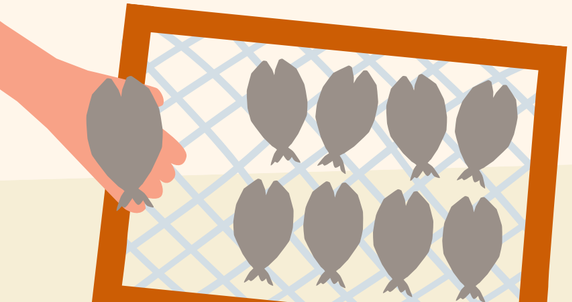
SC1015 Mini Project: Factors Influencing the Risk of Obesity

ECDS Group 7:
Deleah, Nadya, Ying Jie

Contents

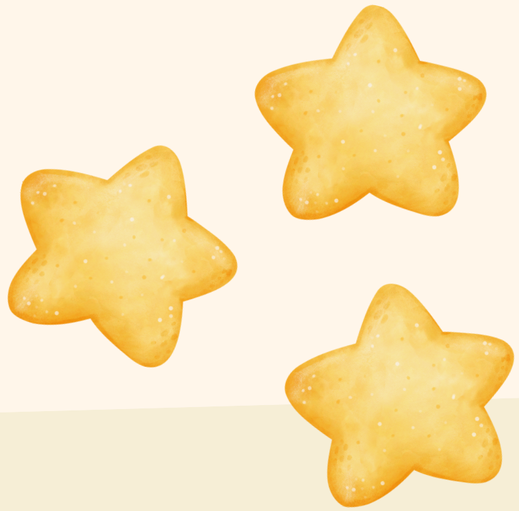


1. Problem Motivation

World Health Organization

According to World Health Organisation,

- In 2022, 1 in 8 people in the world were living with Obesity.
- Worldwide Adult Obesity has more than Doubled since 1990, and adolescent Obesity has Quadrupled.



1. Problem Motivation

The benefits of not being obese

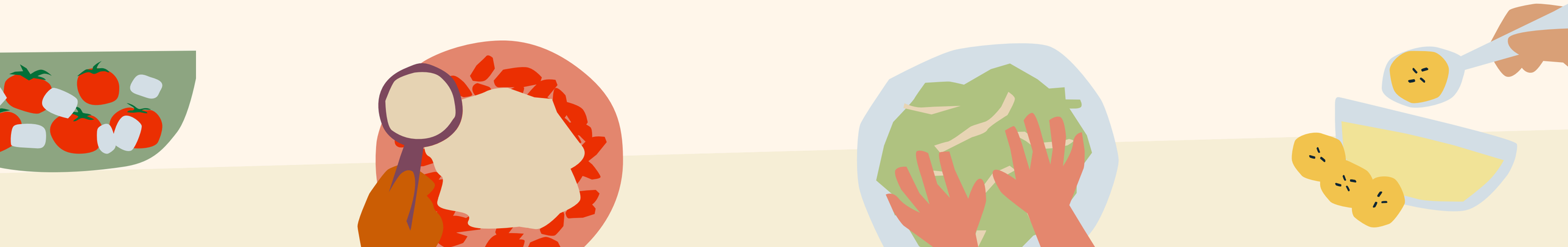
The advantages of losing weight are not merely physical!

- Improved Health
- Enhanced Quality of Life
- Better Mental Health
- Increased Life Expectancy

1. Problem Motivation

The BMI Threshold for Obesity differs from place to place.

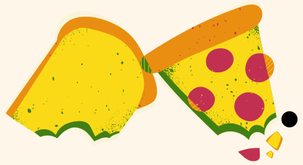
In line with Singapore's standard, we define Obesity as BMI >= 27.5.

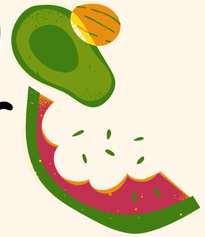


1. Problem Motivation

Our Dataset

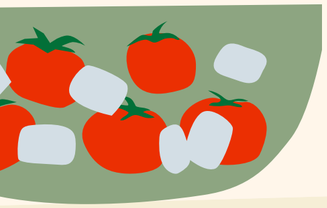
- This dataset contains information on 2111 residents aged 14 to 61 Years Old from Mexico, Peru and Colombia including details about their lifestyle and whether they classified as Obese.

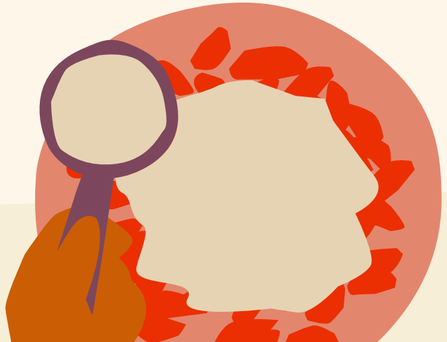




Problem Definition

- Which factor(s) are the most important in predicting obesity?
- Which model does this best?





2. Data Preparation and Cleaning

Remove Certain Irrelevant Columns:

1. Frequency of Consumption of Vegetables (FCVC)
2. Consumption of Water Daily (CH20)
3. Time using Technology Devices (TUE)
4. Transportation Used (MTRANS)
5. Number of Main Meals (NCP)

```
# Columns to remove
columns_to_drop = ['FCVC', 'MTRANS', 'SMOKE', 'CH2O', 'SCC', 'CALC', 'TUE', 'NObeyesdad', 'NCP', 'Age', 'Gender']

# Drop the columns
df_cleaned = df.drop(columns_to_drop, axis=1)
```

- Adding BMI Indicator as a Column

```
# Calculate BMI
df_cleaned['BMI'] = df['Weight'] / (df['Height']**2)
```

2. Data Preparation and Cleaning

- Renaming Columns to Increase Readibility

```
# Rename Columns to increase readability
df_cleaned = df_cleaned.rename(columns={
    'family_history_with_overweight': 'family_ob_hist',
    'FAVC': 'freq_high_cal_food',
    'CALC_numeric': 'consumption_of_alcohol',
    'FAF': 'phy_act_freq'
})
```

- Adding Obesity = 'True' or 'False' Column

```
# Add new column ('Obesity' - True ; False)
df_cleaned['Obesity'] = df_cleaned['BMI'] >= 27.5
```

2. Data Preparation and Cleaning

- Changing the levels 'Yes', 'No' of family_ob_hist and freq_high_cal_food to 1 and 0 respectivey

```
# Replace Values (0- No; 1-Yes)
df_cleaned['family_ob_hist'] = df_cleaned['family_ob_hist'].replace({'no': 0, 'yes': 1})
df_cleaned['freq_high_cal_food'] = df_cleaned['freq_high_cal_food'].replace({'no': 0, 'yes': 1})
```

- Changing the Indicator of Alcohol Consumption to Numeric Figures

```
# Standardize the CALC column to lowercase before dropping it
df['CALC'] = df['CALC'].str.lower()

# Mapping values for alcohol consumption
alc_mapping = {'always': 3, 'frequently': 2, 'sometimes': 1, 'no': 0}

# Apply mapping to the 'CALC' column before dropping it
df_cleaned['CALC_numeric'] = df['CALC'].map(alc_mapping)
```

2. Data Preparation and Cleaning

- Removing Outliers

```
Outliers in Height:
     Height  Weight  family_ob_hist  freq_high_cal_food  phy_act_freq  \
349    1.98   125.0               1                   1           1.0

     consumption_of_alcohol        BMI  Obesity
349                       1  31.884502     True

Outliers in Weight:
     Height  Weight  family_ob_hist  freq_high_cal_food  phy_act_freq  \
344    1.87   173.0               1                   1           2.0

     consumption_of_alcohol       BMI  Obesity
344                       1  49.47239     True
```

1 height outlier detected!

and 1 weight outlier detected!

3. Exploratory Data Analysis

- After cleaning the dataset, we proceed to explore the new dataset by examining the information about the variables

```
Index: 2109 entries, 0 to 2110
Data columns (total 6 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   family_ob_hist         2109 non-null   int64
 1   freq_high_cal_food     2109 non-null   int64
 2   phy_act_freq           2109 non-null   float64
 3   consumption_of_alcohol 2109 non-null   int64
 4   BMI                    2109 non-null   float64
 5   Obesity                2109 non-null   bool
dtypes: bool(1), float64(2), int64(3)
```

gives us an idea how we should analyze each variable according to their type later on !

3. Exploratory Data Analysis

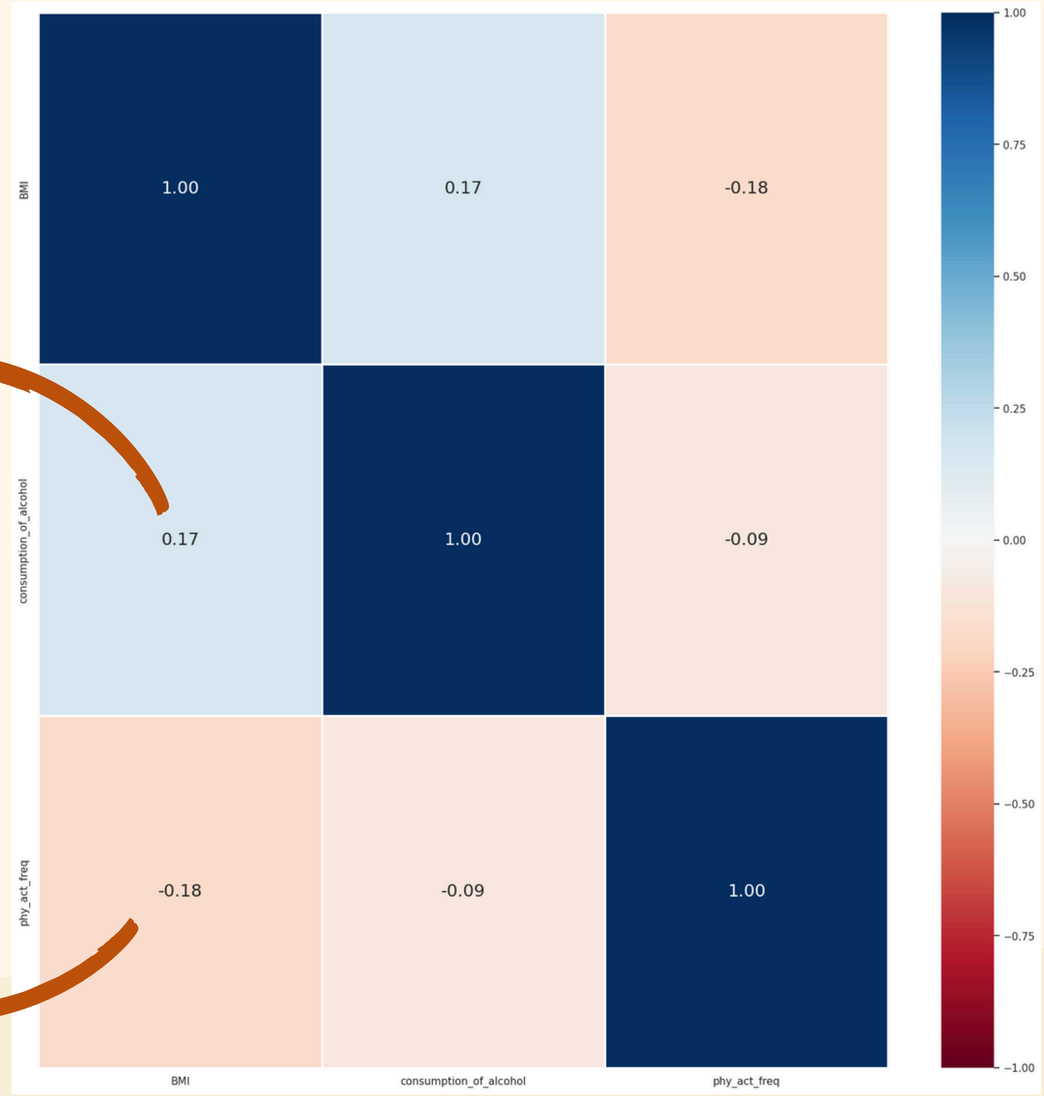
To analyze the relationship between the 3 numeric variables, namely:

1. consumption_of_alcohol
2. phy_act_freq
3. BMI

We will plot the heatmap of the correlation matrix

```
# Correlation Matrix
print (numeric_data.corr())

#Heatmap of the Correlation Matrix
f = plt.figure(figsize=(20, 20))
sb.heatmap(numeric_data.corr(), vmin = -1, vmax = 1, linewidths = 1,
           annot = True, fmt = ".2f", annot_kws = {"size": 18}, cmap = "RdBu")
```

3. Exploratory Data Analysis



low positive correlation between BMI and consumption_of_alcohol!

low negative correlation between BMI and phy_act_freq!

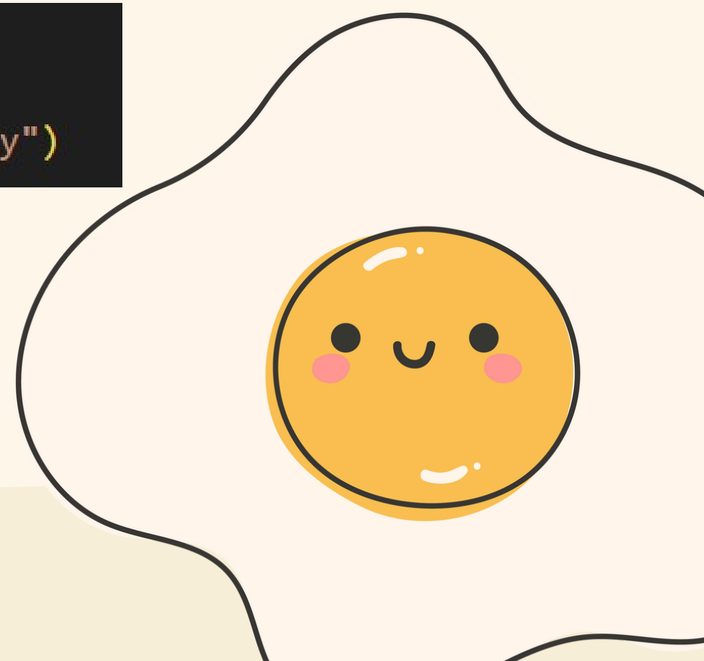
3. Exploratory Data Analysis

We now go on to analyze how the 2 categorical variables in the dataset are related to obesity :

1. family_ob_hist
2. freq_high_cal_food

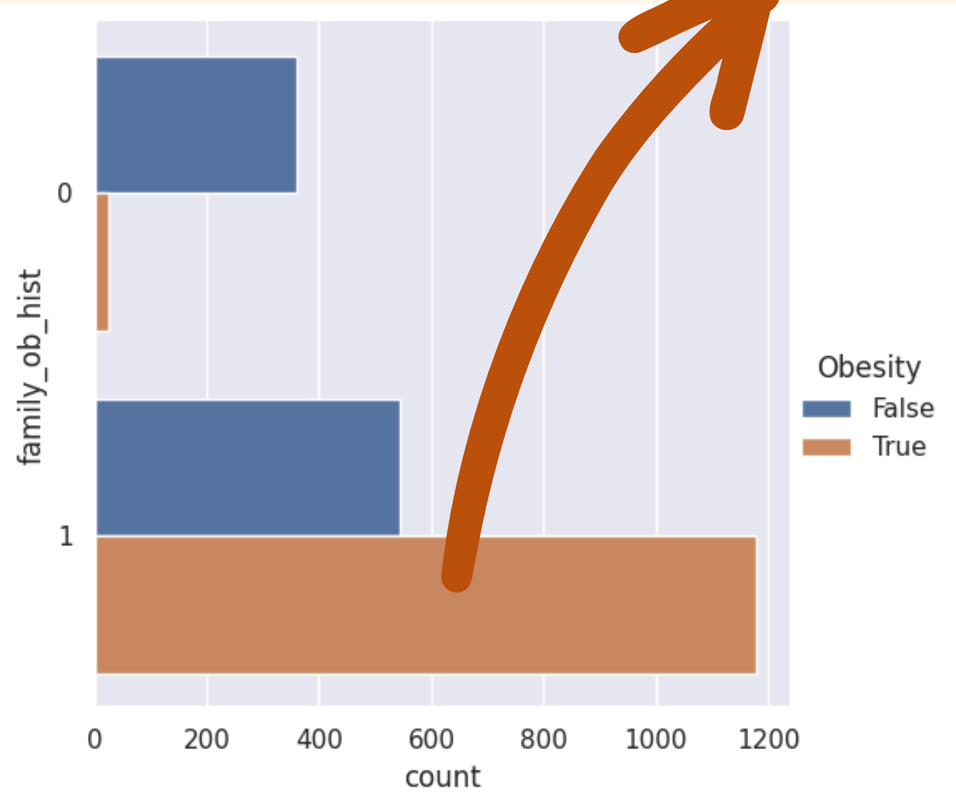
Categorical plots can be used to analyze the relationships

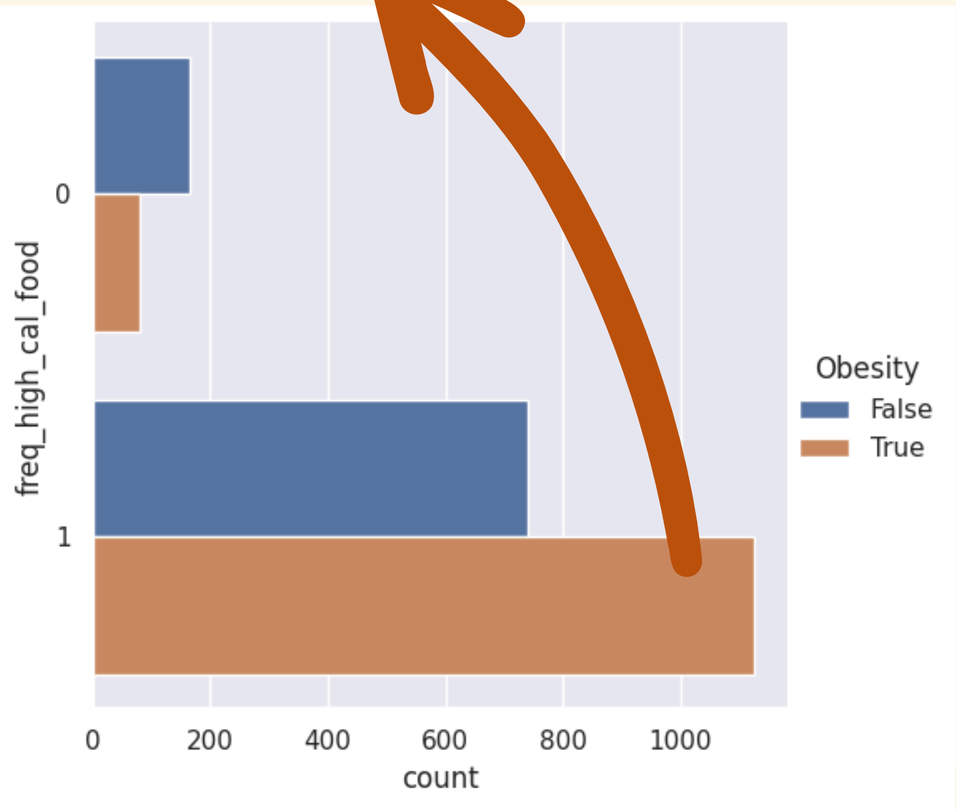
```
#Analyze the relationship between categorical variables and obesity
for var in categorical_data:
    sb.catplot(y = var, data = df_cleaned, kind = "count", hue = "Obesity")
```



3. Exploratory Data Analysis

significantly greater count of obese people with family obese history backgrounds and high caloric food intake!





4. Data Splitting

- 🎯 Target Variable (y):
- Obesity column

- 🧠 Predictor Variables (X):

1. family_ob_hist
2. freq_high_cal_food
3. phy_act_freq
4. consumption_of_alcohol

- 🔀 Train-Test Split:
- 75% training and 25% testing.

```
from sklearn.model_selection import train_test_split
random_seed = 11

# Extract Response and Predictors
y = pd.DataFrame(df_cleaned['Obesity'])
X = pd.DataFrame(df_cleaned[["family_ob_hist", "freq_high_cal_food", "phy_act_freq", "consumption_of_alcohol"]])


# Split the Dataset into Train and Test (0.75:0.25 ratio)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state=random_seed)

#Save Train and Test Datasets as Csv Files
X_train.to_csv('X_train.csv', index=False)
X_test.to_csv('X_test.csv', index=False)
y_train.to_csv('y_train.csv', index=False)
y_test.to_csv('y_test.csv', index=False)
```

4. Binary Trees – Individual Factors

- 🔁 Looped Analysis:
- Repeated decision tree modeling for each predictor variable individually

- ✂ Train-Test Split (per variable):
- Each variable was split independently using a 75% training / 25% testing ratio.

- 🧠 Model Training:
- Trees were limited to a max depth of 3 for simplicity and interpretability.

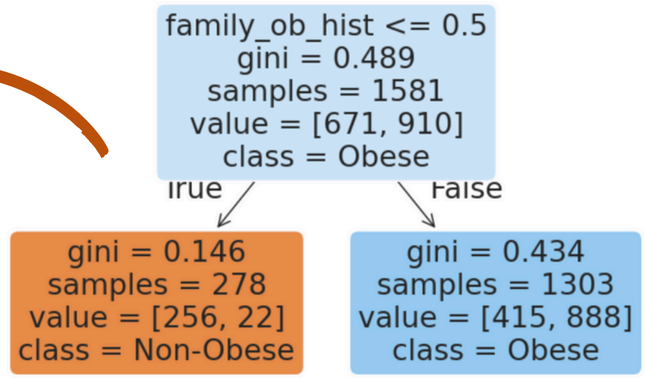
- 📈 Evaluation Metrics:
- Calculated accuracy for each model.
- Displayed confusion matrices

```
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.metrics import accuracy_score, confusion_matrix

# Set random seed and define target/predictors
random_seed = 11
y = pd.DataFrame(df_cleaned['Obesity'])
predictor_columns = ["family_ob_hist", "freq_high_cal_food", "phy_act_freq", "consumpti

# Loop through each predictor
for predictor in predictor_columns:
    print(f"\n=== Decision Tree for: {predictor} ===")

    # Single predictor dataframe
    X = pd.DataFrame(df_cleaned[[predictor]])

    # Train-test split
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_st

    # Train decision tree
    dectree = DecisionTreeClassifier(max_depth=3)
    dectree.fit(X_train, y_train)

    # Predict
    y_test_pred = dectree.predict(X_test)

    # Accuracy and Confusion Matrix
    acc = accuracy_score(y_test, y_test_pred)
    cm = confusion_matrix(y_test, y_test_pred)

    # Display results
    print(f"Accuracy: {acc:.4f}")
    print("Confusion Matrix:")
    print(cm)

    # Plot tree
    plt.figure(figsize=(8, 5))
    plot_tree(dectree, filled=True, rounded=True,
              feature_names=[predictor],
              class_names=["Non-Obese", "Obese"])
    plt.title(f"Decision Tree: {predictor}")
    plt.show()
```

4. Binary Trees – Individual Factors

```
=== Decision Tree for: family_ob_hist ===
Accuracy: 0.7481
Confusion Matrix:
[[104 130]
 [  3 291]]
```

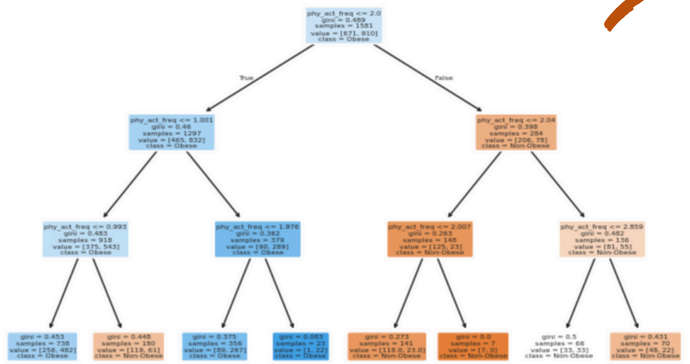
Decision Tree: family_ob_hist



family_ob_hist: HIGHEST ACCURACY

```
=== Decision Tree for: phy_act_freq ===
Accuracy: 0.7045
Confusion Matrix:
[[116 118]
 [ 38 256]]
```

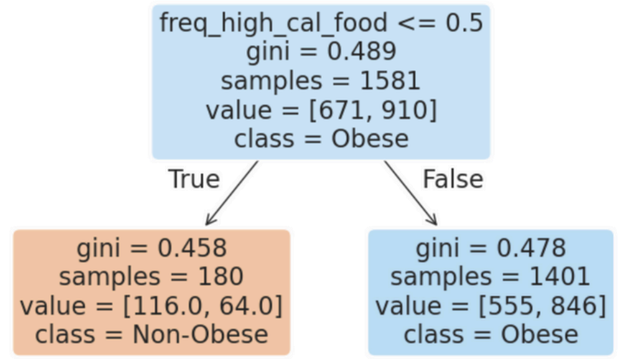
Decision Tree: phy_act_freq



phy_act_freq: moderate accuracy

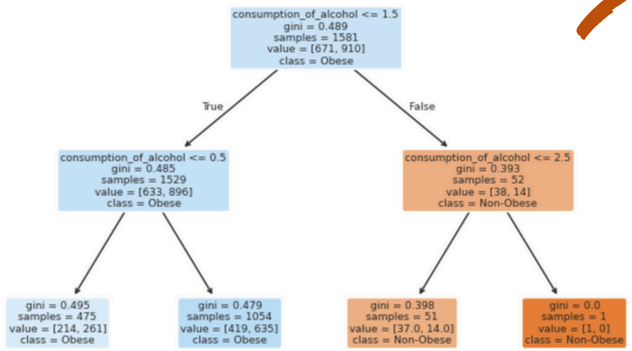
```
=== Decision Tree for: freq_high_cal_food ===
Accuracy: 0.6231
Confusion Matrix:
[[ 50 184]
 [ 15 279]]
```

Decision Tree: freq_high_cal_food




```
=== Decision Tree for: consumption_of_alcohol ===
Accuracy: 0.5511
Confusion Matrix:
[[  8 226]
 [ 11 283]]
```

Decision Tree: consumption_of_alcohol



consumption_of_alcohhol: LOWEST ACCURACY

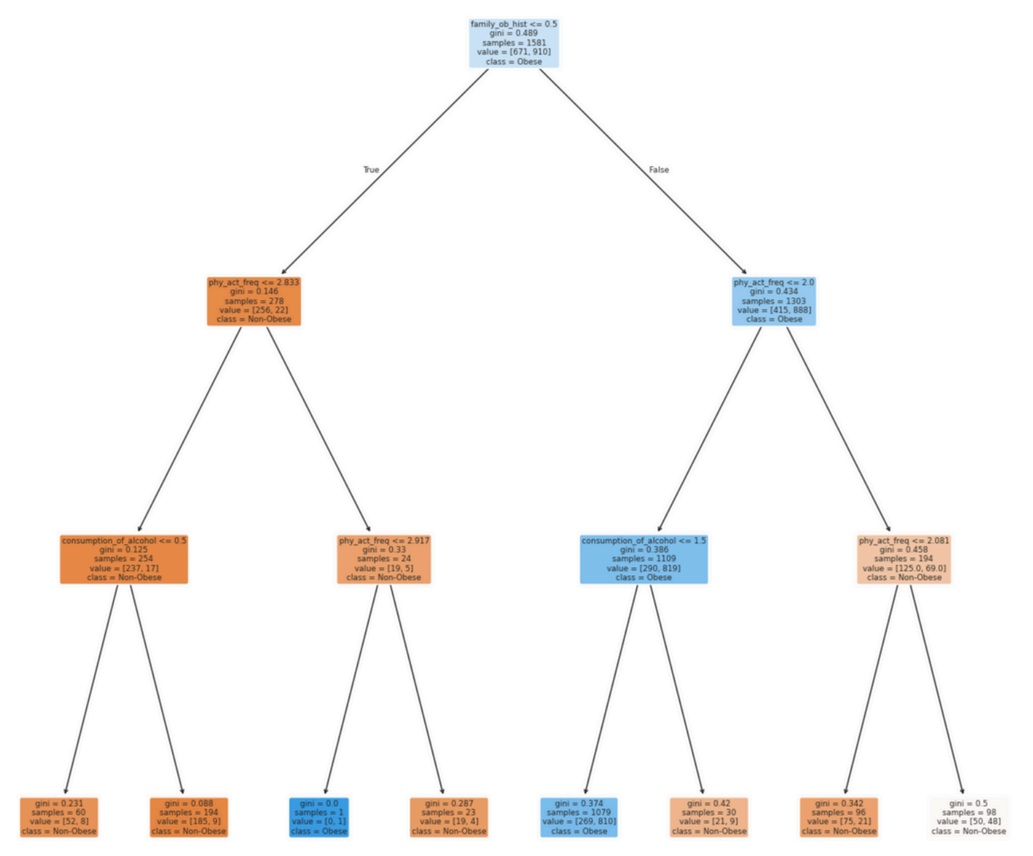
4. Binary Trees – Individual Factors

Conclusion:

1. freq_high_cal_food and consumption_of_alcohol datasets present challenges in classification
2. need more comprehensive data to improve predictive performance

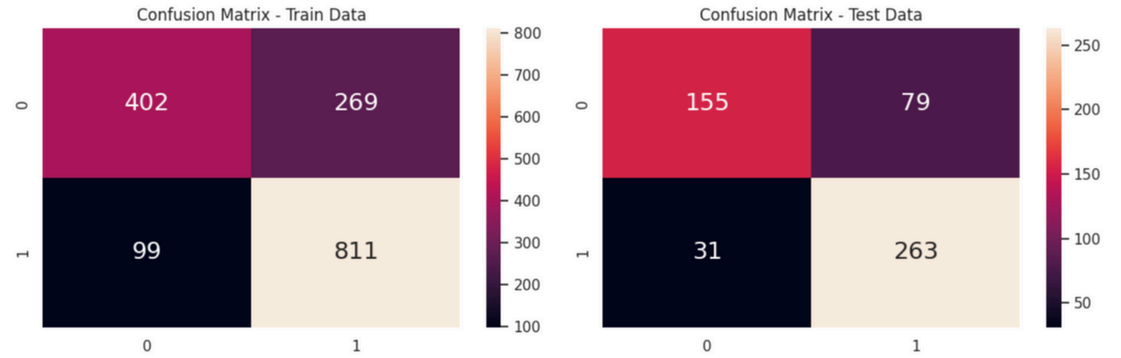
So, let's put everything together!

4. Model 1 – Binary Tree Decision Classification




```
# Check the Goodness of Fit (on Train Data)
print("Goodness of Fit of Model \tTrain Dataset")
print("Classification Accuracy \t:", dectree.score(X_train, y_train))
print()

# Check the Goodness of Fit (on Test Data)
print("Goodness of Fit of Model \tTest Dataset")
print("Classification Accuracy \t:", dectree.score(X_test, y_test))
print()

# Plot the Confusion Matrix for Train and Test
f, axes = plt.subplots(1, 2, figsize=(12, 4))
sb.heatmap(confusion_matrix(y_train, y_train_pred),
           annot = True, fmt=".0f", annot_kws={"size": 18}, ax = axes[0])
axes[0].set_title('Confusion Matrix - Train Data')
sb.heatmap(confusion_matrix(y_test, y_test_pred),
           annot = True, fmt=".0f", annot_kws={"size": 18}, ax = axes[1])
axes[1].set_title('Confusion Matrix - Test Data')
plt.tight_layout()
```

```
Goodness of Fit of Model        Train Dataset
Classification Accuracy         : 0.767235926628716

Goodness of Fit of Model        Test Dataset
Classification Accuracy         : 0.7916666666666666
```



4. Model 1 – Binary Tree Decision Classification

Training Set

TPR: 0.891

FPR: 0.400

TNR: 0.599

FNR: 0.109

Test Set

TPR: 0.895

FPR: 0.338

TNR: 0.660

FNR: 0.105

TPR = How many real obesity cases we catch

FPR = How many non-obese mistakenly flagged

Training Set :
✅ **TPR: 89.1% — Caught most obesity cases!**
⚠️ **FPR: 40% — Some misclassifications**

Test Set :
✅ **TPR: 89.5% — Caught most obesity cases!**
⚠️ **FPR: 33.8% — Slightly better than training**

4. Model 1 – Binary Tree Decision Classification

Justifying why our FPR rate is okay

False Negatives = 🚨 Danger Zone

- Missing someone who is obese = BIG risk.
- They miss interventions, stay in the dark, and health issues can pile up.
- → Our model keeps this low

Playing it Safe

- Favouring sensitivity over specificity
- Reducing FPR too much would make us miss real cases. No thanks.

False Positives Aren't All Bad!

- Some are borderline – they could use the nudge.
- Casting a wide net helps more people.

🎉 We ring the alarm a bit early — but it's to save lives.

4. Model 1 – Binary Tree Decision Classification

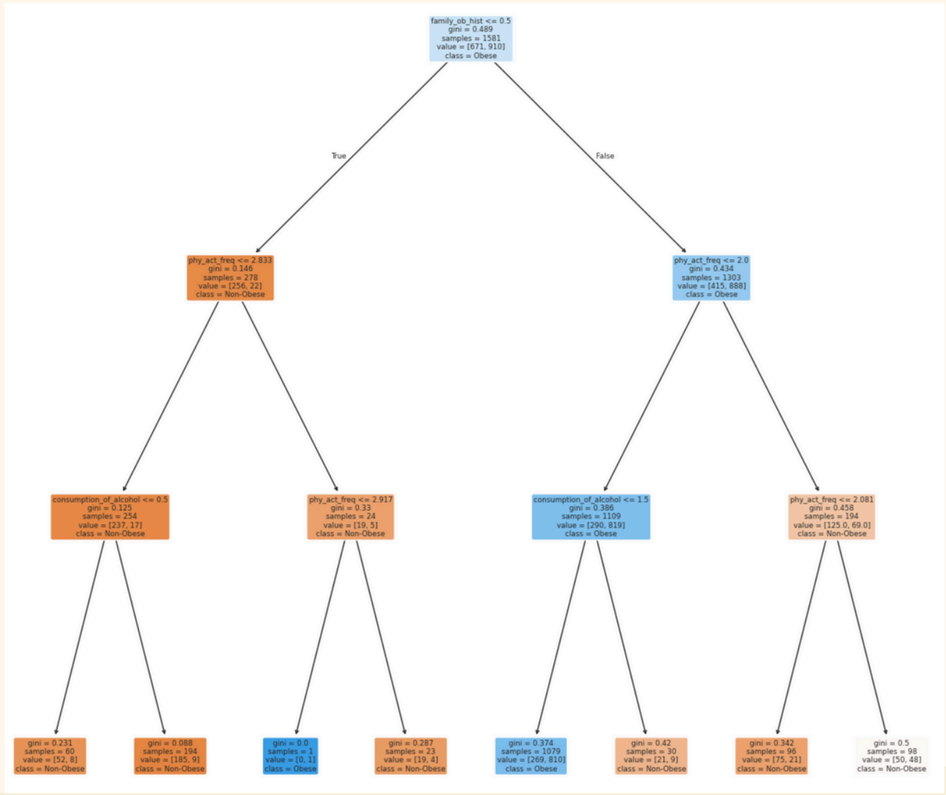
So, what's going on?

🧠 Top = Family History
This split makes the biggest impact — the model starts here every time.
Gini = 0.489 → Decent mix!

🏃 Next, Physical Activity
These splits lead to very confident results (Gini as low as 0.088)

🍃 Final Leaf Nodes: Pretty Clean!
Most paths lead to clear outcomes
Only one small group (98) is unsure (Gini = 0.5).

🎯 Big Takeaways:
Family history + physical activity = best factors
Alcohol plays a minor role
High-calorie food is not that useful here! ❌
The tree is short (depth = 3), clear, and not overfit — perfect balance between accuracy and explainability.



5. Model 2 – Random Forest Classification

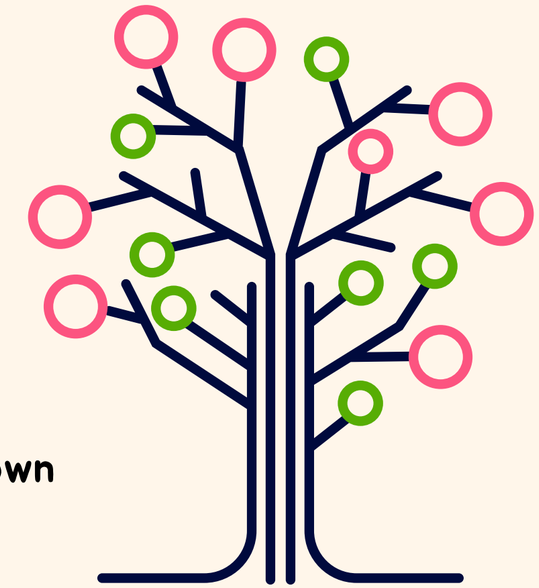
- We previously used Decision Tree Classification, which tends to overfit on training data.
- Moreover, Decision trees can give misleading importance rankings due to their greedy nature, while random forests provide more reliable and consistent feature importance estimates.
- This second model predicts categorical outcomes (Obese or Non-Obese) by building a ‘forest’ of multiple decision trees and combines their results, improving generalisation and reducing overfitting.
- Classification Accuracy of 0.777, similar to that of Decision Trees

5. Model 2 – Random Forest Classification

How it works?

1. Take Random Samples of Data
   - Instead of using the full dataset, the algorithm creates many smaller datasets by randomly picking data points with replacement (some can repeat).

2. Build Many Decision Trees
   - Each mini dataset is used to train a different decision tree.
   - At each split in the tree, it only looks at a random selection of features, not all of them.
   - This makes each tree slightly different.

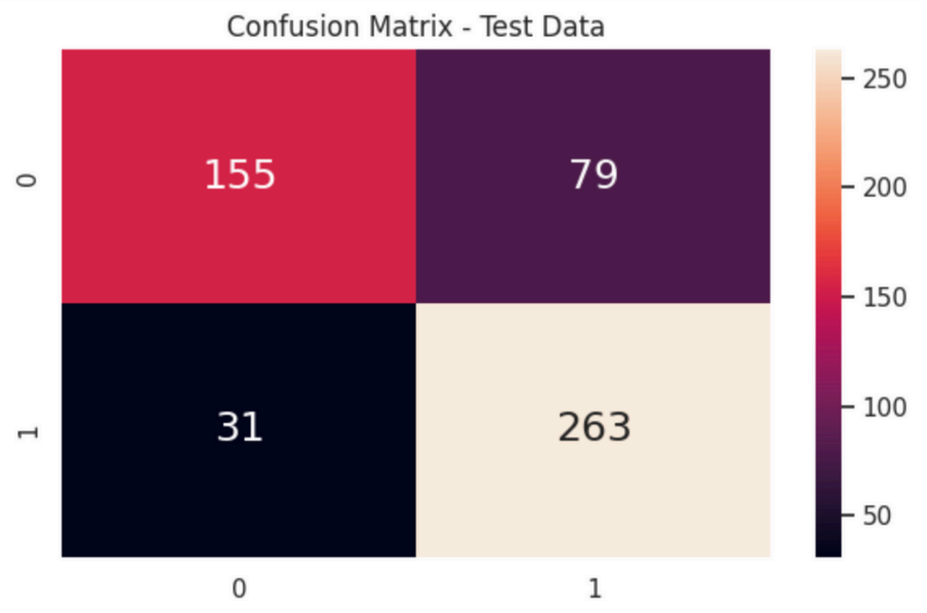


3. Let the Trees Vote
   - When you give the Random Forest a new input (like a person's data), each tree makes its own prediction (e.g., "Obese" or "Not Obese").

4. Majority Wins
   - The final prediction is based on what most trees say.
   - If most trees say "Obese", that's the prediction!

5. Model 2 – Random Forest Classification

Binary Decision Tree Classification

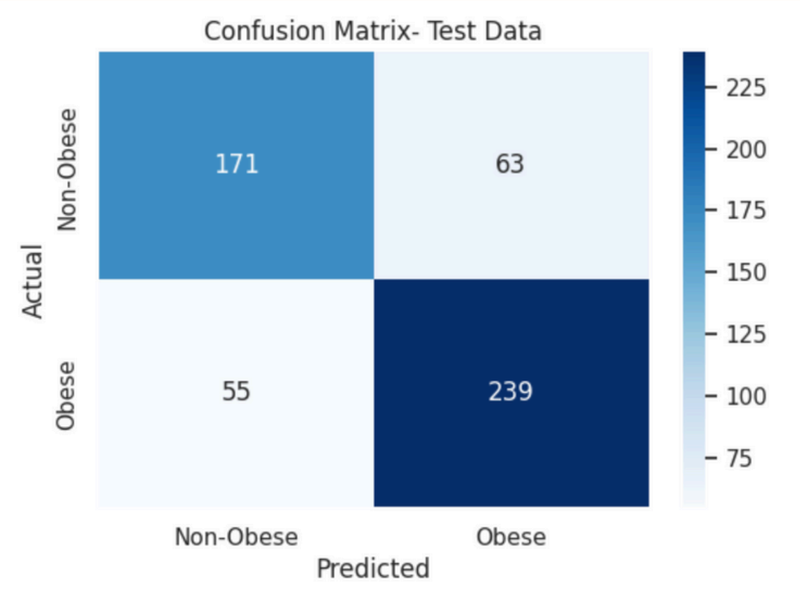


FPR= 0.338 TPR = 0.895

TNR = 0.66 FNR = 0.105

Classification Accuracy = 0.792

Random Forest Classification



FPR= 0.270 TPR = 0.813

TNR = 0.731 FNR = 0.187

Classification Accuracy = 0.777

GridSearch Cross Validation

- GridSearchCV is a method to find the best combination of hyperparameters for a model
- For Random Forest, these hyperparameters may include:

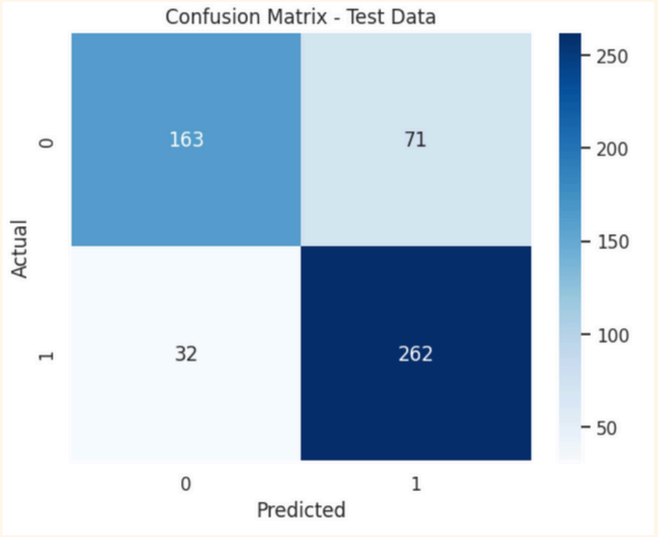
✔ n_estimators: number of trees
✔ max_depth: how deep each tree can go
✔ max_features: number of features to consider at each split
✔ min_samples_split: minimum samples needed to split a node

- GridSearchCV incorporates cross-validation (5-fold)
- Cross-validation score of 0.770 indicates that the model generalizes well to new data
- Model is trained and validated on different data splits, reducing the risk of overfitting and providing a more reliable estimate of how well the model will perform on unseen data.

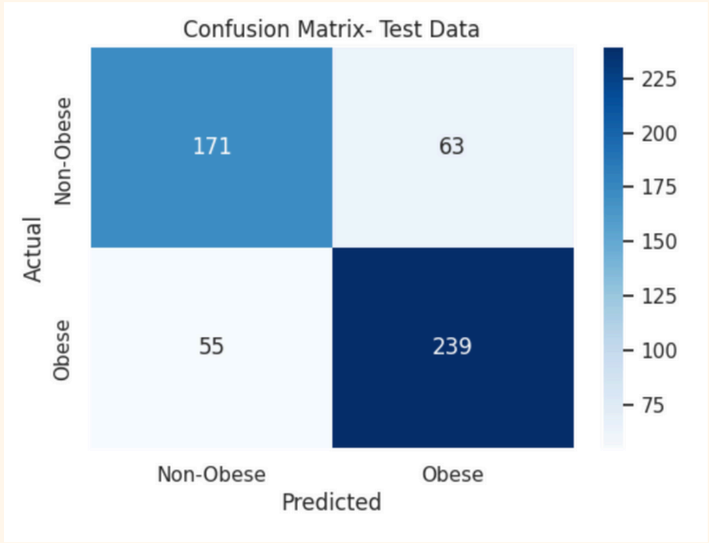


5. Model 2 – Random Forest Classification

Random Forest Classification (after GSCV)



Random Forest Classification



Slight improvement in model after GSCV

FPR= 0.303 TPR = 0.891

TNR = 0.697 FNR = 0.109

Classification Accuracy = 0.805

FPR= 0.270 TPR = 0.813

TNR = 0.731 FNR = 0.187

Classification Accuracy = 0.777

5. Model 2 – Random Forest Classification

Random Forest Classification (after GSCV)

	Importance
phy_act_freq	0.506065
family_ob_hist	0.383939
consumption_of_alcohol	0.076465
freq_high_cal_food	0.033531

Random Forest Classification

family_ob_hist	0.273292
freq_high_cal_food	0.025129
phy_act_freq	0.651216
consumption_of_alcohol	0.050363

✓ Same Ranking of Factors:

1. phy_act_freq
2. family_ob_hist
3. consumption_of_alcohol
4. freq_high_cal_food

6. Data-Driven Insights

Both models gave similar rankings, with the top 2 rankings swapped:

1. Physical Activity Frequency
2. Family Obesity History
3. Consumption of Alcohol
4. Frequent consumption of high caloric food

6. Data-Driven Insights

- Higher **physical activity frequency** lowers obesity risk most significantly.
- A **family history of obesity** also has a strong impact on obesity risk.
- **Lower alcohol consumption** reduces obesity risk, but only to a small extent.
- **Less frequent consumption of high-calorie foods** lowers obesity risk to the smallest extent.

6. Data-Driven Insights

Promotion of Physical Activity is Most Effective at Reducing Obesity

1. Develop more free or low-cost access to exercise spaces
2. Encourage daily activity through urban design and workplace wellness programs.

Family History should be Addressed

1. Early screening and targeted interventions for individuals with obesity family history
2. More promotion of genetic literacy by healthcare professionals, health agencies and schools

6. Data-Driven Insights

Reducing Alcohol can be Part of a Comprehensive Obesity Strategy

1. Health Warnings on Alcohol Beverage Labels
2. Stricter Ads Regulation on Alcohol Promotion

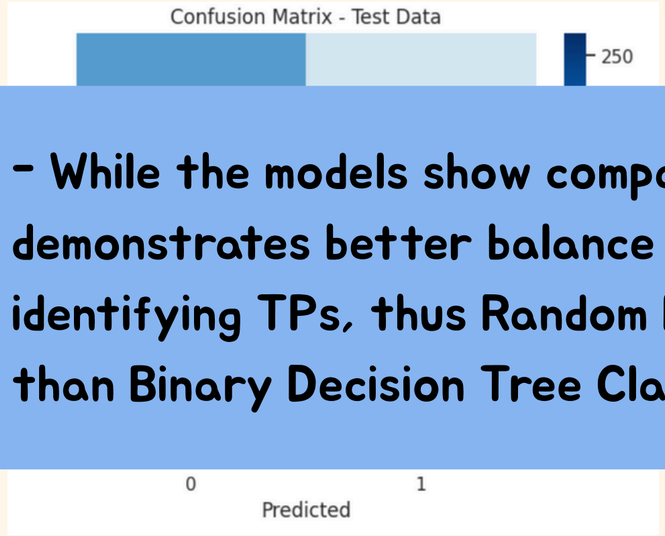
Efforts should not over-prioritise food restriction alone but promote balanced eating patterns

1. Educational campaigns suggest portion control, mindful eating, and reducing emotional eating triggers.

7. Conclusion

Evaluation of models:

Random Forest Classification (after GSCV)



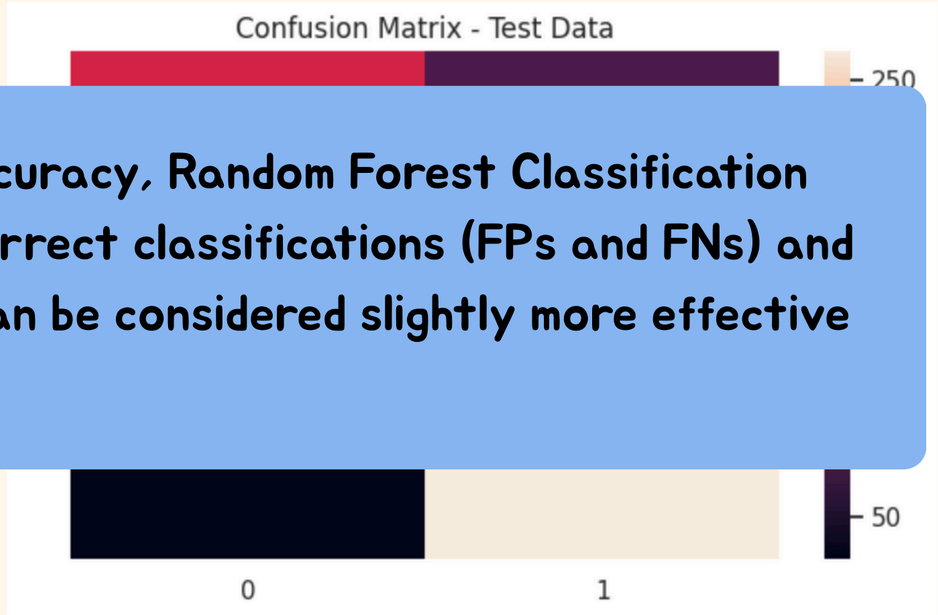
Best

FPR= 0.303 TPR = 0.891

TNR = 0.697 FNR = 0.109

Classification Accuracy = 0.805

Binary Decision Tree Classification



FPR= 0.338 TPR = 0.895

TNR = 0.66 FNR = 0.105

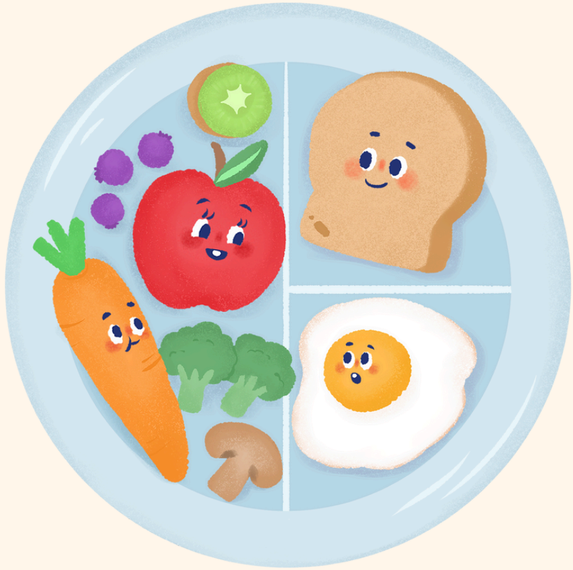
Classification Accuracy = 0.792

- While the models show comparable classification accuracy, Random Forest Classification demonstrates better balance between minimising incorrect classifications (FPs and FNs) and identifying TPs, thus Random Forest Classification can be considered slightly more effective than Binary Decision Tree Classification.

Obesity is a highly concerning health issue that poses serious consequences on an individual, societal and global level.

Let's use what we've learned to promote healthier habits and work together to prevent Obesity!

Thank You !



References

https://www.who.int/news-room/fact-sheets/detail/obesity-and-overweight
https://www.nuhs.edu.sg/patient-care/find-a-condition/obesity