Predicting the Obesity Level

Authors: Siming Yin and Xingzhi Ma Team: N.N

- Executive Summary -

Our project aims to develop a predictive model that can identify the risk of obesity in individuals based on their eating habits and physical condition. By leveraging data-driven insights, this project seeks to contribute to early intervention and personalized health recommendations to reduce the chance of having obesity. In this project, we will mainly focus on predictive and prescriptive analysis by applying classification and visualization models.

[Decisions to be impacted]

1. **Public Investment on preventing obesity**

   Predicting the possibility of obesity and implementing preventive measures can help reduce the cost of public investments in treating obesity-related health issues. For example, the predictive model can help identify individuals at higher risk of obesity based on the physical condition, eating habits or lifestyle. With early identification, public health programs can educate people on healthy eating and exercise so that they reduce the chance of developing obesity issues and the cost of treating obesity. Also with the help of technology, we can develop a health app that combined with predictive models, can enable remote monitoring of individuals at risk of obesity. It can provide guidance, support, and feedback to individuals in order to help reduce the need for hospital visits and the related costs.

2. **Identification of health treatment**

   Predicting the possibility of obesity can be a valuable tool in identifying health treatment needs and setting achievable weight loss or weight maintenance goals. With the help of the predictive model, individuals can identify whether or not they are at the risk of obesity and provide some guidelines and recommendations on predicting obesity. The predictive model can be integrated into monitoring systems that track an individual's progress in real-time and build up personalized medication and therapies.

3. **Provide social support to maintain a healthy lifestyle**

   Since obesity is often a chronic condition that requires long-term management, predictive models can assist healthcare providers in designing treatment plans that are sustainable and adaptable as the patient's health evolves over time. It helps provide social support to people to maintain a healthy lifestyle and reduce the chance of getting obesity.

[Business value]

1. **Personal Health Improvement: Reduce the possibility of individuals suffering from obesity.**

   One of the business values of this predictive model is that it helps improving personal health involves making lifestyle changes and adopting healthy habits. For example, it helps people realize the importance of quality sleep and efficient stress management, and educate themselves with nutrition knowledge. Setting achievable and specific health goals to maintain healthy weight.
2. **Public Health Improvement: Help the government to lower the expenditure on preventing obesity.**

   The predictive model also can play an important role in public health improvement through helping the government to lower the cost of preventing obesity. By prioritizing preventive measures, such as health education, social programs, and policies targeting healthier environments, the government can effectively reduce the chance of people getting obesity in the community.
3. **Research Cost Reduction: Generate valuable research findings and insights into the relationship between eating habits, physical condition, and obesity risk.**

   Beside that, the other business value of the predictive model would be to reduce the cost of obesity research. The model employs efficient methodologies and technologies to generate valuable insights into the relationship between eating habits, physical condition, and the risk of obesity. It helps researchers uncover the patterns between these features and discover some potential research with different other fields, such as nutrition and epidemiology.

[Data assets]

Dataset for estimation of obesity levels based on eating habits and physical condition in individuals from Colombia, Peru and Mexico.

- Data Preprocessing -

[Data Description]

We are using the Dataset for estimation of obesity levels based on eating habits and physical condition in individuals from Colombia, Peru and Mexico. This dataset contains the information of people from the countries of Mexico, Peru and Colombia, with ages between 14 and 61, obesity level and diverse eating habits and physical condition. The data was collected using a web platform with a survey (see Table 1) where anonymous users answered each question. It comprises 2111 rows, representing 2111 individuals, and 17 columns, representing 16 features and 1 target (obesity level) for each individual. The dataset features can be categorized into two groups based on their data type: categorical and ratio. Our ratio data are Ages, Height, and Weight. And our categorical data includes: Gender, Family History With Overweight (FHWO), Consumption of High Caloric Food (FAVC), Consumption of Vegetables(FCVC), Number of Main Meals (NCP), Consumption of Food Between Meals (CAEC), Smoke Consumption of Water Daily (CH2O), Calories Consumption Monitoring (SCC), Physical Activity Frequency (FAF), Time Using Technology Devices (TUE), Consumption of Alcohol (CALC) Transportation Used (MTRANS). Obesity level is categorized by the calculation of BMI (Body Mass Index): • Underweight Less than 18.5 •Normal 18.5 to 24.9 • Overweight 25.0 to 29.9 • Obesity I 30.0 to 34.9 • Obesity II 35.0 to 39.9 • Obesity III Higher than 40. The entire data is composed of 77% generated data by Weka tool and the SMOTE filter and 23% directly collected data from the survey. Since the obesity level is calculated by BMI, we will not use weight and height as our features to build the predictive model.

Since our simple model is only taking use of 14 features, including , we decided to drop columns that are not being used (weight and height).

$MLM1: ML_{clean1} = (R^{2111\times14}, R^{2111\times14}, F_1(x;\theta) = \theta x, P_\Theta(\theta) = [1100111111111111], L_1: trivial)$

[Data Visualization]

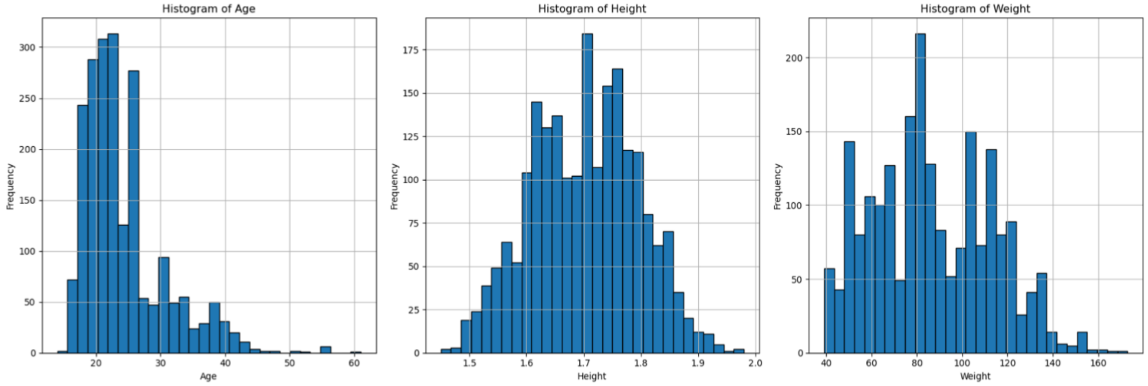


Fig: Histogram of Age (Left) Histogram of Height (Middle) Histogram of Weight (Right)

We can see that most of our samples are taken from people under 30. From the Histogram of Age on the left, we can see that most of the surveyees are aged around 15 to 28. The data of Height has a maximum of 198 and minimum of 145, and the median value is about 170. From the plot of Height, we find that most data are in the interval between 160 to 177. There is a peak around 163, a second peak at around 170, and a third peak at around 175. The data of Weight is ranged from 39 to 173. From the histogram of Weight on the right, we see that our data has a major peak at around 80, and several secondary peaks around 45, 100, and 105. From all three plots, we see that they all seem likely to form a multimodal distribution. The histogram of Age is positively skewed. The histogram of Weight is slightly right skewed. The histogram of Height has a normal-distribution-like shape compared to the other two plots because of the central tendency, but not in general. The reason lead to this could be the natural difference in Height and Weight between male and female. So we split the data by gender and generate the following plots.

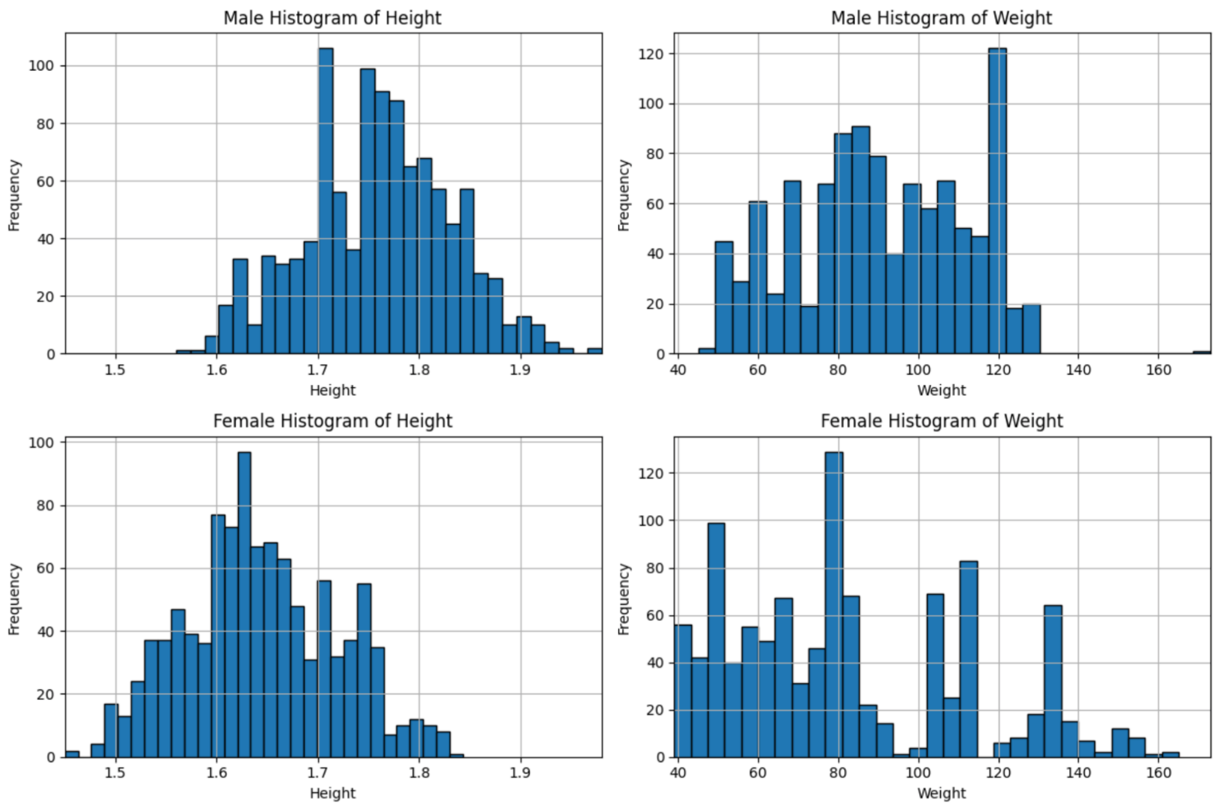


Fig: Male Histogram of Height(Up Left) Male Histogram of Weight (Up Right) Female Histogram of Height (Up Left) Female Histogram of Weight (Down Right)

From the Histograms shown above, we observe that, though we split our data by gender, the histograms of Height still display multiple peaks in both male and female. The data of Weight are still inconsistent and tend to form multimodal distribution. The potential explanation of this could be the regional difference between the sample we collected.

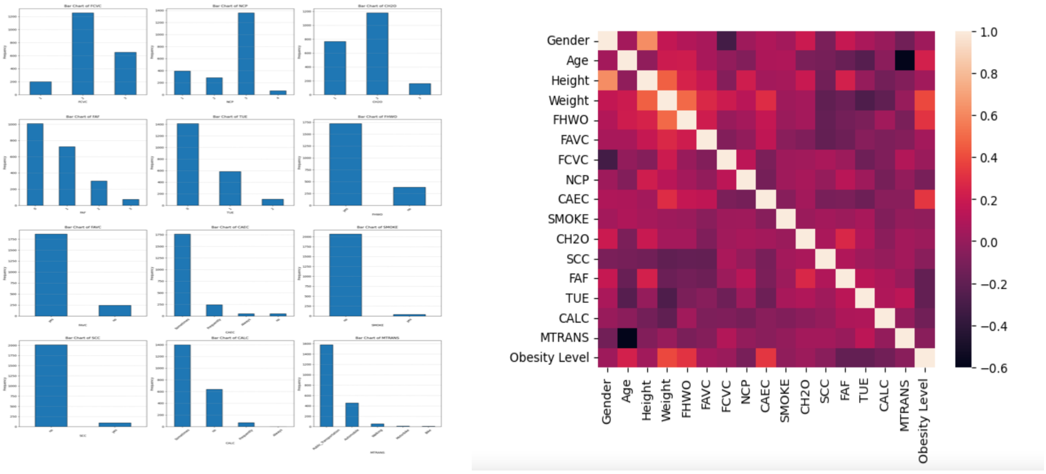


Fig: Barplot of the categorical data (Left) and correlation plot (Right)

The left bar plots show the frequency of categories of each feature. Some of the features have a dominant category that has a really high percentage. The most prominent one is the feature smoke (SMOKE). 97% of the data are in the category of *No*, and only 3% of the data are in *Yes*. The feature Calories Consumption Monitoring (SCC) has 95% of the data in the category of *Yes* and 5% of data are in *No*. The feature Consumption of High Caloric Food (FAVC) on the other hand has 88% of the data in *Yes* and 12% in *No*. So, we can say that the majority of our surveyees are non-smokers who did not monitor their calories consumption, and have consumed high caloric food. The correlation plot on the right shows the relationship between different features. We see from the plot that Weigh and Family History With Overweight (FHWO) are most correlated features with Obesity level. This is intuitive, since Obesity is very likely genetically inherited. We also see that Obesity level is correlated with Consumption of Food Between Meals (CAEC) and Age. It is unexpected that Height is not highly correlated with Obesity level, since BMI (Body Mass Index) is calculated by Height and Weight.

[Data Cleaning]

The dataset that we are using synthetic data, most of the data preprocessing has been completed. There is no null value or missing data, atypical data has been deleted, and data normalization has been completed. The dataset has used the tool Weka and the filter SMOTE to generate the synthetic data in order to balance the categories of obesity levels. The balancing process decreases the probability of skewed learning in favor of a majority class.

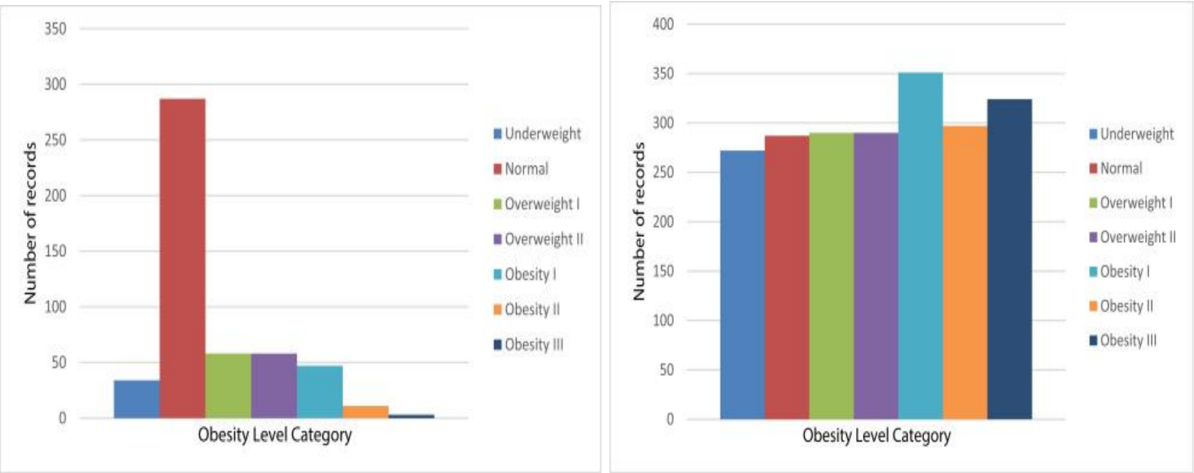


Fig: Unbalanced distribution of data regarding the obesity levels category (Left) Balanced distribution of data regarding the obesity levels category (Right)

(https://doi.org/10.1016/j.dib.2019.104344)

To make the column name consistent, we renamed some of the column names to make it easy to understand and check out the number of the features and decide the target column, which is the obesity level. Since the data are collected by the survey, most of the columns are categorical data. Some of them have been encoded into numerical data and some are not. We have transformed those numerical data into integers and the rest. For example, one column is called "Frequency of consumption of vegetables" which has the values 'Always', 'Frequently', 'Sometimes', 'no'. Then we labeled them into number 0, 1, 2, 3.

$$MLM2: ML_{clean2} = (R^{2111\times14}, R^{2111\times14}, F_2 embedding\ parameter, L_2: \ trivial)$$

[Outlier Detection]

Since most of the features in our dataset are categorical data, we can use bar plot to determine the outlier. For example, the columns "Smoke" and "SCC" have a pretty high percentage of one specific value. Take "Smoke" as an example, around 97% of the population are non-smoker, which means the outlier would be those who smokes. By looking at the target value, Obesity level, we are using pie charts and bar charts to check the outliers. In the data description part, we have mentioned that the dataset has been balanced by filter SMOTE, then there are no outliers for our target value. In our dataset, there are three numerical data, we do a boxplot of these three columns.

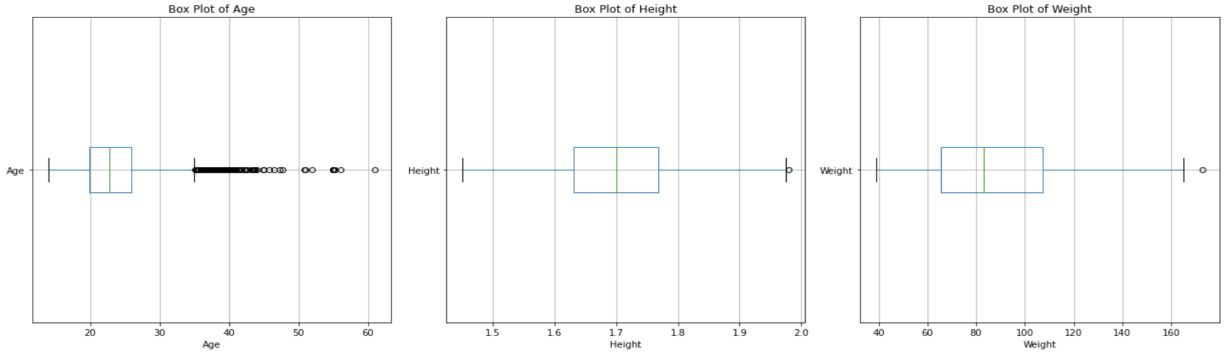


Fig: Box plot of Age (Left) Box plot of Height (Mid) Box plot of Weight (Right)

Notice that there is only one outlier for "Height" and "Weight". In general, these two columns distribute pretty normally. However, for the "Age", we found that with the minimum of column is 14, the 25% of Interquartile range (IQR) is 19.95, the mean is 24.31, the medium is 22.78, the 75% of IQR is 26, and the maximum is 61. We discovered that the score data is skewed to the left with plenty of younger ages. Lastly, we have also performed PCA on

Obesity level with the two highest correlated features (FHWO, CAEC). There's a significant overlap between several obesity levels in the central region of the plot. This indicates that, based on the first two principal components, many individuals from different obesity levels share some similar characteristics. The plot seems to show a major cluster in the center with some scatter around it. The “Insufficient_Weight" seems more distinct compared to other levels, especially in certain regions of the plot. In general, The majority of data points, particularly “Obesity_Type_I”, and “Overweight_Level_I”, are clustered around the center. The wide distribution of data points suggests there is considerable variability within the dataset.

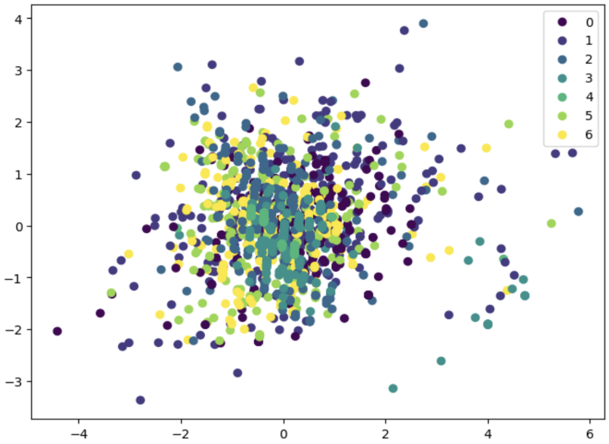
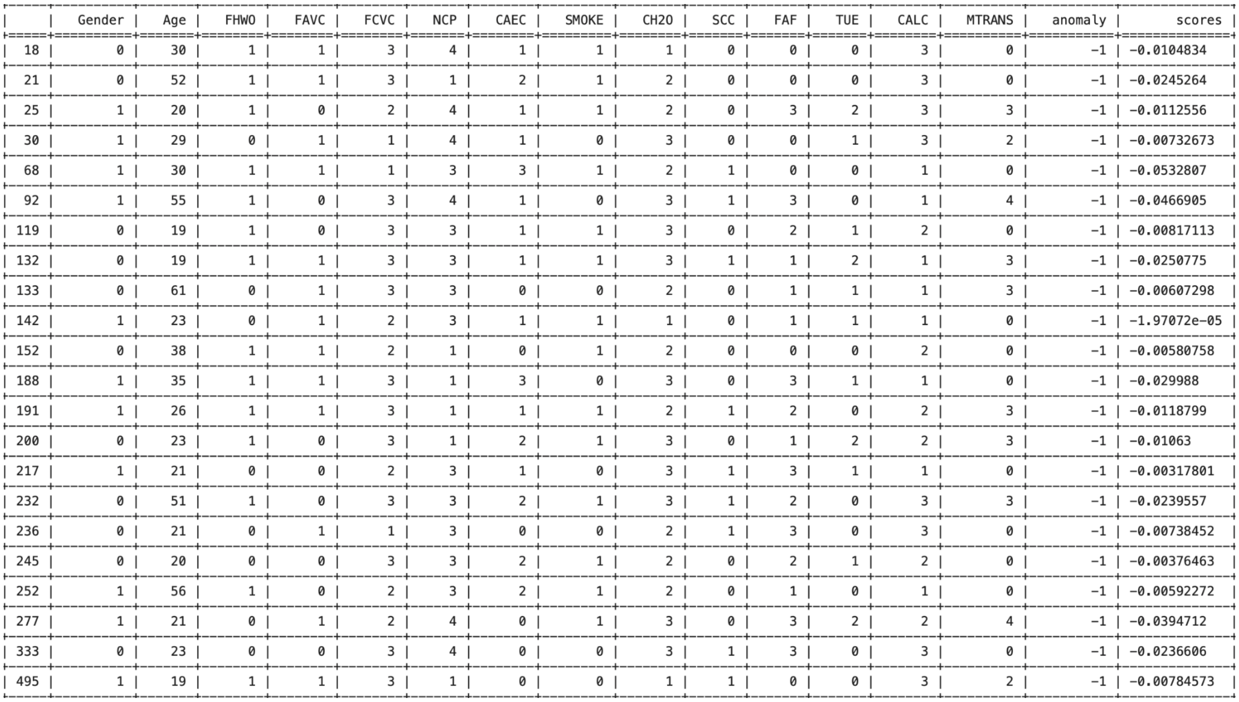


Fig: 2D PCA scatter plot for obesity level

The other thing that we do to perform the outlier detection is using the Isolation Forest to detect the anomaly data point in our data set. Isolation forest is an unsupervised learning algorithm that identifies anomalies by isolating outliers in the data. Because we have a small dataset, we have defined the expected proportion of outliers in the data set to be 1%, which is the contamination parameter in the model. There are 22 predictive outliers that have been identified.



```
+-----+----------+------+--------+--------+--------+-------+--------+---------+--------+-------+-------+-------+--------+----------+-----------+--------------+
|     |   Gender |  Age |   FHWO |   FAVC |   FCVC |   NCP |   CAEC |   SMOKE |   CH2O |   SCC |   FAF |   TUE |   CALC |   MTRANS |   anomaly |       scores |
+=====+==========+======+========+========+========+=======+========+=========+========+=======+=======+=======+========+==========+===========+==============+
|  18 |        0 |   30 |      1 |      1 |      3 |     4 |      1 |       1 |      1 |     0 |     0 |     0 |      3 |        0 |        -1 | -0.0104834   |
+-----+----------+------+--------+--------+--------+-------+--------+---------+--------+-------+-------+-------+--------+----------+-----------+--------------+
|  21 |        0 |   52 |      1 |      1 |      3 |     1 |      2 |       1 |      2 |     0 |     0 |     0 |      3 |        0 |        -1 | -0.0245264   |
+-----+----------+------+--------+--------+--------+-------+--------+---------+--------+-------+-------+-------+--------+----------+-----------+--------------+
|  25 |        1 |   20 |      1 |      0 |      2 |     4 |      1 |       1 |      2 |     0 |     3 |     2 |      3 |        3 |        -1 | -0.0112556   |
+-----+----------+------+--------+--------+--------+-------+--------+---------+--------+-------+-------+-------+--------+----------+-----------+--------------+
|  30 |        1 |   29 |      0 |      1 |      1 |     4 |      1 |       0 |      3 |     0 |     0 |     1 |      3 |        2 |        -1 | -0.00732673  |
+-----+----------+------+--------+--------+--------+-------+--------+---------+--------+-------+-------+-------+--------+----------+-----------+--------------+
|  68 |        1 |   30 |      1 |      1 |      1 |     3 |      3 |       1 |      2 |     1 |     0 |     0 |      1 |        0 |        -1 | -0.0532807   |
+-----+----------+------+--------+--------+--------+-------+--------+---------+--------+-------+-------+-------+--------+----------+-----------+--------------+
|  92 |        1 |   55 |      1 |      0 |      3 |     4 |      1 |       0 |      3 |     1 |     3 |     0 |      1 |        4 |        -1 | -0.0466905   |
+-----+----------+------+--------+--------+--------+-------+--------+---------+--------+-------+-------+-------+--------+----------+-----------+--------------+
| 119 |        0 |   19 |      1 |      0 |      3 |     3 |      1 |       1 |      3 |     0 |     2 |     1 |      2 |        0 |        -1 | -0.00817113  |
+-----+----------+------+--------+--------+--------+-------+--------+---------+--------+-------+-------+-------+--------+----------+-----------+--------------+
| 132 |        0 |   19 |      1 |      1 |      3 |     3 |      1 |       1 |      3 |     1 |     1 |     2 |      1 |        3 |        -1 | -0.0250775   |
+-----+----------+------+--------+--------+--------+-------+--------+---------+--------+-------+-------+-------+--------+----------+-----------+--------------+
| 133 |        0 |   61 |      0 |      1 |      3 |     3 |      0 |       0 |      2 |     0 |     1 |     1 |      1 |        3 |        -1 | -0.00607298  |
+-----+----------+------+--------+--------+--------+-------+--------+---------+--------+-------+-------+-------+--------+----------+-----------+--------------+
| 142 |        1 |   23 |      0 |      1 |      2 |     3 |      1 |       1 |      1 |     0 |     1 |     1 |      1 |        0 |        -1 | -1.97072e-05 |
+-----+----------+------+--------+--------+--------+-------+--------+---------+--------+-------+-------+-------+--------+----------+-----------+--------------+
| 152 |        0 |   38 |      1 |      1 |      2 |     1 |      0 |       1 |      2 |     0 |     0 |     0 |      2 |        0 |        -1 | -0.00580758  |
+-----+----------+------+--------+--------+--------+-------+--------+---------+--------+-------+-------+-------+--------+----------+-----------+--------------+
| 188 |        1 |   35 |      1 |      1 |      3 |     1 |      3 |       0 |      3 |     0 |     3 |     1 |      1 |        0 |        -1 | -0.029988    |
+-----+----------+------+--------+--------+--------+-------+--------+---------+--------+-------+-------+-------+--------+----------+-----------+--------------+
| 191 |        1 |   26 |      1 |      1 |      3 |     1 |      1 |       1 |      2 |     1 |     2 |     0 |      2 |        3 |        -1 | -0.0118799   |
+-----+----------+------+--------+--------+--------+-------+--------+---------+--------+-------+-------+-------+--------+----------+-----------+--------------+
| 200 |        0 |   23 |      1 |      0 |      3 |     1 |      2 |       1 |      3 |     0 |     1 |     2 |      2 |        3 |        -1 | -0.01063     |
+-----+----------+------+--------+--------+--------+-------+--------+---------+--------+-------+-------+-------+--------+----------+-----------+--------------+
| 217 |        1 |   21 |      0 |      0 |      2 |     3 |      1 |       0 |      3 |     1 |     3 |     1 |      1 |        0 |        -1 | -0.00317801  |
+-----+----------+------+--------+--------+--------+-------+--------+---------+--------+-------+-------+-------+--------+----------+-----------+--------------+
| 232 |        0 |   51 |      1 |      0 |      3 |     3 |      2 |       1 |      3 |     1 |     2 |     0 |      3 |        3 |        -1 | -0.0239557   |
+-----+----------+------+--------+--------+--------+-------+--------+---------+--------+-------+-------+-------+--------+----------+-----------+--------------+
| 236 |        0 |   21 |      0 |      1 |      1 |     3 |      0 |       0 |      2 |     1 |     3 |     0 |      3 |        0 |        -1 | -0.00738452  |
+-----+----------+------+--------+--------+--------+-------+--------+---------+--------+-------+-------+-------+--------+----------+-----------+--------------+
| 245 |        0 |   20 |      0 |      0 |      3 |     3 |      2 |       1 |      2 |     0 |     2 |     1 |      2 |        0 |        -1 | -0.00376463  |
+-----+----------+------+--------+--------+--------+-------+--------+---------+--------+-------+-------+-------+--------+----------+-----------+--------------+
| 252 |        1 |   56 |      1 |      0 |      2 |     3 |      2 |       1 |      2 |     0 |     1 |     0 |      1 |        0 |        -1 | -0.00592272  |
+-----+----------+------+--------+--------+--------+-------+--------+---------+--------+-------+-------+-------+--------+----------+-----------+--------------+
| 277 |        1 |   21 |      0 |      1 |      2 |     4 |      0 |       1 |      3 |     0 |     3 |     2 |      2 |        4 |        -1 | -0.0394712   |
+-----+----------+------+--------+--------+--------+-------+--------+---------+--------+-------+-------+-------+--------+----------+-----------+--------------+
| 333 |        0 |   23 |      0 |      0 |      3 |     4 |      0 |       0 |      3 |     1 |     3 |     0 |      3 |        0 |        -1 | -0.0236606   |
+-----+----------+------+--------+--------+--------+-------+--------+---------+--------+-------+-------+-------+--------+----------+-----------+--------------+
| 495 |        1 |   19 |      1 |      1 |      3 |     1 |      0 |       0 |      1 |     1 |     0 |     0 |      3 |        2 |        -1 | -0.00784573  |
+-----+----------+------+--------+--------+--------+-------+--------+---------+--------+-------+-------+-------+--------+----------+-----------+--------------+
```

Fig: Outliers in the data set by using Isolation Forest Anomaly Detection method

$$MLM3: ML_{outlier} = (R^{2089\times14},\ R^{2089\times14},\ F_3(F_1(x;\theta)) = \theta x,\ embedding\ parameter, L_3:\ trivial)$$

- Model Approach -

Because the target value obesity level is calculated by BMI formulate which is using the height and weight. The correlation between height and weight should be highly correlated to Obesity level, which can cause an issue in our predictive model. Thus we decided to use all the features except weight and height in our model.

[Predictive Model]

First, we built up Random Forest and KNN models. For each of Random Forest and KNN, We first will include all the data points in the first model and exclude outliers in the second model. We then used Grid Search, Random Search, Bayes Optimization for hyperparameter tuning and used Nested Cross-Validation for feature selections and hyperparameter tuning. We also used the PyCaret package for model selection. After the PyCaret model selection, Random Forest has the best model performance, followed by LightGBM and Extra Tree Classifier. Thus, we also implement the LightGBM machine learning algorithm and Extra Tree Classifier.

Random Forest

In this project, we are going to first implement a Random Forest algorithm. Among all other supervised classification models, Random Forest has several outstanding properties that give us more accurate and explicit outputs, particularly in the topic of predicting obesity and classifying obesity level. Random Forest is capable of dealing with multifactorial conditions, overfitting, and able to process categorical data and aggregate feature importance. Random Forest is an ensemble learning technique that builds multiple decision trees using random subsets of the data and features. It operates by creating diverse trees through sampling both data and features, reducing overfitting and improving accuracy. Each tree independently makes a prediction, and in classification tasks, the final prediction is determined by the most frequent class among the trees. At the data preparation and preprocessing stage, we have set the feature Obesity Level as the target value, and converted all the categorical data into numerical format. When we do train test split, we use bootstrap to randomly select training dataset and testing dataset.

$$\text{Gini Impurity(GI)} = 1 - \sum_{i=1}^{k} p_i^2$$

$$MLM4: ML_{model1} = (R^{1067\times14}, R^{481\times14}, F_4(F_3(x:\theta), O_x) = x, P_\Theta(O_x) = O_x, L_4 = GI)$$

$$MLM5: ML_{result1} = ML_{model1} \circ ML_{clean3} = (R^{1067\times14}, R^{481\times14}, F_5(F_3(x:\theta), O_x) = x, P_\Theta(O_x) = O_x, L_5 = GI)$$

KNN (K-nearest Neighbors)

KNN is used to classify different obesity levels, it counts the number of data points in each obesity level among the K nearest neighbors and assigns the obesity level label that is most common among them to the new data points. First we decided to use Manhattan distance to calculate the distance in order to find the nearest neighbors of given data points. Since we have 16 features in our dataset, the Manhattan distance for multiple features can measure the distance between two points in a multidimensional space.

$$Manhattan\ Distance = (\sum_{i=1}^{n} |X_i - Y_i|)$$

$MLM6: ML_{model2} = (R^{1067\times14}, R^{2089\times14}, F_6(F_3(x: \theta), O_x) = x, P_{\Theta}(O_x) = S_x, L_6 = MAE)$

$where\ O_x$ = *the similarity* obesity level of data point and input and MAE = $\frac{\sum_{i=1}^{n}|s_i|}{n}$

MLM7:

$ML_{result2} = ML_{model1} \circ ML_{clean1} = (R^{1067\times14}, R^{481\times14}, F_7(F_3(x: \theta), O_x) = x, P_{\Theta}(O_x) = O_x, L_7 = MAE)$

LightGBM

Using PyCaret for model selection, the second best model for our dataset is Light GBM. So we choose to implement this model. Light GBM is a gradient boosting framework that uses the tree based learning algorithm. LightGBM grows trees leaf-wise (best-first), while most other trees learning algorithms grow trees by level (depth)-wise. The target value contains multiple classes, we choose the objective as 'multiclass' for multiclass classification with cross-entropy loss.

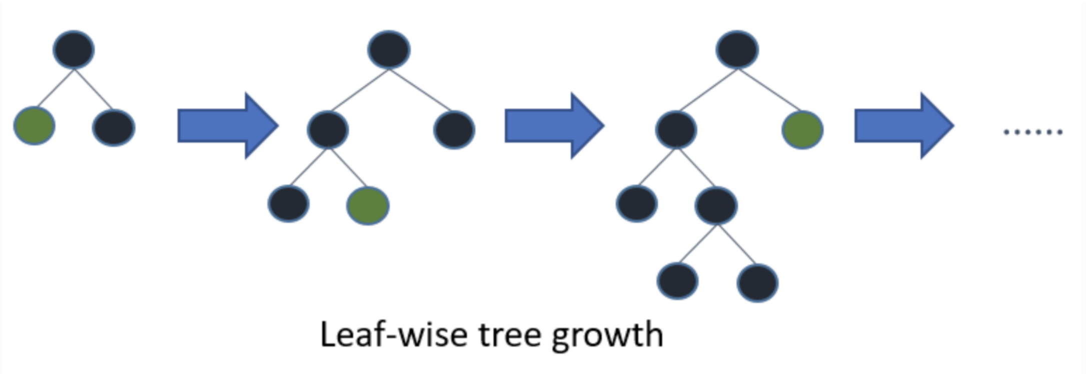


Fig: Simple Schematic Diagram for LightGBM

Softmax Cross Entropy (SCE) $= -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} y_{i,k} \cdot log(p_{i,k})$

$MLM8: ML_{model3} = (R^{1067\times14}, R^{481\times14}, F_8(F_3(x: \theta), O_x) = x, P_{\Theta}(D\_max), L_8 = SCE)$

MLM9:

$ML_{result3} = ML_{model3} \circ ML_{clean3} = (R^{1067\times14}, R^{481\times14}, F_9(F_3(x: \theta), O_x) = x, P_{\Theta}(D\_max), L_8 = SCE)$

Extra Tree Classifier

The third best model would be Extra Tree Classifier. Similar to Random Forests, ExtraTrees is an ensemble machine learning approach that trains numerous decision trees and aggregates the results from the group of decision trees to output a prediction. Instead of using bagging to select different variations of the training data to ensure decision trees are sufficiently different, Extra Trees uses the entire dataset to train decision trees. It introduces higher randomness by utilizing random thresholds for node splitting, eliminating the search for optimal thresholds. This approach enhances diversity among the trees, reducing correlation and potentially mitigating overfitting. The ExtraTrees have higher bias and lower variance than Random Forest, and it has better computational efficiency than Random Forest. There are three main parameters in the Extra Tree Classifier: K(max_feature), nmin(min_sample_leaf), and M(n_estimator). Since the Extra Tree Classifier mainly reduces the computational cost and our dataset is relatively small, that’s why the model performance might not be as good as the Random Forest model.

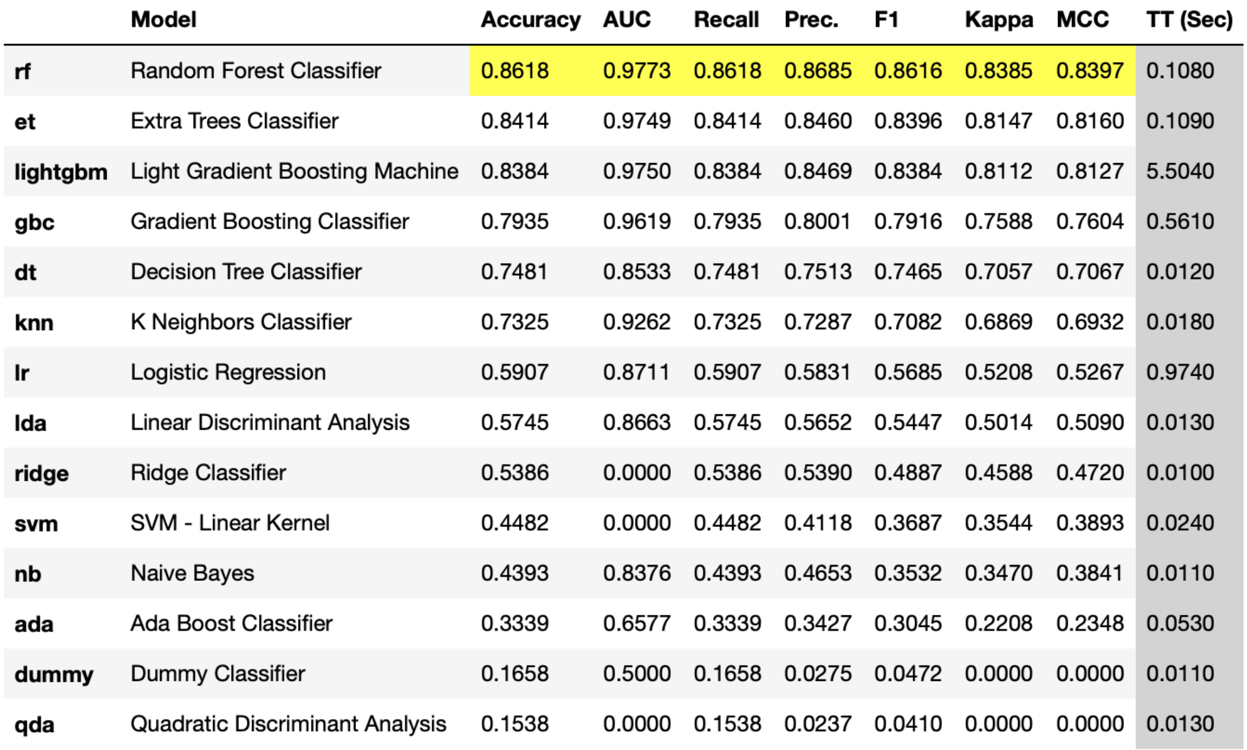
$$MLM10: ML_{model4} = (R^{1067\times14}, R^{481\times14}, F_{10}(F_3(x: \theta), O_x) = x, P_{\Theta}(K, nmin, M), L_6 = MAE)$$

MLM11:

$$ML_{result4} = ML_{model4} \circ ML_{clean3} = (R^{1067\times14}, R^{481\times14}, F_{11}(F_3(x: \theta), O_x) = x, P_{\Theta}(K, nmin, M), L_6 = MAE)$$

Model Selection – PyCaret

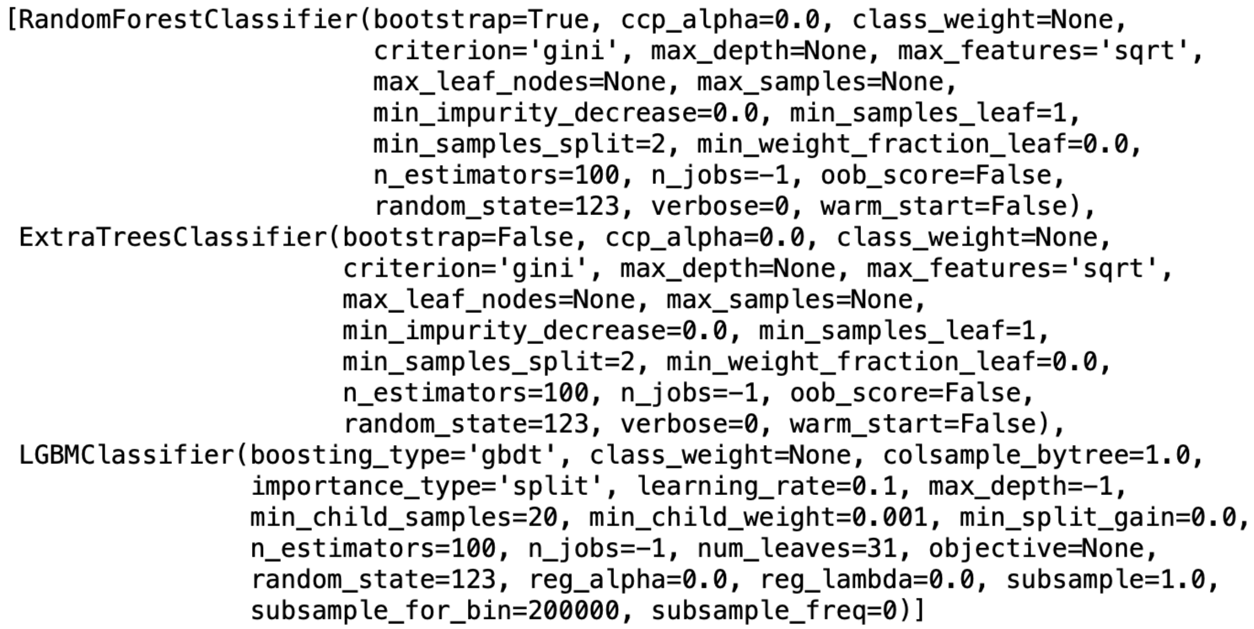
We are using an open-source, low-code machine learning library in Python that automates machine learning workflows which is called Pycaret for model selection. It is an end-to-end machine learning and model management tool that speeds up the experiment cycle exponentially. Pycaret provides an easy-to-use interface that automates various steps in the machine learning process, such as data preprocessing, model selection, hyperparameter tuning, model evaluation, and deployment. With the help of Pycaret, it can help me quickly select the best models and best features or hyperparameters in my model. It supports a wide range of algorithms to perform the tasks like model comparison or features engineering.



	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
rf	Random Forest Classifier	0.8618	0.9773	0.8618	0.8685	0.8616	0.8385	0.8397	0.1080
et	Extra Trees Classifier	0.8414	0.9749	0.8414	0.8460	0.8396	0.8147	0.8160	0.1090
lightgbm	Light Gradient Boosting Machine	0.8384	0.9750	0.8384	0.8469	0.8384	0.8112	0.8127	5.5040
gbc	Gradient Boosting Classifier	0.7935	0.9619	0.7935	0.8001	0.7916	0.7588	0.7604	0.5610
dt	Decision Tree Classifier	0.7481	0.8533	0.7481	0.7513	0.7465	0.7057	0.7067	0.0120
knn	K Neighbors Classifier	0.7325	0.9262	0.7325	0.7287	0.7082	0.6869	0.6932	0.0180
lr	Logistic Regression	0.5907	0.8711	0.5907	0.5831	0.5685	0.5208	0.5267	0.9740
lda	Linear Discriminant Analysis	0.5745	0.8663	0.5745	0.5652	0.5447	0.5014	0.5090	0.0130
ridge	Ridge Classifier	0.5386	0.0000	0.5386	0.5390	0.4887	0.4588	0.4720	0.0100
svm	SVM - Linear Kernel	0.4482	0.0000	0.4482	0.4118	0.3687	0.3544	0.3893	0.0240
nb	Naive Bayes	0.4393	0.8376	0.4393	0.4653	0.3532	0.3470	0.3841	0.0110
ada	Ada Boost Classifier	0.3339	0.6577	0.3339	0.3427	0.3045	0.2208	0.2348	0.0530
dummy	Dummy Classifier	0.1658	0.5000	0.1658	0.0275	0.0472	0.0000	0.0000	0.0110
qda	Quadratic Discriminant Analysis	0.1538	0.0000	0.1538	0.0237	0.0410	0.0000	0.0000	0.0130

Fig: Result of Model Comparison

The Pycaret multiclass classification model selection has shown that Random Forest has best accuracy. By using the predict_model function, we have used the test data for scoring and got the accuracy at 0.8618. Since the Random Forest gives us the best result, we have saved the entire pipeline. And we also have selected the top 3 highest accuracy models with our dataset. It gives us all the information of these three models (RandomForestClassifier, LGBMClassifier, ExtraTreesClassifier) including the result after tuning the hyperparameter.



```
[RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                        criterion='gini', max_depth=None, max_features='sqrt',
                        max_leaf_nodes=None, max_samples=None,
                        min_impurity_decrease=0.0, min_samples_leaf=1,
                        min_samples_split=2, min_weight_fraction_leaf=0.0,
                        n_estimators=100, n_jobs=-1, oob_score=False,
                        random_state=123, verbose=0, warm_start=False),
 ExtraTreesClassifier(bootstrap=False, ccp_alpha=0.0, class_weight=None,
                      criterion='gini', max_depth=None, max_features='sqrt',
                      max_leaf_nodes=None, max_samples=None,
                      min_impurity_decrease=0.0, min_samples_leaf=1,
                      min_samples_split=2, min_weight_fraction_leaf=0.0,
                      n_estimators=100, n_jobs=-1, oob_score=False,
                      random_state=123, verbose=0, warm_start=False),
 LGBMClassifier(boosting_type='gbdt', class_weight=None, colsample_bytree=1.0,
                importance_type='split', learning_rate=0.1, max_depth=-1,
                min_child_samples=20, min_child_weight=0.001, min_split_gain=0.0,
                n_estimators=100, n_jobs=-1, num_leaves=31, objective=None,
                random_state=123, reg_alpha=0.0, reg_lambda=0.0, subsample=1.0,
                subsample_for_bin=200000, subsample_freq=0)]
```

Fig: Top 3 highest accuracy model

Hyperparameter Tuning

1. Grid Search

We applied Grid Search on Random Forest. Grid Search is a tuning tool that helps us to test all the values of hyperparameters under a given range and output the performance evaluation for corresponding values of hyperparameters. For instance, in Random Forest, we considered hyperparameters including number of estimators *'n_estimators'*, maximum number of features in each splitting node *'max_features'*, etc. We set a given range for each hyperparameter for Grid search to test on. For *'n_estimators'*, we set a range from 100 to 1000, and we let Grid Search run the model while trying all the values in this given range, and find the best performing one.

2. Random Search

Random search is a tuning technique that randomly selects combinations of hyperparameters from a given range. Unlike grid search, which exhaustively tries all possible parameter combinations, random search samples a subset of combinations.

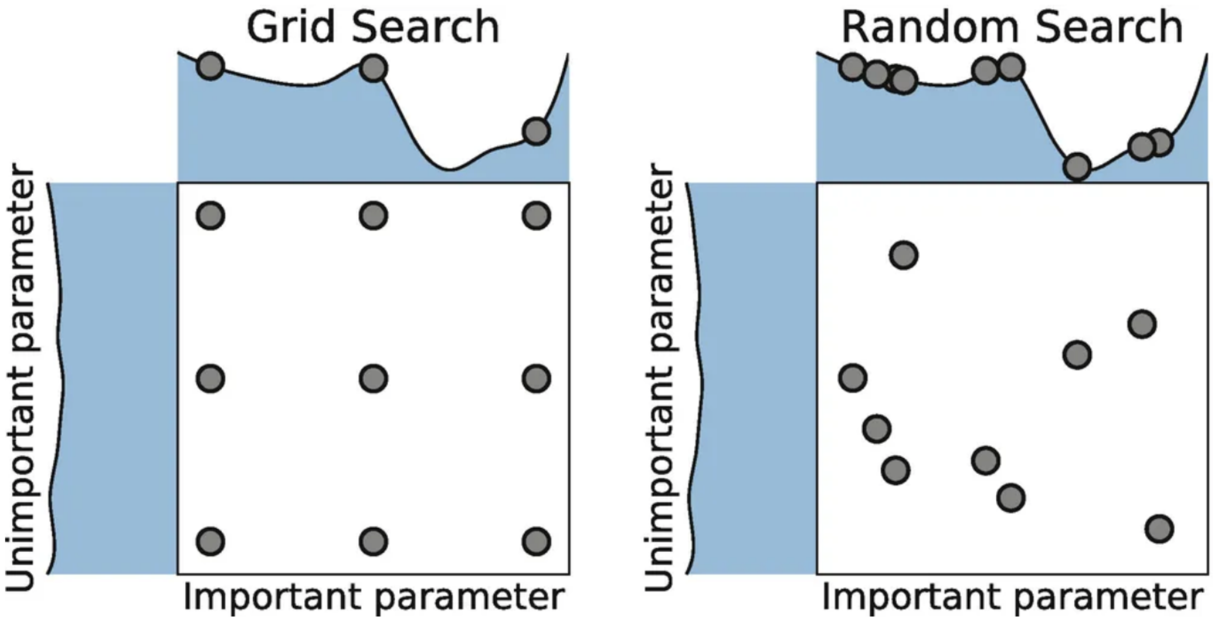


Fig: Diagram for Grid Search and Random Search Algorithm

3. Bayes Optimization

Bayesian Optimization is a strategy for optimizing objective functions by building a probabilistic model of the function and using it to select the most promising points to evaluate. It starts by building a guessing model, normally Gaussian, to predict and to guess how different hyperparameters improve the model performance. After evaluating the performance, it will give feedback to the guessing model and update it until receiving the best hyperparameter.

Leave One Out Cross-Validation (LOOCV)

For our KNN model, We use Leave One Out Cross-Validation(LOOCV) for cross validation and finding the best k in our model. We first split a dataset into a training set and a testing set, using all but one observation as part of the training set. We use the model to predict the response value of the one observation left out of the model and find the accuracy value. At the same time we will find the K_value in each iteration. Repeat this process and get the average accuracy score and the best K_value.

Nested Cross-Validation

(Outer – Cross Validation; Inner – Feature Selection & Tuning Hyperparameter)

Nested Cross-Validation is an approach that evaluates model performance with consideration of the result hyperparameter tuning feature selection inside each model evaluation. It is a complex Cross-Validation with a inner loop and a outer loop, where the inner loop process k-fold Cross-Validation for features selection & hyperparameter tuning, whereas the outer loop train entire inner loop, and test models and their corresponding hyperparameters separately in k-fold Cross-Validation. Instead of using Cross-Validation, we use Nested Cross-Validation as it provides unbiased model performance estimation and robust hyperparameter tuning and features selection. In each outer loop, we split the data into train set and test set, then use the train data to evaluate the hyperparameters and features in the inner loop and generate the result with the best features and hyperparameter. Then, we go to the outer loop and re-train using the best configuration and report the accuracy. We repeat the same process until we finish. By the end of the process, we'll have a table that reports the estimated generalization error for each model, and compare the result to find the best model.

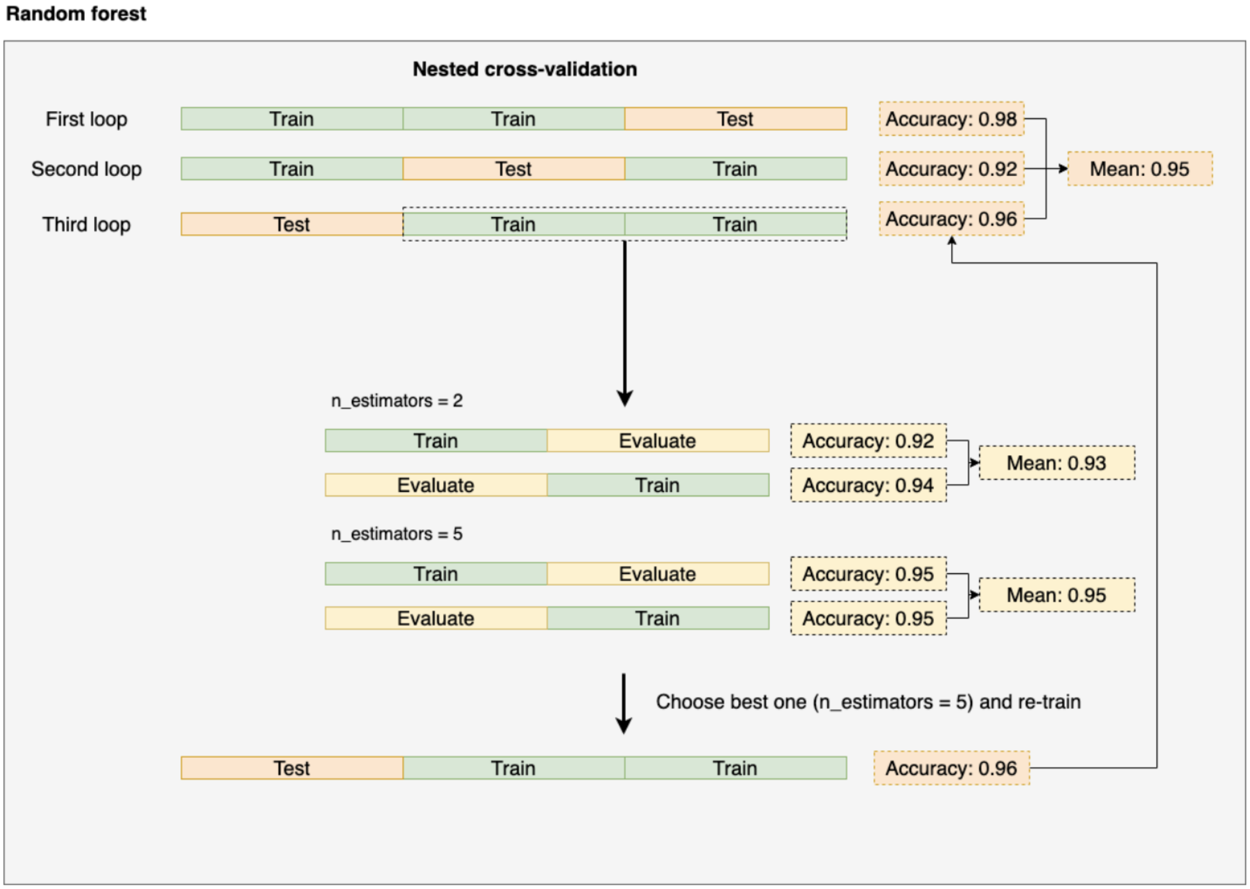
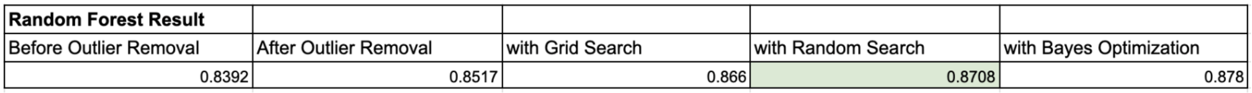


Fig: Sample Nested Cross Validation for Random Forest Hyperparameter Tuning

- Result and insight -

Random Forest

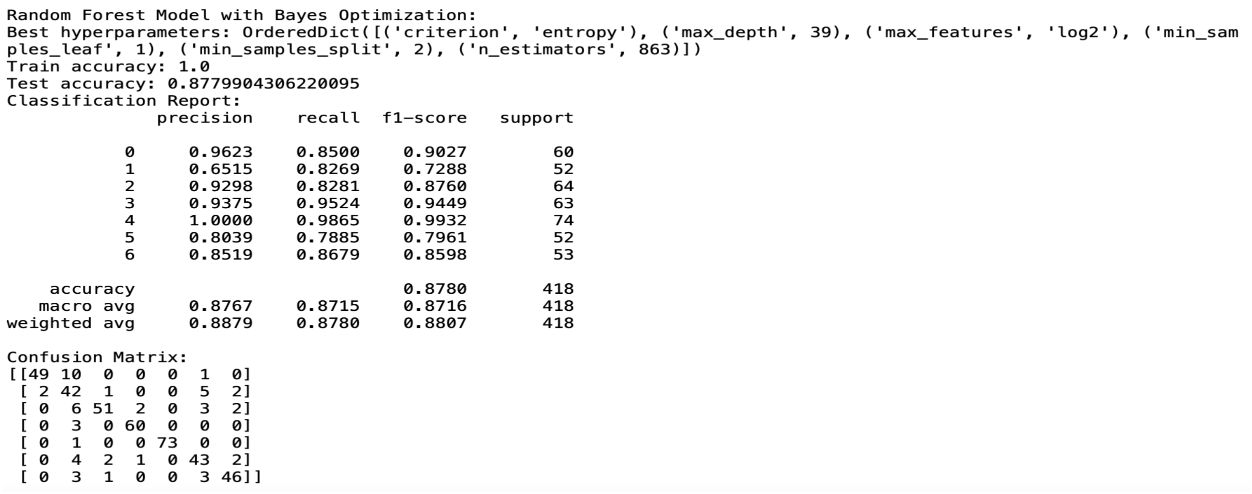
For Random Forest, our model with all data sets displays a testing accuracy of 0.8392, and the model with outliers removed shows a score of 0.8517. The accuracy of Grid search, Random search, and Bayes Optimization are 0.8660, 0.8708, and 8.8780 respectively. Among all six models, the Random Forest with Bayes Optimization model has the best accuracy score.



Random Forest Result				
Before Outlier Removal	After Outlier Removal	with Grid Search	with Random Search	with Bayes Optimization
0.8392	0.8517	0.866	0.8708	0.878

Fig: Random Forest Result

From the graph above, we see that outlier removal and hyperparameter tuning did improve the overall performance. The f1-score, which is a measure of test accuracy with consideration of both precision and recall, shows the model performance with respect to each class in our target. Bayes Optimization indicates that the best maximum number of features in each split is the log2 of the total number of features, and the best number of trees is 863. As the figures show, only class 1 and 5 have a relatively small f1-score. Based on the classification report of Bayes Optimization, as the graph shown below, Class 4 has the best performance. It has a perfect precision of 1.0000, which means all the class 4 predictions are correct. A recall of 0.9865 indicates that 98.65% of actual data are predicted correctly as class 4. Only one class 4 instance is mistakenly classified as class 1, as shown in the confusion matrix. Whereas, class 1 has the lowest F1-score of 0.7288, which suggests that the model is least effective at classifying this class compared to others. The precision of 0.6515 indicates that 65.15% of class 1 predictions are correct, and the recall of 0.8269 indicates that 82.69% of actual data are correctly predicted as class 1. As shown in the confusion matrix, only 42 cases are identified correctly , and 10 cases are classified incorrectly.

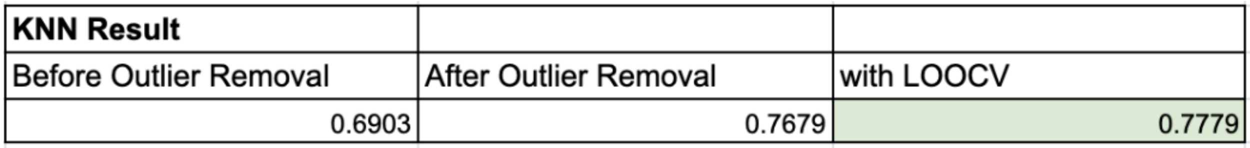


```
Random Forest Model with Bayes Optimization:
Best hyperparameters: OrderedDict([('criterion', 'entropy'), ('max_depth', 39), ('max_features', 'log2'), ('min_sam
ples_leaf', 1), ('min_samples_split', 2), ('n_estimators', 863)])
Train accuracy: 1.0
Test accuracy: 0.8779904306220095
Classification Report:
              precision    recall  f1-score   support

           0     0.9623    0.8500    0.9027        60
           1     0.6515    0.8269    0.7288        52
           2     0.9298    0.8281    0.8760        64
           3     0.9375    0.9524    0.9449        63
           4     1.0000    0.9865    0.9932        74
           5     0.8039    0.7885    0.7961        52
           6     0.8519    0.8679    0.8598        53

    accuracy                         0.8780       418
   macro avg     0.8767    0.8715    0.8716       418
weighted avg     0.8879    0.8780    0.8807       418

Confusion Matrix:
[[49 10  0  0  0  1  0]
 [ 2 42  1  0  0  5  2]
 [ 0  6 51  2  0  3  2]
 [ 0  3  0 60  0  0  0]
 [ 0  1  0  0 73  0  0]
 [ 0  4  2  1  0 43  2]
 [ 0  3  1  0  0  3 46]]
```

Fig: Result of Random Forest with Bayes Optimization

KNN

For the KNN, we also made models with all data, outliers removal, and Leave One Out Cross-Validation, with accuracy scores of 0.6903, 0.7679, 0.7779 respectively, as figures shown below.



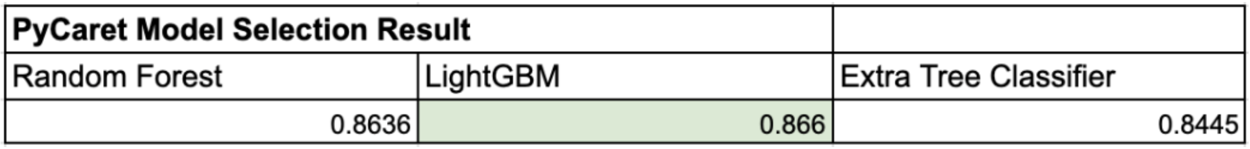
KNN Result		
Before Outlier Removal	After Outlier Removal	with LOOCV
0.6903	0.7679	0.7779

Fig: Testing results of KNN

With LOOCV, we also find the best K in our model, which is 1. That means that the prediction for a new data point will be based only on the single nearest neighbor. The LOOCV accuracy score represents the accuracy of the model when trained on all data except for one point and then tested on that single point. We have an average accuracy of 0.7779 across all datasets.

PyCaret Result

Using the Hyperparameters that were generated by the PyCaret, we got the result as Random Forest model with accuracy score is 0.8636, LightGBM model with accuracy score is 0.866, and Extra Tree Classifier with accuracy score is 0.8445. The result is slightly different from the model selection results that were generated by PyCaret in our previous section. One possible explanation can be the Random Forest algorithm involves randomness in their training process, because of the random weight initialization. Due to this randomness, the results might vary between runs, causing fluctuations in performance.



PyCaret Model Selection Result		
Random Forest	LightGBM	Extra Tree Classifier
0.8636	0.866	0.8445

Fig: Result of KNN with Random Search

Looking at the confusion matrix for these three models. For all the models, class 4 has the best performance, and followed by class 3, class2, and class 0. And class 1 has the worst performance. It is consistent with our Random Forest with Bayes Optimization Model. Class 4 is Obesity_Type_III (with the highest BMI), and Class 1 is Normal weight (with the second lowest BMI). This might be incurred by the separability of classes. As Obesity_Type_III may have more distinctive features that are easier for the model to learn and recognize. Classes in which individuals with high BMI might have less overlapping with other classes, leading to clearer decision boundaries. Whereas people who are normal weighted may not share distinctive common features.

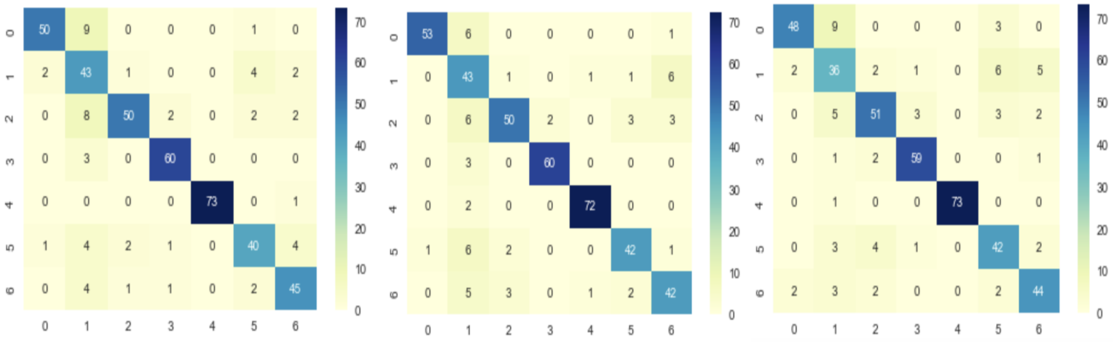
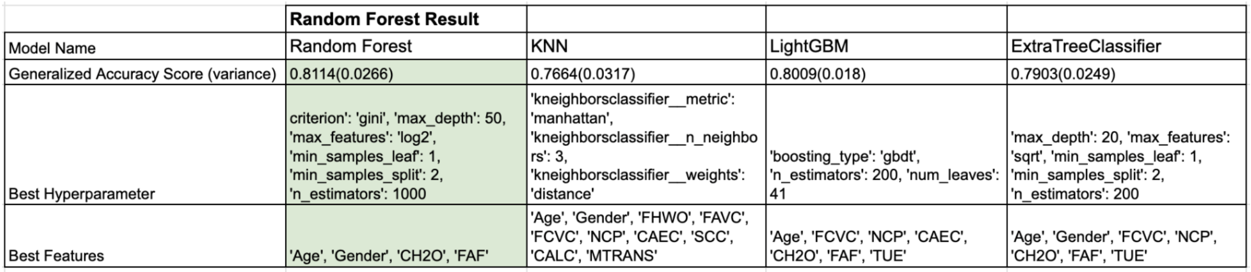


Fig: Confusion Matrix of Random Forest (Left), LightGBM (Mid), ExtraTreeClassifier (Right)

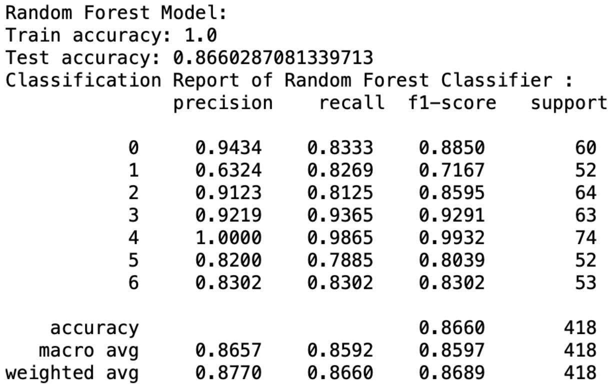
Nest Cross_Validation



	Random Forest Result			
Model Name	Random Forest	KNN	LightGBM	ExtraTreeClassifier
Generalized Accuracy Score (variance)	0.8114(0.0266)	0.7664(0.0317)	0.8009(0.018)	0.7903(0.0249)
Best Hyperparameter	criterion': 'gini', 'max_depth': 50, 'max_features': 'log2', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 1000	'kneighborsclassifier__metric': 'manhattan', 'kneighborsclassifier__n_neighbors': 3, 'kneighborsclassifier__weights': 'distance'	'boosting_type': 'gbdt', 'n_estimators': 200, 'num_leaves': 41	'max_depth': 20, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200
Best Features	'Age', 'Gender', 'CH2O', 'FAF'	'Age', 'Gender', 'FHWO', 'FAVC', 'FCVC', 'NCP', 'CAEC', 'SCC', 'CALC', 'MTRANS'	'Age', 'FCVC', 'NCP', 'CAEC', 'CH2O', 'FAF', 'TUE'	'Age', 'Gender', 'FCVC', 'NCP', 'CH2O', 'FAF', 'TUE'

Fig: Result of Models Nested Cross-Validation

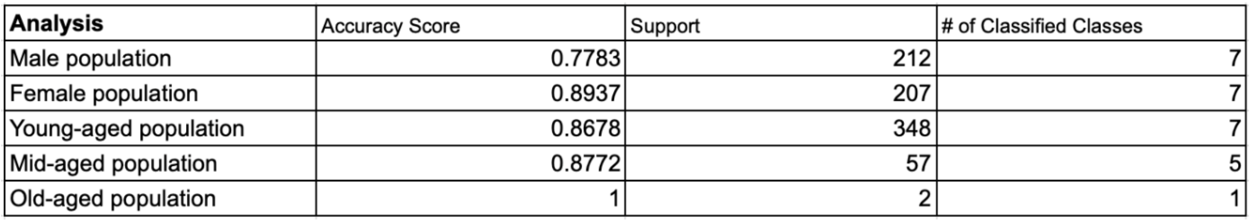
Random ForestNested Cross-Validation outputs an average accuracy of 0.8114, which ranked the highest among all 4 models. We also used it to generate the best parameter and best feature for each model. Using the best models with the best features and best parameters, we are getting a 0.866 accuracy score.



```
Random Forest Model:
Train accuracy: 1.0
Test accuracy: 0.8660287081339713
Classification Report of Random Forest Classifier :
               precision    recall  f1-score   support

           0     0.9434    0.8333    0.8850        60
           1     0.6324    0.8269    0.7167        52
           2     0.9123    0.8125    0.8595        64
           3     0.9219    0.9365    0.9291        63
           4     1.0000    0.9865    0.9932        74
           5     0.8200    0.7885    0.8039        52
           6     0.8302    0.8302    0.8302        53

    accuracy                         0.8660       418
   macro avg     0.8657    0.8592    0.8597       418
weighted avg     0.8770    0.8660    0.8689       418
```

Fig: Result of Best Model

Insight



Analysis	Accuracy Score	Support	# of Classified Classes
Male population	0.7783	212	7
Female population	0.8937	207	7
Young-aged population	0.8678	348	7
Mid-aged population	0.8772	57	5
Old-aged population	1	2	1

Fig: Random Forest model's results with different groups of population

From the feature selection in the nested cross validation, we noticed that gender and age are the most noticeable features. We are using those with our model to analyze our data deeper. For gender, we are using the male population and female population to rerun our best model. The two gender populations have comparable numbers of data, so we don’t need to consider the problem of imbalance. In terms of the accuracy score, the female population has a much higher score than male population. By the review on the global gender disparities in obesity, more women are obese than men overall. Thus the female population resulting in higher accuracy score has the similar reason as our analysis on confusion matrix. When women are more obese than men, the percentage of high BMI score in the female population. Classes where individuals with a higher BMI may present more distinct characteristics, resulting in more defined decision boundaries. Because of that, females may have a better prediction. Also, we have created a PCA plot to visualize and compare the variance and class separations between male and female populations. From the plot, we have noticed that the female points are closer together and male points are more spread out, it would suggest more variance in the male data. Even though the number of samples in each group is fairly balanced between males and females, the male population data may have more variance or less distinct class separations, making it harder for the model to achieve high accuracy.

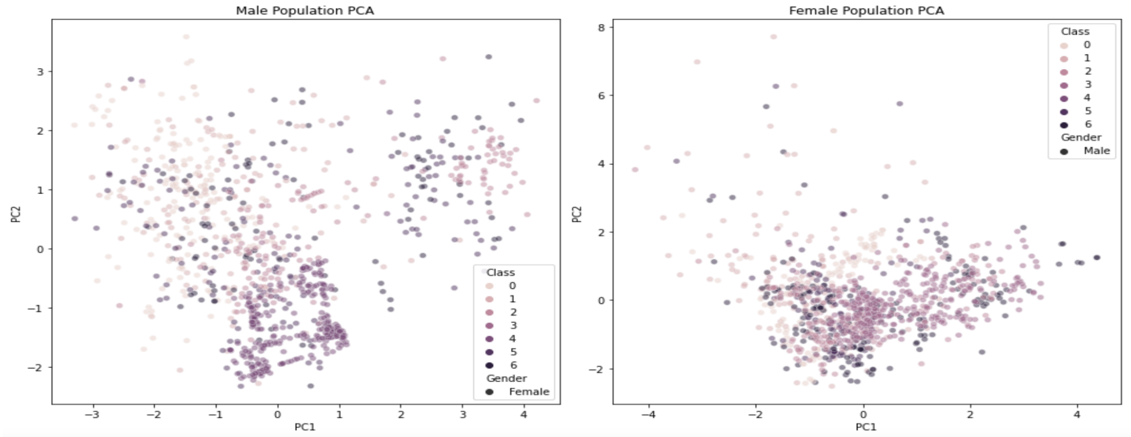


Fig: PCA Male Population (left) and CA Female Population (right)

For the age, we are using the definition of Human Age Group Classification to classify our data into three groups: Young-aged population, Mid-aged population, and Old-Aged population. Less than 30 yrs old is classified as young-aged adults, the age is between 31 ~ 45 yrs old is mid-aged adults, and the rest are old-aged adults. Most of our dataset is collected from the young-aged group, the number of samples in each group is unfairly imbalanced among these age groups. Only two samples are in the old-aged group. Beside the old-aged group, the accuracy score between the young-aged group and the mid-aged group is pretty close. This indicates that features relevant to obesity or the condition being predicted are more consistent and distinct among young adults and mid-aged adults, leading to clearer patterns that the model can learn. However, since the mid-aged group and old-aged group have the smallest sample size, the model might be capturing noise as patterns due to the lower complexity of the data. With fewer classes represented in the mid-aged and old-aged groups, the model has fewer distinctions to make, which might artificially inflate accuracy scores.

- Source Code -

https://github.com/simiy1/predicting_obesity

- Conclusions -

In this project, we build four models to determine people’s obesity level based on their eating habits and physical conditions. Even though the accuracy score is relatively high, there are some issues about the data that have been raised. For example, as we have discussed in the result and insight part, for the age population. The mid-aged group and old-aged group have small populations, especially for the old-aged group, it may not represent the true distribution of the condition, leading to less reliable accuracy scores. In order to draw more reliable conclusions with the aged group, gathering more data for the mid-aged and old-aged groups is helpful to ensuring that there is a balanced representation of all classes. The other thing is that the dataset contains the information of people from the countries of Mexico, Peru and Colombia. However, the dataset doesn't include the geographic location information. Since biological and environmental factors can play a crucial role in obesity, missing geographic location information is preventing us from diving deeper on the analysis of the determination of obesity.

In general, our model can classify different obesity levels based on the input of people’s eating habits and their physical conditions mostly. It is very helpful when people want to learn about their body health and be aware of their weight control. With the help of our model, it can help identify individuals at higher risk of obesity based on the physical condition, eating habits or lifestyle. With early identification, public health programs can educate people on healthy eating and exercise so that they reduce the chance of developing obesity issues and the cost of treating obesity.

Our model employs efficient methodologies and technologies to generate valuable insights into the relationship between eating habits, physical condition, and the risk of obesity. There are some new views of our projects: 1. mobile health applications, and telemedicine can play a role in monitoring, preventing, and managing obesity; 2. obesity not just as a local or national issue, the model can be discussed more in the global health level; 3. Future studies on these features and discover some potential research with different other fields, such as nutrition and epidemiology.

Reference:

Kho, J. (2019, March 12). Why random forest is My Favorite Machine Learning Model. Medium. https://towardsdatascience.com/why-random-forest-is-my-favorite-machine-learning-model-b97651fa3706

Palechor, F. M., &amp; Manotas, A. de. (2019). Dataset for estimation of obesity levels based on eating habits and physical condition in individuals from Colombia, Peru and Mexico. Data in Brief, 25, 104344. https://doi.org/10.1016/j.dib.2019.104344

Centers for Disease Control and Prevention. (2022, September 24). Health effects of overweight and obesity. Centers for Disease Control and Prevention. https://www.cdc.gov/healthyweight/effects/index.html

What is the K-nearest neighbors algorithm?. IBM. (n.d.). https://www.ibm.com/topics/knn#:~:text=Next%20steps-,K%2DNearest%20Neighbors%20Algorithm,of%20an%20individual%20data%20point

K, D. (2021, April 9). Anomaly detection using isolation forest in python. Paperspace Blog. https://blog.paperspace.com/anomaly-detection-isolation-forest

Gupta, R. (2023, May 6). The power of Crosstab function in pandas for data analysis and visualization. Medium. https://medium.com/geekculture/the-power-of-crosstab-function-in-pandas-for-data-analysis-and-visualization-6c085c269fcd#:~:text=One%20of%20the%20most%20useful,the%20relationships%20between%20the%20variables.

PyCarey - Home. PyCaret. (2023, June 28). https://pycaret.org/

Brownlee, J. (2021, November 19). *Nested cross-validation for Machine Learning with python*. MachineLearningMastery.com. https://machinelearningmastery.com/nested-cross-validation-for-machine-learning-with-python/

Mandot, Pushkar. “What Is LIGHTGBM, How to Implement It? How to Fine Tune the Parameters?” *Medium*, Medium, 1 Dec. 2018, medium.com/@pushkarmandot/https-medium-com-pushkarmandot-what-is-lightgbm-how-to-implement-it-how-to-fine-tune-the-parameters-60347819b7fc.

Kanter, Rebecca, and Benjamin Caballero. "Global gender disparities in obesity: a review." Advances in nutrition 3.4 (2012): 491-498.

Brownlee, Jason. “LOOCV for Evaluating Machine Learning Algorithms.” *MachineLearningMastery.Com*, 26 Aug. 2020, machinelearningmastery.com/loocv-for-evaluating-machine-learning-algorithms/.

“Welcome to LIGHTGBM’s Documentation!.” *Welcome to LightGBM’s Documentation! - LightGBM 4.0.0 Documentation*, lightgbm.readthedocs.io/en/stable/. Accessed 10 Dec. 2023.

Bhat, D., and V. K. Patil. "Human Age Group Classification Using Facial Features." International Journal of Modern Trends in Engineering and Research 3.6 (2016): 123-132.

Thankachan, Karun. “What? When? How?: Extratrees Classifier.” *Medium*, Towards Data Science, 9 Aug. 2022, towardsdatascience.com/what-when-how-extratrees-classifier-c939f905851c#:~:text=ExtraTrees%20Classifier%20is%20an%20ensemble,(compared%20to%20Random%20Forest).

Wang, Wei. “Bayesian Optimization Concept Explained in Layman Terms.” *Medium*, Towards Data Science, 22 Mar. 2022, towardsdatascience.com/bayesian-optimization-concept-explained-in-layman-terms-1d2bcdeaf12f.