Interpretabilidad del Deep Learning

Explicabilidad genérica

Christian Oliva Moya

Contenido del curso

- Primera semana:
  - Introducción de conceptos
  - **Explicabilidad genérica**
    - **SHAP**
    - **LIME**
  - Coeficiente de Explicabilidad-Rendimiento

Repaso (I)

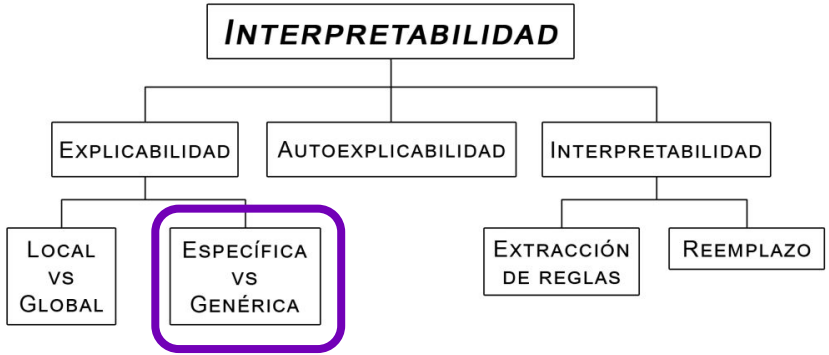
- **Explicabilidad:** Identificación de los atributos del conjunto de datos que tienen una mayor influencia en las predicciones de un modelo
- Responde a la pregunta:

  ¿Por qué este dato de entrada genera esta respuesta en el modelo?

- Los algoritmos de explicabilidad definen una **relevancia** Rx_i para cada atributo de entrada

Repaso (II)

- **Explicabilidad específica vs genérica**



- **Genérica**: el algoritmo se puede aplicar a cualquier modelo
- **Específica**: el algoritmo es específico para un modelo particular (por ejemplo, específico para redes neuronales)

Algoritmos de explicabilidad genéricos

- **Importancia por permutación** ←Todos creemos que este es un algoritmo de selección de atributos cuando **no es así**
- **Relevancia por Oclusión 1x1**
- **SHapley Additive exPlanations** (SHAP)
- **Local Interpretable Model-agnostic Explanations** (LIME)

Todos estos algoritmos de explicabilidad son genéricos, es decir, se pueden utilizar para cualquier modelo de ML y DL

Importancia por Permutación

Importancia por permutación (I)

- Permutamos (barajamos) uno a uno los atributos del dataset y evaluamos el modelo M para ver cómo cambia el rendimiento

$$R_i = M(X) - \frac{1}{N} \sum_{j=1}^{N} M(X'_i)$$

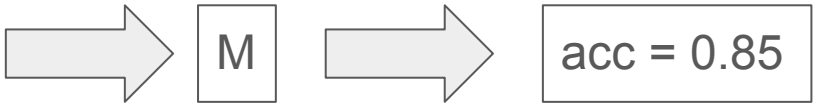
  - M(X) es el rendimiento (accuracy, R2, etc) del modelo M sobre el dataset X
  - X'$_i$ es el dataset resultante después de permutar el atributo i

Importancia por permutación (II)

$$R_i = M(X) - \frac{1}{N}\sum_{j=1}^{N} M(X_i')$$

- 1. Calculamos el score base (accuracy):

Attr1	Attr2	Attr3
11	12	13
21	22	23
31	32	33
41	42	43
51	52	53



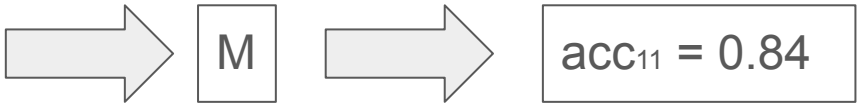
Importancia por permutación (III)

$$R_i = M(X) - \frac{1}{N}\sum_{j=1}^{N} M(X_i')$$

- Para cada atributo i, vamos permutando N=2 veces (i=1, n=1):

Attr1	Attr2	Attr3
21	12	13
11	22	23
51	32	33
31	42	43
41	52	53

acc = 0.85



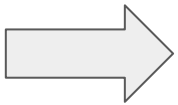
Importancia por permutación (IV)

$$R_i = M(X) - \frac{1}{N}\sum_{j=1}^{N} M(X_i')$$

- Para cada atributo i, vamos permutando N=2 veces (i=1, n=2):

Attr1	Attr2	Attr3
11	12	13
41	22	23
21	32	33
51	42	43
31	52	53

acc = 0.85



acc_{11} = 0.84

acc_{12} = 0.83

Importancia por permutación (V)

$$R_i = M(X) - \frac{1}{N}\sum_{j=1}^{N} M(X_i')$$

- Aplicamos la fórmula para el atributo Attr1:

Attr1	Attr2	Attr3
11	12	13
21	22	23
31	32	33
41	42	43
51	52	53

acc = 0.85

$R_1 = 0.015$

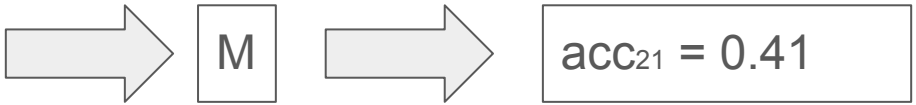
Importancia por permutación (VI)

$$R_i = M(X) - \frac{1}{N}\sum_{j=1}^{N} M(X_i')$$

- Para cada atributo i, vamos permutando N=2 veces (i=2, n=1):

Attr1	Attr2	Attr3
11	32	13
21	42	23
31	12	33
41	52	43
51	22	53

acc = 0.85



$R_1 = 0.015$

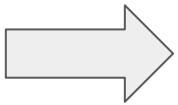
Importancia por permutación (VII)

$$R_i = M(X) - \frac{1}{N}\sum_{j=1}^{N} M(X_i')$$

- Para cada atributo i, vamos permutando N=2 veces (i=2, n=2):

Attr1	Attr2	Attr3
11	22	13
21	52	23
31	42	33
41	12	43
51	32	53

acc = 0.85



acc_{21} = 0.41

acc_{22} = 0.39

R_1 = 0.015

Importancia por permutación (VIII)

$$R_i = M(X) - \frac{1}{N}\sum_{j=1}^{N} M(X_i')$$

- Aplicamos la fórmula para el atributo Attr2:

Attr1	Attr2	Attr3
11	12	13
21	22	23
31	32	33
41	42	43
51	52	53

acc = 0.85

$R_1 = 0.015$

$R_2 = 0.45$

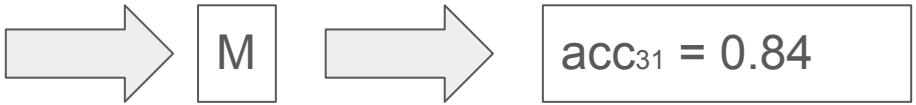
Importancia por permutación (IX)

$$R_i = M(X) - \frac{1}{N}\sum_{j=1}^{N} M(X_i')$$

- Para cada atributo i, vamos permutando N=2 veces (i=3, n=1):

Attr1	Attr2	Attr3
11	12	53
21	22	33
31	32	23
41	42	43
51	52	13

acc = 0.85



$R_1 = 0.015$

$R_2 = 0.45$

Importancia por permutación (X)

$$R_i = M(X) - \frac{1}{N}\sum_{j=1}^{N} M(X_i')$$

- Para cada atributo i, vamos permutando N=2 veces (i=3, n=2):

Attr1	Attr2	Attr3
11	12	53
21	22	33
31	32	13
41	42	43
51	52	23

acc = 0.85

acc_{31} = 0.84

M

acc_{32} = 0.85

R_1 = 0.015

R_2 = 0.45

Importancia por permutación (XI)

$$R_i = M(X) - \frac{1}{N}\sum_{j=1}^{N} M(X_i')$$

- Aplicamos la fórmula para el atributo Attr3:

Attr1	Attr2	Attr3
11	12	13
21	22	23
31	32	33
41	42	43
51	52	53

acc = 0.85

$R_1 = 0.015$

$R_2 = 0.45$

$R_3 = 0.005$

Importancia por permutación (XII)

$$R_i = M(X) - \frac{1}{N}\sum_{j=1}^{N} M(X_i')$$

- Ya tenemos todas las relevancias:

$R_1 = 0.015$	$R_2 = 0.45$	$R_3 = 0.005$

- **El atributo más relevante es Attr2** porque cuando se perturba afecta mucho al rendimiento del modelo.
- Los atributos Attr1 y Attr3 prácticamente **no tienen impacto en el modelo**

Importancia por permutación (XIII)

$$R_i = M(X) - \frac{1}{N}\sum_{j=1}^{N} M(X_i')$$

- La importancia por permutación es un algoritmo de explicabilidad…
  - **Global**, porque se aplica al dataset completo
  - **Genérico**, porque se puede aplicar a cualquier modelo
  - **Sencillo**, porque es bastante intuitiva la idea que hay detrás
  - **Costoso**, porque necesitamos ejecutar N veces por cada atributo

Importancia por permutación (XIV)

- Vamos a implementarlo a mano. Vamos al notebook!
    - Notebook *2.1 Explicabilidad Genérica*

Relevancia por Oclusión

Relevancia por oclusión (I)

- Es una técnica de explicabilidad local utilizada en procesamiento de imágenes
- Se puede utilizar como técnica de explicabilidad tanto global como local para procesamiento de datos tabulares
- Consiste en **anular** una región del espacio de atributos de entrada

Relevancia por oclusión (II)

- Consiste en **anular** una región del espacio de atributos de entrada
- Relevancia por oclusión global para datos tabulares:

$$R_i = M(X) - M(X_i')$$

- Relevancia por oclusión local para datos tabulares:

$$R_i(x) = f(x) - f(x_i')$$

Es una idea parecida a la importancia por permutación, donde en vez de permutar, se anulan atributos

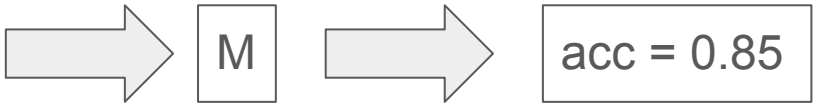
$$R_i = M(X) - M(X_i')$$

$$R_i(x) = f(x) - f(x_i')$$

Relevancia por oclusión (III)

- Vamos primero con la explicabilidad **global**:

Attr1	Attr2	Attr3
11	12	13
21	22	23
31	32	33
41	42	43
51	52	53



$$R_i = M(X) - M(X_i')$$

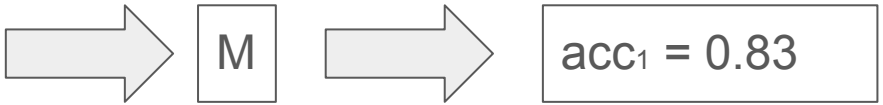
$$R_i(x) = f(x) - f(x_i')$$

Relevancia por oclusión (IV)

- Oclusión del atributo Attr1:

Attr1	Attr2	Attr3
0	12	13
0	22	23
0	32	33
0	42	43
0	52	53

acc = 0.85



Relevancia por oclusión (V)

$$R_i = M(X) - M(X_i')$$

$$R_i(x) = f(x) - f(x_i')$$

- Aplicamos la fórmula al atributo Attr1:

Attr1	Attr2	Attr3
0	12	13
0	22	23
0	32	33
0	42	43
0	52	53

acc = 0.85

M

$acc_1 = 0.83$

$R_1 = 0.02$

$$R_i = M(X) - M(X_i')$$

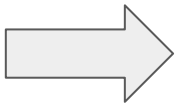
$$R_i(x) = f(x) - f(x_i')$$

Relevancia por oclusión (VI)

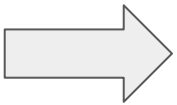
- Oclusión del atributo Attr2:

Attr1	Attr2	Attr3
11	0	13
21	0	23
31	0	33
41	0	43
51	0	53

acc = 0.85



M



$acc_2 = 0.48$

$R_1 = 0.02$

Relevancia por oclusión (VII)

$$R_i = M(X) - M(X_i')$$

$$R_i(x) = f(x) - f(x_i')$$

- Aplicamos la fórmula al atributo Attr2:

Attr1	Attr2	Attr3
11	0	13
21	0	23
31	0	33
41	0	43
51	0	53

acc = 0.85

→ M →

$acc_2 = 0.48$

$R_1 = 0.02$

$R_2 = 0.37$

$$R_i = M(X) - M(X_i')$$

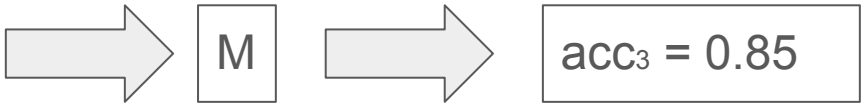
$$R_i(x) = f(x) - f(x_i')$$

Relevancia por oclusión (VIII)

- Oclusión del atributo Attr3:

Attr1	Attr2	Attr3
11	12	0
21	22	0
31	32	0
41	42	0
51	52	0

acc = 0.85



$R_1 = 0.02$

$R_2 = 0.37$

Relevancia por oclusión (IX)

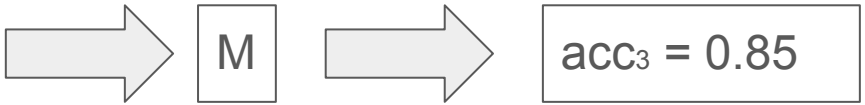
$$R_i = M(X) - M(X_i')$$

$$R_i(x) = f(x) - f(x_i')$$

- Aplicamos la fórmula al atributo Attr3:

Attr1	Attr2	Attr3
11	12	0
21	22	0
31	32	0
41	42	0
51	52	0

acc = 0.85



$R_1 = 0.02$ $R_2 = 0.37$ $R_3 = 0.0$

Relevancia por oclusión (X)

- Ya tenemos todas las relevancias:

$R_1 = 0.02$	$R_2 = 0.37$	$R_3 = 0.0$

- **El atributo más relevante es Attr2** porque cuando se anula afecta mucho al rendimiento del modelo.
- Los atributos Attr1 y Attr3 prácticamente **no tienen impacto en el modelo**

Da un resultado similar a la relevancia por permutación

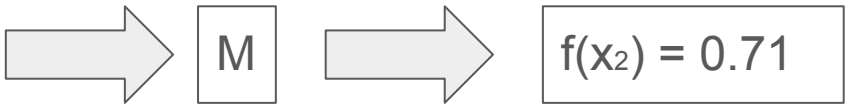
$$R_i = M(X) - M(X_i')$$

$$R_i(x) = f(x) - f(x_i')$$

Relevancia por oclusión (XI)

- Vamos ahora con la explicabilidad **local** para la instancia x_2:

Attr1	Attr2	Attr3
11	12	13
21	22	23
31	32	33
41	42	43
51	52	53



$$R_i = M(X) - M(X_i')$$

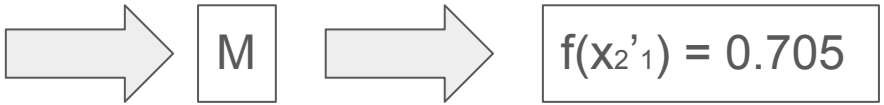
$$R_i(x) = f(x) - f(x_i')$$

Relevancia por oclusión (XII)

- Anulamos el atributo Attr1 en la instancia x_2:

Attr1	Attr2	Attr3
11	12	13
0	22	23
31	32	33
41	42	43
51	52	53

$f(x_2) = 0.71$



$$R_i = M(X) - M(X_i')$$

Relevancia por oclusión (XIII)

$$R_i(x) = f(x) - f(x_i')$$

- Aplicamos la fórmula:

Attr1	Attr2	Attr3
11	12	13
0	22	23
31	32	33
41	42	43
51	52	53

$f(x_2) = 0.71$

→ M →

$f(x_2'_1) = 0.705$

$R_1 = 0.005$

$$R_i = M(X) - M(X_i')$$

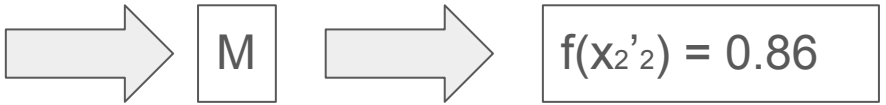
$$R_i(x) = f(x) - f(x_i')$$

Relevancia por oclusión (XIV)

- Anulamos el atributo Attr2 en la instancia x_2:

Attr1	Attr2	Attr3
11	12	13
21	0	23
31	32	33
41	42	43
51	52	53

$f(x_2) = 0.71$



$R_1 = 0.005$

$$R_i = M(X) - M(X_i')$$

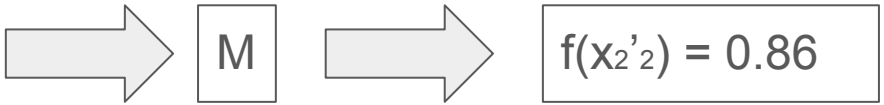
Relevancia por oclusión (XV)

$$R_i(x) = f(x) - f(x_i')$$

- Aplicamos la fórmula:

Attr1	Attr2	Attr3
11	12	13
21	0	23
31	32	33
41	42	43
51	52	53

$f(x_2) = 0.71$



$R_1 = 0.005$

$R_2 = -0.15$

$$R_i = M(X) - M(X_i')$$

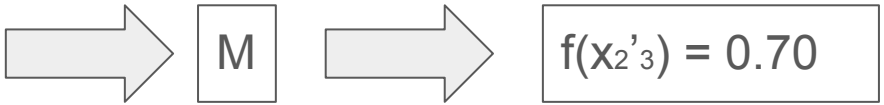
$$R_i(x) = f(x) - f(x_i')$$

Relevancia por oclusión (XVI)

- Anulamos el atributo Attr3 en la instancia x_2:

Attr1	Attr2	Attr3
11	12	13
21	22	0
31	32	33
41	42	43
51	52	53

$f(x_2) = 0.71$



$R_1 = 0.005$

$R_2 = -0.15$

$$R_i = M(X) - M(X_i')$$

Relevancia por oclusión (XVII)

$$R_i(x) = f(x) - f(x_i')$$

- Aplicamos la fórmula:

Attr1	Attr2	Attr3
11	12	13
21	22	0
31	32	33
41	42	43
51	52	53

$f(x_2) = 0.71$

→ M →

$f(x_2'_3) = 0.70$

$R_1 = 0.005$

$R_2 = -0.15$

$R_3 = 0.01$

Relevancia por oclusión (XVIII)

$$R_i(x) = f(x) - f(x'_i)$$

- Ya tenemos todas las relevancias para la instancia x_2:

$R_1 = 0.005$	$R_2 = -0.15$	$R_3 = 0.01$

- ¿Qué significa relevancia negativa o positiva?

Relevancia por oclusión (XIX)

$$R_i(x) = f(x) - f(x'_i)$$

- Ya tenemos todas las relevancias para la instancia x_2:

$R_1 = 0.005$	$R_2 = -0.15$	$R_3 = 0.01$

- ¿Qué significa relevancia negativa o positiva?
  - Relevancia **positiva**: Anular hace que la predicción baje
  - Relevancia **negativa**: Anular hace que la predicción suba

Relevancia por oclusión (XX)

- La relevancia por oclusión es un algoritmo de explicabilidad…
  - **Híbrido**, porque puede ser tanto global como local
  - **Genérico**, porque se puede aplicar a cualquier modelo
  - **Sencillo**, porque es bastante intuitiva la idea que hay detrás
  - **Ligero**, porque solamente itera una vez por atributo

Relevancia por oclusión (XXI)

- Vamos a implementarlo a mano. Vamos al notebook!
    - Notebook *2.1 Explicabilidad Genérica*

SHAP

SHAP (I)

- El valor de Shapley en teoría de juegos para una característica i es:

$$\phi_i(x) = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [f_{S \cup \{i\}}(x) - f_S(x)]$$

Esto no es viable, escala exponencialmente

- El algoritmo SHapley Additive exPlanations (SHAP) se basa en esta idea

  pero muestreando permutaciones aleatorias de atributos

SHAP (II)

- ¿En qué consiste la idea?
  - **Introducir de forma aleatoria** atributos uno a uno (imputando el resto)
  - **Evaluar cómo afecta** el haber introducido dicho atributo al resto
- Podemos basarnos en esta idea:

$$contribución_i = f_{S \cup \{i\}}(x) - f_S(x)$$

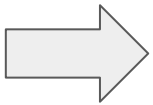
Esta ecuación sencilla nos gusta más

$$contribución_i = f_{S\cup\{i\}}(x) - f_S(x)$$

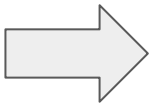
SHAP (III)

- Vamos a probarlo con la instancia x_2: Primero calculamos un valor base: E[f(X)]

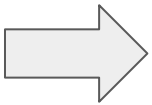
Attr1	Attr2	Attr3
11	12	13
21	22	23
31	32	33
41	42	43
51	52	53



M



Pred
0.81
0.71
0.24
0.93
0.37



base = 0.61

$$contribución_i = f_{S\cup\{i\}}(x) - f_S(x)$$

SHAP (IV)

- Luego tenemos que ejecutar N=2 veces seleccionando aleatoriamente el orden de entrada de los atributos: $PERM_1$ = [2, 1, 3]

Attr1	Attr2	Attr3
11	12	13
21	22	23
31	32	33
41	42	43
51	52	53

base = 0.61

$$contribución_i = f_{S\cup\{i\}}(x) - f_S(x)$$

SHAP (V)

- Imputamos todos los atributos y vamos en orden: $PERM_1$ = [2, 1, 3]

Attr1	Attr2	Attr3
11	12	13
0	0	0
31	32	33
41	42	43
51	52	53

base = 0.61

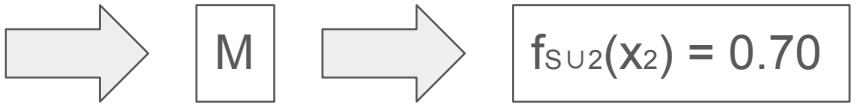
$$contribución_i = f_{S \cup \{i\}}(x) - f_S(x)$$

SHAP (VI)

- Imputamos todos los atributos y vamos en orden: $PERM_1 = [2, 1, 3]$

Attr1	Attr2	Attr3
11	12	13
0	22	0
31	32	33
41	42	43
51	52	53

base = 0.61



$$contribución_i = f_{S\cup\{i\}}(x) - f_S(x)$$

SHAP (VII)

- Imputamos todos los atributos y vamos en orden: $PERM_1 = [2, 1, 3]$

Attr1	Attr2	Attr3
11	12	13
0	22	0
31	32	33
41	42	43
51	52	53

base = 0.61

M

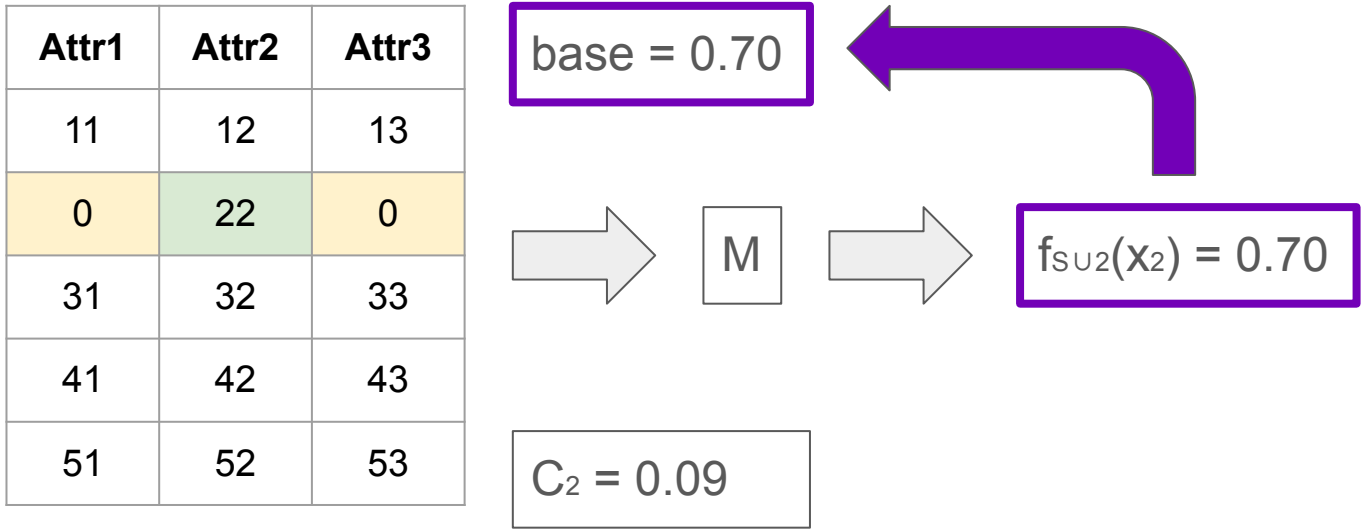
$f_{S\cup 2}(x_2) = 0.70$

$C_2 = 0.09$

$$contribución_i = f_{S \cup \{i\}}(x) - f_S(x)$$

SHAP (VIII)

- Imputamos todos los atributos y vamos en orden: $PERM_1 = [2, 1, 3]$



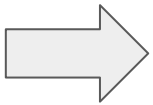
$$contribución_i = f_{S \cup \{i\}}(x) - f_S(x)$$

SHAP (IX)

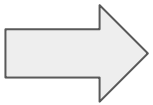
- Imputamos todos los atributos y vamos en orden: $PERM_1 = [2, 1, 3]$

Attr1	Attr2	Attr3
11	12	13
11	22	0
31	32	33
41	42	43
51	52	53

base = 0.70



M



$f_{S \cup 1}(x_2) = 0.71$

$C_2 = 0.09$

$$contribución_i = f_{S\cup\{i\}}(x) - f_S(x)$$

SHAP (X)

- Imputamos todos los atributos y vamos en orden: $PERM_1 = [2, 1, 3]$

Attr1	Attr2	Attr3
11	12	13
11	22	0
31	32	33
41	42	43
51	52	53

base = 0.70

M

$f_{S\cup 1}(x_2) = 0.71$

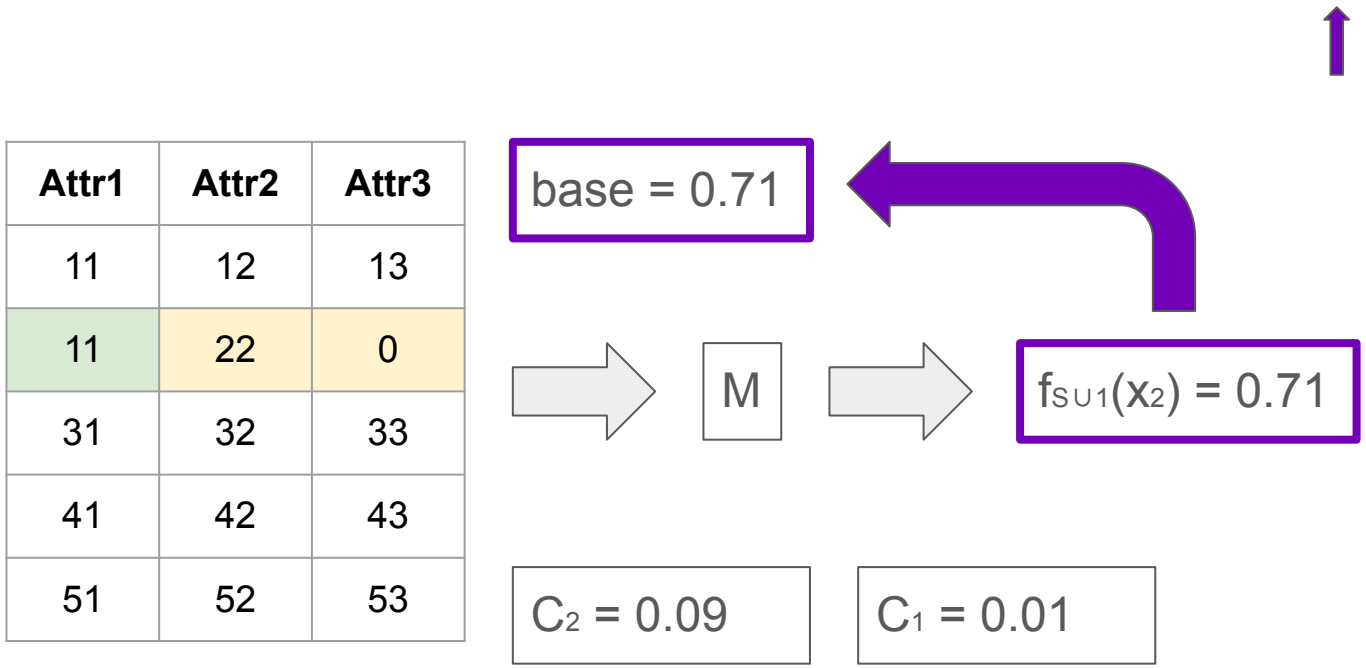
$C_2 = 0.09$

$C_1 = 0.01$

$$contribución_i = f_{S \cup \{i\}}(x) - f_S(x)$$

SHAP (XI)

- Imputamos todos los atributos y vamos en orden: $PERM_1 = [2, 1, 3]$



$$contribución_i = f_{S \cup \{i\}}(x) - f_S(x)$$

SHAP (XII)

- Imputamos todos los atributos y vamos en orden: $PERM_1 = [2, 1, 3]$

Attr1	Attr2	Attr3
11	12	13
11	22	23
31	32	33
41	42	43
51	52	53

base = 0.71

M

$f_{S \cup 3}(x_2) = 0.71$

$C_2 = 0.09$

$C_1 = 0.01$

$$contribución_i = f_{S\cup\{i\}}(x) - f_S(x)$$

SHAP (XIII)

- Imputamos todos los atributos y vamos en orden: $PERM_1 = [2, 1, 3]$

Attr1	Attr2	Attr3
11	12	13
11	22	23
31	32	33
41	42	43
51	52	53

base = 0.71

M

$f_{S\cup 3}(x_2) = 0.71$

$C_2 = 0.09$

$C_1 = 0.01$

$C_3 = 0.0$

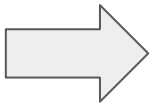
$$contribución_i = f_{S \cup \{i\}}(x) - f_S(x)$$

SHAP (XIV)

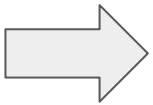
- Imputamos todos los atributos y vamos en orden: $PERM_1 = [2, 1, 3]$

Attr1	Attr2	Attr3
11	12	13
11	22	23
31	32	33
41	42	43
51	52	53

base = 0.71



M



$f_{S \cup 3}(x_2) = 0.71$

$C_2 = 0.09$

$C_1 = 0.01$

$C_3 = 0.0$

$$contribución_i = f_{S\cup\{i\}}(x) - f_S(x)$$

SHAP (XV)

- Repetimos una vez más para llegar a N=2 veces: $PERM_2$ = [3, 1, 2]

Attr1	Attr2	Attr3
11	12	13
21	22	23
31	32	33
41	42	43
51	52	53

base = 0.61

Volvemos al valor base inicial

C_2 = 0.09

C_1 = 0.01

C_3 = 0.0

$$contribución_i = f_{S\cup\{i\}}(x) - f_S(x)$$

SHAP (XVI)

- Imputamos todos los atributos y vamos en orden: $PERM_2$ = [3, 1, 2]

Attr1	Attr2	Attr3
11	12	13
0	0	0
31	32	33
41	42	43
51	52	53

base = 0.61

C_2 = 0.09

C_1 = 0.01

C_3 = 0.0

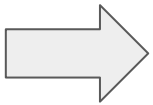
$$contribución_i = f_{S \cup \{i\}}(x) - f_S(x)$$

SHAP (XVII)

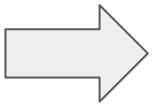
- Imputamos todos los atributos y vamos en orden: $PERM_2 = [3, 1, 2]$

Attr1	Attr2	Attr3
11	12	13
0	0	23
31	32	33
41	42	43
51	52	53

base = 0.61



M



$f_{S \cup 3}(x_2) = 0.61$

$C_2 = 0.09$

$C_1 = 0.01$

$C_3 = 0.0$

$$contribución_i = f_{S\cup\{i\}}(x) - f_S(x)$$

SHAP (XVIII)

- Imputamos todos los atributos y vamos en orden: $PERM_2 = [3, 1, 2]$

Attr1	Attr2	Attr3
11	12	13
0	0	23
31	32	33
41	42	43
51	52	53

base = 0.61

M

$f_{S\cup 3}(x_2) = 0.61$

$C_2 = 0.09$

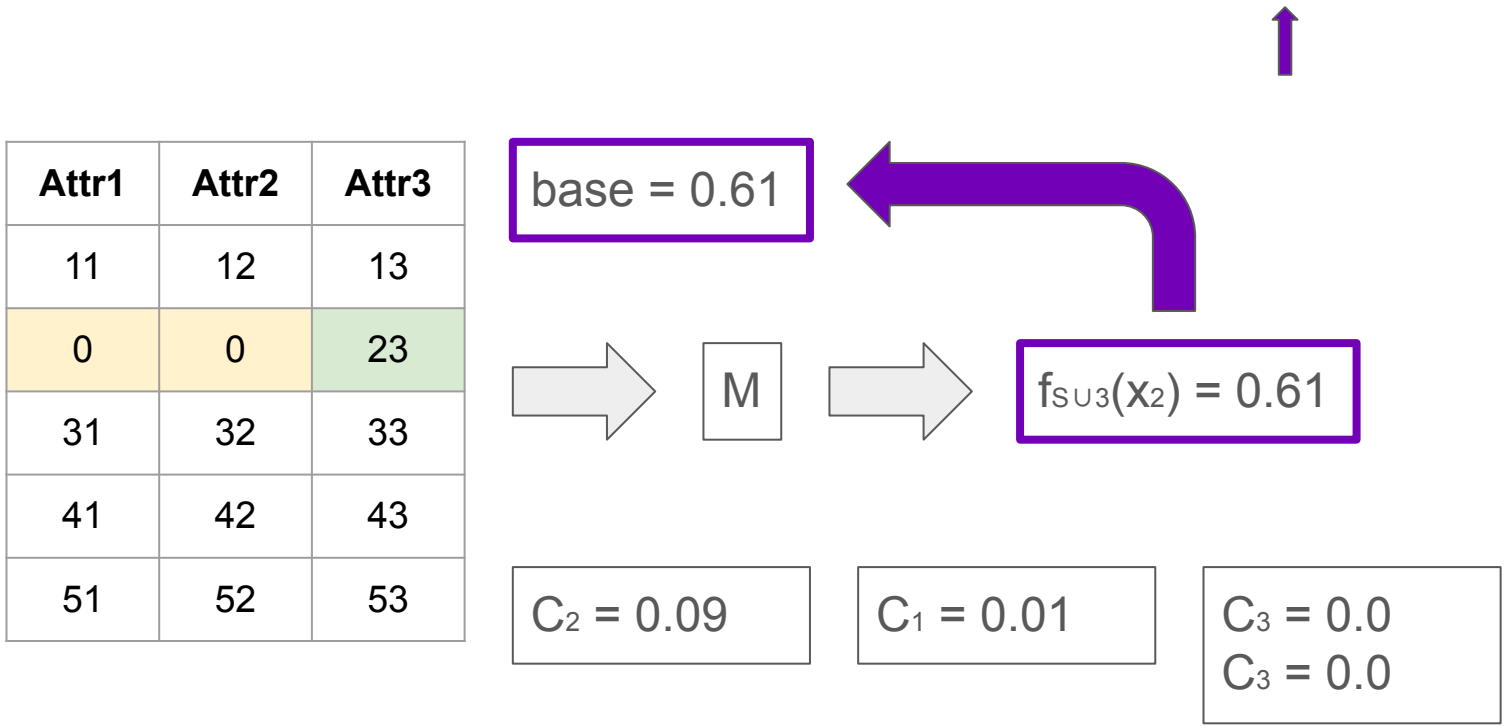
$C_1 = 0.01$

$C_3 = 0.0$
$C_3 = 0.0$

$$contribución_i = f_{S\cup\{i\}}(x) - f_S(x)$$

SHAP (XIX)

- Imputamos todos los atributos y vamos en orden: $PERM_2 = [3, 1, 2]$



$$contribución_i = f_{S\cup\{i\}}(x) - f_S(x)$$

SHAP (XX)

- Imputamos todos los atributos y vamos en orden: $PERM_2$ = [3, 1, 2]

Attr1	Attr2	Attr3
11	12	13
21	0	23
31	32	33
41	42	43
51	52	53

base = 0.61

M

$f_{S\cup1}(x_2) = 0.63$

$C_2 = 0.09$

$C_1 = 0.01$

$C_3 = 0.0$
$C_3 = 0.0$

$$contribución_i = f_{S\cup\{i\}}(x) - f_S(x)$$

SHAP (XXI)

- Imputamos todos los atributos y vamos en orden: $PERM_2 = [3, 1, 2]$

Attr1	Attr2	Attr3
11	12	13
21	0	23
31	32	33
41	42	43
51	52	53

base = 0.61

M

$f_{S\cup 1}(x_2) = 0.63$

$C_2 = 0.09$

$C_1 = 0.01$
$C_1 = 0.02$

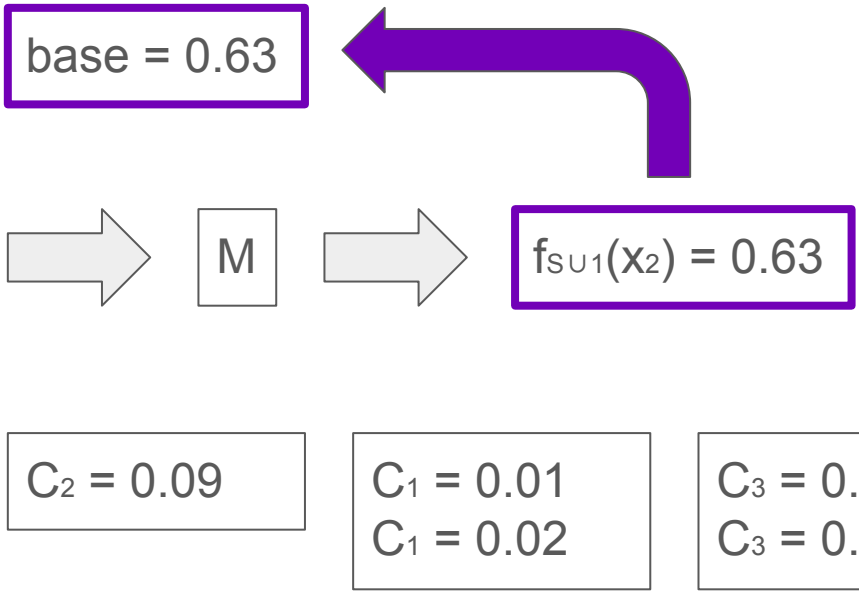
$C_3 = 0.0$
$C_3 = 0.0$

$$contribución_i = f_{S \cup \{i\}}(x) - f_S(x)$$

SHAP (XXII)

- Imputamos todos los atributos y vamos en orden: $PERM_2 = [3, 1, 2]$

Attr1	Attr2	Attr3
11	12	13
21	0	23
31	32	33
41	42	43
51	52	53



$$contribución_i = f_{S\cup\{i\}}(x) - f_S(x)$$

SHAP (XXIII)

- Imputamos todos los atributos y vamos en orden: $PERM_2$ = [3, 1, 2]

Attr1	Attr2	Attr3
11	12	13
21	22	23
31	32	33
41	42	43
51	52	53

base = 0.63

M

$f_{S\cup 2}(x_2) = 0.71$

$C_2 = 0.09$

$C_1 = 0.01$
$C_1 = 0.02$

$C_3 = 0.0$
$C_3 = 0.0$

$$contribución_i = f_{S\cup\{i\}}(x) - f_S(x)$$

SHAP (XXIV)

- Imputamos todos los atributos y vamos en orden: $PERM_2 = [3, 1, 2]$

Attr1	Attr2	Attr3
11	12	13
21	22	23
31	32	33
41	42	43
51	52	53

base = 0.63

M

$f_{S\cup 2}(x_2) = 0.71$

$C_2 = 0.09$
$C_2 = 0.08$

$C_1 = 0.01$
$C_1 = 0.02$

$C_3 = 0.0$
$C_3 = 0.0$

SHAP (XXV)

- Ya tenemos todos los valores de SHAP mediante permutaciones:

$C_1 = 0.01$ $C_1 = 0.02$	$C_2 = 0.09$ $C_2 = 0.08$	$C_3 = 0.0$ $C_3 = 0.0$

- Promediamos:

$C_1 = 0.015$	$C_2 = 0.085$	$C_3 = 0.0$

- El atributo Attr2 vuelve a ser el más relevante

SHAP (XXV)

- Ya t ones:

OJO! SHAP también puede dar relevancias negativas y positivas, que se interpretan igual que la relevancia por oclusión

- Promediamos:

$C_1 = 0.015$	$C_2 = 0.085$	$C_3 = 0.0$

- El atributo Attr2 vuelve a ser el más relevante

SHAP (XXVI)

- SHAP es un algoritmo de explicabilidad…
  - **Local**, porque se aplica a instancias individuales del dataset
  - **Genérico**, porque se puede aplicar a cualquier modelo
  - **Algo complicado**, porque se basa en una idea compleja
  - **Costoso**, porque necesitamos ejecutar N veces por cada atributo

SHAP (XXVII)

- Vamos a implementarlo a mano. Vamos al notebook!
    - Notebook *2.1 Explicabilidad Genérica*

LIME

LIME (I)

- Es un algoritmo local que consiste en:

  Aproximar el comportamiento del modelo complejo con puntos cercanos a la instancia que se quiere explicar **utilizando un modelo lineal**

- Es un algoritmo que estudia el **comportamiento cercano de los vecinos**

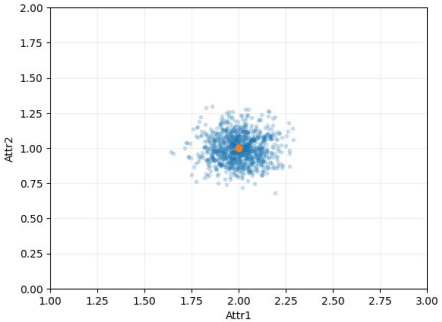
LIME (II)

- La idea consiste en lo siguiente:
  1. Generamos un dataset sintético *X'* alrededor de la instancia *x*
  2. Se calculan las predicciones del modelo complejo sobre *X'*
  3. Se asignan pesos a los puntos de *X'* según su cercanía a *x*
  4. Se entrena un modelo lineal con *X'* de forma ponderada según cercanía
  5. Los coeficientes (pesos) del modelo lineal son la relevancia *Rx*

LIME (III)

1. Generamos un dataset sintético *X'* con D puntos alrededor de la instancia *x*

Attr1	Attr2
11	12
21	22
31	32
41	42
51	52



Attr1	Attr2
21_1	22_1
21_2	22_2
21_3	22_3
…	…
21_D	22_D

LIME (IV)

2. Se calculan las predicciones del modelo complejo sobre *X'*

Attr1	**Attr2**
21_1	22_1
21_2	22_2
21_3	22_3
…	…
21_D	22_D

Pred
P_1
P_2
P_3
…
P_D

LIME (V)

3. Se asignan pesos a los puntos de *X'* según su cercanía a *x*

Attr1	Attr2
21_1	22_1
21_2	22_2
21_3	22_3
…	…
21_D	22_D

Dist

Attr1	Attr2
21	22

Dist
D_1
D_2
D_3
…
D_D

W
$1/D_1$
$1/D_2$
$1/D_3$
…
$1/D_D$

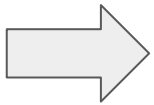
Pred
P_1
P_2
P_3
…
P_D

LIME (VI)

4. Se entrena un modelo lineal con *X'* de forma ponderada según cercanía

Attr1	Attr2	Pred	W
21_1	22_1	P_1	$1/D_1$
21_2	22_2	P_2	$1/D_2$
21_3	22_3	P_3	$1/D_3$
…	…	…	…
21_D	22_D	P_D	$1/D_D$

No olvidemos entrenar con las ponderaciones.
en SKLearn y Keras: *sample_weights*



RegLin = x @ coef + bias

LIME (VII)

5. Los coeficientes (pesos) del modelo lineal son la relevancia *Rx*

RegLin = x @ **coef** + bias

- El modelo lineal aprende cómo cambian las predicciones alrededor de *x*
- LIME, al contrario que SHAP, solamente mira el vecindario cercano a *x*

LIME (VIII)

- LIME es un algoritmo de explicabilidad…
  - **Local**, porque se aplica a instancias individuales del dataset
  - **Genérico**, porque se puede aplicar a cualquier modelo
  - **Algo complicado**, porque se basa en una idea compleja
  - **Algo costoso**, porque necesitamos entrenar un modelo adicional

LIME (IX)

- Vamos a implementarlo a mano. Vamos al notebook!
    - Notebook *2.1 Explicabilidad Genérica*