

## Introducción:

Para esta entrega se entrenó un modelo de redes neuronales y se probaron diferentes hyper parámetros, agregando capas ocultas y diferentes funciones de activación enfocándonos principalmente en el accuracy.

## Checkpoint 4: Redes Neuronales:

Para el entrenamiento del modelo de redes neuronales escalamos los datos para poder reducir los sesgos y también para poder mejorar la convergencia del mismo. Además, realizamos varias pruebas con diferentes cantidades de capas ocultas con ReLU como función de activación y para la capa de salida optamos por la función Sigmoidea ya que nos devuelve un valor entre 0 y 1 lo cual es conveniente para nuestra predicción en el contexto dado.

### Funciones de activación:

Para las capas de entrada se probó utilizar Leaky\_ReLU y ReLU. Pero se optó por solo utilizar ReLU, ya que daba la mejor métrica.

En las capas ocultas se decidió utilizar, ReLU, ELU, SeLU, tanh. Se vio una pequeña mejora al sumar capas con las distintas funciones, pero no fueron casi nulas.

Como solo se debía predecir 0 o 1 se decidió utilizar la función sigmoideal en la capa de salida, ya que esta devuelve valores entre 0 y 1.

A pesar de que SoftMax sea utilizado en las capas de salida para predicciones multiclase, decidimos probarlo en las capas de entrada y en las ocultas, con la intención de ver si había mejoras en la performance y/o predicción del modelo. No se vieron mejoras, solo se mantuvieron o empeoraron.

### Métricas

Ante la ausencia de la métrica F1 Score se decidió utilizar AUC y Accuracy ya que evalúan la capacidad de clasificación global del modelo.

### Optimizadores:

Como optimizador utilizamos Adam ya que el modelo en sí no tardó mucho en entrenarse por lo que no fue necesario estar evaluando otros

### Métodos de regularización:

Tomamos uno de los modelos entrenados (modelo 1 del colab) y se aplicaron los siguientes métodos de regularización: L1, L2, Dropout y Early stopping. En general Las regularizaciones Dropout y early stopping mantuvieron casi exactamente las mismas métricas (diferencia en la cantidad de aciertos +-15 observaciones), por otra parte L2 disminuyó un la performance del modelo (2-3%) y L1 disminuye la performance enormemente (30-40% menos, solo dio predicciones positivas)

## Conclusión

Como conclusión, fuimos agregando muchas neuronas con una sola capa oculta que si bien era más performante, no cambiaba mucho las métricas. Cuando comenzamos a agregar más capas, esto impactó en la performance del modelo (fue más lento) pero comenzamos a ver mejoras significativas en las métricas.