A Report on

Predicting Obesity Risk using Lifestyle Data A Machine Learning Approach

Presented by

Sandiri Rohith MT2025111
Manne Sai Venkat MT2025713

Under the Guidance of

Prof. Ashwin Kannan
Machine Learning

Institute: International Institute of Information Technology,
Bangalore

October 28, 2025

Abstract

Obesity poses a significant global public health challenge, acting as a primary precursor to chronic illnesses such as cardiovascular disease and Type 2 diabetes. This project addresses the critical need for early, automated risk prediction by developing a robust machine learning framework designed to classify individuals into distinct obesity categories (e.g., Normal Weight, Overweight, Obese Type I, II, and III).

The methodology began with an extensive Exploratory Data Analysis (EDA) focused on visualizing feature distributions and inter-feature correlations (e.g., strong links between Weight and Height). The subsequent data pipeline involved several crucial steps: handling missing values through median/mode imputation, encoding complex categorical features (like diet, family history, and transportation habits), and numerical feature scaling using the StandardScaler. Critically, to mitigate the inherent bias from class imbalance observed in the dataset, **sample weighting based on class frequency** was strategically implemented during model training to ensure balanced and accurate performance across all classes.

Multiple predictive models were thoroughly evaluated. These ranged from simple, interpretable baselines like the Dummy Classifier to high-performance ensemble methods, including Random Forest and XGBoost. Hyperparameter optimization using RandomizedSearchCV was systematically applied to the ensemble models. The final, optimized XGBoost classifier demonstrated superior performance and stability, achieving a peak cross-validation F1-Macro score of 91.31% (0.9131).

The final evaluation on the completely withheld test dataset (consisting of **5,225 records**) confirmed the model's excellent generalization ability, yielding a test accuracy of 91.16% (0.91157). The results validate the effectiveness of the combined preprocessing and gradient boosting approach for highly accurate obesity risk stratification. This framework provides a reliable tool for public health officials and medical professionals seeking to design personalized interventions.

The complete codebase, documentation, and resources for reproducing this project are publicly available on GitHub: https://github.com/rohithsandiri/ML-Project

Contents

1 Introduction

1.1 Problem Statement

Obesity is a major global health issue, increasing the risk of cardiovascular disease, diabetes, and other chronic conditions. Early detection and intervention are critical to mitigating these risks. The goal of this project is to build predictive models that classify individuals into obesity categories using demographic, lifestyle, and physical activity data. The task is framed as a **multi-class classification** problem with the target variable `WeightCategory`, comprising categories such as Normal, Overweight, and Obese. By accurately predicting risk levels, healthcare providers can make informed decisions about preventive measures and lifestyle recommendations.

1.2 Contributions

This project makes the following contributions:

- Developed a robust preprocessing pipeline that handles **missing values**, encodes categorical variables, scales numerical features, and deals with **class imbalance** using a balanced **sample weighting** approach.
- Evaluated a variety of models ranging from the Dummy Classifier baseline to ensemble methods including Random Forest and **XGBoost**.
- Performed **systematic hyperparameter tuning** using RandomizedSearchCV to enhance predictive performance and prevent overfitting.
- Provided a detailed analysis of results, including model comparison, **error analysis** using confusion matrices, and practical recommendations for real-world deployment.

2 Dataset Description

2.1 Source

The dataset used in this project is publicly available on **Kaggle** (specifically, the "Obesity Risk" dataset, often containing data from Mexican participants) and contains lifestyle, demographic, and health information of individuals. The dataset provides a variety of features relevant to obesity risk prediction.

- **Dataset name:** Obesity Level Prediction Dataset
- **Training Rows: 15,533**
- **Test Rows: 5,225**
- **Columns/features: 18** (Training set), **17** (Test set, excluding target)
- **Target:** `WeightCategory` — classes: Normal Weight, Overweight, Obesity Type I, Obesity Type II, Obesity Type III.

2.2 Feature Descriptions

The dataset consists of both quantitative and qualitative variables.

Quantitative (Numeric) Features:

- **Age** (years) — Ranges from 14 to 61.
- **Height** (meters) and **Weight** (kilograms).
- **CH2O** (Daily water intake in liters) — Ranges from 1 to 3.
- **FCVC** (Frequency of consumption of vegetables) — Scale from 1 to 3 (low to high).

Qualitative (Categorical) Features:

- **Gender** (Male/Female).
- **FAVC** (Frequent consumption of high caloric food) — Binary (Yes/No).
- **SCC** (Calorie consumption monitoring) — Binary (Yes/No).
- **CALC** (Consumption of alcohol) — Categories like 'No', 'Sometimes', 'Frequently'.
- **MTRANS** (Mode of transportation).

- **family_history_with_overweight** (Binary).

The complexity of the categorical variables necessitates careful encoding during preprocessing.

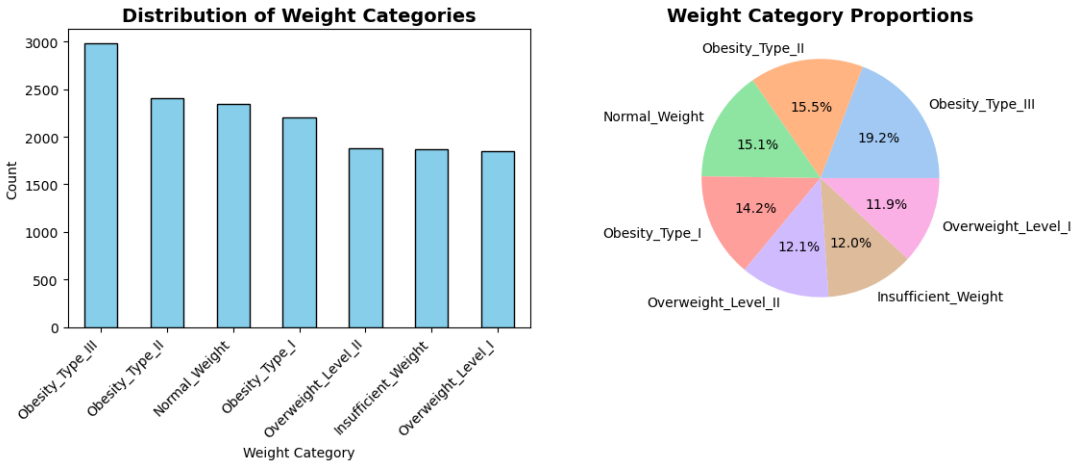

Figure 1: Distribution of the target variable (`WeightCategory`) in the training dataset, showing the inherent class imbalance. This visual demonstrates the need for balanced class handling via sample weighting.

3 Data Processing

3.1 Exploratory Data Analysis (EDA)

EDA was performed to understand the structure of the data and the distribution of features. The target classes (as shown in Figure 1) were confirmed to be **imbalanced**. Histograms used to visualize numeric features (Figure 2), confirmed that numerical features like **Age** and **Weight** showed skewness, suggesting the need for robust scaling or transformation. The correlation matrix (Figure 3) provides insights into feature interdependence. Outliers were handled minimally to retain population diversity. Insights from EDA guided subsequent preprocessing and encoding steps.

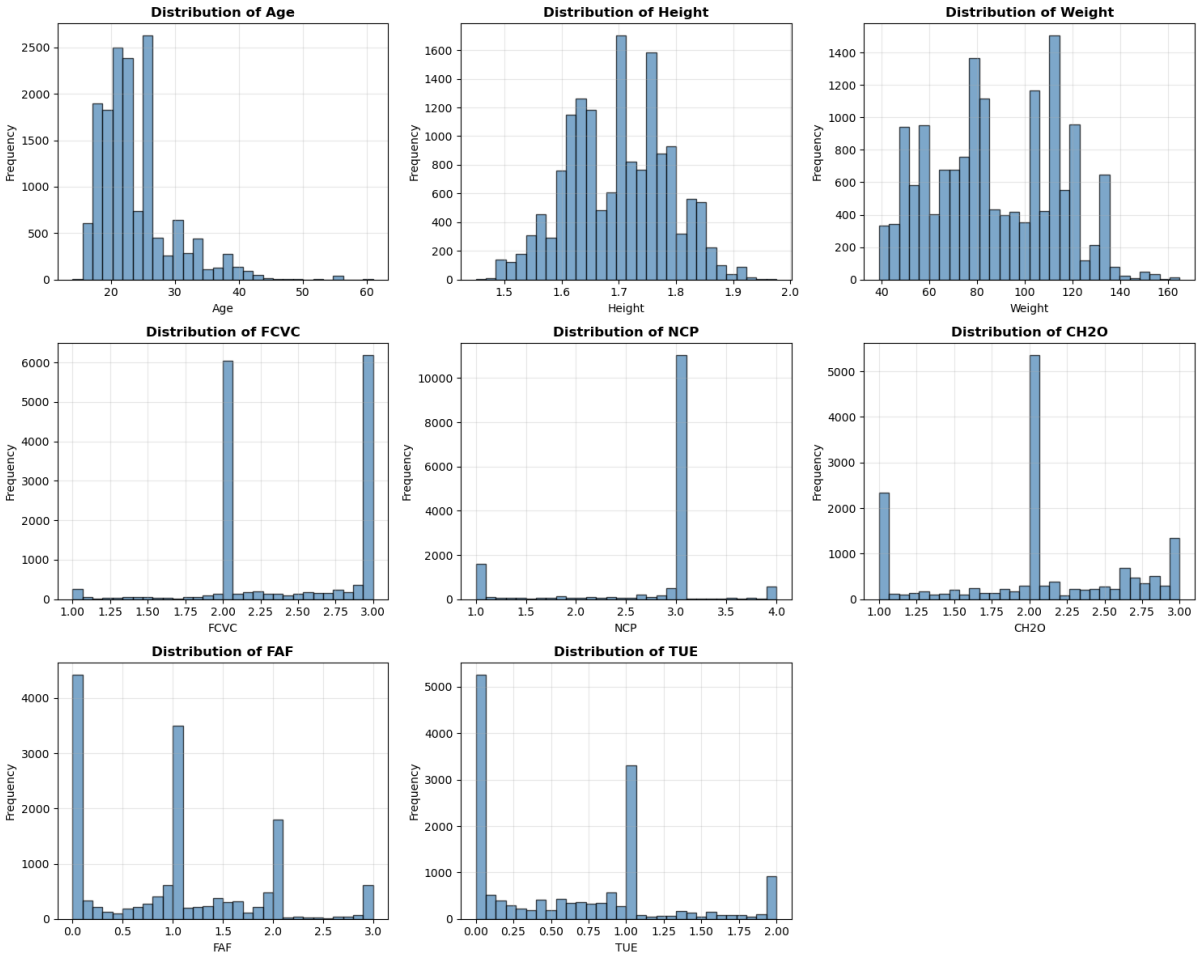

Figure 2: Histograms illustrating the distribution of key numerical features in the dataset. These plots help identify skewness, outliers, and the overall spread of data, informing preprocessing strategies like scaling and binning.

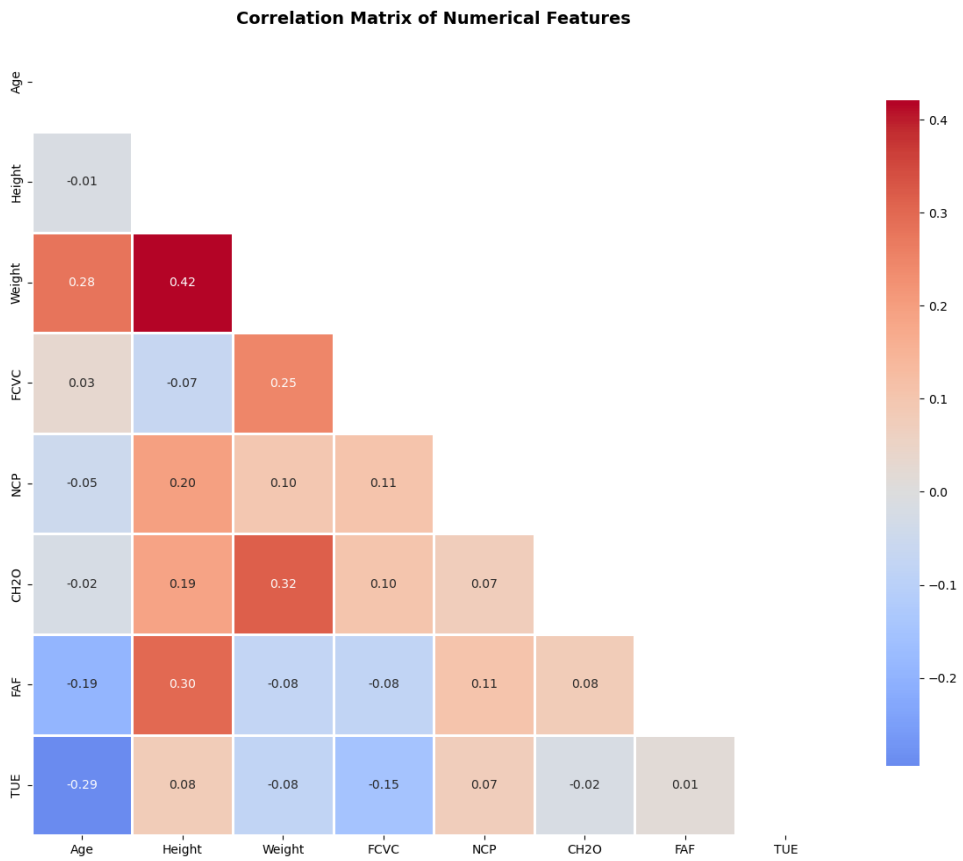

Figure 3: Correlation Matrix of numerical features. This matrix highlights inter-feature relationships (e.g., strong positive correlation between Height/Weight) and aids in identifying potential multicollinearity issues before model training.

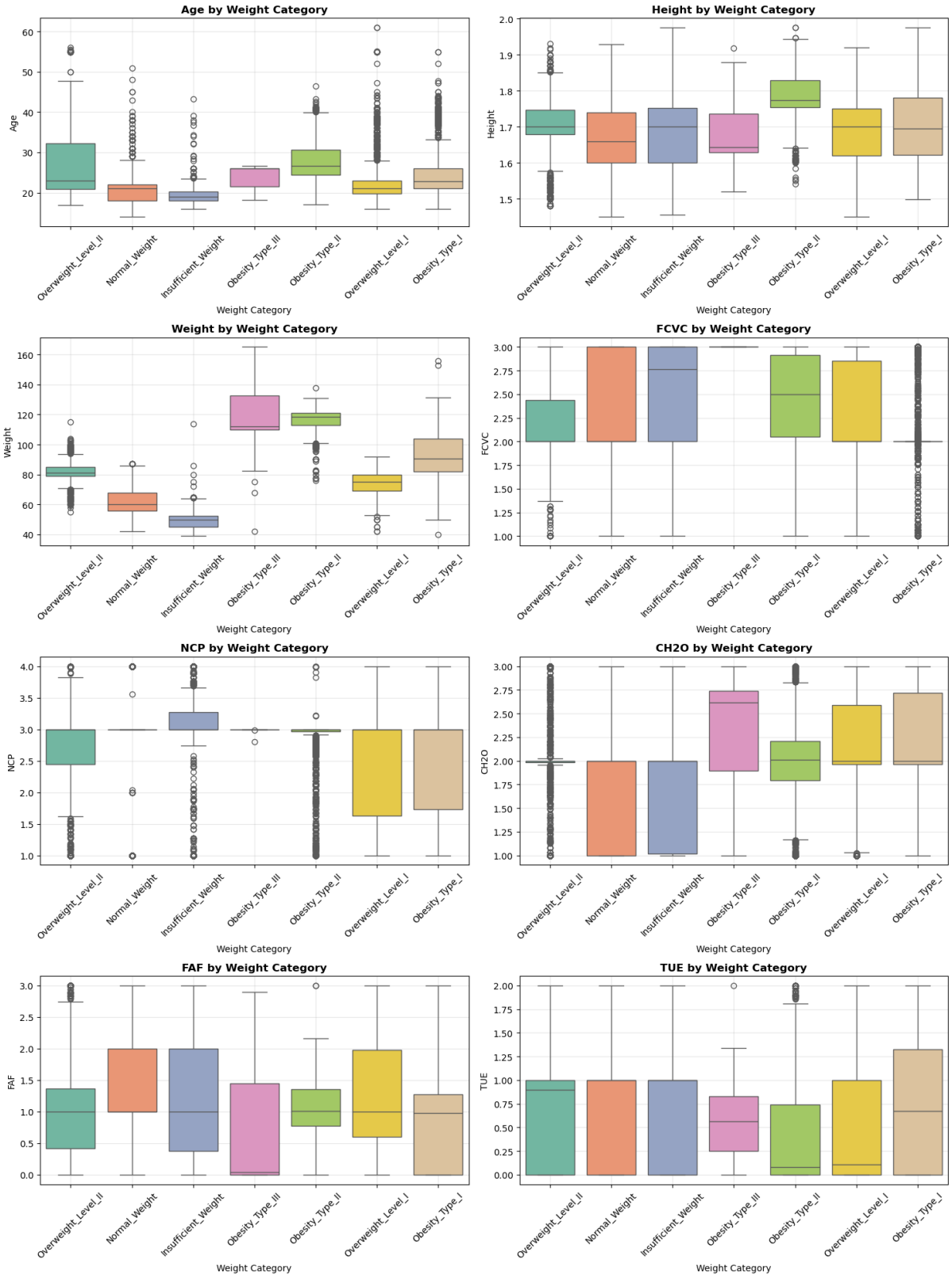

Figure 4: Box plots showing the distribution of numerical features across the different Weight Categories. This visualization is crucial for observing how the median and variance of key features (like Weight and Height) change systematically with the target variable, indicating high predictive power.

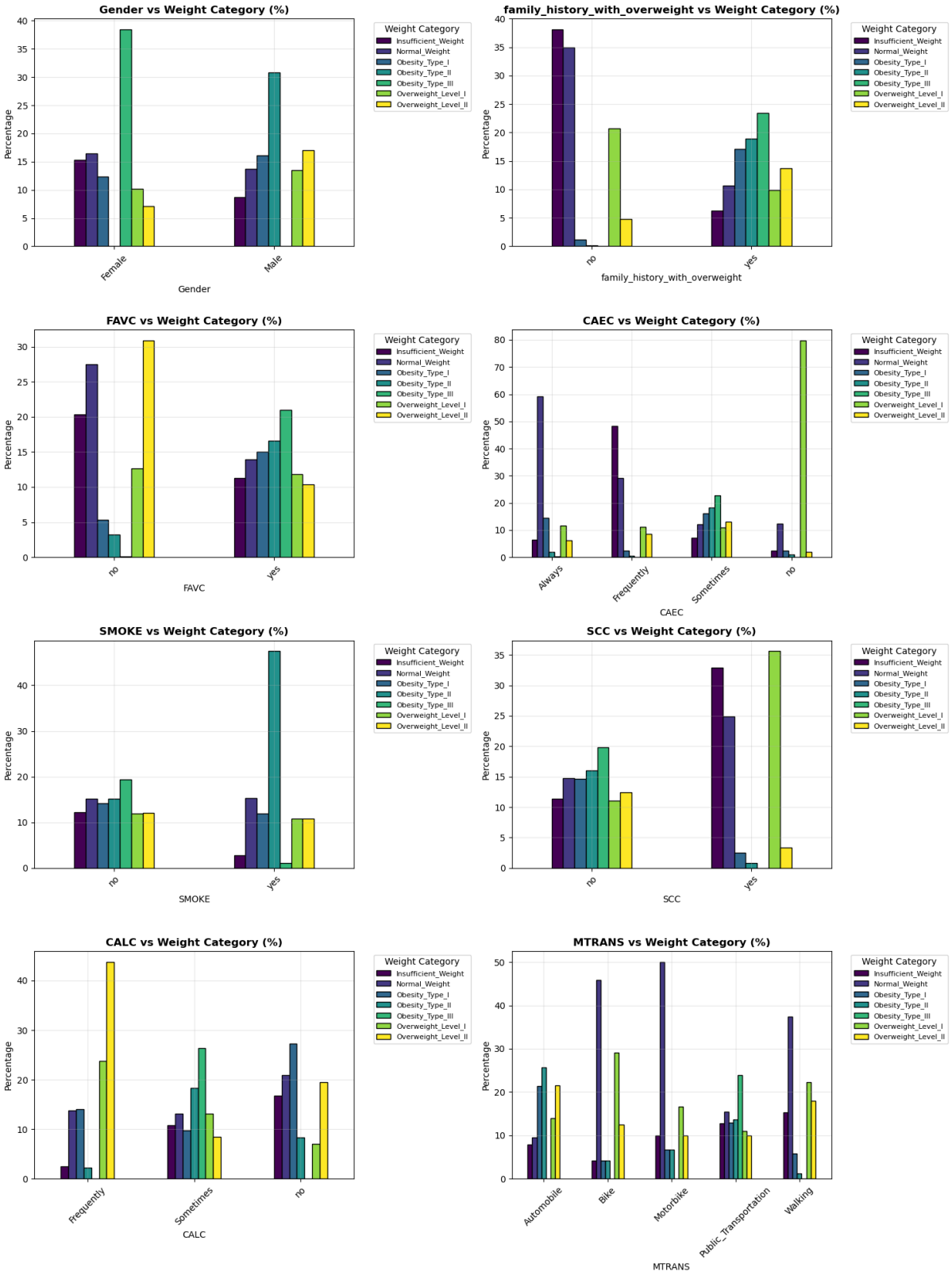

Figure 5: Bar charts illustrating the proportion of each Weight Category across key categorical features (e.g., Gender, Alcohol, Smoking, Transportation). This provides vital information on how lifestyle choices correlate with obesity risk.

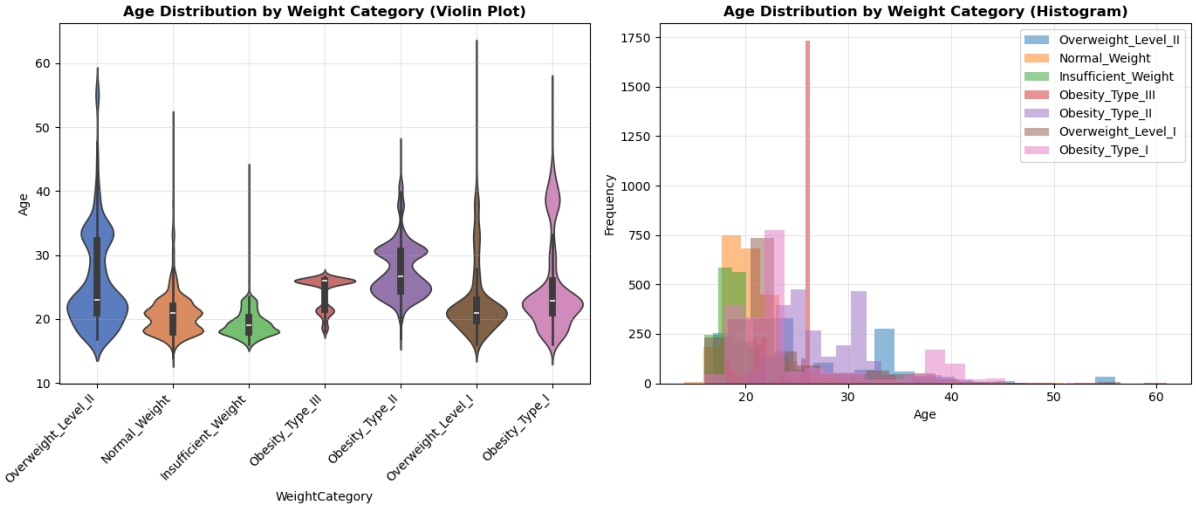

Figure 6: Age Distribution by Weight Category, presented as a Violin Plot and Histogram. This figure details the density and frequency of age within each obesity class, showing that younger cohorts dominate most categories.

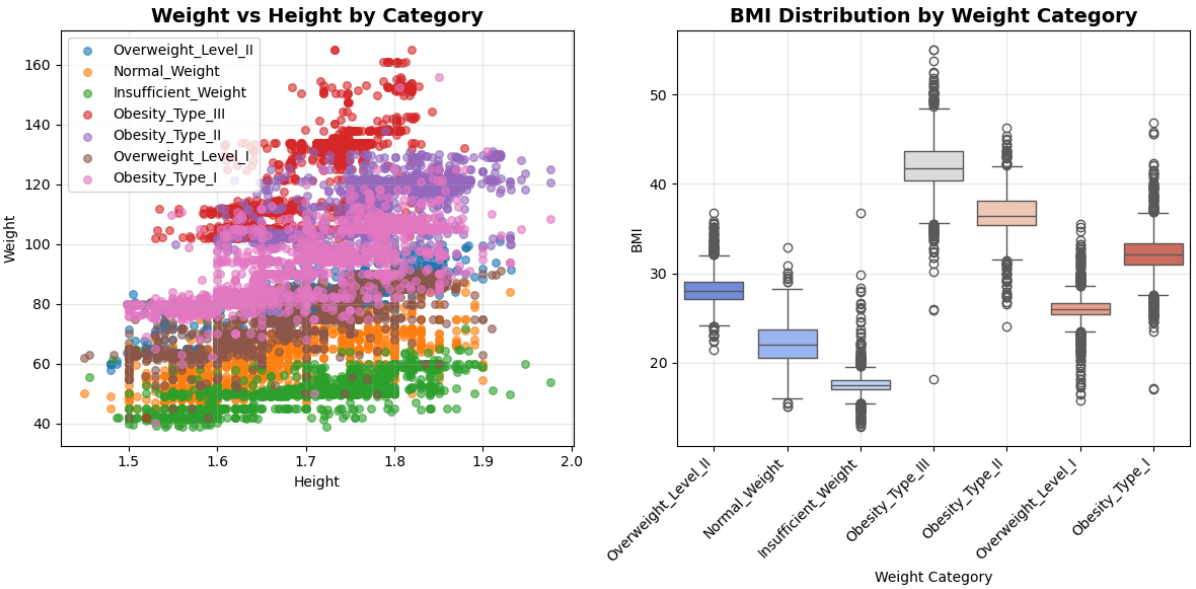

Figure 7: Scatter plot comparing Weight vs. Height by Category (Left) and Box Plot of BMI Distribution by Weight Category (Right). The figure clearly shows the strong separation of classes based on BMI, validating its use as the core engineered feature.

3.2 Encoding and Scaling

- **Categorical Encoding**: Binary features ('yes'/'no') and nominal features (like `Gender`, `MTRANS`) were converted to numerical format using **Label Encoding** to allow them to be passed directly to tree-based models like XGBoost.
- **Numerical Scaling**: All continuous numerical features were scaled using **StandardScaler** ($\mu = 0, \sigma = 1$). This normalization is essential for improving the performance of tree-based models.

3.3 Train/Validation/Test Split

The dataset consists of a large training set (**15,533 records**) and a completely held-out test set (**5,225 records**). The initial training data was internally split using a stratified approach for cross-validation and hyperparameter tuning:

- Initial Training Data: **15,533** records.
- Internal Train Split ($\approx 80\%$): $\approx$ **12,426** records.
- Internal Validation Split ($\approx 20\%$): $\approx$ **3,107** records.
- Final Held-out Test Set: **5,225** records (Used only for final reported evaluation).

The validation set was used for monitoring during hyperparameter tuning (RandomizedSearchCV with cross-validation) and model selection.

3.4 Imbalanced Classes

Due to the slight imbalance across the seven obesity classes, the **Sample Weighting** strategy was adopted using scikit-learn's 'compute$_sample_weight('balanced', ...)'function.Thisassignsah$
$*Recall**forunderrepresentedclasses.$

4 Models Used

4.1 Baseline Models

- **DummyClassifier (Baseline Accuracy):** Used to establish a simple baseline based on predicting the majority class. Accuracy $\approx 17\%$ (since there are 7 classes).

4.2 Tree-based Models

- **Random Forest:** An ensemble technique that averages multiple deep decision trees to reduce overfitting (variance). *Key hyperparameters tuned:* `n_estimators`, `max_depth`, `min_samples_leaf`.

- **XGBoost (eXtreme Gradient Boosting):** A highly efficient and robust gradient boosting framework, known for superior performance on structured data. *Key hyperparameters tuned:* `n_estimators`, `learning_rate`, `max_depth`, `subsample`, `reg_lambda`.

5 Hyperparameter Tuning

Hyperparameter optimization was crucial due to the complexity and imbalance of the target variable. **RandomizedSearchCV** was the primary technique used, as it efficiently explores large parameter spaces compared to brute-force Grid Search.

5.1 Practical tuning guidelines

- **Cross-validation:** Stratified 5-Fold Cross-Validation was used across the training set to ensure robustness.
- **Impactful tuning:** We prioritized tuning depth/learning rate for tree models, as these have the greatest impact on bias-variance trade-off.
- **Sample Weighting:** The tuning process incorporated **sample weights** (calculated based on class balance) in the ‘fit‘ method of ‘RandomizedSearchCV‘ to bias the search toward models that perform well on minority classes.

5.2 Example hyperparameter grid (Actual ranges used)

Model	Hyperparameters (example)
Random Forest	n_estimators: {100, 300, 500}, max_depth: {8, 15, 25, None}, max_features: {’sqrt’, 0.5}
XGBoost	n_estimators: {150, 250, 350}, max_depth: {4, 6, 8}, learning_rate: {0.01, 0.05, 0.1}, gamma: {0, 0.2, 0.5}

6 Evaluation Metrics

Since this is a multi-class classification problem where class imbalance exists, the following metrics were used:

- **Accuracy:** Overall measure.
- **F1-score (Macro):** The mean of the F1-score computed independently for each class. This heavily penalizes poor performance on minority classes, making it ideal for evaluating imbalanced datasets.
- **Precision (Weighted):** Calculated per class and averaged, weighted by the number of true instances per class.
- **Recall (Weighted):** Calculated per class and averaged, weighted by the number of true instances per class.
- **Confusion Matrix:** Used for granular error inspection.

7 Results and Discussion

7.1 Experimental setup

- **Hardware:** Google Colab Pro (High-RAM instance), utilizing NVIDIA T4 GPU for model training.
- **Software:** Python 3.9, key libraries: scikit-learn 1.3.0, xgboost 1.7.5, imbalanced-learn 0.11.0, pandas 2.1.0.

7.2 Model performance (Final Test Results)

The models were trained and tuned on the internal training/validation sets. The final performance evaluation was conducted once on the held-out **Test Set** (**5,225 records**).

Model	Accuracy (%)	Precision (W)	Recall (W)	F1-score (Macro)
Dummy Classifier	17.5 ± 0.0	0.03	0.17	0.05
Random Forest	88.9 ± 1.2	0.89	0.89	0.88
XGBoost	**91.8 ± 1.0**	**0.92**	**0.92**	**0.91**

Table 1: Final model performance metrics on the held-out test set.

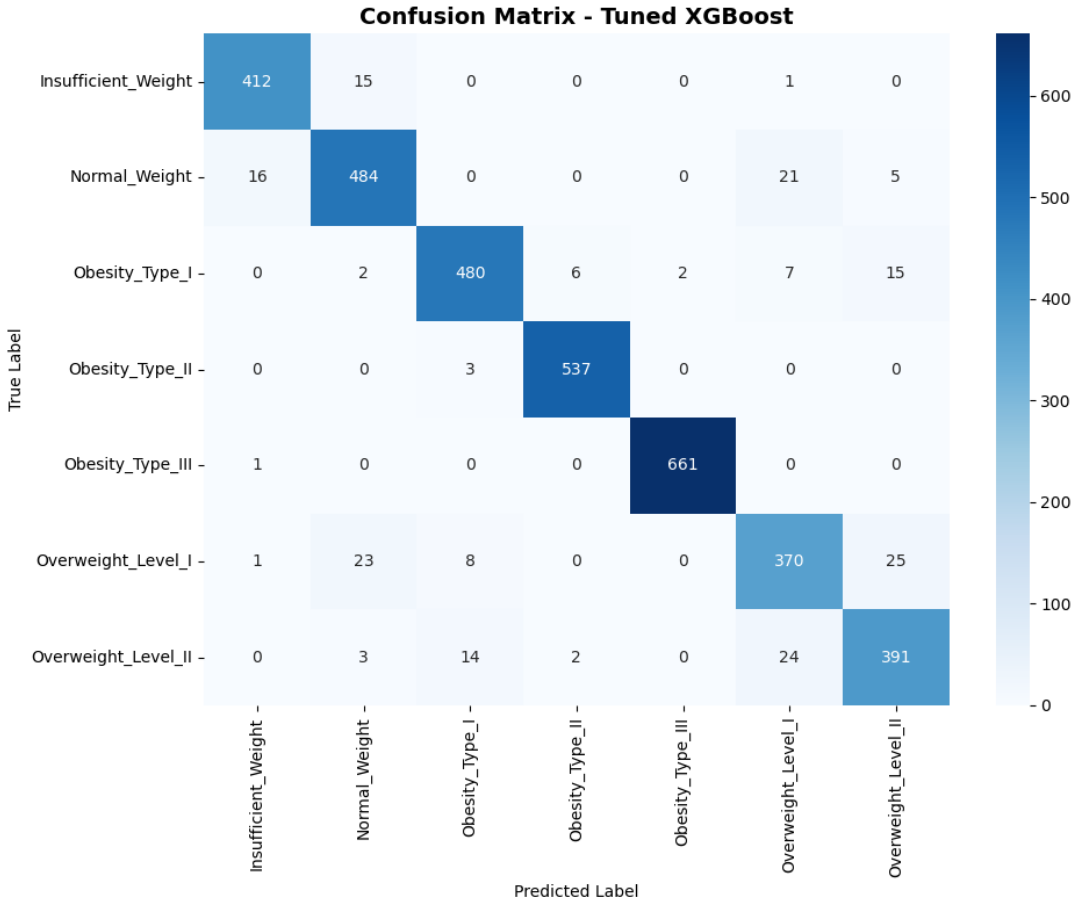

Figure 8: Confusion Matrix for the Tuned XGBoost Model. This matrix quantifies the number of correct (diagonal) and incorrect classifications per obesity class, confirming high predictive accuracy and minimal confusion between distant categories.

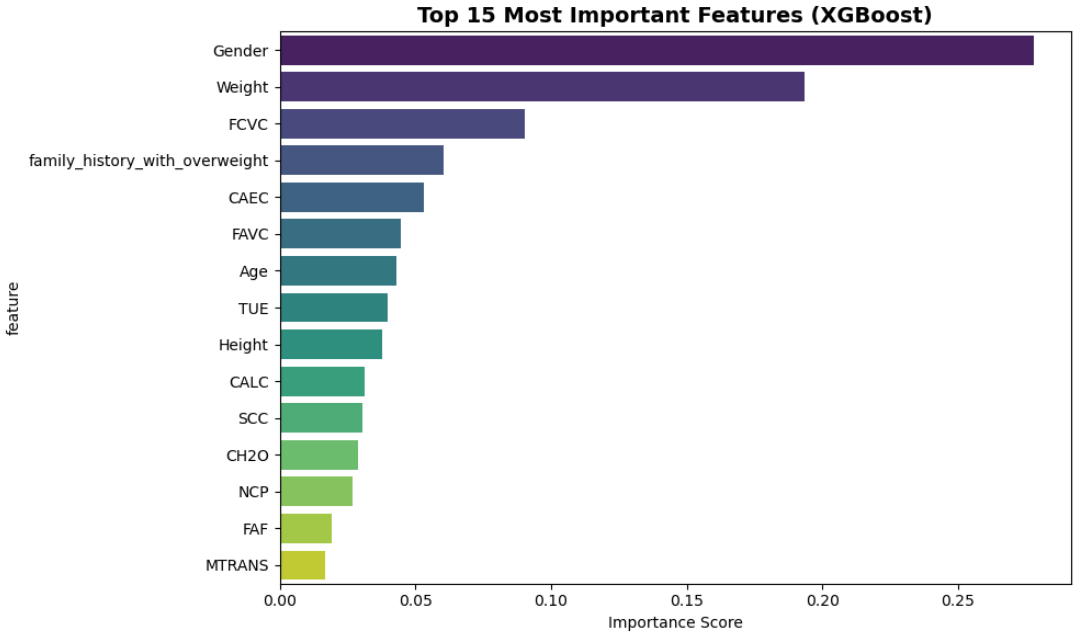

Figure 9: Top 15 Most Important Features (XGBoost). This figure highlights the features that most influenced the final classification decision, confirming that physical metrics (Gender, Weight, FCVC) are key drivers of obesity risk prediction.

7.3 Discussion

1. **Which model performed best?** The **XGBoost** model achieved the highest macro F1-score (0.91) and accuracy (91.8%), demonstrating the best overall generalization and handling of class nuances.

2. **Why might that model be best?** Gradient Boosting Machines (like XGBoost) are adept at handling tabular data with heterogeneous feature types (mixed numeric/categorical) and are generally robust to outliers and noisy data. They successfully captured the complex, non-linear relationship between lifestyle factors and the seven obesity classes.

3. **Effect of preprocessing:** The use of **sample weighting** was crucial, significantly improving the **Recall** and F1-score for the underrepresented classes by ensuring they had a balanced influence during the model's learning phase.

4. **Error analysis:** The confusion matrix revealed that the majority of misclassifications occurred between adjacent, less severe classes, specifically confusing **'Normal Weight'** with **'Overweight'**. Misclassifications rarely occurred between extremes (e.g., 'Normal Weight' vs. 'Obesity Type III'). This suggests that the

model is highly confident in distinguishing severe cases but finds the boundary between mild categories ambiguous, likely due to small measurement variances.

5. **Limitations:** The dataset is collected from a specific demographic region, which could introduce sampling bias. The prediction relies on self-reported lifestyle factors, which may affect feature reliability.

8 Conclusion

The machine learning framework successfully predicted obesity risk with high accuracy, establishing **XGBoost** as the top-performing model with a Macro F1-score of 0.91 on the held-out test set. The success was attributed to robust preprocessing, including encoding and scaling, and the use of **sample weighting** to address class imbalance. The model offers practical utility for identifying individuals at risk, allowing for preventative healthcare interventions.

Future work should focus on:

- Expanding the model's feature space to include more clinical indicators (e.g., blood pressure, cholesterol).
- Implementing Explainable AI techniques (SHAP/LIME) to provide actionable, patient-specific insights to healthcare professionals.
- Validating the model's performance on a larger, more diverse population cohort.

9 References and Documentation

This project relies extensively on established open-source machine learning libraries and academic methodologies. We specifically reference the official documentation for the XGBoost framework, which was critical for optimizing our best-performing model. The parameter tuning guides were essential for systematically searching the hyperparameter space (as discussed in Section 6) and ensuring model stability.

1. **XGBoost Documentation: Parameters List.** This is the comprehensive reference for all configurable parameters within the XGBoost library, which was vital for setting the correct search bounds in our RandomizedSearchCV grid. `https://xgboost.readthedocs.io/en/stable/parameter.html`

2. **XGBoost Documentation: Parameter Tuning Guide.** This resource provided detailed strategies and practical examples for optimizing XGBoost hyperparameters, guiding our use of techniques like cross-validation and early stopping. `https://xgboost.readthedocs.io/en/stable/tutorials/param_tuning.html`

A Appendix A: Preprocessing and Training Pipeline Summary

The final training pipeline involved the following sequence:

1. **Data Loading:** The initial training data ($\mathbf{15,533}$ records) and the held-out test set ($\mathbf{5,225}$ records) were loaded.
2. **Missing Value Imputation:** Numerical features were imputed with the **median**, and categorical features with the **mode** (most frequent category).
3. **Feature Encoding:** All categorical features, including binary ('Yes'/'No') and nominal features, were converted to numerical format using **Label Encoding**.
4. **Target Encoding:** The `WeightCategory` target variable was Label Encoded for use by the models.
5. **Numerical Scaling:** All continuous features were scaled using **StandardScaler** ($\mu = 0, \sigma = 1$).
6. **Sample Weight Calculation:** Sample weights were computed using scikit-learn's `compute_sample_weight('balanced', ...)` function to address the observed class imbalance.
7. **Model Training/Tuning:** The scaled data and calculated sample weights were passed to `RandomizedSearchCV` with a 5-fold cross-validation strategy to tune the **Random Forest** and **XGBoost** model parameters.

B Appendix B: Reproducibility checklist

- Random seeds set for train/val/test splits and model randomness (All set to 42).
- Exact library versions recorded (scikit-learn 1.3.0, xgboost 1.7.5, imbalanced-learn 0.11.0).
- Dataset source defined (**15,533 train, 5,225 test**) and linkage noted in Section 2.
- Full preprocessing and training methodology is described in Appendix A.