

Experiencia de uso de Google Colaboratory en Sistemas Inteligentes

Leonardo Esnaola¹

¹ Licenciado en Sistemas, Universidad Nacional del Noroeste de la Provincia de Buenos Aires, Junín,
Argentina

leonardo.esnaola@itt.unnoba.edu.ar

Asignatura: Sistemas Inteligentes (UNNOBA)

Nombre del eje: Uso de herramientas tecnológicas aplicadas a la educación.

Resumen

Como docentes procuramos seleccionar adecuadamente los recursos más apropiados para sustentar nuestro proceso de enseñanza. Particularmente en lo que respecta a las herramientas de software, elegimos aquellas que sean lo suficientemente simples para implementarlas en el aula pero que, al mismo tiempo, también permitan que nuestras alumnas y alumnos las utilicen para consolidar su proceso de aprendizaje, lo que muchas veces requiere de un trabajo colaborativo con sus pares, o de la asistencia del equipo docente cuando algo no marcha bien. En este sentido, Google Colaboratory constituye una muy buena opción, pues se ofrece como un servicio gratuito de Google; proporciona una interfaz de Jupyter Notebook; todo el procesamiento se realiza en la nube, en hardware potente provisto y mantenido por Google; está orientado a la colaboración; y sólo requiere contar con un navegador Web y conexión a Internet. Esta narrativa describe la experiencia de su utilización en el marco de la asignatura Sistemas Inteligentes.

Palabras clave: Google Colaboratory; Sistemas Inteligentes; Herramientas tecnológicas.

1. Introducción

Sistemas Inteligentes es una asignatura con un importante enfoque práctico. Aborda técnicas inspiradas en procesos biológicos e inteligencia natural, como las redes neuronales artificiales (Krogh, 2008), los algoritmos evolutivos (Fogel, 2000) y de búsqueda local (Johnson, Papadimitriou, & Yannakakis, 1988), y los sistemas difusos (Zadeh, 1988).

Para la implementación de las actividades prácticas, se programa utilizando lenguaje Python (Python Software Foundation, s.f.) que, junto con R (The R Foundation, s.f.), constituye uno de los lenguajes de programación preferidos (Larose & Larose, 2019) para ciencia de datos (Cao, 2017) y *machine learning* (Mahesh, 2020). Además, se utilizan bibliotecas de código específicas, como *PyTorch*

(Facebook Open Source, s.f.) para la implementación de modelos redes neuronales artificiales; *Deap* (DEAP Project, s.f.) para computación evolutiva; *NumPy* (The NumPy project, s.f.) y *Pandas* (The Pandas project, s.f.) para tareas generales de manipulación de datos; *Matplotlib* (Matplotlib development team, s.f.) para crear y visualizar datos a través de gráficos; *scikit-learn* (scikit-learn developers, s.f.) para el análisis de datos; y algunas implementaciones específicas de algoritmos muy difundidos.

Además, las actividades introducidas pretenden no sólo que el alumnado resuelva los problemas presentados eficazmente, sino que además puedan explicar cada etapa de su proceso de resolución y las razones detrás de las decisiones tomadas. En este sentido, el entorno de desarrollo más apropiado resulta ser *Jupyter Notebook* (Project Jupyter, s.f.), por sus características de interactividad; por la posibilidad de combinar código ejecutable con descripciones textuales y con texto enriquecido, por ejemplo, LaTeX (The LaTeX Project, s.f.); por la posibilidad de incluir imágenes, gráficos y muchos otros recursos multimediales; y porque además es el formato preferido por los científicos de datos (Perkel, 2018).

Una de las primeras tareas encaradas en la asignatura, en este sentido, consiste en preparar el entorno de trabajo que cada alumna y alumno deberá utilizar durante el resto de la cursada. Una primera opción, en lugar de instalar cada componente necesario por separado, es utilizar alguna distribución que incorpore, en un mismo paquete, no sólo el lenguaje Python, sino también el entorno de desarrollo *Jupyter Notebook* y todas las bibliotecas de código necesarias (o al menos las más importantes). En este sentido, Anaconda (Anaconda Inc., s.f.) es una excelente opción, pues permite adicionalmente administrar paquetes de código, dando la posibilidad de instalar bibliotecas de código extra (y todas sus dependencias) utilizando un simple comando, que en general tiene la forma: “*conda install <nombre_de_biblioteca_a_instalar>*”.

Anaconda simplifica mucho el proceso, pues libera a las y los usuarios de tener que lidiar con la instalación de múltiples dependencias y los problemas que pueden generarse por las versiones de las mismas. Muchas veces sucede que nuevas dependencias entran en conflicto con las versiones disponibles y las requeridas desde distintas bibliotecas. Sin embargo, aún deben instalar software; necesitan contar con un equipo que cumpla con los requisitos mínimos; y deben contar con algún mecanismo para compartir un trabajo en proceso, algo que resulta muy beneficioso en el marco de esta asignatura, ya sea con colegas o con el equipo docente. Sin embargo, esta última tarea resulta todo un desafío, pues en general implica guardar el trabajo actual en un archivo con extensión “.ipynb” (es decir, un cuaderno de trabajo de IPython) y enviarlo a la persona en cuestión, quien deberá abrirlo localmente en su propio entorno de trabajo, que podría tener versiones instaladas de bibliotecas de código distintas a las del origen y, por ende, podrían presentarse errores al querer ejecutarlo. En fin, un proceso tedioso que podría exigir malgastar el tiempo resolviendo conflictos, en lugar de dedicar ese valioso tiempo para comprender y aplicar las técnicas y herramientas, que es lo que verdaderamente importa.

Así, con el objetivo de simplificar el proceso de preparar el entorno de trabajo y, más importante aún, de tener la posibilidad de compartir el trabajo realizado, e incluso de hacerlo de forma colaborativa, es que se propone la utilización de Google Colaboratory (Google Research, s.f.), o simplemente “Colab” como se lo suele referir.

La presente narrativa se organiza de la siguiente manera. En la sección 2 se realiza una descripción de la asignatura, con el objetivo de contextualizar el presente trabajo. En la sección 3 se realiza una descripción de Colab y de su utilización en el marco de la asignatura Sistemas Inteligentes. Finalmente, se incluyen las conclusiones.

2. Descripción de la asignatura

Sistemas Inteligentes es una asignatura perteneciente al 4° año de la carrera Ingeniería en Informática (Universidad Nacional del Noroeste de la Provincia de Buenos Aires, s.f.) de la Escuela de Tecnología de la UNNOBA. Tiene por objetivo introducir al alumnado en técnicas inspiradas en procesos biológicos e inteligencia natural, aplicables en la resolución de problemas complejos.

El dictado se organiza en clases teóricas y prácticas, implementando actividades incrementales, las cuáles se sustentan tanto en los conocimientos previos, como en la necesidad de integrar nuevos conocimientos a través de la experimentación e investigación. La exposición de cada tema consiste de un planteo por parte del docente de los aspectos teóricos-conceptuales involucrados, iniciando cada clase con una revisión breve de los puntos tratados en la clase anterior, para verificar la apropiación de los mismos por parte del alumnado. El tratamiento de los contenidos se basa en la exposición de los conceptos desde el conocimiento del docente; desde los recursos recomendados para su abordaje y desde la investigación de los alumnos bajo la dirección del docente. Se trata de una didáctica guiada, participativa desde la investigación, auto responsable desde la bibliografía, y siguiendo siempre un esquema comparativo-integrador. La práctica de la asignatura resulta fundamental para una correcta aprehensión de los conceptos, y para una correcta aplicación de las estrategias y las herramientas necesarias para la resolución de los diferentes problemas abordados.

La acreditación de la cursada se consigue a través de la entrega y la aprobación de todas las actividades obligatorias, o en su defecto de las actividades remediales previstas, y el desarrollo de un trabajo práctico integrador, de resolución individual y asincrónica, a través del cual se ponen en práctica todos los conocimientos adquiridos. Este trabajo representa el examen parcial de la asignatura y tiene otras dos instancias: un recuperatorio y un totalizador con la misma modalidad.

3. Utilización de Colab

Una preocupación que, como docentes, suele presentárcenos es la de seleccionar adecuadamente los recursos más apropiados para sustentar nuestro proceso de enseñanza. Particularmente, en referencia a las herramientas de software, procuramos elegir aquellas que sean lo suficientemente simples para implementarlas en el aula pero que, al mismo tiempo, también permitan que nuestras alumnas y alumnos las utilicen para consolidar su proceso de aprendizaje, lo que muchas veces requiere de un trabajo colaborativo con sus pares, o de la asistencia del equipo docente cuando algo no marcha bien. Para conseguir esto, es importante que las mismas cuenten con una interfaz simple, una buena comunidad de respaldo, y que estén orientadas para compartir y facilitar el trabajo colaborativo. Este último aspecto es especialmente importante cuando los contenidos abordados revisten una cierta complejidad, como es el caso de los temas tratados en Sistemas Inteligentes.

Otra preocupación, no menos importante que la primera esgrimida, es que estas herramientas puedan ser utilizadas sin requerir recursos informáticos potentes. Muchas veces no contamos con acceso a buenos equipos informáticos y, si bien es posible utilizar los que la institución dispone en sus laboratorios, la reciente pandemia de COVID nos demostró que a veces debemos utilizar los que tenemos disponibles en casa. Equipos que, en muchos casos, deben ser compartidos con otros miembros de la familia o que, simplemente, están un poco desactualizados para los requisitos actuales de la mayoría de las aplicaciones que necesitamos utilizar.

En este sentido, Colab constituye una excelente opción, dado que se ofrece como un servicio gratuito de Google; proporciona una interfaz de *Jupyter Notebook*; permite compartir el trabajo simplemente otorgando acceso al cuaderno de trabajo, como se hace habitualmente con los archivos de Google Drive; sólo se necesita contar con un navegador Web y conexión a Internet y, lo mejor de todo, tiene acceso al hardware provisto por Google. Así, no se necesita contar con ningún equipo potente, todo el procesamiento se realiza en la nube sobre *hardware* externo, ver Figura 1: Interfaz de Colab.

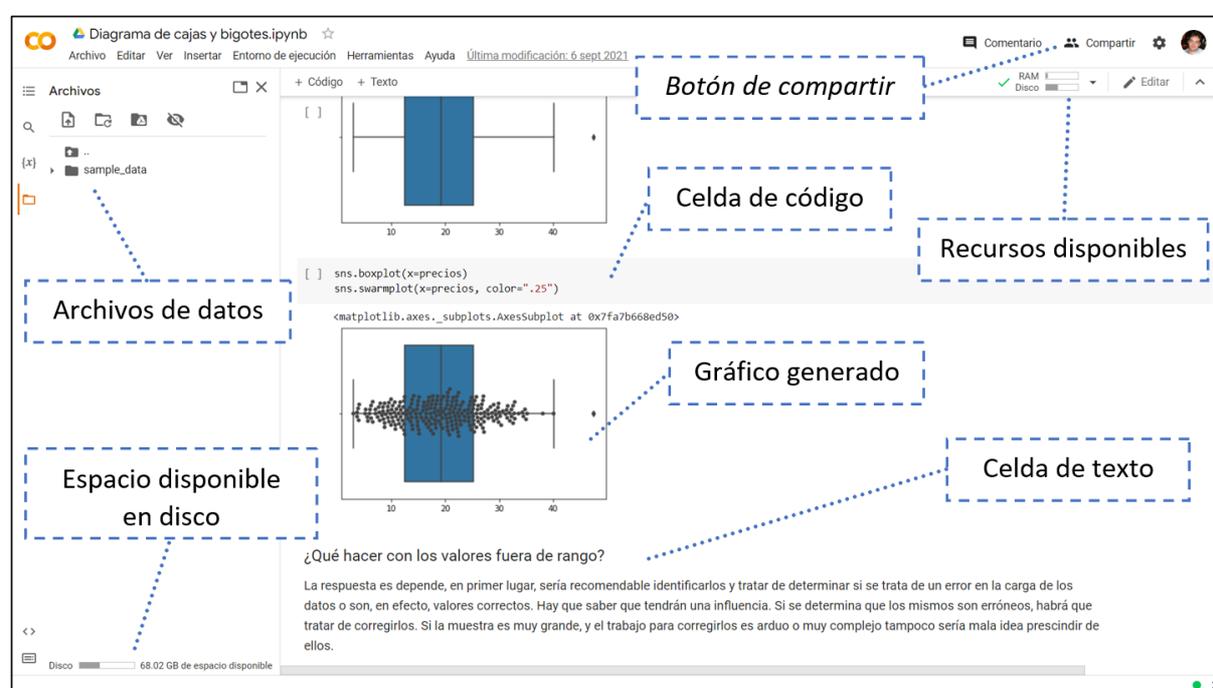


Figura 1: Interfaz de Colab

3.1. Características de Colab

Los cuadernos de trabajo (*Jupyter Notebooks*) de Colab se ejecutan en máquinas virtuales (MVs) basadas en Linux, que son proporcionadas y mantenidas por Google (Nelson & Hoover, 2020). El cálculo se puede realizar con unidades de procesamiento central, o se lo puede acelerar a través de unidades especializadas de procesamiento gráfico y de procesamiento de tensores.

El hardware disponible para cada MV varía según la sesión (Nelson & Hoover, 2020), pero generalmente incluye una GPU NVIDIA de primera línea (K80, T4 o P100); una CPU Intel® Xeon® de 2 a 2.3 GHz; alrededor de 8 a 13 GB de RAM; y de 39 a 70 GB de espacio libre en el disco duro de la

MV. Dado que los cuadernos de trabajo de Colab están diseñados para uso interactivo, en lugar de experimentos de ejecución prolongada, las MVs se desconectan después de un tiempo de inactividad y limitan las sesiones a 12 horas diarias. Sin embargo, estas características se pueden mejorar suscribiendo a los planes “Pro” o “Pro+” (Google Research, s.f.) que requieren un pago mensual.

Además de proporcionar recursos computacionales, las máquinas virtuales basadas en la nube de Colab están precargadas con paquetes comunes de inteligencia artificial (*numpy*, *torch*, *tensorflow*, etc.). Y, si alguna biblioteca de código no está precargada en la instalación, como el *framework Deap* que utilizamos para implementar computación evolutiva, se lo puede instalar fácilmente ejecutando en una celda del cuaderno de trabajo de Colab: “*!pip install deap*”.

Otro punto fuerte de Colab es su conexión nativa con el servicio Google Drive (Google, s.f.), lo que posibilita utilizar el propio espacio de almacenamiento en la nube para guardar los cuadernos de trabajo; los archivos de datos; los gráficos generados, entre otros; y compartirlos fácilmente con otros usuarios administrando los permisos de acceso, tal como ocurre con los demás documentos de Google Drive. Esta característica permite potenciar el proceso de aprendizaje, pues se puede compartir el trabajo en progreso y solicitar asistencia ante un error determinado, ya sea por parte del equipo docente o de colegas.

3.2. Utilización de Colab en Sistemas Inteligentes

Colab, en función de todas las características mencionadas, constituye una plataforma educativa. En la asignatura se lo ha utilizado para introducir contenidos complejos de forma empírica, teniendo la posibilidad de analizar paso a paso el comportamiento de los distintos algoritmos. Cuando compartimos un cuaderno de trabajo, los demás tienen la posibilidad de guardar una copia en su propio espacio de Google Drive, intervenirlos de una u otra forma, o ejecutarlos “en línea” sin necesidad de guardar los cambios. Esta característica resulta atractiva para, partiendo de una explicación, experimentar “qué pasaría si...”, sin temor a romper nada, lo que permite profundizar la aprehensión de los contenidos.

Además, Colab constituye el mecanismo recomendado para desarrollar las actividades de la asignatura. Los enunciados de los ejercicios de cada actividad se comparten en un cuaderno de trabajo que acompaña a la consigna. Cada alumna o alumno desarrolla la resolución de los ejercicios presentados en una copia personal del mismo. Finalmente, pueden subir el cuaderno de trabajo resultante a través de una actividad “tarea” de la Plataforma de Educación Digital, o podrían directamente compartir su cuaderno de trabajo con el equipo docente.

Conclusiones

Google Colab constituye un gran aliado como herramienta para potenciar nuestras clases, cuando se requiere trabajar con Python y con bibliotecas de código especializadas. Libera al alumnado, y al equipo docente, de tener que lidiar con la instalación de software y de mantenerlo actualizado. Además, no requiere contar con equipos potentes, dado que todo el procesamiento se realiza en la nube, en hardware provisto por Google de forma gratuita. Sólo se requiere contar con un navegador Web y conexión a Internet. Además, está orientado a la colaboración, permitiendo compartir fácilmente los

cuadernos de trabajo. Esta última, considero que es una de las mayores ventajas de la plataforma, pues resulta fundamental para potenciar el proceso de enseñanza-aprendizaje.

Se ofrece hace ya varios años como servicio gratuito. Una posible desventaja sería que en algún momento se transformase en un servicio exclusivamente pago. Sin embargo, dado que Colab se basa en *Jupyter Notebooks*, sería posible migrar los cuadernos de trabajo a otros proveedores o utilizarlos en una instalación local. Incluso, también sería posible alojar los cuadernos de trabajo en MVs en la infraestructura propia de la Universidad, utilizando la plataforma *JupyterHub* (GitHub, Inc., s.f.) de código abierto.

Habiendo utilizado antes instalaciones locales del lenguaje Python y de las bibliotecas de código necesarias, se puede afirmar que Colab solucionó el problema de lidiar con la instalación del software, y no sólo para el alumnado y para el equipo docente, sino también para quienes gestionan la infraestructura de los laboratorios. Ya no se necesita solicitar la instalación de ningún software adicional para el correcto desarrollo de la asignatura. Por otro lado, el contar con la posibilidad de compartir fácilmente los cuadernos de trabajo permitió dinamizar el proceso de tutorización. El alumnado comparte sus cuadernos con el equipo docente para resolver sus dudas y solucionar errores que podrían presentárseles. El equipo docente, por su parte, puede brindar sus cuadernos como recursos y, quitando el permiso de edición, tiene la tranquilidad de que nadie los modifique inadvertidamente.

Referencias

Anaconda Inc. (s.f.). *Data science technology for a better world*. Obtenido de <https://www.anaconda.com/>

Cao, L. (2017). Data science: challenges and directions. *Communications of the ACM*, 60(8), 59 - 68.

DEAP Project. (s.f.). *Distributed Evolutionary Algorithms in Python (DEAP)*. Obtenido de <https://deap.readthedocs.io/>

Facebook Open Source. (s.f.). *PyTorch*. Obtenido de <https://pytorch.org/>

Fogel, D. (Febrero de 2000). What is evolutionary computation? *IEEE Spectrum*, 37(2), 26 - 32.

GitHub, Inc. (s.f.). *JupyterHub*. Obtenido de <https://github.com/jupyterhub/jupyterhub>

Google. (s.f.). *Google Drive - Acceso sencillo y seguro a todo tu contenido*. Obtenido de <https://www.google.com/intl/es/drive/>

Google Research. (s.f.). *Elige el plan de Colab adecuado para ti*. Obtenido de <https://colab.research.google.com/signup>

Google Research. (s.f.). *Te damos la bienvenida a Colaboratory*. Obtenido de <https://colab.research.google.com/>

Johnson, D. S., Papadimitriou, C. H., & Yannakakis, M. (Agosto de 1988). How easy is local search? *Journal of Computer and System Sciences*, 37(1), 79-100.

Krogh, A. (2008). What are artificial neural networks? *Nature Biotechnology*, pages195–197.

Larose, C. D., & Larose, D. T. (2019). *Data Science using Python and R*. Hoboken: Wiley.

Mahesh, B. (2020). Machine learning algorithms-a review. *International Journal of Science and Research (IJSR)*, 381-386.

Matplotlib development team. (s.f.). *Matplotlib: Visualization with Python*. Obtenido de <https://matplotlib.org/>

Nelson, M. J., & Hoover, A. K. (Junio de 2020). Notes on Using Google Colaboratory in AI Education. *ITiCSE '20: Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education*, 533–534.

Perkel, J. M. (Noviembre de 2018). Why Jupyter is data scientists' computational notebook of choice. *Nature*, 563(7732), 145+.

Project Jupyter. (s.f.). *Jupyter - Free software, open standards, and web services for interactive computing across all programming languages*. Obtenido de <https://jupyter.org/>

Python Software Foundation. (s.f.). *Python*. Obtenido de <https://www.python.org/>

scikit-learn developers. (s.f.). *scikit-learn: Machine Learning in Python*. Obtenido de <https://scikit-learn.org/>

The LaTeX Project. (s.f.). *LaTeX – A document preparation system*. Obtenido de <https://www.latex-project.org/>

The NumPy project. (s.f.). *NumPy - The fundamental package for scientific computing with Python*. Obtenido de <https://numpy.org/>

The Pandas project. (s.f.). *pandas*. Obtenido de <https://pandas.pydata.org/>

The R Foundation. (s.f.). *The R Project for Statistical Computing*. Obtenido de <https://www.r-project.org/>

Universidad Nacional del Noroeste de la Provincia de Buenos Aires. (s.f.). *Carrera: 045 - INGENIERO EN INFORMÁTICA - Plan: P0455 (Version 1)*. Obtenido de <https://programas.unnoba.edu.ar/planes.xhtml?version=306&plan=P0455&carrera=045>

Zadeh, L. (Abril de 1988). Fuzzy logic. *Computer*, 21(4), 83 - 93.