Project Title: Prediction of Obesity Levels Based on Eating Habits and Physical Activities

Project Team:

- Joseph Christian T. Caser II
- Joshua A. Bartolata
- Juan Gabriel P. Potestades

Project Summary:

This project aims to develop a predictive model for obesity levels based on eating habits and physical activities. Using a dataset from Kaggle, the team will train a machine learning model to classify individuals into different obesity categories. The project will primarily utilize a Random Forest classifier, a robust and versatile algorithm suitable for this type of classification problem. However, the team will also explore and compare the performance of other classification algorithms, including Logistic Regression Algorithm and Support Vector Machines (SVM), to identify the most effective model for this prediction task. A user-friendly graphical interface (GUI) will be developed using Tkinter, allowing users to easily load data, run the models, and visualize the results. The project will provide valuable insights into the factors contributing to obesity and potentially assist in developing targeted interventions.

Project Objectives:

- **Main Objective**
  - To develop a reliable and accurate predictive model for classifying individuals into different obesity level categories based on their eating habits and physical activity patterns.

- **Specific Objectives:**
  - To preprocess and clean the Kaggle dataset, handling missing values and inconsistencies.
  - To implement and train a Random Forest classifier, a Logistic Regression model, and a Support Vector Machine (SVM) for obesity level prediction.
  - To evaluate the performance of the trained models using appropriate metrics and compare their effectiveness.
  - To develop a user-friendly GUI for data loading, model execution, and results visualization.

Project Scope:

- **Algorithms:**
  - **Random Forest Classifier**: The primary machine learning algorithm used for classification due to its robustness and versatility.
  - **Logistic Regression**: A linear model used for binary and multi-class classification problems. It provides probabilities of class membership.

    - **Support Vector Machines (SVM):** A powerful algorithm that finds the optimal hyperplane to separate data points of different classes. We will explore linear and potentially non-linear kernels.

- **Dataset:**
    - **Kaggle**: The dataset used for training and testing the model will be sourced from Kaggle (*https://www.kaggle.com/datasets/fatemehmehrparvar/obesity-levels/data*). This dataset includes features related to eating habits, physical activity, and demographics, along with the corresponding obesity level classification.

- **Evaluation Metrics:**
    - **Accuracy:** The percentage of correctly classified instances.
    - **Classification Report:** Includes precision, recall, F1-score, and support for each class.
    - **Confusion Matrix:** A table showing the number of true positive, true negative, false positive, and false negative predictions.
    - **ROC Curve and AUC (Area Under the Curve):** For evaluating the performance of the classifiers, especially in terms of their ability to distinguish between classes.

- **Tools:**
    - **Python:** The programming language used for development.
    - **Pandas:** For data manipulation and analysis.
    - **Matplotlib and Seaborn:** For data visualization.
    - **Scikit-learn:** For machine learning algorithms, evaluation metrics, and model selection.
    - **Tkinter:** For creating the GUI.

Project Timeline:

Task	Start Date	End Date	Duration	Status
Requirements Gathering & Data Exploration	Feb 6	Feb 7	1 day	**Done**
Data Cleaning & Preprocessing Script	Feb 7	Feb 9	2 days	**Done**
Model Selection & Implementation	Feb 9	Feb 10	1 day	**Done**
GUI Design & Development	Feb 10	Feb 11	1 day	**Done**
Model Training & Evaluation	Feb 11	Feb 12	1 day	**Done**
Testing & Debugging	Feb 12	Feb 15	3 days	**In progress**
Documentation & Report Writing	Feb 15	Feb 17	2 days	**In progress**
Presentation Preparation	Feb 17	Feb 18	1 day	**Not Started**

Project Deliverables:

- **Code:** The complete Python code for data preprocessing, model training, evaluation, and GUI, including implementations for Random Forest, Logistic Regression, and SVM.

- **Report:** A comprehensive document including:
    - Project overview and objectives
    - Data description and preprocessing steps
    - Model descriptions, training processes, and evaluation results for all implemented algorithms
    - Comparison of model performance
    - User manual for the GUI application

- **Presentation:** A PowerPoint presentation summarizing the project, methodology, and findings, including a comparison of the different algorithms.

Project Success Criteria:

- At least one of the trained models achieves a satisfactory accuracy score at least above 75% of the target.
- The GUI application is functional, user-friendly, and easy to navigate, allowing users to select and run different models.
- The project deliverables (code, report, presentation) are complete and well-documented, with clear explanations of the implemented algorithms and their performance