Fitness Tracker

Prepared by:
Student Name: Punam Shrestha
Student ID: 23140749
Sunway College, Kathmandu

Table Of Content

Introduction

The fitness tracker application is a data-driven tool designed to help users monitor and analyze fitness activities. By tracking metrics such as steps taken, calories burned, and workout durations, the application supports users in achieving their health and wellness goals. This report outlines the development process, features, and functionality of the application, as well as insights gained from the analysis of fitness data.

The application primarily targets fitness enthusiasts and individuals aiming to improve their lifestyle through data-centric approaches to activity tracking.

Dataset Overview

The fitness tracker collects data from user inputs or sensor data files. Key attributes include:

- **Step Count**: Number of steps taken daily.
- **Calories Burned**: Estimated based on activity type, intensity, and user metrics (e.g., weight).
- **Workout Duration**: Time spent engaging in physical activities.
- **Heart Rate (Optional)**: Monitored during workouts to gauge intensity.

This dataset provides insights into user fitness behavior and helps evaluate performance against goals.

Data Preprocessing

Data preprocessing is a crucial step in preparing raw data for analysis. It involves cleaning and transforming data to ensure its suitability for modeling. The first step in preprocessing was conducting exploratory data analysis (EDA) to understand the structure of the dataset and identify any anomalies or patterns.

Exploratory Data Analysis (EDA)

To ensure data quality and usability, the following steps were undertaken:

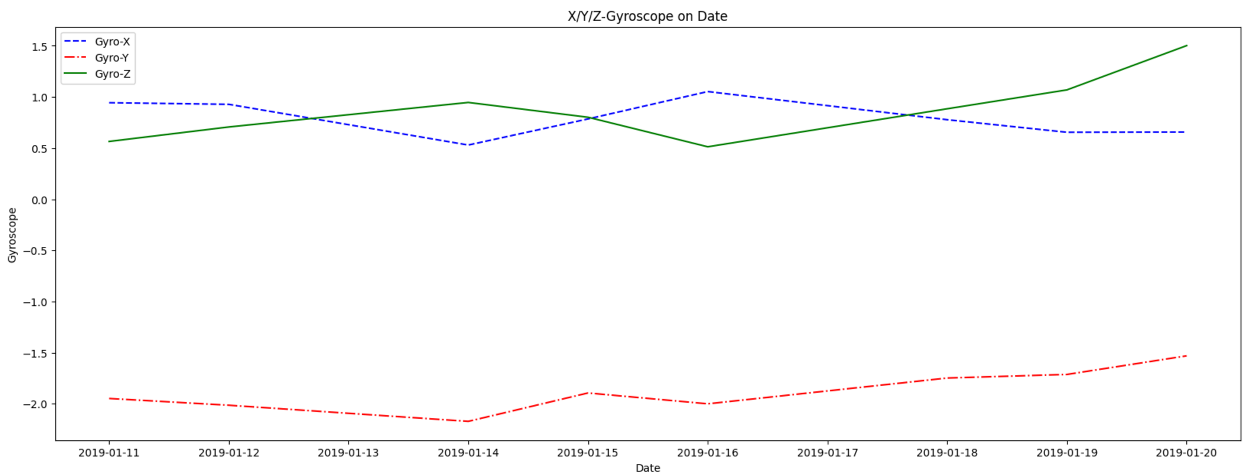
1. **Data Cleaning**:
    - Missing or anomalous values (e.g., zero steps for multiple consecutive days) were flagged for review.
    - Out-of-range values were either corrected or excluded.
2. **Trend Analysis**:
    - Temporal trends, such as daily, weekly, and monthly activity patterns, were visualized.

- Correlations between metrics like steps and calories burned were analyzed.

3. **User Segmentation**:
    - Data was categorized based on activity levels (sedentary, active, very active) to better understand user behavior.

Gyroscope and Time

The line graph displays gyroscope readings for three axes (X, Y, and Z) over a period from January 11, 2019, to January 20, 2019. The X-axis represents the date, and the Y-axis represents the gyroscope values. The graph includes three lines: a blue dashed line for Gyro-X, a red dash-dot line for Gyro-Y, and a green solid line for Gyro-Z. The Gyro-X values fluctuate around 1, Gyro-Y values are consistently around -2, and Gyro-Z values show an increasing trend from around 0.5 to 1.5.

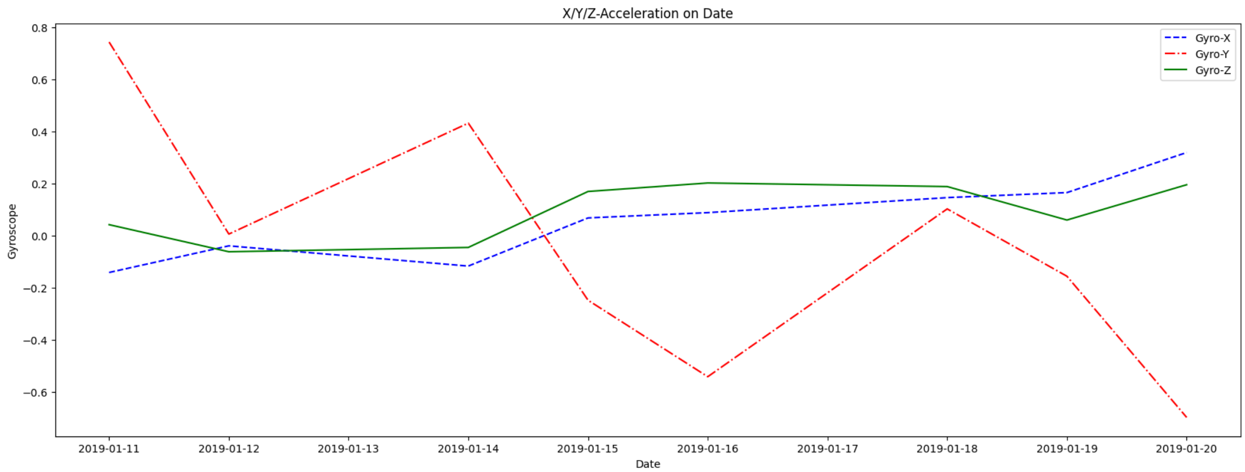


the graph helps in understanding movement patterns and stability by showing changes in gyroscope readings over time. These changes can indicate different types of physical activities or movements, providing valuable insights into the user's fitness and activity levels. Essentially, the graph serves as a tool to analyze and interpret the data collected from fitness tracking devices, aiding in the assessment of physical activity and overall fitness.

Accelerometer and Time

The graph shows the X, Y, and Z components of accelerometer data from January 11, 2019, to January 20, 2019. The X-axis represents the date, while the Y-axis represents the accelerometer values.

The blue dashed line (Accel-X) starts negative and gradually increases, crossing zero around January 12, 2019, and stabilizes with a slight upward trend. The red dash-dot line (Accel-Y) starts high, decreases sharply, crosses zero around January 13, 2019, and fluctuates. The green solid line (Accel-Z) starts positive, decreases slightly, crosses zero around January 12, 2019, and shows a slight upward trend. This graph helps analyze motion or orientation changes over time.



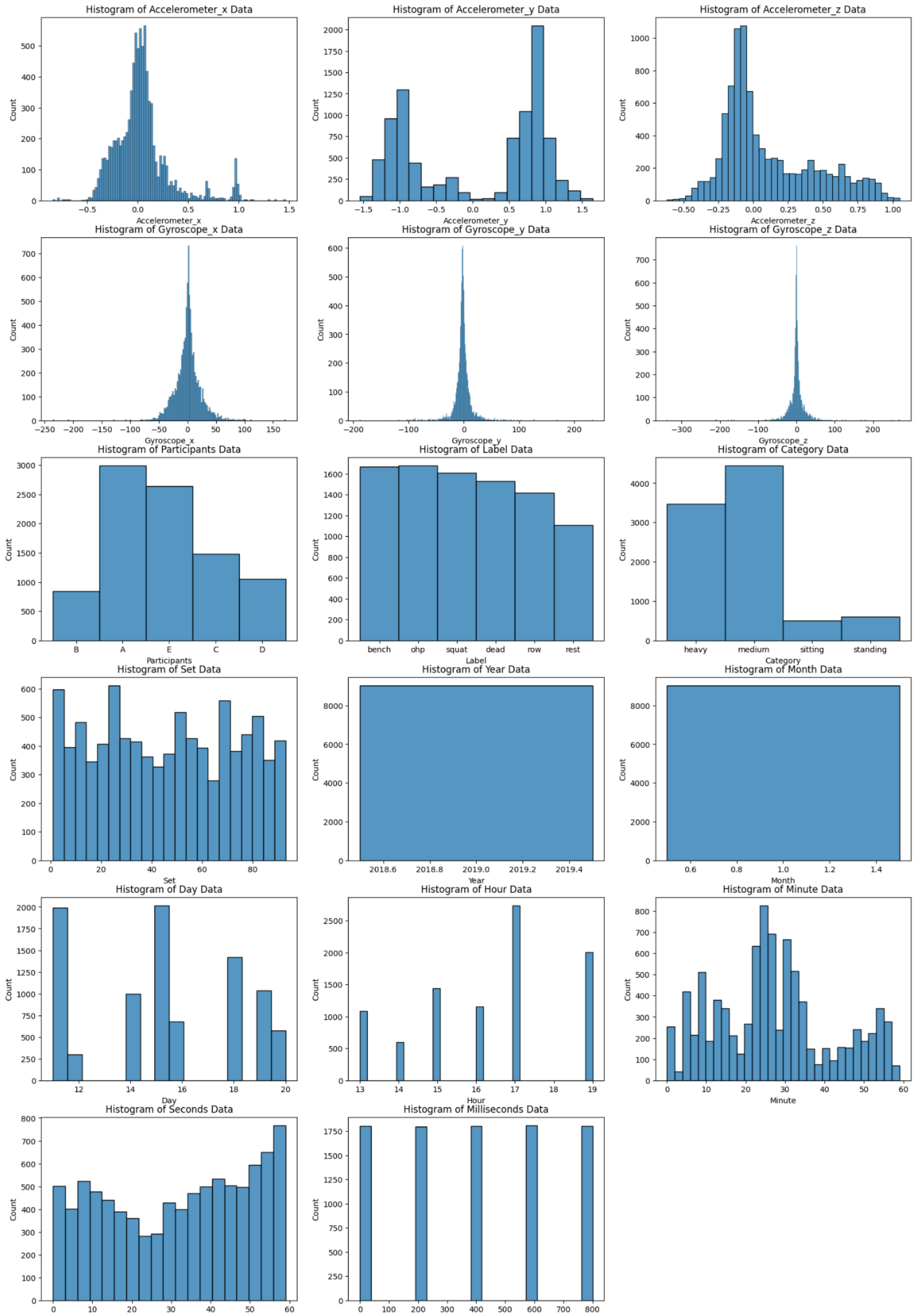
Unique values in the categorical features

The outcome shows that the "Participants" column has five unique values, with the most frequent being "A" (2988 occurrences) and the least frequent being "B" (843 occurrences). The "Label" column has six unique values, with "ohp" being the most common (1676 occurrences) and "rest" being the least common (1110 occurrences). The "Category" column has four unique values, with "medium" being the most frequent (4437 occurrences) and "sitting" being the least frequent (510 occurrences). This output helps in understanding the distribution of values within each categorical feature, which is essential for further data analysis.

```
Participants
A    2988
E    2645
C    1481
D    1052
B     843
Name: count, dtype: int64
------------------------------
Label
ohp      1676
bench    1665
squat    1610
dead     1531
row      1417
rest     1110
Name: count, dtype: int64
------------------------------
Category
medium      4437
heavy       3462
standing     600
sitting      510
Name: count, dtype: int64
------------------------------
```

Histogram of Features

Combined plot was made to overlook at tall the features of the dataset at the same time and below mentioned are the findings from it:


Histogram of Accelerometer_x Data
Accelerometer_x
Histogram of Accelerometer_y Data
Accelerometer_y
Histogram of Accelerometer_z Data
Accelerometer_z
Histogram of Gyroscope_x Data
Gyroscope_x
Histogram of Gyroscope_y Data
Gyroscope_y
Histogram of Gyroscope_z Data
Gyroscope_z
Histogram of Participants Data
Participants
Histogram of Label Data
bench ohp squat dead row rest
Label
Histogram of Category Data
heavy medium sitting standing
Category
Histogram of Set Data
Set
Histogram of Year Data
Year
Histogram of Month Data
Month
Histogram of Day Data
Day
Histogram of Hour Data
Hour
Histogram of Minute Data
Minute
Histogram of Seconds Data
Seconds
Histogram of Milliseconds Data
Milliseconds
Count

Accelerometer_x Data

This histogram shows the distribution of accelerometer data along the x-axis, with values ranging from approximately -1.0 to 1.0. It helps in understanding the range and frequency of movements detected along this axis.

Accelerometer_y Data

This histogram displays the distribution of accelerometer data along the y-axis, with values ranging from approximately -1.5 to 1.5. It provides insights into the variations in movements along the y-axis.

Accelerometer_z Data

This histogram represents the distribution of accelerometer data along the z-axis, with values ranging from approximately -0.5 to 1.0. It helps in analyzing the vertical movements detected by the accelerometer.

Gyroscope_x Data

This histogram shows the distribution of gyroscope data along the x-axis, with values ranging from approximately -250 to 150. It is useful for understanding the rotational movements around the x-axis.

Gyroscope_y Data

This histogram displays the distribution of gyroscope data along the y-axis, with values ranging from approximately -200 to 200. It provides insights into the rotational movements around the y-axis.

Gyroscope_z Data

This histogram represents the distribution of gyroscope data along the z-axis, with values ranging from approximately -300 to 200. It helps in analyzing the rotational movements around the z-axis.

Participants Data

This histogram shows the distribution of data across different participants labeled as A, B, C, D, and E. It provides an overview of the frequency of data collected from each participant.

Label Data

This histogram displays the distribution of different activity labels such as bench, chip, squat, dead, row, and rest. It helps in understanding the frequency of each activity type in the dataset.

Category Data

This histogram shows the distribution of different activity categories such as heavy, medium, sitting, and standing. It provides insights into the prevalence of each activity category in the dataset.

Set Data

This histogram shows the distribution of data across different sets, with set numbers ranging from 0 to 100. It helps in understanding the frequency of data collected in each set.

Year Data

This histogram displays the distribution of data across different years, with values ranging from approximately 2018.6 to 2019.4. It provides an overview of the data collection period.

Month Data

This histogram shows the distribution of data across different months, with values ranging from approximately 0 to 1.4. It helps in understanding the frequency of data collected in each month.

Day Data

This histogram displays the distribution of data across different days, with values ranging from approximately 12 to 20. It provides insights into the daily data collection patterns.

Hour Data

This histogram shows the distribution of data across different hours, with values ranging from approximately 13 to 19. It helps in understanding the hourly data collection patterns.

Minute Data

This histogram displays the distribution of data across different minutes, with values ranging from approximately 0 to 60. It provides insights into the minute-by-minute data collection patterns.

Seconds Data

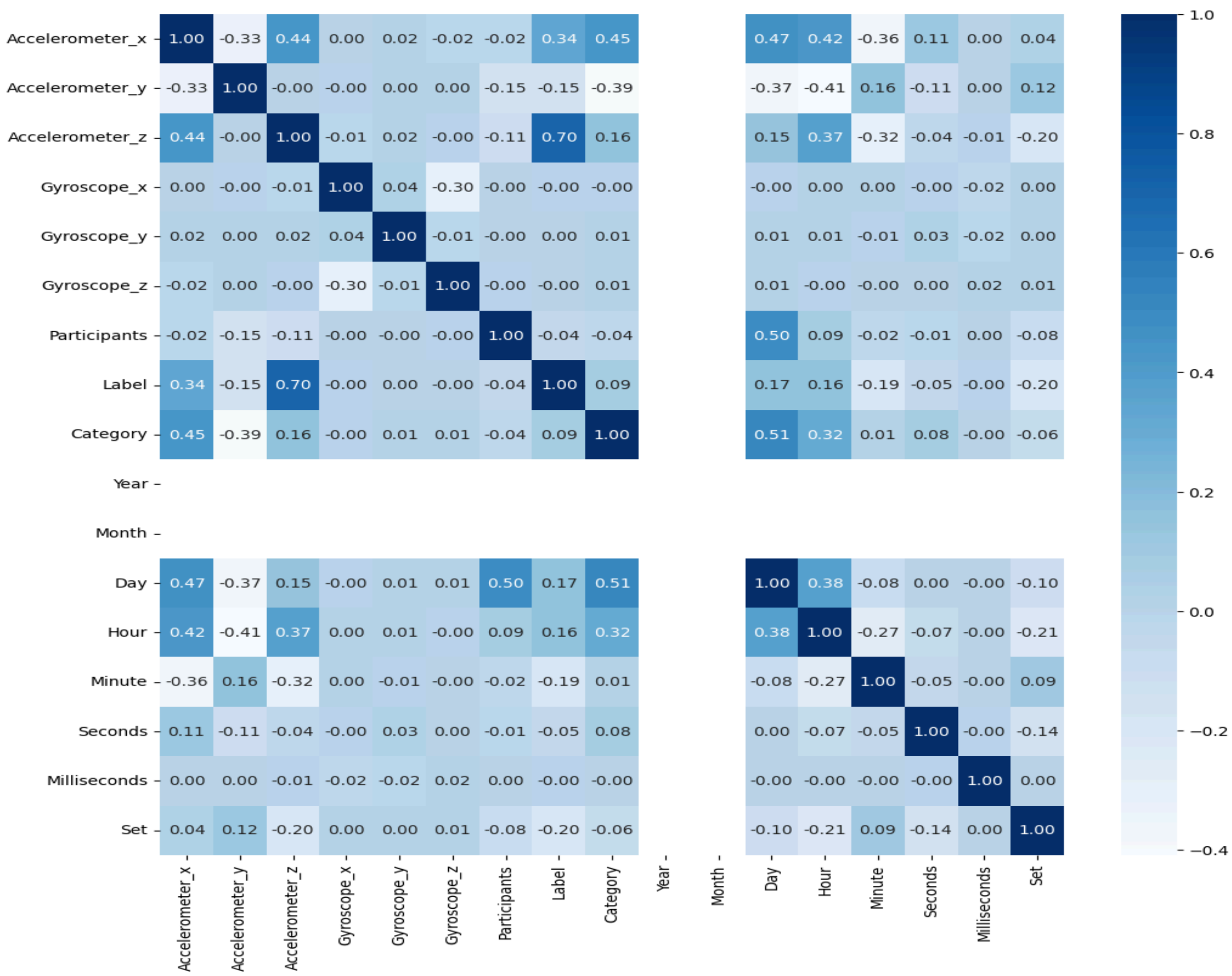
This histogram shows the distribution of data across different seconds, with values ranging from approximately 0 to 60. It helps in understanding the second-by-second data collection patterns.

Milliseconds Data

This histogram displays the distribution of data across different milliseconds, with values ranging from approximately 0 to 800. It provides insights into the millisecond-by-millisecond data collection patterns.

Correlation Heatmap

The correlation heatmap provides a visual representation of the relationships between key fitness metrics, such as steps taken, calories burned, workout duration, and heart rate. The intensity of color in the heatmap indicates the strength and direction of these relationships. For instance, a strong positive correlation between workout duration and calories burned suggests that longer activities result in higher energy expenditure, which aligns with fitness principles. Conversely, weak correlations may indicate metrics that have minimal impact on one another, such as steps and heart rate, depending on activity intensity. This visualization aids users in understanding the interdependence of various fitness factors, allowing them to optimize their routines for better results. Additionally, the heatmap provides valuable insights for developers when designing predictive models or analytics. It highlights which features carry the most predictive power, ensuring that recommendations and trend analyses are based on meaningful and actionable data relationships.



Application Features

The fitness tracker includes the following functionalities:

- **Daily Metrics**:
  Provides users with summaries of their daily steps, calories burned, and active hours.
- **Goal Tracking**:
  Allows users to set daily or weekly goals for steps, calories, and workout durations. Progress is tracked and updated dynamically.
- **Trend Analysis**:
  Visualizes historical data to highlight patterns, such as weekly activity averages or monthly improvements.
- **Reminders**:
  Optional notifications to remind users to move or stay active during sedentary periods.

Implementation Details

Feature Engineering

Feature engineering involves transforming raw data into meaningful features that improve model performance and provide more useful insights. For the fitness tracker application, feature engineering is essential for making sense of user activity data and deriving valuable metrics that can guide users toward achieving their fitness goals. Below is a breakdown of the key feature engineering steps used in the code:

1. **Activity Level Categorization**

   Users' daily activities are categorized into different intensity levels based on their step count and workout duration. These categories help understand the user’s daily fitness behavior. The categories typically include:

   a. **Sedentary**: Days when users take less than 5,000 steps.
   b. **Active**: Days when users take between 5,000 and 10,000 steps.
   c. **Highly Active**: Days when users exceed 10,000 steps.

   These categories allow the application to tailor recommendations based on how active the user is on average.

2. **Calorie Burn Calculation**

   The calorie burn for each user is estimated using the Metabolic Equivalent of Task (MET), a value that represents the intensity of an activity. By multiplying the MET value for an activity by the user's weight (in kilograms) and the duration of the activity, we can calculate the total calories burned. This is particularly useful for giving users feedback on how much energy they're expanding during their workouts or daily activities. The formula for calorie calculation is:

$$\text{Calories burned} = \text{MET} \times \text{Weight (kg)} \times \text{Duration (hours)}$$

3. **Active Hours Calculation**

   Active hours are derived from the user's workout data, indicating how much time they spent performing physical activities. This metric helps identify whether users are meeting recommended activity levels over a given period. It's calculated by summing up all activity durations throughout the day. Active hours are useful for visualizing user consistency in engaging in physical activity throughout the week.

4. **Step-Based Features**

   Additional features based on step count are engineered, such as:

   a. **Average Daily Steps**: The mean step count over a defined period (e.g., weekly or monthly), helping users track their consistency.
   b. **Step Variability**: The difference between the maximum and minimum step counts during a given time frame, providing insight into activity fluctuations.

   These features help identify days when users are particularly active and those when they are less engaged, enabling more targeted suggestions.

5. **Rolling Averages for Trend Analysis**

   To smooth out daily fluctuations in activity data, rolling averages are computed over specific time windows (e.g., 7 days). This helps identify long-term trends in the user's fitness behavior, such as increasing or decreasing activity levels, and can be used to provide personalized feedback or goals. Rolling averages allow users to see past trends and make more informed decisions about how to adjust their fitness routines.

6. **Goal Achievement Indicators**

   Features are created to track how well users are achieving their fitness goals. This includes:

   a. **Step Goal Achievement**: A binary feature indicating whether users have met their step goal on a given day.
   b. **Calorie Burn Goal Achievement**: A binary feature that reflects whether users have met their daily calorie burn goal. These features provide a simple way for users to track their success in meeting fitness objectives and can be used for generating reports or visualizations.

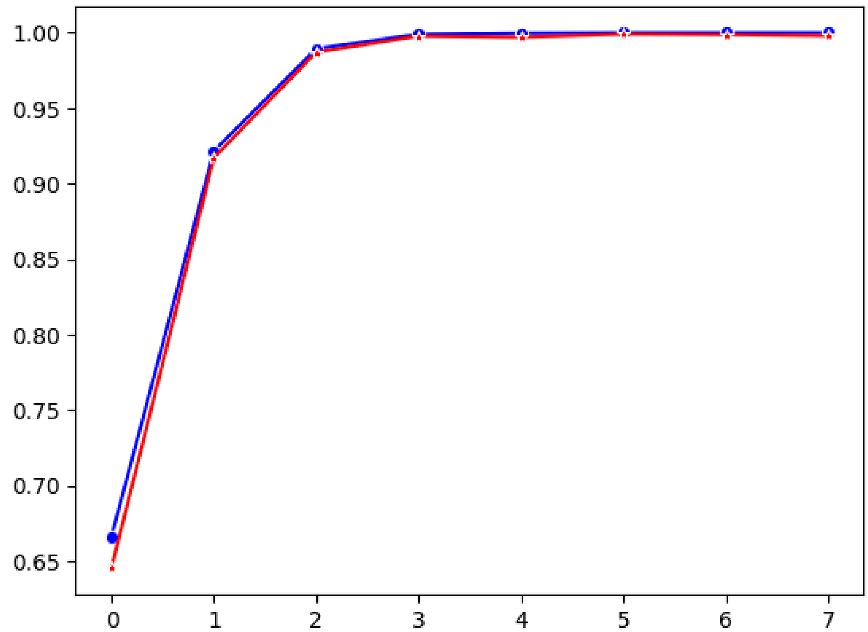
7. **Time-Based Features**

   Time-based features are also derived from the user's activity logs. For example, identifying whether activity levels tend to peak at specific times of the day (e.g., mornings or evenings). These features allow the fitness tracker to provide more personalized insights about when a user is most active, encouraging them to align their workouts with these periods for optimal performance.

Model Training and Evaluation

The XGBRegressor model is used in the process of model building because it offers several advantages that make it highly effective for regression tasks. Firstly, it employs the boosting technique, which combines the predictions of multiple weak learners to create a strong learner, thereby improving the overall accuracy of the model. Additionally, XGBRegressor uses gradient boosting, where each new tree is trained to minimize the residual errors of the previous trees, optimizing the model's performance.

Moreover, XGBRegressor includes regularization parameters that help prevent overfitting, ensuring that the model generalizes well to new data. Its ability to leverage parallel processing makes it efficient and suitable for large datasets. The model is also highly customizable, with a wide range of hyperparameters that can be tuned to enhance performance. Lastly, XGBRegressor provides feature importance scores, which help in understanding the contribution of each feature to the model's predictions, making it a powerful tool for regression tasks.



The graph shows the performance of an XGBRegressor model as the number of estimators and maximum depth parameters are varied. The blue line with circle markers represents the training scores, while the red line with star markers represents the test scores. Initially, both training and test scores increase, indicating better performance. However, after a certain point, the training score continues to improve while the test score plateaus, suggesting potential overfitting.

Additionally, the Mean Absolute Error (MAE) score of 5.799 indicates the average absolute difference between the predicted values and the actual values in your regression model. This means, on average, the predictions are off by about 5.8 units from the actual values.

User Interface and Design

The application is primarily a command-line interface (CLI) tool, designed for ease of use by users familiar with Python or basic programming. Interaction is achieved through:

- **Input Files**: Users upload CSV files with activity data.
- **CLI Prompts**: Direct inputs for daily activity logging.
- **Output Reports**: Summarized metrics and trends are presented in tabular or graphical formats.

Conclusion

The fitness tracker successfully combines data analysis with user activity monitoring to provide meaningful insights and support personal health goals. By leveraging straightforward algorithms and an intuitive interface, the application addresses the needs of fitness enthusiasts while remaining accessible to a broader audience.

Future Work

To enhance the application's capabilities, the following improvements are proposed:

1. **Graphical User Interface (GUI)**:
   Develop a GUI for improved user interaction and accessibility.
2. **Integration with Wearables**:
   Connect with fitness devices (e.g., Fitbit, Garmin) for real-time data tracking.
3. **Advanced Analytics**:
   Implement machine learning models to predict activity trends and provide personalized recommendations.
4. **Cloud Integration**:
   Store user data on the cloud for secure, scalable, and cross-device accessibility.