TRABAJO FIN DE MÁSTER

Máster en Ciencia de Datos e Ingeniería de Datos en la Nube

WiDS Datathon 2024 - Challenge 2: Modelos de regresión para estimación del periodo de diagnóstico metastático

Autor: Luna Jiménez Fernández

Tutor: Juan Carlos Alfaro Jiménez

Junio, 2025

Dedicado a la gente que, pese a todo,
sigue persiguiendo sus sueños.
Nunca os rindáis.

Declaración de autoría

Yo, **Luna Jiménez Fernández**, con DNI **47092045M**, declaro que soy la única autora del Trabajo Fin de Master titulado ***"WiDS Datathon 2024 - Challenge 2: Modelos de regresión para estimación del periodo de diagnóstico metastático"***, que el citado trabajo no infringe las leyes en vigor sobre propiedad intelectual, y que todo el material no original contenido en dicho trabajo está apropiadamente atribuido a sus legítimos autores.

Albacete, a **10** de **Junio** de **2025**

Fdo.: **Luna Jiménez Fernández**

Resumen

El acceso equitativo a una **atención sanitaria de calidad** es un problema de interés a nivel global - existiendo diferencias significativas tanto en **calidad** como en **posibilidad** de acceso entre poblaciones. Este problema afecta especialmente al **cáncer de mama triple negativo** y sus metástasis - unos de los cánceres más difíciles de tratar, y en los que más pueden influir dichos sesgos retrasando su diagnóstico y tratamiento.

Por tanto, el objetivo de este trabajo consiste en **desarrollar un modelo de regresión** capaz de predecir el tiempo de diagnóstico de metástasis - incidiendo a la vez en la relevancia de los factores **geográficos, socioeconómicos y climáticos** - a través de un proceso de **ciencia de datos**.

Se ha realizado una **revisión de las principales técnicas** utilizadas durante el trabajo: el ciclo de vida de la ciencia de datos y algunos de los principales modelos de aprendizaje automático de regresión, con especial foco en los *ensembles* de *Gradient Boosting*.

Además, se ha realizado un **análisis exploratorio del conjunto de datos**, observando la gran influencia de los códigos de diagnóstico a la vez que la irrelevancia de los atributos numéricos - representativos de factores geográficos, socioeconómicos y climáticos. Esta información se ha podido utilizar para realizar un **pre-procesamiento** de los datos, proponiendo tres subconjuntos de atributos a estudiar y varias transformaciones para imputar valores perdidos.

Para el desarrollo del modelo, se ha realizado una fase de **modelado** en la que se han propuesto 16 modelos de regresión - realizando sobre éstos un ajuste de hiperparámetros y una selección de modelo en base al error. Además, el modelo seleccionado ha sido **evaluado** - estudiando su rendimiento tanto durante las fases de ajuste de hiperparámetros como de selección de modelos, y contrastando su rendimiento con un equivalente, ganador de la competición en la que se ha basado este problema.

Finalmente, se ha **desplegado** el modelo a través de una aplicación web para permitir su uso de forma pública.

Abstract

Equitative access to **quality healthcare** is a problem of global interest - with significant differences in both **quality** and **availability** existing depending on the population. This is especially true for **triple negative breast cancer** and its metastasis - one of the most dangerous and difficult to treat cancers, and one in which negative biases can greatly influence by delaying its diagnosis and treatment.

The goal of this work is **developing a regression model** capable of predicting the metastasis diagnosis period of a patient - with a special focus on studying the relevance of **geographic, socioeconomic and climatic** factors - via data science.

A survey of the **state of the art** of the main techniques used during this workk has been performed: studying the data science life cycle and some of the main regression machine learning models, focusing on *Gradient Boosting ensembles*.

In addition, an **exploratory data analysis** has been done on the dataset, finding that diagnosis codes carry a significant influence on the diagnosis period, while most numerical attributes - representing geographic, socioeconomic and climatic attributes - are mostly unrelated to the prediction. This knowledge has been used to **pre-process** the data, proposing three feature subsets to study and several transformations, including missing value imputing.

For the model development, a **modelling** phase has been done, in which 16 regression models have been proposed - performing hyperparameter optimization and model selection on them based on their error. In addition, the chosen model has been **evaluated** - studying its performance in both previous phases, and comparing it with a similar model that placed first on the competition that originally proposed this problem.

Finally, a web application has been **deployed** - to allow users to access the developed model publically.

Agradecimientos

En primer lugar, quiero agradecer a todos mis compañeros y amigos del grupo de **Sistemas Informáticos y Minería de Datos (SIMD)** - y, especialmente, a mi amigo y director **Juan Carlos Alfaro Jiménez** - por su apoyo, recursos y consejos durante la realización de este trabajo. Aunque ya no sea formalmente parte de este grupo, siempre me sentiré vinculada a él.

Además, quiero a gradecer a mis amigos y familia del **Curso de Comic Online de la Escola Joso** - **Arai, Aina, Arkaitz, Clara, Irene, Martín, Pau, Rafi...** -, con los que compartí un proyecto de gran importancia personal, mi primer comic publicado, y en los que he encontrado un grupo al que pertenecer. Muchas gracias por todo.

Finalmente, quiero agradecer a **mi familia y seres queridos** - tanto los que me acompañan presencialmente como los que se encuentran a distancia. Vuestro apoyo y cariño continuo me ha ayudado a seguir adelante y acabar este trabajo a pesar de todas las dificultades.

Índice general

Índice de figuras

Índice de tablas

1. Introducción

El acceso equitativo a una **atención sanitaria de calidad** es un problema de gran interés a nivel global, existiendo desigualdades sustanciales en la **calidad y acceso** a dicho servicio entre distintas poblaciones. Estos problemas, además, se pueden llegar a exacerbar por distintos factores: geográficos, socioeconómicos y climáticos.

Con el fin de estudiar la influencia de dichos factores en la atención sanitaria, la iniciativa *Women in Data Science* propuso en el año 2024 una **competición** [1] con el objetivo de **estimar el tiempo necesario para realizar un diagnóstico de metástasis para cáncer de mama** a partir de un conjunto de datos médico ampliado con información geográfica, socioeconómica y climática — y, a su vez, estudiar como dichos factores pueden influir al tiempo necesario para realizar un diagnóstico —.

Por tanto, la meta de este trabajo es la creación de **modelos de regresión** capaces de estimar dicho tiempo de diagnóstico con el menor error posible - utilizando, para ello, el proceso completo de **ciencia de datos**.

1.1. Objetivos

El principal objetivo de este trabajo es el **desarrollo de un modelo de regresión** capaz de resolver el problema propuesto por la competición: la predicción del tiempo necesario para realizar un diagnóstico de metástasis de cáncer de mama, evaluando su rendimiento y dejando disponible el modelo para ser accesible por los hipotéticos usuarios finales.

Para alcanzar dicho objetivo, es necesario llevar a cabo los siguientes pasos, siguiendo el **ciclo de vida de la ciencia de datos**:

1. Análisis exploratorio de los datos disponibles en la competición, para comprender su comportamiento y características.
2. Pre-procesamiento de los datos para la propuesta de subconjuntos de atributos reducidos y preparación posterior para el uso con modelos.
3. Estudio, selección y caracterización de los modelos y sus hiperparámetros a estudiar durante el proceso.

4. Experimentación y estudio de los resultados para seleccionar un modelo definitivo a ser utilizado.
5. Creación de una aplicación web para desplegar el modelo final entrenado, con el fin de ser utilizado por expertos en el campo de la medicina sin experiencia previa en ciencia de datos.

A su vez, este trabajo aborda el segundo objetivo planteado por la propia competición: el **estudio de la influencia de los factores geográficos, socioeconómicos y climáticos** en la calidad de la atención sanitaria.

1.2. Estructura de la memoria

La memoria está dividida en un total de **7** capítulos, como se describen a continuación:

- **Capítulo 1:** En este capítulo se introduce el problema a resolver, los objetivos que se busca cumplir con el trabajo y la estructura general de la memoria.
- **Capítulo 2:** En este capítulo se realiza una breve revisión de las principales técnicas a utilizar durante la memoria: tanto el proceso de ciencia de datos y sus etapas como los modelos a utilizar durante la experimentación - desde los modelos simples como las regresiones lineales y los árboles de decisiones hasta los *ensembles* de modelos simples.
- **Capítulo 3:** En este capítulo se realiza un estudio más exhaustivo del problema: tanto su definición como un análisis exploratorio de los datos disponibles, estudiando el comportamiento de la variable objetivo y la relevancia y correlación de los atributos respecto al tiempo de diagnóstico.
- **Capítulo 4:** En este capítulo se introduce el pre-procesamiento a realizar sobre el conjunto de datos, obteniendo varios subconjuntos de atributos reducidos a ser estudiado posteriormente y preparando *pipelines* automáticos para realizar todas las transformaciones necesarias para el uso de los datos por parte de los modelos.
- **Capítulo 5:** En este capítulo se detalla la experimentación a realizar. Se proponen varios modelos sobre los que se realizará un proceso de ajuste de hiperparámetros y selección de modelos, con el fin de obtener un modelo definitivo a ser utilizado para resolver el problema. Además, se presentan y estudian los resultados de dicha experimentación.
- **Capítulo 6:** En este capítulo se presenta una aplicación web a través de la cual se hace disponible a los usuarios expertos el modelo obtenido en el capítulo anterior - detallando la interfaz gráfica y las distintas funcionalidades ofrecidas.
- **Capítulo 7:** Finalmente, en este capítulo se muestran las conclusiones alcanzadas tras el desarrollo del trabajo, proponiendo posibles lineas de trabajo futuro para ampliarlo.

2. Revisión de técnicas

En este capítulo se describen los procesos y algoritmos utilizados a lo largo del trabajo descrito en esta memoria. Concretamente, se comienza explicando el concepto de la **ciencia de datos** y su ciclo de vida, haciendo énfasis en **CRISP-DM** como metodología utilizada a lo largo del proyecto para resolver el problema propuesto. Tras esto, se estudian conceptos de aprendizaje automático como la **selección de modelos** o los **modelos de regresión** - haciendo especial énfasis en los modelos de *ensemble* basados en técnicas de ***Gradient Boosting***.

2.1. Ciencia de datos y el ciclo de vida de los datos

2.1.1. Ciencia de datos

La **ciencia de datos** es el estudio de la extracción de conocimiento útil a partir de datos, y de la generalización de dicho proceso a cualquier problema [2]. Dicho proceso incluye la recolección y almacenamiento, mantenimiento, procesamiento, análisis y visualización de enormes cantidades de datos heterogéneos - asociados a un gran abanico de aplicaciones y dominios en muchas ocasiones multidisciplinarios [3].

Desde su origen, la ciencia de datos ha evolucionado como un campo interdisciplinar que integra conocimientos y técnicas de otras disciplinas afines como el análisis de datos, la estadística o la minería de datos [4]. Ahora bien, la principal diferencia con estos campos se encuentra en el fin: el aprendizaje a partir de los datos [2] y la capacidad de adquirir nuevo conocimiento capaz de ser utilizado para la toma de decisiones y la predicción [3].

Por definición, la ciencia de datos depende de los datos sobre los que se está trabajando. Por esto, el proceso de trabajo de la ciencia de datos depende generalmente del **ciclo de vida de los datos**: las distintas etapas por las que pasa un conjunto de datos desde su recolección e investigación hasta su uso final [5]. Como se observa en la **Figura 2.1**, este ciclo está tradicionalmente dividido en **cinco** apartados [4]:

1. **Adquisición:** En la actualidad, los datos se generan en cantidades masivas - del orden de **exabytes por hora** [6]. Por tanto, el primer paso del ciclo consiste en la adquisición y almacenamiento eficiente de los datos necesarios para el proceso.

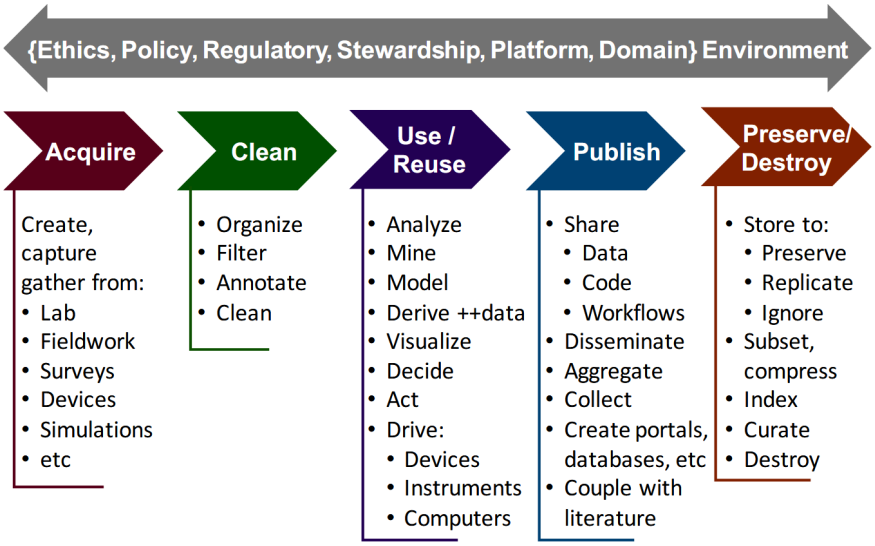


Figura 2.1: Ciclo de vida de los datos [4]

2. **Limpieza:** Tras la adquisición, el segundo paso del ciclo consiste en la transformación de los datos originales en datos utilizables posteriormente - a través de procesos de limpieza, imputación, formateo, etc.
3. **Uso y re-uso:** El tercer paso del ciclo consiste en el uso de los datos procesados con el fin de adquirir conocimiento y tomar decisiones a partir de estos. este apartado se puede dividir, a su vez, en tres subapartados [6]:

   (a) **Análisis exploratorio:** El estudio del comportamiento de los datos con el fin de plantear hipótesis para guiar el resto del ciclo de datos [7].

   (b) **Modelado:** El uso de técnicas computacionales y estadísticas para extraer conocimiento y predicciones a partir del conjunto de datos.

   (c) **Visualización, interpretación y actuación:** La representación gráfica de los resultados del uso de los datos, con el fin de facilitar la toma de decisiones posterior a las personas.

4. **Publicación:** El cuarto paso del ciclo consiste en la diseminación de los resultados del proceso - con el fin de que el conocimiento creado pueda ser conocido y reutilizado por el mayor número de personas posible.
5. **Preservación o destrucción:** El quinto y último paso del ciclo consiste en la preservación o destrucción de los datos utilizados - cumpliendo con otros factores como pueden ser las consideraciones éticas o regulatorias.

Con el fin de regularizar, estandarizar y hacer reproducible el proceso completo de la ciencia de datos - desde la adquisición de los conjuntos de datos hasta la distribución de los resultados -, se han propuesto varias ampliaciones y adaptaciones del ciclo de datos estudiado, conocidas como **ciclos de vida de la ciencia de datos** [5].

Aunque actualmente no existe un ciclo estandarizado, uno de los procesos más utilizados para ciencia de datos es el **Cross-Industry Standard Process for Data Mining (*CRISP-DM*)**, propuesto originalmente para el campo de la minería de datos pero adaptado a las necesidades de la ciencia de datos [8] - siendo el proceso utilizado a lo largo del trabajo descrito en esta memoria.

2.1.2. Cross-Industry Standard Process for Data Mining - CRISP-DM

Cross-Industry Standard Process for Data Mining (abreviado como *CRISP-DM*) es una metodología desarrollada con el fin de ofrecer un proceso de trabajo completo de principio a fin para la minería de datos; independientemente del campo, las herramientas o la aplicación final de los datos [8]. Si bien fue propuesto originalmente en el año 2000, en la actualidad sigue siendo uno de los procesos más utilizados tanto en minería de datos como en ciencia de datos [9].

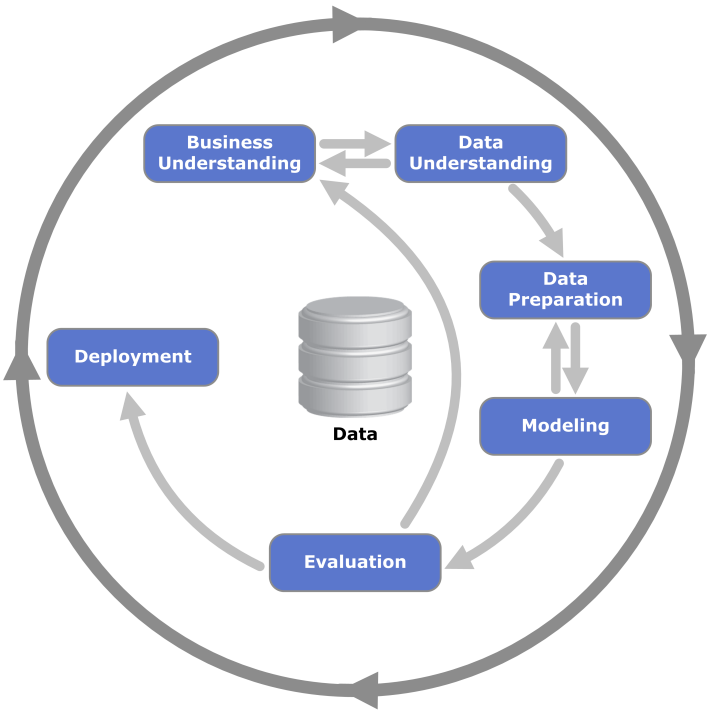


Figura 2.2: Ciclo de vida de CRISP-DM [8]

Como se observa en la **Figura 2.2**, el ciclo de CRISP-DM está dividido en **seis fases** [8], similares al ciclo de datos estudiado:

1. **Conocimiento del campo (*Business Understanding*):** El primer paso del ciclo consiste en entender el problema y los objetivos a resolver - estudiando la situación actual y estableciendo los pasos para alcanzar las metas propuestas.

2. **Conocimiento de los datos (*Data Understanding*):** El segundo paso del ciclo consiste en adquirir y estudiar los datos - tanto de forma superficial como en un análisis exploratorio más profundo -, además de verificar que los datos disponibles son útiles para los objetivos propuestos.
3. **Preparación de los datos (*Data Preparation*):** Tras la adquisición de conocimiento, el tercer paso consiste en preparar los datos obtenidos para su uso posterior - seleccionando las instancias relevantes, limpiando los datos para eliminar valores perdidos, enriqueciendo los datos con información externa, etc.
4. **Modelado (*Modeling*):** Con los datos preparados, la cuarta fase del ciclo consiste en el uso y calibración de modelos de aprendizaje automático a aplicar sobre los datos - definiendo los estudios y experimentos a realizar sobre los modelos, y evaluando el rendimiento final de estos.
5. **Evaluación (*Evaluation*):** Antes de desplegar el modelo final, la quinta fase del ciclo consiste en evaluar si los resultados obtenidos satisfacen los objetivos propuestos y si el proceso de ciencia de datos se ha aplicado de forma adecuada.
6. **Despliegue (*Deployment*):** La última fase del ciclo es el despliegue y diseminación de los resultados obtenidos - haciendo disponible el modelo y los resultados a los usuarios finales.

Es importante destacar que, como indican las flechas de la **Figura 2.2**, la metodología propuesta no es linear, sino que el flujo entre los distintos pasos se puede ver alterado:

- Las fases tienen dependencias entre sí - los descubrimientos en algunas fases pueden producir que sea necesario volver a fases anteriores para perfeccionar el proceso.
- El proceso es **cíclico** - los conocimientos adquiridos durante las distintas fases se aplican para refinar futuros procesos, ya sean sobre el mismo conjunto de datos o datos nuevos.

2.2. Aprendizaje automático y ajuste de modelos

2.2.1. Aprendizaje automático

El **aprendizaje automático** (también conocido en inglés como *Machine Learning*) es una rama de la inteligencia artificial que consiste en la creación de programas capaces de **aprender** - es decir, de mejorar su rendimiento en una tarea - a través de la experiencia y de la información que se les aporta [10].

También se puede definir el término como el conjunto de métodos capaces de detectar patrones en los datos de forma autónoma, y de utilizar dichos patrones para predecir datos futuros [11] - siendo este último de mayor interés a los principios de la ciencia de datos.

Generalmente, los algoritmos de aprendizaje automático se dividen en dos grandes familias, en función del tipo de datos e información que se aporta a los algoritmos [11], [12]:

- **Aprendizaje supervisado:** El objetivo del algoritmo es aprender una función capaz de, dados unos datos de entrada X, predecir una salida Y. Esta función se aprende a partir de un conjunto de datos $D = (x_i, y_i)_{i=1}^{N}$ donde a cada instancia x_i del conjunto de datos D se le asocia un valor esperado y_i.

  Este tipo de aprendizaje se puede dividir a su vez en dos categorías dependiendo del tipo de salida Y que se espera [12]:

  - **Clasificación**: El algoritmo busca obtener para cada entrada x_i un valor concreto **dentro de un conjunto finito de posibles valores**.
  - **Regresión**: El algoritmo busca obtener, para cada entrada x_i, un **valor numérico continuo**.

- **Aprendizaje no supervisado:** El objetivo del algoritmo es aprender patrones subyacentes de los datos de entrada X ofrecidos, sin buscar predecir una salida. Esta función se aprende a partir de un conjunto de datos $D = x_{i}{}_{i=1}^{N}$ donde no se ofrece ningún tipo de etiqueta a cada instancia x_i.

Un **modelo** es el resultado del proceso de aprendizaje automático: una función capaz de predecir una salida para una entrada dada, y cuyos parámetros e hiperparámetros - parámetros que ajustan el aprendizaje del modelo - han sido ajustados a través de un entrenamiento sobre un conjunto de datos para **minimizar un error** [13].

De cara a cumplir el objetivo propuesto por el trabajo descrito en esta memoria - la creación de un modelo capaz de **predecir el tiempo de diagnóstico** -, se van a trabajar con modelos supervisados de **regresión**. Por esto, resulta de interés describir los principales modelos a utilizar y el ajuste que se va a realizar sobre ellos.

2.2.2. Selección de modelos

Durante el entrenamiento de los modelos, se pueden encontrar algunos problemas:

- Se han entrenado varios modelos, y es necesario elegir de forma objetiva uno de ellos en base a su rendimiento.
- Se necesitan elegir los hiperparámetros de un modelo que mejor rendimiento ofrecen sobre el conjunto de datos, de forma objetiva.
- Los modelos entrenados han aprendido variaciones insignificantes y patrones falsos - ruido interpretado como información real - a partir de los datos de entrada, llevando a un problema de **sobreajuste** [11] que puede afectar de forma negativa al rendimiento del modelo.

En estos casos, nos interesa seleccionar de entre todos los modelos a evaluar el modelo más **generalizable** - es decir, el modelo que tendría el **menor error esperado** si se evaluase con un conjunto de datos distinto al utilizado durante el entrenamiento [11].

Ahora bien, a la hora de la verdad no es común tener acceso a dicho hipotético conjunto de test - o si se tiene, solo debería ser utilizado para evaluar el rendimiento del modelo seleccionado finalmente para evitar que se sesgue la selección de modelos [12].

Para solucionar este problema y realizar una selección de modelos **honesta**, se pueden utilizar las siguientes opciones [11]:

- **Conjunto de validación:** Se particiona el conjunto de datos inicial en entrenamiento y validación - utilizando el primero para entrenar los modelos, y el segundo para evaluar su rendimiento honesto y seleccionar el modelo.
- k**-Validación cruzada:** En caso de que el conjunto de datos no sea suficientemente grande para particionarse, se puede optar por particionar el conjunto de datos en k **trozos** de igual tamaño. Para cada uno de estas particiones, se entrenan los modelos con el resto de particiones y se evalúan los rendimientos sobre la partición seleccionada. Tras realizar este proceso k veces, se puede utilizar el error promedio de los k entrenamientos como una aproximación al rendimiento honesto del modelo para realizar la selección.

2.2.3. Modelos de regresión

En el caso de la **regresión**, el objetivo del modelo es aprender una función capaz de predecir un valor numérico continuo para cada instancia del conjunto de datos de entrada [11]. Dicha función se ajusta buscando encontrar el conjunto de parámetros que **minimiza** la diferencia entre los valores predichos por la función y los valores reales asociados a los datos de entrada [13].

Se han propuesto y estudiado un gran número de modelos de regresión, con parametrizaciones y funcionamientos diversos, en la bibliografía [14]. Pese a esta variedad, es posible dividir todas estas familias de modelos en dos grandes grupos: modelos **tradicionales** y modelos de **conjuntos o *ensembles*** [12]

Modelos tradicionales: regresión lineal, árboles de decisiones y máquinas de vectores de soporte

Si bien no hay una definición consensuada sobre su definición, se puede entender como **modelo tradicional** a un modelo que entrena una única función con el fin de realizar predicciones sobre la salida esperada para cada entrada de datos [12].

Existe una gran cantidad de familias de modelos con una larga trayectoria en la bibliografía existente [13]. Ahora bien, el estudio realizado en la memoria se centra en las siguientes tres familias de modelos utilizadas en el trabajo:

- **Regresión lineal:**

  Los modelos más simples, trabajando con la suposición de que **existe una correlación lineal** entre los atributos de entrada y la salida del modelo [11]. Por lo general, la salida y para una entrada x se predice utilizando la siguiente fórmula:

$$y(x) = \sum_{j=1}^{D} w_j x_j$$

Donde x_j representa cada atributo de la entrada, w_j el peso asignado a cada atributo y D el conjunto de datos; siendo el objetivo de estos modelos ajustar los pesos asignados a cada atributo para minimizar el error cuadrado [12]. Ahora bien, cuando se trabaja con conjuntos de datos de gran dimensionalidad, el gran número de atributos puede afectar de forma negativa al rendimiento del modelo, causando un sobreajuste al conjunto de entrenamiento [15].

Para evitar este problema, se proponen técnicas de **regularización** - penalizaciones aplicadas a la fórmula del error con el objetivo de conseguir modelos menos complejos y más generalizables [16]. Los tres modelos de regularización más utilizados son los siguientes:

- **Ridge (L2) [17]:** Como factor de penalización, se utiliza $\sum_{j=1}^{D}(w_j)^2$ - la suma de los pesos cuadrados del modelo, buscando reducir de forma generalizada la influencia de los atributos para evitar sobreajustes y correlaciones.
- **Lasso (L1) [18]:** Como factor de penalización, se utiliza $\sum_{j=1}^{D}|w_j|$ - la suma del valor absoluto de los pesos del modelo, buscando eliminar los atributos irrelevantes reduciendo su peso a 0.
- **Elastic-Net [16]:** Como factor de penalización, se utiliza $\lambda\left(\sum_{j=1}^{D}(w_j)^2\right) + (1-\lambda)\left(\sum_{j=1}^{D}|w_j|\right)$ - utilizando a la vez las regularizaciones L1 y L2 de forma ponderada, buscando aunar los beneficios de ambas aproximaciones.

- **Máquinas de vectores de soporte (SVM):**

  Las máquinas de vectores de soporte se pueden entender como una evolución de los modelos de regresión lineal donde, en vez de buscar la linea que mejor se ajusta al conjunto de datos, se busca el **hiperplano** capaz de ajustarse al conjunto de datos con el **mayor margen** [12].

  En regresión, esto se traduce en la búsqueda de la función representando al mejor hiperplano capaz de ajustarse a todas las instancias del conjunto de datos a la vez que es capaz de mantener una distancia inferior a un margen ϵ con todos los puntos [19].

  La principal utilidad de estos modelos radica en las dos siguientes características [12]:

  - **Funciones *kernel*:** Un problema de los modelos lineales es que los conjuntos de datos no siempre son linealmente separables. Para solventar este problema, las máquinas de vectores de soporte son capaces de utilizar **funciones *kernel*** para transformar los datos a una mayor dimensionalidad - donde si es posible ajustar un hiperplano con mayor margen.
  - **Vectores de soporte:** Para definir el modelo no es necesario almacenar información sobre el conjunto de datos completo, sino que es suficiente con almacenar información sobre los **puntos que definen la frontera entre el hiperplano y el margen** - conocidos como los vectores de soporte.

- **Árboles de decisión:**

  Los árboles de decisión son modelos de reglas representando su función a través de grafos dirigidos acíclicos [13] donde la predicción se obtiene realizando una serie de comprobaciones secuenciales empezando desde la raíz, ramificando hasta llegar a una hoja final [12]. Estos árboles se dividen en los siguientes componentes:

  - **Nodos:** Nodos internos del árbol donde se realiza una comprobación sobre el valor de un atributo. Dependiendo del resultado de la comprobación, el nodo se **ramifica** a otros nodos u hojas.
  - **Hojas:** El valor final predicho para una entrada, alcanzado tras una serie de comprobaciones en nodos.

  El objetivo del modelo es, por tanto, aprender el conjunto de reglas que minimiza el error del modelo para el conjunto de datos dado. Ahora bien, estos modelos tienden a **sobreajustar** creando árboles de gran profundidad [12].

Modelos de ensemble: *bagging* y *boosting*

Como contraste a los modelos tradicionales, un **modelo de conjunto, meta-modelo o *ensemble*** es un modelo que, durante su entrenamiento, ha aprendido un **conjunto de funciones o modelos sencillos** - por lo general, una agrupación de modelos tradicionales -, agrupando las predicciones de todos éstos para obtener una predicción general de la salida esperada para cada entrada de datos [12].

Los algoritmos de *ensemble* buscan aprender un gran número de modelos simples con rendimiento ligeramente mejor que un modelo aleatorio - conocidos como **modelos de aprendizaje débil**, generalmente **árboles de decisión** [13]. Suponiendo que cada uno de estos modelos es completamente independiente al resto, la unión de sus resultados lleva a una predicción final **más precisa y generalizable** que un único modelo entrenado [13].

Dependiendo de la metodología utilizada para entrenar los modelos simples - ya sea de forma secuencial o paralela -, se pueden dividir los algoritmos en dos familias [12]:

- ***Bootstrap Aggregating* (*Bagging*):**

  Los modelos de *ensemble* trabajan con la suposición de que cada uno de los modelos individuales que lo componen es totalmente independiente al resto de modelos. Ahora bien, en la práctica no suele ser factible entrenar a cada modelo individual sobre un conjunto de datos independiente [20].

  Para solventar este problema, los modelos de ***bagging*** entrenan cada modelo sobre una **muestra uniformemente aleatoria con reemplazo** (*bootstrap* en inglés) del conjunto de datos original [20] - obteniendo como resultados modelos sencillos e independientes, y siendo la predicción final el **promedio** de las predicciones de cada modelo.

  Algunos de los modelos de *bagging* más importantes son los siguientes:

  - ***Random Forests* [21]:** Un modelo de *ensemble bagging* de **árboles de decisión profundos** en el que se añade un segundo proceso de muestreo aleatorio para aumentar la independencia entre los predictores simples entrenados:

  - **Muestreo de instancias:** Cada árbol se entrena con un subconjunto aleatorio de instancias del conjunto de datos.
  - **Muestreo de atributos:** Cada nodo del árbol es entrenado sobre un subconjunto aleatorio de atributos del conjunto de datos.

- ***Extremely Randomized Trees [22]:*** Una evolución del modelo de *Random Forests* en el que se añade un proceso de muestreo adicional, con los siguientes cambios:
  - Cada árbol pasa a entrenarse sobre el **conjunto de datos completo**, sin muestreo - aunque se sigue realizando un muestreo de los atributos a considerar por cada nodo del árbol.
  - Durante la construcción del árbol, se generan de forma aleatoria varias **particiones del conjunto de datos** para cada atributo - en vez de calcular la partición óptima. A la hora de construir cada nodo, se elige la partición que mejor puntuación obtiene de todas las generadas.

- ***Boosting*:**

  Para garantizar la independencia entre los modelos entrenados, los modelos de *ensemble* de tipo ***boosting*** optan por entrenar sus modelos de forma **secuencial** sobre un **conjunto de datos con pesos** - donde, para cada modelo, se da más peso a las instancias del conjunto de datos que se han predicho erróneamente en los modelos anteriores [12].

  Durante el trabajo, se ha considerado el siguiente modelo de *boosting*:

  - ***Adaptive Boosting [23]:*** Un modelo básico de *boosting* que sigue estrictamente el proceso descrito. Concretamente, se comienza con un conjunto de datos de pesos uniformes sobre el que se entrena el primer modelo, ajustando los pesos de las instancias - aumentando el peso de las instancias con mayor error, y reduciendo el peso de las instancias con menor error. Tras esto, se repite el proceso de entrenamiento de modelos y ajuste de pesos hasta entrenar todos los predictores [12].

    La predicción final es una **media ponderada por el error** de la predicción de todos los modelos - donde los modelos con menor error tienen un mayor peso en la ponderación.

  Una subfamilia dentro de estos algoritmos son los modelos de ***Gradient Boosting***. Estos modelos también utilizan la metodología de *boosting* - entrenar modelos secuencialmente ajustándose a los errores del modelo anterior -, con la diferencia de que el entrenamiento no se hace sobre el conjunto de datos directamente, sino sobre los **errores residuales** (la diferencia entre la predicción y el valor real) de cada instancia [24].

  Este comportamiento es similar al **gradiente descendiente** utilizado para entrenar otros modelos como la regresión lineal o las redes neuronales [13]. En concreto, se llevan a cabo los siguientes pasos:

  1. Se comienza realizando una predicción inicial, generalmente el valor promedio de todas las instancias.

2. Utilizando el valor estimado, se calculan los **errores pseudo-residuales** de cada instancia - el **error** entre la predicción y el valor real. Estas pseudo-residuales dependen de la función de error que se elija.
3. Repitiéndose para cada modelo que se tiene que entrenar:
   (a) A partir de los valores pseudo-residuales, se entrena un modelo simple que **predice el valor residual de cada instancia**.
   (b) Utilizando los nuevos residuales estimados, se recalculan los valores pseudo-residuales para que el siguiente modelo ajuste mejor a las instancias clasificadas erroneamente.

Para obtener una predicción final, se suma al valor promedio inicial los residuales calculados por cada uno de los modelos - ponderados por un **factor de aprendizaje** para evitar el sobreajuste [24].

Actualmente, los modelos de *Gradient Boosting* son considerados el estado del arte para la mayoría de problemas de predicción estructurada [13], siendo algunos de los modelos más populares los siguientes:

- ***Extreme Gradient Boosting* [25]:** Un algoritmo de *Gradient Boosting* utilizando el método de Newton-Raphson - en vez de calcular los errores pseudo-residuales, se calcula una función de la segunda y la primera derivada de la función de error. Además, el modelo está diseñado para permitir el entrenamiento en paralelo de los árboles.
- ***Categorical Boosting* [26]:** Un algoritmo de *Gradient Boosting* diseñado para trabajar de forma nativa con atributos categóricos sin necesidad de convertirlos previamente a valores numéricos.

  El modelo se entrena utilizando ***Ordered Boosting*** - utilizando una permutación aleatoria del conjunto de entrenamiento para cada modelo, donde para calcular las pseudo-residuales de cada instancia se consideran solo las instancias anteriores en la permutación [27] - para evitar introducir sesgos.
- ***Light Gradient-Boosting Model* [28]:** Un algoritmo de *Gradient Boosting* con las siguientes características:
  - **Histogramas:** Para optimizar el rendimiento, los valores de los atributos continuos se agrupan en histogramas.
  - **Crecimiento del arbol por hojas:** Frente a otros algoritmos que entrenan los árboles nivel a nivel, el modelo ramifica siempre por la hoja que minimizaría el error - llevando a árboles más ajustados.

  Existe otra implementación de este algoritmo, conocida como **Histogram-Based Gradient Boosting**, ofrecida por la librería de ciencia de datos *Scikit-Learn* [29] - aunque ambas se basan en el mismo modelo y no presentan diferencias significativas.

3. Estudio exploratorio del problema

En este capítulo se estudia en detalle el problema a resolver a través del proceso de ciencia de datos. Se comienza realizando una definición del problema y sus objetivos, seguido por un **análisis exploratorio de los datos** que lo definen. En este análisis se estudian los atributos que describen los datos junto a sus distribuciones y comportamientos, haciendo hincapié en los atributos de carácter geográfico, social y económico.

3.1. Definición y objetivos del problema

El **cáncer de mama triple negativo** es uno de los cánceres de mama más agresivos y difíciles de tratar. En el caso de que además se agravase con una **metástasis**, se necesita un tratamiento rápido y urgente, sin retrasos innecesarios, para aumentar al máximo las posibilidades de éxito. Ahora bien, el tiempo de espera para acceder a dicho tratamiento no es igual para todos los pacientes, y existe la posibilidad de que hayan sesgos influyendo en el tiempo de diagnóstico - como pueden ser algunos factores geográficos, socioeconómicos o incluso climáticos [1].

El principal objetivo del problema - y, por tanto, el objetivo que guía el proceso de ciencia de datos - es **crear un modelo de regresión** capaz de predecir el tiempo de diagnóstico de metástasis en base a la información dada de un paciente. Además, se busca estudiar la **influencia de factores geográficos, socioeconómicos y climáticos** en dicho tiempo de diagnóstico, para comprobar si existe un sesgo real en el trato a los pacientes. El problema a resolver se planteó originalmente como el segundo de los desafíos ofrecidos por la institución **Women in Data Science** como parte de su *Datathon* de 2024 [1].

3.2. Análisis exploratorio de datos

El primer paso en el proceso de ciencia de datos es realizar un estudio exhaustivo del conjunto de datos con el fin de comprender mejor su comportamiento y la distribución de sus datos.

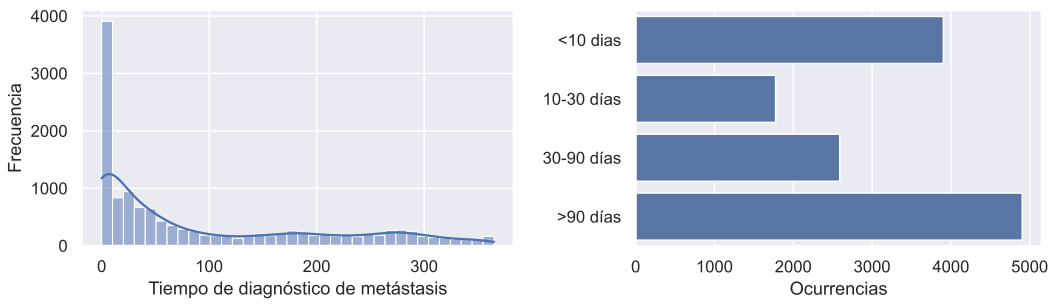
3.2.1. Distribución del conjunto de datos

El conjunto de datos contiene un total de **13173 instancias**, cada una de ellas descrita por **150 atributos** - divididos en **11 atributos categóricos** y **139 atributos numéricos** - y una **variable objetivo numérica**. Describir individualmente todos los atributos en la memoria no sería factible, por lo que se describen los principales grupos de atributos:

- **Atributos médicos (13: 11 categóricos y 2 numéricos)**: Datos identificativos e información sobre el diagnóstico, tratamiento y seguro del paciente.
- **Atributos socioeconómicos (65: 2 categóricos y 63 numéricos)**: Por lo general, datos **estadísticos** reflejando información socioeconómica relacionada con la población del paciente. Estos estadísticos se pueden dividir, a su vez, en:
  - **Porcentajes (49 atributos numéricos):** Porcentajes en el rango $[0, 100]$ representando valores estadísticos: matrimonios, educación, demografía, etc. de la ubicación del paciente.
  - **Medianas (10 atributos numéricos):** Valores representando la mediana de algunos estadísticos: edad, ingresos, alquileres, etc. de la ubicación del paciente.
  - **Información geográfica (6: 2 categóricos y 4 numéricos):** Valores concretos - población, densidad... - de la ubicación del paciente, sin ser representados a través de un porcentaje o una mediana.
- **Atributos climáticos (72 atributos numéricos)**: Temperatura promedio (en grados **Farenheit**) de la población del paciente - representada de forma mensual entre los años 2013 y 2018.

Variable objetivo - tiempo de diagnóstico

La variable objetivo - el **tiempo de diagnóstico de una metástasis** - es una variable **numérica entera** con valores en el conjunto de entrenamiento en el rango $[0 - 365]$, cuya distribución se puede observar en la **Figura 3.1**.



(a) Distribución general **(b)** Distribución agrupada

Figura 3.1: Distribución del tiempo de diagnóstico

Se puede ver que los tiempos de diagnóstico siguen una **distribución de Poisson** - con la mayoría de casos diagnosticados en un rango de $[0 - 10]$ días. Ahora bien, como se observa en la **Figura 3.1b**, si se agrupan los valores en rangos **la mayoría de casos tardan más de 90 días en ser diagnosticados**.

Valores perdidos

Antes de realizar un estudio más exhaustivo de los atributos, es de interés estudiar el comportamiento de los **valores perdidos** en el conjunto de datos, para comprobar si hay un gran número de éstos, si existen atributos irrelevantes por tener un alto grado de información perdida y si sería necesario realizar algún tipo de tratamiento sobre éstos valores.

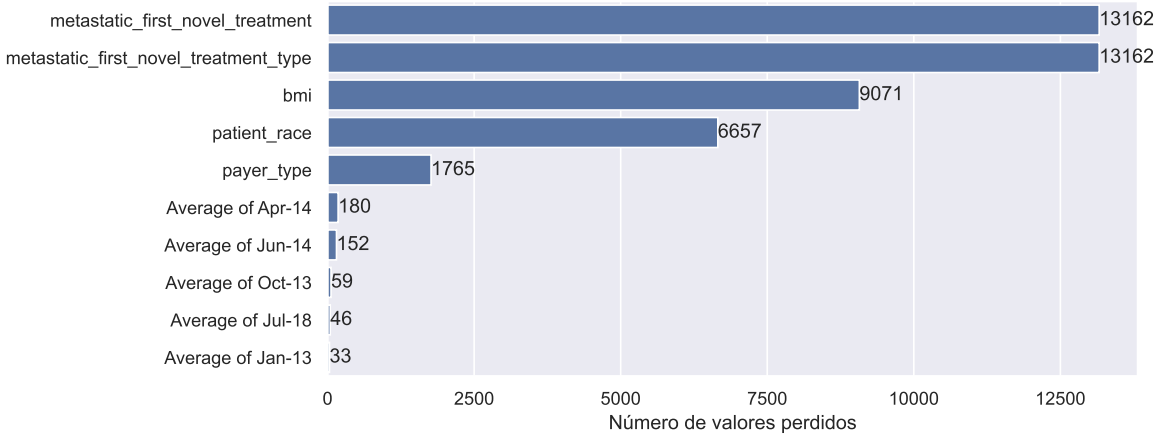


Figura 3.2: Distribución de valores perdidos en el conjunto de datos

Estudiando la distribución, **72** de los 150 atributos disponibles presentan valores perdidos - con un promedio de **624 instancias perdidas** por atributo. En primera instancia puede parecer un número muy elevado de valores perdidos, si se observa cómo se distribuyen los valores perdidos — como se representa en la **Figura 3.2** — se puede observar un **sesgo** claro, donde la amplia mayoría de valores perdidos se agrupan alrededor de cinco atributos:

- **Tratamiento:** Debido al número tan elevado de valores perdidos en ambos atributos, **solo se tiene información sobre el tratamiento de 11 pacientes** - lo que significa que no sería relevante el atributo debido a la falta de información.
- **Índice de masa corporal (IMC) del paciente:** Se conoce el índice de masa corporal de **menos de la mitad de los pacientes.** Además, al ser información numérica **no existe un valor por defecto** por el que se puedan reemplazar los valores perdidos - por lo que sería razonable no estudiar en más detalle el atributo.
- **Raza y tipo de seguro médico del paciente:** En ambos casos hay un número considerable de instancias con valores desconocidos. Ahora bien y a diferencia del IMC, al ser atributos categóricos puede considerar que **es significativo para el estudio que no se conozcan estos valores** - tratándolos como una categoría adicional, *"Desconocido"*.

En el resto de atributos el número de valores perdidos es más reducido - en el orden de **100 instancias** o menor -, por lo que el tratamiento es más simple, pudiendo descartar las instancias con valores perdidos o realizando una imputación simple con el valor promedio.

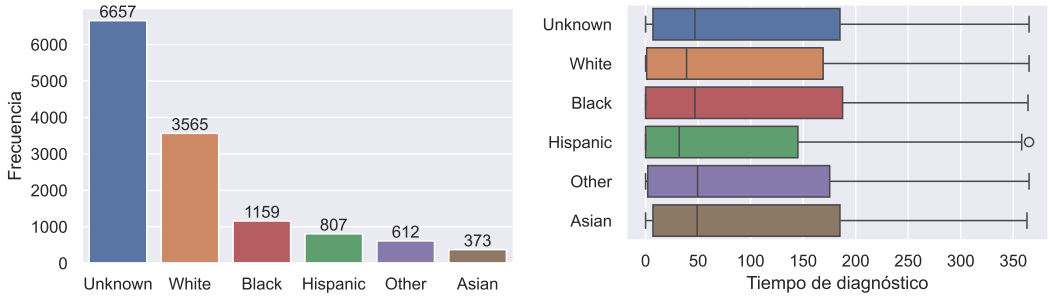
3.2.2. Estudio de atributos categóricos

Tras un primer análisis - en el que se ha estudiado la distribución de la variable objetivo, los atributos y sus valores perdidos -, la segunda parte del análisis exploratorio es realizar un **estudio exhaustivo individualizado** de los atributos de interés y su relación con la variable predictora. Al ser reducido el número de atributos categóricos (con un total de **11**) es posible realizar un análisis individual de cada uno de estos atributos:

Datos personales: raza, género y tipo de seguro del paciente

- **Raza del paciente:**

  Como se ha comentado durante el estudio de los valores perdidos, hay **un número significativo de valores perdidos** de este atributo - que serán tratados como una categoría adicional, *"Unknown"*.



(a) General **(b)** Respecto al tiempo de diagnóstico

Figura 3.3: Distribución de la raza del paciente

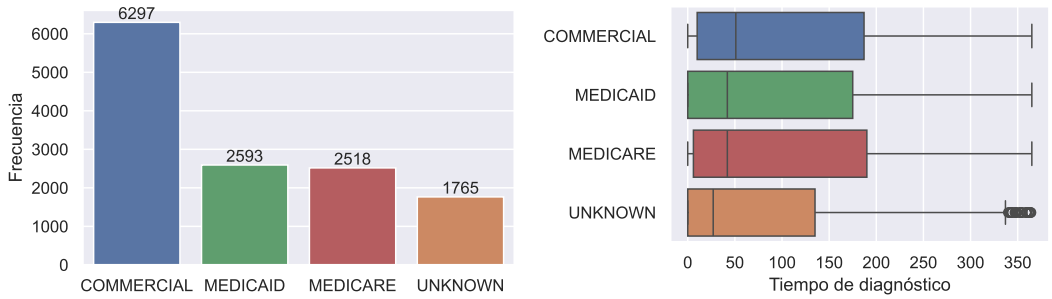
En la **Figura 3.3** se puede observar que:

- **Distribución general:** Como se podía esperar, la mayoría de pacientes presentan una raza desconocida - algo que se puede interpretar como que **la mayoría de los pacientes no se sienten cómodos especificando su raza**. Tras esto, la raza **blanca** es la más frecuente - siendo tres veces más frecuente que la raza negra -, siendo la raza asiática la menos frecuente.
- **Relación con el tiempo de diagnóstico:** Si bien todas las razas tienen un rango de tiempos de diagnóstico amplio, **las razas blanca e hispánica tienen una mediana ligeramente inferior al resto** - sugiriendo que **la raza puede influir en el tiempo de diagnóstico**.

Dada estas observaciones, puede resultar de interés considerar la raza del paciente a la hora de hacer una selección de atributos.

- **Tipo de seguro médico del paciente:**

  Igual que con la raza, hay una cantidad significativa de valores perdidos de este atributo - que serán categorizados como un nuevo valor, *"UNKNOWN"*.



(a) General **(b)** Respecto al tiempo de diagnóstico

Figura 3.4: Distribución del tipo de seguro médico del paciente

En la **Figura 3.4** se puede estudiar que:

- **Distribución general:** El seguro más frecuente - siendo la mitad del conjunto de datos - es el **seguro comercial privado**. Los dos seguros públicos - **Medicaid** y **Medicare Advanced** - tienen proporciones similares entre sí, siendo en conjunto algo inferior al número de seguros privados. Finalmente, hay una cantidad ligeramente inferior de seguros desconocidos - que podría referirse a **pacientes sin seguro médico**.
- **Relación con el tiempo de diagnóstico:** En contra de lo que se podría esperar, los seguros desconocidos presentan **un tiempo de diagnóstico mediano y un rango sustancialmente inferior** al del resto de seguros. Aunque los tres tipos de seguros restantes tienen distribuciones similares, parece que **los seguros privados tienen un tiempo de diagnóstico ligeramente superior**.

Dadas estas diferencias, es posible que **el tipo de seguro del paciente influya en el tiempo de diagnóstico** - por lo que será considerado posteriormente a la hora de realizar una selección de atributos.

- **Género del paciente:**

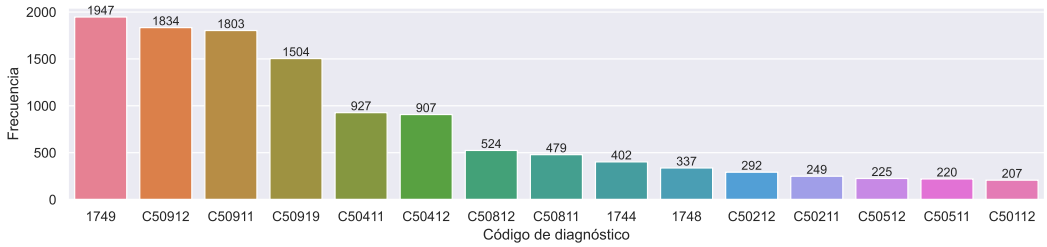
  Pese a no haber ningún valor perdido, el atributo presenta un problema de cara a su uso posterior: **todas las instancias del conjunto de datos tienen el mismo valor (mujer)**. Por tanto, el uso de este atributo no ofrece ninguna capacidad discriminatoria y puede ser descartado sin problema.

Datos médicos: códigos de diagnóstico y tipos de tratamiento

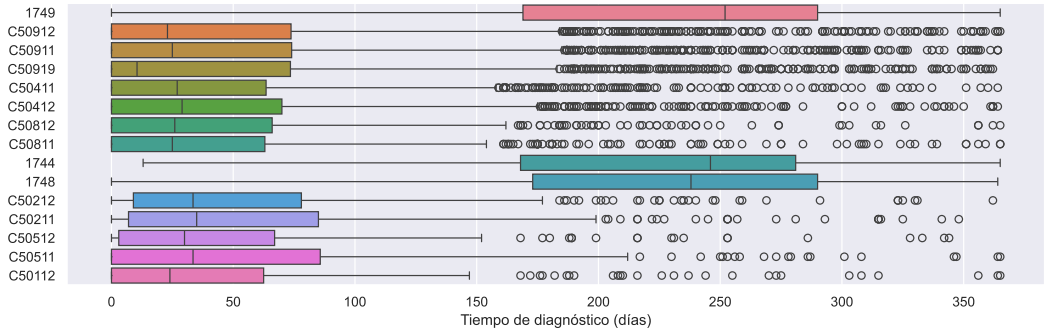
- **Código de diagnóstico de cáncer de mama:**

  Existen dos atributos en el conjunto de datos clasificando la misma información: **código de diagnóstico** y **descripción del diagnóstico**. Al representar la misma información - y tras comprobar que hay una correlación directa entre ambos atributos -, es suficiente con estudiar **uno de los dos atributos**, eligiendo estudiar el **código de diagnóstico de cáncer de mama**.

  El principal problema a la hora de estudiar este atributo es que se tienen **47 valores únicos** para el atributo, siendo un número demasiado elevado para estudiar en detalle. Además, no hay garantía de que **estos valores sean exhaustivos** - es decir, es posible que **existan códigos de diagnóstico no incluidos en el conjunto de entrenamiento**. Por esto, se realizará el estudio sobre los **15 códigos más frecuentes**.



(a) General



(b) Respecto al tiempo de diagnóstico

Figura 3.5: Distribución de los 15 valores más frecuentes del código de diagnóstico de cáncer de mama

En la **Figura 3.5a** se estudia la distribución general, y como se puede observar, **existen dos tipos de codificaciones** - las codificaciones que empiezan por la letra **C** (ICD10, más moderno) y las que empiezan por un número (ICD9). Además, se puede ver que la mayoría de diagnósticos se encuentran agrupados en cuatro códigos - con la frecuencia del resto de atributos bajando rápidamente hasta llegar a los diagnósticos con una o dos instancias no representados. Estos diagnósticos, si se comprueba su código descriptivo, hacen referencia a **cánceres en sitios sin especificar** - es decir, los códigos más genéricos y por tanto

aplicables a un mayor número de casos.

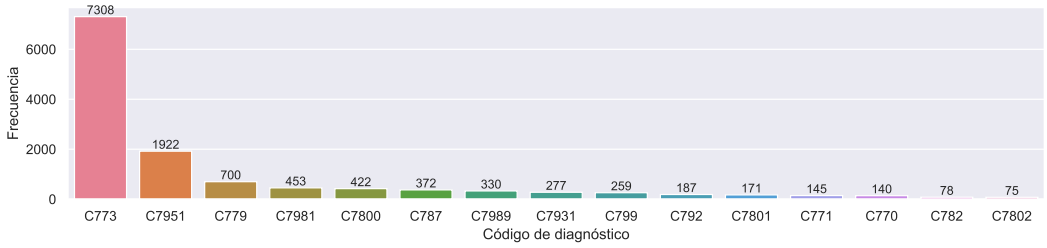
En la **Figura 3.5b** se puede apreciar una diferencia clara en tiempos de diagnóstico dependiendo de la **codificación utilizada**, con los diagnósticos con codificación de tipo **ICD9** teniendo un tiempo de diagnóstico promedio notablemente superior. Si bien no hay una explicación clara para ésto, puede deberse a que hagan referencia a casos más antiguos -y, por tanto, casos con menor conocimiento y recursos.

Por esto, resulta evidente que el código de diagnóstico de cáncer de mama ofrece información discriminatoria muy relevante de cara a ser utilizada en el modelo posterior.

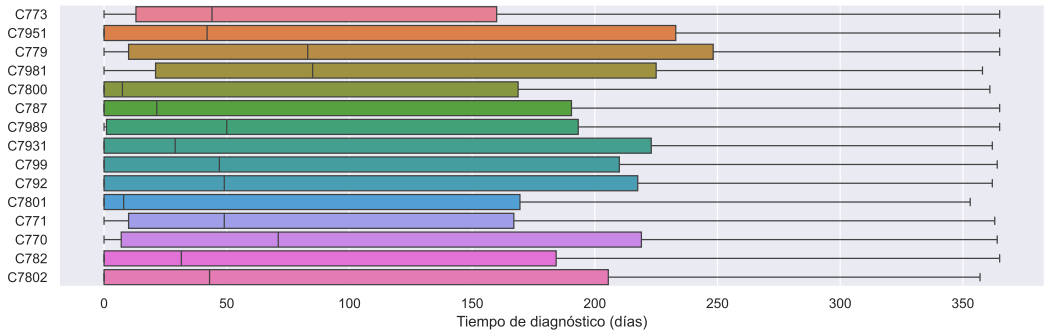
- **Código de diagnóstico de cáncer metastásico:**

  A diferencia del código de diagnóstico para cáncer de mama, para el **diagnóstico de cáncer metastásico** solo se tiene un atributo.

  Ahora bien, se sigue teniendo el mismo problema de dimensionalidad: el conjunto de datos contiene **43 valores únicos** para este atributo, sin garantía de que sea un conjunto **exhaustivo**. Por esto, se realizará el estudio sobre los **15 códigos más frecuentes**.



(a) General



(b) Respecto al tiempo de diagnóstico

Figura 3.6: Distribución de los 15 valores más frecuentes del código de diagnóstico de cáncer de mama

Como se observa en la **Figura 3.6a** y a diferencia del diagnóstico de cáncer de mama, en este caso **la mayoría de diagnósticos se concentran alrededor de un único diagnóstico** - $C773$, metástasis en nodos linfáticos auxiliares y de las extremidades superiores -, con el resto de diagnósticos reduciendo su frecuencia rápidamente.

Respecto a la relación con el tiempo de diagnóstico, en la **Figura 3.6b** se puede ver que, si bien no hay una diferencia tan pronunciada como en el caso del código de diagnóstico de cáncer de mama, **el tipo de metástasis diagnosticado parece tener influencia sobre el tiempo necesario para su diagnóstico**. Ahora bien, si se estudia la localización de la metástasis de cada código estudiado **no se observa correlación entre la localización y el tiempo de diagnóstico**.

Por esto, se puede considerar al **código de diagnóstico del cáncer metastásico** otra variable de gran interés para los modelos desarrollados posteriormente - pudiendo ofrecer una gran capacidad discriminatoria.

- **Tipo de tratamiento:**

  Como se mencionó durante el estudio de los valores perdidos, **solo se tienen 11 valores** para este atributo - de **13173** instancias totales. Por tanto, no tiene ningún sentido estudiar este atributo, al no contener suficiente información para ser significativo.

Datos geográficos: estado de residencia y ubicación geográfica

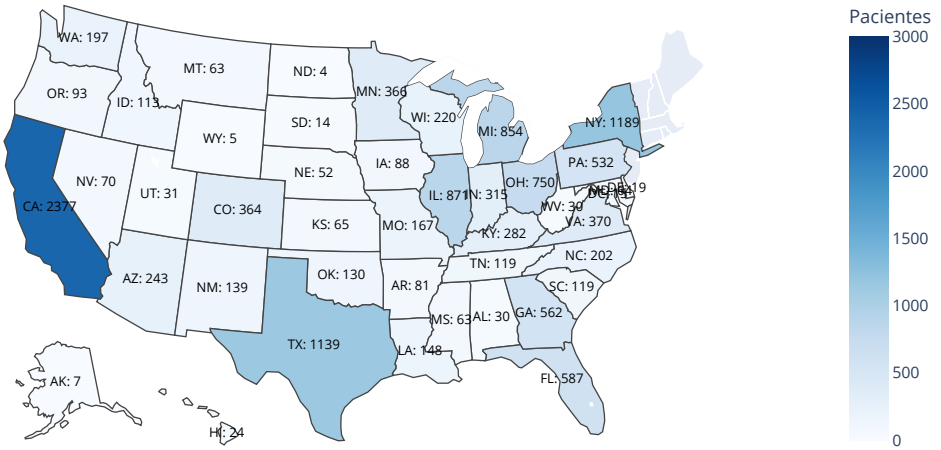
El estudio de la información geográfica del conjunto de datos es de especial importancia al ser uno de los objetivos planteados por el problema a resolver. Además, toda la información socio-económica y climática del conjunto de datos está **asociada al código zip de los pacientes**, por lo que estos atributos codifican además de forma innata todos estos factores de sesgo.

Ahora bien, la información se encuentra representada en el conjunto de datos a través de **4 atributos jerárquicos**, donde cada atributo inferior describe con más granularidad el atributo superior. Al codificar la misma información, es de interés seleccionar **un único atributo** sobre el que realizar el estudio.

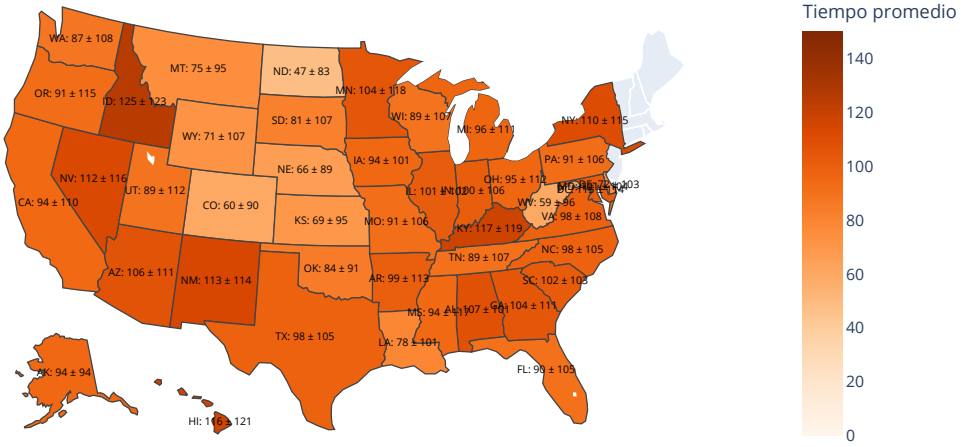
Atributo	Valores totales	p-valores (Tests de hipótesis)	
		Paramétrico (ANOVA)	No paramétrico (Kruskal)
Región	4	$2,80 \times 10^{-3}$	$1,93 \times 10^{-6}$
División	8	$6,13 \times 10^{-3}$	$1,68 \times 10^{-5}$
Estado	44 Número m	5.86×10^{-10}	1.06×10^{-16}
Código zip	751	—	—

Tabla 3.1: p-valores de los atributos geográficos

Para realizar la selección se han realizado **tests de hipótesis** - tanto paramétricos (**ANOVA**, para estudiar desviaciones en la media) como no paramétricos (**Kruskal-Wallis**, para estudiar desviaciones en la mediana) - sobre todos los atributos excepto el código zip, debido a su alta dimensionalidad. Los resultados se pueden observar en la **Tabla 3.1**, siendo el atributo a seleccionar el **estado**, al tener el p-valor más bajo - y, por tanto, tener la mayor certeza de que **su valor influye sobre el promedio del tiempo de diagnóstico**.



(a) Frecuencia



(b) Tiempo de diagnóstico promedio

Figura 3.7: Distribución geográfica de los pacientes

En el mapa de la **Figura 3.7a** se representa la distribución de los pacientes en el mapa de los Estados Unidos, donde se observa que los pacientes están **agrupados en estados concretos** - en general, los estados de mayor población -, sin haber una correlación geográfica clara en su ubicación. También se observa que **existen algunos estados sin pacientes** - por lo que, igual que con los códigos de diagnóstico, **los valores del atributo no son exhaustivos**.

Estudiando el valor promedio del tiempo de diagnóstico, la **Figura 3.7b** muestra que el **tiempo de diagnóstico promedio es similar entre todos los estados** - rondando alrededor de los **80 días**, pero ubicado en el rango de los **60 a 120 días**. También es llamativo el hecho de que **la desviación estándar es muy elevada** - en la práctica totalidad de los estados se trabaja con desviaciones estándar de alrededor de **100 días**.

El test de hipótesis indica que el estado del paciente **influye de forma significativa sobre el tiempo de diagnóstico**, por lo que es un atributo de interés de cara a la creación posterior de modelos. Ahora bien, el estudio gráfico de la distribución también muestra que existe un **error sustancial** en dicha diferencia, al haber un rango muy elevado de posibles valores dentro de cada estado.

3.2.3. Estudio de atributos numéricos

Tras el estudio de los atributos categóricos, el siguiente paso es realizar un **estudio exhaustivo** de los atributos numéricos más significativos. Sin embargo, el elevado número de atributos, con un total de **138 variables numéricas**, hace imposible el estudio individualizado. Por tanto, el objetivo es seleccionar los **principales atributos numéricos** - entendiendo como tales los **atributos con mayor influencia sobre el tiempo de diagnóstico**.

Una forma de realizar esta selección es mediante el **coeficiente de correlación de Pearson** - un valor numérico en el rango $[-1, 1]$ indicando la **relación lineal entre dos atributos**, donde los valores cercanos a los extremos indican una correlación fuerte y un valor cercano a 0 indica independencia entre los atributos.

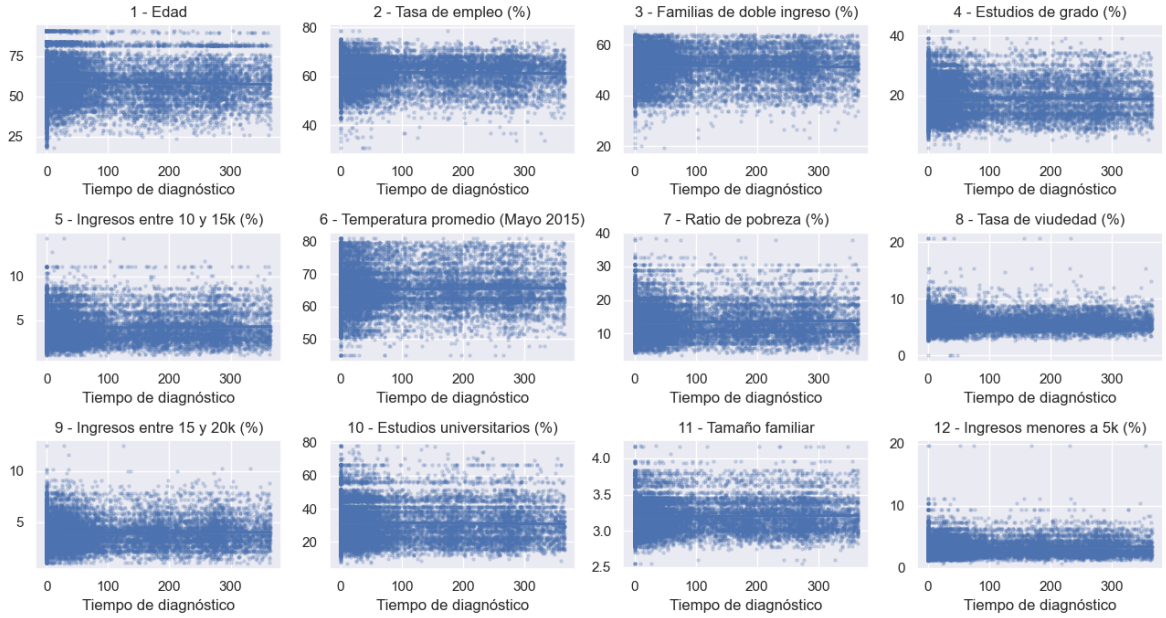


Figura 3.8: Relación entre valor y tiempo de diagnósticos para los 12 atributos de mayor correlación

Ahora bien, como se refleja en la **Figura 3.8**, cuando se calcula la correlación entre los **138 atributos** y el tiempo de diagnóstico se observa claramente que **los valores de correlación de Pearson son muy bajos** - siendo el valor más alto de 0.055. Estos valores se traducen en que **los atributos numéricos no tienen apenas influencia sobre el tiempo de diagnóstico** - y, por tanto, **pueden ser descartados** en las siguientes etapas del proceso de ciencia de datos sin ninguna repercusión.

Al ser la práctica totalidad de atributos socioeconómicos y climáticos de tipo numérico, esto también se traduce en una respuesta al segundo objetivo del problema: identificar que, en contra de lo que se podría esperar, **los factores socioeconómicos y climáticos no parecen tener influencia sobre el tiempo de diagnóstico de metástasis** - al menos, para el conjunto de datos proporcionado.

4. Preparación del conjunto de datos

En este capítulo se describe el conjunto de transformaciones y técnicas aplicadas sobre el conjunto de datos para transformarlo, para su uso posterior durante las etapas de entrenamiento y experimentación.

En primer lugar se propone un número de **subconjuntos de atributos** de cara a reducir la dimensionalidad del conjunto de datos. Tras esto, se plantean y describen las **transformaciones** aplicadas al conjunto de datos previo al entrenamiento para estandarizar los datos y mejorar el rendimiento de los modelos.

4.1. Selección de atributos

Durante el análisis exploratorio se realizó un estudio exhaustivo de los atributos contenidos en el conjunto de datos, en el que se han identificado los siguientes problemas:

- **Alta dimensionalidad:** El conjunto de datos tiene **150 atributos** en total, donde la mayoría de atributos categóricos tienen **40 o más valores únicos**. Esto, unido al número de instancias bajo para dicha complejidad, puede significar que el modelo acabaría **sobreajustándose** al no poder aprender generalizaciones de forma adecuada.
- **Irrelevancia de los atributos:** De los 150 atributos estudiados, **la amplia mayoría no presentan correlación con la variable objetivo** - por lo que mantenerlos puede implicar una disminución del rendimiento final del modelo y un aumento del tiempo de entrenamiento.

Debido a esto, resulta necesario realizar una **selección de atributos** - proponiendo varios **subconjuntos de atributos** a evaluar durante la experimentación y selección de modelos, con el objetivo de optimizar el rendimiento del modelo reduciendo la dimensionalidad.

4.1.1. Selección manual de atributos

El primer subconjunto de atributos propuesto se realizar a partir de las observaciones obtenidas a través del análisis exploratorio de datos realizado en el capítulo anterior - estando este formado por las **variables con mayor significancia para la predicción del tiempo**

de diagnóstico, según las gráficas y tests realizados. Tras este estudio, se han seleccionado los siguientes **5 atributos**:

- Código de diagnóstico del cáncer de mama.
- Código de diagnóstico del cáncer metastásico.
- Estado de residencia del paciente.
- Raza del paciente.
- Tipo de seguro médico del paciente.

Como se puede observar, **todos los atributos seleccionados son categóricos**. Esto se debe a la correlación prácticamente nula entre los atributos numéricos y el tiempo de diagnóstico. Además, se ha optado por representar la información geográfica a través del **estado de residencia** - al ser el atributo geográfico con menor valor en los tests de hipótesis.

A través de esta selección se ha reducido el conjunto de atributos de **150 a 5 atributos**, reduciendo sustancialmente la dimensionalidad. Ahora bien, los atributos elegidos siguen siendo complejos debido al gran número de valores posibles, por lo que será necesario un procesamiento posterior para **agrupar los valores menos frecuentes**.

4.1.2. Selección automática de atributos

El subconjunto de atributos propuesto en la sección anterior está basado en el análisis exploratorio realizado. Ahora bien, al basar la decisión únicamente en tests estadísticos y gráficas - sin evaluar el rendimiento real en modelos, -, existe la posibilidad de que se haya introducido un **sesgo personal** o que existan otros subconjuntos de atributos que puedan ofrecer un mejor rendimiento.

Con el fin de solucionar estos problemas y de acercar el proceso de selección de atributos al funcionamiento real de los modelos, se proponen **dos subconjuntos adicionales** obtenidos a través de **algoritmos de selección automática de variables** [30] - basados en técnicas estadísticas y en entrenamiento de modelos.

Filter: Selección mediante tests estadísticos

Los **métodos de filtrado** (también conocidos como *filter*) son algoritmos que evalúan la **relevancia de cada atributo** a través de tests estadísticos, sin necesidad de entrenar ningún modelo - lo que los hace más ágiles que otros métodos, pero más genéricos e incapaces de encontrar todas las correlaciones entre grupos de atributos [30].

Para este problema se ha utilizado un **test estadístico F** - una medida de la **dependencia lineal** entre atributos -, calculando la correlación entre cada atributo numérico y el tiempo de diagnóstico. A partir de estas puntuaciones, se eligen los **10 atributos** con mayor dependencia lineal:

- **Atributos categóricos (5):**
  - **Código de diagnóstico:** Cáncer de mama y cáncer metastásico.
  - **Atributos del paciente:** Tipo de seguro médico, raza y estado de residencia del paciente.

- **Atributos numéricos (5):**
  - **Atributos del paciente:** Edad del paciente.
  - **Estadísticos socioeconómicos (porcentajes):** Tasa de empleo, habitantes con estudios de grado, familias con dos o mas ingresos y habitantes con estudios universitarios o superiores.

El subconjunto de atributos obtenido reafirma la selección manual realizada, al tener ambos conjuntos los **mismos atributos categóricos** - siendo el **código de diagnóstico del cáncer de mama** el atributo con mayor relevancia con diferencia. La principal diferencia se encuentra en que se han seleccionado además **atributos numéricos**, algunos de ellos teniendo incluso mayor relevancia que otros atributos categóricos - como la **edad del paciente**.

Wrapper: Selección mediante entrenamiento de modelos

Los **métodos de envoltura** (también conocidos como *wrapper*) son algoritmos que realizan su selección de atributos a través del **entrenamiento de un modelo de aprendizaje automático** y la selección de las variables más relevantes en base a los parámetros y pesos aprendidos por el modelo. A diferencia de los métodos de *filter*, el proceso de selección suele ser más lento, pero los resultados suelen ser más fiables al trabajar de forma directa con modelos reales [30].

Para este problema se ha utilizado un modelo de ***Random Forests***, entrenado con los hiperparámetros por defecto de su implementación en *Scikit-Learn* - **100 árboles** sin profundidad máxima. A partir de este modelo entrenado se extraen los **10 atributos** con mayor peso sobre el modelo entrenado:

- **Atributos categóricos (4):**
  - **Código de diagnóstico:** Cáncer de mama y cáncer metastásico.
  - **Atributos del paciente:** Tipo de seguro médico y raza del paciente.
- **Atributos numéricos (6):**
  - **Atributos del paciente:** Edad e índice de masa corporal del paciente.
  - **Estadísticos socioeconómicos**: Tiempo de viaje al trabajo promedio, porcentaje de personas de raza nativa, porcentaje de habitantes con estudios STEM, porcentaje de habitantes con edades entre 40 y 49 años.

Frente a las selecciones manuales y de filtrado, el **subconjunto *wrapper* no incluye información geográfica en su selección** - optando, en su lugar, por incluir un mayor número de atributos numéricos tanto médicos como socioeconómicos.

4.2. Pre-procesamiento de los datos

A la par que se propone una selección de atributos para reducir la dimensionalidad del conjunto de datos, resulta también necesario realizar un **pre-procesamiento** - una serie de transformaciones secuenciales sobre los datos - para reducir la complejidad y paliar posibles

problemas como los valores perdidos o la codificación de los atributos categóricos. De esta forma, se busca mejorar el rendimiento de los modelos entrenados.

Ahora bien, los atributos necesitan **transformaciones distintas** dependiendo del tipo de datos que representen - siendo necesario distinguir entre atributos numéricos y categóricos. Para el trabajo descrito en la memoria, se han propuesto las siguientes transformaciones dependiendo del tipo de datos del atributo:

- **Atributos categóricos:**

  1. **Imputación de valores perdidos:** Como se estudió durante el análisis exploratorio de datos, para la mayoría de atributos categóricos **resulta de interés tratar los valores perdidos como categorías separadas**, al ser relevante para el estudio que faltasen valores.

     Por esto, se opta por reemplazar todos los valores perdidos por un **valor constante, *"UNKNOWN"***.
  2. **Codificación:** De forma inherente, la mayoría de modelos propuestos son incapaces de trabajar con atributos categóricos - necesitando transformar estos atributos en algún tipo de codificación numérica.

     Para estos casos, lo estándar es utilizar una codificación de tipo ***One-Hot*** [31]: dividiendo el atributo original en **tantos atributos como valores tiene la variable original**, donde el atributo que se corresponde con el valor de la variable original tiene un valor de 1 y el resto tiene valores de 0 - representando de esta manera los valores categóricos en un formato numérico.

     Es importante destacar las siguientes particularidades para el problema actual:

     - Debido a que los atributos categóricos del conjunto de datos tienen una **alta dimensionalidad** y **valores no exhaustivos**, es necesario realizar una **agrupación de los atributos menos frecuentes y desconocidos** bajo un único atributo at_{other}. El umbral para considerar un atributo como poco frecuente será determinado durante el proceso de experimentación y selección de modelos.
     - La implementación de los modelos de ***Gradient Boosting*** codifican de forma inherente los atributos categóricos, por lo que no es necesario aplicar este paso para ellos.

- **Atributos numéricos:**

  1. **Imputación de valores perdidos:** Debido a la presencia de valores extremos en la mayoría de atributos numéricos, se reemplazan los valores perdidos por el **valor mediano del atributo al que pertenece** - ofreciendo de esta forma un valor promedio resistente a sesgos y *outliers*.
  2. **Escalado:** Por lo general, es necesario **escalar los datos** - transformar los valores de todos los atributos para que se encuentren en el mismo rango - de los atributos numéricos para el funcionamiento adecuado de los modelos.

     Para el problema descrito, se intenta evitar los problemas introducidos por los valores extremos utilizando un escalado alrededor de la **mediana y el rango intercuartil**, donde cada valor se transforma utilizando la fórmula $z(x) = \frac{x - mediana}{IQR}$.

5. Modelado y evaluación

En este capítulo se detalla el modelado y evaluación llevado a cabo para resolver el problema de ciencia de datos propuesto y obtener un modelo de regresión final capaz de predecir el tiempo de diagnóstico de los pacientes.

En primer lugar se describe la **experimentación** a realizar - empezando por los modelos propuestos, los hiperparámetros planteados para cada uno de ellos y el proceso de **búsqueda** realizado para ajustarlos; y siguiendo con el proceso final de **selección de modelos** evaluando sobre un conjunto de datos separado. Tras esto, se muestran y estudian los **resultados** - tanto el rendimiento de los modelos y sus atributos como el comportamiento del modelo final elegido en comparación con el modelo ganador de la competición.

5.1. Modelado y experimentación

Tras el análisis exploratorio y el preprocesamiento del conjunto de datos realizado en los capítulos anteriores, la siguiente etapa del ciclo de vida de la ciencia de datos es el **modelado**: la propuesta de modelos de aprendizaje automático e hiperparámetros, el ajuste de éstos y el proceso de selección final para obtener un **modelo de regresión entrenado** capaz de resolver el problema propuesto con el menor error posible.

Para llevar a cabo este ajuste y selección es necesario realizar a la vez un proceso paralelo de **experimentación**, donde se evalua el rendimiento de los modelos bajo distintos conjuntos de parámetros y circunstancias, con el fin de seleccionar el mejor modelo de entre todos los propuestos.

5.1.1. Modelos e hiperparámetros propuestos

Se han propuesto un total de **16 modelos** para resolver el problema de la predicción del tiempo de diagnóstico de cáncer - correspondiéndose con los algoritmos estudiados durante la revisión de técnicas realizada en el Capítulo 2. Además, para cada uno de estos modelos se ha planteado una **malla de hiperparámetros** - un rango de posibles valores a ajustar para cada uno de los hiperparámetros.

Los modelos propuestos y sus mallas de hiperparámetros se describen a continuación agrupados según la **familia de modelos** a la que pertenecen - al compartir todos los modelos de una misma familia los mismos hiperparámetros a ajustar, salvo excepciones.

Modelos de regresión lineal

Los modelos de regresión lineal son la familia de algoritmos más sencilla utilizada, planteados para servir como ***baselines*** - resultados a utilizar como punto de referencia para estudiar la mejora del error en algoritmos más complejos. Concretamente, se han planteado los siguientes modelos - siendo los hiperparámetros asociados los descritos en la **Tabla 5.1**:

- Regresión lineal simple.
- Regresión con regularización *Ridge* (L2).
- Regresión con regularización *Lasso* (L1).
- Regresión con *Elastic-Net*.

Hiperparámetro	Rango	Descripción	Aplicable a
alfa	$\{10^{-6}, 10^{-5}, \ldots, 10^{6}\}$	Factor de penalización aplicado a los parámetros. Valores más altos implican una mayor penalización.	L1, L2, Elastic-Net
ratio	$\{0.25, 0.5, 0.75\}$	Ratio en el que se aplica la penalización de Lasso - L1. 0 significa un modelo Ridge (L2), 1 significa un modelo Lasso (L1)	Elastic-Net

Tabla 5.1: Hiperparámetros planteados para modelos de regresión lineal

Árboles de decisión

Solo se estudia un modelo de árbol, pudiendo observarse los hiperparámetros planteados en la **Tabla 5.2**.

Hiperparámetro	Rango	Descripción
max_depth	$\{1, 2, \ldots, 10, None\}$	Profundidad máxima permitida para el crecimiento del árbol. "None" indica que no se limita la profundidad.
min_samples_split	$\{2, 3, \ldots, 50\}$	Número mínimo de instancias requerido para particionar un nodo, creando hojas a partir de éste.
min_samples_leaf	$\{1, 2, \ldots, 50\}$	Número mínimo de instancias en las hojas resultantes de una partición. Un nodo no puede particionarse si alguna de las hojas resultantes tiene un número menor de instancias.
criterion	{"squared_error", "friedman_mse", "absolute_error}	Función de error utilizada para elegir la partición más valiosa.

Tabla 5.2: Hiperparámetros planteados para árboles de decisión

Máquinas de vector de soporte

Se han planteado **cuatro** modelos de máquinas de vector de soporte, según la **función *kernel*** utilizada - siendo los hiperparámetros asociados a dichas máquinas los descritos en la **Tabla 5.3**.

- *Kernel* lineal.
- *Kernel* polinómico.
- *Kernel* gaussiano.
- *Kernel* sigmoide.

Hiperparámetro	Rango	Descripción	Aplicable a
epsilon	$[10^{-3}, 1]$	Margen de error del hiperplano de máxima confianza. Los puntos a una distancia menor a ϵ del hiperplano se predicen con el valor del hiperplano directamente.	Todos
tol	$[10^{-7}, 1]$	Tolerancia durante el entrenamiento. Si la mejora en el error no es superior a tol, se detiene el entrenamiento.	Todos
C	$[10^{-2}, 10^{3}]$	Parámetro de regularización Ridge - L2 para penalizar pesos elevados. La fuerza de regularización es inversamente proporcional a C - valores más bajos suponen regularizaciones más altas.	Todos
degree	$\{1, 2, \ldots, 6\}$	Grado del polinomio utilizado para definir el hiperplano.	SVR polinómica

Tabla 5.3: Hiperparámetros planteados para máquinas de vector de soporte

Ensembles de Bagging

Se han planteado **dos** *ensembles* de tipo *bagging*, utilizando **árboles de decisiones** como modelos sencillos a agrupar:

- *Random Forests*.
- *Extremely Randomized Trees*.

Los hiperparámetros de los modelos se encuentran descritos en la **Tabla 5.4**:

Hiperparámetro	Rango	Descripción	Aplicable a
n_estimators	$\{50, 51, \ldots, 200\}$	Número de árboles a entrenar en el ensemble.	Todos
max_features	$[0.3, 1.0]$	Porcentaje de atributos muestreados para el entrenamiento de cada árbol.	Todos
max_depth	$\{1, 2, \ldots, 50\}$	Profundidad máxima para cada árbol entrenado.	Todos
min_samples_split	$\{2, 3, \ldots, 50\}$	Número mínimo de instancias requerido para particionar un nodo en cada árbol entrenado.	Todos

Tabla 5.4: Hiperparámetros planteados para ensembles de bagging

Ensembles de Boosting

Se ha planteado un único modelo de *boosting* — ***AdaBoost*** —, utilizando **árboles de decisiones** como modelo sencillo a agrupar. Sus hiperparámetros se encuentran descritos en la **Tabla 5.5**.

Hiperparámetro	Rango	Descripción
n_estimators	$\{50, 51, \ldots, 200\}$	Número de árboles a entrenar en el ensemble.
learning_rate	$[10^{-4}, 10]$	Ponderación aplicada a cada modelo nuevo. En general, representa la velocidad de aprendizaje del modelo, donde valores altos implican cambios más rápidos de los pesos.

Tabla 5.5: Hiperparámetros planteados para ensembles de boosting

Ensembles de Gradient Boosting

Se han planteado **cuatro** ensembles con la técnica de *Gradient Boosting*, utilizando **árboles de decisiones** como modelos sencillos a agrupar - siendo sus hiperparámetros comunes los descritos en la **Tabla 5.6**.

- *eXtreme Gradient Boosting*.
- *Categorical Boosting*.
- *Light Gradient Boosting Machine*.
- *Histogram Based Gradient Boosting*.

A diferencia del resto de modelos propuestos, estos modelos tienen algunas peculiaridades:

Hiperparámetro	Rango	Descripción
Número de estimadores	$\{50, 51, \ldots, 200\}$	Número de árboles a entrenar en el ensemble.
Profundidad máxima	$\{1, 2, \ldots, 10\}]$	Profundidad máxima para cada arbol entrenado.
Tasa de aprendizaje	$[0.01, 1.0]$	Ponderación aplicada a cada modelo nuevo. Velocidad de aprendizaje del modelo, donde valores altos implican cambios más rápidos en los pesos.

Tabla 5.6: Hiperparámetros comunes a los modelos de Gradient Boosting

- **Librerías externas:** A diferencia del resto de modelos - trabajando siempre con implementaciones de *scikit-learn* -, los modelos de *Gradient Boosting* están disponibles a través de librerías independientes. Esto se traduce en que **presentan hiperparámetros y ajustes distintos entre sí**.
- **GPU:** Las implementaciones de estos modelos están diseñadas para agilizar el entrenamiento a través de una **tarjeta gráfica** - lo que puede llegar a hacerlos más rápidos que otros modelos más simples.
- **Atributos categóricos:** Los modelos de *Gradient Boosting* trabajan de forma autónoma con atributos categóricos, por lo que **no es necesario realizar codificación**.

Debido a estas diferencias, es necesario realizar algunos ajustes adicionales de hiperparámetros específicos para cada modelo. Estos hiperparámetros adicionales quedan descritos en la **Tabla 5.7**.

Hiperparámetro	Rango	Descripción	Modelo
Profundidad máxima	$\{4, 5, \ldots, 10\}$	Profundidad máxima para cada árbol entrenado.	XGBoost
Umbral de mejora	$[0, 10^4]$	Error mínimo a reducir para considerar la partición de un nodo.	XGBoost
Porcentaje de instancias	$[0.3, 1.0]$	Porcentaje de instancias del conjunto de datos muestreadas.	XGBoost
Porcentaje de atributos	$[0.3, 1.0]$	Porcentaje de atributos del conjunto de datos muestreados.	XGBoost, HistGradientBoost
Regularización	$[10^{-3}, 10]$	Coeficiente aplicado a la regularización de tipo Ridge (L2) para penalizar ensembles con pesos elevados.	CatBoost
Intensidad de la aleatoriedad	$[1, 2]$	Multiplicador aplicado a la varianza de cada posible partición para forzar aleatoriedad.	CatBoost
Número máximo de hojas	$\{10, 11, \ldots, 100\}$	Número máximo de hojas que puede tener cada árbol - independientemente de su profundidad.	LGBM, HistGradientBoost
Número mínimo de instancias por hoja	$\{20, 21, \ldots, 200\}$	Número mínimo de instancias en las hojas resultantes de una partición.	LGBM, HistGradientBoost

Tabla 5.7: Hiperparámetros específicos para modelos de Gradient Boosting

5.1.2. Experimentación

Tras la definición de modelos, el siguiente paso es definir la **experimentación** a realizar: el proceso guiado de búsqueda a través del cual se realiza el **ajuste de hiperparámetros** y **selección de atributos** para cada modelo y la **selección del modelo final** para resolver el problema de regresión.

En todos los casos, la selección se hace con el objetivo de **minimizar el error del modelo** - es decir, se eligen los hiperparámetros y el modelo que llevan al menor error posible de todas las opciones. Para todos los experimentos, la métrica de error utilizada es la **raíz del**

error cuadrático medio (*RMSE*), siguiendo la fórmula:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(y^{(n)} - \hat{y}^{(n)})^2}$$

Donde $y^{(n)}$ es el valor esperado y $\hat{y}^{(n)}$ es el valor predicho para la instancia n del conjunto de datos.

Para realizar los experimentos se cuentan con los siguientes **conjuntos de datos**, cuyo uso será descrito en las secciones posteriores:

- **Conjunto de entrenamiento:** 9879 instancias, obtenido a partir del 75% del conjunto de datos inicial.
- **Conjunto de validación:** 3294 instancias, obtenido a partir del 25% del conjunto de datos inicial.
- **Conjunto de test:** 5646 instancias, ofrecido por separado al conjunto de datos inicial. **No se tienen los valores esperados para el tiempo de diagnóstico** - teniendo que evaluarse el error a través de una plataforma externa.

Ajuste de hiperparámetros

La primera fase de experimentación es el **ajuste de hiperparámetros:** la selección, para cada modelo propuesto, tanto de los **hiperparámetros** como del **subconjunto de atributos** que minimizan el error final del modelo.

Dicho ajuste se realiza a través de una **selección de hiperparámetros con validación cruzada** - se entrena un modelo para cada par de hiperparámetros y subconjunto de atributos sobre el conjunto de entrenamiento utilizando **validación cruzada de 5-*folds*** para obtener un error promedio honesto. Ahora bien, como se ha visto en la sección anterior, **el número de posibles hiperparámetros para cada modelo puede resultar excesivo** - lo cual, mezclado con el tiempo de entrenamiento de los modelos más complejos, puede hacer la evaluación de todas las combinaciones de hiperparámetros en un tiempo razonable imposible.

Para solucionar este problema se plantean varias metodologías de **búsqueda de hiperparámetros** - es decir, distintas estrategias para determinar **cuáles de todos los posibles conjuntos de hiperparámetros propuestos se evalúan**:

- **Búsqueda exhaustiva:** Se prueban **todas las posibles combinaciones de hiperparámetros**. Si bien ofrece la garantía de encontrar una solución óptima, también acarrea un coste exponencial que deja de ser factible para modelos complejos.
- **Búsqueda aleatoria uniforme:** Para evitar una búsqueda exhaustiva, la primera opción es realizar en su lugar una **búsqueda totalmente aleatoria**, evaluando un número fijo de combinaciones aleatorias para seleccionar el mejor. Si bien es sustancialmente más rápido y suele dar resultados aceptables para un número suficiente de búsquedas, se pierde la garantía de encontrar una solución óptima.

  Para los modelos propuestos, se ha elegido de forma empírica evaluar **100 conjuntos de hiperparámetros aleatorios**.

- **Búsqueda aleatoria Gaussiana:** La búsqueda aleatoria uniforme puede tener problemas al trabajar con modelos muy complejos con tiempos largo de entrenamiento, al tener que probar un gran número de hiperparámetros para encontrar un modelo razonable. Una solución a este algoritmo es guiar las búsquedas aleatorias con un modelo **probabilístico**, donde los hiperparámetros con menor error son más probables. De esta forma se puede reducir el número necesario de conjuntos de hiperparámetros evaluados para obtener un buen resultado.

  Para los modelos propuestos, se ha elegido de forma empírica evaluar **50 conjuntos de hiperparámetros aleatorios**.

	Regresión Lineal	Ridge L2	Lasso L1	Elastic-Net	Árbol de decisión	SVR Lineal	SVR Polinomica	SVR Gaussiana	SVR Sigmoide	Random Forest	Extremely Random Trees	Ada Boost	XGBoost	CatBoost	LGBM	HistGradient Boost
Tipo de búsqueda	Exhaustiva	Exhaustiva	Exhaustiva	Exhaustiva	Aleatoria	Aleatoria	Aleatoria	Aleatoria	Aleatoria	Gaussiana	Gaussiana	Gaussiana	Gaussiana	Gaussiana	Gaussiana	Gaussiana
Conjuntos estudiados	7	91	91	273	100	100	100	100	100	50	50	50	50	50	50	50
Modelos totales entrenados	140	1820	1820	5460	2000	2000	2000	2000	2000	1000	1000	1000	1000	1000	1000	1000

Tabla 5.8: Número total de conjuntos de hiperparámetros y modelos entrenados para cada modelo.

En la **Tabla 5.8** se encuentra detallado el **número total de modelos entrenados** para cada modelo en base al tipo de búsqueda realizado, siguiendo el siguiente orden:

1. **Subconjunto de atributos:** Uno de los hiperparámetros a elegir para cada modelo es el **subconjunto de atributos utilizado** - optando entre los tres subconjuntos propuestos o el conjunto de atributos completo. Por tanto, es necesario realizar la búsqueda **4 veces** para cada modelo.
2. **Ajuste de hiperparámetros:** Dependiendo del tipo de búsqueda realizado, el número de conjuntos de hiperparámetros a evaluar cambia. Hay que destacar que un hiperparámetro a ajustar por todos los modelos es el **agrupamiento de valores durante la codificación** - siendo especialmente significativo en la búsqueda exhaustiva, donde crece el número de conjuntos a evaluar.
3. **Validación cruzada:** Para cada par de subconjuntos de atributos e hiperparámetros se realiza una **5-validación cruzada** entrenando sobre el **conjunto de entrenamiento**, por lo que se entrenan cinco modelos y se obtiene el error promedio de éstos.

Tras todos los modelos entrenados y evaluados, se elige para cada algoritmo **un único modelo final** - aquel que minimiza el error - para ser considerado durante el proceso de selección de modelos.

Selección de modelos

Tras el paso anterior de ajuste de hiperparámetros se ha obtenido un total de **16 modelos ajustados y entrenados** - uno por cada algoritmo considerado. El objetivo de la siguiente y última fase de la experimentación, por tanto, es la **selección del modelo final** - elegir el modelo con menor error **generalizable** a datos no conocidos previamente.

Con el fin de obtener una evaluación **honesta** del rendimiento de cada modelo, cada uno de estos modelos entrenados es evaluado sobre el **conjunto de validación** - un conjunto de

datos al que no han tenido acceso hasta este momento.

El modelo final seleccionado es aquel que presenta el **menor error sobre el conjunto de validación**, siendo el modelo que se utilizará para resolver el problema de predicción de tiempo de diagnóstico de cáncer de metástasis, y cuyo rendimiento real se evaluará sobre el **conjunto de test**.

5.2. Evaluación

Tras el modelado y experimentación, el siguiente paso del ciclo de ciencia de datos es la **evaluación:** el estudio del **rendimiento real del modelo seleccionado**, para determinar si es capaz de resolver el problema propuesto - la predicción del tiempo de diagnóstico - con un **error aceptable**.

Además de este estudio del modelo final, resulta de interés académico estudiar el **rendimiento** de los modelos entrenados durante la experimentación estudiando estadísticos como el **error** o el **tiempo de entrenamiento** durante las fases de ajuste de hiperparámetros y selección de modelos.

5.2.1. Rendimiento durante el ajuste de hiperparámetros

El primer paso de la evaluación consiste en estudiar los resultados del ajuste de hiperparámetros - concretamente observar el **error y tiempo de entrenamiento** de los modelos, y estudiar como puede influir en éste los **subconjuntos de atributos**.

Los resultados completos de la fase de ajuste de hiperparámetros están disponibles en la **Tabla A.1** - realizándose el siguiente estudio a partir de gráficas.

Error de los modelos durante el ajuste

La **Figura 5.1** representa la distribución del **menor error** para cada par de modelo y subconjunto de atributos - donde un menor error representa un **mejor modelo** - incluyendo, además, los **valores promedios** de cada subconjunto de atributos a través de lineas discontinuas. A partir de esta figura, se pueden realizar las siguientes observaciones:

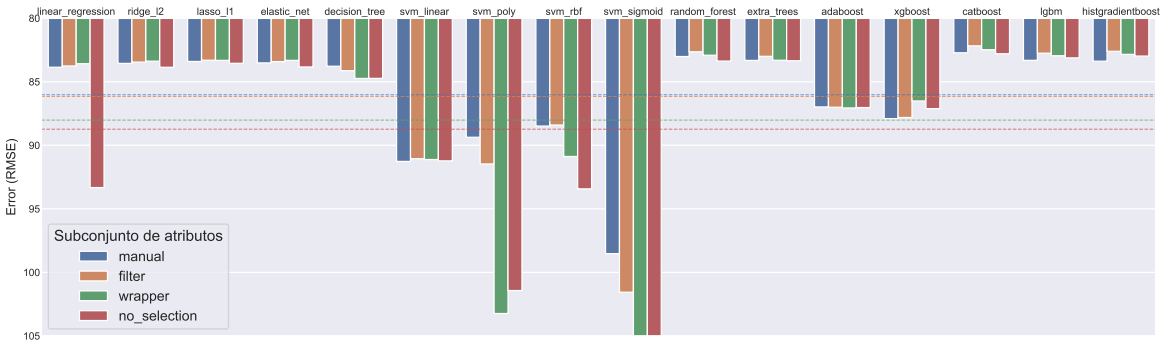


Figura 5.1: Error promedio durante el entrenamiento para cada modelo y subconjunto de atributos (acotado en un error máximo de 105)

- En general, **todos los modelos presentan errores similares y elevados** - en el rango de $82 - 84$ unidades de error -, habiendo cierta variación dependiendo del subconjunto de atributos. También se cumple, como se esperaba, que **el error se reduce con la complejidad de los modelos**. Ahora bien, hay unos modelos comportándose como *outliers*:
  - **Máquinas de vector de soporte:** En todas las ocasiones las máquinas de vector de soporte ofrecen errores sustancialmente peores al del resto de modelos, siendo especialmente significativo para las **máquinas sigmoides**.
  - ***AdaBoost y XGBoost*:** A pesar de ser modelos complejos - *ensembles* de *Boosting* y *Gradient Boosting* respectivamente -, sus errores son mayores al promedio del resto de modelos.
- Se pueden observar **tres agrupamientos de subconjuntos de atributos**, con promedios de error similares:
  - **Manual y *Filter***: Los subconjuntos con menor error promedio, ambos presentan comportamientos y atributos muy similares - siendo la principal diferencia que *filter* añade información numérica y ofrece ligeramente mejores resultados, especialmente en modelos **complejos** como *ensembles*.
  - ***Wrapper***: Un subconjunto con menos atributos categóricos y más atributos numéricos, parece ofrecer los mejores resultados en los modelos de regresión lineal - los más simples y susceptibles a la alta dimensionalidad que introducen los atributos categóricos.
  - **Conjunto completo:** Como era de esperar, el error tiende a ser mayor utilizando el conjunto completo de 150 atributos - presentando picos sustanciales en algunos modelos .

  Es importante destacar que los promedios están **sesgados** por los errores elevados de las máquinas de vectores de soporte - lo que puede influir en que sean más elevados de lo que serían realmente.

Tiempo de entrenamiento de los modelos durante el ajuste

La **Figura 5.2** representa el **tiempo de entrenamiento** para el modelo entrenado, para cada par de modelo y subconjunto de atributos. A partir de esta figura, se pueden realizar las siguientes observaciones:

- Como era de esperar, **la dimensionalidad del conjunto de datos aumenta el tiempo de entrenamiento** - los subconjuntos de atributos reducidos son, por lo general, sustancialmente más rápidos que los modelos entrenados sobre el conjunto de datos completo.

  Dentro de los subconjuntos, en general *Filter* y *Wrapper* son más lentos, al tener **10 atributos** frente a los **5** de *Manual*. Ahora bien, en algunos modelos parece ser más rápido *Wrapper*, posiblemente por el menor número de atributos categóricos.

- Por lo general, **la complejidad del modelo influye en el tiempo de entrenamiento** - con los modelos tradicionales más simples necesitando menos tiempo que los *ensembles.*

  Las únicas excepciones se encuentran en las **máquinas de vectores de soporte** - que requieren un gran tiempo para aplicar funciones kernel complejas - y los ***ensembles de Gradient Boosting*** - que, en contra de lo que se podría esperar por su gran complejidad, son

de los modelos más rápidos al utilizar una GPU para paralelizar su entrenamiento.

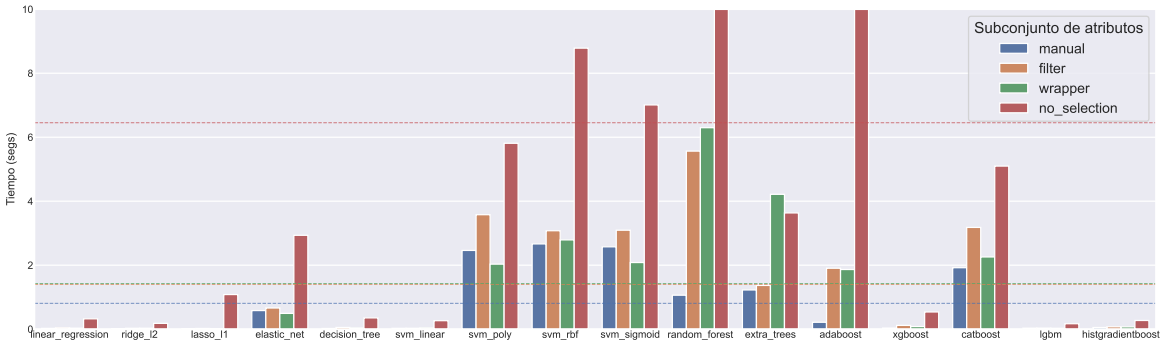


Figura 5.2: Tiempo promedio de entrenamiento para cada modelo y subconjunto de atributos (acotado en 10 segundos)

5.2.2. Rendimiento durante la selección de modelos

Tras el estudio del rendimiento durante el ajuste, el siguiente paso consiste en analizar el rendimiento de los **modelos entrenados** - concretamente estudiar los **subconjuntos de atributos seleccionados** y los **errores finales de los modelos ajustados sobre el conjunto de validación**.

Los resultados completos de la fase de selección de modelo están disponibles en la **Tabla A.2** - realizándose el siguiente estudio a partir de gráficas.

Subconjuntos de hiperparámetros utilizados

En la **Figura 5.3** se representa la distribución de los subconjuntos de atributos seleccionados durante el ajuste de hiperparámetros. Se pueden observar las siguientes características en el comportamiento:

(a) General **(b)** Agrupado por familias

Figura 5.3: Distribución de los subconjuntos de atributos seleccionados

- Los modelos tienden a utilizar subconjuntos con **atributos numéricos** - *Filter y Wrapper* frente a *Manual*. En contra de lo que se podría haber deducido a partir del análisis exploratorio, los **atributos numéricos aportan información util al modelo**. Concretamente:

- Los **modelos lineales** suelen utilizar *Wrapper*, un subconjunto con menos atributos categóricos.
- Los **árboles de decisiones** y **máquinas de vector de soporte** utilizan *Manual* con mayor frecuencia - posiblemente influyendo el ser el subconjunto más pequeño.
- Los ***ensembles***, más complejos, tienden a preferir *Filter* - un subconjunto con la misma información categórica que *Manual* pero información numérica adicional.

- **Ningún modelo utiliza el conjunto de atributos completo** - por lo que se puede afirmar que la selección de atributos ha mejorado el rendimiento.

Error de los modelos seleccionados

La **Figura 5.4** representa el error sobre el **conjunto de validación** de cada modelo de regresión entrenado - ordenados de **menor a mayor error**. A partir de dicha gráfica, se pueden extraer las siguientes conclusiones:

- **El error generalizado de los modelos es muy elevado:** Incluso para el mejor modelo - *Categorical Boost*, con un error de **81.61** - el error es muy elevado si se tiene en cuenta que el rango de la variable objetivo en el conjunto de entrenamiento es aproximadamente $[0, 365]$.
  Al ser este error generalizado, lo más factible es asumir que **el conjunto de datos no tiene suficiente información para predecir esta información**.
- **Como era de esperar, los *ensembles* ofrecen mejor rendimiento:** Los modelos con mejor rendimiento son los *ensembles* - concretamente ***Gradient Boosting*** seguido de ***Bagging***. Sorprendentemente, los modelos de regresión lineal, en principio incluidos como *baselines*, ofrecen rendimientos **superiores** al resto de modelos propuestos.
- **Algunos modelos ofrecen rendimientos anómalos:**
  - ***XGBoost* y *AdaBoost***: Pese a ser ensembles, el error es superior al de un único árbol de decisión. Esto puede deberse a problemas a la hora de ajustar hiperparámetros.
  - **Máquinas de vector de soporte:** El error de estos modelos es sustancialmente superior al de todos los demás modelos. Sumado a su elevado tiempo de entrenamiento, hace que esta familia de modelos no sea útil para el problema propuesto.

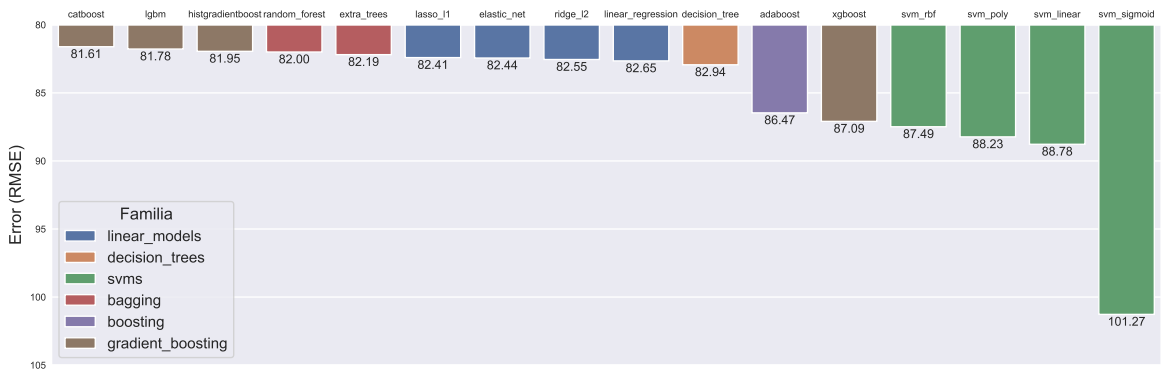


Figura 5.4: Distribución del error de los modelos entrenados

5.2.3. Rendimiento del modelo final

El último paso de la evaluación es el estudio del **rendimiento del modelo seleccionado final** - estudiando los hiperparámetros seleccionados, comprobando su **error real** sobre el conjunto de test disponible en *Kaggle* y evaluando la posición hipotética que hubiera tenido durante la competición.

Modelo seleccionado

Tras el proceso de selección de modelo, el modelo de regresión seleccionado ha sido ***Categorical Boost*** - un *ensemble de* ***Gradient Boosting*** utilizado como estado del arte actualmente en numerosos problemas estructurados. A través del ajuste de hiperparámetros se han seleccionado los siguientes parámetros:

- **Subconjunto de atributos:** *Filter* - **10 atributos** incluyendo los **5 atributos categóricos** seleccionados durante el análisis exploratorio de datos y **5 atributos numéricos** adicionales con información socioeconómica.
- **Modelos contenidos:** El *ensemble* está formado por **200 árboles** (*iterations*) con una profundidad máxima acotada de **8** (*max_depth*).
- **Ratio de aprendizaje (*learning_rate*):** 0.05 - el aprendizaje es lento y se da poco peso a los árboles posteriores, pero sigue siendo más elevado que el valor por defecto $(0.05 > 0.03)$.
- **Regularización (*l2_leaf_reg*):** 5.64 - más elevada que el valor por defecto $(5.36 > 3.00)$, se penalizan los modelos complejos.
- **Aleatorización (*random_strength*):** 2 - aleatoriedad añadida a los umbrales a la hora de elegir particiones para los nodos, se opta por introducir un mayor grado de aleatoriedad para tener árboles más diversos.

Resultados sobre el conjunto de test y posición en la competición

En la **Tabla 5.9** se puede observar el **error del modelo seleccionado** en los dos conjuntos de test disponibles: **público** (disponible durante la competición) y **privado** (mostrado tras finalizar la competición).

Además, se muestra la posición hipotética que hubiera tenido el modelo desarrollado a lo largo de este trabajo si se hubiera inscrito en la competición - junto a una comparativa del error respecto al modelo ganador de la competición y al modelo en la posición mediana.

Conjunto de test	Error	Posición	Primer puesto		Mediana (puesto 271)	
			Error	Diferencia	Error	Diferencia
Público	**83,093**	249 / 542	82,170	+0,92	83,280	-0,187
Privado	**81,032**	199 / 542	80,410	+0,62	81,447	-0,415

Tabla 5.9: Error del modelo seleccionado en el conjunto de test

A partir de esta información se puede determinar que:

- **Sin sobreajuste:** El error del modelo se ha mantenido estable para todos los conjuntos de datos estudiados - entrenamiento, validación y test. Esto implica que no se ha sobreajustado el modelo a los datos en ningún punto, y que sus resultados son generalizables a nuevos datos.
- **Resultados razonables:** Si bien el modelo tiene una posición en el ranking mediocre (superior a la mediana pero lejana del primer puesto), **el error del modelo es muy cercano al del ganador de la competición** - con una diferencia inferior a 1 punto en ambos conjuntos de datos.
- **Error alto generalizado:** Tal y como se observó durante la selección de modelos, **el error general de los modelos es muy elevado** - siendo prácticamente equivalente a un tercio del rango de la variable objetivo, $[0, 365]$.

Por todo esto, se puede afirmar que **el modelo entrenado y seleccionado es un buen predictor para el problema a resolver** - especialmente si se considera el error alto generalizado -, estando listo para ser **desplegado** a través de una aplicación.

6. Despliegue y aplicación web

Tras los pasos realizados del ciclo de vida de la ciencia de datos - análisis exploratorio, preprocesamiento, modelado y evaluación -, en este capítulo se describe el resultado del último paso: el **despliegue** del modelo final entrenado obtenido para su uso por los usuarios.

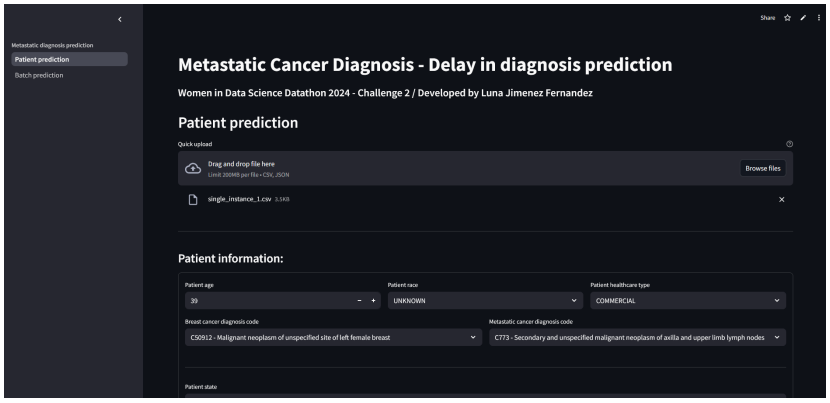


Figura 6.1: Página principal de la aplicación web

El despliegue se ha realizado a través de una **aplicación web**, cuyo aspecto se puede observar en la **Figura 6.1**. Esta aplicación es un **producto mínimo viable** desarrollado a través de *Streamlit* - una librería de Python para prototipado de aplicaciones-, funcionando mediante un **modelo embebido** incrustado directamente en la página web.

Esta aplicación se encuentra disponible en el enlace `https://cidaen-m5-thesis-lunajimenezfernandez.streamlit.app/`, teniendo **dos** funcionalidades principales.

6.1. Aplicación para usuario - predicción individual

La funcionalidad principal de la aplicación es la **predicción individual del tiempo de diagnóstico**, como se puede observar en la **Figura 6.2**.

La aplicación permite introducir manualmente los valores de los **10 atributos** para obtener de forma automática una predicción del tiempo de diagnóstico de cáncer metastásico

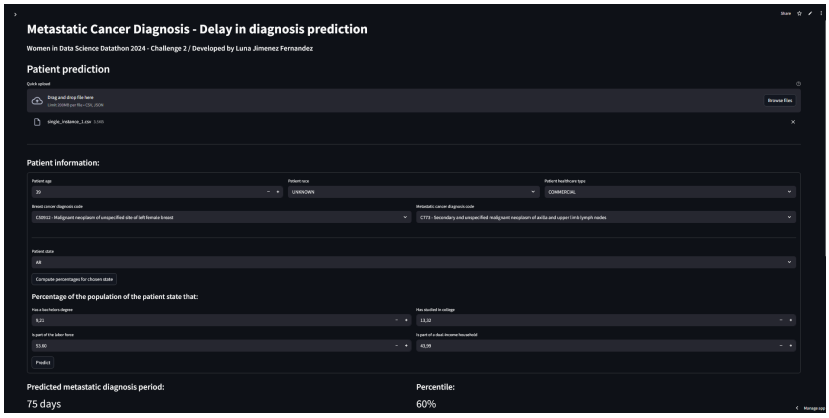


Figura 6.2: Predicción de tiempo de diagnóstico individual

para un paciente. Para simplificar la tarea de introducción de datos, se ofrecen las siguientes facilidades:

- Se incluye una **descripción completa** de todos los códigos de diagnóstico y la posibilidad de buscar por descripción.
- Se incluye opción para introducir automáticamente toda la información del paciente a partir de un fichero *JSON* o *CSV*.
- No es factible asumir que un doctor va a conocer la información socioeconómica de la región de sus pacientes, por los que se incluye un botón **"Compute percentages for chosen state"** que calcula automáticamente los valores para el estado actual.

6.2. Aplicación *batch* - predicción en bloque

La segunda funcionalidad ofrecida por la aplicación es la **predicción de tiempos de diagnóstico en bloque**, como se puede observar en la **Figura 6.3**.

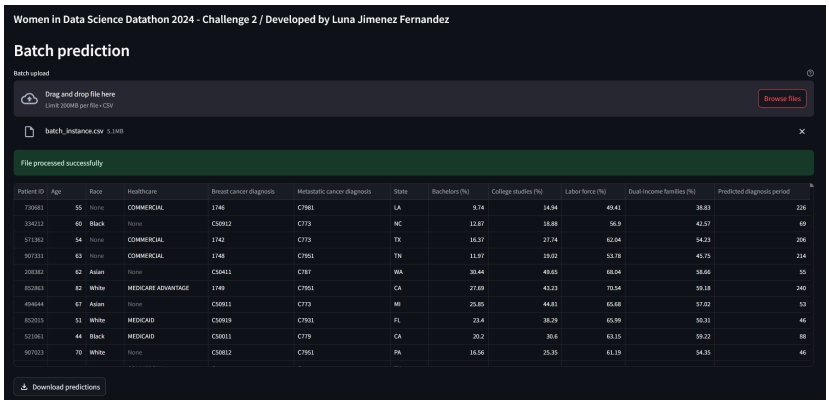


Figura 6.3: Predicción de tiempo de diagnóstico en bloque

El funcionamiento está más enfocado a la automatización del diagnóstico de un gran número de pacientes - introduciendo un **fichero CSV** con los datos clínicos relevantes y generando de forma automática la predicción para todos los pacientes. Esta predicción puede ser descargada posteriormente a través del botón **"Download Predictions"**.

7. Conclusiones

A lo largo del trabajo realizado y de la memoria, se ha descrito en detalle el **ciclo de vida de la ciencia de datos** llevado a cabo para resolver el problema planteado: la **predicción del tiempo de diagnóstico de cáncer metastásico**, en base a datos médicos, geográficos, socioeconómicos y climáticos.

El primer paso ha sido un **análisis exploratorio de datos**, para conocer y entender mejor el comportamiento del conjunto de datos - lo que ha permitido realizar un **preprocesamiento** posterior en el que se han seleccionando atributos y transformando los datos para su uso durante el **modelado**, donde se han planteado una serie de posibles modelos para solucionar el problema.

El mejor de dichos modelos ha sido seleccionado a través de una **evaluación** de su rendimiento en comparación con el resto de modelos y con los ganadores de la competición. Finalmente, este modelo se ha dejado disponible a los usuarios finales a través del **despliegue** de una aplicación web.

Tras el trabajo realizado, se puede afirmar que el **primer objetivo** - la creación de un modelo de regresión capaz de predecir el tiempo de diagnóstico - se ha **cumplido con éxito** dentro de las posibilidades:

- El modelo entrenado **presenta un error elevado** - alrededor de **82 días** en promedio -, si bien dicho error se encuentra cercano al error de los modelos ganadores de la competición.
- El funcionamiento del modelo es **ágil** - en el orden de los segundos para el entrenamiento y de los milisegundos para la predicción -, lo que lo hace **utilizable en tiempo real**.
- La **aplicación web** permite el uso simple del modelo por parte de los usuarios finales.

Respecto al **segundo objetivo** - el estudio de la influencia de factores geográficos, socioeconómicos y climáticos -, se han extraído las siguientes conclusiones:

- El análisis exploratorio de datos indica que **solo la información médica es relevante para la predicción del tiempo** - los atributos socioeconómicos y climáticos **no presentan correlación clara con el tiempo de diagnóstico**, y el único atributo geográfico con cierta relevancia es el **estado de residencia del paciente**.

- Pese a esto, **el modelo final seleccionado incluye información socioeconómica** - en la forma de información estadística sobre los estudios universitarios y los ingresos de las familias. Ahora bien, en la práctica **dichas variables no tienen apenas influencia sobre el tiempo predicho**.

Actualmente, el trabajo completo se encuentra disponible en un repositorio público de *Github* (`https://github.com/MoonDollLuna/cidaen_m5_thesis`) bajo licencia *MIT*, incluyendo tanto la memoria como el **código Python** utilizado para realizar todos los pasos del proceso de ciencia de datos.

Además, como ya se ha comentado en el Capítulo 6, la aplicación web a través de la que se ha desplegado el modelo se encuentra disponible en el enlace `https://cidaen-m5-thesis-lunajimenezfernandez.streamlit.app/`.

7.1. Trabajo futuro

El trabajo propuesto fue concebido como parte de una competición ya clausurada, por lo que no tiene sentido continuarlo directamente. Ahora bien, sí es posible plantear una serie de **ampliaciones al proceso realizado**, de cara a futuros trabajos de ciencia de datos:

- **Ampliación del número de modelos:** Aun utilizando los modelos más comunes para problemas de ciencia de datos estructurados, es posible utilizar otra serie de modelos más complejos - como pueden ser las **redes neuronales** o los ***stacks*** de *ensembles*.
- **Creación de atributos:** Si bien se ha realizado un pre-procesamiento de los datos, podría haber resultado de interés añadir **atributos sintéticos** para ampliar la información útil contenida en el conjunto de datos.
- **Modelo disponible a través de cliente - servidor:** Actualmente, el modelo se encuentra embebido en la aplicación web - algo factible debido al tamaño ligero del modelo entrenado, pero que se haría imposible para modelos más complejos o páginas web más concurridas.

  Para solucionar ésto, sería razonable crear una **API (Application Web Interface)** a través de la cual se llamaría al modelo, para hospedarlo en un servidor separado independiente al funcionamiento de la propia aplicación.

Bibliografía

[1] Women in Data Science, *WiDS Datathon 2024 Challenge 2*, 2024.

[2] D. D. and, "50 Years of Data Science", *Journal of Computational and Graphical Statistics*, vol. 26, nº 4, págs. 745-766, 2017.

[3] V. Dhar, "Data science and prediction", *Commun. ACM*, vol. 56, nº 12, págs. 64-73, dic. de 2013.

[4] F. Berman et al., "Realizing the potential of data science", *Commun. ACM*, vol. 61, nº 4, págs. 67-72, mar. de 2018.

[5] V. Stodden, "The data science life cycle: a disciplined approach to advancing data science as a science", *Commun. ACM*, vol. 63, nº 7, págs. 58-66, jun. de 2020.

[6] J. M. Wing, "The Data Life Cycle", *Harvard Data Science Review*, vol. 1, nº 1, jul. de 2019, https://hdsr.mitpress.mit.edu/pub/577rq08d.

[7] M. Komorowski, D. Marshall, J. Salciccioli e Y. Crutain, "Exploratory Data Analysis", en sep. de 2016, págs. 185-203.

[8] C. Shearer, "The CRISP-DM model: the new blueprint for data mining", *Journal of data warehousing*, vol. 5, nº 4, págs. 13-22, 2000.

[9] J. Saltz, *CRISP-DM is Still the Most Popular Framework for Executing Data Science Projects - Data Science PM — datascience-pm.com*, `https://www.datascience-pm.com/crisp-dm-still-most-popular/`, [Accessed 28-05-2025], 2024.

[10] T. M. Mitchell, *Machine learning*. McGraw-hill New York, 1997, vol. 1.

[11] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[12] S. Russell y P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd. USA: Prentice Hall Press, 2009.

[13] A. Burkov, *The Hundred-Page Machine Learning Book*. 2019.

[14] Y. Tai, *A Survey Of Regression Algorithms And Connections With Deep Learning*, 2021.

[15] A. Y. Ng, “Feature selection, L1 vs. L2 regularization, and rotational invariance”, en *Proceedings of the Twenty-First International Conference on Machine Learning*, ép. ICML ’04, Banff, Alberta, Canada: Association for Computing Machinery, 2004, pág. 78.

[16] H. Zou y T. Hastie, “Regularization and Variable Selection Via the Elastic Net”, *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 67, nº 2, págs. 301-320, mar. de 2005.

[17] A. E. Hoerl y R. W. Kennard, “Ridge Regression: Biased Estimation for Nonorthogonal Problems”, *Technometrics*, vol. 42, nº 1, págs. 80-86, 2000.

[18] R. Tibshirani, “Regression Shrinkage and Selection via the Lasso”, *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, nº 1, págs. 267-288, 1996.

[19] A. Smola y B. Schölkopf, “A tutorial on support vector regression”, *Statistics and Computing*, vol. 14, págs. 199-222, ago. de 2004.

[20] L. Breiman, “Bagging predictors”, *Mach. Learn.*, vol. 24, nº 2, págs. 123-140, ago. de 1996.

[21] L. Breiman, “Random Forests”, *Mach. Learn.*, vol. 45, nº 1, págs. 5-32, oct. de 2001.

[22] P. Geurts, D. Ernst y L. Wehenkel, “Extremely randomized trees”, *Machine Learning*, vol. 63, nº 1, págs. 3-42, abr. de 2006.

[23] Y. Freund y R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting”, en *Proceedings of the Second European Conference on Computational Learning Theory*, ép. EuroCOLT ’95, 1995, págs. 23-37.

[24] J. H. Friedman, “Greedy function approximation: A gradient boosting machine.”, *The Annals of Statistics*, vol. 29, nº 5, págs. 1189-1232, 2001.

[25] T. Chen y C. Guestrin, “XGBoost: A Scalable Tree Boosting System”, en *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ép. KDD ’16, ACM, ago. de 2016, págs. 785-794.

[26] A. V. Dorogush, V. Ershov y A. Gulin, *CatBoost: gradient boosting with categorical features support*, 2018.

[27] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush y A. Gulin, *CatBoost: unbiased boosting with categorical features*, 2019.

[28] G. Ke et al., “LightGBM: A Highly Efficient Gradient Boosting Decision Tree”, en *Advances in Neural Information Processing Systems*, I. Guyon et al., eds., vol. 30, Curran Associates, Inc., 2017.

[29] F. Pedregosa et al., “Scikit-learn: Machine Learning in Python”, *Journal of Machine Learning Research*, vol. 12, págs. 2825-2830, 2011.

[30] I. Guyon y A. Elisseeff, “An Introduction of Variable and Feature Selection”, *J. Machine Learning Research Special Issue on Variable and Feature Selection*, vol. 3, págs. 1157-1182, ene. de 2003.

[31] N. Draper y H. Smith, *Applied regression analysis* (Wiley series in probability and mathematical statistics). New York [u.a.]: Wiley, 1966, IX, 407.

A. Resultados

En este anexo se incluyen los resultados completos de la experimentación, tanto los **estadísticos** del entrenamiento y validación de los modelos como los **hiperparámetros seleccionados** para cada uno de los modelos.

A.1. Estadísticos

En esta sección se incluyen los resultados completos de los procesos de entrenamiento y validación presentados en forma de tablas.

A.1.1. Ajuste de hiperparámetros

Modelo	Subconjunto de atributos	Tipo de búsqueda	Iteraciones de búsqueda	Tiempo de búsqueda (segs)			Error promedio
				Total	Por iteración	Final	
linear_regression	manual	grid	7	1.6383299827575684	0.2340471403939383	0.0238795280456542	83.8568112648107
linear_regression	filter	grid	7	0.3313274383544922	0.0473324911934988	0.0416181087493896	83.75820570897277
linear_regression	wrapper	grid	7	0.3210222721099853	0.0458603245871407	0.0307879447937011	83.57752118121913
linear_regression	no_selection	grid	7	1.8199312686920168	0.2599901812417166	0.3231534957885742	93.32674499694888
ridge_l2	manual	grid	91	1.5099420547485352	0.0165927698324014	0.0136852264404296	83.55469321331245
ridge_l2	filter	grid	91	1.9106590747833248	0.0209962535690475	0.0280930995941162	83.45235277991227
ridge_l2	wrapper	grid	91	1.7659635543823242	0.0194061929053002	0.0286035537719726	83.37960870257116
ridge_l2	no_selection	grid	91	10.773196697235107	0.1183867768926934	0.1813464164733886	83.86179717659111
lasso_l1	manual	grid	91	3.995432615280152	0.0439058529151665	0.0164384841918945	83.40286406661048
lasso_l1	filter	grid	91	4.819518566131592	0.0529617424849625	0.0214715003967285	83.30629026098315
lasso_l1	wrapper	grid	91	3.9306271076202393	0.0431937044793432	0.0169079303741455	83.31664928110077
lasso_l1	no_selection	grid	91	53.92193555831909	0.5925487423991109	1.0820424556732178	83.54799847993567
elastic_net	manual	grid	273	11.721351385116575	0.042935353059035	0.5812702178955078	83.51526405439998
elastic_net	filter	grid	273	14.64482307434082	0.0536440405653509	0.6614060401916504	83.41333496392983
elastic_net	wrapper	grid	273	12.51454734802246	0.045840832776639	0.4930276870727539	83.31637514887375
elastic_net	no_selection	grid	273	167.6494746208191	0.6141006396366999	2.934457540512085	83.83496942582158
decision_tree	manual	randomized	100	9.67495584487915	0.0967495584487915	0.0208790302276611	83.7780860558013
decision_tree	filter	randomized	100	20.14728045463562	0.2014728045463562	0.0531642436981201	84.12740067171896
decision_tree	wrapper	randomized	100	19.533833742141724	0.1953383374214172	0.0393133163452148	84.73891963892252
decision_tree	no_selection	randomized	100	229.85245037078855	2.2985245037078856	0.3523120880126953	84.73284059274799
svm_linear	manual	randomized	100	1.4980344772338867	0.0149803447723388	0.0259366035461425	91.26781654784642
svm_linear	filter	randomized	100	1.861642599105835	0.0186164259910583	0.0272941589355468	91.05567296412444
svm_linear	wrapper	randomized	100	3.111152172088623	0.0311115217208862	0.0407259464263916	91.11874517574594
svm_linear	no_selection	randomized	100	97.80935859680176	0.9780935859680177	0.2651453018188476	91.22543953579348
svm_poly	manual	randomized	100	48.84676384925842	0.4884676384925842	2.4611942768096924	89.36720565282685

Modelo	Subconjunto de atributos	Tipo de búsqueda	Iteraciones de búsqueda	Tiempo de búsqueda (segs)			Error promedio
				Total	Por iteración	Final	
svm_poly	filter	randomized	100	54.5289785861969	0.545289785861969	3.5780694484710693	91.46895941664454
svm_poly	wrapper	randomized	100	56.0568106174469	0.560568106174469	2.032531976699829	103.23746378648907
svm_poly	no_selection	randomized	100	644.3820505142212	6.443820505142212	5.810599088668823	101.4302023212575
svm_rbf	manual	randomized	100	63.00397539138794	0.6300397539138793	2.6649532318115234	88.48592337147325
svm_rbf	filter	randomized	100	67.57193088531494	0.6757193088531495	3.077049016952514	88.40458104489666
svm_rbf	wrapper	randomized	100	64.81695699691772	0.6481695699691773	2.79060959815979	90.88068235544974
svm_rbf	no_selection	randomized	100	681.3171155452728	6.8131711554527286	8.784799814224243	93.42802891084526
svm_sigmoid	manual	randomized	100	68.20164012908936	0.6820164012908936	2.5738353729248047	98.52864570703784
svm_sigmoid	filter	randomized	100	60.86087465286255	0.6086087465286255	3.0932424068450928	101.56179050963154
svm_sigmoid	wrapper	randomized	100	57.571099519729614	0.5757109951972962	2.0827889442443848	116.6249476341536
svm_sigmoid	no_selection	randomized	100	686.9058706760406	6.869058706760407	7.009771823883057	114.56071608568432
random_forest	manual	bayes	50	67.10657596588135	1.3421315193176269	1.0617177486419678	83.01366339811862
random_forest	filter	bayes	50	257.291220664978	5.145824413299561	5.565766096115112	82.64006694163541
random_forest	wrapper	bayes	50	143.32606840133667	2.866521368026733	6.297638416290283	82.91317663829959
random_forest	no_selection	bayes	50	1896.0526087284088	37.921052174568175	49.28590178489685	83.37177900561274
extra_trees	manual	bayes	50	70.03935527801514	1.4007871055603027	1.225649118423462	83.32129708620833
extra_trees	filter	bayes	50	127.64828777313232	2.5529657554626466	1.3669829368591309	82.99732955990197
extra_trees	wrapper	bayes	50	107.57072877883913	2.151414575576782	4.215499401092529	83.30860795243186
extra_trees	no_selection	bayes	50	472.1313157081604	9.442626314163208	3.633639335632324	83.34617076279878
adaboost	manual	bayes	50	25.39012169837952	0.5078024339675903	0.215623140335083	86.98043185143206
adaboost	filter	bayes	50	37.90988659858704	0.7581977319717407	1.9048182964324951	87.0056344510263
adaboost	wrapper	bayes	50	75.38741755485535	1.507748351097107	1.864696741104126	87.06043695047784
adaboost	no_selection	bayes	50	688.5746812820435	13.771493625640868	17.499237537384033	87.03001535473632
xgboost	manual	bayes	50	35.20437026023865	0.704087405204773	0.0331051349639892	87.90340747391754
xgboost	filter	bayes	50	49.24621248245239	0.984924249649048	0.1175014972686767	87.81372800884735
xgboost	wrapper	bayes	50	44.32945442199707	0.8865890884399414	0.0812370777130127	86.51321162424438
xgboost	no_selection	bayes	50	196.10614681243896	3.9221229362487793	0.5374188423156738	87.11544175892757
catboost	manual	bayes	50	246.1163387298584	4.922326774597168	1.920884132385254	82.71968370465952
catboost	filter	bayes	50	620.5296370983124	12.410592741966248	3.180443525314331	82.17582430193673
catboost	wrapper	bayes	50	330.7876410484314	6.615752820968628	2.2572925090789795	82.48079419827772
catboost	no_selection	bayes	50	530.1788566112518	10.603577132225036	5.097702026367188	82.79572110039553
lgbm	manual	bayes	50	38.05036544799805	0.7610073089599609	0.0299694538116455	83.314620119142
lgbm	filter	bayes	50	29.31800413131714	0.5863600826263428	0.0324811935424804	82.75239373414163
lgbm	wrapper	bayes	50	29.38965344429016	0.5877930688858032	0.0278058052062988	82.95309627512827
lgbm	no_selection	bayes	50	43.04041337966919	0.8608082675933838	0.1700525283813476	83.11972338375003
histgradientboost	manual	bayes	50	28.465141534805294	0.569302830696106	0.0509958267211914	83.38184375003458
histgradientboost	filter	bayes	50	28.340924501419067	0.5668184900283814	0.0732359886169433	82.60312820261804
histgradientboost	wrapper	bayes	50	26.02744150161743	0.5205488300323486	0.0717401504516601	82.84526131726716
histgradientboost	no_selection	bayes	50	44.81061744689941	0.8962123489379883	0.2714483737945556	82.97741054803666

Tabla A.1: Estadísticos del mejor resultado para cada par de modelo y subconjunto de atributos

A.1.2. Selección de modelos

Modelo	Subconjunto de atributos	Error		Tiempo de entrenamiento (segs)	
		Entrenamiento	Validación	Entrenamiento	Validación
linear_regression	wrapper	83.57752118121913	82.64944468760879	0.030787944793701172	0.0
ridge_l2	wrapper	83.37960870257116	82.54723807247451	0.028603553771972656	0.012679338455200195
lasso_l1	filter	83.30629026098315	82.40722984005937	0.021471500396728516	0.0
elastic_net	wrapper	83.31637514887375	82.43928985039425	0.4930276870727539	0.0
decision_tree	manual	83.7780860558013	82.93619776291433	0.020879030227661133	0.0
svm_linear	filter	91.05567296412445	88.77516348329344	0.027294158935546875	0.017313480377197266
svm_poly	manual	89.36720565282685	88.23183926282347	2.4611942768096924	0.5066883563995361
svm_rbf	filter	88.40458104489666	87.48841351157962	3.0770490169525146	0.6509499549865723
svm_sigmoid	manual	98.52864570703784	101.27207439472834	2.5738353729248047	0.5883753299713135
random_forest	filter	82.64006694163541	81.99680936766894	5.565766096115112	0.030972242355346668
extra_trees	filter	82.99732955990197	82.19038700513607	1.3669829368591309	0.026447296142578125
adaboost	manual	86.98043185143206	86.46795403506125	0.215623140335083	0.0
xgboost	wrapper	86.51321162424438	87.09145865628854	0.0812370777130127	0.01599717140197754
catboost	filter	82.17582430193673	81.60903130606145	3.180443525314331	0.023774147033691406
lgbm	filter	82.75239373414163	81.77821474896575	0.03248119354248047	0.00975489616394043
histgradientboost	filter	82.60312820261804	81.9457649049038	0.07323598861694336	0.01062774658203125

Tabla A.2: Estadísticos del modelo seleccionada para cada modelo

A.2. Hiperparámetros

En esta sección se presentan los hiperparámetros finales seleccionados para cada modelo. Los hiperparámetros seleccionados para cada modelo y subconjunto de atributos no se incluyen, pero están disponibles en los ficheros *CSV* resultantes de la experimentación.

Modelos de regresión lineal

Modelo	Frecuencia mínima (Agrupación one-hot)	Alfa	Ratio L1
Regresión lineal	0	—	—
Ridge (L2)	0	1.0	—
Lasso (L1)	0	0.1	—
Elastic-Net	0	0.001	0.5

Tabla A.3: Hiperparámetros de los modelos de regresión lineal

Árboles de decisión

Modelo	Frecuencia mínima (Agrupación one-hot)	Criterio de partición	Profundidad máxima	Instancias mínimas por hoja	Instancias mínimas para partición
Árbol de decisión	0.002965985859578346	friedman_mse	9	40	33

Tabla A.4: Hiperparámetros de los árboles de decisión

Máquinas de vectores de soporte

Modelo	Frecuencia mínima (Agrupación one-hot)	C	Epsilon	Tolerancia	Grado
SVM lineal	0.005067687245650943	71.0411544538531	0.26633253593242695	0.041821644626507753	—
SVM polinómica	0.0018707496062876693	32.75143592003466	0.030225289960459558	0.07982332010050937	4.0
SVM gaussiana	0.005067687245650943	71.0411544538531	0.26633253593242695	0.041821644626507753	—
SVM sigmoide	0.02969275048082812	1.4982252118587012	0.3582545677649474	0.0001225375937095268	—

Tabla A.5: Hiperparámetros de las máquinas de vectores de soporte

Ensembles de Bagging

Modelo	Frecuencia mínima (Agrupación one-hot)	Profundidad máxima	Porcentaje de atributos	Instancias mínimas para partición	Número de árboles
Random Forests	0.0001	17	0.6474131133913984	50	200
Extremely Random Trees	0.0001	16	0.3	50	106

Tabla A.6: Hiperparámetros de los ensembles de bagging

Ensembles de Boosting

Modelo	Frecuencia mínima (Agrupación one-hot)	Ratio de aprendizaje	Número de árboles
Adaptive Boosting	0.0001	0.0001	50

Tabla A.7: Hiperparámetros de los ensembles de boosting

Ensembles de Gradient Boosting

Modelo	Umbral de mejora	Ratio de aprendizaje	Profunidad máxima	Número de árboles	Porcentaje de atributos	Porcentaje de instancias
eXtreme Gradient Boost	9132.97187266263	0.02968498714490269	4	153.0	0.4645360146077211	0.3

Tabla A.8: Hiperparámetros de Extreme Gradient Boosting

Modelo	Ratio de aprendizaje	Profundidad máxima	Número de árboles	Regularización	Intensidad de la aleatoriedad
Categorical Boosting	0.04893017698331533	8	200.0	5.637922884126309	2.0

Tabla A.9: Hiperparámetros de Categorical Boosting

Modelo	Ratio de aprendizaje	Profundidad máxima	Número de árboles	Número de hojas	Instancias mínimas por hoja
Light Gradient Boosting Machine	0.19674635138648194	3	74.0	10.0	79.0

Tabla A.10: Hiperparámetros de Light Gradient Boosting Machine

Modelo	Ratio de aprendizaje	Profundidad máxima	Porcentaje de atributos	Número de árboles	Número de hojas	Instancias mínimas por hoja
Histogram Gradient Boosting	0.13430802331510008	8	0.4913920710479611	50.0	9.0	20.0

Tabla A.11: Hiperparámetros de Histogram Gradient Boosting

B. Contenidos del repositorio

El repositorio en el que se almacena el trabajo realizado y la memoria contiene los siguientes recursos principales:

- **WiDS Datathon 2024 C2 - Part 1 (EDA).ipynb:** Una libreta de *Jupyter* en la que se incluye el proceso de **exploración de datos**.
- **WiDS Datathon 2024 C2 - Part 2 (Regression Models).ipynb:** Una libreta de *Jupyter* en la que se incluyen los procesos de **pre-procesamiento**, **modelado** y **evaluación** del proceso de ciencia de datos.
- **streamlit:** Una carpeta en la que se contiene todos los ficheros de código **Python** para el **despliegue** de la aplicación web a través de la que se hace disponible el modelo.
    - Debido al funcionamiento de *Streamlit* a la hora de desplegarse, la mayoría de recursos se encuentran duplicados en la raíz del repositorio.
- **thesis:** Una carpeta en la que se incluyen todos los ficheros de **LaTeX** necesarios para la compilación de la memoria.

Además, el repositorio contiene las siguientes carpetas con información adicional utilizada durante el trabajo:

- **data:** Una carpeta con los conjuntos de datos de la competición.
- **models:** Una carpeta con todos los modelos entrenados, tanto los 16 modelos ajustados como el modelo final seleccionado.
- **results:** Una carpeta con todos los resultados de la experimentación, tanto las estadísticas como los hiperparámetros - en formato *CSV*.