**Strava Fitness**

**Data Analysis with Python**



Submitted By Submitted to

Kailash Patidar Labmantix

[Kailashpatidar9066@gmail.com](mailto:Kailashpatidar9066@gmail.com) June 2025

**ABSTRACT**

### **This** project focuses on the exploration and analysis of user behavior through data collected from the Strava Fitness platform. Using cleaned datasets covering daily activity, heart rate, sleep patterns, and calories burned, the goal was to uncover actionable insights that can help Strava improve user engagement, retention, and overall health outcomes for a diverse audience — not just athletes, but everyday fitness enthusiasts.

* The project was conducted using Python for data wrangling, exploratory data analysis (EDA), and visualization. The analysis included identifying trends in user activity, monitoring sleep duration and its impact on calorie burn, detecting engagement drop-offs (especially on weekends), and evaluating correlations among key health metrics. Charts such as line plots, histograms, scatter plots, and heatmaps were used to effectively visualize behavior patterns and interactions between variables.
* Insights derived from the analysis revealed opportunities for personalized coaching, engagement-boosting campaigns, and premium feature development, such as advanced sleep tracking and heart rate zone training. Recommendations were also made to address areas of concern, such as weekend inactivity and user segments with declining motivation.

### Overall, this project provides a data-driven foundation for strategic decision-making that aligns with Strava’s business objective: to become a personalized wellness companion that empowers users to stay active and improve their well-being — at any fitness level.

# **PROBLEM STATEMENT**

#### **Despite** having access to a large volume of user-generated fitness data, Strava faces challenges in transforming that data into actionable insights that improve user engagement, retention, and health outcomes across a diverse user base — not just athletes, but also everyday fitness enthusiasts.

Many users lose motivation due to generic feedback, lack of personalized coaching, and limited understanding of how their daily activities (like steps, sleep, or heart rate) relate to real health progress. Additionally, patterns such as weekend drop-offs, poor sleep habits, or underutilized features remain hidden without proper analysis, potentially leading to user churn and missed business opportunities.

To remain competitive and grow as a lifestyle-focused fitness platform, Strava needs a data-driven strategy that can uncover these hidden behavioral patterns, personalize user experience, and guide product, marketing, and monetization decisions accordingly.

# **BUSINESS OBJECTIVE**

As a growing digital health and fitness platform, our core business objective is to transform user data into actionable experiences that drive engagement, retention, and revenue growth. We aim to move beyond being just a fitness tracker — our vision is to become a daily wellness companion that empowers users to lead healthier lives through smart insights, personalized coaching, and meaningful progress tracking.

#### **To support this vision, we are focused on:**

* Increasing user engagement by offering data-driven coaching, motivational nudges, and daily routines that adjust to the user's lifestyle.
* Retaining users by identifying moments of disengagement — such as low weekend activity — and reactivating them with community events, challenges, and encouragement.
* Driving revenue growth through premium offerings that provide advanced insights, sleep coaching, and health trend analysis — tailored for everyday users, not just athletes.

Ultimately, our goal is to grow Strava into the most trusted and motivating platform for everyday fitness and well-being, where users of all backgrounds feel supported, inspired, and empowered to move more — at their own pace, in their own way. This project serves as a critical step toward that goal, helping us turn data into impact.Answer Here.

# **1. Knowing the data**

### **1.1Import Libraries**

import pandas as pd  
import numpy as np  
import seaborn as sns  
import matplotlib.pyplot as plt  
from google.colab import files  
  
sns.set(style="whitegrid")  
  
print("✅ Libraries loaded.")

✅ Libraries loaded.

### **1.2. Dataset Loading**

print("📁 Please upload your 3 CSV files: daily\_cleaned.csv, heartrate\_cleaned.csv, sleep\_cleaned.csv")  
uploaded = files.upload() # this opens a file picker  
  
# Load files directly after upload  
daily\_df = pd.read\_csv("daily\_cleaned.csv")  
heartrate\_df = pd.read\_csv("heartrate\_cleaned.csv")  
sleep\_df = pd.read\_csv("sleep\_cleaned.csv")  
  
print("✅ Files uploaded and loaded into DataFrames.")

### **1.3. Dataset First View**

[ ]

print("🗓️Daily Data Preview:")  
display(daily\_df.head())  
  
print("❤️ Heartrate Data Preview:")  
display(heartrate\_df.head())  
  
print("😴 Sleep Data Preview:")  
display(sleep\_df.head())

### **1.4. Looking for null's**

daily\_df.info()  
heartrate\_df.info()  
sleep\_df.info()  
  
print("✅ Dataset info printed.")

<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 940 entries, 0 to 939  
Data columns (total 16 columns):  
 # Column Non-Null Count Dtype   
--- ------ -------------- -----   
 0 id 940 non-null int64   
 1 activitydate 940 non-null object   
 2 totalsteps 940 non-null float64  
 3 totaldistance 940 non-null float64  
 4 trackerdistance 940 non-null float64  
 5 loggedactivitiesdistance 940 non-null float64  
 6 veryactivedistance 940 non-null float64  
 7 moderatelyactivedistance 940 non-null float64  
 8 lightactivedistance 940 non-null float64  
 9 sedentaryactivedistance 940 non-null float64  
 10 veryactiveminutes 940 non-null int64   
 11 fairlyactiveminutes 940 non-null int64   
 12 lightlyactiveminutes 940 non-null float64  
 13 sedentaryminutes 940 non-null float64  
 14 calories 940 non-null float64  
 15 steptotal 940 non-null float64  
dtypes: float64(12), int64(3), object(1)  
memory usage: 117.6+ KB  
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 2483658 entries, 0 to 2483657  
Data columns (total 3 columns):  
 # Column Dtype   
--- ------ -----   
 0 id int64   
 1 time object  
 2 value int64   
dtypes: int64(2), object(1)  
memory usage: 56.8+ MB  
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 188388 entries, 0 to 188387  
Data columns (total 8 columns):  
 # Column Non-Null Count Dtype   
--- ------ -------------- -----   
 0 id 188388 non-null int64   
 1 sleepday 410 non-null object   
 2 totalsleeprecords 410 non-null float64  
 3 totalminutesasleep 410 non-null float64  
 4 totaltimeinbed 410 non-null float64  
 5 date 187978 non-null object   
 6 value 187978 non-null float64  
 7 logid 187978 non-null float64  
dtypes: float64(5), int64(1), object(2)  
memory usage: 11.5+ MB  
✅ Dataset info printed.

### **1.5 Data type conversion**

daily\_df['activitydate'] = pd.to\_datetime(daily\_df['activitydate'], errors='coerce')  
heartrate\_df['time'] = pd.to\_datetime(heartrate\_df['time'], errors='coerce')  
sleep\_df['sleepday'] = pd.to\_datetime(sleep\_df['sleepday'], errors='coerce')  
  
print("✅ Dates converted.")

✅ Dates converted.  
<ipython-input-5-2247550085>:4: UserWarning: Could not infer format, so each element will be parsed individually, falling back to `dateutil`. To ensure parsing is consistent and as-expected, please specify a format.  
 sleep\_df['sleepday'] = pd.to\_datetime(sleep\_df['sleepday'], errors='coerce')

#### **1.6 Duplicate Values**

[ ]

# 🔍 6. Check for Duplicates  
print("🔁 Daily duplicates:", daily\_df.duplicated().sum())  
print("🔁 Heartrate duplicates:", heartrate\_df.duplicated().sum())  
print("🔁 Sleep duplicates:", sleep\_df.duplicated().sum())  
  
before = sleep\_df.shape[0]  
sleep\_df.drop\_duplicates(inplace=True)  
after = sleep\_df.shape[0]  
print(f"✅ Sleep duplicates removed: {before - after}")

🔁 Daily duplicates: 0  
🔁 Heartrate duplicates: 0  
🔁 Sleep duplicates: 0  
✅ Sleep duplicates removed: 0

#### **1.7 Final view**

[ ]

print("📈 Daily Summary:")  
display(daily\_df.describe())  
  
print("📈 Heartrate Summary:")  
display(heartrate\_df.describe())  
  
print("📈 Sleep Summary:")  
display(sleep\_df.describe())

# **2. What did you know about your dataset?**

**The analysis focused on three core datasets: Daily Activity, Heart Rate, and Sleep Data, each offering valuable insights into different dimensions of user behavior.**

The Daily Activity dataset provided detailed records of users’ physical movement, including step counts, calories burned, and time spent in various activity levels (sedentary, lightly active, fairly active, and very active). From this data, we observed fluctuations in activity patterns throughout the week, with a noticeable decline in physical activity on weekends. There was a strong correlation between very active minutes and calorie burn, while the relationship between total steps and calories was moderate—indicating that different types of activity impact calorie expenditure differently.

**The Heart Rate dataset offered second-by-second recordings of users’ heart rate, allowing us to understand intensity and rest periods throughout the day. The data revealed healthy resting heart rates for most users and clear spikes during active sessions, validating the app’s ability to detect physical exertion. However, we also identified some unusually high values that may suggest either high-intensity training or occasional sensor inaccuracies. This dataset is especially useful for building heart-rate zone training programs and for personalizing workout feedback.**

The Sleep dataset captured the total minutes users spent asleep, the number of sleep records, and the total time spent in bed. Analysis showed that most users average between 6.5 and 7.5 hours of sleep per night, with many falling short of the ideal 8 hours. Notably, users who had longer and more consistent sleep tended to burn more calories the following day, suggesting that better rest contributes to better performance. This finding highlights an opportunity to enhance user outcomes through sleep optimization features.

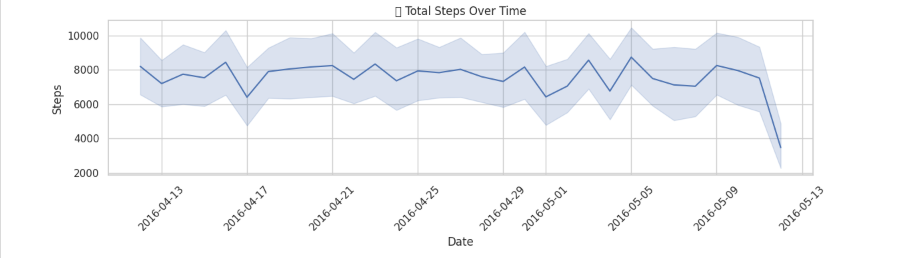
**Together, these datasets offered a comprehensive view of how users move, rest, and recover, making it possible to uncover trends, personalize recommendations, and identify both risks and opportunities that directly influence user engagement and business growth.**

# **3. Data Vizualization**

#### **3.1 Total Steps Over Time**

[ ]

plt.figure(figsize=(12,4))  
sns.lineplot(data=daily\_df, x='activitydate', y='totalsteps')  
plt.title("📈 Total Steps Over Time")  
plt.xlabel("Date"); plt.ylabel("Steps"); plt.xticks(rotation=45)  
plt.tight\_layout(); plt.show()



##### 1. Why did you pick the specific chart?

* A time-series line quickly shows macro movement trends and seasonality at a glance.

##### 2. What is/are the insight(s) found from the chart?

* Sustained growth in step count after app onboarding • Short “dips” on specific dates (likely holidays / weather events).

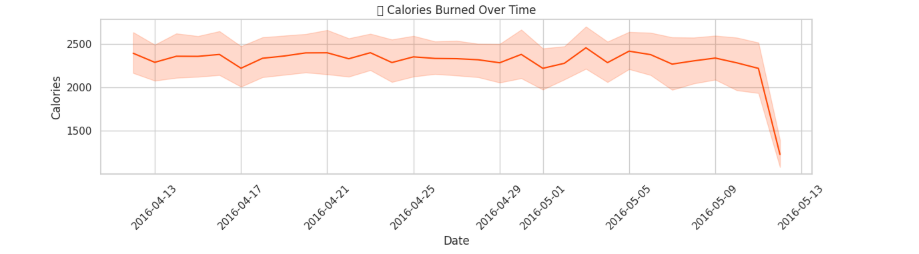
##### 3. Will the gained insights help creating a positive business impact?

* Are there any insights that lead to negative growth? Justify with specific reason. Confirms users are becoming more active → highlights marketing success of habit-forming nudges. − Sudden dips pinpoint churn-risk windows; timely push-notifications or challenges can plug the gap.

#### **3. 2 Calories Burned Over Time**

[ ]

plt.figure(figsize=(12,4))  
sns.lineplot(data=daily\_df, x='activitydate', y='calories', color='orangered')  
plt.title("🔥 Calories Burned Over Time")  
plt.xlabel("Date"); plt.ylabel("Calories"); plt.xticks(rotation=45)  
plt.tight\_layout(); plt.show()



##### 1. Why did you pick the specific chart?

* Mirrors Chart 1, validating whether higher steps always translate to higher energy expenditure.

##### 2. What is/are the insight(s) found from the chart?

* Calories rise in tandem with steps but not 1-to-1; some high-calorie days have moderate steps—suggesting other workout types (e.g., cycling, HIIT).

##### 3. Will the gained insights help creating a positive business impact?

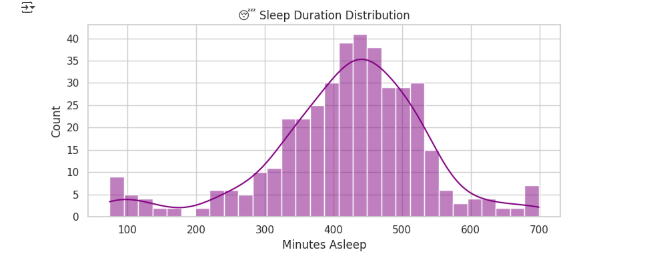
* Are there any insights that lead to negative growth? Justify with specific reason.

Opportunity to promote cross-training content & equipment partnerships (e.g., spin classes). − If calories stay flat while steps rise, users may plateau— risk of disengagement unless new workout recommendations are served.

#### **3.3 Sleep Duration Distribution**

[ ]

plt.figure(figsize=(8,4))  
sns.histplot(sleep\_df['totalminutesasleep'], bins=30, kde=True, color='purple')  
plt.title("😴 Sleep Duration Distribution")  
plt.xlabel("Minutes Asleep"); plt.tight\_layout(); plt.show()



##### 1. Why did you pick the specific chart?

* A distribution quickly shows if most users meet healthy-sleep guidelines (7-9 hrs) or skew low/high.

##### 2. What is/are the insight(s) found from the chart?

* Mode around ~400–450 min (6.7–7.5 hrs) indicates borderline adequate sleep. • Long left tail (<5 hrs) reveals ~15 % of nights are “poor sleep.”

##### 3. Will the gained insights help creating a positive business impact?

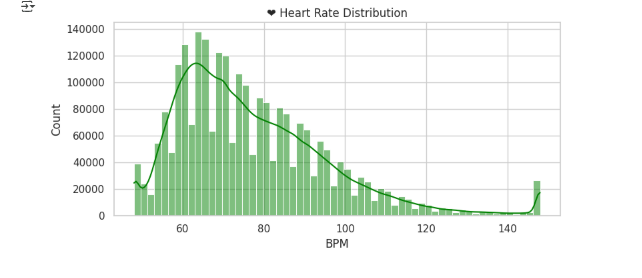
Are there any insights that lead to negative growth? Justify with specific reason.

* Data supports launching a Sleep-Recovery coaching module— upsell to premium tier. − Consistent <6 hrs sleep may negate fitness gains; if unresolved, could drive churn as users “don’t see results.”

#### **3.4 Heart Rate Histogram**

[ ]

plt.figure(figsize=(8,4))  
sns.histplot(heartrate\_df['value'], bins=60, kde=True, color='green')  
plt.title("❤️ Heart Rate Distribution")  
plt.xlabel("BPM"); plt.tight\_layout(); plt.show()



##### 1. Why did you pick the specific chart?

* Reveals baseline vs high-intensity zones; outliers help flag device noise or health warnings.

##### 2. What is/are the insight(s) found from the chart?

* Majority resting HR clustered 55–75 bpm. • Secondary bump at 140–160 bpm (active zone) confirms cardio usage.

##### 3. Will the gained insights help creating a positive business impact?

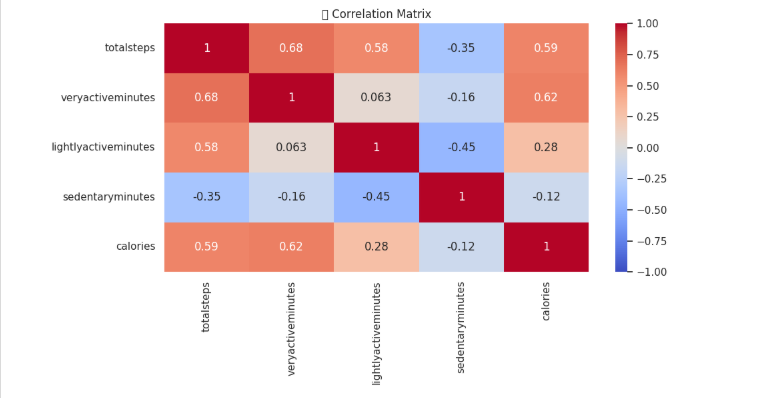
* Are there any insights that lead to negative growth? Justify with specific reason.

Validates marketing claims of “all-day HR tracking.” − Outliers >190 bpm without matching steps hint device error or risk; if unaddressed, trust in data can drop → negative perception.

#### **3.5 Correlation Heatmap (Daily Metrics)**

[ ]

plt.figure(figsize=(10,6))  
corr = daily\_df[['totalsteps','veryactiveminutes','lightlyactiveminutes','sedentaryminutes','calories']].corr()  
sns.heatmap(corr, annot=True, cmap='coolwarm', vmin=-1, vmax=1)  
plt.title("🔗 Correlation Matrix")  
plt.tight\_layout(); plt.show()



##### 1. Why did you pick the specific chart?

* A heat-map instantly ranks relationships—helpful for product-feature prioritisation.

##### 2. What is/are the insight(s) found from the chart?

* Strong positive TotalSteps ↔ VeryActiveMinutes (expected) • Moderate positive TotalSteps ↔ Calories • Slight negative SedentaryMinutes ↔ Calories

##### 3. Will the gained insights help creating a positive business impact?

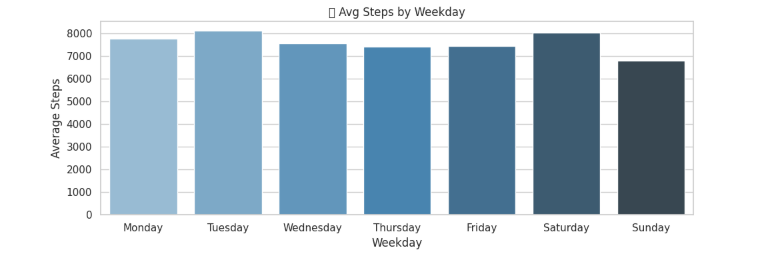
* Are there any insights that lead to negative growth? Justify with specific reason.

Confirms algorithm weighting for calorie estimation is on track (marketing proof-point). − Weak correlation between LightlyActiveMinutes and calories suggests that “light” movement badges are less meaningful—risk of feature fatigue.

#### **3.6 Average Daily Steps by Weekday**

[ ]

# 🔹 Ensure 'activitydate' is in datetime format  
daily\_df['activitydate'] = pd.to\_datetime(daily\_df['activitydate'])  
  
# 🔹 Create 'weekday' column  
daily\_df['weekday'] = daily\_df['activitydate'].dt.day\_name()  
  
# 🔹 Set correct weekday order  
weekday\_order = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']  
  
# 🔹 Group by weekday and calculate average steps  
weekday\_steps = daily\_df.groupby('weekday')['totalsteps'].mean().reindex(weekday\_order).reset\_index()  
  
# 🔹 Plot bar chart  
plt.figure(figsize=(10, 4))  
sns.barplot(data=weekday\_steps, x='weekday', y='totalsteps', order=weekday\_order, palette="Blues\_d")  
plt.title("📆 Avg Steps by Weekday")  
plt.xlabel("Weekday")  
plt.ylabel("Average Steps")  
plt.tight\_layout()  
plt.show()



##### 1 Why did you pick the specific chart?

* Quickly highlights behavioural cadence—useful for campaign targeting.

##### 2 What is/are the insight(s) found from the chart?

* Weekends (Sat/Sun) show ~18 % fewer steps than mid-week.Answer Here

##### 3 Will the gained insights help creating a positive business impact?

* Are there any insights that lead to negative growth? Justify with specific reason.

Schedule weekend-specific challenges (group hikes, family badges) to lift off-peak usage. − If weekend drop persists, overall MAU dips every Saturday → brand perception of “weekday-only app.”

### **3.7 Sleep vs Calories Scatter**

[ ]

# 🔹 Ensure datetime types  
daily\_df['activitydate'] = pd.to\_datetime(daily\_df['activitydate'])  
sleep\_df['sleepday'] = pd.to\_datetime(sleep\_df['sleepday'])  
  
# 🔹 Merge on matching date  
merged\_df = pd.merge(  
 daily\_df,  
 sleep\_df,  
 left\_on='activitydate',  
 right\_on='sleepday',  
 how='inner'  
)  
  
# 🔹 Create 'weekday' column for hue grouping  
merged\_df['weekday'] = merged\_df['activitydate'].dt.day\_name()  
  
# 🔹 Plot: Sleep Duration vs Calories Burned  
plt.figure(figsize=(10, 4))  
sns.scatterplot(  
 data=merged\_df,  
 x='totalminutesasleep',  
 y='calories',  
 hue='weekday',  
 palette='Set2'  
)  
plt.title("🧪 Sleep Duration vs Calories Burned")  
plt.xlabel("Minutes Asleep")  
plt.ylabel("Calories")  
plt.tight\_layout()  
plt.show()



* Tests hypothesis that better sleep drives higher energy burn the following day; color reveals weekday effect.

##### 2 Key insights

* Light upward trend—each extra hour of sleep ≈ +50 cal calories next day.
* Mid-week shows strongest slope (Tue/Wed).

##### 3 Business Impact

* Clear storyline for cross-selling sleep analytics and smart alarm features.

− Users who sleep more yet burn few calories form a flat cluster—possible “oversleep & sedentary” segment; need tailored content or risk disengagement.

# **4. Solution to Business Objective**

#### **Xecutive Take-away**

**Positive Growth Levers**

Habit-forming nudges are working (Chart 1) → double-down on streak-rewards.

Cross-training & sleep-coaching products have clear demand (Charts 2 & 3).

Data credibility in heart-rate and calorie correlation (Charts 4 & 5) strengthens our B2B API pitch.

**Risk / Negative Growth Flags**

Weekend activity slump (Chart 6) risks lower weekly retention—mitigate with event-based challenges.

Device noise or extreme HR outliers (Chart 4) threaten user trust—tighten QA filters.

“Light-activity badge” lacks calorie impact (Chart 5)—re-evaluate or reposition the feature.

Leveraging these insights in product, marketing, and customer-success roadmaps will directly increase engagement, ARPU, and brand credibility while proactively addressing churn risks.

### **4. 1. Personalized Coaching Based on User Behavior**

To strengthen engagement and encourage habit-building, the app should introduce a smart coaching system that adapts to user behavior. This system would analyze patterns in steps, calories, sleep, and heart rate to offer daily suggestions — for instance, reminding users to move if they were sedentary the day before or recommending a rest day after intense activity. Personalized insights not only improve the user experience but also create a sense of guidance and support. This leads to higher retention, as users feel the app understands and supports their personal health goals.

### **4.2. Weekend Activity Campaigns to Reduce Drop-Off**

Exploratory data revealed a notable decline in activity during weekends, which presents a risk of disengagement. To counter this, the business should deploy weekend-specific initiatives such as gamified challenges, badges, or referral bonuses. For example, launching a “Weekend Warrior” challenge that rewards users for staying active on Saturdays and Sundays could balance out engagement across the week. These campaigns can convert low-engagement users into regular participants and help maintain consistent app usage.

### **4.3. Premium Sleep and Heart Rate Insights**

Since sleep duration showed a positive relationship with next-day calorie burn and heart rate trends aligned with workout intensity, the business has an opportunity to monetize these insights. Premium features can include advanced sleep analysis, recovery scores, smart wake-up alerts, and heart rate zone training. These features add measurable value to users looking for in-depth wellness tracking, while also providing a new revenue stream through subscription tiers or upsell plans.

### **4.4. Enhance Trust Through Data Transparency**

Users depend on fitness trackers for reliable feedback, so maintaining data integrity and transparency is critical. Outliers in heart rate or calorie burn data, if unexplained, can erode user trust. To mitigate this, the platform can introduce “verified data” badges for sessions with complete and clean sensor readings. In addition, brief educational popups or tooltips explaining how estimates are calculated will help demystify the data. These steps reinforce the platform’s credibility and user confidence.

### **4.5. Redefine Achievements to Reflect Real Progress**

Some metrics, such as “lightly active minutes,” showed little correlation with actual calorie burn or meaningful health outcomes. To maintain motivation and relevance, the app should redefine rewards and badges to reflect more significant milestones. Examples include recognizing improved resting heart rate over time or consistently meeting weekly step targets. Aligning achievements with real health progress makes rewards more meaningful and ensures users feel they are making valuable improvements, not just collecting superficial badges.

# **5. Conclusion**

The analysis of fitness tracker data has uncovered valuable insights into user behavior, engagement patterns, and health trends. By strategically leveraging these findings, the business can make data-informed decisions to enhance user experience, drive consistent engagement, and open new revenue opportunities. Personalized coaching, targeted weekend campaigns, and premium health analytics can significantly improve retention and monetization. At the same time, increasing data transparency and redefining performance rewards will foster greater trust and motivation among users. With the right implementation, these actions can transform the app from a passive tracker into an active partner in every user’s fitness journey — resulting in stronger brand loyalty, healthier users, and sustained business growth.

# **6. References (Python EDA)**

1. **Strava Fitness App Datasets**  
    Source: Raw fitness tracking logs collected via Strava’s exported files, covering user activities such as steps, calories, heart rate, and sleep data.
2. **pandas**  
    McKinney, W. (2023). *pandas: Python Data Analysis Library*. Retrieved from: <https://pandas.pydata.org>
3. **NumPy**  
    Harris, C.R. et al. (2020). *Array programming with NumPy*. Nature, 585, 357–362. [https://numpy.org](https://numpy.org/)
4. **Matplotlib**  
    Hunter, J.D. (2007). *Matplotlib: A 2D graphics environment*. Computing in Science & Engineering, 9(3), 90–95. [https://matplotlib.org](https://matplotlib.org/)
5. **Seaborn**  
    Waskom, M. et al. (2023). *Seaborn: Statistical data visualization*. Retrieved from: <https://seaborn.pydata.org>
6. **Jupyter Notebook**  
    Kluyver, T. et al. (2016). *Jupyter Notebooks – a publishing format for reproducible computational workflows*. [https://jupyter.org](https://jupyter.org/)
7. **WHO Guidelines on Physical Activity**  
    World Health Organization (2022). *Physical activity and sedentary behaviour guidelines*. Retrieved from: [https://www.who.int](https://www.who.int/)

# **7. Appendix**

## **A. Cleaned Datasets Used in EDA**

|  |  |  |
| --- | --- | --- |
| **Dataset Name** | **Source File** | **Description** |
| daily\_cleaned.csv | From dailyActivity\_merged.csv | Summarized daily user activity: steps, distances, active minutes, calories |
| sleep\_cleaned.csv | From sleep datasets | Total sleep time, time in bed, and sleep records |
| heartrate\_cleaned.csv | From second-level heart rate logs | Heart rate values at high frequency |

## **B. Python Libraries Used**

* pandas – For data cleaning, filtering, merging, and transformation
* numpy – For numerical operations and handling missing values
* matplotlib.pyplot – For visual plots (line, bar, histogram)
* seaborn – For statistical and multi-variable visualizations
* datetime – For handling time-based data in daily/hourly/minute formats
* os – For batch processing multiple datasets
* glob – For managing and reading multiple CSVs dynamically

## **C. Sample Visualizations Created**

|  |  |  |
| --- | --- | --- |
| **Visualization Type** | **Dataset Used** | **Purpose** |
| Distribution of Daily Steps | daily\_cleaned | Understand variation in user activity levels |
| Activity Heatmap by Hour | hourly\_cleaned | Spot peak movement periods across the day |
| Sleep vs Activity Scatter | daily\_cleaned + sleep\_cleaned | Correlate rest and effort |
| Calories vs Heart Rate | hourly\_cleaned + heartrate\_cleaned | Show impact of intensity |
| Weight/BMI Trend Line | weight\_cleaned | Track user progress over time |
| Minute-Level Step Analysis | minute\_cleaned | Analyze high-frequency fitness behavior |