Trabajo Práctico 1 - Reservas de Hotel

Checkpoint 3

Integrantes:
Ignacio Latorre - Padrón: 101305
Nicolas Ronchese - Padrón: 108169
Gastón Avila - Padrón: 104482

Esta parte del TP consistió en generar distintos ensambles de modelos, optimizar sus hiperparametros y entrenarlos con los datos del dataframe hotels para generar nuevas predicciones y obtener la mejor puntuación posible en Kaggle.

Estos ensambles son: KNN, SVM, Random Forest y XGBoost.

KNN

Se realizaron distintas corridas optimizando sus hiperparámetros. Finalmente se optó por un algorithm= 'kd_tree', metric='manhattan' , weights='distance 'y se obtuvo un **0.80** de f1. Se puede ver el desarrollo en el notebook "**knn.ipynb**".
Aclaración: se trataron de utilizar diferentes algoritmos en el pero el tiempo que tardaban en procesarse redujo eso al los 3 que se ven la notebook

SVM

Se realizaron distintas corridas probando los diferentes tipos de kernel y tratando de utilizar el pipeline . En la notebook se puede ver el resultado de los diferentes kernel perreo en nuestra opinión el mejor resultado que se obtuvo fue un **0.80** de f1 con kernel='rbf', C=5, gamma=2. Se puede ver el desarrollo en el notebook "**svm.ipynb**".
Aclaración: Se trató de utilizar el RandomizedSearchCV pero por el tiempo que demoraba en devolver los resultados tuvimos que eliminarlo. Esto mismo ocurrió con el pipeline y el kernel poly

RANDOM FOREST

Se realizó una GridSearch con diversos valores para hallar los mejores hiperparametros del Random Forest haciendo Cross Validation con 8 K-Folds. Entrenamos y obtuvimos que los mejores parámetros eran con criterion: 'entropy' , min_samples_leaf: 1, min_samples_split: 4, n_estimators: 50. El mejor resultado fue un F1 score de **0,84**. Se puede ver el desarrollo en la notebook "**RF.ipynb**"

XGBOOST

Se realizaron distintas corridas optimizando sus hiperparámetros. Finalmente se optó por un learning_rate = 0,5 y se obtuvo 0.82 de puntaje en la competencia. Se puede ver el desarrollo en el notebook "**XGBoost.ipynb**".

Stacking y Voting

Utilizamos los resultados de los modelos anteriores en los ensambles voting y stacking
Aclaración: a la hora de hacer el submit los resultados nos dieron 0.50 y 0.68 respectivamente, pero en nuestra notebook el resultado del f1 era superior al 80 en ambos casos. Estos resultados no son los esperados, lo cual amerita una profunda revisión en el procesamiento de los datos y los algoritmos utilizados.