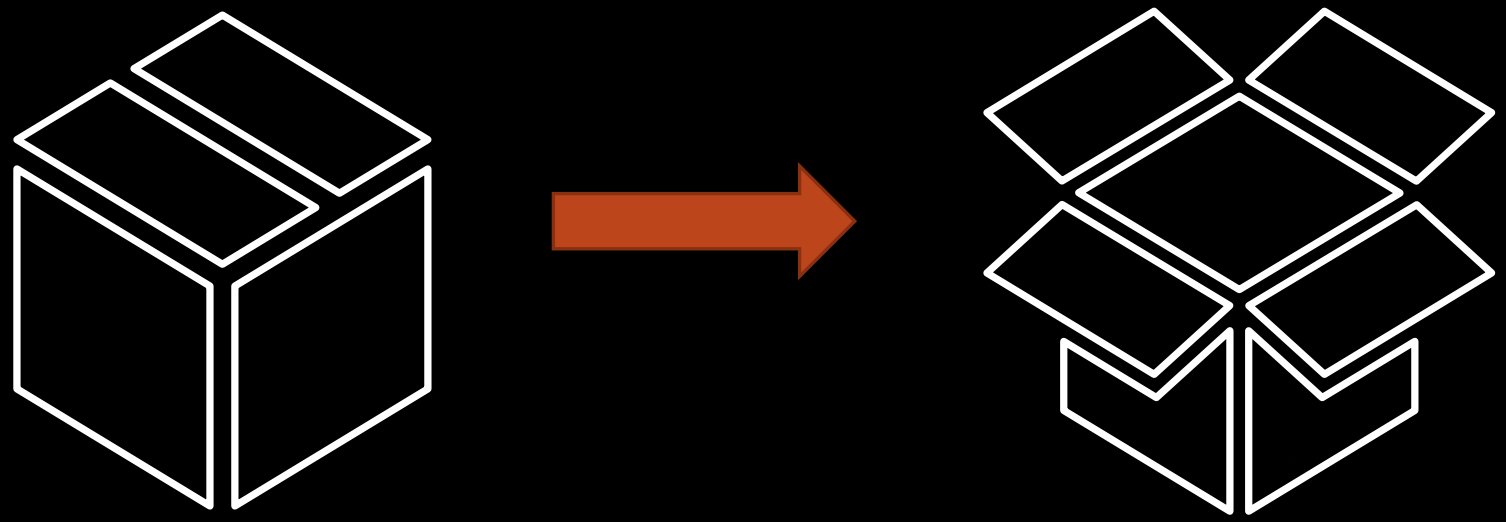
Explicando modelos de Aprendizaje Automático



Ing. Daniel Hernández Mota, Científico de datos.

Agosto, 2021

Daniel Hernández Mota:

Ingeniero en Nanotecnología
(Computación Cuántica)

Científico de datos en Kueski
(Modelo de fraude)

Mentor en SaturdaysAi
(Guadalajara 2 y 3, LATAM 1 y 2)

Saturdays.AI

Casi un IronHacker

IRON HACK

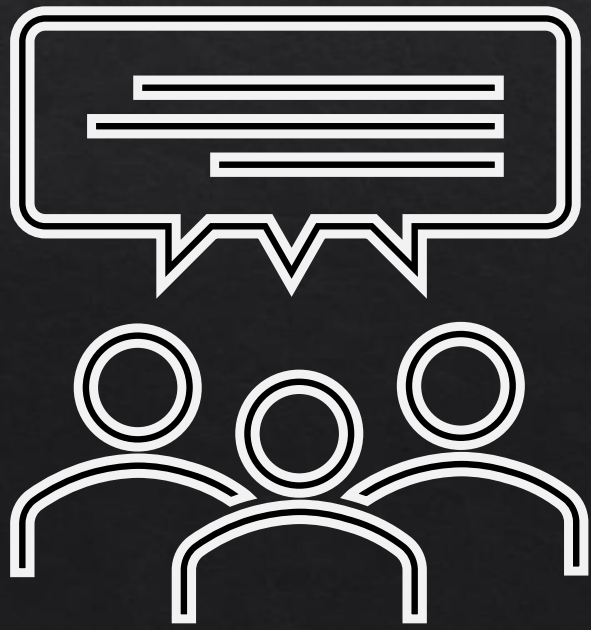
dhdzmota

¿Por qué es importante explicar un modelo de Aprendizaje Automático?

Confianza

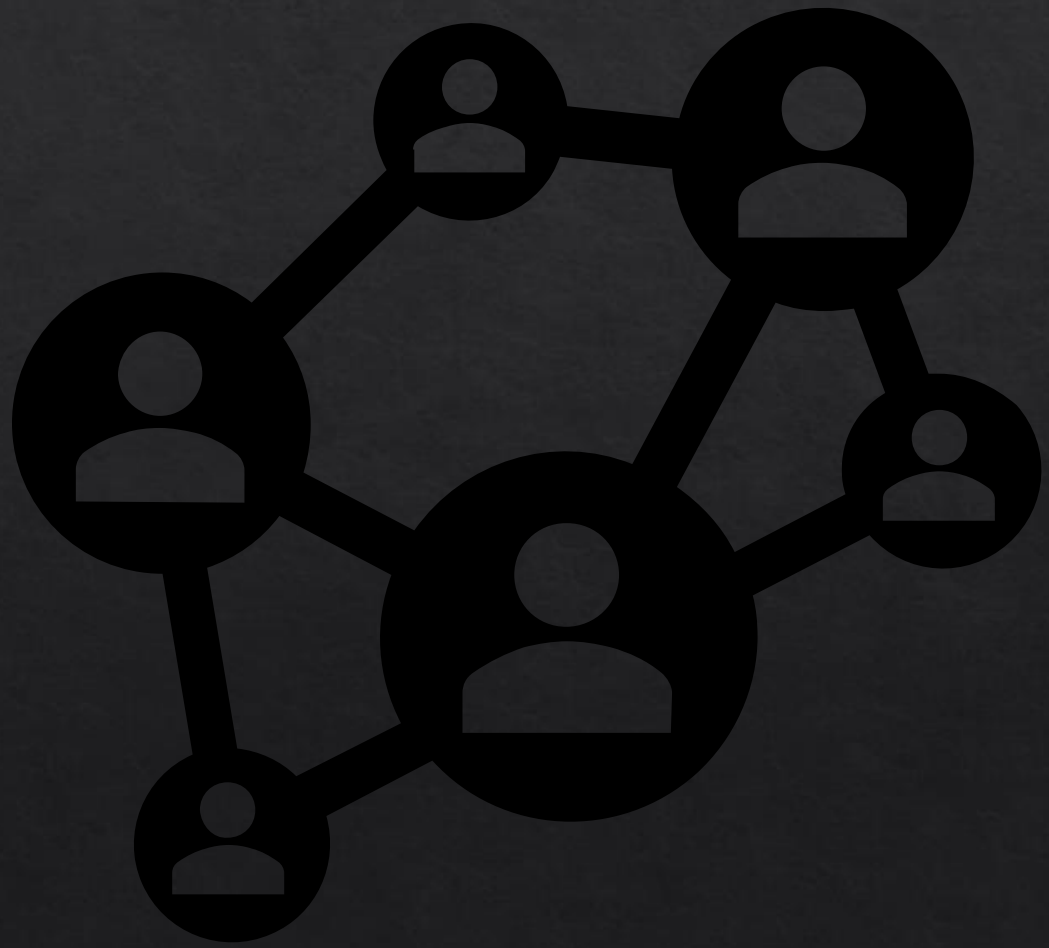
- Sentido.
- Lógica.
- Apego a la realidad.

Si los usuarios no confían en el desempeño del modelo o en la predicción, no lo usarán.



La IA se usa en todos lados…

- Identificación de objetos
- Seguros
- Redes Sociales
- Medicina
- Bancos
- Préstamos
- Medio ambiente
- Terrorismo
- Prisión



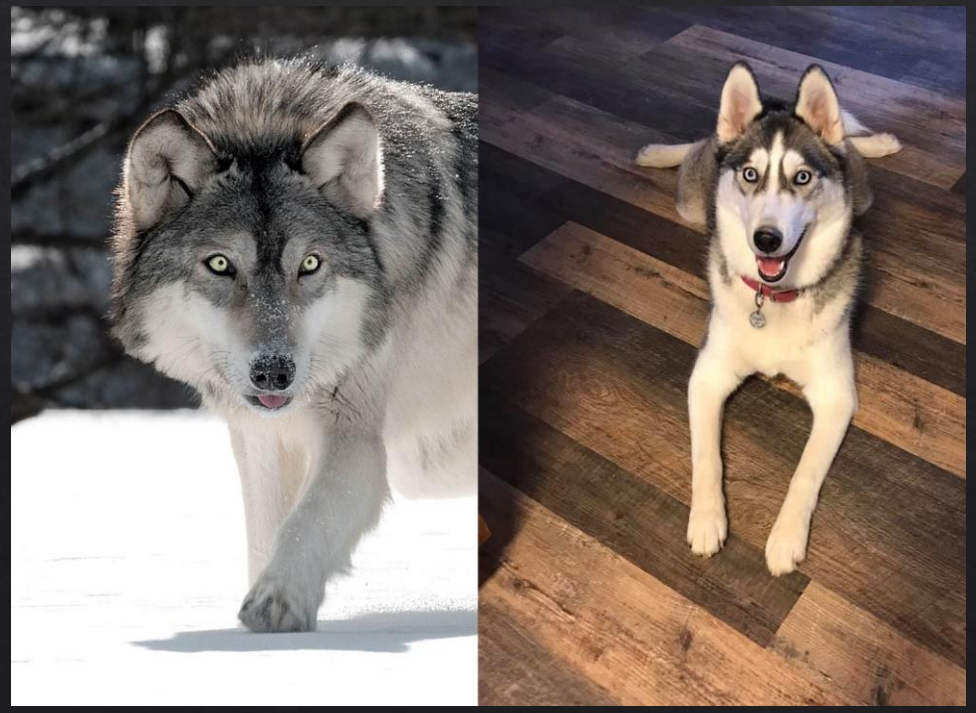
¿Cómo sabemos que podemos confiar en un modelo?

Modelo con **muy buenas métricas**

[Incluso en el conjunto de prueba más riguroso]

¿Cómo sabemos que podemos confiar en un modelo?

Modelo de clasificación: ¿Husky o Lobo?

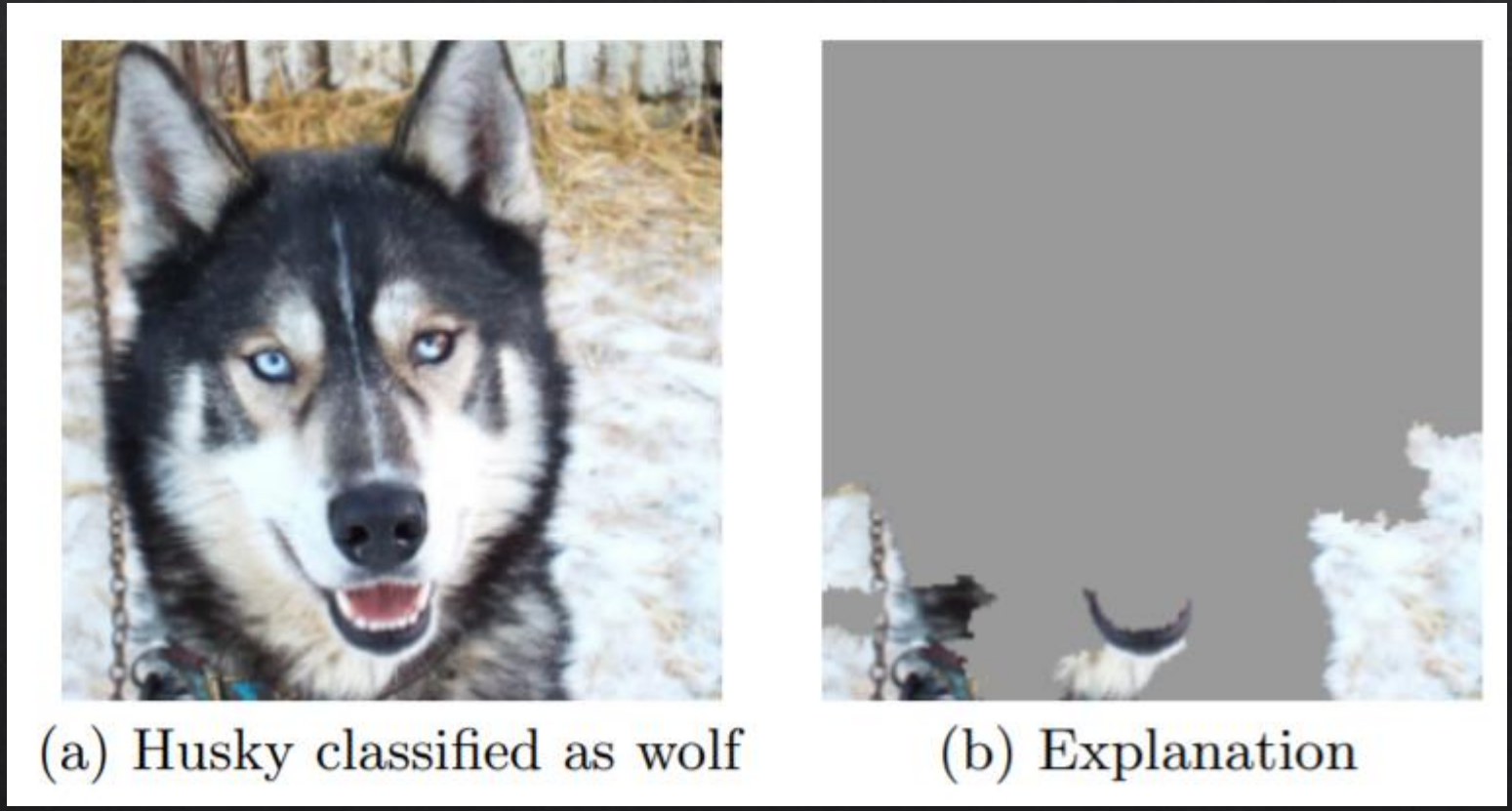


Resultados sorprendentes:

- **Métricas de evaluación (precisión, exactitud, exhaustividad, etc.) casi perfectas.**
- **Todo corroborado en el conjunto de prueba.**
- **Modelo funcionaba de maravilla… pero comenzó a fallar rápidamente…**

¿Cómo sabemos que podemos confiar en un modelo?

Falsos positivos…



(a) Husky classified as wolf

(b) Explanation

Algunas técnicas…

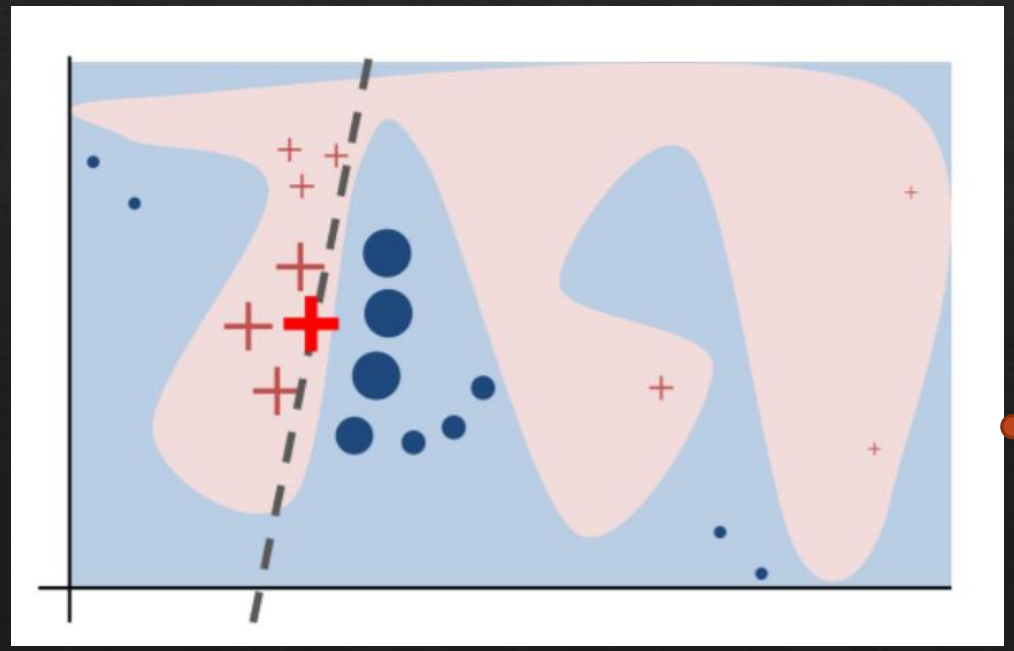
- **LIME** (Explicación local de modelos agnósticos interpretables)
- **ANCHORS** (Explicaciones de alta precisión de modelos agnósticos)
- **SHAP** (Explicaciones aditivas de Shapley)

Buscan explicar el comportamiento de los modelos a través de las variables.

LIME

LIME (modelo sustituto local)

- LIME: aproxima un modelo interpretable (regresión lineal) de manera local alrededor de una predicción.



Variables que son importantes de manera local puede que no sean importantes de manera global (vice versa)

$$LIME\ explanation = argmin_{g \in G}\ \mathcal{L}(f, g, \pi_x) + \Omega(\mathrm{g})$$

LIME (modelo sustituto local)

Objetivo formal: Identificar un modelo interpretable en el espacio de ~~variables~~ interpretación representativa, que es confiable de manera local, para un clasificador. Asegurando interpretabilidad y fidelidad local

$$LIME\ explanation = argmin_{g \in G}\ \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

- Modelo perteneciente a una clase de modelos interpretables (lineales, árboles de decisión, etc.)
- Medida de qué tan mal el modelo g interpreta f, en la localidad de π_x
- Métrica de complejidad (número de pesos que no son cero, profundidad de árboles)

LIME (modelo sustituto local)

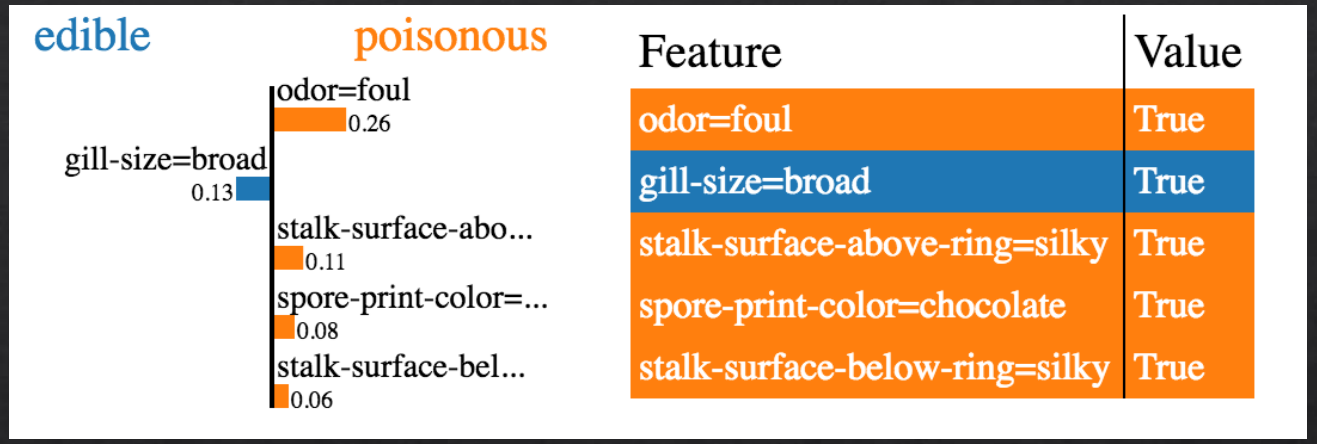
$$\mathcal{L}(f, g, \pi_x) = \sum_{z,z' \in Z} \pi_x(z)\big(f(z) - g(z')\big)^2$$

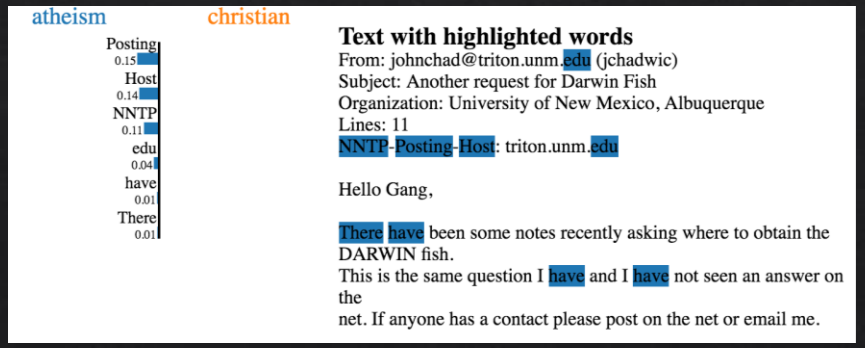
$$\pi_x(z) = e^{-D(x,z)^2/(\sigma^2)}$$

La receta…

- Selecciona la instancia de interés: datos de interés para la cual se desea obtener una explicación.
- Perturbar esa instancia y hacer que el modelo haga predicciones para esas nuevas instancias.
- Ponderar las nuevas instancias de acuerdo a la proximidad que tienen con la instancia de interés.
- Entrenar un modelo interpretable en estos conjunto de datos generado a partir de las instancias.
- **Explicar la predicción interpretando el modelo local.**

LIME (modelo sustituto local)





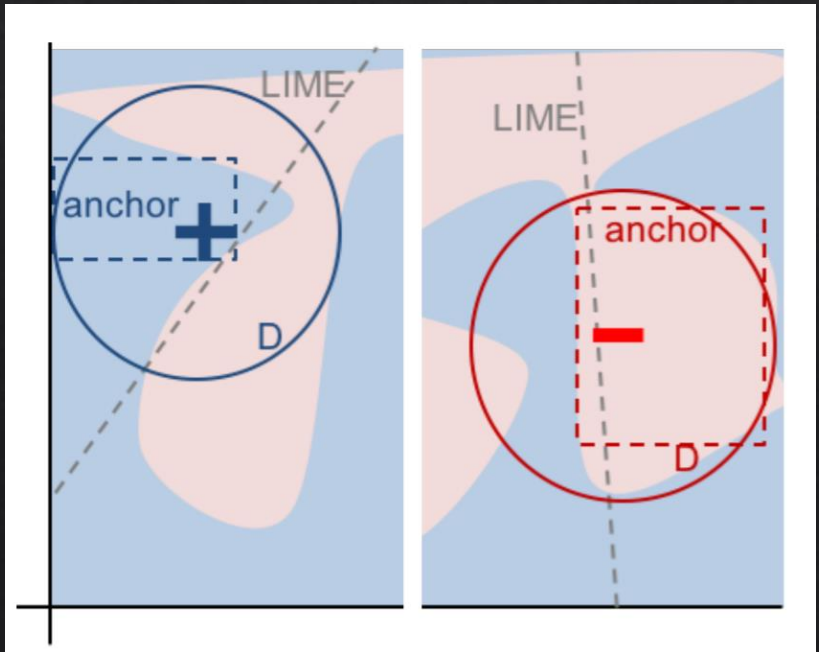
LIME (modelo sustituto local)

Ventajas	Desventajas
Sirve para debuggear modelos de Aprendizaje Máquina LIME es uno de los pocos métodos que utilizan datos tabulares, texto, e imágenes. Es extremadamente fácil de usar (librerías en R y Python)	La definición de vecindad de perturbación no es tan sencilla de hacer. Inestabilidad de explicación: la confianza no es tan buena. Es algo lento (en particular para imágenes)

Anchor

Anchor

- Esta técnica explica un modelo generando una regla de decisión (anchor), tal que, dado a que se cumple esa regla, la predicción no se verá modificada sustancialmente.



IF (country == 'United_states' AND
capital_loss < 1000 AND
race == 'white' AND
relationship == 'Husband' AND
28 < Age < 37 AND
sex == 'male' AND
occupation == 'Data Scientist')

THEN PREDICT salary > $50K

Anchor

Definición formal: A es una regla (conjunto de sentencias) que actúa sobre variables tal que $A(x) = 1$ si todas las sentencias se cumplen para la instancia x. Entonces se puede definr que A es un ancla si $A(x) = 1$ y A es una condición suficiente para $f(x)$.

$$\max_{A \text{ s.t. } P(\text{prec}(A) \geq \tau) \geq 1-\delta} \text{cov}(A).$$

$$\text{prec}(A) = \mathbb{E}_{\mathcal{D}(z|A)}\left[\mathbb{1}_{f(x)=f(z)}\right]$$

Precisión: Proporción de valores correctos, de las instancias generadas, dentro del espacio del ancla.

$$\text{cov}(A) = \mathbb{E}_{\mathcal{D}(z)}[A(z)]$$

Cobertura: Proporción de valores dentro de la población de perturbación a los cuales les aplica el ancla.

Anchor

Ventajas	Desventajas
Reglas permiten identificar las predicciones de instancias no vistas. Explicaciones son confiables por diseño. Las explicaciones especifican para cuales instancias son válidas. Es extremadamente fácil de usar (librerías en R y Python)	Dado el gran espacio de variables, más de un ancla puede aplicar a la misma instancia (esto podría generar menor interpretabilidad) Generar la distribución de perturbación es difícil. Sigue siendo una explicación local (aunque en realidad, te dice la cobertura)

SHAP

SHAP (Shapley Additive exPlanations)

- **Técnica que emplea teoría de juegos para explicar el comportamiento del modelo.**

Valores Shapley: Método de teoría de juegos (de coaliciones) que te permite determinar cómo repartir la "paga" de manera justa entre todos los jugadores.

Una predicción se puede explicar asumiendo que cada variable es un JUGADOR en un JUEGO (tarea de predecir) donde la predicción es la "paga".

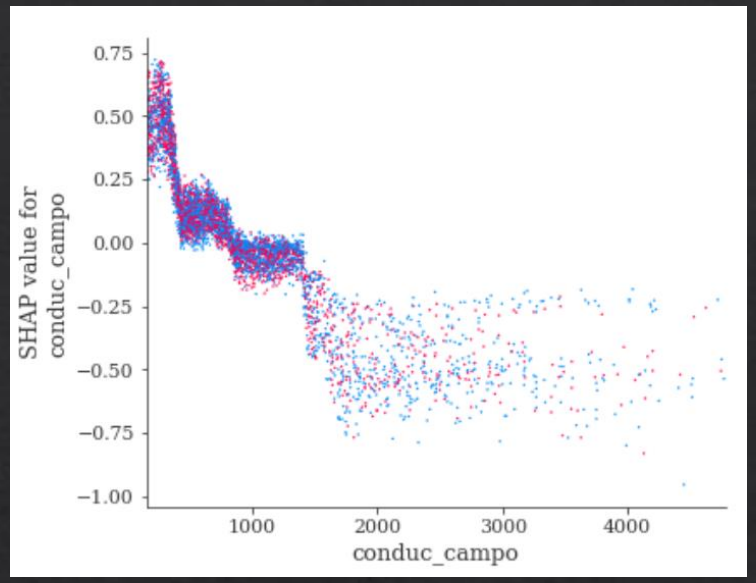
SHAP (Shapley Additive exPlanations)

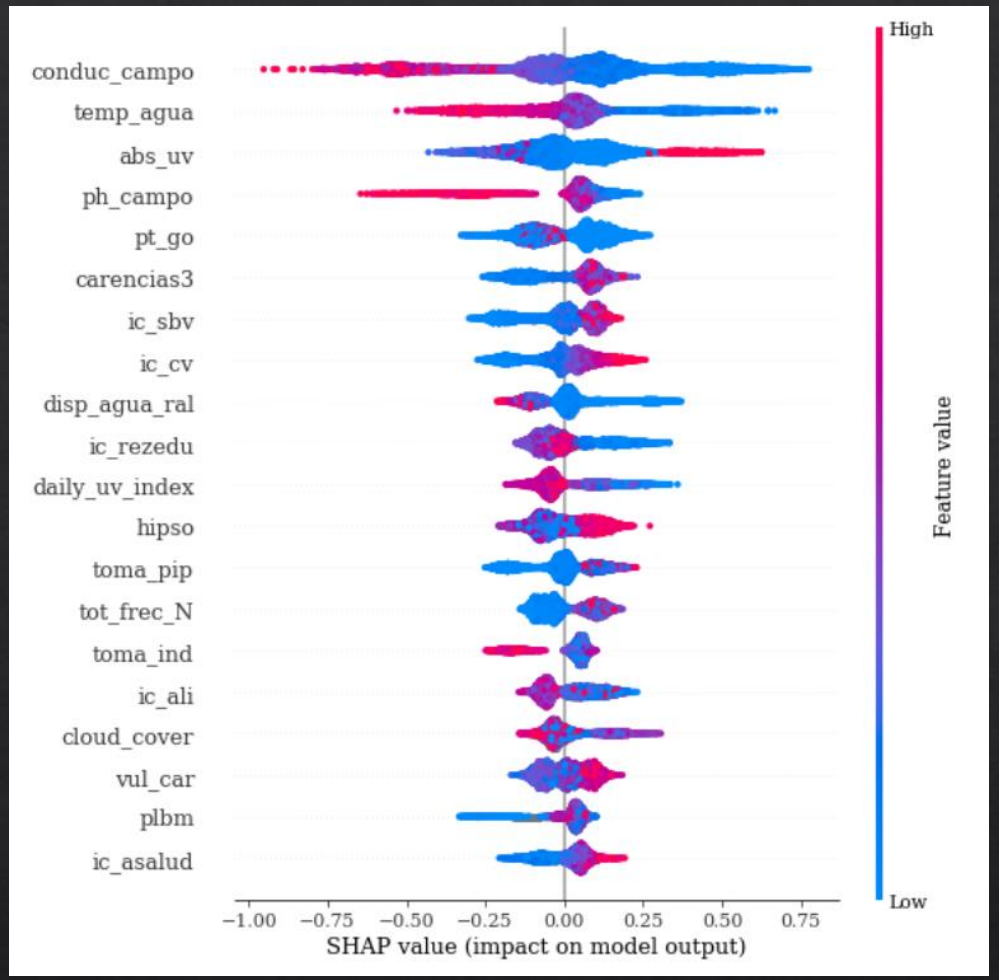
SHAP se basa en la magnitud de las contribuciones marginales promedio de las variables a lo largo de muchas muestras.

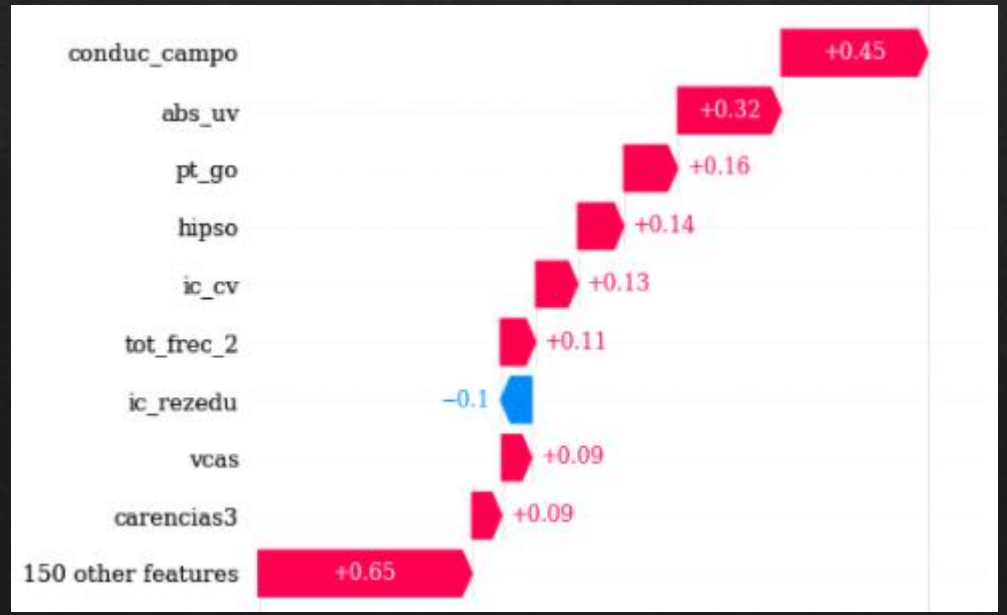
$$\phi_j(val) = \sum_{S \subseteq \{x_1,\ldots,x_p\} \setminus \{x_j\}} \frac{|S|!\,(p-|S|-1)!}{p!} \left(val\left(S \cup \{x_j\}\right) - val(S)\right)$$

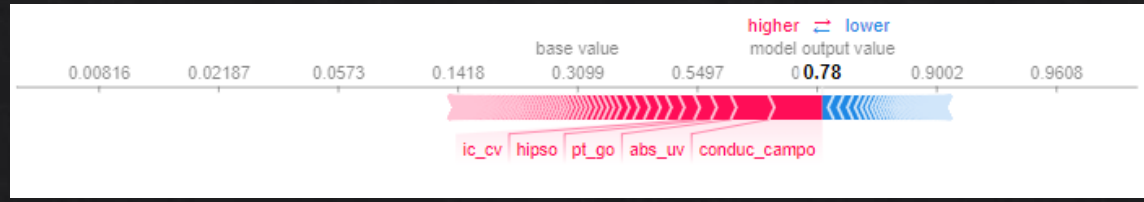
En realidad es la contribución a la paga (sumado y ponderado) entre todas las posibles combinaciones, para así detectar la contribución marginal de las variables.

SHAP


SHAP value for
conduc_campo
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
1000
2000
3000
4000
conduc_campo


conduc_campo
temp_agua
abs_uv
ph_campo
pt_go
carencias3
ic_sbv
ic_cv
disp_agua_ral
ic_rezedu
daily_uv_index
hipso
toma_pip
tot_frec_N
toma_ind
ic_ali
cloud_cover
vul_car
plbm
ic_asalud
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
SHAP value (impact on model output)
High
Feature value
Low


conduc_campo
+0.45
abs_uv
+0.32
pt_go
+0.16
hipso
+0.14
ic_cv
+0.13
tot_frec_2
+0.11
ic_rezedu
−0.1
vcas
+0.09
carencias3
+0.09
150 other features
+0.65


higher ⇄ lower
base value
model output value
0.00816
0.02187
0.0573
0.1418
0.3099
0.5497
0.78
0.9002
0.9608
ic_cv
hipso
pt_go
abs_uv
conduc_campo

SHAP

Ventajas	Desventajas
Distribución Equitativa por justicia (solución única demostrada por teoremas) Explicación completa y comprensiva por cada variable Explicaciones locales y globales Implementación optimizada para árboles (XGBOOST, CatBoost LGBM) Es extremadamente fácil de usar (librerías en R y Python)	Obtener valores Shapley requiere mucho tiempo computacional Se deben de usar aproximaciones para atenuar ese tiempo. Siempre se requiere información para calcular el valor shapley (no hay perturbación) Explicabilidad no es multivariada.

CONCLUCIÓN:

Explicar un modelo de Aprendizaje Automático brinda confianza, seguridad, e interpretación a las decisiones que está realizando el modelo para predecir.

Ayuda a encontrar:

- Patrones y sesgos en los datos.
- Importancia en las variables (target/feature leaking).
- La razón de la predicción.

Vamos a programar…

- **https://github.com/dhdzmota/explaining_ml_ironhack**

Referencias

Tulio, M., Singh, S., y Guestrin, C. (2016) Why should I trust you? Explaining the predictions of any classifier. Recuperado el 17 de octubre del 2020 de: https://arxiv.org/abs/1602.04938

Tulio, M. (2020) Lime: Explaining the predictions of any machine learning classifier. GitHub. Recuperado el 17 de octubre del 2020 de : https://github.com/marcotcr/lime

Tulio, M. (2016) Lime 0.2.0.1. GitHub. Recuperado el 01 de Noviembre del 2020 de : https://pypi.org/project/lime/#history

Sharma, A. (2018) Decrypting your Machine Learning model using LIME. Recuperado el 01 de noviembre del 2020 de: https://towardsdatascience.com/decrypting-your-machine-learning-model-using-lime-5adc035109b5

Korobov, M. y Lopuhin, K., (2016) ELI5. Recuperado el 17 de octubre del 2020 de: https://eli5.readthedocs.io/en/latest/overview.html

Korobov, M. y Lopuhin, K., (2016) ELI5. Recuperado el 17 de octubre del 2020 de: https://github.com/TeamHG-Memex/eli5

Tulio, M., Singh, S y Guestrin C. (2018) Anchors: High-Precision Model-Agnostic Explanations. Recuperado el 17 de octubre del 2020 de: https://homes.cs.washington.edu/~marcotcr/aaai18.pdf

Tulio, M. (2019) Anchor. Github. Recuperado el 17 de octubre del 2020 de: https://github.com/marcotcr/anchor

Lundber, S. Su-In, L. (2017) A unified approach to interpreting model predictions. Advances in Neural Information Processing Systems. Recuperado el 17 de octubre.

Shapley, L., (1953) A value for n-person games. Contributions to the Theory of Games 2.28:

Molnar, C. (2020) Interpretable Machine Learning. Recuperado el 18 de septiembre del 2020 de: https://christophm.github.io/interpretable-ml-book/shap.html