

Informe TP1: Reservas de Hotel

Integrantes:

Agama Avila, Arely - 105829
Martinez, Selene Anahi - 100439
Meichtri, Melany - 102330

Checkpoint 4:

Primero creamos modelos manualmente probando distintos parámetros, las funciones de activación: sigmoid, relu, el learning rate, que nos daban un mejor f1 score. En nuestra red base usamos el optimizador SGD.

El mejor modelo que encontramos fue utilizando un learning_rate de 0.01, y la función usada fue: sigmoid.

Luego pasamos a encontrar los mejores hiperparametros con GridSearch utilizando como base los “mejores parámetros” que ya habíamos encontrado manualmente.

Los mejores hiperparametros encontrados utilizando Grid Search para la validación cruzada fueron:

batch_size: 100

epoch: 50

Decidimos buscar otros valores para nuestros hiperparametros quedandonos:

batch_size: 75

epoch: 50

Procedimos probar con Adam y Nadam.

Nuestro mejor modelo fue obtenido utilizando:

Epoch: 100

Batch: 75

Optimizador Adam

Funcion de activacion relu

32 capas ocultas

loss: mse

Regularizacion: Early stopping

Métricas:

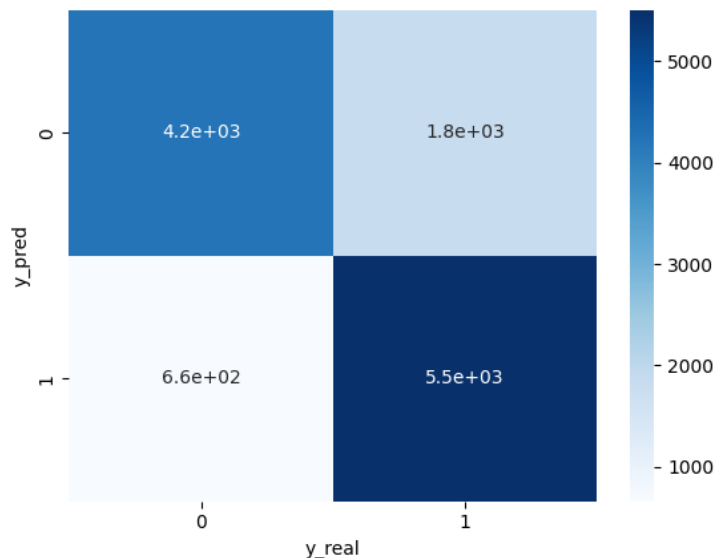
f1_score: 0.816

accuracy: 0.796

recall: 0.892

precisión: 0.751

Matriz de confusión:



Conclusiones del TP1:

Para realizar un trabajo eficiente de preprocesamiento es esencial normalizar los datos a los modelos. Dado que nos desligamos de las unidades de medida, evitando darle mayor peso a algunos datos.

Los árboles del checkpoint 2 nos permitieron encontrar cuales son los features más relevantes para nuestro análisis.

Nuestra métrica a analizar fue el f1 score ya que es una métrica que equilibra la precisión y el recall.

SVM polinomial nos dio un buen f1 score pero en la competencia nos dio un bajo rendimiento, esto debe suceder a que overfitteo y aprendió hasta el ruido.

SVM fue un modelo que demoró mucho tiempo en analizar los hiperparametros con grid search y en generar las predicciones, fue muy bajo el rendimiento para coste temporal, comparándolo por ejemplo con random forest.

Nuestros mejores modelos fueron Random Forest y XG Boost. Sin embargo no mejoraron siendo usados para ensamble híbridos.

En conclusión, es esencial para lograr una buena predicción:

- Realizar un buen trabajo de preprocesamiento
- Diferenciar variables cuantitativas de las cualitativas

- Utilizar one hot encoding para variables cualitativas(para los modelos que requieran usar datos numéricos).
- Normalizar los datos
- Distinguir entre variables relevantes e irrelevantes
- Realizar una búsqueda de parámetros con Cross Validation
- Usar parámetros de regularización evitando el overfitting.

Una vez realizados estos pasos, se procede a determinar cuál es el modelo que mejor se adapta a nuestro dataset.