CSE422: Artificial Intelligence

Project Name: Obesity Level Detection

Group: 8

Fahad Al Shahid

ID: 24141187

Lab Section:4

Md. Naimuzzaman Fahim

ID: 21201140

Lab Section:4 (Theory Section: 5)

Table of Contents

1. Introduction:

Obesity is one of the largest growing public health concerns in the world. It affects people's lives by increasing the risk of many chronic diseases. So in our project, we wanted to develop a model that can detect the obesity level of an individual's lifestyle and health-related factors. So the motivation behind the project is to detect obesity at an early stage. So he or she can change his lifestyle and health. So he can have a better life and prevent many chronic diseases.

2. Dataset Description

- **Source:** https://www.kaggle.com/datasets/jpkochar/obesity-risk-dataset/data
- **Dataset Description:**

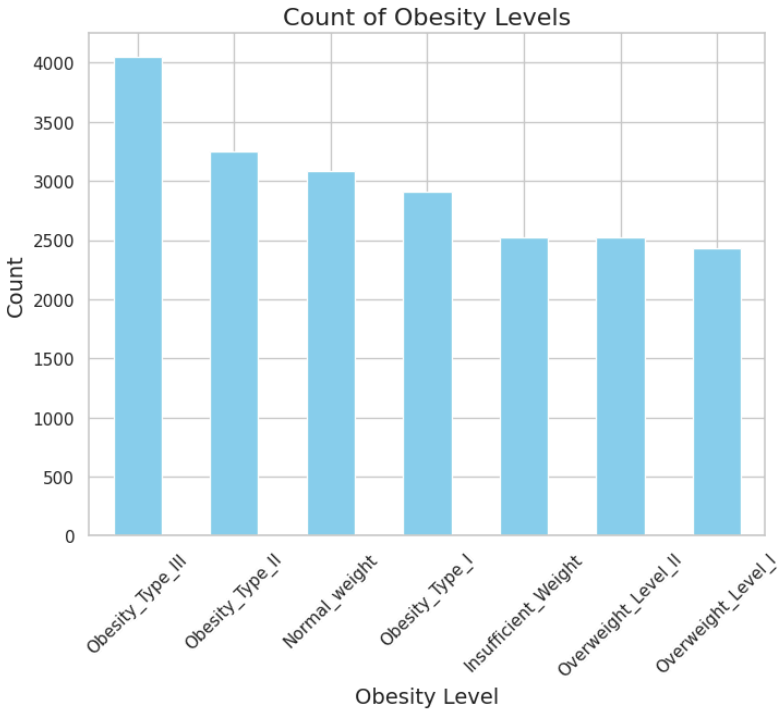
  There are 12 features in the dataset.

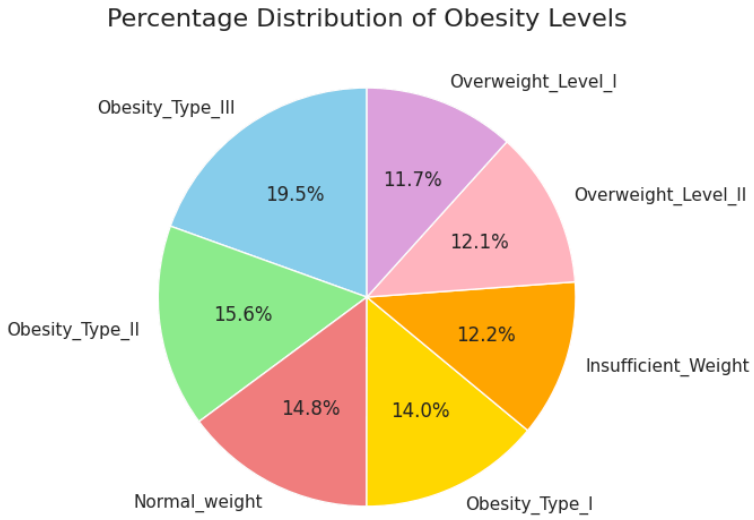
  Features:

  - **id:** Unique ID of the person
  - **Gender:** Is the person Male or Female? (Male / Female)
  - **Age:** Age of the person.
  - **Height:** Height of the person.
  - **Weight:** Weight of the person.
  - **family_history_with_overweight:** Is there any person in the family with an overweight issue? (1=Yes / 0=No)
  - **Frequent consumption of high-caloric food:** Does the person frequently eat high-caloric food? (1=Yes / 0=No)
  - **Number of main meals:** How many times does the person eat daily?
  - **SMOKE:** Does the person smoke or not? (Yes / No)
  - **Consumption of alcohol:** How many times does the person drink alcohol? (Sometimes / Never / Frequently)
  - **transportation:** Which types of transportation does the person use? (Public_Transportation / Automobile / Walking / Motorbike / Bike)
  - **Obesity_level:** What is the person's level of obesity? (Overweight_Level_II / Normal_weight / Insufficient_Weight / Obesity_Type_III / Obesity_Type_II / Overweight_Level_I / Obesity_Type_I)

    This is the target variable.

- **Imbalanced Dataset:**
    - The classes in the `Obesity_level` feature do not have an equal number of instances. The distribution varies, with the most frequent class, `Obesity_Type_III` (19.49%), and the least frequent class, `Overweight_Level_I` (11.69%).

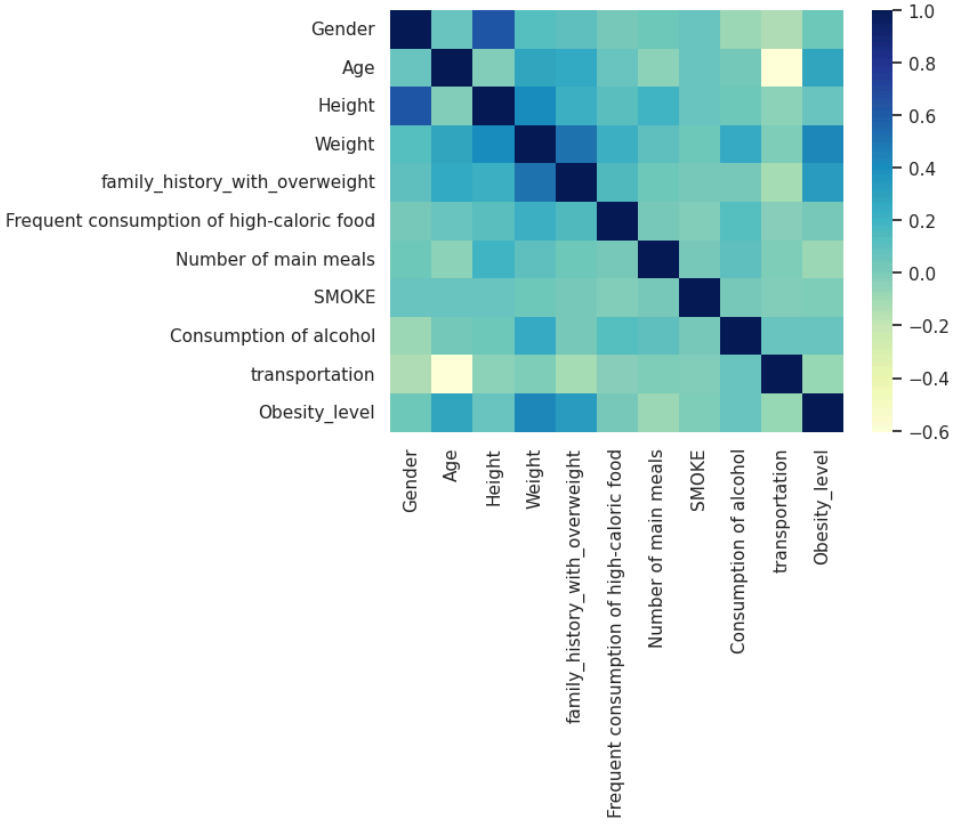




3. Data Pre-processing

- **Faults:**
    - There are 2 columns, "**Number of main meals"** and **"transportation,"** which have null values of **200** and **25**, respectively.
    - There are 5 columns, **"Gender"**, **"SMOKE"**, **"Consumption of alcohol"**, **"transportation"**, and **"Obesity_level"**, which have categorical data.
- **Solutions:**
    - Simply we drop the rows of the null value since the "**Number of main meals"** is **0.96%** and **"transportation"** is **0.12%** which are a very small percentage to consider.
    - So for the **"Gender"**, **"SMOKE"**, **"Consumption of alcohol"**, **"transportation"**, and **"Obesity_level"**, we have to use a Level encoder. Which encodes every unique categorical value to an integer starting from 0.

Heatmap



4. Feature Scaling

We had to feature scaling to ensure faster convergence, numerical stability, and unbiased learning for the Logistic regression Model. It also improves the accuracy of the model.

5. Dataset Splitting

- Random
- Train set (70%)
- Test set (30%)

6. Model Training & Testing

We split the data 70 for training and 30 for testing using the random train_test_split from the Sklearn library. Then we fit the training data to the Decision tree, KNN, and logistic regression. And predict the test data from the model. predict function and run an accuracy test.

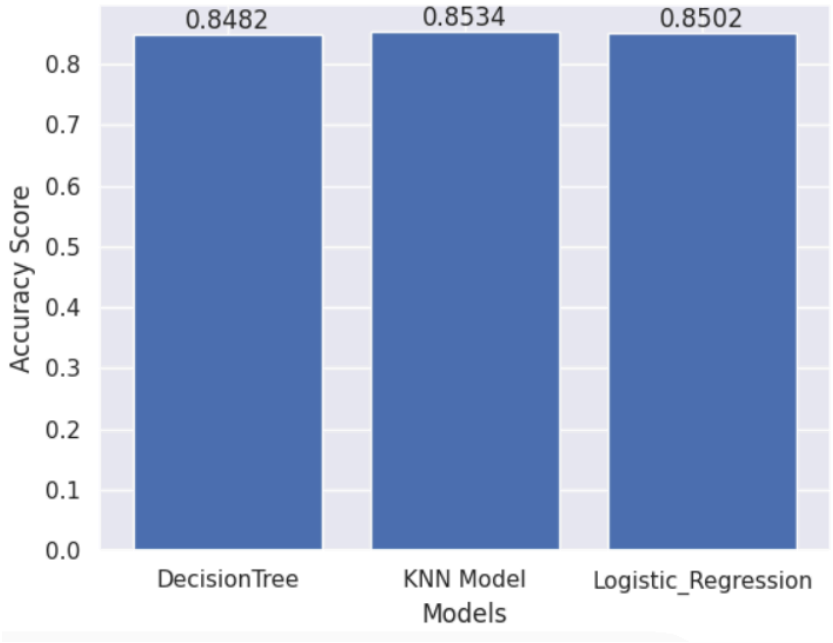
7. Model Selection

This is a classification problem so we choose 3 model

- **Decision tree:** We have loaded the Decision Tree Classifier from Sklearn library and defined the DecisionTreeClassifier and trained with the X_train and y_train datasets. Then test the model using the X_test dataset.
- **KNN:** we have loaded the KNN algorithm KNeighborsClassifier() and used the classifier knn.fit(X_train,y_train) to train the model. then we used knn.predict(X_test) to test the dataset
- **Logistic Regression:** we loaded the Logistic Regression and defined LogisticRegression and trained with the X_train_std (scaled data) and y_train dataset. Then test the model using the X_test_std (scaled data)dataset.

8. Conclusion

Model Name	Accuracy rate
Decision tree	84.82%
KNN	85.34%
Logistic regression	85.02%



So we can see from the model **Decision tree, KNN,** and **Logistic regression.** The **KNN** model has the highest accuracy rate.