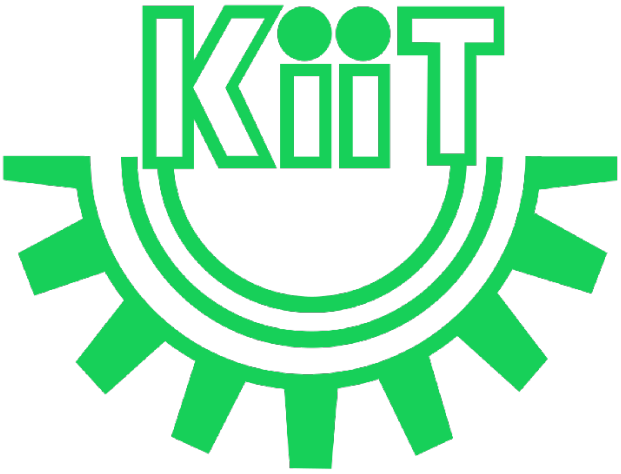
Smartwatch Data Analytics



Name: Ayan Kanti Das

Roll No: 21051386

Branch: Cse

Cse 32

Acknowledgement

I would like to express my sincere gratitude to Mrs. Sarita Tripathy for her invaluable guidance and support throughout the duration of my smartwatch data analytics project. Her expertise, dedication, and encouragement have been instrumental in shaping my understanding and approach towards this challenging field.

Mam's insightful feedback, patience, and unwavering commitment to my learning have significantly contributed to the success of this project. Her mentorship has not only enriched my technical skills but has also inspired me to strive for excellence in my academic pursuits.

I am deeply appreciative of Mrs. Sarita Tripathy's tireless efforts in fostering a stimulating learning environment and her willingness to go above and beyond to ensure my academic growth and success. Her passion for teaching and dedication to her students have left an indelible mark on my educational journey.

I extend my heartfelt thanks to mam for her unwavering support and guidance, which have been instrumental in my learning and development.

Content

- Abstract
- Introduction
- Objective
- Implementation
- Code
- Conclusion

Abstract

This abstract encapsulates a detailed exploration into the realm of smartwatch data analytics. The study delves into the vast array of data generated by smartwatches, encompassing physiological measurements, activity tracking, and other relevant metrics. Through a systematic approach, the research aims to extract meaningful insights from this data, employing various analytical techniques such as machine learning algorithms, statistical analysis, and data visualization methods. The findings shed light on patterns, trends, and correlations within the smartwatch data, offering valuable insights into individual health and lifestyle behaviours. Furthermore, the study discusses the potential applications of smartwatch data analytics in healthcare, fitness monitoring, and personalized recommendations. By leveraging the power of smartwatch data analytics, this research contributes to advancing our understanding of human behaviour, enhancing healthcare practices, and promoting well-being in society.

Introduction

In recent years, the proliferation of wearable devices, particularly smartwatches, has revolutionized the way we monitor and track various aspects of our lives, including physical activity, sleep patterns, and overall health. Smartwatches have become ubiquitous companions, offering users real-time access to a wealth of data about their well-being and daily activities. This surge in wearable technology has generated vast amounts of data, presenting both opportunities and challenges for meaningful analysis and interpretation.

The field of smartwatch data analytics has emerged as a dynamic and interdisciplinary domain, encompassing techniques from data science, machine learning, signal processing, and healthcare. The rich and diverse nature of smartwatch data presents exciting possibilities for extracting valuable insights into individual health behaviours, lifestyle patterns, and overall wellness.

This introduction sets the stage for exploring the multifaceted landscape of smartwatch data analytics. We will delve into the various types of data collected by smartwatches, ranging from heart rate and step counts

to sleep quality and stress levels. Moreover, we will examine the methodologies and tools employed to analyse and make sense of this data, including statistical techniques, machine learning algorithms, and data visualization approaches.

Furthermore, we will discuss the potential applications and implications of smartwatch data analytics across different sectors, including healthcare, fitness, and personalized recommendation systems. By harnessing the power of smartwatch data analytics, we can gain deeper insights into individual health dynamics, develop more effective interventions for chronic diseases, and promote healthier lifestyles.

Through this exploration, we aim to underscore the transformative potential of smartwatch data analytics in shaping the future of personalized health monitoring, preventive care, and well-being enhancement. This journey into the realm of smartwatch data analytics promises to unlock new avenues for innovation, research, and societal impact.

Objective

The primary objective of this smartwatch data analytics project is to explore and analyse the daily activity data collected from users wearing smartwatches. By leveraging various analytical techniques and visualization tools, we aim to gain insights into users' activity patterns, calorie expenditure, and sedentary behaviour. The project encompasses the following main themes:

- Data Exploration:

  Load the daily activity data from the provided CSV file.

  Understand the structure and characteristics of the dataset.

  Preprocess the data, handling missing values and converting date columns to appropriate formats.

- Descriptive Statistics:

Compute descriptive statistics to gain an overview of key variables such as total steps, active minutes, sedentary minutes, and calories burned.

Visualize the distribution and summary statistics of these variables using histograms, box plots, and descriptive tables.

- Exploratory Data Analysis (EDA):

Explore the relationship between different activity metrics, such as total steps, very active minutes, and calorie expenditure.

Investigate how activity levels vary across different days of the week and visualize these patterns using bar charts and pie charts.

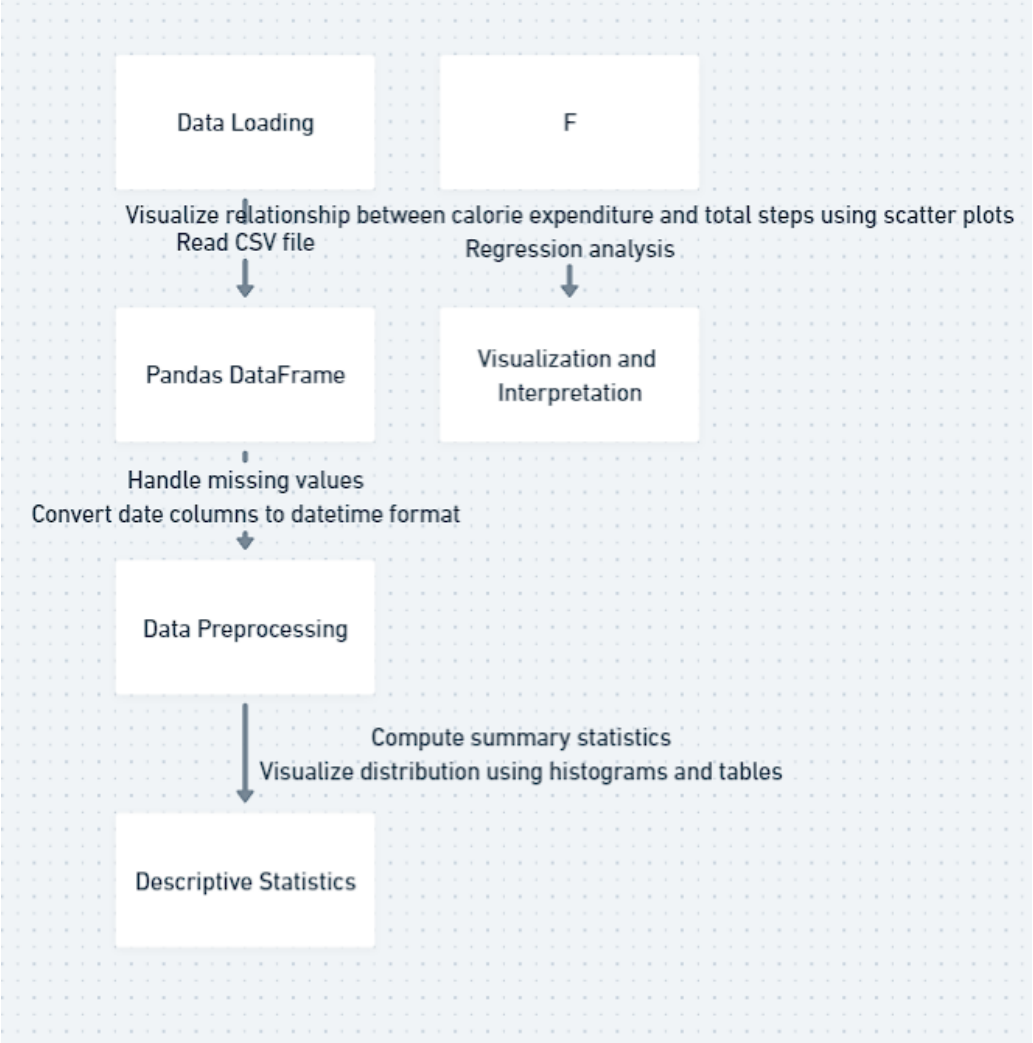
Analyse trends in sedentary behaviour and identify potential correlations with other activity metrics.

- Visualization and Interpretation:

Utilize scatter plots, regression analysis, and interactive visualizations to uncover insights into the relationship between calorie expenditure and total steps.

Visualize the distribution of active minutes across different activity levels and days of the week using bar charts and pie charts.

Interpret the findings and discuss the implications for understanding users' activity behaviours and promoting healthier lifestyles.



Implementation

1. Importing Libraries:
   Import the necessary libraries including pandas, NumPy, matplotlib.pyplot, plotly.express, and plotly.graph_objects.

2. Reading Data:
   Read the CSV file containing the data into a panda DataFrame.

3. Data Exploration:
   i. Display the first few rows of the DataFrame to **understand its structure using 'data.head()'.**
   ii. Check the shape of the DataFrame using **'data.shape'.**
   iii. Find the number of unique IDs in the dataset **using 'data['Id'].nunique()'.**
   iv. Select relevant columns for analysis and store them in a new DataFrame.

4. Handling Missing Values:
   Check for missing values using **'data.isnull().sum()'** and handle them if necessary.

5. Data Preprocessing:
   i. Convert the 'ActivityDate' column to datetime **format using 'pd.to_datetime()'.**
   ii. Create a new column 'TotalMinutes' by summing up minutes spent in different activity levels.

6. Statistical Analysis:
   Compute descriptive statistics for the dataset using **'data.describe()'.**

7. Visualization with Matplotlib and Seaborn:
   i) Create a scatter plot showing the relationship between 'Calories' and 'TotalSteps' with 'VeryActiveMinutes' as size using sns.scatterplot() and sns.regplot().
   ii) **Display the plot using 'plt.show()'.**

8. Visualization with Plotly:
   Create a pie chart showing the distribution of different activity levels using **'go.Pie()'** and **'fig.show()'.**

9. Analysing Weekly Trends:
   - Extract the day name from the 'ActivityDate' column and create a new column 'Day'.

- Create a grouped bar chart showing activity minutes for each day of the week using **'go.Bar()' and 'fig.show()'.**

10. Visualization of Inactive Minutes:

- Create a pie chart showing the distribution of sedentary minutes across different days using **'go.Pie()' and 'fig.show()'.**

11. Visualization of Daily Calories Burnt:

- Create a pie chart showing the distribution of calories burned across different days using **'go.Pie()' and 'fig.show()'.**

Code

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import plotly.express as px
import plotly.graph_objects as go
```

[2]:
```
data = pd.read_csv('dailyActivity_merged.csv')
```

[3]:
```
data.head()
```

[3]:

	Id	ActivityDate	TotalSteps	TotalDistance	TrackerDistance	LoggedActivitiesDistance	VeryActiveDistance	ModeratelyActiveDistance	LightActiveDistance	SedentaryActiveDistance	VeryActiveMinutes	FairlyActiveMinutes	LightlyActiveMinutes	SedentaryMinutes	Calories
0	1503960366	4/12/2016	13162	8.50	8.50	0.0	1.88	0.55	6.06	0.0	25	13	328	728	1985
1	1503960366	4/13/2016	10735	6.97	6.97	0.0	1.57	0.69	4.71	0.0	21	19	217	776	1797
2	1503960366	4/14/2016	10460	6.74	6.74	0.0	2.44	0.40	3.91	0.0	30	11	181	1218	1776
3	1503960366	4/15/2016	9762	6.28	6.28	0.0	2.14	1.26	2.83	0.0	29	34	209	726	1745
4	1503960366	4/16/2016	12669	8.16	8.16	0.0	2.71	0.41	5.04	0.0	36	10	221	773	1863

[4]:
```
data.shape
```

[4]:
```
(940, 15)
```

[5]:
```
data['Id'].nunique()
```

[5]:
```
33
```

[6]:
```
cols␣
↪=['Id','ActivityDate','TotalSteps','VeryActiveMinutes','FairlyActiveMinutes','LightlyActive

df = data[cols]
```

[7]:
```
df
```

[7]:

	Id	ActivityDate	TotalSteps	VeryActiveMinutes	\
0	1503960366	4/12/2016	13162	25	
1	1503960366	4/13/2016	10735	21	
2	1503960366	4/14/2016	10460	30	
3	1503960366	4/15/2016	9762	29	
4	1503960366	4/16/2016	12669	36	
..	…	…	…	…	
935	8877689391	5/8/2016	10686	17	
936	8877689391	5/9/2016	20226	73	
937	8877689391	5/10/2016	10733	18	
938	8877689391	5/11/2016	21420	88	
939	8877689391	5/12/2016	8064	23	

	FairlyActiveMinutes	LightlyActiveMinutes	SedentaryMinutes	Calories
0	13	328	728	1985
1	19	217	776	1797
2	11	181	1218	1776
3	34	209	726	1745
4	10	221	773	1863
..	…	…	…	…
935	4	245	1174	2847
936	19	217	1131	3710
937	11	224	1187	2832
938	12	213	1127	3832
939	1	137	770	1849

[940 rows x 8 columns]

[8]:
```
print(data.isnull().sum())
```

```
Id                          0
ActivityDate                0
```

```
TotalSteps                0
TotalDistance             0
TrackerDistance           0
LoggedActivitiesDistance  0
VeryActiveDistance        0
ModeratelyActiveDistance  0
LightActiveDistance       0
SedentaryActiveDistance   0
VeryActiveMinutes         0
FairlyActiveMinutes       0
LightlyActiveMinutes      0
SedentaryMinutes          0
Calories                  0
dtype: int64
```

[9]:
```
print(data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 940 entries, 0 to 939
Data columns (total 15 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Id                        940 non-null    int64
 1   ActivityDate              940 non-null    object
 2   TotalSteps                940 non-null    int64
 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    int64
 13  SedentaryMinutes          940 non-null    int64
 14  Calories                  940 non-null    int64
dtypes: float64(7), int64(7), object(1)
memory usage: 110.3+ KB
None
```

[10]:
```
data["ActivityDate"] = pd.to_datetime(data["ActivityDate"], format="%m/%d/%Y")
print(data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 940 entries, 0 to 939
Data columns (total 15 columns):
 #   Column                    Non-Null Count  Dtype
```

```
---  ------                     --------------  -----
 0   Id                         940 non-null    int64
 1   ActivityDate               940 non-null    datetime64[ns]
 2   TotalSteps                 940 non-null    int64
 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    int64
 13  SedentaryMinutes           940 non-null    int64
 14  Calories                   940 non-null    int64
dtypes: datetime64[ns](1), float64(7), int64(7)
memory usage: 110.3 KB
None
```

[11]:
```
data["TotalMinutes"] = data["VeryActiveMinutes"] + data["FairlyActiveMinutes"]␣
↪+ data["LightlyActiveMinutes"] + data["SedentaryMinutes"]
print(data["TotalMinutes"].sample(5))
```

```
151    1440
922    1440
342     984
154    1440
78     1440
Name: TotalMinutes, dtype: int64
```

[12]:
```
print(data.describe())
```

```
                 Id                    ActivityDate    TotalSteps  \
count  9.400000e+02                             940    940.000000
mean   4.855407e+09  2016-04-26 06:53:37.021276672   7637.910638
min    1.503960e+09             2016-04-12 00:00:00      0.000000
25%    2.320127e+09             2016-04-19 00:00:00   3789.750000
50%    4.445115e+09             2016-04-26 00:00:00   7405.500000
75%    6.962181e+09             2016-05-04 00:00:00  10727.000000
max    8.877689e+09             2016-05-12 00:00:00  36019.000000
std    2.424805e+09                             NaN   5087.150742

       TotalDistance  TrackerDistance  LoggedActivitiesDistance  \
count     940.000000       940.000000                940.000000
mean        5.489702         5.475351                  0.108171
min         0.000000         0.000000                  0.000000
25%         2.620000         2.620000                  0.000000
```

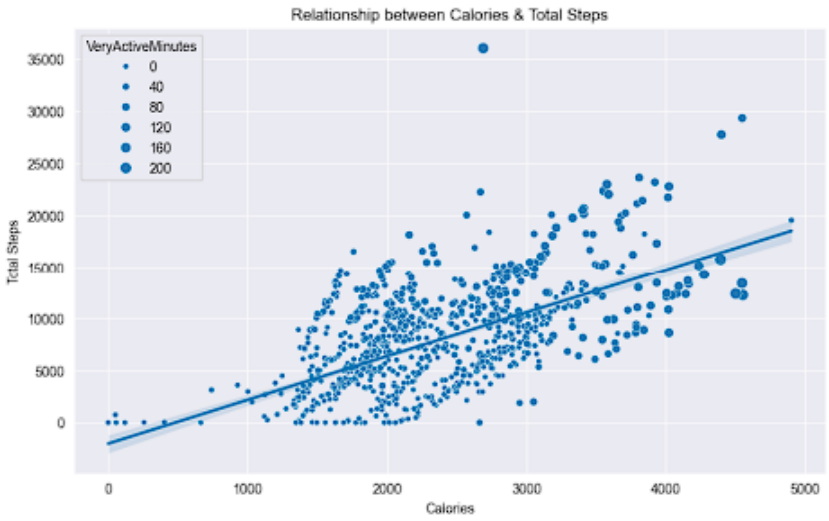
50%	5.245000	5.245000	0.000000
75%	7.712500	7.710000	0.000000
max	28.030001	28.030001	4.942142
std	3.924606	3.907276	0.619897

	VeryActiveDistance	ModeratelyActiveDistance	LightActiveDistance	\
count	940.000000	940.000000	940.000000	
mean	1.502681	0.567543	3.340819	
min	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	1.945000	
50%	0.210000	0.240000	3.365000	
75%	2.052500	0.800000	4.782500	
max	21.920000	6.480000	10.710000	
std	2.658941	0.883580	2.040655	

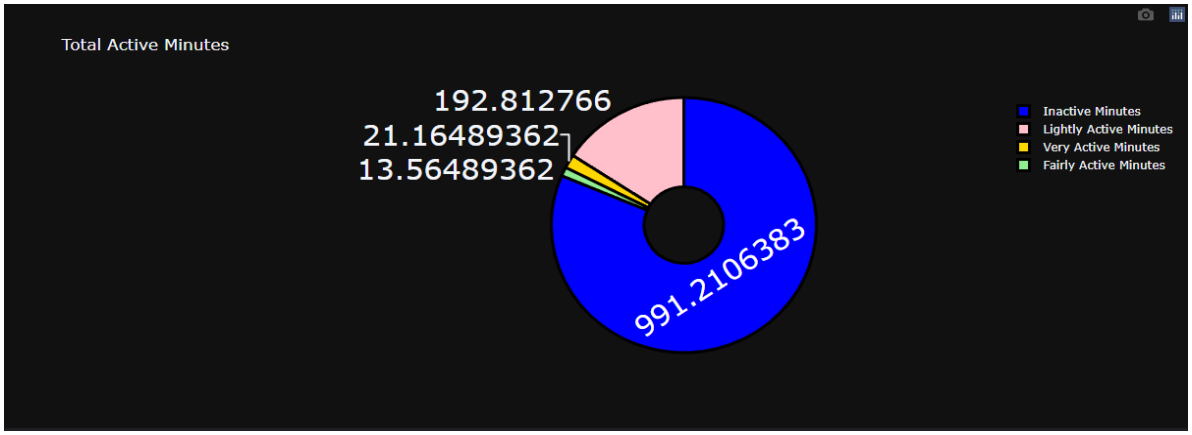
	SedentaryActiveDistance	VeryActiveMinutes	FairlyActiveMinutes	\
count	940.000000	940.000000	940.000000	
mean	0.001606	21.164894	13.564894	
min	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	0.000000	
50%	0.000000	4.000000	6.000000	
75%	0.000000	32.000000	19.000000	
max	0.110000	210.000000	143.000000	
std	0.007346	32.844803	19.987404	

	LightlyActiveMinutes	SedentaryMinutes	Calories	TotalMinutes
count	940.000000	940.000000	940.000000	940.000000
mean	192.812766	991.210638	2303.609574	1218.753191
min	0.000000	0.000000	0.000000	2.000000
25%	127.000000	729.750000	1828.500000	989.750000
50%	199.000000	1057.500000	2134.000000	1440.000000
75%	264.000000	1229.500000	2793.250000	1440.000000
max	518.000000	1440.000000	4900.000000	1440.000000
std	109.174700	301.267437	718.166862	265.931767

[13]:
```
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(10, 6))
sns.scatterplot(data=data, x="Calories", y="TotalSteps",␣
↪size="VeryActiveMinutes")
sns.regplot(data=data, x="Calories", y="TotalSteps", scatter=False)
plt.title("Relationship between Calories & Total Steps")
plt.xlabel("Calories")
plt.ylabel("Total Steps")
plt.show()
```



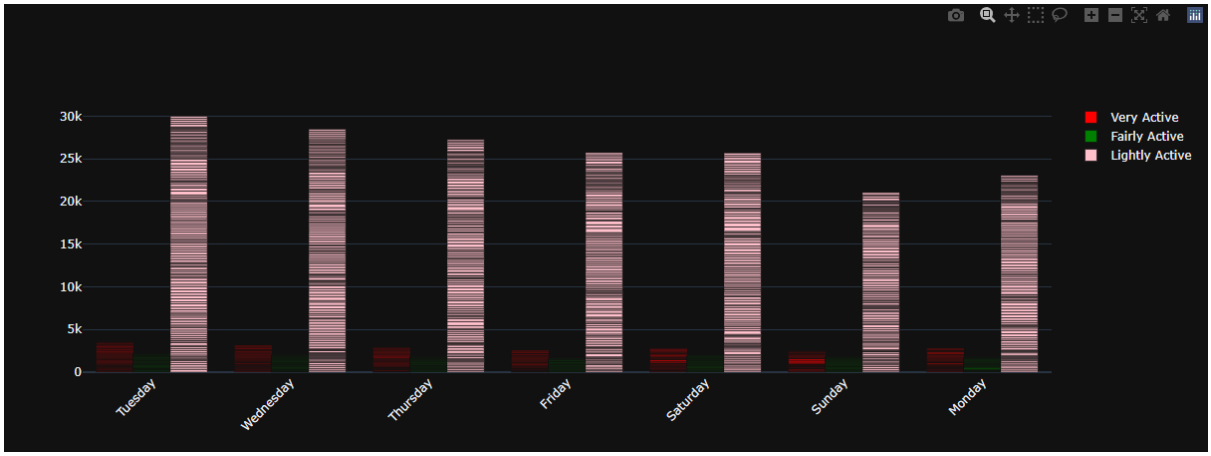
[14]:
```
import plotly.graph_objects as go

label = ["Very Active Minutes", "Fairly Active Minutes",
         "Lightly Active Minutes", "Inactive Minutes"]
counts = data[["VeryActiveMinutes", "FairlyActiveMinutes",
               "LightlyActiveMinutes", "SedentaryMinutes"]].mean()
colors = ['gold','lightgreen', "pink", "blue"]

fig = go.Figure(data=[go.Pie(labels=label, values=counts, hole=0.3)])
fig.update_layout(title_text='Total Active Minutes',␣
 ↪scene=dict(aspectmode="cube"))
fig.update_traces(hoverinfo='label+percent', textinfo='value', textfont_size=30,
                  marker=dict(colors=colors, line=dict(color='black', width=3)))
fig.show()
```



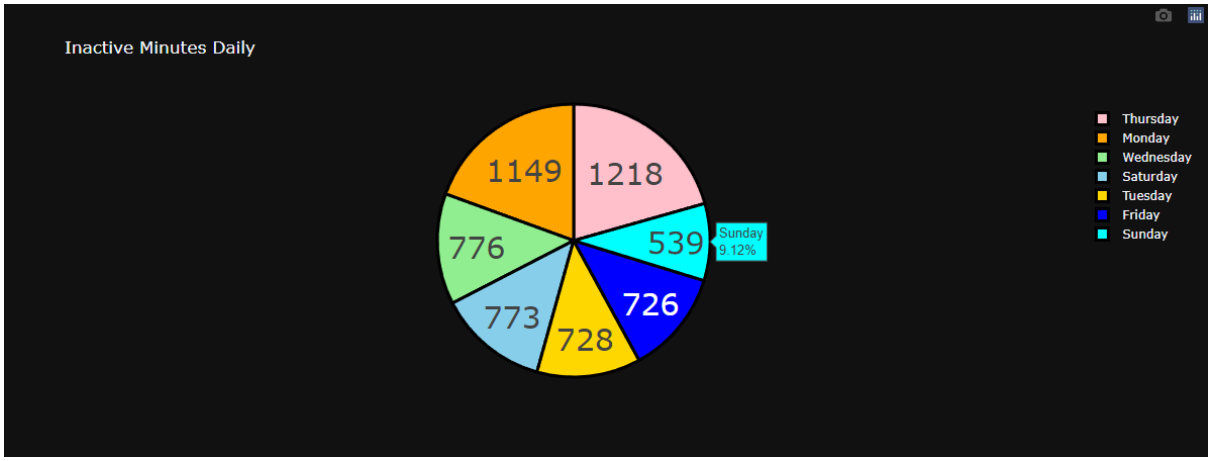
[15]:
```
data["Day"] = data["ActivityDate"].dt.day_name()
print(data["Day"].head())
```

```
0      Tuesday
1    Wednesday
2     Thursday
3       Friday
4     Saturday
Name: Day, dtype: object
```

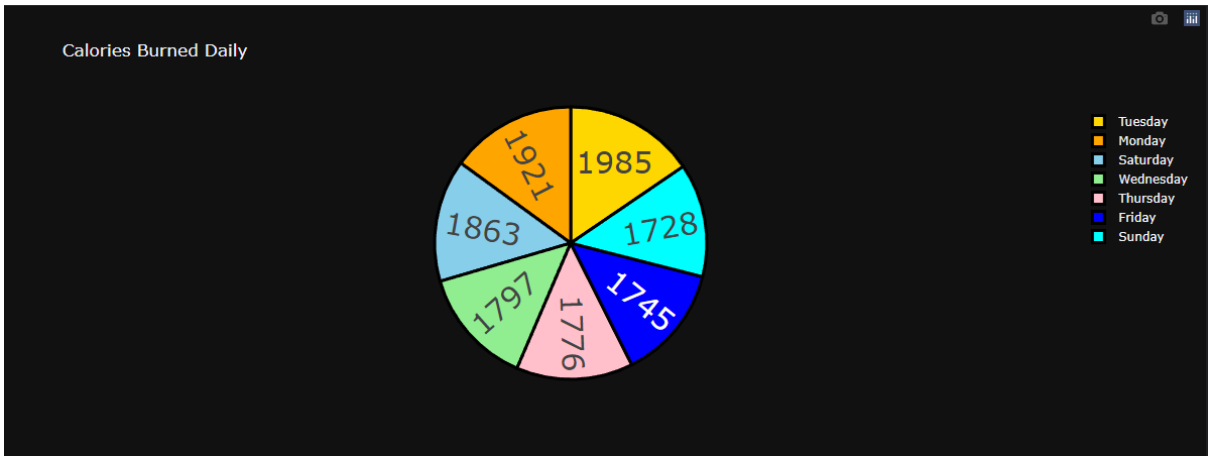
[16]:
```
fig = go.Figure()
fig.add_trace(go.Bar(
    x=data["Day"],
    y=data["VeryActiveMinutes"],
    name='Very Active',
    marker_color='red'
))
fig.add_trace(go.Bar(
    x=data["Day"],
    y=data["FairlyActiveMinutes"],
    name='Fairly Active',
    marker_color='green'
))
fig.add_trace(go.Bar(
    x=data["Day"],
    y=data["LightlyActiveMinutes"],
    name='Lightly Active',
    marker_color='pink'
))
fig.update_layout(barmode='group', xaxis_tickangle=-45)
fig.show()
```



[17]:
```
day = data["Day"].value_counts()
label = day.index
counts = data["SedentaryMinutes"]
colors = ['gold','lightgreen', "pink", "blue", "skyblue", "cyan", "orange"]

fig = go.Figure(data=[go.Pie(labels=label, values=counts)])
fig.update_layout(title_text='Inactive Minutes Daily')
fig.update_traces(hoverinfo='label+percent', textinfo='value', textfont_size=30,
                  marker=dict(colors=colors, line=dict(color='black', width=3)))
fig.show()
```



[18]:
```
calories = data["Day"].value_counts()
label = calories.index
counts = data["Calories"]
colors = ['gold','lightgreen', "pink", "blue", "skyblue", "cyan", "orange"]

fig = go.Figure(data=[go.Pie(labels=label, values=counts)])
fig.update_layout(title_text='Calories Burned Daily')
fig.update_traces(hoverinfo='label+percent', textinfo='value', textfont_size=30,
                  marker=dict(colors=colors, line=dict(color='black', width=3)))
fig.show()
```



Conclusion

The dataset, comprising daily activity records of individuals, offers a comprehensive glimpse into their physical activity levels, with metrics including steps taken, active minutes, sedentary time, and calories burned. With no missing values and appropriately formatted data, the analysis proceeds smoothly. Descriptive statistics unveil central tendencies and distributions of various activity metrics, while visualizations such as scatter plots and pie charts provide intuitive insights into relationships between activity levels, calories expended, and daily trends. Weekly patterns emerge as activity levels fluctuate across different days, depicted vividly through grouped bar charts. Moreover, pie charts showcase the distribution of sedentary time and calories burned throughout the week, shedding light on periods of inactivity and energy expenditure. Together, these analyses equip individuals with valuable insights to better understand their daily activity patterns, facilitating informed decisions aimed at enhancing overall health and fitness levels.