Automated Diet and Exercise Suggestion based on Obesity Classification

E. Annapoorna1*, Palreddy Nithin Sai1, Konda Raj Shreyas Goud1, Kourikanti Koushik1, Manisha Saini2

1Department of CSBS, GRIET, Hyderabad, Telangana State, India
2Uttaranchal Institute of Technology, Uttaranchal University, Dehradun, 248007, India

Abstract. Obesity is a problem that affects the general health and wellness of many people around the world. The paper seeks to provide an automated individualised dietary and exercise suggestions based on an individual's obesity class. The system classifies the user's level of obesity using Artificial Neural Networks (ANN), Support Vector Machines (SVM), and random forest machine learning algorithms. To provide appropriate food and activity procedures in accordance with the classification of obesity, we use machine learning techniques like decision trees and recommendation algorithms.

1 Introduction

Obesity is a medical disorder characterised by excessive fat accumulation to the point where it negatively impacts an individual's health. Body Mass Index (BMI) is a typical definition, which is computed by dividing a person's weight in kilogrammes by their height in metres squared. Obesity is defined as a BMI of 30 or more, whereas overweight is defined as a BMI of 25 to 29.9. Obesity is linked to a variety of health issues, including type 2 diabetes, heart disease, stroke, high blood pressure, and certain types of cancer, esteem, despair, and social isolation. Obesity treatment usually includes making lifestyle changes, such as increasing physical activity and eating a nutritious diet. The long-lasting solution for fit body is to maintain proper diet and follow an exercise routine. So a machine learning algorithm which recommends automated diet and exercise based on each individual's need and body type to transform the user and help maintaining a fit and healthy life is vital and helps all its users. In this paper, we are using a decision tree classification algorithm.

1.1 Working of decision tree algorithm

A machine learning approach called decision tree classification is utilised for both regression and classification tasks. It is a supervised learning method that constructs a tree-like model of decisions and their potential outcomes. The tree is built by iteratively
separating the data on different attributes, with the purpose of maximising class separation or minimising impurity within each class.

1.2 Methodology

Our system is a web-based application that provides personalized diet and exercise recommendations to users. The user interacts with the system through a web interface, where they input information about their diet and exercise preferences, goals, and other relevant data. It receives the user's inputs and forwards them to the appropriate components for further processing.

This step includes tasks like data normalization, handling missing values, and feature selection. The preprocessed data is then used to build a classification model. The model is trained on existing data and algorithms to predict appropriate diet and exercise recommendations based on the user's inputs. The built model is tested and evaluated using validation techniques to ensure its accuracy and reliability. Based on the classification results, personalized diet and exercise recommendations are generated. Once the recommendations are generated, the system generates a response that contains the personalized diet and exercise recommendations. This response is sent back to the web server. Finally, the web server presents the diet and exercise recommendations to the user through the web interface.

2 Existing methods

2.1 Overview

Existing approaches in obesity classification typically involve assessing an individual's body composition, primarily using body mass index (BMI), and considering additional factors such as waist circumference, waist-to-hip ratio, body composition analysis, obesity-related comorbidities and overall health status. Table 1 is a summary of common obesity classification existing approaches.

Authors [14] explored the distinct ML applications in predicting heart attacks using patient health records. It compares Random Forest and CNN methods, and findings showed that Random Forest’s better performance in terms of accuracy. Authors [15] highlighted the significance of ML in prediction, pattern recognition and error reduction across diverse fields, emphasizing the impact of AI in broad domain.

Authors [16] suggested data mining techniques to predict disease-prevalence based on symptoms in healthcare data. The appropriate prediction helps healthcare organizations avoid drug shortages and further ensures timely treatment of patients.

Authors [17] discussed the role of Intelligent Decision Support Systems (IDSS) in Healthcare Monitoring, especially for heart disease. Results claimed that IDSS enhances decision-making functionalities in uncertain healthcare scenarios, thereby significantly improving the monitoring and remedial activities.

Authors [18] employed Independent Component Analysis (ICA) and HDFS-based algorithm to diminish the dimensionality in huge patient datasets, followed by removing irrelevant data and reducing storage and computation time, with superior accuracy compared to Principal Component Analysis (PCA).
Table 1. Summary of few existing approaches.

<table>
<thead>
<tr>
<th>Ref. No.</th>
<th>Description</th>
<th>Dataset</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Predicting Obesity in Adults using machine learning techniques</td>
<td>The RISKESDAS undertaken by the Indonesian Ministry of Health survey.</td>
<td>Logistic Regression gave best accuracy i.e. 72%</td>
</tr>
<tr>
<td>[2]</td>
<td>Classification algorithms for Prediction of obesity levels</td>
<td>The data is used from UK’s Millennium Cohort Study (MCS)</td>
<td>J48 gave the best accuracy of 83.06%</td>
</tr>
<tr>
<td>[3]</td>
<td>Machine learning techniques to predict overweight</td>
<td>The study had data about dietary practises and physical health</td>
<td>Random Forest gave the best accuracy of 78%</td>
</tr>
<tr>
<td>[4]</td>
<td>Classification method for obesity level prediction</td>
<td>Dataset based on a survey for college students</td>
<td>LMT gave the highest accuracy of 96%</td>
</tr>
<tr>
<td>[5]</td>
<td>Obesity prediction using machine learning techniques</td>
<td>Here dataset used is public clinical available dataset</td>
<td>The gradient boost classifier model has results, with accuracy of 90%</td>
</tr>
<tr>
<td>[6]</td>
<td>Predicting Overweight and Obesity Status Among Malaysian Working Adults with Machine Learning.</td>
<td>This is a retrospective study of predictive model derivation using data from the Malaysia’s Healthiest Workplace by AIA Vitality 2019 survey</td>
<td>Model has shown interesting results as the Gradient boosting algorithm achieves the best accuracy of 73%</td>
</tr>
<tr>
<td>[7]</td>
<td>Comparison Of Different Machine Learning Methods Applied to Obesity Classification</td>
<td>A publicly available dataset</td>
<td>ANN and Decision tree’s accuracy of prediction is higher than 90%</td>
</tr>
<tr>
<td>[8]</td>
<td>Obesity Level Classification Based on Decision Tree and Naïve Bayes Classifiers</td>
<td>A recent, standard, and the publicly available dataset is used to validate the model</td>
<td>Decision Tree has highest accuracy percentage which is 96%</td>
</tr>
<tr>
<td>[9]</td>
<td>Diet &amp; Exercise Classification using Machine Learning</td>
<td>A publicly available dataset</td>
<td>Decision tree provides the best accuracy 71.54%</td>
</tr>
<tr>
<td>[10]</td>
<td>Predicting the risk of obesity and meal planning to reduce obesity in adulthood</td>
<td>A dataset with characteristics of one's physical description and eating habits from UCI ML repository</td>
<td>Gradient Boosting classifier achieved the highest accuracy of 95%</td>
</tr>
<tr>
<td>[11]</td>
<td>Obesity level estimation software based on decision tree classifier</td>
<td>An open source Dataset</td>
<td>J48 has the best precision rate and accuracy of 95%</td>
</tr>
<tr>
<td>[12]</td>
<td>Using Machine Learning Method for classification Body Mass Index(BMI) for clinical decision</td>
<td>Dataset used has people who selected randomly from all area of a city.</td>
<td>The DT, LR and MLP algorithms showed maximum precision</td>
</tr>
<tr>
<td>[13]</td>
<td>Machine Learning Based Adult Obesity Prediction</td>
<td>Datasets were gathered from the UCI Machine Learning Repository</td>
<td>Logistic Regression model has the best prediction accuracy in the models generated</td>
</tr>
</tbody>
</table>
3 Proposed method

3.1 Problem statement

It is crucial to highlight that healthy eating habits are beneficial to both physical and mental health. A diet rich in fruits, vegetables, whole grains, and lean meats has been demonstrated in studies to help reduce the risk of depression and anxiety. Exercise can help lower your risk of chronic diseases like heart disease and diabetes, as well as enhance your mood and cognitive performance.

Ultimately, the key to adopting a healthy lifestyle is to make small, sustainable changes over time. By gradually incorporating healthy habits into your routine, such as eating more fruits and vegetables, and taking short walks throughout the day, you can make significant improvements in your health and well-being over time.

3.2 Objectives of paper

- To take control of their health and improve overall well-being.
- To classify obesity and provide automated diet recommendations.
- To empower individuals to take control of their health and improve overall well-being.

3.3 Connectivity diagram

Fig. 1. Connectivity diagram of the proposed work.

1. Data Collection Module: This module is related to collecting data on the individual's personal information, such as age, sex, weight, height, and body composition measurements, as well as their dietary preferences and exercise habits.
2. Data Pre-processing Module: This module is responsible for cleaning and transforming the collected data into a format suitable for analysis. It involves tasks such as data normalization, missing value imputation, and feature selection.
3. Obesity Classification Module: This module is responsible for classifying the individual's level of obesity based on their personal information and body composition
measurements. It can use techniques such as BMI or waist circumference measurement to classify the individual's obesity level.

4. Diet Recommendation Module: Based on a person's caloric needs, macronutrient requirements, and food preferences, this module creates automated dietary suggestions. It can optimise the diet plan using algorithms like linear programming.

5. Exercise Recommendation Module: This module is responsible for generating personalized exercise plan based on the individual's obesity level, exercise choices, and time constraints. It can use techniques such as decision trees to recommend appropriate exercises.

6. Results Visualization Module: This module is responsible for presenting the automated diet and exercise recommendations in an easily understandable format, such as a dashboard or a mobile app. It can use visualization tools such as charts or graphs to display the recommendations.

7. Deployment Module: This module is responsible for deploying the final model into a production environment, such as a web application or mobile app. It involves selecting appropriate deployment tools, testing and debugging the application, and ensuring that the solution is scalable and reliable.

8. Maintenance Module: This module is responsible for maintaining and updating the deployed solution over time. It involves monitoring the performance of the application, identifying and fixing bugs, and updating the solution to incorporate new data or features.

3.4 Architecture of the proposed work

The system described in the architecture diagram is a web-based application for providing personalized diet and exercise recommendations. Users interact with the system through a web interface, providing inputs regarding their diet and exercise preferences, goals, and other relevant information. The inputs are received by a web server, which forwards them to the appropriate components for further processing. The inputs undergo data preprocessing to clean and transform the data into a suitable format. A classification model is then built using the preprocessed data, trained on existing data and algorithms to predict appropriate recommendations. The model is tested to ensure its accuracy and reliability.
4 Results and discussions

4.1 Description about dataset

The Obesity Dataset - raw-and-data-synthetic is a large-scale dataset used to work on papers and studies related to obesity. This dataset is related to obesity, and it contains both raw and synthetic data. It likely includes information about various factors that contribute to obesity, such as diet, exercise, lifestyle, and medical history.

![Fig. 3. Dataset.](image)

The class variable NO obesity was then constructed with the values of Insufficient Weight, Normal Weight, Overweight Level I, Overweight Level II, Obesity Type I, Obesity Type II, and Obesity Type III. This was done after all the data had been labelled.

4.2 Experimental results

4.2.1 Relation between attributes

![Fig. 4. Relation between attributes.](image)

The above fig shows the distribution of weight, age and height. It also depicts how weight and age are correlated and how weight and height are correlated. Now the above data frame is modified into a newer data frame which has less categorical values and more numerical values. We convert the columns by using OneHotEncoder and assign different numerical values based on the values in the dataset.
4.2.2 Encoding the dataframe

After Encoding the data frame upon which we are training appears as in the fig that follows

1. Importing Required Libraries: It is assumed that the preprocessing module from the sklearn package has been imported earlier in the code.
2. Initializing the OneHotEncoder: The code creates an instance of the OneHotEncoder class and assigns it to the variable enc. This encoder is used to convert categorical variables into one-hot encoded binary vectors.
3. Fitting the Encoder: The fit method is called on the enc object, with df_bool passed as the argument. This step analyzes the input data and determines the categories present in each column to prepare the encoder for transformation.
4. Transforming the DataFrame: The transform method is applied to the df_bool DataFrame using the fitted encoder (enc). It converts the categorical variables in df_bool into one-hot encoded binary vectors.
5. Converting to NumPy Array: The toarray method is used to convert the resulting sparse matrix from the previous step (enc.transform(df_bool)) into a dense NumPy array. The transformed data is assigned to a new DataFrame called df_xOHE.

Fig. 5. Encoded dataframe.

4.2.3 Training and test data

1. Importing Required Libraries: The code begins by importing the train_test_split function from the sklearn.model_selection module.
2. Splitting the Data: The train_test_split function is then used to split the data into training and testing subsets. The input data is divided into four separate variables: df_x_train, df_x_test, df_y_train, and df_y_test.
3. df_x and df_y are the input datasets that are being split.
4.2.4 Decision tree classifier

1. Importing Required Libraries: It is assumed that the necessary libraries, such as Decision Tree Classifier from sklearn.tree, pd from pandas, and accuracy_score from sklearn.metrics, have been imported earlier in the code.

2. Initializing and Training the Classifier: The code creates an instance of the Decision Tree Classifier class and assigns it to the variable tree_clf. The classifier is then trained on the training data (dfx_train and dfy_train) using the fit method.

3. Printing Classifier Configuration: The code prints information about the trained decision tree classifier. This output provides details such as the hyperparameters and default configuration of the classifier.

4. Making Predictions on Test Data: The code uses the trained classifier (tree_clf) to make predictions on the test data (dfx_test). The predicted labels are stored in a DataFrame called Test_predict.

5. Calculating Accuracy: The code calculates the accuracy of the classifier's predictions by comparing them to the true labels (dfy_test) using the accuracy_score function from the sklearn.metrics module. The accuracy score is printed as "Accuracy: <accuracy_value>".

Here, Using DecisionTreeClassifier we obtained an accuracy of 93% for the given classification model.

4.2.5 Original data vs. predicted data

1. Importing Required Libraries: It is assumed that the necessary libraries, such as matplotlib.pyplot (usually imported as plt) and pd from pandas, have been imported earlier in the code. Setting Figure Size and Subplot Configuration: The code uses plt.figure(figsize=(15, 3)) to create a figure with a width of 15 inches and a height of 3 inches. The plt.subplot(131) call specifies that the figure will have three subplots, and the first subplot will be active. Plotting Predicted Label Distribution: The code calls Test_predict.value_counts().plot.bar(color='Orange') to generate a bar chart showing the count of each predicted label. The bars are colored orange.

2. Plotting Original Test Label Distribution: The code calls dfy_test.value_counts().plot.bar() to generate a bar chart displaying the count of each original test label. The default colors are used for the bars. The axes labels and title of the subplot are set using plt.xlabel, plt.ylabel, and plt.title.

3. Setting Up the Second Subplot: The code uses plt.subplot(132) to activate the second subplot.

4. Calculating and Printing Value Counts: The code calculates the value counts of the predicted labels (Test_predict) and the original test labels (dfy_test) and assigns them to variables Val1 and Val2, respectively.

5. Printing Prediction Results: The code prints the value counts of the predicted labels with the header "Prediction Results" using print(Val1).

6. Printing Original Test Data: The code prints the value counts of the original test labels with the header "Original Test Data" using print(Val2).

To summarize, this code snippet performs one-hot encoding using the OneHotEncoder from the sklearn.preprocessing module. It fits the encoder on the df_bool DataFrame, transforms the data into a one-hot encoded format, converts it to a NumPy array, stores it in df_xOHE, and then displays the shape and contents of df_xOHE.
4.2.6. User Interface

Finally, after the model is trained on the dataset and obtained good accuracy, we take inputs from the users regarding their eating habits, personal lifestyle habits and some basic information such as age, weight and height. After filling all the required data fields the model takes the inputs and runs on the trained model and classifies the instance into one of the existing classes. After all the fields are filled your personalised diet and exercise plan is ready for you in an instance as follows.

![Graph showing prediction results vs. original results.](image)

**Fig. 6.** Predict results vs. original results.

![User interface.](image)

**Fig. 7.** User interface.
4.2.7 Exercise plan

Here we have different types of exercises and their examples.

**Exercise Plan:**

1. Cardiovascular exercises: Brisk walking, jogging, running, swimming, cycling, dancing, and playing sports like tennis or basketball. Aim for at least 150 minutes of moderate-intensity aerobic activity or 75 minutes of vigorous-intensity aerobic activity per week.
2. Strength training
   - resistance band exercises, bodyweight exercises like push-ups, squats, lunges, and planks, or using weight machines at the gym. Aim to do strength training exercises at least two days a week.
3. Flexibility and stretching exercises: Static stretching or dynamic stretching, or participate in activities like yoga or Pilates that emphasize flexibility and body awareness.
4. High-intensity interval training (HIIT): HIIT involves alternating between short bursts of intense exercise and periods of active recovery. It includes exercises such as jumping jacks, burpees, mountain climbers, or high-knee running.
5. Mind-body exercises: Practices like yoga, tai chi, or meditation can improve flexibility, balance, strength, and mental clarity.
6. Active hobbies and recreational activities: Hiking, dancing, swimming, cycling, gardening, or playing a sport. These activities not only provide exercise but also offer enjoyment and help you maintain an active lifestyle.
5 Conclusion and future enhancements

The use of machine learning algorithms in diet and exercise recommendation systems is a successful strategy for predicting weight loss in obese people. Automated diet and exercise plan will improve the individual’s health. The food and exercise data were most accurately represented by the Decision Tree algorithm, which can be used by medical professionals to effectively counsel and prescribe for obese patients. Integration with Fitness Trackers - We can integrate the system with popular fitness trackers or wearable devices to collect real-time data such as heart rate, steps taken, and calories burned. Goal Tracking and Progress Visualization - Provide users with tools to set goals, track their progress, and visualize their achievements. This can include visual charts, graphs, or dashboards that display weight loss/gain, calorie intake, exercise duration, and other relevant metrics. Mobile Application - Develop a dedicated mobile application for easy access to the system's features on smartphones and tablets. This will enhance user convenience and enable them to receive automated recommendations, track progress, and access resources on-the-go.

References

7. Z. He, Comparison of Different Machine Learning Methods applied to Obesity Classification, in Proceedings of the 2022 International Conference on Machine Learning and Intelligent Systems Engineering (MLISE), Guangzhou, China (2022)
10. R. Kaur, R. Kumar, M. Gupta, Endocrine 76, 4 (2022)
12. F. Amani, A. Mohammadnia, P. Amani, A. Asl, M. Bahadoram, Teoflo Redondo 8 (2022)
14. R. P. Ram Kumar, S. Polepaka, Performance Comparison of Random Forest Classifier and Convolution Neural Network in Predicting Heart Diseases, in Proceedings of the Third International Conference on Computational Intelligence and Informatics, (eds) K. Raju, A. Govardhan, B. Rani, R. Sridevi, M. Murty, Advances in Intelligent Systems and Computing, 1090. Springer, Singapore (2020)