

## Checkpoint 4 - Grupo 12

### Introducción

Para este checkpoint se buscó crear arquitecturas distintas de redes neuronales especializadas en problemas de clasificación, para luego hacer la búsqueda de los hiperparámetros que optimizaban la performance de las respectivas arquitecturas y la búsqueda de los hiperparámetros que optimizaban la performance de los entrenamientos de los mejores modelos. Previo a estas búsquedas, se realizaron transformaciones sobre el dataset de entrenamiento, que incluyó además técnicas como la normalización de datos. Luego, vamos a comparar los mejores modelos obtenidos desde cada arquitectura, para ver cuál modelo performó mejor.

### Construcción del modelo

- ¿Cuál fue la arquitectura escogida?
  - Capa entrada
    - Neuronas de entrada: 128
    - Función de activación: ReLu
    - Tipo de conexión: fully\_connected (todas las neuronas de una capa se conectan con todas las de la siguiente capa)
  - Capas ocultas:
    - Neuronas por capa: 128
    - Función de activación: ReLu
    - Tipo de conexión: fully\_connected (todas las neuronas de una capa se conectan con todas las de la siguiente capa)
    - Cantidad: 2
  - Capa de salida
    - cantidad de neuronas: 1
    - función de activación: sigmoide
- ¿Qué hiperparámetros se optimizaron? Se optimizaron hiperparámetros tanto de arquitectura (neurons, hidden\_layers, optimizer) como de entrenamiento (epochs, batch\_size)
- ¿Qué optimizador se utilizó? Se utilizó SGD (Descenso por gradiente estocástico)
- ¿Se utilizó alguna técnica de regularización? ¿Cuál? No se utilizó una técnica de regularización por cómo definimos a la arquitectura.
- ¿Cuántos ciclos de entrenamiento utilizó? Se usaron 10 epochs con batch\_size de 20

Primero lo que se hizo fue definir la arquitectura base de la red neuronal cual “constructor” que recibe parámetros y que puede construir una red neuronal distinta pero de misma base. Luego se definió un rango de valores para los hiperparámetros que recibía ese constructor y se hizo la búsqueda de los mejores valores de los hiperparámetros de mi arquitectura tal que minimice la función de pérdida de ‘binary\_crossentropy’. Y una vez encontrado el mejor modelo de dicha arquitectura, hicimos una búsqueda de los mejores valores de los hiperparámetros de entrenamiento (epoch y batch\_size) tal que hagan que el mejor modelo performe aún mejor.

### Cuadro de Resultados

Modelo	F1-Test	Precision Test	Recall Test	Accuracy	Kaggle
modelo_1	0.80	0.76	0.84	0.79	0.72893
modelo_2	0.78	0.74	0.83	0.77	0.76431
modelo_3	0.77	0.73	0.81	0.77	0.73961
modelo_4	0.78	0.73	0.83	0.76	0.73366

**Nota: indicar brevemente en qué consiste cada modelo de la tabla y detallar el caso del mejor modelo.**

modelo\_1: Utiliza el optimizador adam, compuesto por 128 neuronas y 2 capas ocultas. Tiene como base la arquitectura 1

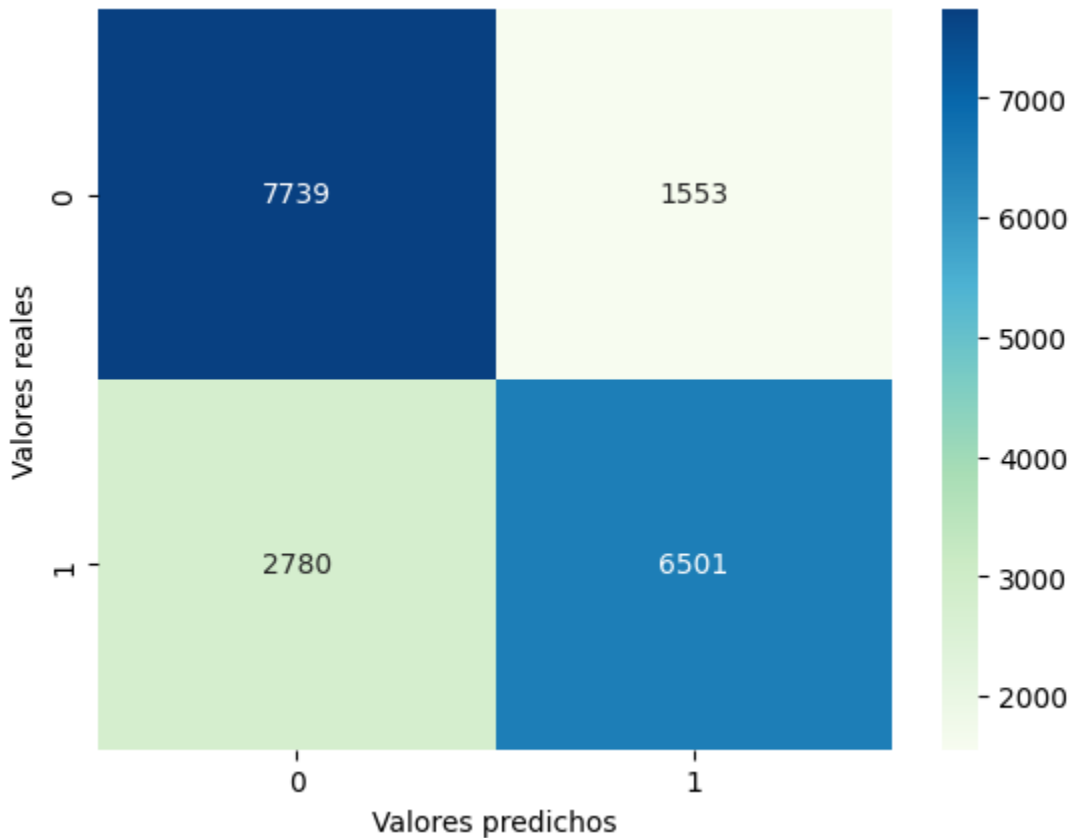
modelo\_2: (Mejor predictor en kaggle) Utiliza el optimizador sgd,, compuesto por 128 neuronas y 2 hidden\_layers, y en la mejora del entrenamiento se usaron 10 epochs y un batch\_size de 20. Tiene como base la arquitectura 2

modelo\_3: Utiliza el optimizador sgd, neurons = 128, hidden\_layer=1 (capas ocultas) y se utilizaron las técnicas de regularización L2 con un learning rate de 0.01 en la capa de entrada y de salida, y Dropout con un valor de 0.2 en las capas ocultas. En entrenamiento se utilizaron los hiperparámetros: epochs = 30, batch\_size = 25. Tiene como base la arquitectura 3.

modelo\_4: Utiliza el optimizador rmsprop, con regulador L2 y un learning\_rate=0.01 en la capa

de entrada. Las capas ocultas utilizan el regulador L2 y Dropout. Se utilizaron los hiperparámetros epoch = 2 , batch\_size = 10, neurons = 64 y hidden\_layers = 2. Tiene como base la arquitectura 4.

## Matriz de Confusion



Podemos observar que el modelo tiene un alto nivel de predicción, también podemos ver que es mejor predictor para valores no cancelados que para valores cancelados.

## Tareas Realizadas

Integrante	Tarea
Gaston Sabaj	Creación de las arquitecturas, optimización de hiperparámetros de arquitecturas y de la fase de entrenamiento.
Juan Yago Pimenta	Construcción de red neuronal base. Optimización de hiperparámetros.