

Econometría aplicada y ética medioambiental*

Applied econometrics and environmental ethics

Javier Manuel CAPILLA ROMEROSA
Universidad Complutense de Madrid
capillaromerosa@gmail.com

Resumen: En el presente trabajo se explora la relación entre la ética medioambiental y la econometría aplicada, poniendo el foco en el consumo energético del ajuste de los modelos econométricos. Como ejemplo de práctica deseable, se ha implementado en Python el algoritmo RETINA siguiendo un enfoque de Green AI. Se han eliminado cuellos de botella que tenía el algoritmo original y añadido la posibilidad de medir las emisiones de CO₂ del entrenamiento del algoritmo. Además, se han añadido nuevas funcionalidades que no poseía el procedimiento original. Se ha llamado a esta nueva versión Eco-RETINA.

Abstract: The present work explores the relationship between environmental ethics and applied econometrics, focusing on the energy consumption of econometric models training. As an example of good practices, the RETINA algorithm has been programmed in Python following a Green AI approach. Bottlenecks present in the original algorithm have been eliminated, and the ability to measure CO₂ emissions during the algorithm's training has been added. Additionally, new functionalities that were not present in the original procedure have been incorporated. This new version has been named Eco-RETINA.

Palabras clave: Ética medioambiental. Consumo energético. RETINA. Green AI. Eco-RETINA.

* Este artículo obtuvo, el pasado curso académico 2022-2023, el Premio Reina María Cristina, en su novena edición, en la modalidad de Economía. El jurado calificador estuvo formado por: D. Carlos Rivero Rodríguez, Decano de la Facultad de Ciencias Económicas y Empresariales de la Universidad Complutense de Madrid; D^a María Elena Martínez Rodríguez, Vicedecana de Ordenación Académica de la Facultad de Ciencias Económicas y Empresariales de la Universidad Complutense de Madrid; D. Luis Antonio Puch González, Catedrático del Instituto Complutense de Análisis Económico; P. Agustín Alonso Rodríguez, Decano de Administración y Dirección de Empresas del “Real Centro Universitario Escorial María Cristina”, en calidad de Secretario.

Keywords: Environmental ethics. Energy consumption. RETINA. Green AI. Eco-RETINA.

Sumario:

I. Introducción.

II. Ética medioambiental.

III. Economía y ética medioambiental.

IV. El enfoque de Green AI.

V. RETINA.

5.1. El procedimiento de RETINA.

5.2. RETINA y Green AI.

VI. Eco-RETINA.

VII. El estimador y la interfaz de Eco-RETINA.

7.1. Requisitos de uso.

7.2. El estimador.

7.3. Ejemplo de uso del estimador.

7.4. La interfaz de usuario.

VIII. Caso práctico.

8.1. Datos.

8.2. Metodología de los experimentos.

8.3. Hiperparámetros y arquitectura.

8.4. Comparación de resultados de Eco-RETINA y RETINA.

8.5. Comparación de Resultados de Eco-RETINA y una red neuronal.

IX. Conclusiones.

X. Bibliografía.

I. INTRODUCCIÓN

El coste energético del entrenamiento de modelos de econometría ha experimentado un aumento significativo en los últimos años. Esto es debido a la mayor complejidad y tamaño de los modelos, a la utilización de técnicas avanzadas como redes neuronales, al uso intensivo de Big Data y la demanda de mayor precisión, que requieren recursos computacionales significativos. El empleo de hardware avanzado como GPUs y TPUs, aunque más eficiente en tiempo, también contribuye al alto consumo energético. Como resultado, se ha vuelto crucial para la comunidad de investigación y la industria de la econometría y de la inteligencia artificial, en general, explorar y desarrollar tanto nuevos algoritmos como nuevas técnicas de entrenamiento de modelos econométricos y de redes neuronales que sean más eficientes energéticamente.

No es la primera vez que urge encontrar alternativas y nuevos enfoques en el contexto de los modelos de redes neuronales. A pesar de los avances que éstas experimentaron en los años 80 gracias a la introducción del perceptrón multicapa y del algoritmo de retropropagación del gradiente, su uso se veía limitado debido a la falta de capacidad computacional y a elevados tiempos de entrenamiento debido a la lenta convergencia. En este contexto, en el año 2004, surgió RETINA. Este algoritmo no sólo ofrecía una capacidad predictiva similar a la de las redes neuronales, sino que también requería tiempos de entrenamiento más cortos y menor capacidad computacional para ejecutarlo. Además, el procedimiento era transparente y sus resultados podían incluso ser interpretados. Esto suponía una ventaja adicional frente a las redes neuronales, que son un algoritmo de “caja negra”.

En la década del 2010, las redes neuronales experimentaron un notable éxito, debido en gran medida a la superación de los problemas mencionados. La duración del entrenamiento de las redes neuronales se vio reducida gracias a numerosos avances como fueron potentes recursos de computación paralela, nuevas técnicas; como la normalización de lotes, algoritmos de descenso de gradiente mejorados; como Adam y al desarrollo del aprendizaje por transferencia, entre otros. RETINA, sin embargo, no tuvo la misma suerte. A pesar de las ventajas que ofrecía, su falta de popularidad jugó en su contra. Aunque RETINA

exigía menos recursos computacionales que las redes neuronales, los tiempos de ejecución del algoritmo también eran considerables, lo que dificultaba su uso. El hecho de que estuviera pensado para usarse con grandes bases de datos, también dificultó su difusión.

Este trabajo tiene dos objetivos. El primero es promover la implementación y el uso de algoritmos en econometría e inteligencia artificial que se adhieran a los principios de la ética medioambiental. Para ello, se toma un algoritmo econométrico como RETINA y se modifica e implementa desde un enfoque de Green AI. El segundo objetivo consiste en promover el uso de RETINA, ya que, como se verá, es un algoritmo coherente con la filosofía de Green AI. Para lograr este objetivo, se ha implementado RETINA en un lenguaje de programación popular, como Python. Se han usado técnicas de programación avanzadas como son el paralelismo y las operaciones vectorizadas. También se han añadido nuevas transformaciones y características, entre las que destaca la medición de la energía consumida al entrenar al algoritmo; y se han eliminado potenciales cuellos de botella, reduciendo así su coste energético. Además, se ha realizado un caso práctico en el que se compara el rendimiento de RETINA con el de las redes neuronales para mostrar que RETINA puede competir tanto en poder predictivo como en sostenibilidad con los algoritmos actuales.

II. ÉTICA MEDIOAMBIENTAL

Como explica Cochrane¹, el campo de la ética medioambiental se refiere a la relación ética de los seres humanos con el medio ambiente natural. Aunque numerosos filósofos han escrito sobre este tema a lo largo de la historia, la ética medioambiental realmente solo se desarrolló como una disciplina filosófica específica en la década de 1970. La razón de este surgimiento sin duda se debió al aumento de la conciencia en la década de 1960 sobre los efectos que la tecnología, la industria, la expansión económica y el crecimiento poblacional estaban teniendo en el medio ambiente. El desarrollo de esta conciencia fue facilitado por la publicación en este momento de dos libros importantes. “Primavera silenciosa” de Rachel Carson, publicado por primera vez en 1962, alertó a los lectores sobre cómo el uso generalizado de pesticidas químicos estaba representando una seria amenaza para la salud pública y también estaba llevando a la destrucción de la vida silvestre. De similar importancia fue el libro de 1968 de Paul Ehrlich, “La explosión demográfica”, que advirtió sobre los efectos devastadores en los recursos del planeta de una población humana en espiral.

¹ COCHRANE, A., *Environmental ethics. Internet encyclopedia of philosophy*, 2006.

Por supuesto, la contaminación y la disminución de los recursos naturales no han sido las únicas preocupaciones medioambientales desde entonces: la disminución de la biodiversidad de plantas y animales, la pérdida de la naturaleza salvaje, la degradación de los ecosistemas y el cambio climático son parte de una serie de problemas “verdes” que se han implantado tanto en la conciencia pública como en la política pública en los años siguientes. El trabajo de la ética medioambiental es delinear nuestras obligaciones morales frente a tales preocupaciones. En pocas palabras, las dos preguntas fundamentales que la ética medioambiental debe abordar son: ¿qué deberes tienen los seres humanos con respecto al medio ambiente y por qué? Por lo general, la segunda pregunta debe considerarse antes que la primera; para abordar qué obligaciones tenemos, generalmente se considera necesario primero analizar por qué las tenemos. Por ejemplo, ¿tenemos obligaciones medioambientales por el bien de los seres humanos que viven en el mundo hoy, por los humanos que vivirán en el futuro o por el bien de las entidades dentro del medio ambiente mismo, independientemente de cualquier beneficio humano? Diferentes filósofos han dado respuestas bastante diferentes a esta pregunta fundamental que, lo que ha llevado al surgimiento de diversas corrientes. Mientras que algunos pensadores han defendido el antropocentrismo, otros sostienen que nuestras obligaciones morales deberían extenderse más allá de los seres humanos, otorgando consideración moral a otras formas de vida, como los animales. La cuestión de qué deberes tienen los seres humanos con respecto al medio ambiente también ha contribuido al nacimiento de varias extensiones de la ética ambiental. Las principales son la extensión libertaria; que promueve el derecho del individuo a utilizar los recursos naturales, la extensión ecológica; que alienta a los humanos a trabajar en armonía con la naturaleza, y la ética de la conservación; que enfatiza el uso sostenible de los recursos naturales.

Pese a la gran variedad de corrientes, la ética medioambiental tiene unos principios fundamentales que todas sus ramas comparten. De acuerdo a GeeksforGeeks, 2023², estos son:

- Respeto por el valor intrínseco de la naturaleza: La naturaleza no debe ser tratada como una mercancía o un recurso para ser explotado y descartado.
- Interdependencia de especies y ecosistemas: Los humanos dependen de la naturaleza y los sistemas naturales. Debemos reconocer nuestro papel en la preservación y protección del medio ambiente.

² GEEKSFORGEEEKS, *Environmental Ethics: Types, Importance, Examples*, 2023: <https://www.geeksforgeeks.org/environmental-ethics/>

- Sostenibilidad ecológica: Debemos esforzarnos por usar los recursos de manera responsable y con miras a preservar los ecosistemas y la biodiversidad.
- Responsabilidad humana: Somos responsables de nuestras propias acciones y decisiones y de sus consecuencias para el medio ambiente.
- Equidad humana: Debemos esforzarnos por un mundo justo donde se respeten y protejan los derechos y las necesidades de los humanos, los animales y las plantas.
- Principio de precaución: Debemos tomar precauciones contra el daño ambiental, incluso cuando la evidencia científica sea inconclusa.
- Derecho a saber: Las personas tienen derecho a acceder a información sobre cuestiones ambientales.
- Derecho a participar: Los ciudadanos tienen derecho a participar en los procesos de toma de decisiones ambientales.

III. ECONOMÍA Y ÉTICA MEDIOAMBIENTAL

La relación entre la economía y la ética medioambiental es muy estrecha. Para ver la gran influencia de la economía en la ética ambiental basta con leer el informe Brundtland. Este informe, oficialmente titulado “Nuestro Futuro Común” (Our Common Future), es un documento publicado en 1987 por la Comisión Mundial sobre el Medio Ambiente y el Desarrollo de las Naciones Unidas, presidida por la ex primera ministra de Noruega, Gro Harlem Brundtland. Este documento es ampliamente reconocido por haber popularizado el concepto de desarrollo sostenible y por su influencia en la formulación de políticas ambientales y de desarrollo a nivel global. El informe Brundtland propone definir todos los objetivos de desarrollo económico y social en todos los países en términos de desarrollo sostenible, el cual como aquel que satisface las necesidades del presente sin comprometer la capacidad de las futuras generaciones para satisfacer sus propias necesidades. Esta definición contiene dos conceptos clave:

El concepto de “necesidades”, en particular las necesidades esenciales de los pobres del mundo, a las que se debe dar prioridad absoluta; y la idea de las limitaciones impuestas por el estado de la tecnología y la organización social sobre la capacidad del medio ambiente para satisfacer las necesidades presentes y futuras.

Keeble³ también argumenta que “el mundo industrial ya ha utilizado gran parte del capital ecológico del planeta. Esta desigualdad es el principal problema 'ambiental' del planeta; también es su principal problema 'de desarrollo’”. En el concepto de desarrollo sostenible, el informe combina la noción del economista de recursos de “rendimiento sostenible” con el reconocimiento de que los países en desarrollo del mundo tienen derecho al crecimiento económico y la prosperidad. De acuerdo a Brennan y Lo, 2016⁴, la noción de rendimiento sostenible implica pensar en los bosques, ríos, océanos y otros ecosistemas, incluidas las especies naturales que viven en ellos, como un conjunto de “capital ecológico” del cual fluyen todo tipo de bienes y servicios. Siempre que el flujo de tales bienes y servicios no reduzca la capacidad del propio capital para mantener su productividad, se considera que el uso de los sistemas en cuestión es sostenible. Por lo tanto, Keeble, 1988 argumenta que “el rendimiento máximo sostenible debe definirse después de tener en cuenta los efectos sistémicos de la explotación” del capital ecológico.

El lenguaje del informe demuestra la gran influencia de la economía en la ética medioambiental al utilizar términos como capital ecológico y rendimiento sostenible. De hecho, el mismo concepto de desarrollo sostenible bebe de la influencia de economistas como John Stuart Mill y Thomas Malthus. John Stuart Mill distinguió entre el estado estacionario y el estado progresivo y argumentó que al final del estado progresivo yace el estado estacionario, ya que el aumento de la riqueza no es ilimitado. Por su parte, Thomas Malthus es conocido por conjeturar que la población tiende a aumentar geométricamente mientras que los recursos alimentarios aumentan solo aritméticamente en el mejor de los casos, por lo que la demanda de alimentos eventualmente superará la oferta. No obstante, el informe Brundtland pone menos énfasis en los límites que lo hacen Mill, Malthus. En él la sostenibilidad se representa como un desafío y una oportunidad para que el mundo se vuelva más justo social, política y ambientalmente.

Del mismo modelo que la ética medioambiental está muy influenciada por la ciencia económica, la ética medioambiental también está muy presente en la economía. Por ejemplo, los principios de la ética ambiental están muy presentes en la economía pública y se tienen en cuenta a la hora de plantear soluciones a fallos de mercado como externalidades y bienes públicos. De hecho, la importancia del desarrollo sostenible ha llevado a algunos economistas a abrazar planteamientos decrecentistas los cuales proponen una reducción

³ KEEBLE, B. R., “The Brundtland report: «Our common future»”, en *Medicine and war*, 4(1) (1988) 17-25.

⁴ BRENNAN, A., & LO, Y., *Environmental Ethics Stanford Encyclopedia of Philosophy* ed EN Zalta 2016.

deliberada de la producción y el consumo de bienes y servicios, con el objetivo de minimizar los impactos ambientales y sociales negativos asociados con el crecimiento económico sin restricciones.

Entre las distintas ramas de la economía, la econometría está especialmente ligada a la ética medioambiental, ya que proporciona las herramientas y técnicas para cuantificar el impacto económico de diversas acciones y políticas en el medio ambiente. Además, como muestran Castle y Hendry⁵, 2022 la econometría también puede usarse para modelar el cambio climático. Sin embargo, entrenar algunos modelos econométricos, también pueden presentar un coste energético considerable. No ha sido hasta recientemente, con el surgimiento del enfoque de Green AI cuando se ha empezado a considerar el coste energético y no sólo el error de predicción en la elección de modelos econométricos.

IV. EL ENFOQUE DE GREEN AI

En este apartado se profundizará en el término Green AI para evitar confusiones y entender mejor su enfoque. Schwartz et al., 2020⁶ manifiestan que el concepto de Green AI se refiere a la investigación en IA que brinda nuevos resultados sin incrementar el coste computacional e, idealmente, lo reduce. Barbierato y Gatti, 2024⁷ explican que la Green AI logra su objetivo gracias al uso de conjuntos de datos más pequeños, técnicas de entrenamiento de los modelos menos intensas computacionalmente o recursos de energía renovables. Además, añaden que el enfoque de Green AI enfatiza en la eficiencia y la sostenibilidad y no en la precisión y el rendimiento. Schwartz et al., 2020 declaran la importancia de las métricas de eficiencia, además de las de predicción, para evaluar los modelos, de manera que los investigadores puedan poner el foco en la eficiencia de sus modelos que tengan un impacto medioambiental positivo.

El concepto opuesto al de Green AI es el conocido como Red AI. Schwartz et al., 2020 explican que este concepto se refiere a la investigación en IA que busca obtener novedosos resultados en términos de precisión o de métricas similares a través del uso de poder computacional masivo. Así, el enfoque de Red AI “compra” sus resultados. Aunque a través de la Red AI se han alcanzado importantes resultados, este enfoque tiene importantes limitaciones. En general,

⁵ CASTLE, J. L., & HENDRY, D. F., “Econometrics for modelling climate change”, en *Oxford Research Encyclopedia of Economics and Finance*, 2022.

⁶ Schwartz, R., Dodge, J., Smith, N. A., & Etzioni, O., “Green ai”. *Communications of the ACM*, 63(12) (2020) 54-63.

⁷ BARBIERATO, E., & GATTI, A., *Towards Green AI. A methodological survey of the scientific literature*. IEEE Access, 2024.

la relación entre el rendimiento de un modelo y su complejidad (medida como el número de parámetros o tiempo de ajuste) es, en el mejor de los casos, logarítmica, por lo que para obtener una ganancia lineal en el rendimiento se necesita un modelo que sea exponencialmente mayor. Resultados similares se dan para el aumento del tamaño del conjunto de datos de entrenamiento y del número de experimentos. En estos casos, resultados con cada vez peor rendimiento requieren un creciente coste computacional.

Una cuestión que no siempre queda clara es la distinción entre Green AI e inteligencia artificial aplicada a problemas medioambientales. Como afirman Barbierato y Gatti, 2024, aunque ambas comparten el objetivo de enfrentar retos y cuestiones medioambientales, sus enfoques, implementaciones y resultados difieren. La Green AI se centra en buscar el balance entre rendimiento y eficiencia ya que busca minimizar el coste computacional y el uso de recursos. El enfoque de Green AI es un concepto muy amplio que engloba numerosas estrategias de investigación y desarrollo y que no se limita a ciertas industrias o aplicaciones.

La relación entre la ética medioambiental y el enfoque de Green AI es muy estrecha. El enfoque de Green AI comparte los principios de la ética medioambiental, en especial los de sostenibilidad ecológica y derecho a saber. Este segundo muy relevante, ya que para el enfoque de Green AI es crucial que los investigadores conozcan el coste energético de los distintos modelos para así poder elegir aquellos que sean más sostenibles en relación a su error de predicción. Las buenas propiedades del enfoque de Green AI hacen que este enfoque sea adecuado para evaluar modelos econométricos desde el punto de vista de la ética medioambiental.

V. RETINA

En este apartado se describirá el algoritmo de RETINA y su relación con el enfoque de Green AI.

5.1. *El procedimiento de RETINA*

RETINA (Relevant Transformation of the Inputs Network Approach) es un método que se utiliza para la construcción de modelos. Está diseñado en torno a la flexibilidad (con transformaciones no lineales de los predictores de interés), búsqueda selectiva dentro del rango de posibles modelos, capacidad de pronóstico fuera de muestra y simplicidad computacional.

En Pérez-Amaral et al., 2003⁸ se explica que el algoritmo de RETINA es el siguiente:

Etapa 0 – Preliminar

1. Construcción y ordenación de datos

a) Generar el conjunto de variables transformadas $\zeta(X) = \{W_1, \dots, W_n\}$

b) Dividir la muestra en tres submuestras

Etapa I – Aislamiento de un modelo 'candidato'

2. Utilizando datos de la primera submuestra

Ordenar las variables en $\zeta(X)$ según su correlación muestral (absoluta) con la variable dependiente solo en la primera submuestra de manera que $W_{(1)}$ sea la variable con la mayor correlación absoluta con Y , $W_{(2)}$ sea la segunda más correlacionada, y así sucesivamente.

Considerar varios conjuntos de regresores que incluyan una constante y $W_{(1)}$: cada conjunto de regresores $\zeta_\lambda(X)$ está indexado por un 'umbral de colinealidad' $0 \leq \lambda \leq 1$ y se construye incluyendo $W_{(j)}$ ($j = 2, \dots, m$) en $\zeta_\lambda(X)$ si el R^2 de la regresión de $W_{(j)}$ sobre las variables ya incluidas es $\leq \lambda$.

El número de conjuntos de regresores está controlado por el número de valores de λ entre 0 y 1 elegidos, digamos, v .

3. Utilizando datos tanto de la primera como de la segunda submuestra

Estimar cada modelo regresando Y en cada conjunto de regresores $\zeta_\lambda(X)$ usando los datos de sólo la primera submuestra y computar un criterio de predicción de fuera de la muestra (el error cuadrático medio) utilizando los datos de sólo la segunda submuestra. Esto requiere la estimación de v modelos.

Seleccionar un modelo 'candidato' que se corresponde con el que tiene un mejor rendimiento fuera de la muestra $\zeta_\lambda^*(X)$.

Etapa II – Estrategia de búsqueda

4. Utilizando datos tanto de la segunda como de la tercera submuestra

⁸ PÉREZ-AMARAL, T.; GALLO, G. M., & WHITE, H., "Aflexible tool for model building: the relevant transformation of the inputs network approach (RETINA)", en *Oxford Bulletin of Economics and Statistics*, 65 (2003) 821-838.

Buscar un modelo más parsimonioso: estimar todos los modelos que incluyan una constante y todos los regresores en $\zeta_{\lambda}^*(X)$ uno a la vez en el orden en que se incluyeron originalmente, pero también en el orden producido por el procedimiento 2a, esta vez en base a las correlaciones en la segunda submuestra.

Realizar una evaluación de los modelos fuera de la muestra (utilizando los datos de la tercera submuestra) calculando una medida de rendimiento (el error cuadrático medio, posiblemente aumentado por un término de penalización por el número de parámetros en el modelo).

Etapa III – Selección de modelo

5. Repetir las etapas I y II cambiando el orden de las submuestras; producir un candidato para cada ordenamiento de submuestras.

6. Seleccionar el modelo que tenga el mejor rendimiento sobre toda la muestra.

5.2. RETINA y Green AI

Aunque el enfoque de Green AI es relativamente reciente y muy posterior a RETINA, se podría decir que al algoritmo RETINA comparte la filosofía de Green AI. Esto se debe a que RETINA realiza una búsqueda selectiva y no exhaustiva. Así, para un número m de regresores candidatos, una búsqueda exhaustiva requeriría evaluar 2^m modelos, lo que puede volverse muy costoso computacionalmente si m es relativamente grande. RETINA, por su parte, sólo evalúa una cantidad de modelos del orden de $2m$, una tarea que requiere tanto un tiempo de ejecución mucho menor como menos recursos. De este modo, evaluando un número significativamente menor de modelos, RETINA encuentra un modelo que ofrece un rendimiento similar al que encontraríamos si evaluáramos todos los posibles modelos. La búsqueda selectiva y la Green AI comparten los objetivos de optimizar el uso de recursos y el de mejorar la eficiencia, lo que resulta en un menor consumo energético. Esto hace que la búsqueda selectiva y la Green AI puedan verse como enfoques complementarios en el contexto del desarrollo de IA sostenible.

VI. ECO-RETINA

Marinucci, 2005⁹ programó una interfaz gráfica de usuario de RETINA en Gauss y posteriormente, implementaría una nueva versión en MATLAB. Estos programas contaban con funcionalidades bastante útiles, como la posibilidad de eliminar valores extremos, y extendieron el algoritmo original, incluyendo la posibilidad de aplicarlo con variables ficticias. Estas ampliaciones han sido tenidas en cuentas en el desarrollo del estimador y de la interfaz y se han añadido otras nuevas. Enumeramos todas las nuevas características que posee Eco-RETINA:

Medición del consumo energético y de las emisiones de CO₂ que conlleva entrenar el modelo. Esta funcionalidad es la más interesante y la responsable del nombre de esta versión. Las mediciones se realizan gracias al paquete de Python CodeCarbon, cuya información puede consultarse en Schmidt et al, 2021¹⁰.

La disponibilidad de información sobre el consumo energético del entrenamiento de los diversos modelos facilita la aplicación del enfoque de Green AI.

Eliminación de cuellos de botella en el paso 2b. Si leemos la descripción del paso 2b de RETINA y pensamos en su implementación, es fácil darse cuenta de que este paso puede ser muy costoso computacionalmente. A la hora de programarlo, lo más lógico es seleccionar un parámetro que determine la máxima colinealidad permitida y otro que sirva para formar una rejilla. Cada elemento de la rejilla λ , siguiendo la notación de Pérez-Amaral et al., 2003, λ , se utilizaría para indicar los distintos niveles de colinealidad e indexar cada conjunto de regresores. Esto resulta problemático pues no sólo hay que realizar múltiples regresiones para cada valor de la rejilla, sino que además, si se toma una rejilla muy fina, el proceso tendría que repetirse un número muy elevado de veces. Por ejemplo, con un nivel máximo de multicolinealidad de 0.5 y una rejilla con separación 0.001, habría que repetir el paso para 500 λ distintos.

Además, en general, resulta innecesario realizar el paso un número tan elevado de ocasiones porque después de obtener el primer conjunto de regresores para el primer valor de la rejilla, suele ocurrir que para los siguientes valores de la rejilla se vuelva a obtener el mismo conjunto de regresores y que sólo se añada alguno para valores de λ más altos. Teniendo todo esto en cuenta, se ha

⁹ MARINUCCI, M., *RETINA Winpack for real data: A quick guide for automatic model selection. Report, Universidad Complutense de Madrid*, 2005.

¹⁰ SCHMIDT, V.; GOYAL, K.; JOSHI, A.; FELD, B.; CONELL, L.; LASKARIS, N.; BLANK, D.; WILSON, J.; FRIEDLER, S., & LUCCIONI, S., *CodeCarbon: Estimate and Track Carbon Emissions from Machine Learning Computing*, 2021: <https://github.com/mlco2/codecarbon> <https://doi.org/10.5281/zenodo.4658424>.

cambiado el paso *2b* de la siguiente manera. En primer lugar, para no tener que realizar numerosas regresiones para cada valor de λ , se han utilizado operaciones vectorizadas de la librería Numpy para que baste hacer una regresión para cada valor de la rejilla. En segundo lugar, en vez de hacer una regresión para cada valor de la rejilla, Eco-RETINA realiza el paso sólo para el primer valor de la rejilla original y luego adapta la rejilla y la vuelve menos fina, limitando el número de regresiones a 50. Esto permite que el primer conjunto de regresores se obtenga con una rejilla tan fina como se quiera, pero que no se desperdicie tiempo en hacer cientos de regresiones en las que no se amplía el conjunto de regresores.

Eliminación de los cuellos de botella en el paso *4b*. Si el modelo candidato obtenido en el paso *3b* es muy grande, el paso *4b* pueden consumir mucho tiempo al tener que realizar un gran número de regresiones. Esto se debe a que si el modelo candidato tiene n regresores, en el paso *4b* hay que realizar $2n$ regresiones. Esto es problemático porque en algunas ocasiones el modelo candidato puede tener cientos de regresores. Eco-RETINA soluciona esto limitando a $2k$ el número de regresiones, siendo k el máximo entre 50 y el número de regresores originales. Así, el número de regresiones deja de depender del tamaño del modelo candidato. Eco-RETINA, en vez de considerar todos los submodelos del modelo candidato, sólo considera los 10 submodelos más simples, los 10 submodelos más complejos y un número $k-20$ de submodelos en los que el número de regresores varía uniformemente.

Posibilidad de que algunas transformaciones continuas sólo tengan transformaciones en forma de productos cruzados. Esto es bastante útil ya que, en algunas ocasiones, algunas variables continuas presentan muchos ceros. En estos casos, la transformación cuadrática no resulta especialmente útil y las transformaciones inversas resultan contraproducentes, ya que muchas filas de nuestra tabla de datos pasarían a contener valores nulos y serían ignoradas al hacer una regresión lineal.

Uso de variables ficticias. Esta característica ya fue incluida en la versión de 2004, sin embargo, en el conjunto de variables transformadas se incluían las variables continuas transformadas y sus productos por las variables dummies. Así, en el caso de k variables continuas y d ficticias, el conjunto de variables transformadas constaría de $d \times (2k^2 + 2k + 1)$ regresores. Esta es una cantidad muy grande de variables y, en general, los productos de variables ficticias y transformaciones de variables continuas no son muy usados en econometría por introducir excesiva complejidad en los modelos y por favorecer el sobreajuste. Eco-RETINA corrige esto y sólo considera los productos de variables dummies por las variables continuas originales. De esta manera, con k variables continuas

y d ficticias, el conjunto de regresores estaría formado por $2k^2 + (2 + d)k + d + 1$ variables.

Búsqueda automática del umbral de colinealidad en función del tamaño máximo del modelo final. Al elegir un modelo candidato, Perez-Amaral et al., 2003 mencionan la existencia de un umbral de colinealidad $0 \leq \lambda \leq 1$ que indexa diferentes conjuntos de regresores, pero no explican cómo determinar este umbral. Eco-RETINA aborda este punto proponiendo un umbral en función del número de variables que se permita que tenga el modelo final. El interés de esta característica de Eco-RETINA no sólo yace en que se resuelve una cuestión que quedó abierta en el artículo original, sino en que esta solución permite controlar que el tamaño de los modelos propuestos por RETINA no superen un tamaño determinado. Esto es clave, pues RETINA está pensado para proponer modelos relativamente pequeños.

Eliminación de valores extremos. Esta funcionalidad es muy similar a la que poseía la versión de 2004. No obstante, Eco-RETINA permite seleccionar el multiplicador del rango intercuartílico que se use para eliminar los outliers.

Múltiples transformaciones polinómicas. En RETINA, el conjunto $\zeta(X)$ estaba formado por las variables transformadas W_{ij} , cada una de las cuales se obtenía como $X_{ij}^a \times X_{il}^b$, $a, b = -1, 0, 1$; $j, l = 1, \dots, k^2$. Eco-RETINA va un paso más allá y permite que a y b tomen cualquier valor, incluso valores fraccionales. A pesar de las posibilidades que ofrece esta característica, no se recomienda elegir un número muy elevado de transformaciones. Las ganancias de ampliar las transformaciones son limitadas y sólo añadir un par de transformaciones a las originales supone un incremento notable del coste computacional de Eco-RETINA.

Más métricas de selección de modelos. La versión original de RETINA usa el error cuadrático medio como medida del rendimiento de los modelos. Eco-RETINA añade la posibilidad de usar el criterio de información de Akaike (AIC) y el criterio de información bayesiano (BIC).

Nuevos tipos de regresión. RETINA sólo podía usarse en casos en los que la variable objetivo era continua. Eco-RETINA va más allá, permitiendo usarse problemas de clasificación binaria, pues incorpora la posibilidad de realizar el modelo Logit, el Probit y el Modelo Lineal de Probabilidad (MLP).

Posibilidad de usar covarianza robusta a heterocedasticidad. Se puede a elegir entre covarianza no robusta o robusta del tipo HC0, HC1, HC2 o HC3.

Transformaciones logarítmicas. Eco-RETINA incorpora regresiones de los tipos $Y \sim \log X$, $\log Y \sim X$, $\log Y \sim \log X$.

Paralelización de las 6 iteraciones de RETINA. Eco-RETINA puede ejecutar en paralelo las 6 iteraciones que realiza RETINA, lo que supone una reducción notable de los tiempos de ejecución.

Validación de los resultados. Eco-RETINA ofrece la opción de dividir la muestra en 4 submuestras en lugar de en 3 y usar esta cuarta submuestra como conjunto de validación. Esta opción es muy interesante cuando se quiere evaluar un modelo, pero no se tiene un conjunto de datos de test o cuando se quiere determinar qué hiperparámetros del modelo seleccionar. Por ejemplo, quizás se duda entre si usar las transformaciones estándar de RETINA o de si añadir una transformación cúbica. Pueden evaluarse las dos opciones en el conjunto de validación y elegir la opción con menor error de predicción. De esta manera, al no participar los datos de test en la selección de hiperparámetros, ésta no se ve influenciada ni sesgada.

Estandarización. Eco-RETINA permite estandarizar las variables continuas, lo que reduce el efecto de las diferencias de escala entre las variables.

VII. EL ESTIMADOR Y LA INTERFAZ DE ECO-RETINA

Esta sección se utilizará para explicar cómo usar las herramientas desarrolladas en este trabajo.

Eco-RETINA se ha implementado como un estimador de Python y también como una interfaz de usuario. Aunque ya se han adelantado varios hiperparámetros de Eco-RETINA al hablar de las nuevas características que ofrece, en este apartado se entrará más en detalle y además de describir los hiperparámetros, se mostrará cómo usar el estimador y la interfaz de usuario.

Todos los archivos necesarios para usar Eco-RETINA o para reproducir los experimentos de este trabajo estarán disponibles en Capilla, 2024¹¹ próximamente.

7.1. Requisitos de uso

Tanto el estimador como la interfaz de usuario de Eco-RETINA requieren conexión a internet y un procesador Intel. Además, para poder usar el estimador

¹¹ CAPILLA ROMEROSA J. M., Página de GitHub, 2024: <https://github.com/jcapilla780/Eco-RETINA>

hay que o tener instalados Python y un editor de código, como VSCode o una distribución de Python, como Anaconda.

Para usar el estimador de Python, es necesario haber descargado los archivos `eco_retina.py` y `my_functions_eco.py`. Además, todos los paquetes que contiene el `requirements.txt` deben haberse instalado en el entorno virtual. En cuanto a la interfaz de usuario, para usarla sólo hay que descargar el archivo `eco_retina.exe` que hay en el repositorio de GitHub. Una vez descargado e instalado, ya se puede usar sin necesidad de tener nada más instalado.

7.2. El estimador

Como todo estimador de Python, Eco-RETINA cuenta con los métodos `fit` para entrenar el modelo y `predict` para, una vez entrenado el modelo, usarlo para predecir, generalmente, fuera de la muestra.

El método `fit` tiene como inputs los siguientes valores:

Un dataframe `y`, que contiene la variable objetivo. Si no se da un valor de `y`, aparece un mensaje de error.

Un dataframe `x_c_t`, que contiene las variables continuas que van a tener todas las transformaciones posibles. Tiene valor `None` por defecto.

Un dataframe `x_c_nt` que contiene las variables continuas que sólo van a tener transformaciones en forma de productos cruzados. Tiene valor `None` por defecto.

Un dataframe `x_d` que contiene variables ficticias. Tiene valor `None` por defecto.

No es necesario dar valores a `x_c_t`, `x_c_nt` y `x_d`, pero sí al menos a uno de ellos. Si no, aparece un mensaje de error.

Además, el método `fit` cuenta con los siguientes hiperparámetros:

`seed`. Sirve para fijar la semilla de la división en submuestras de Eco-RETINA. Toma valores enteros positivos. Tiene por defecto el valor 8.

`max_collinearity`. Establece la máxima multicolinealidad permitida por Eco-RETINA. Toma un valor entre 0 y 1 y tiene por defecto el valor 0.5.

`col_params_search` y `max_num_variables`. Estos dos hiperparámetros se utilizan para realizar una búsqueda automática de la máxima colinealidad (menor que `max_collinearity`) y de la distancia entre los valores que forman el mallado de niveles de colinealidad. Mientras que `col_params_search` toma valores booleanos (True o False), dependiendo de si se desea utilizar o no la búsqueda automática, `max_num_variables` toma valores enteros positivos. El valor por defecto de `col_params_search` es True y el de `max_num_variables` es 50.

`collinearity_grid`. A partir de este hiperparámetro se forma el mallado de niveles de multicolinealidad. Toma valores entre 0 y 1. Detallaremos un poco más la función del hiperparámetro con un ejemplo. Si `max_collinearity` es igual a 0.5 y `collinearity_grid` es igual a 0.01, el mallado o la rejilla de niveles de colinealidad serán los valores 0, 0.01, 0.02, 0.03, ..., 0.48, 0.49, 0.5. Los distintos niveles de colinealidad indexan los conjuntos de regresores de la fase 2b de RETINA. El valor por defecto de `collinearity_grid` es 0.005 y debe tenerse en cuenta que, si `col_params_search` es igual a True, Eco-RETINA ignorará el valor asignado a `collinearity_grid` pues seleccionará un valor automáticamente en función del número de variables.

`iqr_multiplier` y `remove`. Estos hiperparámetros se utilizan para eliminar outliers. `iqr_multiplier` establece el multiplicador del rango intercuartílico que se utiliza para la eliminación de valores extremos. Toma un valor positivo y su valor por defecto es 1.5. `remove` se utiliza para eliminar o no los valores extremos (usando el `iqr_multiplier`). Toma un valor booleano, True si se desean eliminar los outliers y False en caso contrario. El valor por defecto de `remove` es False.

`power_list`. Es una lista que contiene los valores de alfa y beta que se utilizan para obtener las variables transformadas, tal y como explican Perez-Amaral et al., 2003. El valor por defecto de `power_list` es [-1, 0, 1]. Con estos valores las variables transformadas incluyen las variables originales, sus cuadrados, sus productos cruzados, sus inversos y sus ratios cruzados. Los valores de alfa y beta que se añadan a `power_list` deben estar ordenados de menor a mayor. Por ejemplo, si se quieren añadir potencias de orden 3 y 4 (incluyendo sus inversas, productos cruzados y demás), `power_list` debería ser una lista igual a [-2, -1, 0, 1, 2].

`metric`. Establece la métrica que se utilize en las fases 3a y 4b para la elección de modelos. Puede tomar los valores 'MSE', 'AIC' o 'BIC'. Tiene por defecto el valor 'MSE'.

`regression_type`. Se refiere al tipo de regresión que se desea realizar. Puede tomar los valores 'Linear' para realizar una regresión lineal, 'Binary (LPM)'

para usar el modelo lineal de probabilidad, 'Binary (logit)' para realizar una regresión logística o 'Binary (probit)' para realizar un modelo probit. Tiene por defecto el valor 'Linear'.

`cov_type`. Se refiere al tipo de covarianza que se usa en la regresión. Puede ser 'nonrobust' si se quiere usar una covarianza que no sea robusta a heterocedasticidad o 'HC0', 'HC1', 'HC2' o 'HC3' si se quiere un tipo de covarianza robusta a heterocedasticidad. El valor por defecto de `cov_type` es 'nonrobust'.

`log_type`. Este hiperparámetro se utiliza para usar transformaciones logarítmicas. Puede tomar los valores 'no' si no se desea aplicar este tipo de transformación, 'x' si se desea aplicar sobre los regresores continuos originales, 'y' si se desea aplicar sobre la variable objetivo o 'x & y' si se desean aplicar transformaciones logarítmicas sobre los regresores continuos originales y sobre la variable objetivo. El valor por defecto de `log_type` es 'no'.

`standardize`. Se utiliza para estandarizar las variables originales continuas. Toma el valor True si se quieren estandarizar y False en caso contrario. El valor por defecto es False.

`multiprocessing`. Este hiperparámetro se usa para ejecutar cada iteración de Eco-RETINA en paralelo o de manera secuencial. La ejecución será en paralelo si `multiprocessing` es igual a True y secuencial si es False. El valor por defecto es True.

`validation` y `random_state`. El hiperparámetro `validation` se utiliza para dividir el conjunto de datos en 4 submuestras en lugar de en 3 y utilizar la cuarta submuestra como conjunto de validación. Toma como valores True, si se quiere obtener un conjunto de validación y False en caso contrario. El valor por defecto es False. Si `validation` es igual a True, se puede acceder a los atributos `score1` y `score2`, que contienen métricas para evaluar el rendimiento del modelo en el conjunto de validación. Si `regression_type` es 'Linear' `score1` devuelve el MSE y `score2` el MAE. Si `regression_type` es 'Binary (LPM)', 'Binary (logit)' logística o 'Binary (probit)', `score1` devuelve el accuracy score y `score2` el `roc_auc` score. El hiperparámetro `random_state` se utiliza para fijar una semilla en la separación del conjunto de validación. Toma valores enteros positivos. El valor por defecto es 42.

`eco`. Se utiliza para usar RETINA o Eco-RETINA. Si `eco` es igual a True, se utilizará Eco-RETINA y si es False, RETINA. El valor por defecto es True. La única diferencia entre los dos procedimientos es que Eco-RETINA no tiene los cuellos de botella que sí tiene RETINA.

7.3. Ejemplo de uso del estimador

Para usar Eco-RETINA, debemos tener los inputs en el formato requerido. Debemos tener un dataframe de pandas con la variable objetivo (en nuestro ejemplo `y_train`), otro con las variables continuas que van a tener todas las transformaciones (en nuestro ejemplo `x_c_t_train`), otro con las variables continuas que sólo van a tener transformaciones en forma de productos cruzados y otro con las variables dummy (en nuestro ejemplo `x_d_train`). En nuestro ejemplo, permitiremos que todas las variables continuas tengan todas las transformaciones posibles, por lo que `x_c_nt` será igual a `None`. Una vez tengamos los inputs, sólo hay que importar la clase `Eco_RETINA` del archivo `eco_retina.py`, crear un objeto de esa clase (en nuestro ejemplo lo hemos llamado `eco_retina`) y usar el método `fit` con los inputs y los hiperparámetros deseados. En la figura 1 se muestra cómo llevar a cabo los pasos mencionados. En ese ejemplo se ha dado al hiperparámetro `max_num_variables` el valor 100, a `max_collinearity`; 0.7 y a `validation`; `True`. El resto de hiperparámetros tiene sus valores por defecto.

Figura 1: Entrenamiento de Eco-RETINA

```
from eco_retina import Eco_RETINA
eco_retina=Eco_RETINA()
eco_retina.fit(y=y_train,x_c_t=x_c_t_train,x_c_nt=None,x_d=d_train, max_num_variables=100, max_collinearity=0.7, validation=True)
✓ 55.8s
```

Una vez entrenado el modelo, podemos ver un resumen del resultado con el método `model.summary`. En la figura 2 se muestra cómo.

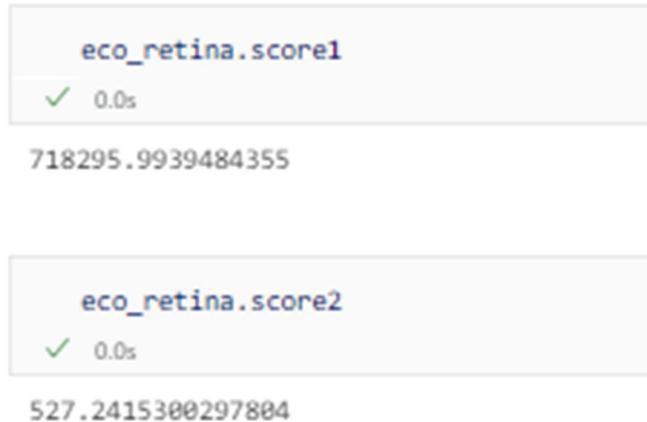
Figura 2: Resumen Modelo de Eco-RETINA

```
eco_retina.model.summary()
✓ 0.0s
```

| OLS Regression Results | | | | | | |
|------------------------|---------------------------------|---------------------|-------------|---------|-------|---------------------|
| Dep. Variable: | obj_variable | R-squared: | 0.839 | | | |
| Model: | OLS | Adj. R-squared: | 0.839 | | | |
| Method: | Least Squares | F-statistic: | 5.631e+04 | | | |
| Date: | Sun, 14 Apr 2024 | Prob (F-statistic): | 0.00 | | | |
| Time: | 18:39:16 | Log-Likelihood: | -7.0650e+05 | | | |
| No. Observations: | 86553 | AIC: | 1.413e+06 | | | |
| Df Residuals: | 86544 | BIC: | 1.413e+06 | | | |
| Df Model: | 8 | | | | | |
| Covariance Type: | nonrobust | | | | | |
| | | coef | std err | t | P> t | [0.025 0.975] |
| | constant | 2406.9761 | 1093.325 | 2.202 | 0.028 | 264.068 4549.884 |
| | carat_weight^1*meas_width^1 | 1012.5979 | 1.530 | 662.004 | 0.000 | 1009.600 1015.596 |
| | meas_width^1*meas_length^1 | 265.2951 | 18.546 | 14.304 | 0.000 | 228.944 301.646 |
| | polish_Excellent*depth_percent | 110.3873 | 25.818 | 4.276 | 0.000 | 59.784 160.991 |
| | polish_Excellent*table_percent | -75.9719 | 27.756 | -2.737 | 0.006 | -130.373 -21.571 |
| | polish_Very Good | 2149.2133 | 603.394 | 3.562 | 0.000 | 966.566 3331.860 |
| | polish_Excellent | 257.7628 | 584.608 | 0.441 | 0.659 | -888.064 1403.590 |
| | polish_Very Good*depth_percent | 91.2128 | 26.122 | 3.492 | 0.000 | 40.013 142.412 |
| | polish_Very Good*table_percent | -91.0710 | 27.773 | -3.279 | 0.001 | -145.506 -36.636 |
| | table_percent^1*depth_percent^1 | -5306.3593 | 1510.614 | -3.513 | 0.000 | -8267.151 -2345.568 |
| Omnibus: | 13326.421 | Durbin-Watson: | 2.006 | | | |
| Prob(Omnibus): | 0.000 | Jarque-Bera (JB): | 166250.786 | | | |
| Skew: | 0.336 | Prob(JB): | 0.00 | | | |
| Kurtosis: | 9.756 | Cond. No. | 2.34e+16 | | | |

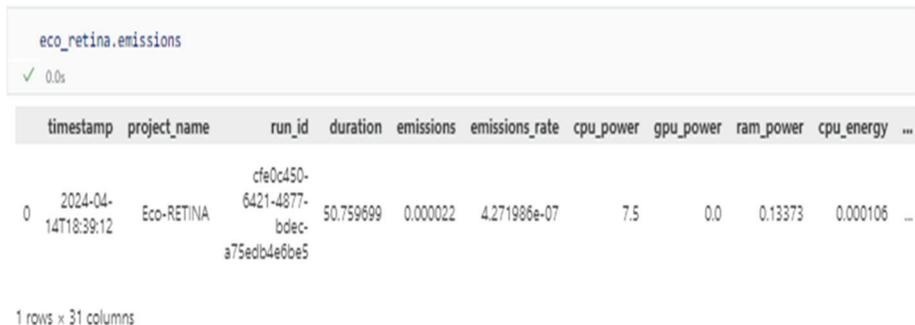
Al haber asignado a validation el valor True, podemos acceder a los atributos score1 y score2. Se muestra en la figura 3.

Figura 3: Score1 y Score2 de Eco-RETINA



Otro atributo importante es el de emissions, que permite ver información sobre las emisiones y el consumo energético del entrenamiento del algoritmo. Se muestra un ejemplo en la figura 4.

Figura 4: Método emissions de Eco-RETINA



Finalmente, si se quiere usar el modelo para predecir, se puede utilizar el método predict. El input de este método es un dataframe `x_c_t`, otro `x_c_nt` y otro `x_d`, los cuales son equivalentes a los del método fit. Se muestra un ejemplo en la figura 6.

Figura 5: Predicciones de Eco-RETINA

```
eco_retina.predict(x_c_t=x_c_t_test,x_c_nt=None, x_d=x_d_test)
✓ 0.2s
```

| | |
|-------|-------------|
| 0 | 820.603389 |
| 1 | 2287.127329 |
| 2 | 6361.782568 |
| 3 | 1007.559120 |
| 4 | 1113.049448 |
| ... | |
| 38464 | 1256.808352 |
| 38465 | 428.632915 |
| 38466 | 920.766638 |
| 38467 | 1113.942725 |
| 38468 | 1758.643490 |

Length: 38469, dtype: float64

7.4. La interfaz de usuario

El uso de la interfaz de usuario es bastante intuitivo. Para comenzar, basta con seleccionar el tipo de archivo en el que se encuentran los datos y encontrarlo con el buscador. Tras la selección de hiperparámetros, el entrenamiento se iniciará cuando se pulse el botón Run Eco-RETINA. Si se quiere ver el consumo energético y las emisiones del entrenamiento hay que pulsar el botón View Emissions y para eliminar el registro de emisiones, sólo hay que pulsar el botón Delete Emissions Record. Si se quiere hacer una predicción puntual, hay que pulsar el botón Point Prediction y para hacer una predicción de un conjunto de datos, hay que pulsar el botón Set Prediction. Las imágenes 7 y 8 muestran cómo luce cada pestaña de la interfaz de Eco-RETINA.

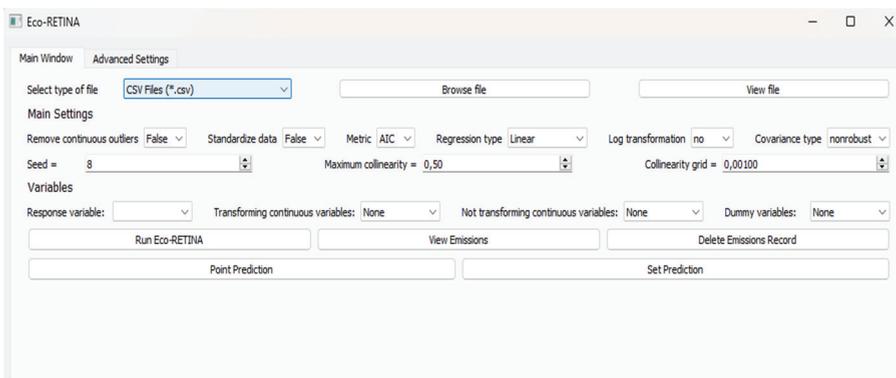
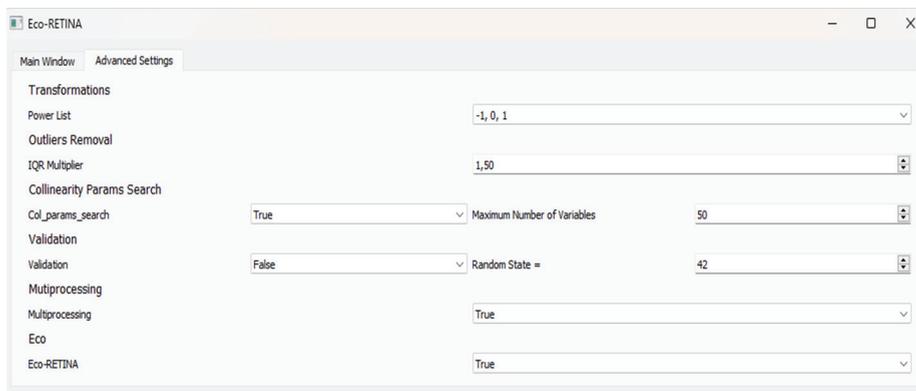
Figura 6: Pestaña Principal de la GUI de Eco-RETINA

Figura 7: Pestaña de Ajustes Avanzados de la GUI de Eco-RETINA

VIII. CASO PRÁCTICO

Los datos usados para los experimentos proceden de hrokr, 2023¹² y serán subidos a Capilla, 2024 junto con el código utilizado.

8.1. Datos

El conjunto de datos usado contiene precios de venta y diferentes características de diamantes. Los datos usados se han obtenido tras limpiar la base original, eliminando outliers y transformando en binarias las variables categóricas. Además de la variable objetivo, el precio de venta de los diamantes, el dataset consta de 50 variables explicativas, de los cuales 6 son continuas y 44 son binarias, que toman el valor 1 si una determinada observación cumple una característica y 0 en caso contrario. Las 6 variables continuas se refieren a aspectos relacionados con el tamaño y el peso de los diamantes. En lo que respecta a las variables binarias, hay 2 relacionadas con el pulido, 2 con la simetría, 11 con la claridad, 10 con el color, 10 con el corte, 6 con la fluorescencia y 3 con el laboratorio que ha evaluado y certificado el diamante

El conjunto de datos utilizado tiene en total 153874 observaciones, divididas en un conjunto de datos de entrenamiento de 115405 observaciones y un conjunto de datos de test de 38469 observaciones.

¹² HROKR, *The largest diamond dataset currently on Kaggle*, 2023:
<https://www.kaggle.com/datasets/hrokrin/thelargest-diamond-dataset-currely-on-kagglekee>.

8.2. Metodología de los experimentos

El objetivo de los experimentos es medir el rendimiento de Eco-RETINA y compararlo con el de RETINA y el de una red neuronal. El rendimiento de los modelos se medirá en términos del error de predicción y de las emisiones de CO₂ necesarias para entrenar los modelos. Se ha elegido el error porcentual medio absoluto como medida del error de predicción de los modelos, ya que las cantidades con las que vamos a trabajar, al tratarse de precios de diamantes, pueden ser bastante elevadas y medidas como el error cuadrático medio pueden resultar poco intuitivas.

Se hará una comparación integral entre los distintos algoritmos. Se medirá cómo se ven afectados por el número de variables, el número de observaciones y el entorno en el que se ejecuten. En el caso de la comparación entre RETINA y Eco-RETINA se omitirá este último aspecto, ya que, lo más interesante de esta comparación yace en ver si existe una gran diferencia entre la capacidad predictiva de ambos algoritmos. Al ser Eco-RETINA una versión simplificada de RETINA, siempre va a tener unos tiempos de ejecución más cortos y su entrenamiento va a causar menos emisiones de CO₂.

La variación en el número de variables se realizará de esta forma. Para los tres algoritmos, primero se han utilizado las 6 variables continuas, las 2 relacionadas con el pulido y las 2 que informaban sobre la simetría. A continuación, a estas 10 se han añadido las 11 variables relacionadas con la simetría y posteriormente; el conjunto de variables se ha ampliado con las 10 relacionadas con el color. Luego, se han añadido los 10 regresores relativos al corte y, finalmente; se han añadido las 6 variables relacionadas con el grado de fluorescencia y las 3 relativas al laboratorio certificador. De esta manera, para cada algoritmo, hemos hecho experimentos con 10, 21, 31, 41 y 50 variables.

Al hablar de los datos utilizados, también se ha comentado que el conjunto de datos de entrenamiento tiene 115405 observaciones. Esto es así para un conjunto de experimentos, pero no para todos. Para ver cómo el tamaño del conjunto de datos de entrenamiento afectaba al rendimiento de los algoritmos, se han realizado los diferentes experimentos entrenando los modelos con el conjunto de datos de entrenamiento original, con una submuestra que contenía el 10% de los datos de entrenamiento y con otra que contenía el 1% de las observaciones.

También se han realizado los experimentos en distintos entornos de ejecución. Para hacer esto, se han repetido los experimentos en algunos de los distintos entornos que ofrece Google Colab. Hemos utilizado el entorno CPU, el V100

(con alta capacidad de RAM) y el A100. El entorno CPU tiene 2 procesadores lógicos, una CPU Intel(R) Xeon(R) CPU @ 2.20GHz y una RAM de 12.674789 GB. El entorno V100 tiene 8 procesadores lógicos, una CPU Intel(R) Xeon(R) CPU @ 2.00GHz, una memoria RAM de 50.99369 GB y una GPU Tesla V100-SXM2-16GB. Finalmente, el entorno A100 consta de 12 procesadores lógicos, una CPU Intel(R) Xeon(R) CPU @ 2.20GHz, una memoria RAM de 83.477112 GB y una GPU NVIDIA A100-SXM4-40GB. Los 3 entornos tienen un sistema operativo Linux-6.1.58+-x86_64-with-glibc2.35. Cabe decir que, debido al funcionamiento de Google Colab, no podemos comparar directamente las emisiones entre los distintos entornos de ejecución. Esto se debe a que Google Colab asigna al usuario una máquina aleatoriamente y, generalmente, estas máquinas suelen estar en distintos puntos geográficos. En el caso de nuestros experimentos, el entorno CPU estaba en Carolina del Sur, el V100 en Groningen y el A100 en Iowa. Para poder concluir si entrenar un modelo tiene mayor consumo energético que hacerlo en otro, ambos deberían encontrarse en la misma ubicación, por lo que nuestros resultados son intraentorno y no interentorno.

8.3. Hiperparámetros y arquitectura

El objetivo de los experimentos no es encontrar el modelo con el menor error de predicción. No se realizará selección de hiperparámetros, ya que, lo que se busca es hacerse una idea de la capacidad predictiva de Eco-RETINA y de cómo de verde es. Para ello, estudiamos su MAPE y las emisiones de CO₂ de su entrenamiento y las comparamos con las de RETINA y con las de una red neuronal.

En lo que respecta a RETINA y a Eco-RETINA, ambos algoritmos usarán los mismos hiperparámetros, los cuales serán los más básicos. No se añadirán transformaciones polinómicas, ni se usarán transformaciones logarítmicas. Tampoco se estandarizarán los datos ni se eliminarán valores extremos. Tampoco se restringirá el tamaño del modelo. Se permitirá la máxima colinealidad posible, 1, y se tomará una rejilla con separación de 0.005, que es la Eco-RETINA tiene por defecto.

En cuanto a la red neuronal, programada con la librería TensorFlow2, se usará un perceptrón multicapa con una arquitectura estándar. Aplicamos los principios generales de Ranjan¹³ para diseñarla. Tendrá dos capas ocultas y cada capa tiene un número de nodos de una serie geométrica de 2. La primera

¹³ RANJAN, C., *Understanding deep learning: Application in rare event prediction. Imbalanced Learning; Connaissance*. Publishing: Atlanta, GA, USA (2020) 19-27.

capa tendrá un número de nodos cercano a la mitad del número de regresores y la siguiente capa, la mitad de nodos que la primera. Así, para 10 variables, la primera capa tendrá 4 nodos y la segunda, 2. En el caso de 21 variables, la primera capa será de 8 nodos y la segunda de 4. En los experimentos con 31 y 41 variables, la primera capa tendrá 16 nodos y la segunda, 8. Finalmente, para 50 regresores, la primera capa tendrá 32 nodos y la segunda, 16. En las dos capas ocultas se utilizará la función de activación ReLU. En todos los experimentos el número de lotes será igual a 16, se utilizará el error cuadrático medio como función de pérdida y se usará el optimizador Adam con una tasa de aprendizaje de 0.001. Los datos serán escalados con un escalado MinMax antes de entrenar la red neuronal con ellos.

Aunque ya se ha especificado la arquitectura que se va a seguir en cada experimento, aún falta determinar un hiperparámetro fundamental para hacer los experimentos con el perceptrón multicapa: el número de épocas de cada experimento. Este parámetro es crucial, pues está directamente relacionado con el tiempo de entrenamiento de la red neuronal y, por tanto, también con el consumo energético de dicho entrenamiento. Si se elige un número bajo de épocas, el consumo energético será más bajo, pero es más probable que la capacidad predictiva de la red no sea la óptima. Por el contrario, si se entrena la red durante un número más elevado de épocas, siempre que no se caiga en el sobreajuste; la probabilidad de converger hacia una solución óptima será mayor. No obstante, también lo será el consumo energético del entrenamiento. Una estrategia que se suele usar para evitar entrenar una red neuronal durante un número de épocas excesivo consiste en utilizar un conjunto de datos de validación y, si la función de pérdida en el conjunto de validación no disminuye durante un número determinado de épocas, conocido como paciencia, se detiene el entrenamiento. El único inconveniente que presenta este método es que, al tener que usar un conjunto de validación, la red neuronal se entrena con menos datos.

Para entrenar a nuestra red neuronal durante un número adecuado de épocas sin tener que reducir el conjunto de entrenamiento, se procederá de la siguiente manera. Primero, se separará el conjunto de datos de entrenamiento en un conjunto de datos de entrenamiento y otro de validación y, se entrenará la red neuronal durante un cierto número de iteraciones. Se detendrá el entrenamiento cuando no haya mejora en la métrica de evaluación elegida sobre el conjunto de validación para una paciencia concreta. A continuación, usando el número de iteraciones obtenido, se entrenará la red neuronal nuevamente, pero esta vez sobre todo el conjunto de entrenamiento. Sólo se medirán las emisiones de este último entrenamiento. Esta manera de proceder combina las ventajas de todas las alternativas mencionadas y también supera todos sus inconvenientes. Lo único

que faltaría por determinar es qué paciencia elegir. Normalmente se elige un número entre 1 y 10, aunque a veces se toman valores mayores. Para evitar tomar un valor demasiado grande o pequeño, usaremos 5 y 10 como paciencia. Si observamos una mejora en el error cuadrático medio de las predicciones en el conjunto de validación al pasar de 5 a 10, tomaremos el número de épocas que determine tener una paciencia de 10 épocas y, si el error cuadrático medio es el mismo, elegiremos el número de épocas que se obtenga con una paciencia de 5 épocas. Como es lógico, para poder proceder de esta manera hay que fijar la semilla global del entorno para que los pesos de la red se inicialicen de la misma manera

En las siguientes tablas se observa el número de épocas y el valor de la función de pérdida en el conjunto de datos de validación para distinto número de variables y tamaño del conjunto de datos de entrenamiento. Las filas que están en negrita son las que se van a usar en los experimentos.

Tabla 1: Número de Épocas 100% Datos de Entrenamiento

| Número de Variables | Paciencia | Pérdida de Validación | Número de Épocas |
|---------------------|-----------|-----------------------|------------------|
| 10 | 5 | 0.00572904 | 34 |
| 10 | 10 | 0.00572904 | 39 |
| 21 | 5 | 0.004340689 | 23 |
| 21 | 10 | 0.00434069 | 28 |
| 31 | 5 | 0.001081986 | 41 |
| 31 | 10 | 0.001080723 | 53 |
| 41 | 5 | 0.000984115 | 23 |
| 41 | 10 | 0.00095931 | 51 |
| 50 | 5 | 0.000659944 | 31 |
| 50 | 10 | 0.000659944 | 36 |

Tabla 2: Número de Épocas 10% Datos de Entrenamiento

| Número de Variables | Paciencia | Pérdida de Validación | Número de Épocas |
|---------------------|-----------|-----------------------|------------------|
| 10 | 5 | 0.005682512 | 24 |
| 10 | 10 | 0.005453233 | 70 |
| 21 | 5 | 0.004315536 | 31 |
| 21 | 10 | 0.004282544 | 56 |
| 31 | 5 | 0.001637506 | 30 |
| 31 | 10 | 0.001536177 | 72 |
| 41 | 5 | 0.001450007 | 20 |
| 41 | 10 | 0.001450007 | 25 |
| 50 | 5 | 0.001265879 | 21 |
| 50 | 10 | 0.001265879 | 26 |

Tabla 3: Número de Épocas 1% Datos de Entrenamiento

| Número de Variables | Paciencia | Pérdida de Validación | Número de Épocas |
|---------------------|-----------|-----------------------|------------------|
| 10 | 5 | 0.007167303 | 39 |
| 10 | 10 | 0.007122321 | 58 |
| 21 | 5 | 0.005078375 | 36 |
| 21 | 10 | 0.005078375 | 41 |
| 31 | 5 | 0.003554217 | 29 |
| 31 | 10 | 0.003524646 | 44 |
| 41 | 5 | 0.002637084 | 37 |
| 41 | 10 | 0.002637084 | 42 |
| 50 | 5 | 0.004101982 | 21 |
| 50 | 10 | 0.004099518 | 32 |

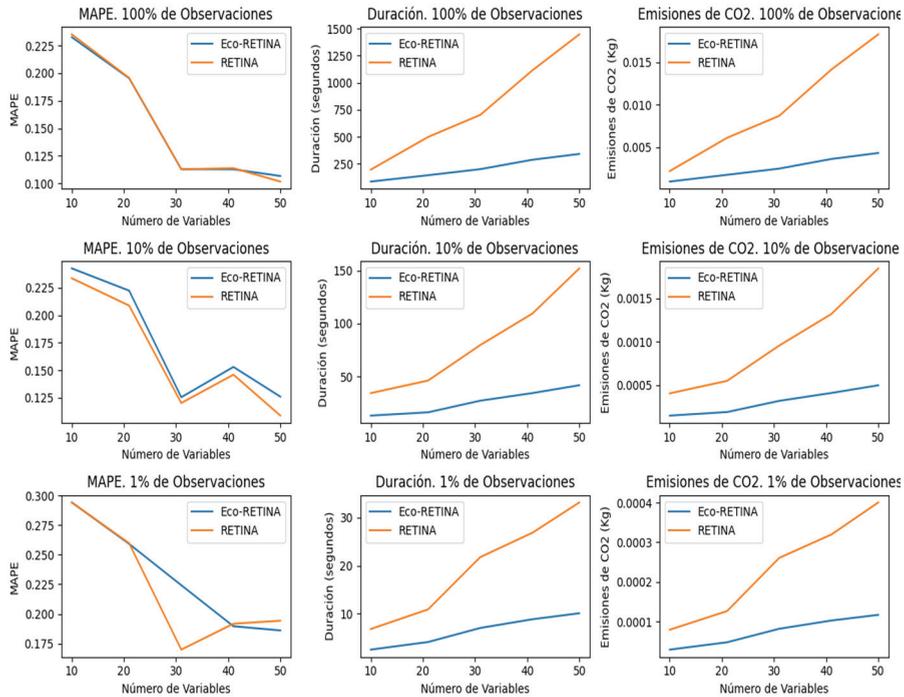
Obsérvese que tanto los datos como la metodología que se seguirá benefician a la red neuronal en comparación a Eco-RETINA. RETINA es un algoritmo optimizado para datos económicos, en los que suele haber pocas variables, sin embargo, en nuestros experimentos, vamos a usar datos sobre precios de diamantes y hasta 50 variables. A esto hay que añadir que, aunque no hemos usado un conjunto de validación para encontrar los parámetros óptimos de Eco-RETINA, en el caso de la red neuronal, para encontrar el número adecuado de épocas, hacemos dos entrenamientos: uno usando una paciencia de 5 y otra de 10. Es cierto que no se tunean hiperparámetros que son clave en la arquitectura de una red neuronal, como el número de capas o de nodos, pero para cada experimento hemos elegido arquitecturas estándar y que suelen funcionar bien en la mayoría de los casos.

8.4. Comparación de Resultados de Eco-RETINA y RETINA

En la figura 8 se realiza una comparación del rendimiento de Eco-RETINA con el de RETINA. Ambos algoritmos se han ejecutado en el entorno A100 de Google Colab y con los mismos hiperparámetros. La única diferencia entre los dos es que Eco-RETINA no posee los cuellos de botella que hemos señalado en la sección 4.

En la primera columna se observa la comparación de los dos modelos en términos de MAPE, en la segunda; del tiempo de ejecución y en la tercera; de las emisiones del entrenamiento. En la primera fila se usa el 100% de los datos de entrenamiento, en la segunda fila; el 10% y en la tercera fila se utilizan sólo el 1% de observaciones del conjunto de datos de entrenamiento.

Figura 8: Comparación entre Eco-RETINA y RETINA



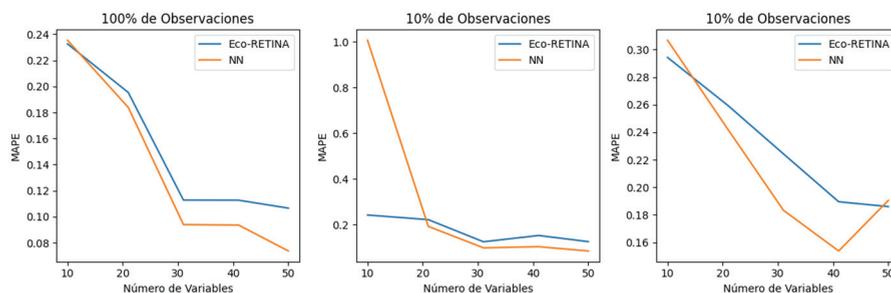
En lo que respecta al error de predicción, para el 100% de observaciones en el conjunto de entrenamiento, Eco-RETINA y RETINA tienen un MAPE similar. Con 10 variables, Eco-RETINA tiene un MAPE de 0.2326, con 21; de 0.1954, con 31; de 0.1128; con 41; de 0.1127 y, con 50; de 0.1066. Por su parte, RETINA presenta unos MAPEs de 0.2352, 0.1956, 0.1128, 0.1137 y 0.1016 puntos para 10, 21, 31, 41 y 50 variables respectivamente. Sorprende el hecho de que Eco-RETINA tenga un MAPE inferior para 10, 21 y 41 variables, lo que puede indicar algo de sobreajuste por parte de RETINA. Para el 10% de observaciones en los datos de entrenamiento, sí se observa un menor error de predicción de RETINA en todos los experimentos. No obstante, la mayor diferencia entre los errores medios porcentuales de Eco-RETINA y RETINA es de 0.017, lo que tampoco supone una gran diferencia. En el caso de la submuestra con el 1% de observaciones en el conjunto de entrenamiento Eco-RETINA logra unos MAPEs de 0.2943, 0.2592, 0.2242, 0.1895 y 0.186 puntos, mientras que los de RETINA son de 0.2943, 0.2597, 0.1698, 0.1917 y 0.1942 puntos. Así, Eco-RETINA logra un menor MAPE en el caso de 21, 41 y 51 variables, RETINA logra un menor MAPE con 31 variables y los dos modelos tienen el mismo MAPE para 10 variables.

En cuanto al tiempo de ejecución y a las emisiones de CO₂ del entrenamiento, como se aprecia en 8, están directamente relacionados. Eco-RETINA tiene unos tiempos de ejecución más cortos y su entrenamiento tiene unas emisiones de CO₂ menores. Se observa que ambas magnitudes, a medida que aumenta el número de variable, crecen a una tasa más alta para RETINA que para Eco-RETINA. En efecto, el caso del 100% de los datos de entrenamiento, para 10 variables RETINA tiene un tiempo de ejecución de 195.41 segundos y Eco-RETINA de 85.32 segundos, es decir, 2.29 veces menos. Sin embargo, para 50 variables el entrenamiento de RETINA tiene una duración de 1146.21 segundos, mientras que el de Eco-RETINA es de 340.64 segundos, el cual es aproximadamente 4.25 veces menor. Este resultado refleja las ventajas de Eco-RETINA y se debe a que, a medida que aumenta el número de variables, cada regresión del paso 2b se vuelve más costosa de realizar computacionalmente y, al aumentar el tamaño del modelo candidato, aumenta el número de regresiones que hay que hacer en el paso 4b. También se puede observar la reducción en el tiempo de ejecución y en las emisiones de CO₂ que implica reducir el número de observaciones. En los experimentos con 50 variables, con el 100% de observaciones en el dataset de entrenamiento, el entrenamiento de Eco-RETINA tiene una duración de 340.64 segundos y unas emisiones de CO₂ de 0.004319 kg. Con el 10% de observaciones, la duración es de 41.69 segundos y las emisiones de 0.000500 kg. Finalmente, para el 1% de observaciones en el conjunto de datos de entrenamiento, Eco-RETINA tiene un tiempo de ejecución de 10.12 segundos y unas emisiones de CO₂ de 0.000117 kg.

La comparación entre Eco-RETINA y RETINA refleja que, aunque, en general, RETINA goza de un menor error de predicción, la diferencia es muy baja. Por su parte, el entrenamiento de Eco-RETINA tiene unos tiempos de ejecución y unas emisiones menores y su diferencia con RETINA se agranda a medida que aumenta el número de variables.

8.5. Comparación de resultados de Eco-RETINA y una red neuronal

En la figura 9 se realiza una comparación entre el MAPE de Eco-RETINA con el de una red neuronal con la arquitectura descrita en el apartado 5.3. En la primera columna se tiene el caso en el que ambos modelos se han entrenado con el 100% de observaciones de los datos de entrenamiento, en la segunda columna; con el 10% y, en la tercera; con el 1%.

Figura 9: Comparación entre Eco-RETINA y una NN. MAPE

Los MAPEs de Eco-RETINA se han mencionado en la sección anterior. En cuanto a los de la red neuronal, para el 100% de observaciones en los datos de entrenamiento, los MAPEs son de 0.2354, 0.1839, 0.0940, 0.0937 y 0.0738 puntos para 10, 21, 31, 41 y 50 variables, respectivamente. Con el 10% de observaciones en el dataset de entrenamiento, la red neuronal tiene unos MAPEs de 1.0066, 0.1928, 0.099, 0.1039 y 0.0852. Finalmente, con el 1% de observaciones en el conjunto de entrenamiento, los MAPEs de la red neuronal son 0.3067, 0.2417, 0.1833, 0.1538 y 0.1905.

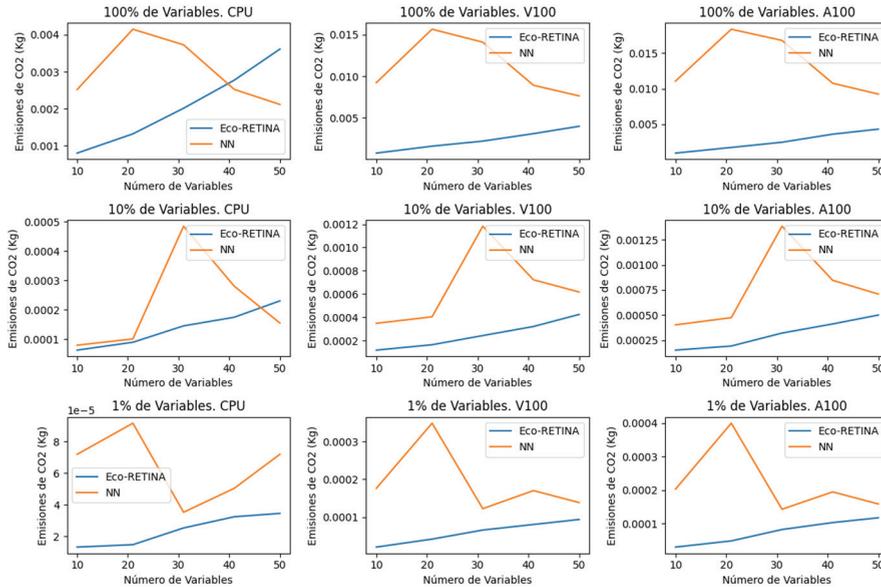
El resultado que más sorprende inicialmente es el MAPE de la red neuronal en el experimento con 10 variables y la submuestra del 10% de los datos de entrenamiento, que es igual a 1.0066. Aunque un MAPE tan elevado sea un caso puntual y probablemente se solucione cambiando la arquitectura de la red neuronal, refleja una mayor sensibilidad a los hiperparámetros por parte de la red neuronal que de Eco-RETINA, que tiene un MAPE moderado en todos los experimentos.

Otro resultado que se ha repetido a lo largo de los experimentos y que quizás puede pasar desapercibido es que Eco-RETINA ha logrado un menor error de predicción que la red neuronal para 10 variables con todos los conjuntos de entrenamiento. Además de ser un resultado interesante, es coherente con el procedimiento de RETINA y con el tipo de datos para el que está pensado. Eco-RETINA es un procedimiento orientado a la previsión fuera de la muestra de datos económicos, por lo que es natural que destaque con datos parecidos a los de naturaleza económica, en los que no es habitual que haya un elevado número de variables.

En la imagen 10 se muestran las emisiones de Eco-RETINA y de la red neuronal en los distintos experimentos. En la primera columna se muestran los experimentos realizados en el entorno CPU de Google Colab, en la segunda; en el de V100 y en la tercera; en el de A100. En cuanto a las filas, en la primera están

los experimentos en los que se ha usado el conjunto de datos de entrenamiento con el 100% de observaciones, en la segunda; el que tiene el 10% de observaciones y, en la tercera; el que sólo consta del 1%.

Figura 10: Comparación entre Eco-RETINA y una NN. Emisiones de CO2



Se observa que, mientras que para Eco-RETINA las emisiones aumentan a medida que se incrementa el número de variables, no sucede lo mismo para la red neuronal. Esto se debe a que, dado un entorno de ejecución, las emisiones están directamente relacionadas con el tiempo de ejecución, que está vinculado al número de épocas con el que se entrena la red. Como vemos en las tablas 1, 2 y 3, un mayor número de variables no siempre se corresponde con un mayor número de épocas.

En el entorno de la CPU, para el 100% de observaciones en el conjunto de entrenamiento, Eco-RETINA tiene unas emisiones de CO2 de 0.0008 kg, 0.0013 kg, 0.002 kg, 0.0028 kg y 0.0036 kg, mientras que las de la red neuronal son de 0.0025 kg, 0.0042 kg, 0.0037 kg, 0.0025 kg y 0.0021 kg. Con el dataset de entrenamiento con el 10% de observaciones, las emisiones de CO2 de Eco-RETINA son de 6.33×10^{-5} kg, 9.01×10^{-5} kg, 1.46×10^{-4} kg, 1.75×10^{-4} kg y 2.31×10^{-4} kg. En cambio, las de la red neuronal son de 8×10^{-5} kg, 1.01×10^{-4} kg, 4.84×10^{-4} kg, 2.8×10^{-4} kg y 1.55×10^{-4} kg. En cuanto a la submuestra con el 1% de observaciones, mientras que Eco-RETINA muestra unas emisiones de 1.33×10^{-5} kg, 1.48×10^{-5} kg, 2.54×10^{-5} kg, 3.24×10^{-5}

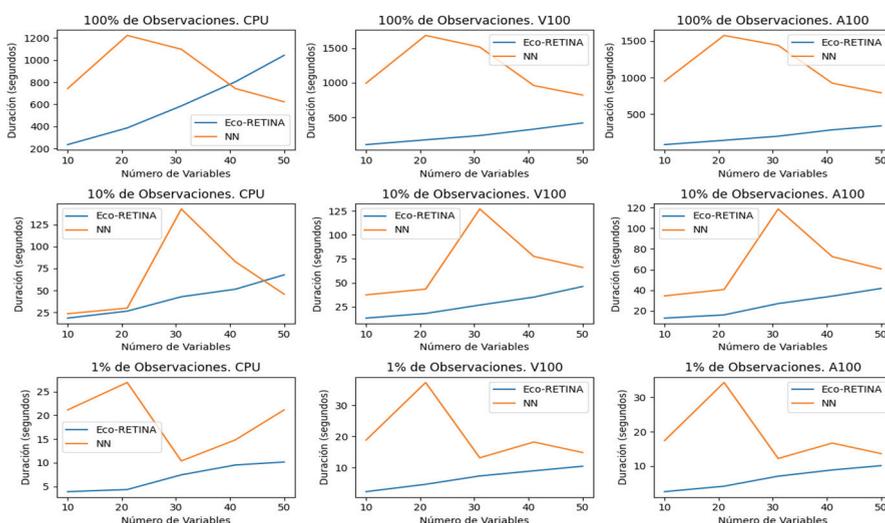
kg y 3.46×10^{-5} kg, las de la red neuronal son de 7.18×10^{-5} kg, 9.15×10^{-5} kg, 3.53×10^{-5} kg, 5.04×10^{-5} kg y 7.19×10^{-5} kg.

En los entornos de V100 y A100, el entrenamiento de Eco-RETINA logra menores emisiones que el de la red neuronal para cualquier tamaño del conjunto de entrenamiento y cualquier número de variables. En el caso del 100% de los datos en el conjunto de entrenamiento, sorprende el caso 10 y 21 variables pues para los dos entornos de ejecución las emisiones de Eco-RETINA son de unas 10 veces menos que las de la red neuronal.

Las figuras 8 y 10 reflejan que las emisiones de CO2 son significativas cuando el entrenamiento de los algoritmos se realiza con menos datos. Dado que la diferencia entre los MAPEs de los modelos entrenados con el 100%, el 10% y el 1% de observaciones es moderada, merece la pena cuestionarse si cuando se tiene una gran cantidad de datos se puede utilizar sólo una submuestra, tal como se sugiere en Varian, 2014¹⁴.

La figura 11, que se muestra a continuación, muestra los tiempos de ejecución de los distintos experimentos. Los gráficos están distribuidos de la misma manera que en la figura 10 y lo único que cambia es que en el eje de ordenadas de los gráficos aparece la duración del entrenamiento en lugar de las emisiones de CO2.

**Figura 11: Comparación entre Eco-RETINA y una NN.
Duración del Entrenamiento**



¹⁴ VARIAN, H. R., “Big data: New tricks for econometrics”, en *Journal of economic perspectives*, 28(2) (2024) 3-28.

Las figuras 10 y 11 parecen reflejar una relación directa entre tiempo de entrenamiento y emisiones de CO₂, ya que ambas curvas tienen la misma forma. Así, los resultados muestran que, dado un entorno de ejecución, mayor tiempo de entrenamiento implica mayor cantidad de emisiones. Además, también se aprecia que la relación tiempo-emisiones se ve influenciada por el entorno de ejecución, ya que, para un mismo experimento hay entornos en los que, pese a que el tiempo de entrenamiento de un experimento es menor, las emisiones son mayores. El estudio de la influencia de los entornos de ejecución en las emisiones queda fuera del alcance de este trabajo, pero sería interesante estudiar cuánta de esa influencia se debe al entorno de ejecución, cuánta se debe a la localización geográfica de la máquina y cuánta proviene de otros factores, como por ejemplo la hora o el momento del año en el que se realiza el experimento.

Si analizamos los resultados en su conjunto, comparando el MAPE y las emisiones medias de Eco-RETINA y de la red neuronal, tenemos que Eco-RETINA tiene de media un MAPE un 14,23% menor al de la red neuronal y unas emisiones que son 73,06% menores a las de la red. No obstante, el menor MAPE se debe a que la red neuronal tuvo un MAPE superior a 1 en el experimento con 10 variables y un entrenamiento con el 10% de observaciones en el dataset de entrenamiento. Si excluimos ese experimento del análisis, el MAPE medio de los experimentos con Eco-RETINA es un 13,51% superior al de la red neuronal y sus emisiones son un 73,13% inferior. Este resultado refleja que, aunque Eco-RETINA tiene, de media, un mayor error de predicción que una red neuronal, también tiene un consumo energético considerablemente menor.

IX. CONCLUSIONES

En este trabajo se ha estudiado la relación entre la ética medioambiental y la econometría aplicada. Para ello, se ha implementado y utilizado el algoritmo Eco-RETINA, desarrollado a partir de RETINA, siguiendo un enfoque de Green AI, el cual comparte los principios de la ética medioambiental. Se ha comprobado que Eco-RETINA logra un error de predicción similar al de RETINA con un consumo energético mucho menor. También se ha comparado Eco-RETINA con una red neuronal y se ha obtenido que, si bien el error de predicción de la red neuronal es algo menor, las emisiones del entrenamiento de Eco-RETINA son considerablemente menores. Los experimentos también reflejan que Eco-RETINA tiene un mejor rendimiento que la red neuronal cuando se usa con pocas variables. Si a todo esto añadimos que Eco-RETINA es un procedimiento completamente transparente y cuyos resultados pueden ser interpretables, Eco-RETINA parece cumplir todas las condiciones para convertirse en un estimador a tener en cuenta en el marco del Green AI. Sería interesante realizar nuevos

experimentos y comparaciones utilizando otro tipo de datos, entre ellos económicos para ver el desempeño de Eco-RETINA y el de otros modelos.

Los resultados de este trabajo muestran que es posible realizar modelado econométrico buscando no sólo maximizar el poder predictivo, sino también la eficiencia. Se anima al resto de investigadores a seguir las prácticas sugeridas por el enfoque de Green AI y a realizar un seguimiento de las emisiones de sus modelos y algoritmos. También se invita a usar Eco-RETINA en futuras investigaciones, una vez esté subida la versión definitiva. Para cerrar, también hay que añadir que la implementación de Eco-RETINA está todavía sujeta a mejoras. El desarrollo de nuevas librerías en Python que puedan ayudar a acelerar los tiempos de ejecución o a reducir las emisiones, pueden hacer que Eco-RETINA sufra modificaciones en el futuro. Por supuesto, cualquier sugerencia o código que tenga el propósito de ayudar a mejorar Eco-RETINA es bienvenido.

X. BIBLIOGRAFÍA

- BARBIERATO, E., & GATTI, A., *Towards Green AI. A methodological survey of the scientific literature*, IEEE Access, 2024.
- BRENNAN, A., & LO, Y., *Environmental Ethics Stanford Encyclopedia of Philosophy*, ed. en Zalta, 2016.
- CASTLE, J. L., & HENDRY, D. F., “Econometrics for modelling climate change”, en *Oxford Research Encyclopedia of Economics and Finance*, 2022.
- COCHRANE, A., “Environmental ethics”, en *Internet encyclopedia of philosophy*, 2006.
- CAPILLA ROMEROSA J. M., *Página de GitHub*, 2024: <https://github.com/jcapilla780/Eco-RETINA>
- GEEKSFORGEEEKS, *Environmental Ethics: Types, Importance, Examples*, 2023: <https://www.geeksforgeeks.org/environmental-ethics/>
- HROKR, *The largest diamond dataset currently on Kaggle*, 2023: <https://www.kaggle.com/datasets/hrokrin/thelargest-diamond-dataset-currely-on-kaggle>
- KEEBLE, B. R., “The Brundtland report: «Our common future»”, en *Medicine and war*, 4(1) (21998) 17-25.

- MARINUCCI, M., *RETINA Winpack for real data: A quick guide for automatic model selection. Report*, Universidad Complutense de Madrid, 2005.
- PÉREZ-AMARAL, T.; GALLO, G. M., & WHITE, H., “A flexible tool for model building: the relevant transformation of the inputs network approach (RETINA)“, en *Oxford Bulletin of Economics and Statistics*, 65 (2003) 821-838.
- RANJAN, C., *Understanding deep learning: Application in rare event prediction. Imbalanced Learning; Connaissance*. Publishing: Atlanta, GA, USA (2020) 19-27.
- SCHMIDT, V.; GOYAL, K.; JOSHI, A.; FELD, B.; CONELL, L.; LASKARIS, N.; BLANK, D.; WILSON, J.; FRIEDLER, S., & LUCCIONI, S., *CodeCarbon: Estimate and Track Carbon Emissions from Machine Learning Computing*, 2021: <https://github.com/mlco2/codecarbon>
<https://doi.org/10.5281/zenodo.4658424>
- SCHWARTZ, R.; DODGE, J.; SMITH, N. A., & ETZIONI, O., “Green ai”. *Communications of the ACM*, 63(12) (2020) 54-63.
- VARIAN, H. R., “Big data: New tricks for econometrics“, en *Journal of economic perspectives*, 28(2) (2014) 3-28.