CHARLES DARWIN UNIVERSITY

Sydney Campus

Assessment two

A Comparative Study for Machine Learning Models for Predicting Obesity.

Submitted by:
Maheshwor Tiwari
Student No.: S365452

An assignment presented as part of the Master of Data Science's program

(SDASC2-2024)

S224 PRT565 MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

September 5, 2024

Contents

List of Figures

List of Tables

1 Problem Description

Obesity has become a global epidemic and is considered one of the most pressing public health challenges of the 21st century. According to the World Health Organization (WHO), in 2016, over 1.9 billion adults were classified as overweight, and more than 650 million were classified as obese. The prevalence of obesity has nearly tripled globally since 1975, contributing to significant public health concerns, particularly the rise in noncommunicable diseases (NCDs) such as cardiovascular diseases (CVDs), type 2 diabetes, musculoskeletal disorders, and certain types of cancer (World Health Organization, 2021). Obesity is responsible for approximately 2.8 million deaths annually due to these related conditions (World Health Organization, 2021).

Obesity arises from a combination of factors, including genetic predispositions, dietary habits, physical inactivity, and environmental influences. The increased consumption of calorie-dense foods and the reduction in physical activity due to more sedentary lifestyles have exacerbated the obesity crisis worldwide, affecting populations across various age groups and income levels (World Obesity Federation, 2022).

The objective of this project is to predict obesity levels using machine learning techniques. The dataset contains various lifestyle and demographic attributes, including age, weight, height, dietary habits, and physical activity. These factors allow for the classification of individuals into different obesity levels, such as Insufficient Weight, Normal Weight, Overweight Levels I & II, and Obesity Types I, II, & III. Early identification of obesity risks through machine learning can enable timely interventions and reduce the long-term impact of obesity-related diseases (World Obesity Federation, 2022).

2 Dataset Description

The dataset used in this project contains 17 attributes related to individuals' eating habits, physical condition, and demographic data. These attributes include both numeric variables (e.g., **age**, **weight**, **height**) and categorical variables (e.g., **family history of overweight**, **transportation methods**, etc.). More details can be found at (*Estimation of Obesity Levels Based On Eating Habits and Physical Condition [Dataset]*, 2019).

Key attributes include:

- **Age**: Age of the individual.
- **Height**: Height in meters.
- **Weight**: Weight in kilograms.
- **favc**: Whether the individual frequently consumes high-calorie foods.
- **ncp**: Number of main meals per day.
- **caec**: Whether the individual consumes food between meals.
- **nobeyesdad**: The target variable, indicating the obesity level.

3 Choice of Algorithm

Three machine learning algorithms were selected to predict obesity levels based on individuals' lifestyle data:

3.1 Logistic Regression

Logistic Regression is effective for predicting obesity levels by modeling the probability of each class (e.g., Normal Weight, Obesity Types I-III) from the input features. It is used here as a baseline model to compare with more complex algorithms.

3.2 Decision Tree Classifier

Decision Trees are ideal for handling both categorical and numeric data, making them suitable for identifying patterns between lifestyle factors like food habits and obesity. The model's interpretability helps visualize how decisions are made based on key features.

3.3 Random Forest Classifier

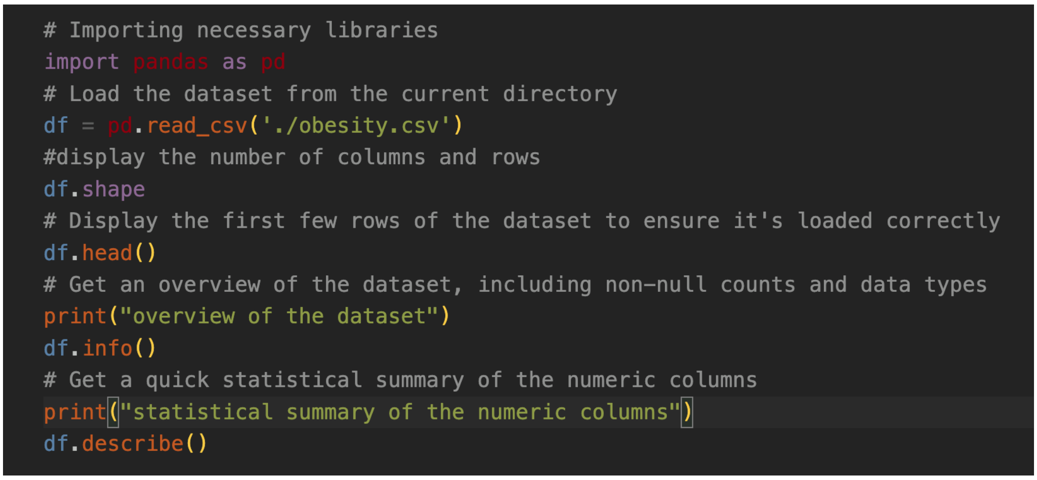
Random Forest enhances the accuracy and stability of predictions by averaging multiple decision trees. It helps reduce overfitting and handles complex interactions between features, ensuring robust predictions of obesity levels across varied data.

4 Description of Key Steps

4.1 Understanding the Data

Data exploration and cleaning involved the following steps:

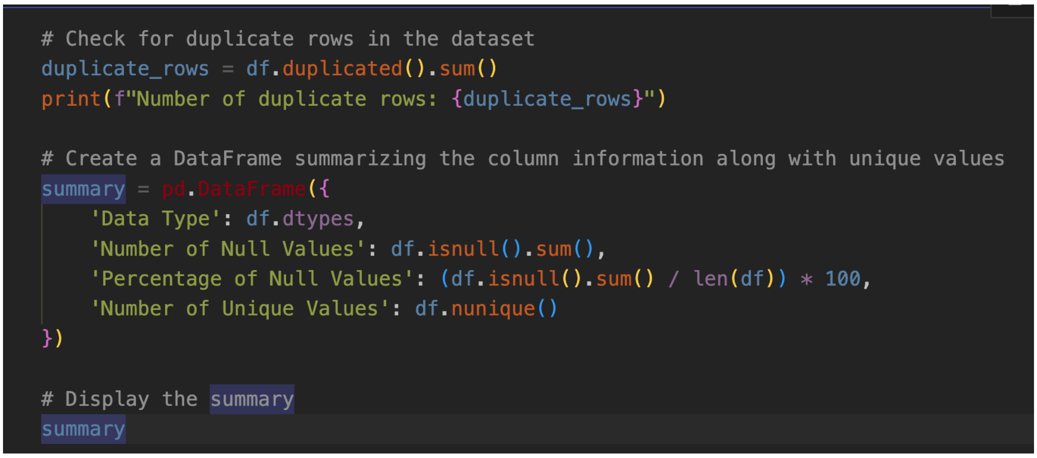
- The dataset was inspected using `head()`, `info()`, and `describe()` to understand its structure, data types, and summary statistics.



```python
# Importing necessary libraries
import pandas as pd
# Load the dataset from the current directory
df = pd.read_csv('./obesity.csv')
#display the number of columns and rows
df.shape
# Display the first few rows of the dataset to ensure it's loaded correctly
df.head()
# Get an overview of the dataset, including non-null counts and data types
print("overview of the dataset")
df.info()
# Get a quick statistical summary of the numeric columns
print("statistical summary of the numeric columns")
df.describe()
```

Figure 1: Dataset inspection using `head()`, `info()`, and `describe()`

- Missing values were checked, and duplicates were removed to ensure data integrity.



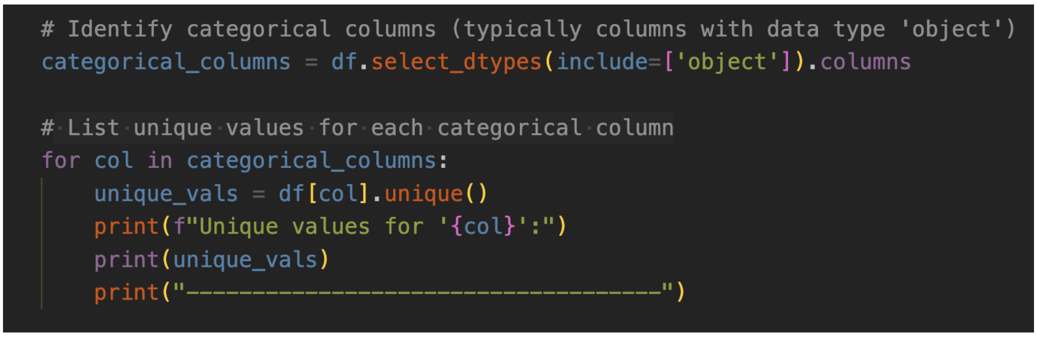
```python
# Check for duplicate rows in the dataset
duplicate_rows = df.duplicated().sum()
print(f"Number of duplicate rows: {duplicate_rows}")

# Create a DataFrame summarizing the column information along with unique values
summary = pd.DataFrame({
    'Data Type': df.dtypes,
    'Number of Null Values': df.isnull().sum(),
    'Percentage of Null Values': (df.isnull().sum() / len(df)) * 100,
    'Number of Unique Values': df.nunique()
})

# Display the summary
summary
```

Figure 2: Checking for missing values and removing duplicates

- Categorical variables were identified, and their unique values were listed to prepare for encoding. Numeric features were standardized for model compatibility.



```python
# Identify categorical columns (typically columns with data type 'object')
categorical_columns = df.select_dtypes(include=['object']).columns

# List unique values for each categorical column
for col in categorical_columns:
    unique_vals = df[col].unique()
    print(f"Unique values for '{col}':")
    print(unique_vals)
    print("------------------------------------")
```

Figure 3: Listing unique categorical values and preparing data for encoding and standardization

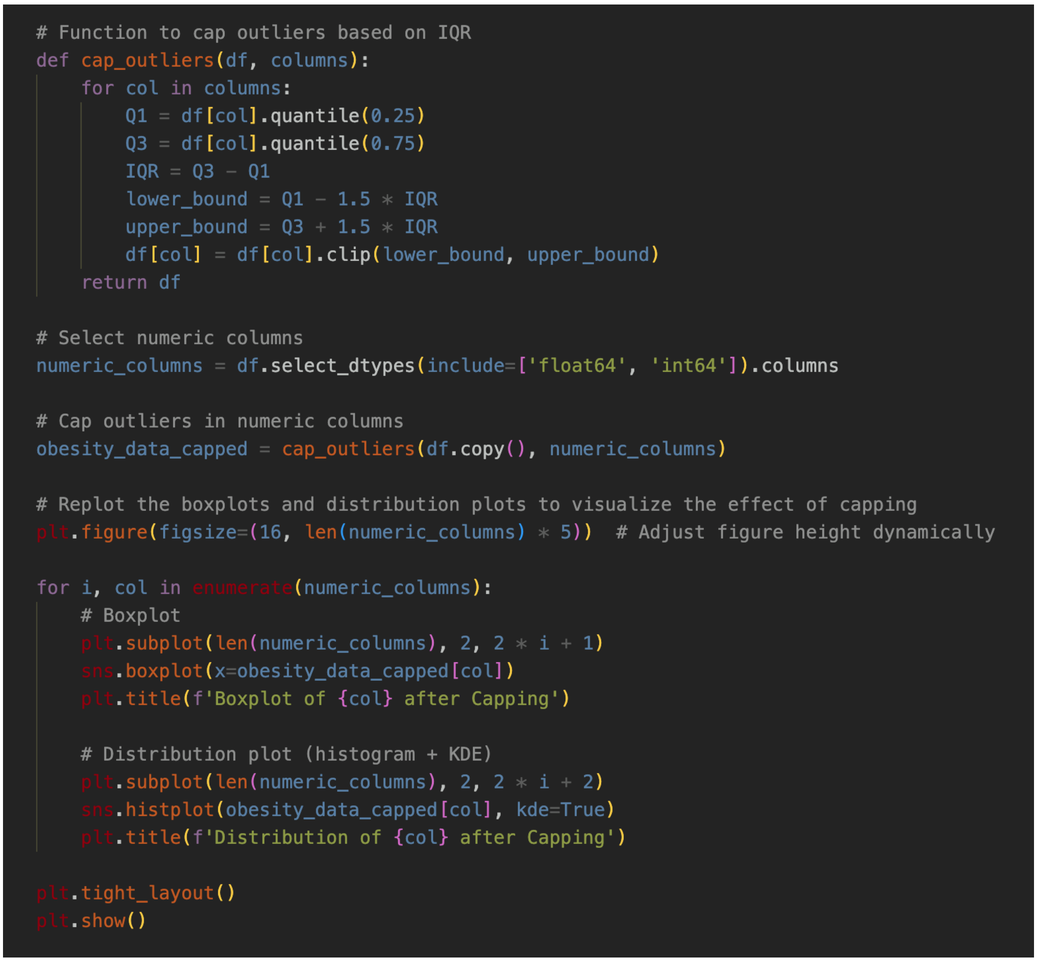
- After preprocessing the dataset, it was saved and subsequently used in building and evaluating the machine learning models.

4.2 Data Pre-processing

Data pre-processing is a crucial step in ensuring that the dataset is clean, consistent, and ready for analysis. In this project, the following actions were taken to prepare the data:

- **Handling Missing Data**: Since the dataset had no missing values, no imputation was necessary.
- **Converting Categorical Variables**: Categorical variables were encoded using Label Encoding or One-Hot Encoding, depending on the machine learning algorithm used.
- **Outlier Detection and Treatment**: Outliers in numeric variables like *weight* and *age* were capped using the Interquartile Range (IQR) method to reduce skewed results and

prevent model bias.



```python
# Function to cap outliers based on IQR
def cap_outliers(df, columns):
    for col in columns:
        Q1 = df[col].quantile(0.25)
        Q3 = df[col].quantile(0.75)
        IQR = Q3 - Q1
        lower_bound = Q1 - 1.5 * IQR
        upper_bound = Q3 + 1.5 * IQR
        df[col] = df[col].clip(lower_bound, upper_bound)
    return df

# Select numeric columns
numeric_columns = df.select_dtypes(include=['float64', 'int64']).columns

# Cap outliers in numeric columns
obesity_data_capped = cap_outliers(df.copy(), numeric_columns)

# Replot the boxplots and distribution plots to visualize the effect of capping
plt.figure(figsize=(16, len(numeric_columns) * 5))  # Adjust figure height dynamically

for i, col in enumerate(numeric_columns):
    # Boxplot
    plt.subplot(len(numeric_columns), 2, 2 * i + 1)
    sns.boxplot(x=obesity_data_capped[col])
    plt.title(f'Boxplot of {col} after Capping')

    # Distribution plot (histogram + KDE)
    plt.subplot(len(numeric_columns), 2, 2 * i + 2)
    sns.histplot(obesity_data_capped[col], kde=True)
    plt.title(f'Distribution of {col} after Capping')

plt.tight_layout()
plt.show()
```

Figure 4: Handling outliers using the IQR method.

- **Scaling Numeric Features**: Numeric features were standardized to ensure that the models performed optimally, especially for algorithms sensitive to feature scaling, like Logistic Regression.

4.3 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) was conducted to explore the dataset and identify key patterns:

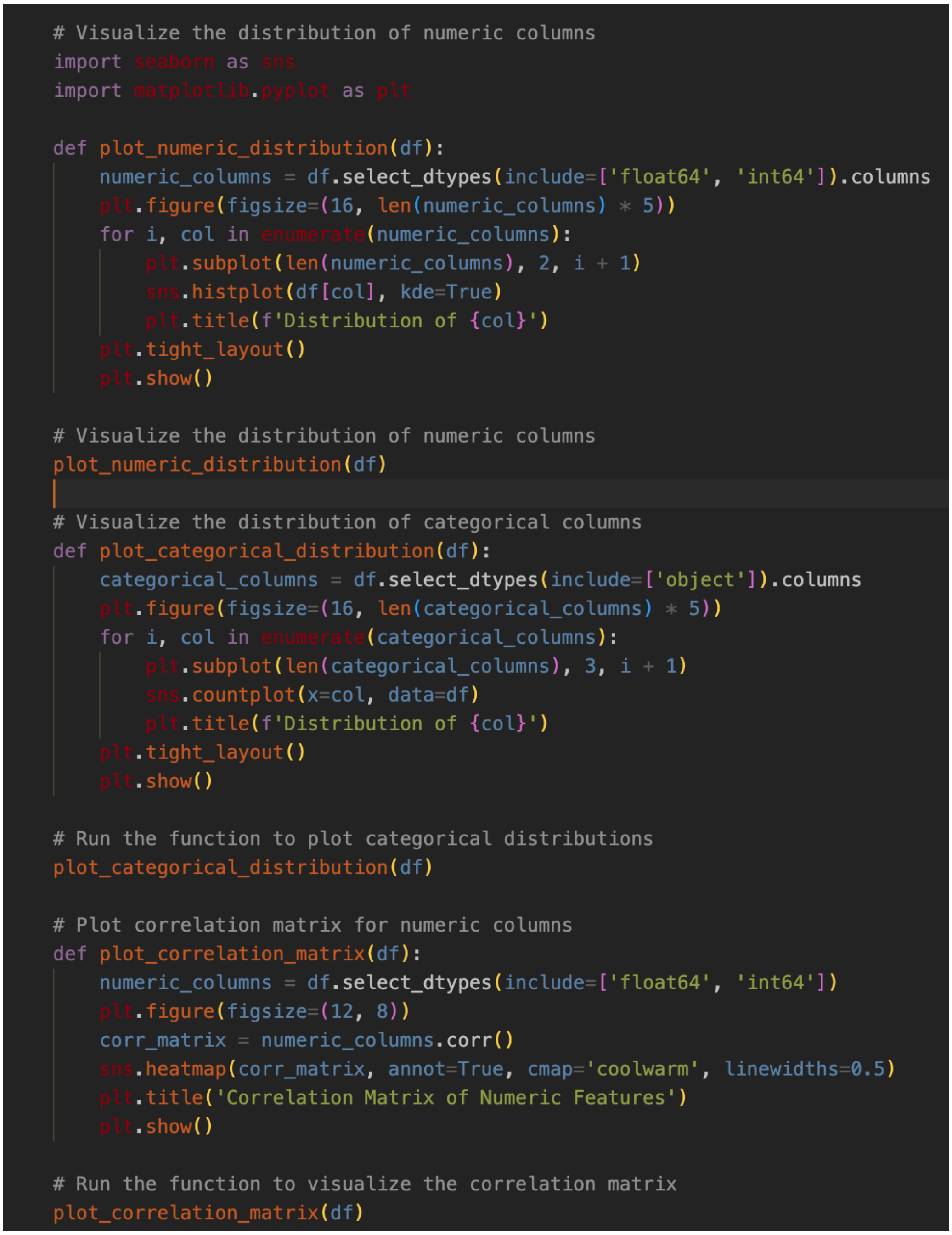
- **Numeric Feature Distribution**: Histograms and density plots showed the spread of features like *age*, *height*, and *weight*, revealing skewed distributions, especially for *weight*.

- **Categorical Feature Distribution**: Bar plots illustrated the balance of categorical features helping understand the class distributions.

- **Relationships Between Variables**: Box plots highlighted how features like *age* and *weight* varied across obesity levels, showing distinct differences between categories.

- **Correlation Analysis**: A correlation matrix identified strong relationships, especially between *weight* and obesity levels.

- **Outlier Detection**: Outliers in *age* and *weight* were identified and later capped to prevent skewing model results.



```python
# Visualize the distribution of numeric columns
import seaborn as sns
import matplotlib.pyplot as plt

def plot_numeric_distribution(df):
    numeric_columns = df.select_dtypes(include=['float64', 'int64']).columns
    plt.figure(figsize=(16, len(numeric_columns) * 5))
    for i, col in enumerate(numeric_columns):
        plt.subplot(len(numeric_columns), 2, i + 1)
        sns.histplot(df[col], kde=True)
        plt.title(f'Distribution of {col}')
    plt.tight_layout()
    plt.show()

# Visualize the distribution of numeric columns
plot_numeric_distribution(df)

# Visualize the distribution of categorical columns
def plot_categorical_distribution(df):
    categorical_columns = df.select_dtypes(include=['object']).columns
    plt.figure(figsize=(16, len(categorical_columns) * 5))
    for i, col in enumerate(categorical_columns):
        plt.subplot(len(categorical_columns), 3, i + 1)
        sns.countplot(x=col, data=df)
        plt.title(f'Distribution of {col}')
    plt.tight_layout()
    plt.show()

# Run the function to plot categorical distributions
plot_categorical_distribution(df)

# Plot correlation matrix for numeric columns
def plot_correlation_matrix(df):
    numeric_columns = df.select_dtypes(include=['float64', 'int64'])
    plt.figure(figsize=(12, 8))
    corr_matrix = numeric_columns.corr()
    sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', linewidths=0.5)
    plt.title('Correlation Matrix of Numeric Features')
    plt.show()

# Run the function to visualize the correlation matrix
plot_correlation_matrix(df)
```

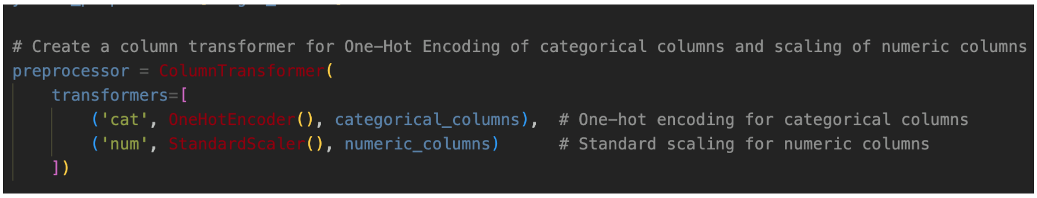
Figure 5: Code to visualize of categorical columns, correlation matrix, and numeric feature distributions.

4.4 Building the Model

In this section, we focus on training and evaluating various machine learning models to predict obesity levels based on the preprocessed dataset. Different algorithms were implemented to compare their effectiveness in predicting the target variable.

4.4.1 Logistic Regression

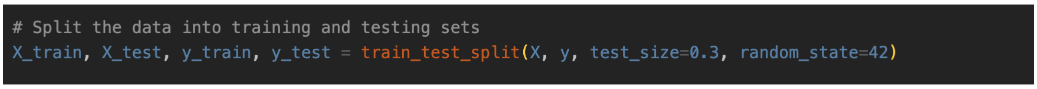
Logistic Regression was used as a baseline model to classify obesity levels. The target variable (*nobeyesdad*) represents the obesity categories, while features like *age*, *weight*, and *dietary habits* were used as inputs.



```
# Create a column transformer for One-Hot Encoding of categorical columns and scaling of numeric columns
preprocessor = ColumnTransformer(
    transformers=[
        ('cat', OneHotEncoder(), categorical_columns),  # One-hot encoding for categorical columns
        ('num', StandardScaler(), numeric_columns)      # Standard scaling for numeric columns
    ])
```

Figure 6: One-hot encoding of categorical features and standardization of numeric features.

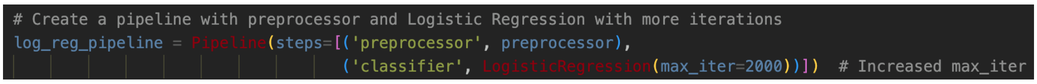
Categorical features were transformed using one-hot encoding, while numeric features were standardized. This preprocessing ensured that the Logistic Regression model could effectively capture relationships between the predictors and the target variable.The dataset was split into 70% for training and 30% for testing.



```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

Figure 7: Splitting the dataset into 70% for training and 30% for testing.

A pipeline was constructed to include both preprocessing and model training, optimizing workflow and performance. The trained model was evaluated on the test set.



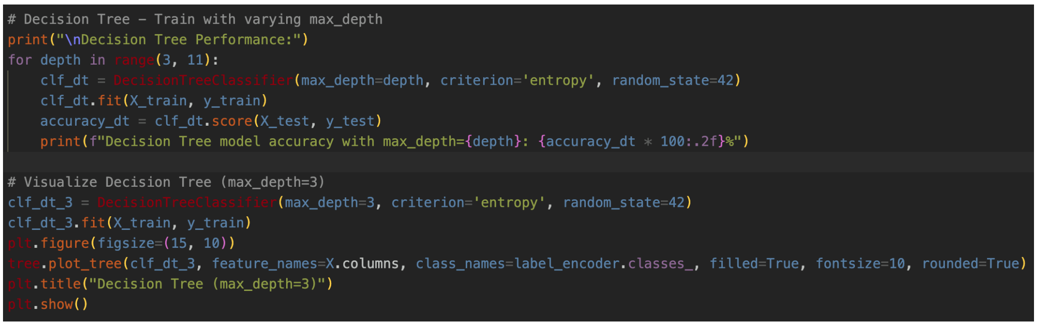
```
# Create a pipeline with preprocessor and Logistic Regression with more iterations
log_reg_pipeline = Pipeline(steps=[('preprocessor', preprocessor),
                                   ('classifier', LogisticRegression(max_iter=2000))])  # Increased max_iter
```

Figure 8: Pipeline construction for preprocessing and model training.

4.4.2 Decision Tree Classifier

The Decision Tree Classifier was used to predict obesity levels by splitting the dataset based on the most informative features. Categorical features, were label-encoded to enable the model to handle both numeric and categorical data effectively.

The dataset was divided into 70% for training and 30% for testing. The model was trained with varying depths, ranging from 3 to 10, to evaluate performance at different complexity levels.



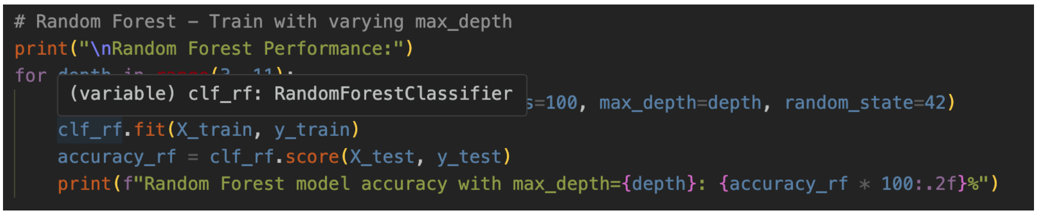
```
# Decision Tree - Train with varying max_depth
print("\nDecision Tree Performance:")
for depth in range(3, 11):
    clf_dt = DecisionTreeClassifier(max_depth=depth, criterion='entropy', random_state=42)
    clf_dt.fit(X_train, y_train)
    accuracy_dt = clf_dt.score(X_test, y_test)
    print(f"Decision Tree model accuracy with max_depth={depth}: {accuracy_dt * 100:.2f}%")

# Visualize Decision Tree (max_depth=3)
clf_dt_3 = DecisionTreeClassifier(max_depth=3, criterion='entropy', random_state=42)
clf_dt_3.fit(X_train, y_train)
plt.figure(figsize=(15, 10))
tree.plot_tree(clf_dt_3, feature_names=X.columns, class_names=label_encoder.classes_, filled=True, fontsize=10, rounded=True)
plt.title("Decision Tree (max_depth=3)")
plt.show()
```

Figure 9: Training Decision Tree with varying *max_depth* from 3 to 10.

4.4.3 Random Forest Classifier

The Random Forest algorithm was applied to enhance the accuracy of obesity level predictions. This ensemble technique constructs multiple decision trees, each trained on random subsets of the data, and combines their predictions to reduce overfitting and improve robustness.



```
# Random Forest - Train with varying max_depth
print("\nRandom Forest Performance:")
for depth in range(3, 11):
    (variable) clf_rf: RandomForestClassifier s=100, max_depth=depth, random_state=42)
    clf_rf.fit(X_train, y_train)
    accuracy_rf = clf_rf.score(X_test, y_test)
    print(f"Random Forest model accuracy with max_depth={depth}: {accuracy_rf * 100:.2f}%")
```

Figure 10: Training the Random Forest model with varying *max_depth* to improve performance.

As with the Decision Tree model, the dataset was split into 70% for training and 30% for testing. The Random Forest model was trained with varying *max_depth* values to determine the best-performing model. Feature importance analysis was also conducted, which highlighted the most influential attributes in predicting obesity levels.

5 Results

The results obtained from the analysis and model evaluations provided valuable insights into the dataset and the performance of various machine learning algorithms.

5.1 Data Exploration and Pre-processing

No missing values were found, and 24 duplicate entries were removed, leaving 2,087 unique records. The summary statistics revealed significant variability in key features such as *Weight* and *Age*, which are directly related to obesity levels. Outliers in these variables were identified and treated to avoid skewing the results. Categorical features related to lifestyle, including *family history of overweight* and *high-caloric food consumption (FAVC)*, were analyzed and prepared for modeling. These features provided crucial insights into the factors contributing to different obesity levels across individuals.

5.2 Exploratory Data Analysis (EDA)

Exploratory Data Analysis was conducted to visualize the distribution of both categorical and numeric features in the dataset, providing key insights into the data structure and the relationships between various attributes and obesity levels.

5.2.1 Visualisation of categorical and numerical variables

The EDA revealed key patterns in both categorical and numeric variables relevant to obesity prediction. Most individuals reported a family history of overweight, no smoking habits, and a balanced distribution across obesity levels (*NObeyesdad*). Numeric variables like *weight*, *age*, and *height* showed skewed distributions, particularly *weight*, where most individuals weighed between 60 and 100 kg. Behavioral factors such as *FCVC* (vegetable consumption) and *FAF* (physical activity) aligned with known contributors to obesity, providing a solid foundation for model building.

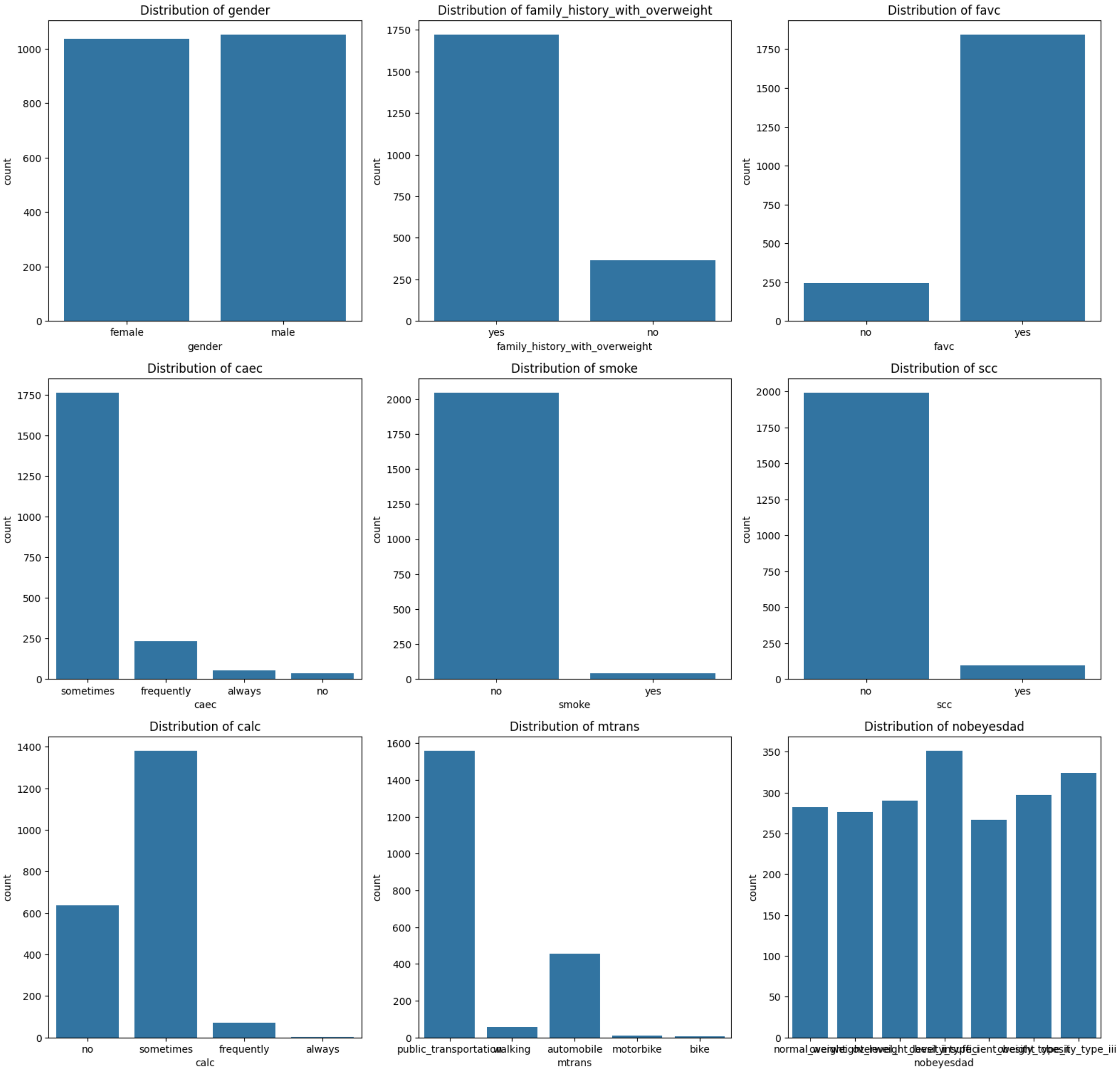


Figure 11: Distribution of categorical features.

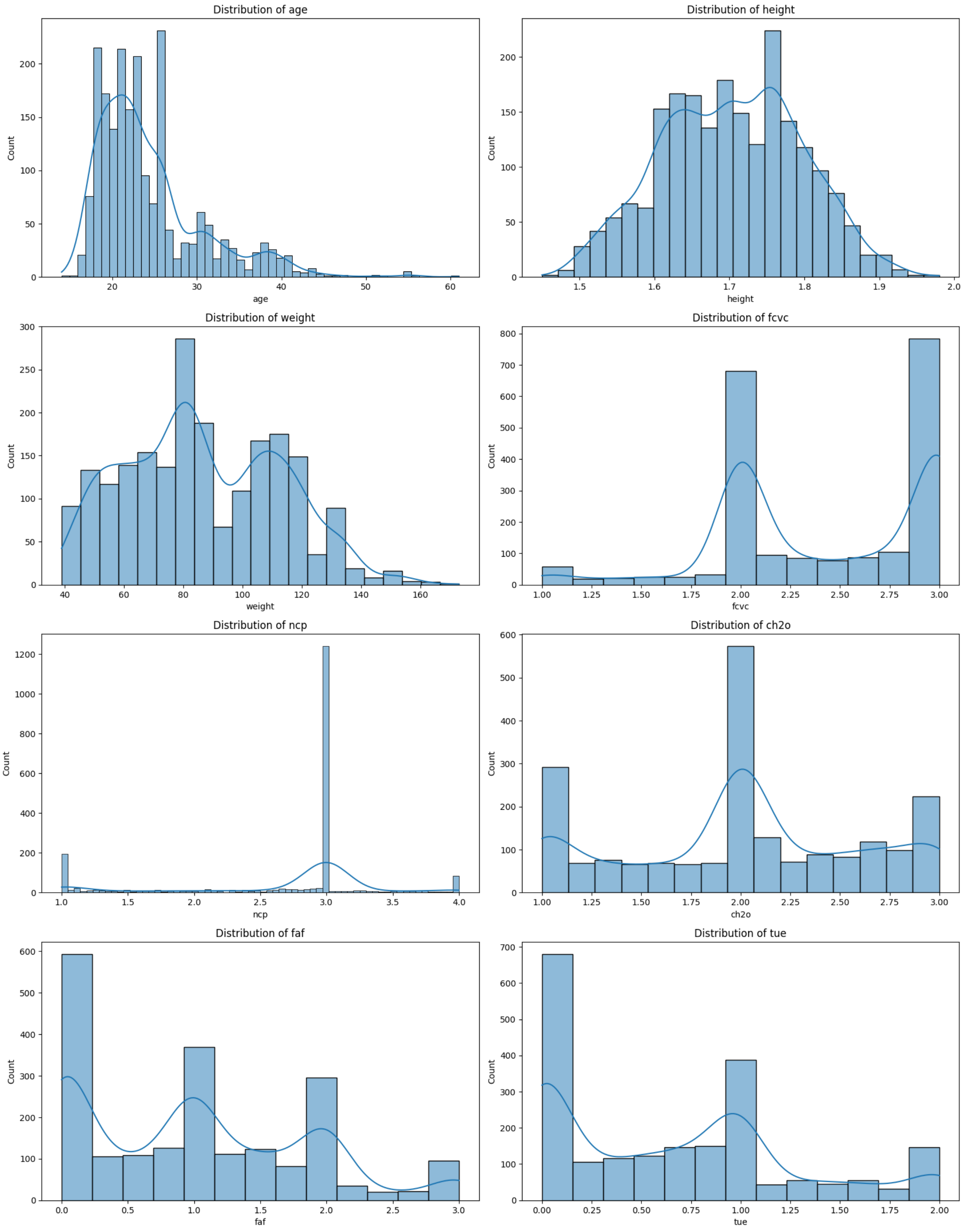


Figure 12: Distribution of numeric features

5.2.2 Correlation and Relationship Between Features

The correlation matrix in Figure 14 shows that features like *weight* and *height* exhibit a moderately positive correlation (0.46), which is expected, as taller individuals tend to weigh more. Most other correlations are weak, indicating that a diverse range of factors contributes to obesity levels. This highlights the necessity of multi-feature analysis in building a robust predictive model.

Figure 13 further explores relationships between *age*, *weight*, and *height* across obesity levels. It demonstrates that individuals in higher obesity categories (*Obesity Type I-III*) tend to have higher body weight and slightly lower heights, reinforcing the importance of these features in predicting obesity levels. These trends provide critical insights for the machine learning model, which aims to predict obesity based on lifestyle and physical attributes (Estimation of Obesity Levels Based on Eating Habits and Physical Condition Dataset, 2019).

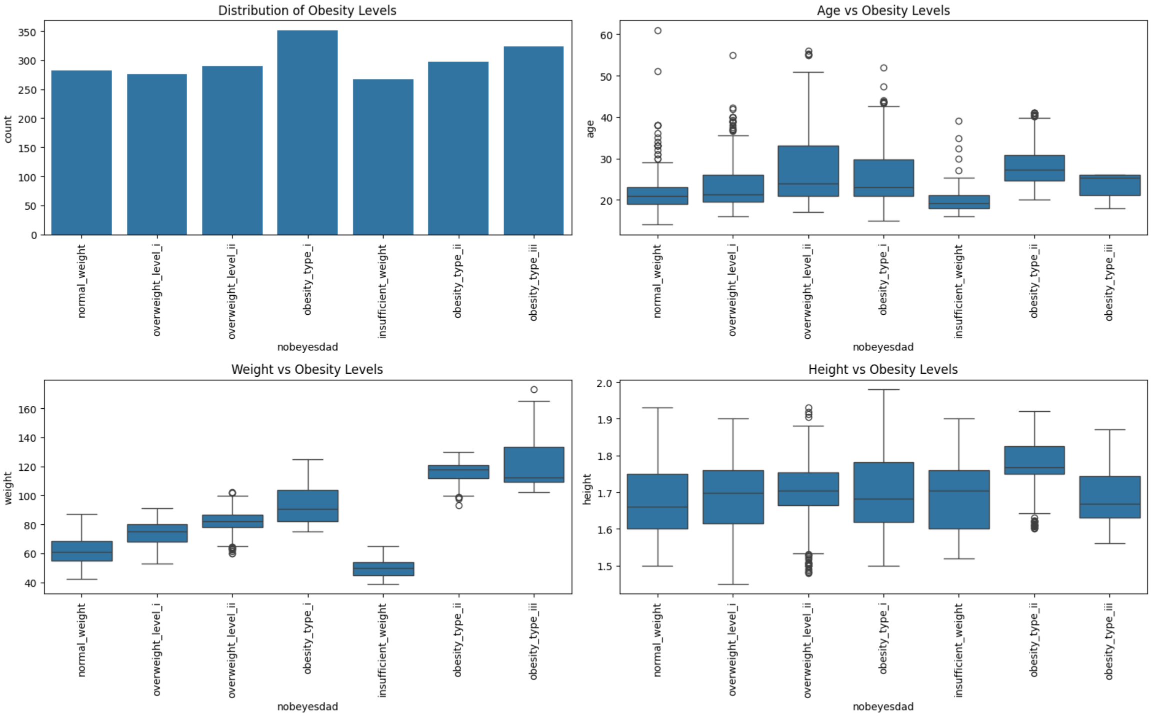


Figure 13: Exploring Relationships Between Age, Weight, Height, and Obesity Levels.

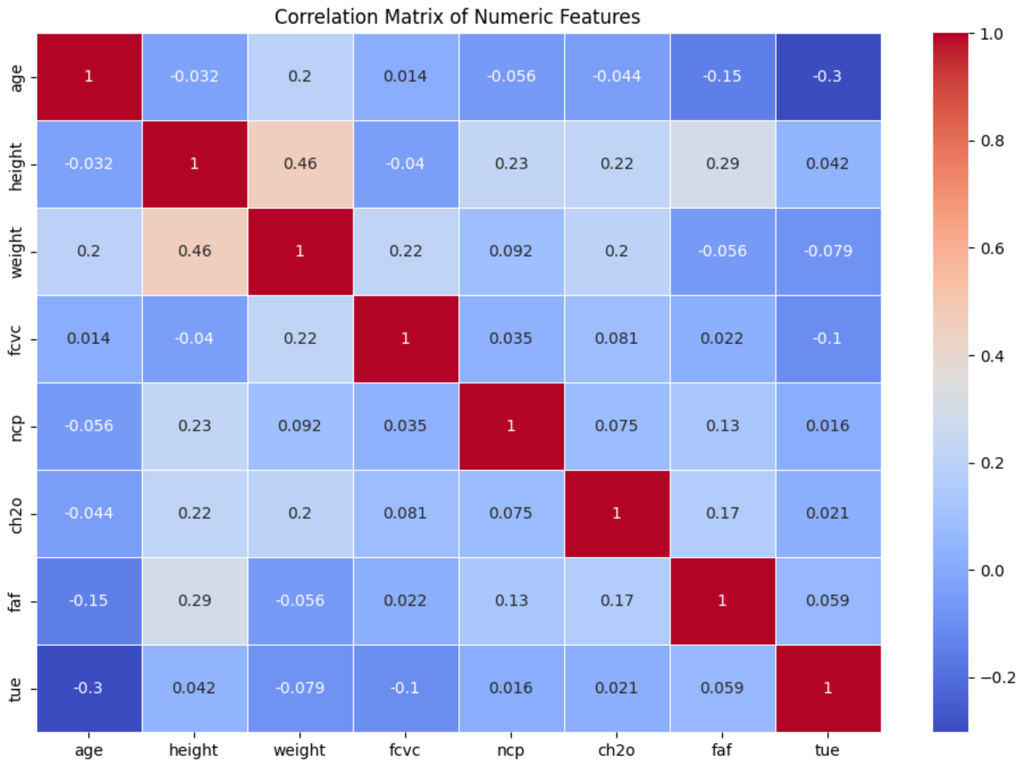


Figure 14: Correlation Matrix of Numeric Features.

5.2.3 Outlier Detection and Capping

Outliers were detected in numeric variables such as *age*, *height*, *weight*, and other behavioral features. These outliers can significantly skew the results of machine learning models, especially for algorithms sensitive to outlier values. As shown in Figure 15, outliers were identified using the Interquartile Range (IQR) method, and they were capped to fall within acceptable bounds to ensure the robustness of the model. The box plots demonstrate the effect of capping, with all extreme values now brought within reasonable limits, and the distributions adjusted accordingly in the histograms.

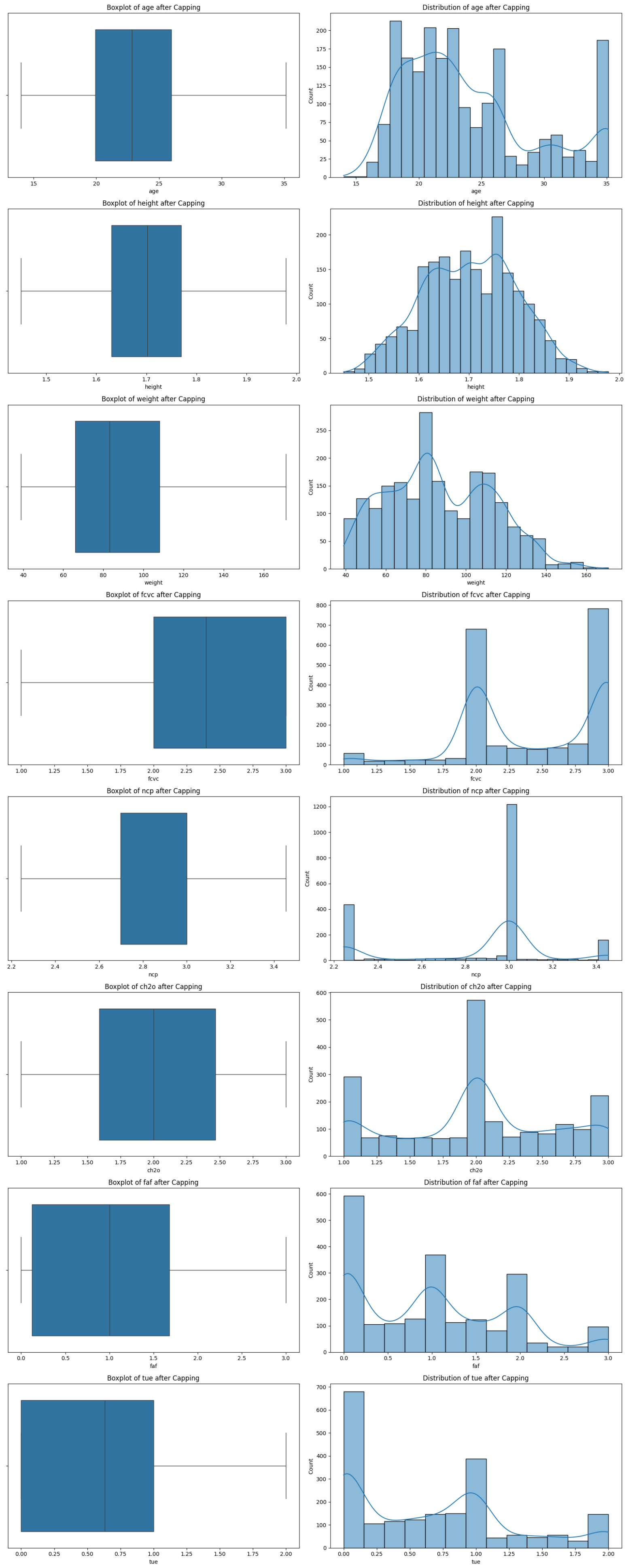

Boxplot of age after Capping
15
20
25
30
35
age
Distribution of age after Capping
Count
0
25
50
75
100
125
150
175
200
Boxplot of height after Capping
1.5
1.6
1.7
1.8
1.9
2.0
height
Distribution of height after Capping
0
50
100
150
200
Boxplot of weight after Capping
40
60
80
100
120
140
160
weight
Distribution of weight after Capping
0
50
100
150
200
250
Boxplot of fcvc after Capping
1.00
1.25
1.50
1.75
2.00
2.25
2.50
2.75
3.00
fcvc
Distribution of fcvc after Capping
0
100
200
300
400
500
600
700
800
Boxplot of ncp after Capping
2.2
2.4
2.6
2.8
3.0
3.2
3.4
ncp
Distribution of ncp after Capping
0
200
400
600
800
1000
1200
Boxplot of ch2o after Capping
ch2o
Distribution of ch2o after Capping
0
100
200
300
400
500
600
Boxplot of faf after Capping
0.0
0.5
1.0
1.5
2.0
2.5
3.0
faf
Distribution of faf after Capping
Boxplot of tue after Capping
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
tue
Distribution of tue after Capping
700

Figure 15: Boxplot and Distribution of Numeric Features After Capping Outliers.

5.3 Logistic Regression

The Logistic Regression model was used as the baseline classifier to predict obesity levels. After preprocessing, including one-hot encoding and feature scaling, the model achieved an accuracy of 85.01% on the test set.

Obesity Level	**Precision**	**Recall**	**F1-Score**	**Support**
Insufficient Weight	0.87	0.95	0.91	87
Normal Weight	0.87	0.65	0.75	95
Obesity Type I	0.93	0.88	0.90	105
Obesity Type II	0.97	0.99	0.98	95
Obesity Type III	0.98	1.00	0.99	91
Overweight Level I	0.65	0.65	0.65	80
Overweight Level II	0.64	0.80	0.71	74
Accuracy	85.01% (627 samples)			
Macro Avg	0.85	0.85	0.84	627
Weighted Avg	0.86	0.85	0.85	627

Table 1: Logistic Regression Classification Report

From the classification report, we observe that the model performs particularly well for categories like *Obesity Type I*, *Obesity Type II*, and *Obesity Type III*, with F1-scores close to or above 0.90. The model struggles slightly with categories like *Overweight Level I* and *Overweight Level II*, reflecting lower precision and recall scores. This suggests that the model may require further tuning or additional features to improve the classification of these intermediate obesity levels.

5.4 Decision Tree Classifier

The Decision Tree Classifier was employed to classify obesity levels by splitting the data based on features such as *weight*, *height*, and *ncp*. Different tree depths were evaluated to observe the accuracy of the model, as shown in Table 2.

Max Depth	**Accuracy (%)**
3	65.87
4	74.32
5	85.49
6	89.00
7	93.14
8	93.30
9	94.10
10	93.78

Table 2: Decision Tree Performance with Varying Depths

The highest accuracy of 94.10% was achieved at a maximum depth of 9. A Decision Tree with a maximum depth of 3 is shown below, illustrating the splits based on important features.

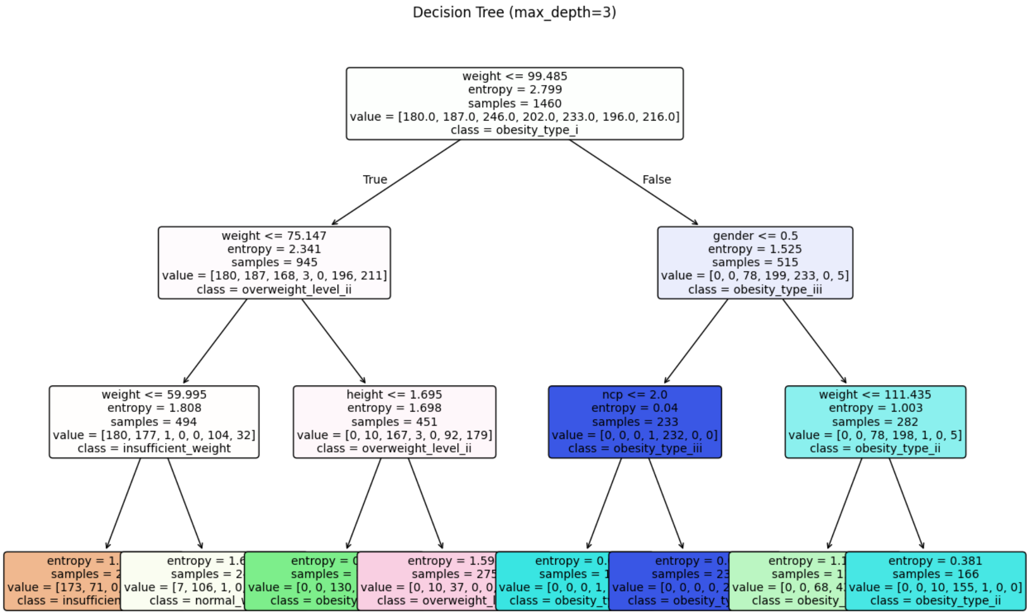


Figure 16: Decision Tree with Max Depth of 3

5.5 Random Forest Classifier

The Random Forest Classifier demonstrated improved performance over the Decision Tree by aggregating the results from multiple decision trees to enhance the prediction accuracy and robustness. Categorical variables were label-encoded to facilitate their inclusion in the model, while numeric variables were scaled. The model was trained with varying depths, ranging from 3 to 10, and the best results were observed at deeper tree levels.

Max Depth	**Random Forest Accuracy**
3	70.81%
4	82.93%
5	88.04%
6	91.87%
7	92.50%
8	93.94%
9	94.42%
10	95.22%

Table 3: Random Forest Performance with Varying Max Depths

The feature importance analysis (as shown in Figure 17) highlights that *family history of overweight* and *weight* were the most significant contributors to predicting obesity levels. This finding aligns with established research linking these factors to obesity risk. Other important features included *physical activity frequency (FAF)* and *vegetable consumption (FCVC)*. The

Random Forest model achieved an accuracy of **95.22%** at a maximum depth of 10, which was the best performance among the models tested.

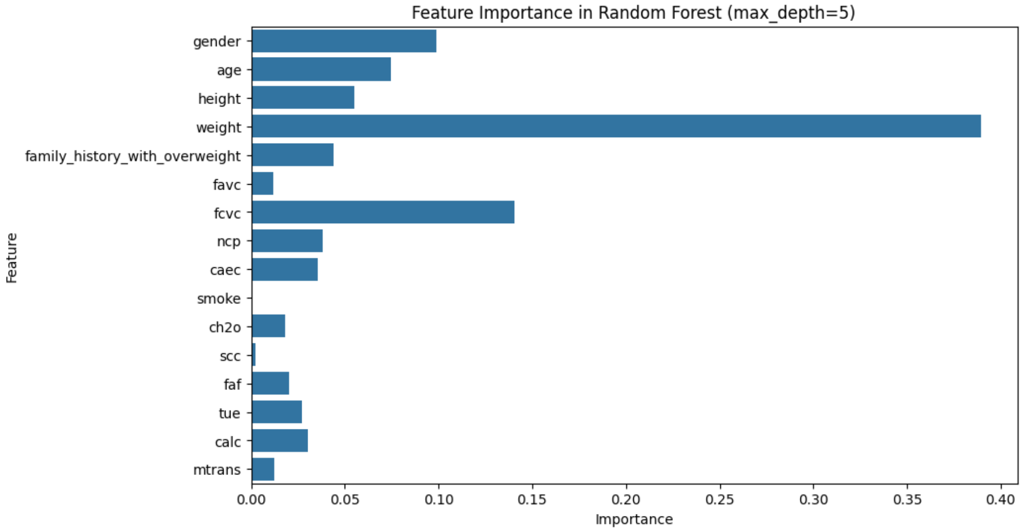


Figure 17: Feature Importance in Random Forest Classifier (max_depth=5)

6 Conclusion

This project compared three machine learning models — Logistic Regression, Decision Tree, and Random Forest — to predict obesity levels. Logistic Regression, as a baseline model, achieved an accuracy of 85.01% but showed limitations in handling complex, non-linear relationships in the dataset. The Decision Tree model, with a maximum depth of 10, improved accuracy to 94.10%, providing better interpretability but also being prone to overfitting. The Random Forest classifier, leveraging an ensemble of decision trees, achieved the highest accuracy of 95.22% with greater robustness and feature importance insights, such as family history of overweight and physical activity levels. The Random Forest model outperformed the other models in both predictive power and stability, making it the most suitable for obesity level prediction in this dataset.

References

Estimation of obesity levels based on eating habits and physical condition [dataset]. (2019). UCI Machine Learning Repository. Retrieved from https://doi.org/10.24432/C5H31Z (Accessed: 2024-09-05)

World Health Organization. (2021). *Obesity and overweight.* Accessed: 2024-09-05. (https://www.who.int/news-room/fact-sheets/detail/obesity-and-overweight)

World Obesity Federation. (2022). *World obesity atlas 2022.* Accessed: 2024-09-05. (https://www.worldobesity.org/resources/resource-library/world-obesity-atlas-2022)