Mental Health &Tech Usage Dataset Analysis

By:
Sara Naif Aljohani
Nour Salem Al-Abdulaziz
Reem Rasem Melebari
Maryam Abdullah Bawazir

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

This study investigates the **relationship between mental health and technology usage** through machine learning analysis, aiming to explore how screen time, social media use, sleep patterns, and physical activity relate to mental health indicators and stress levels.

The analysis utilized the Mental Health and Technology Usage Dataset containing 6,635 records with 14 variables. The dataset **underwent preprocessing** including normalization and categorical encoding, with no missing values identified. Multiple methodologies were employed: **regression analysis, classification techniques** (Decision Tree, Random Forest, Logistic Regression, SVM), **clustering approaches** (K-Means and Hierarchical), and **anomaly detection methods.**

Key findings revealed significant limitations in predictive modeling. **Regression models showed poor performance** with negative R^2 values, indicating age cannot predict technology usage or stress outcomes. **Classification models achieved modest accuracy** (25-51%), with behavioral variables like sleep hours, gaming hours, and social media usage emerging as key stress predictors. **Clustering analysis provided more interpretable insights**, with Hierarchical Clustering revealing groupings based on support system access and mental health status, while K-Means separated users primarily by gaming behavior. **Anomaly detection identified unusual behavioural patterns**, though these were not directly linked to mental health outcomes.

The study's primary challenge was **dataset limitations**, including potential random data generation and missing confounding variables. Results suggest that while behavioral features may relate to mental health, they are **insufficient for reliable prediction without additional contextual variables**. These findings underscore the complexity of predicting mental health outcomes from technology usage alone and highlight the need for more comprehensive datasets and advanced modeling approaches in this critical area of public health research.

INTRODUCTION

This project investigates the complex relationship between mental health and technology usage using a variety of machine learning techniques. With the growing impact of digital behavior on wellbeing, analyzing how factors like screen time, social media use, sleep, and physical activity relate to mental health is both relevant and valuable. We applied **multiple machine learning methods**, including **regression**, **classification**, **clustering**, and **anomaly detection**. Each technique helps uncover different aspects of the data—such as predicting mental health indicators, identifying user groups, and detecting unusual patterns.

Objective: To analyze and model the relationship between technology use and mental health through diverse machine learning approaches.

Research Questions:

- How does technology usage impact mental health status?
- Can we predict the behavior (gaming hours, screen time hours, tech usage hours... etc) of an age group? (e.g., Teens aged 13-18 spend most of their time playing games. This could help with targeting ads.)
- Are there any common characteristics of individuals with high stress levels?

This study aims to support better understanding of digital lifestyles and their effects on mental wellbeing, which could inform future tools, policies, or health strategies.

METHODOLGY

Data Source & Description:

We chose Mental Health and Technology Usage Dataset. This dataset offers insights into how daily technology usage, including social media and screen time, impacts mental health. It captures various behavioral patterns and their correlations with mental health indicators like stress levels, sleep quality, and productivity.

The dataset contains 6635 rows and 14 columns. Including 7 columns of numerical variables like 'age', 'Technology_Usage_Hours', and 'Sleep_Hours'. And 6 columns of categorical variables like 'gender', 'Mental_Health_Status', and 'Stress_Level'.

Criteria	Description	Check (✓/✗)	How the Dataset Meets This Requirement
Topic of Data	The dataset should focus on a meaningful subject to make the analysis more valuable.	✓	The dataset focuses on the relationship between technology usage and mental health factors.
Structured Data	The dataset should be in a table format, with rows as records and columns as features.	✓	Data is organized in rows and columns; each row represents an individual and columns represent attributes.
Unique Identifier	There should be a column with a unique identifier (e.g., ID, username, serial number).	✓	The dataset contains a column named 'User_ID' with unique values for each individual.
Dataset Size	The dataset should be large enough for meaningful analysis (e.g., 1,000+ rows).	✓	The dataset contains 6635 rows and 14 column, making it suitable for analysis.
Numerical Variables	The dataset should include numerical variables (e.g., age, price).	✓	The dataset contains 7 columns of numerical variables like 'age', 'Technology_Usage_Hours', 'Sleep_Hours' and 'Gaming_Hours'.
Categorical Variables	The dataset should include categorical variables (e.g., gender, product type).	✓	The dataset contains 6 columns of categorical variables like 'gender', 'Mental_Health_Status', 'Stress_Level' and 'Work_Environment_Impact'.

Column Description:

1. **User_ID:** A unique identifier for each user in the dataset.
2. **Age:** The age of the user.
3. **Gender:** The gender of the user (e.g., Male, Female).
4. **Technology_Usage_Hours:** The average number of hours per day the user spends on technology.
5. **Social_Media_Usage_Hours:** The average number of hours per day the user spends on social media.
6. **Gaming_Hours:** The average number of hours per day the user spends playing video games.
7. **Screen_Time_Hours:** The total average screen time per day, combining all hours stated above.
8. **Mental_Health_Status:** The user's reported mental health status (e.g., Good, Fair, Excellent, Poor).
9. **Stress_Level:** The level of stress reported by the user (e.g., Low, Medium, High).
10. **Sleep_Hours:** The average number of hours the user sleeps per night.
11. **Physical_Activity_Hours:** The average number of hours the user engages in physical activity per week.
12. **Support_Systems_Access:** Indicates whether the user has access to support systems (e.g., Yes, No).
13. **Work_Environment_Impact:** The impact of the work environment on the user's wellbeing (e.g., Positive, Negative, Neutral).
14. **Online_Support_Usage:** Indicates whether the user utilizes online support resources (e.g., Yes, No).

N	M	L	K	J	I	H	G	F	E	D	C	B	A	
Online_Support_Usage	Work_Environment_Impac	Support_Systems_Access	Physical_Activity_Hours	Sleep_Hours	Stress_Leve	Mental_Health_Status	Screen_Time_Hours	Gaming_Hours	Social_Media_Usage_Hour	Technology_Usage_Hours	Gender	Age	User_ID	1
Yes	Negative	No	6.71	8.01	Low	Good	12.36	0.68	6	6.57	Female	23	USER-00001	2
No	Positive	Yes	5.88	7.28	High	Poor	7.61	3.74	2.57	3.01	Male	21	USER-00002	3
No	Negative	No	9.81	8.04	High	Fair	3.16	1.26	6.14	3.04	Male	51	USER-00003	4
Yes	Negative	Yes	5.28	5.62	Medium	Excellent	13.08	2.59	4.48	3.84	Female	25	USER-00004	5
Yes	Positive	No	4	5.55	Low	Good	12.63	0.29	0.56	1.2	Male	53	USER-00005	6
Yes	Neutral	Yes	6.54	8.61	Low	Poor	1.34	0.11	5.74	5.59	Male	59	USER-00006	7
No	Neutral	Yes	5.27	7.11	Medium	Excellent	8.14	4.74	4.1	7.18	Female	51	USER-00007	8
Yes	Negative	Yes	0.49	7.9	High	Poor	10.65	2.54	4.06	3.06	Female	40	USER-00012	9
No	Negative	Yes	7.14	7.13	High	Poor	10.27	0.94	4.53	10.48	Female	38	USER-00014	10
No	Negative	Yes	5.05	8.53	Medium	Good	5.08	4.17	0.21	7.51	Male	53	USER-00015	11
Yes	Neutral	Yes	9.65	4.62	High	Fair	11.4	4.76	5.4	6.4	Female	26	USER-00016	12
No	Negative	Yes	8.96	8.95	Low	Fair	14.88	2.01	6.56	2.03	Female	55	USER-00017	13
No	Neutral	Yes	7.51	7.71	High	Excellent	11.49	2.13	0.61	2.94	Male	55	USER-00018	14
No	Neutral	Yes	5.65	7.07	Low	Fair	9.55	3.16	7.33	7.69	Female	57	USER-00020	15
Yes	Negative	No	0.06	4.76	High	Fair	1.51	2.28	1.14	8.44	Male	34	USER-00021	16
Yes	Neutral	No	6.48	4.01	Low	Poor	2.83	2.92	1.22	1.27	Female	38	USER-00022	17
Yes	Negative	Yes	7.59	5.83	Low	Excellent	9.8	3.96	5.99	2.06	Male	39	USER-00023	18
Yes	Positive	No	6.55	5.33	Medium	Poor	12.79	2.49	0.06	2.2	Male	23	USER-00026	19
Yes	Neutral	No	4.97	4.44	High	Good	14.72	3.2	0.51	1.16	Female	50	USER-00027	20
No	Neutral	Yes	9.28	8.01	Medium	Excellent	1.63	0.32	7.48	3.65	Female	20	USER-00029	21
No	Positive	Yes	9.78	8.27	Low	Fair	8.48	2.18	4.07	5.98	Female	40	USER-00030	22
Yes	Positive	No	9.74	6.24	Medium	Fair	8.05	2.86	6.11	11.8	Female	29	USER-00031	23
No	Negative	No	3.69	4.4	High	Poor	5.32	2.47	1.42	11.51	Female	55	USER-00032	24
Yes	Negative	No	7.84	5.02	Low	Good	2.41	3.53	3.53	1.33	Male	30	USER-00033	25
No	Neutral	Yes	7.99	7.1	Low	Fair	1.69	0.61	4.52	9.91	Male	47	USER-00034	26
No	Negative	Yes	8.17	5.77	Low	Poor	13.89	1.75	7.42	5.66	Male	37	USER-00035	27
Yes	Positive	No	4.69	4.47	Low	Good	12.79	1.1	4.66	6.65	Female	44	USER-00037	28
Yes	Positive	Yes	2.25	4.7	Low	Excellent	5.37	4.69	2.98	10.87	Male	23	USER-00038	29
Yes	Neutral	Yes	5.26	6.7	Low	Fair	1.22	3.62	1.62	7.05	Male	48	USER-00040	30
Yes	Negative	No	8.4	7.59	Medium	Excellent	13.84	2.46	1.95	1.04	Male	57	USER-00041	31
Yes	Negative	Yes	3.62	5.73	Low	Good	14.03	0.81	4.5	4.45	Male	50	USER-00043	32
No	Neutral	Yes	0.9	4.46	High	Excellent	8.76	1.97	5.93	2.06	Female	23	USER-00045	33
No	Positive	Yes	1.48	4.38	Medium	Good	2.75	2.33	4.26	3.34	Female	30	USER-00046	34
No	Negative	No	2.07	8.02	Medium	Poor	6.9	1.96	3.16	2.95	Male	45	USER-00048	35
Yes	Neutral	No	8.6	4.41	Low	Excellent	1.2	2.76	0.82	3.06	Female	38	USER-00049	36
Yes	Neutral	Yes	4.22	4.35	High	Excellent	5.84	1.97	5.02	4.53	Female	37	USER-00050	37
Yes	Neutral	No	9.31	5.53	Medium	Excellent	14.31	1.64	7.47	7.29	Male	23	USER-00052	38
No	Neutral	Yes	4.2	7.35	High	Poor	9.01	2.29	7.78	9.24	Male	40	USER-00055	39
Yes	Negative	Yes	1.84	6.03	High	Fair	8.35	1.54	3.38	6.98	Female	62	USER-00056	40
No	Negative	No	9.67	8.2	Medium	Poor	5.94	2.39	5.35	5.55	Male	44	USER-00057	41
No	Negative	Yes	4.82	8.6	Low	Fair	8.57	2.66	7.07	11.15	Male	59	USER-00059	42
No	Negative	No	9.41	4.08	Low	Poor	2.4	3.15	3.21	4.53	Female	39	USER-00060	43
Yes	Positive	Yes	8.89	5.37	Medium	Excellent	5.66	4.15	2.13	3.22	Female	32	USER-00061	44

Figure 1: Preview of the Dataset Displayed in Excel.

Data Preprocessing:

To ensure the quality and reliability of the dataset used in this study, a comprehensive data cleaning process was conducted. The following steps were applied:

Data Type Consistency:
The data types of the columns were inspected using df.info(). The dataset contains consistent types—numerical columns are stored as int64 or float64, while categorical data are stored as object.

Missing Values:
The presence of missing values was checked using df.isnull().sum(). All columns showed zero missing values.

Duplicate Records:
Duplicate rows were identified using df.duplicated().sum(), and none were found.

Outlier Detection:
Outliers in numerical columns were identified by calculating IQR. This showed that no outliers existed in any of the numerical features. This was visually validated with boxplots and histograms.

Normalization:
To ensure all features contribute equally to future statistical or machine learning models, Normalization was applied using Min-Max Scaling to bring all numerical values into a range between 0 and 1.

The dataset was thus confirmed to be clean, consistent, and ready for analysis.

Commonly Used Libraries and Functions:

Data Handling:
Used NumPy and Pandas for efficient numerical operations and structured data manipulation.

Data Visualization:
Utilized Matplotlib and Seaborn for creating visualizations like histograms, boxplots, and heatmaps.

Preprocessing:
Applied Scikit-learn tools such as LabelEncoder, OrdinalEncoder, MinMaxScaler, StandardScaler, PolynomialFeatures, and PCA for encoding, scaling, and dimensionality reduction.

Model Selection & Evaluation:
Used tools like train_test_split, KFold, cross_val_score, and metrics including accuracy_score, f1_score, etc., to train and evaluate models.

Machine Learning Models:
Implemented Linear/Logistic Regression, Decision Tree, Random Forest, SVM, One-vs-Rest, and One-Class SVM, each discussed in relevant sections.

These libraries formed the backbone of the analysis pipeline, allowing for a smooth and efficient machine learning workflow.

Linear and Nonlinear Regression

Simple Linear Regression:

We imported the LinearRegression model using sklearn.linear_model library. And the metrics using sklearn.metrics.

First, we decided to check if there's a linear relationship between Age and Screen Usage Hours using Simple Linear Regression.

X, The independent variable: *Age.*
Y, Dependent variable: *Screen Time Hours*.

we divided the dataset by both 80% training - 20% testing and K-fold (k=5) methods.

to evaluate the model, we calculate: RMSE, R-squared, and adjusted R-squared.

we use a scatter plot to visualize the data and see if there is a relationship that the model can predict.

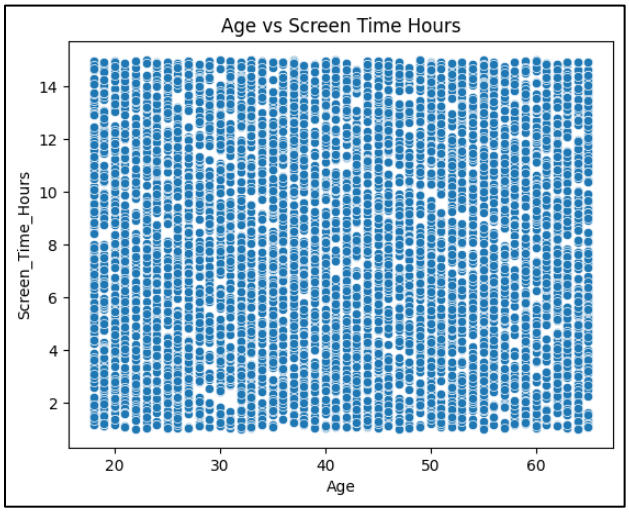

Figure 2: scatter plot for Simple Linear Regression.

Simple Nonlinear Regression:

We imported the SVR model from the sklearn.svm library, along with evaluation metrics from sklearn.metrics.

We decided to explore whether a non-linear relationship exists between Age and Screen Time Hours using Support Vector Regression (SVR) with the Radial Basis Function (RBF) kernel. To ensure fairness in model evaluation.

X, The independent variable: *Age.*
Y, Dependent variable: *Screen Time Hours*.

we applied both 80%-20% splitting and K-Fold (k=5) cross-validation to dividing the data.

to evaluate the model, we calculate: RMSE, R-squared, and adjusted R-squared.

to visualize the data, we use predicted data vs actual data plot.

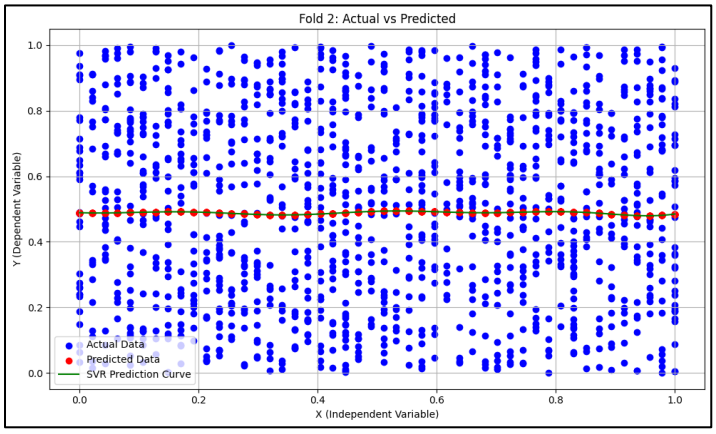

Figure 3: plot of predicted data vs actual data for Simple Nonlinear Regression.

Multiple Linear Regression:

We imported the LinearRegression model from the sklearn.linear_model library, along with evaluation metrics from sklearn.metrics.

by using multiple linear model, we decided to explore whether there is an effect on stress levels from various variables.

X, The independent variable: *Sleep Hours, Technology Usage Hours, Work Environment Impact.*
Y, Dependent variable: *Stress Level.*

to dividing the data, we applied: an 80%-20% train-test split and k-fold cross-validation with k=5.

we applied the data presentation using the chart shown to clearly depict the model's performance based on the target variable.

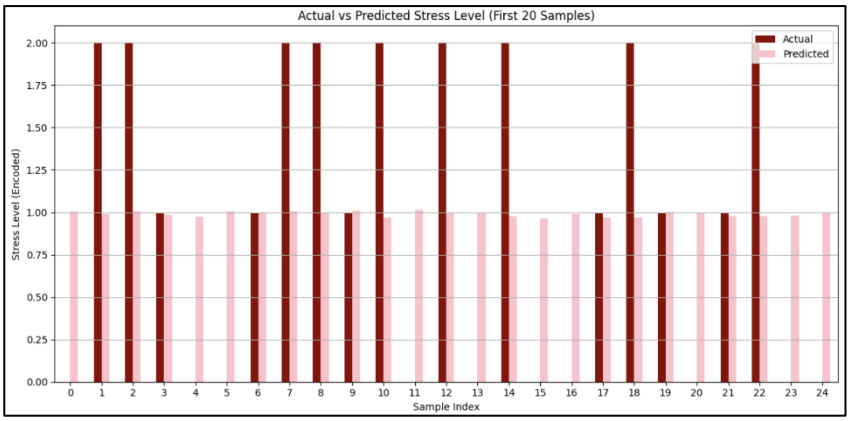

Figure 4: plot of predicted data vs actual data as a chart for Multiple Linear Regression.

Multiple Nonlinear Regression:

We used the non-LinearRegression model from sklearn.linear_model, combined with PolynomialFeatures from sklearn.preprocessing, to explore potential nonlinear relationships in the data. The evaluation metrics were imported from sklearn.metrics.

we decided to explore whether there are some different variables that effect on the number of playing hours.

X, The independent variable: *Age, Gender, Social Media Usage Hours, Sleep Hours.*

Y, Dependent variable: *Gaming Hours.*

We applied two data splitting methods: 80% training - 20% testing split and K-Fold Cross-Validation (k=5) to ensure consistency and reduce bias.

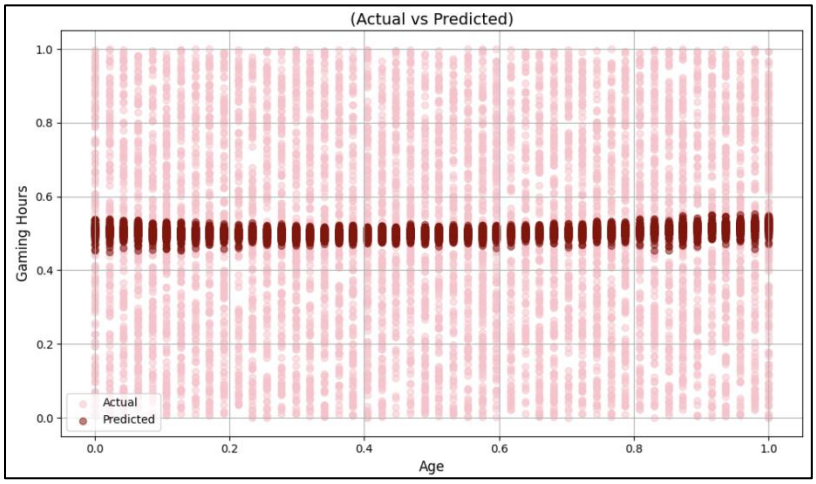

Figure 5: plot of predicted data vs actual data for Multiple Nonlinear Regression.

Classification

Decision Tree:

We used the *DecisionTreeClassifier* from *sklearn.tree*, in combination with *SMOTE* from *imblearn.over_sampling*, to address class imbalance in predicting stress levels. Label encoding was applied to convert categorical features into numerical values.
We aimed to investigate how different user-related features influence stress level classification.
X, The independent variables: *Age, Screen Time, Sleep Hours, Device Type, Physical Activity*, and other behavioral factors.
Y, Dependent variable*: Stress_Binary (0 for Low, 1 for Medium/High).*
To ensure robust evaluation, we used both 80%-20% train-test split and 5-Fold Cross-Validation. SMOTE was applied within each fold to oversample the minority class. The model's performance was evaluated using a confusion matrix, classification report, and average accuracy score.

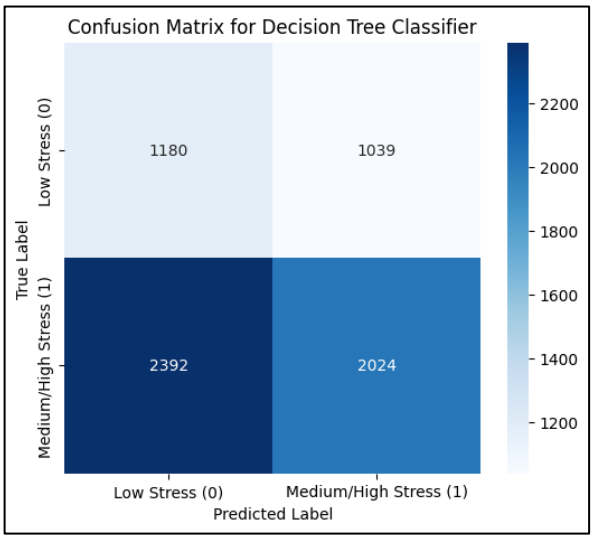

Figure 7: Confusion Matrix for Decision Tree Classifier comparing actual vs predicted stress levels (Low vs Medium/High), highlighting classification performance.

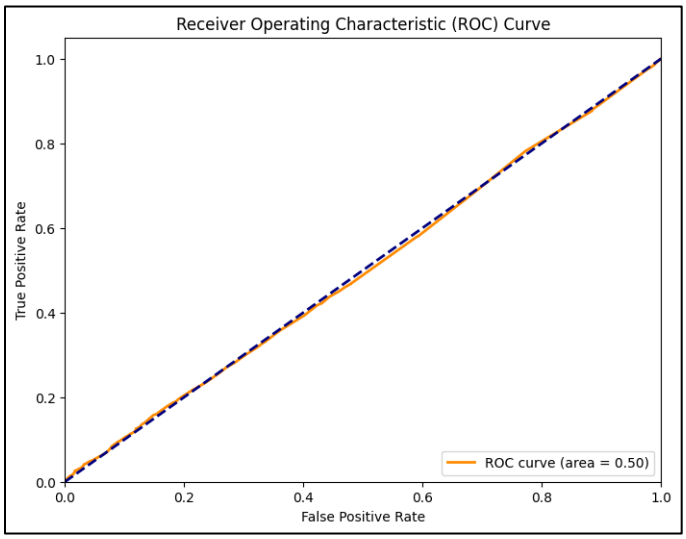

Figure 6: ROC Curve for Decision Tree Classifier showing the model's performance in distinguishing between Low Stress and Medium/High Stress. AUC indicates overall accuracy.

Random Forest:

We used the *RandomForestClassifier* from *sklearn.ensemble,* combined with a preprocessing pipeline to handle both categorical and numerical features. Categorical encoding was performed using *OrdinalEncoder* through a Column Transformer, while numerical features were passed through unchanged.
Our goal was to predict users' stress levels based on selected factors.
X, The independent variables: *Age, Mental Health Status, and Sleep Hours.*
Y, Dependent variable*: Stress_Level (categorical: Low, Medium, High).*
We split the dataset using an 80% training – 20% testing ratio. The model was trained within a pipeline that first transformed categorical data, then applied the random forest classifier.
After training, we evaluated the model by comparing predicted stress levels with actual values from the test set to observe its performance on unseen data.

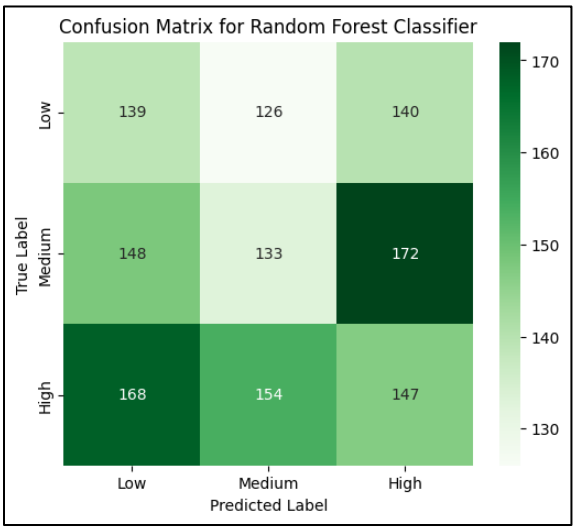

Figure 8: Confusion Matrix for Random Forest Classifier comparing actual vs predicted stress levels across three classes: Low, Medium, and High.

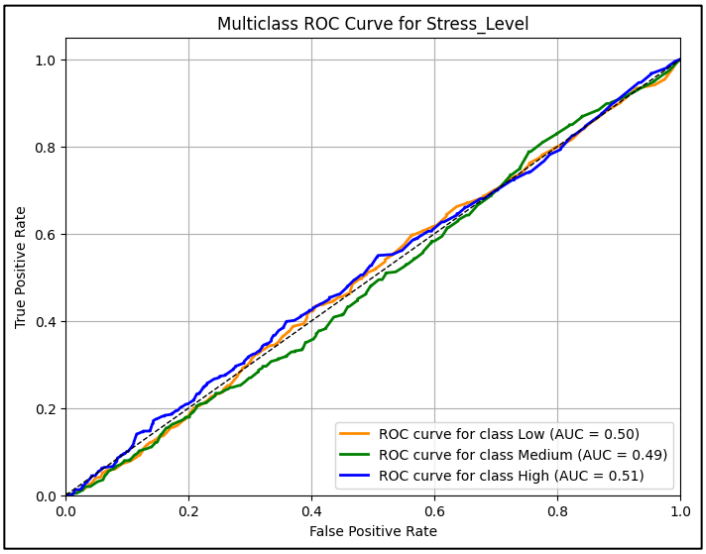

Figure 9: Multiclass ROC Curve for Random Forest Classifier showing the model's performance in predicting stress levels (Low, Medium, High). Each curve represents one class with its corresponding AUC.

Logistic Regression:

We used the *LogisticRegression* model from *sklearn.linear_model* to predict the likelihood of online support usage based on behavioral and mental health features. All categorical columns were encoded using *LabelEncoder* to prepare the data for modeling.
Our objective was to explore how stress levels, mental health status, and technology usage relate to the use of online support resources.
X, The independent variables: *Stress_Level, Mental_Health_Status, Technology_Usage_Hours.*
Y, Dependent variable: *Online_Support_Usage (binary: 0 or 1).*
We applied 5-Fold Cross-Validation using *KFold* to ensure robust model evaluation. Predictions and probabilities were generated using *cross_val_predict.*
Model performance was assessed using accuracy, precision, recall, F1-score, confusion matrix, and a detailed classification report. These metrics provided insights into the model's ability to identify individuals likely to use online support services.

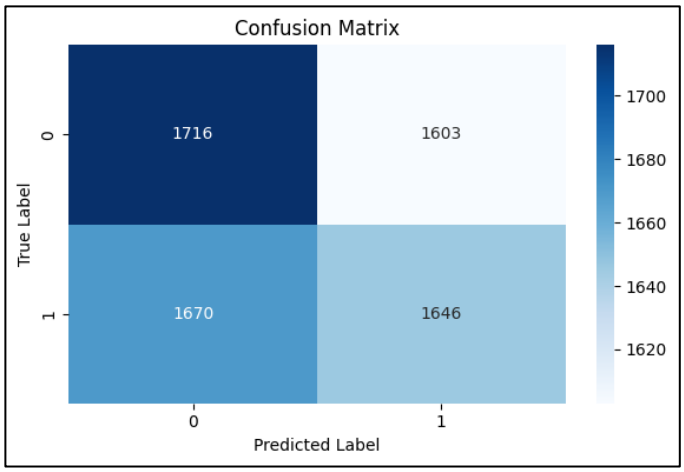

Figure 10: Confusion Matrix for Logistic Regression Classifier comparing actual vs predicted Online Support Usage across two classes: Yes and No.

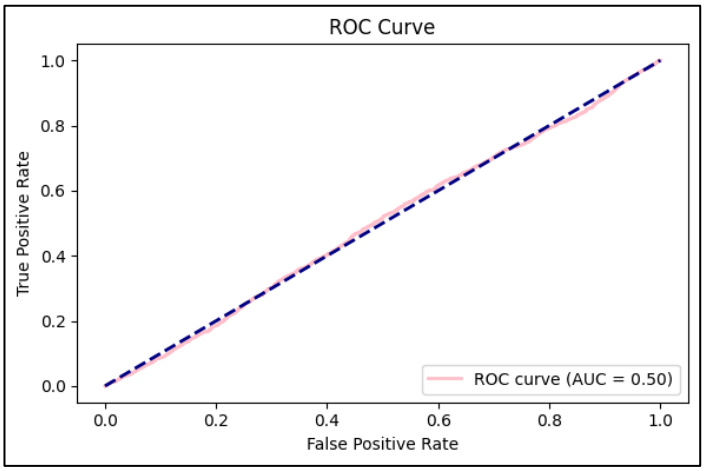

Figure 11:ROC Curve illustrating the trade-off between true positive rate and false positive rate for the classifier, with the AUC indicating overall model performance

SVC (Support Vector Classifier):

We used the SVC model from *sklearn.svm* with a linear kernel and balanced class weights to classify mental health status based on various lifestyle and support-related features.
Categorical variables, including *Stress_Level* and *Online_Support_Usage*, were encoded using *LabelEncoder*, and feature scaling was applied using *StandardScaler* to standardize the input data.
X, The independent variables: *Stress_Level, Sleep_Hours, Physical_Activity_Hours, Online_Support_Usage*.
Y, Dependent variable: *Mental_Health_Status (encoded).*
The dataset was split into 75% training and 25% testing. After training the model, we evaluated its performance using a confusion matrix and classification report. These metrics helped assess the model's ability to differentiate between mental health categories based on behavioral and support usage patterns.

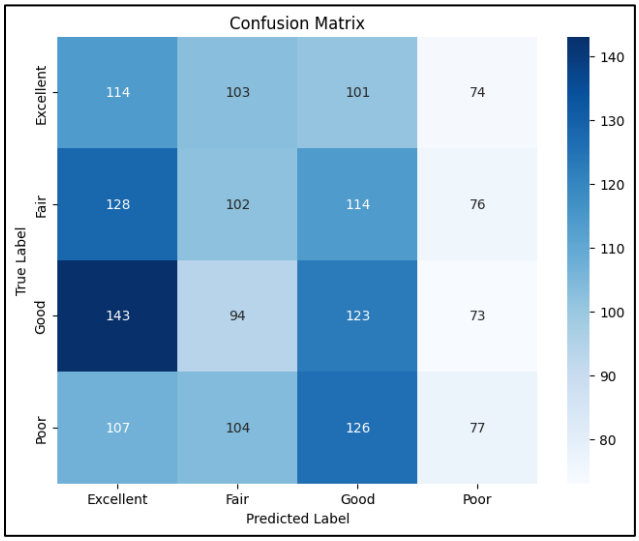

Figure 12: Confusion Matrix for SVM Classifier comparing actual vs predicted Mental Health Status across four classes: Excellent, Fair, Good, and Poor.

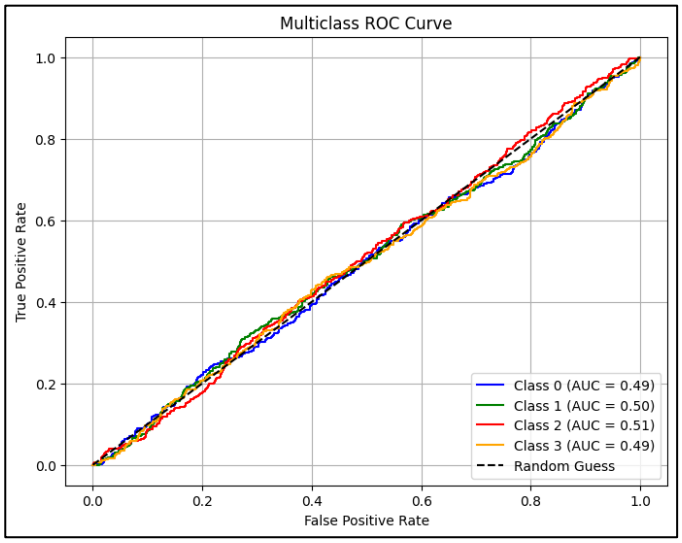

Figure 13: Multiclass ROC Curve showing One-vs-Rest SVM performance across Mental Health Status classes with AUC scores.

Clustering

K-Means:

We imported the K-Means algorithm from sklearn.cluster, and used StandardScaler from sklearn.preprocessing to normalize our numeric data.
To begin, we selected only numerical columns to ensure compatibility with clustering, and then applied scaling to bring all features to a similar range.

```
First 5 rows of the data:
      User_ID  Age  Gender  Technology_Usage_Hours  Social_Media_Usage_Hours  \
0  USER-00001   23  Female                    6.57                      6.00
1  USER-00002   21    Male                    3.01                      2.57
2  USER-00003   51    Male                    3.04                      6.14
3  USER-00004   25  Female                    3.84                      4.48
4  USER-00005   53    Male                    1.20                      0.56

   Gaming_Hours  Screen_Time_Hours Mental_Health_Status Stress_Level  \
0          0.68              12.36                Good          Low
1          3.74               7.61                Poor         High
2          1.26               3.16                Fair         High
3          2.59              13.08           Excellent       Medium
4          0.29              12.63                Good          Low

   Sleep_Hours  Physical_Activity_Hours Support_Systems_Access  \
0         8.01                     6.71                     No
1         7.28                     5.88                    Yes
2         8.04                     9.81                     No
3         5.62                     5.28                    Yes
4         5.55                     4.00                     No

  Work_Environment_Impact Online_Support_Usage  Cluster
0                Negative                  Yes        1
1                Positive                   No        0
2                Negative                   No        1
3                Negative                  Yes        1
4                Positive                  Yes        2

Data after scaling:
        Age  Technology_Usage_Hours  Social_Media_Usage_Hours  Gaming_Hours  \
0 -1.344112                0.034400                  0.874272     -1.274948
1 -1.488592               -1.089069                 -0.608490      0.847462
2  0.678611               -1.079601                  0.934792     -0.872661
3 -1.199632               -0.827136                  0.217188      0.049824
4  0.823091               -1.660270                 -1.477397     -1.545451

   Screen_Time_Hours  Sleep_Hours  Physical_Activity_Hours
0           1.084261     1.040530                 0.587604
1          -0.089387     0.537752                 0.301057
2          -1.188910     1.061192                 1.657838
3           1.262161    -0.605552                 0.093915
4           1.150973    -0.653764                -0.347988
```

Figure 14: First five rows of the data before and after normalizing numerical features.

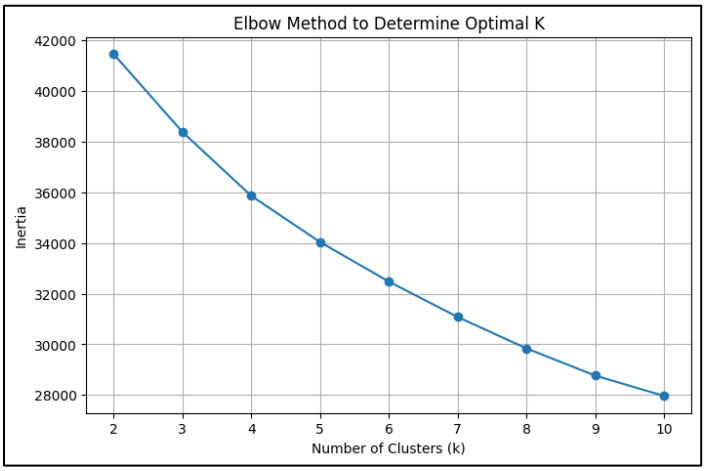

Figure 15: Elbow method result.

Before clustering, we applied the **Elbow Method** to determine the optimal number of clusters (k). The plot, in *Figure 15*, suggested that **k = 2** would be a reasonable choice, where the drop in inertia started to level off.

Figure 16: 2D scatter plots (Without PCA) to visualize the clusters.

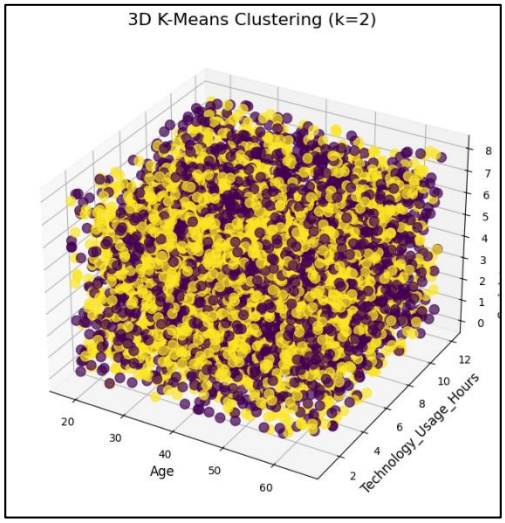

Figure 17: 3D scatter plots (Without PCA) to visualize the clusters.

We, then, trained the K-Means model with k = 2, and the resulting cluster labels were added to the original dataset. The *Figure 16* and *Figure 17* shows visualization using 2D and 3D scatter plots (based on selected numeric features) that illustrate a clear overlapping of the two clusters' points.

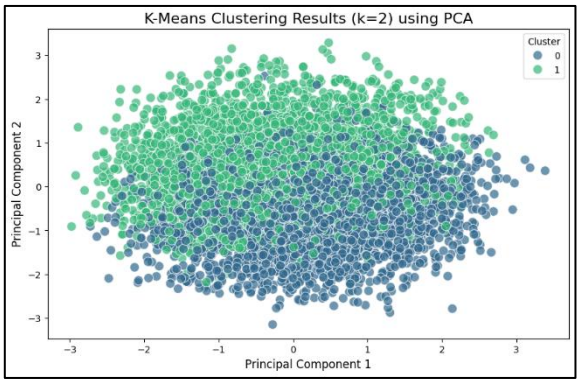

Figure 18: Scatter plot using the same results used in Figure 16, but after applying PCA.

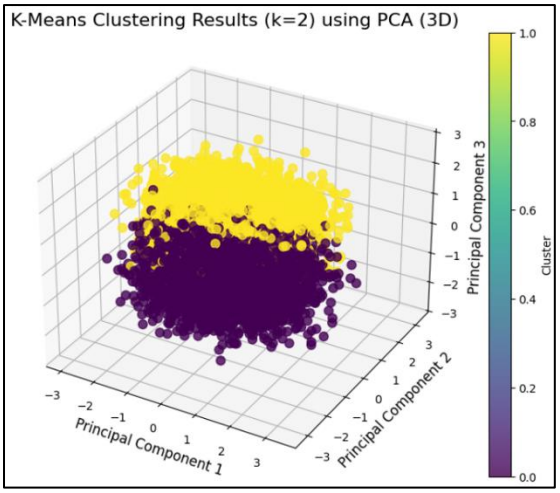

Figure 19: Scatter plot using the same results used in Figure 17, but after applying PCA.

In addition to feature-based plots, we also applied **PCA (Principal Component Analysis)** to reduce dimensionality and visualize the clusters in PCA space. Seen in *Figure 18* and *Figure 19*, the 2D and 3D PCA plots showed a distinguishable separation between the clusters, introducing reliability of the K-Means result.

Although we didn't use target labels in this unsupervised method, the clustering revealed structure within the data and showed that there are potentially distinct groups with similar behavioral patterns across the numeric features.

Hierarchical Agglomerative Clustering:

Before applying clustering, we used **dendrogram cutting** to find the best number of clusters for our data. We started by removing the User_ID column since it doesn't help with grouping. Then, we **encoded categorical features** using LabelEncoder and **scaled the data** with StandardScaler to make sure all values are on the same scale.

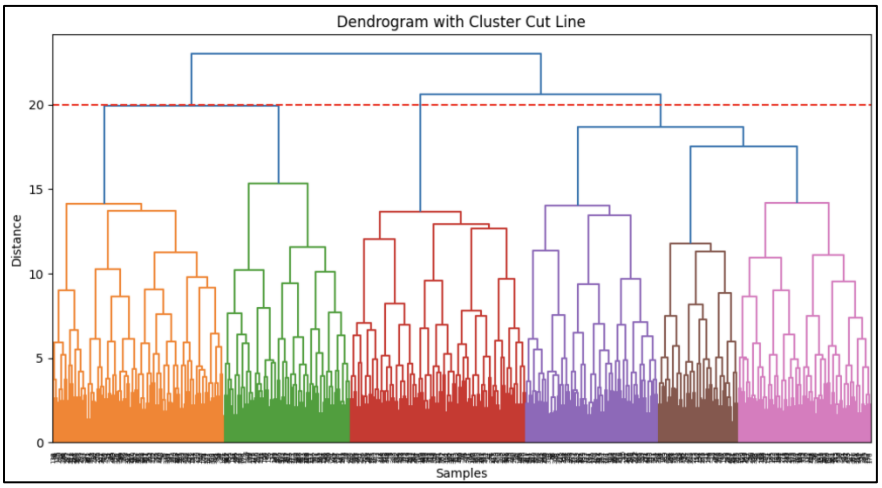

Figure 20: Dendrogram Cutting result.

Seen in *Figure 20*, by drawing a horizontal cut line at a specific height, the dendrogram showed that the best separation occurs at **2 clusters**. Based on this result, we moved forward with **Hierarchical Agglomerative Clustering** using 2 clusters.

To explore potential patterns in user behavior, we used AgglomerativeClustering imported by sklearn.cluster.

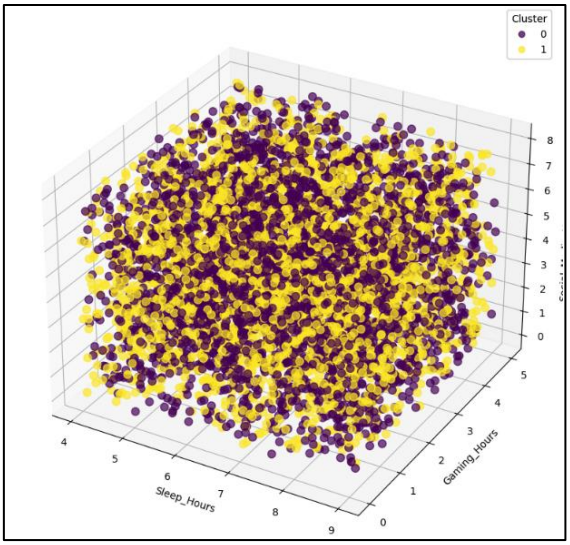

Figure 21: scatter plot visualization before applying PCA.

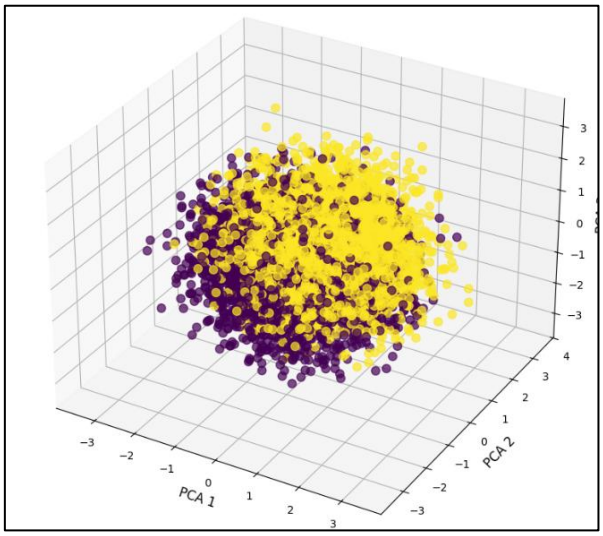

Figure 22: 3D scatter plot visualization after applying PCA.

Figure 21 and *Figure 22* shows that the two clusters found by Agglomerative Clustering are **not very well separated**.

To better understand the separation between clusters, we applied **PCA** to reduce the dataset's high dimensionality, as shown in Figure 0.0, allowing for a 3D scatter plot.

PCA helped to visualize how users were grouped. And **despite some overlapping, the two clusters appeared to occupy distinct regions** in the 3D PCA space.

This suggests underlying differences in behavior between the two groups, even though they might not be linearly separable in the original feature space.

Anomaly Detection

Statistical methods:

Z-Score:

We applied the Z-Score statistical method to detect anomalies across the following numerical variables: Technology_Usage_Hours, Social_Media_Usage_Hours, Gaming_Hours, Screen_Time_Hours, Sleep_Hours, and Physical_Activity_Hours.

To begin, we standardized these features using the StandardScaler to normalize the data. Then, Z-Scores were calculated for each observation to measure how far each value deviates from the mean in terms of standard deviations. Our plan was to flag any observation with a Z-Score greater than +3 or less than -3 as a potential outlier, **as this typically indicates abnormal behavior within normally distributed data.**

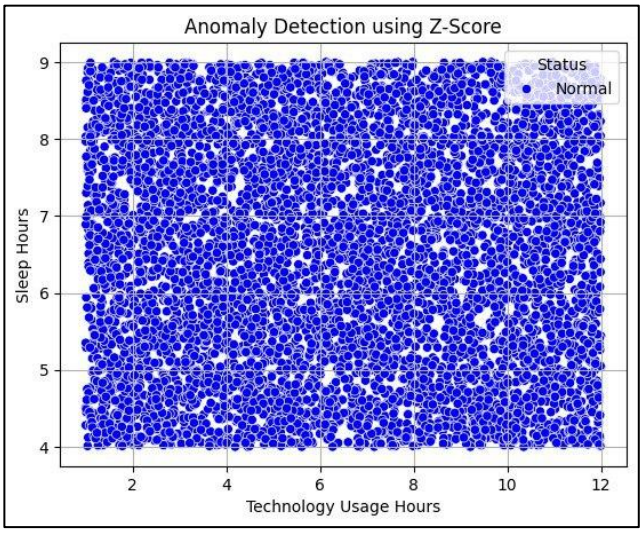

Figure 23: Scatter Plot Visualizations for Z-Score

IQR:

We applied the IQR (Interquartile Range) statistical method to detect anomalies within the numerical variables: Technology_Usage_Hours, Social_Media_Usage_Hours, Gaming_Hours, Screen_Time_Hours, Sleep_Hours, and Physical_Activity_Hours.

To identify potential outliers, we calculated the first (Q1) and third quartiles (Q3) for each variable. Using these, we determined the lower and upper bounds based on the IQR. **Any data point falling below the lower bound or above the upper bound was flagged as a potential anomaly.**

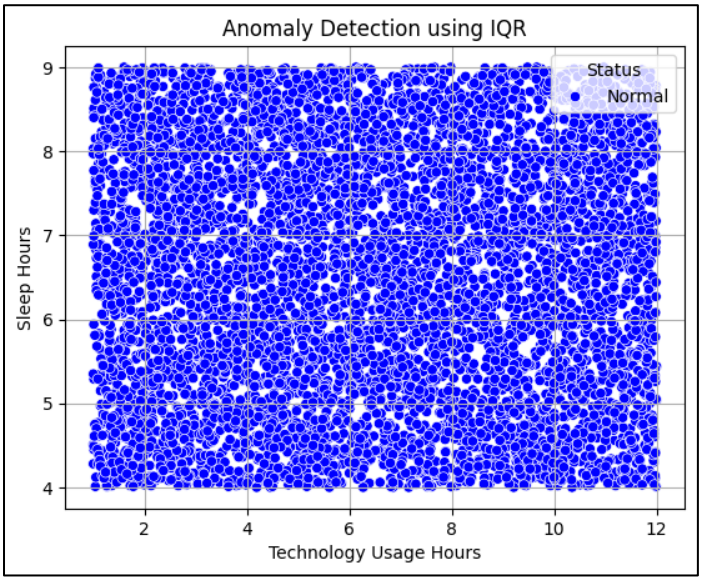

Figure 24: Scatter Plot Visualization for IQR

Machine Learning methods:

Isolation Forest:

We applied the Isolation Forest algorithm to a set of numerical behavioral features: Technology_Usage_Hours, Social_Media_Usage_Hours, Gaming_Hours, Screen_Time_Hours, and Sleep_Hours.

We began by training the model on the original data, where the model learns the patterns of normal data and distinguishes them from abnormal points. The model analyzes the data distribution to detect different patterns and classify points based on their deviation from the common behavior.
For visualization purposes**, we used Principal Component Analysis (PCA)** to reduce the dimensionality to two dimensions, **which helped us visualize the data clearly and differentiate between points more effectively**.

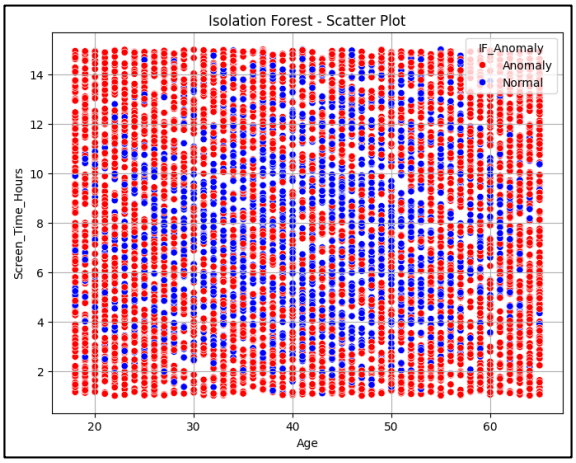

Figure 25: scatter plot visualization before applying PCA

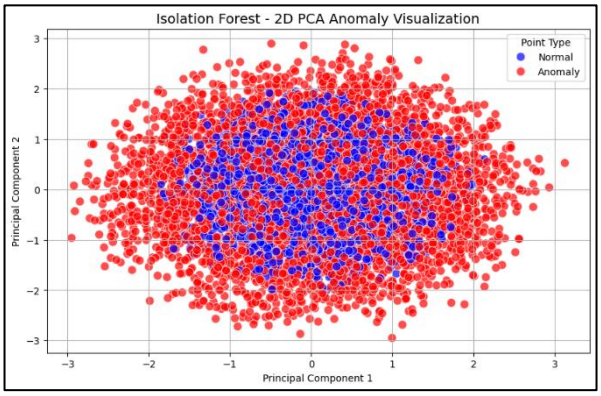

Figure 26: 2D scatter plot visualization after applying PCA.

One-Class SVM:

We applied a **One-Class SVM model** from sklearn to detect anomalies in our dataset, using three input features: **Technology Usage Hours, Sleep Hours, and Stress Level** (numerically encoded). Before modeling, the data was standardized using StandardScaler to ensure fair distance calculations.

We explored different values of nu (0.1, 0.05, and 0.01) in the One-Class SVM model to analyze how the sensitivity to anomalies changes. The parameter nu controls the proportion of data that can be considered outliers.

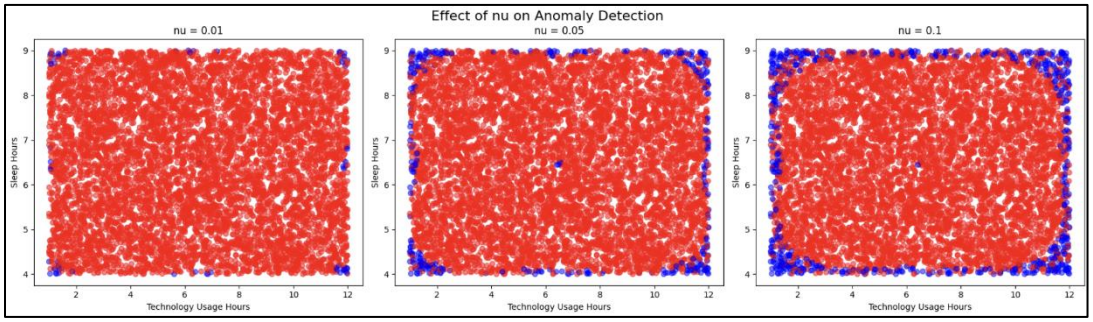

Figure 27: nu Impact on Anomaly Detection

In the visualization, **regular points were shown in red and detected anomalies in blue**.

A higher nu (0.1) resulted in more detected anomalies, including some within the central region of the data. When nu = 0.01, **most anomalies were located only at the edges of the data distribution, and the center appeared normal.**

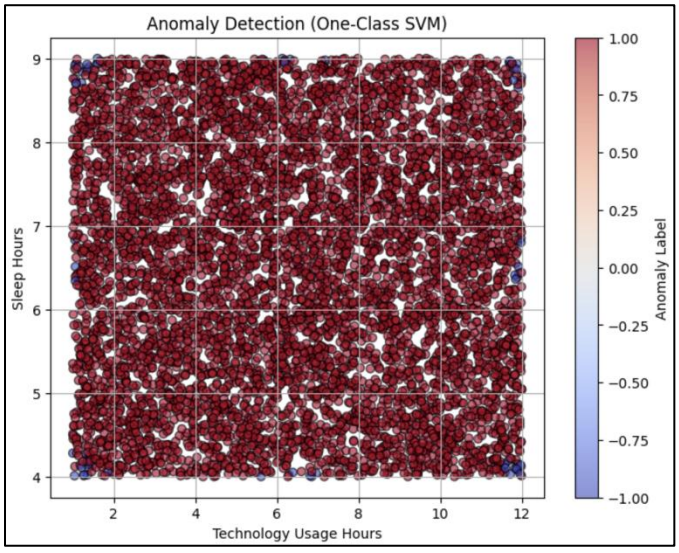

Figure 28: Scatter Plot - Tech & Sleep Hours

LOF (Local Outlier Factor) *Model*:

We applied the LOF algorithm to detect outliers using the following numerical variables: Technology_Usage_Hours, Social_Media_Usage_Hours, Gaming_Hours, Screen_Time_Hours, and Sleep_Hours, and Physical _Activity _Hours. The algorithm compares each point to its neighbors and measures its relative density.

In the first analysis, **without applying PCA**, the data was evaluated in its original high-dimensional form, allowing the model to consider all features in detail and identify points that differ from their immediate surroundings.

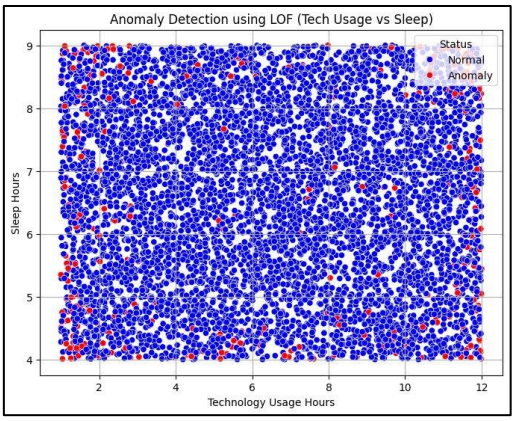

Figure 29: scatter plot visualization before applying PCA.

After applying PCA and reducing dimensionality, it became easier to visualize clusters and relationships between points. Outliers appeared more clearly in simplified representation, although this may slightly reduce detection precision.

Both approaches follow the same principle: points located in areas of lower density compared to their neighbors are considered outliers, with the difference lying in the data representation and level of detail retained.

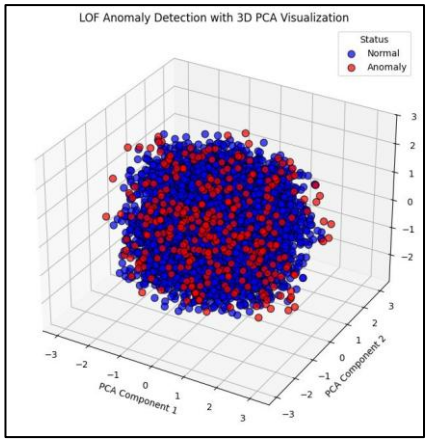

Figure 30: 3D scatter plot visualization after applying PCA.

Results And Discussion

This section interprets the performance and implications of the machine learning models and approaches applied to the dataset on mental health and technology usage.

Regression Analysis:

Linear and nonlinear regression were used to test whether behavioral traits (e.g., screen time, gaming hours) could be predicted from demographic attributes such as age. However, all regression models performed poorly.

(**Note:** *This table aims to clarify the results, not compare the models*.)

Evaluation Techniques	*Simple Linear Regression*		*Simple Nonlinear Regression*		*Multiple Linear Regression*		*Multiple Nonlinear Regression*	
	80%|20%	*K-Fold*	*80%|20%*	*K-Fold*	*80%|20%*	*K-Fold*	*80%|20%*	*K-Fold*
RMSE	*4.0840*	*4.0511*	*0.2919*	*0.2873*	*0.9760*	*0.8064*	*0.2884*	*0.2888*
R^2	*-0.0004*	*-0.0041*	*-0.0022*	*-0.0071*	*-0.4515*	*-0.0001*	*-0.0020*	*-0.0037*
Adjusted R^2	*-0.0012*	*-0.0042*	*-0.0030*	*-0.0079*	*-0.4537*	*-0.0004*	*-0.0127*	*-0.0058*

Table 1: Evaluation Results for all Regression models.

Seen in *Table 1*, the R^2 values were less than zero, and predictions remained flat regardless of the inputs. This indicates that **age does not predict technology usage behavior** like screen time or gaming. Multiple regression models also **failed to meaningfully predict stress level or gaming activity**, suggesting weak linear or nonlinear relationships in the dataset.

The **Simple Linear** plot in *Figure 2*. supports the model's weakness; we can observe the spread of the data, confirming the model's **failure to predict the relationship**.

We can observe from *Table 1*, the RMSE value in **Simple Nonlinear and Multiple Nonlinear** it was somewhat good; however, the plot in *Figure 3* and *Figure 5* shows that the model could not predict correctly and the actual values were spread far from the predicted values, confirming **the model's failure to find a relationship**.

The **Multiple Linear** plot in *Figure 4* shown that the expected pressure levels remain almost constant around the "Medium" level, and none of the "High" level was predicted. This indicates that **the model failed to capture meaningful patterns**, demonstrating poor predictive performance.

These results were consistent across both train-test splits and K-Fold cross-validation, confirming that the low performance was not due to overfitting or data partitioning.

Classification Models:

Model evaluation was conducted using four classifiers:

Decision Tree, Random Forest, Logistic Regression, and Support Vector Classifier (SVC). The models were assessed based on accuracy, precision, recall, F1-score, and confusion matrices.

The confusion matrix for **the Decision Tree** model is presented in *Figure 7*. It displays the model's classification outcomes for the two stress classes: low stress (negative class) and medium/high stress (positive class). The matrix shows that:

- **926** instances were correctly classified as low stress (true negatives),
- **2589** instances were correctly classified as medium/high stress (true positives).

However, the model also made significant misclassifications:

- **1293** low stress cases were incorrectly predicted as medium/high stress (false positives),
- **1827** medium/high stress cases were incorrectly predicted as low stress (false negatives).

This indicates that although the model can detect many true stress cases, it struggles to consistently differentiate between the classes, leading to a moderate overall accuracy of about 48%.

The features that most influenced the model's predictions are illustrated in *Error! Reference source not found.*. Key contributors include Sleep Hours, Gaming Hours, and Social Media Usage Hours, suggesting that lifestyle behaviors have a notable impact on stress classification. Understanding these influential features can help in refining the model and guiding targeted interventions.

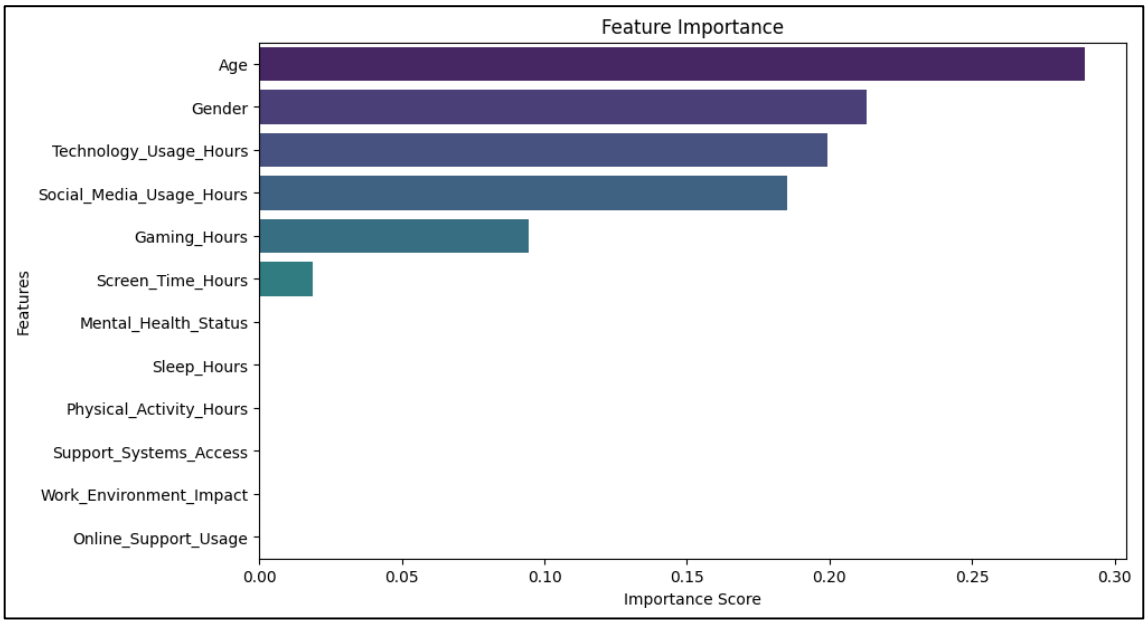

Figure 31 :Feature Importance Bar Chart for Decision Tree

The Random Forest model achieved an overall accuracy of approximately 32% in classifying stress levels into High, Low, and Medium categories, as shown in the **Classification Report in** *Figure 32*. The precision, recall, and F1-score values are fairly balanced across the three classes, each hovering around 0.31 to 0.34.

- For the **High** stress category, precision is 0.32, recall is 0.31, and F1-score is 0.32.
- The **Low** stress category shows slightly better recall (0.34) but similar precision and F1-score (~0.32).
- The **Medium** stress category has the lowest recall (0.29) and an F1-score of 0.31.

These relatively low and close metrics across all classes indicate that the model struggles to distinguish between different stress levels effectively. This could suggest that the current features do not provide sufficient discriminatory power or that the model parameters need further tuning.

The classification report (*Figure 32*) clearly illustrates these limitations, highlighting the need for improved feature selection or advanced modeling techniques to enhance prediction accuracy

	precision	recall	f1-score	support
High	0.32	0.31	0.32	469
Low	0.31	0.34	0.32	405
Medium	0.32	0.29	0.31	453
accuracy			0.32	1327
macro avg	0.32	0.32	0.32	1327
weighted avg	0.32	0.32	0.32	1327

Figure 32: Classification Report Summary of Random Forest

The Logistic Regression model performed moderately better than the Decision Tree, achieving an accuracy close to 51%. The confusion matrix shown in *Figure 10* reflects the model's ability to classify the two stress categories (Low vs. Medium/High) with balanced precision and recall.

- The model correctly classified 1716 instances of the negative class (True Negatives) and 1646 instances of the positive class (True Positives).
- However, there are still a significant number of misclassifications, with 1603 false positives and 1670 false negatives.

This balance between true and false classifications indicates the model maintains moderate performance in distinguishing stress levels but struggles with some overlap between the classes. Overall, the confusion matrix highlights the model's moderate success in classifying stress with reasonable but improvable accuracy.

The Support Vector Classifier (SVC) showed the poorest performance among the models, with an overall accuracy close to 25% across four stress categories. The confusion matrix in *Figure 12* reveals that the model struggles to distinguish clearly between the different classes, as evidenced by a high number of misclassifications distributed relatively evenly across all categories.

- The diagonal values, representing correct classifications, are relatively low compared to the off-diagonal values, which indicate frequent misclassification.
- Precision, recall, and F1-scores remain low and balanced, reflecting the model's limited ability to separate the classes effectively.

These results suggest that the current feature set may not be sufficient for SVC to perform well, and further feature engineering or exploring alternative modeling techniques may be necessary to improve classification performance.

Overall, while none of the models demonstrated high predictive accuracy, the Decision Tree and Logistic Regression models offered relatively better classification results. The Decision Tree model highlighted behavioral variables such as Sleep Hours, Gaming Hours, and Social Media Usage Hours as key contributors to stress prediction. This insight suggests that lifestyle factors play a significant role and can guide future data collection efforts and model refinement.

In contrast, the Random Forest and Support Vector Classifier struggled to effectively distinguish between stress levels, indicating the need for enhanced feature engineering or alternative modeling approaches to improve performance.

Clustering:

Clustering analysis was conducted using both **K-Means** and **Hierarchical Agglomerative Clustering**, each with two clusters (as adding more reduced silhouette scores).

Model	*Silhouette Score*
K-Means Clustering	*0.106*
Hierarchical Clustering	*0.064*

Table 2:Evaluation scores for both clustering methods.

Table 2 shows the scores of our models which indicates that K-means model is better than Hierarchical, albeit the difference is insignificant. This alone doesn't help us with our goals, so we moved to finding what differs between the two clusters in each model.

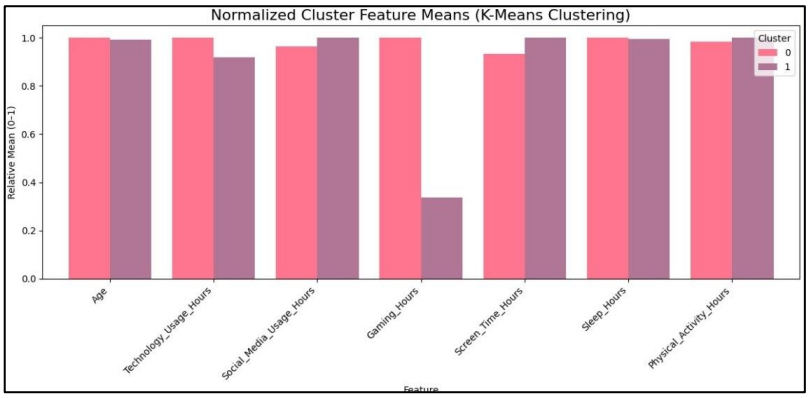

Figure 33: Visualization of how the two clusters were divided using k-means.

K-Means clustering showed slightly better visual separation, but, as Table 2 shows, the **silhouette score was only 0.106**, indicating weak cluster quality.

As shown in *Figure 33*, we calculated the average of each numeric feature for both K-Means clusters to explore what differentiates them. The results indicate that:

- **Cluster 0** users had significantly higher average gaming hours compared to Cluster 1, suggesting that gaming behavior is a key factor separating the groups.
- **Cluster 1** users showed slightly higher screen time and marginally lower gaming activity, but the differences were relatively small.
- Other features such as age, sleep hours, technology usage, and physical activity were nearly identical between clusters, showing minimal influence on the grouping.

- The clustering results imply that, despite using several features, the K-Means algorithm primarily separated individuals based on **gaming behavior**, with limited contribution from other variables.

Hierarchical clustering, despite a lower silhouette score of **0.064**, captured **more interpretable groupings**.

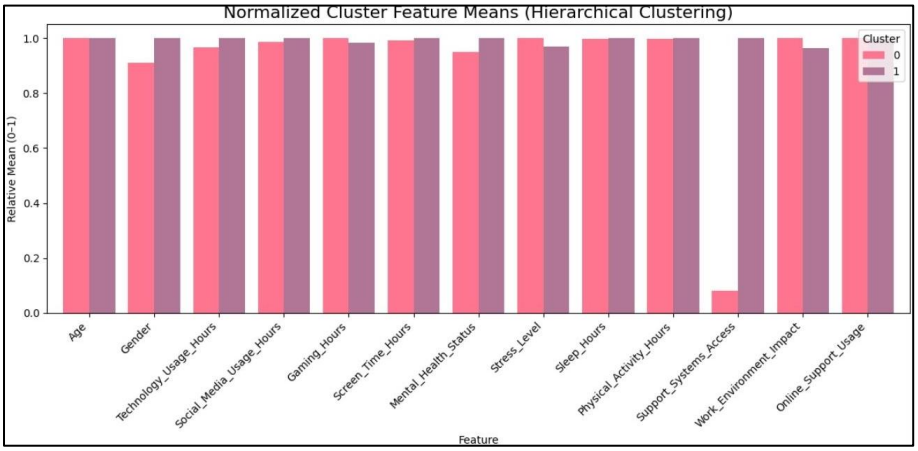

Figure 34: Visualization of how the two clusters were divided using Hierarchical clustering.

Similarly, we calculated the average of each numeric feature for both clusters to understand what differentiates them. Using *Figure 34*, we concluded that the results suggest that:

- **Cluster 1** users reported **slightly higher technology and social media usage**, **noticeably more access to support systems**, and **better mental health scores**.
- **Cluster 0** users had **less support system access** and **lower mental health status** on average.
- Other features, such as **sleep hours**, **screen time**, and even **stress level**, were **very close** between clusters, showing only minor differences.
- **Gender and age distribution** between clusters were nearly balanced, indicating it does not meaningfully distinguish the groups.

According to Ahmad and Dey (2007), traditional K-Means clustering does not handle categorical features well, making it less suitable for our analysis, where such variables are crucial (e.g., support systems, work environment, mental health status). Hierarchical clustering, on the other hand, allowed us to include both numerical and categorical data, resulting in more meaningful clusters that aligned with our project goals, even if the silhouette score was lower.

Anomaly Detection:

Statistical and model-based anomaly detection techniques were applied, including **Z-score**, **IQR**, **Local Outlier Factor**, **One-Class SVM**, and **Isolation Forest**. Although Statistical methods showed no anomalies, it was because they **detect outliers purely based on numerical deviation** but not based on **contextual or relational anomalies**. Machine learning methods identified individuals with unusually high or low values in behavioral features. Choosing key features for detection (e.g., screen time vs. sleep) helped reveal outliers meaningfully. However, the anomalies were not directly tied to stress or mental health labels, so **their relevance remained exploratory.**

The **Z-Score** analysis in *Figure 23* highlights the absence of significant anomalies; the data points are evenly distributed within the expected range, **confirming that no outliers were detected by the model**. This indicates that the data approximately follows a normal distribution without any extreme points or unusual patterns across the selected numerical variables. The reason for this could be that the samples are relatively homogeneous, or that outliers are very rare or nonexistent in this dataset, **making the Z-Score model unable to identify any substantial deviations from the mean.**

The **IQR** analysis in *Figure 24* demonstrates the absence of notable outliers; data points are well-distributed within the interquartile range, confirming that no anomalies were detected by the model. This suggests that the data is consistently spread without extreme values or unusual patterns across the selected numerical variables. It may indicate that the dataset is relatively homogeneous, or that any potential outliers are minimal or nonexistent, **thus the IQR method did not identify significant deviations beyond the acceptable range.**

Isolation Forest -When applying the Isolation Forest algorithm without PCA *Figure 25*, the model analyzed the original behavioral variables and identified a large number of anomalies. However, these anomalies were scattered randomly across the high-dimensional space, with no clear structure or clustering. This suggests that either **the selected features may not be ideal for this type of model**, or **that further feature engineering is needed to better capture meaningful patterns**.

After applying PCA for dimensionality reduction *Figure 26*, the anomalies became more visually distinct, separating from the dense cluster of normal points. While this improved interpretability, the scattered nature of the anomalies remained, reinforcing the idea that the model may have struggled to find consistent relationships in the input space.
These observations highlight that Isolation Forest can detect outliers, **but its effectiveness strongly depends on the relevance and quality of the features used.**

When applying the **One-Class SVM model** without dimensionality reduction *Figure 27*, the model analyzed the **numerical features** using a low nu value (0.01), which makes the model strict in identifying anomalies. A limited number of anomalies were detected, mostly concentrated at the edges of the data distribution. Normal data points appeared clustered in the center in red, while the anomalies were scattered around the periphery in blue.
After applying PCA for dimensionality reduction and visualization *Figure 28*, the differences between normal and anomalous points became easier to distinguish visually. However, the scattered distribution of anomalies around the edges remained noticeable, indicating that the model preserved the same interpretation pattern in both the original space and the reduced-dimensional space.

This distribution suggests that the One-Class SVM shows high sensitivity to boundary values, especially when using low nu settings. This means that the model is capable of isolating abnormal values.

When applying the **Local Outlier Factor (LOF) model** without dimensionality reduction *Figure 29*, the model analyzed the numerical features in their original high-dimensional space. Anomalies were detected based on local density deviations, where points with significantly lower density compared to their neighbors were considered outliers. **The results showed that normal data points formed dense clusters, while outliers appeared more sparsely distributed.**

After applying PCA for dimensionality reduction *Figure 30*, the anomalous points became more visually distinguishable in the simplified 3D space. Although PCA may slightly reduce precision, the separation between normal and outlier points was clearly visible, making interpretation easier.

This demonstrates that LOF is effective in identifying local deviations both in the original feature space and after dimensionality reduction. The clear visualization of outliers in the PCA-transformed space highlights the model's ability to isolate abnormal patterns while maintaining a **good balance between interpretability and accuracy.**

Limitations:

Several limitations affected the quality and interpretability of the results:

- **Complex and resistant dataset**
  The dataset presented challenges in uncovering strong, consistent patterns due to overlapping variables and subtle inter-group differences. Additionally, community feedback on the original Kaggle source has raised concerns that the data may be randomly generated, which could further limit the reliability and depth of insights.
- **Confounding variables**
  Important factors affecting outcomes may not have been captured, reducing model accuracy and interpretability.
- **Balanced use of subjective interpretation**
  Although analysts' insights are helpful, we had to have them managed carefully to avoid over-interpreting weak or noisy signals.
- **Limited scope for experimentation**
  While many methods were explored, time and structure constraints limited deeper tuning or trying more complex techniques.
- **Model performance was modest at best**
  Most models produced weak to moderate results, so conclusions should be seen as exploratory, not definitive.

Overall Interpretation:

Regression showed **no meaningful relationships between age and digital behavior**, and Hierarchical clustering methods revealed that individuals with **less support access and lower mental health scores were more likely to be stressed**. Our analysis suggests that the dataset may be **missing important confounding variables** influencing mental health, **preventing us from confidently determining the impact of screen usage on mental health.**

These findings, though not conclusive, offer tentative insights into how digital behavior and support systems relate to stress and mental wellbeing.

CONCLUSION

This project sets out to explore the relationship between technology usage and mental health using various machine learning techniques on a dataset sourced from Kaggle (waqi786, 2020).

We applied regression, classification, clustering, and anomaly detection models to investigate patterns and draw meaningful insights.

Despite our efforts, the results revealed several limitations. **Regression models showed no significant predictive power**; age could not meaningfully predict technology usage or stress-related outcomes. **Classification models achieved only modest accuracy**. These outcomes suggest that while some behavioral and lifestyle features may relate to stress, they are insufficient for reliable prediction, likely due to missing or confounding variables not captured in the dataset.

Clustering analysis offered more interpretable insights. K-Means clustering primarily separated users by gaming behavior, while Hierarchical Clustering revealed more meaningful groupings based on support system access and mental health status, despite a lower silhouette score. This highlighted the value of including categorical features in unsupervised learning when examining mental health patterns.

Anomaly detection techniques surfaced unusual patterns in behavioral traits, though these anomalies were not directly linked to stress or mental health, limiting their interpretive value.

Overall, our findings emphasized the complexity of predicting mental health outcomes. **The dataset's limitations**, including potential randomness in data generation and the absence of crucial confounding variables, **significantly impacted model performance**. Consequently, **interpretations should be considered exploratory rather than conclusive.**

Future work should involve **collecting more reliable and representative data**, engineering more informative features, tuning models more deeply, and **experimenting with advanced algorithms.** Including variables such as chronic illness, socioeconomic background, and personality traits could improve the model's ability to capture the multifaceted nature of mental health. Despite its challenges, **this analysis contributes to the growing effort behind understanding how digital behaviors intersect with psychological wellbeing.**

REFERENCES

- Google. (n.d.). *Google Colaboratory*. Mental Health &Tech Usage Analysis Notebook.
- waqi786. (2020). **Mental health and technology usage dataset** [Data set]. Kaggle. https://www.kaggle.com/datasets/waqi786/mental-health-and-technology-usage-dataset/data
- **Ahmad, A., & Dey, L. (2007).** A k-means clustering algorithm for mixed numeric and categorical data. *Data & Knowledge Engineering, 63*(2), 503–527. https://doi.org/10.1016/j.datak.2007.03.002
