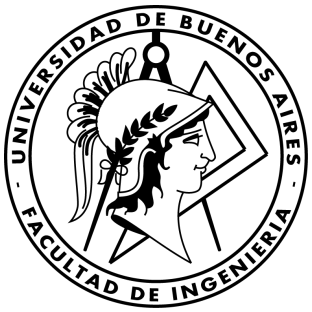

Trabajo Práctico 1
Reservas de hotel - Resumen checkpoint 3

[75.06] Organización de datos
Primer cuatrimestre 2023
Grupo 19: Sudanalytics

Integrantes

Nombre	Padrón	Mail
Re, Adrián Leandro	105025	are@fi.uba.ar
Lorenzo, Luciano Andrés	108951	llorenzo@fi.uba.ar
Toulouse, Alan	105343	atoulouse@fi.uba.ar
Tonizzo, Nicolas	107820	ntonizzo@fi.uba.ar

Fecha de entrega: 11/05/2023

Índice

1. Modelos

1.1. K-Nearest Neighbours

Para la busqueda de los mejores hiperparametros de este modelo utilizamos

1.2. SVM

En SVM decidimos utilizarlo con PCA, para poder reducir los tiempos de entrenamiento, y poder utilizar mas para la búsqueda de hiperparametros. Probamos el kernel polinómico y el RBF, y este ultimo es el que nos trajo mejores resultados. Esto podría deberse a la gran cantidad de variables categóricas en el dataset, y los polinomios no son óptimos para encontrar este patrón en los datos.

1.3. Random Forest

en este modelo, hacemos la búsqueda de hiperparametros usando 3 métricas diferentes que consideramos las mas importantes que son 'precision','recall' y 'f1'. buscamos usar los parámetros donde todas las métricas tienen en promedio el mayor valor para así poder encontrar los mejores resultados.

1.4. Extreme Gradient Boost

Para este modelo, buscamos por randomized search los mejores hiperparametros, y calculamos la métrica de f1 score. Fue el que menos tiempo consumió de todos los modelos.

2. Ensambles

Con los modelos construidos anteriormente, ensamblamos dos nuevos modelos. Vamos a dejar afuera el modelo de SVM, ya que probamos distintas formas de hacer los ensambles y los que entrenamos con este modelo consumían demasiado tiempo en ser entrenados.

2.1. Voting

Cada modelo dará una predicción acerca de la instancia que se esta intentando clasificar, este ensamble las analizará y se quedará con la predicción que más veces haya salido. En este caso, los estimadores serán de tipo XGBoost, Random Forest y K-Nearest Neighbours.

2.2. Stacking

La idea detrás de este ensamble es usar la salida de los modelos entrenados anteriormente, y en la salida de todos ellos, entrenar otro que decida qué predicción usar de todas las dadas. En este caso, usamos los modelos XGBoost, Random Forest y Decision Tree como estimadores, y en la salida el modelo es de tipo K-Nearest Neighbours.

3. Conclusiones

En principio, el modelo que mejor resultados nos dio fue el XGBoost. Al hacer los ensambles hibridos, esto solo mejoró los resultados tanto en stacking como en voting, por lo que concluimos que varios modelos en un ensamble híbrido mejoran su rendimiento.

4. Referencias

- Paper "Hotel booking demand datasets"proporcionado por la cátedra.