International Journal of Scientific Research in Engineering & Technology
Volume6, Issue2 (March-April 2026), PP: 175-189.
https://www.doi.org/10.59256/ijsreat.20260602024
www.ijsreat.com
ISSN No: 2583-1240

Stacked ensemble learning with XAI for Accurate Obesity level Prediction

G.M.G. Madhuri[1], Tumu Venkata Prasanna Lakshmi[2], Simhadri Harsha Vardhan[3], Pasala Preethi[4], Tipparti Jahnavi[5], Salagala Seshu[6]

[1]Assistant Professor, Department of Computer Science and Engineering Dhanekula Institute of Engineering and Technology Ganguru, India.

[2, 3,4,5,6] Department of Computer Science and Engineering Dhanekula Institute of Engineering and Technology Ganguru, India.

To Cite this Article: *G.M.G. Madhuri [1], Tumu Venkata Prasanna Lakshmi[2], Simhadri Harsha Vardhan[3], Pasala Preethi[4], Tipparti Jahnavi[5], Salagala Seshu[6], "Stacked ensemble learning with XAI for Accurate Obesity level Prediction", International Journal of Scientific Research in Engineering & Technology, Volume 06, Issue 02, March-April 2026, PP: 175-189.*

Abstract: *Obesity has emerged as one of the most urgent public health challenges of our time, and identifying its contributing risk factors at an early stage is vital for preventing the wide range of associated complications. Despite considerable research directed at classifying obesity levels, many existing approaches fall short in terms of consistency and trustworthiness, largely because they rarely incorporate explainable artificial intelligence (XAI). This work presents a machine learning framework that brings together predictive accuracy and model transparency through the use of XAI techniques. Our model is trained on a widely-used dataset assembled by Palechor and Manotas, available through the UCI Machine Learning Repository, which captures both physical measurements and self-reported lifestyle behaviours of individuals. At the heart of our methodology is a stacking-based ensemble that draws on the complementary strengths of four base classifiers — Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), XGBoost, and Multilayer Perceptron (MLP) — whose combined outputs feed into a meta-learner for the final classification. To shed light on how the model arrives at its decisions, we apply two well-regarded XAI methods: Local Interpretable Model-agnostic Explanations (LIME) and SHapley Additive exPlanations (SHAP). Together, these tools make feature contributions visible and interpretable, which is especially valuable in health-related settings where clinicians need to understand and trust model outputs. The result is a framework that not only performs well but also supports more informed, personalised obesity risk management.*

Key Words: *Obesity, machine learning, stacking, explainable AI.*

I. INTRODUCTION

Obesity affects our bodies the air we breathe and how we live our lives. It is a problem that can cause a lot of harm leading to conditions like heart disease, stroke, type 2 diabetes and even certain cancers. These problems make obesity a big threat to our health around the world. In words obesity means having too much fat in our body. Unfortunately no matter where we live the number of people dealing with obesity keeps going up. To find out if someone is obese doctors often look at something called BMI. A calculation based on a persons height and weight. If your BMI is than 25 you are considered overweight. If it is than 30 that is considered obese. However this way of measuring is not perfect. It does not differentiate between muscle and fat. So a strong athlete might have a BMI and be considered obese even if they have very little body fat.

Also factors like age, gender, ethnicity and lifestyle are not considered by BMI. These things have an impact on our health. Nonetheless than four million people die every year because of obesity. That is a big number of lives lost. People are spending time sitting around nowadays which is part of the reason why obesity rates keep going up.. It is not just because people are sitting more. Not sleeping well and having access to unhealthy food also play a big role. In Saudi Arabia this problem is really bad. A huge 70% of the people there are overweight with than a third of them struggling with obesity. According to the World Health Organization obesity is one of the health problems in the world. In the United States dealing with obesity uses up about 10% of all the money spent on healthcare.

On a positive note making a routine that includes regular exercise and healthier eating habits can help fight this problem. Eating a diet can really reduce the chances of becoming obese. Today scientists have a lot of tools to help them research obesity. They can look at a lot of data, such as health records, insurance information and even information from fitness trackers on smartphones. This allows them to find trends in obesity earlier and step in when needed.

In our research we looked at information from 2,111 people aged 14 to 61 from Peru, Mexico and Colombia. We considered sixteen things like body measurements, habits and diet details. Some of which were reported directly by the people themselves. We started by cleaning up the data. We turned categories into numbers. Made sure all the values were the same. To make sure each group had data we used a special technique to add more data. Next we used a method to narrow down to fourteen key things that affect obesity. To predict obesity we made a model that used four different machine learning models: XGBoost,

Multilayer Perceptron, Quadratic Discriminant Analysis and Linear Discriminant Analysis. Each model made predictions on its own. While some models are good at finding boundaries, XGBoost and Multilayer Perceptron are good at finding patterns. By combining the predictions from these models and using a technique to link them the system becomes more accurate and reliable than using just one approach. We tested our strategy with all the metrics and the results were really good: our model was accurate 98.92% of the time. It also scored high in precision, recall and F1. This is better than methods. However being accurate is not enough in healthcare; people need to understand why a model makes a prediction. Therefore we used two tools: LIME, which explains each prediction by making a model for it and SHAP for more insights. This approach uses game theory to figure out how each feature affects a prediction. By using these methods doctors and clinicians can see what influences the models decisions. This clarity helps build trust and makes it easier to use the models insights. When it comes to understanding obesity, research and models like ours play a role. They help people make decisions, about their health. Obesity is a challenge. However using the tools and strategies can lead to real change.

The key contributions of this study are summarized as follows:

1. I made a system to prepare my data. This system does the following It converts features into a format. It scales the data using Min-Max normalization.
2. I created a team of models that work together. This team, called a stacking model includes: LDA, QDA, XGBoost, MLP. These models work together under a main model called a meta-classifier. This team works better than any model. It has classification accuracy.
3. I used two methods, LIME and SHAP to explain how my model makes decisions. LIME helps to understand why the model made a prediction.
4. I optimized the models settings in a way. I used search to try out different settings. I tested all the settings, on a test set to confirm the results.

II.PREVIOUS STUDY

Earlier research did not use methods that make Artificial Intelligence easier to understand. Now people are starting to use Explainable Artificial Intelligence in similar areas. The main goal of Explainable Artificial Intelligence is to make Machine Learning models clearer because they are often thought of as boxes. This makes them more reliable. This change is very important in healthcare, where ethics and clear decision making're crucial. People have started using techniques like Local Interpretable Model- Explanations and SHapley Additive exPlanations.

Kibria and colleagues explored ways to predict diabetes using a combination of six Machine Learning models. They used methods like Local Interpretable Model- Explanations and SHapley Additive exPlanations to understand how the models worked. Raihan and team used an algorithm to get a better view of how their model made decisions. They used this to predict kidney disease. SHapley Additive exPlanations helped them see how different features affected the predictions.

Jahan and team worked on predicting Alzheimers disease by looking at types of data. They used SHapley Additive exPlanations to understand the models outputs. This showed how important it is to make Artificial Intelligence systems understandable in medicine.

Research on obesity classification uses different datasets and Machine Learning methods. Some studies use data from places like Mexico, Peru, Colombia, Bangladesh and Indonesia. Others use combination or hybrid models. Lately people have been using Explainable Artificial Intelligence tools like Local Interpretable Model- Explanations and SHapley Additive exPlanations. This makes healthcare applications more transparent and easier to understand. This progress is necessary, for advancing healthcare and making Explainable Artificial Intelligence a key part of it.

III.MATERIALS AND METHODS

In this part you will find a description of the data and the steps taken before starting the modeling process. The data and the steps taken before starting the modeling process are explained here. After that there is an explanation of the architecture and details about the computing environment used for the experiments on the ensemble architecture and the computing environment. The ensemble architecture and the computing environment are explained in detail. The LIME-based explanations of the data and the ensemble architecture are created after the performance evaluation of the data and the ensemble architecture.

A. Data Collection

` The research uses the obesity dataset from the UCI Machine Learning Repository [22]. This obesity dataset has a lot of information. It has 2,111 entries about people. The obesity dataset tells us about the people in it like what they're like and what they do every day. The obesity dataset has things like gender and age and height and weight. It also tells us what people do and how they live. To see all the details look at Table 1. The obesity dataset is really useful for the research, on obesity. The obesity dataset is what we are using for this.

B. Data Exploitation Analysis

We will begin by looking at our data to find information for the study. Our dataset is complete with no information. Look at Figure 2. It shows how different values are spread out across each feature. The x-axis shows the features and the y-axis shows how unique entries each feature has. Features with unique values are probably categories and those with many unique entries are considered numbers or continuous. Bars show these features in a visual way. Among them Height has the unique values and NCP has the least. Categories help us understand our analysis better. Numerical features measure things we can count and other types give us information. Both are necessary for our model to learn and work well. For example people who are the height and

weight might be in different obesity categories because of their gender. This shows why we need to include both types of variables in our analysis. In this dataset NO be yes dad is the variable we are looking at which represents levels of obesity. You can see in Table 2 that the data is divided into seven categories: Normal Weight, Overweight Level I, Overweight Level II Insufficient Weight, Obesity Type I, Obesity Type II and Obesity Type III. The samples are not evenly distributed among these categories, which means some categories have samples than others. Notably Obesity Type I has the samples compared to the others. Insufficient Weight has the samples.

Figure 3 shows a comparison of eight numerical features using boxplots. These boxplots help us see how each variable is spread out where the middle point is and how much it varies. Age and Weight stand out because they have a range and more variation than the other features. On the hand things like how often people eat vegetables how often they exercise and how much time they spend on technology do not vary as much. Points that are away from the rest of the data highlight unusual values in variables like Age and NCP but these are not extreme cases. There are also some values in Height and Weight. Even though there are unusual values overall none of them seem very unusual or alarming. The data was treated differently. Of being removed it was normalized during the preparation step. This helped reduce the impact of values while keeping the data complete.

You can see the correlation matrix for the features in Figure 4 which uses Pearson correlation coefficients to show the relationships among eight variables: Age, Height, Weight how often people eat vegetables NCP, water intake how often people exercise and how much time they spend on technology. Dark colors mean strong relationships and light colors mean ones. The diagonal shows relationships with a value of 1. Among all pairs Height and Weight have the relationship with a coefficient of 0.46, which means they have a moderate linear relationship. For pairs of features the relationships are weak which means there are no strong linear connections, between them.

Features	Description
Family_history_with_overweight	Participant's family history with overweight
FAVC	Frequent Consumption of high caloric food
FCVC	Frequent consumption of vegetables
NCP	Number of main meals
CACE	Consumption of food between meals
SMOKE	If the participant smokes
CH2O	Consumption of water daily
SCC	Calories consumption monitoring
FAF	Physical activity frequency
TUE	Time using technology device
CALC	Consumption of alcohol
MTRANS	Transportation used
NObeyesdad	Obesity level

Table 1. Description of The Features of the Dataset.

C. Preprocessing Stages

Before we started making a model we had to get the raw dataset ready. We did this by changing it in a ways so that machine learning algorithms could work with it easily. We had to turn the -numeric things into numbers. We also had to make sure the continuous features were on the scale. The dataset had some problems with class frequencies being uneven so we had to fix that. We looked at all the features. Picked the ones that were most important for the model to learn from.

Each of these steps was important for making a model. The raw dataset had to be changed so that machine learning algorithms could handle it well. We did all these things to the dataset to get it ready, for the model.

1. Data Mapping

We had a lot of information in the dataset that was not just numbers. Some of it was text or a mix of things. We had to change this information into numbers so we could use it in our models.

We took things like gender, family history with overweight, FAVC, SMOKE and SCC. Turned them into simple yes or no answers, which were 0 or 1. We had some features like CAEC and CALC that had levels. We gave these four numbers from 0 to 3. We also had information about how people got which was listed under MTRANS. We turned this information into numbers too.

The main thing we were trying to figure out which was NObeyesdad was given a number from 0, to 6. This was because there were seven categories of obesity.

You can see how we did all of this in Table 3.

2. Data Normalization

So we want to make sure all the features in our model are on the scale. This helps our model work better and be more stable when we are training it. The dataset we are using has a mix of numerical attributes. The numerical features, like Age, Height, Weight, FCVC, NCP, CH2O, FAF and TUE all have ranges of values.

To deal with this issue we used something called Min-Max scaling. This transforms the features into a standard range

from 0 to 1. We did this to the features, such, as Age, Height, Weight, FCVC, NCP, CH2O, FAF and TUE to get them all on the same scale. This transformation is defined by the formula:

$$X_{scaled} = \frac{x - \min(X)}{\max(X) - \min(X)}$$

In this equation(1), x x represents the original feature value, while min(X) and max (X) correspond to the minimum and maximum values of that feature within the dataset. The resulting x scaled x scaled is the normalized value.

In Equation (1), x = 72 x=72 denotes the original feature value, with min(X)=39 and max (X)= 165.06; consequently, the normalized value x scaled x scaled is approximately 0.26.

Class name	Number of Samples
Insufficient_Weight	272
Normal_Weight	287
OverWeight_Level_I	290
Overweight_Level_II	290
Obesity_Type_I	351
Obesity_Type_II	297
Obesity_Type_III	324

Table 2. Distribution Of Occurrences Among All Classes

3. Data Splitting

The dataset was split into two parts: 80% for training and 20% for testing. This 80-20 split is commonly used in machine learning. It is also used in studies. To keep the subsets similar we used sampling. This way the test set has a class distribution, to the overall dataset.

4. Data Balancing

When we looked at how the classes were divided we saw that some categories of obesity had a lot information than others. If we do not fix this it can cause problems because the classifiers will likely pick the category that shows up the most, which makes them less accurate for categories that do not show up much. To make this better we used the Synthetic Minority Oversampling Technique, which is also known as SMOTE. The SMOTE method creates examples for the categories that do not have enough information by looking at the space between the actual data points instead of just making copies of them. It is important to note that we only used this method on the training set and we did not touch the test set. The results we get will be, like what we would see in real life. Our dataset has both categories and numbers so we picked the SMOTENC version from the Imbalanced-Learn library. This version works with different types of data.

5. Feature Selection

When we are working with a dataset not every feature is going to be helpful for what we're trying to do. If we include features that're not necessary or are not relevant it can take longer to train our model. It can also make our model overfit. Hide the patterns that are really there.

To pick the features that're the most useful we used something called Recursive Feature Elimination with Cross-Validation or RFECV for short. This is a process that works in steps. Each time it goes through a step it gets rid of the feature that's the least important based on how well our model can classify things. It does this by trying out combinations of features and seeing which ones work best.

A practical advantage of RFECV over fixed-threshold selection methods is that the optimal number of features does not need to be decided in advance — the algorithm determines it empirically. LDA served as the base estimator throughout this process. Its assumption of a shared covariance structure across classes leads to linear separating boundaries, and its behaviour is well-understood for structured tabular data of the kind found here, where features like weight and height are known a priori to be meaningful predictors of obesity category.

To ensure comprehensive evaluation, the minimum number of selected features was set to one, allowing all possible subsets to be considered. Additionally, stratified k-fold cross-validation with 5 folds was used to maintain balanced class distributions during the feature selection process. In each iteration, one feature was removed, and model performance was evaluated using cross-validation accuracy. As shown in Fig. 6, the highest cross-validation scores were achieved when 14 features were retained, indicating that these features significantly influence model performance. Based on this process, two features--"SMOKE" (smoking habit) and "MTRANS" (mode of transportation)—were eliminated. The remaining 14 features were then used for training and testing, as presented in Table 1. Fig. 7 illustrates the relative contribution of the selected features based on the LDA model. It can be observed that "Weight," "Height," "Age," and "family_history_with_overweight" have a stronger influence on the predictions, while the remaining features contribute to a lesser extent. However, none of the selected features were removed further, as Fig. 6 demonstrates that all 14 features collectively contribute to achieving higher cross-validation accuracy.

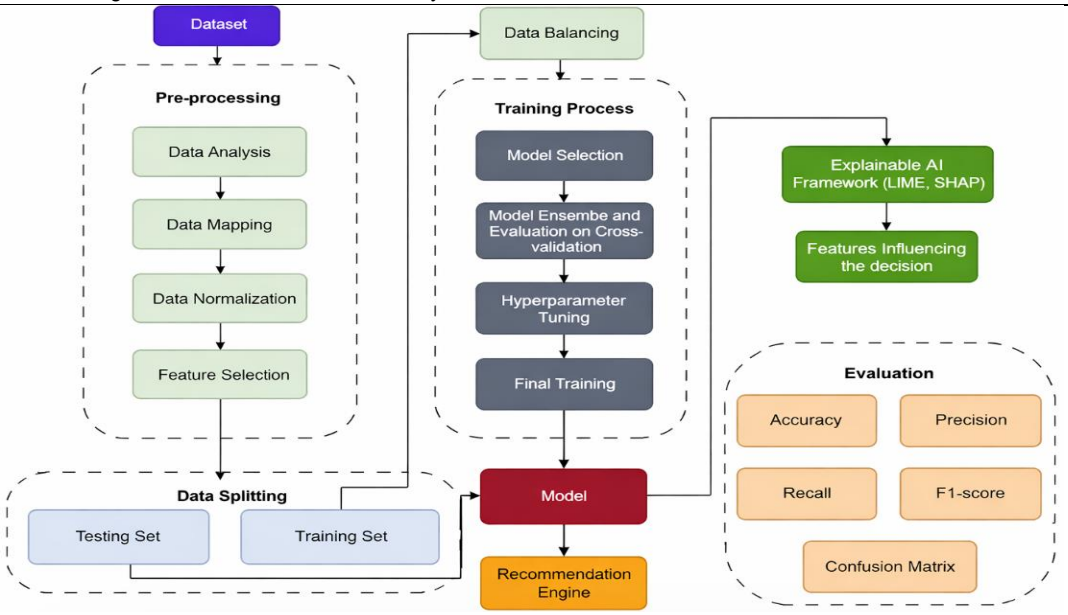


Figure 1. Flowchart Illustrating The Entire Workflow, Including Data Preprocessing, Model Training, Evaluation, And Explainability With Lime

A practical advantage of RFECV over fixed-threshold selection methods is that the optimal number of features does not need to be decided in advance — the algorithm determines it empirically. LDA served as the base estimator throughout this process. Its assumption of a shared covariance structure across classes leads to linear separating boundaries, and its behaviour is well-understood for structured tabular data of the kind found here, where features like weight and height are known a priori to be meaningful predictors of obesity category.

To ensure comprehensive evaluation, the minimum number of selected features was set to one, allowing all possible subsets to be considered. Additionally, stratified k-fold cross-validation with 5 folds was used to maintain balanced class distributions during the feature selection process. In each iteration, one feature was removed, and model performance was evaluated using cross-validation accuracy. As shown in Fig. 6, the highest cross-validation scores were achieved when 14 features were retained, indicating that these features significantly influence model performance. Based on this process, two features--"SMOKE" (smoking habit) and "MTRANS" (mode of transportation)—were eliminated. The remaining 14 features were then used for training and testing, as presented in Table 1. Fig. 7 illustrates the relative contribution of the selected features based on the LDA model. It can be observed that "Weight," "Height," "Age," and "family_history_with_overweight" have a stronger influence on the predictions, while the remaining features contribute to a lesser extent. However, none of the selected features were removed further, as Fig. 6 demonstrates that all 14 features collectively contribute to achieving higher cross-validation accuracy.

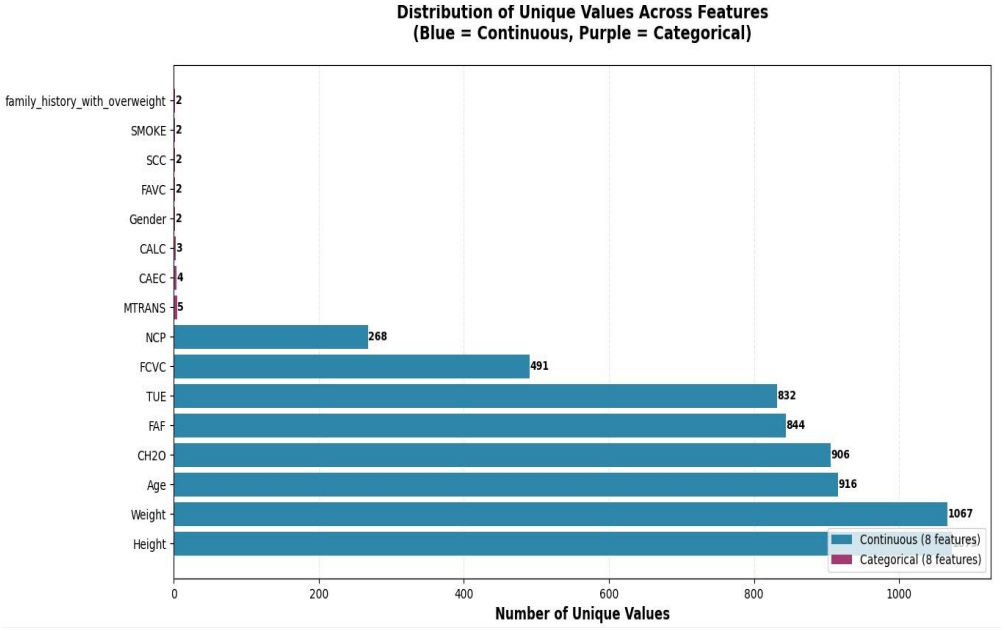


Figure 2. Distribution Of Unique Values Across Features. Bars Represent Continuous Variables And The Remaining (Gender, Family_History_With_Overweight, FAVC, CAEC, SMOKE, SCC, CALC, MTRANS And Nobeyesdad) Indicate Categorical Variables.

Features

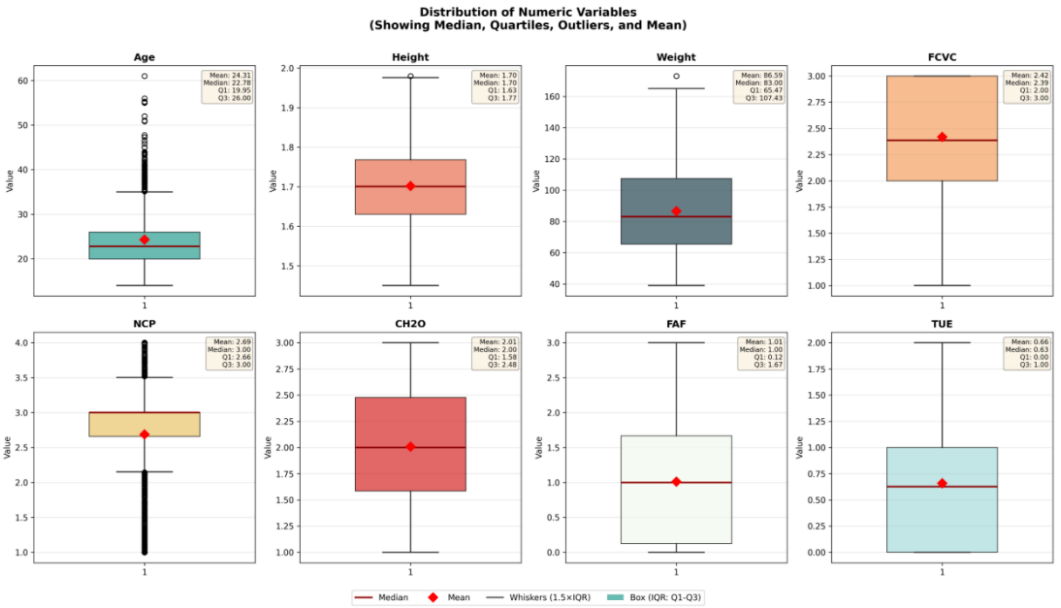


Figure 3. Boxplots illustrating the distribution of values of eight numeric variables. The plots visually represent the distribution of each variable, including median, quartiles, and potential outliers.

D. Stacking Method

In machine learning, stacking is an ensemble learning approach in which multiple classification or regression models are combined to improve predictive performance.

The outputs generated by these individual models are used as inputs to another model, referred to as a meta-classifier, which produces the final prediction. This technique enables the integration of several models into a unified framework, thereby enhancing accuracy and robustness.

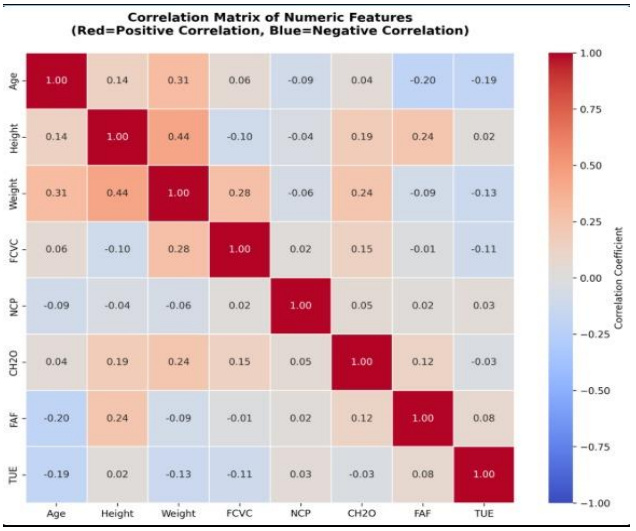


Figure 4. Correlation matrix of eight numeric features. The color intensity represents the strength and direction of the correlation, with deeper color indicating positive correlations and lighter color indicating negative correlations.

The component models are different on purpose. Different algorithms make types of mistakes. A tuned meta-learner can use these differences to make the whole system more reliable.

We chose four base learners for the ensemble: LDA, QDA, XGBoost, MLP. Each one works in a way. LDA uses lines to separate classes. XGBoost and MLP can handle relationships. Each one did well on its own with the training data. We picked the meta-classifier by testing configurations. We reported these tests in Section IV. The meta-classifier was trained on the outputs of all four base learners. Section IV has all the details, about how we set up and evaluated the experiment. The procedure unfolds in two stages. In the first stage, each of the four base learners (the level-1 models) is trained on the prepared dataset. Their probabilistic predictions are then pooled and, together with the original feature values, fed as inputs to the meta-classifier in the second stage. Concatenating the original features with the base models' outputs gave the meta-learner access to both the raw data and the higher-level representations produced by each component, which we found empirically to improve calibration. The entire pipeline was implemented using Scikit-learn, and Fig. 8 provides a visual summary of the architecture. Brief descriptions of the four base models follow below:

1) **Linear Discriminant Analysis (LDA)** is a well-established supervised learning method that serves dual purposes: it classifies observations while simultaneously reducing the dimensionality of the input space. The key idea is to find a low-dimensional projection that maximises the separation between class means relative to within-class spread. LDA operates under the assumption that each class follows a Gaussian distribution and that all classes share a single covariance matrix.

Feature	**Map**
Gender	'Male':0 'Female':1
Family_history_with_overweight	'no':0 'yes':1
FAVC	'no':0 'yes':1
CAEC	'no':0 'sometimes':1 'frequently':2 'always':3
SMOKE	'no':0 'yes':1
SCC	'no':0 'sometimes':1 'frequently':2 'always':3
MTRANS	'walking':0 'Bike':1 'Moterbike':2 'Public_Transport':3 'Automobile':4
NObeyesdad	'Insufficient_weight':0 'Normal_weight':1 'Overweight_Level_I':2 'Overweight_Level_II':3 'Obesity_Type_I':4 'Obesity_Type_II':5 'Obesity_Type_III':6

Table 3. Data Mapping

More formally, LDA seeks the projection vector w that maximises the ratio of between-class scatter to within-class scatter, expressed as

$$J(w) = \frac{W^T S_B\ w}{W^T S_W\ w} \qquad \text{----- (3)}$$

where w w denotes the projection vector, S_B represents the between-class scatter matrix, and S_W is the within-class scatter matrix. The projection vector that solves this problem is:

$$w = S_w^{-1}\ (\mu 1 - \mu 2) \qquad \text{------(4)}$$

where μ 1 μ 1 and μ 2 are the mean vectors of the respective classes, and $S\ W - 1$ S W −1 is the inverse of the within-class scatter matrix. For classification, LDA employs a discriminant function given by

$$\delta_k(x) = X^T \sum_{\mu_k}^{-1} \frac{1}{2} \mu_k^T\ \sum_{\mu_k}^{-1} \log \pi_k \text{ ----(5)}$$

where $\delta_k(x)$ is the discriminant score for class k, Σ denotes the shared covariance matrix, μ_k is the class mean vector, and π_K represents the prior probability of class k. Thus, LDA provides an efficient framework for classification by maximizing inter-class separability while minimizing intra-class variability, making it suitable for various pattern recognition and machine learning applications.

2) Quadratic Discriminant Analysis (QDA):

Quadratic Discriminant Analysis relaxes the rule that all classes must share the covariance. This means each class in Quadratic Discriminant Analysis can have its covariance matrix. The result of this is that Quadratic Discriminant Analysis has flexible decision boundaries. These decision boundaries are not lines like in Linear Discriminant Analysis they are curves. This makes Quadratic Discriminant Analysis better at dealing with classes that have different shapes or sizes. However Quadratic Discriminant Analysis needs to know things to work well. So it works best when we have a lot of training data, for Discriminant Analysis.

The class-specific discriminant function takes the form $\delta_k(x) = -\frac{1}{2}\log|\sum_k| - \frac{1}{2}(x - \mu_k)^T \sum_k^{-1}(x - \mu_k) + \log \pi_k$ ----(6) where $\delta_k(x)$ represents the discriminant score for class k, and a data point x is assigned to the class with the highest score.

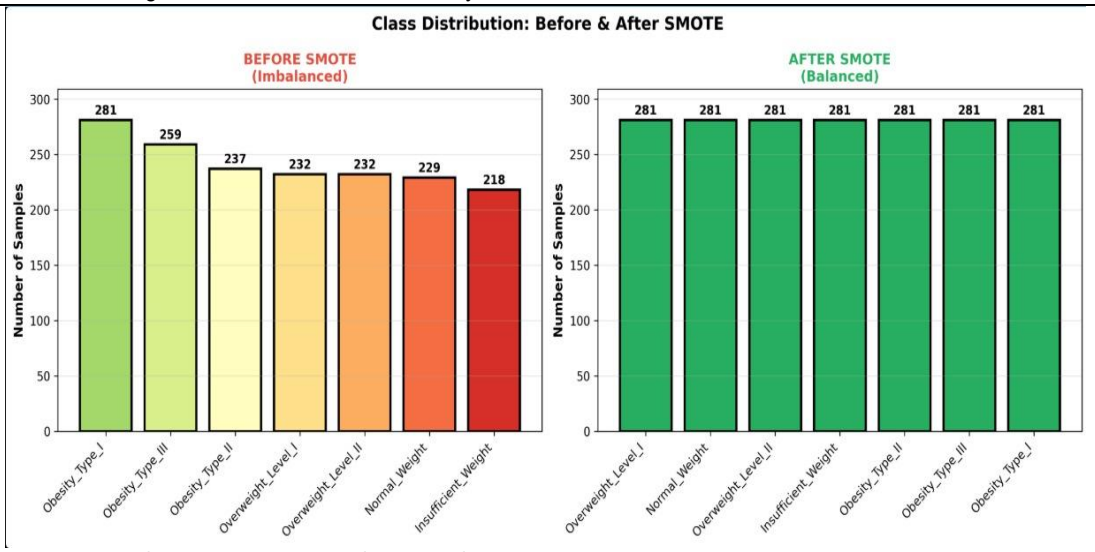


Figure 5. Comparison of class distributions before and after applying the Synthetic Minority Oversampling Technique (SMOTE). The original dataset exhibits a significant class imbalance, with 'Obesity_Type_I' being the most frequent class. After SMOTE, all classes are balanced with an equal number of instances.

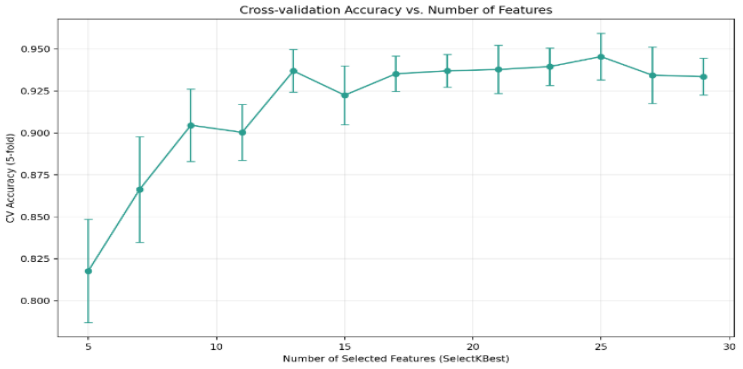


Figure 6. Cross-validation accuracy for different number of features

In this equation the mean vector of class k is what μk is. The covariance matrix for that class is what Σk is. The determinant of the covariance matrix is what |Σk| is. This determinant shows how spread out the data is.

The expression -1/2 〖(x. M_k)〗 ^T ∑_k^(-1)▒〖(x-μ_K) 〗 is the Mahalanobis distance. This distance measures how far the data point x is from the vector of class k. It also takes into account the covariance structure. The prior probability of class k is what πk is.

The first term is like a penalty for classes that have a lot of variance. The second term measures the distance between the data point x. The center of class k. The third term uses what we already know about the probabilities of each class. The mean vector of class k, the covariance matrix of class k and the prior probability of class k all work together. They help QDA make decision boundaries. This means QDA can classify data that has distributions for each class. The mean vector of class k and the covariance matrix of class k are important, for this.

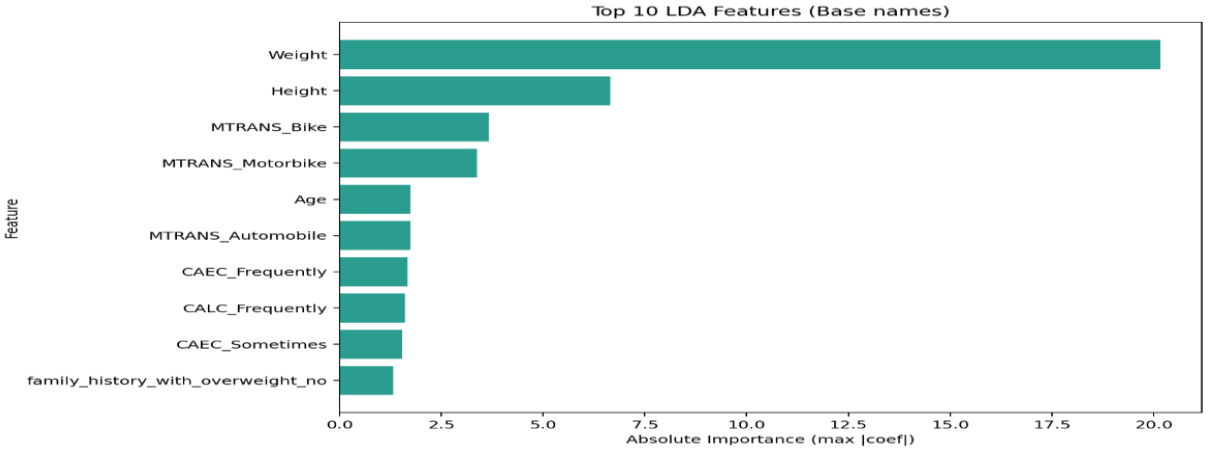


Figure 7. Absolute importance of the different selected features for the Linear Discriminant Analysis model.

Extreme Gradient Boosting (XGBoost): XGBoost is a good way to make a team of decision trees work together. It does this by building each tree in a line and each tree looks at the mistakes the trees before it made. This helps the next tree do a job so over time the mistakes get smaller and smaller. XGBoost also has some built in safety nets to stop it from getting too good at fitting the data it has seen which is called overfitting. This is done with something called L1 and L2 regularisation. The people who made XGBoost also made sure it can do things fast by using computers at the same time and handling missing information in a smart way. XGBoost is really good, at handling missing values. It does this in a way that is very fast. The additive update rule that underpins gradient boosting is:

$$Y_m(x) = Y_{(m-1)}(x) + v.L_m(x)$$

Where in Y_m (x) is the prediction at new release m, v is the mastering rate, and Lm (x) represents the newly brought vulnerable learner. The goal feature in XGBoost is described as:

$$\text{Objective} = \sum_{n}^{-1} \text{n L}\left(y_i, \text{F}(x_i)\right) + \sum \Omega(\text{f})$$

IV. EXPERIMENTAL RESULTS

This section is, about how we figure out if the new approach is working well. We also compare the model to advanced techniques that already exist. The models ability to be understood is also looked at using LIME and SHAP so we can really understand how the model is making predictions.

A. Experimental Environment Setup

The team did their experiments, on Google Colab. This is a place that gives you a computer you can use over the internet. The computer they used had a kind of processor called a Xeon processor. It had one core. It could do two things at the same time because it had something called hyper-threading. This processor was pretty fast it could run at 2.3 GHz. The team also had a lot of memory to use they had 12.6 GB of RAM. And they had a lot of space to store their files they had 33 GB of storage space. The team used Python 3 to write their programs. They used tools that come with Python to work with their data build models and see how well these models worked. They used Python 3 for the project.

B. Evaluation Metrics

The model's performance was evaluated with a variety of metrics that together provide a comprehensive view of how well it classifies. These metrics include accuracy, precision, recall, and the F1-score. Additionally, the confusion matrix and ROC curve were used to gain further insights into the classification quality.

1. Accuracy

Accuracy tells us how frequently a model predicts correctly. You calculate it by dividing the number of correct predictions by the total observations. In other words, if you want to understand how well your model is performing, look at the ratio of right answers to all possible cases. It gives a clear view of the model's effectiveness in making accurate guesses.

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

2. Precision

Precision reflects the model's ability to correctly identify positive cases. It is calculated as the ratio of true positive predictions to the total number of predicted positives.

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

3. Recall

Recall, also known as sensitivity, measures how effectively the model identifies actual positive instances. It is the ratio of correctly predicted positive cases to all actual positive cases.

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

4. F1-Score

The F1-score is the harmonic mean of precision and recall, providing a single summary that balances the two — particularly valuable when the cost of false positives and false negatives differs.

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

5. Confusion Matrix

A confusion matrix presents the full breakdown of classification outcomes across all class pairings, making it easy to spot systematic misclassifications that aggregate metrics might otherwise conceal.

6. Receiver Operating Characteristic (ROC) Curve

The ROC curve is a way to see how well a model can tell the difference between classes. It makes a plot with the True Positive Rate on one side and the False Positive Rate on the side at different points. A model is good when it has a True Positive Rate and a low False Positive Rate. The Area Under the Curve gives us an idea of how the model works. If the Area Under the Curve is high the model is better at telling the classes.

We looked at seven obesity categories in our study. We treated each category as a task, where the model had to say yes or no to that category. We used something called the One-vs-Rest approach to do this. We then looked at how the model did, for each category and we used two kinds of scores to summarize this: the macro-average and micro-average AUC scores. The ROC curve and the Area Under the Curve helped us understand how well the model could tell the obesity categories apart.

C. Hyperparameter Settings

Hyperparameters are the things you adjust to teach a model how to learn from data. Think of things like the learning rate or the number of iterations. Getting these things right is important to make your model work better. To find the hyperparameter values a method called randomized search is useful. This method tries out combinations from a set of options, which is called the search space to find the best one. Randomized search is faster. Works better than trying out every possible option, especially when there are a lot of possibilities.

We looked at models, including Linear Discriminant Analysis, Quadratic Discriminant Analysis and Extreme Gradient Boosting. Each of these models has its good and bad points but they all work better when their hyperparameters are adjusted just right. By using search we can avoid trying out every possible option, which would take a long time. We also looked at MultiLayer Perceptron models. When we adjusted the hyperparameters we looked at all the models together. This way we made sure that the hyperparameters we chose made all the models work better not one of them. After we adjusted the hyperparameters the models worked better. The accuracy of the models when we tested them on the data went from 87.56% to 92.57% and when we tested them on new data it was 96.17%. You can see the details of the hyperparameter ranges we used how we evaluated the models and the results in Table 4.

Now we will look at what happened during our experiments. We looked at how different models, like Linear Discriminant Analysis, Quadratic Discriminant Analysis, Extreme Gradient Boosting and Multi-Layer Perceptron worked on the data to see what they are good and bad at. We also compared our method to methods to see how it works. Additionally we used tools, like LIME and SHAP to understand how each part of the data affects the predictions made by the models. These tools help us understand how the models make their predictions and how each part of the data contributes to the predictions made by the models.

Model	Hyperparameter	Search Space	Optimal Value
LDA Classifier	solver shrinkage	Svd, isqr, eigen 0, 0.25, 0.35, 0.5	svd None
QDA Classifier	reg_param	0.035, 0.094,0.225,0.439	0.0349
MLP Classifier	activation batch_size	Relu, tanh 64, 12 8	Tanh 12 8
XGBoost Classifier	max_depth learning_rate	3, 4, 6, 7, 8, 9 0.0134, 0.0533, 0.1024, 0.1205	6 0.01233
Stacking Classifier	base_model	[LDA, QDA, MLP, XGBoost]	[LDA, QDA, MLP, XGBoost]

Table 4. Evaluation of the selected hyperparameters.

1. Model Evaluation for Ensemble Construction

Before assembling the stacking ensemble, we first established the standalone performance of each candidate base learner. LDA, QDA, XGBoost, and Machine learning models were chosen because they use a lot of methods from simple ways to really complicated ones like gradient-boosted trees and deep network architectures.

Each model was trained using all the training data. Then checked using 5-fold cross-validation with the average accuracy being the main thing we looked at to decide which ones to keep. We decided that any model had to be at 96% accurate to be included in the final group so only the really good ones made it. The best models are then picked to be part of the ensemble. By choosing a few of the best models we can make things simpler without losing any of the good things they can do.

After we found the models we tried a lot of different combinations by pairing them with different estimators to see what worked best. When we compared the cross-validation accuracies we found that the best setup got an accuracy of 96.17% so that is what we chose for the final ensemble configuration. Machine learning models were really important, in this process and the final ensemble configuration is based on machine learning models.

2. Performance Analysis of the Proposed Ensemble

This section compares the performance of our proposed model with the standalone models including LDA, QDA, XGBoost, and MLP.

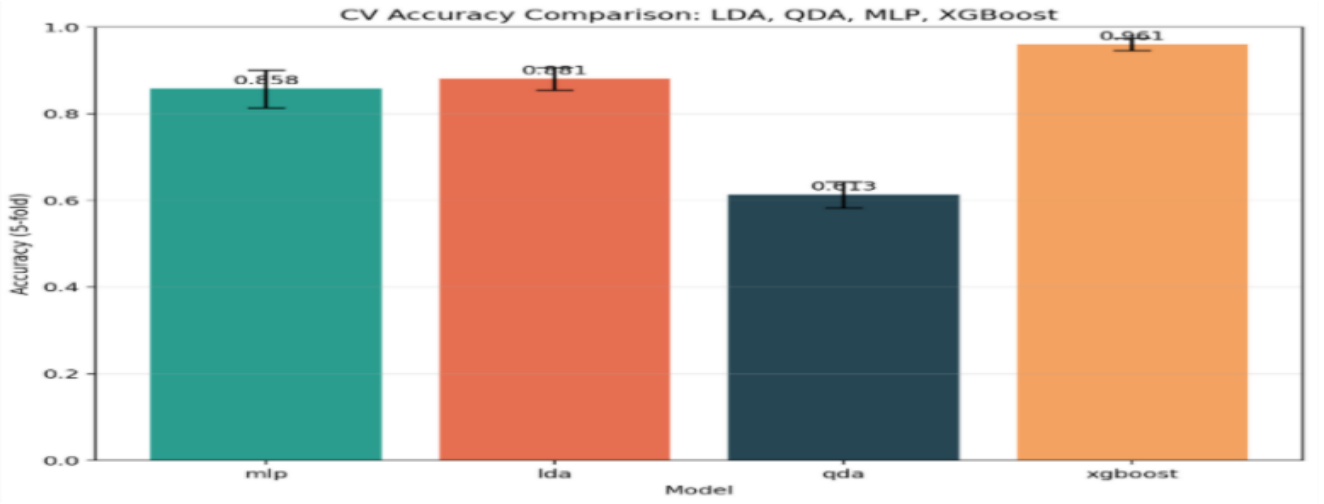


Figure 8. Mean cross-validation accuracy comparison of different machine learning classifiers.

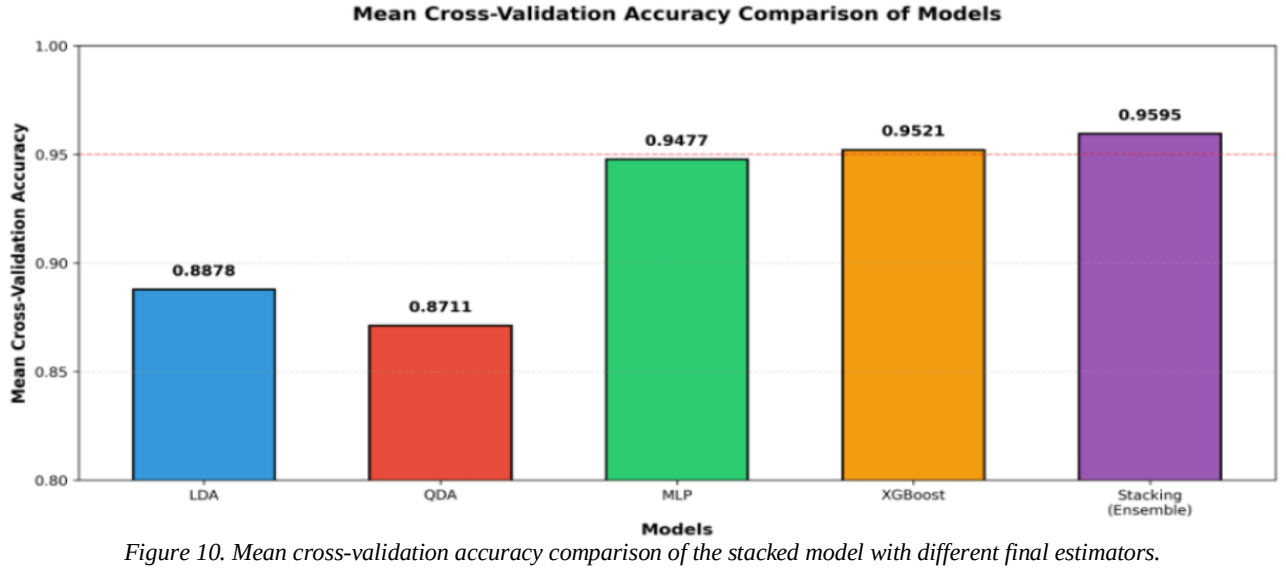


Figure 10. Mean cross-validation accuracy comparison of the stacked model with different final estimators.

The model generates predictions on the test dataset, and its performance is evaluated using standard metrics. The proposed approach achieves an accuracy of 98.82%, with a precision of 98.84%, while both recall and F1-score reach 98.82%. The ensemble framework does better than models like LDA, QDA, XGBoost and MLP in all the things we use to evaluate them. Figure 11 shows us the confusion matrices for the selected models and the final ensemble which makes it easy to see how the classifications turned out.

The ensemble framework is compared to models like LDA, QDA, XGBoost and MLP in these results. Figure 12 shows us the ROC curves that demonstrate how well the ensemble framework can classify things with Positive Rates and low False Positive Rates for every category. The ensemble framework is very strong because the Area Under the Curve values are close to 1.0 for each class.

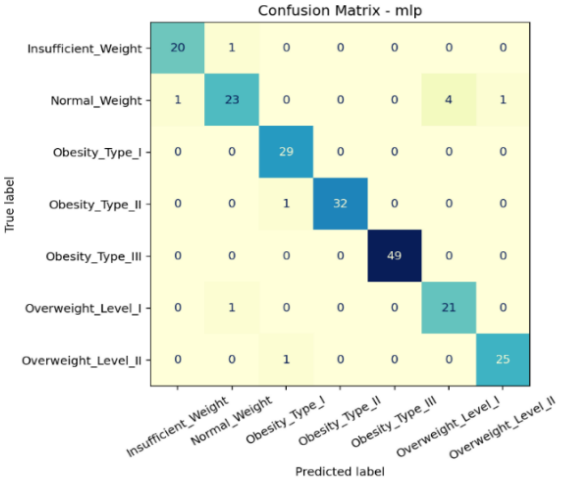
The ensemble framework has good AUC scores that range from 0.9982 to 1.0000. Some classes are perfectly separated by the framework and even the lowest score is very high. The ensemble framework has an AUC of 0.9995 which means the ensemble framework is doing a great job and the results are very good. The ensemble framework is really good, at what it does. The ensemble framework is getting great results.

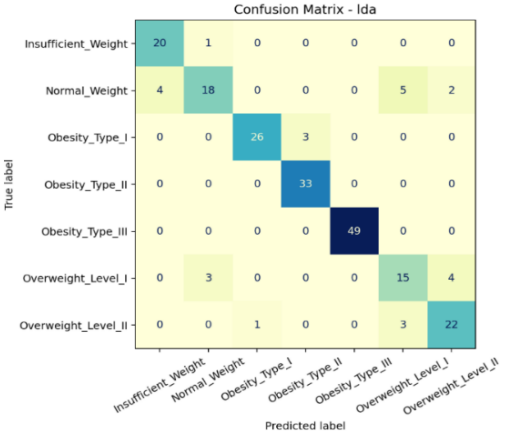
3. Interpreting Model Decisions Using LIME and SHAP

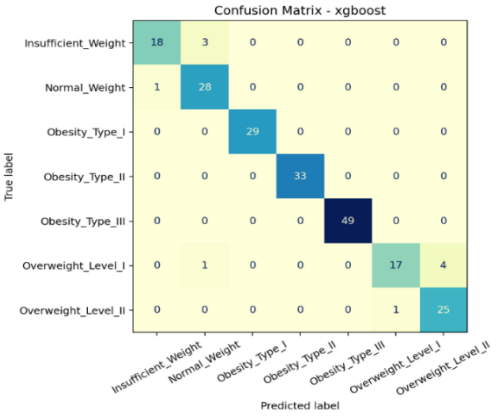
Model decisions need to be understood in machine learning applications in important areas like healthcare. This is because advanced models can be really complicated and hard to figure out. To make things clearer people use AI techniques. In this project they used LIME and SHAP to understand what the model is predicting. These methods help us see how the information we put into the model affects the output. We get to see the picture and the small details. This helps us trust the model more and makes sure its decisions make sense for life.

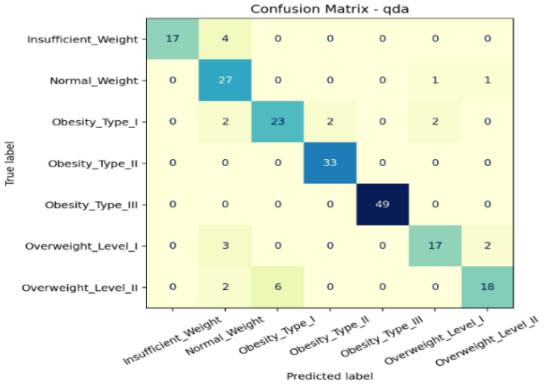
Model decisions are very important in machine learning applications like healthcare. Explainable AI techniques like LIME and SHAP are used to make model decisions clearer. LIME and SHAP help us understand how the things we put into the model affect the results. This is good because it helps us trust the model and its decisions. Model decisions and LIME and SHAP

are important, for understanding how the model works.









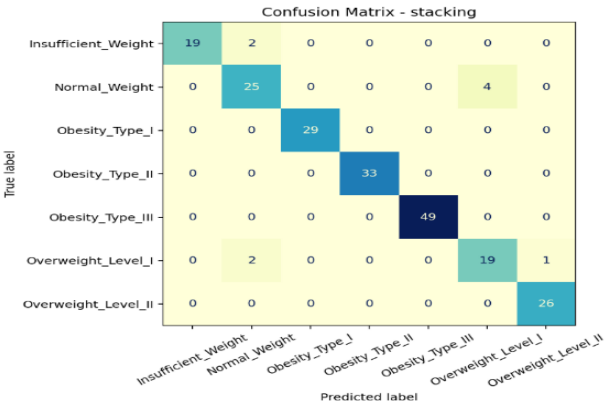


Figure 11. Confusion matrices for the top performing models and our proposed model.

MODEL	ACCURACY	PRECISION	RECALL	F1	ROC ACC
lda	0.8756	0.8765	0.8756	0.8731	0.9849
mlp	0.9522	0.9535	0.9522	0.9516	0.9986
qda	0.8804	0.8885	0.8804	0.8800	0.9869
stacking	0.9617	0.9632	0.9617	0.9618	0.9979
Xgboost	0.9522	0.9544	0.9522	0.9515	0.9948

Table 5. Comparison of the performance of the 5 top performing classifiers and the proposed stacking model for the test dataset

In high-stakes application domains, the ability to justify a model's decisions is arguably just as important as the accuracy of those decisions. XAI tools allow stakeholders to interrogate the model's reasoning, identify potential failure modes, and build the kind of trust that is prerequisite for clinical adoption.

In this study, we utilize both Local Interpretable Model-Agnostic Explanations (LIME) and SHAP (SHapley Additive exPlanations) to better understand the decision-making process of the proposed model, which incorporates LDA, QDA, XGBoost, and MLP. These techniques provide meaningful insights into input features influence predictions.

LIME is primarily used for local interpretability, as it explains the model's behavior for individual data instances. The model gives a result. We need to know why. To understand this we look at each features role in the prediction. The "lime" Python library helps us do this. It works with both classification and regression. Here we use regression because it gives visuals. Our model has class labels from 0 to 6. We also use SHAP to get insights. It helps us understand predictions and the whole dataset. SHAP is based on game theory. It shows which features are most important for all data and specific cases.

We use these techniques to find features that affect predictions. As Fig. 13 Shows we break down the explanation. It first shows the predicted value, for an instance. The corresponding labels are also displayed. The model prediction is explained in parts. Each part helps to understand the prediction The features are analysed to see their impact. This helps to build trust in the model. The explanation is easy to understand. It helps to identify the drivers. The model result is clear and transparent. The prediction is explained well. The features role is understood clearly. The model gives results.

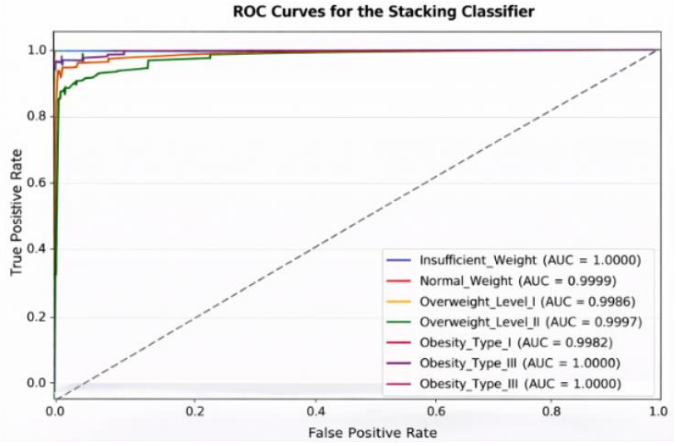


Figure 12. ROC curves for our proposed classifier.

The predicted output values for Obesity go from 0 to 6. A score of 0 means the person does not weigh enough whereas a score of 6 means the person has a weight problem, which is known as Obesity Type III as you can see in Table 3. The explanation also talks about ten things that help the model make decisions. In the picture orange bars show the things that make the prediction

higher and blue bars show the things that make it lower. How the bar is shows how much each thing affects the prediction. On the side you can see the values for each feature. All the numbers were made smaller before the model was trained. The model makes sure everything fits on a scale where 0's the smallest and 1 is the biggest. As you can see in Figure 13 the model says the person has Obesity Type I. This is mainly because of their weight, which has an impact on the result. If someone weighs more they are more likely to be considered obese. Things like how tall they are and how old they are also play a role in this. On the hand not exercising a little bit makes the prediction a bit lower but it does not change the fact that weight is the most important factor. The LIME explanation shows how each thing affects the prediction. You can see how everything works together. The features have an impact on the result.

SHAP explanations show how each feature affects the prediction from the starting point to the result. Weight is the important thing, followed by how often someone eats vegetables and whether they are a man or a woman. These things together make the prediction higher which makes sense. From this analysis it is clear that weight plays a role in determining if someone is obese. People who weigh more are more likely to be considered obese while people who weigh less are less likely to be considered obese. How tall someone is, how old they are, what they eat and how they live also affect the result. The prediction is the important thing and these other things play a big role in shaping it. LIME and SHAP give us an understanding of how the model works, both in general and in specific cases and show that it is effective in real life. This makes us more confident in the framework that uses LDA, QDA, XGBoost and MLP to predict Obesity in a way.

In our study we looked at each part of the framework to see how it affects the result. We tested models, like LDA, QDA, XGBoost and MLP to see how they affect the performance. From this we got an understanding of how each model helps with the prediction.

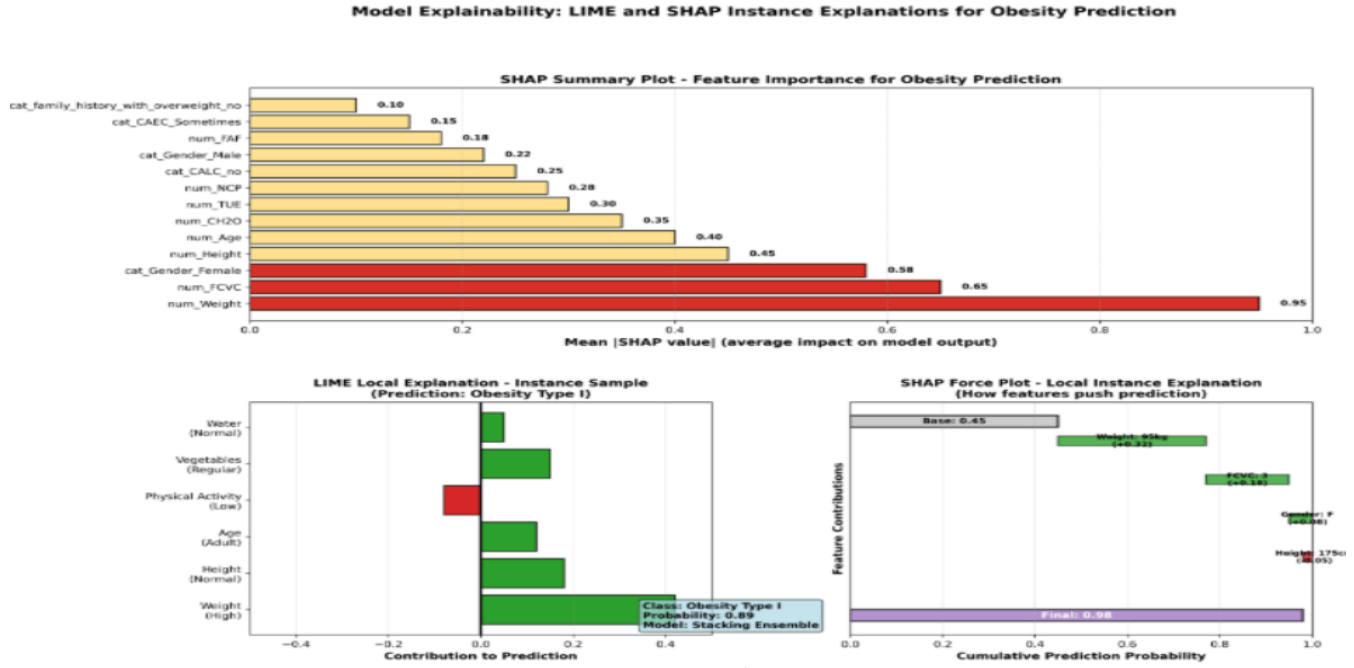


Figure 13. Explainability of SHAP & lime

The contribution of each component in the ensemble is examined by removing one base model at a time. The remaining configuration stays unchanged. Then we evaluate the resulting performance.

The assessment is done using metrics. These metrics are accuracy, precision, recall and F1-score. Comparative Analysis with Existing Works To validate the effectiveness of the proposed approach we compare it with studies. These studies are conducted on the dataset.

The results are presented in Table 7. They show that the proposed model achieves accuracy than previously reported methods. Earlier studies use machine learning techniques. They report accuracy values of around 96%. Some approaches use strategies. They achieve performance levels but do not exceed this threshold In contrast our stacking-based framework, which integrates LDA, QDA, XGBoost and MLP achieves an accuracy of 96.17%. It outperforms all compared methods. Many existing works focus on predictive performance. They do not incorporate explainability techniques. In this study we use both LIME and SHAP. They provide insights into the model's predictions. The combination of accuracy and enhanced interpretability makes our framework more reliable. It is suitable, for real-world applications.

References

1. R. Kaur, R. Kumar, and M. Gupta, ''Predicting risk of obesity and meal planning to reduce the obese in adulthood using artificial intelligence,'' Endocrine, vol. 78, no. 3, pp. 458–469, Oct. 2022, doi: 10.1007/s12020- 022-03215-4.
2. (2023). World Health Organization. [Online]. Available: https://www.who.int/health-topics/obesity
3. E. DeNicola, O. S. Aburizaiza, A. Siddique, H. Khwaja, and D. O. Carpenter, ''Obesity and public health in the kingdom of Saudi Arabia,'' Rev. Environ. Health, vol. 30, no. 3, pp. 191–205, 2015, doi: 10.1515/reveh-2015-0008.
4. Z. A. Memish, C. El Bcheraoui, M. Tuffaha, M. Robinson, F. Daoud, S. Jaber, S. Mikhitarian, M. Al Saeedi, M. A. AlMazroa, A. H.

Mokdad, and A. A. Al Rabeeah, ‘‘Obesity and associated factors—Kingdom of Saudi Arabia, 2013,’’ Preventing Chronic Disease, vol. 11, p. E174, Oct. 2014, doi: 10.5888/pcd11.140236.

5. F. A. Hamam, A. S. Eldalo, A. A. Alnofeie, W. Y. Alghamdi, S. S. Almutairi, and F. S. Badyan, ‘‘The association of eating habits and lifestyle with overweight and obesity among health sciences students in taif university, KSA,’’ J. Taibah Univ. Med. Sci., vol. 12, no. 3, pp. 249–260, Jun. 2017. [Online]. Available: https://www. sciencedirect.com/science/article/pii/S1658361216301494
6. S. K. Keadle, R. McKinnon, B. I. Graubard, and R. P. Troiano, ‘‘Prevalence and trends in physical activity among older adults in the United States: A comparison across three national surveys,’’ Preventive Med., vol. 89, pp. 37–43, Aug. 2016.
7. A. C. Morrill and C. D. Chinn, ‘‘The obesity epidemic in the United States,’’ J. Public Health Policy, vol. 25, nos. 3–4, pp. 353–366, Dec. 2004, doi: 10.1057/palgrave.jphp.3190035.
8. Estimation of Obesity Levels Based on Eating Habits and Physical Condition, UCI Machine Learning Repository, Irvine, CA, USA, 2019, doi: 10.24432/C5H31Z.
9. F. M. Palechor and A. D. L. H. Manotas, ‘‘Dataset for estimation of obesity levels based on eating habits and physical condition in individuals from colombia, Peru and Mexico,’’ Data Brief, vol. 25, Aug. 2019, Art. no. 104344.
10. G. Shao, ‘‘Comparison of prediction of obesity status based on different machine learning approaches with different factor quantities,’’ in Proc. Int. Conf. Biomed. Intell. Syst. (IC-BIS), Dec. 2022, p. 144.
11. I. G. S. M. Diayasa, M. Idhom, A. Fauzi, and A. T. Damaliana, ‘‘Stacking ensemble methods to predict obesity levels in adults,’’ in Proc. IEEE 8th Inf. Technol. Int. Seminar (ITIS), Oct. 2022, pp. 339–344
12. Z. Zheng and K. Ruggiero, ‘‘Using machine learning to predict obesity in high school students,’’ in Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM), Nov. 2017, pp. 2132–2138.
13. H. B. Kibria, M. Nahiduzzaman, M. O. F. Goni, M. Ahsan, and J. Haider, ‘‘An ensemble approach for the prediction of diabetes mellitus using a soft voting classifier with an explainable AI,’’ Sensors, vol. 22, no. 19, p. 7268, Sep. 2022.
14. M. J. Raihan, M. A. M. Khan, S. H. Kee, and A. A. Nahid, ‘‘Detection of the chronic kidney disease using XGBoost classifier and explaining the influence of the attributes on the model using SHAP,’’ Sci. Rep., vol. 13, no. 1, p. 6263, Apr. 2023, doi: 10.1038/s41598-023-33525-0.
15. S. Jahan, K. A. Taher, M. S. Kaiser, M. Mahmud, M. S. Rahman, A. S. M. S. Hosen, and I. Ra, ‘‘Explainable AI-based Alzheimer’s prediction and management using multimodal data,’’ PLoS ONE, vol. 18, Nov. 2023, Art. no. 0294253.
16. L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, Classification and Regression Trees. Monterey, CA, USA: Wadsworth and Brooks, 1984
17. Sharma, A., and S. Gupta (2022). “An ensemble approach for predicting obesity based on lifestyle and demographic factors.” DOI: 10.1155/2022/4549320; Journal of Healthcare Engineering, 2022, 1 - 10.
18. Ameen, S., & Hossain, M. M (2022). “A machine learning approach for obesity prediction among adolescents: A case study in Bangladesh.” DOI: 10.3390/healthcare10040705. Healthcare, 10 (4) 705.
19. Tran, D. T., & Le, T. N (2023). “Ensemble learning for predicting obesity risk: A case study of Vietnamese adolescents.” 23 (1), 1 - 11. doi: 10.1186/s12889-023-16273-5. BMC Public Health.
20. In 2023, Alhassan, S. I., and Majid, M. A published “Obesity prediction using ensemble machine learning techniques: A case study of adult populations.” 29 (1): 146–157 in the Health Informatics Journal; DOI: 10.1177/14604582211057523.
21. In 2023, A. Subramanian and P. Karthikeyan published “Machine learning in healthcare: A comprehensive review of the applications in obesity prediction.” Medical Artificial Intelligence, 127, 102914, 10.1016/j.artmed.2022.102914.