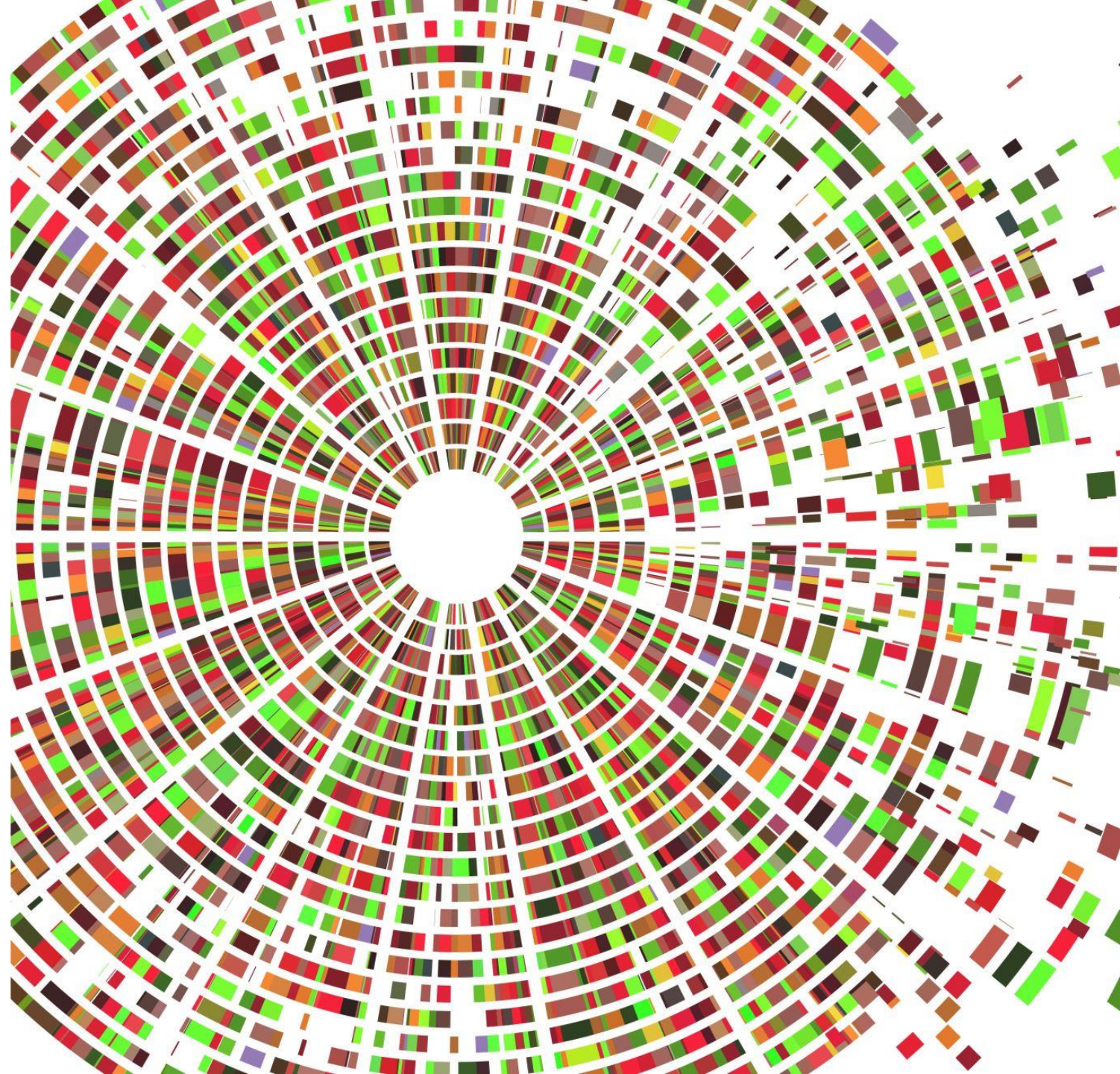


Métodos avanzados de ciencia de datos

Prof. Emily Díaz



Contenido



Visualizaciones e
interpretabilidad

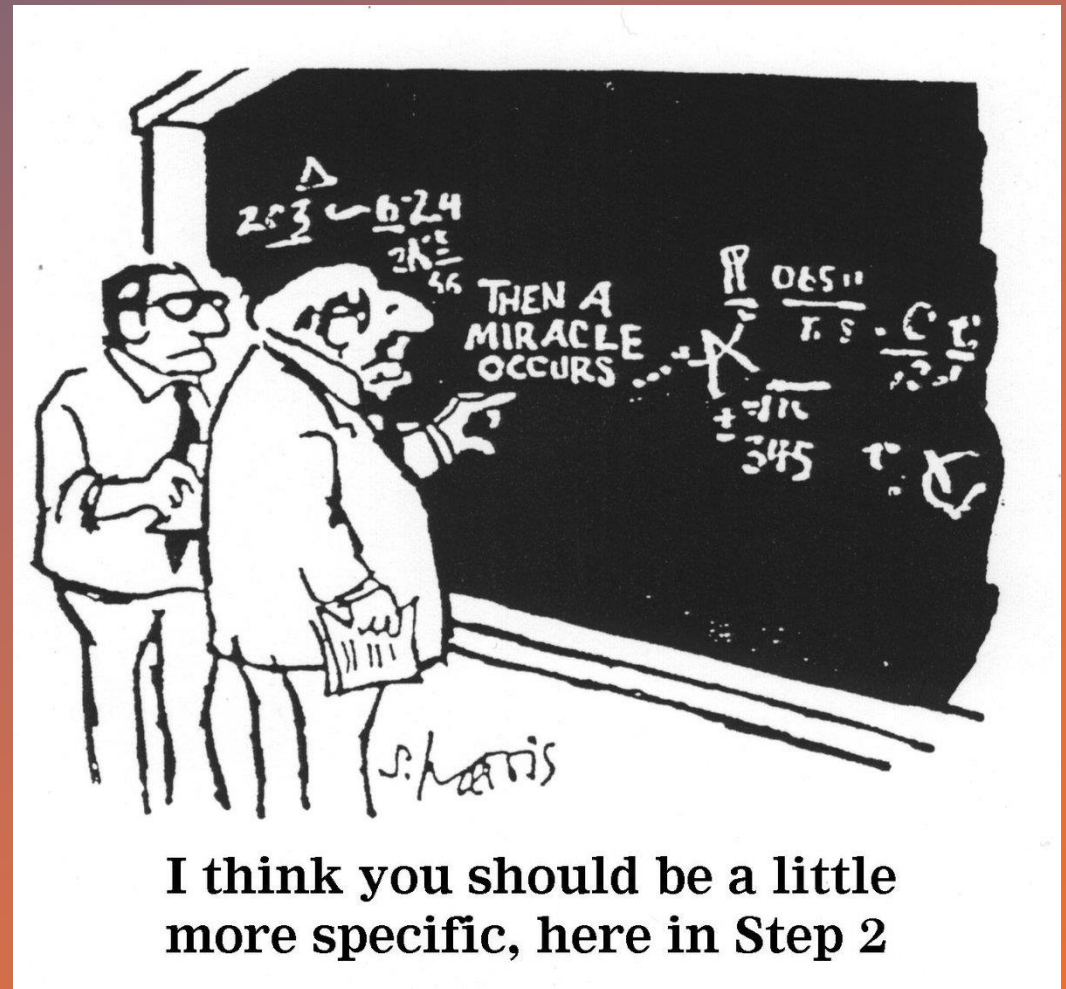


Paralelización y poder
computacional



Modelos pre-entrenados

Interpretabilidad de redes neuronales



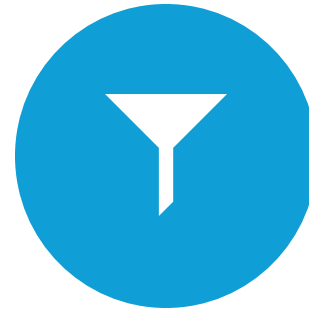
Interpretabilidad de redes neuronales



ESTOS MODELOS SUELEN CONSIDERARSE "**CAJAS NEGRAS**" PORQUE PUEDEN HACER PREDICCIONES MUY PRECISAS, PERO SON DIFÍCILES DE ENTENDER



LAS TÉCNICAS DE INTERPRETABILIDAD TIENEN COMO OBJETIVO HACER QUE ESTOS MODELOS SEAN MÁS TRANSPARENTES AL REVELAR CÓMO TOMAN DECISIONES



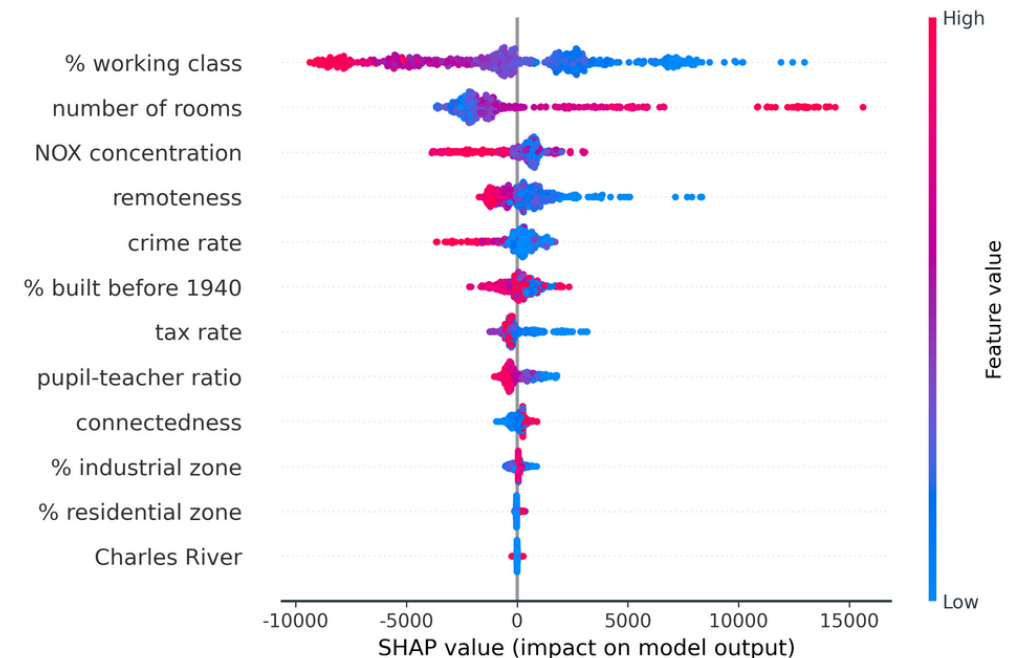
EXISTEN DIVERSOS MÉTODOS Y NO TODOS LLEGAN A LA MISMA CONCLUSIÓN POR CÓMO SON CALCULADOS. POR EJEMPLO: LIME, SHAP, MAPAS DE SALIENCIA



POR QUÉ SON IMPORTANTES? CONFIANZA EN EL MODELO, CHEQUEO DE ERRORES, REGULACIONES EN ALGUNOS SECTORES

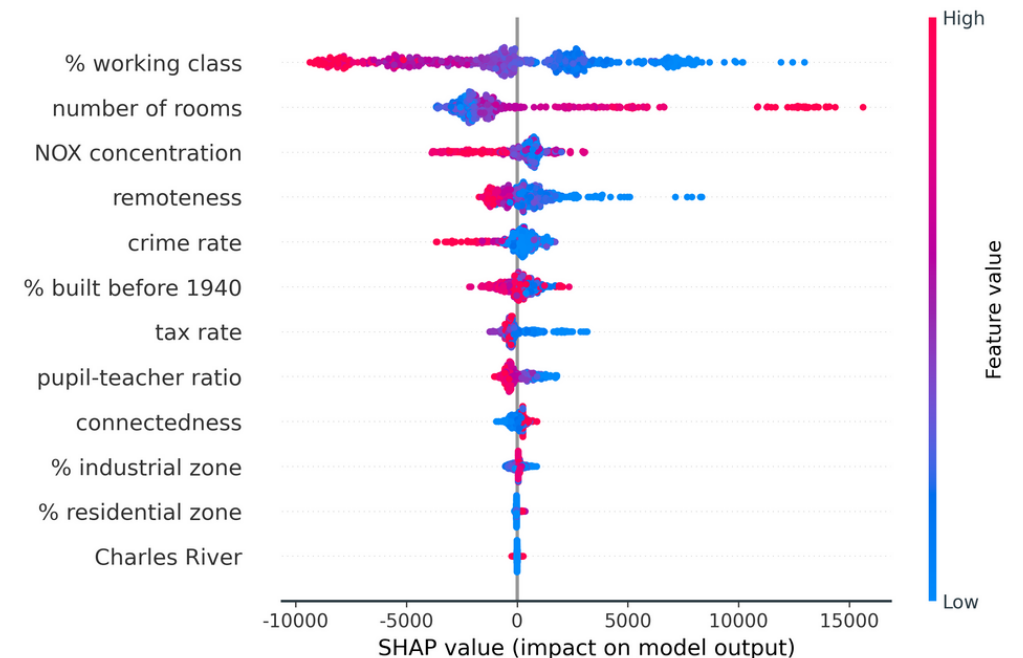
SHAP (SHapley Additive exPlanations)

- No solo para Deep Learning, puede utilizarse para la mayoría de modelos de Machine Learning
- Utiliza un enfoque de teoría de juegos que mide la contribución de cada jugador al resultado final (en nuestro caso, las variables)
- Propiedades de los valores SHAP:
 - **Aditividad:** La contribución de cada característica a la predicción final se puede calcular independientemente y luego sumar
 - **Precisión local:** Proporciona una interpretación precisa y local de la predicción del modelo para una entrada determinada (pero también da interpretación global)
 - **Robusto hacia valores perdidos:** Los valores SHAP son cero para las características faltantes o irrelevantes para una predicción.
 - **Consistencia:** Los valores SHAP no cambian cuando cambia el modelo a menos que cambie la contribución de una característica.



SHAP (SHapley Additive exPlanations)

- El valor SHAP de una característica **representa su contribución para mover la predicción desde la predicción promedio** (la línea de base) a la predicción real para una instancia particular
- La contribución marginal de la característica i es la diferencia entre la predicción del modelo con y sin la característica i



Cómo leer este gráfico?

Gráficos que provee

- En el gráfico de dispersión, cada punto representa una observación. En el eje X tenemos el valor de esa variable y en el “y”, cuanto cambia el resultado final este valor particular.
- El color representa una variable que tiene interacción con
- Si tiene pendiente positiva, valores más altos incrementan la predicción, pendientes negativas la reducen. Relaciones no lineales tienen mayor complejidad

Gráfico de resumen

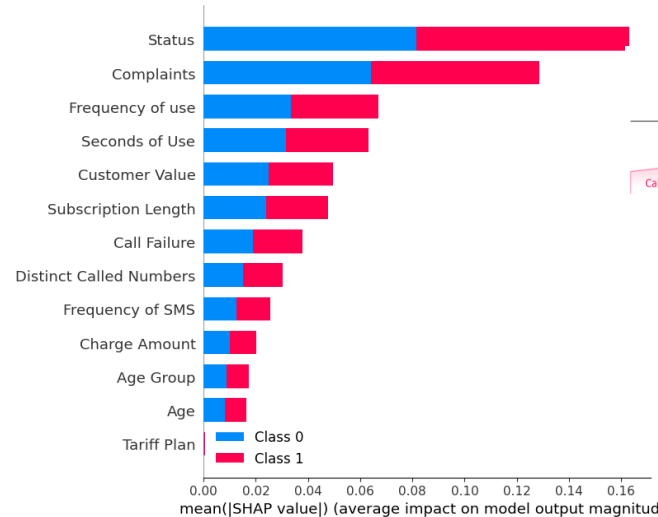


Gráfico de fuerza

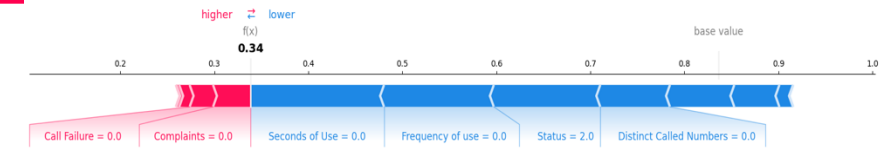


Gráfico de decisión

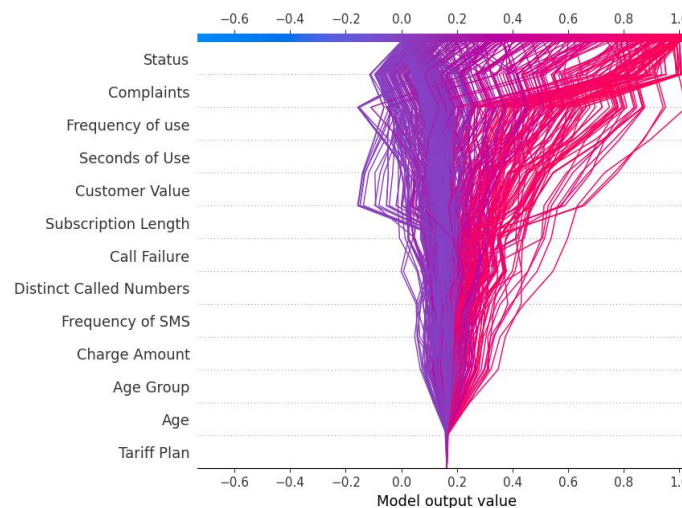
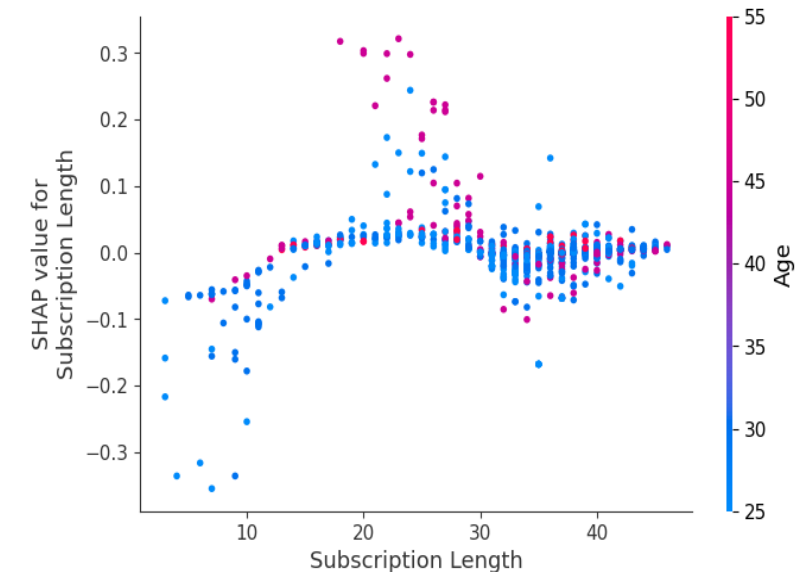


Gráfico de dependencia



LIME (Local Interpretable Model-agnostic Explanations)

- LIME se centra en proporcionar interpretabilidad local, explicando predicciones individuales en lugar del modelo como un todo
- Funciona aproximando el comportamiento del modelo en las proximidades de una instancia específica utilizando un **modelo más simple e interpretable**, normalmente un modelo lineal o un árbol de decisión.
- La idea es que, si bien el modelo original puede ser complejo y opaco, se puede aproximar localmente mediante un modelo más simple que sea más fácil de entender.

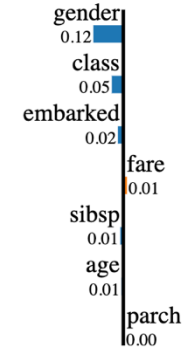
```
Intercept 0.3625304713701771  
Prediction_local [0.38617485]  
Right: 0.41
```

Prediction probabilities



died

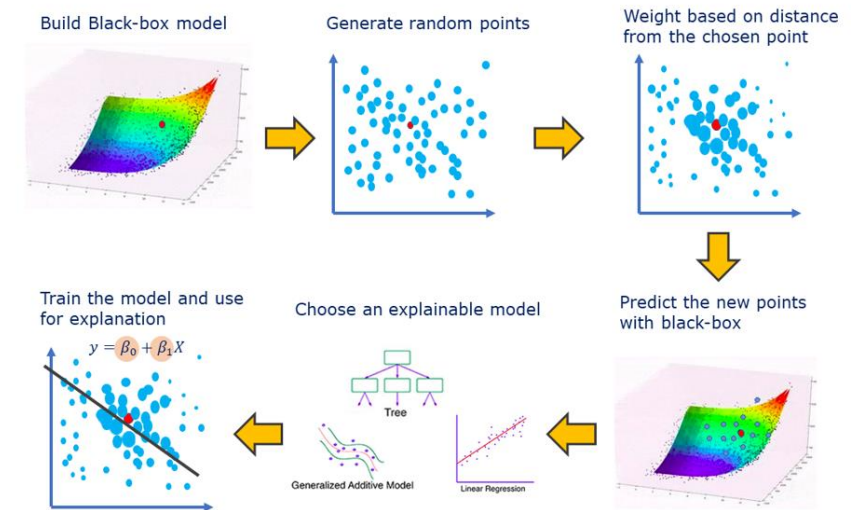
survived



Feature	Value
gender	1.00
class	0.00
embarked	1.00
fare	25.00
sibsp	0.00
age	47.00
parch	0.00

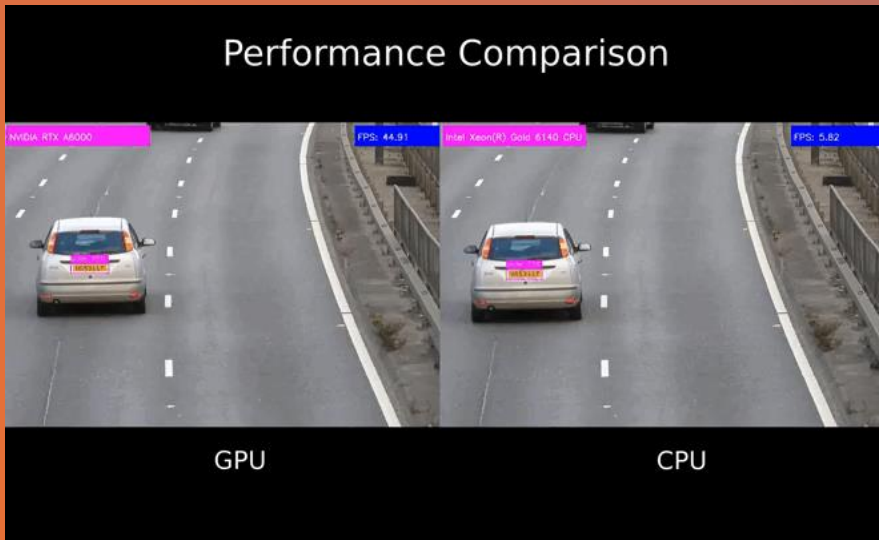
LIME – proceso interno para obtener la interpretación

- **LIME genera nuevos puntos de datos** (perturbaciones) modificando levemente la instancia original. (puede cambiar aleatoriamente algunos de los valores de las características y crear un conjunto de datos sintéticos en torno a la instancia original)
- El modelo original se utiliza para realizar **predicciones sobre estas muestras perturbadas**. Este paso genera un conjunto de predicciones correspondientes a las instancias perturbadas.
- **LIME pondera estas muestras perturbadas en función de su similitud con la instancia original**. Cuanto más similar sea una muestra perturbada a la instancia original, mayor será su peso. Esto ayuda a garantizar que la aproximación local se centre en la vecindad de la instancia que se está explicando.
- Luego ajusta un **modelo interpretable a estas muestras perturbadas** ponderadas. Este modelo interpretable aproxima el comportamiento del modelo original localmente en torno a la instancia.
- **Los coeficientes del modelo interpretable se utilizan para explicar las contribuciones** de cada característica a la predicción del modelo original para la instancia.





Paralelización y poder computacional



Qué es paralelización?



Construir una casa tiene ciertos pasos secuenciales pero las tareas pueden dividirse en partes que distintas personas pueden trabajar



Pasos para construir la casa

¿El trabajo
lo podría
hacer una
sola
persona?



**Equipos de construcción que pueden
dividirse tareas en el día a día**

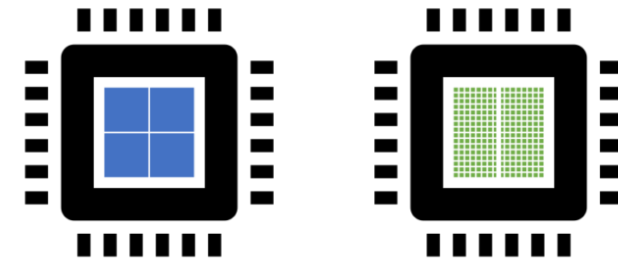
Paralelización en computación

- La misma idea de construir una casa en conjunto para ser más eficientes es la aplicada a paralelización.
- En el contexto del aprendizaje profundo, la "gran tarea" suele ser la realización de cálculos matemáticos complejos.
- En lugar de que el procesador de una computadora (CPU) se encargue de estos cálculos paso a paso, puede **dividir el trabajo entre varios procesadores para realizar varios cálculos a la vez**. Esto acelera el proceso significativamente.



GPUs

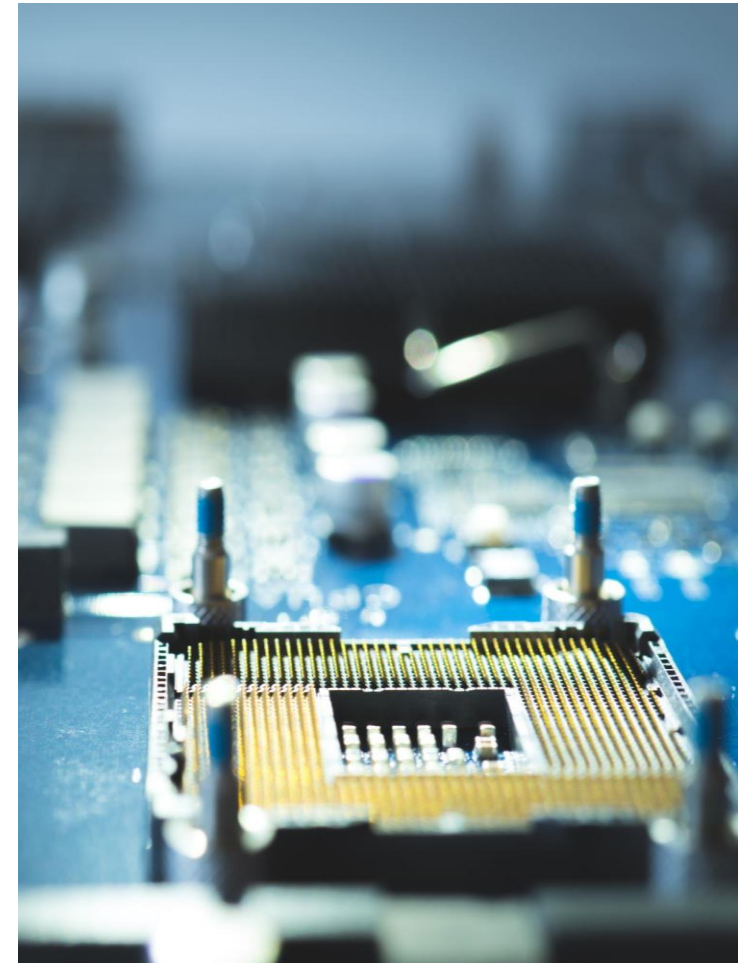
- Originalmente, las GPU se diseñaron para **manejar los gráficos de los videojuegos**, que requieren muchos cálculos rápidos para representar imágenes en la pantalla.
- Son muy buenas para **realizar muchos cálculos al mismo tiempo** (procesamiento paralelo), que es exactamente lo que se necesita en el aprendizaje profundo.



CPU	GPU
Central Processing Unit	Graphics Processing Unit
4-8 Cores	100s or 1000s of Cores
Low Latency	High Throughput
Good for Serial Processing	Good for Parallel Processing
Quickly Process Tasks That Require Interactivity	Breaks Jobs Into Separate Tasks To Process Simultaneously
Traditional Programming Are Written For CPU Sequential Execution	Requires Additional Software To Convert CPU Functions to GPU Functions for Parallel Execution

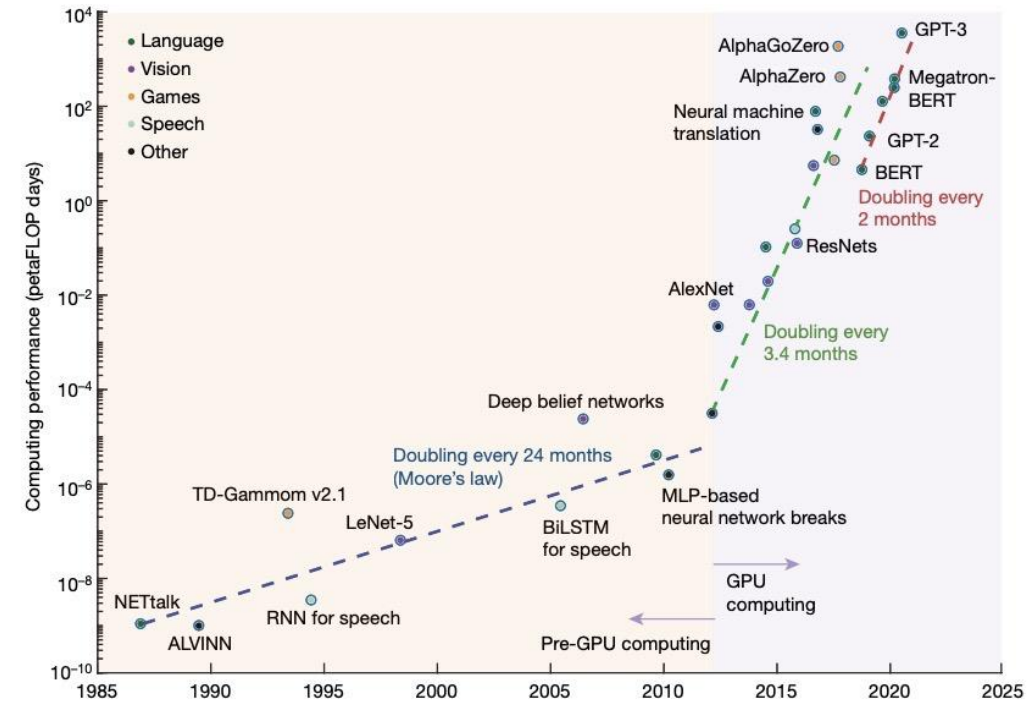
Por qué esto es necesario en Deep learning?

- El aprendizaje profundo implica entrenar modelos utilizando **grandes cantidades de datos**, lo que requiere realizar millones o incluso miles de millones de cálculos
- Hacer esto en una CPU podría llevar días, semanas o incluso más tiempo
- Al utilizar GPU, que pueden manejar muchos cálculos simultáneamente, este proceso **se puede acelerar drásticamente**
- DL implica muchas operaciones matemáticas con matrices. Estas operaciones incluyen cosas como multiplicar matrices, sumarlas y aplicar funciones a cada elemento de una matriz. **Dado que estas operaciones suelen ser independientes entre sí** (es decir, el resultado de una operación no afecta a otra), **se pueden realizar al mismo tiempo**.
- Por ejemplo, cuando se introducen datos a través de una red neuronal, **cada neurona de una capa procesa su entrada independientemente** de las demás neuronas de la misma capa.
- Durante la **retropropagación**, se calculan gradientes para cada parámetro de la red para actualizarlos. **Cada gradiente se puede calcular a menudo de forma independiente de los demás**, lo que hace que este proceso sea paralelizable. Por ejemplo, el cálculo del gradiente de un único peso en la red no depende de los gradientes de otros pesos en esa capa.
- **Si se tiene un lote de 64 imágenes**, puede calcular los pasos hacia adelante y hacia atrás de las 64 **imágenes al mismo tiempo**, en lugar de hacerlo uno por uno. Este es otro nivel de paralelización.

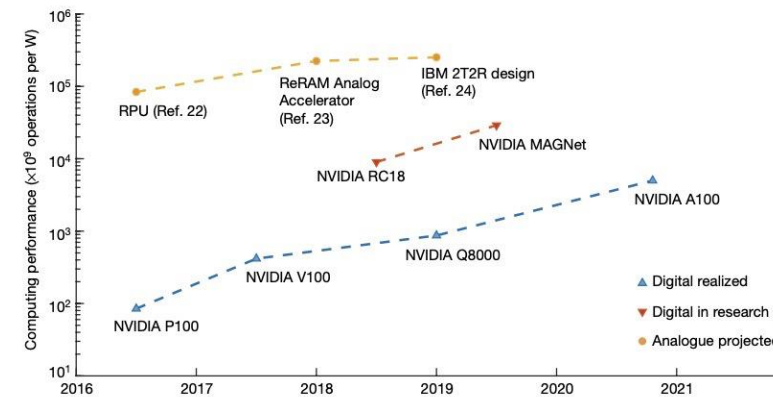


- Hasta 2012, la demanda de potencia computacional se duplicaba cada 24 meses; recientemente, esto se ha acortado a aproximadamente cada 2 meses
- Las soluciones de vanguardia han impulsado aumentos en la eficiencia computacional de más de 300 veces.

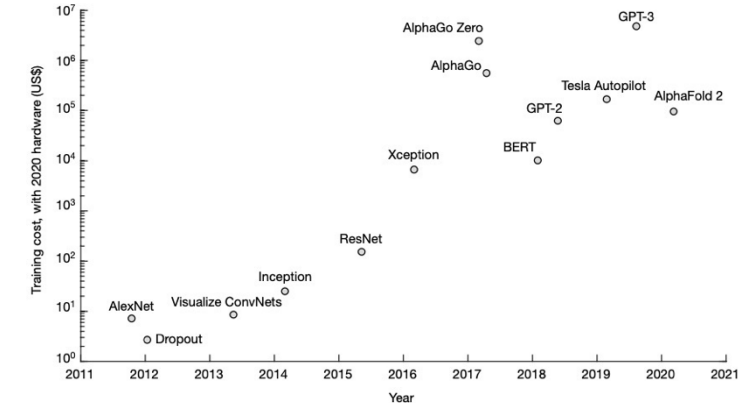
a Computing power demands



b Hardware development



c Cost






Modelos pre- entrenados



Comparación a aprender francés cuando se es hispano- hablante

- Cuando se aprende un idioma desde cero, hay mucho que considerar: **el alfabeto, el vocabulario básico, las reglas gramaticales, las estructuras de las oraciones**, etc. Este proceso requiere **mucho tiempo y esfuerzo**.
- **Si se sabe un idioma de la misma raíz/familia**, este proceso se facilita. Ya se conocen muchas palabras, la estructura general de las oraciones e incluso algunas reglas gramaticales.
- Al no empezar de cero, **se puede concentrar en aprender las diferencias** entre los dos idiomas, como el vocabulario específico y la pronunciación. **Disminuyendo el tiempo y dificultad de aprenderlo**



Castellano	Francés
fácil	facile
hijo	fils
mujer	femme
oro	or
febrero	février
pan	pain
escuela	école
piedra	pierre

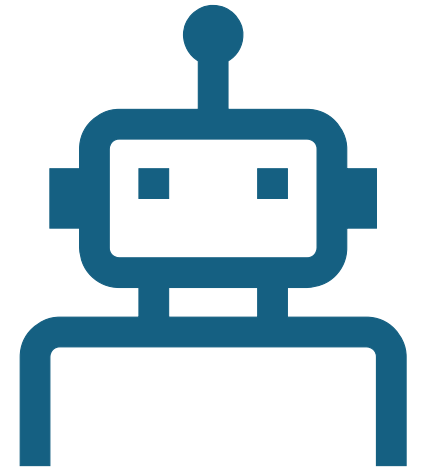
Cómo se asemeja a modelos pre-entrenados

- En Deep Learning, un modelo entrenado previamente es como saber español antes de aprender francés.
- El modelo ya ha aprendido patrones, características y estructuras de un gran conjunto de datos (como aprender español).
- Cuando tienes una nueva tarea (aprender francés), no necesitas entrenar un modelo desde cero.
- En cambio, tomas este modelo entrenado previamente y lo ajustas con datos específicos a la tarea, ahorrando tiempo y esfuerzo y logrando buenos resultados.



Qué es un modelo pre-entrenado

- Es un modelo que entrenamos idealmente en **grandes masas de datos para cierta tarea** y luego puede ser utilizado como **punto de inicio para tareas similares**
- El concepto detrás es que los **coeficientes no son inicializados de manera aleatoria**, si no que se les da un inicio mucho mejor y cercano a su valor óptimo
- Es útil en muchos casos en donde **no hay suficientes datos para tener un buen rendimiento**
- **Otra comparación del proceso es:** Comprar un vestido o una camisa y luego ajustarlo a tu tamaño ideal, en lugar de comenzar con tela, hilo y aguja.



Hay varias formas de afinar el modelo



EXTRACCIÓN DE VARIABLES

- Aprovechar el conocimiento que el modelo ya ha aprendido de un gran conjunto de datos para **extraer características útiles** de nuevos datos.
- **Ejemplo de uso:** Se pasan los nuevos datos a través del modelo previamente entrenado hasta la capa justo antes de la capa de salida. Se pasan estos resultados en una nueva capa de clasificación/regresión que se entrena para el nuevo objetivo.
- Este método es particularmente **útil cuando tienes un conjunto de datos pequeño** para tu nueva tarea porque la mayoría de los parámetros del modelo permanecen fijos, lo que **reduce el riesgo de sobreajuste**.



ENTRENAR SOLO ALGUNAS CAPAS Y DEJAR LAS OTRAS "CONGELADAS"

- Este método es un compromiso entre la extracción de características y el ajuste de todo el modelo.
- **Se congelan los pesos de las capas anteriores en el modelo** (que capturan características más generales) **y solo entrena las capas posteriores** (que capturan características más específicas).
- La idea es que las capas al inicio hayan aprendido características que son ampliamente útiles (como bordes, texturas y formas, en el caso de imágenes), mientras que **las capas finales son más especializadas y podrían necesitar más adaptación a la nueva tarea**.



ENTRENAR TODA LA ARQUITECTURA DESDE EL PUNTO DE PARTIDA DADO

- Se retoma el entrenamiento a partir del pre-entrenamiento solo que en un nuevo set de datos.
- **Al ajustar todo el modelo**, se le permite ajustar todos sus parámetros para que se adapten mejor a los nuevos datos, lo que puede generar una mayor precisión.
- Sin embargo, también **es costoso** en términos computacionales y puede generar un sobreajuste si el nuevo conjunto de datos es pequeño.





Donde encontrarlos?

Dado que son modelos que alguien más ha entrenado, debemos acceder al modelo entrenado. Algunas plataformas/librerías en donde podemos encontrarlos son:

- Huggingface (particularmente buena para NLP/texto)
- Keras/tensorflow
- Pytorch
- Github (encontrar repositorios de modelos en específico)



Veremos algunos de estos modelos cuando hablemos de...

- CNNs para imagenes
- Transformers
- Modelos con texto

Algunos ejemplos de usos

1

- Clasificando imágenes de cierto tipo de animales con categorías muy específicas
- Podemos utilizar modelos pre-entrenados con imagenet u otras que han clasificado grupos de animales más generales como punto de partida

2

- Clasificando imágenes médicas donde tenemos cientos de imágenes únicamente
- Podemos utilizar modelos pre-entrenados existentes en imágenes médicas (preferiblemente mismo tipo de modalidad y sector del cuerpo)

3

- Frente a un problema de clasificación de imágenes desbalanceadas (70% de los datos en una clase)
- Utilizar un modelo que haya enfrentado un problema similar ayuda a que se parta de un punto en donde ya se ha aprendido a discernir características importantes pese al desbalance

Consejos generales



ASEGURARSE QUE EL SET DE DATOS Y OBJETIVO ERAN SIMILARES A LO QUE SE QUIERE HACER, O AL MENOS, LO MÁS SIMILAR DE LOS MODELOS PRE-ENTRENADOS EXISTENTES.



SI EL MODELO FUE ENTRENADO CON OTRO OBJETIVO EN MENTE, CONSIDERAR SI LAS VARIABLES APRENDIDAS SERÁN DE UTILIDAD PARA EL NUEVO PROBLEMA



ENTENDER LOS REQUERIMIENTOS DE DATOS DE ENTRADA Y SALIDA (ALGUNOS MODELOS REQUIEREN IMAGENES O TEXTO PROCESADO DE CIERTA MANERA)



EVALUAR EL TAMAÑO DEL MODELO ANTE LOS RECURSOS DISPONIBLES (ESTOS MODELOS TIENDEN A SER GRANDES)



SI LA TAREA ES SIMPLE Y SE TIENEN DATOS SUFICIENTES, PUEDE QUE SEA MÁS RÁPIDO Y EFECTIVO ENTRENAR UN MODELO DESDE CERO QUE AJUSTAR UN MODELO PREVIAMENTE ENTRENADO



Preguntas?

