Departamento de Matemáticas, Facultad de Ciencias
Universidad Autónoma de Madrid

Interpretability of classification models

Trabajo de fin de grado

Grado en Matemáticas

Autor: Gregorio Blázquez Martínez
Tutor: Patrizio Guagliardo
Co-tutor: Damián Álvarez Piqueras

Curso 2023-2024

A mi padre y a mi hermano Joaquín, por ser las primeras razones por las que me enamoré de las matemáticas, y el motivo por el que he llegado hasta aquí.

Agradecimientos

Quiero dar las gracias a mis tutores Damián Álvarez y Patrizio Guagliardo por haberme brindado la oportunidad de descubrir este tema tan apasionante, por haber dedicado tanto tiempo y esfuerzo en tantas reuniones durante la elaboración de este TFG y por su apoyo. Quiero agradecerles que me hayan mostrado lo lejos que pueden llegar la curiosidad y ganas de aprender.

También quiero agradecer a mi familia, amigos y a todas las personas que me han apoyado a lo largo de este año.

Resumen

El campo de la interpretabilidad de modelos de clasificación está cobrando cada vez más importancia en un mundo donde la toma de decisiones precisas es fundamental. Aunque los modelos de clasificación han alcanzado niveles de precisión sin precedentes gracias al aprendizaje automático y grandes volúmenes de datos, comprender el razonamiento detrás de las decisiones de estos modelos es igualmente crucial. Este trabajo se centra en proporcionar una base formal para comprender los conceptos clave relacionados con la interpretabilidad en modelos de clasificación, con un enfoque particular en el método SHAP (SHapley Additive exPlanations), una herramienta que a pesar de haber surgido hace tan solo siete años, ya se coloca como un estándar.

Como complemento, este trabajo termina con una implementación práctica de SHAP en un caso real, una demostración tangible de su utilidad. A través de explicaciones individuales y globales, SHAP ofrece una comprensión profunda y robusta de las decisiones del modelo, aumentando así la confianza en las predicciones y facilitando una toma de decisiones más informada. En conclusión, este trabajo destaca la importancia de combinar precisión con interpretabilidad en modelos de clasificación, y muestra cómo SHAP emerge como una herramienta indispensable en este campo en constante evolución.

Abstract

The field of interpretability in classification models is gaining increasing importance in a world where precise decision-making is crucial. Although classification models have achieved unprecedented levels of accuracy thanks to machine learning and large volumes of data, understanding the reasoning behind these models' decisions is equally critical. This work focuses on providing a formal foundation for understanding key concepts related to interpretability in classification models, with a particular emphasis on the SHAP method (SHapley Additive exPlanations), a tool that, despite emerging only seven years ago, has already become a standard.

Additionally, this work concludes with a practical implementation of SHAP in a real-world case, demonstrating its tangible utility. Through both individual and global explanations, SHAP offers a deep and robust understanding of model decisions, thereby increasing confidence in predictions and facilitating more informed decision-making. In conclusion, this work highlights the importance of combining accuracy with interpretability in classification models and shows how SHAP emerges as an indispensable tool in this constantly evolving field.

Índice general

CAPÍTULO 1
Introducción y preliminares

Vivimos en un mundo cada vez más interconectado, donde la información se vuelve más relevante con cada día que pasa.. En ciertos campos como la economía y la salud, la toma de decisiones precisas juega un papel fundamental, por ejemplo, en el ámbito económico la capacidad para predecir tendencias del mercado o identificar riesgos puede significar la diferencia entre el éxito y el fracaso de una empresa. Del mismo modo, en el campo de la salud es incluso más importante, pues la precisión en la identificación y tratamiento de enfermedades puede salvar vidas o tener un impacto directo en la calidad de vida de los pacientes.

En este contexto, los modelos de clasificación han surgido como herramientas indispensables para realizar predicciones y tomar decisiones fundamentadas. Gracias a los grandes volúmenes de datos y los avances en técnicas de aprendizaje automático, la precisión de estos modelos ha alcanzado niveles sin precedentes. Sin embargo, la precisión por sí sola no es suficiente; a menudo es crucial comprender el porqué de la toma de decisiones de estos modelos.

La interpretación de los resultados de los modelos de clasificación es esencial para establecer la confianza en ellos y poder seguir avanzando, en muchos casos, entender por qué un modelo ha tomado una decisión particular puede ser tan importante como la propia precisión del modelo. Volvemos a poner los dos ejemplos de antes: la salud, donde las decisiones basadas en modelos incorrectos pueden tener consecuencias devastadoras; y la economía, donde la decisión puede poner en riesgo grandes cantidades de dinero que no se pueden asumir sin la confianza necesaria. La existencia y necesidad de modelos basados en relaciones complejas, que a menudo no tienen una intepretación intuitiva, genera la necesidad de trabajos de interpretación como este.

Es en el ámbito económico donde ha surgido la semilla que ha dado paso a este TFG. En los modelos bancarios, la regulación actual impide usar modelos complejos por la falta de entendimiento e interpretablidad de sus decisiones, obligando a aplicar modelos muy simples con menor precisión. El objetivo principal de este trabajo es proporcionar una base formal para comprender los conceptos clave relacionados con la interpretabilidad en modelos de clasificación. Además de buscar mejorar la precisión de los modelos, nos centramos en garantizar su interpretabilidad, lo que contribuirá a una toma de decisiones más informada y ética en diversos contextos.

1.1. Estructura del trabajo

Este trabajo está estructurado en varias secciones de la siguiente manera:

- Primero hablaremos de los dos modelos interpretables por excelencia: Regresión Lineal y Regresión Logística. Son modelos que están estudiados matemáticamente pero en los cuales no vamos a profundizar tanto, el objetivo de esta sección es mostrar sus principales características e interpretación.
- Tras esto vamos a hablar de métodos específicos para mejorar la interpretabilidad de los modelos. En particular hablaremos de LIME (necesario para entender bien el último método), Shapley Values (la base matemática más importante de la interpretabilidad de modelos de clasificación) y SHAP (una combinación de los dos anteriores y que es el objetivo final de este TFG).
- Finalmente, mostramos la aplicación práctica de lo tratado en las dos secciones anteriores con una base de datos real prestada por una empresa privada y comentamos algunas conclusiones generales sobre los resultados obtenidos.

1.2. Terminología

En este apartado vamos a dar sólo unos comentarios debido a que los conceptos se han ido definiendo a lo largo de este trabajo cuando han sido necesarios, para facilitar el entendimiento. Partimos de conceptos bastante básicos pues, como ya se ha comentado, la idea de este TFG es formalizar de la manera más clara posible los conceptos que hoy se usan a diario en ámbitos relacionados con la ciencia de datos sin necesidad de una gran base matemática. La primera definición que debemos mostrar es la siguiente:

Definición 1.1. Modelo

Sea $X = \{X_1, X_2, ..., X_n\}$ un conjunto de variables independientes (predictoras) y Y una variable dependiente (predicción) que se pretende explicar. Denominamos modelo a la función $f : \mathbb{R}^n \to \mathbb{R}$ que mapea un vector de características $\mathbf{x} = (x_1, x_2, ..., x_n)$ a un valor y que representa la variable dependiente. Lo formalizamos como $Y = f(X_1, X_2, ..., X_n) + \epsilon$, donde ϵ es un término de error que captura la variabilidad no explicada por el modelo.

El objetivo del modelado es estimar la función f utilizando datos observados $(x_1^{(1)}, x_2^{(1)}, ..., x_n^{(1)}), (x_1^{(2)}, x_2^{(2)}, ..., x_n^{(2)}), ..., (x_1^{(m)}, x_2^{(m)}, ..., x_n^{(m)})$ junto con las respuestas observadas correspondientes $y^{(1)}, y^{(2)}, ..., y^{(m)}$, donde m es el número de muestras en el conjunto de datos.

En resumen, un modelo es una función que describe la relación entre las variables independientes y la variable dependiente.

Por otro lado, vamos a hablar continuamente del concepto de interpretabilidad. Esta idea no tiene definición matemática y depende de demasiados factores, la transparencia, tipo y complejidad del modelo son algunos ejemplos. No se pretende extenderse en un concepto que no se va a definir matemáticamente, a lo largo del TFG vamos a aceptar el uso extendido de este concepto como la capacidad de entender la predicción de un modelo en función de las variables independientes, para profundizar más en este término léase la sección de interpretabilidad del Molnar [1].

CAPÍTULO 2
Modelos interpretables

2.1. Regresión lineal

2.1.1. Formalización del modelo

La principal característica de este modelo es, como indica su nombre, que la relación que se establece entre la variable dependiente y las variables independientes es lineal. Supongamos que tenemos n variables independientes en nuestro modelo, que pueden ser representadas en el espacio $\mathbb{R}^n$ como un vector x con $x_j \in \mathbb{R} \quad \forall j = 1...n$. Este modelo predice la variable dependiente que queremos calcular ($\theta = f(x)$) como una combinación lineal del valor de cada una de las variables independientes (x_j) ponderadas por unos pesos o coeficientes (β_j) más el valor que le asignamos cuando todas las variables independientes son 0, llamado término de intercepción (β_0). Siendo y el valor real de la variable dependiente y ϵ el error cometido al estimarlo, podemos escribir la igualdad:

$$y - \theta = \epsilon \Rightarrow y = \beta_0 + \beta_1 \cdot x_1 + \cdots + \beta_n \cdot x_n + \epsilon \tag{2.1}$$

Dado un conjunto de k muestras $(y^{(i)}, x_1^{(i)} \dots x_n^{(i)})$, podemos escribir la ecuación anterior de forma matricial como:

$$\begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \cdots \\ y^{(k)} \end{bmatrix} = \begin{bmatrix} 1 & x_1^{(1)} & x_2^{(1)} & \cdots & x_n^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & \cdots & x_n^{(2)} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & x_1^{(k)} & x_2^{(k)} & \cdots & x_n^{(k)} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \cdots \\ \beta_n \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \cdots \\ \epsilon_k \end{bmatrix} \tag{2.2}$$

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} = \boldsymbol{\theta} + \boldsymbol{\epsilon}$$

Siendo:

- Y: un vector de dimensión k que contiene las variables dependientes.

- X: una matriz de dimensiones $k \times (n+1)$ que contiene las variables independientes, incluyendo una columna de unos para el término de intercepción.
- $\boldsymbol{\beta}$: un vector de coeficientes de dimensión $n+1$.
- $\boldsymbol{\epsilon}$: un vector de dimensiones k que contiene los términos de error.
- $\boldsymbol{\theta}$: un vector de dimensiones k que contiene las estimaciones obtenidas a partir de los pesos y las variables independientes.

Vamos a suponer que tenemos más datos que variables independientes, es decir $k > n+1$, algo que resulta muy intuitivo pues en caso contrario nuestro problema tendría poco interés.

2.1.2. Obtención de los pesos

Una vez planteado el modelo, debemos tratar un problema fundamental; necesitamos calcular los pesos adecuados para que el modelo se ajuste a nuestros datos. Para esto existen varios métodos, pero, dado los resultados que ofrece el Teorema de Gauss-Markov 2.2, se suele usar el ajuste por mínimos cuadrados. Buscamos el punto $\boldsymbol{\beta}$ de $\mathbb{R}^{n+1}$ que minimice el error cuadrático, esto podemos expresarlo en la siguiente ecuación:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\beta_0,\dots,\beta_n} \sum_{i=1}^{k} \left(y^{(i)} - \left(\beta_0 + \sum_{j=1}^{n} \beta_j x_j^{(i)} \right) \right)^2 \tag{2.3}$$

Es mucho más cómodo si lo vemos de forma matricial a partir de la Ecuación 2.2. Nuestro objetivo es minimizar el error $\boldsymbol{\epsilon}$ entre las variables dependientes $\mathbf{Y}$ y las estimaciones $\boldsymbol{\theta}$. En vez de verlo como el sumatorio de los errores al cuadrado $\sum_{j=1}^{k} {e_j}^2$, lo podemos ver con matrices y minimizar $\|\boldsymbol{\epsilon}\|^2 = \boldsymbol{\epsilon}^\intercal \boldsymbol{\epsilon} = \|\mathbf{Y} - \boldsymbol{\theta}\|^2$. Sea $\Omega = \{\boldsymbol{\theta} : \boldsymbol{\theta} = \mathbf{X}\boldsymbol{\beta} \quad \forall \boldsymbol{\beta}\}$ el subespacio generado por la matriz $\mathbf{X}$. A la hora de buscar los pesos óptimos por el método de mínimos cuadrados, permitimos a $\boldsymbol{\theta}$ variar en $\boldsymbol{\Omega}$.

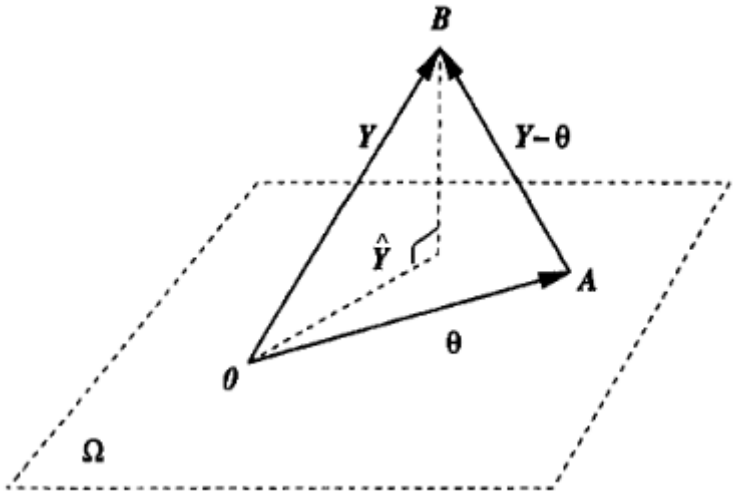
Lema 2.1. *Dado un vector $v \in \mathbb{R}^n$ y un subespacio U de $\mathbb{R}^n$, existe un único vector de U que minimiza la distancia a v y es la proyección ortogonal de v sobre U, denotada como $p_U(v)$. Es decir,*

$$\|v - p_U(v)\| < \|v - w\| \quad \forall w \in U \quad \textit{con } u \neq p_U(v)$$

Demostración. Sea $u = p_U(v)$, $\forall w \in U \quad v - w = (v - u) + (u - w)$ con $v - u \in \Omega^\perp$ y $u - w \in U$, por lo que tenemos $(v - u) \cdot (u - w) = 0$. Usando que son ortogonales podemos descomponer la norma $\|v-w\|^2 = \|v-u\|^2 + \|u-w\|^2 \Rightarrow \|v-w\|^2 \geq \|v-u\|^2$ y en particular $\|u - w\| = 0 \Leftrightarrow u = w$. □

Aplicando el Lema 2.1, obtenemos que $|\mathbf{Y} - \boldsymbol{\theta}|^2$ será mínimo cuando $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}$, que es la proyección ortogonal al subespacio Ω. Por lo tanto, el vector error será perpendicular

Figura 2.1: Representación del uso del lema con nuestra notación



al subespacio $((\mathbf{Y} - \hat{\boldsymbol{\theta}}) \perp \Omega)$. Por lo tanto, tenemos que el producto escalar entre $\mathbf{X}$ y $\boldsymbol{\epsilon}$ se anula, $\mathbf{X}^\intercal(\mathbf{Y} - \hat{\boldsymbol{\theta}}) = 0$, lo cual es equivalente a

$$\mathbf{X}^\intercal \hat{\boldsymbol{\theta}} = \mathbf{X}^\intercal \mathbf{Y}. \tag{2.4}$$

Siendo $\hat{\mathbf{Y}}$ la única proyección ortogonal de $\mathbf{Y}$ en el subespacio Ω, y pidiendo que las columnas de $\mathbf{X}$ sean linealmente independientes, tenemos un único vector $\hat{\boldsymbol{\beta}}$ tal que $\hat{\mathbf{Y}} = \mathbf{X}\hat{\boldsymbol{\beta}}$. Sustituimos en la Ecuación 2.4:

$$\mathbf{X}^\intercal \mathbf{X} \hat{\boldsymbol{\beta}} = \mathbf{X}^\intercal \mathbf{Y}. \tag{2.5}$$

Suponemos que $\mathbf{X}$ tiene rango k al tener columnas linealmente independientes, por lo que $\mathbf{X}^\intercal\mathbf{X}$ también es una matriz no singular y podemos calcular su inversa. Finalmente obtenemos una única solución.

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\intercal \mathbf{X})^{-1} \mathbf{X}^\intercal \mathbf{Y}. \tag{2.6}$$

Veamos otra manera de obtener el mismo resultado:

$$\boldsymbol{\epsilon}^\intercal \boldsymbol{\epsilon} = (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^\intercal (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) = \mathbf{Y}^\intercal \mathbf{Y} - 2\boldsymbol{\beta}^\intercal \mathbf{X}^\intercal \mathbf{Y} + \boldsymbol{\beta}^\intercal \mathbf{X}^\intercal \mathbf{X} \boldsymbol{\beta}. \tag{2.7}$$

Vamos a calcular la matriz diferencial, podemeos denotarlo $\frac{d}{d\boldsymbol{\beta}} = \left[\left(\frac{d}{d\beta_i}\right)\right]$ con $i = 0, 1, \ldots, n.$ y ver fácilmente:

Proposición 1. *1.* $\frac{d(\boldsymbol{\beta}^\intercal \mathbf{a})}{d\boldsymbol{\beta}} = \mathbf{a}$, $\mathbf{a} \in \mathbb{R}^{n+1}$

2. $\frac{d(\boldsymbol{\beta}^\intercal \mathbf{A} \boldsymbol{\beta})}{d\boldsymbol{\beta}} = 2\mathbf{A}\boldsymbol{\beta}$, *siendo* $\mathbf{A}$ *simétrica.*

Aplicando la proposición obtenemos igual que antes

$$-2\mathbf{X}^\intercal\mathbf{Y} + 2\mathbf{X}^\intercal\mathbf{X}\boldsymbol{\beta} = 0$$

o

$$\mathbf{X}^\intercal\mathbf{X}\boldsymbol{\beta} = \mathbf{X}^\intercal\mathbf{Y}.$$

2.1.3. Gauss-Markov

Este capítulo trata de desglosar el teorema de Gauss-Markov, en vez de formularlo sin más explicaciones, para que se entienda por qué pedimos las hipótesis del teorema y qué nos ofrece como resultado. Hemos visto cómo obtener los pesos que minimizan el error para unos valores determinados, y en esta sección vamos a ver qué condiciones nos aseguran que los pesos que hemos calculado sean óptimos al minimizar la varianza, con ello terminaremos obteniendo el mejor estimador lineal insesgado.

Una primera característica básica que le vamos a pedir a la estimación que hemos obtenido es que sea insesgada, abusando de notación definimos $E[\hat{\boldsymbol{\beta}}] = \left[(E(\hat{\beta}_i))\right]$ para $i = 0, \ldots, n$.

Usando el estimador que hemos obtenido en la Ecuacion 2.6 y sustituyendo $\mathbf{Y}$:

$$E[\hat{\boldsymbol{\beta}}] = E[(\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal\mathbf{Y}] = E[\boldsymbol{\beta} + (\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal\boldsymbol{\epsilon}] = \boldsymbol{\beta} + (\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal\mathbf{E}[\boldsymbol{\epsilon}]$$

Como estamos buscando $E[\hat{\boldsymbol{\beta}}] = \boldsymbol{\beta}$, nos basta pedir $E[\boldsymbol{\epsilon}] = 0$. Para el siguiente paso, nos interesa tener el estimador óptimo de todos los estimadores lineales insesgados, y lo vamos a conseguir buscando que tenga la menor varianza, es decir, que esté más concentrado en torno a $\boldsymbol{\beta}$. Abusando otra vez la notación definimos $\mathrm{Var}[\hat{\boldsymbol{\beta}}] = \left[(\mathrm{cov}[\hat{\beta}_i, \hat{\beta}_j])\right]$ y usando $\mathrm{Var}[\mathbf{Ax}] = \mathbf{A}\mathrm{Var}[\mathbf{x}]\mathbf{A}^\intercal$ obtenemos:

$$\mathrm{Var}[\hat{\boldsymbol{\beta}}] = \mathrm{Var}[\boldsymbol{\beta}+(\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal\boldsymbol{\epsilon}] = \mathrm{Var}[(\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal\boldsymbol{\epsilon}] = (\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal\mathrm{Var}[\boldsymbol{\epsilon}]\mathbf{X}(\mathbf{X}^\intercal\mathbf{X})^{-1}$$

Al igual que ocurría anteriormente con la esperanza, la varianza de nuestro estimador depende de la varianza del error. Vamos a asumir que las variables no están correlacionadas entre sí (como se verá en la siguiente sección), y que la varianza sea la misma para todos los ϵ_i. De esta manera podemos sustituir $\mathrm{Var}[\boldsymbol{\epsilon}] = \sigma^2\mathbf{I}$ en la fórmula anterior:

$$\mathrm{Var}[\hat{\boldsymbol{\beta}}] = (\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal\sigma^2\mathbf{I}\mathbf{X}(\mathbf{X}^\intercal\mathbf{X})^{-1} = \sigma^2(\mathbf{X}^\intercal\mathbf{X})^{-1}$$

Nótese que la varianza aumentará si σ aumenta y, dado que $(\mathbf{X}^\intercal\mathbf{X})_{ij} = \sum_{k=1}^{n} x_{ik}x_{kj}$, disminuye si n crece. Ya podemos enunciar el teorema que asegura que el estimador obtenido por mínimos cuadrados es el mejor, ya que tiene la menor varianza entre los estimadores de su tipo.

Teorema 2.2. *Sea $\check{\boldsymbol{\beta}}$ un estimador lineal insesgado de $\boldsymbol{\beta}$ en nuestro modelo con $E[\boldsymbol{\epsilon}] = 0$ y $Var[\boldsymbol{\epsilon}] = \sigma^2 I$. Entonces,*

$$Var[\check{\beta}_i] \geq Var[\hat{\beta}_i] \quad \forall i = 0, \ldots, n$$

siendo $\hat{\boldsymbol{\beta}}$ nuestro estimador obtenido por mínimos cuadrados.

Demostración. Para la demostración, consulte el Teorema 3.2 del capítulo 3.2 "Properties of Least Squares Estimates" en el libro de Seber [2]. □

La conclusión de este teorema se puede resumir en la siguiente frase: asumiendo las hipótesis que se explican a continuación, el estimador que hemos obtenido mediante mínimos cuadrados es el mejor estimador lineal insesgado.

2.1.4. Hipótesis del modelo

Para que el modelo funcione correctamente debemos asumir que la muestra cumple varias condiciones. En particular, estamos asumiendo que la relación entre la variable dependiente y las variables independientes es lineal, lo cual no suele ser muy realista y limita las muestras en las que se puede usar el modelo. Además, hemos requerido que la matriz $\mathbf{X}$ tenga las columnas linealmente independientes para poder obtener las Ecuaciones 2.5 y 2.6, es decir, pedimos que las variables X_j sean linealmente independientes. Otro factor que puede afectar a nuestro modelo es que las variables independientes estén altamente correlacionada.

Supongamos que tenemos dos variables altamente correlacionadas, podemos asumir sin pérdida de generalidad que son X_1 y X_2, y podemos afirmar que $X_2 = \alpha X_1 + \text{otros términos}$. Donde α es un coeficiente no nulo y "otros términos" representa el efecto de las otras variables independientes en X_2. Si sustituimos en la Ecuación 2.1 obtenemos

$$y = \beta_0 + \beta_1 \cdot x_1 + \beta_2 \alpha \cdot x_1 + \text{otros términos} + \cdots + \beta_n \cdot x_n + \epsilon.$$

Sacando factor común, podemos obtener que y depende de $\beta_1 x_1 + \beta_2 \alpha x_1$, obteniendo que para la variable independiente X_1 tiene dos parámetros libres de nuestro modelo. Por lo que a la hora de intentar obtener los pesos óptimos tenemos infinitas posibilidades de tomar β_1 y β_2.

Otras condiciones que no se derivan del cálculo de los pesos, sino que se requieren para que el modelo funcione correctamente (es decir, que podamos obtener resultados precisos y confiables) y podamos aplicar Gauss-Markov son:

- Asumimos que las variables independientes son constantes fijas y no sujetas a errores de medición para simplificar el modelo y no introducir variabilidad no deseada en el modelo.
- En la práctica asumimos normalidad de los errores ($\varepsilon \sim \mathcal{N}(0, \sigma)$) para que podamos tener estabilidad de los resultados y necesarios para poder aplicar Gauss-Markov y obtener estimadores lineales insesgados.
- Desarrollando lo anterior, pedimos homocedasticidad (varianza constante), es decir que la diferencias entre los valores observados y los valores predichos sea constante en todo el rango de los valores de las variables independientes. De no ser así, el modelo dará más peso a las observaciones con menor varianza de los residuos, lo que puede resultar en estimaciones sesgadas de los coeficientes. Además, necesitamos pedir esta condición para aplicar Gauss-Markov.

2.1.5. Interpretabilidad

Respecto a la interpretabilidad que nos interesa para este proyecto, este modelo es muy fácil de interpretar debido a su sencillez. Vemos que los pesos β_j representan el cambio en la variable dependiente que supone cambiar en una unidad el valor de x_j, dejando el resto de variables independientes fijas. Conceptualmente, esto significa que β_j indica la influencia directa de la variable x_j sobre la variable dependiente. Un valor positivo de β_j sugiere que un aumento en x_j está asociado con un aumento en la variable dependiente, mientras que un valor negativo indica una asociación inversa. Además, la magnitud de β_j refleja la fuerza de esta relación: valores absolutos mayores indican un efecto más fuerte de x_j sobre la variable dependiente.

Para una mejor comprensión, también es útil considerar el contexto y la escala de las variables. Por ejemplo, si x_j representa años de experiencia laboral y la variable dependiente es el salario, un β_j de 2000 implicaría que, en promedio, cada año adicional de experiencia laboral está asociado con un aumento de 2000 unidades monetarias en el salario, manteniendo constantes las demás variables.

Es importante notar que la interpretabilidad intrínseca de los coeficientes en una regresión lineal nos permite hacer estas inferencias directas, lo cual es una ventaja significativa en comparación con modelos más complejos donde la relación entre variables puede ser menos transparente.

2.2. Regresión logística

2.2.1. Formalización del modelo

Supongamos que tenemos un problema de clasificación con dos posibles categorías, la regresión lineal no funciona bien en este problema por su naturaleza continua. En este sentido, la regresión logística se presenta como una extensión natural de la regresión lineal para abordar problemas de clasificación binaria. La regresión logística emplea la función logística, también conocida como sigmoide, para modelar y predecir probabilidades.

$$\text{logistic}(x) = \frac{1}{1 + \exp(-x)}$$

La función logística transforma la combinación lineal de las variables independientes mediante una curva en forma de "S", confinando las predicciones en el intervalo $(0, 1)$, lo cual es crucial para interpretarlas como probabilidades. Si ahora sustituimos esta por la función de regresión lineal de la Ecuación 2.1 quitando el error epsilon, pues nos interesa la estimación $\hat{y}$ en vez del valor real que no conocemos y, podemos escribir:

$$\mathbb{P}(y = 1|\mathbf{x}) = \frac{1}{1 + \exp(-(\beta_0 + \beta_1 \cdot x_1 + \cdots + \beta_n \cdot x_n))} \tag{2.8}$$

Al igual que en el capítulo anterior nos será útil usar la notación vectorial:

$$\mathbb{P}(y = 1|\mathbf{x}) = \frac{1}{1 + \exp(-\boldsymbol{\beta}^\intercal \mathbf{x})} \tag{2.9}$$

Nótese que $\mathbf{x}$ es un vector columna, formado por un 1 en la primera coordenada seguido de las variables independientes x_i, y no un vector fila como definimos en el apartado de regresión lineal. La matriz $\mathbf{X}$ que definimos en la Ecuación 2.2, es ahora la traspuesta.

2.2.2. Función de máxima verosimilitud

Usando la siguiente notación para simplificar las fórmulas:

$$\mathbb{P}(y^{(i)} = 1|\mathbf{x^{(i)}}) = p(\mathbf{x^{(i)}})$$

Queremos que nuestro modelo se ajuste lo mejor posible a nuestros datos, para esto suponemos que las probabilidades de cada observación son independientes y buscamos $\boldsymbol{\beta}$ que maximice el siguiente producto, denominado función de verosimilitud:

$$L(\boldsymbol{\beta}) = \prod_{y^{(i)}=1} p(\mathbf{x^{(i)}}) * \prod_{y^{(i)}=0} (1 - p(\mathbf{x^{(i)}})) = \prod_{i=1}^{k} p(\mathbf{x^{(i)}})^{y^{(i)}} * (1 - p(\mathbf{x^{(i)}}))^{1-y^{(i)}}$$

Esto es equivalente a que $\boldsymbol{\beta}$ minimize la log-verosimilitud:

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{k} y^{(i)} \log\left(p(\mathbf{x^{(i)}})\right) + (1 - y^{(i)}) \log\left(1 - p(\mathbf{x^{(i)}})\right)$$

Sustituyendo $p(\mathbf{x^{(i)}})$ podemos simplificar (véanse los cálculos en el apéndice Demostraciones o Cálculos) y terminamos con la forma final de la función de probabilidad logarítmica que se va a optimizar:

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{k} y^{(i)} \boldsymbol{\beta}^{\mathsf{T}} \mathbf{x^{(i)}} - \log\left(1 + \exp\left(\boldsymbol{\beta}^{\mathsf{T}} \mathbf{x^{(i)}}\right)\right) \tag{2.10}$$

2.2.3. Obtención de los pesos

En el apéndice Método Newton-raphson, hay unos pocos comentarios sobre el método de Newton-raphson que vamos a aplicar en este apartado. El método de Newton nos da la siguiente ecuación:

$$\boldsymbol{\beta^{(s+1)}} = \boldsymbol{\beta^{(s)}} - [H_l(\boldsymbol{\beta}^{(s)})]^{-1} \nabla l(\boldsymbol{\beta^{(s)}}) \tag{2.11}$$

Proposición 2.

$$\nabla_{\boldsymbol{\beta}}(l(\boldsymbol{\beta})) = \sum_{i=1}^{k} (y^{(i)} - p(\mathbf{x^{(i)}})) * \mathbf{x^{(i)}}$$

Demostración. Véase los cálculos en el apéndice Demostraciones o Cálculos. □

Podemos escribir esta ecuación en forma matricial:

$$\nabla_{\boldsymbol{\beta}} l = \mathbf{X}(\mathbf{Y} - \hat{\mathbf{Y}})$$

Proposición 3.

$$H_l = \sum_{i=1}^{k} p(\mathbf{x^{(i)}})(1 - p(\mathbf{x^{(i)}}))\mathbf{x^{(i)}}^{\mathsf{T}} \mathbf{x}^{(i)}$$

Demostración. Véase los cálculos en el apéndice Demostraciones o Cálculos. □

Podemos escribir esta ecuación en forma matricial:

$$H_l = \mathbf{X}\mathbf{W}\mathbf{X}^{\mathsf{T}}$$

siendo $\mathbf{W}$ la siguiente matriz diagonal:

$$\mathbf{W} = \begin{bmatrix} p(x^{(1)})(1 - p(x^{(1)})) & 0 & \dots & 0 \\ 0 & p(x^{(2)})(1 - p(x^{(2)})) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & p(x^{(k)})(1 - p(x^{(k)})) \end{bmatrix}$$

Una vez hemos calculado el gradiente y el hessiano podemos sustituir en el método de Newton obteniendo:

$$\boldsymbol{\beta}^{(s+1)} = \boldsymbol{\beta}^{(s)} - [\mathbf{X}\mathbf{W}^{(s)}\mathbf{X}^{\intercal}]^{-1}\mathbf{X}(\mathbf{Y} - \hat{\mathbf{Y}}^{(s)}) \tag{2.12}$$

2.2.4. Interpretabilidad

Respecto a la interpretabilidad que nos interesa, la regresión logística es más compleja que la regresión lineal ya que debido a la función logística, la interpretación de los pesos β_j no es lineal, sino que es aditiva por ser la multiplicación de factores exponenciales. Desarrollemos esta idea con las fórmulas.

Utilizando el término inglés *odds* vamos a estudiar la relación entre la probabilidad de obtener $y = 1$ y su complementario $y = 0$.

$$\text{odds} = \frac{\mathbb{P}(y=1)}{1-\mathbb{P}(y=1)} = \frac{\mathbb{P}(y=1)}{\mathbb{P}(y=0)} = \exp\left(\beta_0 + \beta_1 \cdot x_1 + \cdots + \beta_n \cdot x_n\right)$$

Si aplicamos el logaritmo a la expresión anterior obtenemos una regresión lineal que ya hemos estudiado en el capítulo anterior:

$$\log(\text{odds}) = \beta_0 + \beta_1 \cdot x_1 + \cdots + \beta_n \cdot x_n$$

Quizás resulta más intuitivo si observamos el ratio dado por una predicción y la misma sumando una unidad a un x_j cualquiera.

$$\frac{\text{odds}_{x_j+1}}{\text{odds}} = \frac{\exp\left(\beta_0 + \beta_1 \cdot x_1 + \cdots + \beta_j \cdot (x_j+1) + \cdots + \beta_n \cdot x_n\right)}{\exp\left(\beta_0 + \beta_1 \cdot x_1 + \cdots + \beta_j \cdot x_j + \cdots + \beta_n \cdot x_n\right)}$$

Juntando los exponentes y eliminando términos obtenemos:

$$\frac{\text{odds}_{x_j+1}}{\text{odds}} = \exp\left(\beta_j \cdot (x_j+1) - \beta_j \cdot x_j\right) = \exp\beta_j$$

Esto es equivalente a que:

$$\text{odds}_{x_j+1} = \exp\beta_j \text{odds}_{x_j}$$

Es decir, aumentar la variable x_j en una unidad incrementa la relación entre la probabilidad de que y sea 1 con la probabilidad de que y sea 0 en un factor $\exp\beta_j$.

CAPÍTULO 3
Métodos interpretabilidad

Según se menciona en [5], *The best explanation of a simple model is the model itself; it perfectly represents itself and is easy to understand.* Ya hemos hablado de modelos simples que podemos interpretar directamente, pero en la práctica generalmente es necesario recurrir a modelos ensamblados o demasiado complejos, como redes neuronales, para obtener una mayor precisión. En esta sección partimos de una idea tan sencilla como es utilizar un modelo más simple para explicar modelos más complejos, pero que en la práctica es enormemente útil. En esta línea definimos:

Definición 3.1. Denominamos *modelo de explicación* a cualquier modelo interpretable que aproxima el modelo original.

3.1. LIME (Local Interpretable Model-Agnostic Explanations)

LIME es una técnica utilizada para explicar clasificaciones individuales de un modelo de clasificación de manera que pueda ser contrastada y podamos confiar en el resultado. Fue presentada en 2016 el siguiente artículo [3], *Why Should I Trust You?: Explaining the Predictions of Any Classifier*, y su objetivo fundamental es proporcionar una explicación **local** de por qué un modelo en particular ha tomado una decisión específica para una instancia de datos dada. En otras palabras, LIME se centra en la interpretación de predicciones en un contexto de "punto único", lo que significa que no se esfuerza por comprender todo el modelo en su conjunto, sino que se concentra en una instancia individual y su resultado.

Una de las ventajas de LIME es que es un enfoque **agnóstico** al modelo, es decir, que se puede aplicar a cualquier modelo de clasificación sin necesidad de conocer su estructura interna. Esto hace que LIME sea una herramienta versátil y aplicable a una amplia gama de problemas.

3.1.1. Formalización

Siendo:

- X: el espacio de características de entrada.
- Y: el espacio de etiquetas de salida.
- $f : X \to Y$: el modelo complejo que se desea interpretar.
- x: una instancia de datos de X que se pretende explicar.
- $g : X \to Y$: el modelo interpretable local que se utilizará como aproximación de f en un entorno de x.
- D: un conjunto de datos generados alrededor de x.

El objetivo de LIME es encontrar un modelo interpretable g que sea una buena aproximación de f en un vecindario D de x, es decir, g es óptimo en términos de minimización de la función de pérdida L y restricciones de complejidad $\Omega(g)$. De esta manera podemos definir el siguiente problema de optimización:

$$\min_{g} L(f, g, \pi_x) + \Omega(g) \tag{3.1}$$

donde:

- $L(f, g, \pi_x) = \sum_i \pi_{x_i} \cdot l(f(x_i), g(x_i))$ es la función de pérdida que mide la diferencia entre las predicciones de f y g en D.
- l es una función de pérdida para un punto en particular y mide la diferencia entre las predicciones de f y g. Esta función suele ser el valor absoluto de la diferencia o la diferencia al cuadrado.
- $\pi_x(x_i)$ es la función que asigna a cada punto en D un peso con respecto a su similitud (puede obtenerse fácilmente a partir de una distancia) con x.
- $\Omega(g)$ es una medida de complejidad del modelo interpretable g.

La solución óptima de este problema de optimización proporciona el modelo interpretable g con sus parámetros que mejor aproxima a f en el vecindario de x. LIME utiliza técnicas como el muestreo de D y la minimización de $\Omega(g)$ para encontrar g de manera eficiente y efectiva.

No vamos a comentar más de momento sobre LIME pues veremos que poco después de un año de la aparición de LIME, SHAP surgió como una alternativa más sólida. En el capítulo sobre SHAP 3.3, haremos referencia a LIME y comentaremos un resultado de esta técnica que se obtuvo gracias a SHAP.

En resumen, LIME es un enfoque matemático que busca aproximar un modelo de aprendizaje automático complejo f en un vecindario local de una instancia x mediante la minimización de una función de pérdida L y restricciones de complejidad $\Omega(g)$. Esto permite obtener explicaciones locales y comprensibles de las predicciones de f en términos de g. Como curiosidad y complemento a este TFG, en el siguiente enlace se puede acceder a un archivo Jupyter-Notebook donde se ha implementado por pasos el funcionamiento de LIME: Ejemplo de LIME en Jupyter-Notebook.

3.2. Shapley Values

Los valores de Shapley son sin duda el culpable de que exista este TFG, son la base matemática de SHAP y la parte más interesante de este trabajo; pero antes de introducir su formalización, vamos a intuir de dónde proviene la idea de los valores Shapley.

Estamos interesados en cómo cada característica afecta la predicción de un punto en particular de nuestra base de datos. En un modelo lineal, es fácil calcular los efectos individuales. Vimos en el capítulo 2 que así es como se ve la predicción de un modelo lineal para una instancia de datos:

$$\hat{f}(x) = \beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n$$

La contribución ϕ_j de la j-ésima característica en la predicción $\hat{f}(x)$ es:

$$\phi_j(\hat{f}) = \beta_j x_j - E(\beta_j X_j) = \beta_j x_j - \beta_j E(X_j)$$

donde $E(\beta_j X_j)$ es la estimación del efecto medio para la característica j. La contribución es la diferencia entre el efecto de la característica menos el efecto promedio. Si sumamos todas las contribuciones de las características para una instancia, el resultado es el siguiente:

$$\begin{aligned}\sum_{j=1}^{n} \phi_j(\hat{f}) &= \sum_{j=1}^{n}(\beta_j x_j - E(\beta_j X_j)) \\ &= (\beta_0 + \sum_{j=1}^{n} \beta_j x_j) - (\beta_0 + \sum_{j=1}^{n} E(\beta_j X_j)) \\ &= \hat{f}(x) - E(\hat{f}(X)) \quad \text{(Denotamos)}\end{aligned}$$

Esto es el valor predicho para el punto de datos x menos el valor predicho promedio. Nótese que las contribuciones de las características pueden ser negativas.

¿Podemos hacer lo mismo para cualquier tipo de modelo? Dado que generalmente no tenemos pesos similares en otros tipos de modelos, necesitamos una solución diferente. La solución viene de la teoría de juegos cooperativos. Los valores de Shapley son una solución para calcular las contribuciones de las características para predicciones individuales para cualquier modelo de aprendizaje automático.

3.2.1. Formalización

Antes de poder aplicar los valores de Shapley a los modelos de clasificación, necesitamos comprenderlos y mostrarlos con su formulación original a través de la teoría de juegos. Durante la década de 1940 y principios de la década de 1950, hubo un gran interés en la teoría de juegos, en parte debido a su aplicación en campos como la economía, la ciencia política y la biología. Lloyd Shapley, matemático y economista

estadounidense, publicó en 1953 *A Value for n-Person Games* [4], basándose en los trabajos previos de von Neumann y Morgenstern de la década de 1940. Los resultados mostrados a continuación se basan en esta referencia.

En el contexto de la teoría de juegos y los valores de Shapley, el juego coalicional se define formalmente de la siguiente manera:

Definición 3.2. Se tiene un conjunto N de n jugadores y una función superaditiva v que asigna subconjuntos de jugadores a números reales: $v : U = 2^N \to \mathbb{R}$, donde $v(\emptyset) = 0$, siendo $\emptyset$ el conjunto vacío. La función v se llama función característica.

La función v tiene la siguiente interpretación: si S es una coalición de jugadores, entonces $v(S)$, llamado el valor de la coalición S, describe la suma total esperada de pagos que los miembros de S pueden obtener mediante la cooperación.

Definición 3.3. Un carrier o base de v es cualquier conjunto $N \subseteq U$ con

$$v(S) = v(N \cap S) \quad (\text{para todo } S \subseteq U).$$

De esta manera evitamos jugadores que no aportan nada para definir correctamente un juego. Vamos a estudiar solo los juegos con carriers finitos.

Definición 3.4. Sea $\Pi(U)$ el conjunto de permutaciones de U. Si $\pi \in \Pi(U)$, entonces, escribiendo πS para la imagen de S bajo π, podemos definir la función πv por

$$\pi v(\pi S) = v(S) \quad (\text{para todo } S \subseteq U).$$

Shapley usó esta función para definir un *juego abstracto*, quedándose con el conjunto de reglas de un juego y no con los jugadores particulares.

Definición 3.5. Por la función valor $\phi[v]$ del juego v entendemos una función que asocia a cada $i \in U$ un número real $\phi_i[v]$.

Nos quedamos con la función valor de Shapley, la cual satisface las condiciones de los siguientes tres axiomas:

AXIOMA 1. Para cada π en $\Pi(U)$,

$$\phi_{\pi i}[\pi v] = \phi_i[v] \quad (\text{para todo } i \in U).$$

Este axioma de simetría nos sirve para afirmar que no importa el orden de los jugadores sino sus aportaciones. Como hemos comentado antes, esto nos da que el valor es una propiedad del juego abstracto.

AXIOMA 2. Para cada carrier N de v,

$$\sum_{i \in N} \phi_i[v] = v(N).$$

Este axioma se usa para afirmar que todo el valor del juego está explicado dentro del carrier, por lo que podemos restringirnos a este. Esto será especialmente importante

en la aplicación práctica a los modelos, pues nos sirve para no tener que preocuparnos por variables que no aportan nada a nuestro modelo.

AXIOMA 3. Para cualquier par de juegos v y w,

$$\phi[v+w] = \phi[v] + \phi[w].$$

Este axioma nos sirve para poder relacionar juegos independientes. También es especialmente útil en modelos si podemos descomponer la decisión general en varias decisiones independientes. Aunque no hemos definido este tipo de modelo, para modelos ensamblados (por ejemplo, un Random Forest), que funcionan como la ponderación de predicciones individuales, este axioma nos permite calcular los valores de Shapley de la predicción general como la ponderación de los valores de Shapley de las predicciones individuales. De hecho, esta propiedad se utiliza en la práctica en un estimador específico para modelos ensemble de árboles de decisión llamado TreeSHAP.

3.2.2. Teorema

Ya tenemos todo lo necesario para afirmar el teorema de existencia y unicidad de los valores de Shapley. Denotamos $s = |S|$, $n = |N|$ y $r = |R|$.

Teorema 3.6. *Para un juego coalicional dado v con un carrier finito N existe una única función valor que satisface los tres axiomas anteriores y viene dada por la siguiente fórmula:*

$$\phi_i[v] = \sum_{S \subseteq N \setminus \{i\}} \frac{s!(n-s-1)!}{n!}[v(S \cup \{i\}) - v(S)] \quad (\textit{para todo } i \in U).$$

Vamos a demostrar la unicidad de manera constructiva, es decir, si una función satisface los tres axiomas debe ser el valor de Shapley. La existencia, ver que los valores de Shapley cumplen los tres axiomas, se demuestra por comprobación directa y no va a realizarse para no extenderme demasiado.

Lema 3.7. *Si N es un carrier finito de v, entonces, para $i \notin N$,*

$$\phi_i[v] = 0.$$

Demostración. Tomamos $i \notin N$. Tanto N como $N \cup \{i\}$ son carriers de v; y $v(N) = v(N \cup \{i\})$ por la definición 3.3. Por el Axioma 2 $\sum_N \phi_j[v] = v(N) = v(N \cup \{i\}) = \sum_{N \cup \{i\}} \phi_j[v]$ y por tanto $\phi_i[v] = 0$. □

Ahora para cualquier $R \subseteq U$, $R \neq \varnothing$, definimos el juego v_R como sigue:

$$v_R(S) = \begin{cases} 1, & \text{si } R \subseteq S, \\ 0, & \text{si } R \not\subseteq S. \end{cases} \tag{3.2}$$

Queda definido también el juego cv_R, para cualquier $c \geq 0$, siendo R un carrier.

Lema 3.8. *Para $c \geq 0$ y $0 < r < \infty$, tenemos*

$$\phi_i[cv_R] = \begin{cases} \frac{c}{r}, & si\ i \in R, \\ 0, & si\ i \notin R. \end{cases}$$

Demostración. Tomamos $i,j \in R$ y elegimos $\pi \in \Pi(U)$ tal que $\pi R = R$ y $\pi i = j$. Entonces por definición $\pi v_R = v_R$ y por el axioma 1:

$$\phi_j[cv_R] = \phi_i[cv_R]$$

Volvemos a aplicar el Axioma 2,

$$c = cv_R(R) = \sum_{j \in R} \phi_j[cv_R] = r\phi_i[cv_R]$$

para cualquier $i \in R$. Esto, con el Lema 1, completa la prueba. □

Lema 3.9. *Cualquier juego con carrier finito es una combinación lineal de juegos simétricos v_R:*

$$v = \sum_{R \subseteq N, R \neq \varnothing} c_R(v) v_R,$$

siendo N cualquier carrier finito de v. Los coeficientes son independientes de N, y están dados por

$$c_R(v) = \sum_{T \subseteq R} (-1)^{r-t} v(T) \quad (0 < r < \infty). \tag{3.3}$$

Demostración. Debemos verificar que

$$v(S) = \sum_{R \subseteq N, R \neq \varnothing} c_R(v) v_R(S) \tag{3.4}$$

se cumple para todo $S \subseteq U$, y para cualquier carrier finito N de v.

Suponemos $S \subseteq N$, entonces 3.4 se reduce, por las definiciones 3.2 y 3.3, a

$$\begin{aligned} v(S) &= \sum_{R \subseteq S} \sum_{T \subseteq R} (-1)^{r-t} v(T) \\ &= \sum_{T \subseteq R} \left[\sum_{r=t}^{s} (-1)^{r-t} \binom{s-t}{r-t} \right] v(T). \end{aligned}$$

La expresión entre corchetes se anula excepto para $s = t$, muestro como esto se deriva del binomio de Newton para $(-1+1)^k$ en el apéndice Demostraciones o Cálculos. Por lo que nos queda la identidad $v(S) = v(S)$.

En general para S no necesariamente en N tenemos, por la definición de carrier 3.3:

$$\begin{aligned} v(S) &= v(N \cap S) \\ &= \sum_{R \subseteq N} c_R(v) v_R(N \cap S) \\ &= \sum_{R \subseteq N} c_R v_R(S). \end{aligned}$$

Esto completa la prueba. □

Vistos estos tres lemas ya podemos demostrar el teorema.

Demostración. Aplicando el Lema 2 a la representación del Lema 3, obtenemos la siguiente fórmula que nos da el valor de un juego v cualquiera:

$$\phi_i[v] = \sum_{R \subseteq N, i \in R} \frac{c_R(v)}{r} \quad \text{(para todo } i \in N\text{).} \tag{3.5}$$

Sustituyendo 3.3 y simplificando el resultado, obtenemos:

$$\phi_i[v] = \sum_{S \subseteq N, i \in S} \frac{(s-1)!(n-s)!}{n!} v(S) - \sum_{S \subseteq N, i \notin S} \frac{s!(n-s-1)!}{n!} v(S) \quad \text{(para todo } i \in N\text{).}$$

Que podemos reescribir como:

$$\phi_i[v] = \sum_{S \subseteq N \setminus \{i\}} \frac{s!(n-s-1)!}{n!} [v(S \cup \{i\}) - v(S)] \quad \text{(para todo } i \in N\text{).}$$

Finalmente, aplicamos el Lema 1 para extender la definición del carrier N a U y obtenemos la unicidad que estábamos buscando. □

3.2.3. Interpretación

Como es el objetivo de este TFG, vamos a hablar sobre la interpretación de los valores de Shapley. La fórmula que hemos obtenido puede interpretarse como la contribución marginal de i a cada coalición, promediada sobre todas las permutaciones posibles en las que se puede formar la coalición. Resulta más intuitivo mostrarlo con un ejemplo:

Supongamos que dos personas (A y B) se presentan a concurso cuyo primer premio es 10000€, el segundo es 7500€y el tercer premio es 5000€. Juntos, A y B obtienen el primer premio, pero eso no significa que los dos hayan aportado lo mismo y se merezcan la mitad cada uno. Supongamos que podemos conocer los resultados del concurso si cambiamos a estos concursantes. Comprobamos que si el concursante A se presenta solo obtiene el segundo premio, si el concursante B se presenta solo obtiene el tercer premio y si no se presentan ninguno como es lógico no obtienen nada.

El teorema de Shapley afirma que la única distribución justa según los axiomas definidos es la siguiente:

- Calcular las contribuciones marginales de A: si A se une a la coalición vacía tenemos $7500 - 0 = 7500$, si A se une a la coalición $\{B\}$ tenemos $10000 - 5000 = 5000$.
- Promediamos sobre las posibles maneras de unirse a la coalición: obtenemos la contribución de A, y por tanto su recompensa, como $\frac{7500+5000}{2} = 6250$.

Si hacemos lo mismo con B obtendremos el resto del premio a repartir.

3.2.4. Modelos de clasificación

A continuación vamos a introducir la teoría de los valores de Shapley a los modelos de clasificación, por lo que ya no vamos a hablar más de juegos coalicionales. A partir de ahora vamos a tratar las características como los jugadores del juego y vamos a construir una función valor a partir de las predicciones.

Función Característica

Para calcular los valores de Shapley en modelos de clasificación, la función característica v determina la interpretación de la contribución de las características. Es importante señalar que la elección de la función característica puede realizarse de diversas maneras, y esta elección determina directamente los valores de Shapley resultantes. La flexibilidad en su definición permite adaptarse a diferentes necesidades y requisitos de explicabilidad.

Recordemos que estamos usando los valores de Shapley para explicar una predicción particular de x, por lo que vamos a usar la función $v_x(S)$, la cual representa la predicción para los valores de características en el conjunto S marginalizados sobre las características que no están incluidas en S:

$$v_x(S) = \int \hat{f}(x_1, \ldots, x_n)\, d\mathbb{P}_{x \notin S} - E_X(\hat{f}(X))$$

Aquí para la distribución de probabilidad tomamos la distribución empírica y $E_X(\hat{f}(X)) = \sum_x \hat{f}(x) \cdot \mathbb{P}(X = x)$.

En la expresión anterior, realizamos múltiples integraciones para cada característica que no está contenida en S. Un ejemplo concreto sería evaluar la predicción para la coalición S que consiste en los valores de características x_1 y x_3:

$$v_x(S) = v_x(\{x_1, x_3\}) = \int_R \int_R \hat{f}(x_1, X_2, x_3, X_4)\, d\mathbb{P}_{X_2 X_4} - E_X(\hat{f}(X))$$

Valores de Shapley

Aplicando lo anterior y denotando v_x por v, los valores de Shapley se definen en función de los jugadores en S. La contribución del valor de una característica x_j al pago total o predicción se calcula ponderando y sumando sobre todas las posibles combinaciones de valores de características:

$$\phi_j(v) = \sum_{S \subseteq \{x_1, \ldots, x_n\} \setminus \{x_j\}} \frac{|S|!(n - |S| - 1)!}{n!} \left(v(S \cup \{x_j\}) - v(S)\right)$$

Donde S es un subconjunto de las características utilizadas en el modelo, x es el vector de valores de características de la instancia a explicar, donde n es el número total de características.

Propiedades

Finalmente, para concluir esta sección, debemos mencionar que los valores de Shapley (ϕ_j) son únicos en su capacidad para satisfacer las cuatro propiedades, derivadas de los axiomas mencionados antes, que mostramos a continuación: Eficiencia, Simetría, Jugador Nulo y Aditividad.

Eficiencia Las contribuciones de las características deben sumar la diferencia de la predicción para x y el promedio. En la práctica se denota ϕ_0 a $E_X(\hat{f}(X))$.

$$\sum_{j=1}^{n} \phi_j = \hat{f}(x) - E_X(\hat{f}(X))$$

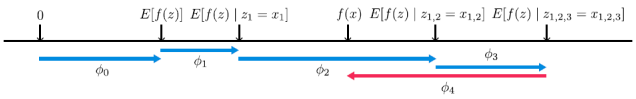


Figura 3.1: Eficiencia valores de Shapley.

Simetría Las contribuciones de dos valores de características j y k deben ser iguales si contribuyen igualmente a todas las coaliciones posibles.

Si $v(S \cup \{x_j\}) = v(S \cup \{x_n\})$ para todas las $S \subseteq \{x_1, \ldots, x_n\} \setminus \{x_j, x_k\}$, entonces $\phi_j = \phi_k$.

Jugador nulo Una característica j que no cambia el valor predicho, independientemente de la coalición a la que se añada, debe tener un valor de Shapley de 0.

Si $v(S \cup \{x_j\}) = v(S)$ para todas las $S \subseteq \{x_1, \ldots, x_n\}$, entonces $\phi_j = 0$.

Aditividad Para un juego con pagos combinados $v + v^+$, los valores de Shapley respectivos son los siguientes:

$$\phi_j + \phi_j^+$$

3.3. SHAP (SHapley Additive exPlanations)

Finalmente llegamos al culmen práctico de LIME y los valores de Shapley. SHAP es muy reciente y fue desarrollado por Lundberg y Lee (2017) en el artículo *A Unified Approach to Interpreting Model Predictions* [5]. Es un método para explicar predicciones individuales basado en los valores de Shapley y, en particular, para la propuesta original de SHAP (KernelSHAP), en LIME. Además, se ha extendido mucho su uso desde entonces (se ha convertido en un estándar) y actualmente también se utiliza para interpretaciones globales del modelo.

3.3.1. Método de atribución de características aditivas

Sea $\hat{f}$ el modelo de predicción original que se desea explicar y g el modelo de explicación. Aquí, nos enfocamos en métodos locales diseñados para explicar una predicción $\hat{f}(\mathbf{x})$ basada en una sola entrada $\mathbf{x}$, como se propone en LIME 3.1. Los modelos de explicación a menudo utilizan menos características y entradas simplificadas $\mathbf{z}'$ que se mapean a las entradas originales. Para ello podemos tomar $M <= n$ características $j_1, \ldots, j_M$ de las originales y definir:

Definición 3.10. Dada una instancia $\mathbf{x} \in \mathbb{R}^n$ y un vector $\mathbf{z} \in \mathbb{R}^M$, denominamos a $\mathbf{z}' \in \{0,1\}^M$ como el *Vector de Coalición* de $\mathbf{z}$ dado $\mathbf{x}$, donde cada entrada representa la presencia o ausencia de una característica en la coalición. Entendemos que la característica j_i-ésima está presente si $x_{j_i} = z_i$.

Definición 3.11. Una *función de mapeo* para la instancia $\mathbf{x}$ es una función $h_x : \{0,1\}^M \to \mathbb{R}^n$ donde $\mathbf{x} = h_x(\mathbf{x}')$, siendo $\mathbf{x}'$ el vector de coalición de $\mathbf{x}$ dado $\mathbf{x}$.

Definición 3.12. Denominamos *método de atribución de características aditivas* a aquellos métodos con un modelo de explicación que es una función lineal de variables binarias, definido como:

$$g(\mathbf{z}') = \phi_0 + \sum_{i=1}^{M} \phi_i z_i'$$

donde $\mathbf{z}'$ es un vector binario de tamaño M y ϕ_i es un valor en $\mathbb{R}$.

Los métodos locales intentan asegurar que $g(\mathbf{z}') \approx \hat{f}(h_x(\mathbf{x}'))$ siempre que $\mathbf{z}' \approx \mathbf{x}'$. Nótese que $h_x(\mathbf{x}') = \mathbf{x}$ aunque $\mathbf{x}'$ puede contener menos información que $\mathbf{x}$ porque h_x es específico para la entrada actual $\mathbf{x}$.

3.3.2. Existencia única bajo propiedades

Precisión Local

La precisión local se expresa como:

$$\hat{f}(x) = g(x') = \phi_0 + \sum_{j=1}^{n} \phi_j x'_j$$

Esto asegura que la función de explicación g sea precisa localmente alrededor de la instancia de interés x.

Ausencia

La propiedad de ausencia se define como:

$$x'_j = 0 \Rightarrow \phi_j = 0$$

Esto significa que una característica ausente recibe una atribución de Shapley igual a cero.

Consistencia

La consistencia se formula como:

$$\phi_j(\hat{f}', x) \geq \phi_j(\hat{f}, x)$$

Esto establece que si el cambio en el modelo aumenta o mantiene constante la contribución marginal de una característica, entonces el valor de Shapley correspondiente también aumenta o permanece constante.

Al igual que con Shapley Values podemos obtener existencia y unicidad de los métodos de atribución de características aditivas con el siguiente teorema:

Teorema 3.13. *Solo hay un modelo de explicación posible g que sigue la Definición 3.12 y satisface las Propiedades 1, 2 y 3:*

$$\phi_i(f; x) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!} [f(h_x(z')) - f(h_x(z' \setminus i))] \tag{3.6}$$

donde $|z'|$ es el número de entradas no nulas en z', y $z' \subseteq x'$ representa todos los vectores z' donde las entradas no nulas son un subconjunto de las entradas no nulas en x'.

3.3.3. SHAP

En dicho artículo [5], se propone SHAP values con el objetivo de explicar la predicción de una instancia x calculando la contribución de cada característica a la predicción. Este método se basa en los valores de Shapley derivados de la teoría de juegos coalicionales. Las características de una instancia actúan como jugadores en una coalición, y los valores de Shapley indican cómo distribuir de manera justa la predicción entre las características.

Para ello se usa la función de expectativa condicional del modelo original, obteniendo la solución a la ecuación 3.6, donde $f(h_x(z')) = E[f(z)|z_S]$, y S es el conjunto de índices no cero en z'. SHAP values proporciona una medida única de importancia de características aditivas que se adhiere a las Propiedades 1-3 y utiliza expectativas condicionales para definir entradas simplificadas. Implícita en esta definición de SHAP values hay un mapeo de entrada simplificado, $h_x(z') = z_S$, donde z_S tiene valores faltantes para características que no están en el conjunto S. Dado que la mayoría de los modelos no pueden manejar patrones arbitrarios de valores de entrada faltantes, aproximamos $f(z_S)$ con $E[f(z)|z_S]$.

La computación exacta de los valores de SHAP es, por su elevado coste compuatcional, desafiante. Sin embargo, gracias a las características de los métodos atribución de características aditivas, Lundberg y Lee proponen como aproximarlos de manera eficiente.

3.3.4. KernelSHAP

KernelSHAP utiliza un enfoque de regresión lineal ponderada para estimar los valores de Shapley, combina el enfoque de explicaciones lineales de LIME con los valores de Shapley para aproximar localmente $\hat{f}$. A primera vista, la formulación de LIME en la Ecuación 3.1 parece muy diferente de la formulación clásica de valores de Shapley en la Ecuación 3.6. Sin embargo, dado que LIME lineal es un método de atribución de características aditivas, sabemos que los valores de Shapley son la única solución posible que satisface las Propiedades 1-3: precisión local, ausencia y consistencia.

Una pregunta natural es si la solución a la Ecuación 3.1 recupera estos valores. La respuesta depende de la elección de la función de pérdida L, el núcleo de ponderación π_x y el término de regularización Ω. Las elecciones de LIME para estos parámetros se hacen de manera heurística; así que podemos afirmar que en general no recupera los valores de Shapley. Una consecuencia es que se viola la precisión local y/o la consistencia, lo que a su vez conduce a un comportamiento no intuitivo en ciertas circunstancias.

En cuanto a los valores adecuados tenemos el siguiente teorema:

Teorema 3.14. *Bajo la definición de LIME como un método de atribución de características aditivas como en la definición 3.12, las formas específicas de π_x, L y Ω que hacen que las soluciones de la Ecuación 3.1 sean consistentes con las Propiedades 1 a 3 son:*

$$\begin{aligned}
\Omega(g) &= 0; \\
\pi_{x'}(z') &= \frac{(M-1)}{\binom{M}{|z'|}|z'|(M-|z'|)} \\
L(f;g;\phi_{x_0}) &= \sum_{z' \in Z} \left(f(h_{x'}^{-1}(z')) - g(z')\right)^2 \pi_{x'}(z')
\end{aligned}$$

donde $|z'|$ es el número de elementos no nulos en z'.

Es importante tener en cuenta que $\pi_x(z') = 1$ cuando $|z'| \in \{0, M\}$, lo que impone $\phi_0 = f_x(\emptyset)$ y $f(x) = \sum_{i=0}^{M} \phi_i$. En la práctica, estos pesos infinitos pueden evitarse durante la optimización mediante la eliminación analítica de dos variables utilizando estas restricciones.

Lo destacable sobre esto es que $g(z')$ se asume que sigue una forma lineal, y L es una pérdida cuadratica. Como consecuencia podemos estimar los valores de Shapley utilizando una regresión lineal ponderada. En la última sección del paper [5] se comenta que estimar conjuntamente todos los valores de SHAP utilizando una regresión proporciona una mejor eficiencia de muestra que el uso directo de ecuaciones de Shapley clásicas.

Estimación

1. Muestreo de coaliciones $z'_k \in \{0,1\}^n, k \in \{1, \ldots, K\}$.
2. Obtención de la predicción para cada z'_k mediante la conversión a espacio de características original y la aplicación del modelo $\hat{f}$.
3. Cálculo del peso para cada z'_k con el núcleo SHAP.
4. Ajuste de un modelo lineal ponderado.
5. Devolución de los valores de Shapley ϕ_k, los coeficientes del modelo lineal.

Se emplea un muestreo estratégico de coaliciones, priorizando aquellas con mayor peso según el núcleo SHAP. Este núcleo, denotado por $\pi_x(z')$, garantiza la ponderación adecuada para obtener valores de Shapley precisos.

En resumen, KernelSHAP combina la teoría de juegos con técnicas de regresión ponderada para proporcionar explicaciones precisas y locales para las predicciones de modelos de aprendizaje automático.

CAPÍTULO 4
Desarrollo práctico

La parte práctica de este TFG, y objetivo de este capítulo, se centra en la implementación de los conceptos teóricos discutidos anteriormente. Aquí, se aplica un modelo de clasificación y técnicas de interpretabilidad a un conjunto de datos real proporcionado por una empresa privada. Esta sección es fundamental, ya que muestra cómo la teoría estudiada puede ser llevada a la práctica y utilizada para obtener interpretaciones específicas.

4.1. Implementación

En este capítulo, hablamos sobre el proceso de implementación realizado. Dado que estamos tratando un TFG de matemáticas, el enfoque se ha dado principalmente en el desarrollo teórico y esta parte ha quedado muy reducida. No obstante, la implementación práctica juega un papel crucial al demostrar la aplicabilidad de los conceptos estudiados.

Con el objetivo de aumentar la claridad, se ha creado un Jupyter Notebook que guía al lector a través de cada paso del proceso de implementación. Este notebook está disponible en el siguiente enlace: Desarrollo en Jupyter-Notebook. A continuación, se presenta una pequeña muestra de lo que hemos obtenido:

4.1.1. Previo

Primero, debemos empezar comentando que partimos de un problema de regresión muy complicado, donde cuesta mucho encontrar relaciones fáciles de interpretar entre las variables que tenemos y el target. Voy a quedarme sólo con la primera base de datos para explicar el funcionamiento de ese modelo.

Vemos en la figura 4.1 que, con tan solo la visualización de los datos que obtenemos representando la variable a predecir junto al resto de variables individualmente, no obtenemos ninguna información.

La base de datos que se está usando venía con los modelos ya entrenados y con los valores calculados de *Feature Importance (FI)* y *Permutation Feature Importance*

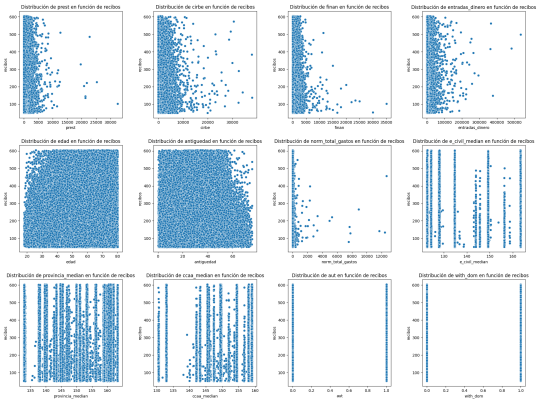
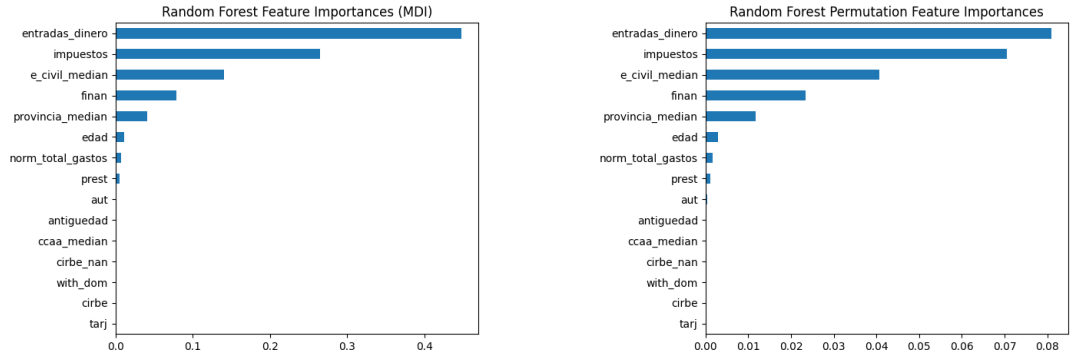


Figura 4.1: Previsualización de los datos.

(PFI). Vemos en las gráficas 4.2 estos valores, que nos indican que características son más importantes para el modelo y que características afectan más a la precisión del modelo respectivamente.



(a) Feature Importance.

(b) Permutation Feature Importance.

Figura 4.2: Resultados importancia de características previo a SHAP.

Los dos principales problemas que tenemos son los siguientes:

- Solo somos capaces de intuir qué variables pueden ser más importantes.
- No tenemos el respaldo teórico que nos ofrece SHAP con todas sus implicaiones.

4.1.2. Interpretabilidad con SHAP

A continuación, vamos a mostar los resultados de interpretabilidad que obtenemos con la librería de SHAP.

Dado que el origen de SHAP es local, debemos empezar mostrando primero los *Force Plots* locales, que son la aplicación directa del capítulo 3.3. Simplemente se muestran los valores SHAP como fuerzas que aumentan o disminuyen el valor base (la experanza del modelo que ya comentamos) luchando entre sí. Esta explicación

es extremadamente sencilla y no requiere ninguna explicación adicional. Vemos tres ejemplos para tres instancias diferentes en las figuras 4.3, 4.4 y 4.5.

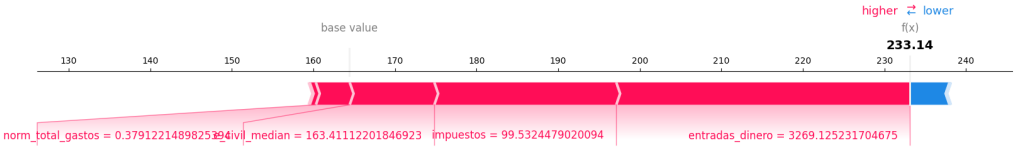


Figura 4.3: Force Plot para una predicción alta.

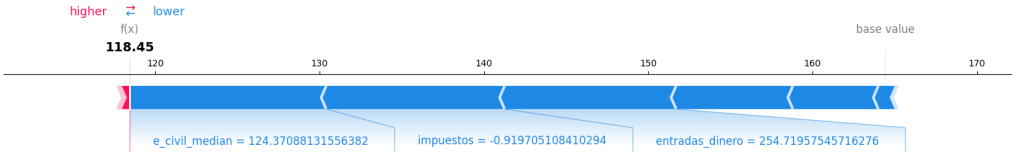


Figura 4.4: Force Plot para una predicción baja.

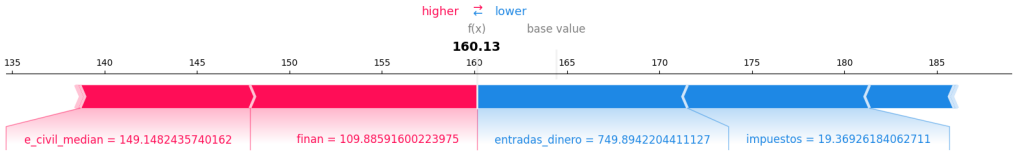


Figura 4.5: Force Plot para una predicción cercana a la predicción base.

Si promediamos los valores absolutos de los valores SHAP de todas las instancias, obtenemos una representación por cada característica de la contribución a las predicciones del modelo, la alternativa de SHAP a FI y PFI. Este gráfico se denomina *Bar Plot* y se muestra en la figura 4.6, podemos comprobar que los resultados son muy similares a los mostrados en el apartado previo.

Otra opción muy interesante que ofrece SHAP es el gráfico *Summary Plot* 4.7, donde se muestran por características las distribuciones de los valores SHAP (la densidad de la distribución en función de la característica se muestra como la superposición de puntos).

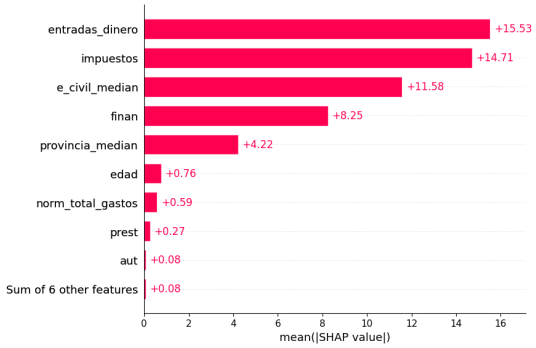


Figura 4.6: Importancia de las características con SHAP.

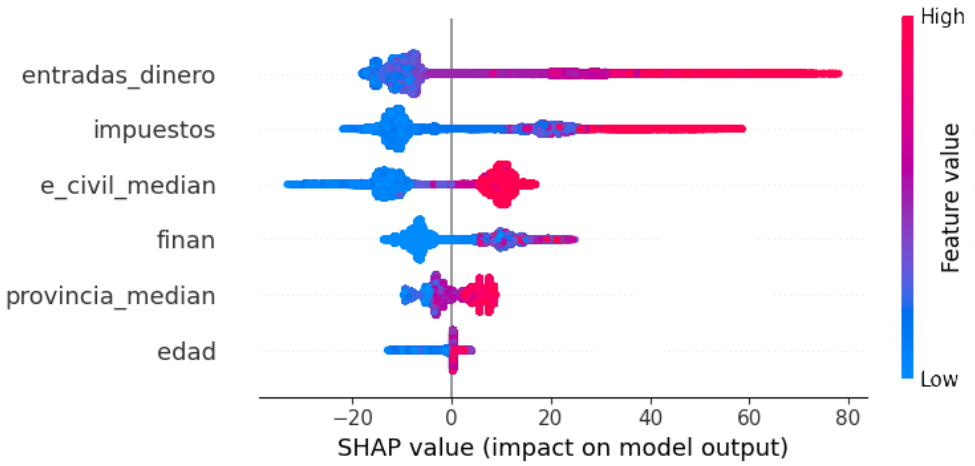
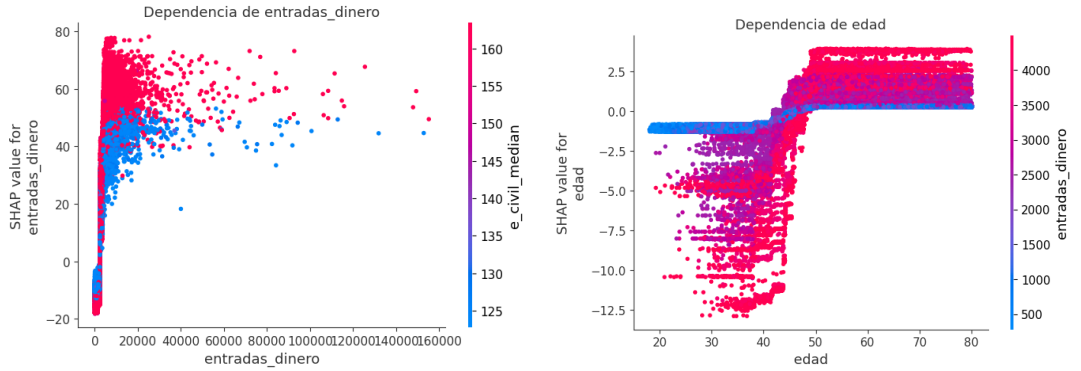


Figura 4.7: Summary Plot SHAP.

Como último gráfico, mostramos los *Dependence Plots* para características específicas que nos genera SHAP de manera fácil, y buscando automáticamente la característica extra que más información puede aportarnos junto a la seleccionada. En la figura 4.8 podemos ver estos gráficos para dos variables.



(a) Dependencia de entradas dinero.

(b) Dependencia de edad.

Figura 4.8: Dependence Plots específicos.

4.2. Evaluación y Conclusiones

Con esta implementación, se muestra no solo la importancia de la precisión en los modelos de clasificación, sino también la necesidad de comprender y confiar en las decisiones que estos modelos producen, contribuyendo así a una toma de decisiones más informada en un problema del campo económico.

SHAP ha sido capaz de darnos explicaciones entendibles de predicciones individuales y del modelo global, con la confianza que aporta la base matemática sobre el reparto justo que hemos discutido previamente.

Explicaciones individuales: Partimos de las predicciones individuales con los **Force Plots** para una única predicción. Con la visualización de fuerzas hemos sido capaces de entender predicciones individuales como el resultado de una competición de fuerza entre las variables que luchan por sumar o restar valor a la predicción. Gracias a las demostraciones vistas en los apartados 3.2 y 3.3 podemos confiar en este reparto de fuerzas como el único reparto justo posible.

Explicaciones globales: A partir de ahí, la librería de SHAP hace uso de estas predicciones individuales para generar explicaciones globales del modelo.

Hemos usado los **Bar Plots** para representar los valores absolutos de las contribuciones promediadas de los SHAP values, pudiendo interpretarlas como cuánto afecta, en promedio, cada característica al juego de la predicción.

Siguiendo con la línea de interpretaciones globales, hemos observado el uso de **Summary Plot** para visualizar de manera sencilla la distribución de los SHAP values respecto a cada variable y los **Dependence Plots** para hacerlo de manera más detallada, con la ventaja extra de poder encontrar fácilmente relaciones entre las características.

Finalmente, aunque no hemos comentado mucho al respecto por ser un gráfico interactivo, SHAP ofrece la posibilidad de visualizar todos los Force Plots individuales de manera conjunta y probando distintas combinaciones para permitirnos buscar más relaciones y explicaciones que no hayamos detectado con los métodos anteriores.

En conclusión, SHAP se ha mostrado como una herramienta poderosa y versátil para la interpretabilidad de modelos de clasificación. Su capacidad para proporcionar información sólida, detallada y robusta sobre las decisiones del modelo supera ampliamente las técnicas tradicionales previas. Esta mejora en la interpretabilidad no solo ayuda a comprender mejor los modelos, sino que también aumenta la confianza en sus predicciones, lo cual es crucial en campos como la economía y la salud. Al proporcionar una base matemática sólida y una implementación práctica accesible, SHAP se posiciona como una herramienta indispensable para cualquier profesional que busque entender y confiar en los modelos de clasificación que utiliza.

4.3. Futuros trabajos

En esta sección, se proponen algunas líneas de investigación y desarrollo futuras que podrían complementar y expandir el trabajo realizado en este TFG.

Ampliación de Técnicas de Interpretabilidad. Además de LIME, los valored de Shapley y SHAP, existen otras técnicas emergentes y complementarias que podrían explorarse. La integración y comparación de estas técnicas con SHAP podría ofrecer un panorama más completo y variado de herramientas de interpretabilidad.

Aplicaciones en Diferentes Sectores. Sería interesante explorar la aplicabilidad de los métodos de interpretabilidad en otros ámbitos como el marketing, la seguridad informática, la energía y las ciencias ambientales. Cada uno de estos campos presenta desafíos y oportunidades únicas.

Automatización y Herramientas de Interpretabilidad. Sin duda, una de las líneas más interesantes es desarrollar herramientas y marcos de trabajo que automaticen el proceso de interpretabilidad y lo integren en el flujo de trabajo de desarrollo de modelos puede ser de gran valor. Además, estas herramientas podrían generar reportes automatizados que expliquen las decisiones del modelo de manera clara y comprensible para lectores no especializados.

Validación y Robustez de Interpretaciones. Es crucial asegurar que las interpretaciones proporcionadas por técnicas como SHAP sean robustas y válidas en diferentes contextos. Futuros trabajos podrían enfocarse en desarrollar métodos para validar y evaluar la calidad de las interpretaciones. Esto podría incluir la creación de métricas específicas para medir la fidelidad y la coherencia de las explicaciones, así como estudios empíricos que comparen la eficacia de diferentes técnicas de interpretabilidad en diversos conjuntos de datos y escenarios.

Consideraciones Éticas y de Sesgo. La interpretabilidad de modelos también tiene importantes implicaciones éticas. Es importante explorar cómo las técnicas de interpretabilidad pueden ayudar a identificar y mitigar sesgos en los modelos de clasificación e incluso desarrollar metodologías para asegurar la equidad y la transparencia en los modelos.

En resumen, hay muchas direcciones prometedoras para continuar el trabajo iniciado en este TFG. Ampliar y mejorar las técnicas de interpretabilidad, explorar su aplicabilidad en diversos sectores, desarrollar herramientas automatizadas, validar las interpretaciones y considerar las implicaciones éticas son solo algunas de las áreas que pueden beneficiar significativamente del avance en esta línea de investigación. Estos futuros trabajos no solo contribuirán a la comprensión y confianza en los modelos de clasificación, sino que también promoverán su uso responsable y ético en la toma de decisiones.

Bibliografía

[1] Molnar, C.: *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. 2019.

[2] Seber, G. A. F.: *Linear Regression Analysis*. Wiley, 1977.

[3] Ribeiro, M. T., Singh, S., y Guestrin, C.: *"Why Should I Trust You?": Explaining the Predictions of Any Classifier*. arXiv:1602.04938 [cs.LG], 2016.

[4] Shapley, L. S.: *A value of n-person games. Contributions to the Theory of Games (1953)*, 307–317.

[5] Lundberg, S. M. and Lee, S.-I.: *A Unified Approach to Interpreting Model Predictions*. Paul G. Allen School of Computer Science, University of Washington, Seattle, WA 98105. slund1@cs.washington.edu, suinlee@cs.washington.edu.

[6] *Logistic Regression and Newton's Method*. 36-402, Advanced Data Analysis. 15 March 2011. *Disponible en:* https://www.stat.cmu.edu/~cshalizi/402/lectures/14-logistic-regression/lecture-14.pdf.

[7] Peter Hoff: *Best Linear Unbiased Estimation*. 2022. *Disponible en:* https://www2.stat.duke.edu/~pdh10/Teaching/721/Materials/ch2blue.pdf.

[8] Hastie, T., Tibshirani, R., y Friedman, J.: *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Segunda Edición, Springer, 2009.

[9] Lundberg, S. M. y otros: *SHAP (SHapley Additive exPlanations) Documentation. Disponible en:* https://shap.readthedocs.io/en/latest/. Accedido por última vez: mayo de 2024.

APÉNDICE A
Método Newton-raphson

A.1. Caso unidimensional

Consideremos la minimización de una función unidimensional $f(\beta)$ para encontrar el mínimo global β^*. Supongamos que f es suave y que β^* es un mínimo interior regular, lo que implica que la derivada en β^* es cero y la segunda derivada es positiva. Cerca del mínimo, podemos realizar una expansión de Taylor:

$$f(\beta) \approx f(\beta^*) + \frac{1}{2}(\beta - \beta^*)^2 \frac{d^2 f}{d\beta^2}\bigg|_{\beta=\beta^*}$$

Newton propone minimizar esta aproximación cuadrática, reemplazando el problema original con uno más manejable. Si elegimos un punto inicial $\beta^{(0)}$ cercano al mínimo, podemos realizar una expansión de Taylor de segundo orden alrededor de $\beta^{(0)}$:

$$f(\beta) \approx f(\beta^{(0)}) + (\beta - \beta^{(0)}) \frac{df}{d\beta}\bigg|_{\beta=\beta^{(0)}} + \frac{1}{2}(\beta - \beta^{(0)})^2 \frac{d^2 f}{d\beta^2}\bigg|_{\beta=\beta^{(0)}}$$

Al minimizar esta expresión, se obtiene una nueva estimación $\beta^{(1)}$. Este proceso iterativo se repite utilizando la nueva estimación para obtener aproximaciones cada vez mejores:

$$\beta^{(n+1)} = \beta^{(n)} - \frac{\frac{df}{d\beta}(\beta^{(n)})}{\frac{d^2 f}{d\beta^2}(\beta^{(n)})}$$

Se puede demostrar que si $\beta^{(0)}$ está lo suficientemente cerca de β^*, entonces $\beta^{(n)} \to \beta^*$ y $|\beta^{(n)} - \beta^*| = O(n^{-2})$, una tasa de convergencia rápida.

A.2. Método de Newton en más de una dimensión

Si la función objetivo f depende de múltiples parámetros $\beta_1, \beta_2, ..., \beta_p$, se agrupan en un vector w. La actualización de Newton en más de una dimensión se expresa como:

$$\beta^{(n+1)} = \beta^{(n)} - [H(\beta^{(n)})]^{-1} \nabla f(\beta^{(n)})$$

donde ∇f es el gradiente de f y H es la hessiana de f, la matriz de segundas derivadas parciales. Calcular H y ∇f generalmente no es costoso, pero calcular la inversa de H sí lo es. Esto ha llevado al desarrollo de métodos de Newton modificados, como los métodos cuasi-Newton, que aproximan H por una matriz diagonal o actualizan la inversa de manera eficiente.

APÉNDICE B
Demostraciones o Cálculos

Proposición 4. *Simplificación de la log-verosimilitud:*

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{k} (y^{(i)} - p(\mathbf{x^{(i)}})) * \mathbf{x^{(i)}}$$

Demostración.

$$\begin{aligned}
l(\boldsymbol{\beta}) &= \sum_{i=1}^{k} y^{(i)} \log\left(\frac{1}{1+\exp\left(-\boldsymbol{\beta}^{\mathsf{T}}\mathbf{x^{(i)}}\right)}\right) + (1-y^{(i)}) \log\left(\frac{\exp\left(-\boldsymbol{\beta}^{\mathsf{T}}\mathbf{x^{(i)}}\right)}{1+\exp\left(-\boldsymbol{\beta}^{\mathsf{T}}\mathbf{x^{(i)}}\right)}\right) \\
&= \sum_{i=1}^{k} y^{(i)} \left[\log\left(\frac{1}{1+\exp\left(-\boldsymbol{\beta}^{\mathsf{T}}\mathbf{x^{(i)}}\right)}\right) - \log\left(\frac{\exp\left(-\boldsymbol{\beta}^{\mathsf{T}}\mathbf{x^{(i)}}\right)}{1+\exp\left(-\boldsymbol{\beta}^{\mathsf{T}}\mathbf{x^{(i)}}\right)}\right)\right] + \log\left(\frac{\exp\left(-\boldsymbol{\beta}^{\mathsf{T}}\mathbf{x^{(i)}}\right)}{1+\exp\left(-\boldsymbol{\beta}^{\mathsf{T}}\mathbf{x^{(i)}}\right)}\right) \\
&= \sum_{i=1}^{k} y^{(i)} [\log\left(\exp\left(\boldsymbol{\beta}^{\mathsf{T}}\mathbf{x^{(i)}}\right)\right] + \log\left(\frac{\exp\left(-\boldsymbol{\beta}^{\mathsf{T}}\mathbf{x^{(i)}}\right)}{1+\exp\left(-\boldsymbol{\beta}^{\mathsf{T}}\mathbf{x^{(i)}}\right)} * \frac{\exp\left(\boldsymbol{\beta}^{\mathsf{T}}\mathbf{x^{(i)}}\right)}{\exp\left(\boldsymbol{\beta}^{\mathsf{T}}\mathbf{x^{(i)}}\right)}\right) \\
&= \sum_{i=1}^{k} y^{(i)} \boldsymbol{\beta}^{\mathsf{T}}\mathbf{x^{(i)}} + \log\left(\frac{1}{1+\exp\left(\boldsymbol{\beta}^{\mathsf{T}}\mathbf{x^{(i)}}\right)}\right) \\
&= \sum_{i=1}^{k} (y^{(i)} - p(\mathbf{x^{(i)}})) * \mathbf{x^{(i)}}
\end{aligned}$$

□

Proposición 5. *Cálculo de la diferencial de la log-verosimilitud:*

$$\nabla_{\boldsymbol{\beta}}(l(\boldsymbol{\beta})) = \sum_{i=1}^{k} (y^{(i)} - p(\mathbf{x^{(i)}})) * \mathbf{x^{(i)}}$$

Demostración.

$$\begin{aligned}
\nabla_{\boldsymbol{\beta}} l &= \nabla_{\boldsymbol{\beta}} \sum_{i=1}^{k} y^{(i)} \boldsymbol{\beta}^{\mathsf{T}} \mathbf{x}^{(\mathbf{i})} - \log\left(1 + \exp\left(\boldsymbol{\beta}^{\mathsf{T}} \mathbf{x}^{(\mathbf{i})}\right)\right) \\
&= \sum_{i=1}^{k} \nabla_{\boldsymbol{\beta}} [y^{(i)} \boldsymbol{\beta}^{\mathsf{T}} \mathbf{x}^{(\mathbf{i})} - \log\left(1 + \exp\left(\boldsymbol{\beta}^{\mathsf{T}} \mathbf{x}^{(\mathbf{i})}\right)\right)] \\
&= \sum_{i=1}^{k} \nabla_{\boldsymbol{\beta}} [y^{(i)} \boldsymbol{\beta}^{\mathsf{T}} \mathbf{x}^{(\mathbf{i})}] - \nabla_{\boldsymbol{\beta}} [\log\left(1 + \exp\left(\boldsymbol{\beta}^{\mathsf{T}} \mathbf{x}^{(\mathbf{i})}\right)\right)] \\
&= \sum_{i=1}^{k} y^{(i)} \mathbf{x}^{(i)} - \left[\frac{1}{1 + \exp\left(\boldsymbol{\beta}^{\mathsf{T}} \mathbf{x}^{(i)}\right)} * \exp\left(\boldsymbol{\beta}^{\mathsf{T}} \mathbf{x}^{(i)}\right) \mathbf{x}^{(i)}\right] \\
&= \sum_{i=1}^{k} y^{(i)} \mathbf{x}^{(i)} - \left[\frac{1}{1 + \exp\left(-\boldsymbol{\beta}^{\mathsf{T}} \mathbf{x}^{(i)}\right)} * \mathbf{x}^{(i)}\right] \\
&= \sum_{i=1}^{k} y^{(i)} \mathbf{x}^{(i)} - p(\mathbf{x}^{(\mathbf{i})}) \mathbf{x}^{(i)}
\end{aligned}$$

□

Proposición 6. *Cálculo del hessiano de la log-verosimilitud:*

$$H_l = \sum_{i=1}^{k} p(\mathbf{x}^{(\mathbf{i})})(1 - p(\mathbf{x}^{(\mathbf{i})})) \mathbf{x}^{(i)\mathsf{T}} \mathbf{x}^{(i)}$$

Demostración.

$$\begin{aligned}
H_l &= \nabla_{\boldsymbol{\beta}} (\nabla_{\boldsymbol{\beta}} l) \\
&= \nabla_{\boldsymbol{\beta}} \sum_{i=1}^{k} (y^{(i)} - p(\mathbf{x}^{(\mathbf{i})}))) * \mathbf{x}^{(\mathbf{i})} \\
&= \sum_{i=1}^{k} \nabla_{\boldsymbol{\beta}} [p(\mathbf{x}^{(\mathbf{i})}) * \mathbf{x}^{(\mathbf{i})}] \\
&= \sum_{i=1}^{k} \nabla_{\boldsymbol{\beta}} \left[\frac{1}{1 + \exp\left(-\boldsymbol{\beta}^{\mathsf{T}} \mathbf{x}^{(i)}\right)} * \mathbf{x}^{(\mathbf{i})}\right] \\
&= \sum_{i=1}^{k} \left[\frac{1}{1 + \exp\left(-\boldsymbol{\beta}^{\mathsf{T}} \mathbf{x}^{(i)}\right)}\right]^2 \exp\left(-\boldsymbol{\beta}^{\mathsf{T}} \mathbf{x}^{(i)}\right) (\mathbf{x}^{(\mathbf{i})}) \mathbf{x}^{(\mathbf{i})} \\
&= \sum_{i=1}^{k} \left[\frac{1}{1 + \exp\left(-\boldsymbol{\beta}^{\mathsf{T}} \mathbf{x}^{(i)}\right)}\right] \left[\frac{\exp\left(-\boldsymbol{\beta}^{\mathsf{T}} \mathbf{x}^{(i)}\right)}{1 + \exp\left(-\boldsymbol{\beta}^{\mathsf{T}} \mathbf{x}^{(i)}\right)}\right] \mathbf{x}^{(\mathbf{i})\mathsf{T}} \mathbf{x}^{(\mathbf{i})} \\
&= \sum_{i=1}^{k} p(\mathbf{x}^{(\mathbf{i})})(1 - p(\mathbf{x}^{(\mathbf{i})})) \mathbf{x}^{(i)\mathsf{T}} \mathbf{x}^{(i)}
\end{aligned}$$

□

Proposición 7. $\sum_{r=t}^{s}(-1)^{r-t}\binom{s-t}{r-t}$ *se anula excepto para* $s{=}t$.

Demostración. No tenemos en cuenta el caso $t = 0$ pues entonces $T = \emptyset$ y se anula fuera de esta expresión, ni el caso $s = t$ que es trivial comprobar que se iguala a 1.

$$\left[\sum_{r=t}^{s}(-1)^{r-t}\binom{s-t}{r-t}\right] = \left[\sum_{r=0}^{s-t}(-1)^{r}\binom{s-t}{r}\right] = \left[\sum_{r=0}^{k}(-1)^{r}\binom{k}{r}\right] \quad (k = s-t)$$

Aplicando el binomio de Newton $(x+y)^k = \sum_{r=0}^{k}\binom{k}{r}x^{k-r}y^r$ a $0 = (-1+1)^k$,

$$0 = (-1+1)^k = \sum_{r=0}^{k}\binom{k}{r}(-1)^{k-r}1^r = \sum_{r=0}^{k}\binom{k}{r}(-1)^{k-r}$$

Finalmente,

$$(-1)^{-k}\cdot 0 = (-1)^{-k}\sum_{r=0}^{k}(-1)^{k-r}\binom{k}{r}$$

lo que implica, usando que $(-1)^{-r} = (-1)^r$, que

$$0 = \sum_{r=0}^{k}(-1)^{r}\binom{k}{r}$$

□