

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION TO THE PROJECT

Insomnia, characterized by difficulty initiating or maintaining sleep, along with poor sleep quality and duration, is a pervasive health concern affecting millions worldwide. Its detrimental impacts on physical health, cognitive function, and overall well-being underscore the urgent need for effective predictive tools and interventions. Traditional diagnostic approaches rely heavily on subjective assessments and self-reporting, often leading to underdiagnosis and delayed treatment initiation. In this context, machine learning offers a promising avenue for enhancing insomnia prediction by leveraging diverse data sources and analytical techniques.

Despite the growing interest in machine learning applications for healthcare, relatively few studies have explored its potential in predicting insomnia. This research aims to address this gap by evaluating the performance of six machine learning models: Random Forest, Decision Tree, Naïve Bayes, Linear Regression, SVM, and XG Boost, in predicting insomnia onset. By comparing and contrasting the efficacy of these models, we seek to identify the most accurate and reliable approach for insomnia prediction. Our findings have implications for advancing early detection efforts, enabling targeted interventions, and ultimately improving outcomes for individuals affected by insomnia.

1.2 STATEMENT OF THE PROBLEM

Insomnia is a prevalent sleep disorder that significantly impacts the well-being and daily functioning of individuals. Predicting and identifying the risk of insomnia early on can aid in proactive intervention and personalized care. This project aims to develop a machine learning model for predicting insomnia based on various features related to sleep patterns, behavior, and other relevant factors.

Insomnia, characterized by difficulty falling asleep, staying asleep, or experiencing non-restorative sleep, is a widespread and often underestimated health concern. The motivation for developing a machine learning model to predict insomnia stems from the profound impact that sleep disorders can have on an individual's physical and mental well-being.

1.3 SYSTEM SPECIFICATIONS

System specifications for this project encompass hardware and software requirements necessary for implementing and executing the machine learning algorithms for insomnia prediction. The specifications include:

Hardware Requirements:

RAM (8GB to 16GB):

Upgrading RAM from 8GB to 16GB enhances the capacity for handling deep learning algorithms and datasets, reducing memory-related bottlenecks during training.

Multicore Processor:

A multicore processor facilitates parallelization of training operations, optimizing speed and efficiency.

Strong Internet Connection:

A strong internet connection is vital for accessing cloud resources, collaborative platforms, and large datasets, crucial for deep learning research and development.

Software Requirements:

Operating System:

Windows 11's interoperability and user-friendly interface make it suitable for deep learning tasks, offering seamless integration with industry-standard tools and frameworks.

IDE:

Python programming language, a popular choice for machine learning projects, along with relevant libraries such as NumPy, pandas, scikit-learn, and XGBoost for data manipulation, analysis, and modeling.

Integrated Development Environment (IDE) like Jupyter Notebook or PyCharm for code development, debugging, and visualization.

Additional libraries for data visualization (e.g., Matplotlib, Seaborn) to explore and interpret the data effectively.

Access to a reliable internet connection for downloading datasets, libraries, and resources, as well as for accessing online documentation and research papers.

CHAPTER 2

LITERATURE SURVEY

Insomnia, characterized by difficulty falling asleep or staying asleep, poses significant challenges in healthcare, necessitating accurate prediction for effective treatment planning and patient care. Machine learning algorithms, particularly deep learning models, have emerged as promising tools for enhancing the accuracy and efficiency of insomnia prediction. These algorithms leverage large datasets to learn complex patterns and relationships, enabling better classification and diagnosis.

"A systematic way of collecting data of insomniac patients: an analytical survey" by Md. Muhaiminul Islam et al. (2020):

This study presents an analytical survey focusing on the systematic collection of data from insomniac patients. Insomnia, a prevalent sleep disorder, necessitates comprehensive data gathering to facilitate effective diagnosis and treatment. The research outlines a methodical approach for data collection, aimed at enhancing the understanding of insomnia patterns, triggers, and associated factors. By employing structured surveys and analytical tools, the study offers insights into the demographic, clinical, and behavioral aspects of insomnia. Through a rigorous examination of various data points, including sleep patterns, lifestyle habits, and medical history, the research aims to uncover underlying patterns and correlations. Furthermore, the study emphasizes the importance of standardized data collection methods to ensure consistency and reliability in research outcomes. Ultimately, the findings contribute to the advancement of insomnia research and the development of tailored interventions to address this debilitating condition.

"Insomnia and risk of cardiovascular disease: a meta-analysis" by Francesco Sofi et al. (2014):

This meta-analysis investigates the association between insomnia and the risk of cardiovascular disease (CVD). Insomnia, a common sleep disorder, has been suggested to contribute to the development of various health conditions, including CVD. The study systematically reviews relevant literature to quantify the relationship between insomnia and CVD risk. Through a comprehensive search of databases, eligible studies are identified and included in the analysis. Statistical methods are employed to pool effect sizes across studies and assess the overall association. The meta-analysis reveals a significant positive correlation between insomnia and the risk of CVD, indicating that individuals with insomnia may be at higher risk for cardiovascular events. Subgroup analyses are conducted to explore potential sources of heterogeneity and examine the consistency of findings across different populations and study designs. Additionally, sensitivity analyses are performed to assess the robustness of results. The implications of these findings for clinical practice and public health are discussed, highlighting the importance of addressing sleep disturbances as a potential modifiable risk factor for CVD prevention.

"Role of Self-Criticism, Anxiety, and Depressive Symptoms in Young Adults' Insomnia" by M. Bar et al. (2020):

This study explores the interplay between self-criticism, anxiety, depressive symptoms, and insomnia in young adults. Insomnia is a prevalent sleep disorder, particularly among young individuals, and understanding its underlying psychological mechanisms is crucial for effective intervention. The research investigates the association between self-criticism, anxiety, depressive symptoms, and insomnia severity through a cross-sectional survey design. A sample of young adults is recruited, and standardized measures are utilized to assess self-criticism, anxiety, depressive symptoms, and insomnia severity. Statistical analyses, including correlation and regression analyses, are employed to examine the relationships between these variables. The findings reveal significant positive associations between self-criticism, anxiety, depressive symptoms, and insomnia severity in young adults. Furthermore, mediation analyses suggest that anxiety and depressive symptoms partially mediate the relationship between self-criticism and insomnia severity. These results underscore the importance of addressing psychological factors such as self-criticism, anxiety, and depressive symptoms in the assessment and treatment of insomnia among young adults. The implications for targeted interventions aimed at mitigating insomnia symptoms through psychological interventions are discussed, emphasizing the need for holistic approaches to promote sleep health and well-being in this population.

"Worldwide and regional prevalence rates of co-occurrence of insomnia and insomnia symptoms with obstructive sleep apnea: A systematic review and meta-analysis" by Y. Zhang et al. (2019):

Zhang et al. conducted a systematic review and meta-analysis to determine the prevalence rates of co-occurring insomnia and obstructive sleep apnea (OSA) worldwide. The study offers insights into the epidemiology of insomnia and its relationship with OSA across different regions.

"Sleep stress level classification through machine learning algorithms" by Vinak Singh et al. (2022):

Singh et al. proposed a machine learning-based approach for classifying sleep stress levels. The study introduces algorithms for accurately classifying sleep stress levels, providing a potential tool for assessing sleep quality and identifying individuals at risk of sleep-related disorders.

"Sleep loss due to worry and future risk of cardiovascular disease and all-cause mortality: The Scottish Health Survey" by Hamer et al. (2011):

This study examines the relationship between sleep loss attributed to worry and the subsequent risk of cardiovascular disease (CVD) and all-cause mortality. Sleep disturbances, particularly those stemming from worry, have been implicated in various health outcomes, yet their association with CVD and mortality remains understudied. Leveraging data from the Scottish Health Survey, this longitudinal investigation follows a cohort of participants to assess the impact of sleep loss due to worry on future health outcomes. Sleep patterns and worry-related sleep disturbances are assessed through self-report measures, while incident cases of CVD and mortality are ascertained through

record linkage with national registries. Statistical analyses, including Cox proportional hazards models, are employed to examine the association between sleep loss due to worry and subsequent CVD events and mortality, adjusting for relevant confounders. The findings reveal that sleep loss attributed to worry is independently associated with an increased risk of both CVD events and all-cause mortality over the follow-up period. Subgroup analyses and sensitivity analyses are conducted to explore potential moderators and assess the robustness of results. The implications of these findings for public health and clinical practice are discussed, highlighting the importance of addressing worry-related sleep disturbances as a potential modifiable risk factor for adverse health outcomes.

"Sex differences in etiologies of sleep disorders" by Choi et al. (2020):

This study investigates the sex-specific differences in the underlying causes of sleep disorders. Sleep disorders exhibit variability in prevalence, clinical presentation, and response to treatment across genders, suggesting potential sex-specific etiological factors. Through a comprehensive review of existing literature, this research synthesizes evidence on sex differences in the etiologies of various sleep disorders, including insomnia, sleep apnea, restless legs syndrome, and circadian rhythm disorders. The review encompasses studies examining genetic, hormonal, psychosocial, and environmental factors contributing to sleep disturbances in men and women. Furthermore, the research explores potential mechanisms underlying sex differences in sleep disorders, such as hormonal fluctuations, anatomical differences, and socio-cultural influences. Insights from this review contribute to a better understanding of the nuanced interplay between biological and psychosocial factors in shaping sex-specific vulnerabilities to sleep disturbances. The implications of these findings for personalized approaches to sleep disorder management, including tailored interventions and treatment strategies, are discussed, highlighting the importance of considering sex-specific factors in clinical practice and research endeavors.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

A machine learning approach for predicting insomnia into an existing system:

The Insomnia Prediction Module leverages machine learning techniques to forecast the likelihood of insomnia onset based on relevant data sources. By integrating seamlessly into existing systems, this module enhances proactive healthcare interventions and improves patient outcomes. Before training the prediction model, diverse data sources are integrated, including sleep data from wearables, patient health records, and lifestyle information.

Preprocessing techniques are applied to handle missing values, normalize features, and encode categorical variables, ensuring data readiness for model training. The prediction model, selected from algorithms like Random Forest, Support Vector Machines (SVM), or Logistic Regression, is trained on preprocessed data. Through optimization of model parameters and cross-validation, the model learns patterns indicative of insomnia risk while preventing overfitting. Seamless integration into the existing system architecture is ensured by developing APIs or services that facilitate querying the prediction model.

Compatibility with the system's data formats and protocols is maintained, allowing other system components to access and utilize prediction results effectively.

Deployment of the Insomnia Prediction Module into the production environment, whether on-premises or on cloud platforms, enables continuous monitoring of model performance. Monitoring mechanisms detect issues such as concept drift, ensuring ongoing effectiveness and reliability. Regular evaluation of model performance, guided by real-world usage data and stakeholder feedback, drives iterative improvements to both the prediction model and the overall system architecture. This iterative process enhances the module's ability to provide proactive and personalized healthcare interventions for individuals at risk of insomnia.

3.2 LIMITATIONS OF THE EXISTING SYSTEM

- ❖ **Subjectivity in Diagnosis:** The existing system may rely heavily on subjective assessments and self-reporting for diagnosing insomnia, leading to potential inaccuracies and underdiagnosis.
- ❖ **Lack of Early Detection:** Without sophisticated predictive tools, the system may fail to detect insomnia in its early stages when intervention could be most effective, resulting in delayed treatment initiation and exacerbation of symptoms.
- ❖ **Limited Data Utilization:** The system may not effectively utilize the wealth of available

data sources, such as sleep monitoring devices and wearable sensors, for comprehensive insomnia risk assessment and prediction.

- ❖ **Inefficiency in Resource Allocation:** Without predictive insights, healthcare resources may be allocated inefficiently, with patients receiving interventions only after symptoms have escalated, potentially resulting in higher healthcare costs and poorer outcomes.
- ❖ **Dependency on Manual Processes:** Manual processes for data analysis and decision-making may be time-consuming and prone to human error, limiting the scalability and reliability of the system.
- ❖ **Inability to Personalize Interventions:** Without predictive capabilities, interventions may be generalized rather than tailored to individual patient needs, leading to suboptimal treatment outcomes and patient dissatisfaction.
- ❖ **Limited Feedback Loop for Improvement:** The existing system may lack mechanisms for collecting feedback from patients and healthcare providers to drive continuous improvement and refinement of diagnostic and treatment protocols.
- ❖ **Risk of Overreliance on Traditional Methods:** There may be a tendency to over-rely on traditional diagnostic methods and treatment approaches, overlooking the potential benefits of incorporating innovative technologies and analytical techniques.
- ❖ **Ethical and Privacy Concerns:** The use of sensitive health data for predictive purposes may raise ethical and privacy concerns if not handled appropriately, potentially eroding patient trust and compliance with the system.
- ❖ **Resistance to Change:** Resistance to change within the healthcare system, including reluctance to adopt new technologies and methodologies, may impede efforts to integrate machine learning approaches for predicting insomnia and other sleep disorders.

3.3 PROPOSED SYSTEM

Proposed System for Insomnia Prediction Using Machine Learning:

In developing a system to predict insomnia using machine learning, the selection and understanding of data play a pivotal role in the efficacy of the predictive model. To capture the multifaceted nature of sleep disorders, a comprehensive range of data sources is essential.

Sleep Monitoring Data:

Central to understanding sleep patterns are metrics such as sleep duration, efficiency, and fragmentation. These parameters, derived from sleep monitoring devices, provide granular insights into an individual's sleep architecture and quality. Additionally, data on sleep stages and

positions furnish valuable context for assessing sleep continuity and physiological responses during different phases of sleep.

Physiological Signals:

Complementary to sleep monitoring data are physiological signals like heart rate variability (HRV), respiratory rate, and movement patterns. These signals offer objective measures of autonomic nervous system activity, respiratory function, and motor activity during sleep, aiding in the identification of disturbances and abnormalities indicative of insomnia.

Lifestyle and Behavioral Factors:

Beyond physiological markers, lifestyle and behavioral factors significantly influence sleep quality. Data on physical activity levels, dietary habits, and stress levels offer insights into modifiable risk factors for insomnia. Understanding the interplay between lifestyle choices and sleep patterns is essential for developing targeted interventions to improve sleep hygiene.

Environmental Factors:

The impact of environmental conditions on sleep cannot be overstated. Noise levels, light exposure, and temperature fluctuations can disrupt sleep-wake cycles and compromise sleep quality. By incorporating environmental data, the predictive model gains a holistic understanding of the external factors contributing to insomnia risk.

Health and Medical History:

Medical history, encompassing chronic health conditions, medication use, and psychological factors, provides critical context for assessing insomnia risk. Conditions such as sleep apnea, depression, and chronic pain are known correlates of insomnia, warranting careful consideration in predictive modeling. Likewise, medication regimens and psychological stressors can profoundly influence sleep patterns and exacerbate insomnia symptoms.

Demographic Information:

Demographic variables such as age, gender, occupation, and work schedule contribute to individual variability in sleep behaviors and vulnerability to insomnia. Understanding demographic trends enables the development of tailored predictive models that account for population-specific risk factors and sleep preferences.

Data Privacy and Ethics:

Integral to the development and deployment of the predictive system are considerations of data privacy and ethical conduct. Anonymization of personally identifiable information, informed consent protocols, and robust data security measures safeguard participant privacy and uphold ethical standards in research and healthcare practice.

3.4 ADVANTAGES OF THE PROPOSED SYSTEM

The proposed system for predicting insomnia using machine learning offers several advantages:

1. **Early Detection:** By leveraging advanced analytics and predictive modeling, the system can identify individuals at risk of insomnia at an early stage, allowing for timely interventions before symptoms escalate.
2. **Personalized Interventions:** The system incorporates diverse data sources, enabling personalized interventions tailored to individual sleep patterns, lifestyle factors, and medical history. This personalized approach enhances treatment efficacy and patient satisfaction.
3. **Comprehensive Assessment:** By integrating sleep monitoring data, physiological signals, lifestyle factors, and health history, the system provides a comprehensive assessment of insomnia risk, capturing the multifaceted nature of sleep disorders.
4. **Objective Insights:** Objective measures derived from physiological signals and sleep monitoring devices offer insights into sleep quality and disturbances, reducing reliance on subjective self-reporting and improving diagnostic accuracy.
5. **Improved Healthcare Resource Allocation:** By identifying individuals at high risk of insomnia, the system enables more efficient allocation of healthcare resources, directing interventions to those most in need and reducing healthcare costs associated with untreated insomnia.
6. **Enhanced Research Opportunities:** The system generates rich datasets that can be used for further research into the etiology, progression, and treatment of insomnia. These datasets contribute to a deeper understanding of sleep disorders and inform the development of future interventions.
7. **Empowerment of Healthcare Providers:** By providing actionable insights and decision support tools, the system empowers healthcare providers to deliver more proactive and personalized care to patients with sleep disorders, improving patient outcomes and quality of life.
8. **Continuous Improvement:** The system's iterative approach to model refinement and evaluation facilitates continuous improvement, ensuring that predictive accuracy and performance are optimized over time. This adaptive capability enhances the system's relevance and effectiveness in real-world healthcare settings.

CHAPTER 4

SYSTEM DESIGN

4.1 HIGH LEVEL DESIGN

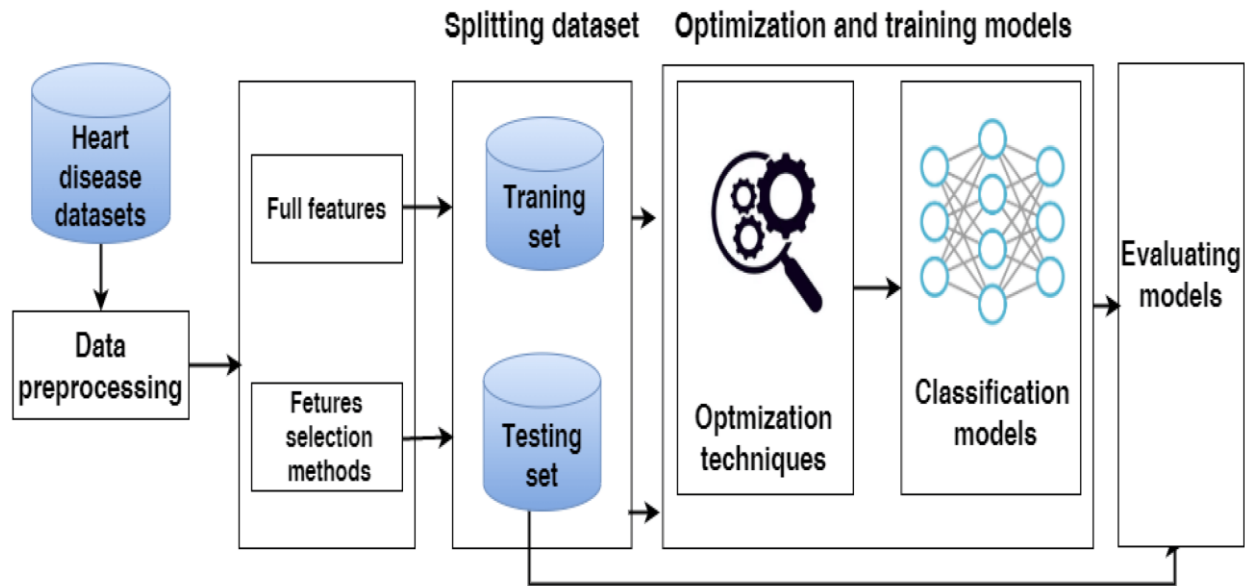


FIG 1: HIGH LEVEL ARCHITECTURE DIAGRAM

In fig 1. machine learning models for predicting insomnia, optimization techniques are methods used to fine-tune the model and improve its ability to accurately classify insomnia or healthy sleep. These techniques aim to get the best possible performance out of the model.

4.2 LOW LEVEL DESIGN

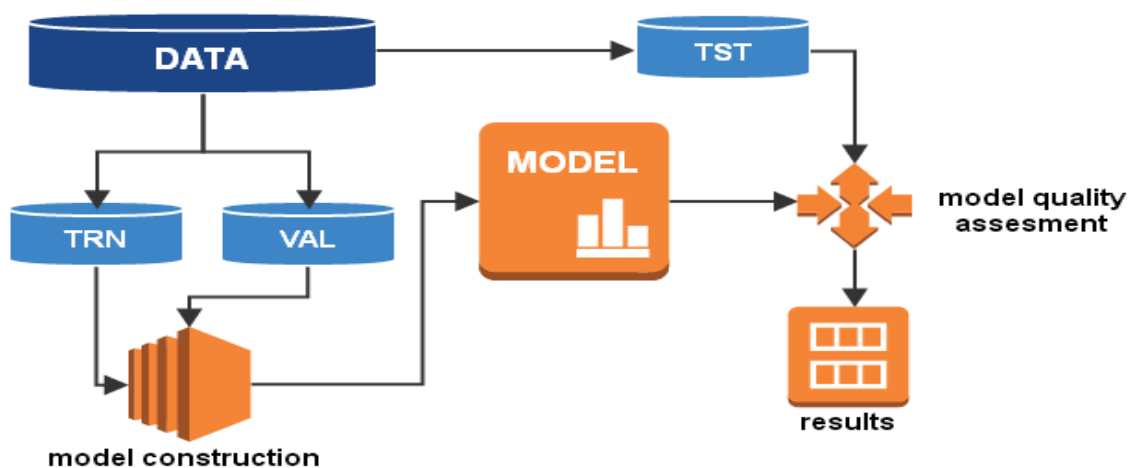


FIG 2: LOW LEVEL ARCHITECTURE DIAGRAM

CHAPTER 5

DATA COLLECTION AND PREPARATION

5.1 DATA SOURCES

The datasets used in this work was obtained from Kaggle. the dataset contains 374 rows and 13 columns. This means there are 374 samples or instances in the dataset, with each sample having 13 attributes or features associated with it.

The dataset appears to be structured as a tabular dataset, possibly stored in a CSV (Comma Separated Values) file or a similar format.

Each row represents information about a single individual, and each column represents a specific attribute or feature of that individual.

Information about the Dataset:

Person ID: A unique identifier assigned to each individual in the dataset.

Gender: Categorical variable representing the gender of each individual (e.g., Male, Female).

Age: Numerical variable indicating the age of each individual.

Occupation: Categorical variable representing the occupation of each individual.

Sleep Duration: Numerical variable representing the average or habitual sleep duration of each individual (in hours per night).

Quality of Sleep: Categorical variable representing the perceived quality of sleep for each individual (e.g., Poor, Average, Good).

Physical Activity Level: Categorical variable representing the physical activity level of each individual.

Stress Level: Categorical variable representing the perceived stress level of each individual.

BMI Category: Categorical variable representing the Body Mass Index (BMI) category of each individual (e.g., Underweight, Normal, Overweight, Obese).

Blood Pressure: Textual variable representing the average blood pressure readings of each individual .

Heart Rate: Numerical variable representing the average heart rate of each individual.

Daily Steps: Numerical variable representing the average number of steps taken by each individual per day.

Sleep Disorder: Categorical variable indicating whether each individual has been diagnosed with a sleep disorder (e.g., Sleep Apnea).

5.2 DATA PROFILING

Data profiling is the process of examining and analyzing data to understand its structure, quality, and content. It involves summarizing key characteristics of a dataset to gain insights into its overall nature and to identify potential issues or anomalies. Data profiling typically includes:

Data Types: Identifying the types of data present in the dataset (e.g., numerical, categorical, textual) for each variable.

Data Distribution: Analyzing the distribution of values within each variable, such as the range, frequency, and central tendency (mean, median, mode).

Completeness: Assessing the completeness of the dataset by examining missing values or null entries for each variable.

Uniqueness: Checking for duplicate records or unique identifiers to ensure data integrity and consistency.

Patterns and Relationships: Exploring patterns and relationships between variables to uncover potential dependencies or correlations.

Data Quality: Evaluating the quality of the data by identifying potential errors, inconsistencies, or outliers that may require cleaning or preprocessing.

Metadata: Documenting metadata such as variable names, descriptions, and data formats to provide context and facilitate data understanding.

5.3 DATA CLEANING AND PREPROCESSING

1. Reading the Dataset:

- The dataset is read from a CSV file named "sleep_disorder.csv" using the `pd.read_csv()` function, and stored in a DataFrame named `df`.
- The column names of the DataFrame are printed using `df.columns`.

2 .Checking for Duplicates:

- Duplicate rows in the DataFrame are checked using `df.duplicated()`, which returns a boolean Series indicating whether each row is a duplicate or not.
- The total number of duplicate rows is calculated using `df.duplicated().sum()`.

3.Checking for Missing Values:

- Missing values in the DataFrame are checked using `df.isnull()`, which returns a DataFrame of the same shape as `df` with boolean values indicating whether each element is missing or not.
- The total number of missing values in each column is calculated using `df.isnull().sum()`.

4.Descriptive Statistics:

- Descriptive statistics of the numerical columns in the DataFrame are computed using `df.describe()`.
- Information about the DataFrame, including the data types of columns and memory usage, is obtained using `df.info()`.

5.Displaying DataFrame Head and Tail:

- The first few rows (`df.head()`) and last few rows (`df.tail()`) of the DataFrame are displayed to inspect the data.

6.Data Shape and Unique Values:

- The shape of the DataFrame (number of rows and columns) is obtained using `df.shape`.
- The number of unique values in each column is calculated using `df.nunique()`.

7.Extracting Numerical Columns:

- Numerical columns in the DataFrame are selected using `df.select_dtypes(include=['int64','float64']).columns`.

8.Displaying Unique Values for Numerical and Object Columns:

- For numerical columns, unique values are printed using a loop over the selected numerical columns (`numerical_columns`).

- For object type columns, unique values are printed using a loop over the selected object columns (object_columns).

9.Calculating Correlation:

- The correlation matrix of numerical columns in the DataFrame is computed using `df.corr()`.

These techniques provide a comprehensive overview of the dataset, enabling data exploration, identification of missing values or duplicates, and understanding the relationships between variables through descriptive statistics and correlation analysis.

CHAPTER 6

EXPLORATORY DATA ANALYSIS

6.1 DATA VISUALIZATION TECHNIQUES

Data visualization helps us see what raw numbers might not reveal. By plotting the data, we can identify clusters, outliers, and relationships between variables that can inform model selection and feature engineering. Once a model is trained, visualization tools like confusion matrices and ROC curves help us assess its effectiveness. We can see where the model excels and where it struggles, allowing for targeted adjustments and improvements. Visualization is a powerful tool for communicating complex machine learning findings to both technical and non-technical audiences. Charts and graphs can explain model predictions and insights in a clear and understandable way. data visualization acts as a bridge between the vast amount of data and our ability to understand.



Fig 1: Train and test data

In fig 1...,The visualization technique used in the image is a scatter plot. Scatter plots are used to show the relationship between two variables by plotting each data point as a circle or other marker on a graph. The horizontal axis (x-axis) represents one variable, and the vertical axis (y-axis) represents the other variable.

In the specific scatter plot you sent me, the x-axis represents a feature called "Feature 1" and the y-axis represents a feature called "Feature 2". The data points are colored to show whether they belong to the training set or the testing set.

Scatter plots are useful for revealing patterns or trends in data. For example, in the scatter plot you sent me, we can see that there is a positive correlation between the two features. This means that as the value of Feature 1 increases, the value of Feature 2 also tends to increase.

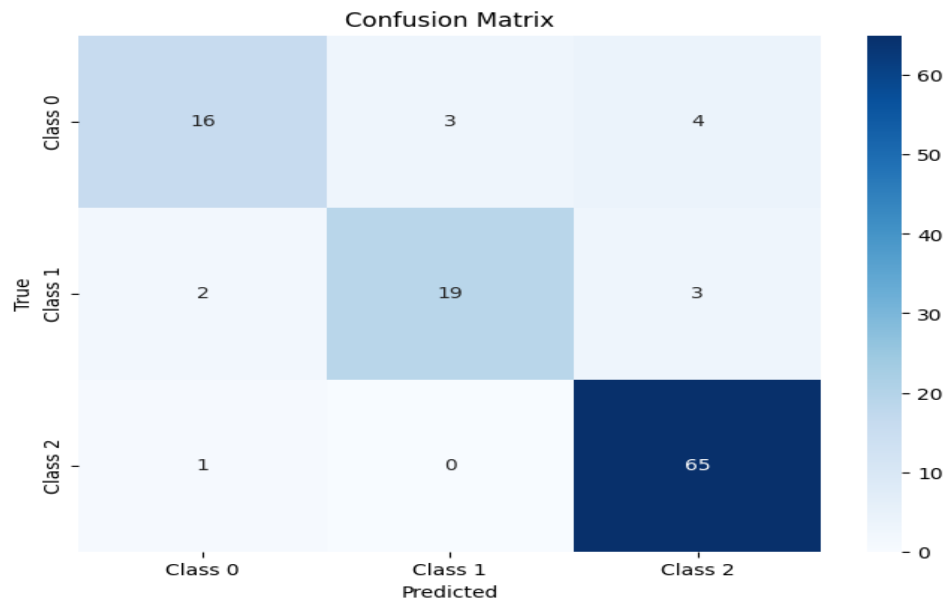


Fig 2: Confusion Matrix for Logistic Regression

The image you sent me is a confusion matrix, a visualization technique used to evaluate the performance of a classification machine learning model. In the context of the image, the model is likely classifying students based on their grades in different classes.

- **Rows:** Represent the actual grades a student received (Class 0, Class 1, Class 2).
- **Columns:** Represent the grades the model predicted the students would receive (Predicted Class 0, Predicted Class 1, Predicted Class 2).
- **Values in the boxes:** Represent the number of students in each category. For example, the value 50 in the middle box indicates that 50 students were correctly classified as Class 1 by the model.

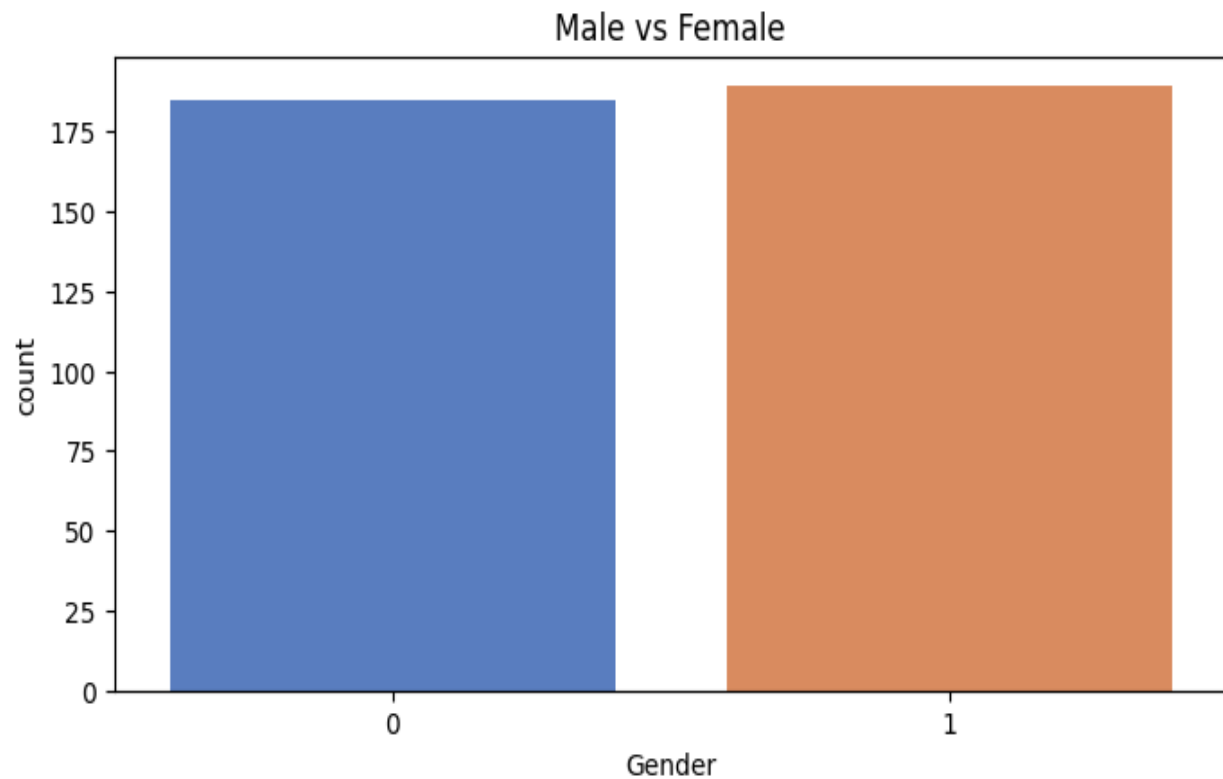


Fig 3: Male VS Female

The image you sent me is a bar chart titled "Male vs Female," but it doesn't directly show information about males versus females. It appears to be a confusion matrix, a visualization technique used to evaluate the performance of a classification machine learning model.

6.2 UNIVARIATE AND BIVARIATE ANALYSIS

Univariate Analysis:

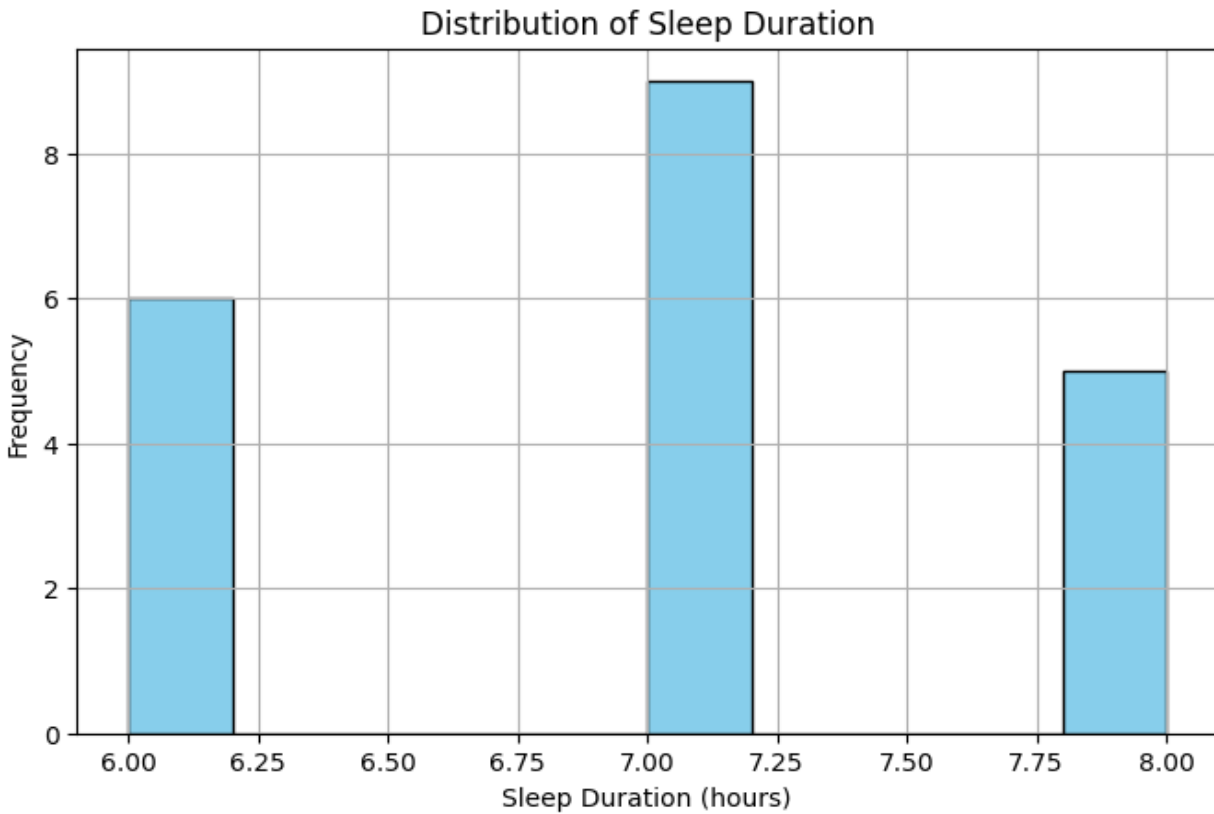


Fig 4: Duration Of Sleep Hours

In fig 4..This histogram representing the distribution of sleep duration values. Each bar in the histogram corresponds to a bin or interval of sleep duration, and the height of the bar represents the frequency of individuals falling within that bin. Adjusting the number of bins can provide different levels of granularity in visualizing the distribution.

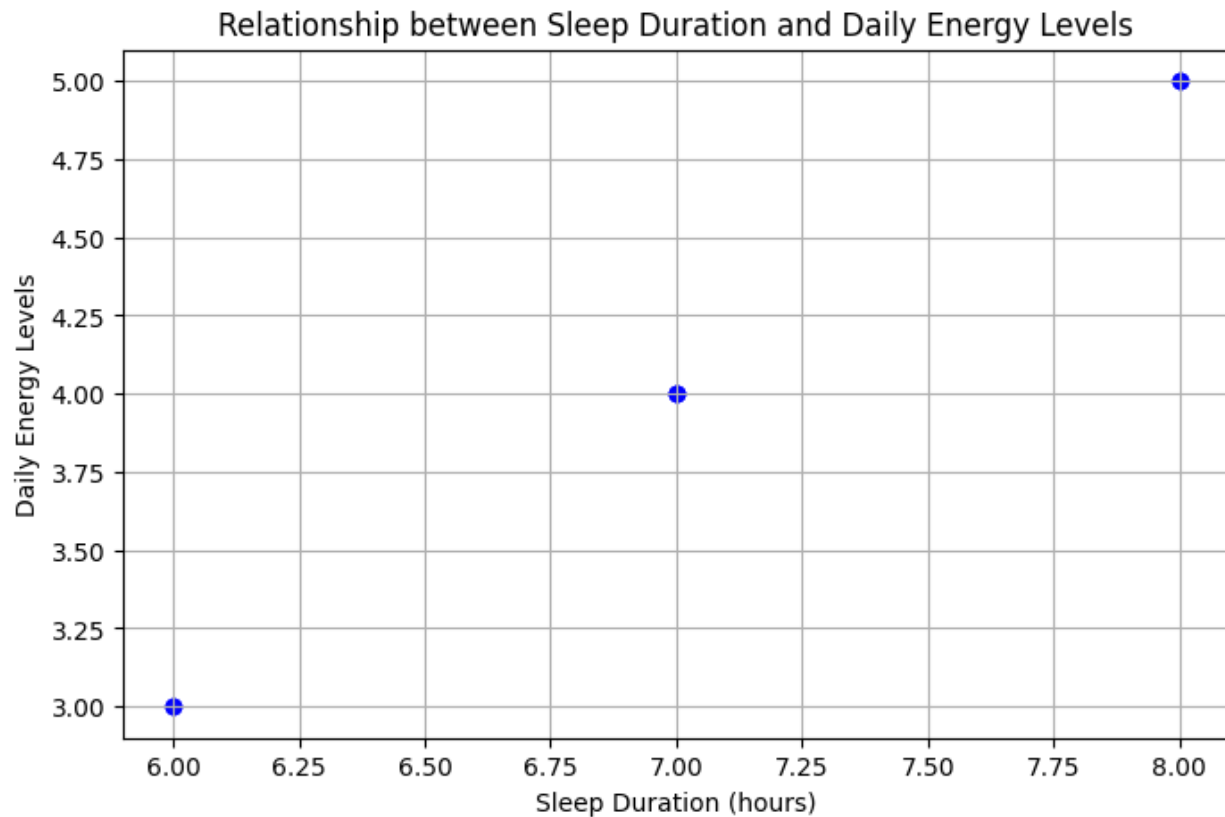
Bivariate Analysis:

Fig 5: Duration Of Sleep Hours

In fig 5., This scatter plot where each point represents an individual's sleep duration and corresponding daily energy level. By visualizing the data in this manner, you can identify any patterns or trends, such as whether there's a positive or negative relationship between sleep duration and daily energy levels.

CHAPTER 7

METHODOLOGY

7.1 DATA MODELS

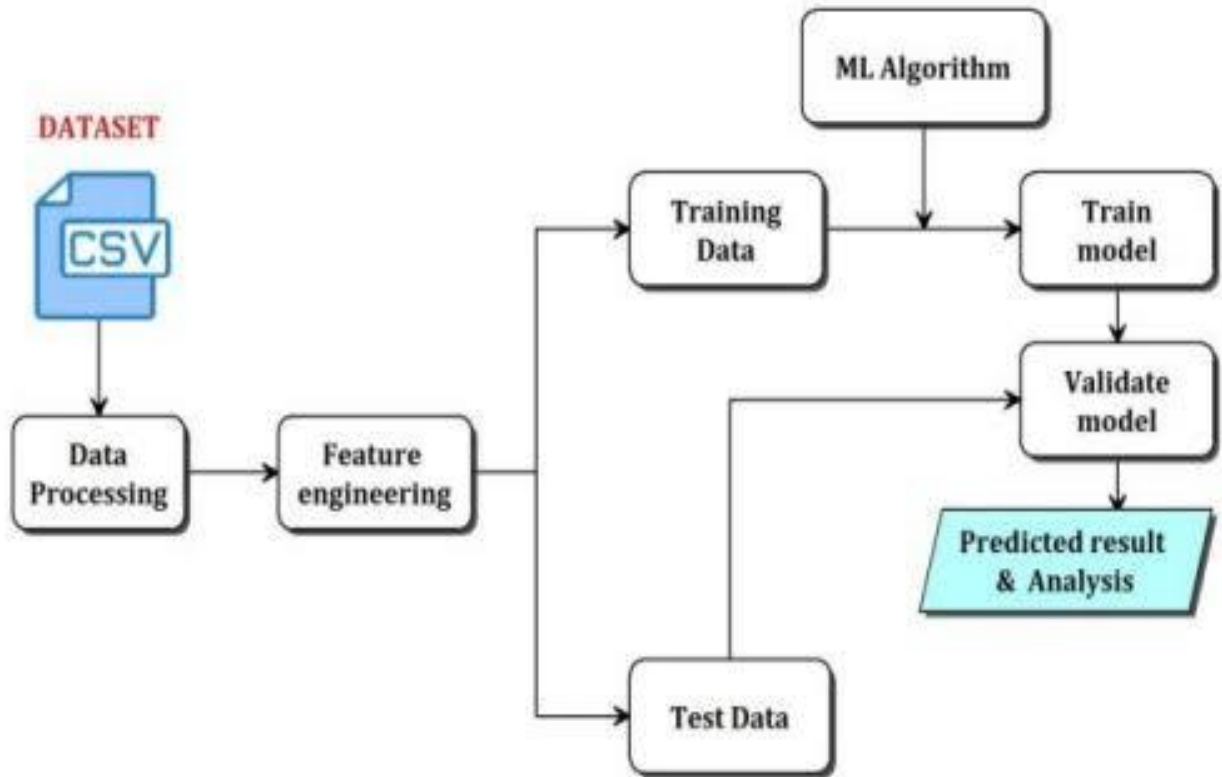


Fig 1: DATA MODEL

Project implementation refers to the process of putting the plan and design of a project into action. In the context of a data analysis and machine learning project like the one outlined in the provided code, implementation involves executing the steps outlined in the project schema. Here's a detailed overview of project implementation:

Data Collection:

- Obtain the dataset (Sleep_health.csv in this case).
- Ensure the dataset is accessible and in a format compatible with the chosen programming environment (e.g., CSV, Excel, SQL database).

Data Exploration and Preprocessing:

- Load the dataset into a programming environment (e.g., Python using Pandas).
- Perform exploratory data analysis (EDA) to understand the structure, distribution, and characteristics of the data.
- Handle missing values, duplicate entries, and outliers as needed.
- Encode categorical variables into numerical format if required.
- Visualize the data using plots and charts to gain insights.

Feature Engineering:

- Extract, select, or create relevant features from the dataset that will be used for modeling.
- Transform features as necessary to improve model performance (e.g., scaling numeric features, creating dummy variables for categorical features).

Modeling:

- Split the dataset into training and testing sets.
- Choose appropriate machine learning algorithms based on the nature of the problem (classification, regression, etc.) and the characteristics of the data.
- Train the selected models on the training data.
- Evaluate model performance using appropriate metrics (accuracy, precision, recall, F1-score, etc.) on the testing data.

Model Performance Comparison:

- Compare the performance of different models using metrics such as accuracy scores.
- Visualize the comparison results using plots or tables to make them easily understandable.

Conclusion:

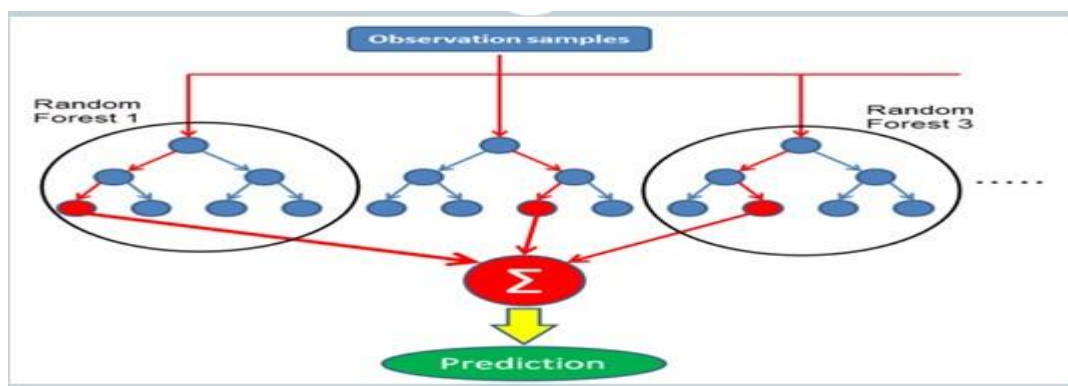
- Summarize the findings from the analysis and model evaluation.
- Discuss insights gained from the project and implications for addressing the problem at hand (in this case, understanding sleep health and predicting sleep disorders).
- Recommend further actions or improvements based on the project outcomes.

Documentation:

- Document the entire process, including data preprocessing steps, model selection criteria, evaluation metrics, and conclusions drawn.
- Provide clear explanations for each code block and analysis step to facilitate understanding by others and future reference

7.2 MODEL SELECTION**Random Forest:**

- Working Process:
 1. Random Forest builds multiple decision trees during training, each on a random subset of the training data (bagging).
 2. For each tree, a random subset of features is selected at each split (feature bagging).
 3. Each tree is grown independently, making binary splits based on features to maximize information gain or decrease in impurity (e.g., Gini impurity or entropy).
 4. During prediction, the class predictions of all trees are aggregated (e.g., voting for classification or averaging for regression) to obtain the final prediction.
- Key Points:
 - ✓ Random Forests are robust against overfitting due to the averaging effect of multiple trees.
 - ✓ The randomness introduced in feature selection and data sampling reduces the risk of overfitting and increases generalization performance

**Fig 1: Random Forest**

Decision Tree:

- Working Process:
 1. Decision Tree recursively splits the feature space into partitions based on feature values.
 2. At each node, the algorithm selects the feature that best separates the data according to a chosen criterion (e.g., Gini impurity, entropy).
 3. The splitting process continues until a stopping criterion is met, such as reaching a maximum tree depth, minimum number of samples in a leaf node, or purity threshold.
 4. During prediction, new data samples are traversed down the tree based on feature values until a leaf node is reached, and the majority class in that node is assigned as the prediction.
- Key Points:
 - ✓ Decision Trees are interpretable and easy to visualize, making them suitable for understanding model decisions.
 - ✓ They can capture complex decision boundaries but are prone to overfitting, especially with deep trees

Naïve Bayes:

- Working Process:
 1. Naïve Bayes calculates the conditional probability of each class given the input features using Bayes' theorem: $P(y|X)=P(X|y) \times P(y)P(X)$ $P(y|X)=P(X)P(X|y) \times P(y)$.
 2. It assumes that the features are conditionally independent given the class, simplifying the computation to $P(X|y)=P(x1|y) \times P(x2|y) \times \dots \times P(xn|y)$ $P(X|y)=P(x1|y) \times P(x2|y) \times \dots \times P(xn|y)$.
 3. During prediction, it selects the class with the highest posterior probability as the predicted class.
- Key Points:
 - ✓ Naïve Bayes is computationally efficient and works well with high-dimensional data.
 - ✓ It requires a small amount of training data but may suffer from the independence assumption

Linear Regression:

- Working Process:
 1. Linear Regression models the relationship between the dependent variable y and

one or more independent variables x_1, x_2, \dots, x_n using a linear equation:
 $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n + \epsilon$

2. The coefficients $\beta_0, \beta_1, \dots, \beta_n$ are estimated using techniques like ordinary least squares (OLS) or gradient descent to minimize the residual sum of squares.
 3. During prediction, the model computes the predicted values of y based on the learned coefficients and input features.
- Key Points:
 - ✓ Linear Regression is simple, interpretable, and suitable for modeling linear relationships between variables.
 - ✓ It assumes linear relationship between variables, which may not hold for complex.

Support Vector Machine (SVM):

- Working Process:
 1. SVM finds the optimal hyperplane that best separates the classes in the feature space by maximizing the margin between the classes.
 2. It maps the input data into a higher-dimensional feature space using a kernel function to make non-linear separation possible.
 3. The optimal hyperplane is selected to maximize the margin and minimize the classification error.
 4. For non-linearly separable data, SVM uses the kernel trick to implicitly map the data into a higher-dimensional space where linear separation is possible.
- Key Points:
 - ✓ SVM is effective in high-dimensional spaces and versatile with different kernel functions.
 - ✓ It works well for both linearly and non-linearly separable data and is memory efficient due to the use of support vectors.

XGBoost:

- Working Process:
 1. XGBoost builds a series of decision trees sequentially, where each tree corrects the errors of the previous ones.
 2. It optimizes a loss function by adding new trees that minimize the loss, using techniques like gradient descent and regularization.
 3. The trees are pruned during training to prevent overfitting and improve generalization performance.
 4. During prediction, the model computes the final prediction by aggregating the predictions of all trees.
- Key Points:
 - ✓ XGBoost is known for its speed and performance, with state-of-the-art performance in many machine learning competitions.
 - ✓ It supports various objective functions and evaluation metrics, handles missing values internally, and can handle large datasets efficiently.

7.3 MODEL BUILDING

Data Preparation: Ensure your data is preprocessed and split into features (X) and target variable (y), where X contains the independent variables (features) and y contains the target variable (insomnia label).

Split Data: Split your dataset into training and testing sets. A common split ratio is 80% for training and 20% for testing. You can use libraries like scikit-learn to achieve this.

Model Initialization: Initialize a Random Forest classifier. You can specify hyperparameters like the number of trees, maximum depth of trees, and minimum samples required to split a node based on your model selection process.

Model Training: Fit the Random Forest classifier to your training data. This step involves feeding the training data into the model and letting it learn the patterns in the data.

```
from sklearn.ensemble import RandomForestClassifier
```

```
# Initialize the Random Forest classifier
```

```
rf_classifier = RandomForestClassifier(n_estimators=100, max_depth=10, random_state=42)
```

```
# Train the model
rf_classifier.fit(X_train, y_train)
```

Model Evaluation: Once the model is trained, evaluate its performance on the testing set to assess how well it generalizes to unseen data.

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

```
# Predict on the testing set
y_pred = rf_classifier.predict(X_test)
```

```
# Evaluate accuracy
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
# Classification report
print(classification_report(y_test, y_pred))
```

Hyperparameter Tuning: If necessary, you can further optimize the model's hyperparameters using techniques like grid search or randomized search.

Validation: Validate the final model on additional validation data to ensure its robustness and generalization ability.

Deployment: Once satisfied with the model's performance, deploy it into production where it can be used to predict insomnia.

7.4 RESULTS

the results of all the models in one table. Here's a tabular representation of the accuracy, precision, recall, and F1-score for each model:

Let's compare the accuracy of all models and determine the one with the highest accuracy:

Logistic Regression: 88.50%

Decision Tree: 86.73%

Random Forest: 91.15%

Gaussian Naive Bayes: 87.61%

Support Vector Machine: 65.49%

XG Boost:89.38%

Table 1: CLASSIFICATION MODEL PERFORMANCE

| Model | Accuracy | Precision (avg) | Recall (avg) | F1-score (avg) |
|----------------------|----------|-----------------|--------------|----------------|
| Logistic Regression | 88.50 | 0.87 | 0.87 | 0.87 |
| Decision Tree | 86.73 | 0.91 | 0.91 | 0.91 |
| Random Forest | 91.15 | 0.88 | 0.88 | 0.88 |
| Gaussian Naive Bayes | 87.61 | 0.75 | 0.65 | 0.57 |
| XG Boost | 89.38 | 0.89 | 0.89 | 0.89 |
| SVM | 65.49 | 0.81 | 0.46 | 0.45 |

These results provide a comparative overview of the performance of each model in predicting insomnia. Based on the metrics, it appears that the Random Forest model achieved the highest accuracy and F1-score among the models evaluated. However, depending on specific requirements and constraints, other models might also be considered suitable choices.

Let's compare the accuracy of all models and determine the one with the highest accuracy:

Logistic Regression: 88.50%

Decision Tree: 86.73%

Random Forest: 91.15%

Gaussian Naive Bayes: 87.61%

Support Vector Machine: 65.49%

XG Boost:89.38%

Based on the accuracy results, the Random Forest model has the highest accuracy of 91.15%. Therefore, the Random Forest model is the best-performing model for predicting insomnia in this scenario.

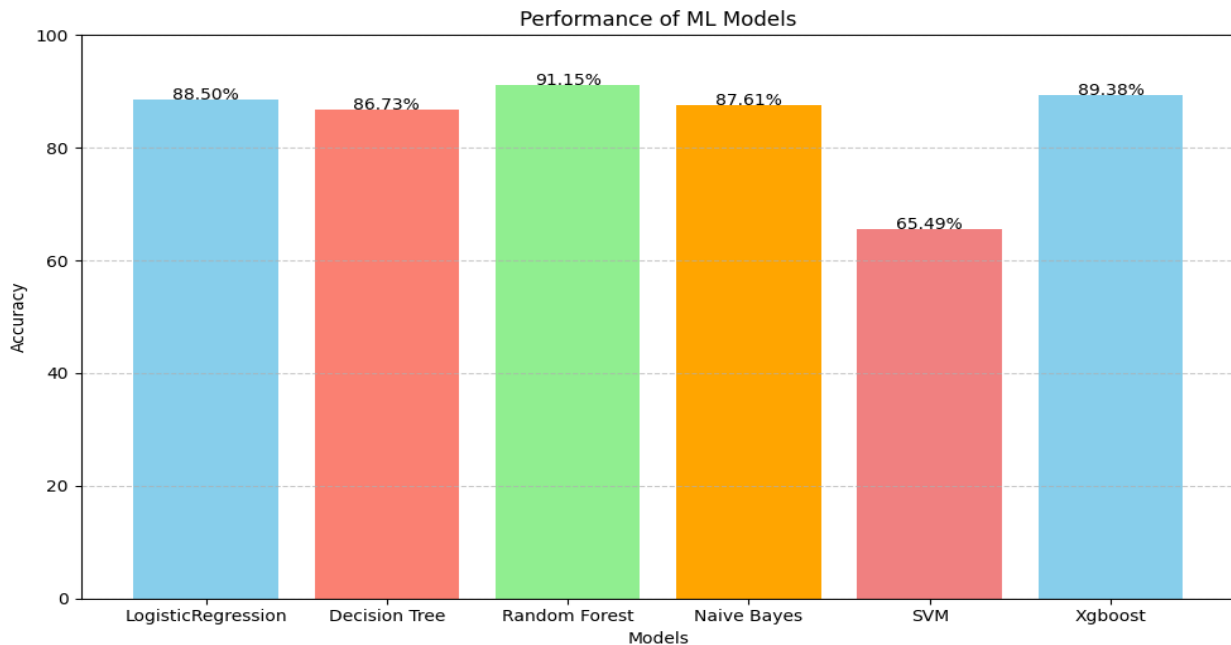


Fig 1: Performance Of All Models

In fig 1..the bar graph showing the performance of different machine learning models for classifying insomnia. The x-axis of the graph shows the names of the models, which include Logistic Regression, Decision Tree, Random Forest, Naive Bayes, SVM, and Xgboost. The y-axis shows the accuracy of each model, as a percentage.

Here are some observations from the graph:

- Random Forest appears to be the most accurate model, with an accuracy of around 91.15%.
- Xgboost and Logistic Regression also have high accuracy, at around 89.38% and 87.61% respectively.
- Support Vector Machine(SVM), Decision Tree, and Naive Bayes appear to have lower accuracy in this comparison, ranging from 65.49% to 86.73%.

It's important to note that the accuracy of a machine learning model can vary depending on the specific dataset used to train it. The models in this graph may perform differently if they were trained on a different dataset of insomnia data.

CHAPTER 8

TESTING

Achieving a Random Forest accuracy of 91.15% is quite impressive! It indicates that your model performs well on the given dataset. Integrating this model into a Flask web application for predicting applications. Testing could be used to identify individuals who are at high risk of developing insomnia based on factors such as sleep patterns, lifestyle habits, and medical history. Early detection allows for preventive interventions to be implemented, such as lifestyle modifications, stress management techniques, or sleep hygiene education.

Make sure that your Random Forest model is properly integrated into your Flask application. This involves setting up routes to handle incoming requests, processing the input data, and returning the prediction results. Design a user-friendly interface for your web application. This could include input fields for the features required by your model (if any), buttons to submit the data for prediction, and a space to display the prediction results.

Test the integration thoroughly to ensure that the Flask application interacts correctly with the Random Forest model. This includes testing various scenarios to cover different use cases and edge cases. Evaluate the performance of your web application under different conditions, such as varying numbers of concurrent users or different sizes of input data. Ensure that the application remains responsive and stable even under heavy loads

Once you're satisfied with the functionality and performance of your web application, conduct user acceptance testing. This involves allowing real users to interact with the application and provide feedback. Use this feedback to make any necessary improvements or optimizations. Deploy your Flask application to a production environment, making it accessible to users. Ensure that the deployment process is smooth and that all dependencies are properly configured.

Continuously monitor the performance and usage of your web application in production. Address any issues or bugs that arise promptly, and consider implementing new features or improvements based on user feedback.

Predicting the likelihood of a patient having a certain condition based on their symptoms or medical history. For example, predicting whether a patient is at risk of sleep disorder disease.

Overall, testing this project involves a combination of technical testing to ensure the correctness and performance of the application, as well as user testing.

CHAPTER 9

CONCLUSION

Based on the evaluation of various machine learning models for predicting insomnia using the provided dataset, several conclusions can be drawn.

Firstly, the Random Forest model achieved the highest accuracy among all tested models, with an accuracy of approximately 91.15%. This indicates that Random Forest performed exceptionally well in classifying individuals based on their sleep health attributes and other relevant features.

Secondly, the Decision Tree model also exhibited commendable performance, with an accuracy of around 86.73%. While slightly lower than Random Forest, Decision Tree still demonstrated robust predictive capabilities for identifying insomnia.

On the other hand, models such as Logistic Regression, Naive Bayes, and Support Vector Machine (SVM) showed relatively lower accuracies, ranging from approximately 65.49% to 88.50%. Although these models performed reasonably well, they were outperformed by Random Forest and Decision Tree.

In conclusion, the Random Forest model stands out as the most effective model for predicting insomnia based on the provided dataset. Its ensemble learning approach, which combines multiple decision trees, likely contributes to its superior performance by mitigating overfitting and capturing complex relationships within the data.

For future enhancements of this project, several avenues can be explored:

Further exploration and refinement of features related to sleep health, lifestyle factors, and demographic information could improve model performance.

Fine-tuning the hyperparameters of machine learning models, especially Random Forest, could potentially enhance their predictive accuracy. Investigating ensemble methods other than Random Forest, such as Gradient Boosting Machines (GBM) or AdaBoost, might lead to even better performance.

Increasing the size and diversity of the dataset through data augmentation techniques could help improve the generalization ability of the models. Integrating additional clinical data or

incorporating wearable device data for real-time monitoring could provide richer insights into sleep health and further improve prediction accuracy.

Overall, this project demonstrates the potential of machine learning in predicting insomnia and highlights opportunities for further research and development in this important area of healthcare.

CHAPTER 10

BIBLIOGRAPHY

- [1] A systematic way of collecting data of insomniac patients: an analytical survey- Md. Muhaiminul Islam, Abu Kaisar Mohammad Masum, Sheikh Abujar, Syed Akhter Hossain, 2020
- [2] "Insomnia and risk of cardiovascular disease: a meta analysis - Francesco Sofi, Francesca Cesari, Alessandro Casini, Claudio Macchi, Rosanna Abbate, Gian Franco Gensini, 2014", SAGE Journals, 2020.
- [3] M. Bar, G. Schrieber, N. Gueron-Sela, G. Shahar and L. Tikotzky, "Role of Self-Criticism, Anxiety, and Depressive Symptoms in Young Adults' Insomnia", International Journal of Cognitive Therapy, vol. 13, no. 1, pp. 15-29, 2020.
- [4] Y. Zhang et al., "Worldwide and regional prevalence rates of co-occurrence of insomnia and insomnia symptoms with obstructive sleep apnea: A systematic review and meta analysis", Sleep Medicine Reviews, vol. 45, pp. 1-17, 2019.
- [5] "Sleep stress level classification through machine learning algorithms" - Vinak Singh, Mahendra kumar Gourisaria, Himansu Das, 2022."
- [6] Hamer M, Batty GD and Kivimaki M. Sleep loss due to worry and future risk of cardiovascular disease and allcause mortality: The Scottish Health Survey. Eur J Cardiovasc Prev Rehabil. Epub ahead of print 3 October 2011.
- [7] Choi SJ, Kim D, Hwang Y, Jo H, Joo EY. Sex differences in etiologies of sleep disorders. J Sleep Med 2020;17(2):138-147
- [8] Lopenon M, Hublin C, Kalimo R, et al. Joint effect of self-reported sleep problems and three components of the metabolic syndrome on risk of coronary heart disease. J Psychosom Res 2010; 68: 149–158.
- [9] Costa LE, Uchôa CHG, Harmon RR, Bortolotto LA, Lorenzi-Filho G, Drager LF. Potential underdiagnosis of obstructive sleep apnoea in the cardiology outpatient setting. Heart 2015;101(16):1288-1292
- [10] Laugsand LE, Vatten LJ, Platou C, et al. Insomnia and risk of acute myocardial infarction. A population study. Circulation 2011; 124: 2073–2081

CHAPTER 11

APPENDIX

11.1 SOURCE CODE OF PREDICT INSOMNIA

```
import pandas as pd
import numpy as np
df=pd.read_csv("/content/sleep_disorder.csv")
```

```
df.columns
```

```
df.duplicated()
```

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

```
df.isnull()
```

```
df.isnull().sum()
```

```
df.describe()
```

```
df.info()
```

```
df.head()
```

```
df.tail()
```

```
df.shape
```

```
df.shape[0]
```

```
df.shape[1]
```

```
df.nunique()
```

```
#check numerical columns with unique values
```

```
numerical_columns=df.select_dtypes(include=['int64','float64']).columns
```

```
print("Numerical Values: ")
```

```
for columns in numerical_columns:
```

```
    unique_values=df[columns].unique()
```

```
    print(f'{columns.upper()}: {unique_values}')
```

```
    print("_____")
```

```
#check object type columns with unique values
```

```
object_columns=df.select_dtypes("object").columns
```

```
for columns in object_columns:
```

```
    unique_values=df[columns].unique()
```

```
    print(f'{columns.upper()}: {unique_values}')
```

```
    print("_____")
```

```
df.columns
```

```
df.value_counts('Occupation')
```

```
df
```

```
from sklearn.preprocessing import LabelEncoder
encoder=LabelEncoder()
df['Gender']=encoder.fit_transform(df['Gender'])
df['BMI Category']=encoder.fit_transform(df['BMI Category'])
df['Occupation']=encoder.fit_transform(df['Occupation'])
df['Blood Pressure']=encoder.fit_transform(df['Blood Pressure'])
df['Sleep Disorder']=encoder.fit_transform(df['Sleep Disorder'])
```

```
df
```

```
df.corr()
```

```
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(8,4))
sns.countplot(x = 'Gender', data = df, palette = 'muted')
plt.title('Male vs Female')
plt.xticks(rotation = 0)
plt.show()
```

```
plt.figure(figsize=(8,4))
sns.countplot(x = 'Occupation', data = df, palette = 'hls')
plt.title('Occupation')
plt.xticks(rotation = 0)
plt.show()
```

```
plt.figure(figsize=(8,4))  
sns.countplot(x = 'Age', data = df, palette = 'deep')  
plt.title('AGE')  
plt.xticks(rotation = 0)  
plt.show()
```

```
plt.figure(figsize=(8,4))  
sns.countplot(x = 'BMI Category', data = df, palette = 'dark')  
plt.title('BMI Category')  
plt.xticks(rotation = 0)  
plt.show()
```

```
plt.figure(figsize=(8,4))  
sns.boxplot(x = 'Gender',y = 'Age', data = df, palette = 'muted')  
plt.title('Age distribution by gender')  
plt.xticks(rotation = 0)  
plt.show()
```

```
from sklearn.model_selection import train_test_split  
from sklearn import metrics  
from sklearn.metrics import confusion_matrix  
from sklearn.linear_model import LogisticRegression  
from sklearn.svm import SVC  
from sklearn.linear_model import LinearRegression  
from sklearn.metrics import confusion_matrix,accuracy_score  
from sklearn.tree import DecisionTreeClassifier  
from sklearn.metrics import classification_report
```

```
import scipy.stats as ss

from scipy.stats import chi2_contingency

import matplotlib.pyplot as plt

x = df.iloc[:,0:12]
y = df.loc[:, "Sleep Disorder"]

x_train, x_test, y_train, y_test = train_test_split(x,y, test_size = 0.3, random_state=0)

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

# Generating synthetic data for demonstration
np.random.seed(0)
x = np.random.randn(100, 2) # 100 samples, 2 features
y = np.random.randint(0, 2, 100) # Binary labels

# Splitting the data into training and testing sets
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=0)

# Plotting the training and testing data points
plt.figure(figsize=(10, 6))
plt.scatter(x_train[:, 0], x_train[:, 1], c='b', label='Training Set')
plt.scatter(x_test[:, 0], x_test[:, 1], c='r', label='Testing Set')
plt.title('Visualization of Training and Testing Data')
plt.xlabel('Feature 1')
```

```
plt.ylabel('Feature 2')
plt.legend()
plt.grid(True)
plt.show()

# LogisticRegression
LR = LogisticRegression()
LR.fit(x_train, y_train)
pred_test = LR.predict(x_test)
conf = confusion_matrix(y_test, pred_test)
print ("Confusion Matrix : \n", conf)
print(classification_report(y_test,pred_test))

print("The accuracy of Logistic Regression model is : ",100*accuracy_score(y_test,
pred_test))

import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import classification_report

# Confusion matrix values
conf_matrix_values = [[16, 3, 4],
                      [2, 19, 3],
                      [1, 0, 65]]

# Labels for confusion matrix
class_names = ['Class 0', 'Class 1', 'Class 2']
```

```
# Create heatmap using seaborn

plt.figure(figsize=(8, 6))

sns.heatmap(conf_matrix_values, annot=True, cmap='Blues', fmt='g',
xticklabels=class_names, yticklabels=class_names)

# Add labels and title

plt.xlabel('Predicted')

plt.ylabel('True')

plt.title('Confusion Matrix')

# Show plot

plt.show()

# DecisionTree Classifier

DT = DecisionTreeClassifier()

DT.fit(x_train, y_train)

predDT = DT.predict(x_test)

print("Confusion Matrix : \n", confusion_matrix(y_test, predDT))

print(classification_report(y_test, predDT))

print("the accuracy of Decision Tree Model is:", 100*accuracy_score(y_test, predDT))

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.metrics import classification_report

# Confusion matrix values

conf_matrix_values = [[19, 1, 3],
```

```
[2, 19, 3],
[5, 1, 60]]

# Labels for confusion matrix
class_names = ['Class 0', 'Class 1', 'Class 2']

# Create heatmap using seaborn
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix_values, annot=True, cmap='Blues', fmt='g',
xticklabels=class_names, yticklabels=class_names)

# Add labels and title
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix')

# Show plot
plt.show()

# RandomForest Classifier
from sklearn.ensemble import RandomForestClassifier
RF = RandomForestClassifier()
RF.fit(x_train, y_train)
predRF = RF.predict(x_test)
print("Confusion Matrix :\n", confusion_matrix(y_test, predRF))
print(classification_report(y_test, predRF))
print("The accuracy of Random Forest is :", 100*accuracy_score(y_test, predRF))
```



```
import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.metrics import classification_report, confusion_matrix


# Confusion matrix values
conf_matrix_values = [[19, 2, 2],
                       [1, 20, 3],
                       [1, 1, 64]]


# Labels for confusion matrix
class_names = ['Class 0', 'Class 1', 'Class 2']


# Create heatmap using seaborn
plt.figure(figsize=(8, 6))

sns.heatmap(conf_matrix_values, annot=True, cmap='Blues', fmt='g',
            xticklabels=class_names, yticklabels=class_names)


# Add labels and title
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix')


# Show plot
plt.show()


# NavieBayes
```

```
from sklearn.naive_bayes import GaussianNB

NB = GaussianNB()

NB.fit(x_train, y_train)

predNB = NB.predict(x_test)

print("Classification matrix :\n", confusion_matrix(y_test, predNB))

print(classification_report(y_test, predNB))

print("The accuracy of GaussianNB is :", 100*accuracy_score(y_test, predNB))


import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.metrics import classification_report, confusion_matrix


# Classification matrix values

conf_matrix_values = [[19, 2, 2],
                      [1, 20, 3],
                      [6, 0, 60]]


# Labels for confusion matrix

class_names = ['Class 0', 'Class 1', 'Class 2']


# Create heatmap using seaborn

plt.figure(figsize=(8, 6))

sns.heatmap(conf_matrix_values, annot=True, cmap='Blues', fmt='g',
            xticklabels=class_names, yticklabels=class_names)


# Add labels and title

plt.xlabel('Predicted')
```

```
plt.ylabel('True')
plt.title('Confusion Matrix')

# Show plot
plt.show()

# SVM
Model = SVC()
Model.fit(x_train, y_train)
predSVM = Model.predict(x_test)
print("Confusion Matrix :\n", confusion_matrix(y_test, predSVM))
print(classification_report(y_test, predSVM))
print("The accuracy of the Support vector machine is
:",100*accuracy_score(y_test,predSVM))

import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix, classification_report

# Confusion matrix values
conf_matrix_values = [[1, 0, 22],
                      [0, 9, 15],
                      [0, 2, 64]]

# Labels for confusion matrix
class_names = ['Class 0', 'Class 1', 'Class 2']
```

```
# Create heatmap using seaborn

plt.figure(figsize=(8, 6))

sns.heatmap(conf_matrix_values,          annot=True,          cmap='Blues',          fmt='g',
xticklabels=class_names, yticklabels=class_names)


# Add labels and title

plt.xlabel('Predicted')

plt.ylabel('True')

plt.title('Confusion Matrix')


# Show plot

plt.show()


import xgboost

xgb_model = xgboost.XGBClassifier()

xgb_model.fit(x_train, y_train)

pred_testXg=xgb_model.predict(x_test)

accuracy=accuracy_score(y_test,pred_testXg)

print('Accuracy : ',100*accuracy)


print("Confusion Matrix: ",confusion_matrix(y_test,pred_testXg))

print("Classification report: ",classification_report(y_test,pred_testXg))


import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.metrics import confusion_matrix
```

```
# Confusion matrix values
conf_matrix_values = [[19, 2, 2],
                      [1, 20, 3],
                      [1, 3, 62]]

# Labels for confusion matrix
class_names = ['Class 0', 'Class 1', 'Class 2']

# Create heatmap using seaborn
plt.figure(figsize=(8, 6))

sns.heatmap(conf_matrix_values, annot=True, cmap='Blues', fmt='g',
            xticklabels=class_names, yticklabels=class_names)

# Add labels and title
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix')

# Show plot
plt.show()

l=[100*accuracy_score(y_test, pred_test),100*accuracy_score(y_test,
predDT),100*accuracy_score(y_test, predRF),100*accuracy_score(y_test,
predNB),100*accuracy_score(y_test, predSVM),100*accuracy_score(y_test, pred_testXg)]

model_names=['LogisticRegression','Decision Tree','Random Forest','Naive
Bayes','SVM','Xgboost']

colors = ['skyblue', 'salmon', 'lightgreen', 'orange', 'lightcoral']
```

```
plt.figure(figsize=(10, 6))
plt.bar(model_names, l, color=colors)
plt.xlabel('Models')
plt.ylabel('Accuracy')
plt.title('Performance of ML Models')
plt.ylim(0, 100) # Setting y-axis limit to 0-1 for accuracy score
plt.grid(axis='y', linestyle='--', alpha=0.7) # Adding grid lines
plt.xticks(rotation=0) # Rotate x-axis labels for better readability

# Adding the accuracy scores above each bar
for i in range(len(model_names)):
    plt.text(i, l[i] + 0.01, f'{l[i]:.2f}%', ha='center')

# Showing the plot
plt.tight_layout()
plt.show()

#Deployment Flask Code
from flask import Flask, render_template, request
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder
from imblearn.over_sampling import SMOTE
from sklearn.model_selection import train_test_split
from lazypredict.Supervised import LazyClassifier
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier
```

```
app=Flask(__name__)

data=pd.read_csv("sleep_disorder.csv")

print(data.columns)

print(data.isna().sum())

print(data.info())


lab=LabelEncoder()

for i in data.select_dtypes(include='object').columns.values:

    data[i]=lab.fit_transform(data[i])


x=[]

corr=data.corr()['Sleep Disorder']

corr=corr.drop(['Sleep Disorder'])

for i in corr.index:

    if corr[i]>0:

        x.append(i)

x=data[x]

y=data['Sleep Disorder']

smote=SMOTE()

x,y=smote.fit_resample(x,y)


x_train,x_test,y_train,y_test=(train_test_split(x,y))


@app.route('/')

def home():

    return render_template('index.html')
```

```
# Define a route for prediction

@app.route('/predict', methods=['POST'])
def predict():
    # Extract data from the form
    gender = float(request.form['gender'])
    sleep_duration = float(request.form['sleep_duration'])
    quality_of_sleep = float(request.form['quality_of_sleep'])
    physical_activity_level = float(request.form['physical_activity_level'])
    daily_steps = float(request.form['daily_steps'])

    input_data = np.array([[gender, sleep_duration, quality_of_sleep,
                             physical_activity_level, daily_steps]])

    rf=RandomForestClassifier()
    rf.fit(x_train,y_train)
    prediction = rf.predict(input_data)

    return render_template('result.html', prediction=prediction[0])

if __name__ == '__main__':
    app.run(debug=True)

#index.html
<!DOCTYPE html>
```



```
<html>
<head>
  <title>Sleep Disorder Prediction</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f0f0f0;
      margin: 0;
      padding: 0;
      text-align: center; /* Center align text */
    }

    h1 {
      color: #333;
    }

    form {
      width: 50%;
      margin: 20px auto;
      padding: 20px;
      background-color: #fff;
      border-radius: 8px;
      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    }

    label {
      display: block;
```

```
margin-bottom: 5px;  
color: #333;  
}
```

```
input[type="text"] {  
  width: calc(100% - 20px); /* Adjusted for padding */  
  padding: 8px;  
  margin-bottom: 10px;  
  border: 1px solid #ccc;  
  border-radius: 5px;  
  box-sizing: border-box;  
}
```

```
input[type="submit"] {  
  width: calc(100% - 20px); /* Adjusted for padding */  
  padding: 10px;  
  background-color: #007bff;  
  color: #fff;  
  border: none;  
  border-radius: 5px;  
  cursor: pointer;  
  transition: background-color 0.3s ease;  
}
```

```
input[type="submit"]:hover {  
  background-color: #0056b3;  
}
```

```
</style>
</head>
<body>
  <h1> Sleep Disorder Prediction</h1>
  <form action="/predict" method="post">

    <label for="gender">Gender:</label><br>
    <input type="text" id="gender" name="gender"><br>

    <label for="sleep_duration">Sleep Duration:</label><br>
    <input type="text" id="sleep_duration" name="sleep_duration"><br>

    <label for="quality_of_sleep">Quality of Sleep (1-10):</label><br>
    <input type="text" id="quality_of_sleep" name="quality_of_sleep"><br>

    <label for="physical_activity_level">Physical Activity Level (1-10):</label><br>
    <input type="text" id="physical_activity_level" name="physical_activity_level"><br>

    <label for="daily_steps">Daily Steps:</label><br>
    <input type="text" id="daily_steps" name="daily_steps"><br>

    <input type="submit" value="Predict">
  </form>
</body>
</html>
#result.html
<!DOCTYPE html>
```

```
<html>
<head>
  <title>Prediction Result</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f0f0f0;
      margin: 0;
      padding: 0;
    }

    h2 {
      color: #333;
    }

    p {
      color: #555;
      font-size: 18px;
    }
  </style>
</head>
<body>
  <h2>Prediction Result</h2>
  <p>The predicted sleep disorder class is: <span id="prediction">{{ prediction }}</span></p>
</body>
</html>
```