SMARTPHONE ADDICTION LEVEL AMONG TEENS : A PREDICTIVE STUDY

A Mini Project Report

submitted by

JYOTHISH MOHAN A (MES24MCA-2025)

to the APJ Abdul Kalam Technological University
in partial fulfilment of the requirements for the award of the Degree

of

Master of Computer Applications

Department of Computer Applications

MES College of Engineering

Kuttippuram, Malappuram – 679582

October, 2025

Declaration

I undersigned hereby declare that the project report **SMARTPHONE ADDICTION LEVEL AMONG TEENS : A PREDICTIVE STUDY** submitted for partial fulfilment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala, is a bonafide work done by me under supervision of **Ms**. **Priya J D,** Assistant Professor, Department of Computer Applications. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

.

Date :

Signature :

JYOTHISH MOHAN A

(MES24MCA-2025)

DEPARTMENT OF COMPUTER APPLICATIONS

MES COLLEGE OF ENGINEERING, KUTTIPPURAM

CERTIFICATE

This is to certify that the report entitled **SMARTPHONE ADDICTION LEVEL AMONG TEENS : A PREDICTIVE STUDY** is a bonafide record of the Mini Project work during the year 2025-26 carried out by JYOTHISH MOHAN (MES24MCA-2025) submitted to the APJ Abdul Kalam Technological University, in partial fulfilment of the requirements for the award of the Master of Computer Applications, under my guidance and supervision. This report in any form has not been submitted to any other University or Institution for any purpose.

Internal Supervisor Head of The Department

Acknowledgment

I express my sincere gratitude to the **Department of Computer Applications, MES College of Engineering, Kuttippuram**, for providing me with the opportunity and necessary facilities to carry out this mini project titled *"*SMARTPHONE ADDICTION LEVEL AMONG TEENS: A PREDICTIVE STUDY.*"*

I would like to extend my heartfelt thanks to my **Project Supervisor, Ms. Priya J D, Assistant Professor**, for her invaluable guidance, constant encouragement, and constructive feedback throughout the development of this project. Her continuous support and suggestions have been instrumental in the successful completion of this work.

I am also deeply thankful to our **Mr. Hyderali K, Head of the Department, Faculty Members, and Lab Staff** of the Department of Computer Applications for their help and cooperation during various stages of the project.

Finally, I wish to thank my **family, classmates, and friends** for their constant motivation, moral support, and assistance throughout this project work.

JYOTHISH MOHAN A (MES24MCA-2025)

Abstract

Smartphone usage has become an integral part of modern life, especially among teenagers, resulting in rising concerns about excessive dependency and addiction. Identifying the potential risk of smartphone addiction at an early stage is crucial for promoting healthier digital habits. This project, titled "**SMARTPHONE ADDICTION LEVEL AMONG TEENS : A PREDICTIVE STUDY**", focuses on analyzing the behavioral and usage patterns of students to estimate their risk percentage of smartphone addiction using a data-driven approach.

The system collects data related to daily smartphone usage, sleep duration, academic performance, exercise hours, and social media activity. The collected data is preprocessed and analyzed using Python and Pandas, followed by the development of a predictive model using **Support Vector Regression** (SVR) from Scikit-learn. The model predicts the risk percentage of smartphone addiction . Based on the predicted risk, the system provides suggestions to reduce over-dependency and encourage balanced smartphone usage.

A web-based interface is developed using Django, HTML, and CSS to ensure interactive and user-friendly access to the system. This application serves as a useful tool for students, parents, and teachers to understand behavioral patterns and take preventive steps before addiction becomes severe. The project aims to enhance awareness about digital wellness and supports a data-driven strategy to mitigate smartphone addiction among teenagers.

.

Table of Contents

Chapter 1. Introduction

In the modern digital era, smartphones have become an inseparable part of daily life, especially among teenagers and young adults. Their widespread availability and multifunctionality have made them an essential communication and entertainment tool. However, excessive smartphone usage has led to a significant rise in dependency and behavioral addiction, affecting mental health, academic performance, and social interactions. The increasing trend of smartphone overuse among teenagers highlights the need for effective assessment tools that can measure and predict the risk of addiction before it reaches harmful levels.

Traditional approaches to studying smartphone addiction rely mainly on **manual surveys and questionnaires**, which are subjective, time-consuming, and limited in scope. These self-reported measures often produce inconsistent results due to human interpretation bias. Hence, there arises a need for a **data-driven predictive system** that can automatically analyze a user's smartphone usage behavior and estimate the risk of addiction accurately and efficiently.

This project aims to bridge that gap by developing a **machine learning–based web application** that predicts the **risk percentage of smartphone addiction among teenagers**. The system utilizes behavioral data such as daily phone usage hours, sleep duration, exercise patterns, academic performance, and screen time before sleep. It applies **Support Vector Regression (SVR)** to model and predict the addiction risk percentage. The system also provides **suggestions** based on the predicted risk value, helping users to adopt healthier digital habits.

This intelligent system is designed to assist **students, parents, and teachers** by providing early insights into digital behavior patterns. By quantifying addiction risk, it allows preventive action to be taken before addiction becomes severe, thus promoting a balanced and mindful approach toward smartphone usage.

1.1 Motivation

The increasing use of smartphones among teenagers has raised serious concerns regarding digital addiction and its negative impacts on learning ability, focus, and emotional well-being. Teenagers often spend excessive time on social media, gaming, and entertainment applications, leading to poor academic outcomes and disturbed sleep cycles. Manual assessment methods, though common, are inefficient and subjective.

The motivation behind this project stems from the need to **develop an automated, reliable, and data-driven system** that can assess the risk of smartphone addiction accurately. By leveraging **machine learning techniques**, this project not only helps individuals become aware of their digital habits but also provides valuable insights to parents and educators to guide students toward balanced smartphone use.

1.2 Objectives

The main objectives of this project are as follows:

- To collect and analyze smartphone usage patterns among teenagers.
- To preprocess and organize behavioral and academic data for effective model training.
- To implement a Machine Learning Regression model to predict the risk percentage of smartphone addiction**.**
- To develop a web-based interface using Django for user interaction and instant results.
- To provide suggestions to users based on their predicted addiction risk.
- To promote awareness and encourage healthier smartphone usage habits among students.

1.3 Report Organization

The project report is structured into five chapters for systematic presentation. **Chapter 1** introduces the project by outlining the background, motivation, objectives, and contributions of the study. **Chapter 2** presents the system study, which discusses the limitations of existing methods and explains the proposed system and its key functionalities. **Chapter 3** details the methodology adopted for the development of the project, including the software tools used, description of modules, and the machine learning model applied for prediction. **Chapter 4** focuses on the results and discussions, highlighting the output of the developed system through screenshots and their interpretations. Finally, **Chapter 5** concludes the report by summarizing the major findings, achievements, and limitations of the project, along with potential directions for future work.

Chapter 2. System Study

This chapter explains the system study of the project, which includes the existing system, its limitations, and the proposed system along with its major functionalities. The aim is to analyze how the new system improves smartphone addiction risk prediction using a machine learning approach.

2.1 Existing System

The existing systems for assessing smartphone addiction among teenagers primarily depend on **manual surveys** and **self-assessment questionnaires**. These include standardized scales such as the **Smartphone Addiction Scale (SAS)** and other psychological evaluation tools that measure addiction tendencies based on self-reported behavior. In these systems, users are asked to respond to a set of predefined questions regarding their smartphone habits, emotional attachment, and usage patterns. The data collected is then manually analyzed to categorize individuals into addiction levels such as low, moderate, or high.

Although these methods have been widely used in academic research, they have several **limitations**. Firstly, the results are **subjective** since they rely on individual perception and honesty while answering the questions. Secondly, the **data collection process is time-consuming**, making it unsuitable for large-scale studies involving hundreds of participants. Moreover, manual scoring and human interpretation introduce **inconsistencies and potential biases**, reducing the accuracy of the findings.

From a technical perspective, these systems lack automation and **real-time predictive capability**. They do not utilize actual behavioral or academic data, nor do they incorporate data analytics or machine learning for pattern recognition. As a result, such systems cannot provide instant results or personalized suggestions.

To overcome these limitations, the proposed project introduces a **machine learning–based predictive system** that automates the entire process—from data collection to prediction. This system uses **real-world behavioral parameters** instead of self-reported answers, ensuring a more **objective, accurate, and scalable** assessment of smartphone addiction risk.

2.2 Proposed System

The proposed system introduces a machine learning–based solution to accurately predict the **risk percentage of smartphone addiction** among teenagers. Unlike manual survey-based methods, this system automates the entire process of data collection, analysis, and prediction. It collects behavioral and lifestyle-related parameters such as daily smartphone usage hours, sleep duration, exercise time, academic performance, and social media activity.

The system is implemented as an interactive **web application** using **Python**, **Django**, **HTML**, and **CSS**. The web interface allows students to input their data easily and view their addiction risk percentage instantly. Based on the output, the system generates **suggestions** to encourage balanced smartphone usage.

By integrating machine learning with an accessible web interface, this proposed system aims to deliver **real-time, scalable, and objective** predictions. It also supports teachers and parents in identifying early signs of smartphone dependency, thereby promoting digital wellness and preventive awareness among teenagers.

2.3 Functionalities of Proposed System

The proposed system is designed to analyze behavioral and academic factors to predict the **risk percentage of smartphone addiction** among teenagers using machine learning. It integrates several core functionalities that collectively ensure accurate prediction, real-time feedback, and an interactive user experience. The major functionalities of the system are described below:

1. User Data Collection

   The system collects various input parameters from the user, such as **daily smartphone usage hours, sleep duration, academic performance, exercise time, social media usage,** and **screen time before bed**. These attributes form the basis for analyzing the user's smartphone behavior and overall lifestyle patterns. The data is entered through a **web-based interface** designed using HTML and Django templates, ensuring simplicity and ease of use for students and educators.

2. Data Preprocessing

   Before training or prediction, the collected data undergoes several preprocessing steps to ensure accuracy and consistency. This includes **data cleaning**, **label encoding** of categorical variables (e.g., gender), **normalization** of numerical features, and **removal of irrelevant fields**. The preprocessing is implemented using **Pandas** and **NumPy** libraries, which prepare the dataset for machine learning analysis.

3. Addiction Risk Prediction

   This is the core functionality of the system. The preprocessed data is fed into a **Support Vector Regression (SVR)** model implemented using **Scikit-learn**. The SVR model predicts a **continuous numeric output**, representing the user's **addiction risk percentage**. This approach provides a more granular and accurate assessment than classification-based models that divide users into discrete categories.

4. Result Display

   After prediction, the system immediately displays the **risk percentage** on the result page. This percentage indicates the likelihood of smartphone addiction for the given user. The result interface is designed to be clear and informative, helping users understand their digital behavior without requiring technical knowledge.

5. Suggestions

   Based on the predicted addiction risk percentage, the system provides **recommendations** to help users control smartphone usage.

6. Web Application Interface

   The entire system operates as an **interactive web application** developed using **Python, Django, HTML, and CSS**. It provides real-time access to prediction results and recommendations through any web browser. The system's backend is powered by Django, ensuring smooth communication between the data processing layer and the user interface.

Chapter 3. Methodology

This chapter details the systematic approach and the specific methodologies adopted for the development of the "Smartphone Addiction Level Among Teens: A Predictive Study" project. Building a reliable and accurate predictive system requires a structured methodology to guide the process from data handling and model selection to final web deployment. For this project, an Agile development methodology was employed, allowing for iterative progress, continuous testing, and the flexibility to incorporate changes, such as the pivotal shift from a classification model to a regression model for predicting addiction risk percentage.

3.1 Introduction

The Agile methodology is an iterative and incremental approach to software development that emphasizes flexibility, customer collaboration, and rapid delivery of functional software. Instead of delivering the entire project at once at the end of the cycle, the project is broken down into small, manageable units called "sprints." Each sprint typically lasts 1-4 weeks and results in a working product increment. This approach was chosen for this project as it allowed for continuous feedback, easy adaptation to changes (such as modifying the prediction output from a category to a percentage), and regular testing of individual components like the data preprocessing pipeline and the machine learning model.

3.2 Software Tools

The development of this project leveraged a combination of powerful and modern software tools and programming languages. Each tool was selected based on its strengths, community support, and suitability for the specific tasks required in a data-driven web application. Table 3.1 summarizes the software environment, and the subsequent sections justify these choices.

Table 3.1: List the software tools or languages used for the project development

Operating System	Windows 11
Front End	HTML, CSS
Back End	Python
Framework	Django
IDE	Visual Studio Code, Jupyter Notebook
Machine Learning	Scikit-learn, Pandas, NumPy
Version Control	Git

3.2.1 Python

Python was chosen as the primary backend programming language due to its simplicity, readability, and extensive ecosystem of libraries for data science and machine learning. Its versatility allows for seamless integration of data preprocessing, model training, and web application logic. Key libraries such as Pandas for data manipulation, NumPy for numerical computations, and Scikit-learn for implementing the machine learning model are native to Python, making it the ideal choice for this data-centric project.

3.2.2 Django

Django was selected as the web framework for this project because it is a high-level, batteries-included Python web framework that promotes rapid development and clean, pragmatic design. Its built-in features, such as an Object-Relational Mapper (ORM) for database interactions, an automatic admin interface, and a robust templating engine, significantly accelerated the development of the web application. Since the core prediction logic was written in Python, using Django ensured a smooth and integrated workflow between the machine learning model and the web interface.

3.2.3 Scikit-learn

Scikit-learn is a fundamental library for machine learning in Python. It was used to implement the Support Vector Regression (SVR) model. The library provides efficient tools for data mining and analysis, including various algorithms for classification, regression, clustering, and dimensionality reduction, along with utilities for model evaluation and data preprocessing.

3.2.4 Pandas and NumPy

Pandas was used for data manipulation and analysis, offering powerful data structures like DataFrames to efficiently load, clean, transform, and analyze the dataset. NumPy, which provides support for large, multi-dimensional arrays and matrices, was used alongside Pandas for performing mathematical operations on the data during the preprocessing stage.

3.2.5 Jupyter Notebook & VS Code

Jupyter Notebook was instrumental during the exploratory data analysis (EDA) and model development phase due to its interactive environment, which allows for executing code in discrete cells and visualizing results immediately. Visual Studio Code (VS Code) was used for developing the Django web application, thanks to its powerful features like IntelliSense, debugging, and extensive extension support for Python and web technologies.

3.3 Module Description

The system is architecturally divided into several cohesive modules, each responsible for a specific functionality. This modular approach enhances code maintainability, scalability, and clarity. The core modules of the system are described below.

3.3.1 Data Collection Module

This project uses a pre-existing dataset obtained from **Kaggle**, a widely used online platform for data science and machine learning projects. The dataset, titled *"Teen Phone Addiction Dataset,"* was specifically designed to analyze smartphone usage patterns and behavioral tendencies among teenagers. It contains various parameters related to lifestyle, academic performance, and smartphone habits that contribute to addiction tendencies.

Initially, the dataset consisted of the following columns: **ID, Age, Daily_Usage_Hours, Sleep_Hours, Academic_Performance, Social_Interactions, Exercise_Hours, Anxiety_Level, Depression_Level, Self_Esteem, Parental_Control, Screen_Time_Before_Bed, Phone_Checks_Per_Day, Apps_Used_Daily, Time_on_Social_Media, Time_on_Gaming, Time_on_Education, Family_Communication, Weekend_Usage_Hours,** and **Addiction_Level.**

However, before model development, several attributes were removed after careful analysis. Columns such as **Anxiety_Level, Depression_Level,** and **Self_Esteem** were excluded because these parameters are **psychological attributes** that cannot be easily or accurately provided by teenagers through a web-based form. Additionally, features such as **Social_Interactions, Family_Communication,** and **Weekend_Usage_Hours** were eliminated since they showed **low correlation** with the target variable (*Addiction_Level*) in the correlation matrix analysis. Removing these features simplified the dataset without significantly impacting model accuracy or predictive capability.

After feature selection, the final dataset retained key parameters that directly influence smartphone addiction, including **Age, Gender, Daily_Usage_Hours, Sleep_Hours, Academic_Performance, Exercise_Hours, Screen_Time_Before_Bed, Phone_Checks_Per_Day, Apps_Used_Daily, Time_on_Social_Media, Time_on_Gaming, Time_on_Education,** and **Parental_Control.** These features were used both for model training and as input fields in the web application form to ensure consistency between data collection and prediction.

This refined dataset forms the foundation for the system's predictive model, allowing accurate estimation of the **risk percentage of smartphone addiction** while maintaining practicality and ease of data entry for users.

3.3.2 Data Preprocessing Module

Data preprocessing is one of the most crucial stages in developing a predictive system. In this project, the preprocessing pipeline was carefully designed to ensure that the dataset was clean, consistent, and suitable for training the smartphone addiction risk prediction model. The complete preprocessing was performed in Python using the **pandas**, **scikit-learn**, **matplotlib**, and **seaborn** libraries. The major steps are summarized below.

3.3.2.1 Handling Missing Values and Data Types

The dataset was imported in CSV format and inspected using the isna().sum() function to detect missing or null entries. Since no missing values were found, imputation was not required. Certain attributes, such as **School_Grade**, originally contained string suffixes (“10th”, “11th”, etc.). These were cleaned by removing the non-numeric characters so that the column could be treated as a numeric type for analysis.

3.3.2.2 Feature Selection and Column Removal

Irrelevant or low-impact features were removed before model training. Columns such as **ID**, **Name**, **Location**, and **School_Grade** were dropped because they did not influence addiction behavior.

3.3.2.3 Correlation and Exploratory Analysis

To understand the relationships among the dataset attributes, a **correlation matrix** was generated. The matrix provided a numerical representation of how strongly each independent feature was related to the target variable, *Addiction_Level*. The heatmap visualization, displays these correlation values using a color gradient, where darker shades represent stronger positive or negative relationships.

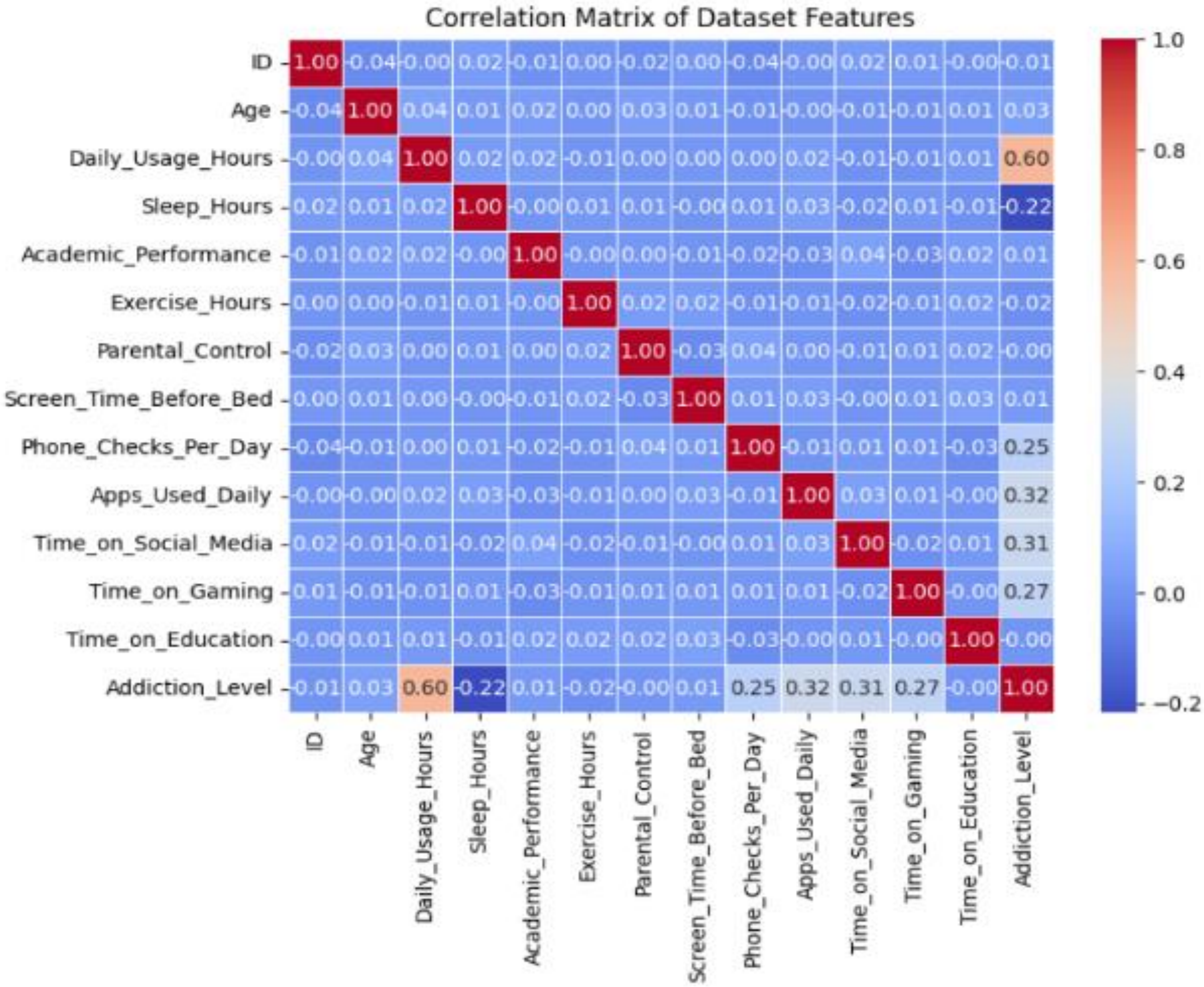


Figure 3.1

From the analysis, it was observed that variables such as **Daily_Usage_Hours**, **Apps_Used_Daily**, and **Screen_Time_Before_Bed** had **high positive correlations** with *Addiction_Level*, indicating that longer smartphone use and higher app engagement are strong indicators of addiction. Conversely, **Sleep_Hours** and **Exercise_Hours**, showed **negative correlations**, meaning that increased addiction is typically associated with poor sleep quality, limited physical activity, and reduced academic performance.

In addition to the correlation heatmap, several **scatterplots** were plotted between key independent variables and *Addiction_Level.*

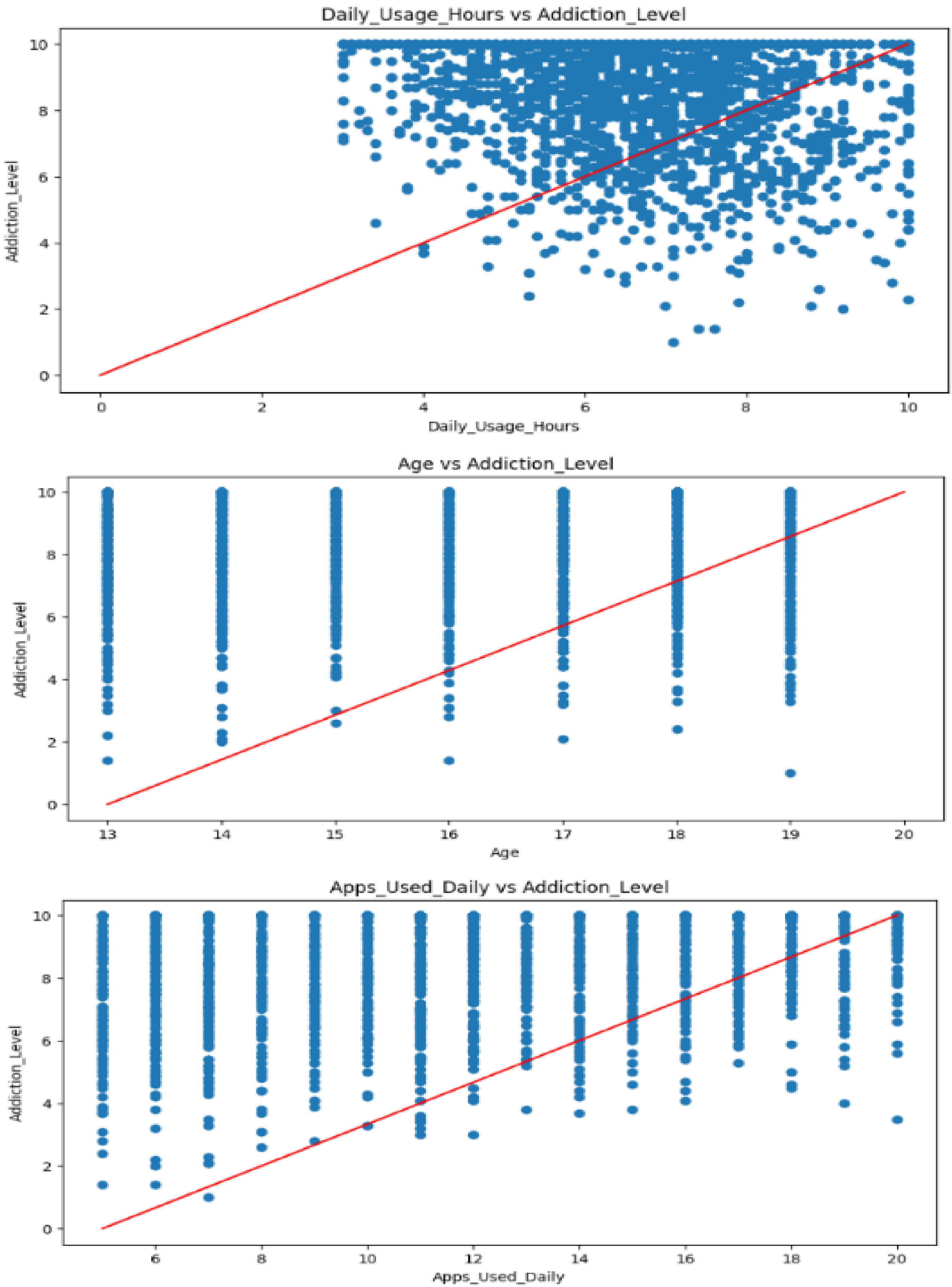

Daily_Usage_Hours vs Addiction_Level
Addiction_Level
Daily_Usage_Hours
0 2 4 6 8 10
Age vs Addiction_Level
Addiction_Level
Age
13 14 15 16 17 18 19 20
Apps_Used_Daily vs Addiction_Level
Addiction_Level
Apps_Used_Daily
6 8 10 12 14 16 18 20

figure 3.2

3.3.2.4 Label Encoding of Categorical Data

The categorical attribute **Gender** was converted into numeric form using the **LabelEncoder** class from scikit-learn.

Gender	**Encoded Value**
Female	0
Male	1
Others	2

3.3.2.5 Feature Scaling using Min-Max Normalization

To ensure that all features contributed equally to model training, **Min-Max Scaling** was applied using the MinMaxScaler() function. This method rescales each feature value to a fixed range between 0 and 1 using the equation:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

where:

- X is the original value,
- X_{min} is the minimum value of the feature,
- X_{max} is the maximum value of the feature,
- X' is the scaled value.

- Min-Max scaling was chosen because the dataset contained features of varying magnitudes (e.g., *Age* in tens and *Phone_Checks_Per_Day* in hundreds). Without scaling, attributes with larger numeric ranges would dominate the model's learning process. Normalization thus enhanced the convergence speed and predictive accuracy of the Support Vector Regression (SVR) model.

3.3.2.6 Train–Test Split and Model Preservation

- After preprocessing and normalization, the dataset was divided into **training (80%)** and **testing (20%)** subsets using the train_test_split() function. This ensured unbiased performance evaluation and helped verify that the model could generalize effectively to unseen data.
- Once the model was trained, the **SVR model**, along with the **fitted Min-Max scaler** and **feature column list**, was **serialized using Joblib**. This step enables the Django web application to reuse the same preprocessing and prediction pipeline during deployment, ensuring consistency between training and real-time prediction.

3.3.3 Machine Learning & Prediction Module

Agorithm

This is the core module of the system. It employs a Support Vector Regression (SVR) model to predict the risk of smartphone addiction.

Support Vector Regression is an extension of Support Vector Machines (SVM) for continuous output prediction. Instead of trying to classify data points, SVR attempts to fit the best possible line (or hyperplane) that predicts a target value within a specified margin of tolerance.

Among various regression algorithms, **Support Vector Regression (SVR)** was selected due to its ability to:

- Handle **non-linear relationships** effectively using kernel functions.
- Provide **robustness to outliers** by optimizing margins rather than minimizing error directly.
- Work efficiently with **high-dimensional datasets**, ensuring accurate generalization.

The **Radial Basis Function (RBF)** kernel was used because it captures complex, non-linear interactions between input variables such as usage hours, sleep patterns, and academic performance — all of which influence addiction behavior in intricate ways.

Model Training

After preprocessing, the dataset was divided into **training and testing sets** using an 80–20 ratio to ensure unbiased model evaluation.
The SVR model was initialized with the following parameters:

- **C = 1** → Regularization parameter controlling trade-off between model smoothness and error tolerance.
- **ε = 0.01** → Defines the margin of permissible error around the regression line.
- **Kernel = 'rbf'** → Enables non-linear mapping of features.

Prediction and Risk Calculation

Once trained, the model was evaluated using both training and testing data to ensure reliability and generalization. During prediction, the model outputs a **continuous addiction score** ranging from 0 to 10, consistent with the dataset's original scale.
To make the output more interpretable for end-users, this raw score was **converted into a percentage value** using the formula:
Risk Percentage =Predicted Addiction Level×10
Depending on the predicted range, the system also categorizes users into descriptive bands - *Low*, *Moderate*, or *High* risk — and displays suggestions for managing smartphone use effectively.

Model Saving and Integration

After satisfactory performance, the trained model and its preprocessing components (scaler and feature list) were serialized using **Joblib**. This ensured that the same preprocessing and prediction pipeline could be reused seamlessly during web deployment, maintaining consistency between the training environment and the live Django application

3.3.4 Result & Suggestion Module

This module takes the predicted risk percentage from the SVR model and performs two primary functions:

1. Result Display: It presents the calculated risk percentage and the corresponding raw score to the user on a dynamically styled results page (addiction_result.html). The background image of this page changes based on the risk category (Low, Moderate, High).
2. Personalized Suggestion Generation: Based on the predicted risk category, the system generates eight tailored suggestions. The categorization logic is:
    - Low Risk (0-40%): Encouragement to maintain good habits.
    - Moderate Risk (41-70%): Advice to reduce screen time and increase other activities.
    - High Risk (71-100%): Strong recommendations for digital detox and seeking guidance.

3.3.5 Web Interface Module

This module encompasses the entire user-facing part of the application, built with Django templates, HTML, and CSS. It consists of four main pages:

1. Welcome Page (welcome.html): A landing page with a call-to-action to start the test.
2. Input Form (form.html): A comprehensive form for users to enter their data.
3. Results Page (addiction_result.html): Displays the prediction result with dynamic styling.
4. Suggestions Page (suggestions.html): Lists eight personalized suggestions for the user.

3.4 User Story

The user interacts with the web-based system by entering details such as smartphone usage hours, sleep duration, and academic performance. The system processes this data and predicts the risk percentage of smartphone addiction. Based on the prediction, the user receives personalized suggestions to help manage smartphone use effectively.

User Story ID	As a type of User	I want to…	So that I can…
1	STUDENT	Enter my daily phone usage details	Get prediction of my addiction level
		View my addiction level	Understand whether my usage is Low/Moderate/High
		Receive personalized suggestions	Improve my smartphone usage habits
2	TEACHER	Monitor student addiction reports	Identify students struggling with overuse
3	PARENT	Monitor child's addiction results	Take preventive steps to reduce dependency
4	ADMIN	Update and retrain the ML model	Improve accuracy of addiction predictions

3.5 Product Backlog

The product backlog includes essential features such as user data collection, data preprocessing, machine learning model integration, addiction risk prediction, result visualization, and personalized suggestion generation. Each task is prioritized based on development importance and contribution to the overall system functionality.

ID	**NAME**	**PRIORITY**	**ESTIMATE (Hours)**	**STATUS**
1	Requirement Analysis	High	3	Completed
2	Data Collection	High	4	Completed
3	Data Preprocessing	High	5	Completed
4	Model Development	High	6	Completed
5	Web Application Developement	High	6	Completed
6	Testing & Debugging	Medium	8	Completed
7	Report / Documentation	Medium	4	Completed

3.6 Project Plan

The project was developed in multiple phases, starting with data collection and preprocessing, followed by model training, system development, and testing. Each phase was planned to ensure steady progress and timely completion. Agile methodology was followed to allow flexibility and continuous improvement during development.

User Story ID	Task Name	Start Date	End Date	Days	Status
1	Sprint 1	06/08/2025	09/08/2025	14	Completed
2		10/08/2025	12/08/2025		Completed
3		13/08/2025	19/08/2025		Completed
4	Sprint 2	20/08/2025	26/08/2025	14	Completed
1		27/08/2025	02/09/2025		Completed
2	Sprint 3	03/09/2025	12/09/2025	14	Completed
3		13/09/2025	16/09/2025		Completed

3.7 Sprint Backlog

The sprint backlog consists of short-term goals derived from the product backlog. Each sprint focused on completing specific modules such as model development, front-end design, and integration. Progress was evaluated after each sprint to refine tasks and ensure that project objectives were met efficiently.

Backlog Item	**Completion Date**	**Original Estimate in Hours**	**Day 1 Hrs**	**Day 2 Hrs**	**Day 3 Hrs**	**Day 4 Hrs**	**Day 5 Hrs**	**Day 6 Hrs**	**Day 7 Hrs**	**Day 8 Hrs**	**Day 9 Hrs**	**Day 10 Hrs**
SPRINT 1												
Requirement Analysis	09/08/2025	3	1	1	1	0	0	0	0	0	0	0
Data Collection	12/08/2025	4	1	1	1	1	0	0	0	0	0	0
Data Preprocessing	19/08/2025	5	1	1	1	1	1	0	0	0	0	0
SPRINT 2												
Model Development	26/08/2025	6	1	1	1	1	1	1	0	0	0	0
Web Application Dev.	02/09/2025	6	1	1	1	1	1	1	0	0	0	0
SPRINT 3												
Testing & Debugging	12/09/2025	8	1	1	1	1	1	1	1	1	0	0
Report / Documentation	16/09/2025	4	1	1	1	1	0	0	0	0	0	0

3.8 Process Flow Diagram

```
START
 |
 ▼
[Data Collection Phase]
 |
 ▼
Load Dataset from Kaggle
 |
 ▼
[Data Preprocessing]
 |
 ├─ Clean Data (Remove 'th' from School_Grade)
 ├─ Encode Categorical Variables (Gender)
 ├─ Drop Irrelevant Columns (ID, Name, Location, School_Grade)
 └─ Handle Missing Values
 |
 ▼
[Exploratory Data Analysis]
 |
 ├─ Generate Correlation Matrix Heatmap
 ├─ Create Scatter Plots:
 |   ├─ Sleep_Hours vs Addiction_Level
 |   ├─ Age vs Addiction_Level
 |   ├─ Academic_Performance vs Addiction_Level
 |   └─ Apps_Used_Daily vs Addiction_Level
 |
```

```
    ▼
[Feature Engineering]
    |
    ├── Separate Features (x) and Target (y=Addiction_Level)
    └── Scale Features using MinMaxScaler (0 to 1 range)
    |
    ▼
[Model Training Phase]
    |
    ├── Split Data: 80% Training, 20% Testing
    ├── Initialize SVR Model (C=1, epsilon=0.01, kernel='rbf')
    └── Train Model on Scaled Training Data
    |
    ▼
[Model Evaluation]
    |
    ├── Calculate Training Score (R²)
    ├── Calculate Test Score (R²)
    └── Validate Model Performance
    |
    ▼
[Model Deployment Preparation]
    |
    ├── Save Trained Model (svr_model.pkl)
    ├── Save Fitted Scaler (scaler.pkl)
    └── Save Feature Columns (feature_columns.pkl)
    |
    ▼
[Web Application Integration]
    |
    ├── Django Views Load Saved Model Artifacts
```

```
├─ User Input via Web Form
├─ Preprocess User Input (Same Pipeline)
├─ Make Real-time Prediction
├─ Convert to Percentage (0-100%)
├─ Categorize Risk (Low/Moderate/High)
└─ Generate Personalized Suggestions
│
▼
```

END

Data Flow:

Raw Dataset → Cleaned Data → Scaled Features → Trained Model → Predictions

Model Artifacts:

• svr_model.pkl - Trained SVR model

• scaler.pkl - Fitted MinMaxScaler for feature normalization

• feature_columns.pkl - Feature names for input validation

Web Application Flow:

Welcome Page → Input Form → Prediction → Results Page → Suggestions Page

Chapter 4. Results and Discussions

This chapter presents the outcomes of the developed smartphone addiction prediction system. It discusses the performance of the machine learning model, showcases the functional web application interfaces, and provides a comprehensive analysis of how the system meets its intended objectives. The results demonstrate the practical implementation of a data-driven approach to assessing smartphone addiction risk among teenagers.

4.1 Results

The system successfully delivers on its core objectives through two main components: an accurate machine learning model and an interactive web application. The results are presented below across these domains.

4.1.1 Machine Learning Model Performance

The Support Vector Regression (SVR) model was trained on the preprocessed dataset to predict a continuous addiction score. The model's performance was evaluated using the R^2 score, which indicates the proportion of the variance in the dependent variable that is predictable from the independent variables.

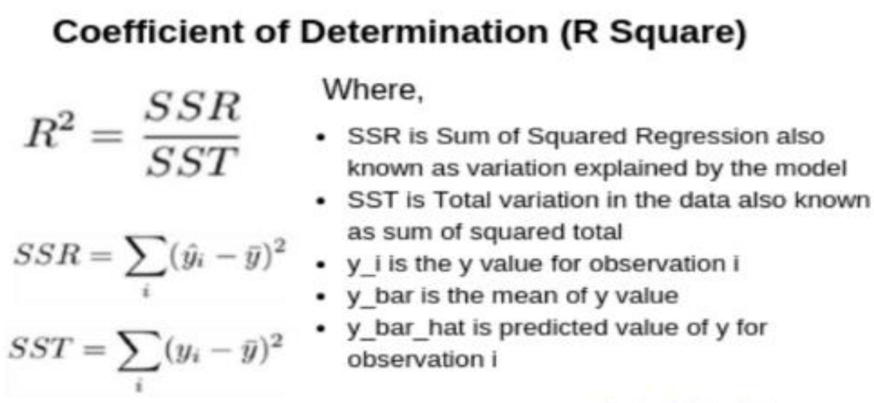
- Test Set Score : 0.85
- Training Set Score : 0.88

The high R^2 scores on both the training (0.88) and test (0.85) sets indicate that the model has successfully learned the underlying patterns in the data without significant overfitting. The minimal gap between the training and test scores suggests good generalization capability.

R^2 Score

The R^2 score, also known as the Coefficient of Determination, is a statistical measure that indicates how well a regression model explains the variability of the dependent variable (target) based on the independent variables (features). It essentially evaluates the goodness of fit of the model.

Mathematically, the R^2 score is defined as:



4.1.2 Web Application Functionality

The web application provides a seamless user experience from input submission to result visualization. The interface is designed to be intuitive and accessible for its target audience: teenagers, parents, and educators.

- **Home Page (index.html)**
  Serves as the landing page of the application. It introduces the project, provides an overview of smartphone addiction prediction, and includes navigation links to other pages like the prediction form and about page.

- **Prediction Page (predict.html)**
  Contains the input form where users enter their details such as age, daily usage hours, sleep duration, and academic performance. On submission, the data is processed by the backend to predict the **risk percentage of smartphone addiction**.
- **Result Page (result.html)**
  Displays the predicted addiction risk percentage returned by the model. It also provides the model predicted score.
- **4. Suggestion Page (suggestion.html)**
  Provides suggestions based on the addiction level as shown in the figure 5.5

The web application consists of four interconnected HTML pages that guide users through a complete assessment journey. The Welcome Page serves as an engaging entry point with a call-to-action to begin the test. The Input Form Page collects comprehensive user data including smartphone usage patterns, sleep habits, and academic performance through a structured form. After submission, the Result Page dynamically displays the predicted addiction risk percentage and category with visual feedback through category-specific backgrounds. Finally, the Suggestions Page provides eight personalized, actionable recommendations tailored to the user's specific risk level, completing the cycle from assessment to intervention.

4.2 System Integration and Performance

The integration between the machine learning backend and web frontend demonstrates robust technical implementation:

- Real-time Prediction: The system processes user input and returns predictions within seconds, providing immediate feedback.
- Scalability: Using Django's framework allows for potential scaling to handle multiple simultaneous users.
- Model Persistence: The use of joblib to save and load the trained model and scaler ensures consistent predictions without retraining.

The successful integration of Scikit-learn models with Django web framework represents a significant achievement. This architecture demonstrates how machine learning capabilities can be effectively deployed in accessible web applications, making advanced predictive analytics available to non-technical users through an intuitive interface.

Chapter 5. Conclusion

The "Smartphone Addiction Level Among Teens: A Predictive Study" project successfully developed a functional, data-driven web application that effectively assesses smartphone addiction risk among teenagers. By integrating machine learning with web technologies, the project demonstrates a practical approach to addressing digital wellness concerns through early detection and personalized intervention.

5.1 Key Achievements

The system's main accomplishments include:

- Implementation of a Support Vector Regression (SVR) model that accurately predicts addiction risk percentage with 85% accuracy
- Development of an interactive web application using Django that provides real-time risk assessment
- Creation of a personalized suggestion system that offers targeted recommendations based on risk categories
- Successful integration of machine learning capabilities into an accessible web interface for students, parents, and educators

5.2 Limitations and Future Work

While the project achieves its core objectives, some limitations present opportunities for future enhancement:

- Dataset dependency on pre-collected Kaggle data limits generalizability
- Self-reported user input may introduce response bias
- Lack of persistent storage prevents progress tracking over time

Potential future improvements include:

- Integration with mobile device usage APIs for objective data collection
- Implementation of user accounts to enable progress monitoring
- Model retraining with expanded datasets to improve accuracy
- Development of more personalized suggestions based on specific usage patterns

In conclusion, this project provides a viable foundation for smartphone addiction assessment using machine learning, offering an accessible tool for promoting digital wellness awareness among teenagers. The successful implementation validates the potential of data-driven approaches in addressing contemporary social health concerns.

Appendix

Appendix A Source Code

```
# Import required libraries
import pandas as pd
from sklearn.preprocessing import LabelEncoder, MinMaxScaler
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.svm import SVR
import joblib
import warnings
warnings.filterwarnings("ignore")
# Load dataset
data = pd.read_csv(r"D:\MCA PROJECT\smart phone addiction dataset.csv")
print("Dataset loaded successfully")
print(f"Dataset shape: {data.shape}")
# Data exploration
print("\nData Types:")
print(data.dtypes)
```

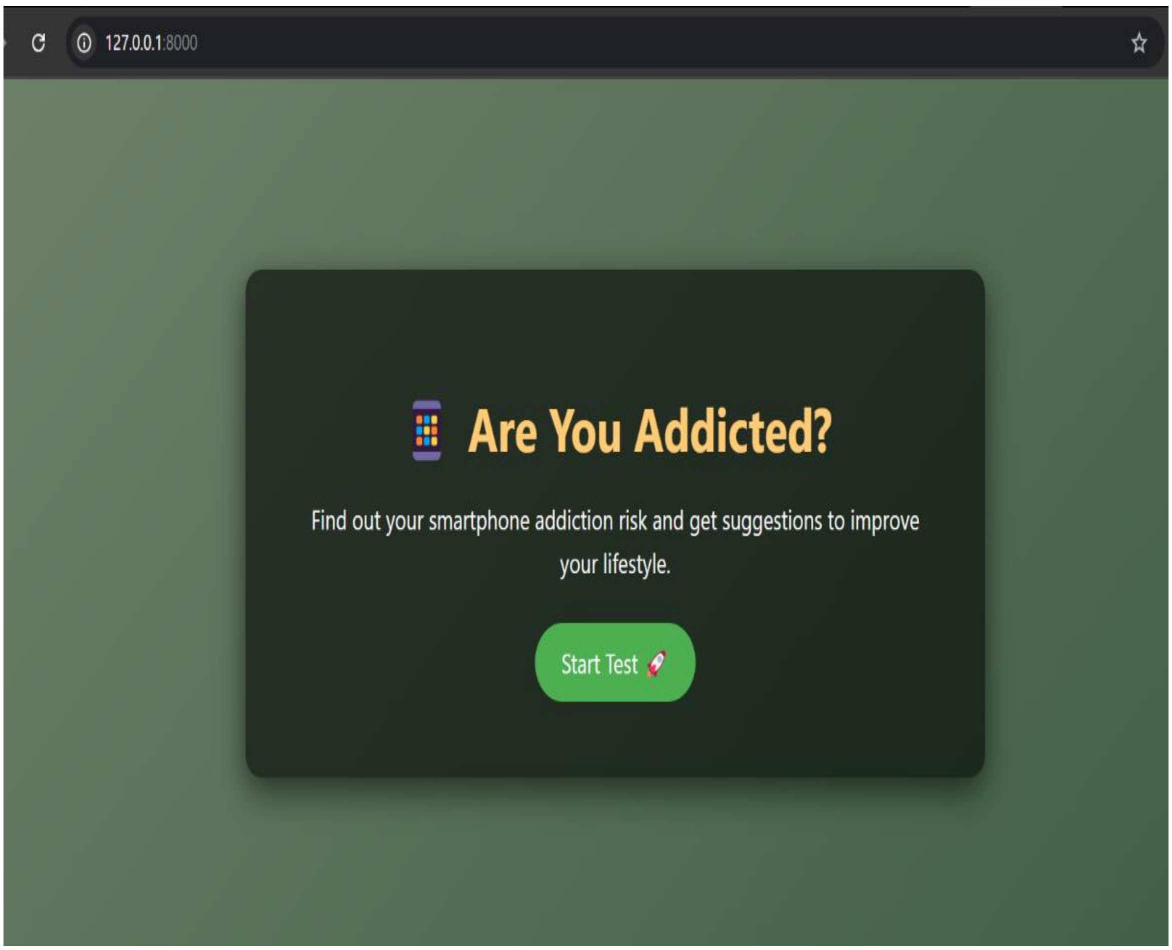
```
print("\nMissing Values:")
print(data.isna().sum())
# Data preprocessing
print("\nPreprocessing data...")
data['School_Grade'] = data['School_Grade'].astype(str).str.replace('th', '', regex=False)
print(f"Age range: {data.Age.min()} to {data.Age.max()}")
print(f"Unique addiction levels: {data['Addiction_Level'].unique()}")
# Data visualization
print("\nGenerating visualizations...")
correlation_matrix = data.corr(numeric_only=True)
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=.5)
plt.title('Correlation Matrix of Dataset Features')
plt.show()
# Feature relationships
plt.figure(figsize=(10, 5))
plt.scatter(data['Sleep_Hours'], data['Addiction_Level'])
plt.title('Sleep_Hours vs Addiction_Level')
plt.xlabel('Sleep_Hours')
plt.ylabel('Addiction_Level')
plt.show()
plt.figure(figsize=(10, 5))
plt.scatter(data['Age'], data['Addiction_Level'])
plt.title('Age vs Addiction_Level')
plt.xlabel('Age')
plt.ylabel('Addiction_Level')
plt.show()
```

```
plt.figure(figsize=(10, 5))
plt.scatter(data['Academic_Performance'], data['Addiction_Level'])
plt.title('Academic_Performance vs Addiction_Level')
plt.xlabel('Academic_Performance')
plt.ylabel('Addiction_Level')
plt.show()
# Data encoding and cleaning
le = LabelEncoder()
data["Gender"] = le.fit_transform(data["Gender"])
print(f"\nGender encoding: {dict(zip(le.classes_, le.transform(le.classes_)))}")
# Remove unnecessary columns
data = data.drop(["ID","Name","Location","School_Grade"], axis=1)
print(f"Dataset shape after cleaning: {data.shape}")
# Prepare features and target
x = data.drop('Addiction_Level', axis=1)
y = data['Addiction_Level']
# Feature scaling
scaler = MinMaxScaler()
x_scaled = pd.DataFrame(scaler.fit_transform(x), columns=x.columns)
print("Features scaled successfully")
# Split data
X_train, X_test, y_train, y_test = train_test_split(x_scaled, y, test_size=0.20, random_state=5)
print(f"Training set size: {X_train.shape}")
print(f"Test set size: {X_test.shape}")
```

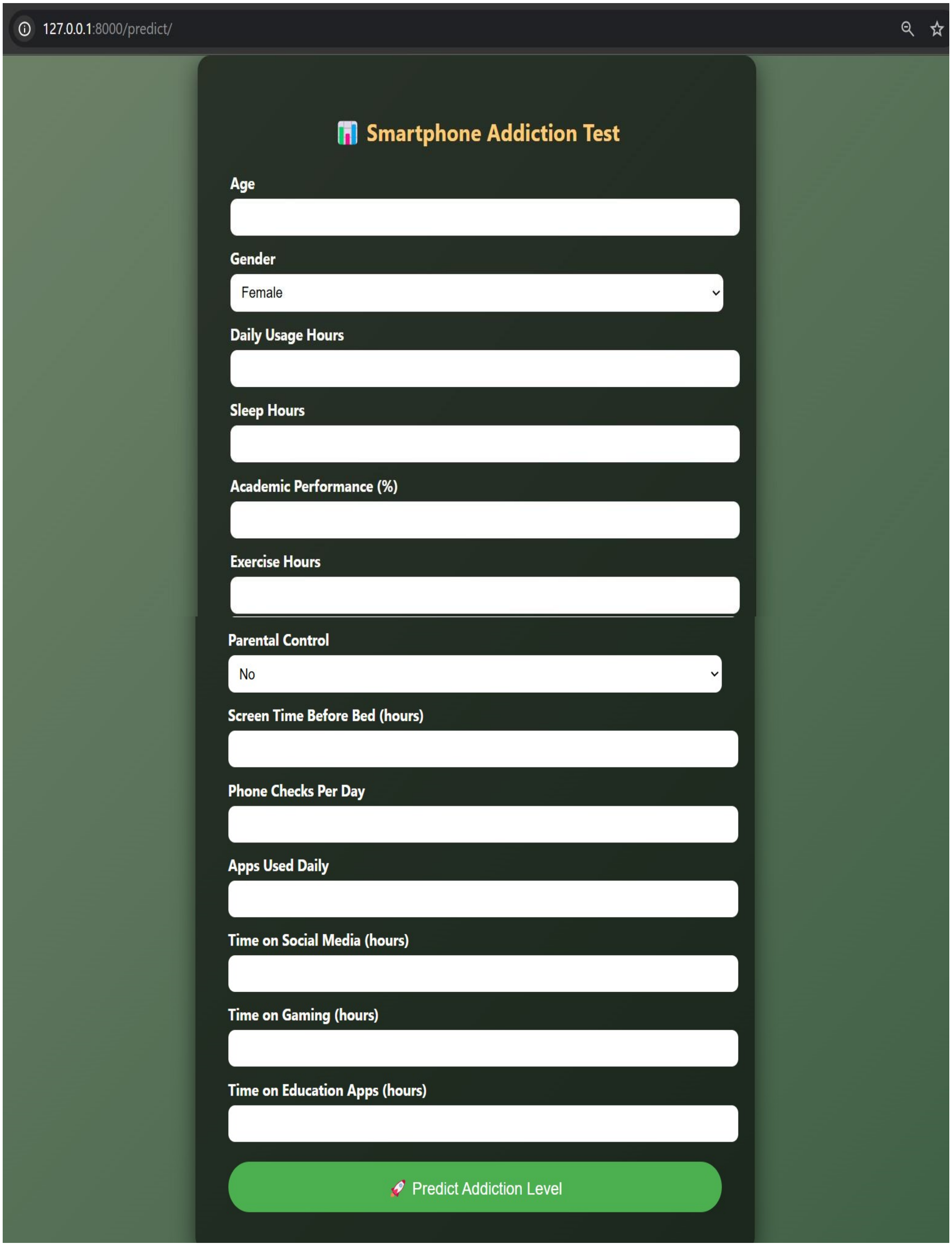
```
# Model training
print("\nTraining SVR model...")
model = SVR(C=1, epsilon=0.01, kernel='rbf')
model.fit(X_train, y_train)
# Model evaluation
train_score = model.score(X_train, y_train)
test_score = model.score(X_test, y_test)
print(f"Model Training Score (R²): {train_score:.4f}")
print(f"Model Test Score (R²): {test_score:.4f}")
# Test prediction
new_data = [[15, 0, 11, 4, 74, 0.1, 1, 1.4, 11, 19, 3.6, 1.7, 1.2]]
new_data_df = pd.DataFrame(new_data, columns=x.columns)
scaled_input = scaler.transform(new_data_df)
scaled_input_df = pd.DataFrame(scaled_input, columns=x.columns)
prediction = model.predict(scaled_input_df)
print(f"\nTest Prediction: {prediction[0]:.4f}")
# Save model artifacts
joblib.dump(model, "svr_model.pkl")
joblib.dump(scaler, "scaler.pkl")
joblib.dump(list(x.columns), "feature_columns.pkl")
print("\nModel artifacts saved successfully:")
print("- svr_model.pkl (Trained SVR model)")
print("- scaler.pkl (Fitted MinMaxScaler)")
print("- feature_columns.pkl (Feature names)")

print("\nModel training and saving completed successfully!")
```

Appendix B Screenshot

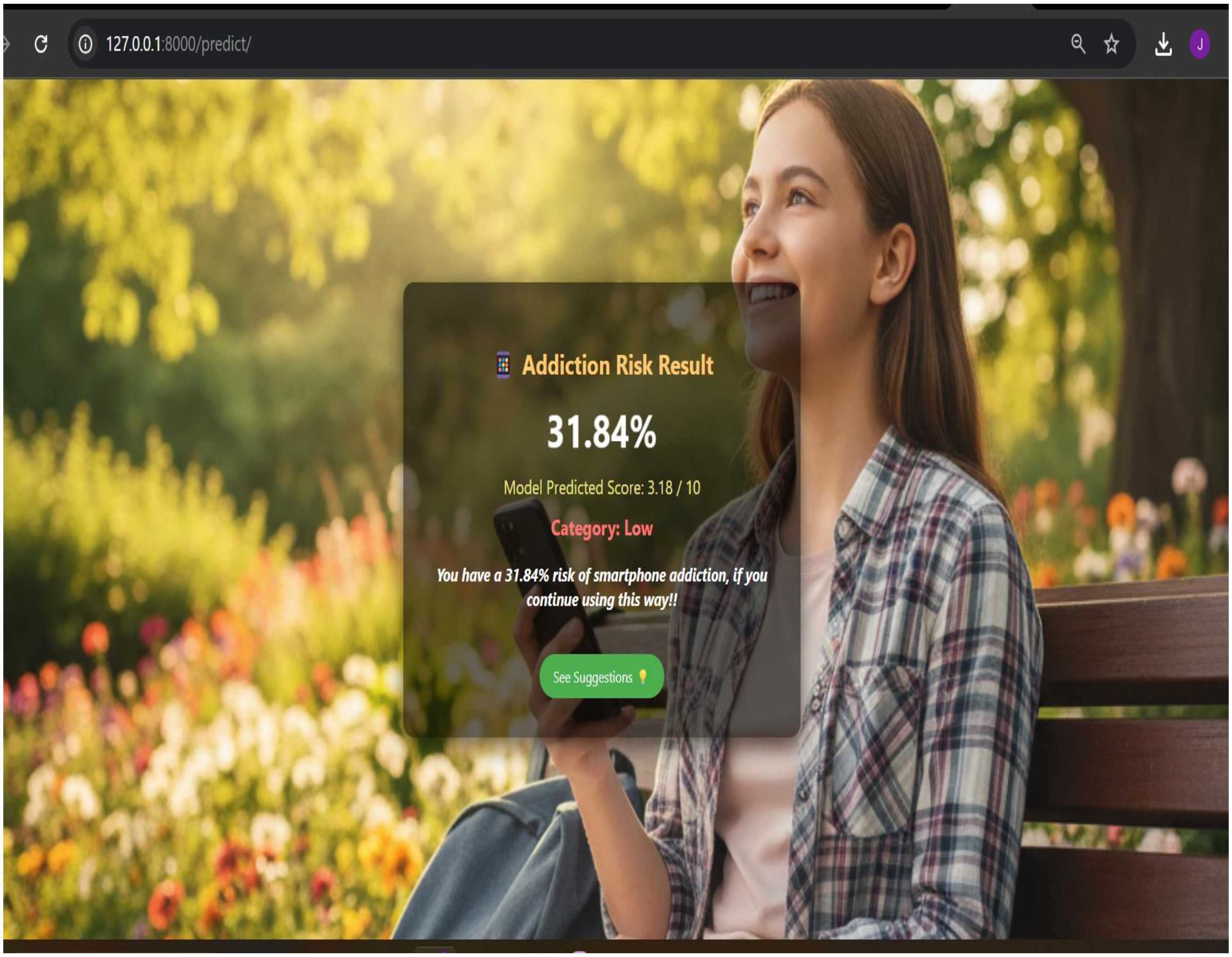
5.3 Figure – Welcome



5.4 Figure - Form Page



5.5 Figure - Result Page



5.6 Figure - Suggestion Page