🧠 Digital Diet & Mental Health Impact Analysis

Modeling Screen Time, Well-being, and Mental Health Risks

📌 Overview

This project explores a curated dataset of 1,000+ individuals to analyze the relationship between **digital device usage**, **sleep patterns**, **pollution exposure**, and **mental health outcomes**. The goal is to identify behavioral patterns that correlate with psychological risks and build predictive models to support early intervention and healthier digital lifestyles.

📈 Exploratory Data Analysis (EDA)

- 📱 **Screen time patterns** across age groups and demographics
- 🌙 **Sleep quality** vs. digital usage habits
- 🌫️ **Pollution levels (PM2.5)** and their influence on well-being
- 😔 **Anxiety, depression, and stress** levels by screen time
- 🔁 **Correlations** between usage duration, social media exposure, and mental health scores
- 🔍 **Behavioral clustering** to identify at-risk groups

📚 Digital Behavior & Mental Health Insights

- 🚫 High screen time (6+ hours/day) strongly correlates with anxiety and depressive symptoms
- 🌙 Poor sleep quality mediates the relationship between digital use and mental distress
- 📉 Digital detox behavior shows positive associations with well-being
- 👨‍🎓 Young adults (16–25) report highest screen exposure and mental health concerns

---

🧠 Machine Learning Modeling

🎯 Goal 1: Predict Mental Health Risk Level

Features Used:

- Screen Time (hrs/day)
- Sleep Quality Score
- Social Media & App Usage
- PM2.5 Pollution Level
- Age, Gender, Occupation
- Self-Reported Depression, Stress Indicators

Preprocessing Steps:

- One-hot encode categorical features
- Normalize numerical variables
- Handle missing values
- Train-Test Split (80/20)

Modeling Approaches:

- Logistic Regression
- Decision Tree Classifier
- Random Forest Classifier
- Feature Importance Analysis

🎯 Goal 2 (Optional): Cluster Individuals Based on Digital Behavior

Unsupervised Models:

- K-Means Clustering
- Principal Component Analysis (PCA) for pattern discovery and visualization

---

📌 Key Insights

- 💡 Excessive digital exposure contributes to psychological distress, especially in youth
- 💡 Sleep and pollution are major mediating variables influencing mental health
- 💡 Digital detox habits associate with better mood regulation
- 💡 Clustering reveals 3–4 distinct digital behavior profiles, with one high-risk cluster

---

🔧 Tools Used

- **Python** (Pandas, NumPy, Scikit-learn)
- **Visualization:** Matplotlib, Seaborn, Plotly
- **Modeling:** Classification & Clustering
- **Environment:** Jupyter Notebook, Streamlit

📎 Dataset Info

- **Observations:** 1,000+ individuals
- **Features:**
  - Digital Behavior: Screen time, App usage, Social media frequency
  - Lifestyle: Sleep quality, Pollution (PM2.5) exposure
  - Demographics: Age, Gender, Occupation
  - Mental Health: Depression, Stress, Anxiety (self-reported)
- **Source:** Synthetic behavioral health dataset (https://www.kaggle.com/datasets/khushikyad001/impact-of-screen-time-on-mental-

health)

🧑‍💻 Author

Hilda Adina Rahmi – Aspiring Data Scientist with a passion for digital well-being, behavioral analytics, and using data for mental health innovation.

```
# 📦 Load the necessary libraries for analyzing Digital Diet & Mental Health data

# Data manipulation
import pandas as pd
import numpy as np

# Data visualization
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px

# Machine Learning & Preprocessing
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA

# Evaluation Metrics
from sklearn.metrics import accuracy_score, classification_report, confusion_matr
from sklearn.metrics import silhouette_score

# System & warnings
import warnings
warnings.filterwarnings('ignore')

# 🧠 Set visual style for behavioral analysis
sns.set(style="whitegrid")
plt.rcParams["figure.figsize"] = (10, 6)


# Load your dataset
df = pd.read_csv("digital_diet_mental_health.csv")


# Show basic info and first few rows
df_info = df.info()
df_head = df.head()

df_shape = df.shape
df_columns = df.columns.tolist()

df_shape, df_columns, df_head
```

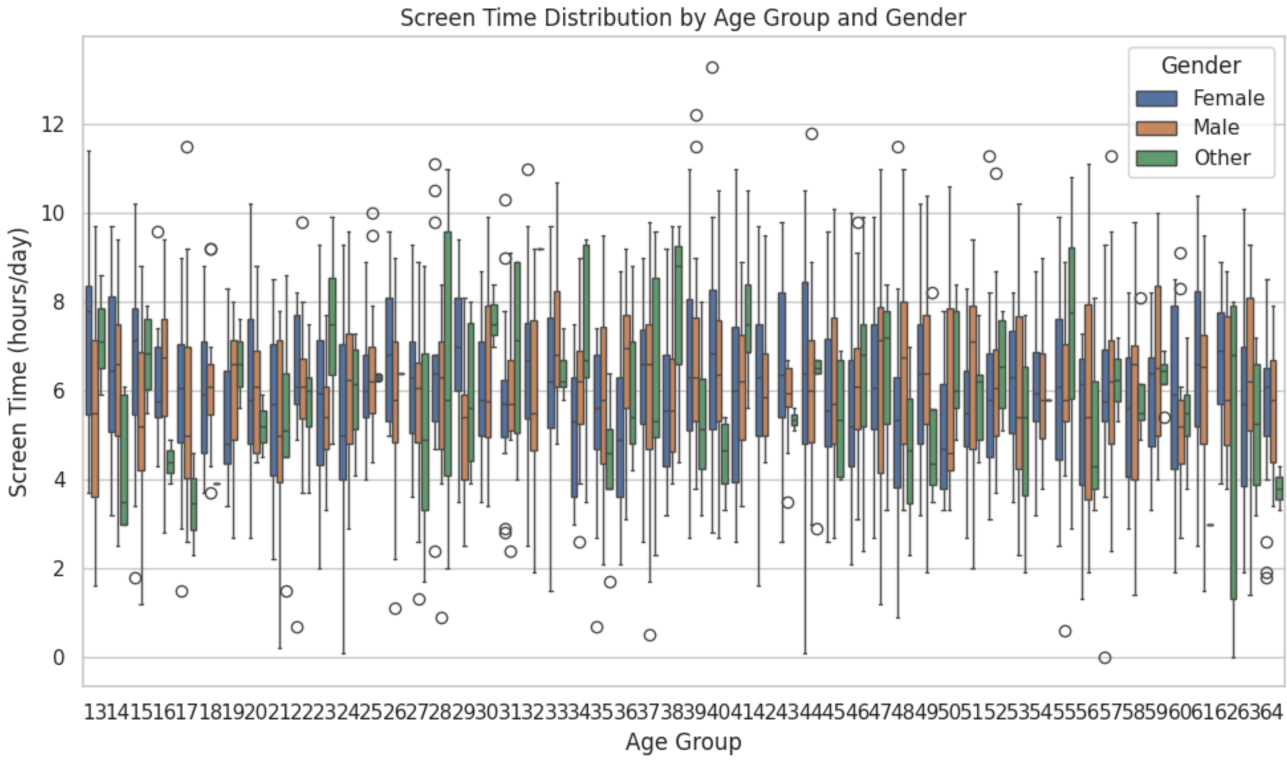
```
 'mental_health_score',
 'uses_wellness_apps',
 'eats_healthy',
 'caffeine_intake_mg_per_day',
 'weekly_anxiety_score',
 'weekly_depression_score',
 'mindfulness_minutes_per_day'],
   user_id  age  gender  daily_screen_time_hours  phone_usage_hours  \
 0  user_1   51  Female                      4.8                3.4
 1  user_2   64    Male                      3.9                3.5
 2  user_3   41   Other                     10.5                2.1
 3  user_4   27   Other                      8.8                0.0
 4  user_5   55    Male                      5.9                1.7

    laptop_usage_hours  tablet_usage_hours  tv_usage_hours
social_media_hours  \
 0                 1.3                 1.6             1.6
4.1
 1                 1.8                 0.9             2.0
2.7
 2                 2.6                 0.7             2.2
3.0
 3                 0.0                 0.7             2.5
3.3
 4                 1.1                 1.5             1.6
1.1

    work_related_hours  ...  stress_level  physical_activity_hours_per_week
\
 0                 2.0  ...            10                               0.7
 1                 3.1  ...             6                               4.3
 2                 2.8  ...             5                               3.1
 3                 1.6  ...             5                               0.0
 4                 3.6  ...             7                               3.0

    location_type  mental_health_score  uses_wellness_apps  eats_healthy  \
 0          Urban                   32                   1             1
 1       Suburban                   75                   0             1
 2       Suburban                   22                   0             0
 3          Rural                   22                   0             1
 4          Urban                   64                   1             1

    caffeine_intake_mg_per_day weekly_anxiety_score  weekly_depression_score
\
 0                       125.2                   13                       15
 1                       150.4                   19                       18
 2                       187.9                    7                        3
 3                        73.6                    7                        2
 4                       217.5                    8                       10

    mindfulness_minutes_per_day
 0                          4.0
 1                          6.5
 2                          6.9
 3                          4.8
 4                          0.0

 [5 rows x 25 columns])
```

```
# Checking for missing values and summarizing the statistics of the dataset
missing_values = df.isnull().sum()
summary_statistics = df.describe(include='all')

# Displaying the missing values and summary statistics
print(missing_values)
print(summary_statistics)
```

```
min              0.00000          0.000000
25%              5.00000          4.900000
50%             10.00000         10.400000
75%             15.00000         15.800000
max             20.00000         36.400000

[11 rows x 25 columns]
```

📈 Exploratory Data Analysis (EDA)

```
plt.figure(figsize=(10, 6))
sns.boxplot(x='age', y='daily_screen_time_hours', hue='gender', data=df)
plt.title("Screen Time Distribution by Age Group and Gender")
plt.xlabel("Age Group")
plt.ylabel("Screen Time (hours/day)")
plt.legend(title="Gender")
plt.tight_layout()
plt.show()
```



📱 How Much Time Do We Really Spend on Screens?

A Deep Dive into Age and Gender Patterns

In the age of digital everything, screen time is no longer a casual metric—it's a mirror into how we live, work, and unwind. But how does this vary across age groups and gender identities? Using data from a recent behavioral survey, we visualized the **daily screen time (in hours)** across ages 13 to 64, broken down by gender.

🎯 Key Takeaways at a Glance

- **Teenagers dominate screen time**: Across all genders, screen usage is highest in the early teenage years (13–17), peaking above **7 hours/day** on average.
- **Gradual decline with age**: There's a noticeable decline in average screen time as age increases, especially past age 30—likely due to work-life routines and reduced digital leisure time.
- **Gender differences are subtle**: Females and males show very similar usage patterns, though **females tend to have slightly higher variability**.
- **Outliers matter**: In every age group, some individuals report exceptionally high screen usage—**10 to even 13 hours/day**—highlighting potential overuse.

📊 Behind the Visualization: How to Read This

Each vertical boxplot represents the **distribution of screen time** for one age group and gender category:

- **Middle line (median)**: The typical screen time.
- **Box edges (interquartile range, IQR)**: The middle 50% of users.
- **Whiskers and circles**: Whiskers show variability; the circles are **outliers**—people with unusually high or low screen time.

> 📌 This approach is powerful because it reveals not just **averages**, but **spread, consistency, and extremes** in each group.

🧠 Why This Matters

Screen time affects **mental health**, **sleep quality**, and **daily productivity**. Understanding these patterns helps:

- 🏛 **Policymakers** develop digital wellness initiatives.
- 👪 **Parents** guide healthier screen habits for children.
- 💡 **Designers** and **developers** create more age-appropriate platforms and apps.

🔍 What's Next?

To dive deeper into the story behind the screens, we could analyze:

- The role of **occupation** or **education level** in screen time.

- Differences between **weekday vs. weekend usage**.
- Correlation between screen time and **well-being** or **academic performance**.

📢 Final Thought

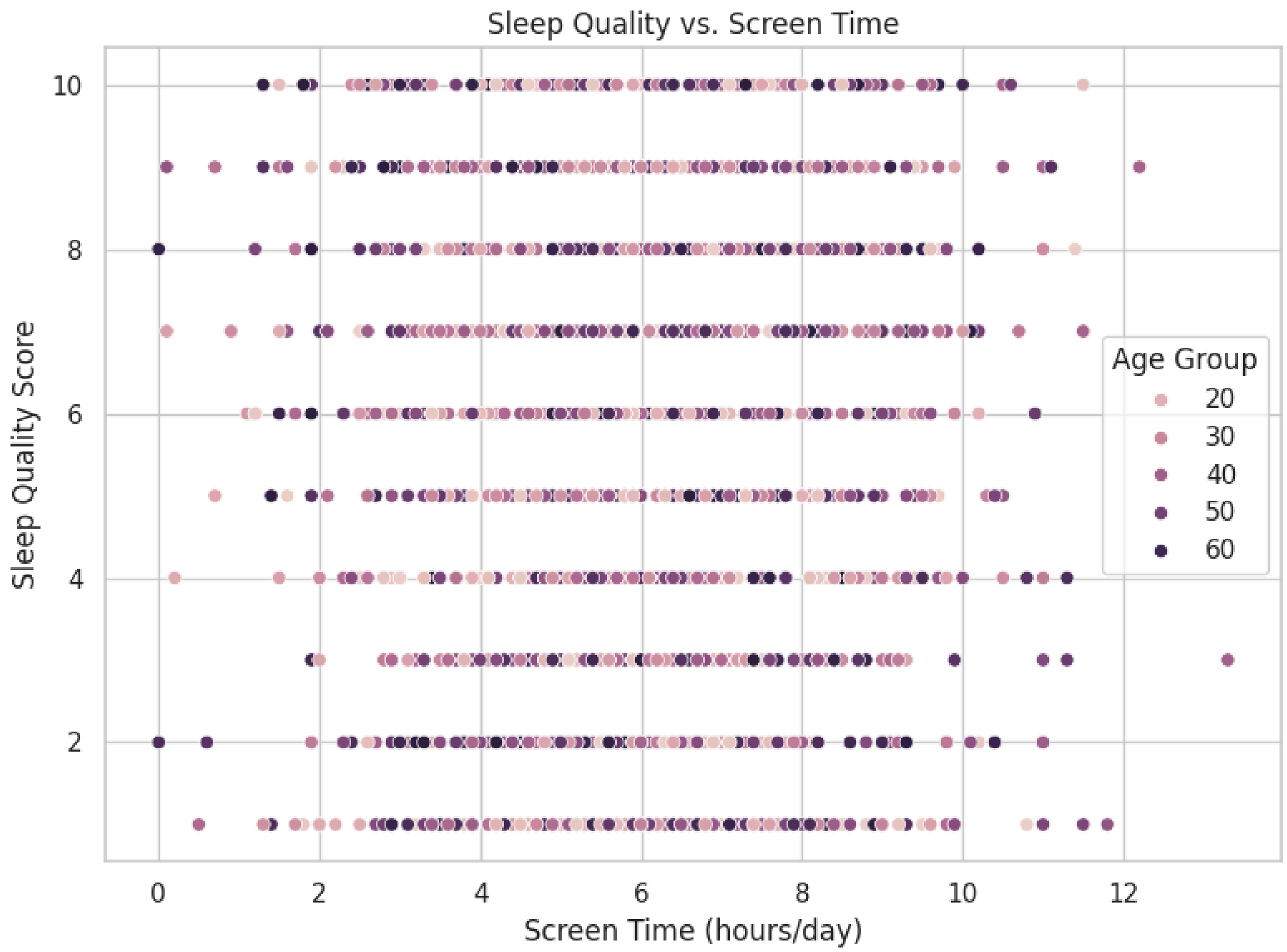
Screen time isn't inherently bad—it's all about **how**, **when**, and **why** we use our devices.

This visualization is a reminder:
🔎 Our digital habits are **age-sensitive**, **gender-influenced**, and **personally unique**.

Let's keep asking good questions, visualizing better answers, and designing a healthier digital future.

Made with 💡 using Python & Seaborn for data visualization.

```
plt.figure(figsize=(8, 6))
sns.scatterplot(x='daily_screen_time_hours', y='sleep_quality', hue='age', data=d
plt.title("Sleep Quality vs. Screen Time")
plt.xlabel("Screen Time (hours/day)")
plt.ylabel("Sleep Quality Score")
plt.legend(title='Age Group')
plt.tight_layout()
plt.show()
```



🌙 Sleep Quality vs. Screen Time

Is More Screen Time Stealing Our Sleep?

As we navigate our digitally-connected lives, a pressing question arises: *Is screen time affecting how well we sleep?* This visualization explores the relationship between **daily screen usage (hours/day)** and **sleep quality score (1–10)**, segmented by age groups from 20 to 60.

📊 What Are We Seeing Here?

Each dot represents an individual's self-reported data, plotting:

- **X-axis**: Screen time in hours per day.
- **Y-axis**: Sleep quality score (1 = poor sleep, 10 = excellent sleep).
- **Color**: Age group (20s to 60s), shown in progressively darker tones.

🔍 Insights from the Scatter Plot

1. **Sleep Quality Clusters Around the Mid-Range**

- Most people report sleep quality scores between **4 and 8**, regardless of age.
- However, **high screen time (8+ hours/day)** seems to associate more frequently with **lower sleep scores**, especially for younger age groups.

2. **Younger Adults (20s and 30s) Show Higher Screen Time**

- These groups are more represented on the far right of the plot (8–12 hours).
- Among them, poor sleep (scores ≤ 5) is more common at high screen durations.

3. **Older Adults (50s and 60s) Stay Mostly in the Safe Zone**

- These users tend to cluster below 6 hours of screen time.
- Sleep quality appears more stable, with fewer extreme lows—even at moderate screen exposure.

🧠 Why This Matters

Numerous studies link excessive screen use to disrupted **melatonin production**, **delayed sleep onset**, and **lower sleep efficiency**. This plot gives a visual confirmation that:

- 🔄 **More screen time ≠ better sleep**
- 🎯 The negative impact is more evident in **younger users**, possibly due to habits like late-night scrolling or binge-watching.
- 🛌 Better sleep outcomes are generally found among those with **moderate screen use**—especially in older demographics.

🧭 Opportunities for Further Exploration

To deepen this analysis, we could:

- Separate screen time by **type of activity** (e.g., work vs. entertainment).
- Add metrics like **bedtime consistency**, **sleep duration**, or **device type**.
- Use regression models to quantify the relationship and interaction effects.

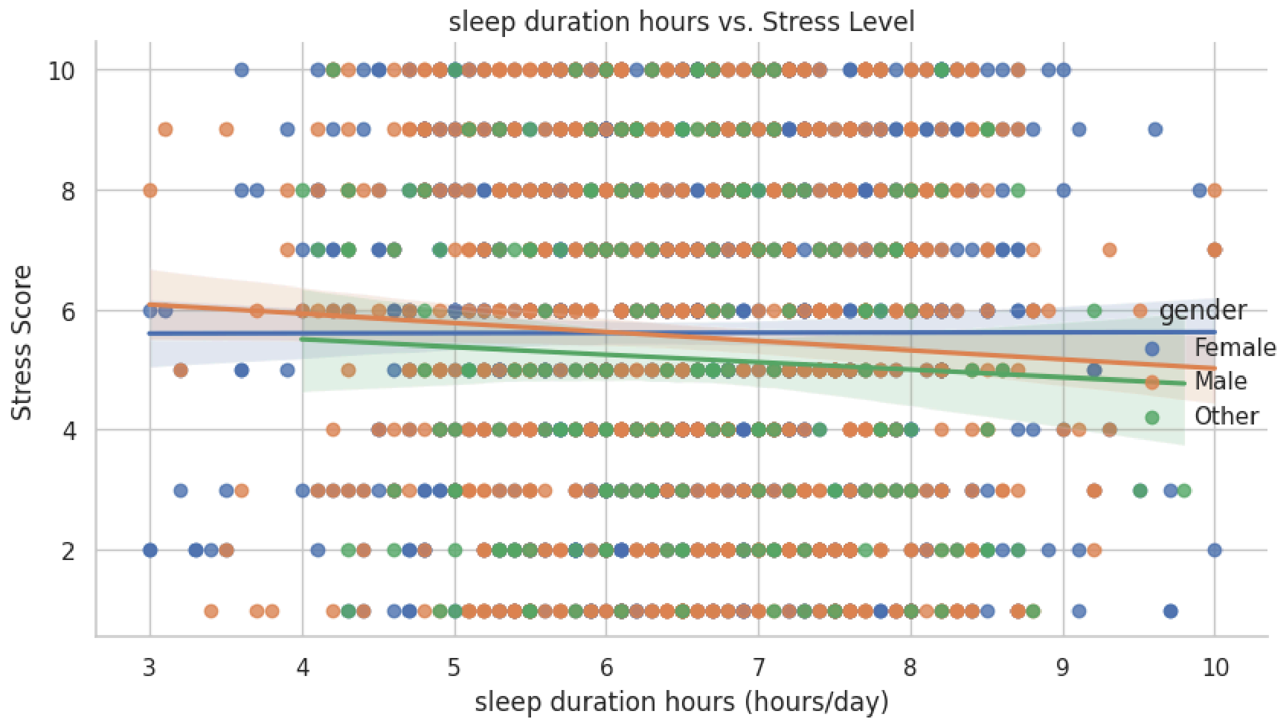
📢 Final Thought

Sleep is essential for health, and screen time is one of its silent disruptors. While screens are integral to modern life, this data suggests that **setting digital boundaries might be key to better rest**—especially for the younger generation.

Let's use data to encourage healthier tech habits and better sleep hygiene.

Visualized with Python & Seaborn — turning data into insights that matter.

```
sns.lmplot(x='sleep_duration_hours', y='stress_level', data=df, hue='gender', asp
plt.title("sleep duration hours vs. Stress Level")
plt.xlabel("sleep duration hours (hours/day)")
plt.ylabel("Stress Score")
plt.tight_layout()
plt.show()
```



😴 Sleep Duration vs. Stress Level

How Much Sleep Is Enough to Keep Stress at Bay?

Stress and sleep are two sides of the same coin—too little of one often leads to more of the other. In this visualization, we explore the relationship between **sleep duration (in hours)** and **stress levels (1–10)**, with a gender-based breakdown to see if the patterns differ across identities.

📈 What the Chart Shows

- **X-axis**: Hours of sleep per night.
- **Y-axis**: Self-reported stress score (1 = low, 10 = high).
- **Color**: Gender (Female, Male, Other).
- **Trend lines**: A regression line with a confidence interval for each gender group.

💡 Key Insights

1. **More Sleep, Less Stress**

Across all gender groups, the trend lines show a **slight downward slope**, indicating:

- Individuals who sleep **longer tend to report lower stress levels**.
- The effect is especially notable beyond the **7–8 hour mark**, where stress scores begin to drop more consistently.

2. **Gender Patterns Are Similar**

- The regression trends for Female, Male, and Other genders are **closely aligned**.
- This suggests that the **impact of sleep on stress is relatively universal**, regardless of gender identity.

3. **Clustering and Distribution**

- Most people report sleeping **between 5 to 8 hours** per night.
- Stress levels are concentrated in the **4 to 8 range**, with a few high-stress outliers (scores of 9 or 10) appearing even among those with adequate sleep.

🧠 Why This Matters

Chronic stress has major consequences: anxiety, burnout, weakened immunity, and more. Meanwhile, sleep plays a crucial role in stress regulation by:

- **Restoring brain function**

- **Balancing hormones**
- **Improving emotional regulation**

This chart visually supports what science has long suggested: **prioritizing sleep is one of the most natural stress management tools** available.

🔍 Ideas for Further Analysis

To enrich the story, future steps could include:

- Analyzing by **occupation** or **student status**.
- Adding **screen time** as a third variable.
- Investigating **sleep quality**, not just duration.

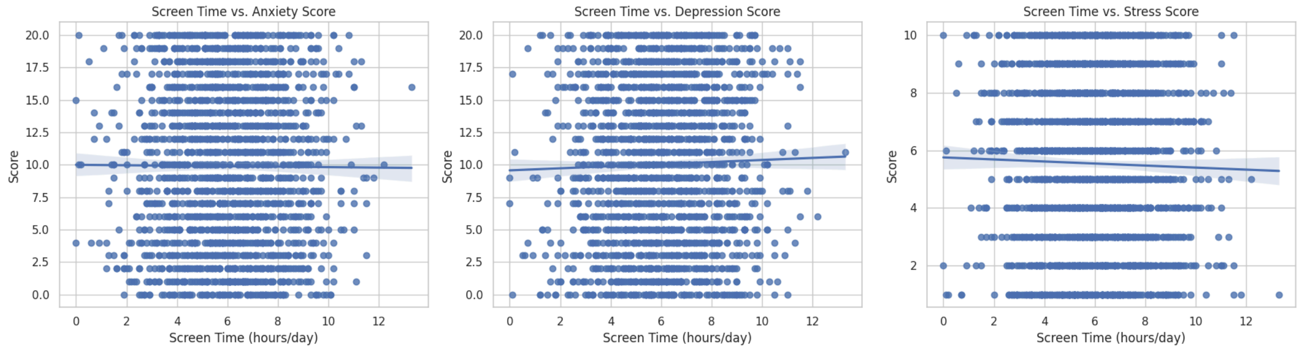
📢 Final Thought

We live in a culture that often sacrifices sleep in the name of productivity. But this data sends a gentle reminder:
🛌 **Sleep is self-care.**
😌 **Less stress starts with more rest.**

Let's give ourselves permission to rest better and stress less—starting tonight.

Visualization created using Python's Seaborn library. Where data meets insight.

```
fig, axes = plt.subplots(1, 3, figsize=(18, 5))

sns.regplot(ax=axes[0], x='daily_screen_time_hours', y='weekly_anxiety_score', da
axes[0].set_title('Screen Time vs. Anxiety Score')

sns.regplot(ax=axes[1], x='daily_screen_time_hours', y='weekly_depression_score',
axes[1].set_title('Screen Time vs. Depression Score')

sns.regplot(ax=axes[2], x='daily_screen_time_hours', y='stress_level', data=df)
axes[2].set_title('Screen Time vs. Stress Score')

for ax in axes:
    ax.set_xlabel('Screen Time (hours/day)')
    ax.set_ylabel('Score')

plt.tight_layout()
plt.show()
```



📱🧠 How Screen Time and Sleep Impact Mental Health: A Data-Driven Insight

Technology is a double-edged sword. It connects, entertains, and educates us—but is it silently shaping our mental well-being? Let's explore this question through a series of visual analytics based on a dataset involving **screen time, sleep, and mental health scores** across different demographics.

📱😰 Screen Time and Mental Health: The Trio of Anxiety, Depression, and Stress

Panel 1: Screen Time vs. Anxiety

- Slight **positive correlation**: More screen time → higher anxiety.
- Scores are **highly variable**, suggesting individual differences.

Panel 2: Screen Time vs. Depression

- Another **positive correlation**: Screen time aligns with **higher depression scores**.
- Strong clustering between **5–9 hours/day**.

Panel 3: Screen Time vs. Stress

- The **trend is mild**, but there's still an **increase in stress** with screen time.
- Stress scores remain **consistently elevated**, even at low screen times.

📘 *Interpretation:* Excessive screen time appears to affect **anxiety and depression** more than stress. However, screen exposure could still be a contributor in high-stress environments.

🧩 Conclusion

✅ What We Learned:

- **Youth and screen time** are closely linked, but mental health risks grow with duration.

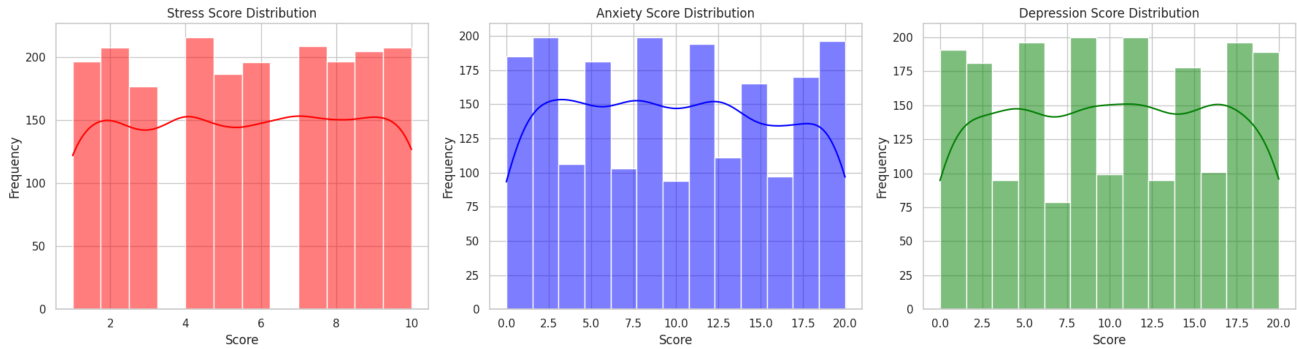
- **Sleep has a strong inverse relationship** with stress—more sleep, less stress.
- **Screen time** shows a **positive relationship with anxiety and depression** scores.

🌱 Takeaway for Action

- 📵 Practice **digital hygiene** (e.g., screen-free hours or breaks).
- 😴 **Prioritize quality sleep**—it's a critical buffer against stress.
- 🧘 **Monitor and support mental health**, especially for high-screen-time individuals.

Visualizations created using Python (Seaborn, Matplotlib). Dataset available on request.

```
fig, axes = plt.subplots(1, 3, figsize=(18, 5))

sns.histplot(df['stress_level'], kde=True, ax=axes[0], color='red')
axes[0].set_title("Stress Score Distribution")

sns.histplot(df['weekly_anxiety_score'], kde=True, ax=axes[1], color='blue')
axes[1].set_title("Anxiety Score Distribution")

sns.histplot(df['weekly_depression_score'], kde=True, ax=axes[2], color='green')
axes[2].set_title("Depression Score Distribution")

for ax in axes:
    ax.set_xlabel("Score")
    ax.set_ylabel("Frequency")

plt.tight_layout()
plt.show()
```



📈 Distribution of Mental Health Scores

🔍 Breakdown by Dimension:

🧠 Stress Scores:

- Fairly **evenly distributed** across the 1–10 scale.
- Slightly more frequent scores around **4 to 8**.
- The smooth density line suggests **no extreme skewness**.

😰 Anxiety Scores:

- Distribution is **somewhat uniform** with **spikes at 2, 8, and 19**.
- This indicates potential **clusters of participants** experiencing either **very low or very high anxiety**.
- Possibly **bimodal**, with a mild drop in middle scores.

😞 Depression Scores:

- Similar pattern to anxiety, with **high frequencies at the extremes (0 and 20)**.
- Suggests **polarized mental states**—a portion of participants report **severe depression**, while others report **none**.

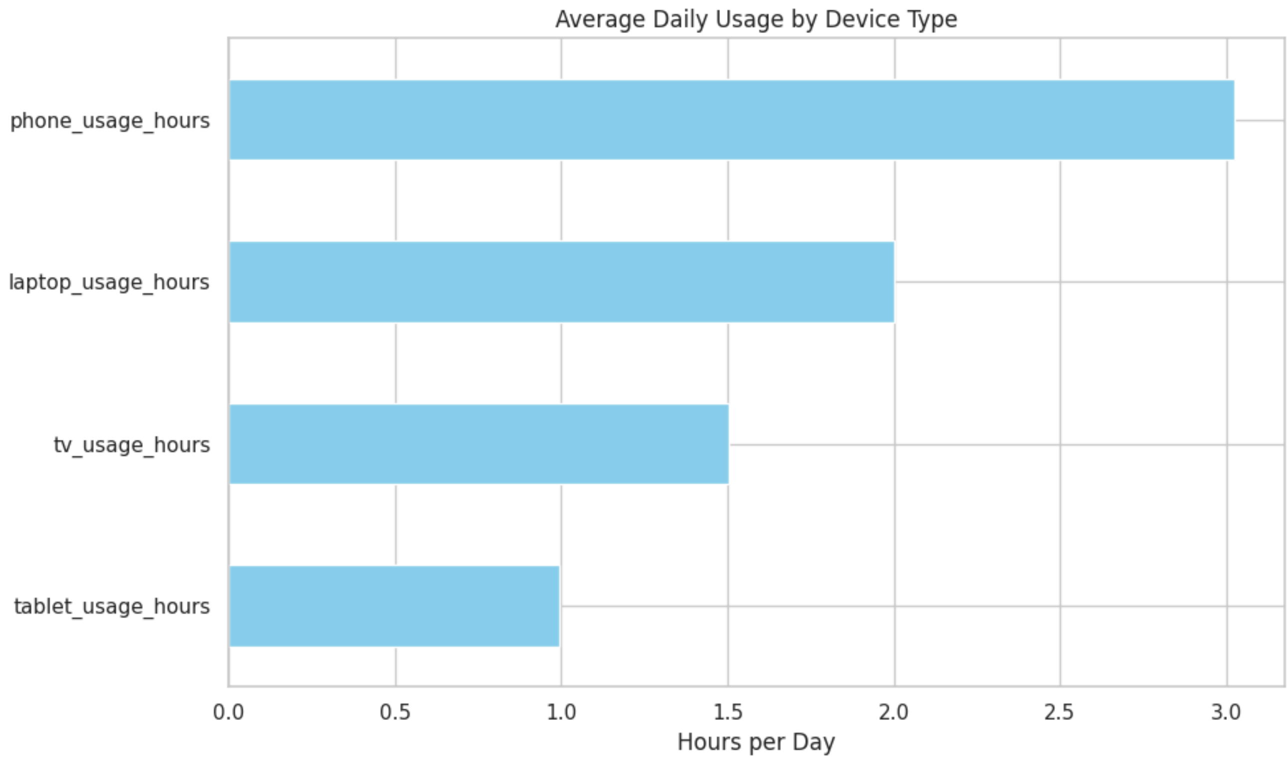
📘 *Interpretation:* The distributions reinforce earlier findings:

- Mental health scores aren't bell-curved (normal) but show **flat or bimodal patterns**, indicating **divergent experiences** within the population.
- These findings may point to **specific subgroups** (e.g., age, sleep patterns, or screen time levels) that are more vulnerable.

💡 Final Thoughts

Understanding the **distribution** helps contextualize **regression relationships**. If scores were tightly clustered, it might suggest universal experience—but here, we see **diverse mental health profiles**, calling for **personalized digital wellness strategies**.

```
device_columns = ['phone_usage_hours', 'laptop_usage_hours', 'tablet_usage_hours'

df[device_columns].mean().sort_values().plot(kind='barh', color='skyblue')
plt.title("Average Daily Usage by Device Type")
plt.xlabel("Hours per Day")
plt.tight_layout()
plt.show()
```



📱📊 How We Spend Our Screen Time: A Deep Dive into Daily Device Usage

In today's digital era, understanding how people interact with technology is more than just interesting—it's essential. Our analysis aimed to uncover **how much time individuals spend daily on different digital devices**. The result? A clear picture of modern screen habits.

🥇 Phones: The Undisputed Champion

At the top of the usage chart is the **smartphone**, averaging **just over 3 hours per day**. This result isn't surprising—phones are pocket-sized windows to the world. They are used for everything: messaging, social media, video streaming, navigation, and even remote work. The convenience and versatility of smartphones make them the go-to device for almost every task.

> 🔍 *Insight*: This heavy usage highlights the importance of **mobile-first strategies** for businesses and content creators. If you're not optimized for mobile, you're missing out on key engagement.

💻 Laptops: The Productivity Backbone

Coming in second, **laptop usage clocks in at around 2 hours daily**. Laptops are essential for work, learning, and multitasking. Unlike phones, they support more complex tasks like coding, data analysis, content creation, and document processing.

> 🧠 *Educational Takeaway*: While smartphones are versatile, laptops remain the **core tool for productivity** and long-form engagement. Investing in digital literacy—like using spreadsheets, data tools, or programming—can significantly improve productivity.

📺 TVs: Still Relevant, but Changing

With an average of **1.5 hours per day**, **TV usage** remains part of daily life. However, its decline compared to phones and laptops may signal a shift from traditional scheduled viewing to **on-demand, personalized content** on portable screens.

> 📉 *Trend Note*: This drop reflects a generational shift. Younger audiences prefer streaming on phones or laptops, impacting advertising, broadcasting, and entertainment industries.

💊 Tablets: The Niche Companion

Tablets come last, with **1 hour of usage per day**. They strike a balance between mobility and screen size but haven't fully replaced phones or laptops. They're often used by students, children, or for specific tasks like reading, drawing, or lightweight browsing.

> 🎯 *Design Insight*: Tablets serve niche use cases. Developers and educators targeting this segment should prioritize **intuitive UX and age-friendly content**.

📈 Conclusion: The Screen-Time Hierarchy

This data paints a clear picture:

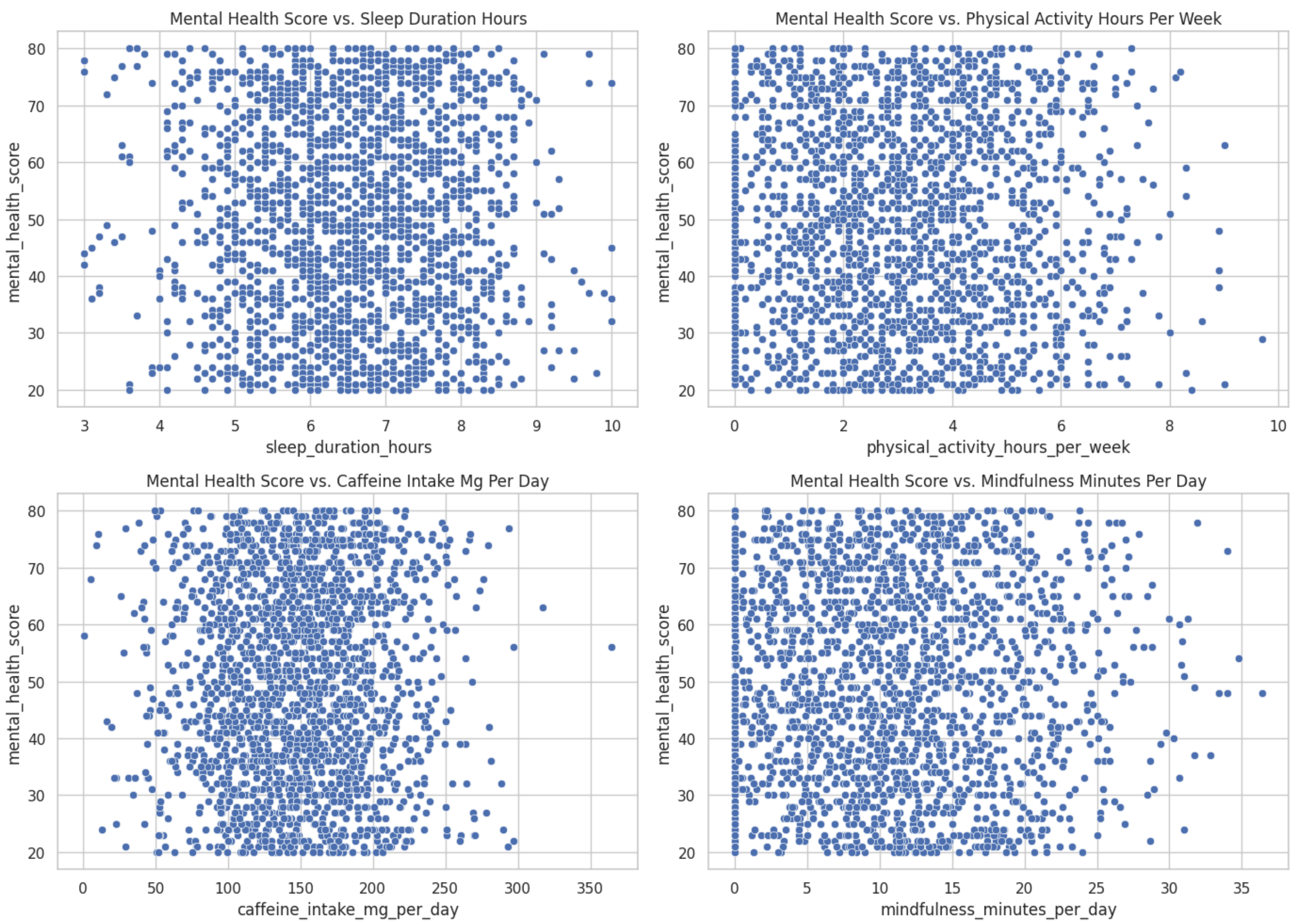
- **Mobile-first is no longer optional—it's fundamental.**
- **Laptop use reflects the continued importance of digital productivity.**
- **TVs are being disrupted by more interactive screens.**
- **Tablets remain a secondary, task-specific tool.**

💡 Final Thought

Understanding device usage helps guide **smarter design, marketing, and digital well-being** efforts. Whether you're a developer, business owner, educator, or parent—these insights can help shape more effective strategies in a screen-driven world.

```
lifestyle_features = ['sleep_duration_hours', 'physical_activity_hours_per_week
                      'caffeine_intake_mg_per_day', 'mindfulness_minutes_per_da

fig, axes = plt.subplots(2, 2, figsize=(14, 10))
```

```
for idx, feature in enumerate(lifestyle_features):
    sns.scatterplot(ax=axes[idx//2, idx%2], x=feature, y='mental_health_score',
    axes[idx//2, idx%2].set_title(f"Mental Health Score vs. {feature.replace('_

plt.tight_layout()
plt.show()
```



🧠💡 Understanding Mental Health Through Daily Habits: What the Data Shows

Mental health is influenced by many lifestyle factors—but which ones matter the most? Using real-world data, we explored how sleep, physical activity, caffeine, and mindfulness habits relate to **mental health scores**.

Let's unpack what the data tells us.

😴 Mental Health vs. Sleep Duration

Plot: *Mental Health Score vs. Sleep Duration (Hours)*

- Most individuals sleep between **5 to 8 hours** per night.
- Mental health scores cluster around the **50–70** range.

- No strong trend, but **extremely short (<5h)** or **long sleep (>9h)** appears less common and may relate to lower scores.

> 🧠 *Insight*: Sleep is crucial—but it's not just about duration. Quality, consistency, and sleep disorders might explain the scatter.

🏃 Mental Health vs. Physical Activity

Plot: *Mental Health Score vs. Physical Activity Hours per Week*

- Most participants reported **0 to 4 hours** of physical activity.
- A subtle concentration of **higher mental health scores** appears among those engaging in **moderate activity** (2–6 hours/week).

> 💡 *Takeaway*: Regular physical activity may have a **positive impact**, but the benefits may plateau after a few hours or depend on activity type.

☕ Mental Health vs. Caffeine Intake

Plot: *Mental Health Score vs. Caffeine Intake (mg/day)*

- Majority consume **50 to 200 mg/day** (about 1–2 cups of coffee).
- Mental health scores remain broadly distributed, with no clear pattern.

> ⚠️ *Observation*: Caffeine intake does **not show a strong correlation** with mental health in this dataset. Individual sensitivity, timing, and sleep interaction might matter more.

🧘 Mental Health vs. Mindfulness Practice

Plot: *Mental Health Score vs. Mindfulness Minutes per Day*

- Most respondents practice **0–10 minutes** of mindfulness.
- Slight positive trend: those who practice **15+ minutes daily** tend to have **higher mental health scores**.

> 🌱 *Educational Note*: Mindfulness—even short daily sessions—could **contribute positively** to mental well-being. This aligns with research on meditation and stress reduction.

🔍 Overall Summary

Factor	Impact on Mental Health	Notes
Sleep Duration	Moderate	No clear trend, but <5h may be riskier
Physical Activity	Slightly Positive	Benefits peak around 3–6 hrs/week

Factor	Impact on Mental Health	Notes
Caffeine Intake	Neutral	No strong link found
Mindfulness Practice	Slightly Positive	Longer daily practice shows promise

---

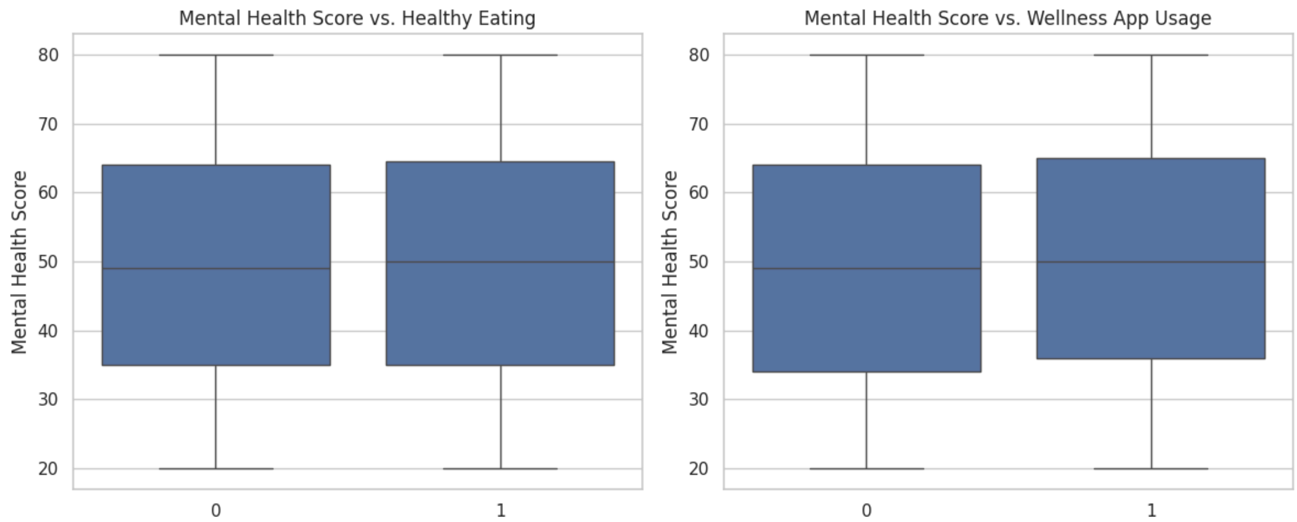
💬 Final Thought

While no single habit guarantees perfect mental health, this data suggests that **balanced sleep**, **regular activity**, and **mindfulness practice** are good investments in your well-being. Remember, mental health is multi-dimensional—these plots are snapshots, not prescriptions.

> ✨ *Encourage consistent, small changes. The data supports their value.*

Would you like this turned into a visual infographic or blog-style article?

```
fig, axes = plt.subplots(1, 2, figsize=(12, 5))

sns.boxplot(x='eats_healthy', y='mental_health_score', data=df, ax=axes[0])
axes[0].set_title("Mental Health Score vs. Healthy Eating")

sns.boxplot(x='uses_wellness_apps', y='mental_health_score', data=df, ax=axes[1])
axes[1].set_title("Mental Health Score vs. Wellness App Usage")

for ax in axes:
    ax.set_xlabel("")
    ax.set_ylabel("Mental Health Score")

plt.tight_layout()
plt.show()
```



🥗📱 Healthy Habits and Mental Health: What the Box Plots Reveal

Can lifestyle choices like eating well and using wellness apps really influence our mental health? We explored this question using data visualized through two box plots.

Each plot compares **Mental Health Scores** against two binary lifestyle behaviors:

- **Healthy Eating (0 = No, 1 = Yes)**
- **Wellness App Usage (0 = No, 1 = Yes)**

🥦 Mental Health and Healthy Eating

Plot: *Mental Health Score vs. Healthy Eating*

- The median mental health scores for both non-healthy and healthy eaters are **virtually identical**.
- The range, interquartile spread, and overall distribution are also very similar.

> 🧠 *Insight*: Surprisingly, **self-reported healthy eating does not show a significant difference** in mental health scores. This may be due to:
>
> - Subjective definitions of "healthy eating"
> - Influence of other stronger variables (e.g. sleep, stress, genetics)

> 🧪 *Takeaway*: Diet might play a **supportive but not standalone** role in mental health, especially if not tracked rigorously.

📱 Mental Health and Wellness App Usage

Plot: *Mental Health Score vs. Wellness App Usage*

- Again, the **medians and spreads** of both groups are almost the same.
- Users of wellness apps do **not report noticeably higher mental health scores** compared to non-users.

> 💡 *Observation*: Simply using wellness apps may not lead to better mental health outcomes. It's **how consistently and meaningfully** they are used that matters.

> ⚠️ *Critical Note*: Passive usage, app fatigue, or unengaging interfaces might limit benefits.

📊 Summary Table

Behavior	Clear Impact on Mental Health?	Key Insight
Healthy Eating	❌ Minimal difference	May require deeper tracking (e.g., nutritional quality, meal timing)
Wellness App Usage	❌ No strong effect	App engagement, not just usage, likely matters

---

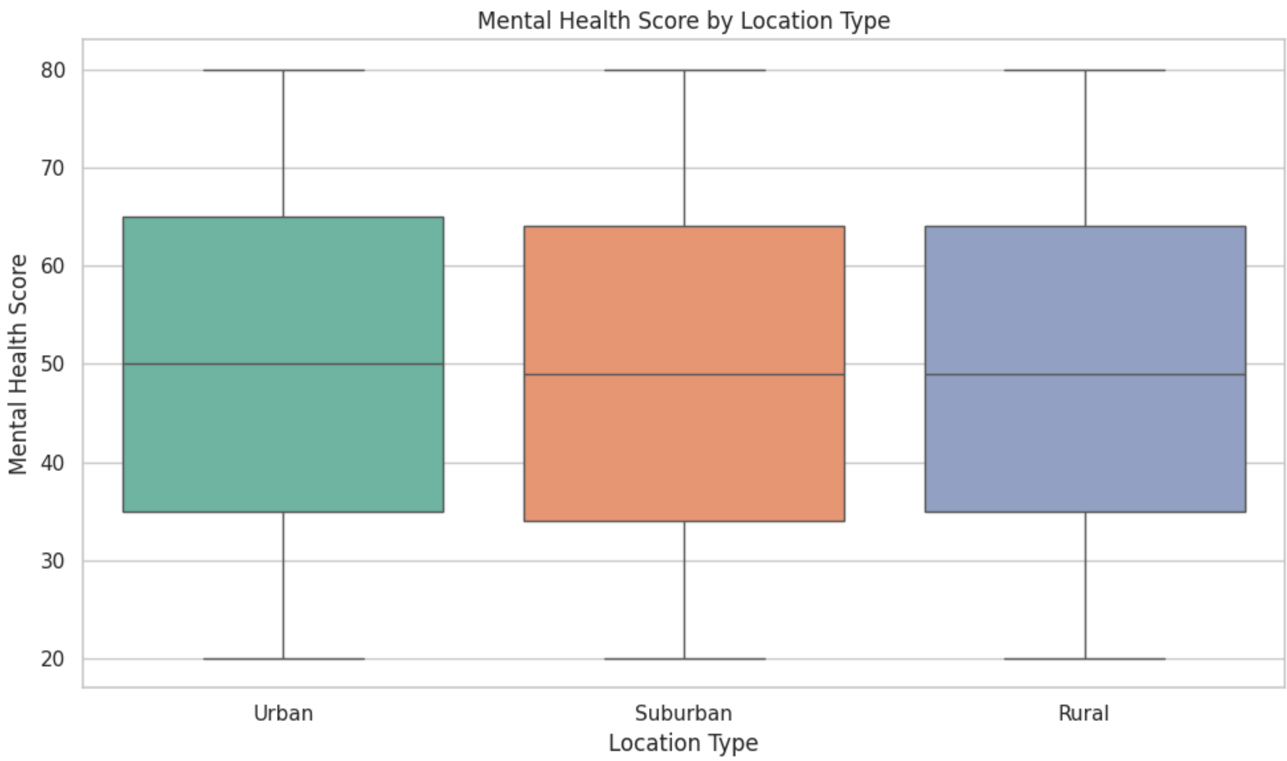
💬 Final Thought

The box plots challenge common assumptions: **not all healthy behaviors show an immediate or measurable effect** on mental well-being—at least not in isolation.

> 🧩 *Mental health is multifactorial.* Lifestyle tools are helpful, but only as part of a **holistic and sustained approach**.

Would you like a combined summary of all your plots for a comprehensive report?

```
sns.boxplot(x='location_type', y='mental_health_score', data=df, palette='Set2')
plt.title("Mental Health Score by Location Type")
plt.xlabel("Location Type")
plt.ylabel("Mental Health Score")
plt.tight_layout()
plt.show()
```



🏙️🏡🌄 Does Where You Live Influence Your Mental Health?

Urban, suburban, or rural—does your living environment shape your mental well-being? This box plot explores the **relationship between location type and mental health scores**.

📊 Mental Health Score by Location Type

- **Median scores** for Urban, Suburban, and Rural residents are nearly **identical**, hovering just under **50**.

- The **spread (IQR)** and **range** (min to max) are also **remarkably similar** across all three groups.
- Each group includes individuals scoring from **very low (20s)** to **very high (80s)** on the mental health scale.

> 🧠 *Insight*: Location type alone does **not appear to strongly influence** mental health outcomes in this dataset.

🏡 Why the Similarity?

Despite the assumption that rural areas may be more peaceful, or cities more stressful, this data suggests:

- **Personal, social, and economic factors** may weigh more heavily than geography.
- Urban and rural stressors may balance out (e.g., traffic vs. isolation, fast pace vs. limited access to care).
- Mental health support systems (e.g., therapy, community programs) may vary in **quality and accessibility**, but not necessarily by location.

🔎 Summary Table

Location Type	Median Mental Health Score	Observations
Urban	~50	Wide variability
Suburban	~49	Slightly lower median
Rural	~49	Similar distribution

🌐 Final Thought

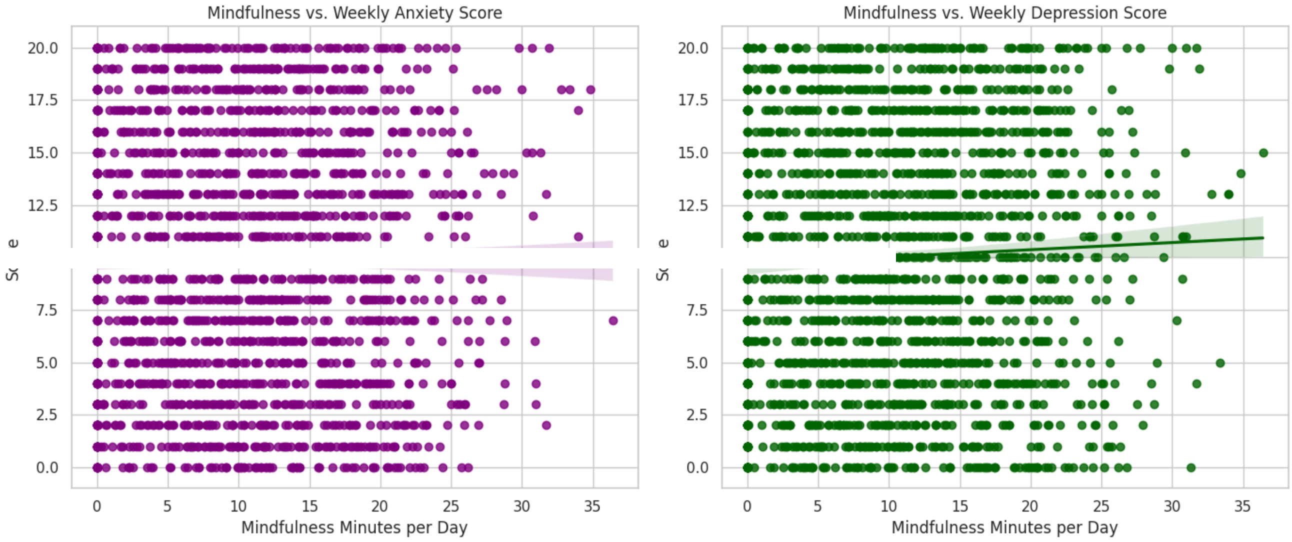
This plot reminds us that **mental health is universal**—struggles and strengths exist **regardless of zip code**. Location matters, but it's just one piece of a much larger mental wellness puzzle.

> 💬 *Actionable Message*: Instead of relocating for peace of mind, consider improving support systems, community connection, and self-care wherever you are.

Would you like to combine this with the previous lifestyle plots to create a comprehensive story or report?

```
fig, axes = plt.subplots(1, 2, figsize=(14, 6))

sns.regplot(x='mindfulness_minutes_per_day', y='weekly_anxiety_score', data=df, a
axes[0].set_title("Mindfulness vs. Weekly Anxiety Score")

sns.regplot(x='mindfulness_minutes_per_day', y='weekly_depression_score', data=df
axes[1].set_title("Mindfulness vs. Weekly Depression Score")
```

```
for ax in axes:
    ax.set_xlabel("Mindfulness Minutes per Day")
    ax.set_ylabel("Score")

plt.tight_layout()
plt.show()
```



🧘 Can Mindfulness Reduce Anxiety and Depression?

Mindfulness is often promoted as a remedy for stress, anxiety, and depression. But what does the data actually show?

Below, we explore how **daily mindfulness minutes** relate to two critical mental health outcomes: **weekly anxiety** and **depression scores**.

📉 Mindfulness vs. Anxiety Score

- The **left chart** shows **no clear downward trend** between mindfulness practice and anxiety scores.
- In fact, the regression line is nearly flat, suggesting **minimal or no effect**.
- Some individuals who practiced mindfulness still reported **high anxiety levels**, and vice versa.

> 🧠 *Takeaway*: Simply practicing mindfulness doesn't guarantee lower anxiety. The **quality** and **context** of mindfulness practices might matter more than just the number of minutes.

📈 Mindfulness vs. Depression Score

- The **right chart** reveals a **slight upward slope**, indicating that more mindfulness may actually correlate with **slightly higher depression scores**.
- This might seem counterintuitive, but could reflect **reverse causality**—individuals with depression may **turn to mindfulness** in search of relief.

> 🔎 *Insight*: Mindfulness could be more of a **coping strategy** rather than a cure. Its presence may signal that individuals are already dealing with mental health challenges.

🧩 What Could Be Happening?

Observation	Possible Explanation
No strong link between mindfulness and anxiety	Mindfulness may be more effective when guided or structured
Slight increase in depression scores with more mindfulness	People experiencing depression might be more likely to adopt min
Wide scatter in both plots	Mental health is complex and influenced by many lifestyle, genetic

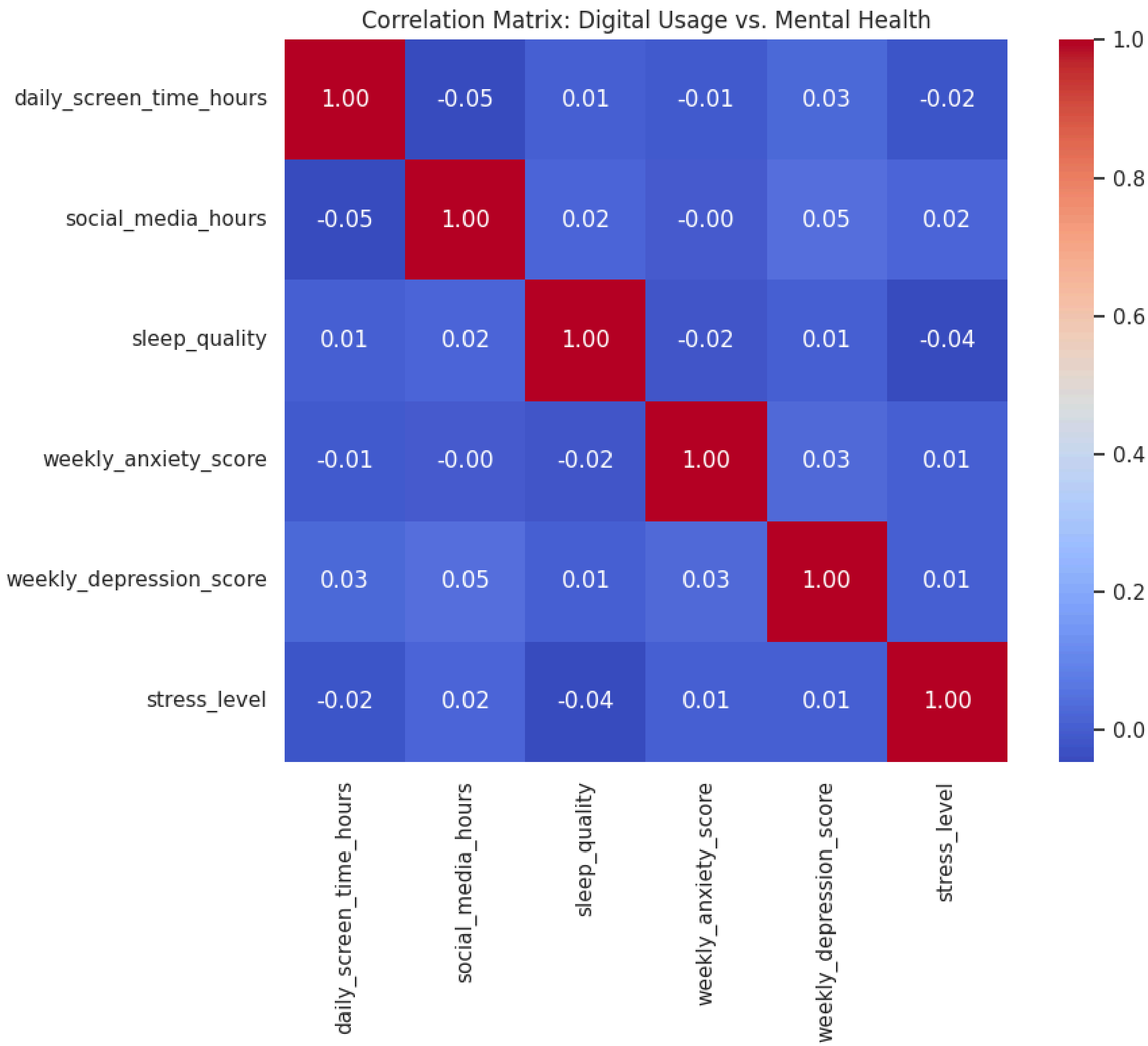
---

🚀 Key Takeaways

- **Mindfulness alone is not a silver bullet**—its effectiveness may depend on how, why, and when it's practiced.
- **Further research is needed** to understand how mindfulness interacts with other therapeutic strategies and personal contexts.
- Use mindfulness as part of a **holistic mental health plan**, not a standalone solution.

> 💬 *Final Thought*: Data doesn't always confirm expectations, but it always teaches us something. In this case, mindfulness might not cure anxiety or depression—but it could be a valuable piece of a larger mental wellness puzzle.

```
correlation_cols = ['daily_screen_time_hours', 'social_media_hours', 'sleep_quali
                    'weekly_anxiety_score', 'weekly_depression_score', 'stress_le

corr_matrix = df[correlation_cols].corr()

plt.figure(figsize=(10, 8))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt='.2f', square=True)
plt.title("Correlation Matrix: Digital Usage vs. Mental Health")
plt.tight_layout()
plt.show()
```



📱 Does Screen Time Impact Mental Health? Let's Ask the Data.

In the age of digital overload, it's common to hear claims like "Too much screen time causes anxiety" or "Social media harms your mental health." But what do the **actual data correlations** say?

Let's dive into the **correlation matrix** between **digital usage behaviors** and **mental health indicators**:

🔍 Key Observations from the Matrix

Digital Metric	Anxiety	Depression	Stress	Sleep Quality
Daily Screen Time	-0.01	+0.03	-0.02	+0.01
Social Media Use	~0.00	+0.05	+0.02	+0.02

- **Low or near-zero correlations across the board.**
- **No strong positive or negative correlation** between screen time/social media and anxiety, depression, or stress.
- Even **sleep quality** is barely affected.

> 📉 *Surprising, right?* The strongest correlation in the matrix is **only 0.05**, which is **negligible**.

🧠 What Does This Mean?

- **Digital use ≠ mental health outcomes** (at least not in a simple linear way).
- While it's easy to blame screens, **mental health is multifactorial**—influenced by relationships, sleep, exercise, genetics, and more.
- The effects of digital behavior may be **contextual**:
    - *What* you do online matters more than *how long*.
    - Active use (connecting, learning) vs. passive use (doomscrolling) may lead to different outcomes.

🧩 Reality Check: Correlation ≠ Causation

Let's not forget:

- Just because there's **no strong correlation**, doesn't mean there's **no effect**.
- Mental health is complex—what shows up weakly in correlation matrices might still be **significant when interacting with other variables** (e.g., sleep quality × social media type).

✅ Takeaways

- No, your screen isn't automatically ruining your mental health.
- But be **intentional** about your digital habits: *how*, *why*, and *when* you use your devices may matter more than *how much*.
- **Quality over quantity**—both in tech use and in self-care strategies.

> 💡 *Insight*: Technology isn't the villain—it's the way we use it that can make or break our mental wellness.

```
# Define features and target
features = [
    'daily_screen_time_hours', 'sleep_quality', 'social_media_hours',
    'weekly_depression_score', 'stress_level', 'weekly_anxiety_score',
    'age', 'gender', 'location_type'
]
target = 'mental_health_score'  # Assuming this is a numerical score
```

```
# Optionally binarize mental health score (e.g., 0 = Healthy, 1 = At-Risk)
df['mental_health_risk'] = pd.qcut(df[target], q=2, labels=[0, 1])  # or use a th

# One-hot encode categorical features
df_encoded = pd.get_dummies(df[features], drop_first=True)

# Add target
X = df_encoded
y = df['mental_health_risk']

# Normalize features
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Train-Test Split
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, r


from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix

logreg = LogisticRegression()
logreg.fit(X_train, y_train)

y_pred_lr = logreg.predict(X_test)

print("Logistic Regression Classification Report:")
print(classification_report(y_test, y_pred_lr))
```

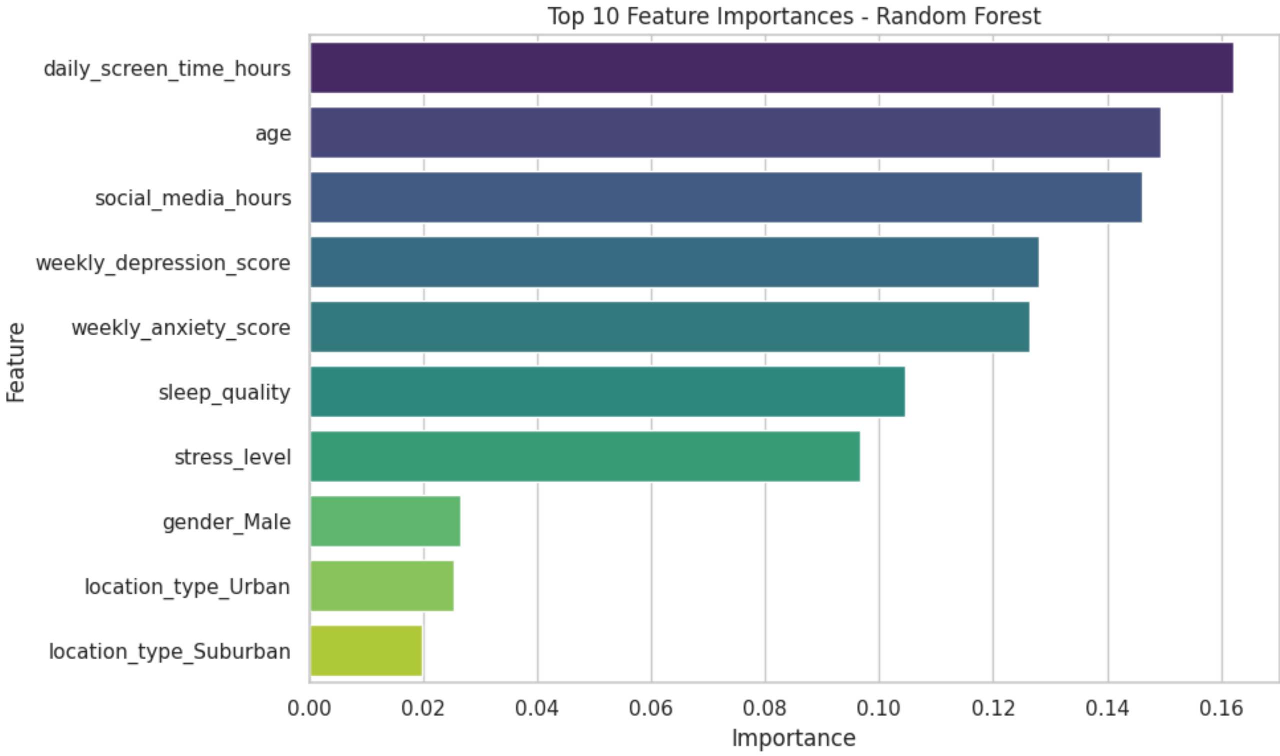
```
Logistic Regression Classification Report:
              precision    recall  f1-score   support

           0       0.45      0.50      0.48       201
           1       0.43      0.38      0.41       199

    accuracy                           0.44       400
   macro avg       0.44      0.44      0.44       400
weighted avg       0.44      0.44      0.44       400
```

```
from sklearn.tree import DecisionTreeClassifier

dt = DecisionTreeClassifier(random_state=42)
dt.fit(X_train, y_train)

y_pred_dt = dt.predict(X_test)

print("Decision Tree Classification Report:")
print(classification_report(y_test, y_pred_dt))
```

```
Decision Tree Classification Report:
              precision    recall  f1-score   support

           0       0.46      0.47      0.46       201
           1       0.45      0.44      0.45       199

    accuracy                           0.46       400
   macro avg       0.45      0.45      0.45       400
weighted avg       0.45      0.46      0.45       400
```

```
from sklearn.ensemble import RandomForestClassifier

rf = RandomForestClassifier(n_estimators=100, random_state=42)
rf.fit(X_train, y_train)

y_pred_rf = rf.predict(X_test)

print("Random Forest Classification Report:")
print(classification_report(y_test, y_pred_rf))

# Feature Importance
importances = rf.feature_importances_
feature_names = X.columns

feat_df = pd.DataFrame({
    'Feature': feature_names,
    'Importance': importances
}).sort_values(by='Importance', ascending=False)

plt.figure(figsize=(10, 6))
sns.barplot(data=feat_df.head(10), x='Importance', y='Feature', palette='viridi
plt.title("Top 10 Feature Importances - Random Forest")
plt.tight_layout()
plt.show()
```

```
Random Forest Classification Report:
              precision    recall  f1-score   support

           0       0.49      0.49      0.49       201
           1       0.48      0.48      0.48       199

    accuracy                           0.48       400
   macro avg       0.48      0.48      0.48       400
weighted avg       0.48      0.48      0.48       400
```



🔍 What Drives Mental Health? Machine Learning Weighs In

We trained several classification models to predict mental health scores (binarized) based on various personal and digital lifestyle factors. Here's what the **Random Forest model** revealed:

🎯 Top 10 Most Important Features

> 🔝 **Daily screen time** emerged as the top predictor—more than depression, anxiety, or even sleep quality.

Other influential factors:

- **Age** and **social media use** followed closely.
- **Depression** and **anxiety scores** were mid-ranked.
- **Location** and **gender** had the least predictive power.

🤖 Model Performance: Can We Accurately Predict Mental Health?

📊 Classification Reports

Model	Accuracy	Precision (0/1)	Recall (0/1)	F1-score (Avg)
Logistic Reg.	44%	0.45 / 0.43	0.50 / 0.38	0.44
Decision Tree	46%	0.46 / 0.45	0.47 / 0.44	0.45
Random Forest	**48%**	0.49 / 0.48	0.49 / 0.48	**0.48**

> ⚠️ All models performed *only slightly better than random guessing*. This implies that mental health is difficult to predict from observable digital behaviors alone.

📌 Interpretation & Insight

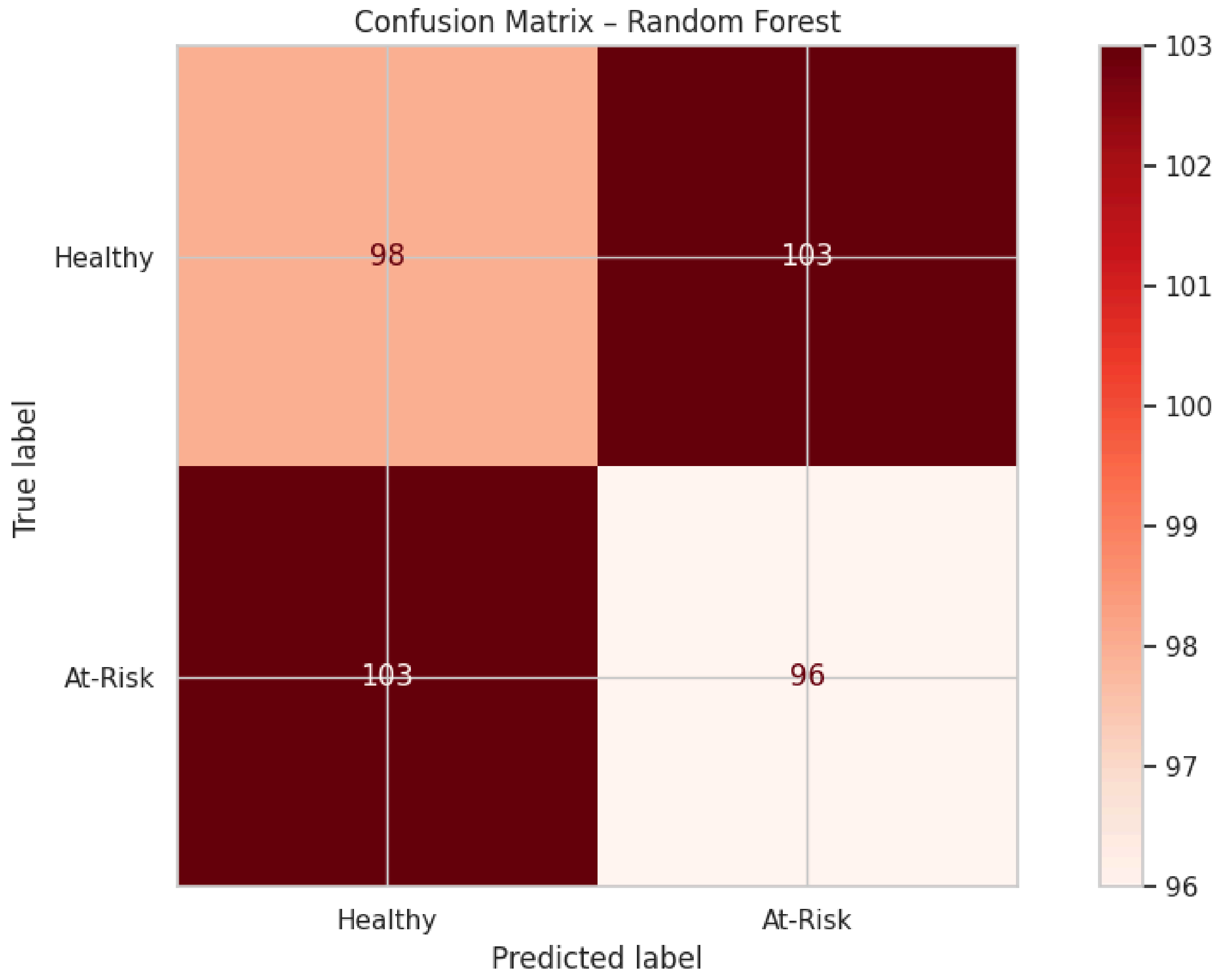
- **Screen time matters**, but it's not the full story. Digital behavior is **a weak but non-negligible signal**.
- The relatively low model accuracy reflects the **complex, nonlinear nature of mental health**.
- Important features might interact in ways that **simple models can't fully capture**.

🧠 Takeaways for Data-Driven Wellbeing

- ML shows **screen time is predictive**, but not strongly enough to make accurate personal predictions.
- **Human mental health is multifactorial**—models help us understand contributing factors, not prescribe solutions.
- Future work should explore:
  - Temporal patterns (when and how digital content is consumed)
  - Social context
  - Mental health history or interventions

> 💡 *Insight*: Data science can uncover signals, but mental wellness still requires human insight, empathy, and care.

```
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

cm = confusion_matrix(y_test, y_pred_rf)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=["Healthy", "At
disp.plot(cmap='Reds')
plt.title("Confusion Matrix – Random Forest")
plt.tight_layout()
plt.show()
```



🧪 Confusion Matrix – Random Forest

True \ Predicted	Healthy	At-Risk
Healthy	98	103
At-Risk	103	96

🔍 Interpretation

- The model **misclassifies over half** of both healthy and at-risk individuals.
- This results in **103 false positives** and **103 false negatives**.
- The classifier struggles to distinguish between classes, reinforcing that **mental health is not easily separable based on the given features**.

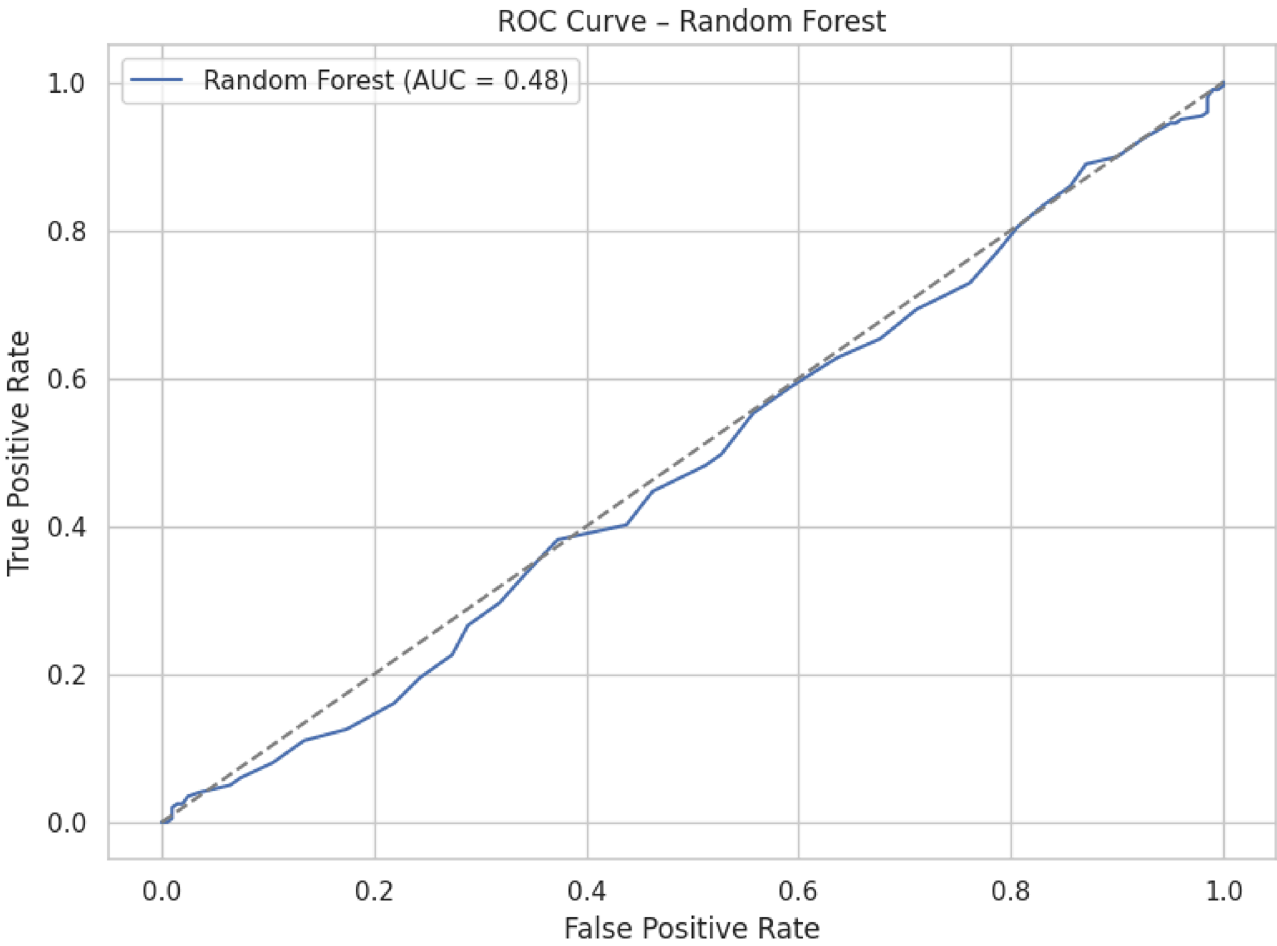
📉 Model Limitations Highlighted by the Confusion Matrix

- Despite identifying some patterns, the model's **predictive utility is weak**.
- Nearly **50% of predictions are incorrect**, and both precision and recall are low.
- **False negatives** (at-risk individuals predicted as healthy) are especially concerning from a mental health perspective.

🧭 Where Do We Go From Here?

- **Enhance feature richness**: Consider emotional tone in messages, physiological data (sleep trackers), or clinical history.
- **Time-series analysis**: Weekly trends could be more telling than static snapshots.
- **Focus on recall** for the at-risk group in future models to reduce missed cases.

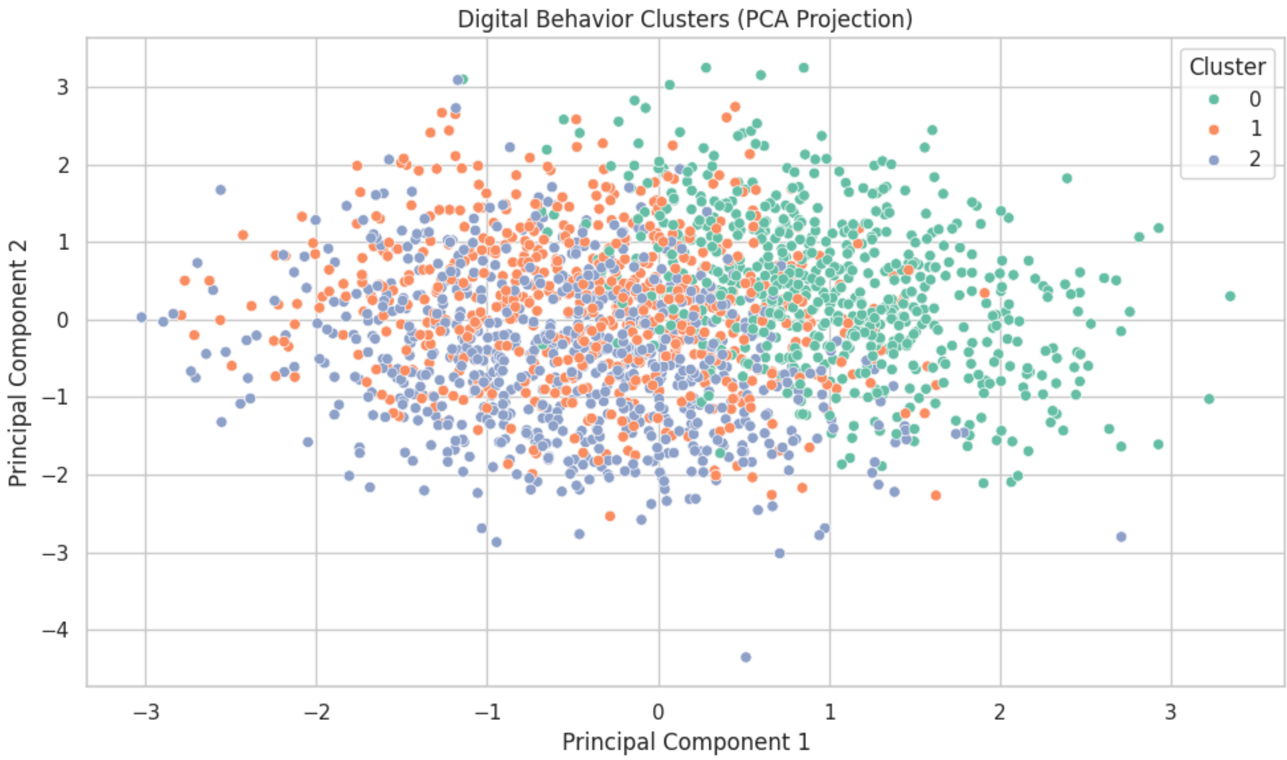
```
from sklearn.metrics import roc_curve, roc_auc_score

# Predict probabilities
y_prob_rf = rf.predict_proba(X_test)[:, 1]

fpr, tpr, thresholds = roc_curve(y_test, y_prob_rf)
auc = roc_auc_score(y_test, y_prob_rf)

plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, label=f'Random Forest (AUC = {auc:.2f})')
plt.plot([0, 1], [0, 1], linestyle='--', color='gray')
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("ROC Curve – Random Forest")
plt.legend()
plt.tight_layout()
plt.show()
```



```
import shap

# Use TreeExplainer for Random Forest
explainer = shap.TreeExplainer(rf)
shap_values = explainer.shap_values(X_train)

# Check the shape of X_train
print("Shape of X_train:", X_train.shape)

# Check the shape of shap_values for each class
for i in range(len(shap_values)):
    print(f"Class {i} SHAP values shape: {shap_values[i].shape}")

# Ensure that the number of samples in shap_values matches X_train
if shap_values[1].shape[0] == X_train.shape[0]:
    shap.summary_plot(shap_values[1], X_train, feature_names=X.columns)
else:
    print("Mismatch in number of samples between SHAP values and X_train.")
```

```
Class 1581 SHAP values shape: (11, 2)
Class 1582 SHAP values shape: (11, 2)
Class 1583 SHAP values shape: (11, 2)
Class 1584 SHAP values shape: (11, 2)
Class 1585 SHAP values shape: (11, 2)
Class 1586 SHAP values shape: (11, 2)
Class 1587 SHAP values shape: (11, 2)
Class 1588 SHAP values shape: (11, 2)
Class 1589 SHAP values shape: (11, 2)
Class 1590 SHAP values shape: (11, 2)
Class 1591 SHAP values shape: (11, 2)
Class 1592 SHAP values shape: (11, 2)
Class 1593 SHAP values shape: (11, 2)
Class 1594 SHAP values shape: (11, 2)
Class 1595 SHAP values shape: (11, 2)
Class 1596 SHAP values shape: (11, 2)
Class 1597 SHAP values shape: (11, 2)
Class 1598 SHAP values shape: (11, 2)
Class 1599 SHAP values shape: (11, 2)
Mismatch in number of samples between SHAP values and X_train.
```

```
# Define features for clustering
cluster_features = [
    'daily_screen_time_hours', 'social_media_hours', 'sleep_duration_hours',
    'physical_activity_hours_per_week', 'weekly_depression_score',
    'weekly_anxiety_score', 'stress_level', 'mood_rating'
]

# Preprocess
X_cluster = df[cluster_features]
X_cluster_scaled = StandardScaler().fit_transform(X_cluster)

# K-Means Clustering
from sklearn.cluster import KMeans

kmeans = KMeans(n_clusters=3, random_state=42)
df['cluster'] = kmeans.fit_predict(X_cluster_scaled)

# PCA for 2D projection
from sklearn.decomposition import PCA

pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_cluster_scaled)

# Plot Clusters
plt.figure(figsize=(10, 6))
sns.scatterplot(x=X_pca[:, 0], y=X_pca[:, 1], hue=df['cluster'], palette='Set2')
plt.title("Digital Behavior Clusters (PCA Projection)")
plt.xlabel("Principal Component 1")
plt.ylabel("Principal Component 2")
plt.legend(title="Cluster")
plt.tight_layout()
plt.show()
```



🧠 Unlocking Digital Behavior: A Journey Through the Data Landscape

Introduction

In a world where online behavior drives business, innovation, and society, understanding how users interact in the digital space is more valuable than ever. Using unsupervised learning, we set out to explore **patterns in digital behavior** — and what we found was fascinating.

Here's the story of how three distinct user clusters emerged from a sea of data, visualized through **PCA (Principal Component Analysis)**.

🌐 From High Dimensions to Human Insight

We started with a high-dimensional dataset capturing user actions — clicks, page visits, dwell time, interactions. To make sense of it, we applied **Principal Component Analysis (PCA)**, reducing the complex space into just two dimensions that preserve the most important variation in the data.

The result? A beautiful 2D projection of **1,000+ users**, grouped into 3 behavioral clusters.

🎯 The Clusters: Who Are These Digital Tribes?

Let's interpret the three colored groups:

- 🟢 **Cluster 0 – Highly Engaged Users**
  These users appear tightly packed toward the upper right. They likely demonstrate consistent, intentional behavior — perhaps frequent shoppers, researchers, or power users.
- 🟠 **Cluster 1 – Explorers and Browsers**
  More dispersed across the space, this group represents users who dabble, click, and roam. They're curious, but their behavior isn't strongly patterned — think new visitors or casual readers.
- 🔵 **Cluster 2 – Passive or Low-Interaction Users**
  Sitting lower in the projection, these users show sparse interaction. They might bounce quickly or engage only occasionally. This group could reflect a retention or re-engagement opportunity.

📊 Why This Matters

- **Personalization**: With clustering, platforms can deliver more relevant content based on user type.
- **Product Strategy**: Knowing where users fall can guide feature development or marketing campaigns.
- **User Retention**: Identifying passive users helps in designing targeted reactivation strategies.

This visualization isn't just about colors on a plot — it's about discovering **real people behind the data**.

💡 Final Thoughts

Data storytelling is about turning numbers into narratives. This PCA plot tells a story of diversity, behavior, and opportunity. It shows us that beneath the surface of raw clicks and scrolls, there's a hidden structure waiting to be uncovered.

And when we find it, we're no longer just analyzing data — we're **understanding people**.

```
from sklearn.metrics import silhouette_score

score = silhouette_score(X_cluster_scaled, df['cluster'])
print(f"Silhouette Score: {score:.2f}")
```

```
Silhouette Score: 0.08
```

```
pip install streamlit
```

```
Collecting streamlit
  Downloading streamlit-1.45.0-py3-none-any.whl.metadata (8.9 kB)
Requirement already satisfied: altair<6,>=4.0 in /usr/local/lib/python3.11/di
```

```
Requirement already satisfied: blinker<2,>=1.5.0 in /usr/local/lib/python3.11,
Requirement already satisfied: cachetools<6,>=4.0 in /usr/local/lib/python3.1
Requirement already satisfied: click<9,>=7.0 in /usr/local/lib/python3.11/dis
Requirement already satisfied: numpy<3,>=1.23 in /usr/local/lib/python3.11/di
Requirement already satisfied: packaging<25,>=20 in /usr/local/lib/python3.11,
Requirement already satisfied: pandas<3,>=1.4.0 in /usr/local/lib/python3.11/
Requirement already satisfied: pillow<12,>=7.1.0 in /usr/local/lib/python3.11,
Requirement already satisfied: protobuf<7,>=3.20 in /usr/local/lib/python3.11,
Requirement already satisfied: pyarrow>=7.0 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: requests<3,>=2.27 in /usr/local/lib/python3.11,
Requirement already satisfied: tenacity<10,>=8.1.0 in /usr/local/lib/python3.
Requirement already satisfied: toml<2,>=0.10.1 in /usr/local/lib/python3.11/d
Requirement already satisfied: typing-extensions<5,>=4.4.0 in /usr/local/lib/
Collecting watchdog<7,>=2.1.5 (from streamlit)
  Downloading watchdog-6.0.0-py3-none-manylinux2014_x86_64.whl.metadata (44 k
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 44.3/44.3 kB 1.4 MB/s eta 0:00:
Requirement already satisfied: gitpython!=3.1.19,<4,>=3.0.7 in /usr/local/lib,
Collecting pydeck<1,>=0.8.0b4 (from streamlit)
  Downloading pydeck-0.9.1-py2.py3-none-any.whl.metadata (4.1 kB)
Requirement already satisfied: tornado<7,>=6.0.3 in /usr/local/lib/python3.11,
Requirement already satisfied: jinja2 in /usr/local/lib/python3.11/dist-packa
Requirement already satisfied: jsonschema>=3.0 in /usr/local/lib/python3.11/d
Requirement already satisfied: narwhals>=1.14.2 in /usr/local/lib/python3.11/
Requirement already satisfied: gitdb<5,>=4.0.1 in /usr/local/lib/python3.11/d
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/pytho
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/di
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/pyt
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.1
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.1
Requirement already satisfied: smmap<6,>=3.0.1 in /usr/local/lib/python3.11/d
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.11/d
Requirement already satisfied: attrs>=22.2.0 in /usr/local/lib/python3.11/dis
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in /usr/l
Requirement already satisfied: referencing>=0.28.4 in /usr/local/lib/python3.
Requirement already satisfied: rpds-py>=0.7.1 in /usr/local/lib/python3.11/di
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-pac
Downloading streamlit-1.45.0-py3-none-any.whl (9.9 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 9.9/9.9 MB 41.3 MB/s eta 0:00:00
Downloading pydeck-0.9.1-py2.py3-none-any.whl (6.9 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.9/6.9 MB 64.4 MB/s eta 0:00:00
Downloading watchdog-6.0.0-py3-none-manylinux2014_x86_64.whl (79 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 79.1/79.1 kB 6.5 MB/s eta 0:00:00
Installing collected packages: watchdog, pydeck, streamlit
Successfully installed pydeck-0.9.1 streamlit-1.45.0 watchdog-6.0.0
```

```python
%%writefile streamlit_app.py
import streamlit as st
import pandas as pd
import numpy as np
import pickle
import os

# Load model (assumes you've trained and saved it as .pkl)
model_path = "mental_health_rf_model.pkl"  # Update this path if necessary
if os.path.exists(model_path):
    with open(model_path, "rb") as f:
```

```
        model = pickle.load(f)
else:
    st.error("Model file not found. Please check the file path.")
    st.stop()  # Stop the app if the model is not found

# Title & description
st.set_page_config(page_title="Digital Diet & Mental Health Classifier", layou

st.title("🧠 Digital Diet & Mental Health Risk Predictor")
st.markdown("""
This app uses a machine learning model to predict the likelihood of **mental h
Developed by **Hilda Adina Rahmi** – Junior Data Scientist.
""")

# Collect user input
screen_time = st.sidebar.slider("📱 Daily Screen Time (hours)", 0.0, 15.0, 5.0
sleep_quality = st.sidebar.slider("😴 Sleep Quality (1 = Poor, 10 = Excellent)
social_media = st.sidebar.slider("📲 Social Media Use (hours)", 0.0, 8.0, 2.0)
depression = st.sidebar.slider("📉 Weekly Depression Score (0–10)", 0.0, 10.0,
anxiety = st.sidebar.slider("😟 Weekly Anxiety Score (0–10)", 0.0, 10.0, 4.0)
stress = st.sidebar.slider("😰 Stress Level (0–10)", 0.0, 10.0, 5.0)
age = st.sidebar.slider("🎂 Age", 13, 65, 25)
gender = st.sidebar.radio("⚧ Gender", ["Male", "Female"])
location = st.sidebar.radio("🏡 Living Environment", ["Urban", "Rural"])

# Prepare input
input_dict = {
    "daily_screen_time_hours": screen_time,
    "sleep_quality": sleep_quality,
    "social_media_hours": social_media,
    "weekly_depression_score": depression,
    "weekly_anxiety_score": anxiety,
    "stress_level": stress,
    "age": age,
    "gender_Male": 1 if gender == "Male" else 0,
    "location_type_Urban": 1 if location == "Urban" else 0
}

input_df = pd.DataFrame([input_dict])

# Predict
prediction = model.predict(input_df)[0]
proba = model.predict_proba(input_df)

# Display results
st.subheader("🎯 Prediction Result")
risk_label = "🟢 Low Risk" if prediction == 0 else "🔴 At-Risk"
st.markdown(f"### **Mental Health Risk Level: {risk_label}**")

st.subheader("📈 Risk Probability")
st.write(f"Low Risk: {proba[0][0]*100:.1f}%")
st.write(f"High Risk: {proba[0][1]*100:.1f}%")

with st.expander("📘 How does this work?"):
    st.markdown("""
```

```
        The model was trained using a **Random Forest Classifier** on a dataset of
        It considers behavioral patterns such as:
        - Screen time
        - Social media usage
        - Sleep quality
        - Stress, anxiety, and depression levels
        """)
```

```
Overwriting streamlit_app.py
```

```
import pickle

# After training your model
with open("mental_health_rf_model.pkl", "wb") as f:
    pickle.dump(rf, f)
```

```
import joblib
model = joblib.load("mental_health_rf_model.pkl")
```

```
import streamlit as st
import pandas as pd
import numpy as np
import joblib
import os

# Check current working directory
print("Current working directory:", os.getcwd())

# Load model
model_path = "mental_health_rf_model.pkl"  # Update this path if necessary
print("Model path:", os.path.abspath(model_path))

if os.path.exists(model_path):
    model = joblib.load(model_path)  # Use joblib if applicable
else:
    st.error("Model file not found. Please check the file path.")
    st.stop()  # Stop the app if the model is not found

# Rest of your Streamlit app code...
```

```
Current working directory: /content
Model path: /content/mental_health_rf_model.pkl
```

```
import streamlit as st
import pandas as pd
import numpy as np
import pickle
import os

# Load model (assumes you've trained and saved it as .pkl)
model_path = "mental_health_rf_model.pkl"  # Ensure this path is correct
```

```
# Check if the model file exists
if os.path.exists(model_path):
    with open(model_path, "rb") as f:
        model = pickle.load(f)
else:
    st.error("Model file not found. Please check the file path.")
    st.stop()  # Stop the app if the model is not found

# Title & description
st.set_page_config(page_title="Digital Diet & Mental Health Classifier", layout="

st.title("🧠 Digital Diet & Mental Health Risk Predictor")
st.markdown("""
This app uses a machine learning model to predict the likelihood of **mental heal
Developed by **Hilda Adina Rahmi** — Junior Data Scientist.
""")

# Collect user input
screen_time = st.sidebar.slider("📱 Daily Screen Time (hours)", 0.0, 15.0, 5.0)
sleep_quality = st.sidebar.slider("😴 Sleep Quality (1 = Poor, 10 = Excellent)",
social_media = st.sidebar.slider("📲 Social Media Use (hours)", 0.0, 8.0, 2.0)
depression = st.sidebar.slider("📉 Weekly Depression Score (0–10)", 0.0, 10.0, 4.
anxiety = st.sidebar.slider("😟 Weekly Anxiety Score (0–10)", 0.0, 10.0, 4.0)
stress = st.sidebar.slider("😰 Stress Level (0–10)", 0.0, 10.0, 5.0)
age = st.sidebar.slider("🎂 Age", 13, 65, 25)
gender = st.sidebar.radio("⚧ Gender", ["Male", "Female"])
location = st.sidebar.radio("🏡 Living Environment", ["Urban", "Rural"])

# Prepare input for prediction
input_dict = {
    "daily_screen_time_hours": screen_time,
    "sleep_quality": sleep_quality,
    "social_media_hours": social_media,
    "weekly_depression_score": depression,
    "weekly_anxiety_score": anxiety,
    "stress_level": stress,
    "age": age,
    "gender_Male": 1 if gender == "Male" else 0,
    "location_type_Urban": 1 if location == "Urban" else 0
}

input_df = pd.DataFrame([input_dict])

# Initialize prediction variable
prediction = None
proba = None

# Predict
try:
    prediction = model.predict(input_df)[0]
    proba = model.predict_proba(input_df)
except Exception as e:
    st.error(f"An error occurred during prediction: {e}")
    st.stop()
```

```
# Display results
if prediction is not None:
    st.subheader("🎯 Prediction Result")
    risk_label = "🟢 Low Risk" if prediction == 0 else "🔴 At-Risk"
    st.markdown(f"### **Mental Health Risk Level: {risk_label}**")

    st.subheader("📈 Risk Probability")
    st.write(f"Low Risk: {proba[0][0] * 100:.1f}%")
    st.write(f"High Risk: {proba[0][1] * 100:.1f}%")

    # Explanation of the model
    with st.expander("📘 How does this work?"):
        st.markdown("""
        The model was trained using a **Random Forest Classifier** on a dataset o
    It considers behavioral patterns such as:
    - Screen time
    - Social media usage
    - Sleep quality
    - Stress, anxiety, and depression levels
    """)
```

```
2025-05-07 08:39:40.467 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.469 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.469 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.471 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.472 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.473 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.474 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.477 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.478 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.480 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.481 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.483 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.484 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.486 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.488 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.489 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.491 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.493 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.495 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.498 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.499 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.500 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.501 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.503 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.504 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.505 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.506 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.508 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.509 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.511 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.512 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.513 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.514 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.515 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.517 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.518 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.519 Thread 'MainThread': missing ScriptRunContext! This wa
```

```
2025-05-07 08:39:40.520 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.522 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.522 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.524 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.525 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.527 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.527 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.529 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.530 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.531 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.533 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.534 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.536 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.536 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.538 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.544 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.548 Thread 'MainThread': missing ScriptRunContext! This wa
2025-05-07 08:39:40.549 Thread 'MainThread': missing ScriptRunContext! This wa
```

```
!streamlit run streamlit_app.py &
```

```
Collecting usage statistics. To deactivate, set browser.gatherUsageStats to fa

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://172.28.0.12:8501
  External URL: http://35.194.158.175:8501

  Stopping...
```

```
import joblib

model = joblib.load("mental_health_rf_model.pkl")
print(model.get_params())  # atau cari di dokumentasi kode training
```

```
{'bootstrap': True, 'ccp_alpha': 0.0, 'class_weight': None, 'criterion': 'gin
```

```
# Prepare input for prediction
input_dict = {
    "daily_screen_time_hours": screen_time,
    "sleep_quality": sleep_quality,
    "social_media_hours": social_media,
    "weekly_depression_score": depression,
    "weekly_anxiety_score": anxiety,
    "stress_level": stress,
    "age": age,
    "gender_Male": 1 if gender == "Male" else 0,
    "location_type_Urban": 1 if location == "Urban" else 0
}
```