Checkpoint 3 - Grupo Alan Taylor

Introducción

En este checkpoint probamos predecir con los modelos vistos y buscamos sus hiperparametros para encontrar los que más se ajustaban a nuestra data. Para mayor entendimiento y orden reescribimos nuestra data en otro archivo. Observamos las métricas más destacables y optimizamos modelos para encontrar la mejor predicción posible.

Construcción del modelo

KNN: (K vecinos más cercanos) se basa en el principio de que objetos similares tienden a estar cerca unos de otros. Encuentra los K puntos de entrenamiento más cercanos en función de alguna métrica de distancia.
Hyperparameters :`n_neighbors=17,weights='distance',algorithm='ball_tree',p=1`
SVM: Encuentra el mejor hiperplano que separa los datos de distintas clases en un espacio de características de alta dimensión.
Hyperparameters: `kernel='linear',C=10,gamma="auto",degree=2,coef0=0,class_weight= None`
Random Forest: Conjunto de árboles de decisión, cada árbol se entrena en un subconjunto aleatorio de los datos de entrenamiento y en conjunto generan predicciones.
Hyperparameters:`n_estimators=551,bootstrap=True,max_depth=28,max_features=22,min_samples_split=2,min_samples_leaf=1,random_state=71`
XGBoost: Similar a RF pero su enfoque principal es la optimización de gradientes, ajusta iterativamente sus árboles para minimizar una función de pérdida (Aprende de sus errores).
Hyperparameters:`n_estimators=1000,max_depth=26,learning_rate=0.01,subsample=1,colsample_bytree=0.8,gamma=0.3,objective="reg:logistic",random_state=30`
Voting: Conjunto de modelos que votan, utilizamos los modelos XG Boost, Random Forest y SVM, fuimos optimizando individualmente cada modelo para mejorar este ensamble.
Stacking: Modelos apilados, utilizamos XG Boost y Random Forest, y como meta modelo Logistic Regression CV. Optimizamos al máximo los modelos de XGB y RF, y los hiperparametros de Logistic Regression para que haga predicciones más eficientes.

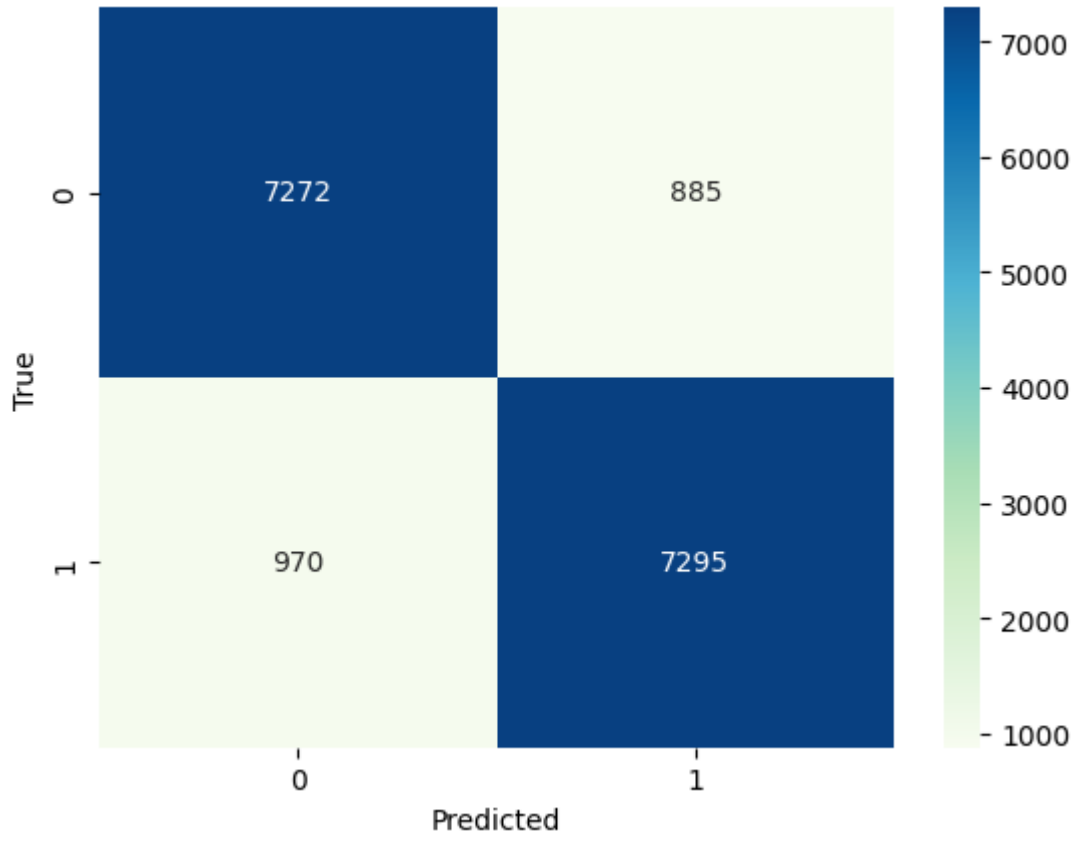
Cuadro de Resultados

Modelo	F1-Test	Precision Test	Recall Test	Accuracy	Kaggle
KNN	0.7953	0.7750	0.8167	0.7910	0.7867
SVM	0.7937	0.8029	0.7846	0.7964	0.7934
Random Forest	0.8871	0.8905	0.8837	0.8867	0.88015
XG Boost	0.8808	0.8808	0.8809	0.8810	0.8787
Voting	0.8871	0.8918	0.8826	0.8870	0.8815
Stacking	0.8859	0.8919	0.8800	0.8860	0.8815

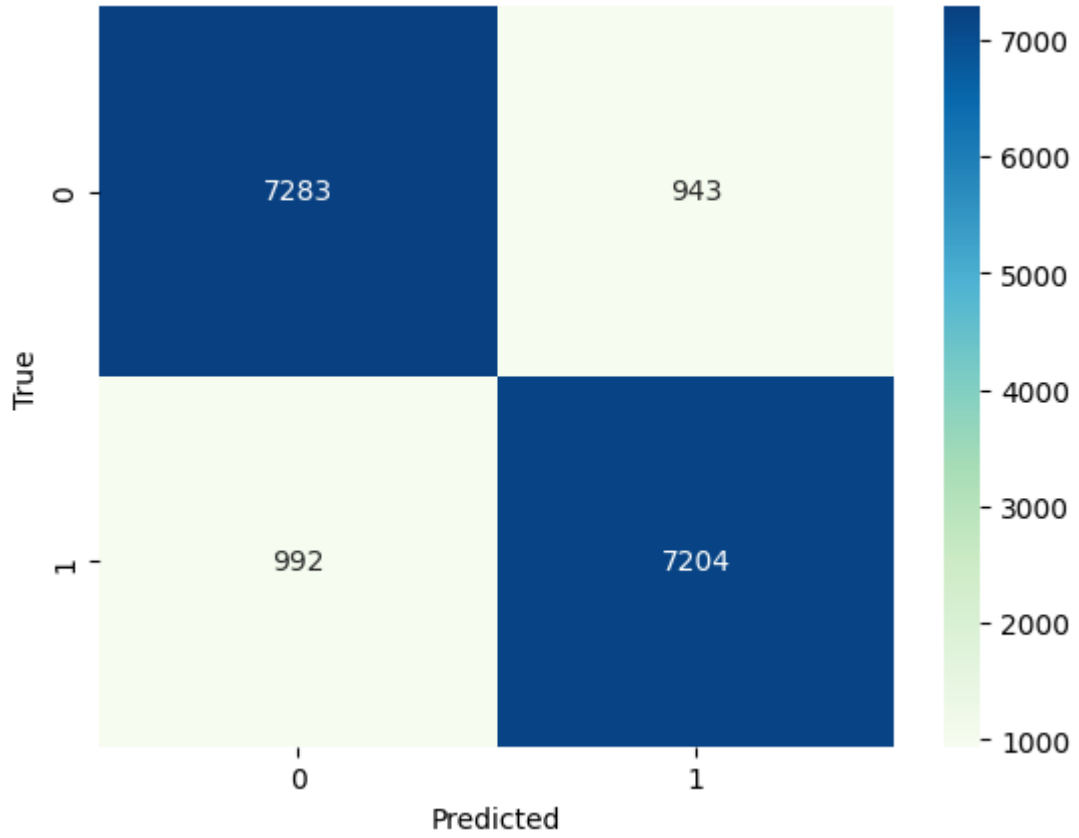
Matriz de Confusión

Nuestro mejor modelo es Voting y Stacking, esta es su matriz de confusión.

Voting:



Stacking:



Tareas Realizadas

Integrante	Tarea
Estefano Polizzi	Optimización de Hiperparametros, Optimización de las métricas, Armado de ensambles, Informe, Submits en kaggle, Modificación de la Notebook.
Santiago Bautista Trezeguet	Optimización de Hiperparametros, Armado de ensambles, Informe, Modificación de la Notebook.
Ignacio Oviedo	Optimización de Hiperparametros, Optimización de las métricas, Armado de ensambles,Modificación de la Notebook.