Organización de Datos (75.06)
Trabajo Práctico 1: Reservas de Hotel. Checkpoint 4
Cuatrimestre, año y grupo: 2ºC 2023
Grupo: 11 - Los Pandas
Corrector: Mateo Suster
Fecha de entrega: 3 de noviembre de 2023
Datos de los integrantes:

Labollita, Francisco	106861	flabollita@fi.uba.ar
Mundani Vegega, Ezequiel	102312	emundani@fi.uba.ar
Otegui, Matías Iñaki	97263	motegui@fi.uba.ar

Introducción

En este checkpoint se implementaron redes neuronales secuenciales para predecir el dataset de hoteles. Se probaron arquitecturas sin capa oculta, con una capa oculta y con dos capas ocultas, cada una de estas con diferentes cantidades de neuronas. La conexión entre la distintas capas fue del tipo *dense*, es decir todas las de una capa con todas las de la siguiente. Además se optimizaron sus hiperparámetros haciendo un *gridsearch* y validación cruzada. La mejor arquitectura fue optimizada nuevamente con una búsqueda más exhaustiva.

Las modificaciones realizadas sobre el dataset son las mismas que se hicieron para entrenar los ensambles voting y stacking pero con una modificación. Se creó una columna que dice si la habitación pedida fue la asignada, se eliminaron las filas con datos faltantes en *country*, *market_segment* y *distribution_channel*. Se eliminaron las columnas consideradas irrelevantes: *arrival_date_day_of_month*, *stays_in_weeknd_nights*, *stays_in_week_days*, *children*, *babies* , *reserved_room_type*, *deposit_type*, *company*, *adr* e *id*. También se eliminaron los valores considerados outliers de la columna *adults*. Se hizo one-hot encoding de las columnas categóricas y se normalizaron las columnas cuantitativas numéricas.

A diferencia de lo que se hizo en voting y stacking, se decidio por no eliminar la columna *arrival_date_year* y transformarla en categórica, se verificó que agregaba información útil (y hacerle el one-hot encoding correspondiente).

Construcción de los modelos

Se entrenaron varias redes neuronales, una sin capa oculta, otra con una capa oculta y otra con dos capas ocultas. Se optimizaron los hiperparámetros de cada una y luego se realizó una búsqueda de hiperparámetros más exhaustiva teniendo en cuenta el mejor modelo obtenido.

Cabe aclarar que no se buscaron diferentes optimizaciones dependiendo del optimizador dado que no tenía sentido, ya que se supone que los más avanzados alcanzan todos el mismo resultado, solo que a distinta velocidad. No tendría sentido ver cuál llega más rápido, así que se utilizó uno directamente. Se utilizó Nadam.

Para la última capa de la red neuronal habrá una única neurona con función de activación sigmoide, ya que se busca una clasificación binaria. También cabe aclarar que entre las distintas capas hay una conexión "densa".

El modelo que tuvo mejor rendimiento luego de una optimización de hiperparámetros sencilla fue el de una capa oculta. Luego se optimizó más rigurosamente solo el de una capa oculta, primero utilizando siempre la función de activación ReLu; luego cambiando solamente dicha función. Si bien sería mejor hacer la optimización con todas las funciones de activación directamente, se optó por hacerlo con una sola para no tener que hacer tantas iteraciones (89 iteraciones en vez de 567).

Se optimizaron los siguientes hiperparámetros, obteniéndose los resultados numéricos que se presentan a continuación:

- Cantidad de neuronas: 24
- Épocas: 25
- Función de activación: ReLu
- Tasa de aprendizaje: $10^{-2,5} \simeq 0,00316$
- Tamaño de lote: 32

Cuadro de resultados para el conjunto test

Modelo	F1 Score	Precision	Recall	Accuracy	Kaggle
Red neuronal sin capa escondida	0,848	0,861	0,836	0,850	-
Red neuronal con una capa escondida	0,852	0,868	0,825	0,849	-
Red neuronal con dos capas escondidas	0,850	0,867	0,831	0,852	-
Red neuronal optimizada	**0,855**	**0,823**	**0,890**	**0,849**	**0,838**

Cuadro 1: Tabla de resultados

En la tabla se pueden comparar los resultados de todos las redes neuronales utilizadas. Todas las redes neuronales tienen la capa de salida igual: contiene una neurona y utiliza una función de activación sigmoidea. Además, todas las capas tienen tipo de conexión Dense. La primera red neuronal contiene solamente la capa de entrada y la de salida. La capa de entrada contiene treinta y dos neuronas y utiliza una función de activación ReLu. Además el modelo usa el optimizador Nadam, cincuenta épocas y un tamaño de lote de treinta y dos. La segunda red neuronal contiene una capa escondida. La primeras dos capas tienen dieciséis neuronas y utilizan una función de activación ReLu. Además el modelo usa el optimizador Nadam, cincuenta épocas y un tamaño de lote de cincuenta. La tercera red neuronal contiene dos capas escondidas. La primeras tres capas tienen treinta y dos neuronas y utilizan una función de activación sigmoidea. Además el modelo usa el optimizador Nadam, cincuenta épocas y un tamaño de lote de veinticinco. Finalmente, la última red neuronal en el cuadro es una versión optimizada de la red neuronal con una capa escondida. Esta detallada con mayor detalle en el punto anterior.

Matriz de Confusión

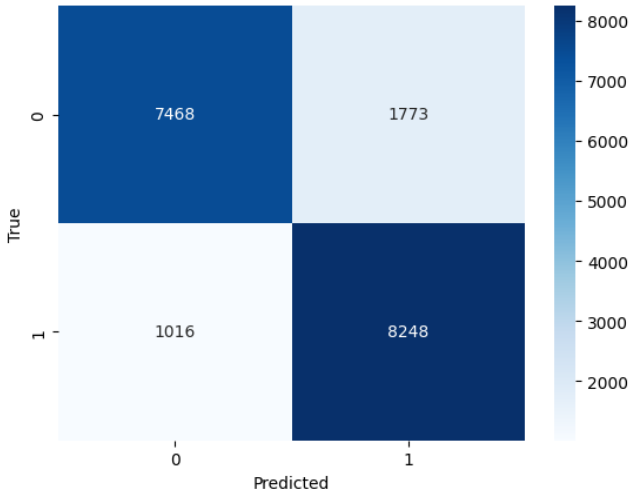

Figura 1: Matriz de confusión de la mejor red neuronal.

Comparando con la matriz de confusión del Random Forest del checkpoint anterior, se puede ver que aumentaron los verdaderos y falsos positivos mientras que bajaron los verdaderos y falsos negativos. Esto también se refleja en la métrica de Recall, que es más elevada que las otras métricas analizadas.

Distribución de tareas

Integrante	Tareas
Labollita, Francisco	Modelo, Informe
Mundani Vegega, Ezequiel	Modelo, Informe
Otegui, Matías Iñaki	Revisión, Informe