

# XAI: LIME & SHAP



**UNIVERSIDAD  
DE GRANADA**

Asignatura: Tratamiento Inteligente de Datos

# Índice

1. Motivación: problema y contexto.....	4
2. Algoritmos.....	5
2.1. SHAP (SHapley Additive Explanations).....	5
2.1.1. Cómo funciona SHAP.....	6
2.1.2. Aplicaciones prácticas.....	7
2.1.3. Ventajas.....	7
2.1.4. Desafíos.....	7
2.2. LIME (Local Interpretable Model-agnostic Explanations).....	7
2.2.1 Descripción general.....	7
2.2.2. Cómo funciona LIME.....	8
2.2.3. Aplicaciones prácticas.....	8
2.2.4. Ventajas.....	8
2.2.5. Desafíos.....	9
3. Demostraciones.....	9
3.1. Demostración 1 SHAP.....	9
3.2. Demostración 2: Interpretabilidad local y global con SHAP.....	13
3.3. Demostración: Explicación local con LIME.....	14
4. Comparación con entre SHAP y LIME.....	15
5. Aplicaciones: casos reales y resultados obtenidos.....	16
5.1 Aplicaciones de LIME.....	16
5.2 Aplicaciones de SHAP.....	19
6. Conclusión.....	22
Apéndice A. Código de las demostraciones.....	23
Apéndice B. Descripción detallada de los datasets.....	24
Apéndice C. Glosario de términos.....	25
5. Referencias:.....	26

# Índice de figuras

Figura 1: SHAP.....	6
Figura 2: Carga de los datos.....	9
Figura 3: División en dos subconjuntos.....	10
Figura 4: Entrenamiento.....	10
Figura 5: aplicación de SHAP.....	10
Figura 6: Gráfica 1.....	11
Figura 7: Gráfica 2.....	12
Figura 8: Gráfica 3.....	13
Figura 9: Gráfica 4.....	14
Figura 10: Visualización de la explicación.....	15
Tabla 1: Comparación SHAP y LIME.....	16
Figura 11: Ejemplo de explicación de una predicción en clasificación de texto mediante LIME, mostrando las palabras más relevantes y su contribución a cada clase (adaptado de Ribeiro et al., 2016).....	17
Figura 12: Explicación de una predicción en clasificación de imágenes mediante LIME, resaltando los superpíxeles relevantes para distintas clases predichas (adaptado de Ribeiro et al., 2016).....	17
Tabla 2: Comparación del rendimiento de distintos métodos de explicación en la evaluación de la confianza en predicciones individuales (adaptado de Ribeiro et al., 2016).....	18
Figura 13: Evolución del rendimiento del modelo tras sucesivas rondas de mejora guiadas por explicaciones LIME y SP-LIME (adaptado de Ribeiro et al., 2016).....	19
Figura 14: Explicación de una predicción en clasificación de imágenes mediante SHAP, LIME y DeepLIFT, resaltando las regiones visuales más relevantes para la clase predicha y el efecto de su eliminación sobre la salida del modelo (adaptado de Lundberg y Lee, 2017).....	20
Figura 15: Comparación entre explicaciones humanas y explicaciones generadas por SHAP, LIME y DeepLIFT en distintos escenarios experimentales, mostrando la mayor coherencia de SHAP con la asignación de importancia humana (adaptado de Lundberg y Lee, 2017).....	21
Figura 16: Comparación de la estabilidad y eficiencia de las explicaciones generadas por SHAP, Shapley Sampling y LIME en función del número de evaluaciones del modelo, mostrando la mayor precisión y consistencia de SHAP (adaptado de Lundberg y Lee, 2017).....	22

## 1. Motivación: problema y contexto

En los últimos años, el aprendizaje automático ha experimentado un notable avance gracias al desarrollo de modelos cada vez más complejos y precisos, como las redes neuronales profundas, los ensamblados de modelos y otros métodos de alto rendimiento. Estos enfoques han demostrado una gran eficacia en tareas de clasificación, regresión y predicción en múltiples dominios. Sin embargo, esta mejora en el rendimiento suele ir acompañada de una pérdida significativa de interpretabilidad, dando lugar a los denominados modelos de caja negra, cuyo funcionamiento interno resulta difícil o incluso imposible de entender para los humanos (*Barredo Arrieta et al., 2020*).

La falta de transparencia en los modelos de aprendizaje automático plantea problemas importantes en contextos donde las decisiones tienen un impacto directo sobre las personas y la sociedad. En ámbitos como la medicina, las finanzas, la justicia o la gestión de recursos humanos, no basta con obtener una predicción precisa; hace falta comprender por qué el modelo ha tomado una determinada decisión. Esta necesidad está relacionada con aspectos clave como la confianza en el sistema, la detección de sesgos, la rendición de cuentas (accountability), el cumplimiento de normativas y la garantía de un uso ético y responsable de la inteligencia artificial (*Doshi-Velez and Kim, 2017* y *Barredo Arrieta et al., 2020*).

En este contexto surge el campo de la *eXplainable Artificial Intelligence (XAI)*, cuyo objetivo principal es dotar a los sistemas de aprendizaje automático de mecanismos que permitan explicar su comportamiento de forma comprensible para los humanos. Según *Gunning (2017)*, la XAI busca desarrollar técnicas que permitan a los usuarios entender, confiar y gestionar adecuadamente los sistemas inteligentes. De este modo, la explicabilidad se convierte en un requisito fundamental para la adopción real y responsable de la inteligencia artificial en aplicaciones críticas.

Dentro de las técnicas de XAI, LIME (Local Interpretable Model-agnostic Explanations) y SHAP (SHapley Additive exPlanations) destacan como dos de los métodos más influyentes y usados. Ambas son técnicas post-hoc<sup>1</sup> y agnósticas al modelo<sup>2</sup>, lo que significa que pueden aplicarse a cualquier tipo de clasificador o regresor sin necesidad de conocer su estructura interna.

LIME, se basa en la idea de aproximar localmente el comportamiento de un modelo complejo mediante un modelo interpretable, como una regresión lineal o un pequeño árbol de decisión (*Ribeiro et al, 2016*). Para una predicción concreta, LIME genera perturbaciones alrededor de la instancia de interés y aprende un modelo sencillo que imita la respuesta del modelo original en esa vecindad. De este modo, se obtiene una explicación local que identifica qué características han sido más relevantes para esa predicción específica. Su principal ventaja es la simplicidad conceptual y la facilidad de interpretación de los resultados.

Por otro lado, SHAP, se fundamenta en la teoría de juegos cooperativos y, en particular, en los valores de Shapley (*Lundberg y Lee, 2017*). Este enfoque proporciona una asignación teóricamente justificada de la contribución de cada característica a la predicción de un modelo. SHAP cumple una serie de propiedades deseables, como la consistencia y la precisión local, que garantizan que las explicaciones sean coherentes y comparables entre distintos modelos. Además, SHAP permite obtener tanto explicaciones locales (para una predicción individual) como explicaciones globales (para comprender el comportamiento general del modelo).

La motivación principal de este trabajo radica, por tanto, en la necesidad de abordar el problema de la opacidad de los modelos de aprendizaje automático mediante herramientas que permitan interpretar y justificar sus decisiones. LIME y SHAP representan dos aproximaciones complementarias a este desafío: LIME prioriza la interpretabilidad local a través de modelos simples, mientras que SHAP ofrece un marco teórico sólido para la asignación de importancia a las características. Ambos ayudando a promover el desarrollo de sistemas de inteligencia artificial más transparentes y confiables.

## 2. Algoritmos

### 2.1. SHAP

SHAP (SHapley Additive Explanations) es una técnica de interpretabilidad en el ámbito del aprendizaje automático que se basa en los valores de Shapley, un concepto matemático derivado de la teoría de juegos cooperativos desarrollado por Lloyd Shapley en 1953. Este método busca asignar una contribución justa a cada participante en un juego. En el contexto de aprendizaje automático, los "jugadores" son las características del modelo, y el "juego" es la predicción generada por el modelo.

SHAP proporciona una manera equitativa y matemáticamente sólida de descomponer una predicción en contribuciones individuales de las características, permitiendo entender cómo y por qué un modelo genera sus resultados. Esto lo hace especialmente útil en entornos críticos donde la explicabilidad es fundamental, como la medicina, las finanzas y los sistemas judiciales.



# SHAP

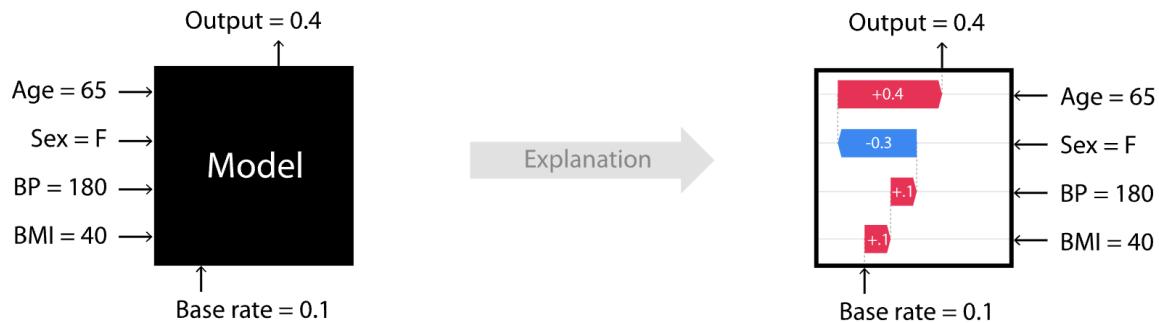


Figura 1: SHAP

### 2.1.1. Cómo funciona SHAP

SHAP mide el impacto promedio de cada característica en todas las posibles combinaciones del conjunto de características. Para calcular el valor de Shapley ( $\phi$ ) de una característica  $i$ , considera todas las permutaciones posibles de las características y evalúa cómo cambia la predicción del modelo cuando la característica  $i$  se incluye o excluye:

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

Donde:

- $N$  es el conjunto de todas las características.
- $S$  es un subconjunto de  $N$  que no incluye  $i$ .
- $f(S)$  es la predicción del modelo considerando sólo las características en  $S$ .

Este enfoque garantiza una atribución justa y consistente del impacto de cada característica en la predicción del modelo.

SHAP presenta los resultados de manera visual e intuitiva:

- Gráficos de barras: resaltan las características más influyentes a nivel global
- Mapas de calor: muestran cómo varían las contribuciones en diferentes instancias

- Gráficos de dispersión: relacionan las contribuciones de las características con sus valores

### 2.1.2. Aplicaciones prácticas

- **Modelos financieros**

En un modelo de aprobación de créditos, SHAP ayuda a identificar las razones detrás de cada decisión. Por ejemplo, puede mostrar que han rechazado a un cliente por que tiene un ingreso mensual bajo o un historial crediticio irregular.

- **Diagnóstico médico**

SHAP permite descomponer las decisiones de un modelo de diagnóstico para identificar qué síntomas, pruebas o factores clínicos influyeron más en un diagnóstico. Esto es crucial para aumentar la confianza de los médicos en los modelos predictivos.

- **Sistemas de recomendación**

En plataformas como e-commerce, SHAP explica qué atributos de un producto (precio, reseñas, categoría) han influido en su recomendación para un usuario.

### 2.1.3. Ventajas

SHAP asegura que si una característica tiene mayor impacto en la predicción, recibirá un valor proporcionalmente mayor, asegurando que las explicaciones sean fiables y matemáticamente coherentes.

Además, puede explicar tanto una sola predicción (nivel local) como el comportamiento general del modelo (nivel global), y aunque puede aplicarse a cualquier modelo, existen versiones optimizadas, como Tree SHAP, que lo hacen eficiente para modelos basados en árboles (e.g., XGBoost, LightGBM).

### 2.1.4. Desafíos

El principal desafío es el costo computacional, ya que calcular todas las combinaciones posibles es lento en modelos grandes, aunque existen optimizaciones como Tree SHAP para modelos basados en árboles.

## 2.2. LIME

LIME (Local Interpretable Model-agnostic Explanations) es una técnica diseñada para explicar predicciones individuales de modelos de aprendizaje automático mediante la construcción de modelos interpretables simples que aproximan el comportamiento de un modelo complejo en torno a una instancia específica. Fue introducida por *Ribeiro et al.* en 2016 como una forma de aumentar la confianza y comprensión de los usuarios en los modelos predictivos, especialmente en aquellos que funcionan como "cajas negras"<sup>3</sup>.

El principio clave de LIME es que, aunque un modelo global puede ser difícil de interpretar, es posible aproximar su comportamiento localmente alrededor de un dato

específico utilizando un modelo sencillo, como una regresión lineal o un árbol de decisión de baja profundidad. Esto permite a los usuarios entender por qué el modelo realizó una predicción concreta.

### 2.2.1. Cómo funciona LIME

#### 1. Perturbaciones locales

LIME comienza generando un conjunto de datos artificiales similares a la instancia que se quiere explicar. Esto se hace alterando aleatoriamente las características de la instancia original para crear ejemplos perturbados. Por ejemplo, si la instancia original tiene una característica "edad = 30", LIME podría generar ejemplos con "edad = 25" o "edad = 35".

#### 2. Modelo interpretativo

Una vez generadas las perturbaciones, se calcula la predicción del modelo complejo para cada uno de estos ejemplos. LIME ajusta un modelo simple (como una regresión lineal o un árbol de decisión) utilizando las características perturbadas y las predicciones del modelo complejo. Este modelo simple se pondera según la proximidad de los ejemplos perturbados a la instancia original, dándole mayor importancia a los puntos cercanos.

#### 3. Resultados visuales

El modelo interpretativo generado identifica las características más importantes que influyen en la predicción del modelo complejo. LIME presenta estos resultados en gráficos fáciles de interpretar, como gráficos de barras que muestran el impacto positivo o negativo de cada característica.

### 2.2.2. Aplicaciones prácticas

#### 1. Diagnósticos médicos

LIME puede ayudar a explicar por qué un modelo predice que un paciente tiene una enfermedad específica. Por ejemplo, puede mostrar que síntomas como fiebre alta o presión arterial elevada han sido los factores más relevantes en la predicción.

#### 2. Modelos financieros

En sistemas de aprobación de créditos, LIME permite entender por qué un cliente ha sido rechazado o aprobado. Por ejemplo, puede indicar que un historial crediticio sólido contribuyó positivamente, mientras que un ingreso mensual bajo tuvo un impacto negativo.

#### 3. Sistemas de recomendación

En aplicaciones como plataformas de streaming, LIME puede explicar por qué se recomendó una película o serie específica a un usuario, destacando atributos como el género, el director o las preferencias anteriores del usuario.

### 2.2.3. Ventajas

LIME puede trabajar con cualquier tipo de modelo de aprendizaje automático, ya sea lineal, basado en árboles, redes neuronales profundas, entre otros. Por otra parte, al

construir modelos locales simples, las explicaciones generadas son fáciles de entender incluso para usuarios sin experiencia técnica. Además, comparado con técnicas más complejas como SHAP, LIME tiende a ser más rápido en la generación de explicaciones locales, ya que no requiere evaluar todas las combinaciones posibles de características.

#### 2.2.4. Desafíos

Los resultados de LIME pueden depender en gran medida de cómo se eligen los parámetros, como la proximidad utilizada para ponderar las perturbaciones y el tamaño del modelo interpretativo. Por otra parte, dado que LIME se enfoca en explicaciones locales, no proporciona una visión general del comportamiento del modelo completo. Esto puede ser un inconveniente si se requiere una explicación global. Además, las perturbaciones generadas por LIME pueden no ser representativas del dominio del problema, lo que podría llevar a explicaciones que no sean completamente fiables.

### 3. Demostraciones

Con el objetivo de evaluar la aplicabilidad práctica de las técnicas SHAP y LIME, se han hecho varios experimentos utilizando conjuntos de datos relacionados con la diabetes. Estas demostraciones permiten analizar tanto la interpretación global del modelo como la explicación local de predicciones individuales, mostrando la capacidad de estas técnicas para aumentar la transparencia de modelos predictivos.

El código de las demostraciones se encuentra disponible en:  
[https://colab.research.google.com/drive/1y\\_f1qcWOqPpS5S7GWi0mFQuNus4G9TgQ?usp=sharing](https://colab.research.google.com/drive/1y_f1qcWOqPpS5S7GWi0mFQuNus4G9TgQ?usp=sharing)

#### 3.1. Demostración 1 SHAP

Se va a entrenar un modelo de regresión lineal utilizando el conjunto de datos Diabetes proporcionado por la biblioteca *scikit-learn* mediante la función *load\_diabetes()*. Este conjunto de datos contiene información de 442 pacientes y 10 variables predictoras, entre las que se incluyen la edad, el índice de masa corporal (BMI), la presión sanguínea y distintos indicadores relacionados con el colesterol y la glucosa en sangre. La variable objetivo representa una medida cuantitativa de la progresión de la enfermedad.

```
data = load_diabetes()
X, y = data.data, data.target
```

Figura 2: Carga de los datos

El conjunto de datos se divide en dos subconjuntos:

- El 80% de las muestras se utiliza para entrenamiento
- El 20% restante se reserva para prueba, empleando la función `train_test_split()`

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

*Figura 3: División en dos subconjuntos*

Posteriormente, se entrena un modelo de regresión lineal mediante la clase `LinearRegression`, con el objetivo de modelar la relación entre las características clínicas de los pacientes y la progresión de la diabetes.

```
model = LinearRegression()
model.fit(X_train, y_train)
```

*Figura 4: Entrenamiento*

Una vez entrenado el modelo, se aplica la técnica SHAP para interpretar sus predicciones. Para ello, se crea un explicador SHAP utilizando el modelo entrenado y el conjunto de entrenamiento. Este explicador permite calcular los valores SHAP, que representan la contribución individual de cada característica a la predicción final del modelo.

```
explainer = shap.Explainer(model, X_train)
shap_values = explainer(X_test)

shap.summary_plot(shap_values, X_test)
```

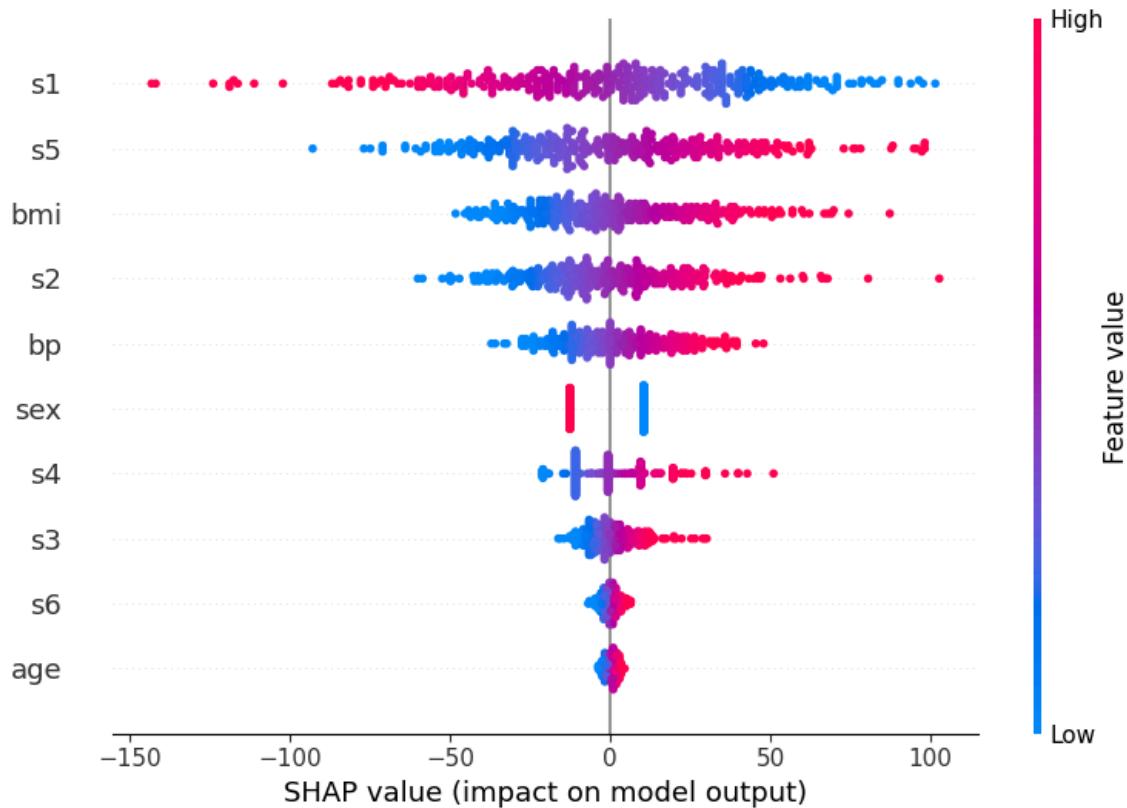
*Figura 5: aplicación de SHAP*

La visualización principal de los resultados se realiza mediante un summary plot, que muestra la importancia global de las variables y la distribución de sus contribuciones sobre todas las instancias del conjunto de datos.

## Interpretación de la Gráfica 1

En esta gráfica:

- El eje vertical representa las características del conjunto de datos
- El eje horizontal representa los valores SHAP, que indican el impacto de cada característica en la predicción
  - Valores positivos incrementan la predicción
  - Valores negativos la disminuyen
- El color de los puntos representa el valor real de la característica:
  - Rojo: valores altos
  - Azul: valores bajos



*Figura 6: Gráfica 1*

Las variables analizadas son:

- s1: colesterol LDL (colesterol “malo”)
- s5: niveles relacionados con triglicéridos o glucosa en suero
- bmi: índice de masa corporal
- s2: glucosa en sangre a largo plazo
- bp: presión sanguínea media
- sex: sexo del paciente (1 para hombres, 0 para mujeres)
- s4: albúmina sérica
- s3: colesterol HDL (colesterol “bueno”)
- s6: niveles de glucosa en suero
- age: edad del paciente.

Se observa que las características con mayor impacto global en el modelo son s1, s5 y bmi, ya que aparecen en la parte superior de la gráfica y presentan una mayor dispersión horizontal. Esto indica que el modelo basa principalmente sus predicciones en estas variables. Además:

- Para la característica s5, los valores altos tienden a generar contribuciones positivas a la predicción
- Para s2, los valores bajos presentan un impacto negativo más acusado
- La variable sex muestra valores SHAP cercanos a cero, lo que indica que su influencia en el modelo es reducida.

La dispersión de los puntos en cada característica pone de manifiesto que el impacto de una variable varía según el contexto de cada instancia, lo que refuerza la utilidad de SHAP para analizar el comportamiento del modelo.

### Interpretación de la Gráfica 2:

La segunda visualización representa la media de los valores SHAP absolutos para cada característica. Esta gráfica permite cuantificar la importancia global de las variables.

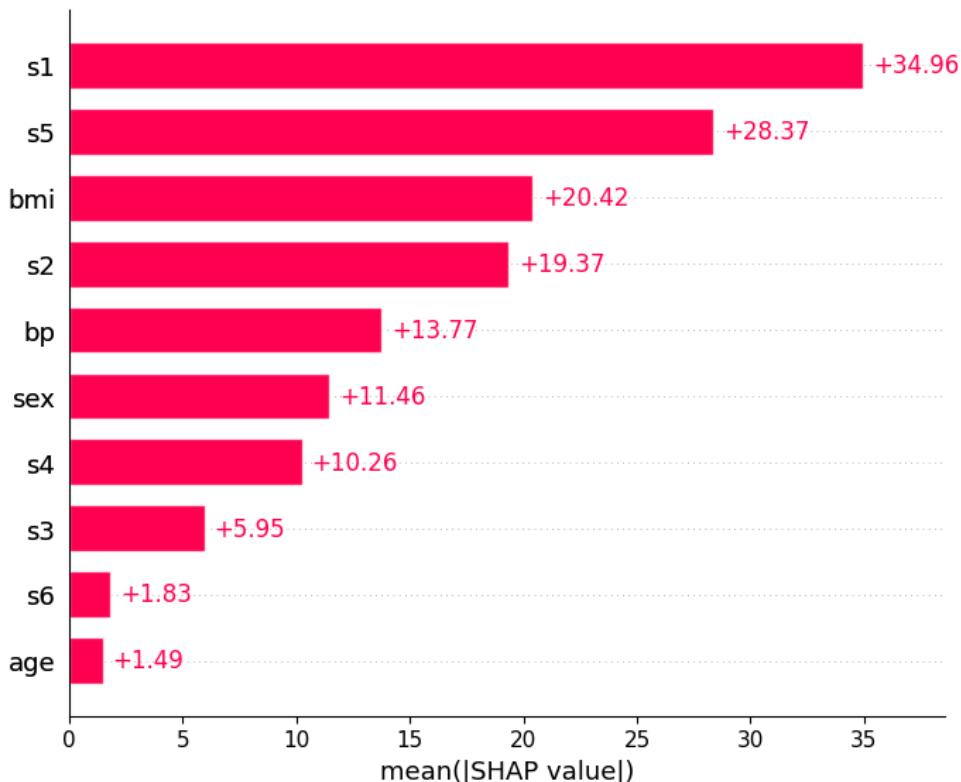


Figura 7: Gráfica 2

Se aprecia que:

- s1 y s5 presentan los valores medios SHAP más elevados (34.23 y 29.12 respectivamente)
- Estas variables son, por tanto, las más influyentes en las predicciones del modelo

Este resultado confirma que los factores relacionados con el colesterol y la glucosa desempeñan un papel central en la progresión de la diabetes según el modelo entrenado

### 3.2. Demostración 2: Interpretabilidad local y global con SHAP

En este segundo experimento se utiliza un conjunto de datos sobre diabetes enfocado en un problema de clasificación binaria, donde la variable objetivo indica la presencia o ausencia de diabetes en pacientes. El objetivo es demostrar cómo SHAP permite interpretar tanto la predicción de casos individuales como el comportamiento general del modelo.

SHAP se aplica:

- A nivel local, para explicar la predicción correspondiente a una instancia concreta. Por ejemplo, se puede observar cómo valores individuales de variables como Glucose, BMI o Age influyen positiva o negativamente en la probabilidad de que un paciente sea diagnosticado con diabetes.
- A nivel global, para identificar las características más importantes del modelo.

El análisis local revela que, para un paciente concreto, un valor bajo de Glucose empuja la predicción hacia “no diabetes”, mientras que una mayor edad incrementa la probabilidad de diagnóstico positivo. Esta visualización permite comprender cómo el modelo combina las características de un individuo para generar su predicción.

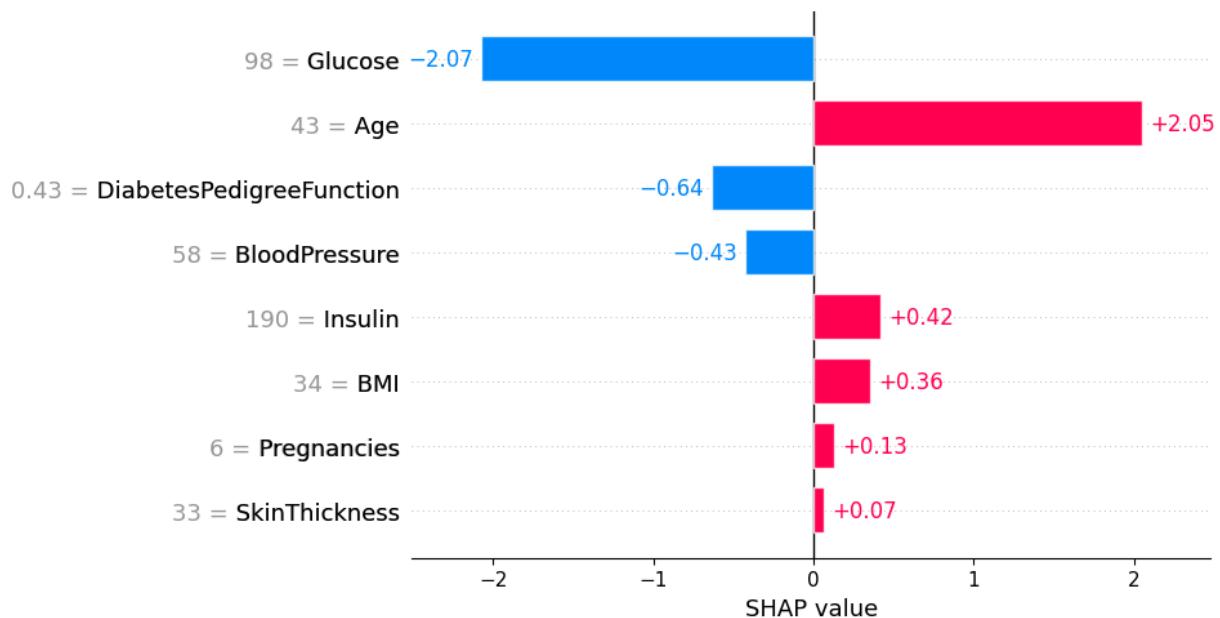
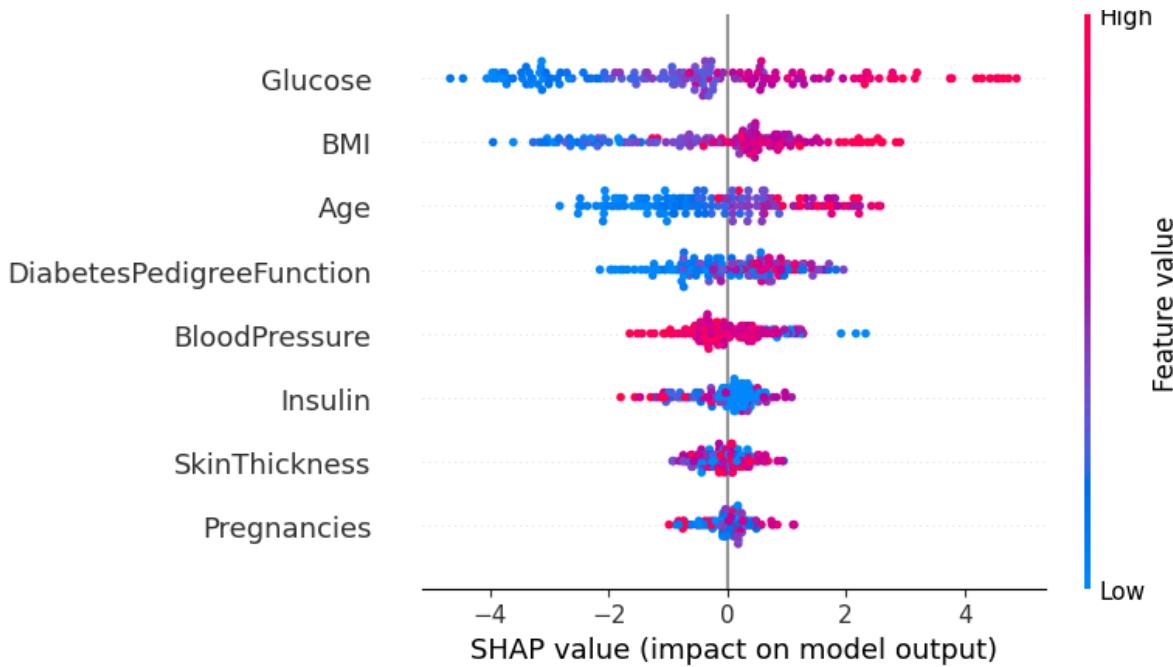


Figura 8: Gráfica 3

A nivel global, los valores de Glucose se muestran como la variable más influyente: en general, pacientes con glucosa elevada presentan mayor riesgo de diabetes. La edad también se observa como un factor relevante, con valores altos asociados a un aumento en la probabilidad de enfermedad. Otras características como BMI, BloodPressure o Pregnancies presentan contribuciones menores, pero aún relevantes según el contexto de cada paciente.



*Figura 9: Gráfica 4*

Estos resultados son coherentes con el conocimiento médico, lo que refuerza la utilidad de SHAP como herramienta de interpretabilidad.

### 3.3. Demostración 3: Explicación local con LIME

En este ejemplo, vamos a seguir con el dataset de la diabetes.

- Cargar y preparar los datos: cargamos los de diabetes y asignamos nombres a las columnas. Luego, se separan las características (X) y la variable objetivo (y), que es si la persona tiene diabetes o no (Outcome).
- Dividir los datos: dividimos el conjunto de datos en un conjunto de entrenamiento (80%) y uno de prueba (20%).
- Estandarización: normalizamos los datos para asegurar que todas las características están en la misma escala.
- Entrenamiento del modelo: entrenamos un clasificador Random Forest utilizando las características estandarizadas de entrenamiento.
- Predicción: elegimos la primera muestra del conjunto de prueba, se imprime la información del paciente y la predicción del modelo (si es diabético o no).
- Explicación con LIME: usando LIME, explicamos la predicción del modelo para la muestra seleccionada

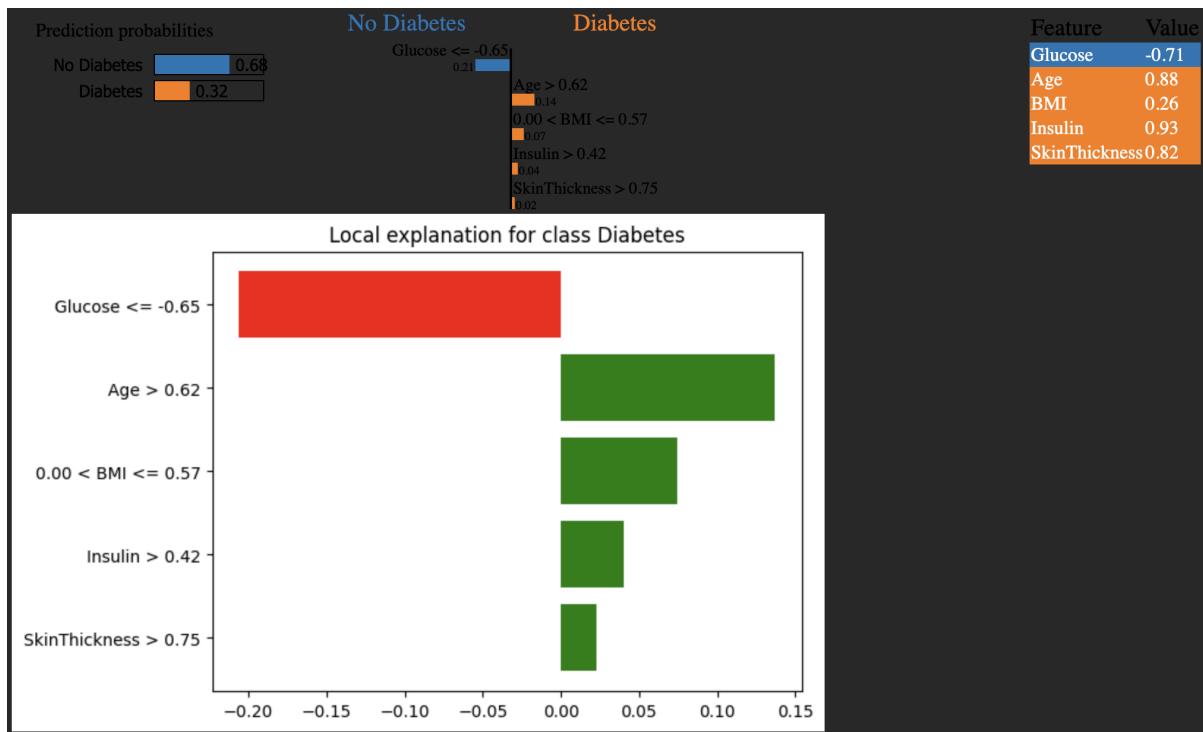


Figura 10: Visualización de la explicación

Según el modelo, existe un 68% de probabilidad de que la persona no sea diabética y hay un 32% de probabilidad de que la persona sea diagnosticada con diabetes.

Un nivel de glucosa inferior a -0.65 favorece que el paciente no tenga diabetes, mientras que una edad superior a los 62 años favorece que el paciente tenga diabetes.

#### 4. Comparación con entre SHAP y LIME

LIME y SHAP comparten el objetivo de hacer que los modelos de aprendizaje automático sean interpretables, pero sus enfoques son distintos. Mientras SHAP proporciona una explicación consistente tanto a nivel local como global, LIME se enfoca únicamente en explicaciones locales. Esto hace que LIME sea más rápido pero menos preciso en ciertas situaciones.

	LIME	SHAP
Nivel de explicación	Local únicamente	Local y global
Modelo-agnóstico	Sí	Sí
Costo computacional	Bajo a moderado	Moderado alto
Simplicidad	Alta	Moderada

Tabla 1: Comparación SHAP y LIME

## 5. Aplicaciones: casos reales y resultados obtenidos

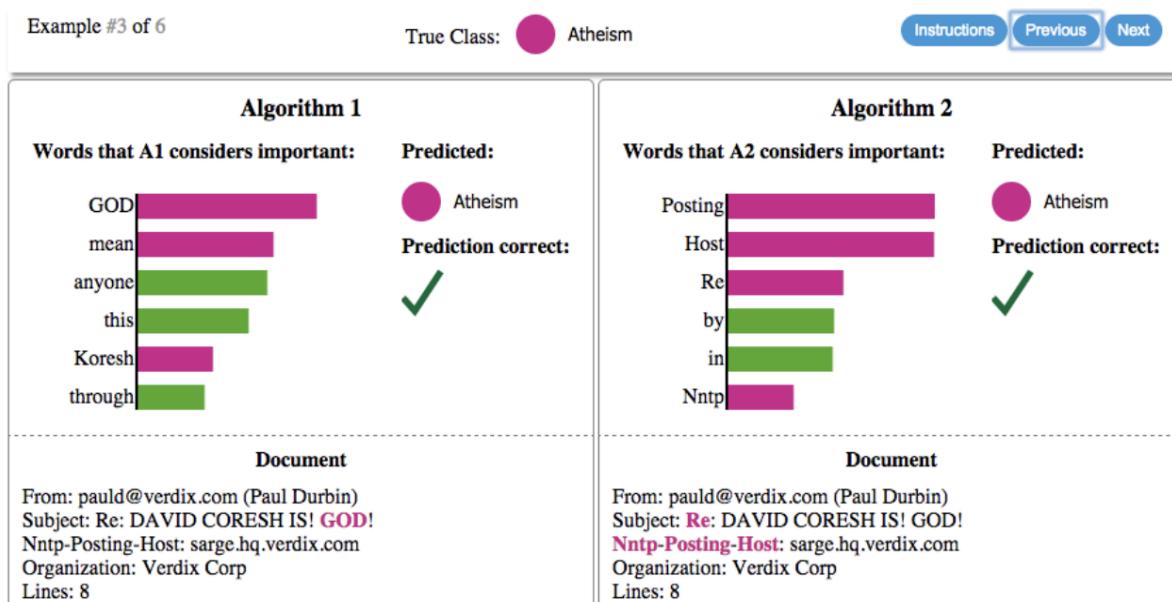
Como se ha comentado anteriormente, los métodos LIME y SHAP se aplican a la interpretación de modelos predictivos complejos con el objetivo de mejorar la comprensión humana de las decisiones automatizadas y apoyar la evaluación de confianza en sistemas de aprendizaje automático. Ambos enfoques se validan mediante estudios experimentales y casos de uso en distintos dominios, como texto, imágenes y datos tabulares.

### 5.1 Aplicaciones de LIME

#### Aplicaciones en clasificación de texto

En tareas de clasificación de texto, LIME explica predicciones individuales mediante la identificación de las palabras con mayor contribución a una clase concreta. En el dominio de clasificación temática, como el ejemplo de la distinción entre documentos relacionados con Cristianismo y Ateísmo, las explicaciones revelan qué términos influyen de forma decisiva en cada predicción (*Ribeiro et al., 2016*).

Las explicaciones muestran que modelos con alta precisión en conjuntos de validación pueden basar sus decisiones en características no semánticas, como encabezados de correo electrónico o nombres propios frecuentes en una clase. Este comportamiento indica la presencia de correlaciones “engañosas” que no generalizan a datos reales. La visualización explícita de estas características permite detectar de forma directa problemas de generalización que no resultan evidentes a partir de métricas agregadas.



*Figura 11: Ejemplo de explicación de una predicción en clasificación de texto mediante LIME, mostrando las palabras más relevantes y su contribución a cada clase (adaptado de Ribeiro et al., 2016)*

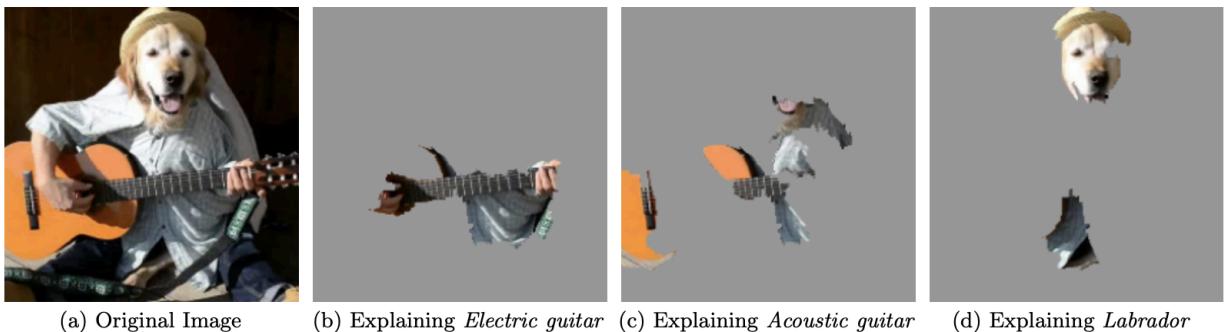
Las explicaciones facilitan la identificación de modelos poco fiables a pesar de presentar altos valores de precisión. Usuarios no expertos logran discriminar entre modelos que generalizan correctamente y modelos que dependen de artefactos del conjunto de entrenamiento (Ribeiro et al., 2016).

### **Aplicaciones en clasificación de imágenes**

LIME se aplica a modelos de visión por computador mediante representaciones interpretables basadas en superpíxeles. Cada explicación resalta las regiones de la imagen que contribuyen de forma positiva a una clase determinada, lo que permite interpretar predicciones generadas por redes neuronales profundas.

En modelos de clasificación de imágenes como Inception, las explicaciones destacan patrones visuales coherentes con el razonamiento humano, como partes específicas de un objeto o regiones del fondo. Estas explicaciones permiten comprender por qué el modelo asigna probabilidades elevadas a determinadas clases, incluso cuando la predicción principal no coincide con la etiqueta correcta (Ribeiro et al., 2016).

Las explicaciones visuales evidencian que el modelo utiliza características razonables y comprensibles, lo que incrementa la confianza en su comportamiento y facilita la detección de errores debidos a ambigüedades visuales.



*Figura 12: Explicación de una predicción en clasificación de imágenes mediante LIME, resaltando los superpíxeles relevantes para distintas clases predichas (adaptado de Ribeiro et al., 2016)*

### **Evaluación de la confianza en predicciones individuales**

LIME se utiliza para apoyar la evaluación de la confianza en predicciones individuales. En este contexto, las explicaciones permiten identificar si una predicción

depende de características consideradas no fiables o irrelevantes desde el punto de vista del dominio.

Los experimentos muestran que las explicaciones generadas por LIME permiten clasificar correctamente las predicciones como confiables o no confiables con alta precisión y exhaustividad. Este rendimiento supera al de otros métodos de explicación basados en gradientes o elecciones aleatorias (*Ribeiro et al., 2016*)

	Books				DVDs			
	LR	NN	RF	SVM	LR	NN	RF	SVM
Random	14.6	14.8	14.7	14.7	14.2	14.3	14.5	14.4
Parzen	84.0	87.6	94.3	92.3	87.0	81.7	94.2	87.3
Greedy	53.7	47.4	45.0	53.3	52.4	58.1	46.6	55.1
LIME	<b>96.6</b>	<b>94.5</b>	<b>96.2</b>	<b>96.7</b>	<b>96.6</b>	<b>91.8</b>	<b>96.1</b>	<b>95.6</b>

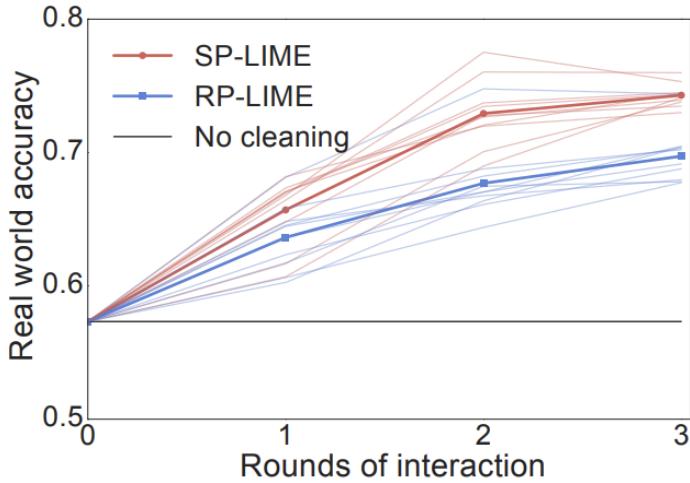
*Tabla 2: Comparación del rendimiento de distintos métodos de explicación en la evaluación de la confianza en predicciones individuales (adaptado de Ribeiro et al., 2016).*

Las explicaciones proporcionan un mecanismo eficaz para evaluar la confianza en predicciones individuales de modelos considerados cajas negras.

### Selección y mejora de modelos mediante explicaciones

LIME se combina con el método SP-LIME para seleccionar un conjunto reducido y representativo de explicaciones que refleje el comportamiento global de un modelo. Esta estrategia permite comparar modelos con rendimientos similares en validación pero con diferencias significativas en capacidad de generalización.

Las explicaciones seleccionadas permiten identificar características problemáticas y guían procesos de ingeniería de características. Usuarios sin formación especializada consiguen mejorar el rendimiento de los modelos eliminando términos irrelevantes basándose únicamente en la información proporcionada por las explicaciones (*Ribeiro et al., 2016*).



*Figura 13: Evolución del rendimiento del modelo tras sucesivas rondas de mejora guiadas por explicaciones LIME y SP-LIME (adaptado de Ribeiro et al., 2016).*

Las explicaciones facilitan la selección de modelos más robustos y permiten mejorar su generalización mediante intervenciones simples guiadas por usuarios humanos.

## 5.2 Aplicaciones de SHAP

Los métodos basados en SHAP se aplican a la interpretación de modelos predictivos complejos mediante la asignación de valores de importancia a cada característica en una predicción individual. Estas explicaciones se fundamentan en la teoría de valores de Shapley y garantizan propiedades formales como exactitud local, consistencia y manejo explícito de valores ausentes, lo que favorece interpretaciones estables y alineadas con la intuición humana (*Lundberg y Lee, 2017*)

### Aplicaciones en datos tabulares

En modelos entrenados sobre datos tabulares<sup>4</sup>, como árboles de decisión, modelos lineales y ensamblados, SHAP permite descomponer cada predicción en contribuciones aditivas de las variables de entrada. Las explicaciones muestran cómo la predicción del modelo parte de un valor base, correspondiente a la salida media del modelo, y cómo cada característica contribuye de forma positiva o negativa a desplazar dicha predicción hasta el valor final asociado a una instancia concreta.

Este enfoque facilita la comparación directa entre características y permite analizar tanto explicaciones locales como patrones globales mediante la agregación de valores SHAP. En contextos como predicción de riesgo, scoring crediticio o análisis biomédico, estas explicaciones apoyan la identificación de variables dominantes y la detección de dependencias no deseadas entre atributos (*Lundberg y Lee, 2017*).

Por ejemplo:

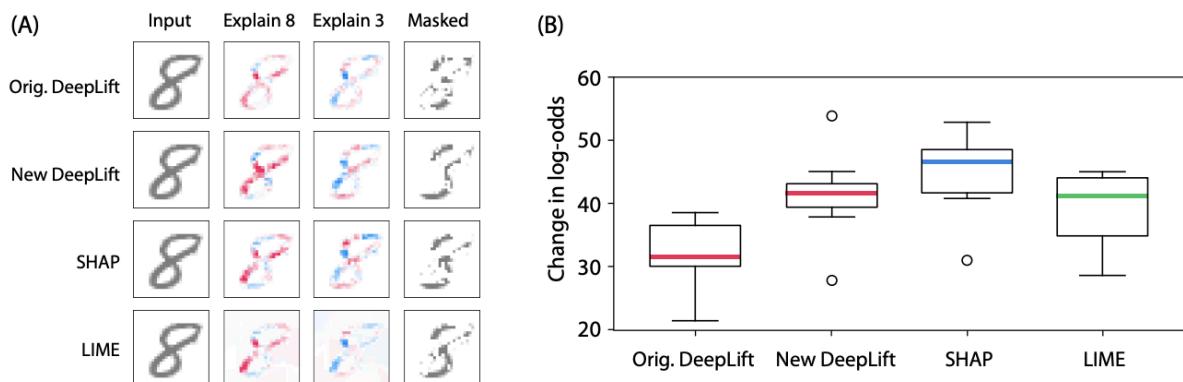
Considérese un modelo de scoring crediticio entrenado sobre datos tabulares donde cada solicitante se representa como una fila de una tabla y cada variable explicativa (historial

crediticio, nivel de ingresos y edad) corresponde a una columna, que estima la probabilidad de impago de un solicitante. El valor base del modelo, definido como la salida media del modelo sobre el conjunto de entrenamiento, es del 25 %. Para una instancia concreta, el cálculo de los valores SHAP muestra que la variable historial crediticio incrementa la predicción en 10 puntos porcentuales, la variable nivel de ingresos la reduce en 5 puntos porcentuales y la variable edad la incrementa en 3 puntos porcentuales. Estas contribuciones, obtenidas como efectos marginales medios de cada característica considerando todas las combinaciones posibles de variables, desplazan la predicción desde el valor base hasta una probabilidad final de impago del 33 %. De este modo, SHAP proporciona una descomposición aditiva y coherente de la predicción, permitiendo interpretar de forma explícita el impacto individual de cada variable en la decisión del modelo.

### Interpretación de modelos complejos y redes neuronales profundas

SHAP se adapta a modelos no lineales y de alta complejidad mediante aproximaciones específicas como Kernel SHAP<sup>5</sup> y Deep SHAP<sup>6</sup>. En redes neuronales profundas, Deep SHAP combina principios de propagación de relevancia con valores de Shapley para generar explicaciones coherentes a lo largo de las distintas capas del modelo.

En tareas de clasificación de imágenes, como el reconocimiento de dígitos manuscritos, las explicaciones resaltan regiones visuales cuya contribución resulta decisiva para la clase predicha. Estas visualizaciones permiten analizar qué partes de la imagen influyen de forma positiva o negativa en la decisión del modelo y facilitan la identificación de errores de razonamiento o ambigüedades visuales (*Lundberg y Lee, 2017*)



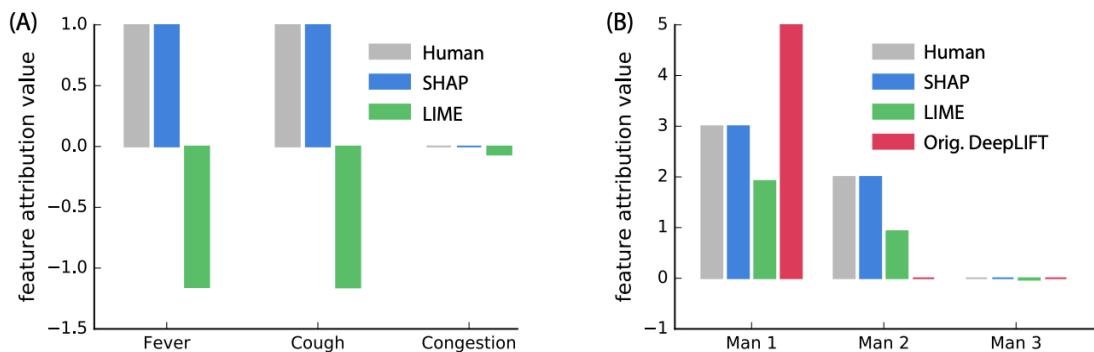
*Figura 14: Explicación de una predicción en clasificación de imágenes mediante SHAP, LIME y DeepLIFT, resaltando las regiones visuales más relevantes para la clase predicha y el efecto de su eliminación sobre la salida del modelo (adaptado de Lundberg y Lee, 2017).*

### Consistencia con la intuición humana y confianza en el modelo

Los valores SHAP muestran una coherencia alta con explicaciones humanas en estudios experimentales. En tareas simples de asignación de crédito, las explicaciones

basadas en SHAP coinciden de forma sistemática con la forma en que las personas distribuyen la responsabilidad entre las variables de entrada, incluso en funciones no lineales como operadores máximo.

Esta consistencia permite utilizar SHAP como herramienta para evaluar la confianza en predicciones individuales. Cuando una predicción se apoya en características relevantes desde el punto de vista del dominio, la explicación refuerza la credibilidad del modelo. En cambio, cuando la predicción depende de atributos irrelevantes o contraintuitivos, la explicación alerta sobre posibles problemas de fiabilidad (*Lundberg y Lee, 2017*)

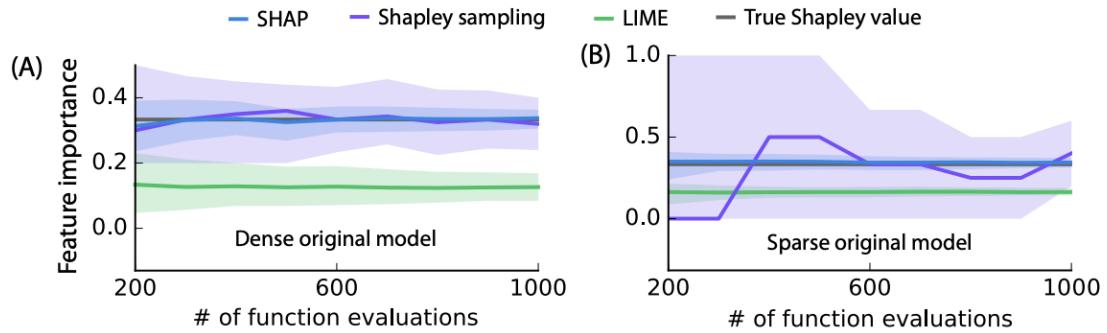


*Figura 15: Comparación entre explicaciones humanas y explicaciones generadas por SHAP, LIME y DeepLIFT en distintos escenarios experimentales, mostrando la mayor coherencia de SHAP con la asignación de importancia humana (adaptado de Lundberg y Lee, 2017).*

### Comparación y mejora de modelos mediante explicaciones SHAP

Las explicaciones SHAP facilitan la comparación entre modelos con rendimientos similares en métricas agregadas. Al analizar la distribución de importancias, resulta posible identificar modelos más estables, menos sensibles a ruido o mejor alineados con el conocimiento experto.

Además, las explicaciones guían procesos de depuración y mejora del modelo. La identificación sistemática de variables con contribuciones inconsistentes permite ajustar conjuntos de características, revisar supuestos de independencia o modificar arquitecturas complejas. Este uso convierte a SHAP en una herramienta no solo explicativa, sino también diagnóstica dentro del ciclo de desarrollo de modelos predictivos (*Lundberg y Lee, 2017*)



*Figura 16: Comparación de la estabilidad y eficiencia de las explicaciones generadas por SHAP, Shapley Sampling y LIME en función del número de evaluaciones del modelo, mostrando la mayor precisión y consistencia de SHAP (adaptado de Lundberg y Lee, 2017).*

## 6. Conclusión

Las técnicas de interpretabilidad como SHAP y LIME son importantes para hacer que los modelos de aprendizaje automático sean más comprensibles y confiables. A medida que los modelos se vuelven más complejos, estas herramientas ayudan a que los usuarios entiendan mejor las razones detrás de una predicción.

SHAP ofrece una explicación más consistente y matemática, permitiendo una comprensión tanto a nivel local como global de las predicciones, mientras que LIME se enfoca en explicar decisiones específicas de una forma un poco más rápida y sencilla. Ambas técnicas ayudan a que aumente la confianza en los modelos y a tomar decisiones más informadas.

El uso de estas herramientas no solo mejora la transparencia sino que también facilita su adopción en entornos donde la confianza y la explicación de los resultados son importantes, ya que a medida que avanzamos en el desarrollo de modelos más sofisticados, la interpretabilidad seguirá siendo una prioridad para garantizar que estas tecnologías se utilicen de manera ética y efectiva.

## Apéndice A. Código de las demostraciones

Código disponible en:

[https://colab.research.google.com/drive/1y\\_f1qcWOqPpS5S7GWi0mFQuNus4G9TgQ?usp=sharing](https://colab.research.google.com/drive/1y_f1qcWOqPpS5S7GWi0mFQuNus4G9TgQ?usp=sharing)

### Demostración 1 SHAP

```
import numpy as np
import pandas as pd
import shap
import matplotlib.pyplot as plt

from sklearn.datasets import load_diabetes
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

# 1. Carga del conjunto de datos
diabetes = load_diabetes()

X = pd.DataFrame(diabetes.data, columns=diabetes.feature_names)
y = diabetes.target

print(X.head())

# 2. División en entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

# 3. Entrenamiento del modelo
model = LinearRegression()
model.fit(X_train, y_train)

# 4. Explicador SHAP
explainer = shap.Explainer(model, X_train)
shap_values = explainer(X_train)

# 5. Visualización
# Summary plot (Figura 6: Gráfica 1)
shap.summary_plot(shap_values, X_train, show=True)

# Bar plot (importancia global promedio, correspondiente a Figura 7: Gráfica 2)
shap.plots.bar(shap_values, show=True)
```

## Demostración 2 SHAP

```
import pandas as pd
import xgboost as xgb
import shap
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split

# 1. Carga del dataset
url = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.data.csv"
columns = ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness',
           'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome']
data = pd.read_csv(url, names=columns)

X = data.drop('Outcome', axis=1)
y = data['Outcome']

# 2. División en entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

# 3. Entrenamiento XGBoost
model = xgb.XGBClassifier(eval_metric='logloss', use_label_encoder=False, random_state=42)
model.fit(X_train, y_train)

# 4. Explicador SHAP
explainer = shap.TreeExplainer(model)
shap_values = explainer(X_test)

# 5. Visualización
shap.plots.bar(shap_values[0], show=True, max_display=8) #Figura 8: Impacto local de cada variable
shap.summary_plot(shap_values, X_test, show=True) #Figura 9: Importancia global y relación de variables
```

### Demostración 3: Explicación local con LIME

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
import lime
import lime.lime_tabular
import matplotlib.pyplot as plt

# 1. Carga de datos
url = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.data.csv"
columns = ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness',
           'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome']
data = pd.read_csv(url, names=columns)

X = data.drop('Outcome', axis=1)
y = data['Outcome']

# 2. División en entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 3. Estandarización de las características
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# 4. Entrenamiento Random Forest
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train_scaled, y_train)

# 5. Predicción para una instancia del test
sample_idx = 0
sample = X_test_scaled[sample_idx].reshape(1, -1)
pred_proba = model.predict_proba(sample)[0]
print("Paciente seleccionado (valores estandarizados):")
print(X_test.iloc[sample_idx])
print("\nPredicción del modelo:")
print(f"Probabilidad de no tener diabetes: {pred_proba[0]*100:.2f}%")
print(f"Probabilidad de tener diabetes: {pred_proba[1]*100:.2f}%")
```

```
# 6. Explicación local con LIME
explainer = lime.lime_tabular.LimeTabularExplainer(
    X_train_scaled,
    feature_names=X.columns,
    class_names=['No Diabetes', 'Diabetes'],
    discretize_continuous=True,
    random_state=42
)

lime_exp = explainer.explain_instance(
    sample.flatten(),
    model.predict_proba,
    num_features=5 # 5 características más influyentes
)

# 7. Visualización
lime_exp.show_in_notebook(show_table=True, show_all=False)

fig = lime_exp.as_pyplot_figure() #Figura 10: Explicación local con LIME
plt.show()
```

## Apéndice B. Descripción detallada de los datasets

Se consideran dos conjuntos de datos relacionados con la diabetes:

### **Dataset de regresión (scikit-learn Diabetes Dataset)**

El conjunto de datos contiene 442 registros con 10 variables predictoras normalizadas:

- age (edad), sex (sexo), bmi (índice de masa corporal), bp (presión sanguínea), y s1-s6 (que son indicadores bioquímicos relacionados con colesterol y la glucosa)
- La variable objetivo representa una medida cuantitativa de la progresión de la enfermedad
- Las variables se centran alrededor de cero y presentan desviaciones estándar similares. Este dataset se utiliza para tareas de regresión y permite evaluar la interpretabilidad global de modelos mediante SHAP.

### **Dataset de clasificación (Pima Indians Diabetes Dataset)**

El conjunto de datos incluye 768 registros con 8 variables clínicas:

- Pregnancies (número de embarazos), Glucose (glucosa), BloodPressure (presión sanguínea), SkinThickness (grosor de pliegue cutáneo), Insulin (niveles de insulina), BMI, DiabetesPedigreeFunction (historial familiar de diabetes) y Age (edad)
- La variable objetivo Outcome indica si la persona presenta diabetes (1) o no (0)
- En este dataset hay rangos amplios en glucosa, insulina y BMI, lo que refleja la variabilidad biológica entre los pacientes. Este dataset permite realizar tareas de clasificación y analizar la interpretabilidad local y global mediante SHAP y LIME.

Dataset	Número de muestras	Número de variables	Variables
Diabetes regresión	442	10	age, sex, bmi, bp, s1, s2, s3, s4, s5, s6
Diabetes clasificación	768	8	Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, Age

## Apéndice C. Glosario de términos

### Técnicas post-hoc<sup>1</sup>

Métodos de interpretabilidad que se aplican después de que un modelo de aprendizaje automático ha sido entrenado, sin modificar su arquitectura interna ni el proceso de aprendizaje. Estas técnicas analizan la relación entre las entradas y las salidas del modelo para generar explicaciones comprensibles para el usuario.

### Técnicas agnósticas al modelo<sup>2</sup>

Métodos de explicación que pueden aplicarse a cualquier tipo de modelo de aprendizaje automático, independientemente de su estructura interna o del algoritmo utilizado. Estas técnicas tratan al modelo como una caja negra, basándose únicamente en las predicciones que produce ante diferentes entradas.

### Caja negra<sup>3</sup>

En el contexto del aprendizaje automático, un modelo se considera una caja negra cuando su funcionamiento interno no es interpretable o no es accesible para el usuario. En estos casos, solo se observan las entradas y las salidas del modelo, sin conocer cómo se transforman internamente los datos para producir una predicción.

### Datos tabulares<sup>4</sup>

Los datos tabulares son datos estructurados organizados en forma de tabla, compuesta por filas y columnas, donde cada fila representa una observación o instancia y cada columna corresponde a una variable o atributo. Este formato es característico de hojas de cálculo y bases de datos relacionales, y es ampliamente utilizado en tareas de análisis estadístico y aprendizaje automático debido a la claridad con la que se definen las características y sus valores.

### Kernel SHAP<sup>5</sup>

Kernel SHAP es una variante de SHAP agnóstica al modelo que aproxima los valores de Shapley mediante un modelo lineal ponderado en el espacio de las características. Trata al modelo original como una caja negra y estima la contribución de cada variable evaluando la salida del modelo para distintas combinaciones de características, ponderadas por un núcleo (kernel) basado en la teoría de Shapley. Este enfoque permite aplicar SHAP a modelos no lineales y con más complejidad cuando no se dispone de información sobre su estructura interna.

### Deep SHAP<sup>6</sup>

Deep SHAP es una extensión de SHAP diseñada específicamente para redes neuronales profundas. Combina la teoría de valores de Shapley con técnicas de propagación de relevancia y retropropagación para calcular de forma eficiente la contribución de cada característica de entrada. Este método permite generar explicaciones coherentes a lo largo de las distintas capas del modelo, identificando cómo las características iniciales influyen en la predicción final, y es especialmente utilizado en tareas como la clasificación de imágenes y otras aplicaciones basadas en aprendizaje profundo.

## 5. Referencias

Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., Chatila, R., & Herrera, F. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58, 82–115. <https://doi.org/10.1016/j.inffus.2019.12.012>

Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. arXiv. <https://arxiv.org/abs/1702.08608>

Gunning, D. (2017). Explainable Artificial Intelligence (XAI). Defense Advanced Research Projects Agency (DARPA). <https://www.darpa.mil/program/explainable-artificial-intelligence>

Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30, 4765–4774. <https://doi.org/10.48550/arXiv.1705.07874>

MarkovML. (s. f.). LIME vs SHAP: Model explainability techniques compared. <https://www.markovml.com/blog/lime-vs-shap>

Qu4nt. (s. f.). Nombres de las variables del dataset de diabetes (scikit-learn). [https://qu4nt.github.io/scikit-learn-doc-es/auto\\_examples/feature\\_selection/plot\\_select\\_from\\_model\\_diabetes.html](https://qu4nt.github.io/scikit-learn-doc-es/auto_examples/feature_selection/plot_select_from_model_diabetes.html)

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). “Why should I trust you?”: Explaining the predictions of any classifier. En Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1135–1144). ACM. <https://doi.org/10.1145/2939672.2939778>

Ribeiro, M. T., Singh, S., & Guestrin, C. (2018). Anchors: High-precision model-agnostic explanations. En Proceedings of the 32nd AAAI Conference on Artificial Intelligence (pp. 1527–1535). AAAI Press. <https://doi.org/10.1609/aaai.v32i1.11491>

SHAP Contributors. (s. f.). Issue #750. GitHub. <https://github.com/shap/shap/issues/750>

DataCamp. (s. f.). Introduction to SHAP values: Machine learning interpretability. <https://www.datacamp.com/es/tutorial/introduction-to-shap-values-machine-learning-interpretability>

Demostraciones en Google Collab:  
[https://colab.research.google.com/drive/1y\\_f1qcWOqPpS5S7GWi0mFQuNus4G9TgQ?usp=sharing](https://colab.research.google.com/drive/1y_f1qcWOqPpS5S7GWi0mFQuNus4G9TgQ?usp=sharing)

Presentación en Canvas

[https://www.canva.com/design/DAGcSM4t1BQ/NT-i6mVZEuu2\\_xTfRE8UwA/edit?utm\\_content=DAGcSM4t1BQ&utm\\_campaign=designshare&utm\\_medium=link2&utm\\_source=sharebutton](https://www.canva.com/design/DAGcSM4t1BQ/NT-i6mVZEuu2_xTfRE8UwA/edit?utm_content=DAGcSM4t1BQ&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton)