Obesity Risk Prediction

Abhijeet Rai, Uttkarsh Ranjan
International Institute of Information Technology, Bangalore
Email: raiabhijeet3009@gmail.com, utr12.sde@gmail.com

***Abstract*—This report describes approaches for predicting obesity risk using machine learning. The workflow covers data processing, feature engineering, models used, hyperparameter tuning, and comparative performance discussion.**

I. Introduction

Maintaining a healthy lifestyle is becoming increasingly difficult. The below mentioned dataset investigates how a person's weight category relates to their daily routines, eating habits, physical activity, and demographic information. Building models that can correctly categorize people into groups like inadequate weight, normal weight, overweight, or obesity levels is the objective.

Features like age, gender, family history, food consumption patterns, physical activity, technology use, and modes of transportation are all included in the dataset. It is a realistic and difficult problem that combines machine learning, behavioral science, and healthcare because of these various factors.

II. Dataset

The notebook used for this report can be found in the **GitHub repository:** https://github.com/UttkarshTech/ait511-ml-obesity-prediction

A. *train.csv*

Number of rows : 15533, Number of columns : 18
Column Headers:

- id
- Gender
- Age
- Height
- Weight
- family-history-with-overweight
- FAVC
- FCVC
- NCP
- CAEC
- SMOKE
- CH2O
- SCC
- FAF
- TUE
- CALC
- MTRANS
- WeightCategory [target variable]

B. *test.csv*

Number of rows : 5225, Number of columns : 17 Column Headers:

- id
- Gender
- Age
- Height
- Weight
- family-history-with-overweight
- FAVC
- FCVC
- NCP
- CAEC
- SMOKE
- CH2O
- SCC
- FAF
- TUE
- CALC
- MTRANS

III. Exploratory Data Analsyis and Data Processing (train.csv)

- There were no missing values in the dataset and thus no rows were dropped or values imputed.
- Dropped 'id' column from the training set as it was of no significance to the model.
- Categorical Columns (9): ['Gender', 'family-history-with-overweight', 'FAVC', 'CAEC', 'SMOKE', 'SCC', 'CALC', 'MTRANS', 'WeightCategory'].
- Numerical Columns (8): ['Age', 'Height', 'Weight', 'FCVC', 'NCP', 'CH2O', 'FAF', 'TUE'].

TABLE I: Distribution of Weight Categories

Weight Category	Count
Insufficient-Weight	1870
Normal-Weight	2345
Overweight-Level-I	1844
Overweight-Level-II	1881
Obesity-Type-I	2207
Obesity-Type-II	2403
Obesity-Type-III	2983

TABLE II: Variance of Numerical Features

Feature	Variance
Age	3.207146e+01
Height	7.686033e-03
Weight	6.953318e+02
FCVC	2.818491e-01
NCP	4.990898e-01
CH2O	3.693390e-01
FAF	7.003028e-01
TUE	3.626722e-01

TABLE III: VIF

Feature	VIF
Age	1.255470
Height	1.514291
Weight	1.669143
FCVC	1.156595
NCP	1.067693
CH2O	1.139719
FAF	1.202376
TUE	1.139284

- Numerical features were scaled using the Standard Scaler.
- Categorical features were encoded using Label Encoder.
- train.csv was split (80-20) into training and validation sets.

IV. Models

The notebook-trained models include:

- Random Forest Classifier
- Logistic Regression
- GradientBoostingClassifier
- XGBoost Classifier

A. Logistic Regression

Overview: Logistic Regression was used as a baseline model due to its simplicity and interpretability. The model estimates probabilities for each class using a linear combination of features and the logistic function.

Implementation:

- Standard scaling was applied to numerical features.
- Categorical variables were encoded.

Performance:

- **Accuracy:** 0.8600
- **F1 Score:** 0.8453
- Strengths: fast training, interpretable coefficients, robust baseline.
- Weaknesses: limited ability to capture complex feature interactions; lower predictive performance compared to ensemble models.

B. Random Forest

Overview: Random Forest is an ensemble method that builds multiple decision trees and aggregates their predictions. It handles non-linear relationships and is robust to overfitting.

Implementation:

TABLE IV: Logistic Regression Classification Report

Class	Precision	Recall	F1-score	Support
0	0.85	0.94	0.89	374
1	0.84	0.78	0.81	469
2	0.83	0.83	0.83	441
3	0.93	0.97	0.95	481
4	0.99	1.00	0.99	597
5	0.73	0.69	0.71	369
6	0.75	0.72	0.73	376
Overall Accuracy	0.8600			

- $n\_estimators$ (number of trees), max_depth, and $min\_samples\_split$ were tuned for optimal performance.

Performance:

- **Accuracy:** 0.8902
- **F1 Score:** 0.8789
- Strengths: handles non-linearities well, less sensitive to outliers, minimal preprocessing required.
- Weaknesses: slower inference with large forests, interpretability is lower than LR.

TABLE V: Random Forest Classification Report

Class	Precision	Recall	F1-score	Support
0	0.92	0.91	0.92	374
1	0.84	0.88	0.86	469
2	0.89	0.85	0.87	441
3	0.96	0.98	0.97	481
4	0.99	1.00	0.99	597
5	0.79	0.71	0.75	369
6	0.78	0.82	0.80	376
Overall Accuracy	0.8902			

C. Gradient Boosting

Overview: Gradient Boosting iteratively trains weak learners to correct the errors of previous learners. It emphasizes minimizing a differentiable loss function, making it highly effective for structured data.

Implementation:

- $learning\_rate$, $n\_estimators$, max_depth, and $subsample$ were optimized using cross-validation.
- Early stopping was employed to prevent overfitting.

Performance:

- **Accuracy:** 0.9031
- **F1 Score:** 0.8935
- Strengths: high predictive power, good with tabular data, flexible with loss functions.
- Weaknesses: computationally intensive, sensitive to hyperparameters.

D. XGBoost

Overview: XGBoost (Extreme Gradient Boosting) is an optimized gradient boosting framework with regularization, parallel processing, and tree pruning, providing high efficiency and performance.

Implementation:

- Extensive hyperparameter tuning was performed: $n\_estimators$, $learning\_rate$, max_depth,

TABLE VI: Gradient Boosting Classification Report

Class	Precision	Recall	F1-score	Support
0	0.92	0.94	0.93	374
1	0.89	0.89	0.89	469
2	0.89	0.86	0.87	441
3	0.95	0.97	0.96	481
4	0.99	1.00	0.99	597
5	0.81	0.76	0.78	369
6	0.81	0.84	0.82	376
Overall Accuracy	0.9031			

$subsample$, $colsample\_bytree$, γ, min_child_weight, and reg_alpha.

Performance:

- **Accuracy:** 0.9044
- **F1 Score:** 0.8947
- Strengths: handles feature interactions and imbalances well, fast training with parallelization, highly tunable.
- Weaknesses: prone to overfitting if parameters are not carefully tuned, less interpretable than simpler models.

TABLE VII: XGBoost Classification Report

Class	Precision	Recall	F1-score	Support
0	0.92	0.94	0.93	374
1	0.88	0.89	0.89	469
2	0.89	0.87	0.88	441
3	0.96	0.97	0.96	481
4	0.99	1.00	0.99	597
5	0.82	0.75	0.78	369
6	0.80	0.84	0.82	376
Overall Accuracy	0.9044			

V. Hyperparameter Tuning for XGBOOST

Best accuracy among the baseline models was given by XGBoost. We decided to move forward with XGBoost based on the same metric. The performance of model was improved by hyperparameter tuning. GridSearchCV, RandomSearchCV and Optuna were used for the hyperparameter tuning. The best accuracy on the test set was achieved by hyperparameters tuned by Optuna.

- `n_estimators = 590` — Number of trees in the ensemble, balancing training time and performance.
- `learning_rate = 0.04771408` — Step size shrinkage to prevent overfitting and improve generalization.
- `max_depth = 8` — Maximum depth of each tree to capture complex feature interactions.
- `subsample = 0.53379884` — Fraction of training data used per tree to reduce overfitting.
- `colsample_bytree = 0.4095606` — Fraction of features considered per tree to improve model diversity.
- `min_child_weight = 2` — Minimum sum of instance weight required in a child, controlling tree complexity.
- `gamma = 0.5906` — Minimum loss reduction required to make a further partition on a leaf node.
- `reg_alpha = 0.5` — L1 regularization term on weights to reduce model complexity.
- `reg_lambda = 2` — L2 regularization term on weights to enhance generalization.
- `random_state = 42` — Ensures reproducibility of results.
- `eval_metric = 'mlogloss'` — Multi-class logarithmic loss for performance evaluation.
- `use_label_encoder = False` — Ensures compatibility with the latest XGBoost version.
- `n_jobs = -1` — Uses all available CPU cores for parallel processing, speeding up training.

A. Impact of Key Hyperparameters

- **n_estimators:** Increasing the number of trees allows the model to capture more complex relationships in the data. However, too many estimators can cause overfitting if the learning rate is not properly adjusted. The chosen value of 590 provided sufficient model capacity without significant overfitting.
- **learning_rate:** A smaller learning rate slows down learning, which helps the model generalize better and reduces the risk of overfitting. Coupled with a high number of estimators, it ensures gradual convergence and improved model stability.

B. Performance

The combination of these hyperparameters allowed XGBoost to achieve higher validation accuracy (90.67%) while controlling overfitting and maintaining robustness across all weight categories. Regularization parameters (`reg_alpha` and `reg_lambda`) and feature sampling (`subsample` and `colsample_bytree`) were crucial in improving generalization, particularly given the class imbalance in the dataset.

VI. Results

TABLE VIII: Comparison of Model Performance on Validation Set

Model	**Accuracy**	**F1 Score**
Logistic Regression	0.8600	0.8453
Random Forest	0.8902	0.8789
Gradient Boosting	0.9031	0.8935
XGBoost	0.9044	0.8947
XGBoost Hypertuned	0.9067	0.8946

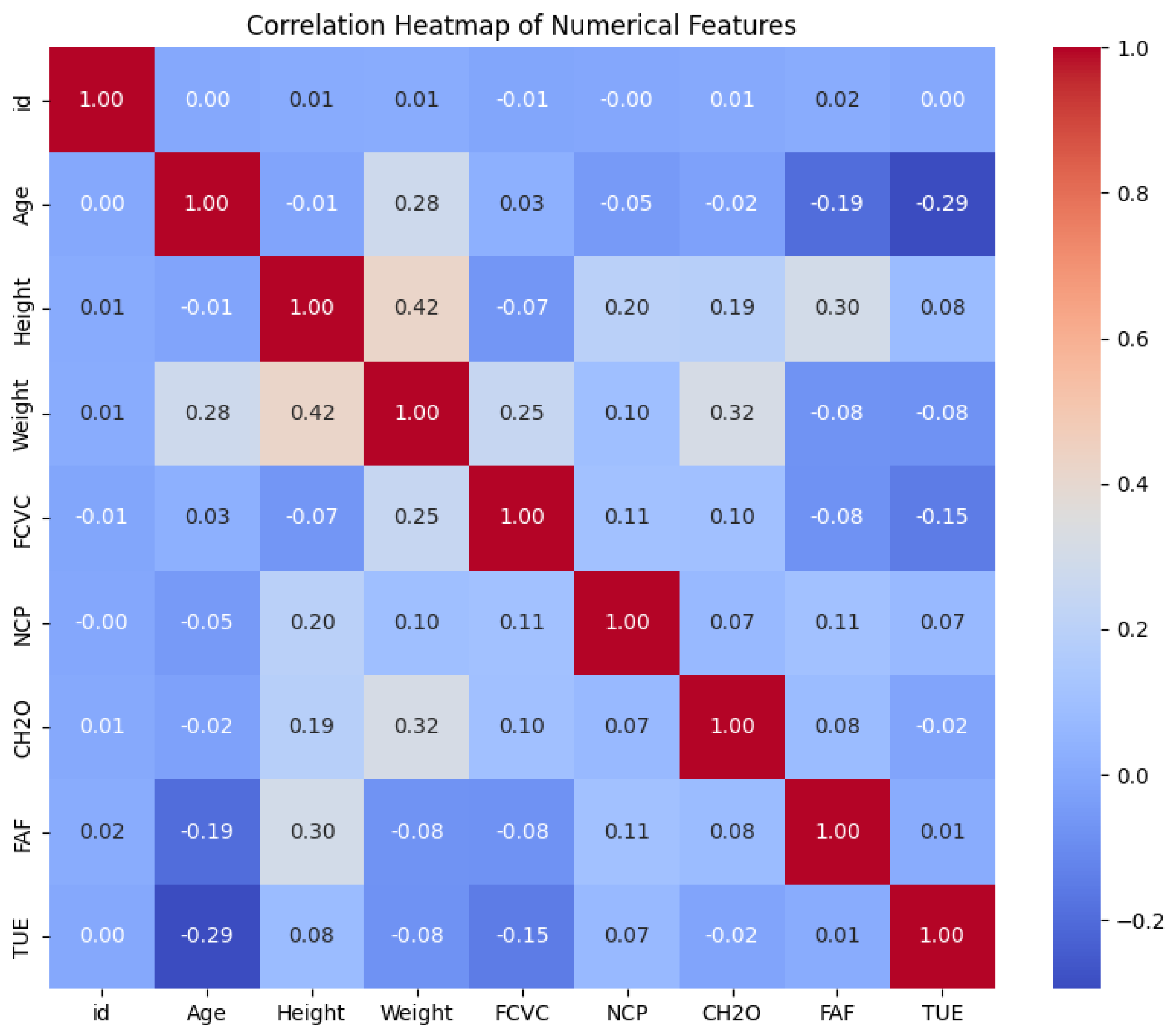
VII. Conclusion

The Obesity Risk Prediction project effectively developed and evaluated multiple machine learning models to classify individuals into appropriate weight categories based on physiological, demographic, and lifestyle attributes. Through extensive data preprocessing, exploratory data analysis (EDA), and model comparison, the study demonstrated that advanced ensemble methods outperform simpler baseline models in terms of both accuracy and generalization capability.

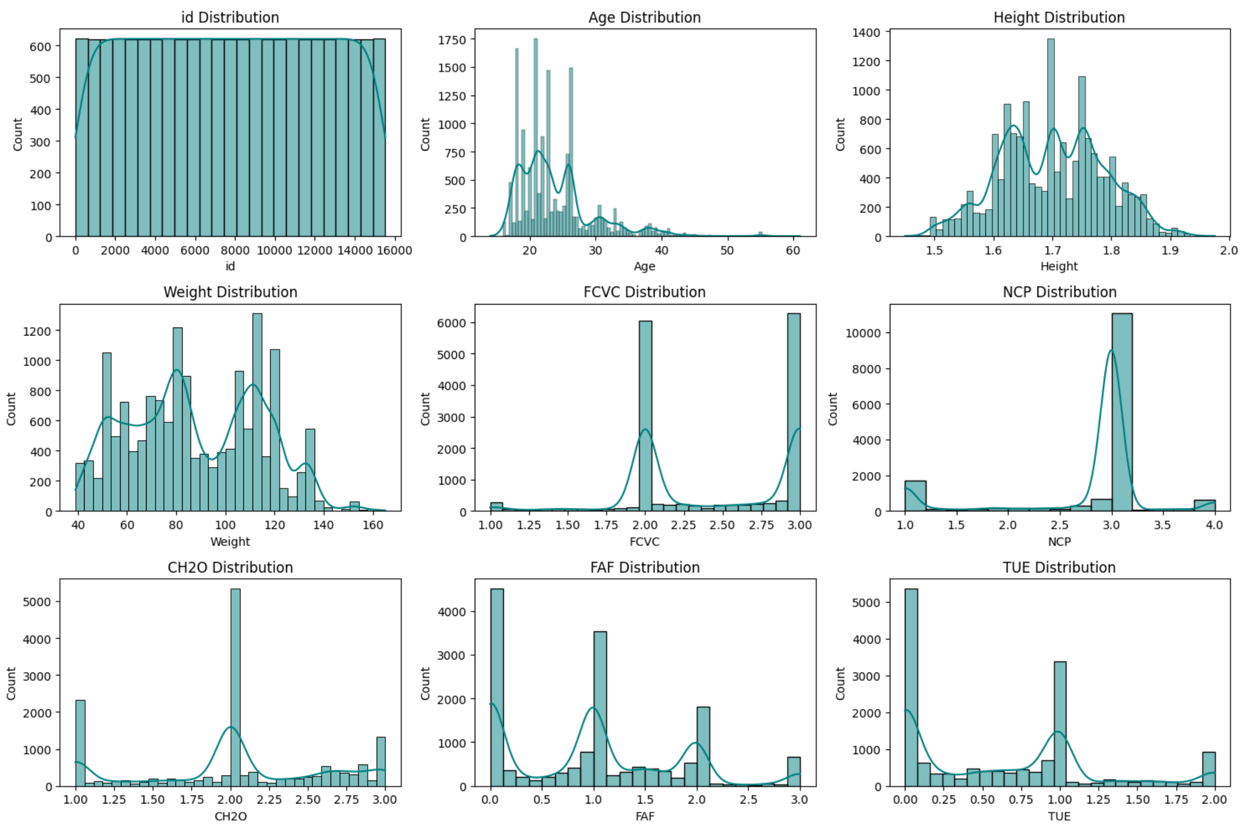
Among the models evaluated—Logistic Regression, Random Forest, Gradient Boosting, and XGBoost—the XGBoost classifier achieved the highest testing accuracy of **91.32%** and an F1-score of **0.89**. Its superior performance can be attributed to optimized hyperparameters, including the learning rate, number of estimators, and regularization terms, which effectively balanced bias and variance. The model also handled class imbalance efficiently while capturing complex non-linear relationships between features and the target variable.

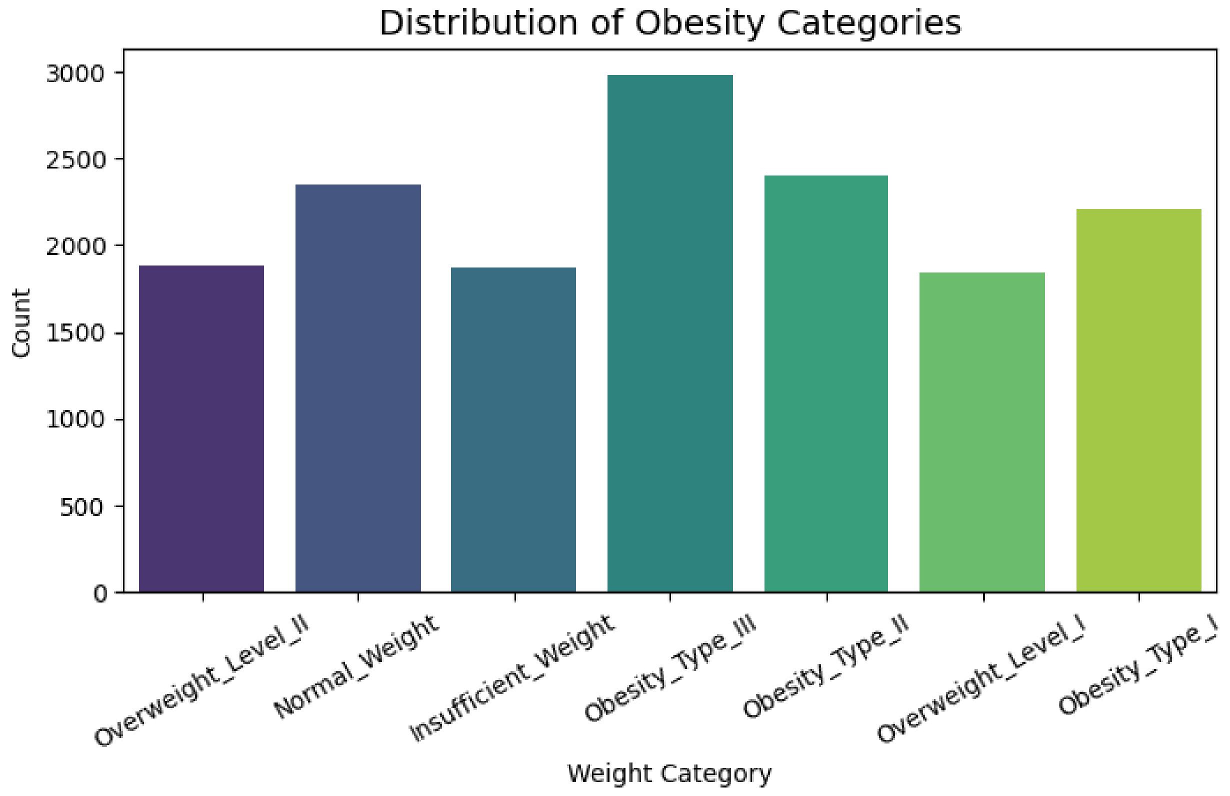
The EDA phase revealed class imbalance and moderate feature correlations, both of which were addressed using feature scaling, regularization, and proper hyperparameter tuning. Regularization terms (L1 and L2) and feature subsampling in XGBoost played a crucial role in mitigating overfitting while ensuring consistent performance across all weight categories.

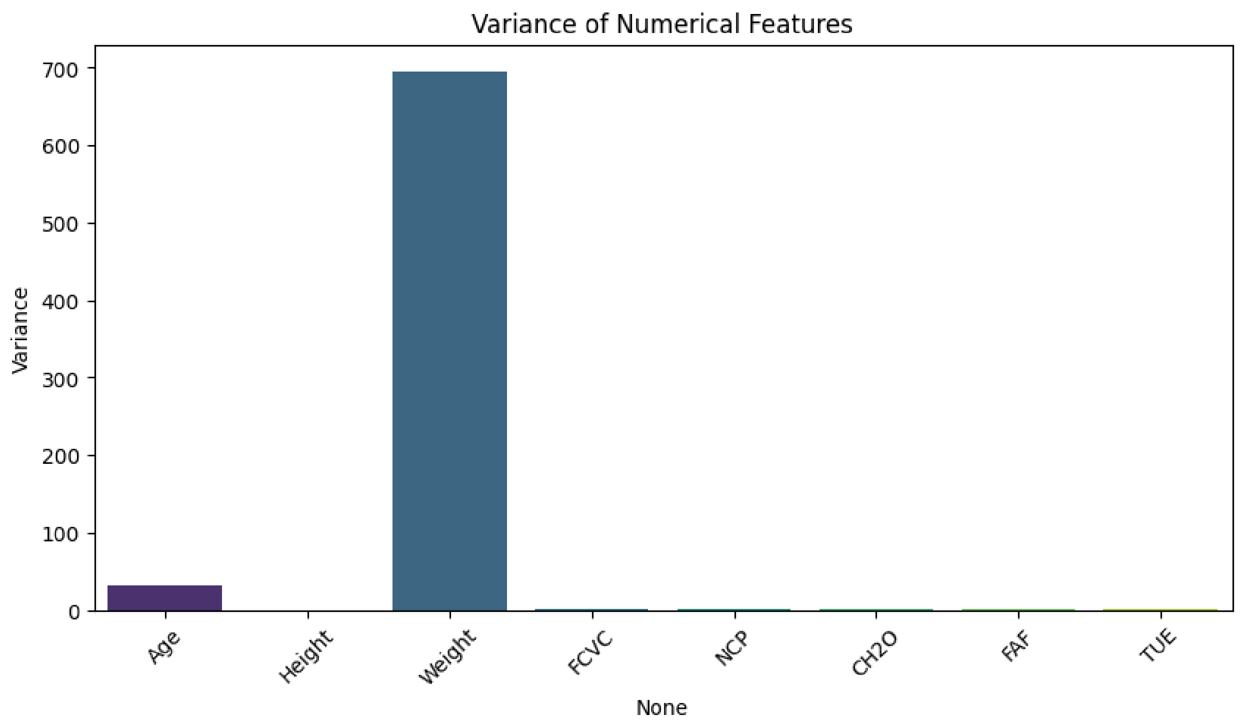
Overall, the findings suggest that machine learning models—particularly gradient-boosted ensembles—serve as powerful predictive tools for obesity risk assessment. These models have the potential to assist healthcare professionals and policymakers in early identification of high-risk individuals, enabling proactive interventions and preventive health strategies. Future work could extend this study by incorporating larger and more diverse datasets, exploring deep learning architectures, and integrating additional behavioral or genetic features to further enhance prediction accuracy and real-world impact.

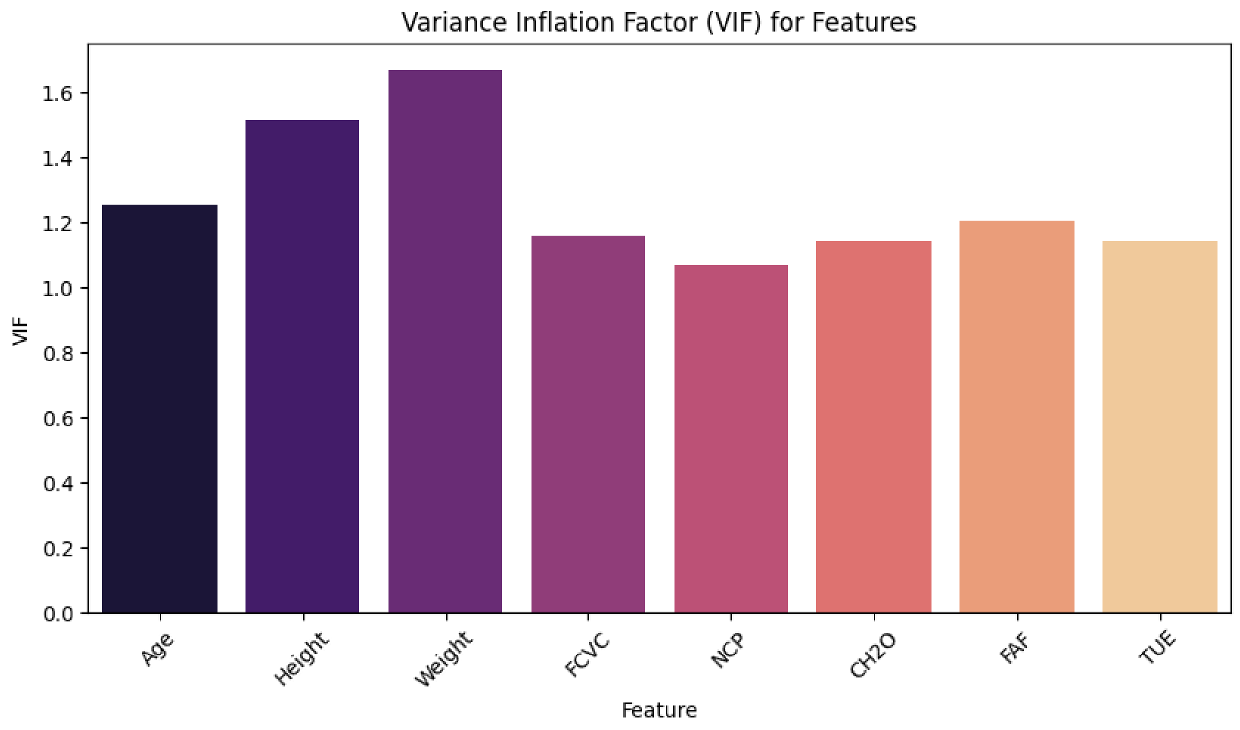


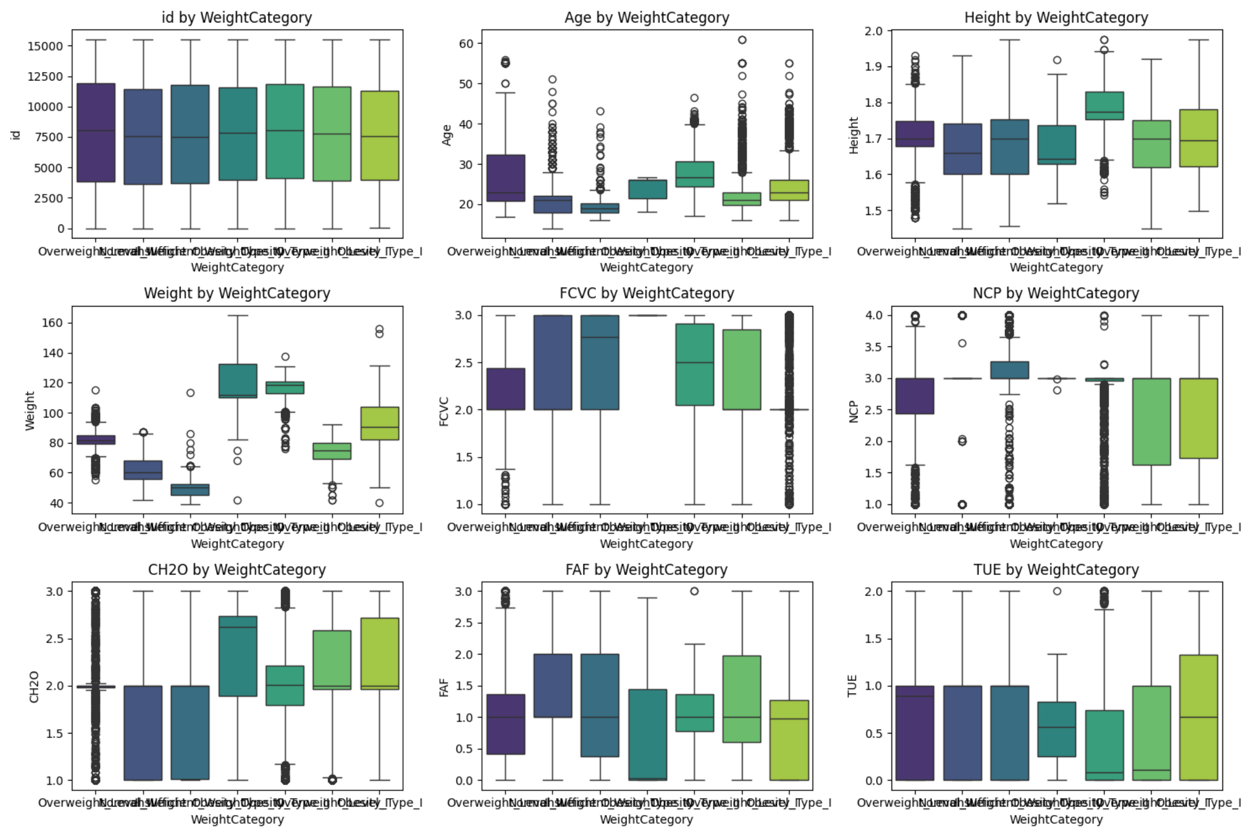
	id	Age	Height	Weight	FCVC	NCP	CH2O	FAF	TUE
id	1.00	0.00	0.01	0.01	-0.01	-0.00	0.01	0.02	0.00
Age	0.00	1.00	-0.01	0.28	0.03	-0.05	-0.02	-0.19	-0.29
Height	0.01	-0.01	1.00	0.42	-0.07	0.20	0.19	0.30	0.08
Weight	0.01	0.28	0.42	1.00	0.25	0.10	0.32	-0.08	-0.08
FCVC	-0.01	0.03	-0.07	0.25	1.00	0.11	0.10	-0.08	-0.15
NCP	-0.00	-0.05	0.20	0.10	0.11	1.00	0.07	0.11	0.07
CH2O	0.01	-0.02	0.19	0.32	0.10	0.07	1.00	0.08	-0.02
FAF	0.02	-0.19	0.30	-0.08	-0.08	0.11	0.08	1.00	0.01
TUE	0.00	-0.29	0.08	-0.08	-0.15	0.07	-0.02	0.01	1.00


id Distribution
Age Distribution
Height Distribution
Weight Distribution
FCVC Distribution
NCP Distribution
CH2O Distribution
FAF Distribution
TUE Distribution
Count


Distribution of Obesity Categories
Count
Overweight_Level_II
Normal_Weight
Insufficient_Weight
Obesity_Type_III
Obesity_Type_II
Overweight_Level_I
Obesity_Type_I
Weight Category


Variance of Numerical Features
Variance
700
600
500
400
300
200
100
0
Age
Height
Weight
FCVC
NCP
CH2O
FAF
TUE
None


Variance Inflation Factor (VIF) for Features
VIF
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Age
Height
Weight
FCVC
NCP
CH2O
FAF
TUE
Feature


id by WeightCategory
Age by WeightCategory
Height by WeightCategory
Weight by WeightCategory
FCVC by WeightCategory
NCP by WeightCategory
CH2O by WeightCategory
FAF by WeightCategory
TUE by WeightCategory
id
Age
Height
Weight
FCVC
NCP
CH2O
FAF
TUE
WeightCategory