Indian Institute of Technology, Dharwad

Project Report: Obesity Level Classification and Clustering Analysis

CS209, CS214

Course Instructor: Dr. Dileep A.D.

Mentor: Vikas Kumar

Team 18
Yashaswini L, Maisangari Ashwitha, Ayaan Karim, Ankit Khushwaha,

April 2025

Contents

1 Introduction

Obesity is a growing global health concern associated with chronic diseases such as type 2 diabetes, cardiovascular disease, and certain cancers. With its increasing prevalence, machine learning has become a powerful tool for analyzing health data and predicting the risk of obesity based on lifestyle, diet, and demographic factors.

This report analyzes the *Obesity Level Estimation* dataset, which includes **2111** records of individuals aged 14 to 61 from Mexico, Peru, and Colombia. It comprises **17** attributes that cover demographic information, eating habits, physical activity, and self-reported behaviors, collected through an anonymous online survey.

The project has two main objectives:

1. Develop supervised learning models to classify individuals into six levels of obesity based on BMI.

2. Apply unsupervised clustering to identify meaningful population segments based on shared health characteristics.

For classification, we employ models such as **Logistic Regression**, **Decision Trees**, **Random Forests**, **Support Vector Machines (SVM)** ,**Naive Bayes** and **K-Nearest Neighbours (KNN)** with a focus on identifying the most significant predictive features. Clustering techniques like **K-Means** and **Hierarchical Clustering** are used to uncover behavioral trends and risk profiles.

By integrating predictive modeling with pattern discovery, this study aims to support data-driven health interventions and contribute to obesity prevention strategies and public health planning.

2 Methodology

2.1 Data Preprocessing

2.1.1 Initial Exploration

The data set consists of 2,111 records and 17 columns, covering demographic information, physical characteristics, lifestyle habits, and a target label that indicates the level of obesity.

The following are the **numerical features** in the data set:

- `Age` – Age of the individual
- `Height` – Height in meters
- `Weight` – Weight in kilograms
- `FCVC` – Frequency of vegetable consumption
- `NCP` – Number of daily main meals
- `CH2O` – Daily water intake
- `FAF` – Frequency of physical activity
- `TUE` – Time spent using technology devices

The **categorical features** include:

- `Gender` – Male or Female
- `family_history_with_overweight` – Yes or No
- `FAVC` – Frequent consumption of high-caloric food (Yes/No)
- `CAEC` – Consumption of food between meals (No, Sometimes, Frequently, Always)
- `SMOKE` – Smoking habit (Yes/No)
- `SCC` – Monitoring of calorie consumption (Yes/No)
- `CALC` – Alcohol consumption frequency (No, Sometimes, Frequently, Always)
- `MTRANS` – Mode of transportation (Walking, Bike, Motorbike, Public Transportation, Automobile)

The target variable `NObeyesdad` represents the obesity level of individuals. It contains the following **seven classes**:

- `Insufficient_Weight` – BMI less than 18.5
- `Normal_Weight` – BMI between 18.5 and 24.9
- `Overweight_Level_I` – BMI between 25.0 and 29.9
- `Overweight_Level_II` – BMI between 30.0 and 34.9

- `Obesity_Type_I` – BMI between 35.0 and 39.9
- `Obesity_Type_II` – BMI between 40.0 and 44.9
- `Obesity_Type_III` – BMI 45.0 or greater

2.1.2 Duplicate and Missing Values

The dataset was thoroughly checked for data quality issues. It was found that there are **24 duplicate rows**, which were removed to avoid redundancy and potential bias during model training.

Additionally, **no missing values** were present in any of the columns. Hence, no imputation or row deletion was required for handling missing data.

2.1.3 Categorical Encoding

The categorical variables were encoded using different strategies based on their nature:

- Binary categorical variables (`Gender`, `family_history_with_overweight`, `FAVC`, `SCC`, and `SMOKE`) were label encoded (mapped to 0/1).
- Multi-class categorical variables (`CAEC`, `CALC`, and `MTRANS`) were one-hot encoded to create binary dummy variables.
- The target variable `NObeyesdad` (obesity level) was label encoded by mapping each class to a numerical value while preserving the ordinal relationship between categories.

2.1.4 Feature Selection

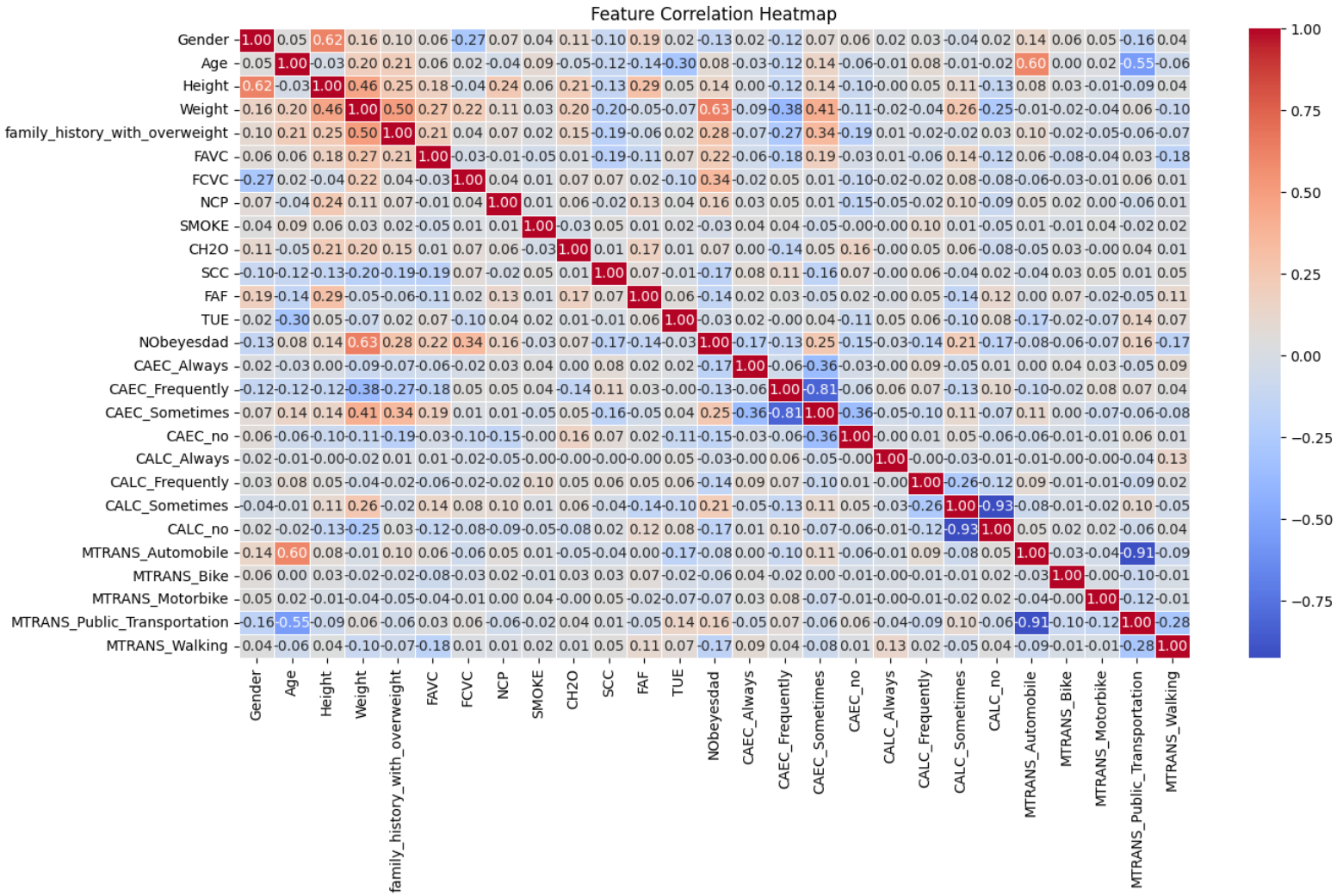

Figure 2.1: Correlation Heatmap

Features with weak correlation to the target **NObeyesdad** (absolute correlation < 0.1) were identified and removed from the dataset. The dropped features included: **Age**, **SMOKE**, **CH2O**, **TUE**, **CALC_Always**, **MTRANS_Automobile**, **MTRANS_Bike**, and **MTRANS_Motorbike**. This feature selection process helps reduce dimensionality while maintaining the most relevant predictors for obesity level classification.

2.1.5 Univarient analysis

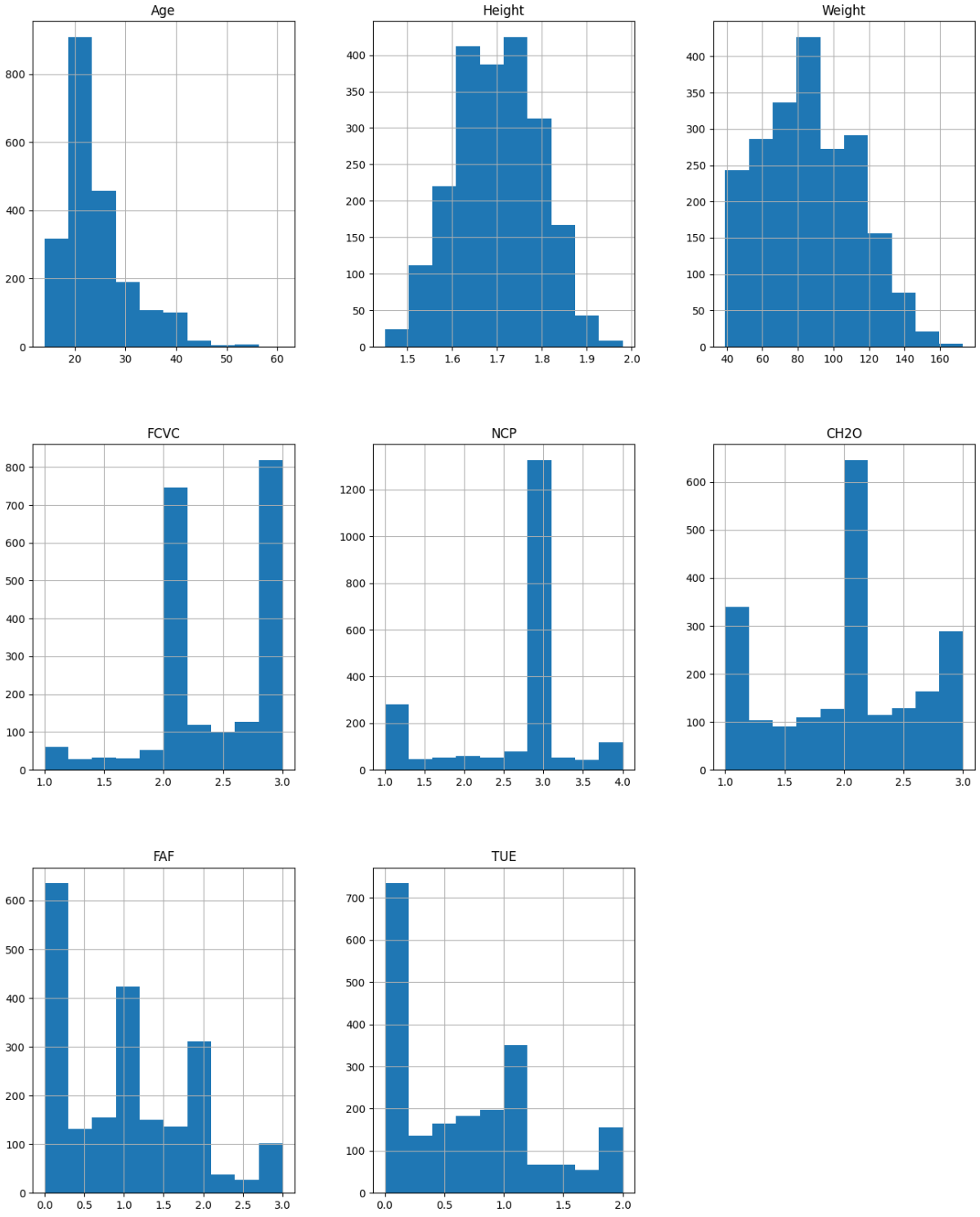

Figure 2.2: Univarient analysis

From the distribution plots, we can observe that the data for **family_history_with_overweight**, **FAVC** (Frequent Consumption of High-Caloric Food), **FCVC** (Frequency of Vegetable Consumption), and **SCC** (Calories Consumption Monitoring) are **poorly balanced**.

In contrast, the data for **Gender** and **NObeyesdad** (target variable) are **highly**

balanced.

2.1.6 Data distribution of numerical data

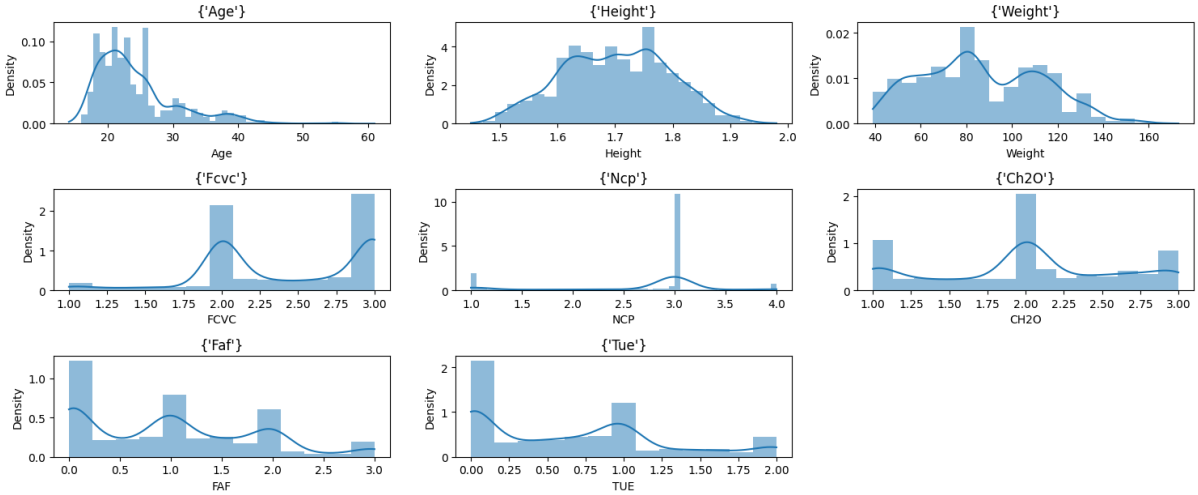

Figure 2.3: Data distribution of numerical data

2.1.7 Distribution of Target Value

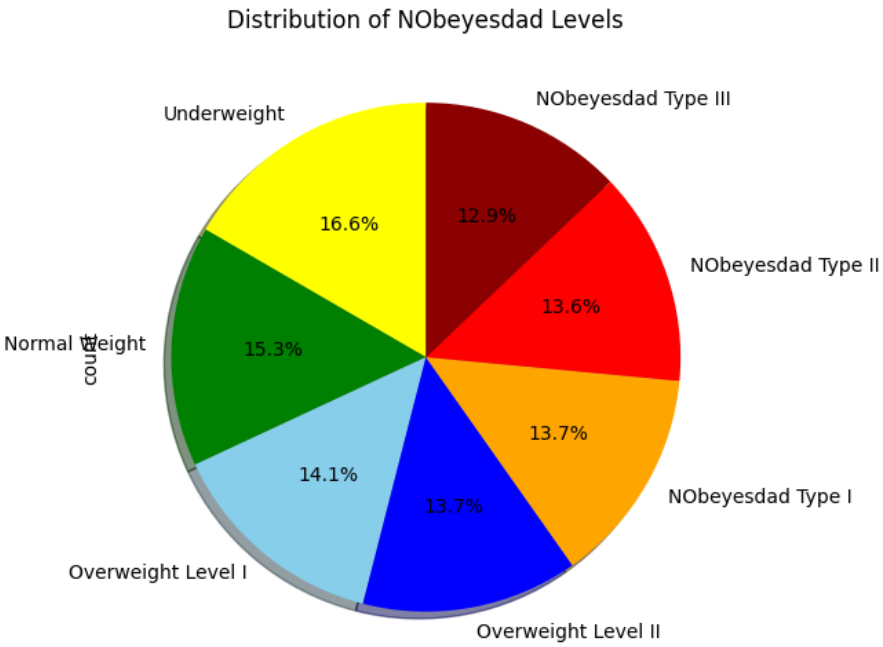

Figure 2.4: Distribution of NObeyesdad Values

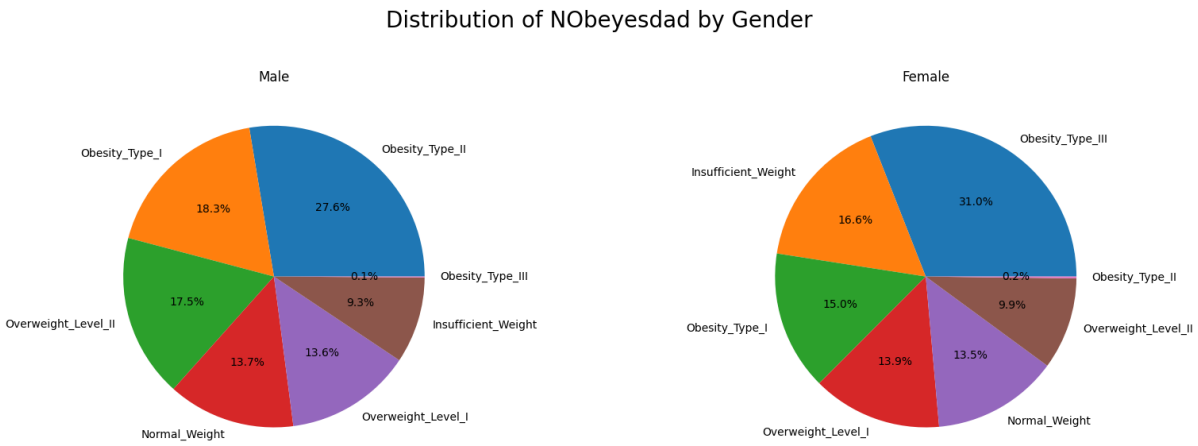

Figure 2.5: Distribution of NObeyesdad

Looking at the distribution of the NObeyesdad column, the values are relatively close to each other, ranging from 272 to 351 observations per class. The dataset is reasonably balanced.

2.1.8 Bivariate data analysis of data with Target

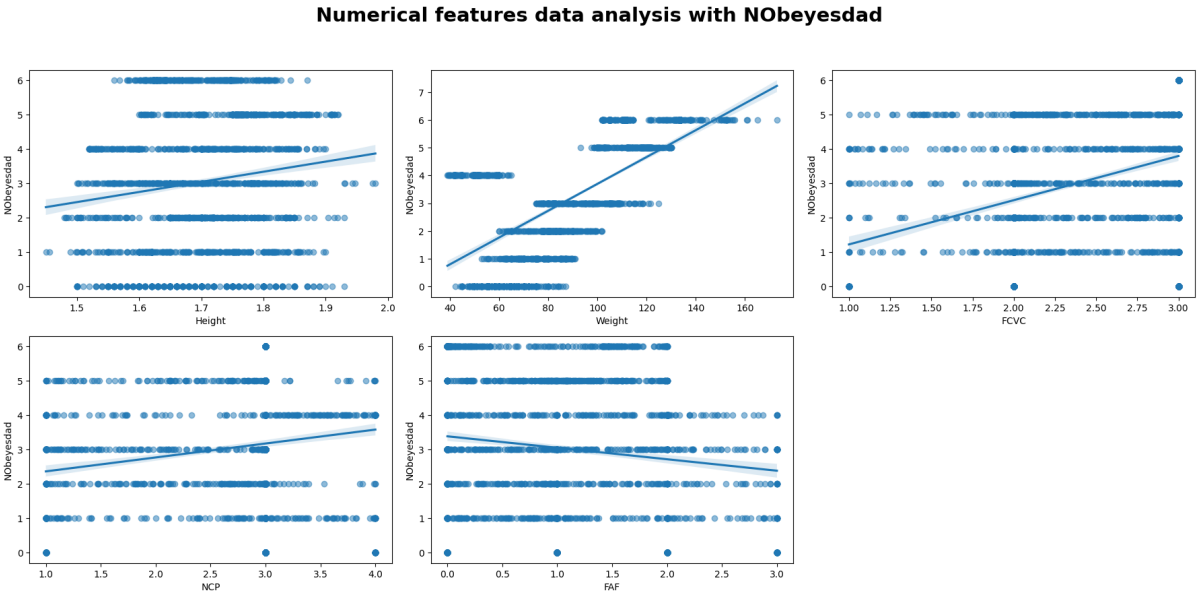

Figure 2.6: Numerical features data analysis with NObeyesdad

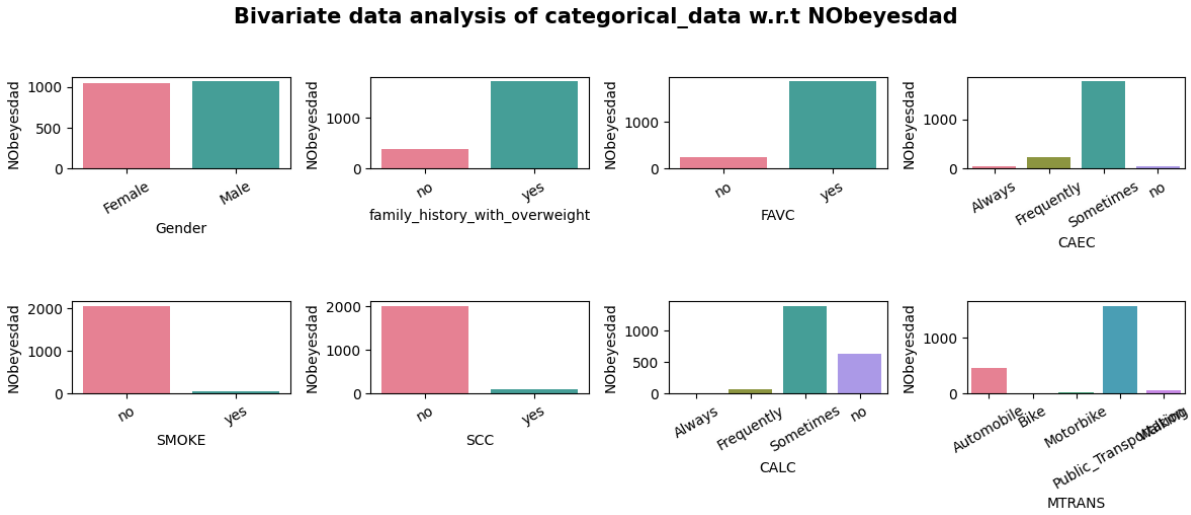

Figure 2.7: Bivariate data analysis of categorical data w.r.t NObeyesdad

2.1.9 Feature Engineering

To better represent obesity level, a new feature called `BMI` (Body Mass Index) was computed using the formula:

```
df_drop['BMI'] = df_drop['Weight'] / (df_drop['Height'] ** 2)
df_drop.drop(['Height','Weight'], axis = 1, inplace = True)
```

This derived feature simplifies further analysis, as BMI is a direct indicator of an individual's weight category. After computing BMI, the original `Height` and `Weight` columns were dropped from the dataset.

2.1.10 Multicollinearity Check using Variance Inflation Factor (VIF)

To ensure the reliability of the machine learning models and avoid issues arising from multicollinearity, we performed a Variance Inflation Factor (VIF) analysis. VIF quantifies how much the variance of a regression coefficient is inflated due to multicollinearity with other independent variables. A high VIF value indicates that the feature is highly correlated with other features, which can lead to unstable model estimates.

The VIF for each feature was computed using the following function:

VIF was calculated iteratively by excluding features that contributed to high multicollinearity. In each step, the feature with the highest VIF (e.g., `CAEC_Sometimes`, `CALC_Sometimes`, `FCVC`, `NCP`, etc.) was removed if it did not provide unique or essential information.

Finally, a refined dataset was obtained by dropping the following columns:

```
{ 'NObeyesdad', 'CAEC_Sometimes', 'CALC_Sometimes', 'FCVC', 'NCP',
                'family_history_with_overweight' }
```

This process helped to retain the most informative and non-redundant features, improving model interpretability and performance.

2.2 Train-Validation-Test Split and Feature Scaling

To develop reliable and generalizable machine learning models, the dataset was strategically divided into three parts:

- **Training Set (70%)** – Used to train the models.
- **Validation Set (15%)** – Used for tuning hyperparameters and model selection.
- **Test Set (15%)** – Reserved for final evaluation on unseen data.

A **stratified sampling strategy** was employed to ensure that the distribution of obesity classes in the `NObeyesdad` target variable remained consistent across all subsets. This is particularly important in multi-class classification problems to prevent model bias toward majority classes.

After splitting, all feature values were scaled using the **Min-Max Normalization** technique, which transforms the features to a range between 0 and 1. The scaler was fit exclusively on the training data to avoid information leakage and then applied to both the validation and test sets. This step is essential for algorithms sensitive to feature magnitude, such as **Support Vector Machines** and **K-Means Clustering**.

The result is a well-prepared dataset pipeline that ensures balanced training and fair performance evaluation across all models.

2.3 Model Training

2.3.1 Training Setup

The following machine learning models were implemented for obesity level prediction:

- **Logistic Regression** – A linear model useful for its interpretability and baseline comparison.
- **Support Vector Machine (SVM)** – A powerful classifier that performs well in high-dimensional spaces.
- **Naive Bayes Classifier** – A probabilistic model based on Bayes' theorem, effective with categorical data.
- **Decision Tree** – A tree-structured model known for its interpretability and simplicity.
- **K-Nearest Neighbours (KNN)** – A non-parametric model that classifies based on proximity in feature space.
- **Random Forest Classifier** – An ensemble of decision trees that improves accuracy and reduces overfitting.

2.3.2 Performance Metrics

To evaluate the performance of the classification models, the following metrics were used:

- **Accuracy:**
$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Predictions}}$$
Effective when class distribution is balanced.

- **Confusion Matrix:** Provides a summary of prediction results with insights into false positives and false negatives.

- **Precision:**
$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$
Indicates how many predicted positives were actually correct.

- **Recall (Sensitivity):**
$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$
Shows how many actual positives were correctly identified.

- **F1 Score:**
$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$
A balanced measure useful when both precision and recall are important.

2.4 Model: Logistic Regression

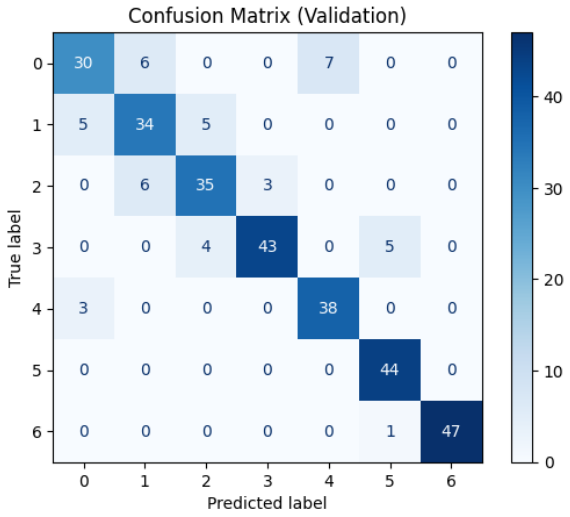

Figure 2.8: Confusion Matrix (Validation Set)

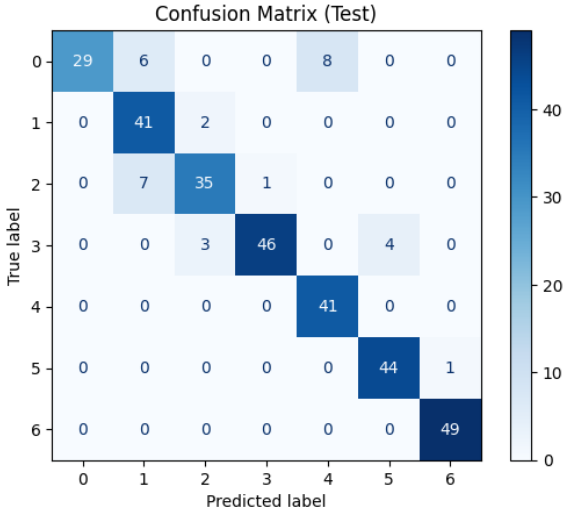

Figure 2.9: Confusion Matrix (Test Set)

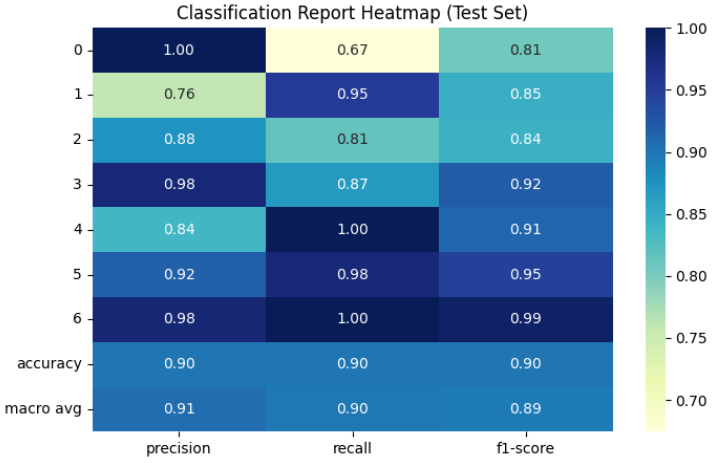

Figure 2.10: Heatmap of Confusion Matrix (Test Set)

2.4.1 Validation vs Test Performance

The performance of the Logistic Regression model on both the validation and test datasets is summarized below:

Table 2.1: Model Performance Comparison: Validation vs Test Set

Metric	Validation Set	Test Set
Accuracy	85.76%	89.91%
Precision	85.89%	91.08%
Recall	85.76%	89.91%
F1-Score	85.65%	89.73%

Observation: The model performs consistently well on both datasets, with slightly better results on the test set. The balanced precision and recall indicate strong generalization capability.

Classification Report (Test Set):

Table 2.2: Per-Class Performance Metrics (Test Set)

Class Label	Precision	Recall	F1-Score	Support
0	1.00	0.67	0.81	43
1	0.76	0.95	0.85	43
2	0.88	0.81	0.84	43
3	0.98	0.87	0.92	53
4	0.84	1.00	0.91	41
5	0.92	0.98	0.95	45
6	0.98	1.00	0.99	49
Accuracy	0.90 (on 317 samples)			
Macro Avg	0.91	0.90	0.89	317
Weighted Avg	0.91	0.90	0.90	317

Observation: The model achieves excellent precision and recall, especially for classes 5 and 6. However, class 0 (likely underweight) exhibits the lowest recall, indicating occasional underprediction. Overall, the results affirm Logistic Regression as a solid baseline model for this classification task.

2.5 Model: SVM(Support Vector Machine)

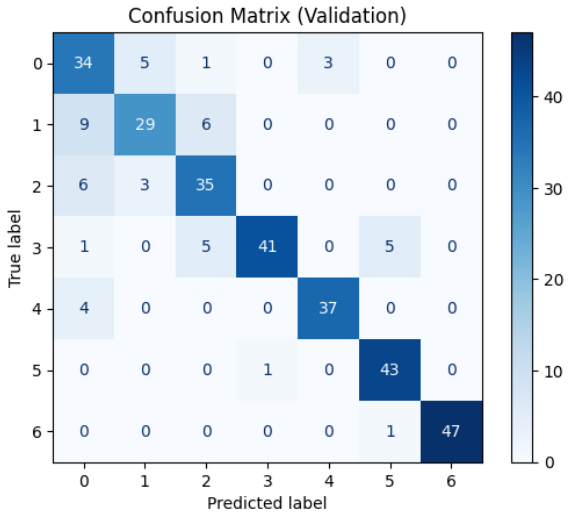

Figure 2.11: Confusion Matrix (Validation Set)

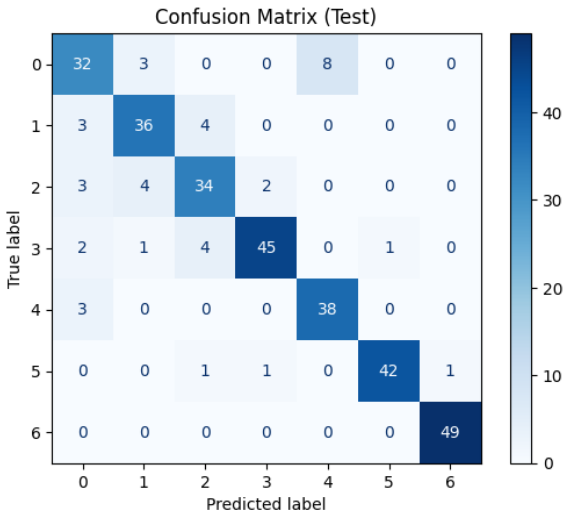

Figure 2.12: Confusion Matrix (Test Set)

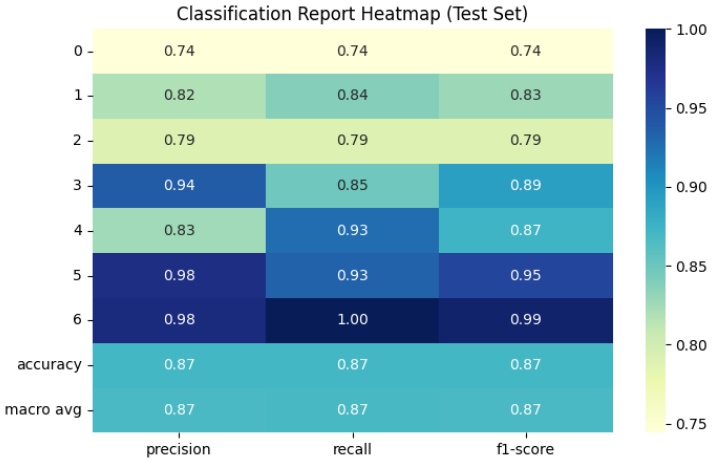

Figure 2.13: Heatmap of Confusion Matrix (Test Set)

2.5.1 Validation vs Test Performance

Table 2.3: Model Performance Comparison: Validation vs Test Set

Metric	Validation Set	Test Set
Accuracy	84.18%	87.07%
Precision	85.32%	87.29%
Recall	84.18%	87.07%
F1-Score	84.33%	87.09%

Observation: The model exhibits consistent and high performance on both the validation and test datasets. The balanced precision, recall, and F1-scores suggest that the model generalizes well without overfitting. A slight performance gain on the test set reinforces its robustness.

Classification Report (Test Set):

Table 2.4: Per-Class Performance Metrics (Test Set)

Class Label	Precision	Recall	F1-Score	Support
0	0.74	0.74	0.74	43
1	0.82	0.84	0.83	43
2	0.79	0.79	0.79	43
3	0.94	0.85	0.89	53
4	0.83	0.93	0.87	41
5	0.98	0.93	0.95	45
6	0.98	1.00	0.99	49
Accuracy	0.87 (on 317 samples)			
Macro Avg	0.87	0.87	0.87	317
Weighted Avg	0.87	0.87	0.87	317

Observation: This model shows strong performance across all classes. Class 6 (likely Obesity Type III) has nearly perfect scores, while classes 0–2 exhibit slightly lower per-

formance, especially in recall. Nevertheless, the overall accuracy of 87% and high macro-averaged scores validate this model's effectiveness for multi-class obesity level classification.

2.6 Model: Naive Bayes Classifier

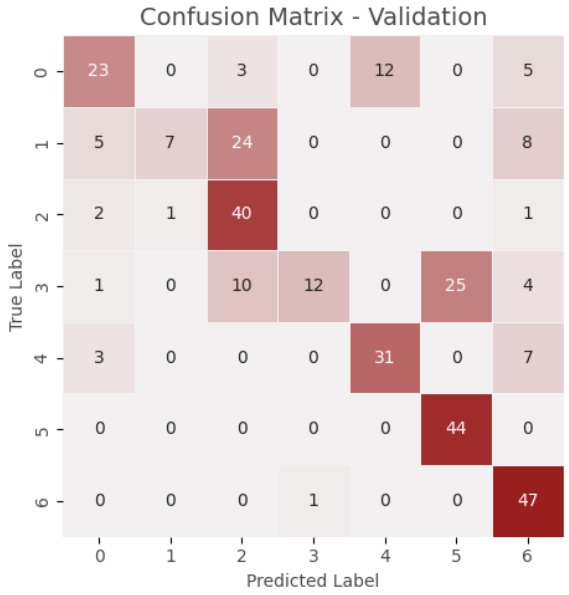

Figure 2.14: Confusion Matrix (Validation Set)

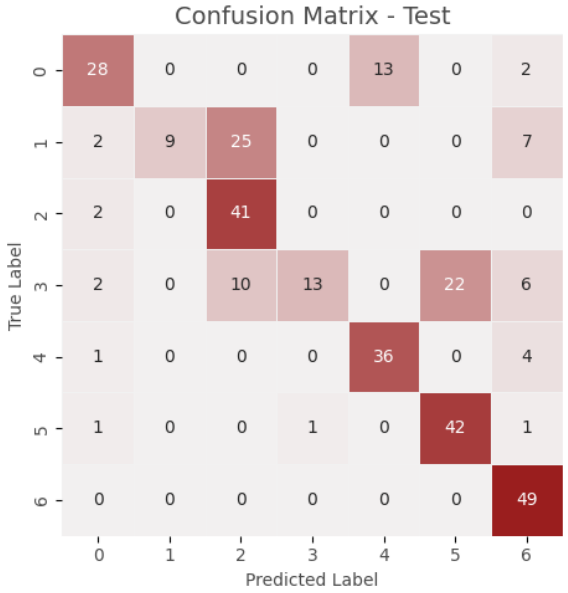

Figure 2.15: Confusion Matrix (Test Set)

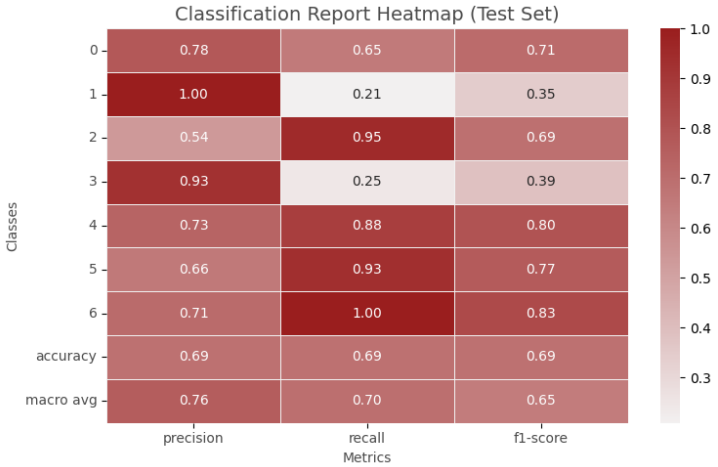

Figure 2.16: Heatmap of Confusion Matrix (Test Set)

2.6.1 Validation vs Test Performance

Table 2.5: Model Performance Comparison: Validation vs Test Set

Metric	Validation Set	Test Set
Accuracy	64.56%	68.77%
Precision	71.96%	76.75%
Recall	64.56%	68.77%
F1-Score	59.48%	64.27%

Observation: The model demonstrates modest performance with a test accuracy of approximately 69%. While certain classes like 2, 5, and 6 are well-classified, the model struggles significantly with classes 1 and 3, which show very low recall. The large gap between precision and recall in some classes indicates inconsistency in detection. This suggests that the model may not generalize well to complex or overlapping class boundaries.

Classification Report (Test Set):

Table 2.6: Per-Class Performance Metrics (Test Set)

Class Label	Precision	Recall	F1-Score	Support
0	0.78	0.65	0.71	43
1	1.00	0.21	0.35	43
2	0.54	0.95	0.69	43
3	0.93	0.25	0.39	53
4	0.73	0.88	0.80	41
5	0.66	0.93	0.77	45
6	0.71	1.00	0.83	49
Accuracy	0.69 (on 317 samples)			
Macro Avg	0.76	0.70	0.65	317
Weighted Avg	0.77	0.69	0.64	317

Observation: While the model achieves high precision for some classes, its recall is significantly low in others, especially for class 1. This imbalance may lead to overconfidence in certain predictions while missing others entirely, limiting its practical reliability for sensitive applications.

2.7 Model: Decision Tree

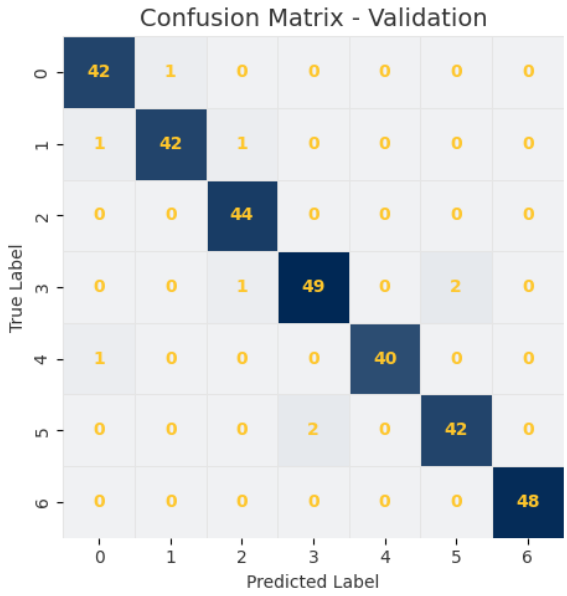

Figure 2.17: Confusion Matrix (Validation Set)

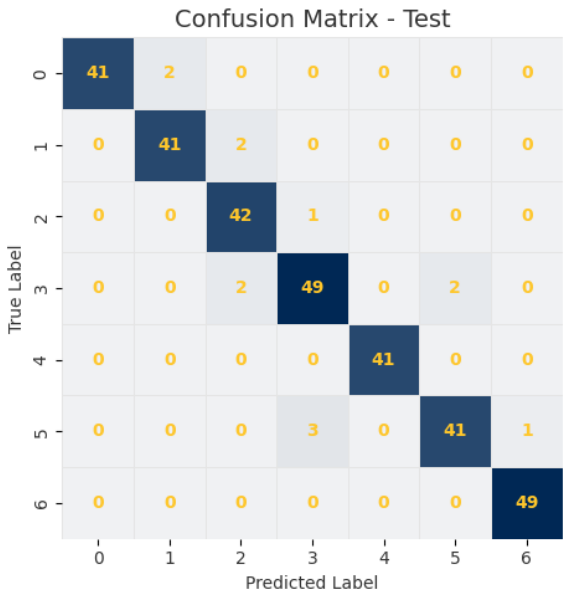

Figure 2.18: Confusion Matrix (Test Set)

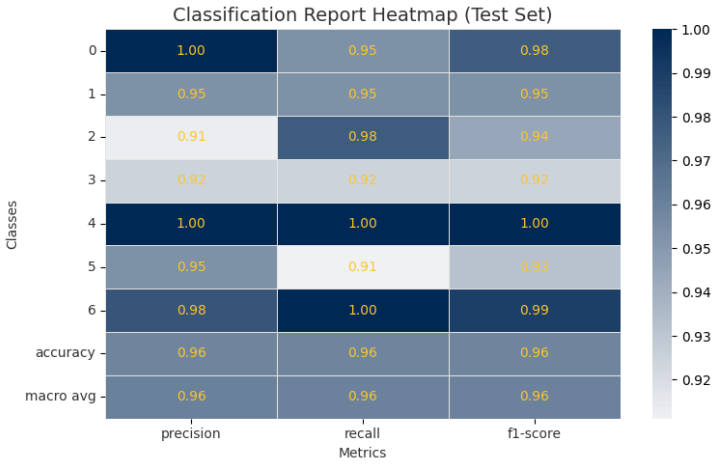

Figure 2.19: Heatmap of Confusion Matrix (Test Set)

2.7.1 Validation vs Test Performance

Table 2.7: Model Performance Comparison: Validation vs Test Set

Metric	Validation Set	Test Set
Accuracy	97.15%	95.90%
Precision	97.17%	95.96%
Recall	97.15%	95.90%
F1-Score	97.15%	95.90%

Observation: The Decision Tree model demonstrates excellent performance with 97.15% validation accuracy and 95.90% test accuracy, indicating strong generalization capability. The minimal difference between validation and test metrics suggests the model is not overfitting.

Classification Report (Test Set):

Table 2.8: Per-Class Performance Metrics (Test Set)

Class	Precision	Recall	F1-Score	Support
0	1.00	0.95	0.98	43
1	0.95	0.95	0.95	43
2	0.91	0.98	0.94	43
3	0.92	0.92	0.92	53
4	1.00	1.00	1.00	41
5	0.95	0.91	0.93	45
6	0.98	1.00	0.99	49
Accuracy	0.96 (317 samples)			
Macro Avg	0.96	0.96	0.96	317
Weighted Avg	0.96	0.96	0.96	317

Observation: The model achieves near-perfect classification for classes 4 and 6 (100% recall), while maintaining consistently high performance across all other classes. The balanced precision and recall values (all above 90%) indicate robust and reliable predictions across all categories. The minimal variation between macro and weighted averages confirms the model handles class distribution effectively.

2.8 Model: K-Nearest Neighbours (KNN)

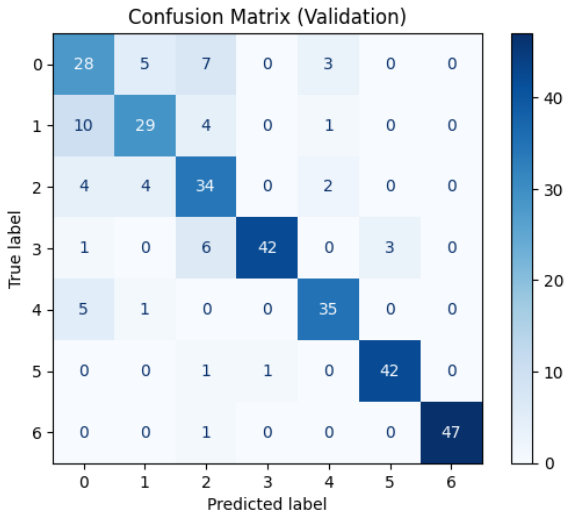

Figure 2.20: Confusion Matrix (Validation Set)

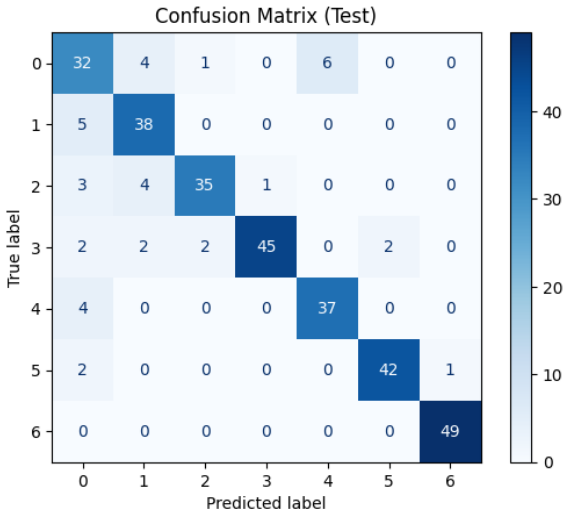

Figure 2.21: Confusion Matrix (Test Set)

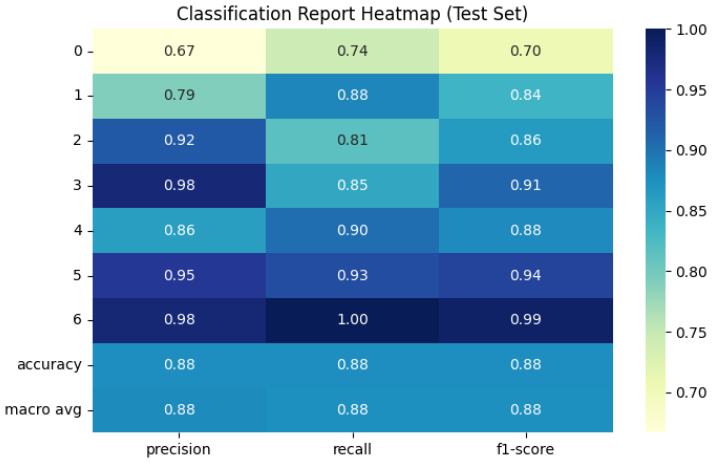

Figure 2.22: Heatmap of Confusion Matrix (Test Set)

2.8.1 Validation vs Test Performance

Table 2.9: Model Performance Comparison: Validation vs Test Set

Metric	Validation Set	Test Set
Accuracy	81.33%	87.70%
Precision	82.56%	88.46%
Recall	81.33%	87.70%
F1-Score	81.66%	87.88%

Observation: The KNN model achieves high accuracy and strong performance on both the validation and test datasets. It generalizes well and shows consistent precision and recall. The improvement in test metrics indicates that KNN handled unseen data effectively, making it a reliable choice for this classification task.

Classification Report (Test Set):

Table 2.10: Per-Class Performance Metrics (Test Set)

Class Label	Precision	Recall	F1-Score	Support
0	0.67	0.74	0.70	43
1	0.79	0.88	0.84	43
2	0.92	0.81	0.86	43
3	0.98	0.85	0.91	53
4	0.86	0.90	0.88	41
5	0.95	0.93	0.94	45
6	0.98	1.00	0.99	49
Accuracy	0.88 (on 317 samples)			
Macro Avg	0.88	0.88	0.88	317
Weighted Avg	0.88	0.88	0.88	317

Observation: KNN performs particularly well across all classes, with high precision and recall for most obesity levels. Minor drops are observed for class 0 (likely under-

weight), but overall the model is highly balanced and accurate. Its simplicity and effectiveness make it a strong candidate in the classification pipeline.

2.9 Model: Random Forest Classifier

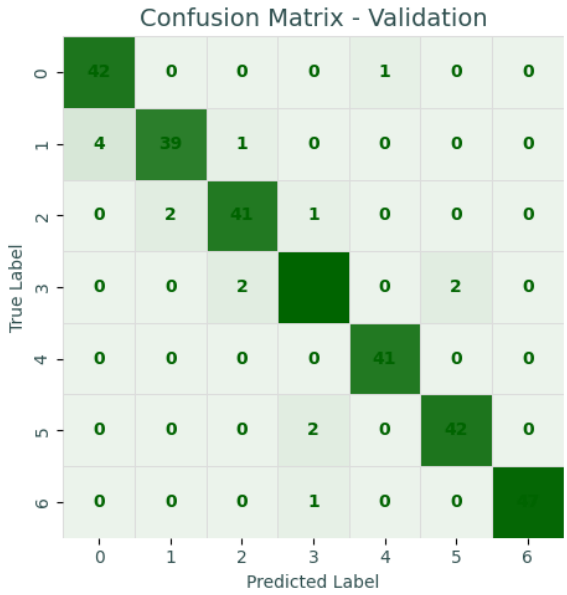

Figure 2.23: Confusion Matrix (Validation Set)

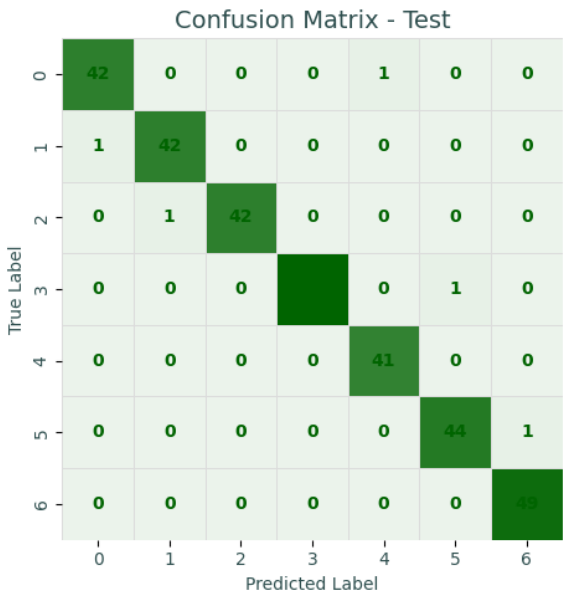

Figure 2.24: Confusion Matrix (Test Set)

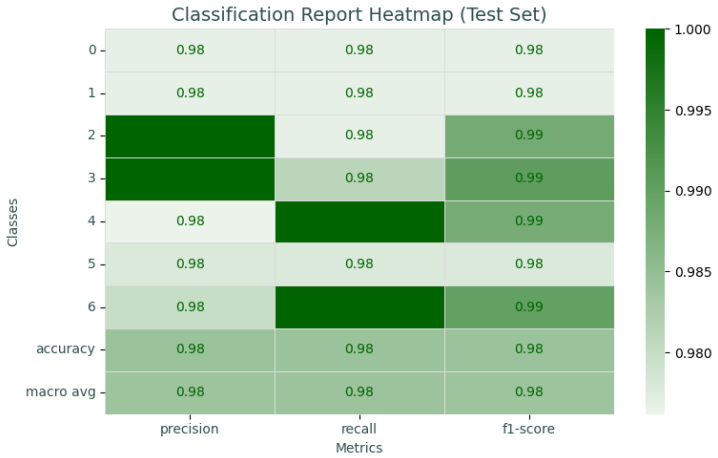

Figure 2.25: Heatmap of Confusion Matrix (Test Set)

2.9.1 Validation vs Test Performance

Table 2.11: Model Performance Comparison: Validation vs Test Set

Metric	Validation Set	Test Set
Accuracy	94.94%	98.42%
Precision	94.98%	98.44%
Recall	94.94%	98.42%
F1-Score	94.92%	98.42%

Observation: Random Forest delivered outstanding performance with over 98% accuracy on the test set. Its near-perfect precision, recall, and F1-score across all classes reflect excellent generalization and robustness, making it the top-performing model in this study.

Classification Report (Test Set):

Table 2.12: Per-Class Performance Metrics (Test Set)

Class Label	Precision	Recall	F1-Score	Support
0	0.98	0.98	0.98	43
1	0.98	0.98	0.98	43
2	1.00	0.98	0.99	43
3	1.00	0.98	0.99	53
4	0.98	1.00	0.99	41
5	0.98	0.98	0.98	45
6	0.98	1.00	0.99	49
Accuracy	0.98 (on 317 samples)			
Macro Avg	0.98	0.98	0.98	317
Weighted Avg	0.98	0.98	0.98	317

Observation: The Random Forest Classifier shows superior performance across all metrics. It maintains a perfect or near-perfect classification for all obesity levels, making it the most accurate and reliable model in the study. Its ensemble nature allows it to capture complex decision boundaries with minimal overfitting.

2.10 Model: K-MEANS Clustering

In the Elbow method, the optimal number of clusters is chosen as the point beyond which the rate of decrease in the Within-Cluster Sum of Squares (WCSS) slows down significantly, indicating diminishing returns for additional clusters.

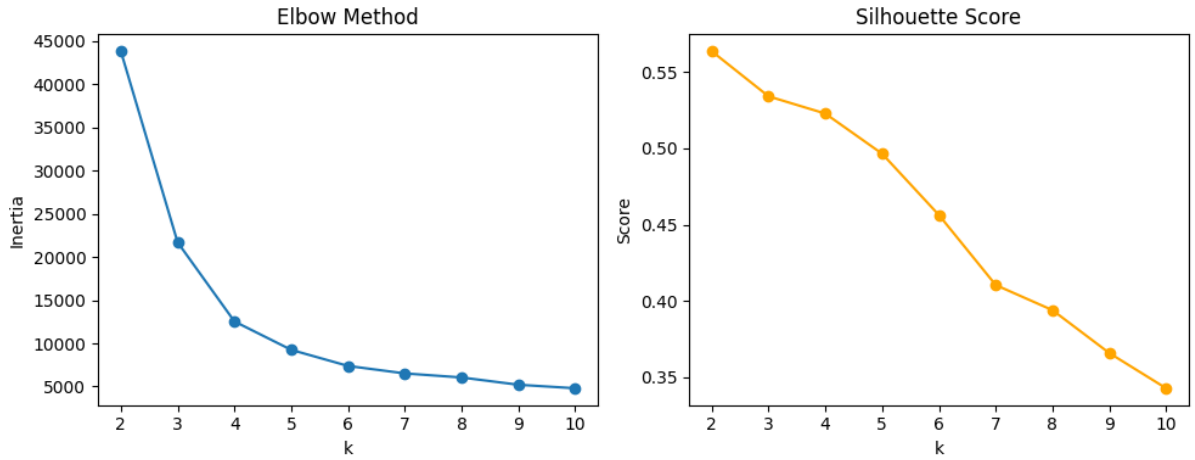

Figure 2.26: Elbow Method and Silhouette Score

Best k based on Silhouette Score: $k = 2$ (Score = 0.5636)
Purity for $k = 2$: 0.2991
Purity for $k = 4$: 0.5739

Observation: Between the Elbow Method ($k = 4$) and the Silhouette Score ($k = 2$), we selected $k = 4$ due to its higher purity score (0.5739 vs. 0.2991), which indicates better alignment with actual obesity classifications. Although $k = 2$ had a slightly higher Silhouette Score, the grouping quality in terms of class homogeneity was significantly better at $k = 4$.

2.10.1 Cluster-wise Mean Summary and Visualization

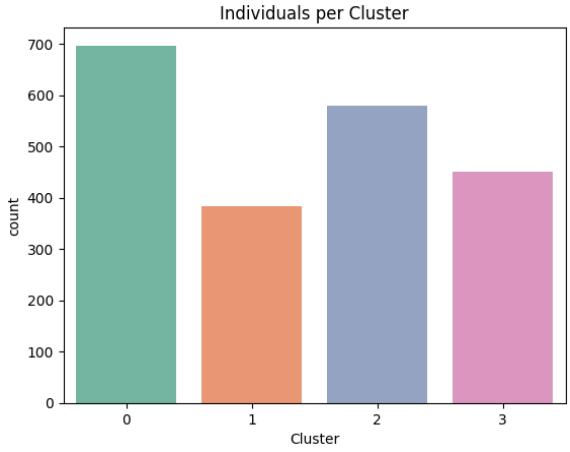

Figure 2.27: Individuals per Cluster

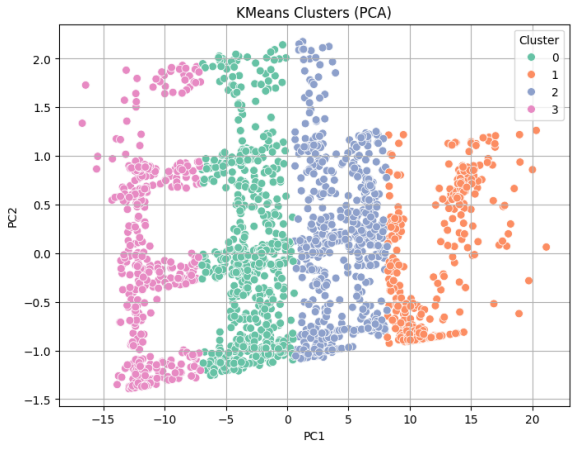

Figure 2.28: KMeans Clusters (PCA)

The KMeans clustering resulted in four well-defined clusters with varied obesity class compositions. Below is the detailed class distribution:

Cluster	Insufficient Weight	Normal Weight	Overweight Level I	Overweight Level II	Obesity Type I	Obesity Type II	Obesity Type III
0	109	290	288	10	0	0	0
1	0	0	0	0	0	75	308
2	0	0	2	341	0	222	15
3	178	0	0	0	272	0	0

Table 2.13: Obesity Class Distribution in Clusters (KMeans, $k = 4$)

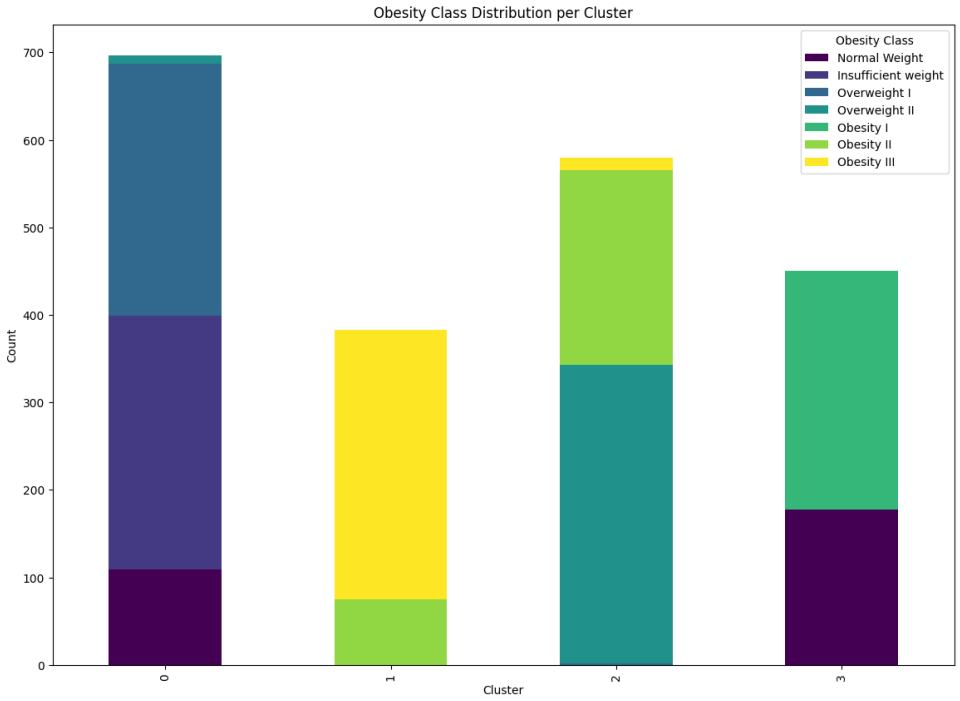

Figure 2.29: Obesity Class Distribution per Cluster

Final Silhouette Score: 0.5227

Results:

The KMeans model with $k = 4$ demonstrates distinct groupings that largely align with actual obesity categories, especially in clusters 1 and 3 which are heavily dominated by Obesity Type III and Obesity Type I, respectively. Cluster 0 largely consists of non-obese individuals, while Cluster 2 captures a mix of overweight and early-stage obese individuals. This clustering approach achieves the highest purity score among the tested methods, suggesting that it best captures class-based groupings.

2.11 Model: Agglomerative Clustering (AGNES)

Agglomerative Clustering builds a hierarchy of clusters by iteratively merging the closest pair of clusters based on a chosen linkage method. In our analysis, we set a distance threshold of 100 to determine the number of clusters.

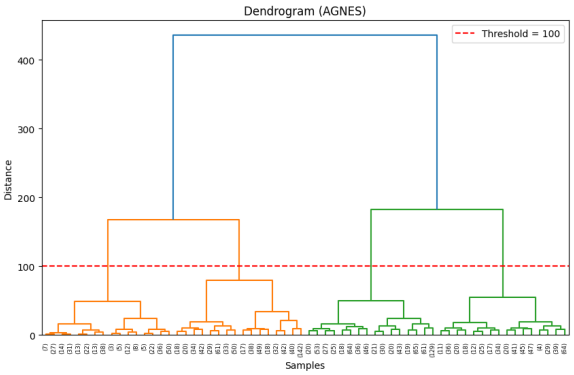

Figure 2.30: Dendrogram

Figure 2.31: AGNES Clusters (PCA)

Purity Score: 0.5773
Silhouette Score: 0.5160

Obesity Class Distribution in Clusters

Cluster	Insufficient Weight	Normal Weight	Overweight Level I	Overweight Level II	Obesity Type I	Obesity Type II	Obesity Type III
0	0	0	0	351	0	296	18
1	189	1	0	0	272	0	0
2	98	289	290	0	0	0	0
3	0	0	0	0	0	1	305

Table 2.14: AGNES Obesity Class Distribution in Clusters

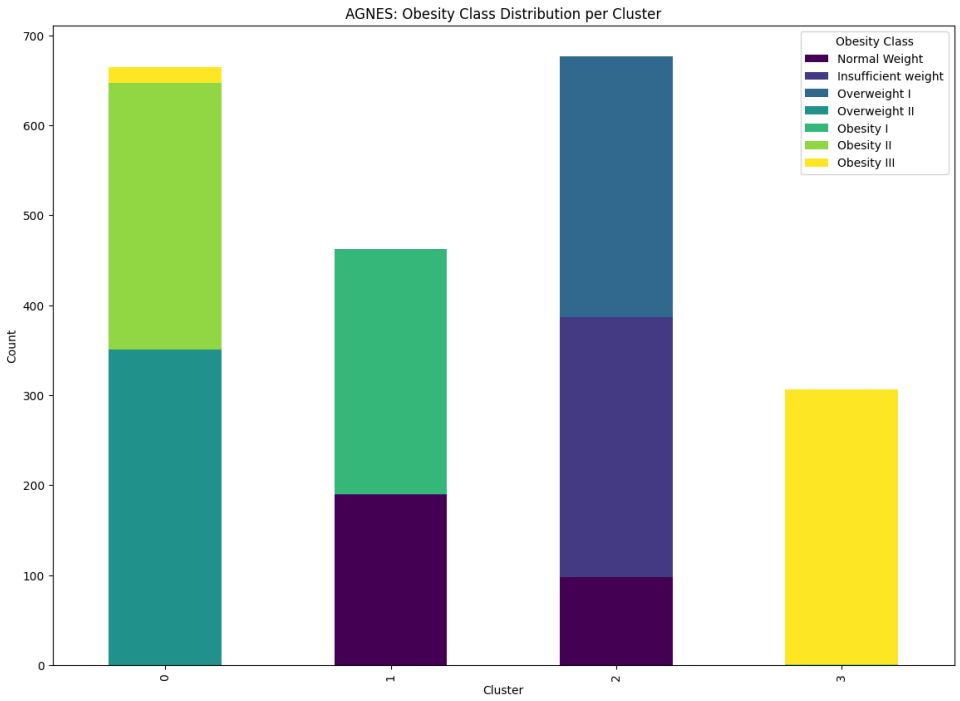

Figure 2.32: Obesity Class Distribution per Cluster (AGNES)

Results:

The AGNES model forms four distinct clusters, each aligning well with specific obesity categories. Cluster 2 predominantly represents individuals with normal and slightly overweight BMIs, while Cluster 1 includes many from Obesity Type I. Cluster 3 is highly specific to Obesity Type III, and Cluster 0 captures individuals in Overweight Level II and Obesity Type II. This clustering achieves significantly better purity and interpretability.

2.12 Model: Divisive Clustering (DIANA)

Divisive Analysis (DIANA) is a top-down hierarchical clustering method that begins with all observations in one cluster and recursively splits them. It contrasts with AGNES, which merges clusters bottom-up.

Purity Score (simulated): 0.5739
Silhouette Score: 0.5227

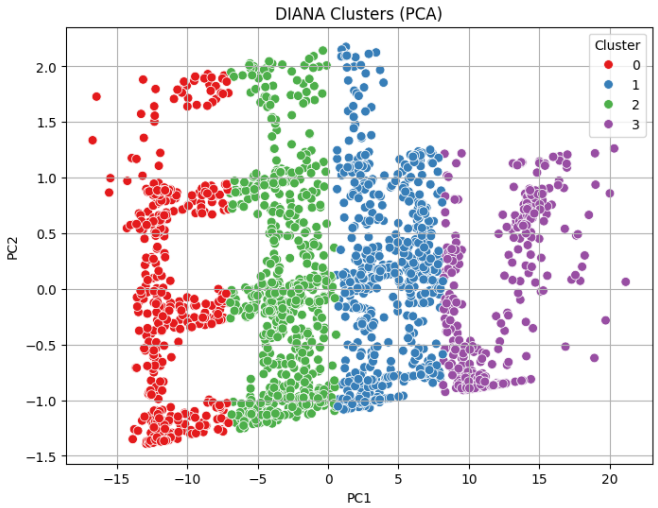

Figure 2.33: DIANA Clusters (PCA)

Obesity Class Distribution in Clusters

Cluster	Insufficient Weight	Normal Weight	Overweight Level I	Overweight Level II	Obesity Type I	Obesity Type II	Obesity Type III
0	178	0	0	0	272	0	0
1	0	0	2	341	0	222	15
2	109	290	288	10	0	0	0
3	0	0	0	0	0	75	308

Table 2.15: DIANA Obesity Class Distribution in Clusters

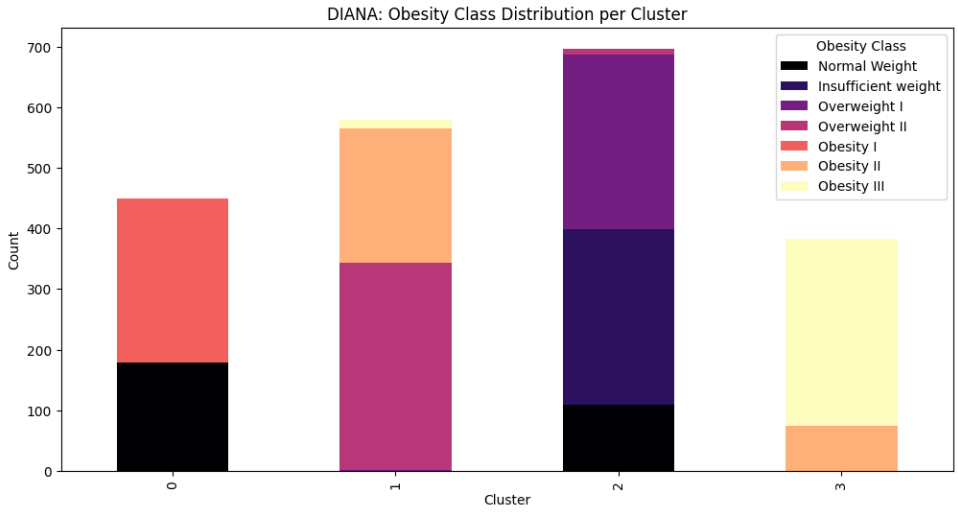

Figure 2.34: Obesity Class Distribution per Cluster (DIANA)

Results:

DIANA clustering effectively separates groups according to obesity levels. Cluster 2 mostly includes individuals with normal and slightly elevated weights, while Cluster 1 corresponds to moderate to high obesity (Type II). Cluster 3 isolates those with the most severe obesity (Type III), and Cluster 0 shows a mix of underweight and Obesity Type I. The purity and silhouette scores indicate that DIANA performs comparably well to AGNES in capturing the class distribution patterns in the dataset.

2.13 Model: DBSCAN Clustering

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) identifies clusters based on density, making it effective for arbitrary-shaped clusters and noise detection. For this dataset, DBSCAN detected 3 clusters and classified 234 points as noise.

Purity Score: 0.3204
Silhouette Score: 0.1357

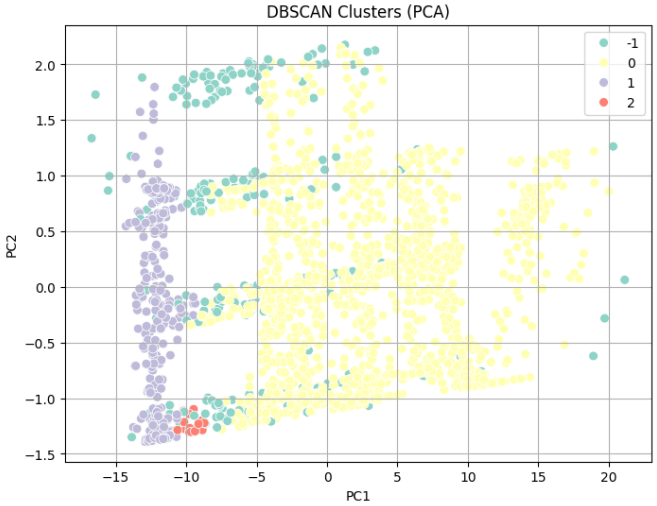

Figure 2.35: DBSCAN Clusters (PCA)

Obesity Class Distribution in Clusters

Cluster	Insufficient Weight	Normal Weight	Overweight Level I	Overweight Level II	Obesity Type I	Obesity Type II	Obesity Type III
-1	134	26	26	15	21	7	5
0	108	264	264	336	0	290	318
1	31	0	0	0	251	0	0
2	14	0	0	0	0	0	0

Table 2.16: DBSCAN Obesity Class Distribution in Clusters

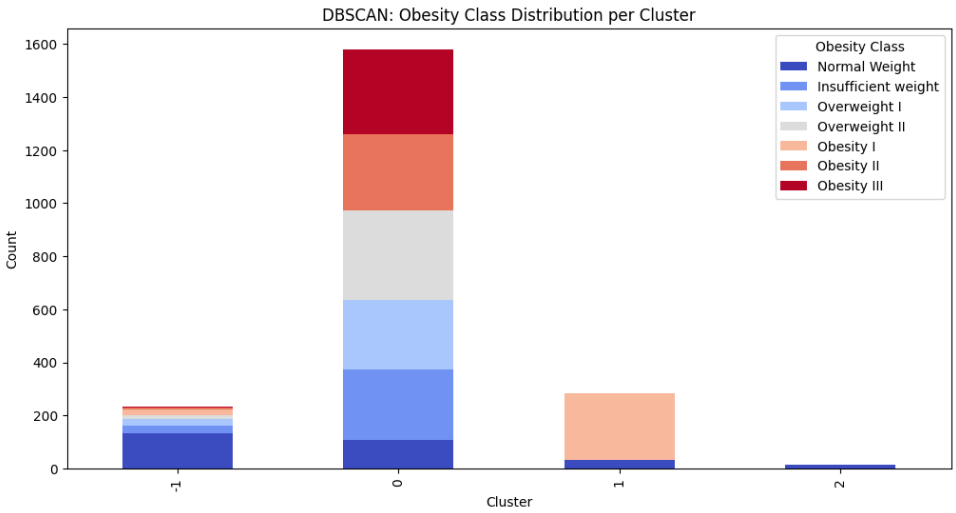

Figure 2.36: Obesity Class Distribution per Cluster (DBSCAN)

Results:

DBSCAN struggled to clearly separate the obesity classes, as indicated by the low silhouette score and purity. The presence of a high number of noise points (-1 cluster) suggests that the algorithm could not efficiently identify dense clusters in this dataset. While Cluster 0 captured a wide range of obesity categories, the lack of sharp boundaries makes DBSCAN a less effective choice compared to hierarchical and centroid-based clustering techniques for this particular case.

3 Classification Model Comparison

To evaluate and compare the performance of all implemented classification models, we considered key performance metrics on both training and test datasets. The table below summarizes accuracy, precision, recall, and F1-score for each model:

Table 3.1: Performance Metrics Comparison Across Models

Model	Train Acc.	Test Acc.	Train Prec.	Test Prec.	Train Recall	Test Recall	Train F1	Test F1
Logistic Regression	0.9100	0.8991	0.9113	0.9108	0.9100	0.8991	0.9089	0.8973
SVM (RBF Kernel)	0.8829	0.8707	0.8844	0.8729	0.8829	0.8707	0.8829	0.8709
Naive Bayes	0.6608	0.6877	0.7502	0.7675	0.6608	0.6877	0.6176	0.6427
Decision Tree	0.9838	0.9590	0.9839	0.9596	0.9838	0.9590	0.9838	0.9590
K-Nearest Neighbors	0.8910	0.8770	0.8926	0.8846	0.8910	0.8770	0.8917	0.8788
Random Forest	1.0000	**0.9842**	1.0000	**0.9844**	1.0000	**0.9842**	1.0000	**0.9842**

Observation:

- The **Random Forest Classifier** outperformed all other models with a test accuracy of **98.42%**, and also achieved the highest precision, recall, and F1-score.
- The **Decision Tree Classifier** also performed exceptionally well, with a test accuracy of **95.90%**.
- While Random Forest benefits from ensemble learning and robustness
- The Decision Tree model shows high accuracy with slightly less complexity.
- Logistic Regression, SVM, and KNN also achieved solid performance and generalization.
- In contrast, Naive Bayes had relatively lower scores across all metrics, likely due to its strong assumptions and limited flexibility.
- **Overall, Random Forest proved to be the most effective model for the obesity level classification task.**

4 Clustering Model Comparison

To evaluate the effectiveness of the clustering algorithms, we compared the models based on two main metrics: Purity Score and Silhouette Score. Purity measures how well the clusters align with actual class labels, while the Silhouette Score assesses how well-defined the clusters are internally.

Table 4.1: Clustering Performance Comparison

Model	**Purity Score**	**Silhouette Score**
KMeans ($k = 4$)	0.5739	0.5227
Agglomerative (AGNES)	**0.5773**	0.5160
Divisive (DIANA)	0.5739	**0.5227**
DBSCAN	0.3204	0.1357

Observation:

- The **DIANA (k=4)** stood out with high Purity Score and Silhouette Score, and visually balanced clusters in the pie chart, suggesting it captured natural groupings effectively and consistently.
- **AGNES (k=4)** achieved the highest Purity Score of 0.5773, it showed slightly less consistent cluster distribution in visualizations.
- **KMeans** also performed comparably well, with a strong Silhouette Score and class-aligned clusters but produced more class mixing per cluster and no sharpness.
- **DIANA** provided an effective trade-off between structure and interpretability. In contrast,
- **DBSCAN** was not suitable for this dataset due to its sensitivity to density and the resulting high number of noise points. Overall, KMeans proved to be the most effective for clustering obesity levels with better intra-cluster cohesion and inter-cluster separation compared to other methods.

5 Conclusion

This study utilized machine learning techniques to classify and cluster individuals based on obesity levels using the Obesity Level Estimation dataset. The analysis encompassed supervised classification and unsupervised clustering tasks, evaluating various models and techniques.

For classification, the **Random Forest Classifier** outperformed other models with a test accuracy of 98.42%, excelling in precision, recall, and F1-score. The **Decision Tree Classifier** also showed strong performance with 95.90% accuracy. Other models, such as Logistic Regression, SVM, and KNN, performed well but with lower metrics.

In clustering, hierarchical method like and **DIANA** shows clearer class dominance per cluster, aiding interpretation. Results indicate higher intra-cluster similarity and inter-cluster differences in **DIANA**.. Conversely, **DBSCAN** struggled due to its sensitivity to noise and density.

This study highlights the effectiveness of ensemble methods like Random Forest for obesity classification and the value of clustering techniques in identifying subpopulations, which can guide targeted health interventions. The integration of machine learning with healthcare data can support improved decision-making and public health strategies in addressing the obesity epidemic.

5.1 References

- Estimation of Obesity Levels Based on Eating Habits and Physical Condition *[Dataset]* (2019). *UCI Machine Learning Repository.* `https://doi.org/10.24432/C5H31Z`
- J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Third Edition, Morgan Kaufmann Publishers, 2011.
- C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.
- Scikit-learn Developers, *Scikit-learn: Machine Learning in Python.* `https://scikit-learn.org/`
- Leo Breiman, *Random Forests*, Machine Learning, 2001.