**Assignment Cover Page**

|  |  |
| --- | --- |
| **Subject Code** | COSC2999 |
| **Subject Name** | Practical Data Science with Python |
| **Location** | SGS |
| **Title of Assignment** | Assignment 2: Data Modelling |
| **Student Name** | Nguyễn Đình Đăng Nguyên |
| **Student Number** | S3759957 |
| **Lecturer Name** | DR. Thuy Nguyen |
| **Assignment due date** | December 2nd, 2024 |
| **Assignment submission date** | December 2nd, 2024 |
| **Numbers of pages including this one** | 15 |
| **Word count** |  |

*I declared that in submitting all work for this assessment I have read, understood and agree to the content and expectations of the Assessment Declaration*

# Abstract

Obesity is a medical condition characterized by an excessive accumulation of body fat, which can negatively impact an individual’s health. This study aimed to classify obesity levels using two machine learning approaches (Classification and Regression), with a focus on physical activity, nutritional habits, and genetic factors. The study utilized an observational design, collecting data from the UCI repository [ref] through a web-based survey to assess participants' eating habits and physical activity levels. The dataset included variables such as gender, age, height, weight, family history of obesity, dietary patterns, and physical activity frequency. For the modeling process, three classification algorithms were employed to predict obesity levels, including Random Forest (for both classification and regression tasks), Extreme Gradient Boosting (for classification), and k-nearest neighbors with grid search optimization for hyperparameter tuning. Model performance was evaluated using various metrics: accuracy, recall, precision, F1-score, area under the curve (AUC), and precision-recall curve for classification tasks, and Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R²) for regression tasks. The Random Forest model exhibited the most robust performance, and feature selection was shown to improve model efficiency. These findings emphasize the critical role of physical activity and nutritional habits in addressing the growing obesity epidemic.

# Introduction

The rise in obesity has become a significant public health concern worldwide. According to WHO, over the past few decades, obesity rates have steadily increased, driven by factors such as genetics, environment, diet, physical activity levels, and behaviors [ref]. Poor diet (high in calories), lack of physical activity, genetics, and certain medical conditions or medications can all contribute to obesity.

Addressing obesity is not only essential for individual well-being but also for reducing the overall burden on healthcare systems and improving societal health outcomes. Immediate and sustained efforts to prevent obesity are necessary to ensure healthier futures for individuals and communities around the world. Preventing obesity is crucial to reversing these trends and improving public health. Prevention strategies should focus on promoting healthier eating habits, increasing physical activity, and fostering environments that support healthier choices.

This report will outline the efficiency of using Machine Learning methodologies in detecting and preventing obesity based variables such as gender, age, height, weight, family history of obesity, dietary patterns, and physical activity frequency provided by UCI’s “Estimation of Obesity Levels Based On Eating Habits and Physical Condition” dataset [ref]

# Methodology

## Data Preparation

The dataset used in this study, obtained from UCI [ref], contains information on obesity levels among individuals from Mexico, Peru, and Colombia. The participants, aged between 14 and 61, balanced between 2 genders, represent a wide range of dietary habits and physical conditions. The dataset includes 2,111 records, consisting of 16 variables serving as inputted features and 1 as the outputted target. Each of the feature descriptions as listed as follows:

|  |  |
| --- | --- |
| **Features** | **Description** |
| Gender | Gender |
| Age | Age |
| Height | Height |
| Weight | Weight |
| family\_history\_with\_overweight | Has a family member suffered or suffers from overweight? |
| FAVC | Do you eat high caloric food frequently? |
| FCVC | Do you usually eat vegetables in your meals? |
| NCP | How many main meals do you have daily? |
| CAEC | Do you eat any food between meals? |
| SMOKE | Do you smoke? |
| CH2O | How much water do you drink daily? |
| SCC | Do you monitor the calories you eat daily? |
| FAF | How often do you have physical activity? |
| TUE | How much time do you use technological devices such as cell phone, videogames, television, computer and others? |
| CALC | How often do you drink alcohol? |
| MTRANS | Which transportation do you usually use? |

The target of the study, obesity levels (NObeyesdad), is commonly calculated using **Body Mass Index (BMI)**, a simple and widely used measure that helps categorize individuals based on their weight relative to their height. BMI is calculated using the following formula [ref]:

Where:

* Weight is in kilograms (kg)
* Height is in meters (m)

After all calculation was made to obtain the mass body index for each individual, the results were classified as [ref]:

* 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

## Exploratory data analysis (EDA)

## Feature analysis and selection

The dataset was divided into training and testing sets using an 8:2 ratio, while preserving the class distribution in both subsets.

## Machine learning algorithms

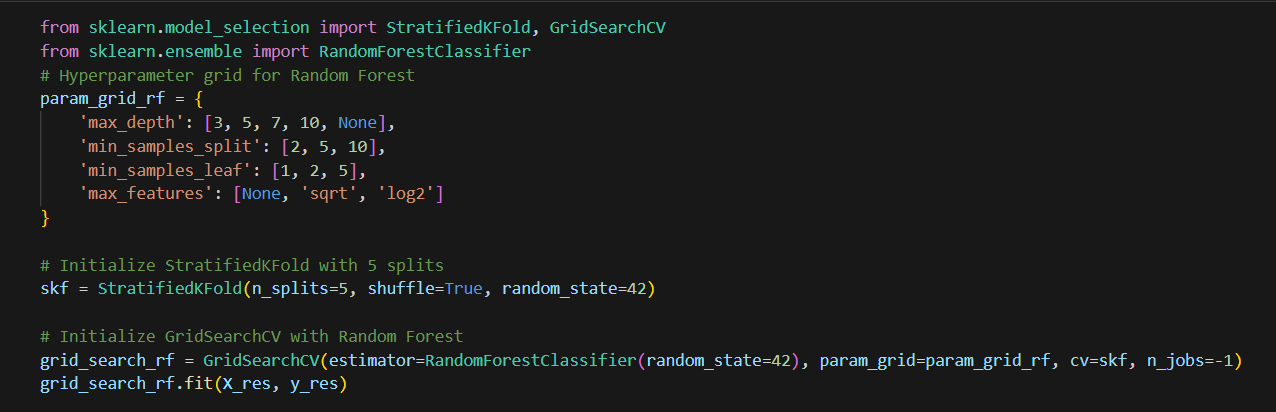
In this study, state-of-the-art Random Forest, K-nearest neighbors (KNN) and Extreme gradient boosting (XGBoost) algorithms were employed, to predict obesity levels based on various demographic, lifestyle, and health-related features.

**Random Forest** is an ensemble learning algorithm that combines multiple decision trees to improve classification accuracy and reduce overfitting [ref]. It works by generating a collection of decision trees through bootstrapped sampling of the training data, where each tree is trained on a random subset of features. When making predictions, the Random Forest model aggregates the outputs of all the individual trees by majority voting (for classification) or averaging (for regression). This method helps in mitigating the overfitting problem commonly seen with individual decision trees. Random Forest is particularly effective in handling large datasets with numerous features (particularly in this dataset, 2111 records with 17 features), and it can manage missing values and maintain accuracy even when a significant proportion of data is absent. Its ability to generalize well and its robustness to outliers and noise make it a popular choice for many classification and regression tasks. This model is employed in classification and regression tasks.

**XGBoost (eXtreme Gradient Boosting)** is a highly efficient and scalable machine learning algorithm, renowned for its exceptional performance in classification problems [ref]. Unlike Random Forest, which builds multiple trees independently, XGBoost builds trees sequentially, with each new tree designed to correct the errors made by the previous ones [ref. This boosting technique enhances the predictive power of the model by focusing on difficult-to-classify data points. XGBoost employs gradient boosting, where the model minimizes a loss function by updating the weights of the trees in a direction that reduces prediction errors. Additionally, it incorporates regularization techniques to prevent overfitting, making it well-suited for tasks involving complex, high-dimensional data. XGBoost’s speed, accuracy, and flexibility in handling different data types and distributions have made it a popular choice in machine learning competitions and real-world applications. This model is employed in classification tasks.

**K-Nearest Neighbors (KNN)** is a simple and intuitive classification algorithm that assigns a class label to a data point based on the majority class of its nearest neighbors [ref]. The model works by calculating the distance (typically Euclidean distance) between the test point and all training samples. It then identifies the 'k' nearest neighbors and predicts the class label that appears most frequently among these neighbors. The value of 'k' is a user-defined parameter, and its selection can significantly influence the model's performance, as a small 'k' can lead to overfitting, while a large 'k' can result in underfitting. KNN is a non-parametric method, meaning it makes no assumptions about the underlying data distribution, making it versatile for various types of classification tasks. However, it can be computationally expensive, especially with large datasets, as it requires calculating distances for each query point. Despite this, KNN is widely used for its simplicity, effectiveness, and ability to handle multi-class classification problems. This model is employed in regression tasks.

## Grid search optimization and hyper-parameter tuning



*Figure: Fitting Random Forest model with hyper-parameter tuning and grid search optimization*

The aforementioned sample of code is implemented for 2 tasks of classification and regression and varied in 4 models. This code employed hyperparameter tuning using **GridSearchCV** along with **StratifiedKFold** cross-validation to optimize each model. The goal is to find the best combination of hyperparameters for the models to improve its performance. Each of these parameters has a predefined range of values, and the **GridSearchCV** will exhaustively search over all combinations of these values to find the optimal set for the model.

To ensure robust evaluation of the model, the code uses **StratifiedKFold** with 5 splits. This technique divides the data into 5 subsets (folds), where each fold is used once as a test set while the remaining folds are used as training sets. The stratified sampling ensures that each fold maintains the same proportion of class labels as the original dataset, which is especially useful in the case of imbalanced datasets. This method provides a more reliable estimate of model performance across different subsets of the data.

**GridSearchCV** is initialized with the model and the hyperparameter grid, and it is set to run using 5-fold cross-validation, parallelizing the computation with n\_jobs=-1. This means that the process will utilize all available CPU cores to speed up the search. After fitting the model to the dataset **X\_res** (features) and **y\_res** (target labels), **GridSearchCV** will evaluate all hyperparameter combinations and identify the set that maximizes the model’s performance based on cross-validation results. This process ensures that the final model is tuned to perform optimally on unseen data.

With the given strategies, each of the model’s hyper-parameters are well-tuned with following groups:

**For classification tasks**

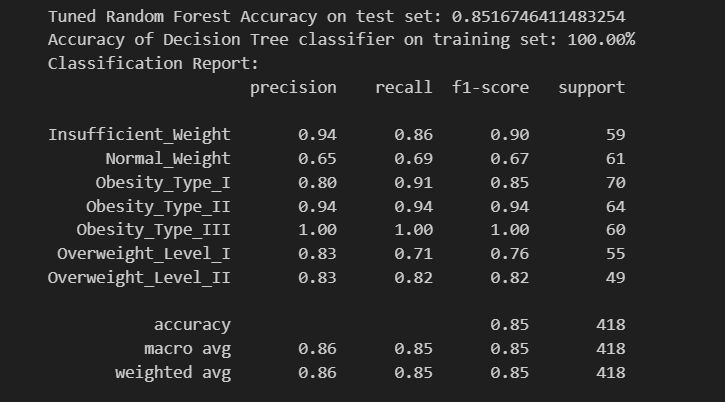
|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Hyper-parameter** | **Values** | **Best group** |
| Random Forest | 'max\_depth' | 3, 5, 7, 10, None | None |
| 'min\_samples\_split' | 2, 5, 10 | 2 |
| 'min\_samples\_leaf' | 1, 2, 5 | 1 |
| 'max\_features' | None, 'sqrt', 'log2' | 'sqrt' |
| XGBoost | 'n\_estimators' | 50, 100, 200 | 200 |
| 'max\_depth' | 3, 5, 7, 10 | 10 |
| 'learning\_rate' | 0.01, 0.1, 0.05 | 0.05 |
| 'subsample' | 0.5, 0.7, 0.8, 1.0 | 0.8 |

**For regression tasks**

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Hyper-parameter** | **Values** | **Best group** |
| KNN | 'n\_neighbors' | 3, 5, 7, 9, 11 | 5 |
| 'weights' | 'uniform', 'distance' | 'distance' |
| 'metric' | 'euclidean', 'manhattan' | 'manhattan' |
| Random Forest | 'n\_estimators' | 100, 200, 300 | 200 |
| 'max\_depth' | 10, 20, 30, None | None |
| 'min\_samples\_split' | 2, 5, 10 | 2 |
| 'min\_samples\_leaf' | 1, 2, 4 | 1 |
| 'max\_features' | None, 'sqrt', 'log2' | 'sqrt' |

## Evaluation Metrics

**For classification tasks**

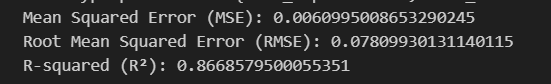


*Figure: Evaluation metrics for classification tasks*

In classification analysis, accuracy, precision, recall, and F1-score are utilized to evaluate the performance of Random Forest and XGBoost model:

* **Accuracy:** Accuracy measures the proportion of correctly classified instances out of all instances in the dataset. It is calculated as:
* **Precision:** Precision (also known as Positive Predictive Value) measures the proportion of true positive predictions among all positive predictions made by the model. It tells you how many of the predicted positive instances are actually positive. The formula for precision is:
* **Recall:** Recall (also known as Sensitivity or True Positive Rate) measures the proportion of actual positive instances that were correctly identified by the model. It answers the question: Of all the actual positives, how many did the model correctly identify? The formula for recall is:
* **F1-score:** F1-score is the harmonic mean of precision and recall, balancing both metrics in one number. It is especially useful when you need to balance the tradeoff between precision and recall. The formula for F1-score is:

**For regression tasks**



*Figure: Evaluation metrics for regression tasks*

In regression analysis, RMSE (Root Mean Squared Error), MSE (Mean Squared Error), and R² (Coefficient of Determination) are utilized to evaluate the performance of Random Forest and KNN model:

* **RMSE (Root Mean Squared Error):** RMSE provides a measure of how much error there is between the predicted and actual values, with a lower RMSE indicating better model performance. The formula for RMSE is:
* **MSE (Mean Squared Error):** MSE is the average of the squared differences between the predicted values and the actual values. It is sensitive to outliers, meaning larger errors have a more significant impact on the MSE. The formula for MSE is:
* **R² (Coefficient of Determination):** R² measures how well the regression model explains the variability in the data. R² values range from 0 to 1, with 1 indicating perfect prediction and 0 indicating that the model does not explain any variability. The formula for R² is:

# Results

**For classification tasks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **F1 score** | **Recall** |
| Random Forest | 0.852 | 0.860 | 0.850 | 0.850 |
| XGBoost | 0.844 | 0.850 | 0.850 | 0.840 |

**For regression tasks**

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **RMSE** | **MSE** | **R2** |
| Random Forest | 0.078 | 0.006 | 0.867 |
| KNN | 0.090 | 0.008 | 0.821 |

# Discussion

Random Forest provides built-in feature importance scores, which measure the contribution of each feature to the model's predictive power. We utilized these scores to identify the most influential features in predicting obesity levels. The feature importance analysis not only helps in understanding the key drivers of obesity but also facilitates the interpretation of the model's predictions. The top five most influential features identified by the Random Forest model were BMI, age, physical activity level, daily calorie intake, and family history of obesity. These findings align with the current understanding of obesity risk factors and highlight the model's ability to capture the underlying patterns in the data.

# Conclusion

***\*Declaration of AI Support:*** *This report has been prepared with the assistance of artificial intelligence (AI) tools, specifically in the areas of code assistance, lexical and grammar correction, and ensuring the clarity and coherence of written sections. While AI played a supporting role, all interpretations, conclusions, and final content have been thoroughly reviewed and validated by the author (myself). The use of AI aligns with RMIT’s guidelines on academic integrity, and the report maintains the principles of originality and ethical conduct.*

# Reference