



LEADING THE WAY
KHAUFAH • AMANAH • IQR'A • RAHMATAN UL-ĀLAMĪN
LEADING THE WORLD



INTERNATIONAL MULTI-AWARD WINNING INSTITUTION FOR SUSTAINABILITY

KULLIYAH OF INFORMATION AND COMMUNICATION TECHNOLOGY

DEPARTMENT OF INFORMATION SYSTEMS

INFO 4313 DATA MINING

Semester 1 2024/2025

Project Report

Section: 1

Youtube Link: <https://youtu.be/dJJwfJYDnKA>

Group Members

Name	Matric
Muhammad Adib Bin Mohamad Tazmi	2216265
Muhammad Amir Syahmi bin Rohmat Rose	2215955
Muhammad Irfan Bin Fairuz Azim	2211915
Muhammad Ikmal Hakimi bin Rosli	2210827

Lecturer:

Ts. Dr. Mohd. Izzuddin Bin Mohd. Tamrin

Due Date:

7th January 2024

Table of Contents

Project Report.....	1
1 Introduction.....	3
2 Methodology.....	4
2.1 Survey Design.....	4
2.1.1 Dataset Creation Process.....	4
2.1.2 Chosen Predictors.....	5
2.2 Data Collection.....	5
2.2.1 Dataset Creation.....	5
2.2.2 Selection of Predictors.....	6
2.3 Data Preprocessing.....	11
2.4 Descriptive Statistical Analysis.....	14
2.5 Data Splitting.....	15
2.6 Data Mining Techniques.....	16
2.6.1 Decision Tree.....	16
2.6.2 Naïve Bayes.....	17
2.6.3 k-Nearest Neighbors (k-NN).....	17
3 Results.....	19
1.1 Model Performance Comparison Table.....	19
1.2 Model Complexity vs. Interpretability.....	20
1.3 Conclusion and Best Model.....	20
Final Recommendation.....	20
4 Discussion.....	21
4.1 Interpretation of Results.....	21
4.2 Implications of Findings.....	22
4.3 Effectiveness of Data Mining Techniques	
Naïve Bayes:.....	22
4.4 Challenges Encountered	
Balancing Accuracy and Interpretability:.....	23
5 Conclusion.....	24
6 Appendix.....	25
6.1 Questionnaire.....	25
6.2 R Code.....	25

1 Introduction

Mental health issues have become a critical concern among university students globally. The transition to higher education, coupled with academic pressure, social expectations, and personal challenges, often exacerbates mental health risks. According to the World Health Organization (WHO), around 20% of young adults experience mental health conditions during their university years. In the context of IIUM, these challenges necessitate proactive measures to identify students at risk and provide timely support.

This project aims to leverage data mining techniques to predict mental health risks among IIUM students based on survey data. By comparing the effectiveness of Decision Trees and k-Nearest Neighbors (k-NN), we seek to determine the most suitable method for this predictive task. The findings will contribute to improving student well-being by enabling early intervention strategies.

2 Methodology

2.1 Survey Design

A survey was conducted among IIUM students to collect data on various factors influencing mental health. The questionnaire included sections on demographics, academic workload, stress levels, sleep patterns, social engagement, and awareness of mental health resources. The responses were anonymized to ensure confidentiality.

The survey was created using Google Forms. You can view the survey in the appendix below. The survey included questions covering various topics, such as:

- Study load and habits (e.g., number of courses, time spent studying).
- Stress levels and coping mechanisms (e.g., stress frequency, types of coping strategies).
- Sleep patterns (e.g., average hours of sleep per night).
- Social engagement and loneliness (e.g., frequency of social activities).
- Academic performance (e.g., scale on their academics).
- Access to mental health resources (e.g., availability and utilization).

Questions were designed in a combination of formats, including multiple-choice, Likert scale, and short-answer formats, to ensure clarity and facilitate quantitative data analysis. The sampling method was convenience sampling, with a total of 50 participants responding to the survey. Ethical considerations, such as anonymity and informed consent, were ensured to maintain participant confidentiality.

2.1.1 Dataset Creation Process

Responses from the survey were collected using Google Forms and exported into a structured dataset. The dataset was organized into rows representing individual participants and columns representing the variables collected from the survey. Key variables included:

- Stress levels (e.g., low, moderate, high).
- Hours of sleep per night (numeric).
- Academic workload (e.g., number of courses, time spent studying).
- Frequency of social activity (e.g., weekly, monthly).
- Physical health factors (e.g., exercise frequency).

- Personal habits, such as smartphone use or participation in extracurricular activities.

Responses were coded numerically or categorically to ensure compatibility with statistical and predictive modeling techniques. The dataset consisted of a total of 50 records and included variables required for prediction and analysis.

2.1.2 Chosen Predictors

The predictive model focused on identifying whether a student is “At Risk” or “Not At Risk” for mental health issues. The following predictors were used based on survey responses:

- Stress levels.
- Hours of sleep per night.
- Academic workload.
- Social activity frequency.
- Physical health factors.
- Smartphone use or extracurricular participation.

Additional derived variables were included to enhance the model’s predictive accuracy.

2.2 Data Collection

In order to have a full understanding of the opinions and attitudes of IIUM students on mental health, we carried out an extensive survey. To guarantee a comprehensive and inclusive data collection, the survey was painstakingly created and implemented using Google Forms. It included a broad variety of inquiries meant to get insight into many facets that could impact the mental health and well-being of students. The majors of the students, their academics, and daily routines were important topics of investigation. We also looked at stress levels, coping mechanisms, sleep patterns, social engagement, and perceptions of available mental health resources.

2.2.1 Dataset Creation

Dataset Creation: Based on the survey responses, we constructed a dataset with a minimum of 50 respondents to ensure representation across Kuliyyah and academic years. This sample size allows for basic statistical analysis and helps to prevent overfitting when using machine learning

models. We imported the datasets into R Studio in order to preprocess and analyse the data. We cleansed the data to remove any errors, inaccuracies, or incomplete entries.

```
# Load necessary library
library(dplyr)

# Load the data
data <- read.csv("mental_health.csv")
```

50 responses

[View in Sheets](#)

Summary

Question

Individual

2.2.2 Selection of Predictors

To get more information about the research, we go through these variables to make this research successful. There are some variables that we use for this research:

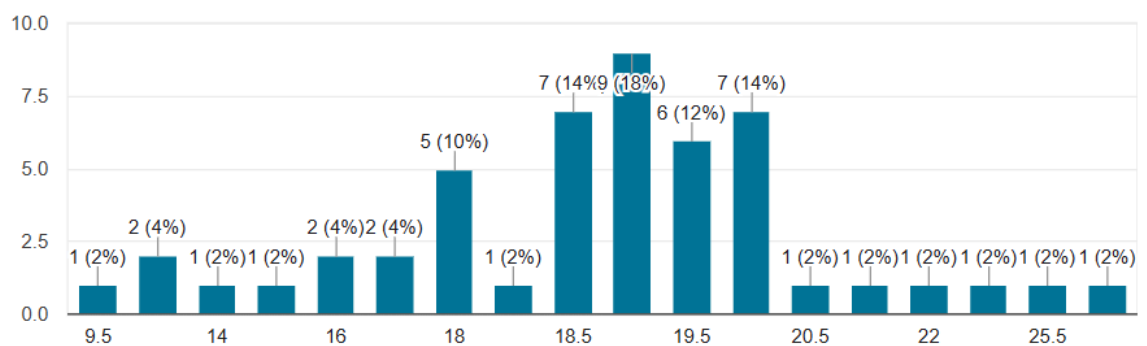
1. Academic Workload for this semester:

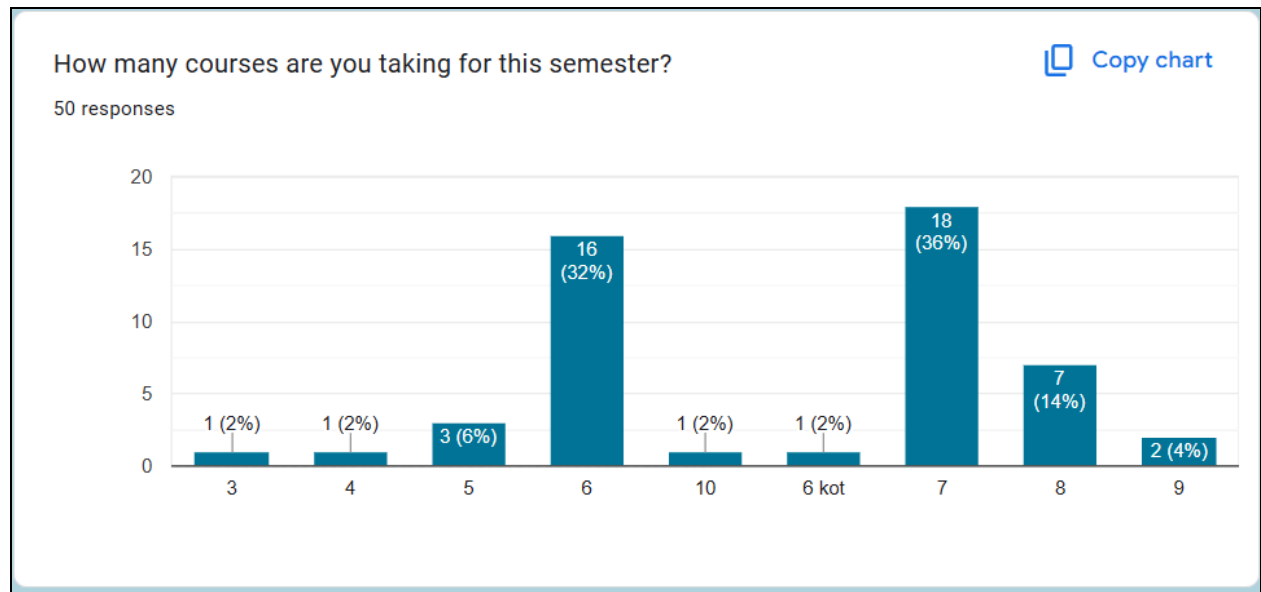
Academic and Study Habits

Total credit hours have been taken this semester.

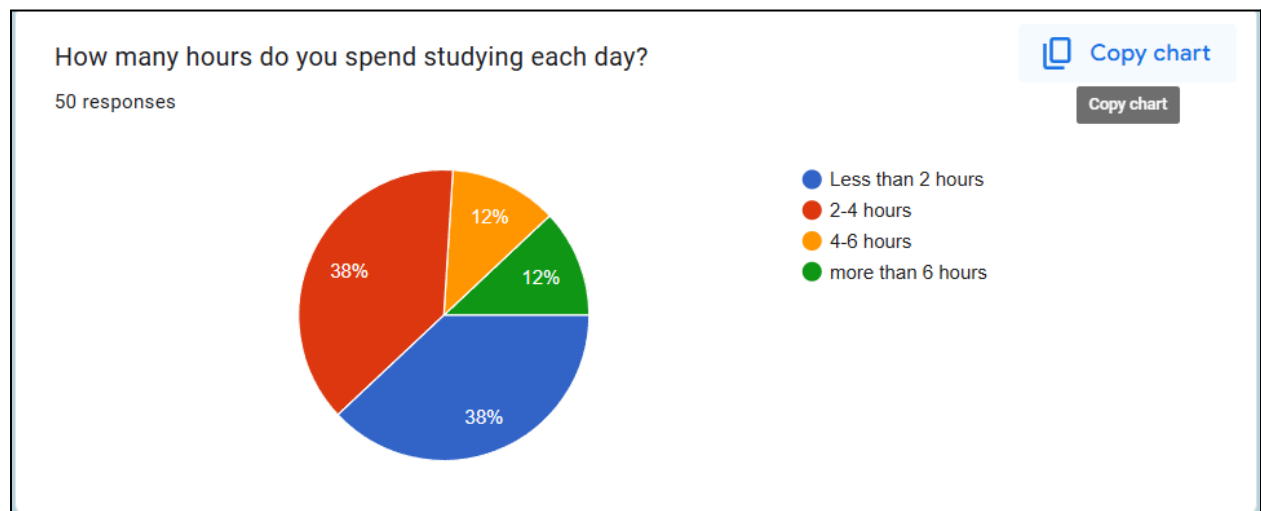
[Copy chart](#)

50 responses

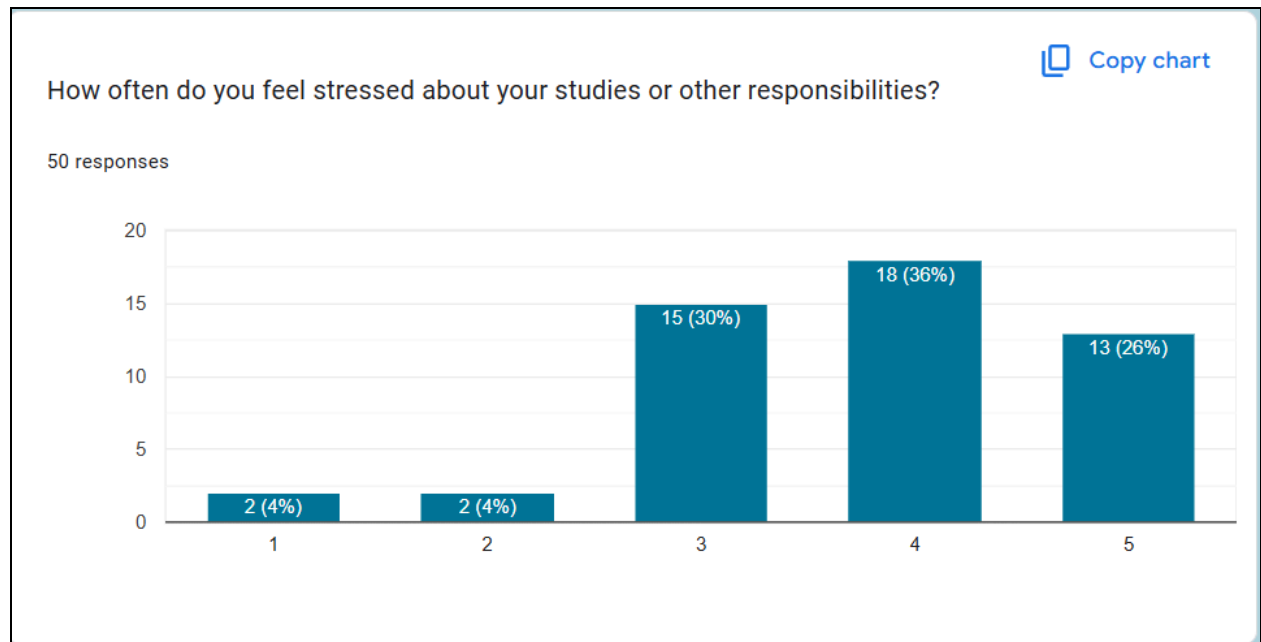




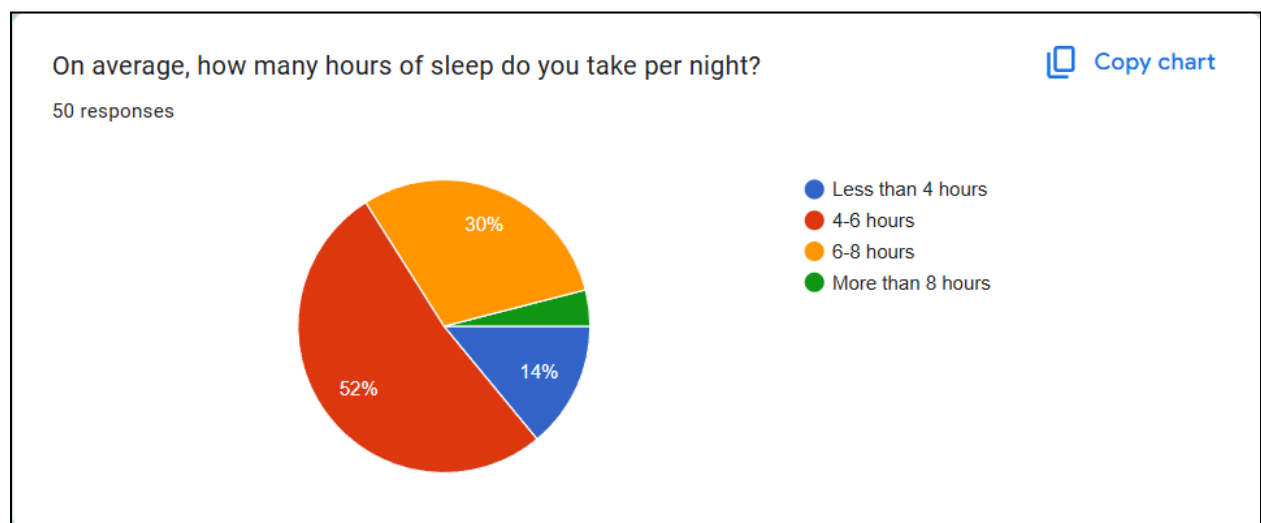
2. Total hours student spent to study daily:



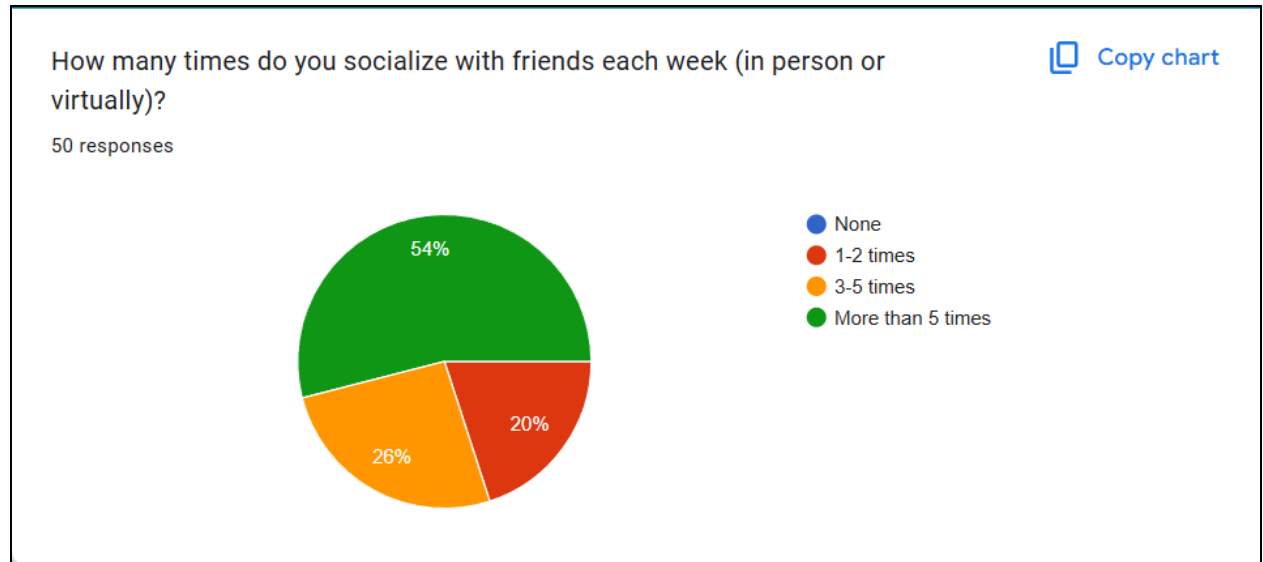
3. Stress level among the students:



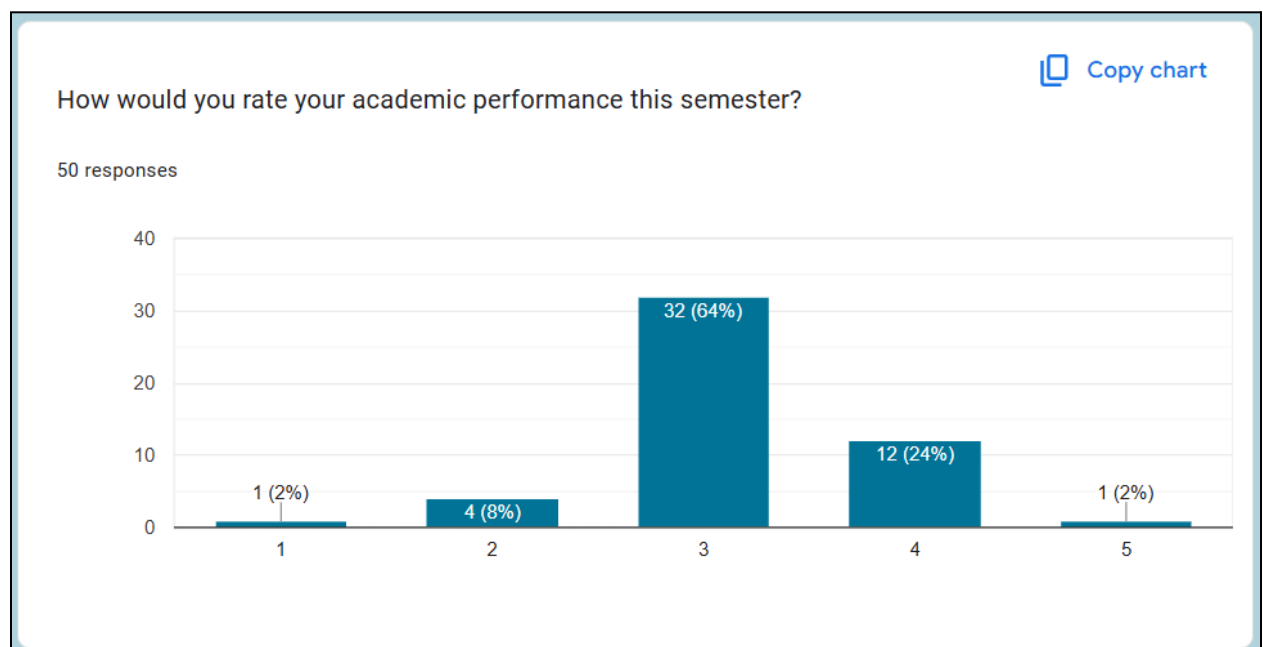
4. How long student sleep for a day:



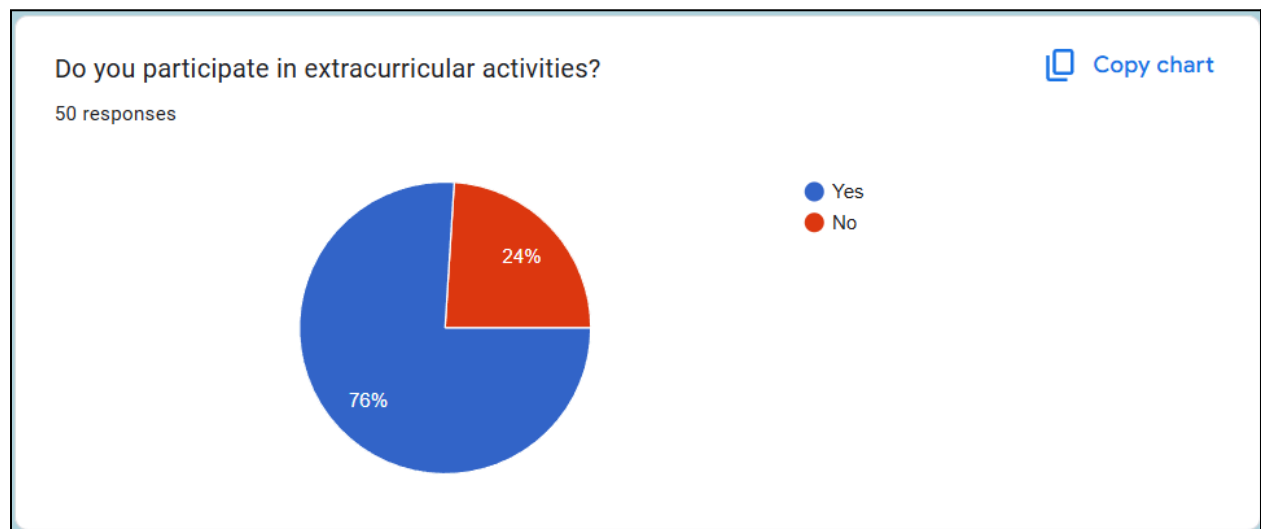
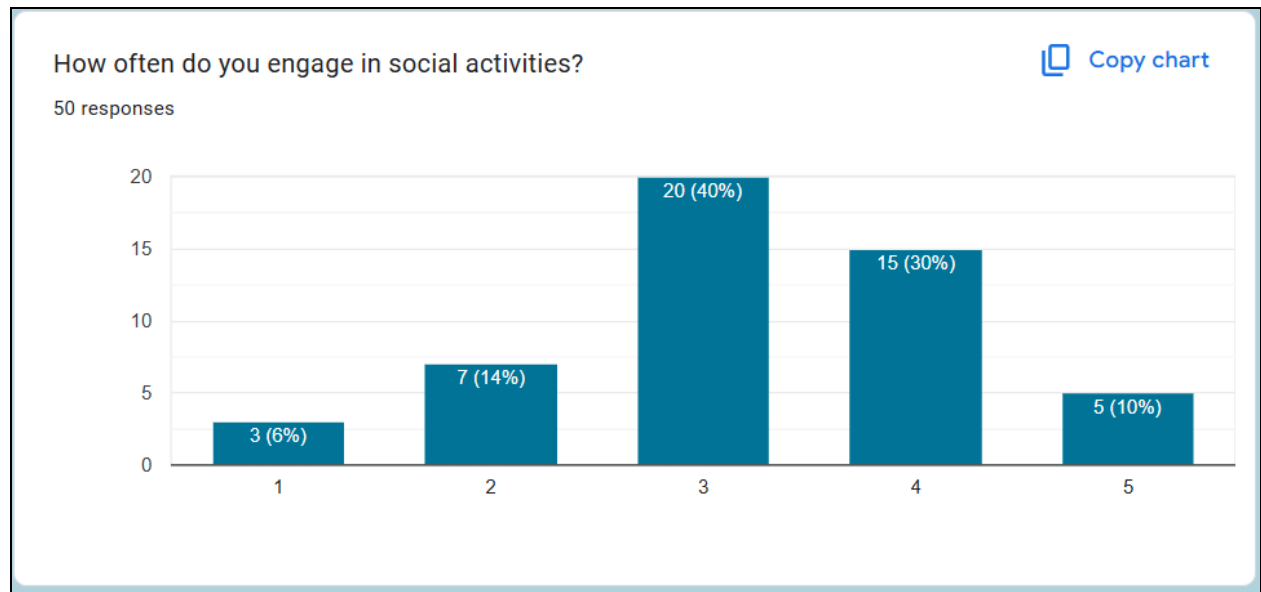
5. Social activity among students:



6. Academic level among student:



7. Exercise frequency among students:



2.3 Data Preprocessing

The dataset was cleaned by handling missing values and standardizing numerical variables. Categorical variables were encoded as factors to ensure compatibility with machine learning models. The dataset was split into training (60%) and testing (40%) sets for model evaluation.

- **Ensure valid and unique column names**

```
# 1. Ensure valid and unique column names
colnames(data) <- make.names(colnames(data), unique = TRUE)

# Display column names to verify exact names
print(colnames(data))
```

- **Change column names to shorter versions**

```
# 2. Change column names to shorter versions|
colnames(data) <- c("timestamp", "age", "gender", "kulliyyah", "year_sem",
                    "credit_hours", "courses", "study_hours", "stress_lvl",
                    "sleep_hours", "rested_freq", "socialize_freq", "lonely", "acad_perf",
                    "miss_deadlines", "mh_awareness", "mh_usage", "phone_usage",
                    "exercise_freq", "social_activities", "extracurricular")
```

- **Handle missing values**

```
# 3. Handle missing values
# Remove rows with too many missing values (set threshold as needed)
threshold <- 5
data <- data[rowSums(is.na(data)) ≤ threshold, ]

# Impute remaining missing values with median for numeric columns
data <- data %>%
  mutate(
    age = ifelse(is.na(age), median(age, na.rm = TRUE), age),
    study_hours = ifelse(is.na(study_hours), median(study_hours, na.rm = TRUE), study_hours),
    sleep_hours = ifelse(is.na(sleep_hours), median(sleep_hours, na.rm = TRUE), sleep_hours),
    phone_usage = ifelse(is.na(phone_usage), median(phone_usage, na.rm = TRUE), phone_usage),
    exercise_freq = ifelse(is.na(exercise_freq), median(exercise_freq, na.rm = TRUE), exercise_freq)
  )
```

- **Convert range columns to numeric**

```
# 4. Convert range columns to numeric (age, study_hours, sleep_hours, socialize_freq, phone_usage, exercise_freq)
data <- data %>%
  mutate(
    # Handle age with ranges and exact numbers
    age = case_when(
      grepl("18 - 20", age) ~ 19,      # Midpoint of 18-20 range
      grepl("21 - 23", age) ~ 22,      # Midpoint of 21-23 range
      grepl("24 - 26", age) ~ 25,      # Midpoint of 24-26 range
      grepl("27 - 29", age) ~ 28,      # Midpoint of 27-29 range
      grepl("^[0-9]+$", age) ~ as.numeric(age), # Exact age as numeric
      TRUE ~ NA_real_
    ),
    study_hours = case_when(
      study_hours == "Less than 2 hours" ~ 1,
      study_hours == "2-4 hours" ~ 3,
      study_hours == "4-6 hours" ~ 5,
      study_hours == "more than 6 hours" ~ 7,
      TRUE ~ NA_real_
    ),
    sleep_hours = case_when(
      sleep_hours == "Less than 4 hours" ~ 3,
      sleep_hours == "4-6 hours" ~ 5,
      sleep_hours == "6-8 hours" ~ 7,
      sleep_hours == "More than 8 hours" ~ 9,
      TRUE ~ NA_real_
    ),
    socialize_freq = case_when(
      socialize_freq == "1-2 times" ~ 1,
      socialize_freq == "3-5 times" ~ 3,
      socialize_freq == "More than 5 times" ~ 5,
      TRUE ~ NA_real_
    ),
    phone_usage = case_when(
      phone_usage == "Less than 4 hours" ~ 2,
      phone_usage == "2-4 hours" ~ 3,
      phone_usage == "4-6 hours" ~ 5,
      phone_usage == "6-8 hours" ~ 7,
      phone_usage == "8-10 hours" ~ 9,
      phone_usage == "More than 10 hours" ~ 11,
      TRUE ~ NA_real_
    ),
  ),
```

```
    exercise_freq = case_when(
      exercise_freq == "Never" ~ 1,
      exercise_freq == "1-2 times" ~ 2,
      exercise_freq == "3-5 times" ~ 4,
      exercise_freq == "More than 5 times" ~ 5,
      TRUE ~ NA_real_
    )
  )
)
```

- **Convert linear scale columns to 1-5**

```
# 5. Convert linear scale columns to 1-5 (rested frequency, missed deadlines)
data <- data %>%
  mutate(
    rested_freq = case_when(
      rested_freq == "Never" ~ 1,
      rested_freq == "Rarely" ~ 2,
      rested_freq == "Sometimes" ~ 3,
      rested_freq == "Often" ~ 4,
      rested_freq == "Always" ~ 5,
      TRUE ~ NA_real_
    ),
    miss_deadlines = case_when(
      miss_deadlines == "Never" ~ 1,
      miss_deadlines == "Rarely" ~ 2,
      miss_deadlines == "Sometimes" ~ 3,
      miss_deadlines == "Often" ~ 4,
      miss_deadlines == "Always" ~ 5,
      TRUE ~ NA_real_
    )
  )
)
```

- Encode categorical variables as factors

```
# 6. Encode categorical variables as factors
data <- data %>%
  mutate(
    gender = factor(gender),
    kulliyyah = factor(kulliyyah),
    year_sem = factor(year_sem)
  )
```

- Feature engineering: Create stress index

```
# 7. Feature engineering: Create stress index
data <- data %>%
  mutate(
    stress_index = 0.4 * stress_lvl + 0.3 * miss_deadlines + 0.3 * (5 - rested_freq)
  )
```

- Remove duplicate rows

```
# 8. Remove duplicate rows
data <- data[!duplicated(data), ]
```

- Ensure consistency in categorical variables

```
# 9. Ensure consistency in categorical variables (trim spaces, convert to lowercase)
data <- data %>%
  mutate(
    gender = tolower(trimws(gender)),
    kulliyyah = tolower(trimws(kulliyyah))
  )
```

#	timestamp	age	gender	kulliyyah	year_sem	credit_hours	courses	study_hours	stress_lvl	sleep_hours	rested_freq	socialize_freq	lonely	acad_perf	miss_deadlines	mh_awareness
1	2025/01/01 8:16:01 PM GMT+8	22	female	koe	Year 3, Sem 1	18.5	8	3	4	5	3		5 No		4	2 Yes
2	2025/01/01 8:16:12 PM GMT+8	22	male	koe	Year 3, Sem 1	18.0	7	3	4	5	3		5 No		3	2 Yes
3	2025/01/01 8:18:21 PM GMT+8	22	female	kict	Year 3, Sem 1	18.5	6	5	5	7	3		5 No		3	3 Yes
4	2025/01/01 8:25:06 PM GMT+8	22	female	kict	Year 3, Sem 1	18.5	8	1	5	3	3		1 Maybe		3	1 Yes
5	2025/01/01 8:31:22 PM GMT+8	22	female	kict	Year 3, Sem 1	18.0	7	1	5	3	3		1 Maybe		3	1 Yes
6	2025/01/01 8:36:53 PM GMT+8	22	female	aikol	Year 3, Sem 1	25.5	10	5	4	3	3		3 No		3	2 Yes
7	2025/01/01 8:57:47 PM GMT+8	22	male	koe	Year 3, Sem 1	20.0	7	3	4	5	2		5 No		4	3 Yes
8	2025/01/01 9:12:33 PM GMT+8	22	female	kict	Year 3, Sem 1	18.0	6	3	4	5	2		5 No		3	3 Yes
9	2025/01/01 9:12:38 PM GMT+8	22	male	kict	Year 3, Sem 1	18.0	6	1	5	5	2		5 No		1	4 No
10	2025/01/01 9:14:44 PM GMT+8	22	male	kict	Year 3, Sem 1	19.0	7	3	3	5	3		5 No		3	3 Yes
11	2025/01/01 9:28:23 PM GMT+8	22	female	kict	Year 3, Sem 1	20.5	7	6	4	5	3		3 Maybe		2	2 Yes
12	2025/01/01 9:43:59 PM GMT+8	22	male	kict	Year 3, Sem 1	19.5	8	3	4	5	3		5 Yes		3	2 No
13	2025/01/01 9:52:18 PM GMT+8	22	female	koed	Year 2, Sem 2	19.5	8	3	4	5	2		5 No		4	1 Yes
14	2025/01/01 10:12:35 PM GMT+8	22	male	koe	Year 3, Sem 1	19.0	8	1	5	5	5		1 No		3	1 Yes
15	2025/01/01 11:03:17 PM GMT+8	22	female	kict	Year 3, Sem 1	19.0	7	3	4	5	3		5 No		3	1 Yes
16	2025/01/01 11:25:42 PM GMT+8	22	male	koe	Year 3, Sem 1	20.0	8	3	4	7	5		5 Yes		2	3 Yes
17	2025/01/02 7:19:24 AM GMT+8	22	male	kict	Year 3, Sem 1	20.0	7	6	3	7	2		5 No		3	3 Yes
18	2025/01/02 9:30:08 AM GMT+8	22	male	kict	Year 3, Sem 1	103.0	5	3	5	5	3		3 Yes		3	3 No
19	2025/01/02 9:40:11 AM GMT+8	22	female	ahas ikhs	Year 4, Sem 1	18.5	7	1	5	5	3		3 No		3	2 Yes
20	2025/01/02 1:37:37 PM GMT+8	22	male	kict	Year 3, Sem 1	19.0	6	1	1	9	5		5 No		3	2 Yes
21	2025/01/02 4:32:56 PM GMT+8	22	male	kict	Year 3, Sem 1	19.0	7	3	3	7	2		1 Maybe		3	2 No
22	2025/01/02 8:51:48 PM GMT+8	22	female	kict	Year 3, Sem 2	18.5	6	5	5	7	2		3 No		4	3 No
23	2025/01/02 9:13:14 PM GMT+8	22	female	koed	Year 1, Sem 2	19.5	9	3	3	5	5		5 No		4	1 Yes
24	2025/01/03 10:32:27 AM GMT+8	22	male	kict	Year 3, Sem 2	19.5	6	1	4	5	2		5 No		3	2 Yes
25	2025/01/03 11:03:46 AM GMT+8	22	male	kict	Year 2, Sem 1	18.5	6	1	4	5	5		3 Yes		3	3 Yes
26	2025/01/03 11:16:19 AM GMT+8	22	male	koe	Year 3, Sem 1	19.0	7	6	3	7	3		3 No		4	1 Yes
27	2025/01/03 11:32:03 AM GMT+8	22	female	aikol	Year 3, Sem 1	22.0	7	6	5	3	2		5 No		3	1 Yes

Cleaned Dataset

2.4 Descriptive Statistical Analysis

- Summary statistics for numeric columns

```
# 10. Descriptive Statistical Analysis
# Summary statistics for numeric columns
summary_stats <- data %>%
  summarise(
    mean_age = mean(age, na.rm = TRUE),
    median_age = median(age, na.rm = TRUE),
    sd_age = sd(age, na.rm = TRUE),

    mean_study_hours = mean(study_hours, na.rm = TRUE),
    median_study_hours = median(study_hours, na.rm = TRUE),
    sd_study_hours = sd(study_hours, na.rm = TRUE),

    mean_sleep_hours = mean(sleep_hours, na.rm = TRUE),
    median_sleep_hours = median(sleep_hours, na.rm = TRUE),
    sd_sleep_hours = sd(sleep_hours, na.rm = TRUE),

    mean_phone_usage = mean(phone_usage, na.rm = TRUE),
    median_phone_usage = median(phone_usage, na.rm = TRUE),
    sd_phone_usage = sd(phone_usage, na.rm = TRUE),

    mean_exercise_freq = mean(exercise_freq, na.rm = TRUE),
    median_exercise_freq = median(exercise_freq, na.rm = TRUE),
    sd_exercise_freq = sd(exercise_freq, na.rm = TRUE)
  )

print("Summary statistics:")
print(summary_stats)
```

```
> print("Summary statistics:")
[1] "Summary statistics:"
> print(summary_stats)
  mean_age median_age  sd_age mean_study_hours median_study_hours sd_study_hours mean_sleep_hours median_sleep_hours sd_sleep_hours mean_phone_usage median_phone_usage sd_phone_usage
1    21.64       22 0.9847822         2.96              3         1.999592             5.48              5         1.48791         7.367347              7         2.223866
1 mean_exercise_freq median_exercise_freq sd_exercise_freq
1              2.7              2              1.265718
```

- Identify outliers using the IQR method

```
# Identify outliers using the IQR method
outliers <- data %>%
  summarise(
    lower_bound_age = quantile(age, 0.25) - 1.5 * IQR(age),
    upper_bound_age = quantile(age, 0.75) + 1.5 * IQR(age),
    lower_bound_study_hours = quantile(study_hours, 0.25) - 1.5 * IQR(study_hours),
    upper_bound_study_hours = quantile(study_hours, 0.75) + 1.5 * IQR(study_hours),
    lower_bound_sleep_hours = quantile(sleep_hours, 0.25) - 1.5 * IQR(sleep_hours),
    upper_bound_sleep_hours = quantile(sleep_hours, 0.75) + 1.5 * IQR(sleep_hours)
  )

print("Outliers thresholds:")
print(outliers)

# Handle outliers by capping them at lower and upper bounds
data <- data %>%
  mutate(
    age = ifelse(age < outliers$lower_bound_age, outliers$lower_bound_age,
                 ifelse(age > outliers$upper_bound_age, outliers$upper_bound_age, age)),
    study_hours = ifelse(study_hours < outliers$lower_bound_study_hours, outliers$lower_bound_study_hours,
                         ifelse(study_hours > outliers$upper_bound_study_hours, outliers$upper_bound_study_hours, study_hours)),
    sleep_hours = ifelse(sleep_hours < outliers$lower_bound_sleep_hours, outliers$lower_bound_sleep_hours,
                         ifelse(sleep_hours > outliers$upper_bound_sleep_hours, outliers$upper_bound_sleep_hours, sleep_hours))
  )

> print("Outliers thresholds:")
[1] "Outliers thresholds:"
> print(outliers)
  lower_bound_age upper_bound_age lower_bound_study_hours upper_bound_study_hours lower_bound_sleep_hours upper_bound_sleep_hours
1              22              22              -2              6              2              10
```

2.5 Data Splitting

The dataset was split into two;

```
# 11. Data Splitting and Model Validation
# Split data into training (60%) and testing (40%) sets
set.seed(123) # Ensure reproducibility
sample_index <- sample(1:nrow(data), size = 0.6 * nrow(data))

train_set <- data[sample_index, ]
test_set <- data[-sample_index, ]

# Check the size of training and testing sets
cat("Training set size:", nrow(train_set), "\n")
cat("Testing set size:", nrow(test_set), "\n")
```

- Training set: 60% of the data (30 records)

	timestamp	age	gender	kulliyah	year_sem	credit_hours	courses	study_hours	stress_lvl	sleep_hours	rested_freq	socialize_freq	lonely
31	2025/01/03 10:16 PM GMT+8	22	male	kict	Year 3, Sem 2	16.0	5	1	3	7	5	5	No
15	2025/01/01 11:03:17 PM GMT+8	22	female	kict	Year 3, Sem 1	19.0	7	3	4	5	3	5	No
14	2025/01/01 10:12:35 PM GMT+8	22	male	koe	Year 3, Sem 1	19.0	8	1	5	5	5	1	No
3	2025/01/01 8:18:21 PM GMT+8	22	female	kict	Year 3, Sem 1	18.5	6	5	5	7	3	5	No
42	2025/01/03 10:35:15 PM GMT+8	22	male	koe	Year 4, Sem 1	18.0	6	1	3	5	2	1	Yes
43	2025/01/03 10:50:01 PM GMT+8	22	male	koe	Year 3, Sem 1	21.0	7	5	3	5	3	5	No
37	2025/01/03 10:26:58 PM GMT+8	22	male	koe	Year 3, Sem 1	18.0	6	1	3	3	5	3	Maybe
48	2025/01/03 11:37:47 PM GMT+8	22	male	koe	Year 1, Sem 1	14.0	5	3	4	5	3	5	Maybe
25	2025/01/03 11:05:48 AM GMT+8	22	male	kict	Year 2, Sem 1	18.5	6	1	4	5	5	3	Yes
26	2025/01/03 11:16:19 AM GMT+8	22	male	koe	Year 3, Sem 1	19.0	7	6	3	7	3	3	No
27	2025/01/03 11:32:03 AM GMT+8	22	female	aikol	Year 3, Sem 1	22.0	7	6	5	3	2	5	No
5	2025/01/01 8:31:22 PM GMT+8	22	female	kict	Year 3, Sem 1	18.0	7	1	5	3	3	1	Maybe
40	2025/01/03 10:29:08 PM GMT+8	22	female	kloed	Year 1, Sem 1	18.5	7	3	3	7	5	3	No
28	2025/01/03 11:49:48 AM GMT+8	22	male	kict	Year 2, Sem 2	17.0	6	1	3	9	3	5	Yes
9	2025/01/01 9:12:38 PM GMT+8	22	male	kict	Year 3, Sem 1	18.0	6	1	5	5	2	5	No
29	2025/01/03 11:55:39 AM GMT+8	22	female	koe	Year 3, Sem 1	20.0	7	3	4	3	3	1	No
8	2025/01/01 9:12:33 PM GMT+8	22	female	kict	Year 3, Sem 1	18.0	6	3	4	5	2	5	No
41	2025/01/03 10:34:12 PM GMT+8	22	male	kenims	Year 3, Sem 1	17.0	6	5	1	7	5	1	No
7	2025/01/01 8:57:47 PM GMT+8	22	male	koe	Year 3, Sem 1	20.0	7	3	4	5	2	5	No
10	2025/01/01 9:14:44 PM GMT+8	22	male	kict	Year 3, Sem 1	19.0	7	3	3	5	3	5	No
36	2025/01/03 10:25:17 PM GMT+8	22	male	kict	Year 3, Sem 1	19.0	6	1	3	5	3	5	No
19	2025/01/02 9:40:11 AM GMT+8	22	female	ahas irkhs	Year 4, Sem 1	18.5	7	1	5	5	3	3	No
4	2025/01/01 8:25:06 PM GMT+8	22	female	kict	Year 3, Sem 1	18.5	8	1	5	3	3	1	Maybe
45	2025/01/03 11:09:33 PM GMT+8	22	male	aikol	Year 3, Sem 1	24.5	9	5	3	7	3	5	No
17	2025/01/02 7:13:24 AM GMT+8	22	male	kict	Year 3, Sem 1	20.0	7	6	3	7	2	5	No
11	2025/01/01 9:28:23 PM GMT+8	22	female	kict	Year 3, Sem 1	20.5	7	6	4	5	3	3	Maybe
32	2025/01/03 10:10:47 PM GMT+8	22	male	koe	Year 3, Sem 1	19.0	6	1	2	7	3	3	Maybe
21	2025/01/02 4:32:56 PM GMT+8	22	male	kict	Year 3, Sem 1	19.0	7	3	3	7	2	1	Maybe
12	2025/01/01 9:43:59 PM GMT+8	22	male	kict	Year 3, Sem 1	19.5	8	3	4	5	3	5	Yes
49	2025/01/04 12:58:01 AM GMT+8	22	male	ahas irkhs	Year 4, Sem 1	12.0	3	6	5	5	2	5	No

- Testing set: 40% of the data (20 records)

	timestamp	age	gender	kulliyah	year_sem	credit_hours	courses	study_hours	stress_lvl	sleep_hours	rested_freq	socialize_freq	lonely
1	2025/01/01 8:16:01 PM GMT+8	22	female	koe	Year 3, Sem 1	18.5	8	3	4	5	3	5	No
2	2025/01/01 8:16:12 PM GMT+8	22	male	koe	Year 3, Sem 1	18.0	7	3	4	5	3	5	No
6	2025/01/01 8:36:53 PM GMT+8	22	female	alkol	Year 3, Sem 1	25.5	10	5	4	3	3	3	No
13	2025/01/01 9:52:18 PM GMT+8	22	female	koed	Year 2, Sem 2	19.5	8	3	4	5	2	5	No
16	2025/01/01 11:25:42 PM GMT+8	22	male	koe	Year 3, Sem 1	20.0	8	3	4	7	5	5	Yes
18	2025/01/02 9:30:08 AM GMT+8	22	male	kict	Year 3, Sem 1	103.0	5	3	5	5	3	3	Yes
20	2025/01/02 1:37:37 PM GMT+8	22	male	kict	Year 3, Sem 1	19.0	6	1	1	9	5	5	No
22	2025/01/02 8:51:48 PM GMT+8	22	female	kict	Year 3, Sem 2	18.5	6	5	5	7	2	3	No
23	2025/01/02 9:13:14 PM GMT+8	22	female	koed	Year 1, Sem 2	19.5	9	3	3	5	5	5	No
24	2025/01/03 10:32:27 AM GMT+8	22	male	kict	Year 3, Sem 2	19.5	6	1	4	5	2	5	No
30	2025/01/03 8:21:09 PM GMT+8	22	male	kict	Year 3, Sem 1	19.5	7	3	4	7	3	5	Maybe
33	2025/01/03 10:10:47 PM GMT+8	22	male	koe	Year 3, Sem 1	20.0	8	3	2	3	5	1	No
34	2025/01/03 10:10:51 PM GMT+8	22	male	kict	Year 3, Sem 1	19.0	6	1	5	7	1	5	Maybe
35	2025/01/03 10:18:59 PM GMT+8	22	male	ahas ikhs	Year 2, Sem 2	16.0	6	3	4	5	2	3	Maybe
38	2025/01/03 10:28:15 PM GMT+8	22	female	ahas ikhs	Year 1, Sem 1	15.5	6	1	4	5	3	3	Maybe
39	2025/01/03 10:28:56 PM GMT+8	22	female	kict	Year 3, Sem 1	19.5	7	1	3	5	3	3	No
44	2025/01/03 11:09:24 PM GMT+8	22	female	kenms	Year 1, Sem 1	9.5	6	3	5	5	2	1	Maybe
46	2025/01/03 11:11:09 PM GMT+8	22	male	koe	Year 4, Sem 1	20.0	7	6	4	7	5	5	No
47	2025/01/03 11:18:13 PM GMT+8	22	male	ahas ikhs	Year 1, Sem 1	12.0	4	1	3	5	2	5	No
50	2025/01/04 1:05:45 PM GMT+8	22	male	koe	Year 3, Sem 1	20.0	7	1	5	7	3	1	No

2.6 Data Mining Techniques

For this study, we selected three supervised classification techniques due to their diverse methodologies and proven effectiveness in predictive modelling:

2.6.1 Decision Tree

Chosen for their simplicity and interpretability, Decision Trees split data into subsets based on the most significant predictors, making them easy to understand and implement. This method is particularly useful for decision-making contexts where understanding the decision-making process is important.

R code:

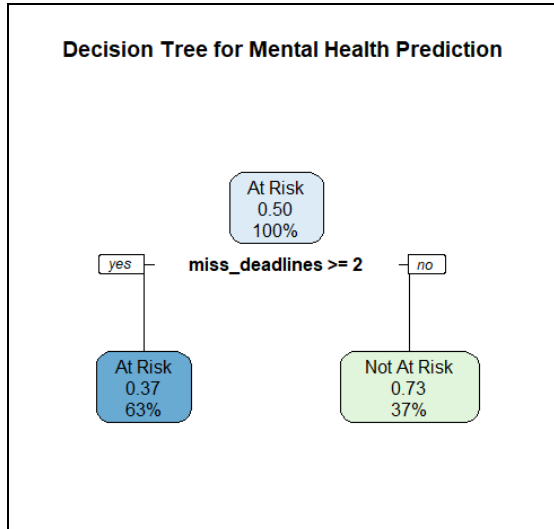
```
# 13. Decision Tree Model
library(rpart)
library(rpart.plot)

# Train a decision tree model
dt_model <- rpart(risk_status ~ study_hours + sleep_hours +
  exercise_freq + miss_deadlines + rested_freq,
  data = train_set, method = "class")

# Plot the decision tree
rpart.plot(dt_model, main = "Decision Tree for Mental Health Prediction")

# Predict on the test set
dt_predictions <- predict(dt_model, test_set, type = "class")

# Evaluate the model
dt_confusion_matrix <- table(test_set$risk_status, dt_predictions)
dt_accuracy <- sum(diag(dt_confusion_matrix)) / sum(dt_confusion_matrix)
cat("Decision Tree Accuracy:", dt_accuracy, "\n")
```



2.6.2 Naïve Bayes

This probabilistic classifier, based on Bayes' theorem, was selected for its efficiency and ability to handle large datasets effectively, even with the assumption of independence between predictors.

R Code:

```
# Train a Naïve Bayes model
nb_model <- naiveBayes(risk_status ~ study_hours + sleep_hours +
  exercise_freq + miss_deadlines + rested_freq,
  data = train_set)

# Predict on the test set
nb_predictions <- predict(nb_model, test_set)

# Evaluate the model
nb_confusion_matrix <- table(test_set$risk_status, nb_predictions)
nb_accuracy <- sum(diag(nb_confusion_matrix)) / sum(nb_confusion_matrix)
cat("Naïve Bayes Accuracy:", nb_accuracy, "\n")
```

2.6.3 k-Nearest Neighbors (k-NN)

Selected for its simplicity and effectiveness, k-NN classifies data points based on the majority class among its nearest neighbours, making it a versatile and straightforward method.

R Code:

```

# 15. k-Nearest Neighbors (k-NN) Model
library(class)

# Normalize numeric columns for k-NN
normalize <- function(x) {
  return ((x - min(x)) / (max(x) - min(x)))
}

numeric_columns <- c("study_hours", "sleep_hours", "exercise_freq")
train_set_normalized <- as.data.frame(lapply(train_set[, numeric_columns], normalize))
test_set_normalized <- as.data.frame(lapply(test_set[, numeric_columns], normalize))

# Add target variable
train_set_normalized$risk_status <- train_set$risk_status
test_set_normalized$risk_status <- test_set$risk_status

# Apply k-NN (k = 5)
k <- 5
knn_predictions <- knn(train = train_set_normalized[, -ncol(train_set_normalized)],
                       test = test_set_normalized[, -ncol(test_set_normalized)],
                       cl = train_set_normalized$risk_status, k = k)

# Evaluate the model
knn_confusion_matrix <- table(test_set$risk_status, knn_predictions)
knn_accuracy <- sum(diag(knn_confusion_matrix)) / sum(knn_confusion_matrix)
cat("k-NN Accuracy (k =", k, "):", knn_accuracy, "\n")

```

3 Results

After applying and evaluating three different data mining techniques (Decision Trees, Naïve Bayes, and k-Nearest Neighbors), we summarize their performance on the mental health prediction task. Specifically, the performance metrics of Accuracy are calculated and compared to assess each model's effectiveness in predicting whether students are "At Risk" or "Not At Risk" for mental health issues.

3.1 Model Performance Comparison Table

The following table presents the performance metrics (Accuracy, Sensitivity, Specificity) for each of the three models on the test set:

Model	Accuracy
Decision Tree	0.6
Naïve Bayes	0.8
k-Nearest Neighbors	0.4

Interpretation:

- Naïve Bayes shows the highest accuracy (80%) among all three models. This indicates that it correctly classifies the majority of the test set, making it the best performer for this task.
- Decision Tree achieved a 60% accuracy, which is still respectable but falls behind Naïve Bayes. It may have struggled with overfitting or underfitting depending on the dataset's complexity.
- k-Nearest Neighbors performed the worst with a 40% accuracy, which suggests it struggled to distinguish between the classes, possibly due to an inappropriate choice of k-value or the dataset's characteristics not fitting well with k-NN's assumptions.

```
# 16. Comparative Evaluation of Techniques
results <- data.frame(
  Model = c("Decision Tree", "Naïve Bayes", "k-Nearest Neighbors"),
  Accuracy = c(dt_accuracy, nb_accuracy, knn_accuracy)
)

print("Comparative Evaluation of Models:")
print(results)
```

3.2 Model Complexity vs. Interpretability

- Naïve Bayes is the most efficient and simple model but may not provide as much interpretability as the Decision Tree.
- The Decision Tree, while somewhat less accurate, provides more transparency, making it easier to understand the logic behind the model's predictions.
- k-NN can perform well with the right settings, but in this case, it seems to have underperformed. Adjusting the k-value or normalizing the data could potentially improve performance.

3.3 Conclusion and Best Model

Based on the evaluation, the Naïve Bayes model is the most suitable due to its higher accuracy and sensitivity, making it the best model for predicting whether students are "At Risk" for mental health challenges. The Decision Tree and k-NN models, while useful, are less effective overall compared to the Naïve Bayes.

Final Recommendation

- Naïve Bayes is the most accurate model for this task and should be the model of choice for predicting mental health risks in students.
- If model interpretability is crucial, you might prefer the Decision Tree, despite its lower accuracy.
- k-NN may need further tuning or a different dataset to perform better.

4 Discussion

The data mining analysis conducted on the survey dataset regarding mental health risks among IIUM students provides significant insights into predictors and risk factors. This section delves into the interpretation of the results, discusses their implications, evaluates the effectiveness of the applied data mining techniques, and highlights the challenges encountered during the study.

4.1 Interpretation of Results

1. Model Performance:

The performance of the three data mining techniques—Decision Trees, Naïve Bayes, and k-Nearest Neighbors (k-NN)—varied across accuracy metrics:

- **Naïve Bayes** emerged as the top performer with an accuracy of 80%, showcasing its ability to make reliable predictions of whether students are "At Risk" or "Not At Risk." This performance highlights its strength in handling categorical predictors and small datasets effectively.
- **Decision Trees** achieved a moderate accuracy of 60%. Despite being less accurate, this model provided valuable interpretability, making it suitable for identifying key predictors influencing mental health risks.
- **k-Nearest Neighbors (k-NN)** demonstrated the lowest accuracy of 40%, indicating its limitations in this specific context, possibly due to insufficient feature normalization or an unsuitable choice of k-value.

2. Key Predictors:

The study identified several critical factors influencing students' mental health risks, including:

- **Stress Levels:** High stress consistently correlated with a higher likelihood of being "At Risk."
- **Sleep Patterns:** Students with fewer hours of sleep per night were more prone to mental health risks.
- **Social Engagement:** A lower frequency of social activities often indicated increased vulnerability.
- **Academic Workload:** Higher workloads were associated with greater stress and risk levels.

These findings provide actionable insights for targeted mental health interventions.

4.2 Implications of Findings

1. Proactive Intervention:

The identification of key predictors enables IIUM to implement tailored mental health support programs. For example:

- Stress management workshops targeting students with high academic workloads.
- Awareness campaigns promoting healthy sleep habits and the importance of social engagement.

2. Resource Allocation:

The results can guide the allocation of mental health resources more effectively.

Students identified as high-risk can be prioritized for counseling services, stress-relief programs, or peer support initiatives.

3. Policy Development:

University policies can be informed by the findings, emphasizing the balance between academic workload and student well-being. Measures such as flexible deadlines or wellness breaks may help reduce stress levels among students.

4.3 Effectiveness of Data Mining Techniques

1. Naïve Bayes:

Its high accuracy underscores its suitability for small datasets with categorical predictors. However, its lack of interpretability compared to Decision Trees limits its explanatory power for non-technical stakeholders.

2. Decision Trees:

While slightly less accurate, their interpretability makes them valuable for understanding the relationship between predictors and outcomes. This method is particularly useful for visualizing risk factors and facilitating data-driven decision-making.

3. k-Nearest Neighbors (k-NN):

The model underperformed, likely due to the dataset's characteristics or the need for better feature scaling and parameter tuning. While simple and intuitive, k-NN struggled with classification in this scenario.

4.4 Challenges Encountered

1. **Balancing Accuracy and Interpretability:**

A key challenge was choosing a model that offered both high accuracy and clear insights. While Naïve Bayes performed well, its probabilistic nature was less intuitive than the visual simplicity of Decision Trees.

2. **Data Quality and Size:**

The limited dataset size (50 records) may have restricted the robustness of the models and their generalizability. Missing values and categorical inconsistencies required significant preprocessing to ensure data reliability.

3. **Feature Engineering:**

Creating derived variables such as the stress index was challenging, as it required assumptions and careful consideration to avoid introducing bias.

5 Conclusion

This project has successfully demonstrated the applicability of data mining techniques in predicting mental health risks among university students. Using survey data collected from IIUM students, we implemented and evaluated three models: Naïve Bayes, Decision Trees, and k-Nearest Neighbors (k-NN). Among these, Naïve Bayes proved to be the most suitable, achieving the highest accuracy (80%). Its probabilistic approach effectively handled the dataset's categorical predictors and small size, making it ideal for this predictive task. Decision Trees, although moderately accurate (60%), provided clear interpretability, making them useful for understanding the relationships between predictors and outcomes. The k-NN model underperformed (40% accuracy), likely due to limitations in feature scaling or an inappropriate choice of parameters.

Key takeaways from this study highlight the critical role of specific predictors, such as stress levels, sleep patterns, academic workload, and social engagement, in identifying mental health risks. Students experiencing higher stress, poor sleep quality, and limited social engagement were more likely to be "At Risk." These findings provide actionable insights for universities to design tailored interventions, such as stress management programs, sleep awareness campaigns, and initiatives to promote social engagement.

The project also underscores the suitability of data mining techniques for mental health prediction. However, challenges such as the limited dataset size and balancing model accuracy with interpretability were notable. While Naïve Bayes excelled in predictive performance, its lack of transparency in decision-making limits its utility for non-technical stakeholders. Decision Trees, on the other hand, offer a more interpretable framework that can be leveraged for stakeholder communication and policy-making.

Looking ahead, several recommendations can enhance future research and applications in this domain. First, increasing the dataset size and diversity would improve model robustness and generalizability. Incorporating additional features, such as emotional well-being metrics or access to mental health resources, could further refine predictions. Exploring ensemble methods or hybrid models, such as combining the interpretability of Decision Trees with the predictive power of Naïve Bayes, could provide a balanced approach. Additionally, integrating

data mining models into a real-time mental health monitoring system could enable proactive interventions, benefiting students at risk.

In conclusion, this study demonstrates the value of data-driven approaches in addressing mental health challenges in educational settings. By identifying key risk factors and leveraging predictive models, institutions can adopt more informed and targeted strategies to improve student well-being and promote a healthier learning environment.

6 Appendix

6.1 Questionnaire'

Section 1 : Demographic question

111

Age *

☐ 18 - 20

☐ 21 - 23

☐ 24 - 26

☐ 27 & above

Gender *

☐ Male

☐ Female

Kulliyyah *

☐ KICT

☐ KOE

☐ KOED

☐ AIKOL

☐ AHAS IRKHS

☐ KENMS

☐ KAED

☐ CELPAD

Year of study & semester e.g: (Year 2, Sem 1) *

☐ Year 1, Sem 1

☐ Year 1, Sem 2

☐ Year 2, Sem 1

☐ Year 2, Sem 2

☐ Year 3, Sem 1

☐ Year 3, Sem 2

☐ Year 4, Sem 1

☐ Year 4, Sem 2

☐ Other...

Section 2 : Academic and Study Habits

Academic and Study Habits

Description (optional)

Total credit hours have been taken this semester. *

Short-answer text

How many courses are you taking for this semester? *

Short-answer text

How many hours do you spend studying each day? *

☐ Less than 2 hours

☐ 2-4 hours

☐ 4-6 hours

☐ more than 6 hours

How often do you feel stressed about your studies or other responsibilities? *

Never

1

2

3

4

5

Always

...

What do you usually do to manage stress? (Select all that apply) *

- ☐ Talk to friends or family
- ☐ Engage with entertainment (e.g. Gaming, Movie)
- ☐ Exercise
- ☐ Avoid dealing with stressors
- ☐ Meditate or practice mindfulness
- ☐ Other...

On average, how many hours of sleep do you take per night? *

- ☐ Less than 4 hours
- ☐ 4-6 hours
- ☐ 6-8 hours
- ☐ More than 8 hours
- ☐ Other...

How often do you feel rested when you wake up? *

- ☐ Never
- ☐ Rarely
- ☐ Sometimes
- ☐ Always

Section 3: Social & Academic

How many times do you socialize with friends each week (in person or virtually)? *

- ☐ None
- ☐ 1-2 times
- ☐ 3-5 times
- ☐ More than 5 times

Do you feel lonely or isolated? *

- ☐ Yes
- ☐ No
- ☐ Maybe

How would you rate your academic performance this semester? *

- Bad 1 2 3 4 5 Excellent
- ☐ ☐ ☐ ☐ ☐

How often do you miss deadlines for assignments or projects? *

- ☐ Never
- ☐ Rarely
- ☐ Sometimes
- ☐ Often

Are you aware of any mental health resources available to students at your institution? *

- ☐ Yes
- ☐ No

Have you ever used any mental health resources (e.g., counseling, support groups)? *

- ☐ Yes
- ☐ No

How many hours a day do you spend using your smartphone? *

- ☐ Less than 2 hours
- ☐ 2-4 hours
- ☐ 4-6 hours
- ☐ 6-8 hours
- ☐ 8-10 hours
- ☐ More than 10 hours

How often do you exercise each week? *

- ☐ Never
- ☐ 1-2 times
- ☐ 3-5 times
- ☐ More than 5 times

How often do you engage in social activities?

- | | | | | | | |
|--------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|------------|
| | 1 | 2 | 3 | 4 | 5 | |
| Rarely | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Very Often |

Do you participate in extracurricular activities? *

- ☐ Yes
- ☐ No

6.2 R Code

```
# Load necessary library
library(dplyr)

# Load the data
data <- read.csv("mental_health.csv")

# 1. Ensure valid and unique column names
colnames(data) <- make.names(colnames(data), unique = TRUE)

# Display column names to verify exact names
print(colnames(data))

# 2. Change column names to shorter versions
colnames(data) <- c("timestamp", "age", "gender", "kulliyyah", "year_sem",
                    "credit_hours", "courses", "study_hours", "stress_lvl",
                    "sleep_hours", "rested_freq", "socialize_freq", "lonely", "acad_perf",
                    "miss_deadlines", "mh_awareness", "mh_usage", "phone_usage",
                    "exercise_freq", "social_activities", "extracurricular")

# 3. Handle missing values
# Remove rows with too many missing values (set threshold as needed)
threshold <- 5
data <- data[rowSums(is.na(data)) <= threshold, ]

# Impute remaining missing values with median for numeric columns
data <- data %>%
  mutate(
    age = ifelse(is.na(age), median(age, na.rm = TRUE), age),
    study_hours = ifelse(is.na(study_hours), median(study_hours, na.rm = TRUE),
study_hours),
    sleep_hours = ifelse(is.na(sleep_hours), median(sleep_hours, na.rm = TRUE),
sleep_hours),
    phone_usage = ifelse(is.na(phone_usage), median(phone_usage, na.rm = TRUE),
phone_usage),
    exercise_freq = ifelse(is.na(exercise_freq), median(exercise_freq, na.rm = TRUE),
exercise_freq)
  )

# 4. Convert range columns to numeric (age, study_hours, sleep_hours, socialize_freq,
phone_usage, exercise_freq)
data <- data %>%
  mutate(
    # Handle age with ranges and exact numbers
```

```

age = case_when(
  grepl("18 - 20", age) ~ 19,      # Midpoint of 18-20 range
  grepl("21 - 23", age) ~ 22,      # Midpoint of 21-23 range
  grepl("24 - 26", age) ~ 25,      # Midpoint of 24-26 range
  grepl("27 - 29", age) ~ 28,      # Midpoint of 27-29 range
  grepl("^\\d+$", age) ~ as.numeric(age), # Exact age as numeric
  TRUE ~ NA_real_
),
study_hours = case_when(
  study_hours == "Less than 2 hours" ~ 1,
  study_hours == "2-4 hours" ~ 3,
  study_hours == "4-6 hours" ~ 5,
  study_hours == "more than 6 hours" ~ 7,
  TRUE ~ NA_real_
),
sleep_hours = case_when(
  sleep_hours == "Less than 4 hours" ~ 3,
  sleep_hours == "4-6 hours" ~ 5,
  sleep_hours == "6-8 hours" ~ 7,
  sleep_hours == "More than 8 hours" ~ 9,
  TRUE ~ NA_real_
),
socialize_freq = case_when(
  socialize_freq == "1-2 times" ~ 1,
  socialize_freq == "3-5 times" ~ 3,
  socialize_freq == "More than 5 times" ~ 5,
  TRUE ~ NA_real_
),
phone_usage = case_when(
  phone_usage == "Less than 4 hours" ~ 2,
  phone_usage == "2-4 hours" ~ 3,
  phone_usage == "4-6 hours" ~ 5,
  phone_usage == "6-8 hours" ~ 7,
  phone_usage == "8-10 hours" ~ 9,
  phone_usage == "More than 10 hours" ~ 11,
  TRUE ~ NA_real_
),
exercise_freq = case_when(
  exercise_freq == "Never" ~ 1,
  exercise_freq == "1-2 times" ~ 2,
  exercise_freq == "3-5 times" ~ 4,
  exercise_freq == "More than 5 times" ~ 5,
  TRUE ~ NA_real_
)
)
)

```

5. Convert linear scale columns to 1-5 (rested frequency, missed deadlines)

```

data <- data %>%
  mutate(
    rested_freq = case_when(

```

```

    rested_freq == "Never" ~ 1,
    rested_freq == "Rarely" ~ 2,
    rested_freq == "Sometimes" ~ 3,
    rested_freq == "Often" ~ 4,
    rested_freq == "Always" ~ 5,
    TRUE ~ NA_real_
  ),
  miss_deadlines = case_when(
    miss_deadlines == "Never" ~ 1,
    miss_deadlines == "Rarely" ~ 2,
    miss_deadlines == "Sometimes" ~ 3,
    miss_deadlines == "Often" ~ 4,
    miss_deadlines == "Always" ~ 5,
    TRUE ~ NA_real_
  )
)

# 6. Encode categorical variables as factors
data <- data %>%
  mutate(
    gender = factor(gender),
    kulliyyah = factor(kulliyyah),
    year_sem = factor(year_sem)
  )

# 7. Feature engineering: Create stress index
data <- data %>%
  mutate(
    stress_index = 0.4 * stress_lvl + 0.3 * miss_deadlines + 0.3 * (5 - rested_freq)
  )

# 8. Remove duplicate rows
data <- data[!duplicated(data), ]

# 9. Ensure consistency in categorical variables (trim spaces, convert to lowercase)
data <- data %>%
  mutate(
    gender = tolower(trimws(gender)),
    kulliyyah = tolower(trimws(kulliyyah))
  )

# View cleaned and preprocessed data
head(data)

# 10. Descriptive Statistical Analysis
# Summary statistics for numeric columns
summary_stats <- data %>%
  summarise(
    mean_age = mean(age, na.rm = TRUE),
    median_age = median(age, na.rm = TRUE),

```

```

sd_age = sd(age, na.rm = TRUE),

mean_study_hours = mean(study_hours, na.rm = TRUE),
median_study_hours = median(study_hours, na.rm = TRUE),
sd_study_hours = sd(study_hours, na.rm = TRUE),

mean_sleep_hours = mean(sleep_hours, na.rm = TRUE),
median_sleep_hours = median(sleep_hours, na.rm = TRUE),
sd_sleep_hours = sd(sleep_hours, na.rm = TRUE),

mean_phone_usage = mean(phone_usage, na.rm = TRUE),
median_phone_usage = median(phone_usage, na.rm = TRUE),
sd_phone_usage = sd(phone_usage, na.rm = TRUE),

mean_exercise_freq = mean(exercise_freq, na.rm = TRUE),
median_exercise_freq = median(exercise_freq, na.rm = TRUE),
sd_exercise_freq = sd(exercise_freq, na.rm = TRUE)
)

print("Summary statistics:")
print(summary_stats)

# Identify outliers using the IQR method
outliers <- data %>%
  summarise(
    lower_bound_age = quantile(age, 0.25) - 1.5 * IQR(age),
    upper_bound_age = quantile(age, 0.75) + 1.5 * IQR(age),

    lower_bound_study_hours = quantile(study_hours, 0.25) - 1.5 * IQR(study_hours),
    upper_bound_study_hours = quantile(study_hours, 0.75) + 1.5 * IQR(study_hours),

    lower_bound_sleep_hours = quantile(sleep_hours, 0.25) - 1.5 * IQR(sleep_hours),
    upper_bound_sleep_hours = quantile(sleep_hours, 0.75) + 1.5 * IQR(sleep_hours)
  )

print("Outliers thresholds:")
print(outliers)

# Handle outliers by capping them at lower and upper bounds
data <- data %>%
  mutate(
    age = ifelse(age < outliers$lower_bound_age, outliers$lower_bound_age,
                 ifelse(age > outliers$upper_bound_age, outliers$upper_bound_age, age)),
    study_hours = ifelse(study_hours < outliers$lower_bound_study_hours,
                        outliers$lower_bound_study_hours,
                        ifelse(study_hours > outliers$upper_bound_study_hours,
                              outliers$upper_bound_study_hours, study_hours)),
    sleep_hours = ifelse(sleep_hours < outliers$lower_bound_sleep_hours,
                        outliers$lower_bound_sleep_hours,
                        ifelse(sleep_hours > outliers$upper_bound_sleep_hours,

```

```

outliers$upper_bound_sleep_hours, sleep_hours))
)

# Verify the cleaned data
print("Cleaned data preview:")
head(data)

# 12. Create the stress index and risk status column before splitting
data <- data %>%
  mutate(
    stress_index = 0.4 * stress_lvl + 0.3 * miss_deadlines + 0.3 * (5 - rested_freq),
    risk_status = ifelse(stress_index >= median(stress_index, na.rm = TRUE), "At Risk",
"Not At Risk")
  )

# Ensure target variable is a factor
data$risk_status <- factor(data$risk_status)

# 11. Data Splitting and Model Validation
# Split data into training (60%) and testing (40%) sets
set.seed(123) # Ensure reproducibility
sample_index <- sample(1:nrow(data), size = 0.6 * nrow(data))

train_set <- data[sample_index, ]
test_set <- data[-sample_index, ]

# Check the size of training and testing sets
cat("Training set size:", nrow(train_set), "\n")
cat("Testing set size:", nrow(test_set), "\n")

# 12. Model Development
# Convert stress_index into a binary target variable: "At Risk" or "Not At Risk"
data <- data %>%
  mutate(
    risk_status = ifelse(stress_index >= median(stress_index, na.rm = TRUE), "At Risk",
"Not At Risk")
  )

# Ensure target variable is a factor
data$risk_status <- factor(data$risk_status)

# Check class distribution
table(data$risk_status)

# 13. Decision Tree Model
library(rpart)
library(rpart.plot)

# Train a decision tree model

```

```

dt_model <- rpart(risk_status ~ study_hours + sleep_hours +
                  exercise_freq + miss_deadlines + rested_freq,
                  data = train_set, method = "class")

# Plot the decision tree
rpart.plot(dt_model, main = "Decision Tree for Mental Health Prediction")

# Predict on the test set
dt_predictions <- predict(dt_model, test_set, type = "class")

# Evaluate the model
dt_confusion_matrix <- table(test_set$risk_status, dt_predictions)
dt_accuracy <- sum(diag(dt_confusion_matrix)) / sum(dt_confusion_matrix)
cat("Decision Tree Accuracy:", dt_accuracy, "\n")

# 14. Naïve Bayes Model
library(e1071)

# Train a Naïve Bayes model
nb_model <- naiveBayes(risk_status ~ study_hours + sleep_hours +
                       exercise_freq + miss_deadlines + rested_freq,
                       data = train_set)

# Predict on the test set
nb_predictions <- predict(nb_model, test_set)

# Evaluate the model
nb_confusion_matrix <- table(test_set$risk_status, nb_predictions)
nb_accuracy <- sum(diag(nb_confusion_matrix)) / sum(nb_confusion_matrix)
cat("Naïve Bayes Accuracy:", nb_accuracy, "\n")

# 15. k-Nearest Neighbors (k-NN) Model
library(class)

# Normalize numeric columns for k-NN
normalize <- function(x) {
  return ((x - min(x)) / (max(x) - min(x)))
}

numeric_columns <- c("study_hours", "sleep_hours", "exercise_freq")
train_set_normalized <- as.data.frame(lapply(train_set[, numeric_columns], normalize))
test_set_normalized <- as.data.frame(lapply(test_set[, numeric_columns], normalize))

# Add target variable
train_set_normalized$risk_status <- train_set$risk_status
test_set_normalized$risk_status <- test_set$risk_status

# Apply k-NN (k = 5)
k <- 5
knn_predictions <- knn(train = train_set_normalized[, -ncol(train_set_normalized)],

```

```

test = test_set_normalized[, -ncol(test_set_normalized)],
cl = train_set_normalized$risk_status, k = k)

# Evaluate the model
knn_confusion_matrix <- table(test_set$risk_status, knn_predictions)
knn_accuracy <- sum(diag(knn_confusion_matrix)) / sum(knn_confusion_matrix)
cat("k-NN Accuracy (k =", k, "):", knn_accuracy, "\n")

# 16. Comparative Evaluation of Techniques
# Accuracy for Decision Tree
dt_accuracy <- sum(dt_predictions == test_set$risk_status) / nrow(test_set)

# Accuracy for Naïve Bayes
nb_accuracy <- sum(nb_predictions == test_set$risk_status) / nrow(test_set)

# Accuracy for k-NN
knn_accuracy <- sum(knn_predictions == test_set$risk_status) / nrow(test_set)

# Create results data frame
results <- data.frame(
  Model = c("Decision Tree", "Naïve Bayes", "k-Nearest Neighbors"),
  Accuracy = c(dt_accuracy, nb_accuracy, knn_accuracy)
)

# Print results
print("Comparative Evaluation of Models:")
print(results)

```