

Título

Antonio Molner Domenech

Trabajo de Fin de Grado
Ingeniería Informática

Supervisado por:
Alberto Guillén



**UNIVERSIDAD
DE GRANADA**

Universidad de Granada, España
Junio 2020

Índice general

Listado de figuras	III
Listado de tablas	IV
1. Objetivos	v
1.1. Alcance de los objetivos	VI
2. Introducción	vii
2.1. El problema de la reproducibilidad	VII
2.2. Clasificación de primarios	VIII
3. Fundamentos	IX
3.1. Reproducibilidad	IX
3.2. Tipos de reproducibilidad	XI
3.2.1. Aspectos críticos	XIII
3.3. Aspectos críticos	XIII
3.4. Proceso de ciencia de datos. ETL	XV
3.5. DevOps aplicado a Machine Learning. MLOps	XV
3.6. Autoencoders	XV
4. Planificación del trabajo	xvi
5. Presupuesto	xvii
6. Diseño del marco de trabajo	xviii
6.1. Herramientas utilizadas	XVIII
6.2. Estructura general	XVIII
6.3. Tracking de experimentos	XVIII
6.4. Hiperparametrización y entrenamiento distribuido	XVIII

6.5. Sistema de notificaciones y callbacks	XVIII
6.6. Interfaz Web	XVIII
6.7. Otras herramientas para la reproducibilidad	XVIII
6.8. Futuro desarrollo	XVIII
7. Diseño del autoencoder	XIX
8. Experimentos	XX
8.1. Cuantificación del ahorro del tiempo de desarrollo (opcional)	XX
8.2. Resultados	XX
9. Anexo: Manual de Usuario	XXI
Referencias	XXII

Listado de figuras

Figure 4.1 This is an example figure . . .	pp
Figure x.x Short title of the figure . . .	pp

Listado de tablas

Table 5.1 This is an example table . . .	pp
Table x.x Short title of the figure . . .	pp

Capítulo 1

Objetivos

El objetivo de este proyecto es el de desarrollar un marco de trabajo para machine learning enfocado en la reproducibilidad y buenas prácticas que explicaremos más adelante. Por otro lado, como objetivo secundario tenemos la aplicación de dicho framework para resolver un problema real.

A modo de resumen, los principales objetivos son:

- Diseño e implementación de un framework de reproducibilidad: El desarrollo de una herramienta que permita instrumentalizar proyectos de Machine Learning con mínimo esfuerzo, orientada a mantener unas buenas prácticas de desarrollo y seguir una filosofía MLOps. Dentro de este objetivo, de manera secundaria, incluimos una contribución de código a uno de los proyectos de código libre que componen el módulo central de nuestra herramienta, Mlflow.
- Especificación de buenas prácticas: La creación de una lista de pautas y requisitos necesarios para hacer reproducible un proyecto. Desde la recolección de datos hasta la gestión de experimentos.
- Aplicación de la herramienta a la resolución de un problema real: Este objetivo está orientado a la experimentación, trata de la aplicación de diferentes técnicas de Machine Learning tradicional y Deep learning para la resolución de un problema común en física, la detección de

primarios. En dicha aplicación, hacemos un uso extensivo de la herramienta y valoramos los beneficios y el coste en recursos de tiempo y capitales de su uso para este caso concreto.

1.1. Alcance de los objetivos

Para el primer objetivo, el alcance incluye el desarrollo integral de una herramienta en Python que permita cumplir con la mayoría de requisitos que consideramos necesarios para que un proyecto sea reproducible fácilmente por la comunidad científica. Esta herramienta debe ser flexible y permitir integrarse con frameworks de Machine Learning o Deep learning existentes, así como con proyectos orientados al análisis de datos exclusivamente en lugar de al modelado.

En relación con el primer objetivo, se debe desarrollar una especificación de buenas prácticas basadas en problemas existentes, con el objetivo de reducir aquella deuda técnica que concierne a este tipo de proyectos, tanto durante el desarrollo o experimentación, como en el momento de compartir el trabajo con otras personas. Estas buenas prácticas son bastante comunes en el desarrollo de software, pero no tanto en ciencia de datos, debido, entre otros motivos, a la heterogeneidad de perfiles que componen este campo. Dentro de esta relación entre el desarrollo de software y el desarrollo de proyectos de machine learning o ciencia de datos en general, se van tener en cuenta también aspectos relacionados con el despliegue e integración de software, lo que se conoce como DevOps, cuya aplicación al machine learning es más bien conocida como MLOps.

El tercer y último objetivo comprende el desarrollo de un proyecto de machine learning real, enfocado al modelado y a la experimentación. El alcance comprende el entendimiento del problema, procesamiento de datos, y modelado.

Capítulo 2

Introducción

2.1. El problema de la reproducibilidad

Hoy en día, los proyectos de ciencia de datos se desarrollan de una forma desestructurada en la mayoría de los casos, lo cual lo hace muy difícil de reproducir. Siendo conscientes de las dificultades que conlleva ser rigurosos con el desarrollo de este tipo de trabajos para asegurar la reproducibilidad, este trabajo presenta un framework que facilita el rastreo de experimentos y la operacionalización del machine learning, combinando tecnologías open source existentes y apoyadas fuertemente por la comunidad. Estas tecnologías incluyen Docker, Mlflow, Ray, entre otros.

El framework ofrece un flujo de trabajo opionionado para el diseño y ejecución de experimentos en un entorno local o remoto. Para facilitar la integración con código existente, se ofrece además un sistema de rastreo automático para los frameworks de Deep Learning más famosos: Tensorflow, Keras, Fastai, además de otros paquetes de Machine Learning como Xgboost y Lightgdm. Por otro parte, se ofrece un soporte de primera clase para el entrenamiento de modelos y la hyperparametrización en entornos distribuidos. Todas estas características se hacen accesibles al usuario por medio de un paquete de Python con el que instrumentalizar el código existente, y un CLI con el que empaquetar y ejecutar trabajos.

La reproducibilidad es un reto en la investigación moderna y produce bastante debate (Hutson 2018) (Freire et al. 2016) (Freire et al. 2012). Entre los diferentes tipos de trabajos reproducibles, este trabajo se centra en trabajos computacionales, desarrollando un flujo de trabajo específico basado en los principios de Control de Versiones, Automatización, Rastreo y Aislamiento del entorno (Olorisade et al. 2017) (Wilson 2017). El control de versiones permite rastrear los diferentes ficheros del proyecto y sus cambios, así como facilitar la colaboración. Automatizar los procesos, desde ficheros de shell hasta pipelines de alto nivel, permite que otra persona puede reproducir los pasos del trabajo fácilmente. Estos pasos incluyen: creación de ficheros, preprocesado de datos, ajuste de modelos, etc. Durante la ejecución de estos pasos, se generan gráficos, artifacts, nuevos datos, etc. Por este motivo, es necesario proporcionar una forma sistemática de recolectar toda esa información generada y mostrarla desde un único punto.

Finalmente, el aislamiento del sistema anfitrión mediante el uso de contenedores o máquinas virtuales, permite ampliar el ámbito de control sobre los experimentos, proporcionando un “escenario común” para la ejecución de los mismo. De otra forma, los factores externos al proyecto, como las versiones de las paquetes de análisis, los drivers de la GPU, o la propia versión del sistema operativo donde se ejecuten pueden incrementar la estocasticidad del experimento. Otra ventaja de aislar las dependencias y la imagen del sistema operativo (entre otros factores), combinado con la automatización de los diferentes procesos, es que que facilita enormemente la ejecución de los experimentos y los hace dependiente de la plataforma, evitando tener que instalar las diferentes dependencias, modificar ficheros de configuración, etc. Por no decir que las dependencias del proyecto pueden ser incompatibles con las globales instaladas en el sistema.

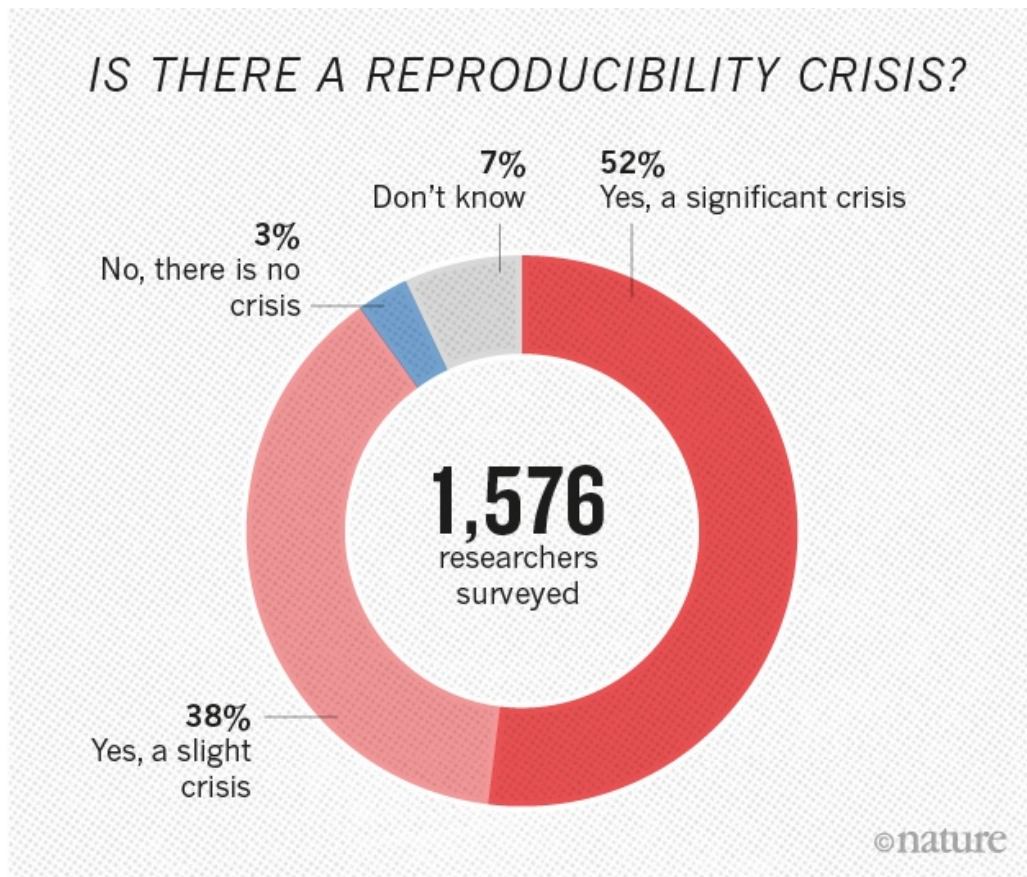
2.2. Clasificación de primarios

Capítulo 3

Fundamentos

3.1. Reproducibilidad

Según una encuesta realizada por Nature, una de las más prestigiosas revistas científicas a nivel mundial, más del 70 por ciento de los 1,576 investigadores encuestados no han podido reproducir alguno de sus propios experimentos. Además, los datos son claros, la mayoría piensa que existe una *crisis de reproducibilidad*.



A día de hoy, los estudios suelen ofrecer los resultados en forma de gráficas y tablas, pero en muchos casos carecen de la información necesaria para poder contrastar los resultados. Esta información suele ser, el entorno de ejecución, los datos originales y la implementación de los propios métodos (modelos, algoritmos, etc) entre otros. Para aumentar la accesibilidad de los estudios, los investigadores deben asegurarse de ofrecer esta información además de las gráficas y tablas.

La verificación independiente tiene como objetivo la confirmación de credibilidad y la extensión del conocimiento en un área. La investigación relativa al Machine Learning o a otras áreas donde se haga uso del mismo, no está exenta de este requisito de la investigación científica. Por tanto, adoptando un flujo de trabajo reproducible, estamos ofreciendo a la audiencia las herramientas necesarias que demuestran las decisiones tomadas y que permiten validar nuestros resultados. Por otro lado, para que un estudio computacional pueda ser reproducido correctamente por un investigador independiente

es necesario el acceso completo a los datos, código, parametros de los experimentos, información sobre el entorno de ejecución, etc.

Otro motivo de interés para la búsqueda de la reproducibilidad es el de facilitar el uso de nuestros métodos por el resto de la comunidad científica o incluso en aplicaciones comerciales. Ofreciendo acceso a los datos y al código, como se ha comentado antes, permitimos que nuestros métodos se puedan aplicar a otros problemas, tanto en investigación como para fines comerciales, así como facilita la extensión de nuestro trabajo.

En los últimos años nos hemos encontrado con muchos casos de publicaciones científicas que muestran resultados difíciles o incluso imposibles de reproducir. Este fenómeno se conoce como la crisis de la reproducibilidad, donde incluso estudios prominentes no se pueden reproducir. Este fenómeno ha estudiado de manera extensiva en otros campos, pero en el area del Machine Learning está tomando últimamente mucho importancia. Esto es debido a que tradicionalmente, los experimentos científicos se deben describir de tal forma que cualquiera pueda replicarlos, sin embargo, los experimentos computacionales tienen varias complicaciones que los hacen particularmente difíciles de replicar: versiones de software, dependencias concretas, variaciones del hardware, etc.

Con motivo de esta crisis de la reproducibilidad que afecta en gran medida a AI/ML, conferencias como NeurIPS han optado por añadir este factor en su proceso de revisión, e implementan políticas para alentar el código compartido. Por otro lado, algunos autores (incluido nosotros) han propuesto herramientas para facilitar la reproducibilidad, mientras que otros han propuesto una serie de reglas o heurísticas que para evaluar este aspecto.

3.2. Tipos de reproducibilidad

Para poder atajar de una manera directa y eficiente el problema de la reproducibilidad es necesario separarla en diferentes niveles. Esta separación nos permite desarrollar una serie de buenas prácticas y herramientas específicas para cada nivel, así como ver de una manera clara que aspectos se pueden

recoger en un framework común, y cuales son inherentes del estudio científico en cuestión. Entre los niveles de reproducibilidad podemos destacar:

- Reproducibilidad computacional: Cuando se provee con información detallada del código, software, hardware y decisiones de implementación.
- Reproducibilidad empírica: Cuando se provee información sobre experimentación empírica no computacional u observaciones.
- Reproducibilidad estadística: Cuando se provee información sobre la elección de los test estadísticos, umbrales, p-valores, etc.

Una vez hecha separación del problema en tres capas, podemos ver claramente que la reproducibilidad computacional debe ser nuestro objetivo a la hora de desarrollar el framework. Mientras que la reproducibilidad empírica se puede conseguir en mayor medida, haciendo los datos accesibles, la reproducibilidad estadística se consigue mediante el desarrollo de un diseño inicial del estudio. El establecer las bases estadísticas sobre las que se va a desarrollar el estudio de antemano, nos puede ayudar además a evitar problemas como el p-hacking.

The terms “reproducible research” and “reproducibility” are used in many different ways to encompass diverse aspects of the desire to make research based on computation more credible and extensible. Lively discussion over the course of the workshop has led to some suggestions for terminology, listed below. We encourage authors who use such terms in their work to clarify what they mean in order to avoid confusion. There are several possible levels of reproducibility, and it seems valuable to distinguish between the following:

- Reviewable Research. The descriptions of the research methods can be independently assessed and the results judged credible. (This includes both traditional peer review and community review, and does not necessarily imply reproducibility.)
- Replicable Research. Tools are made available that would allow one to duplicate the results of the research, for example by running the

authors' code to produce the plots shown in the publication. (Here tools might be limited in scope, e.g., only essential data or executables, and might only be made available to referees or only upon request.)

- **Confirmable Research.** The main conclusions of the research can be attained independently without the use of software provided by the author. (But using the complete description of algorithms and methodology provided in the publication and any supplementary materials.)
- **Auditable Research.** Sufficient records (including data and software) have been archived so that the research can be defended later if necessary or differences between independent confirmations resolved. The archive might be private, as with traditional laboratory notebooks.
- **Open or Reproducible Research.** Auditable research made openly available. This comprised well-documented and fully open code and data that are publicly available that would allow one to (a) fully audit the computational procedure, (b) replicate and also independently reproduce the results of the research, and (c) extend the results or apply the method to new problems.

3.2.1. ASPECTOS CRÍTICOS

- **Conjunto de datos:** La información sobre la localización y el proceso de extracción de los datos.
- **Preprocesado de datos:** Los diferentes pasos del proceso de transformación de los datos.
-

3.3. Aspectos críticos

- **Dataset:** Information about the location and the retrieval process of the dataset is needed to ensure access to the dataset as used in the study.

- Data preprocessing: The process of ridding the input data of noise and encoding it into a format acceptable to the learning algorithm. Explicit preprocessing information is the first step towards a successful reproduction exercise. An independent researcher should be able to follow and repeat how the data was preprocessed in the study. Also, it will be useful to find preprocessing output information to compare to e.g. final feature vector dimension.
- Dataset Partitions: Details of how the dataset was divided for use as training and test data.
- Model training: The process of fitting the model to the data. Making available, as much information as possible regarding every decision made during this process is particularly crucial to reproduction. Necessary information include but not limited to: 1. Study parameters 2. Proposed technique details – codes, algorithms etc. (if applicable)
- Model assessment: Measuring the performance of the model trained in 2. Similar information as in 2 applies here as well.
- Randomization control: Most operations of machine learning algorithms involves randomization. Therefore, it is essential to set seed values to control the randomization process in order to be able to repeat the same process again.
- Software environment: Due to the fact that software packages/modules are in continual development with possible alterations to internal implementation algorithms, it is important that the details of the software environment used (modules, packages and version numbers) be made available.
- Hardware environment (for large data volume): Some data intensive studies are only reproducible on the same machine capacity as was used to produce the original result. So, the hardware information are sometimes essential.

3.4. Proceso de ciencia de datos. ETL

3.5. DevOps aplicado a Machine Learning. MLOps

3.6. Autoencoders

Capítulo 4

Planificación del trabajo

- Planificación optimista
- Planificación real

Capítulo 5

Presupuesto

- Comparativa cluster propio vs AWS, Azure, GDC
- Coste de titulado superior (36€)

Capítulo 6

Diseño del marco de trabajo

- 6.1. Herramientas utilizadas
- 6.2. Estructura general
- 6.3. Tracking de experimentos
- 6.4. Hiperparametrización y entrenamiento distribuido
- 6.5. Sistema de notificaciones y callbacks
- 6.6. Interfaz Web
- 6.7. Otras herramientas para la reproducibilidad
- 6.8. Futuro desarrollo

Capítulo 7

Diseño del autoencoder

- Que es una red neuronal
- Que es un autoencoder
- Autoencoder simple
- Autoencoder profundo
- Autoencoder variacional

Capítulo 8

Experimentos

8.1. Cuantificación del ahorro del tiempo de desarrollo
(opcional)

8.2. Resultados

Capítulo 9

Anexo: Manual de Usuario

Referencias

- Freire, J., Bonnet, P. & Shasha, D., 2012. Computational reproducibility: State-of-the-art, challenges, and database research opportunities. In *Proceedings of the 2012 acm sigmod international conference on management of data*. SIGMOD '12. New York, NY, USA: Association for Computing Machinery, pp. 593–596. Available at: <https://doi.org/10.1145/2213836.2213908>.
- Freire, J., Fuhr, N. & Rauber, A., 2016. Reproducibility of Data-Oriented Experiments in e-Science (Dagstuhl Seminar 16041) J. Freire, N. Fuhr, & A. Rauber, eds. *Dagstuhl Reports*, 6(1), pp.108–159. Available at: <http://drops.dagstuhl.de/opus/volltexte/2016/5817>.
- Hutson, M., 2018. Artificial intelligence faces reproducibility crisis. *Science*, 359(6377), pp.725–726. Available at: <https://science.sciencemag.org/content/359/6377/725>.
- Olorisade, B.K., Brereton, P. & Andras, P., 2017. Reproducibility in machine learning-based studies: An example of text mining.
- Wilson, J.A.C., Greg AND Bryan, 2017. Good enough practices in scientific computing. *PLOS Computational Biology*, 13(6), pp.1–20. Available at: <https://doi.org/10.1371/journal.pcbi.1005510>.