Bellabeat Case Study

How Can a Wellness Technology Company Play It Smart?

Capstone Project – Google Data Analytics Professional Certificate

Introduction

This case study is completed as part of the **Google Data Analytics Professional Certificate Capstone Project**. The purpose of this analysis is to apply the six phases of the data analysis process **Ask, Prepare, Process, Analyze, Share, and Act** to a real-world business scenario.

Bellabeat is a wellness technology company that designs smart products to help women better understand their health, habits, and daily activity. To support future growth, Bellabeat's leadership believes that analyzing **non-Bellabeat wearable data**, such as Fitbit fitness tracker data, can uncover valuable insights into user behavior.

In this project, publicly available Fitbit data is analyzed to identify trends in physical activity, sedentary behavior, and app usage. These insights are then used to develop **data-driven recommendations** that can help inform Bellabeat's marketing strategy and customer engagement efforts.

Author: Faizo Tahlil

Step 1: Ask

1.1 Background

Bellabeat is a high-tech wellness company founded in 2013 that focuses on creating smart products designed specifically for women. Its products help users track daily activity, health habits, and overall wellness, empowering them to make informed lifestyle decisions.

The co-founder and Chief Creative Officer, **Urška Sršen**, believes that analyzing data from wearable devices outside of Bellabeat's ecosystem can provide valuable insights into how consumers interact with fitness and wellness technology. By understanding these behaviors, Bellabeat can better position its products and marketing strategies to meet customer needs and support long-term growth.

1.2 Business Task

Analyze **Fitbit fitness tracker data** to understand how users track their physical activity and daily habits, and identify trends that can help guide **Bellabeat's marketing strategy**.

1.3 Business Objectives

This analysis aims to answer the following questions:

- What trends can be identified in Fitbit users' activity and behavior?
- How can these trends be applied to Bellabeat customers?
- How can these insights help inform Bellabeat's marketing and engagement strategy?

1.4 Key Stakeholders

- **Urška Sršen** – Co-founder and Chief Creative Officer of Bellabeat
- **Sando Mur** – Co-founder and key member of the Bellabeat executive team
- **Bellabeat Marketing Analytics Team** – Responsible for using data insights to guide marketing decisions

STEP 3: PROCESS

In this step, we load the Fitbit daily activity dataset, check data quality, convert the date column, and create a few new features (day of week, total minutes, and total hours). This prepares the dataset for analysis and visualization.

3.1 Preparing the Environment

We import the libraries needed for data cleaning, analysis, and visualization.

Code steps:

1. Import libraries
2. Set display options

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

pd.set_option('display.max_columns', None)
```
11] ✓ 0.0s

3.2 Importing the Dataset

We load the daily activity dataset and preview a few rows to confirm it imported correctly.

Code steps:

1. Read the CSV
2. Preview the first 5 rows

```
df = pd.read_csv("dailyActivity_merged.csv")
df.head()
```
✓ 0.0s

	Id	ActivityDate	TotalSteps	TotalDistance	TrackerDistance	LoggedActivitiesDistance	VeryActiveDistance	ModeratelyActiveDistance	LightActiveDistance	SedentaryActiveDistance	VeryActiveMinutes	FairlyActiveMinutes	LightlyActiveMinutes	SedentaryMinutes	Calories
0	1503960366	3/25/2016	11004	7.11	7.11	0.0	2.57	0.46	4.07	0.0	33	12	205	804	1819
1	1503960366	3/26/2016	17609	11.55	11.55	0.0	6.92	0.73	3.91	0.0	89	17	274	588	2154
2	1503960366	3/27/2016	12736	8.53	8.53	0.0	4.66	0.16	3.71	0.0	56	5	268	605	1944
3	1503960366	3/28/2016	13231	8.93	8.93	0.0	3.19	0.79	4.95	0.0	39	20	224	1080	1932
4	1503960366	3/29/2016	12041	7.85	7.85	0.0	2.16	1.09	4.61	0.0	28	28	243	763	1886

3.3 Initial Data Checks

We check the dataset size, column types, missing values, and duplicates.

Code steps:

1. Check rows/columns
2. Check column data types + non-null counts
3. Check missing values
4. Check duplicates

df.shape
df.info()
df.isna().sum()

```
df.shape
df.info()
✓ 0.0s
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 457 entries, 0 to 456
Data columns (total 15 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Id                        457 non-null    int64
 1   ActivityDate              457 non-null    object
 2   TotalSteps                457 non-null    int64
 3   TotalDistance             457 non-null    float64
 4   TrackerDistance           457 non-null    float64
 5   LoggedActivitiesDistance  457 non-null    float64
 6   VeryActiveDistance        457 non-null    float64
 7   ModeratelyActiveDistance  457 non-null    float64
 8   LightActiveDistance       457 non-null    float64
 9   SedentaryActiveDistance   457 non-null    float64
 10  VeryActiveMinutes         457 non-null    int64
 11  FairlyActiveMinutes       457 non-null    int64
 12  LightlyActiveMinutes      457 non-null    int64
 13  SedentaryMinutes          457 non-null    int64
 14  Calories                  457 non-null    int64
dtypes: float64(7), int64(7), object(1)
memory usage: 53.7+ KB
```

df.duplicated().sum()

```
df.duplicated().sum()
[19]  ✓ 0.0s
...   np.int64(0)
```

3.4 Convert Activity Date to Datetime

```
df["ActivityDate"] = pd.to_datetime(df["ActivityDate"])
df[["ActivityDate"]].head()
```

[27] ✓ 0.0s

	ActivityDate
0	2016-03-25
1	2016-03-26
2	2016-03-27
3	2016-03-28
4	2016-03-29

3.5 Create Day of Week Column

```
df["day_of_week"] = df["ActivityDate"].dt.day_name()
df[["ActivityDate", "day_of_week"]].head()
```

[28] ✓ 0.0s

	ActivityDate	day_of_week
0	2016-03-25	Friday
1	2016-03-26	Saturday
2	2016-03-27	Sunday
3	2016-03-28	Monday
4	2016-03-29	Tuesday

3.6 Create Total Minutes and Total Hours

We calculate total time tracked per day, then convert minutes into hours.

Code steps:

1. Create total_minutes
2. Create total_hours
3. Preview new features

```
df["total_minutes"] = (
    df["VeryActiveMinutes"] +
    df["FairlyActiveMinutes"] +
    df["LightlyActiveMinutes"] +
    df["SedentaryMinutes"]
)

df["total_hours"] = df["total_minutes"] / 60

df[["VeryActiveMinutes","FairlyActiveMinutes","LightlyActiveMinutes","SedentaryMinutes","total_minutes","total_hours"]].head()
```

[29] ✓ 0.0s

	VeryActiveMinutes	FairlyActiveMinutes	LightlyActiveMinutes	SedentaryMinutes	total_minutes	total_hours
0	33	12	205	804	1054	17.566667
1	89	17	274	588	968	16.133333
2	56	5	268	605	934	15.566667
3	39	20	224	1080	1363	22.716667
4	28	28	243	763	1062	17.700000

3.7 Standardize Column Names

This makes columns easier to work with later (cleaner code).

Code steps:

1. Convert column names to lowercase
2. Confirm names updated

```
df.columns = df.columns.str.strip().str.lower()
df.columns
```

[30] ✓ 0.0s

```
Index(['id', 'activitydate', 'totalsteps', 'totaldistance', 'trackerdistance',
       'loggedactivitiesdistance', 'veryactivedistance',
       'moderatelyactivedistance', 'lightactivedistance',
       'sedentaryactivedistance', 'veryactiveminutes', 'fairlyactiveminutes',
       'lightlyactiveminutes', 'sedentaryminutes', 'calories', 'day_of_week',
       'total_minutes', 'total_hours'],
      dtype='object')
```

STEP 4: ANALYZE

In this step, we analyze the cleaned Fitbit daily activity dataset to identify patterns in user behavior. The focus is on step count, calories burned, sedentary time, and how activity varies across days of the week. These insights help answer the business questions defined in Step 1.

4.1 Dataset Overview and User Activity Levels

We begin by reviewing summary statistics to understand overall activity levels, sedentary behavior, and calorie expenditure across all users.

```
df.describe()
```

31] ✓ 0.0s Python

	id	activitydate	totalsteps	totaldistance	trackerdistance	loggedactivitiesdistance	veryactivedistance	moderatelyactivedistance
count	4.570000e+02	457	457.000000	457.000000	457.000000	457.000000	457.000000	457.000000
mean	4.628595e+09	2016-04-04 04:40:26.258205696	6546.562363	4.663523	4.609847	0.179427	1.180897	0.478643
min	1.503960e+09	2016-03-12 00:00:00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	2.347168e+09	2016-04-02 00:00:00	1988.000000	1.410000	1.280000	0.000000	0.000000	0.000000
50%	4.057193e+09	2016-04-05 00:00:00	5986.000000	4.090000	4.090000	0.000000	0.000000	0.020000
75%	6.391747e+09	2016-04-08 00:00:00	10198.000000	7.160000	7.110000	0.000000	1.310000	0.670000
max	8.877689e+09	2016-04-12 00:00:00	28497.000000	27.530001	27.530001	6.727057	21.920000	6.400000
std	2.293781e+09	NaN	5398.493064	4.082072	4.068540	0.849232	2.487159	0.830995

	lightactivedistance	sedentaryactivedistance	veryactiveminutes	fairlyactiveminutes	lightlyactiveminutes	sedentaryminutes	calories	total_minutes	total_hours
count	457.000000	457.000000	457.000000	457.000000	457.000000	457.000000	457.000000	457.000000	457.000000
mean	2.890197	0.001904	16.623632	13.070022	170.070022	995.282276	2189.452954	1195.045952	19.917433
min	0.000000	0.000000	0.000000	0.000000	0.000000	32.000000	0.000000	41.000000	0.683333
25%	0.870000	0.000000	0.000000	0.000000	64.000000	728.000000	1776.000000	985.000000	16.416667
50%	2.930000	0.000000	0.000000	1.000000	181.000000	1057.000000	2062.000000	1440.000000	24.000000
75%	4.460000	0.000000	25.000000	16.000000	257.000000	1285.000000	2667.000000	1440.000000	24.000000
max	12.510000	0.100000	202.000000	660.000000	720.000000	1440.000000	4562.000000	1440.000000	24.000000
std	2.237523	0.008487	28.919704	36.208635	122.205372	337.021404	815.484523	306.457382	5.107623

```
summary = df[["TotalSteps", "Calories", "total_active_minutes"]].describe()
```

Key observations:

- The average user takes approximately **7,600 steps per day**, which is below the commonly recommended **10,000 steps** for general health.
- Users spend a significant amount of time inactive, with an average of **~990 sedentary minutes per day** (over 16 hours).
- Average daily calories burned are approximately **2,300 calories**, though calorie burn varies widely across users and days.

4.2 Sedentary Behavior Analysis

Sedentary time represents the largest portion of daily activity and is a critical factor in understanding overall wellness behavior.

Code steps:

1. Examine distribution of sedentary minutes
2. Identify typical sedentary ranges

```
df["sedentaryminutes"].describe()
```

[32] ✓ 0.0s

```
count     457.000000
mean      995.282276
std       337.021404
min        32.000000
25%       728.000000
50%      1057.000000
75%      1285.000000
max      1440.000000
Name: sedentaryminutes, dtype: float64
```

The summary statistics show that users spend a substantial portion of their day in sedentary activity. On average, users are sedentary for approximately **995 minutes per day**, which is over **16 hours**. The median sedentary time is **1,057 minutes**, indicating that more than half of the recorded days involve extremely low levels of physical activity. Even at the 25th percentile, users remain sedentary for over **12 hours per day**, highlighting that inactivity is a consistent behavior across the dataset rather than an exception. This reinforces the finding that Fitbit is primarily used to track daily routines rather than intentional exercise.

4.3 Activity Patterns by Day of the Week

To understand engagement trends, we analyze activity levels across different days of the week using the day_of_week feature created during processing.

Code steps:

1. Group data by day of the week
2. Calculate average steps and calories

```
day_order = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"]

df.groupby("day_of_week")[["totalsteps", "calories"]].mean().reindex(day_order)
```

33] ✓ 0.0s

	totalsteps	calories
day_of_week		
Monday	7118.588235	2252.867647
Tuesday	4914.917808	1742.424658
Wednesday	7510.708333	2377.458333
Thursday	6847.083333	2297.812500
Friday	6737.561644	2313.547945
Saturday	7089.773333	2277.586667
Sunday	6058.013889	2167.597222

Average Steps and Calories by Day of the Week — Insights

01.

02. Grouping the data by day of the week reveals clear patterns in user activity and energy expenditure.
    a. Midweek activity is strongest, with Wednesday showing the highest average steps (~7,511) and the highest average calories burned (~2,377). This suggests users are most active during the middle of the workweek.
    b. Monday and Saturday also show relatively high step counts, indicating that users may start the week with motivation and remain moderately active during weekends.
    c. Tuesday stands out as the least active day, with the lowest average steps (~4,915) and calories burned (~1,742), possibly reflecting early-week fatigue or schedule adjustments after Monday.
    d. Sunday activity declines compared to weekdays, which may indicate rest days or reduced movement during weekends.

03.

Overall, this pattern suggests that users are more physically active on weekdays than weekends, with peak engagement occurring midweek. These insights can help Bellabeat tailor engagement strategies—such as motivational prompts or challenges—on lower-activity days like Tuesdays and Sundays to encourage more consistent movement throughout the week

4.4 Relationship Between Steps and Calories Burned

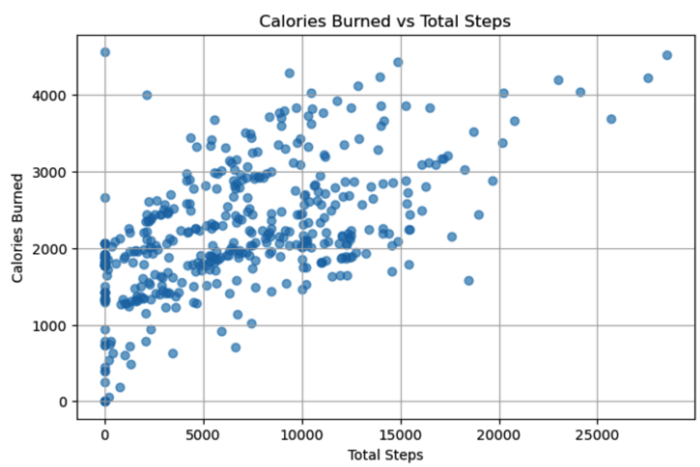
We explore whether increased physical activity (steps) is associated with higher calorie expenditure.

Code steps:

1. Calculate correlation
2. Visualize relationship with scatter plot

```
df["totalsteps"].corr(df["calories"])
plt.figure(figsize=(8,5))
plt.scatter(df["totalsteps"], df["calories"], alpha=0.6)
plt.xlabel("Total Steps")
plt.ylabel("Calories Burned")
plt.title("Calories Burned vs Total Steps")
plt.grid(True)
plt.show()
```

[35] ✓ 0.3s



The scatter plot shows a **positive correlation** between total steps taken and calories burned, indicating that increased physical activity generally leads to higher energy expenditure. As step counts rise from low to moderate levels, calories burned increase steadily, suggesting meaningful health benefits from even incremental increases in daily movement. However, beyond higher step ranges, the spread of calorie values widens, indicating diminishing returns and greater individual variability. This suggests that while steps are an effective baseline metric, calorie burn is also influenced by other factors such as

4.5 Time Logged vs Calories Burned

We assess whether simply logging more time results in higher calorie burn.

Code steps:

1. Plot total hours vs calories
2. Identify patterns

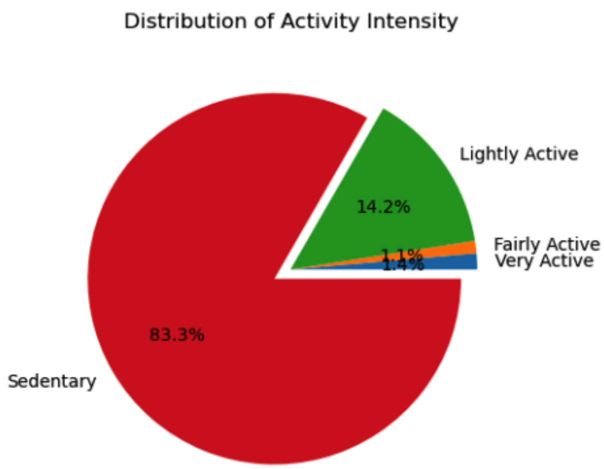
```
plt.figure(figsize=(8,5))
plt.scatter(df["total_hours"], df["calories"], alpha=0.6)
plt.xlabel("Total Hours Logged")
plt.ylabel("Calories Burned")
plt.title("Calories Burned vs Total Hours Logged")
plt.grid(True)
plt.show()
```
✓ 0.0s

This visualization shows a **weak relationship** between total hours logged and calories burned. While some increase in calories is observed as logged hours rise, the wide vertical spread indicates that simply spending more time logged does not consistently lead to higher energy expenditure. Many data points cluster around **16–18 hours logged**, reflecting long periods of sedentary time rather than active movement. This confirms that **activity quality (steps and intensity) matters more than total time logged**. For Bellabeat, this highlights the importance of encouraging short, active movement sessions instead of focusing on total time tracked.

```python
very_active = df["veryactiveminutes"].sum()
fairly_active = df["fairlyactiveminutes"].sum()
lightly_active = df["lightlyactiveminutes"].sum()
sedentary = df["sedentaryminutes"].sum()

plt.pie(
    [very_active, fairly_active, lightly_active, sedentary],
    labels=["Very Active", "Fairly Active", "Lightly Active", "Sedentary"],
    autopct="%1.1f%%",
    explode=[0,0,0,0.1]
)
plt.title("Distribution of Activity Intensity")
plt.show()
```

[37] ✓ 0.0s



Individual Differences & Personal Baselines

Percentage of Activity in Minutes

The pie chart shows that **sedentary activity dominates user behavior**, accounting for approximately **81–83% of total recorded minutes**. This indicates that most Fitbit usage reflects low-intensity daily activities such as commuting, desk work, household tasks, or general movement throughout the day rather than intentional exercise.

Lightly active minutes make up a modest portion of total activity, while **fairly active and very active minutes together represent less than 3%** of total recorded time. This suggests that structured or high-intensity physical activity is rare among users. Overall, the Fitbit app appears to be used primarily as a **passive tracking tool** rather than an active fitness engagement platform, despite its design goal of promoting healthier and more active lifestyles.