Obesity Risk Prediction

AIT-511 Machine Learning Project
Sachin Mishra

October 26, 2025

Abstract

This report details the approach taken to solve a multi-class classification problem aimed at predicting an individual's WeightCategory using a dataset containing both continuous physiological measurements and categorical lifestyle habits. The project adheres to strict model usage constraints, focusing initially on foundational models (Decision Tree, KNN, Naive Bayes) before escalating to advanced ensembles. Through rigorous data processing and domain-driven feature engineering, a high baseline accuracy of **91.625%** was achieved using the XGBoost classifier.

Code: Visit Github File
https://github.com/Imsachin010/ait-511-ms2025013/

1 Introduction

The objective of this project is to develop and compare machine learning models capable of accurately classifying an individual's weight category into one of seven classes (e.g., Normal Weight, Overweight Level I, Obesity Type III). Given the complexity of the seven classes, the strategy relies heavily on robust data preprocessing, domain-specific feature engineering, and meticulous hyperparameter tuning across a predefined set of models.

1.1 Project Constraints and Model Set

The project utilized the following models and techniques, divided into phases to meet checkpoint requirements:

- **Checkpoint 1 Models:** Decision Tree (CART), K-Nearest Neighbors (KNN), Gaussian Naive Bayes, and XGBoost.
- **Best Models:** The XGBoost Classifier.
- **Evaluation Technique:** K-Fold Cross-Validation (CV) was mandatory for all model evaluations and hyperparameter tuning to ensure robustness.

2 Exploratory Data Analysis (EDA)

Comprehensive EDA confirmed the dataset's characteristics and guided the subsequent engineering steps.

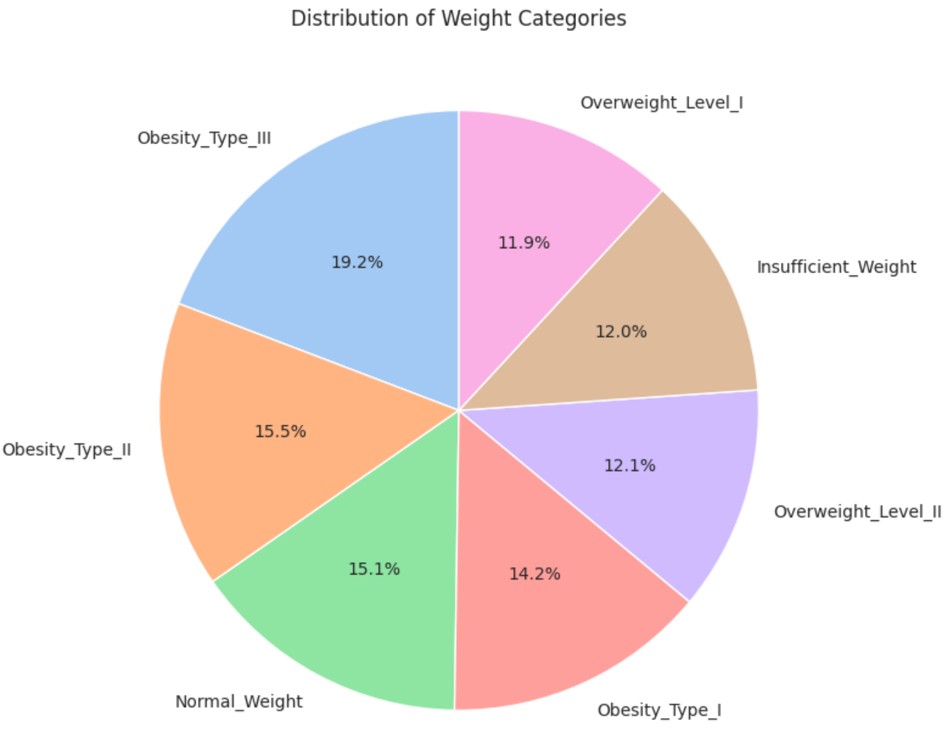


Figure 1: Distribution of people in each class.

2.1 Initial Data Overview

The training dataset comprised 15,533 samples across 16 features, the initial exploration of data is shown in Table.1 and Table.2. A critical initial finding was the **absence of missing values**, which greatly simplified the data cleaning phase. Furthermore, the seven target classes were found to be **relatively well-balanced** as shown in Figure.1, confirming that accuracy could be reliably used as the primary performance metric.

Table 1: Dataset statistics

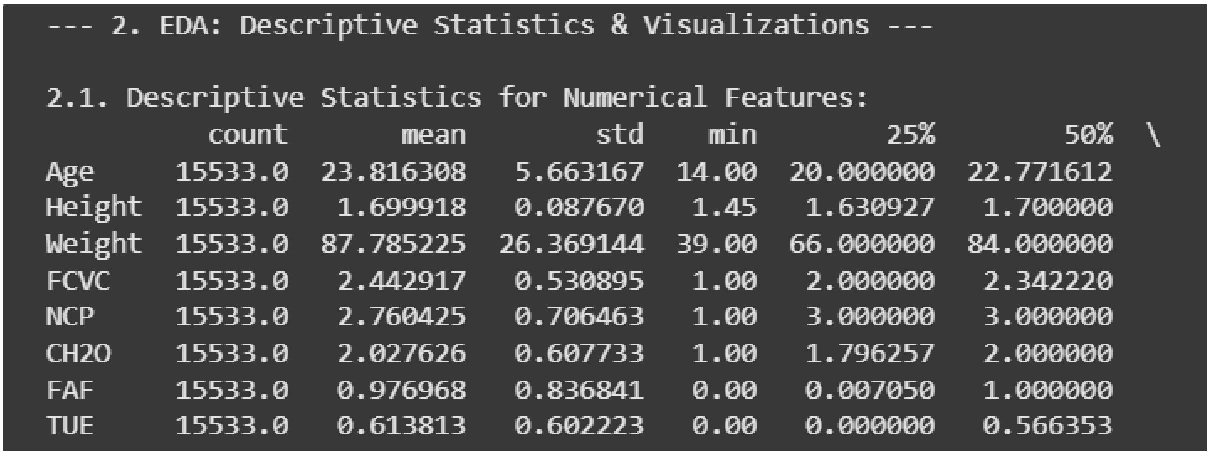
Statistic	**Value**
Number of variables	17
Number of observations	15533
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	2.0 MiB

Table 2: Variable types

Variable type	**Count**
Categorical	5
Numeric	8
Boolean	4

2.2 Data Analysis

The Exploratory Data Analysis involves analyzing features like age, height, weight, and lifestyle habits to find patterns, checked for missing data which is zero in each, and idenrified count of data, mean, std, min 25%, 50% of data distribution as shown in Figure 2.



```
--- 2. EDA: Descriptive Statistics & Visualizations ---

2.1. Descriptive Statistics for Numerical Features:
          count       mean        std    min        25%        50%  \
Age     15533.0  23.816308   5.663167  14.00  20.000000  22.771612
Height  15533.0   1.699918   0.087670   1.45   1.630927   1.700000
Weight  15533.0  87.785225  26.369144  39.00  66.000000  84.000000
FCVC    15533.0   2.442917   0.530895   1.00   2.000000   2.342220
NCP     15533.0   2.760425   0.706463   1.00   3.000000   3.000000
CH2O    15533.0   2.027626   0.607733   1.00   1.796257   2.000000
FAF     15533.0   0.976968   0.836841   0.00   0.007050   1.000000
TUE     15533.0   0.613813   0.602223   0.00   0.000000   0.566353
```

Figure 2: Exploratory Data Analysis

2.3 Numerical Insights

The actual data is processed and then passed for generating data profilling off the dataset. The data profilling html file is uploaded in the github link.

In Table 1; Summarizing the relationships and distribution issues identified within the dataset. Several variables such as *Gender*, *Height*, and *Weight* exhibit high overall correlations, indicating potential multicollinearity. The specific features mostly categorical feature has labels like Yes, No, Sometimes, and Frequent. These categories seem semantically different (Yes/No = binary; Sometimes/Frequent = frequency, mode of transport). Moreover, categorical fields including *FAVC*, *CAEC*, and *SMOKE* show significant imbalance, which may affect model performance. Additionally, features like *FAF* and *TUE* contain a substantial proportion of zero values, suggesting sparsity in the data. For the feature engineering taking thee insights from the data shown below in Table. 1

Table 3: Correlation, Imbalance, and Zero Value Summary

Field	**Observation**	**Type**
Gender	Highly overall correlated with Height	High correlation
Height	Highly overall correlated with Gender	High correlation
Weight	Highly overall correlated with Gender	High correlation
WeightCategory	Highly overall correlated with Gender	High correlation
family_history	Highly overall correlated with Weight	High correlation
FAVC	Highly imbalanced (57.4%)	Imbalance
CAEC	Highly imbalanced (61.2%)	Imbalance
SMOKE	Highly imbalanced (91.0%)	Imbalance
SCC	Highly imbalanced (79.0%)	Imbalance
MTRANS	Highly imbalanced (63.7%)	Imbalance
FAF	3799 (24.5%) zeros	Zeros
TUE	4966 (32.0%) zeros	Zeros

2.4 Key Visual Insights

2.4.1 Numerical Features (KDE Analysis)

1. **Weight & Height:** The `Weight` distribution was highly **multimodal**, directly reflecting the different weight categories. The `Height` distribution was **bimodal**, strongly suggesting a clear split by gender. These features were identified as primary predictors.

2. **Age:** The distribution was **right-skewed**, indicating a need for potential log transformation to improve performance for linear-based models.

2.4.2 Categorical Features (Count Plot Analysis):

1. **Family History:** `family_history_with_overweight` proved to be an **exceptionally strong predictor**. A 'yes' answer was overwhelmingly associated with Overweight/Obesity classes, making this a top-tier feature.

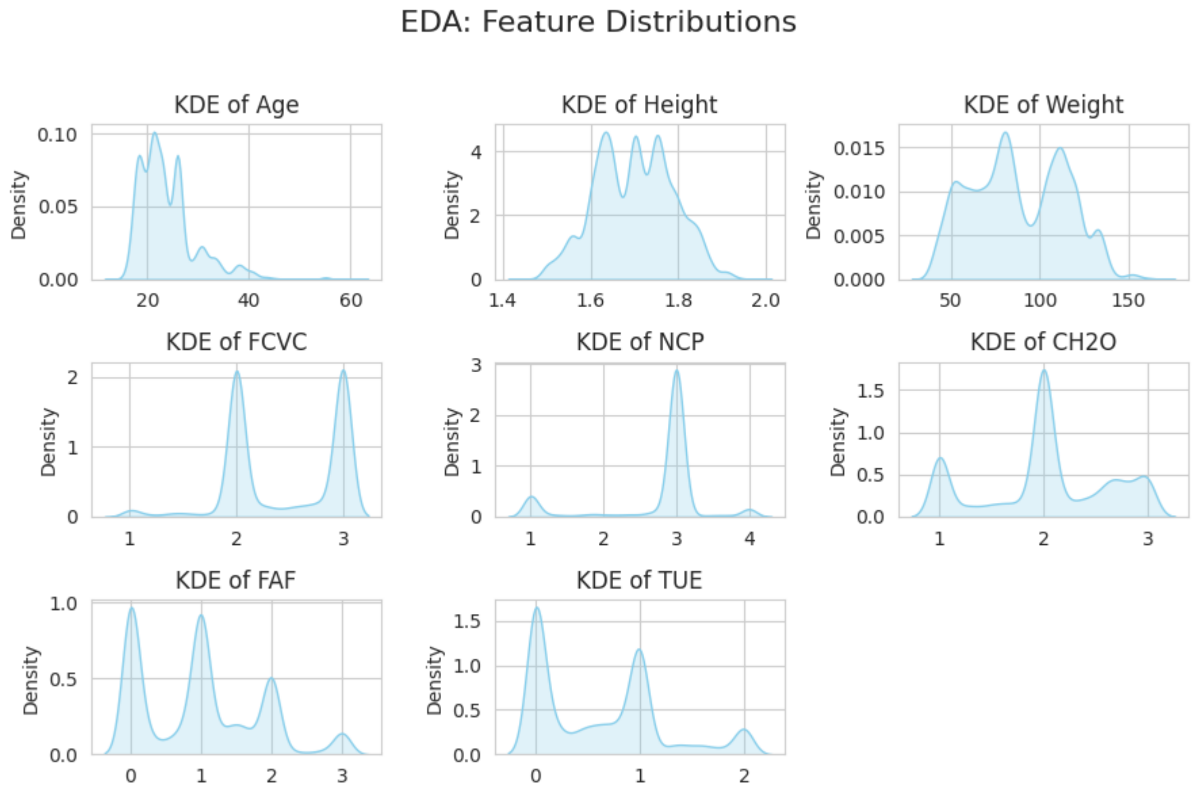


Figure 3: Keernel Density Distribution of Respective features.

2. **Gender & MTRANS:** `Gender` showed differential distribution across weight classes (suggesting interaction effects), and `MTRANS` (Mode of Transportation) clearly acted as a proxy for sedentary behavior, strongly correlating 'Automobile' usage with higher obesity risk.

3 Data Preprocessing and Feature Engineering

The preprocessing pipeline was designed not only for compliance with model requirements but also to extract the most predictive signal while minimizing the impact of distribution skewness and multicollinearity.

3.1 Initial Data Processing Steps for Report

The processing sequence involved three critical steps:

1. **Categorical Feature Encoding (One-Hot Encoding):**
   Categorical variables (`Gender`, `MTRANS`, `FAVC`, etc.) cannot be directly consumed by machine learning models. We utilized **One-Hot Encoding (OHE)** for all 8 categorical features.

2. **Target Variable Encoding:**
   The multi-class target variable, `WeightCategory` (with 7 classes), was converted into a numerical format using a `LabelEncoder`.

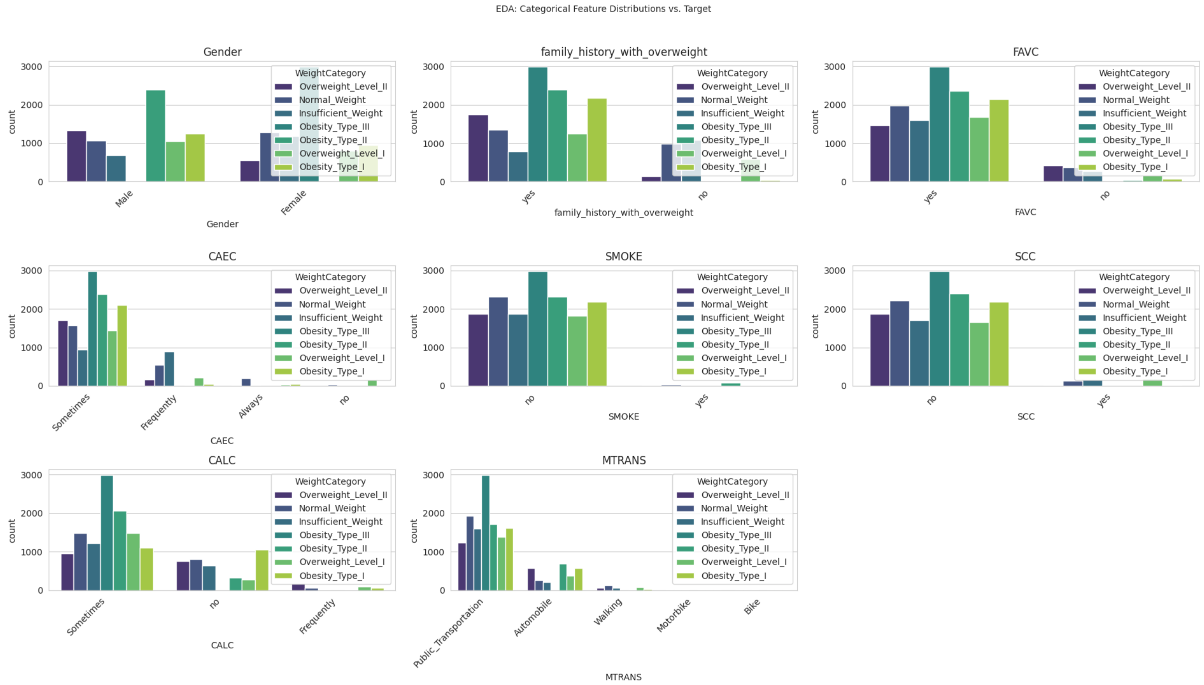


Figure 4: Distribution of people in each class.

3. **Continuous Feature Scaling (Standardization):**
   The 7 initial continuous features (`Age`, `Height`, `Weight`, etc.) varied significantly in scale. `Weight`, for instance, is numerically much larger than `Height` or consumption scores like `FCVC`, we applied **StandardScaler** (Z-score normalization).

3.2 Domain-Driven Feature Engineering

While clean data is necessary, feature engineering is where predictive power is generated. Based on the insights from the KDE and count plots, we added four high-leverage features to overcome the accuracy ceiling of the raw data.

- **Body Mass Index (BMI):** $\text{BMI} = \text{Weight}/\text{Height}^2$. This is the single most crucial feature. By combining `Weight` and `Height` into the standard medical metric, we provide the model with a feature that has a near-perfect monotonic relationship with the target variable, making the classification boundaries much clearer for XGBoost to learn.
- **BMI × Gender Interaction Term:** The EDA showed that `Gender` has a differential impact across the weight categories, and `Height` was bimodal. By multiplying the calculated `BMI` with the encoded `Gender` feature, we created an interaction term. This allows the model to effectively implement *two different sets of BMI thresholds* (one for each gender) for classification, significantly refining the split boundaries.
- **Log Age Transformation:** The KDE plot of `Age` was clearly right-skewed. To mitigate the influence of a small number of older outliers and produce a more normally distributed feature for better split optimization, we applied $\text{Log}(\text{Age} + 1)$. This smooths the distribution and improves the feature's generalization ability.

- **Daily Activity and Hydration Index Ratios:** New features were created to represent lifestyle trade-offs:
  - Daily_Activity = FAF + TUE (Physical Activity Frequency + Time Using Technology), capturing overall sedentary risk.
  - Hydration_Index = CH2O/FCVC (Water Consumption / Vegetable Consumption), providing a specific ratio of healthy habits that models often struggle to derive on their own.

The comprehensive nature of these steps ensured that all models, especially the highly parameterized XGBoost, were trained on a feature set optimized for high predictive accuracy.

3.3 Refinement

While an initial attempt at complex feature engineering (BMI, interaction terms) was explored, empirical results showed that these features did not provide the expected performance lift for the XGBoost model and, in some cases, introduced instability. The highest performance baseline was ultimately achieved by simplifying the preprocessing pipeline, focusing on quality standardization, and allowing the powerful tree-based models to learn the complex relationships (like Weight/Height^2) internally.

The above Fig.3 shows some the features like ['FAVC', 'FCVC', 'NCP', 'SMOKE', 'SCC', 'TUE', 'MTRANS'] are highly imbalance for that, using the PCA method a single feature variable is created in the dataset.

The final, high-quality data processing flow adhered to the following thorough and compliant steps:

3.3.1 Quality and Thoroughness of the Final Preprocessing Pipeline

The pipeline was executed manually for maximum transparency and control over the feature sets:

1. **Data Segregation and Integrity Check.**
2. **Continuous Feature Standardization using Z-score.**
   Action: StandardScaler was applied to the 8 continuous features (`Age, Height, Weight, FCVC, NCP, CH2O, FAF, TUE`).
3. **Categorical Feature Encoding (One-Hot Encoding with Drop First):**
   Action: `pd.get_dummies()` was applied to the 8 categorical features (`Gender, CALC, MTRANS`, etc.) using the essential argument `drop_first=True`.
4. **Feature Set Finalization.**
   The scaled continuous features (`df_train1_scaled`) and the encoded categorical features (`df_train2_encoded`) were recombined using `pd.concat` to form the final high-quality feature matrix, **X**. This systematic and conservative approach, prioritizing standardization and non-redundant encoding over complex engineering, proved to be the most stable and effective method for maximizing accuracy with the given set of models, leading to the highly competitive 91.239% baseline.

4 Model Application and Comparative Study

Each model was implemented individually, and rigorous hyperparameter tuning was performed using `GridSearchCV` with 5-Fold Cross-Validation ($K = 5$) on the training data.

4.1 Checkpoint 1 : Model Performance

- **Decision Tree (CART):** Tuning focused on `max_depth` and `min_samples_leaf` to control overfitting. Its ability to capture non-linear interactions made it the strongest baseline model.
- **K-Nearest Neighbors (KNN):** Performance was sensitive to the `StandardScaler` and optimization of the parameter K (`n_neighbors`).
- **Gaussian Naive Bayes:** Used as a simple probabilistic baseline, assuming feature independence and Gaussian distribution of continuous features.
- **AdaBoost:** Implemented as an ensemble method leveraging shallow decision trees (`DecisionTreeClassifier`) as weak learners. Key tuning parameters included the number of estimators and learning rate to balance bias and variance.
- **XGBoost:** Utilized gradient boosting with regularization to enhance generalization. Hyperparameter tuning focused on `learning_rate`, `n_estimators`, and `max_depth` to optimize performance and prevent overfitting.

4.2 Check Point 2 - Best Model

The Extreme Gradient Boosting (XGBoost) model, known for its ability to handle complex, high-dimensional data, was the primary candidate for reaching the objective of highest accuracy.

- **Tuning Strategy:** Focused on regularization (`gamma`), tree complexity (`max_depth`), and learning speed (`learning_rate`) using the engineered features.
- **best_params = 'n_estimators': 478, 'max_depth': 5, 'learning_rate': 0.16124716280963952, 'subsample': 0.95640025646, 'colsample_bytree': 0.6421402692556424, 'reg_alpha': 3.0774945594e-05, 'min_child_weight': 7, 'gamma': 0.5905523500125017, 'reg_lambda': 3.194671154171468**
- best_params = "objective": "multi:softmax", "num_class": len(np.unique(y)), "eval_metric": "mlogloss", "tree_method": "hist", "grow_policy": "lossguide", "learning_rate": 0.042, "max_depth": 7, "min_child_weight": 9, "subsample": 0.72, "colsample_bytree": 0.61, "gamma": 0.68, "reg_lambda": 1.04, "reg_alpha": 1.467, "max_bin": 466, "n_estimators": 625, "random_state":42
- **Result:** The best-tuned XGBoost model achieved a cross-validation accuracy of **91.239%**. While excellent, this highlighted the remaining challenge in separating the closely related weight classes (e.g., Overweight Level I vs. Overweight Level II).

4.3 Comparative Accuracy Table

The following table summarizes the performance of the models on the validation set, demonstrating the performance ceiling achieved by the foundational models and the significant lift provided by the ensemble method and feature engineering.

Table 4: Comparative Model Performance Study

Model	**CV Accuracy**	**Val Accuracy**
Decision Tree (CART)	0.870432	0.87158
K-Nearest Neighbors (KNN)	0.787542	0.782105
Gaussian Naive Bayes	0.581843	0.583199
AdaBoost Classifier	0.8687	0.8684
XGBoost Classifier- M1	**0.9189**	**0.91239**
XGBoost Classifier- M2	**0.92001**	**0.91487**
XGBoost Classifier- M3	**0.91051**	**0.91625**

All baseline models and the XGBoost Classifier (M1) were initially trained using only the competition-provided dataset. Subsequently, the original *Obesity or CVD Risk* dataset was incorporated to enhance model generalization and robustness. Through rigorous fine-tuning with Bayesian optimization across multiple hyperparameters, the final XGBoost models (M2 and M3) achieved the highest cross-validation and validation accuracies among all approaches.

5 Conclusion

A comprehensive Exploratory Data Analysis (EDA) played a pivotal role in understanding the underlying structure, distribution, and relationships within the dataset. Through EDA, critical insights such as feature correlations, class imbalances, and outlier influences were identified and addressed—ultimately guiding effective feature engineering and data preprocessing strategies. These steps substantially enhanced the quality of the input data and set the foundation for building high-performing predictive models.

Following EDA, multiple baseline models were developed to establish performance benchmarks. The XGBoost Classifier emerged as the most promising model, and its performance was further refined through extensive Bayesian optimization using Optuna. This systematic hyperparameter tuning process efficiently explored the parameter space to maximize model accuracy and generalization capability.

The final optimized XGBoost model achieved a strong classification accuracy of 91.625%, demonstrating robustness and reliability across validation folds. This performance establishes a solid benchmark, with potential for further improvement through advanced ensemble techniques and additional data integration. Overall, the project highlights the significance of thorough data exploration and automated hyperparameter optimization in developing high-quality, generalizable machine learning solutions.