Obesity Risk Prediction: Model Development Documentation

1. Introduction

Objective: Develop a classification model to predict obesity risk levels ("NObeyesdad") using health and lifestyle features.
Dataset: Training data (train.csv) and test data (test.csv) containing features like age, dietary habits, physical activity, and BMI-related metrics.

2. Data Loading & Initial Exploration

Approach

- Loaded train.csv and test.csv using pd.read_csv().
- Verified data integrity by checking dimensions, missing values, and data types.

Key Steps & Outputs

1. **Dimensions:**

- Training data: (20758, 18) rows and columns.
- Test data: (13840, 17) rows and columns.

2. **Missing Values:**

- No missing values found in training or test datasets.

3. **Data Types:**

- Identified categorical (Gender, FAVC, CAEC) and numerical (Age, Height, Weight) features.

3. Data Visualization & Feature Engineering

Approach

- **Visualization**: Analyzed target distribution, feature correlations, and relationships between variables.
- **Feature Engineering:** Created new features to improve model performance.

Key Steps & Outputs

1.Target Distribution:

- Used sns.countplot() to visualize NObeyesdad classes.

2.Categorical Features:

- Plotted relationships between Gender, FAVC (Frequent Caloric Consumption), and the target

3.Numerical Features:

- Generated histograms for Age, Height, and Weight.
- Used sns.boxplot() to compare Age across obesity classes.

4.New Features:

- Created a new feature **BMI** using the **height** and **weight** columns.

4. Data Preprocessing

Approach

- **Categorical Encoding**: Converted text categories to numerical labels.
- **Scaling**: Standardized numerical features to normalize their ranges.

Key Steps & Outputs

1. **Label Encoding:**
   - Mapped categorical variables (e.g., Gender: Male → 0, Female → 1).
2. **Target Variable Encoding**:
   - Mapped `NObeyesdad` classes to numerical labels (e.g., `Obesity_Type_I` → 3).
3. **Standard Scaling**:
   - Applied `StandardScaler` to numerical features like `Age` and `Height`.

5. Model Training & Evaluation

Approach

- **Algorithms:** Tested XGBoost, LightGBM, Decision Tree, and an ensemble**.**
- **Hyperparameter Tuning:** Used Optuna for automated optimization.

Key Steps & Outputs

1. **XGBoost with Optuna:**
   - Best parameters: max_depth=7, learning_rate=0.1.
   - Validation accuracy: 90.6%.

2. **LightGBM with Optuna:**
   - Best parameters: num_leaves=31, min_child_samples=20.
   - Validation accuracy: 90.9%.
3. **Decision Tree:**
   - Achieved 84% accuracy with max_depth=10.
4. **Ensemble Model:**
   - Combined XGBoost and LightGBM predictions using majority voting.
   - Improved accuracy to 91%.
5. **Feature Importance:**

   Top features: Weight, BMI, FAF (physical activity frequency).