Cairo University

Faculty of Engineering
Cairo University

Machine Learning Project

Supervised by: Eng. Mohamed Shawky Sabae

Team 8

Presented by:

Name	Sec.	B.N.
Ahmed Emad	1	7
Ahmed Ibrahim	1	8
Ahmed Mahmoud Hafez	1	11
Gaser Ashraf	1	23

Spring 2023

• The problem definition, motivation and evaluation metrics

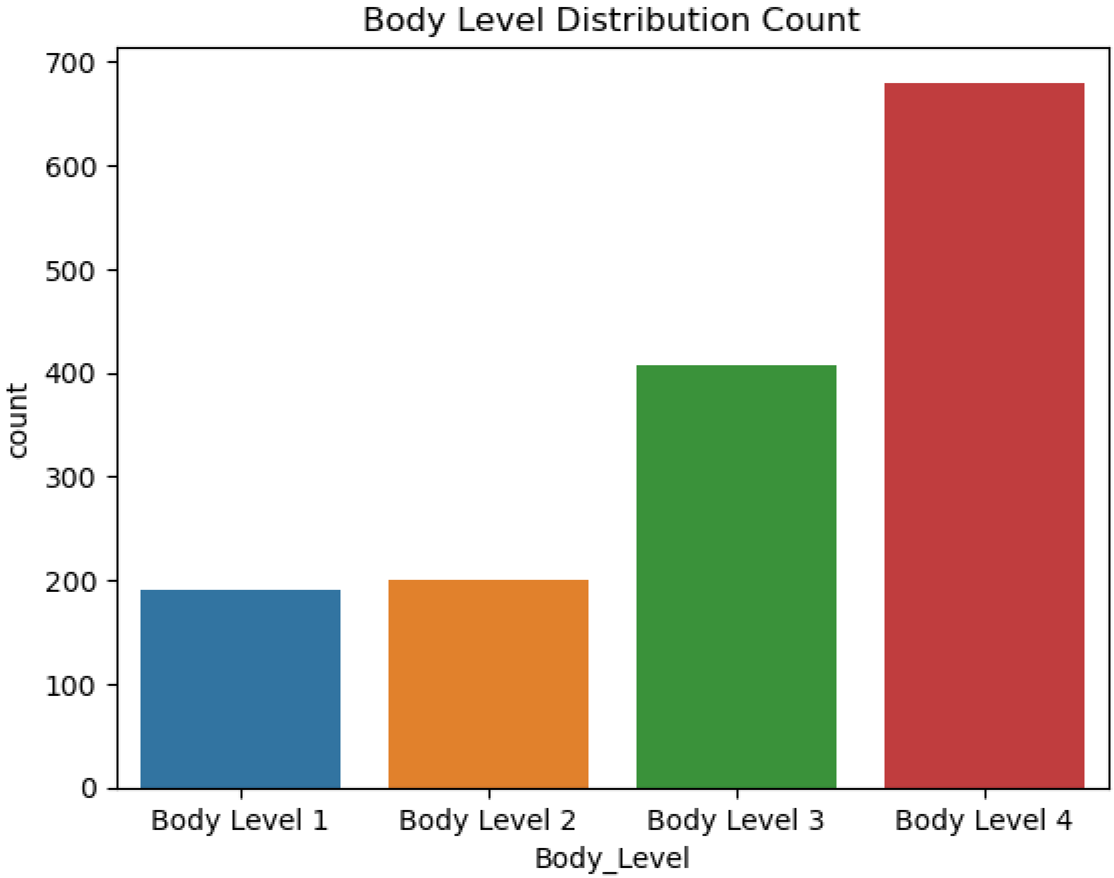
The main objective of this project is to apply machine learning concepts and algorithms to a real-world problem. The selected problem for this semester is “Body Level Classification”.

We are required to solve a classification problem for human body level based on some given attributes related to the physical, genetic and habitual conditions. The given attributes are both categorical and continuous. The human body level can be categorized into (4 levels/classes). We are given 16 attributes and 1477 data samples, where classes are not evenly distributed.

Evaluation metrics weighted F1 score.

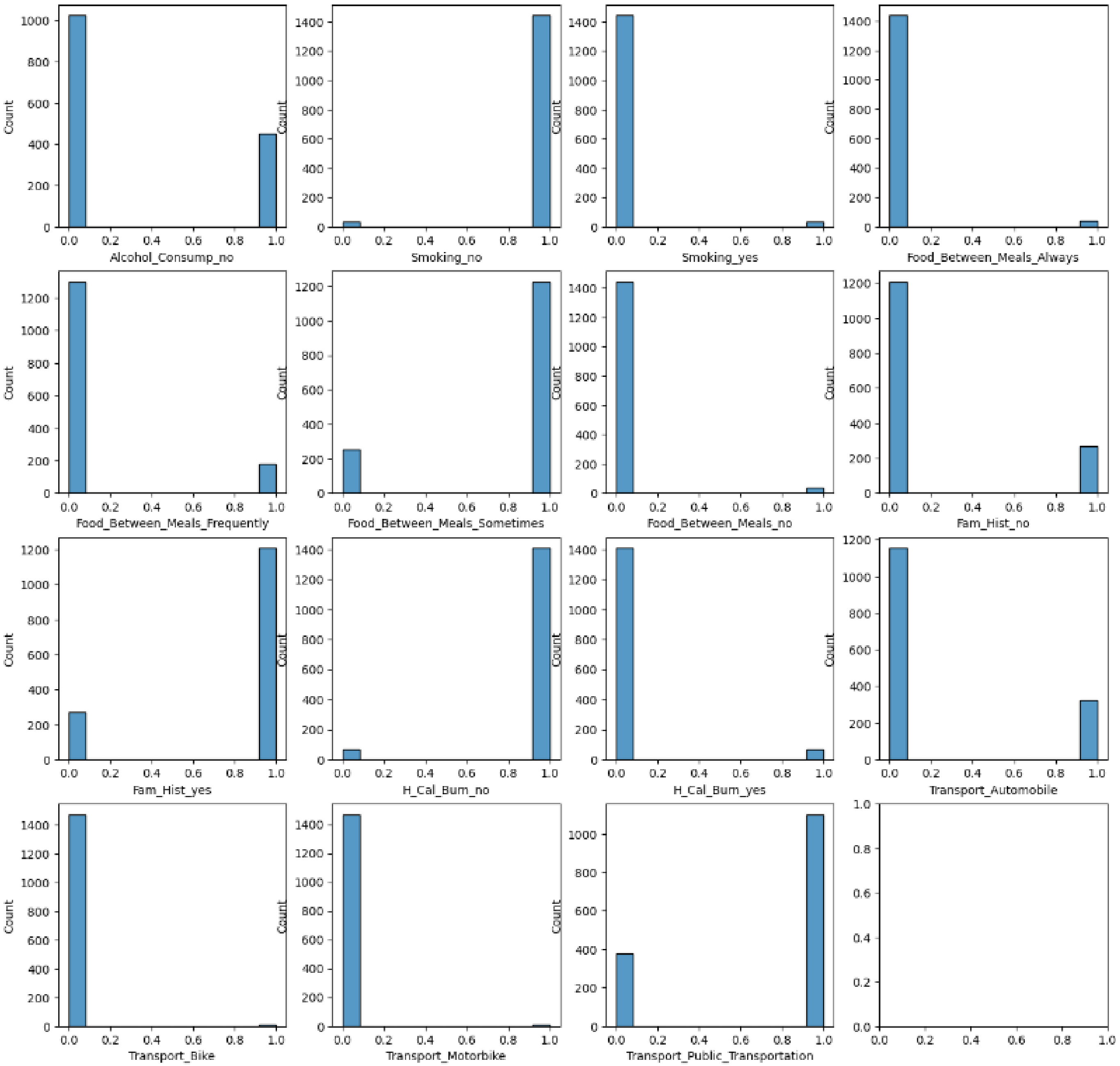
- **Results (the dataset analysis results and the experimental results).**

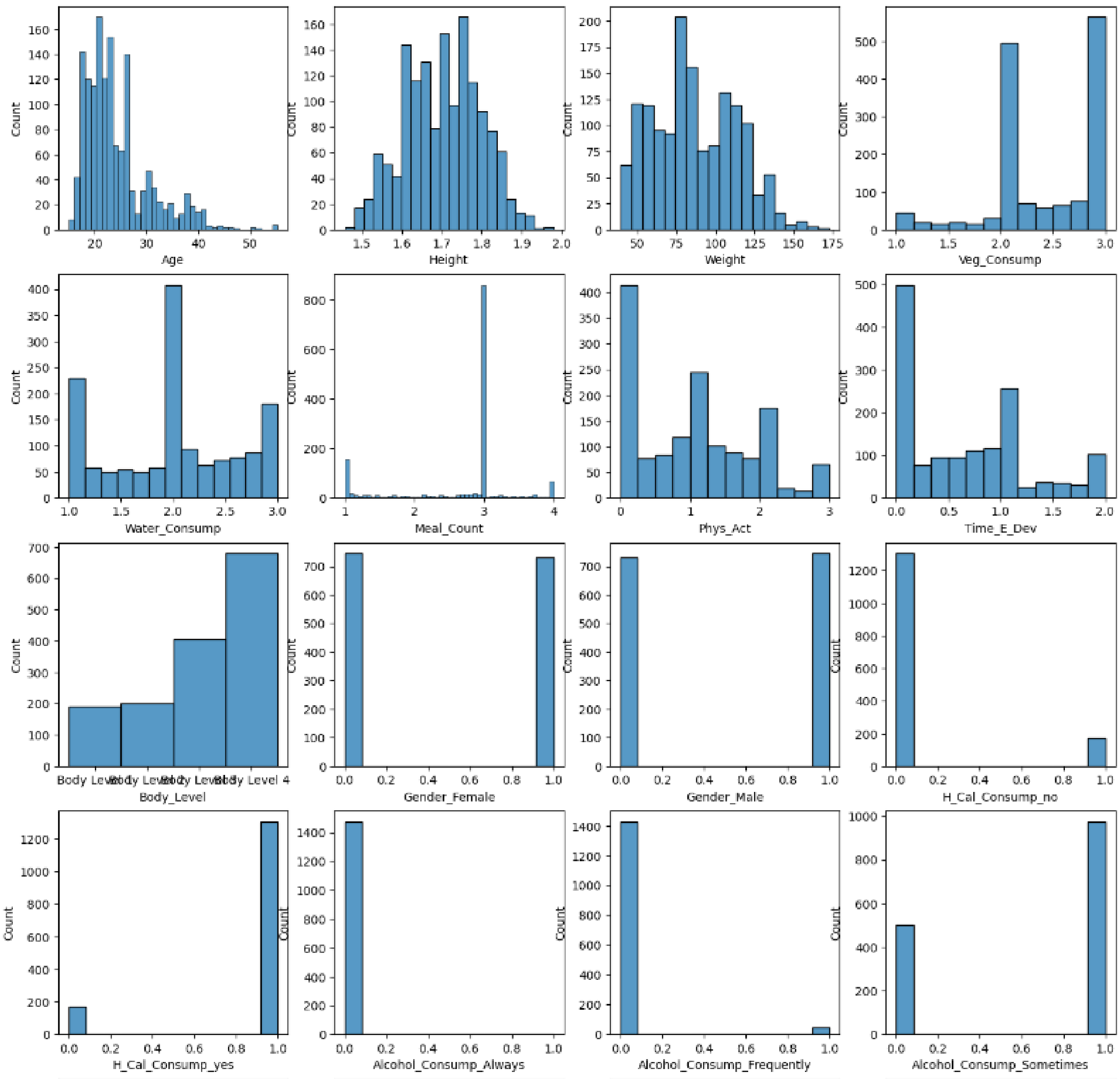
1- Target Class Distribution



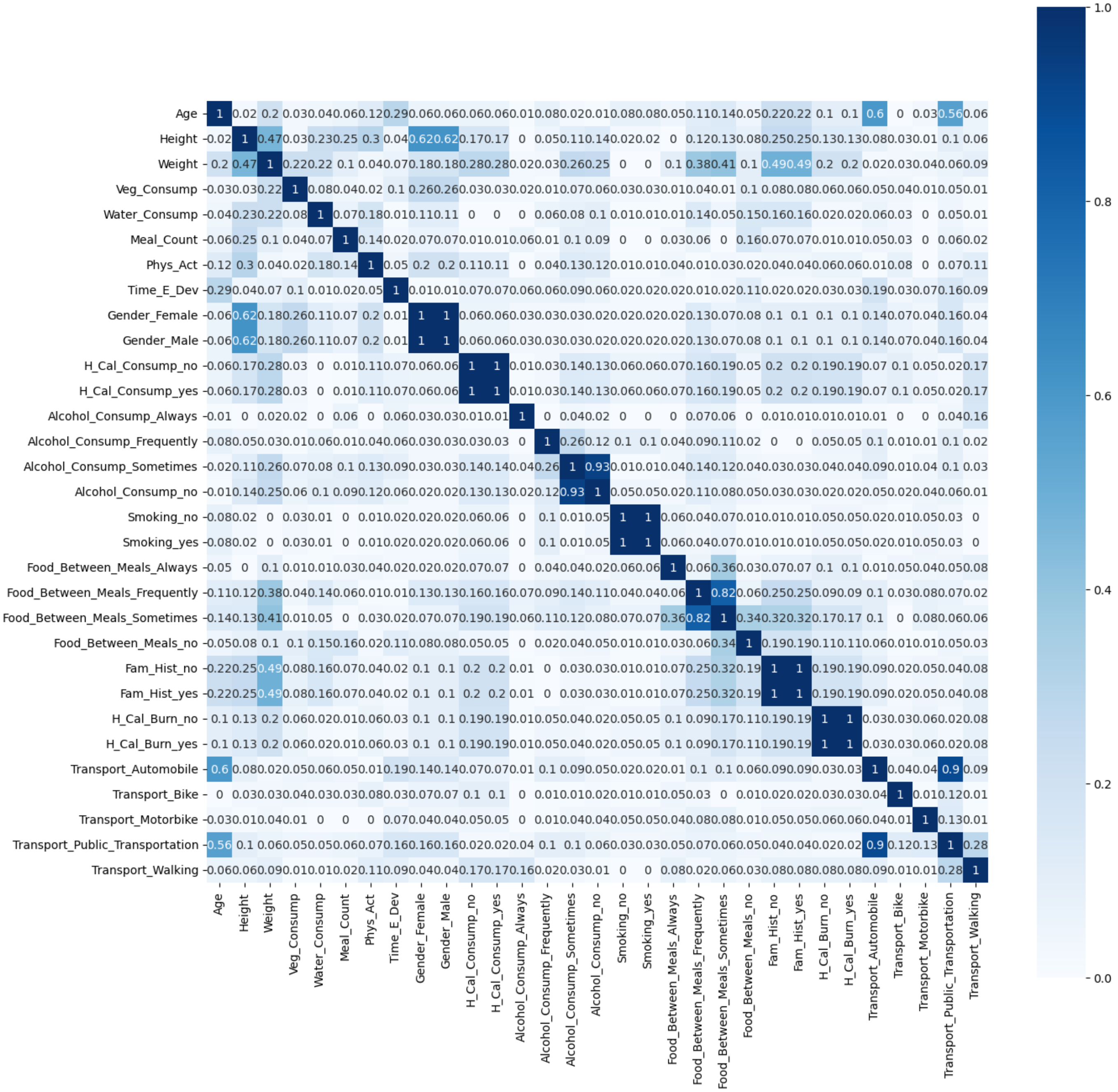
Can be seen that is suffered from class imbalance

2- Features Distribution

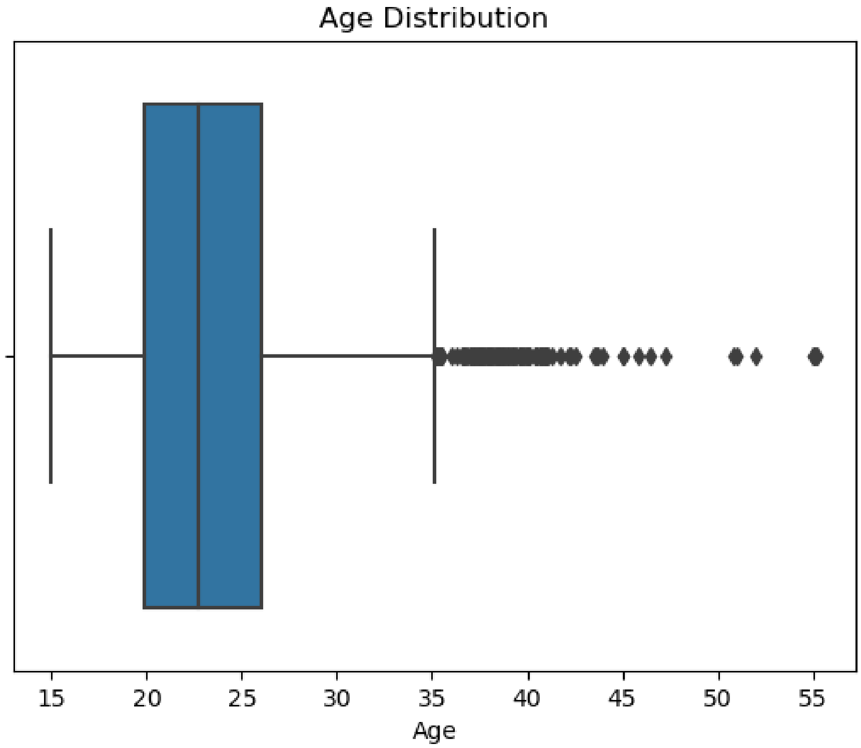


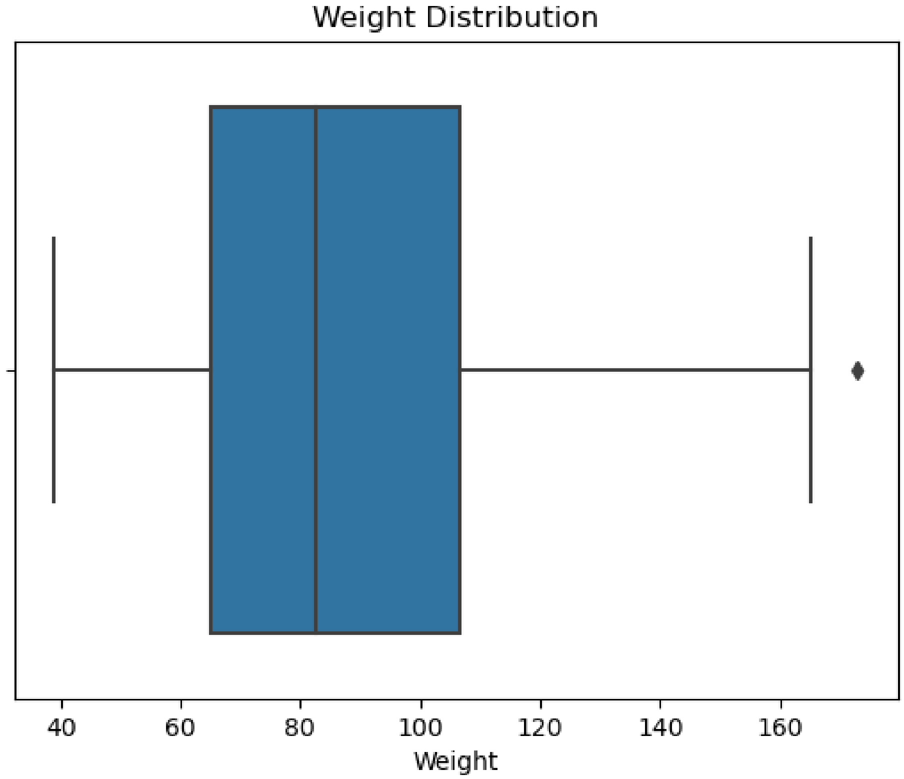

Count
Age
0 20 40 60 80 100 120 140 160
20 30 40 50
Height
1.5 1.6 1.7 1.8 1.9 2.0
0 20 40 60 80 100 120 140 160
Weight
50 75 100 125 150 175
0 25 50 75 100 125 150 175 200
Veg_Consump
1.0 1.5 2.0 2.5 3.0
0 100 200 300 400 500
Water_Consump
1.0 1.5 2.0 2.5 3.0
0 50 100 150 200 250 300 350 400
Meal_Count
1 2 3 4
0 200 400 600 800
Phys_Act
0 1 2 3
0 50 100 150 200 250 300 350 400
Time_E_Dev
0.0 0.5 1.0 1.5 2.0
0 100 200 300 400 500
Body_Level
Body Level 1 Body Level 2 Body Level 3 Body Level 4
0 100 200 300 400 500 600 700
Gender_Female
Gender_Male
0.0 0.2 0.4 0.6 0.8 1.0
0 100 200 300 400 500 600 700
H_Cal_Consump_no
0 200 400 600 800 1000 1200
H_Cal_Consump_yes
0 200 400 600 800 1000 1200
Alcohol_Consump_Always
Alcohol_Consump_Frequently
0 200 400 600 800 1000 1200 1400
Alcohol_Consump_Sometimes
0 200 400 600 800 1000

3-Correlation between Classes



4-Some features Outliers





- **A discussion of your experimental results (an in-depth analysis).**

1-Models Evaluation

- **Models using Random OverSample**

	Before Cross Validation	After Cross Validation
Logistic Reg	.78	.88
SVM	.93	.99
Adaboost	.98	.99
Random Forrest	.97	.96

- **Models using SMOT**

	Before Cross Validation	After Cross Validation
Logistic Reg	80	89
SVM	94	98
Adaboost	96	99
Random Forrest	97	98

2-Models HyperParameter And Regularization Analysis(using Cross Validation)

- **Logistic Regression HyperParameters**

  Regularization in logistic regression helps prevent overfitting by adding a penalty term to the loss function. The penalty term discourages the model from assigning high weights to features, which can lead to overfitting. By increasing the regularization strength (decreasing C), the penalty on the weights becomes stronger, resulting in smaller weight values

  ### C

  - *The C hyperparameter controls the inverse of regularization strength in logistic regression.*
  - *A smaller value of C increases the regularization strength.*
  - *Conversely, a larger value of C decreases the regularization strength.*

  ### penalty

  - *'l2': It adds an L2 penalty term to the loss function, promoting smaller weights in the model.*
  - *'l1': It adds an L1 penalty term to the loss function, promoting sparsity by driving some weights to exactly zero.*

  ### solver

  The solver hyperparameter specifies the algorithm used for optimization in logistic regression.

  - *'liblinear': Limited to one-versus-rest schemes, suitable for small-to-medium-sized datasets.*
  - *'saga': Supports multinomial loss and is suitable for large datasets. Note: The solver options 'liblinear' and 'saga' both support L1 and L2 penalties*

- **SVM HyperParameters**

C

- *if c is small then the margin is large and the model is more tolerant of misclassifications.*
- *if c is large then the margin is small and the model is less tolerant of misclassifications.*

Kernel

- **linear kernel**

The linear kernel is the simplest type of kernel. It represents a linear transformation of the input data into a higher-dimensional space. The linear kernel is suitable for linearly separable data, where the classes can be separated by a straight line or hyperplane.

- **polynomial kernel**

The polynomial kernel is a generalization of the linear kernel. It transforms the input data into a higher-dimensional space using a polynomial function. The polynomial kernel is suitable for non-linearly separable data, where the classes can be separated by a curved line or curved hyperplane.

- **RBF kernel**

The RBF kernel is a generalization of the polynomial kernel. It transforms the input data into a higher-dimensional space using a radial basis function. The RBF kernel is suitable for non-linearly separable data, where the classes can be separated by a curved line or curved hyperplane.

Gamma

- *controls the distance of influence of a single training example:*
- *A small gamma value implies a large influence, while a large gamma value means a smaller influence.*
- *Higher values of gamma can lead to overfitting, so it's important to test different values and find an optimal balance.*
- *small gamma may lead to underfitting but better generalization*
- *large gamma may lead to overfitting but poor generalization*
- *gamma = 'auto' uses 1/n_features good for large datasets with many features*
- *gamma = 'scale' uses 1/(n_features * X.var()) good for small datasets and X.var() is the variance of the features*

- **Adaboost HyperParameters**

  base estimator criterion

  Specifies the function to measure the quality of a split in the decision tree.

  It can take either 'gini' or 'entropy' as options.

  base estimator max depth

  Defines the maximum depth of the decision tree.

  None means there is no maximum depth, allowing the tree to grow until all leaves are pure or contain a minimum number of samples defined by other parameters.

  base estimator min samples split

  Specifies the minimum number of samples required to split an internal node.

  base estimator min samples leaf

  Defines the minimum number of samples required to be at a leaf node.

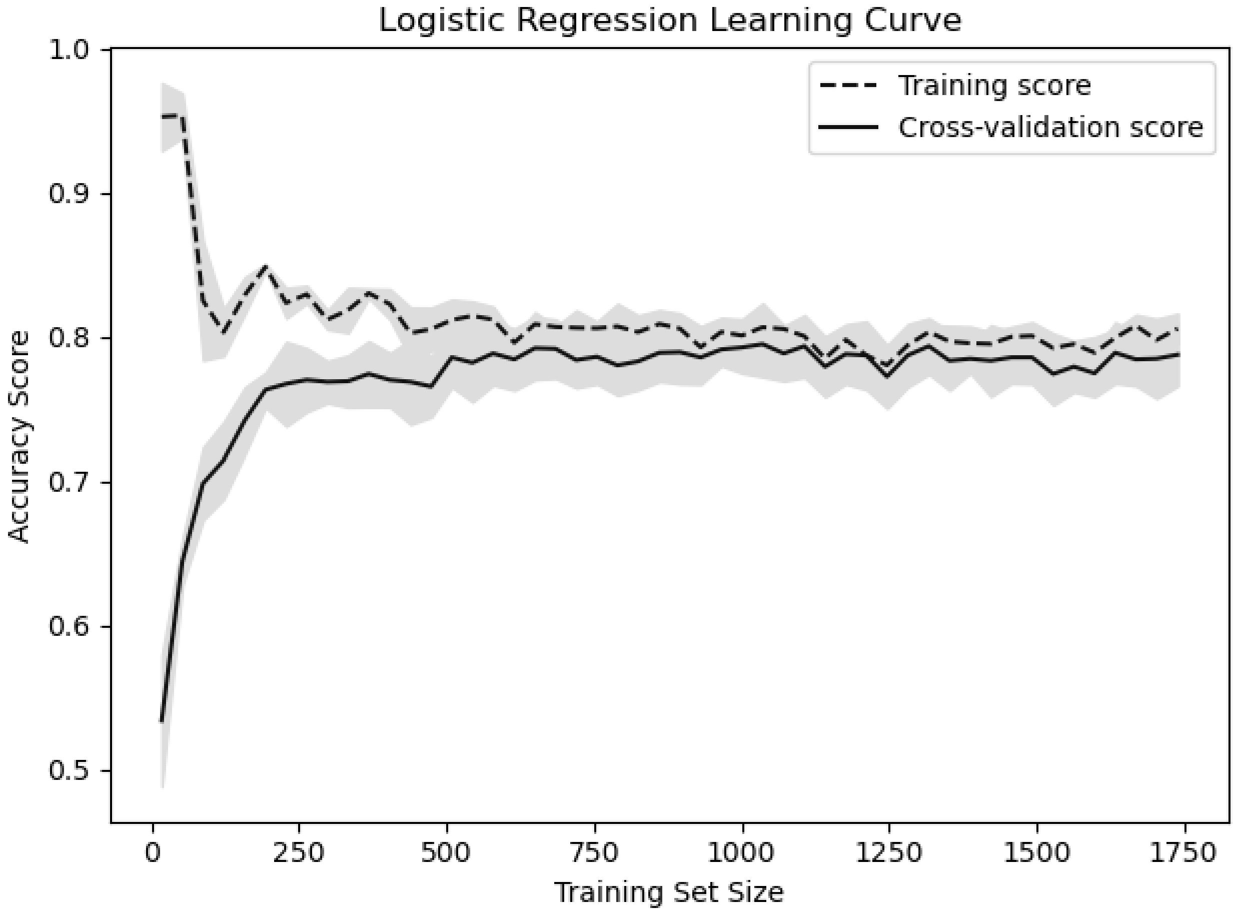
  n estimators

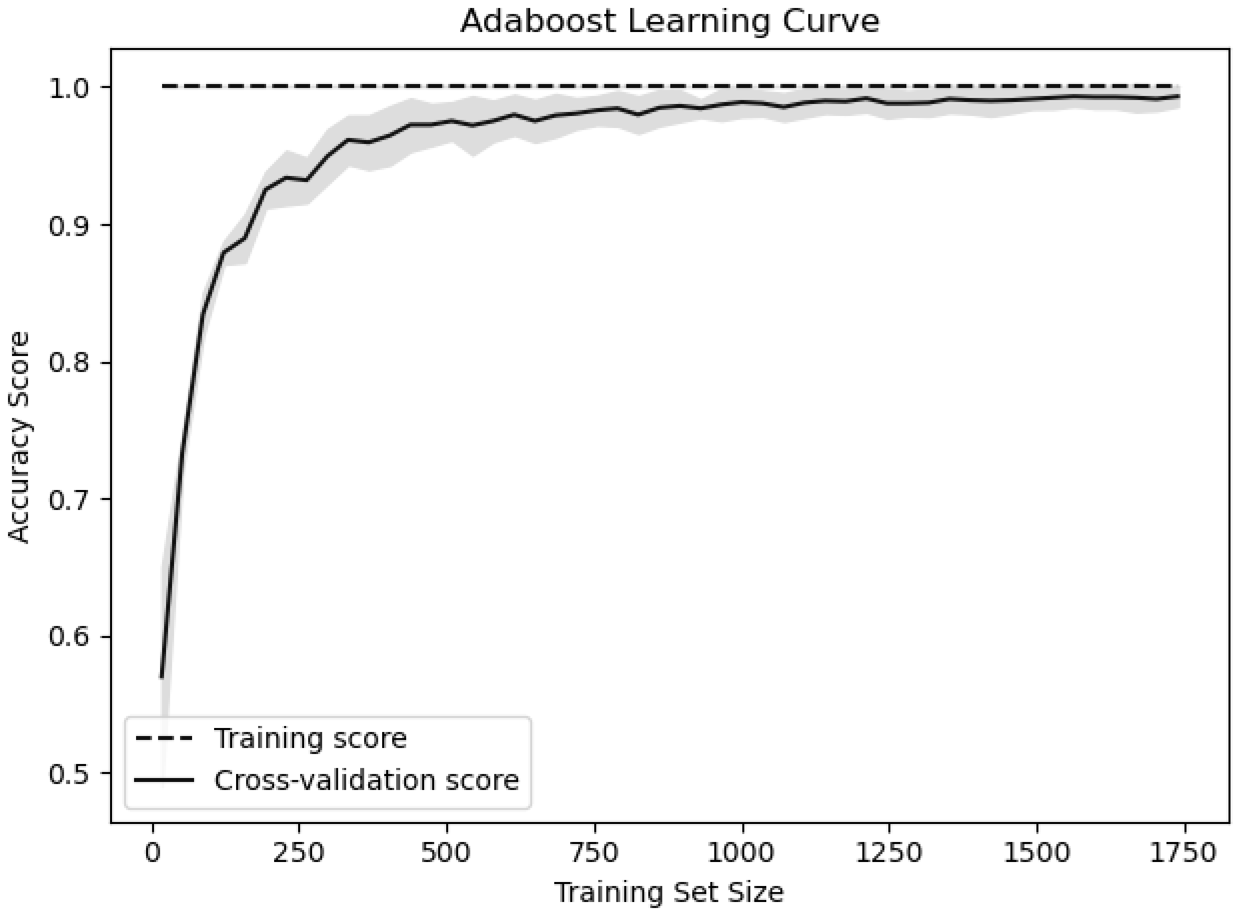
  Indicates the number of base estimators (decision trees) to be included in the AdaBoost ensemble.

  learning rate

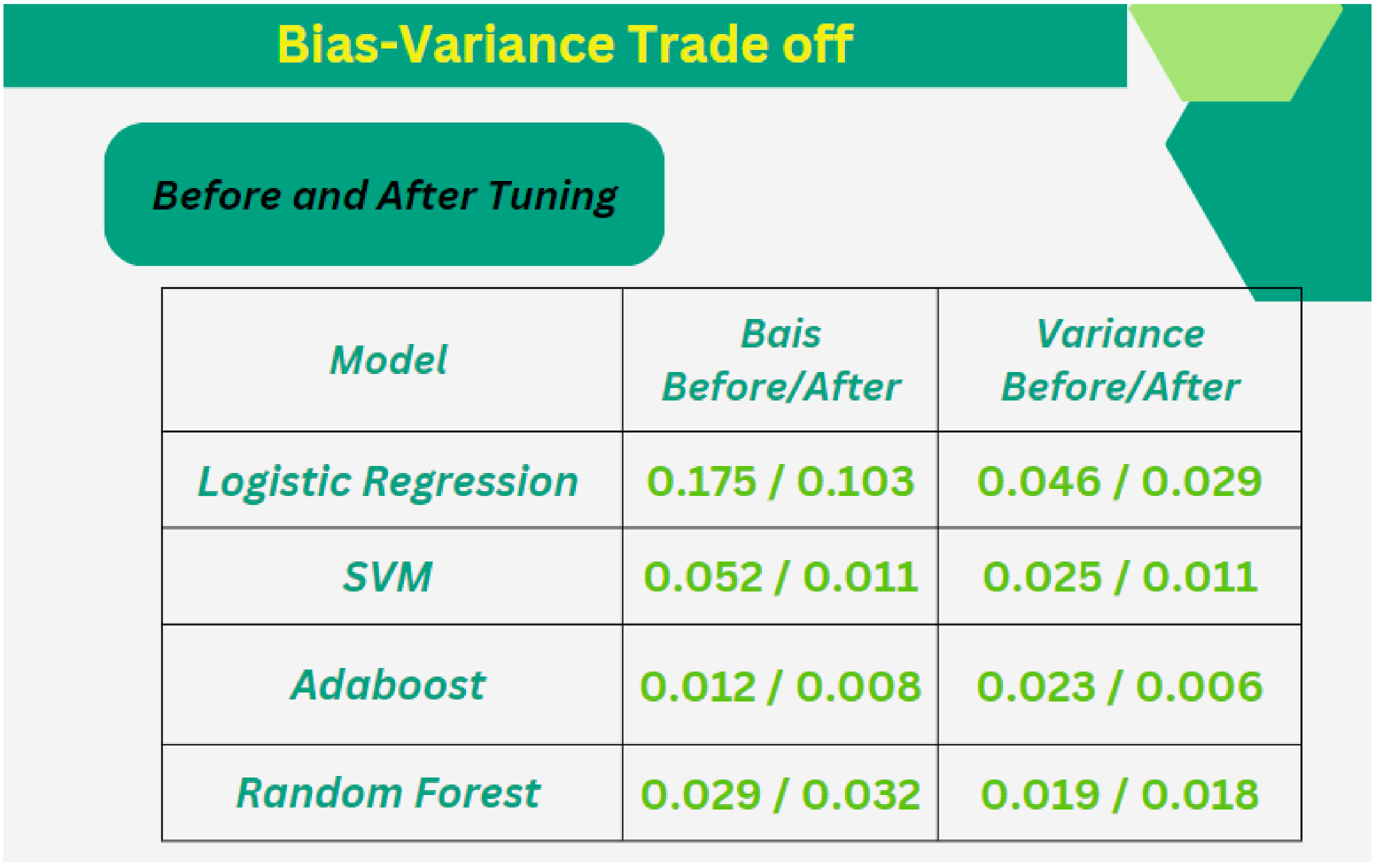
  Determines the contribution of each base estimator to the ensemble.

3-Learning Theory




Adaboost Learning Curve
1.0
0.9
0.8
0.7
0.6
0.5
Accuracy Score
0
250
500
750
1000
1250
1500
1750
Training Set Size
Training score
Cross-validation score

4-Bias-Variance Trade off



Bias-Variance Trade off

Before and After Tuning

Model	*Bais Before/After*	*Variance Before/After*
Logistic Regression	0.175 / 0.103	0.046 / 0.029
SVM	0.052 / 0.011	0.025 / 0.011
Adaboost	0.012 / 0.008	0.023 / 0.006
Random Forest	0.029 / 0.032	0.019 / 0.018

- **Your conclusion.**

❖ *HperParameters plays an important rule in Machine Learning*

❖ *SVM and Adaboost achieved the highest accuracy and SVM is much faster as Adabost Has too much combinations to consider*

❖ *Hyperparameters helps in getting rid of overfitting*

❖ *Oversampling is essential in machine learning case class imbalance*

❖ *Understanding data well before starting the project is a must in machine learning*