



Universitat Autònoma de Barcelona
Escola de Enginyeria

Grado en Ingeniería de Datos
Sistema Interactivo para el Seguimiento de Salud y Predicción
de Hábitos

Alumno: Pau Montesinos Cáliz

Tutor: Oriol Cortés Comellas

Fecha: 20/04/2025

1. Objetivos

El objetivo principal de este proyecto sigue siendo el desarrollo de una plataforma capaz de recoger hábitos y datos sobre las personas con el fin de analizar su estado de salud, estimar la probabilidad de padecer ciertas enfermedades y ofrecer recomendaciones personalizadas para mejorar sus hábitos. En el primer informe, se ofrecieron unos objetivos a cumplir y, varios de ellos, han sido ya solventados:

- Identificar uno o varios datasets útiles: se han identificado tres datasets. Uno de ellos podría acabar descartándose, más no habría problema en si no se acaba realizando, pues este ya tiene su código totalmente definido.
- Revisar y limpiar las impurezas de los datasets escogidos: por cada dataset se han identificado las columnas necesarias de cambios y se ha aplicado la limpieza correspondiente.
- Utilizar los datos para entrenar varios modelos de predicción tanto de Machine Learning como de Deep Learning: por el momento, se han creado los modelos basados en Machine Learning y los basados en Deep Learning se crearán en un futuro cercano.

Hechos estos avances, sigue habiendo varios objetivos más a cumplir, más es necesario sumar algunos objetivos a los que ya teníamos a raíz del proceso que se ha ido realizando en el trabajo:

- Identificación de mejores parámetros de entrenamiento de los modelos creados.
- Identificación de mejores estrategias para conseguir métricas con valores mayores a los obtenidos.
- Creación y entrenamiento de modelos de Deep Learning.
- Aplicación de estrategias identificadas previamente y creación y desarrollo de nuevas.
- Evaluación de estrategias y resultados.
- Desarrollo de la plataforma que contenga todos los modelos creados anteriormente.

2. Metodología

Para la realización del proyecto se ha decidido usar una metodología ágil, concretamente Kanban, con la cual se ha podido organizar el trabajo en pequeñas tareas, permitiendo de esta manera mejorar y evaluar el trabajo realizado hasta el momento.

En cada tarea se ha definido un objetivo claro, por lo que se ha podido estructurar el proyecto de una forma más sencilla y así, por lo tanto, adaptarse de mejor manera a posibles imprevistos.

Por el momento, se ha estructurado el proyecto de una forma modular. De esta manera, cada módulo es independiente de los demás. Esto, además de ser una forma de tener estructurado el código, es una manera también de facilitar la adaptación a cambios futuros, debido a la posibilidad de corrección de errores e implementación de mejoras a sabiendas de que, en todo momento, el resto del sistema no estará comprometido.

3. Desarrollo

Para la realización del progreso actual se ha creado un repositorio en [GitHub](#) que contiene el código actualizado. Este desarrollo se está llevando a cabo en la rama de desarrollo y se hará un *merge* a la rama definitiva cuando el código esté totalmente madurado.

Dentro del repositorio, podremos ver un conjunto de paquetes de Python con todas las funciones necesarias para el entrenamiento final de los modelos (de Machine Learning por el momento). En nuestro caso particular, los archivos más destacables serían:

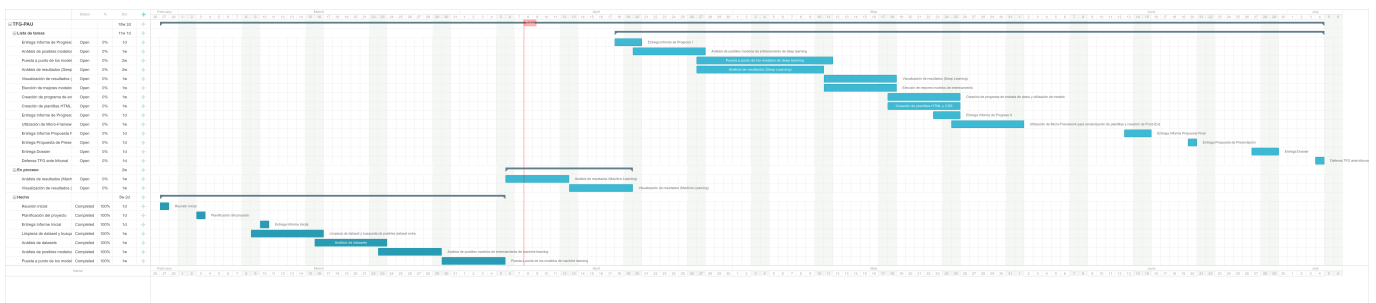
- **utils.py:** Archivo que contiene funciones generales reutilizables a lo largo de los diferentes módulos que forman el proyecto.
- **Limpieza de datos:** Se han creado tres archivos diferentes con el nombre de clean.py. Cada uno de estos archivos se encarga de hacer el proceso de limpieza de uno de los datasets para el posterior consumo de sus datos.
- **Modelos de Machine Learning:** Tres archivos independientes para los distintos modelos utilizados: LogisticRegression.py, RandomForest.py y XGBoost.py. Cada uno de estos archivos contiene una clase que ayuda a realizar las funcionalidades básicas de un modelo: entrenamiento, predicción y evaluación. Estas clases también contienen funcionalidades de guardado y carga de modelos, por lo que un modelo funcional se puede guardar para ser usado en un futuro y no tener que entrenarlo de nuevo.

4. Planificación

La planificación propuesta en el informe inicial sigue su cauce sin mayores problemas. Desde este link se puede seguir el tablero Kanban:

[Planificación del proyecto](#)

De igual manera, este diagrama de Gantt indica en que punto de la planificación está el proyecto, lo que ya se ha acabado y lo que aún queda por hacer:



5. Bibliografía

- [1] Pandas, “pandas - Python Data Analysis Library,” *Pydata.org*, 2024. <https://pandas.pydata.org/> (accessed Mar. 05, 2025).
- [2] S. Developers, “scikit-learn: machine learning in Python — scikit-learn 1.6.1 documentation,” *Scikit-learn.org*, 2025. <https://scikit-learn.org/stable/> (accessed Mar. 05, 2025).
- [3] WHATWG, “HTML Standard,” *Whatwg.org*, 2025. <https://html.spec.whatwg.org/> (accessed Mar. 05, 2025).
- [4] W3C, “Cascading Style Sheets,” *W3.org*, 2024. <https://www.w3.org/Style/CSS/> (accessed Mar. 05, 2025).
- [5] P. Projects, “Welcome to Flask — Flask Documentation (3.1.x),” *Palletsprojects.com*, 2025. <https://flask.palletsprojects.com/en/stable/> (accessed Mar. 05, 2025).
- [6] Python Software Foundation, “sqlite3 — DB-API 2.0 interface for SQLite databases,” *Python documentation*, 2025. <https://docs.python.org/3/library/sqlite3.html> (accessed Mar. 05, 2025).
- [7] Plotly, “Plotly,” *Plotly.com*, 2025. <https://plotly.com/python/> (accessed Mar. 05, 2025).
- [8] K. Team, “Keras documentation: Keras 3 API documentation,” *Keras.io*, 2025. <https://keras.io/api/> (accessed Mar. 05, 2025).
- [9] P. Team, “PyTorch documentation — PyTorch 2.6 documentation,” *Pytorch.org*, 2023. <https://pytorch.org/docs/stable/index.html> (accessed Mar. 05, 2025).
- [10] Gaurav, “An Introduction to Gradient Boosting Decision Trees - Machine Learning Plus,” *Machine Learning Plus*, Jun. 12, 2021. <https://www.machinelearningplus.com/machine-learning/an-introduction-to-gradient-boosting-decision-trees/> (accessed Mar. 06, 2025).
- [11] IBM and E. K. Erika Russi, “XGBoost,” *Ibm.com*, May 09, 2024. <https://www.ibm.com/es-es/think/topics/xgboost> (accessed Mar. 06, 2025).
- [12] Z. Xiang, Z. Zhang, and Q. Liu, “Sparse Attention-Based Neural Networks for Code Classification,” *arXiv.org*, 2023. <https://arxiv.org/abs/2311.06575> (accessed Mar. 08, 2025).
- [13] S. Developers, “sklearn.model_selection,” *scikit-learn*, 2025. https://scikit-learn.org/stable/api/sklearn.model_selection.html (accessed Mar. 19, 2025).
- [14] Python Software Foundation, “typing — Support for type hints,” *Python documentation*, 2025. <https://docs.python.org/3/library/typing.html> (accessed Mar. 19, 2025).

[15] S. Developers, “LogisticRegression,” *scikit-learn*, 2025. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html (accessed Mar. 21, 2025).

[16] S. Developers, “RandomForestClassifier,” *scikit-learn*, 2025. <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html> (accessed Mar. 22, 2025).

[17] Xgb. Developers, “XGBoost Documentation — xgboost 3.0.0 documentation,” *Readthedocs.io*, 2025. https://xgboost.readthedocs.io/en/release_3.0.0/ (accessed Mar. 24, 2025).