CMPT 459 - Report

Data Selection

- The Estimation of Obesity Levels Based on Eating Habits and Physical Condition dataset was selected for the following reasons:
- Relevance: Obesity is a significant global health issue. This dataset provides an opportunity to address a real-world problem with potential impact on public health.
- Data Quality: The dataset contains no missing values, simplifying preprocessing and ensuring data completeness.
- Diverse Features: It includes a mix of categorical, binary, and continuous variables, allowing for rich analysis and feature engineering opportunities.
- Multiple Tasks: The dataset supports various machine learning tasks including classification, regression, and clustering, providing flexibility in approach.
- Size and Complexity: With 2111 instances and 16 features, the dataset is substantial enough to train robust models while remaining manageable for computational resources.
- Synthetic Data: 77% of the data is synthetically generated, which can help in addressing potential privacy concerns while still maintaining realistic patterns.
- Multi-country Data:The inclusion of data from three countries (Mexico, Peru, Colombia) allows for potential cross-cultural analysis of obesity factors.
- Clear Target Variable: The NObesity variable provides clear classification labels, making it suitable for supervised learning tasks.

Main Goal:

Data Preprocessing

1. **Dataset Loading**
    - Loaded the data using Pandas
    - Configured to display all columns for easier inspection
2. **Duplicate Removal and Null value**
    - Identified and removed duplicate rows to ensure data integrity
        i. Data = data.drop_duplicates()
3. **Handling Missing Values:**
    - Checked for missing Values:
        i. Missing values were evaluated by calculating the total number of null entries in each column using `data.isnull().sum()`. The analysis confirmed that there are no missing values in the dataset, as the sum was `0` for all columns.
4. **Normalization and standardization**
    - After analyzing the dataset, normalization and standardization techniques (e.g., Min-Max Scaling and StandardScaler) were considered for feature scaling. However, upon inspection, it was determined that the dataset's numerical

features are already scaled to an appropriate range, eliminating the need for additional scaling transformations.

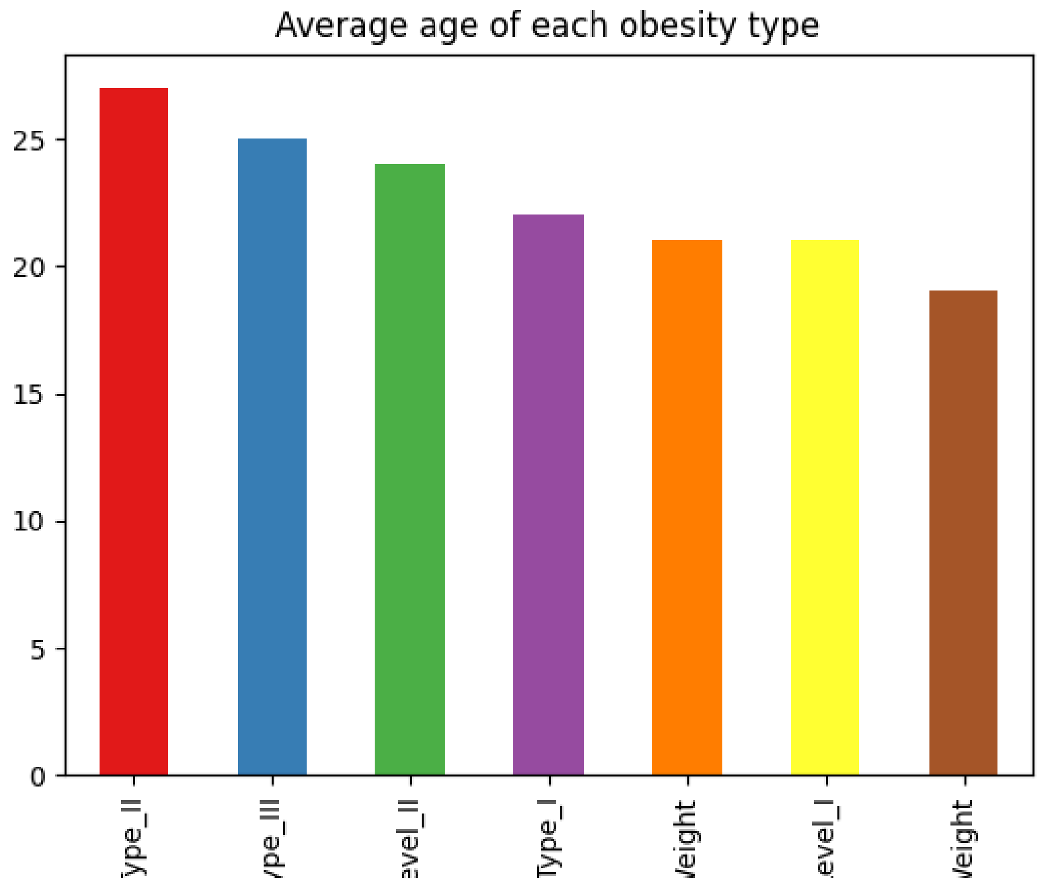
EDA

Facts

Average age of each obesity type

```
Insufficient_Weight    19.0
Normal_Weight          21.0
Obesity_Type_I         22.0
Obesity_Type_II        27.0
Obesity_Type_III       25.0
Overweight_Level_I     21.0
Overweight_Level_II    24.0
Name: Age, dtype: float64
```

Figure 1. The average age of each obesity type

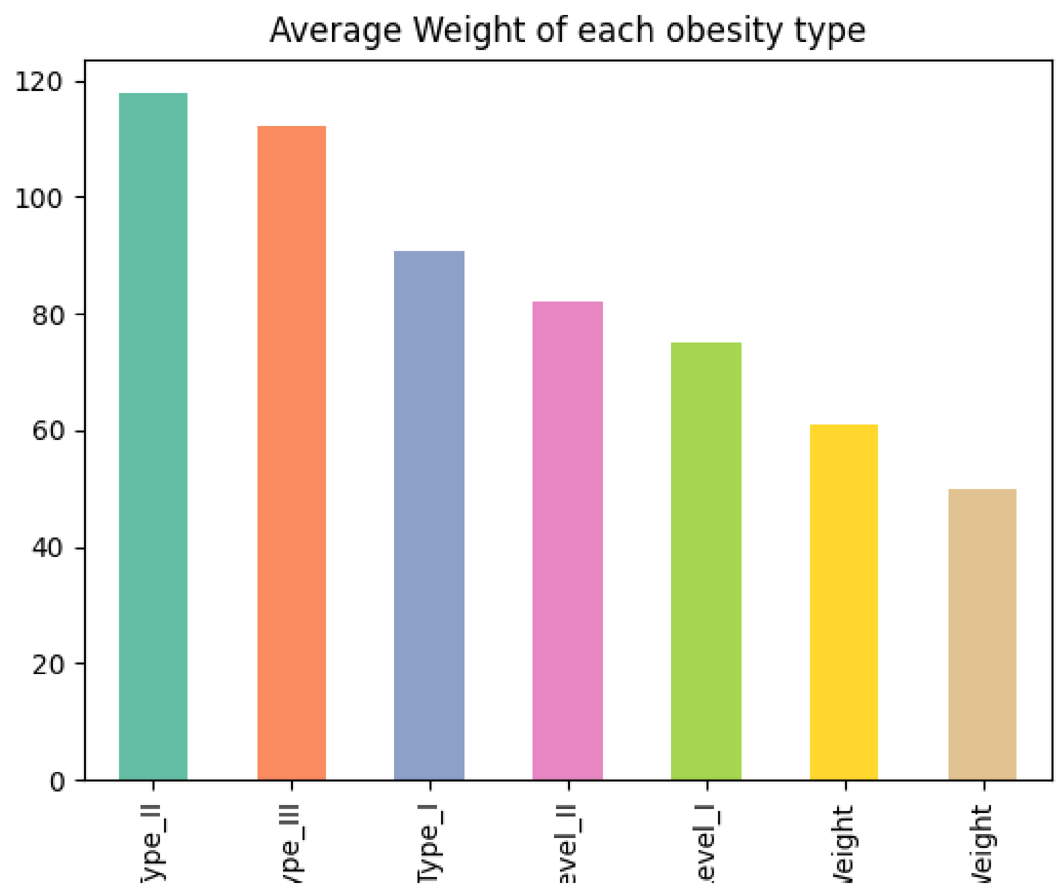


Notice that the average age is lowest in insufficient weight and is highest in **obesity type II** followed by **type III and I**. Concluding age has a positive correlation with weight

Average Weight of each obesity type

```
Insufficient_Weight     50.00
Normal_Weight           61.00
Obesity_Type_I          90.74
Obesity_Type_II        117.79
Obesity_Type_III       112.05
Overweight_Level_I      75.00
Overweight_Level_II     82.00
Name: Weight, dtype: float64
```

Figure 2. Average Weight of each obesity type



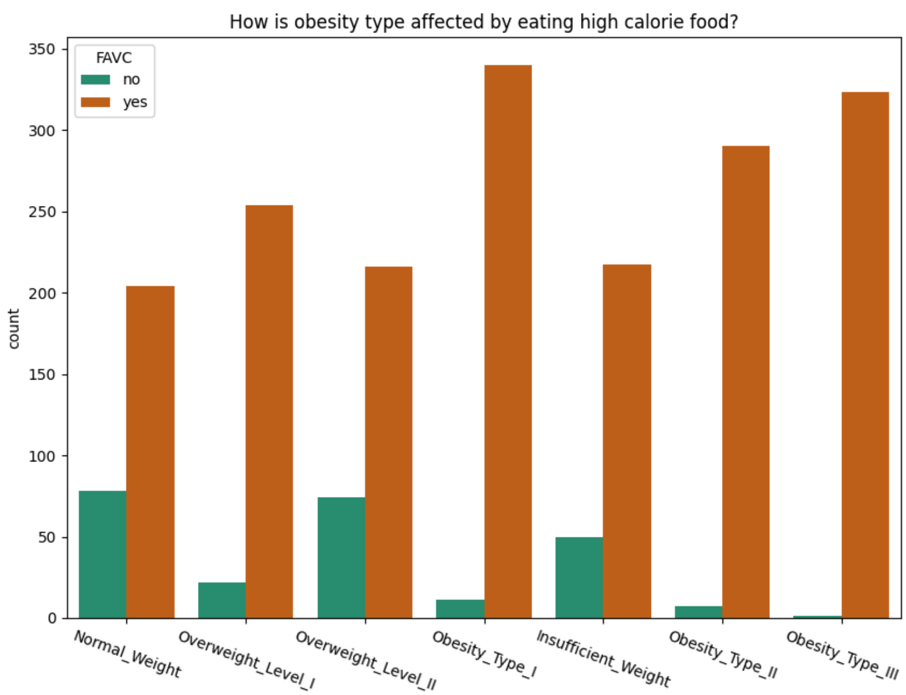
The way we have preprocessed the data, **obesity type II** has higher weight numerics than **type III**.

How is obesity type affected by eating high calorie food?

```
Insufficient_Weight  no      50
                     yes    217
Normal_Weight        no      78
                     yes    204
Obesity_Type_I       no      11
                     yes    340
Obesity_Type_II      no       7
                     yes    290
```

```
Obesity_Type_III      no        1
                      yes     323
Overweight_Level_I    no       22
                      yes     254
Overweight_Level_II   no       74
                      yes     216
Name: FAVC, dtype: int64
```

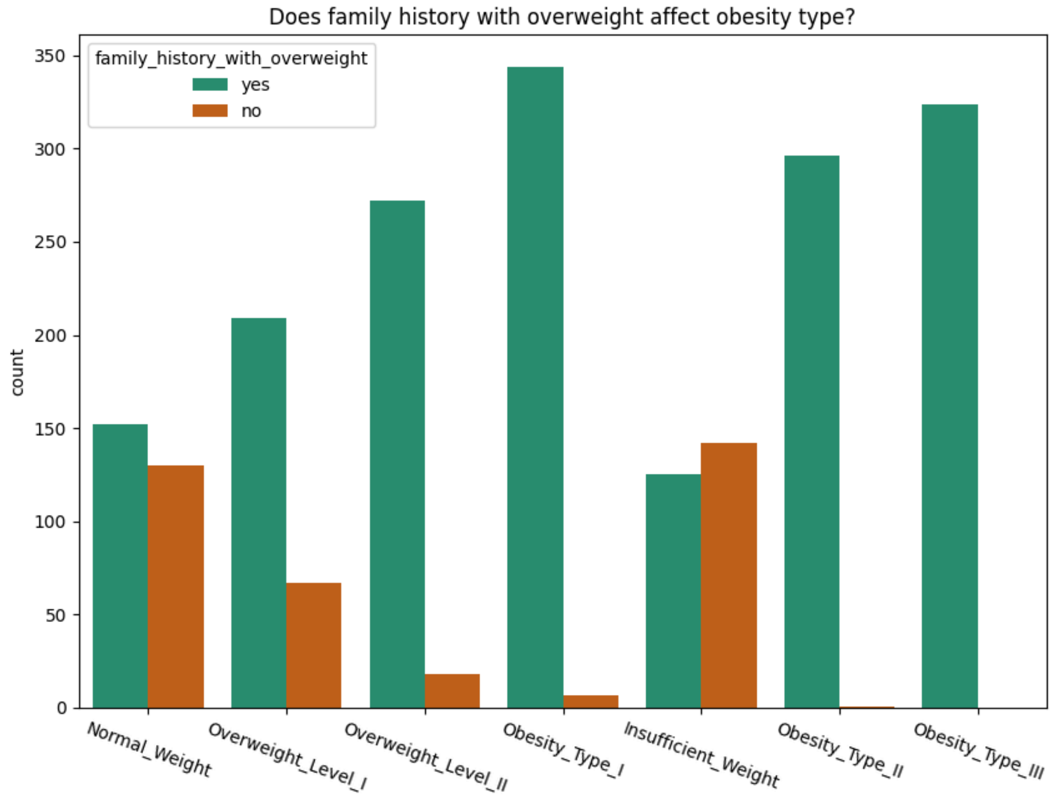
Figure 3. How is obesity type affected by eating high-calorie food?



What is interesting to note here is **obesity type III** seems to have no one eating low-calorie food.

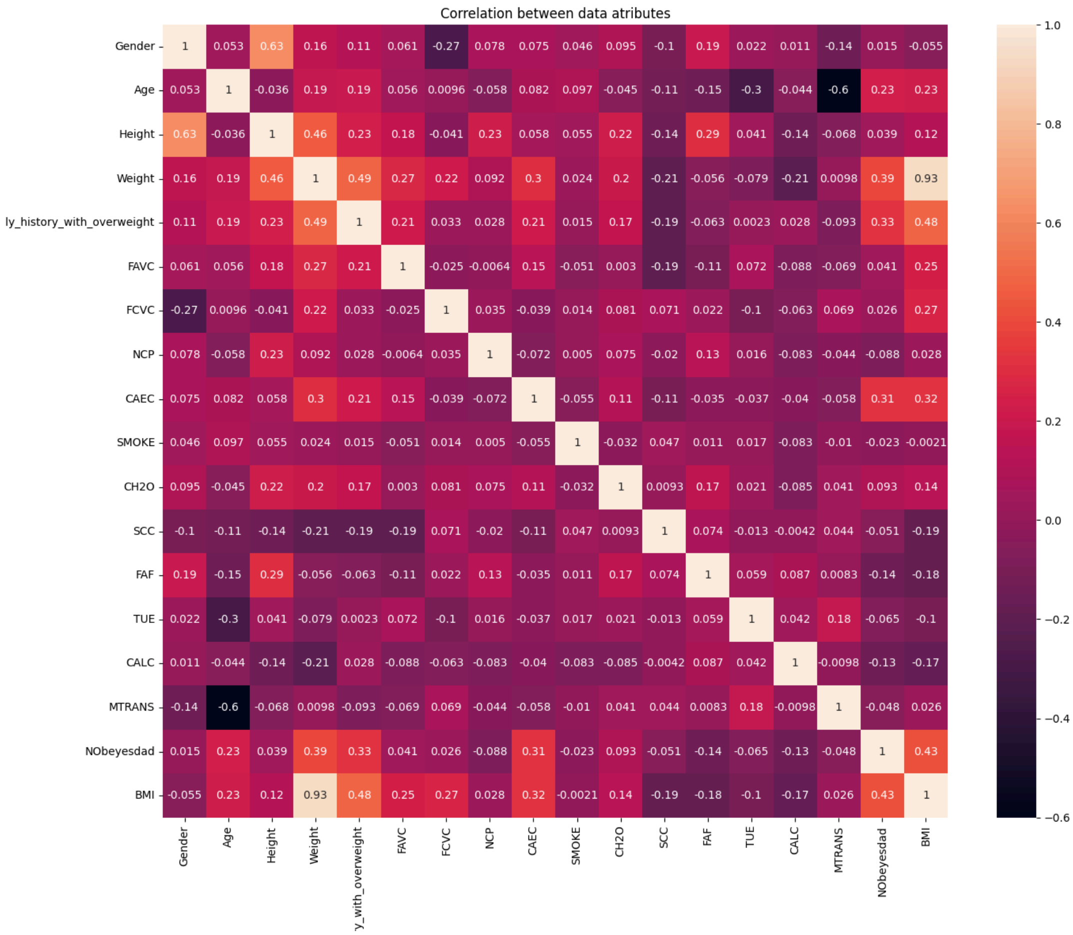
Does family history with overweight affect obesity type?

Figure 4. Family history with obesity



Having issues with weights run in your family seems to have a positive effect on increasing weights in **obesity type I,II,III**

Correlation between attributes



Addition of new feature BMI (Body Mass Index)

```
data['BMI'] = round(data['Weight'] / (data['Height']) ** 2, 2)
```

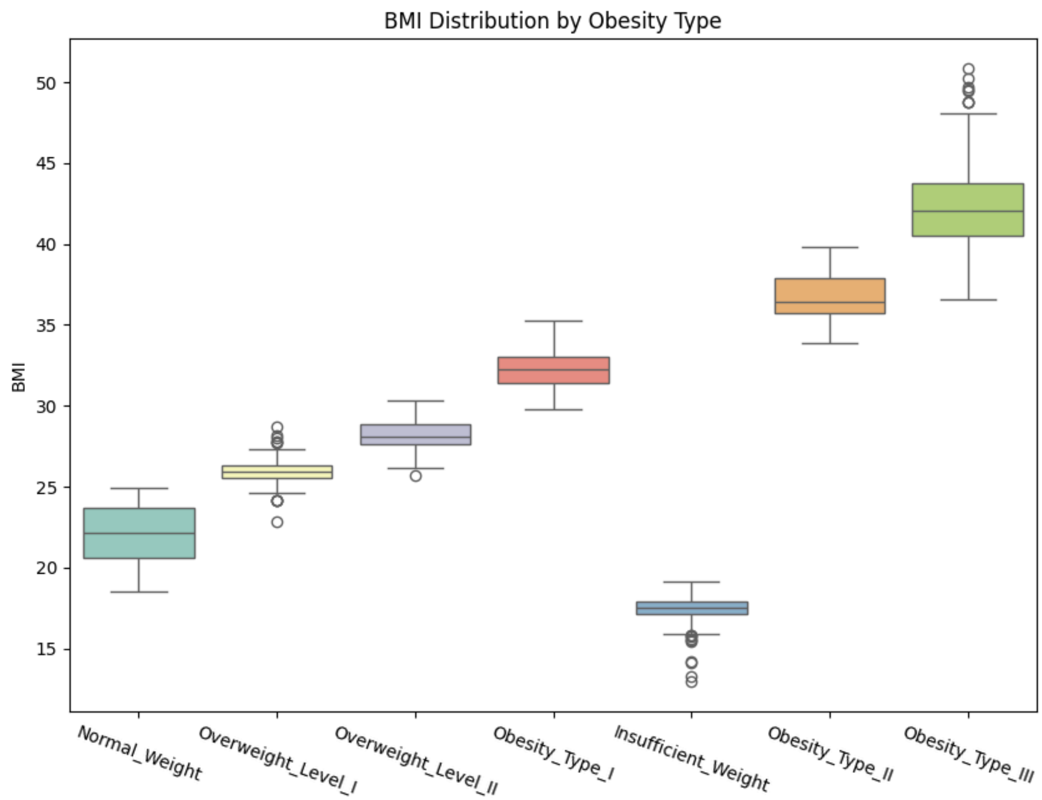
Advantages:

1. Direct relation with obesity:

   BMI is a direct indicator of weight status (underweight, normal, overweight, or obese).

2. Feature for Model Training:

BMI could act as a strong predictor in machine learning models for classifying obesity types since it is derived from weight and height.



Insights about the results obtained:

BMI Gradually Increases Across Obesity Types:

- The BMI values are progressively higher as you move from **Normal Weight** to **Overweight Levels I & II**, and then to **Obesity Types I, II, and III**.
- This shows that BMI is strongly indicative of the assigned obesity types, as expected.

Clear Separation of Obesity Categories:

- The categories are well-separated by BMI, suggesting that BMI is an effective feature for distinguishing between obesity levels in the dataset.
- For instance:
    - **Normal Weight** has a median BMI close to 22.
    - **Obesity Type III** has the highest median BMI, near 45, with a broader range.
- **Insufficient Weight** lies at the lowest BMI end (around 16-18).

Outliers:

- Some outliers are visible in categories like **Overweight Levels I & II** and **Insufficient Weight**, which may represent rare cases that deviate from typical BMI ranges.

Consistency of BMI Within Groups:

- The spread (interquartile range) of BMI within each category is relatively small, indicating consistency within each group.
- **Obesity Type III**, however, has the widest range, suggesting that individuals in this category have a broader variation in BMI

Clustering

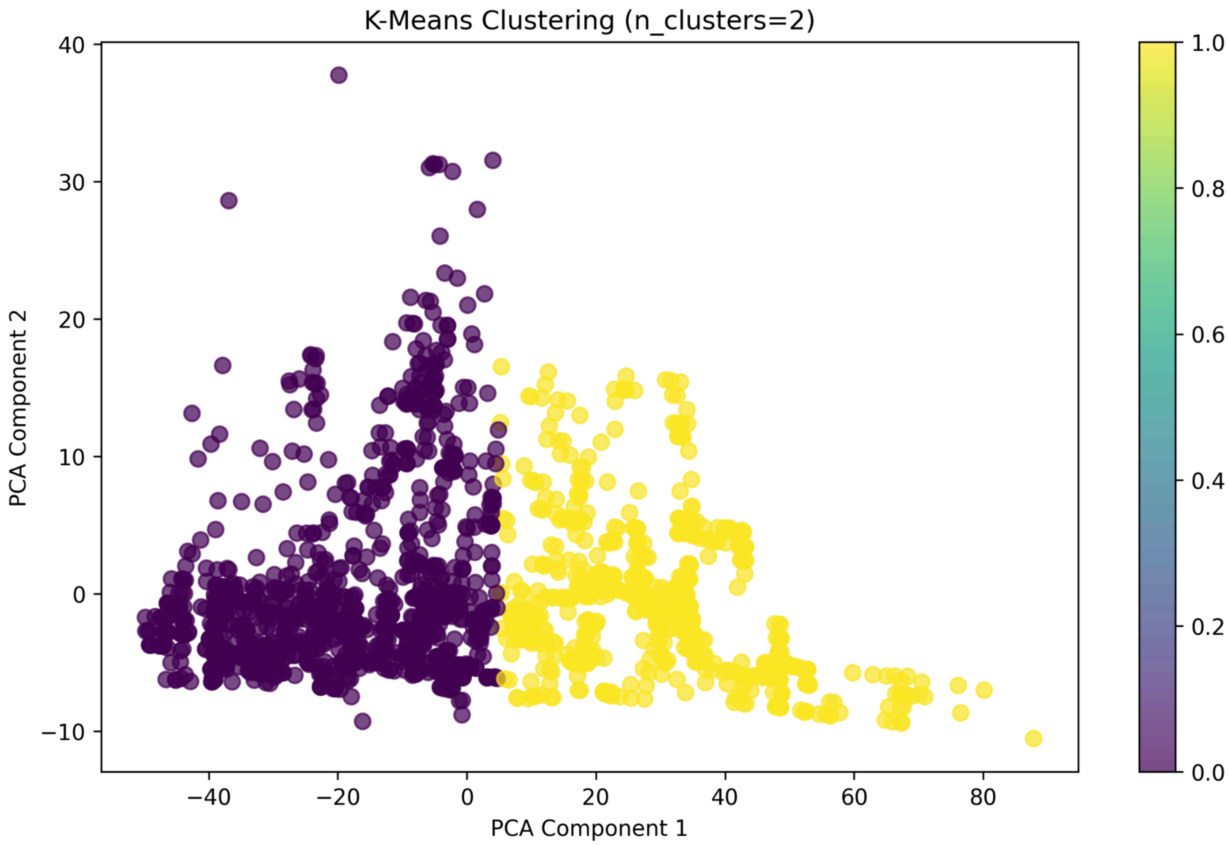
K-Means

- **We begin clustering with a low value of k = 2 and try to tune it for bigger cluster sizes. In our case, we do a range of k = 2 to 9. Obtaining results:**

```
Normalized Scores:
Silhouette:  [1.          0.6463781   0.55012837  0.39690318  0.31398566
0.25680637
 0.16581162 0.        ]
Calinski-Harabasz: [0.48318981 1.         0.97880425 0.47626443 0.72975219
0.
 0.44278261 0.2878056 ]
Inverted Davies-Bouldin:  [1.          0.68748531  0.72277838  0.26132563
0.41331726 0.17644861
 0.08984546 0.        ]
Average  Scores:  [0.82772994  0.77795447  0.75057033  0.37816441  0.48568504
0.14441833
 0.23281323 0.0959352 ]
Best number of clusters: 2
   Number of Clusters  Silhouette Score  Calinski-Harabasz Index  \
0                   2          0.561510              4233.633560
1                   3          0.499831              4623.288144
2                   4          0.483043              4607.307384
3                   5          0.456317              4228.412099
4                   6          0.441854              4419.531910
5                   7          0.431881              3869.327427
6                   8          0.416010              4203.168122
7                   9          0.387089              4086.321542

   Davies-Bouldin Index
0              0.597637
```

```
1               0.671880
2               0.662584
3               0.808910
4               0.754060
5               0.843160
6               0.881230
7               0.924537
```

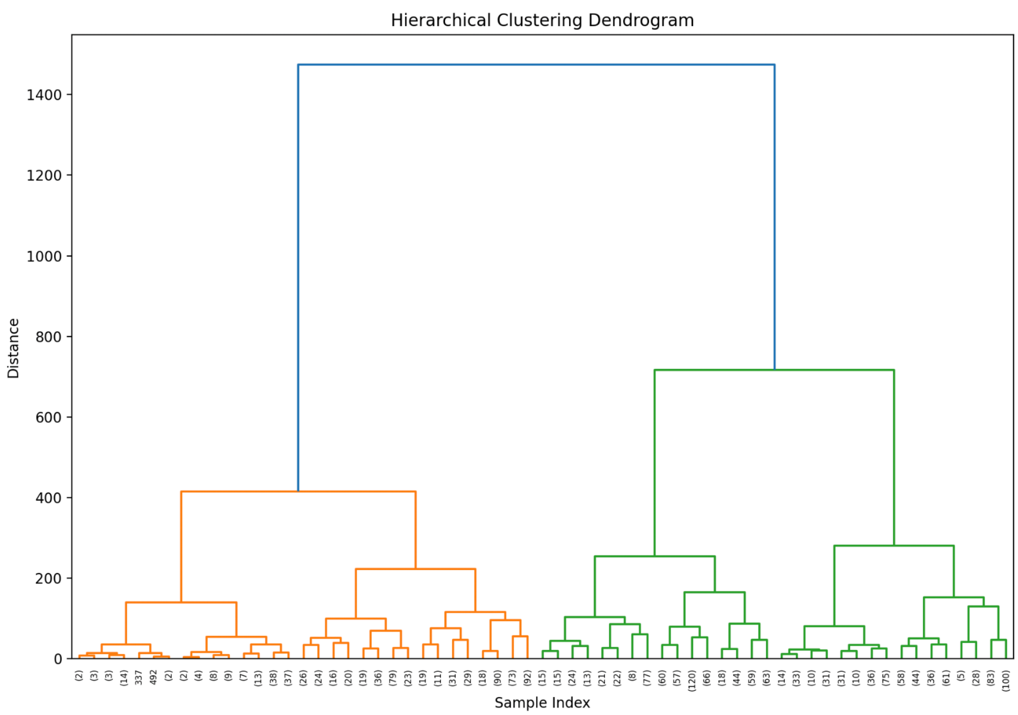
- **Including the target column in the clustering yields the same results. The best cluster is always obtained on the same number of clusters k = 2. We calculated this number using MInMaxSacling on all the scores and averaged the score for each cluster. Whichever cluster yields the highest average is chosen.**



- **Two clusters:**
    - The two clusters likely represent broad groupings in the population based on shared behavioural and physiological characteristics.
    - Potential groupings could be:
        - **Cluster 1**: Individuals closer to normal weight or having healthier lifestyles.
        - **Cluster 2**: Individuals with obesity or overweight tendencies based on BMI and related features.
- **Cluster Separation:**

    - From the scatterplot, the clusters are fairly well-separated, suggesting meaningful grouping in the PCA-reduced feature space.
    - This separation indicates that the PCA components effectively capture the variance in the original features and that K-Means clustering can differentiate distinct groups.
- **Factors:**
    - **Cluster 1**: Can include people with lower BMI, fewer unhealthy eating habits, and more physical activity
    - **Cluster 2**: Captures individuals with higher BMI, Frequent unhealthy eating, a family history of overweight.

Hierarchical Clustering:

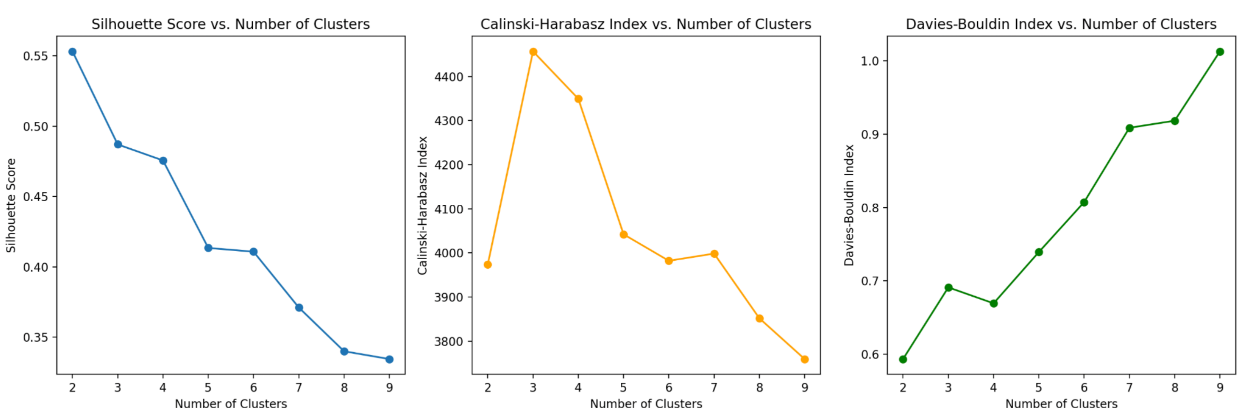


Key Observations from the Dendrogram

1. **Structure of the Dendrogram**:
    - The dendrogram reveals a clear separation of the data into two main clusters at a relatively large height (distance ~800).
    - Cutting the dendrogram at this height results in **2 primary clusters**. This is supported by the long vertical line connecting the two clusters, indicating high dissimilarity between them.

2. **Sub-Cluster Formation**:
   - At lower heights (distance ~200), smaller sub-clusters within each main cluster can be observed.
   - This suggests further divisions in the data, which could be explored if finer granularity is needed.

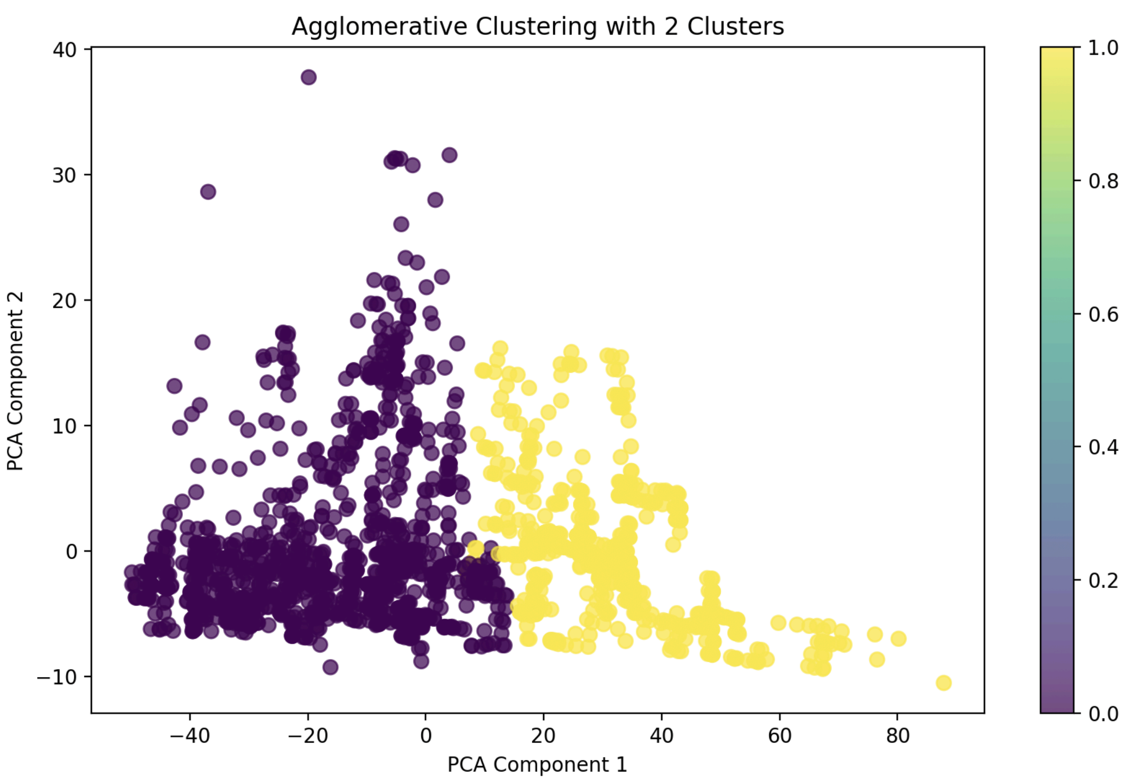
```
Hierarchical Clustering Results
   Number of Clusters  Silhouette Score  Calinski-Harabasz Index  \
0                   2          0.553126              3973.795004
1                   3          0.486990              4456.775007
2                   4          0.475618              4349.417708
3                   5          0.413437              4042.282084
4                   6          0.410793              3982.273169
5                   7          0.371111              3998.565327
6                   8          0.340028              3851.967826
7                   9          0.334509              3759.642530

   Davies-Bouldin Index
0              0.593365
1              0.691122
2              0.669340
3              0.739289
4              0.807392
5              0.908770
6              0.918445
7              1.013045

Best Results:
Best Number of Clusters (Silhouette): 2
Silhouette Score: 0.5531
Calinski-Harabasz Index: 3973.7950
Davies-Bouldin Index: 0.5934
```



Performance Metrics and Optimal Clusters

1. **Silhouette Score:**
   - The highest Silhouette Score (0.553) is observed when 2 clusters are formed. This indicates that data points are well-separated and cohesive within their respective clusters.
   - The Silhouette Score decreases consistently as the number of clusters increases, indicating reduced separation quality.

2. **Calinski-Harabasz Index:**
   - The highest Calinski-Harabasz Index (4456.775) occurs at 3 clusters, suggesting slightly better compactness and separation compared to 2 clusters.
   - However, the drop in Silhouette Score for 3 clusters indicates that the separation quality might not be as strong as for 2 clusters.
3. **Davies-Bouldin Index:**
   - The lowest Davies-Bouldin Index (0.593) is observed for 2 clusters, reinforcing that 2 clusters provide the best separation with minimal intra-cluster variance.



Agglomerative Cluster Visualization

The 2-cluster solution was visualized using PCA-reduced components:

- **Cluster 1 (Purple):** Contains data points that form a distinct, compact group.
- **Cluster 2 (Yellow):** Represents another well-separated group.
- The clear boundary between the clusters in the PCA plot further supports the 2-cluster solution as optimal.

Therefore:

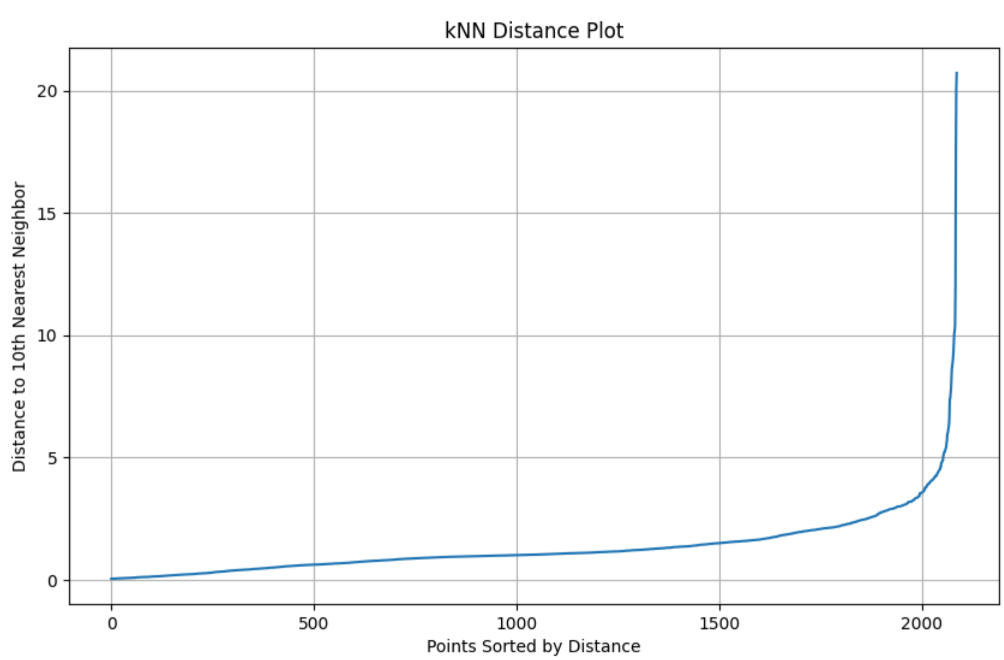
Based on the dendrogram and clustering evaluation metrics, 2 clusters is the most suitable choice for this dataset:

- It achieves the highest Silhouette Score (0.553) and the lowest Davies-Bouldin Index (0.593), indicating strong separation and compactness.
- While 3 clusters show a slightly higher Calinski-Harabasz Index, the reduction in Silhouette Score and the increase in Davies-Bouldin Index suggest that the additional cluster reduces overall clustering quality.

DBSCAN Clustering:

Role of kNN in DBSCAN

The kNN distance plot was used to estimate the eps parameter. By plotting the distance to the 10th nearest neighbor for all points and sorting them, the "elbow" point (a sharp increase in distance) around **eps=4.0** was identified. This threshold balances cluster density while minimizing noise. This informed our starting value for eps.



```
DBSCAN Normalized Scores:
Silhouette: [0.         0.3695213  0.94361671 1.         1.        ]
Calinski-Harabasz:  [0.            0.40715451  0.9981448   1.            1.
]
Inverted Davies-Bouldin:  [1.           0.64276527  0.           0.14921504
0.14921504]
Average Scores: [0.33333333 0.47314703 0.64725384 0.71640501 0.71640501]
Best eps value: 4.8
DBSCAN Results Summary:
   Eps  Min Samples  Clusters  Silhouette Score  Calinski-Harabasz Index  \
0  4.0           10         4         -0.308187                43.698329
1  4.2           10         3         -0.092836                54.360162
2  4.5           10         2          0.241739                69.835957
3  4.8           10         2          0.274599                69.884538
4  5.0           10         2          0.274599                69.884538

   Davies-Bouldin Index
0              2.110660
1              2.289275
2              2.700458
3              2.592366
4              2.592366
```

Key Metrics Evaluation

1. Silhouette Score:

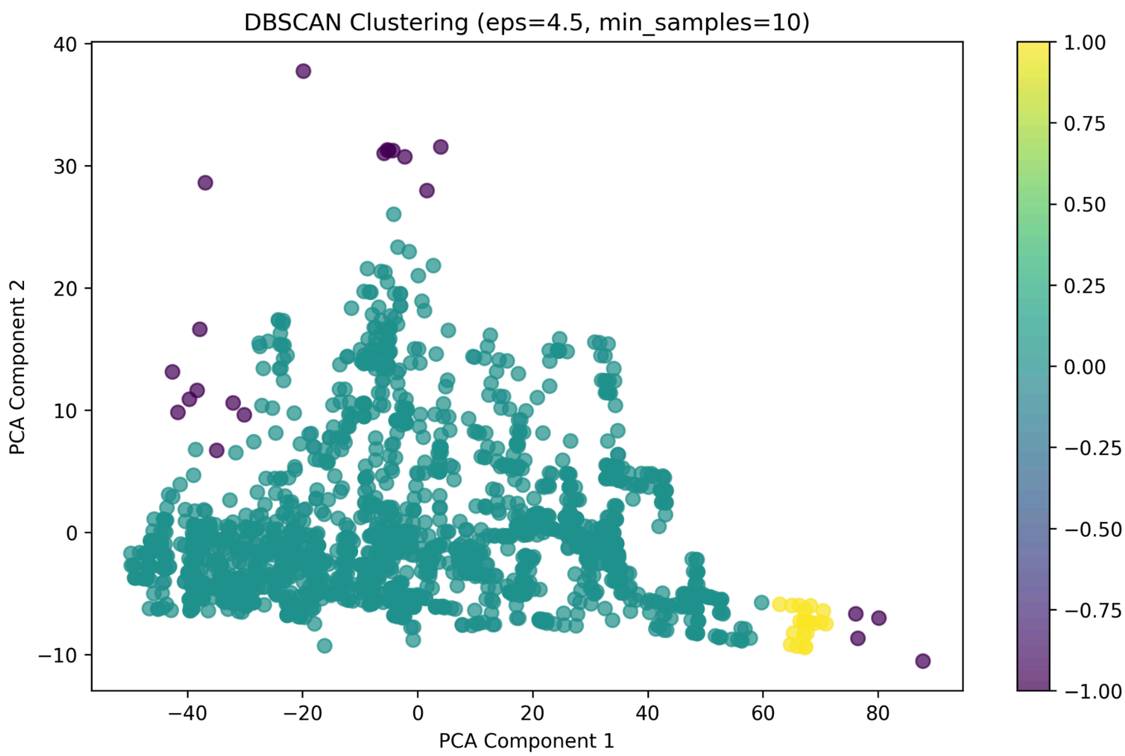
- At eps=4.0, the Silhouette Score was **-0.308**, indicating poorly defined clusters with significant overlap and noise.
- At eps=4.8, the score improved to **0.275**, showing better-defined clusters with minimized overlap.
- The increasing Silhouette Score with higher eps values highlights better cohesion and separation.
- Scores peaked at eps=4.8, suggesting this value best balances density and separation of clusters.

2. Calinski-Harabasz Index:

- At eps=4.0, the index was 43.7, suggesting loosely compact clusters.
- At eps=4.8, the index stabilized at 69.9, indicating clusters that are both compact and reasonably well-separated.

3. Davies-Bouldin Index:

- At eps=4.0, the Davies-Bouldin Index was **2.11**, showing moderate separation between clusters.
- At eps=4.8, the index slightly increased to **2.59**, reflecting slightly less distinct separation but balanced by improvements in cohesion.
- Stabilization at 4.8 confirms a reasonable trade-off between compactness and separation.



Cluster Visualization:

- **Higher eps values (>4.5)** result in better Silhouette and Calinski-Harabasz scores while reducing noise.
- **eps=4.5 and eps=4.8** yield similar results, but **eps=4.5** might be preferred for slightly better cohesion and separation based on Davies-Bouldin Index.
- For `eps=4.5`: **Clusters formed**: 2 (yellow and green regions are valid clusters).
- **Purple points** represent **noise points** (outliers) and are not considered part of any cluster.

Final Selection:

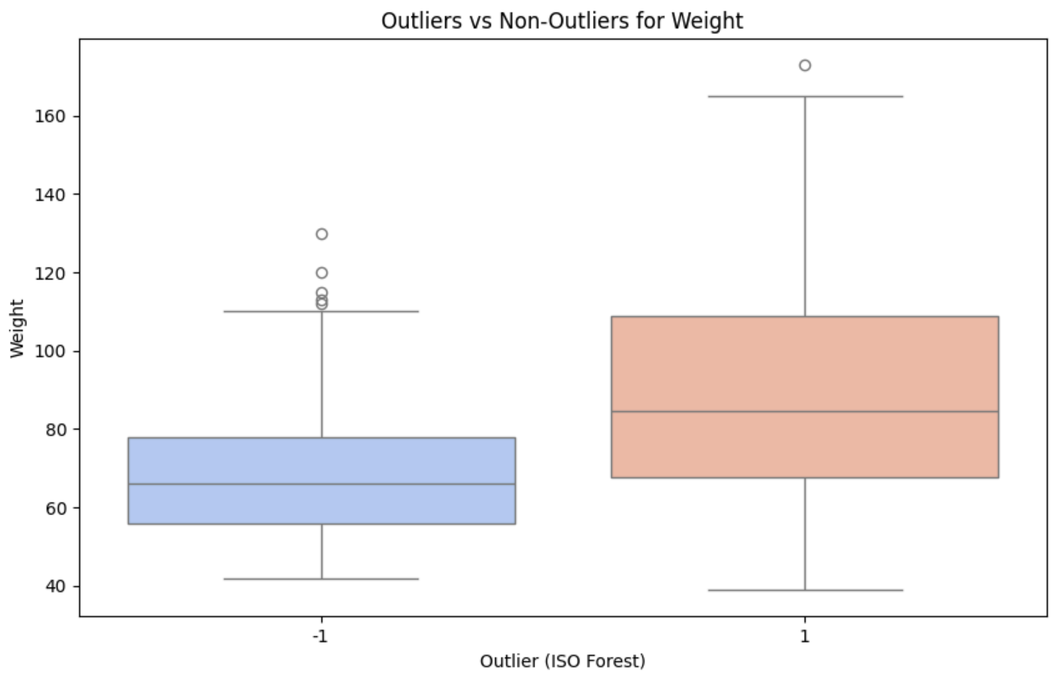
- Clustering at **eps=4.5** with **2 clusters** confirms well-separated and interpretable clusters, making it the optimal choice.

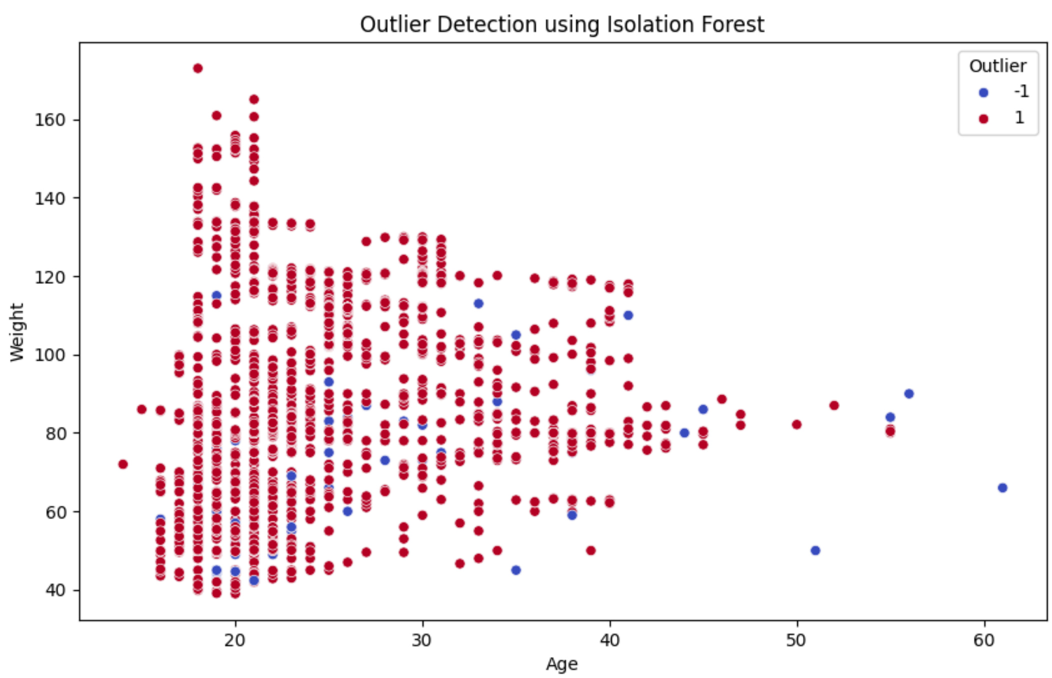
- Achieves a balance between noise reduction and maintaining meaningful clusters.
- This aligns with the results from Silhouette and Calinski-Harabasz scores, which improve significantly at this value.

Outlier Detection

We do remove the extreme outliers using z-score.

Isolation Forest:





```
Outliers for Age:
252    31.990896
92     30.990896
232    26.990896
492    20.990896
137    19.990896
Name: Age, dtype: float64
----------------------------------------
Outliers for Height:
334    0.252616
464    0.227384
165    0.217384
178    0.207384
152    0.202616
Name: Height, dtype: float64
----------------------------------------
Outliers for Weight:
339    44.858706
632    44.848706
712    44.318706
165    43.141294
600    42.668706
Name: Weight, dtype: float64
----------------------------------------
Outliers for FCVC:
68     1.421409
30     1.421409
236    1.421409
1      0.578591
333    0.578591
Name: FCVC, dtype: float64
----------------------------------------
Outliers for NCP:
399    1.701203
193    1.701203
414    1.701203
152    1.701203
384    1.701203
```

```
Name: NCP, dtype: float64
----------------------------------------
Outliers for CH2O:
416    1.004792
424    1.004792
339    1.004792
156    1.004792
350    1.004792
Name: CH2O, dtype: float64
----------------------------------------
Outliers for FAF:
1      1.987173
92     1.987173
380    1.987173
138    1.987173
356    1.987173
Name: FAF, dtype: float64
----------------------------------------
Outliers for TUE:
220    1.336943
302    1.336943
132    1.336943
334    1.336943
120    1.336943
Name: TUE, dtype: float64
----------------------------------------
Outliers for NObeyesdad:
245    3.014375
83     3.014375
712    3.014375
660    3.014375
640    3.014375
Name: NObeyesdad, dtype: float64
----------------------------------------
Outliers for BMI:
302    16.478888
519    14.008888
122    13.438888
356    13.148888
712    12.938888
```

```
Name: BMI, dtype: float64
----------------------------------------
Outliers for Gender_Female:
1      0.504073
339    0.504073
152    0.504073
156    0.504073
216    0.504073
Name: Gender_Female, dtype: float64
```

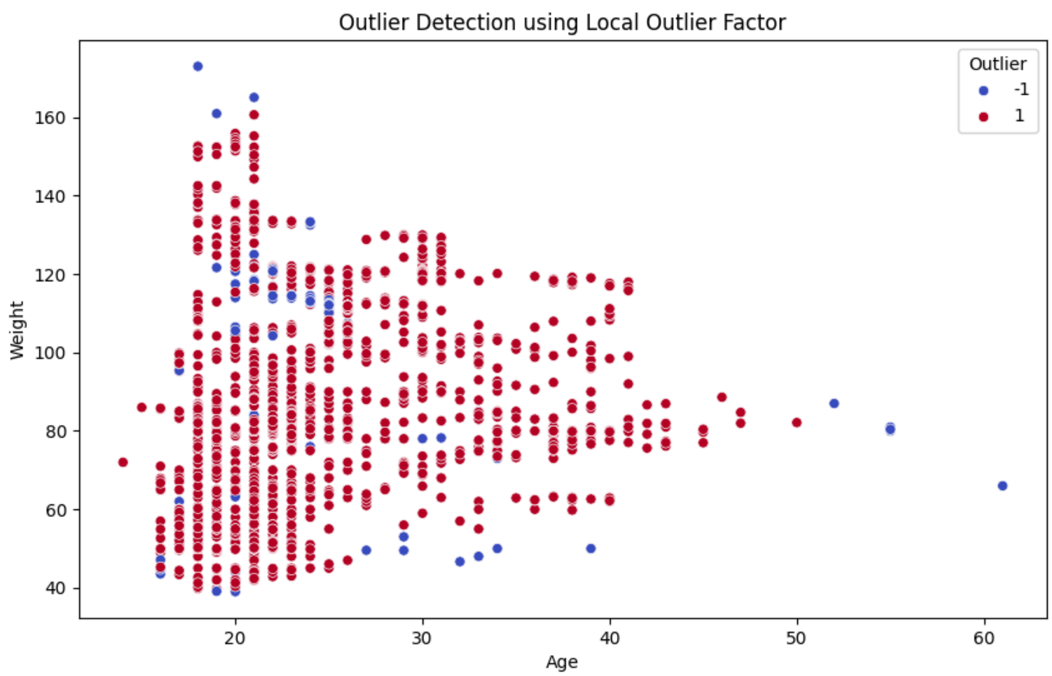
There are 105 outliers identified by the ISOLATIONFOREST() forest algorithm. We tried to find out how much they deviate from their original values, and some results have been produced above in descending order for deviation from the mean. Each of the data points above could been removed from the data, next to them is their deviation from the mean of the column. From the results, the IsolationForest() algorithm works efficiently for our dataset.

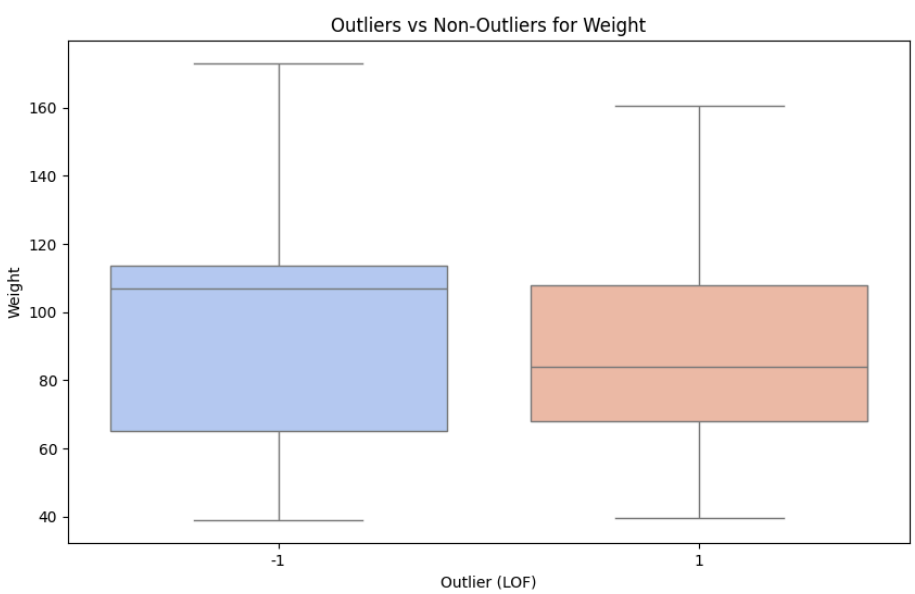
The output lists outliers for several numerical and categorical features (e.g., `Age`, `Weight`, `BMI`, `FCVC`, `SMOKE_no`, etc.).
For example:

- `Age` has outliers like `133`, `252`, `92` with significant deviations from the average.
- `Weight` shows high deviations for points like `339`, `632`, `652`, and `712`.
- `BMI` also has significant outliers, with `302`, `519`, `122`, and `356` having unusually high values.

The results show a larger variation for these data points so that we can chop them off from the original dataset. We repeated the algorithm with different contamination scores to obtain

LOF - Local outlier factor





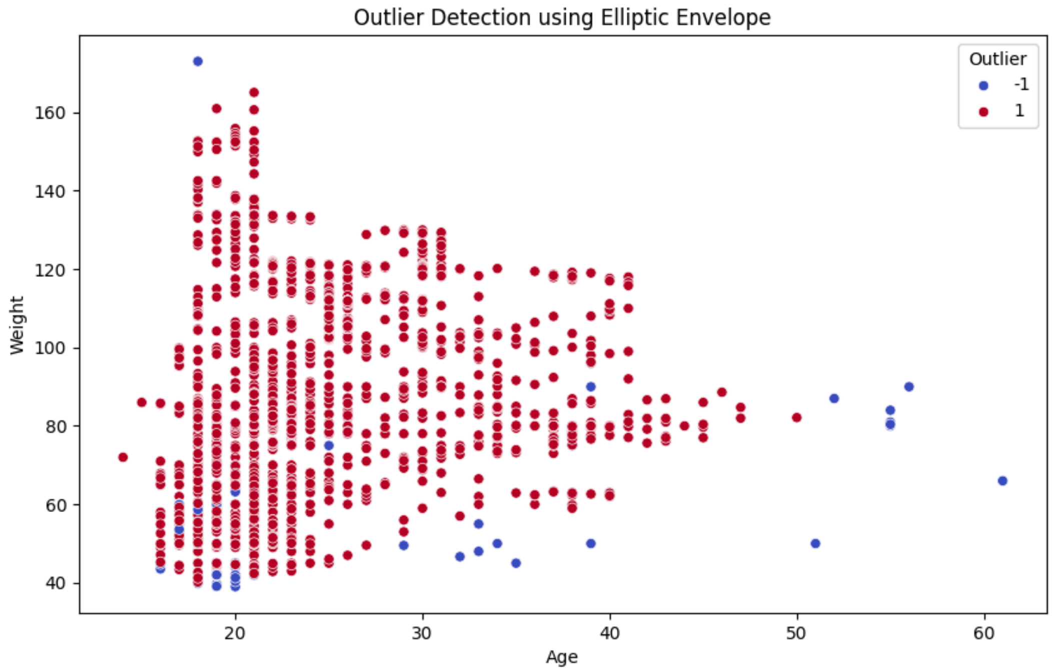

```
133      37.02775
1088     31.02775
1013     31.02775
161      31.02775
1158     31.02775
```

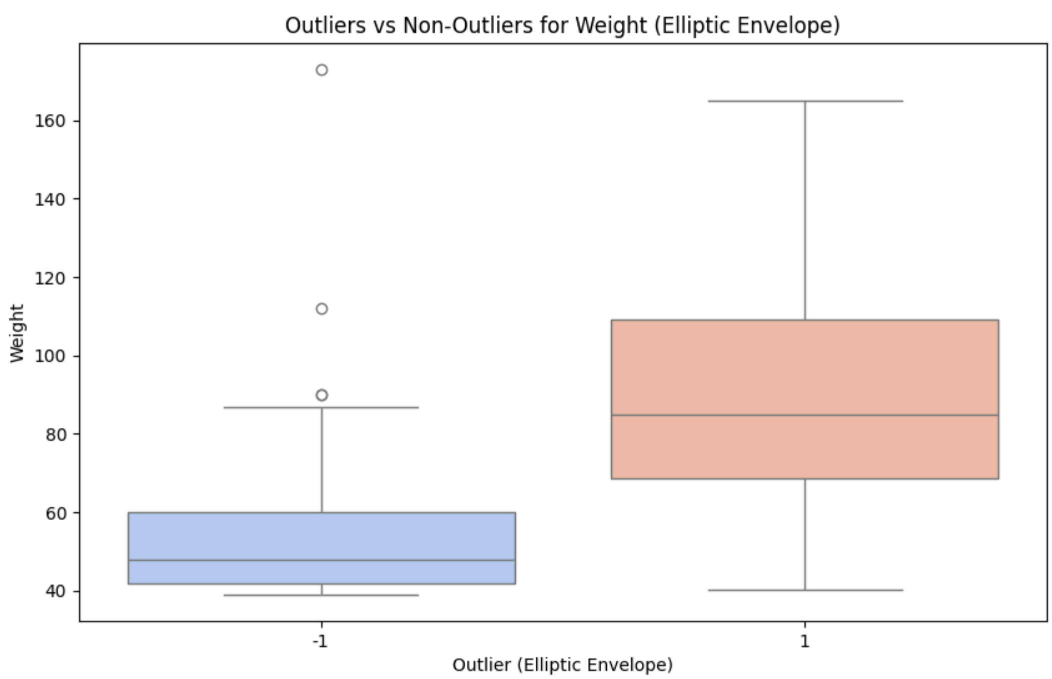
```
Name: Age, dtype: float64
----------------------------------------
Outliers for Height:
349     0.276983
1350    0.276983
1262    0.246983
1349    0.236983
1261    0.226983
Name: Height, dtype: float64
----------------------------------------
Outliers for Weight:
344     85.158885
502     77.218885
1898    73.098885
395     48.841115
725     48.741115
Name: Weight, dtype: float64
----------------------------------------
Outliers for FCVC:
395     1.413688
218     1.413688
725     1.293688
589     0.893688
1609    0.653688
Name: FCVC, dtype: float64
----------------------------------------
Outliers for NCP:
21     1.690131
86     1.690131
17     1.690131
398    1.690131
198    1.690131
Name: NCP, dtype: float64
----------------------------------------
Outliers for CH2O:
198     1.002634
51      1.002634
370     1.002634
398     1.002634
1261    0.997366
```

```
Name: CH2O, dtype: float64
----------------------------------------
Outliers for FAF:
404    2.01552
311    2.01552
395    2.01552
177    2.01552
218    2.01552
Name: FAF, dtype: float64
----------------------------------------
Outliers for TUE:
198    1.334435
395    1.334435
725    0.964435
698    0.884435
627    0.804435
Name: TUE, dtype: float64
----------------------------------------
Outliers for NObeyesdad:
266    3.066599
589    3.066599
276    3.066599
623    3.066599
627    3.066599
Name: NObeyesdad, dtype: float64
----------------------------------------
Outliers for BMI:
1898    20.132941
344     19.372941
502     18.652941
627     17.097059
698     15.967059
Name: BMI, dtype: float64
```

There are 100 outliers identified by the LOF algorithm which will be considered for removal. More than ISOLATIONFOREST() algorithm and also deviates more than it for this dataset.

Elliptic Envelope






```
Outliers for Age:
```

```
133     36.990896
252     31.990896
1158    30.990896
1088    30.990896
1013    30.990896
Name: Age, dtype: float64
----------------------------------------
Outliers for Height:
334    0.252616
865    0.242616
464    0.227384
951    0.212616
457    0.202616
Name: Height, dtype: float64
----------------------------------------
Outliers for Weight:
344    86.141294
395    47.858706
725    47.758706
589    47.488706
636    47.158706
Name: Weight, dtype: float64
----------------------------------------
Outliers for FCVC:
395    1.421409
218    1.421409
725    1.301409
588    1.211409
589    0.901409
Name: FCVC, dtype: float64
----------------------------------------
Outliers for NCP:
17     1.701203
539    1.701203
409    1.701203
605    1.701203
469    1.701203
Name: NCP, dtype: float64
----------------------------------------
Outliers for CH2O:
```

```
457    1.004792
198    1.004792
713    1.004792
522    1.004792
523    1.004792
Name: CH2O, dtype: float64
----------------------------------------
Outliers for FAF:
302    1.987173
321    1.987173
25     1.987173
445    1.987173
434    1.987173
Name: FAF, dtype: float64
----------------------------------------
Outliers for TUE:
588    1.336943
198    1.336943
334    1.336943
302    1.336943
271    1.336943
Name: TUE, dtype: float64
----------------------------------------
Outliers for BMI:
344    19.701112
627    16.768888
302    16.478888
698    15.638888
198    13.948888
Name: BMI, dtype: float64
----------------------------------------
Outliers for Gender_Female:
17     0.504073
588    0.504073
623    0.504073
615    0.504073
614    0.504073
Name: Gender_Female, dtype: float64
```

Now that we have all the data needed to perform the chops. One approach, that we have decided on is to identify common rows among these outliers as the graphs suggest that most outliers could be potentially real data. The heatmap from earlier suggested that Age, and Weight have the most correlation from other columns. So, we don't have to sacrifice real data, and just chop off common rows among the entire dataset. We obtained 14 common outliers which could be removed, to improve the accuracy. We also chopped off extreme value cases to produce better results in our classifier using Z-score values.

Feature Selection

1. Mutual Information:

These results were obtained from one of the runs. We first calculated the mutual information(MI) of numerical columns and ran through one iteration of KNN. One significant boost in our results is obtained through the inclusion of categorical data columns. The results obtained are shown above with different classifiers playing a role.

```
Mutual Information Scores:
  Feature  MI Score
2  Weight  0.429766
0     Age  0.156311
1  Height  0.119271
7     TUE  0.092754
6     FAF  0.091266
5    CH2O  0.055161
4     NCP  0.046697
3    FCVC  0.009528
Selected Features by Mutual Information: ['Weight', 'Age', 'Height', 'TUE', 'FAF', 'CH2O', 'NCP']
Cross-Validation Scores: [0.92113565 0.92113565 0.92429022 0.90851735 0.94637224]
Mean Cross-Validation Score: 0.9242902208201894
Accuracy: 0.9370277078085643
Precision: 0.9475409836065574
Recall: 0.9697986577181208
F1 Score: 0.9585406301824212
AUC-ROC: 0.982746932411362
```

Key observations:

- While MI provided a good starting point for selecting numerical features, ignoring categorical features led to suboptimal results because important predictors were excluded.
- Many categorical features capture unique and essential information about obesity-related factors that numerical features cannot fully explain.
- The performance improvement is observed across various classifiers (e.g., KNN, Random Forest, SVM), indicating that including categorical features is universally beneficial for the classification task.

Classification

Results from one of the runs:

```
Model: Random Forest

Classification Report:
                      precision    recall  f1-score   support

Insufficient_Weight       0.98      0.97      0.97        59
      Normal_Weight       0.91      0.98      0.94        61
     Obesity_Type_I       1.00      0.99      0.99        70
    Obesity_Type_II       1.00      1.00      1.00        64
   Obesity_Type_III       1.00      1.00      1.00        60
 Overweight_Level_I       1.00      0.91      0.95        55
Overweight_Level_II       0.96      1.00      0.98        49

           accuracy                           0.98       418
          macro avg       0.98      0.98      0.98       418
       weighted avg       0.98      0.98      0.98       418

Model: Gradient Boosting

Classification Report:
                      precision    recall  f1-score   support

Insufficient_Weight       1.00      0.97      0.98        59
      Normal_Weight       0.95      1.00      0.98        61
     Obesity_Type_I       1.00      0.97      0.99        70
    Obesity_Type_II       0.97      1.00      0.98        64
   Obesity_Type_III       1.00      1.00      1.00        60
 Overweight_Level_I       0.98      0.89      0.93        55
```

```
Overweight_Level_II       0.91      0.98      0.94        49

           accuracy                           0.97       418
          macro avg       0.97      0.97      0.97       418
       weighted avg       0.97      0.97      0.97       418

Model: SVM

Classification Report:
                     precision    recall  f1-score   support

Insufficient_Weight       0.75      1.00      0.86        59
      Normal_Weight       0.81      0.48      0.60        61
     Obesity_Type_I       0.90      0.51      0.65        70
    Obesity_Type_II       0.88      0.95      0.92        64
   Obesity_Type_III       0.97      0.95      0.96        60
 Overweight_Level_I       0.65      0.60      0.62        55
Overweight_Level_II       0.44      0.76      0.56        49

           accuracy                           0.75       418
          macro avg       0.77      0.75      0.74       418
       weighted avg       0.78      0.75      0.74       418

Model: KNN

Classification Report:
                     precision    recall  f1-score   support

Insufficient_Weight       0.89      1.00      0.94        59
      Normal_Weight       1.00      0.75      0.86        61
     Obesity_Type_I       1.00      0.99      0.99        70
    Obesity_Type_II       0.98      0.98      0.98        64
   Obesity_Type_III       0.98      1.00      0.99        60
 Overweight_Level_I       0.87      0.95      0.90        55
Overweight_Level_II       0.94      1.00      0.97        49

           accuracy                           0.95       418
          macro avg       0.95      0.95      0.95       418
       weighted avg       0.96      0.95      0.95       418
```

```
Model: Decision Tree

Classification Report:
                      precision    recall  f1-score   support

Insufficient_Weight       1.00      0.97      0.98        59
      Normal_Weight       0.95      0.89      0.92        61
     Obesity_Type_I       0.99      0.97      0.98        70
    Obesity_Type_II       0.97      0.98      0.98        64
   Obesity_Type_III       1.00      1.00      1.00        60
 Overweight_Level_I       0.85      0.91      0.88        55
Overweight_Level_II       0.92      0.96      0.94        49

           accuracy                           0.95       418
          macro avg       0.95      0.95      0.95       418
       weighted avg       0.96      0.95      0.95       418
```

Hyperparameter tuning

With Hyperparameter tuning:

```
Model: Random Forest

Classification Report (ss):
                      precision    recall  f1-score   support

Insufficient_Weight       1.00      0.96      0.98        46
      Normal_Weight       0.90      1.00      0.95        54
     Obesity_Type_I       1.00      1.00      1.00        64
    Obesity_Type_II       1.00      0.98      0.99        64
   Obesity_Type_III       0.99      1.00      0.99        70
 Overweight_Level_I       0.98      0.92      0.95        53
Overweight_Level_II       0.98      0.97      0.98        64

           accuracy                           0.98       415
          macro avg       0.98      0.98      0.98       415
       weighted avg       0.98      0.98      0.98       415

Model: Gradient Boosting
```

```
Classification Report (ss):
                      precision    recall  f1-score   support

Insufficient_Weight       0.98      0.98      0.98        46
      Normal_Weight       0.98      0.96      0.97        54
     Obesity_Type_I       0.98      1.00      0.99        64
    Obesity_Type_II       1.00      0.97      0.98        64
   Obesity_Type_III       0.99      1.00      0.99        70
 Overweight_Level_I       0.98      1.00      0.99        53
Overweight_Level_II       1.00      1.00      1.00        64

           accuracy                           0.99       415
          macro avg       0.99      0.99      0.99       415
       weighted avg       0.99      0.99      0.99       415

Model: SVM

Classification Report (ss):
                      precision    recall  f1-score   support

Insufficient_Weight       0.96      0.93      0.95        46
      Normal_Weight       0.95      0.96      0.95        54
     Obesity_Type_I       0.98      1.00      0.99        64
    Obesity_Type_II       1.00      0.98      0.99        64
   Obesity_Type_III       0.99      1.00      0.99        70
 Overweight_Level_I       1.00      1.00      1.00        53
Overweight_Level_II       1.00      0.98      0.99        64

           accuracy                           0.98       415
          macro avg       0.98      0.98      0.98       415
       weighted avg       0.98      0.98      0.98       415

Model: KNN

Classification Report (ss):
                      precision    recall  f1-score   support

Insufficient_Weight       0.94      0.96      0.95        46
      Normal_Weight       0.96      0.80      0.87        54
     Obesity_Type_I       1.00      1.00      1.00        64
```

```
    Obesity_Type_II       1.00      1.00      1.00        64
   Obesity_Type_III       1.00      1.00      1.00        70
 Overweight_Level_I       0.87      1.00      0.93        53
Overweight_Level_II       0.98      0.98      0.98        64

           accuracy                           0.97       415
          macro avg       0.96      0.96      0.96       415
       weighted avg       0.97      0.97      0.97       415

Model: Decision Tree

Classification Report (ss):
                      precision    recall  f1-score   support

Insufficient_Weight       1.00      0.96      0.98        46
      Normal_Weight       0.95      0.98      0.96        54
     Obesity_Type_I       1.00      1.00      1.00        64
    Obesity_Type_II       1.00      0.98      0.99        64
   Obesity_Type_III       0.99      1.00      0.99        70
 Overweight_Level_I       0.96      0.96      0.96        53
Overweight_Level_II       0.98      0.98      0.98        64

           accuracy                           0.98       415
          macro avg       0.98      0.98      0.98       415
       weighted avg       0.98      0.98      0.98       415
```

SVM had a boot in it score, which strengthens the fact that our tuning worked. Why would this happen? The improvement in the SVM model's performance after hyperparameter tuning is likely due to better alignment of the hyperparameters (`C`, `gamma`, and `kernel`). We don't see much changes in other classifiers.

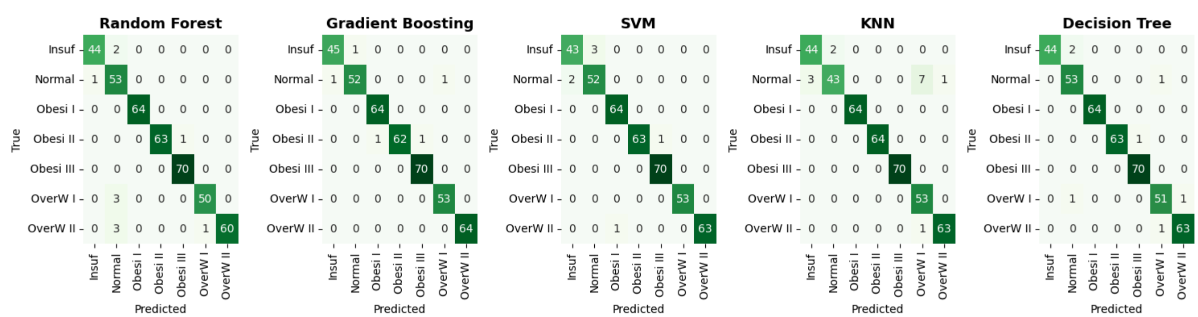
```
'SVM': {
        'C': [0.1, 1, 10],
        'gamma': [0.1, 1, 10],
        'kernel': ['rbf', 'linear']
    },
```

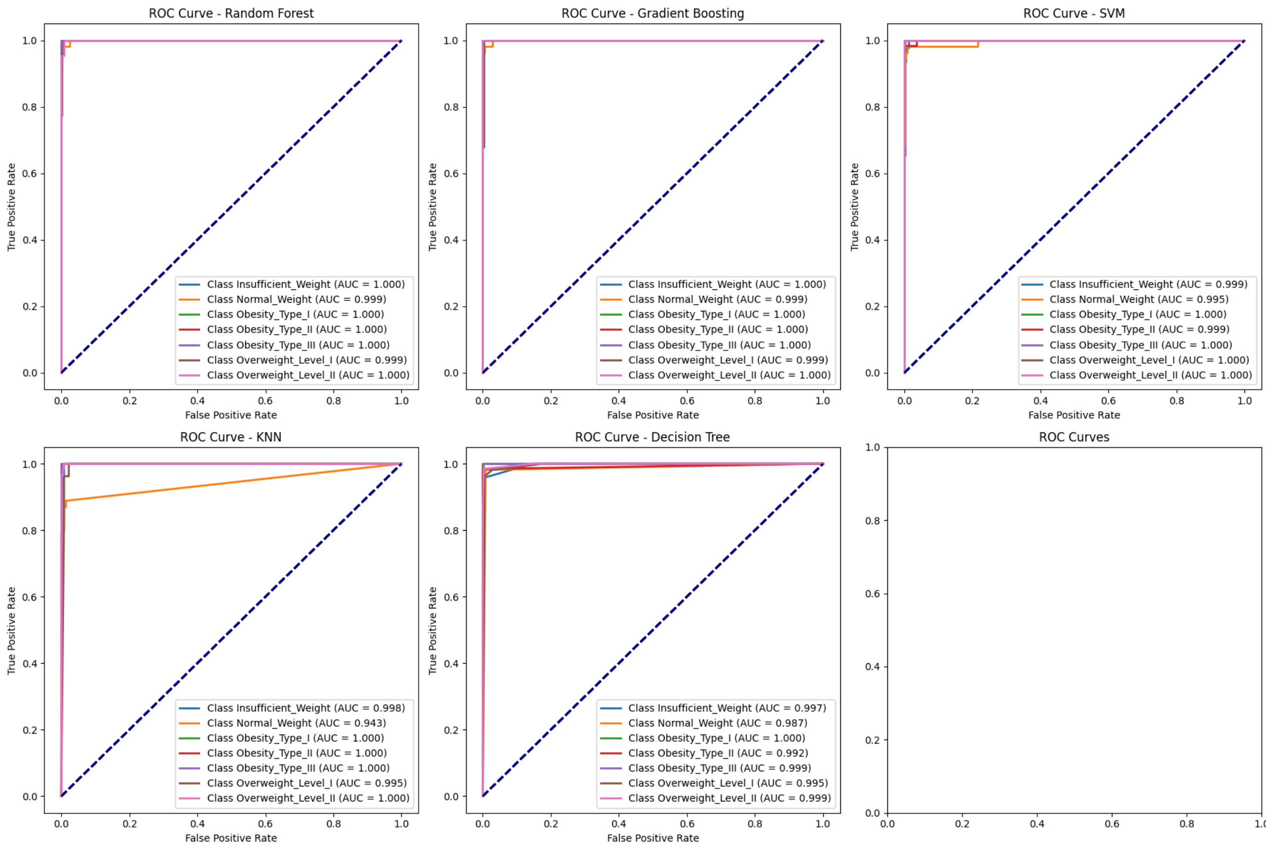
C parameter: The `C` parameter controls the trade-off between achieving a low error on the training data and maintaining a smooth decision boundary.A high `C` focuses on fitting the training data more precisely, potentially capturing complex relationships in the data. During tuning, the optimal `C` was likely found, balancing overfitting and underfitting for your data.

gamma parameter: `gamma` defines how far the influence of a single training example reaches. A high `gamma` means the model focuses on closer points, potentially leading to overfitting, while a low `gamma` considers points further away, resulting in a smoother decision boundary. The tuned `gamma` might have improved the model's ability to capture non-linear relationships in your data, especially if there are clusters or complex patterns.

Kernel Parameter: The kernel determines how SVM maps input data into a higher-dimensional space for better separability. The `linear` kernel works well for linearly separable data, while the `rbf` kernel captures non-linear relationships.

GridSearchCV: The grid search systematically explored combinations of `C`, `gamma`, and `kernel`, identifying the optimal set of parameters for your data. Before tuning, the default parameters might not have been suitable for your dataset, leading to suboptimal performance.




ROC Curve - Random Forest
ROC Curve - Gradient Boosting
ROC Curve - SVM
ROC Curve - KNN
ROC Curve - Decision Tree
ROC Curves
True Positive Rate
False Positive Rate
Random Forest: Class Insufficient_Weight (AUC = 1.000)
Class Normal_Weight (AUC = 0.999)
Class Obesity_Type_I (AUC = 1.000)
Class Obesity_Type_II (AUC = 1.000)
Class Obesity_Type_III (AUC = 1.000)
Class Overweight_Level_I (AUC = 0.999)
Class Overweight_Level_II (AUC = 1.000)
Gradient Boosting: Class Insufficient_Weight (AUC = 1.000)
Class Normal_Weight (AUC = 0.999)
Class Obesity_Type_I (AUC = 1.000)
Class Obesity_Type_II (AUC = 1.000)
Class Obesity_Type_III (AUC = 1.000)
Class Overweight_Level_I (AUC = 0.999)
Class Overweight_Level_II (AUC = 1.000)
SVM: Class Insufficient_Weight (AUC = 0.999)
Class Normal_Weight (AUC = 0.995)
Class Obesity_Type_I (AUC = 1.000)
Class Obesity_Type_II (AUC = 0.999)
Class Obesity_Type_III (AUC = 1.000)
Class Overweight_Level_I (AUC = 1.000)
Class Overweight_Level_II (AUC = 1.000)
KNN: Class Insufficient_Weight (AUC = 0.998)
Class Normal_Weight (AUC = 0.943)
Class Obesity_Type_I (AUC = 1.000)
Class Obesity_Type_II (AUC = 1.000)
Class Obesity_Type_III (AUC = 1.000)
Class Overweight_Level_I (AUC = 0.995)
Class Overweight_Level_II (AUC = 1.000)
Decision Tree: Class Insufficient_Weight (AUC = 0.997)
Class Normal_Weight (AUC = 0.987)
Class Obesity_Type_I (AUC = 1.000)
Class Obesity_Type_II (AUC = 0.992)
Class Obesity_Type_III (AUC = 0.999)
Class Overweight_Level_I (AUC = 0.995)
Class Overweight_Level_II (AUC = 0.999)

Learnings and Outcome

Best Classifier with hyper tuning:

```
Best Classifier: Gradient Boosting
Best F1-Score: 0.9879
```

Insights about the Domain

Obesity is closely linked to age, weight, and lifestyle factors like food consumption and family history. THis dataset highlights the importance of both biological and behavioural factors in the development of obesity. The synthetic data with merged with the real data, limited our scalability, and methodologies that could have used to preprocess the data.

Lessons Learned about Data Mining

- **Feature Engineering**: Adding features like **BMI** significantly improved the model's ability to predict obesity types.
- **Outlier Detection**: Identifying and removing outliers, especially using techniques like **Isolation Forest**, can greatly improve model performance by removing noise. In our case, the synthetic data, had little to no noise which could be chopped off. Although, we did find a way around this, by removing the common outliers obtained by the three detection methods.
- **Clustering**: Clustering helped segment individuals based on shared characteristics, offering further insight into the different obesity types. The k=2 clustering for k-means divided unhealthy and healthy points from the dataset with a higher average score in all algorithms.
- **Hyperparameter Tuning**: Proper hyperparameter tuning, especially for models like SVM, can lead to significant improvements in predictive performance, The best overall performance with tuning, was shown by Gradient boosting, Each classifier algorithm does not overfit or underfit the data, as both of them stay close to each other.

```
Model: Decision Tree

Training Accuracy: 0.9832

Test Accuracy: 0.9545
```

```
Model: KNN
Training Accuracy: 0.9754
Test Accuracy: 0.9593
```

```
Model: SVM
```

```
Training Accuracy: 0.9862
Test Accuracy: 0.9737
```

Contribution

Bhavneet

Kmeans (clustering), Elliptic Envelope (outlier detection), Random forest (Classification)

- Helped various aspects of the project such as visualization and data analysis, including a new column BMI (Body Mass Index, calculated from the dataset.
- Refined the whole classification algorithm to obtain hyper-tuned and efficient classifiers.
- Visualization of Confusion matrix and ROC curves for different classifiers
- Wrote EDA for relationships between different columns, to help identify correlations and build a better classifier.
- Outlier detection strategies were refined and cleaned up
- Recorded each outcome onto the report, to help with data analysis

Damanpreet

Hierarchical (clustering), LOF(outlier detection), KNN classification and hyperparameter tuning

- Recorded each outcome onto the report, to help with data analysis
- Helped with ideation aspects of project such how project can be proceeded and multiple outcomes model and plotting on ROC curves

Aroofa

DBSCAN (clustering), Isolation forerst (outlier detection), SVM (Classification) and Feature Selection

- Helped with data analysis and visualization through clustering, outlier detection and classification.
- Recorded the outcomes onto the report and evaluated the methods used.
- Feature selection using Mututal information
- Computed the best classifier from the results of hyperparameter tunning
- Visualized ROC curve and relevant plots to the report