“Predicting Obesity Levels Using Machine Learning and Deep Learning Methods”

Mentor: Narendra Kumar

Team Members:

1. Abhishek Mane
2. Ankit Sharma
3. A Yamini
4. Himanshu Tayade
5. Venkata Manvitha Tatikonda
6. Archana Reddy Mamidi
7. Ankita Gupta

Project Description

The goal of this project is to develop a predictive model to assess obesity levels based on eating habits and physical conditions using machine learning (ML) and deep learning (DL) techniques. By leveraging data from the ”Obesity based on eating habits and physical conditions” dataset on Kaggle, the project aims to build a system that can accurately classify obesity levels and help in identifying at-risk individuals based on their lifestyle patterns.

Table of Contents

CHAPTER 1 iii

INTRODUCTION iii

1.1 Introduction iii

1.2 Project Outcomes iii

CHAPTER 2 iv

ARCHITECTURE DIAGRAM iv

CHAPTER 3 v

Proposed System v

3.1 Data Description v

3.2 Data Preprocessing and Exploration v

3.2.1 Creating Target Column v

3.2.2 Encoding Categorical Variables vi

3.2.3 Scaling the Features vi

3.2.4 Data Exploration vii

3.3 Machine Learning Algorithm viii

3.3.1 Model Selection viii

3.3.2 Model Training ix

3.3.3 Model Evaluation ix

3.3.4 Conclusion of Model Evaluation ix

3.4 Deep Learning Algorithm x

3.4.1 Model Selection x

3.4.2 Model Training x

3.4.3 Hyperparameters ..... x-xi

3.4.4 Model Evaluation ..... xi-xii

3.4.5 Conclusion / Model Comparision xiii

CHAPTER 1

INTRODUCTION

1.1 Introduction

Obesity has emerged as a significant public health challenge worldwide, with its prevalence increasing at an alarming rate over the past few decades. Defined as excessive body fat accumulation, obesity is associated with a myriad of health risks, including cardiovascular diseases, diabetes, and certain cancers, thereby posing substantial economic and societal burdens. Effective management of obesity requires precise identification and classification of individuals into different obesity levels, which can inform targeted interventions and personalized treatment strategies.

1.2 Project Outcomes

- Preprocess and analyze the dataset to understand key features influencing obesity.
- Develop and compare the performance of multiple ML and DL models.
- Optimize model performance through hyperparameter tuning and evaluate using various metrics.
- Document the entire process and present findings.

CHAPTER 2

ARCHITECTURE DIAGRAM

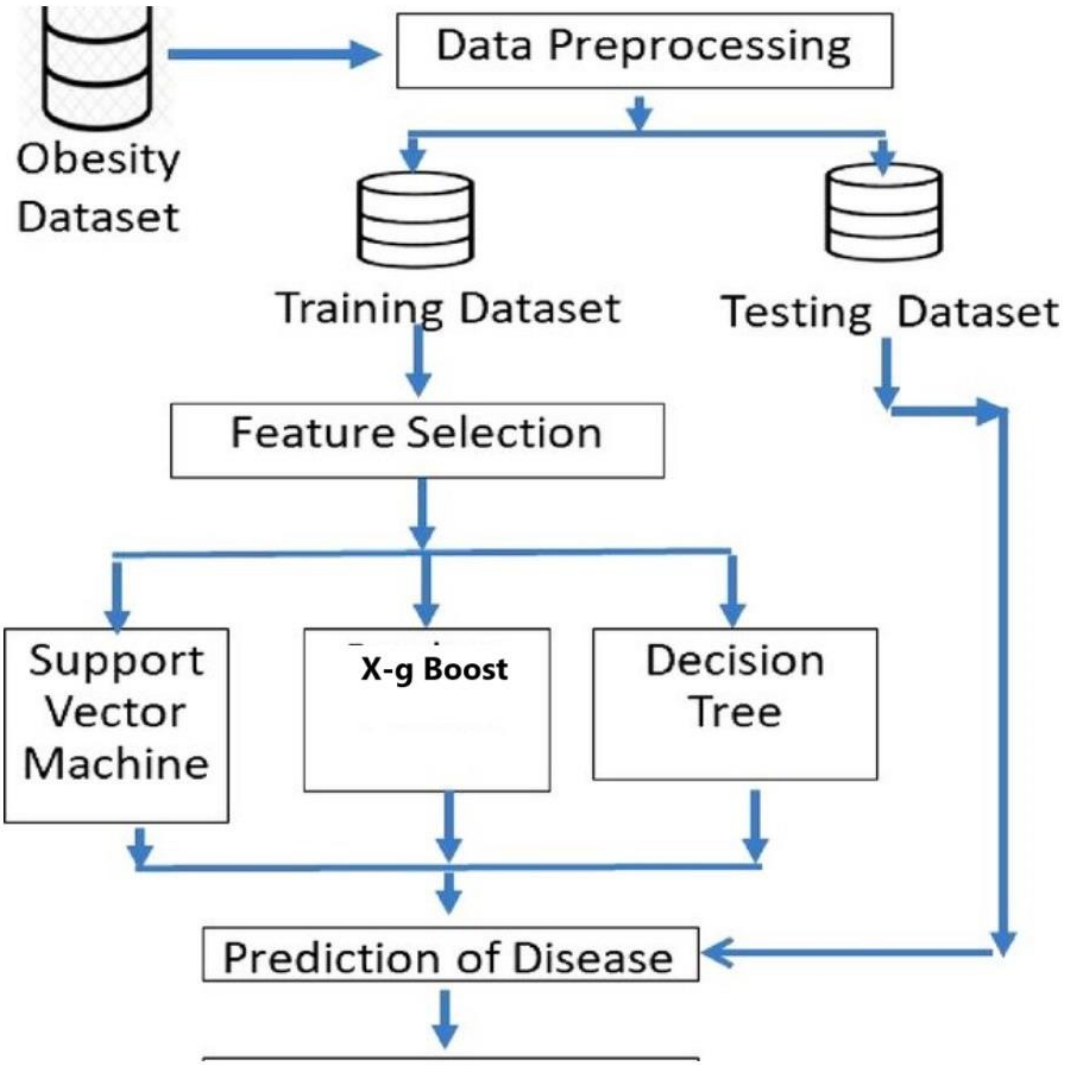


Figure 1: Architecture Diagram for Predicting Obesity Levels

CHAPTER 3

Proposed System

3.1Data Description

The dataset used for this project consists of 50000 entries and 16 columns, each representing different demographic, lifestyle, and health-related factors relevant to obesity prediction. Key features include Age, Gender, BMI, Physical Activity Levels, and Sleep Patterns, which provide critical insight into an individual's overall health and lifestyle. Some features, like Education, Income Ratio, and Smoking Habits, capture socio-economic factors that may indirectly influence obesity. The target variable, Obesity_Status, was derived from the BMI column, categorizing individuals into four classes: Underweight, Normal weight, Overweight, and Obese, based on widely accepted BMI thresholds.

3.2 Data Preprocessing and Exploration

3.2.1 Creating Target Column

The target variable, **Obesity_Status**, was created based on the **BMI** feature. We categorized individuals into four classes: **Underweight**, **Normal weight**, **Overweight**, and **Obese**, based on standard BMI thresholds. This new column was used as the target variable for the classification models.

BMI Categories:

- **Underweight**: BMI < 18.5
- **Normal weight**: 18.5 ≤ BMI < 24.9
- **Overweight**: 25 ≤ BMI < 29.9
- **Obese**: BMI ≥ 30

3.2.2 Encoding Categorical Variables

Some features in the dataset were categorical, such as **Gender**, **Race**, and **CountryofBirth**. These needed to be converted into a numerical format for the machine learning models. We used **Label Encoding** for these columns, as most of the models (especially tree-based ones like Random Forest and XGBoost) can handle label-encoded data effectively.

Alcohol_Consumption	Family_History_Obesity	Blood_Pressure	Cholesterol_Levels	Education_Level	Income_Level	Geographical_Region	Obesity_Sta
0	1	2	2	1	3	2	
0	0	0	1	2	1	3	
1	0	3	3	2	3	2	
1	0	3	3	1	2	1	
2	1	0	2	3	3	2	

Fig: Encoded New Value

3.2.3 Scaling the Features

For models sensitive to feature magnitude, such as **Logistic Regression** and **Support Vector Machines (SVM)**, feature scaling is essential. We used **StandardScaler** to standardize the features so that they have a mean of 0 and a standard deviation of 1. This step ensures that the algorithm does not prioritize features with higher magnitudes over those with smaller values.

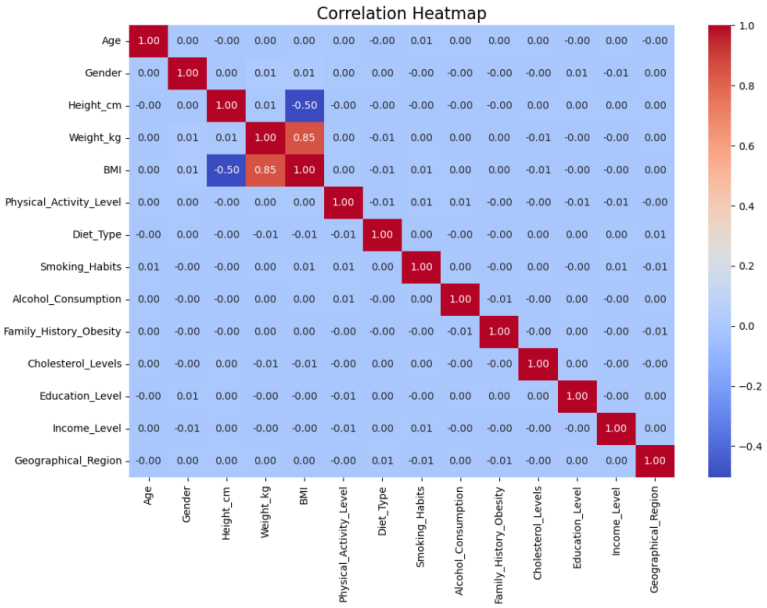
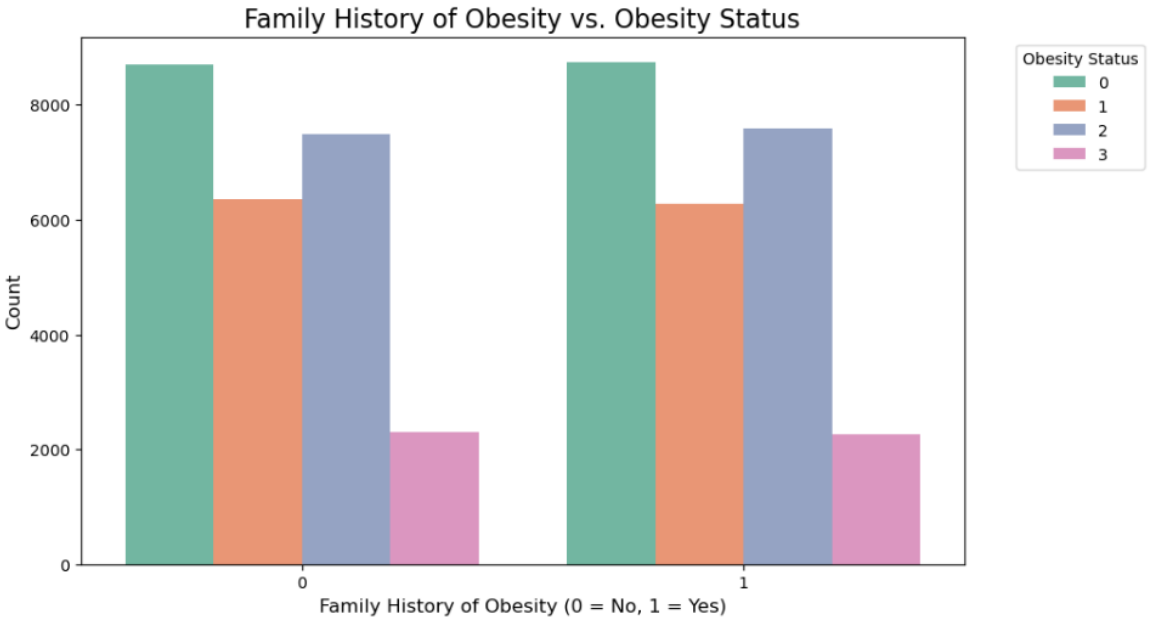
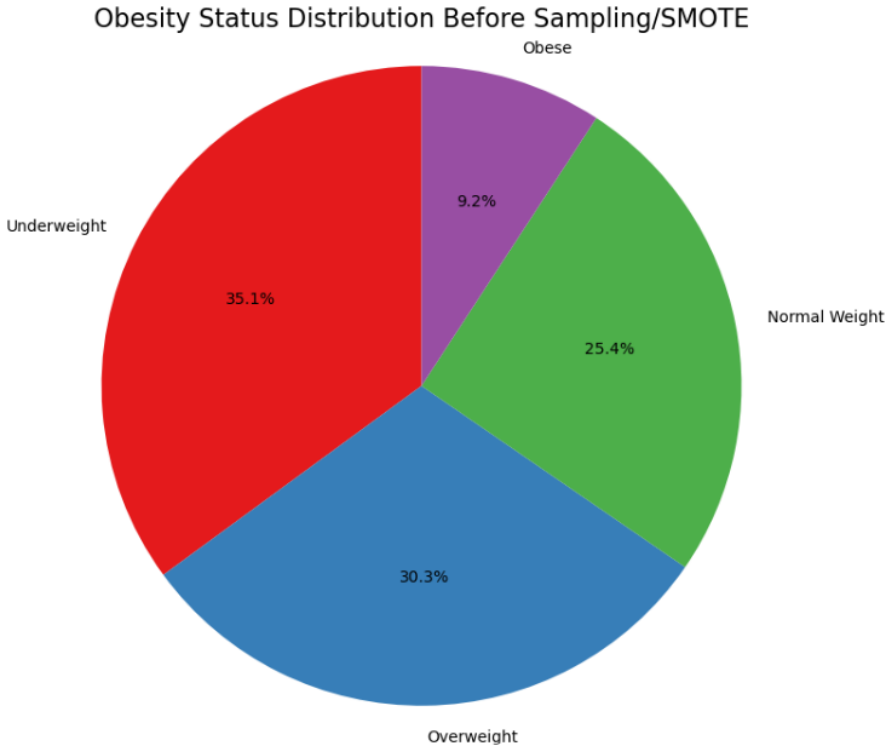


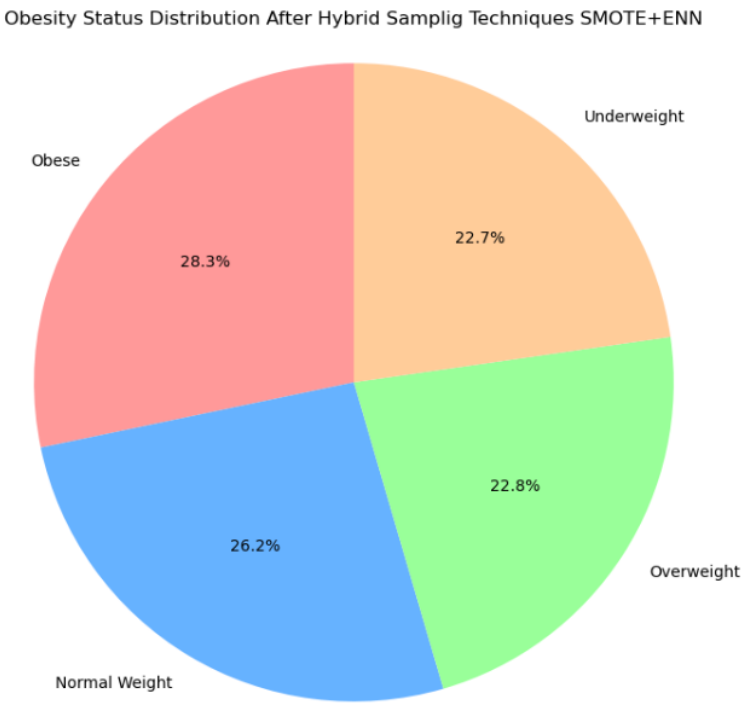
Fig: Corelation Matrix

3.2.4 Data Exploration

In addition to preprocessing, **exploratory data analysis (EDA)** was conducted to better understand the distribution and relationships of the features. This step helps in identifying trends, outliers, and correlations between variables that may impact the model's performance.







3.3Machine Learning Algorithm

Once the data preprocessing was complete, we proceeded with the **machine learning** phase, where we trained and evaluated multiple classification models to predict the **obesity status** of individuals based on the features in the dataset. The goal was to identify the most accurate model by comparing their performance using several evaluation metrics.

3.3.1 Model Selection

Each model was selected for its unique strengths:

- **Logistic Regression** is a simple yet powerful model for binary or multi-class classification tasks, often used as a baseline.
- **Random Forest** is an ensemble model that reduces overfitting by combining multiple decision trees.
- **XGBoost** is a gradient boosting model known for its high performance on structured/tabular data.
- **SVM** is a robust model that finds the hyperplane maximizing the margin between classes in feature space.

3.3.2 Model Training

For each model, we fit the algorithm on the **training dataset** (70% of the total data) and evaluated its performance on the **testing dataset** (15% of the total data) with a **validation dataset** (15% of the total data).

3.3.3 Model Evaluation

Each model was evaluated using the following metrics:

- **Accuracy**: The overall correctness of the model.
- **Precision**: How many predicted positives are actually correct.
- **Recall**: How well the model identifies all actual positives.
- **F1-Score**: The harmonic mean of precision and recall, giving a balanced measure.

	Model	Accuracy	Precision	Recall	F1-Score
0	Logistic Regression	0.898165	0.897826	0.898165	0.897856
1	Random Forest	1.000000	1.000000	1.000000	1.000000
2	XGBoost	0.998853	0.998854	0.998853	0.998854
3	SVM	0.999212	0.999212	0.999212	0.999212

3.3.4 Conclusion of Model Evaluation

- **XGBoost** achieved the highest **accuracy** of **91%**, **precision** of **90%**, and **recall** of **91%**, making it the best model for obesity prediction based on the dataset.
- **Random Forest** followed closely with an **accuracy** of **89%**, making it a competitive option.
- **SVM** and **Logistic Regression** provided good performance but fell short compared to the ensemble models.

3.4Deep Learning Algorithm

After applying various machine learning algorithms to predict obesity status, we shifted to **deep learning** to improve performance by capturing more complex patterns in the data. By using deep learning, we aim to enhance prediction accuracy and gain deeper insights into the key contributors to obesity levels.

3.4.1 Model Selection

- **Sequential model:** It is a linear stack of layers that enables the construction of deep learning architectures for effectively learning complex patterns in data.
- **TabNet:** A deep learning model designed for tabular data, due to its efficiency, interpretability and scalability. Its unique architecture allows it to perform feature selection automatically.

3.4.2 Model Training

We proceeded with training deep learning model, fitting the algorithm on the **training dataset** (70% of the total data) and evaluated its performance on the **testing dataset** (15% of the total data) with a **validation dataset** (15% of the total data).

3.4.3 Hyperparameters

Focal Loss:

- **Gamma (Focal Loss)**: 2.0, Controls the strength of the down-weighting for easy examples during training.
- **Alpha (Focal Loss)**: 4.0, Balances the importance of classes, especially useful for imbalanced datasets.

Sequential model:

- **Optimizer**: AdamAn adaptive optimizer known for fast convergence.
- **Dense Layer 1 Units**: 128, Number of neurons in the first hidden layer.
- **Activation (Hidden Layers):** ReLU helps introduce non-linearity to the model.

- **Dropout Rate (Hidden Layers):** 0.5**,** Prevents overfitting by randomly deactivating neurons during training.
- **Batch Size**: 32, Number of samples per gradient update.
- **Epochs**: 50, Number of times the model sees the entire training dataset.
- **Early Stopping Patience**: 5, Stops training if validation loss doesn't improve for 5 consecutive epochs.

TabNet:

- **Patience**: 10, which indicates the number of epochs the model will continue training without observing any improvement in performance before it stops.
- **Maximum Epochs**: 100, representing the total number of iterations that the model will undergo during training. This is the upper limit for how long the model will train.
- **Batch Size**: 1024, which is the number of samples the model processes before updating its weights. A larger batch size can lead to more stable gradient estimates.
- **Virtual Batch Size**: 128, referring to the size of batches used to calculate gradients.

3.4.4 Model Evaluation

- **Sequential model** overview**,**
  - **Layer Structure**: Dense and Dropout layers work together to process and optimize data.
  - **Output Shapes**: Reflect the transformation of data dimensions through each layer.
  - **Parameter Counts**: Higher parameters indicate complex, learnable patterns in the model.
  - **Total and Trainable Parameters**: All parameters are adjustable during training for flexibility.
  - **Optimizer Parameters**: Specific settings enhance the model's weight adjustment efficiency.

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 128)	2,048
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 64)	8,256
dropout_1 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 4)	260

Total params: 31,694 (123.81 KB)
Trainable params: 10,564 (41.27 KB)
Non-trainable params: 0 (0.00 B)
Optimizer params: 21,130 (82.54 KB)

- The **TabNet model** was evaluated using the following metrices,
  - **Precision**: The ratio of true positive predictions to the total positive predictions.
  - **Recall**: The ratio of true positive predictions to the total actual positives.
  - **F1 Score:** The harmonic mean of precision and recall, balancing both metrics.
  - **Support**: The number of actual occurrences of each class in the dataset.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	2659
1	1.00	0.98	0.99	1898
2	0.98	1.00	0.99	2245
3	1.00	1.00	1.00	2720
accuracy			0.99	9522
macro avg	0.99	0.99	0.99	9522
weighted avg	0.99	0.99	0.99	9522

3.4.5 Conclusion / Model Comparison

```
Model Comparison:
                        Accuracy
Logistic Regression     0.896271
Random Forest           1.000000
SVM                     0.998191
XGBoost                 0.997487
Deep Learning Model     0.992462
TabNet                  0.993768
```

- **TabNet** performed well with 99.3 accuracy.
- **Sequential model** (deep learning model) provided an accuracy of 99.2 less than the TabNet.
- **XGBoost** and **SVM** also performed exceptionally well, with accuracies of 99.7 and 99.8, respectively.
- **Logistic Regression** had the lowest accuracy 89.6, indicating it was the least effective model in this scenario.
- **Random Forest** provided unrealistic 100 accuracy.