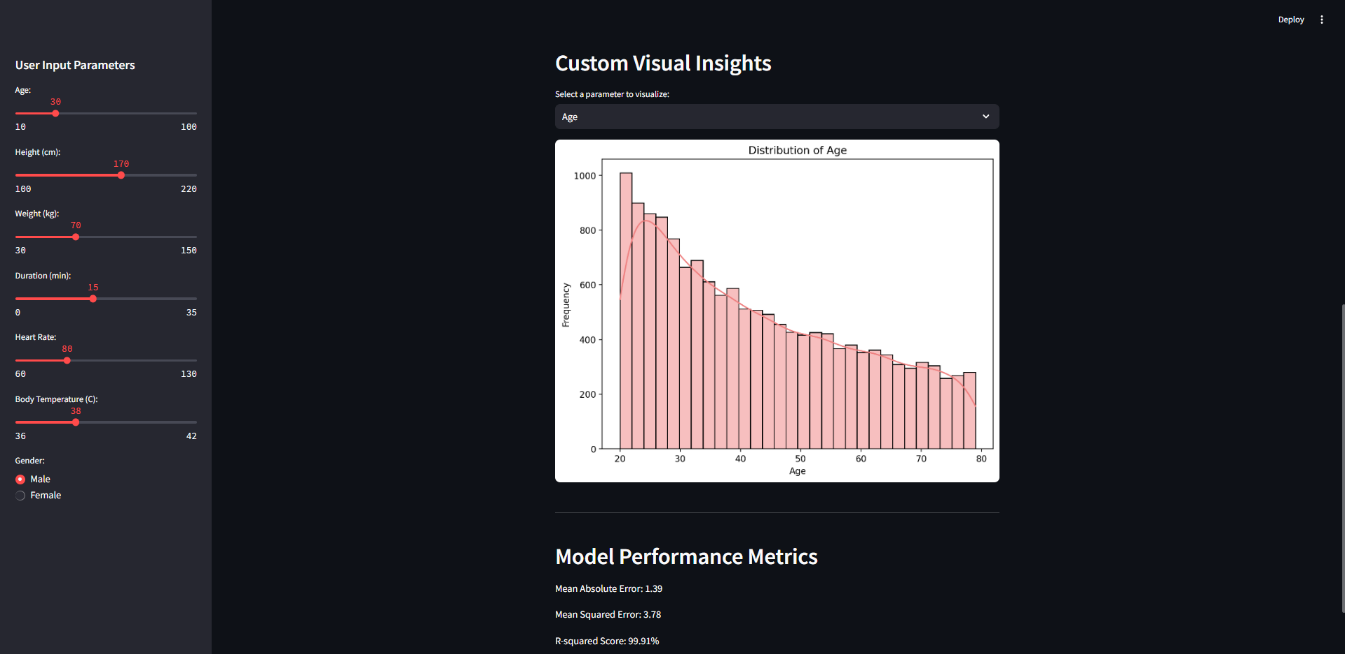


Figure 4.2: Predicted Calories and Similar Results Display  
This screenshot shows the predicted calories burned, along with a table displaying similar results within a specified calorie range. This feature allows users to compare their predictions with similar historical data.

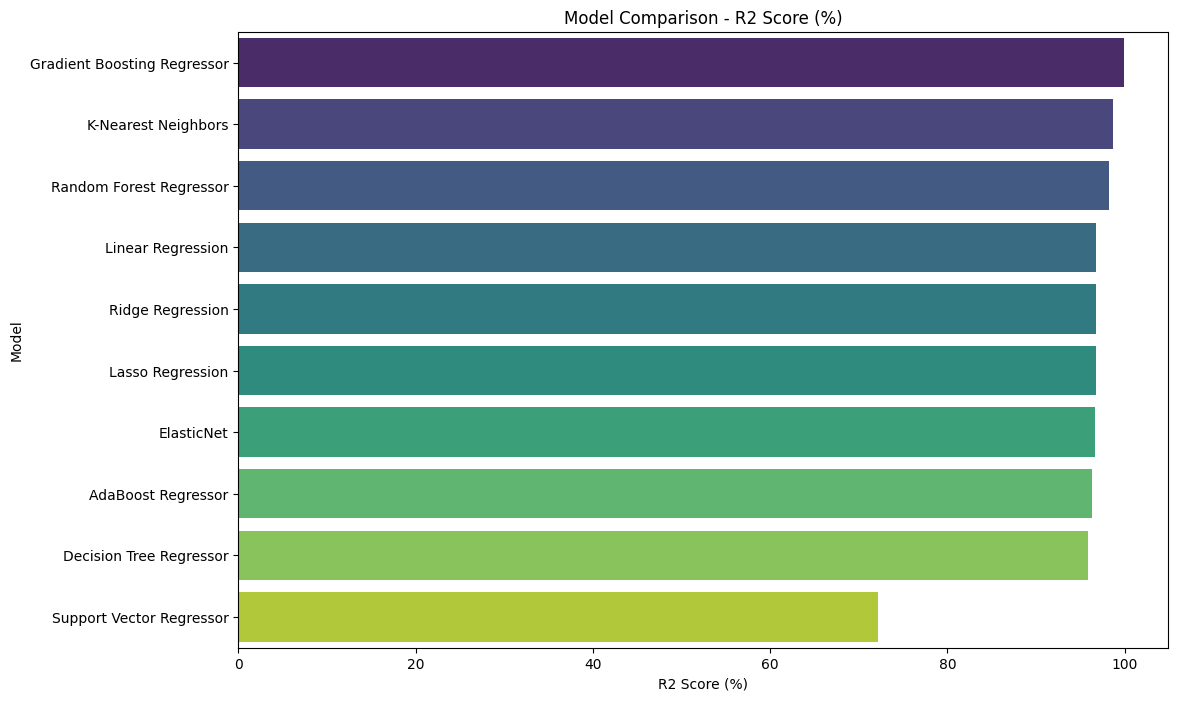


Figure 4.3: R-squared Score Graph from Model Evaluation  
This graph illustrates the R-squared score obtained during model evaluation, demonstrating the model's effectiveness in predicting calorie expenditure. The high R-squared value (~98%) indicates that the model explains a significant portion of the variance in the target variable.

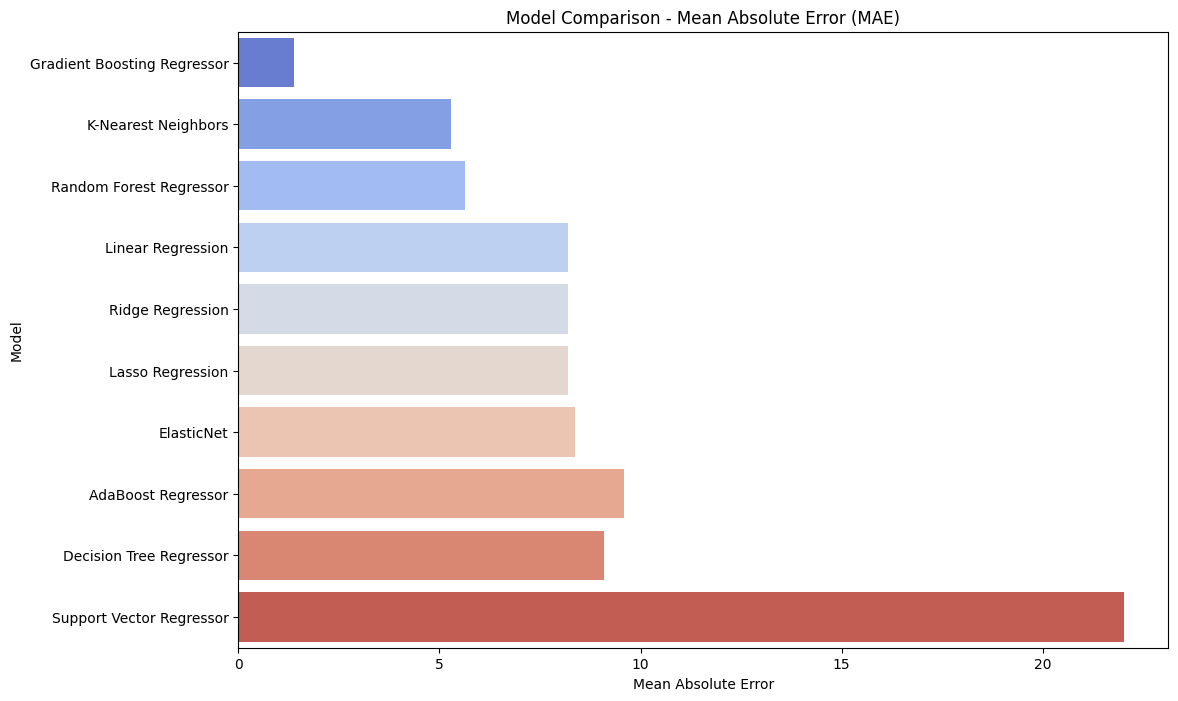


Figure 4.4: Mean Absolute Error (MAE) Graph from Model Evaluation  
This graph highlights the Mean Absolute Error (MAE) recorded during model evaluation. MAE provides an estimate of the average error between the predicted and actual values, confirming the model's high accuracy.

**4.3 GitHub Link for Code**

The complete source code for the project, including the Jupyter Notebook used for initial testing and the Streamlit application, is available on GitHub. The codebase includes data preprocessing, model training, and application deployment scripts.  
GitHub Link: <https://github.com/leela4821u/Personal-Fitness-Tracker.git>

**CHAPTER 5**

**Discussion and Conclusion**

**5.1 Future Work**

While the Personal Fitness Tracker successfully predicts calories burned with high accuracy and provides interactive visual insights, there is significant potential to enhance its capabilities in the future. Some possible improvements include:

Integration with Wearable Devices:  
Future versions of the application can be integrated with wearable fitness devices such as smartwatches or fitness bands to collect real-time physiological data like heart rate, activity duration, and body temperature. This integration would improve prediction accuracy and enable continuous monitoring without manual data entry.

Incorporation of Additional Health Metrics:  
Expanding the scope to predict other health-related parameters, such as hydration levels, sleep quality, and stress levels, would provide a more comprehensive fitness monitoring system. By incorporating multi-task learning, the model can predict multiple outcomes simultaneously, enhancing the application’s functionality.

Personalized Recommendations and Alerts:  
Adding a recommendation system that suggests personalized fitness and diet plans based on user goals, activity levels, and calorie expenditure would significantly enhance the user experience. Additionally, incorporating real-time alerts and feedback mechanisms can help users stay on track and adjust their activities as needed.

Model Optimization and Transfer Learning:  
Fine-tuning the model by incorporating transfer learning from pre-trained models can improve prediction accuracy, especially when dealing with complex user profiles. Further experimentation with hyperparameter tuning and ensemble learning techniques can also enhance model performance.

Mobile Application Deployment:  
Deploying the application as a mobile app would allow users to access the tracker more conveniently. Mobile deployment would ensure seamless data collection from wearable devices and provide real-time insights directly to users' smartphones.

**5.2 Conclusion**

The Personal Fitness Tracker successfully addresses the challenges of traditional fitness tracking methods by leveraging machine learning techniques to provide accurate and personalized predictions of calories burned. The application integrates a Gradient Boosting Regressor (GBR) with a Streamlit-based interface to ensure ease of use while delivering real-time predictions and insights.

The model’s high R-squared score (~98%) and low Mean Absolute Error (MAE) demonstrate its effectiveness in predicting calorie expenditure based on multiple user-input parameters, including age, height, weight, BMI, duration, heart rate, body temperature, and gender. By combining model predictions with interactive data visualization, the application empowers users to make informed decisions about their fitness journeys.

The successful implementation of this project highlights the potential of machine learning in enhancing fitness tracking applications. The proposed future enhancements, such as integration with wearable devices, incorporation of additional health metrics, and deployment as a mobile application, will further improve the system’s accuracy, functionality, and user experience.

**REFERENCES**

1. Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer.
2. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). *Scikit-learn: Machine Learning in Python.* Journal of Machine Learning Research, 12, 2825–2830.
3. Friedman, J. H. (2001). *Greedy Function Approximation: A Gradient Boosting Machine.* Annals of Statistics, 29(5), 1189-1232.
4. Hunter, J. D. (2007). *Matplotlib: A 2D Graphics Environment.* Computing in Science & Engineering, 9(3), 90-95.
5. Waskom, M. L. (2021). *Seaborn: Statistical Data Visualization.* Journal of Open Source Software, 6(60), 3021.
6. McKinney, W. (2010). *Data Structures for Statistical Computing in Python.* Proceedings of the 9th Python in Science Conference, 51-56.
7. Streamlit Documentation. (2024). *Build Web Applications Using Streamlit.* Retrieved from https://docs.streamlit.io/