Apuntes semana 6 - 27/03/2025

Pamela González López 2019390545

***Abstract*— Estos apuntes presentan algunas noticias relevantes en inteligencia artificial, como robots que imitan movimientos humanos y mejoras en la creación de imágenes con IA. También explica cómo funciona la regresión logística dando un repaso de los temas abordados la clase anterior. Además se explica el notebook de regresión lineal, se habla de las diferentes métricas utilizadas en IA y del ROC y AUC.**

I. Noticias de la semana sobre IA

Noticias relevantes de la semana sobre inteligencia artificial:

- **Newton:** Nvidia y Disney lanzaron Newton, un robot inspirado en Star Wars para mejorar la movilidad y precisión de robots humanoides.
- **Otra noticia sobre robots:** Ahora los robots caminan mucho mejor, simulando con mayor precisión el movimiento, incluido el balanceo natural de la cadera al caminar.
- **OpenAI y su generador de imagen:** Lanzaron la generación de imágenes en el modelo 4. Solo hay que darle un prompt, y el modelo crea una imagen. Ahora interpreta mejor las instrucciones y soluciona un problema común en modelos anteriores: la falta de nitidez en el texto generado.

  Otra mejora es que guarda el contexto de la imagen, es decir, recuerda lo que acaba de crear. Sin embargo, aún hay detalles a considerar, como generar varias imágenes para acercarse más a lo que se quiere y mejorar elementos como las manos, que a veces no quedan bien.

  Antes, OpenAI usaba el modelo DALL-E, un sistema de texto a imagen que interpretaba una descripción y generaba una imagen. En versiones anteriores de ChatGPT, la herramienta pedía al propio modelo que creara el prompt antes de generar la imagen. Ahora, el nuevo modelo es multimodal, es decir, no hay modelos separados para cada tarea, sino uno solo que puede hacerlo todo.
- **Sesgos en IA:** Se mencionó que ChatGPT no puede generar una copa de vino completamente llena. Esto se debe al sesgo en su entrenamiento, si el modelo se entrena con imágenes donde las copas de vino suelen estar a la mitad, tomará eso como referencia, ya que no tiene ejemplos de copas completamente llenas. Este es un problema importante, y hay toda un área de investigación dedicada a reducir el sesgo en los datasets para mejorar la diversidad de ejemplos.

  Otro caso relevante es el de la investigadora Timnit, quien descubrió sesgos aún más graves, especialmente sociales. Por ejemplo, en un modelo donde una persona blanca sostenía una banana, la descripción generada era correcta: "Una persona blanca sosteniendo una banana". Sin embargo, cuando una persona de color sostenía el mismo objeto, el modelo lo interpretaba como "una persona sosteniendo un arma". Estos sesgos son un problema serio y pueden tener un impacto negativo en la sociedad.

Al final del curso veremos temas de ética y el impacto social de la inteligencia artificial, como el costo energético frente a sus efectos en la sociedad y los datasets con los que se entrenan los modelos.

II. Regresión logística

Repaso rápido de algunos temas:

A. Derivada de la función Sigmoid

$$\sigma'(x) = \sigma(x) \cdot (1 - \sigma(x))$$

B. Verosimilitud

Es la probabilidad de observar cada uno de los datos cambiando ciertos parámetros, los cuales son: w y b.

C. MSE vs Maximize Likelihood

Uno se usa para probabilidades (Maximize Likelihood) y el otro para valores continuos (MSE).

D. Función de costo:

Probabilidad de observar una etiqueta determinada.

$$L = \prod f_{w,b}(x_i)^{y_i}(1 - f_{w,b}(x_i))^{(1-y_i)}, i = 1...N$$

Caso $y_i = 1$

- $f_{w,b}(x_i)^{y_i}\cancel{(1 - f_{w,b}(x_i))^{1-y_i}}$
- $= f_{w,b}(x_i)^1(1 - f_{w,b}(x_i))^0$
- $= f_{w,b}(x_i)^1$

Caso $y_i = 1$
Calabaza no es naranja.

- $wx + b = 1.458$
- $f_{w,b}(x_i) = \sigma(1.458) = 0.81$
- $f_{w,b}(x_i)^1(1 - f_{w,b}(x_i))^0$
- $= f_{w,b}(x_i)^1 = \sigma(1.458)$
- $= 0.81$

La probabilidad de x_i no sea naranja es 0.81.

Caso $y_i = 0$

- $f_{w,b}(x_i)^{y_i}(1 - f_{w,b}(x_i))^{1-y_i}$
- $= f_{w,b}(x_i)^0(1 - f_{w,b}(x_i))^1$
- $= (1 - f_{w,b}(x_i))^1$

Caso $y_i = 0$
Calabaza es naranja.

- $wx + b = -1.32$
- $f_{w,b}(x_i) = \sigma(wx_i + b) = \sigma(-1.32) = 0.21$
- $f_{w,b}(x_i)^0(1 - f_{w,b}(x_i))^{1-0}$
- $= (1 - f_{w,b}(x_i))^1 = (1 - \sigma(-1.32))$
- $= (1 - 0.21) = 0.79$

La probabilidad de x_i sea naranja es 0.79.

E. Derivada - Función de costo:

$$L = \prod f_{w,b}(x_i)^{y_i}(1 - f_{w,b}(x_i))^{(1-y_i)}, i = 1...N$$

- Para calcular la verosimilitud se necesita encontrar la probabilidad de observar cada sample.
- La derivada de una multiplicación es compleja.
- Tenemos que buscar una equivalencia que no tengamos que multiplicar.
- Para poder hacer esto, podemos aplicar logaritmo.

Logaritmo:
- $Ln(a^n) = n \cdot\ Ln(a)$
- $Ln(a \cdot\ b) = Ln(a) + Ln(b)$
- $Ln(a^n \cdot\ b^n) = n \cdot\ Ln(a) + n \cdot\ Ln(b)$ -> Esta es la que usaremos.
- Estrictamente creciente.

Aplicar logaritmo a la verosimilitud:
- $L = \prod f_{w,b}(x_i)^{y_i} \cdot (1 - f_{w,b}(x_i))^{(1-y_i)}$
- $Ln(L) = \sum Ln(f_{w,b}(x_i)^{y_i}) + Ln((1 - f_{w,b}(x_i))^{(1-y_i)}$
- $Ln(L) = \sum y_i \cdot Ln(f_{w,b}(x_i)) + (1 - y_i) \cdot Ln((1 - f_{w,b}(x_i))$
- Es más fácil de computar (NaN).
- Es una función más sencilla de derivar.
- Se llama Log-Likelihood.
- Me quita errores al momento de estar computando las multiplicaciones de probabilidades.

Para minimizar maximizando:
- En regresión lineal encontramos un algoritmo para minimizar el MSE.
- Imaginemos que tenemos todo el código, y solo queremos cambiar la función.
- Se puede invertir la función multiplicándola por -1.
- $L = \frac{1}{N}\sum y_i \cdot In(f_{w,b}(x_i)) + (1 - y_i) \cdot In(1 - f_{w,b}(x_i))$
- $L = -\frac{1}{N}[\sum y_i \cdot In(f_{w,b}(x_i)) + (1 - y_i) \cdot In(1 - f_{w,b}(x_i))]$

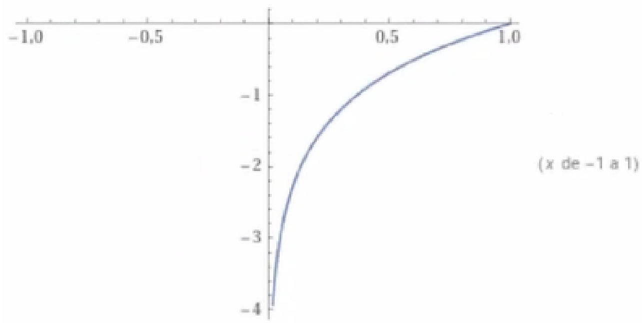


Fig. 1. Antes de darle la vuelta

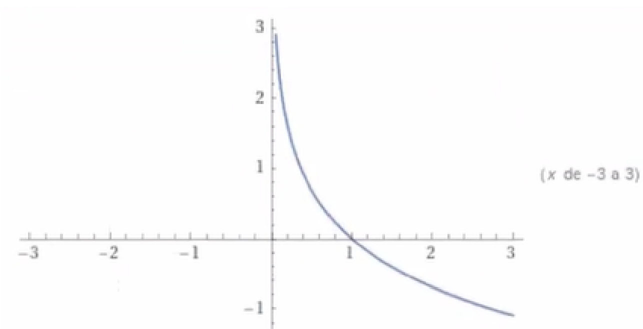


Fig. 2. Después de darle la vuelta

Después de invertir la función, ya se puede minimizar y aplicar el algoritmo de descenso del gradiente.

Lo ideal es acercarse al punto donde loss es 0. Si se tiene un loss negativo, algo estaríamos haciendo mal en los cálculos (para tenerlo en cuenta en las tareas o proyectos).

F. Actualizar parámetros w y b

Se necesita actualizar los parámetros w y b porque son los que nos permiten cambiar los resultados de las probabiidades que vamos a obtener.

- Actalizar w: $\frac{\partial L}{\partial w}$
- Actalizar b: $\frac{\partial L}{\partial b}$

Calculando las derivadas, por medio de la composición de funciones:
- $L = y_i \cdot In(f_{w,b}(x_i)) + (1 - y_i) \cdot In(1 - f_{w,b}(x_i))$ Función de costo derivada.
- $f_{w,b}(x) = a(z(x))$ -> Modelo compuesto por dos funciones.
- $a(x) = \frac{1}{1+e^{-x}}$ -> 1. F. Sigmoide.
- $z(x) = wx + b$ -> 2. Regresión lineal.

- $L = y_i \cdot In(a(z(x))) + (1 - y_i) \cdot In(1 - a(z(x)))$

Cuando se trabaja con la composición de funciones, se aplica la regla de la cadena. Es importante notar que algunos cálculos se repiten, por lo que solo se realizan una vez.

Por ejemplo, al calcular la derivada parcial de L con respecto a la sigmoide, el resultado se obtiene una única vez para ambos casos. Lo mismo ocurre con las funciones de activación. La única diferencia radica en la actualización de los parámetros w y b, ya que, en ese punto, se calcula la derivada parcial de la función lineal con respecto a w o b.

- $L = y_i \cdot \ln\left(a\left(z(x)\right)\right) + (1 - y_i) \cdot \ln\left(1 - a\left(z(x)\right)\right)$
- $\frac{\partial L}{\partial w} = \frac{\partial L}{\partial a} \cdot \frac{\partial a}{\partial z} \cdot \frac{\partial z}{\partial w}$
- $\frac{\partial L}{\partial b} = \frac{\partial L}{\partial a} \cdot \frac{\partial a}{\partial z} \cdot \frac{\partial z}{\partial b}$

Primero se calcula la derivada parcial de L con respecto a la sigmoide.

- $L = -\left[y_i \cdot \ln\left(a\left(z(x)\right)\right) + (1 - y_i) \cdot \ln\left(1 - a\left(z(x)\right)\right)\right]$
- $\frac{\partial L}{\partial a} = -\left[\left(y_i \cdot \frac{1}{a(x)} \cdot a(x)'\right) + \left((1 - y_i) \cdot \frac{1}{1-a(x)} \cdot \left(1 - a(x)\right)'\right)\right]$
- $\frac{\partial L}{\partial a} = -\left[\left(\frac{y_i}{a(x)} \cdot 1\right) + \left(\frac{(1-y_i)}{1-a(x)} \cdot -1\right)\right]$
- $\frac{\partial L}{\partial a} = -\left[\left(\frac{y_i}{a(x)}\right) - \left(\frac{(1-y_i)}{1-a(x)}\right)\right]$
- $\frac{\partial L}{\partial a} = \frac{-y_i}{a(x)} + \frac{(1-y_i)}{1-a(x)}$

Lo siguiente que se debe calcular es la derivada parcial de la función sigmoide con respecto a z. Que sería la derivada de la sigmoide.

- $a(x) = \sigma(x) = \frac{1}{1+e^{-x}}$
- $a\left(z(x)\right) = \frac{1}{1+e^{-z(x)}}$
- $\frac{\partial a}{\partial z} = \sigma\left(z(x)\right) \cdot \left(1 - \sigma\left(z(x)\right)\right)$

Por último se calcula la derivada parcial de de cada uno de los parámetros con respecto a la regresión lineal.

- $z(x) = wx + b$
- $\frac{\partial z}{\partial w} = x$
- $\frac{\partial z}{\partial b} = 1$

Luego de calcular todo por separado, se debe hacer las multiplicaciones correspondientes.

Primero, la derivada parcial de L con respecto a z.

- $\frac{\partial L}{\partial z} = \frac{\partial L}{\partial a} \cdot \frac{\partial a}{\partial z} = \frac{-y_i}{a(z(x))} + \frac{1-y_i}{\left(1-a(z(x))\right)} \cdot a\left(z(x)\right) \cdot \left(1 - a\left(z(x)\right)\right)$
- $\frac{\partial L}{\partial z} = \frac{-y_i + a(z(x)) \cdot y_i + a(z(x)) - a(z(x)) \cdot y_i}{a(z(x)) - a(z(x))^2} \cdot \left[a\left(z(x)\right) - a\left(z(x)\right)^2\right]$
- $\frac{\partial L}{\partial z} = \frac{-y_i + \cancel{a(z(x)) \cdot y_i} + a(z(x)) \cancel{- a(z(x)) \cdot y_i}}{a(z(x)) - a(z(x))^2} \cdot \left[a\left(z(x)\right) - a\left(z(x)\right)^2\right]$
- $\frac{\partial L}{\partial z} = \frac{-y_i + a(z(x))}{\cancel{a(z(x)) - a(z(x))^2}} \cdot \left[\cancel{a\left(z(x)\right) - a\left(z(x)\right)^2}\right]$
- $\frac{\partial L}{\partial z} = a\left(z(x)\right) - y_i$

Segundo la derivada parcial con respecto a w y b.

- $\frac{\partial L}{\partial w} = \frac{\partial L}{\partial z} \cdot \frac{\partial z}{\partial w} = \left(a\left(z(x)\right) - y_i\right) \cdot x$
- $\frac{\partial L}{\partial b} = \frac{\partial L}{\partial z} \cdot \frac{\partial z}{\partial b} = \left(a\left(z(x)\right) - y_i\right) \cdot 1$

Tener en cuenta que mi modelo es: $a(z(x))$.
Actaulizamos parámetros:

- $w = w - \alpha \frac{\partial L}{\partial w}$
- $b = b - \alpha \frac{\partial L}{\partial b}$
- α es un hiperparámetro, (Learnig rate).
- Gradiente descendente.

III. CÓDIGO DE REGRESIÓN LOGÍSTICA

Se hace una revisión del código de regresión logística que se encuentra en el notebook que se llama 6-RegresiónLogistica.

Se implementa la regresión logística de dos formas: utilizando la librería *scikit-learn* y programándola desde cero. Su objetivo es entrenar un modelo de clasificación binaria y evaluar su rendimiento.

Primero, se importan las librerías necesarias como *numpy*, *pandas*, *matplotlib* y *scikit-learn*, que se utilizan para manejar datos, visualizar gráficos y entrenar modelos de aprendizaje automático. Luego, se genera un conjunto de datos sintético usando *make_classification*. Este dataset tiene dos características (*Feature_1* y *Feature_2*) y se visualiza con un gráfico de dispersión para entender su distribución.

A continuación, los datos se dividen en dos partes: un 80% para entrenamiento y un 20% para prueba, usando la función *train_test_split*. Esta separación permite evaluar el modelo con datos no vistos después del entrenamiento.

Luego, se define la clase *LogisticRegressionAI*, que implementa la regresión logística manualmente. Dentro de esta clase, se incluyen métodos clave como la función sigmoide, que convierte las predicciones en probabilidades, y la función *binary_cross_entropy_loss*, que calcula el error del modelo. El método *fit()* ajusta los pesos (*w*) y el sesgo (*b*) mediante el descenso de gradiente, mientras que *predict()* clasifica nuevas muestras basándose en estos valores.

Finalmente, el modelo programado desde cero se entrena con 6000 epochs y una tasa de aprendizaje de 0.001. Luego, se hacen predicciones en el conjunto de prueba y se evalúa el rendimiento con la métrica de precisión *accuracy_score* y el informe de clasificación (*classification_report*), mostrando qué tan bien clasificó el modelo.

IV. MÉTRICAS

Las métricas son medidas utilizadas para indicar el rendimeinto de un modelo predictivo. Es la forma más objetiva de evaluar y comparar un modelo A del B.

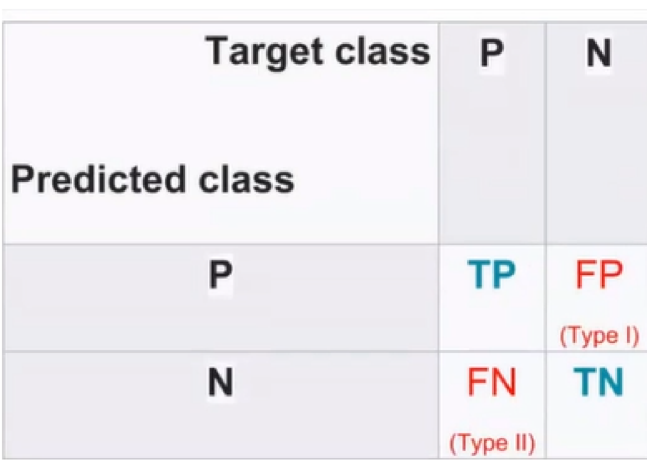
Pueden ser:

- Rendimiento.
- Horas de entrenamiento, etc.

A. Matriz de confusión

Problema de clasificación binaria donde tenemos un evento positivo (1) y un evento negativo (0).

- El modelo predice la clase P(1) y efectivamente eran la clase P(1). Se conoce como **true positive (TP)**.
- El modelo predice la clase N(0) y efectivamente eran la clase N(0). Se conoce como **true negative (TN)**.
- El modelo predice la clase P(1) cuando tenía que predecir la clase N(0). En pruebas de hipótesis se le conoce como **error tipo 1** un **falso positivo (FP)**.
- El modelo predice la clase N(0) cuando tenía que predecir la clase P(1). En pruebas de hipótesis se le conoce como **error tipo 2** un **falso negativo (FN)**.



Target class / Predicted class	P	N
P	TP	FP (Type I)
N	FN (Type II)	TN

Fig. 3. Matríz de confusión

Ejemplo: En un caso cotidiano. Imaginando que tenemos un doctor que será nuestro modelo, se pueden presentar los siguientes casos en la revisión: "Está embarazada" o "No está embarazada". Usando la matriz de confusión, se tiene ilustrado el ejemplo con la siguente imagen:

Fig. 4. Ejemplo cotidiano de la matríz de confusión

Se presentarán diversas métricas, la mayoría de las cuales se espera que estén **incluidas en el proyecto**. Estas métricas serán fundamentales para comparar los modelos y determinar cuál es el más adecuado.

B. Métrica Accuracy

- La clasificación correcta entre todos los intentos.
- $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$
- Es útil cuando los errores por clase **son igual de importantes**.
- Otorga importancia igual a todas las clases. (Clases no balanceadas).
- Puede no ser suficiente para evaluar el modelo.

C. Métrica Precision

- Mide los errores tipo I.
- La tasa de predicciones positivas correctas entre todas las predicciones positivas.
- $Precision = \frac{TP}{TP+FP}$

D. Métrica Recall (Sensivity

- Mide los errores tipo II.
- La tasa de predicciones correctas entre todos los ejemplos positivos del conjunto de datos.
- $Recall = \frac{TP}{TP+FN}$

E. Métrica F1-Score

- Métrica comúnmente usada en problemas de clasificación, especialmente cuando tenemos desequilibrio de clases.
- $F1 = \frac{2 \cdot precision \cdot recall}{precision+recall}$

Caso de estudio

- Total de pacientes: 1000
- Pacientes con cáncer (Clase Positiva): 30

- Pacientes sin cáncer (Clase Negativa): 970
- TP: 25 pacientes con cáncer correctamente identificados.
- FN: 5 pacientes con cáncer que el modelo clasificó incorrectamente como no cancerosos.
- TN: 950 pacientes sin cáncer correctamente identificados.
- FP: 20 pacientes sin cáncer que el modelo clasificó incorrectamente como con cancer.

Accuracy:

$$Accuracy = \frac{TP + TN}{TOTALPACIENTES} = \frac{25 + 950}{1000} = 0.975 \tag{1}$$

Buen modelo, si solo se calcula con esta métrica.

Recall:

$$Recall = \frac{TP}{TP + FN} = \frac{25}{25 + 5} = 0.833 \tag{2}$$

El recall del 83.3 sugiere margen de mejora, ya que sigue siendo insuficiente para diagnosticar con precisión a las personas con cáncer.

Precision:

$$Precision = \frac{TP}{TP + FP} = \frac{25}{25 + 10} = 0.55 = 55\% \tag{3}$$

La precisión del 55 es baja, lo que indica margen de mejora. Aunque hay falsos positivos, estos no representan un riesgo para los pacientes.

F1-Score:

$$F1 = \frac{1 \cdot precision \cdot recall}{precision + recall} = \frac{2 \cdot 0.55 \cdot 0.833}{0.55 + 0.833} = 0.662 \tag{4}$$

El 62.2 es bajo y refleja la dificultad del modelo para clasificar las clases minoritarias.

V. ROC y AUC

El área bajo la curva (AUC) mide la relación entre falsos positivos y falsos negativos según los ajustes del modelo.

- Si el AUC está cerca de 0.5, el modelo no tiene capacidad de clasificación.
- Un valor de 0.7 indica un desempeño aceptable.
- Entre 0.7 y 0.8 se considera razonable.
- Un valor de 1 significa una clasificación perfecta.