Checkpoint 4 - Grupo 39

Introducción

Primero pre procesamos los datos haciendo la correspondiente normalización y escalado de los mismos, además utilizamos el dataset del checkpoint 3 que utilizaba one-hot endcoding para transformar las variables cualitativas a numéricas.
Luego comenzamos a crear varios modelos de redes neuronales probando distintas arquitecturas modificando la cantidad de capas y neuronas por capa. Elegimos no variar la función de activación para poder evaluar aisladamente la arquitectura del modelo.
Finalmente, elegimos el mejor modelo basándonos en las métricas obtenidas y optimizamos sus hiperparametros.

Construcción del modelo

- Arquitectura
  - Capa entrada
    - 1 neurona con un input shape del tamaño de las columnas de nuestro dataset (213)
    - Función de activación: sigmoid
    - Tipo de conexión: dense
  - Capa oculta
    - 300 neuronas
    - Función de activación: sigmoid
    - Tipo de conexión: dropout
  - Capa de salida
    - 1 neurona
    - Función de activación: sigmoid
- Optimizamos los siguientes hiperparámentros:
  - Funciones de activación: sigmoid
  - Optimizador: Nadam
  - Función de loss: probamos con hinge y binary_crossentropy y decidimos utilizar binary_crossentropy ya que es la recomendada para problemas de clasificación binaria y no vimos diferencias significativas entre ambas
  - Epochs: 40, ya que a partir de ese número el cambio en la métrica AUC y Loss era mínimo

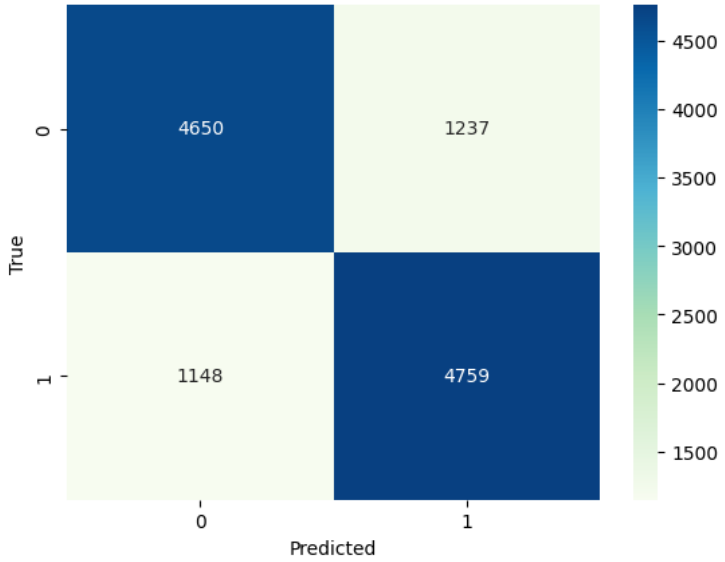
- Optimizador: utilizamos Nadam para reducir el tiempo de entrenamiento ya que es un poco más rápido que Adam
- Regularización: utilizamos Dropout para evitar el overfitting de la red ya que se encarga de apagar alguna neuronas aleatoriamente para obligar a la red a construir nuevos caminos
- Ciclos de entrenamiento: utilizamos 40 epochs

Cuadro de Resultados

Modelo	F1-Test	Precision Test	Recall Test	Accuracy	Kaggle
modelo_1	0.800	0.795	0.800	0.800	0.80226
modelo_2	0.795	0.800	0.795	0.800	0.79542
modelo_3	0.795	0.795	0.800	0.800	0.79773
Modelo_4	0.800	0.800	0.800	0.800	0.79636

El modelo que seleccionamos como mejor fue el modelo 1 que detallamos en el punto anterior
Modelo 2: Cantidad de neuronas de la capa oculta igual a cantidad de columnas de nuestro dataset
Modelo 3: 2 capas ocultas de 500 neuronas. A pesar de ser el modelo con más neuronas no muestra mejoras significativas
Modelo 4: es el modelo 1 con los parámetros optimizados. Elegimos relu como función de activación ya que las métricas mejoraron, sin embargo la predicción en Kaggle fue peor por lo que lo descartamos como mejor modelo

Matriz de Confusion



Vemos que predijo correctamente 9404 de 11794 reservas. Y predice ligeramente mejor las reservas no canceladas al tener menor cantidad de errores cometidos.

Conclusiones

Cuadro comparativo del mejor modelo de cada checkpoint

CHPN	F1-Test	Precision Test	Recall Test	Accuracy	Kaggle	Modelo
2	0.7899	0.6844	0.9337	0.7502	0.7957	Arbol de decision
3	0.890	0.890	0.895	0.890	0.866	Voting
4	0.800	0.795	0.800	0.800	0.80226	Redes neuronales

Podemos observar que el mejor modelo resultó ser un ensamble, en particular el de Voting el cual está compuesto por 3 modelos base: Random Forest, SVM (Radial) y XG Boost. Destacamos que el modelo ganador se basa en votos con ninguna

ponderación lo cual parece simple pero termina siendo superior a modelos más complejos como por ejemplo la red neuronal o Stacking que contiene un meta-modelo que aumenta el costo computacional.
Las opciones que nos quedaron por explorar fueron probar distintas combinaciones de modelos base en Voting y modificar el hiperparámetro de votación ya que elegimos únicamente “hard”.

Tareas Realizadas

Integrante	Tarea
Agustín Ezequiel Sánchez Decouflet	Preprocesamiento Armado de reporte Arquitectura del modelo Métricas
Franco Darío Mazzaro	Preprocesamiento Armado de reporte Arquitectura del modelo Métricas Gráficos
Ariadna Antonella Cattaneo	Armado de reporte Arquitectura del modelo Optimizado de hiperparámetros Métricas