LABORATORY PART 1

270228 Advanced topics in data 2

GitHub repository:

https://github.com/taed2-2526q1-gced-upc/TAED2-SmartHealth.AI/tree/main

Matilde

Steffen

Renaux

September 29, 2025

Contents

1 Introduction

1.1 Goal of the project

The goal of this project is to develop a SmartHealth-AI tool that provides user guidance in relation to obesity. The project aims to build a model that takes health metrics as input and translates these into a personalized recommendation, helping to reduce the chance of obesity if detected.

1.2 Teammates' evaluation

☒ *"All team members agree that they had an equal contribution to this delivery and project: completing a fair share of the team's work with acceptable/high quality, keeping commitments, and completing assignments on time, helping teammates who are having difficulty when it is easy or important".*

2 Methodology

2.1 Milestone 1: Inception

2.1.1 Selection of problem and requirements engineering for ML

This specific project was chosen, because it is a topic of high relevance in contemporary society. According to World Obesity Federation "the total number of adults living with obesity will increase by more than 115 % between 2010 and 2030"[1]. Therefore, the need for preventive measures is urgent and this is where the proposed SmartHealth-AI hopefully can provide value as an accessible and user-friendly tool. ø

In order to succeed with the project, a set of requirements have been defined, which will be taken under consideration throughout the different phases of development. Below is a descriptive overview of them.

Non-functional requirements

The non functional requirements defines the model's capabilities in terms of performance, scalability, data restraints and reproducibility.

The first requirement concerns the accuracy the model must achieve, which is defined to be at least 0.8 (i.e. ≥ 0.8).

Furthermore, the quality of predictions has to be monitored to account for potential class imbalance in the health data. Another important requirement related to the data is that the model must be able to generalize beyond the training set. Specifically, it should not be overly influenced by regional, cultural, or demographic biases present in the training data. If the training data primarily originates from a specific geographic area or population group, the system must be validated against independent datasets representing diverse user profiles to ensure that recommendations remain reliable and equitable across different contexts. Otherwise the geographical area that the model can take as input, must be defined accordingly.

Since the tool is intended for interactive use, inference time and generation of recommendation should remain below three minutes per user input.

In terms of scalability, the system should be able to handle multiple users simultaneously and be adaptable to larger datasets without significant changes to the overall architecture.

Finally, reproducibility is ensured by tracking iterations with fixed random seeds, version-controlling of datasets and defining dependencies in a shared environment file, so results can be replicated across different environments.

Functional requirements

The functional requirements of SmartHealth-AI is defined by the capabilities of the supervised machine learning system and the interaction with the users input. Therefore, the system should be able to process lifestyle and health information from the user and classify the individual into one of the obesity level categories.

The user should be able to enter the data manually through a simple interface. Then the system uses the trained Random Forest classifier to predict the obesity level. The system must then provide both the classification output and personalized recommendations. The recommendation system will be designed to provide personalized health guidance based on a user's obesity level. The core idea is to compare the user's profile with individuals who share similar characteristics but fall within a healthy weight range. This comparison hopefully gives realistic and achievable lifestyle adjustments.

The recommendations should be generated and ready within 3 minutes, and a message should be sent to the user, so the experience is fast and practical for the user.

Future functional and non-functional requirements

At this time in the process we have not decided whether to make a website or APP where the system should be implemented. With a website our idea is that the system sends the recommendations directly on the website, so no personal details are needed. In that way we don't have to think about GDPR regulations. The system would be a simple one-time interaction tool.

On the other hand with an APP would open the possibility for a more personalized assistant tool for long term guidance. However, this method introduces a lot of new aspects we have to consider and implement, like data privacy/security and ethical considerations.

2.1.2 Dataset card

The dataset card describes the 'Estimation of Obesity Levels Based on Eating Habits and Physical Condition' dataset introduced by Palechor & de la Hoz Manotas (2019) [2]. It consists of 2111 records and 17 attributes, including demographic, dietary, and lifestyle features, with a multi-class target variable (`NObeyesdad`) covering seven obesity categories ranging from insufficient weight to obesity type III.

- **Dataset details:** Collected via survey from individuals in Colombia, Peru, and Mexico (ages 14–61). Includes demographic, anthropometric, dietary, and lifestyle attributes.
- **Preprocessing:** Data cleaning, normalization, and oversampling with SMOTE to balance class distributions.

- **Intended use:** Educational and research purposes in obesity estimation and preventive health guidance.
- **Limitations:** 77% of the dataset is synthetically generated, limiting generalizability. Data is self-reported, and only represents three Latin American countries.
- **Ethical considerations:** Contains sensitive health-related data (weight, eating habits, activity). No personally identifiable information is included, but responsible use is required.

The full dataset card is available in the project repository:

https://github.com/taed2-2526q1-gced-upc/TAED2-SmartHealth.AI/blob/main/docs/datasetcard.md.

2.1.3 Model card

The model card has been made to provide a structured description of the SmartHealth-AI model, which is based on the Random Forest classifier. The model flow is showed in Figure 1 and shows how the lifestyle data is the input to the Random Forest model, which then categorizes the features and assigns each user to a category that makes the foundation for the individual recommendation.

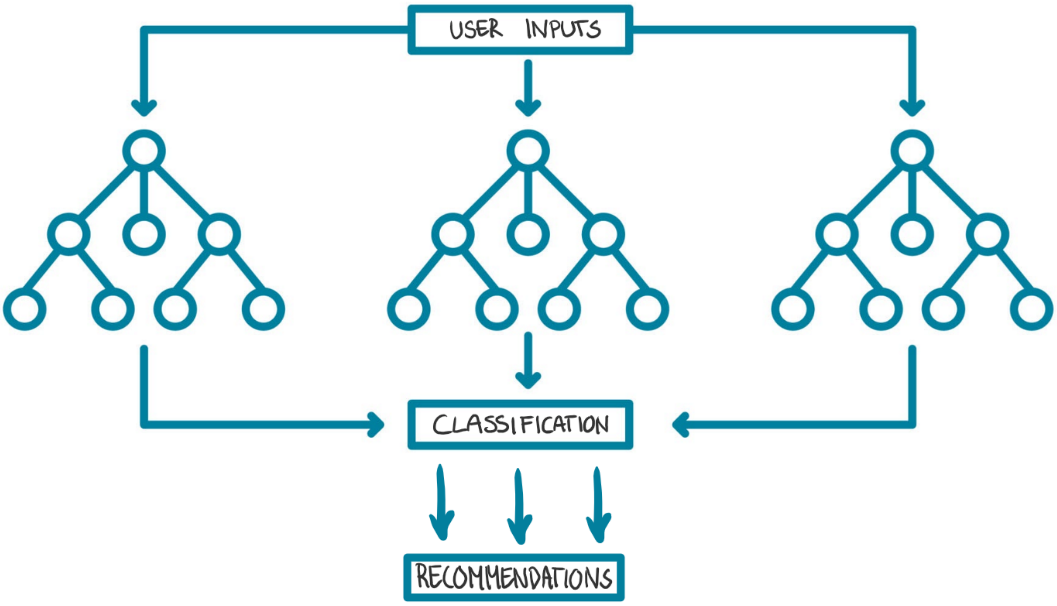

Figure 1: Fugure of the modelflow

The card includes the following main points:

- **Model details:** A supervised learning Random Forest model developed as part of the TAED2 course at UPC, intended solely for educational purposes.
- **Intended use:** To guide users toward healthier habits by providing preventive lifestyle recommendations when obesity risk is detected. The model is not designed for medical decision-making or professional diagnosis.
- **Limitations and risks:** The dataset contains synthetic components and may not fully represent real-world populations. As a result, predictions may carry bias and should not be interpreted as clinical advice or as a global
- **Ethical considerations:** User privacy and transparency are emphasized, and the model is recommended for experimentation and coursework only.

The full model card is available in the project repository:

https://github.com/taed2-2526q1-gced-upc/TAED2-SmartHealth.AI/blob/main/models/modelcard.md.

2.1.4 Project coordination and communication

GitHub has been the primary platform used for project coordination and communication. It has been chosen as it is good for code management and version control, ensuring reproducibility. Furthermore DagsHub has been connected to the GitHub repository for this project, as it allows for data management.

2.1.5 Selection of the cloud provider

The current project setup does not employ a dataset storage service. If a choice had been required during initialization, AWS S3 would have been selected due to its widespread use and seamless integration with machine learning workflows. However, upon further research, the UPC cloud services are identified as a more suitable solution for this project. The institutional availability of the UPC cloud for students, together with its free access and ease of use, makes it the preferred choice for future milestones. Even though the data presented in the project is relatively small in size, the use of a cloud provider still offers clear advantages, as it ensures that models and data are stored in a stable and reproducible environment that supports collaborative work.

2.2 Milestone 2: Model Building – Reproducibility

2.2.1 Project structure

In order to ensure a well-structured and reproducible workflow, the project was initialized using Cookiecutter, which provides an automatically generated setup with predefined folders and configuration files. This approach guarantees consistency across different environments and facilitates collaboration within the team.

When setting up the structure, a series of predefined parameters were selected to tailor the template to the specific needs of the project. The project was configured under the name `TAED2_SmartHealth.AI`, with the internal Python package named `taed2_smarthealth_ai`. It was set to run on Python 3.10, with environment management handled by `uv` and dependencies specified in a `pyproject.toml` file. No additional PyData packages or testing frameworks were included. For linting and formatting, the combined tools `flake8`, `black`, and `isort` were selected. This configuration enforces compliance with the PEP 8 style guide by detecting violations (`flake8`), automatically reformatting code to a consistent standard (`black`), and maintaining properly structured imports (`isort`). Together, these tools ensure not only compatibility with PEP 8 but also a uniform and reproducible code style across the project.

The project was released under the MIT license, while documentation support was disabled at initialization. Finally, the optional code scaffold was included to provide a basic starting point for development.

2.2.2 Code versioning

For the SmartHeath-AI project Git is used as the version control system. It's a great system for collaboration and development where Github Flow is served as primary workflow. Github flow is a effective branching strategy where the process is around a main branch and new feature branches can be made for development for models, functionalities etc. For our project different branches have been made. The first feature branch in our workflow was for data cleaning, so the data was fitted to future analysis and machine model training.

Thereafter, the data was suitable for model training. First off, a supervised decision tree model was created as a baseline model. Later on a random forest model was implemented and inserted in our main branch together with early results and findings.

This approach offers many advantages in terms of reproducibility. With Github flow it's easy to keep track of the development history of the project and how it will evolve.

2.2.3 Data versioning

To ensure reproducibility and traceability of the datasets used in the SmartHealth-AI project, we adopted Data Version Control (DVC). While Git provides versioning for source code, it is not designed to handle large or frequently changing data files. DVC complements Git by enabling lightweight tracking of data and model artifacts, while the actual files are stored in external storage (in our case, the remote repository provided by DagsHub, integrated with GitHub).

DVC structures the workflow into stages, corresponding to the main steps of the pipeline such as preprocessing, data splitting, and model training. Each stage specifies its inputs (e.g., raw data or scripts) and outputs (e.g., cleaned datasets, processed splits, trained models). These dependencies are declared in the dvc.yaml file, ensuring that if an input changes, only the affected stages are re-executed. The resulting lock file records the exact versions of inputs and outputs, guaranteeing reproducibility.

Both data and models are versioned in this way, with Git managing code and configuration, while DVC manages large files through external storage. This provides a clean separation between source control and data storage, while keeping them fully synchronized.

2.2.4 Experiment tracking

In order to track the experiments throughout the development of the model, MLflow has been implemented. It provides a way to log parameters, metrics, and artifacts in a structured manner, which makes it easier to compare results, reproduce experiments, and manage models over time. In the project, MLflow is expected to display the iterative development process by ensuring that changes and improvements to the model are properly documented and organized

However, as the implementation of MLflow was initiated a bit later in the process of the model development, some of the changes that have already been made, are not documented. Specifically, the initial model was implemented as a decision tree classifier. However, it was quite early observed that a random forest model achieved significantly higher accuracy, and the model-type was therefore changed accordingly for the further development.

MLflow has now been implemented, and it is expected to provide structure and support for future iterations of the model. Through MLflow the model development can be automatically logged, making it possible to reproduce experiments and compare results in a systematic way. This is particularly important as the project aims to refine the model iteratively, with performance being evaluated against defined non-functional requirements such as an accuracy of at least 0.8 and robustness across diverse datasets.

3 Bibliography

References

[1] World Obesity Federation. *World Obesity Atlas 2025: Majority of countries unprepared for rising obesity level*. Accessed: 2025-09-25. Mar. 2025. URL: https://www.worldobesity.org/news/world-obesity-atlas-2025-majority-of-countries-unprepared-for-rising-obesity-level.

[2] Fabio Mendoza Palechor and Alexis de la Hoz Manotas. "Dataset for estimation of obesity levels based on eating habits and physical condition in individuals from Colombia, Peru and Mexico". In: *Data in Brief* 25 (2019), p. 104344. DOI: 10.1016/j.dib.2019.104344. URL: https://archive.ics.uci.edu/dataset/544/estimation+of+obesity+levels+based+on+eating+habits+and+physical+condition.