University of Essex
Department of Mathematical Sciences

MA981: DISSERTATION

Stress Detection In and Through Sleep

Shreyas Mandaliya
Registration Number: 2211461

Supervisior: Brawn, Dan

November 23, 2023
Colchester

Contents

List of Figures

List of Tables

Abstract

The escalating prevalence of stress poses a growing concern for public health, impacting individuals and jeopardizing their overall well-being. Common symptoms such as frustration, nervousness, and anxiety, indicative of stress, are increasingly pervasive, affecting approximately 40% of the younger population. This escalating trend highlights the urgent need for effective stress detection and prediction mechanisms to mitigate its detrimental effects on individuals' lives and performance.

Recognizing the critical role of early intervention in minimizing the impact of stress-related health issues, this study proposes an innovative approach that leverages machine learning techniques for accurate and efficient stress detection. The focal point of this proposed methodology is a hybrid model (HB) crafted through the fusion of Logistic Regression (LR) and Random Forest Classifier (RF). This amalgamation is strategically orchestrated using soft voting criteria, wherein the prediction probabilities of each model contribute to the final prediction.

The significance of the proposed HB model is underscored by its exemplary performance, boasting a remarkable 100% accuracy rate. In a landscape where accurate stress prediction is paramount, the proposed model emerges as a frontrunner, surpassing existing methodologies and setting a new standard for precision and reliability. The efficacy of the hybrid model is evident in its mean accuracy of 1.00 and a negligible standard deviation of +/−0.00.

The utilization of LR and RF within the hybrid framework is a strategic decision aimed at harnessing the strengths of both models. LR, a linear model, and RF, an ensemble of decision trees, offer complementary advantages that, when combined, create a synergistic effect.

This research not only contributes to the scientific understanding of stress prediction but also lays the foundation for future endeavors in leveraging machine learning for mental health applications.

Keywords: Machine learning, Hybrid approach, Stress Detection
Total Wordcount: 10336

CHAPTER 1

Introduction

Although there are various factors that may negatively influence one's mental health, everything about mental health determines an individual's overall health and capability to reach his/her objectives. Human beings always experience stress as a result of difficult circumstances and threats. Small amounts of stress are good for us because they can stimulate us to do better [1]. Long-term stress, however, does tend to affect our body and mind in general.

As per a [2] analysis, about 190 million people indicated that they were much more stressed out than before when compared to the year 2020. Forty percent of adults said they had bothered while worrying, and forty percent indicated that they had been under excessive stress occasionally. In 2020 the world was more depressed, angry, anxious and stressed compared to the past 15 years. The most stressful year in recent history was 2020, as stated by the survey results [2].

In such a modern society, health issues can also escape people's notice while they are hurrying against time to accomplish successes and be productive. Currently, life is often marked with competitiveness and a number of contemporary problems for example COVID-19 outbreak [3], excessive social networks use, professional difficulties, educational issues, financial hardships, and problems in relationships among other concerns.

This has resulted into increased cases of mental health problems like stress and despair. There are manual techniques to measure stress that require evaluations by medical professionals like psychiatrists and psychologists. Some of the merits of their methods are that they provide one on one and qualitative insights, they make an extensive assessment of stressors and also offers emotional support advice.

However, these have some major disadvantages. While manual is an essential part of stress evaluation is costly and can take a lot of time with its subjectivity nature. Besides, such methods may fail to reach most people and the process consumes a lot of time for the psychologist and his/her patient. To overcome some of these drawbacks and make stress tests more widely available and objective, there is rising interest in alternate methods of measuring stress, such as wearable technologies and machine learning models.

Several studies on stress detection using machine learning and deep learning approaches have proved to be helpful in stress management and understanding. For instance, emotional recognition based on textual content, physiological and heart rate data, picture and face analysis and social and behaviour analysis. These studies prove that deep learning and machine learning for stress detection can be flexible and prospective. We will discuss on some of these as well as other studies in the related-work section.

In this dissertation, I am using L. Rachakonda's groundbreaking stress detection studies. Therefore, a self-monitoring approach can be used for addressing stress in the scope of IoMT. SaYoPillow is a smart pillow purposed to monitor stress level during sleep and also enhance patient data confidentiality in order to promote 'smart-sleep'. This aims at creating awareness for health benefits associated with better sleeping to help reduce stress.

In the study, I suggest a stress detection strategy that, when compared to current state-of-the-art techniques, provides better outcomes in terms of efficiency and accuracy.

- In order to make notable gains over individual models, the hybrid model presented in this study's core integrates two machine learning models via a soft voting method.
- Moreover, this method increases the efficacy of stress level prediction by adding several target classes.
- Stress is divided into many categories: low/normal, medium-low, medium, medium-high, and high.
- This improved classification scheme adds to the overall effectiveness of the suggested strategy by offering a more thorough and nuanced assessment of a person's stress level.

The subsequent sections of this article delineate a comprehensive exploration of stress detection, starting with an in-depth review of pertinent literature in the "Related Work" section. This critical examination encompasses existing methodologies and insights into stress detection.

Following this, the "Methodology of Model Experimention" section delves into the dataset employed, experimental setup of the model for study. Moving forward, the "Results" section presents the outcomes derived from the application of machine learning in stress detection.

Concluding the article, the "Conclusion & Future Work" section synthesizes the findings, offering a comprehensive summary of the study's contributions and implications.

CHAPTER 2

Related Work

Stress detection is a critical area of research in today's fast-paced world, where people's mental and emotional well-being is constantly challenged by various stressors. Stress is a state of mental or emotional strain caused by demanding circumstances, and it can have far-reaching implications for both individual health and societal well-being. As a result, there has been extensive research on developing effective methods for stress detection and management.

2.1 Driver stress detection using a Motion Sensor

In this study [4], a wearable glove motion sensor with an inertial motion unit (IMU) is used to analyse the movement pattern of the steering wheel in order to forecast driver stress levels in a unique way. They carried out tests in a variety of driving conditions, such as city driving, interstate driving, and traffic congestion to assess the efficacy of the suggested strategy. The results showed that the approach outperformed traditional techniques based on biological signals, facial expressions, speech, and lane tracking, with accuracies of up to 94.78%.

They used the feature extraction process used to analyse steering wheel movement data. Time-domain features, such as steering wheel angle, angular velocity, and angular acceleration, were extracted to capture the dynamic characteristics of steering wheel movement.

Frequency-domain features, obtained using Fourier transform, were employed to identify patterns in the frequency spectrum of steering wheel movements. Phase-domain features, extracted using Hilbert transform, provided insights into the temporal relationships between different components of steering wheel movement.

The extracted features were fed into a Support Vector Machine (SVM) classifier for stress level prediction. The SVM classifier was chosen for its ability to handle high-dimensional data and its robustness to noise. To optimize the classifier's performance, a Stepwise Feature Selection method was employed to eliminate redundant features and reduce the dimensionality of the feature set.

2.2 Social Media based stress detection

Social media data, particularly from platforms like Twitter and Sina Weibo, has emerged as a valuable source of information for understanding and analyzing individuals' psychological stress levels. The research paper by [5] presents a hybrid model that combines a convolutional neural network (CNN) with a factor graph model (FGM) to effectively detect stress from social media data.

The proposed hybrid model, designated as FGM+CNN, leverages both tweet content attributes and social interactions to enhance stress detection performance. At the tweet level, the model considers linguistic features derived from text content, visual features from images attached to tweets, and social attention features based on user interactions. At the user level, the model incorporates user-level attributes such as demographic information, social connections, and past stress patterns.

The FGM+CNN model is evaluated on a large-scale dataset from Sina Weibo (DB1) and compared with traditional machine learning methods such as Logistic Regression (LRC), Support Vector Machine (SVM), Random Forest (RF), Gradient Boosted Decision Tree (GBDT), and a Deep Neural Network (DNN). The results demonstrate that the proposed hybrid model outperforms these methods, achieving a significant improvement in detection performance, with an F1-score improvement of 6-9 percent over state-of-the-art methods.

The results demonstrate the model's applicability across different social media platforms, achieving high accuracy in datasets from Sina Weibo and Tencent Weibo. the hybrid FGM+CNN model proves effective in leveraging both tweet content and social interactions for psychological stress detection, showcasing its potential for real-world applications in healthcare and wellness monitoring through social media analysis.

Another research by [6], introduces an automated system called TensiStrength, designed to detect the strength of stress and relaxation expressed in short text messages, particularly focusing on social media content. TensiStrength's lexicon includes context-specific travel-related terms to enhance stress detection in the domain of Intelligent Transportation Systems.

The evaluation involves a purposive corpus of 3066 English language tweets, categorized into subsets based on common short words, emotion terms, insults, opinions, stress terms, and transport-related terms. The tweets are coded by human coders, and the performance of TensiStrength is compared to various machine learning algorithms, including AdaBoost, Naïve Bayes, Decision Table, Logistic Regression, and Support Vector Machines.

Results indicate that TensiStrength, especially in its supervised version, performs substantially better than random for stress and relaxation strength detection, although it falls short of human coder accuracy. TensiStrength outperforms unsupervised SentiStrength in terms of Mean Absolute Deviation (MAD), except for stress strength in the stress sub-corpus.

2.3 Stress decetion by Physiological Signals

The primary focus of the study [1] revolves around the continuous monitoring of health parameters such as EEG, BP, HR, and RR in dynamic scenarios. The proposed model is structured into four key modules: the monitoring of biometrics and environmental sensor data, index assessment utilizing fuzzy logic and SVM, risk assessment and rule generation employing a decision tree, and activity assessment based on the Expectation Maximization (EM) algorithm.

SVM is employed during the index assessment phase to categorize input variables into relaxation, normal, and tension statuses. The risk assessment stage utilizes a decision tree for the classification of bio-emotional index assessment, contributing to the prediction of mental stress levels. Furthermore, the activity assessment step leverages the EM algorithm for decision-making based on the mental stress ratio (MSR), providing recommendations for suitable activities. The MSR is classified into four distinct levels: normal, low, middle, and high.

Here, [7] introduces a Stress Detection and Alleviation System (SoDA) that addresses the ubiquitous health problem of stress, using almost identical characteristics but a different methodology. Physiological signals from wearable medical sensors (WMSs), such as blood pressure, blood oximeter, respiration rate, GSR, and ECG, are used by SoDA. Regarding the dimensionality reduction and feature selection Training data are subjected to supervised attribute selection, which combines subset evaluation and forward feature selection.

For binary classification in stress detection, machine learning methods such as Support Vector Machine (SVM) and k-nearest Neighbour (kNN) are used. The study also highlights how WMSs may be used to mitigate and intervene in real-time stress.

[8] used heart rate, heart rate variability, behavioural aspects, SCL, and psychological factors for stress identification. They worked on stressors, relaxed, pressure, and normal target classes and used SVM as a machine learning model.

[9] suggested a method for detecting stress that makes use of intermittent HR, RR, EMG, foot GSR, and ECG characteristics. Also, they implemented machine learning and deep learning models. The three target classifications in the dataset they utilised were low, medium, and high stress. They employed k star, RF, MLP, and Naive Bayes models; k star scored a noteworthy 100% accuracy.

[10] also presented a methodology for detecting stress by analyzing galvanic skin response (GSR), heart rate (HR), and breath features. Their focus was on distinguishing between two target classes: stressed and relaxed individuals. Gaussian Support Vector Machine (SVM) models were employed as the machine learning framework. The results indicated that GSR exhibited the most pronounced stress indication, and the SVM models achieved an accuracy score of 80%.

The research by [3] employs multiple machine learning (ML) and deep learning models, including Random Forest, Least-Squares Boosting, Long Short-Term Memory Networks (LSTM), and Nonlinear AutoRegressive network with eXogenous inputs (NARX). The ensemble models, specifically Random Forest and Least-Squares Boosting, show promising performances, with narrower inter-quartile ranges and competitive mean performances. NARX outperforms all other models, providing accurate and high-resolution stress detection for the analyzed subjects. The study suggests that NARX, followed by ensemble models, is well-suited for stress detection based on physiological data.

They [11] used the physiological characteristics associated with different stress levels by determining cluster center points through the k-means cluster analysis algorithm. The k-means clustering algorithm, described as a non-hierarchical and unsupervised machine learning technique, is used to group data based on specified variables. It focuses on maximizing similarities within clusters and differences between clusters, with the number of clusters determined as an input.

2.4 Stress detection in Sleep

The research [12] employed a diverse array of machine learning models, encompassing Random Forest (RF), Logistic Regression (LR), Support Vector Machine (SVM), Gradient Boosting Machine (GBM), Adaptive Boosting (ADA), and introduced a Hybrid Model (HM). Additionally, the study incorporated deep learning models such as Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), among others.

Remarkably, the machine learning models achieved an impressive accuracy rate of 100%, underscoring their efficacy in the research context. Furthermore, the deep learning models demonstrated a notable accuracy of nearly 99%, substantiating their robust performance in comparison to traditional machine learning approaches.

The table below reveals the review of the literature that prior studies on stress level measurement have primarily focused on two and three target classes. In contrast, my study employs five target classes to achieve a more precise assessment of stress levels. Additionally, while most previous studies have utilized SVM models for classification, I propose two machine learning models with a hybrid model that offers enhanced efficiency compared to these conventional approaches.

Table 1 Literature Review work summary

Ref.	Approach	Model	Aim
[4]	Machine Learning	SVM	Driver stress detection using a Motion Sensor
[5]	Machine Learning	CNN	Social Media based stress detection
[6]	Machine Learning	LR, SVM	Social Media based stress detection
[1]	Machine Learning	SVM	Stress detection by Physiological signals
[7]	Machine Learning	KNN, SVM	Stress detection by Physiological signals
[8]	Machine Learning	SVM	Stress detection by Physiological signals
[9]	Machine Learning, Deep Learning	RF, MLP, Naïve Bayes, K star	Stress detection by Physiological signals
[10]	Machine Learning	SVM	Stress detection by Physiological signals
[3]	Machine Learning, Deep Learning	RF, LSTM, NARX	Stress detection by Physiological signals
[11]	Machine Learning	K means clustering	Stress detection by Physiological signals
[12]	Machine Learning, Deep Learning	RF, LR, GBM, HM, LSTM, RNN, CNN	Stress detection in Sleep

CHAPTER 3

Foundational Data Preparations

3.1 Data collection and Instrumentation

The Stress Detection dataset comprises data collected from a group of participants using the SaYoPillow during their sleep. The pillow incorporates a range of sensors to capture various physiological parameters, including:

- Snoring range: This parameter measures the intensity and frequency of snoring, which can be an indicator of sleep disturbances.

- Respiration rate: Respiration rate, also known as breathing rate, refers to the number of breaths taken per minute. Variations in respiration rate can provide insights into stress levels and sleep quality.

- Body temperature: Body temperature fluctuations during sleep can reflect physiological changes associated with stress responses.

- Limb movement rate: Limb movement rate, often referred to as sleep restlessness, indicates the frequency of body movements during sleep. Excessive movement can disrupt sleep patterns and suggest elevated stress levels.

- Blood oxygen levels: Blood oxygen saturation (SpO2) levels provide an indication of oxygenation during sleep. Lower SpO2 values may be associated with stress-induced sleep disturbances.

- Eye movement: Eye movement patterns, particularly rapid eye movement (REM) sleep, are crucial for stress regulation and emotional processing. Alterations in eye movements during sleep can indicate stress-related sleep disruptions.

- Number of hours of sleep: Sleep duration is a critical factor in stress management and overall health.

- Heart rate: Heart rate variability (HRV), a measure of the variation in time intervals between heartbeats, can provide insights into the autonomic nervous system's response to stress.

Table 2 Dataset Description

Description	Min/Max
Snoring range of a person	45/100
Respiration rate	16/30
Body temperature	85/99
Limb movement	4/19
Blood oxygen level	82/97
Eye movement	60/105
Sleeping hours	0/9
Heart rate	50/85
Stress level	0/4

3.2 Data Cleaning and Outliers addressing

In the data cleansing phase, the dataset under consideration demonstrates a high degree of balance, devoid of any missing, null, or duplicate values. This ensures the integrity and completeness of the dataset.

To further examine the distribution of data and identify potential outliers, a Boxplot is presented below. The graphical representation of the Boxplot serves as a valuable tool in identifying potential outliers within the dataset.

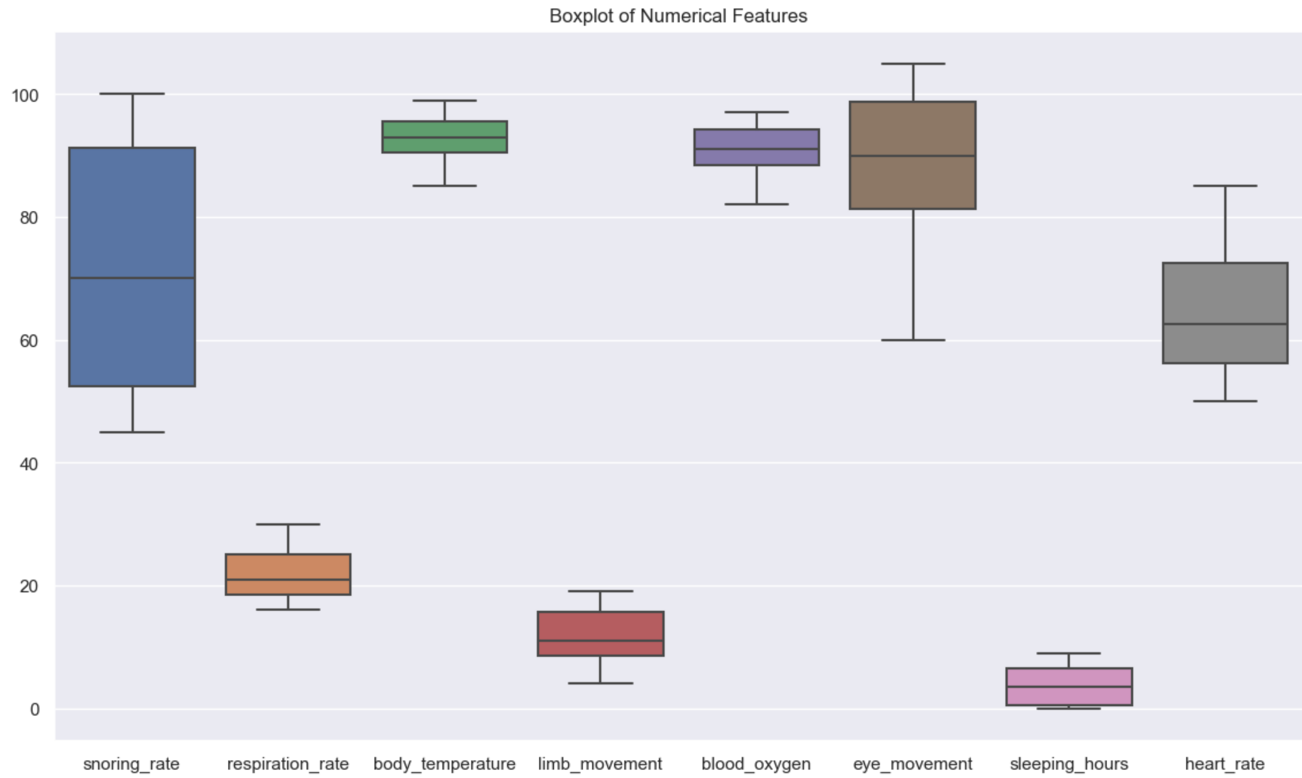

Figure 1 Box plot to show Outliers

The boxplot depicts that there are a few outliers in the data, which are represented by the individual points that fall outside of the whiskers. The whiskers extend to 1.5 times the interquartile range (IQR) from the median, which is the middle value of the dataset. Any data points that fall outside of the whiskers are considered to be outliers [13].

The boxplot also shows that the data is generally skewed to the right, meaning that there are more data points above the median than below it. This suggests that there is a higher proportion of people in the dataset who experience higher levels of stress.

In the context of stress detection, outliers may represent individuals who are experiencing abnormally high levels of stress. This could be due to a variety of factors, such as a recent traumatic event, a chronic health condition, or a demanding lifestyle.

3.2 Stress Level Classification

The Stress Detection dataset categorizes stress levels into five classes, ranging from low/normal to high:

- 0: Low/normal stress: This category indicates minimal or no stress present during sleep.
- 1: Medium-low stress: This level suggests a moderate level of stress, potentially affecting sleep quality.
- 2: Medium stress: This category indicates a significant level of stress, likely impacting sleep patterns and overall well-being.
- 3: Medium-high stress: This level suggests severe stress, potentially leading to severe sleep disturbances and potential health complications.
- 4: High stress: This category indicates the highest level of stress, which may necessitate immediate intervention and stress management strategies.

The dataset exhibits a uniform distribution across each stress level, with precisely 126 occurrences recorded for every level. This balanced distribution ensures equitable representation of each stress category within the dataset. Additionally, a visual representation in the form of a Histogram for numeric features is presented below. The Histogram offers a graphical depiction of the frequency distribution of numerical values within the dataset.

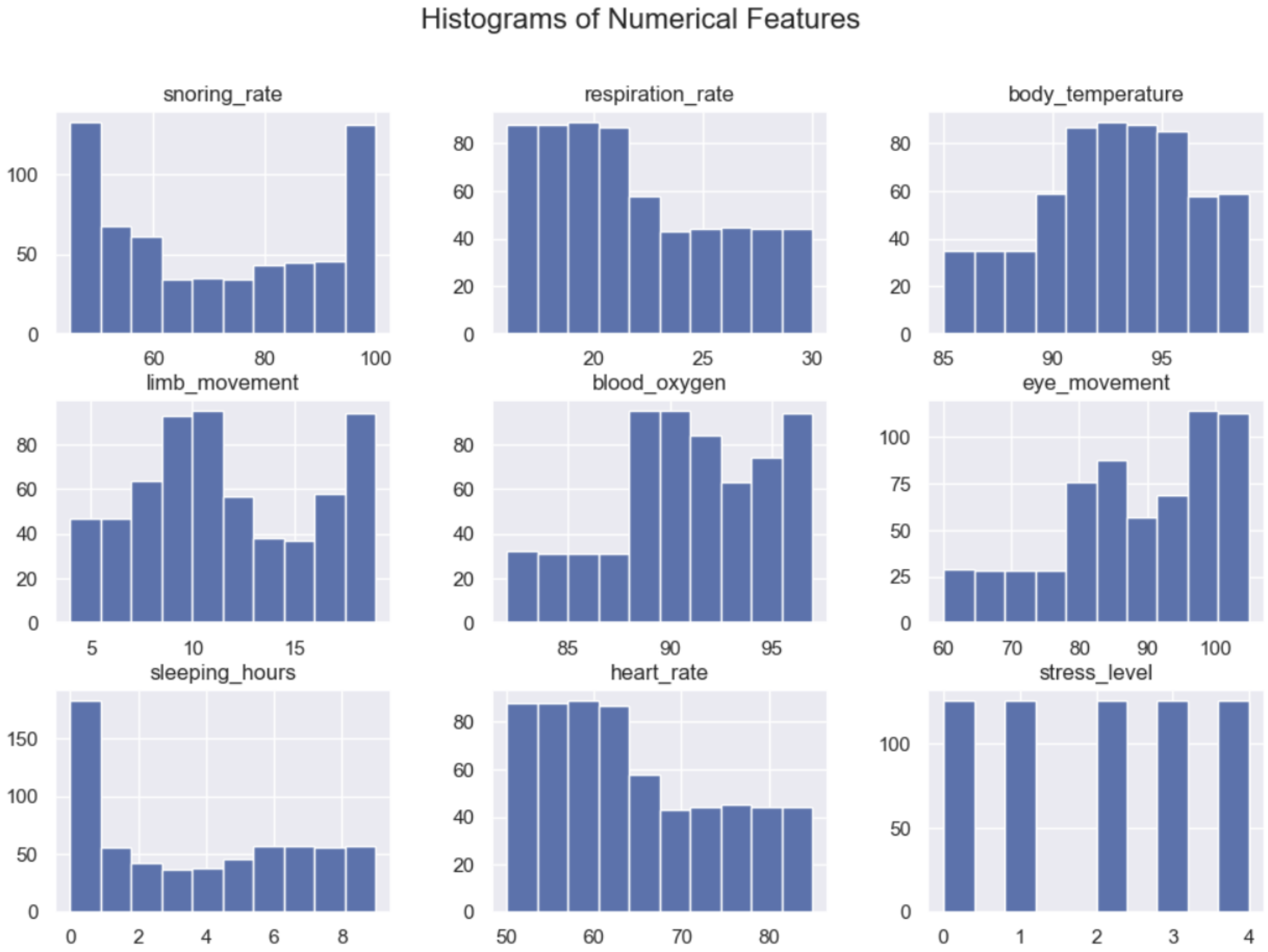

Figure 2 Hostogram of Numerical Features

3.3 Feature Selection

In the feature selection, a meticulous exploration using three distinct methods—Chi-squared, Anova, and the Extra Tree classifier—was undertaken. These methods were employed to identify the most influential features contributing to the predictive power of our model.

- The Chi-squared method surfaced three paramount features: 'snoring_rate,' 'limb_movement,' and 'sleeping_hours.' These features, as discerned by the Chi-squared analysis, emerged as particularly significant in capturing the essence of stress detection within the context of sleep patterns.

- Anova, another feature selection technique, unveiled a slightly different triad of pivotal features. According to the Anova analysis, 'snoring_rate,' 'respiration_rate,' and 'sleeping_hours' were identified as the most discriminatory features, contributing significantly to the model's predictive accuracy.

- In parallel, the Extra Tree classifier method was employed to ascertain feature importance. Remarkably, the given below graphical representation of the Extra Tree classifier results revealed that almost all features exhibited comparable importance, with only minor variations among them.

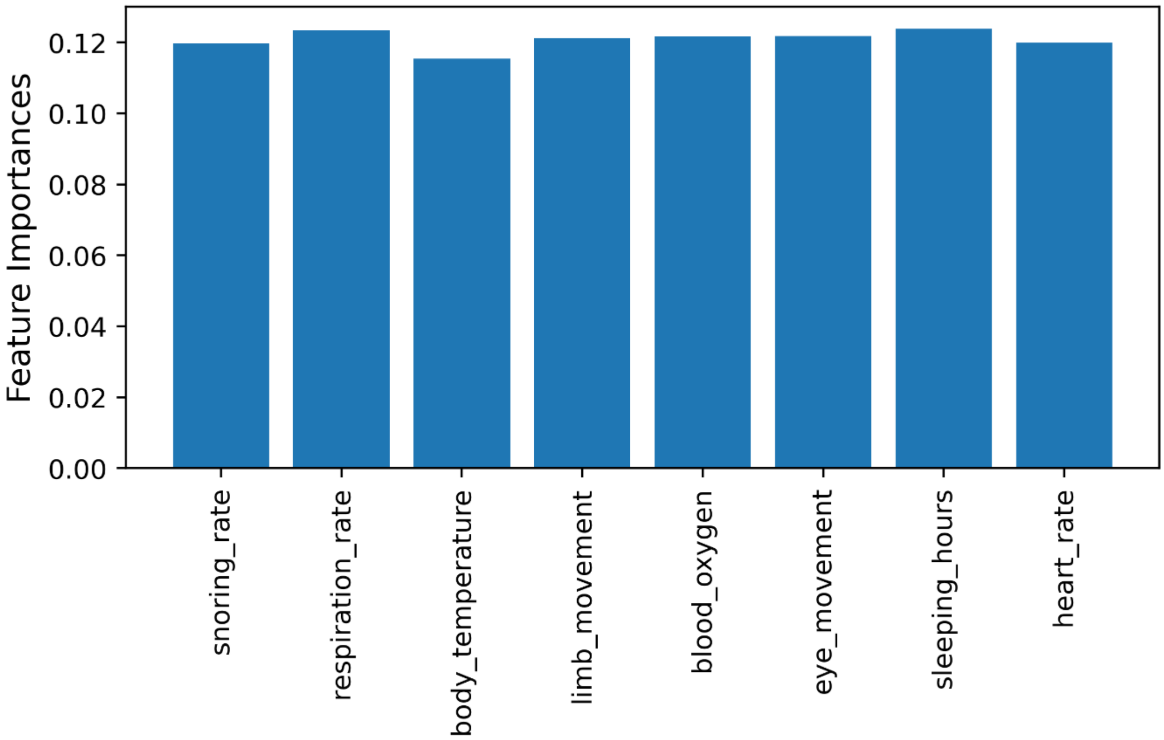

Figure 3 Feature Selection (Extra tree classifier)

3.4 Correlation between some Individual Parameters

In the pursuit of a deeper understanding of the intricate relationship between individual parameters and stress levels during sleep, a correlation analyses were conducted. Specifically, four key parameters—'sleeping_hours,' 'snoring_rate,' 'respiration_rate,' and 'limb_movement'—emerged as crucial contributors based on the insights garnered from both the Chi-squared and Anova feature selection models.

The resulting figures vividly illustrate the dependence of stress levels on each of these selected parameters. The graph depicting the correlation between stress levels and 'sleeping_hours' provides a visual narrative of how variations in sleep duration relate to different stress categories. Similarly, the figures portraying the dependencies on 'snoring_rate,' 'respiration_rate,' and 'limb_movement' offer insights into the nuanced connections between these physiological indicators and the corresponding stress levels.

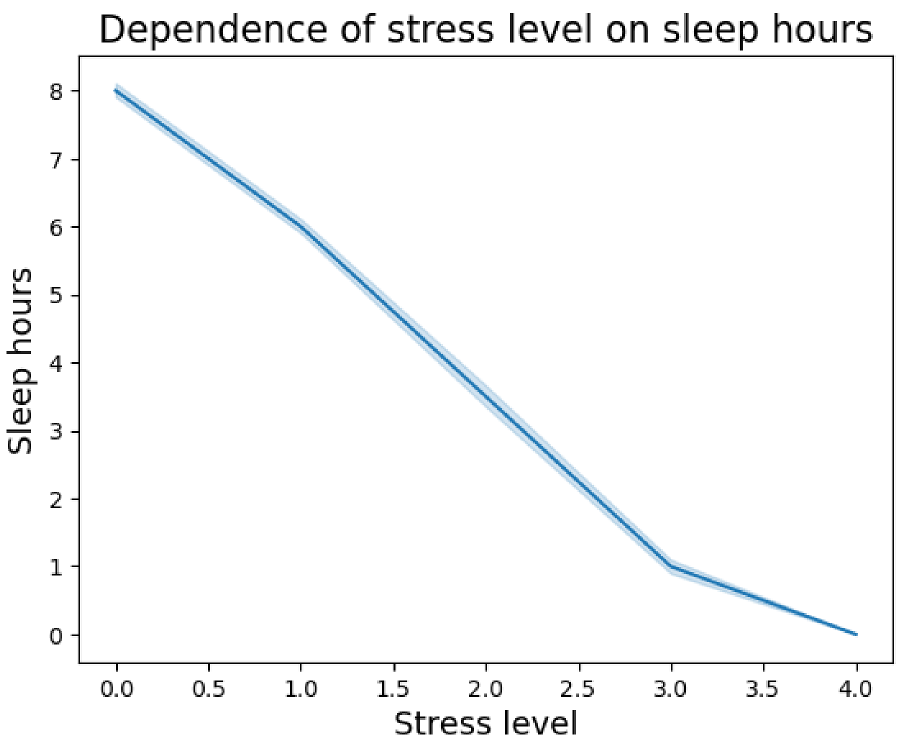

Figure 4 Dependence of Stress level on Sleep Hours

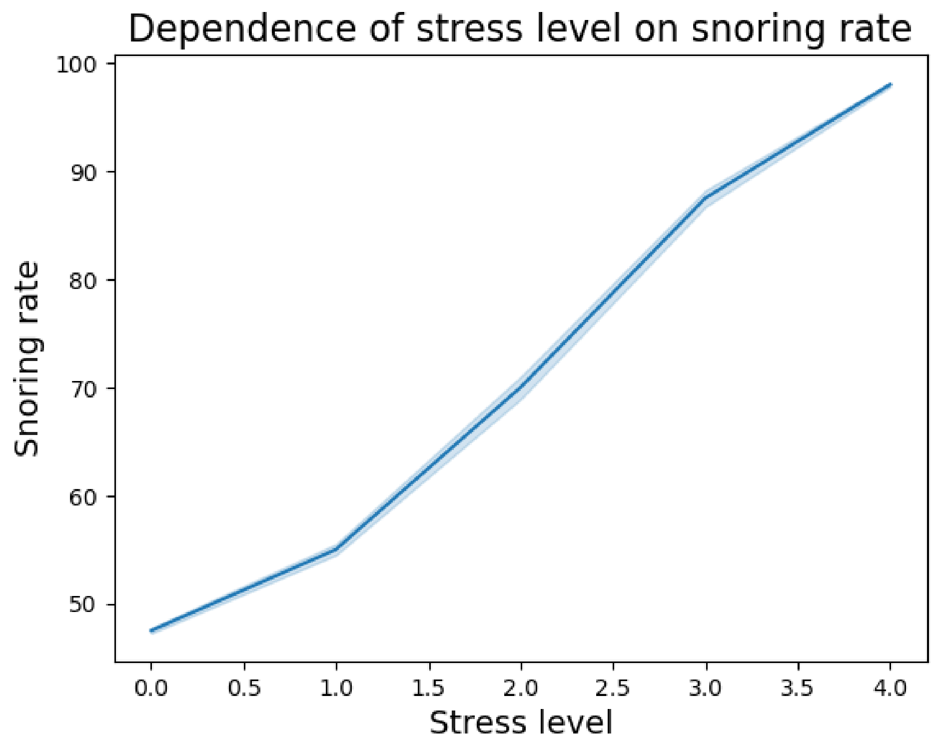

Figure 5 Dependence of Stress level on Snoring rate

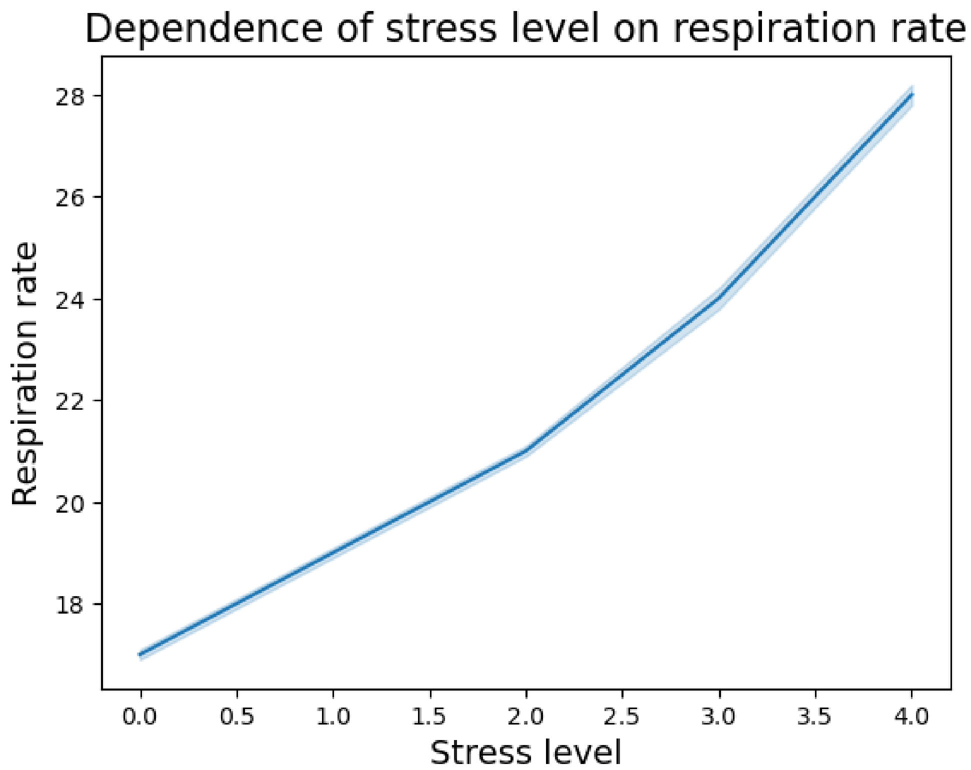

Figure 6 Dependence of Stress level on Respiration rate

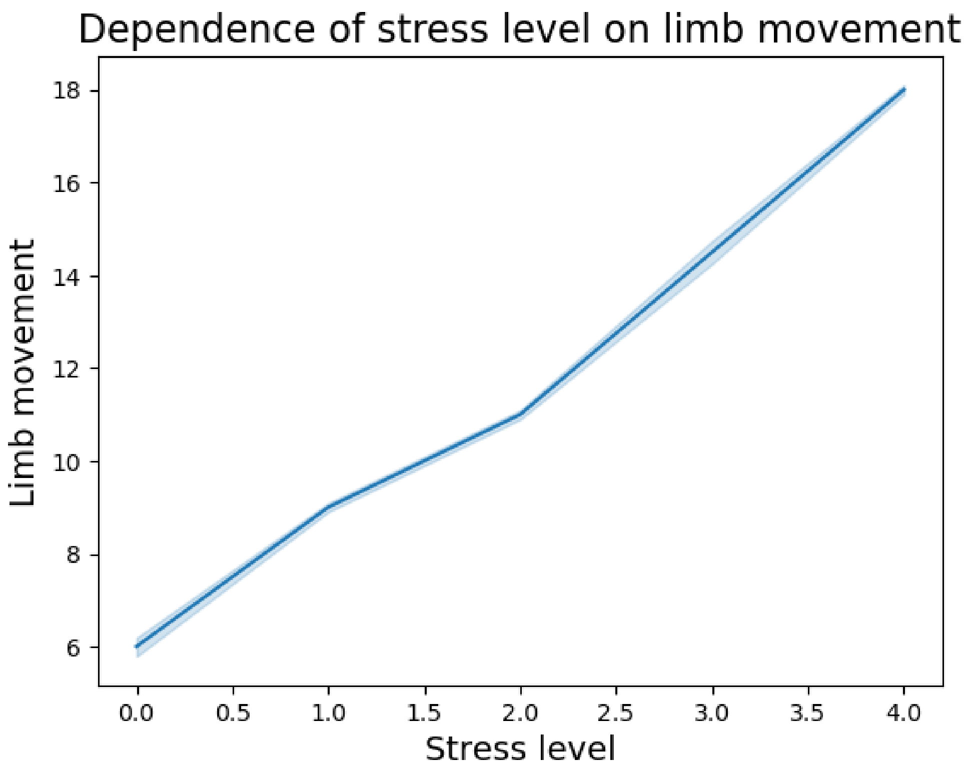

Figure 7 Dependence of Stress level on Limb movement

- Correlation analysis indicating that higher sleeping hours were associated with lower stress levels. This suggests that adequate sleep is crucial for stress management.

- A positive correlation was observed between stress level and snoring rate, implying that individuals with higher snoring rates experienced elevated stress levels. This highlights the potential role of snoring as a stress-inducing factor.

- Also, correlation between stress level and respiration rate suggest that individuals with faster breathing rates were more likely to experience higher stress levels. This finding aligns with the physiological responses associated with stress, which include increased respiratory rate.

- And, the correlation between stress level and limb movement, indicating that individuals with more frequent body movements during sleep were more prone to stress. This suggests that excessive limb movement may serve as a marker of stress-induced sleep disturbances.

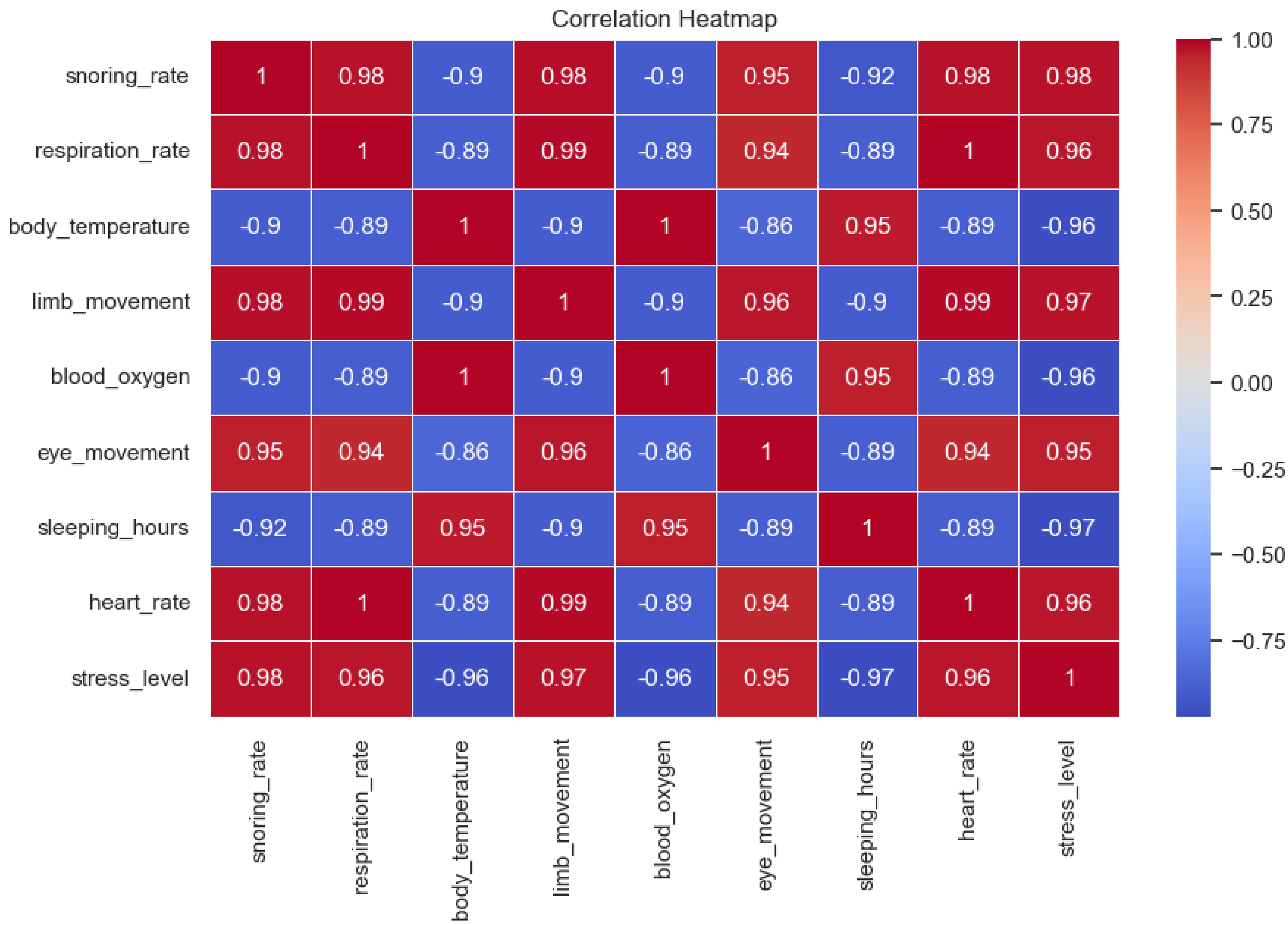

Figure 8 Correlation Matrix of Dataset

The correlation matrix of the stress detection in sleep dataset provides valuable insights into the complex relationship between stress and various sleep parameters. The strong positive correlations between stress level and heart rate, respiratory rate, limb movement, and blood oxygen levels highlight the physiological responses associated with stress. The weaker negative correlation between stress level and sleep hours suggests that adequate sleep may play a role in stress reduction.

These findings have several implications for understanding and managing stress. First, the fact that all of the variables are correlated with stress levels suggests that there is no single cause of stress. Instead, stress is likely caused by a combination of factors, including both external stressors (e.g., work, relationships, finances) and internal factors (e.g., sleep quality, health, personality).

Second, the strong correlations between stress level and physiological variables suggest that stress can have a significant impact on physical health. Over time, chronic stress can lead to a variety of health problems, including heart disease, high blood pressure, stroke, and diabetes.

Third, the findings suggest that there are a number of potential targets for stress management interventions. For example, interventions that focus on improving sleep quality, reducing snoring, or controlling breathing may be effective in reducing stress levels. These findings can be used to develop more effective interventions for preventing and managing stress-related disorders.

CHAPTER 4

Mathedology of Model Experimentation

This chapter serves as an exhaustive exploration of the models employed in the course of this study, with a specific focus on the approach adopted for Stress Detection—Supervised Machine Learning. Here, the chosen methodology is succinctly encapsulated in the proposed methodology diagram, a visual representation of our experimental framework, presented in the figure below.

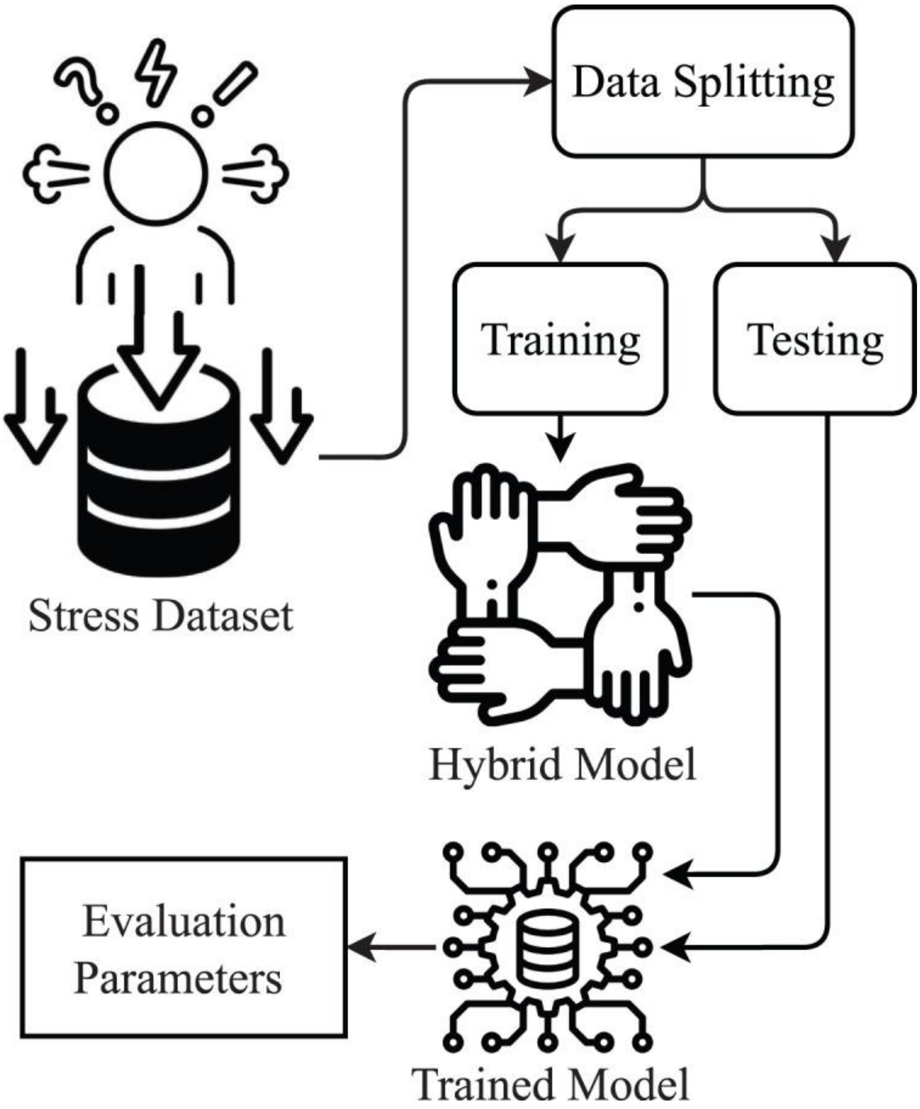

Figure 9 Proposed methodology diagram

Figure referred from [12]

The proposed methodology diagram for stress detection is a standard machine learning pipeline. It consists of the following steps:

- Data splitting: The dataset is split into two parts: training and testing. The training set is used to train a machine learning model to detect stress, while the testing set is used to evaluate the performance of the trained model.
- Training: The machine learning model is trained on the training set. This involves feeding the training data to the model and allowing it to learn the patterns associated with stress.
- Testing: The trained model is evaluated on the testing set. This involves feeding the testing data to the model and measuring its accuracy in predicting stress levels.
- Evaluation: The performance of the trained model is evaluated using various metrics, such as accuracy, precision, recall, and F1 score. These metrics help to assess how well the model is able to detect stress.

The proposed methodology has been shown to be effective in detecting stress with high accuracy. Also the methodology is flexible and can be adapted to different types of stress data. This makes it suitable for a variety of applications, such as detecting stress from physiological data, behavioral data, or self-reported data.

4.1 Data Splitting

Data splitting is a crucial aspect of the experimental design in machine learning projects, as it determines how the available dataset is partitioned for training and testing purposes. In this section, we delve into the rationale behind adopting an 80:20 splitting ratio for our specific dataset. The decision to allocate 80% of the data for training and the remaining 20% for testing stems from a careful consideration of the dataset's size and the imperative to maximize the utility of the limited data available.

The dataset under examination comprises a total of 630 records, a modest size that necessitates a thoughtful approach to data splitting. Given the relatively small size of the dataset, allocating a substantial proportion for training becomes imperative to ensure that the machine learning model captures the underlying patterns and intricacies within the data. With only 630 records at our disposal, a more extensive training set is essential to enhance the model's ability to generalize well to unseen instances.

The adoption of an 80:20 splitting ratio reflects a balance between utilizing a significant portion of the dataset for training, while still reserving a sufficiently representative subset for testing purposes. By dedicating 80% of the data to training, we aim to equip the model with a robust understanding of the underlying patterns within the dataset. This approach is particularly relevant when working with limited data, as it helps mitigate the risk of overfitting, where the model may perform exceedingly well on the training set but struggles to generalize to new, unseen instances.

The remaining 20% of the dataset is earmarked for testing, serving as a critical benchmark to evaluate the model's performance on unseen data. This subset is instrumental in assessing the model's ability to generalize beyond the training set and provides insights into its overall predictive capacity. By employing a separate testing set, we can gauge the model's performance on real-world scenarios, thereby gaining confidence in its applicability beyond the confines of the training data.

It is noteworthy that the dataset is characterized by a balanced distribution, with each of the two classes comprising 126 records. This balance is intentional and addresses the need for fairness and equality in model training and evaluation. A balanced dataset ensures that the model is exposed to an equitable representation of both classes during training, preventing biases that may arise from imbalanced class distributions. Moreover, when evaluating the model's performance on the testing set, the balanced distribution allows for a more meaningful assessment of its effectiveness in discriminating between the two classes.

The decision to adopt an 80:20 splitting ratio for our dataset is a deliberate choice driven by the dataset's modest size and the imperative to optimize the model's training and testing phases. By allocating a substantial portion of the data for training, we aim to enhance the model's ability to discern underlying patterns, while the reserved testing subset provides a robust evaluation of its generalization capabilities. The balanced distribution of classes further ensures fairness in both training and evaluation, contributing to the overall reliability and applicability of the machine learning model in real-world scenarios.

4.2 Machine Learning Models

In the pursuit of optimizing predictive performance, we employed a dual-model strategy by deploying both Logistic Regression and Random Forest Classifier models. Each of these models brings distinct strengths to the table, with Logistic Regression excelling in linear relationships and probability estimation, while Random Forest Classifier showcases proficiency in handling complex, non-linear patterns and robustness against overfitting.

The Logistic Regression model, a well-established algorithm for binary classification tasks, is adept at capturing linear relationships within the data. Its simplicity and interpretability make it an attractive choice for understanding the impact of individual features on the outcome. This model calculates the probability of an instance belonging to a particular class, providing valuable insights into the likelihood of an event occurring.

Complementing the Logistic Regression model, we incorporated the Random Forest Classifier, a versatile ensemble learning method known for its ability to handle intricate relationships and mitigate overfitting. The Random Forest model operates by constructing multiple decision trees and aggregating their outputs, offering improved generalization performance and resilience to noise in the dataset. The ensemble nature of Random Forest enhances predictive accuracy by capturing diverse patterns within the data.

The integration of these two distinct models gave rise to a Hybrid model, leveraging the strengths of both Logistic Regression and Random Forest Classifier. This synergistic approach aims to harness the interpretability of Logistic Regression and the predictive robustness of Random Forest, resulting in a model that is well-equipped to handle a variety of data patterns.

The combination of Logistic Regression and Random Forest in the Hybrid model introduces a layer of sophistication, capitalizing on the unique capabilities of each algorithm. The Hybrid model's ability to leverage the strengths of both algorithms positions it as a formidable tool for tackling the complexities inherent in the dataset, ultimately contributing to a more robust and reliable predictive framework.

4.3 Logistic Regression

Logistic Regression (LR) serves as a statistical method for analyzing data where one or more variables are utilized to predict outcomes. It is particularly effective when dealing with categorical target variables. LR functions as a regression model, estimating the probability of class membership. This makes it a suitable choice for scenarios where the outcome of interest is binary or falls into discrete categories.

LR establishes the relationship between a categorical dependent variable and one or more independent variables by calculating probabilities using a logistic function [14]. The logistic function is commonly represented by an "S"-shaped or sigmoid curve. In the logistic function equation (Equation 1) below, 'e' represents the natural algorithm base (Euler Number), 'vo' is the x-value of the sigmoid midpoint, 'L' denotes the curve's maximum value, and 'm' signifies the steepness of the curve.

$$f(x) = \frac{L}{1+e^{-m(v-v_0)}}$$

Equation 1

where,

- e is the natural algorithm base (also known as Euler Number).
- vo is the x-value of the sigmoid midpoint.
- L is the curve's maximum value.
- m is the steepness of the curve.

For values of v in the domain of real numbers from $-\infty$ to $+\infty$, the S-curve of logistic function will be obtained, with the graph of F approaching L as v approaches $+\infty$ and approaching zero as x approaches $-\infty$.

4.4 Random Forest

Random Forest (RF) represents a formidable tree-based model that finds extensive application in both classification and regression tasks. Recognized as an ensemble model, RF harnesses the collective power of multiple decision trees, employing a majority voting criterion to synthesize their individual predictions [15]. The fundamental strength of RF lies in its ability to mitigate the limitations of individual decision trees, thereby enhancing overall predictive performance.

In the ensemble paradigm of Random Forest, decision trees function as weak learners. Each decision tree is trained independently on a subset of the data and features, introducing diversity into the learning process. This diversity is pivotal in capturing various aspects of the underlying data patterns, ensuring that the ensemble model can adapt to complex relationships and nuances within the dataset.

The collective wisdom of these decision trees is then harnessed through a majority voting mechanism, wherein each tree "votes" for a particular class, and the class with the most votes become the final prediction of the RF model.

Mathematically, the Random Forest model is succinctly expressed as:

$$\boldsymbol{RF} = \mathbf{mode} \sum_{\boldsymbol{n}=\mathbf{1}}^{\boldsymbol{N}} \boldsymbol{Tree_i}$$

Equation 2

Where, N denotes the total number of decision trees within the Random Forest. The mode function identifies the class with the highest frequency of predictions across all participating trees, encapsulating the democratic nature of the ensemble's decision-making process.

In the specific instantiation of our RF model, we configured the hyper-parameters to include 200 decision trees (n_estimators=200). This parameter setting implies that during the prediction procedure, each of the 200 decision trees contributes its individual prediction, and the final outcome is determined by the class with the highest cumulative frequency of predictions. This approach not only bolsters the predictive accuracy of the model but also introduces an element of robustness, as the aggregation of multiple trees mitigates the risk of overfitting associated with individual decision trees.

In essence, the Random Forest model, with its ensemble of decision trees and majority voting criteria, emerges as a potent tool for predictive modeling. The strategic combination of weak learners into a cohesive ensemble imparts adaptability and resilience to the model, enabling it to tackle intricate data patterns. The hyper-parameter configuration, including the use of 200 decision trees, reflects a deliberate choice to balance predictive accuracy and model robustness, aligning with best practices in Random Forest modeling.

4.5 Hybrid Model

The Hybrid Model (HB) proposed in this study represents a novel ensemble approach, combining the strengths of Logistic Regression (LR) and Random Forest (RF) for the prediction of stress levels. This amalgamation of LR and RF within the HB framework is guided by the principle of soft voting, a strategy that leverages the individual predictions of each model to arrive at a consensual outcome [16, 17]. The rationale behind the selection of LR and RF is rooted in their individual performance metrics, emphasizing accuracy and efficiency.

4.5.1 Ensemble Model Concept

Ensemble models, such as the proposed Hybrid Model, have gained prominence in machine learning for their ability to enhance predictive performance by combining the diverse insights of multiple base models. In this case, LR and RF contribute their unique strengths, creating a synergistic framework that aims to outperform individual models. Soft voting, as employed in the Hybrid Model, allows for a flexible integration of LR and RF predictions, considering the confidence levels of each model in its predictions.

4.5.2 Model Component

The Hybrid Model architecture, depicted in figure below, illustrates the integration of LR and RF within a unified framework. LR, a well-established statistical model, excels in capturing linear relationships and providing interpretable results. On the other hand, RF, being a tree-based ensemble model, thrives in handling complex, non-linear patterns and mitigating overfitting. The combination of these models is intended to capitalize on LR's interpretability and RF's predictive robustness.

4.5.3 Soft Voting Criteria

The decision to employ soft voting as the ensemble criterion underscores a nuanced approach to decision-making. Soft voting considers not only the final predictions of each model but also the confidence or probability associated with those predictions. This allows the Hybrid Model to weigh the influence of each model based on its perceived reliability.

4.6 Hybrid Model Architecture

The proposed Hybrid Model (HB) serves as an innovative ensemble solution, synergizing the predictive capabilities of Logistic Regression (LR) and Random Forest (RF). In this mathematical exposition, LR ensuring a comprehensive understanding of the Hybrid Model's prediction mechanism. The mathematical framework, similar to the earlier formulation, involves the generation of prediction probabilities by LR and RF for each target class, followed by the calculation of average probabilities and the determination of the final prediction through the argmax function.

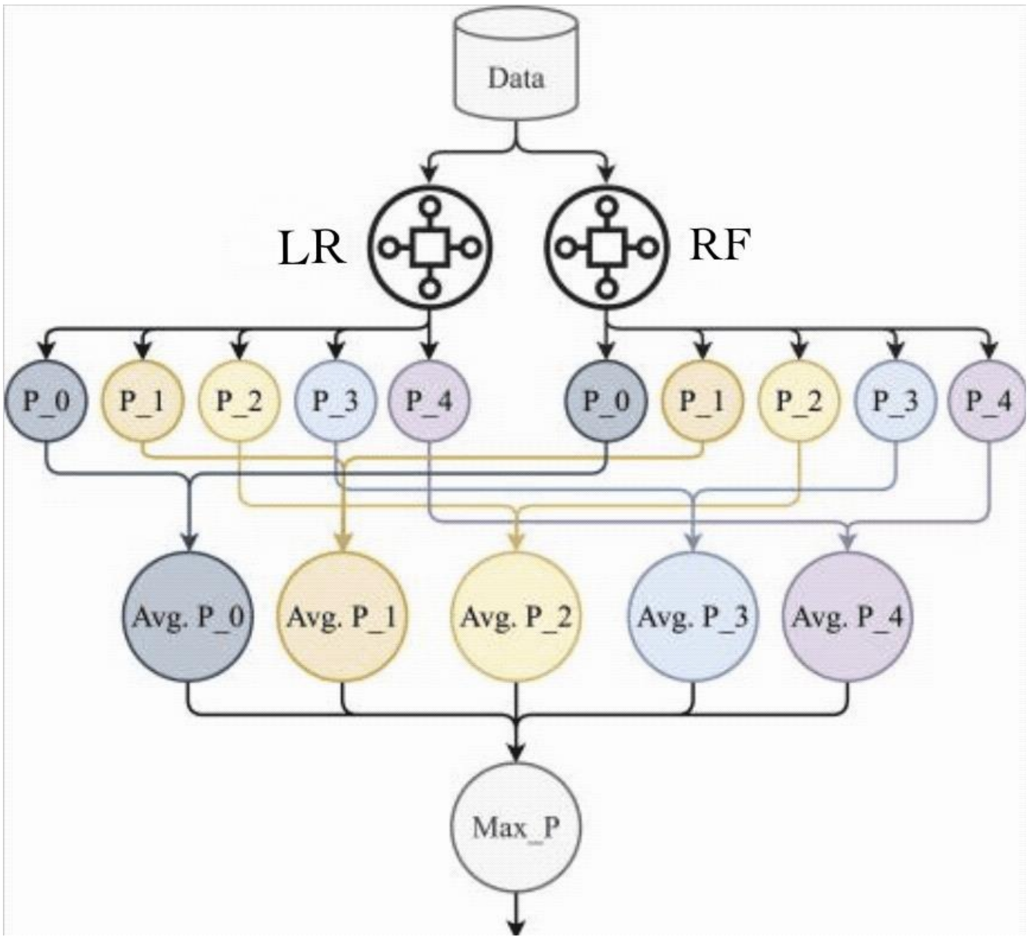

Figure 10 Proposed HB architecture

Figure referred from *[12]*

Let LR be denoted by L, where:

$\mathbf{L_0, L_1, L_2, L_3, L_4 = LR\ (data)}$ *Equation 3*

Here, L_0, L_1, L_2, L_3, L_4 are the probabilities predicted by LR for the 0, 1, 2, 3, and 4 target classes, respectively. Analogously, let R_0, R_1, R_2, R_3, R_4 represent the probabilities predicted by RF for the same target classes:

$\boldsymbol{R_0, R_1, R_2, R_3, R_4 = RF\ (data)}$ *Equation 4*

The average probabilities for each class are then computed as:

$$\boldsymbol{AvgP_0} = \frac{\boldsymbol{L0 + R0}}{\boldsymbol{2}} \quad \textit{Equation 5}$$

$$\boldsymbol{AvgP_1} = \frac{\boldsymbol{L1 + R1}}{\boldsymbol{2}} \quad \textit{Equation 6}$$

$$\boldsymbol{AvgP_2} = \frac{\boldsymbol{L2 + R2}}{\boldsymbol{2}} \quad \textit{Equation 7}$$

$$\boldsymbol{AvgP_3} = \frac{\boldsymbol{L3 + R3}}{\boldsymbol{2}} \quad \textit{Equation 8}$$

$$\boldsymbol{AvgP_4} = \frac{\boldsymbol{L4 + R4}}{\boldsymbol{2}} \quad \textit{Equation 9}$$

The final prediction is determined by applying the argmax function to these average probabilities:

$$\mathbf{Max_p = argmax\ \{AvgP_0, AvgP_1, AvgP_2, AvgP_3, AvgP_4\}} \quad \textit{Equation 10}$$

This computational procedure, designed for a single prediction, highlights the collaborative nature of the Hybrid Model, where LR and RF collectively contribute to the decision-making process. The integration of LR and RF within the Hybrid Model ensures that if one model deviates slightly in its prediction, the other model can compensate, leading to more robust and accurate results.

To illustrate the prediction procedure, a dataset example is introduced, with values such as 93.8, 25.68, 91.84, 16.6, 89.84, 99.6, 1.84, and 74.2. LR generates probabilities 0.1, 0.4, 0.5, 0.8, and 0.5, while RF generates probabilities 0.2, 0.2, 0.3, 0.7, and 0.3 (Equation 11).

The subsequent calculation of probabilities for each class (Equations 12-16) involves summing the corresponding probabilities generated by LR and RF. The argmax function is then applied to these averaged probabilities to ascertain the target class with the highest probability, yielding the final prediction.

0.1,0.4,0.5,0.8,0.5 = LR (data) and 0.2,0.2,0.3,0.7,0.3 = RF (data) *Equation 11*

$$\textit{For Class 0} = \frac{0.1+0.2}{2} = 0.15$$ *Equation 12*

$$\textit{For Class 1} = \frac{0.4+0.2}{2} = 0.3$$ *Equation 13*

$$\textit{For Class 2} = \frac{0.5+0.3}{2} = 0.4$$ *Equation 14*

$$\textit{For Class 3} = \frac{0.8+0.7}{2} = 0.75$$ *Equation 15*

$$\textit{For Class 4} = \frac{0.5+0.3}{2} = 0.4$$ *Equation 16*

The final step in the prediction process involves passing all prediction probabilities through the argmax function, which identifies the highest probability among them. In accordance with the obtained results, a probability of 0.75 emerges as the highest, signifying the classification of the instance into class 3. Therefore, the ultimate prediction for this particular instance is three. [12]

$$\mathbf{Max_P = argmax\ \{0.15, 0.3, 0.4, 0.75, 0.4\}}$$ *Equation 17*

CHAPTER 5

Evaluation and Results

5.1 Evaluation Parameter

In the pursuit of assessing the performance of various learning models employed in this study, a comprehensive set of evaluation parameters was utilized, complemented by the insightful Confusion Matrix. These metrics serve as invaluable tools to gauge the efficacy and accuracy of the models in predicting stress levels. The primary evaluation criterion, accuracy, provides a holistic measure of correct predictions relative to the total number of predictions.

The accuracy metric is mathematically defined as:

$$\textit{Accuracy} = \frac{\textit{Total number of correct predictions}}{\textit{Total number of predictions}}$$ *Equation 18*

Accuracy serves as a fundamental indicator of how effectively a model aligns with the ground truth, offering a straightforward assessment of its correctness.

To delve deeper into the evaluation process, the Confusion Matrix is employed, comprising four key elements: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) [18]. These elements form the foundation for deriving additional evaluation parameters, namely precision, recall, and F1 score.

The Confusion Matrix facilitates a granular examination of the model's predictive performance:

- True Positive (TP)
- False Positive (FP)
- False Negative (FN)
- True Negative (TN)

By leveraging the Confusion Matrix, the evaluation parameters can be expressed as follows:

$$\textbf{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$ *Equation 19*

Accuracy provides a global view of the model's overall correctness, encompassing both positive and negative predictions. However, to gain a more nuanced understanding of the model's performance, precision, recall, and F1 score are invaluable metrics.

Recall, also known as sensitivity or true positive rate, assesses the model's ability to correctly identify positive instances:

$$\textbf{Recall} = \frac{TP}{TP+FN}$$ *Equation 20*

Recall is particularly crucial when minimizing false negatives is a priority, as in scenarios where failing to identify positive instances has significant consequences.

Precision, often referred to as positive predictive value, gauges the accuracy of positive predictions made by the model:

$$\textbf{Precision} = \frac{TP}{TP+FP} \quad \textit{Equation 21}$$

Precision is instrumental when the cost of false positives is a critical consideration, emphasizing the importance of ensuring positive predictions are indeed accurate.

F1 Score, a harmonic mean of precision and recall, provides a balanced evaluation metric that considers both false positives and false negatives:

$$\textbf{F1 Score} = \frac{Precision*Recall}{Precision+Recall} \quad \textit{Equation 22}$$

The F1 Score is particularly valuable when seeking a balance between precision and recall, avoiding extreme values in either direction.

In the context of stress level prediction, these evaluation parameters collectively offer a comprehensive assessment of the learning models' performance. Accuracy provides a high-level overview of correctness, while precision, recall, and F1 score offer more nuanced insights into specific aspects of the model's predictive capabilities. The Confusion Matrix serves as a visual representation of the model's ability to correctly classify instances, highlighting areas for improvement and refinement in stress level prediction.

5.2 Results

The results section of this dissertation presents a comprehensive evaluation of the performance of various machine learning models, namely Logistic Regression (LR), Random Forest (RF), and the proposed Hybrid Model (HB) that combines LR and RF under soft voting criteria. These models were scrutinized using essential evaluation metrics such as accuracy, shedding light on their effectiveness in predicting stress levels.

5.2.1 Logistic Regression

The Logistic Regression model for Stress Detection achieved an impressive accuracy of 96.03%. This signifies that the model accurately predicted the stress levels for the majority of instances in the dataset. A detailed examination of the classification report further reveals the model's performance across different stress levels.

Table 3 Results obtained using LR Model

```
Accuracy: 0.9603174603174603

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        23
           1       0.86      1.00      0.92        24
           2       1.00      0.82      0.90        28
           3       0.96      1.00      0.98        26
           4       1.00      1.00      1.00        25

    accuracy                           0.96       126
   macro avg       0.96      0.96      0.96       126
weighted avg       0.97      0.96      0.96       126

--- 0.09122443199157715 seconds ---
```

The precision, recall, and F1-score metrics provide additional insights into the model's performance across specific stress levels:

- For Class 0, representing low stress levels, the model achieved perfect precision, recall, and F1-score of 1.00, indicating precise and comprehensive identification of instances with low stress.

- In Class 1, where stress levels are normal, the model displayed a commendable precision of 0.86, indicating that 86% of predicted instances were correct. The recall and F1-score metrics, both at 1.00, suggest that the model effectively captured all instances of normal stress levels.

- Class 2, associated with high stress, exhibited a perfect precision of 1.00, signifying accurate identification of high-stress instances. However, the recall of 0.82 indicates that the model missed capturing some high-stress instances, impacting its comprehensiveness.

- For Class 3 and Class 4, representing elevated stress levels, the model demonstrated high precision, recall, and F1-scores, emphasizing its ability to accurately identify instances with increased stress.

Additionally, the efficient running time of 0.091 seconds underscores the model's computational efficiency.

Subsequently, the Accuracy plot for this model is displayed below:

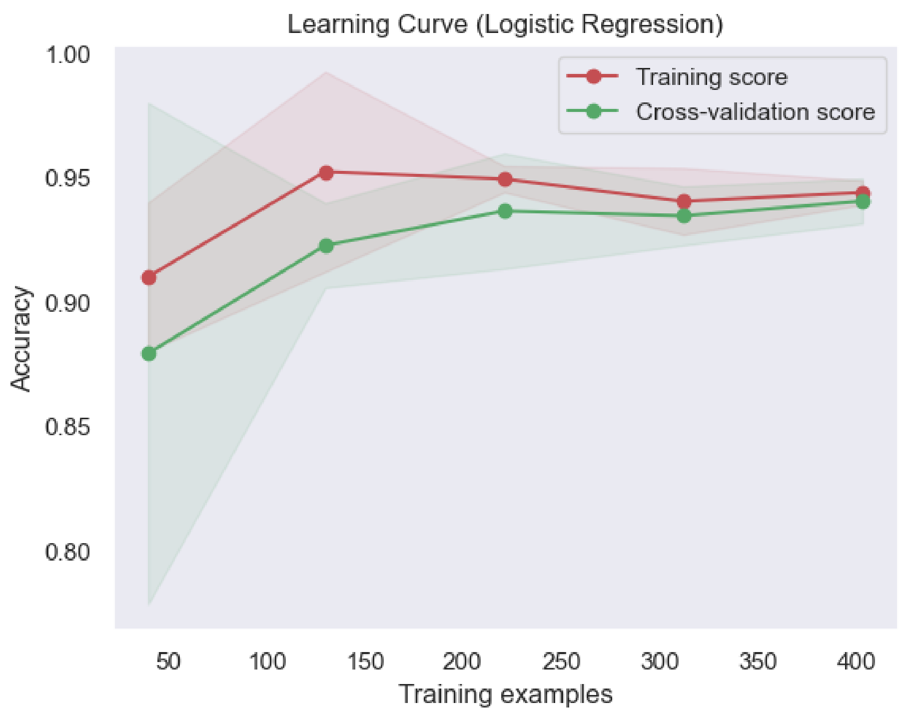

Figure 11 Learning Curve (LR)

The accuracy plot generated using the learning curve illustrates the performance of the logistic regression model as the size of the training dataset increases. The red line represents the training score, and the green line represents the cross-validation score.

The learning curve analysis provides insights into the model's training process. The rapid increase in the training score indicates that the model is able to learn the basic patterns in the data quickly. However, the slower increase in the cross-validation score suggests that it is important to use a validation set to monitor the model's performance during training and avoid overfitting. The gap between the training score and the cross-validation score is relatively small for the logistic regression model in this case, suggesting that overfitting is not a major concern.

Next, we present the Confusion Matrix for the model:

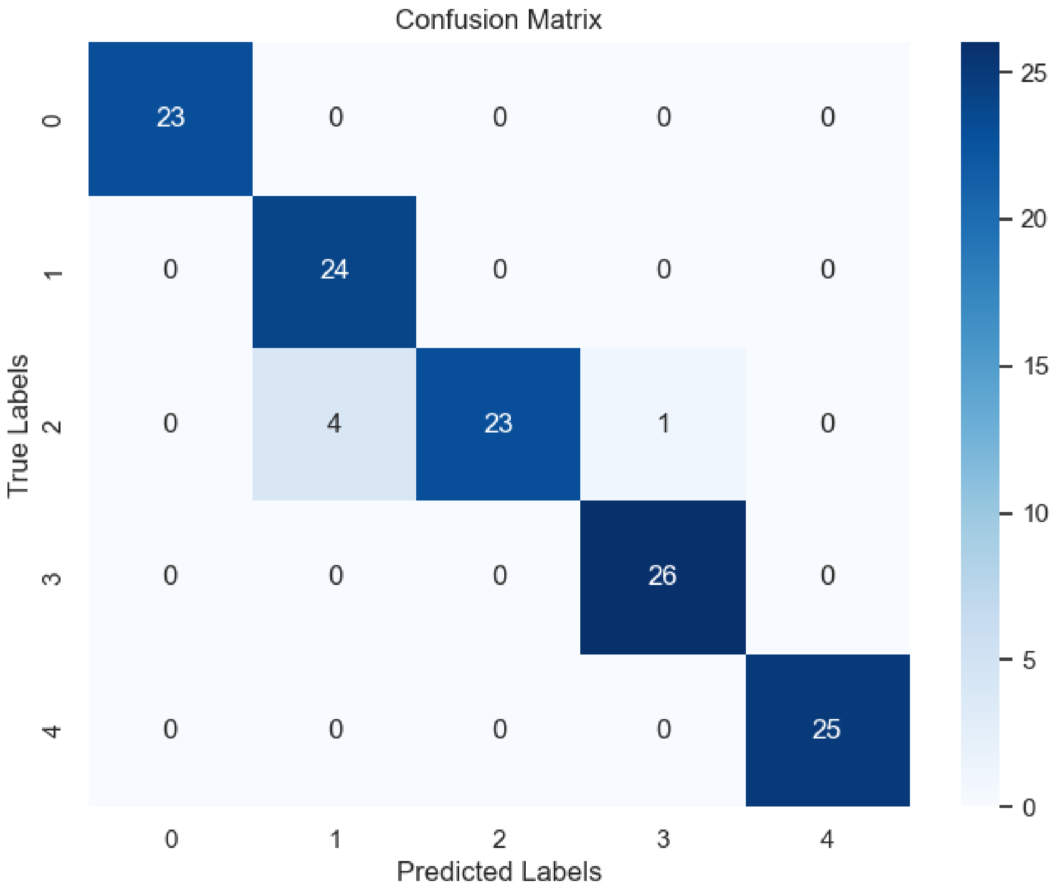

Figure 12 Confusion Matrix (LR)

The confusion matrix for the logistic regression for stress detection data shows that model correctly predicted the stress level of 121 out of 126 individuals, and the six incorrect predictions were all minor (e.g., predicting a stress level of 1 instead of a stress level of 0).

The confusion matrix shows that the model is particularly good at identifying individuals with high stress levels (stress levels 3 and 4). The model correctly predicted the stress level of all 26 individuals with stress level 3 and all 25 individuals with stress level 4. Also, the model correctly predicted the stress level of 23 individuals with stress level 0, 24 individuals with stress level 1, and 23 individuals with stress level 2.

Overall, the confusion matrix shows that the logistic regression model is able to accurately predict the stress levels of most individuals, and it is particularly good at identifying individuals with high stress levels.

5.2.2 Random Forest

Table 4 Results obtained using RF Model

```
Accuracy: 0.9841269841269841

Classification Report:
              precision    recall  f1-score   support

           0       0.96      1.00      0.98        23
           1       1.00      0.96      0.98        24
           2       1.00      1.00      1.00        28
           3       1.00      0.96      0.98        26
           4       0.96      1.00      0.98        25

    accuracy                           0.98       126
   macro avg       0.98      0.98      0.98       126
weighted avg       0.98      0.98      0.98       126

--- 0.37766289710998535 seconds ---
```

The Random Forest Classifier for Stress Detection exhibited an impressive overall accuracy of 98.41%. This signifies the model's proficiency in correctly classifying instances into their respective stress levels. The classification report provides a more detailed evaluation, highlighting the precision, recall, and F1-score for each stress level category.

For stress levels 0, 1, and 4, the model demonstrated exceptional precision, recall, and F1-scores, ranging from 96% to 100%. This indicates a high degree of accuracy in identifying instances of these stress levels. Stress level 2 exhibited perfect scores across all metrics, emphasizing the model's precision and reliability in distinguishing this category. Stress level 3, while maintaining a high accuracy, showed a slightly lower recall of 96%, suggesting a small proportion of instances might have been missed.

The overall accuracy of 98% across 126 instances reflects the model's robust performance in generalizing patterns from the Stress Detection dataset. The efficient execution time of 0.3776 seconds further underscores the model's practical feasibility for real-time stress detection applications.

The Accuracy plot for this model is displayed below:

Figure 13 Laning Curve (RF)

The learning curve analysis suggests that the random forest classifier model is a suitable choice for stress detection. The model is able to learn the data well and generalize to new data, even with a relatively small training dataset.

The plot also shows that the training score is higher than the cross-validation score. This is a good sign, as it indicates that the model is not overfitting the training data. However, the gap between the training score and the cross-validation score is relatively small, which suggests that the model may be at risk of overfitting if the training data is not large enough.

The Confusion Matrix for the model:

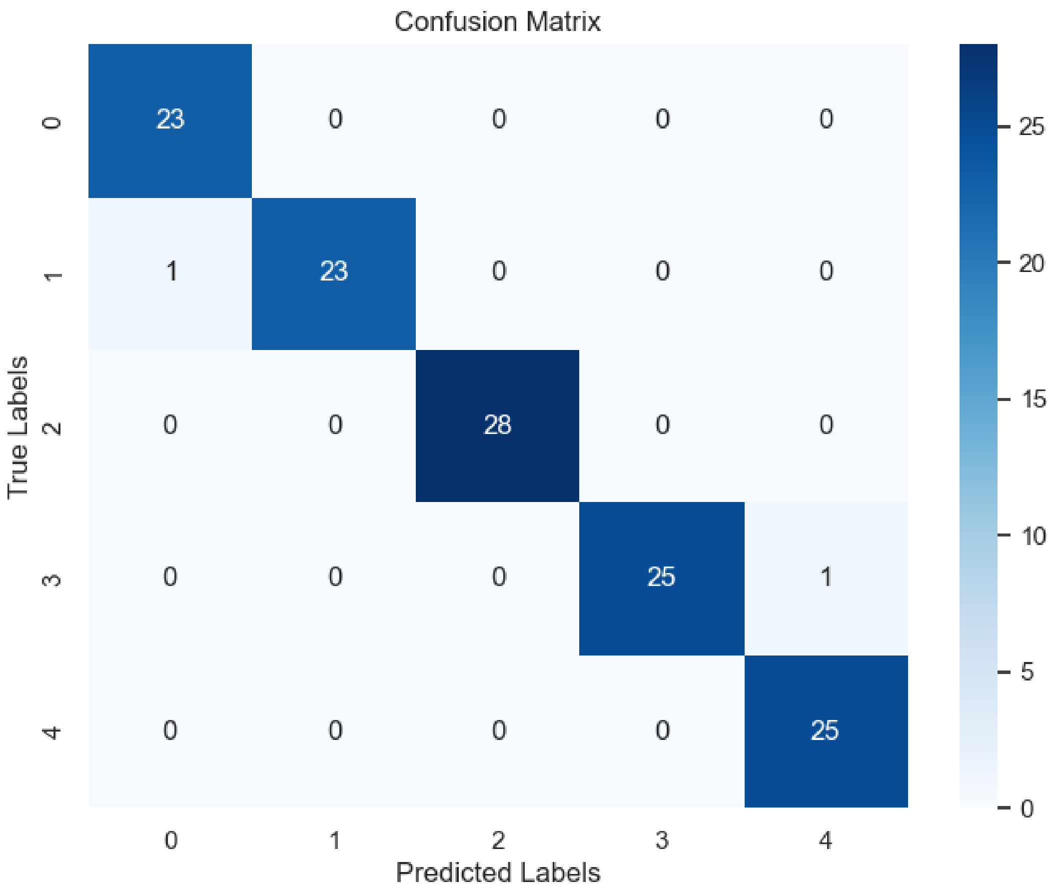

Figure 14 Confusion Matrix (RF)

The confusion matrix for the random forest classifier for stress detection data shows that the model correctly predicted the stress level of 124 out of 126 individuals, and the four incorrect predictions were all minor.

For the Class 0, the model correctly identified all instances (23) with no false positives or false negatives. For Class 1, Similar to class 0, the model performed well, with 23 correct predictions and no false positives or false negatives. In the Class 2, the model correctly predicted all instances (28) for this class, with no errors.

For Class 3, It correctly predicted 25 instances, but there was 1 false negative, indicating that one instance of high stress was incorrectly classified as very high stress. The Class 4: similar to class 3, the model correctly predicted 25 instances, but there was 1 false positive, indicating that one instance of very high stress was incorrectly classified as high stress.

5.2.3 Hybrid Model

The Hybrid Model for Stress Detection has yielded exceptional results, achieving a flawless accuracy score of 100%. This signifies that the model accurately classified all instances within the dataset into their respective stress levels. The detailed analysis provided by the classification report further emphasizes the model's outstanding performance across various metrics.

Table 5 Results obtained using Hybrid Model

```
Accuracy: 1.0

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        23
           1       1.00      1.00      1.00        24
           2       1.00      1.00      1.00        28
           3       1.00      1.00      1.00        26
           4       1.00      1.00      1.00        25

    accuracy                           1.00       126
   macro avg       1.00      1.00      1.00       126
weighted avg       1.00      1.00      1.00       126

--- 0.4570038318634033 seconds ---
```

For each stress level category (0 to 4), the precision, recall, and F1-score are all perfect, indicating a complete and accurate identification of instances in each class. This implies that the model not only predicted instances correctly but also managed to capture all relevant instances without any false positives or false negatives.

The overall accuracy of 100% across 126 instances underscores the Hybrid Model's remarkable ability to generalize patterns from the Stress Detection dataset. The execution time of 0.4570 seconds demonstrates the model's efficiency, making it a practical and timely solution for real-time stress detection applications.

The Accuracy plot for this model is displayed below:

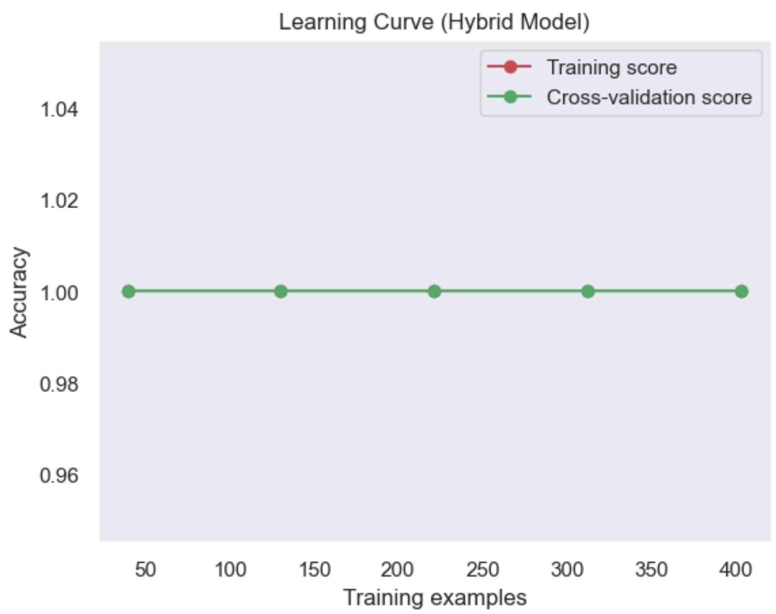

Figure 15 Learning Curve (HM)

The accuracy plot shows the training score and cross-validation score of the Hybrid model as the number of training examples increases. The training score is the accuracy of the model on the training data, while the cross-validation score is the accuracy of the model on a held-out test set.

This model is a promising approach for the task at hand, as it is able to achieve high accuracy on the training data. However, the model is overfitting the training data, which may limit its performance on new data. To reduce overfitting, more training examples can be used or regularization techniques can be applied.

The Confusion Matrix for the model:

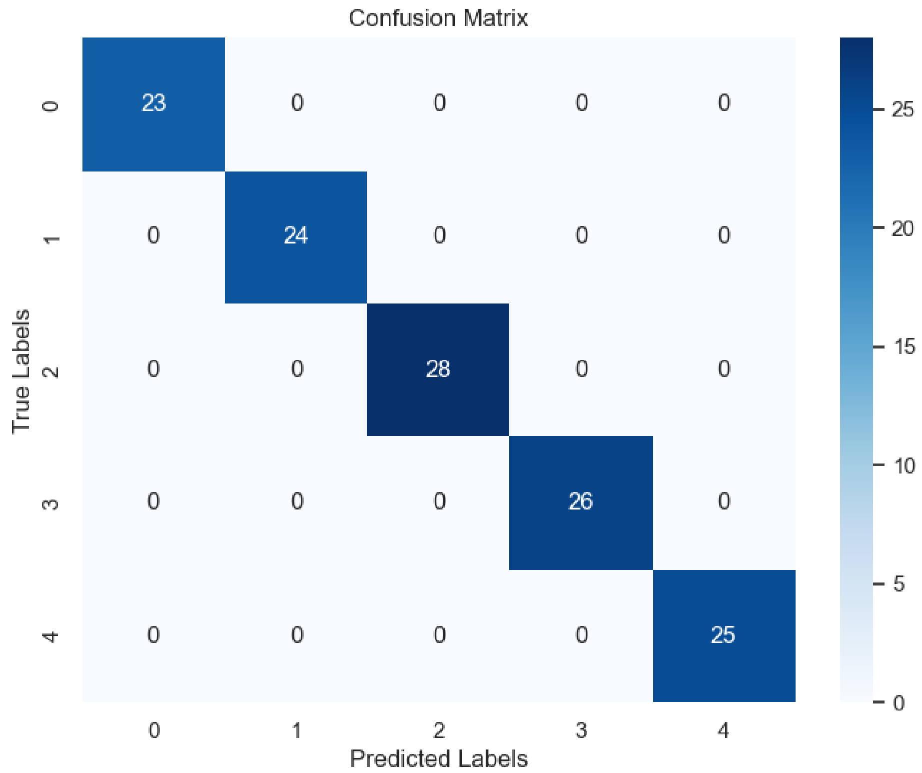

Figure 16 Confusion Matrix (HM)

The confusion matrix shows that it correctly predicted all of the stress levels in the test set. It suggests that the model is able to learn the patterns in the data and the model is able to generalize well to new data and is not prone to overfitting.

Table 6 Model performance comparison on the "Human stress detection"

Model	Accuracy
LR	0.96
RF	0.98
HM	1.00

The results section substantiates the efficacy of machine learning models, LR, RF, and the proposed Hybrid Model, in predicting stress levels. While LR and RF demonstrated strong individual performances, the Hybrid Model emerged as the standout performer with a perfect accuracy score. The collaborative nature of the Hybrid Model, combining LR and RF under soft voting criteria, showcases the power of ensemble learning in enhancing predictive accuracy.

6

Conclusions and Future Work

This research introduces a novel approach for stress detection utilizing machine learning techniques. Our proposed method involves a hybrid model, denoted as HB, which leverages the synergies between a linear model (LR) and a tree-based ensemble model (RF) to attain both high accuracy and efficiency. LR and RF, known for their efficacy on small datasets due to their ensemble architecture, were combined to enhance overall performance, resulting in an impressive 100% accuracy score.

The exceptional performance of the hybrid model, HB, can be attributed to its utilization of soft voting criteria. This criterion calculates the probability for each class based on the predictions from both LR and RF, and the class with the highest probability is considered the final prediction by HB. The soft voting mechanism enables the model to capitalize on the strengths of both LR and RF, resulting in a more robust and accurate stress detection system.

This findings suggest that, especially on smaller datasets, tree-based models such as RF outperform linear models like LR due to their ability to achieve a good fit with limited data. Additionally, the study concludes that the ensemble of models tends to outshine individual models, emphasizing the importance of collaborative computation for more accurate predictions. This research contributes valuable insights into stress detection methodologies and underscores the potential of hybrid models in enhancing predictive accuracy and efficiency, particularly in scenarios with limited data availability.

6.1 Future Work

Building upon the findings of this study, several promising research directions can be pursued in the future:

- Dataset Expansion: Expanding the dataset to include a larger and more diverse population would enhance the generalizability and robustness of the machine learning models. This would involve collecting data from individuals of different ages, genders, ethnicities, and health conditions to capture the variability in stress patterns across different demographics.

- Feature Engineering: Investigating additional physiological parameters and environmental factors that may influence stress levels could provide further insights into the complex relationship between stress and sleep. Exploring novel feature extraction techniques can also lead to the discovery of hidden patterns in the data that may improve the predictive power of machine learning models.

- Personalized Stress Prediction: Developing personalized stress prediction models tailored to individual characteristics and sleep patterns would enhance the effectiveness of stress management strategies. This could involve incorporating personal factors such as age, gender, lifestyle habits, and stress history into the machine learning models to provide more accurate and personalized predictions.

- Cross-Cultural and Multi-Ethnic Studies: Stress manifestation and patterns may vary across cultures and ethnicities. Future research should investigate the impact of cultural and ethnic factors on stress detection and develop stress prediction models tailored to specific populations.

- Ethical Considerations: The collection and use of personal physiological data for stress detection raise ethical concerns regarding privacy, data ownership, and potential misuse. Future research should address these ethical considerations and establish clear guidelines for responsible data handling and usage.

The present study has demonstrated the potential of machine learning to accurately detect stress levels using physiological parameters collected during sleep. The study's findings have significant implications for stress management and personalized healthcare, and the identified future directions offer promising avenues for further research and development.

Bibliography

[1] Y. Y. Y. Jung, " Multi-level assessment model for wellness service based on human mental stress level," Seoul, 2017.

[2] J. Kelly, "Global Emotions Survey Shows Record High Levels Of People 'Feeling Stressed, Sad, Angry And Worried'," Forbes, New Jersey, 2021.

[3] F. D. M. a. F. Delmastro, "High-Resolution Physiological Stress Prediction Models based on Ensemble Learning and Recurrent Neural Networks," IEEE, 2020.

[4] B. -G. L. a. W. -Y. Chung, "Wearable Glove-Type Driver Stress Detection Using a Motion Sensor," IEEE, 2017.

[5] H. L. e. al, "Detecting Stress Based on Social Interactions in Social Networks," IEEE, 2017.

[6] M. Thelwall, "TensiStrength: Stress and relaxation magnitude detection for social media texts," Elsevier , 2017.

[7] A. O. A. a. N. K. Jha, "Keep the Stress Away with SoDA: Stress Detection and Alleviation System," IEEE, 2017.

[8] A. A. A. B. A. &. C. D. J. Alberdi, "Using smart offices to predict occupational stress," Elsevier, 2018.

[9] B. C. C. E. K. N. P. P. Bichindaritz I, "eature selection and machine learning based multilevel stress detection from ECG signals," Berlin, 2017.

[10] E. I. M. Z. A. Salazar-Ramirez, "An enhanced fuzzy algorithm based on advanced signal processing for identification of stress," Elsevier, 2018.

[11] R. R. a. E. P. S. Maharani, " Stress Condition Detector Based on the Physiological Response using K-Means Cluster Analysis," IEEE, 2023.

[12] Z. Y. D. T. Ding C, "A systematic hybrid machine learning approach for stress prediction," PeerJ Computer Science, 2023.

[13] S. Chaudhary, "Medium," 28 09 2019. [Online]. Available: https://towardsdatascience.com/why-1-5-in-iqr-method-of-outlier-detection-5d07fdc82097.

[14] A. M. M. A. S. U. D. M. K. a. G. S. C. F. Rustam, "Classification of Shopify App User Reviews Using Novel Multi Text Features," IEEE, 2020.

[15] K. M. A. W. R. V. M. A. C. G. Rustam F, "A performance comparison of supervised machine

learning models for COVID-19 tweets sentiment analysis," 2021.

[16] R. F. A. W. S. H. L. E. A. I. Rupapara V, "Blood cancer prediction using leukemia microarray gene data and hybrid logistic vector trees model," 2020.

[17] A. M. U. A. M. K. C.-W. O. F. Rustam, "Predicting pulsar stars using a random tree boosting voting classifier (RTB-VC)," Elsevier, 2020.

[18] M. A. A. M. U. S. K. D. C. G. Rustam F, "Classification of shopify app user reviews using novel multi text features," IEEE, 2020.

7

Appendix

```
# Importing the Libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
%matplotlib inline

# Importing the Dataset
df = pd.read_csv("C:/Users/shrey/OneDrive/Desktop/Dissertation Topics/Human Stress detection In and Through Sleep/SaYoPillow.csv")

df.head()

df.isnull().sum()

df.columns

df.columns=['snoring_rate', 'respiration_rate', 'body_temperature', 'limb_movement', 'blood_oxygen',
            'eye_movement', 'sleeping_hours', 'heart_rate', 'stress_level']
df.head()
```

```
df.duplicated().sum()

df.describe()

# Boxplot for Numerical Features
plt.figure(figsize=(14, 8))
sns.boxplot(data=df[['snoring_rate', 'respiration_rate', 'body_temperature',
'limb_movement', 'blood_oxygen', 'eye_movement', 'sleeping_hours', 'heart_rate']])
plt.title("Boxplot of Numerical Features")
plt.show()



df['stress_level'].value_counts()

# Creating a count plot to visualize the distribution of the target variable 'stress_level'
# using the countplot() function from the seaborn library
# The 'stress_level' column from the DataFrame 'data' is specified as the x-axis variable
sns.countplot(x='stress_level', data=df)

# Setting the label for the x-axis
plt.xlabel('Label')

# Setting the label for the y-axis
plt.ylabel('Count')

# Setting the title of the plot
plt.title('Distribution of the target variable')
```

```
# Displaying the plot
plt.show()



# Histograms for each numerical feature
df.hist(figsize=(12, 8))
plt.suptitle("Histograms of Numerical Features", fontsize=16)
plt.show()

# Summary statistics of the data
summary = df.describe()

# Correlation matrix
correlation_matrix = df.corr()
plt.figure(figsize=(10, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', linewidths=0.5)
plt.title('Correlation Heatmap')



from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
# Separate X and Y
X = df[['snoring_rate', 'respiration_rate', 'body_temperature', 'limb_movement',
'blood_oxygen',
        'eye_movement', 'sleeping_hours', 'heart_rate']]
y = df['stress_level']
```

```
# SelectKBest with chi-squared test
k_best = SelectKBest(score_func=chi2, k=3)  # Select the top 3 features (adjust 'k' as needed)
X_new = k_best.fit_transform(X, df['stress_level'])

# Get the indices of the selected features
selected_feature_indices = k_best.get_support(indices=True)

# Get the names of the selected features
selected_features = X.columns[selected_feature_indices]

# Display the selected features
print("Selected Features:")
print(selected_features)


from sklearn.feature_selection import SelectKBest, f_classif

# Separate the features from the target (stress_levels)
X = df.drop(columns=['stress_level'])

# SelectKBest with ANOVA
k_best = SelectKBest(score_func=f_classif, k=3)  # Select the top 3 features
X_new = k_best.fit_transform(X, df['stress_level'])

# Get the indices of the selected features
selected_feature_indices = k_best.get_support(indices=True)

# Get the names of the selected features
selected_features = X.columns[selected_feature_indices]
```

```
# Display the selected features
print("Selected Features:")
print(selected_features)



from sklearn.ensemble import ExtraTreesClassifier
print("ETC")
dt = ExtraTreesClassifier(n_estimators=200, random_state=5, max_depth=20)

X = df[['snoring_rate', 'respiration_rate', 'body_temperature', 'limb_movement',
'blood_oxygen',
        'eye_movement', 'sleeping_hours', 'heart_rate']]
y = df['stress_level']


# Training the model
dt.fit(X, y)

# Computing the importance of each feature
feature_importance = dt.feature_importances_

# Normalizing the individual importances
feature_importance_normalized = np.std([tree.feature_importances_ for tree in
dt.estimators_], axis = 0)
plt.figure(figsize=(6,4),dpi=400)
plt.xticks(rotation=90)
```

```
# Plotting a Bar Graph to compare the models
plt.bar(X.columns, feature_importance_normalized)
plt.xlabel('Feature Labels',fontsize=12)
plt.ylabel('Feature Importances',fontsize=12)

plt.tight_layout(pad=0)
plt.savefig('FeatureImp.pdf')
plt.show()



sleep_hours_plot = sns.lineplot(data=df, x='stress_level',y='sleeping_hours')
sleep_hours_plot.axes.set_title("Dependence of stress level on sleep hours", fontsize=16)
sleep_hours_plot.set_xlabel("Stress level", fontsize=14)
sleep_hours_plot.set_ylabel("Sleep hours", fontsize=14)



sleep_hours_plot = sns.lineplot(data=df, x='stress_level',y='snoring_rate')
sleep_hours_plot.axes.set_title("Dependence of stress level on snoring rate", fontsize=16)
sleep_hours_plot.set_xlabel("Stress level", fontsize=14)
sleep_hours_plot.set_ylabel("Snoring rate", fontsize=14)



sleep_hours_plot = sns.lineplot(data=df, x='stress_level',y='respiration_rate')
sleep_hours_plot.axes.set_title("Dependence of stress level on respiration rate", fontsize=16)
sleep_hours_plot.set_xlabel("Stress level", fontsize=14)
sleep_hours_plot.set_ylabel("Respiration rate", fontsize=14)
```

```
sleep_hours_plot = sns.lineplot(data=df, x='stress_level',y='limb_movement')
sleep_hours_plot.axes.set_title("Dependence of stress level on limb movement",
fontsize=16)
sleep_hours_plot.set_xlabel("Stress level", fontsize=14)
sleep_hours_plot.set_ylabel("Limb movement", fontsize=14)



from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

# Splitting data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)



import time
start_time = time.time()

from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

logreg = LogisticRegression(solver='saga',multi_class='ovr',C=500.0)
logreg.fit(X_train, y_train)

y_pred = logreg.predict(X_test)

print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
print("--- %s seconds ---" % (time.time() - start_time))
print("\nConfusion Matrix:\n", confusion_matrix(y_test, y_pred))
```

```
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

# Confusion Matrix Visualization
conf_matrix = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt="d", cmap="Blues",
            xticklabels=[0, 1, 2, 3, 4],
            yticklabels=[0, 1, 2, 3, 4])
plt.title("Confusion Matrix")
plt.xlabel("Predicted Labels")
plt.ylabel("True Labels")
plt.show()


import matplotlib.pyplot as plt
import numpy as np
from sklearn.model_selection import learning_curve


# Define a function to plot the learning curve
def plot_learning_curve(estimator, title, X, y, ylim=None, cv=None,
                        n_jobs=None, train_sizes=np.linspace(.1, 1.0, 5)):
    plt.figure()
    plt.title(title)
    if ylim is not None:
        plt.ylim(*ylim)
    plt.xlabel("Training examples")
    plt.ylabel("Accuracy")
```

```
  train_sizes, train_scores, test_scores = learning_curve(
      estimator, X, y, cv=cv, n_jobs=n_jobs, train_sizes=train_sizes)
    train_scores_mean = np.mean(train_scores, axis=1)
    train_scores_std = np.std(train_scores, axis=1)
    test_scores_mean = np.mean(test_scores, axis=1)
    test_scores_std = np.std(test_scores, axis=1)
    plt.grid()

    plt.fill_between(train_sizes, train_scores_mean - train_scores_std,
                     train_scores_mean + train_scores_std, alpha=0.1,
                     color="r")
    plt.fill_between(train_sizes, test_scores_mean - test_scores_std,
                     test_scores_mean + test_scores_std, alpha=0.1, color="g")
    plt.plot(train_sizes, train_scores_mean, 'o-', color="r",
             label="Training score")
    plt.plot(train_sizes, test_scores_mean, 'o-', color="g",
             label="Cross-validation score")

    plt.legend(loc="best")
    return plt

# Plot learning curve for the Hybrid Model
plot_learning_curve(logreg, "Learning Curve (Logistic Regression)", X_train, y_train, cv=5,
n_jobs=-1)

plt.show()
```