Introduction to Data Science

Do It Yourself (DIY) - 2

In this **Do It Yourself (DIY) – 2** assignment, student teams are required to perform the following tasks: (1) **Modeling**, (2) **Evaluation**, and (3) **Deployment**, which have been covered in the course *Introduction to Data Science*, using a provided real-world dataset.

Before we start

Please pay attention to the following notices:

- This is a **GROUP** assignment; all members must be present and complete the assignment in class under the supervision of the instructors.
- Since this is "Do It Yourself", all AI tools and Copilot are strictly prohibited.
  - Second warning: 50% of the score will be deducted.
  - Third warning: the entire score will be deducted.
- Duration: 3 hours.
- Any form of plagiarism, dishonesty, or misconduct will result in a grade of zero for the course.

Good luck!

Introduction

Student groups are required to develop a website that implements a model for **predicting obesity levels based on eating habits**. To accomplish this task, each group will be provided with an obesity dataset (the file `ObesityDataset.csv`), which was collected through a web-based survey platform where anonymous users responded to the questionnaire.

The dataset contains 2111 records and 14 attributes (features) as follows:

Attribute	**Description**
`gender`	The gender of the individual
`age`	The age of the individual
`family_history_with_overweight`	Family history of overweight
`FAVC`	Frequent consumption of high caloric food
`FCVC`	Frequency of consumption of vegetables
`NCP`	Number of main meals
`CAEC`	Consumption of food between meals
`SMOKE`	Smoking habit
`CH2O`	Consumption of water daily
`SCC`	Calories consumption monitoring
`FAF`	Physical activity frequency
`TUE`	Time using technology devices
`CALC`	Consumption of alcohol
`MTRANS`	Transportation used

The dataset also contains a target variable `NObesity` with six values, including:

- **Underweight**: BMI less than 18.5
- **Normal**: BMI from 18.5 to 24.9
- **Overweight**: BMI from 25.0 to 29.9
- **Obesity I**: BMI from 30.0 to 34.9
- **Obesity II**: BMI from 35.0 to 39.9
- **Obesity III**: BMI higher than 40

1 Modeling

In this section, student teams must build and train multiple machine learning models to predict the target variable `NObesity` using `sklearn` library.

1.1 Data preparation for modeling

Train-test split:

- Split the dataset into training and testing sets using an 80/20 ratio.
- Use `stratify=y` to preserve class distribution.
- Fix `random_state=42`.

Preprocessing:

- Encode categorical features using **One-Hot Encoding**.
- Apply **feature scaling** (e.g., StandardScaler) where appropriate.
- Implement preprocessing using **Pipeline** and **ColumnTransformer**.

Manual preprocessing outside the pipeline is not allowed.

1.2 Build classification models

Train at least $n + 1$ (n is number of team members) different multi-class classification models on the training set. Each model must be implemented as an end-to-end pipeline:

$$\text{Preprocessing} \rightarrow \text{Classifier}$$

Recommend models: Logistic Regression, Decision Tree, Random Forest, K-Nearest Neighbors, Naive Bayes, etc.

1.3 Model outputs for evaluation

After training, prepare the following outputs for the Evaluation section:

- Predicted class labels on the test set.
- Predicted class probabilities on the test set (required for ROC–AUC).

2 Evaluation

Evaluate each model using the following metrics, categorized by type:

- **Performance Overview:**
  - **Accuracy**: Overall proportion of correctly classified samples across all classes.
  - **Confusion Matrix**: Visualize a heatmap to analyze misclassifications between classes.
- **Classification Report:**
  - **Macro Precision**: Average precision across all classes.
  - **Macro Recall**: Average recall across all classes.
  - **Macro F1-score**: Harmonic mean of macro-averaged precision and recall.
- **Model Discrimination Ability:**
  - **Macro ROC - AUC**: Compute the macro-averaged Area Under the ROC Curve (AUC) to evaluate the model's overall class discrimination capability.

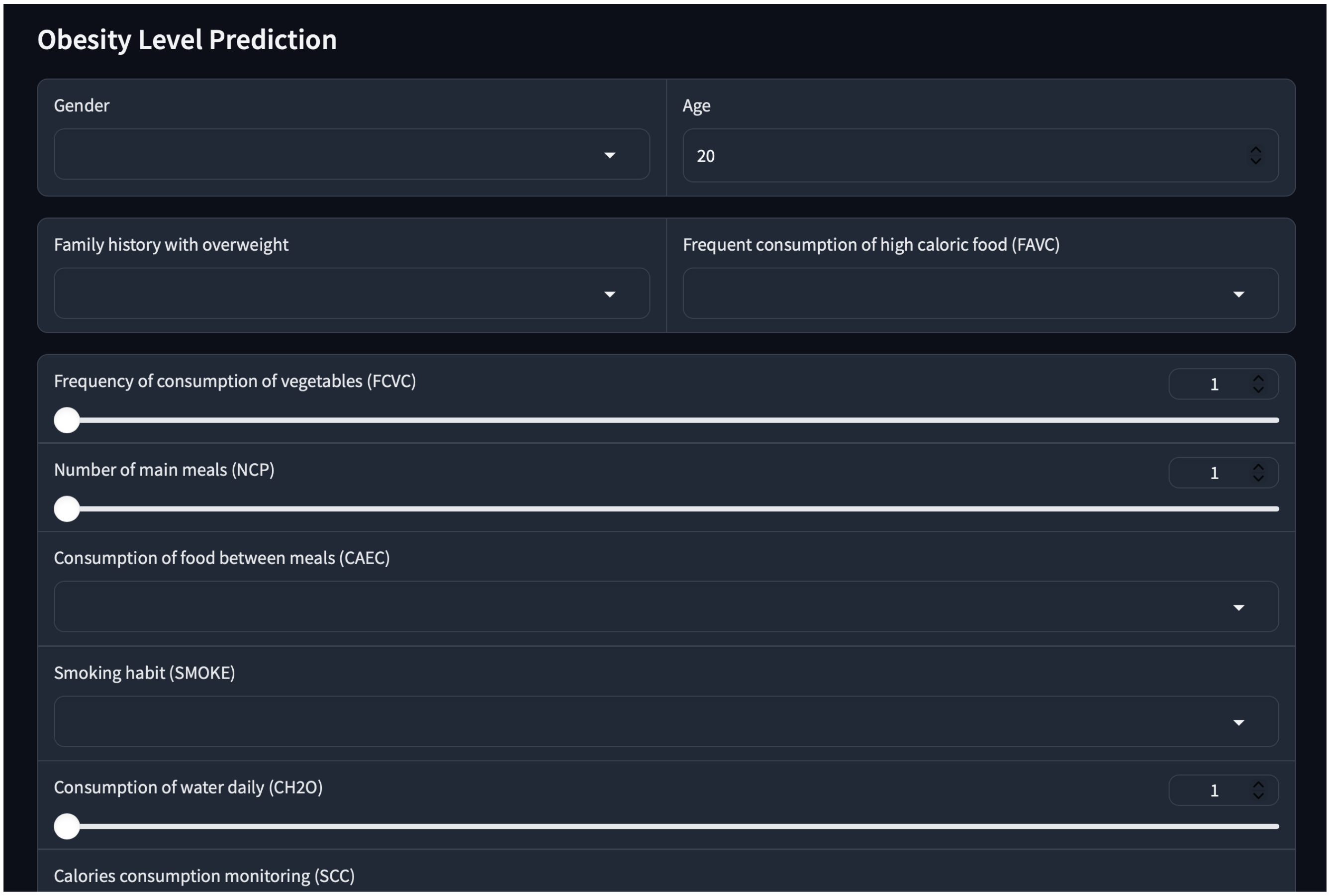
After implementing and evaluating all classification algorithms, analyze their performance across all metrics and provide detailed insights:

- Compare on evaluation metrics, such as accuracy, etc., to identify the best-performing model. Use the confusion matrix to analyze misclassifications and discuss potential behind reason (e.g., class imbalance).
- Recommend the best model(s) based on evaluation metrics and dataset characteristics.

3 Deployment

The best-performing model is deployed as a web application using **Gradio**. The trained end-to-end pipeline is saved and loaded to returns the predicted obesity level (`NObesity`) as the output.

The application is deployed on **Hugging Face Spaces** using the Gradio SDK, allowing users to interact with the model through a simple web interface for predicting obesity levels.



Obesity Level Prediction

Gender

Age: 20

Family history with overweight

Frequent consumption of high caloric food (FAVC)

Frequency of consumption of vegetables (FCVC): 1

Number of main meals (NCP): 1

Consumption of food between meals (CAEC)

Smoking habit (SMOKE)

Consumption of water daily (CH2O): 1

Calories consumption monitoring (SCC)

Figure 1: A demo for Obesity level prediction based on eating habits.

Assessment

No.	Details	Score
1	Classification algorithms implementation.	50%
2	Evaluation.	25%
3	Comparison and Analysis.	25%

The end.