

Explicabilidad de modelos complejos

15 febrero 2019 - Diego Yus



Índice

Kernel Analytics

- Introducción
- Explicabilidad con gráficos
- xgboostExplainer
- SHAP
- BreakDown
- LIME



Introducción

[[07|09||14|10|10||08|32|97||11|097||08|121||16|0999||16||3
|6B|55|72|6E|65|6C|20|61|6E|61|6C|79|74|69|63|73|0D
0110|0111|01100|01|01100|01|01110|01|01110|01100|01|0110110
00100000|00000|01|01110|01|00001|01|01100|01|0111001|00001|0110110|

¿Qué es la explicabilidad y por qué es importante?

Explicabilidad (explainability or interpretability): the ability to explain or to present [something] in understandable terms to a human^[1].

Un poco de contexto:

- Creciente adopción de IA en todo tipo de industrias (frente al uso previo de modelos simples).
- Diferente velocidad adopción y diferentes necesidades:
 - Industrias basadas en tecnología con decisiones no críticas (recomendación de películas)
 - Industrias establecidas con decisiones críticas (concesión hipoteca, diagnóstico médico, etc.)
- Humanos necesitan explicaciones, especialmente por temas regulatorios.
- Hasta hace algunos años, la mayoría de modelos complejos de ML eran cajas negras.

[1] Towards A Rigorous Science of Interpretable Machine Learning – Finale Doshi-Velez, Been Kim (2017) - <https://arxiv.org/abs/1702.08608>



Explicabilidad del modelos

A medida que se emplean modelos más complejos se hace más difícil entender el sentido en el que las variables impactan en la variable objetivo. Los métodos de explicabilidad vienen a completar modelos de caja negra ofreciendo información sobre cómo las variables impactan tanto en la globalidad como a nivel observación individual.

- **Explicabilidad global vs. explicabilidad local**
 - Explicabilidad global: Explica cómo funciona el modelo cuando tiene que predecir nuevas observaciones
 - Explicabilidad local: Explica las razones detrás de una predicción específica (i.e. para una sola observación). Se puede dividir en:
 - Enfoque basado en aproximación de estructura local (white box)
 - Enfoque basado en descomposición de predicción
- **Precisión vs. Explicabilidad (tradeoff)**
 - Modelos muy precisos con baja explicabilidad (deep learning)
 - Modelos aproximados con alta explicabilidad (regresión lineal)
- **Métodos de interpretabilidad model-agnostic vs. model-specific**

Introducción

Clasificación de métodos explicativos (no exhaustiva)

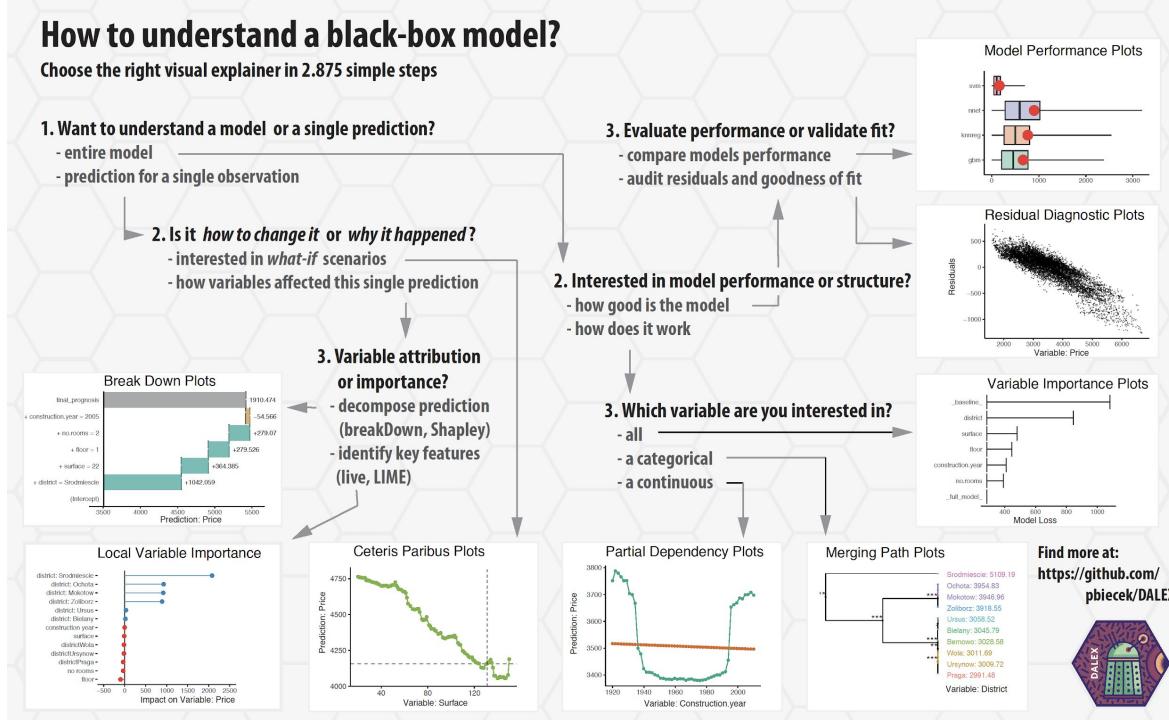


Image Source: <https://github.com/pbiecek/DALEX>

• Explicabilidad global

- Funcionamiento
 - Importancia de variable
 - Plots simples (PDP, ICE, etc.)

• Explicabilidad local

- Escenario "What if...?"
- Escenario "Por qué ha ocurrido?"
 - Descomponer predicción (SHAP)
 - Identificar variables relevantes (LIME)

Explicabilidad con gráficos

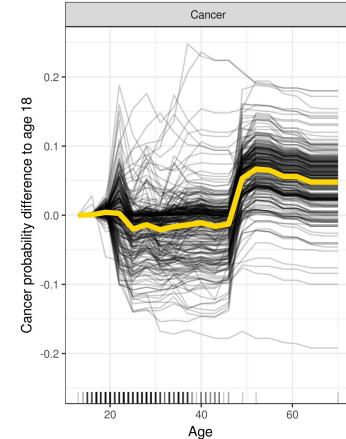
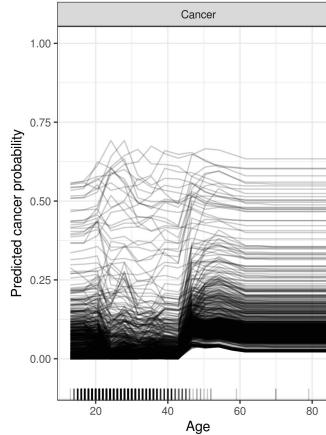
Explicabilidad a través de plots

Proporcionan explicabilidad con respecto a una sola variable (máx. dos) después de haber realizado la predicción.

Individual Conditional Expectations (ICE)



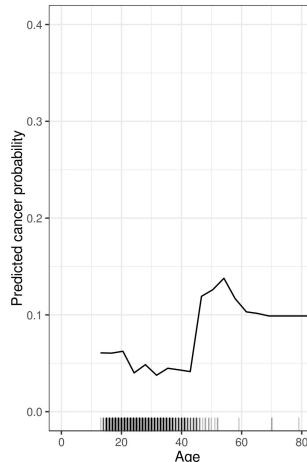
Para un feature, ICE dibuja una línea por observación, representando cómo cambiaría la predicción cuando el feature cambia. Explicabilidad a nivel individual.



Partial Dependence Plots (PDP)



La curva para un determinado valor de un feature representa la media de las predicciones cuando se fuerza a todos los data points a tomar ese valor del feature.



Problemas si variables están correladas y si hay pocos data points con ese valor de feature. Explicabilidad global.

Librería en R: *ceteris paribus*. Además permite realizar otros tipos de plots. Existen otras librerías (Python: sklearn, PyCEbox)

XGBOOST EXPLAINER

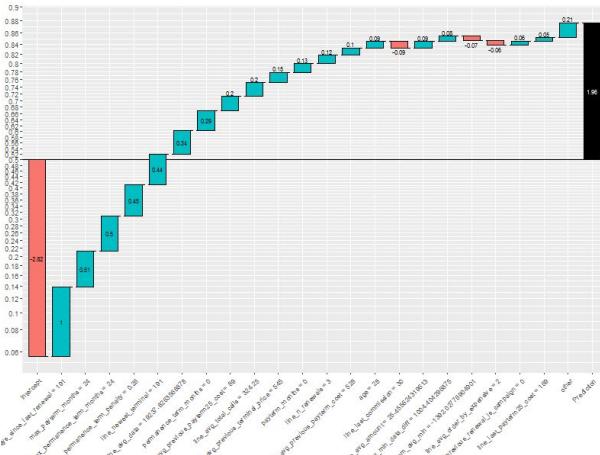
XGBoost explainer

Librería que produce explicaciones de modelo xgboost o light gbm a nivel global y local (model specific).

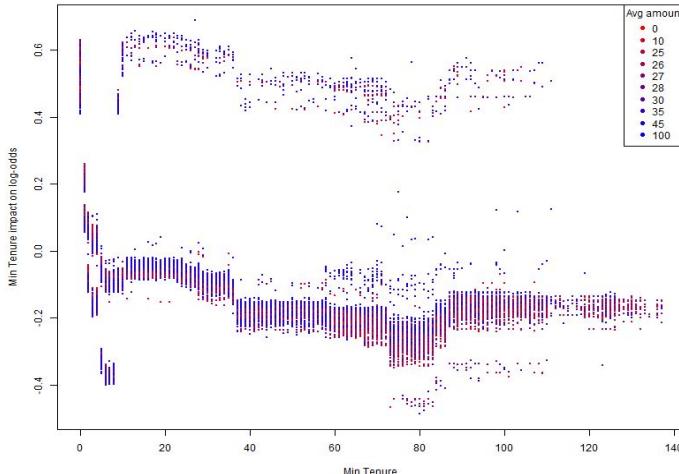
Funciona sumando las contribuciones de cada feature para cada árbol del ensemble, exactamente de la misma manera en que se realiza para un decision tree. Similar a SHAP (Contribuciones marginales de cada feature, gráfico apilado/en cascada).

- ✓ Presente para R y Python.
 - ✓ Fácil de usar y rápido.

Descomposición de atribuciones de probabilidad por variable a nivel individual



Explicaciones de impacto en la predicción de una única variable a nivel global. Aunque no es capaz de identificar interacciones con otras variables



SHAP

|6B|55|72|6E|65|6C|20|61|6E|61|6C|79|74|69|63|73|0D

73%
K.B
41391538
2.159871

10%
K.M
40462952
3693742

Shapley additive explanations

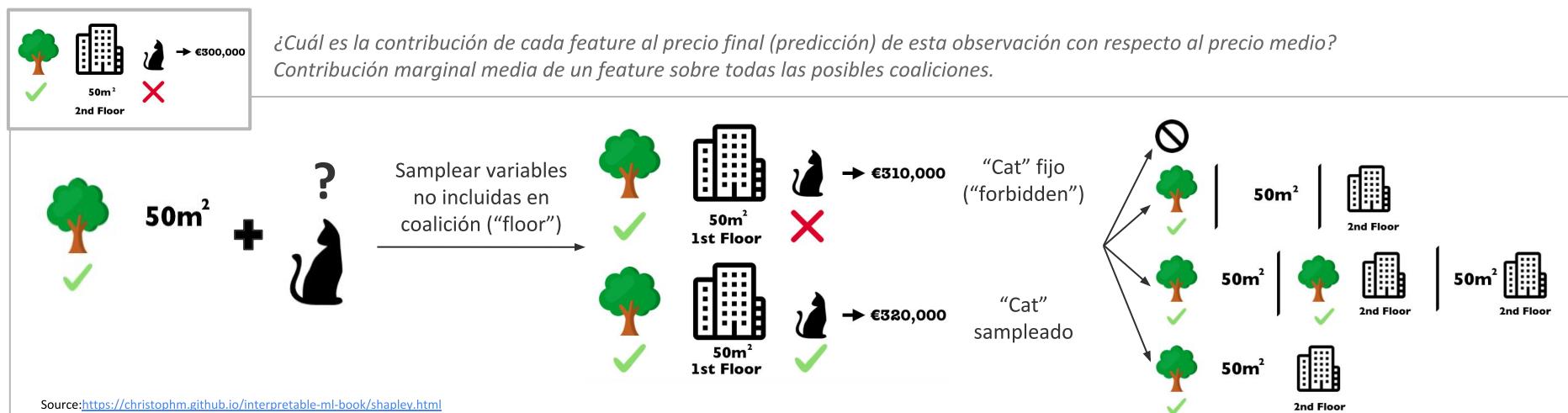
SHapley Additive exPlanations es una metodología para atribuir un valor a la contribución de cada variable para cada observación.

SHAP unifica 6 tipos de métodos para atribuir la importancia de características a las observaciones y tiene 3 características altamente deseables:



Ejemplo: Modelo (black-box) predice precio de la vivienda. Precio medio: 310K €. Predicción observación individual: 300K €. ¿Por qué?

¿Cuál es la contribución de cada feature al precio final (predicción) de esta observación con respecto al precio medio?
Contribución marginal media de un feature sobre todas las posibles coaliciones.



Shapley additive explanations

Ventajas:

1. Descomposición completa de la predicción (a diferencia de LIME). ¿Aspectos legales?
2. Único método explicativo con teoría sólida detrás (consistencia, missingness y accuracy local).
3. Permite explicaciones comparativas (a diferencia de LIME).
4. Model-agnostic

Desventajas:

1. Posible malinterpretación. No expresa cuánto cambia la predicción si eliminamos el feature, sino la contribución total del feature a la diferencia entre predicción real y la media de predicciones.
2. Usa todos los features, no crea sparse explanations. Librería SHAP corrige esto.
3. Mucho tiempo de computación, pero librería SHAP tiene implementación eficiente (exacto para trees, aprox. para resto).
4. Devuelve valor, no un modelo de predicción (imposibilita los “what-if”).
5. Necesita acceso a los datos (modelo no es suficiente).

Librerías/Implementación:

1. **shap** (python): madura, optimizada para distintos modelos, explicabilidad a nivel global y local.
Plot functions integradas.
2. **shapleyR** (R), desarrollo, solo devuelve los valores, no plots.

SHAP (Python). Ejemplo práctico.

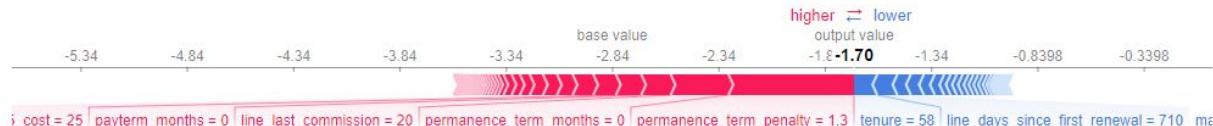
Dataset: Propensión a la compra de terminales en compañía Telco

Modelo: xgboost

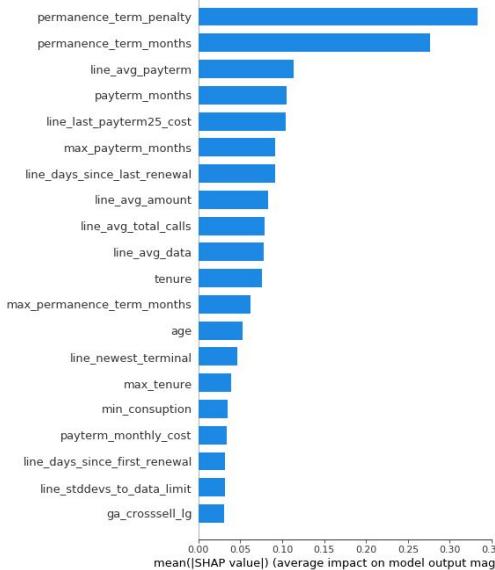
Factores importantes en la toma de la decisión:

- Penalización por permanencia
- Meses de permanencia
- Plazo de pago
- Etc.

Individual explanation



Feature Importance



p	Log odds
0.1	-2.20
0.5	0.0
0.9	2.20

SHAP (Python). Ejemplo práctico.

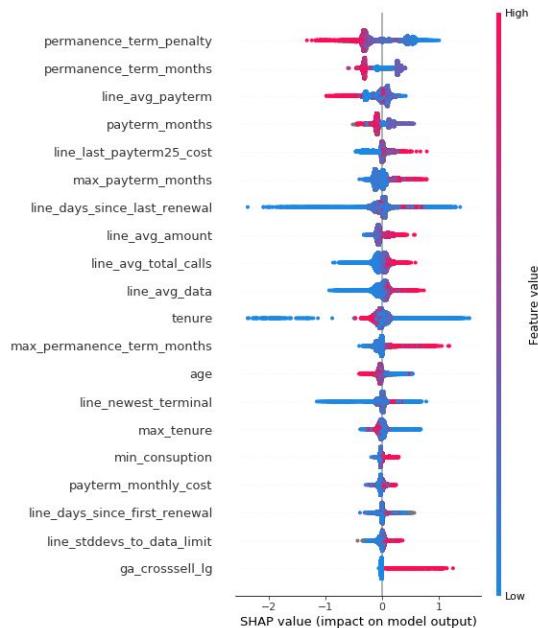
Dataset: Propensión a la compra de terminales en compañía Telco

Modelo: xgboost

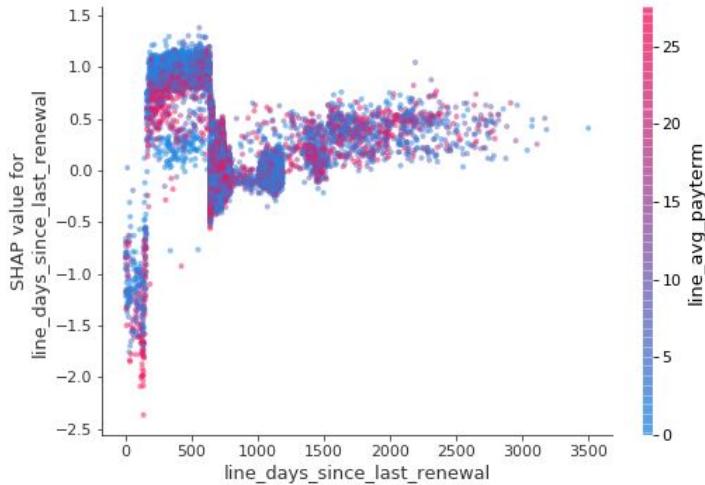
Factores importantes en la toma de la decisión:

- Penalización por permanencia
- Meses de permanencia
- Plazo de pago
- Etc.

All features explanation



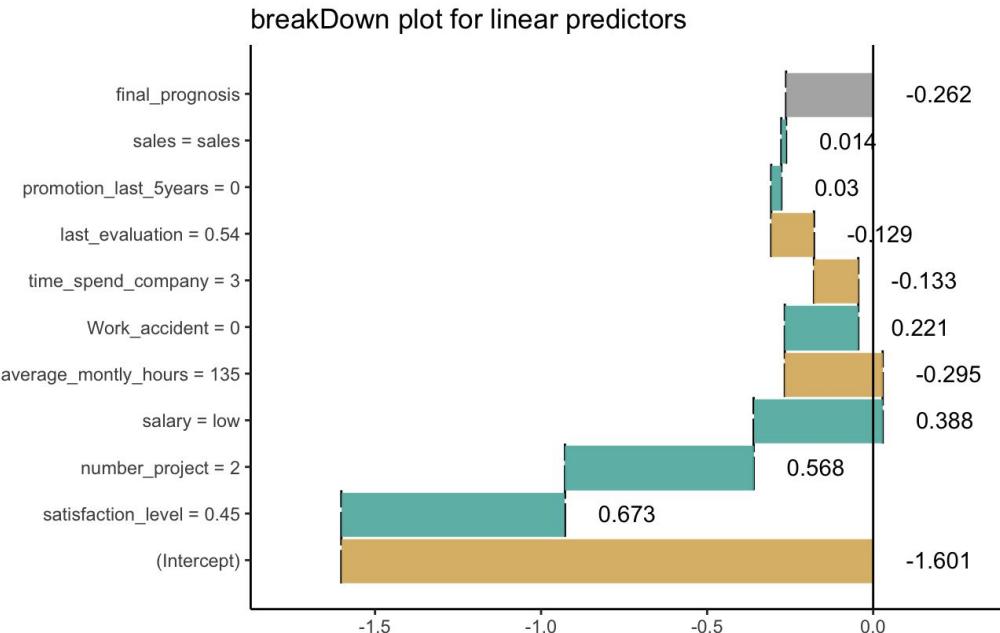
Global explanations for a single feature



BreakDown

BreakDown

- Explainer de modelo lineal (lm) o de modelo lineal generalizado (glm).
- Descomposición de predicción en atribuciones de variables.
- Basado en misma idea que shapley values (atribución por coalición), pero simplificado. Enfoque greedy: no se prueban todas las posibles coaliciones, sino que se van añadiendo variables en orden creciente de contribución.
- Problema: puede no funcionar bien para modelos complejos debido al enfoque greedy.
- Librería de R. Produce waterfall plots.



Source: <https://pbiecek.github.io/breakDown/>

LIME

73%

10%
K.M
40401952
36937

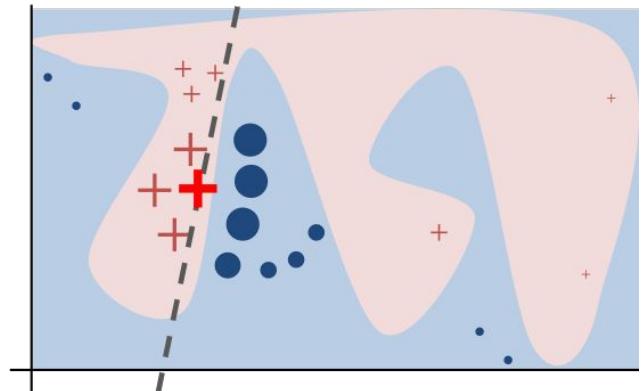


LIME (Local Interpretable Model-agnostic Explanations)

Asume que un modelo complejo es lineal en un **entorno local** (no está probado) y lo aproxima con un modelo simple en ese entorno.

Cómo funciona:

1. Crea perturbaciones del dataset.
2. Computa la respuesta del modelo complejo a esas nuevas samples.
3. Pesa las nuevas samples por su distancia a la observación original.
¿Qué es distancia? Imagen, texto y tabular data.
4. Ajusta un modelo simple a las nuevas samples ponderadas eligiendo el número de features a considerar.
5. Extrae los pesos/coeficientes de los features del modelo simple y los usa como explicación para el comportamiento del modelo complejo en un entorno local de la predicción.



Source: <https://github.com/marcotcr/lime>

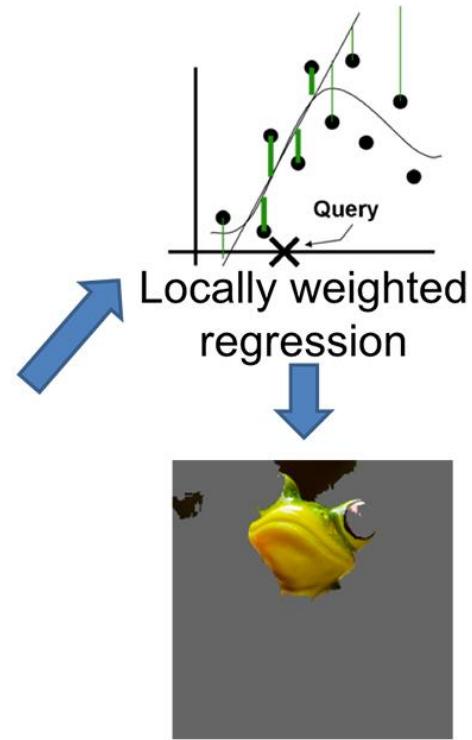
Perturbaciones LIME

Clasificador de imágenes



Original Image
 $P(\text{tree frog}) = 0.54$

Perturbed Instances	$P(\text{tree frog})$
	0.85
	0.00001
	0.52



Explanation

LIME



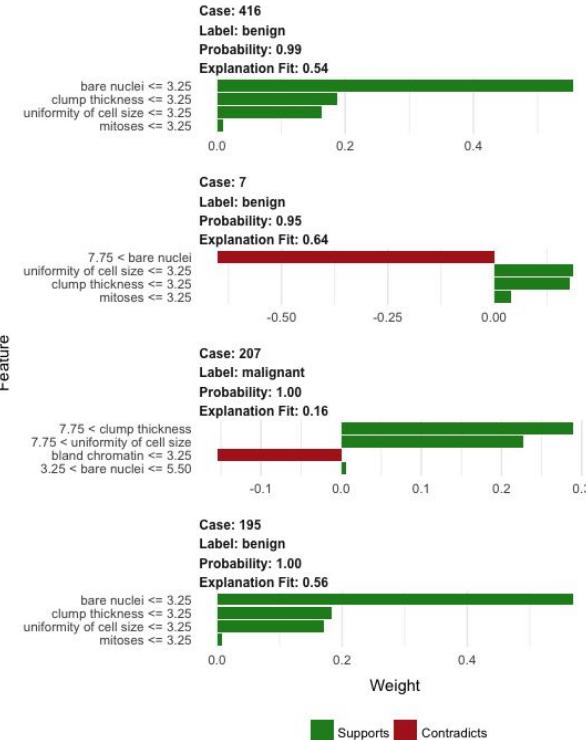
Ventajas:

1. Crea sparse/selective explanations. Número de features explicativos es parámetro tuneable.
2. Válido para tabular data, texto e imagen.
3. Devuelve un modelo de predicción. Permite ver cómo cambiaría la predicción (local) cambiando el input.
4. Model-agnostic.

Desventajas:

1. Produce valores que entre sí no suman el valor de la predicción.
2. Suposición modelo complejo es linear en entorno local no está basada en teoría.
3. Definición de distancia es arbitraria.
4. Permutaciones de tabular data en alta dimensión no son triviales.
5. Produce explicaciones no comparables.

Librería **lime** para R y Python. Madura, produce buenas visualizaciones. En R, integración directa con *mlr* y *caret* pero no con el resto de modelos.



Source: https://cran.r-project.org/web/packages/lime/vignettes/Understanding_lime.html

Referencias

- Book: “[Interpretable Machine Learning, a guide for making black box models explainable](#)”, by *Christoph Molnar* (capítulo 6 especialmente, centrado en model-agnostics methods)
- SHAP: [package \(Python\)](#), [explicación detallada](#) (por el autor del paper), [explicación sencilla](#)
- XGBoostExplainer: [explicación teórica](#) y [package \(en R\)](#)
- LIME: [explicación teórica \(1\)](#), [intuición teórica \(2\)](#) y [ejemplo práctico \(en R\)](#)
- BreakDown: [package \(en R\)](#), [paper original](#)
- Ceteris Paribus: [Introducción al package \(en R\)](#)
- Towards A Rigorous Science of Interpretable Machine Learning – Finale Doshi-Velez, Been Kim (2017),
<https://arxiv.org/abs/1702.08608>
- Hands On Tutorial: [Hands-on Machine Learning Model Interpretation](#)



Gracias

Diego Yus

diego.yus@kernel-analytics.com

Kernel Analytics, S.L.

Madrid

Joaquín Bau, 2 1º C 28036 Madrid +34 915022390

Barcelona

Balmes 89, Planta 6º, pta. 4ª, 08008 +34 932506437

For further information:
www.kernel-analytics.com
info@kernel-analytics.com

Follow us:

twitter.com/kernelanalytics
linkedin.com/company/kernel-analytics