Checkpoint 3 - Grupo 17

Systeam

Integrantes

Del Rio, Juan Sebastián - 103337

Brizuela, Sebastián - 105288

Agha Zadeh Dehdeh, Lucía - 106905

Fecha de entrega: 19 de Octubre de 2023

Introducción

Se implementaron los modelos Knn, SVC, Random Forest y XGBoost, para luego ejecutar los procesos de entrenamiento de los modelos Voting y Stacking usando como estimador final al XGBoost.
Para agilizar la ejecución de los algoritmos se utilizaron copias de la notebook para entrenar los modelos en paralelo y luego importarlos desde el original. Para esta instancia no se efectuó ninguna modificación sobre el dataset.

Construcción del modelo

KNN: n_neighbors: range(*10,15*); weights: [*distance, uniform*]; algorithm: [*ball_tree, kd_tree, brute*]; metric:[*euclidean, manhattan, chebyshev*].
SVC: kernel: [*linear, radial*]; C=5; (en el kernel radial se utilizó gamma: 0.1).
Random Forest: criterion: [*gini, entropy*]; min_samples_leaf: range(*30, 35*); min_samples_split: range(*50, 55*); ccp_alpha: np.linspace(*0, 0.05, n*); max_depth: range(*35, 40*); n_estimators: range(*199, 350*).
XGBoost: n_estimators: range(*50, 200, 10*); max_depth: range(*3, 10*); learning_rate: [*0.01, 0.1, 0.2, 0.3, 0.5*]; subsample: [*0.7, 0.8, 0.9, 1.0*]; colsample_bytree: [*0.6, 0.7, 0.8, 0.9, 1.0*].
Voting: estimators: [XGBoost, Random Forest, KNN]; voting: hard.
Stacking: estimators: Random Forest(n_estimators: 100), KNN(n_neighbors: 15); final_estimator: XGBoost; passthrough: True; cv: 5; verbose: 2.

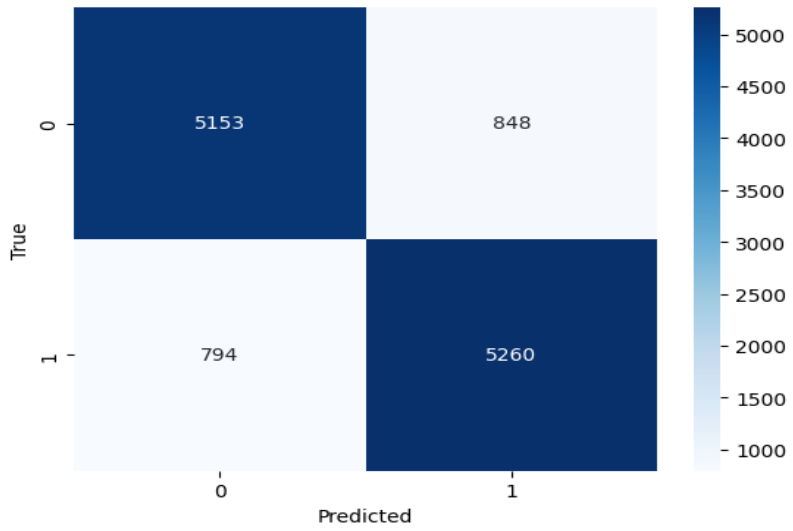
Cuadro de resultados

Modelo	F1-Test	Presicion Test	Recall Test	Accuracy Test	Kaggle
Stacking	0.86116	0.86498	0.86884	0.86379	0.86193
Voting	0.86497	0.85295	0.84126	0.85433	0.84667
Random Forest	0.86411	0.81076	0.76362	0.82098	0.81729
XGBoost	0.85691	0.86267	0.86851	0.86113	0.85477
SVC Lineal	0.80918	0.82324	0.83779	0.819	0.46341
SVC Radial	0.83663	0.84041	0.84423	0.83898	0.33351
KNN	0.77896	0.81632	0.85744	0.80622	0.80411

Stacking: Ensamble que entrena modelos base para combinar predicciones y luego usar un modelo meta para la estimación final, en nuestro caso usamos como modelos base al Random Forest y al Knn, y como modelo meta al XGBoost.
Voting: Algoritmo que utiliza las predicciones de otros modelos para tomar una decisión final. Cada modelo hace su propia predicción y luego se realiza un voto (mayoritario o ponderado) de las predicciones y se queda con el que realizó una predicción mayoritaria.
Random Forest: Crea múltiples árboles de decisión y combina sus resultados para tener una predicción más precisa.
XGBoost: Crea árboles de decisión en serie y en cada iteración corrige los errores de los árboles anteriores realizando una regularización para evitar un sobreajuste. La cantidad de árboles creados son según los que uno quiera o hasta que los árboles no logren mejorar el ajuste. Se puede paralelizar entre clústeres y tiene un entrenamiento veloz.
SVC: Clasificador que construye un hiperplano o conjunto de hiperplanos en el espacio para la búsqueda del mejor hiperplano de separación entre las clases.
KNN: Clasifica un nuevo dato basado en la mayoría de las etiquetas de los K vecinos más cercanos en un conjunto de entrenamiento. La elección de K y la distancia utilizada son factores importantes en su rendimiento.

Matriz de Confusión del mejor modelo

Como se observa en el cuadro de resultados nuestro mejor modelo es el **Stacking.**



Observaciones:

En la matriz se observa que el modelo fue capaz de predecir 5260 Verdaderos Positivos y 5153 Verdaderos Negativos. Y por otro lado el modelo tuvo 794 Falsos Negativos y 848 Falsos Positivos.

La diferencia de cifras entre los Verdaderos y los Falsos es bastante favorable en relación con el total de datos.

Tareas Realizadas: Se realizaron todas las tareas siempre en grupo por reuniones en meet por lo tanto no hubo una división clara de las mismas.