# Machine Learning Project Report

Student Social Media Addiction

**Prepared by**
Abhijith P

**Project Type:** Data Science / Machine Learning
**Tools Used:** Python, Jupyter Notebook

January 18, 2026

# Contents

# Chapter 1

# Machine Learning Project

## 1.1 Project Overview

Social media usage among students has grown rapidly in recent years, affecting daily routines, academic performance, mental health, and sleep patterns.

This project applies machine learning techniques to analyze student social media usage and predict addiction levels using behavioral and lifestyle data.

## 1.2 Project Objectives

- Perform Exploratory Data Analysis (EDA)
- Preprocess data using encoding and scaling
- Build Linear Regression and KNN models
- Evaluate models using $R^2$ and MAE

# Chapter 2

# Problem Statement

Raw social media usage data is difficult to interpret without structured analysis. The dataset sourced from Kaggle (2025 timeline) requires preprocessing and machine learning to extract meaningful insights.

- Raw data lacks clear patterns
- Relationships are not directly visible
- Machine learning is required for prediction

# Chapter 3

# Dataset Description

## 3.1 Data Source

The dataset is sourced from Kaggle and contains information related to students' social media usage behavior, lifestyle attributes, and social media addiction scores.

## 3.2 Key Features

- Student ID
- Age
- Gender
- Daily Screen Time
- Platform Usage
- Sleep Duration
- Academic Impact
- Addiction Score (Target Variable)

## 3.3 Dataset Snapshot

| Student ID | Age | Gender | Academic Level | Country | Avg Daily Usage (hrs) | Most Used Platform | Affects Academic Performance | Sleep Hours Per Night | Mental Health Score | Relationship Status | Conflicts Over Social Media | Addicted Score |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 19 | Female | Undergraduate | Bangladesh | 5.2 | Instagram | Yes | 6.5 | 6 | In Relationship | 3 | 8 |
| 2 | 22 | Male | Graduate | India | 2.1 | Twitter | No | 7.5 | 8 | Single | 0 | 3 |
| 3 | 20 | Female | Undergraduate | USA | 6.0 | TikTok | Yes | 5.0 | 5 | Complicated | 4 | 9 |
| 4 | 18 | Male | High School | UK | 3.0 | YouTube | No | 7.0 | 7 | Single | 1 | 4 |
| 5 | 21 | Male | Graduate | Canada | 4.5 | Facebook | Yes | 6.0 | 6 | In Relationship | 2 | 7 |

Figure 3.1: Sample View of the Student Social Media Addiction Dataset

# Chapter 4

# Exploratory Data Analysis

## Contents

## 4.1 Overview

Exploratory Data Analysis (EDA) was conducted to understand the dataset structure, assess data quality, and identify inconsistencies before preprocessing and model development. This step focuses on examining feature distributions, detecting anomalies, and validating relationships related to students' social media usage behavior.

The following Pandas functions were used for initial data exploration:

- `df.shape()` – dataset dimensions
- `df.columns` – feature names
- `df.dtypes()` – data types
- `df.info()` – missing values
- `df.describe()` – statistical summary
- `df.head()` / `df.tail()` – sample records

### Dropping Unwanted Columns

Certain features were removed to reduce noise and improve model performance:

- **Student_ID**: Unique identifier with no predictive value.
- **Conflicts_Over_Social_Media**: Subjective and noisy.
- **Country**: Highly categorical and not behavior-specific.

```
[8]: df.drop(columns=['Student_ID','Conflicts_Over_Social_Media',"Country"],inplace=True)

[9]: df.columns

[9]: Index(['Age', 'Gender', 'Academic_Level', 'Avg_Daily_Usage_Hours',
       'Most_Used_Platform', 'Affects_Academic_Performance',
       'Sleep_Hours_Per_Night', 'Mental_Health_Score', 'Relationship_Status',
       'Addicted_Score'],
      dtype='object')
```

Figure 4.1: Removal of Unnecessary Columns and Updated Feature Set

## 4.2 Null Value Detection

Identifying null values is a critical step in data preprocessing, as missing data can negatively impact exploratory analysis and machine learning model performance. Detecting null values early helps determine whether imputation, removal, or other handling strategies are required.

To examine the presence of missing values in the dataset, built-in Pandas functions were used to compute the count of null values across all features.

The analysis confirmed that the dataset is complete and does not contain any missing or null values in any column. As a result, no additional null value handling or imputation techniques were required before proceeding to further preprocessing steps.

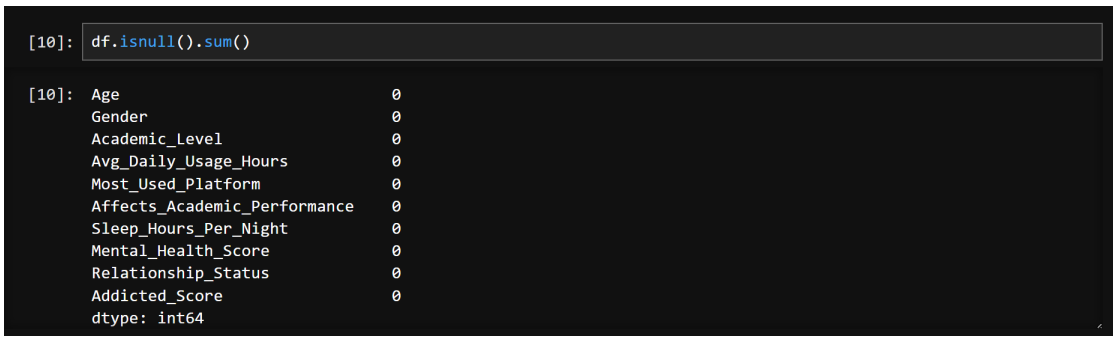



Figure 4.2: Null Value Detection Using Pandas

## 4.3 Outlier Detection

Handling outliers is an important step in data preprocessing, as extreme values can influence statistical analysis and machine learning model performance. Identifying and evaluating outliers helps determine whether they should be treated or retained based on their relevance and validity.

To visually inspect the presence of outliers, boxplots were generated for selected numerical features related to usage behavior and mental health.

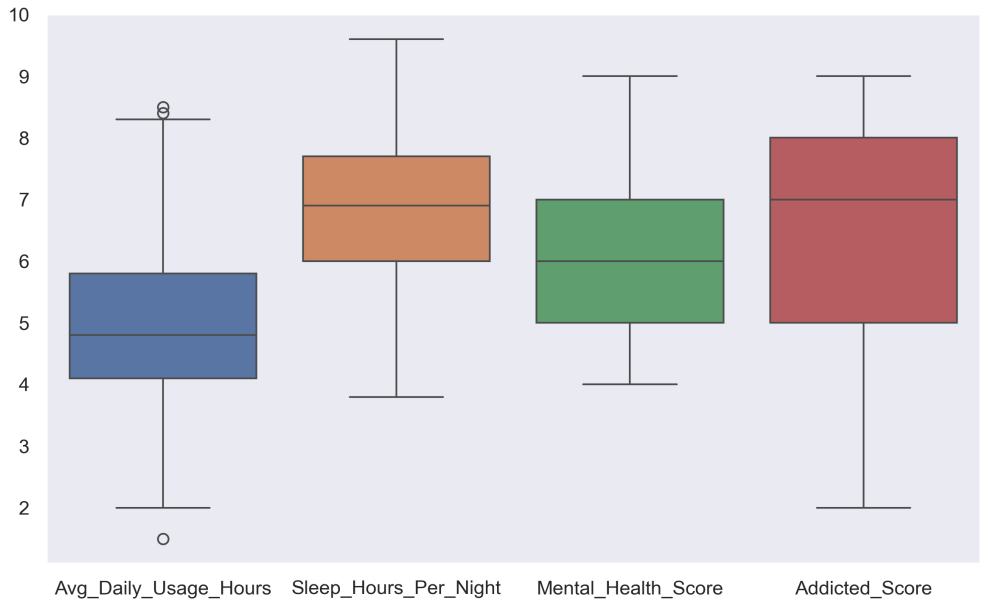



Figure 4.3: Outlier Detection Using Boxplot Visualization

**Boxplot Analysis:** From the boxplot visualization, it was observed that the feature `Avg_Daily_Usage_Hours` contains only a small number of outliers, while most observations fall within a reasonable and expected range.

To further identify these outliers numerically, the Interquartile Range (IQR) method was applied. Quartiles (Q1 and Q3) were computed, and upper and lower bounds were defined using the standard $1.5 \times IQR$ rule. Values outside this range were flagged as potential outliers.

The identified outliers were retained in the dataset, as they represent realistic variations in students' daily social media usage rather than data entry errors or anomalies. Removing these values could lead to loss of important behavioral information.

## 4.4 Statistical Summary

Statistical summary analysis was performed to understand the central tendency, spread, and distribution of numerical features in the dataset. This step provides insights into the range, mean, and variability of key variables related to students' social media usage behavior.

Descriptive statistics such as mean, median, standard deviation, minimum, and maximum values were computed using Pandas to evaluate the overall data distribution and detect potential anomalies.

The statistical summary highlights variations in daily screen time, sleep duration, mental health scores, and social media addiction scores, which are critical for understanding behavioral patterns prior to preprocessing and model training.

```
[6]: df.describe()
```

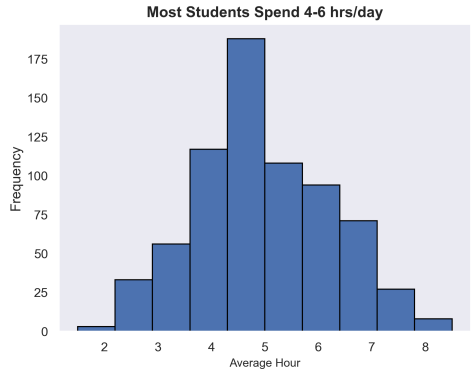
[6]:

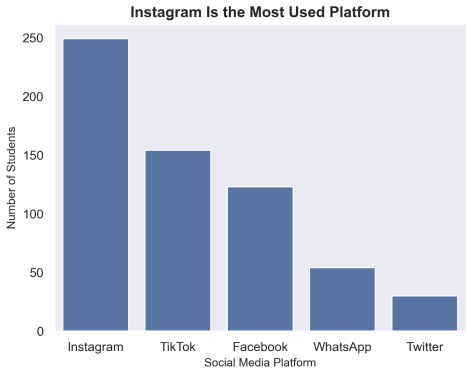
| | Student_ID | Age | Avg_Daily_Usage_Hours | Sleep_Hours_Per_Night | Mental_Health_Score | Conflicts_Over_Social_Media | Addicted_Score |
|---|---|---|---|---|---|---|---|
| count | 705.000000 | 705.000000 | 705.000000 | 705.000000 | 705.000000 | 705.000000 | 705.000000 |
| mean | 353.000000 | 20.659574 | 4.918723 | 6.868936 | 6.226950 | 2.849645 | 6.436879 |
| std | 203.660256 | 1.399217 | 1.257395 | 1.126848 | 1.105055 | 0.957968 | 1.587165 |
| min | 1.000000 | 18.000000 | 1.500000 | 3.800000 | 4.000000 | 0.000000 | 2.000000 |
| 25% | 177.000000 | 19.000000 | 4.100000 | 6.000000 | 5.000000 | 2.000000 | 5.000000 |
| 50% | 353.000000 | 21.000000 | 4.800000 | 6.900000 | 6.000000 | 3.000000 | 7.000000 |
| 75% | 529.000000 | 22.000000 | 5.800000 | 7.700000 | 7.000000 | 4.000000 | 8.000000 |
| max | 705.000000 | 24.000000 | 8.500000 | 9.600000 | 9.000000 | 5.000000 | 9.000000 |

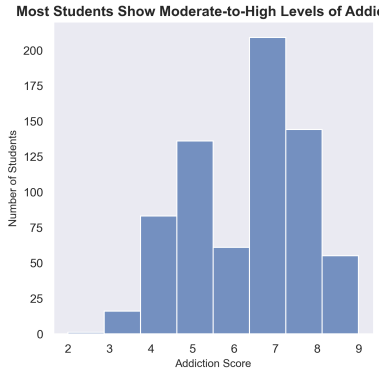
Figure 4.4: Statistical Summary of Numerical Features

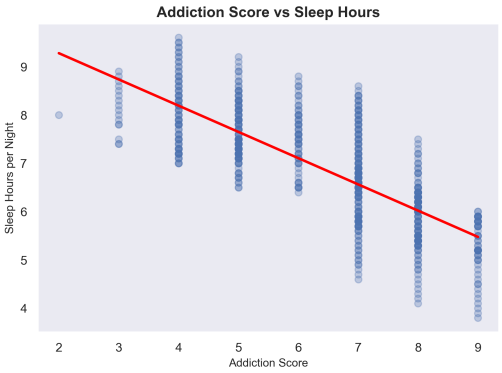
## 4.5 Data Visualization

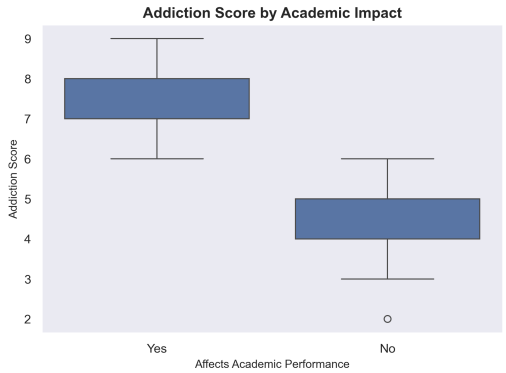
The following visualizations illustrate key patterns and relationships in student social media usage behavior and addiction levels.

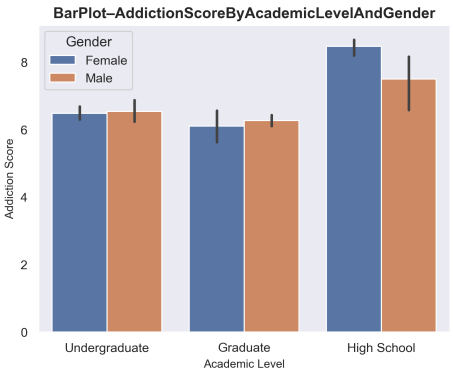


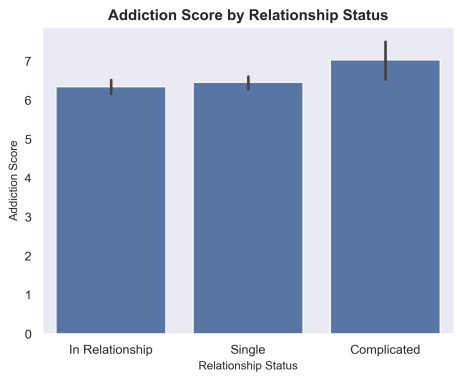



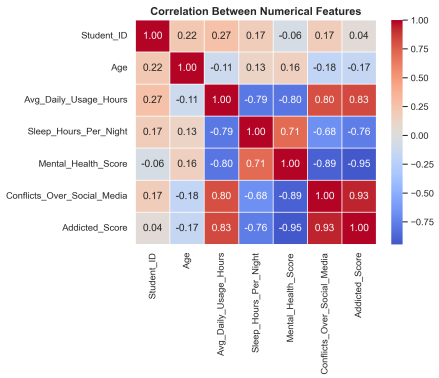



## 4.6 Visualization Insights

The visualizations provide meaningful insights into students' social media usage behavior and its impact on lifestyle, mental health, and academic performance.

- Most students spend approximately 4–6 hours per day on social media, indicating high daily engagement.
- Moderate to high addiction levels are common, with many students scoring around 7.
- Instagram is the most widely used social media platform among students.
- Higher addiction scores are associated with reduced sleep duration and poorer mental health.
- Students whose academic performance is affected tend to show higher addiction levels.
- High school students exhibit higher addiction scores compared to undergraduate and graduate students.
- Students with complicated relationship status show the highest average addiction scores.

# Chapter 5

# Data Preprocessing

## Contents

## 5.1 Train–Test Split

Before building machine learning models, the dataset is divided into training and testing sets. This ensures that the model learns patterns from one portion of the data and is evaluated on unseen data, allowing an unbiased assessment of real-world performance.

### Importance of Train–Test Split

Train–test splitting is essential to:

- Prevent overfitting by evaluating the model on unseen data.
- Ensure fair and reliable model validation.

### Why Train–Test Split Is Performed Before Encoding

The dataset is split before applying encoding and scaling techniques to avoid **data leakage**. Applying preprocessing before splitting may allow information from the test set to influence the training process, leading to misleading performance results.

### Feature and Target Separation

The dataset is first divided into independent (feature) variables and the dependent (target) variable.

### Implementation

The `train_test_split` function from Scikit-learn is used to split the dataset into training and testing sets. A test size of 25% is selected with a fixed random state for reproducibility.

```
# Separating features and target
X = df.drop(columns=['Addicted_Score'])
y = df['Addicted_Score']

# Splitting the dataset
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.25, random_state=42)
```

Listing 5.1: Train–Test Split Implementation

## 5.2 Encoding

Encoding is the process of converting categorical variables into numerical representations so that they can be interpreted by machine learning algorithms. Since most models operate only on numerical data, encoding is a crucial preprocessing step.

In this project, both **Label Encoding** and **One-Hot Encoding** are used based on the nature of the categorical features.

**One-Hot Encoding** is applied to categorical variables with multiple categories:

Encoding is performed separately on training and testing data to prevent data leakage.

```
# Identifying categorical columns
df.select_dtypes('object').columns

# Label Encoding
from sklearn.preprocessing import LabelEncoder
le_gender = LabelEncoder()
le_academic = LabelEncoder()
x_train["Gender"]=le_gender.fit_transform(x_train["Gender"])
x_test["Gender"]=le_gender.transform(x_test["Gender"])

x_train["Affects_Academic_Performance"] = le_academic.
   fit_transform(
    x_train["Affects_Academic_Performance"])
x_test["Affects_Academic_Performance"] = le_academic.
   transform(
    x_test["Affects_Academic_Performance"])

# One-Hot Encoding
cols = ["Academic_Level", "Most_Used_Platform", "
   Relationship_Status"]
x_train = pd.get_dummies(x_train, columns=cols, drop_first=
   True)
x_test = pd.get_dummies(x_test, columns=cols, drop_first=
   True)

x_train = x_train.astype(int)
x_test = x_test.astype(int)
```

Listing 5.2: Encoding Categorical Variables

## 5.3 Feature Scaling

Feature scaling is the process of transforming numerical features to a common scale so that no single feature dominates the model due to its magnitude. Scaling helps machine learning algorithms perform more efficiently and improves convergence during training.

In this project, **Min–Max Scaling** is applied to normalize numerical features within a fixed range.

**Why Min–Max Scaling?**

- Scales values to a fixed range between 0 and 1.
- Preserves the original distribution and relative relationships between values.
- Suitable when the dataset does not contain extreme outliers.

Only numerical features relevant to usage behavior and lifestyle factors are selected for scaling.

```
# Identifying numerical columns
df.select_dtypes('number').columns

# Importing Min-Max Scaler
from sklearn.preprocessing import MinMaxScaler
minmax = MinMaxScaler()

# Columns selected for scaling
cols = [
    'Age',
    'Avg_Daily_Usage_Hours',
    'Sleep_Hours_Per_Night',
    'Mental_Health_Score'
]

# Applying scaling
x_train[cols] = minmax.fit_transform(x_train[cols])
x_test[cols] = minmax.transform(x_test[cols])
```

Listing 5.3: Min–Max Feature Scaling

# Chapter 6

# Machine Learning Models

## Contents



## 6.1 Overview

Machine learning is used to train models that learn patterns from data and make predictions on unseen observations.

### Purpose of Machine Learning in This Project

- Predict the continuous target variable **Addicted_Score**.
- Learn relationships between social media usage, lifestyle, and academic factors.

### Models Used

- **Linear Regression**: Used as the primary baseline model to predict the continuous target variable and analyze the influence of input features on social media addiction scores.
- **K-Nearest Neighbors (KNN) Regression**: Used to validate model performance and capture non-linear relationships by comparing predictions based on nearest data points.

## 6.2 Linear Regression

Linear Regression is used to predict students' social media addiction scores, as the target variable (`Addicted_Score`) is continuous. This model serves as a baseline regression approach to understand the relationship between input features and the predicted output.

### Model Training and Prediction

The Linear Regression model is trained using the training dataset and then used to predict addiction scores on the test dataset.

```
# Importing Linear Regression
from sklearn.linear_model import LinearRegression
lr = LinearRegression()

# Training the model
model = lr.fit(x_train, y_train)

# Making predictions
y_pred = model.predict(x_test)
```

Listing 6.1: Linear Regression Model Implementation

### Model Evaluation

The model performance is evaluated using R-squared ($R^2$) Score and Mean Absolute Error (MAE).

```
from sklearn.metrics import r2_score, mean_absolute_error

r2 = r2_score(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)

print("R2␣Score:", r2)
print("Mean␣Absolute␣Error:", mae)
print("Intercept:", lr.intercept_)
```

Listing 6.2: Linear Regression Evaluation Metrics

## Results and Interpretation

- The $R^2$ score indicates that the model explains approximately 95% of the variance in the target variable.
- The low Mean Absolute Error (MAE) shows that predicted addiction scores are close to the actual values.
- The model demonstrates strong predictive performance on unseen test data.
- The intercept represents the baseline addiction score when all input features are zero.

## 6.3 K-Nearest Neighbors (KNN) Regression

In addition to Linear Regression, KNN Regression is applied to validate and cross-check model predictions. KNN helps capture potential non-linear relationships that linear models may not fully represent, improving confidence in the robustness of the results.

KNN Regression predicts the target value based on the average outcome of the nearest neighboring data points in the feature space.

**Model Selection and Evaluation**

Multiple values of $K$ (number of neighbors) are tested to identify the optimal model configuration. The R-squared ($R^2$) score is used to evaluate performance for each value of $K$.

```
from sklearn.neighbors import KNeighborsRegressor
from sklearn.metrics import r2_score
import numpy as np

metric_k = []
neighbors = np.arange(1, 15)

for k in neighbors:
    knn = KNeighborsRegressor(n_neighbors=k)
    model = knn.fit(x_train, y_train)
    ypred_knn = model.predict(x_test)
    metric_k.append(r2_score(y_test, ypred_knn))
```

Listing 6.3: KNN Model Training for Different K Values

A line plot is used to visualize how the $R^2$ score varies with different values of $K$. Although the highest score occurs at lower values of $K$, $K = 3$ is selected as it provides a good balance between performance and stability.

```
from sklearn.metrics import mean_squared_error

knn = KNeighborsRegressor(n_neighbors=3)
model = knn.fit(x_train, y_train)
ypred_knn = model.predict(x_test)

print("R2␣Score:", r2_score(y_test, ypred_knn))
print("Mean␣Squared␣Error:", mean_squared_error(y_test,
   ypred_knn))
```

Listing 6.4: Final KNN Model Training and Evaluation

**Results and Interpretation**

- The $R^2$ score of approximately 0.94 indicates that the model explains about 93% of the variance in the target variable.
- The low Mean Squared Error suggests that predicted addiction scores are close to actual values.
- The model generalizes well to unseen data and effectively captures non-linear patterns.

## 6.4 Final Insights

- The student social media addiction dataset was thoroughly analyzed using Exploratory Data Analysis (EDA), including null value detection, outlier evaluation, and multiple visualizations to understand usage behavior and trends.
- Data preprocessing steps such as dropping irrelevant features, encoding categorical variables, and applying Min–Max scaling improved data consistency and model performance.
- Linear Regression achieved a high $R^2$ score of approximately **0.95**, indicating a strong linear relationship between input features and the addiction score.
- The low Mean Absolute Error (MAE) obtained from Linear Regression shows that the predicted addiction scores are close to the actual values.
- The intercept value from the Linear Regression model represents the baseline addiction score when all input features are zero, improving model interpretability.
- K-Nearest Neighbors (KNN) Regression achieved an $R^2$ score of approximately **0.94**, demonstrating its ability to capture local and non-linear patterns in the data.
- The similar performance of both Linear Regression and KNN Regression confirms strong generalization on unseen data.
- Overall, the project demonstrates that effective preprocessing, appropriate model selection, and evaluation techniques can reliably predict student social media addiction levels.