



Faculty of Computer Science and Information Technology

***A COMPARATIVE STUDY USING MACHINE LEARNING FOR
OBESITY PREDICTION***

Winnie Tiong Ru Ping

Bachelor of Computer Science with Honours
(Information Systems)

2025

**A COMPARATIVE STUDY USING MACHINE LEARNING FOR OBESITY
PREDICTION**

WINNIE TIONG RU PING

This project is submitted in partial fulfilment of the requirements for
the degree of Bachelor of Computer Science with Honours
(Information System)

Faculty of Computer Science and Information Technology
UNIVERSITY OF MALAYSIA SARAWAK (UNIMAS)

2025

**A COMPARATIVE STUDY USING MACHINE LEARNING FOR OBESITY
PREDICTION**

WINNIE TIONG RU PING

Projek ini merupakan salah satu keperluan untuk
Ijazah Sarjana Muda Sains Komputer dengan Kepujian
(Sistem Maklumat)

Fakulti Sains Komputer dan Teknologi Maklumat

UNIVERSITI MALAYSIA SARAWAK

2025

DECLARATION OF ORIGINALITY

I, WINNIE TIONG RU PING, with student matric number 81505, hereby declare that the work presented in this thesis is entirely my own. I have not used any sources, materials, or works from others without proper citation and acknowledgement. All references to published or unpublished literature, documents, or other sources are duly cited in the appropriate sections of this work. I understand the importance of academic integrity and the consequences of plagiarism. I affirm that the content of this work is the result of my own research, analysis, and creative efforts. I have faithfully followed the academic guidelines and ethical standards set by Faculty of Computer Science and Information Technology (UNIMAS) in producing this work.

ninnietrp

WINNIE TIONG RU PING (81505)

23rd June 2025

Faculty of Computer Science and Information Technology

University of Malaysia Sarawak

ACKNOWLEDGEMENT

I want to sincerely thank Dr. Stephanie Chua Hui Li, my supervisor, for her crucial guidance during this study. She guided me with her expertise in data for data mining. She also gave me a clear instruction and helped me stay on track. Her assistance enabled me to finish this project. I am also deeply grateful to Dr. Wang Yin Chai for sharing useful resources and detailed advice about the project. His insights gave me a clearer direction and helped me improve the quality of my work. I would want to express my gratitude to my friends and project group members for contributing their ideas and providing feedback during discussions. Their willingness to collaborate and share perspectives was truly helpful. I am thankful to my university, University Malaysia Sarawak for giving me access to the necessary tools, software, and learning resources that supported this project. The availability of these resources made the technical aspects of this work much easier to manage. Lastly, I would want to thank my family for their endless support. Their encouragement and understanding enabled me to overcome obstacles during the project.

TABLE OF CONTENTS

DECLARATION OF ORIGINALITY.....	i
ACKNOWLEDGEMENT.....	ii
LIST OF TABLES.....	vii
LIST OF FIGURES	viii
ABSTRACT	xi
ABSTRAK	xii
Chapter 1 : Introduction	1
1.1 Background	1
1.2 Problem Statement.....	2
1.3 Project Scope	3
1.4 Objectives.....	4
1.5 Methodology	4
1.5.1 Business Understanding	4
1.5.2 Data Understanding.....	4
1.5.3 Data Preparation.....	5
1.5.4 Modeling	5
1.5.5 Evaluation.....	5
1.5.6 Deployment.....	5
1.6 Significance of Project	6
1.7 Expected Outcome	6
1.8 Project Schedule.....	7

1.9	Thesis Outline.....	9
	Chapter 1: Introduction.....	9
	Chapter 2: Literature Review	9
	Chapter 3: Methodology	9
	Chapter 4: Data Mining Implementation.....	10
	Chapter 5: Result and Discussion	10
	Chapter 6: Conclusion and Future Works	10
1.10	Summary.....	10
	Chapter 2 : Literature Review	11
2.1	Introduction.....	11
2.2	Background	11
2.3	Machine Learning Algorithms.....	15
2.3.1	Decision Tree (DT).....	17
2.3.2	Random Forest (RF)	18
2.3.3	Support Vector Machines (SVM)	19
2.3.4	XGBoost.....	21
2.3.5	Logistic Regression (LR)	22
2.4	Related Work	23
2.5	Comparison of Related Work	29
2.6	Evaluation Metrics.....	35
2.6.1	Accuracy	35
2.6.2	Precision.....	35

2.6.3	Recall.....	36
2.6.4	F1-Score	36
2.6.5	Area Under ROC Curve (ROC-AUC)	37
2.7	Summary.....	39
Chapter 3 : Methodology.....		40
3.1	Introduction.....	40
3.2	Cross Industry Standard Process for Data Mining (CRISP-DM).....	40
3.2.1	Business Understanding	41
3.2.2	Data Understanding.....	42
3.2.3	Data Preparation.....	45
3.2.4	Modeling	47
3.2.5	Evaluation.....	48
3.2.6	Deployment.....	49
3.3	Summary.....	50
Chapter 4 : Data Mining Implementation		51
4.1	Introduction.....	51
4.2	Cross Industry Standard Process for Data Mining (CRISP-DM).....	52
4.2.1	Data Understanding.....	52
4.2.2	Data Preparation.....	57
4.2.3	Modeling	66
4.2.4	Evaluation.....	68
4.2.5	Deployment.....	70

4.3	Summary.....	75
Chapter 5 : Result and Discussion.....		76
5.1	Introduction.....	76
5.2	Results	76
5.2.1	Result for Decision Tree.....	76
5.2.2	Result for Random Forest	77
5.2.3	Result for Support Vector Machine.....	79
5.2.4	Result for XGBoost	80
5.2.5	Result for Logistic Regression	82
5.3	Comparison of Results.....	83
5.4	Summary.....	87
Chapter 6: Conclusion and Future Works.....		88
6.1	Introduction.....	88
6.2	Contributions.....	88
6.3	Limitations.....	88
6.4	Future Woks	89
6.5	Conclusion	90
References.....		91
APPENDICES.....		94
Appendix A: Turnitin Report.....		94

LIST OF TABLES

Table 2.1: Summary of Related Works	29
Table 3.1: Dataset Attributes and their Data Type and Descriptions.....	43
Table 3.2: Sample of Binary Class Confusion Matrix	48
Table 3.3: Description of the Combinations of Predicted and Actual Values	49
Table 4.1: A Summary of Software Tools, Libraries, and their Purposes.....	51
Table 4.2: Sample Label Encoding Mappings for Selected Features	58
Table 4.3: Hyperparameter Search Space and Optimal Values for Each Classification Model.....	66
Table 5.1: Performance of DT Model Before and After Tuning on Test Set.....	76
Table 5.2: Performance of RF Model Before and After Tuning on Test Set	78
Table 5.3: Performance of SVM Model Before and After Tuning on Test Set	79
Table 5.4: Performance of XGBoost Model Before and After Tuning on Test Set....	81
Table 5.5: Performance of LR Model Before and After Tuning on Test Set.....	82
Table 5.6: Summary of Performance Comparison After Tuning on Test Set.....	86

LIST OF FIGURES

Figure 1.1: Brief Flowchart of CRISP-DM Methodology	6
Figure 1.2: Overall Gantt Chart of the Project.....	7
Figure 1.3: Overall Gantt Chart of the Project.....	8
Figure 2.1: Overview of Obesity Level among Female and Male (Oldham & Robinson, 2018).....	11
Figure 2.2: Types of Machine Learning	15
Figure 2.3: The Architecture of the Decision Tree Model (F. Wang et al., 2024).....	18
Figure 2.4: The Architecture of the Random Forest Model (Li, 2023)	19
Figure 2.5: The Architecture of the Support Vector Machines Model (F. Wang et al., 2024)	21
Figure 2.6: The Architecture of the XGBoost Model (Y. Wang et al., 2019).....	22
Figure 2.7: The Architecture of the Logistic Regression Graph.....	23
Figure 2.8: AUC Graph	38
Figure 3.1: CRISP-DM Methodology Flowchart.....	41
Figure 4.1: Portion of Data Description of the Dataset from Kaggle.....	53
Figure 4.2: Code Snippet for Initial Dataset Exploration	54
Figure 4.3: Pie Chart showing the Distribution of Obesity Level Classes	55
Figure 4.4: Checking Missing Values and Output Summary	55
Figure 4.5: Code Snippet for Checking Handling Duplicated Rows and Output Summary.....	56
Figure 4.6: Checking Outlier Detection and Output Summary	56
Figure 4.7: Checking Inconsistencies and Output Summary	57
Figure 4.8: Cleaned Obesity Dataset.....	60
Figure 4.9: Corresponding Bar Plot for Information Gain.....	61
Figure 4.10: Corresponding Bar Plot for Gain Ratio	62

Figure 4.11: Corresponding Heatmap for Correlation Matrix.....	63
Figure 4.12: Corresponding Bar Plot for ReliefF	64
Figure 4.13: Final List of Selected Features (After Filtering).....	65
Figure 4.14: Train-Test Split with Stratified Sampling (80:20)	66
Figure 4.15: Code Snippet for Computing Accuracy Before and After Hyperparameter Tuning for Both Training and Testing Dataset	68
Figure 4.16: Code Snippet for Generating Classification Reports for each Model After Tuning	69
Figure 4.17: Code Snippet for Generating and Displaying Confusion Matrix for each Model.....	69
Figure 4.18: Code Snippet for Calculating and Plotting ROC curves with AUC Scores Per Class After Tuning.....	70
Figure 4.19: Code Snippet for Saving the Trained Random Forest Model using the Joblib Library	71
Figure 4.20: Web application for User Input Real-Time Prediction	72
Figure 4.21: Output Display with Prediction Category and User Input Summary Section	73
Figure 4.22: Output Display with Probabilities	74
Figure 5.1: Confusion Matrix and ROC Curve for the Tuned DT Model on the Test Set	77
Figure 5.2: Confusion Matrix and ROC Curve for the Tuned RF Model on the Test Set	79
Figure 5.3: Confusion Matrix and ROC Curve for the Tuned SVM Model on the Test Set	80
Figure 5.4: Confusion Matrix and ROC Curve for the Tuned XGBoost Model on the Test Set	81

Figure 5.5: Confusion Matrix and ROC Curve for the Tuned LR Model on the Test Set	83
Figure 5.6: Overall Comparison of Models Performance Metrics	86
Figure 5.7: Averaged ROC Curves of All Models After Hyperparameter Tuning.....	87

ABSTRACT

Machine learning (ML) has proven effective in various fields, including healthcare, for tasks like obesity prediction, personalized medicine, and risk assessment. Cardiovascular diseases and diabetes are linked to obesity, a condition that is becoming a major worldwide health problem. Its increasing prevalence is driven by unhealthy diets, sedentary lifestyles, and genetic predispositions. Due to they do not consider many factors, including lifestyle and environmental impacts, traditional evaluation tools, such as the Body Mass Index (BMI) are insufficient. This study proposes an obesity prediction system using ML techniques to enhance accuracy in detecting and classifying obesity levels. The research employs the CRISP-DM methodology, encompassing stages such as business understanding, data understanding, data preparation, modeling, evaluation, and deployment. The dataset, obtained from Kaggle, includes diverse features related to basic information, eating habits, and physical conditions. Data transformation, feature scaling, and feature selection are examples of data preprocessing that make sure the dataset is appropriate for machine learning models. Five algorithms such as Decision Trees, Random Forest, Support Vector Machines (SVM), XGBoost, and Logistic Regression are used and evaluated for their performance in classifying obesity into categories aligned with WHO standards. Advanced methods like adjusting hyperparameters help to ensure the optimal performance of the models. Metrics such as accuracy, precision, recall, F1-score, and ROC-AUC are used to assess model performance and select the optimal method. Random Forest had the best performance overall. It gave the highest accuracy and balanced results across all metrics. The best model was used to build a web application. This app allows users to enter their data and get their obesity risk in real-time. The system helps people understand their health risks. It also supports doctors in giving faster assessments.

ABSTRAK

Pembelajaran mesin (ML) terbukti berkesan dalam pelbagai bidang, termasuk penjagaan kesihatan, bagi tugas seperti ramalan obesiti, perubahan peribadi, dan penilaian risiko. Penyakit kardiovaskular dan diabetes sering dikaitkan dengan obesiti, yang kini menjadi isu kesihatan utama global. Peningkatan kadar obesiti dipengaruhi oleh pemakanan tidak sihat, gaya hidup tidak aktif, dan faktor genetik. Alat tradisional seperti Indeks Jisim Badan (BMI) tidak mencukupi kerana tidak mengambil kira faktor gaya hidup dan persekitaran. Kajian ini mencadangkan sistem ramalan obesiti menggunakan teknik ML untuk meningkatkan ketepatan pengesanan dan klasifikasi tahap obesiti. Pendekatan CRISP-DM digunakan, melibatkan fasa seperti pemahaman masalah, penyediaan data, pemodelan, penilaian dan pelaksanaan. Dataset diperoleh dari Kaggle, merangkumi maklumat asas, tabiat pemakanan, dan keadaan fizikal. Proses prapemprosesan seperti transformasi data, penskalaan dan pemilihan ciri dijalankan bagi memastikan kesesuaian data. Lima algoritma ML digunakan seperti Pokok Keputusan, Hutan Rawak, SVM, XGBoost dan Regresi Logistik. Model dinilai berdasarkan metrik seperti ketepatan, kepekaan, ketepatan semula, skor F1 dan ROC-AUC, dengan pelarasan hiperparameter untuk prestasi optimum. Hutan Rawak menunjukkan prestasi terbaik secara keseluruhan dengan ketepatan tertinggi dan metrik yang seimbang. Model terbaik ini dibangunkan dalam aplikasi web mesra pengguna, membolehkan pengguna menerima penilaian risiko obesiti masa nyata. Sistem ini membantu pengguna memahami risiko kesihatan dan menyokong doktor membuat penilaian pantas dan berkesan.

Chapter 1 : Introduction

1.1 Background

In today's digital age, maintaining balanced nutrition is essential for good health and a high quality of life. Advances in science and technology have helped reduce malnutrition by improving access to nutritious food and healthcare. However, these developments have also contributed to a rise in overnutrition and excessive calorie intake that can lead to obesity. The World Health Organisation (WHO) defines obesity and overweight as the accumulation of excessive fat in specific body areas that can potentially harm one's health in the domain of health and lifestyle studies (Kumar et al., 2024). This condition is classified as a chronic noncommunicable disease (NCD) that can affect individuals at any stage of life. Obesity is associated with several serious health issues, such as diabetes, heart disease, and some types of cancer, all of which reduce life quality, raise medical expenses, and hasten mortality. For example, approximately 7% of certain cancers worldwide, 23% of heart disease cases, and over 44% of diabetes cases globally are caused by obesity (Kato et al., 2012).

Due to obesity is influenced by complex factors such as lifestyle, genetics, and environment, this makes it a multifaceted issue that requires both comprehensive prevention and early detection. Preventative measures may involve lifestyle changes like improving diet and increasing physical activity. Early prediction of obesity is important because it helps identify individuals at risk before severe health issues arise. Body Mass Index (BMI) is one easy way to measure obesity. A person's height and weight are used by BMI to calculate their body fat percentage. The WHO defines overweight as having a BMI of 25 to 29.9 and obesity as having a BMI of 30 or above (Alam et al., 2023).

In addition, BMI varies by gender, with significant differences observed in obesity trends between men and women. According to recent statistics from the NCD Risk Factor

Collaboration (NCD-RisC), by 2025, the worldwide obesity rate is expected to surpass 21% for women and reach 18% for men. Additionally, about 6% of men and 9% of women are predicted to suffer from extreme obesity (Barzinji et al., 2021). Currently, more than two billion people worldwide are already classified as overweight or obese (Koklu & Sulak, 2024). The increasing number of obesity cases shows that it is becoming a major health concern worldwide.

As a result, along with BMI testing, new technologies such as machine learning and data mining technique have the potential to make a big influence. These tools can analyze various obesity risk factors such as genetic traits, lifestyle habits, and environmental factors. This allows them to identify individuals at risk and create tailored prevention strategies. Nowadays, health research frequently uses data mining and machine learning approaches to help with risk prediction and diagnosis. These methods have also become popular in studying the risk of obesity (Devi et al., 2022).

Given these advancements, the goal of this research project is to use machine learning methods to provide real-time predictions about obesity risk. The study will train and compare various machine learning models by using known obesity data to determine the most accurate model. The best-performing model will then be integrated into a user-friendly web application. This application will allow users to input their personal data to receive a personalized assessment of their obesity risk. This innovative approach integrates health and technology to provide a proactive solution for combating obesity.

1.2 Problem Statement

Globally, obesity is a serious and expanding health issue. In 2016 alone, more than 1.9 billion adults were affected by obesity with over 650 million of them classified as obese (He, 2022). Obesity is not only on the rise among adults but is also steadily increasing in many countries. This trend begins in childhood and continues into adolescence. In fact, this condition

is more dangerous than being underweight since it is linked to serious health problems like diabetes, heart disease, and early death (Alam et al., 2023). Several factors contribute to obesity including stress, lack of physical exercise, doing less physically demanding work, and consuming contemporary meals that are lacking in nutrition and heavy in calories (Fernandes et al., 2023). However, obesity is often not identified early enough to take effective action. Historically, many online tools that assess obesity use Body Mass Index (BMI), which is helpful but has limitations. BMI does not consider other important factors such as family history, lifestyle habits, and environmental influences which can greatly affect a person's obesity risk (Celik et al., 2021). As a result, many people at risk of obesity may not be properly identified. This causes missed opportunities for early intervention. However, recent studies have shown that these variables might significantly affect the risk of obesity (Safaei et al., 2021). Therefore, this project aims to solve this problem by developing a more complete obesity prediction tool that includes additional factors. The tool will offer a more accurate prediction of obesity and enable earlier prevention efforts. This will help individuals and healthcare providers manage obesity more effectively.

1.3 Project Scope

The scope of this project covers a data analysis, model training, and model development to raise awareness about obesity and enhance risk assessment. The dataset will focus on factors contributing to obesity and will be sourced from public and free platforms like Kaggle. Then this project will provide a training and testing on different machine learning models to build the best model to predict obesity. A simple web application will then build to demonstrate the result. Although the project focuses on predicting obesity, it will not provide specific health, diet, or medical advice. However, the outcome is limited to prediction, not recommendation. Its focus remains on prediction accuracy and raising awareness of obesity risks rather than addressing

individual health management or treatment. The predictions are based on available data and should not replace professional medical assessments or advice.

1.4 Objectives

The objectives of this project are as follows:

- To determine the best feature set for obesity prediction.
- To compare the performance of different machine learning models in obesity prediction.
- To determine the best machine learning model for obesity prediction based on performance metrics.

1.5 Methodology

The approach used for this research and development project is the Cross Industry Standard Process for Data Mining (CRISP-DM). The CRISP-DM model outlines six steps which are business understanding, data understanding, data preparation, modeling, evaluation, and deployment (Hayat Suhendar & Widayani, 2023). The whole CRISP-DM structure is seen in Figure 1.1.

1.5.1 Business Understanding

The project's initial phase focusses on analysing the problem from a business perspective before turning it into data mining tasks. First, a project plan and timeline are proposed and established to keep the project on track. Decisions are also made about the tools and techniques that will be used throughout the project. Next, it is crucial to define the prediction goals clearly. This means focusing on what the project is truly aiming to achieve and consider any assumptions or challenges that might arise.

1.5.2 Data Understanding

In this phase, the dataset will be downloaded from an open-source platform like Kaggle. Initial exploration and filtering will be conducted to better understand the data's characteristics

and relevance for our prediction objectives. This process includes assessing data quality, identifying and removing noisy data, duplicates, or inconsistent values, and exploring relationships between features to ensure the dataset aligns with our goals.

1.5.3 Data Preparation

Data preparation, often referred to as data preprocessing, is an important step in the machine learning process. Raw data must be cleaned, transformed, and organised in this phase. Feature selection helps in identifying the most important features and eliminates unnecessary or irrelevant variables to minimise noise and improve model performance. Additionally, the dataset is also divided into training and testing sets.

1.5.4 Modeling

In this step, machine learning models are chosen and used to the prepared dataset with the objective of predicting obesity levels. The models will undergo adjustments to enhance their accuracy and generalization. Standard machine learning tools will be utilized to ensure a structured, consistent, and reliable approach to training and testing the models.

1.5.5 Evaluation

Following training, the models' performance will be assessed using relevant metrics like ROC-AUC, F1-score, recall, accuracy, and precision. These measures show how effectively each model predicts obesity levels in the test set. Prediction results for all obesity levels will be displayed in a confusion matrix. It counts the correct and incorrect predictions. Also, comparing many models will also be part of this step to determine which one performs the best. The evaluation will ensure the final model fits the project's goals to predict obesity accurately.

1.5.6 Deployment

In this last step, the best model will deploy into a web application for demonstration purpose. This simple web application will accept user inputs and provide real-time obesity

predictions. The prediction model will serve as the backend of the web application to determine whether the user is classified as obese based on their input.

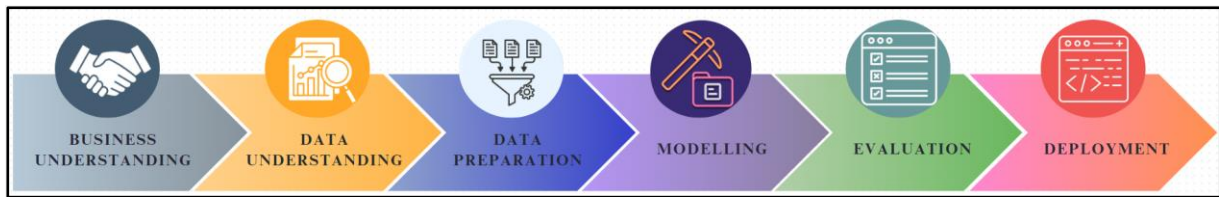


Figure 1.1: Brief Flowchart of CRISP-DM Methodology

1.6 Significance of Project

The significance of this project is to determine the most effective machine learning model for predicting obesity which will provide useful insights into the most accurate techniques. Therefore, by offering precise obesity risk assessments, the project can help individuals better understand their risk for obesity-related conditions like cardiovascular disease and diabetes. Moreover, the findings from this project will be instrumental in supporting the healthcare professionals by aiding early intervention and obesity crisis management. As a result, this project has the potential to increase public knowledge and understanding about their chance of having an obesity within communities.

1.7 Expected Outcome

The expected outcome of this project is to study different machine learning models for predicting obesity levels. Then, the project aims to evaluate and compare the performance of different machine learning models work in predicting obesity levels. It focuses on the important features that affect these predictions. Therefore, the best-performing model will be selected for implementation in a web-based application. This application will allow users to input relevant data, such as age, height, weight, physical activity level, and dietary habits to predict their obesity level in real-time and accurately.

1.8 Project Schedule

The project schedule serves as a tool to monitor task completion by including key details such as start dates, finish dates, and task durations.

Therefore, this project uses a Gantt Chart for a visual representation of the timeline and progress to track the completion of the project.

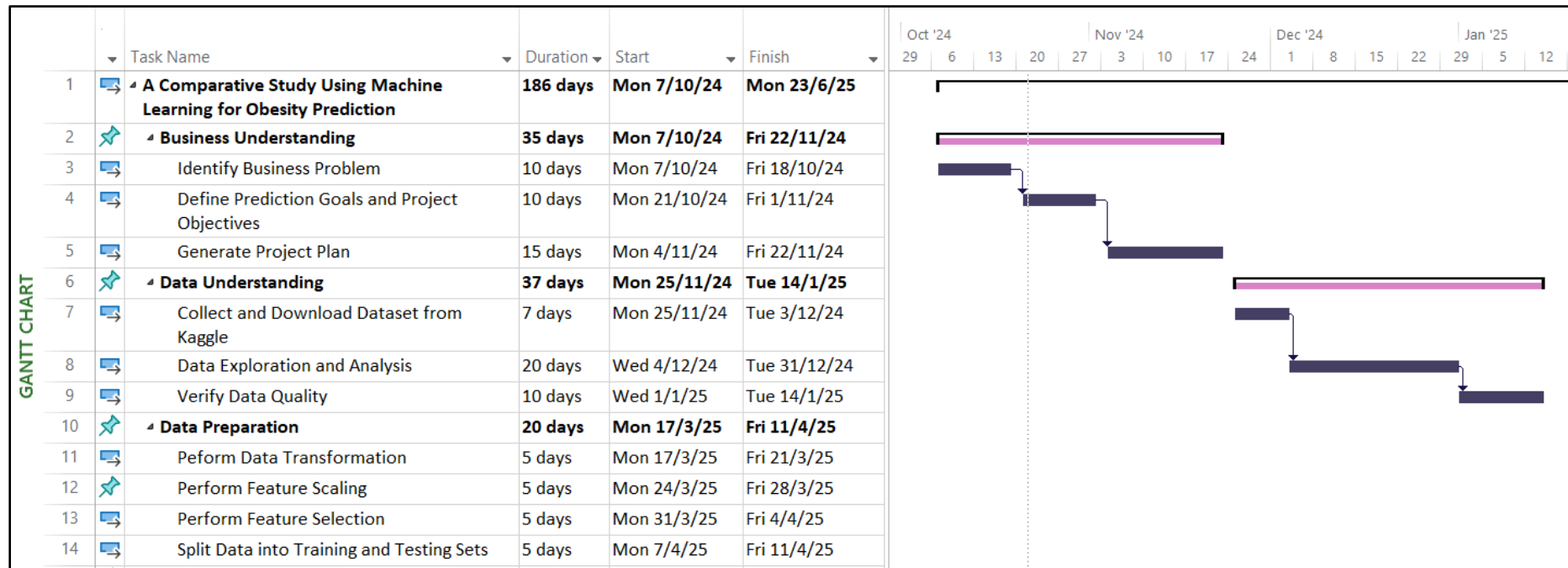


Figure 1.2: Overall Gantt Chart of the Project

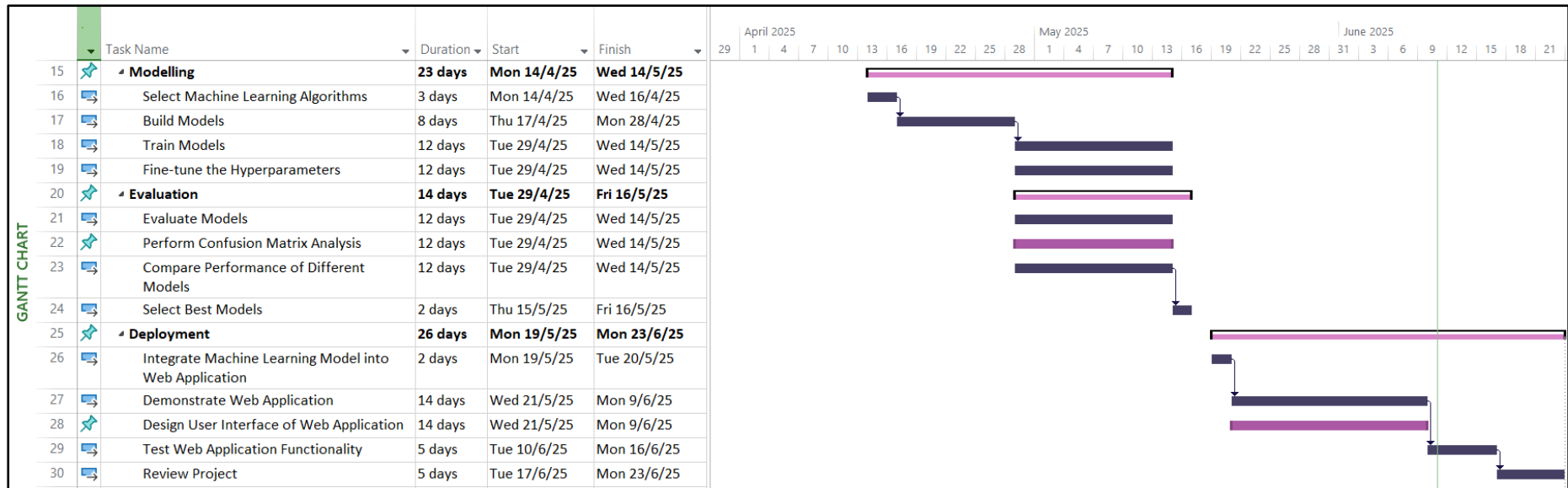


Figure 1.3: Overall Gantt Chart of the Project

1.9 Thesis Outline

Chapter 1: Introduction

The first chapter gives an overview of the project. It starts with the problem statement to explain the main issue that the project is about. It then presents the objectives of the project, followed by the scope, which defines the study's focus and limitations. A summary of the methodology explains the approach and tools used. The significance of the project shows its potential impact. The project schedule lists key milestones. The expected outcome is clear and achievable. The chapter concludes with a thesis outline to offer a roadmap of the content to follow and a summary that highlights the main ideas of the chapter.

Chapter 2: Literature Review

This chapter gives a detailed review of new progress in predicting obesity with machine learning. It begins by providing background information on obesity and machine learning techniques. The chapter explores various algorithms to assess their application and effectiveness in obesity prediction. A comparative analysis of related studies is presented, highlighting gaps and opportunities for improvement in existing methodologies. The evaluation metrics used in these studies are also discussed. This chapter establishes the foundational knowledge required to guide the current project.

Chapter 3: Methodology

This chapter describes the methodology used in this research, based on the CRISP-DM framework. It details the steps involved, starting with understanding the project objectives and preparing the dataset sourced from Kaggle. Data preprocessing ensures the data is optimized for model training. Five machine learning algorithms are implemented and compared. The chapter also describes the use of hyperparameter tuning to refine model performance. Finally, evaluation techniques and metrics are used to check how well the predictive models work.

Chapter 4: Data Mining Implementation

This chapter covers the steps in implementing the project, from data preparation to build a prototype web application. It starts with data preparation to make the dataset ready for training machine learning models. Then, different machine learning techniques are trained and tested to find the most accurate approach for predicting obesity. Finally, the best model is integrated into a prototype web application to show how it works.

Chapter 5: Result and Discussion

In this chapter, the results of the model evaluation will be discussed in detail. The performance of different machine learning models will be compared using evaluation metrics. The most effective model for predicting obesity will be identified. The chapter will explain why certain models performed better and the importance of the selected features in predicting obesity. It will also explore at how the findings are similar to or different from other studies in the field.

Chapter 6: Conclusion and Future Works

This chapter will summarize the key findings. It will talk about the best machine learning model for predicting obesity and how it can be used in healthcare. It will also mention the contribution of the study to improve obesity risk assessments and discuss the limitations of the research. Lastly, the chapter will propose areas for future research.

1.10 Summary

To conclude, a predictive obesity tool is proposed and introduced to be used on predict based on individual data. This project aims to determine the most effective model by comparing various algorithms and then incorporating the best-performing model into a web application. As a result, this project will help increase awareness of obesity risks and offer a practical tool for early risk assessment.

Chapter 2 : Literature Review

2.1 Introduction

This chapter reviews the work done by researchers from 2021 to 2024 that focuses on different techniques used to predict obesity. It begins by providing essential background information on obesity, and machine learning, including its key approaches and algorithms. This foundation is crucial for understanding the important theories, concepts, methodologies, and existing findings related to the research question. A comparative analysis of the selected research papers highlights their contributions and findings to offer insights that can inform further studies in this field.

2.2 Background



Figure 2.1: Overview of Obesity Level among Female and Male (Oldham & Robinson, 2018)

Obesity has become a worldwide issue that puts people's lives at risk by impacting nearly every organ system. It has become a major public health concern and is one of the popular non-communicable diseases (Ruze et al., 2023). Obesity is defined as when a person has unhealthy or excessive body fat that increases health risks. It is usually identified when someone has a BMI of 30 or higher. The WHO states that obesity is one of the leading causes of death and disability. It is the fourth most common risk factor for non-communicable diseases, after high blood pressure, unhealthy diets, and tobacco use (Koklu & Sulak, 2024). The rapid increase in obesity worldwide highlights the urgent need for solutions that bring together scientific understanding and practical actions. Once we understand the factors and effects of obesity, we can take steps to address it effectively.

However, obesity is also a complex condition with both risk factors that can and cannot be changed. Genetics is one of the unchangeable risk factors which means that some people may be more genetically predisposed to obesity. However, there are many factors that we can control. Lack of physical activity, unhealthy lifestyle choices, and environmental influences all play a significant role in obesity (Al-Omar et al., 2024). With the advancement of technology, many people spend more time on computers, watching TV, and using electronic devices. This evolution of technology has resulted in a decrease in physical activity and coupled with the ease of driving rather than walking has caused reduced exercise levels and increased weight gain. Furthermore, maintaining a healthy weight is also challenging for people mainly because high-calorie foods, larger portions and sugary drinks are everywhere. On top of these factors, stress also contributes to obesity. This is because many people overeat or take bad diets as a way of dealing for their stress, which leads to a chain reaction that is difficult to be overcome (Bellaver et al., 2024). This combination of factors has made it harder for many to maintain a balanced lifestyle and healthy body weight no matter is adult or children and male or female.

As a result, this trend is supported by alarming statistics on obesity. Obesity rates have increased dramatically since 1990, with a troubling increase in both adults and children worldwide. For adults, obesity rates have doubled, and for teenagers, they have quadrupled. By 2025, it is estimated that 2.7 billion adults will be overweight, with 1 billion facing obesity and 177 million suffering from severe obesity (Class III). Among children and adolescents, obesity rates have also risen significantly, from 8% to 20% globally. Currently, around 37 million children under the age of 5 are struggling with obesity (Al-Omar et al., 2024). Moreover, obesity is more common in women than in men. Projections indicate that by 2030, about 14% of men and 20% of women, which equals over 1 billion people globally will be affected by obesity. Among these individuals, 18% will fall into the general obese category, with a BMI of 30 or higher. Additionally, 6% will be classified as severely obese, with a BMI of 35 or more, and 2% will be morbidly obese, with a BMI of 40 or greater (Ruze et al., 2023).

The rising trend in obesity has a broad effect that goes beyond just health concerns. It also affects economic stability and quality of life to meet growing demands. For example, it is closely linked to cardiovascular conditions such as heart failure, coronary artery disease, and stroke. Research shows that for every 5 kg/m² increase in BMI, the risk of coronary heart disease rises by 27%, and the risk of stroke increases by 18% (Lam et al., 2023). Moreover, the economic impact of obesity is equally important. Global costs associated with obesity are estimated at \$1.96 trillion and are estimated to rise to \$4 trillion by 2035 (Al-Omar et al., 2024). The costs of obesity arise from increased medical care and specialized treatments, such as weight-loss surgeries and long-term medications. These expenses create a heavy financial burden on people, governments, and healthcare systems by making it harder to manage their limited resources. Beyond physical and economic challenges, obesity also brings a bad things to people's emo health. People with obesity often experience social stigma, which can lower their self-esteem and dissatisfaction with their bodies. This is especially difficult for women,

who are often pressured by society to meet unrealistic beauty standards (Koklu & Sulak, 2024; Lam et al., 2023). The stigma does not only affect personal feelings but can also create challenges at work. Many individuals with obesity may experience discrimination in the workplace or encounter obstacles to advance in their careers. Thus, these issues show how obesity can deeply impact both emotional health.

According to Tee & Voon (2024), the 2023 National Health and Morbidity Survey in Malaysia revealed that 54.4% of the population is either overweight or obese, with 21.8% classified as obese. This concerning statistic highlights the urgency of addressing obesity as a national health priority. So, Malaysia has set up many policies and programs to lower obesity rates and help people live healthier lives. One of the proposed strategies is to make healthy BMI as a performance metric to encourage individuals to sustain good weight. Another initiative focuses on limiting the operating hours of food outlets to reduce access to unhealthy food options, particularly late at night. Also, the Ministry of Health is working hard to make people more aware of the dangers of obesity. This includes promoting the reduction of sugar intake and high-calorie foods, as well as offering dietary guidance to encourage healthier eating habits. Encouraging people to be more active and reduce sedentary behavior is essential for improving overall health. Moreover, regular physical movement not only helps maintain a healthy weight but also enhances cardiovascular fitness and mental health. Studies have shown that physical activity can account for 25% to 50% of the energy used each day, highlighting its importance in daily routines (Koklu & Sulak, 2024). For this reason, early interventions to prevent and manage obesity can significantly benefit individuals and reduce healthcare costs.

Lastly, early prediction of obesity risk is crucial for implementing timely interventions to prevent the condition from becoming harder to manage. Machine learning is a strong tool in healthcare. It gives new ways to deal with problems like obesity and being overweight. Research has shown its expanding role in identifying and managing obesity cases by utilizing

large datasets that include various factors like lifestyle habits and physical characteristics (Shakti et al., 2024). These algorithms analyze complex data, uncover patterns and relationships that traditional diagnostic methods might miss. This allows for more accurate predictions of obesity risk. Furthermore, machine learning models evolve and improve over time. They refine their accuracy by processing new data and updating their algorithms. This adaptability not only makes predictions more precise but also deepens our understanding of the factors contributing to obesity. As a result, machine learning is increasingly valued as a key tool for supporting obesity prevention efforts and enhancing overall healthcare outcomes (Shakti et al., 2024).

2.3 Machine Learning Algorithms

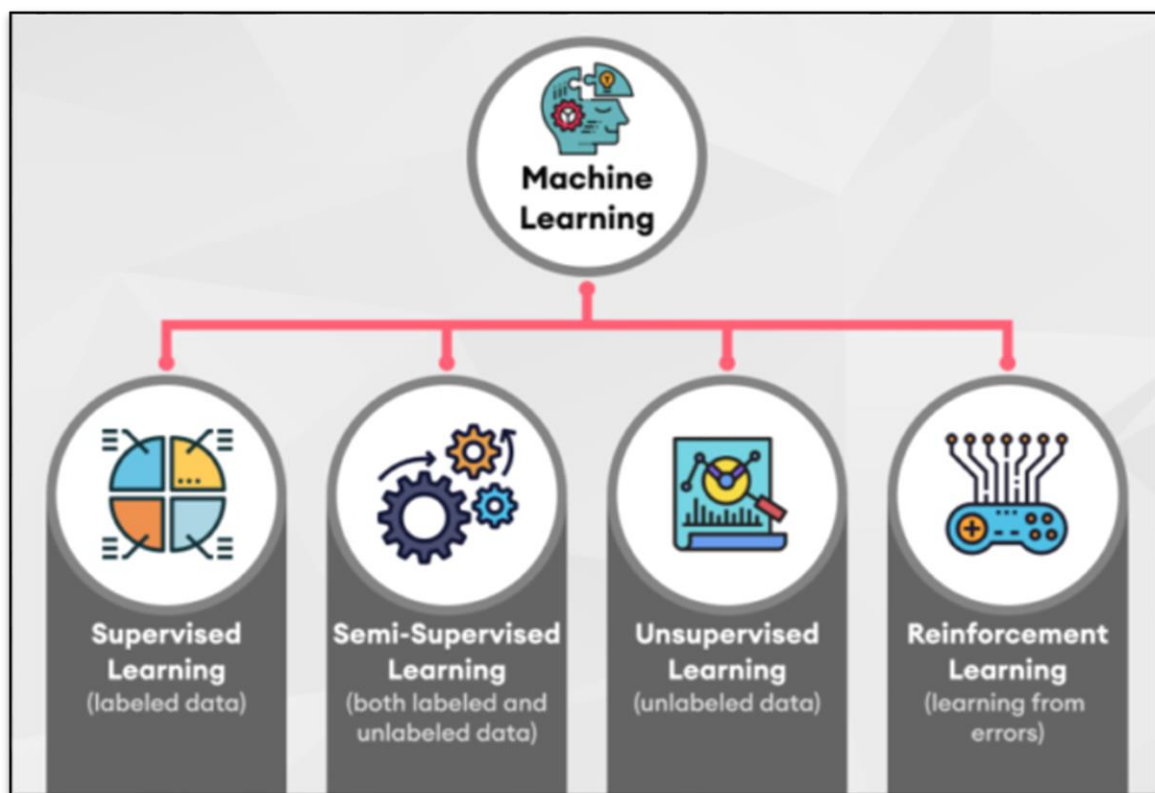


Figure 2.2: Types of Machine Learning

Machine learning (ML) is a field of study focused on creating computational methods that analyze large datasets to identify patterns and build predictive models (Rodríguez et al., 2021). ML combines knowledge with artificial intelligence (AI) from several areas, such as

probability, statistics, psychology, and neurobiology which allow it to apply to many different fields. ML allows computers to learn from data so that they can make better predictions or decisions without needing to be specifically programmed for every task (Hayat Suhendar & Widyani, 2023). Over time, ML has developed from basic statistical methods and pattern recognition into more complex models. As ML algorithms process larger amounts of data, they become more accurate and improve their ability to make reliable predictions (Hayat Suhendar & Widyani, 2023). They are also particularly effective at identifying complex relationships, such as the connections between various risk factors and obesity.

ML includes several types of learning approaches. In supervised learning, algorithms use labeled data to predict specific outcomes. This approach requires datasets that include both inputs and expected outputs to train the model effectively. Supervised learning can be categorized into classification and regression tasks. Classification involves grouping data into categories based on patterns found in the data such as Decision Tree, Random Forest, Support Vector Machines, and K-Nearest Neighbors. Furthermore, regression focuses on understanding how variables are related to each other to make predictions. Rasheed & Glob (2022) found that common regression techniques are linear regression, logistic regression, and polynomial regression.

Unlike supervised learning, unsupervised learning does not require labeled data. Instead, the data on its own to identify patterns or structures was analyze by algorithm (Rasheed & Glob, 2022). This makes it especially useful when experts are unsure about the underlying characteristics of the dataset. It is often applied to tasks like clustering, where data is grouped based on similarities and association, which looks for relationships between variables. Common algorithms used in unsupervised learning include K-means clustering, hierarchical clustering, Gaussian mixture models, and dimensionality reduction techniques like principal component analysis (PCA) and t-SNE.

Moreover, semi-supervised learning combines both labeled and unlabeled data to improve efficiency. The labeled data helps guide the learning process while making use of the large amount of unlabeled data. In contrast, reinforcement learning is another approach where algorithms learn by trial and error. They interact with their environment and receive feedback in the form of rewards or penalties. Over time, this feedback helps them improve their decision-making (Rasheed & Glob, 2022).

Therefore, several supervised learning algorithms such as linear classifiers, support vector machines, logistic regression, decision trees, random forests, K-nearest neighbors, artificial neural networks, and XGBoost are commonly used in healthcare and other fields (Khater et al., 2024).

2.3.1 Decision Tree (DT)

DT is a supervised machine learning method that can work with both numerical and categorical data. It uses a tree-like structure to represent decisions, possible outcomes, and their consequences (Kolli et al., 2023). This approach is highly effective for identifying key factors that influence specific outcomes. DT are interpreted easily by classification and regression tasks. For example, they have been applied to analyze obesity levels and offer actionable insights based on the findings (Shakti et al., 2024).

The structure of a decision tree consists of nodes, branches, and leaves. Nodes represent the attributes being evaluated for classification. Branches indicate the possible values of these attributes. Leaves show the final decisions or outcomes. The root node is located at the top of the tree and serves as the starting point. Branches connect the root node to the leaves to form the complete structure (Kumar et al., 2024). Building a decision tree begins by selecting the root node based on specific metrics. Information Gain is one metric that commonly used which measures how much uncertainty or entropy is reduced when the data is split at a particular node. Entropy calculates level of uncertainty or impurity in the dataset, while methods such as Gini

impurity help further reduce disorder during the splitting process. The tree-building process continues by dividing the data into subsets at each level. This is done by choosing the best attribute for splitting until conditions like maximum tree depth or minimum sample size are met (Islam et al., 2024).

Not only that, but various algorithms have also been developed to optimize decision tree performance and minimize prediction errors. These include ID3, C4.5, CART, SLIQ, CHAID, and Random Forest. DT are easy to understand because their logical structure visually represents the influence of different features on the outcome. DT can handle both continuous and discrete values effectively. DT also manage missing data without significant performance issues (Islam et al., 2024). This combination of simplicity and flexibility makes decision trees a practical solution for predictive problems (F. Wang et al., 2024)

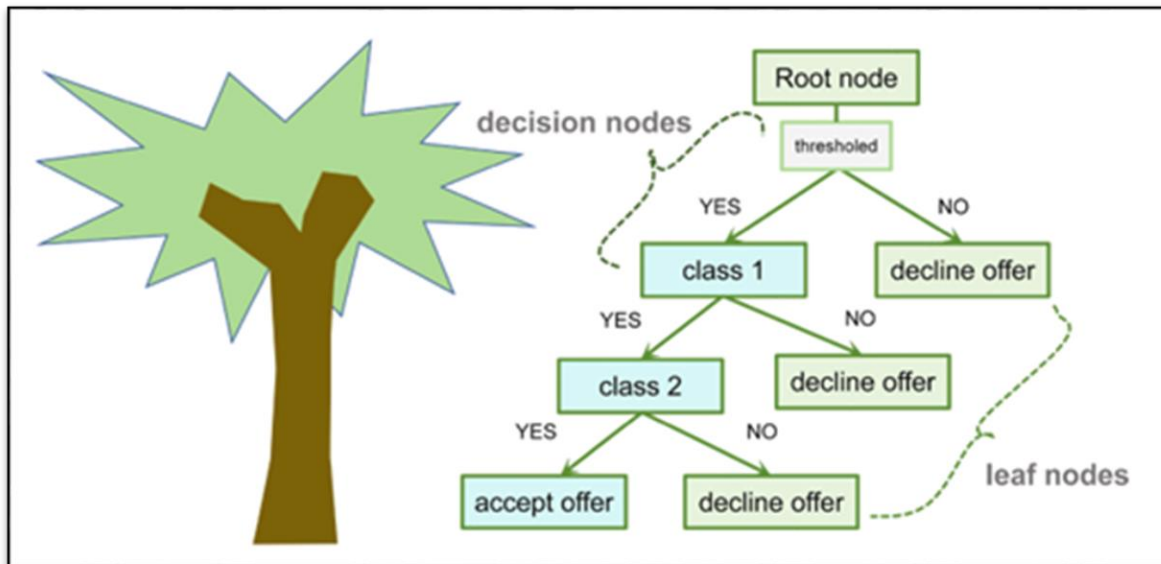


Figure 2.3: The Architecture of the Decision Tree Model (F. Wang et al., 2024)

2.3.2 Random Forest (RF)

RF is an ensemble learning method introduced by Breiman, designed for both classification and regression tasks (Kolli et al., 2023). It works by building many decision trees. Each tree is trained using a random part of the training data. For predictions, RF aggregates the outputs from all trees. In classification tasks, the model takes a majority vote from the

individual trees, while in regression tasks, it averages their predictions. At each decision point, each tree selects the greatest feature for splitting from a random subset of available features (Li, 2023). This random choice helps the forest find different patterns in the data and lowers the chance of overfitting. One unique feature of Random Forest is the use of out-of-bag (OOB) data. During training, some data points are left out of the training set for each tree. These OOB samples are later used to assess the model's accuracy by providing an internal performance evaluation without requiring a separate validation dataset. The algorithm has some good points. It can work with both categorical and numerical data, helps avoid overfitting, and runs well on large datasets. Additionally, because the trees can be trained in parallel, Random Forest is highly scalable and fast (Sudhakar et al., 2023).

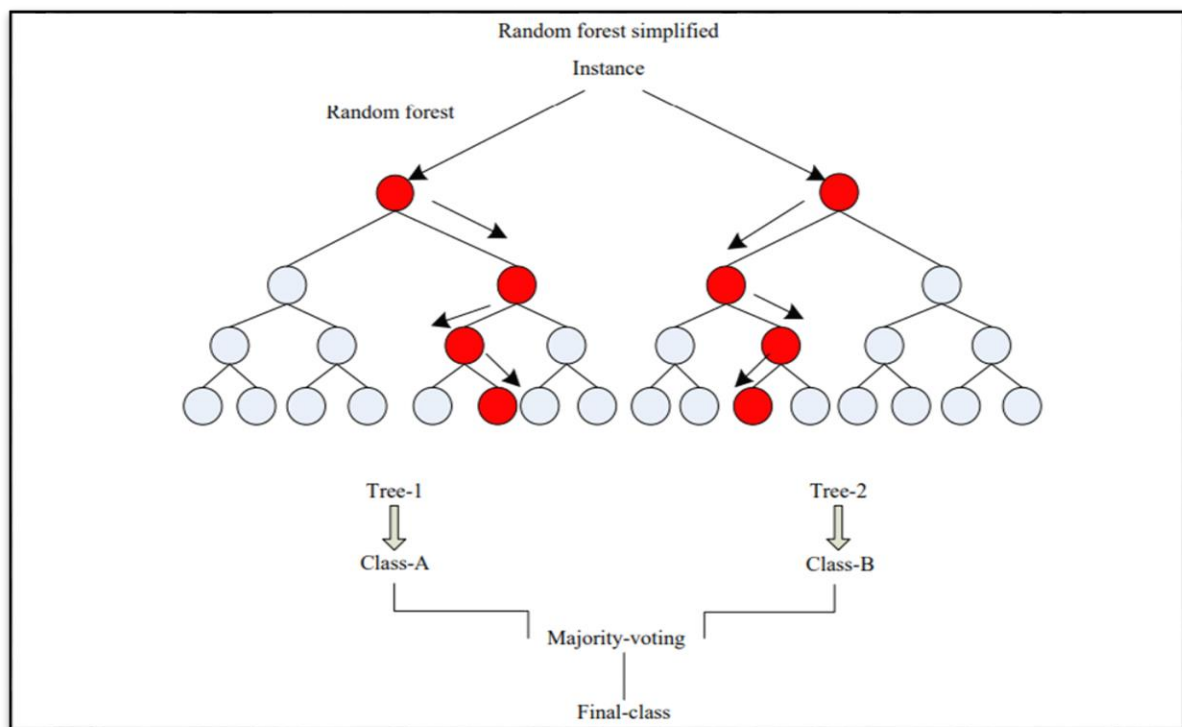


Figure 2.4: The Architecture of the Random Forest Model (Li, 2023)

2.3.3 Support Vector Machines (SVM)

SVM are supervised learning algorithms that are widely used for classification, regression, and outlier detection tasks (Kolli et al., 2023). The objectives of a SVM is to find the optimal hyperplane that best separates data into different classes (He, 2022). The

hyperplane works as a decision line. SVM tries to make the space, called the margin, as wide as it can. This margin is the distance between the line and the closest data points from each class. These closest points are called support vectors. They are important because they set the place and direction of the hyperplane to keep the classes as far apart as possible (Nimmada & Basha, 2024).

SVM are particularly effective when working with linearly separable data. However, their true strength lies in handling non-linear data by using kernel functions. Popular kernel functions include linear, polynomial, radial basis function (RBF), and sigmoid (Kolli et al., 2023).

SVM is built on statistical learning theory and focuses on minimizing classification errors by maximizing the margin between classes. This approach improves the model's ability to generalize to new, unseen data by making it suitable for a wide range of applications from simple binary classifications to multi-class classification problems (Koklu & Sulak, 2024)

SVM is best for performing high-dimensional data while remaining robust to outliers and noisy datasets. It supports both binary and multi-class classification tasks, consistently delivering reliable results. The RBF kernel, in particular, is highly effective in capturing non-linear relationships, making SVM a dependable choice in scenarios involving uncertainty or complexity (Krishna & Hemamalini, 2024 & Hemamalini, 2024).

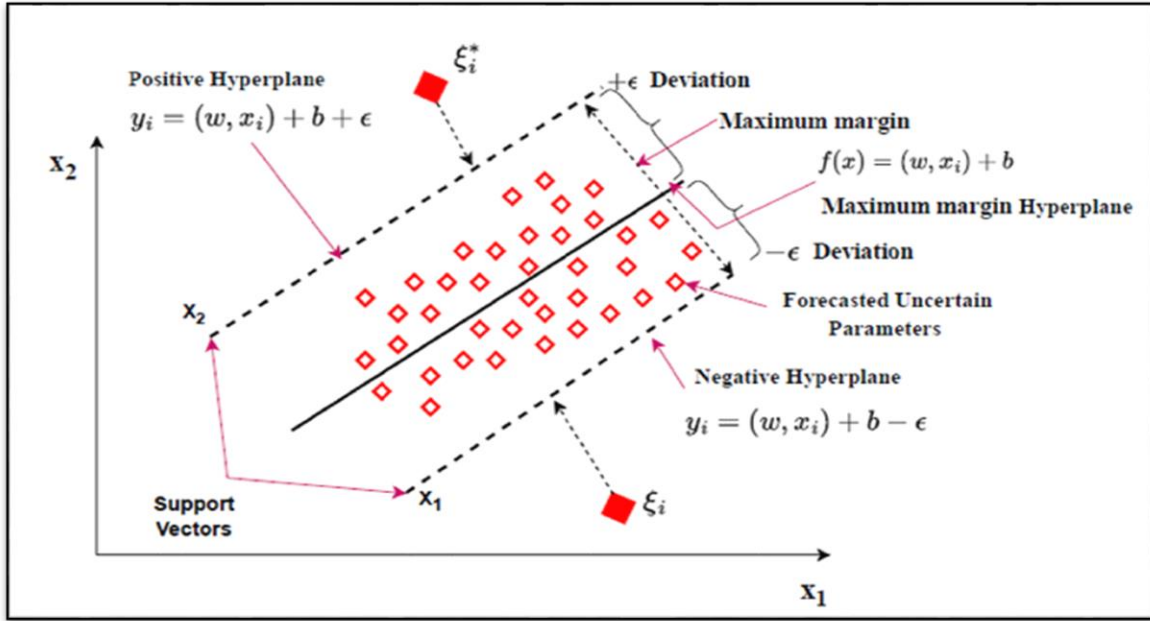


Figure 2.5: The Architecture of the Support Vector Machines Model (F. Wang et al., 2024)

2.3.4 XGBoost

Extreme Gradient Boosting, known as XGBoost, is a strong machine learning method. People often use it to solve classification problems. It is an optimized version of gradient-boosted decision trees and was developed by Tianqi Chen to handle large datasets efficiently (Santisteban Quiroz, 2022). Many people like to use XGBoost because it is fast and works well. The algorithm uses gradient boosting, which puts together many simple models, usually decision trees, to build a better one. Each tree tries to fix the mistakes made by the one before it. This helps make the predictions more accurate over time. Unlike traditional gradient boosting, XGBoost incorporates second-order optimization, which enhances computational speed and performance. It also includes regularization techniques to prevent overfitting to ensure the model generalizes well to unseen data (Nobel et al., 2024). Additionally, XGBoost can automatically handle missing values by simplifying data preparation compared to other ML algorithms. This makes it highly user-friendly and efficient (Yu et al., 2024). The algorithm gives some benefit over conventional methods such as BP neural networks, artificial neural networks (ANN), and k-nearest neighbors (KNN). These features make XGBoost a preferred choice for tackling both small-scale and large-scale machine learning tasks (Yu et al., 2024).

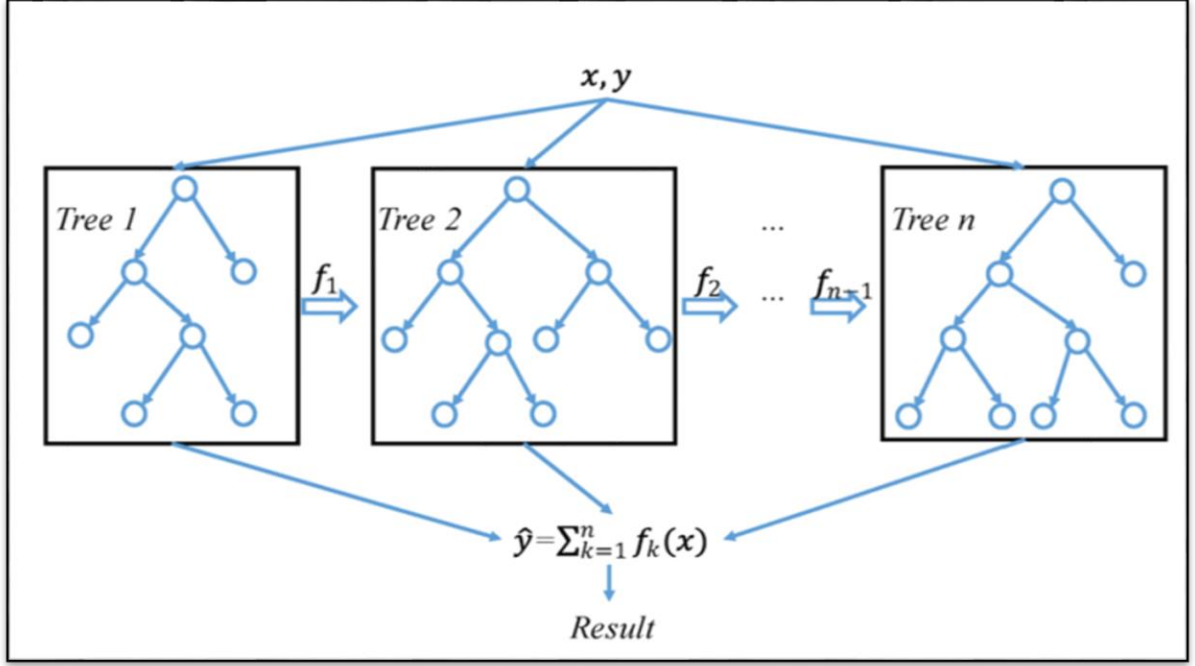


Figure 2.6: The Architecture of the XGBoost Model (Y. Wang et al., 2019)

2.3.5 Logistic Regression (LR)

Logistic Regression (LR) is a statistical method used for binary classification. It helps predict the chance of something happening. LR is different from linear regression, which gives continuous values. Instead, LR gives a probability between 0 and 1 by using the sigmoid function. This makes it good for problems with two possible outcomes (Kumar et al., 2024).

In logistic regression, the probability of an outcome is calculated using the sigmoid function, defined as:

$$z = \frac{1}{1 + e^{-(b_0 + b_1 x)}}$$

where Z is the predicted probability, and b_0 and b_1 are model parameters (Kuriakose et al., 2022). This sigmoid function transforms predicted values into probabilities which are then used to accept or reject variables based on their relevance (Manvitha & Rekha, 2023).

It is computationally efficient and works well with linearly separable data to allow for easy interpretation of results. This is particularly useful in understanding how different

predictors influence the outcome (Bhuvana et al., 2023). Furthermore, logistic regression can handle additional features without requiring significant changes to the model.

However, logistic regression does face challenges, particularly with high-dimensional data. In such cases, issues like overfitting and parameter sparsity can arise. To mitigate these problems, techniques like L1 (lasso) and L2 (ridge) regularization are often employed. These methods reduce the impact of less important variables, improve model accuracy, and help with feature selection, making the model more robust and reliable (Fang & Huy, 2023).

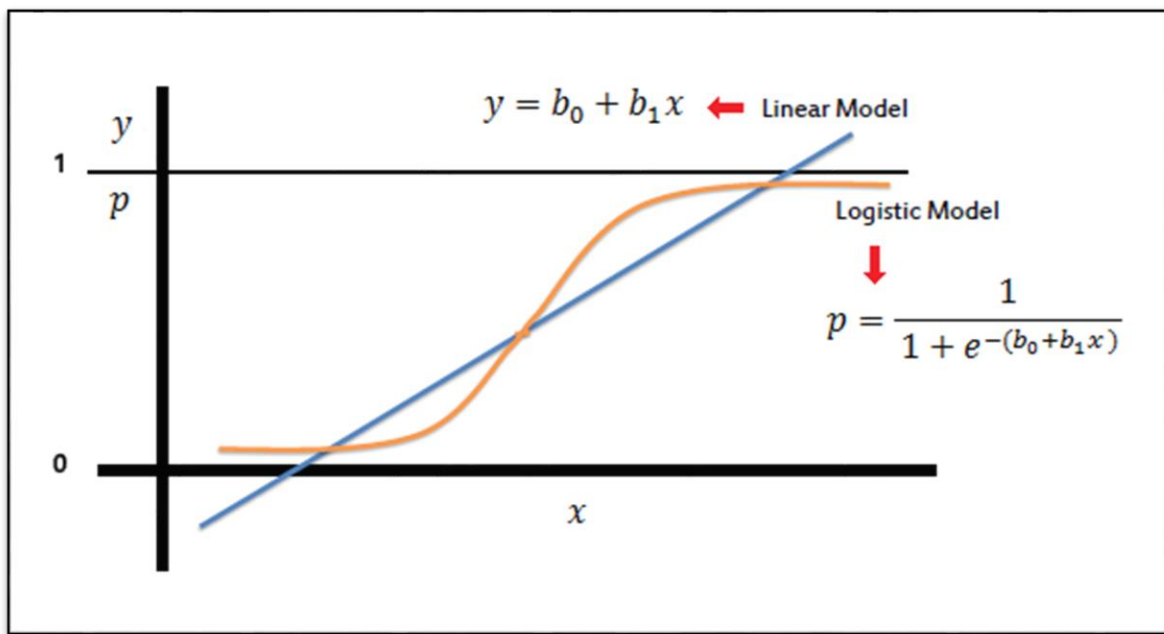


Figure 2.7: The Architecture of the Logistic Regression Graph

2.4 Related Work

Numerous studies have focused on predicting obesity risks using a various of ML models. This section reviews recent advancements in obesity prediction by focusing on the datasets, methodologies, and algorithms employed. It also examines the evaluation metrics used and summarizes the key findings from the research. ML techniques have been applied to improve prediction accuracy and identify key risk factors.

Ferdowsy et al. (2021) proposed a system to predict obesity risks in the Bangladeshi population using machine learning techniques. The authors collected data from surveys of

1,100 individuals which focus on 28 obesity-related factors. The data was prepared by cleaning it, filling in missing values, changing categories into numbers, removing noise using box plots, and normalizing the values. The authors implemented nine supervised machine learning methods, including k-Nearest Neighbors (k-NN), Logistic Regression, Support Vector Machines (SVM), Naïve Bayes, Classification and Regression Trees (CART), Random Forest, Multi-Layer Perceptron (MLP), AdaBoost, and Gradient Boosting Machines (GBM). Each algorithm was evaluated using metrics such as accuracy, precision, recall, F1-score, sensitivity, and specificity. Logistic Regression achieved the highest performance with an accuracy of 97.09%, along with a precision and F1-score of 97%. On the other hand, GBM showed the lowest performance, with an accuracy of 64.08% and a F1-score of 57%. In the end, the study found that Logistic Regression worked best for predicting obesity risk in this case.

Pinto et al. (2021) built a model to predict how likely obese patients are to lose weight. They used data from the patients' diet and exercise habits. This data was gathered using the Samsung Galaxy Active 2 smartwatch, which tracked heart rate, sleep patterns, exercise duration, and dietary habits over four months. For data preprocessing, the authors addressed missing values, organized the data chronologically, and removed irrelevant attributes to focus on essential features. RapidMiner was used to simplify the data preparation process. The study used two ML algorithms which are k-nearest neighbor (KNN) and decision tree, with specific parameter settings. For KNN, the optimal performance was achieved with a value of k equal to 9 for diet data and k equal to 1 for exercise data. The decision tree analysis was performed using J48 in WEKA. The decision tree model achieved an accuracy of 71.54% for diet data and 63.63% for exercise data. In comparison, KNN achieved an accuracy of 65.85% for diet data and 71.59% for exercise data. Based on these results, the study concluded that the decision tree was more effective for predicting diet-related data, while KNN performed better for analysing exercise-related data.

K. N. Devi et al. (2022) focuses on developing a predictive model to identify individuals at risk of obesity. The research was carried out by using two datasets which are one from the UCI Machine Learning Repository and another dataset collected from college students in Tamil Nadu, India. Data preprocessing involved normalizing all feature values by scaling them to a range of 0 to 1. The study utilized LR, DT, RF, SVM, Gradient Boosting, Ada Boost, and Voting Classifier. Hyperparameter tuning was performed using Grid Search and Randomized Search for LR and SVM. Results showed that the Voting Classifier demonstrated the highest performance with 99.48% accuracy on Dataset 1 and 96.92% on Dataset 2. After tuning, Logistic Regression performed exceptionally on Dataset 1 (99.68% accuracy) while the DT performed best on Dataset 2 with 97.28% accuracy. The study concluded that DT, LR were the most effective models after tuning for predicting obesity risk.

Mora-Brito & De Jesús Gil Herrera (2023) conducted research to enhance the early prediction of chronic conditions associated with obesity. The research was carried out by using datasets from the 2018 and 2021 National Health and Nutrition Surveys provided by the Mexican Ministry of Health. These datasets contained relevant physiological and behavioral variables, with 43,070 records in 2018 and 13,402 in 2021. For data preprocessing, the author cleaned records with missing values and identified variables that were present in both datasets. Categorical data was encoded, and the datasets were split into training (70%) and testing (30%) subsets. Four machine learning models were implemented which are LR, DT, RF, and Naive Bayes. Each model was optimized with specific parameters, such as adjusting the depth of the DT and the number of trees in the RF. The models were evaluated using metrics like accuracy, precision, recall, specificity, and the area under the curve (AUC). Results showed that the Random Forest model achieved the highest accuracy (80.43% in 2018 and 84.23% in 2021) and balanced performance across metrics, with AUC values of ~ 0.81 , although sensitivity remained low. Naive Bayes demonstrated inconsistent results, with accuracy ranging from

75.98% (2018) to 64.7% (2021). Based on these results, the study concluded that RF demonstrated the most balanced performance across metrics.

Fernandes et al. (2023) conducted a study to identify the most effective machine learning algorithm for predicting obesity in adults. This study analyzed data collected from Colombia, Peru, and Mexico. The dataset included 2111 records and 17 attributes, focusing on individuals' physical characteristics and eating habits. During data preprocessing, the variables were converted into categorical values, and missing data was addressed. The authors used the WEKA software to evaluate the models and applied a 10-fold cross-validation method. The study tested three ML algorithms such as Naive Bayes, RF, and OneR. These models were evaluated based on accuracy, mean absolute error, area under the ROC curve (AUC), precision, and recall. The Random Forest algorithm showed the best performance which achieved an accuracy of 95.64%, a mean absolute error of 0.1148, an AUC value of 0.998, a precision score of 0.959, and a recall score of 0.956. In comparison, OneR algorithm had slightly lower performance, with an accuracy of 67.01%, a mean absolute error of 0.0941, an AUC of 0.935, a precision of 0.669, and a recall of 0.671. Based on these results, the study concluded that RF is the most effective and reliable algorithm for predicting obesity in adults.

Dwivedi et al., (2024) aimed to detect obesity levels using machine learning models. Data preprocessing included feature selection, correlation analysis, and the removal of irrelevant features to enhance model performance. The study used Support Vector Machine (SVM), Random Forest, Logistic Regression, Decision Tree, Naive Bayes, Gradient Boosting, AdaBoost, and CatBoost. Model tuning involved selecting an appropriate kernel for SVM and adjusting estimators for ensemble methods. Performance metrics such as accuracy, precision, recall, F1-score, and AUC-ROC curves were used for evaluation and comparison. Among the models, SVM demonstrated the highest accuracy of 96.97%, followed closely by AdaBoost

with 96.59%. Naive Bayes, however, showed limitations due to its significantly lower accuracy. The findings emphasized SVM as the most effective model for obesity prediction.

Lee & Chun (2024) aimed to understand the factors contributing to overweight and obesity among Korean adolescents through machine learning. They used a dataset from the Korean Youth Risk Behavior Web-based Survey (KYRBS) which included records from 43,268 adolescents. Data preprocessing involved filling in missing values, feature selection, and splitting data into training and testing subsets, with an 80%-20% split, evaluated using 5-fold cross-validation for robust model assessment. The machine learning algorithms applied included LR, Ridge, LASSO, ElasticNet, DT, Bagging, RF, AdaBoost, and XGBoost, with parameters optimized for each model. The model's performance was evaluated using accuracy, recall, precision, F1 score, and AUC score. As a result, XGBoost algorithm achieved the highest accuracy (0.8403) and AUC score (0.8982), outperforming other models on metrics such as recall (0.6351), precision (0.6497), F1 score (0.6423), and overall prediction accuracy.

Koklu & Sulak (2024) conducted a study to develop machine learning models for classifying obesity status. The research focused on predicting obesity based on social and physical factors using data collected from 1,610 individuals in Turkey through an online survey. The dataset included 14 attributes, such as age, physical activity, dietary habits, and family history of obesity. For data preprocessing, the authors addressed missing values, normalized the data, and corrected any inconsistencies. They used 10-fold cross-validation to validate their models. The study applied four machine learning algorithms which are Artificial Neural Network (ANN), Support Vector Machine (SVM), K Nearest Neighbors (KNN), and Random Forest (RF). Each model was optimized for performance. ANN was structured with a multilayer network, SVM used a radial basis function kernel, KNN was configured with a fixed number of neighbors, and RF utilized multiple decision trees for classification. These models were evaluated based on accuracy, precision, recall, and F1 scores. The results revealed that

RF achieved the highest accuracy at 87.82%, outperforming KNN (80.62%), ANN (74.96%), and SVM (74.03%). Based on these findings, the study concluded that Random Forest was the most effective model for predicting obesity in this dataset.

In summary, the studies reviewed emphasize the potential of machine learning for predicting obesity by analyzing various important features. Each study was summarized by providing a brief description, dataset, the machine learning algorithms employed, the evaluation metrics used, and the results, as outlined in Table 2.1.

2.5 Comparison of Related Work

Table 2.1: Summary of Related Works

Title (Author, Year)	Brief Description	Data Source	Machine Learning Algorithms	Evaluation Metrics	Best Results
A Machine Learning Approach for Obesity Risk Prediction (Ferdowsy et al., 2021)	- Developed a ML system to predict obesity risks in Bangladesh. - Logistic Regression performed best accuracy.	Survey data from 1,100 individuals with 28 obesity-related factors.	- k-NN - Logistic Regression - SVM - CART - Naïve Bayes - Random Forest - MLP - AdaBoost - GBM	- Accuracy - Precision - Recall - F1-Score - Sensitivity - Specificity	Logistic Regression (97.09%)
Diet & Exercise Classification using Machine Learning to	Predicts weight loss likelihood using diet and exercise data.	Data was collected over four months from the Samsung	- k-NN - Decision Tree	- Accuracy - Recall - Precision	Decision Tree (71.54%)

Predict Obese Patient's Weight Loss (Pinto et al., 2021)	• Decision Tree performed best accuracy on diet data. • k-NN performed best accuracy for exercise data.	Galaxy Active 2 smartwatch.		• F1 Score • AUC	• k-NN (71.59%)
Machine Learning Based Adult Obesity Prediction (Devi et al., 2022)	• Developed a predictive model to identify individuals at risk of obesity. • Random Forest achieved the highest accuracy on Dataset 1. • Voting Classifier performed best on Dataset 2. • Logistic Regression reached highest accuracy after tuning on Dataset 1.	Dataset 1: UCI Machine Learning Repository (2711 records, 17 attributes) Dataset 2: College data from Tamil Nadu, India (553 records, 16 attributes)	• Logistic Regression • Decision Tree • Random Forest • SVM • Gradient Boosting • Ada Boost • Voting Classifier	• Accuracy • Precision • Recall • F1 Score • AUC Score	• Random Forest (99.36%) • Voting Classifier (96.92%) • Logistic Regression (99.68%)

Associating Obesity to Chronic Conditions through Machine Learning Techniques: A Mexican Case (Mora-Brito & De Jesús Gil Herrera, 2023)	- Enhanced early prediction of obesity-associated conditions. - Random Forest showed the highest accuracy in both dataset 2018 and 2021. - While accuracy and specificity were high, sensitivity was notably low, especially in 2021.	Dataset from Mexican public health survey data from 2018 (43,069 records, 32 variables) and 2021 (13,400 records, 32 variables).	- Logistic Regression - Decision Tree - Random Forest - Naïve Bayes	- Accuracy - Precision (PPV) - Recall (Sensitivity) - Specificity - AUC	- Random Forest (80.43% accuracy, 67.09% PPV, 34.6% sensitivity, 94.71% specificity, and an AUC of 0.817) in 2018. - For 2021, Random Forest (84.23%)
--	---	---	--	---	--

Comparison of Machine Learning Algorithms for Obesity Prediction (Fernandes et al., 2023)	- Identifies best algorithms for predicting obesity in adults. - Random Forest was the most effective model with the highest accuracy.	Dataset from Colombia, Peru, and Mexico (2111 records, 17 attributes on physical conditions and eating habits).	- Naïve Bayes - Random Forest - OneR	- Accuracy - Mean Absolute Error - Receiver Operating Characteristic Curve (ROC) - Precision - Recall	Random Forest (95.64%)
Obesity Risk Detection Using Machine Learning Techniques (Dwivedi et al., 2024)	- Detect obesity levels using machine learning models. - SVM achieved the highest accuracy and metrics performance.	Online survey data collected from participants in Mexico, Peru, and Colombia. 2111	- SVM - Random Forest - Logistic Regression - Decision Tree - Naïve Bayes	- Accuracy - Precision - Recall - F1 Score - AUC-ROC	SVM (96.97%)

		records with 17 attributes.	• Gradient Boosting • AdaBoost • CatBoost		
Identification of important features in overweight and obesity among Korean adolescents using machine learning (Lee & Chun, 2024)	• Explores obesity factors among Korean adolescents. • XGBoost achieved the highest accuracy and metrics performance.	Dataset from 16th Korean Youth Risk Behavior Web-based Survey (KYRBS), 43,268 records with 71 variables covering health behaviours and sociodemographic factors.	• Logistic Regression • Ridge • LASSO • ElasticNet • Decision Tree • Bagging • Random Forest • AdaBoost • XGBoost	• Accuracy • Precision • Recall • F1 Score • AUC	XGBoost (84.03%)
Using Artificial Intelligence	• Predicts obesity based on social and physical factors.	Online survey data from 1,610	• Artificial Neural Network (ANN)	• Accuracy • Precision	Random Forest (87.82%)

Techniques for the Analysis of Obesity Status According to the Individuals' Social and Physical Activities (Koklu & Sulak, 2024)	• Random Forest provide robust prefictions.	individuals in Turkey with 14 features.	• Support Vector Machine (SVM) • K Nearest Neighbors (KNN) • Random Forest (RF)	• Recall • F1 Score	
--	---	---	---	--	--

The reviewed works used various datasets, machine learning algorithms, and evaluation metrics to predict obesity. Datasets included public health surveys (e.g., Mexican and Korean) and smaller surveys, with sizes ranging from hundreds to tens of thousands of records. Popular machine learning algorithms included Decision Tree, Logistic Regression, Random Forest, SVM, and ensemble methods like AdaBoost and XGBoost. Among these, Random Forest was the most frequently used and showed strong performance, achieving accuracy between 80% and 99%. Logistic Regression performed well with high accuracy and AUC scores. Decision Tree also performed well on smaller datasets and provided interpretable models. SVM also achieved high accuracy in several cases, with one study reporting 96.97%. Not only that, ensemble methods like XGBoost also excelled in specific contexts, achieving high AUC scores, 0.8982 and balanced metric performances. Evaluation metrics like Accuracy, Precision, Recall, F1-Score, and AUC-ROC were widely used, with accuracy being the primary focus in most studies. Overall, the studies highlighted the importance of preprocessing steps, feature selection, and hyperparameter tuning played a key role in improving the results of all algorithms.

2.6 Evaluation Metrics

ML models are evaluated using various metrics to assess their performance. Commonly used metrics include Accuracy (AC), Precision (P), Recall (R), and F1-Score (F) which provide insights into classification effectiveness (Koklu & Sulak, 2024). Additionally, ROC-AUC analysis is employed to compare results from different methods using parameters such as True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) (Y. Wang et al., 2019). These parameters help evaluate classification accuracy and discrimination ability. Below, each metric is described in detail with equations provided for clarity.

2.6.1 Accuracy

Accuracy is a metric that measures the proportion of correct predictions (True Positives and True Negatives) out of the total number of predictions (Mora-Brito & De Jesús Gil Herrera, 2023). It is calculated as:

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} , TP \& TN \uparrow, Accuracy \uparrow$$

Accuracy is intuitive and widely used, especially for datasets where all classes are equally represented balanced datasets. A high accuracy indicates that the model correctly predicts most of the outcomes. However, in imbalanced datasets, where one class significantly outweighs the others, accuracy can be misleading. For instance, a model that predicts only the majority class might achieve high accuracy while failing to identify the minority class altogether. In such scenarios, metrics like Precision, Recall, and F1-Score are preferred to be more effectively handle the complexity of performance.

2.6.2 Precision

Precision measures how many of the positive predictions made by the model are correct. It is calculated as:

$$Precision = \frac{TP}{TP + FP} , ideally = 1 \text{ if balanced dataset}$$

Precision is crucial when it is important to minimize False Positives. A high precision means that when the model predicts a positive case, it is likely to be correct. In obesity prediction, precision evaluates how accurately the model identifies cases of a specific obesity level. For instance, if the model predicts that an individual belongs to a certain obesity category, high precision ensures that this prediction is reliable and not a false alarm (Shakti et al., 2024). This metric is particularly useful in ensuring that positive predictions are trustworthy and helps to focus on correctly identified cases. In balanced datasets, precision ideally approaches 1 to reflect highly accurate positive predictions.

2.6.3 Recall

Recall as Sensitivity or True Positive Rate (TPR) is a measurement that measures the proportion of actual positive instances correctly identified by the model. It is calculated as:

$$Recall = \frac{TP}{TP + FN} , ideally = 1 \text{ if balanced dataset}$$

Recall is critical in scenarios where minimizing false negatives is essential, as it ensures the model identifies most true positive cases without overlooking actual positive instances. In the context of obesity prediction, recall reflects the model's ability to detect all cases of a specific obesity level. However, a high recall may lead to an increase in false positives, potentially lowering precision (Shakti et al., 2024). Therefore, recall is often assessed in conjunction with other metrics, such as precision or the F1-Score to achieve a balanced evaluation of model performance.

2.6.4 F1-Score

The F1-Score is the harmonic mean of Precision and Recall into a single metric that balances the two. It is calculated as:

$$F1 \text{ Score} = 2 * \frac{Precision * Recall}{Precision + Recall} , \text{Harmonic Mean of Precision and Recall}$$

The F1-Score is particularly useful in situations where there is an inverse relationship between Precision and Recall, such as with imbalanced datasets. It provides a way to evaluate models by considering both the ability to correctly identify positive cases (Recall) and the accuracy of those predictions (Precision). In obesity prediction, a high F1-Score ensures the model not only makes accurate predictions for specific obesity levels but also captures a substantial proportion of all actual cases (Shakti et al., 2024).

2.6.5 Area Under ROC Curve (ROC-AUC)

The Receiver Operating Characteristic (ROC) curve is a tool used to evaluate how well a model can separate two classes, like positive and negative cases. It compares the True Positive Rate (TPR), or sensitivity, with the False Positive Rate (FPR), or the rate at which negatives are incorrectly classified as positives at different threshold values. Otherwise, the Area Under the Curve (AUC) is a number that summarizes the ROC curve. It shows how good the model is at distinguishing between the two classes. A higher AUC means the model performs better, with a value between 0.5 (random guessing) and 1 (perfect classification) (Mora-Brito & De Jesús Gil Herrera, 2023). Therefore, in obesity prediction, a higher AUC means the model does a good job separating different levels of obesity from non-obesity cases.

$$\text{True Positive Rate}(TPR) = \frac{TP}{TP + FN}$$

$$\text{False Positive Rate}(FPR) = \frac{FP}{FP + TN}$$

$$AUC = \frac{\text{True Positive Rate}(TPR)}{\text{False Positive Rate}(FPR)}$$

AUC is calculated based on the TPR and FPR values across thresholds to provide a comprehensive evaluation of model performance.

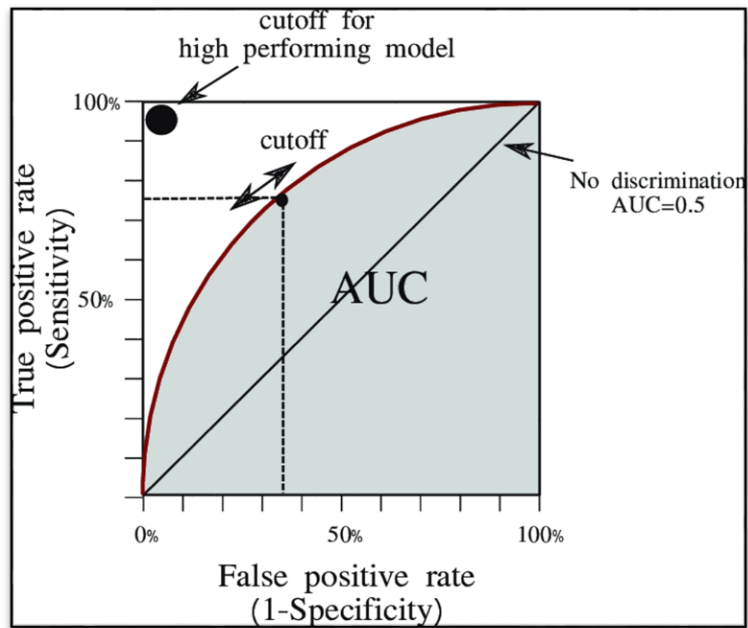


Figure 2.8: AUC Graph

2.7 Summary

As a summary, this chapter provides a comprehensive overview of the literature on obesity prediction which aims on the role of ML techniques in addressing this global health crisis. It begins by discussing the sharp increase in obesity rates worldwide which caused by factors such as genetic predisposition, unhealthy lifestyle choices, and changes in the environment. The chapter explains how obesity significantly impacts physical health by contributing to chronic illnesses. It also highlights the economic strain it places on individuals and healthcare systems due to rising medical costs and lost productivity. Furthermore, the discussion introduces ML models like DT, RF, SVM, XGBoost, and LR. These models are highly effective at analyzing complex data to identify patterns and assess obesity risks. The chapter also reviews previous studies that demonstrate how these ML models are used to predict obesity in different groups, such as children and adults. These studies showcase the potential of ML to analyze critical factors like diet, exercise, and genetics to develop customized prevention strategies. Comparative analyses of recent studies illustrate the strengths of these models in different contexts, with RF frequently achieving high accuracy and robust performance. The study used accuracy, precision, recall, F1-Score, and AUC to check how well the models worked. In conclusion, the chapter emphasizes the importance of combining ML predictions with practical public health actions.

Chapter 3 : Methodology

3.1 Introduction

This chapter discussed the methodology which include the tasks and dataset that are needed for implementation. As a short flashback, DT, RF, SVM, XGBoost, and LR are the machine learning approaches that will be compared to get the best results and to build our obesity prediction model. Thus, our workflow uses the CRISP-DM methodology. It is a simple and well-known framework used in data mining and machine learning projects (Solano et al., 2022). In this chapter, the methods used to implement this project will be discussed in detail. The programming language of Python will be used throughout the project for building the prediction model and Jupyter Notebook will be the platform for coding and execution.

3.2 Cross Industry Standard Process for Data Mining (CRISP-DM)

CRISP-DM (Cross-Industry Standard Process for Data Mining) is a standard used in many industries. It helps solve business problems with data mining. It is flexible and works with different tools and applications. This makes it useful for both business and research (Lathifah et al., 2023). CRISP-DM gives a life cycle approach for AI and machine learning projects. This methodology breaks down data mining into six clear phases which are Business Understanding, Data Understanding, Data Preparation, Modelling, Evaluation, and Deployment. It is a good method for finding knowledge in databases (KDD). This structured yet flexible nature of CRISP-DM makes it ideal for projects like obesity prediction. By following each phase step-by-step, the framework helps us understand the problem. It also helps us prepare data. Then, it helps us build and refine machine learning models and finally interpreting the results in a practical and insightful way. Figure 3.1 below displays the proposed methodology's flowchart.

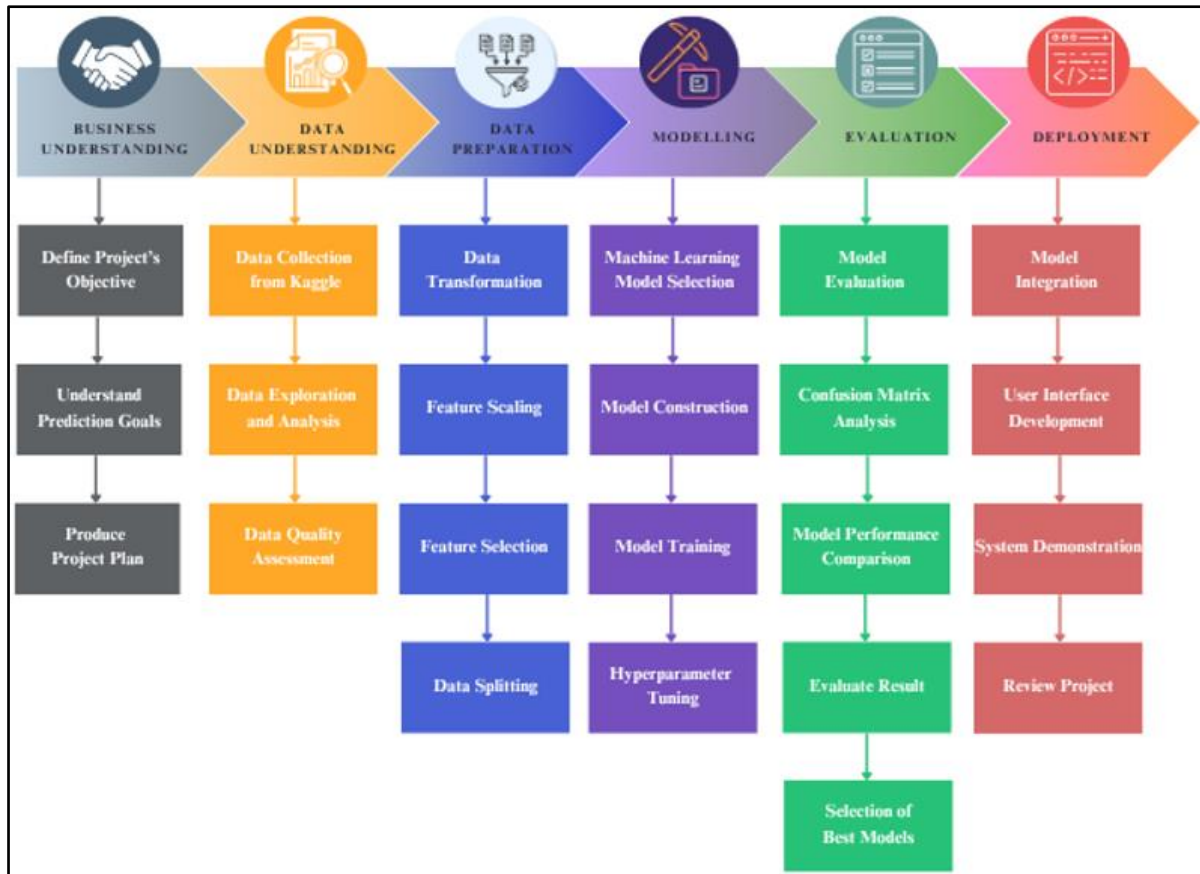


Figure 3.1: CRISP-DM Methodology Flowchart

3.2.1 Business Understanding

The first phase focuses on understanding the project's goals and needs from a business perspective. This information is changed into a clear data mining problem and a basic plan to meet the project's goals (Maataoui et al., 2023). Therefore, there are a few major tasks to be focused on this phase. First, a planning is decided, and project timeline were proposed and determined throughout 2 semesters. To be precise, pre-development phases are all planned and executed in this semester (October – February) and the development would be on latter semester. Tools and techniques, including Python programming with libraries like pandas, scikit-learn, and NumPy, will be used for implementation, with coding and testing conducted on Jupyter Notebook. Then, prediction goals are set to predict obesity by analysing individual inputs across various features associated with obesity. We have determined that the project will

focus on a multiclass outcome representing different levels of obesity. Furthermore, we will carefully consider dataset selection later in the process to ensure we achieve the desired results.

3.2.2 Data Understanding

This phase collecting the dataset, exploring its structure, and preparing it for further analysis. The dataset for this project is from Kaggle in .csv format. It has 17 attributes and 2111 records. The data includes binary, numeric, and categorical types. The target variable, NObesity (Obesity Level), divides individuals into seven obesity levels based on BMI. These levels are Insufficient Weight (BMI < 18.5), Normal Weight (BMI 18.5–24.9), Overweight Level I (BMI 25–29.9), Overweight Level II (BMI 30–34.9), Obesity Type I (BMI 35–39.9), Obesity Type II (BMI 40–49.9), and Obesity Type III (BMI > 50).

Initial exploration is done to understand the structure and distribution of the data. The attributes are put into four groups. The first category is basic information, like Gender, Age, Height, Weight, Family history of overweight, and Smoking. The second category is eating habits, like Frequent consumption of high-calorie food, Frequency of consumption of vegetables, Number of main meals, Consumption of food between meals, Consumption of water daily, and Consumption of alcohol. The third category is physical activity, like Calorie consumption monitoring, Physical activity frequency, Time using technological devices, and Type of transportation used. Lastly, the dependent variable is NObeyesdad.

Additionally, Exploratory Data Analysis (EDA) helps to understand the dataset, find trends, and see how the data is distributed. This is done using tools like Jupyter Notebook for preparing and visualizing data or Python libraries like Matplotlib and Seaborn for creating graphs. EDA uses techniques such as box plots, bar charts, pie charts and other visualizations. These help find patterns and relationships between features and the target variable (Shakti et al., 2024). For example, demographic features like “Age,” “Height,” and “Weight” are studied

using box plots and joint plots. Pie charts are used to show the different levels of obesity and their prevalence.

The dataset is assessed for quality by addressing issues like missing values, noisy data, duplicate entries, and inconsistencies. Noisy data, like outliers and errors, are identified and fixed. Outliers are checked to see if they are valid or mistakes. Duplicate entries are removed. Inconsistencies, like mismatched categories, are corrected. These steps make the dataset ready for data preprocessing.

Table 3.1: Dataset Attributes and their Data Type and Descriptions

Category	Feature	Data Type	Description
Basic Information	Gender	Binary	Gender (Female, Male)
	Age	Numeric	Age (Numeric value in the range from 14 to 61)
	Height	Numeric	Height (Numeric value in meters)
	Weight	Numeric	Weight (Numeric value in kilograms)
	family_history_with_overweight	Binary	Family history with overweight (yes, no)
	SMOKE	Binary	Smoking (yes, no)
	FAVC	Binary	Frequent consumption of high caloric food (yes, no)
	FCVC	Numeric	Frequency of consumption of vegetables (Numeric value in the range 1 to 3)

Related to eating habits	NCP	Numeric	Number of main meals (Numeric value in the range 1 to 4)
	CAEC	Categorical	Consumption of food between meals (Divided into 4 classes: no, Sometimes, Frequently, Always)
	CH2O	Numeric	Consumption of water daily (Numeric value in the range 1 to 3)
	CALC	Categorical	Consumption of alcohol (Divided into 4 classes: no, Sometimes, Frequently, Always)
Related with the physical condition	SCC	Binary	Calorie consumption monitoring (yes, no)
	FAF	Numeric	Physical activity frequency (Numeric value in the range 0 to 3)
	TUE	Numeric	Time using technological devices (Numeric value in the range 0 to 2)
	MTRANS	Categorical	Type of transportation used (Divided into 5 classes:

			Automobile, Motorbike, Bike, Public transportation, Walking)
Dependent variable	NObeyesdad (Target)	Categorical	Obesity level (Insufficient Weight, Normal Weight, Overweight Level I, Overweight Level II, Obesity Type I, II, and III)

3.2.3 Data Preparation

The data preparation or data preprocessing phase ensures the dataset is clean, properly formatted, and optimized for machine learning models. The raw data was processed to make it suitable for training. First, encoding categorical variables ensures compatibility with machine learning. For binary features like “Gender,” “family_history_with_overweight,” “SMOKE,” and “FAVC,” label encoding is used. “Gender” is converted to 0 for Female and 1 for Male. “yes” and “no” are encoded as 1 and 0. For the nominal feature “MTRANS”, which contains multiple categories without a natural order, LabelEncoder was used to assign unique integer labels. On the other hand, ordinal variables such as “CAEC” and “CALC” are also encoded using an LabelEncoder, which preserves their natural order such as no = 0, Sometimes = 1, Frequently = 2, and Always = 3.

During the transformation phase, the target variable, “NObeyesdad,” which is of categorical type, should be transformed into integer labels for compatibility with machine learning algorithms. For example, 0 = Insufficient Weight, 1 = Normal Weight, 2 = Overweight Level I, 3 = Overweight Level II, 4 = Obesity Type I, 5 = Obesity Type II, 6 = Obesity Type III. This process is performed using a LabelEncoder, which assigns a unique integer to each

category. This transformation ensures that the target variable is represented in a format suitable for machine learning models.

For numerical features, two scaling techniques were applied to normalize their ranges. The “Age”, “Height”, and “Weight” columns were standardized using StandardScaler, which centers the data to have a mean of 0 and standard deviation of 1. This standardization prevents features with larger scales such as height in meters and weight in kilograms from dominating the learning process.

Meanwhile, features like “CH2O”, “FCVC”, “NCP”, “FAF”, and “TUE” were scaled using MinMaxScaler to rescale their values between 0 and 1, which can be beneficial for algorithms sensitive to feature ranges.

Feature selection is used to find the most important features for the predictive model. This step removes irrelevant or redundant features, helping to improve the model's accuracy, reduce overfitting, and make it easier to understand. The process begins with a correlation heatmap, which visually assesses the relationships between features. The heatmap reveals strong correlations between pairs of features, enabling the removal of highly correlated or redundant features to prevent multicollinearity. Weakly correlated features that do not provide significant information about the target variable can also be discarded.

Once the relationships between features are understood, weighting methods such as Weight by Relief, Weight by Correlation, Weight by Information Gain, and Weight by Gain Ratio are applied to further refine feature selection. These methods assign scores to each feature. Higher scores show that a feature is more important for predicting the target variable.

Lastly, the dataset is then split into subgroups for training and testing. 80% of the data is used for training the model, while the remaining 20% is used for testing. The model is trained, and its parameters are adjusted using the training set. The model's performance during training

is evaluated using the testing set, which also aids in parameter adjustment. This division enhances the model's performance and generalisability.

3.2.4 Modeling

The modeling phase involved selecting and applying a range of machine learning algorithms to the prepared dataset to predict the obesity level based on the selected features. The following classification algorithms were chosen for training which are Logistic Regression, Random Forest, Support Vector Machine (SVM), Decision Tree, and XGBoost. Each model was selected for its unique strengths. For example, Decision Tree was included because to its ease of use and interpretability while Random Forest was chosen because of its capacity to manage intricate interactions and prevent overfitting. SVM was considered for its effectiveness in high-dimensional spaces. Logistic Regression provided a baseline for linear relationships, and XGBoost was implemented for its efficiency in handling large datasets and strong predictive performance. Each model was trained using the final selected features in the data preparation phase, with obesity level as the target variable.

During the training phase, default hyperparameters settings were applied for each model to establish baseline performance. Hyperparameter tuning followed to optimize the models' performance. Techniques like GridSearchCV or RandomSearchCV are used to explore different combinations of hyperparameters. GridSearchCV tests all possible combinations of a set of hyperparameters, while RandomSearchCV selects a random sample of combinations, which is faster. For instance, the maximum depth, the number of samples necessary in a leaf, and the number of samples required to divide a node were all changed in the Decision Tree. Random Forest was improved by testing different numbers of trees, maximum depths, and splitting rules. SVM was optimized by changing its regularization settings and testing different kernel parameters. The learning rate, the number of boosting rounds, and the tree depth were all changed to modify XGBoost. Logistic Regression was improved by adjusting the

regularization settings and the type of penalty used. This systematic approach ensured that each algorithm was fine-tuned for its best predictive accuracy. Once the best parameters were identified, models were retrained and tested for accuracy and consistency.

3.2.5 Evaluation

This evaluation phase checks how well machine learning models predict obesity levels. It ensures the models meet project goals and technical needs. Different metrics evaluate the models to show their strengths and limitations.

Models are assessed for classification accuracy using a confusion matrix. In this study, the classification results are displayed using a multi-class confusion matrix. For every class, this 7x7 matrix displays the numbers of True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN). For example, TP represents correctly identified cases of a specific obesity level, while FP and FN show incorrect predictions. These misclassifications can have clinical effects, such as confusing Obesity Type III with Obesity Type II. An example of this N x N matrix is shown in Table 3.2 below, where N is the number of target classes. The confusion matrix helps to understand whether the model's predictions align with the actual values. Additionally, Table 3.3 below defines the shorthand terms for the combinations of predicted and actual values.

Table 3.2: Sample of Binary Class Confusion Matrix

		Predicted Class	
		Positive	Negative
Actual Class	Positive	TP	FP
	Negative	FN	TN

Table 3.3: Description of the Combinations of Predicted and Actual Values

Evaluation Criteria	Description
True Positive (TP)	Correctly predicted a specific obesity level (e.g., predicted Obesity Type II and actual was Obesity Type II).
False Positive (FP)	Predicted a higher obesity level than actual (e.g., predicted Obesity Type III but actual was Overweight Level II).
True Negative (TN)	Correctly predicted a non-matching obesity level (e.g., predicted Normal Weight and actual was not Obesity Type I).
False negative (FN)	Misclassified a specific obesity level as another (e.g., predicted Normal Weight but actual was Obesity Type I).

Meanwhile, the model's performance is evaluated using metrics such as accuracy, precision, recall, F1 score, and ROC-AUC. Accuracy tells how often the model gives the right prediction. Precision tells how many predictions of a specific obesity level were correct. Recall shows how many actual cases of an obesity level the model found. The F1 score is a mix of precision and recall, helping when data is not balanced. ROC-AUC shows how well the model separates different obesity levels. Therefore, metrics are calculated for all models to compare their performance. These evaluations help refine the models and select the best one for predicting obesity levels. The final model is prepared for implementation.

3.2.6 Deployment

The deployment phase involves making the trained machine learning model accessible for real-world use. First, the model is saved in a serialized format such as .pkl to allow for easy loading during deployment. A user-friendly interface is then developed using Gradio, which enables users to input data and receive predictions interactively. The complete application, including the model and interface, is deployed to Hugging Face Spaces. It is a platform that hosts machine learning web applications online. This setup allows end users to access the

model through a web browser without requiring complex local installations. Finally, the deployed system is demonstrated to validate its usability and functionality.

3.3 Summary

This chapter concludes with a summary of the methodology, which provides a foundation for the subsequent chapters on implementation, results, and analysis of the obesity prediction model. These phases include business understanding, data understanding, data preparation, modeling, evaluation, and deployment are needed to achieve the objectives and scope of the project, to build an obesity prediction model by comparing various machine learning models including Decision Trees, Random Forest, Support Vector Machine (SVM), XGBoost, and Logistic Regression and select the best performance model for developing our prototype of obesity prediction model.

Chapter 4 : Data Mining Implementation

4.1 Introduction

The table 4.1 below lists all the main tools, packages, and libraries used in this project. Each one had a specific role in helping with the data processing, visualization, machine learning model building, and result analysis. These tools were chosen because they are well-known, reliable, and work well together in the Python environment.

Table 4.1: A Summary of Software Tools, Libraries, and their Purposes

Tool / Library	Version	Purpose
Jupyter Notebook	7.3.3	Used for coding, visualizing data, running experiments, and writing notes
Python	3.13.2	Main programming language used throughout the project
Pip	23.2.1	Used to install and manage Python packages
NumPy	1.26.0	Helped with fast numerical operations and array handling
Pandas	2.2.3	Used for working with data, cleaning, and analyzing it in table format
Matplotlib	3.8.0	Created graphs and plots to show data patterns
Seaborn	0.13.2	Made more detailed and styled data visualizations

IPython	9.0.2	Improved the interactive coding experience in Jupyter
Scikit-learn	1.6.1	Main library for machine learning models and data processing tasks
Skrebate	0.62	Used ReliefF algorithm for choosing important features
XGBoost	3.0.0	Used to build strong gradient boosting models for better predictions
time (Python module)	—	Measured how long certain parts of the code took to run
Joblib	1.3.2	Saved and loaded trained models for future use

4.2 Cross Industry Standard Process for Data Mining (CRISP-DM)

In this chapter, processes of the methodology would be discussed. Besides, the implementation and training of the selected model will be discussed thoroughly with explanations. By sections, the processes involved the processes from the methodology, CRISP-DM. As such, it includes data understanding, data preparation, modeling, evaluation, and deployment.

4.2.1 Data Understanding

Initially, a dataset containing obesity-related data was downloaded from Kaggle. The dataset consists of 16 attributes, and the target variable in the dataset is obesity level. The

dataset downloaded from Kaggle consists of 2111 obesity-related data. Figure 4.1 shows the portion of data description of the dataset from Kaggle.

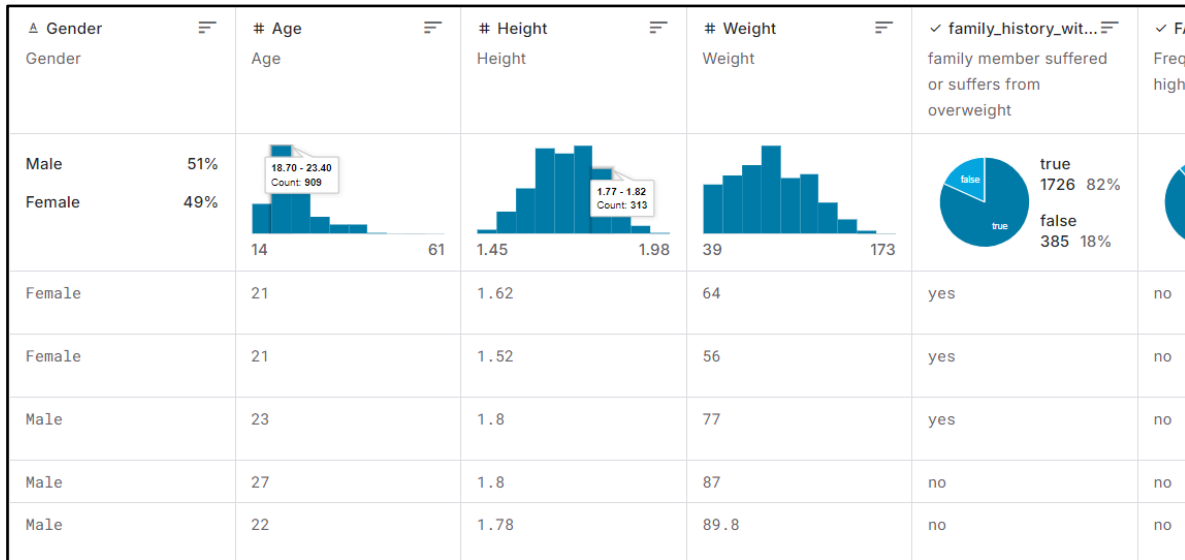


Figure 4.1: Portion of Data Description of the Dataset from Kaggle

As shown in Figure 4.2, an initial data exploration was conducted to understand the structure and composition of the dataset before proceeding with preprocessing. The `df.info()` function revealed that the dataset contains 2,111 records and 17 attributes. Among these, eight features are numerical, including Age, Height, Weight, FCVC, NCP, CH2O, FAF, and TUE. The remaining attributes, excluding the target variable, are categorical and include variables such as Gender, family_history_with_overweight, FAVC, CAEC, SMOKE, SCC, CALC, and MTRANS. The target variable, NObesyesdad was found to have seven unique classes, indicating that this is a multi-class classification task. To further explore the data, `df.nunique()` was used to determine the number of unique values per feature, helping to differentiate between high- and low-cardinality attributes. Summary statistics for the numerical features were obtained using `df.describe()`, which confirmed that values fall within realistic human ranges, suggesting no immediate data quality issues. Similarly, the `df.describe(include='object')` function was used to generate descriptive statistics for categorical features, highlighting the most frequent values and category distributions. These findings confirm that the dataset is structurally sound and ready for further analysis and preprocessing steps.

```

df.info()

df.nunique()

# numerical feature descriptive statistics
df.describe()

# list of categorical variables (excluding the target variable 'NObeyesdad')
categorical = [var for var in df.columns if df[var].dtype == 'object' and var != 'NObeyesdad']

# list of numerical variables (excluding the target variable 'NObeyesdad')
numerical = [var for var in df.columns if df[var].dtype != 'object' and var != 'NObeyesdad']

target = 'NObeyesdad'

# Unique values in the 'NObeyesdad' column in ascending order
NObeyesdad_categories = sorted(df['NObeyesdad'].unique())

# categorical feature descriptive statistics
df.describe(include='object').T.sort_values(by=['unique'], ascending=False)

```

Figure 4.2: Code Snippet for Initial Dataset Exploration

A pie chart was generated to visually represent the distribution of the class labels in the dataset, specifically the different categories of obesity levels. This visualization aids in identifying potential label imbalance issues that could introduce bias into the machine learning models. The pie chart, shown in Figure 4.3, illustrates a relatively balanced distribution across the obesity levels. To complement the visual analysis, the count of each class label was computed using the `value_counts()` method. The results confirmed that no significant class imbalance was present.



Figure 4.3: Pie Chart showing the Distribution of Obesity Level Classes

To further assess the data quality, we performed a thorough check for missing values using the `isnull().sum()` function. As shown in Figure 4.4, the result confirmed that no missing values were present across all columns. This eliminates the need for imputation techniques and allows us to proceed with confidence regarding data integrity.

```
-----
Missing Values Summary
-----
Total columns with missing values: 0

No missing values found in the dataset.
```

Figure 4.4: Checking Missing Values and Output Summary

In addition to missing values, duplicate entries were identified using the `duplicated().sum()` function. As shown in Figure 4.5, 24 duplicate entries were found and subsequently removed using `drop_duplicates()`. This ensures the dataset does not contain

redundant records that may negatively impact the model's learning process. After this step, a verification check confirmed that all duplicates had been successfully eliminated.

```
-----  
Duplicates Rows Summary  
-----  
Total duplicate rows before dropping: 24
```

Figure 4.5: Code Snippet for Checking Handling Duplicated Rows and Output Summary

Outlier detection was conducted using the Interquartile Range (IQR) method, which calculates the lower and upper bounds based on the first (Q1) and third (Q3) quartiles for each numerical feature. The summary of outlier counts is presented in Figure 4.6. Despite the detection of 699 rows with at least one outlier these rows were not removed. After checking the values, these values fall within normal human behavior or body measurements. So, they were not seen as errors or noise. Instead, they were kept in the dataset to keep the data as close to real-life situations as possible.

```
===== Outlier Detection (IQR Method) =====  
Age           : 167 outliers  
Height        : 1 outliers  
Weight        : 1 outliers  
FCVC          : 0 outliers  
NCP           : 577 outliers  
CH2O          : 0 outliers  
FAF           : 0 outliers  
TUE           : 0 outliers  
  
Total rows with at least one outlier: 699
```

Figure 4.6: Checking Outlier Detection and Output Summary

A consistency check was done on all categorical features to ensure data quality. Using `str.strip().lower()`, values were cleaned by removing spaces and converting to lowercase. The code looped through each categorical column and compared the original and cleaned unique values. As shown in the output (Figure 4.7), no inconsistent entries were found. This confirms the categorical data is clean and consistent, and the dataset is ready for the data preparation phase.

```

Show unique values in column 'Gender':
['Female' 'Male']

Show unique values in column 'family_history_with_overweight':
['yes' 'no']

Show unique values in column 'FAVC':
['no' 'yes']

Show unique values in column 'CAEC':
['Sometimes' 'Frequently' 'Always' 'no']

Show unique values in column 'SMOKE':
['no' 'yes']

Show unique values in column 'SCC':
['no' 'yes']

Show unique values in column 'CALC':
['no' 'Sometimes' 'Frequently' 'Always']

Show unique values in column 'MTRANS':
['Public_Transportation' 'Walking' 'Automobile' 'Motorbike' 'Bike']

===== Inconsistent Values Check =====
No inconsistent values found!

```

Figure 4.7: Checking Inconsistencies and Output Summary

4.2.2 Data Preparation

In this stage, the dataset underwent essential preprocessing to ensure compatibility with machine learning algorithms and improve predictive accuracy. This section outlines the key preprocessing steps applied to the obesity dataset, including label encoding of categorical variables, feature scaling, feature selection, and data splitting. These processes are necessary to transform the raw dataset into a numerical and normalized format suitable for machine learning algorithms.

Python code used to apply label encoding to binary, ordinal, and categorical features, as well as the multiclass target. Machine learning models typically require all input features to be numeric. Therefore, these features within the dataset were transformed using Label Encoding, which converts each unique category to an integer value. This process preserved the distinctness of categories while ensuring compatibility with various algorithms.

Binary features, such as Gender, FAVC (Frequent Consumption of High Caloric Food), and SMOKE, were encoded using the LabelEncoder class from the Scikit-learn library. Since

these variables consist of two classes (e.g., “no”/ “yes”), label encoding directly maps them to 0 and 1.

Ordinal features, such as CAEC (Consumption of Food Between Meals) and CALC (Alcohol Consumption), were encoded using ordinal encoding to preserve the inherent order of categories. A predefined sequence, [“no”, “Sometimes”, “Frequently”, “Always”] was used to reflect increasing frequency or intensity. This approach ensures that the encoded values such as 0, 1, 2, 3, accurately represent the ordinal relationship among categories, allowing machine learning models to interpret and leverage the progression in behavior patterns effectively. This is crucial, as incorrect encoding could distort feature importance and negatively impact model performance.

The MTRANS feature, a nominal categorical variable with multiple classes, was encoded using label encoding. Although the categories such as “Automobile”, “Bike”, “Motorbike”, “Public_Transportation”, “Walking” have no inherent order, label encoding was applied to convert them into numeric form for model compatibility.

The target variable, NObeyesdad, which contains multiple obesity-related class labels, was also encoded into integer values to support multiclass classification. All mappings between original and encoded values were stored and tabulated for interpretability (see Table 4.2).

Table 4.2: Sample Label Encoding Mappings for Selected Features

Feature	Original Value	Encoded Value
Gender	Female	0
	Male	1
family_history_with_overweight	no	0
	yes	1
FAVC	no	0
	yes	1

SMOKE	no	0
	yes	1
SCC	no	0
	yes	1
CAEC	Always	3
	Frequently	2
	Sometimes	1
	No	0
CALC	Always	3
	Frequently	2
	Sometimes	1
	No	0
MTRANS	Automobile	0
	Bike	1
	Motorbike	2
	Public_Transportation	3
	Walking	4
NObeyesdad	Insufficient_Weight	0
	Normal_Weight	1
	Obesity_Type_I	4
	Obesity_Type_II	5
	Obesity_Type_III	6
	Overweight_Level_I	2
	Overweight_Level_II	3

Following label encoding, feature scaling was applied to normalize numerical variables and enhance model learning. Two scalers were used based on feature types which are MinMaxScaler was applied to CH2O, FCVC, NCP, FAF, TUE, scaling values to the [0, 1] range whereas StandardScaler was applied Age, Height, Weight, standardizing them to a mean of 0 and standard deviation of 1. As a result, some standardized values may be negative, indicating that they fall below the mean of their respective feature distributions. Figure 4.8 shows the logic behind applying different scalers and the first few rows of the dataset after scaling. Feature scaling ensures that all features contribute proportionately, preventing those with larger ranges (e.g., Weight) from overpowering others (e.g., TUE).

Gender	Age	Height	Weight	family_history_with_overweight	FAVC	FCVC	NCP	CAEC	SMOKE	CH2O	SCC	FAF	TUE	CALC	MTRANS
0	-0.526613	-0.887408	-0.872985	1	0	0.5	0.666667	2	0	0.5	0	0.000000	0.5	3	3
0	-0.526613	-1.960788	-1.178508	1	0	1.0	0.666667	2	1	1.0	1	1.000000	0.0	2	3
1	-0.212507	1.044677	-0.376509	1	0	0.5	0.666667	2	0	0.5	0	0.666667	0.5	1	3
1	0.415705	1.044677	0.005395	0	0	1.0	0.666667	2	0	0.5	0	0.666667	0.0	1	4
1	-0.369560	0.830001	0.112328	0	0	0.5	0.000000	2	0	0.5	0	0.000000	0.0	2	3

Figure 4.8: Cleaned Obesity Dataset

Next, feature selection process was conducted on the dataset. In this study, four feature selection techniques were applied independently such as Information Gain (Mutual Information), Gain Ratio, Correlation heatmap and the ReliefF algorithm. These techniques were chosen based on their varied mathematical foundations and effectiveness in capturing different aspects of feature relevance.

Information Gain (IG) evaluates how much knowing a particular feature reduces uncertainty about the target variable. It was computed using the `mutual_info_classif()` function from Scikit-learn, which is suitable for multiclass classification tasks. As shown in Figure 4.9 below, it visualizes the bar chart of Information Gain scores across all features. Features such as Weight, Age, and Height showed the highest contribution to reducing entropy in the target variable.

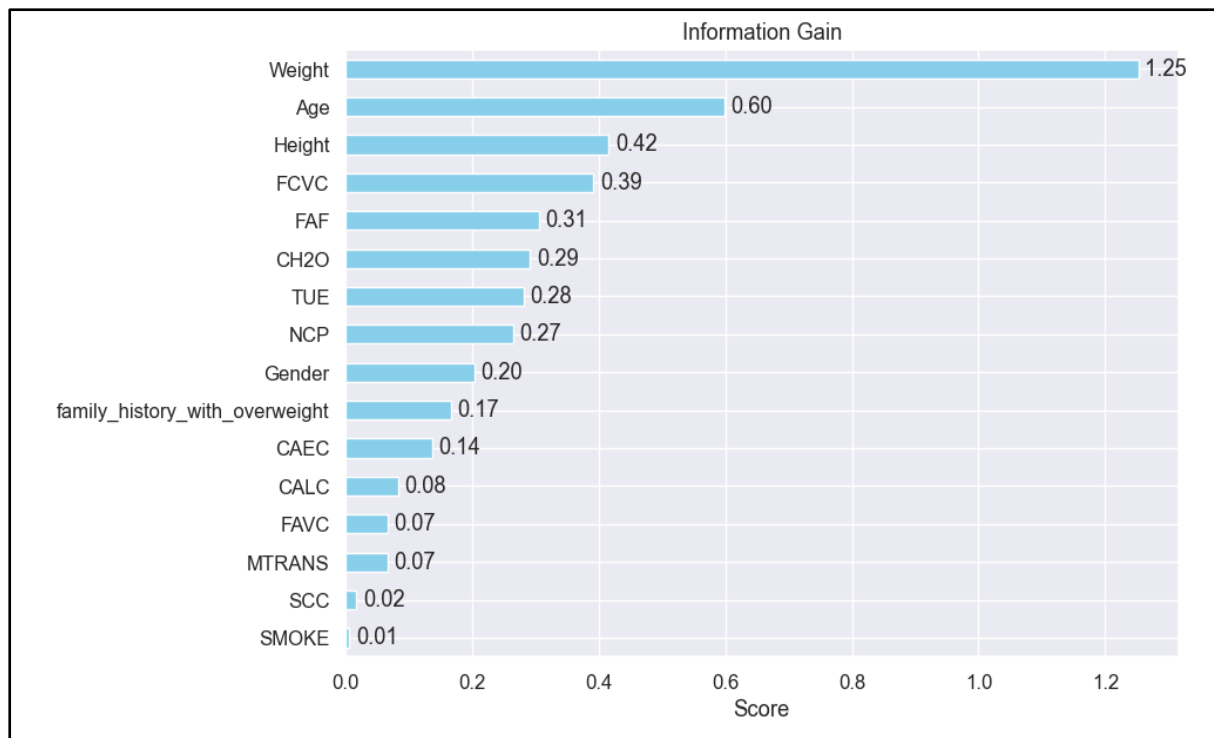


Figure 4.9: Corresponding Bar Plot for Information Gain

To further refine this approach, Gain Ratio (GR) was calculated. This method adjusts the bias of IG towards features with many unique values by dividing the mutual information by the intrinsic value (entropy) of the feature. A custom function was implemented as shown in Figure 4.10 to compute this ratio for each feature. The result, also illustrated in the figure, revealed that features like `family_history_with_overweight` had the highest gain ratios. This adjustment made the importance ranking more balanced and interpretable, particularly for categorical variables.

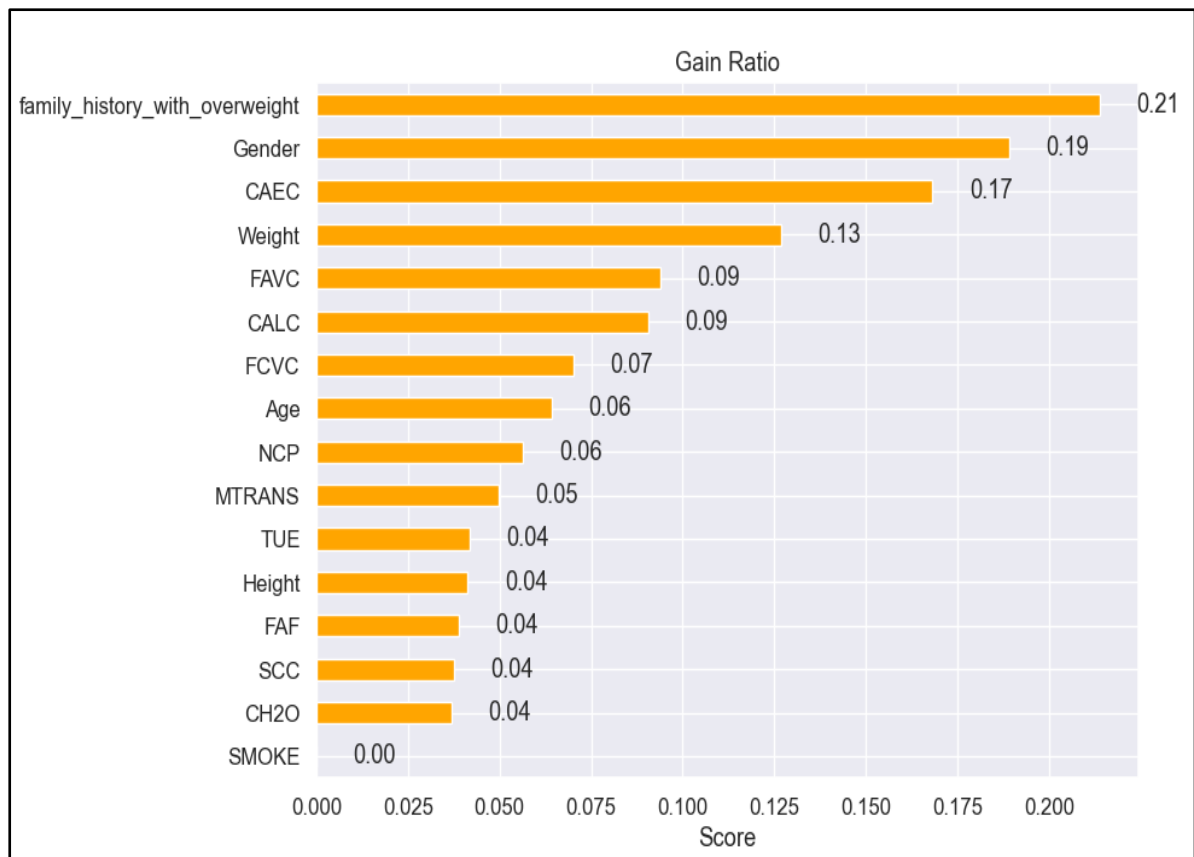


Figure 4.10: Corresponding Bar Plot for Gain Ratio

To assess linear relationships, the correlation heatmap was computed between each feature and the encoded target variable. The encoded form of NObeyesdad was obtained using `pd.factorize()`, making it suitable for correlation computation. The relevant code is presented in Figure 4.11, the corresponding heatmap indicate that Weight was the most linearly correlated with the target variable.

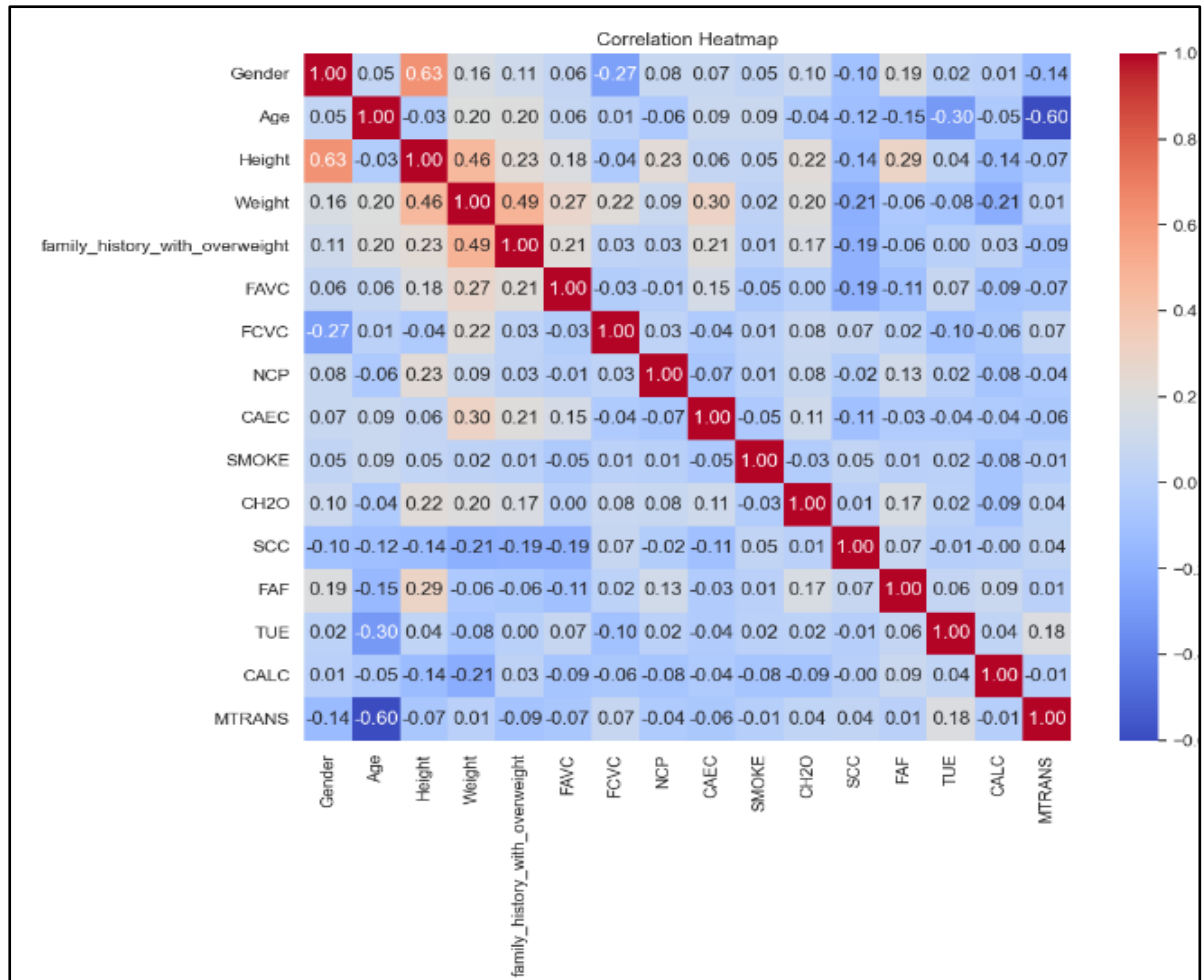


Figure 4.11: Corresponding Heatmap for Correlation Matrix

The fourth method, ReliefF, is a robust, distance-based algorithm that evaluates a feature's ability to distinguish between neighboring instances from different classes. This method is particularly suited for multiclass problems and does not assume feature independence. Due to ReliefF relies on distance calculations to compare feature values between instances, it is sensitive to the scale of the input data. Therefore, all features were normalized using MinMaxScaler() to ensure they fall within a uniform [0, 1] range, preventing any single feature from dominating due to larger numerical values. The algorithm was applied using ReliefF(n_neighbors=30), with inputs scaled accordingly. As illustrated in Figure 4.12, ReliefF implementation depicts the normalized importance scores for each feature. Here, Weight, Gender, and Height ranked among the top predictors.

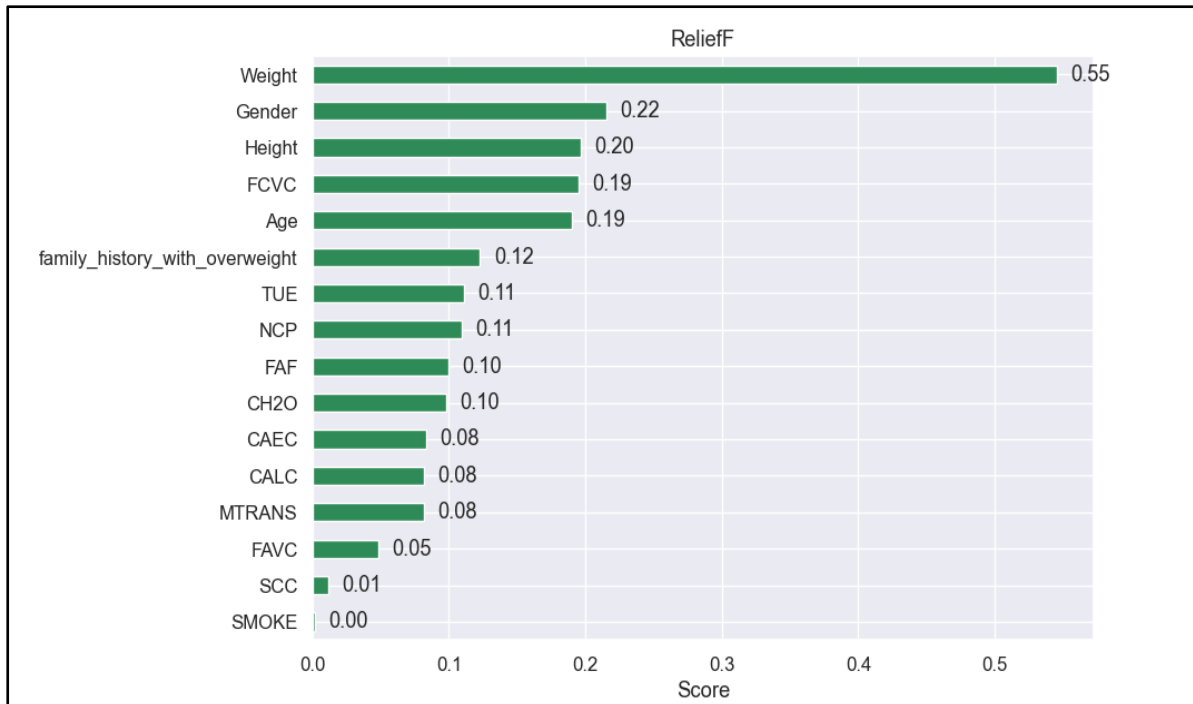


Figure 4.12: Corresponding Bar Plot for ReliefF

Each feature selection method gives scores in different ranges. To compare them fairly, all scores were scaled between 0 and 1 using Min-Max normalization. This made the values consistent across methods. Then, the average score for each feature was calculated using the normalized scores from all four methods. This helped show which features were important in general, not just in one method.

The Union strategy was used to keep any feature that was important in at least one method ($\text{score} \geq 0.1$). This avoided missing useful features that only one method could detect. The final selected features are listed in Figure 4.13, which presents a sorted summary table and the final selected list from the implementation. These features were used for subsequent model training and evaluation.

Final Selected Features Based on UNION (≥ 1 methods ≥ 0.1):						
	Information Gain	Gain Ratio	Correlation	Relieff	Average Score	
Weight	1.000	0.594	1.000	1.000	0.899	
family_history_with_overweight	0.128	1.000	0.391	0.223	0.435	
Gender	0.159	0.885	0.159	0.393	0.399	
FCVC	0.309	0.328	0.510	0.355	0.376	
CAEC	0.105	0.787	0.244	0.149	0.321	
Age	0.475	0.301	0.080	0.346	0.301	
Height	0.329	0.194	0.168	0.359	0.262	
FAVC	0.049	0.439	0.322	0.085	0.224	
NCP	0.208	0.264	0.190	0.198	0.215	
FAF	0.240	0.182	0.183	0.180	0.196	
CALC	0.062	0.422	0.110	0.148	0.185	
CH2O	0.229	0.172	0.081	0.177	0.165	
TUE	0.221	0.196	0.017	0.201	0.159	
MTRANS	0.048	0.234	0.070	0.147	0.125	
SCC	0.009	0.176	0.233	0.017	0.109	
Total features selected: 15						

Figure 4.13: Final List of Selected Features (After Filtering)

After completing the preprocessing and feature selection steps, the dataset was divided into training and testing sets using the `train_test_split` function from Scikit-learn, as shown in Figure 4.14. Stratified sampling was applied to preserve the class distribution of the multiclass target variable (NObyesdad) in both subsets to ensure balanced representation of each class across the training and testing sets. The training set (80%) was used to train and tune the models, while the testing set (20%) was reserved exclusively for evaluating model performance on unseen data. This approach enhances the reliability and generalizability of the results. After this step, the dataset was ready for the modelling phase.

```

Preparing the dataset...
Dataset split completed:
- Training set: 1669 samples
- Testing set:  418 samples
- Features:    15

```

Figure 4.14: Train-Test Split with Stratified Sampling (80:20)

4.2.3 Modeling

In this section, five machine learning models were developed and trained using the training dataset which are Decision Tree, Random Forest, Support Vector Machine (SVM), XGBoost, and Logistic Regression. Each model was first trained with default hyperparameters, followed by hyperparameter tuning to improve performance. A consistent random state (`random_state = 42`) was used across all models for reproducibility. Hyperparameter tuning was conducted using either `GridSearchCV` or `RandomizedSearchCV`, with 5-fold cross-validation (`cv=5`). Grid Search tested all combinations of defined parameters. Randomized Search tested selected combinations. These methods helped find the best settings for each model based on performance scores. Table 4.3 presents the tested hyperparameter ranges and the best values selected for each classification model.

Table 4.3: Hyperparameter Search Space and Optimal Values for Each Classification Model

Classifier	Hyperparameter	Search Space	Value
DecisionTreeClassifier (RandomizedSearchCV)	criterion	('gini', 'entropy')	entropy
	max_depth	(None, 10, 20, 30)	30
	min_samples_split	(2, 5, 10)	10
	min_samples_leaf	(1, 2, 4)	2
	max_features	('sqrt', 'log2', None)	None
	splitter	('best', 'random')	best

RandomForestClassifier (RandomizedSearchCV)	n_estimators	(100, 200, 300)	100
	criterion	('gini', 'entropy')	entropy
	max_depth	(None, 10, 20, 30)	10
	min_samples_split	(2, 5, 10)	5
	min_samples_leaf	(1, 2, 4)	1
	max_features	('sqrt', 'log2', None)	None
	bootstrap	(True, False)	True
SVM Classifier (GridSearchCV)	C	(0.1, 1, 10)	10
	kernel	('linear', 'rbf')	rbf
	gamma	('scale')	scale
	probability	True	True
XGBClassifier (RandomizedSearchCV)	n_estimators	(100, 200)	200
	max_depth	(3, 5, 7)	5
	learning_rate	(0.05 to 0.2)	0.1
	subsample	(0.8, 1.0)	0.8
	colsample_bytree	(0.8, 1.0)	0.8
LogisticRegression Classifier (GridSearchCV)	penalty	'elasticnet'	elasticnet
	solver	saga	saga
	L1 ratio	0.5	0.5
	C	1.0	1.0
	class_weight	balanced	balanced
	max_iter	1000	1000

4.2.4 Evaluation

To evaluate the predictive performance of the trained machine learning models, accuracy, precision, recall, and F1-score were utilized. Not only that, to evaluate the effectiveness of hyperparameter tuning, model predictions were generated on both training and test sets before and after tuning. The `predict()` and `predict_proba()` methods were used to compute predicted labels and probabilities (if available). Accuracy scores were then calculated using the `accuracy_score()` function through a custom `print_metrics()` method. These scores were recorded in a structured `score_table` dictionary for each model. As illustrated in Figure 4.15, this process enabled a clear comparison of model performance before and after tuning, particularly in terms of generalization on testing dataset.

```
y_pred_train_tuned = best_model.predict(X_train)
y_pred_test_tuned = best_model.predict(X_test)

y_proba_train_tuned = best_model.predict_proba(X_train) if hasattr(best_model, "predict_proba") else None
y_proba_test_tuned = best_model.predict_proba(X_test) if hasattr(best_model, "predict_proba") else None

tuned_train_acc = print_metrics(y_train, y_pred_train_tuned, y_proba_train_tuned, "After Tuned - Train")
tuned_test_acc = print_metrics(y_test, y_pred_test_tuned, y_proba_test_tuned, "After Tuned - Test")

score_table.append({
    'Model': name,
    'Train Acc': train_acc,
    'Test Acc': test_acc,
    'Tuned Train Acc': tuned_train_acc,
    'Tuned Test Acc': tuned_test_acc
})
```

Figure 4.15: Code Snippet for Computing Accuracy Before and After Hyperparameter Tuning for Both Training and Testing Dataset

As illustrated in Figure 4.16, the `classification_report()` function from the `sklearn.metrics` module was used to generate a detailed summary of performance for each class. This report includes class-wise precision, recall, F1-score, and support, which help to evaluate how well the model performs across all target classes. The report was particularly useful in identifying any imbalance or inconsistency in classification performance among classes.

```
# Classification Report
class_report = classification_report(y_test, y_pred_test_tuned, output_dict=True)
print(f"\n 📄 Classification Report - {name}:\n{classification_report(y_test, y_pred_test_tuned)}")
classification_reports[name] = class_report
```

Figure 4.16: Code Snippet for Generating Classification Reports for each Model After Tuning

In addition to numeric performance metrics, confusion matrices were used to visualize the classification results. As shown in Figure 4.17, the confusion matrix was computed using the `confusion_matrix()` function and visualized as a heatmap using the `seaborn` library. The matrix provides a clear indication of how many instances were correctly or incorrectly classified for each class. By observing the diagonal and off-diagonal values, misclassification patterns can be identified, thus offering insight into specific weaknesses of each model.

```
# Confusion Matrix
cm = confusion_matrix(y_test, y_pred_test_tuned)
plt.figure(figsize=(12, 5))

plt.subplot(1, 2, 1)
sns.heatmap(cm, annot=True, fmt='d', cmap='YlOrRd', xticklabels=class_names, yticklabels=class_names)
plt.title(f"Confusion Matrix - {name}")
plt.xlabel("Predicted")
plt.ylabel("Actual")
```

Figure 4.17: Code Snippet for Generating and Displaying Confusion Matrix for each Model

Furthermore, after tuning, Receiver Operating Characteristic (ROC) curve was plotted for models that support probability outputs. This helped assess how well each model differentiates between classes. As shown in Figure 4.18, ROC curves for each class were generated using the `roc_curve()` and `auc()` functions, and their Area Under the Curve (AUC) scores were calculated. Higher AUC scores indicate better class separation and overall performance.

```

# ROC-AUC Curve AFTER tuning (if available)
if y_proba_test_tuned is not None:
    y_test_bin = label_binarize(y_test, classes=class_names)
    fpr, tpr, roc_auc = {}, {}, {}
    for i in range(y_test_bin.shape[1]):
        fpr[i], tpr[i], _ = roc_curve(y_test_bin[:, i], y_proba_test_tuned[:, i])
        roc_auc[i] = auc(fpr[i], tpr[i])
    roc_auc_scores[name] = (fpr, tpr, roc_auc)

# Plot ROC-AUC Curve
plt.subplot(1, 2, 2)
for i in range(y_test_bin.shape[1]):
    plt.plot(fpr[i], tpr[i], label=f"Class {i} AUC = {roc_auc[i]:.2f}")
plt.plot([0, 1], [0, 1], 'k--')
plt.title(f"ROC Curve - {name}")
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.legend()

# Update best model if this one has the highest average AUC
avg_auc = np.mean(list(roc_auc.values()))
if avg_auc > best_auc:
    best_auc = avg_auc
    best_model_obj = best_model
    best_model_name = name

```

Figure 4.18: Code Snippet for Calculating and Plotting ROC curves with AUC Scores Per Class After Tuning

4.2.5 Deployment

From the evaluation metrics in the previous sections, Random Forest achieved the best overall performance in terms of accuracy and balanced classification outcomes. Therefore, this model was selected as the prototype to be deployed. For the deployment platform, Hugging Face Spaces was chosen due to its free and accessible hosting for machine learning web applications. Gradio was used due to its simplicity and capability to quickly build interactive interfaces for machine learning models. The entire interface and backend logic were implemented in a Python script named `app.py`.

The trained Random Forest model was serialized and saved using the `joblib` library to facilitate loading during deployment. This process ensured that the model could be efficiently reused without retraining. A screenshot of the code used to save the trained model is shown in Figure 4.19. During runtime, the model was reloaded using `joblib.load()` within the `app.py`

script. The prediction logic was encapsulated in a function that accepted user inputs, performed necessary preprocessing such as label encoding, and returned a prediction label.

```
# Save the trained Random Forest model using joblib
import joblib
model = joblib.load("random_forest_best_model.pkl")
```

Figure 4.19: Code Snippet for Saving the Trained Random Forest Model using the Joblib Library

The Gradio interface was designed to allow users to input their data through sliders and radio buttons, ensuring ease of use and validation of input types. Each input field corresponded to one of the model features, such as personal information, eating habits, and physical activity. The interface was connected to the prediction function and styled with a title, description, and HTML-formatted output to enhance readability.

Finally, the application was successfully deployed on Hugging Face Spaces to provide a user-friendly web interface for obesity prediction. Figure 4.20 displays the implementation of the deployed application interface to illustrate the interactive elements and allow user to input data for real time prediction. Figure 4.21 and Figure 4.22 shows the output display with prediction category result, user input summary section and prediction probabilities across seven obesity levels. Users can access and interact with the deployed prototype at the following URL:

<https://huggingface.co/spaces/WinnieTRP/obesity-prediction>



Obesity Level Prediction

This RF model predicts obesity levels based on several input features

Please provide your information to predict your obesity level.

PERSONAL INFORMATION

Gender

☐ Male ☐ Female

Age

Height (m)

Weight (kg)

Family History with Overweight

☐ yes ☐ no

EATING HABITS

PHYSICAL ACTIVITY

PREDICT OBESITY LEVEL

RESET FORM

Figure 4.20: Web application for User Input Real-Time Prediction

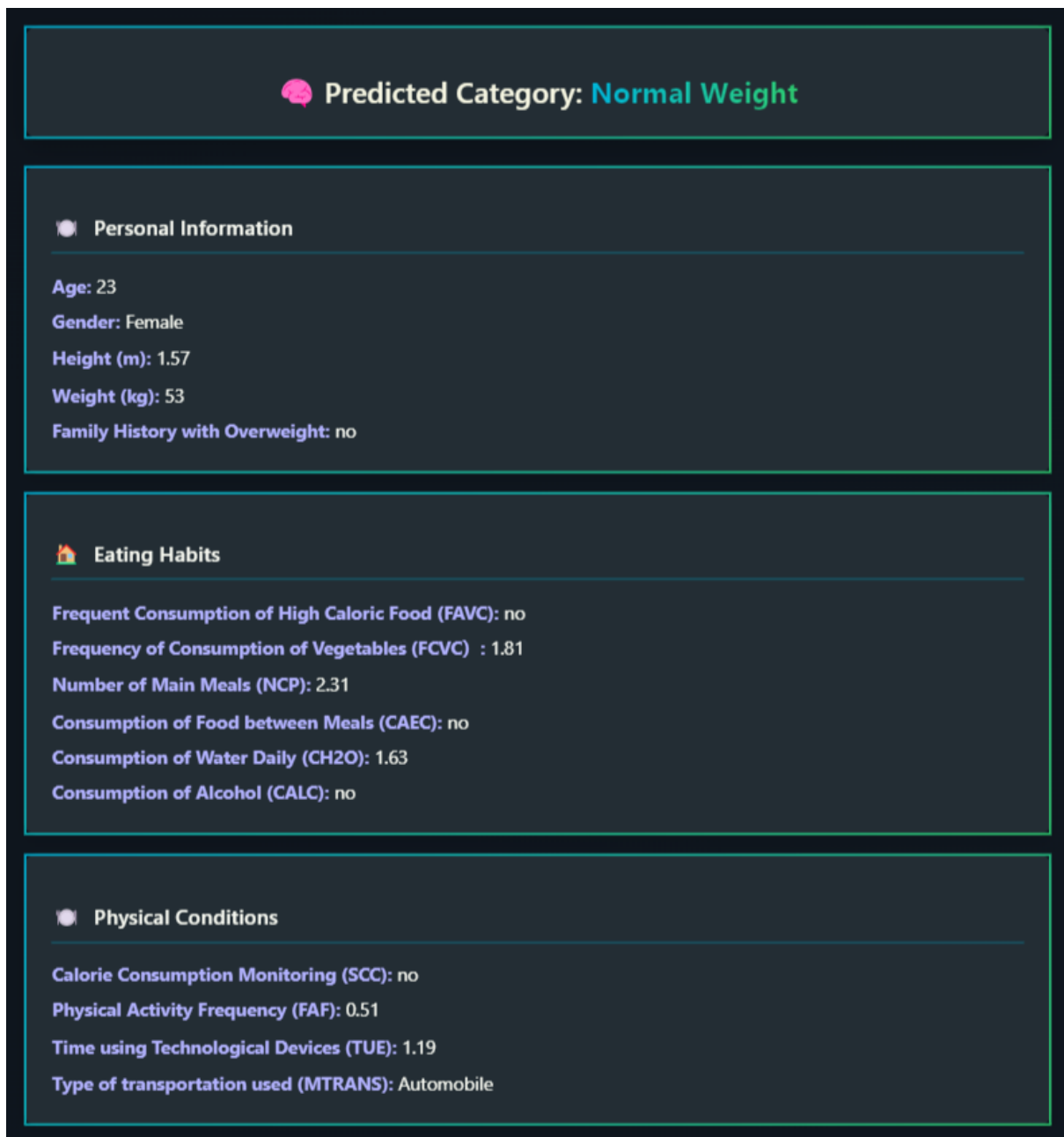


Figure 4.21: Output Display with Prediction Category and User Input Summary Section

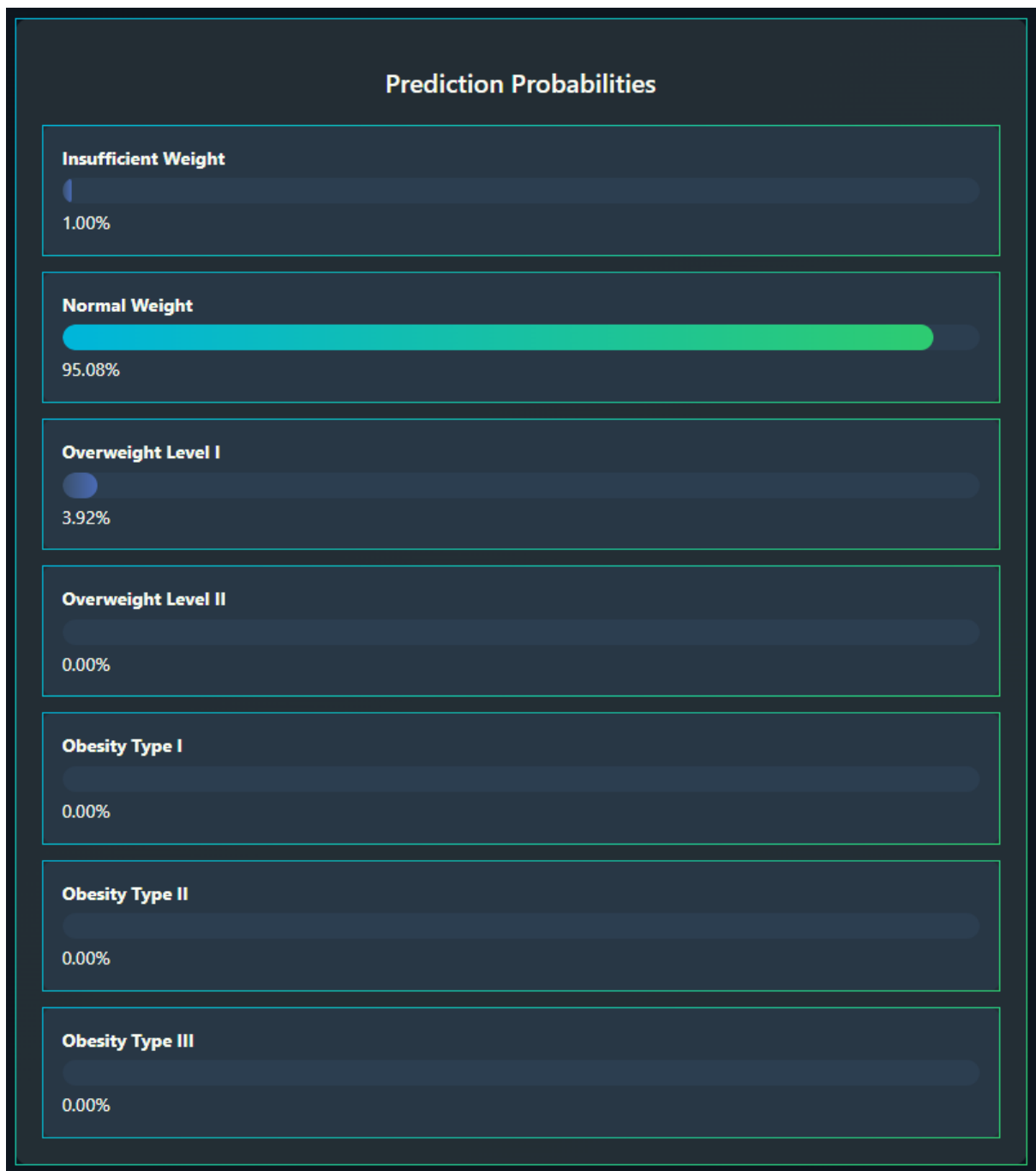


Figure 4.22: Output Display with Probabilities

4.3 Summary

To conclude, the processes and codes used to implement the models and predictions is discussed in detail in this chapter. The implementation process is explained step by step with sample of code. Overall, the proposed Random Forest model had obtained the best result amongst the other models. The results evaluation and comparison would be discussed in the next chapter and there are some insights of refinement as future works can be seen. Lastly, the prototype of this obesity prediction model has been successfully implemented and deployed.

Chapter 5 : Result and Discussion

5.1 Introduction

In this chapter, the model evaluation and the results obtained from training and testing will be discussed in detail for obesity predictions. The evaluation results will be obtained with some evaluation metrics mentioned in previous chapter which are accuracy, precision, recall, f1-score, ROC-AUC value and confusion matrix. Besides, the prediction results obtained for the models will be discuss and analyse in detail to understand the effectiveness and limitations of each model.

5.2 Results

5.2.1 Result for Decision Tree

Table 5.1 presents the performance of the Decision Tree (DT) model before and after hyperparameter tuning on the test set. Initially, the model showed an accuracy of 92.8%, with corresponding values for precision at 0.930, recall, and F1-score at 0.928, and an AUC score of 0.958. After hyperparameter tuning, the model exhibited consistent improvements across all evaluation metrics. Accuracy on test set increased to 94.7%, while precision, recall, and F1-score all rose to 0.947. The AUC score also improved to 0.980. Therefore, tuning helped because it changed the model settings like maximum depth and minimum samples per split, which reduced overfitting. This made the tree learn general patterns better.

Table 5.1: Performance of DT Model Before and After Tuning on Test Set

Metric	Test Set (Default)	Test Set (After Tuned)
Accuracy	0.928	0.947
Precision	0.930	0.948
Recall	0.928	0.947
F1 Score	0.928	0.947
AUC Score	0.958	0.980

Figure 5.1 display the confusion matrix and ROC curve for the tuned Decision Tree model on the test set. The model achieved high performance across all metrics, with minimal misclassifications. For example, class 2 (Overweight_Level_I) had 6 instances misclassified as class 1 (Normal_Weight) and 5 instances misclassified as class 3 (Overweight_Level_II). These errors may be due to overlapping behaviors such as NCP and FAF. These class groups share close feature values, which makes it harder to split them cleanly. Tuning helped the model avoid creating too many small branches that increase these mistakes. The ROC curves further validate this, with AUC scores ranged from 0.94 to 1.00, with an average AUC of 0.980.

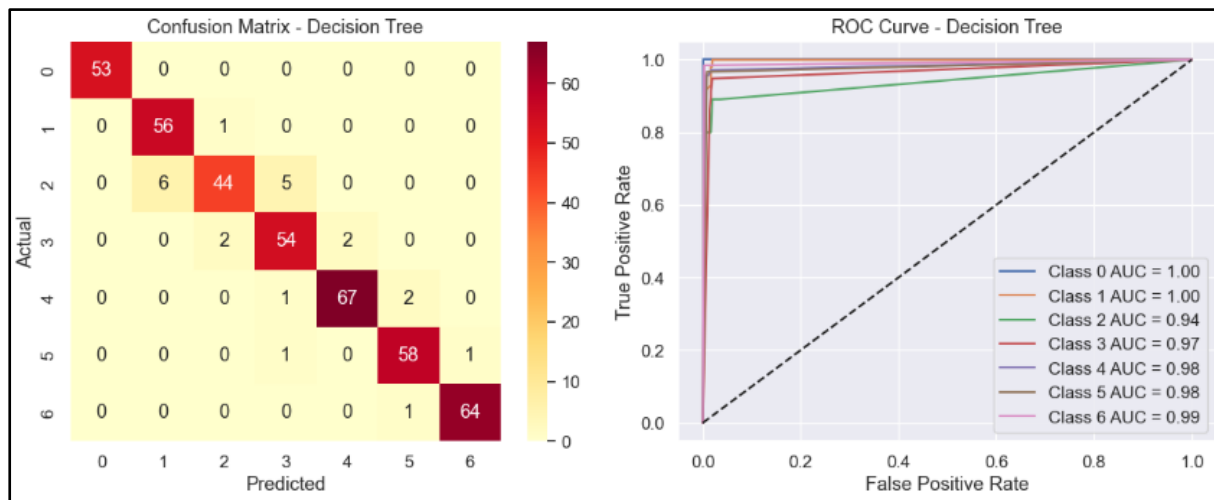


Figure 5.1: Confusion Matrix and ROC Curve for the Tuned DT Model on the Test Set

5.2.2 Result for Random Forest

Table 5.2 presents the performance of the Random Forest (RF) model before and after hyperparameter tuning on the test set. Initially, the model showed an accuracy of 93.8%, with corresponding precision at 0.941, recall, and F1-score all at 0.938, and an AUC score of 0.998. After hyperparameter tuning, the model demonstrated further improvements across all metrics. Accuracy on test set increased to 96.7%, with precision, recall, and F1-score all rising almost to 0.968. Only the AUC score slightly decreased to 0.996. This maybe because tuning made the model more confident in some predictions, which reduced sensitivity to rare or borderline

cases. AUC measures how well the model ranks positive classes, so even small changes in probability thresholds can affect it.

Table 5.2: Performance of RF Model Before and After Tuning on Test Set

Metric	Test Set (Default)	Test Set (After Tuned)
Accuracy	0.938	0.967
Precision	0.941	0.968
Recall	0.938	0.967
F1 Score	0.938	0.967
AUC Score	0.998	0.996

Figure 5.2 display the confusion matrix and ROC curve for the tuned RF model on the test set. The model achieved high performance across all metrics, with minimal misclassifications. For example, class 2 (Overweight_Level_I) had 5 instances misclassified as class 3 (Overweight_Level_II) and 1 instance misclassified as class 1(Normal_Weight). This is likely because these classes have similar values for features like NCP, FCVC, and FAF. These patterns show that even strong models can confuse samples near class boundaries. The ROC curves further validate this, with AUC scores ranged from 0.99 to 1.00, with an average AUC of 0.996. The model worked better because it uses multiple trees, which helps reduce variance. It captured interactions between features like family history with overweight, CAEC, CH2O, and Age, which are important in predicting obesity. It handled class overlap well and was more stable with noisy features, making it the most reliable model in this study.

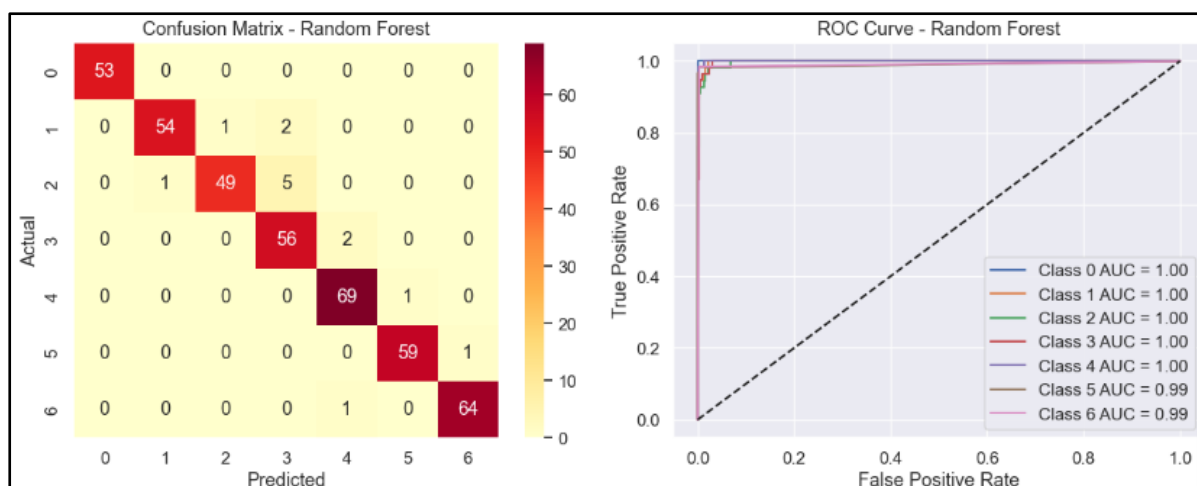


Figure 5.2: Confusion Matrix and ROC Curve for the Tuned RF Model on the Test Set

5.2.3 Result for Support Vector Machine

Table 5.3 summarizes the performance of the Support Vector Machine (SVM) model on the test set before and after hyperparameter tuning. Before tuning, the model achieved an accuracy of 90.7%, with precision at 0.910, recall, and F1-score all at 0.907, and an AUC score of 0.996. After tuning, the model showed noticeable improvements across all metrics. Accuracy on the test set rose to 94.7%, while precision, recall, and F1-score all increased to 0.949. The AUC score also slightly increased to 0.999. This is because SVM used a basic kernel and default regularization before tuning. This caused limited separation between overlapping classes. After tuning, a better kernel like RBF and adjusted C value gave more flexible decision boundaries. This helped improve how the model separated non-linear patterns.

Table 5.3: Performance of SVM Model Before and After Tuning on Test Set

Metric	Test Set (Default)	Test Set (After Tuned)
Accuracy	0.907	0.947
Precision	0.910	0.949
Recall	0.907	0.947
F1 Score	0.907	0.947
AUC Score	0.996	0.999

Figure 5.3 display the confusion matrix and ROC curve for the tuned SVM model on the test set. The model achieved high performance across all metrics, with minimal misclassifications. For example, class 1 (Normal_Weight) had 4 instances misclassified as class 0 (Insufficient_Weight) and 3 instances misclassified as class 2 (Overweight_Level_I). The ROC curve supports this, with all classes achieving near-perfect AUC scores (1.00) and an average AUC of 0.999 across all classes. The AUC is high because SVM gives strong class probability margins, even when actual predicted labels are sometimes wrong. It can confidently rank class scores, but if two classes have close values (like class 1 and 2), it may still pick the wrong one. So, AUC increased because ranking got better.

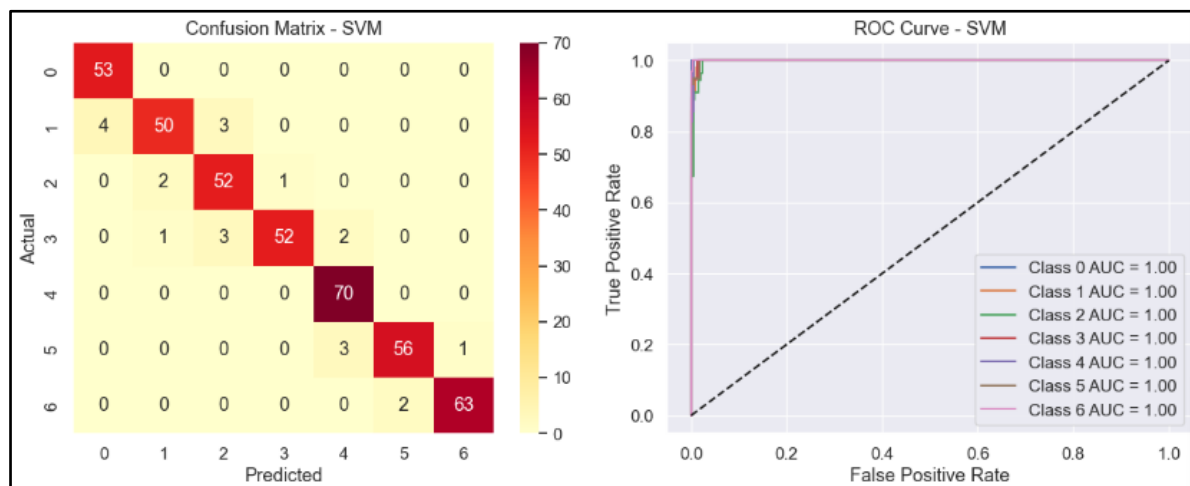


Figure 5.3: Confusion Matrix and ROC Curve for the Tuned SVM Model on the Test Set

5.2.4 Result for XGBoost

Table 5.4 displays the performance of the XGBoost model on the test set before and after hyperparameter tuning. Prior to tuning, the model achieved high performance, with accuracy, precision at 0.960, recall, and F1-score all at 0.959, and an AUC score of 0.999. After tuning, the metrics remained unchanged, with consistent performance observed across all values. This means the model was already well set up from the start. XGBoost often works well without tuning, and in this case, the tuning did not change the result. The default settings were already a good match for this dataset. XGBoost uses gradient boosting, which helps it focus on mistakes and learn from them step by step. In this dataset, features like CH2O, FAF,

FCVC, NCP, and CALC have small but important roles. XGBoost was able to capture these effects and find useful patterns. It also learns feature importance and how features work together. This is helpful when trying to classify difficult cases, like Overweight_Level_I. So, tuning made little difference because the model was already doing well.

Table 5.4: Performance of XGBoost Model Before and After Tuning on Test Set

Metric	Test Set (Default)	Test Set (After Tuned)
Accuracy	0.959	0.959
Precision	0.960	0.960
Recall	0.959	0.959
F1 Score	0.959	0.959
AUC Score	0.999	0.999

Figure 5.4 display the confusion matrix and ROC curve for the tuned XGBoost model on the test set. The model achieved high performance across all metrics, with minimal misclassifications. For example, class 2 (Overweight_Level_I) had 4 instances misclassified as class 3 (Overweight_Level_II) and 2 instances misclassified as class 1 (Normal_Weight). The ROC curve supports this, with all classes achieving near-perfect AUC scores (1.00) and an average AUC of 0.999 across all classes.

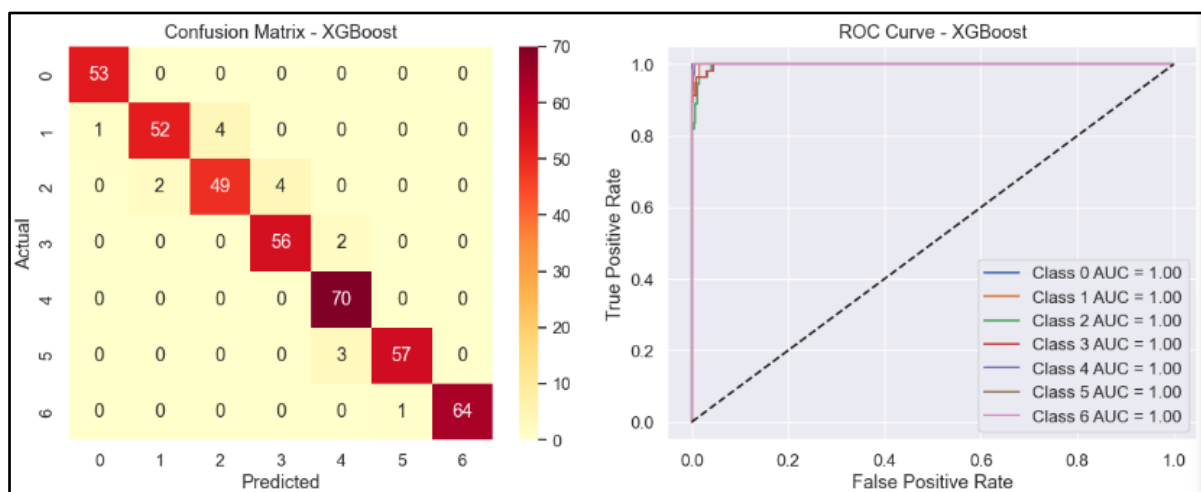


Figure 5.4: Confusion Matrix and ROC Curve for the Tuned XGBoost Model on the Test Set

5.2.5 Result for Logistic Regression

Table 5.5 presents the performance of the Logistic Regression (LR) model before and after hyperparameter tuning on the test set. Initially, the model achieved an accuracy of 87.1%, with a precision of 0.872, recall of 0.871, F1-score of 0.869, and an AUC score of 0.986. After hyperparameter tuning, the model's performance improved across all metrics. The accuracy on the test set increased to 90.0%, precision to 0.902, recall to 0.900, F1-score to 0.899, and AUC score to 0.991. This improvement can be attributed to better hyperparameter selection that enhanced the model's ability to generalize. Specifically, using a well-balanced regularization strength (C), the appropriate penalty type, and a compatible solver helped the model achieve a better bias-variance trade-off. For example, the tuned model may have used an optimal C value that avoided under- or over-regularization, preserving important features while minimizing noise. The solver, saga likely offered better convergence properties with the chosen penalty, improving optimization. The tuning refined the model's internal parameters.

Table 5.5: Performance of LR Model Before and After Tuning on Test Set

Metric	Test Set (Default)	Test Set (After Tuned)
Accuracy	0.871	0.900
Precision	0.872	0.902
Recall	0.871	0.900
F1 Score	0.869	0.899
AUC Score	0.986	0.991

Figure 5.5 displays the confusion matrix and ROC curve for the tuned LR model on the test set. Among all models, logistic regression performed the worst, with higher misclassification rates, particularly for classes 1 to 4. For example, class 1 (Normal_Weight) had 8 instances misclassified as class 0 (Insufficient_Weight), and class 4 (Obesity_Type_I) had 6 instances misclassified as classes 5 (Obesity_Type_II). In contrast, class 0

(Insufficient_Weight) achieved high correct predictions. The ROC curve indicates moderate to strong class separability, with AUC scores ranging from 0.97 to 1.00 and an average AUC of 0.991. Logistic Regression draws straight lines to split the classes. This makes it hard to predict correctly when class boundaries are not clear. This is why misclassification is still high, even after tuning.

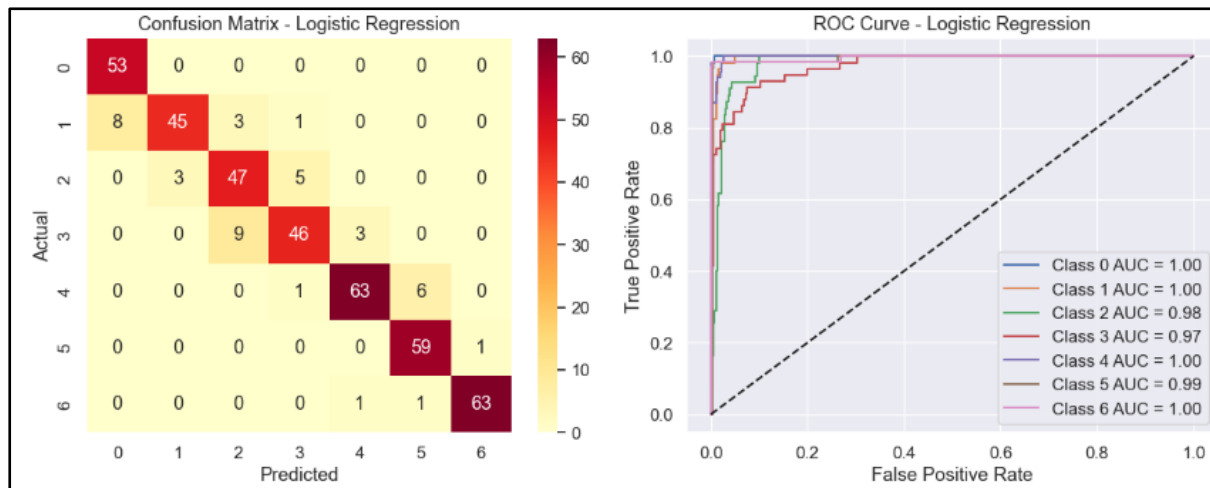


Figure 5.5: Confusion Matrix and ROC Curve for the Tuned LR Model on the Test Set

5.3 Comparison of Results

This section presents a detailed comparison of the five machine learning models which are Decision Tree, Random Forest, Support Vector Machine (SVM), XGBoost, and Logistic Regression after hyperparameter tuning, as shown in Table 5.6, Figure 5.6, and Figure 5.7. The evaluation was conducted using several metrics which are accuracy, precision, recall, F1-score, and Area Under the Curve (AUC), all measured on the test set to reflect the models' generalization performance. Importantly, all models showed performance improvements after hyperparameter tuning, confirming that tuning played a critical role in enhancing their ability to generalize to unseen data.

Among the models evaluated, Random Forest had the best performance overall. It achieved the highest accuracy (0.967), precision (0.968), recall (0.967), F1-score (0.967), and AUC score (0.996). This model used ensemble learning with multiple decision trees. Each tree was trained on a random subset of features, which helped reduce overfitting. This is important

because the dataset has both ordinal and nominal features, and random forest can handles mixed data types well. During preprocessing, features like 'CAEC', 'CALC', and 'FAVC' were encoded and selected. These features are important in eating habits. Random Forest could capture their non-linear interactions better than other models. Compared to related work like Fernandes et al. (2023), who also used a similar approach, the performance in this project is consistent and reliable.

While SVM and XGBoost also showed excellent AUC scores of 0.999, their accuracies were slightly lower than random forest. SVM achieved an accuracy of 0.947 using an RBF kernel, which is good for non-linear classification. It benefitted from feature scaling like StandardScaler, especially for continuous variables like 'CH2O', 'FAF', and 'TUE'. However, SVM requires very precise tuning of hyperparameters such as C and gamma to perform optimally. In this project, while SVM achieved high class separation as shown by the AUC, its slightly lower accuracy and recall suggest that it struggled with capturing the broader range of feature interactions present in the dataset, which tree-based models like Random Forest handled better.

However, XGBoost also delivered strong results, with key metrics such as accuracy, recall, F1-score at 0.959, precision at 0.960 and AUC at 0.999. XGBoost is powerful due to its gradient boosting mechanism, which learns from errors over iterations and includes built-in regularization to prevent overfitting. It also handles mixed data types well. However, in this project, XGBoost did not outperform Random Forest. One possible reason is that XGBoost often shows greater advantages when working with larger or noisier datasets, where regularization becomes more critical. Since this dataset was relatively clean and moderately sized, random forest's averaging method was more effective and efficient.

The Decision Tree model gave good results, with all metrics at 0.947 and AUC at 0.980. This algorithm is simple and interpretable, making it useful for understanding decision logic.

However, a single decision tree lacks the ability to generalize across various patterns because it learns from the data using a fixed structure and can become overly tuned to specific data splits. Even though, I tuned hyperparameters such as `max_depth`, `min_samples_split`, and `min_samples_leaf`, a single tree still lacks the stability and predictive strength that ensemble methods offer. Random forest, by combining multiple trees, avoids these issues and captures more generalized patterns, leading to better results. This aligns with Devi et al. (2022), which showed ensemble models consistently outperform single-tree models on similarly sized datasets.

However, Logistic Regression produced the weakest performance, with an accuracy of 0.900 and AUC of 0.991. This model assumes a linear relationship between the features and the target variable. However, the dataset includes several non-linear and interacting features, such as 'Gender', 'NCP', and 'SCC', where the influence on obesity levels may depend on how they combine. Logistic Regression could not capture these relationships effectively. I also used L1 regularization with $C=10$ to reduce underfitting, but even with this tuning, the model remained limited due to its inherent assumptions and cause it worse than before tuning. In contrast, studies like Ferdowsy et al. (2021) that reported better results with Logistic Regression likely used datasets that were simpler or more linearly separable, unlike the one used in this project.

Looking at ROC curves, all models reached high AUC scores close to 1. This means they were good at separating classes. But only Random Forest achieved both high AUC and high accuracy, which shows it is reliable across different metrics.

In conclusion, Random Forest is the most effective model in this project. It handled various feature types well, captured complex interactions, and maintained robustness against overfitting. Although SVM and XGBoost showed excellent AUC scores, Random Forest

provided the best balance of accuracy, stability, and generalization, making it the most suitable choice for this classification task.

Table 5.6: Summary of Performance Comparison After Tuning on Test Set

Model	Accuracy	Precision	Recall	F1-Score	AUC Score
Decision Tree	0.947	0.948	0.947	0.947	0.980
Random Forest	0.967	0.968	0.967	0.967	0.996
SVM	0.947	0.949	0.947	0.947	0.999
XGBoost	0.959	0.960	0.959	0.959	0.999
Logistic Regression	0.900	0.902	0.900	0.899	0.991

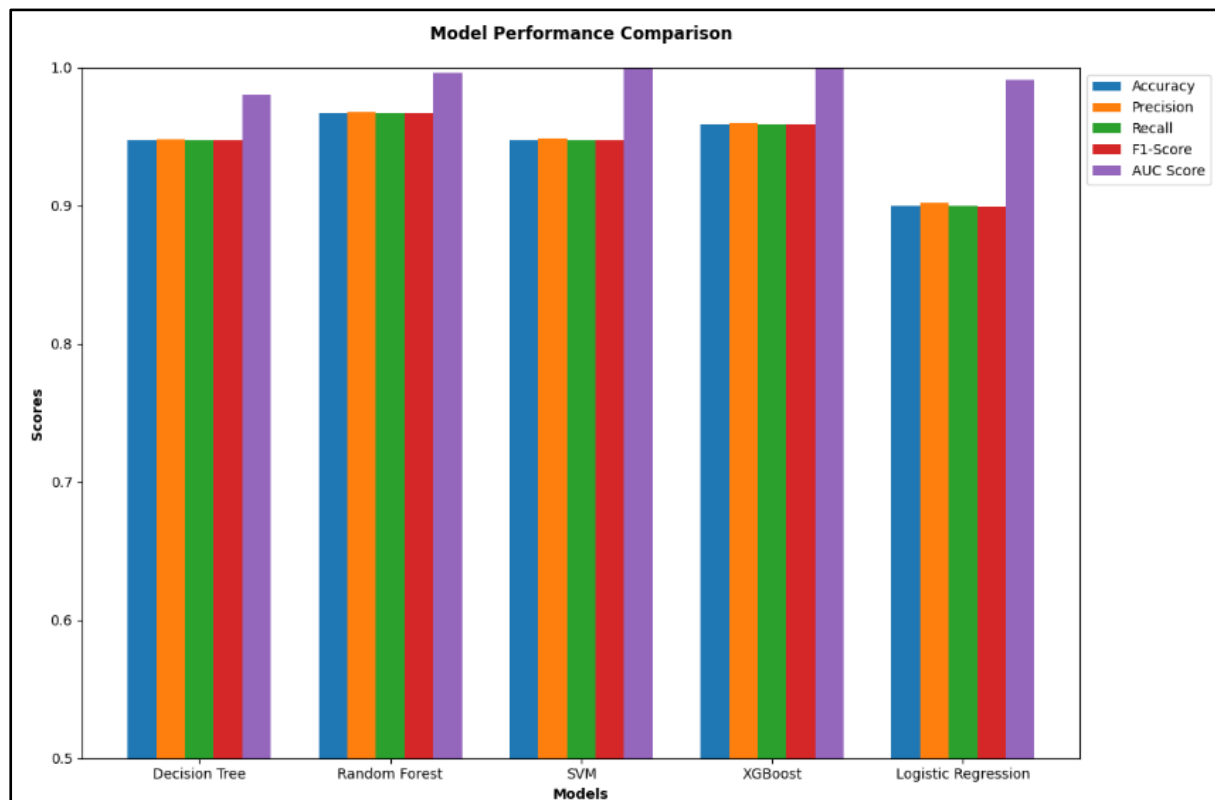


Figure 5.6: Overall Comparison of Models Performance Metrics

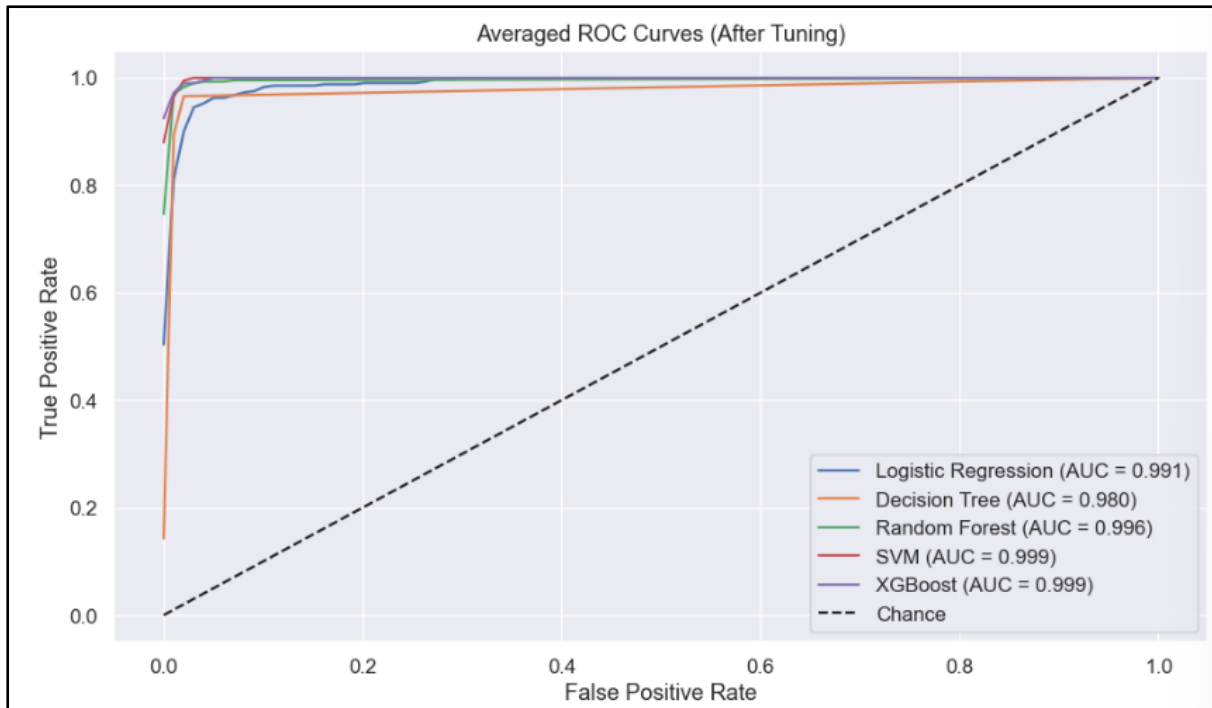


Figure 5.7: Averaged ROC Curves of All Models After Hyperparameter Tuning

5.4 Summary

In conclusion, this chapter evaluated the performance of various machine learning models by comparing their results using key evaluation metrics such as accuracy, precision, recall, f1-score, ROC-AUC value and confusion matrix. The comparison highlighted the strengths and weaknesses of each model, along with explanations for the best and worst performances observed. Among the proposed models, the pre-trained Random Forest model achieved the best results, demonstrating a strong ability to learn and interpret the dataset. Thus, the objectives of this project have been successfully fulfilled.

Chapter 6: Conclusion and Future Works

6.1 Introduction

Chapter 6 will include and highlight the objectives and achievements of this project. Also, the contributions, limitations and the possible future work of this project will be pointed out. This is to ensure that the further development and exploration on the model will be carried out for enhancement purposes.

6.2 Contributions

In this project, a publicly available dataset related to obesity levels was sourced from Kaggle and used to train multiple machine learning models for obesity prediction. A comparative study was conducted using five supervised machine learning algorithms which are Decision Tree, Random Forest, Support Vector Machine (SVM), Logistic Regression, and XGBoost. These models were evaluated based on key performance metrics including Accuracy, Precision, Recall, F1-score, and ROC-AUC to ensure a comprehensive assessment of their effectiveness in classifying obesity levels.

Among all the models tested, the Random Forest classifier yielded the highest overall performance, making it the most suitable model for deployment. This best-performing model was then integrated into a web-based application that allows users to input personal information and receive real-time predictions regarding their obesity level. The application was developed using Python and the Gradio library, with custom HTML and CSS styling to enhance usability and visual presentation. Finally, the web application was successfully deployed on Hugging Face Spaces, making it publicly accessible for demonstration and practical use.

6.3 Limitations

Despite the valuable insights gained from the study, several limitations should be considered. One of the limitations encountered in this study lies in the exclusion of complex contextual and biological factors such as socioeconomic status, stress levels, and sleep quality,

which are known to have a significant influence on obesity. The current model only considers a limited set of lifestyle and physical features, which may reduce its predictive accuracy for individuals with atypical conditions or borderline cases. Additionally, the scope of the study was focused on evaluating commonly used and interpretable machine learning algorithms, which resulted in the exclusion of more advanced techniques such as deep learning that may offer superior performance. While the model has been successfully deployed on Hugging Face Spaces, the current web application lacks robust security features, user authentication, and data privacy protocols, making it unsuitable for use in clinical or sensitive environments without further development. Moreover, the application does not yet include dynamic user features such as health tracking or personalized recommendations, which could greatly enhance user engagement and practical utility.

6.4 Future Works

Building upon the findings and limitations of this research, several avenues for future work are recommended. Firstly, the integration of additional input features such as mental health indicators, stress levels, and environmental factors may contribute to a more comprehensive and personalized assessment of obesity risk. These features would allow the model to capture a broader context that influences individual health outcomes. Secondly, although the current dataset is balanced, future studies may benefit from experimenting with more expressive machine learning models such as deep neural networks, which can identify intricate and non-linear interactions between variables. Furthermore, enhancing the interpretability of the model is essential, especially in health-related applications. In this regard, explainable AI (XAI) techniques such as SHAP (SHapley Additive exPlanations) or LIME (Local Interpretable Model-Agnostic Explanations) could be employed to provide clear, human-readable explanations of model predictions. These techniques offer insights into how each input feature contributes to the final outcome, thus improving transparency, trust, and

decision support. Moreover, the web application can be further developed by implementing secure user authentication, encrypted data processing, and a connected database system to support scalable storage and real-time analytics. Finally, future versions of the web application demo could include health tracking features such as monitoring weight changes, activity levels, and dietary habits and provide personalized advice or recommendations based on users' risk profiles. These enhancements would significantly improve user engagement, support proactive health management, and potentially lead to better long-term health outcomes.

6.5 Conclusion

Primitively, the three objectives that were established and defined in Chapter 1 have been successfully achieved. The first objective was to determine the best feature set for obesity prediction, as illustrated in Chapter 4, Figure 4.13, which presents a sorted summary table of the selected features. The second objective involved comparing the performance of various machine learning models for obesity prediction. Accordingly, Decision Tree, Random Forest, Support Vector Machine (SVM), XGBoost, and Logistic Regression models were discussed and evaluated to identify the best-performing approach. Lastly, Random Forest was found to achieve the best results compared to the other models after tuning. Consequently, the third objective was to determine the best machine learning model based on performance metrics was also accomplished.

References

- Alam, K., Chowdhury, D., & Haque, P. (2023). An Intelligent System for Obesity Detection through Human Activity Recognition and BMI Prediction. *2023 26th International Conference on Computer and Information Technology (ICCIT)*, 1–6. <https://doi.org/10.1109/ICCIT60459.2023.10441019>
- Al-Omar, H. A., Alshehri, A., Alqahtani, S. A., Alabdulkarim, H., Alrumaih, A., & Eldin, M. S. (2024). A systematic review of obesity burden in Saudi Arabia: Prevalence and associated co-morbidities. *Saudi Pharmaceutical Journal*, 102192. <https://doi.org/10.1016/j.jsps.2024.102192>
- Barzinji, A. O., Ma, C., Du, W., & Ma, J. (2021). A Machine Learning Approach to Predict the Trend of Obesity Prevalence at a Global Level. *2021 IEEE/ACIS 6th International Conference on Big Data, Cloud Computing, and Data Science (BCD)*, 25–30. <https://doi.org/10.1109/BCD51206.2021.9581579>
- Bellaver, E. H., Militão Da Costa, I., Redin, E. E., Moroni, L. S., & Kempka, A. P. (2024). The fermented milk can be a natural ally against obesity? Investigation of bovine milk fermentation by *Lactocaseibacillus casei* LBC 237, screening, and In silico predictions of bioactive peptides for obesity control. *Intelligent Pharmacy*, 2(4), 467–484. <https://doi.org/10.1016/j.ipha.2024.05.009>
- Bhuvana, R., Maheshwari, S., & Sasikala, S. (2023). Predict the Heart Disease Using a Logistic Regression Classifier Algorithm. *2023 12th International Conference on System Modeling & Advancement in Research Trends (SMART)*, 649–652. <https://doi.org/10.1109/SMART59791.2023.10428486>
- Celik, Y., Guney, S., & Dengiz, B. (2021). Obesity Level Estimation based on Machine Learning Methods and Artificial Neural Networks. *2021 44th International Conference on Telecommunications and Signal Processing (TSP)*, 329–332. <https://doi.org/10.1109/TSP52935.2021.9522628>
- Devi, K. N., Krishnamoorthy, N., Jayanthi, P., Karthi, S., Karthik, T., & Kiranbharath, K. (2022). Machine Learning Based Adult Obesity Prediction. *2022 International Conference on Computer Communication and Informatics (ICCCI)*, 1–5. <https://doi.org/10.1109/ICCCI54379.2022.9740995>
- Dwivedi, N., Singh, V., Gourisaria, M. K., Chatterjee, R., Bandyopadhyay, A., & Patra, S. S. (2024). Obesity Risk Detection Using Machine Learning Techniques. *2024 2nd International Conference on Sustainable Computing and Smart Systems (ICSCSS)*, 761–766. <https://doi.org/10.1109/ICSCSS60660.2024.10625526>
- Fang, M., & Huy, D. T. N. (2023). Building a cross-border e-commerce talent training platform based on logistic regression model. *The Journal of High Technology Management Research*, 34(2), 100473. <https://doi.org/10.1016/j.hitech.2023.100473>
- Ferdowsy, F., Rahi, K. S. A., Jabiullah, Md. I., & Habib, Md. T. (2021). A machine learning approach for obesity risk prediction. *Current Research in Behavioral Sciences*, 2, 100053. <https://doi.org/10.1016/j.crbeha.2021.100053>
- Fernandes, A., Dahikar, S., Chopra, K., & Saxena, K. (2023). Comparison of Machine Learning Algorithms for Obesity Prediction. *2023 3rd Asian Conference on Innovation in Technology (ASIANCON)*, 1–5. <https://doi.org/10.1109/ASIANCON58793.2023.10270246>
- Hayat Suhendar, M. T., & Widyani, Y. (2023). Machine Learning Application Development Guidelines Using CRISP-DM and Scrum Concept. *2023 IEEE International Conference on Data and Software Engineering (ICoDSE)*, 168–173. <https://doi.org/10.1109/ICoDSE59534.2023.10291438>
- He, Z. (2022). Comparison Of Different Machine Learning Methods Applied To Obesity Classification. *2022 International Conference on Machine Learning and Intelligent Systems Engineering (MLISE)*, 467–472. <https://doi.org/10.1109/MLISE57402.2022.00099>
- Islam, R., Sultana, A., & Tuhin, Md. N. (2024). A comparative analysis of machine learning algorithms with tree-structured parzen estimator for liver disease prediction. *Healthcare Analytics*, 6, 100358. <https://doi.org/10.1016/j.health.2024.100358>
- Kato, Y., Suzuki, T., Kobayashi, K., & Nakauchi, Y. (2012). A web application for an obesity prevention system based on individual lifestyle analysis. *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 1718–1723. <https://doi.org/10.1109/ICSMC.2012.6377985>

- Khater, T., Tawfik, H., & Singh, B. (2024). *Explainable artificial intelligence for investigating the effect of lifestyle factors on obesity*.
- Koklu, N., & Sulak, S. A. (2024). Using Artificial Intelligence Techniques for the Analysis of Obesity Status According to the Individuals' Social and Physical Activities. *Sinop Üniversitesi Fen Bilimleri Dergisi*, 9(1), 217–239. <https://doi.org/10.33484/sinopfb.1445215>
- Kolli, R., Razzaghi, T., Pierce, S., Edwards, R. K., Maxted, M., & Parikh, P. (2023). Predicting cesarean delivery among gravidas with morbid obesity—a machine learning approach. *AJOG Global Reports*, 3(4), 100276. <https://doi.org/10.1016/j.xagr.2023.100276>
- Krishna, R., & Hemamalini, S. (2024). Improved TLBO algorithm for optimal energy management in a hybrid microgrid with support vector machine-based forecasting of uncertain parameters. *Results in Engineering*, 24, 102992. <https://doi.org/10.1016/j.rineng.2024.102992>
- Kumar, A., R, R., & Kumar, K. (2024). Obesity Prediction Using Machine Learning Technique. *2024 2nd International Conference on Networking and Communications (ICNWC)*, 1–7. <https://doi.org/10.1109/ICNWC60771.2024.10537427>
- Kuriakose, S. M., Basa Pati, P., & Singh, T. (2022). Prediction of Diabetes Using Machine Learning: Analysis of 70,000 Clinical Database Patient Record. *2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 1–5. <https://doi.org/10.1109/ICCCNT54827.2022.9984264>
- Lam, B. C. C., Lim, A. Y. L., Chan, S. L., Yum, M. P. S., Koh, N. S. Y., & Finkelstein, E. A. (2023). The impact of obesity: A narrative review. *Singapore Medical Journal*, 64(3), 163–171. <https://doi.org/10.4103/singaporemedj.SMJ-2022-232>
- Lathifah, A., Arham, Z., Hasanati, N., Zulfiandri, & Nurmiati, E. (2023). Cross-Industry Standard Process For Data Mining (CRISP-DM) For Discovering Association Rules in Graduate Tracer Study Data of Islamic Higher Education Institution. *2023 11th International Conference on Cyber and IT Service Management (CITSM)*, 1–6. <https://doi.org/10.1109/CITSM60085.2023.10455691>
- Lee, S., & Chun, J. (2024). Identification of important features in overweight and obesity among Korean adolescents using machine learning. *Children and Youth Services Review*, 161, 107644. <https://doi.org/10.1016/j.childyouth.2024.107644>
- Li, S. (2023). Application of Random Forest Algorithm in New Media Network Operation Data Push. *2023 IEEE 15th International Conference on Computational Intelligence and Communication Networks (CICN)*, 87–92. <https://doi.org/10.1109/CICN59264.2023.10402335>
- Maataoui, S., Bencheikh, G., & Bencheikh, G. (2023). Predictive Maintenance in the Industrial Sector: A CRISP-DM Approach for Developing Accurate Machine Failure Prediction Models. *2023 Fifth International Conference on Advances in Computational Tools for Engineering Applications (ACTEA)*, 223–227. <https://doi.org/10.1109/ACTEA58025.2023.10193983>
- Manvitha, T., & Rekha, K. S. (2023). Improved Accuracy for prediction of leaf wetness using Logistic Regression algorithm compared with Decision Tree algorithm. *2023 Eighth International Conference on Science Technology Engineering and Mathematics (ICONSTEM)*, 1–5. <https://doi.org/10.1109/ICONSTEM56934.2023.10142550>
- Mora-Brito, F., & De Jesús Gil Herrera, R. (2023). Associating Obesity to Chronic Conditions through Machine Learning Techniques: A Mexican Case. *2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, 1–7. <https://doi.org/10.1109/ICECCME57830.2023.10252946>
- Nimmada, G., & Basha, S. M. (2024). A Novel Method for Predicting the Flight Price using Multilayer Perceptron Algorithm Compared with Support Vector Machine Algorithm. *2024 Ninth International Conference on Science Technology Engineering and Mathematics (ICONSTEM)*, 1–5. <https://doi.org/10.1109/ICONSTEM60960.2024.10568803>
- Nobel, S. M. N., Sultana, S., Singha, S. P., Chaki, S., Mahi, Md. J. N., Jan, T., Barros, A., & Whaiduzzaman, M. (2024). Unmasking Banking Fraud: Unleashing the Power of Machine Learning and Explainable AI (XAI) on Imbalanced Data. *Information*, 15(6), 298. <https://doi.org/10.3390/info15060298>
- Oldham, M., & Robinson, E. (2018). Visual body size norms and the under-detection of overweight and obesity. *Obesity Science & Practice*, 4(1), 29–40. <https://doi.org/10.1002/osp4.143>

- Pinto, K. A., Abdullah, N. L., & Keikhosrokiani, P. (2021). Diet & Exercise Classification using Machine Learning to Predict Obese Patient's Weight Loss. *2021 International Congress of Advanced Technology and Engineering (ICOTEN)*, 1–5. <https://doi.org/10.1109/ICOTEN52080.2021.9493560>
- Rasheed, S. S., & Glob, I. H. (2022). Classifying and Prediction for Patient Disease Using Machine Learning Algorithms. *2022 3rd Information Technology To Enhance E-Learning and Other Application (IT-ELA)*, 196–200. <https://doi.org/10.1109/IT-ELA57378.2022.10107935>
- Rodríguez, E., Rodríguez, E., & Nascimento, L. (2021). *Machine learning techniques to predict overweight or obesity*.
- Ruze, R., Liu, T., Zou, X., Song, J., Chen, Y., Xu, R., Yin, X., & Xu, Q. (2023). Obesity and type 2 diabetes mellitus: Connections in epidemiology, pathogenesis, and treatments. *Frontiers in Endocrinology*, 14, 1161521. <https://doi.org/10.3389/fendo.2023.1161521>
- Safaei, M., Sundararajan, E. A., Driss, M., Boulila, W., & Shapi'i, A. (2021). A systematic literature review on obesity: Understanding the causes & consequences of obesity and reviewing various machine learning approaches used to predict obesity. *Computers in Biology and Medicine*, 136, 104754. <https://doi.org/10.1016/j.compbiomed.2021.104754>
- Santisteban Quiroz, J. P. (2022). Estimation of obesity levels based on dietary habits and condition physical using computational intelligence. *Informatics in Medicine Unlocked*, 29, 100901. <https://doi.org/10.1016/j.imu.2022.100901>
- Shakti, M. A. S., Vijayalakshmi, M., Kumar, N., & Vaidhehi, M. (2024). Analysis on Various Machine Learning Framework for Obesity Level Prediction. *2024 1st International Conference on Cognitive, Green and Ubiquitous Computing (IC-CGU)*, 1–6. <https://doi.org/10.1109/IC-CGU58078.2024.10530812>
- Solano, J. A., Lancheros Cuesta, D. J., Umaña Ibáñez, S. F., & Coronado-Hernández, J. R. (2022). Predictive models assessment based on CRISP-DM methodology for students performance in Colombia—Saber 11 Test. *Procedia Computer Science*, 198, 512–517. <https://doi.org/10.1016/j.procs.2021.12.278>
- Sudhakar, K., Reddy Mallireddy, S. M., Tallapureddy, P., Vuddanti, S., Katragadda, B., & Manda, S. R. (2023). Water Level Prediction using Random Forest Algorithm. *2023 7th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 1586–1589. <https://doi.org/10.1109/ICECA58529.2023.10394849>
- Tee, E. S., & Voon, S. H. (2024). Combating obesity in Southeast Asia countries: Current status and the way forward. *Global Health Journal*, 8(3), 147–151. <https://doi.org/10.1016/j.glohj.2024.08.006>
- Wang, F., Liu, R., Yan, H., Liu, D., Han, L., & Yuan, S. (2024). Ground visibility prediction using tree-based and random-forest machine learning algorithm: Comparative study based on atmospheric pollution and atmospheric boundary layer data. *Atmospheric Pollution Research*, 15(11), 102270. <https://doi.org/10.1016/j.apr.2024.102270>
- Wang, Y., Pan, Z., Zheng, J., Qian, L., & Li, M. (2019). A hybrid ensemble method for pulsar candidate classification. *Astrophysics and Space Science*, 364(8), 139. <https://doi.org/10.1007/s10509-019-3602-4>
- Yu, X., Hu, T., Khodadadi, N., Liu, J., & Nanni, A. (2024). Predictive and experimental assessment of chloride ion permeation in concrete subjected to multi-factorial conditions using the XGBoost algorithm. *Journal of Building Engineering*, 98, 111041. <https://doi.org/10.1016/j.jobe.2024.111041>

APPENDICES

Appendix A: Turnitin Report

FYP2_Final Report_81505_Turnitin Check			
ORIGINALITY REPORT			
25%	20%	18%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS
PRIMARY SOURCES			
1	ir.unimas.my Internet Source	1%	
2	www.mdpi.com Internet Source	1%	
3	www.coursehero.com Internet Source	1%	
4	www.frontiersin.org Internet Source	1%	
5	ph01.tci-thaijo.org Internet Source	<1%	
6	Submitted to Liverpool John Moores University Student Paper	<1%	
7	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1%	
8	arxiv.org Internet Source	<1%	
9	Submitted to University of Ulster Student Paper	<1%	
10	dspace.unza.zm Internet Source	<1%	
11	Poonam Nandal, Mamta Dahiya, Meeta Singh, Arvind Dagur, Brijesh Kumar. "Progressive Computational Intelligence, Information Technology and Networking", CRC Press, 2025	<1%	