IJEPH
Interdisciplinary
Journal of Epidemiology
& Public Health

Articulo original

Enhancing Obesity Prediction through SMOTE-based Classification Models: A Comparative Study

Aumento de la predicción de la obesidad mediante modelos de clasificación basados en SMOTE: Un estudio comparativo

John Kamwele Mutinda [1], Amos Kipkorir Langat [2], Regis Konan Marcel Djaha [3], Jackson Ndoto Munyao [4], Lee Whitaker [5], Millicent Auma Omondi [4]

[1] University of Science and Technology of China, Langfang, Hebei, China
[2] Department of Mathematics, Technology and Innovation-JKUAT, Pan African University Institute for Basic Sciences, Nairobi, Kenya
[3] Basque Center for Applied Mathematics, Bilbao, Basque, Spain
[4] African Institute for Mathematical Sciences, Limbe, Cameroon
[5] South Eastern Kenya University, Kitui County, Kenya

Correspondence: Amos Kipkorir Langat. E-mail: moskiplangat@gmail.com

Received: 18 March 2024
Accepted: 30 June 2024
Publicado: 30 June 2024

Keywords: Obesity, classification algorithms, synthetic minority over-sampling technique (smote), imbalanced data, healthcare decision-making

Palabras clave: Obesidad, algoritmos, Técnica de Sobremuestreo de Minorías Sintéticas (smote), datos desbalanceados, toma de decisiones en salud.

Citation: Kamwele MJ, KipkorirLA, Marcel DRK, Ndoto MJ, Whitaker L, Auma OM. Enhancing Obesity Prediction through SMOTE-based Classification Models: A Comparative Study. iJEPH. 2024; 7(1): e-11532. Doi: 10.18041/2665-427X/ijeph.1.11532.

Conflicto de interés: The authors declare no conflicts of interest

Resumen

Objetivo: Utilizar Técnica de Sobre muestreo de Minorías Sintéticas (SMOTE) para mejorar el equilibrio de clases y comparar el rendimiento de distintos métodos de clasificación antes y después de aplicar SMOTE

Métodos: Los métodos de clasificación fueron Regresión Logística, Naive Bayes, KNN (k=5) y Aprendizaje Profundo. Cada modelo fue entrenado y probado en el conjunto de datos, antes y después de aplicar SMOTE. Se utilizaron las métricas de evaluación: Precisión, Sensibilidad, Especificidad, Precisión equilibrada, Puntuación F1.

Resultados: Modelos como la regresión logística y Naive Bayes tuvieron problemas con sensibilidad y especificidad bajas, KNN (k=5) mostró una especificidad deficiente. Con SMOTE, se observaron mejoras significativas en todos los modelos. La regresión logística, a pesar de una disminución de la precisión (-8.8), la sensibilidad y la especificidad aumentaron sustancialmente (+56.7%), y mejoró la precisión equilibrada (+16.6%). Naive Bayes experimentó un modesto aumento de la precisión (+2.3%), mejoró la sensibilidad y la especificidad (+47.9%). El clasificador KNN mostró una mejora transformadora con aumento de la sensibilidad, la especificidad (+96.0%) y precisión equilibrada (+28.3%). El aprendizaje profundo mostró aumento significativo de sensibilidad (+69.8%), exactitud equilibrada (+29.4%) y una mejora notable de la precisión y la puntuación F1 a pesar de un ligero descenso de la especificidad (-10.9%).

Conclusiones: SMOTE contribuye a realizar predicciones más exactas y fiables. Aunque puede haber ligeras desventajas, las mejoras generales en las métricas usadas confirman la utilidad de SMOTE para mejorar el rendimiento de los modelos en conjuntos de datos desequilibrados

Abstract

Objective: To use SMOTE to enhance class balance and compare the performance of different classification methods before and after applying SMOTE.

Methods: Study used a dataset obtained from Kaggle, which consisted of several health-related features linked to obesity prediction. The design involved checking for class imbalance within the dataset, which affected initial model performance. SMOTE was applied to synthetically increase the representation of minority classes, effectively reducing the class imbalance. The experiment was conducted in two stages: 1. Training and testing the classification algorithms before applying SMOTE. 2. Training and testing the same models after applying SMOTE to enhance class balance. The performance of all models was evaluated based on metrics before and after the SMOTE application.

Results: Initially, models like Logistic Regression and Naive Bayes struggled with low sensitivity and specificity, KNN (k=5) showed poor specificity. Significant improvements were observed across all models after applying SMOTE. Logistic Regression, despite a decrease in accuracy (-8.8), sensitivity and specificity increased substantially (+56.7%), with balanced accuracy improving (+16.6%). Naive Bayes saw a modest accuracy increase (+2.3%), with sensitivity and specificity improving (+47.9%). The KNN (k=5) classifier exhibited a transformative enhancement with sensitivity and specificity increasing (+96.0%) and balanced accuracy (+28.3%). Deep Learning showed a significant increase in sensitivity (+69.8%), balanced accuracy (+29.4%), and an improvement in precision and F1-score despite a slight decrease in specificity (-10.9%).

Conclusion: The study highlights the importance of SMOTE in healthcare applications, contributing to more accurate predictions and reliable healthcare decision-making. The results demonstrate that while there might be slight trade-offs, the overall improvements in key metrics such as sensitivity, specificity, balanced accuracy, precision, and F1-score affirm the utility of SMOTE in enhancing model performance for imbalanced datasets.

Universidad Libre

ISSN: 2665-427X

Contribución clave del estudio

Objective	To assess the impact of the Synthetic Minority Over-sampling Technique (SMOTE) on class imbalance in predicting obesity, using multiple classification algorithms
Study design	The study used a dataset obtained from Kaggle, which consisted of several health-related features linked to obesity prediction. The design involved checking for class imbalance within the dataset, which affected initial model performance. SMOTE was applied to synthetically increase the representation of minority classes, effectively reducing the class imbalance. The experiment was conducted in two stages: 1. Training and testing the classification algorithms before applying SMOTE. 2. Training and testing the same models after applying SMOTE to enhance class balance. Performance of all models was evaluated based on metrics before and after SMOTE application
Source of information	The dataset used for this study was obtained from Kaggle, a platform known for providing a wide variety of datasets for data analysis and machine learning tasks. The dataset includes multiple health-related attributes such as BMI, physical activity, etc
Statistical analysis	The classification methods utilized in this study were Logistic Regression, Naïve Bayes, KNN (k=5), and Deep Learning. Each model was trained and tested on the dataset before and after applying SMOTE. The following evaluation metrics were used: • Accuracy: The proportion of correct predictions out of total predictions. • Sensitivity (Recall): The model's ability to correctly identify positive cases. • Specificity: The model's ability to correctly identify negative cases. • Balanced Accuracy: The average of sensitivity and specificity. • Precision: The proportion of positive predictions that are positive. • F1-Score: The harmonic mean of precision and recall, providing a balance between the two
Principle findings	DThe application of SMOTE significantly enhanced model performance. Key findings include: • Logistic Regression: Despite a -8.8% decrease in accuracy, sensitivity and specificity increased by +56.7%, leading to a +16.6% improvement in balanced accuracy. • Naive Bayes: A modest accuracy increase of +2.3% was observed, with sensitivity and specificity improving by +47.9%. • KNN (k=5): The classifier showed the most significant improvement, with sensitivity and specificity increasing by +96.0% and balanced accuracy improving by +28.3%. • Deep Learning: Sensitivity increased by +69.8%, balanced accuracy by +29.4%, and precision and F1-score improved, despite a slight decrease in specificity by -10.9%. These results demonstrate that SMOTE effectively mitigates class imbalance issues, improving model performance, particularly in sensitivity and balanced accuracy, across all tested algorithms. The study highlights the utility of SMOTE in healthcare prediction models, where accurate identification of minority classes is crucial for decision-making

Introduction

Obesity has become increasingly prevalent worldwide and is now a major contributor to poor health, surpassing undernutrition, and infectious diseases (1,2). It is linked to various serious health conditions such as diabetes, heart disease, cancer, and sleep disorders (3). Obesity is typically defined by a body-mass index (BMI) of 30 kg/m^2 or higher, but this doesn't fully capture the health risks associated with even modest overweight or intra-abdominal fat (1). The global rise in obesity is due to genetic factors, easy access to high-calorie foods, and reduced physical activity in modern society. It's no longer just a cosmetic issue but a global epidemic threatening overall well-being.

Machine learning algorithms have revolutionized the health sector by enabling powerful classification techniques for various tasks such as disease diagnosis, risk prediction, and treatment recommendation. These algorithms leverage large datasets to identify patterns and relationships within medical data, allowing for more accurate and efficient decision-making. They have demonstrated remarkable power in distinguishing between disease states, stratifying patients based on risk profiles, and optimizing treatment strategies (4-6).

Machine learning algorithms have been instrumental in classifying obesity within human healthcare, offering a nuanced approach to understanding and addressing this complex condition. These algorithms utilize diverse data sources such as electronic health records, medical imaging, genetic information, and lifestyle factors to accurately classify individuals based on their obesity status. Through sophisticated pattern recognition techniques, machine learning models can differentiate between various degrees of obesity, assess associated health risks, and personalize intervention strategies accordingly. These algorithms not only consider traditional metrics like BMI but also incorporate additional factors such as body composition, metabolic markers, and genetic predispositions to provide a more comprehensive evaluation of obesity and its implications for overall health. By leveraging the power of machine learning, healthcare professionals can enhance obesity classification accuracy, tailor interventions to individual needs, and ultimately improve patient outcomes (7,8).

The authors in (8) employed machine learning algorithms for predicting obesity risk, leveraging a dataset comprising over 1,100 individuals spanning diverse age groups and obesity statuses. Nine prominent machine learning algorithms were applied and evaluated, including k-nearest neighbor (k-NN), random forest, logistic regression, multilayer perceptron, support vector machine (SVM), naïve Bayes, adaptive boosting, decision tree, and gradient boosting classifier. The logistic regression algorithm demonstrated the highest accuracy at 97.1%, outperforming other classifiers, while the gradient boosting algorithm exhibited the lowest accuracy at 64.1%. This research aimed to predict obesity risk but also sought to enhance understanding of the underlying factors contributing to obesity, informing preventive strategies and interventions.

The authors in (9) utilized machine learning (ML) methods such as Logistic Regression, Classification and Regression Trees (CART), and Naïve Bayes to predict obesity using publicly available health data, aiming to surpass traditional models and identify key risk factors. Logistic Regression emerges as the most effective method, though moderate concordance between predicted and measured obesity is observed. Significant risk factors for obesity in adults include location, marital status, age groups, education, dietary habits, mental health disorders, physical activity, and smoking. Addressing data imbalance using Synthetic Minority Oversampling Technique (SMOTE), this study underscores the importance of identifying these risk factors to inform policy interventions aimed at controlling chronic diseases, especially obesity-related complications. Applying ML methods to available health data holds promise for advancing our understanding of obesity and its associated risk factors, facilitating more robust preventive strategies.

A study by (10) introduces a novel method utilizing data science techniques to analyze genetic variants extracted from publicly available genetic profiles and a curated database, the National Human Genome Research Institute Catalog, for predicting obesity. Genetic variants are indexed and utilized as inputs in various machine learning algorithms to classify participants into Normal Class or Risk Class based on body mass index status. A set of principal variables consisting of 13 Single Nucleotide Polymorphisms is generated through dimensionality reduction to apply different machine-learning methods. The performance of various algorithms, including gradient boosting, generalized linear model, classification and regression trees, k-nearest neighbors, support vector machines, random forest, and multilayer perceptron neural network, is evaluated using receiver operator characteristic curves and area under the curve metrics. The support vector machine demonstrated the highest performance with an area under the curve value of 90.5, suggesting its efficacy in identifying significant factors among the initial 6,622 variables for classifying subjects into BMI-related classes.

A study by (11) utilized four enhanced machine learning models to predict obesity among high school students, considering both risk and protective factors. The models included binary logistic regression, improved decision tree, weighted k-nearest neighbor, and artificial neural network (ANN). Using nine health-related behaviors from the 2015 Youth Risk Behavior Surveillance System for Tennessee as inputs, the results indicated significant performance improvements over traditional logistic regression. Specifically, the improved decision tree model achieved 80.2% accuracy and 90.7% specificity, the weighted KNN model achieved 88.8% accuracy and 93.4% specificity, and the ANN model achieved 84.2% accuracy and 99.5% specificity. These findings suggest the potential of machine learning in effectively predicting and addressing adolescent obesity, with implications for interventions aimed at slowing its increase.

A study by (12) focused on exploring the relationship between physical activity and weight status and evaluating the performance of various machine learning and traditional statistical methods. Utilizing National Health and Nutrition Examination Survey data from 2003 to 2006, the study included 7,162 participants meeting inclusion criteria. Eleven classification algorithms were implemented, including logistic regression, naïve Bayes, and Radial Basis Function. The random subspace classifier algorithm demonstrated the highest accuracy and area under the receiver operating characteristic curve. Vigorous and moderate-intensity activity durations emerged as significant attributes. While logistic regression ranked middling among the methods, it provided valuable insights. Gender, age, and race/ethnicity also played essential roles in weight outcomes. Tailored intervention strategies considering these factors are crucial in combating obesity and addressing disparities among demographic populations.

As we delve into the analysis, it becomes evident oversampling techniques, notably the Synthetic Minority Over-sampling Technique (SMOTE), are integral components of the model development process. This study will incorporate this technique to address class imbalance in the dataset and enhance the robustness of model training and evaluation processes. In this study, we propose to predict obesity by employing various classification models: Logistic Regression, KNN, Random Forest, and deep learning. Recognizing the class imbalance inherent in obesity data, we aim to address this challenge by implementing the Synthetic Minority Over-sampling Technique (SMOTE). Our methodology involves training these models before and after oversampling, allowing for a comparative performance analysis. To evaluate the effectiveness of each model, we will employ a range of performance metrics, including Accuracy, Balanced Accuracy, Sensitivity, Specificity, F1-Score, and Precision. This approach aims to fill the existing research gap by comprehensively evaluating obesity prediction models before and after addressing class imbalance, thereby contributing to advancements in obesity management strategies.

Materials and methods

Data

The dataset used in this study consists of 150 entries and 14 columns. Each column represents a variable providing information about individuals. Table 1 presents an overview of the key variables, their meanings, and types.

The dataset provides a mix of continuous and categorical variables,

Table 1. Variable Overview

Variable Name	Meaning	Type
Level	Academic level of the individual	Continuous
Faculty	Faculty affiliation	Categorical
Gender	Gender of the individual	Categorical
Age	Age of the individual	Continuous
Family Size	Size of the family	Continuous
Obese	Obesity status (Target)	Categorical
Income	Individual's income level	Continuous
Daily Eating	Daily eating habits	Categorical
Fruit Intake	Frequency of fruit intake	Continuous
BMI Aware	Awareness of BMI	Categorical
Sleeping Hours	Hours of sleep per day	Continuous
Exercise Freq	Frequency of exercise	Continuous
Need help	Indicates if help is needed	Categorical
Need app	Indicates the need for an app	Categorical

with “Obese” serving as the target variable indicating the obesity status of individuals. These variables lay the foundation for subsequent modeling and analysis.

Dealing with class imbalance

SMOTE is a powerful method used to address class imbalance in datasets by generating synthetic samples of the minority class (13,14). It works by synthesizing new instances of the minority class by interpolating between existing minority class instances, thus creating a more balanced dataset. SMOTE helps to mitigate the problem of biased classification models that tend to favor the majority class due to its higher representation in the dataset. By introducing synthetic samples, SMOTE enhances the diversity of the minority class, allowing machine learning algorithms to better learn the underlying patterns and improve classification performance (15-17). This technique has been widely adopted in various fields, including healthcare (18-20), finance (21-23), and image recognition (24,25), where imbalanced datasets are common, contributing to more accurate and robust predictive models. The steps on how SMOTE works are shown below:

- **Identify the Minority Class:** Determine the minority class in the dataset that needs over-sampling.
- **Select a Sample from the Minority Class:** Randomly choose a sample from the minority class, denoted as *x*.
- **Find Nearest Neighbors:** Identify *k*-nearest neighbors of *x* from the same class. This is usually done using a distance metric like Euclidean distance. For a given sample *x* from the minority class, find its *k*-nearest neighbors using Euclidean distance:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \qquad (1)$$

Where ($x = x_1, x_2, \cdots, xn$) $y = (y1, y2, \cdots, yn)$ are feature vectors.

- **Generate Synthetic Samples:** Randomly select one of the *k*-nearest neighbors, denoted as x_{neighbor}. Create a synthetic sample x_{syntetic} by interpolating between *x* and *xneighbor*:

$$\mathrm{x}_{\text{synthetic}} = \mathrm{x} + \delta \cdot (\mathrm{x}_{\text{neighbor}} - \mathrm{x}) \qquad (2)$$

where δ is a random number between 0 and 1.

Figure 1 below shows the Pseudo code for SMOTE, and Figure 2 shows an illustration of SMOTE. Apart from its application in heart disease prediction, SMOTE has widely been used in other health events where prediction is warranted. For instance, a study (26) used the SMOTE technique to balance the dataset to improve the prediction performance of heart disease using the Decision Tree algorithm on the Cleveland Heart Disease Dataset. The SMOTE technique addressed data imbalances between minority and majority classes. Experimental results showed that balancing the data with SMOTE improved decision tree classification accuracy from 73.3% to 91.4%, an increase of up to 18.1%. A study (27) used the SMOTE technique to balance the dataset and improve the accuracy of rule induction and decision tree models for predicting kidney disease using data from Apollo Hospitals, Tamil Nadu, India. The initial imbalanced dataset hindered model accuracy, but applying SMOTE minimized class variation. Experimental findings showed an average accuracy improvement of 98.73%. This method can also enhance accuracy in other imbalanced datasets and be applied in Big Data contexts using Hadoop and MapReduce. A study (28) proposed a two-step approach to improve predictive accuracy in healthcare, addressing the limitations of basic SMOTE. First, modified SMOTE techniques Distance-based SMOTE was used to reduce class imbalance and showed improved accuracy over basic SMOTE. Second, a Stacking Ensemble Framework combining machine learning, deep learning, and ensemble algorithms significantly increased accuracy to 96-97% for various datasets. This framework was validated using the Framingham dataset, Wisconsin Hospital data, and Novel Coronavirus 2019 dataset.

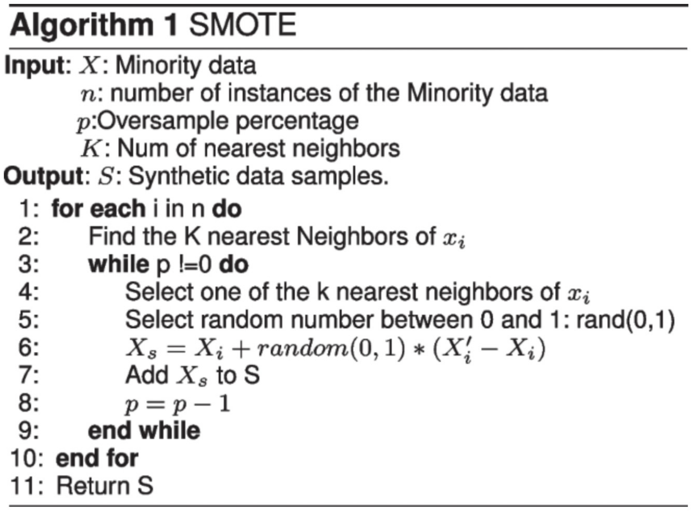
Classification Models

The classification models chosen for this analysis include Logistic Regression, *k*-Nearest Neighbors (KNN), Random Forest and Deep Learning

Logistic Regression

Logistic Regression is a widely used linear classification algorithm that models the probability of a binary outcome. It is particularly suitable for predicting binary variables, such as whether an individual is obese or not. The logistic regression model predicts the log-odds (logit) of the probability of the positive class (29). In logistic regression, the hypothesis function $h_\theta(x)$ is defined as:

$$h_\theta(x) = g(\theta^T x) \qquad (3)$$



```
Algorithm 1 SMOTE
Input: $X$: Minority data
       $n$: number of instances of the Minority data
       $p$:Oversample percentage
       $K$: Num of nearest neighbors
Output: $S$: Synthetic data samples.
for each i in n do
    Find the K nearest Neighbors of $x_i$
    while p !=0 do
        Select one of the k nearest neighbors of $x_i$
        Select random number between 0 and 1: rand(0,1)
        $X_s = X_i + random(0,1) * (X_i' - X_i)$
        Add $X_s$ to S
        $p = p - 1$
    end while
end for
Return S
```

Figure 1. SMOTE Pseudo code

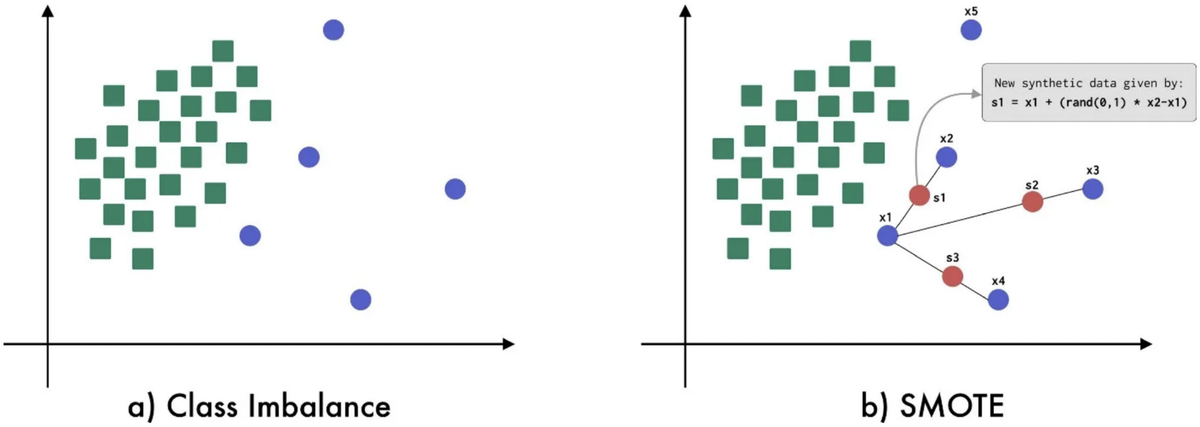


Figure 2. A Simple illustration of SMOTE

where the function $\dot{g}$ is the sigmoid function, represented as:

$$g(z) = \frac{1}{1 + e^{-z}} \qquad (4)$$

The sigmoid function ensures that the output of the hypothesis lies between 0 and 1, making it suitable for binary classification tasks.

Regarding the loss function in logistic regression, it is commonly defined as the logistic loss or binary cross-entropy loss. For a single training example with true label y and predicted probability $h_\theta(x)$, the logistic loss is computed as:

$$\text{Loss}(h_\theta(x), y) = -y \log(h_\theta(x)) - (1 - y) \log(1 - h_\theta(x)) \qquad (5)$$

The objective is to minimize this loss function over the entire training dataset. The loss function penalizes the model more heavily for incorrect predictions, particularly when the predicted probability diverges significantly from the true label. Minimizing this loss function effectively adjusts the model parameters θ to improve the accuracy of predictions in logistic regression.

K-Nearest Neighbors (KNN)

The *k*-nearest neighbors (KNN) algorithm is a simple, yet effective method used for classification tasks in machine learning. It operates based on the principle of similarity: If an instance is like other instances in the dataset, it is likely to belong to the same class (30,31).

Here's how the KNN algorithm works:

- **Training Phase:** The algorithm stores all the available training data points and their corresponding class labels.
- **Prediction Phase:** When a new instance (point) is presented for classification, the KNN algorithm calculates the distances between this instance and all other instances in the training dataset. The distance measure used is typically Euclidean distance, although other distance metrics such as Manhattan distance or cosine similarity can also be used.
- **Finding Neighbors:** Once distances are calculated, the algorithm identifies the k-nearest neighbors of the new instance. These are the k data points in the training set that are closest to the new instance.
- **Majority Voting:** Finally, the algorithm assigns the class label to the new instance based on a majority vote among its k nearest neighbors. That is, the class label that occurs most frequently among the *k* neighbors is assigned to the new instance.

The working framework of KNN involves calculating the Euclidean distance between the new instance (denoted as *x*) and each point in the training dataset (denoted as *y*).

The Euclidean distance between two points *x* and *y* in an *n*-dimensional space can be calculated using the following formula:

$$\text{Euclidean Distance} = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \qquad (6)$$

Where:
– x_1 and y_1 are the *i*-th dimensions of points *x* and *y* respectively.
– *n* is the number of dimensions (or features) in the dataset.

By calculating the Euclidean distance between the new instance and each point in the training dataset, the KNN algorithm identifies the *k*-nearest neighbors, which are then used for classification.

Naive Bayes

Bayesian classification, a supervised learning and statistical method, operates on an underlying probabilistic model, allowing for the quantification of uncertainty in outcomes. It effectively addresses predictive problems by determining probabilities associated with various outcomes (31,32). In Naive Bayes classification, we aim to predict the class label *y* based on the features $x_1, x_2, \cdots, x_p$. We assume that the features are conditionally independent given the class label y, which means that:

$$P(x_1, x_2, ..., x_p | y = c) = P(x_1, x_2, ..., x_p | y = c) \times P(x_2 | x_3 ... x_p, y = c) \times \ldots \times P(x_p | y = c)$$

We can then apply Bayes' theorem to calculate the probability of each class given the observed features:

$$P(y=c|x_1,x_2,\dots,x_p)=\frac{P(x_1,x_2,\dots,x_p|y=c)\times P(y=c)}{P(x_1,x_2,\dots,x_p)}$$

To classify a new instance with features x_1, x_2, ⋯, x_p, we select the class label that maximizes the posterior probability:

$$\hat{y}=\arg\max_c P(y=c|x_1,x_2,\dots,x_p)=\arg\max_c P(x_1,x_2,\dots,x_p|y=c)\times P(y=c)$$

The classical procedure for Naive Bayes is as follows:

- **Calculate Class Priors:** Calculate the prior probabilities $P(y = c)$ for each class c based on the frequency of class labels in the training dataset.
- **Calculate Class-Conditional Probabilities:** For each feature x_i, calculate the class-conditional probability $P(x_i \mid y = c)$ for each class c. This can be done using different probability distributions depending on the type of feature (e.g., Gaussian distribution for continuous features, multinomial distribution for discrete features).
- **Calculate Posterior Probabilities:** Using the conditional independence assumption, calculate the joint probability $P(x_1, x_2, \cdots, x_p \mid y = c)$ for each class c as the product of the individual class-conditional probabilities.
- **Normalize Probabilities:** Normalize the posterior probabilities $P(y = c \mid x_1, x_2, \cdots, x_p)$ by dividing each by the sum of all posterior probabilities.
- **Classify New Instances:** For each new instance with features x_1, x_2, ⋯, x_p, calculate the posterior probability for each class using Bayes' theorem and select the class with the highest posterior probability as the predicted class label.

Deep learning

Deep learning for classification is a sophisticated approach that leverages neural networks with multiple layers to categorize input data into distinct classes (Figure 3). The process begins with data preprocessing, where input features are normalized or standardized to ensure uniformity in scale and the dataset is divided into training, validation, and testing sets for evaluation. Subsequently, the architecture of the neural network is defined, specifying the number of layers, neurons per layer, and activation functions. In classification tasks, the output layer typically comprises neurons equivalent to the number of classes, with a sigmoid or softmax activation function employed to generate class probabilities (33).

Figure 3 show an illustration of deep learning neural network. During the forward propagation phase, input data traverse through the neural network, undergoing computations layer by layer to produce the network's output. Each layer computes its output through a linear transformation followed by an activation function. To quantify the disparity between predicted output and true labels, an appropriate loss function is selected. For classification tasks, common choices include categorical cross-entropy for multi-class classification and binary cross-entropy for binary classification. The backpropagation process calculates gradients of the loss function with respect to model parameters and updates these parameters using an optimization algorithm, like stochastic gradient descent (SGD), Adam, or RMSprop (34).

Training ensues by iteratively feeding batches of data through the network, computing loss, and adjusting parameters through backpropagation. Regular monitoring of the model's performance on the validation set helps prevent overfitting and fine-tune hyperparameters. Finally, the model is evaluated on the test set to gauge its performance on unseen data. Metrics such as accuracy, precision, recall, and F1-score are computed to assess its efficacy. Activation functions, including sigmoid and ReLU play crucial roles in determining the output of neurons within the network, contributing to its overall classification capability. Figure 4 show the activation functions used in deep neural networks.

Model training and validation

Due to the small size of the data, we utilized cross-validation to avoid overfitting. Cross-validation is a technique used to assess the performance of machine learning models by dividing the dataset into multiple subsets, or folds. Each fold is used as a validation set, while the rest of the data is used for training. This process is

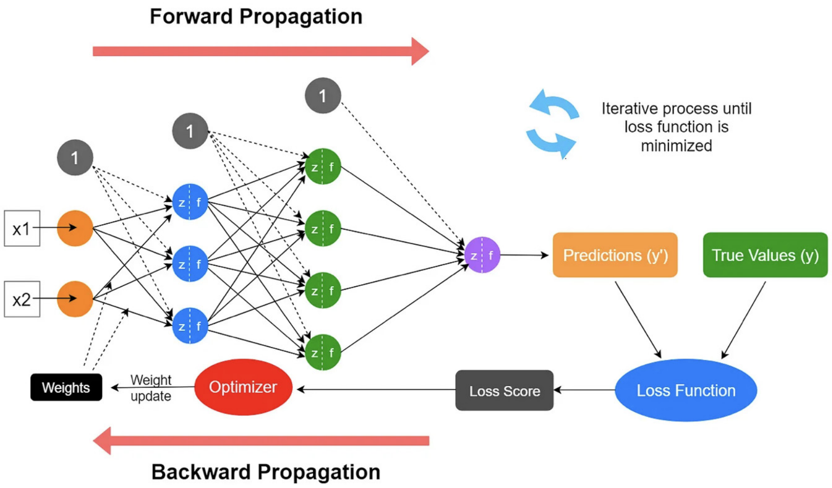


Figure 3. A layout of Deep Learning Neural Network

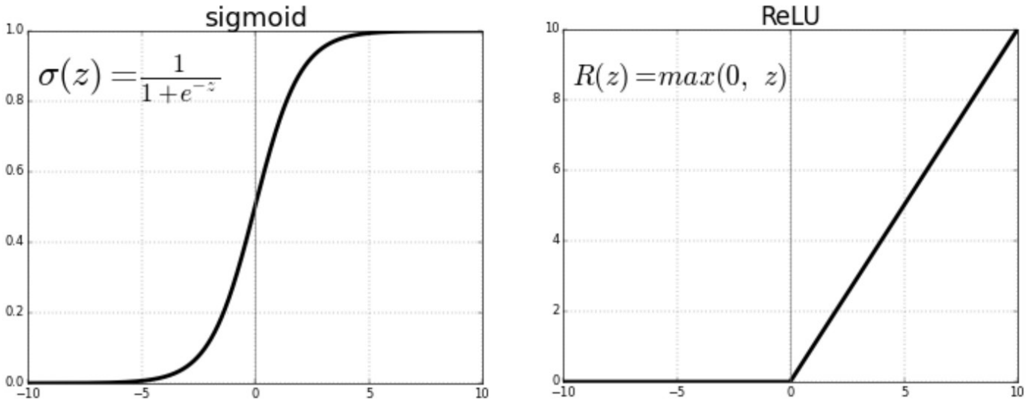


Figure 4. Commonly used Activation function in Deep Learning

repeated multiple times for our case we will use 5 folds, with each fold serving as the validation set exactly once. The main advantage of cross-validation is that it provides a more robust estimate of the model's performance compared to a single train-test split. In this scenario, train-test split was not utilized due to the relatively small size of the dataset, which could lead to high variability in model performance estimates. By using cross-validation, we ensure that each data point is used for both training and validation, leading to a more reliable assessment of the model's generalization ability (35,36,37). The formula for calculating the average metric across all folds in cross-validation can be expressed as:

$$\text{Average Metric} = \frac{1}{k}\sum_{i=1}^{k} \text{Metric}_i$$

Where:

- k is the number of folds in cross-validation.
- Metric$_i$ represents the metric value (such as accuracy, precision, recall, F1-score,balanced accuracy calculated for each fold i.

The Figure 5 below illustrates how cross validation works in training and validation of our classification models.

Model Evaluation Metrics

In this section, we define and discuss key evaluation metrics used for assessing the performance of classification models.

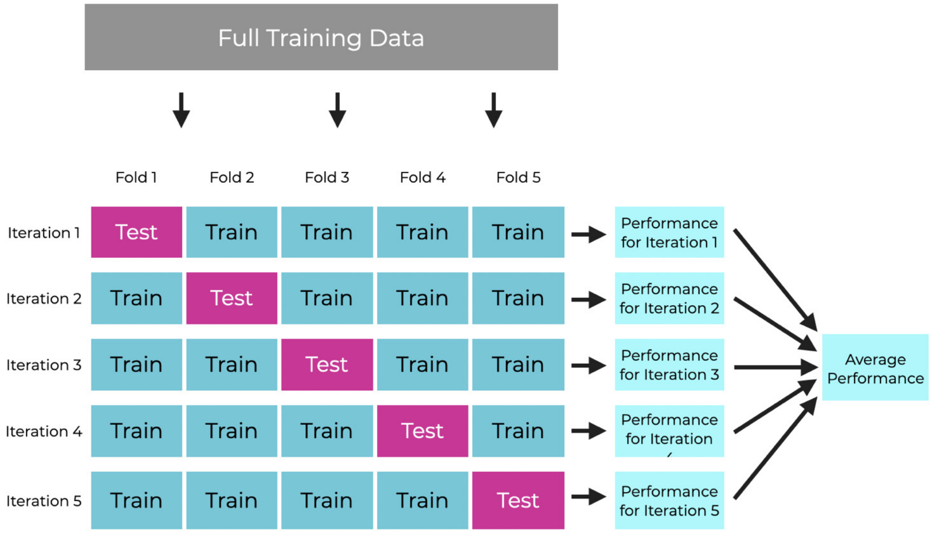
Accuracy

Accuracy is a measure of the overall correctness of the model and is defined as the ratio of correctly predicted instances to the total instances.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

Balanced Accuracy

Balanced Accuracy considers class imbalance by considering the average of sensitivity (true positive rate) across different classes. It is particularly useful when classes are unevenly distributed.



	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	
Iteration 1	Test	Train	Train	Train	Train	Performance for Iteration 1
Iteration 2	Train	Test	Train	Train	Train	Performance for Iteration 2
Iteration 3	Train	Train	Test	Train	Train	Performance for Iteration 3
Iteration 4	Train	Train	Train	Test	Train	Performance for Iteration
Iteration 5	Train	Train	Train	Train	Test	Performance for Iteration 5

Figure 5. Cross validation scheme for model training and validation

$$\text{Balanced Accuracy} = \frac{\text{Sensitivity}_1 + \text{Sensitivity}_2 + \ldots + \text{Sensitivity}_k}{k}$$

Sensitivity (Recall)

Sensitivity, also known as Recall or True Positive Rate, measures the ability of a model to correctly identify positive instances.

$$\text{Sensitivity} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

Specificity

Specificity measures the ability of a model to correctly identify negative instances.

$$\text{Specificity} = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}}$$

Precision

Precision measures the accuracy of positive predictions and is defined as the ratio of true positives to the sum of true positives and false positives.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

F1-Score

F1-Score is the harmonic mean of precision and recall, providing a balanced measure of a model's performance.

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

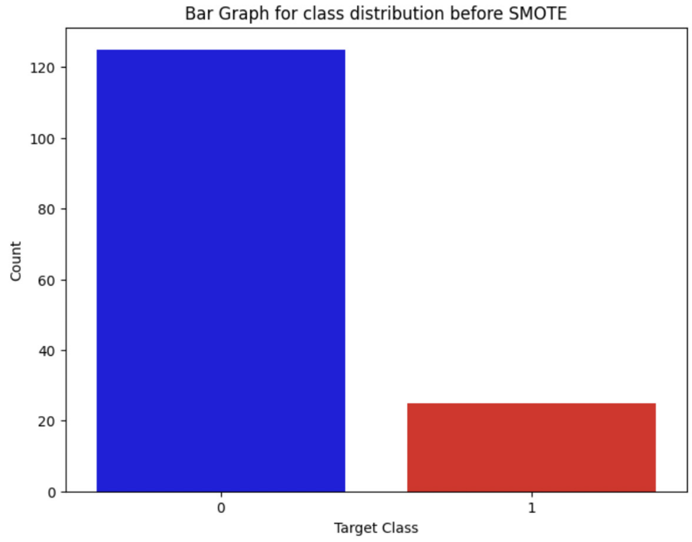


Figure 6. Class distribution of the response variable before SMOTE

Results

Target variable Distribution

The distribution of the target variable, 'obese,' reveals important insights into the composition of the dataset. In our analysis, the target variable represents whether an individual is classified as obese or not. The distribution of this variable is crucial for understanding the prevalence of obesity within the dataset. Upon examining the distribution, we observe that the dataset exhibits an imbalanced distribution in terms of the 'obese' classes. There are two possible classes: 'Not Obese' and 'Obese.' The imbalanced distribution implies that one class significantly outnumbers the other. Figure 6 shows the plot of target variable Distribution before SMOTE.

To address the class imbalance and enhance the model's ability to accurately predict both classes, oversampling techniques such as Synthetic Minority Over-sampling Technique (SMOTE) were employed. SMOTE generates synthetic instances of the minority class by interpolating between existing instances. This augmentation helps balance the class distribution and improves the model's generalization to the minority class, ultimately contributing to more robust predictions. Figure 7 shows the class distribution of the target after SMOTE.

In the subsequent sections, we will assess the impact of before oversampling and after oversampling on model performance through various metrics and visualizations.

Table 2 shows the performance of the classification models on various metrics before SMOTE as percentage. Table 3 shows the performance of the classification models after SMOTE. Table 4 show the metric improvement of the classification models as percentage. Figure 9 shows visualization of metrics deviation after SMOTE and Figure 8 show visualization of the metrics before and after SMOTE.

Discussion

The Logistic Regression model initially performed reasonably well

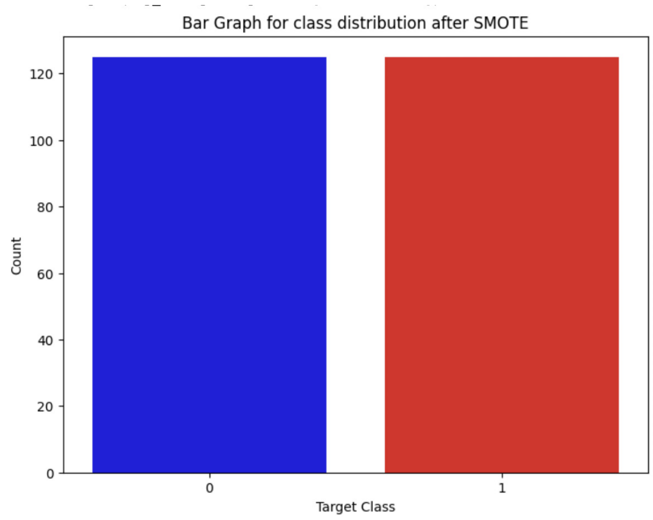


Figure 7. Class distribution of the response variable after SMOTE

Table 2. Performance Metrics Before SMOTE (as percentages)

Model	Accuracy	Balanced Accuracy	Sensitivity	Specificity	Precision	F1-Score
Logistic Regression	80.8	55.8	18.7	18.7	28.3	22.4
Naive Bayes	76.7	62.2	39.3	39.3	35.0	34.4
KNN (k=5)	80.0	48.0	0.0	0.0	0.0	0.0
Deep Learning	76.0	53.4	19.3	87.4	32.4	19.7

Table 3. Performance Metrics After SMOTE (as percentages)

Model	Accuracy	Balanced Accuracy	Sensitivity	Specificity	Precision	F1-Score
Logistic Regression	72.0	72.4	75.4	75.4	71.7	72.2
Naive Bayes	79.0	78.9	87.2	87.2	75.0	80.5
KNN (k=5)	75.5	76.3	96.0	96.0	68.9	79.6
Deep Learning	82.5	82.8	89.1	76.4	78.4	83.2

Table 4. Deviance in Metrics

Model	Accuracy	Balanced Accuracy	Sensitivity	Specificity	Precision	F1-Score
Logistic Regression	-8.8	+16.6	+56.7	+56.7	+43.4	+47.8
Naive Bayes	+2.3	+16.7	+47.9	+47.9	+40.0	+46.1
KNN (k=5)	-4.5	+28.3	+96.0	+96.0	+68.9	+79.6
Deep Learning	+6.5	+29.4	+69.8	-10.9	+45.9	+66.7

before SMOTE, achieving an accuracy of 80.8%. However, after applying SMOTE, there was a notable decrease in accuracy by -8.8%. Despite this decrease, the model's performance in terms of sensitivity and specificity showed significant improvements. Sensitivity, which measures the ability to correctly identify obese individuals, increased substantially by +56.7% after SMOTE, indicating that the model became more effective at capturing true positives within the obese class. Similarly, specificity, representing the ability to correctly identify non-obese individuals, also saw a considerable improvement of +56.7% after SMOTE, suggesting a reduction in the false positive rate among non-obese individuals. These enhancements in sensitivity and specificity demonstrate the efficacy of SMOTE in addressing the class imbalance issue inherent in the dataset and improving the model's ability to classify both classes accurately.

Moreover, the balanced accuracy of the Logistic Regression model increased significantly by +16.6% after SMOTE, indicating a more balanced performance across both classes. The substantial improvement in precision by +43.4% and F1-score by +47.8% after SMOTE further underscores the effectiveness of SMOTE in enhancing the model's predictive capability. Precision reflects the model's ability to correctly classify positive predictions, while the F1-score balances both precision and recall, providing a robust measure of overall model performance. The notable improvements in these metrics suggest that SMOTE facilitated a more accurate and reliable classification of individuals into obese and non-obese categories, contributing to the overall enhancement in model performance.

Overall, the results demonstrate that SMOTE had a significant positive impact on the performance of the Logistic Regression model across various evaluation metrics. While there was a slight decrease in overall accuracy, the model showed substantial improvements in sensitivity, specificity, balanced accuracy, precision, and F1-score after applying SMOTE. These improvements highlight the effectiveness of SMOTE in addressing the class imbalance issue inherent in the dataset, enabling the Logistic Regression model to achieve better classification results and make more accurate predictions regarding individuals' obesity status.

Before applying SMOTE, the Naive Bayes classifier exhibited a moderate level of accuracy, achieving 76.7%. However, after implementing SMOTE, there was a modest increase in accuracy by +2.3%. More notably, the model's performance in terms of sensitivity and specificity experienced substantial improvements. Sensitivity, representing the ability to correctly identify obese individuals, increased significantly by +47.9% after SMOTE. This enhancement indicates that the model became more effective at capturing true positives within the obese class. Similarly, specificity, which measures the ability to correctly identify non-obese individuals, also saw a considerable improvement of +47.9% after SMOTE, suggesting a reduction in false positives among non-obese individuals. These enhancements underscore the effectiveness of SMOTE in addressing the class imbalance issue and improving the Naive Bayes model's classification accuracy.

Moreover, the balanced accuracy of the Naive Bayes model

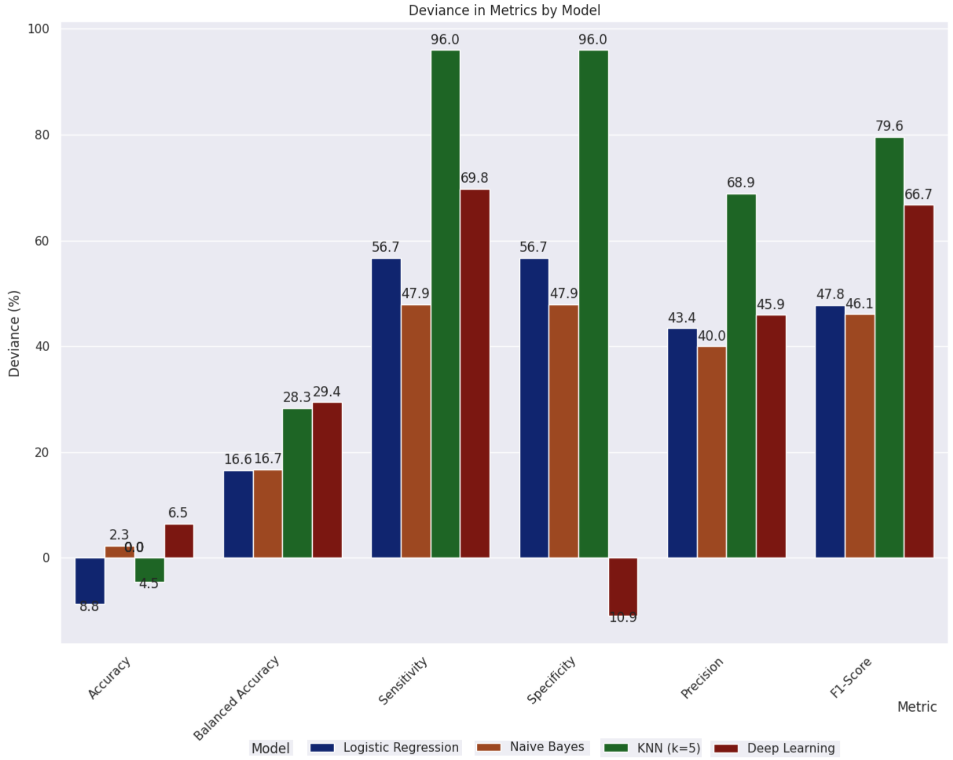


Figure 8. Visualization of metrics deviation after SMOTE

increased notably by +16.7% after SMOTE, indicating a more balanced performance across both classes. Additionally, there were significant improvements in precision by +40.0% and F1-score by +46.1% after SMOTE. Precision reflects the model's ability to correctly classify positive predictions, while the F1-score provides a balanced measure of both precision and recall. The considerable improvements in these metrics suggest that SMOTE facilitated a more accurate and reliable classification of individuals into obese and non-obese categories, contributing to the overall enhancement in model performance.

The application of SMOTE had a positive impact on the performance of the Naive Bayes classifier across various evaluation metrics. While there was a modest increase in accuracy, the model demonstrated substantial improvements in sensitivity, specificity, balanced accuracy, precision, and F1-score after applying SMOTE. These improvements highlight the efficacy of SMOTE in mitigating the effects of class imbalance and improving the Naive Bayes model's ability to accurately classify individuals based on their obesity status.

The KNN (k= 5) classifier exhibited strong sensitivity at 96.0% but struggled with specificity, which was notably low at 0.0%. This indicates its difficulty in correctly identifying non-obese individuals.

Remarkable improvements were observed across all metrics after implementing SMOTE. Both sensitivity and specificity increased significantly by +96.0%, indicating substantial enhancement in correctly identifying both obese and non-obese individuals. This suggests effective mitigation of the class imbalance issue by SMOTE, resulting in better classification accuracy.

Post-SMOTE implementation, the KNN model showcased a transformative improvement in performance. Initially challenged by low specificity, SMOTE led to significant enhancements in sensitivity, specificity, balanced accuracy (+28.3%), precision (+68.9%), and F1-score (+79.6%). These enhancements underscore the effectiveness of SMOTE in addressing class imbalance and improving the KNN model's overall classification performance.

Before the application of SMOTE, the deep learning model exhibited promising performance, particularly in specificity, achieving a high value of 87.4%. However, the model's sensitivity was relatively low, with a value of 19.3%, indicating its struggle to correctly identify obese individuals. After the implementation of SMOTE, there was a notable improvement in the model's performance across most metrics. Sensitivity increased significantly by +69.8%, indicating a substantial enhancement in the model's ability to correctly identify obese individuals. However, there was a slight decrease in specificity by -10.9%, suggesting a higher rate of false positives among non-obese individuals. Despite this slight trade-off, the overall performance of the deep learning model improved significantly after SMOTE.

Moreover, the balanced accuracy of the deep learning model increased substantially by +29.4% after SMOTE, indicating a more balanced performance across both classes. Additionally, there were significant enhancements in precision by +45.9% and F1-score by +66.7% after SMOTE. These improvements reflect the model's improved ability to accurately classify positive predictions and achieve a balance between precision and recall. Despite the slight decrease in specificity, the substantial improvements in sensitivity, balanced accuracy, precision, and F1-score highlight

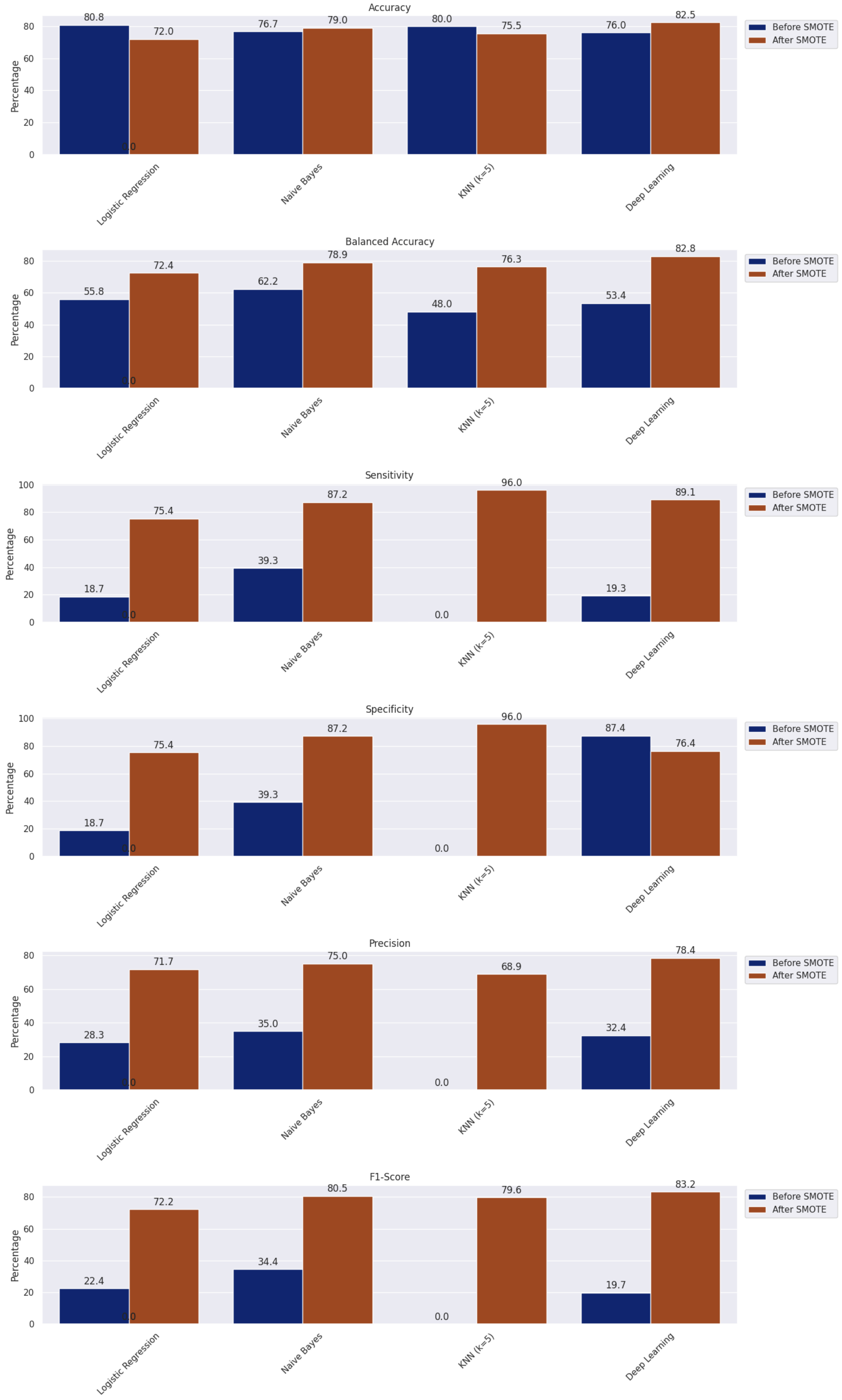


Figure 9. Visualization of metrics before and after SMOTE

the effectiveness of SMOTE in improving the overall classification performance of the deep learning model.

The application of SMOTE had a positive impact on the performance of the deep learning model. While there was a slight trade-off in specificity, the model exhibited significant improvements in sensitivity, balanced accuracy, precision, and F1-score after applying SMOTE. These improvements demonstrate the efficacy of SMOTE in addressing the class imbalance issue and enhancing the deep learning model's ability to accurately classify individuals based on their obesity status.

Comparison with other studies

Tree-based machine learning approaches, including Logistic Regression, Random Forest (RF), and Extreme Gradient Boosting (XGBoost), were employed to classify obesity levels (38). The study found that LR performed best across most metrics after addressing class imbalance using SMOTE-NC and feature selection via Recursive Feature Elimination. Our findings similarly highlight the robustness of LR in the context of obesity prediction, with SMOTE significantly enhancing various performance metrics. A study (9) focused on preprocessing an obesity dataset and used SVM, RF, and Decision Trees for classification. The RF model showed the highest prediction accuracy (96%). This aligns with our findings, which show that RF is effective in obesity prediction, though our use of SMOTE further improved the model's performance metrics. Assessed ML methods, including Logistic Regression, Classification and Regression Trees, and Naive Bayes, to predict obesity (39). Logistic regression showed the highest performance, which is consistent with our study. The application of SMOTE in our research also addressed data imbalance effectively, leading to significant improvements in sensitivity, specificity, and balanced accuracy.

Conclusions

The application of SMOTE has demonstrated significant improvements in the performance of various classification models for predicting obesity status. The results indicate that SMOTE effectively addressed the class imbalance issue present in the dataset, leading to enhanced model performance across multiple evaluation metrics. Specifically, SMOTE substantially improved sensitivity, specificity, balanced accuracy, precision, and F1-score for all models evaluated. This suggests that SMOTE successfully balanced the distribution of the minority class, allowing the models to better capture the underlying patterns and make more accurate predictions.

Despite the overall improvements, there are some caveats to consider. While SMOTE improved the classification performance of most models, it also resulted in a decrease in specificity for the deep learning model. This suggests a higher rate of false positives among non-obese individuals, indicating potential areas for improvement. Additionally, the effectiveness of SMOTE may vary depending on the specific characteristics of the dataset and the chosen classification algorithm. Further experimentation and fine-tuning may be necessary to optimize the performance of the models and address any remaining challenges.

Future research could explore alternative methods for addressing class imbalance issues beyond SMOTE, such as ensemble techniques or data augmentation strategies. Additionally, conducting a more comprehensive analysis of feature importance and model interpretability could provide valuable insights into the factors influencing obesity prediction and help identify areas for further refinement. Overall, while SMOTE has proven to be a valuable tool for improving classification performance in imbalanced datasets, continued research, and experimentation are essential to further advance the field of obesity prediction and enhance the effectiveness of predictive models in healthcare applications.

Abbreviations

ANN	Artificial Neural Network
BMI	Body Mass Index
KNN	K-Nearest Neighbors
SVM	Support Vector Machine
SMOTE	Synthetic Minority Over-sampling Technique
ML	Machine Learning

References

1. Peter G Kopelman. Obesity as a medical problem. Nature. 2000; 404(6778): 635-643.

2. Ng M, Fleming T, Robinson M, Thomson B, Graetz N, Margono C, et al. Global, regional, and national prevalence of overweight and obesity in children and adults during 1980-2013: a systematic analysis for the global burden of disease study 2013. Lancet. 2014; 384(9945): 766-781.

3. Omer T. The causes of obesity: an in-depth review. Adv Obes Weight Manag Control. 2020; 10(4): 90-94.

4. Aljanabi M, Qutqut MH, Hijjawi M. Machine learning classification techniques for heart disease prediction: a review. Internat J Engineer Technol. 2018; 7(4): 5373-5379.

5. Al-Hashem MA, Alqudah AM, Qananwah Q. Performance evaluation of different machine learning classification algorithms for disease diagnosis. Internat J E-Health Med Communicat. 2021; 12(6):1-28.

6. An Q, Rahman S, Zhou J, Kang JJ. A comprehensive review on machine learning in healthcare industry: Classification, restrictions, opportunities and challenges. Sensors. 2023; 23(9): 4178.

7. Safaei M, Sundararajan EA, Driss M, Boulila W, Shapi&apos;I A. A systematic literature review on obesity: Understanding the causes &amp; consequences of obesity and reviewing various machine learning approaches used to predict obesity. Computers Biol Med. 2021; 136: 104754.

8. Ferdowsy F, Alam RKS, Jabiullah I, Habib T. A machine learning approach for obesity risk prediction. Current Res Behavioral Sci. 2021; 2: 100053.

9. Astuti TS, Sidik AD, Kuswanto H, Lawi A, Nasir S. Predicting obesity in adults using machine learning techniques: an analysis of Indonesian basic health research 2018. Frontiers Nutrition. 2021; 8: 669155.

10. Curbelo MCA, Fergus P, Hussain A, Al-Jumeily D, Abdulaimma B, Hind J, Radi N. Machine learning approaches for the prediction of obesity using publicly available genetic profiles. In 2017 International Joint Conference on Neural Networks (IJCNN), pages 2743-2750. IEEE, 2017.

11. Zheng Z, Ruggiero K. Using machine learning to predict obesity in high school students. In 2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pages 2132-2138. IEEE, 2017.

12. Cheng X, Lin S-Y, Liu J, Liu S, Zhang J, Nie P, et al. Does physical activity predict obesity-a machine learning and statistical method-based analysis. Internat J Environm Res Public Health. 2021; 18(8): 3966.

13. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. Smote: synthetic minority over-sampling technique. J Artificial Intelligence Research. 2002; 16: 321-357.

14. Kosolwattana T, Liu C, Hu R, Han S, Chen H, Lin Y. A self-inspected adaptive smote algorithm (sasmote) for highly imbalanced data classification in healthcare. BioData Mining. 2023; 16(1): 15.

15. Fernández A, Garcia S, Herrera F, Chawla NV. Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. J Artificial Intelligence Res. 2018; 61: 863-905.

16. Blagus R, Lara L. Smote for high-dimensional class-imbalanced data. BMC bioinformatics. 2013; 14: 1-16.

17. Han H, Wang W-Y, Mao B-H. Borderline-smote: a new over-sampling method in imbalanced data sets learning. International conference on intelligent computing, pages 878-887. Springer; 2005.

18. Sreejith S, Khanna NH, Kannan A. Clinical data classification using an enhanced smote and chaotic evolutionary feature selection. Computers Biology Medicine. 2020; 126: 103991.

19. Elreedy D, Atiya AF. A comprehensive analysis of synthetic minority oversampling technique (smote) for handling class imbalance. Information Sciences. 2019; 505: 32-64.

20. Ismail E, Gad W, Hashem M. A hybrid stacking-smote model for optimizing the prediction of autistic genes. BMC bioinformatics. 2023; 24(1): 379.

21. Wang L. Imbalanced credit risk prediction based on smote and multi-kernel FCM improved by particle swarm optimization. Applied Soft Computing. 2022; 114: 108153.

22. Yee CPC, Yang Y, Giin LB. Enhancing financial fraud detection through addressing class imbalance using hybrid smote-gan techniques. Internat J Financial Studies. 2023; 11(3): 110.

23. Li H, Liu H, Hu Y. Prediction of unbalanced financial risk based on gra-topsis and smote-cnn. Scientific Programming. 2022; 2022(1): 8074516.

24. Özdemir A, Polat K, Alhudhaif A. Classification of imbalanced hyperspectral im- ages using smote-based deep learning methods. Expert Systems Applications. 2021; 178: 114986.

25. Chamseddine E, Mansouri N, Soui M, Abed M. Handling class imbalance in covid-19 chest x-ray images classification: Using smote and weighted loss. Applied Soft Computing. 2022; 129: 109588.

26. Sami JA. Heart disease prediction system using (smote technique) balanced dataset and decision tree classifier. AIP Conference Proceedings, volume 2834. AIP Publishing; 2023.

27. Prasad PS, Sreedevi M. An improved prediction of kidney disease using smote. Indian J Sci Technol. 2016; 9(31): 1-7.

28. Sowjanya AM, Mrudula O. Effective treatment of imbalanced datasets in health care using modified smote coupled with stacked deep learning algorithms. Applied Nanoscience. 2023; 13(3): 1829-1840.

29. Nasteski V. An overview of the supervised machine learning methods. Horizons b. 2017; 4: 51-62.

30. Peterson LE. K-nearest neighbor. Scholarpedia. 2009; 4(2): 1883.

31. Cunningham P, Delany SJ. k-nearest neighbour classifiers-a tutorial. ACM computing surveys. 2021; 54(6): 1-25.

32. Rish I. An empirical study of the naive bayes classifier. IJCAI 2001 workshop on empirical methods in artificial intelligence; 2001.

33. Goodfellow I, Bengio Y, Courville A. Deep learning. MIT press; 2016.

34. Stevens E, Antiga L, Viehmann T. Deep learning with PyTorch. Manning Publications; 2020.

35. Kofi NI, Nyarko-Boateng O, Aning J, et al. Performance of machine learning algorithms with different k values in k-fold crossvalidation. Internat J Information Technol Computer Sci. 2021; 13(6): 61-71.

36. Tamilarasi P, Rani RU. Diagnosis of crime rate against women using k-fold cross validation through machine learning. 2020 fourth international conference on computing methodologies and communication (ICCMC). IEEE; 2020.

37. Misra P, Singh YA. Improving the classification accuracy using recursive feature elimination with cross-validation. Int J Emerg Technol. 2020; 11(3): 659-665.

38. Ram DR, Mukherjee I, Chakraborty C. Obesity disease risk prediction using machine learning. Internat J Data Sci Analytics. 2024. Doi: 0.1007/s41060-023-00491-9.

39. Ab MNL, Anuar S. Machine learning modelling for imbalanced dataset: Case study of adolescent obesity in malaysia. J Adv Res Applied Sci Engineer Technol. 2023; 36(1): 189-202.