Checkpoint 3 - Grupo 23

Introducción

Las técnicas más utilizadas durante esta etapa fueron la de One Hot Encoding para que nuestros modelos puedan utilizar la información de variables categóricas. Otra técnica utilizada fue el escalador de datos para que nuestro modelo SVM pueda trabajar correctamente y de mejores resultados.
Cabe recalcar que los modelos fueron entrenados con los mismos datos y técnicas para que luego puedan ser utilizados, los que mejor resultado dan, en los ensambles híbridos.
En el dataset no realizamos nuevas modificaciones, mantuvimos el dataset del capítulo anterior.

Construcción del modelo

Detallar como mínimo los siguientes puntos:

- Hiperparámetros otimizados para KNN: algorithm,weight, metric y cantidad de vecinos
- Hiperparámetros optimizados para SVN: kernel, degree y gamma
- Hiperparámetros optimizados para RF: max depth, min samples leaf y n estimators
- Hiperparámetros optimizados para XGBoost: max depth, learning rate, n estimators, gamma, min child weight, subsample, colsample bytree, reg alpha, reg tree
- Modelos utilizados para el ensamble tipo voting: random forest y xg boost
- Modelos utilizados para el ensamble tipo stacking: random forest y xg boost
  Meta modelo ensamble stacking: regresión logística

Cuadro de Resultados

Medidas de rendimiento en el conjunto de TEST:

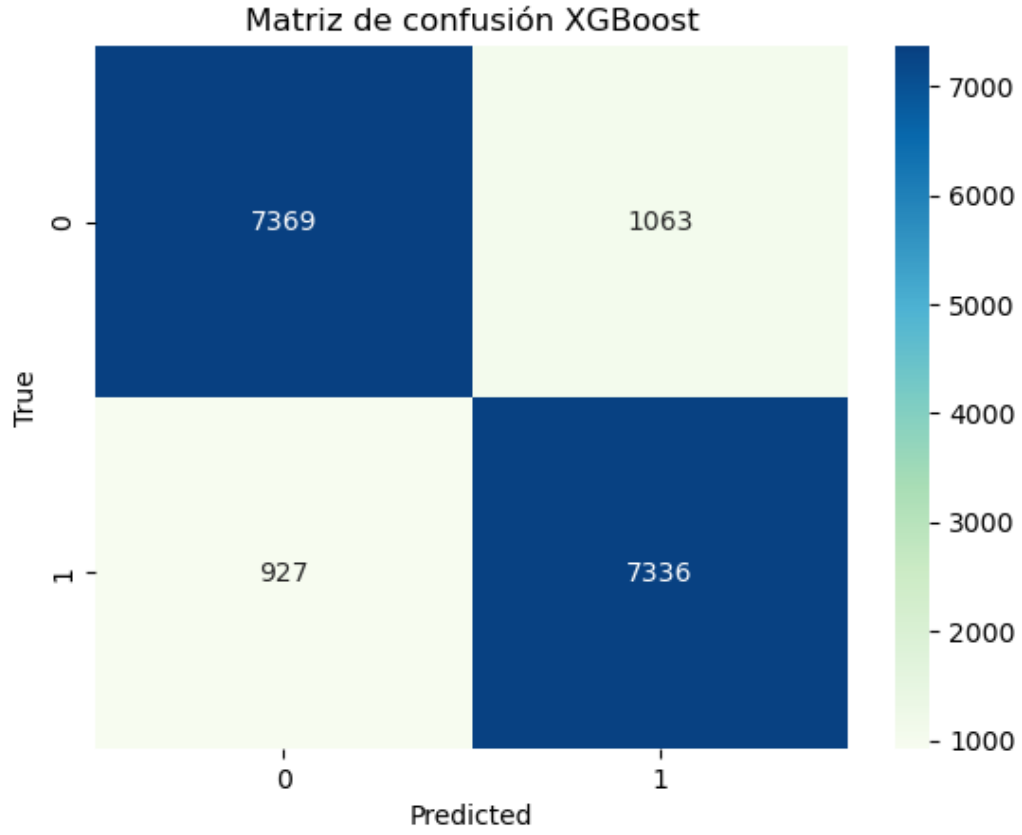
- F1
- Precision
- Recall
- Métrica X (F1 en conjunto de train)
- Resultado obtenido en Kaggle.

La métrica X a utilizar va a ser el f1 score sobre los datos de entrenamiento para determinar y evaluar si hubo overfitting o underfitting.

Modelo	F1-Test	Precision Test	Recall Test	Metrica X	Kaggle
KNN	0,7787	0,7455	0,8152	0,9959	0,77821
SVM	0,8550	0,8742	0,8367	0,8770	0,84595
Random Forest	0,8800	0,8651	0,8953	0,9551	0,8666
XGBoost	0,8805	0,8734	0,8878	0,9482	0,8692
Voting	0,8723	0,8839	0,8609	0,9210	0,8671
Stacking	0,8778	0,8772	0,8785	0,9615	0,8674

- **KNN:** Consiste en buscar los *k* elementos más cercanos para determinar de qué clase es una observación
- **SVM:** Consiste en elevar la dimensión de las observaciones a una donde se puedan separar en dos grupos para poder predecir nuevas observaciones
- **Random Forest:** Consiste en generar muchos árboles pequeños y entre todos ellos realizar las predicciones
- **XGBoost:** Se basa en el concepto de Boosting, armando muchos modelos débiles y entrenandolos secuencialmente para que los siguientes corrijan los errores que cometieron los anteriores. XGBoost lo hace tomando los puntos de corte con mayor ganancia paso a paso. Este fue nuestro mejor modelo, era de esperarse debido a que es el más popular a la hora de realizar predicciones con algoritmos de machine learning, esto es gracias a su robustez por su entrenamiento iterativo.
- **Voting:** Consiste en tomar n modelos entrenados y predecir lo que la mayoría de modelos predicen.
- **Stacking:** Es un concepto muy similar al de voting, salvo que en lugar de predecir lo que la mayoría predice, cada predicción es el input de un último modelo que es el que predice el output.

Matriz de Confusion



Este es el heatmap de las predicciones de nuestro modelo XGBoost optimizado con sus hiperparametros. Como se puede observar, nuestro modelo predice correctamente en la gran mayoría de los casos. Sin embargo, notamos que el modelo tiende a dar más falsos positivos.

Tareas Realizadas

Como somos solo 2 personas, ambos trabajamos todos los modelos en conjunto.