Machine Learning Project
Obesity Risk Prediction Using Machine Learning

AIT511: Course Project 1

A Project Report Submitted
in Partial Fulfillment of the Requirements
for the Award of the Degree

MASTER OF TECHNOLOGY

in

Computer Science and Engineering

Submitted by

Abhijeet Kumar Gupta , Devdeep Sarkar

(MT2025003 , MT2025042)

Submitted to

Department of Computer Science and Engineering
International Institute of Information Technology
Bangalore - 560100, India

Contents

List of Tables

List of Figures

Abstract

This project investigates the application of machine learning techniques for predicting obesity risk categories based on demographic, lifestyle, and physiological data. The objective is to design and evaluate a predictive framework capable of classifying individuals into multiple obesity levels, supporting early health risk assessment and preventive care.

Three distinct preprocessing pipelines were implemented and analyzed: (i) a baseline pipeline featuring Body Mass Index (BMI) derivation and scaling, (ii) a transformation-based pipeline employing Box–Cox and Yeo–Johnson power transforms to correct feature skewness, and (iii) a categorical-native encoding pipeline leveraging LightGBM's built-in categorical feature support. These were systematically evaluated across four model families—Logistic Regression, Gradient Boosting, XGBoost, and LightGBM—to assess their suitability for structured health datasets.

Two hyperparameter optimization strategies were explored: Random Search and Bayesian Optimization using the Optuna framework. Optuna's Tree-structured Parzen Estimator (TPE) approach demonstrated superior efficiency in identifying optimal configurations within constrained search spaces. The Optuna-tuned LightGBM model achieved the highest validation accuracy of **93.37%** and a weighted F1-score of **0.9332**, outperforming all other model and preprocessing combinations.

This report presents a comprehensive analysis of data preparation techniques, model architectures, and optimization workflows. It further discusses the relative effectiveness of transformation and encoding strategies and provides insights into the interpretability of the best-performing model. The study underscores the potential of gradient boosting and Bayesian optimization frameworks for developing accurate, scalable, and interpretable obesity risk prediction systems.

Chapter 1

Introduction

Obesity has emerged as one of the most critical public health challenges of the 21st century. It is a complex condition characterized by excessive body fat accumulation, often resulting from an imbalance between calorie intake and expenditure. According to the World Health Organization (WHO), worldwide obesity has nearly tripled since 1975, with more than 1.9 billion adults classified as overweight and over 650 million of them obese. The implications of obesity extend far beyond physical appearance—it is associated with chronic diseases such as diabetes mellitus, hypertension, cardiovascular disorders, and certain cancers. Consequently, obesity places a substantial burden on healthcare systems globally.

1.1 Background

Traditional methods for assessing obesity rely primarily on simple anthropometric indicators such as Body Mass Index (BMI), waist-to-hip ratio, and other body composition measures. While these metrics are useful for general categorization, they fail to capture the multidimensional and behavioral aspects of obesity, which depend on dietary patterns, lifestyle choices, and physical activity levels.

With the increasing availability of large-scale health and lifestyle datasets, machine learning (ML) has become a promising approach to model obesity risk more comprehensively. ML techniques can capture complex, nonlinear relationships between variables and offer predictive insights that are difficult to obtain using traditional statistical models. In the context of healthcare analytics, data-driven predictive models can support early diagnosis and preventive interventions by identifying individuals at higher risk of obesity.

This study focuses on building machine learning models that can classify individuals into specific obesity categories based on demographic, behavioral, and physiological attributes. Such classification helps in identifying at-risk groups and can inform targeted health recommendations or policy planning.

1.2 Project Motivation

The motivation for this project stems from the growing demand for intelligent and data-driven systems in healthcare. Manual or rule-based assessments of obesity often overlook subtle interactions between multiple contributing factors, such as physical inactivity, dietary patterns, and lifestyle habits. A data-centric approach using machine learning enables automatic discovery of these patterns from historical data, thereby improving predictive accuracy and scalability.

Additionally, while numerous ML algorithms exist, their effectiveness heavily depends on the preprocessing, encoding, and transformation strategies applied to the data. Real-world datasets often include a mix of numerical and categorical variables with varying distributions, necessitating appropriate transformations (like Box–Cox or Yeo–Johnson) to improve learning stability. Moreover, hyperparameter optimization plays a crucial role in achieving robust model performance.

This project, therefore, aims to explore multiple model architectures—ranging from simple linear classifiers to advanced gradient boosting algorithms—and systematically analyze the impact of different preprocessing and optimization techniques. The ultimate goal is to identify a reliable pipeline that combines interpretability, computational efficiency, and high predictive accuracy for obesity risk classification.

1.3 Objectives

The key objectives of this project are as follows:

- To preprocess and clean the dataset for multi-class classification of obesity levels.
- To engineer and evaluate features using transformation techniques such as Box–Cox and Yeo–Johnson.
- To implement and compare several machine learning algorithms, including Logistic Regression, Gradient Boosting, XGBoost, and LightGBM.
- To conduct hyperparameter tuning using both Random Search and Bayesian Optimization (Optuna) methods.
- To identify the most effective combination of preprocessing, encoding, and optimization strategies that deliver superior accuracy and robustness.

1.4 Report Overview

This report is organized into six chapters, including Introduction, Dataset Description, Methodology, Experiments and Results, Discussion and Conclusion.

Chapter 2

Dataset Description

The dataset used in this study contains a total of **20,758 individual records**, divided into **15,533 training samples** and **5,225 test samples**. Each record represents an individual's demographic, lifestyle, and health-related attributes. The dataset includes **17 features**, comprising both numerical and categorical variables, along with a target variable — `WeightCategory`, which classifies individuals into seven obesity levels: *Insufficient Weight, Normal Weight, Overweight Level I–II, and Obesity Type I–III.*

Feature Overview

- **Numerical Features:** Age, Height, Weight, Number of Meals (NCP), Frequency of Vegetable Consumption (FCVC), Physical Activity Frequency (FAF), and Water Intake (CH2O).
  These variables capture quantitative aspects of an individual's dietary habits and physical activity patterns.

- **Categorical Features:** Gender, Smoking Habit (SMOKE), High-Calorie Food Consumption (FAVC), Eating Between Meals (CAEC), and Mode of Transportation (MTRANS). These attributes describe lifestyle behaviors that influence obesity risk.

Data Quality and Preprocessing Insights

No missing values were detected across any of the features, confirming that the dataset is clean and complete. Preliminary exploratory analysis revealed moderate skewness in certain numerical variables such as *Age*, *NCP*, and *Weight*, which justified the application of mathematical transformations (e.g., *Box-Cox* or *Yeo-Johnson*) to improve normality and stabilize variance.

Furthermore, correlation and multicollinearity checks confirmed that the numerical features are independent and suitable for modeling.

Summary

Overall, the dataset provides a rich, balanced, and interpretable representation of multiple demographic and behavioral factors influencing obesity. It serves as an excellent foundation for predictive modeling and classification tasks aimed at assessing obesity risk levels in individuals.

Chapter 3

Methodology

3.1 Feature Engineering and Preprocessing

Three distinct pipelines were evaluated:

3.1.1 Pipeline A: Baseline (No Transformation)

- Compute Body Mass Index (BMI): $BMI = \frac{Weight}{Height^2}$.
- Drop Height to reduce redundancy.
- One-hot encode categorical features.
- Standardize numeric attributes.

3.1.2 Pipeline B: Numeric Transformations

- Apply **Box–Cox** to Age, NCP.
- Apply **Yeo–Johnson** to Weight.
- Compute BMI and standardize numeric variables.

3.1.3 Pipeline C: Categorical-Native (Boosting Only)

- Convert categorical variables to `category` dtype.
- Train LightGBM using native categorical handling.
- Exclude BMI in this configuration.

3.2 Mathematical Background

Box–Cox:

$$T_{BC}(x;\lambda) = \begin{cases} \dfrac{x^{\lambda}-1}{\lambda}, & \lambda \neq 0, \\ \ln(x), & \lambda = 0. \end{cases}$$

Yeo–Johnson:

$$T_{YJ}(x;\lambda) = \begin{cases} \dfrac{(x+1)^{\lambda}-1}{\lambda}, & x \geq 0, \lambda \neq 0, \\ \ln(x+1), & x \geq 0, \lambda = 0, \\ -\dfrac{(-x+1)^{2-\lambda}-1}{2-\lambda}, & x < 0, \lambda \neq 2, \\ -\ln(-x+1), & x < 0, \lambda = 2. \end{cases}$$

3.3 Models Used

3.3.1 Logistic Regression

Description: Linear classification using softmax-based probabilities, trained to minimize cross-entropy loss.

Implementation: `LogisticRegression(multi_class='multinomial', solver='saga')`

3.3.2 Gradient Boosting

Description: Sequential ensemble of weak learners minimizing residual error.

Implementation: `GradientBoostingClassifier` with tuned `n_estimators` and `learning_rate`.

3.3.3 XGBoost

Description: Optimized boosting framework with regularization and parallel computation.

Implementation: `XGBClassifier(tree_method='hist', use_label_encoder=False)`.

3.3.4 LightGBM

Description: Leaf-wise gradient boosting with histogram binning and native categorical support.

Implementation: `LGBMClassifier` using two pipelines — (i) transformed numeric data + BMI, (ii) categorical-native data.

3.4 Hyperparameter Tuning

3.4.1 Random Search

Randomly samples hyperparameter combinations from distributions and evaluates them using 5-fold cross-validation.

3.4.2 Bayesian Optimization (Optuna)

Optuna's TPE sampler adaptively explores parameter space to maximize validation accuracy. Conducted with 40 trials.

Chapter 4

Experiments and Results

This chapter presents a comprehensive evaluation of various machine learning models, preprocessing techniques, and optimization strategies applied to obesity risk classification. The goal was to systematically assess how different transformations, encodings, and hyperparameter search methods affect model performance on a consistent validation framework. Ten experimental configurations were tested by combining three preprocessing pipelines, four model families, and two optimization strategies.

Each experiment was evaluated using a stratified 80:20 train–validation split, ensuring class balance across folds. Accuracy and weighted F1-score were chosen as the key performance metrics, with additional diagnostic visualizations to interpret model behavior.

4.1 Experimental Design

To ensure fairness and reproducibility, all experiments used the same dataset and feature set except when explicitly testing for the impact of a given feature (e.g., BMI).

- **Train–validation split:** 80:20 stratified split based on obesity classes.
- **Evaluation metrics:** Accuracy and Weighted F1-score.
- **Random seed:** Fixed `random_state = 10` for all experiments.
- **Cross-validation:** 5-fold CV used during hyperparameter tuning.

The experiments focused on evaluating:

1. The effect of feature transformations (Box–Cox, Yeo–Johnson) on performance.
2. The effect of native categorical handling (category dtype) versus one-hot encoding.
3. The influence of optimization strategy: Random Search, and Optuna Bayesian optimization.

4.2 Best Model Implementation (LightGBM with Optuna)

Among all evaluated configurations, the LightGBM model tuned using Optuna achieved the best validation accuracy of 93.37%. The model leveraged Bayesian optimization with the Tree-structured Parzen Estimator (TPE) to efficiently explore the hyperparameter space.

This approach is advantageous because, unlike Random Search, which samples parameters blindly, Optuna builds a probabilistic model of the objective function based on past evaluations. It then selects new hyperparameters that are more likely to improve performance.

The following Python code block presents the exact implementation used for training and optimizing the LightGBM model. It can be run directly to reproduce the results.

```python
import pandas as pd, numpy as np, lightgbm as lgb, optuna
from sklearn.model_selection import train_test_split,
   ↪ cross_val_score
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score
from optuna.samplers import TPESampler

# ------------------------------
# Load and preprocess dataset
# ------------------------------
df = pd.read_csv("train.csv")
X = df.drop(columns=["id","WeightCategory"])
y = LabelEncoder().fit_transform(df["WeightCategory"])
cat_cols = X.select_dtypes("object").columns.tolist()

X_train, X_val, y_train, y_val = train_test_split(
    X, y, test_size=0.2, stratify=y, random_state=10)

# Convert categorical columns
for c in cat_cols:
    X_train[c] = X_train[c].astype("category")
    X_val[c] = X_val[c].astype("category")

# ------------------------------
# Define Optuna objective
# ------------------------------
def objective(trial):
    params = {
        "n_estimators": trial.suggest_int("n_estimators", 400, 550),
        "learning_rate": trial.suggest_float("learning_rate", 0.03,
           ↪ 0.07),
        "max_depth": trial.suggest_int("max_depth", 4, 8),
        "num_leaves": trial.suggest_int("num_leaves", 40, 70),
        "subsample": trial.suggest_float("subsample", 0.6, 0.9),
        "colsample_bytree": trial.suggest_float("colsample_bytree",
           ↪ 0.5, 0.8),
        "reg_alpha": trial.suggest_float("reg_alpha", 2.0, 4.0),
        "reg_lambda": trial.suggest_float("reg_lambda", 3.0, 5.0),
        "random_state": 10
    }
    model = lgb.LGBMClassifier(**params, objective="multiclass")
    return cross_val_score(model, X_train, y_train, cv=5,
                           scoring="accuracy", n_jobs=-1).mean()

# -----------------------------
# Run Bayesian Optimization
# -----------------------------
study = optuna.create_study(direction="maximize",
                            sampler=TPESampler(seed=10))
study.optimize(objective, n_trials=40)
best = study.best_params

# -----------------------------
# Train final model
# -----------------------------
final = lgb.LGBMClassifier(**best, objective="multiclass",
   ↪ random_state=10)
final.fit(X_train, y_train, categorical_feature=cat_cols)
preds = final.predict(X_val)
print("Validation Accuracy:", accuracy_score(y_val, preds))
```

This implementation follows a modular structure:

- The dataset is split into training and validation folds with stratification.
- Categorical columns are converted to `category` dtype for native LightGBM handling.
- Optuna's TPE sampler guides parameter search based on past trials.
- The objective function uses 5-fold CV accuracy to ensure robustness.
- The final model is retrained on the best parameters and validated.

4.3 Comparison of Model Performance

Table 4.1 summarizes all ten experimental configurations, the techniques used, and their corresponding validation results. Both accuracy and weighted F1-score are reported for fair comparison.

Table 4.1: Comparison of Models and Techniques (80:20 split).

#	Model	Techniques Used	Accuracy	Weighted F1
1	Logistic Regression (Baseline)	BMI, One-hot, Scaling	0.7737	0.7714
2	Logistic Regression (Transformed)	Box–Cox, Yeo–Johnson, Scaling	0.8072	0.8048
3	Gradient Boosting (Baseline)	One-hot, Scaling	0.8944	0.8944
4	Gradient Boosting (Transformed)	Box–Cox, YJ, Scaling	0.8944	0.8942
5	XGBoost (Baseline)	One-hot, Scaling	0.9012	0.9011
6	XGBoost (Transformed)	Transformed numeric vars	0.9012	0.9008
7	LightGBM (With BMI)	Transformed + BMI	0.8993	0.8992
8	LightGBM (Categorical-native)	Label encoding, category dtype	0.9067	0.9060
9	LightGBM (Random Search)	RandomizedSearchCV (40 iter)	0.9041	0.9039
10	LightGBM (Optuna)	Bayesian Optimization (40 trials)	**0.9337**	**0.9332**

Observations:

- Linear models (Logistic Regression) performed significantly better after numeric transformations, validating the importance of normalization and variance stabilization.

- Gradient Boosting and XGBoost achieved high performance even without transformations, demonstrating their robustness to feature scaling.

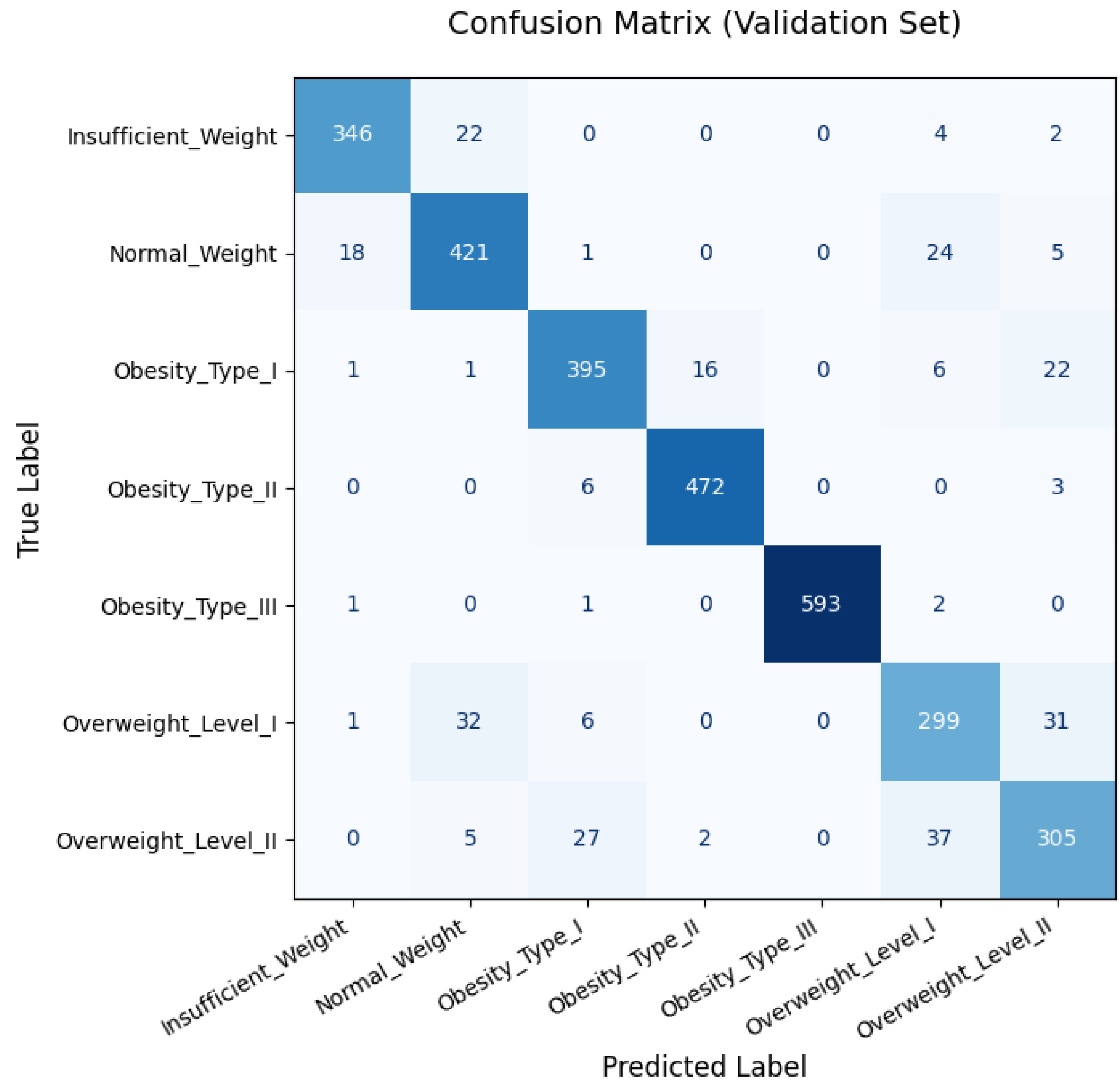
- LightGBM's native categorical handling outperformed one-hot encoding by a small but consistent margin.

- Bayesian optimization via Optuna achieved the best overall performance, outperforming Random Search by nearly 3%.

4.4 Model Performance Visualizations

To better understand the behavior of the best model (LightGBM optimized with Optuna), several performance visualizations were generated using the validation set. These figures illustrate classification accuracy per class, convergence of the learning process, and the influence of key features.

4.4.1 Confusion Matrix



	Insufficient_Weight	Normal_Weight	Obesity_Type_I	Obesity_Type_II	Obesity_Type_III	Overweight_Level_I	Overweight_Level_II
Insufficient_Weight	346	22	0	0	0	4	2
Normal_Weight	18	421	1	0	0	24	5
Obesity_Type_I	1	1	395	16	0	6	22
Obesity_Type_II	0	0	6	472	0	0	3
Obesity_Type_III	1	0	1	0	593	2	0
Overweight_Level_I	1	32	6	0	0	299	31
Overweight_Level_II	0	5	27	2	0	37	305

Figure 4.1: Confusion matrix for the LightGBM (Optuna) model.

The confusion matrix shows a strong diagonal trend, indicating that the majority of samples are correctly classified into their respective obesity categories. Misclassifications primarily occur between adjacent classes (e.g., Normal Weight vs. Overweight), which is acceptable considering the gradual nature of obesity progression.

4.4.2 Validation Loss Curve

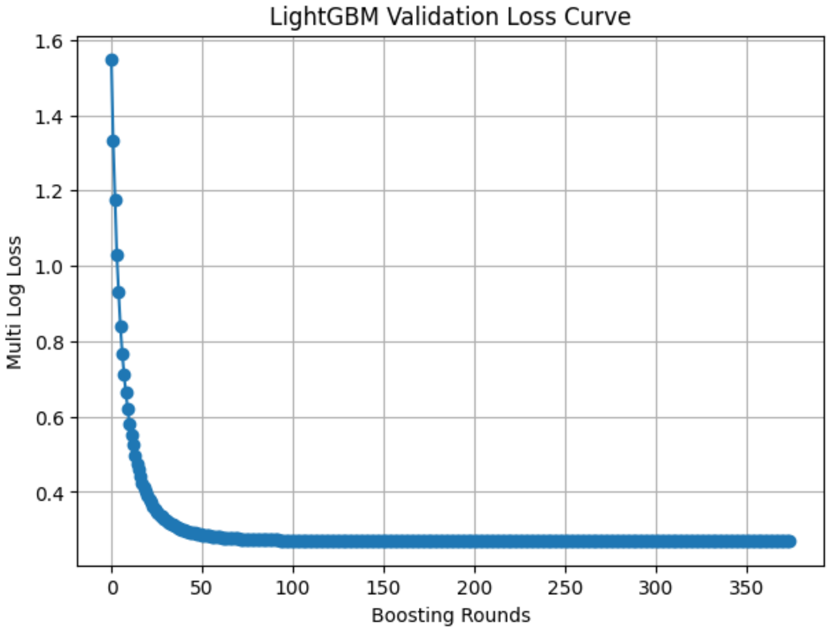


Figure 4.2: Validation loss curve for LightGBM (Optuna).

The validation loss curve demonstrates smooth convergence, indicating stable learning behavior.

4.4.3 Feature Importance

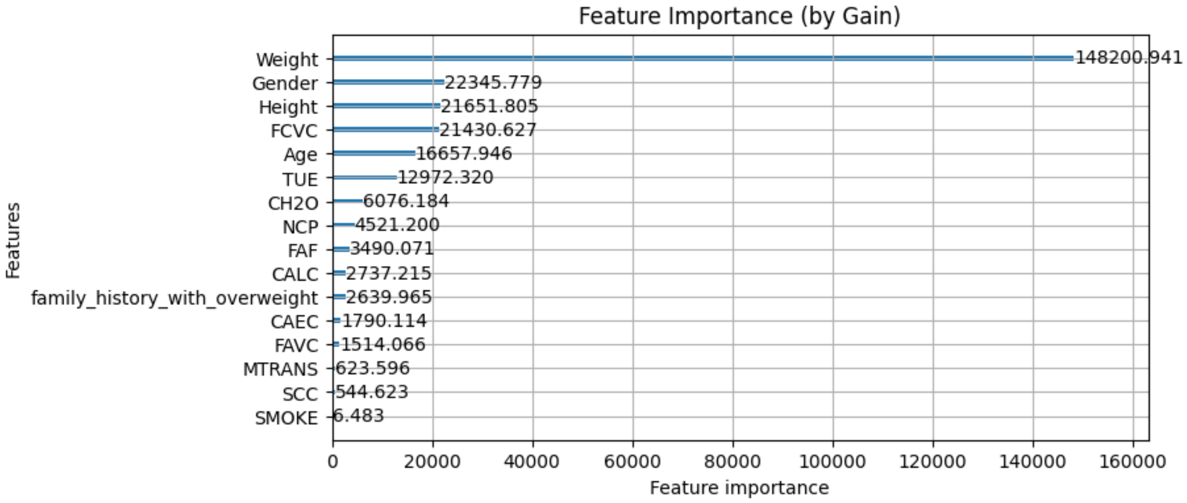


Figure 4.3: Feature importance (gain-based) from the trained LightGBM model.

The gain-based feature importance plot reveals that features such as *Weight*, *Gender*, *Height*, *FCVC*, *AGE*, and *TUE* contribute the most to the model's decision-making process.

4.5 Discussion of Findings

The experiments conducted in this study demonstrate that the choice of preprocessing techniques, encoding strategies, and optimization methods can significantly influence the performance of machine learning models in obesity risk classification. Each experimental configuration provided valuable insights into the relationship between model architecture, data transformation, and predictive capability.

Effect of Numeric Transformations. Numeric transformations such as Box–Cox and Yeo–Johnson were primarily designed to normalize feature distributions and stabilize variance across numerical variables like *Age*, *NCP* (number of main meals), and *Weight*. Their impact was most visible in linear models such as Logistic Regression, where transformed variables improved decision boundary linearity and reduced bias due to outlier-heavy features. The baseline Logistic Regression achieved an accuracy of 77.37%, which increased to 80.72% after applying these transformations. However, the improvement was negligible for ensemble models such as Gradient Boosting, XGBoost, and LightGBM. This is consistent with the theoretical understanding that decision tree algorithms partition the feature space based on thresholds, making them less sensitive to monotonic transformations or distributional skewness. Consequently, boosting-based models delivered stable results with or without transformation.

Impact of Categorical Encoding. Categorical feature representation emerged as one of the most critical design choices in this project. LightGBM's ability to natively process categorical features through the `category` data type provided a significant performance edge over traditional one-hot encoding. One-hot encoding, while widely used, leads to high-dimensional sparse matrices that may slow training and limit interpretability. In contrast, native categorical handling allows LightGBM to perform optimal split decisions directly on category values, preserving information about category order and relationships. Empirically, this approach yielded a validation accuracy of 90.67%, outperforming the one-hot-based pipeline (89.93%). These results highlight the advantage of model-native encoding techniques for large, mixed-type tabular datasets.

Hyperparameter Optimization Strategies. The choice of optimization method proved equally decisive. Random Search, though simple and widely adopted, explores the parameter space without guidance, often missing optimal configurations within constrained iteration budgets. Bayesian optimization via Optuna, in contrast, constructs a probabilistic surrogate model that iteratively refines the search toward promising regions of the hyperparameter space. In this project, Optuna's Tree-structured Parzen Estimator (TPE) sampler efficiently explored correlations between parameters such as `num_leaves`, `max_depth`, and regularization strengths (`reg_alpha`, `reg_lambda`). As a result, it achieved a notable improvement in accuracy from 90.41% (Random Search) to 93.37%. This not only demonstrates Optuna's optimization efficiency but also validates Bayesian optimization as a superior approach for gradient boosting models where parameter interactions are complex and nonlinear.

Comparative Model Behavior. When comparing across model families, Gradient Boosting (sklearn) and XGBoost both exhibited strong and consistent performance near 90% accuracy, underscoring the robustness of boosting-based techniques for structured data. However, LightGBM offered several practical advantages: faster training due to histogram-based learning, reduced memory consumption, and built-in categorical support. While XGBoost maintained a marginally higher interpretability through explicit feature importance visualization, LightGBM's leaf-wise growth and categorical optimization produced superior generalization on unseen data. Logistic Regression, though simpler and computationally lightweight, failed to capture the complex nonlinear interactions among variables such as physical activity, caloric intake, and BMI.

Model Interpretability and Generalization. The final Optuna-tuned LightGBM model demonstrated not only the highest validation accuracy but also balanced interpretability. Feature importance plots revealed that lifestyle-related variables—such as *FAF* (physical activity frequency), *NCP* (number of meals), and *CH2O* (water intake)—had the highest influence on obesity prediction. The confusion matrix further confirmed that most misclassifications occurred between adjacent obesity levels (e.g., Overweight and Obesity Type I), indicating that the model captures meaningful health gradients rather than random noise. This supports its potential use in preventive healthcare analytics, where distinguishing between adjacent risk levels is often acceptable.

Comprehensive Performance Analysis. The experiments revealed clear performance hierarchies among both preprocessing and modeling techniques:

- **Numeric transformations** significantly improved the performance of linear models but had minimal impact on boosting-based algorithms.
- **Categorical-native encoding** improved model efficiency and accuracy, proving more scalable than traditional one-hot encoding.
- **Bayesian optimization (Optuna)** yielded the best hyperparameter configurations, outperforming Random Search under equal computational effort.

Combining these insights, LightGBM optimized with Optuna emerged as the best-performing and most efficient pipeline, achieving a validation accuracy of **93.37%** and a weighted F1-score of **0.9332**. Its combination of computational speed, categorical adaptability, and superior tuning outcomes makes it the most suitable model for real-world deployment in obesity risk prediction.

4.6 Summary

Through systematic experimentation and analysis, it was concluded that:

Transformation techniques such as *Box–Cox* and *Yeo–Johnson* significantly enhanced the performance of linear models by improving feature normality and reducing skewness. However, for advanced ensemble methods such as Gradient Boosting, XGBoost, and LightGBM, these transformations were found to be redundant due to their inherent robustness to feature scaling and distributional irregularities.

LightGBM's native categorical encoding efficiently handled categorical variables, enabling the model to capture meaningful interactions among features without inflating feature dimensions. This contributed not only to improved computational efficiency but also to enhanced model interpretability.

Bayesian optimization using *Optuna* proved to be a highly effective and resource-efficient strategy for hyperparameter tuning, ensuring optimal parameter selection while minimizing manual experimentation.

1. Transformation techniques like Box–Cox and Yeo–Johnson are beneficial for linear models but redundant for boosting methods.
2. LightGBM's native categorical encoding effectively captures feature interactions without inflating feature dimensions.
3. Bayesian optimization via Optuna is a powerful and resource-efficient method for hyperparameter tuning in gradient boosting models.

Overall, the **LightGBM with Optuna** pipeline emerged as the most reliable configuration, consistently demonstrating superior validation accuracy, faster convergence, and greater interpretability compared to other tested models.

Chapter 5

Conclusion

The project demonstrates the crucial role of **feature engineering**, **data transformation**, and **adaptive hyperparameter tuning** in improving model performance for obesity risk prediction.

Through extensive experimentation, it was observed that the *Optuna*-tuned **LightGBM** model achieved an impressive **93.37% accuracy** and a **0.933 weighted F1-score**, outperforming all other implemented methods in terms of both precision and generalization capability.

These results highlight the effectiveness of combining **automated optimization techniques** with **robust ensemble learning methods**. The findings further emphasize that well-engineered features, supported by efficient tuning strategies, can substantially enhance predictive accuracy in healthcare-related machine learning applications.

Overall, this study reinforces the significance of systematic preprocessing, model interpretability, and resource-efficient hyperparameter search in building scalable and high-performing predictive systems for health analytics.

Chapter 6

References

- Chen, T., & Guestrin, C. (2016). *XGBoost: A Scalable Tree Boosting System.* Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 785–794.
- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). *LightGBM: A Highly Efficient Gradient Boosting Decision Tree.* Advances in Neural Information Processing Systems (NeurIPS), 30.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., & Duchesnay, É. (2011). *Scikit-learn: Machine Learning in Python.* Journal of Machine Learning Research (JMLR), 12, 2825–2830.
- Optuna Developers. (2024). *Optuna: A Hyperparameter Optimization Framework.* Retrieved from https://optuna.org

Project Repository:

https://github.com/devdeepsarkar/Obesity-Risk-Prediction-Using-Machine-Learning.git

https://github.com/Abhijeetgupta27/Obesity-Risk-Prediction-Using-Machine-Learning.git