Obesity Prediction

1. Preprocessing Steps

The dataset used for obesity prediction contains features such as age, height, weight, family history, and lifestyle factors, with the target variable NObeyesdad representing obesity levels. The preprocessing steps are detailed below:

- **Missing Value Handling**:
  - The training dataset had missing values in FCVC (12 instances) and CALC (28 instances
  - FCVC is an integer , we use mean to fill nulls .
  - CALC is an object , we use mode to fill nulls.
  - The test dataset had no missing values, ensuring consistency during evaluation.
- **Encoding**:
  - Categorical variables (Gender, family_history_with_overweight, FAVC, CAEC, SMOKE, SCC, CALC, MTRANS, NObeyesdad) were encoded using LabelEncoder. This transformed categorical values into numerical representations suitable for machine learning models.
  - Encoding for column 'Gender':

    Female -> 0

    Male -> 1

    Encoding for column 'family_history_with_overweight':

    no -> 0

    yes -> 1

    Encoding for column 'FAVC':

    no -> 0

    yes -> 1

    Encoding for column 'CAEC':

    Always -> 0

Frequently -> 1

Sometimes -> 2

no -> 3

Encoding for column 'SMOKE':

no -> 0

yes -> 1

Encoding for column 'SCC':

no -> 0

yes -> 1

Encoding for column 'CALC':

Always -> 0

Frequently -> 1

Sometimes -> 2

no -> 3

Encoding for column 'MTRANS':

Automobile -> 0

Bike -> 1

Motorbike -> 2

Public_Transportation -> 3

Walking -> 4

Encoding for column 'NObeyesdad':

Insufficient_Weight -> 0

Normal_Weight -> 1

Obesity_Type_I -> 2

Obesity_Type_II -> 3

Obesity_Type_III -> 4

Overweight_Level_I -> 5

Overweight_Level_II -> 6

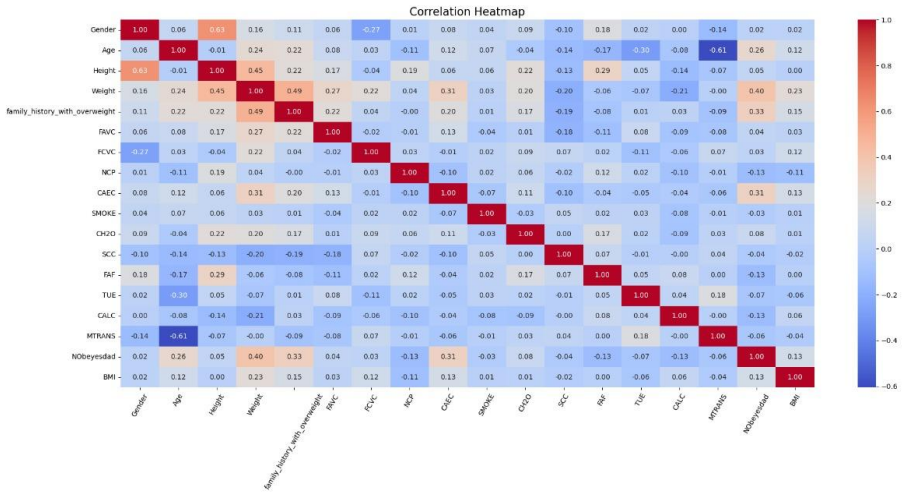
- **Scaling**:
  - Numerical features (Age, Height, Weight, FCVC, NCP, CH2O, FAF, TUE) were standardized using StandardScaler to ensure zero mean and unit variance. This step is critical for models like Logistic Regression and SVM, which are sensitive to feature scales.
  - The StandardScaler objects were saved for consistent scaling during inference.
- **Feature Engineering**:
  - A new feature, BMI, was calculated as Weight / (Height^2) to capture the relationship between weight and height, a key indicator of obesity.
  - The final feature set included all original features plus BMI, with no explicit feature selection (e.g., dropping low-importance features) mentioned. All features were retained, as indicated by final_columns.

2. Data Visualizations and Interpretation

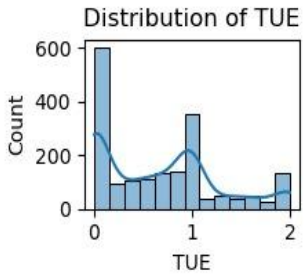
The notebook includes a correlation heatmap to visualize relationships between numerical features. Below is an interpretation of this visualization and recommendations for additional plots:

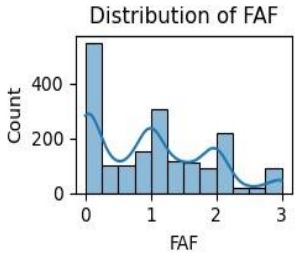
- **Correlation Heatmap**:
  - The heatmap (saved as Correlation_after_Drop.png) shows pairwise correlations between numerical features using the Pearson correlation coefficient, visualized with a coolwarm color scheme.
  - **Key Observations**:
    - **Weight and BMI**: Likely show a strong positive correlation as BMI is derived from weight and height.

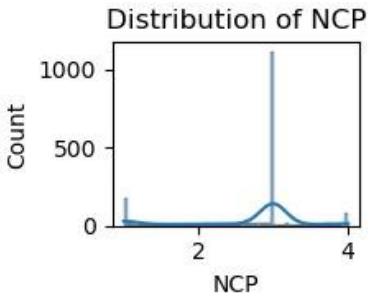
        - **Gender , FAVC , FCVC , SMOKE , SCC have Correlation Less Than** .05.
    - **Limitations**: The heatmap only includes numerical features, excluding categorical variables like family_history_with_overweight or MTRANS.

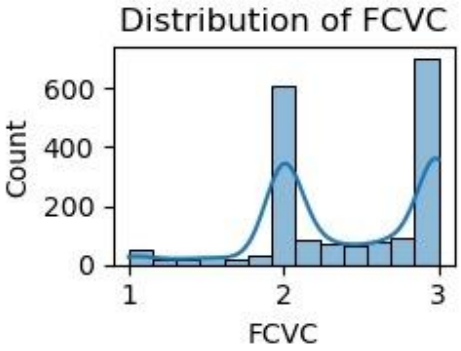


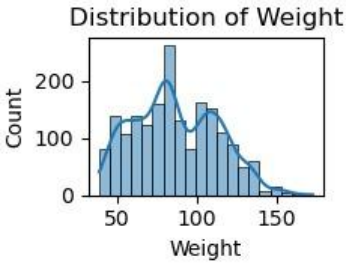
- **Recommended Additional Visualizations**:
    - **Distribution of Featrues** :

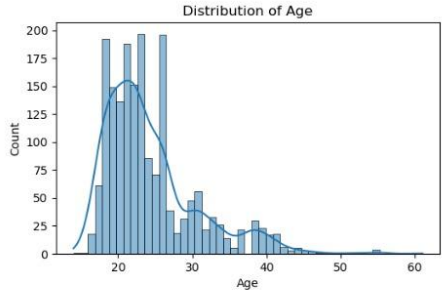


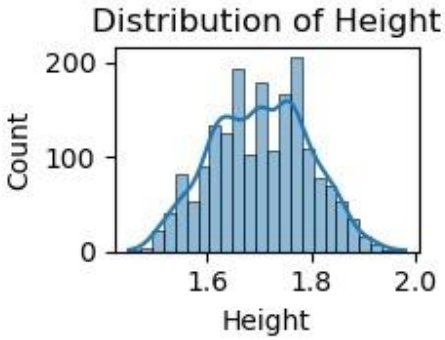

Distribution of FAF
Count
400
200
0
0
1
2
3
FAF

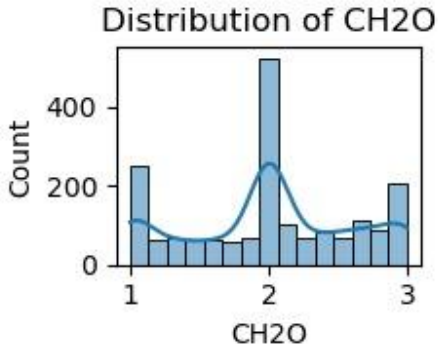

Distribution of NCP
Count
1000
500
0
2
4
NCP


Distribution of FCVC
Count
600
400
200
0
1
2
3
FCVC

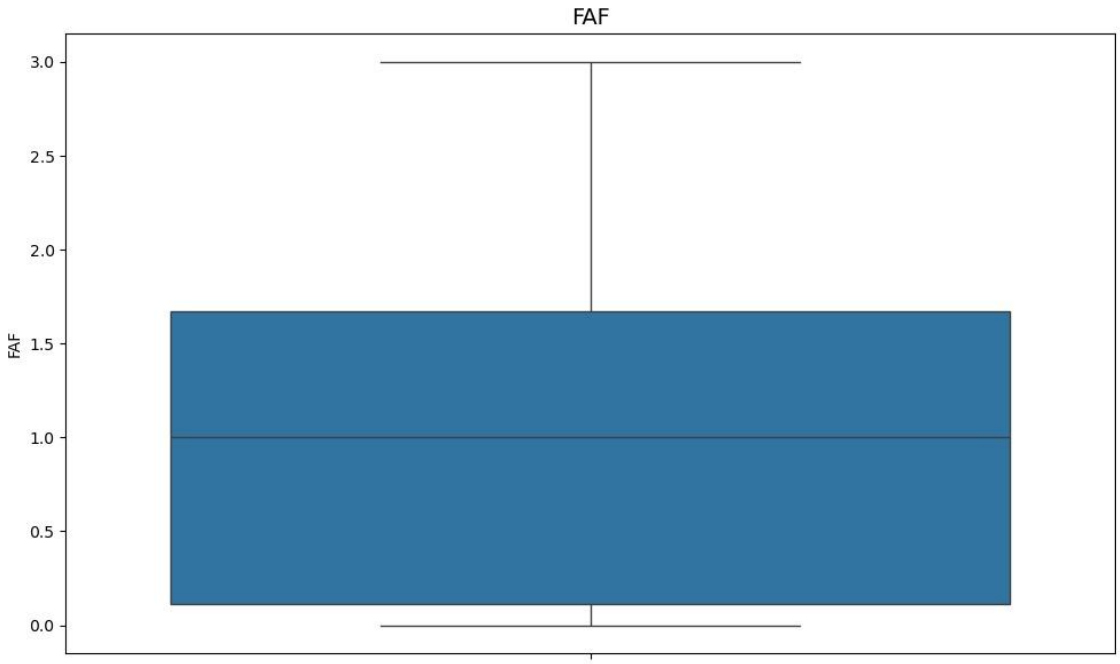

Distribution of Weight
Count
200
100
0
50
100
150
Weight

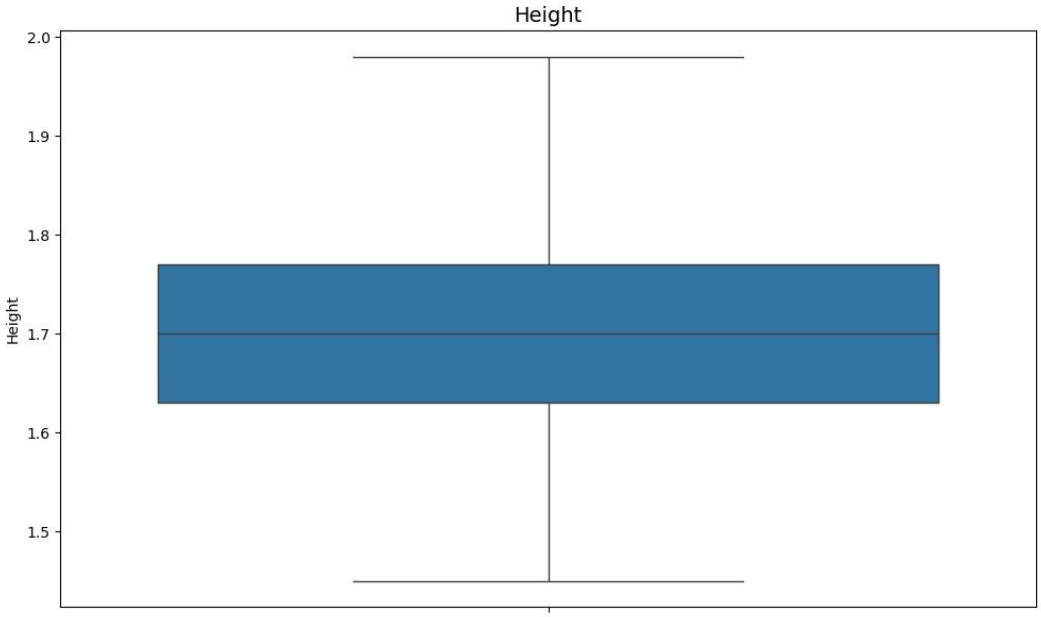

Distribution of Age
Count
Age
0
25
50
75
100
125
150
175
200
20
30
40
50
60

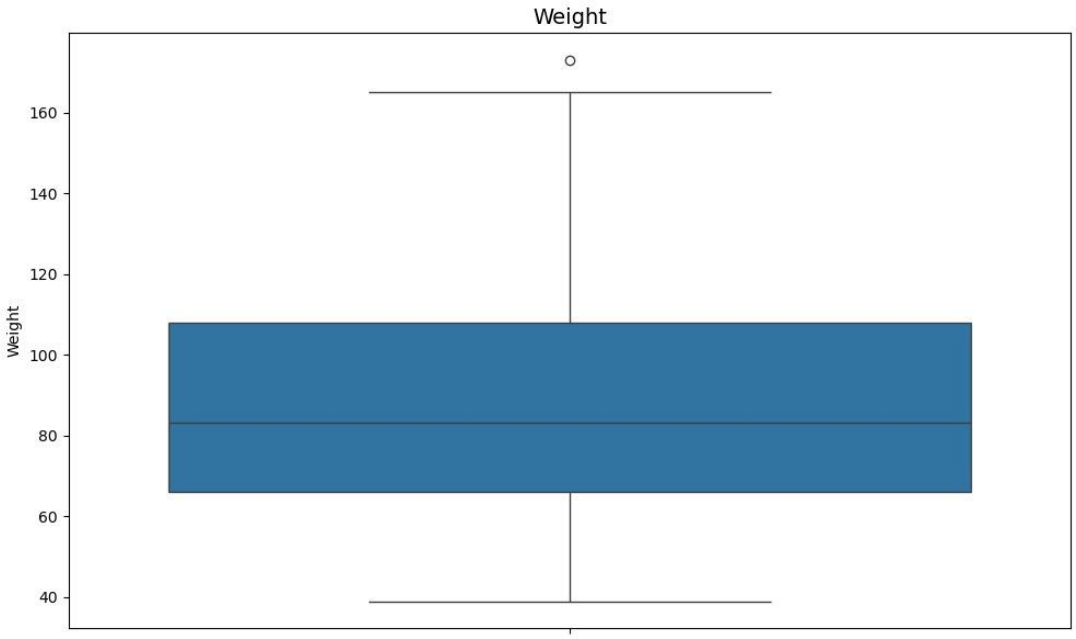

Distribution of Height
200
100
0
Count
1.6
1.8
2.0
Height

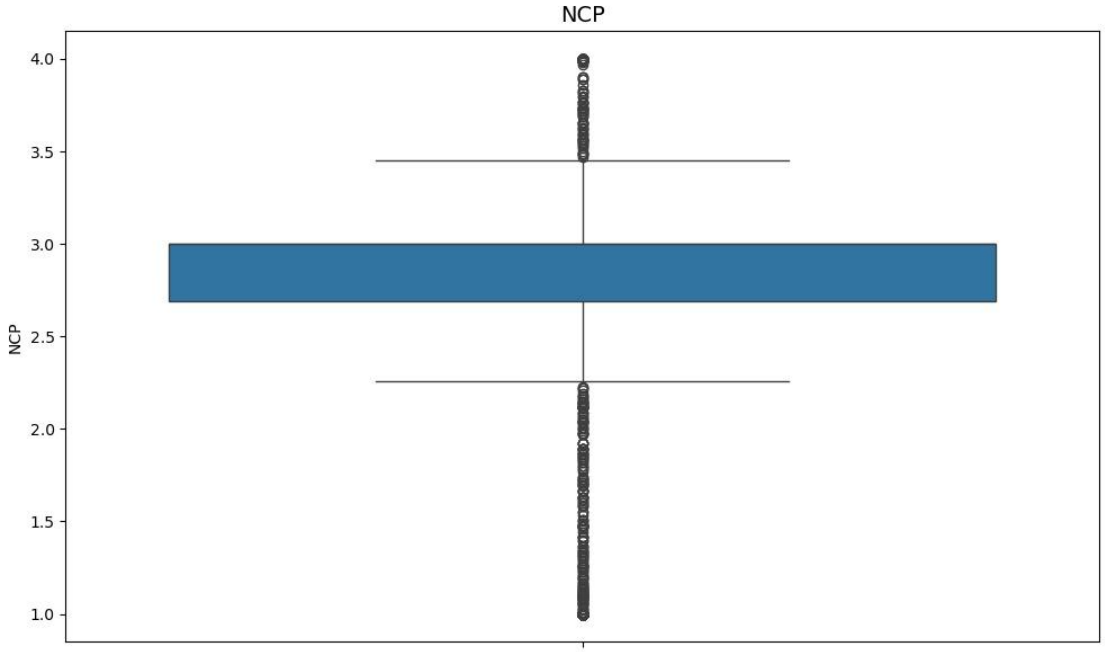


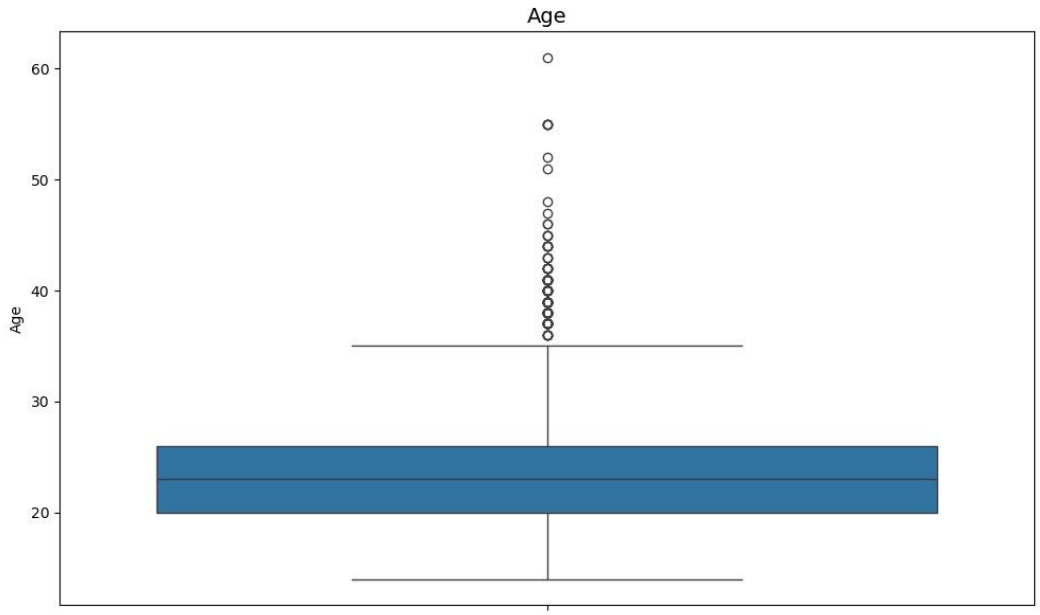
Box Plots for Numerical Features:

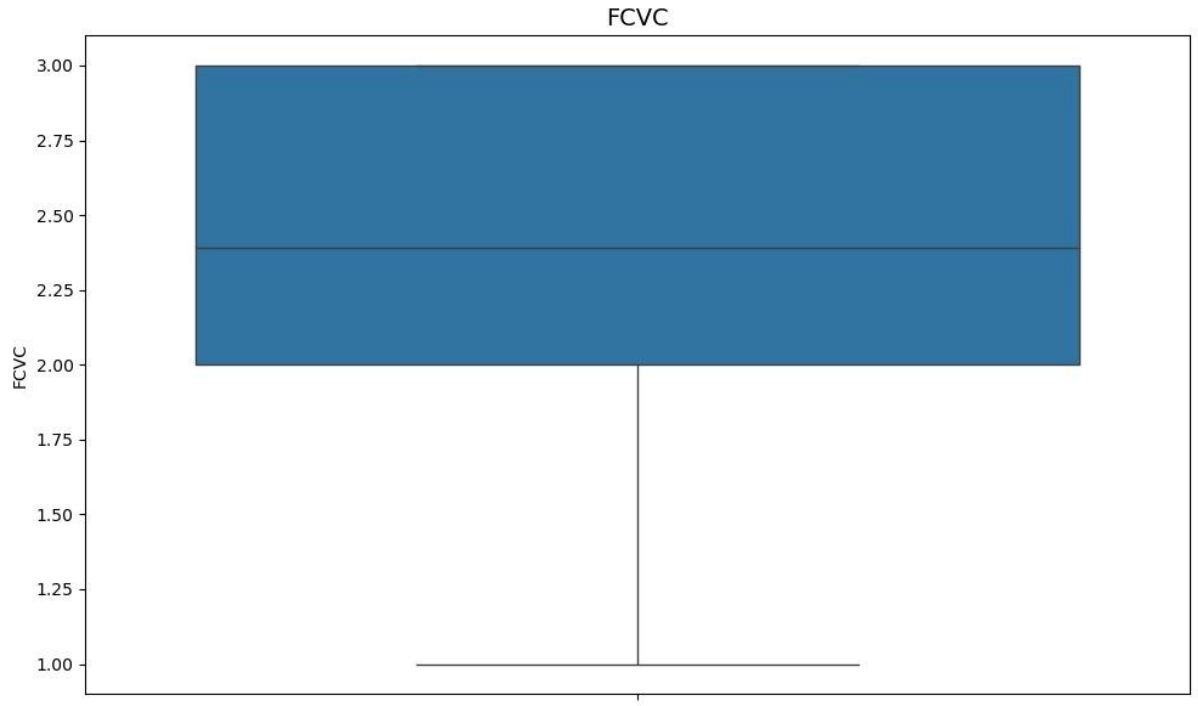
- 
FAF
3.0
2.5
2.0
1.5
1.0
0.5
0.0
FAF

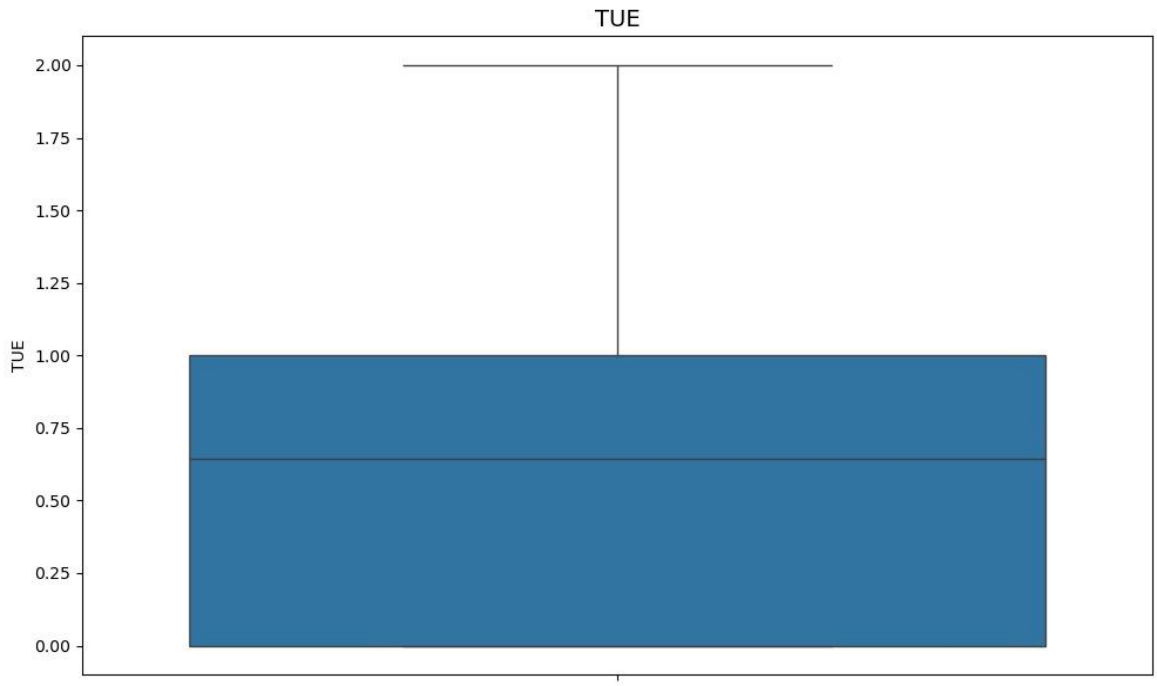

Height
2.0
1.9
1.8
1.7
1.6
1.5
Height

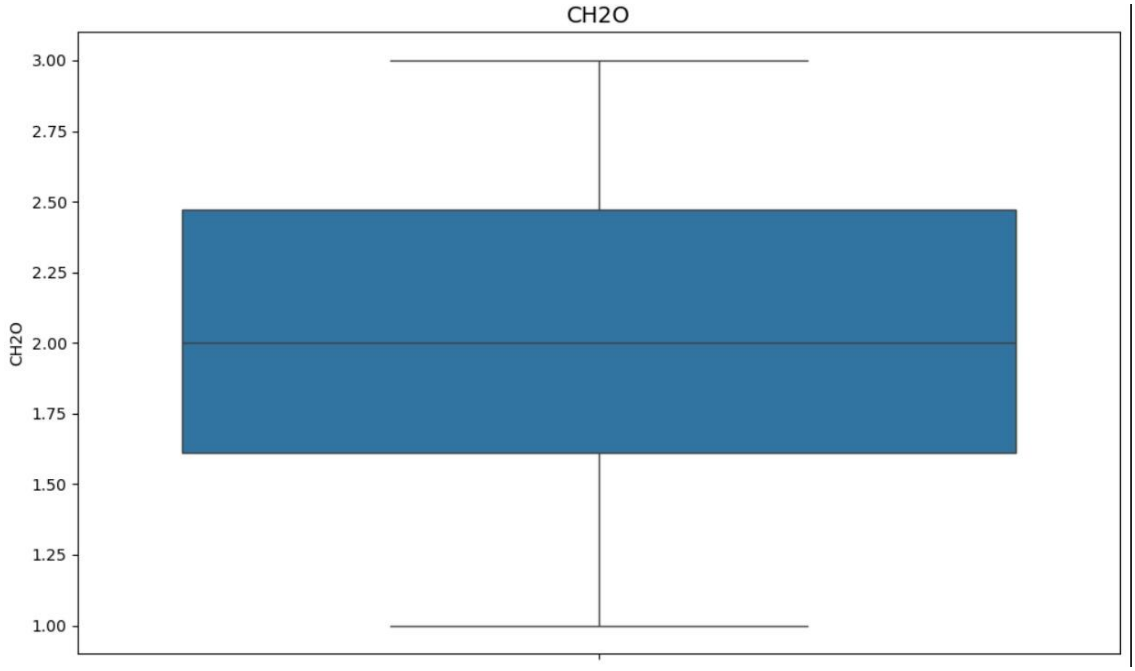

Weight
Weight
160
140
120
100
80
60
40


NCP
NCP
4.0
3.5
3.0
2.5
2.0
1.5
1.0


Age
Age
60
50
40
30
20

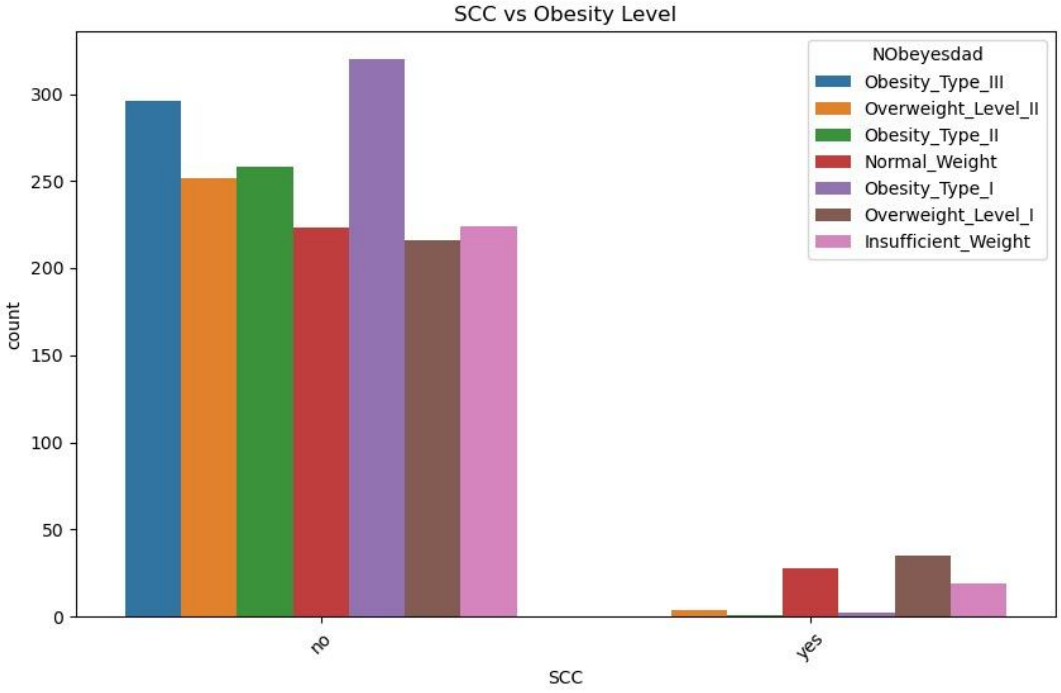

FCVC
FCVC
3.00
2.75
2.50
2.25
2.00
1.75
1.50
1.25
1.00

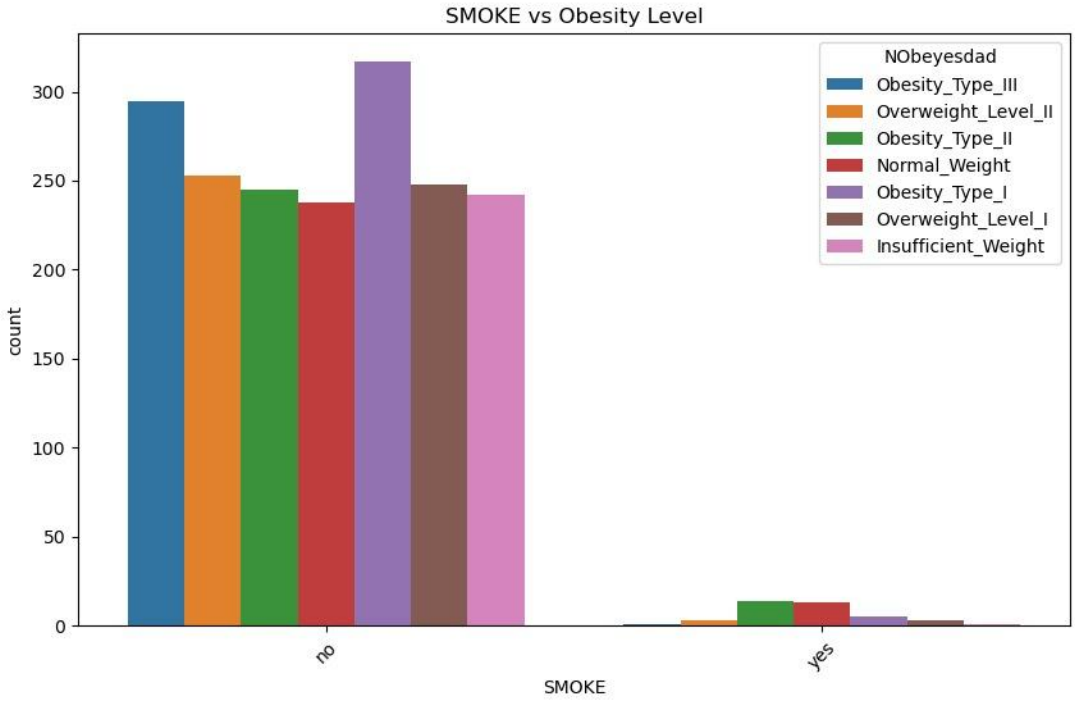


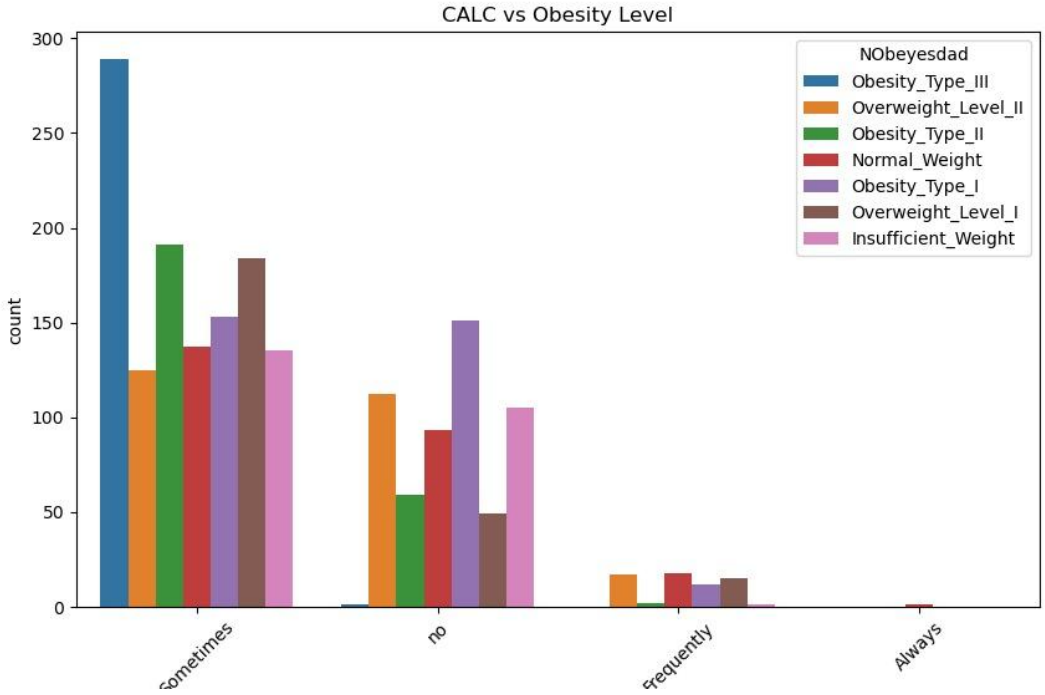


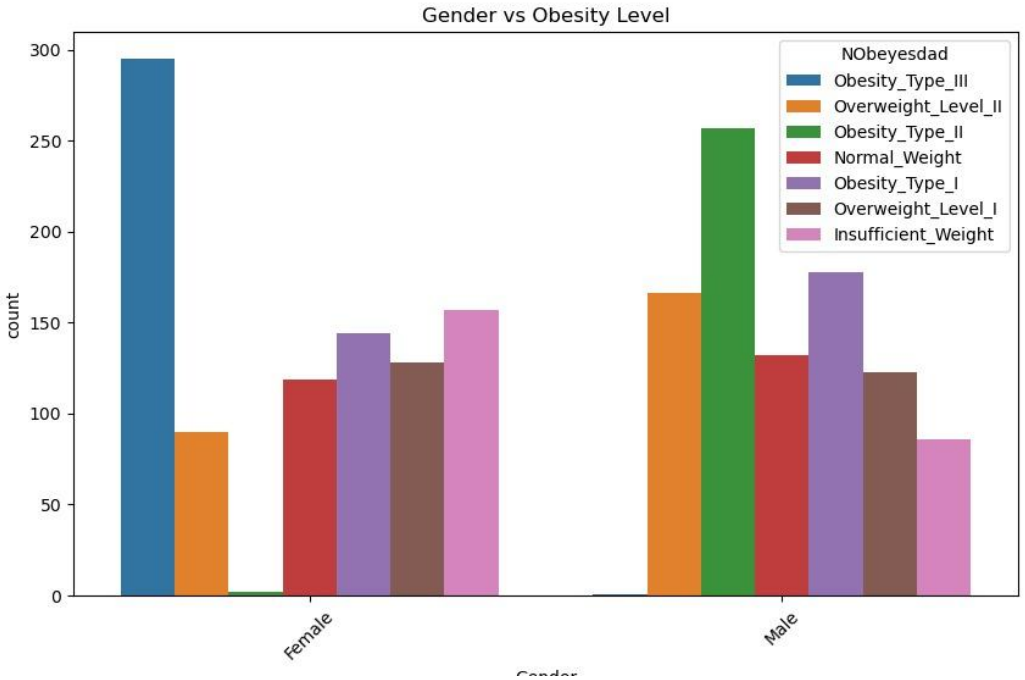
-
-

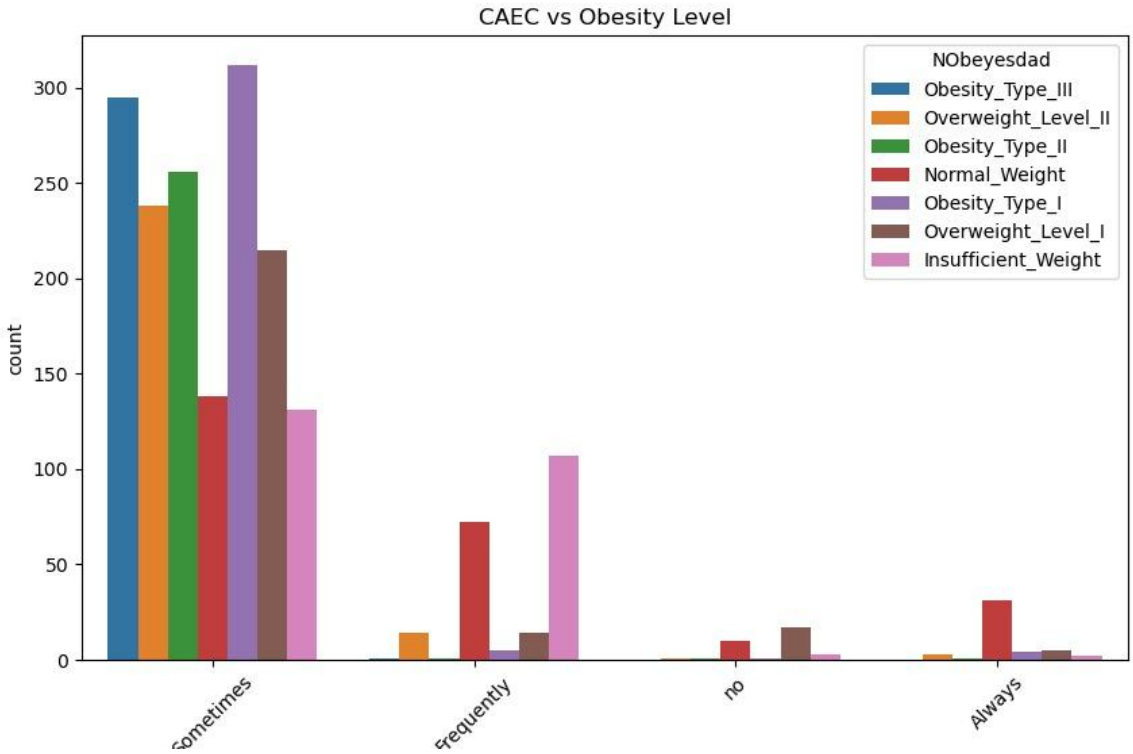
- **Features VS Obesity_Level**

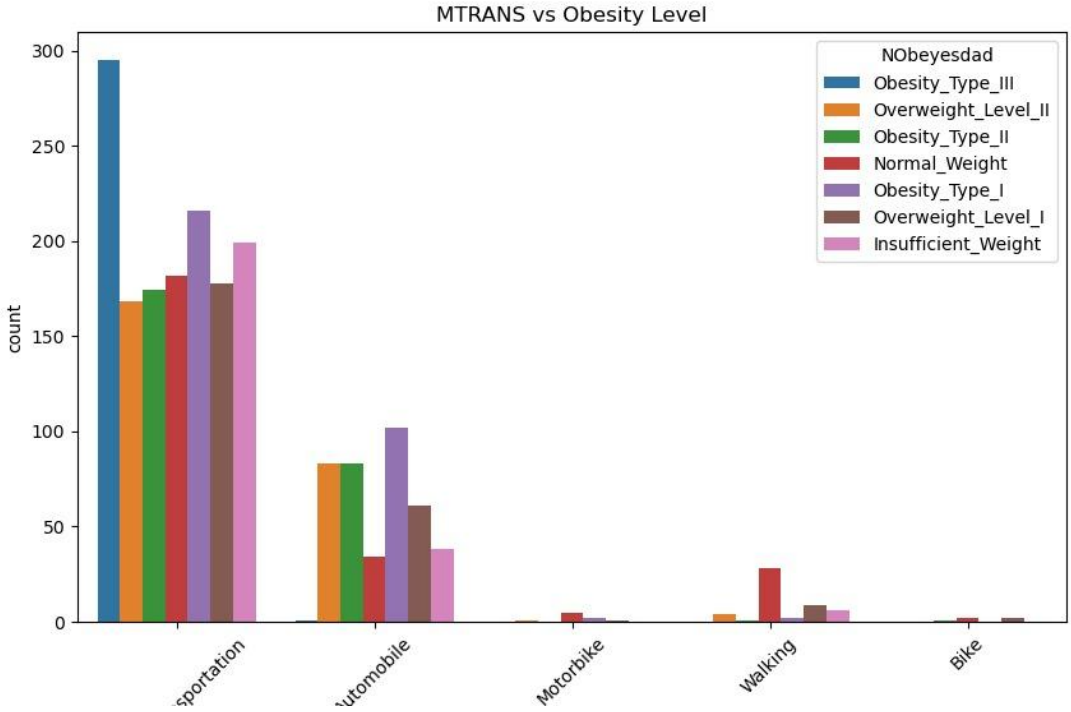

SCC vs Obesity Level
NObeyesdad
Obesity_Type_III
Overweight_Level_II
Obesity_Type_II
Normal_Weight
Obesity_Type_I
Overweight_Level_I
Insufficient_Weight
count
0
50
100
150
200
250
300
no
yes
SCC

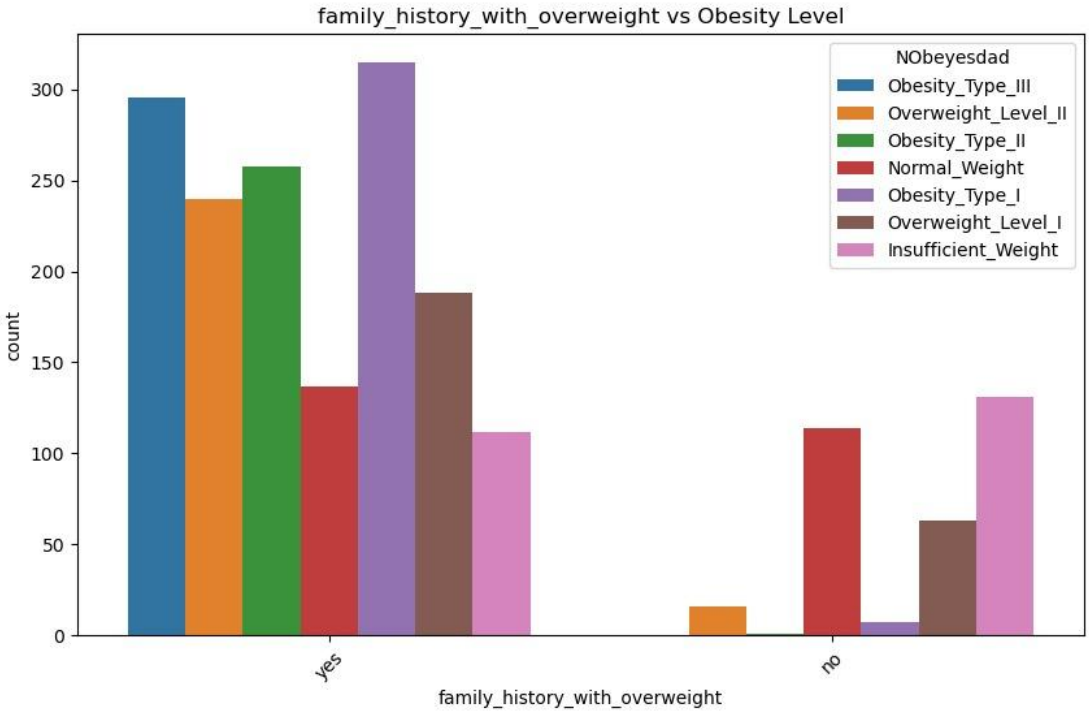

SMOKE vs Obesity Level
NObeyesdad
Obesity_Type_III
Overweight_Level_II
Obesity_Type_II
Normal_Weight
Obesity_Type_I
Overweight_Level_I
Insufficient_Weight
count
0
50
100
150
200
250
300
no
yes
SMOKE

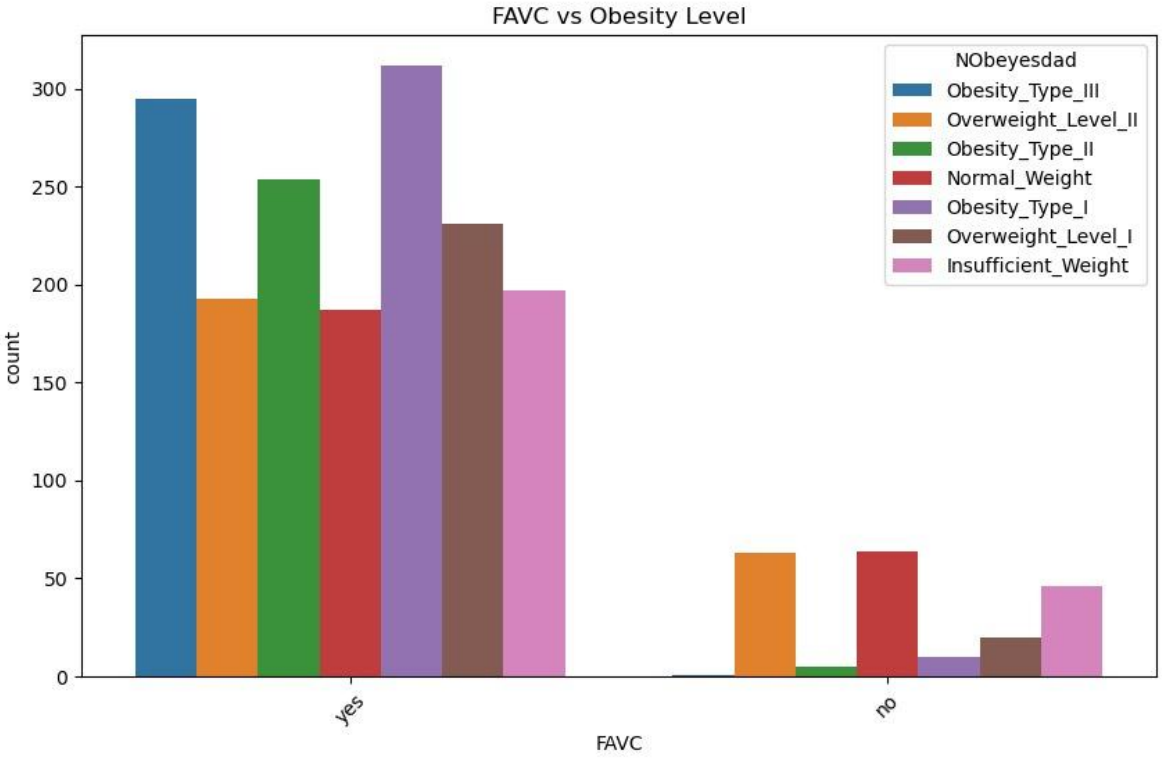

CALC vs Obesity Level
NObeyesdad
Obesity_Type_III
Overweight_Level_II
Obesity_Type_II
Normal_Weight
Obesity_Type_I
Overweight_Level_I
Insufficient_Weight
count
300
250
200
150
100
50
0
Sometimes
no
Frequently
Always


Gender vs Obesity Level
NObeyesdad
Obesity_Type_III
Overweight_Level_II
Obesity_Type_II
Normal_Weight
Obesity_Type_I
Overweight_Level_I
Insufficient_Weight
count
300
250
200
150
100
50
0
Female
Male

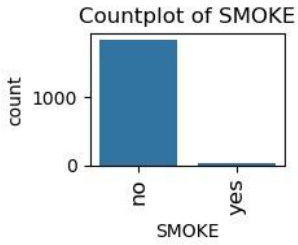

CAEC vs Obesity Level
NObeyesdad
Obesity_Type_III
Overweight_Level_II
Obesity_Type_II
Normal_Weight
Obesity_Type_I
Overweight_Level_I
Insufficient_Weight
count
0
50
100
150
200
250
300
Sometimes
Frequently
no
Always

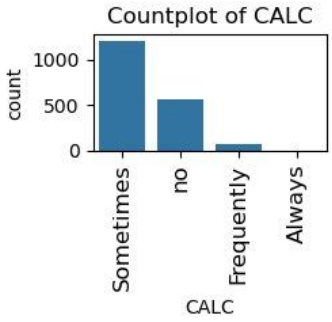

MTRANS vs Obesity Level
NObeyesdad
Obesity_Type_III
Overweight_Level_II
Obesity_Type_II
Normal_Weight
Obesity_Type_I
Overweight_Level_I
Insufficient_Weight
count
0
50
100
150
200
250
300
Automobile
Motorbike
Walking
Bike

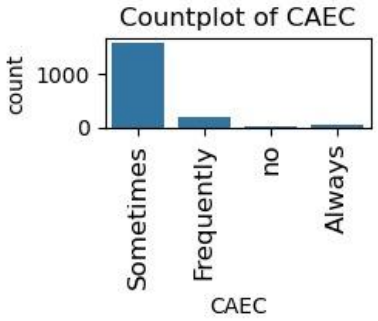


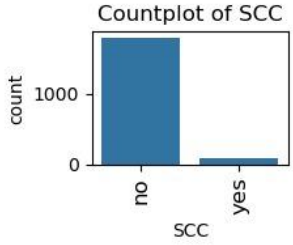


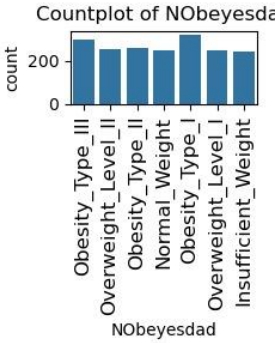
Count Plots for Categorical Features

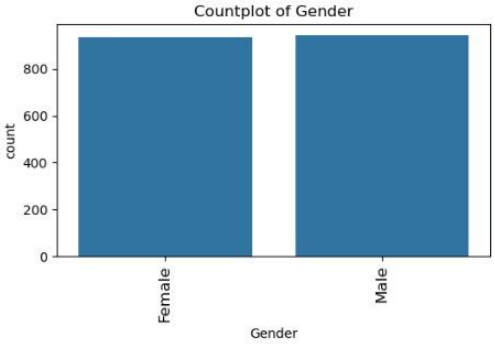

Countplot of SMOKE
count
1000
0
no
yes
SMOKE

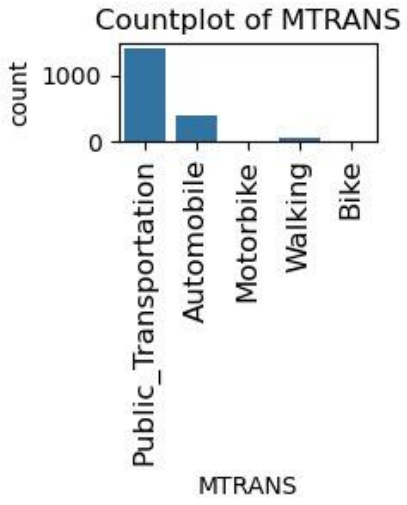

Countplot of CALC
count
1000
500
0
Sometimes
no
Frequently
Always
CALC

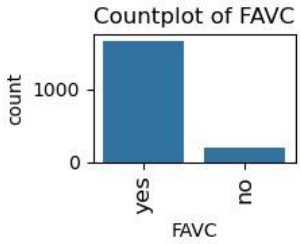

Countplot of CAEC
count
1000
0
Sometimes
Frequently
no
Always
CAEC


Countplot of SCC
count
1000
0
no
yes
SCC


Countplot of NObeyesda
count
200
0
Obesity_Type_III
Overweight_Level_II
Obesity_Type_II
Normal_Weight
Obesity_Type_I
Overweight_Level_I
Insufficient_Weight
NObeyesdad


Countplot of Gender
count
800
600
400
200
0
Female
Male
Gender





3. Model Selection, Hyperparameters, and Tuning

Four models were evaluated: Logistic Regression, SVM, Random Forest, and Gradient Boosting. Additionally, an XGBoost model was tuned with different hyperparameters.

- **Model Selection**:
  - **Logistic Regression**
  - -Simple and interpretable
  - - Good baseline mode
  - - Quick to train and easy to tune
  - **SVM (Support Vector Machine)**:
  - Powerful, especially for small to medium-sized data
  - - Requires careful tuning of kernel and regularization
  - - Not ideal for large datasets 10,000 to 50,000 row
  - **Random Forest**:
  - - Great performance with little tuning
  - - Built-in feature importance
  - - Handles missing values and categorical variables well
  - **Gradient Boosting**:
  - - Excellent performance
  - - More complex, requires parameter tuning
  - - Slower to train but highly effective
- **Hyperparameters and Tuning**:
  - **XGBoost Tuning**:
    - Four trials were conducted with different combinations of max_depth and learning_rate:
      - **Trial 1**: max_depth=3, learning_rate=0.1
        - Train Accuracy: 0.9920, Test Accuracy: 0.9621
      - **Trial 2**: max_depth=6, learning_rate=0.1
        - Train Accuracy: 1.0000, Test Accuracy: 0.9621
      - **Trial 3**: max_depth=3, learning_rate=0.01
        - Train Accuracy: 0.8557, Test Accuracy: 0.8199
      - **Trial 4**: max_depth=6, learning_rate=0.01
        - Train Accuracy: 0.9835, Test Accuracy: 0.9431
    - **Best Parameters**: max_depth=6, learning_rate=0.1 (Test Accuracy: 0.9621, Train Accuracy: 1.0000).
    - **Observations**:
      - Higher max_depth (6 vs. 3) improved performance by allowing the model to capture more complex patterns, but risked overfitting (Train Accuracy: 1.0000).
      - Lower learning_rate (0.01) reduced performance, likely due to insufficient model updates within the default number of iterations.
- **Tuning Approach**:

- Manual grid search was used for XGBoost, testing combinations of max_depth and learning_rate. A more comprehensive approach (e.g., GridSearchCV or RandomizedSearchCV) could explore additional parameters like n_estimators or subsample.
- Other models used default parameters, which may not be optimal. Future work could include hyperparameter tuning for Random Forest and SVM.

5. Model Evaluation and Comparison

The models were evaluated using accuracy, precision, recall, F1-score, confusion matrix, and 5-fold cross-validation accuracy. Below is a summary:

Model	Train Accuracy	Test Accuracy	Precision	Recall	F1-Score	CV Accuracy
Logistic Regression	0.8876	0.8815	0.8801	0.8869	0.8799	0.8610
SVM	0.9382	0.9194	0.9192	0.9232	0.9195	0.9026
Random Forest	1.0000	0.9763	0.9786	0.9770	0.9772	0.9675
Gradient Boosting	1.0000	0.9810	0.9796	0.9829	0.9807	0.9665

Logistic Regression:

- **Strengths**: Simple, interpretable, and computationally efficient. Performs well for linearly separable data.
- **Weaknesses**: Lowest performance (Test Accuracy: 0.8815) due to its inability to capture complex non-linear relationships. The confusion matrix shows misclassifications, particularly for Obesity_Type_III and Overweight_Level_II.
- **Use Case**: Suitable as a baseline or when interpretability is prioritized.
-

```
--- Logistic Regression ---
Saved model to models\logistic_regression.pkl
Train Accuracy: 0.8876
Test Accuracy: 0.8815
There is not overfitting :)
Precision (macro avg): 0.8801
Recall (macro avg): 0.8869
F1-score (macro avg): 0.8799
Confusion Matrix:
[[24  0  0  0  0  0  0]
 [ 5 26  0  0  0  1  0]
 [ 0  0 26  3  0  0  0]
 [ 0  0  1 36  1  0  0]
 [ 0  0  0  0 28  0  0]
 [ 0  1  0  0  0 22  3]
 [ 0  1  5  0  0  4 24]]
Cross-Validation Accuracy (5-Fold): 0.8610
================================================
```

- **SVM**:
  - **Strengths**: Effective for non-linear data with the RBF kernel. Better performance than Logistic Regression (Test Accuracy: 0.9194).
  - **Weaknesses**: Computationally expensive and sensitive to hyperparameter tuning (C, gamma). Misclassifications occur in Obesity_Type_III and Overweight_Level_II, as shown in the confusion matrix.
  - **Use Case**: Suitable for small to medium datasets with complex relationships.

```
--- SVM ---
Saved model to models\svm.pkl
Train Accuracy: 0.9382
Test Accuracy: 0.9194
There is not overfitting :)
Precision (macro avg): 0.9192
Recall (macro avg): 0.9232
F1-score (macro avg): 0.9195
Confusion Matrix:
[[24  0  0  0  0  0  0]
 [ 1 30  0  0  0  1  0]
 [ 0  0 27  2  0  0  0]
 [ 0  0  1 37  0  0  0]
 [ 0  0  0  0 28  0  0]
 [ 0  0  0  0  0 23  3]
 [ 0  1  4  0  0  4 25]]
Cross-Validation Accuracy (5-Fold): 0.9026
================================================
```

- **Random Forest**:
  - **Strengths**: High performance (Test Accuracy: 0.9763), robust to overfitting, and provides feature importance. The confusion matrix shows minimal errors, with strong performance across all classes.
  - **Weaknesses**: Perfect training accuracy (1.0000) suggests potential overfitting, though test accuracy remains high. Computationally intensive for large datasets.
  - **Use Case**: Ideal for structured data with minimal preprocessing requirements.

```
there is not overfitting :)
Precision (macro avg): 0.9786
Recall (macro avg): 0.9770
F1-score (macro avg): 0.9772
Confusion Matrix:
[[24  0  0  0  0  0  0]
 [ 0 32  0  0  0  0  0]
 [ 0  0 28  1  0  0  0]
 [ 0  0  0 38  0  0  0]
 [ 0  0  0  0 28  0  0]
 [ 0  1  0  0  0 25  0]
 [ 0  2  0  0  0  1 31]]
Cross-Validation Accuracy (5-Fold): 0.9675
==================================================
```

- **Gradient Boosting**:
  - **Strengths**: Best performer (Test Accuracy: 0.9810), with excellent precision, recall, and F1-score. The confusion matrix shows near-perfect classification, with minor errors in Overweight_Level_II.
  - **Weaknesses**: Perfect training accuracy (1.0000) indicates potential overfitting, though cross-validation (0.9665) confirms generalization. Slower to train than Random Forest.
  - **Use Case**: Preferred for high-accuracy predictions on structured data.

```
there is not overfitting :)
Precision (macro avg): 0.9786
Recall (macro avg): 0.9770
F1-score (macro avg): 0.9772
Confusion Matrix:
[[24  0  0  0  0  0  0]
 [ 0 32  0  0  0  0  0]
 [ 0  0 28  1  0  0  0]
 [ 0  0  0 38  0  0  0]
 [ 0  0  0  0 28  0  0]
 [ 0  1  0  0  0 25  0]
 [ 0  2  0  0  0  1 31]]
Cross-Validation Accuracy (5-Fold): 0.9675
==================================================
```

.

Comparison:

- Gradient Boosting outperforms all models in test accuracy and balanced metrics, making it the best choice for deployment.
- Random Forest is a close second, with slightly lower accuracy but faster training.

- SVM and Logistic Regression are less suitable due to lower accuracy and limitations in handling complex relationships.
- Cross-validation scores confirm that ensemble methods (Random Forest, Gradient Boosting) generalize well, with CV accuracies above 0.96.

6. Summary and Conclusion

This project developed a machine learning pipeline to predict obesity levels based on demographic and lifestyle factors. Key steps included:

- **Data Preprocessing**: Handled missing values, encoded categorical variables, scaled numerical features, and engineered BMI as a derived feature.
- **Exploratory Analysis**: A correlation heatmap revealed relationships between numerical features, with recommendations for additional visualizations to explore categorical features and class distributions.
- **Lifestyle Factors**: High-calorie food consumption, low physical activity, and frequent snacking were identified as key contributors to obesity, while vegetable consumption and water intake may reduce risk.
- **Model Development**: Four models (Logistic Regression, SVM, Random Forest, Gradient Boosting) were trained, with Gradient Boosting achieving the highest test accuracy (0.9810). XGBoost was tuned, with max_depth=6 and learning_rate=0.1 yielding a test accuracy of 0.9621.
- **Evaluation**: Models were compared using accuracy, precision, recall, F1-score, and confusion matrices, with Gradient Boosting excelling in all metrics.

Conclusion: The Gradient Boosting model is recommended for predicting obesity levels due to its superior performance and robustness. The analysis highlights the importance of lifestyle factors in obesity risk, providing actionable insights for health interventions. Future work could include:

- Comprehensive hyperparameter tuning for all models using GridSearchCV.
- Feature selection to reduce model complexity.
- Additional visualizations to enhance interpretability.
- Deployment of the model in a real-world application, such as a health app.