Uso de ambientes reproducibles para la validación de modelos de aprendizaje estadístico. Una implementación en Docker.

Fabricio Machado

Keywords— Reproducibilidad, Portabilidad, Análisis Discriminante, Validación Cruzada, MLOps

Abstract

Este trabajo presenta la implementación de un modelo de aprendizaje estadístico a partir de la herramienta Docker. Se combinan técnicas de clustering y análisis discriminante para clasificar individuos en grupos sociales. Para validar dicha clasificación y garantizar reproducibilidad se recurrió al uso de ambientes reproducibles en Docker. La relevancia de este trabajo no se limita unicamente a los resultados del modelo y sus aportes a la ciencia social, sino también a la experiencia generada en el uso de herramientas modernas y sus potencialidades.

1 Modelo de aprendizaje estadístico

Utilizamos un modelo discriminante para clasificar individuos en grupos sociales a partir de variables relevantes de la Encuesta Continua de Hogares del Instituto Nacional de Estadística. Los grupos están definidos a partir de técnicas de clusterización sobre un año base, y se aplica el análisis discriminante en los demás años para lograr una misma caracterización a lo largo de un período. En el **análisis discriminante** derivamos una regla para asignar las observaciones externas (los datos fuera del año base) a los grupos haciendo mínima la probabilidad de clasificar incorrectamente. Entre las distintas funciones discriminantes se eligió el discriminante logístico dado que todas las variables son cualitativas (Peña, 2002).

Como se cuenta con más de dos grupos, suponemos que la variable que indica las subpoblaciones proviene de una distribución multinomial. Se presentan a continuación las probabilidades logarítmicas entre grupos (James et al., 2021):

$$log(\frac{Pr(Y=k|X=x)}{Pr(Y=K|X=x)}) = \beta_{k0} + \beta_{k1}x_1 + ... + \beta_{kp}x_p$$

Los individuos se asignan a un grupo o a otro comparando las probabilidad a priori, con esta regla de clasificación el grupo más probable es aquel donde su verosimilitud es más alta. Para evaluar el modelo se utiliza una medida de precisión definida como la cantidad de datos bien clasificados sobre el total, valores altos de la misma implican un buen desempeño del modelo. Como es de interés clasificar datos externos, para evitar problemas de sobreidentificación se utiliza el método de **validación cruzada** $k - folds$ (James et. al., 2021).

El trabajo fue principalmente realizado en R, desde el procesamiento de datos con la librería tidytable y la implementación de los métodos con la librería **nnet** (v7.3-19; Venables y Ripley, 2002). Adicionalmente se crearon las funciones en R necesarias para la aplicación de la validación cruzada.

2 Implementación

Dado el volumen de datos y la intensidad computacional requerida, se optó por el uso de Docker para garantizar la reproducibilidad y portabilidad del entorno de desarrollo. Docker nos permitió encapsular todas las dependencias del proyecto dentro de un contenedor, asegurando que el proceso pueda ejecutarse de manera consistente en diferentes entornos, ya sea en servidores en la nube, en un servidor de cómputo local o en computadoras personales con diferentes sistemas operativos (Matthias y Kane, 2015).

El ambiente de trabajo se configuró mediante un **Dockerfile** que define todas las herramientas y bibliotecas necesarias para la ejecución del modelo. Este archivo incluye desde la instalación de R y sus paquetes relevantes, hasta la configuración de las dependencias del sistema operativo. De esta forma, cualquier usuario puede reproducir el ambiente exacto simplemente construyendo y ejecutando el contenedor.

Para optimizar el rendimiento y facilitar la integración del modelo en entornos de producción, se aplicaron principios de Machine Learning Operations (MLOps). **MLOps** es un enfoque que combina prácticas

de desarrollo de software (DevOps) con metodologías específicas de Machine Learning, con el objetivo de automatizar, escalar y gestionar los ciclos de vida de los modelos de aprendizaje automático de manera eficiente.

Dentro de este marco, el procesamiento en paralelo jugó un papel crucial. Se utilizó el paquete **parallel** (R Core Team, 2023) de R para distribuir el procesamiento de datos y la validación cruzada del modelo en múltiples núcleos de CPU, lo que permitió acelerar significativamente los tiempos de ejecución y manejar grandes volúmenes de datos de manera más efectiva. La integración de estas prácticas asegura que el modelo no solo sea reproducible, sino también escalable y adaptable a diferentes entornos.

Ambientes y ejecución

Con el objetivo de facilitar la colaboración y el acceso al proyecto, se habilitaron ambientes de desarrollo en **Gitpod** y **GitHub Codespace**. Estos servicios permiten a los colaboradores acceder a un entorno preconfigurado en la nube, que ya incluye el contenedor Docker con todas las dependencias necesarias. Esta configuración asegura que todos los participantes trabajen en un entorno idéntico, minimizando los problemas relacionados con la configuración del ambiente de desarrollo.

Para ejecutar el modelo en un entorno de producción, se utilizó una instancia de Amazon EC2, aprovechando su capacidad de escalabilidad y recursos de cómputo. Se configuró la instancia para ejecutar el contenedor Docker previamente construido. Una vez finalizada la ejecución del modelo, los resultados fueron descargados localmente para su análisis posterior. Este enfoque permitió realizar cálculos intensivos en un entorno controlado y luego procesar y revisar los resultados de manera local, asegurando la reproducibilidad y eficiencia del proceso.

3 Resultados y conclusiones finales

La implementación del modelo discriminante alcanzó una **precisión del 94.4%**, demostrando un excelente desempeño en la clasificación de los individuos en los grupos sociales definidos. Además, este trabajo nos permitió aprender y aplicar herramientas modernas como Docker, Gitpod y Amazon EC2, lo que facilitó la reproducibilidad y escalabilidad del proyecto.

Hemos comprobado que Docker puede ser una herramienta valiosa en la investigación en Estadística y Ciencias Sociales, ya que permite encapsular ambientes de desarrollo completos, asegurando que los experimentos puedan replicarse de manera consistente en diferentes entornos. Sus principales fortalezas son la reproducibilidad, el aislamiento de dependencias, la portabilidad y la eficiencia computacional. Particularmente es beneficioso incorporar esta herramienta cuando se trabaja con combinaciones de datos y métodos que requieren gran exigencia computacional y se detectan espacios para paralelizar los procesos.

Recomendamos el uso de Docker y su complementación con otras herramientas modernas.

Referencias:

- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112, p. 18). New York: springer.
- Matthias, K., & Kane, S. P. (2015). Docker: Up & Running: Shipping Reliable Containers in Production. " O'Reilly Media, Inc.".
- Peña, D. (2002). Análisis de datos multivariantes (Vol. 24). Madrid: McGraw-hill.
- R Core Team (2023). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. ¡https://www.R-project.org/¿.
- Venables WN, Ripley BD (2002). Modern Applied Statistics with S, Fourth edition. Springer, New York. ISBN 0-387-95457-0, https://www.stats.ox.ac.uk/pub/MASS4/.