



Universidad Politécnica
de Madrid



**Escuela Técnica Superior de
Ingenieros Informáticos**

Grado en Ciencia de Datos e Inteligencia Artificial

Trabajo Fin de Grado

**EVALUACIÓN de RENDIMIENTO
PREDICTIVO de MÉTODOS
TRANSPARENTES**

Autor: Pablo Martín Escobar

Tutor: Bojan Mihaljevic

Madrid, junio 2024

Este Trabajo Fin de Grado se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

Trabajo Fin de Grado

Grado en Ciencia de Datos e Inteligencia Artificial

Título: EVALUACIÓN de RENDIMIENTO PREDICTIVO de MÉTODOS
TRANSPARENTES

Junio 2024

Autor: Pablo Martín Escobar

Tutor:

Bojan Mihaljevic
Departamento de Inteligencia Artificial
ETSI Informáticos
Universidad Politécnica de Madrid

Resumen

El presente Trabajo de Fin de Grado se enfoca en realizar una comparación del rendimiento predictivo entre métodos de machine learning transparentes y métodos de caja negra. Además, se integrará el clasificador Naive Bayes en la librería `interpretml` obteniendo una interpretabilidad local y global de este modelo, publicando el código y la documentación.

Este proyecto tiene como objetivo principal contribuir al avance del conocimiento en el campo del machine learning mediante una investigación sobre el rendimiento predictivo de diferentes enfoques algorítmicos. Se dará especial énfasis a la importancia del código open-source al interactuar con la librería `interpretml`, donde se explorará la integración del clasificador Naive Bayes de `scikit-learn`. Este proceso de integración no solo busca ampliar las funcionalidades de la librería, sino también mejorar el entendimiento de su desempeño al analizar técnicas de clasificación.

Se ha optado por seleccionar conjuntos de datos ideales para realizar un benchmarking completo, asegurando que cumplan con criterios de calidad, extensión y diversidad. A la hora de elegir los datos se eligen bases de datos en los que la explicabilidad sea importante, pero también otras que puedan abarcar cualquier tipo de contexto.

En el transcurso de este Trabajo de Fin de Grado, se ofrecerá un detallado análisis sobre los conceptos esenciales de los métodos de machine learning, centrándose particularmente en los métodos transparentes y métodos de caja negra que serán implementadas. Se describirán minuciosamente las metodologías de evaluación utilizadas para valorar la eficacia de los modelos desarrollados.

Como parte final del proyecto, se analizarán los resultados obtenidos y se extraerán conclusiones en base a ellos. El objetivo de esta fase es recopilar nueva información útil a través del análisis de los resultados. Estas conclusiones se utilizarán para mejorar nuestra comprensión en esta área de investigación y mejorar las herramientas ya existentes para abordar el problema de la interpretabilidad en la Inteligencia Artificial.

Palabras clave: Machine Learning, modelos transparentes, modelos de caja negra, clasificador Naive Bayes, `interpretml`, `scikit-learn`, código open-source, interpretabilidad local y global, rendimiento predictivo, benchmarking, conjuntos de datos, evaluación de modelos y explicabilidad.

Abstract

This thesis focuses on a comprehensive comparison of the predictive performance between glassbox machine learning models and blackbox models. In addition, the Naive Bayes classifier will be integrated into the interpretml library obtaining a local and global interpretability of this model, publishing the code and documentation.

The main objective of this project is to contribute to the advancement of knowledge in the field of machine learning by means of exhaustive research on the predictive performance of different algorithmic approaches. Special emphasis will be given to the importance of open-source code when interacting with the interpretml library, where the integration of scikit-learn's Naive Bayes classifier will be explored. This integration process not only seeks to extend the functionalities of the library, but also to improve the understanding of its performance when analysing classification techniques.

We have chosen to select ideal datasets for full benchmarking, ensuring that they meet criteria of quality, breadth, and diversity. In the choice of data, databases are chosen where explainability is important, but also others that can cover any kind of context.

In the course of this thesis, a detailed analysis of the essential concepts of machine learning methods will be provided, focusing particularly on the Glassbox models and Blackbox models that will be implemented. The evaluation methodologies used to assess the effectiveness of the developed models will be described in detail.

As a final part of the project, the results obtained will be analysed and conclusions will be drawn based on them. The aim of this phase is to gather new useful information through the analysis of the results. These conclusions will be used to improve our understanding of the research area and to improve existing tools to address the problem of interpretability in Artificial Intelligence.

Keywords: Machine Learning, glassbox models, blackbox models, Naive Bayes classifier, interpretml, scikit-learn, open-source code, local and global interpretability, predictive performance, benchmarking, datasets, model evaluation and explainability.

Tabla de contenidos

1	Introducción.....	1
1.1	Motivación y necesidad del proyecto	1
1.2	Objetivos	4
1.3	Planificación.....	5
2	Alcance del proyecto.....	7
3	Fundamentos.....	9
3.1	Introducción la explicabilidad de la Inteligencia Artificial	9
3.2	Introducción del aprendizaje automático	9
3.1.2	Modelos transparentes y caja negra a aplicar	11
4	Desarrollo	19
4.1	Conjuntos de datos	19
4.1.1	OpenML: Una Plataforma Colaborativa para la Ciencia de Datos	19
4.1.2	Selección y Carga de datos.....	19
4.2	Metodología para la aplicación de los algoritmos	21
4.2.1	Diseño del Benchmark	21
4.2.2	Evaluación de Modelos y Métricas	21
4.2.3	Benchmarking de modelos	22
4.3	Integración de interpretml	23
4.3.1	Ejemplo de uso	27
5	Resultados y conclusiones	30
5.1	Resultados del Benchmark.....	30
5.2	Análisis de los modelos	35
5.3	Conclusiones finales.....	36
5.4	Trabajo futuro	37
6	Análisis de impacto.....	39
7	Referencias.....	40
8	Anexos.....	42

1 Introducción

En la actualidad, la Inteligencia Artificial está en el foco de todos los medios y la opinión popular [1]. Esta disciplina que durante muchos años ha pasado desapercibida, ahora está siendo examinada minuciosamente. Esto se debe a su gran potencial, y a la aparición de novedosas herramientas como por ejemplo ChatGPT de OpenAI, no obstante, esta disciplina va más allá y ofrece una gran gama de posibilidades como son el reconocimiento de imágenes, el análisis de sentimientos, automatización de procesos y muchos más casos de uso.

Para hablar sobre este tema primero hay que preguntarse: ¿Qué es la Inteligencia Artificial?

La Inteligencia Artificial es un campo de la informática que se centra en la transmisión de la inteligencia y el pensamiento antropomórficos a máquinas de modo que puedan ayudar a los humanos, de muchas maneras. El término Inteligencia Artificial fue creado por John McCarthy en 1956. La Inteligencia Artificial ha surgido lentamente y se ha fortalecido en muchos campos como la ingeniería, las matemáticas, la física y la tecnología, todo lo cual ha conducido al tremendo cambio actual en este campo del que ahora somos testigos [2]. Se trata de la idea que las máquinas pueden adquirir inteligencia. Abarca ámbitos como que las máquinas puedan aprender por sí mismas, adaptarse a una circunstancia concreta y autocorregir sus propios errores. Es decir, que las máquinas puedan pensar por sí mismas sin que se les codifiquen órdenes [3].

Una vez que ya se puede estar ubicado en este tema, surgen más dudas como, por ejemplo: ¿Qué es el aprendizaje automático? ¿Qué son los métodos transparentes y que relación tienen en esta área de estudio?

El aprendizaje automático es una rama de la Inteligencia Artificial cuyo objetivo general es permitir que los ordenadores "aprendan" sin ser programados directamente. Tiene su origen en el movimiento de Inteligencia Artificial de los años 50 y hace hincapié en objetivos y aplicaciones prácticas, en particular la predicción y la optimización. Los ordenadores "aprenden", mejorando su rendimiento en las tareas a través de la "experiencia". En la práctica, "experiencia" suele significar ajuste a los datos; de ahí que no exista una frontera clara entre el aprendizaje automático y los enfoques estadísticos. De hecho, el hecho de que una metodología determinada se considere "aprendizaje automático" o "estadística" suele reflejar su historia tanto como diferencias genuinas, y muchos algoritmos (por ejemplo, el operador de selección y reducción mínima absoluta) se han redescubierto y reutilizado en ambos campos [4].

Es crucial comprender que, a medida que la Inteligencia Artificial se integra cada vez más en nuestra vida cotidiana, surge una necesidad imperiosa de abordar cuestiones éticas y de transparencia. La transparencia en el desarrollo y aplicación de la Inteligencia Artificial se ha convertido en un tema candente en la agenda de investigadores, legisladores y la sociedad en general. Y esto es clave para entender que, dentro de la Inteligencia Artificial y el aprendizaje automático, se buscan métodos transparentes e Inteligencia Artificial Explicable (XAI), por sus siglas en inglés.

Para abordar el desafío de equilibrar la precisión predictiva con la capacidad de explicación en los modelos de aprendizaje automático, se han desarrollado recientemente técnicas de interpretabilidad significativas. Estas técnicas, conocidas como XAI, aproximan los modelos complejos de aprendizaje

automático, a menudo considerados "cajas negras", mediante modelos más simples e interpretables. Esto permite inspeccionar y comprender los mecanismos internos del modelo, logrando así una mayor transparencia. El concepto de XAI está ganando una gran atención, ya que ofrece un enfoque prometedor para alcanzar tanto altos niveles de precisión en las predicciones como una mayor interpretabilidad de los modelos. [5].

Si bien las técnicas XAI que aproximan modelos de caja negra con modelos más interpretables y simples son una prometedora vía de investigación, en este trabajo optaremos por un enfoque diferente. Nuestro objetivo es realizar una comparación exhaustiva entre modelos transparentes y modelos de caja negra, evaluando su rendimiento en diversos escenarios y determinando que tipo de modelo tienen mejor rendimiento.

Los métodos transparentes son aquellos modelos y algoritmos cuyo funcionamiento interno es fácil de entender y explicar. Se contraponen a los modelos de caja negra (como redes neuronales profundas o modelos de boosting), donde es difícil o imposible comprender cómo se llega a una predicción específica.

Consideramos que esta comparación directa nos permitirá obtener una visión más clara de las ventajas y desventajas inherentes a cada tipo de modelo, sin depender de aproximaciones o simplificaciones. Al analizar el rendimiento de ambos tipos de modelos en diferentes contextos, podremos identificar patrones y establecer criterios para seleccionar el modelo más apropiado según las características específicas de cada problema.

La creciente preocupación por la transparencia en la Inteligencia Artificial ha llevado a organismos como la Unión Europea a exigir una mayor explicabilidad de estos sistemas. Esto implica no solo poder entender cómo funcionan, sino también que los usuarios sean conscientes de cuándo interactúan con ellos y quiénes son sus responsables. Además, se busca que la Inteligencia Artificial sea capaz de detectar y advertir sobre posibles manipulaciones, como los "deepfakes", fomentando así un entorno digital más seguro y confiable. [6].

El futuro del aprendizaje automático interpretable se basa en tres premisas fundamentales: la digitalización de la información, la automatización de tareas y la dificultad para especificar objetivos de manera perfecta.

1. **Digitalización:** La creciente digitalización de la información en todas las áreas de la vida y la industria impulsará el desarrollo y la aplicación del aprendizaje automático.
2. **Automatización:** La automatización de tareas mediante el aprendizaje automático continuará en aumento, ya que ofrece ventajas económicas y de eficiencia.
3. **Especificación imperfecta de objetivos:** La incapacidad de definir objetivos en el aprendizaje automático, plantea desafíos que pueden ser abordados en parte por los métodos de interpretación.

Estas premisas sugieren que el aprendizaje automático se adoptará cada vez más en diversas industrias e instituciones, impulsado por la necesidad de transparencia y explicabilidad. Los métodos de interpretación, al cerrar la brecha entre los objetivos especificados y los reales, serán cruciales para facilitar esta adopción. Además, se espera que el aprendizaje automático se automatice aún más, incluyendo la interpretación de modelos. Aunque la interpretación final seguirá requiriendo la intervención humana, se prevé que

las herramientas de aprendizaje automático serán más accesibles y fáciles de usar, permitiendo que incluso personas sin conocimientos técnicos puedan entrenar modelos [7].

En la siguiente imagen muestra el aumento de la demanda de una Inteligencia Artificial explicable:

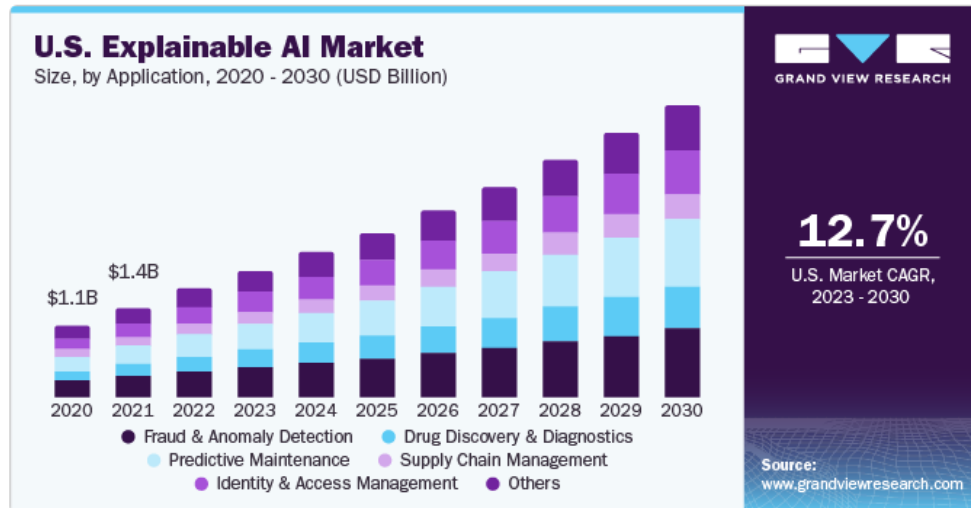


Figura 1. En esta gráfica se muestra una estimación del aumento de la demanda de una Inteligencia Artificial explicable. Figura extraída de [8].

En este contexto, este trabajo sobre el rendimiento predictivo de métodos transparentes es de vital importancia. La transparencia es indispensable a la hora de entender cómo funciona un algoritmo o sistema de IA.

Uno de los modelos interpretables que se estudiarán es el clasificador Naive Bayes. Este modelo utiliza el teorema de Bayes para calcular la probabilidad de que una instancia pertenezca a una clase determinada, basándose en las probabilidades de sus características individuales. El clasificador Naive Bayes asume que las características son independientes entre sí, lo que simplifica el cálculo y la interpretación del modelo. Aunque esta suposición de independencia puede ser una simplificación excesiva en algunos casos, el clasificador Naive Bayes sigue siendo un modelo interpretable popular debido a su sencillez y eficacia en muchas aplicaciones. Es un método transparente con unas características muy interesantes como robustez, baja latencia y rapidez, muchas veces pasado por alto [9].

Para profundizar en este tema se va a analizar una librería de `interpretml`, un paquete Python de código abierto que pone a disposición de profesionales e investigadores algoritmos de interpretabilidad del aprendizaje automático. `Interpretml` expone dos tipos de interpretabilidad - transparente, que son modelos de aprendizaje automático diseñados para la interpretabilidad (por ejemplo: modelos lineales, listas de reglas, modelos aditivos generalizados), y técnicas de explicabilidad de modelos caja negra, como las funciones de explicabilidad local y general [10].

1.1 Motivación y necesidad del proyecto

La creciente presencia de la Inteligencia Artificial en la sociedad plantea desafíos éticos y prácticos que demandan soluciones innovadoras. En este contexto, se

presenta un proyecto centrado en el rendimiento predictivo de los métodos transparentes en IA, un tema crucial para comprender el funcionamiento de estos sistemas y abordar cuestiones de transparencia y posibles sesgos.

La Inteligencia Artificial ofrece un enorme potencial para mejorar nuestra vida, pero su creciente complejidad y la opacidad de algunos algoritmos generan inquietud. En este contexto, la transparencia se convierte en un elemento clave para fomentar la confianza pública y mitigar riesgos. El proyecto aborda esta necesidad urgente, investigando métodos transparentes que permitan comprender y explicar las decisiones tomadas por los modelos de IA.

Hay diferentes investigaciones en las que ya se habla de que es crucial establecer una infraestructura de "ciencia abierta" para compartir datos experimentales y algoritmos. Es en este ámbito donde este Trabajo de Fin de Grado puede aportar a la comunidad investigadora, creando un terreno común para los investigadores que trabajan en la explicación de cajas negras de distintos ámbitos. También es crucial desarrollar plataformas participativas especializadas que permitan la participación de una multitud de usuarios para comprobar la comprensibilidad y utilidad de las explicaciones proporcionadas para la decisión que han tomado, apoyando así campañas de validación de las soluciones técnicas propuestas, y proporcionar datos sobre sus resultados [11].

El objetivo es sentar las bases para futuros trabajos en este campo, estableciendo criterios claros para diferenciar entre modelos transparentes y caja negra, y proporcionando una evaluación rigurosa de su desempeño en diferentes escenarios. Al hacerlo, se contribuirá al debate sobre la transparencia en la IA, un tema de creciente relevancia en nuestra sociedad.

1.2 Objetivos

Para alcanzar los objetivos planteados en el proyecto sobre el rendimiento predictivo de métodos transparentes en el contexto de la Inteligencia Artificial, se deben llevar a cabo y cumplimentar las siguientes metas clave:

- **Revisión bibliográfica:** Realizar un estudio de los métodos pertinentes a aplicar en el proyecto. Esta revisión bibliográfica proporcionará una base sólida de conocimiento para fundamentar las decisiones metodológicas y analíticas.
- **Integración de naive Bayes con interpretml:** Desarrollar funcionalidades de interpretabilidad global y local específicamente diseñadas para el algoritmo de naive Bayes. Esto implica la adaptación y configuración de la librería interpretml para su integración con este método de predicción. Se buscará que sea capaz de alterar y comprender el código repositado.
- **Diseño del experimento:** Proceder a la preparación de los datos y la configuración del entorno de programación necesario para la ejecución del experimento. Esta etapa incluirá la selección adecuada de conjuntos de datos de prueba y la configuración de los parámetros experimentales.
- **Evaluación de los métodos:** Llevar a cabo la ejecución del experimento diseñado y recolectar los datos resultantes. Durante esta fase, se pondrán a prueba los métodos transparentes de predicción, incluyendo el naive Bayes integrado con interpretml, para evaluar su rendimiento y eficacia en diferentes escenarios.
- **Análisis comparativo de los resultados:** Organizar, evaluar y discutir los resultados obtenidos en el experimento. Comparar el rendimiento de

los métodos transparentes, como los modelos de "caja negra". Esta etapa será fundamental para identificar fortalezas, limitaciones y posibles áreas de mejora de los métodos transparentes de predicción.

- **Publicación del código:** Documentar y publicar todo el código desarrollado en el transcurso del proyecto en una plataforma colaborativa como Github. Esto facilitará la reproducibilidad de los resultados y permitirá que otros investigadores y profesionales puedan utilizar y mejorar el código en proyectos futuros

Como ya se ha comentado anteriormente, al alcanzar estos objetivos supone un avance significativo en el campo de la IA, proporcionando información sobre la eficacia y utilidad de los métodos transparentes en la predicción de datos y estableciendo las bases para futuras investigaciones en este campo en constante evolución.

1.3 Planificación

El proyecto se desarrolla en tres fases principales: investigación, desarrollo y análisis.

Fase de Investigación:

1. **Exploración de Métodos Transparentes y Caja Negra:** Se realiza una investigación exhaustiva sobre los métodos transparentes de predicción en IA, con énfasis en los modelos de aprendizaje automático disponibles en `interpretml`. Se analizan tanto modelos transparentes (Explainable Boosting Machine, Linear Model, árbol de decisión) como de caja negra (Boosting, Random Forest), comparando sus características y rendimiento.
2. **Estudio de Herramientas de Interpretabilidad:** Se lleva a cabo una revisión detallada de las herramientas de interpretabilidad de modelos de IA, centrándose en la librería `interpretml`. Se exploran sus funcionalidades y capacidades para comprender y explicar las predicciones de los modelos de aprendizaje automático.

Fase de Desarrollo:

1. **Selección y Preparación de Datos:** Se seleccionan conjuntos de datos relevantes para la experimentación, representando una variedad de escenarios y problemáticas en IA. Se realiza un preprocesamiento de los datos para asegurar su calidad y adecuación para el análisis posterior.
2. **Implementación de Algoritmos Transparentes y caja negra en un benchmark:** Se implementan los algoritmos de predicción transparentes y caja negra seleccionados en la fase de investigación, configurando los modelos y preparando los datos para el análisis.
3. **Integración de Naive Bayes en `interpretml`:** Se trabaja en la implementación de funcionalidades de interpretabilidad global y local para el algoritmo Naive Bayes en `interpretml`. En esta etapa, se busca comprender y adaptar el código.

Fase de Análisis:

1. **Evaluación del Rendimiento de los Modelos:** Se evalúa el rendimiento de los modelos entrenados utilizando la métrica de precisión del área bajo la curva ROC. Se compara el rendimiento de los métodos transparentes con modelos de caja negra.
2. **Análisis de Resultados y Conclusiones:** Se analizan los resultados obtenidos, identificando patrones, tendencias y posibles sesgos. Se utilizan métricas adecuadas para comparar el rendimiento general de los modelos transparentes y de caja negra. Se discuten las implicaciones de los hallazgos y se ofrecen recomendaciones para futuras investigaciones en Inteligencia Artificial transparente. También se analizará el resultado obtenido de la integración de Naive Byaes en la librería y se expondrá que nuevas funcionalidades y ventajas se han podido desplegar.

2 Alcance del proyecto

Este proyecto trasciende la investigación y desarrollo de métodos transparentes de predicción en Inteligencia Artificial . Su enfoque se centra en la ética y el bienestar social, buscando una Inteligencia Artificial comprensible, confiable y beneficiosa para todos.

El impacto ético y social radica en la promoción de una Inteligencia Artificial transparente. Al desarrollar métodos de predicción comprensibles, se fomenta la confianza pública y se empodera a los usuarios para entender y cuestionar las decisiones automatizadas.

El proyecto se desarrolla dentro de un marco legal y ético que incluye normativas como el Reglamento General de Protección de Datos (RGPD), garantizando el cumplimiento de estándares internacionales de transparencia, seguridad y confidencialidad de los datos. Esto asegura una implementación ética y responsable de las técnicas de Inteligencia Artificial desarrolladas, promoviendo un uso justo y equitativo.

Esta aproximación se alinea con la creciente preocupación sobre los riesgos de una implementación inmoral de la IA. El proyecto busca contribuir a un uso transparente de esta tecnología, en línea con la toma de conciencia de gobiernos y ciudadanos sobre sus implicaciones sociales.

Además, el proyecto impulsará la investigación ética en IA, abriendo nuevas discusiones sobre los desafíos asociados con su uso. Se espera inspirar prácticas similares en otros investigadores, fomentando una cultura de ética y responsabilidad en el desarrollo de IA.

Un objetivo clave es comprender por qué diferentes modelos de Inteligencia Artificial tienen un rendimiento variable en distintos contextos. Queremos identificar cuándo y por qué utilizar cada método, considerando la interpretabilidad como un factor crucial. Para ello, analizaremos datos de diversos campos, especialmente aquellos donde la explicabilidad es vital, como el sector sanitario.

El estudio se centrará en la clasificación de datos, dado que los conjuntos de datos empleados asignan una etiqueta o categoría a cada instancia de entrada. El objetivo será predecir esta clase en lugar de un valor numérico continuo, como se haría en una regresión

En la siguiente imagen se muestra en que campos hay más demanda de una Inteligencia Artificial explicable

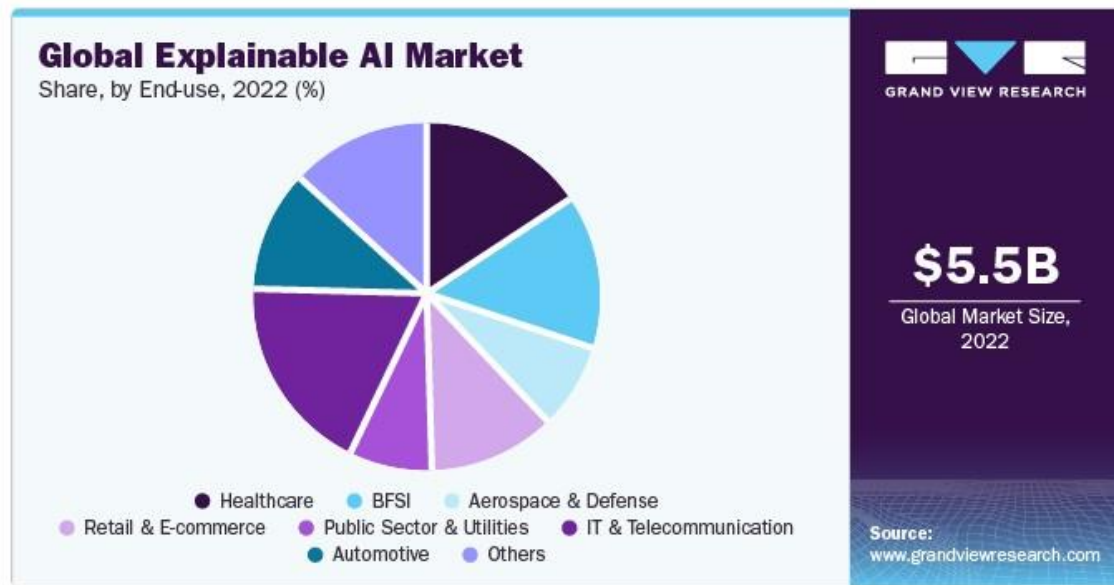


Figura 2. En esta gráfica se muestra en que campos se estima que hay más demanda de una Inteligencia Artificial explicable. Figura extraída de [8].

Podemos concluir que nuestro proyecto tiene un alcance amplio, centrado tanto en la investigación y desarrollo de métodos transparentes de predicción en IA. Uno de los objetivos principales es analizar el rendimiento de los modelos transparentes contra los de caja negra con diferentes bases de datos, calculando para cuales son mejores o el rendimiento. Buscamos identificar cuándo y por qué usar cada método de predicción, destacando la importancia de la explicabilidad.

3 Fundamentos

3.1 Introducción la explicabilidad de la Inteligencia Artificial

Se va a profundizar en los siguientes conceptos clave en el campo de la Inteligencia Artificial Explicable:

- **Explicabilidad:** La explicabilidad se refiere a la capacidad de un modelo de aprendizaje automático para proporcionar detalles o razones que hagan que su funcionamiento sea claro y comprensible para una audiencia específica. No se trata solo de que el modelo sea preciso, sino también de que los humanos puedan entender cómo y por qué llega a sus decisiones.
- **Modelos de caja negra y transparentes:**
 - **Modelos de caja negra:** Son aquellos modelos de aprendizaje automático cuyo funcionamiento interno es complejo y difícil de interpretar directamente. Ejemplos de estos modelos son las redes neuronales profundas, los bosques aleatorios y las máquinas de vectores de soporte.
 - **Modelos transparentes:** Son modelos que son comprensibles por sí mismos, sin necesidad de técnicas externas. Su estructura y parámetros permiten una interpretación directa de cómo llegan a sus predicciones. Ejemplos de estos modelos son la regresión lineal, los árboles de decisión y los modelos basados en reglas.
- **Explicabilidad global y local:**
 - **Explicabilidad global:** Busca comprender el comportamiento general del modelo y cómo utiliza las características de entrada para realizar predicciones. Se enfoca en entender qué características son más importantes en general y cómo influyen en las predicciones del modelo en todo el conjunto de datos.
 - **Explicabilidad local:** Se centra en explicar predicciones individuales. Busca entender por qué el modelo tomó una decisión específica para un caso particular, detallando cómo cada característica contribuyó a esa predicción específica.

Como se comentó anteriormente, la XAI es fundamental en el desarrollo y uso responsable de la IA. Permite generar confianza en los modelos al comprender cómo funcionan, facilita la identificación y corrección de errores o sesgos, garantiza el cumplimiento de regulaciones en áreas críticas y promueve la transparencia en el uso de la Inteligencia Artificial en la sociedad [12].

3.2 Introducción del aprendizaje automático

A lo largo de la historia, la humanidad ha empleado una variedad de herramientas para simplificar sus labores. La imaginación del cerebro humano ha dado lugar a la creación de diversas máquinas que han hecho la vida más fácil al permitir cumplir distintas necesidades esenciales, como el transporte, la producción industrial, el procesamiento de información, etc. Entre estas invenciones se encuentra el aprendizaje automático. Ya hemos tocado el tema del aprendizaje automático en la introducción, pero ahora se profundizará en su estudio.

Según Arthur Samuel, el aprendizaje automático se define como el campo de estudio que confiere a los ordenadores la capacidad de aprender sin ser programados explícitamente. Según Arthur Samuel, el aprendizaje automático se basa en distintos algoritmos para resolver problemas de datos. A los científicos de datos les gusta señalar que no hay un único tipo de algoritmo que sea el mejor para resolver un problema. El tipo de algoritmo empleado depende del tipo de problema que se quiera resolver, el número de variables, el tipo de modelo que mejor se adapte, etc.

El aprendizaje automático abarca diversos enfoques, cada uno con sus propias características y aplicaciones:

- El **aprendizaje supervisado** es la tarea de aprendizaje automático que consiste en aprender una función que asigna una entrada a una salida basándose en pares ejemplos pares de entrada-salida. El conjunto de datos de entrada se divide en conjunto de datos de entrenamiento y de prueba. El conjunto de datos de entrenamiento tiene una variable de salida que debe predecirse o clasificarse.
En el aprendizaje automático supervisado, existen dos enfoques principales:
 - Clasificación: El objetivo es asignar una etiqueta o categoría a cada instancia de entrada. Por ejemplo, clasificar correos electrónicos como spam o no spam, o imágenes de animales según su especie.
 - Regresión: El objetivo es predecir un valor numérico para cada instancia de entrada. Por ejemplo, predecir el precio de una vivienda en función de sus características o la demanda de un producto en función de su precio.
- En el **aprendizaje no supervisado**, a diferencia del aprendizaje supervisado, no hay respuestas correctas. Los algoritmos se dejan a su aire para descubrir y presentar la estructura interesante de los datos. Los algoritmos de aprendizaje no supervisado aprenden de características de los datos sin clasificar. Se utiliza principalmente para la agrupación y la reducción de características.
- El **aprendizaje automático semisupervisado** es una combinación de métodos de aprendizaje automático supervisado y no supervisado. Puede ser fructífero en aquellas áreas del aprendizaje automático y la minería de datos en las que los datos no etiquetados ya están presentes y obtener los datos etiquetados es un proceso tedioso. Con los métodos de aprendizaje automático supervisado más comunes, se entrena un algoritmo de aprendizaje automático en un conjunto de datos "etiquetados" en el que cada registro incluye la información del resultado.
- El **aprendizaje por refuerzo** es un área del aprendizaje automático que se ocupa de cómo los agentes de software deben realizar acciones en un entorno para maximizar alguna noción de recompensa acumulativa. El aprendizaje por refuerzo es uno de los tres paradigmas básicos del aprendizaje automático, junto con el aprendizaje supervisado y el aprendizaje no supervisado.
- Los **modelos generativos** son aquellos que pueden generar datos nuevos similares a los datos en los que fueron entrenados. Estos modelos aprenden la distribución de probabilidad conjunta de las características y la variable objetivo, lo que se denota como $P(x, y)$. $P(x, y)$ representa la probabilidad conjunta de que ocurra una combinación específica de características (x) y la variable objetivo (y). Al

modelar esta distribución de probabilidad, los modelos generativos pueden generar nuevos puntos de datos que son consistentes con los patrones y relaciones aprendidos de los datos de entrenamiento.

En otras palabras, $P(x, y)$ captura la probabilidad de observar un valor particular de y dado un valor particular de x , y viceversa. Por lo tanto, cualquier algoritmo que modele $P(x, y)$ se considera generativo, ya que puede usarse para generar nuevos puntos de datos que siguen la misma distribución que los datos originales.

- **Aprendizaje profundo** es una serie de algoritmos que se esfuerzan por reconocer las relaciones subyacentes en un conjunto de datos mediante un proceso que imita el funcionamiento del cerebro humano. En este sentido, las redes neuronales se refieren a sistemas de neuronas, ya sean de naturaleza orgánica o artificial. Las redes neuronales pueden adaptarse a entradas cambiantes, de modo que la red genera el mejor resultado posible sin necesidad de rediseñar los criterios de salida. El concepto de redes neuronales, que tiene sus raíces en la Inteligencia Artificial, está ganando popularidad rápidamente en el desarrollo de sistemas de negociación [13].

En este proyecto uno de los principales objetivos es analizar el rendimiento de modelos de predicción asociado al aprendizaje supervisado en lugar del aprendizaje no supervisado. El aprendizaje supervisado ofrece un objetivo claro y controlado al predecir o clasificar salidas basadas en entradas etiquetadas, lo que permite una evaluación precisa del modelo. Además, los modelos supervisados tienen una aplicabilidad directa en numerosas áreas, lo que hace que entender su funcionamiento y precisión sea esencial.

Además, la interpretación de los resultados en el aprendizaje supervisado es más fácil debido a la disponibilidad de salidas esperadas durante el entrenamiento y la evaluación. Por lo tanto, en muchos proyectos, se prioriza el análisis del rendimiento de los modelos supervisados debido a su claridad, precisión y aplicabilidad en situaciones del mundo real.

Es importante destacar que el rendimiento de un modelo de aprendizaje supervisado puede variar según diversos factores, como el tipo de algoritmo utilizado, la calidad y cantidad de datos de entrenamiento disponibles, la complejidad del problema y la adecuación del modelo al tipo de datos y al objetivo de la predicción. En este proyecto, exploraremos diferentes algoritmos y técnicas para evaluar y comparar su desempeño en la resolución de problemas específicos.

3.2.1 Modelos transparentes y caja negra a aplicar

El próximo apartado es crucial, ya que proporcionará una base sólida para entender y comparar los diferentes enfoques de predicción en Inteligencia Artificial. Al introducir estos métodos, se contextualizará cómo se relacionan con el objetivo general del proyecto: analizar el rendimiento de métodos transparentes de predicción.

Habiendo definido previamente los modelos transparentes y caja negra, se puede sintetizar diciendo que los primeros destacan por su transparencia y facilidad de interpretación, siendo ideales para aplicaciones donde la explicabilidad es crucial. Por otro lado, los modelos caja negra, aunque pueden ser muy precisos, carecen de transparencia en su proceso de toma de decisiones, siendo a menudo más complejos y utilizando técnicas avanzadas de IA.

Ahora, después de esta introducción, se procederá a formalizar los modelos que buscamos analizar en este proyecto y analizar su rendimiento. Por un lado, están los modelos transparentes: Explainable Boosting Machine, Naive Bayes, árbol de decisión y la regresión logística, y por el otro están los modelos caja negra que son el Random Forest y el XGBoost [13].

Explainable Boosting Machine

Como parte del framework de interpretml, también se incluye un nuevo algoritmo de interpretabilidad: el Explainable Boosting Machine (EBM). EBM es un modelo transparente, diseñado para tener una precisión comparable a los métodos de aprendizaje automático más avanzados, como Random Forest y Boosted Boosted Trees, a la vez que resulta muy inteligible y explicable. El EBM es un modelo aditivo generalizado (GAM) de la forma

$$g(E[y]) = B_0 + \sum_{i=1}^n f_i(x_i)$$

donde g es la función de enlace que adapta el GAM a diferentes escenarios como la regresión o la clasificación. El EBM presenta algunas mejoras importantes con respecto a los GAM tradicionales (Hastie y Tibshirani, 1987). En primer lugar, EBM aprende cada función de característica f_j utilizando técnicas modernas de aprendizaje automático como bagging y gradient boosting. El procedimiento de refuerzo está cuidadosamente restringido para entrenar una característica a la vez en modo round-robin utilizando una tasa de aprendizaje muy baja para que el orden de las características no importe. El procedimiento de round-robin recorre las características para mitigar los efectos de la colinealidad y aprender la mejor f_j para cada característica con el fin de mostrar cómo contribuye cada característica a la predicción del modelo para el problema. En segundo lugar, EBM puede detectar e incluir automáticamente términos de interacción por pares de la forma:

$$g(E[y]) = B_0 + \sum_{i=1}^j f_i(x_i) + \sum_{i=1}^i \sum_{j=1}^j f_{i,j}(x_i, x_j)$$

lo que aumenta aún más la precisión manteniendo la inteligibilidad. EBM es una implementación rápida del algoritmo GA2M (Lou et al., 2013), escrita en C++ y Python. La implementación es paralelizable y aprovecha joblib para proporcionar paralelización multinúcleo y multimáquina. Los detalles algorítmicos del procedimiento de entrenamiento, la selección de términos de interacción por pares y los estudios de casos se pueden encontrar en (Lou et al., 2012, 2013; Caruana et al., 2015).

Los EBM son muy inteligibles, porque la contribución de cada característica a una predicción final puede visualizarse y entenderse trazando f_j . Dado que el EBM es un modelo aditivo, cada característica contribuye a las predicciones de una forma modular que facilita el razonamiento sobre la contribución de cada característica a la predicción.

Ejemplo de Interpretabilidad (f):

Supongamos que tenemos un EBM entrenado para predecir el riesgo de enfermedad cardíaca. Una de las características del modelo es la edad del paciente. Al graficar la función de característica para la edad (f_{edad}), podemos observar cómo el riesgo predicho de enfermedad cardíaca cambia con la edad. Por ejemplo, el gráfico podría mostrar que el riesgo aumenta gradualmente con la edad hasta cierto punto, y luego se estabiliza o incluso disminuye ligeramente.

Esta representación visual nos permite comprender fácilmente cómo la edad influye en la predicción del modelo [11].

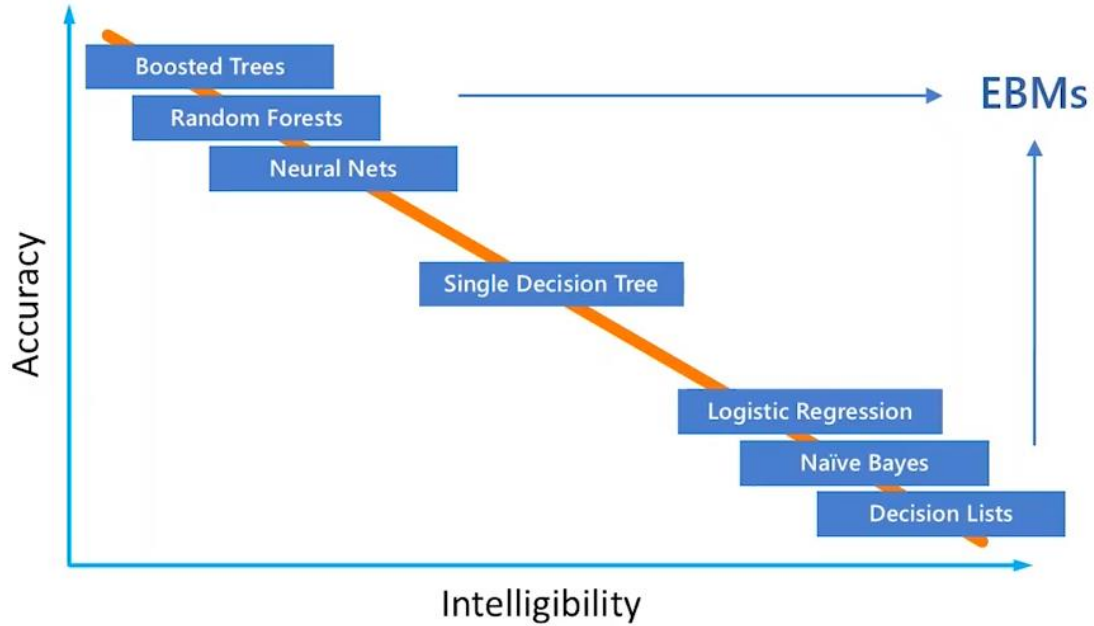


Figura 3. Gráfica en la que se muestra como EBM mantiene una alta precisión a la vez que se obtienen explicaciones de las predicciones. Figura extraída de [14].

Naive Bayes

En clasificación, el objetivo de un algoritmo de aprendizaje es construir un clasificador dado un conjunto de ejemplos de entrenamiento con etiquetas de clase. Normalmente, un ejemplo E se representa mediante una tupla de valores de atributo (x_1, x_2, \dots, x_n) , donde x_i es el valor del atributo X_i . La c y c' representa la variable de clasificación e indican su clase. En este documento, suponemos que sólo hay dos clases: c (la clase positiva) o c' (la clase negativa). Un clasificador es una función que asigna una etiqueta de clase a un ejemplo. Desde el punto de vista de la probabilidad, según la Regla de Bayes, la probabilidad de que un ejemplo $E = (x_1, x_2, \dots, x_n)$, sea de clase c es:

$$p(c|E) = \frac{p(E|c)p(c)}{p(E)}$$

E se clasifica en la clase c si y sólo si

$$f_b(E) = \frac{p(c)}{p(c')} \geq 1$$

Donde $f_b(E)$ se denomina clasificador bayesiano. Supongamos que todos los atributos son independientes dado el valor de la variable de clase; es decir

$$p(E|c) = p(x_1, x_2, \dots, x_n|c) = \prod_{i=1}^n p(x_i|c),$$

el clasificador resultante es entonces:

$$f_{nb}(E) = \frac{p(c)}{p(c')} \prod_{i=1}^n \frac{P(x_i | c)}{P(x_i | c')}$$

La función $f_{nb}(E)$ se denomina clasificador bayesiano ingenuo, o simplemente Naive Bayes (NB). La Figura 4 muestra un ejemplo de la estructura de Naive Bayes. En el Bayes ingenuo, cada nodo de atributo no tiene padre, salvo el nodo de clase:

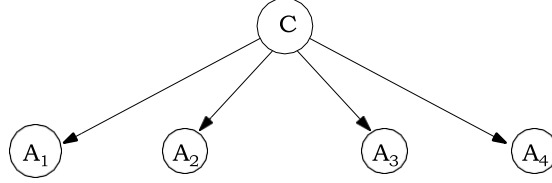


Figura 4: Ejemplo de Naive Bayes

Naive Bayes es la forma más sencilla de red bayesiana, en la que todos los atributos son independientes dado el valor de la variable de clase. Esto se denomina independencia condicional. Es obvio que el supuesto de independencia condicional rara vez se cumple en la mayoría de las aplicaciones del mundo real. Un método sencillo para superar la limitación de Bayes ingenuo es para ampliar su estructura y representar explícitamente las dependencias entre atributos.

Otra alternativa dentro de Naive Bayes es red la bayesiana ingenua aumentada, o simplemente Bayes ingenua aumentada (ANB), es una Bayes ingenua ampliada, en la que el nodo de clase apunta directamente a todos los nodos de atributo, y existen vínculos entre los nodos de atributo. La Figura 5 muestra un ejemplo de ANB. Desde el punto de vista de la probabilidad, un ANB G representa una distribución de probabilidad conjunta representada a continuación:

$$pG(x_1, \dots, x_n, c) = p(c) \sum_{i=1}^n p(x_i | pa(x_i), c),$$

donde $pa(x_i)$ denota una asignación a valores de los padres de X_i . Utilizamos $pa(X_i)$ para designar a los padres de X_i . ANB es una forma especial de redes bayesianas en las que ningún nodo se especifica como nodo de clase

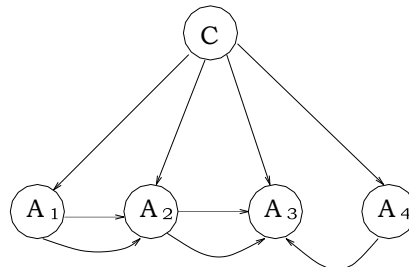


Figura 5: Ejemplo de ANB

Cuando aplicamos un logaritmo a $f_b(E)$ en la Ecuación 1, el clasificador resultante $\log f_b(E)$ es el mismo que $f_b(E)$, en el sentido de que un ejemplo E pertenece a la clase positiva, si y sólo si $\log f_b(E) \geq 0$, f_{nb} en la Ecuación 2 es similar. En este trabajo, asumimos que, dado un clasificador f , un ejemplo E pertenece a la clase positiva, si y sólo si $f(E) \geq 0$ [15].

No obstante, se va a poner el foco en el modelo de Naive Bayes, que ofrece ventajas sustanciales en términos de explicabilidad y simplicidad. Su enfoque ingenuo de asumir independencia condicional entre los atributos conlleva varias ventajas.

A través de su formulación matemática se puede ver como su eficiencia computacional es notable, ya que puede entrenarse rápidamente incluso con grandes conjuntos de datos, lo que lo hace adecuado para aplicaciones con entrenamiento en tiempo real.

Lo que buscamos es poner de manifiesto es como este modelo de aprendizaje automático, que es muchas veces obviado, de ser puesto de manifiesto en el ámbito de la Inteligencia Artificial Explicable. Por esto también vemos que en la librería interpretml tiene cabida como un modelo transparente.

Árbol de decisión)

El árbol de decisión es un gráfico que representa las elecciones y sus resultados en forma de árbol. Los nodos del gráfico representan un suceso o elección y las aristas del gráfico representan las reglas o condiciones de decisión. Cada árbol consta de nodos y ramas. Cada nodo representa atributos de un grupo que hay que clasificar y cada rama representa un valor que puede tomar el nodo [13].

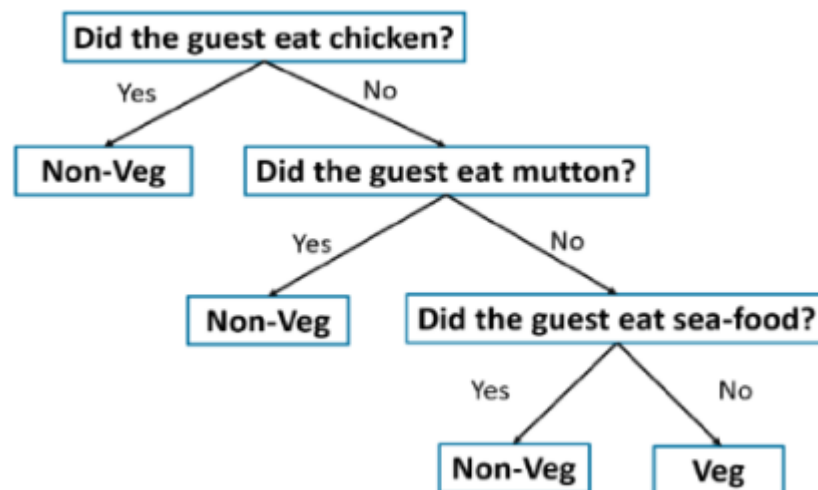


Figura 6. Ejemplo de árbol de decisión. Figura extraída de [13]

Regresión logística

La regresión logística es un modelo estadístico utilizado para predecir la probabilidad de que ocurra un evento binario (por ejemplo, sí o no, éxito o fracaso, presencia o ausencia de una enfermedad) en función de un conjunto de variables explicativas. A diferencia de la regresión lineal, que se utiliza para predecir valores numéricos continuos, la regresión logística se aplica a problemas de clasificación.

La regresión logística modela la relación entre las variables explicativas (X) y la probabilidad de que ocurra el evento de interés (P) utilizando la función logística, también conocida como sigmoide:

$$P(y = k) = \frac{1}{e^{-(B_0 + B_1 X_1 + B_2 X_2 + \dots + B_n X_n)}}$$

Donde:

- $P(y = k)$: Probabilidad de que ocurra el evento de interés ($y = k$).
- e : Base del logaritmo natural.
- β_0 : Intercepto de la ecuación, representa el logaritmo de la probabilidad (log-odds) cuando todas las variables explicativas son cero.
- $\beta_1, \beta_2, \dots, \beta_n$: Coeficientes de regresión, que representan el cambio en el log-odds por cada unidad de cambio en las variables explicativas correspondientes.
- X_1, X_2, \dots, X_n : Variables explicativas.

El término "log-odds" se utiliza principalmente en el contexto de modelos estadísticos y probabilísticos, especialmente en regresión logística y modelos lineales generalizados. Se refiere al logaritmo de las probabilidades (odds) de un evento. Como hemos visto en Naive Bayes los log-odds que miden los pesos de las características se calculan así :

$$\beta_i = \ln \frac{P(x_i | c)}{P(x_i | c')}$$

Los coeficientes de regresión (β) de la regresión logística se interpretan en términos de odds ratios , que representan la razón de probabilidades. Un OR mayor que 0 indica que un aumento en la variable explicativa se asocia con un aumento en la probabilidad del evento de interés, mientras que un OR menor que 0 indica una disminución en la probabilidad. Un OR de 1 indica que no hay asociación entre la variable explicativa y el evento de interés [16]

Random Forest

Un Random Forest (Bosque Aleatorio) es un método de aprendizaje automático que utiliza un conjunto de árboles de decisión para realizar predicciones. En lugar de depender de un solo árbol de decisión, un Random Forest crea múltiples árboles de decisión, cada uno entrenado en una muestra aleatoria de los datos y un subconjunto aleatorio de las características (variables predictoras).

El algoritmo comienza creando múltiples muestras de los datos originales mediante un proceso llamado bootstrapping, que implica seleccionar aleatoriamente con reemplazo elementos del conjunto de datos original. Esto significa que algunos datos pueden aparecer varias veces en una muestra y otros no. Para cada muestra de bootstrapping, se construye un árbol de decisión. En cada nodo del árbol, se selecciona aleatoriamente un subconjunto de características y se elige la mejor característica y punto de corte para dividir los datos en ese nodo.

Una vez que todos los árboles de decisión se han construido, se realiza una agregación de sus predicciones. Para problemas de clasificación, se utiliza votación por mayoría, donde la clase predicha por la mayoría de los árboles es la salida final. Para problemas de regresión, se promedia la predicción de cada árbol.

Además de esta agregación, los bosques aleatorios también incorporan probabilidades en su funcionamiento. Cada árbol de decisión produce una estimación de la probabilidad de que una instancia pertenezca a una clase determinada (en clasificación) o un valor predicho (en regresión). Estas probabilidades o valores se promedian a lo largo de todos los árboles para obtener una predicción final más robusta y precisa.

La combinación de múltiples árboles y el uso de probabilidades contribuyen a la eficacia de los bosques aleatorios, ya que reducen el sobreajuste y mejoran el rendimiento general del modelo en comparación con un solo árbol de decisión.

El uso de múltiples árboles de decisión y la aleatoriedad en la selección de datos y características contribuyen a la robustez y precisión del modelo. Los Random Forests suelen tener un buen rendimiento en términos de precisión predictiva y son menos propensos al sobreajuste en comparación con árboles de decisión individuales. Además, pueden manejar un gran número de características y datos faltantes de manera eficiente. Sin embargo, pueden ser más difíciles de interpretar que los árboles de decisión individuales y requieren más recursos computacionales [17].

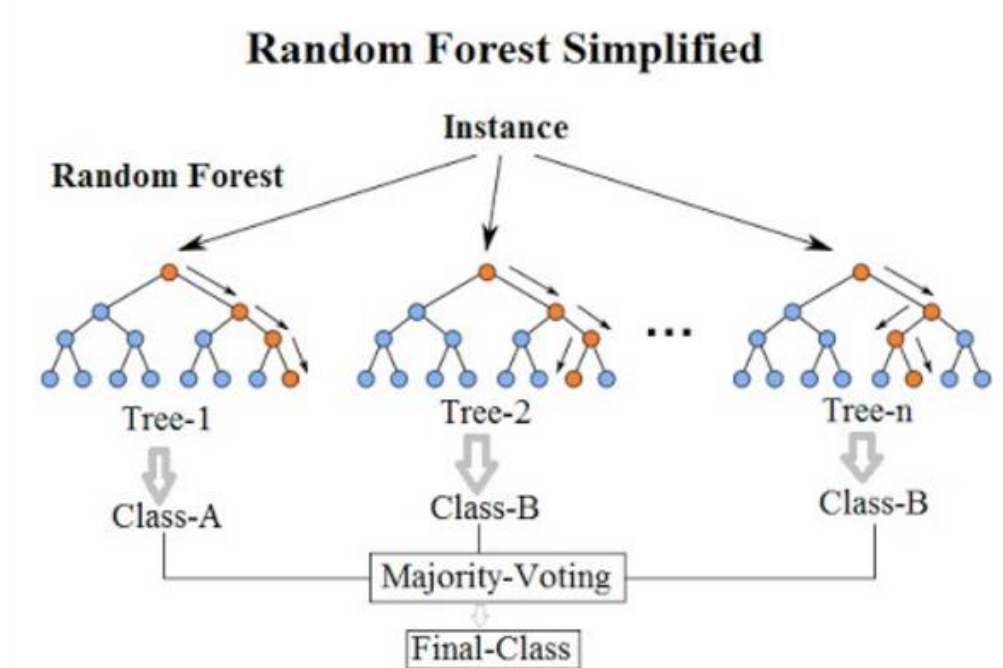


Figura 7. Ejemplo de Random Forest simplificado. Figura extraída de [18]

KGBoost

XGBoost (Extreme Gradient Boosting) es un algoritmo de aprendizaje automático que pertenece a la familia de los métodos de boosting, específicamente a los modelos basados en árboles de decisión. Ha ganado popularidad en el aprendizaje automático debido a su alta precisión y eficiencia computacional.

XGBoost construye un modelo predictivo combinando múltiples árboles de decisión de manera secuencial. El proceso comienza con un árbol de decisión simple que realiza predicciones iniciales. Luego, se calculan los errores (residuales) entre las predicciones del árbol inicial y los valores reales. A continuación, se construye un nuevo árbol de decisión que se enfoca en predecir estos residuales. El modelo se actualiza sumando las predicciones del nuevo árbol a las predicciones del modelo anterior. Este proceso se repite varias veces, agregando árboles sucesivos que se enfocan en los errores residuales del modelo anterior.

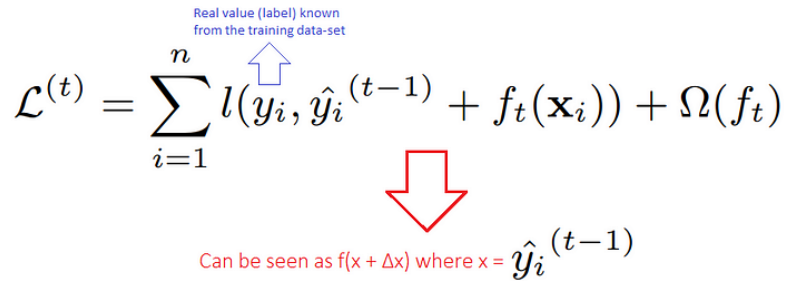
Para evitar el sobreajuste, XGBoost utiliza técnicas de regularización como la poda de árboles y la restricción de la complejidad del modelo. La predicción final de XGBoost se obtiene sumando las predicciones de todos los árboles de decisión individuales.

La tasa de aprendizaje es un parámetro que controla cuánto contribuye cada árbol al modelo final. Un valor bajo para la tasa de aprendizaje significa que cada árbol tiene un impacto menor, lo que resulta en un proceso de aprendizaje más lento, pero potencialmente más preciso, ya que el modelo es menos propenso a sobreajustarse a los datos de entrenamiento.

El subsampling, o submuestreo, es una técnica utilizada en XGBoost para mejorar la generalización del modelo y reducir el sobreajuste. Consiste en entrenar cada árbol en un subconjunto aleatorio de las instancias de entrenamiento y/o un subconjunto aleatorio de las características. Al introducir esta aleatoriedad en el proceso de entrenamiento, se evita que los árboles se ajusten demasiado a patrones específicos en los datos de entrenamiento, lo que resulta en un modelo más robusto y generalizable[19].

La mayoría de las veces función objetivo (función de pérdida y regularización) en la iteración t que necesitamos minimizar es la siguiente:

$$\mathcal{L}^{(t)} = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t-1)} + f_t(\mathbf{x}_i)) + \Omega(f_t)$$



Can be seen as $f(x + \Delta x)$ where $x = \hat{y}_i^{(t-1)}$

Figura 8. La función objetivo de KGBost. Figura extraída de [20]

Los términos en la fórmula son:

- $\mathcal{L}^{(t)}$: Es la función de pérdida total en la iteración t del algoritmo de boosting. Representa el error que el modelo está tratando de minimizar.

- $l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i))$: Es la función de pérdida para la i-ésima instancia. Mide la diferencia entre el valor real (y_i) (conocido del conjunto de datos de entrenamiento) y la predicción actual del modelo ($\hat{y}_i^{(t-1)} + f_t(x_i)$). Aquí, ($\hat{y}_i^{(t-1)}$) es la predicción del modelo en la iteración anterior (t-1) y ($f_t(x_i)$) es la contribución del árbol débil actual a la predicción.
- $f_t(x_i)$: Es la función de pérdida para la i-ésima instancia. Mide la diferencia entre el valor real (y_i) (conocido del conjunto de datos de entrenamiento) y la predicción actual del modelo
- $\Omega(f_t)$: Es el término de regularización para el árbol débil actual (f_t). Ayuda a prevenir el sobreajuste al penalizar la complejidad del modelo.

Hemos desarrollado un análisis más profundo de Naive Bayes y Explainable Boosting Machine, ya que estos modelos son clave en este trabajo. Como ya hemos introducido anteriormente tanto el modelo de Naive Bayes como el de Explainable Boosting Machine tienen un rol importante en los modelos transparentes.

4 Desarrollo

4.1 Conjunto de datos

Para llevar a cabo un análisis exhaustivo del rendimiento de los modelos de clasificación transparentes, se ha optado por utilizar la plataforma OpenML, un repositorio en línea que alberga una vasta colección de conjuntos de datos y tareas de aprendizaje automático. Esta plataforma ofrece una interfaz de programación de aplicaciones (API) que facilita la descarga y el acceso a los datos de manera programática [21].

4.1.1 OpenML: Una Plataforma Colaborativa para la Ciencia de Datos

OpenML es una plataforma abierta y colaborativa diseñada para fomentar la investigación y el desarrollo en el campo del aprendizaje automático. Su objetivo principal es proporcionar un espacio donde investigadores, científicos de datos y entusiastas puedan compartir conjuntos de datos, algoritmos y resultados de experimentos de manera transparente y reproducible.

Algunas de las ventajas clave de utilizar OpenML en este proyecto son:

- **Amplia Colección de Datos:** OpenML alberga una gran variedad de conjuntos de datos de diferentes dominios, lo que permite evaluar el rendimiento de los modelos en una amplia gama de escenarios y problemas del mundo real.
- **Metadatos Detallados:** Cada conjunto de datos en OpenML viene acompañado de metadatos exhaustivos que describen sus características, como el número de instancias, el número de características, el tipo de datos de cada característica y la tarea de aprendizaje automático asociada (clasificación, regresión, etc.).
- **Facilidad de Uso:** La API de OpenML proporciona una forma sencilla y eficiente de acceder a los datos y metadatos de manera programática, lo que facilita la automatización de los procesos de carga y preprocesamiento de datos.
- **Reproducibilidad:** OpenML promueve la reproducibilidad de los experimentos al proporcionar un entorno estandarizado para compartir datos y resultados. Esto permite a otros investigadores verificar y replicar los resultados obtenidos en este proyecto.

4.1.2 Selección y Carga de Datos

En este proyecto, se ha utilizado la API de OpenML para acceder a un benchmark de conjuntos de datos (`suite_id=99`). Este benchmark incluye una selección diversa de conjuntos de datos de clasificación, abarcando diferentes tamaños, número de características y tipos de datos. En el código se obtiene información sobre el benchmark y se listan los ID de los conjuntos de datos incluidos.

Para cada conjunto de datos, el código realiza los siguientes pasos:

1. **Descarga de Datos y Metadatos:** Se utiliza la función `openml.datasets.get_dataset` para descargar los datos y metadatos del conjunto de datos correspondiente. Se especifican opciones para

- descargar las calidades de los datos (`download_qualities=True`) y los metadatos de las características (`download_features_meta_data=True`).
2. **Visualización de Información:** Se imprime el nombre y la descripción del conjunto de datos, así como las primeras filas del DataFrame de pandas que contiene los datos.
 3. **Preparación de Datos:** Los datos se dividen en características (X) y etiquetas (y), y se almacenan en un diccionario junto con información sobre el tipo de problema (clasificación en este caso).

El código también incluye un pipeline de preprocesamiento de datos que se aplica a cada conjunto de datos antes de entrenar los modelos de clasificación. Este pipeline es fundamental para preparar los datos de manera adecuada, garantizando que los algoritmos de aprendizaje automático puedan trabajar con ellos de manera eficiente y precisa. El preprocesamiento se divide en varias etapas:

1. **Identificación de Variables Categóricas:** El primer paso consiste en identificar qué variables del conjunto de datos son categóricas. Esto se logra examinando el tipo de datos de cada columna. Si una columna contiene valores no numéricos (como cadenas de texto o etiquetas), se considera categórica.
2. **Codificación One-Hot:** Las variables categóricas no pueden ser utilizadas directamente por la mayoría de los algoritmos de aprendizaje automático, ya que estos esperan entradas numéricas. Por lo tanto, se utiliza la técnica de One-Hot Encoding para transformar las variables categóricas en un formato adecuado. One-Hot Encoding crea nuevas variables binarias (0 o 1) para cada categoría de la variable original. Por ejemplo, si una variable categórica representa el color de un objeto (rojo, verde, azul), se crearían tres nuevas variables binarias, una para cada color.
3. **Imputación de Valores Faltantes:** Es común que los conjuntos de datos del mundo real contengan valores faltantes. Estos valores pueden deberse a errores de medición, falta de información o problemas en la recopilación de datos. Para abordar este problema, el pipeline utiliza la imputación de valores faltantes. En este caso, los valores faltantes en las variables numéricas y categóricas se reemplazan por la media de cada columna. Esto asegura que todas las instancias tengan valores completos para todas las características.
4. **Escalado Estándar:** Las variables numéricas pueden tener diferentes escalas y rangos, lo que puede afectar el rendimiento de algunos algoritmos de aprendizaje automático. El escalado estándar (o estandarización) se utiliza para llevar todas las variables numéricas a una escala común, con media cero y desviación estándar uno. Esto hace que todas las características tengan la misma importancia relativa en el entrenamiento del modelo.

En resumen, este pipeline de preprocesamiento de datos transforma los datos en un formato adecuado para el entrenamiento de modelos de aprendizaje automático, garantizando que todas las características sean numéricas, no tengan valores faltantes y estén en una escala común. Esto es crucial para obtener resultados precisos y confiables en la evaluación del rendimiento de los modelos de clasificación.

4.2 Metodología para la aplicación de los algoritmos

Para evaluar el rendimiento predictivo de los métodos transparentes en clasificación, hemos diseñado un benchmark exhaustivo utilizando la plataforma OpenML, como ya hemos explicado antes. Este benchmark permite comparar sistemáticamente el desempeño de diferentes algoritmos en una amplia variedad de conjuntos de datos, proporcionando una visión completa de sus fortalezas y debilidades en diferentes escenarios.

4.2.1 Diseño del Benchmark

Como ya se ha explicado en la anterior parte de Selección y Carga de Datos, el benchmark se ha construido utilizando la suite de OpenML con ID 99, que contiene una colección diversa de conjuntos de datos de clasificación. Estas bases de datos tienen diferentes tamaños (desde decenas hasta miles de instancias), número de características (variables predictoras) y tipos de datos (numéricos, categóricos). Esta diversidad permite evaluar el rendimiento de los modelos en una amplia gama de condiciones, desde problemas simples hasta problemas más complejos y desafiantes.

La elección de esta suite en particular se debe a su relevancia para el estudio de métodos transparentes. Muchos de los conjuntos de datos incluidos en este repositorio son problemas de clasificación en los que la interpretabilidad y la explicabilidad de los modelos son importantes. Esto se alinea con nuestro objetivo de evaluar la eficacia de los métodos transparentes en escenarios del mundo real donde la transparencia es un factor clave.

La estructura del benchmark consiste en aplicar un conjunto de modelos de clasificación a cada uno de los conjuntos de datos de la suite, evaluando su rendimiento mediante validación cruzada estratificada y utilizando el área bajo la curva ROC (AUC-ROC) como métrica principal. El objetivo es obtener una visión comparativa del rendimiento de los diferentes modelos en una variedad de escenarios.

En las siguientes secciones, profundizaremos en la metodología de evaluación de los modelos y en los detalles de los algoritmos utilizados en el benchmark.

4.2.2 Evaluación de Modelos y Métricas

La función `process_model` es fundamental en nuestro benchmark, ya que se encarga de evaluar el rendimiento de cada modelo de clasificación en un conjunto de datos dado. Para ello, utiliza la técnica de validación cruzada estratificada, que divide los datos en varios pliegues (en nuestro caso, 3 pliegues) y entrena el modelo en todos los pliegues excepto uno, que se utiliza para evaluar su rendimiento. Este proceso se repite para cada pliegue, y los resultados se promedian para obtener una estimación más robusta del rendimiento del modelo.

La métrica de evaluación utilizada es el área bajo la curva ROC (AUC-ROC). Esta es una métrica numérica derivada de una curva, que resume en un solo valor la capacidad del modelo para discriminar entre clases. Un AUC-ROC de 1 indica un clasificador perfecto, capaz de distinguir completamente entre las dos clases, mientras que un valor de 0.5 representa un rendimiento equivalente al azar.

La curva ROC se construye graficando la tasa de verdaderos positivos (TPR) en el eje vertical y la tasa de falsos positivos (FPR) en el eje horizontal, para diferentes umbrales de clasificación. La TPR mide la proporción de casos positivos que el modelo clasifica correctamente, mientras que la FPR mide la proporción de casos negativos que el modelo clasifica incorrectamente como positivos [22].

El AUC-ROC es una métrica de evaluación más aconsejable que las métricas de precisión, recall y la puntuación F1 en nuestro caso, por varios aspectos clave:

- **Robustez ante clases desbalanceadas:** En conjuntos de datos donde una clase es mucho más común que la otra, la precisión puede ser engañosa, ya que un modelo puede lograr una alta precisión simplemente prediciendo la clase mayoritaria. El AUC-ROC, al no depender de la distribución de clases, ofrece una evaluación más fiable en estos casos.
- **Interpretación probabilística:** El AUC-ROC representa la probabilidad de que el modelo clasifique correctamente un par de instancias (una positiva y una negativa). Esta interpretación probabilística es más intuitiva y fácil de entender que los valores de precisión, recall o F1, que dependen de un umbral de clasificación específico.
- **Independencia del umbral:** El AUC-ROC considera todos los posibles umbrales de clasificación, mientras que la precisión, el recall y la F1 se calculan para un umbral determinado. Esto hace que el AUC-ROC sea más útil para comparar modelos con diferentes estrategias de decisión, ya que no está influenciado por la elección arbitraria de un umbral.

El AUC-ROC es una métrica más completa y versátil que la precisión, el recall y la puntuación F1, especialmente cuando se requiere una evaluación independiente del umbral de clasificación. Aunque las otras métricas pueden ser útiles en contextos específicos, el AUC-ROC proporciona una visión más global y robusta del rendimiento del modelo.

En esta función se almacenan para cada modelo que predice en la base de datos:

- El nombre del modelo.
- Tiempo medio de ajuste (`fit_time_mean`) y desviación estándar (`fit_time_std`).
- Puntuación media de la prueba (`test_score_mean`) y desviación estándar (`test_score_std`).

4.2.3 Benchmarking de Modelos

La función `benchmark_models` es el corazón del benchmark. Se encarga de evaluar varios modelos de clasificación en un conjunto de datos dado, aplicando un preprocesamiento de datos estándar y utilizando la función `process_model` para obtener los resultados de cada modelo. Se va a enumerar los modelos evaluados, ya fueron descritos en el apartado de fundamentos e incluyen:

- Regresión Logística
- Random Forest
- XGBoost
- EBM
- Árbol de Decisión
- Naive Bayes

Al evaluar estos modelos en una variedad de conjuntos de datos, podemos comparar su rendimiento y determinar cuáles son más adecuados para diferentes tipos de problemas y características de datos. Esto permitirá extraer conclusiones sobre la eficacia de los métodos transparentes en general y de cada algoritmo en particular.

4.3 Integración de interpretml

Naive Bayes es un clasificador probabilístico que se basa en el teorema de Bayes para calcular la probabilidad de que una instancia pertenezca a una clase determinada, dadas sus características. En el caso de clasificación binaria (dos clases), la decisión de clasificación se basa en la siguiente función discriminante:

$$d(x) = \ln \frac{P(c|x)}{P(c'|x)} = \ln \frac{P(c)}{P(c')} + \sum_{i=1}^n \ln \frac{P(x_i|c)}{P(x_i|c')}$$

Donde:

- $d(x)$: Valor discriminante para la instancia x .
- $P(c|x)$: Probabilidad posterior de la clase c , dado el vector de características x .
- $P(c'|x)$: Probabilidad posterior de la clase c' , dado el vector de características x .
- $P(c)$: Probabilidad previa de la clase c .
- $P(c')$: Probabilidad previa de la clase c' .
- $P(x_i|c)$: Probabilidad condicional de la característica x_i , dado que la instancia pertenece a la clase c .
- $P(x_i|c')$: Probabilidad condicional de la característica x_i , dado que la instancia pertenece a la clase c' .
- (\ln) : Logaritmo natural.

La decisión de clasificación se basa en un límite de decisión, que podríamos mostrar de esta forma:

$$f(x) = \begin{cases} c & \text{if } d(x) > 0 \\ \text{decision boundary} & \text{if } d(x) = 0 \\ c' & \text{if } d(x) < 0. \end{cases}$$

De esta forma, para una instancia con las características X_n en concreto, dependiendo del resultado que obtengamos de la función discriminante vamos a poder afirmar a que clase en concreto pertenecería.

Los pesos de las variables se derivan de los términos logarítmicos de la razón de verosimilitudes:

$$\ln \frac{P(x_i|c)}{P(x_i|c')}$$

Estos pesos cuantifican la contribución de cada característica individual a la discriminación entre las clases. Un peso positivo para una característica indica que su presencia aumenta la probabilidad de que la instancia pertenezca a la clase c , mientras que un peso negativo sugiere lo contrario.

Para calcular los pesos de las variables en un modelo Naive Bayes Gaussiano, asumimos que cada característica sigue una distribución normal (gaussiana) dentro de cada clase. La "densidad" en este contexto se refiere al valor de la función de densidad de una característica dada una clase específica.

Para calcular los pesos tenemos varias opciones dependiendo de que tipo de variables sean nuestras características.

Variables Binarias (Bernoulli Naive Bayes)

La distribución de Bernoulli es una distribución de probabilidad discreta que toma valor 1 con probabilidad p y valor 0 con probabilidad $1-p$. Es una distribución muy simple y es la base para la distribución binomial.

Cálculo de $p(x_i | c)$:

- $p(x_i=1|c)=p_c$ (probabilidad de que la característica esté presente en la clase c)
- $p(x_i=1|c)=1-p_c$ (probabilidad de que la característica no esté presente en la clase c)

Variables Categóricas (Categorical Naive Bayes)

La distribución categórica es una generalización de la distribución de Bernoulli. En lugar de tener solo dos posibles resultados (como en la distribución de Bernoulli). Cálculo de $p(x_i | c)$:

$$P(X_i = t | c) = \frac{N_{tic} + \alpha}{N_c + \alpha n_i}$$

Donde:

- N_{tic} es el número de veces que la categoría t aparece en la clase c .
- N_c es el número total de muestras en la clase c .
- n_i es el número de categorías posibles para la característica x_i .
- α es un parámetro de suavizado (evita probabilidades cero; normalmente 1 o 0.5).

Variables Continuas (Gaussian Naive Bayes)

La densidad de probabilidad de una variable aleatoria gaussiana se define como:

$$p(x_i | c) = \frac{1}{\sqrt{2\pi}\sigma_c} \exp\left(-\frac{(x_i - \mu_c)^2}{2\sigma_c^2}\right)$$

Donde:

- μ es la media de la característica i en la clase c .
- σ es la desviación estándar de la característica i en la clase c .

Lo que hemos realizado para integrar el nuevo código generado en la librería ha sido generar las funciones de explicación global y local para Naive Bayes.

Explicación del Código

Funciones : `feature_weights_naive_bayes_local` y `feature_weights_naive_bayes_global`. Siguen una lógica similar, pero con algunas diferencias clave:

1. **Manejo de Entradas:**
 - `feature_weights_naive_bayes_local`: Recibe una observación individual (X), los datos de entrenamiento (X_{train}) y las etiquetas de clase (y_{train}).
 - `feature_weights_naive_bayes_global`: Recibe todos los datos (X) y sus etiquetas (y).
2. **Cálculo de Probabilidades Previas:**
 - En ambos casos, se calculan las probabilidades previas de cada clase (`class_prior` y `other_prior`). Se usa un pequeño valor (`epsilon`) para evitar divisiones por cero al calcular la probabilidad de la otra clase.
 - El uso de `epsilon` es crucial para evitar problemas numéricos (divisiones por cero y logaritmos de cero) que podrían llevar a resultados incorrectos o errores en el cálculo de los pesos.
 - Estas funciones examinan que tipo de distribución se amolda a cada característica como hemos visto antes en como calcular los pesos.
3. **Cálculo de Pesos:**
 - **Local:**
 - Itera sobre cada característica (i) y cada clase (c).
 - Calcula la media y desviación estándar de la característica i para la clase c y para las otras clases.
 - Calcula la probabilidad de la característica i dado que pertenece a la clase c (`p_xi_given_c`) y dado que pertenece a otra clase (`p_xi_given_other`), utilizando las diferentes fórmulas que hemos visto antes dependiendo del tipo de variables. Se utiliza `epsilon` para evitar divisiones por cero y logaritmos de cero.
 - Actualiza el peso de la característica i sumando el logaritmo de la razón de verosimilitudes (`np.log(p_xi_given_c / p_xi_given_other)`).
 - **Global:**
 - Similar al caso local, pero en lugar de iterar sobre una observación, se realizan los cálculos para todas las observaciones a la vez usando operaciones vectorizadas de NumPy.
 - Los pesos se calculan promediando los logaritmos de las razones de verosimilitudes para cada característica en todas las observaciones.

4. Retorno:

- Ambas funciones devuelven el intercepto (peso de la clase) y un array de pesos para cada característica.

Funcion: `plot_feature_importance_local_naive_bayes` y `plot_feature_importance_global_naive_bayes`. Siguen una lógica similar, pero con algunas diferencias clave

`plot_feature_importance_local_naive_bayes`

1. Cálculo de Pesos:

- Llama a la función `feature_weights_naive_bayes_local` (que asumimos ya está definida) para calcular los pesos de las características y el intercepto para una observación específica (`x_predict`).

2. Preparación de Datos para el Gráfico:

- Combina el intercepto y los pesos en un solo array (`coefficients`).
- Crea un DataFrame de Pandas (`df`) con los coeficientes y los nombres de las características.
- Reordena el DataFrame por el valor absoluto de los coeficientes en orden ascendente para que el gráfico sea más legible.
- Asigna colores a las barras del gráfico: gris para el intercepto, naranja para coeficientes positivos (características que aumentan la probabilidad de la clase objetivo) y azul cielo para coeficientes negativos (características que disminuyen la probabilidad de la clase objetivo).

3. Creación del Gráfico:

- Crea una figura de matplotlib con un tamaño específico.
- Dibuja un gráfico de barras horizontales (`plt.barh`) con los nombres de las características en el eje y los coeficientes en el eje x.
- Agrega una línea vertical discontinua en cero para separar los coeficientes positivos de los negativos.
- Establece el título del gráfico, etiquetas de los ejes y ajusta el diseño.

`plot_feature_importance_global_naive_bayes`

Esta función es muy similar a `plot_feature_importance_local_naive_bayes`, pero en lugar de calcular los pesos para una observación específica, utiliza la función `feature_weights_naive_bayes_global` para calcular los pesos promediados en todas las observaciones del conjunto de datos. El resto del código es idéntico, generando un gráfico de barras horizontales que muestra la importancia global de cada característica en el modelo Naive Bayes.

Cabe destacar que, aunque se haya integrado las funciones de `explain_local` y `explain_global` de la librería `interpretml` gracias a las funciones que nos permiten calcular los pesos, la parte de visualización se ha desarrollado en local con las funciones que hemos visto ahora. Esto es algo que queda en futuro trabajo ya que su complejidad es reseñable. No obstante, el concepto que vamos a desarrollar es el mismo.

4.3.1 Ejemplo de uso

El código de ejemplo carga el conjunto de datos de cáncer de mama de scikit-learn (`load_breast_cancer()`), convierte los datos y las etiquetas al formato NumPy, y luego llama a las funciones para graficar la importancia de las características tanto a nivel local (para la observación número 20 que ha sido tomada al azar) como global. Hemos decidido tomar estos datos ya que son los que se utilizan en la página de documentación de `interpretml` para poner ejemplos de cómo funcionan para Linear Model y árbol de decisión, así los resultados serán más comparables.

Interpretación de los Gráficos

- **Importancia Local:** Muestra cómo cada característica contribuye a la predicción de la clase para una observación específica. Las barras más largas indican características más importantes, y el color indica si aumentan (naranja) o disminuyen (azul) la probabilidad de la clase objetivo. Se ve como el intercepto muestra que las clases están desbalanceadas decantándose para el lado positivo

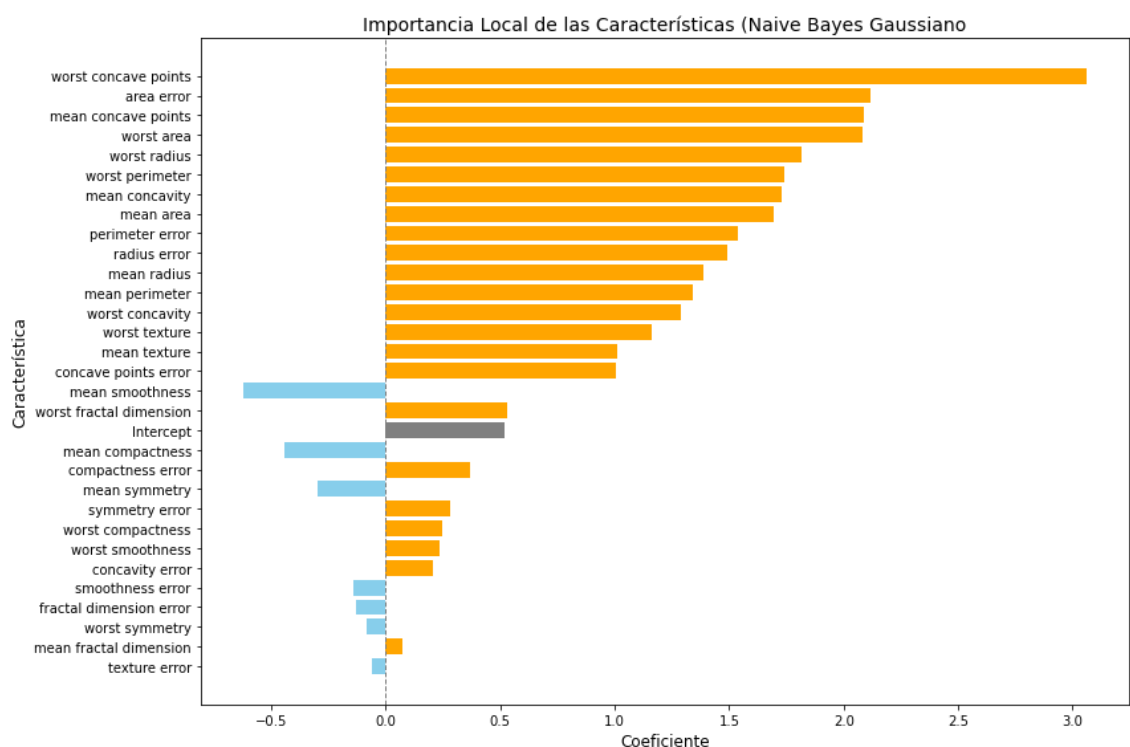


Figura 9 . Muestra la gráfica que saca `plot_feature_importance_global_naive_bayes` para la base de datos `load_breast_cancer()` en la que se ve el peso de todas las variables de forma global.

- **Importancia Global:** Muestra la importancia promedio de cada característica en el modelo para todas las observaciones. Esto nos da una idea de qué características son generalmente más importantes para el modelo al hacer predicciones. Como se ha comentado en la importancia

local aquí también se ve como el intercepto muestra que las clases están desbalanceadas decantándose para el lado positivo

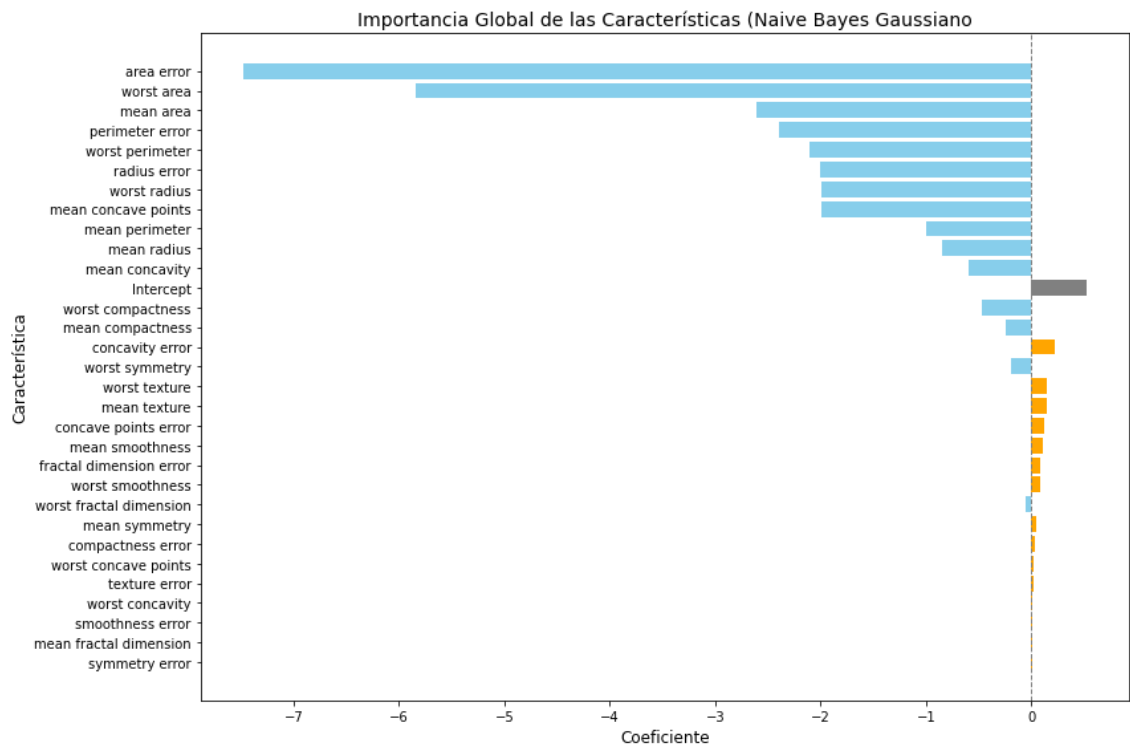


Figura 10 . Muestra la gráfica que saca `plot_feature_importance_local_naive_bayes` para la base de datos `load_breast_cancer()` en la que se ve el peso de todas las variables de para una instancia en concreto.

En el contexto del modelo Naive Bayes y los gráficos generados por las funciones `plot_feature_importance_local_naive_bayes` y `plot_feature_importance_global_naive_bayes`, el signo (positivo o negativo) de los pesos de las características tiene una interpretación crucial para entender cómo cada característica influye en la predicción de la clase.

Pesos Positivos (Barras Naranjas)

Un peso positivo para una característica indica que, en general, un valor más alto de esa característica está asociado con una mayor probabilidad de pertenecer a la clase objetivo. Por ejemplo, en la Figura 10, vemos que características como "worst concave points" y "área error" (tienen pesos positivos, lo que sugiere que tumores con valores más altos en estas características son más propensos a ser malignos).

Pesos Negativos (Barras Azules)

Un peso negativo indica que un valor más alto de la característica está asociado con una menor probabilidad de pertenecer a la clase objetivo (en este caso,

"benigno"). En la Figura 10, "mean smoothness" tiene un peso negativo, lo que implica que tumores más suaves tienden a ser benignos.

Comparación de este ejemplo de uso de Naive Byaes respecto a Regresión Logística en la documentación de interpretml

Se va a analizar los resultados de la importancia de características al utilizar un modelo Naive Bayes gaussiano y una regresión logística sobre el conjunto de datos de cáncer de mama de scikit-learn. (load_breast_cancer()), de la cual hemos hablado anteriormente.

En esta imagen podemos ver como se ha generado la gráfica que pondera los pesos de las diferentes características de la base de datos para la regresión logística.

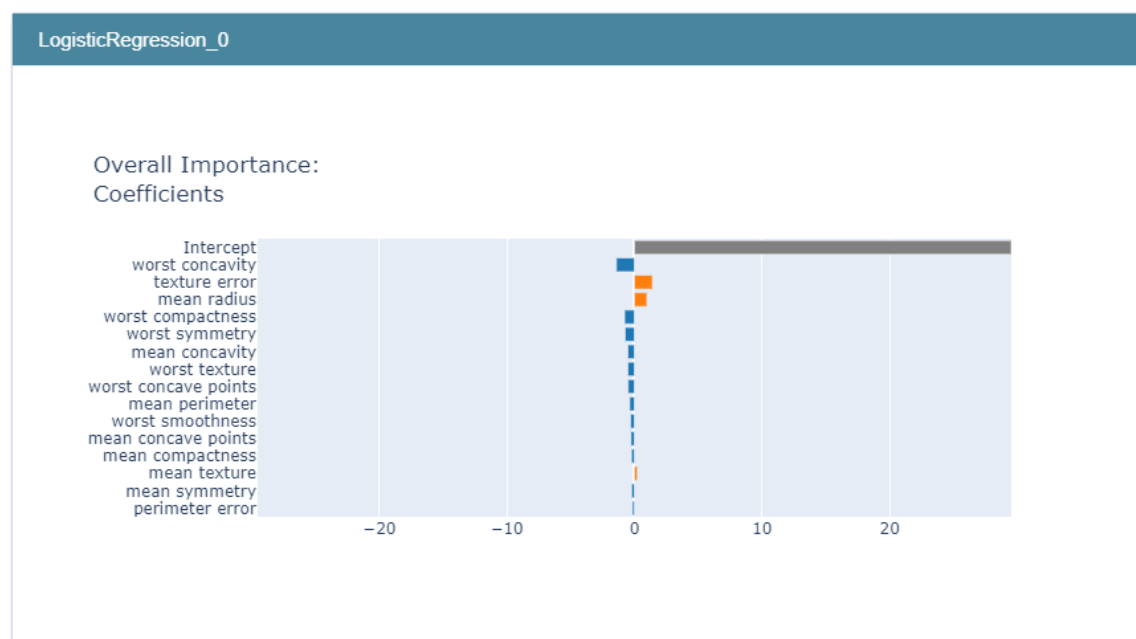


Figura 11. Se muestra la gráfica de explicabilidad global para la regresión logística entrenada con los mismos datos que nosotros. Figura extraída de [23]

Similitudes

Comparando esta gráfica respecto a la Figura 10 se pueden ver varias semejanzas y diferencias. Por un lado, se puede ver como ambas gráficas tienen la mayoría de los coeficientes en negativo, lo cual nos puede hacer indicar que para estos datos y la tendencia de predicción de esa clase es influida por este grupo de características. Aunque, en realidad estos pesos comparándolos con el intercepto no tienen mucha influencia.

Si concretamos más aun podemos ver cómo hay ciertas características con pesos relativamente altos para ambas figuras, este es el caso de 'worst compactness' o 'worst symmetry'.

Diferencias

No obstante, las diferencias también son notables, empezando por el peso del intercepto en la predicción de la clase. Vemos como para la Figura 11 este componente tiene mucho más peso que para la Figura 10, aunque ambos sean positivos, vemos así que para el modelo de regresión logística el intercepto tiene gran influencia.

Se puede ver como para las mismas características tienen diferentes coeficientes, en la Figura 10 vemos como 'area error' y 'worst area' son las variables más influyentes, sin embargo, en la Figura 11 como 'worst concavity' y 'texture error' son las de mayores pesos.

Aunque ambos modelos coinciden en la relevancia de algunas características, existen diferencias notables en la importancia relativa que asignan a cada una y en el papel del intercept. Estas discrepancias resaltan las distintas suposiciones y mecanismos de modelado de cada algoritmo.

Es verdad que la magnitud de los pesos para diferentes modelos o la obtención de unos umbrales que nos permitan saber si una característica es relevante o no, son dilemas que todavía se plantean.

Comparar la importancia de las características entre diferentes modelos, como se ha hecho en este ejemplo, es fundamental para comprender cómo cada algoritmo toma decisiones y qué factores considera más relevantes. Este análisis no solo ayuda a seleccionar el modelo más adecuado para un problema específico, sino que también proporciona información valiosa sobre los datos y el fenómeno que se está estudiando.

5 Resultados y conclusiones

5.1 Resultados del Benchmark

El benchmark exhaustivo, realizado sobre una selección diversa de 20 conjuntos de datos de OpenML, abarcó desde problemas simples con pocas características hasta problemas más complejos con un gran número de características y clases. Esta diversidad permitió evaluar el rendimiento de los modelos en una amplia gama de escenarios del mundo real. En la siguiente tabla están los datos de las bases de datos que se han utilizado.

	Nombre de las bases de datos	Número de datos	Número de características	Número de clases a predecir
1	kr-vs-kp	3196	36	2
2	letter	20000	16	26
3	balance-scale	625	4	3
4	mfeat-factors	1000	20	2
5	mfeat-fourier	768	8	2
6	breast-w	699	9	2
7	mfeat-karhunen	2000	64	10
	mfeat-			
8	morphological	2000	6	10
9	mfeat-zernike	2000	47	10
10	cmc	1473	9	3
	Optical-			
11	Recognition	5620	64	10
12	Credit-Approval	690	15	2
13	Credit-g	1000	20	2
14	pendigits	10992	16	10
15	diabetes	10992	16	10
16	spambase	4601	57	2
17	splice	3190	60	3
18	tic_tac_toe	958	9	2
19	tic_tac_toe	846	18	4
20	electricity	45312	8	2

Tabla 1. Los metadatos de las bases de datos utilizadas.

En este estudio, se han utilizado varios conjuntos de datos que ejemplifican la importancia de la interpretabilidad en diferentes contextos, aquí se van a mostrar dos de ellos:

- **'breast-w' (Cáncer de mama en Wisconsin):** Este conjunto de datos médicos contiene características de células extraídas de biopsias de mama, con el objetivo de clasificarlas como benignas o malignas. En este caso, la interpretabilidad del modelo es crucial, ya que los médicos necesitan comprender las razones detrás de cada diagnóstico para tomar decisiones informadas sobre el tratamiento y el seguimiento de los pacientes. Un modelo de caja negra, aunque preciso, no proporcionaría la información necesaria para justificar un diagnóstico y generar confianza en el paciente. Por lo tanto, en este escenario, un método

transparente como EBM sería más adecuado, ya que permite a los médicos comprender qué características celulares son más relevantes para el diagnóstico y cómo contribuyen a la decisión final.

- **'credit-approval' (Aprobación de crédito):** Estos conjuntos de datos contienen información sobre solicitantes de crédito, y el objetivo es predecir si se les debe otorgar o no un crédito. Aunque la precisión es importante en este contexto para minimizar el riesgo de impago, la interpretabilidad también es relevante. Las instituciones financieras deben poder explicar las razones detrás de la denegación de un crédito para cumplir con las regulaciones y evitar posibles discriminaciones. Un modelo transparente como EBM puede ayudar a identificar qué factores, como el historial crediticio o los ingresos, son más determinantes en la decisión de otorgar un crédito, lo que permite una evaluación más justa y transparente de las solicitudes.

En resumen, estos dos ejemplos ilustran cómo la interpretabilidad puede ser crucial en diferentes dominios, ya sea para tomar decisiones médicas informadas o para garantizar la equidad en la evaluación de solicitudes de crédito. La elección del modelo adecuado debe considerar no solo la precisión, sino también la necesidad de comprender y justificar las decisiones automatizadas.

Aquí podemos ver la tabla de los resultados. La precisión de los diferentes modelos para las bases de datos utiliza la curva AUC y el más menos después de la precisión es la varianza que han tenido los al realizar varias predicciones en forma de validación cruzada :

Dataset	LR	RF-100	XGB	EBM	DT	NB
1 kr-vs-kp	99.4+-0.1	100+-0	100+-0	99.9+-0	99.5+-0.2	81.9+-2.3
2 letter	98.0+-0	99.9+-0	100+-0	99.5+-0	93.3+-0.2	96.7+-0.1
3 balance-scale	96.7+-0,9	82.7+-1.2	91.7+-1.2	99.7+-0.3	73.6+-2.7	86.2+-0.7
4 mfeat-factors	76.2+-0,9	78.9+-1.1	76.8+-1.6	77.0+-0.6	63.5+-3.8	73.7+-1.9
5 mfeat-fourier	84.1+-2	83.8+-3.3	80.7+-5.3	86.0+-2.4	68.1+-1.8	80.3+-3.1
6 breast-w mfeat-	99.5+-0	99.1+-0	99.1+-0.1	99.6+-0.1	91.3+-2.7	98.7+-0.5
7 karhunen mfeat-	99.7+-0	99.8+-0	99.8+-0.1	99.9+-0	88.9+-1.1	99.7+-0
8 morphological	96.4+-0.5	95.2+-0.6	95.6+-0.5	96.1+-0.5	81.2+-1.2	95.4+-0.6
9 mfeat-zernike	98.3+-0.1	96.9+-0.2	96.7+-0	97.6+-0.1	81.1+-0.5	96.1+-0.2
10 cmc Optical-	69.7+-0.6	67.7+-1.5	70.1+-0.8	72.5+-0.8	60.1+-1.4	63.8+-1.2
11 Recognition Credit-	99.9+-0	100+-0	99.9+-0	99.9+-0	94.7+-0.4	98.2+-0.2
12 Approval	91.7+-1.5	93.3+-1	93.4+-0.7	94.4+-0.9	80.5+-1	90.5+-3.1
13 Credit-g	76.2+-0.9	78.9+-1.1	76.8+-1.6	77+-0.6	63.5+-3.8	73.7+-1.9
14 pendigits	99.8+-0	100+-0	100+-0	99.9+-0	97.7+-0.3	98.0+-0.1
15 diabetes	99.8+-0	100+-0	100+-0	99.9+-0	97.7+-0.3	98.0+-0.1
16 spambase	97.0+-0	98.5+-0.1	98.7+-0.1	99.7+-0.1	91.8+-1.5	93.6+-1.1
17 splice	98.7+-0.2	99.5+-0.1	99.5+-0.1	94.4+-1	93.3+-3.2	99.3+-0.6
18 tic_tac_toe	99.3+-0.6	99.9+-0	99.9+-0.1	99.9+-0.1	93.2+-2.1	76.2+-2.5
19 vehicle	94.1+-0.2	92.4+-0.7	92.4+-1	92.9+-0.6	80.7+-1.9	80.1+-0.9

20	electricity	83.0+-0.2	96.9+-0	97.0+-0	96.3+-0.1	88.2+-0.4	79.8+-0.2
----	-------------	-----------	---------	----------------	-----------	-----------	-----------

Tabla 2. Se muestra los resultados de los modelos con las bases de datos utilizadas.

En un análisis preliminar se puede ver que los resultados del benchmark revelaron que los modelos de caja negra, como Random Forest y XGBoost, sobresalieron en problemas complejos con un gran número de características. Estos modelos han alcanzado valores de AUC-ROC cercanos a 1 en muchos casos. Su capacidad para capturar patrones intrincados en los datos los convierte en herramientas poderosas para la predicción precisa. Sin embargo, su falta de interpretabilidad limita su aplicación en áreas donde la transparencia es esencial.

Por otro lado, los métodos transparentes, como EBM y Regresión Logística, demostraron ser competitivos en todos los dataset y podrían destacar en problemas donde la interpretabilidad es crucial. Naive Bayes y el Árbol de Decisión, aunque más simples que EBM, también encontraron su nicho en problemas como por ejemplo lineales y de menor complejidad, respectivamente. Se ve claramente en el caso del conjunto de datos “kr-vs-kp”, donde logran mucha precisión, aunque casi todos los modelos obtienen un buen rendimiento con esos datos.

No obstante, el análisis debe ser más profundo y minucioso por lo que se va a interpretar y mostrar la información relevante.

Esta gráfica nos ayuda a analizar el rendimiento medio de los modelos para las diferentes bases de datos.

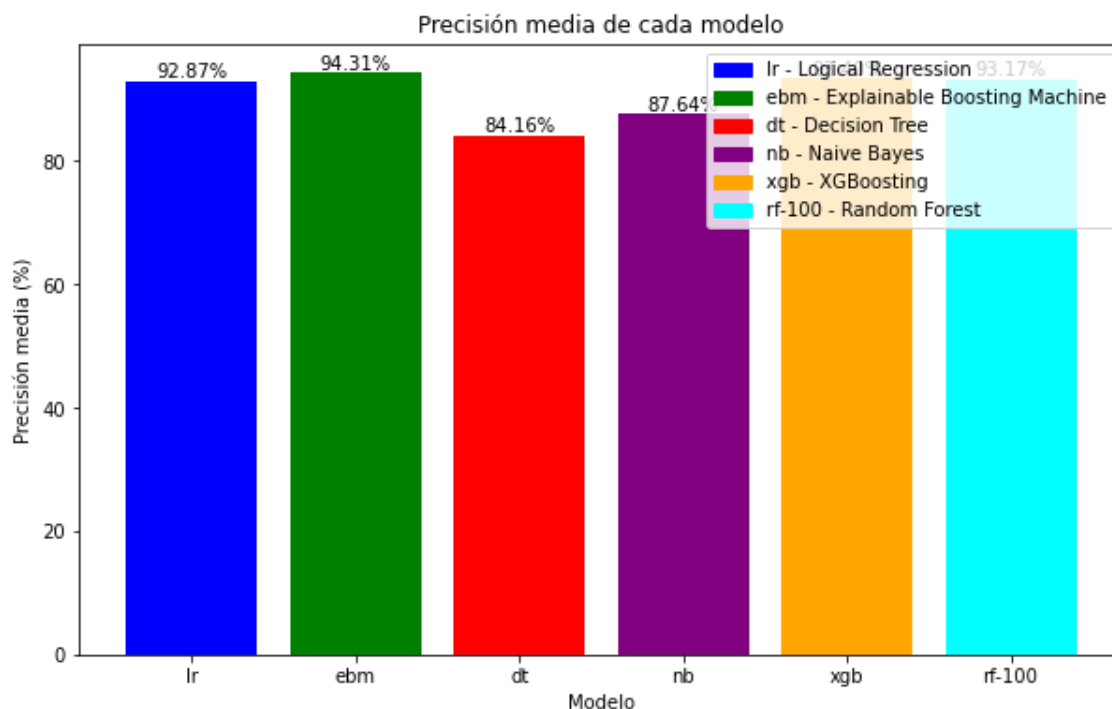


Figura 14. Gráfico de barras que muestra la precisión curva de AUC media de cada modelo.

Como se puede observar en la imagen superior la precisión de clasificación de los diferentes modelos es bastante alta en general. Lo que más destaca es que no hay grandes diferencias entre ellos. Los modelos árbol de decisión y Naive Bayes predicen ligeramente peor respecto al resto, y por otra parte se ve como el modelo EBM es el que ofrece mejor rendimiento.

En la siguiente imagen se va a observar que rendimiento tienen los modelos transparentes respecto a la caja negra.

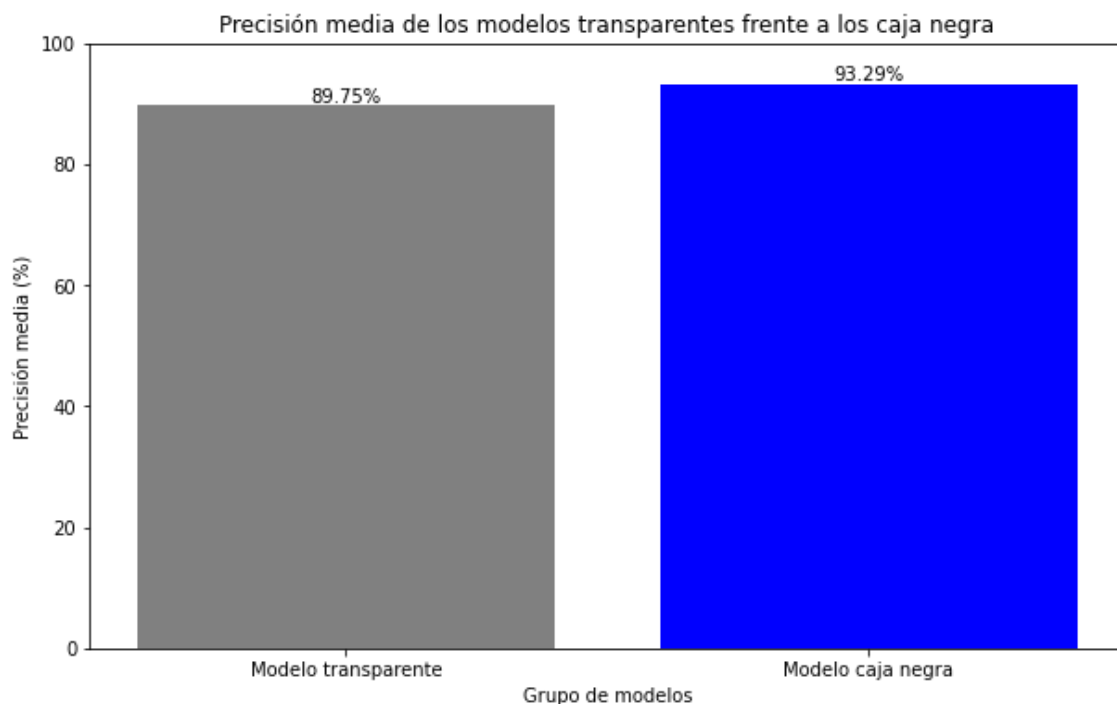


Figura 15. Gráfica de barrar donde compara la precisión media de los modelos transparentes respecto a la caja negra.

Y en esta segunda figura se puede observar como los modelos interpretables y caja negra están muy equiparados en cuanto ver cuál es el mejor modelo y grupo para cada conjunto de datos. Podemos ver como 9/20 datasets tienen como mejor modelo uno transparente, respecto a 11/20 que tienen a uno caja negra.

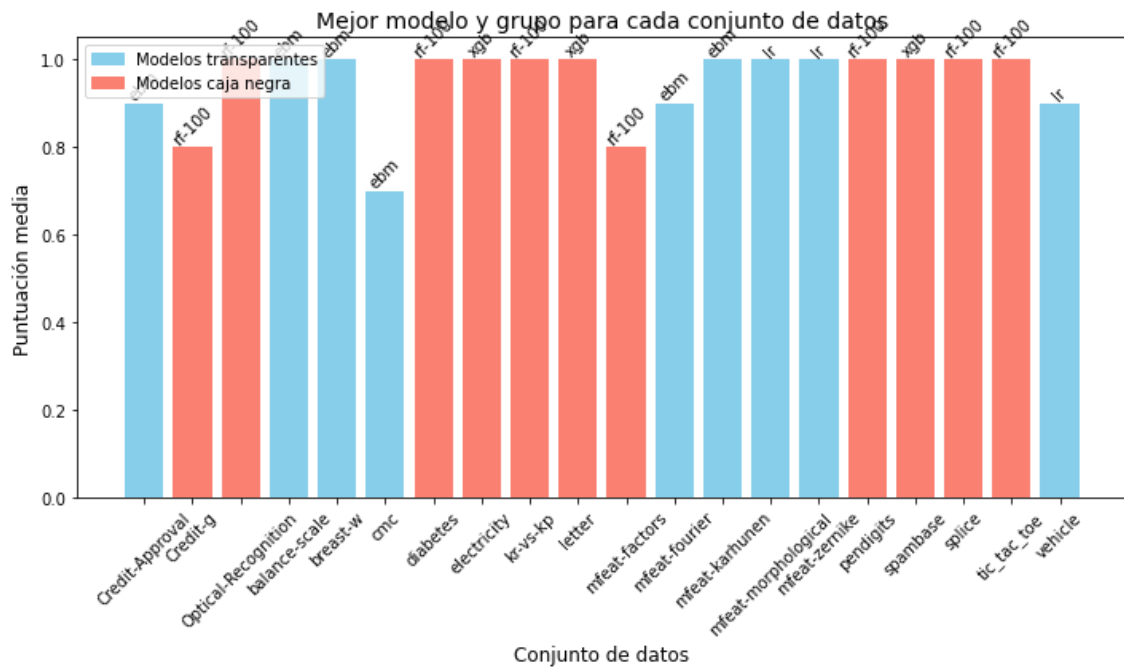
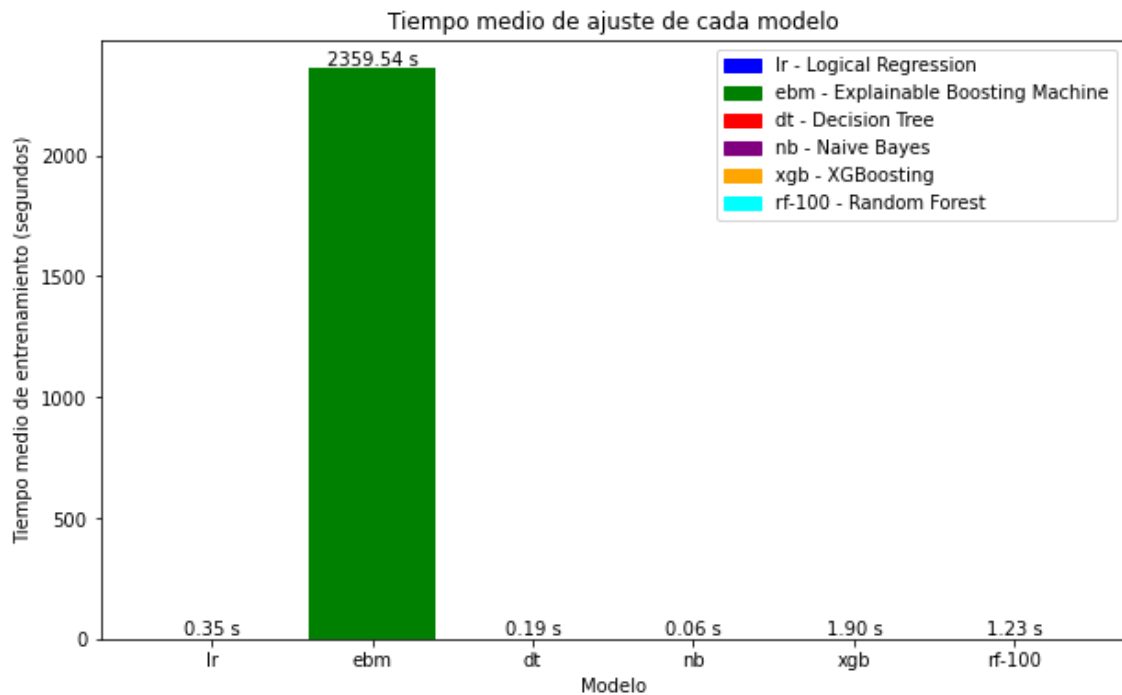


Figura 16. Gráfica de barras que muestra el mejor modelo y grupo para cada conjunto de datos.

Analizando la Figura 17 se puede ver como los tiempos de entrenamiento de los modelos presentan una variabilidad considerable. El EBM destaca por ser extremadamente lento, con un tiempo promedio de entrenamiento de 2359.54 segundos. Este valor es significativamente mayor al del segundo modelo más lento, XGBoost, que promedia 1.90 segundos. De hecho, el EBM es en promedio un asombroso 47619.30% más lento que XGBoost. Los modelos restantes (lr, dt, nb y rf-100) muestran tiempos de entrenamiento mucho más rápidos, siendo el Naive Bayes (nb) el más veloz con tan solo 0.06 segundos en promedio.



*

Figura 17. Gráfica de barras que muestra el tiempo medio que tarda cada modelo en ser entrenado

Esta disparidad en los tiempos de entrenamiento es un factor crucial para considerar al elegir un modelo, especialmente en escenarios donde la velocidad de entrenamiento es un requisito importante. Si bien el EBM ofrece interpretabilidad, su lentitud podría ser un obstáculo. Por otro lado, modelos como Naive Bayes y árbol de decisión ofrecen un equilibrio entre velocidad y rendimiento, siendo opciones atractivas cuando el tiempo de entrenamiento es un factor limitante.

5.2 Análisis de los modelos

En esta sección, profundizaremos en el análisis de los resultados obtenidos en el benchmark de modelos de clasificación. Exploraremos las ventajas y desventajas de cada modelo, considerando su precisión, interpretabilidad y tiempo de entrenamiento. Este análisis nos permitirá comprender mejor las fortalezas y debilidades de cada enfoque y cómo se desempeñan en diferentes escenarios.

Regresión Logística (LR): Este modelo destaca por su sencillez y rapidez en el entrenamiento, lo que lo hace ideal para problemas lineales donde la interpretabilidad es esencial. Aunque su rendimiento es competitivo en la mayoría de los casos, puede verse superado como en problemas no lineales más complejos, como es el caso del conjunto de datos “Electricity”, donde su rendimiento queda muy alejado de los modelos caja negra y EBM. Su velocidad de entrenamiento lo convierte en una opción atractiva para aplicaciones en tiempo real.

Random Forest (RF-100): Random Forest sobresale por su alta precisión en una amplia gama de problemas, siendo robusto frente al sobreajuste y capaz de manejar un gran número de características. Aunque ofrece un excelente rendimiento en general, su interpretabilidad es menor en comparación con otros modelos y puede requerir más recursos computacionales.

XGBoost (XGB): Similar a Random Forest, XGBoost ofrece una alta precisión, especialmente en problemas complejos, y es eficiente en el uso de recursos computacionales. Sin embargo, al igual que Random Forest, su interpretabilidad es limitada y puede requerir un ajuste cuidadoso de los hiperparámetros para obtener los mejores resultados.

Explainable Boosting Machine (EBM): Este modelo destaca por su alta interpretabilidad, proporcionando explicaciones claras de sus predicciones y manteniendo un rendimiento competitivo. Sin embargo, es importante tener en cuenta que, aunque las predicciones son rápidas una vez entrenado, el proceso de entrenamiento en sí mismo puede ser más costoso en términos de tiempo y recursos computacionales en comparación con otros modelos. Esto se debe a la naturaleza iterativa del algoritmo y a la necesidad de ajustar múltiples modelos más simples para lograr un buen rendimiento.

Árbol de Decisión (DT): La principal fortaleza del árbol de decisión es su facilidad de comprensión e interpretación, lo que lo hace útil para problemas simples. Sin embargo, su propensión al sobreajuste y su rendimiento inferior en problemas complejos limitan su aplicabilidad en escenarios más desafiantes.

Naive Bayes (NB): Este modelo destaca por su simplicidad y rapidez en el entrenamiento, siendo sorprendentemente eficaz en algunos problemas con muchas características. A pesar de que su supuesto de independencia condicional puede no ser realista en muchos casos, su velocidad y facilidad de implementación lo convierten en una opción atractiva para prototipado rápido y aplicaciones con recursos limitados.

5.3 Conclusiones

Tras un exhaustivo análisis del rendimiento predictivo de métodos transparentes y de caja negra en una variedad de conjuntos de datos, se ha demostrado que ambos enfoques tienen su lugar en el ámbito del aprendizaje automático. Los modelos de caja negra, como Random Forest y XGBoost, destacan por su capacidad para abordar problemas complejos y ofrecer una alta precisión predictiva. Sin embargo, su opacidad inherente plantea desafíos en términos de interpretabilidad y explicabilidad, lo que puede limitar su aplicabilidad en dominios donde la transparencia es esencial.

Por otro lado, los métodos transparentes, como la Regresión Logística y el EBM, ofrecen un equilibrio entre precisión y explicabilidad. Aunque sean ligeramente menos precisos que los modelos de caja negra en algunos problemas, su capacidad para proporcionar explicaciones claras y comprensibles de sus predicciones los convierte en herramientas valiosas en áreas donde la transparencia es un requisito fundamental, como la medicina, las finanzas y la toma de decisiones críticas.

Gracias a este proyecto se ha llegado a la conclusión de que el pensamiento de que los modelos caja negra siempre son los más precisos es erróneo. Se ha podido comprobar como muchas veces el modelo que mejor predecía era transparente, siendo EBM el que ha posibilita esta mejora de precisión para este grupo de métodos.

Hay que destacar en este trabajo como el EBM se presenta como un modelo disruptivo en el campo de la Inteligencia Artificial explicable. Su capacidad para ofrecer un alto rendimiento predictivo, comparable al de modelos de caja negra como XGBoost, junto con una interpretabilidad excepcional, lo convierte en una herramienta prometedora para abordar problemas complejos donde la transparencia es esencial.

A diferencia de los modelos de caja negra, EBM permite a los usuarios comprender cómo cada característica influye en las predicciones, lo que facilita la toma de decisiones informadas y la detección de posibles sesgos.

Sin embargo, su principal limitación radica en su elevado tiempo de entrenamiento, lo que puede restringir su aplicación en escenarios donde se priorice ahorrar de tiempo y recursos computacionales. A pesar de esta limitación, EBM representa un avance significativo en la búsqueda de modelos de aprendizaje automático que combinen precisión y explicabilidad, abriendo nuevas posibilidades para la aplicación ética y responsable de la Inteligencia Artificial en diversos campos.

Además, la integración del clasificador Naive Bayes en la librería `interpretml` ha demostrado ser un paso prometedor hacia la mejora de la interpretabilidad de este algoritmo ampliamente utilizado. Al proporcionar herramientas para visualizar y comprender el funcionamiento interno de Naive Bayes, se amplía su aplicabilidad en escenarios donde la explicabilidad es un factor determinante.

En conclusión, este estudio ha puesto de manifiesto la importancia de considerar tanto la precisión como la interpretabilidad al seleccionar un modelo de aprendizaje automático. La elección del enfoque adecuado dependerá del contexto específico del problema, la necesidad de transparencia y los requisitos de rendimiento. Los métodos transparentes, aunque no siempre los más precisos, desempeñan un papel fundamental en la construcción de una Inteligencia Artificial más ética, responsable y confiable.

5.4 Trabajo futuro

Este Trabajo de Fin de Grado ha sentado las bases para futuras investigaciones en el campo de la XAI y el rendimiento predictivo de métodos transparentes. A continuación, se presentan algunas líneas de investigación que podrían explorarse en trabajos futuros:

1. **Ampliación del Benchmark:** El benchmark realizado en este Trabajo de Fin de Grado podría ampliarse para incluir una gama aún más diversa de conjuntos de datos y modelos de clasificación. Esto permitiría obtener una visión más completa del rendimiento de los métodos transparentes en diferentes dominios y escenarios, así como identificar áreas donde estos métodos podrían mejorarse.

2. **Exploración de Nuevos Métodos Transparentes:** Se podrían investigar y evaluar nuevos algoritmos de clasificación transparentes que aún no se han incluido en este estudio. Esto podría incluir modelos basados en reglas, modelos lineales generalizados y otros enfoques que ofrezcan un equilibrio entre precisión predictiva e interpretabilidad.
3. **Finalización de la integración de Naive Bayes en interpretml:** El trabajo realizado en este Trabajo de Fin de Grado para integrar Naive Bayes en la librería interpretml podría ampliarse para incluir funcionalidades de visualización más completas y personalizadas. Esto permitiría a los usuarios explorar y comprender mejor las predicciones de los modelos Naive Bayes a través de gráficos y representaciones visuales intuitivas.
4. **Investigación sobre Sesgos y Equidad:** Se podrían llevar a cabo investigaciones para analizar cómo los métodos transparentes pueden ayudar a identificar y mitigar sesgos en los modelos de IA. Esto es especialmente relevante en aplicaciones donde las decisiones automatizadas pueden tener un impacto significativo en la vida de las personas, como en la concesión de préstamos o en el diagnóstico médico.
5. **Combinación de Métodos Transparentes y de Caja Negra:** Se podría explorar la posibilidad de combinar métodos transparentes y de caja negra para aprovechar las fortalezas de ambos enfoques. Por ejemplo, se podrían utilizar modelos de caja negra para realizar predicciones precisas y luego emplear métodos transparentes para explicar las razones detrás de esas predicciones.

6 Análisis de Impacto

En este capítulo, analizaremos el impacto potencial de los resultados obtenidos en este Trabajo de Fin de Grado en diversos contextos, considerando tanto los beneficios esperados como los posibles efectos perjudiciales. Además, relacionaremos nuestro trabajo con los Objetivos de Desarrollo Sostenible (ODS) de la Agenda 2030 y destacaremos las decisiones tomadas a lo largo del proyecto que se basan en la consideración del impacto.

Impacto Personal

A nivel personal, este Trabajo de Fin de Grado ha sido una experiencia enriquecedora, permitiéndome profundizar en el campo de la Inteligencia Artificial Explicable y adquirir conocimientos prácticos sobre la evaluación y comparación de modelos de clasificación. El desarrollo de habilidades técnicas en programación, análisis de datos y evaluación de modelos ha sido significativo. Además, la comprensión de los desafíos éticos y sociales relacionados con la Inteligencia Artificial ha fomentado una mayor conciencia sobre la importancia de la transparencia y la responsabilidad en el desarrollo y uso de esta tecnología.

Impacto Empresarial

En el ámbito empresarial, los resultados de este Trabajo de Fin de Grado pueden tener un impacto positivo al proporcionar a las empresas herramientas y conocimientos para elegir los modelos de Inteligencia Artificial más adecuados para sus necesidades específicas. La evaluación comparativa de métodos transparentes y de caja negra puede ayudar a las empresas a tomar decisiones informadas sobre qué modelos implementar, considerando tanto la precisión predictiva como la necesidad de interpretabilidad y explicabilidad. Esto puede mejorar la eficiencia y eficacia de los procesos empresariales, así como fomentar la confianza de los clientes y las partes interesadas en las soluciones de IA.

Impacto Social

El impacto social de este Trabajo de Fin de Grado radica en su contribución a una Inteligencia Artificial más transparente y ética. Al investigar y promover el uso de métodos transparentes, se busca empoderar a los usuarios y a la sociedad en general para comprender cómo funcionan los modelos de Inteligencia Artificial y cómo se toman las decisiones automatizadas. Esto puede aumentar la confianza pública en la IA, reducir el riesgo de discriminación y sesgos algorítmicos, y fomentar un uso más responsable y equitativo de esta tecnología en beneficio de la sociedad.

Impacto Económico

Desde una perspectiva económica, la adopción de modelos de Inteligencia Artificial transparentes puede generar beneficios significativos para las empresas y la sociedad. Al mejorar la eficiencia y eficacia de los procesos, reducir el riesgo de errores y sesgos, y fomentar la confianza del consumidor, la Inteligencia Artificial transparente puede impulsar la innovación, el crecimiento económico y la creación de empleo en diversos sectores. Además, al abordar los desafíos éticos y sociales de la IA, se pueden evitar costos económicos y reputacionales asociados con el uso irresponsable de esta tecnología.

Impacto Medioambiental

Aunque el impacto medioambiental directo de este Trabajo de Fin de Grado es limitado, es importante destacar que la elección de modelos de Inteligencia

Artificial eficientes en términos de recursos computacionales puede contribuir a la sostenibilidad. Al optimizar el uso de energía y reducir la huella de carbono de los sistemas de IA, se puede minimizar el impacto ambiental de esta tecnología en el largo plazo.

Impacto Cultural

En el ámbito cultural, este Trabajo de Fin de Grado puede generar un impacto al promover un diálogo abierto y transparente sobre el papel de la Inteligencia Artificial en la sociedad. Al destacar la importancia de la explicabilidad y la ética en el desarrollo y uso de la IA, se fomenta una cultura de responsabilidad y conciencia crítica en torno a esta tecnología. Esto puede llevar a una mayor participación ciudadana en la toma de decisiones relacionadas con la Inteligencia Artificial y a una comprensión más profunda de sus implicaciones culturales y sociales.

Relación con los Objetivos de Desarrollo Sostenible (ODS)

Este Trabajo de Fin de Grado se alinea con varios Objetivos de Desarrollo Sostenible (ODS) de la Agenda 2030, en particular:

- **ODS 9: Industria, Innovación e Infraestructura:** Al contribuir al desarrollo de una Inteligencia Artificial más transparente y responsable, se promueve la innovación tecnológica y se fomenta la creación de infraestructuras digitales más equitativas y sostenibles.
- **ODS 10: Reducción de las Desigualdades:** La Inteligencia Artificial transparente puede ayudar a reducir las desigualdades al garantizar que las decisiones automatizadas sean justas, equitativas y no discriminatorias.
- **ODS 16: Paz, Justicia e Instituciones Sólidas:** Al fomentar la transparencia y la responsabilidad en el uso de la IA, se contribuye a la construcción de instituciones más sólidas y justas que protejan los derechos humanos y promuevan la paz.

Decisiones Basadas en la Consideración del Impacto

A lo largo del proyecto, se han tomado diversas decisiones teniendo en cuenta su impacto potencial. Por ejemplo, la elección de la plataforma OpenML para acceder a conjuntos de datos abiertos y diversos refleja un compromiso con la transparencia y la colaboración en la investigación. Además, la selección de métricas de evaluación como el área bajo la curva ROC (AUC-ROC) se basa en su capacidad para proporcionar una evaluación robusta y equitativa del rendimiento de los modelos, incluso en situaciones con clases desbalanceadas.

7 Referencias

- [1] J. M. Delgado, «Computer Hoy Todo el mundo habla de ella, pero: ¿cómo funciona la inteligencia artificial?,» *Computer Hoy*, 6 Junio 2024. [En línea]. Available: <https://computerhoy.com/tecnologia/todo-mundo-habla-ella-pero-como-funciona-inteligencia-artificial-1388779>. [Último acceso: 14 Junio 2024].
- [2] C. L. Attila Benko, «History of artificial intelligence,» *Encyclopedia of Information Science and Technology*, vol. Second Edition, 2009.
- [3] Ku. Chhaya, A. Khanzode, and D. Sarode «Advantages And Disadvantages Of Artificial Intelligence,» *Aayushi International Interdisciplinary Research Journal*, pp. 227-230, 2020.
- [4] Q. Bi, K. E. Goodman, J. Kaminsky and J. Lessler, «“What is Machine Learning? A Primer for the Epidemiologist,» *American Journal of Epidemiology*, vol. 188, n° 12, 2019.
- [5] A. Ra, «Explainable AI: from black box to glass box,» *Journal of the Academy of Marketing Science*, p. 137–141, 2020.
- [6] A. Ortega, «Hacia un régimen europeo de control de la Inteligencia,» *Real Instituto Elcano*, 2021.
- [7] C. Molnar, *Interpretable Machine Learning A Guide for Making Black Box Models Explainable*, Lulu, 2020.
- [8] «Grand View Research,» Next Generation Technologies, 2021. [En línea]. Available: <https://www.grandviewresearch.com/industry-analysis/explainable-ai-market-report#>. [Último acceso: 14 bril 2024].
- [9] F. Giannotti, R. Guidotti, A. Monreale, S. Ruggieri, et al ,«Meaningful Explanations of Black Box AI Decision Systems,» *University of Pisa*, vol. 33, n° 01, pp. 9780-9784, 2019.
- [10] A. k. Muzahidul Islam, A. K. M. Muzahidul, S. Islam, S. Shatabda et al «"Symptom Based Explainable Artificial Intelligence Model for Leukemia Detection,» *Institute of Advanced Research (IAR) & United International University (UIU)*, vol. 10, pp. 57283-57298, 2022. ,
- [11] H. Nori, S. Jenkins, P. Koch, and R. Caruana, «InterpretML: A Unified Framework for Machine Learning,» *InterpretML - Draft*, 2019.
- [12] A. Barredo Arrieta, , N. Díaz-Rodríguez , J. Del Ser , A. Bennetot , et al «Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI,» *Information Fusion*, vol. 58, n° 1566-2535, pp. 82-115, 2020.
- [13] B. Mahesh, «Machine Learning Algorithms - A Review,» *International Journal of Science and Research (IJSR)*, 2018.
- [14] M. Oleszak, «Explainable Boosting Machines,» Medium, 23 Enero 2023. [En línea]. Available: <https://pub.towardsai.net/explainable-boosting-machines-c71b207231b5>. [Último acceso: 19 Mayo 2024].

- [15] H. Zhang, «The Optimality of Naive Bayes,» *Faculty of Computer Science University of New Brunswick*, 2004.
- [16] S. Sperandei, «Understanding logistic regression analysis,» *School of Physical Education and Sports – Federal University of Rio de Janeiro*, vol. 24, n° 1, pp. 12-18, 2014.
- [17] S. J. Rigatti, «Random Forest,» *JOURNAL OF INSURANCE MEDICINE*, vol. 47, p. 31–39, 2017.
- [18] W. Koehrsen, «Random Forest Simple Explanation,» Medium, 27 Diciembre 2017. [En línea]. Available: <https://williamkoehrsen.medium.com/random-forest-simple-explanation-377895a60d2d>. [Último acceso: 25 Mayo 2024].
- [19] Y. Wang, X. Ge, B. Wang, C.-C. J Kuo, «KGBBoost: A classification-based knowledge base completion method,» *Pattern Recognition Letters*, 2022.
- [20] D. Leventis, «XGBoost Mathematics Explained,» Medium, 18 Noviembre 2018. [En línea]. Available: <https://dimleve.medium.com/xgboost-mathematics-explained-58262530904a>. [Último acceso: 11 Mayo 2024].
- [21] «OpenML A worldwide machine learning lab,» OpenML, [En línea]. Available: <https://www.openml.org/>. [Último acceso: 16 Junio 2024].
- [22] S. Narkhede, «Understanding AUC - ROC Curve,» *Towards data science*, vol. 26, n° 1, pp. 220-227, 2018.
- [23] «Linear Model,» Interpretml, [En línea]. Available: <https://interpret.ml/docs/lr.html>. [Último acceso: 31 Mayo 2024].

8 Anexos

Repositorio del Código Fuente

Todo el código fuente desarrollado para este Trabajo de Fin de Grado se encuentra disponible en el siguiente repositorio de GitHub:

[Tfg-interpretml - <https://github.com/bmihaljevic/tfm-interpretml>)

El repositorio contiene:

- Código fuente del desarrollo del benchmark, en él se muestran las diferentes versiones.
- El código generado para integrar con la librería interpretml
- Las imágenes y el código de las gráficas que se muestran tanto en la parte de benchmark como la de integración.