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

Listado de tablas

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, procesado 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 casos, lo cual lo hacen 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 operacionalziación 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 caracteristicas 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 empaquetas 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 bastado en losn 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 sistematica de recolectar toda esa información generada y mostrarla desde un único punto.

Finalmente, el aislamiento del sistema anfitrion 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

2.3. Herramientas para el análisis de rayos cósmicos

- ROOT Framework
- Corkiska
- CERN

Capítulo 3

Fundamentos

3.1. Reproducibilidad

While the studies provide useful reports of their results, they lack information on access to the dataset in the form and order as used in the original study (as against raw data), the software environment used, randomization control and the implementation of proposed techniques. In order to increase the chances of being reproduced, researchers should ensure that details about and/or access to information about these factors are provided in their reports.

Independent verification of published claims for the purpose of credibility confirmation, extension and building a 'body of knowledge' is a standard scientific practice [13]. Machine learning methods based research are not excluded from this strict scientific research requirement. However, it may sometimes be hard or even impossible to replicate computational studies of this nature [12]. This is why the minimum standard expected of any computational study is for it to be reproducible [11]. In order for a study to be reproduced, an independent researcher will need at least full information and artefacts of the experiment - datasets, experiment parameters, similar software and hardware environment etc., as used in the original study. However, the experience in studies today shows a lack of sufficient information that can enable an independent researcher reproduce majority of the studies successfully.

3.2. 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.

Capítulo 4

Planificación del trabajo

- Planificación optimista
- Planificación real

Capítulo 5

Presupuesto

- Comparitiva 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.