

# Informe Checkpoint 4

- **Primera Red Neuronal**

Primero hicimos la importación de datos y el One hot encoding para las variables categóricas. Después hicimos un split del dataset donde el 25% de los datos los tomamos como test y el resto como train y utilizando keras.Sequential hicimos un primer modelo de Red neuronal usando 2 capas, una 1852 neuronas con una función de activación relu y otra con una neurona con una funcion de activacion sigmoid. Hicimos un compile de ese modelo usando Adam como función de optimización y la entropía cruzada binaria como función de pérdida. El modelo fue entrenado con 525 épocas y en 410 lotes.

- **Segunda Red Neuronal**

Para encontrar una mejor arquitectura para nuestra red neuronal lo que hicimos fue hacer un balance de los datos de entrenamiento con un sample de la clase mayoritaria, de tal forma que nos de un mayor recall a la hora de construir el modelo. Construimos otra red neuronal con esos datos balanceados con 4 capas: con 3 usamos la activación de relu (una con 94 otra con 88 y otra con 79 neuronas), una capa de activación sigmoid con una neurona. Para este modelo también usamos un dropout de 0.36. El compile lo hicimos con un optimizador Adam, esta vez con un learning rate de 0.001, y con la misma función de pérdida que la anterior. Para el entrenamiento usamos 257 épocas y 117 lotes. Usamos también un class\_weight para que el peso de los valores del target sean iguales.

- **Utilización del Grid Search**

Finalmente, con el segundo modelo hicimos un Grind Search donde variamos los parámetros de épocas (entre 100, 200 y 250) y de lotes (entre 50, 100 y 120). Las mejores hiperparametros fueron con epocas = 250 y con lotes = 100. El score utilizando esos hiperparametros era de 0.8123031377792358. Entre la Red Neuronal con los hiperparametros y la segunda Red Neuronal que hicimos, nos dio un mejor predict en kaggle esta ya que tiene un precision y acurracy superior a pesar de que el f1\_score haya sido menor.