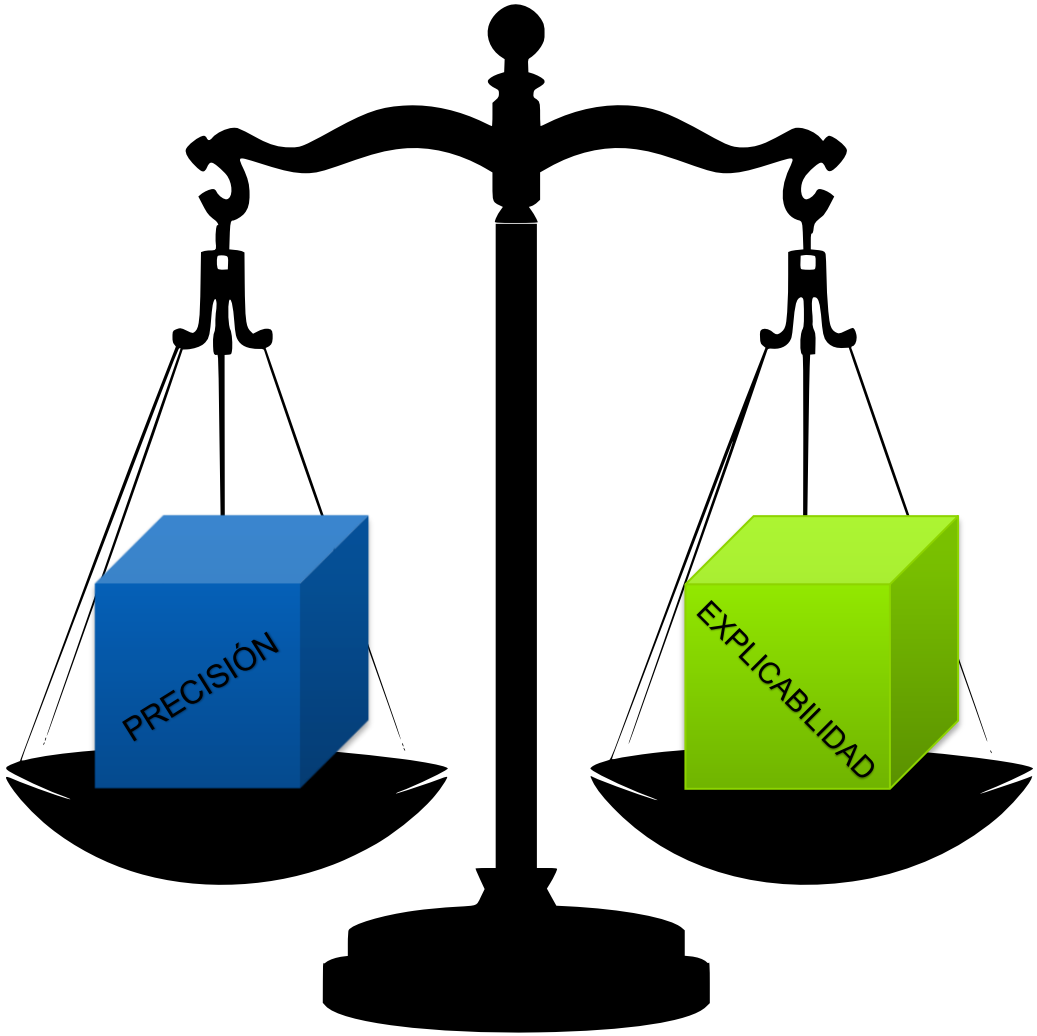
Explicabilidad de modelos

Álvaro Barbero Jiménez
alvaro.barbero.jimenez@gmail.com

La disyuntiva



Permutation importance

Podemos estimar la relevancia de una variable v en un modelo f según cuánto aumente el error de la predicción si la variable se elimina. Formalmente:

$$VI(v) = \sum_{(x,y)\in D} L(\mathbb{E}_{x_v}\left[f(x)|x_{\setminus v}\right], y) - \sum_{(x,y)\in D} L(f(x), y)$$

Donde D es un dataset (idealmente de validación), L es una función de pérdida, y $\mathbb{E}_{x_v}[f(x)|x_{\setminus v}]$ es la esperanza condicionada del modelo sujeta a todas las variables menos v. Equivalentemente, la salida esperada del modelo integrada sobre toda la distribución de posibles valores de x_v

$$\mathbb{E}_{x_v}\left[f(x)|x_{\setminus v}\right] = \int f(x)dx_v$$

Problema ¿Cómo calcular esta integral?

Permutation importance

Puede aproximarse la esperanza condicionada mediante muestreo:

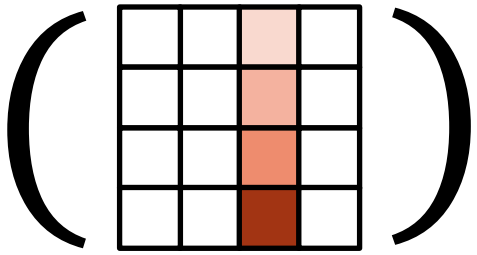
$$\mathbb{E}_{x_v}\left[f(x)|x_{\setminus v}\right] = \int f(x)dx_v \simeq \frac{1}{n}\sum_{i=1}^{n} f([x_{\setminus v}, z_v])$$

Donde $x_{\setminus \mathrm{v}}$ es el dato x sin la variable v, z_v es un valor de la variable v tomado aleatoriamente de entre los datos. Mientras más muestras n se empleen, más precisa será la estimación.

Ojo La aproximación anterior asume que las variables son independientes entre sí. Si esto no se cumple, las estimaciones de relevancia estarán sesgadas.

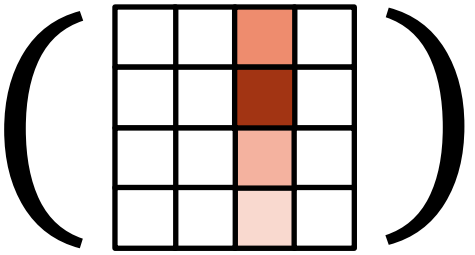
Permutation importance

Un procedimiento práctico y sencillo para calcular relevancias con este método es permutar la columna del Dataframe con la variable a analizar, y comparar el error antes y después de la permutación.

Error 

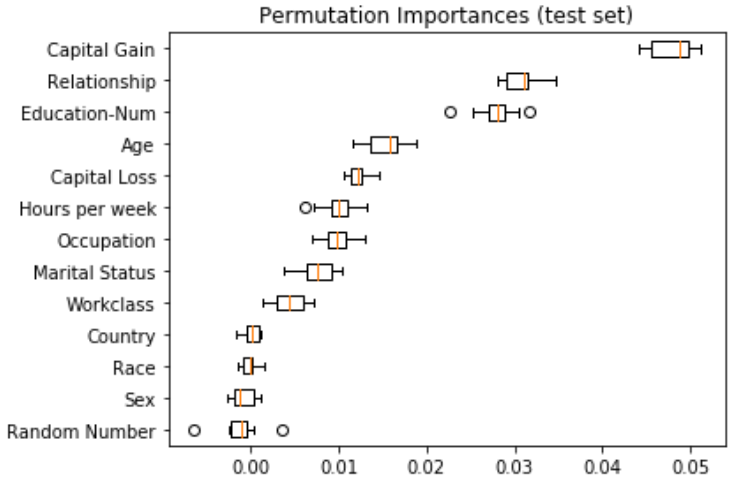
Original data

VS

Error 

Permuted x_v data

Puede repetirse varias veces la permutación sobre una misma columna para aumentar la precisión de la estimación.



Partial Dependence Plots

Además de saber que una variable es relevante en un modelo, puede ser de interés saber cómo influye la variable en el modelo. Para ello podemos usar las gráficas de dependencias parciales.

Procedimiento para extraer la gráfica de una variable x_v:

- Generar una lista de valores $[v_1, v_2, \ldots v_n]$ en el rango de x_v donde queremos medir la influencia sobre el modelo. (Eje x de la gráfica)

- Calcular la salida esperada del modelo en cada uno de estos valores, integrando sobre todas las demás variables. (Eje y de la gráfica)

$$\mathbb{E}_{x_{\setminus v}}\left[f(x)|x_v = v_i\right] = \int f([v_i, x_{\setminus v}])dx_{\setminus v}$$

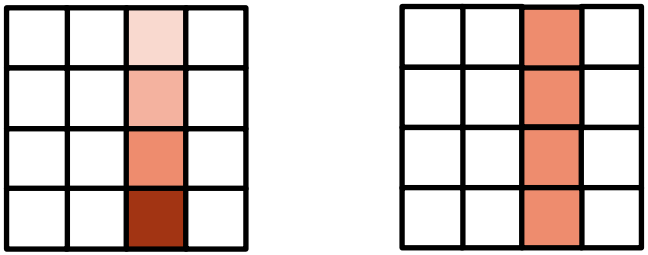
!!! De nuevo, no queremos calcular integrales. ¡Pero podemos aproximarlas por muestreo!

Partial Dependence Plots

En la práctica seguimos el siguiente procedimiento para extraer la gráfica de una variable x_v:

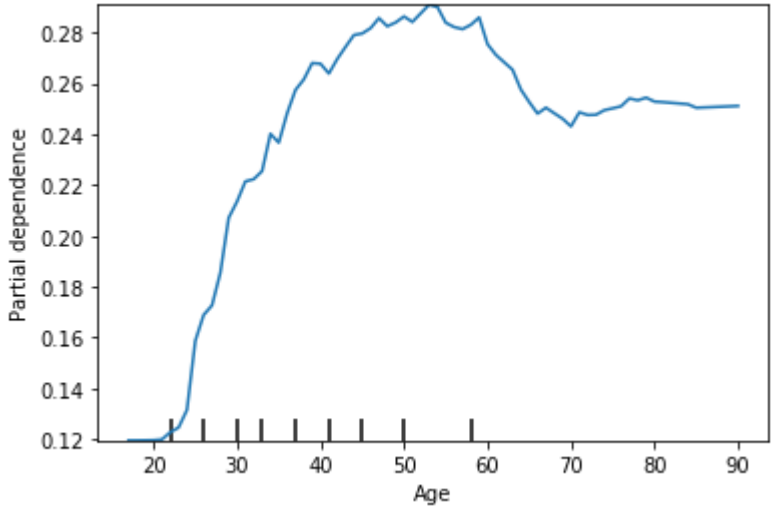
- Generar una lista de valores $[v_1, v_2, \ldots v_n]$ en el rango de x_v donde queremos medir la influencia sobre el modelo. (Eje x de la gráfica)

- Para cada valor v_i, copiamos este valor en la variable x_v de todas las filas del dataset, y calculamos la predicción media del modelo. (Eje y de la gráfica)



Original data

Clamped data



Local Surrogates

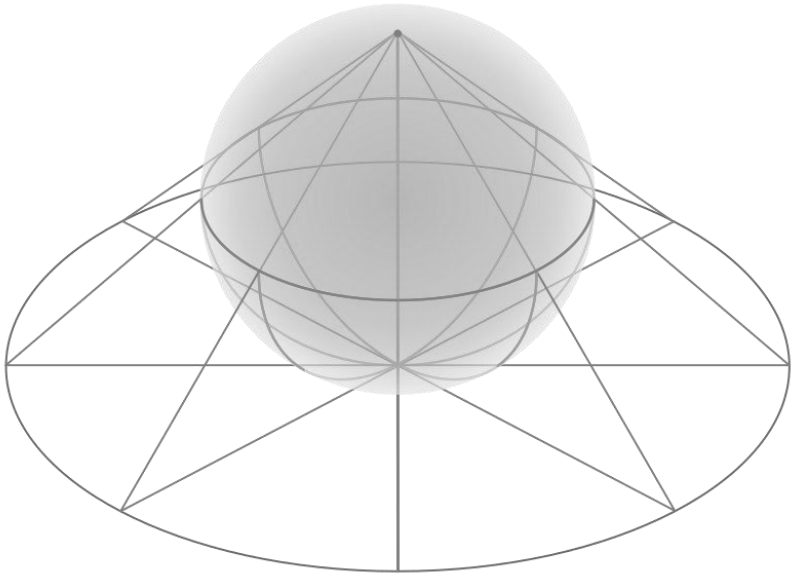
En un modelo no lineal complejo f es imposible obtener explicaciones que sean al mismo tiempo simples (lineales) y representativas del funcionamiento real del modelo.

Pero es posible obtener explicaciones sencillas de una parte concreta de f, esto es, en un dato concreto y su entorno más cercano.

Local Surrogates: entrenar un modelo sencillo con el dato para el que queremos obtener explicabilidad, y otros datos en su proximidad.

- Lasso
- Árbol de decisión pequeño

En general, entrenar un modelo de explicación g que aproxime f en un entorno local



https://commons.wikimedia.org/wiki/File:Stereographic_projection_in_3D.svg

Local Surrogates: LIME

Local Interpretable Model-Agnostic Explantions (LIME) es una forma de aplicar esta técnica. Se entrena un modelo local g de la forma

$$\arg\min_{g \in G} L(f, g, \pi_x) + \Omega(g)$$

Con $L(f, g, \pi_x)$ una función de pérdida que mide la diferencia entre el modelo real f y el modelo explicativo g sobre los datos de un vecindario π_x en torno al dato a explicar x. $\Omega(g)$ es un regularizador que puede añadirse para forzar simplicidad en el modelo explicativo.

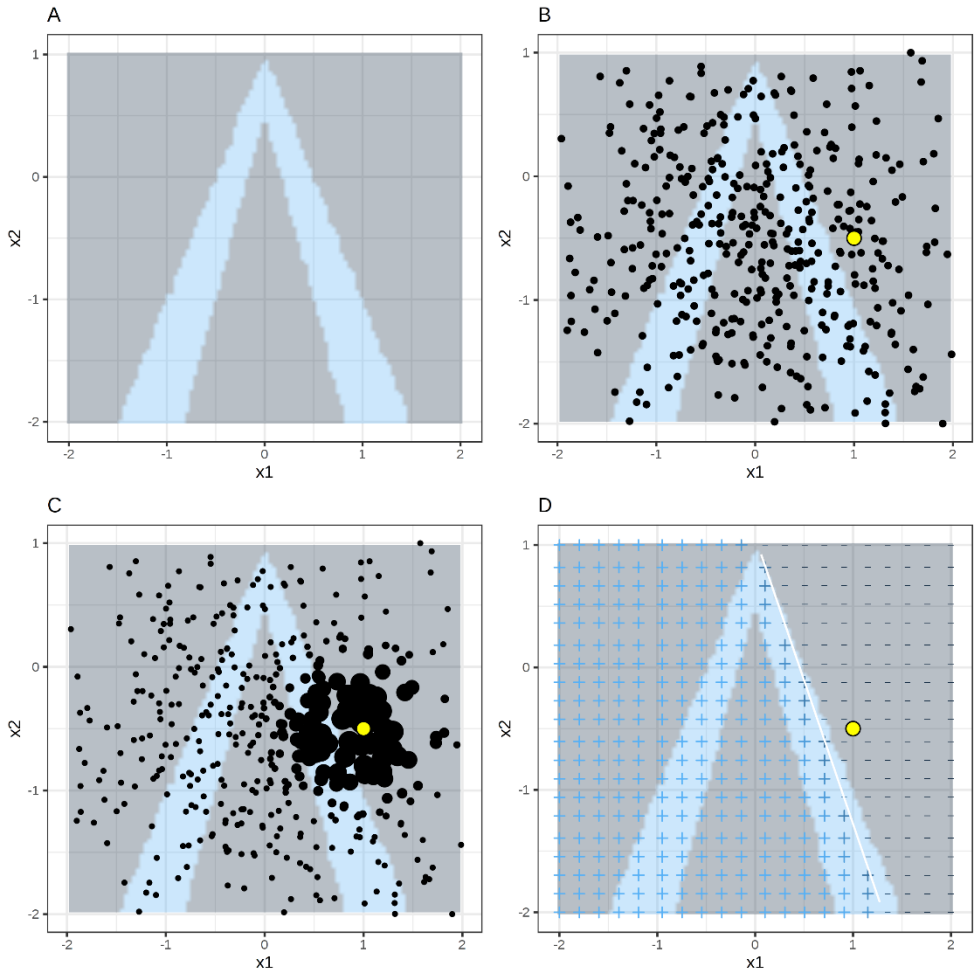
En general suele usarse

$$L(f, g, \pi_x) = \sum_{z \in Z(x)} \pi_x(z)(f(z) - g(z)) \qquad \pi_x(z) = e^{\left(\frac{-D(x,z)^2}{\sigma^2}\right)}$$

Con $Z(x)$ un conjunto de muestras en torno a x, D algún tipo de función de distancia (euclídea, coseno)

Ribeiro et al - "Why Should I Trust You?" Explaining the Predictions of Any Classifier

Local Surrogates: LIME



LIME algorithm for tabular data.

A) Random forest predictions given features x1 and x2. Predicted classes: 1 (dark) or 0 (light).

B) Instance of interest (big dot) and data sampled from a normal distribution (small dots).

C) Assign higher weight to points near the instance of interest.

D) Signs of the grid show the classifications of the locally learned model from the weighted samples. The white line marks the decision boundary (P(class=1) = 0.5).

Christoph Molnar - Interpretable Machine Learning - https://christophm.github.io/interpretable-ml-book/

MEANWHILE IN

BOULDER,

COLORADO

(o cualquier otro sitio random)


SHOWTIME

SHOWTIME

Alicia, Borja y Carlos participan en un concurso de televisión de preguntas y respuestas. Resultan ganadores, y entre los tres consiguen un premio de 900€.

Alicia

Borja

Carlos

¿Cómo deben repartirse el premio de manera justa?

¿Y si uno de los tres ha contribuido más a la victoria que el resto?

https://www.iic.uam.es/innovacion/explicabilidad-algoritmica-metodo-shap/

SHOWTIME

Una distribución más justa del premio debería tener en cuenta cuánto ha contribuido cada miembro del equipo a obtener el premio. Para ello, en primer lugar definimos una **coalición** o subconjunto de jugadores, y medimos cuál habría sido el importe del premio en cada caso.

Coalición	Premio
∅	0 €
A	600 €
B	200 €
C	500 €
A, B	700 €
A, C	800 €
B, C	700 €
A, B, C	900 €

Alicia

Borja

Carlos

Al trabajar en equipo puede conseguirse un mayor premio, ya que cada concursante sabe contestar a preguntas diferentes. ¡Pero no todos los concursantes contribuyen en igual medida! ¿Cómo puede medirse esto formalmente?

https://www.iic.uam.es/innovacion/explicabilidad-algoritmica-metodo-shap/

SHOWTIME

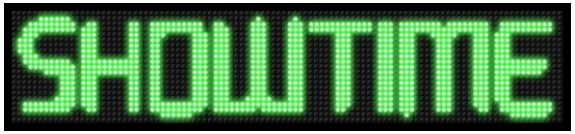
Alicia

Vamos a calcular el valor de **A**licia como concursante. Para ello, consideramos todas las posibles coaliciones en las que Alicia participa, y medimos en cuánto se reduce el valor del premio si ella no hubiera participado.

Coalición	Premio
∅	0 €
A	600 €
B	200 €
C	500 €
A, B	700 €
A, C	800 €
B, C	700 €
A, B, C	900 €

Coalición	Coalición sin Alicia	Premio coalición	Premio sin Alicia	Contribución de Alicia
A	∅	600 €	0 €	600 €
A, B	B	700 €	200 €	500 €
A, C	C	800 €	500 €	300 €
A, B, C	B, C	900 €	700 €	200 €

https://www.iic.uam.es/innovacion/explicabilidad-algoritmica-metodo-shap/



Alicia

Ahora promediamos las contribuciones de Alicia de la siguiente manera:

- Agrupamos sus contribuciones según el tamaño de la coalición
- Calculamos la media de sus contribuciones en grupos con el mismo tamaño de coalición
- Calculamos la media entre tamaños de coalición

Coalición	Contribución de Alicia	Tamaño de coalición	Media por tamaño	Media entre tamaños
A	600 €	1	600 €	400 €
A, B	500 €	2	400 €	
A, C	300 €	2		
A, B, C	200 €	3	200 €	

Alicia **Borja** **Carlos**

Repitiendo el cálculo de forma análoga para Borja y Carlos, obtenemos el reparto más justo de los 900€ de premio.

https://www.iic.uam.es/innovacion/explicabilidad-algoritmica-metodo-shap/

Shapley values

El concepto que acabemos de ver se conoce como valores de Shapley.

Suponer un juego de un conjunto de elementos N (tamaño n) de posible jugadores. Para participar en el juego debe elegirse una coalición (subconjunto) S de jugadores, y el valor que obtiene esta coalición viene dado por una función $v(S)\colon 2^n \to \mathbb{R}$, con $v(\emptyset) = 0$. El coeficiente de Shapley de cada jugador viene dado por:

Promedio entre todos los tamaños posible de coalición

Aporte medio del jugador i entre todas las coaliciones de tamaño k

$$\phi_i(v) = \frac{1}{n}\sum_{k=0}^{n-1} \frac{1}{\binom{n-1}{k}} \sum_{\substack{S \subseteq N \setminus \{i\} \\ |S|=k}} \left(v(S \cup \{i\}) - v(S)\right)$$

Aporte del jugador i a la coalición S

Número de coaliciones de tamaño k escogidas entre $n-1$ jugadores (todos menos el i)

https://en.wikipedia.org/wiki/Shapley_value

Shapley values

La expresión puede comprimirse de la siguiente manera

$$\phi_i(v) = \frac{1}{n}\sum_{k=0}^{n-1}\frac{1}{\binom{n-1}{k}}\sum_{\substack{S\subseteq N\setminus\{i\}\\|S|=k}}(v(S\cup\{i\})-v(S))$$

Definición de número combinatorio

$$= \sum_{k=0}^{n-1}\frac{1}{n}\frac{1}{\frac{(n-1)!}{k!(n-1-k)!}}\sum_{\substack{S\subseteq N\setminus\{i\}\\|S|=k}}(v(S\cup\{i\})-v(S))$$

Unir n con $(n-1)!$ en $n!$
Reorganizar términos

$$= \sum_{k=0}^{n-1}\frac{k!(n-k-1)!}{n!}\sum_{\substack{S\subseteq N\setminus\{i\}\\|S|=k}}(v(S\cup\{i\})-v(S))$$

Unificar sumatorios

$$= \sum_{S\subseteq N\setminus\{i\}}\frac{|S|!(n-|S|-1)!}{n!}(v(S\cup\{i\})-v(S))$$

https://en.wikipedia.org/wiki/Shapley_value

Shapley values

Los valores de Shapley tienen propiedades de utilidad para calcular relevancias:

Eficiencia: la suma de valores de Shapley de los jugadores es igual al valor total del juego, $\sum_{i \in N} \phi_i = v(N)$

Simetría: si hay dos jugadores i, j equivalentes, sus valores de Shapley son iguales. Esto es, si $v(S \cup \{i\}) = v(S \cup \{j\})\ \forall\ S$ tal que $i, j \notin S$, entonces $\phi_i(v) = \phi_j(v)$

Aditividad: si la función de valor del juego se descompone linealmente, los valores de Shapley también. $\phi_i(v + w) = \phi_i(v) + \phi_i(w)$

Jugador nulo: si un jugador nunca aporta nada, su valor de Shapley es 0. Esto es, si $v(S \cup \{i\}) = v(S)\ \forall\ S$ tal que $i \notin S$, entonces $\phi_i(v) = 0$.

Está demostrado que los valores de Shapley son la única manera de asignar valor a los jugadores de forma que se cumplan estas 4 propiedades simultáneamente.

VUELTA AL TEMA

Shapley en modelos predictivos

En modelos predictivos, podemos calcular los valores de Shapley de cada variable explicativa, midiendo así cuánto aportan a la predicción. Para ello, se entrenan tantos modelos como posibles subconjuntos de variables, y se calculan los valores como

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(n - |S| - 1)!}{n!} (v(S \cup \{i\}) - v(S))$$

donde $v(S)$ puede ser el acierto en validación que tiene el modelo al ser entrenado con el subconjunto de variables S.

- Impráctico de calcular salvo con pocos datos y variables

En su lugar, podemos calcular de manera aproximada el valor de los coeficientes de Shapley

SHAP

SHapley Additive exPlanations es un método para calcular coeficientes de Shapley en un modelo de explicación local. Unifica LIME y los coeficientes de Shapley. Para ello se define el modelo de explicación local

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

Donde $z' \in \{0,1\}^M$ es un vector binario que indica qué coaliciones (variables o grupos de variables) están activas en la explicación. ϕ_0 es la predicción media del modelo sin considerar las variables, y cada ϕ_i indica qué añade cada coalición a la predicción si se utiliza en el modelo.

Podemos otorgar a este modelo las propiedades de eficiencia, simetría, linealidad y jugador nulo si forzamos que los ϕ_i sean coeficientes de Shapley. Estos coeficientes se conocen como valores SHAP.

¿Cómo calcularlos eficientemente?

Lundberg and Lee - A Unified Approach to Interpreting Model Predictions

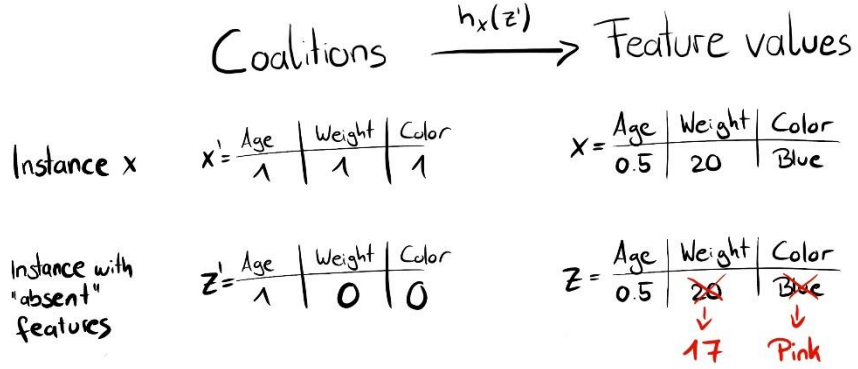
Kernel SHAP

Partiendo de la aproximación de LIME, se calcula el modelo de regresión lineal con pesos siguiente

$$\arg\min_{g \in G} L(f, g, \pi_x) + \Omega(g) \qquad g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i$$

$$L(f, g, \pi_x) = \sum_{z' \in Z(x)} \pi_x(z')(f(h_x(z')) - g(z'))^2 \qquad \pi_x(z') = \frac{(M-1)}{\binom{M}{|z'|} |z'|(M - |z'|)}$$

Donde $Z(x)$ es un dataset con todas las posibles combinaciones de coaliciones, y $h_x(z')$ es una función que genera un patrón con las variables originales de x para las coaliciones activas, y valores tomados aleatoriamente de entre todo el dataset para las coaliciones inactivas.



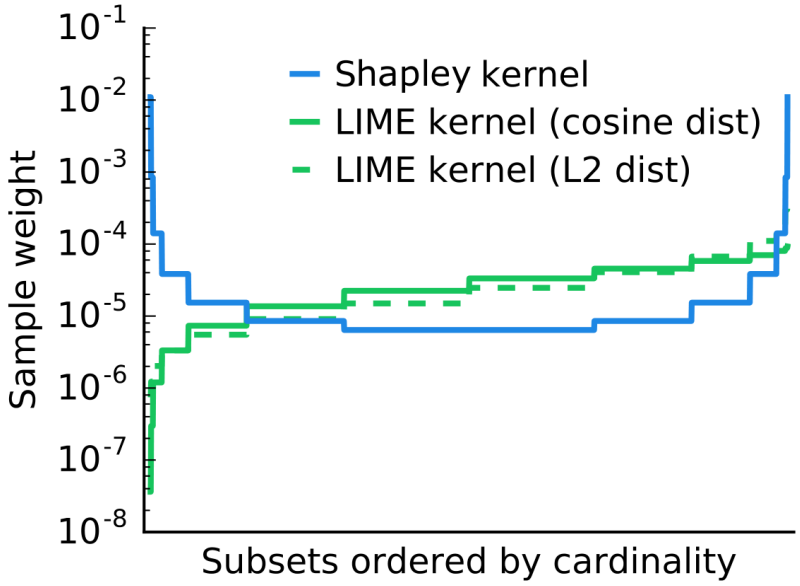
Ojo el procedimiento de muestreo asume independencia entre variables

Christoph Molnar - Interpretable Machine Learning - https://christophm.github.io/interpretable-ml-book/

Kernel SHAP

$$\pi_x(z') = \frac{(M-1)}{\binom{M}{|z'|}|z'|(M-|z'|)}$$

Calcular el modelo de regresión anterior para todas las posibles combinaciones de variable es muy costoso. En su lugar se pueden muestrear las coaliciones con mayor peso en la función de kernel π_x



Las coaliciones con mayor peso son:

- Solo 1 elemento, o M-1 elementos
- Solo 2 elementos, o M-2 elementos
- …

Intuitivamente, se aprende más sobre el aporte de una variable si se usa sola, o se usan todas menos esa.

En la práctica indicamos un “presupuesto” de cuántas muestras vamos a tomar para calcular SHAP, y centramos ese presupuesto en las coaliciones de mayor peso.

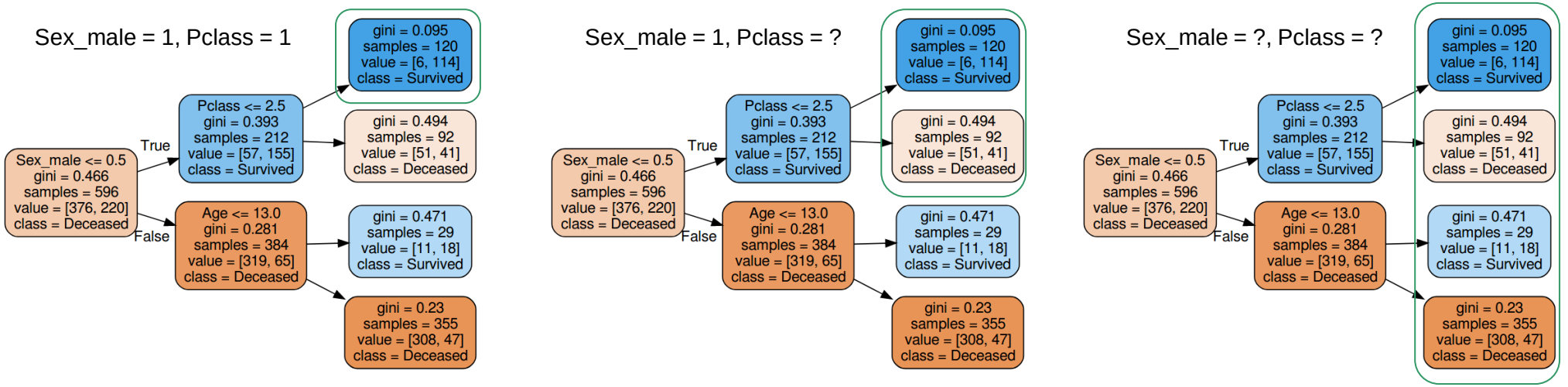
Lundberg and Lee - A Unified Approach to Interpreting Model Predictions

Tree SHAP

En el caso de un ensemble de árboles, es posible calcular los valores de SHAP de forma exacta y con coste polinómico → algoritmo TreeSHAP

Clave 1: por la aditividad de los valores de Shapley, los valores SHAP de un ensemble son la suma de valores SHAP de cada uno de sus árboles.

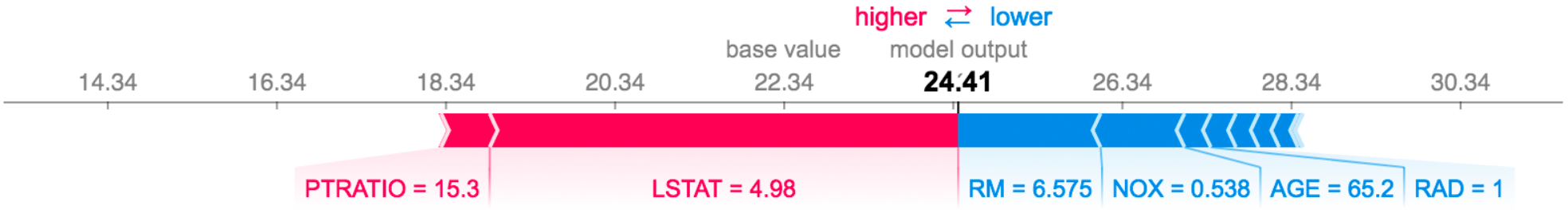
Clave 2: las predicciones esperadas cuando se desactivan variables pueden calcularse de forma exacta en un árbol, promediando las predicciones de todos los nodos hoja a los que podría llegar el dato. (Casos fraccionales)



Lundberg et al - Consistent Individualized Feature Attribution for Tree Ensembles

Librería SHAP

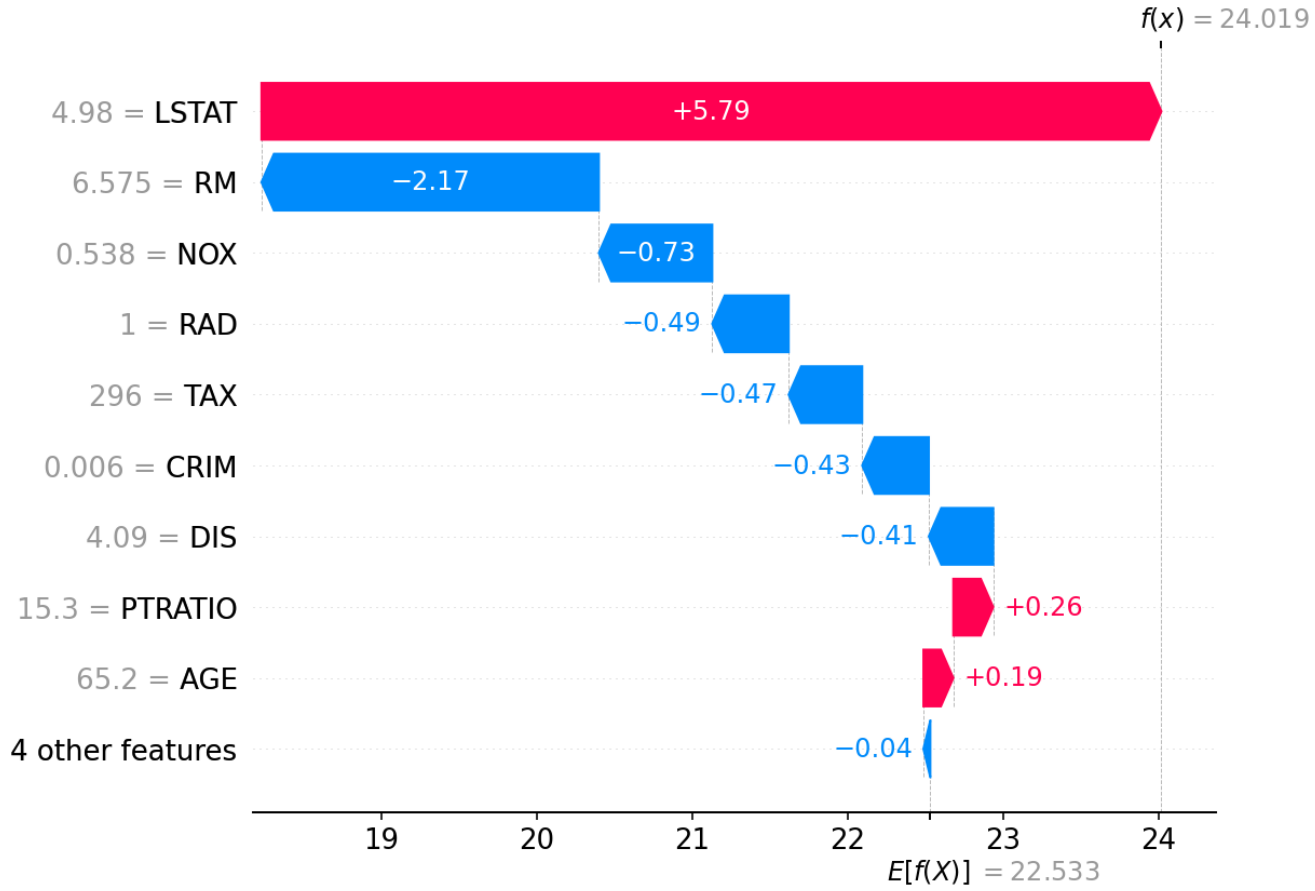
La librería SHAP utiliza la técnica de KernelSHAP (y otras) para convertir un dataset en una matriz de valores de SHAP. Para cada patrón, obtenemos los valores SHAP que representan la importancia de cada variable en la predicción de ese patrón. Eso nos permite crear gráficas de fuerza para cada patrón:



SHAP - https://github.com/slundberg/shap

Librería SHAP

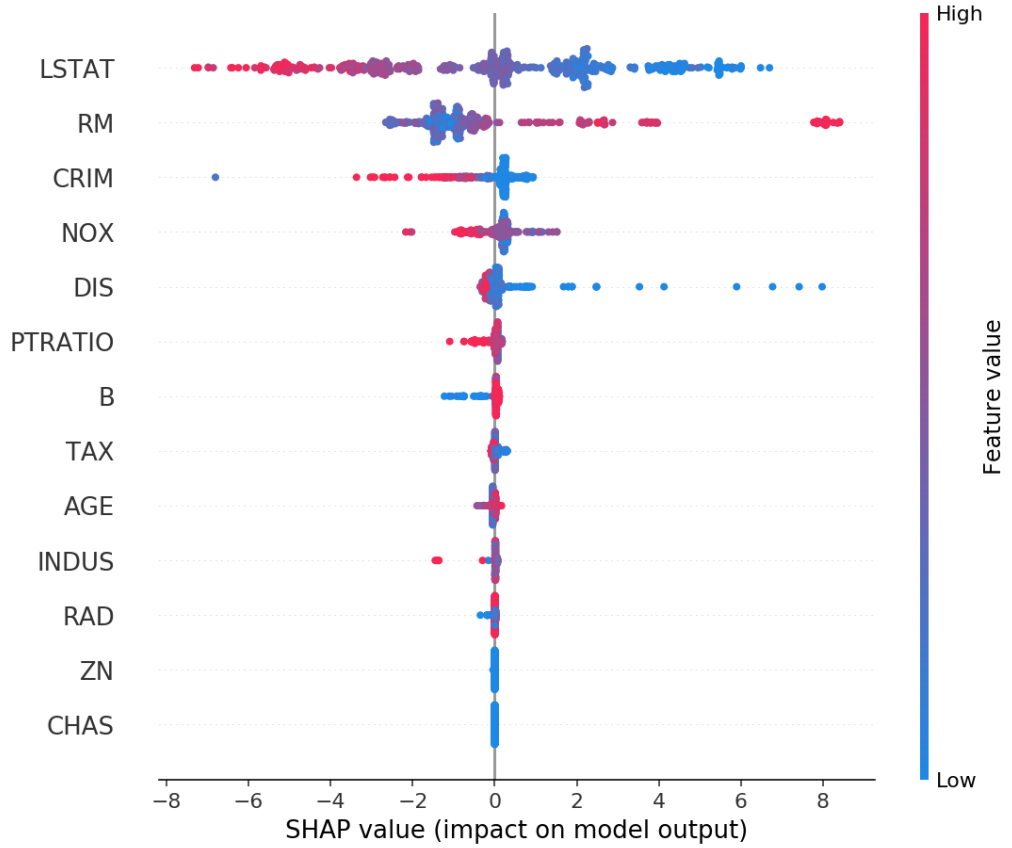
Las gráficas de cascada son otra manera de representar los coeficientes de SHAP de un dato concreto:



SHAP - https://github.com/slundberg/shap

Librería SHAP

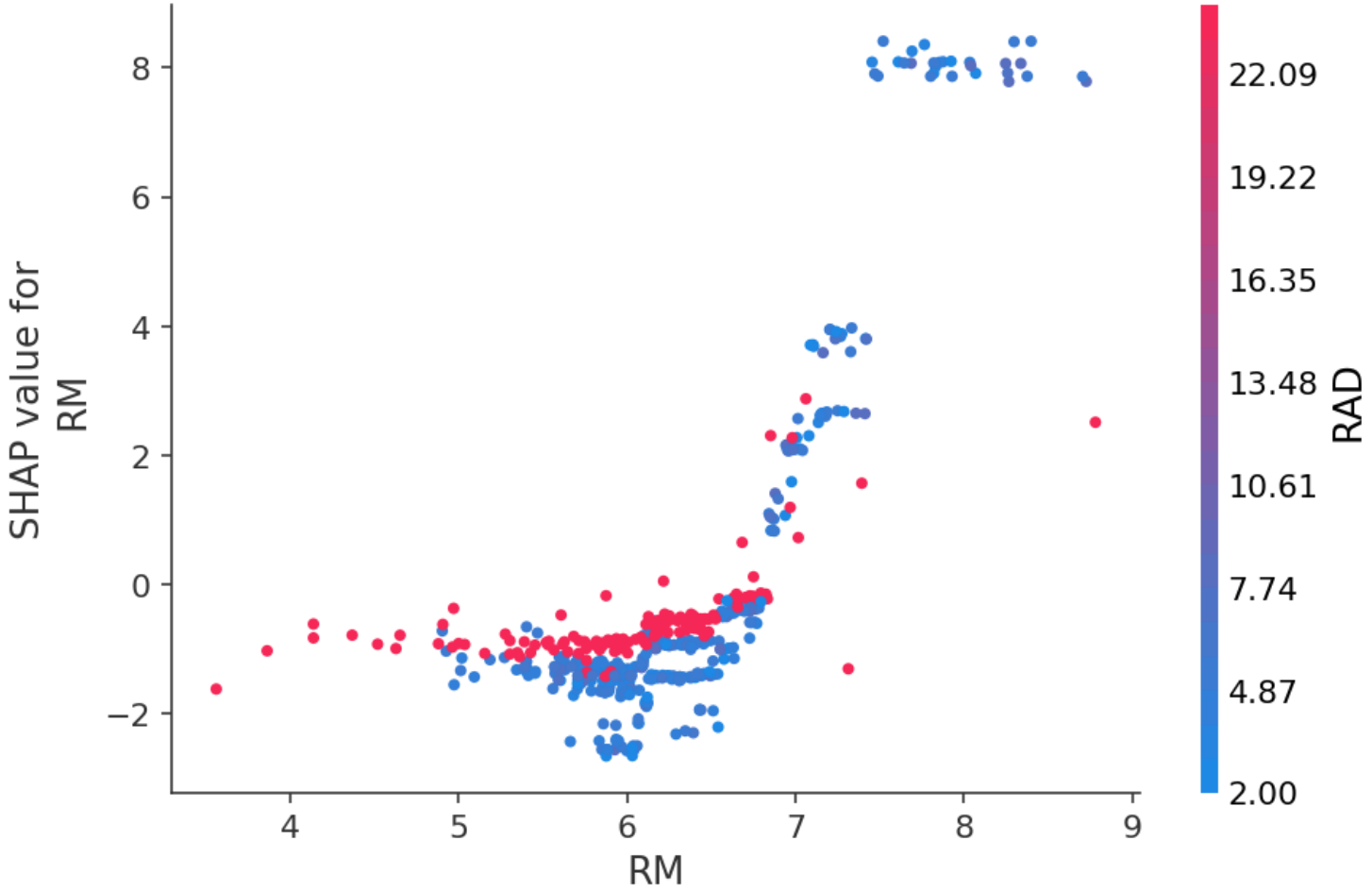
Gracias a la propiedad de aditividad de los valores de Shapley, podemos obtener una explicación global de qué variables influyen más sobre un grupo de patrones o sobre todo el dataset



SHAP - https://github.com/slundberg/shap

Librería SHAP

También es posible obtener gráficas de dependencias parciales con mayor riqueza de información



SHAP - https://github.com/slundberg/shap

Afi Escuela