Estimation of Obesity Levels Based On Eating Habits and Physical Condition

Submitted by**: Sandia Kumari**

Student Number: **501273299**

Submitted to: **Dr. Tamer Abdou**

Submission Date: July 22, 2024

Ryerson University

Table of Contents

Literature Review (Bag, 2023)

Replication Paper

Estimation of Obesity Levels Based On Eating Habits and Physical Condition . (2019). UCI Machine Learning Repository. https://doi.org/10.24432/C5H31Z.

Dataset and Preprocessing

- **Original Dataset**: Included 498 participants with a high level of class imbalance across different obesity levels.
- **SMOTE-NC Technique**: Used to eliminate class imbalance, resulting in 2009 samples and 16 features.
- **Feature Selection**: Conducted using Recursive Feature Elimination (RFE) to identify the most important obesity-related features.
- **Dataset Splitting**: Equal samples from each class were selected, 25% of data was used to form testing dataset, and 75% was used to form training dataset, resulting in a training set with 1512 samples (216 per class) and a testing set with 497 samples (71 per class). The training set was further split for hyperparameter optimization and feature selection.

Classification Models and Evaluation

- **Algorithms Used**: Logistic Regression (LR), Random Forest (RF), and Extreme Gradient Boosting (XGBoost).
- **Hyperparameter Optimization**: Performed using Bayesian optimization techniques implemented with the skopt library in Python.
- **Evaluation Metrics**: Accuracy, recall, precision, F1-score, AUC, and precision-recall curves were used to compare model performance using the scikit-learn metrics library.

Statistical Analysis

- **Software Used**: IBM SPSS Statistics version 28.0 for Windows.
- **Significance Level**: P-values < 0.05 were considered significant.
- **Power Analysis**: A post hoc power analysis revealed a power of 0.9997 with an effect size of 0.09, type I error of 0.05, and a total sample size of 498.

Results

- **Significant Associations**: Found between obesity levels and several variables, including gender, family history of overweight, food and physical activity-related factors, smoking, and transportation methods.
- **Hyperparameter Optimization**: Conducted for each model, with the best hyperparameters selected based on validation accuracy.

- **Feature Selection Impact**:
    - **LR and RF**: Achieved better accuracy with selected features compared to the full feature set.
    - **XGBoost**: Showed a slight improvement in some metrics but a deterioration in F1-score with selected features.
    - **Overall Performance**: LR demonstrated the best performance across all metrics, followed by RF with improved results using selected features. XGBoost had mixed results with a trade-off between precision and F1-score.

Conclusion

- **Model Efficiency**: Feature selection improved the efficiency and effectiveness of LR and RF models, reducing model complexity while enhancing performance.
- **Model Comparison**: LR model exhibited superior performance, making it the most effective for obesity level prediction based on the selected features, with an exceptionally high accuracy of 98.99%. RF also benefited from feature selection, whereas XGBoost had limited improvements.
- The detailed hyperparameter tuning for each model includes parameters specific to each algorithm, significantly contributing to their performance optimization.

Research Questions of the Replication Paper

I have listed the research questions from the replication paper that authors are trying to answer. I have also indicated the sections that I will replicate in my paper, and any changes that I have made during the replication.

Research Paper Questions	**Parts that I replicated**	**Different/Additional Parts done by me**
Dataset used 2009 samples and 16 features.	No	Dataset used 22,869 samples and 16 features
SMOTE	Yes	
Classification Algorithms including (Logistic Regression (LR), Random Forest (RF), and Extreme Gradient Boosting (XGBoost)	Yes	Decision Trees, Support Vector Machine
Feature Selection using RFE	No	

Hyperparameter Optimization using Bayesian optimization	Yes	Hyperparameter Tuning using GridSearch Cross Validation
Evaluation metrics including Accuracy, recall, precision, F1-score, AUC, and precision-recall curves	Yes	

Other Research Paper Review

Based on extensive literature search, there have been multiple studies using different machine learning models to identify and predict the important factors affecting obesity levels. In one study, (Rodríguez, 2021) used eight machine learning models including Decision Tree (DT), Support Vector Machines (SVM), k-nearest neighbors (KNN), Gaussian Naïve Bayes (GNB), Multi-Layer Perceptron (MLP), Random Forest (RF), Gradient Boosting (GB), and Extreme Gradient Boosting (XGB). Their research revealed RF to have superior performance with an accuracy of 77.69%, precision of 78.53%, and F1-score of 78.09%. Similarly, (Jeon, 2023) also noted in their study that RF provided greatest accuracy in predicting obesity amongst most age groups compared to other models such as SVM, Logistic Regression (LR), MLP, Light Gradient Boosting Machine (LGBM), and XGB.

However, in another study, (Ferdowsy, 2021) noted that LR provided highest accuracy of 97.09% in obesity predictions when compared to KNN, RF, MLP, SVM, GNB, DT, GB, and Adaptive Boosting (AdaBoost), while GB performed the poorest with 64.08% accuracy. Note that Ferdowsy's study used a dataset from Bangladesh, which differs from the dataset used in my project.

Another study (Barzinji, 2021) predicted the obesity rates for years 2030, 2040, and 2050, achieving high prediction accuracies of up to 99% R^2 for general predictions and up to 92% R^2 for predictions based on the Socio-Demographic Index (SDI). The models used included Multiple Linear Regression for age and sex-based predictions, Lasso Regression for SDI-based predictions, and Deep Neural Networks (DNN) for the main study predictions, implemented using TensorFlow. Note that Barzinji's study used a dataset from Global Health Data Exchange of the Institute for Health Metrics and Evaluation (USA), which differs from the dataset used in my project

Interestingly, in this study (De-La-Hoz-Correa, 2019), authors have applied SEMMA data mining methodology. Weka tool was used to apply three machine learning algorithms including Decision Trees (J48), Bayesian Networks (Naïve Bayes), and Logistic Regression (Simple Logistic). The evaluation metrics included precision, recall, true positive rate, and false positive rate, with Decision Trees (J48) demonstrating the best performance with a precision rate of 97.4%.

Lastly this (Devi, 2022) study implements Logistic Regression, Random Forest, Decision Tree, Support Vector Machine (SVM), Gradient Boosting, and Ada Boost algorithms. Additionally, ensemble methods such as Bagging and Voting Classifier are employed. Logistic Regression and SVM are further optimized using hyperparameter tuning techniques including Grid Search and Randomized Search. The results indicate that the Logistic Regression model with tuning achieves the highest accuracy of 99.68% on one dataset, while the Voting Classifier ensemble method also shows high accuracy.

Project Abstract

Obesity is a significant public health concern worldwide, linked with various chronic diseases and health risks including cardiovascular diseases, various types of cancer, diabetes mellitus amongst others. It is also a significant drain on world's economy, estimated to cost 3.6% of GDP in all countries by 2060 if current trends hold up (Bag, 2023). It is well known that nutrition, physical activity, and lifestyle greatly impacts the obesity level. Therefore, the ability to accurately estimate and classify obesity levels based on eating habits and physical activity is crucial for preventive healthcare interventions and personalized treatment strategies.

This project aims to develop machine learning models to predict obesity levels using publicly available dataset Multi-Class Prediction of Obesity Risk from Kaggle. The dataset (both train and test) that I am using is generated from a deep learning model trained on the Obesity dataset Estimation of Obesity Levels Based On Eating Habits and Physical Condition from the UCI Machine Learning Repository. Feature distributions are close to, but not exactly the same, as the original. The original dataset is collected from participants in Colombia, Peru and Mexico via web-based survey (Palechor, 2019). It contains information about eating habits, physical condition, and obesity levels of individuals. It includes attributes such as gender, age, height, weight, family history of overweight, dietary patterns, physical activity frequency and more. This dataset comprises both categorical and numerical features, making it suitable for machine learning.

Research questions of the project

1. What type of classification models were used for estimation of multi-class problem obesity levels?
2. How did each model perform in terms of accuracy, precision, recall, F1 score, AUC-ROC, and MCC?
3. Which model had the best overall performance?
4. What patterns do association rules uncover about the link between regular physical activity and maintaining a normal weight?
5. What behaviors or habits are most frequently linked to obesity type III?

GitHub Link: https://github.com/Sandia-Kumari/CIND820Project/tree/main

Understanding the data

It is a multiclass problem with 7 classes.

The dataset used for this project has following features:

- Gender: It is a categorical variable, having two values (Male/Female)
- Age: It is a numerical variable, shows age of a person.
- Height: It is a numerical variable, shows height a of person in meters.
- Weight: It is a numerical variable, shows weight of a person in kilograms.
- Family history of overweight: It is a categorical variable, shows if anyone has family history of overweight/obese, having two values (Yes/No)
- Frequently consumed high-calorie food (FAVC): It is a categorical variable, shows frequency of high calorie if a person often eats high-calorie food (yes or no).
- Frequency of consumption of vegetables (FCVC): It is an ordinal variable, shows the frequency of vegetables consumed by a person (1= never, 2= sometimes, 3= always).
- Number of main meals (NCP): It is an ordinal variable, shows number of main meals a person consumes per day (1 = between 1 & 2, 2 = three, 3 = more than 3, 4 = no answer).
- Consumption of food between meals (CAEC): It is an ordinal variable, shows frequency of food consumption between meals (1 = no, 2 = sometimes, 3 = frequently, 4 = always).
- SMOKE: It is a categorical variable, shows whether the individual smokes or not (yes or no).
- Consumption of water daily (CH2O): It is an ordinal variable, shows the consumption of water by a person per day (1 = less than a liter, 2 = between 1 and 2 L, 3 = more than 2 L).

- Monitor calorie intake (SCC): It is categorical variable, shows if a person monitors their calorie count (yes or no).
- Frequency of physical activity (FAF): It is an ordinal variable, shows frequency of physical activity of a person (1 = never, 2 = once or twice a week, 3 = two or three times a week, 4 = four or five times a week).
- Time using electronic devices (TUE): It is an ordinal variable, shows how long a person uses electronic devices (0 = none, 1 = less than an hour, 2 = between one and three hours, 3 = more than three hours).
- Consumption of alcohol (CALC): It is an ordinal variable, shows the frequency of alcohol consumption by a person (1 = no, 2 = sometimes, 3 = frequently, 4 = always).
- Type of transportation used (MTRANS): It is a categorical variable, shows the type of transportation a person uses (automobile, motorbike, bike, public transportation, walking).
- Level of obesity (NObeyesdad): It is an ordinal variable, shows the obesity level of a person according to their BMI (insufficient weight normal weight, overweight level I, overweight level II, obesity type I, obesity type II, obesity type III). It is the target variable

Exploratory Data Analysis (EDA)

These questions are related to understanding of data.

1. What is the size of training and test datasets

```
The Training dataset has 20758 rows and 18 columns
The Test dataset has 2111 rows and 17 columns
```

2. Are there any missing values in the dataset? (checking data completeness)

```
id                                0
Gender                            0
Age                               0
Height                            0
Weight                            0
family_history_with_overweight    0
FAVC                              0
FCVC                              0
NCP                               0
CAEC                              0
SMOKE                             0
CH2O                              0
SCC                               0
FAF                               0
TUE                               0
CALC                              0
MTRANS                            0
NObeyesdad                        0
```

There are no missing values in the dataset.

3. What is the data type of each column? (to understand how to preprocess different columns)

```
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   id                              20758 non-null  int64
 1   Gender                          20758 non-null  object
 2   Age                             20758 non-null  float64
 3   Height                          20758 non-null  float64
 4   Weight                          20758 non-null  float64
 5   family_history_with_overweight  20758 non-null  object
 6   FAVC                            20758 non-null  object
 7   FCVC                            20758 non-null  float64
 8   NCP                             20758 non-null  float64
 9   CAEC                            20758 non-null  object
 10  SMOKE                           20758 non-null  object
 11  CH2O                            20758 non-null  float64
 12  SCC                             20758 non-null  object
 13  FAF                             20758 non-null  float64
 14  TUE                             20758 non-null  float64
 15  CALC                            20758 non-null  object
 16  MTRANS                          20758 non-null  object
 17  NObeyesdad                      20758 non-null  object
dtypes: float64(8), int64(1), object(9)
memory usage: 2.9+ MB
```

The dataset has different data types including integer, float and object (categorical) which I will deal in later stages

4. What are the descriptive statistics for the numerical columns in the dataset?

	id	Age	Height	Weight	FCVC	NCP	CH2O	FAF	TUE
count	20758.00000	20758.000000	20758.000000	20758.000000	20758.000000	20758.000000	20758.000000	20758.000000	20758.000000
mean	10378.50000	23.841804	1.700245	87.887768	2.445908	2.761332	2.029418	0.981747	0.616756
std	5992.46278	5.688072	0.087312	26.379443	0.533218	0.705375	0.608467	0.838302	0.602113
min	0.00000	14.000000	1.450000	39.000000	1.000000	1.000000	1.000000	0.000000	0.000000
25%	5189.25000	20.000000	1.631856	66.000000	2.000000	3.000000	1.792022	0.008013	0.000000
50%	10378.50000	22.815416	1.700000	84.064875	2.393837	3.000000	2.000000	1.000000	0.573887
75%	15567.75000	26.000000	1.762887	111.600553	3.000000	3.000000	2.549617	1.587406	1.000000
max	20757.00000	61.000000	1.975663	165.057269	3.000000	4.000000	3.000000	3.000000	2.000000

Dropping id attribute as it serves solely as a unique identifier for each observation in the dataset and do not provide any meaningful information for analysis/modeling

```
df_train_d = train_df.drop(columns=["id"])
```

```
df_train_d
```

5. What is the overall structure of the dataset like no. duplicate rows, no. of unique values in each column, minimum and maximum values in each column, mean and standard deviation for numerical columns, most frequent value (mode) for categorical columns, and how often does it appear?

	Data Type	Missing	Duplicate	Unique	Min	Max	avg	Std dev	top value	Freq
Age	float64	0	0	1703	14.0	61.0	23.841804	5.688072	NaN	NaN
CAEC	object	0	0	4	Always	no	NaN	NaN	Sometimes	17529.0
CALC	object	0	0	3	Frequently	no	NaN	NaN	Sometimes	15066.0
CH2O	float64	0	0	1506	1.0	3.0	2.029418	0.608467	NaN	NaN
FAF	float64	0	0	1360	0.0	3.0	0.981747	0.838302	NaN	NaN
FAVC	object	0	0	2	no	yes	NaN	NaN	yes	18982.0
FCVC	float64	0	0	934	1.0	3.0	2.445908	0.533218	NaN	NaN
Gender	object	0	0	2	Female	Male	NaN	NaN	Female	10422.0
Height	float64	0	0	1833	1.45	1.975663	1.700245	0.087312	NaN	NaN
MTRANS	object	0	0	5	Automobile	Walking	NaN	NaN	Public_Transportation	16687.0
NCP	float64	0	0	689	1.0	4.0	2.761332	0.705375	NaN	NaN
NObeyesdad	object	0	0	7	Insufficient_Weight	Overweight_Level_II	NaN	NaN	Obesity_Type_III	4046.0
SCC	object	0	0	2	no	yes	NaN	NaN	no	20071.0
SMOKE	object	0	0	2	no	yes	NaN	NaN	no	20513.0
TUE	float64	0	0	1297	0.0	2.0	0.616756	0.602113	NaN	NaN
Weight	float64	0	0	1979	39.0	165.057269	87.887768	26.379443	NaN	NaN
family_history_with_overweight	object	0	0	2	no	yes	NaN	NaN	yes	17014.0

6. For understanding of the data, viewing first few rows of train and test dataset

Train:

	id	Gender	Age	Height	Weight	family_history_with_overweight	FAVC	FCVC	NCP	CAEC	SMOKE	CH2O	SCC	FAF	TUE	CALC	MTRANS	NObeyesdad
0	0	Male	24.443011	1.699998	81.669950	yes	yes	2.000000	2.983297	Sometimes	no	2.763573	no	0.000000	0.976473	Sometimes	Public_Transportation	Overweight_Level_II
1	1	Female	18.000000	1.560000	57.000000	yes	yes	2.000000	3.000000	Frequently	no	2.000000	no	1.000000	1.000000	no	Automobile	Normal_Weight
2	2	Female	18.000000	1.711460	50.165754	yes	yes	1.880534	1.411685	Sometimes	no	1.910378	no	0.866045	1.673584	no	Public_Transportation	Insufficient_Weight
3	3	Female	20.952737	1.710730	131.274851	yes	yes	3.000000	3.000000	Sometimes	no	1.674061	no	1.467863	0.780199	Sometimes	Public_Transportation	Obesity_Type_III
4	4	Male	31.641081	1.914186	93.798055	yes	yes	2.679664	1.971472	Sometimes	no	1.979848	no	1.967973	0.931721	Sometimes	Public_Transportation	Overweight_Level_II
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
20753	20753	Male	25.137087	1.766626	114.187096	yes	yes	2.919584	3.000000	Sometimes	no	2.151809	no	1.330519	0.196680	Sometimes	Public_Transportation	Obesity_Type_II
20754	20754	Male	18.000000	1.710000	50.000000	no	yes	3.000000	4.000000	Frequently	no	1.000000	no	2.000000	1.000000	Sometimes	Public_Transportation	Insufficient_Weight
20755	20755	Male	20.101026	1.819557	105.580491	yes	yes	2.407817	3.000000	Sometimes	no	2.000000	no	1.158040	1.198439	no	Public_Transportation	Obesity_Type_II
20756	20756	Male	33.852953	1.700000	83.520113	yes	yes	2.671238	1.971472	Sometimes	no	2.144838	no	0.000000	0.973834	no	Automobile	Overweight_Level_II
20757	20757	Male	26.680376	1.816547	118.134898	yes	yes	3.000000	3.000000	Sometimes	no	2.003563	no	0.684487	0.713823	Sometimes	Public_Transportation	Obesity_Type_II

Test:

	id	Gender	Age	Height	Weight	family_history_with_overweight	FAVC	FCVC	NCP	CAEC
0	20758	Male	26.899886	1.848294	120.644178	yes	yes	2.938616	3.000000	Sometimes
1	20759	Female	21.000000	1.600000	66.000000	yes	yes	2.000000	1.000000	Sometimes
2	20760	Female	26.000000	1.643355	111.600553	yes	yes	3.000000	3.000000	Sometimes
3	20761	Male	20.979254	1.553127	103.669116	yes	yes	2.000000	2.977909	Sometimes
4	20762	Female	26.000000	1.627396	104.835346	yes	yes	3.000000	3.000000	Sometimes

SMOKE	CH2O	SCC	FAF	TUE	CALC	MTRANS
no	2.825629	no	0.855400	0.000000	Sometimes	Public_Transportation
no	3.000000	no	1.000000	0.000000	Sometimes	Public_Transportation
no	2.621877	no	0.000000	0.250502	Sometimes	Public_Transportation
no	2.786417	no	0.094851	0.000000	Sometimes	Public_Transportation
no	2.653531	no	0.000000	0.741069	Sometimes	Public_Transportation

7. What are the basic descriptive statistics of the numerical columns in the dataset?

	id	Age	Height	Weight	FCVC	NCP	CH2O	FAF	TUE
count	20758.00000	20758.000000	20758.000000	20758.000000	20758.000000	20758.000000	20758.000000	20758.000000	20758.000000
mean	10378.50000	23.841804	1.700245	87.887768	2.445908	2.761332	2.029418	0.981747	0.616756
std	5992.46278	5.688072	0.087312	26.379443	0.533218	0.705375	0.608467	0.838302	0.602113
min	0.00000	14.000000	1.450000	39.000000	1.000000	1.000000	1.000000	0.000000	0.000000
25%	5189.25000	20.000000	1.631856	66.000000	2.000000	3.000000	1.792022	0.008013	0.000000
50%	10378.50000	22.815416	1.700000	84.064875	2.393837	3.000000	2.000000	1.000000	0.573887
75%	15567.75000	26.000000	1.762887	111.600553	3.000000	3.000000	2.549617	1.587406	1.000000
max	20757.00000	61.000000	1.975663	165.057269	3.000000	4.000000	3.000000	3.000000	2.000000

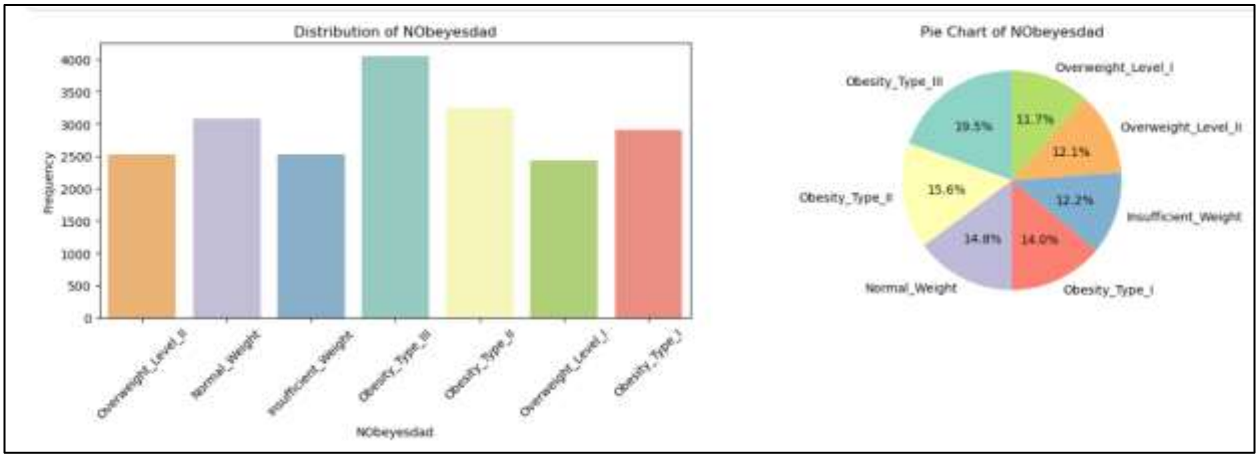
8. What are the continuous and categorical variables in the dataset?

```
Continuous Variables: ['Age', 'Height', 'Weight', 'FCVC', 'NCP', 'CH2O', 'FAF', 'TUE']
Categorical Variables: ['Gender', 'family_history_with_overweight', 'FAVC', 'CAEC', 'SMOKE', 'SCC', 'CALC', 'MTRANS']
```

Target Variable and Categorical Attributes

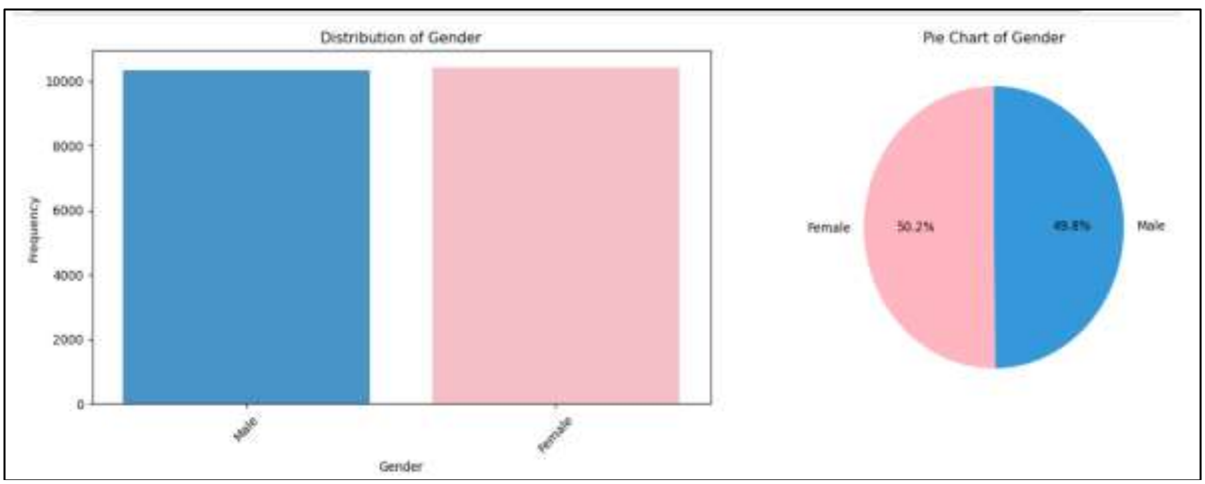
Understanding Target Variable and Categorical attributes

What is the frequency distribution of target variable (understanding target)?

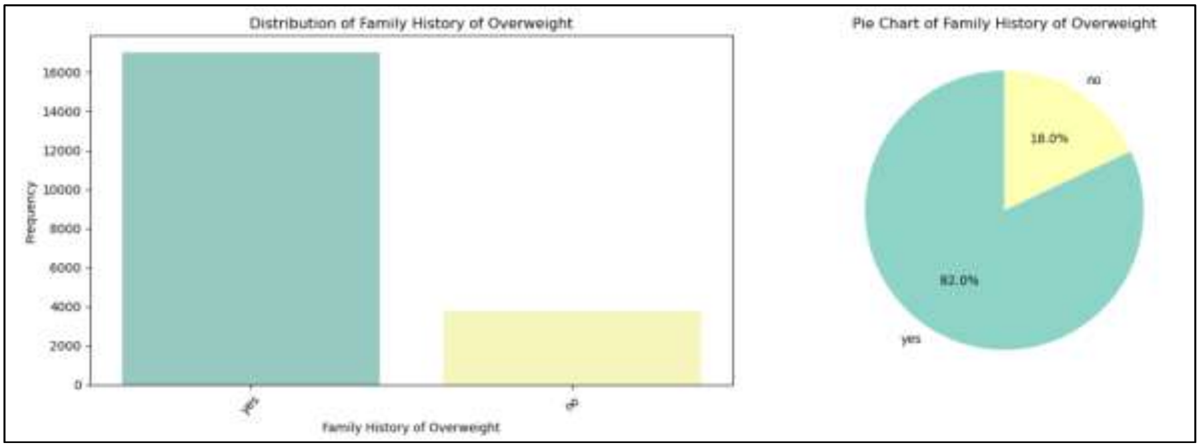


The distribution of target variable shows that Obesity_typeIII is the most common in people, having 19.5% of share, overall, its relatively balanced distribution classes so we can say it’s a balanced dataset.

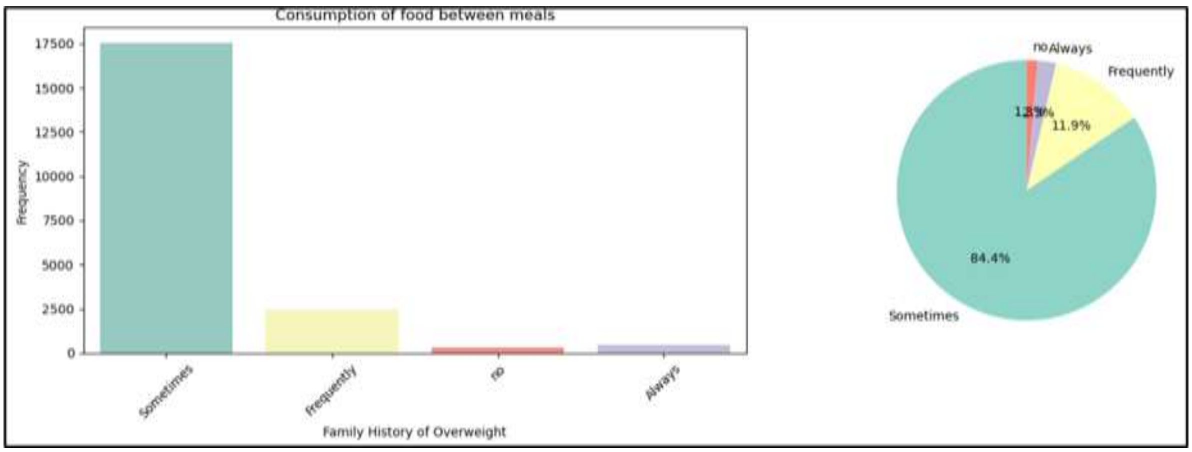
9. What are the distributions of all categorical attributes in the dataset?



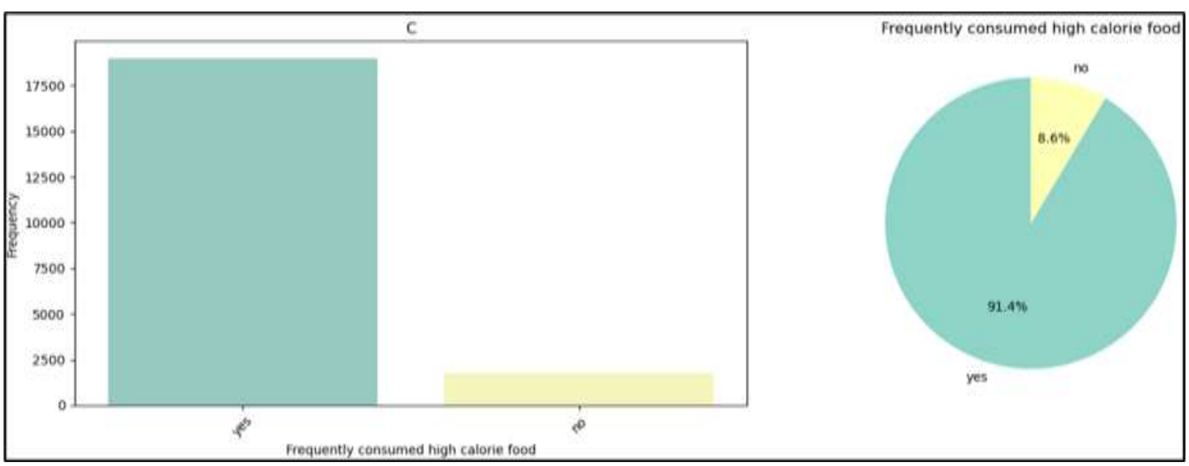
The dataset has balanced distribution gender.



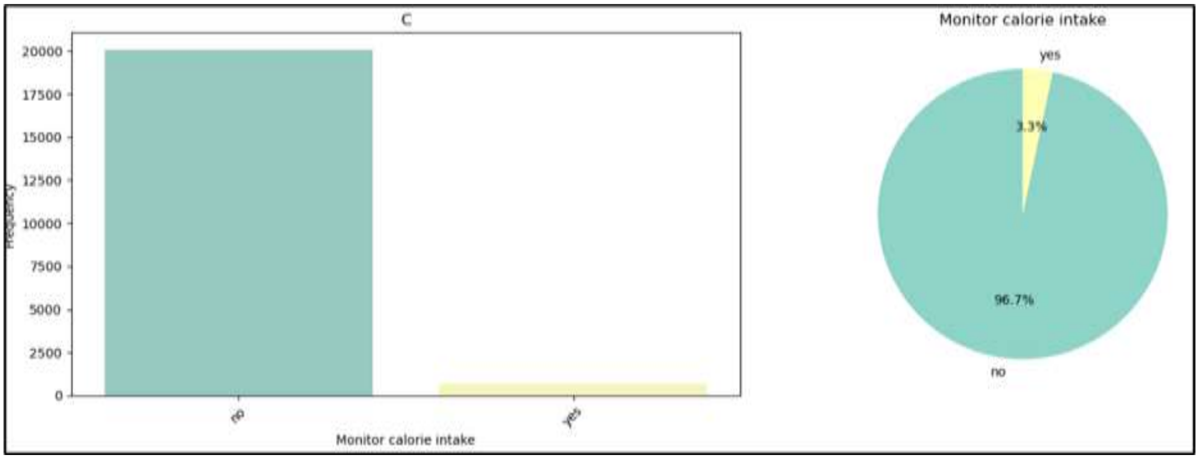
82% of people have a family history of being overweight.



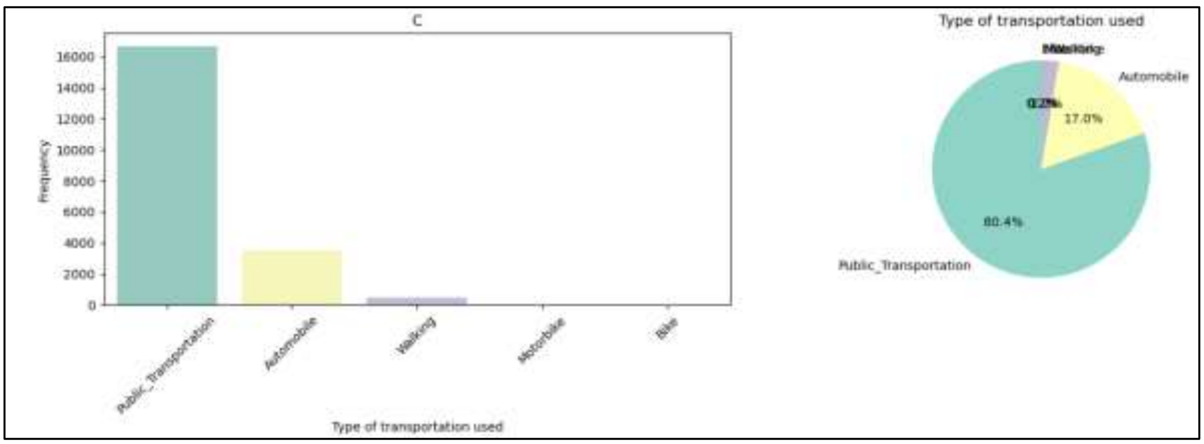
84.4% of individuals **sometimes** consume food between meals, while approximately 1.5% report not eating any meals in between.



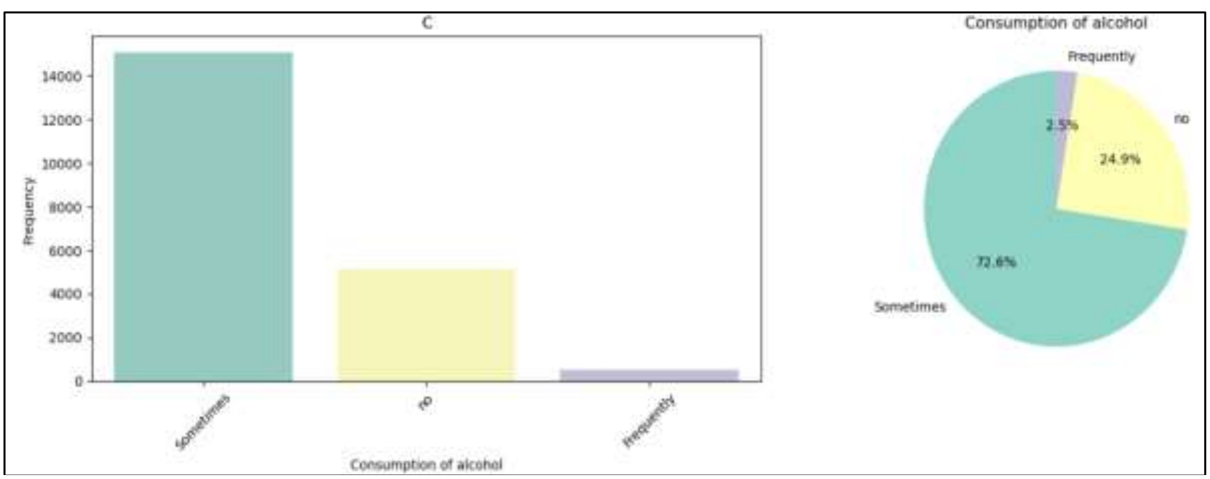
91.4% of people frequently consume high-calorie foods.



96.7% of people don’t monitor calorie intake



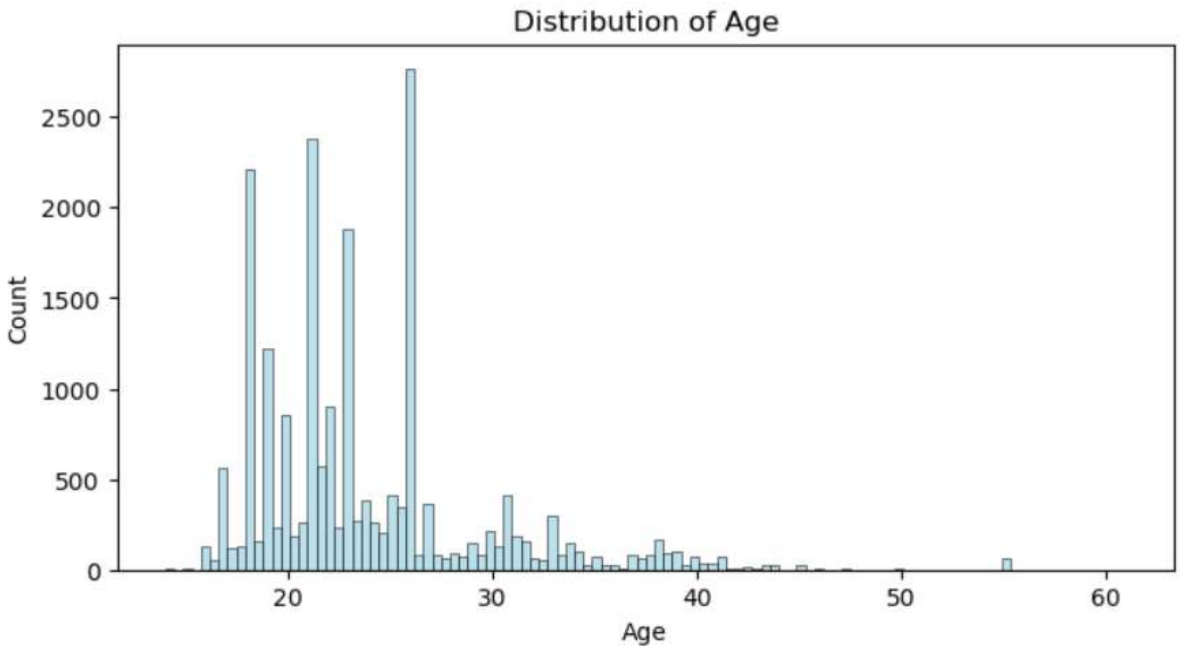
80.4% of people use public transport, 17% of people use automobile and only 0.6% people prefer walking/bike/motorbike



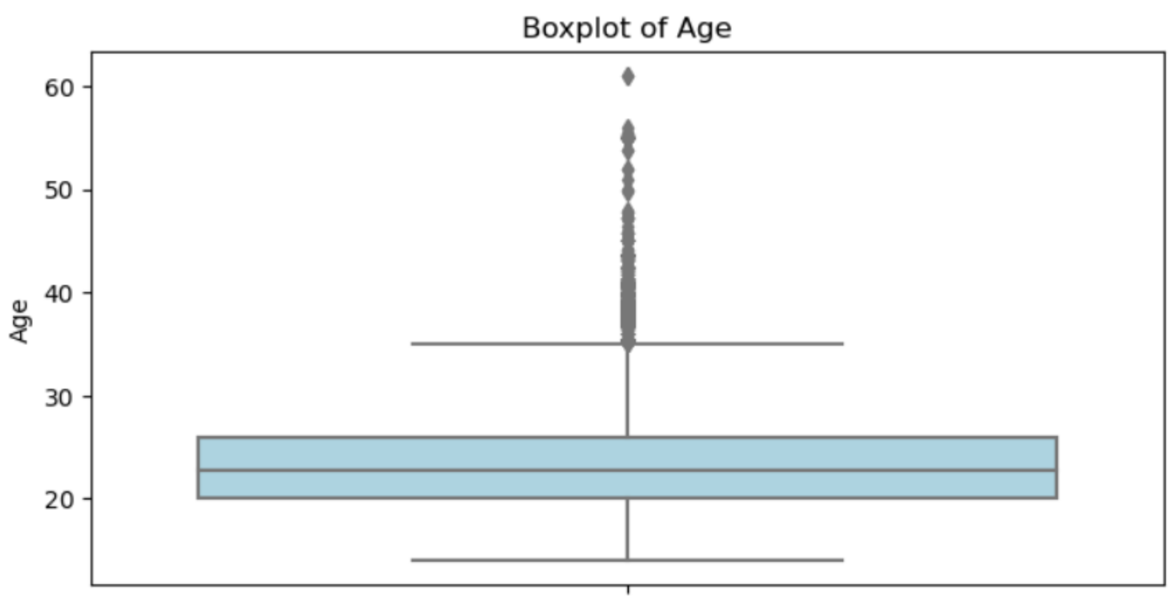
72.6% of people consume alcohol sometimes, while 2.5% people consume if frequently and 24.9% of people don’t consume it

Understanding of Numerical Attributes

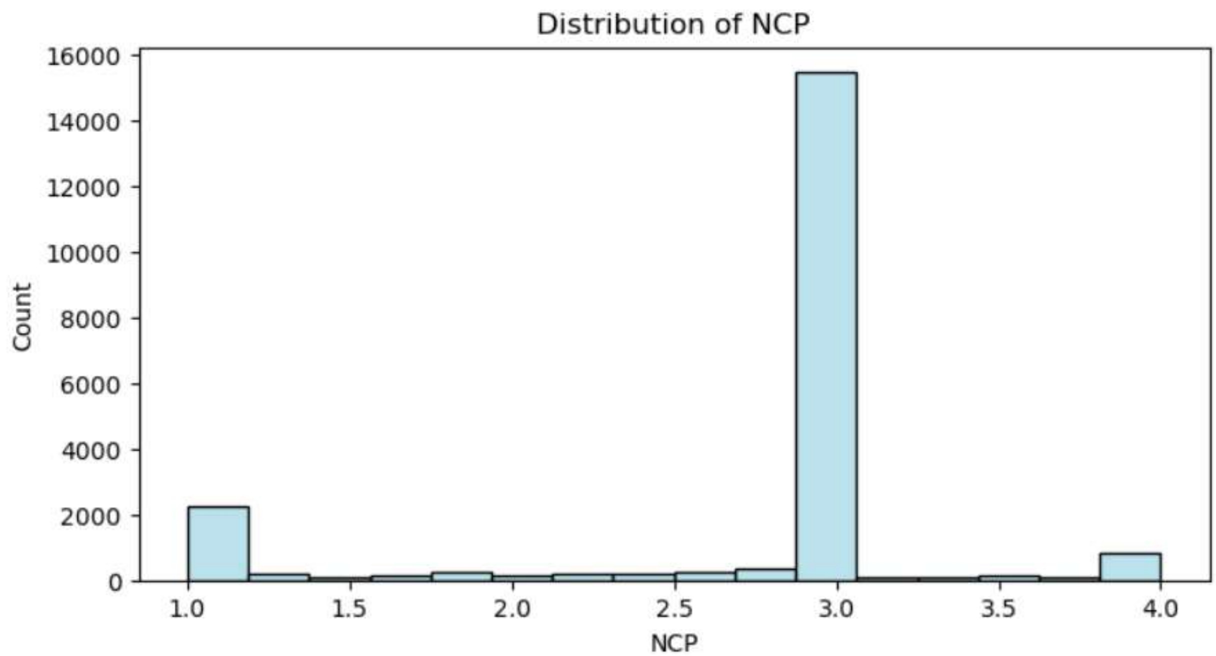
10. What are the distributions of all numerical attributes in the dataset?

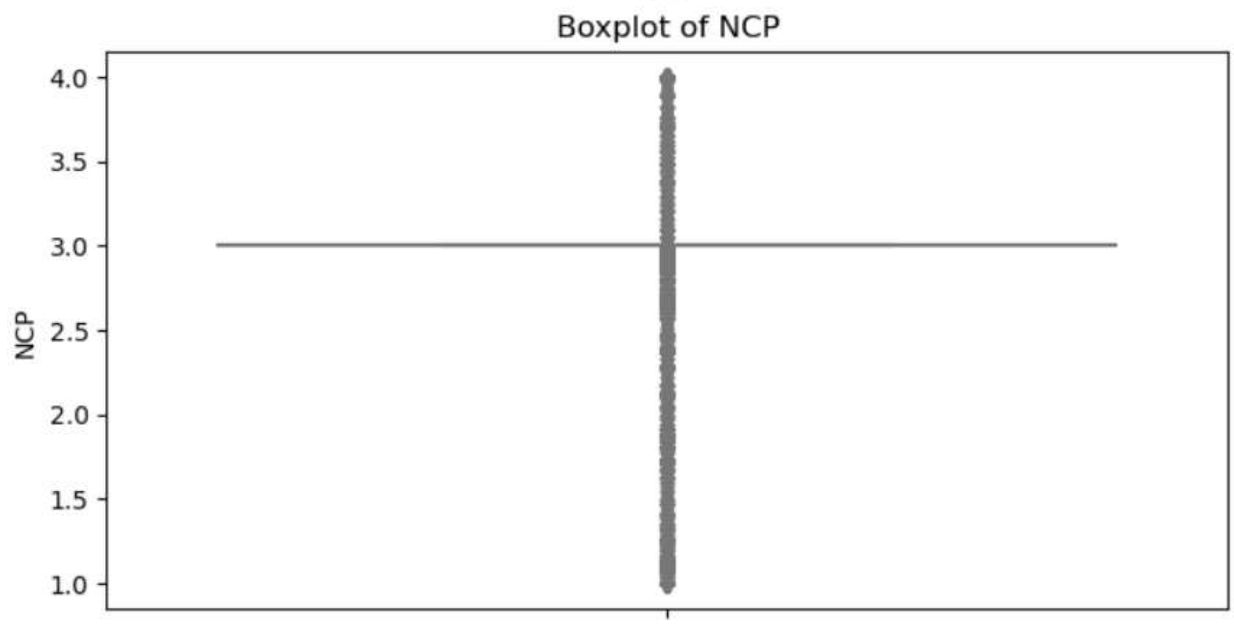


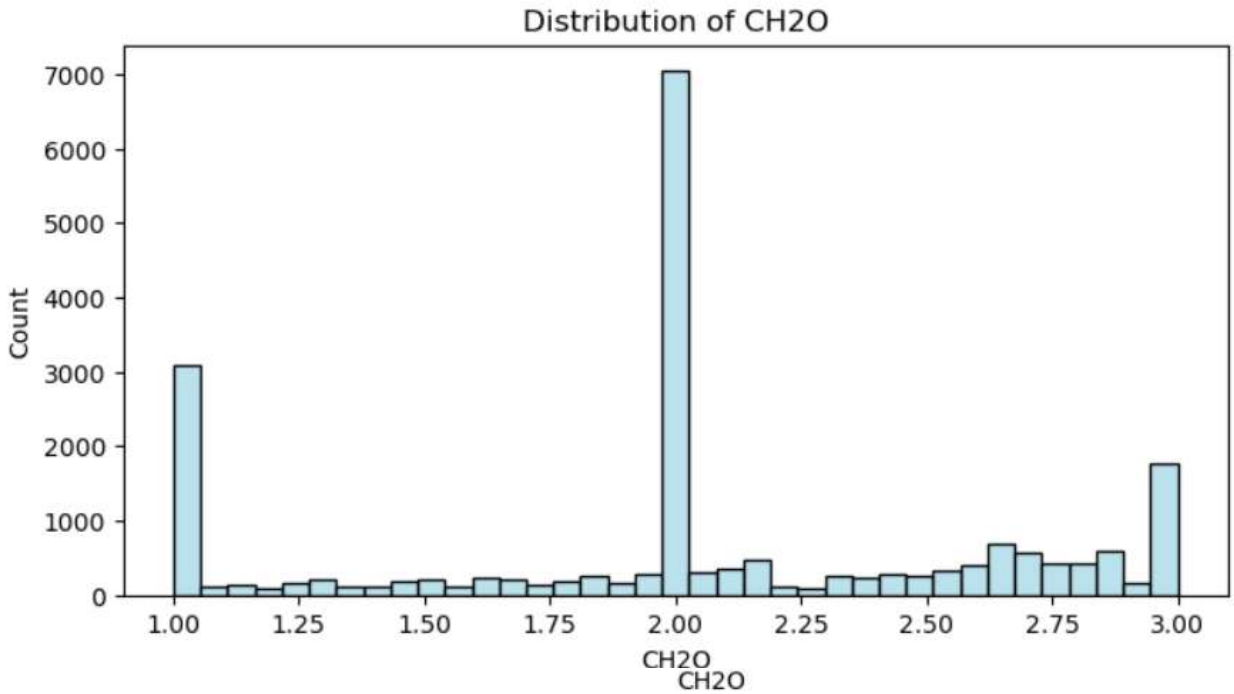
Histogram Age is skewed to the right, for that I can do transformation to have normal distribution.



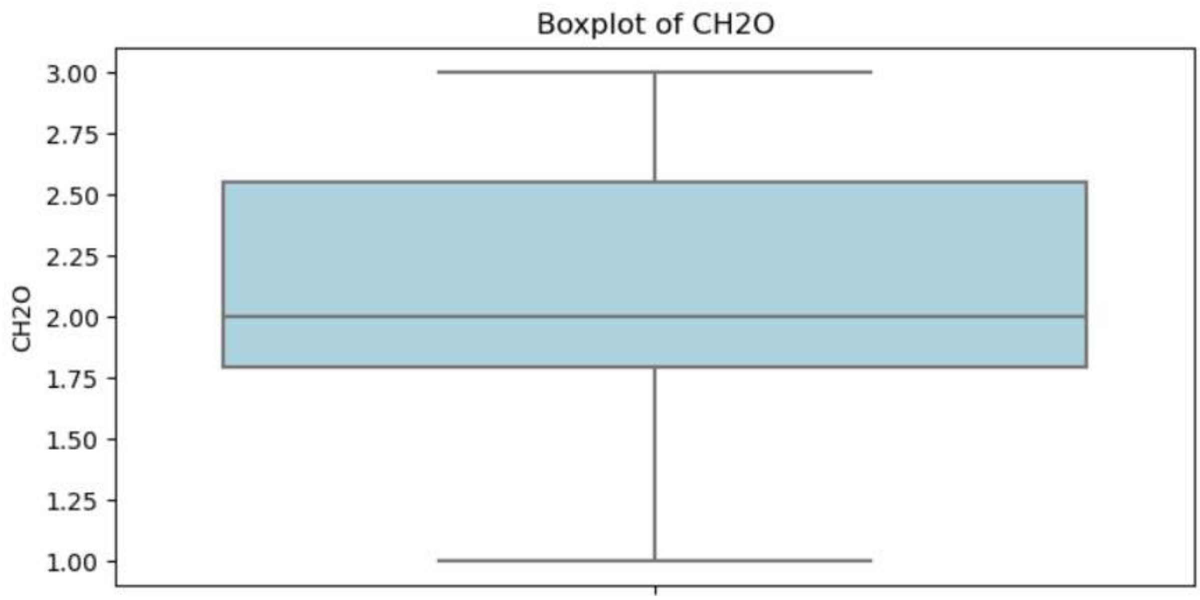
Box plot shows that median is round 23-24 years, box spans from approximately 18 to 28 years (IQR). Whiskers extend from 15 to 35 years suggesting a range where most data lies and several data points are above 35 years, indicating older individuals (outliers)


Distribution of NCP
Count
NCP
16000
14000
12000
10000
8000
6000
4000
2000
0
1.0
1.5
2.0
2.5
3.0
3.5
4.0

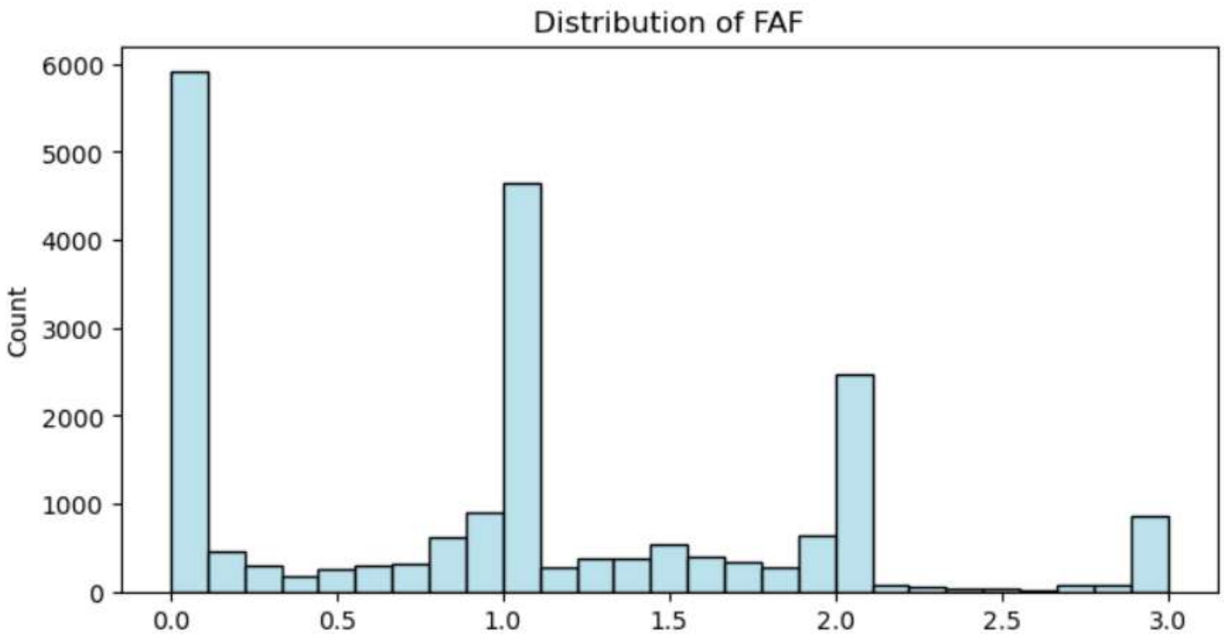

Boxplot of NCP
NCP
4.0
3.5
3.0
2.5
2.0
1.5
1.0

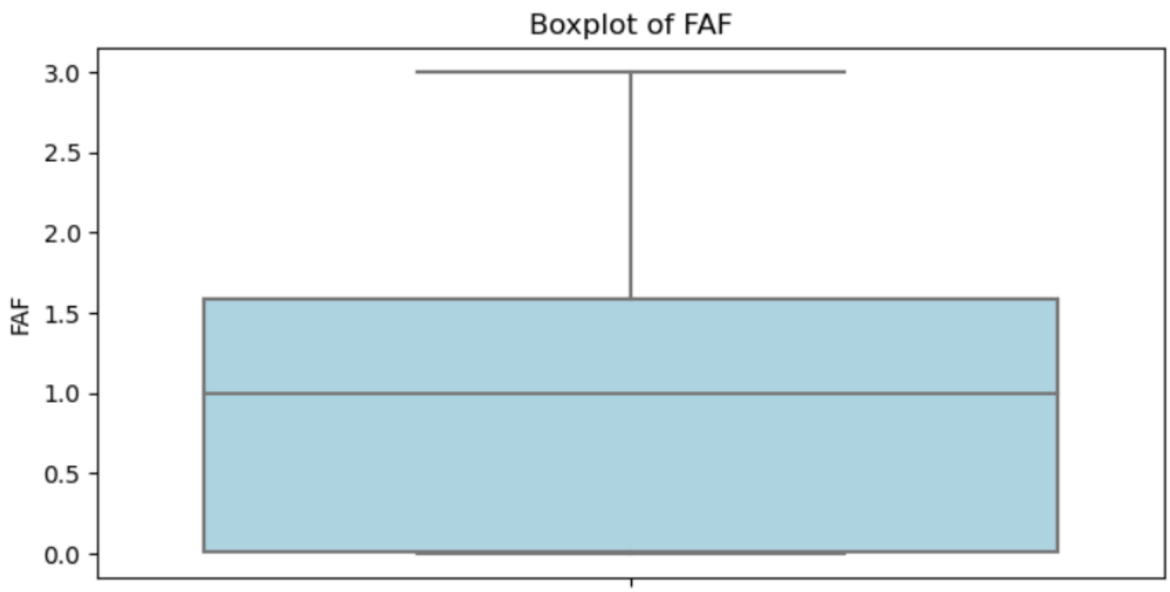


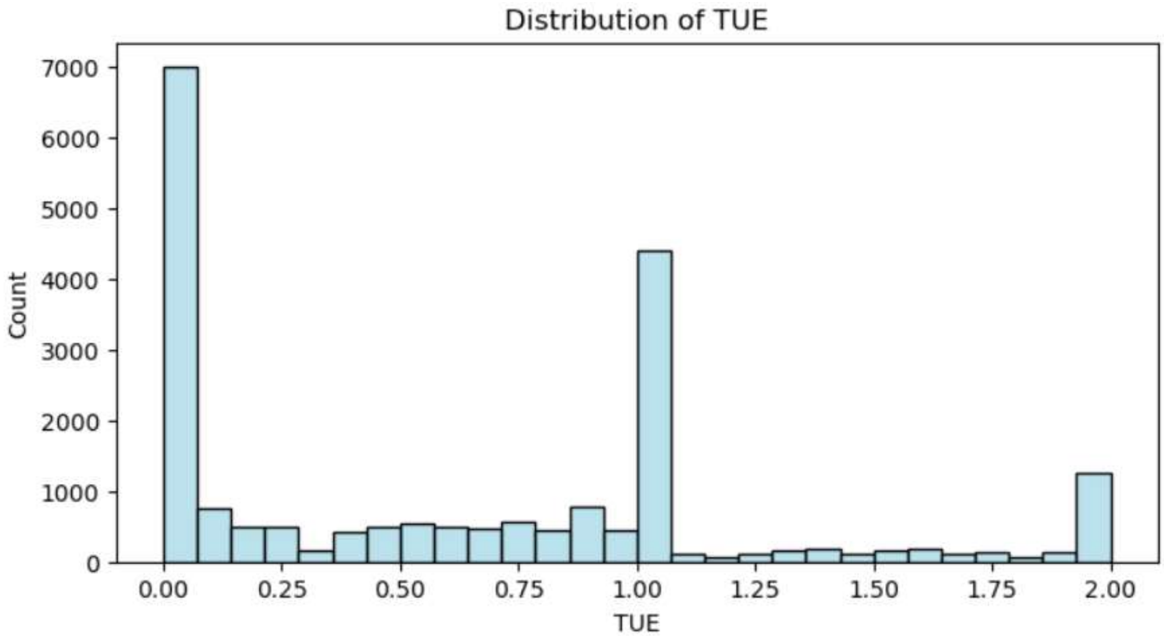
Distribution of CH2O
7000
6000
5000
4000
3000
2000
1000
0
Count
1.00
1.25
1.50
1.75
2.00
2.25
2.50
2.75
3.00
CH2O
CH2O

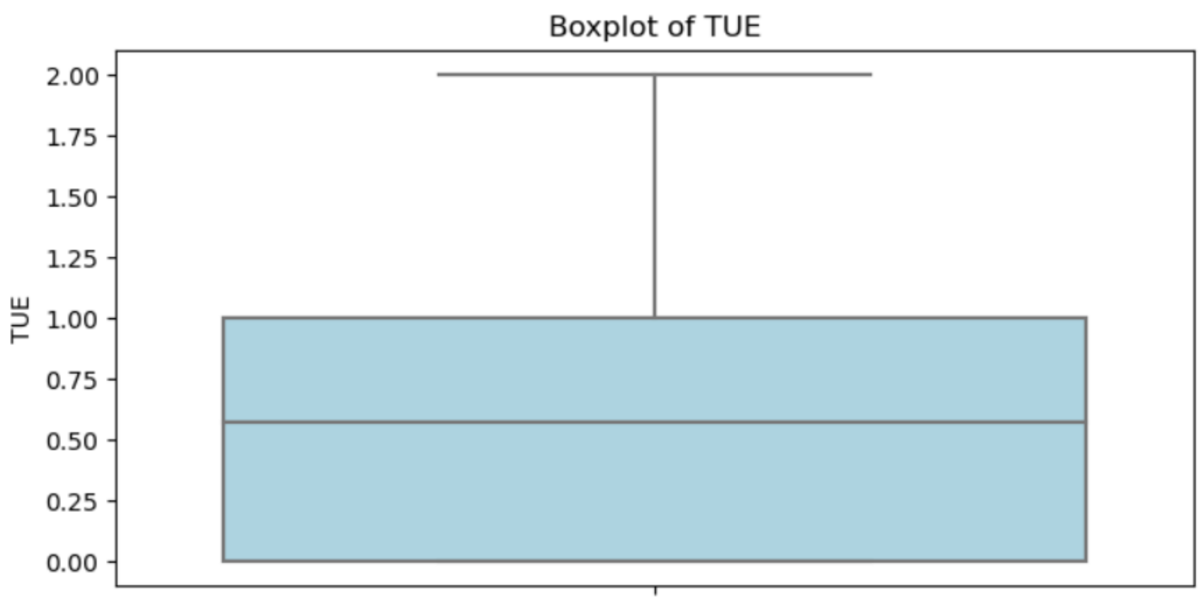


Boxplot of CH2O
3.00
2.75
2.50
2.25
2.00
1.75
1.50
1.25
1.00
CH2O


Distribution of FAF
6000
5000
4000
3000
2000
1000
0
Count
0.0
0.5
1.0
1.5
2.0
2.5
3.0

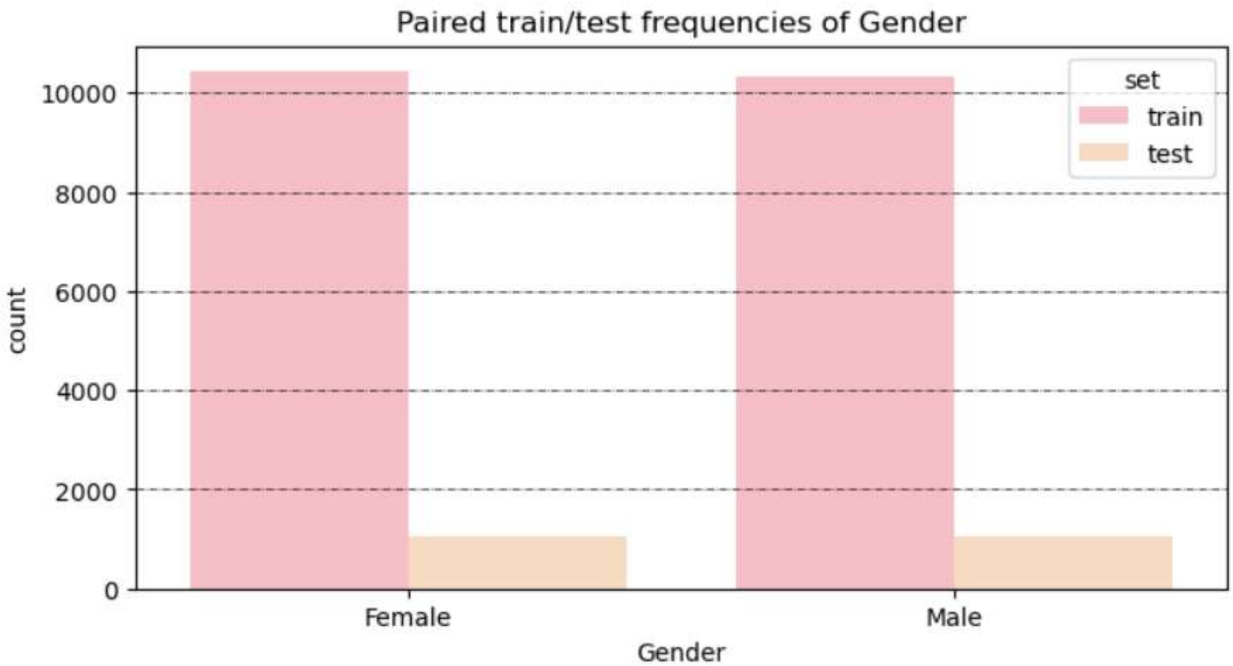

Boxplot of FAF
3.0
2.5
2.0
1.5
1.0
0.5
0.0
FAF

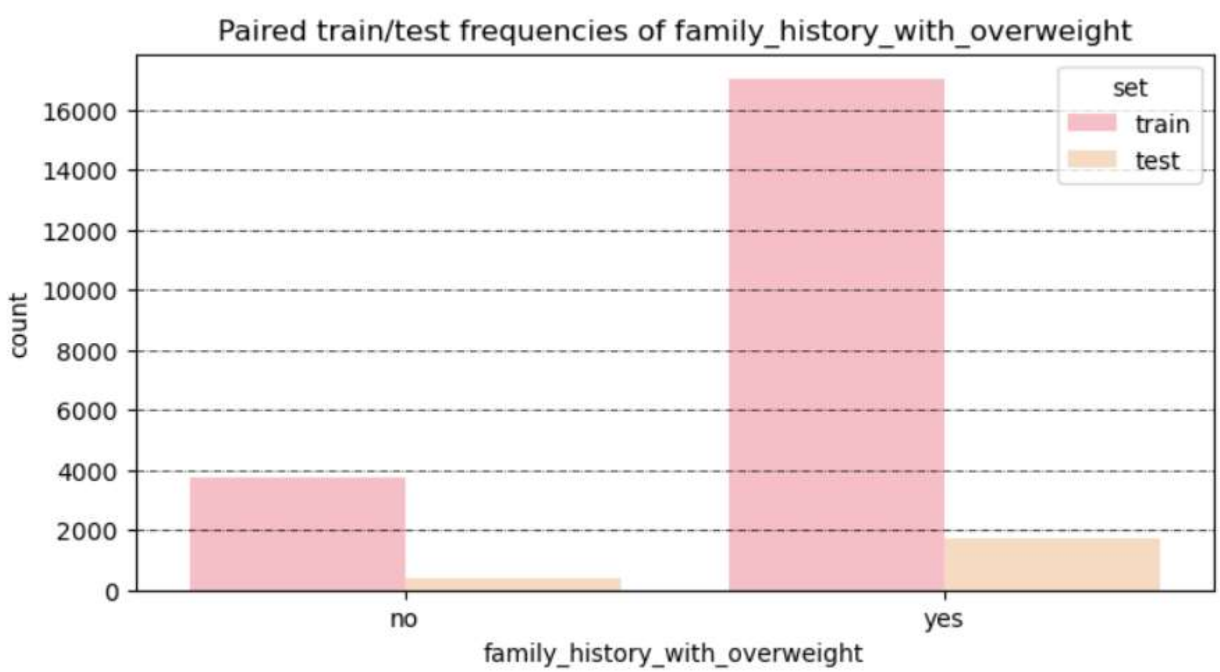

Distribution of TUE
Count
TUE
7000
6000
5000
4000
3000
2000
1000
0
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00

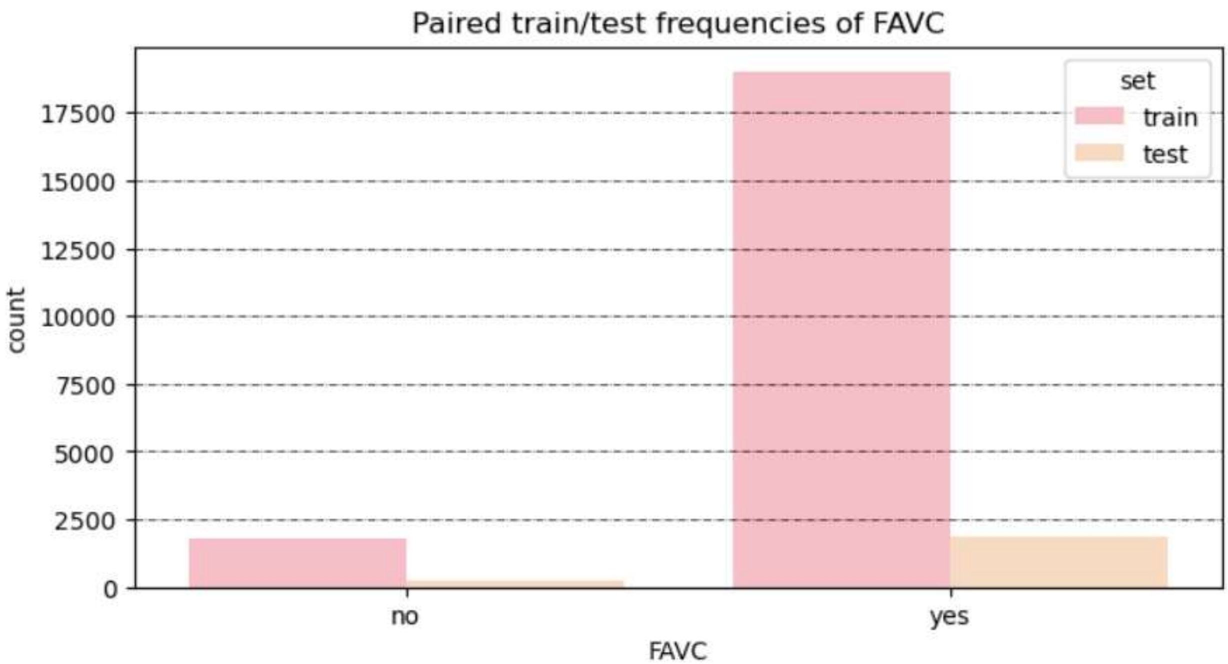

Boxplot of TUE
TUE
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00

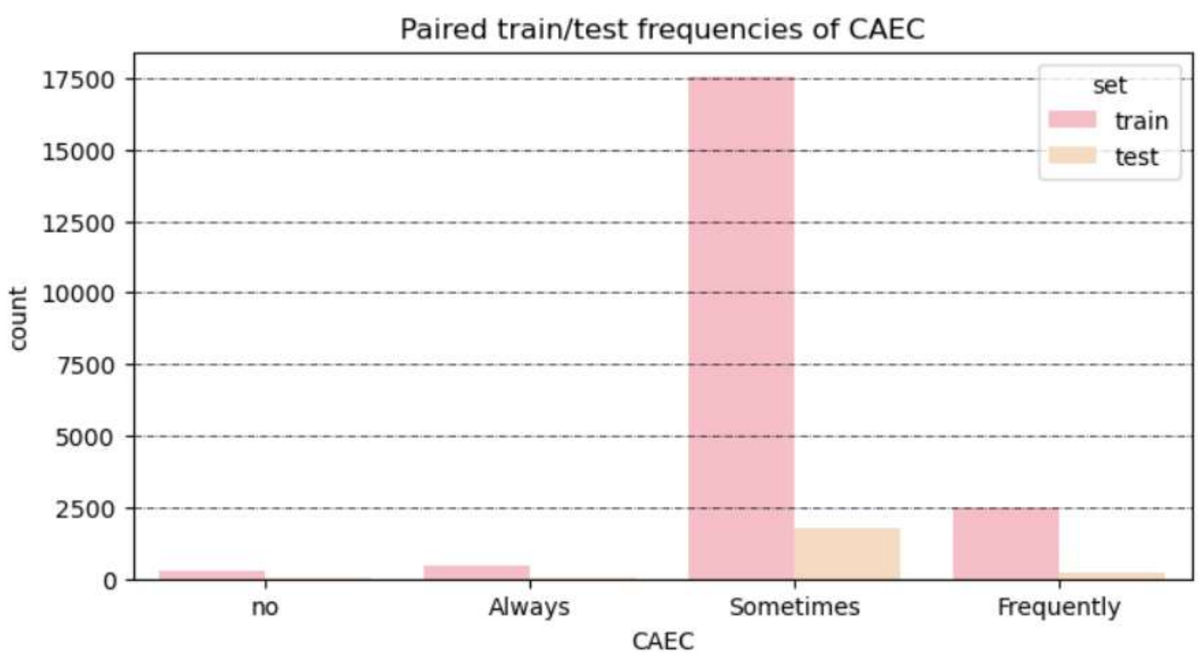
Train and Test frequency distributions

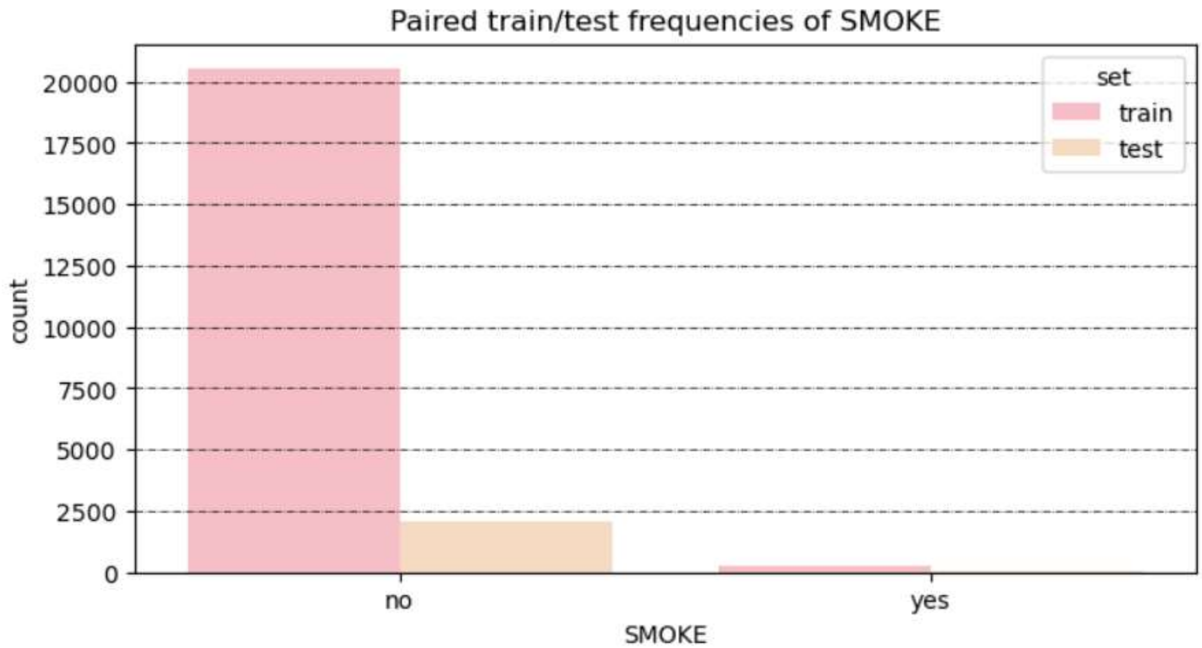
Categorical attribute distribution of train and test dataset



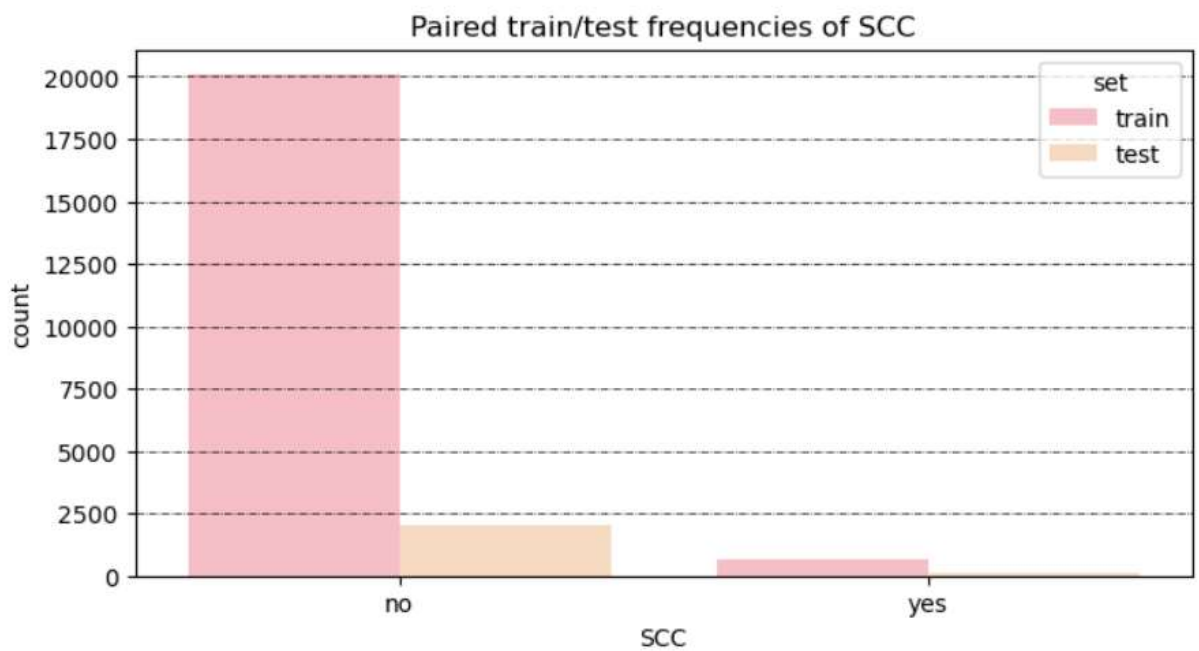



Paired train/test frequencies of FAVC
set
train
test
count
17500
15000
12500
10000
7500
5000
2500
0
no
yes
FAVC

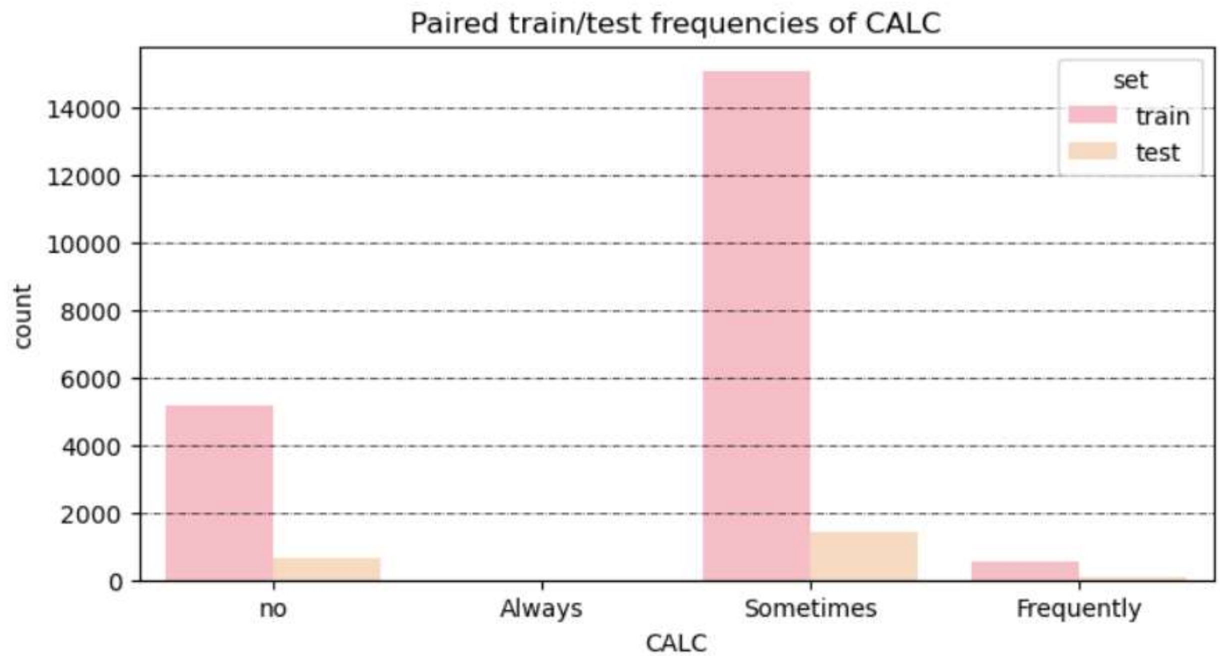

Paired train/test frequencies of CAEC
set
train
test
count
17500
15000
12500
10000
7500
5000
2500
0
no
Always
Sometimes
Frequently
CAEC

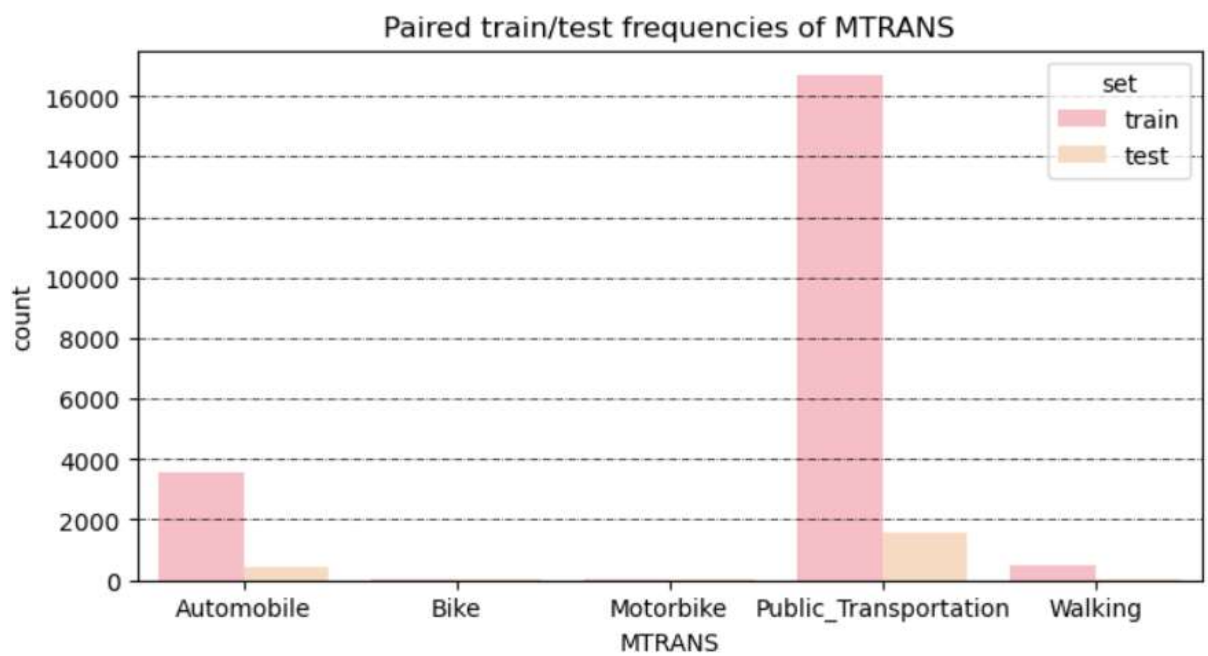


Paired train/test frequencies of SMOKE
set
train
test
count
20000
17500
15000
12500
10000
7500
5000
2500
0
no
yes
SMOKE

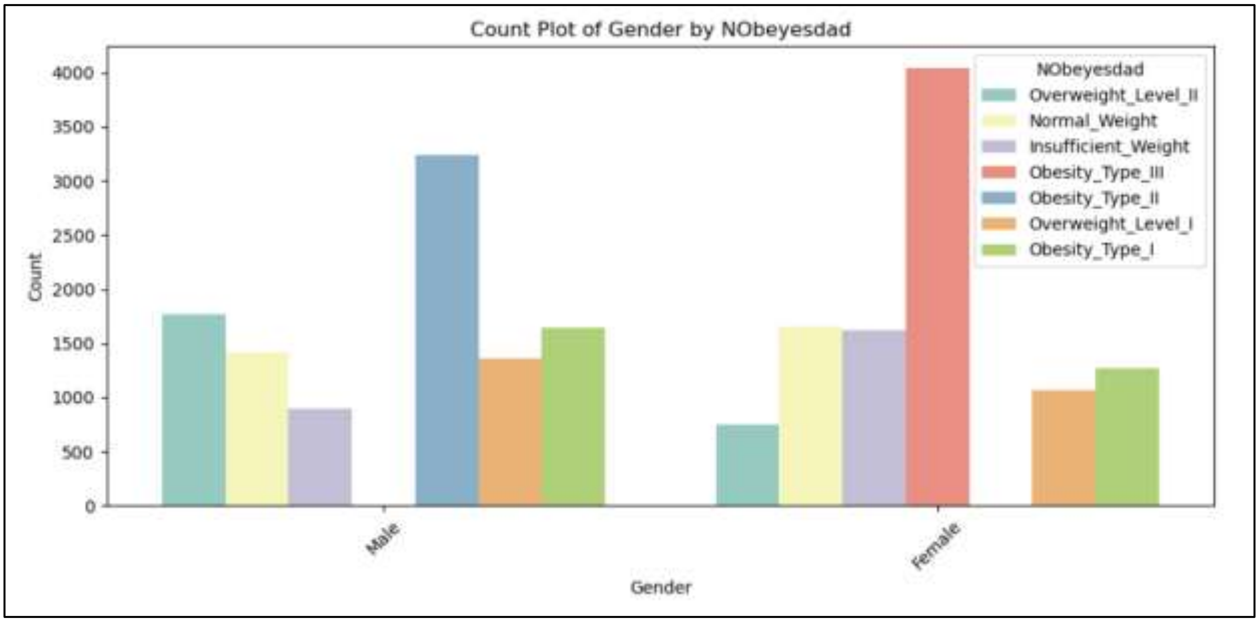


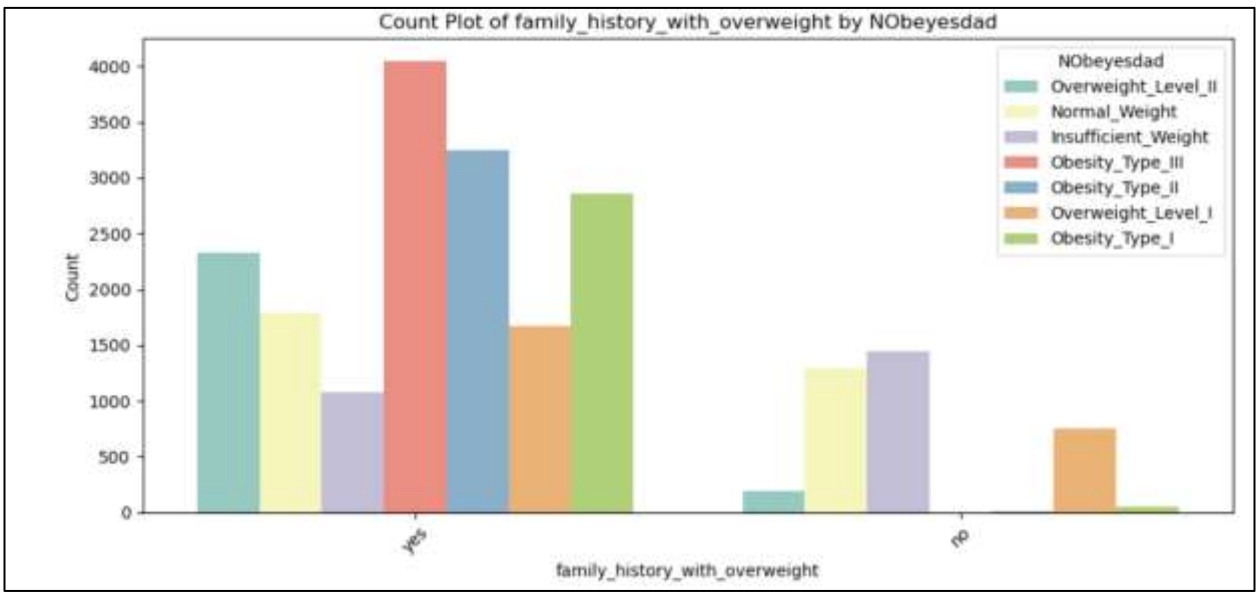
Paired train/test frequencies of SCC
set
train
test
count
20000
17500
15000
12500
10000
7500
5000
2500
0
no
yes
SCC

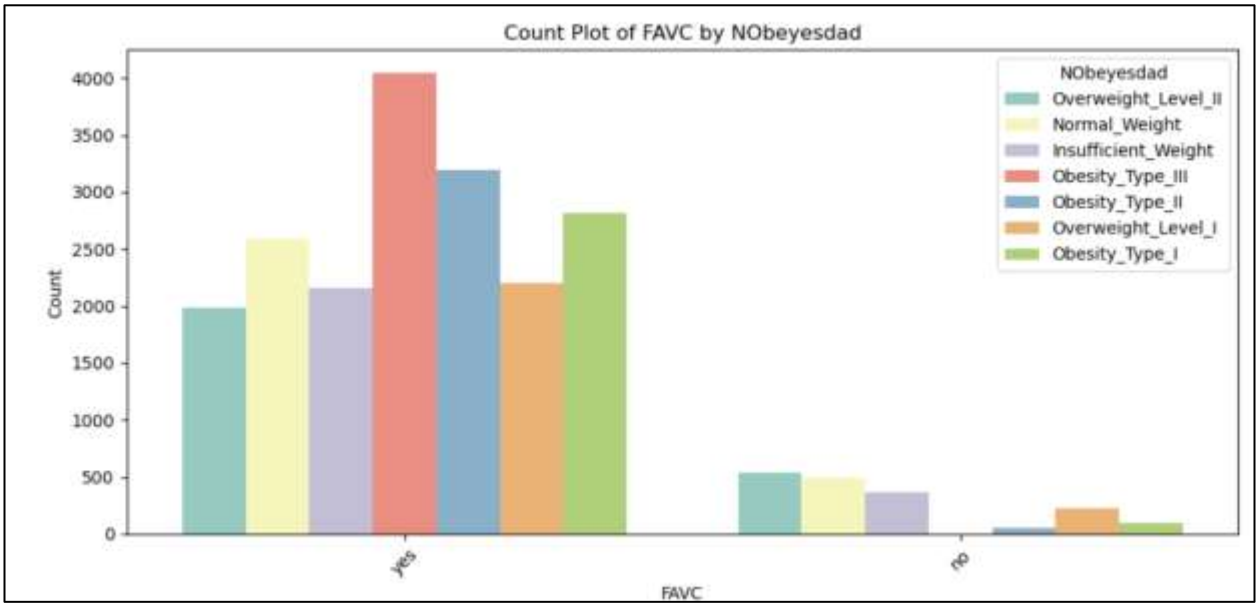

Paired train/test frequencies of CALC
set
train
test
count
14000
12000
10000
8000
6000
4000
2000
0
no
Always
Sometimes
Frequently
CALC

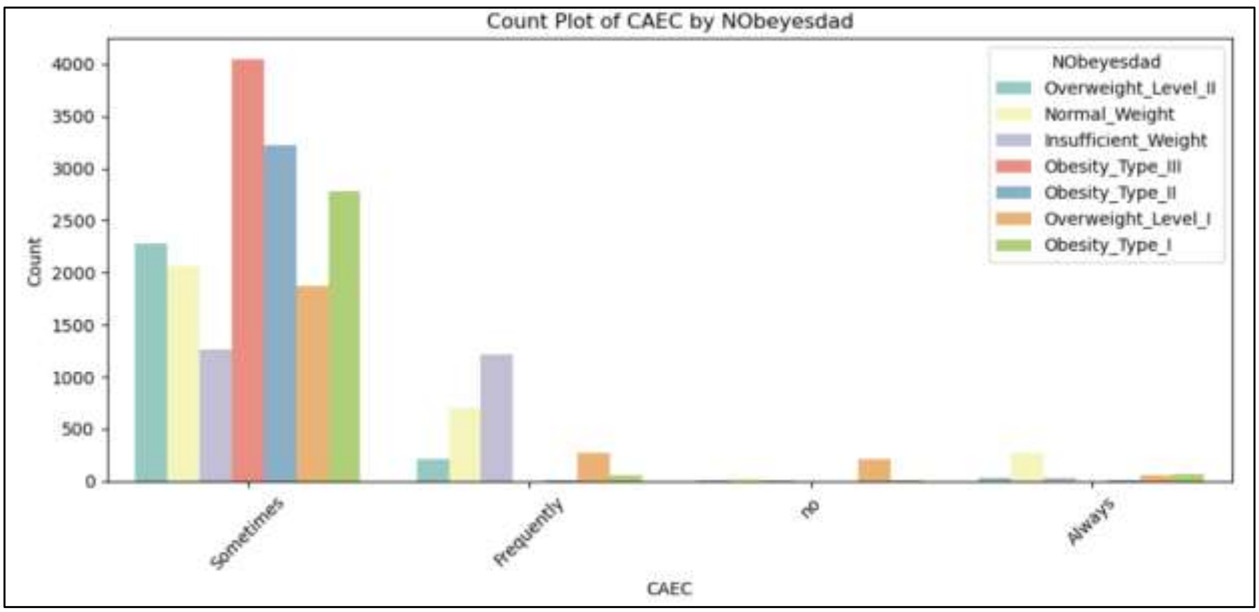

Paired train/test frequencies of MTRANS
set
train
test
count
16000
14000
12000
10000
8000
6000
4000
2000
0
Automobile
Bike
Motorbike
Public_Transportation
Walking
MTRANS

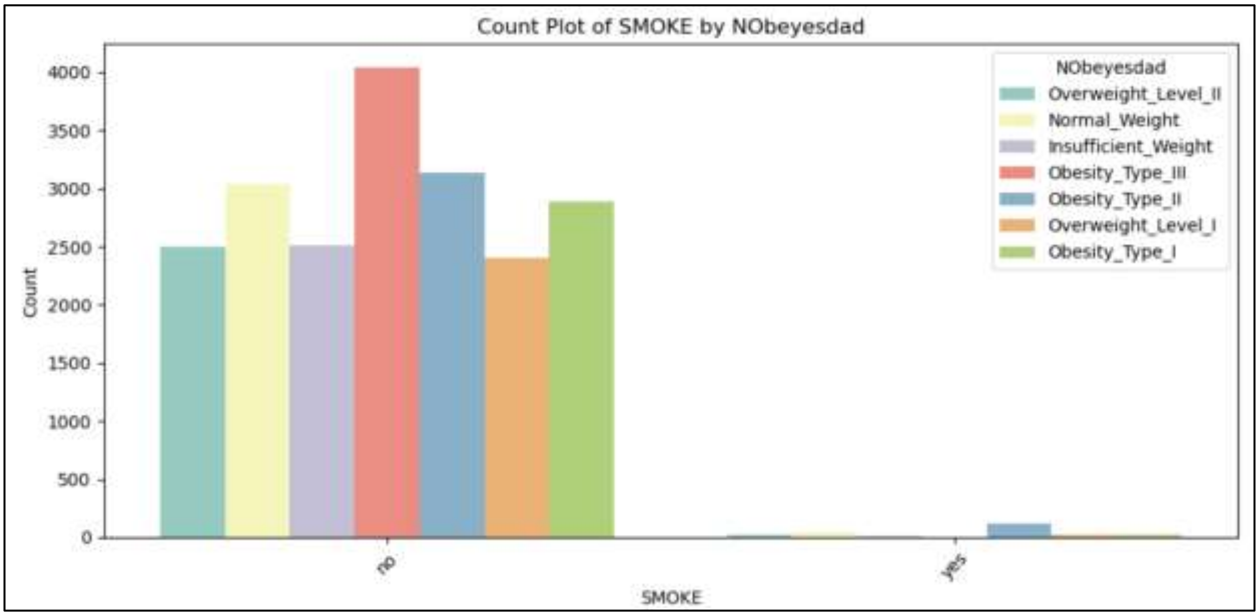
Bivariant Analysis: Count plots of categorical attributes specified by Target

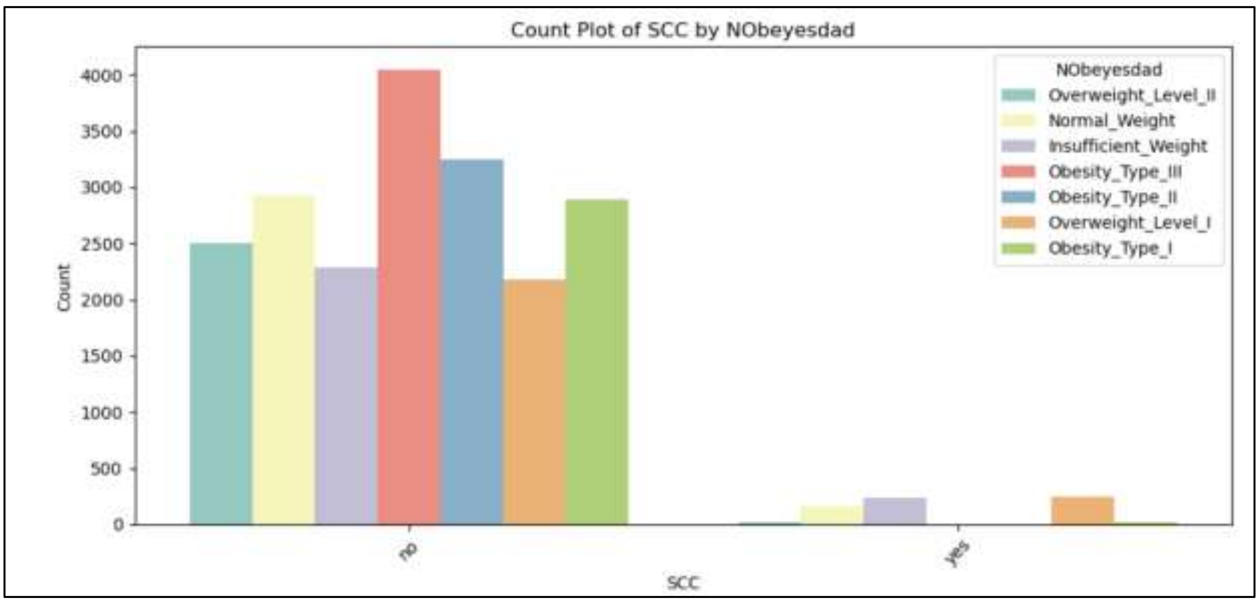


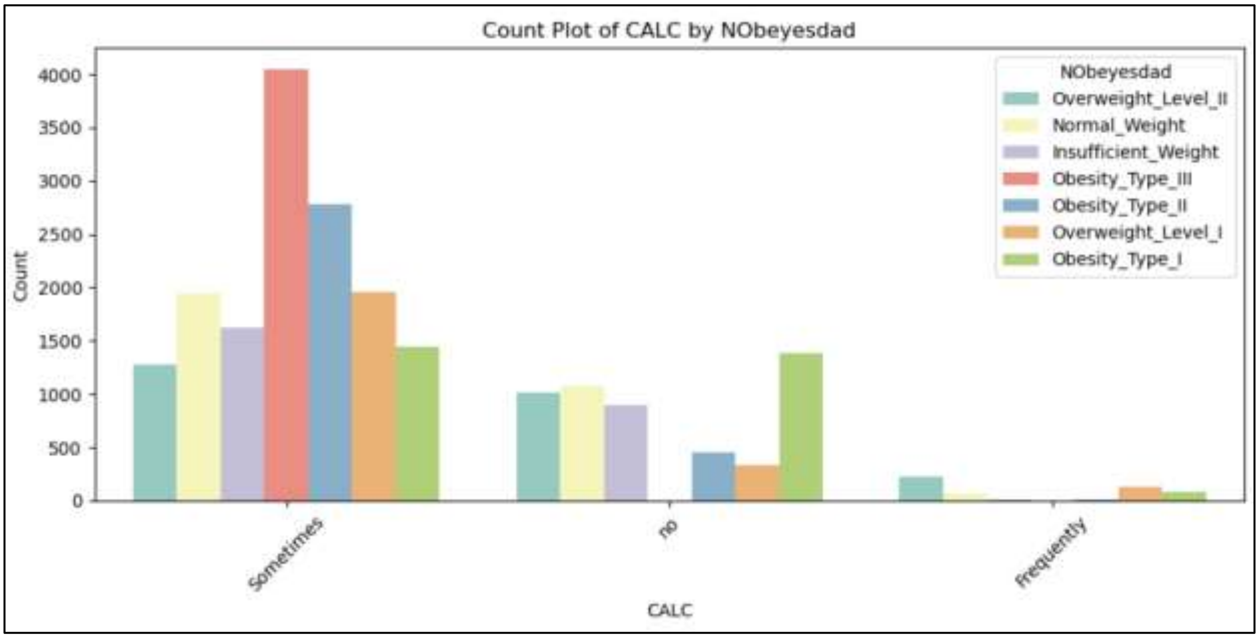


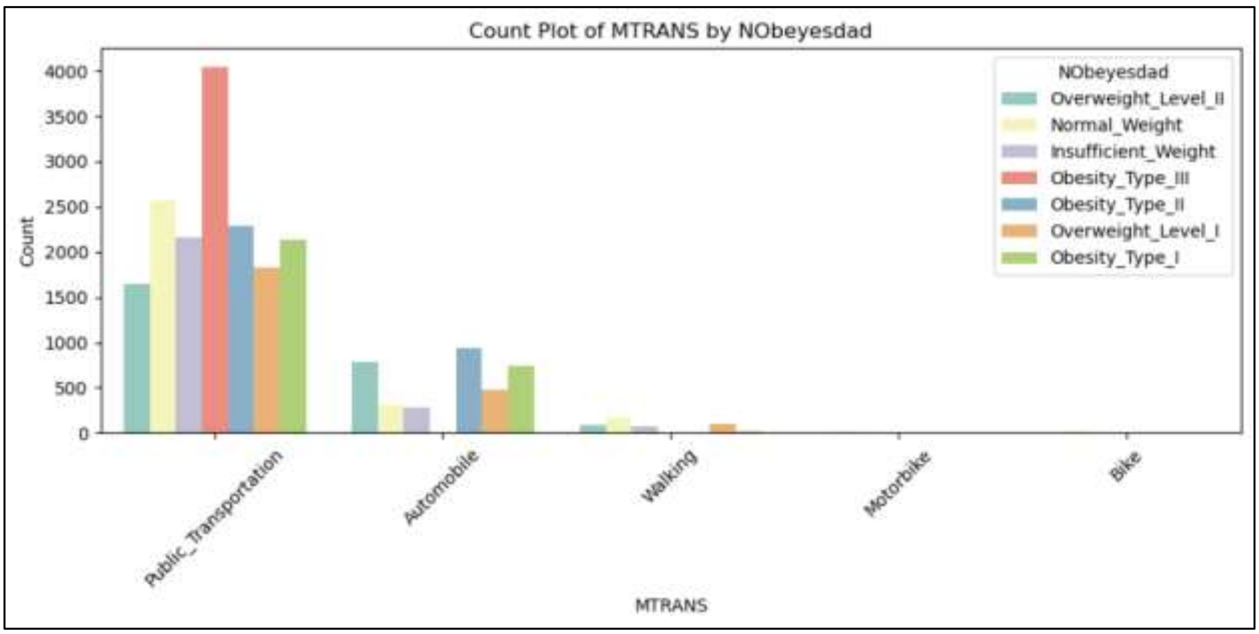

Count Plot of FAVC by NObeyesdad
NObeyesdad
Overweight_Level_II
Normal_Weight
Insufficient_Weight
Obesity_Type_III
Obesity_Type_II
Overweight_Level_I
Obesity_Type_I
Count
4000
3500
3000
2500
2000
1500
1000
500
0
yes
no
FAVC


Count Plot of CAEC by NObeyesdad
NObeyesdad
Overweight_Level_II
Normal_Weight
Insufficient_Weight
Obesity_Type_III
Obesity_Type_II
Overweight_Level_I
Obesity_Type_I
Count
4000
3500
3000
2500
2000
1500
1000
500
0
Sometimes
Frequently
no
Always
CAEC

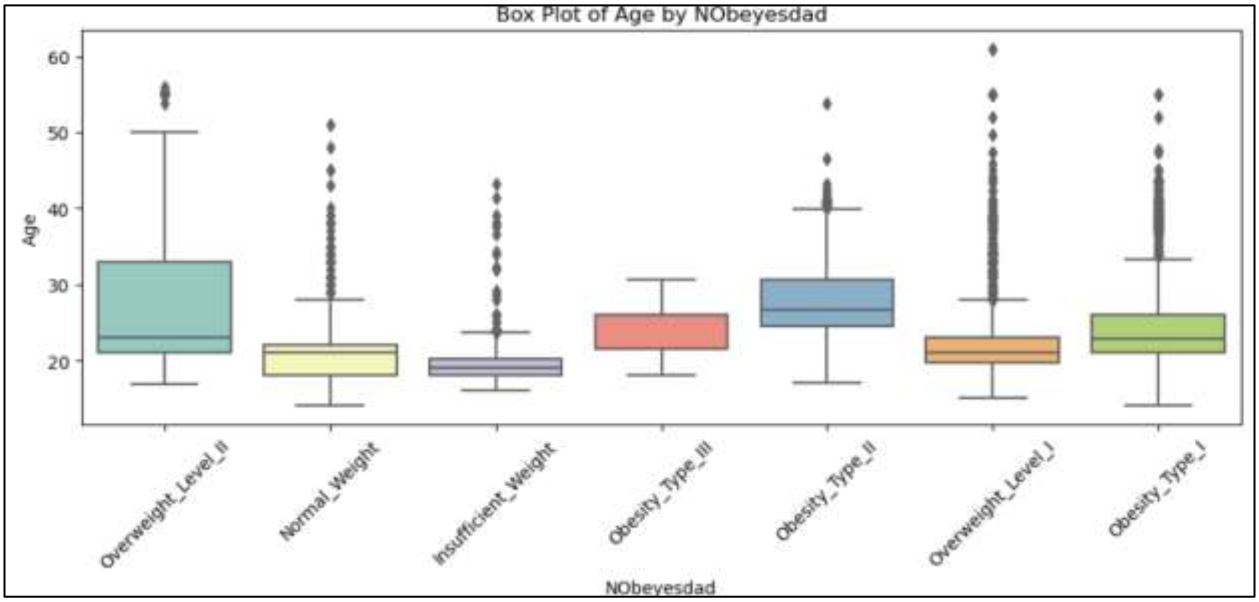

Count Plot of SMOKE by NObeyesdad
NObeyesdad
Overweight_Level_II
Normal_Weight
Insufficient_Weight
Obesity_Type_III
Obesity_Type_II
Overweight_Level_I
Obesity_Type_I
Count
4000
3500
3000
2500
2000
1500
1000
500
0
no
yes
SMOKE

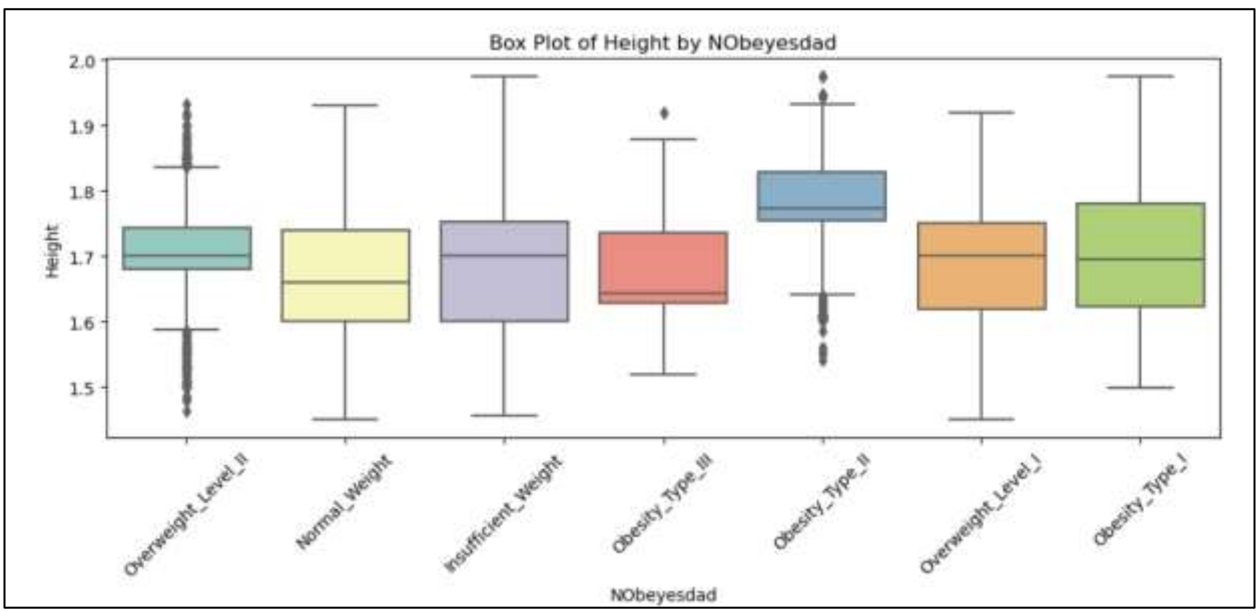

Count Plot of SCC by NObeyesdad
NObeyesdad
Overweight_Level_II
Normal_Weight
Insufficient_Weight
Obesity_Type_III
Obesity_Type_II
Overweight_Level_I
Obesity_Type_I
Count
4000
3500
3000
2500
2000
1500
1000
500
0
no
yes
SCC

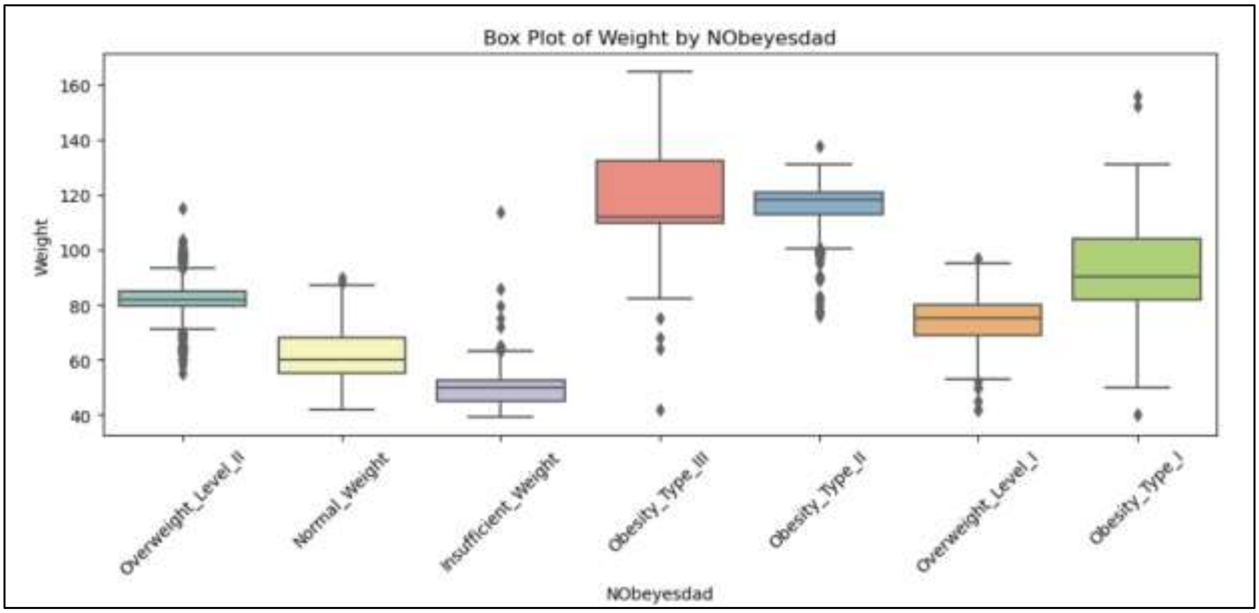

Count Plot of CALC by NObeyesdad
Count
CALC
Sometimes
no
Frequently
NObeyesdad
Overweight_Level_II
Normal_Weight
Insufficient_Weight
Obesity_Type_III
Obesity_Type_II
Overweight_Level_I
Obesity_Type_I
0
500
1000
1500
2000
2500
3000
3500
4000

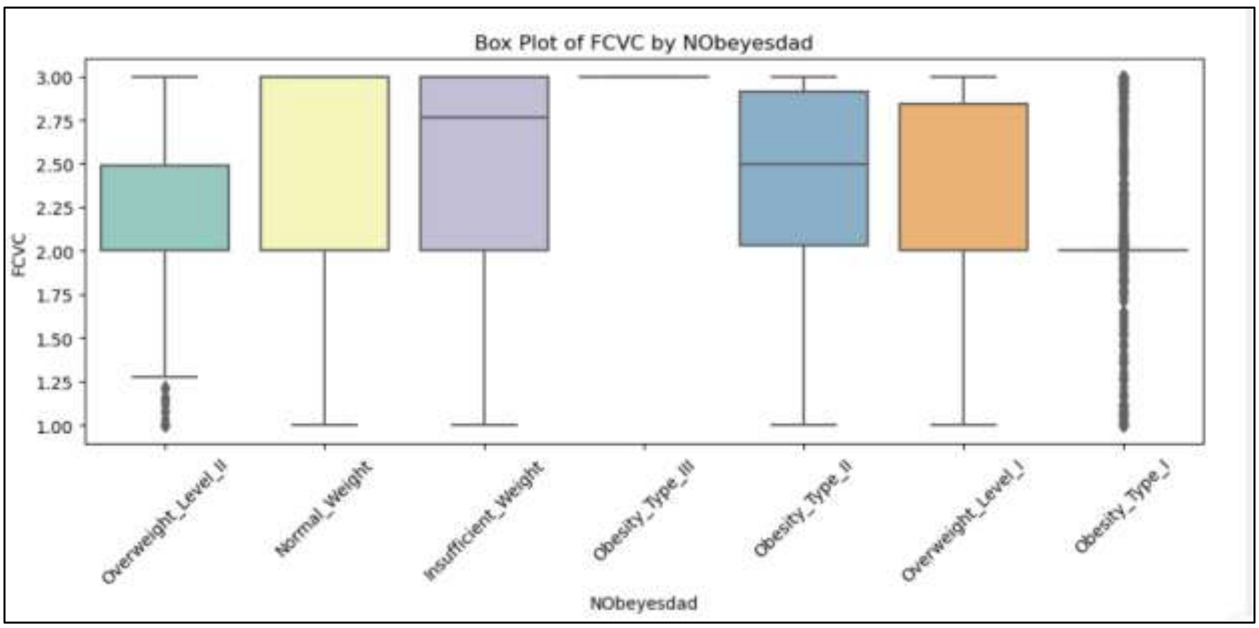

Count Plot of MTRANS by NObeyesdad
Count
MTRANS
Public_Transportation
Automobile
Walking
Motorbike
Bike
NObeyesdad
Overweight_Level_II
Normal_Weight
Insufficient_Weight
Obesity_Type_III
Obesity_Type_II
Overweight_Level_I
Obesity_Type_I
0
500
1000
1500
2000
2500
3000
3500
4000

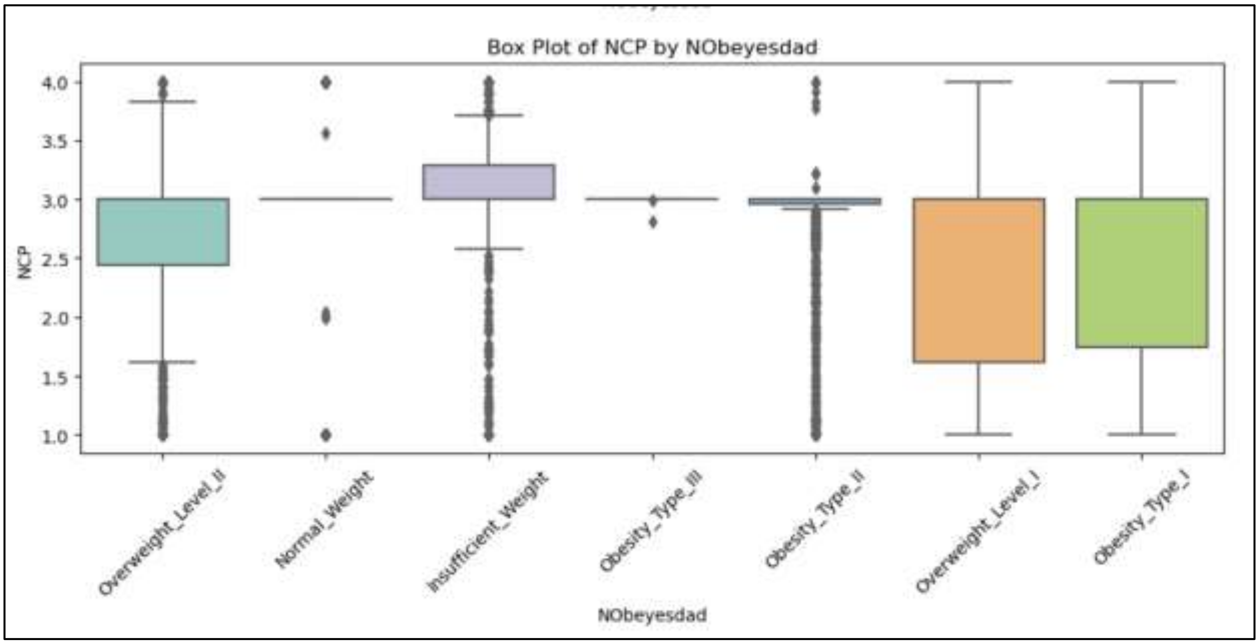
Box plots of numerical attributes specified by Target

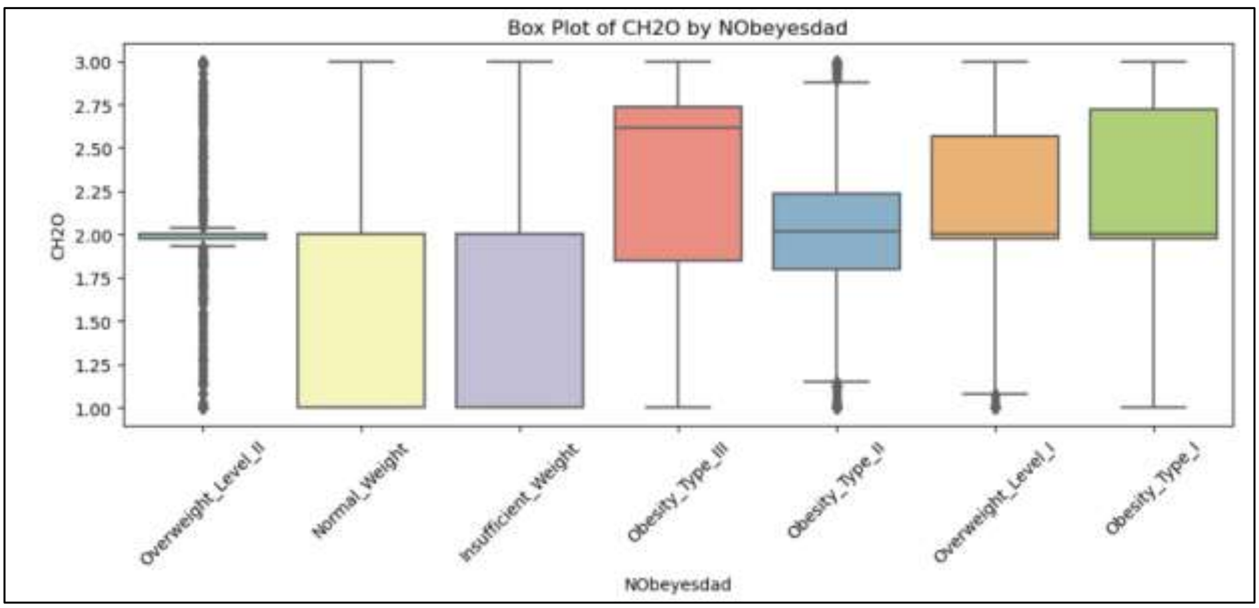


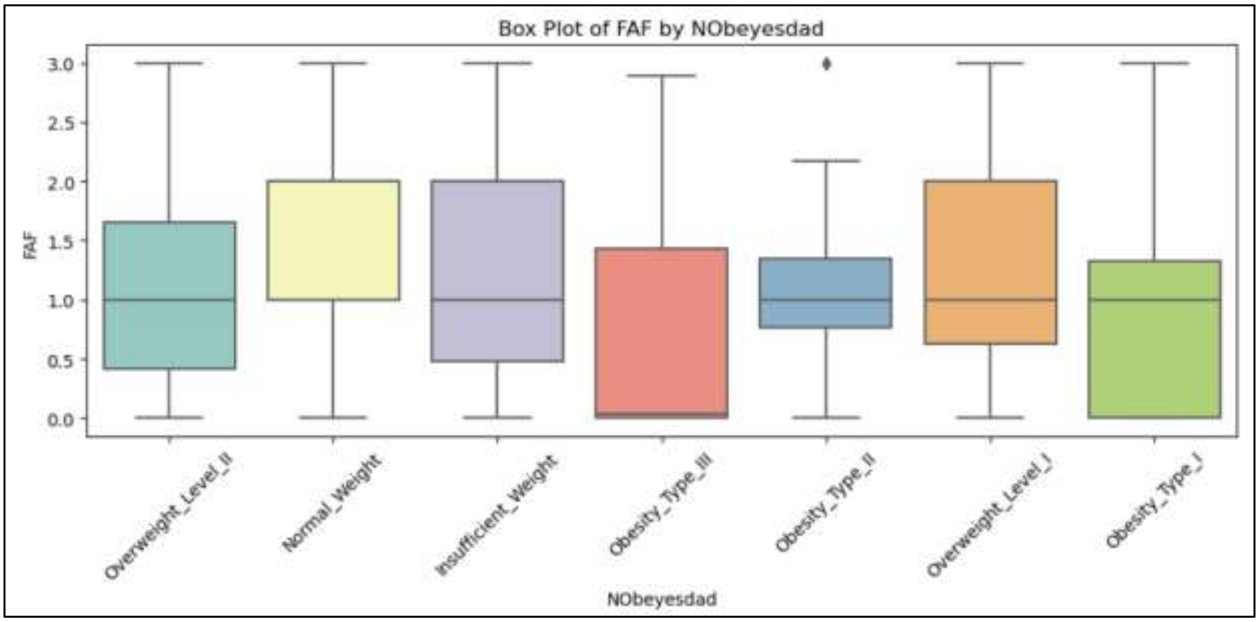


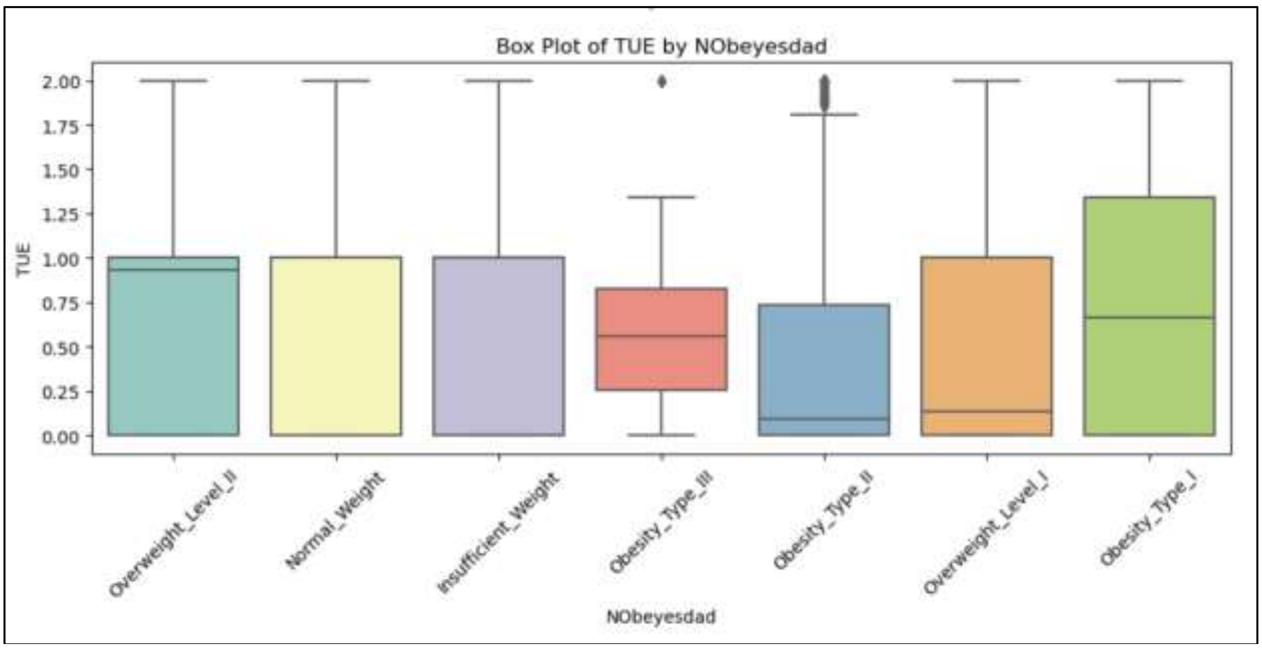

Box Plot of Weight by NObeyesdad
Weight
160
140
120
100
80
60
40
Overweight_Level_II
Normal_Weight
Insufficient_Weight
Obesity_Type_III
Obesity_Type_II
Overweight_Level_I
Obesity_Type_I
NObeyesdad


Box Plot of FCVC by NObeyesdad
FCVC
3.00
2.75
2.50
2.25
2.00
1.75
1.50
1.25
1.00
Overweight_Level_II
Normal_Weight
Insufficient_Weight
Obesity_Type_III
Obesity_Type_II
Overweight_Level_I
Obesity_Type_I
NObeyesdad


Box Plot of NCP by NObeyesdad
NCP
4.0
3.5
3.0
2.5
2.0
1.5
1.0
Overweight_Level_II
Normal_Weight
Insufficient_Weight
Obesity_Type_III
Obesity_Type_II
Overweight_Level_I
Obesity_Type_I
NObeyesdad

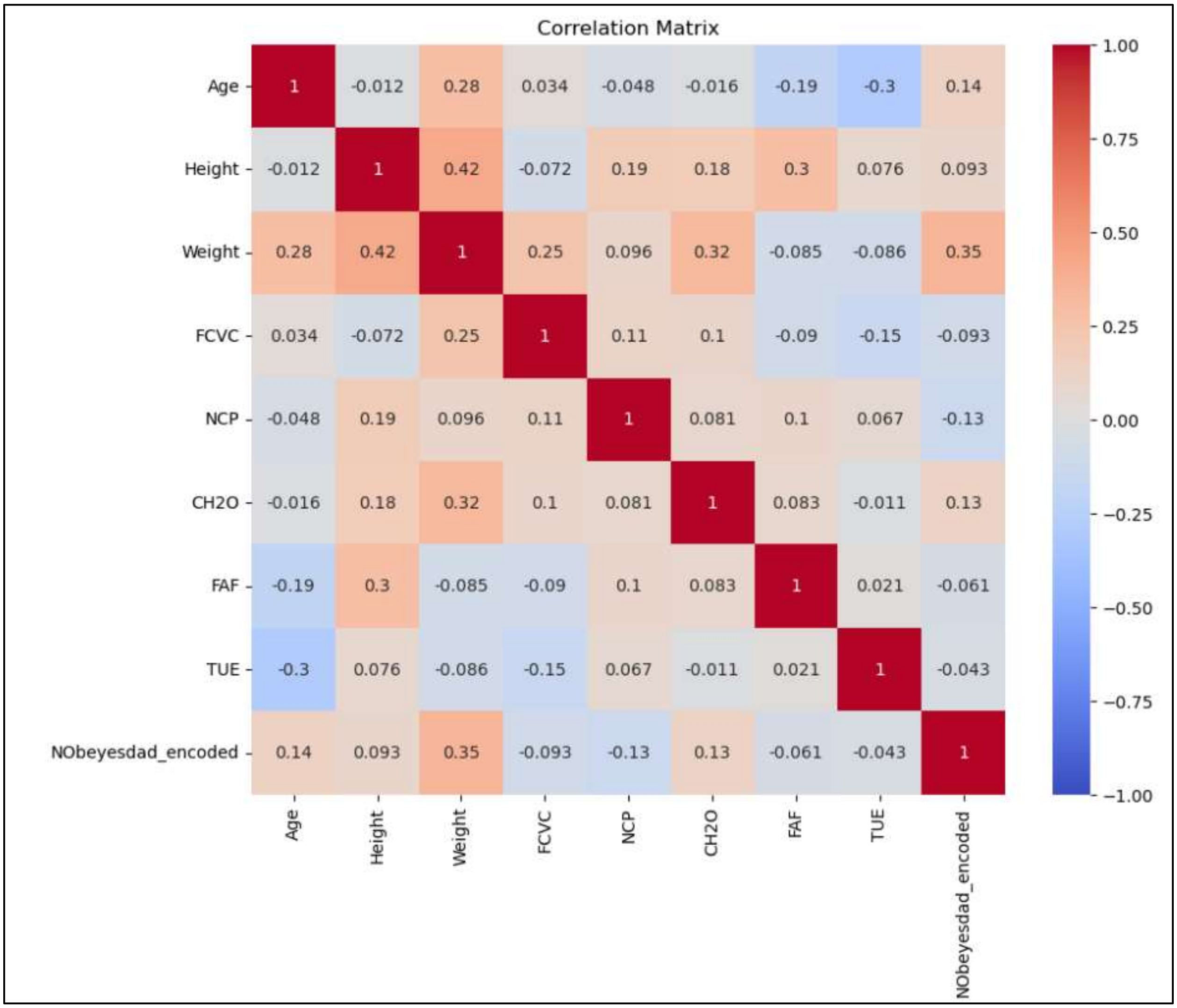

Box Plot of CH2O by NObeyesdad
CH2O
3.00
2.75
2.50
2.25
2.00
1.75
1.50
1.25
1.00
Overweight_Level_II
Normal_Weight
Insufficient_Weight
Obesity_Type_III
Obesity_Type_II
Overweight_Level_I
Obesity_Type_I
NObeyesdad


Box Plot of FAF by NObeyesdad
FAF
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Overweight_Level_II
Normal_Weight
Insufficient_Weight
Obesity_Type_III
Obesity_Type_II
Overweight_Level_I
Obesity_Type_I
NObeyesdad


Box Plot of TUE by NObeyesdad
TUE
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
Overweight_Level_II
Normal_Weight
Insufficient_Weight
Obesity_Type_III
Obesity_Type_II
Overweight_Level_I
Obesity_Type_I
NObeyesdad

Data Preprocessing

Correlation Matrix of Numerical Attributes With Target Variable



ANOVA Test

To check Correlation between Numerical Features and Target variable

Hypothesis:

- **Null Hypothesis (H0):** There is no significant difference in the means of the numerical attribute(s) across the different levels of the obesity category.
- **Alternative Hypothesis (H1):** There is at least one significant difference in the means of the numerical attribute(s) across the different levels of the obesity category.

	Feature	F-statistic	P-value	Significant
0	Age	962.936496	0.000000	True
1	Height	759.579663	0.000000	True
2	Weight	22867.945866	0.000000	True
3	FCVC	1551.918278	0.000000	True
4	NCP	300.259705	0.000000	True
5	CH2O	385.818727	0.000000	True
6	FAF	282.714681	0.000000	True
7	TUE	147.188575	0.000000	True

Chi-Squared Test for Categorical features

Hypothesis

Null Hypothesis (H_0): There is no significant association between the feature and the obesity category.

Alternative Hypothesis (H_1): There is a significant association between the feature and the obesity category.

	Feature	Chi2-statistic	P-value	Significant
0	Gender	7953.767544	0.000000	True
1	family_history_with_overweight	6423.317091	0.000000	True
2	FAVC	1553.629751	0.000000	True
3	CAEC	6897.329566	0.000000	True
4	SMOKE	216.300613	0.000000	True
5	SCC	1024.798467	0.000000	True
6	CALC	4013.082706	0.000000	True
7	MTRANS	2349.082568	0.000000	True

Variable Correlation and Feature Selection:

Correlation matrix is used for numerical features, chi-squared test is used for categorical features and Anova is used for numerical and target variable

- **Data Cleaning/Preprocessing for Classification**
  - **Standardization of numerical variables:** For standardization of features StandardScaler() function from sklearn.preprocessing is used to transform features to have a mean of 0 and standard deviation of 1. It is applied to apply equal weight to all features to avoid bias and improve model performance

$$z = \frac{x - \mu}{\sigma}$$

where:

- x is the original value of the feature.
- μ is the mean of the feature.
- σ is the standard deviation of the feature.
- z is the standardized value.

Variance of the features before scaling

```
Age        40.271313
Height      0.008706
Weight    685.977477
FCVC        0.285078
NCP         0.605344
CH2O        0.375712
FAF         0.723507
TUE         0.370792
dtype: float64
```

Variance of the features after scaling

```
Age       1.000474
Height    1.000474
Weight    1.000474
FCVC      1.000474
NCP       1.000474
CH2O      1.000474
FAF       1.000474
TUE       1.000474
dtype: float64
```

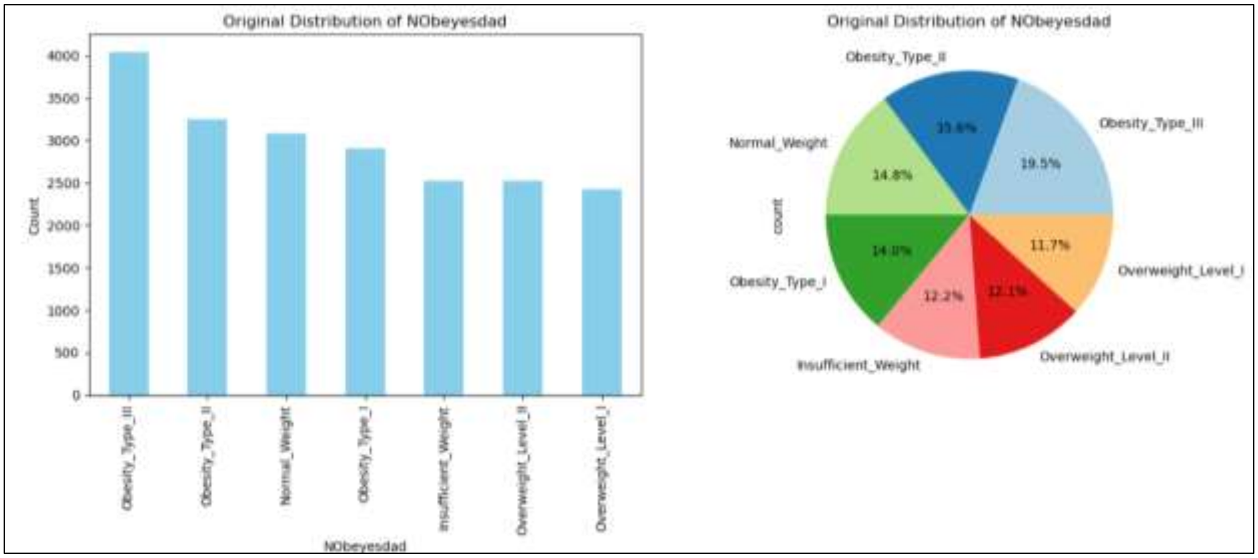
Encoding of categorical features

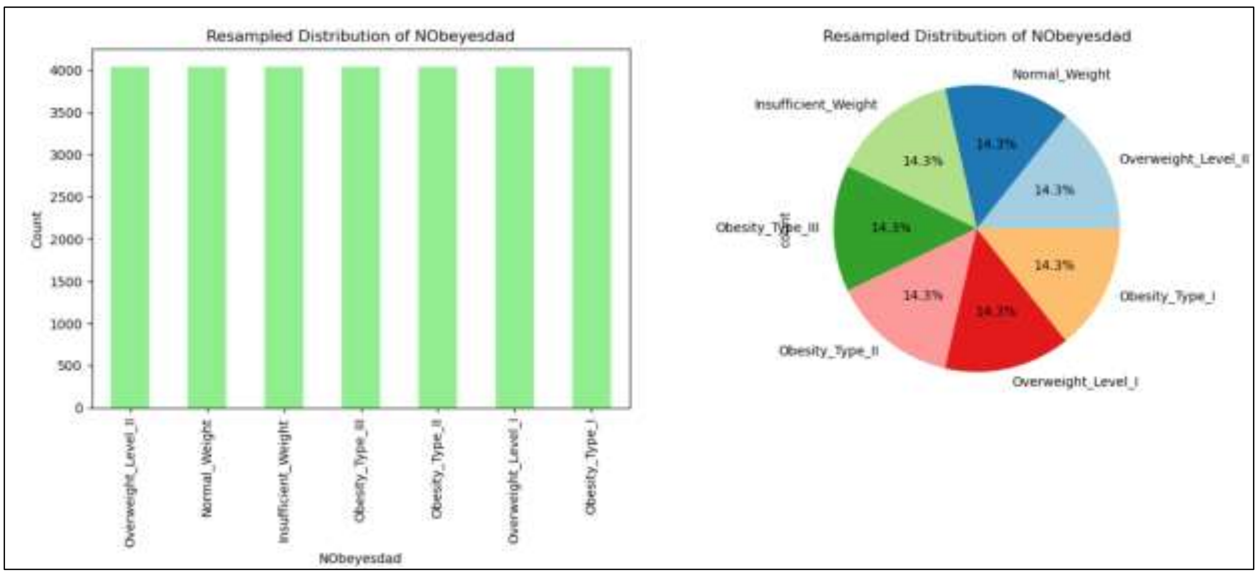
To transform categorical features into format that can be used by machine learning algorithms, following encoding was applied

1. Dropping unnecessary columns: Removing columns that are not needed for analysis or might be irrelevant (e.g., an ID column).
2. Handling missing values: SimpleImputer is used to fill missing values. For numerical features, the SimpleImputer is configured with the strategy 'median' imputation, to replace any missing value in the numerical features with the median of that column. For categorical features, the SimpleImputer is configured with the strategy 'most_frequent' to fill the missing values in categorical features with the most frequent value (mode) of that feature.
3. Binary Encoding: Mapping binary categorical data YES/NO or Male/Female, to 0 and 1 using map() function from pandas library.
4. For Ordinal Encoding: Mapping ordered categorical features to numerical values based on their order
5. One-Hot Encoding: To convert categorical features with multiple values into binary columns, each representing a unique value of the feature from sklearn.preprocessing.
6. Target Encoding: Using label encoder to convert categorical target values into numerical values that can be used by the models from sklearn.preprocessing.

Handling class imbalance:

SMOTE (Synthetic Minority Over-sampling Technique) was used to handle class imbalance. Based on the pie chart of the target variable “NObeyesdad”, the classes are imbalanced to a certain degree. The largest class (Obesity_Type_III) has around 19.5% of the total data, while the smallest class (Overweight_Level_I) has around 11.7%. Although the imbalance is not extreme, it is still significant enough that it might affect the performance of your classifiers, especially for the smaller classes. Logistic regression and SVM can be sensitive to class imbalance.


Original Distribution of NObeyesdad
Count
NObeyesdad
Obesity_Type_III
Obesity_Type_II
Normal_Weight
Obesity_Type_I
Insufficient_Weight
Overweight_Level_II
Overweight_Level_I
Original Distribution of NObeyesdad
count
19.5%
14.8%
12.2%
11.7%


Resampled Distribution of NObeyesdad
Count
NObeyesdad
Overweight_Level_II
Normal_Weight
Insufficient_Weight
Obesity_Type_III
Obesity_Type_II
Overweight_Level_I
Obesity_Type_I
Resampled Distribution of NObeyesdad
14.3%

-

Model Training and Evaluation

Five different models are used for this project

Decision Tree

A decision tree is a supervised learning algorithm used for classification and regression tasks, forming a tree-like model of decisions. It splits data into subsets based on input feature values, with internal nodes representing tests on attributes, branches representing outcomes of tests, and leaf nodes representing class labels or continuous values. The tree uses splitting criteria such as Gini Impurity, Entropy and Information Gain to determine the best attribute for splitting. Decision trees are easy to interpret, can capture non-linear relationships, and provide feature importance rankings, but they are prone to overfitting and instability, which can be mitigated through pruning techniques.

$$Gini(t) = 1 - \sum_{i=1}^{C} (pi^2)$$

$$Entropy(t) = -\sum_{i=1}^{C} (pi) log_2 (pi)$$

$$Information\ Gain(S, A) = Entropy(S) - \sum v \in Values(A) \frac{|S_v|}{|S|} \text{Entropy}(Sv)$$

Logistic Regression

Logistic regression is a widely used statistical method for binary and multi-class classification problems. It models the probability of a categorical dependent variable based on one or more predictor variables. For binary outcomes, it uses the logistic function to transform a linear combination of predictors into a probability. When extended to multi-class classification, the approach can be adapted using methods such as One-vs-Rest (OvR) or Softmax regression (Multinomial Logistic Regression).

In the context of obesity estimation, where there are multiple obesity levels (e.g., normal weight, overweight, obesity type I, etc.), multinomial logistic regression is employed. This method allows for the classification of an outcome variable with more than two categories. It predicts the probability distribution over all classes, assigning each observation to the class with the highest probability.

Support Vector Machine

Support Vector Machine (SVM) is a robust and versatile classification algorithm that aims to find the optimal hyperplane that best separates data into different classes with the maximum margin. For binary classification, SVM identifies this hyperplane based on the support vectors—data points closest to the hyperplane. In multi-class classification, SVM extends this approach using strategies like One-vs-Rest (OvR), where a separate classifier is trained for each class, or One-vs-One (OvO), where classifiers are trained for each pair of classes. SVM can handle both linear and non-linear problems through the use of kernel functions, such as linear, polynomial, and Radial Basis Function (RBF) kernels, which map the input data into higher-dimensional spaces for improved separation. This makes SVM particularly effective for high-dimensional and complex datasets.

Random Forest

Random Forest is an ensemble learning technique that builds multiple decision trees and combines their predictions to enhance classification accuracy and reduce overfitting. By averaging the results of numerous trees, each trained on different subsets of the data and features, it provides a more stable and accurate model. This approach is well-suited for complex datasets with many features, making it effective for tasks like obesity estimation. Random Forest is particularly effective for handling complex datasets with high dimensionality and interactions between features, making it a powerful tool for classification tasks

XGBooost

XGBoost (Extreme Gradient Boosting) is an advanced ensemble learning algorithm that builds a series of decision trees sequentially, where each tree corrects the errors of the previous ones. It is highly efficient due to its parallel processing capabilities, tree pruning, and handling of missing values. Its effectiveness is seen in its ability to capture complex patterns and interactions between features, making it a top performer in many machine learning competitions. XGBoost is also known for its stability, as it incorporates regularization techniques to prevent overfitting, ensuring that the model generalizes well to new data.

For model training model, features and target were separated for both training and test datasets custom scaler and custom transformer were created for data preprocessing and pipelines for different classifiers were defined to streamline the process of training different model consistently to prevent data leakage and bias.

Evaluation Metrics

Accuracy

Accuracy is a measure of the overall correctness of a classification model. It represents the proportion of correctly classified instances (both true positives and true negatives) out of the total instances.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Where:

- TP = True Positives (correctly predicted positive cases)
- TN = True Negatives (correctly predicted negative cases)
- FP = False Positives (incorrectly predicted positive cases)
- FN = False Negatives (incorrectly predicted negative cases)

Precision

Precision, also known as positive predictive value, measures the accuracy of positive predictions. It is the proportion of true positive predictions among all positive predictions made by the model.

$$Precision = \frac{TP}{TP + FP}$$

Where:

- TP = True Positives
- FP = False Positives

Recall

Recall, also known as sensitivity or true positive rate, measures the ability of the model to identify all relevant positive cases. It is the proportion of true positive predictions among all actual positive instances.

$$Recall = \frac{TP}{TP + FN}$$

Where:

- TP = True Positives
- FN = False Negatives

F1 Score

The F1 score is the harmonic mean of precision and recall. It provides a single metric that balances the trade-off between precision and recall, especially useful when dealing with imbalanced datasets.

$$F1\ Score = 2\ x\ \frac{Precision\ x\ Recall}{Precision + \ Recall}$$

Where:

- Precision is the positive predictive value
- Recall is the true positive rate

AUC ROC

AUC ROC is a performance measurement for classification problems at various threshold settings for binary and multi-class classification problems. **ROC** stands for Receiver Operating Characteristic, and **AUC** stands for Area Under the ROC Curve.

ROC Curve

The ROC curve is a graphical representation of a classifier's performance across different thresholds. It plots two parameters:

- **True Positive Rate (TPR)**: Also known as sensitivity or recall.

$$TPR = \frac{TP}{TP + FN}$$

- **False Positive Rate (FPR)**: The proportion of negative instances that are incorrectly classified as positive.

$$FPR = \frac{FP}{FP + TN}$$

The ROC curve is created by plotting the TPR against the FPR at various threshold settings. Each point on the ROC curve represents a different threshold value, with the top-left corner representing a perfect classifier and the diagonal line representing a random guess.

AUC (Area Under the ROC Curve)

AUC measures the entire two-dimensional area underneath the entire ROC curve from (0,0) to (1,1). The value of AUC ranges from 0 to 1, where:

- **AUC = 1**: Perfect classifier. The model correctly classifies all positive and negative instances.
- **AUC = 0.5**: No discrimination ability, equivalent to random guessing.
- **AUC < 0.5**: Indicates that the model is performing worse than random guessing, which suggests that the model's predictions are inversely correlated with the actual classifications.

Matthews Correlation Coefficient (MCC)

The Matthews Correlation Coefficient (MCC) is a performance metric for binary classification that takes into account true and false positives and negatives. It is regarded as a balanced measure which can be used even if the classes are of very different sizes. The MCC returns a value between -1 and +1:

- **+1** indicates a perfect prediction.
- **0** indicates a prediction no better than random guessing.
- **-1** indicates a perfect negative prediction (total disagreement between prediction and observation).

The MCC is calculated using the following formula:

$$MCC = \frac{(TPxTN) - (FPxFN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Where:

- TP = True Positives (correctly predicted positive cases)
- TN = True Negatives (correctly predicted negative cases)
- FP = False Positives (incorrectly predicted positive cases)
- FN = False Negatives (incorrectly predicted negative cases)

Results

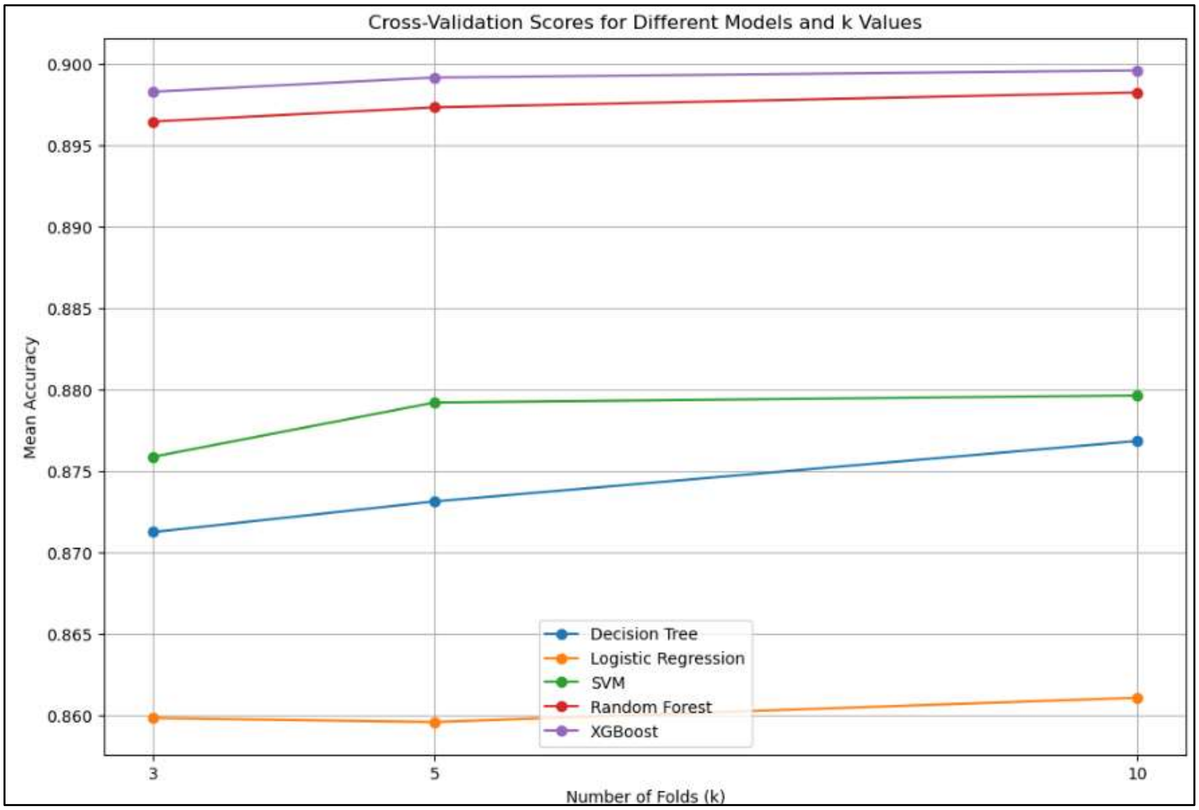
Hold-out Cross Validation

Training data was split into train (80%) and validation (20%) sets then models were trained and tested on validation data and parameters were tweaked and finally prediction on the test data was performed.

Models	Validation Accuracy	Test Accuracy	Precision	Recall	F1-score
Decision Tree	0.84	0.89	0.89	0.89	0.89
Logistic Regression	0.86	0.90	0.91	0.91	0.91
Support Vector Machine	0.86	0.90	0.91	0.90	0.90
Random Forest	0.89	0.94	0.94	0.94	0.94
XGBoost	0.90	0.95	0.94	0.94	0.94

Cross Validation to choose K value

Performing Stratified cross validation for different k values (3, 5, 10) and plot the results according to mean accuracy to evaluate model performance across different cross-validation strategies and select the best performing models.




```
Number of Folds: 3, CV Mean Accuracy: 0.8712 Decision Tree
Number of Folds: 3, CV Mean Accuracy: 0.8598 Logistic Regression
Number of Folds: 3, CV Mean Accuracy: 0.8759 SVM
Number of Folds: 3, CV Mean Accuracy: 0.8964 Random Forest
Number of Folds: 3, CV Mean Accuracy: 0.8983 XGBoost
Number of Folds: 5, CV Mean Accuracy: 0.8731 Decision Tree
Number of Folds: 5, CV Mean Accuracy: 0.8596 Logistic Regression
Number of Folds: 5, CV Mean Accuracy: 0.8792 SVM
Number of Folds: 5, CV Mean Accuracy: 0.8973 Random Forest
Number of Folds: 5, CV Mean Accuracy: 0.8991 XGBoost
Number of Folds: 10, CV Mean Accuracy: 0.8768 Decision Tree
Number of Folds: 10, CV Mean Accuracy: 0.8611 Logistic Regression
Number of Folds: 10, CV Mean Accuracy: 0.8796 SVM
Number of Folds: 10, CV Mean Accuracy: 0.8982 Random Forest
Number of Folds: 10, CV Mean Accuracy: 0.8996 XGBoost
```

Based on the cross-validation scores plotted for different values of *k* (3, 5, and 10):

1. **Consistency**:
   - Random Forest and XGBoost show very consistent performance across different *k* values, indicating they are stable.
   - SVM and Decision Tree show a slight improvement as *k* increases.
   - Logistic Regression shows minimal change, indicating it is less sensitive to the choice of *k*.
2. **Mean Accuracy**:
   - XGBoost has the highest mean accuracy across all *k* values, indicating it performs best overall.
   - Random Forest follows closely, maintaining high and consistent accuracy.

Conclusion: For maximum performance and stability k=10 would be a good choice as it provides most robust and less biased estimate of the model. If computational resources are a concern then k=5 would be a balanced choice between performance and computational cost. For my project I would choose k=10.

Mean Accuracy of Stratified Cross Validation after tweaking all models

Models	Mean Accuracy
Decision Tree	0.88
Logistic Regression	0.86
Support Vector Machine	0.88
Random Forest	0.90
XGBoost	0.90

GridSearch Cross validation for Hyperparameter tuning

Models	Best Parameters	CV Accuracy
Decision Tree	max_depth: 10	0.88
	min_samples_leaf: 10	
	min_samples_split: 2	
Logistic Regression	C: 10	0.86
	Solver: newton-cg	

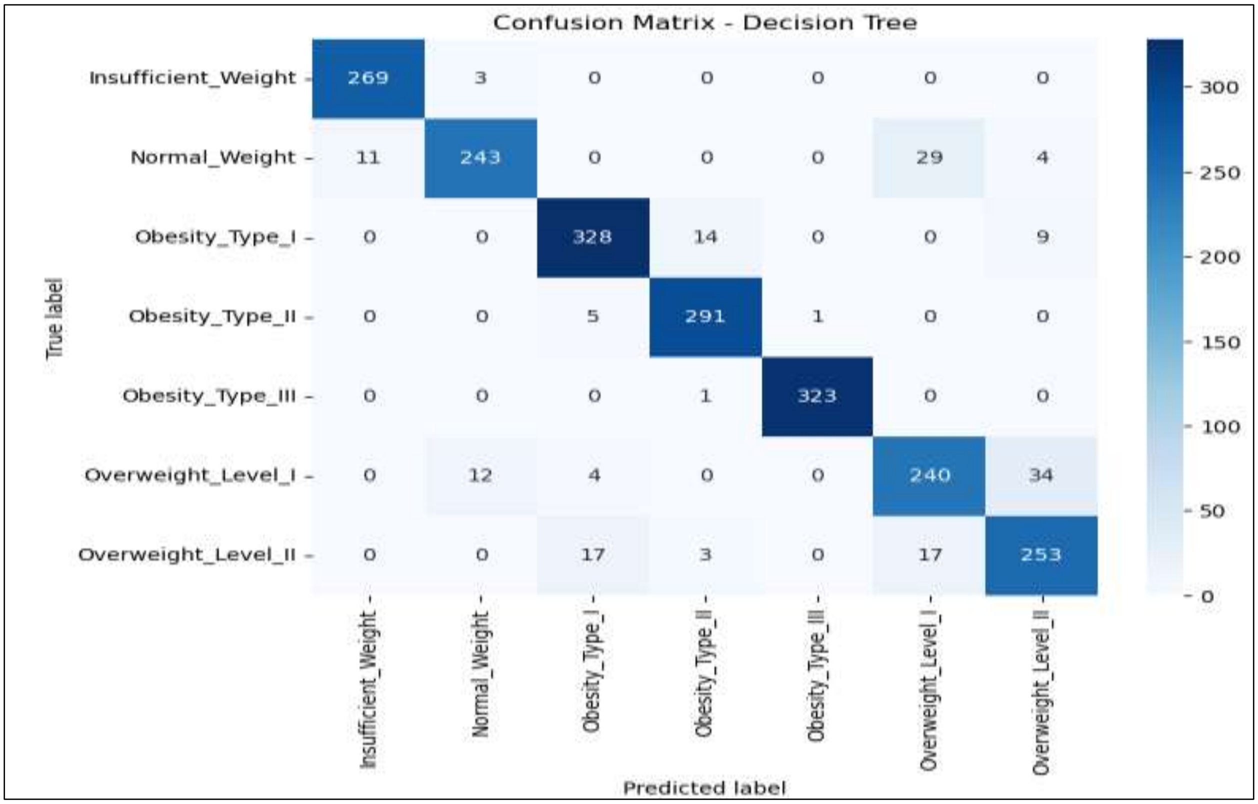
Support Vector Machine	C: 1	0.88
	gamma: scale	
	Kernel: rbf	
Random Forest	Max_depth: 30	0.90
	Min_samples_split: 10	
	N_estimators: 300	
XGBoost	Learning_rate: 0.1	0.90
	Max_depth: 3	
	N_estimators: 300	

Final Results

Models	Precision	Recall	F1-score	Accuracy	MCC	AUC	Training time (sec)
Decision Tree	0.92	0.92	0.92	0.92	0.91	0.99	1.34
Logistic Regression	0.92	0.92	0.92	0.91	0.90	0.99	4.07
Support Vector Machine	0.93	0.93	0.93	0.93	0.92	1.0	14.50
Random Forest	0.94	0.94	0.94	0.94	0.93	1.0	13.73
XGBoost	0.95	0.95	0.95	0.95	0.93	1.0	6.24

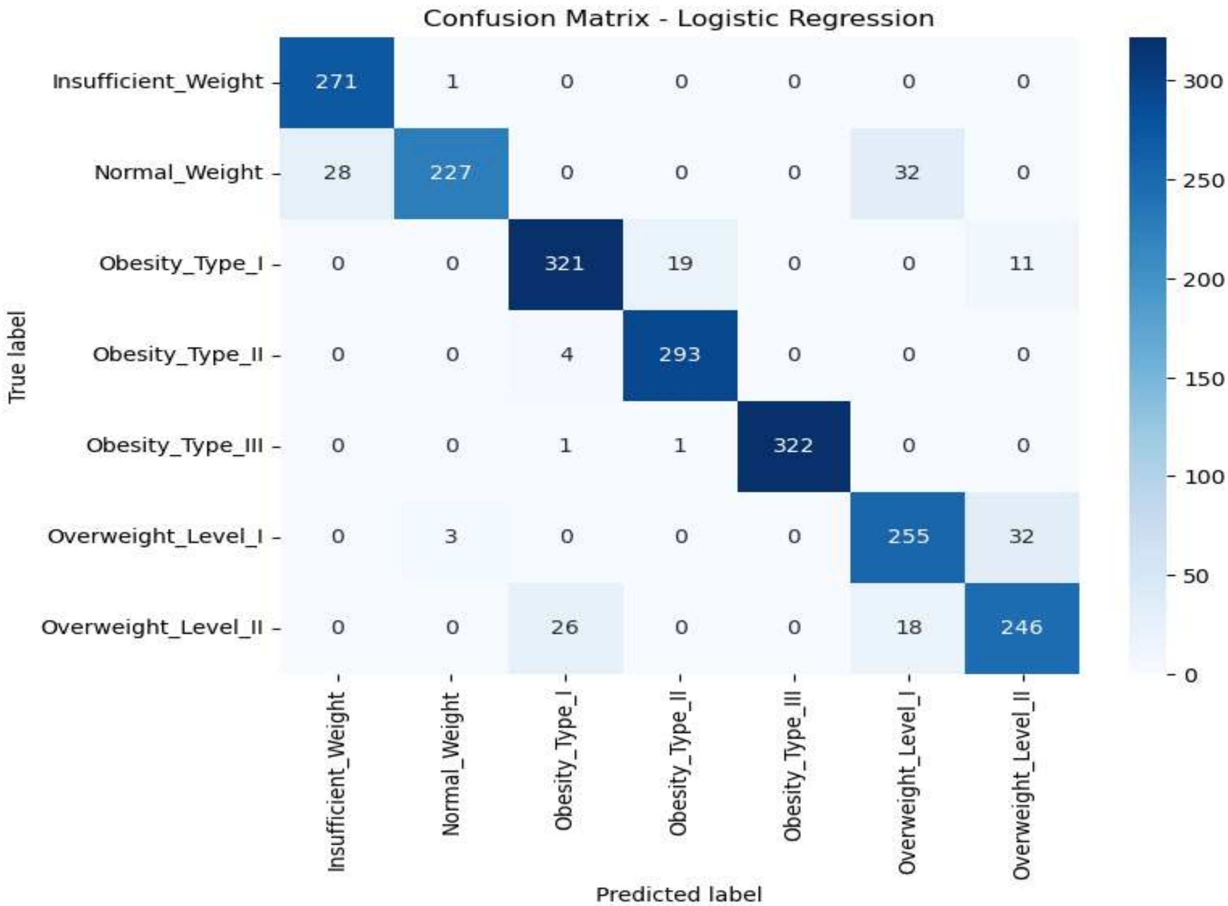
Confusion Matrix and Summary for all models

Confusion Matrix and its Summary (Decision Tree)



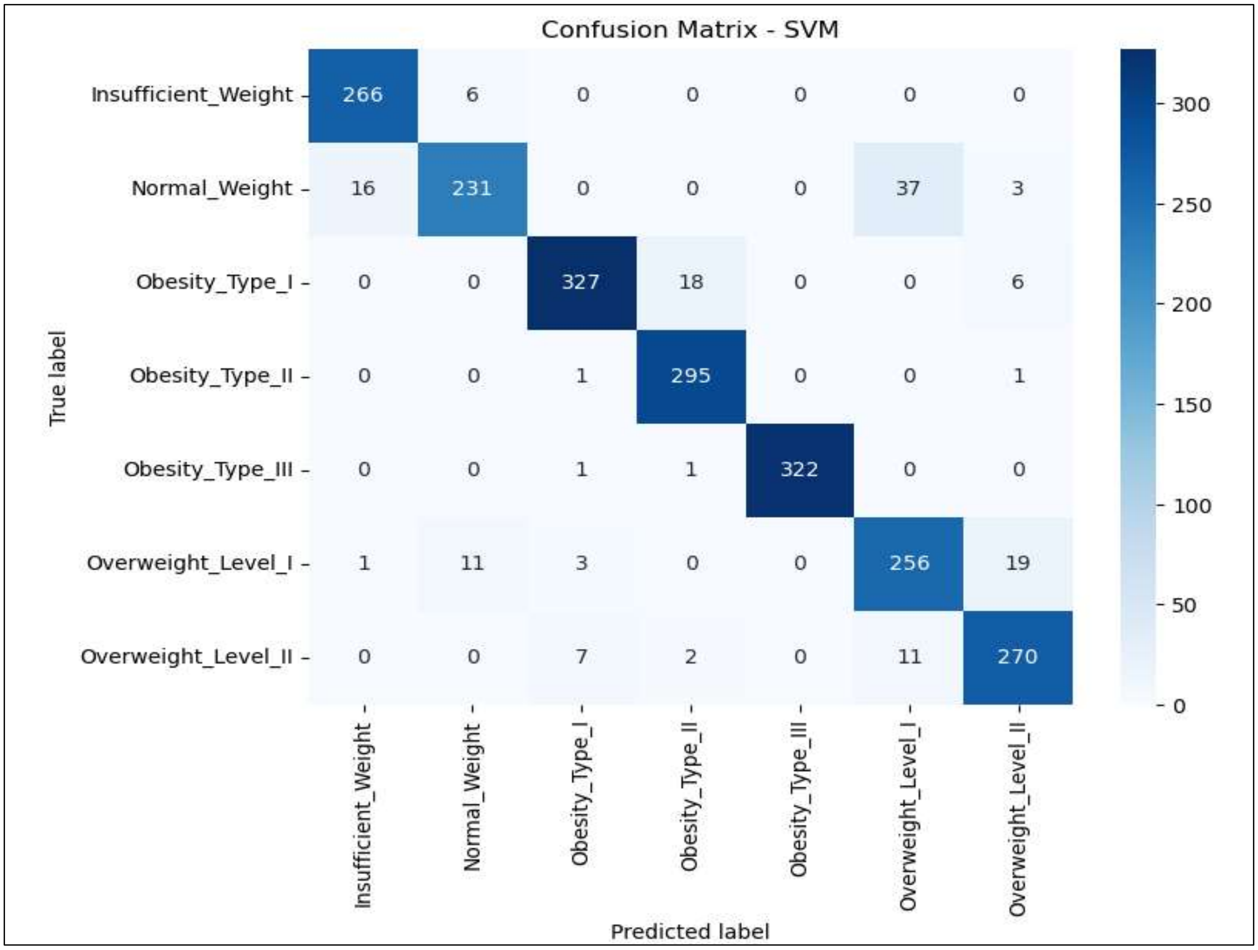
Classes	TP	FP	FN	TN
Insufficient Weight	269	11	3	1828
Normal Weight	243	15	44	1809
Obesity Type I	328	26	23	1734
Obesity Type II	291	18	6	1796
Obesity Type III	323	1	1	1786
Overweight Level I	240	46	50	1775
Overweight Level II	253	47	37	1774

Confusion Matrix and its Summary (Logistic Regression)



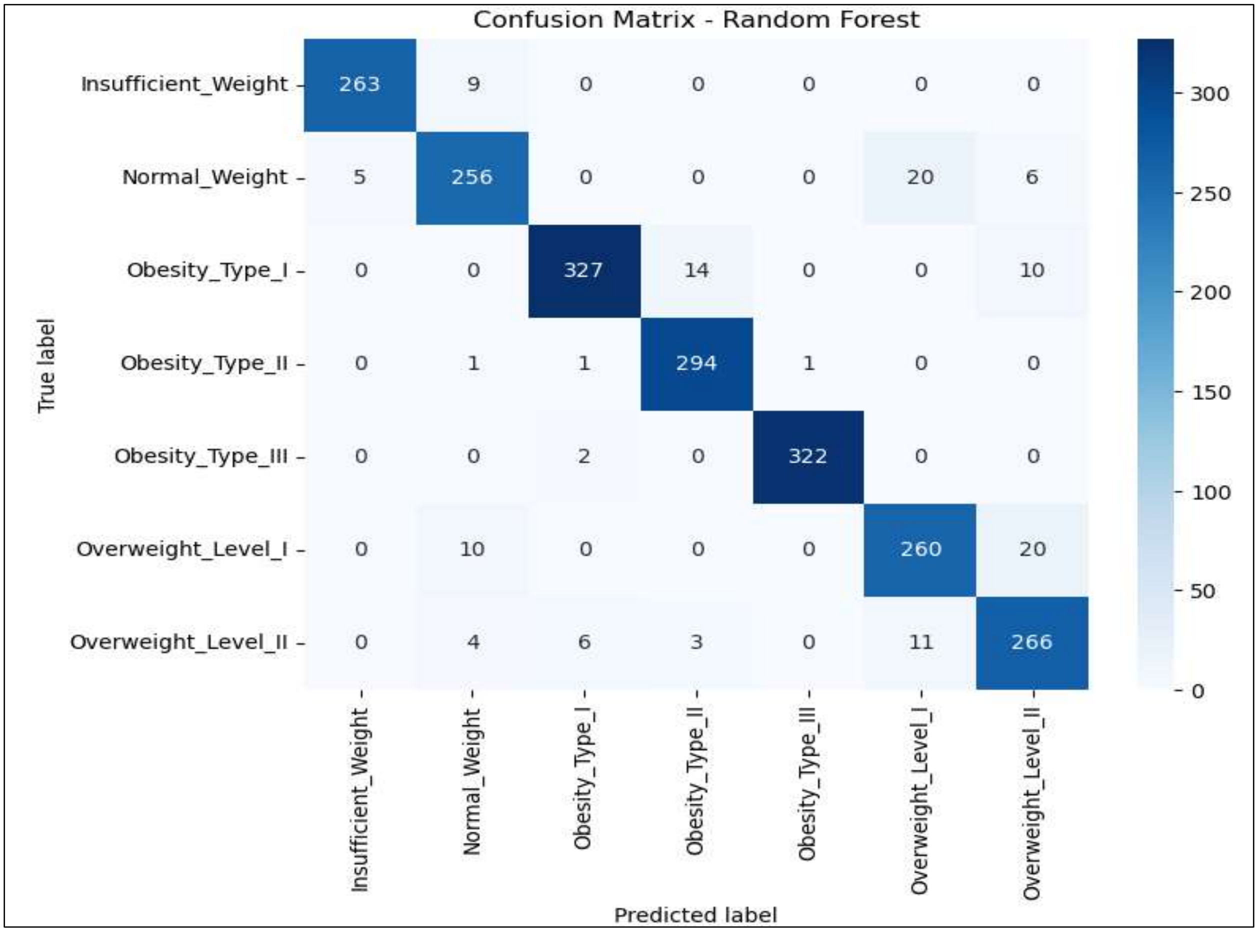
Classes	TP	FP	FN	TN
Insufficient Weight	271	28	1	1811
Normal Weight	227	4	60	1820
Obesity Type I	321	27	30	1729
Obesity Type II	322	20	4	1794
Obesity Type III	322	0	2	1787
Overweight Level I	255	50	35	1771
Overweight Level II	246	43	44	1778

Confusion Matrix and its Summary (SVM)



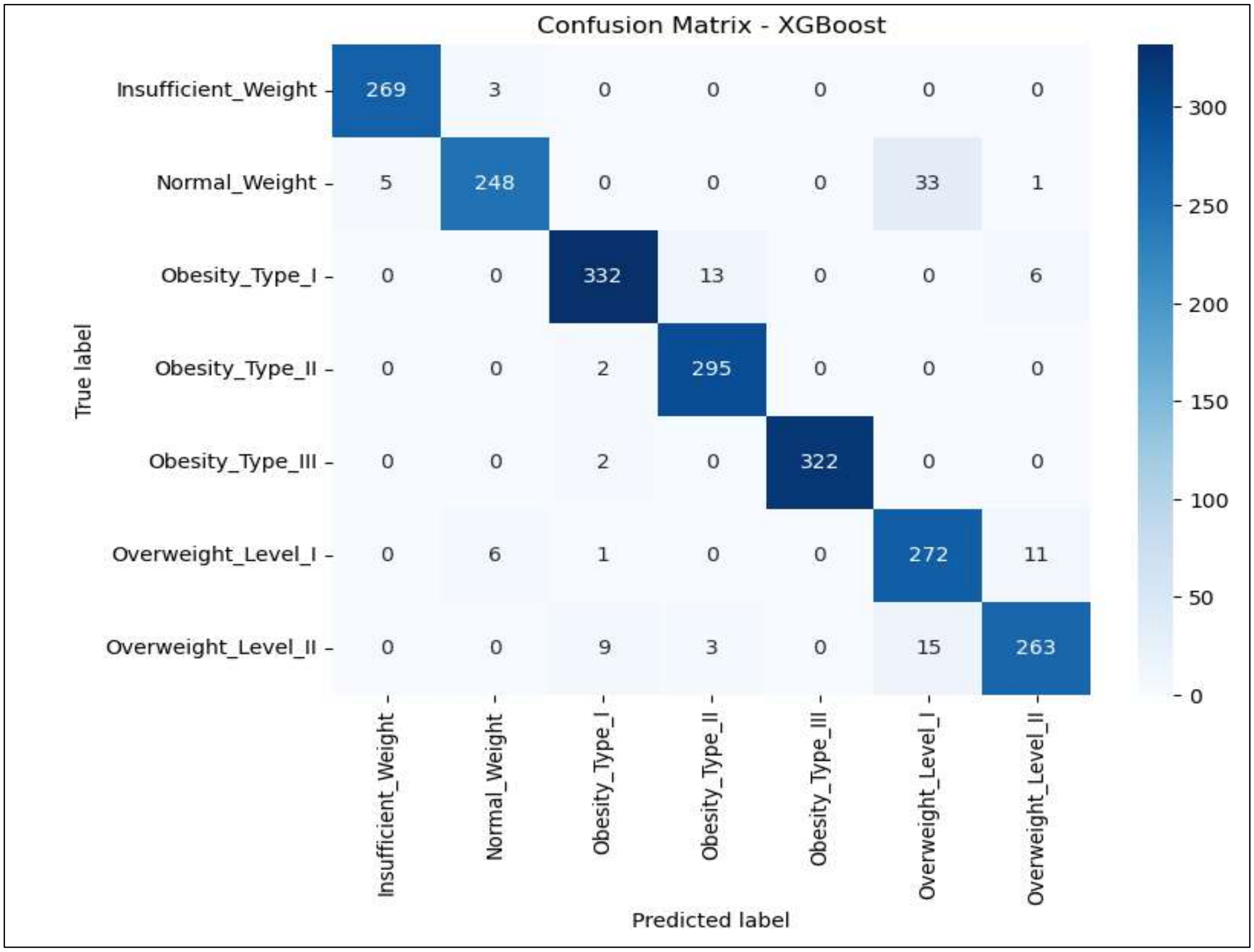
Classes	TP	FP	FN	TN
Insufficient Weight	265	14	7	1825
Normal Weight	232	14	55	1810
Obesity Type I	321	10	30	1750
Obesity Type II	295	23	2	1791
Obesity Type III	322	0	2	1787
Overweight Level I	256	50	34	1771
Overweight Level II	270	39	20	1782

Confusion Matrix and its Summary (Random Forest)



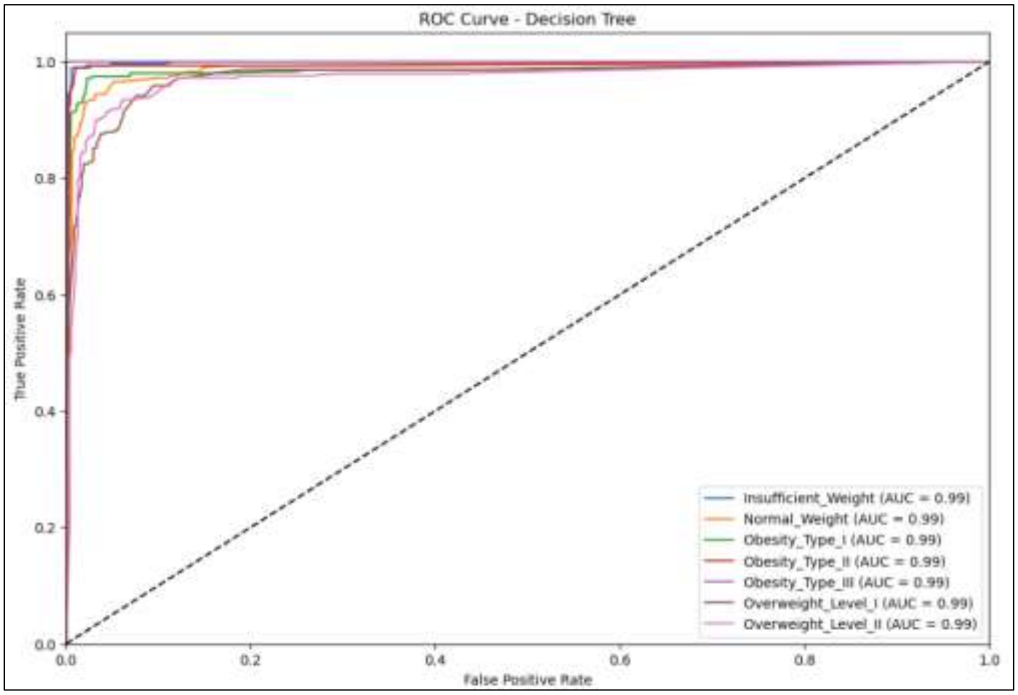
	TP	FP	FN	TN
Insufficient Weight	263	5	9	1834
Normal Weight	256	24	31	1800
Obesity Type I	327	9	24	1751
Obesity Type II	294	17	3	1797
Obesity Type III	322	1	2	1786
Overweight Level I	260	31	30	1790
Overweight Level II	266	36	24	1785

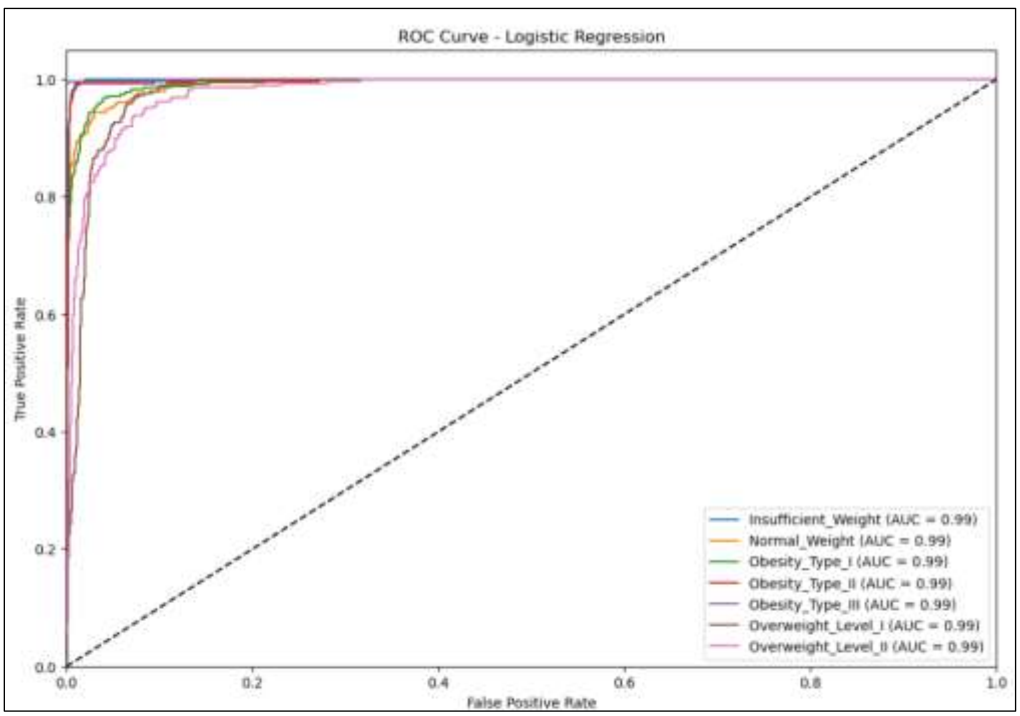
Confusion Matrix and its Summary (XGBoost)

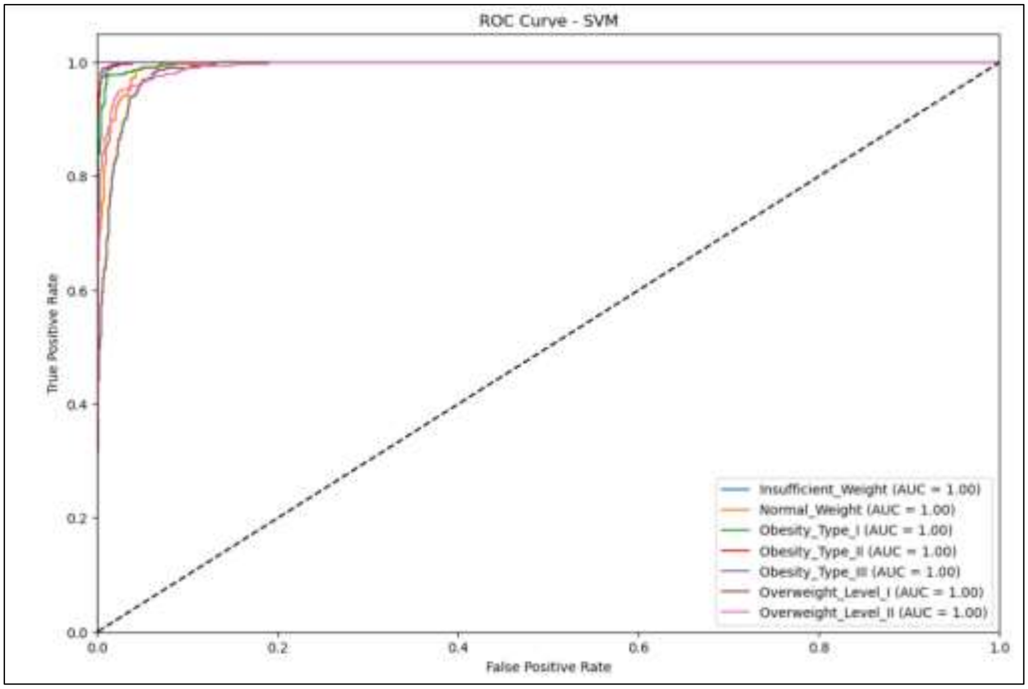


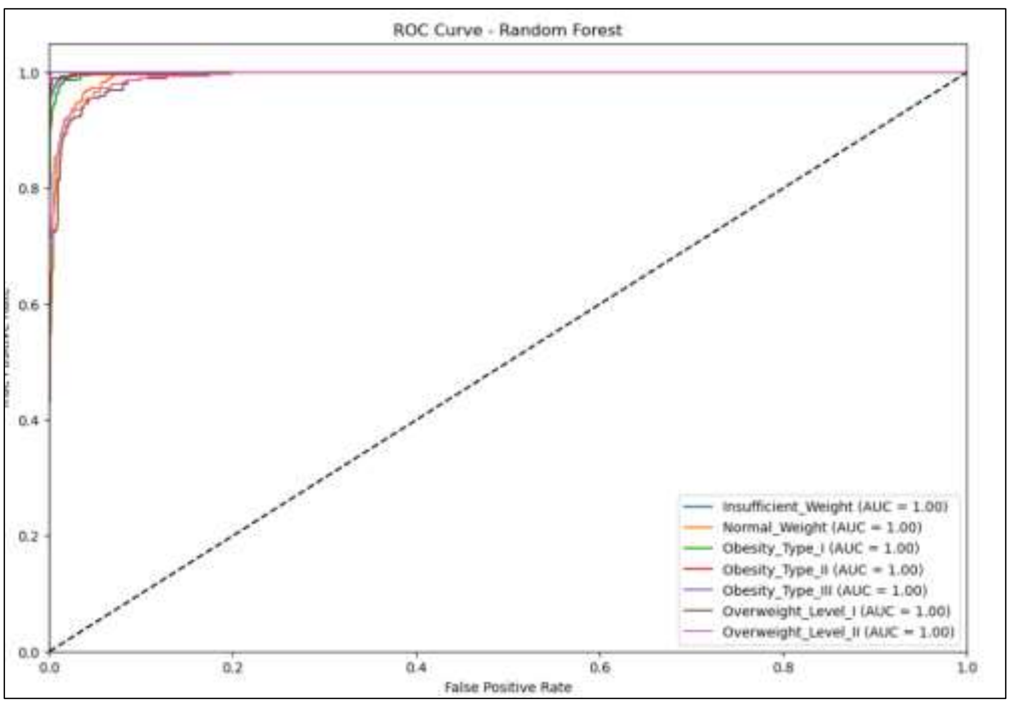
Models	TP	FP	FN	TN
Insufficient Weight	269	7	3	1832
Normal Weight	248	10	39	1814
Obesity Type I	328	18	23	1742
Obesity Type II	294	20	3	1794
Obesity Type III	322	0	2	1787
Overweight Level I	271	48	19	1773
Overweight Level II	257	19	33	1802

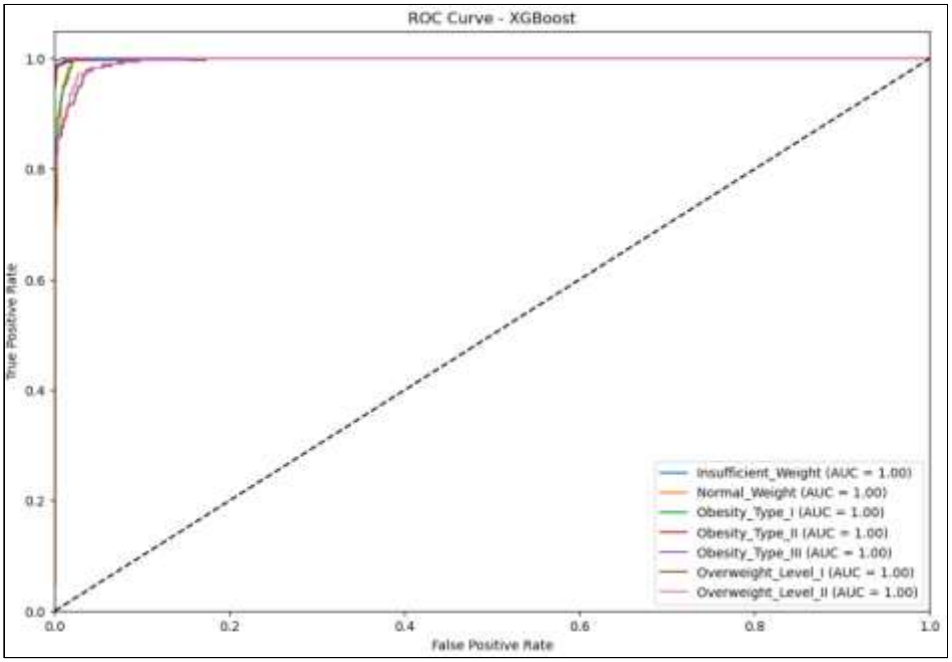
ROC curve for all models



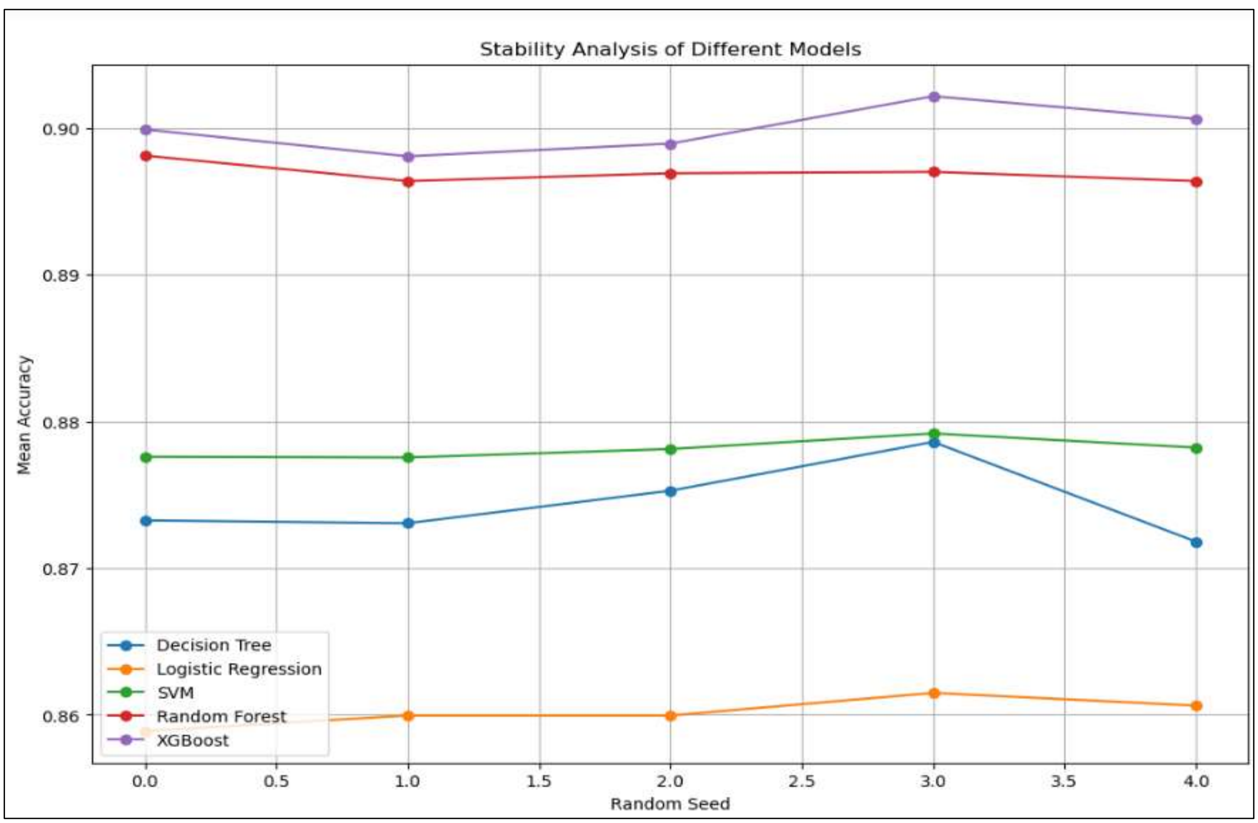



ROC Curve - SVM
True Positive Rate
False Positive Rate
Insufficient_Weight (AUC = 1.00)
Normal_Weight (AUC = 1.00)
Obesity_Type_I (AUC = 1.00)
Obesity_Type_II (AUC = 1.00)
Obesity_Type_III (AUC = 1.00)
Overweight_Level_I (AUC = 1.00)
Overweight_Level_II (AUC = 1.00)


ROC Curve - Random Forest
False Positive Rate
Insufficient_Weight (AUC = 1.00)
Normal_Weight (AUC = 1.00)
Obesity_Type_I (AUC = 1.00)
Obesity_Type_II (AUC = 1.00)
Obesity_Type_III (AUC = 1.00)
Overweight_Level_I (AUC = 1.00)
Overweight_Level_II (AUC = 1.00)



Stability Analysis of different models



In stability analysis, XGBoost and Random Forest emerge as the most robust and high-performing models for your dataset, while Logistic Regression, despite being very stable, shows lower accuracy. Decision Tree and SVM provide a balance of moderate performance and stability. This analysis helps in selecting the most reliable model for practical deployment.

Knowledge Induction using Apriori method

Knowledge induction refers to the process of deriving new knowledge or insights from existing data. This often involves identifying patterns, relationships, or rules that were not explicitly evident before analysis. The Apriori algorithm is employed to identify frequent itemsets, which are combinations of attributes that appear together frequently in the data. From these frequent itemsets, association rules are generated to elucidate the relationships between different attributes. An association rule is expressed in the form A→BA where A (the antecedent) implies B (the consequent). The strength of these rules is measured using two key metrics: support and confidence. Support indicates the proportion of transactions in the dataset that contain both the antecedent and the consequent, reflecting the overall significance of the rule. Confidence measures the likelihood that the consequent occurs in transactions where the antecedent is present, providing an indication of the rule's reliability. By focusing on rules with high support and confidence, the project aims to derive actionable insights and deepen the understanding of the underlying data patterns.

For this project I am using mlxtend.preprocessing and mlxtend.frequent_patterns libraries for data preprocessing and generation of association rules.

Data Processing for Association Rules

The data processing for Association Rules is done by rounding and converting some numerical variables (such as FCVC, NCP, CH2O, FAF, TUE) from float to integer. Mapping was done for one column (family_history_with_overweight) for yes converted to Overweight_FH=yes and no converted to Overweight_FH=no to understand the rules properly.

Other transformation is done for some columns (FCVC, NCP, CH2O, FAF, TUE, CAEC, CALC, FAVC, SMOKE, SCC, Age, Height, Weight) to convert numerical or categorical values in the specified columns to a formatted string that includes the column name to understand the rules properly. The dataframe is converted to list of transactions and finally transactions are encoded using TransactionEncoder() to apply apriori

Apriori Algorithm

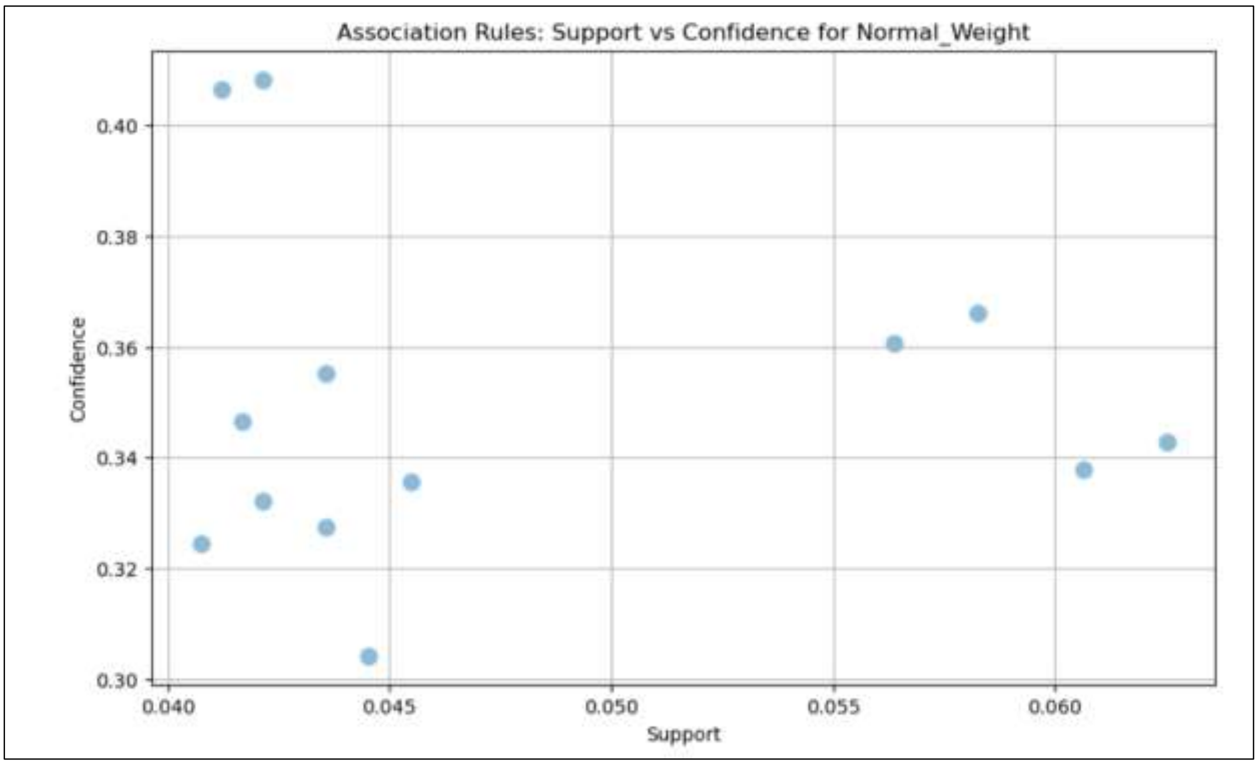
Apriori algorithm is applied and association rules were generated and filtered out for target = Normal_Weight with support threshold of 0.04 and confidence threshold of 0.3

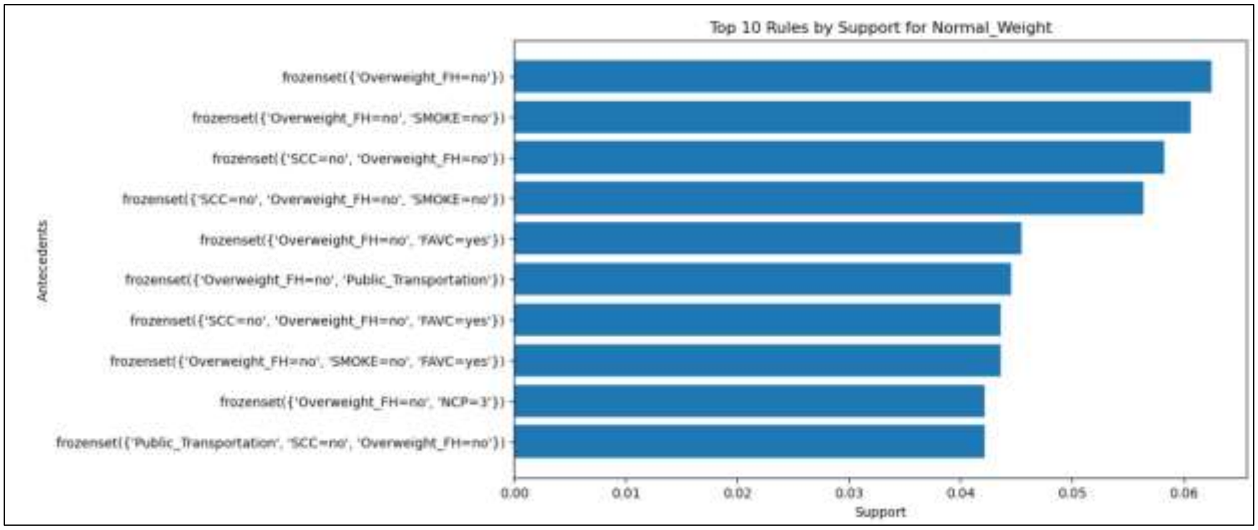
Association Rules for Normal Weight

Antecedents (LHS)	Consequents (RHS)	Support	Confidence
Overweight_FH=no	Normal_Weight	0.0625	0.3429
Overweight_FH=no, FAVC=yes	Normal_Weight	0.0455	0.3357
Overweight_FH=no, NCP=3	Normal_Weight	0.0422	0.4083
Overweight_FH=no, Public_Transportation	Normal_Weight	0.0445	0.3042
SCC=no, Overweight_FH=no	Normal_Weight	0.0583	0.3661
Overweight_FH=no, SMOKE=no	Normal_Weight	0.0606	0.3377
SCC=no, Overweight_FH=no, FAVC=yes	Normal_Weight	0.0436	0.3552
Overweight_FH=no, SMOKE=no, FAVC=yes	Normal_Weight	0.0436	0.3274
Overweight_FH=no, SMOKE=no, NCP=3	Normal_Weight	0.0412	0.4065
Public_Transportation, SCC=no, Overweight_FH=no	Normal_Weight	0.0422	0.3321

SCC=no, Overweight_FH=no, SMOKE=no	Normal_Weight	0.0564	0.3606
SCC=no, Overweight_FH=no, SMOKE=no, FAVC=yes	Normal_Weight	0.0417	0.3465
SCC=no, Overweight_FH=no, Public_Transportation, SMOKE=no	Normal_Weight	0.0407	0.3245

Scatter Plot for Normal_Weight Association Rules





The association rules were generated and filtered out for another target = Obesity_Type_III with support threshold of 0.15 and confidence threshold of 0.95

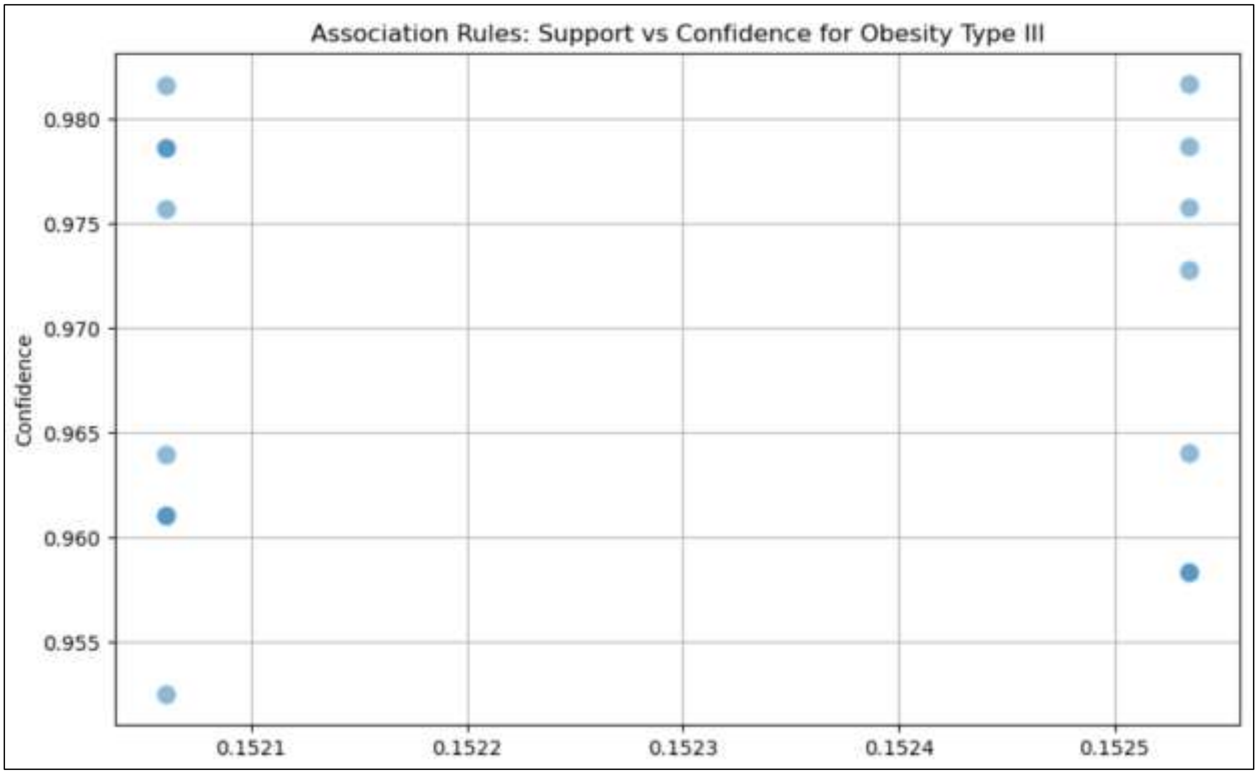
Association Rules for Obesity Type III

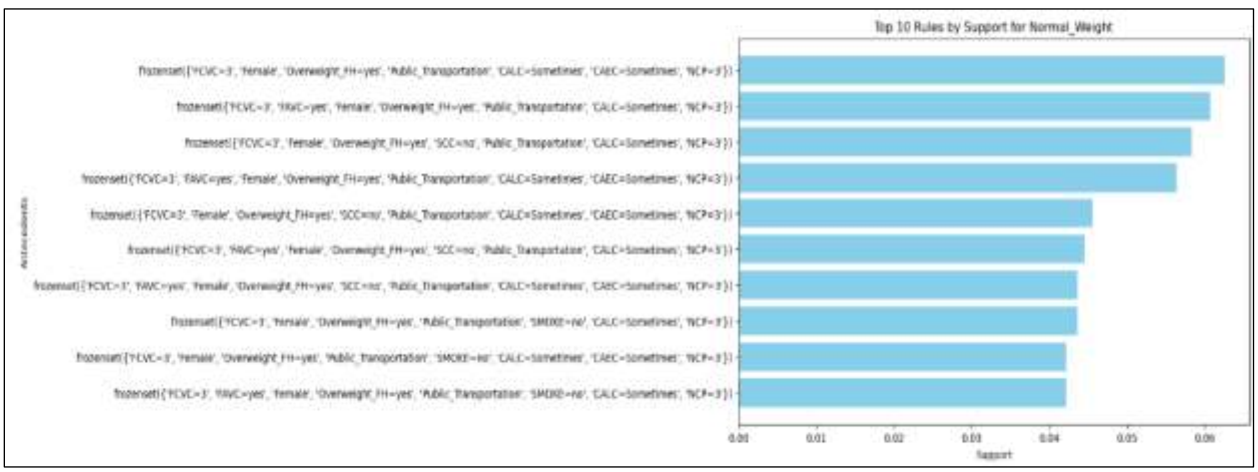
Antecedents (LHS)	Consequents (RHS)	Support	Confidence
FCVC=3, Female, Overweight_FH=yes, Public_Transportation, CALC=Sometimes, CAEC=Sometimes, NCP=3	Obesity_Type_III	0.1525	0.9583
FCVC=3, FAVC=yes, Female, Overweight_FH=yes, Public_Transportation, CALC=Sometimes, NCP=3	Obesity_Type_III	0.1525	0.9728
FCVC=3, Female, Overweight_FH=yes, SCC=no, Public_Transportation, CALC=Sometimes, NCP=3	Obesity_Type_III	0.1525	0.9583

FCVC=3, Female, Overweight_FH=yes, Public_Transportation, SMOKE=no, CALC=Sometimes, NCP=3	Obesity_Type_III	0.1521	0.9525
FCVC=3, FAVC=yes, Female, Overweight_FH=yes, Public_Transportation, CALC=Sometimes, CAEC=Sometimes, NCP=3	Obesity_Type_III	0.1525	0.9787
FCVC=3, Female, Overweight_FH=yes, SCC=no, Public_Transportation, CALC=Sometimes, CAEC=Sometimes, NCP=3	Obesity_Type_III	0.1525	0.9641
FCVC=3, Female, Overweight_FH=yes, Public_Transportation, SMOKE=no, CALC=Sometimes, CAEC=Sometimes, NCP=3	Obesity_Type_III	0.1521	0.9611
FCVC=3, FAVC=yes, Female, Overweight_FH=yes, SCC=no, Public_Transportation, CALC=Sometimes, NCP=3	Obesity_Type_III	0.1525	0.9758
FCVC=3, FAVC=yes, Female, Overweight_FH=yes, Public_Transportation, SMOKE=no, CALC=Sometimes, NCP=3	Obesity_Type_III	0.1521	0.9757
FCVC=3, Female, Overweight_FH=yes, SCC=no, Public_Transportation,	Obesity_Type_III	0.1521	0.9611

SMOKE=no, CALC=Sometimes, NCP=3			
FCVC=3, FAVC=yes, Female, Overweight_FH=yes, SCC=no, Public_Transportation, CALC=Sometimes, CAEC=Sometimes, NCP=3	Obesity_Type_III	0.1525	0.9817
FCVC=3, FAVC=yes, Female, Overweight_FH=yes, Public_Transportation, SMOKE=no, CALC=Sometimes, CAEC=Sometimes, NCP=3	Obesity_Type_III	0.1521	0.9787
FCVC=3, Female, Overweight_FH=yes, SCC=no, Public_Transportation, SMOKE=no, CALC=Sometimes, CAEC=Sometimes, NCP=3	Obesity_Type_III	0.1521	0.9640
FCVC=3, FAVC=yes, Female, Overweight_FH=yes, SCC=no, Public_Transportation, SMOKE=no, CALC=Sometimes, NCP=3	Obesity_Type_III	0.1521	0.9787
FCVC=3, FAVC=yes, Female, Overweight_FH=yes, SCC=no, Public_Transportation, SMOKE=no, CALC=Sometimes, CAEC=Sometimes, NCP=3	Obesity_Type_III	0.1521	0.9817

Scatter Plot for Obesity_Type_III Association Rules





References

Bag, H. G., Yagin, F., Gormez, Y., González, P., Colak, C., Gülü, M., Badicu, G., & Ardigò, L. (2023). Estimation of Obesity Levels through the Proposed Predictive Approach Based on Physical Activity and Nutritional Habits. *Diagnostics, 13*(18), 2949. https://doi.org/10.3390/diagnostics13182949

Rodríguez, E., Rodríguez, E., Nascimento, L., da Silva, A., & Marins, F. (2021). Machine learning Techniques to Predict Overweight or Obesity. *IDDM-2021: 4th International Conference on Informatics & Data-Driven Medicine* (Vol. 3038, pp. 190–204). CEUR Workshop Proceedings. https://ceur-ws.org/Vol-3038/paper13.pdf

Jeon, J., Lee, S., & Oh, C. (2023). Age-specific risk factors for the prediction of obesity using a machine learning approach. *Front. Public Health* 10:998782. https://doi.org/10.3389/fpubh.2022.998782

Ferdowsy, F., Rahi K.S.A., Jabiullah, M. I., & Habib, M. T. (2021). A machine learning approach for obesity risk prediction. *Current Research in Behavioral Sciences* 2:100053. https://doi.org/10.1016/j.crbeha.2021.100053.

Barzinji, A.O., Ma, C., Du, W., & Ma, J. (2021). A Machine Learning Approach to Predict the Trend of Obesity Prevalence at a Global Level. *2021 IEEE/ACIS 6th International Conference on Big Data, Cloud Computing, and Data Science (BCD)* pp. 25-30. https://doi.org/10.1109/BCD51206.2021.9581579

De-La-Hoz-Correa, E., Mendoza-Palechor, F.E., De-La-Hoz-Monatas, A., Morales-Ortega, R.C., Adriana, S.H.B. (2019). Obesity Level Estimation Software based on Decision Trees. *Journal of Computer Science* 15 (1): 67-77. https://doi.org/10.3844/jcssp.2019.67.77

Devi, K.N., Krishnamoorthy, N., Jayanthi, P., Karthi, S., Karthik, T., & Kiranbharath, K. (2022). Machine Learning Based Adult Obesity Prediction. *2022 International Conference on Computer Communication and Informatics (ICCCI)* pp. 1-5. https://doi.org/10.1109/ICCCI54379.2022.9740995