HAND OUT DATE: 22nd September 2020

HAND IN DATE: 2nd October 2020

WEIGHTAGE:

INSTRUCTIONS TO CANDIDATES:

1 **Submit your assignment at the administrative counter**

2 **Students are advised to underpin their answers with the use of references (cited using the Harvard Name System of Referencing)**

3. **Late submission will be awarded zero (0) unless Extenuating Circumstances (EC) are upheld**

4 **Cases of plagiarism will be penalized**

5 **The assignment should be bound in an appropriate style (comb bound or stapled).**

6 **Where the assignment should be submitted in both hardcopy and softcopy, the softcopy of the written assignment and source code (where appropriate) should be on a CD in an envelope / CD cover and attached to the hardcopy.**

7 **You must obtain 50% overall to pass this module.**

GROUP ASSIGNMENT

Provisional Analysis for Obesity Issue using Various Data Mining Techniques

By

Wahidul Alam Riyad (TP043338)

Sylvia Chan Poh Yee (TP050862)

Abdulmalik Jibrin Ala (TP048737)

Madoma Anthony Diallo (TP048246)

UC3F2002

Lecturer Name : Dr. Booma Poolan Marikannan

Subject Code : CT099-3-3 BDA

Hand-Out Date : 22nd Sep, 2020

Hand-In Date : 2nd Oct, 2020

Table of Contents

Provisional Analysis for Obesity Issue using Various Data Mining Techniques

Abstract

Data mining is a method used by corporations to convert valuable knowledge into raw data. Businesses can learn more about their clients and create more efficient marketing campaigns, boost revenue and decrease costs by using algorithms and search for trends in vast batches of data. Effective data collection, storage, and computer processing rely on data mining. In order to collect relevant patterns and trends, data mining means discovering and evaluating vast blocks of information. It can be used in a number of ways, such as targeting databases, handling credit risk, detecting theft, screening spam emails, or even to distinguish users' feelings or opinions. The method of data mining divided into five stages. Organizations first gather information and bring it into their data centers. Next, the data is processed and handled, either on in-house or cloud servers. Company consultants, executive teams and experts in information technology view the data and decide how they want it to be structured. Then, program software philters the data based on the performance of the user, and then, the end-user displays the data in a way that is simple to display, such as a graph or table. There are several data mining techniques been compared in the discussion. The researchers are going to discuss the algorithms to identify which one is more suitable to use to predict the diabetes diseases to get the highest accuracy.

1.0 Introduction

The IT organization, classified by and embraced by enormous data set from sensor networks, online networks, medical agencies, etc., is creating and exploring big data applications. To tackle data analysis, we study the latest algorithms for data mining and information discovery through network property (Raja and Pandian, 2020). The authors discuss the application of data mining techniques to address obesity in the health sector. In order to build the predictive model of diabetes complication disease, classification of data mining technique and its algorithm were studied. The authors had analyzed the assembly from classification methodology in order to construct the most fitting rule-based model for the prediction intent. It can be shown that better results, output and features may be categorized relative to the classification technique. The author also obtain information from comparing different algorithms to get the highest accuracy of the

classification process through the dataset. The future work of this research is to improve the prediction models and get to work on the real dataset to help human being to solve the risk of getting obesity.

1.1 Research Goal

The aim for this paper is to do provisional analysis using numerous data mining techniques to reduce the chance of getting obesity.

1.2 Objectives

This paper aims to classify obesity based on the value of BMI, forecast the profitability of customers based on consumer products, and find the number of customers that are not returning. Numerous algorithms have been chosen for Data Mining Techniques.

2.0 Related Works

2.1 Wahid (Article 1)

Problem Statement

In the prevention and treatment of childhood obesity, digital health approaches based on instruments for Computerized Decision Support (CDS) and Machine Learning (ML), which take advantage of new information, sensing, and communication technologies, can play a crucial role. The author provides a systematic literature analysis of childhood obesity prevention and treatment applications for CDS and ML (Triantafyllidis et al., 2020). To advance their understanding of smart and efficient approaches for childhood obesity treatment, the key characteristics and findings of studies using CDS and ML are illustrated.

Literature Review

To find childhood obesity studies integrating either CDS approaches or advanced data analytics through ML algorithms, a search was performed in the bibliographic databases of PubMed and Scopus. Continuous, case-based, and qualitative studies were omitted, along with those that did not have precise quantitative data (Triantafyllidis et al., 2020). The studies integrating CDS were synthesized according to the critical intervention technology (e.g., mobile app), a form of design (e.g., randomized controlled trial), number of participants participating, children's target age, the

follow-up period of participants, the primary outcome (e.g., Body Mass Index (BMI)) and critical feature(s) of CDS and their results (e.g., caregiver notifications when BMI is high). The ML-integrated studies were synthesized by the number of participants involved and their age, the ML algorithm(s) used (e.g., logistic regression), as well as their primary outcome (e.g., obesity prediction).

Eight studies implementing CDS approaches and nine studies using ML algorithms were found by the literature quest, which met our eligibility criteria. Both studies reported (e.g., in terms of accuracy) statistically relevant interventional or ML model effects. More than half of the interventional studies were planned as randomized controlled trials (n = 5, 63 percent) (Triantafyllidis et al., 2020). Electronic Health Records (EHRs) and warnings for BMI were used as CDS for half of the interventional trials (n = 4, 50%). The highest percentage targeting the prognosis of obesity was from the nine studies using ML (n = 4, 44 percent). It was shown that **random forests** and **artificial neural networks** could reliably predict childhood obesity in studies integrating more than one ML algorithm and reporting accuracy.

This review found that CDS tools can be useful for childhood obesity self-management or remote medical management (Triantafyllidis et al., 2020). At the same time, ML algorithms can be helpful for prediction purposes, such as random forests and artificial neural networks. Considering the low number of studies reported in this study, their methodological limitations, and the lack of interventional studies using ML algorithms in CDS instruments, more systematic reviews in the field of CDS and ML for childhood obesity treatment are required.

2.2 Wahid (Article 2)

Problem Statement

The diabetic disease usually consists of blood sugar levels that are higher than average. Instead, insulin production could be considered inadequate. In recent days, it has been noted that the number of patients affected by diabetes has risen to a greater extent, globally (Raja and Pandian, 2020). This issue must be taken more seriously in the coming days to ensure that the overall number of diabetes individuals is reduced. Several research teams have recently performed extensive research on the data mining platform to assess each other's accuracy. Data mining can synthesize knowledge in the field by parametric modeling of health data, including diabetic patient data sets.

Literature Review

A new model for forecasting type 2 diabetes mellitus (T2DM) based on data mining techniques is proposed in this research. A collection of medical data relating to a diabetes diagnosis problem will be analyzed using the combined **Particle Swarm Optimization (PSO) and Fuzzy Clustering Means (FCM) (PSO-FCM)**. The generic approach of the FCM algorithm is highly sensitive to noisy results. The most significant drawback is the product of the overall effectiveness of the problem (Raja and Pandian, 2020). To solve this problem, a revision of the FCM specification and the input given as the preprocessed information is needed. This paper proposes an efficient method that incorporates the PSO's best attributes into the traditional FCM method, enhancing the participants' weighting of the cluster.

Concerning each cluster in the framework suggested, every point in the entire data set has a special weight. This particular weight plays a significant role in the successful clustering of noisy data. The Pima Indians Diabetes Database is used to conduct research. To evaluate the proposed device's efficacy, reliability, sensitivity, specificity, and accuracy metrics commonly used in medical studies were used (Raja and Pandian, 2020). The prototype has been found to have achieved 8.26% more precision than the other approaches. The results in PSO-FCM are expressed in this paper in an efficient model comprising the best attributes of PSO and FCM. After a careful analysis of the previously generated works, a systematic methodology has now been developed, such as PSO and FCM. The best and best parameters are first observed, then the successful clustering tests are applied to FCM.

While FCM is a good clustering technique, the system's accuracy is reduced due to preterm integration (Raja and Pandian, 2020). To boost efficiency and retain the unique characteristics of FCM, the famous PSO optimization technique was merged. Premature convergence was banned, and a better accurate value was acquired for kappa statistics, showing that the predicted model is useful in predicting diabetic disease at an earlier stage.

2.3 Sylvia (Article 1)

Problem Statement

Based on the research, nowadays, the risk of getting obesity is higher. A popular condition where too much sugar (glucose) flows around in the blood is diabetes mellitus. This is when either the pancreas is unable to produce enough insulin or the insulin resistance cells in your body have become sluggish. The capacity of the human body to use the energy available in food is impaired by diabetes. Basic forms of diabetes are: type 1 is diabetes pancreas does not generate sufficient insulin

and, as a result, the blood glucose level exceeds the normal amount. Individuals with this form of diabetes typically rely on external insulin that is injected into the bloodstream at regular intervals. A hereditary predisposition triggers it. Diabetic retinopathy (eye disorder), diabetic neuropathy (nerve disorder), and diabetic nephropathy (kidney disorder) are medical dangers associated with this form of diabetes. 95 per cent of cases of diabetes are counted. In Type 2 diabetes, owing to insulin resistance, the body is unable to absorb insulin sufficiently. It is usually induced by obesity and overweight kids. It is non-insulin dependent and is milder than diabetes type 1. This has serious effects on heart failure and heart strokes. With adequate diet, exercise and weight maintenance, it cannot be treated but managed. Gestational diabetes involves married women who, according to prior medical records, are not affected by diabetes but are diagnosed with elevated levels of glucose during / after birth. The estimated prevalence of gestational diabetes is between 2 per cent and 10 per cent of births, according to the National Institutes of Health.

Data Mining Technique

One of the data mining technique implemented K-Nearest Neighbor (KNN) is an algorithm for supervised learning used for data classification. K implies choosing points from the specified dataset to choose how many data from the closest neighbor would be chosen. This algorithm selects data for the closest neighbor on the basis of the K value and determines that this point is close to the sample given. On datasets with K values ranging from 1 to 10, we use KNN. First, we classify patient outcomes and split data into 70 percent, 30 percent as training and test information, respectively, and then we do 10-fold cross-validation, also with 20 folds, by randomly sampling the split data value and adding KNN to the defined results. We notice many accuracies using 10 and 20-fold in the KNN algorithm by applying the K-nearest Neighbor algorithm, using 1 to 5 nearest neighbors on the dataset. The highest precision can be shown by using 1 closest neighbor in 20-fold.

Models selected to achieve goals

Dataset includes female records that are at least 21 years old and reside in Phoenix, Arizona, USA. Dataset contains classified data with binary class classification (0 or 1). Dataset contains labelled data. The class attribute value 0 reflects a negative examination and the diagnosis of diabetes is 1. Dataset comprises 768 records of multiple patients of which 268 (34.9 percent) data are positive test reports indicating '1' classification attribute value and 500 (65.1 percent) reports

reflecting negative test of classification '0'. Dataset attributes with their description, type, and units are provided.

Model selected as a best performing

As each algorithm operated on various methods, the results obtained from these 3 implemented algorithms vary. Through adding more pre-processing methods and data filtration, results derived from this dataset can be improved. The accuracy obtained from Decision Tree is best, but the graph is more distributed, which can also be increased. The lowest precision is from KNN. KNN is checked with a wide range of K values from 1 to 10 and with folds from 10 to 20 that shift, but there is still not much precision. Below is a pictorial representation of the data in graph form. (Azrar, et al., 2018)

2.4 Sylvia (Article 2)

Problem Statement

Diabetes Mellitus (DM) is characterized as a series of metabolic complications mainly caused by glucose excess within the circulation system. The World Health Organization says that by 2030, approximately more than 700 million people are likely to develop diabetes. Patients with diabetes exist around the world, but in developed countries it is more usual. The predominance of diabetes in Indonesia was 10.9 percent, and the tendency is rising step by step. Diabetes might affect the veins as metabolic complications, which increase the risk of serious intricacies of well-being that threaten the heart, skin, kidneys and nerves. Diseases are divided into two divided dependents with the more commonly known diabetes issues based on their damage to small veins and damage to the courses. Bunch of microvascular infection into which organ, which are eye, kidney and neural injury, the ailment attack. The main macrovascular intricacies include elevated coronary disease, which is seen as strokes by multiple genuine illnesses. Retinopathy, neuropathy and nephropathy are the best three of the diabetes microvascular intricacy symptoms, as shown by the Indonesian Ministry of Health. In addition, one of the ways in which this can be feasible is to gather data on the danger factor in order to avoid the deterioration of the entanglements. Owing to the high mortality and dismalness of diabetes entanglement disorders, the expectation of anticipation and danger factor becomes a major and growing trend of topic and research review. Several experiments were aimed at gathering knowledge relating to hazard components and diabetes and prediabetes research. Be that as it may, hardly any studies have been led to evaluate the

complexities of diabetes disorders, particularly its risk factors. Therefore, diabetes problem disorders appear to be underused in disease prevention and may not always be found until the disorder has recently been seen in the suffering situation. In light of this clarification, diabetes is otherwise called the silent executioner. Two independent workshops, which are altered and unmodified, measure the risk of diabetes. Modified features, such as color, gender, size, sexual identity, and so on. Unmodified with an unnecessary and passive way of living. In this study, we used properties from the Indonesian diabetes patient's clinical history to assess the risk factor for each of Indonesia's three main diabetes uncertainty disorders.

Models selected to achieve goals

Three outlets, Sri Pamela Hospital and Kumpulan Pane Hospital, Tebing Tinggi, and Dolok Community Health Center, North Sumatra, Indonesia, given the diabetic clinical record used in this test. The information index contains 158 clinical reports, with 15 features, in the wake of the extraction of defective information via the Extraction Transform, Load (ETL) procedure. Scientists use 8 properties as changed and unmodified risk factors for diabetes in the context of taking out patient specific evidence from 15 characteristics. In the unmodified risk factor, the aspect of diabetes is sexual preference and family heritage. The portion indicates typical well-being measures of restraint with regard to modified danger factors, such as BMI, circulatory pressure, duration of diabetes patients, blood glucose level and age of the patient. These threat variables have modified into eight highlights, including silent infections with sophistication that can be used as a goal.

Data Mining Technique

For the purpose of characterization, the researchers agree Naive Bayes (NB) for the portion determination of absolute information, which is sexual orientation and diabetes family ancestry. The Naive Bayes estimation generates the contingent probability of each knowledge trait given a possible value taken by the yield property as a probabilistic model of the learning process. At that point, for the remainder of highlight range, the researchers get C4.5. The researchers chose the calculation of C4.5 because it can reinforce ordinary knowledge and this calculation 's option tree creation law is moderately straightforward. The norms generated focus on determining the most visible approximation of the correct introduction for each grouping disease of each element collection. For each grouping disease of each element category, the concepts generated depend on the most unmistakable estimate of the precision introduction.

None of the diabetes family history (81 percent), hypertension epidemic (76 percent), mild (50 percent) and long-term (50 percent) diabetes period, blood glucose level between 200-400 mg / dL (71 percent), regular BMI (42 percent).

Neuropatient: None in the family history of diabetes (86%), amount of blood glucose between 200-400 mg / dL (62%), period with diabetes more than 10 years (41%), BMI rate greater than 25 (41%).

Patient with nephropathy: Diabetes with a typical BMI range of more than 10 years (87%) (52%).

Other diseases: None in the family history of diabetes (72%), amount of blood glucose between 200-400 mg / dL (62%), period of diabetes between 4 and 10 years (51%).

Model selected as a best performing

In conclusion, the researchers analyze the presentation from the bunching and grouping process as the most rational principle-based paradigm for the expectation cause. It appears to be shown that classification technique offers improved outcomes, implementation, and might organize highlights and sub-include three major microvascular diabetes entanglement diseases in comparison to grouping strategy. We will close the most important risk factor for any diabetes misunderstanding illness from the knowledge mining inquiry. It turns out that considering the fact that the amount of blood glucose and the word of diabetes suffer contribute to difficulty vomiting, but on Neprophaty it is normally visible. It also presumes that the amount and consistency of glucose (family history of diabetes) do not influence the overt difficulty of diabetes. Similarly, in a number of hypertension crises, the most commonly known risk factor for retinopathy is the heartbeat. (Fiarni, et al., 2019)

2.5 Abdulmalik (Article 1)

Literature Review

The regression model is the most generally utilized model to distinguish the connections between the different various numerical, categorical and independent variables for a single dependent variable, a variable selection was made utilizing it. Accordingly, to conduct the variable selection regression technique was selected, separating variable from among 30 extracorporeal and intracorporeal variable that impact corpulence. For the variable determination technique, a regression, which is a different relapse model. Information preprocessing was finished by choosing the variable utilizing the analysis of regression.

They are four(4)algorithms that has been selected to solve the identified problem, the uses of the four models was made a decision about utilizing the accuracy of the model, the accuracy esteem, and the review esteem, the precision of these data was inferred by haphazardly separating 100 data for four kinds of obesity information's so as to choose the model that creates the most higher accuracy between the four selected models and the model that determines the who have the most accurate rate for the obesity information. the four sorts of models. Random forest, multiclass decision jungle, multiclass neural network, and multiclass logistic regression were used (Kim and Youm, 2020).

The random forest algorithm was selected because it had the highest accuracy from among the four models. we utilized the selected model and 11 variables, with just extracorporeal of the data, to predict the information for obesity. Further, online application was made to make inferences from the corpulence data esteems entered by clients. In this web mini-computer, expectations of the four degrees of heftiness, (ordinary weight, underweight, overweight, and hefty) and contrasts them with the existing obesity measurement.

Problem Statement

Obesity results in orthopedic issues, just as maladies identified with most organs, for example, asthma, diabetes, hypertension, cardiovascular sickness, and despondency, with short and long consequences results. Many people know that weight can cause physical issues, however they don't have the foggiest idea how worst their bodies are harmed or see how to improve their condition. The examination just estimated body circuits and didn't think about in general extents, or bulk mass of muscle. On account of hypertension, atherosclerosis, and coronary artery disease (CAD) are found in both thin individuals and obese individuals, it was decided that the clinical finding of instinctive fats may be a higher priority than the obese analysis utilizing BMI. This is something past an issue of excess weight, stoutness also achieves muscular issues, similarly as illnesses related to most organs, corpulent reason physical issues, anyway they don't have the foggiest thought how genuinely and serious their bodies are hurt or perceive how to improve their condition(Kim and Youm, 2020).

2.6 Abdulmalik (Article 2)

Literature Review

The classification data mining technique have been applied to the prediction of overweight and obese young people in their initial years, in view of the Wirral data set. As a rule, expectation from early ages is troublesome, somewhat on the grounds that the reasons prompting overweight and obese are confounded, including physiological as well as hereditary, sociological and even mental components. The most elevated overweight forecast rate is 55–60% in this work. The analysis proposes that weight forecast at an early age is troublesome. Part of the explanation lies in the way that the quantity of corpulence cases is excessively little, being just 3.29% of the entire dataset (Zhang and Tjortjis,2018). The examples between the two classes are earnestly out of parity, this makes the investigation tricky, moreover, numerous non-obese examples are like hefty examples at an early age.

The comparison of logistic regression with six data mining techniques (decision trees (C4.5), association rules, Neural Networks (NNs), naïve Bayes, Bayesian networks and Support Vector Machines (SVMs)) to do prediction for the issue to explicitly, for the prediction of overweight and obese kids at 3 years old utilizing information recorded during childbirth, a month and a half, 8 months and 2 years individually improved precision of expectation by utilizing data mining procedures.

Linear regression is the major types of prediction algorithms, classification is used to predict discrete or ostensible qualities, linear regression is used to continuous predict or value requested. The fundamental target is to recognize and predict the gathering of kids who are in danger of turning out to be overweight and consequently require safeguard activity, as opposed to anticipate the BMI of individual youngsters.

Problem Statement

There is a developing scourge of obesity influencing all age gatherings, with the commonness of obesity in the UK rising quickly in youngsters as youthful as 3 years. It has been accounted for that among two to four-year-old, obesity has multiplied since the mid-1990s, while the rate has trebled for six to 15-year-olds.In the UK, among those under 11, corpulence expanded from 9.6% in 1995, to 13.7% in 2003 (From Health Surveys for England announced in The Times, February 28th 2006). The expansion in youth obesity is causing worry in different nations, just as the UK. Being fat as a kid causes prompt damage, for example, low self-esteem, and has ramifications for grown-up wellbeing including long lasting danger of corpulence and an expanded danger of type 2 diabetes. A few different ways have been recommended to treat corpulence in

youngsters, for example, physical exercise joined with sustenance instruction or conduct change(Zhang and Tjortjis,2018). Nonetheless, there would be a more prominent general wellbeing sway from forestalling weight than treating it.

2.7 Madoma (Article 1)

Literature Review

Sisosdia and Sisodia (2018), looked to automate the early identification of diabetes to enable potential patients to stay away from influencing factors before complications arise. They conducted their research to leverage machine learning to prognosticate the chances of diabetes for pregnant women patients in India. The paper highlighted that diabetes is a health condition where the measure of sugar substance cannot be controlled. The failure to treat diabetes lead to intensified hunger, thirst, diabetic ketoacidosis and in severe cases hyperosmolar coma experienced by the patients.

The research utilised the Pima Indians Diabetes Database (PIDD) and evaluated the performance of the Decision Tree, Naïve Bayes and SVM classification algorithms. For the supervised learning work, the PIDD dataset contained 768 instances of female patients and 8 numeric attributes where a value of 0 was assigned to negative tests of diabetes and 1 for those who tested positive. Their work used WEKA software.

The result of the research showcased Naive Bayes as the best performing algorithm which obtained an accuracy of 76.30%. Although the research showcased the SVM model which achieved an accuracy of 65.10%, it lacked information on the kernel utilised which has influenced the researcher to utilise the SVM model and see how utilising different kernels might influence accuracy.

2.8 Madoma (Article 2)

Literature Review

Babajide et al. (2020) looked to aid a dietary intervention program to help weight and obesity management. The research investigated SVM, Random Forest, Linear Regression and Artificial Neural Networks models and how they could predict body weight at the end of the dietary intervention program. Dietary intervention is tasked around controlling individual diets therefore influencing body weight, obesity, and overall health (Wu, 2013). The work applied supervised

machine learning methods to aid weight-loss prediction in the NUGENOB dietary intervention project.

The dataset utilised was extracted from the NUGENOB database which contained profiles of the program members as attributes. The influential features selected are shown below.

Variable names		
1. Age	2. Body Weight@ Week 0 (Base-line)	3. Gender
4. Mean waist-hip ratio baseline	5. Fat mass baseline	6. Fasting glucose baseline
7. Basic metabolic rate baseline	8. Energy expenditure. T.0	9. Height
10. HOMA Insulin resistance(I0)	11. Fasting insulin baseline	

The 4 models were evaluated on the RMSE, MAE and R2 metrics. The results identified that Random Forest had the lowest error rate and a R-square value of 96%. The research noted the MAE (Mean absolute error) as the main metric used in assessing the models, where the lower the number the better the model.

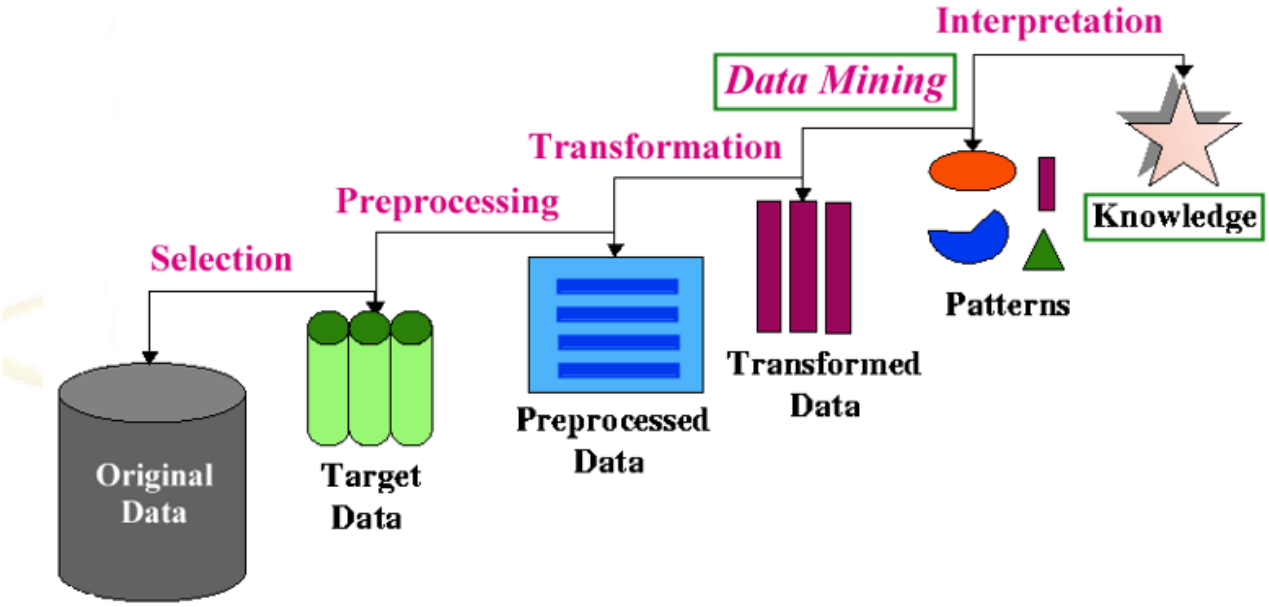
	LM	SVM	RF	ANN	Dynamic model
RMSE	4.309964	4.349037	3.268409	3.55828	4.629934
MAE	3.438419	3.493781	2.64141	2.763006	3.791817
R-squared	0.930191	0.930122	0.960241	0.953391	0.954028

Key: LM - Linear Regression, SVM - Support Vector machine, RF - Random Forest, ANN - Artificial Neural Network

Figure:Model comparison based on different metrics Babajide et al. (2020)

Based on the figure above the Random Forrest was the best performing algorithm. The researchers concluded this was influenced due to the model's ability to handle small data sets better compared to others. Noting the data set used for training was 443.

3.0 Big Data Analytics Lifecycle & Methodologies



This paper studies the classification of obesity using 4 data mining techniques: Random Forrest, K-Nearest Neighbor, Support Vector Machine and Linear Regression. These techniques have been implemented with adherence to the CRISP-DM methodology. The steps that conform with the 6 phases of the data Analytics Lifecycle and the knowledge discovery process will be followed to achieve the aim of this paper.

Figure: CRISP Methodology

The CRISP-DM (Cross Industry Standard Process for Data Mining) methodology is a proven methodology created in 1996 that provides a structured guidance to approach a data mining

project (Vorhies, 2016). The methodology progresses as follows, it starts with Business understanding, data understanding, data preparation, modeling, evaluation and ends with deployment.

Business understanding phase focuses on understanding the business perspective and objectives of the situation. It is where the data mining goals have been formed. For this research, the business objective entails helping serve customers better by helping them maintain better health through suggesting food products the business produces. Therefore, by finding out the BMI index of a customer the business can suggest healthier meals.

The data understanding phase entails the collection of the data and getting familiar with it. Data exploration also is done at this phase. The data quality is also reviewed to find if the original data has any redundant, missing, or noisy data. This phase resembles the selection stage of the knowledge discovery process.

Data preparation covers the construction of the final dataset from our original data. It resembles pre-processing where data cleaning, splitting occurs to attain transformed data. The original data quality is now modified to be usable for the modeling phase. This phase is showcased in chapter 3.5 and 3.6 of this paper.

Modeling is the key phase where data mining techniques are selected, applied. It is the Data mining process of the knowledge discovery process. It also is the model planning and building phases of the data analytics lifecycle.

The evaluation phase consists of testing as well as assessing models to ensure they sufficiently meet the business issues considered at this stage the output is the selection of the best model/technique.

The final stage is the deploying of the model into a system to score on unseen data. This phase is not covered in this research but would be an important phase during a real-world project. The results of this research would suffice to showcase the project in retrospective of what the research brings to the improvements of the business.

4.0 Methods

Dataset collection

This consists of all the information and the description of every aspect of the dataset we are using to complete our tasks.

Table 1: Metadata of the dataset

Dataset name	500 Person Gender-Height-Weight-Body Mass Index
Data size	8 kilobyte(kb)
Usage information	License: GPL 2info Visibility: Public
Maintaners (dataset owner)	Yasin Ersever
Updates	Expected update frequency: Not specified Last updated: 2018-07-03 Date created: 2018-07-03 Current version: Version 2

Data source

The dataset was acquired from the Kaggle website. The Kaggle website is an open platform for modeling prediction and analysis, also use for competition between people from different places around the world for analyzing data. Also, people share their work to know who has the best prediction model and even the steps individuals use to achieving their goal. Some companies too shared dataset for people to use for their predicting and analysis. Kaggle is available to access and view, as it is an open free platform for anyone. Our dataset was recovered from https:// www.kaggle.com/yersever/500-person-gender-height-weight-bodymassindex. The dataset was to use to analyze or predict obesity in a person. The dataset consists of the gender, weight, height, and index of each person in it. The database was chosen because it consists of the information, then we need to analyze/classify and predict obesity.

Data description

The fact that the chosen dataset is from the Kaggle site principally center around The aim of analyzing/classify the obesity and get the prediction, for the analysis/classification to be conducted, they have to be some specific data about the person as giving in the attribute which are the gender, weight, height for us to index/BMI to be probing. The dataset we are using consists of 500 records altogether, they are also five attributes in the chosen dataset. Essential comprehension of data

attributes will diminish the exploration territory and stay away from the pointless outcome or helpful yet false outcome. Will give a short description of data type, attributes name, data description, data category, and sample data.

Table 2 : Summary of attributes

Attributes name	Data type	Category	Description	Sample data
Gender	String	Categorical	Gender type of the person	Male, female
Height	Integer	Numeric	Height of the person	174, 185, 189
Weight	Integer	Numeric	Weight of the person	87, 96, 110
Index	Integer	Numeric	Body max index of the person	5, 4, 3, 2, 1

Table 3: Relationship between data attributes and their representatives

Variable name	Value	Description
Index	5	Extremely obesity
	4	Obesity
	3	Normal
	2	Weak
	1	Extremely Weak

As represented in the table before, some attributes fall under the categorical value, not numeric. The index is the only attribute. This table displays the exact amount for each of the features that use the string, which is under the categorical data type.

Missing value

Missing Value Missing data is the most ongoing issue usually face by the analyzer when conducting the data processing stage. The missing value is distinguished as an invalid value that puts away a variable inside the perception's intrigue region. It will prompt the predisposition and decrease of the measurable variable; the null value reduces the dataset's representativeness and

makes the model forecast and analysis more complicated. For our chosen dataset, this is no missing value been found in it.

A Gender

Gender : Male / Female

Female	51%
Male	49%

Valid	500	100%
Mismatched	0	0%
Missing	0	0%
Unique	2	
Most Common	Female	51%

 # Height

Height : Number (cm)

Valid	500	100%
Mismatched	0	0%
Missing	0	0%
Mean	170	
Std. Deviation	16.4	
Quantiles	140	Min
	156	25%
	171	50%
	184	75%
	199	Max

 # Weight

Weight : Number (Kg)

Valid	500	100%
Mismatched	0	0%
Missing	0	0%
Mean	106	
Std. Deviation	32.4	
Quantiles	50	Min
	80	25%
	106	50%
	136	75%
	160	Max

 # Index

Index : 0 - Extremely Weak 1 - Weak 2 - Normal 3 - Overweight 4 - Obesity 5 - Extreme Obesity

Valid	500	100%
Mismatched	0	0%
Missing	0	0%
Mean	3.75	
Std. Deviation	1.35	
Quantiles	0	Min
	3	25%
	4	50%
	5	75%
	5	Max

Outlier

An outlier is a value with a significant unusual distance between one value to another attribute in the dataset. The least demanding way deals with identifying outliers by checking out the mean and standard deviation of a particular feature. If the standard deviation number is more significant than of mean, that means the attribute may be having outlier.

Data cleaning

Data cleaning may be needed to increase the success percentage for the model to be built when they are missing values or outliers in attribute or a dataset. For our dataset and all our features, none of them consist of the outlier or missing value, so data cleaning is no need to increase the model success percentage.

5.0 Dataset Preparation

Features selection

The feature selection is an attribute importance mining capacity to build the nature of the model. Then again, reduce data size is ready to diminish the system's handling season to construct or assemble a model. Features selection assumes a critical function for model structure as data expert engaged with data examination. It assists with precluding predicting by investigating the variable.

	A	B	C	D	E	F
1	Gender	Height	Weight	Index		
2	Male	174	96	4		
3	Male	189	87	2		
4	Female	185	110	4		
5	Female	195	104	3		
6	Male	149	61	3		
7	Male	189	104	3		
8	Male	147	92	5		
9	Male	154	111	5		
10	Male	174	90	3		

The above figure shows the attribute and structure of all the data available in our dataset.

Gender
Height
Weight

The above table show us the features select, which are the x values. This selected feature will help to analyze/classify the obesity in each person by using their gender, height and weight data.

Data splitting

```
In [8]: X_train=X[:80]
        X_test=X[20:]

        Y_train=Y[:80]
        Y_test=Y[20:]
```

The splitting of the dataset into two (2) other unique use testing and splitting is divided into train and test set. The researcher selected the dataset is split our dataset into 80% and 20%. The building for testing subset your model. The testing subset is for building your model. The testing subset is for utilizing the model on obscure data to assess the performance of the model. Each takes in data and afterward fits a model with boundaries. The boundaries hold values. At that point, the model can take data and predict. As a rule, the training dataset and test dataset must be similar, have similar predictors or variables.

6.0 Algorithms Model Implementation & Model Validation

6.1 Wahid (Random Forest)

Random forest is a versatile, easy-to-use machine learning algorithm that produces, most of the time, a great result even without hyper-parameter tuning. Its simplicity and variety are also

among the most used algorithms (it can be used for both classification and regression tasks). In this post, we can learn how the random forest algorithm operates, how it varies from other algorithms, and how it can be used (Niklas Donges, 2019). A significant benefit of random forest is that it can be used for most existing machine learning systems, both classification and regression problems. Let's look at the classification of random forests since classification is often considered the building block of machine learning. You can see below how a random forest will look with two trees:

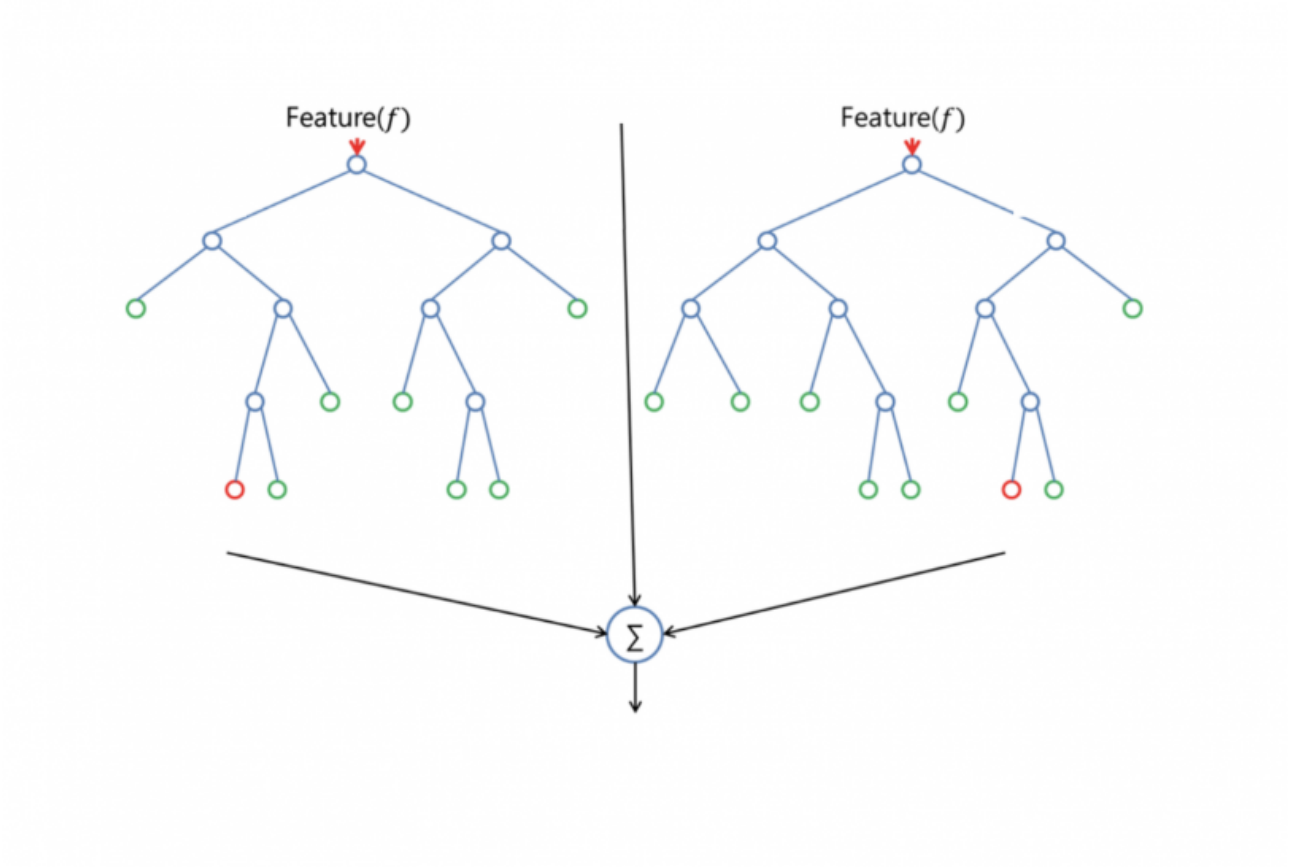


Figure: shows Random Forests (Niklas Donges, 2019)

As a decision tree or a bagging classifier, the Random forest has virtually the same hyper-parameters. Fortunately, it is not appropriate to combine a decision tree with a bagging classifier since the classifier class of random forest can be easily used. With random forest, using the algorithm regressor, you can also deal with regression tasks. While increasing the trees, Random Forest adds additional randomness to the model (Niklas Donges, 2019). It searches for the best feature among a random subset of elements instead of searching for the most relevant aspect when breaking a node. This results in a wide diversity that contributes to a better model in general. Therefore, the algorithm for splitting a node considers only a random subset of the features in the random forest. Using random thresholds for each function rather than looking for the best possible points (like a standard decision tree does), you can even render trees more random.

Pros & Cons of Random Forest Algorithm

The flexibility of the random forest is one of the tremendous benefits. It can be used for both regression and classification tasks, and the relative significance it assigns to the input features is also simple to see. Random forest is also a practical algorithm because the default hyper-parameters always generate an excellent prediction result. It's relatively easy to grasp the hyper-parameters, and there are not that many of them either (Niklas Donges, 2019). Overfitting is one of the significant challenges in machine learning, but thanks to the random forest classifier, this will not happen most of the time. The classifier is not going to overfit the model if there are enough trees in the forest. The random forest's critical drawback is that the algorithm can be too slow and inefficient for real-time predictions for a large number of trees. In general, once they are trained, these algorithms are easy to learn but very slow to make predictions. More trees are required for a more precise prediction, which results in a quieter model. The random forest algorithm is quick enough for most real-world applications, but there will be situations where run-time efficiency is essential, and other approaches are preferred. And, of course, the random forest is a predictive modeling method and not a descriptive way, which means other methods will be better if you're looking for a summary of the relationships in your results.

Random Forests Implementation using Python

Importing Libraries

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import os
print(os.listdir("../input"))
```

```
import seaborn as sns
from matplotlib import pyplot as plt
import matplotlib
%matplotlib inline
```

For this project, libraries such as numpy, pandas, seaborn and matplotlib has been used with python for implementation. In the above code, it clearly shows the implementation of the libraries in the python notebook file.

Reading CSV File

```
data = pd.read_csv('../input/500_Person_Gender_Height_Weight_Index.csv')
data_visual = pd.read_csv('../input/500_Person_Gender_Height_Weight_Index.csv')
```

This code shows how to read a CSV file using python from Kaggle Editor. Pandas has also used for data processing.

Classification of Obesity

```
def convert_status_to_description(x):
    if x['Index'] == 0:
        return 'Extremely Weak'
    elif x['Index'] == 1:
        return 'Weak'
    elif x['Index'] == 2:
        return 'Normal'
    elif x['Index'] == 3:
        return 'Overweight'
    elif x['Index']== 4:
        return 'Obesity'
    elif x['Index'] == 5:
        return 'Extreme Obesity'
data_visual['Status'] = data_visual.apply(convert_status_to_description,axis=1)
data_visual.head()
```

Main Kernel Content

This code represents the classification of obesity. Each number has been assigned with the level of obesity. The dataset contains the index number for the obesity level. For instance, 0 represents Extremely Weak while 5 represents Extreme Obesity. The following figure is the output of the classification status.

	Gender	Height	Weight	Index	Status
0	Male	174	96	4	Obesity
1	Male	189	87	2	Normal
2	Female	185	110	4	Obesity
3	Female	195	104	3	Overweight
4	Male	149	61	3	Overweight

Importing Random Forest Classifier

```
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier(n_estimators=200, criterion='entropy', random_state=0)
rfc.fit(X_train, y_train)
```

This code shows that from scikit learn, we are importing Random Forest Classifier. This will help us later to do the classification with higher accuracy and the most optimum results. See **Chapter 7.0** for more implementation, output and visualization.

6.2 Sylvia (KNN)

One of the easy and quick to apply supervised machine learning algorithms that can be used to solve regression and classification problems is the K-nearest neighbors algorithm, known as the KNN algorithm. Based on similarity measurements, KNN algorithms use data and identify new data points. The naming of its neighbors is carried out by a plurality vote. The knowledge is allocated to the class that has the closest neighbors. If increase the number of nearest neighbors, accuracy will increase the value of k. (Singh, 2018)

For the advantages of KNN algorithm, it is uncomplicated and easy-to-use nature, the KNN algorithm is commonly used for various forms of learning. Instance-based learning is KNN. In the preparation phase, they do not understand much. It does not derive any biased function from the data from the training. There is no preparation time for it, in other words. It stores and learns from the testing dataset just at the moment of making predictions in real time. This makes the KNN algorithm much simpler than other training algorithms.

The disadvantages of this algorithm are the cost of measuring the distance between the new point and each current point in large datasets is high, which degrades the algorithm 's efficiency. With high dimensional details, the KNN algorithm does not operate well because with a large number of dimensions, it becomes difficult for the algorithm to measure the distance in-dimension.

For the implementation, required libraries needed for the implementation of the Python KNN Algorithm. Numpy is imported for the calculation, matplotlib.pyplot for scheming the graph as well as KNeighborsClassifier to implement k nearest neighbor models for classification.

```
import numpy as np
import seaborn as sns
import pandas as pd
from sklearn import svm
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
```

Fetch the data and read the data for 500 persons with Height, Weight, Gender and Index.

```
bmi_health = pd.read_csv("C:\\Users\\scpy2\\OneDrive\\Desktop\\Test\\500_Person_Gender_Height_Weight_Index.csv")
```

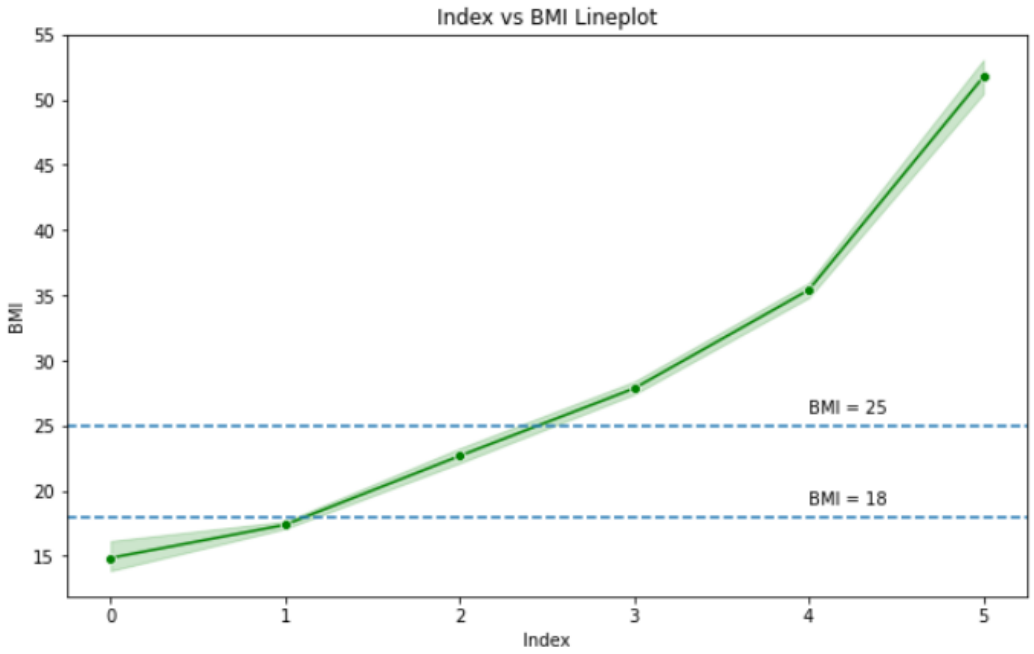
```
bmi_health.head()
```

	Gender	Height	Weight	Index
0	Male	174	96	4
1	Male	189	87	2
2	Female	185	110	4
3	Female	195	104	3
4	Male	149	61	3

Here shows the line plotting graph based on the Index and BMI of the dataset.

```
plt.figure(figsize = (10,6))
plt.title("Index vs BMI Lineplot")
sns.lineplot(x = bmi_health["Index"], y = bmi_health["BMI"], color = "green", marker = 'o')
plt.axhline(18, ls='--')
plt.axhline(25, ls='--')
plt.text(4,26, "BMI = 25")
plt.text(4,19, "BMI = 18")
```

```
Text(4, 19, 'BMI = 18')
```



Importing the KNeighborsClassifier class from the sklearn.neighbors library is a first step. For the K, this is essentially the value. There is no optimal value for K and 3 widely used value for KNN algorithm is chosen after checking and evaluation, but to start out, 3.

6.3 Abdulmalik (Linear Regression)

Linear Regression is a supervised learning based on machine learning algorithm. It plays out a regression task. Regression models an objective prediction esteem based on independent variable. It is generally utilized for discovering the connection among forecasting and variables and also for classifying. Diverse regression models contrast dependent based on the sort of connection among dependent and independent variables, they are thinking about and the quantity of autonomous dependent variable utilized. It also undertaking the task to predict a predict a dependent variable value (y) based on a given independent variable (x). regression method discovers a linear connection between x (input) and y(output). Thus, the name is Linear Regression. This technique is generally utilized for forecasting and discovering circumstances and logical results connection between variables. Regression methods generally vary dependent on the quantity of independent variable and the kind of connection between the two (2) variable.

Importing libraries

```
In [1]: import matplotlib.pyplot as plt
        import numpy as np
        import pandas as pd
```

Figure 3:inserting libraries

To run some programs or code some libraries have to import, this is because they are the files that contains the function that are going to be use in the coding part, without libraries some function will not work, it will also some and error, libraries are for some part of coding. They help with elite multidimensional cluster object and apparatuses for working with these exhibits. And furthermore. they take into consideration quick fast analysis and data cleaning and preparation. It dominates in execution and profitability. It can work with data from a wide assortment of sources.

Importing datasets

```
In [31]: df=pd.read_csv('C:\\Users\\abdul\\Desktop\\500_Person_Gender_Height_Weight_Index.csv')
```

Figure 4: importing dataset

Dataset importation is must, developer have to read his data from the particular dataset he have import, that will be use dataset to be use when the develop like to execute any kind of the data

will come from that dataset, without dataset importation the developer cannot perform any action using any data. As seen above the dataset must be called in a csv file, then the direction and name of the dataset must be mentioned.

Changing Gender columns to binary values

```
In [3]: df=pd.get_dummies(df)
        print(df)

     Height  Weight  Index  Gender_Female  Gender_Male
0       174      96      4              0            1
1       189      87      2              0            1
2       185     110      4              1            0
3       195     104      3              1            0
4       149      61      3              0            1
..      ...     ...    ...            ...          ...
495     150     153      5              1            0
496     184     121      4              1            0
497     141     136      5              1            0
498     150      95      5              0            1
499     173     131      5              0            1

[500 rows x 5 columns]
```

Figure 5: changing categorical to binary

Here we are converting the categorial variable in our dataset which is gender, it contains both male and female they will be converted to dummy/indicator variables, for example if it show 1 on male that means the data on the column is for a male. To analyze/classify the data in the dataset the categorial variable have to be quantitative, this is because the categorical variables are known to hide and mask lots of interesting information in a data set.

```
In [4]: X=df.iloc[:,[0,1,3,4]].values
        Y=df.iloc[:,2].values

In [5]: X_nu=df[["Height","Weight","Index"]]

        X_nu.corr()

        X_nu.hist(bins=50)

Out[5]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x00000226E788D30>,
                <matplotlib.axes._subplots.AxesSubplot object at 0x00000226EE76280>],
               [<matplotlib.axes._subplots.AxesSubplot object at 0x00000226EEA66D0>,
                <matplotlib.axes._subplots.AxesSubplot object at 0x00000226EED4820>]],
              dtype=object)
```

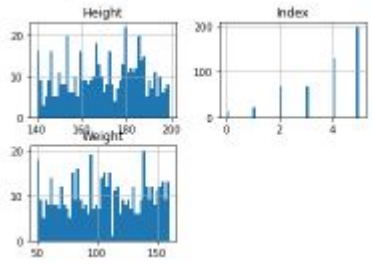


Figure 6:feature selection

The figure above is the features selection, showing the rows the we are going to use as our x and y, which it show the gender, height and weight are our x values and index as the y value, since we are analyzing/classify the obesity we have to use bmi which stand as index in the table as our y value, the index it the target variable which has five(5)status, 5-extreme obesity, 4-obesity, 3-overwight 2-normal, 1-weak. Gender, weight and height are the x variable, which there are the variable that will help us to identify the obesity/index level in each person. The 3 diagram for

height, weight and index shows the average of the height, weight and index per numbers of person in the dataset, for example if at the of the diagram show 50 and down show 10, it means 50 people have same height, weight or index which is 10.

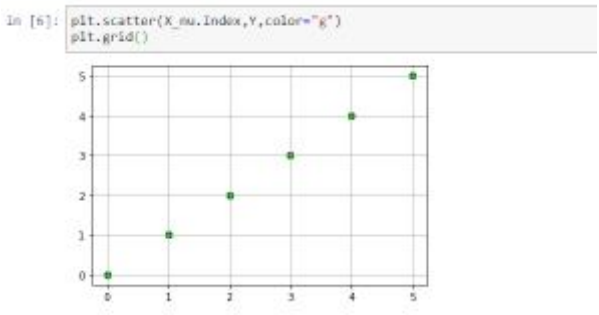


Figure 7: index diagram

The above diagram shows all the index data value of each person in the dataset, which is represented by a dot, is only use for a numeric value. with one variable on each axis, to look for a relationship between them.

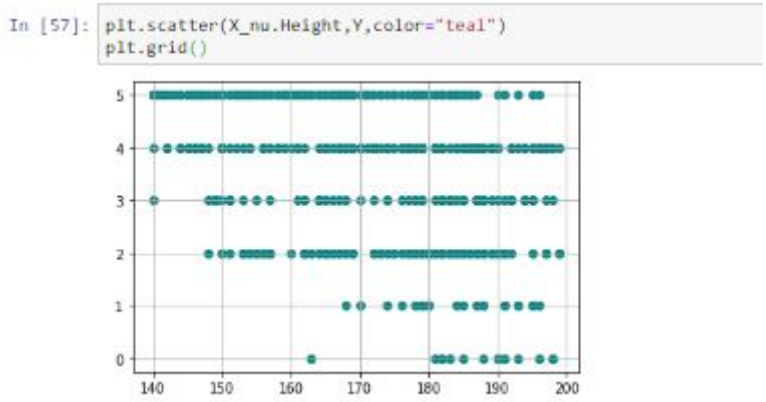
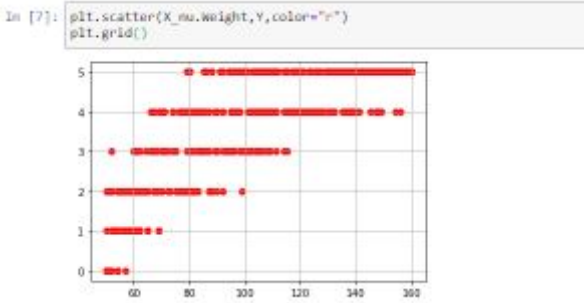


Figure 8: height diagram

The above diagram shows all the height data value of each person in the dataset, which is represented by a dot, is only use for a numeric value. with one variable on each axis, to look for a relationship between them.



The above diagram shows all the weight data value of each person in the dataset, which is represented by a dot, is only use for a numeric value. with one variable on each axis, to look for a relationship between them.

Splitting our dataset to train and test sets

```
In [8]: X_train=X[:80]
        X_test=X[20:]

        Y_train=Y[:80]
        Y_test=Y[20:]
```

Figure 9: data splitting process

The splitting of dataset into two (2) different unique use testing and splitting which is divided in train and test set. The researcher selected dataset is split our dataset into 80% and 20%. The building for testing subset your model. The testing subset is for utilizing the model on obscure data to assess the performance of the model. Each takes in data and afterward fits a model with boundaries. The boundaries fit values. At that point the model can take data and predict. The train dataset and test dataset must be similar, as a rule have similar predictors or variables.

Fitting to our model

```
In [9]: from sklearn.linear_model import LinearRegression
        teacher=LinearRegression()
        learner=teacher.fit(X_train,Y_train)
```

Figure 10: fitting the linear regression algorithm

The above figure show fitting of my selected model, which is linear regression, the obesity prediction cannot be perform without a model, before selecting the model we look into our dataset which is consist of weight, height, gender and index and found out the linear regression if a good fit to the dataset, if wrong model was select the input of the accuracy will be not good enough for the useful for practical decision making.

Making Prediction

```
In [18]: Yp=learner.predict(X_test)
         c=learner.intercept_
         m=learner.coef_
         print("c is {}  \n m is {}  \n Yp is {}".format(c,m,Yp))

c is 6.340006729975826
 m is [-0.03723183  0.03551815 -0.0426395   0.0426395 ]
 Yp is [4.45324565 5.90210164 3.45944885 4.44078876 2.282209   2.9575229
 2.69182322 1.63141989 5.88496484 5.12590761 4.99287256 1.61132486
 1.35484872 5.08215224 5.89524692 3.95794744 3.02876621 3.87171501
 5.33848322 2.29544931 4.05297525 4.59478497 2.53526578 4.58846352
 4.80546135 2.20313758 3.63999781 3.59519999 2.95462881 6.57794881
 4.84637022 1.6715458  4.2298547  3.71717754 3.21431333 2.50513083
 3.25544165 4.27518579 2.68661804 4.193092   1.11306842 6.13577467
 1.12388377 2.26560547 4.67102644 5.25373749 3.31223523 2.14699114
 3.11283579 3.9138609  2.42132335 3.92000434 3.98341049 2.35377856
 2.97029488 4.44149205 3.94228217 3.24885366 4.08677972 1.46411124
 2.94048777 5.0984246  2.96172568 3.65766788 4.58543147 0.79565206
 4.79885631 5.14865458 4.08039413 4.39101991 5.14889673 2.90268515
 3.02590725 1.96822    3.28363504 5.51778763 3.55547986 5.3715764
 1.5821923  1.00822766 4.52334369 4.3873584  5.90263401 4.08459691
 4.13039713 3.60672661 5.89486651 5.94158842 3.33233826 3.18838184
 3.2078636  6.44686959 4.11254905 2.35448984 3.1268761  3.01242479
 2.18765832 4.56440786 4.86622389 5.29439667 3.14149922 3.76250864
 4.33079897 5.13861465 5.71594251 3.91875979 3.25325884 3.71368684
 4.97869399 2.57592497 4.9884428  1.83028606 5.8580151  3.9956484
 5.44503766 4.82066502 4.77315112 3.80463182 5.24617149 4.87503684
 1.93216858 3.18769475 4.98212135 4.69159059 3.98512417 3.49674482
 2.93359552 6.27956004 2.61221853 4.1360073  0.78483671 3.02613423
 2.85268214 5.02396095 4.47794845 3.30023947 2.51369922 5.81960286
 2.96686672 3.38289111 2.49888938 2.15509841 2.83046845 4.60926983
 3.84500742 2.42310117 3.71032282 4.2631259  3.06487734 2.14823569
 4.25976268 4.8148127  5.53717137 3.66188652 4.34918832 3.64513885
 3.59980776 2.90611251 4.60117056 2.9438776  2.90315428 3.95723616
 3.80535063 2.28274227 2.99381647 3.08984667 4.88360284 2.51369922
 5.07147485 3.88324164 4.16180122 4.20975967 3.97780032 5.88543397
 4.48066453 4.82899127 3.1745924  3.87195716 5.24849718 4.56914391
 1.56552464 3.03833701 3.65931742 3.30934114 6.67593485 3.81167208
 4.50932796 3.34214321 2.69010954 3.17577282 3.78511764 6.38664864
 5.7955472  4.6563146  1.26542308 4.59260216 5.17335738 4.07480854
 5.5319662  4.34326386 2.77774936 4.47972627 3.84376287 4.73692169
 1.3100429  5.65084726 2.69743339 3.76079496 4.28190255 4.34458841
 4.24723365 3.38031329 2.51027186 2.59330391 3.76344689 2.93920569
 4.66074436 5.75370761 2.13670906 4.77751674 2.78342367 2.37777008
 1.48249259 5.2216467  1.26885044 4.81870919 3.09623226 6.66683318
 2.58341715 3.55843729 3.18144712 6.2582855  5.10752627 5.92219667
 4.79938058 2.33486395 3.77397112 4.94825275 5.16431985 4.14410657
 4.16567313 4.97573576 5.44946741 2.3879142  3.90209212 4.19651936
 4.88756267 5.80887623 5.39338511 1.91260682 5.61039048 4.39077776
 3.64171149 2.26211397 2.71238738 2.68272155 3.69893907 2.15985184
 2.65163317 1.41659733 1.51458337 5.38857524 4.17199458 4.01160479
 4.09487899 4.58970887 3.43521518 5.44728461 2.34514602 4.3959188
```

Figure 11: making prediction

The above diagram shows the after analyzing/classify, the result of the prediction that has been perform on the x test set, which the x values for the prediction are the gender, height and weight of each select person in the dataset. That help to predict the status of obesity in each person.

List conversion due to data type

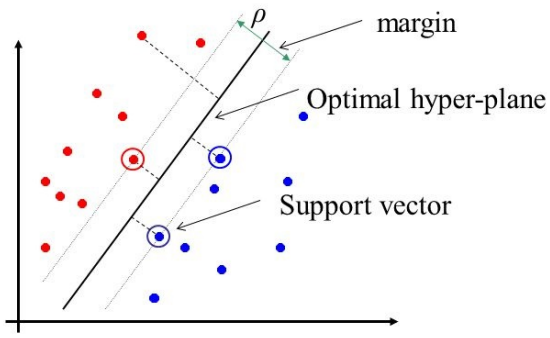
```
In [11]: xlist=list(X_train)
         ylist=list(Y_train)
         yplist=list(Yp)

In [12]: mytable=pd.DataFrame({"input":xlist,"out":ylist})
         print(mytable)

                  input  out
0     [174, 96, 0, 1]    4
1     [189, 87, 0, 1]    2
2    [185, 110, 1, 0]    4
3    [195, 104, 1, 0]    3
4     [149, 61, 0, 1]    3
..                ...  ...
75   [197, 154, 1, 0]    4
76   [165, 104, 0, 1]    4
77    [168, 90, 1, 0]    4
78   [176, 122, 1, 0]    4
79    [181, 51, 0, 1]    0

[80 rows x 2 columns]
```

Figure 12:List conversion due to data type

The figure above show the list of converted data type, the show to the of the data which is using the x values consist of height, weight and gender , which after the convert the gender categorical value to binary change to 0 and 1 the first row of the gender is female while the second

id male, the output coming it is the calcuted bmi/index of each person, which is use to analyze/ classify the obese of the person.

6.4 Madoma (SVM)



SVM is a supervised learning machine learning technique that attempts classify classes using a hyperplane (Li, 2015). A hyperplane is simply a function that segregates the features to be classified. This could be a simple line equation function for 2 features as shown above, or a more complex plane function for higher dimension datasets. SVM maximizes the margins between the closest support vectors to the hyperplane, decreasing the possibility of misclassification.

SVM is capable of classification, regression, and outlier detection. It operates using kernels which allow it to add dimensions to make it easier to segregate higher dimension data. Some kernel examples utilised include linear, polynomial, and radial basis function (Rbf) kernels. SVM is memory efficient, works better where the dimensions are greater than number of samples and can handle both linearly distinguishable and non-linearly distinguishable vectors of classes. However, SVM struggles when the dataset has more noise like the overlapping between target classes.

The implementation of this algorithm has been done in Python using a Jupyter notebook. In this process the researcher will reiterate loading the data, explore, split, and finally generate and evaluate the model. The researcher has chosen to go with SVM because we want to classify obesity based on 3 features/ dimensions. This makes it a multidimensional problem where SVM thrives.

Firstly a few libraries are imported like pandas for data processing as well as csv file I/O operations, seaborn for visualization tools and sklearn for pre-processing and calculation of metrics needed later like accuracy score.

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.metrics import accuracy_score, precision_score
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn import svm
from sklearn.model_selection import train_test_split
```

After importing the needed packages, the data was loaded and saved to the bmi_data variable the dat being used is in csv format and has been described in detail in section 3 of this paper.

```
bmi_data = pd.read_csv('../input/500_Person_Gender_Height_Weight_Index.csv')
```

The data was then explored just for reassurance by checking the top 5 information, last 5 information, no null values, the columns it contains and general information about the data.

```
#shows the top 5 information
bmi_data.head()
```

	Gender	Height	Weight	Index
0	Male	174	96	4
1	Male	189	87	2
2	Female	185	110	4
3	Female	195	104	3
4	Male	149	61	3

```
#last 5 information
bmi_data.tail()
```

	Gender	Height	Weight	Index
495	Female	150	153	5
496	Female	184	121	4
497	Female	141	136	5
498	Male	150	95	5
499	Male	173	131	5

```
# Reassuring data has no null values
bmi_data.isnull().any()
```

```
Gender    False
Height    False
Weight    False
Index     False
dtype: bool
```

```
bmi_data.columns
```

```
Index(['Gender', 'Height', 'Weight', 'Index'], dtype='object')
```

```
bmi_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Gender  500 non-null    object
 1   Height  500 non-null    int64
 2   Weight  500 non-null    int64
 3   Index   500 non-null    int64
dtypes: int64(3), object(1)
memory usage: 15.8+ KB
```

As noted in the screenshot above the Gender column is of type object. For good practice as well as being unable to apply mathematical calculations on these value a conversion of this categorical data type into numeric is done.

From the figure on the left to the right

```
bmi_data['Gender']
```

```
0       Male
1       Male
2     Female
3     Female
4       Male
       ...
495   Female
496   Female
497   Female
498     Male
499     Male
Name: Gender, Length: 500, dtype: object
```

```
gender_label = LabelEncoder()
bmi_data['Gender'] = gender_label.fit_transform(bmi_data['Gender'])
print(bmi_data['Gender'])
```

```
0      1
1      1
2      0
3      0
4      1
      ..
495    0
496    0
497    0
498    1
499    1
Name: Gender, Length: 500, dtype: int32
```

As shown the Male gender is represented by 1 and Female represented by 0.

```
bins = (-1,0,1,2,3,4,5)
health_status = ['Extremely Underweight','Underweight', 'Normal', 'Overweight', 'Obese', 'Extremely Obese']
bmi_data['Index'] = pd.cut(bmi_data['Index'], bins = bins, labels = health_status)
bmi_data['Index'].value_counts()
```

```
Extremely Obese         198
Obese                   130
Normal                   69
Overweight               68
Underweight              22
Extremely Underweight    13
Name: Index, dtype: int64
```

The code above showcases the statistics and frequency of the data. The indexes from the dataset represent the following:-

Index :

0 - Extremely Underweight

1 – Underweight

2 – Normal

3 – Overweight

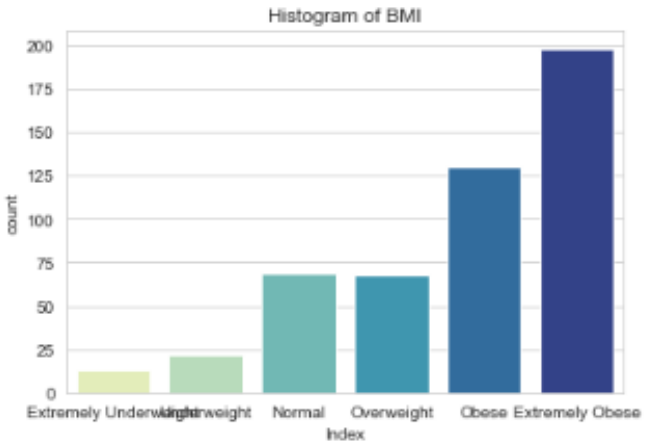
4 – Obesity

5 - Extreme Obesity

These have been taken to consideration to visualize the data better using a histogram.

```
# Set default plot grid
sns.set_style('whitegrid')

sns.countplot(bmi_data['Index'], palette='YlGnBu')
ax = plt.gca()
ax.set_title("Histogram of BMI")
```

```
Text(0.5, 1.0, 'Histogram of BMI')
```



Using an 80:20 ratio for the training and data sets the code below showcases the splitting of the data. The researcher also chooses the features needed for the independent variables, these are all columns excluding the index. The target data/dependent variable/Y is the Index column.

```
X = bmi_data.drop('Index', axis = 1)
y = bmi_data['Index']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2)


print(X_train.shape)

print(X_test.shape)

print(y_train.shape)
```

```
(400, 3)
(100, 3)
(400,)
```

There were 400 rows and 3 columns used in for the x values, 100 rows and 3 columns for the x testing data and 400 rows and 1 column of the y train data.

The SVM model is applied using the linear kernel. The y_predict variable is the prediction based on the testing component X_test.

7.0 Analysis & Recommendations

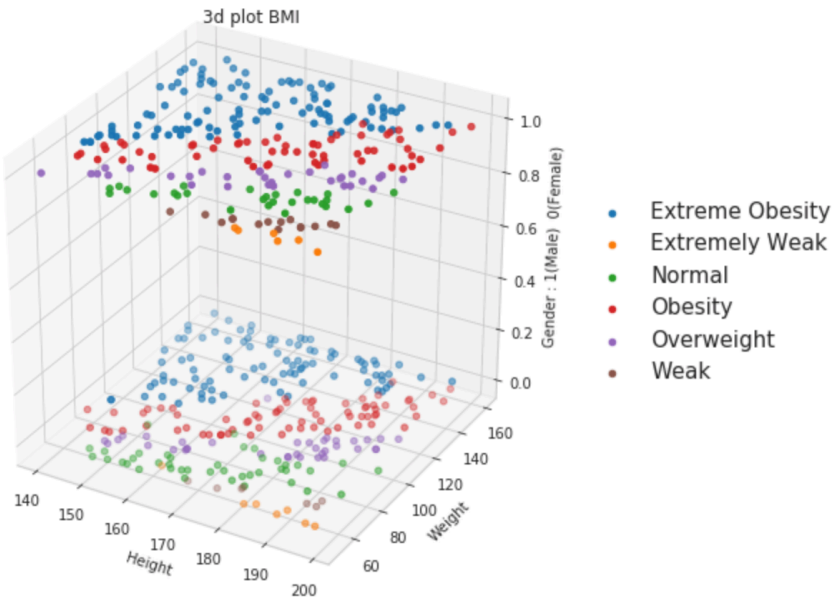
7.1 Accuracy and Error Rate

Random Forest (Wahid)

3D BMI Plot

```
groups = data_visual.groupby('Status')

from mpl_toolkits.mplot3d import Axes3D
colors = ['#e41a1c','#377eb8','#4daf4a','#984ea3','#ff7f00','#ffff33']
fig = plt.figure(figsize=(10,8))
ax = fig.add_subplot(111, projection='3d')

# ax.scatter(data_visual['Height'],data_visual['Weight'] , data_visual['gender_lbl'],
#            c=data_visual['Index'],
#            cmap=matplotlib.colors.ListedColormap(colors))


for name, group in groups:
    ax.scatter(group.Height, group.Weight, group.gender_lbl, label=name)
ax.set_xlabel('Height')
ax.set_ylabel('Weight')
ax.set_zlabel('Gender : 1(Male)  0(Female)')
ax.set_title('3d plot BMI')

box = ax.get_position()
ax.set_position([box.x0, box.y0, box.width * 0.8, box.height])
ax.legend(loc='center left', bbox_to_anchor=(1, 0.5), prop={'size': 15})
plt.show()
```

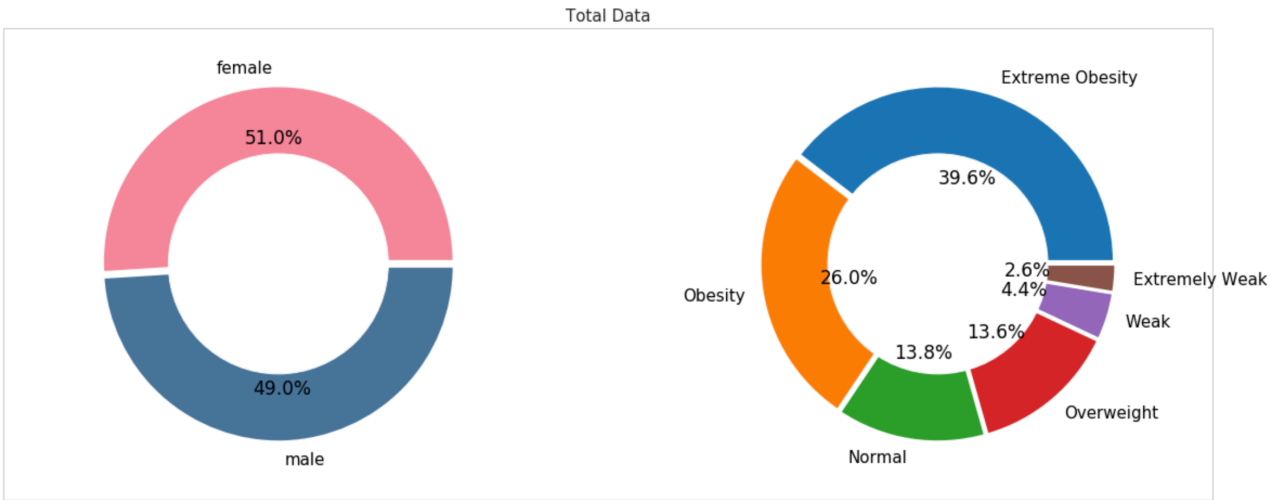
This code represents the three dimensional plotting of BMI for both Male & Female. Pyplot has been used to plot the 3D graph. The following output shows the excited code and the classification of the BMI.



Total BMI Percentage & Gender Percentage

```
fig = plt.figure(figsize=(20,8))
plt.title("Total Data",loc='center',weight=10,size=15)
plt.xticks([]) # to disable xticks
plt.yticks([]) # to disable yticks

# first pie-plot
ax1 = fig.add_subplot(121)
ax1.axis('equal')
explode = (0.01,)*(len(people))

wedges, texts, autotexts =  ax1.pie(people,
                                    radius=0.8,
                                    explode=explode,
                                    labels=['female','male'],
                                    colors=['#f7879a','#49759c'],
                                    autopct="%1.1f%%",
                                    pctdistance=0.7,
                                    textprops=dict(color='k'),
                                    wedgeprops = { 'linewidth' : 3, 'edgecolor' : 'w' }
                                    )
plt.setp(autotexts,size=17)
plt.setp(texts,size=15)
my_circle = plt.Circle((0,0),0.5,color='white')
p = plt.gcf() # get current figure reference
p.gca().add_artist(my_circle) # get current axes

# Second pie-plot
ax2 = fig.add_subplot(122)
ax2.axis('equal')
explode = (0.01,)*(len(categories))
wedges2, texts2, autotexts2 =  ax2.pie(categories,
                                    radius=0.8,
                                    explode=explode,
                                    labels=['Extreme Obesity','Obesity','Normal','Overweight','
Weak','Extremely Weak'],
                                    autopct="%1.1f%%",
                                    pctdistance=0.5,
                                    textprops=dict(color='k'),
                                    wedgeprops = { 'linewidth' : 3, 'edgecolor' : 'w' }
                                    )
plt.setp(autotexts2,size=17)
plt.setp(texts2,size=15)
my_circle = plt.Circle((0,0),0.5,color='white')
p = plt.gcf() # get current figure reference
p.gca().add_artist(my_circle) # get current axes
```

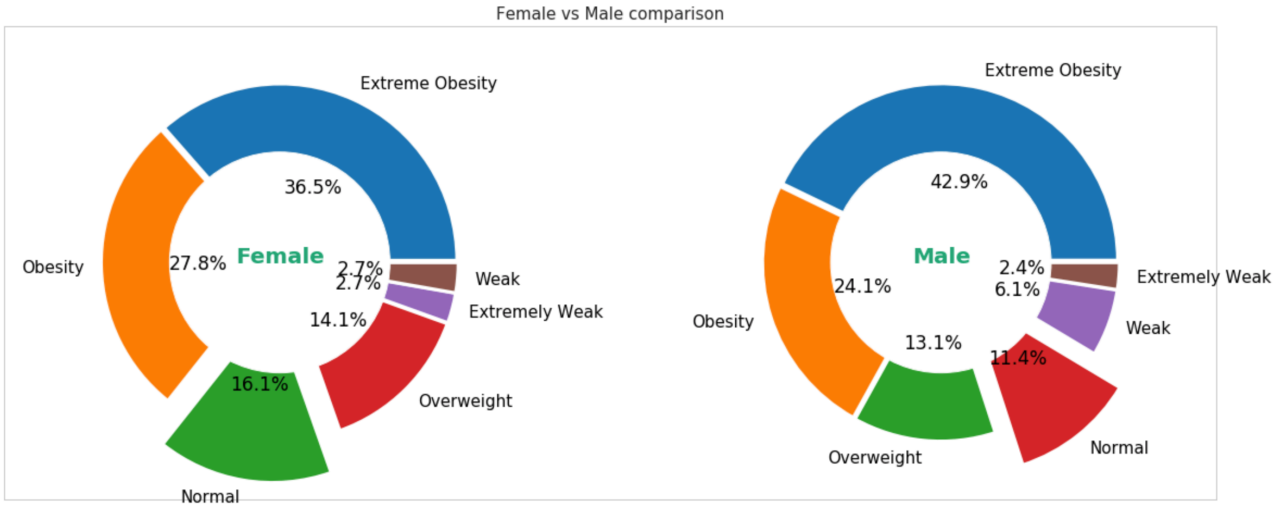
This code shows the total percentage of BMI based on the classification and total percentage of gender in the dataset. The following figure is the output of the code.



Male vs Female BMI Comparison

```
fig = plt.figure(figsize=(20,8))
plt.title("Female vs Male comparison",loc='center',weight=10,size=15)
plt.xticks([]) # to disable xticks
plt.yticks([]) # to disable yticks

# first pie-plot
ax1 = fig.add_subplot(121)
ax1.axis('equal')
explode = (0.01,0.01,0.2,0.01,0.01,0.01)

wedges, texts, autotexts =  ax1.pie(data_visual_female_categories,
                                    radius=0.8,
                                    explode=explode,
                                    labels=['Extreme Obesity','Obesity','Normal','Overweight','
Extremely Weak','Weak'],
                                    autopct="%1.1f%%",
                                    pctdistance=0.45,
                                    textprops=dict(color='k'),
                                    wedgeprops = { 'linewidth' : 3, 'edgecolor' : 'w' }
                                    )
plt.setp(autotexts,size=17)
plt.setp(texts,size=15)
my_circle = plt.Circle((0,0),0.5,color='white')
p = plt.gcf() # get current figure reference
p.gca().add_artist(my_circle) # get current axes
ax1.text(0,0,'Female',size=20,color='#1fa774',horizontalalignment='center',weight='bold')

# Second pie-plot
ax2 = fig.add_subplot(122)
ax2.axis('equal')
explode = (0.01,0.01,0.01,0.2,0.01,0.01)
wedges2, texts2, autotexts2 =  ax2.pie(data_visual_male_categories,
                                    radius=0.8,
                                    explode=explode,
                                    labels=['Extreme Obesity','Obesity','Overweight','Normal','
Weak','Extremely Weak'],
                                    autopct="%1.1f%%",
                                    pctdistance=0.45,
                                    textprops=dict(color='k'),
                                    wedgeprops = { 'linewidth' : 3, 'edgecolor' : 'w' }
                                    )
plt.setp(autotexts2,size=17)
plt.setp(texts2,size=15)
my_circle = plt.Circle((0,0),0.5,color='white')
p = plt.gcf() # get current figure reference
p.gca().add_artist(my_circle) # get current axes
ax2.text(0,0,'Male',size=20,color='#1fa774',horizontalalignment='center',weight='bold')
```

This code shows the Male vs Female BMI comparison based on the classification in the dataset. The following figure is the output of the code.



Predicting Accuracy

```
from sklearn.metrics import accuracy_score
rfc_acc = accuracy_score(y_test, y_pred_rfc)
rfc_acc*100
```

```
91.0
```

This code shows how to predict and represent the accuracy from the classification. Random Forest algorithm has **91.0 % Accuracy**.

Number of Trees vs Accuracy Implementation

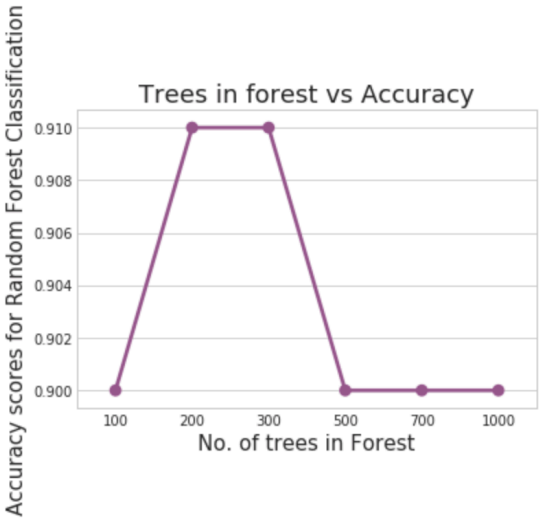
```
def trees_in_forest_vs_acc(trees, X_train=X_train, y_train=y_train, X_test=X_test, y_test=y_test):
    rfc = RandomForestClassifier(n_estimators=trees, criterion='entropy', random_state=0)
    rfc.fit(X_train, y_train)
    y_pred = rfc.predict(X_test)
    acc = accuracy_score(y_test, y_pred)
    return acc
```

```
trees_list_for_randomForest = [100, 200, 300, 500, 700, 1000]
acc_scores_for_trees_RFC = []
for x in trees_list_for_randomForest:
    acc_scores_for_trees_RFC.append(trees_in_forest_vs_acc(x))
acc_scores_for_trees_RFC
```

```
[0.9, 0.91, 0.91, 0.9, 0.9, 0.9]
```

This code uses the random forest classifier and predict the accuracy based on the ascending number of trees. The output the shows the value of the accuracy.

Number of Trees vs Accuracy Plot



This shows the plot of number of trees vs the accuracy of Random Forest Algorithm. We can see that as the number of trees increases from 100 to 200, the Accuracy increases. Then it stays constant till 300 number of tree. Afterwards the Accuracy declines as the number of trees reduces.

KNN (Sylvia)

```
# Now lets to predict Index value from Height and Weight using KneighboourClassifier
X = bmi_health[["Height", "Weight"]]
X = np.array(X)
Y = bmi_health["Index"]
```

```
x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size = 0.1)
```

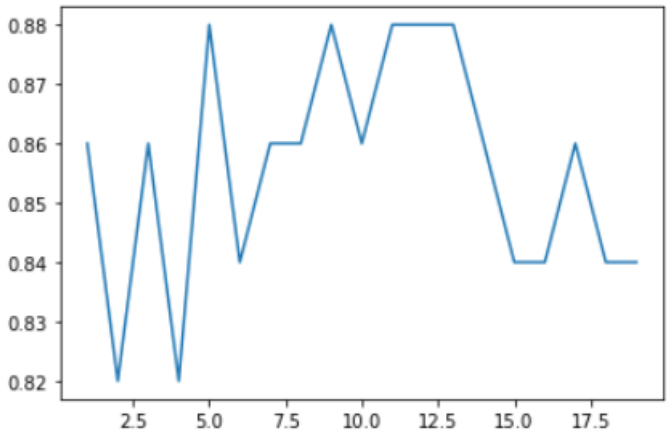
```
X_train=X[:80]
X_test=X[20:]

Y_train=Y[:80]
Y_test=Y[20:]
```

```
KNC = KNeighborsClassifier(n_neighbors=3)
KNC.fit(x_train, y_train)
```

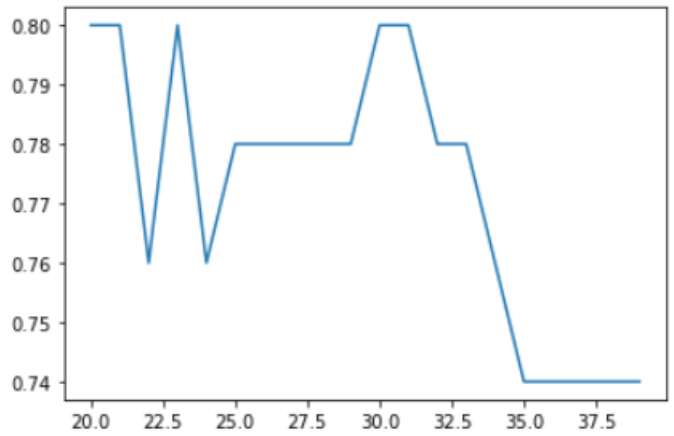
```
KNeighborsClassifier(n_neighbors=3)
```

For the expected test value set for all K values between 1 and 20, another group 20 to 40, the mean score was plotted.

```
from sklearn.neighbors import KNeighborsClassifier
score_list=[]
score_list2=[]
for i in range(1,20):
    knn = KNeighborsClassifier(n_neighbors=i)
    knn.fit(x_train,y_train)
    score_list.append(knn.score(x_test,y_test))
plt.plot(range(1,20),score_list)
plt.show()
```



```
score_list2=[]
for i in range(20,40):
    knn2 = KNeighborsClassifier(n_neighbors=i)
    knn2.fit(x_train,y_train)
    score_list2.append(knn2.score(x_test,y_test))
plt.plot(range(20,40),score_list2)
plt.show()
```



The script above performs a loop from 1 to 20 and from 20 to 40. The mean score for the expected test set values is determined for each iteration and the outcome is appended to the score list. The final result of the K-neighbors Classifier had shown below.

```
knn3 = KNeighborsClassifier(n_neighbors=33)

knn3.fit(x_train,y_train)
print(knn3.score(x_test,y_test))
```

```
0.78
```

Linear Regression (Abdulmalik)

Error rate

```
In [101]: from sklearn.metrics import mean_squared_error,accuracy_score
          Error=mean_squared_error(Yp,Y_test)
          np.sqrt(Error)

Out[101]: 0.5660854728613318
```

Figure 13: error rate

The above figure shows the error rate of the incorrectly predicted values for the test set in the dataset.

```
In [25]: import seaborn as sns
         sns.barplot(x=Y_test,y=Yp,data=df)

Out[25]: <matplotlib.axes._subplots.AxesSubplot at 0x2026f0ff130>
```

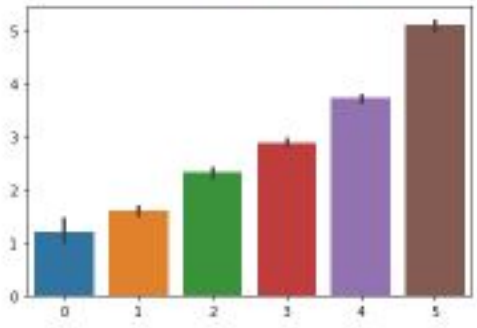


Figure 14:data frame

The above figure helps you explore and understand your data. Its plotting functions operate on data frames and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots.

Predicting the accuracy

```
In [26]: y_pred_cls=np.zeros_like(Yp)
         y_pred_cls[Yp>2.5]=1

         y_test_cls=np.zeros_like(Yp)
         y_test_cls[Y_test>2.5]=1

In [27]: mytable=pd.DataFrame({"input":xlist,"out":ylist})
         print(mytable)

                  input  out
0     [174, 96, 0, 1]    4
1     [189, 87, 0, 1]    2
2    [185, 110, 1, 0]    4
3    [195, 104, 1, 0]    3
4     [149, 61, 0, 1]    3
..                ...  ...
75   [197, 154, 1, 0]    4
76   [165, 104, 0, 1]    4
77    [168, 90, 1, 0]    4
78   [176, 122, 1, 0]    4
79    [181, 51, 0, 1]    0

[80 rows x 2 columns]
```

Figure 15:predicting process

The above figure show the how the predicting process undergo, we are using our y which is the index/bmi to predict the accuracy of our dataset which we are getting by analyzing/classify the x values in the dataset for each person in the test set.

Accuracy rate

```
In [108]: print(accuracy_score(y_test_cls,y_pred_cls))
          0.9625
```

Figure 16: accuracy rate

This is the accuracy rate of the dataset, after analysis/classification for the people with the obesity regarding their data which is the x values are index, height and weight and we use age as the y value. For the linear regression the accuracy is only the ways to measure how frequently the algorithms classifies the data point effectively. Is also the quantity of effectively and data point predicted correctly out of the all data point.

SVM (Madoma)

```
classifier = svm.SVC(kernel='linear')
#train the model
classifier.fit(X_train, y_train)
#predict the response
y_predict = classifier.predict(X_test)
```

Finally, the researcher has evaluated the results by comparing the predicted value to the actual value using the classification_report function imported from the sklearn.metrics package.

```
#compare predicted vs actual value

print(classification_report(y_test,y_predict))
```

	precision	recall	f1-score	support
0	1.00	0.67	0.80	3
1	1.00	1.00	1.00	3
2	0.67	1.00	0.80	8
3	0.92	0.75	0.83	16
4	0.94	0.94	0.94	32
5	0.97	0.97	0.97	38
accuracy			0.92	100
macro avg	0.92	0.89	0.89	100
weighted avg	0.93	0.92	0.92	100

```
print("accuracy:", accuracy_score(y_test,y_pred = y_predict))
```

```
accuracy: 0.92
```

Recall Index : 0 - Extremely Underweight 1 - Underweight 2 - Normal 3 - Overweight 4 - Obesity 5 - Extreme Obesity. In the diagram above the support denotes how many instances of each index class were found. The precision score obtained by ($\frac{True\ Positive}{Total\ Predicted\ Positive}$), Recall = $\frac{True\ Positive}{True\ Positive + False\ negative}$ and f1 = $2 * \frac{Precision*Recall}{Precision + Recall}$.

```
print(confusion_matrix(y_test, y_predict))
```

```
[[ 2  0  1  0  0  0]
 [ 0  3  0  0  0  0]
 [ 0  0  8  0  0  0]
 [ 0  0  3 12  1  0]
 [ 0  0  0  1 30  1]
 [ 0  0  0  0  1 37]]
```

Using the linear and polynomial kernels the accuracy of the model was 92 %. If using the rbf kernel the accuracy dropped by 14% to 78%. Therefore, the linear and polynomial kernel yielded better output compared to the default rbf kernel. The accuracy given by the former kernels is very good.

7.2 Most Suitable Algorithm

Algorithm Name	Accuracy
Linear Regression	96.0
SVM	92.0
Random Forest	91.0
KNN	86.0

Based on the accuracy table we can conclude that, the most suitable algorithm for predicting obesity is **Linear Regression** with an accuracy of 96.0. The second most optimized algorithm for prediction is SVM with an accuracy of 92.0. Random Forest is also one of the most optimized algorithm for predicting obesity with an accuracy of 91.0.

8.0 Conclusion

This project was a new endeavor by the researcher into both the analysis/classification domain and the data mining processes. Numerous regions have been uncovered all through this project execution stage, and the data pre-preparing method is finished gratitude to the knowledge procured during the lab and lecturer classes. With the correct guidance of the requirements, all through the data cleaning and preprocessing in this analysis is proficient in delivering quickly and straightforwardly. Even though the researcher has met different challenges in getting an appropriate

dataset, it contains classified individual data. The researcher also endeavors to search for many websites, books, and articles for information and materials. The researcher has implemented linear regression. Finally, the researcher has concluded that the job level will be the most dependable factor to affect the monthly income as the linear regression value is highest among the other elements. The researcher uses linear regression as his selected algorithm. The researcher concludes to analyze or classify obesity that linear regression will be one of the best algorithms to use.

Acknowledgement

I would like to thank Dr. Booma Poolan Marikannan, the study's supervisor, for the vital role she played in promoting the best results of the investigation. Dr. Booma Poolan Marikannan tried hard to lead us towards our target and continue to step down the right path. We are very grateful for her encouragement and advice on all the different topics. She does not hesitate to help us accomplish our objectives in the light of her efforts and the extra time she has given us. Her passion, her inspiration, and her trust helped. She was always happy to answer our questions and kindly gave us her time and tremendous knowledge. We affirm that this work is our responsibility and has earned the recognition of information services.

References

1. Azrar, A., Awais, M., Ali, Y. & Zaheer, K., 2018. Data Mining Models Comparison for Diabetes Prediction. International Journal of Advanced Computer Science and Applications,, 9(8), pp. 320-323.
2. Babajide, O., Hissam, T., Anna, P., Anatoliy, G., Astrup, A., Martinez, J.A., Oppert, J.M. and Sørensen, T.I., (2020). A Machine Learning Approach to Short-Term Body Weight Prediction in a Dietary Intervention Program. In International Conference on Computational Science (pp. 441-455). Springer, Cham.
3. Fiarni, C., M, E., Sipayung & Maemunah, S., 2019. Analysis and Prediction of Diabetes Complication Disease using Data Mining Algorithm. The Fifth Information Systems International Conference 2019 , p. 449–457.
4. Kim, C. and Youm, S., 2020. Development of a Web Application Based on Human Body Obesity Index and Self-Obesity Diagnosis Model Using the Data Mining Methodology. Sustainability, 12(9), p.3702.

5. Li, R., (2015). Top 10 Data Mining Algorithms, Explained. KDnuggets. Available at: <https://www.kdnuggets.com/top-10-data-mining-algorithms-explained.html/> [Accessed 1 Oct. 2020].
6. Niklas Donges, N., 2019. A Complete Guide To The Random Forest Algorithm. [online] Built In. Available at: <https://builtin.com/data-science/random-forest-algorithm> [Accessed 2 October 2020].
7. Raja, J. and Pandian, S., 2020. PSO-FCM based data mining model to predict diabetic disease. Computer Methods and Programs in Biomedicine, 196, p.105659.
8. Sisodia, D. and Sisodia, D.S., 2018. Prediction of diabetes using classification algorithms. Procedia computer science, 132, pp.1578-1585.
9. Triantafyllidis, A., Polychronidou, E., Alexiadis, A., Rocha, C., Oliveira, D., da Silva, A., Freire, A., Macedo, C., Sousa, I., Werbet, E., Lillo, E., Luengo, H., Ellacuría, M., Votis, K. and Tzovaras, D., 2020. Computerized decision support and machine learning applications for the prevention and treatment of childhood obesity: A systematic review of the literature. Artificial Intelligence in Medicine, 104, p.101844.
10. Vorhies, W., (2016). CRISP-DM – a Standard Methodology to Ensure a Good Outcome. [online] Available at: <https://www.datasciencecentral.com/profiles/blogs/crisp-dm-a-standard-methodology-to-ensure-a-good-outcome> [Accessed 1 Oct. 2020].
11. Wu, H., Wylie-Rosett, J. and Qi, Q., (2013). Dietary Interventions for Weight Loss and Maintenance: Preference or Genetic Personalization? Current Nutrition Reports, 2(4), pp.189–198.
12. Zhang, S., Tjortjis, C., Zeng, X., Qiao, H., Buchan, I. and Keane, J., 2018. Comparing data mining methods with logistic regression in childhood obesity prediction. Information Systems Frontiers, 11(4), pp.449-460.