Date of Submission June 27, 2024.
Digital Object Identifier Not Applicable

A Decision Tree Classifier Model for Obesity Level Classification (June 2024)

Pragyan Bhattarai [1], Prashant Raj Bista[1]
bhattaraipragyan.pb@gmail.com, prashant.bista.18@gmail.com

[1]Department of Electronics and Computer Engineering, IOE, Thapathali Engineering Campus, Kathmandu, 44600 Nepal

ABSTRACT Obesity is a major growing problem in the world. The main causes of obesity include lifestyle. By keeping, such a problem in mind, an analysis tool known as a Decision Tree was used for classifying whether a certain person is obese or not. The Decision Tree classifier is a supervised learning technique widely used for classification problems. This article presents a Decision Tree model for the classification of obesity levels based on eating habits and physical condition in individuals from Colombia, Peru, and Mexico. The model was able to obtain an accuracy of 96.00%, a precision, recall, and F1-score of 0.96 when entropy was used as the criterion. The same model when Gini was used as a criterion the accuracy decreased to 94.00% with precision, recall and F1-score being 0.94.

INDEX TERMS Decision Tree, Entropy, Gini Index, Obesity

I. INTRODUCTION

Obesity has become a major public health concern globally, leading to numerous health complications such as diabetes, cardiovascular diseases, and increased mortality rates. The World Health Organization (WHO) has classified obesity as an epidemic due to its rapidly increasing prevalence and significant impact on health and well-being. Understanding the factors that contribute to obesity and accurately estimating obesity levels based on these factors is crucial for developing effective intervention and prevention strategies. Recent advancements in machine learning have provided powerful tools for analyzing complex datasets and uncovering patterns that may not be immediately apparent through traditional statistical methods. One such tool is the decision tree algorithm. This study utilizes a comprehensive dataset from the UCI Machine Learning Repository titled "Estimation of Obesity Levels Based on Eating Habits and Physical Condition" [1].

A Decision Tree is a supervised traditional machine-learning algorithm used for classification and regression tasks. The decision tree has a hierarchical structure i.e. it has a root node at the highest, an internal node (child node) at the middle, and a leaf node at the bottom. A decision tree is a non-parametric type of machine learning algorithm meaning it does not use a fixed set of parameters for making decisions.

A decision tree algorithm is used for the task of classification as well as regression. This article will be mainly focused on the classification task of the Decision Tree. Decision Tree starts from the root node and divides the data of the root node based on certain criteria. The criteria can be given by entropy, Gini, or information gain. Based on the given criteria, the internal node gets divided till it reaches the leaf node. The leaf node is pure i.e. it contains the data of a single class only. A decision tree algorithm is one of the simplest algorithms yet can break complex data into manageable parts. A decision tree is useful in handling the combination of numerical as well as non-numerical data.

A decision tree can handle a variety of data types i.e. discrete or continuous values. Also, it can handle continuous values that can be converted to categorical values.

A. PROBLEM STATEMENT

Obesity has become a major public health concern globally, leading to numerous health issues such as diabetes, cardiovascular diseases, and increased mortality rates. Understanding and predicting obesity levels based on lifestyle and physical condition is essential for creating effective prevention and intervention programs. The specific problem this study addresses is the estimation of obesity levels based on individuals' eating habits and physical conditions using a decision tree classifier. Accurate estimation of obesity levels can aid in the early identification of at-risk individuals and the implementation of personalized health recommendations. For this study, a dataset that contains attributes about dietary patterns, frequency of physical activities, and various health indicators was used.

B. OBJECTIVES

The main objectives of this study are:

- To predict the obesity levels based on eating habits and physical conditions in individuals using decision tree model

II. LITERATURE REVIEW

O. Iparraguirre-Villanueva et al. [2] have utilized the `CRISP-DM (Cross Industry Standard Process for Data Mining) framework for predicting obesity in nutritional patients. Their work analyzes the physical and dietary habits of the nutritional patients to predict obesity using the Decision Tree (DT) model. During the experiment, the authors have performed Exploratory Data Analysis. The histogram for the class label, the correlation between age and class label, and the correlation matrix of variables were visualized during EDA. Additionally, the minimum, maximum, mean, median, and standard deviation values of the variables were also calculated, to determine if scaling, balancing or transformation techniques were required. In this work, the dataset was divided into 80% for training and validation, and 20% for testing. Cross-validation of the generated Decision Tree was also performed to estimate the optimal depth of the tree and increase the efficiency of the model. The work has obtained an accuracy rate of 92.89%, a sensitivity rate of 94%, and an F1 score of 93% on a dataset collected from the Kaggle platform. In detail, the accuracy of 92% for underweight classes, 94% for normal weight, 98% for overweight, 90% for obesity I, 88% for obesity II, and 95% for obesity III was obtained in this paper. The paper also recommends comparing the DT method with other ML models like Support Vector Machines, Neural Networks, and Random Forest.

In the study by E. De-La-Hoz-Correa et al. [3], the authors used the SEMMA (Sample, Explore, Modify, Model, and Assess) data mining methodology. They have selected three models: Decision Trees (J48), Bayesian networks (Naive Bayes), and Logistic Regression (Simple Logistic). The study used a dataset of 712 records (324 men and 388 women) collected from 18 to 25-year-olds from Colombia, Mexico, and Peru. The dataset had 18 variables and 6 target classes. The implemented techniques resulted in a Precision rate of 97.4% for J48, 90.1% for Naive Bayes and 90.4% for Simple Logistics. The Decision Tree technique also shows a TP Rate of 97.8% and an FP Rate of 0.2%.

III. METHODOLOGY

In this section, the method used during the process of model training and testing are described.

A. THEORETICAL BACKGROUND

1) Terminologies

- Root node

The topmost node from which the decision tree starts is said as the root node. Root node has no parent node. The root node represents the entire dataset that later gets divided into two or more homogeneous sets.

- Internal/Decision node

Internal nodes (also known as decision nodes) are those nodes in the decision tree which are the decision points on attributes in the tree. These nodes have at least one child node.

- Leaf/Terminal node

Leaf nodes (also known as terminal nodes) are the nodes that represent the final decision or the prediction. The leaf node has no other child node.

- Parent node

A parent node is a node that has one or more child nodes. In the decision tree, the parent node is at one level higher hierarchy than the child node directly connected to it.

- Child node

A child node is a node that has a parent. The child node is connected directly to a parent node and is at one level lower in the hierarchy of the decision tree.

2) Types of Decision Tree Algorithms

- ID3 (Iterative Dichotomiser 3)

Iterative Dichotomiser 3 uses entropy and information gain to split data for generating a decision tree. It is suitable to use ID3 algorithm for simple datasets with categorical attributes and no missing values.

- C4.5

C4.5 is an extension of ID3 algorithm. Mitigating some drawbacks of ID3, it can handle continuous attributes and missing values. Hence, it can be used for datasets with continuous attributes and missing values. It also offers better accuracy as compared to ID3.

- CART (Classification and Regression Trees)

Classification and Regression Trees generate binary decision trees for classification and regression problems using the Gini index. It is capable of handling continuous and categorical data. As the name suggests, it is suitable for both classification and regression tasks.

- SLIQ (Supervised Learning in Quest)

SLIQ (Supervised Learning in Quest) is a Decision Tree Algorithm that is best suited for large data where efficient memory usage and speed are critical. It constructs decision trees using a pre-sorting technique and a breadth-first tree growth strategy to handle large datasets efficiently.

- SPRINT (Scalable Parallelizable Induction of Decision Trees)

SPRINT is a decision tree algorithm ideally used for very large datasets and parallel computing environments. It is preferred where scalability and distributed processing are required.

B. BLOCK DIAGRAM

The block diagram Figure 1 shows the overall process conducted during the process of development of the model. The dataset was obtained from the repository of the UCI. The dataset was then pre-processed by removing null values and encoding the categorical field. The processed data was then divided into two sets: training data and testing data in the ratio 4:1. The training data was then passed to the Decision Tree algorithm. The algorithm decides the best attributes using the mentioned criteria. The best attributes is used to divide the dataset into smaller subset. The partition is done recursively until all the leaf node contains the data of same type. The remaining test data was then passed through the trained model. The trained model evaluated and assessed the model and gives various metrics such as accuracy, precision, recall, f1-score.

C. FLOWCHART

The flowchart Figure 3 depicts the overall flow of the system. The system starts by pre-processing the data which includes removing the null values, encoding the categorical data, removing columns that are not relevant etc. The pre-processed data is split into training data and testing data. The training data is passed to the decision tree for training. After training of the decision tree, the test data is fed into the tree for testing the system. The test data when fed is first checked by the decision tree whether they belong to the same class or not. If all the data belong to the same class then the code is considered as a leaf node and labeled a class. If all the data in the nodes do not belong to the same class then according to the split criteria chosen, the node is split into two parts. The split two parts are further checked if they are pure or not. The same cycle continues until all the nodes are pure or certain stopping criteria that were set before met.

D. ALGORITHMIC STEPS

The generalized algorithmic steps for a decision tree classifier are as below:

1) Begin the decision tree by placing the entire dataset in the root node.
2) Apply the Attribute Selection method to find the best splitting criterion (attribute).
3) For the best attribute, divide the parent node into subsets and generate the child node.
4) For each child node, check whether the tuples in a child node are of the same class or not.
    a) If yes, assign the child node as the leaf node with its class label.
    b) Otherwise, recursively make sub-trees following from step 2.
5) Continue these steps until no further classification of nodes is possible.

E. MATHEMATICAL FORMULAE

1) Entropy

Entropy is the measure of the information or randomness of the random variable. According to the information theory, the information content in a highly likely event is low and the information content in a highly unlikely event is high. Entropy measures the expected amount of information conveyed by identifying the outcome of a random trial. The value of entropy ranges from 0 to 1. The entropy of the event is 0 when the probability of the event is 0 or 1 i.e. event has a high chance of occurrence or no chance of occurrence. The higher the entropy higher the uncertainty and vice-versa. The formula for entropy is given in Equation (1)

2) Gini Index

The Gini index or Gini coefficient is a metric that is used to measure the impurity or diversity of a dataset. A low Gini index indicates that the node is pure i.e. node has data points of the same class. A high Gini indicates that the instances in the node are distributed among the classes. In a decision tree, the Gini index can also be used for splitting the dataset by choosing the attribute that results in the lowest Gini index. The formula for the Gini index is given in Equation (2).

3) Classification Error

Classification error is a metric that is used to evaluate the performance of a classification model. In the context of the decision tree, classification error is used to evaluate the quality of the splits. The classification error does not take into consideration the certainty of the model about the prediction. It is a less commonly used metric for splitting. The formula for the Classification error is given in Equation (3).

4) Information Gain

Information gain is the amount of entropy reduced when a set with attributes is partitioned. It is commonly used in the construction of a decision tree by choosing the attribute that provides the highest information gain as the decision node. The calculation of the information gain is based on the entropy. The features that perfectly partition the dataset have maximum information gain and unrelated attributes have no information. The formula for information gain is given in Equation (4).

5) Information Gain Ratio

The Information Gain Ratio is an extension of information gain that addresses a bias in information gain. While information gain tends to favor attributes with many distinct values, the gain ratio normalizes the information gain by the intrinsic information of a split, making it more suitable for selecting attributes in decision tree algorithms. The formula for Information Gain Ratio is given in Equation (5).

F. PERFORMANCE METRICS

1) Accuracy

Accuracy is defined as the ratio of correctly predicted instance (True Positive and True Negative) to the total instances. It indicates the overall efficiency of the model and is best for balanced datasets. The equation for accuracy is given in equation (7).

2) Precision

The fraction of correctly predicted positive instances (True Positive) to the total of positively predicted instances (True Positive and False Positive) is referred to as Precision. High precision is important since false positives can be expensive or dangerous. The equation for precision is given in equation (8)

3) Recall (Sensitivity or True Positive Rate)

The fraction of correctly predicted positive instances (True Positive) to the total of actual positive instances (True Positive and False Negative) is termed as Recall. It measures the ability of the model to find all the relevant cases within a dataset. It is essential when the cost of False Negatives is high. The equation for recall is given in Equation (9).

The F1 score refers to the harmonic mean of precision and recall. The value of the F1 score lies between 0 and 1. It assesses the balance between precision and recall, especially in imbalanced datasets. The equation for F1 score is given in equation (10).

G. DATA PRE-PROCESSING PIPELINE

The data pre-processing techniques include removing null values in the data, converting the continuous data into categorical data, and encoding the data in machine-readable form. In the given dataset, there were no null values. The correlation was checked among the numerical features of the data that can be seen using the *following heatmap.* The heatmap shows that there is a low correlation among the data points. The categorical data that contains two types of attributes are encoded using the label encoder. The categorical data that contains more than two types of attributes are encoded using the one-hot encoding. The label encoder is useful when there is inherited order among the data and one-hot encoding is useful for nominal categories of data. The label encoder when used to encode the data encodes data in such a way that there is order among the data. The one-hot encoding does not make such assumptions but increases columns in the data. The data so formed was split into training and testing set in the ratio of 80:20, 80% for training and 20% for testing.

H. SOFTWARE USED

- Python

A high-level general-purpose programming language widely used for scientific computation due to its ease of understanding. It has a large collection of libraries for different tasks.

- Numpy

A python-library that is widely used for the computation of multi-dimensional matrixes or arrays.

- Pandas

A Python library widely used for the manipulation and analysis of data. Pandas are widely used for cleaning, exploring, and manipulating data.

- Matplotlib

A Python library widely used for plotting the numerical value.

- Scikit-Learn

A free and open-source machine learning library for the Python programming language.

- Seaborn

A Python data visualization library based on matplotlib.

IV. DATASET EXPLORATION

The dataset [1] presented by F. M. Palechor and A. de la H. Manotas consists of data from the countries of Mexico, Peru, and Columbia, based on their eating habits and physical condition. This dataset can be used for the estimation of obesity levels in individuals. So, it is associated with the area of Health and Medicine. The dataset contains 17 attributes, 2111 records, and 7 class labels. The Class Labels include Insufficient Weight, Normal Weight, Overweight Level I, Overweight Level II, Obesity Type I, Obesity Type II, and Obesity Type III. These data were labeled using the equation (11). 23% of the total data in the dataset was collected directly from users of the age between 14 and 61, using a web platform. From the collected data of 485 records, it was observed that the distribution of data was quite unbalanced. The population with the 'Normal' class label (Mass Body Index from 18.5 to 24.9) was over 280 and the rest of the class label was below 60 in number. So, there was a balancing class problem. In order to overcome this problem, 77% of the data was synthesized by using the tool Weka and the filter SMOTE (Synthetic Minority Over-Sampling Technique). After summing up the collected data and the synthetic data, the final result was 2111 records.

The dataset consists of 17 features including the class attribute. The Table 4 shows the details about the dataset features. Out of 17 features, 9 are categorical and remaining 8 are continuous data. All features contain 2111 data points with none of the features having missing values. Here NObeyesdad is the target feature that needs to the predicted based on the other given features.

V. RESULT

A. CONFUSION MATRIX

The confusion matrix Figure 7 shows the correct prediction made by the algorithm along the diagonal. The algorithm was able to correctly predict 50 instances of insufficient weight category, 46 instances of normal weight, 63 instances of type I obesity, 54 instances of type II obesity, 73 instances of type III obesity, 68 instances of level I overweight, and 52 instances of level II overweight. In total, the algorithm classified a total of 406 instances correctly out of 423 data that was used for testing the algorithm. The algorithm was not able to classify 17 data points. The algorithm misclassified 6 instances of normal weight with insufficient weight, 1 insufficient weight label with normal weight, one overweight level 1 instance with normal weight instance. The algorithm was also confused for two instances in obesity type II and obesity type, 1 instance for obesity type III and type I, one instance for overweight level II and obesity type I.

The algorithm was also confused at normal weight and overweight level I for 2 instances, 2 instances for obesity level one and obesity level II and with one instance of overweight level II and obesity level I. The algorithm predicted the most correct instance of obesity type II and the least correct of normal weight. The classification report when entropy was used as criterion is summarized in Table 2.

Figure 6 plots all the features on the x-axis and the importance given by the algorithm on the y-axis. The plot shows that the algorithm has given nearly 65% of the importance to the weight feature of the data, followed by nearly 19% to height and nearly 12% to the gender. All other feature were considered with the importance less than 1%.

B. RESULT FOR GINI

The confusion matrix when the Gini index was used as the criteria for defining the decision tree can be Figure 5. The algorithm was able to correctly classify 49 instances of insufficient weight, 44 instances of normal weight class, 62 instances of obesity type I, 55 instances of obesity type II, 72 instances of obesity type III, 66 instances of overweight type I, and 51 instances of overweight level 2. The algorithm was most incorrect in classifying the normal weight class.

The classification report when the Gini index was used as criteria for the decision tree gives the report shown in Table 3. The classification report shows the accuracy obtained was 94%, less than when entropy was used as a criterion. The macro average for precision, recall, and f1-score was 0.94. The weighted average was also found to be 0.94. The prediction for class Obesity_Type_III was the best with the highest score in precision, recall, and f1-score.

The decision tree when Gini was used as criteria gives the above graph. In the Figure 4, the features are plotted on the x-axis, and importance is presented on the y-axis. Here, it can be seen that the algorithm has given more priority to the weight class in this case as well. However, the importance given has decreased to 50% from 60%. The height class was given nearly 27%, gender class to nearly 18%, and other classes were given importance less than 1%.

Further tests were done with tree pruning. To find the best parameters for pruning the tree, a grid search technique along with cross-validation. For cross-validation, the number of folds was set to 5. The result of the grid search showed that the best parameters for pruning were found, as shown in the table below.

When the values that were derived from the grid search were used for the Decision Tree classifier, the accuracy, and other metrics did not change. But when these values were altered the value of accuracy, precision, recall, and f1-score decreased. When maximum features were set to any random number between 1 and 27 the accuracy was below i.e. 64% when 1 and 95% when 24. When the value was set to 25, 26 the value was close to the original accuracy. When set higher than 27 the accuracy did not change. Similarly, for the splitter criteria when set to random, the accuracy was 92%.

VI. DISCUSSION AND CONCLUSION

A. ENTROPY AS A CRITERION

The performance of the decision tree classifier using both entropy and log loss criteria was evaluated, and the results demonstrate robust classification capabilities across different weight categories. The identical results obtained from using entropy and log loss as criteria indicate that for this dataset, both measures of impurity are equally effective in producing a decision tree model with similar predictive power. The model achieved a high overall accuracy of 96%, indicating that it is well-suited for this classification task. The macro-average and weighted average of precision, recall, and F1-score are all 0.96, reinforcing the consistency and reliability of the model across different classes.

Obesity_Type_III class achieved the highest F1-score of 0.99, with perfect precision and very high recall. This suggests that the model is extremely accurate in identifying instances of Obesity_Type_III and has a low rate of false positives and false negatives. Obesity_Type_II and Overweight_Level_II classes also demonstrated strong performance with F1-scores of 0.98. The high precision and recall values indicate that these categories are well-captured by the model. Insufficient_Weight and Obesity_Type_I classes also showed high F1-scores of 0.96 and 0.98, respectively, reflecting the model's ability to accurately classify these categories. Overweight_Level_I class had a consistent F1-score of 0.94, suggesting good performance, though slightly lower compared to other classes. Normal_Weight class has the lowest F1-score of 0.91, primarily due to a lower recall of 0.89. This indicates that while the model is fairly precise in identifying Normal_Weight instances, it misses a few true instances of this class, leading to a higher false negative rate.

B. GINI AS A CRITERION

The performance of the decision tree classifier using the Gini index as a criterion was evaluated and compared to its performance using entropy. The results indicate that while the Gini index provides solid classification capabilities, it falls slightly short of the performance achieved with entropy. The decision tree model using the Gini index achieved an accuracy of 94%, which is slightly lower than the 96% accuracy obtained with entropy. The macro-average and weighted average of precision, recall, and F1-score are all 0.94 indicating consistent performance across different classes.

Obesity_Type_III class achieved the highest precision, recall, and F1-score, with values of 1, 0.99, and 0.99, respectively. This suggests that the model is extremely accurate in identifying instances of Obesity_Type_III, with very few false positives and false negatives. Obesity_Type_II class also demonstrated strong performance with a precision of 0.98, recall of 0.96, and F1-score of 0.97. Overweight_Level_II class achieved an F1-score of 0.93, indicating good model performance with high precision and recall values. Insufficient_Weight, Obesity_Type_I, and Overweight_Level_I classes had F1-scores of 0.94, 0.93, and 0.93, respectively, reflecting reliable classification but with slight room for improvement. Normal_Weight class had the lowest F1-score of 0.87, primarily due to a lower recall of 0.81. This indicates that while the model is precise in identifying Normal_Weight instances, it misses more true instances of this class compared to using entropy.

C. DISCUSSION ON GRID SEARCH

Upon changing the best parameters for the Decision Tree, different values were obtained. Most in the case decreased in accuracy. When the value of maximum features was decreased to 1 the accuracy was found to be 64% meaning that the model only used single features for splitting which caused the model to not be able to generalize the data well. Due to this, the model could not classify the instances well and accuracy decreased.

When the maximum depth of the model was decreased, the accuracy of the model decreased. The reason behind this is model could not capture the complexity of the data and had to make the generalized rule to make decisions about the data point. When the splitter was set to random, the accuracy decreased. The reason behind this is tree will select a random subset of features and thresholds to consider for splitting, rather than evaluating all possible splits. The splits done by selecting random features are not optimal.

APPENDIX A
Equations

- **Entropy**

$$H = -\sum_{i=1}^{m} p_i \, log_2 \, p_i \qquad (1)$$

- **Gini Index**

$$Gini = 1 - \sum_{i=1}^{K} p_i^2 \qquad (2)$$

- **Classification Error**

$$CE = 1 - \max_i p_i \qquad (3)$$

- **Information Gain**

$$\Delta H = H - \frac{m_L}{m} H_L - \frac{m_R}{m} H_R \qquad (4)$$

- **Information Gain Ratio**

$$GainRatio_{split} = \frac{Gain_{split}}{SplitINFO} \qquad (5)$$

$$SplitINFO = -\sum_{i=1}^{k} \frac{n_i}{n} \log \frac{n_i}{n} \qquad (6)$$

- **Accuracy**

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \qquad (7)$$

- **Precision**

$$Precision = \frac{TP}{TP+FP} \qquad (8)$$

- **Recall**

$$Recall = \frac{TP}{TP+FN} \qquad (9)$$

- **F1 Score**

$$F1\ Score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \qquad (10)$$

- **Mass Body Index**

$$Mass\ Body\ Index = \frac{weight}{height \times height} \qquad (11)$$

APPENDIX B
Figures

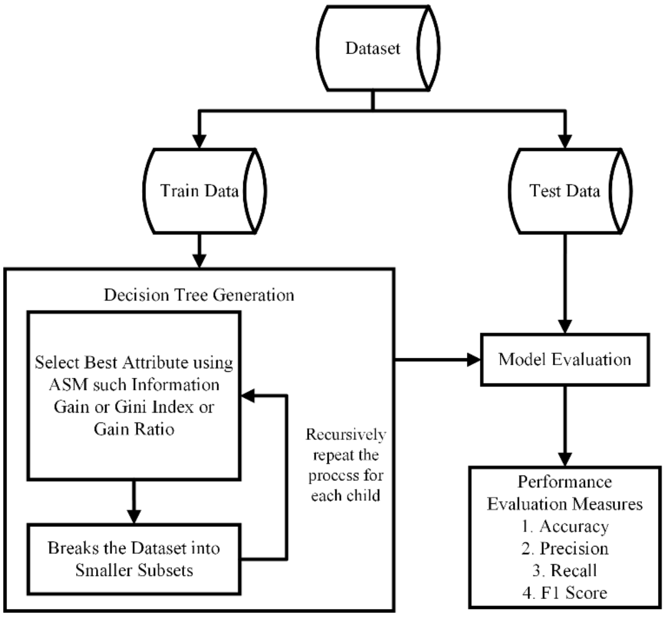
Dataset
Train Data
Test Data
Decision Tree Generation
Select Best Attribute using ASM such Information Gain or Gini Index or Gain Ratio
Breaks the Dataset into Smaller Subsets
Recursively repeat the process for each child
Model Evaluation
Performance Evaluation Measures
1. Accuracy
2. Precision
3. Recall
4. F1 Score

Figure 1: Block Diagram for Decision Tree

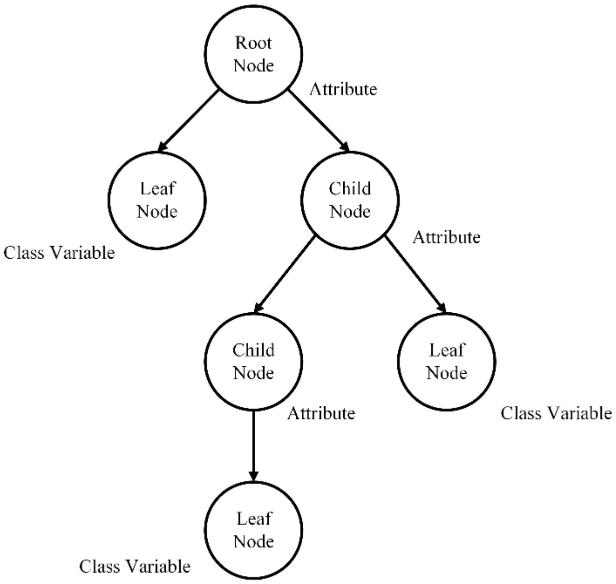

Figure 2: Decision Tree Structure

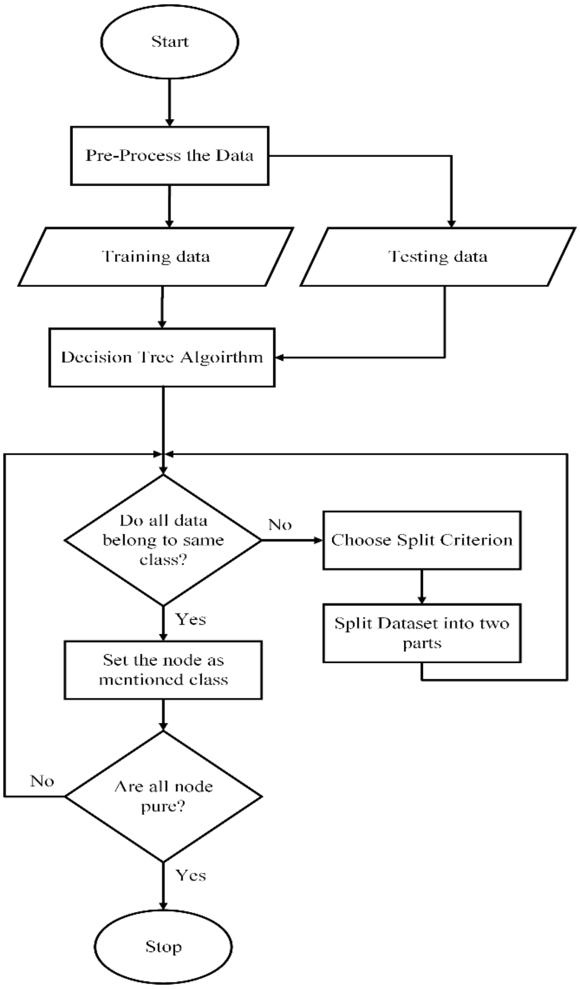

Figure 3: Flowchart for Decision Tree

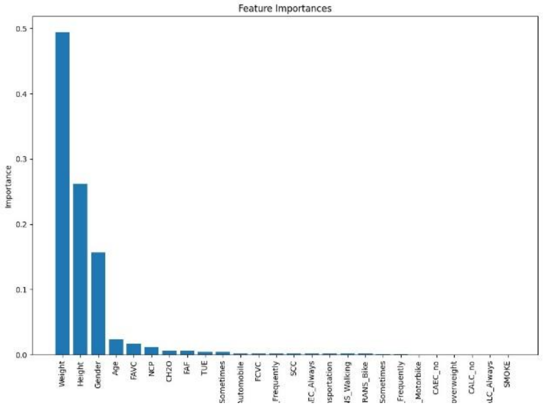

Figure 4: Feature vs Importance Plot when Gini was used as Criterion

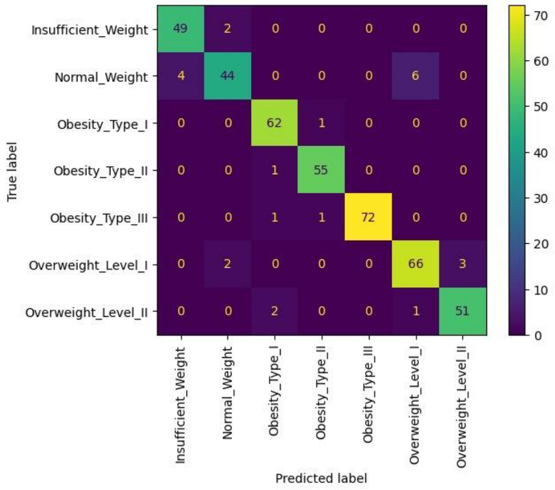

Figure 5: Confusion Matrix (Gini)

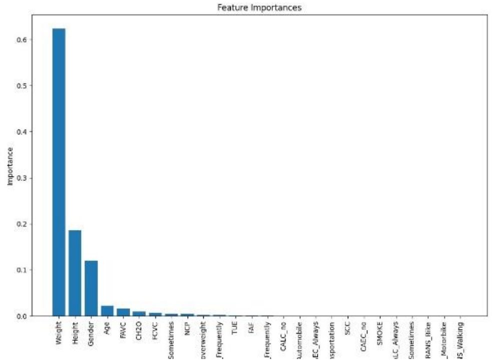

Figure 6: Feature vs Importance Plot when Entropy was used as Criterion

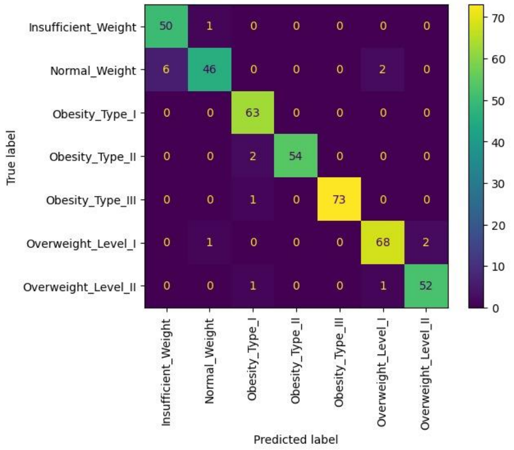

Figure 7: Confusion Matrix (Entropy)

APPENDIX C
Tables

Table 1: Class Label and Mass Body Index

Class Label	**Mass Body Index**
Underweight	Less than 18.5
Normal	18.5 to 24.9
Overweight	25.0 to 29.9
Obesity I	30.0 to 34.9
Obesity II	35.0 to 39.0
Obesity III	Higher than 40

Table 2: Classification Report when Entropy was used as Criterion

Category	Precision	Recall	F1-Score	Support
Insufficient_Weight	0.94	0.98	0.96	51
Normal_Weight	0.92	0.89	0.91	54
Obesity_Type_I	0.97	1	0.98	63
Obesity_Type_II	0.98	0.98	0.98	56
Obesity_Type_III	1	0.97	0.99	74
Overweight_Level_I	0.94	0.94	0.94	71
Overweight_Level_II	0.98	0.98	0.98	54
accuracy	0.96			423
macro avg	0.96	0.96	0.96	423
weighted avg	0.96	0.96	0.96	423

Table 3: Classification Report when Gini was used as Criterion

Category	Precision	Recall	F1-Score	Instances
Insufficient_Weight	0.92	0.96	0.94	51
Normal_Weight	0.94	0.81	0.87	54
Obesity_Type_I	0.92	0.94	0.93	63
Obesity_Type_II	0.98	0.96	0.97	56
Obesity_Type_III	1	0.99	0.99	74
Overweight_Level_I	0.92	0.94	0.93	71
Overweight_Level_II	0.9	0.96	0.93	54
accuracy	0.94			423
macro avg	0.94	0.94	0.94	423
weighted avg	0.94	0.94	0.94	423

Table 4:Dataset Description

Feature	Meaning	Type
Age	What is your age?	Continuous
Height	What is your height?	
Weight	What is your weight?	
FCVC	Do you eat Vegetable in your meal?	
NCP	How many main meals do you have daily?	
CH2O	How much water do you drink?	
FAF	How often do you have physical activity?	
Gender	Are you male or female?	Categorical
family_history_with_overweight	Has a family member suffered or suffers from overweight?	
FAVC	Do you eat high caloric food frequently?	
CAEC	Do you eat any food between meals?	
SMOKE	Do you smoke?	
SCC	Do you monitor the calories you eat daily?	
TUE	How much time do you use technological devices?	
CALC	How often do you drink alcohol?	
MTRANS	Which transportation do you usually use?	
NObeyesdad	Obesity Level	

REFERENCES

[1] F. M. Palechor and A. de la H. Manotas, “Dataset for estimation of obesity levels based on eating habits and physical condition in individuals from Colombia, Peru and Mexico,” Data in Brief, vol. 25, p. 104344, Aug. 2019, doi: 10.1016/j.dib.2019.104344.

[2] O. Iparraguirre-Villanueva, L. Mirano-Portilla, M. Gamarra-Mendoza, and W. Robles-Espiritu, “Predicting Obesity in Nutritional Patients using Decision Tree Modeling,” International Journal of Advanced Computer Science and Applications, vol. 15, no. 3, 2024, doi: 10.14569/ijacsa.2024.0150326.

[3] E. De-La-Hoz-Correa, F. E. Mendoza-Palechor, A. De-La-Hoz-Manotas, R. C. Morales-Ortega, and S. H. Beatriz Adriana, “Obesity Level Estimation Software based on Decision Trees,” Journal of Computer Science, vol. 15, no. 1, pp. 67–77, Jan. 2019, doi: 10.3844/jcssp.2019.67.77.

[4] P. H. Swain and H. Hauska, “The decision tree classifier: Design and potential,” IEEE Transactions on Geoscience Electronics, vol. 15, no. 3, pp. 142–147, Jul. 1977, doi: 10.1109/tge.1977.6498972.

[5] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques. Elsevier, 2011.

PRAGYAN BHATTARAI is currently pursuing Bachelor’s degree in Electronics, Communication and Information engineering in IOE, Thapathali Campus. He is deeply passionate about working with data. His journey began with learning the basics of Python and gaining an understanding of data science. Since then, he has been getting acquainted with data visualization techniques. He is eager to further strengthen his statistical knowledge and venture into the field of machine learning.

PRASHANT R. BISTA. is currently pursuing Bachelor’s degree in Electronics, Communication and Information engineering in IOE, Thapathali Campus. He is fascinated by open-source software’s and open-source community. He is currently learning about machine learning and its application on making human life easier. His hobby includes reading novels, exploring machines and web development.