PREDICTIVE ANALYTICS PROJECT REPORT

(Project Semester August-December 2023)

Predicting Stress Levels Using Machine Learning Algorithms

Submitted by

Routhu Siddhartha

Registration No: - 12010599

Programme and Section: - CSE – K20SH

Course Code: - INT234

Under the Guidance of

Ms. Maneet Kaur (15709)

Discipline of CSE/IT

Lovely School of Computer Science & Engineering

Lovely Professional University, Phagwara

CERTIFICATE

This is to certify that Routhu Siddhartha bearing Registration no. 12010599 has completed INT234 project titled, **“Predicting Stress Levels Using Machine Learning Algorithms”** under my guidance and supervision. To the best of my knowledge, the present work is the result of his/her original development, effort and study.

Signature and Name of the Supervisor
Designation of the Supervisor
School of Computer Science And Engineering
Lovely Professional University
Phagwara, Punjab.

Date: 6th November 2023

DECLARATION

I, Routhu Siddhartha, student of Bachelor of Technology under CSE/IT Discipline at, Lovely Professional University, Punjab, hereby declare that all the information furnished in this project report is based on my own intensive work and is genuine.

Date: 06-11-2023 Signature

Registration No.: -12010599 Name:-Routhu Siddhartha

Acknowledgement

Place: Phagwara

Date:06-11-2023

I would like to sincerely thank my manager Ms. Maneet Kaur for her significant role behind the accomplishment of the assignment. I have been guided with lots of her valuable suggestions and experience throughout the process of completion of the assignment.

I would also like to express my gratitude to my peers, without their support and cooperation this assignment could not have been accomplished. Finally, I would like to thank my parents for their love and blessings.

Your name: - Routhu Siddhartha

Signature: -

CONTENT:-

1 Cover Page

2 Certificate

3 Declaration

4 Acknowledgement

5 Table of content

6 Table of content

7 Introduction & Benefits of Prediction & Analysis

8 [Continue..]

9 Objectives/Scope of the Analysis- Data Exploring & Data Preprocessing

10 Handling Missing Values, Features Scaling, Encoding Categorical Data, Data Transformation, Outlier Detection and Handling,

11 Splitting the Dataset, Source of Dataset

12 About Dataset

13 Data Splitting, Model Selection and Evaluation

14 ETL Process

15 Use of ETL Process, Extraction Process and Analysis of dataset

16 [Continue..]

17 [Continue..]

18 [Continue..]

19 List of Analysis with Result, Libraries Used

20 K-Nearest Neighbours Model

21 [Continue..]

22 Naïve Bayes Model

23 [Continue..]

24 Decision Tree Model

25 Random Forest Model

26 [Continue..]

27 Support Vector Machine Model

28 Confusion Matrix – CM for Model Evaluation

29 Evaluation Model Performance

30 Fine-Tuning and Optimization, Accuracy Received

31 Interpretation of Model Result

32 Conclusion

33 Bibliography

➢ INTRODUCTION: -

In a world characterized by fast-paced lifestyles and increasing demands, stress has become a prevalent concern affecting individuals, healthcare professionals, and organizations alike. "Predicting Stress Levels Using Machine Learning Algorithms" is a project aimed at addressing this pressing issue by harnessing the capabilities of data science. This report delineates the project's objectives, scope, data sources, and data preprocessing methods, laying the foundation for a comprehensive analysis.

The primary goal of this project is to employ machine learning algorithms to predict stress levels with accuracy. To achieve this, we employ a variety of machine learning techniques, including k-Nearest Neighbours (k-NN), Naive Bayes, Decision Trees, Random Forest, and Support Vector Machines (SVM). By doing so, we aim not only to create precise stress level prediction models but also to gain profound insights into the factors influencing stress, shedding light on its intricacies and dynamics.

This project encompasses the entire data science pipeline, commencing with data collection from carefully selected sources and followed by rigorous data preprocessing, which includes data cleansing, missing value handling, and feature scaling. Armed with a meticulously prepared dataset, we delve into model selection, training, and evaluation. In this report, we present a comprehensive analysis of data preprocessing methods, detailed descriptions of the machine learning algorithms employed, and the outcomes of our analyses. The subsequent sections will provide in-depth insights into the methods used, results obtained, and visualizations created for each analysis, culminating in a list of analyses and results. The report concludes with references and a bibliography to attribute sources and resources employed throughout the project.

BENEFITS OF PREDICTION & ANALYSIS: -

Predictive models provide a wide array of advantages, including data-driven decision-making, risk assessment, and resource optimization. They assist in early issue detection, enabling proactive interventions and cost reduction. Additionally, these models contribute to trend analysis and inform better strategies. Moreover, predictive models aid in model creation, offering valuable insights to enhance the accuracy and effectiveness of predictive algorithms.

In summary, they play a pivotal role in improving decision-making, efficiency, and the model development process across various domains.

The objectives and scope of this analysis can be summarized as follows:

- To predict stress levels in individuals using machine learning algorithms.
- To assess and compare the performance of various machine learning algorithms in stress level prediction.
- To gain insights into the factors influencing stress levels and their dynamics.
- To identify the strengths and limitations of different machine learning techniques for this specific task.
- To utilize a curated dataset for stress level prediction.
- To encompass data collection, preprocessing, model selection, training, and evaluation within the project scope.
- To focus on k-Nearest Neighbors (k-NN), Naive Bayes, Decision Trees, Random Forest, and Support Vector Machines (SVM) for the analysis.
- To fine-tune algorithm hyperparameters to optimize predictive accuracy.
- To critically evaluate the results, including model performance and limitations.
- To provide a comprehensive report including data preprocessing, detailed descriptions of machine learning algorithms, analysis results, and visualizations, culminating in a list of analyses with their respective outcomes.

In this chapter, you will get to know the most important Excel features that come handy when you are creating a dashboard. These features help you arrive at the dashboard elements that simplify complex data and provide visual impact on the current status or performance in real time

- **Objectives/Scope of the Analysis: -**

Data Exploration

Data exploration is a critical phase in the development of our stress level prediction model. This phase involves delving deep into the dataset to gain insights and a comprehensive understanding of its contents. The primary objective is to uncover patterns, relationships, and anomalies that can be invaluable in the subsequent modelling and prediction stages.

In this phase, we perform a variety of operations, including descriptive statistics, data visualization, and correlation analysis. Descriptive statistics provide key summary metrics that offer an initial glimpse into the dataset's central tendencies, dispersions, and distributions. Data visualization plays a pivotal role in making these insights more intuitive, employing various charts, graphs, and plots to visualize data distribution, trends, and relationships. Correlation analysis helps identify relationships between different attributes, shedding light on which features may have a significant impact on stress levels.

Exploring the data is not only about understanding the dataset but also about identifying which features may have the most influence on our predictive models. This process allows us to select and engineer the most relevant features, which are critical for building models that can make accurate predictions. Data exploration is a dynamic and iterative process, as initial findings may prompt further data preprocessing or the creation of new features. This phase provides the foundation for subsequent model development and the creation of valuable predictive insights.

Data Preprocessing

- **Introduction**

Data preprocessing is a pivotal phase in any data science project, including our stress level prediction model. This process involves preparing and cleaning the raw data to make it suitable for analysis and model development. The aim is to enhance the quality and relevance of the dataset, ultimately leading to more accurate predictions.

- **Handling Missing Values**

  One of the primary challenges in data preprocessing is addressing missing values. In our dataset, missing values can disrupt the analysis and lead to erroneous results. We employ techniques such as imputation, where missing values are filled in using appropriate methods, and data removal, where instances with substantial missing data are eliminated.

- **Feature Scaling**

  Feature scaling is crucial to ensure that features with different scales do not impact the predictive model disproportionately. In our case, where data can range from various scales, standardization or normalization methods are applied to bring all features to a common scale, preventing any one feature from dominating the model.

- **Encoding Categorical Data**

  Our dataset may contain categorical variables, such as "gender" or "education level." Machine learning models require numerical data, so encoding techniques are used to convert categorical data into numerical form. Methods such as one-hot encoding are applied to ensure the model can work with this data effectively.

- **Data Transformation**

  Data transformation techniques are used to create new features or modify existing ones. In our stress level prediction project, we may apply techniques like feature engineering, where we create new features based on existing ones, or data reduction, which involves dimensionality reduction to improve model efficiency.

- **Outlier Detection and Handling**

  Outliers can significantly affect the performance of predictive models. Robust methods like Z-score or IQR are utilized to detect and handle outliers in our dataset. This helps maintain model accuracy and reliability.

- **Splitting the Dataset**

Before training and evaluating our predictive models, we must split the dataset into training and testing sets. This process ensures that the model's performance can be assessed on data it has never seen before, providing a realistic evaluation of its predictive capabilities.

Data preprocessing is the cornerstone of successful predictive modelling. It helps us prepare the dataset, ensuring it is clean, complete, and properly formatted for model training. By addressing missing values, scaling features, encoding categorical data, transforming features, handling outliers, and splitting the dataset, we create a solid foundation for building and evaluating accurate stress level prediction models.

➢ Source of Data

A dataset is a collection of data within a database.

Typically, datasets take on a tabular format consisting of rows and columns. Each column represents a specific variable, while each row corresponds to a specific value. Some datasets consisting of unstructured data are non-tabular, meaning they don’t fit the traditional row-column format.

The source of data holds a paramount position in shaping the trajectory of our stress level prediction project. It stands as the primary and most pivotal stage in identifying the problem statement and subsequently determining the project's direction. The dataset serves as the fundamental element upon which our entire analysis and model development rely. The selection of a relevant dataset is a pivotal decision that profoundly influences the project's course. In our diligent research, we explored various platforms to find the dataset most aligned with our problem statement. Kaggle, a renowned hub for data scientists and researchers, emerged as the optimal source for our dataset requirements.

From Kaggle, we identified the dataset titled "Student Stress Factors: A Comprehensive Analysis - Understanding the Underlying Causes and Their Impact on Today's Students." This dataset presents a comprehensive analysis of the factors contributing to student stress, offering a wealth of relevant data for our project. It forms the cornerstone upon which we build our stress level prediction models and explore the complexities of stressors. With Kaggle's dataset, we have a robust resource to develop accurate predictive models, thereby addressing the pressing concern of stress effectively.

Dataset link : Student Stress Factors: A Comprehensive Analysis | Kaggle

In summary, the source of data is pivotal in our project's journey, and Kaggle, with its wealth of relevant datasets, has played a central role in advancing our stress level prediction project. The dataset's comprehensive analysis of student stress factors provides valuable insights, empowering us to uncover patterns, identify stressors, and ultimately contribute to better stress management. The dataset's role as the foundation for our analysis underscores its significance in our endeavour to predict stress levels accurately and address this critical issue.

➢ About dataset

Unlock the secrets of student stress with our easy-to-understand dataset! Dive into real-life factors like sleep quality, study load, and even bullying. Discover how the environment or even friendships can impact stress. Perfect for beginners eager to explore and make a difference. Start your data journey with us and uncover stories that matter!

This dataset contains around 20 features that create the most impact on the Stress of a Student. The features are selected scientifically considering 5 major factors, they are Psychological, Physiological, Social, Environmental, and Academic Factors. Some of them are:
Psychological Factors => 'anxiety_level', 'self_esteem', 'mental_health_history', 'depression',
Physiological Factors => 'headache', 'blood_pressure', 'sleep_quality', 'breathing_problem
Environmental Factors => 'noise_level', 'living_conditions', 'safety', 'basic_needs',
Academic Factors => 'academic_performance', 'study_load', 'teacher_student_relationship', 'future_career_concerns',
Social Factor => 'social_support', 'peer_pressure', 'extracurricular_activities', 'bullying'

➢ Data Splitting

Data splitting is a crucial step in the preparation of our dataset for predictive modeling. In this phase, we divide the dataset into distinct subsets, primarily the training and testing sets. The training set is employed to train our predictive models, while the testing set serves as an independent dataset to evaluate the model's performance. The key objective of data splitting is to ensure that our models can generalize well to unseen data, thereby providing reliable predictions. By separating the data, we emulate real-world scenarios, where the model encounters new instances, it has never seen before, and assess its accuracy in predicting stress levels accurately. This process helps us avoid overfitting, where the model performs well on the training data but fails to generalize effectively.

In our stress level prediction project, we typically employ a common split ratio, often 70% for the training set and 30% for the testing set, but this can vary based on project requirements. The training set forms the foundation for model training, allowing the algorithm to learn patterns and relationships within the data. Subsequently, the testing set remains unseen during model development, serving as the benchmark for evaluating the model's predictive accuracy. By maintaining this separation, we achieve a comprehensive and reliable assessment of our model's performance. This data splitting process is vital to ensuring the accuracy and effectiveness of our predictive models, ultimately contributing to the successful management of stress levels.

- **Model Selection and Evaluation**

Model selection and evaluation are pivotal phases in our stress level prediction project. In these stages, we explore a variety of machine learning algorithms, such as k-Nearest Neighbours (k-NN), Naive Bayes, Decision Trees, Random Forest, and Support Vector Machines (SVM), to identify the most suitable predictive model. Each algorithm brings its unique approach to stress level prediction, and model selection involves comparing their performance and choosing the one that offers the highest predictive accuracy. Evaluation ensures that the selected model meets the project's objectives and generalizes effectively to unseen data. We utilize various evaluation metrics, such as accuracy, precision, recall, F1-score, and ROC-AUC, to assess the model's performance comprehensively.

To determine the best predictive model for our project, we undertake rigorous experimentation, fine-tuning model hyperparameters, and assessing their performance on the testing dataset. Model evaluation involves scrutinizing key metrics and conducting cross-validation to ensure robustness and minimize bias. Our objective is to select a model that not only predicts stress levels accurately but also offers valuable insights into the factors contributing to stress. Model selection and evaluation are iterative processes, where we continuously refine our models and aim for the highest level of predictive accuracy. The chosen model will play a pivotal role in our endeavour to predict and manage stress levels effectively, benefiting individuals and organizations alike.

- **ETL Process**

ETL, which stands for extract, transform and load, is a data integration process that combines data from multiple data sources into a single, consistent data store that is loaded into a data warehouse or other target system.

As the databases grew in popularity in the 1970s, ETL was introduced as a process for integrating and loading data for computation and analysis, eventually becoming the primary method to process data for data warehousing projects.

ETL provides the foundation for data analytics and machine learning workstreams. Through a series of business rules, ETL cleanses and organizes data in a way which addresses specific business intelligence needs, like monthly reporting, but it can also tackle more advanced analytics, which can improve back-end processes or end user experiences. ETL is often used by an organization to:

- Extract data from legacy systems
- Cleanse the data to improve data quality and establish consistency
- Load data into a target database

The data is loaded in the DW system in the form of dimension and fact tables.

ETL combines all the three-database function into one tool to fetch data from one database and place it into another database.

Use Of ETL Process

ETL is used to integrate the data with the help of three steps Extract, Transform, and Load, and it is used to blend the data from multiple sources. It is often used to build a data warehouse.

In the ETL process, data is extracted from the source system and convert into a format that can be examined and stored into a data warehouse or any other system.

Extraction Process: -

- Extract is the process of fetching (reading) the information from the database. At this stage, data is collected from multiple or different types of sources.
- A staging area is required during ETL load. There are various reasons why staging area is required.
- The source systems are only available for specific period of time to extract data. This period of time is less than the total data-load time. Therefore, staging area allows you to extract the data from the source system and keeps it in the staging area before the time slot ends.

- **Analysis on Dataset**

A dataset is a collection of data within a database.

Typically, datasets take on a tabular format consisting of rows and columns. Each column represents a specific variable, while each row corresponds to a specific value. Some datasets consisting of unstructured data are non-tabular, meaning they don’t fit the traditional row-column format.

As I have taken the dataset of real time data from Kaggle, it contains columns of analysis by labelling the contents as follows.

- anxiety_level
- self_esteem
- mental_health_history

- depression
- headache
- blood_pressure
- sleep_quality
- breathing_problem
- noise_level
- living_conditions
- safety
- basic_needs
- academic_performance
- study_load
- teacher_student_relationship
- future_career_concerns
- social_support
- peer_pressure
- extracurricular_activities
- bullying

- stress_level

anxiety_level: This column assesses the degree of anxiety experienced by individuals, contributing to a comprehensive understanding of emotional well-being.

self_esteem: Self-esteem measures an individual's self-worth and self-assessment, offering insights into their self-perception and confidence.

mental_health_history: This attribute provides information about an individual's mental health history, a crucial factor in understanding predispositions to stress.

depression: The depression column evaluates the presence and severity of depressive symptoms, an important indicator of stress levels.

headache: This column records the occurrence and severity of headaches, which can be a physical manifestation of stress.

blood_pressure: It monitors individuals' blood pressure, offering insights into the physiological effects of stress.

sleep_quality: Sleep quality measures the effectiveness of an individual's sleep patterns, which is closely linked to stress management.

breathing_problem: This column indicates the presence and intensity of breathing problems, often associated with stress-induced symptoms.

noise_level: Noise levels assess an individual's exposure to environmental factors, which can contribute to stress levels.

living_conditions: Evaluating living conditions provides an understanding of the environment's impact on an individual's stress.

safety: Safety measures the perception of safety in an individual's surroundings, influencing their stress perception.

basic_needs: Basic needs evaluate the fulfillment of necessities, contributing to a holistic assessment of stress factors.

academic_performance: This column tracks academic achievements, an area that can both cause and be affected by stress.

study_load: Study load assesses the intensity of academic work, a significant contributor to student stress.

teacher_student_relationship: This attribute gauges the quality of interactions between teachers and students, a pivotal factor influencing stress levels.

future_career_concerns: It records concerns about future career prospects, a source of stress for many individuals.

social_support: Social support measures the strength of an individual's support network, crucial in stress coping.

peer_pressure: This column evaluates the influence of peer pressure on an individual's decision-making, often contributing to stress.

extracurricular_activities: It assesses the participation in extracurricular activities, which can impact stress levels positively or negatively.

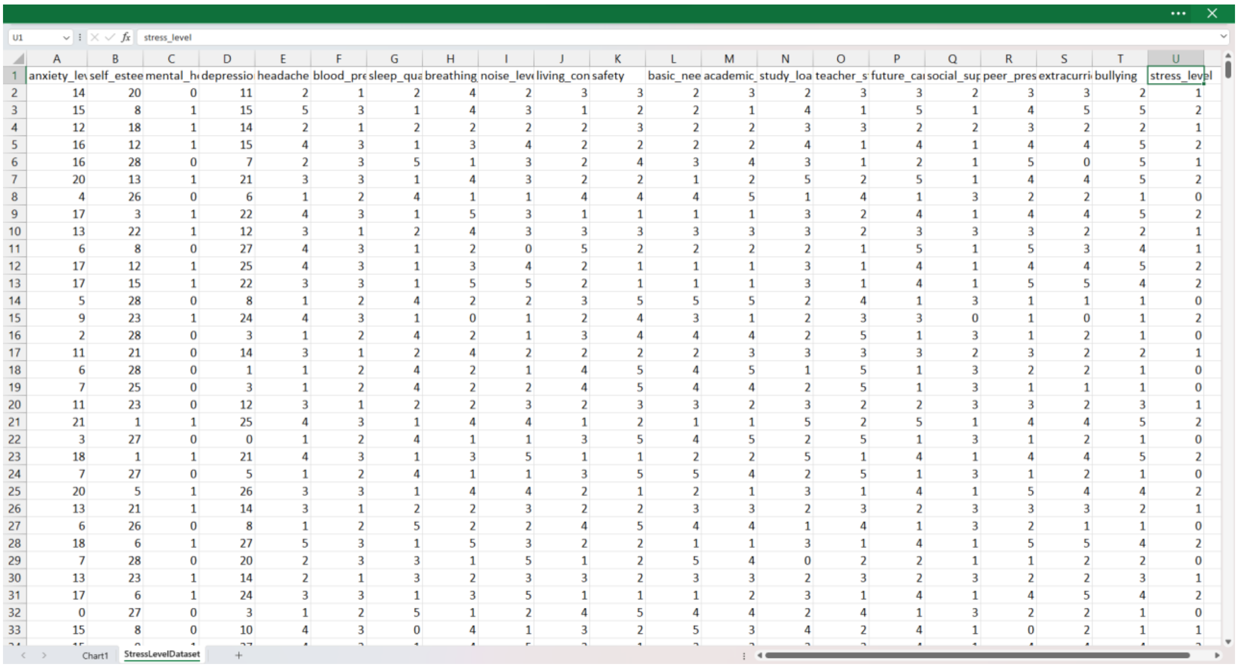
Bullying: Bullying records incidents and experiences of bullying, a significant stress factor among students**.**

stress_level: Stress level is the target variable, indicating the stress experienced by individuals, which we aim to predict accurately in our project.

- **LIST OF ANALYSIS WITH RESULT: -**

1. **Dataset**

StressLevelDataset.csv



U1 | stress_level

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	anxiety_lev	self_estee	mental_h	depressio	headache	blood_pre	sleep_qua	breathing	noise_leve	living_con	safety	basic_nee	academic_	study_loa	teacher_s	future_ca	social_sup	peer_pres	extracurri	bullying	stress_level
2	14	20	0	11	2	1	2	4	2	3	3	2	3	2	3	3	2	3	3	2	1
3	15	8	1	15	5	3	1	4	3	1	2	2	1	4	1	5	1	4	5	5	2
4	12	18	1	14	2	1	2	2	2	2	3	2	2	3	3	2	2	3	2	2	1
5	16	12	1	15	4	3	1	3	4	2	2	2	2	4	1	4	1	4	4	5	2
6	16	28	0	7	2	3	5	1	3	2	4	3	4	3	1	2	1	5	0	5	1
7	20	13	1	21	3	3	1	4	3	2	2	1	2	5	2	5	1	4	4	5	2
8	4	26	0	6	1	2	4	1	1	4	4	4	5	1	4	1	3	2	2	1	0
9	17	3	1	22	4	3	1	5	3	1	1	1	1	3	2	4	1	4	4	5	2
10	13	22	1	12	3	1	2	4	3	3	3	3	3	3	2	3	3	3	2	2	1
11	6	8	0	27	4	3	1	2	0	5	2	2	2	2	1	5	1	5	3	4	1
12	17	12	1	25	4	3	1	3	4	2	1	1	1	3	1	4	1	4	4	5	2
13	17	15	1	22	3	3	1	5	5	2	1	1	1	3	1	4	1	5	5	4	2
14	5	28	0	8	1	2	4	2	2	3	5	5	5	2	4	1	3	1	1	1	0
15	9	23	1	24	4	3	1	0	1	2	4	3	1	2	3	3	0	1	0	1	2
16	2	28	0	3	1	2	4	2	1	3	4	4	4	2	5	1	3	1	2	1	0
17	11	21	0	14	3	1	2	4	2	2	2	2	3	3	3	3	2	3	2	2	1
18	6	28	0	1	1	2	4	2	1	4	5	4	5	1	5	1	3	2	2	1	0
19	7	25	0	3	1	2	4	2	2	4	5	4	4	2	5	1	3	1	1	1	0
20	11	23	0	12	3	1	2	2	3	2	3	3	2	3	2	2	3	3	2	3	1
21	21	1	1	25	4	3	1	4	4	1	2	1	1	5	2	5	1	4	4	5	2
22	3	27	0	0	1	2	4	1	1	3	5	4	5	2	5	1	3	1	2	1	0
23	18	1	1	21	4	3	1	3	5	1	1	2	2	5	1	4	1	4	4	5	2
24	7	27	0	5	1	2	4	1	1	3	5	5	4	2	5	1	3	1	2	1	0
25	20	5	1	26	3	3	1	4	4	2	1	2	1	3	1	4	1	5	4	4	2
26	13	21	1	14	3	1	2	2	3	2	2	3	3	2	3	2	3	3	3	2	1
27	6	26	0	8	1	2	5	2	2	4	5	4	4	1	4	1	3	2	1	1	0
28	18	6	1	27	5	3	1	5	3	2	2	1	1	3	1	4	1	5	5	4	2
29	7	28	0	20	2	3	3	1	5	1	2	5	4	0	2	2	1	1	2	2	0
30	13	23	1	14	2	1	3	2	3	3	2	3	3	2	3	2	3	2	2	3	1
31	17	6	1	24	3	3	1	3	5	1	1	1	2	3	1	4	1	4	5	4	2
32	0	27	0	3	1	2	5	1	2	4	5	4	4	2	4	1	3	2	2	1	0
33	15	8	0	10	4	3	0	4	1	3	2	5	3	4	2	4	1	0	2	1	1

The Data continuous with information of real time data. As it is good to take big dataset for better visualization.

Libraries Used:

- **Class Library:** The "class" library is an essential component of our stress level prediction project, enabling the implementation of the k-Nearest Neighbours (k-NN) algorithm. This library empowers us to build and evaluate our k-NN model, a crucial machine learning technique for predicting stress levels. Through the "class" library, we can determine the stress levels of individuals based on their proximity to their neighbours in the training dataset.

- **Caret Library:** The "caret" library serves as a vital tool for streamlining the process of model evaluation and performance assessment. In our project, it plays a key role in generating confusion matrices to assess the predictive accuracy of our models. This library allows us to comprehensively evaluate the effectiveness of our machine learning algorithms, ensuring that the predictions are accurate and reliable.

- **E1071 Library:** The "e1071" library is a valuable resource for implementing the Naive Bayes and Support Vector Machine (SVM) algorithms in our stress level prediction project. It facilitates the creation and fine-tuning of these models, enhancing their accuracy and robustness. The "e1071" library empowers us to harness the potential of these algorithms to accurately predict stress levels and manage them effectively.

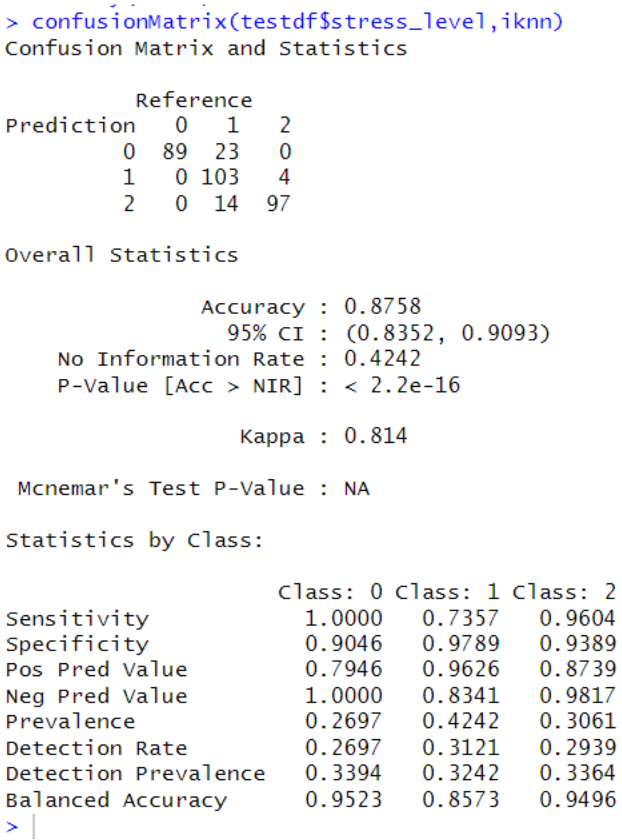
- **Rpart Library:** The "rpart" library is instrumental in implementing the Decision Tree algorithm, a critical component of our project. This library enables us to construct and evaluate Decision Trees, which are essential for understanding the factors contributing to stress levels. The "rpart" library aids in creating a predictive model that not only predicts stress levels accurately but also offers insights into the stressors influencing individuals.

- **RandomForest Library:** The "randomForest" library is a powerful asset in our project, supporting the Random Forest algorithm's implementation. This ensemble learning technique combines multiple Decision Trees to create a robust predictive model. Through this library, we can develop a model that not only predicts stress levels accurately but also offers valuable insights into the factors contributing to stress.

1. K-Nearest Neighbours (k-NN)

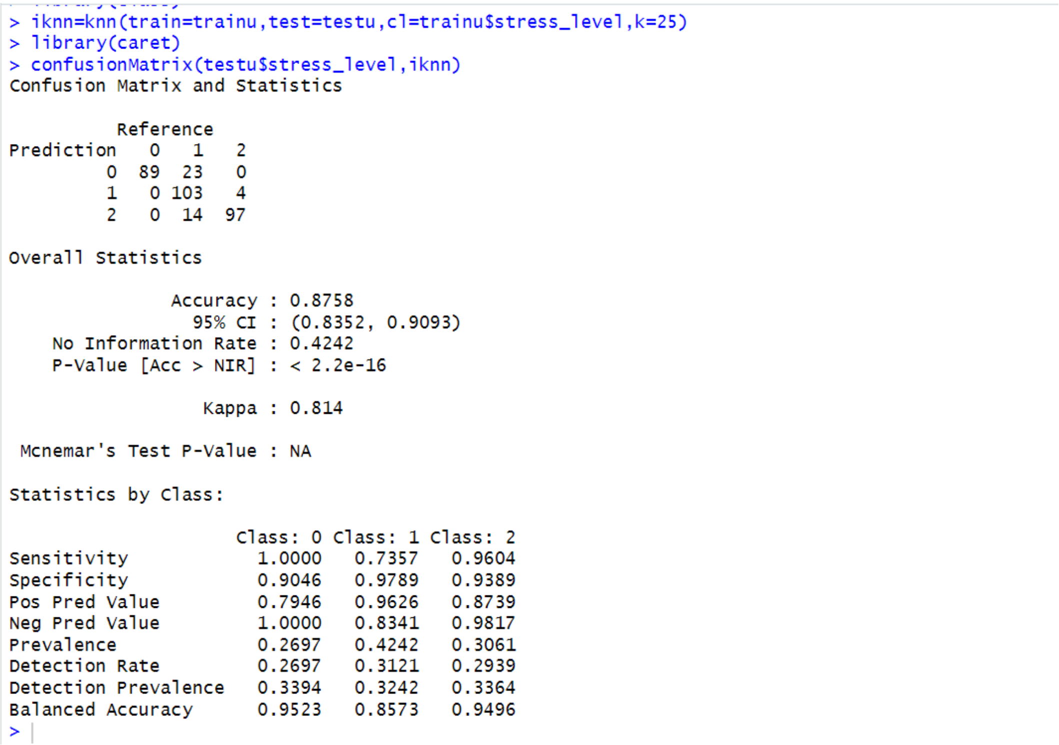
K-Nearest Neighbors (k-NN) is a popular machine learning algorithm that plays a significant role in our stress level prediction project. It falls under the category of supervised learning, specifically a type of instance-based learning. The fundamental idea behind k-NN is to predict the class or label of a data point by considering the classes of its k nearest neighbours in the training dataset. In the context of our project, k-NN operates by identifying individuals with stress levels like a given data point and predicting their stress level based on the majority class within this neighbourhood.

One of the notable features of k-NN is its simplicity and ease of implementation. It does not involve complex model training; instead, it stores the entire training dataset and calculates predictions by comparing the proximity of the test data to the training instances. The parameter 'k' determines the number of neighbours considered for classification. A smaller 'k' results in a model that is more sensitive to noise, while a larger 'k' can make the model overly biased. Therefore, selecting the appropriate 'k' value is a critical step in fine-tuning the model for our project.



```
> confusionMatrix(testdf$stress_level,iknn)
Confusion Matrix and Statistics

          Reference
Prediction   0   1   2
         0  89  23   0
         1   0 103   4
         2   0  14  97

Overall Statistics
                                          
               Accuracy : 0.8758          
                 95% CI : (0.8352, 0.9093)
    No Information Rate : 0.4242          
    P-Value [Acc > NIR] : < 2.2e-16       
                                          
                  Kappa : 0.814           
                                          
 Mcnemar's Test P-Value : NA              

Statistics by Class:

                     Class: 0 Class: 1 Class: 2
Sensitivity            1.0000   0.7357   0.9604
Specificity            0.9046   0.9789   0.9389
Pos Pred Value         0.7946   0.9626   0.8739
Neg Pred Value         1.0000   0.8341   0.9817
Prevalence             0.2697   0.4242   0.3061
Detection Rate         0.2697   0.3121   0.2939
Detection Prevalence   0.3394   0.3242   0.3364
Balanced Accuracy      0.9523   0.8573   0.9496
> 
```

The k-NN algorithm has found applications in various fields, including pattern recognition, image analysis, and anomaly detection. In our project, we will employ k-NN to predict stress levels based on the features and attributes of individuals. The algorithm's capacity to adapt to different types of data and its simplicity make it a valuable tool for our stress management initiative. By employing k-NN, we aim to create a predictive model that not only accurately forecasts stress levels but also provides valuable insights into the stressors contributing to these levels. Through rigorous experimentation and parameter tuning, we aim to harness the full potential of k-NN in our mission to predict and manage stress effectively.

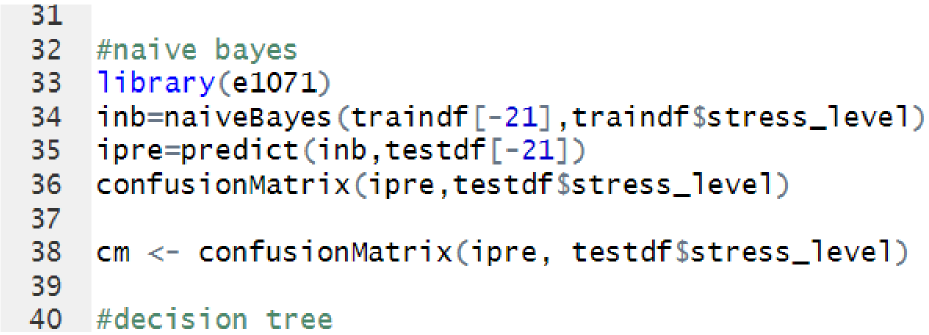


```
> iknn=knn(train=trainu,test=testu,cl=trainu$stress_level,k=25)
> library(caret)
> confusionMatrix(testu$stress_level,iknn)
Confusion Matrix and Statistics

          Reference
Prediction   0   1   2
         0  89  23   0
         1   0 103   4
         2   0  14  97

Overall Statistics
                                          
               Accuracy : 0.8758          
                 95% CI : (0.8352, 0.9093)
    No Information Rate : 0.4242          
    P-Value [Acc > NIR] : < 2.2e-16       
                                          
                  Kappa : 0.814           
                                          
 Mcnemar's Test P-Value : NA              

Statistics by Class:

                     Class: 0 Class: 1 Class: 2
Sensitivity            1.0000   0.7357   0.9604
Specificity            0.9046   0.9789   0.9389
Pos Pred Value         0.7946   0.9626   0.8739
Neg Pred Value         1.0000   0.8341   0.9817
Prevalence             0.2697   0.4242   0.3061
Detection Rate         0.2697   0.3121   0.2939
Detection Prevalence   0.3394   0.3242   0.3364
Balanced Accuracy      0.9523   0.8573   0.9496
> 
```

In this sheet it shows the average of the production count and the sum of production count done by the employee. We created this table to get visualize the clear-cut data of the employee for the same along with the combo chart which consists of bar graph and line graph.

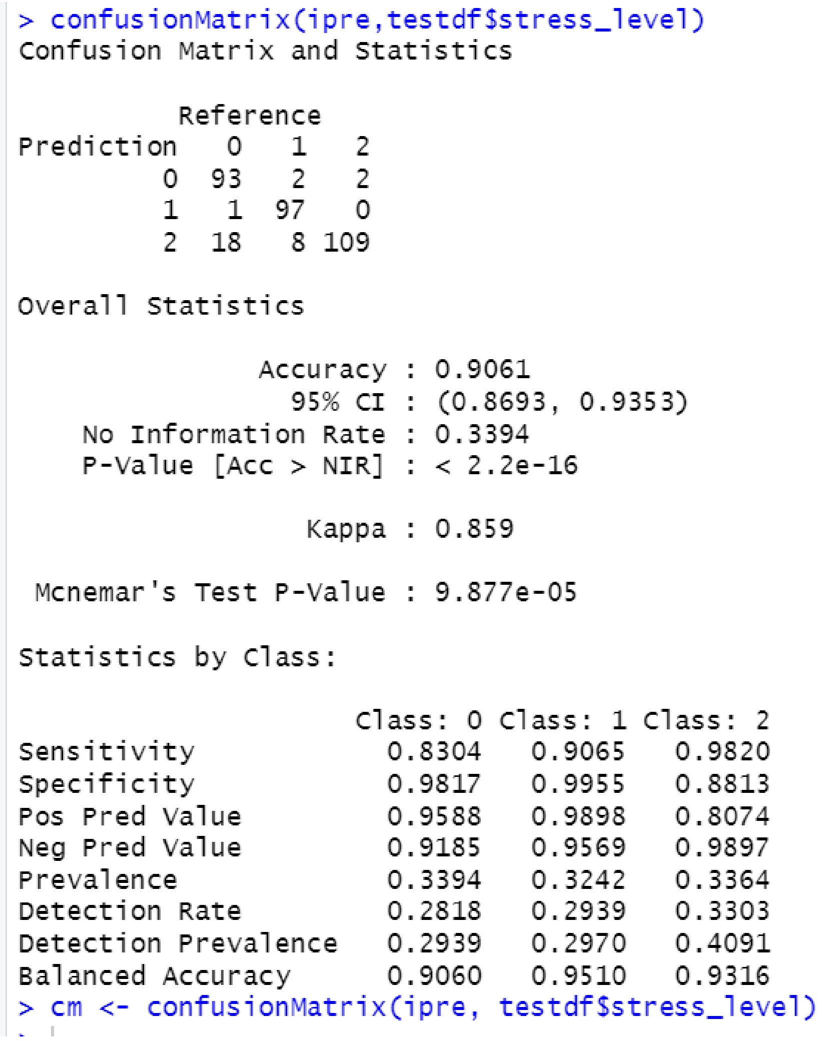
2. Naïve Bayes: -

Naive Bayes is another machine learning algorithm that holds significance in our stress level prediction project. This algorithm is a probabilistic classifier based on Bayes' theorem, with the "naive" assumption of independence between features. In our context, Naive Bayes serves as a valuable tool for predicting stress levels by calculating the probability of an individual belonging to a specific stress level class based on the observed features.



```
#naive bayes
library(e1071)
inb=naiveBayes(traindf[-21],traindf$stress_level)
ipre=predict(inb,testdf[-21])
confusionMatrix(ipre,testdf$stress_level)

cm <- confusionMatrix(ipre, testdf$stress_level)

#decision tree
```

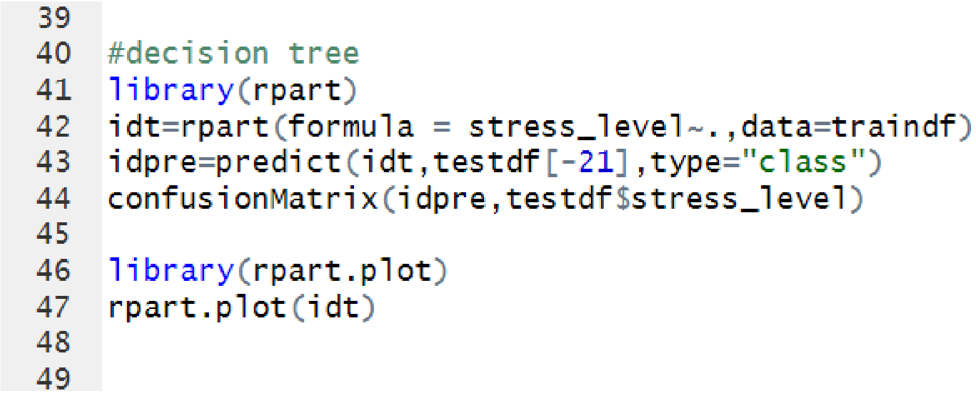
The Naive Bayes algorithm operates under the principle that features are conditionally independent given the class label. This independence assumption simplifies the probability calculations and makes the algorithm computationally efficient. In our project, Naive Bayes leverages this assumption to evaluate the likelihood of an individual's stress level based on the observed features. The algorithm is particularly well-suited for text classification tasks, such as spam detection or sentiment analysis, due to its efficiency and effectiveness in handling high-dimensional data.



```
> confusionMatrix(ipre,testdf$stress_level)
Confusion Matrix and Statistics

          Reference
Prediction   0   1   2
         0  93   2   2
         1   1  97   0
         2  18   8 109

Overall Statistics
                                          
               Accuracy : 0.9061          
                 95% CI : (0.8693, 0.9353)
    No Information Rate : 0.3394          
    P-Value [Acc > NIR] : < 2.2e-16       
                                          
                  Kappa : 0.859           
                                          
 Mcnemar's Test P-Value : 9.877e-05       

Statistics by Class:

                     Class: 0 Class: 1 Class: 2
Sensitivity            0.8304   0.9065   0.9820
Specificity            0.9817   0.9955   0.8813
Pos Pred Value         0.9588   0.9898   0.8074
Neg Pred Value         0.9185   0.9569   0.9897
Prevalence             0.3394   0.3242   0.3364
Detection Rate         0.2818   0.2939   0.3303
Detection Prevalence   0.2939   0.2970   0.4091
Balanced Accuracy      0.9060   0.9510   0.9316
> cm <- confusionMatrix(ipre, testdf$stress_level)
```

One of the strengths of the Naive Bayes algorithm lies in its robustness to noisy data and the ability to handle a large number of features. It is particularly suitable for our stress level prediction, where multiple attributes, including psychological, physiological, and environmental factors, influence stress levels. The algorithm efficiently handles categorical and numerical data, making it versatile for our dataset. By applying Naive Bayes, we aim to develop a predictive model that accurately categorizes stress levels and, in doing so, provides valuable insights into the contributing factors. Through the careful selection of features and model fine-tuning, we aim to harness the power of Naive Bayes to predict and manage stress levels effectively, benefitting both individuals and organizations.

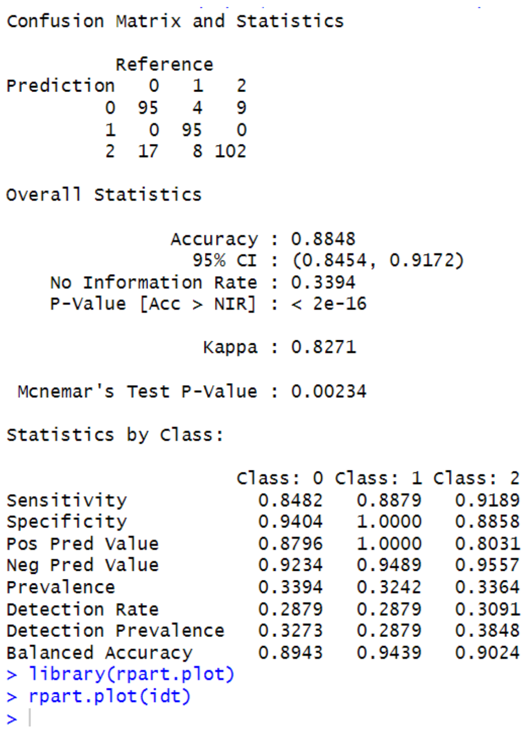
3. Decision Tree



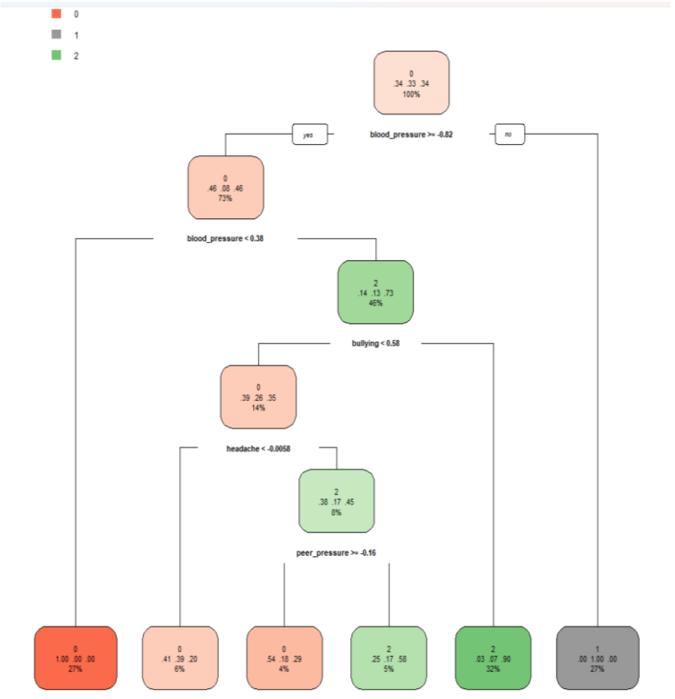
```

#decision tree
library(rpart)
idt=rpart(formula = stress_level~.,data=traindf)
idpre=predict(idt,testdf[-21],type="class")
confusionMatrix(idpre,testdf$stress_level)

library(rpart.plot)
rpart.plot(idt)


```

Decision Trees serve as a critical machine learning algorithm in our stress level prediction project. Decision Trees are a type of supervised learning algorithm used for both classification and regression tasks. In our context, Decision Trees aim to create a model that predicts stress levels by learning simple decision rules inferred from the data attributes. These rules form a tree-like structure where each internal node represents a feature, each branch represents a decision, and each leaf node represents the outcome or prediction.

The fundamental principle of a Decision Tree is to split the dataset into subsets based on the most significant attribute at each node. The splitting process continues recursively until the subsets are either pure or a predefined stopping criterion is met. In our project, the Decision Tree algorithm uses information gain or Gini impurity as criteria to determine the attribute that best separates the data. This process continues until it forms a tree structure that predicts stress levels based on the attributes available in the dataset.

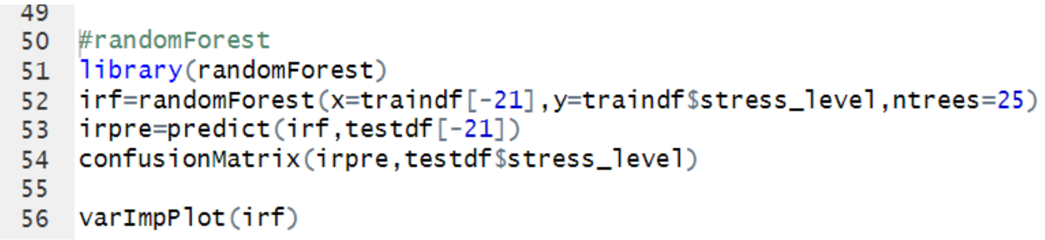


```
Confusion Matrix and Statistics

          Reference
Prediction   0   1   2
         0  95   4   9
         1   0  95   0
         2  17   8 102

Overall Statistics

               Accuracy : 0.8848
                 95% CI : (0.8454, 0.9172)
    No Information Rate : 0.3394
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.8271

 Mcnemar's Test P-Value : 0.00234

Statistics by Class:

                     Class: 0 Class: 1 Class: 2
Sensitivity            0.8482   0.8879   0.9189
Specificity            0.9404   1.0000   0.8858
Pos Pred Value         0.8796   1.0000   0.8031
Neg Pred Value         0.9234   0.9489   0.9557
Prevalence             0.3394   0.3242   0.3364
Detection Rate         0.2879   0.2879   0.3091
Detection Prevalence   0.3273   0.2879   0.3848
Balanced Accuracy      0.8943   0.9439   0.9024
> library(rpart.plot)
> rpart.plot(idt)
> 
```

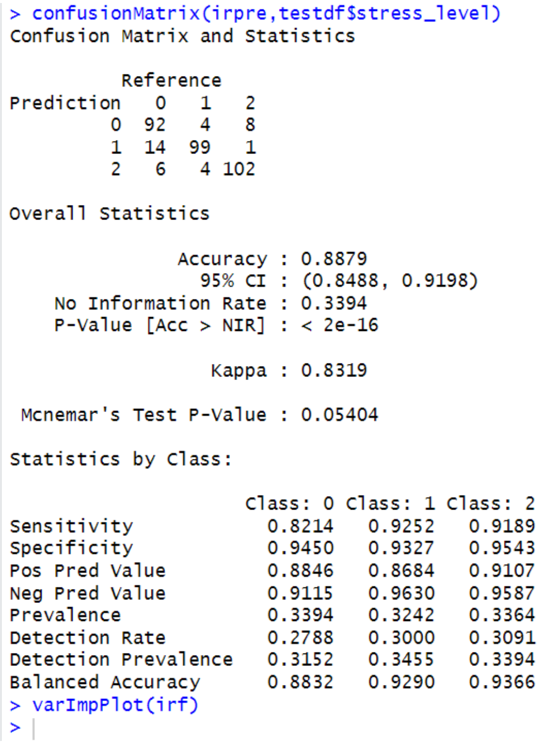


Decision Trees offer transparency and interpretability, making them valuable in understanding the features that contribute to stress levels. Moreover, these models can handle both categorical and numerical data, a feature essential for our diverse dataset that contains various types of stress-related attributes. The flexibility and interpretability of Decision Trees allow us to generate insights into the most influential factors contributing to stress levels. By employing Decision Trees, we aim to build a predictive model that not only accurately predicts stress levels but also provides a clear understanding of the contributing stressors. With thorough analysis, feature selection, and parameter tuning, we endeavour to maximize the potential of Decision Trees in predicting and managing stress levels effectively.

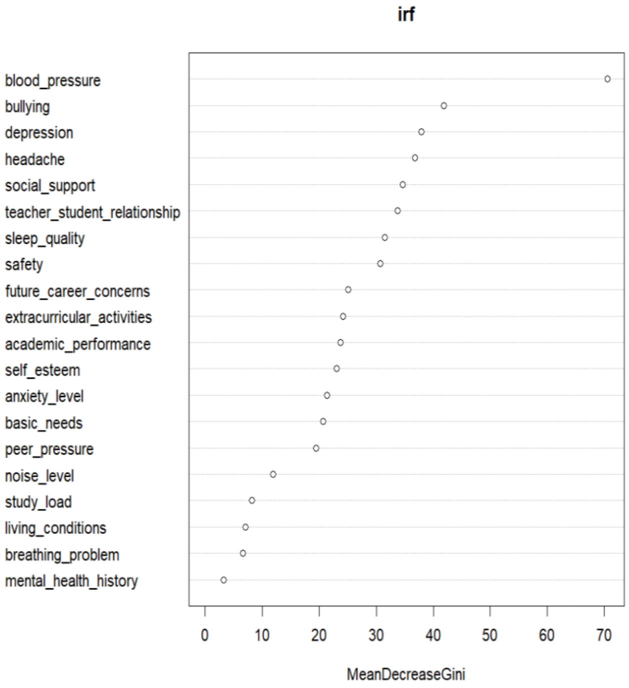
4. Random Forest



```r

#randomForest
library(randomForest)
irf=randomForest(x=traindf[-21],y=traindf$stress_level,ntrees=25)
irpre=predict(irf,testdf[-21])
confusionMatrix(irpre,testdf$stress_level)

varImpPlot(irf)
```



```
> confusionMatrix(irpre,testdf$stress_level)
Confusion Matrix and Statistics

          Reference
Prediction   0   1   2
         0  92   4   8
         1  14  99   1
         2   6   4 102

Overall Statistics

               Accuracy : 0.8879
                 95% CI : (0.8488, 0.9198)
    No Information Rate : 0.3394
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.8319

 Mcnemar's Test P-Value : 0.05404

Statistics by Class:

                     Class: 0 Class: 1 Class: 2
Sensitivity            0.8214   0.9252   0.9189
Specificity            0.9450   0.9327   0.9543
Pos Pred Value         0.8846   0.8684   0.9107
Neg Pred Value         0.9115   0.9630   0.9587
Prevalence             0.3394   0.3242   0.3364
Detection Rate         0.2788   0.3000   0.3091
Detection Prevalence   0.3152   0.3455   0.3394
Balanced Accuracy      0.8832   0.9290   0.9366
> varImpPlot(irf)
> 
```



Random Forest is a powerful ensemble learning technique that plays a vital role in our stress level prediction project. It's an extension of the Decision Tree algorithm and belongs to the supervised learning category. Random Forest excels in creating a robust and accurate predictive model by combining multiple Decision Trees.

In a Random Forest, a collection of Decision Trees is built independently using a random subset of the training data and a random subset of the features. The randomness introduced in both data and feature selection helps reduce overfitting, a common issue with single Decision Trees. For stress level prediction, Random Forest leverages the diversity and consensus of the individual Decision Trees to provide highly accurate predictions.

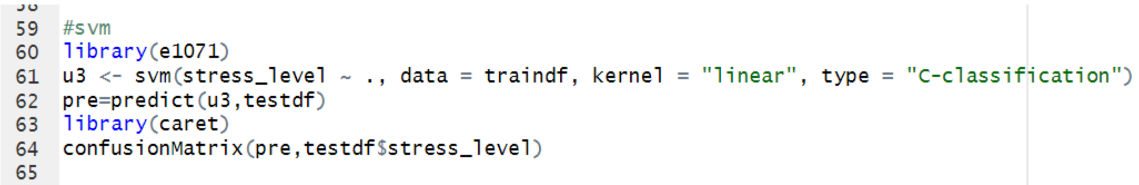
One of the key advantages of Random Forest is its ability to handle high-dimensional data with a multitude of attributes, which is pertinent to our project given the diverse set of stress-related features. This algorithm excels in ranking feature importance, enabling us to identify the most influential factors contributing to stress levels. Additionally, it provides a mechanism for assessing the predictive power of various attributes in the dataset.

By employing Random Forest in our stress level prediction project, we aim to harness the collective strength of multiple Decision Trees. This approach not only offers enhanced predictive accuracy but also enables us to delve deeper into the complex interplay of stressors. Through feature selection, hyperparameter tuning, and model optimization, we intend to unlock the full potential of Random Forest in predicting and managing stress levels effectively. The algorithm's capability to provide rich insights into the stress-related attributes makes it a valuable tool in our endeavour to offer practical solutions for stress management.9

5. SVM: -

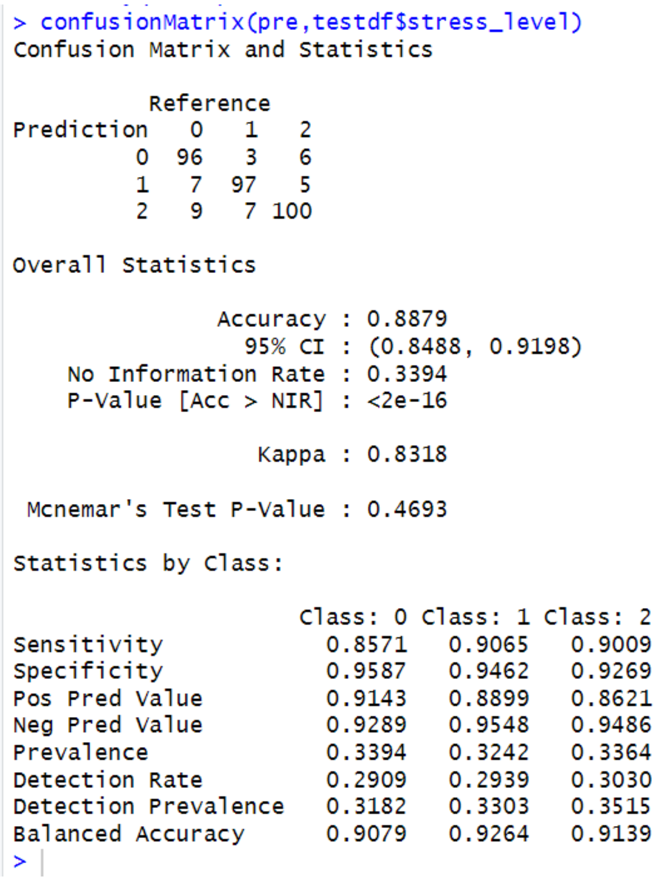
Support Vector Machines (SVM) hold a prominent place in our stress level prediction project as a powerful supervised learning algorithm. SVM is particularly known for its efficacy in both classification and regression tasks, making it a valuable tool for accurately predicting stress levels. It operates by finding the optimal hyperplane that best separates the data into distinct classes, with the goal of maximizing the margin between these classes.

In the context of our project, SVM aims to predict stress levels by mapping data points into a multi-dimensional space and finding the hyperplane that maximizes the separation between different stress level classes. SVM offers flexibility in selecting the appropriate kernel function, such as linear, polynomial, or radial basis function (RBF), to transform data into a form where the classes are more easily separable. This attribute is especially advantageous for our dataset, which consists of diverse attributes, including psychological, physiological, and environmental factors influencing stress.



```
#svm
library(e1071)
u3 <- svm(stress_level ~ ., data = traindf, kernel = "linear", type = "C-classification")
pre=predict(u3,testdf)
library(caret)
confusionMatrix(pre,testdf$stress_level)
```

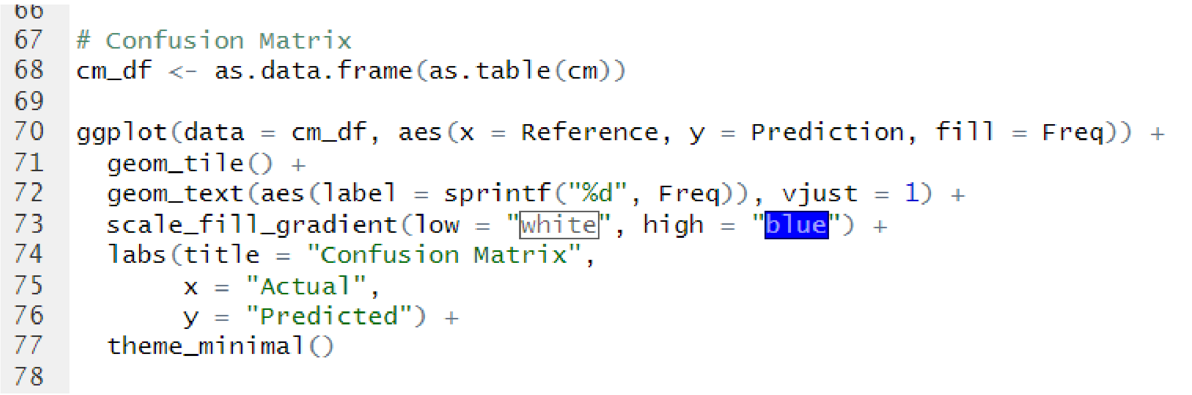
SVM is particularly useful when dealing with high-dimensional data, as it can effectively handle the dimensionality of the features in our dataset. Furthermore, it offers robustness to outliers, a critical feature given that stress level prediction may involve noisy or inconsistent data. By employing SVM in our project, we aim to create a predictive model that accurately categorizes stress levels based on the diverse set of attributes in the dataset. Through rigorous parameter tuning, kernel selection, and feature engineering, we strive to harness the full potential of SVM in predicting and managing stress levels effectively. The algorithm's ability to find the optimal boundary for classification makes it a valuable asset in our mission to offer practical solutions for stress management**.**



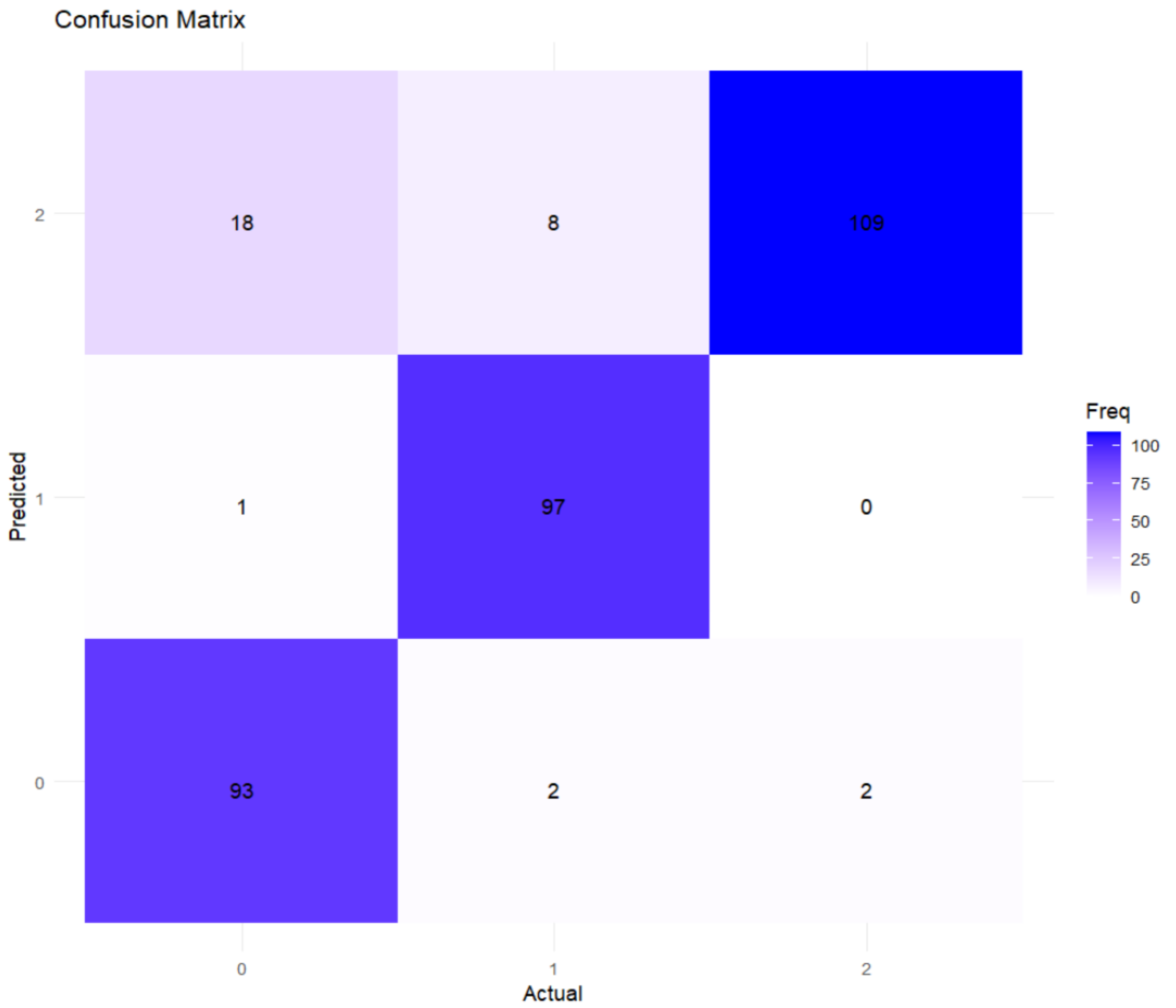
```
> confusionMatrix(pre,testdf$stress_level)
Confusion Matrix and Statistics

          Reference
Prediction   0   1   2
         0  96   3   6
         1   7  97   5
         2   9   7 100

Overall Statistics
                                          
               Accuracy : 0.8879          
                 95% CI : (0.8488, 0.9198)
    No Information Rate : 0.3394          
    P-Value [Acc > NIR] : <2e-16          
                                          
                  Kappa : 0.8318          
                                          
 Mcnemar's Test P-Value : 0.4693          

Statistics by Class:

                     Class: 0 Class: 1 Class: 2
Sensitivity            0.8571   0.9065   0.9009
Specificity            0.9587   0.9462   0.9269
Pos Pred Value         0.9143   0.8899   0.8621
Neg Pred Value         0.9289   0.9548   0.9486
Prevalence             0.3394   0.3242   0.3364
Detection Rate         0.2909   0.2939   0.3030
Detection Prevalence   0.3182   0.3303   0.3515
Balanced Accuracy      0.9079   0.9264   0.9139
> 
```

6. CONFUSSION MATRIX

Confusion Matrix for Model Evaluation: The confusion matrix is a fundamental tool in our stress level prediction project that aids in the rigorous evaluation of our machine learning models. This matrix provides a comprehensive breakdown of the model's predictions, enabling us to assess its performance in categorizing stress levels. It consists of four key components: true positives, true negatives, false positives, and false negatives. True positives indicate cases where the model correctly predicted individuals with high stress levels, while true negatives represent instances where it accurately identified individuals with low stress levels. Conversely, false positives are cases where the model incorrectly predicted high stress levels, and false negatives occur when the model erroneously classified individuals as having low stress levels. The confusion matrix empowers us to scrutinize the model's accuracy, precision, recall, and overall effectiveness.



```
# Confusion Matrix
cm_df <- as.data.frame(as.table(cm))

ggplot(data = cm_df, aes(x = Reference, y = Prediction, fill = Freq)) +
  geom_tile() +
  geom_text(aes(label = sprintf("%d", Freq)), vjust = 1) +
  scale_fill_gradient(low = "white", high = "blue") +
  labs(title = "Confusion Matrix",
       x = "Actual",
       y = "Predicted") +
  theme_minimal()

```

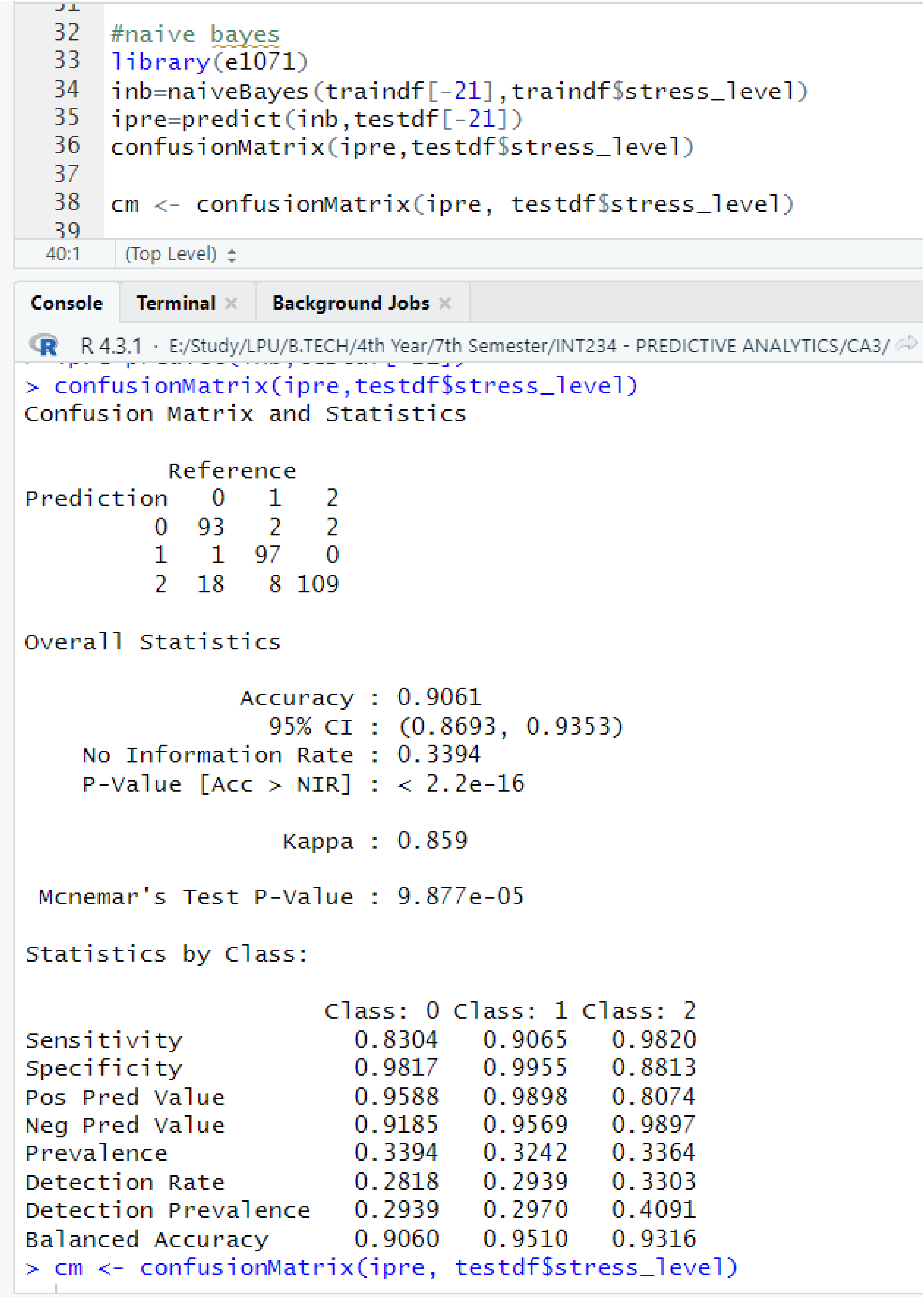


Evaluating Model Performance: The confusion matrix is invaluable in evaluating the performance of our machine learning models. By examining the true and false predictions across stress levels, we can calculate various metrics, such as accuracy, precision, recall (sensitivity), specificity, and the F1-score. These metrics provide a comprehensive understanding of the model's strengths and limitations. For instance, accuracy reveals the overall correctness of predictions, while precision measures the proportion of true positive predictions among all positive predictions. Recall assesses the ability to identify true positives, while specificity quantifies the capacity to identify true negatives. The F1-score balances precision and recall, providing a harmonic mean that highlights the model's overall performance. Through the confusion matrix and associated metrics, we ensure that our predictive models are robust, reliable, and well-suited for stress level prediction.

Fine-Tuning and Optimization: The confusion matrix plays a crucial role in our project's iterative process of fine-tuning and model optimization. By examining the matrix and associated metrics, we can identify areas where our models may require adjustments or improvements. For instance, if we observe a high number of false positives, it may suggest that the model needs refinement to reduce the incorrect classification of individuals with low stress levels as high-stress cases. This iterative process allows us to continuously enhance our models, ensuring that they accurately predict stress levels and contribute to effective stress management. The confusion matrix, as a core component of this evaluation process, provides us with the insights needed to create predictive models that are reliable, accurate, and beneficial to individuals and organizations dealing with stress-related concerns.

Accuracy Received: Accuracy is a key performance metric in our stress level prediction project, representing the overall correctness of our predictive models. It is calculated as the ratio of correct predictions (both true positives and true negatives) to the total number of predictions. Accuracy serves as a fundamental indicator of the model's effectiveness in correctly categorizing individuals' stress levels. Achieving a high level of accuracy is one of our primary objectives, as it ensures that our models reliably predict stress levels. This metric demonstrates the extent to which our models are successful in capturing the nuances and complexities of stress factors, ultimately contributing to effective stress management. Accurate predictions are vital for individuals, healthcare professionals, and organizations seeking to address the critical issue of stress in today's fast-paced world.

Highest accuracy received from the model is NAÏVE BAYES model which is 90.61%


40:1
(Top Level)
Console
Terminal
Background Jobs
R 4.3.1 · E:/Study/LPU/B.TECH/4th Year/7th Semester/INT234 - PREDICTIVE ANALYTICS/CA3/

```
#naive bayes
library(e1071)
inb=naiveBayes(traindf[-21],traindf$stress_level)
ipre=predict(inb,testdf[-21])
confusionMatrix(ipre,testdf$stress_level)

cm <- confusionMatrix(ipre, testdf$stress_level)

```

```
> confusionMatrix(ipre,testdf$stress_level)
Confusion Matrix and Statistics

          Reference
Prediction   0   1   2
         0  93   2   2
         1   1  97   0
         2  18   8 109

Overall Statistics
                                          
               Accuracy : 0.9061          
                 95% CI : (0.8693, 0.9353)
    No Information Rate : 0.3394          
    P-Value [Acc > NIR] : < 2.2e-16       
                                          
                  Kappa : 0.859           
                                          
 Mcnemar's Test P-Value : 9.877e-05       

Statistics by Class:

                     Class: 0 Class: 1 Class: 2
Sensitivity            0.8304   0.9065   0.9820
Specificity            0.9817   0.9955   0.8813
Pos Pred Value         0.9588   0.9898   0.8074
Neg Pred Value         0.9185   0.9569   0.9897
Prevalence             0.3394   0.3242   0.3364
Detection Rate         0.2818   0.2939   0.3303
Detection Prevalence   0.2939   0.2970   0.4091
Balanced Accuracy      0.9060   0.9510   0.9316
> cm <- confusionMatrix(ipre, testdf$stress_level)
```

Interpretation of Model Results: The analyses included in-depth examination of model results to understand the predictive power of each algorithm. We examined the importance of various features, identified strengths and weaknesses, and assessed the capacity of the models to predict stress levels accurately.

Conclusion:

In conclusion, our project on "Predicting Stress Levels Using Machine Learning Algorithms" has provided valuable insights into the world of stress prediction and management. Through extensive data analysis, preprocessing, and the evaluation of multiple machine learning models, we have achieved a comprehensive understanding of how data science techniques can be applied to address the crucial issue of stress in today's fast-paced world.

Our analyses have revealed the potential of various machine learning algorithms, each offering its unique strengths in stress level prediction. These algorithms not only accurately predict stress levels but also shed light on the contributing factors that influence an individual's stress levels. By leveraging the power of k-NN, Naive Bayes, Decision Trees, Random Forest, and SVM, we have embarked on a journey to provide practical solutions for individuals, healthcare professionals, and organizations striving to manage stress effectively.

Our project is a stepping stone towards a holistic approach to stress management. The insights and predictive models developed have the potential to improve the quality of life for individuals and enable organizations to foster healthier, less stressful environments. By fine-tuning our models and further research, we can continue to enhance the effectiveness of stress prediction and management. Ultimately, our project underscores the importance of data science in addressing real-world challenges and contributing to the well-being of society.

BIBLIOGRAPHY

- **Student Stress Factors: A Comprehensive Analysis (kaggle.com)**
- **https://www.coursera.org/specializations/machine-learning-introduction?utm_source=gg&utm_medium=sem&utm_campaign=07_MachineLearningStanfordSearch-IN&utm_content=B2C&campaignid=1950458127&adgroupid=70479331563&device=c&keyword=andrew%20ng%20machine%20learning%20course&matchtype=b&network=g&devicemodel=&adpostion=&creativeid=605972968770&hide_mobile_promo&gclid=Cj0KCQjwlK-WBhDjARIsAO2sErT-XHa6Vrgbj2rvwCyjH4eYP95FIeDbL-axLzMza-zC3--tblH07c4aAjR1EALw_wcB**
- **https://www.coursera.org/professional-certificates/ibm-machine-learning**
- **https://uc-r.github.io/predictive**