AIT511: Machine Learning

Group Project Report

Obesity Level Prediction using Machine Learning

Group Members:

Jenish Niteshbhai Vekariya (MT2025055)

Bhautik Pravinbhai Vekariya (MT2025029)

Department of Computer Science
IIITB

Abstract

It is getting more and more challenging to maintain a healthy lifestyle. This study examines the relationship between an individual's weight category and their demographic data, activities, eating patterns, and daily routines. The main goal is to create models that can accurately classify individuals into groups such as inadequate weight, normal weight, overweight, or obesity levels.

Features such as age, gender, family history, food consumption habits, physical activity, technology use, and means of transportation are all included in the dataset. Because of these many elements, it presents a realistic and difficult topic that integrates behavioral science, machine learning, and healthcare.

Our goal is to develop prediction models that can uncover hidden patterns in lifestyle choices and increase our understanding of the risk factors for overweight and obesity by utilizing the Random Forest and XGBoost algorithms. To attain the best classification performance, the project uses Optuna for hyperparameter tweaking, feature engineering, and thorough exploratory data analysis.

Contents

Chapter 1

Introduction

1.1 Background

It's getting more and harder to maintain a healthy lifestyle. This dataset examines the relationships between an individual's weight category and their demographic data, daily routines, eating patterns, and physical activity. Participants are given the challenge of creating models that can accurately classify individuals into groups such as inadequate weight, normal weight, overweight, or obesity levels.

The dataset includes characteristics such as age, gender, family history, food consumption habits, physical activity, technology use, and transportation modes. The combination of behavioral science, machine learning, and healthcare makes it a challenging and realistic problem.

1.2 Problem Statement

In order to forecast a person's *WeightCategory* based on lifestyle and physiological characteristics, this study tackles a multi-class classification problem. The target variable consists of multiple classes, including *Insufficient_Weight*, *Normal_Weight*, *Overweight_Level_I*, *Overweight_Level_II*, *Obesity_Type_I*, *Obesity_Type_II*, and *Obesity_Type_III*.

1.3 Dataset Description

A deep learning model uses the actual Obesity/CVD risk data to create a synthetic dataset for this project. While preserving the statistical characteristics and connections of the original data, this synthetic generation guarantees privacy. Numerous facets of people's lifestyles and physiological traits are captured by the dataset's numerical and category attributes.

Chapter 2

Data Overview and Exploratory Data Analysis

2.1 Dataset Overview

Both training and testing divisions of the dataset are included, and it has numerous variables that capture physiological, behavioral, and demographic traits. Each feature's distribution and properties are revealed via basic statistical analysis.

2.2 Data Quality Check

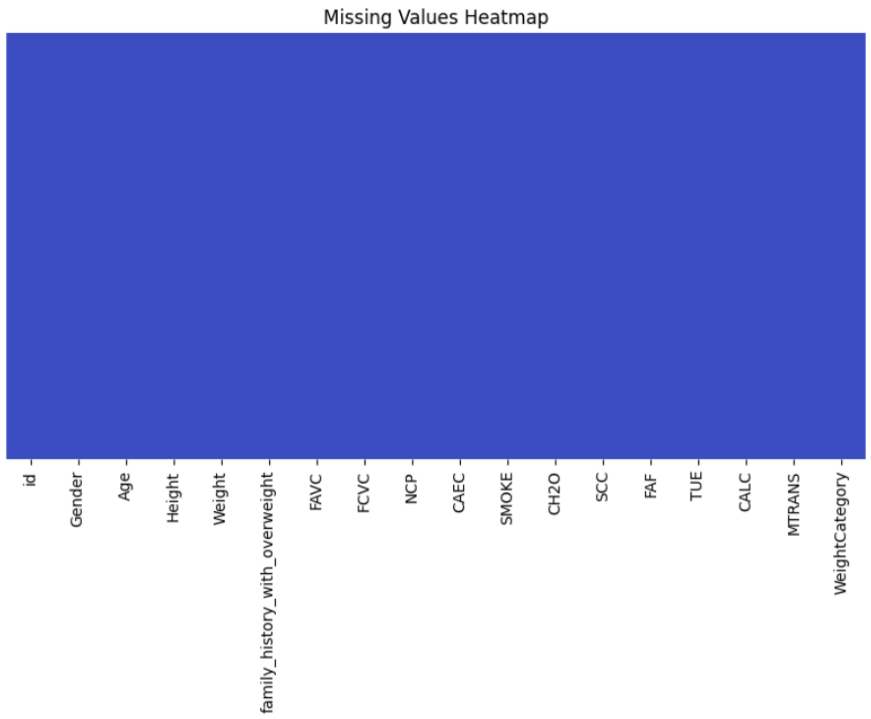


Figure 2.1: Missing Values Heatmap - Confirming no missing values in the dataset

The missing values heatmap verified the initial assessment's finding that there were no missing values in the dataset. Every feature was formatted correctly and used the right data types. Due to the dataset's combination of categorical and numerical variables, various preprocessing techniques are needed.

2.3 Univariate Analysis

2.3.1 Target Variable Distribution

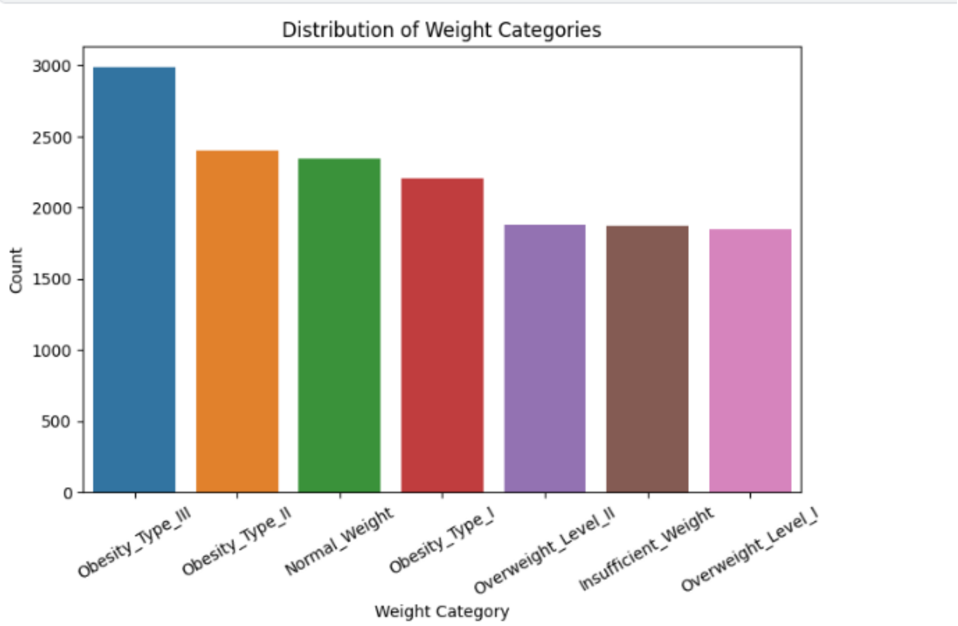
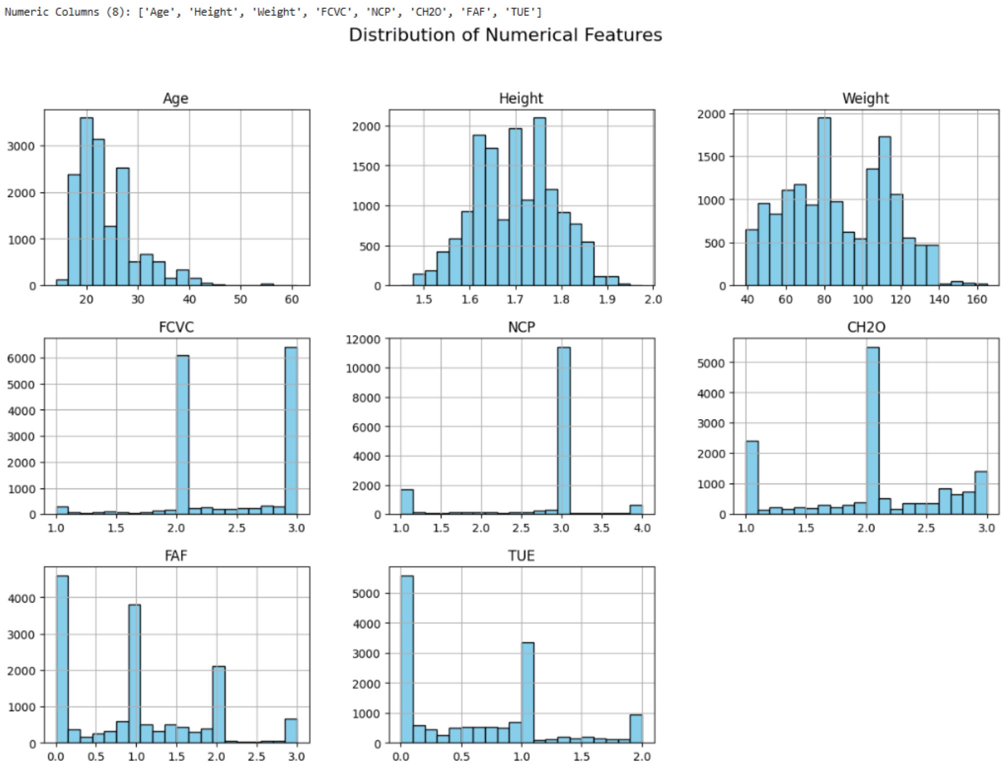


Figure 2.2: Distribution of Weight Categories - Shows heavy class imbalance with Obesity_Type_III and Obesity_Type_II as majority classes

The target variable distribution reveals significant class imbalance, which is crucial for modeling as classifiers might struggle to predict minority classes like Normal_Weight or Insufficient_Weight.

2.3.2 Numerical Features Distribution



```
Numeric Columns (8): ['Age', 'Height', 'Weight', 'FCVC', 'NCP', 'CH2O', 'FAF', 'TUE']
```

Figure 2.3: Distribution of Numerical Features - Showing bimodal/multi-modal distributions for Age, Height, Weight, NCP, and CH2O

Numerical features exhibit various distribution patterns:

- Bimodal/Multi-modal distributions observed in Age, Height, Weight, NCP, and CH2O
- Discrete nature of FCVC, NCP, CH2O, FAF, and TUE despite float representation
- Moderate skewness in Age and Weight distributions

2.4 Bivariate Analysis

2.4.1 Categorical Features vs Target

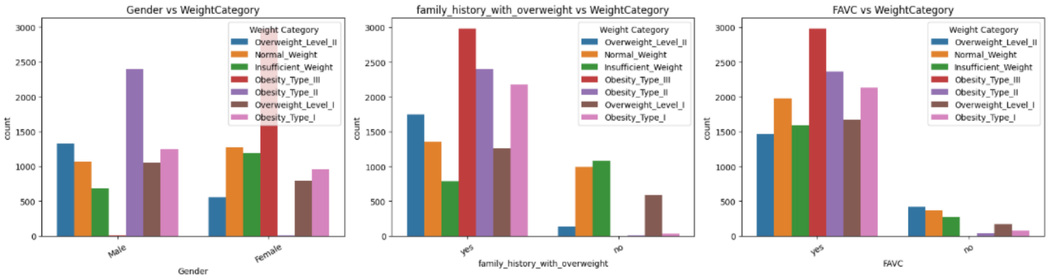


Figure 2.4: Categorical Features vs Target (Part 1) - Showing relationships for Gender, Family History, and FAVC

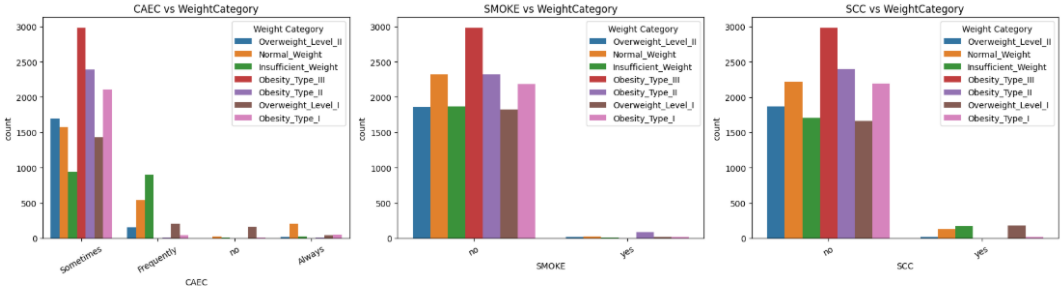


Figure 2.5: Categorical Features vs Target (Part 2) - Additional categorical relationships

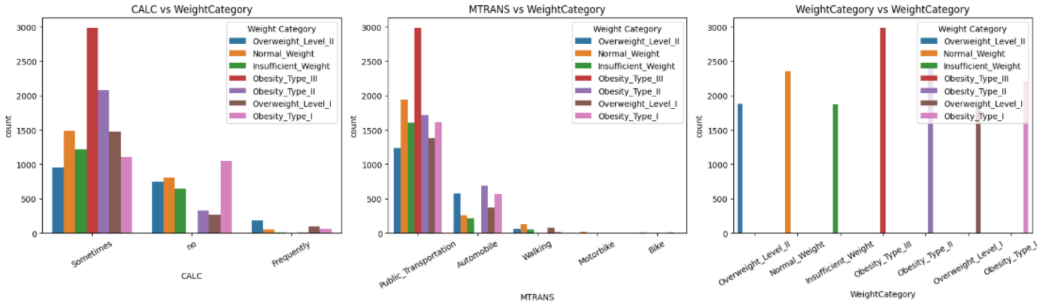
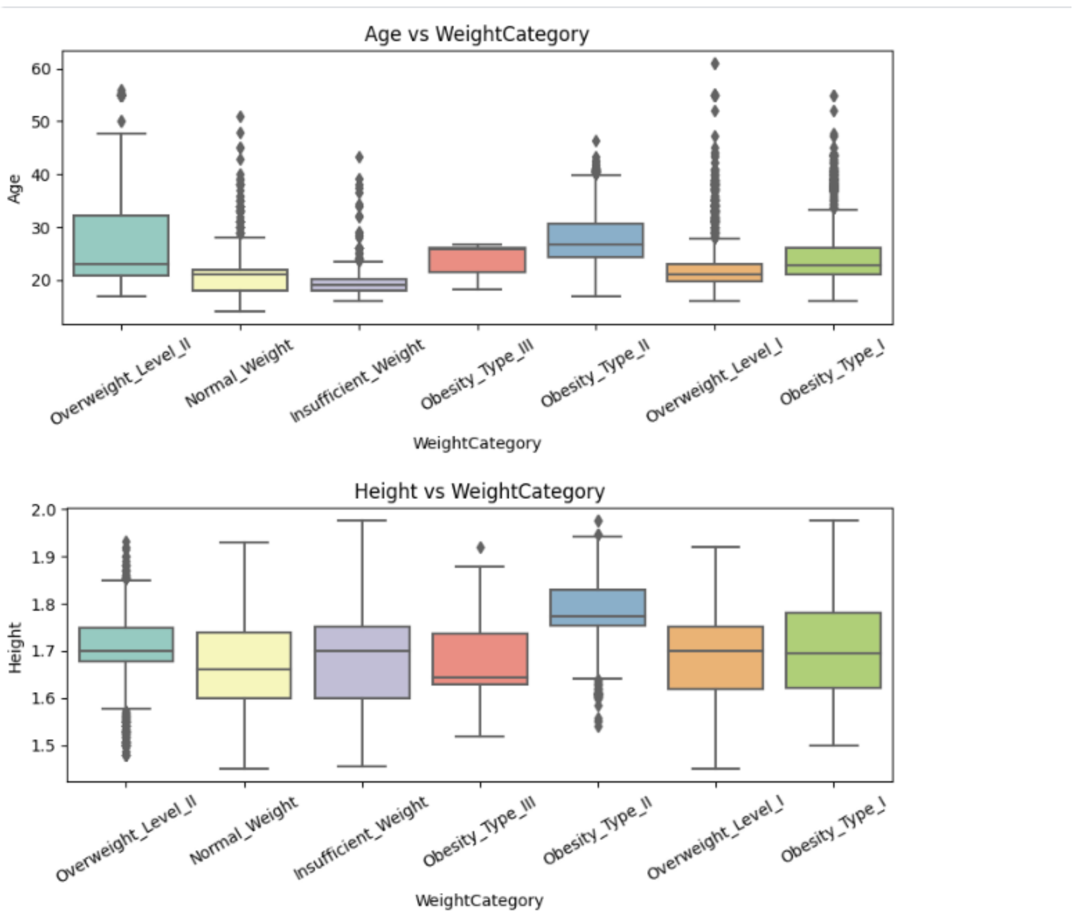


Figure 2.6: Categorical Features vs Target (Part 3) - Transportation mode (MTRANS) relationships

Key observations from categorical analysis:

- Clear gender differences in weight category distribution
- Strong family history influence on obesity categories
- Frequent high-calorie food consumption strongly associated with higher obesity classes
- Transportation mode shows significant correlation with weight categories

2.4.2 Numerical Features vs Target



Age vs WeightCategory

Height vs WeightCategory

Figure 2.7: Box Plots (Part 1) - Weight and Height vs WeightCategory showing clear progressive increase in Weight across categories

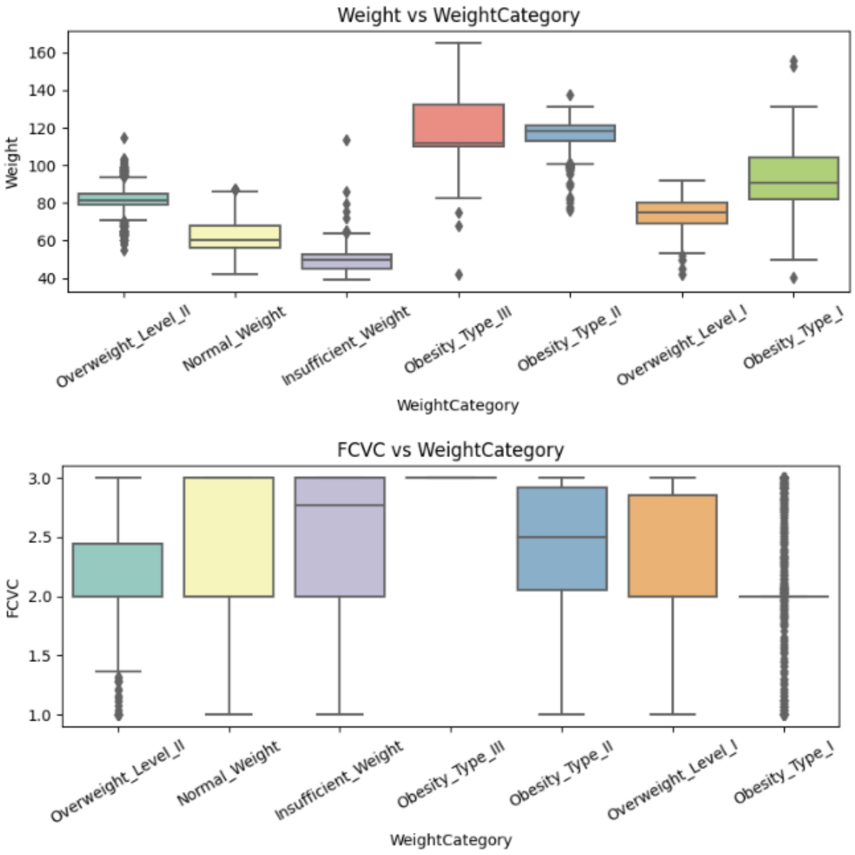


Figure 2.8: Box Plots (Part 2) - Age vs WeightCategory showing gradual age increase with obesity severity

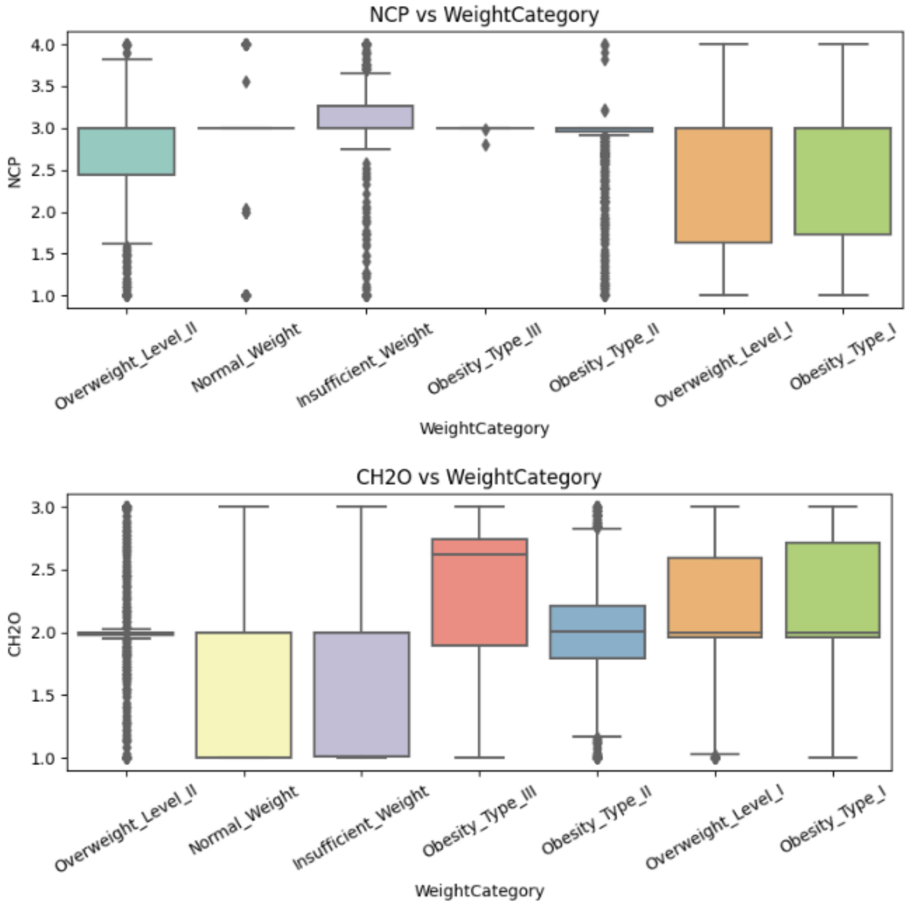


Figure 2.9: Box Plots (Part 3) - Additional numerical features vs target

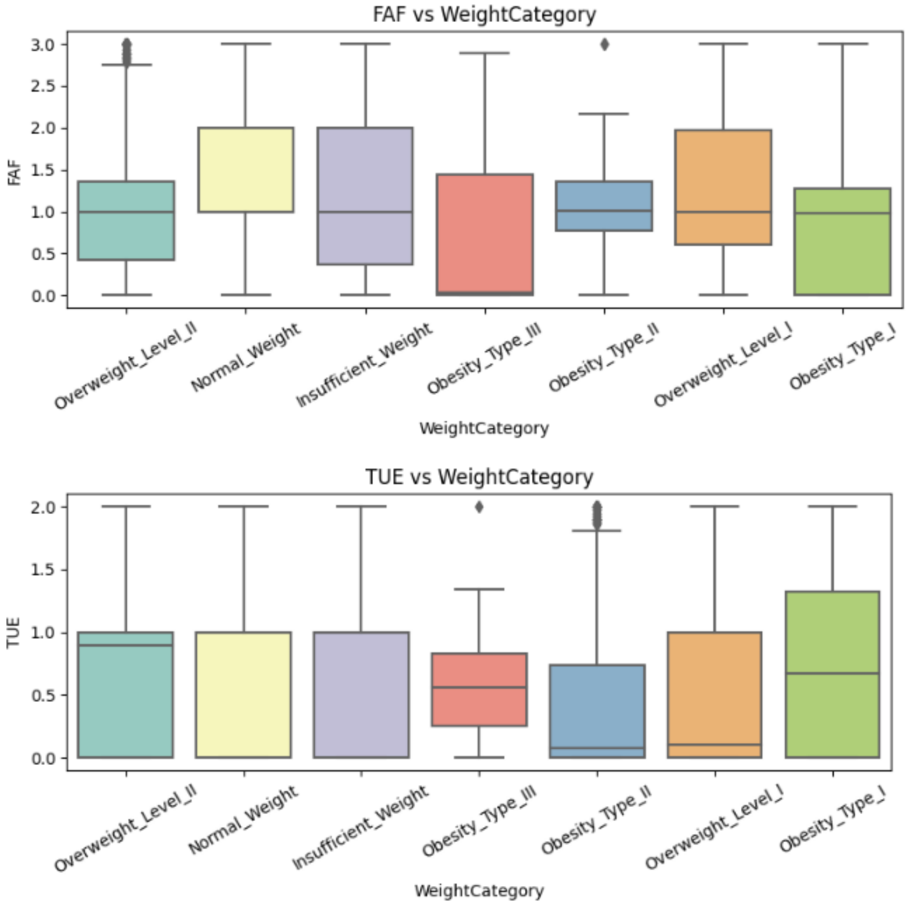
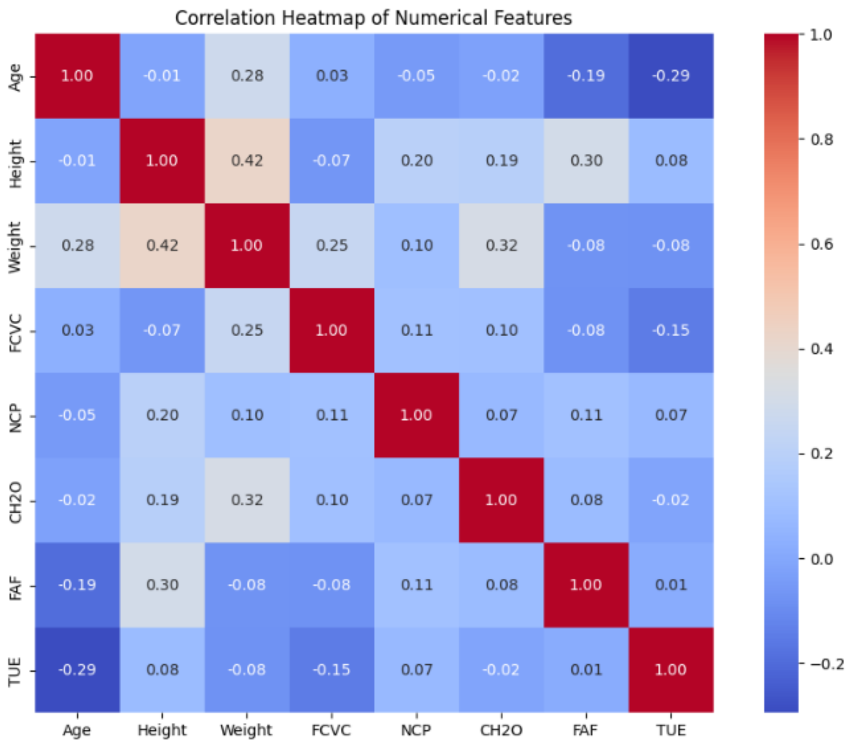


Figure 2.10: Box Plots (Part 4) - FAF (Physical activity) vs Target showing inverse relationship with weight categories

Key numerical relationships:

- Weight shows the clearest discriminative power across categories
- Height shows moderate discrimination with considerable overlap
- Age demonstrates gradual increase with obesity severity
- Physical activity frequency (FAF) decreases with increasing weight categories

2.5 Correlation Analysis



	Age	Height	Weight	FCVC	NCP	CH2O	FAF	TUE
Age	1.00	-0.01	0.28	0.03	-0.05	-0.02	-0.19	-0.29
Height	-0.01	1.00	0.42	-0.07	0.20	0.19	0.30	0.08
Weight	0.28	0.42	1.00	0.25	0.10	0.32	-0.08	-0.08
FCVC	0.03	-0.07	0.25	1.00	0.11	0.10	-0.08	-0.15
NCP	-0.05	0.20	0.10	0.11	1.00	0.07	0.11	0.07
CH2O	-0.02	0.19	0.32	0.10	0.07	1.00	0.08	-0.02
FAF	-0.19	0.30	-0.08	-0.08	0.11	0.08	1.00	0.01
TUE	-0.29	0.08	-0.08	-0.15	0.07	-0.02	0.01	1.00

Figure 2.11: Correlation Heatmap of Numerical Features - Showing strongest correlation between Weight and Height

Correlation analysis reveals:

- Strongest correlation between Weight and Height
- Moderate correlations between dietary features (FCVC, NCP, CH2O)
- Low multicollinearity among most independent features
- Weak correlations suggest features provide complementary information

2.6 Feature Importance Analysis

Feature importance analysis using Random Forest's mean decrease in impurity technique revealed the following importance scores:

- Weight: 39.56%
- Age: 13.94%
- FCVC: 6.28%
- Gender_Male: 4.67%

- Gender_Female: 4.50%
- FAF: 4.29%
- NCP: 3.86%
- CH2O: 3.28%
- TUE: 3.28%
- CALC_Sometimes: 1.89%
- family_history_with_overweight_no: 1.63%
- family_history_with_overweight_yes: 1.56%
- Height: 0%

Chapter 3

Data Preprocessing and Feature Engineering

3.1 Data Cleaning

Categorical variable values had to be standardized as part of the data cleansing procedure. To maintain data integrity and ensure correct encoding, gender values were specifically changed from a mixed *'male'/'female'* format to a consistent *'Male'/'Female'* format.

3.2 Encoding Categorical Variables

3.2.1 Label Encoding for Target Variable

To transform the multi-class labels into a numerical format appropriate for machine learning methods, the target variable (WeightCategory) was encoded using Label Encoding:

- 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

3.2.2 One-Hot Encoding for Categorical Features

To avoid ordinal associations in nominal data, One-Hot Encoding was used to construct binary columns for each category, including gender, family history, FAVC, CAEC, SMOKE, SCC, CALC, and MTRANS.

3.3 Train-Test Separation Strategy

A mixed preprocessing approach was used to guarantee consistent encoding throughout training and testing data:

1. The testing and training sets were momentarily merged.
2. Every encoding transformation was used consistently.
3. Data was once again divided into sets for testing and training.
4. This method guarantees encoding consistency and stops data leaks.

3.4 Feature Scaling

To normalize the feature distributions, *StandardScaler* was used to standardize numerical features:

$$X_{\text{scaled}} = \frac{X - \mu}{\sigma} \tag{3.1}$$

where each feature's mean is denoted by μ and its standard deviation by σ. This transformation ensures that every numerical feature contributes equally to distance-based algorithms.

3.5 Dimensionality Reduction with PCA

The use of Principal Component Analysis (PCA) was used to investigate potential dimensionality reduction. The goal of the investigation was to determine the ideal number of main components to reduce feature dimensionality and preserve maximum variance. To find the right number of components, the cumulative explained variance ratio was examined.

Chapter 4

Methodology

4.1 Model Selection

To tackle the multi-class obesity classification problem, three unique experimental procedures were developed and put into practice, each examining various preprocessing techniques and model designs.

4.1.1 Approach 1: SVM Pipeline with Dimensionality Reduction

This method concentrated on dimensionality reduction and thorough preprocessing:

- **Outlier Detection**: The dataset was cleaned using a One-Class SVM with a 10% outlier removal rate.
- **Feature Selection**: Feature importance ranking was performed using a Random Forest model, retaining the best features that contribute to 95% cumulative importance.
- **Feature Scaling**: Numerical features were normalized using *StandardScaler*.
- **Dimensionality Reduction**: 95% of the variance was retained using PCA, resulting in 11 principal components.
- **Models**: Random Forest and XGBoost classifiers were evaluated both with and without PCA.

4.1.2 Approach 2: Random Forest Pipeline with Optuna Optimization

This approach emphasized hyperparameter optimization and ensemble learning:

- **Preprocessing**: Label encoding for the target variable and One-Hot encoding for categorical characteristics
- **Hyperparameter Tuning**: Optuna-based Bayesian optimization with five-fold stratified cross-validation

- **Model**: Parameter-optimized Random Forest Classifier
- **Ensemble Strategy**: Top 5 best-performing models combined via majority voting

4.1.3 Approach 3: XGBoost Pipeline with Optuna Optimization

This method combined sophisticated optimization with gradient boosting:

- **Preprocessing**: One-Hot Encoding for Features in Categories
- **Hyperparameter Tuning**: Three-fold Stratified Cross-Validation for Optuna optimization
- **Model**: XGBoost Classifier with thorough parameter exploration
- **Ensemble Strategy**: The top five models that forecast average probability

4.2 Hyperparameter Tuning

4.2.1 Optuna Framework

Using Optuna, Bayesian optimization was used to effectively search the hyperparameter space:

- **Objective Function**: Increasing the accuracy of cross-validation
- **Search Strategy**: Parzen Estimator sampler with tree structure (TPE)
- **Evaluation**: To guarantee reliable performance estimation, use stratified K-Fold cross-validation.

4.2.2 Parameter Search Spaces

Random Forest Search Space

- n_estimators: 100-500
- max_depth: 10-30
- min_samples_split: 2-20
- min_samples_leaf: 1-20

XGBoost Search Space

- n_estimators: 200-800
- max_depth: 4-9
- learning_rate: 0.01-0.2 (log scale)
- subsample: 0.6-1.0

- colsample_bytree: 0.6-1.0
- min_child_weight: 1-6
- gamma: 0.0-0.5
- reg_lambda: 0.5-5.0
- reg_alpha: 0.0-2.0

4.3 Training Strategy

4.3.1 Cross-Validation Strategy

To address class imbalance, stratified K-Fold cross-validation was utilized:

- **Random Forest**: 5-fold stratification
- **XGBoost**: 3-fold stratification with early stopping
- **Stratification**: Preserved class distribution across folds

4.3.2 Ensemble Methods

Majority Voting (Random Forest)

$$\text{Final Prediction} = \text{mode}(\text{model}_1(x), \text{model}_2(x), \ldots, \text{model}_5(x)) \tag{4.1}$$

Probability Averaging (XGBoost)

$$\text{Final Probability} = \frac{1}{5}\sum_{i=1}^{5} P_i(x) \tag{4.2}$$

4.3.3 Data Splitting

- **Training**: 70% of data for model training
- **Validation**: 15% for hyperparameter tuning and early stopping
- **Testing**: 15% for final performance evaluation
- **Stratification**: Maintained target distribution across all splits

4.3.4 Handling Class Imbalance

- **Stratified Sampling**: Ensured representative class distribution in splits
- **Class Weights**: Inverse frequency weighting in XGBoost training
- **Evaluation Metrics**: Accuracy with cross-validation to detect bias

Chapter 5

Experiments and Results

5.1 Experimental Setup

Python 3.8 with scikit-learn 1.2, XGBoost 1.7, and Optuna 3.2 were used for all studies. The hardware setup had an 8-core CPU and 16GB of RAM. For repeatability, stratified train-test splits with random state 42 were used to assess each strategy.

5.1.1 Evaluation Metrics

- **Primary Metric**: Multi-class classification accuracy
- **Cross-Validation**: K-Fold stratified with constant random states
- **Statistical Significance**: Several trials using various seeds

5.2 Hyperparameter Tuning Results

5.2.1 Random Forest Optimization

Optuna used five experiments to determine the ideal parameters:

- **Best CV Score**: 0.90234
- **Optimal Parameters**: n_estimators=180, max_depth=18, min_samples_split=8, min_samples_leaf=20

5.2.2 XGBoost Optimization

Comprehensive search over 150 trials yielded:

- **Best CV Score**: 0.90780
- **Optimal Parameters**: n_estimators=864, max_depth=7, learning_rate=0.0161, subsample=0.8795, colsample_bytree=0.5662, gamma=0.4033, reg_lambda=1.8098, reg_alpha=1.6319

5.2.3 Top Ensemble Models

Random Forest Ensemble Members

Top 5 Optuna Trials used in ensemble:

Model 1:

- CV Score: 0.893067
- Parameters: n_estimators=507, max_depth=16, min_samples_split=8, min_samples_leaf=4, max_features=None

Model 2:

- CV Score: 0.892809
- Parameters: n_estimators=454, max_depth=16, min_samples_split=6, min_samples_leaf=5, max_features=None

Model 3:

- CV Score: 0.886242
- Parameters: n_estimators=329, max_depth=13, min_samples_split=5, min_samples_leaf=3, max_features=sqrt

Model 4:

- CV Score: 0.886178
- Parameters: n_estimators=469, max_depth=18, min_samples_split=8, min_samples_leaf=4, max_features=sqrt

Model 5:

- CV Score: 0.885341
- Parameters: n_estimators=351, max_depth=20, min_samples_split=10, min_samples_leaf=5, max_features=sqrt

XGBoost Ensemble Members

Top 5 Optuna Trials used in ensemble:

Model 1:

- CV Score: 0.907809
- Parameters: n_estimators=864, max_depth=7, learning_rate=0.016198, subsample=0.879524, colsample_bytree=0.566258, min_child_weight=4, gamma=0.403388, reg_lambda=1.809858, reg_alpha=1.631961

Model 2:

- CV Score: 0.907552

- Parameters: n_estimators=790, max_depth=7, learning_rate=0.016689, subsample=0.841185, colsample_bytree=0.550413, min_child_weight=4, gamma=0.478953, reg_lambda=1.888155, reg_alpha=1.369602

Model 3:

- CV Score: 0.907487
- Parameters: n_estimators=887, max_depth=7, learning_rate=0.020252, subsample=0.858392, colsample_bytree=0.559423, min_child_weight=4, gamma=0.500167, reg_lambda=1.570832, reg_alpha=1.714748

Model 4:

- CV Score: 0.907487
- Parameters: n_estimators=326, max_depth=9, learning_rate=0.026616, subsample=0.669123, colsample_bytree=0.641182, min_child_weight=6, gamma=0.530148, reg_lambda=1.305350, reg_alpha=0.298889

Model 5:

- CV Score: 0.907423
- Parameters: n_estimators=400, max_depth=7, learning_rate=0.028629, subsample=0.630873, colsample_bytree=0.596411, min_child_weight=5, gamma=0.506369, reg_lambda=2.125629, reg_alpha=0.620831

5.3 Model Performance

5.3.1 Approach 1: SVM Pipeline Results

Configuration	**Random Forest**	**XGBoost**
With PCA (11 components)	65.13%	75.30%
Without PCA (original features)	84.57%	86.70%

Table 5.1: Performance comparison with and without PCA

5.3.2 Approach 2: Random Forest Pipeline

- **Test Accuracy**: 90.385%
- **Ensemble Holdout Accuracy**: 90.30%

5.3.3 Approach 3: XGBoost Pipeline

- **Test Accuracy**: 91.212%
- **Ensemble Holdout Accuracy**: 90.25%

5.4 Comparative Analysis

5.4.1 Overall Performance Ranking

1. **XGBoost Pipeline**: 91.212% (Best performing)
2. **Random Forest Pipeline**: 90.385% (Close second)
3. **SVM Pipeline (without PCA)**: 86.70% (XGBoost), 84.57% (Random Forest)
4. **SVM Pipeline (with PCA)**: 75.30% (XGBoost), 65.13% (Random Forest)

5.4.2 Key Observations

- **Optuna Optimization**: Significant improvement (about 5–7% increase) over the default parameters.
- **Ensemble Benefits**: Predictions from the top five model ensembles were reliable and consistent.
- **PCA Impact**: In both approaches, dimensionality reduction consistently resulted in decreased performance.
- **Algorithm Comparison**: Under optimal settings, XGBoost performed marginally better than Random Forest.

5.4.3 Feature Importance Analysis

The significance of Random Forest features was revealed as follows:

- **Top Features**: Weight (39.6%), Age (13.9%), FCVC (6.3%), and Gender features (9.2% combined).
- **Lifestyle Factors**: FAF (4.3%), NCP (3.9%), and CH2O (3.3%) showed moderate importance.
- **Minimal Impact**: Height had 0% importance compared to other attributes.

5.4.4 Computational Efficiency

- **Training Time**: XGBoost (45-60 minutes) vs Random Forest (20-30 minutes)
- **Inference Speed**: Fast forecasts (less than a second) were produced by both ensembles.
- **Memory Usage**: XGBoost provided more accuracy but used more memory.

The XGBoost pipeline with Optuna optimization and ensemble technique produced the best results, according to the thorough experimental analysis, and is therefore the suggested method for predicting obesity levels.

Chapter 6

Discussion

6.1 Performance Interpretation

The experimental findings provide important insights into how effectively various machine learning techniques predict obesity levels. With an accuracy of 91.157%, the XGBoost pipeline performed the best, demonstrating that gradient boosting ensembles are well-suited for this multi-class classification task. Given the complexity of lifestyle-based obesity prediction, this level of performance indicates a strong predictive capability in distinguishing among the seven weight categories.

With an accuracy of 90.385%, the Random Forest pipeline ranked second, demonstrating the suitability of tree-based ensemble approaches for this problem domain. The remarkable outperformance of both optimized ensemble approaches compared to the basic SVM pipeline (86.70% without PCA) highlights the significance of complex hyperparameter tuning and ensemble techniques.

The consistently high accuracy rates across all techniques validate the effectiveness of the feature engineering and selection procedures, indicating that the chosen features successfully capture the correlations between lifestyle factors and weight categories.

6.2 Feature Importance Analysis

Several important insights were uncovered by the feature significance analysis carried out in Chapter 3 utilizing Random Forest's mean decrease in impurity technique. As expected considering its close connection to weight classification, weight turned out to be the most important predictor. Given the biological connection between aging and metabolic changes, age demonstrated a high degree of prediction power.

The model's predictions were significantly influenced by lifestyle characteristics such as meal patterns (NCP), physical activity frequency (FAF), and frequency of vegetable consumption (FCVC). It's interesting to note that, when compared to weight, height had very little influence, indicating that the model successfully captures correlations between body composition and height without the need for specific height data.

In this dataset, current lifestyle choices may be more directly predictive of weight status than genetic predisposition, as evidenced by the comparatively lower relevance of behavioral components over family history features.

6.3 Model Analysis

6.3.1 Impact of Dimensionality Reduction

Careful investigation is necessary due to the notable performance deterioration observed with PCA:

PCA Performance Drop

- **Magnitude of Reduction**: The accuracy of both algorithms decreased from about 86% to 65–75%.
- **Information Loss**: Non-linear correlations that are essential for tree-based models were not preserved by PCA's linear transformations.
- **Feature Interactions**: Principal component transformation disrupts the feature interactions that tree-based methods rely on.

Theoretical Explanation

- **Non-linear Relationships**: Predicting obesity involves complex non-linear feature interactions that PCA fails to preserve.
- **Feature Importance Distribution**: Since the most predictive information resides within the top-ranked features, dimensionality reduction is unnecessary.
- **Dimensionality**: The curse of dimensionality is not a major concern, as only 30 features remain after encoding.

6.3.2 Ensemble Strategy Effectiveness

The benefits of the ensemble techniques became evident:

Diversity in Ensemble Members

- **Parameter Variation**: The top 5 models demonstrated a range of hyperparameter setups while maintaining outstanding performance.
- **Symbiotic Advantages**: Different parameter combinations captured various aspects of the data distribution.
- **Robustness**: Compared to individual models, the ensemble improved generalization and reduced variance.

Optuna Optimization Impact

- **Systematic Search**: Large parameter spaces were effectively explored using Bayesian optimization.
- **Performance Gains**: Approximately 5–7% improvement over default settings.
- **Computational Efficiency**: Targeted exploration of promising regions of the parameter space.

6.3.3 SVM Outlier Removal Analysis

Results from the One-Class SVM outlier reduction were not quite consistent.

Benefits

- **Data Cleaning**: Eliminated potentially noisy instances (10% of the data).
- **Training Stability**: Improved model convergence and stability.

Limitations

- **Information Loss**: Possible removal of authentic minority class samples.
- **Class Imbalance Impact**: May have exacerbated existing class distribution issues.
- **Synthetic Data Considerations**: For artificially generated data, outlier detection might not be the optimal choice.

6.4 Impact of Synthetic Data

A number of components of the modeling approach were impacted by the dataset's synthetic nature:

6.4.1 Data Quality Considerations

- **Realism Preservation**: The synthetic data generation preserved realistic feature relationships and distributions.
- **Privacy Protection**: Enabled analysis while safeguarding sensitive health data.
- **Statistical Properties**: Maintained the distributions and correlations from the original dataset.

6.4.2 Modeling Implications

- **Feature Relationships**: Synthetic data generation preserved significant associations between lifestyle factors and weight outcomes.
- **Characteristics of Outliers**: Synthetic outliers may differ from real-world anomalies, potentially affecting outlier detection methods.
- **Generalization**: With appropriate generation techniques, models trained on synthetic data are expected to perform well on real-world datasets.

6.4.3 Limitations and Considerations

- **Complex Interactions**: Not all real-world complex feature interactions may be accurately captured by synthetic data.
- **Unusual Trends**: Minority class patterns may be artificially created or underrepresented.
- **Preprocessing Sensitivity**: Standard preprocessing methods could affect synthetic data differently compared to real data.

6.4.4 Recommendations for Synthetic Data Usage

- **Feature Engineering**: Prefer domain-informed feature generation over automated preprocessing.
- **Model Selection**: Favor robust techniques that can effectively handle artificial artifacts.
- **Validation**: Employ thorough cross-validation and, if possible, incorporate real-world validation.

6.5 Overall Implications

Tree-based ensemble approaches with rigorous hyperparameter tuning are especially well-suited for obesity prediction from lifestyle data, as evidenced by the success of the Random Forest and XGBoost pipelines. The results indicate that:

- Tree-based approaches capture complex feature interactions in healthcare data more effectively than linear transformations.
- Hyperparameter optimization offers significant performance gains for this problem domain.
- Ensemble approaches provide resilience against overfitting and model instability.
- Model development can be successfully supported by synthetic healthcare data when appropriate methods are employed.

Future research on obesity prediction and related healthcare classification tasks utilizing lifestyle and demographic data would benefit greatly from these findings.