Stress Level Prediction

PROJECT-II REPORT

Submitted By

Adnan Alvi

Enrolment No: 2021-310-021

In partial fulfillment for the award of the degree of

B. Tech (CSE)

Under the supervision of

Dr. Javed Ahmed

Department of Computer Science & Engineering

JAMIA HAMDARD

New Delhi – 110062

(2024)

ABSTRACT

Stress Level Prediction: Monitoring Well-Being Through Data Insights

Stress Level Prediction is a user-friendly Python-based application developed to utilize the potential of machine learning in estimating stress levels. Built with Python and Streamlit, the application integrates a pre-trained machine learning model to provide an interactive and effective solution for predicting stress levels. By analyzing inputs such as humidity, body temperature, and step count, users gain valuable insights into their stress levels, enabling proactive management of their health.

Key Features:

Stress Level Estimation: Predicts stress levels based on physiological data, categorizing them into Low, Medium, and High levels.

Interactive Input Fields: Accepts user inputs for humidity, body temperature, and step count through an intuitive form-based interface.

Machine Learning Integration: Employs a pre-trained model for efficient and accurate predictions of stress levels.

Real-Time Feedback: Provides immediate results with user-friendly explanations and visual feedback.

Simple and Accessible UI: Designed using Streamlit, the interface ensures a sleek, responsive, and engaging user experience.

This application simplifies the process of monitoring stress, combining machine learning precision with a modern, intuitive interface to enhance user engagement and utility.

DECLARATION

I, **Adnan Alvi** a student of **B. Tech (CSE) (Enrollment No.: 2021-310-021)** hereby declare that the Project entitled **"Stress Level Prediction: Monitoring Well-Being Through Data Insights"** which is being submitted by me to the Department of Computer Science, Jamia Hamdard, New Delhi in partial fulfillment of the requirement for the award of the degree of **B. Tech (CSE),** is my original work and has not been submitted anywhere else for the award of any Degree, Diploma, Associateship, Fellowship or other similar title or recognition.

Sign:

Name: Adnan Alvi

Date: 20/11/24

Place: New Delhi

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to all those who have contributed to the successful completion of this project. Firstly, I would like to express my deepest appreciation to my supervisor, **Dr. Javed Ahmed**, for his valuable guidance, insightful feedback, and constant support throughout the project. His expertise and encouragement have been instrumental in shaping this project into its final form.

I am also grateful to the Department faculty and staff for their assistance and support throughout the project. Their expertise and guidance have been invaluable in helping me to understand the technical aspects of the project and overcome any challenges that I encountered.

Furthermore, I would like to extend my thanks to my colleagues and friends who have provided me with support and encouragement throughout the project. Their feedback, discussions, and critiques have been valuable in shaping my ideas and approach to the project.

Finally, I would like to express my appreciation to my family for their unwavering support and encouragement throughout the project. Their love and support have been a constant source of motivation for me.

Sign:

Name: Adnan Alvi

Date: 20/11/24

Place: New Delhi

Table of Content

1. INTRODUCTION TO STRESS LEVEL PREDICTION

1.1. Objective

The primary objective of the **Stress Level Prediction** Application is to provide users with an efficient and accessible tool for monitoring their stress levels. By leveraging machine learning, this application simplifies the process of stress assessment using physiological and activity-based parameters such as humidity, body temperature, and step count. The project aims to empower users with real-time insights, promoting proactive health management in a seamless and user-friendly manner.

The specific goals of this project are:

- **Accurate Stress Estimation:** Utilize a pre-trained machine learning model to predict stress levels with high accuracy.
- **Intuitive User Interface:** Build a simple, interactive interface using Streamlit for ease of use across diverse user groups.
- **Real-Time Feedback:** Provide instant results based on user inputs, enabling quick assessment of stress levels.
- **Efficiency and Accessibility:** Ensure the application runs efficiently with minimal computational overhead.
- **Health Awareness:** Support users in understanding and addressing stress by delivering actionable insights through data-driven predictions.

1.2. Introduction

Stress Level Prediction: A Smart Approach to Well-Being

Stress has become a significant health concern, impacting individuals across all walks of life. The need for accessible and reliable tools to monitor stress levels has never been greater. The Stress Level Prediction Application emerges as a practical solution, empowering users with insights into their stress conditions through an easy-to-use interface powered by machine learning.

Whether users want to track daily stress indicators, monitor changes in physical activity, or understand how external factors influence their well-being, this application simplifies the

process. By combining a lightweight machine learning model with a sleek Streamlit interface, the application ensures both accuracy and convenience.

Key Features:

1. **Stress Level Estimation:** Predicts stress levels (Low, Medium, High) using humidity, body temperature, and step count as inputs.
2. **Interactive Input Fields:** Offers an intuitive form-based interface for entering parameters.
3. **Machine Learning Integration:** Employs a pre-trained model to ensure efficient and accurate predictions.
4. **Real-Time Feedback:** Delivers immediate stress analysis, fostering proactive health management.
5. **User-Friendly Design:** A responsive and accessible interface ensures ease of use for all demographics.

Why Choose the Stress Level Prediction Application?

- **Efficiency:** Delivers instant predictions with minimal computational requirements.
- **Accuracy:** Leverages advanced machine learning techniques for reliable insights.
- **Simplicity:** Provides an easy-to-navigate interface, reducing user learning curves.
- **Proactivity:** Helps users take actionable steps towards improving their mental and physical health.

1.3.Problem Statement

Stress is a widespread issue that significantly impacts both mental and physical well-being. Monitoring stress levels is essential for preventing long-term health issues, but current methods are often either invasive, time-consuming, or require specialized equipment. Additionally, most available tools do not integrate seamlessly into everyday life, creating a gap between the need for stress assessment and the solutions available.

The Stress Level Prediction Application bridges this gap by offering a practical, user-friendly, and non-invasive way to monitor stress levels. Utilizing machine learning, the

application processes easily available data—humidity, body temperature, and step count—to provide instant insights into stress levels.

Key challenges addressed by this application:

Time Efficiency: Traditional stress-monitoring methods are often slow and cumbersome.

Accessibility: Many solutions are inaccessible to non-expert users due to their complexity or cost.

Accuracy: Manual or non-digital methods are prone to errors, limiting reliability.

This application delivers a unified solution that simplifies stress monitoring for individuals of all technical backgrounds, empowering them to take control of their health using cutting-edge technology.

2. Software Requirements Specification (SRS)

Project Name: Stress Level Prediction

Prepared by: Adnan Alvi

Date: 20-11-2024

2.1. Introduction

2.1.1. Purpose

The purpose of this document is to define the functional, non-functional, and technical requirements for the Stress Level Prediction Application, a machine learning-powered Python application designed to predict stress levels based on input parameters such as humidity, body temperature, and step count.

2.1.2. Scope

The Stress Level Prediction Application aims to provide users with a simple and accessible tool for monitoring their stress levels. By integrating a pre-trained machine learning model, the application enhances health awareness and encourages proactive stress management. It targets individuals who seek a quick, reliable, and user-friendly method to assess their well-being.

2.1.3. Definitions, Acronyms, and Abbreviations

- ML: Machine Learning.
- Streamlit: An open-source Python framework for creating web-based applications.
- SRS: Software Requirements Specification.
- UI/UX: User Interface/User Experience.
- Pickle: Python library for object serialization.

2.1.4.Overview

This SRS document provides a detailed description of the application's functional and non-functional requirements, design constraints, system architecture, and dependencies. It serves as a reference for the design, development, and future enhancements of the project.

2.2. Functional Requirements

2.2.1. Stress Level Prediction Module

- Accepts three numerical inputs: Humidity, Body Temperature, and Step Count.
- Uses a pre-trained machine learning model to predict stress levels.
- Displays stress levels categorized into Low, Medium, and High.

2.2.2. Input Validation

- Ensures all inputs are numeric and within valid ranges.
- Displays error messages for invalid or incomplete inputs.

2.2.3. User Interface

- Provides a web-based interface developed using Streamlit.
- Includes a form for entering inputs and a submit button for predictions.
- Displays results clearly with visual feedback for stress categories.

2.3. Non-Functional Requirements

2.3.1. Performance

- Ensures predictions are delivered within 1 second of submission.
- Optimizes model loading to minimize application startup time.

2.3.2. Usability

- Offers a clean and intuitive interface for users with no technical expertise.
- Includes error messages and guidance for invalid inputs.

2.3.3. Security

- Does not store user data locally or on external servers.
- Ensures data confidentiality during processing.

2.3.4. Compatibility

- Supports major web browsers, such as Chrome, Firefox, and Edge.
- Works on both desktop and mobile devices with stable performance.

2.3.5. Scalability

- Allows for future integration of additional stress prediction features, such as real-time data from IoT devices.

2.4. System Architecture

- **Front-End:** Developed using Streamlit for the user interface.
- **Back-End:** Implements a pre-trained machine learning model serialized using Pickle.
- **Data Processing:** Processes inputs locally on the user's machine without requiring internet connectivity.

2.5. Design Constraints

- The application must operate offline for all features.
- Input fields must accept only valid numerical values for predictions.
- The interface design should prioritize simplicity and ease of navigation.

2.6. Assumptions and Dependencies

- Users will have a stable Python environment to run the application.
- The pre-trained model file (.sav) must be available in the specified directory for successful predictions.
- The application relies on the compatibility of libraries such as NumPy, Pickle, and Streamlit, and their updates may affect performance.

2.7. Appendices

2.7.1. Tools and Technologies

- Development Environment: Python 3.7+
- Framework: Streamlit
- Libraries: NumPy, Pickle

2.7.2. References

- Streamlit Documentation: Streamlit.io
- Pickle Library Documentation: Python.org
- Machine Learning References: General guides for training and deploying ML models.

3. Implementation and Result

The implementation of the stress detection model involves several key stages. Initially, the dataset, comprising 2001 samples with attributes such as body humidity, body temperature, and step count, is preprocessed. The target variable categorizes stress into three levels: low, normal, and high. Data cleaning includes analyzing numerical and categorical features, converting categorical data to numerical, handling missing values, and removing duplicates. Exploratory Data Analysis (EDA) is performed to identify trends and correlations in the data. Following preprocessing, machine learning models are trained using the cleaned dataset, leveraging techniques to ensure robust performance. Finally, the trained model is evaluated and deployed for practical stress level detection based on physical activity data**.**

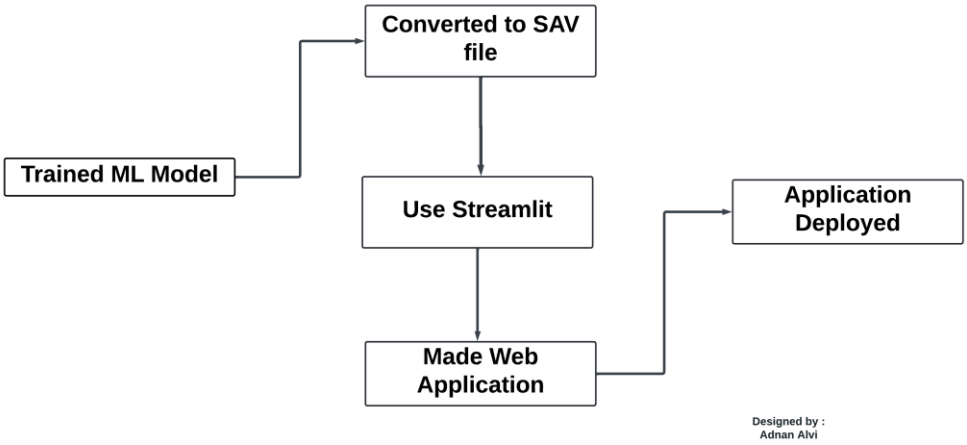
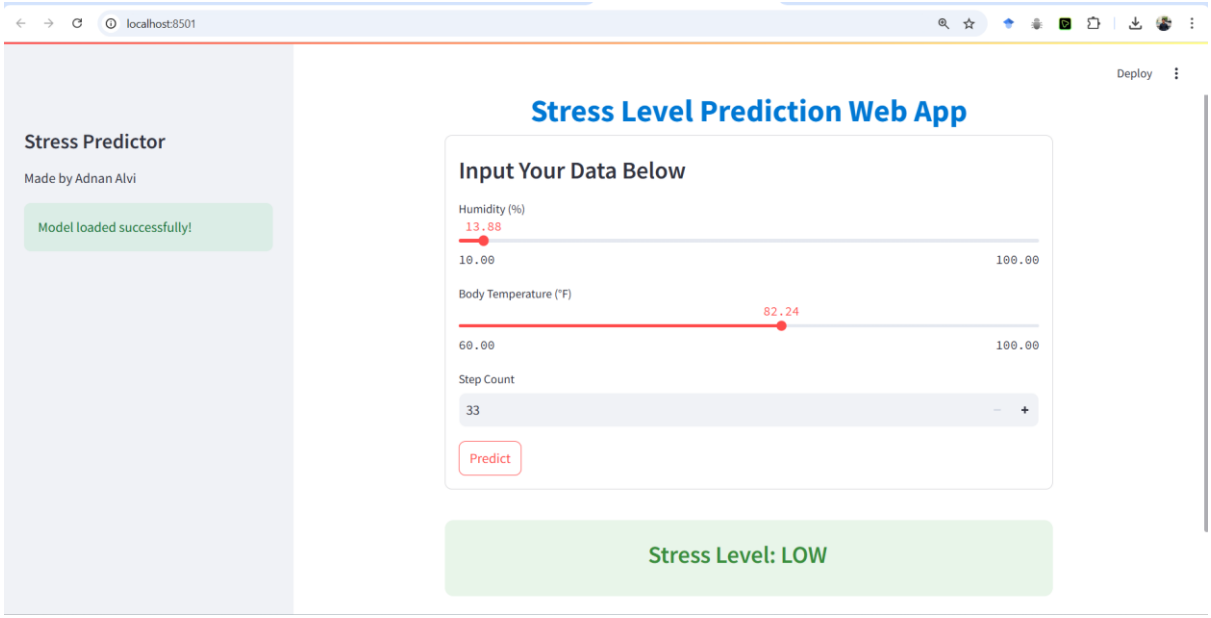


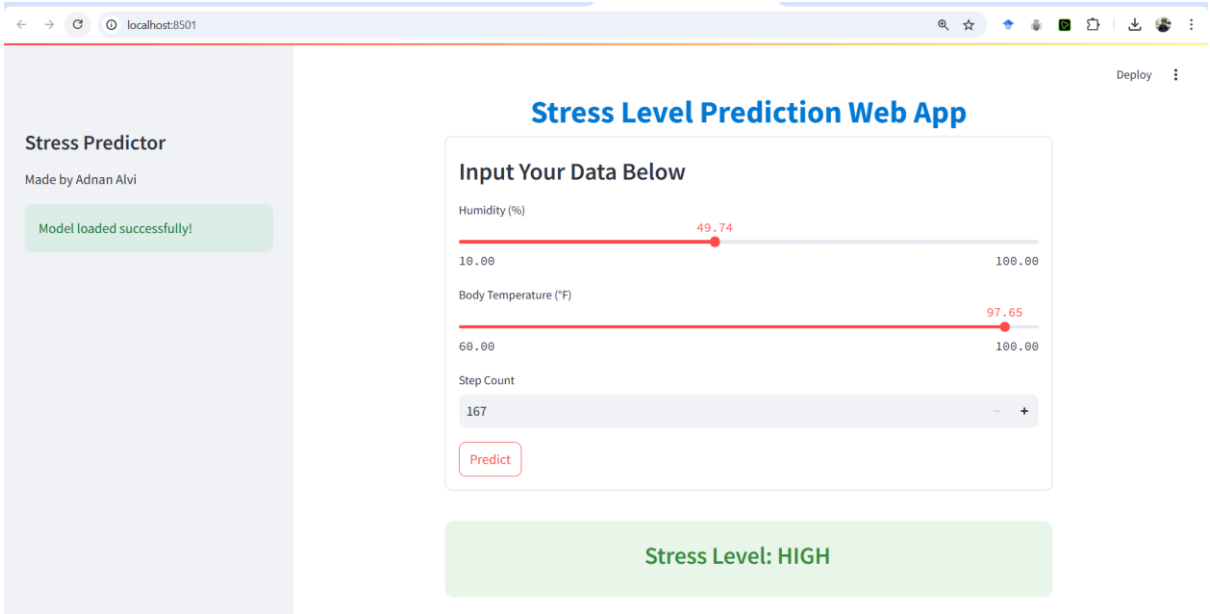
Figure 1: Deployment Process

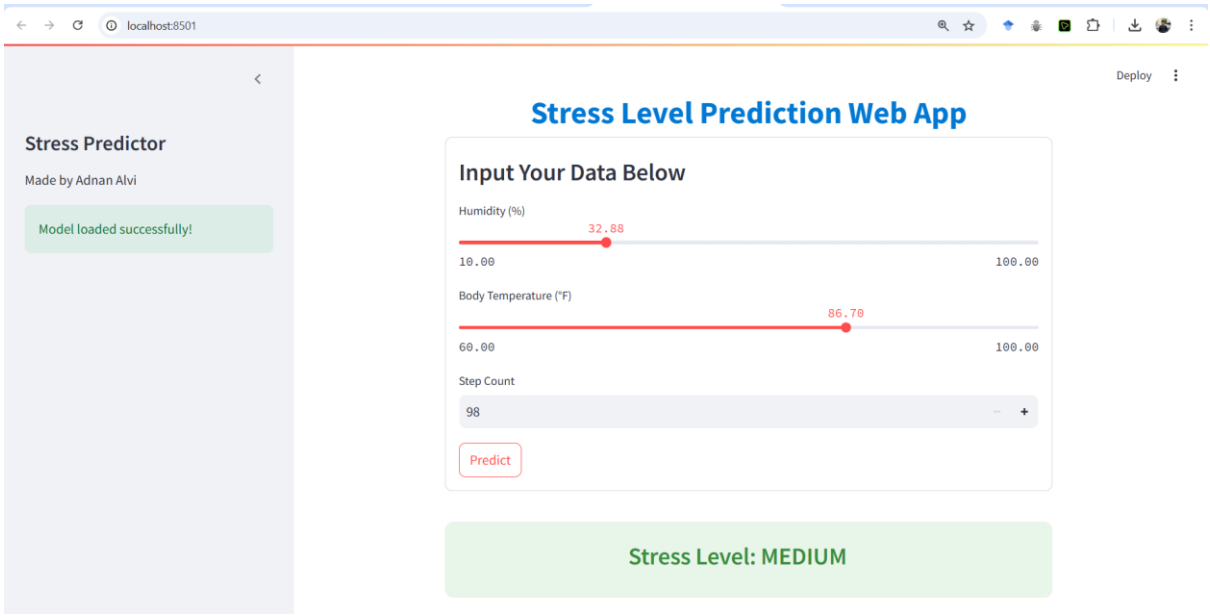
3.1.Results

The web application for stress level prediction was successfully developed and deployed, integrating a trained machine learning model for real-time stress analysis. The application features a clean, user-friendly interface that allows users to input key parameters such as humidity (%), body temperature (°F), and step count using interactive sliders and input fields. Once the data is submitted, the backend model processes the inputs and predicts the stress level, which is displayed prominently as "Low," "Normal," or "High" in a visually appealing format. The interface also includes status messages to confirm model loading and successful operation, ensuring a seamless user experience. As shown in the screenshots, the app

accurately reflects the stress levels based on the provided inputs, with different combinations of parameters yielding distinct results. This implementation showcases the practicality and accuracy of the model, demonstrating its potential use in health monitoring systems or stress management tools. The interactive design ensures usability for a diverse audience while effectively highlighting the application's predictive capabilities.







Conclusion

3.2.Innovation and Integration

The Stress Level Prediction Application showcases the potential of machine learning in health monitoring. By integrating a pre-trained machine learning model with an intuitive web-based interface, the application provides a seamless solution for predicting stress levels based on physiological and activity-based parameters. This innovation simplifies the process of stress assessment, making it accessible to a broad audience.

5.2.Technical Excellence

The development of this application demonstrates advanced proficiency in Python programming and machine learning integration. Utilizing the Streamlit framework ensures a responsive and user-friendly interface, while the back-end leverages efficient model serialization techniques using Pickle. The modular architecture of the application enables scalability and the potential for future enhancements.

5.3.Impact and Practicality

Designed to address real-world needs, the application offers a reliable solution for individuals seeking to monitor and manage their stress levels. By combining technical excellence with practical utility, the project not only highlights the benefits of machine learning in health applications but also provides a meaningful tool for personal well-being.

5.4.Future Scope

The scalable design of the application sets the stage for future developments, such as:
Integration with IoT devices for real-time data collection.
Expansion of prediction capabilities to include additional health indicators.
Improved customization options to enhance user engagement.
Offline support for greater accessibility and convenience.

6. Limitations

6.1. Dependency on Model Accuracy

The prediction accuracy depends on the quality of the pre-trained machine learning model. Errors or biases in the model could lead to inaccurate predictions.

6.2. Resource Utilization

Running the application on systems with limited resources may lead to slower performance, especially during model loading or prediction processing.

6.3. Input Validation Constraints

The application relies on accurate input data. Invalid or poorly formatted inputs could result in errors or mispredictions.

6.4. Limited Functionality

The current version focuses solely on stress level prediction. Additional health-related features, such as mental health assessments or stress-relief suggestions, are not yet implemented.

6.5. Learning Curve for Non-Tech Users

For users unfamiliar with web-based tools or machine learning applications, understanding the input format and interpreting the results may require guidance or an onboarding process.

References

1. Python Documentation

"Python Official Documentation." Python.org. Accessed for information on libraries like Streamlit and Pickle.

https://www.python.org

2. Streamlit Documentation

"Building Interactive Applications with Streamlit." Streamlit.io. Accessed for creating the user interface and interactive features.

https://streamlit.io

3. Machine Learning References

"Supervised Learning and Model Deployment." General guides for training and deploying machine learning models.

4. NumPy Documentation

"NumPy Array Manipulation and Calculations." NumPy.org. Referenced for data preprocessing techniques.

https://numpy.org