UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería en Informática

TFG del Grado en Ingeniería Informática

Investigación, creación y explotación de modelos para el estudio del COVID-19 en pacientes del Servicio Cántabro de Salud.

Documentación Técnica

Presentado por Segundo González González
en Universidad de Burgos — 6 de julio de 2023
Tutor: Carlos López Nozal
Cotutor: José Francisco Díez Pastor

Índice general

Índice de figuras

Índice de tablas

Apéndice A

Plan de Proyecto Software

A.1. Introducción

Como primera aproximación, en este apartado trataremos de explicar la planificación del proyecto y qué fases se han llevado a cabo.

A.2. Planificación temporal

Se ha considerado una metodología ágil (scrum) para llevar la planificación y el desarrollo de este proyecto, utilizando los conceptos teóricos y prácticos de la asignatura *Gestión de proyectos*.

La plataforma elegida fue Github y en ella se fueron escribiendo tanto los *milestones* como los *issues*.

El proyecto se comenzó realizando sprints cada dos semanas y finalizando cada sprint con una reunión con los tutores, en las que se ponía en común los avances en el proyecto, proponiendo mejoras o cambios, y se establecían los objetivos para el siguiente sprint.

Estas sprints que comenzaron en septiembre de 2021 se vieron interrumpidos hasta junio de 2023 por diferentes motivos personales y se retomaron para poder terminar este proyecto.

A continuación, se detalla el trabajo realizado en cada sprint:

14/09/2021 – 30/09/2021
Sprint 0: Kickoff software project

El primer sprint lo comenzamos con el número 0 debido a su carácter inicial que básicamente consistió en ver cómo se iban a planificar los sprints,

exponer el alcance del proyecto y dar forma al inicio del proyecto de forma oficial.

- Creación de repositorio en Github para poder llevar la planificación del proyecto, así como alojamiento de los futuros modelos desarrollados.
- Exposición del alcance del proyecto e información acerca de la cohorte de datos aún en proceso de petición al Instituto de de Investigación Sanitaria de Valdecilla (IDIVAL)
- Revisión preliminar de bibliografía en la que empezar a poner las bases de cómo acometer el proyecto.
- Consulta de TFG's presentados en la UBU relacionados con el proyecto a acometer
- Formalización de registro del proyecto en UBU.

01/10/2021 – 14/10/2021
Sprint 1: Replicación trabajos relacionados

Aquí se comienza a implantar los cimientos del proyecto comenzando una investigación profunda de trabajos relacionados que permitiese entender cómo acometer un proyecto de aprendizaje automático sobre datos sanitarios.

- Investigación de bibliotecas en python para la interpretación de modelos de caja negra
- Escoger un trabajo relacionado con este proyecto para poder replicarlo
- Comenzar a añadir a la memoria los trabajos relacionados

15/10/2021 – 31/10/2021
Sprint 2: Cumplimentar primeros capítulos de la memoria

Comenzamos a cumplimentar la memoria comenzando por el primer acercamiento a LaTeX y repasar la plantilla de la memoria.

- Primeros pasos con LaTex y acercamiento a la utilización de la herramienta web Overleaf.
- Cumplimentar memoria: Introducción
- Cumplimentar memoria: Objetivos del proyecto

01/11/2021 – 15/11/2021
Sprint 3: Replica de trabajos relacionados

Se recibe la cohorte de datos desde el IDIVAL y se comienza el desarrollo de un modelo de réplica del trabajo relacionado escogido *Innate and Adaptive Immune Assessment at Admission to Predict Clinical Outcome in COVID-19 Patients*[4]

- Replicar el modelo en python.
- Comparar resultados de la replicación con los obtenidos en los trabajos
- Comparación de resultados con otros clasificadores

16/11/2021 – 30/11/2021
Sprint 4: Mejoras en la réplica del trabajo

En este sprint se intenta mejorar el cuaderno jupyter de la réplica del trabajo, puesto que el resultado que arroja el actual cuaderno está bastante lejos del resultado del trabajo original.

- Implementar Validación cruzada 2x5
- Utilizar pipeline en código

01/12/2021 – 23/12/2021
Sprint 5: Buscar clasificador con AutoSKlearn

Se propone seguir con el concepto experimental del proyecto y se acude a la biblioteca autosklearn para buscar mejores clasificadores para nuestra cohorte de datos.

- Estudiar uso de AutoSKlearn.
- Dockerización de Anaconda para incluir AutoSKLearn y demás librerías necesarias.
- Creación de cuaderno jupyter para uso de Autosklearn.
- Recoger resultados de AutoSKLearn aplicados a la cohorte de datos.
- Instalación y uso de librería Shap.
- Instalación y investigación de librerías de optimización de parámetros.

21/06/2023 - 06/07/2023
Sprint 6: Retomar el trabajo

Se vuelve a retomar el trabajo dónde se dejó en 2022. Se revisan todos los cuadernos jupyter y la documentación para implementar los cambios que sean necesarios.

- Revisión de la documentación de la memoria
- Reorganización del espacio GitHub
- Actualización contenedor con la última versión de Jupyter Notebooks
- Revisión y ejecución de cuadernos jupyter realizados
- Cumplimentación de los apartados de la memoria que aún no estaban documentados
- Realización de presentación para los vídeos a entregar
- Entrega de TFG en los espacios asignados en UBU Virtual

A.3. Estudio de viabilidad

A continuación se expone la posible viabilidad legal y económica de este proyecto.

Viabilidad económica

Por las características experimentales de este proyecto, creemos que este proyecto no se enmarca dentro de unas aplicaciones çomerciales"; el interés es puramente científico y en aras de aumentar el conocimiento sobre el aprendizaje automático con datos sanitarios.

También es correcto indicar, que llegado el momento en que se pudiera establecer un modelo de clasificación de pacientes en el triaje de una admisión hospitalaria con una muy alta precisión, es muy posible que las empresas de software hospitalario quisieran implementarlo en sus cores, si bien es cierto que tratándose en una pandemia, el uso del modelo distribuido debería ser de acceso universal.

El colectivo principal que mayor interés podría despertar en este proyecto sería, por supuesto, el de los profesionales sanitarios, sin dejar de lado a los entes gubernamentales, ya que estos últimos serían los posibles encargados de su distribución.

Viabilidad legal

En el caso de las herramientas, no se considera ningún impedimento la legislación, puesto que todas las herramientas pueden utilizarse en cualquier aplicación comercial, debido a sus licencias (BSD o MIT en general).

Por contra, al estar tratando datos clínicos que se enmarcan dentro del nivel más alto de segurida de la información, se deberían establecer fuertes medidas de seguridad sobre ellos.

Apéndice B

Especificación de Requisitos

B.1. Introducción

En este apartado se exponen los principales requisitos del proyecto.

B.2. Objetivos generales

Estos son los objetivos que se marcaron en el proyecto:

- Replicar el trabajo *Innate and Adaptive Immune Assessment at Admission to Predict Clinical Outcome in COVID-19 Patients* [4].
- Buscar trabajos relacionados y clasificar las variables, técnicas y herramientas que se utilizaron.
- Estudiar los datos proporcionados por el Instituto de Investigación Sanitaria de Valdecilla (IDIVAL) y considerar como procesarlos para poder obtener un un modelo de predicción que mejore los rendimientos del trabajo *Innate and Adaptive Immune Assessment at Admission to Predict Clinical Outcome in COVID-19 Patients* [4].
- Utilizar librerías de aprendizaje automático que ayuden a encontrar el mejor método de clasificación de entre diferentes métodos a partir de las variables/marcadores para el Covid-19 y disponibles en el conjunto de datos.
- Minimizar las características a utilizar para su utilización en el modelo a desarrollar.
- Investigar qué hiperparámetros pueden mejorar el método de clasificación elegido para el modelo final.

B.3. Catálogo de requisitos

Debido al carácter fundamentalmente experimental de nuestro proyecto, distinguiremos en requisitos funcionales (qué realizan los modelos desarrollados) y no funcionales (cómo lo consigue).

Requisitos funcionales

- **RF-1** Los datos deben poder ser divididos en conjuntos de entrenamiento y validación.
  - **RF-1.1** Se indicará qué porcentaje de datos del total se utilizará en el conjunto de validación.
  - **RF-1.2** Aleatorización de la inclusión de instancias en el conjunto de entrenamiento o validación, permitiendo siempre una división paritaria en ambos conjuntos.
- **RF-2** Se debe poder aplicar técnicas y algoritmos de preparación de los datos.
  - **RF-2.1** Se debe poder eliminar los valores nulos de la característica a entrenar.
  - **RF-2.2** Se debe poder imputar los valores perdidos o nulos en el resto del conjunto de datos
- **RF-3** Se debe poder entrenar un modelo.
  - **RF-3.1** Ofrecer la posibilidad de escoger entre diferentes algoritmos de clasificación.
  - **RF-3.2** Debe guardarse el mejor modelo generado durante el entrenamiento.
  - **RF-3.3** Se debe poder mostrar gráficamente los resultados del entrenamiento.
- **RF-4** Se debe poder realizar búsquedas de mejor clasificador de forma automática.
  - **RF-4.1** Se debe poder cambiar el dataset a utilizar.
  - **RF-4.2** Se debe poder cambiar los parámetros de búsqueda.
  - **RF-4.3** Se debe obtener una solución gráfica en la que observar los resultados.
- **RF-5** Se debe poder optimizar los hiperparámetros de un modelo.
  - **RF-5.1** Se debe poder cambiar el modelo a optimizar.

  - **RF-5.2** Se debe obtener un gráfico con los resultados.

- **RF-6** Se debe poder probar un modelo guardado.
  - **RF-6.1** Se debe poder encontrar el modelo a cargar.

Requisitos no funcionales

- **RNF-1** Portabilidad: Se puede ejecutar entre diferentes sistemas operativos y entornos de desarrollo, lo que evita problemas de configuración y compatibilidad.
- **RNF-2** Reproducibilidad: El contenedor contiene todas las dependencias y configuraciones específicas, lo que facilita la reproducción de resultados y la colaboración.
- **RNF-2** Aislamiento: Los paquetes o bibliotecas instalados en el contenedor no afectarán a tu sistema host, y viceversa.
- **RNF-2** Escalabilidad: Permite manejar grandes conjuntos de datos y ejecutar cálculos intensivos de manera eficiente utilizando recursos distribuidos.
- **RNF-2** Administración simplificada: Gestión de múltiples Jupyter Notebooks y entornos. Se pueen crear y administrar diferentes contenedores para proyectos específicos o diferentes configuraciones de entorno.
- **RNF-2** Flexibilidad: Permite personalizar y configurar el entorno para instalar las bibliotecas y herramientas necesarias, configurar extensiones y temas, y ajustar los recursos del contenedor para optimizar el rendimiento.

B.4. Especificación de requisitos

Diagrama de caso de uso

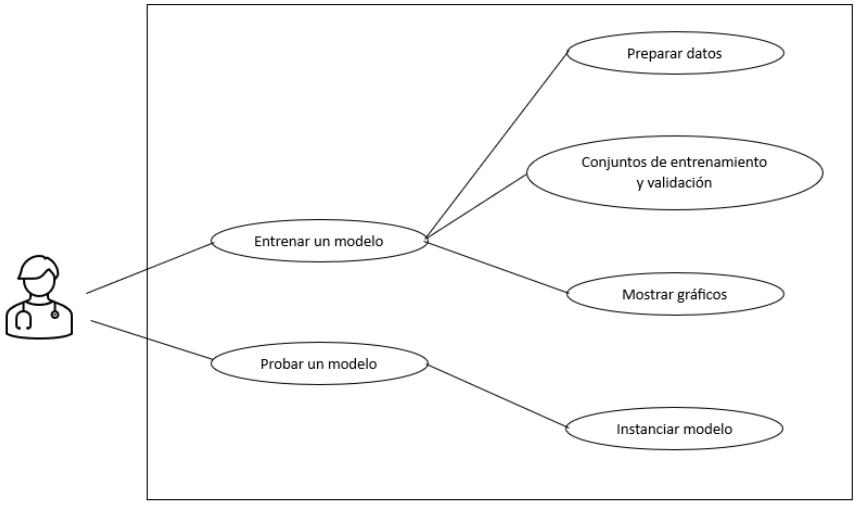

Figura B.1: Diagrama de caso de uso

Especificaciones de los casos de uso

Caso de uso	División de datos en entrenamiento y validación
Requisitos	RF-1 RF-1.1 RF-1.2
Descripción	Los datos deben poder ser divididos en conjuntos de entrenamiento y validación.
Precondiciones	Se debe disponer de un conjunto de datos.
Acciones	1. Se indicará qué porcentaje de datos del total se utilizará en el conjunto de validación. 2. Aleatorización de la inclusión de instancias en el conjunto de entrenamiento o validación, permitiendo siempre una división paritaria en ambos conjuntos.

continúa en la página siguiente

continúa desde la página anterior

Postcondiciones	Datos divididos en dos conjuntos.
Excepciones	El conjunto no lleva datos o está corrupto.
Importancia	Media

Tabla B.1: Caso de Uso *División de datos en conjuntos de entrenamiento y validación*

Caso de uso	Preparación de datos
Requisitos	RF-2 RF-2.1 RF-2.2
Descripción	Los datos deben poder ser divididos en conjuntos de entrenamiento y validación.
Precondiciones	Se debe disponer de un conjunto de datos.
Acciones	1. Se debe poder eliminar los valores nulos de la característica a entrenar. 2. Se debe poder imputar los valores perdidos o nulos en el resto del conjunto de datos.
Postcondiciones	Ddatos preparados.
Excepciones	El conjunto no lleva datos o está corrupto.
Importancia	Media

Tabla B.2: Caso de Uso *Aplicar técnicas y algoritmos de preparación de los datos*

Caso de uso	Entrenar un modelo
Requisitos	RF-3 RF-3.1 RF-3.2 RF-3.3
Descripción	Se debe poder entrenar un modelo.
Precondiciones	Se debe disponer de un conjunto de datos.
Acciones	1. Ofrecer la posibilidad de escoger entre diferentes algoritmos de clasificación. 2. Debe guardarse el mejor modelo generado durante el entrenamiento. 3. Se debe poder mostrar gráficamente los resultados del entrenamiento.

continúa en la página siguiente

continúa desde la página anterior

Postcondiciones	Modelo entrenado.
Excepciones	El conjunto no lleva datos o está corrupto.
Importancia	Alta

Tabla B.3: Caso de Uso *Entrenar un modelo*

Caso de uso	Búsqueda automática de mejor clasificador
Requisitos	RF-4 RF-4.1 RF-4.2 RF-4.3
Descripción	Se debe poder realizar búsquedas de mejor clasificador de forma automática.
Precondiciones	Se debe instalar la librería de automl.
Acciones	1. Se debe poder cambiar el dataset a utilizar. 2. Se debe poder cambiar los parámetros de búsqueda. 3. Se debe obtener una solución gráfica en la que observar los resultados.
Postcondiciones	Mejor clasificador encontrado.
Excepciones	Error al realizar la tarea de búsqueda.
Importancia	Media

Tabla B.4: Caso de Uso *Búsqueda de mejor clasificador de forma automática.*

Caso de uso	Optimización de hiperparámetros
Requisitos	RF-5 RF-5.1 RF-5.2
Descripción	Se debe poder optimizar los hiperparámetros de un modelo.
Precondiciones	Se debe instalar la librería de optimización de hiperparámetros.
Acciones	1. Se debe poder cambiar el modelo a optimizar. 2. Se debe obtener un gráfico con los resultados.
Postcondiciones	Mejores hiperparámetros encontrados.
Excepciones	Error al realizar la tarea de optimización.

continúa en la página siguiente

continúa desde la página anterior	
Importancia	Media

Tabla B.5: Caso de Uso *Optimización de hiperparámetros.*

Caso de uso	Prueba de modelos
Requisitos	RF-6 RF-6.1
Descripción	Se debe poder probar un modelo guardado.
Precondiciones	Modelos desarrollados alojados correctamente
Acciones	1. Se debe poder encontrar el modelo a cargar.
Postcondiciones	Prueba correcta de modelo
Excepciones	No se encuentra el modelo en la ubicación
Importancia	Media

Tabla B.6: Caso de Uso *Prueba de modelos*

Apéndice C

Especificación de diseño

C.1. Introducción

En este apartado recogemos el diseño experimental desarrollado en el proyecto.

C.2. Diseño de datos

La cohorte de datos han sido proporcionados por el Instituto de Investigación Sanitaria de Valdecilla (IDIVAL) sobre una población de 305 pacientes, cuyas muestras fueron recogidas entre Abril de 2020 y Marzo de 2021 y que tuvieron un resultado positivo en COVID-19. El valor de clasificación es el **score** de evolución clínica que tiene un valor de 1 a 5, siendo 1=asintomático; 2=tratamiento con gafas nasales; 3=tratamiento con ventimask; 4=Ingreso en UCI; 5=exitus.

El dataset se compone de un archivo csv con las siguientes características:

- 420 filas o determinaciones.
- 97 columnas o marcadores.
- Sólo hay 158 filas con el valor de clasificación score cumplimentado.

Los valores correspondientes a la intensidad de luz presentan una amplia variedad, moviéndose generalmente entre los intervalos de -10^5 a 10^5.

idMuestra	IDInm	NH	AntiN	AntiS	AntiM	age	gender	score	Line Timepoint	Timepoint	...	TH27p28n	TH27n28n	CTLDRn38n
0	1407962	3084327	NaN	NaN	NaN	88	1	NaN	NaN	0.0	...	0.007514	0.037952	0.075808
1	1407963	587116	NaN	NaN	NaN	46	0	NaN	NaN	0.0	...	0.001384	0.000951	0.057471
2	1408025	764584	NaN	NaN	NaN	82	0	3.0	1.0	1.0	...	0.027458	0.085781	0.232804

Figura C.1: Ejemplo de conjunto de datos

C.3. Diseño procedimental

En la realización de un nuevo experimento, el científico de datos posee total libertad para elegir las herramientas y las técnicas que va a aplicar. En cualquier caso, suele coincidir de forma general el flujo de trabajo.

El proceso de creación de un experimento lleva habitualmente las siguentes fases: carga de datos, preparación de estos, configuración del modelo y entrenamiento. Puede verse en la imagen Diagrama de flujo de realización de un experimento.

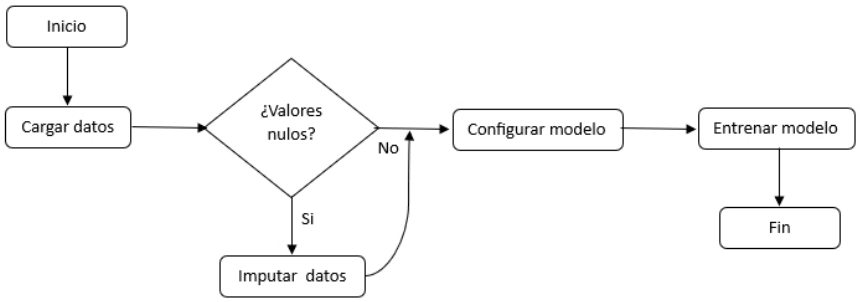

Figura C.2: Diagrama de flujo de realización de un experimento

C.4. Diseño arquitectónico

Apéndice D

Documentación técnica de programación

D.1. Introducción

Comentaremos aquí los detalles más técnicos sobre el proyecto de forma que sea posible para cualquiera recrear los experimentos y continuar el trabajo aquí presentado.

D.2. Estructura de directorios

El código fuente del proyecto se encuentra en Github en el directorio 2. Cuadernos TFG

D.3. Compilación, instalación y ejecución del proyecto

El entorno de trabajo se contenedor Docker [1]. Bajo la aplicación *Container Station* se instaló la imagen del contenedor *jupyter/tensorflow-notebook* [3] que se descargó de la plataforma Docker Hub. En la instalación se añadió la creación de un volumen de almacenamiento persistente para poder ir guardando de forma segura los cuadernos que se iban realizando. Este contenedor se escogió porque ya incorpora gran parte del ecosistema de bibliotecas científicas de python; otras bibliotecas fueron instaladas en el arranque del contenedor ya que son utilizadas en este trabajo, como son Autosklearn, Pipeline profiler, Shap y Optuna.

D.4. Manual del programador

El entorno de ejecución es el mismo que el de desarrollo, realizándose en Jupyter Notebooks [2] y en el lenguaje de programación Python [5]

Apéndice E

Documentación de usuario

Debido al carácter plenamente experimental de este proyecto, no existe una distribución formal de los modelos desarrollados para un usuario convencional. Es por esto que este apartado no aplica en este proyecto.

Bibliografía

[1] Solomon Hykes. Docker, 2013. https://en.wikipedia.org/wiki/Docker_(software).

[2] Jupyter Labs. Jupyter notebooks, 2014. https://jupyter-notebook.readthedocs.io/en/latest/.

[3] Jupyter Labs. Jupyter docker stacks, 2015. https://jupyter-docker-stacks.readthedocs.io/en/latest/#.

[4] David San Segundo, Francisco Arnáiz de las Revillas, Patricia Lamadrid-Perojo, Alejandra Comins-Boo, Claudia González-Rico, Marta Alonso-Peña, Juan Irure-Ventura, José M. Olmos, María C. Fariñas, and Marcos López-Hoyos. Innate and adaptive immune assessment at admission to predict clinical outcome in covid-19 patients. *Biomedicines*, 9(8):917, 2021. https://doi.org/10.3390/biomedicines9080917.

[5] Guido van Rossum. Python, 1991. https://es.wikipedia.org/wiki/Python.