Ethar Mostafa 22-101121 Noureen Ibrahim 22-101056 Kareem ElKenany 22-101031
Arwa Mostafa 22-101120 Rana Wael 22-101219 Abdelrahman Mahdi 22-101173

Mobile Addiction and Stress Level

Abstract

This project looks at how stress levels might be linked to mobile phone addiction among students. We used two datasets: one focusing on general user behavior, and the other on mobile addiction indicators. The main goal was to see if students who experience more stress are more likely to be addicted to their phones, and whether stress could help predict phone addiction. We used data visualization, regression, and classification models to explore the data. Our results showed strong patterns, students with higher stress also had higher screen time, used apps more frequently, and got more notifications. The classification models, especially Random Forest and Naïve Bayes, were able to predict addiction status with very high accuracy. Regression models like Random Forest also gave solid predictions for daily screen time. Overall, the findings suggest a real connection between stress and phone use. Based on this, we recommend encouraging healthier phone habits and stress management techniques, especially for students who rely heavily on their phones.

Introduction

Background and Context

These days, almost everyone uses a smartphone for communication, entertainment, or staying connected to the world. While phones are useful, using them too much can become a problem, developing something known as mobile phone addiction. This happens when someone feels like they can't stop using their phone, even if it's affecting their daily life, school, or mental health. At the same time, stress has become a big issue, especially for students. School work, exams, social pressure, and other challenges can get overwhelming. Since both stress and phone addiction are common among students, we started wondering, could they be related? That's what our project is about.

Literature Review

Previous research suggests there might be a link between stress and mobile phone addiction. For example, one study conducted by Gao et al. (2022) found that students in a sample of Chinese college students who reported higher stress levels also tended to use their phones more in ways that could be harmful or addictive. Another study conducted by Jeong et al. (2022) showed that stress can lower self-control, which might make people more likely to spend extra time on their phones to escape or feel better temporarily. These findings made us think that students might use their phones to cope with stress, but that could lead to unhealthy habits.

Project Objectives

In our project, we want to find out if stress levels are connected to mobile phone addiction. Our main goals are:

1. To see if students who feel more stressed are more likely to show signs of mobile phone addiction.

2. To understand whether stress could be used as a predictor of phone addiction in students.

By the end of this project, we hope to better understand how stress and phone use are connected. This could help raise awareness and maybe even lead to ideas for how to manage stress and phone habits in healthier ways.

Materials and Methods

Data Description

In this project we will be using 2 datasets:

1. User behavior dataset
2. Mobile addiction dataset

Dataset 1: user behavior dataset

- **Source of Data:**
  This dataset was sourced from Kaggle and includes behavioral data for 700 smartphone users.
- **Key Variables and Types:**
  The dataset includes both numerical and categorical variables. Some of the key variables are:
  - Numerical: App usage time (minutes), Screen on time (hours), Battery drain (mAh), Data usage (MB), Number of apps installed, Age.
  - Categorical: Device model, Operating system, Gender, User behavior class.
- **Data Size and Structure:**
  The dataset contains 700 rows and multiple columns representing device details, user demographics, and phone usage statistics. It is well-structured with no missing or duplicate values.
- **Data Cleaning and Preprocessing:**
  - No missing or duplicate data was found.
  - Descriptive statistics (mean, standard deviation, quartiles) were calculated for all numerical variables.
  - Outlier detection was performed using the IQR method, but no significant outliers were identified.
  - The data showed logical consistency (e.g., screen time > app usage time).
- **Will be used for Regression Models**

Dataset 2: mobile addiction dataset

- **Source of Data:**
  This dataset was also sourced from Kaggle and includes detailed behavioral data for 13,600 participants (reduced to 13,331 after preprocessing).

- **Key Variables and Types:**
  The dataset consists of mostly numerical variables. Some of the key variables are:
  - Numerical: Daily screen time (hours), App sessions, social media usage (hours), Gaming time (hours), Notifications received, Night usage (hours), Age, Work/study hours, Stress level (on a scale), Apps installed.
  - Categorical: Addicted (Addicted/Not Addicted)
- **Data Size and Structure:**
  The dataset originally had 13,600 records, with each row representing a participant. After removing outliers, the final size was 13,331.
- **Data Cleaning and Preprocessing:**
  - Checked for and confirmed no null or duplicate values.
  - Converted the Addicted column to a categorical variable.
  - Performed statistical analysis (mean, standard deviation, min, max) for all numerical columns.
  - Detected and removed 271 outliers using the Z-test method.
  - Conducted the Anderson-Darling test, which showed the data does not follow a normal distribution.
- **Will be used for the Classification Models**

Methodology

- **Data Visualization**
  To better understand the trends, patterns, and relationships within our datasets, we relied on data visualization techniques using Matplotlib and Seaborn. These tools helped us explore both the User Behavior dataset and the Mobile Addiction dataset.
  We chose different types of plots depending on what kind of data we were working with and the information we wanted to gain:
  - **Correlation Heatmaps:** were used in both datasets to check how strongly different numerical variables relate to each other. This was particularly useful in identifying patterns between usage time, data consumption, and other behavioral indicators.
  - **Histograms:** helped us understand the distribution of continuous variables like app usage time, screen time, night-time usage, and age.
  - **Bar Charts:** were used to compare average values (like app usage, screen-on time, and stress levels) across categories such as operating systems, behavior classes, age groups, and addiction status.
  - **Pie Charts:** offered a quick visual of the proportions in categorical variables, such as gender distribution or operating system share.
  - **Grouped Bar Plots:** allowed us to compare multiple metrics (e.g., data usage, battery drain, app usage) across behavior classes in a single graph.
  - **Scatter Plots**: were applied to visualize relationships between two numeric variables, for instance, app usage time vs data usage or notifications vs stress level.

These visualizations guided the rest of our analysis by pointing out which variables were worth focusing on.

- **Regression**

We used six regression models on the user behavior dataset:

1. **Linear Regression**

How it works:

Linear Regression models the relationship between the input features and the target variable by fitting a straight line (or hyperplane) that minimizes the sum of squared residuals (errors between actual and predicted values). It assumes linear relationships between independent variables and the dependent variable.

Why chosen:

- Fast and simple to implement.
- Provides a clear baseline for performance comparison.

2. **Ridge Regression**

How it works:

Ridge Regression extends linear regression by adding L2 regularization, which penalizes the sum of the squared coefficients. This discourages large weights and helps prevent overfitting, especially in the presence of multicollinearity (when features are highly correlated).

Why chosen:

- Addresses multicollinearity effectively.
- Maintains all features while reducing model complexity.
- Useful when interpretability is important, but feature selection is not necessary.

3. **Lasso Regression**

How it works:

Lasso regression uses L1 regularization to minimize the sum of squared errors while penalizing the absolute values of coefficients. This encourages sparsity, effectively performing feature selection by shrinking some coefficients to exactly zero.

Why chosen:

- Suitable for datasets with many correlated features.
- Helps reduce overfitting and improves interpretability.
- Acts as a strong baseline linear model.

4. **Elastic Net Regression**

   How it works:

   Elastic Net combines L1 (Lasso) and L2 (Ridge) regularization in a single model. This allows it to inherit both sparsity (from L1) and coefficient stability (from L2), making it more flexible in handling correlated features.

   Why chosen:

   - Better suited for correlated features than Lasso alone.
   - Provides stable and generalizable predictions.

5. **XGBoost Regression**

   How it works:

   XGBoost is a gradient boosting algorithm that builds an ensemble of decision trees sequentially. Each tree is trained to correct the errors of its predecessor by minimizing a regularized objective function. It incorporates both L1 and L2 regularization and uses advanced techniques like tree pruning, column subsampling, and parallelization.

   Why Chosen:

   - Powerful and scalable model.
   - Captures complex patterns and interactions.
   - Handles missing values and regularizes well.

6. **Random Forest Regression**

   How it works:

   Random Forest builds multiple decision trees using bootstrapped subsets of the data and averages their predictions to improve generalization. It introduces randomness by selecting a random subset of features at each split, which reduces variance and overfitting.

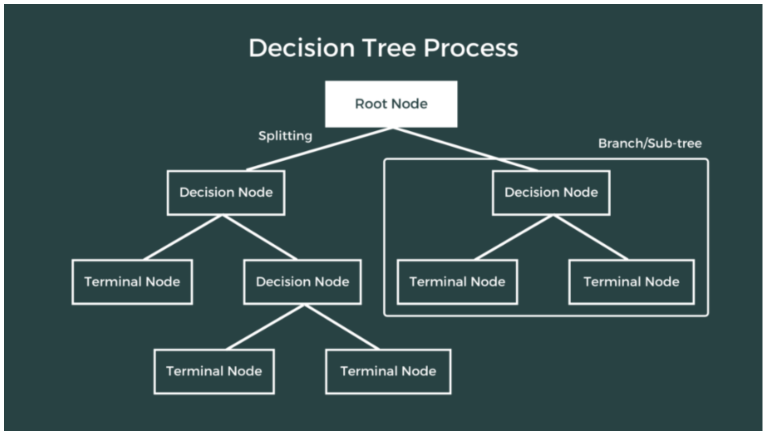
   Why Chosen:

   - Handles nonlinearity and feature interactions well.
   - Robust to multicollinearity and overfitting.
   - Effective on moderately sized datasets.

7. **Decision Tree Regression**

   Decision Tree Regression splits the data into subsets based on feature values that minimize the variance of the target variable within each split. The process continues recursively, creating a tree where each internal node represents a decision rule and each leaf node represents a predicted value. The model makes predictions by traversing the tree from root to leaf based on the input features.

Ethar Mostafa 22-101121 Noureen Ibrahim 22-101056 Kareem ElKenany 22-101031
Arwa Mostafa 22-101120 Rana Wael 22-101219 Abdelrahman Mahdi 22-101173



Why Chosen:

- Simple and interpretable model.
- Captures nonlinear relationships and interactions.
- Useful as a baseline for tree-based ensemble methods.

To Evaluate our Regression Models, We Used:

- MSE
- R^2

- **Classification**

  We used four classification models:

  - **Gaussian Naïve Bayes**

    Justification for Choosing Naïve Bayes

    I chose to use the Naïve Bayes classifier because it's a simple and powerful algorithm, especially for binary classification problems like predicting whether a user is addicted or not. The dataset included several numerical features such daily screen time, app sessions, gaming time, which are fit for the Gaussian Naive Bayes variant since it handles continuous data under the assumption of normal distribution.

Ethar Mostafa 22-101121 Noureen Ibrahim 22-101056 Kareem ElKenany 22-101031
Arwa Mostafa 22-101120 Rana Wael 22-101219 Abdelrahman Mahdi 22-101173

- **K-Nearest Neighbors**

  Justification for Choosing KNN

  I chose KNN because it's a simple algorithm that makes predictions based on how similar (close) the data points are across all selected features by calculating the Euclidean distance. It is non-parametric which means it doesn't assume any specific distribution of the data. Since my dataset (mobile_addiction_cleaned.csv) is made up of clean numerical features (e.g. daily screen time, app sessions, notifications), it fits KNN very well because these types of models work best with numerical, scaled data. Also, KNN is sensitive to the structure of the data, so it was a good candidate for exploring whether natural clusters of "addicted" vs "not addicted" users exist based on mobile usage behavior.

- **Decision Tree Classifier**

  Justification for Choosing Decision Tree Classifier

  I applied a Decision Tree Classifier to the mobile addiction dataset to explore how well it can distinguish between "addicted" and "not addicted" users based on various behavioral features. A decision tree is a supervised machine learning algorithm that splits the dataset into subsets based on feature values, creating a tree-like structure where each parent node represents a feature decision, each branch represents an outcome, and each leaf node represents a class label (in this case, not addicted = 0 or addicted = 1).

  Why Decision Tree?

  I chose a Decision Tree because:

  - It visually represents how decisions are made.
  - It is non-parametric meaning it makes no assumptions about data distribution, which is ideal given that many of the features in the dataset are not normally distributed.
  - It is effective in modelling complex feature interactions like those observed between social media use, notifications, and stress level.

  The core concept behind a decision tree is selecting the best feature or best split to split the data at each node. This is done using impurity measures like Information Gain (Entropy). For example, Entropy is calculated as:

$$Entropy = \sum - p_i \ \log(p_i)$$

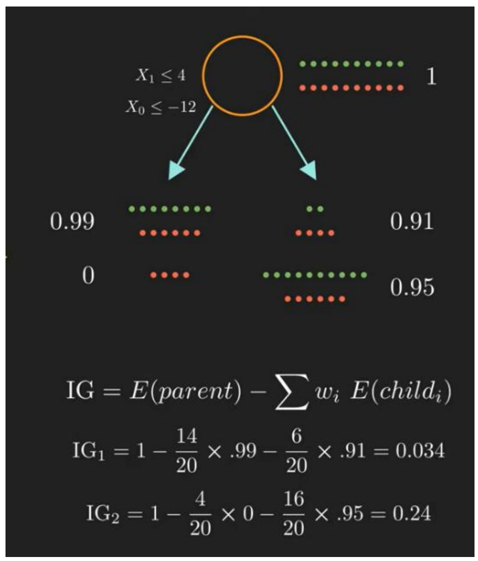
  where p_i is the probability of class i. The algorithm calculates the entropy for each node (parent and children) then calculates the information gain using the following equation:

$$\mathrm{IG} = E(parent) - \sum w_i \ E(child_i)$$

where w_i is the weight of the i_{th} child, it is calculated by:

$$\text{weight}(\text{child}_i) = \frac{\text{number of samples in child}_i}{\text{number of samples in parent}}$$

The algorithm tries every possible split and chooses the split with the highest information gain. NOTE: a "pure" state (all samples belong to one class) has an entropy of 0, while an entropy of 1 indicates equal probabilities of each class (in binary classification, each class would have a probability of 0.5)



- ### Random Forest Classifier

  Justification for Choosing Random Forest

  I used the Random Forest classifier because it works well with structured data and is especially effective when the relationship between features and the target variable is non-linear or complex. Unlike single decision trees, which are prone to overfitting, Random Forest builds multiple decision trees and combines their outputs, improving accuracy and reducing variance.

  The dataset contains a mix of numerical features with varying levels of correlation. This makes Random Forest a suitable choice because it can naturally handle feature interactions and is non-parametric meaning it does not assume feature independence or any specific distribution.

How it Works

The Random Forest Classifier is an optimized version of decision trees that addresses their limitations, such as overfitting and sensitivity to training data. Rather than relying on a single decision tree, a Random Forest constructs a collection (or "forest") of decision trees, each trained on a different subset of the data.

Its steps are:

1. The algorithm generates multiple training datasets by randomly sampling from the original dataset with replacement (bootstrapping). Each of these datasets may contain duplicate samples and will be used to train a separate decision tree.
2. When constructing each tree, the algorithm selects a random subset of features (typically log2 or sqrt of the total number of features) at each split, rather than using all available features. This introduces additional diversity among the trees and reduces correlation between them.
3. Each decision tree is independently trained on its respective bootstrapped dataset using the selected random features.
4. When making a prediction, the input data is passed through all trees in the forest. Each tree produces a classification result, and the final prediction is determined by the majority outcome, the most common class among all trees.

To Evaluate our Classification Models, we used:

- Accuracy score
- F1 score
- Confusion Matrix
- AUC score
- Cross Validation AUC score

Results

- **Key Findings**
  - **Data Visualization**

Key Findings from the User Behavior Dataset

1. **Correlation Heatmap (Refer to Fig 1.7):** showed a strong positive relationship between app usage time, data usage, and battery drain, suggesting that users who spend more time on their phones also use more data and drain their batteries faster.
2. **Histogram of App Usage Time (Refer to Fig 1.6):** revealed a right-skewed distribution. Most users had moderate usage, but a small number were heavy users.

3. **Bar Chart (OS-wise Data Usage) (Refer to Fig 1.5):** showed that Android users, on average, consumed more data than iOS users.
4. **Pie Chart (Gender Distribution) (Refer to Fig 1.8):** indicated that the dataset had a balanced gender distribution, with a slight male majority.
5. **Bar Chart (Behavior Class) (Refer to Fig 1.9):** highlighted that most users fell into Classes 2 and 3, which might indicate moderate usage behavior, while fewer users were in extreme behavior classes like 1 and 5.
6. **Bar Chart (Average App Usage by Age Group) (Refer to Fig 1.3):** showed that the 18–25 age group had the highest app usage, and usage declined steadily with older age groups.
7. **Pie Chart (Operating System Distribution) (Refer to Fig 1.2):** confirmed that most users in the dataset were on Android.
8. **Grouped Bar Plot (Metrics by Behavior Class) (Refer to Fig 1.4):** demonstrated that users in higher behavior classes (especially Class 5) used their phones more intensively, showing higher app usage, data consumption, and battery drain.
9. **Bar Chart (Screen-On Time by Gender) (Refer to Fig 1.1):** suggested that males generally had slightly longer screen-on times, although the difference wasn't very large.

Key Findings from the Mobile Addiction Dataset

1. **Histogram (Daily Screen Time) (Refer to Fig 1.10)**: showed that while many users had moderate screen time, a noticeable group had very high daily usage, hinting at possible overuse or addiction.
2. **Scatter Plot (Notifications vs Stress Level) (Refer to Fig 1.11):** revealed a positive trend, more notifications seemed to be linked with higher stress levels.
3. **Correlation Heatmap (Refer to Fig 1.12):** showed strong relationships among screen time, number of notifications, and app sessions, with stress level showing a moderate correlation as well.
4. **Bar Plot (Social Media Usage by Addiction Status) (Refer to Fig 1.15):** showed that users classified as addicted spent significantly more time on social media.
5. **Pie Chart (Addiction Class Distribution) (Refer to Fig 1.18):** indicated that the dataset was almost balanced in terms of addicted vs non-addicted users.
6. **Scatter Plot (Apps Installed vs Social Media Usage) (Refer to Fig 1.19):** suggested a weak positive relationship, meaning users with more apps installed might spend slightly more time on social media.
7. **Bar Plot (Work/Study Hours by Addiction Status) (Refer to Fig 1.14):** showed that non-addicted users tended to report more time spent on productive activities.
8. **Scatter Plot (Daily Screen Time vs Stress Level) (Refer to Fig 1.20):** showed that higher screen time was associated with increased stress levels.

9. **Scatter Plot (App Sessions vs Notifications) (Refer to Fig 1.17):** revealed a strong positive relationship, users who opened apps more often also received more notifications.
10. **Histogram (Age Distribution) (Refer to Fig 1.13):** indicated that most users were young adults, especially in their early 20s.
11. **Histogram (Night Usage) (Refer to Fig 1.16):** showed a noticeable number of users who used their phones late at night, suggesting possible sleep disruption.
12. **Bar Plot (Stress Level by Addiction Status) (Refer to Fig 1.21):** highlighted that addicted users had higher average stress levels.
13. **Bar Plot (Gaming Time by Addiction Status) (Refer to Fig 1.22):** showed that addicted users spend more time gaming on average.
14. **Scatter Plot (Night Usage vs Notifications) (Refer to Fig 1.23):** revealed that people who used their phones more at night also tended to get more notifications, which may contribute to late-night screen time habits.

- **Regression**

Our regression models aimed to predict the daily screentime of each user based on their mobile phone usage behavior. Below are the key findings from each model:

1. **Linear Regression:**
   R^2 Score: 0.93189
   MSE: 0.61109
   Observations:
   - Strong baseline performance due to highly linear relationships between features and target.

2. **Ridge Regression:**
   R^2 Score: 0.93181
   MSE: 0.611849
   Observations:
   - Very similar to Linear Regression with improved stability.

3. **Lasso Regression:**
   R^2 Score: 0.92896
   MSE: 0.63736
   Observations:
   - Slightly lower performance.
   - L1 regularization might have removed some mildly important features.
   - Indicates possible slight sparsity in the dataset.

4. **Elastic Net Regression**
   R^2 Score: 0.9253

MSE: 0.6701

Observations:

- Lowest among the linear models.
- Suggests that neither strong sparsity nor multicollinearity correction was essential.

5. **XGBoost Regression**

   R^2 Score: 0.92117

   MSE: 0.7073

   Observations:

   - Very high performance.
   - Slightly below Random Forest, likely due to sensitivity to hyperparameters.
   - Regularization and boosting improved generalization.

6. **Random Forest Regression**

   R^2 Score: 0.94123

   MSE: 0.5273

   Observations:

   - Best performer overall.
   - Captured non-linear patterns and handled interactions effectively.
   - Robust to noise and overfitting due to ensemble averaging.

7. **Decision Tree Regression**

   R^2 Score: 0.88372

   MSE: 1.04336

   Observations:

   - Performed the worst among all models.
   - Likely overfitted due to lack of ensemble averaging.
   - Struggled with capturing strong linear trends in the data.

- **Classification**

  Our classification task aimed to predict whether a user is "Addicted" or "Not Addicted" based on mobile phone usage behavior. Across several classification models, we found consistently high accuracy, balanced F1-scores, and strong AUC values. Below are the key findings from each model:

  1. **Naïve Bayes:**
     - **Training Accuracy:** 98%
     - **Testing Accuracy:** 98%

- **F1 Score:**
  - Addicted: 0.97
  - Not Addicted: 0.97
- **AUC Score:** 1.0
- **Observations:**
  - Low misclassification in the confusion matrix (Figure 3.2).
  - Training and testing accuracy are close, indicating a balanced model.
  - Most features followed near-normal distributions and had low to moderate correlations (Figure 1.14), aligning well with Naive Bayes assumptions.

2. **KNN:**

- **Training Accuracy:** 98%
- **Testing Accuracy:** 97%
- **F1 Score:**
  - Addicted: 0.97
  - Not Addicted: 0.97
- **AUC Score:** 0.99
- **Observations:**
  - The confusion matrix (Figure 3.4) confirmed few misclassifications in either class.
  - ROC curve (Figure 3.3) showed near-perfect separation between classes.
  - Strong feature correlations (e.g., notifications: –0.79, app_sessions: –0.67, stress_level: –0.54) helped the model achieve high precision and recall.
  - StandardScaler was applied to ensure all features contributed equally, as KNN is sensitive to feature scaling.

3. **Decision Tree:**

- **Training Accuracy:** 100%
- **Testing Accuracy:** 95%
- **F1 Score:**
  - Addicted: 0.95
  - Not Addicted: 0.95
- **AUC Score:** 0.95
- **Observations:**
  - Maintained consistent performance across 5-fold cross-validation (mean AUC: 0.956), indicating good generalization.
  - Feature correlations supported effective decision splitting.
  - Slightly less accurate than KNN, possibly due to Decision Trees being more prone to overfitting on minor noise.

Ethar Mostafa 22-101121 Noureen Ibrahim 22-101056 Kareem ElKenany 22-101031
Arwa Mostafa 22-101120 Rana Wael 22-101219 Abdelrahman Mahdi 22-101173

4. **Random Forest:**
- **Training Accuracy:** 100%
- **Testing Accuracy:** 98%
- **F1 Score:**
  - Addicted: 0.98
  - Not Addicted: 0.98
- **AUC Score:** 1.0
- **Observations:**
  - Very few classification errors in the confusion matrix (Figure 3.8).
  - The ROC curve (Figure 3.7) shows perfect separation.
  - Despite 100% training accuracy, 5-fold cross-validation shows stable validation performance, suggesting only slight overfitting.
  - Strong, structured features with clear boundaries made Random Forest highly effective.

- **Visualizations**
  - **Data Visualization**
    1. **User Behavior Dataset**

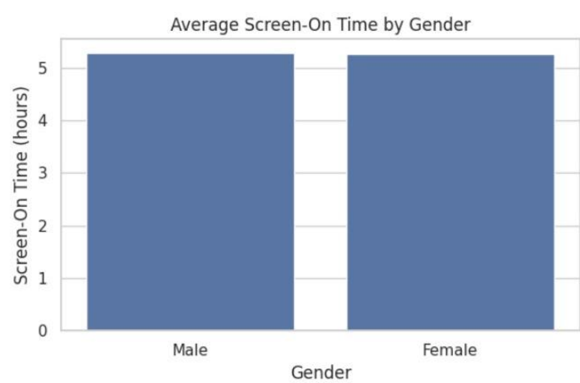


Fig 1.1 Average Screen-On Time by Gender

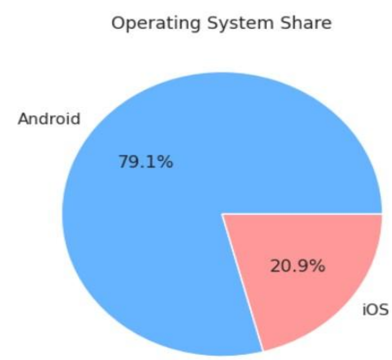


Fig 1.2 Operating System Distribution

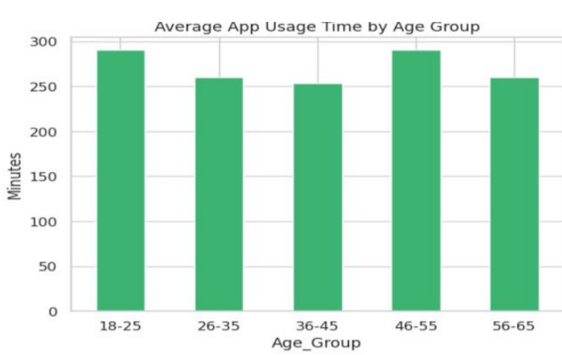


Fig 1.3 App Usage Time by Age Group

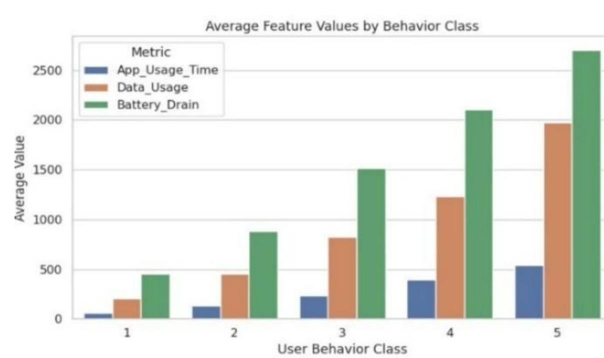


Fig 1.4 Average Feature Values by Behavior Class

Ethar Mostafa 22-101121 Noureen Ibrahim 22-101056 Kareem ElKenany 22-101031
Arwa Mostafa 22-101120 Rana Wael 22-101219 Abdelrahman Mahdi 22-101173

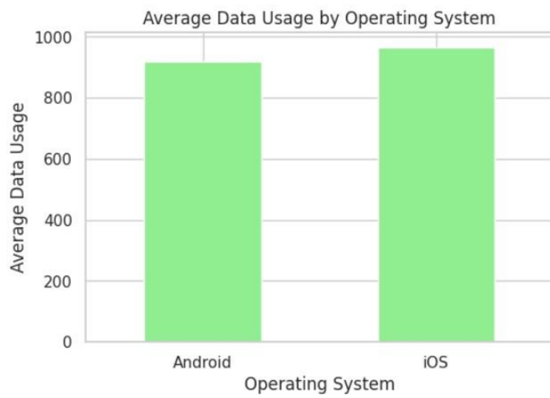


Fig 1.5 Average Data Usage by Operating System

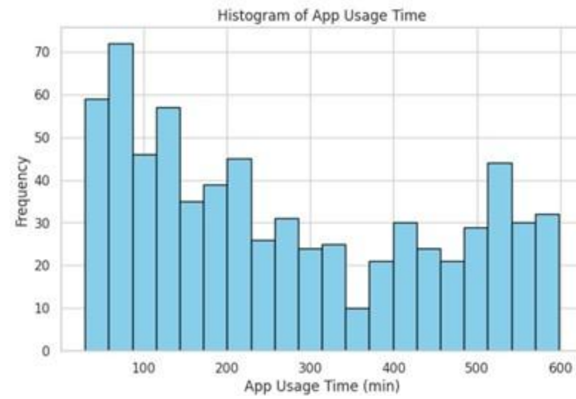


Fig 1.6 Histogram of App Usage Time

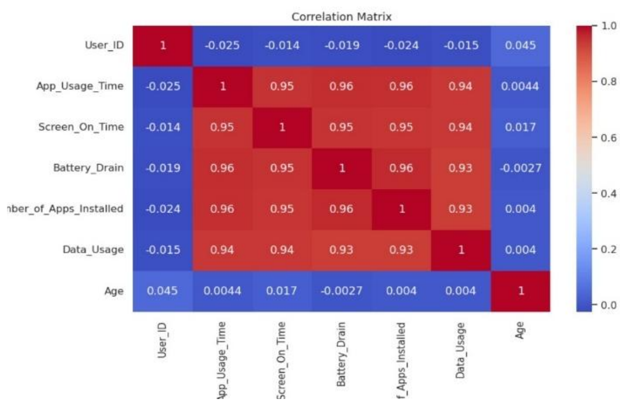


Fig 1.7 Correlation Matrix

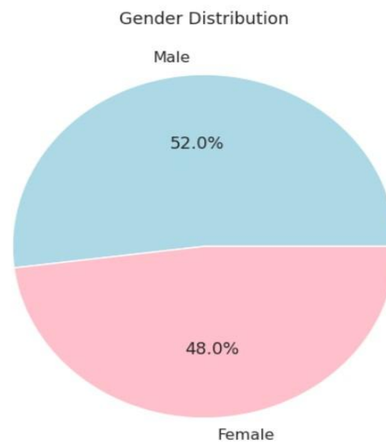


Fig 1.8 Gender Distribution

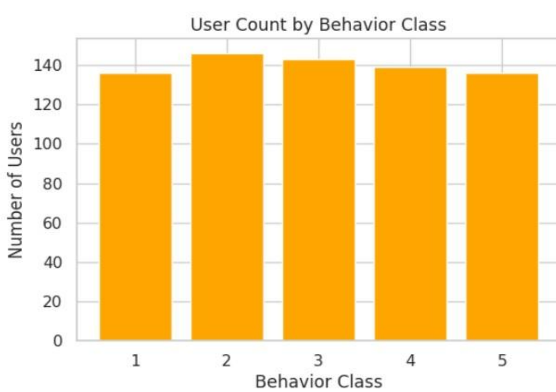


Fig 1.9 Number of Users in Each Class

Ethar Mostafa 22-101121 Noureen Ibrahim 22-101056 Kareem ElKenany 22-101031
Arwa Mostafa 22-101120 Rana Wael 22-101219 Abdelrahman Mahdi 22-101173

2. Mobile Addiction Dataset

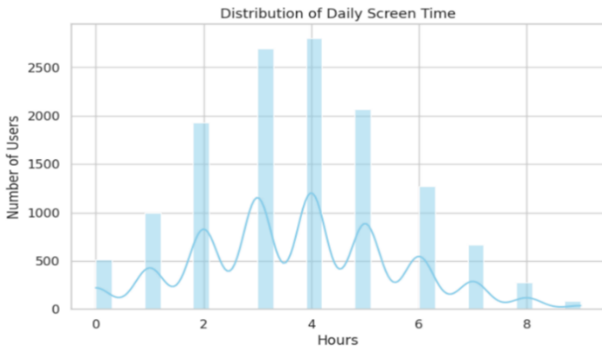


Fig 1.10 Histogram of Daily Screen Time

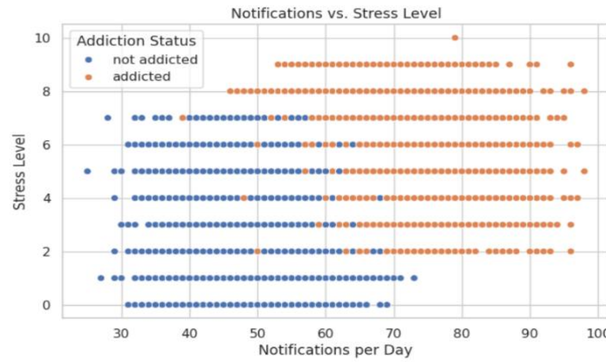


Fig 1.11 Notifications vs Stress Level

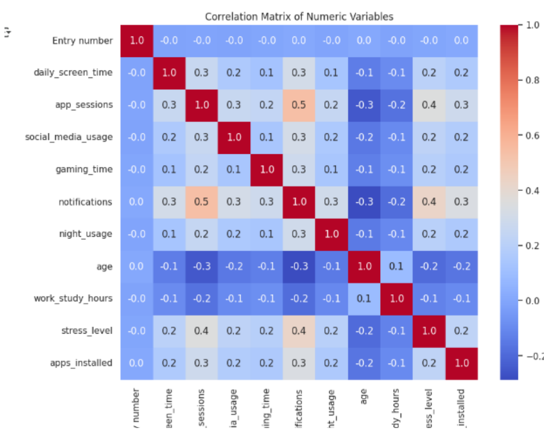


Fig 1.12 Correlation Matrix

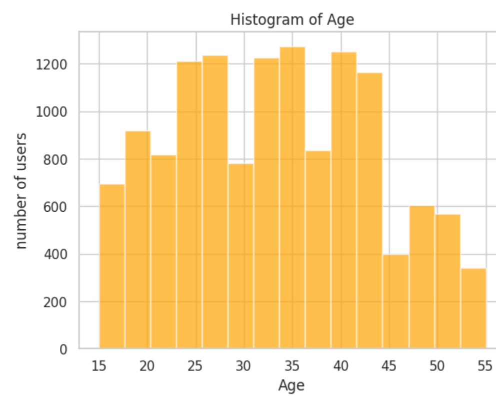


Fig 1.13 Distribution of Age

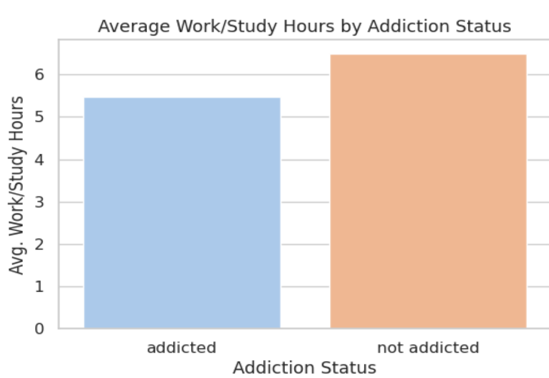


Fig 1.14 Average Work/Study Hours Between Addicted and Not Addicted Users

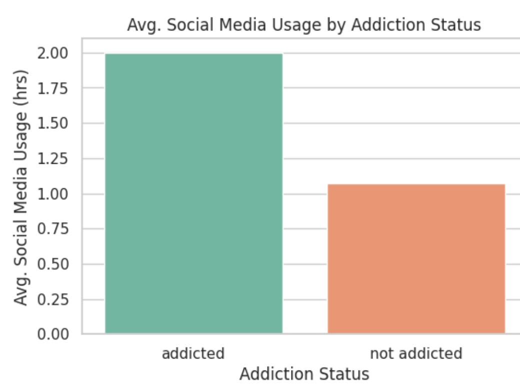


Fig 1.15 Average Social Media Usage Between Addicted and Not Addicted Users

Ethar Mostafa 22-101121 Noureen Ibrahim 22-101056 Kareem ElKenany 22-101031
Arwa Mostafa 22-101120 Rana Wael 22-101219 Abdelrahman Mahdi 22-101173

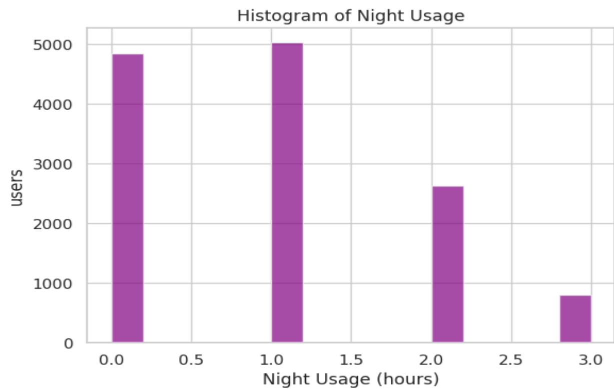


Fig 1.16 Histogram of Night Usage

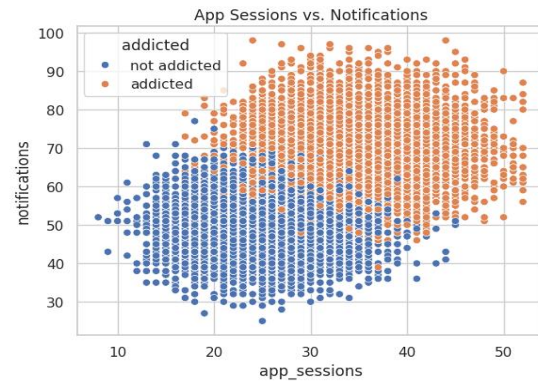


Fig 1.17 App Sessions vs Notifications

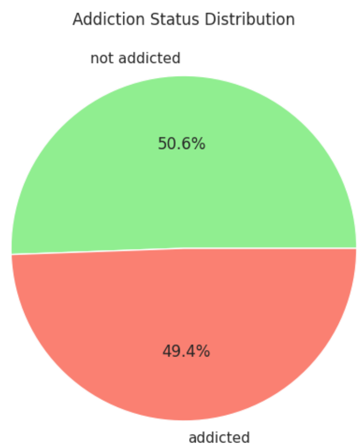


Fig 1.18 Addiction Class Distribution

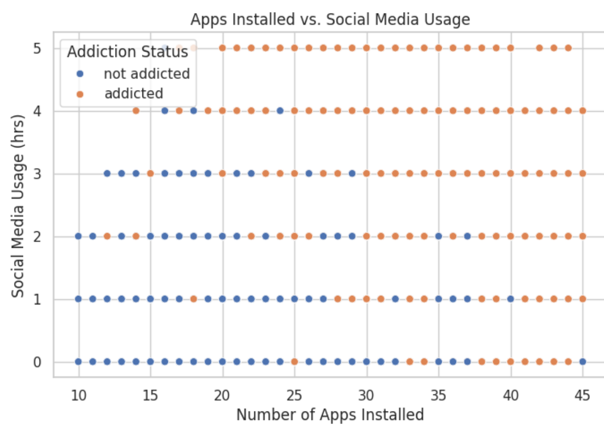


Fig 1.19 Apps Installed vs Social Media Usage

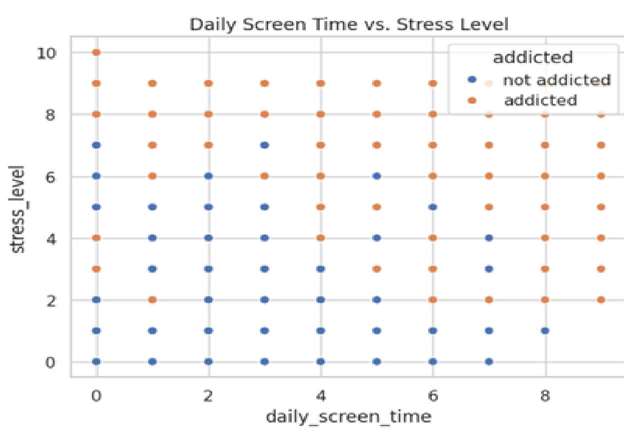


Fig 1.20 Daily Screen Time vs Stress Level

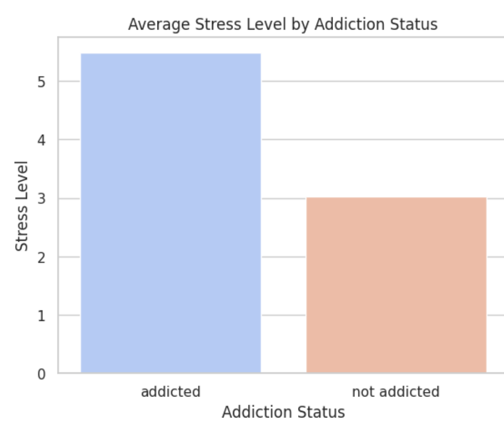


Fig 1.21 Average Stress Level by Addiction Status

Ethar Mostafa 22-101121 Noureen Ibrahim 22-101056 Kareem ElKenany 22-101031
Arwa Mostafa 22-101120 Rana Wael 22-101219 Abdelrahman Mahdi 22-101173

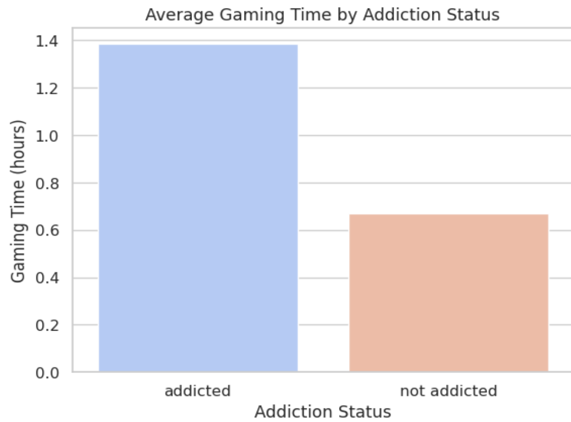


Fig 1.22 Average Gaming Time by Addiction Status

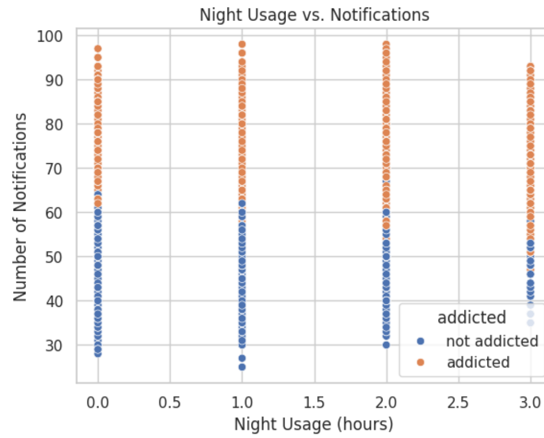


Fig 1.23 Night Usage vs Notifications

- **Regression**

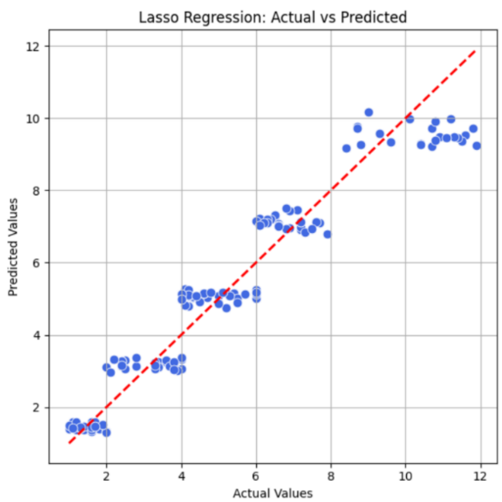


Fig 2.1 Shows how well the Lasso Regression model's predictions match the actual values

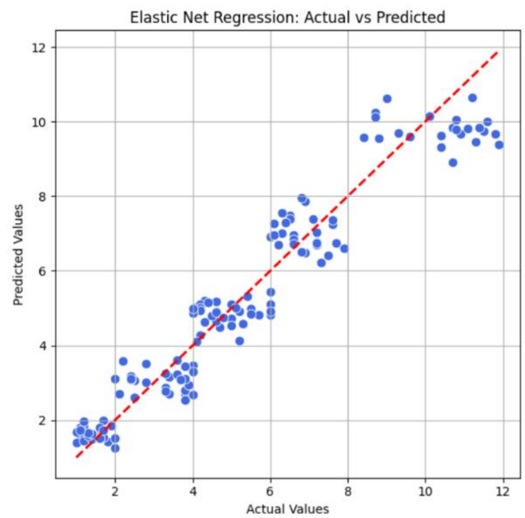


Fig 2.2 Shows how well the Elastic Net Regression model's predictions match the actual values

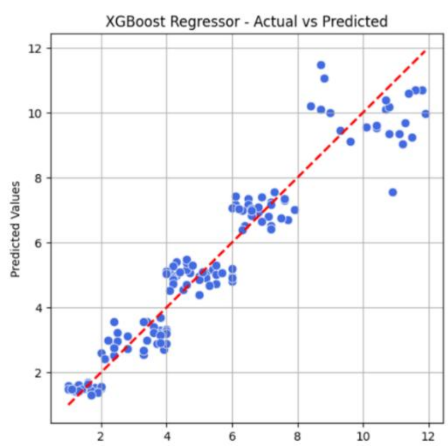


Fig 2.3 Shows how well the XGBoost Regression model's predictions match the actual values

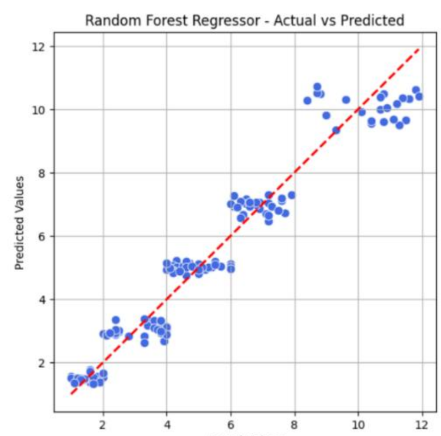


Fig 2.4 Shows how well the Random Forest Regression model's predictions match the actual

Ethar Mostafa 22-101121 Noureen Ibrahim 22-101056 Kareem ElKenany 22-101031
Arwa Mostafa 22-101120 Rana Wael 22-101219 Abdelrahman Mahdi 22-101173

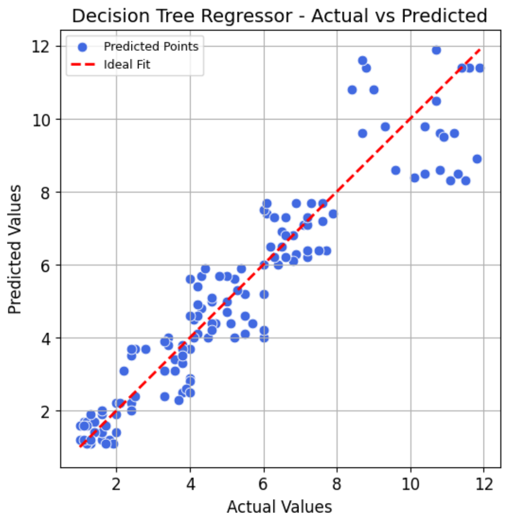


Fig 2.5 Shows how well the Decision Tree Regression model's predictions match the actual values

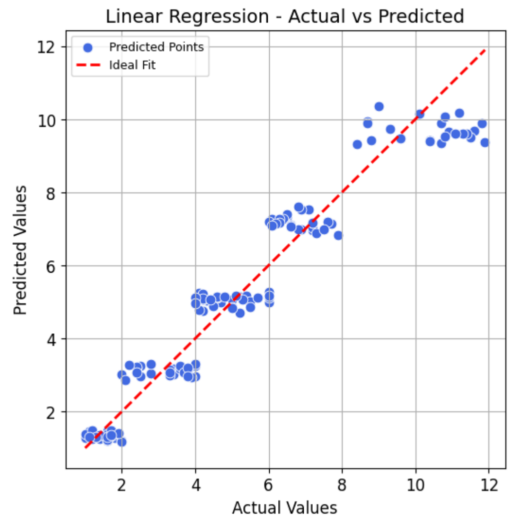


Fig 2.6 Shows how well the Linear Regression model's predictions match the actual values

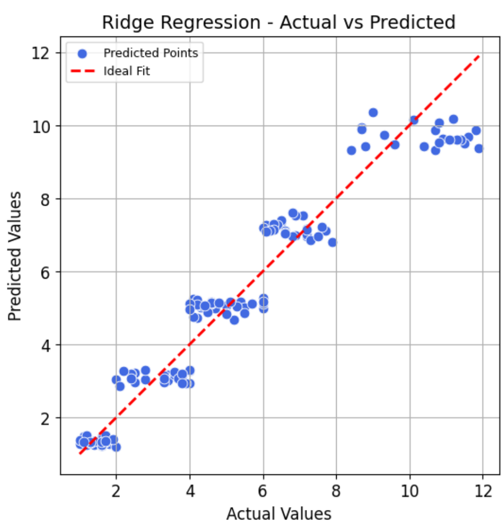


Fig 2.7 Shows how well the Ridge Regression model's predictions match the actual values

- **Classification**

1. Naïve Bays:

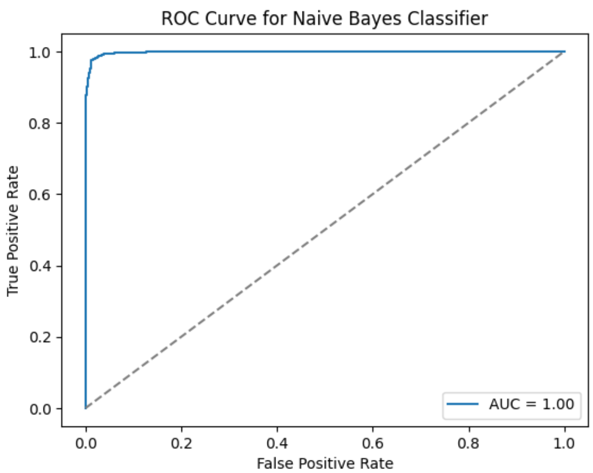


Fig 3.1 ROC Curve of the Naïve Bays Classifier

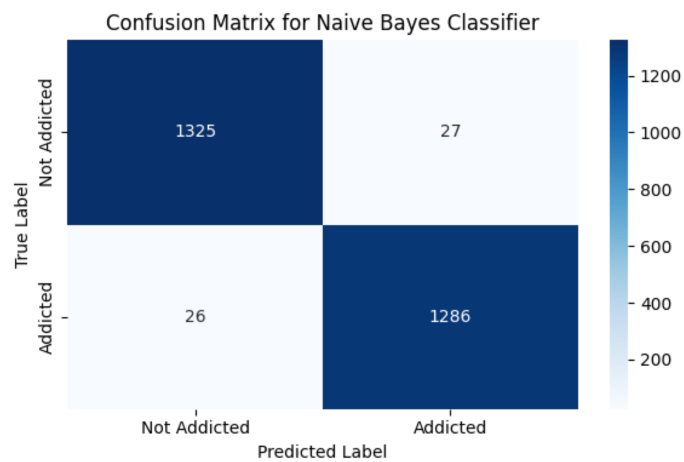


Fig 3.2 Confusion Matrix of the Naïve Bays Classifier

2. K-Nearest Neighbors (KNN):

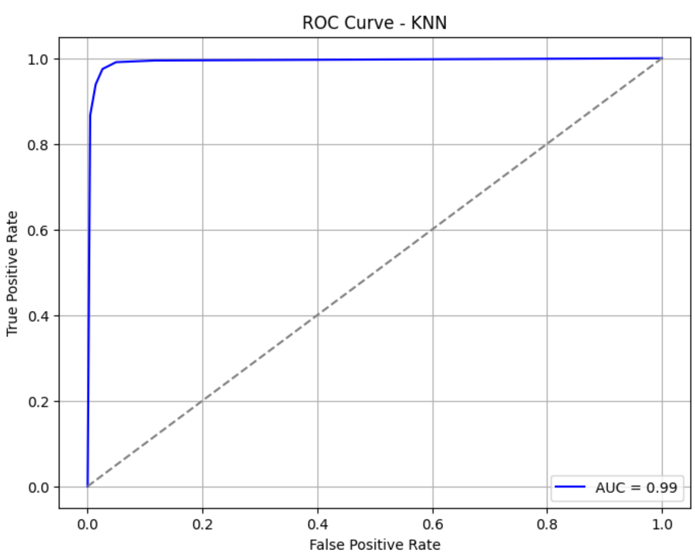


Fig 3.3 ROC Curve of the KNN Classifier

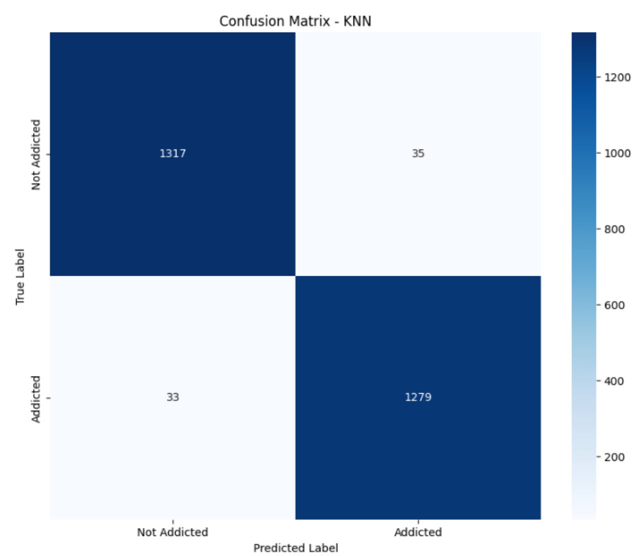


Fig 3.4 Confusion Matrix of the KNN Classifier

Ethar Mostafa	22-101121	Noureen Ibrahim	22-101056	Kareem ElKenany	22-101031
Arwa Mostafa	22-101120	Rana Wael	22-101219	Abdelrahman Mahdi	22-101173

3. Decision Tree:

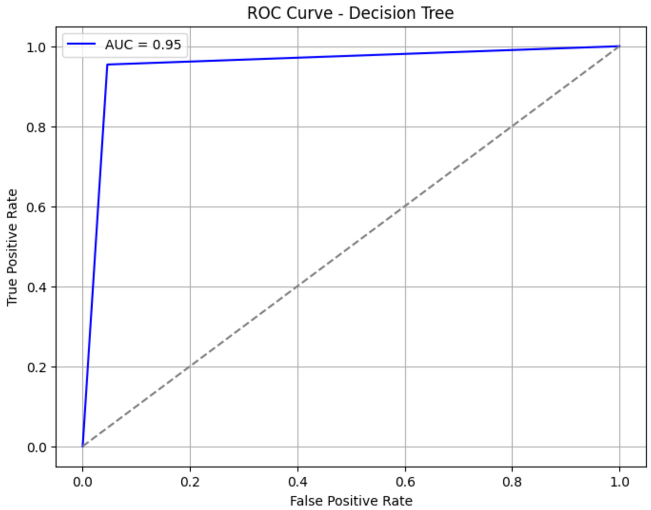


Fig 3.5 ROC Curve of the Decision Tree Classifier

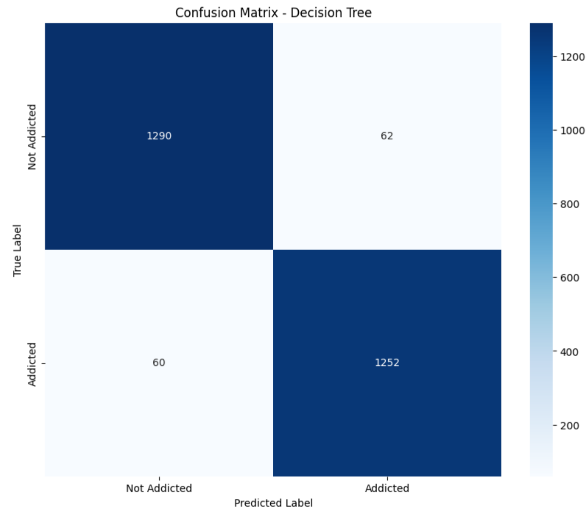


Fig 3.6 Confusion Matrix of the Decision Tree Classifier

4. Random Forest:

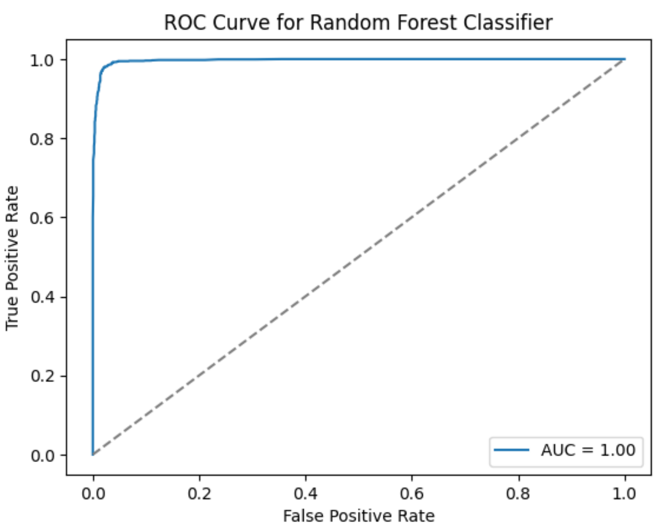


Fig 3.7 ROC Curve of the Random Forest Classifier

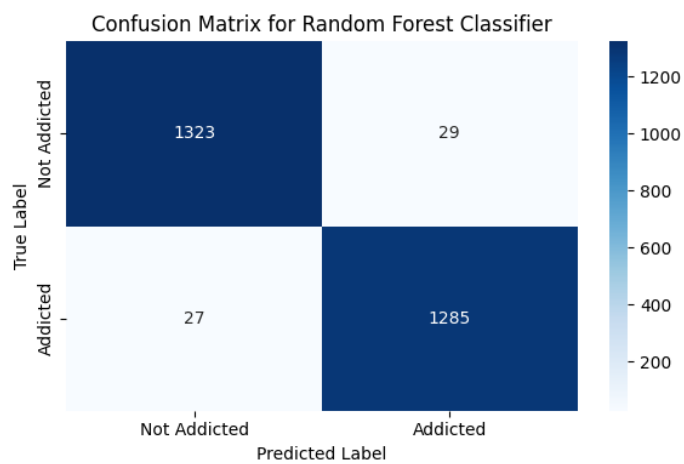


Fig 3.8 Confusion Matrix of the Random Forest Classifier

- **Model Performance**
  - **Regression**

Model	R^2 Score	MSE
Linear Regression	93.2%	0.61
Ridge Regression	93.2%	0.61
Lasso Regression	92.9%	0.64
Elastic Net Regression	92.5%	0.67
Random Forest Regression	94.4%	0.51
XGBoost Regression	93.9%	0.55
Decision Tree Regression	88.3%	1.04

Ensemble models, particularly Random Forest and XGBoost, outperformed the others, achieving the highest R^2 scores and the lowest mean squared errors. This is expected as these models combine the predictions of many decision trees, allowing them to better capture non-linear relationships and interactions among features. In contrast, linear models such as Linear Regression, Ridge, Lasso, and Elastic Net performed slightly worse but still demonstrated strong predictive capabilities, indicating that the relationship between features and the target is largely linear with minor complexities. The Decision Tree Regressor, being a single tree, underperformed significantly due to its tendency to overfit, especially in comparison to ensemble methods that average across many trees to reduce variance.

  - **Classification**

Model	Accuracy	F1 (Addicted)	F1 (Not Addicted)	AUC
KNN	97.52%	0.98	0.97	0.99
Decision Tree	95.65%	0.96	0.96	0.96
Random Forest	97.7%	0.98	0.98	1.00
Naive Bayes	98.0%	0.98	0.98	1.00

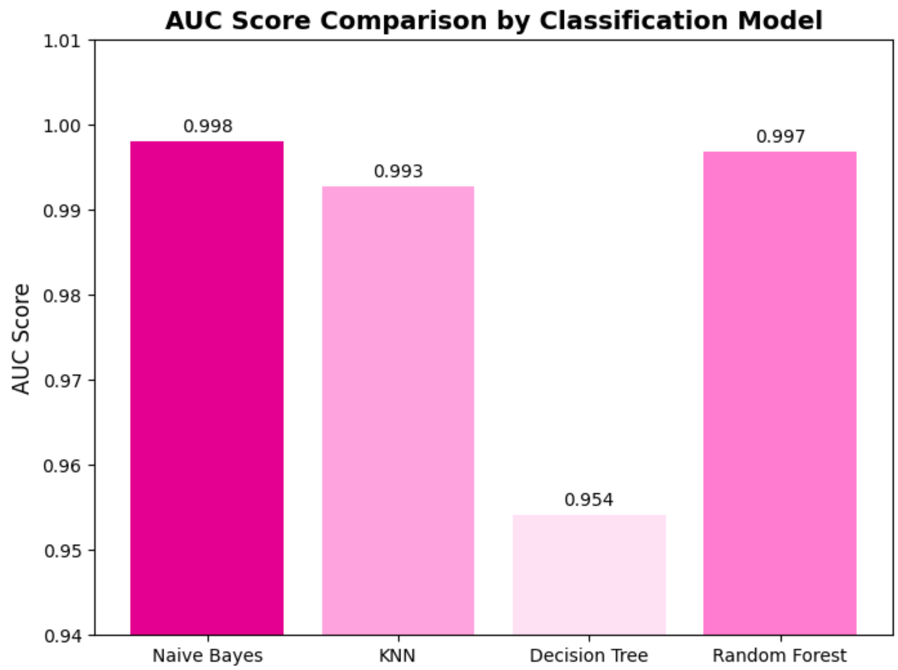


Fig 3.9 Comparison of AUC scores of the classification models

four models performed extremely well, but Random Forest and Naïve Bayes slightly outperformed others in terms of AUC and F1 balance. Random Forest excelled due to its flexibility with non-linear data and ensemble robustness, while Naive Bayes performed well because of the clean data, near-normal numeric features, and moderate feature independence. Our dataset was well-suited to both, which explains their top-tier performance compared to KNN or Decision Tree alone. These results suggest that mobile phone addiction can be effectively predicted using behavioral and psychological variables such as screen time, app sessions, notifications, and stress level.

Conclusions

- **Summary of Findings**
  Our analysis found a clear link between stress and mobile phone addiction. Students with higher stress levels tended to spend more time on their phones, especially at night, and received more notifications. Addicted users also spent less time on work or study-related activities. Classification models like Random Forest and Naïve Bayes were very effective in identifying addicted users, while Random Forest Regression gave the most accurate screen time predictions.

- **Recommendations**
  It's worth encouraging students to track their phone usage, especially screen time and notifications, and to try managing stress in healthier ways. Schools and universities could also provide support or workshops on digital wellbeing and stress reduction.
- **Limitations**
  The data we used was self-reported, so there's a chance it's not 100% accurate. Also, since it's cross-sectional, we can't say for sure if stress causes phone addiction or the other way around. Also, the data mostly came from one group of users, which might not represent everyone.
- **Future Work**
  For future research, using real-time data (like from phone apps) and including more psychological or lifestyle factors could give a clearer picture. A long-term study could also help show how these behaviors change over time.

Acknowledgements

- **Team Contributions:**
  1. **Data Wrangling:** Abdelrahman Mahdi
  2. **Data Visualization:** Noureen Ibrahim
  3. **Regression Models:** Rana Wael & Kareem El Kenany
  4. **Classification Models:** Ethar Mostafa & Arwa Mostafa
- **External Support:**
  1. **Libraries:** numpy, seaborn, matplotlb, plotly, sklearn
  2. **Datasets:** mobile_addiction and user_behavior_dataset from Kaggle
  3. **Mentors:** Dr Fatma Al Sahabi and TA Nervana Abdullah

Ethar Mostafa 22-101121 Noureen Ibrahim 22-101056 Kareem ElKenany 22-101031
Arwa Mostafa 22-101120 Rana Wael 22-101219 Abdelrahman Mahdi 22-101173

References

- Zhang, A., Xiong, S., Peng, Y., Zeng, Y., Zeng, C., Yang, Y., & Zhang, B. (2022). Perceived stress and mobile phone addiction among college students: The roles of self-control and security. *Frontiers in Psychiatry*, *13*. https://doi.org/10.3389/fpsyt.2022.1005062
- Zhang, X., Gao, F., Kang, Z., Zhou, H., Zhang, J., Li, J., Yan, J., Wang, J., Liu, H., Wu, Q., & Liu, B. (2022). Perceived academic stress and Depression: The mediation role of mobile phone addiction and sleep quality. *Frontiers in Public Health*, *10*. https://doi.org/10.3389/fpubh.2022.760387
- *Mobile_User_Behaviour_Dataset*. (2024, November 6). Kaggle. https://www.kaggle.com/datasets/haneeshathasnin/mobile-user-behaviour-dataset
- *mobile addiction data*. (2025, April 17). Kaggle. https://www.kaggle.com/datasets/cloudymts/mobile-addiction-data
- Normalized Nerd. (2021, January 13). *Decision tree classification clearly explained!* [Video]. YouTube. https://www.youtube.com/watch?v=ZVR2Way4nwQ
- Normalized Nerd. (2021b, April 21). *Random Forest algorithm clearly explained!* [Video]. YouTube. https://www.youtube.com/watch?v=v6VJ2RO66Ag
- Rithp. (2023, January 16). Optimizing XGBOOSt: A Guide to Hyperparameter tuning. *Medium*. https://medium.com/@rithpansanga/optimizing-xgboost-a-guide-to-hyperparameter-tuning-77b6e48e289d
- GeeksforGeeks. (2025, April 7). *Random forest regression in Python*. GeeksforGeeks. https://www.geeksforgeeks.org/random-forest-regression-in-python/
- SkillCurb. (2020, September 25). *What is Data Wrangling and Data Cleaning for beginners* [Video]. YouTube. https://www.youtube.com/watch?v=v7RvZiMUtt0
- GeeksforGeeks. (2024, July 4). *Introduction to Seaborn Python*. GeeksforGeeks. https://www.geeksforgeeks.org/introduction-to-seaborn-python/