A HYBRID MACHINE LEARNING MODEL FOR ESTIMATION OF OBESITY LEVELS

By

Akash Choudhuri

(Roll Number: E1773U194002, University Registration Number: 239/ 2019)

A THESIS

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

In Mathematics with Data Science

INSTITUTE OF MATHEMATICS AND APPLICATIONS, BHUBANESWAR

2021

This thesis has been approved in fulfillment of the requirements for the Degree of MASTER OF SCIENCE in Mathematics with Data Science (Capstone Project).

Institute of Mathematics and Applications, Bhubaneswar

Thesis Advisor: *Prof. Sudarsan Padhy*

Director: *Prof. Jasobanta Jena*

Certificate

This is to certify that the work contained in the thesis entitled “A Hybrid Machine Learning Model for Estimation of Obesity Levels”, submitted by Akash Choudhuri (Roll No: 2019D014) for the award of the degree of Master of Science in Mathematics with Data Science to the Institute of Mathematics and Applications, Bhubaneswar, is a record of bonafide research works carried out by him under my direct supervision and guidance.

I considered that the thesis has reached the standards and fulfilling the requirements of the rules and regulations relating to the nature of the degree. The contents embodied in the thesis have not been submitted for the award of any other degree or diploma in this or any other university.

Prof. Sudarsan Padhy.
Supervisor.

Date: 2nd September, 2021.

Declaration

I certify that:

a. The work contained in the thesis is original and has been done by myself under the supervision of my supervisor.
b. The work has not been submitted to any other Institute for any degree or diploma.
c. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.
d. Whenever I have used materials (data, theoretical analysis, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.
e. Whenever I have quoted written materials from other sources and due credit is given to the sources by citing them.

Akash Choudhuri

Akash Choudhuri.
Roll: 2019D014.

Date: 2nd September, 2021.
Place: Kolkata, West Bengal, India.

Table of Contents

List of Figures

List of Tables

Dedicated to my grandfather, late Shri Bimal Ratan Roy.

I know you would have been proud to see this.

This Page is intentionally left blank

This Page is intentionally left blank

Acknowledgements

The present master thesis is the document to support my candidature for the academic degree Master of Science (M.Sc.), in which is detailed explanation of my research and findings on the topic *A Hybrid Machine Learning Model for estimation of Obesity Levels*. Mentioned study was developed in Institute of Mathematics and Applications, Bhubaneswar at Utkal University, India.

I would like to thank Prof. Sudarsan Padhy, who has supervised the academic and professional development of investigation from two different points of views. First, the implementation of the Machine Learning Algorithms and second, the understanding of obesity. Also, I kindly appreciate my family and personal friends; especially to Tanuka Choudhury and Himangshujyoti Choudhuri for their enthusiastic support, constant motivation, and constant encouragement, and to Pitambar Muduli for the constructive comments that improved the presentation of these findings.

Definitions

Obesity: Abnormal or excessive fat accumulation that presents a risk to health. A body mass index (BMI) over 25 is considered overweight, and over 30 is obese.

Github: It is a web-based version-control and collaboration platform for software developers. It is owned by Microsoft.

WEKA: Waikato Environment for Knowledge Analysis (Weka), developed at the University of Waikato, New Zealand, is free software licensed under the GNU General Public License and contains a collection of visualization tools and algorithms for data analysis and predictive modeling, together with graphical user interfaces for easy access to these functions

Kaggle: It is a subsidiary of Google LLC, is an online community of data scientists and machine learning practitioners.

J48:J48 algorithm is a Decision Tree Algorithm and is one of the best machine learning algorithms to examine the data categorically and continuously. When it is used for instance purpose, it occupies more memory space and depletes the performance and accuracy in classifying medical data.

List of Abbreviations

KDD: Knowledge Discovery in Databases.

MLP: Multilayer Perceptron.

ML: Machine Learning.

RFE: Recursive Feature Elimination.

BMI: Body Mass Index.

SVM: Support Vector Machine.

DT: Decision Tree.

PWO: People with Obesity.

Abstract

Obesity has always been a problem which has plagued humans for many generations, which, since the 1975, almost doubled to turn into a global epidemic. The current human dependence on technology has contributed to the problem even more, with the effects visibly pronounced in late teenagers and early adults. Researchers till date, have tried numerous ways to determine the factors that cause obesity in early adults.

On that frontier, our hybrid machine-learning model uses the help of some supervised and unsupervised data mining methods like Extremely Randomized Trees, Multilayer Perceptron and XGBoost using Python to detect and predict obesity levels and help healthcare professionals to combat this phenomenon. Our dataset is a publicly available dataset in the UCI Machine Learning Repository, containing the data for the estimation of obesity levels in individuals from the countries of Mexico, Peru, and Colombia, based on their eating habits and physical condition. 77% of the data was generated synthetically using the Weka tool and the SMOTE filter, 23% of the data was collected directly from users through a web platform.

Keywords: Obesity, Extremely Randomized Trees, Multilayer Perceptron, XGBoost.

1 Introduction

“I’m fat but I’m thin inside…. There’s a thin man inside every fat man”

-George Orwell

The concept of obesity is a fairly relevant topic in the modern society. With the advent of modern technology, mankind is trying its best to eradicate the consequences of unhealthy lifestyle, which has led to be the major contributor towards obesity. However, researchers are yet to discover and develop foolproof methods to try to detect obesity at the very onset in order to combat the phenomenon. With the current developments in the domain of Machine Learning, one can construct a contemporary investigative model to design an accurate and precise programming routine at low computational cost, which would optimally involve a quality compromise among the typical method features.

The present chapter consists of a comprehensive and summarized description about the motivation behind this study and aptly includes the background information with the most relevant scientific articles related to the statistical and machine learning approaches previously used to study the phenomenon. Furthermore, it introduces the restrictions of the past investigations as challenges ahead, following by a proposal to overcome them as current master’s thesis goals.

1.1 History of Obesity

The problem of obesity is a prevalent problem that has grown exponentially in sync with the advancement of modern technological tools. With its origins tracing back up to 30000 years [1],even before the origin of some of the Ancient Civilizations, obesity has extensive mentions in early literature with Hippocrates mentioning how obesity led to infertility and early death, by mentioning, “In the beginning, man made use of the same food with the beasts, and it was the many distempers brought upon him by such indigestible aliment, which taught him, in length of time, to find out a different diet, better adapted to his constitution, teaching that ... the spontaneous and crude productions of the earth must have shortened rather than lengthened their lives” [2].

Regarding poor diet and the importance of having a healthy meal, he further comments, “The distempers arising from the coarse aliment which men at first made use of, obliged them to study the most proper methods of preparing bread from grain, and of dressing other vegetables as should render them more wholesome… One cause which made it necessary to study the art of restoring lost health, was the great difference to be observed between the diet of the healthy and that of the sick” [2].

Over time, obesity had various implications in the human society. While in some civilizations, obesity was considered to be a valuable quality to have, indicating wealth and reach, most other civilizations tried to understand the dangers obesity brought with it in the long term. However, the degree of acceptance of obesity in the human society has been comparatively slow over other major diseases. The Ancient Greeks and Egyptians seemed to despise the phenomenon. The Egyptians, in particular seemed to consider diet primarily as a source of preservation of health. Their method of limiting food intake was primitive. According to Diodorus Siculus, they had to “prevent distempers by glisters, purging, vomiting or fasting every second, third or fourth day”, as “the greatest part of the aliment we take is superfluous, which superfluity is cause of our distempers” [4]. According to Herodotus, “Egyptians vomit and purge themselves thrice every month, with a view to preserve their health, which in their opinion is chiefly injured by their aliment” [4].

The major scientific approach was initially applied by Pythagoras, which stated “No man, who values his health, ought to trespass on the bounds of moderation, either in labor, diet or concubinage” [5]. In fact, Hippocrates later developed his subsequent theorems of trying to preserve health by his Energy Balance Equation whose excerpts were “It is very injurious to health to take in more food than the constitution will bear, when, at the same time one uses no exercise to carry off this excess” [6]…. “For as aliment fills, and exercise empties the body, the result of an exact equipoise between them must be to leave the body in the same state they found it, that is, in perfect health” [2].

Over time, medical research seemed to align towards a more modern method to tackle obesity. This was also due to increase in the number of obese people in the society. With philosophers and writers in the 18th Century realizing preservation of health is a major reason to avoid obesity. George Cheyne MD was one of the pioneers who recognized obesity as a disease. He also stated that obesity was linked with depression [7]. Later, Joannes Baptista Morgagni reasserted Cheyne’s claim and also proved, by dissection, that fat was crucially positioned to determine obesity. In his Epistola anatoma clinica XXI, he describes a female with severe obesity and 2 spect 2 spect– manly, or virile aspect. The abdomen was prominent containing a large amount of fat accumulated in the intra-abdominal spaces and at the mediastinal level, with a raised diaphragm [8]. Later William Banting notably illustrated that obesity caused gout [9] and William Buchan argued that obesity affected female health and fertility [10].

One of the chief comorbidities of obesity is diabetes, which was first illustrated by Ebers Papyrus as a condition which caused ‘excessive urination’ [11]. Over time, obesity was also later linked to a multitude of conditions including coronary heart diseases and abdominal fat.

At the current time, studies in 2013 have shown that in USA, obesity prevalence based on self-reported BRFSS data was 29%, in contrast to 34% using objectively-measured height and weight data from the National Health and Nutrition Examination Survey (NHANES) [12].

Thus, it is clearly visible that human knowledge about obesity has developed over time. However, the current challenge lies in trying to properly determine the extent to which external factors affect this disease, which would enable healthcare professionals to model appropriate measures to eradicate obesity.

1.2 Motivation

As clearly illustrated in the previous section, mankind extensively wants to combat the problem of obesity. According to Eduardo De-La-Hoz-Correa et al., Obesity has become a global epidemic that has doubled since 1980, with serious consequences for health in children, teenagers and adults [13]. In fact, the number of people that suffers from obesity has doubled since 1980 and also in 2014 more than 1900million adults, 18 years old or older, are suffering from alteration of their weight. This has chiefly been the result of various factors, some of which are rapid urbanization, increase in intake of fast foods, decrease in physical activity and updated modes of transport. In fact, smoking habits have also been major contributors towards obesity.

Obesity has serious consequences on our cardiovascular systems. However, the condition seems to develop in the intersection of late childhood and early adulthood. On the segment of obesity in children Cecil et al. showed that metabolic disorders which lead to obesity are equally pronounced in children as that in case of adults [14]. However, very little research has been performed on the samples of data concerning early adults. This could probably be due to the lack of resources for data collection in this segment. However, the current Covid-19 Pandemic has affected the group of late teenagers and early adults the most, with Undergraduate and Graduate University students being highly susceptible to obesity due to online education and lack of avenues of physical activity in Covid-19 restrictions in public places [16]. With prior evidence that obesity causes depression and increased symptoms of depression being found in younger subjects, women, and those with poorer body image [15], it is extremely important to conduct a comprehensive study with health data of young adults.

On that note, the motivation of this work has germinated from an effort to inculcate a sense of awareness about obesity amongst the young adults while also trying to outline a method for the usage of hybrid Machine Learning Models for healthcare. In fact, the chief motivation of this work is to try to devise a fool-proof method to detect obesity in young adults and also assert some inherent assumptions about the target symptom to look out for in further cases of obesity in future.

1.3 Related Works

Previously, most of the related works in this domain have focussed on trying to determine the factors that caused obesity. However, of late, with the advent and popularity of Machine Learning Algorithms and software, the process of trying to estimate obesity level has also become a major topic of research.

Davila-Payan et al. [17] provided a logistic regression model for estimating the likelihood of mass body index in children aged 2 to 17 years in small geographic areas. This essentially showed that minute geographical regions essentially generated more information to try to solve the problem. A computational model using fuzzy architecture is presented by Manna and Jewkes [18] to understand and manage intricacies on the data of children obesity and a solution that could handle the risk associated with early obesity and children motor development. The work used fuzz signatures to handle external factors like imprecision and uncertainty.

Adnan and Husain developed a model based on the Naïve Bayes' Model [19] with a hybrid approach using genetic algorithms to optimise factors used in the prediction of juvenile obesity, with a low percentage of negative samples vs positive samples. They obtained 19 parameters to be implemented in prediction with a precision of 75%. This was practically the inspiration from their earlier work [20], where they collected information from primary sources and identified risk factors like obesity and level of education of the parents, lifestyle and habits of the children and influence of the environment. The proposed framework uses a hybrid technique of Naïve Bayes and decision trees called NBTree. This, by all means was also an inspiration from a previous work of Adnan et al. [21] used data mining to predict children obesity. The purpose of the proposed survey was to provide the necessary knowledge for the obesity problem, introduce data mining for prediction, describe the current efforts in that area and show the benefits and weaknesses of each technique used. The techniques involved were Neural Networks, Naïve Bayes and Decision Trees.

Dugan et al. generated a predictive study [22] of children obesity with subjects older than 2 years old, using exclusively the data previous to their second birthday using a decision-making system called CHICA. The methods analysed included RandomTree, RandomForest, J48, ID3, Naïve Bayes and Bayes. Their results showed that ID3 had better behaviour with 85% precision and 89% sensibility.

Zhang et al. [23] compared logistic regression with six data mining strategies for predicting childhood overweight and obesity in 3-year-old participants, using data at birth, six weeks, eight months, and two years old. The authors observed an increase in prediction precision of more than 10% in cases of 8 months and 2 years old. The techniques used were Decision Trees, Association Rules, Neural Networks, Naïve Bayes, Bayesian Networks and Support Vector Machines. Suguna [24] offered a methodology for analysing obesity in children aged 10 to 17 years old using the Child and Adolescent

Health Measurement Initiative (CAHMI) dataset. The proposed model uses Decision Trees with three different algorithms: Simple Cart, J47 and NB Tree.

Abdullah et al. [25] showed a children obesity classification in grade school 6, from two different Malaysia districts. From the information collected, the authors created 4245 full datasets and applied Bayesian Networks, Decision Trees, Neural Networks and Support Vector Machines (SVM). They proposed a review to show Artificial Intelligent applications to obesity management and discussed their effectiveness. They performed the research in the following databases: Public Medline (PubMed), Web of Science, Biblioteca Regional de Medicina (BIREME), and Google Academic, by using the following keywords, “artificial intelligence” and “obesity”. The results led to some Artificial Intelligence systems used in obesity handling, which were: the Decision Support System to bariatric surgery patients; the MOPET app to motivate physical activity; Parameter Decreasing Methods and Artificial Neural Network to correlate obesity to cardiovascular disease; Artificial Neural Network to predict resting energy expenditure; a Neuro-Fuzzy Model to refine body mass index result; an Image Processing Algorithm; and a Support Vector Machine that monitors food intake.

In [26] Correa Eduardo et al. presented data for the estimation of obesity levels in individuals from the countries of Mexico, Peru, and Colombia, based on their eating habits and physical condition. The data contains 17attributes and 2111 records; they labelled the records with the class variable Nobesity (Obesity Level) that allows classification of the data using the values of Insufficient Weight, Normal Weight, Overweight Level I, Overweight Level II, Obesity Type I, Obesity Type II and Obesity Type III. They generated 77% of the data synthetically using the Weka tool and the SMOTE filter, and they collected 23% of the data directly from users through a web portal. This data can be used to generate intelligent computational tools to identify the obesity level of an individual and to build recommender systems that monitor obesity levels. We have used the same data and thus used their results as baseline.

Ward et al. [27] presented accurate methods to measure food and energy intake are crucial for the battle against obesity. Providing users/patients with convenient and intelligent solutions that help them measure their food intake and collect dietary information are the most valuable insights toward long-term prevention and successful treatment programs. The authors proposed an assistive calorie measurement system to help patients and doctors succeed in their fight against diet-related health conditions. The system runs on smartphones, which allows the user to take a picture of the food and measure the amount of calorie intake automatically. To identify the food accurately in the system, they use deep convolutional neural networks to classify 10,000high-resolution food images for system training. The results show that the accuracy of the method for food recognition of single food portions is99%.

Gomez and Avila [28] proposed a new dataset for the evaluation of food recognition algorithms that can be used in dietary surveillance applications. Each image represents a real dining tray with dishes and foods arranged in different ways. Each tray contains multiple instances of food types. The dataset contains 1027 dining trays for a total of

3616 instances of food belonging to 73 food types. The food on the tray images has been manually segmented using carefully drawn polygonal boundaries. They have made a comparative evaluation of the dataset by designing an automatic tray analysis pipeline that takes an image of the tray as input, finds the regions of interest, and predicts for each region the corresponding food type. They have experimented with three different classification strategies using also various visual descriptors. They achieved about 79% accuracy in food and tray recognition using features based on convolutional neural networks.

Joachims in [29] mentioned that recent large-scale genome-wide association studies have identified tens of genetic loci robustly associated with Body Mass Index. They also found their associated gene expression profiles with BMI. However, accurate prediction of obesity risk utilising genetic data remains challenging. In a cohort of 75 individuals, the authors integrated 27 BMI-associated SNPs and obesity-associated gene expression profiles. They computed the genetic risk score by adding BMI-increasing alleles. They correlated the genetic risk score with BMI when they used an optimization algorithm that excluded some SNPs. They built linear regression and support vector machine models to predict obesity risk using gene expression profiles and the genetic risk score. They achieved an adjusted R2 value of 0.556 and an accuracy of 76% for the linear regression and support vector machine models, respectively. The authors report a new mathematical method to predict obesity genetic risk. They constructed obesity prediction models based on genetic information for a small cohort. The computational framework serves as an example of using genetic information to predict obesity risk for specific cohorts.

Cervantes et al. [30] proposed and compared a SVM Model with a Decision Tree model using the obesity dataset generated from the University students of Latin American countries and achieved high precision and recall values using the decision tree. Further, they proposed a Decision Tree + Simple K-Means model, where they achieved 98.5% precision and 98.5% recall. We have used this model as a baseline model to evaluate our results.

1.4 Purpose and Research Question

In this thesis, a hybrid machine learning technique will be used to estimate the level of obesity in a dataset containing lifestyle habits of young undergraduate students. This thesis will also try to explore the relationship between the various lifestyle habits to obesity. In addition to that, we will try to perform a scenario testing on the dataset with respect to Covid-19. Concerning this issue, we will try to answer the following implications:

- Is our model robust? How well does it perform with respect to change in trends?
- How well does our model generalize and detect obesity?

1.5 Approach and Methodology

This work focuses on using machine learning methods and algorithms in order to classify obesity levels in young adults. There are four different problems that will be solved in this thesis.

Firstly, as the size of the original data was very small, synthetic data was generated using data mining techniques. So, the first task is to validate the characteristics of the synthetic data with respect to the original data. Any outliers present in the synthetic data needs to be duly removed.

Secondly, character type entries need to be encoded into a numeric vector space so that it is fit into the machine learning model. There are numerous ways to implement this. The choice of technique totally depends on the overall performance. In addition to this, all the features thus obtained need to be scaled appropriately. Likewise, there are various feature scaling techniques available. However, the choice of the appropriate scaling method totally depends on the problem statement and dataset.

Thirdly, it is extremely important to devise an appropriate feature selection methodology to weed out the less important features. This is an extremely important step as it reduces the subsequent computations, which lead to increase in speed without compromising too much on the overall performance. In some cases, it also leads to improved performance of predictions.

Fourthly, a hybrid machine learning algorithm will be created and then optimized by training it on a subset of the data. The model will then predict for new instances and performance will then be evaluated.

The planned procedure for our master thesis is the following: Based on research on existing methods and metrics, a hybrid machine learning model will be constructed to answer the given research questions. Then the created model’s performance will be compared to that of the previous models created on similar studies. Furthermore, the model will also be validated on a given scenario and would then be reviewed and evaluated. Then, the possible limitations would be pointed out and methods would be suggested to overcome them.

1.6 Scope and Limitation

Due to the fixed time frame, some limitations have to be set on this research to ensure that the work can be finished in time. They are as follows:

- The baseline assumption is that the data collected is correct and there has been no error while taking it manually.
- As the given dataset describes the obesity trend of people of a particular age group belonging from a particular region and as not generalized dataset was found, the given model might not work well with other trends of data collected from a different region in the world or from a different age group. However, the basic idea is to serve as a source of inspiration in using hybrid machine learning models for obesity detection.
- The work can be greatly improved if more data is collected, which would prevent the creation of synthetic datasets, which can reduce bias in the model.

1.7 Target Group

Firstly, this work aims to grab the interest of healthcare professionals to utilize machine learning algorithms to automate their tasks. As mentioned in section 1.2, it is also aimed towards the society in general, especially the young adults. This work aims to build a level of awareness to prevent the consumption of fast foods and maintaining unhealthy habits.

Finally, this work also aims to draw the attention of the researchers working in the areas of machine learning and applications of statistical and machine learning models in healthcare.

1.8 Outline

The following section (Section 2) will describe and explore the theoretical background concerning this work, which will focus on supervised learning and classification problems as a whole, the machine learning techniques used in this work and performance metrics. After describing dataset and the taken methodology in Section 3, the respective results will be shown. Section 4 will discuss the results, critically review the taken approaches and methods and compare our results with those published by previous research on this topic as well as examine the validity and reliability of the presented results. Finally, Section 5 will summarize the work and give an outlook for possible future research and expansion on this topic. The Appendix will contain the code snippets.

2 Theoretical Background

This section gives a detailed overview of the various theoretical foundations that have been used in this work. With respect to the experiments conducted, it answers the following questions:

- What is Supervised Learning and what is Classification?
- What are the advantages of using Machine Learning Algorithms and which algorithms are widely used in the field of obesity detection?
- How do the algorithms we have used work and what are hybrid machine learning models?
- What metrics are commonly used to measure multi-class classification problems?

To answer these questions, the first section deals with knowledge discovery and statistical learning including Supervised and Unsupervised Learning and Classification problems followed by relevant algorithms. Then there will be a brief elaboration on the feature selection, feature extraction and normalization techniques. Finally, this section ends by elaborating the common performance metrics for multi-class classification problems.

2.1 Knowledge Discovery

Knowledge discovery broadly refers to the overall process of extracting useful information from data. According to Frawley et al. [32], Knowledge discovery is the nontrivial extraction of implicit, previously unknown, and potentially useful information from data. Given a set of facts (data) F, a language L and some measure of certainty C, we define pattern as a statement S in L that describes relationships among a subset FS of F with certainty c, such that S is simpler (in some sense) than the enumeration of all facts in F_s. A pattern that is interesting (according to a user-imposed interest measure) and certain enough (again according to the user’s criteria) is called knowledge. The output of a program that monitors the set of facts in a database and produces patterns in this sense is discovered knowledge.

Fayyad et al. [33] outlines nine major steps to assert their claims that Knowledge Discovery in Databases (KDD) is an iterative process. They are, as follows:

1. Develop an understanding of the application domain to obtain prior knowledge and identifying the goal of the KDD process.
2. Create a target dataset on which Knowledge Discovery will be performed.
3. Data cleaning and pre-processing to remove noise, handling missing values and extracting relevant information.

4. Data reduction and projection to find useful information to represent data by performing dimensionality reduction or transformation by keeping the essence of the original data intact.
5. Matching the goals of the KDD process (defined in step 1) to a particular data mining method.
6. Exploratory analysis and model hypothesis selection to decide the appropriate models and parameters and matching a data mining method to the overall criteria of the KDD process.
7. Search pattern(s) of interest in a particular representational form or a set of representations.
8. Interpret mining patterns, possibly returning to any of steps between 1 through 7 for further iteration.
9. Act on the discovered knowledge by either incorporating the knowledge directly or create a documentation and present it to the interested parties.

By all means, this is an iterative process and one might need to constantly travel between the above steps to obtain the optimal policy.

2.1.1 Statistical Learning: An overview

Statistical learning essentially refers to the vast set of tools and techniques associated with understanding data [34]. The process of Knowledge discovery extensively uses statistical learning in order to build the appropriate methodologies to generate insights about a given dataset. Although the field of statistical learning is practically new, many of its underlying theories and concepts have been in research for a long time.

This started at the beginning of the 19th Century, when the method of least squares was developed, which led to the formulation and implementation of the current popularly known linear regression algorithm. Linear regression had broad applications in the field of astronomy and quantitative analysis. On that note, the concept of linear discriminant analysis was discovered in 1936 for stock price predictions followed by logistic regression in 1940s to handle with classification problems. The concept of generalized linear models was discovered in late 1970s, followed by the foundations of classification and regression trees and later generalised additive models in 1980s. The 1980s were also characterised by the popularity of neural network models followed by the popularity of support vector machines in the 1990s.

This progress of statistical learning in the 20th Century has heralded its emergence as a new field of applied statistics, focussing on the various supervised and unsupervised learning methods. In the recent years, with the availability of more data and programming languages and relevant libraries, statistical learning has surely become one of the widest research techniques.

Broadly, statistical learning is divided into two broad subcategories. They are:

- **Supervised Learning:** This method of statistical learning broadly includes the creation of a relevant statistical model to estimate or predict an output (also called target) based on a given set of inputs (also called features). In simpler terms, supervised learning involves processes where some data is already known and our job is to assign new data to correct classes.
- **Unsupervised Learning:** This method of statistical learning is very similar to supervised learning, where a given set of inputs (or features) is provided, but no supervising output (or target) is provided. This kind of learning chiefly focuses on trying to explore patterns (relationships and structures) in the data, which are not visible otherwise.

Generally, variables can either be classified as qualitative or quantitative. Quantitative variables take on numerical values whereas qualitative variables take values in one of the different classes that are available. On that note, supervised learning is also further divided into two sub-sections. They are, as follows:

- **Classification:** Supervised learning problems where the output/ target/response variable is a qualitative variable (that is, the variables take on values in one of K different classes). Some examples of target variables in classification problems include a person's marital status (married or not), or whether a person is a loan defaulter (yes or no) or the brand of product purchased (Brand A, Brand B, or Brand C), etc.
- **Regression:** Supervised learning problems where the output/ target/response variable is a quantitative variable (that is, the variable takes on numerical values). Some examples of target variables in regression problems include age of a person, the monetary value of a property, the price of a stock, etc.

Figure 1 gives a clear overview of the above differentiation.

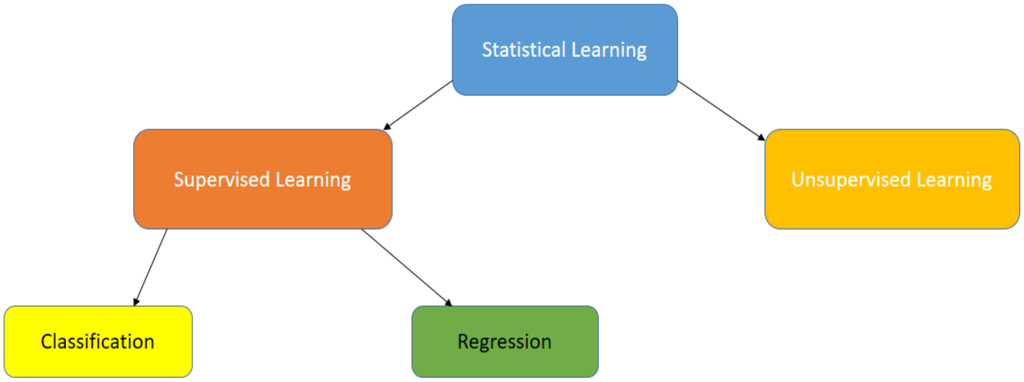


Figure 1:The Different classifications of Statistical Learning

In our given problem, as we will be performing multi-class classification in trying to predict the state of obesity a particular person is in, this work will focus on supervised learning methods.

2.2 Machine Learning

In the context of modelling, it is extremely important to choose the appropriate approach to tackle the task effectively. This is done by Machine Learning methods. A major difference between humans and computers has been for a long time that a human beings tend to improve their way of tackling a problem and learn from previous mistakes to try and solve them by looking for new approaches to address the problem. Traditional computer programs do not look at the outcome of the tasks and are therefore unable to improve their behaviour. The field of machine learning addresses this exact problem and involves the creation of computer programs that are able to learn and therefore improve their performances by gathering more data and experience.

In general, machine learning systems are normally classified by their underlying learning strategies, which are often identified by the amount of inference the computer program is able to perform [35]:

- **Rote learning** uses direct information to be given to the learner. No inference or transformation of knowledge is required and this learning requires no effort from the learner.
- **Learning from instruction** consists of getting information from some sort of a knowledgeable organized source. The learner transforms the input language into internal information representation and periodically updates it. Thus, the learner’s knowledge increases with more availability of information sources.
- **Learning by analogy** is based on the concept of trying to acquire new knowledge by transforming existing knowledge. It requires more inference on the part of the learner that does rote learning or learning from instruction.
- **Learning from examples** is the most generalized and popular concept that is used in machine learning. Given a set of examples of a concept, the learner induces a general concept description that describe the examples. The amount of inference performed by the learner is much greater than in learning from instruction and in learning by analogy. Learning from examples has become so popular in the last years that it is often called simply learning. In a similar way, the learner and examples are respectively referred as learning machine and data.

On that note, the machine learning algorithms used in this work would be discussed briefly.

2.2.1 Multilayer Perceptron (MLP)

By its architecture, the multilayer perceptron (MLP) model (popularly known as artificial neural networks), is essentially an updated variant of the Rosenblatt Perceptron Model [36]. Compared to regular machine learning models, multilayer perceptron model is highly useful for solving complex problems, especially in cases where abundant data is available. The multilayer perceptron model architecture is highly inspired from the structure of the human brain. This enables the model to solve highly complicated and non-linear types of problems, which the Rosenblatt Model [36] could not do.

A multilayer perceptron model consists of fundamental structural parts called neurons (also called nodes). The weighted connections between the neurons can be adapted during the learning process of the network and the activation function determines the output of each neuron given a set of inputs. The neurons arrange themselves to form layers and a typical multilayer perceptron architecture consists of an input layer which accepts the given feature information, few hidden layers, and an output layer which gives the response of the model. A noticeable feature of the MLP model is that each neuron in a particular layer would be connected to another neuron in the next layer.

The basic features of the multilayer perceptron model according to [37] are:

- Each neuron in the model contains a nonlinear and differentiable activation function.
- The model contains one or more hidden layer(s).
- The network exhibits a high degree of connectivity, the extent of which is determined by synaptic weights of the network.

The differentiation of the multilayer perceptron model from its simple variant differs in its method of training by the *back-propagation algorithm* [38]. The training of a multilayer perceptron model proceeds in two phases:

- **The Forward Phase:** In this phase, the weights are fixed and the input signal is passed through the network, layer by layer, until it reaches to the output layer. This means that the changes are confined by the activation thresholds and outputs of neurons in the network.
- **The Backward Phase:** In this phase, error is calculated by comparing the output of the output layer and the desired response. The resultant error is backward propagated through the network, layer by layer. Then, significant adjustments are made to the weights of the network by computing the gradient.

A MLP model is generally fully connected, which means that a neuron in any layer of the network is connected to all the neurons (nodes) in the previous layer. Signal flow through the network progresses in a forward direction, from left to right and on a layer-by-layer basis. Figure 2 shows a fully connected Multilayer Perceptron model.

As previously discussed, activation functions are extremely important in this model. Some popular activation functions used are:

- **Logistic function:** This function in its general form is defined as:

  $$\varphi_j(v_j(n)) = \frac{1}{1 + \exp(-av_j(n))}, \qquad a > 0$$

  Where $\upsilon_j(n)$ is the induced local field of neuron j and a is an adjustable positive parameter.
- **Hyperbolic Tangent Function:** This is another commonly used form of sigmoidal non linearity. In its most generalised form, the hyperbolic tangent function is defined by:

  $$\varphi_j(v_j(n)) = a \tanh(bv_j(n))$$

  Where a and b are positive constants.

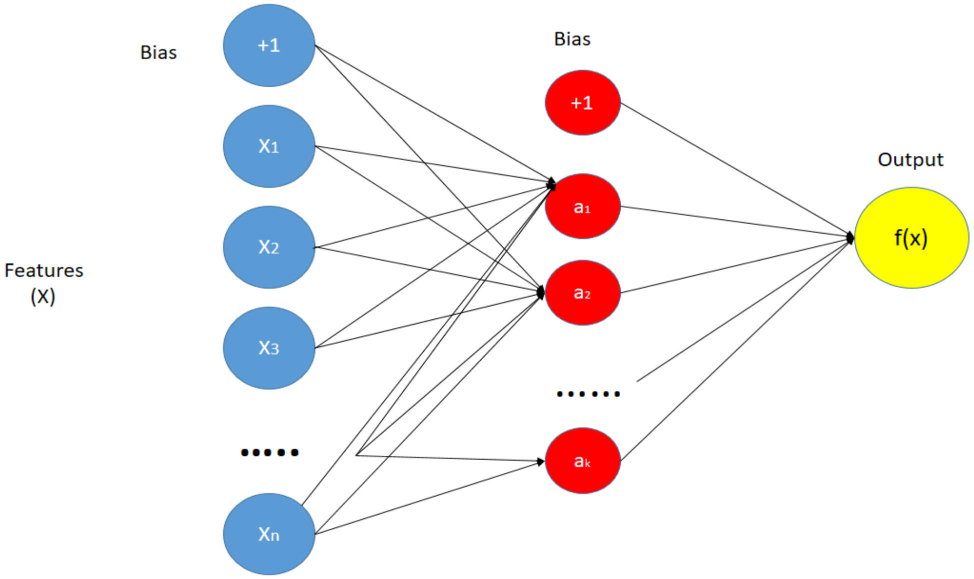


Figure 2:A multilayer perceptron model with 1 hidden layer.

Back-propagation

Back-propagation in neural networks describes the process of using a local error of the network to readjust the weights of the interconnections backwards through the neural net. Explicitly, this means that after a prediction for a set of input values has been made, the actual output value is compared to the prediction value and an error is calculated. This error is then used to readjust the weights of the connections starting at the edges that are directly connected to the output nodes of the network and then proceeding further into it [39].

The back-propagation algorithm provides an “approximation” to the trajectory in weight space computed by the method of steepest descent. The smaller we make the learning rate parameter η, the smaller the changes to the synaptic weights in the network will be from one iteration to the next, and the smoother will be the trajectory in weight space. This improvement, however, is attained at the cost of a slower rate of learning. If, on the other hand, we make the learning-rate parameter η too large in order to speed up the rate of learning, the resulting large changes in the synaptic weights assume such a form that the network may become unstable (i.e., oscillatory). A simple method of increasing the rate of learning while avoiding the danger of instability is to modify the delta rule by including a momentum term.

Haykin [37] gives the formulation of the back- propagation algorithm by stating that the algorithm cycles through the training sample {x(n), d(n)} as follows:

1. **Initialization:** Assuming that no prior information is available, pick the synaptic weights and thresholds from a uniform distribution whose mean is zero and whose variance is chosen to make the standard deviation of the induced local fields of the neurons lie at the transition between the linear and standards parts of the sigmoid activation function.
2. **Presentations of Training Examples:** Present the network an epoch of training examples. For each example in the sample, ordered in some fashion, perform the sequence of forward and backward computations described under points 3 and 4, respectively.
3. **Forward Computation:** Let a training example in the epoch be denoted by (x(n), d(n)), with the input vector x(n) applied to the input layer of sensory nodes and the desired response vector d(n) presented to the output layer of computation nodes. Compute the induced local fields and function signals of the network by proceeding forward through the network, layer by layer. The induced local field for neuron j in layer l is:

   $$v_j^{(l)}(n) = \sum_i w_{ji}^{(l)}(n) y_i^{(l-1)}(n)$$

   Assuming the use of a sigmoid function, the output signal of neuron j in layer l is:

   $$y_j^{(l)} = \varphi_j(v_j(n))$$

   If neuron j is in the first hidden layer (i.e., l=1), set:

   $$y_j^{(0)}(n) = x_j(n)$$

   Where x_j (n) is the j^th element of the input vector x(n). If neuron j is in the output layer (i.e., l= L, where L is referred to as the depth of the network), set:

   $$y_j^{(L)} = o_j(n)$$

   Compute the error signal:

   $$e_j(n) = d_j(n) - o_j(n)$$

4. **Backward Computation:** Compute the δs (i.e., local gradients) of the network, defined by:

   $$\delta_j^{(l)}(n) = \begin{cases} e_j^{(L)}(n)\varphi_j'(v_j^{(L)}(n)) & \text{for neuron } j \text{ in output layer } L \\ \varphi_j'(v_j^{(l)}(n)) \sum_k \delta_k^{(l+1)}(n) w_{kj}^{(l+1)}(n) & \text{for neuron } j \text{ in hidden layer } l \end{cases}$$

Adjust the synaptic weights of the network in layer l according to the generalized delta rule:

$$w_{ji}^{(l)}(n+1) = w_{ji}^{(l)}(n) + \alpha[\Delta w_{ji}^{(l)}(n-1)] + \eta\delta_j^{(l)}(n)y_i^{(l-1)}(n)$$

Where η is the learning-rate parameter and α is the momentum constant.

5. **Iteration:** Iterate the forward and backward computations under points 3 and4 by presenting new epochs of training examples to the network until the chosen stopping criterion is met.

Stopping Criteria for Back-propagation:

In general, the back-propagation algorithm cannot be shown to converge, and there are no well-defined criteria for stopping its operation. Rather, there are some reasonable criteria, each with its own practical merit that may be used to terminate the weight adjustments. They are:

- The back-propagation algorithm is considered to have converged when the Euclidean norm of the gradient vector reaches a sufficiently small gradient threshold [40].
- The back-propagation algorithm is considered to have converged when the absolute rate of change in the average squared error per epoch is sufficiently small.

2.2.2 XGBoost

As the introductory paper by Chen and Guestrin [41], XGBoost is a scalable Tree Boosting System. It is actually the implementation of gradient boosting trees designed for speed and performance. The main contributions of XGBoost are as follows:

- Designing and building a highly scalable end-to-end tree boosting system.
- A theoretically justified weighted quantile sketch for efficient proposal calculation.
- Introduction of a novel sparsity-aware algorithm for parallel tree learning.
- An effective cache-aware block structure for out-of-core tree learning.

The gradient boosting decision tree algorithm is implemented in the XGBoost library. Boosting is an ensemble technique that adds new models to correct errors made by existing models. Models are added in a sequential order until no further advancements can be made. Gradient boosting is a method in which new models are created that predict the residuals or errors of prior models and are then combined to make the final prediction. It is called gradient boosting because it employs a gradient descent algorithm to reduce loss when adding new models. Figure 3 shows the evolution of XGBoost from Decision Trees.

Since its introduction, this algorithm has not only been credited with winning numerous Kaggle competitions but also for being the driving force under the hood for several cutting-edge industry applications. As a result, there is a strong community of data scientists contributing to the XGBoost open-source projects with ~350 contributors and ~3,600 commits on GitHub. [42].

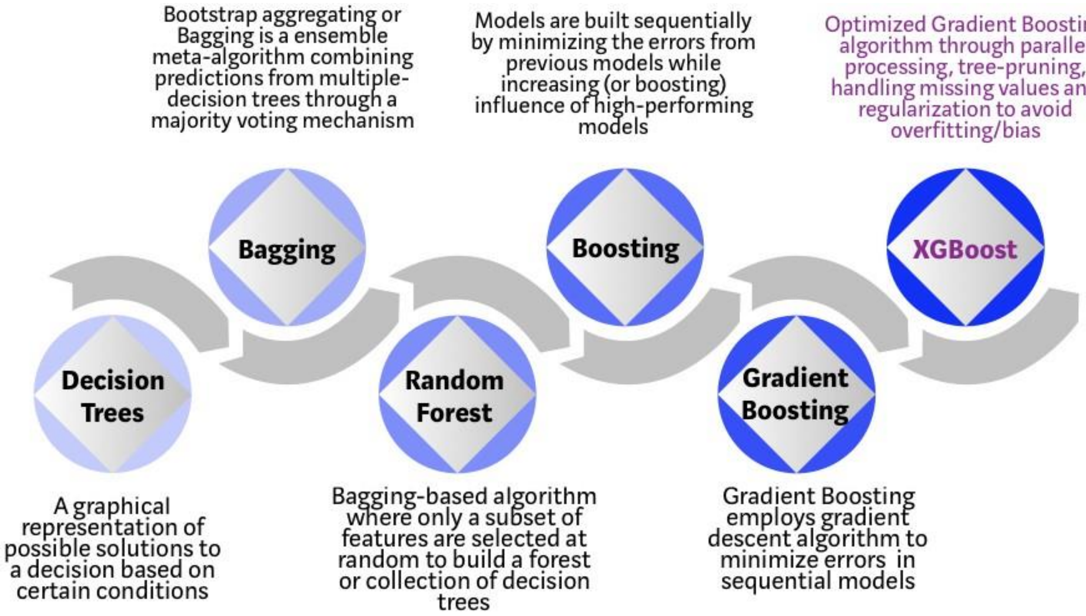

Bootstrap aggregating or Bagging is a ensemble meta-algorithm combining predictions from multiple-decision trees through a majority voting mechanism
Models are built sequentially by minimizing the errors from previous models while increasing (or boosting) influence of high-performing models
Optimized Gradient Boosting algorithm through parallel processing, tree-pruning, handling missing values and regularization to avoid overfitting/bias
Bagging
Boosting
XGBoost
Decision Trees
Random Forest
Gradient Boosting
A graphical representation of possible solutions to a decision based on certain conditions
Bagging-based algorithm where only a subset of features are selected at random to build a forest or collection of decision trees
Gradient Boosting employs gradient descent algorithm to minimize errors in sequential models

Figure 3:Evolution of XGBoost Algorithm from Decision Trees.

Source: https://miro.medium.com/max/1400/1*QJZ6W-Pck_W7RlIDwUIN9Q.jpeg

XGBoost is an ensemble learning method. Ensemble learning provides an answer to amalgamate the predictive power of multiple learners. The resultant is a single model which gives the aggregated output from several models.

The models that form the ensemble, also known as base learners, could be either from the same learning algorithm or different learning algorithms. Bagging and boosting are two widely used ensemble learners. Though these two techniques can be used with several statistical models, the most predominant usage has been with decision trees.

Bootstrap aggregation, or bagging, is a general-purpose procedure for reducing the variance of a statistical learning method; we introduce it here because it is particularly useful and frequently used in the context of decision trees. Given a set of n independent observations Z_1, . . ., Zn, each with variance σ^2, the variance of the mean of the observations is given by σ^2/n. In other words, averaging a set of observations reduces variance.

While bagging can help with many regression methods, it is especially useful for decision trees. To apply bagging to regression trees, we simply build B regression trees with B bootstrapped training sets and average the predictions. These trees have a deep root system and are not pruned. As a result, each individual tree has a high variance but a low bias. The variance is reduced by averaging these B trees. Bagging has been shown to improve accuracy significantly by combining hundreds or even thousands of trees into a single procedure.

Bagging or boosting aggregation helps to minimize any learner’s variation. The foundational learners of bagging technique are several decision trees generated in parallel. These learners are fed data sampled with replacement. The last forecast is the mean performance of all the students. Unlike packaging techniques like Random Forests, where trees are cultivated to their fullest extent, the boosting process uses less divided trees. However, it could lead to a large number of trees. Therefore, the stop criteria for boosting must be selected carefully.

Figure 4 elaborates the boosting procedure. The train dataset is fed into classifier number one. The yellow background denotes that the classifier predicted hyphen, while the blue background denotes that it predicted plus. The classifier 1 model predicts two hyphens and one plus incorrectly. These are denoted by a circle. These incorrectly predicted data points have their weights increased and are sent to the next classifier. This refers to classifier 2. Classifier 2 correctly predicts the two hyphens that classifier 1 did not. However, classifier 2 makes a few other mistakes. This process is repeated until we have a combined final classifier that correctly predicts all of the data points. The classifier models can be added until all the items in the training dataset is predicted correctly or a maximum number of classifier models are added. The optimal maximum number of classifier models to train can be determined using hyper-parameter tuning.

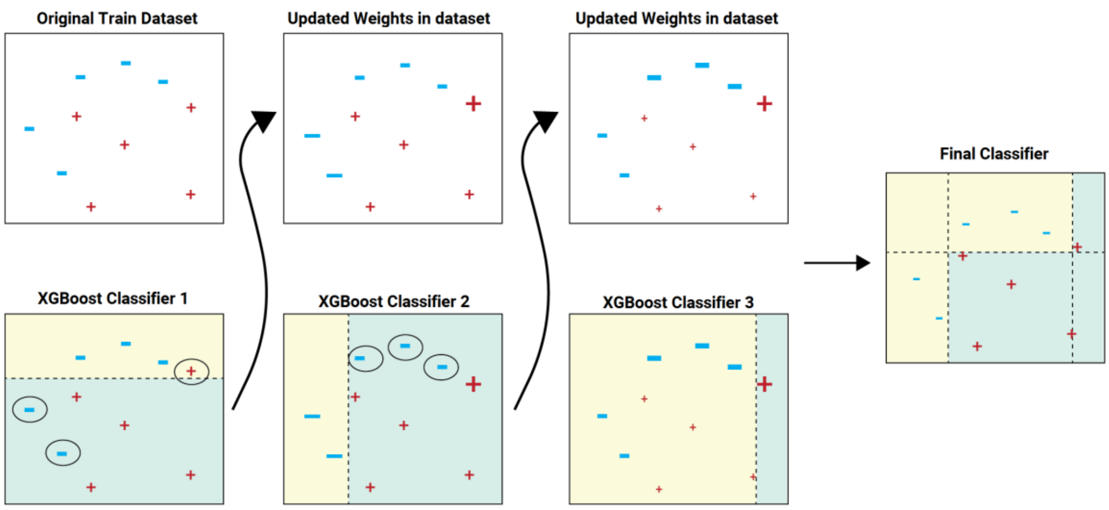


Figure 4: Boosting Procedure.

Source: https://blog.quantinsti.com/xgboost-python/

A Boosting Ensemble Technique consists of the following steps:

1. An initial model F0 is defined to predict the target variable y. This model will be associated with a residual (y – F0)
2. A new model h1 is fit to the residuals from the previous step
3. Now, F0 and h1 are combined to give F1, the boosted version of F0. The mean squared error from F1 will be lower than that from F0:
$$F_1(x) = F_0(x) + h_1(x)$$
To improve the performance of F1, we could model after the residuals of F1 and create a new model F2:
$$F_2(x) = F_1(x) + h_2(x)$$
This can be done for 'm' iterations, until residuals have been minimized as much as possible:
$$F_m(x) = F_{m-1}(x) + h_m(x)$$

Gradient descent helps us to reduce differentiated functions to a minimum. Previously for each terminal node of the tree, the return tree for $h_m(x)$ foresees the mean residual. The average gradient component is calculated in gradient boosting.

There is a γ factor for each node, multiplying $h_m(x)$. The effects of each branch of the split are different. In contrast to the classical gradient descent techniques which reduce error on the output at each iteration, the gradient boost helps to predict an optimum gradient for the added model.

The following steps are involved in gradient boosting:

1. **Definition:** Define F_0 by:

$$F_0(x) = argmin_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma)$$

2. **Iteration:** The gradient of the loss function is computed iteratively:

$$r_{im} = -\alpha \left[\frac{\partial(L(y_i, F(x_i)))}{\partial F(x_i)} \right]_{F(x)=F_{m-1}(x)}, \text{ where } \alpha \text{ is the learning rate}$$

3. Each $h_m(x)$ is fit on the gradient obtained at each step
4. The multiplicative factor γ_m for each terminal node is derived and the boosted model $F_m(x)$ is defined:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$$

The principal advantage of the XGBoost algorithm is that it can penalize complex models by regularizing both L1 and L2. Regularization contributes to over fitting prevention. In addition, XGBoost can use multiple cores on the CPU for faster computing. Due to a block construction in its system design, this is possible. In the memory units called blocks the data is sorted and stored. This allows the data layout, unlike other algorithms, to be reused by subsequent iterations rather than re-calculated. This feature also helps to find divisions and sub-samples of columns.

For XGBoost to receive the statistics by row index, non-continuous memory access is required. XGBoost has therefore been designed to use hardware optimally. This is done by assigning internal buffers to every thread that can store the gradient stats. Missing values or data processing steps like one-hot encoding make data sparse. XGBoost incorporates a split algorithm to determine the sparsity pattern of different data types.

Thus, XGBoost is a really useful performance boosting method. In our model (as shown later) XGBoost helps to increase the accuracy of the model drastically, which allows it to beat the performance measures of the models used in previous works of research.

2.3 Data Scaling

To put all features into the same standing in many machine learning methods, we must scale so that one big number does not damage the model simply because of its high magnitude. One of the most important phases in the pre-processing of data before developing a machine learning model is feature scaling in machine learning. Scaling can be the difference between a poor and a good machine learning model.

Machine learning methods that determine distances between data require feature scaling. If the feature does not scale, the feature with a larger value range begins to dominate while calculating distances. The ML method is sensitive to “relative scales of features,” which occurs when the features are represented numerically rather than by their rank. When we want faster convergence in numerous algorithms, such as neural networks, scaling is a must. Because the range of values in raw data fluctuates greatly, objective functions in some machine learning algorithms do not work correctly without normalization. Some methods that have been used are as follows:

- **Standardization:** Many machine learning estimators implemented in scikit-learn require dataset standardization; otherwise, they may behave poorly if the individual features do not resemble standard normally distributed data: Gaussian having a mean of zero and a variance of one. In practice, we frequently ignore the distribution's structure and simply convert the data to center it by deleting the mean value of each feature, then scale it by dividing non-constant characteristics by their standard deviation.
  For instance, many elements used in the objective function of a learning algorithm (such as the RBF kernel of Support Vector Machines or the l1 and l2 regularizes of linear models) assume that all features are centered on zero and have variance in the same order. If a feature has a variance that is orders of magnitude larger than others, it might dominate the objective function and make the estimator unable to learn from other features correctly as expected.
- **Maximum Absolute Value Scaling:** In this method, the features are scaled by their maximum absolute value. This estimator scales and translates each feature individually such that the maximal absolute value of each feature in the training set will be 1.0. It does not shift/center the data, and thus does not destroy any sparsity. This scaler can also be applied to sparse CSR or CSC matrices.
- **Scaling Data with outliers:** If the data contains many outliers, scaling using the mean and variance of the data is likely to not work very well. In these cases, you can use RobustScaler as a drop-in replacement instead. It uses more robust estimates for the center and range of the data. RobustScaler removes median and scales according to interquartile range.

2.4 Feature Selection Methods

The dimensionality of a dataset is defined as the number of input variables or characteristics. Dimensionality reduction techniques are those that reduce the number of variables in a dataset. More input features frequently make a predictive modelling task more difficult to model, which is known as the curse of dimensionality. For data visualization, high-dimensionality statistics and dimensionality reduction techniques are frequently used. Nonetheless, in applied machine learning, these techniques can be used to simplify a classification or regression dataset in order to better fit a predictive model.

According to Der Maaten et al. [44], if the dataset is represented by a (n X D) matrix **X** consisting of n data vectors x_i (I ϵ {1, 2,….,n}) with dimensionality D. Assume that this dataset has intrinsic dimensionality d (where d < D, and often d << D). In this case, intrinsic dimensionality refers to the fact that the points in dataset X are located on or near a manifold of dimensionality d that is embedded in the D-dimensional space. The manifold may be non-Riemannian because of discontinuities (i.e., the manifold may consist of a number of disconnected sub manifolds). Dimensionality reduction techniques transform dataset X with dimensionality D into a new dataset Y with dimensionality d, while retaining the geometry of the data as much as possible. In general, neither the geometry of the data manifold, nor the intrinsic dimensionality d of the dataset X are known. Therefore, dimensionality reduction is an ill-posed problem that can only be solved by assuming certain properties of the data (such as its intrinsic dimensionality).

Fewer input dimensions frequently imply fewer parameters or a simpler structure in the machine learning model, which is referred to as degrees of freedom. A model with too many degrees of freedom is likely to over fit the training dataset and so perform poorly on new data. Simple models that generalize well, and hence input data with few input variables, are preferred. This is especially true for linear models, where the number of inputs and degrees of freedom are frequently connected.

Dimensionality reduction is a data preparation technique used on data prior to modelling. It could be done after data cleansing and scaling and before building a prediction model. As a result, any dimensionality reduction performed on training data must also be performed on new data, such as a test dataset, validation dataset, and data when making a prediction with the final model.

Amongst the various dimensionality reduction and feature selection methods, we will highlight two methods that have been used in this work.

2.4.1 Recursive Feature Elimination (RFE)

Recursive Feature Elimination (RFE) is an extremely popular feature selection method. RFE is popular because it is simple to set up and use, and it is effective in identifying the features (columns) in a training dataset that are more or more relevant in predicting the target variable.

When utilizing RFE, there are two crucial configuration options: the number of features to pick and the algorithm used to help choose features. Both of these hyper parameters can be investigated, albeit the method’s success is not heavily reliant on them being properly configured.

RFE is a wrapper-type feature selection algorithm. This means that a different machine learning algorithm is given and used in the core of the method, is wrapped by RFE, and used to help select features. This is in contrast to filter-based feature selections that score each feature and select those features with the largest (or smallest) score. Technically, RFE is a wrapper-style feature selection algorithm that also uses filter-based feature selection internally. RFE works by searching for a subset of features by starting with all features in the training dataset and successfully removing features until the desired number remains.

We chose to use ensemble methods as our feature selection algorithm as well. As we have discussed about ensemble techniques, bagging and boosting in the previous section, we are not repeating the same here.

In our RFE, we have used Extremely Randomized Trees proposed by Geurts et al. [31] as our Machine Learning Algorithm. As is the known fact, ensemble methods generally work better than Decision Trees. Amongst the ensemble methods, Random Forest and Extremely Randomized Trees are very similar. Both are made up of a huge number of decision trees, with the ultimate choice taking into account the predictions of each tree. In particular, in classification problems, the majority vote is used, and in regression problems, the arithmetic mean is used. In addition, both methods use the same growing tree procedure (with one exception explained below). Furthermore, while choosing the partition of each node, they both choose a subset of features at random.

The major differences between these algorithms are:

- Random forest uses bootstrap replicas, that is to say, it subsamples the input data with replacement, whereas Extra Trees use the whole original sample. In the Extra Trees sklearn implementation there is an optional parameter that allows users to bootstrap replicas, but by default, it uses the entire input sample. This may increase variance because bootstrapping makes it more diversified.
- Random Forest chooses the optimum split while Extra Trees chooses it randomly. However, once the split points are selected, the two algorithms choose the best one between all the subset of features. Therefore, Extra Trees adds randomization but still has optimization.

These distinctions drive the lowering of both bias and variation. On the one hand, employing the entire original sample rather than a bootstrap replica reduces bias. Choosing the split point of each node at random, on the other hand, will reduce variance. The Extra Trees algorithm is faster in terms of computational cost and thus execution time. This technique saves time because the entire operation is the same, but it chooses the split point at random rather than calculating the ideal one. Extra Trees gets their name from these factors (Extremely Randomized Trees).

2.4.2 Select from Model

This is a meta- transformer feature selection method that can be used alongside any estimator that assigns importance to each feature through a specific attribute (weights).

The features are considered unimportant and removed if the corresponding importance of the feature values are below the provided threshold parameter. Apart from specifying the threshold numerically, there are built-in heuristics for finding a threshold using a string argument.

As in the meta-transformer we have again used the same Extremely Randomized Trees as our tree- based estimator to compute impurity-based feature importance, which in turn can be used to discard irrelevant features.

2.5 Performance Metrics

To measure the performance of the Binary classification, true positives (TP), true negatives (TN), false positives (FP), false negatives (FN) and confusion matrix (Table 1) are considered. These parameters are used to compute the performance measures like Accuracy, Precision, and Recall (Table 2).

But in the case of multiclass classification problem, as in the case of our work, true positives (TP_i), true negatives (TN_i), false positives (FP_i), false negatives (FN_i) for ith class are computed.

Data Class	Classified as Positive	Classified as Negative
Positive Class	True Positive (TP)	False Negative (FN)
Negative Class	False Positive (FP)	False Positive (FP)

Table 1: Confusion Matrix for Binary Classification

Measure	Formula	Evaluation Focus
Accuracy	$\frac{TP + TN}{TP + FP + TN + FN}$	Overall effectiveness of a classifier
Precision	$\frac{TP}{TP + FP}$	Class agreement of the data labels with the positive class

		labels given by the classifiers
Recall	$\frac{TP}{TP + FN}$	Effectiveness of a classifier to identify positive labels

Table 2: Performance Metrics for Binary Classification

Using these parameters, the Average Accuracy, Precision$_{\mu}$, Recall$_{\mu}$, Precision$_{M}$ and Recall$_{M}$ are computed (Table 3) [43].

Measure	Formula	Evaluation Focus
Average Accuracy	$\frac{\sum_{i=1}^{l} \frac{TPi+TNi}{TPi+TNi+FPi+FNi}}{l}$	The Average per-class effectiveness of a classifier.
Precision$_{\mu}$	$\frac{\sum_{i=1}^{l} TPi}{\sum_{i=1}^{l} TPi + FPi}$	Arrangement of the data class labels with those of classifiers if calculated from sums of pretext decisions.
Recall$_{\mu}$	$\frac{\sum_{i=1}^{l} TPi}{\sum_{i=1}^{l} TPi + FNi}$	Effectiveness of a classifier to identify class labels if calculated from sums of pre-text decisions.
Precision$_{M}$	$\frac{\frac{\sum_{i=1}^{l} TPi}{\sum_{i=1}^{l} TPi+FPi}}{l}$	An average per-class agreement of the data class labels with those of classifiers.
Recall$_{M}$	$\frac{\frac{\sum_{i=1}^{l} TPi}{\sum_{i=1}^{l} TPi+FNi}}{l}$	An average per-class effectiveness of a classifier to identify class labels.

Table 3: Performance Measure of Multiclass Classifier

3 Results/ Empirical Data

This chapter will give an overview over the achieved results, the used data and the experiment process to solve the given research questions. The following questions will be answered in detail, with Section 3.1 focussing on the dataset in general, Section 3.2 describing the model architecture and Section 3.3 presenting the achieved results.

3.1 Empirical Data

The following section gives an outlook about the data described used in the process. This work has obtained the popular UCI Machine Learning repository [45]. As our job is to create a model that will determine the obesity condition of a particular individual based on the individual's habit, we find that we are dealing with a multi-class classification model. The dataset was based on a study performed which was used as a primary source of the data collected from a set of students of institutions of Colombia, Mexico, and Peru. The students were aged between 18 to 25 years. The data can be used for estimation of the obesity level of individuals using seven categories, allowing a detailed analysis of the affectation level of an individual. The dataset description is given in Table 4.

Attributes	Values
Gender	• Male • Female
Age	Numeric value in years.
Height	Numeric Value in metres.
Weight	Numeric Value in Kilograms.
Family Member Overweight	• Yes • No
Regular High Calorie Food Intake	• Yes • No
Consumption of Vegetables in Meals	• Never • Sometimes • Always

Number of Daily Main Meals Intake	• Between 1-2 • 3 • More than 3
Food Intake between Meals	• No • Sometimes • Frequently • Always
Smoke	• Yes • No
Daily Water Intake	• Less than 1 litre • 1-2 litre • More than 2 litres
Calories Consumption Calculation	• Yes • No
Physical Activity in a Week	• No physical activity • 1-2 days • 2-4 days • 4-5 days
Alcohol Consumption	• Non drinker • Sometimes • Frequently • Always
Schedule Given to Technology	• 0-2 hours • 3-5 hours • More than 5 hours
Mode of Transportation Used	• Automobile • Motorbike • Bike • Public Transportation • Walking

Table 4: Dataset Description

The Data was then labeled using the formula for BMI, given by:

$$\text{BMI (in Kg/m}^2\text{)} = \frac{Weight\ (In\ Kg)}{Height\ (inm)\ X\ \ Height\ (in\ m)}$$

Table 5 elaborates the obesity classification with the data provided by WHO and the Mexican Normativity [46].

BMI Range (in Kg/m^2)	Obesity Classification
Less than 18.5	Underweight
18.5 – 24.9	Normal Weight
25.0 – 29.9	Obesity
30.0 – 34.9	Obesity I
35.0 – 39.9	Obesity II
40 and above	Obesity III

Table 5: BMI classification according to WHO and Mexican normativity

The categories of obesity levels were uneven after the labelling procedure had ended, and this posed a learning challenge for the methods of data mining, as the category could be correctly identified with most records compared to categories with fewer data. In [47], if the classifying categories are not evenly represented, you can tell a data collection is not balanced.

Upon identification of the balanced class problem, the tool Weka and SMOTE filter proposed by [47] produced synthetic information up to 77 percent of the data. The filter needed to designate the class for synthetic data production, the number of neighbours that are nearest, the percentage required for increased class selection and the random seed used for random sampling. The identification of unusual and missing information was also evaluated. Finally, the final outcome of the filter was 2111 records for each category.

It is vital to note that data (deletion of missing data, unusual data or normalisation, etc.) must be prepared before SMOTE is used, because it could include noises or disturbances from the selected neighbour to make synthetic data, and the resulting data would be of bad quality. Nevertheless, utilising the SMOTE filter has a favourable influence when the data is imbalanced since the balance process lowers the chances that the majority class will receive skewed learning. Features included in the data were chose based in literacy analysis such as [47], [48], [49], [50], [51], [52] and [53].

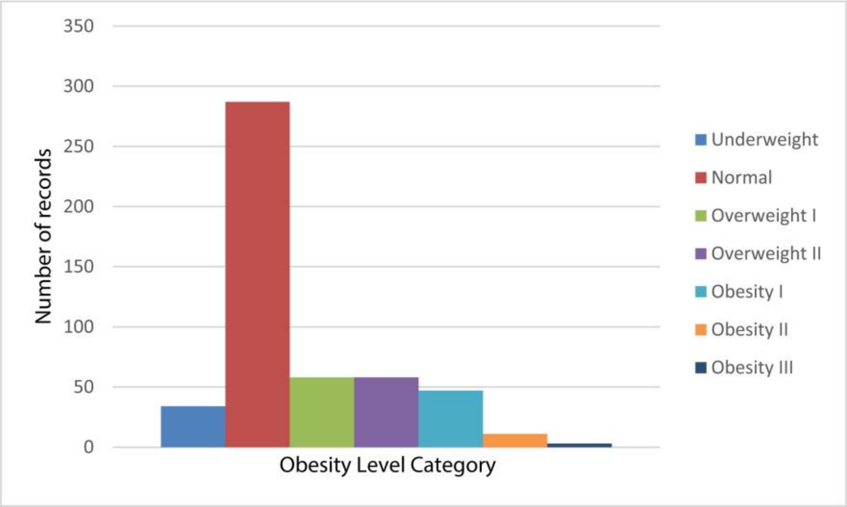


Figure 5: Unbalanced distribution of data regarding the obesity levels category

Source: https://www.sciencedirect.com/science/article/pii/S2352340919306985?via%3Dihub

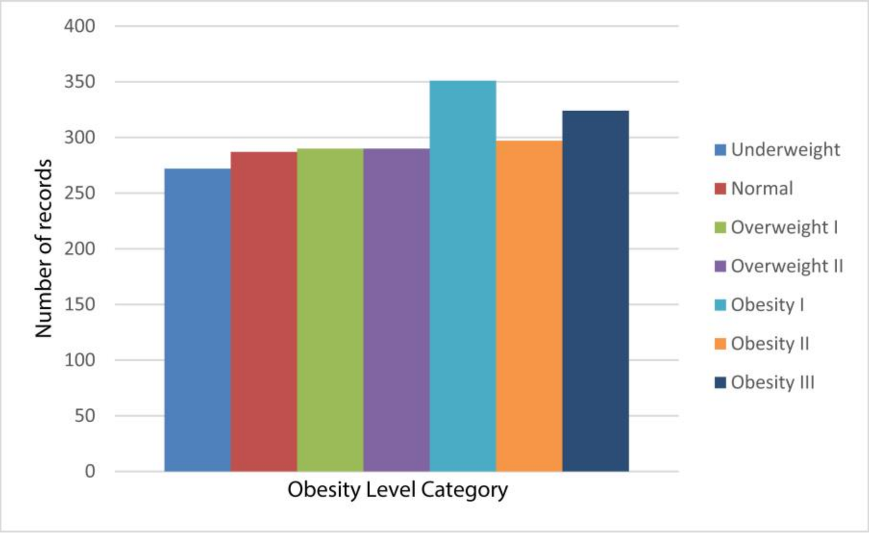
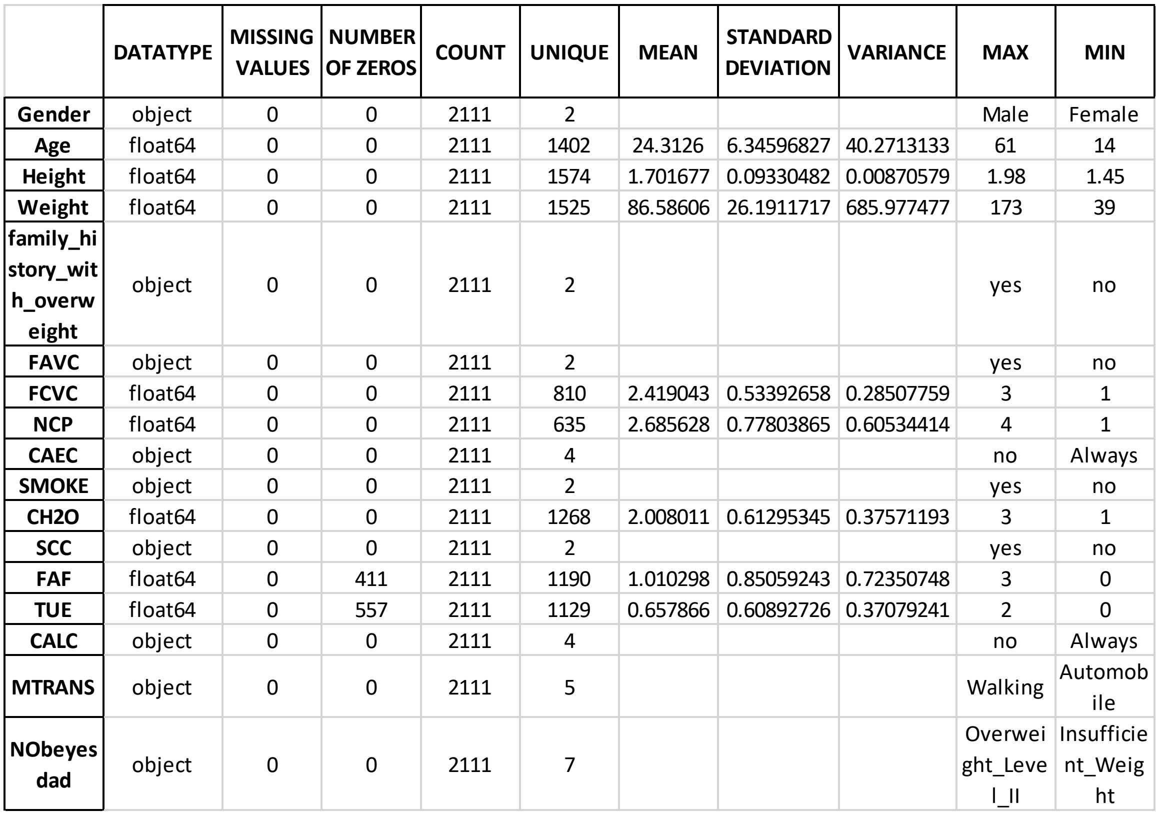


Figure 6: Balanced Distribution of data regarding the obesity levels category

Source: https://www.sciencedirect.com/science/article/pii/S2352340919306985?via%3Dihub

This means that our classification problem is thus a 6-class classification problem.



	DATATYPE	MISSING VALUES	NUMBER OF ZEROS	COUNT	UNIQUE	MEAN	STANDARD DEVIATION	VARIANCE	MAX	MIN
Gender	object	0	0	2111	2				Male	Female
Age	float64	0	0	2111	1402	24.3126	6.34596827	40.2713133	61	14
Height	float64	0	0	2111	1574	1.701677	0.09330482	0.00870579	1.98	1.45
Weight	float64	0	0	2111	1525	86.58606	26.1911717	685.977477	173	39
family_history_with_overweight	object	0	0	2111	2				yes	no
FAVC	object	0	0	2111	2				yes	no
FCVC	float64	0	0	2111	810	2.419043	0.53392658	0.28507759	3	1
NCP	float64	0	0	2111	635	2.685628	0.77803865	0.60534414	4	1
CAEC	object	0	0	2111	4				no	Always
SMOKE	object	0	0	2111	2				yes	no
CH2O	float64	0	0	2111	1268	2.008011	0.61295345	0.37571193	3	1
SCC	object	0	0	2111	2				yes	no
FAF	float64	0	411	2111	1190	1.010298	0.85059243	0.72350748	3	0
TUE	float64	0	557	2111	1129	0.657866	0.60892726	0.37079241	2	0
CALC	object	0	0	2111	4				no	Always
MTRANS	object	0	0	2111	5				Walking	Automobile
NObeyesdad	object	0	0	2111	7				Overweight_Level_II	Insufficient_Weight

Table 6: Descriptive Statistics of the Dataset

3.2 Model Architecture

As the theoretical backgrounds about the Machine Learning Techniques, Feature Extraction and Feature Scaling has been elaborated in the previous section, we are free to refer to them in order to explain the model architecture. As popularly known terminology in Machine Learning there is no ‘best’ Machine Learning Algorithm. There can be a variety of techniques and models which are available. However, if one models fit extremely well in one particular problem/ dataset, it is not guaranteed that the particular technique will fit well in another problem/ dataset as well. However, with a wide range of Machine Learning Techniques and Algorithms available for a research in the current age, it is extremely important to judiciously choose and find the best method and have some sort of an intuition for defining an appropriate architecture. Figure 7 shows the generalised knowledge discovery process.

containing the taken steps from gathering the data to interpreting the results and using them for further tasks

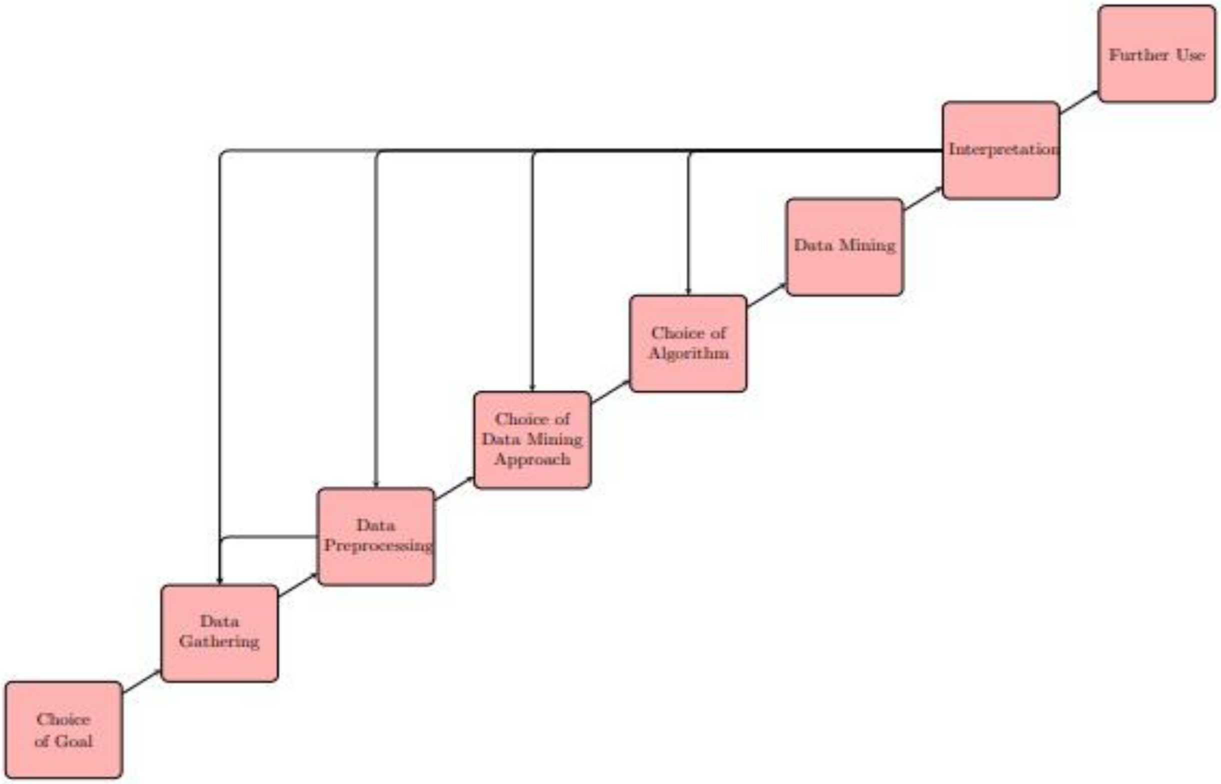


Figure 7: Generalized Knowledge Discovery Process containing the taken steps from gathering the data to interpreting the results and using them for further tasks

Source: https://www.diva-portal.org/smash/get/diva2:920202/FULLTEXT01.pdf

As we observed prior works using this dataset, [53] had used decision trees using the Weka software and achieved high precision and recall values using the J48 algorithm. However, we noticed that in the current Kaggle competitions and research work XGBoost seemed to beat baseline tree-based models in terms of predictive performance significantly. This was the intuition behind choosing the XGBoost Model. Also, another thought process that was passed was to try to maximize the usage of the Multilayer Perceptron Model as it had been used widely for classification problems in the current Financial Modelling industry.

Albeit the current problem statement was not in any way linked to Finance, however, the generalised effort to model classification problems was the chief motivation behind choosing the model. However, as this work attempted to utilize a hybrid model for solving this 6-class classification problem, we used the popular automated Machine Learning tool TPOT [54], [55], [56] that optimizes machine learning pipelines using genetic programming. TPOT will automate the most tedious part of machine learning by intelligently exploring thousands of possible pipelines to find the best one for the data.

Once TPOT is finished searching (or one gets tired of waiting), it provides with the Python code for the best pipeline it found so one can tinker with the pipeline from there.

Thus, our model architecture and steps are as follows:

- The string-type categorical features as well as the target variable (type of obesity) in the data is One-Hot Encoded (ie, creating a binary expression of all the string-type categorical features). We then get a data whose all features are either Integer-type or Float-type.
- We apply z-score feature transformation to all the float type feature columns except the columns which have been One-Hot Encoded.
- We then divide our dataset into Training and Testing Samples with 75% of the whole data used to train the model and 25% of the data used to test the performance of the model.
- We perform initial feature selection using the SelectFromModel method in sci-kit learn library on Python. The tree-based algorithm that was used was Extremely Randomized Trees.
- We perform another set of feature selection process using the Recursive Feature Elimination method. We follow the sci-kit learn library on Python. In this case, we again use a tree-based algorithm which, similar as the previous step, the Extremely Randomized Trees Algorithm.
- We then again scale the features of the data by scaling it down to the maximum absolute value scaling.
- The scaled features are then passed to the Multilayer Perceptron Classifier. The Multilayer Perceptron Classifier does an initial classification of 6 classes with intermediate float values.
- The predictions of the Multilayer Perceptron are again scaled by removing the median and scales the data according to the quantile range (defaults to IQR: Interquartile Range). The IQR is the range between the 1st quartile (25thquantile) and the 3rd quartile (75thquantile).
- The scaled outputs of the Multilayer Perceptron are then fed into the XGBoost Model and the predictions of the overall model are then generated.

Figure 8 here shows the schematic representation of the overall model and Table 7 shows the parameter values for the various techniques used.

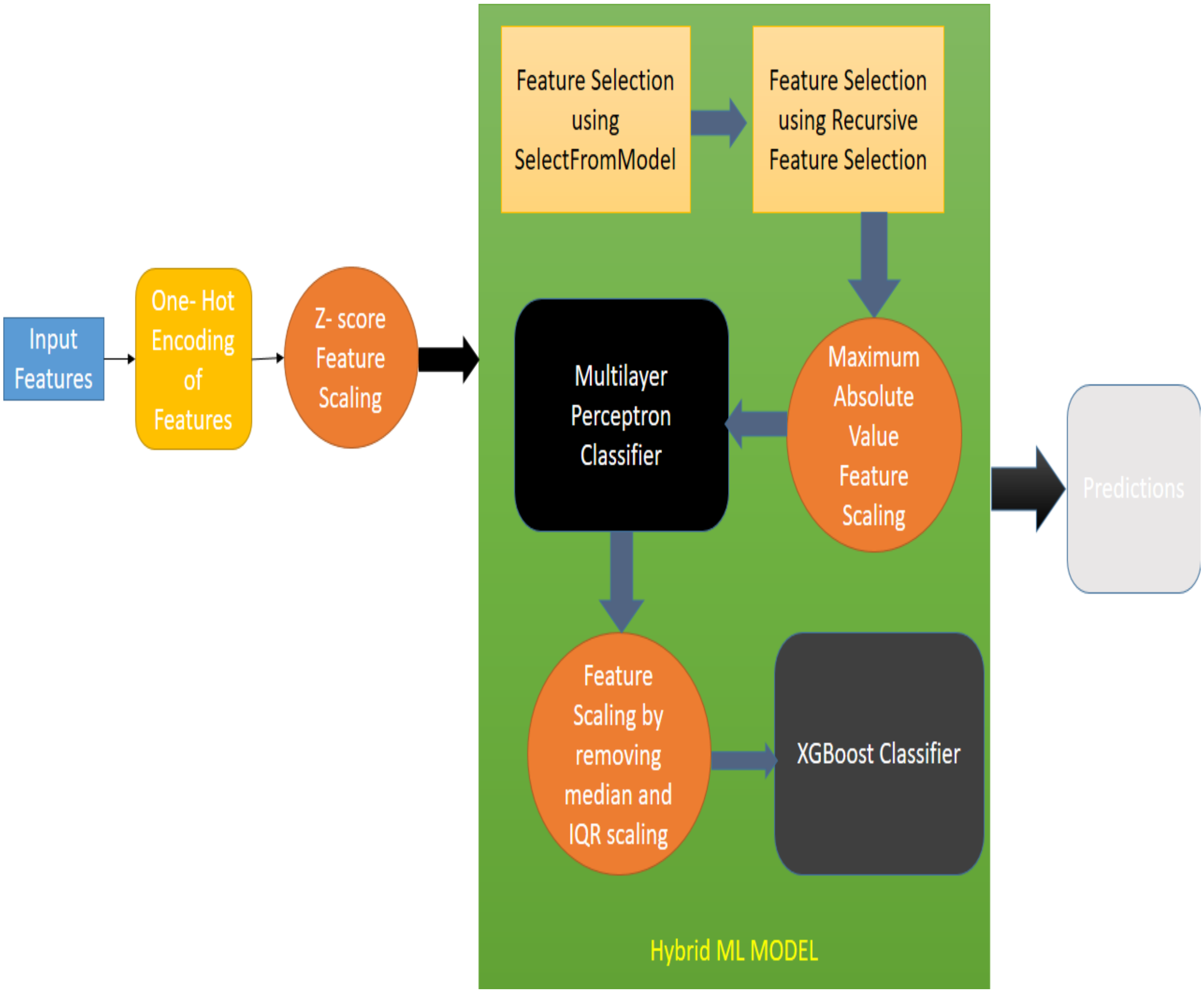

Input Features
One- Hot Encoding of Features
Z- score Feature Scaling
Feature Selection using SelectFromModel
Feature Selection using Recursive Feature Selection
Maximum Absolute Value Feature Scaling
Multilayer Perceptron Classifier
Feature Scaling by removing median and IQR scaling
XGBoost Classifier
Hybrid ML MODEL
Predictions

Figure 8: The Hybrid ML Model Architecture

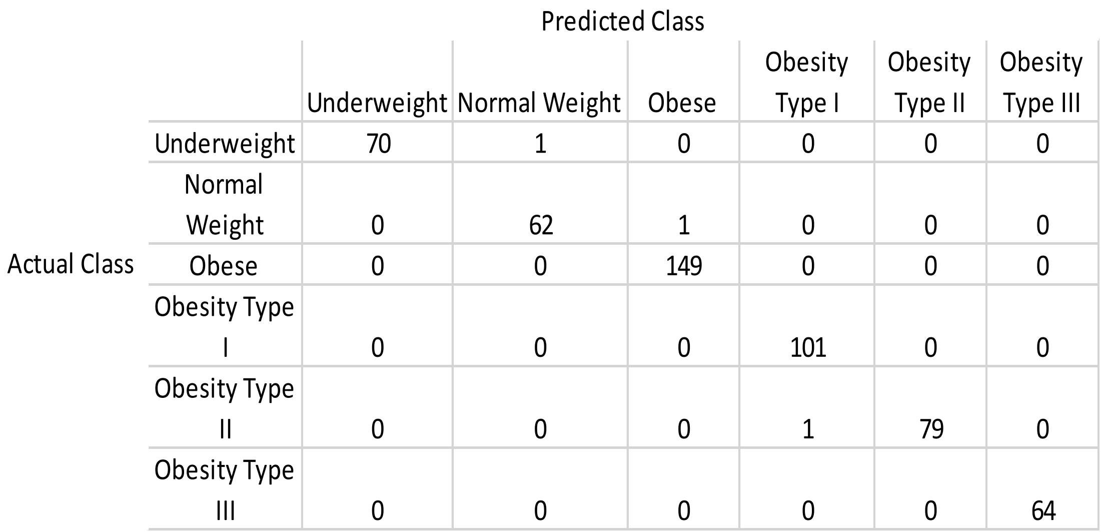
The model took 1.3 seconds to train and predict the for the test data. The machine configuration was Intel(R) Core (TM) i5-8265U CPU @ 1.60GHz   1.80 GHz, 8 GB RAM and Windows 64-bit operating system.

Model Part	Parameter Values
Feature Selection (Select From Model)	Extremely Randomized Trees Classifier: • Number of trees in forest = 100. • Criterion = Gini. • Maximum Number of Features to consider best split = 0.1
Feature Selection (Recursive Feature Elimination)	Extremely Randomized Trees Classifier: • Number of trees in forest = 100. • Criterion = Gini. • Maximum Number of Features to consider best split = 0.4
Multilayer Perceptron Classifier	• Number of hidden layers = 1. • Number of neurons in hidden layer = 100. • Optimizer = Adam (Learning rate = 0.5). • L2 Regularization Parameter value = 0.1.
XGBoost Classifier	• Learning Rate = 0.5. • Maximum Depth of a Tree = 7. • Minimum sum of instance weight (hessian) needed in a child = 1. • Subsample ratio of the training instances = 0.8500000000000001.

Table 7: Hybrid Model Parameters used

3.3 Results

To train our model, we used the training data consisting of 1583 samples and trained out hybrid model. Then we tested our model using the test data. We achieved an overall average accuracy of 99.43 %. Table 8 summarizes the confusion matrix and Table 9 summarizes the performance measures for the predictions as well.



		Predicted Class					
		Underweight	Normal Weight	Obese	Obesity Type I	Obesity Type II	Obesity Type III
	Underweight	70	1	0	0	0	0
	Normal Weight	0	62	1	0	0	0
Actual Class	Obese	0	0	149	0	0	0
	Obesity Type I	0	0	0	101	0	0
	Obesity Type II	0	0	0	1	79	0
	Obesity Type III	0	0	0	0	0	64

Table 8: Confusion Matrix for the 6-class classification Problem

Performance Measure	Value
Accuracy	99. 4318 %
Precision$_{\mu}$	0. 994349
Recall$_{\mu}$	0. 994318
Precision $_{M}$	0.994609
Recall $_{M}$	0.992923
Balanced F- Score	0. 994313

Table 9: Performance Measures for the 6-class classification Problem using Hybrid Model

One important thing to also note would be to also view the Individual Accuracy, Precision, and Balanced F-Score for the classes as well. Table 10 Elaborates that.

Class	Accuracy	Precision	Recall	f1- Score	Support
Underweight	99.81%	100%	98.59%	99.29%	71
Normal Weight	99.62%	98.41%	98.41%	98.41%	63
Obese	99.81%	99.33%	100%	99.66%	149
Obesity Type I	99.81%	99.01%	100%	99.50%	101
Obesity Type II	99.81%	100%	98.75%	99.37%	80
Obesity Type III	100%	100%	100%	100%	64

Table 10: Class Wise Performance metrics of the hybrid model

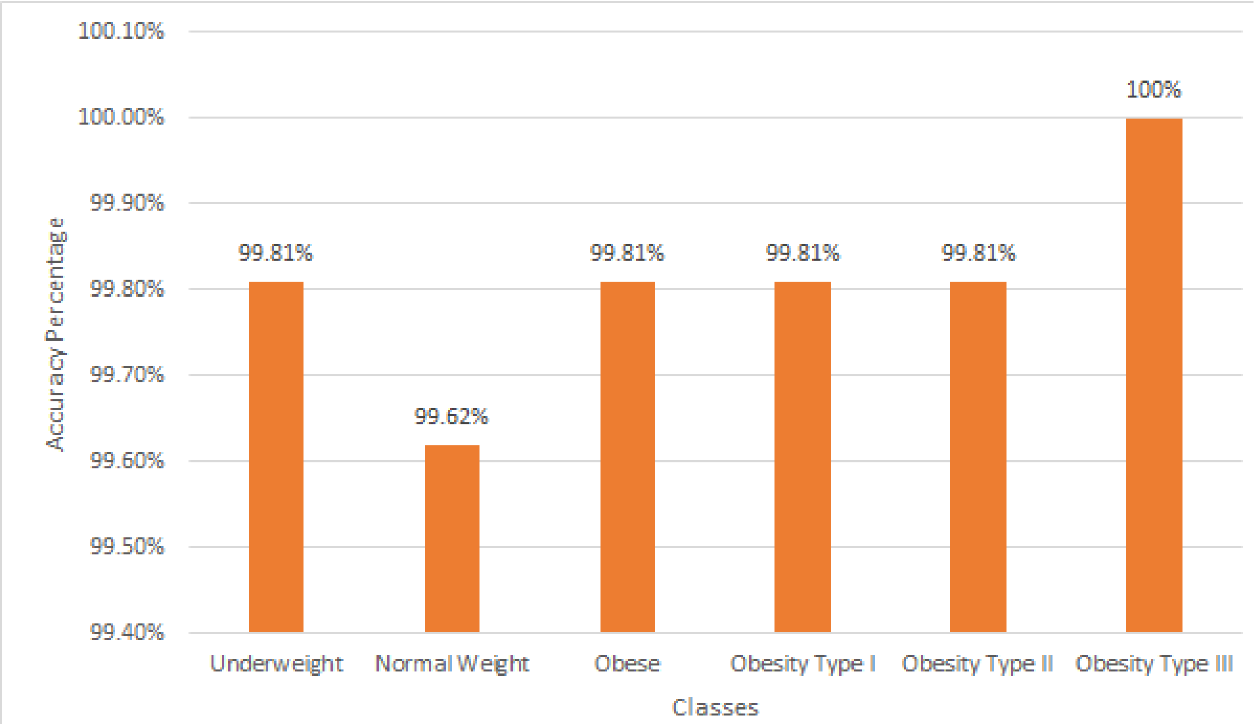


Figure 9: Class Wise Accuracy of the Hybrid Model

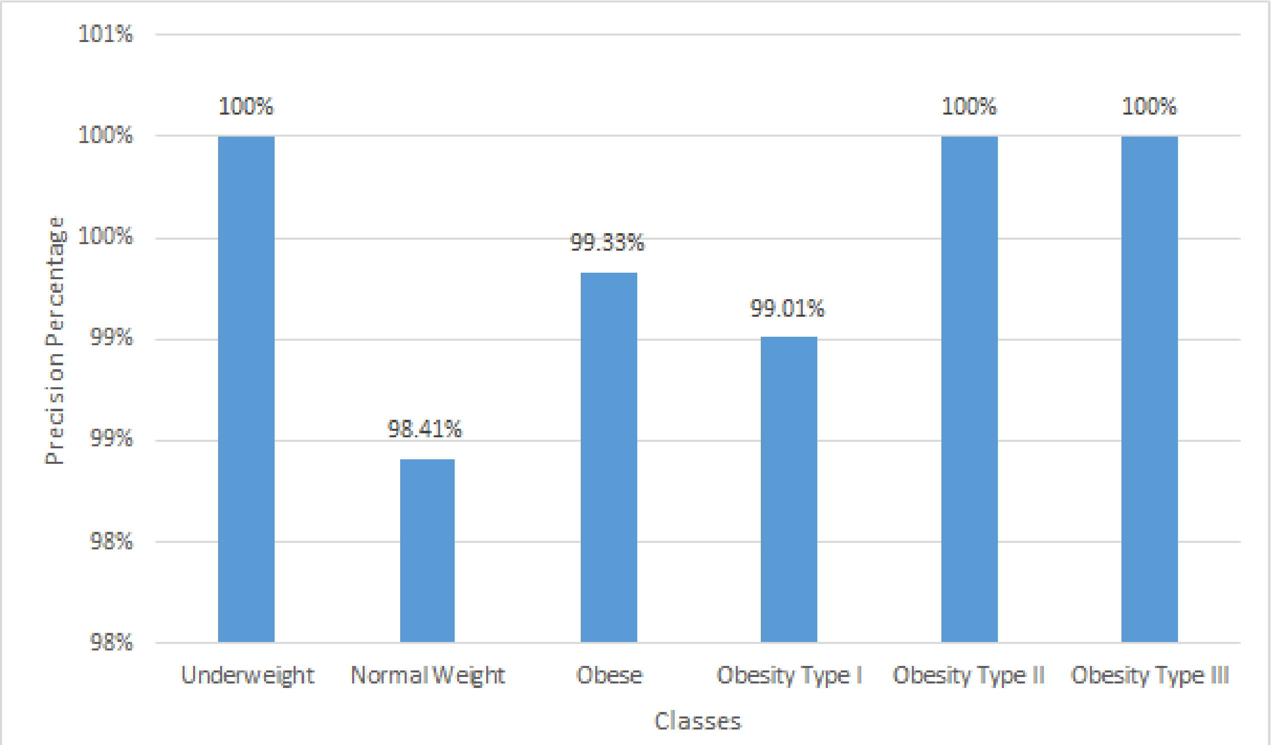


Figure 10: Class Wise Precision of the Hybrid Model

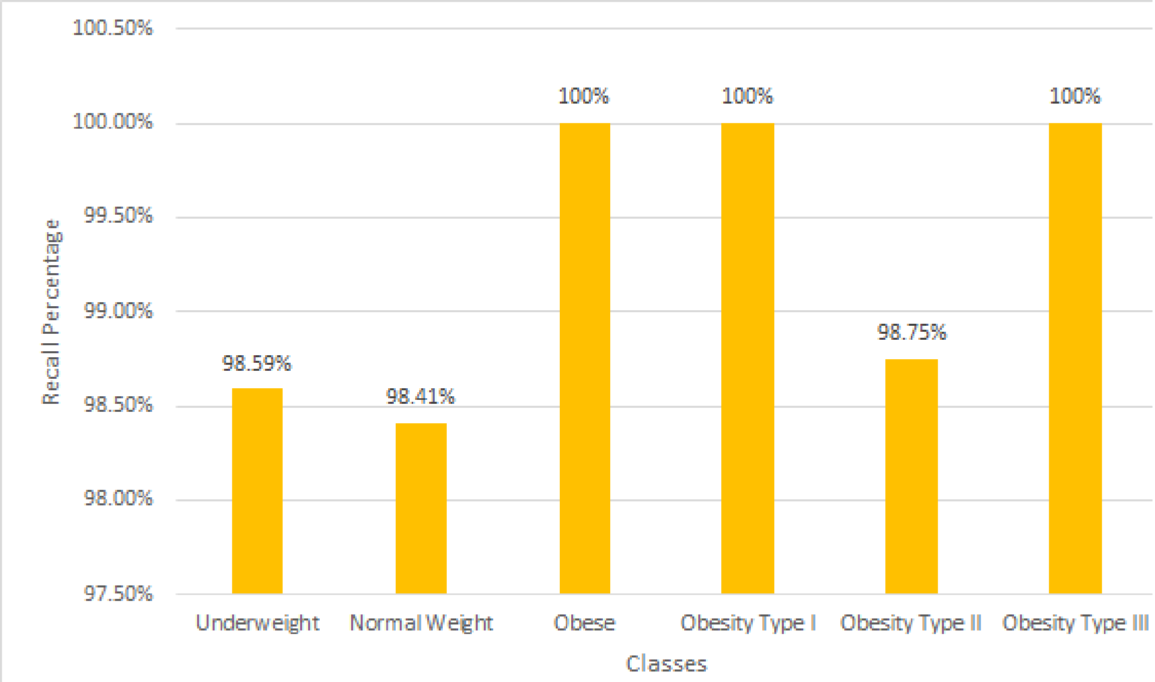


Figure 11: Class Wise Recall of the Hybrid Model

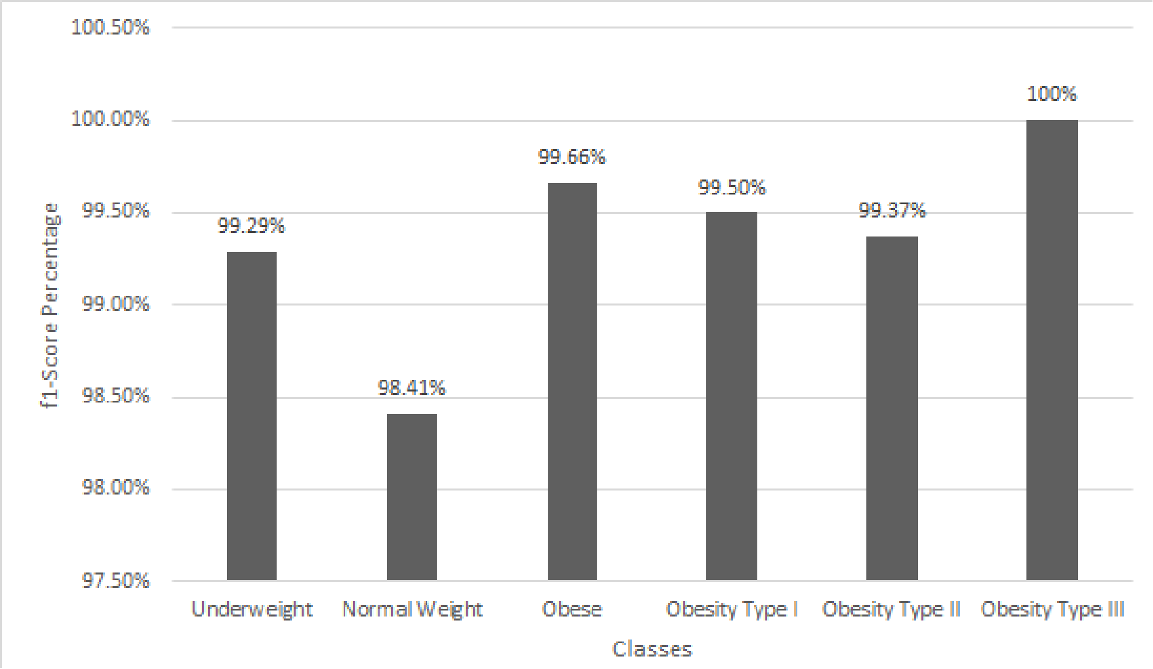


Figure 12: Class Wise F1-score of the Hybrid Model

4 Discussion

This section will focus on the assessment and analysis of the results acquired, the methodology selected and the validity and reliability of the experiments. This section would be divided in two parts. In Section 4.1, we will compare our results to those of the current state of the art research methodology proposed and try to provide a newer horizon of further research in the domain of obesity detection and classification. Section 4.2 will reflect on the taken approach towards the research tasks, point out its benefits and flaws as well as address whether the right method was chosen to solve the given problem. Section 4.3 would elaborate on the performance of the model of providing external stress (in this case, Covid-19).

4.1 Comparison to Previous Results

We had previously mentioned that the Decision Tree model proposed [53] is our baseline research model. Another similar work with better results were proposed by [30]. Thus, we will try to compare our results to the results reported by the authors. There was no way to validate or verify their results and so we are assuming that the values reported were true values. Table 10 shows the comparison of the results.

Model Name	Precision	Recall
Decision Tree (J48) [53]	97.4 %	97.8 %
Decision Tree + Simple K-Means [30]	98.5 %	98.5%
Hybrid Current Model	**99.4 %**	**99.4 %**

Table 11: Precision and Recall for the previous works of research on the same dataset are shown

We notice that the proposed model outperforms each of the previous models in both performance metrics.

The proposed MLP + XGBoost model outperforms all previous methods on the two metrics. The previous state of art on the Precision metric is the Decision Tree + Simple K-Means [30] with a score of 98.5%. The current model has a precision score of 99.4%, which is a sizeable improvement. Similarly, the previous state of art on the Recall metric is the Decision Tree + Simple K-Means [30] with a score of 98.5%. Our model outperforms this metric as well with a score of 99.4%. Interestingly, our true positive rate is also much higher and false positive rates are also consequently lower than that obtained in the previous models. This can also be visualized through the figures shown below.

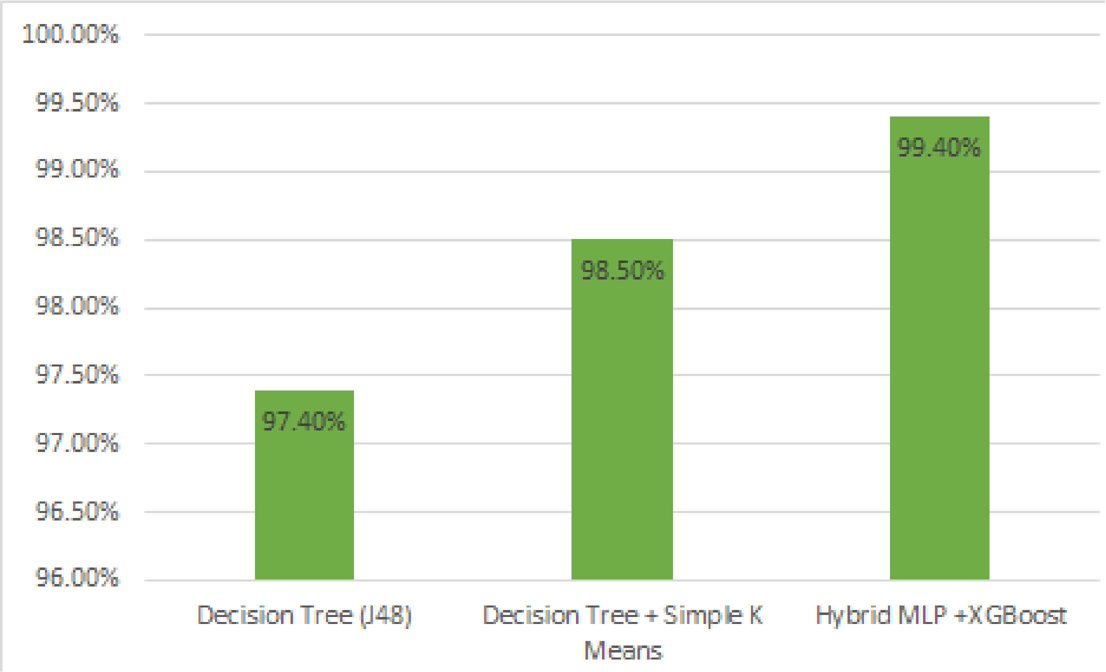


Figure 13: Comparison of precision amongst the champion- challenger models

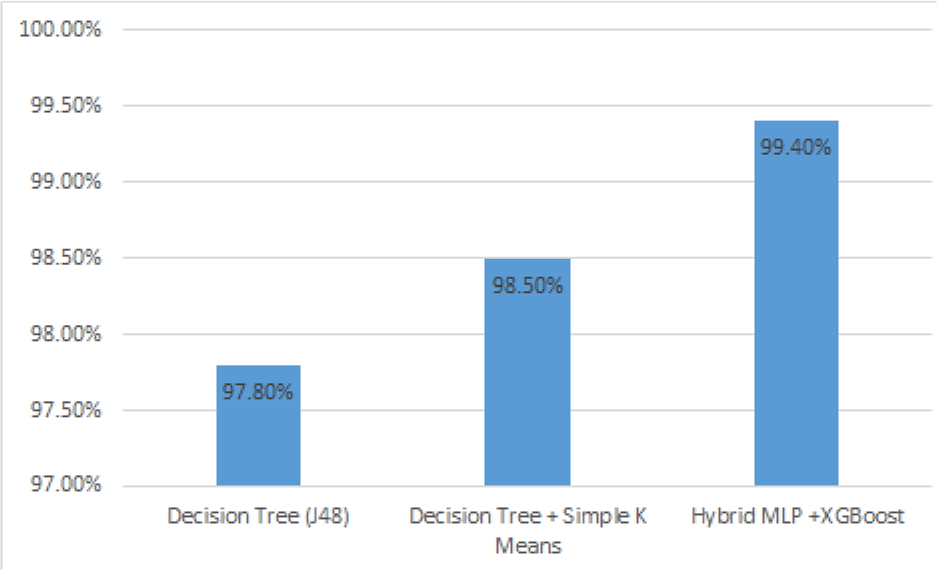


Figure 14: Comparison of recall amongst the champion- challenger models

So, this work reasonably outlines the usage of hybrid machine learning models in this domain of obesity detection in human. Also, the current model also significantly urges the researchers and practitioners in the domain to explore more into the currently newer performance enhanced models available in the industry. Also, this work also shows how important data pre-processing steps are. The whole process of frequent feature extraction and feature scaling were pivotal in improving performance.

4.2 Method Reflection

The process utilised by KDD showed enough suitability to resolve both concerns of research. The standardised technique enabled the flexibility to explore various experimental settings without overlooking the initial target. The suggested method was able to generate a work dataset in adequate amount and quality for subsequent application on various algorithms, starting with data extraction and pre-processing. The adoption of various assessment measurements or performance metrics permits the combination of separate attribute strengths and so creates the basis for computing reliable results.

The use of machine translations as pseudo references (encoding categorical data) proved to be suitable, in order to meet the requirements for certain scoring metrics for multi-class classification. The usage of pseudo marks also allowed the study questions to be correctly answered. The preparation of the data set used and their attributes was the outlier removal, the discovery of strong correlation characteristics and many standardisation of attribute values produced at various stages leading to a well-established classification data base.

To address possible multi-class classification characteristics, various machine learning algorithms were used. Amongst all the combinations, we discovered that the Multilayer Perceptron + XGBoost gave the highest performance. However, it must be remembered that this performance was only with respect to the dataset provided which captures the obesity dynamics of a particular region for a particular age group. Thus, due to the lack of generalization in the given dataset and lack of publicly available data sources on this domain, it was not possible to provide a model that will predict correctly for all general cases.

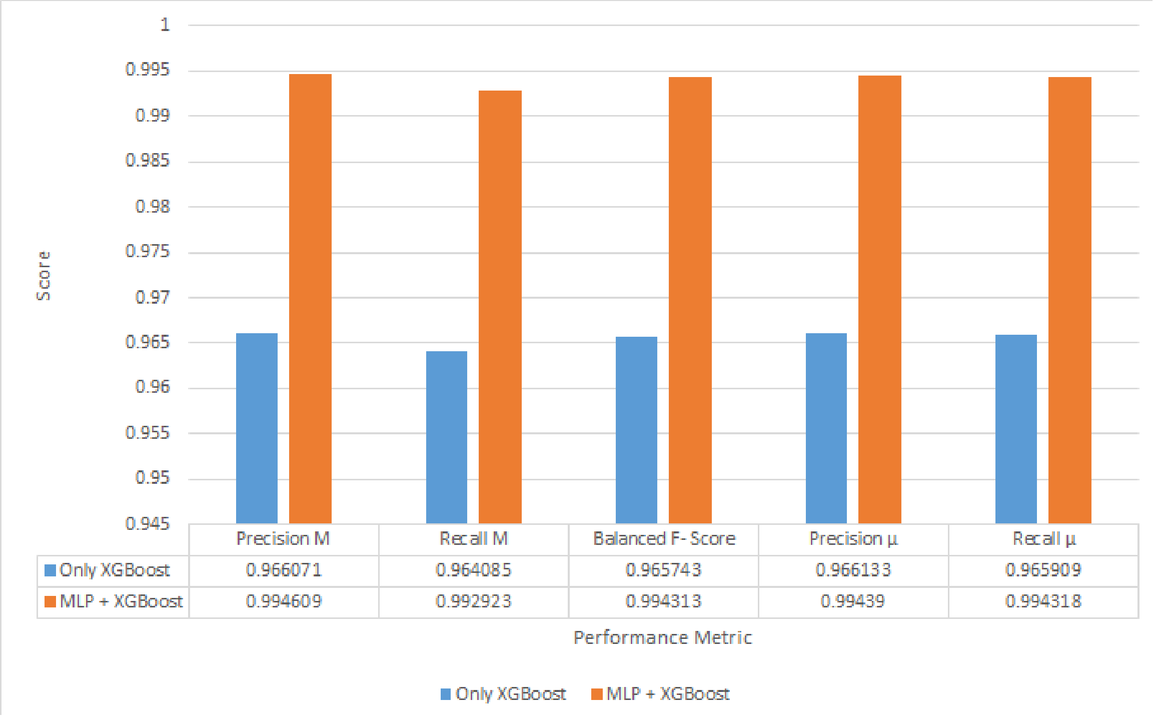
Using different varieties of machine learning methods proved to be very effective in improving the performance of the model. On running a diagnostic, we observed that the models, in a stand-alone condition, performed well. But their performance rose rapidly, when included in the pipeline, one after the other. The results are given below.

In the experiment, we tried to test for the independent prediction power of the XGBoost model without the MLP Model. However, we have used all the dimensionality reduction methods and feature standardization methods as previously mentioned in the pipeline.

By the results, we can clearly see that the hybrid model, despite its shortcomings of generalization, as elaborated earlier, performs to be the best model for this problem.

Performance Metric	Only XGBoost	MLP + XGBoost
Accuracy	96. 5909 %	99. 4318 %
Precision $_{\mu}$	0.966133	0. 994349
Recall $_{\mu}$	0. 965909	0. 994318
Precision $_{M}$	0.966071	0.994609
Recall $_{M}$	0.964085	0.992923
Balanced F- Score	0.965743	0. 994313

Table 12: Performance comparison between hybrid and normal model



	Precision M	Recall M	Balanced F- Score	Precision µ	Recall µ
■ Only XGBoost	0.966071	0.964085	0.965743	0.966133	0.965909
■ MLP + XGBoost	0.994609	0.992923	0.994313	0.99439	0.994318

Figure 15: Comparative Study of all Performance Metrics of the Hybrid vs Normal Model

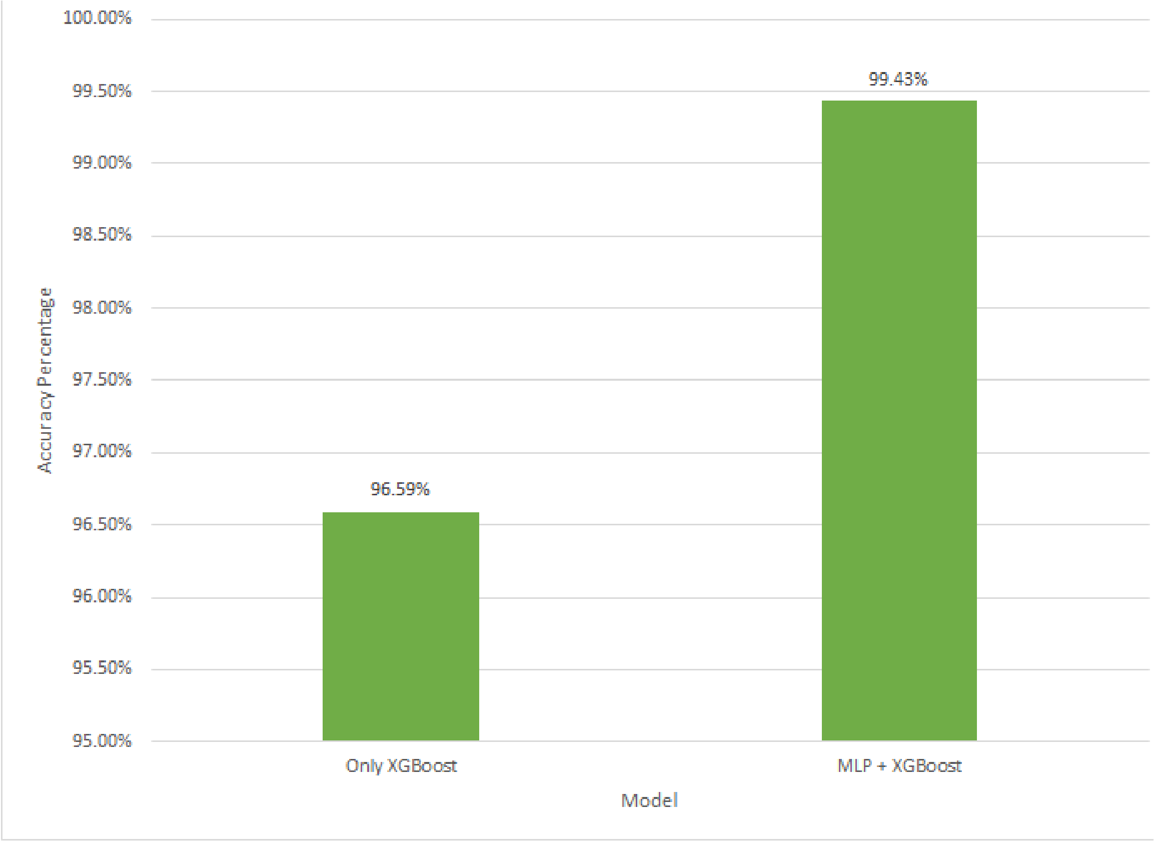


Figure 16: Accuracy Comparison between hybrid and Normal Model

Class	Accuracy	Precision	Recall	f1- Score	Support
Underweight	99.05%	94.44%	98.55%	96.45%	69
Normal Weight	98.29%	97.26%	91.02%	94.03%	78
Obese	98.86%	96.06%	99.18%	97.60%	123
Obesity Type I	98.67%	96.87%	95.87%	96.37%	97
Obesity Type II	98.67%	96.25%	95.06%	95.65%	81
Obesity Type III	99.62%	98.75%	98.75%	98.75%	80

Table 13: Class Wise Performance metrics of the XGBoost model

5 Performance of the Model in Covid-19

5.1 Introduction

The new coronavirus illness, later shortened as COVID-19 by the World Health Organization (WHO) [57], is a respiratory viral influenza that can spread rapidly by water-forming droplets and diverse inclusions that may produce droplets when speech, breathing, cough or sneezing [58]. It initially appeared in December 2019 in the city of Wuhan, Hubei province, China [59]. Soon after, on 30 January [60], the COVID-19 emergency was an international public health emergency, and on 11 March COVID-19 was considered to be pandemic [61].

As it is unlikely that final COVID-19 vaccines and treatments will be developed soon [62], the most efficient COVID-19 epidemic management method appears to be social isolation [63]. This technique is to flatten the curve of new infections caused by transmission from person to person, reduce morbidity, mortality and the subsequent rise in demand in the health system. As a basic strategy to restrict human exposure to virus the public has therefore also been recommended to reduce movement and stay at home. Health authorities, including the National Health Commission of the People’s Republic of China [64], WHO [65], and U.S. Centers for Disease Control and Prevention (CDC) [66], have issued safety recommendations for taking related precautions to reduce exposure to and transmission of the virus.

The extended life at home might unintentionally lead to sedentary behaviors, such as spending excessive time seating, recline, or lying for screen activities (playing games, watching television, using mobile devices), decreasing regular physical activity (reducing the energy spend). Critically addressed were some of the main unfavorable outcomes of prolonged residence including physical inactivity, weight gain, disorders of behavioral dependence, insufficient exposure to sunlight, and social insulation. In particular, weight gain during adulthood was associated with a significantly increased risk of major chronic diseases and decreased odds of healthy aging [68].

5.2 Methodology and Modifications to Data

According to Lin et al. study [67] on a total of 7444 weight measurements from 269 unique study participants (residing in 37 states and Washington, District of Columbia, USA), post shelter in place participants experienced steady weight gain of 0.27Kg every 10 days irrespective of geographic location or comorbidities. Dicker et al. [69] asserted that obesity is a risk factor for Covid-19 and highly obese adults are more prone to

Covid-19. They elaborate the illness severity by the figure given below. They have also stated that adults with obesity have longer viral load and lower vaccination effect.

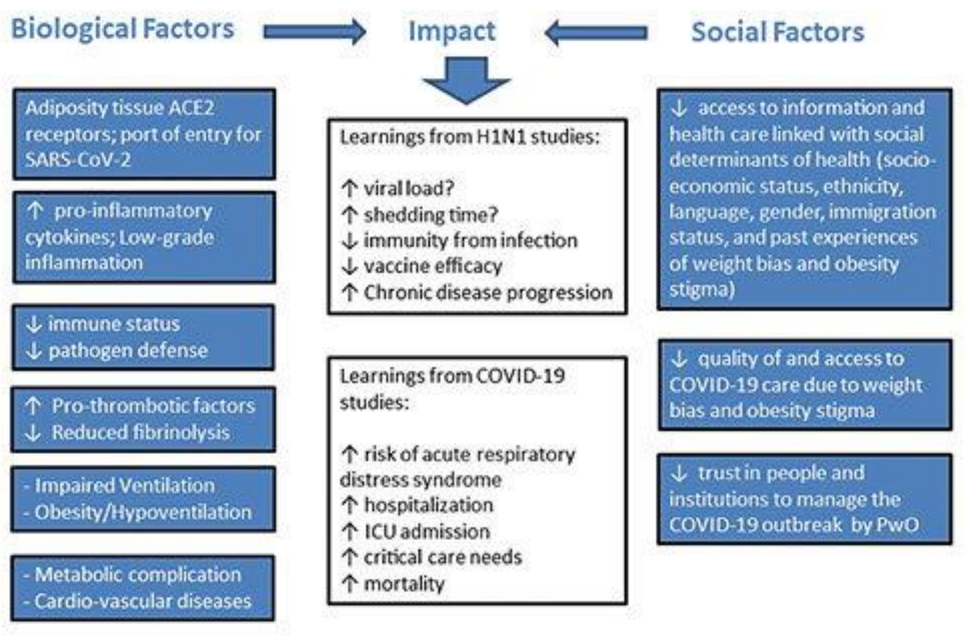

Biological Factors
Impact
Social Factors
Adiposity tissue ACE2 receptors; port of entry for SARS-CoV-2
↑ pro-inflammatory cytokines; Low-grade inflammation
↓ immune status
↓ pathogen defense
↑ Pro-thrombotic factors
↓ Reduced fibrinolysis
- Impaired Ventilation
- Obesity/Hypoventilation
- Metabolic complication
- Cardio-vascular diseases
Learnings from H1N1 studies:
↑ viral load?
↑ shedding time?
↓ immunity from infection
↓ vaccine efficacy
↑ Chronic disease progression
Learnings from COVID-19 studies:
↑ risk of acute respiratory distress syndrome
↑ hospitalization
↑ ICU admission
↑ critical care needs
↑ mortality
↓ access to information and health care linked with social determinants of health (socio-economic status, ethnicity, language, gender, immigration status, and past experiences of weight bias and obesity stigma)
↓ quality of and access to COVID-19 care due to weight bias and obesity stigma
↓ trust in people and institutions to manage the COVID-19 outbreak by PwO

Figure 17: Biological and social factors linking obesity with COVID-19 illness severity. ACE, angiotensin-converting enzyme; ICUs, intensive care units; PwO, people with obesity.

Source: https://www.karger.com/WebMaterial/ShowPic/1210498

Thus, we decided to stress-test our hybrid model architecture on the onset of Covid-19. As we already know, the given data was taken much before Covid-19. So, we decided to use the findings of [67] and assumed the period of Covid-19 to be on peak and lockdowns in various countries to be actively during the time Covid-19 peaked in Latin America. On 20th March, 2020, the Columbian President had declared an initial 19-day lockdown from 24th March, 2020, which was extended up to 27th April, 2020, being again extended to 31st May, 2020, finally going up to 1st September, 2020. Lockdown was declared in Peru from 16th March, 2020 to 8th September, 2020. Venezuela also went into lockdown from 17th March, 2020 up to July, 2020.

Thus, we are considering the period of lockdown due to Covid-19 to be held from 20th March, 2020 up till 1st September, 2020 and using the value of average weight gain mentioned in [67] by initializing a random multiplier between 0-0.27Kg of every 10 days for the given period. The table below elaborates the study.

Parameter	Value/ Usage
Start Date	20/3/2020
End Date	1/9/2020
Total Duration in days	165
Random Multiplier Value	Increase of [0, 0.27] Kg weight for every 10 days.

Table 14: Details of the Covid-19 stress testing study

The resultant weight histogram is shown as below and compared to the previous value in Figure 18 and Figure 19. In fact, the general trend of weights has also changed significantly.

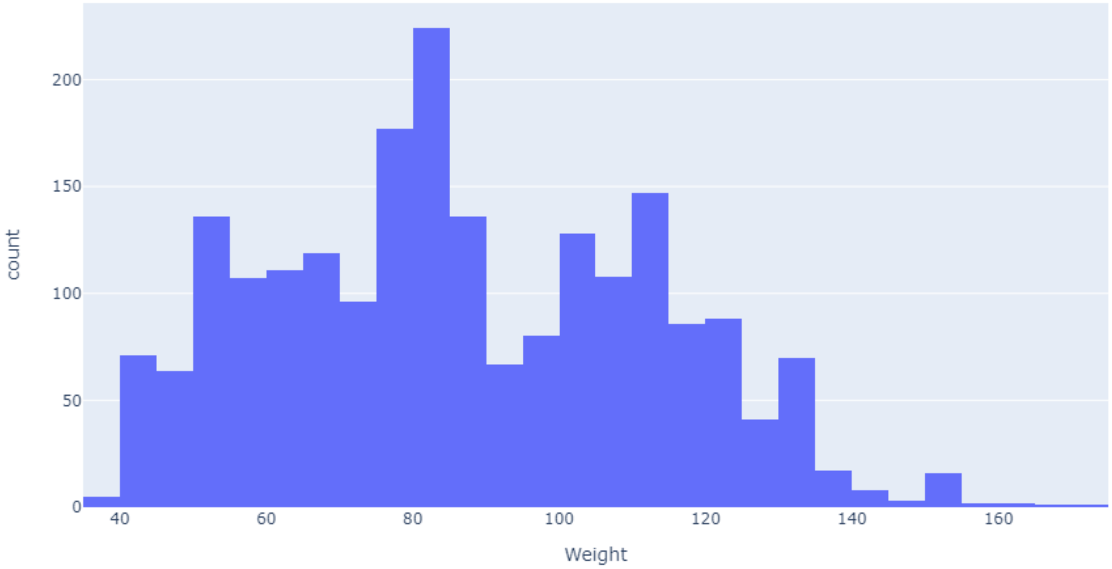


Figure 18: Histogram of Weights before Covid-19

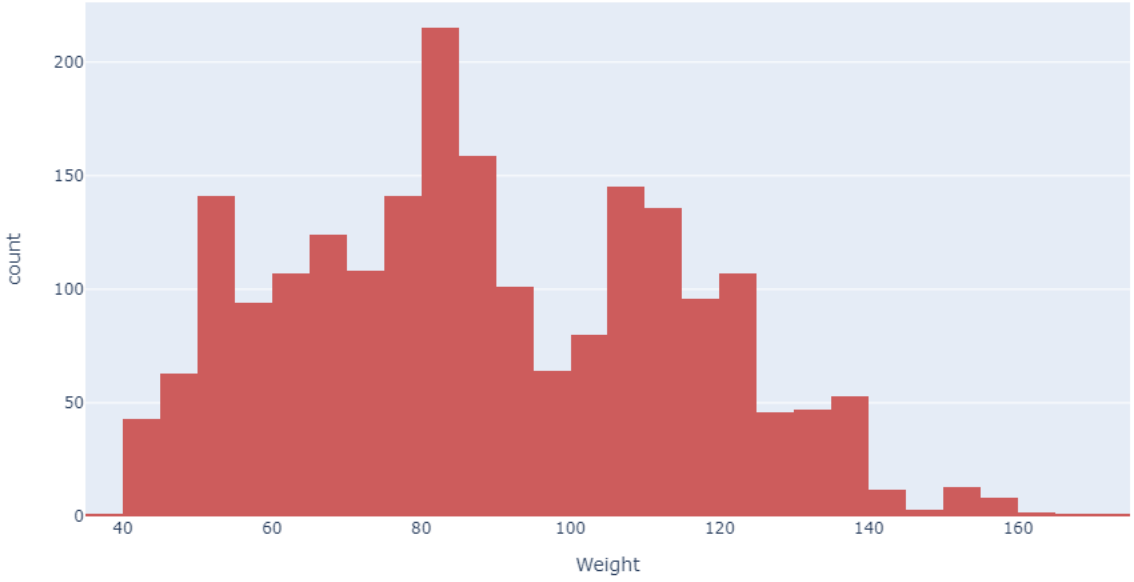


Figure 19: Histogram of Simulated Weights after Covid-19

5.3 Results

The model Performance Metrics evaluated are given by the following set of tables.

		Predicted Classes					
		Underweight	Normal Weight	Obese	Obesity Type I	Obesity Type II	Obesity Type III
Actual Classes	Underweight	34	3	0	0	0	0
	Normal Weight	0	88	1	0	0	0
	Obese	0	0	126	1	0	0
	Obesity Type I	0	0	0	96	0	0
	Obesity Type II	0	0	0	0	87	0
	Obesity Type III	0	0	0	0	0	92

Table 15: Confusion Matrix for 6-class classification by the hybrid model using the Covid-19 updated data

The overall Performance metrics are also updated.

Performance Measure	Value
Accuracy	99. 053030 %
Precision $_{\mu}$	0.990674
Recall $_{\mu}$	0.990530
Precision $_{M}$	0.991474
Recall $_{M}$	0.983301
Balanced F- Score	0.990457

Table 16: Performance Measures for the 6-class classification Problem using Hybrid Model in Covid-19 simulated Dataset

Class	Accuracy	Precision	Recall	f1- Score	Support
Underweight	99.43%	100%	91.89%	95.77%	37
Normal Weight	99.24%	96.70%	98.87%	97.77%	89
Obese	99.62%	99.21%	99.21%	99.21%	127
Obesity Type I	99.81%	98.96%	100%	99.48%	96
Obesity Type II	100%	100%	100%	100%	87
Obesity Type III	100%	100%	100%	100%	92

Table 17: Class Wise Performance metrics of the hybrid model

Thus, we can find that that the accuracy is predictions by our model is being maintained although our data is being modified. This means that our model is stable and is correctly being able to make accurate predictions about obesity of a person based on input characteristics.

This, by all means was a helpful exercise in validating our model and attest to the claim that it can be used for future specialized situations as well.

The relevant Python implementation is given in the Appendix.

6 Conclusion

6.1 Conclusions

This work answered the following questions:

- How robust is our model?
- How well does it perform in case of changes in trends?
- How well does our model generalize and detect obesity?
- How much time is needed for computation?

This was done by using a knowledge discovery process consisting of the phases, gathering data, pre-processing data, choosing an appropriate data mining approach to find patterns among the data and interpreting them. Finally, the results were used for further research. We tested for the need of the hybrid model by comparing its results to that of a stand-alone model. We also tested the performance of our model on the context of Covid-19.

In our model pipeline, we utilized a plethora of dimensionality reduction and feature scaling methods, which were very important in maximizing the performance of predictions of our said model. Also, we employed adequate amount of time and effort in performing hyper parameter tuning for all the internal constituents of this overall model structure.

We employed the usage of two widely popular prediction supervised learning methods, namely the Multilayer Perceptron Model (also known as Artificial Neural Network) and the tree-based ensemble technique XGBoost. Also, we employed judicious usage of Extremely Randomized Trees Classifier as the tree-based dimensionality reduction method, both in Recursive Feature Elimination and SelectFromModel.

Our constructed model gave a prediction accuracy of 99.43%, a precision value of 0.994 and a recall value of 0.994, which, by all means, were sizeable improvements from the performance metrics generated from previous works of research working with the same dataset. We also tried to estimate the values of the different performance metrics by creating a simulated dataset keeping the onset of Covid-19 in mind and used the same model parameters and pipelines to validate the power of predictions of the model. Unsurprisingly, we did not find any major discrepancy in this experiment, with the model architecture reporting accuracy of 99.05%, precision score of 0.990 and a recall score of 0.990.

Thus, our model is valid and very flexible to changes and this work has been able to create a newer Machine Learning Method to predict obesity in human beings.

6.2 Further Research

This work presents a hybrid Machine Learning Model for Human Obesity Detection using various underlying statistical and machine learning concepts. Extending the work on this topic, an attempt to further improve classification quality on this dataset could be made by aggregating and fine tuning more Machine Learning models into the model pipeline. This approach would result in a more fine-grained obesity classification as with the passage of time, we are making bigger breakthroughs in Machine Learning research and algorithms.

Some of the used machine learning approaches could not be optimized as extensively as it would be desirable due to computation resource limitations. Therefore, especially concerning Multilayer Perceptron (Artificial Neural Network), a more in-depth optimization process would be desirable to come closer to a global optimum in terms of classification results. In addition to that, efforts can be made to present more publicly available datasets on obesity, which would be homogenous in terms of the features and descriptions. This would harbinger a newer direction of research in trying to build a model that will be able to identify the global trend as a whole, and make accurate predictions.

Concerning the suggested evaluation framework, an in-depth analysis of the proposed classes is necessary to verify and optimize them, since the recommended method has only been validated using sample-based testing. In order to put the proposed approach into context with different approaches on machine learning methods for evaluation of machine translation, a second database could be created using domain independent data to compare the results on a non-specific data set with the results of a domain focused one. This would allow drawing conclusions concerning the implicit additional knowledge that is gained by focusing on the domain of obesity detection in humans.

In conclusion, the presented work extends the related research on this topic by combining multiple machine translation evaluation metrics with the use of machine learning methods focusing on the obesity dataset.

7 References

1. Haslam, D. (2007), Obesity: a medical history. *Obesity Reviews*, 8: 31-36. *https://doi.org/10.1111/j.1467-789X.2007.00314.x*
2. Hippocrates 400bc *De PriscinaMedicina*.
3. Siculus 90bc–20bc *Bibliotheca historica*.
4. Herodotus ~440bc. Euterpe, section 77.
5. Laertius, Diogenes 3rd century Life of Pythagoras. segm 9. Menag (ed.)
6. Hippocrates ~400bc *De Flatibus*.
7. Cheyne G. An Essay of Health and Long Life. George Strahan, London: London, 1724.
8. Morgagni JB. Epistola Anatoma Clinica XXI. De Sedibuset Causis Morborum per Anatomenindagata. Translated from the Latin by Benjamin Alexander, M.D., with a Preface, Introduction and a new Translation of five letters by Paul Klemperer. Published under the auspices of the Library of the New York Academy of Medicine by Harner Publishing Co., NY, 1960. 1765.
9. Banting W. Letter on Corpulence, Addressed to the Public. Harrison: London, 1864.
10. Buchan N. Domestic Medicine (American Edition). Dobson: Philadelphia, PA, 1795.
11. Diabetes in the Ebers papyrus. http://www-unix.oit.umass.edu/~abhu000/diabetes/ebers.html (accessed July 2006).
12. Ward ZJ, Long MW, Resch SC, Gortmaker SL, Cradock AL, Giles C, et al. Redrawing the US obesity landscape: bias-corrected estimates of state-specific adult obesity prevalence. *PloS One* 2016;11(3):e0150735. https://doi.org/10.1371/journal.pone.0150735
13. De la Hoz Manotas, Alexis & De la Hoz Correa, Eduardo & Mendoza, Fabio & Morales, Roberto & Sanchez, Beatriz. (2019). Obesity Level Estimation Software based on Decision Trees. *Journal of Computer Science*. 15. 10. 10.3844/jcssp.2019.67.77.
14. J.E. Cecil, R. Tavendale, P. Watt, M.M. Hetheringnon, C.N.A. Palmer an obesity-associated FTO gene variant and increased energy intake in children N Engl J Med, 359 (2008), pp. 2558-2566.
15. Dixon JB, Dixon ME, O'Brien PE. Depression in Association with Severe Obesity: Changes With Weight Loss. Arch Intern Med. 2003;163(17):2058–2065. doi:10.1001/archinte.163.17.2058.
16. Lin AL, Vittinghoff E, Olgin JE, Pletcher MJ, Marcus GM. Body Weight Changes During Pandemic-Related Shelter-in-Place in a Longitudinal Cohort Study. *JAMA Netw Open*. 2021;4(3):e212536. doi:10.1001/jamanetworkopen.2021.2536
17. Davila-Payan C, DeGuzman M, Johnson K, Serban N, Swann J. Estimating prevalence of overweight or obese children and adolescents in small geographic areas using publicly available data. Prev Chronic Dis 2015;12:140229. https://doi.org/10.5888/pcd12.140229.

18. Manna, S., & Jewkes, A. M. “Understanding early childhood obesity risks: an empirical study using fuzzy signatures”, In Fuzzy systems (FUZZ-IEEE). *2014 IEEE international conference on (pp. 1333-1339). IEEE*.
19. Adnan MHBM, Husain W. A hybrid approach using Naïve Bayes and Genetic Algorithm for childhood obesity prediction”. In: Computer & information science (ICCIS), 2012 international conference on, vol. 1. *IEEE*; 2012. p. 281–5.
20. Adnan MHM, Husain W. A framework for childhood obesity classifications and predictions using NBtree”. In: Information technology in Asia (CITA 11), 2011 7th international conference on. *IEEE*; 2011. p. 1–6.
21. Adnan, M. H. B. M., Husain, W., &Damanhoori, F. “A survey on utilization of data mining for childhood obesity prediction”, In Information and telecommunication technologies (APSITT). 2010 8th Asia-pacific symposium on (pp. 1-6). *IEEE*.
22. Dugan TM, Mukhopadhyay S, Carroll A, Downs S. Machine learning techniques forprediction of early childhood obesity. ApplClinInf 2015;6(3):506–20.
23. Zhang ML, Zhou ZH. Multi-instance clustering with applications to multi-instance prediction. ApplIntell 2009;31(1):47–68.
24. Suguna M. Childhood obesity epidemic analysis using classification algorithms. Int. J. Mod. Comput. Sci 2016;4(1):22–6.
25. Abdullah FS, Manan NSA, Ahmad A, Wafa SW, Shahril MR, Zulaily N, Ahmed A. Data mining techniques for classification of childhood obesity among year 6 school children. In: International conference on Soft Computing and Data Mining. Cham: *Springer*; 2016. p. 465–74.
26. De-La-Hoz-Correa Eduardo, Mendoza-Palechor Fabio E, De-La-Hoz-Manotas Alexis,Morales-Ortega Roberto C, S´anchezHern´andez Beatriz Adriana. Obesity level estimation software based on decision trees. *J ComputSci* 2019;15(Issue 1):67–77. https://doi.org/10.3844/jcssp.2019.67.77.
27. Ward ZJ, Long MW, Resch SC, Gortmaker SL, Cradock AL, Giles C, et al. Redrawing the US obesity landscape: bias-corrected estimates of state-specific adult obesity prevalence. *PloS One* 2016;11(3):e0150735. https://doi.org/10.1371/journal.pone.0150735.
28. G´omez M, ´Avila L. La obesidad: un factor de riesgocardiometab´olico. In: Medicina de Familia, vol. 8; 2008. p. 91–7. Nº. 2.
29. Joachims T. Text categorization with support vector machines. Proceedings of the European C~njerence on machine learning. *Springer-Verlrtg*; 1998.
30. Cervantes, Rodolfo & Palacio, Ubaldo. (2020). Estimation of obesity levels based on computational intelligence. Informatics in Medicine Unlocked. 21. 100472. 10.1016/j.imu.2020.100472.
31. Geurts, P., Ernst, D. &Wehenkel, L. Extremely randomized trees. Mach Learn 63, 3–42 (2006). https://doi.org/10.1007/s10994-006-6226-1.
32. Frawley, W. J., Piatetsky-Shapiro, G., &Matheus, C. J. (1992). Knowledge Discovery in Databases: An Overview. AI Magazine, 13(3), 57. https://doi.org/10.1609/aimag.v13i3.1011
33. U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, “From data mining to knowledge discovery in databases,” Ai Magazine, vol. 17, pp. 37–54, 1996.

34. G. James et al., An Introduction to Statistical Learning, *Springer Texts in Statistics*, https://doi.org/10.1007/978-1-0716-1418-1_2
35. F. Camastra and A. Vinciarelli, Machine Learning for Audio, Image and Video Analysis - Theory and Applications, Second Edition. Advanced Information and Knowledge Processing, *Springer*, 2015.
36. Rosenblatt, “The Perceptron: A Theory of Statistical Separability in Cognitive Systems”, Cornell Aeronautical Laboratory, Report No. VG1196-G-1, January, 1958.
37. Haykin, Simon S. Neural Networks: A Comprehensive Foundation. Upper Saddle River, N.J.: *Prentice Hall*, 1999.
38. Rumelhart, D., Hinton, G. & Williams, R. Learning representations by back-propagating errors. *Nature* 323, 533–536 (1986). https://doi.org/10.1038/323533a0
39. Michael Luckert, Mortiz Schaefer-Kehnert, WelfLöwe, Morgan Ericsson, Anna Wingkvist: A Classifier to Determine Whether a Document is Professionally or Machine Translated. BIR 2016: 339-353.
40. A.H. Kramer and Alberto L. Sangiovanni-Vincentelli: Optimization Techniques for Neural Networks, EECS Department University of California, Berkeley Technical Report No. UCB/ERL M89/1 January 1989. http://www2.eecs.berkeley.edu/Pubs/TechRpts/1989/ERL-89-1.pdf
41. Chen, T., &Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794). New York, NY, USA: *ACM*. https://doi.org/10.1145/2939672.2939785
42. Vishal Morde and VenkatAnurag Shetty (2019). XGBoost Algorithm: Long May She Reign! https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d
43. Sokolova, Marina, and Guy Lapalme. "A systematic analysis of performance measures for classification tasks." Information Processing & Management, Vol. 45, no. 4 (2009): 427-437.
44. Van Der Maaten, Laurens, Eric Postma, and Jaap Van den Herik. "Dimensionality reduction: a comparative." J Mach Learn Res 10.66-71 (2009): 13.
45. Palechor, F. M., & de la HozManotas, A. (2019). Dataset for estimation of obesity levels based on eating habits and physical condition in individuals from Colombia, Peru and Mexico. Data in Brief, 104344.
46. NORMA Oficial Mexicana NOM-008-SSA3-2010, Para el tratamiento integral delsobrepeso y la obesidad. DiarioOficial (2010).
47. N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer. SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res., 16 (2002), pp. 321-357
48. M.V. Olmedo. La obesidad: unproblema de saludpública. Revista de divulgacióciéntífica y tecnológica de la Universidad Veracruzana (2011).
49. C. Davila-Payan, M. DeGuzman, K. Johnson, N. Serban, J. Swann. Estimating prevalence of overweight or obese children and adolescents in small geographic areas using publicly available data. Prev. Chronic Dis., 12 (2015).

50. S. Manna, A.M. Jewkes. Understanding early childhood obesity risks: an empirical study using fuzzy signatures. Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on, *IEEE* (2014, July).
51. M.H.B.M. Adnan, W. Husain. A hybrid approach using Naïve Bayes and Genetic Algorithm for childhood obesity prediction. Computer & Information Science (ICCIS), 2012 International Conference on, vol. 1, *IEEE* (2012, June).
52. T.M. Dugan, S. Mukhopadhyay, A. Carroll, S. Downs. Machine learning techniques for prediction of early childhood obesity. Appl. Clin. Inf., 6 (3) (2015).
53. Eduardo De-La-Hoz-Correa, Fabio E. Mendoza-Palechor, Alexis De-La-Hoz-Manotas, Roberto C. Morales-Ortega, Beatriz Adriana Sánchez Hernández. Obesity level estimation software based on decision Trees. J. Comput. Sci., 15 (Issue 1) (2019).
54. Trang T. Le, Weixuan Fu and Jason H. Moore (2020). Scaling tree-based automated machine learning to biomedical big data with a feature set selector. Bioinformatics.36(1): 250-256.
55. Randal S. Olson, Ryan J. Urbanowicz, Peter C. Andrews, Nicole A. Lavender, La Creis Kidd, and Jason H. Moore (2016). Automating biomedical data science through tree-based pipeline optimization. Applications of Evolutionary Computation, pages 123-137.
56. Randal S. Olson, Nathan Bartley, Ryan J. Urbanowicz, and Jason H. Moore (2016). Evaluation of a Tree-based Pipeline Optimization Tool for Automating Data Science. *Proceedings of GECCO* 2016, pages 485-492.
57. John Hopkins University COVID-19 Dashboard by the Center for Systems Science and Engineering (CSSE) at John Hopkins University (JHU) [(accessed on 8 December 2020)];2020 Available online: https://coronavirus.jhu.edu/map.html.
58. Di Renzo L., Gualtieri P., Pivari F., Soldati L., Attinà A., Cinelli G., Leggeri C., Caparello G., Barrea L., Scerbo F., et al. Eating habits and lifestyle changes during COVID-19 lockdown: An Italian survey. J. Transl. Med. 2020;18:229. doi: 10.1186/s12967-020-02399-5.
59. Rodríguez-Pérez C., Molina-Montes E., Verardo V., Artacho R., García-Villanova B., Guerra-Hernández E.J., Ruíz-López M.D. Changes in Dietary Behaviours during the COVID-19 Outbreak Confinement in the Spanish COVIDiet Study. Nutrients. 2020;12:1730. doi: 10.3390/nu12061730.
60. Reyes-Olavarría D., Latorre-Román P., Guzmán-Guzmán I.P., Jerez-Mayorga D., Caamaño-Navarrete F., Delgado-Floody P. Positive and Negative Changes in Food Habits, Physical Activity Patterns, and Weight Status during COVID-19 Confinement: Associated Factors in the Chilean Population. Int. J. Environ. *Res. Public Health*. 2020;17:5431. doi: 10.3390/ijerph17155431.
61. Keel P.K., Gomez M.M., Harris L., Kennedy G.A., Ribeiro J., Joiner T.E. Gaining "The Quarantine 15": Perceived versus observed weight changes in college students in the wake of COVID-19. Int. J. Eat. Dis. 2020;53:1801–1808. doi: 10.1002/eat.23375.
62. Bhutani S., Cooper J.A. COVID-19–Related Home Confinement in Adults: Weight Gain Risks and Opportunities. Obesity. 2020;28:1576–1577. doi: 10.1002/oby.22904.

63. Nakeshbandi M., Maini R., Daniel P., Rosengarten S., Parmar P., Wilson C., Kim J.M., Oommen A., Mecklenburg M., Salvani J., et al. The impact of obesity on COVID-19 complications: A retrospective cohort study. Int. J. Obes. 2020; 44:1832–1837. doi: 10.1038/s41366-020-0648-x.
64. Gao F., Zheng K.I., Wang X.-B., Sun Q.F., Pan K.H., Wang T.Y., Chen Y.P., Targher G., Byrne C.D., George J., et al. Obesity is a risk factor for greater COVID-19 severity. Diabetes Care. 2020;43: e72–e74. doi: 10.2337/dc20-0682.
65. Földi M., Farkas N., Kiss S., Zádori N., Váncsa S., Szakó L., Dembrovszky F., Solymár M., Bartalis E., Szakács Z., et al. Obesity is a risk factor for developing critical condition in COVID-19 patients: A systematic review and meta-analysis. Obes. Rev. 2020;21: e13095. doi: 10.1111/obr.13095.
66. Malik P., Patel U., Patel K., Martin M., Shah C., Mehta D., Malik F.A., Sharma A. Obesity a predictor of outcomes of COVID-19 hospitalized patients-A systematic review and meta-analysis. J. Med. Virol. 2020; 93:1188–1193. doi: 10.1002/jmv.26555.
67. Lin AL, Vittinghoff E, Olgin JE, Pletcher MJ, Marcus GM. Body Weight Changes During Pandemic-Related Shelter-in-Place in a Longitudinal Cohort Study. *JAMA Netw Open*. 2021;4(3):e212536. doi:10.1001/jamanetworkopen.2021.2536.
68. Changzheng, Y.; Manson, J.E.; Yuan, C.; Liang, M.H.; Grodstein, F.; Stampfer, M.J.; Willett, W.C.; Hu, F.B. Associations of weight gain from early to middle adulthood with major health outcomes later in life. *JAMA 2017*, 318, 255–272.
69. Dicker D, Bettini S, Farpour-Lambert N, Frühbeck G, Golan R, Goossens G, Halford J, O'Malley G, Mullerova D, Ramos Salas X, Hassapiou M, N, Sagen J, Woodward E, Yumuk V, Busetto L: Obesity and COVID-19: The Two Sides of the Coin. *Obes Facts* 2020;13:430-438. doi: 10.1159/000510005.

A APPENDIX A: Required Python Code for Building and Testing the Hybrid Model

The required Python code is as follows:

Function for building and deploying model is as follows:

```
# Importing necessary Libraries
import numpy as np
import pandas as pd
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.feature_selection import SelectFwe, f_classif
from sklearn.model_selection import train_test_split
from sklearn.pipeline import make_pipeline, make_union
from tpot.builtins import StackingEstimator
from sklearn.preprocessing import FunctionTransformer
from copy import copy
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.feature_selection import RFE, SelectFromModel
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
from sklearn.pipeline import make_pipeline, make_union
from sklearn.preprocessing import MaxAbsScaler, RobustScaler
from sklearn.preprocessing import StandardScaler
from tpot.builtins import StackingEstimator
from sklearn.model_selection import train_test_split
from sklearn.metrics import recall_score
from xgboost import XGBClassifier
from sklearn.metrics import precision_score
from sklearn.metrics import accuracy_score
import warnings
from sklearn.metrics import confusion_matrix
from statistics import mean
from sklearn.metrics import f1_score
from sklearn.metrics import classification_report
warnings.filterwarnings ('ignore')

def obesity ():
    #Method to build, train and test hybrid model
    df=pd.read_csv('ObesityDataSet_raw_and_data_sinthetic.csv') # Source of csv file
    #Preprocessing the Synthetic data
    #Rounding of synthetic data points
    # df['Age'] =df['Age']. round(0)
    df['FCVC'] = df['FCVC']. round(0)
    df['NCP'] =df['NCP']. round(0)
    df['CH2O'] =df['CH2O']. round(0)
    df['FAF'] =df['FAF']. round(0)
    df['TUE'] =df['TUE']. round(0)
    #Encoding Categorical columns
    df = pd.get_dummies(df,prefix=['Gender'], columns = ['Gender'], drop_first=True)
    df = pd.get_dummies(df,prefix=['family_history_with_overweight_'], columns =
['family_history_with_overweight'], drop_first=True)
    df = pd.get_dummies(df,prefix=['FAVC'], columns = ['FAVC'], drop_first=True)
    df = pd.get_dummies(df,prefix=['FCVC’], columns = ['FCVC'], drop_first=False)
    df = pd.get_dummies(df,prefix=['NCP'], columns = ['NCP'], drop_first=False)
    df = pd.get_dummies(df,prefix=['CAEC'], columns = ['CAEC'], drop_first=False)
    df = pd.get_dummies(df,prefix=['SMOKE'], columns = ['SMOKE'], drop_first=True)
    df = pd.get_dummies(df,prefix=['CH2O'], columns = ['CH2O'], drop_first=False)
    df = pd.get_dummies(df,prefix=['SCC'], columns = ['SCC'], drop_first=True)
    df = pd.get_dummies(df,prefix=['FAF'], columns = ['FAF'], drop_first=False)
    df = pd.get_dummies(df,prefix=['CALC'], columns = ['CALC'], drop_first=False)
    df = pd.get_dummies(df,prefix=['MTRANS'], columns = ['MTRANS'], drop_first=False)
    #Target Column Wrong
    del df['NObeyesdad']
    df['BMI’] =df['Weight']/(df['Height'] * df['Height']) # Calculating Obesity according to formula
    # Classifying Obesity according to WHO Formula
    df.loc[df['BMI'] <18.50,'OBESITY'] = 1
    df.loc[(df['BMI']>=18.50) & (df['BMI'] <25),'OBESITY'] = 2
    df.loc[(df['BMI']>=25) & (df['BMI''] <30), 'OBESITY'] = 3
    df.loc[(df['BMI']>=30) & (df['BMI'] <35), 'OBESITY'] = 4
    df.loc[(df['BMI']>=35) & (df["BMI'] <40), 'OBESITY'] = 5
    df.loc[df['BMI']>=40,'OBESITY'] = 6
    del df['BMI']
    df3=df.copy()
    scaler2 = StandardScaler()
    df3[['Age','Height','Weight']] = scaler2.fit_transform(df3[['Age','Height','Weight']])
    target=df3['OBESITY']
    df3.pop('OBESITY')
    df3.pop('CALC_Always')
    X_train, X_test, y_train, y_test = train_test_split (df3, target, train_size=0.75, test_size=0.25,
shuffle=True)
    exported_pipeline = make_pipeline(
SelectFromModel(estimator=ExtraTreesClassifier(criterion="gini", max_features=0.1,
n_estimators=100), threshold=0.05),
RFE (estimator=ExtraTreesClassifier (criterion="gini", max_features=0.4, n_estimators=100),
step=0.6000000000000001), MaxAbsScaler(),
StackingEstimator(estimator=MLPClassifier(alpha=0.1, learning_rate_init=0.5)), RobustScaler(),
XGBClassifier(learning_rate=0.5, max_depth=7, min_child_weight=1, n_estimators=100,
n_jobs=1, subsample=0.8500000000000001, verbosity=0))
    exported_pipeline.fit(X_train,y_train)
    results = exported_pipeline.predict(X_test)
    #Comparing results
    print ('Overall recall')
    print (recall_score (y_test, results, average= 'weighted'))
    print ('Overall precision')
    print (precision_score (y_test, results, average= 'weighted'))
    print ('Overall Accuracy')
```

```
  print (accuracy_score (y_test, results))
  print ('Overall f1-score')
  print (f1_score (y_test, results, average= 'weighted'))
  print ('Confusion Matrix')
  print (confusion_matrix(y_test,results))
  cfm=confusion_matrix(y_test,results)
  avacc=(np.sum(cfm[i,i] for i in range(6)))/(np.sum(cfm))
  print ('Average Accuracy=')
  print (avacc)
  print (classification_report (y_test, results,digits=6, target_names=['Class 1', 'Class 2', 'Class 3','Class 4','Class 5','Class 6']))
  tp,tn,fp,fn,accuracy,precision,recall=indimetric(cfm)
  print ('Recall Mew')
  print (sum(recall)/6) # Recall mew
  print ('Precision Mew')
  print (sum(precision)/6) # Precision mew
```

The Python method to generate the True Positives, True Negatives, False Positives, False Negatives given a Confusion Matrix for a multi-class classification problem is given by:

```
def indimetric(cfm):
# Method to calculate the True positive, true negative, false positive and false negative for each class for a given confusion matrix
 import numpy as np
 inprec=[]
 inacc=[]
 inrec=[]
 tp=[]
 fp=[]
 tn=[]
 fn=[]
 for i in range(len(cfm)):
     tp.append(cfm[i,i])
     tn.append(np.sum(cfm[(i+1):,(i+1):])+np.sum(cfm[0:i,0:i])+np.sum(cfm[0:i,(i+1):]))
     fp.append(np.sum(cfm[0:i,i])+np.sum(cfm[(i+1):,i]))
     fn.append(np.sum(cfm[i,0:i])+np.sum(cfm[i,(i+1):]))
     inprec.append((tp[i])/(tp[i]+fp[i]))
     inacc.append(((tp[i]+tn[i])/(tp[i]+tn[i]+fp[i]+fn[i]))*100)
     inrec.append((tp[i])/(tp[i]+fn[i]))
#print (tp,tn,fp,fn)
#print(inprec)
 return tp,tn,fp,fn,inacc,inprec,inrec
```

B APPENDIX B: Required Python Code for Creating Synthetic Data with Respect to Covid-19

The required Python code is as follows:

```
def encode (df):
  # Changes a Given Pandas DataFrame df to match COVID-19 Scenario
  import pandas as pd
  import random
  wl=df['Weight'].to_list()
  for i in range(len(wl)):
    rand=random.uniform(0,4.455)
    wl[i]=wl[i]+rand
  df['Weight’] = wl
  return df
```

C APPENDIX C: Python Code for Running Tpot Simulations for AutoMl

The Tpot Model Generation Python code is as follows:

```
from tpot import TPOTClassifier
tpot = TPOTClassifier(verbosity=2, n_jobs= 3, warm_start= True, periodic_checkpoint_folder=
r'C:\Users\Akash\Desktop\Capstone Project\pipes')
tpot.fit(X_train, y_train)
print (tpot.score(X_test, y_test))
tpot.export('tpot_code_pipeline.py')
```

A sample generated Pipeline Python code is:

```
import numpy as np
import pandas as pd
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.feature_selection import SelectFromModel
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
from sklearn.pipeline import make_pipeline, make_union
from tpot.builtins import StackingEstimator
from xgboost import XGBClassifier

# NOTE: Make sure that the outcome column is labeled 'target' in the data file
tpot_data = pd.read_csv('PATH/TO/DATA/FILE', sep='COLUMN_SEPARATOR',
dtype=np.float64)
features = tpot_data.drop('target', axis=1)
training_features, testing_features, training_target, testing_target = \
train_test_split(features, tpot_data['target'], random_state=None)

# Average CV score on the training set was: 0.9946675328779871
exported_pipeline = make_pipeline(
SelectFromModel(estimator=ExtraTreesClassifier(criterion="gini",
max_features=0.7000000000000001, n_estimators=100), threshold=0.1),
StackingEstimator(estimator=LogisticRegression(C=25.0, dual=False, penalty=”l2”)),
StackingEstimator(estimator=MLPClassifier(alpha=0.0001, learning_rate_init=0.5)),
XGBClassifier(learning_rate=0.5, max_depth=3, min_child_weight=2, n_estimators=100,
n_jobs=1, subsample=0.6000000000000001, verbosity=0)
)

exported_pipeline.fit(training_features, training_target)
results = exported_pipeline.predict(testing_features)
```

D Copyright documentation

All images in this document are either publicly available or plotted by the author or licensed for reuse under Creative Commons license 3.0. Please see below for full citation and attribution information.

Figure 3: https://miro.medium.com/max/1400/1*QJZ6W-Pck_W7RlIDwUIN9Q.jpeg

Figure 4: https://blog.quantinsti.com/xgboost-python/

Figure5:https://www.sciencedirect.com/science/article/pii/S2352340919306985?via%3Dihub

Figure6:https://www.sciencedirect.com/science/article/pii/S2352340919306985?via%3Dihub

Figure 7: https://www.diva-portal.org/smash/get/diva2:920202/FULLTEXT01.pdf

Figure 17: https://www.karger.com/WebMaterial/ShowPic/1210498