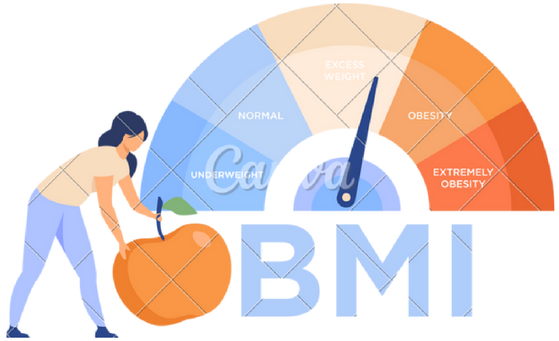
SC1015 MINI PROJECT
OBESITY & CARDIOVASCULAR RISK

Group 4

Bui Khanh Hung - U2322190J
Bui Gia Nhat Minh - U2320021L
Tran Son Viet - U2322219G

PROBLEM FORMULATION

- High obesity rate across the world
- According to the Sydney Morning Herald, more people are obese than are starving
- Not just related to food consumption, lack of exercise, but various factors

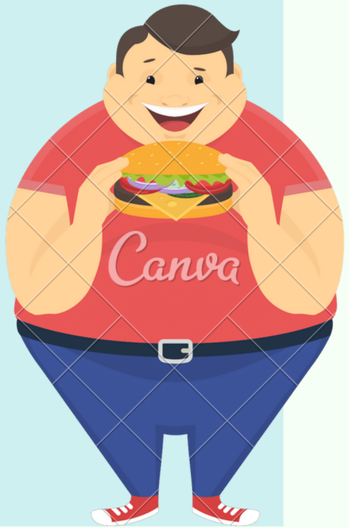


PRACTICAL MOTIVATION

- Level of obesity an individual might have a high possibility to fall into

- Aspects of people's life have the big impact on the risk of being obese and suffering from CVD

→ Aim: educational website & online webminar to improve awareness



DATA COLLECTION

The data was collected from the dataset “Obesity or CVD risk” on Kaggle

DATA CLEANING & PREPARATION

- Check and remove NULL values (if any)
- Check and remove Duplicate values (if any)
- Check types of variables (categorical or numerical)
- One-hot encoding the categorical variables for ML models

DATA CLEANING & PREPARATION

```
In [10]: obese.isnull().sum()

Out[10]: id                                0
         Gender                            0
         Age                               0
         Height                            0
         Weight                            0
         family_history_with_overweight    0
         FAVC                              0
         FCVC                              0
         NCP                               0
         CAEC                              0
         SMOKE                             0
         CH2O                              0
         SCC                               0
         FAF                               0
         TUE                               0
         CALC                              0
         MTRANS                            0
         NObeyesdad                        0
         dtype: int64
```

The data set is clean without any NULL values

DATA CLEANING & PREPARATION

```
In [10]: obese = obese.drop_duplicates()

         obese.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20758 entries, 0 to 20757
Data columns (total 18 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   id                              20758 non-null  int64
 1   Gender                          20758 non-null  object
 2   Age                             20758 non-null  float64
 3   Height                          20758 non-null  float64
 4   Weight                          20758 non-null  float64
 5   family_history_with_overweight  20758 non-null  object
 6   FAVC                            20758 non-null  object
 7   FCVC                            20758 non-null  float64
 8   NCP                             20758 non-null  float64
 9   CAEC                            20758 non-null  object
 10  SMOKE                           20758 non-null  object
 11  CH2O                            20758 non-null  float64
 12  SCC                             20758 non-null  object
 13  FAF                             20758 non-null  float64
 14  TUE                             20758 non-null  float64
 15  CALC                            20758 non-null  object
 16  MTRANS                          20758 non-null  object
 17  NObeyesdad                      20758 non-null  object
dtypes: float64(8), int64(1), object(9)
memory usage: 2.9+ MB
```

The data set is clean without any DUPLICATE values

DATA CLEANING & PREPARATION

```
In [8]: obese[numeric_features].nunique()

Out[8]: id        20758
        Age        1703
        Height     1833
        Weight     1979
        FCVC        934
        NCP         689
        CH2O       1506
        FAF        1360
        TUE        1297
        dtype: int64
```

All the variable with “int” or “float” data type are truly continous numerical variables

DATA CLEANING & PREPARATION

family_history_with_overweight	FAVC	FCVC	NCP	CAEC	SMOKE	CH2O	SCC	FAF	TUE	CALC	MTRANS
yes	yes	2.000000	2.983297	Sometimes	no	2.763573	no	0.000000	0.976473	Sometimes	Public_Transportation
yes	yes	2.000000	3.000000	Frequently	no	2.000000	no	1.000000	1.000000	no	Automobile
yes	yes	1.880534	1.411685	Sometimes	no	1.910378	no	0.866045	1.673584	no	Public_Transportation
yes	yes	3.000000	3.000000	Sometimes	no	1.674061	no	1.467863	0.780199	Sometimes	Public_Transportation
yes	yes	2.679664	1.971472	Sometimes	no	1.979848	no	1.967973	0.931721	Sometimes	Public_Transportation

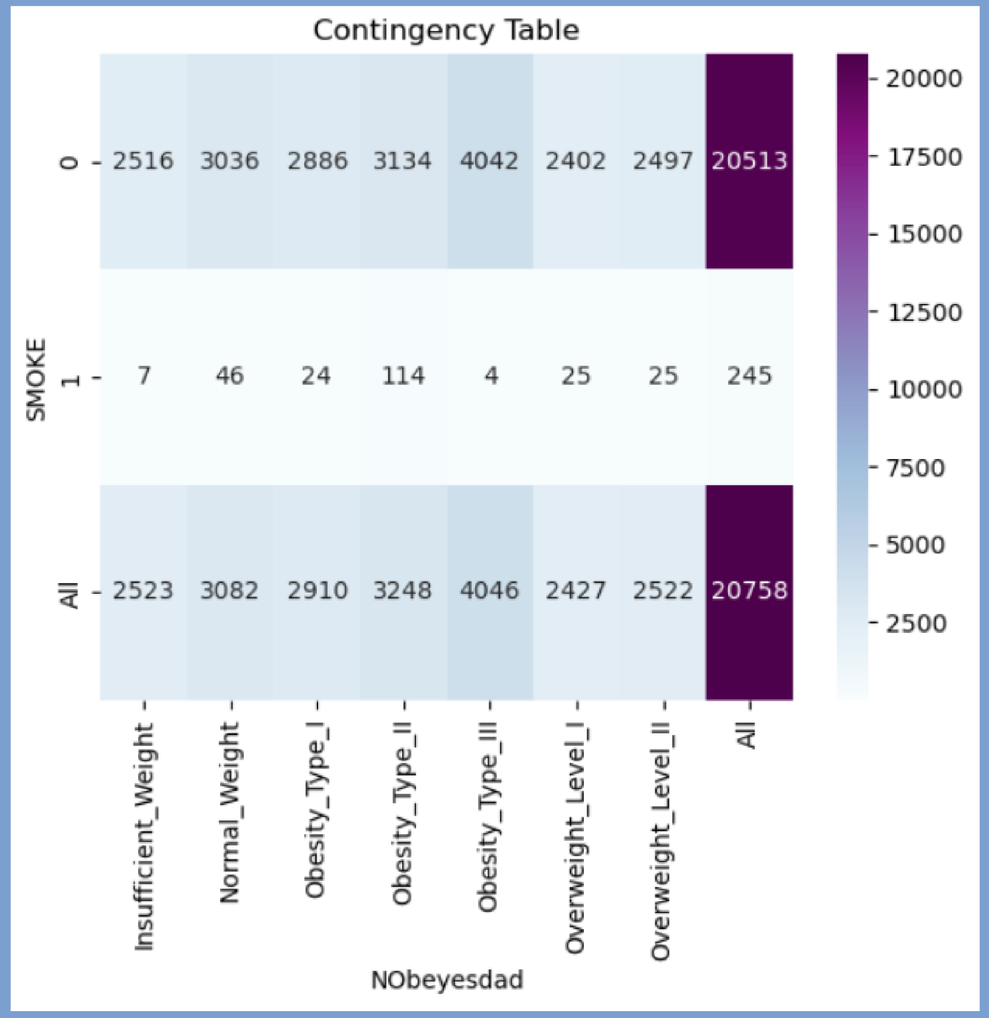
family_history_with_overweight	FAVC	FCVC	NCP	SMOKE	...	CAEC_Sometimes	CAEC_no	CALC_Frequently	CALC_Sometimes	CALC_no
1	1	2.000000	2.983297	0	...	True	False	False	True	False
1	1	2.000000	3.000000	0	...	False	False	False	False	True
1	1	1.880534	1.411685	0	...	True	False	False	False	True
1	1	3.000000	3.000000	0	...	True	False	False	True	False
1	1	2.679664	1.971472	0	...	True	False	False	True	False

The data set before and after the encoding for categorical variables

EXPLORATORY DATA ANALYSIS

EXPLORATORY DATA ANALYSIS

CATEGORICAL VARIABLES



- Many variables are extremely unbalanced (like this one), therefore can not observe directly from contingency table
- Need to use “Chi-squared” Test and “Mutual Information” Test instead

EXPLORATORY DATA ANALYSIS

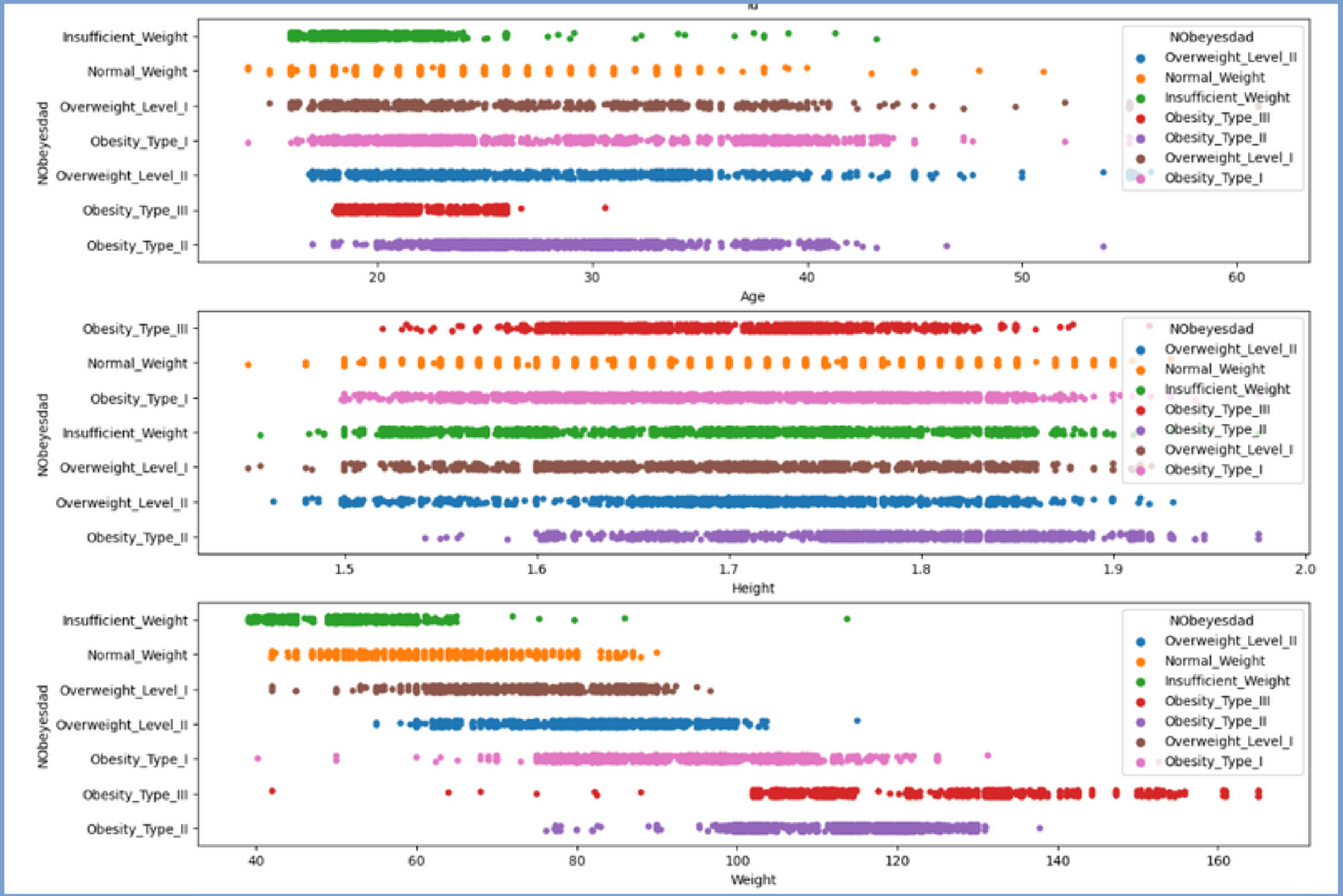
CATEGORICAL VARIABLES

```
The mutual information between Gender and NObeyesdad is: 0.26141861531874167
The p_value of the chi-squared test is:  0.0

The mutual information between family_history_with_overweight and NObeyesdad is: 0.16597330340890304
The p_value of the chi-squared test is:  0.0

 The mutual information between CAEC_Frequently and NObeyesdad is: 0.10581579732106805
 The p_value of the chi-squared test is:  0.0

 The mutual information between CAEC_Sometimes and NObeyesdad is: 0.11694753879854991
 The p_value of the chi-squared test is:  0.0

 The mutual information between CALC_Sometimes and NObeyesdad is: 0.10281643254013817
 The p_value of the chi-squared test is:  0.0

 The mutual information between CALC_no and NObeyesdad is: 0.1012318621092474
 The p_value of the chi-squared test is:  0.0
```

From mutual information index, Gender, family history with overweight, consumption of food between meals, and consumption of alcohol seem to be most closely related to obesity level

EXPLORATORY DATA ANALYSIS

NUMERICAL VARIABLES

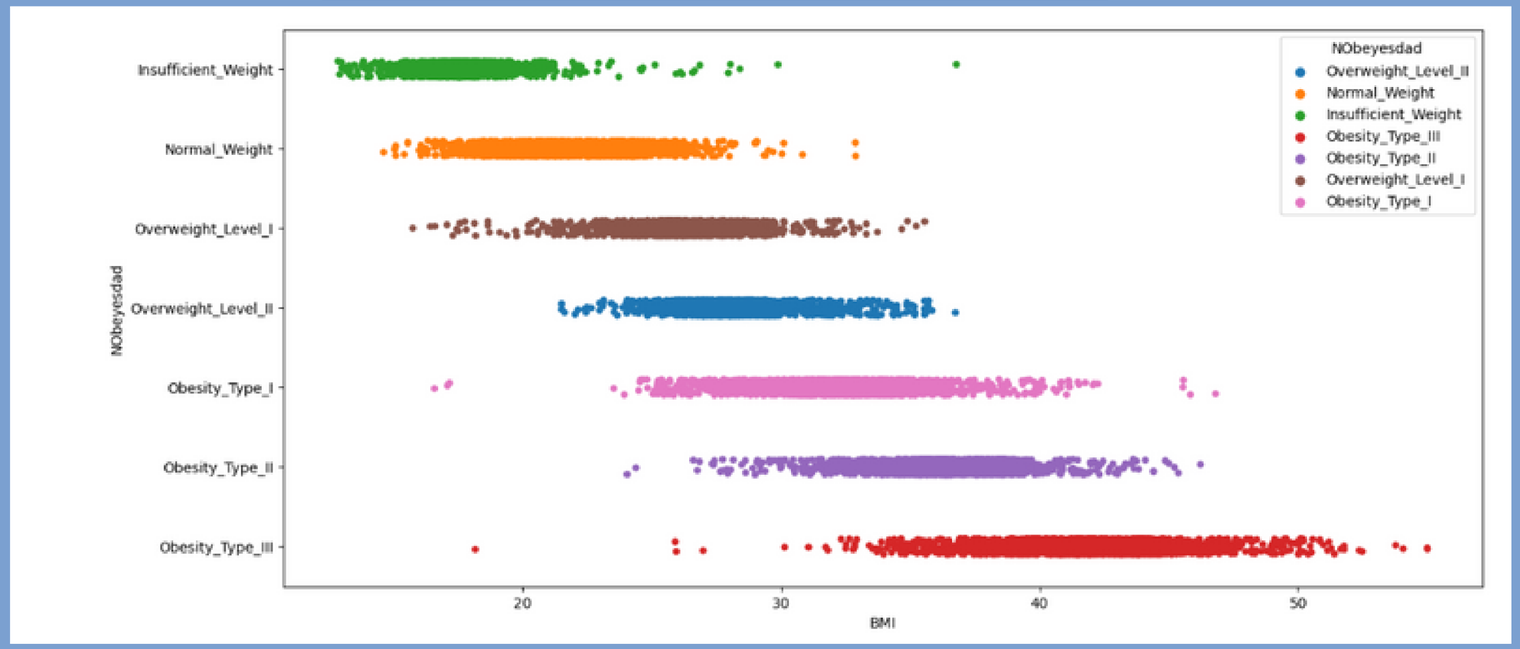


From strip plot, only Age, Height, and Weight has clear enough differences between different levels of Obesity

EXPLORATORY DATA ANALYSIS

NUMERICAL VARIABLES

$$BMI = \frac{\text{Weight in kilogram}}{\left(\text{Height in meter}\right)^2}$$



- Introduce new variable: BMI

- There is a clear relationship between them as the distribution is very distinct between the levels

MODEL

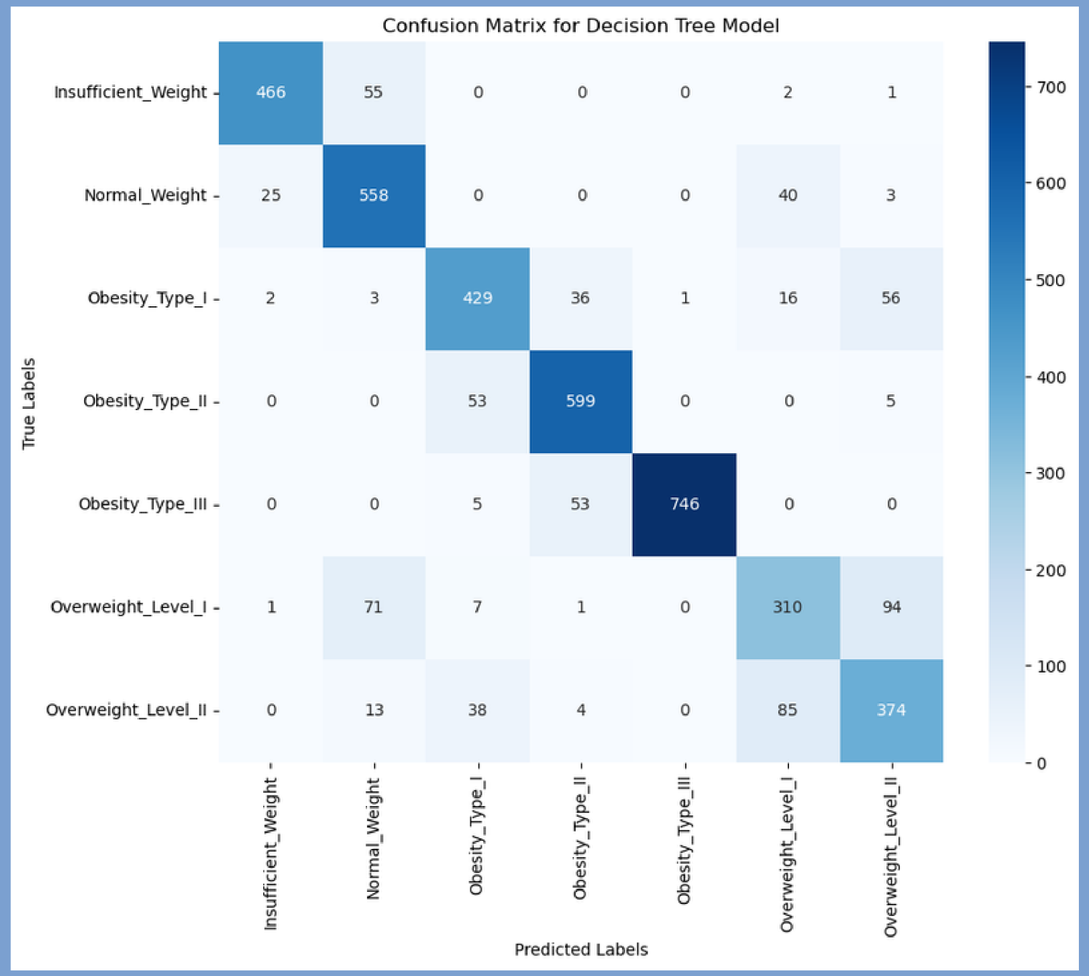
MODEL

- Both numerical and categorical data
- Handling Non-Linear Relationships

Decision Tree

MODEL

DECISION TREE



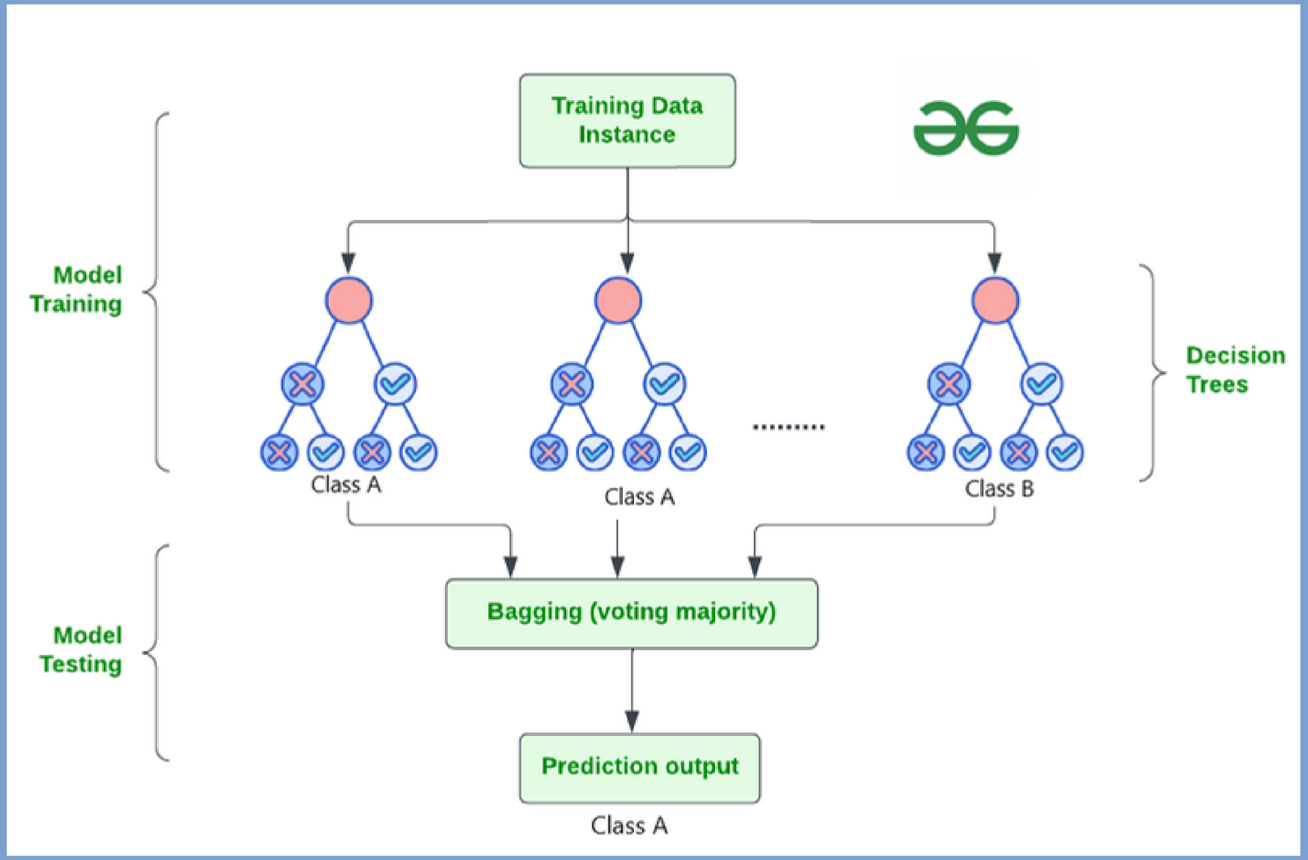
Accuracy: 0.8386319845857418

- tendency to overfit
- unstable

MODEL

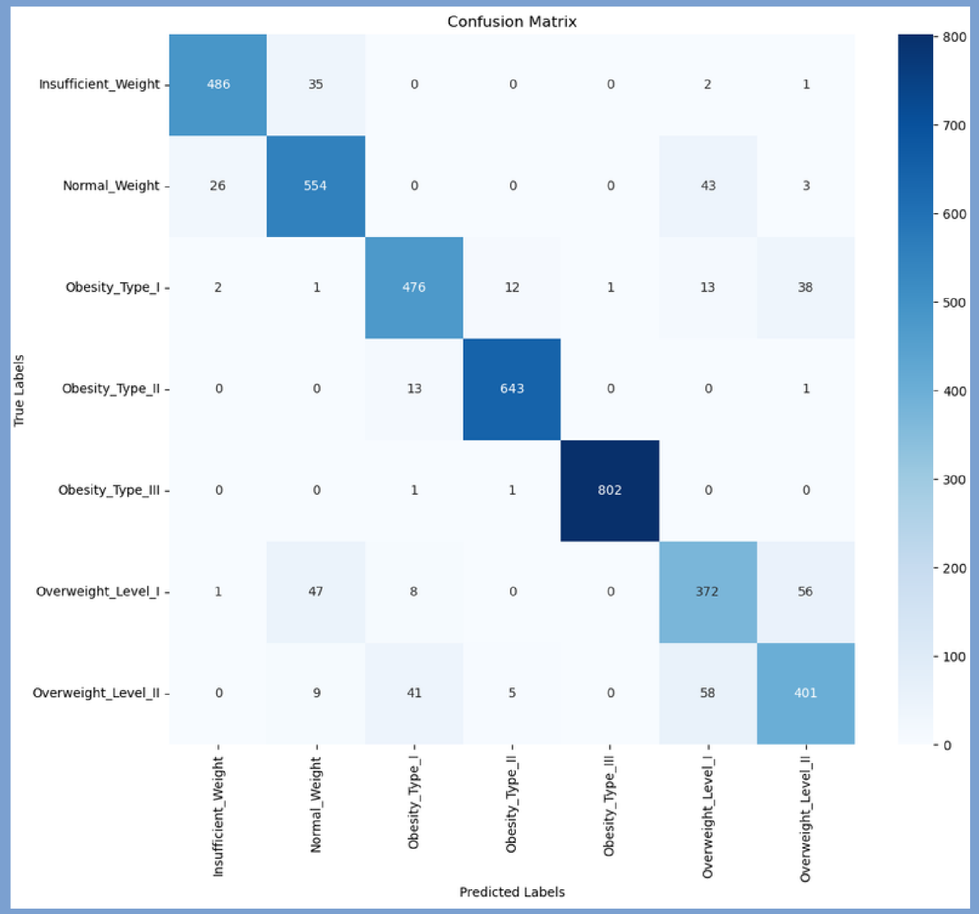
RANDOM FOREST

How it works?

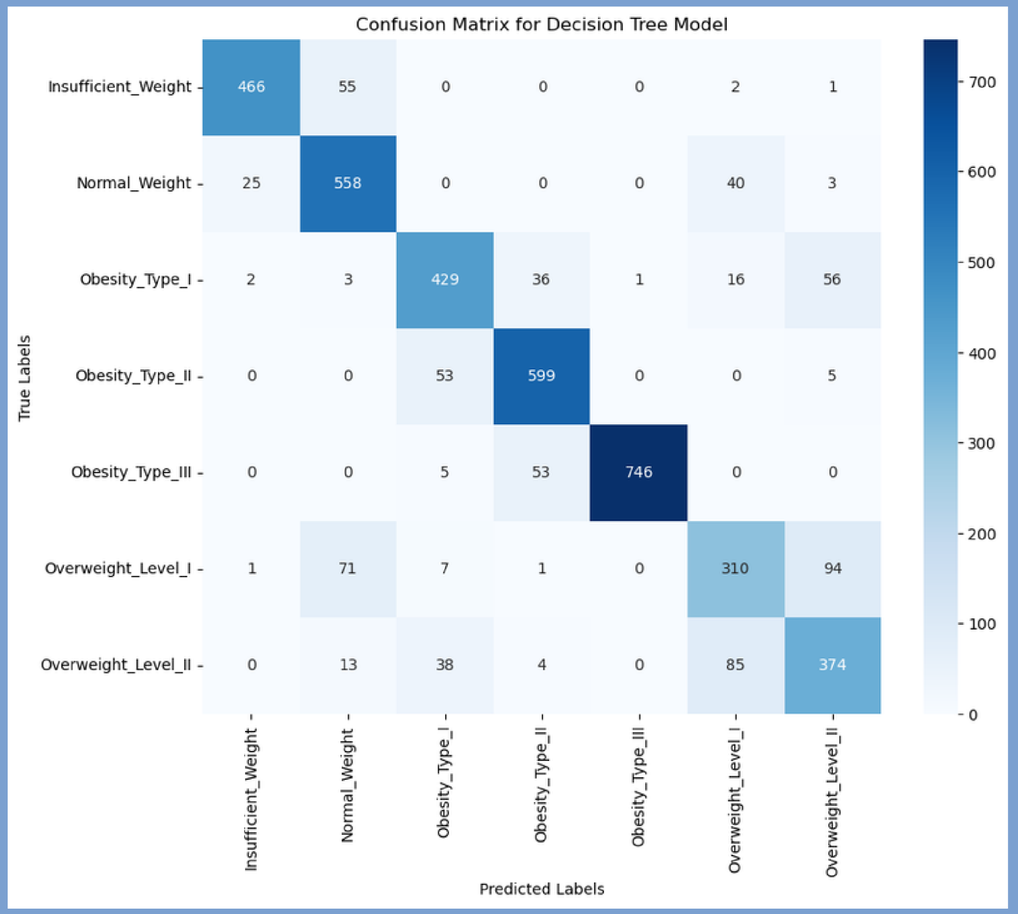


MODEL

RANDOM FOREST



0.899325626204239



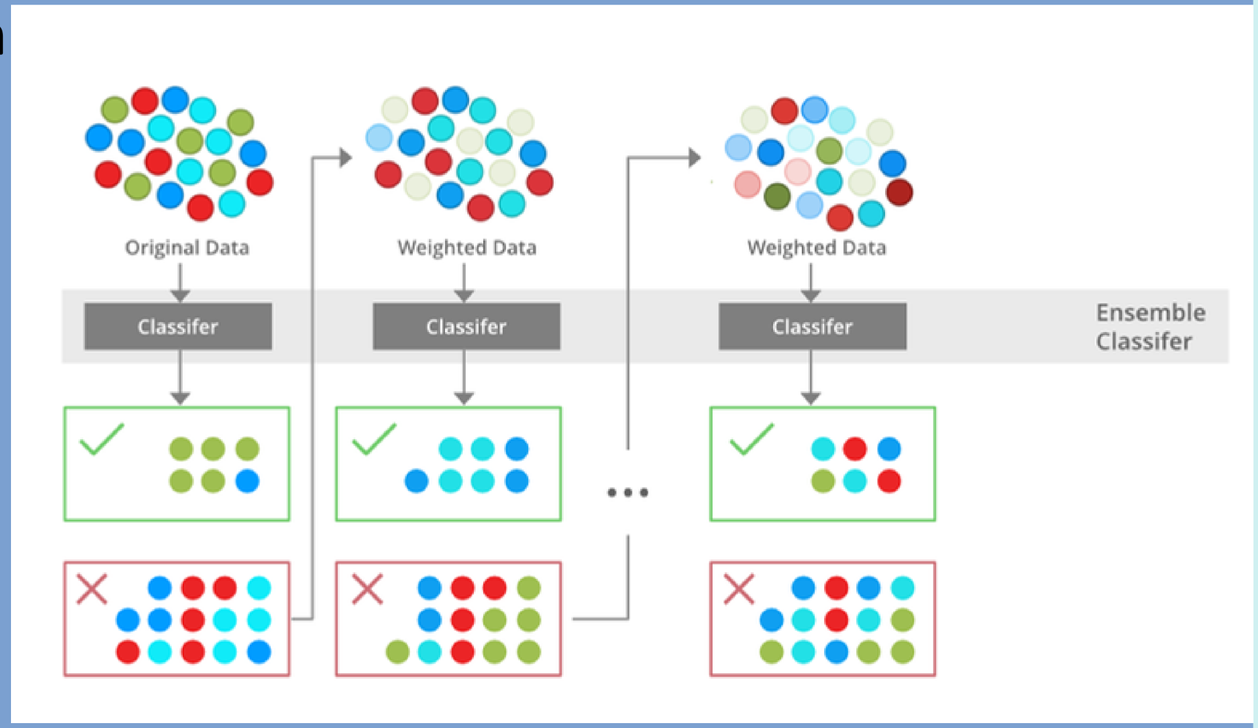
0.8386319845857418

MODEL

XGBOOST

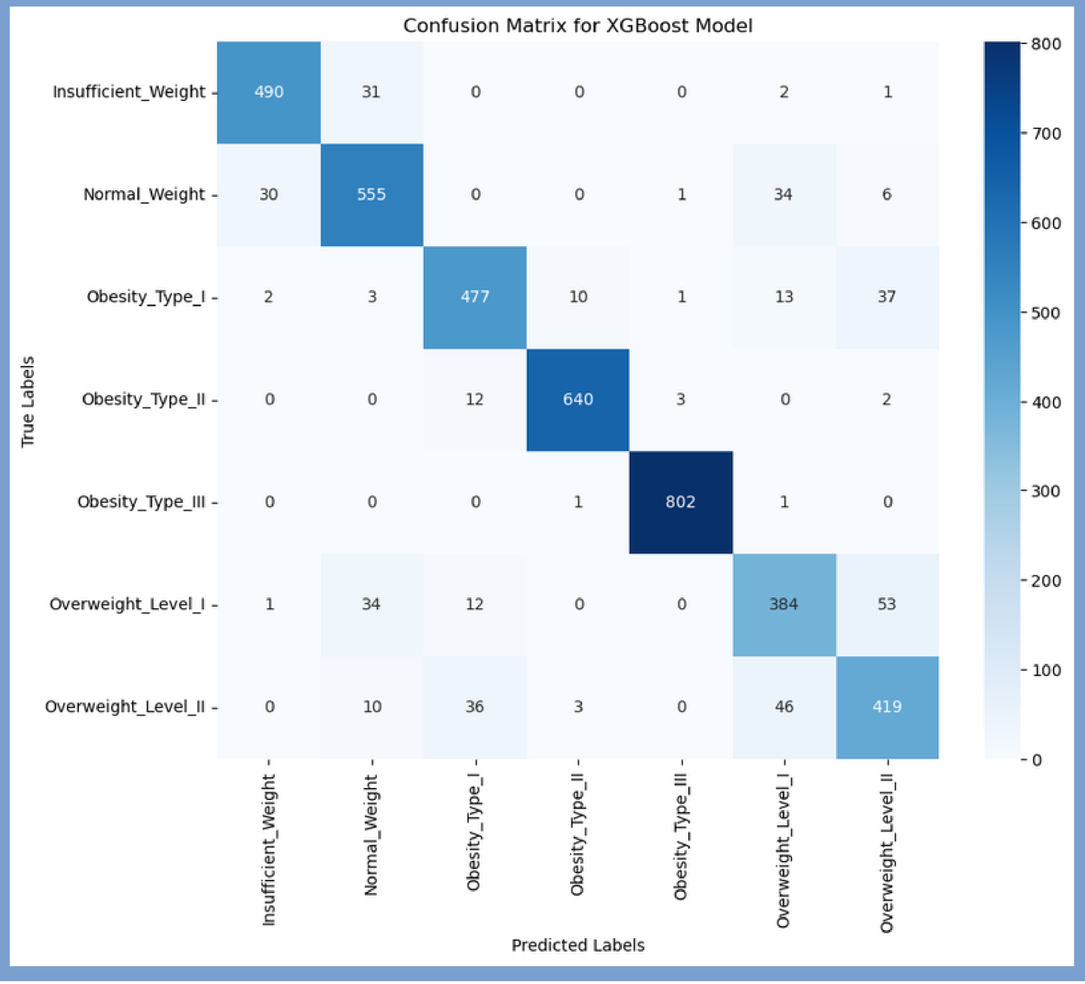
How it works?

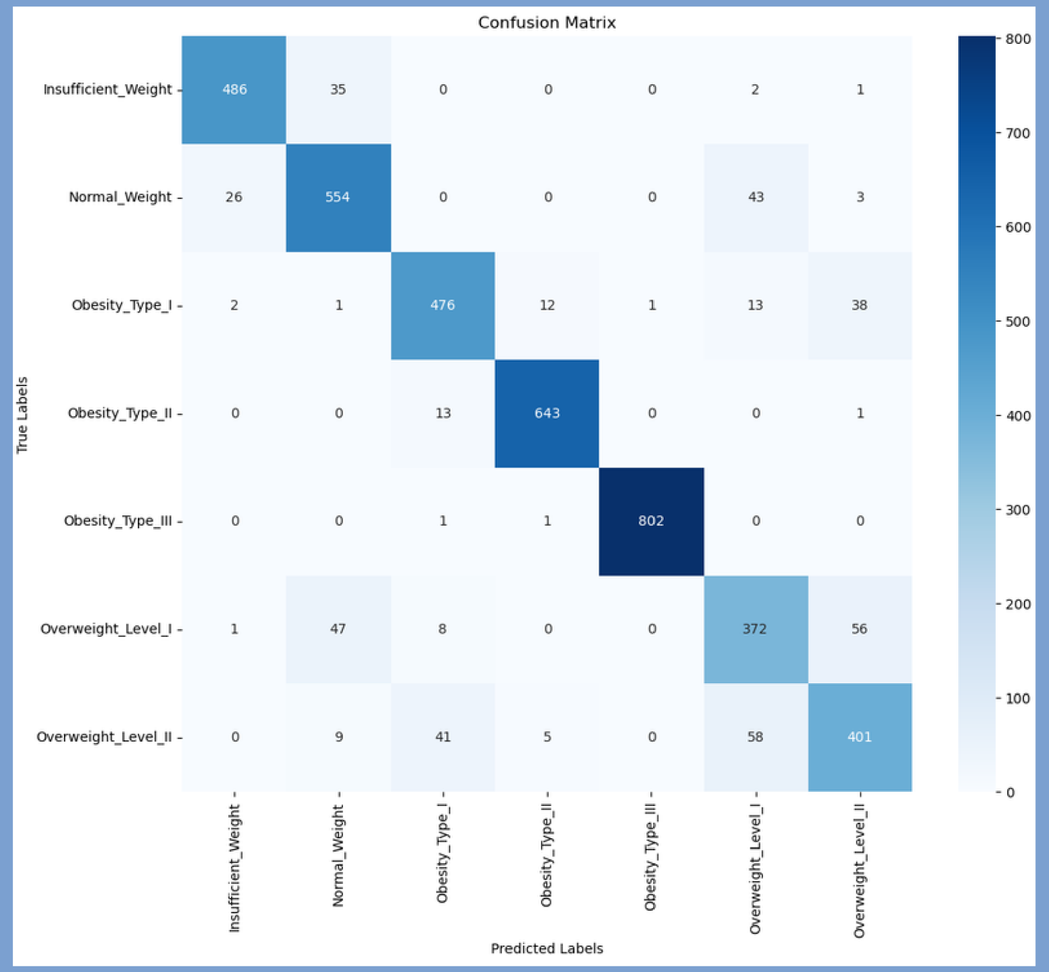
- Building an initial decision tree based on the training data.
- Calculates the errors made by this tree and assigns weights to the data points based on these errors.
- Building a new tree to correct these errors, with focus on misclassified data or data with higher errors
- This process is repeated for a specified number of tree or until a stopping criterion is met



MODEL

XGBOOST

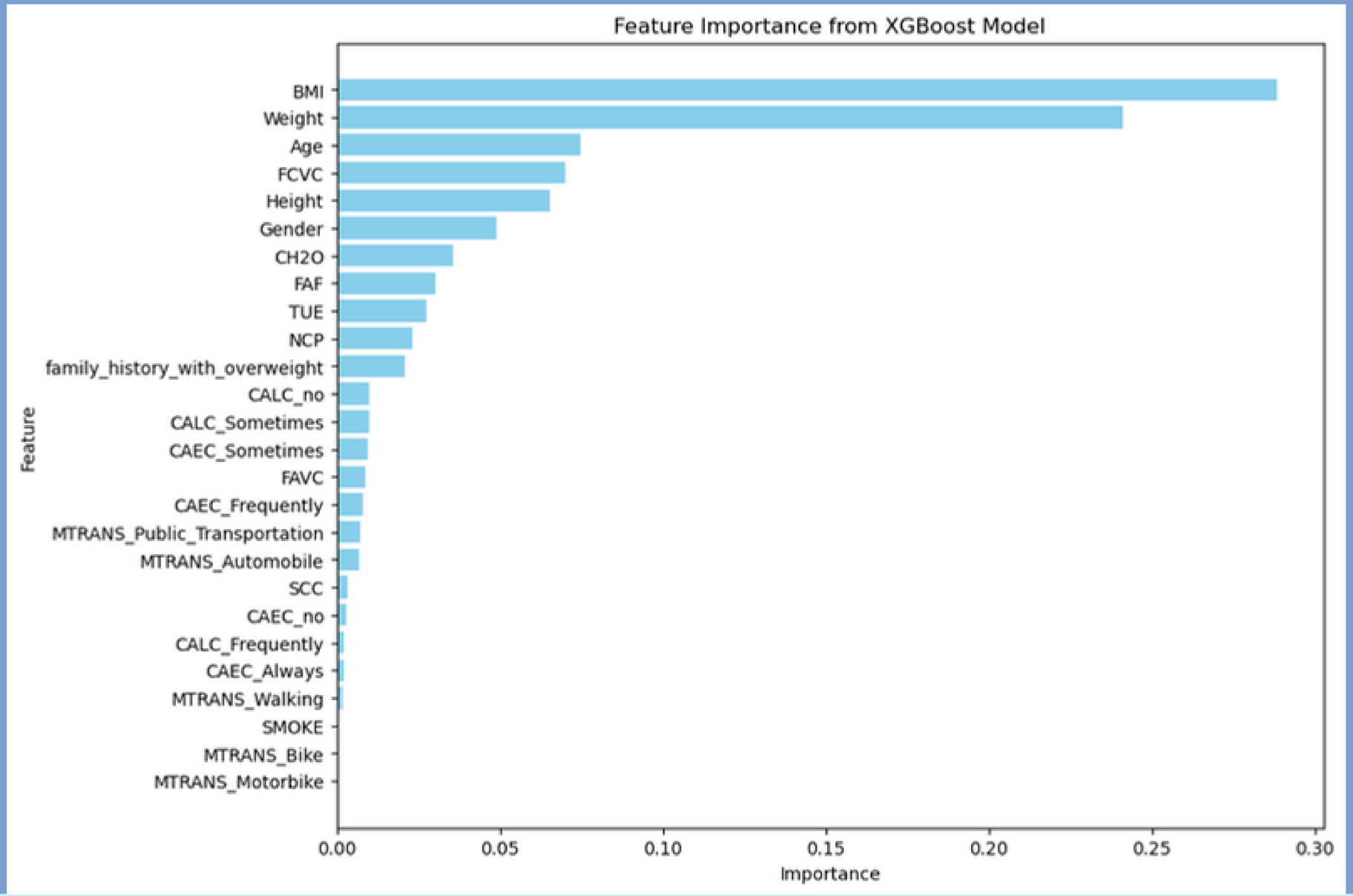




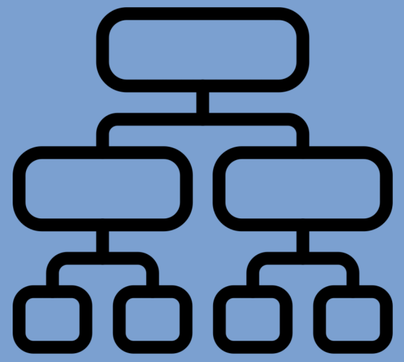
(0.9040110358960302, 0.006443857943949225,

0.899325626204239

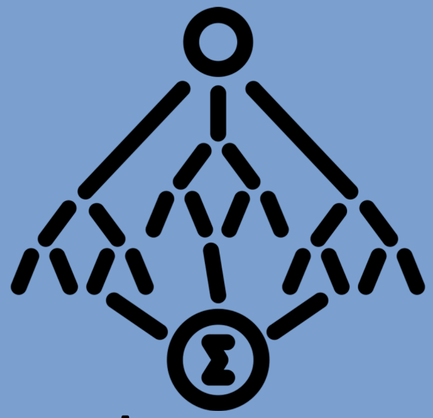
OUTCOMES


Feature Importance from XGBoost Model
BMI
Weight
Age
FCVC
Height
Gender
CH2O
FAF
TUE
NCP
family_history_with_overweight
CALC_no
CALC_Sometimes
CAEC_Sometimes
FAVC
CAEC_Frequently
MTRANS_Public_Transportation
MTRANS_Automobile
SCC
CAEC_no
CALC_Frequently
CAEC_Always
MTRANS_Walking
SMOKE
MTRANS_Bike
MTRANS_Motorbike
Feature
0.00
0.05
0.10
0.15
0.20
0.25
0.30
Importance

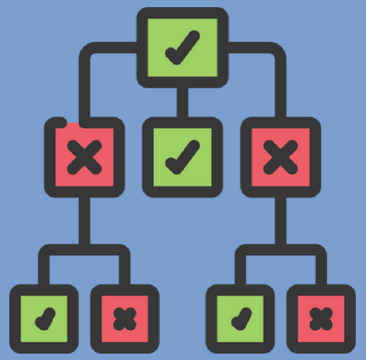
OUTCOMES



Decision tree
Accuracy: 83%



Random Forest
Accuracy: 89%



XGBoost
Accuracy: 90%

REFERENCES

XGBoost explain:
https://medium.com/@techynilesh/xgboost-algorithm-explained-in-less-than-5-minutes-b561dcc1ccee
Random Forest explain:
https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/
Dataset:
https://www.kaggle.com/competitions/playground-series-s4e2/overview
https://www.smh.com.au/national/more-fat-people-in-world-than-there-are-starving-study-finds-20060815-gdo6f2.html