Checkpoint 3 - Grupo 32

Introducción

Para este checkpoint separamos la entrega en dos notebooks; en la primera se encuentran los mismos cambios realizados en el checkpoint 2, no hubo modificaciones, y en la segunda notebook se encuentran la realización de la consigna en sí.
Al momento de la construcción de los modelos SVM realizamos un breve análisis y prueba donde concluimos que debíamos normalizar los datos ya que mejoraba considerablemente el rendimiento en general (ver sub-sección "*Datos normalizados vs NO normalizados*" en "*SVM model*" para mayor detalle).

Construcción del modelo

Aclaración: La explicación de cada hiperparámetro está detallada en cada modelo, en la notebook número 2.

Hiperparámetros optimizados para KNN
- N_neighbors
- Weights
- Algorithm
- Metric

Hiperparámetros optimizados para SVM
- Kernel
- C
- Gamma
- Degree
- coef0

Hiperparámetros optimizados para RF
- N_estimators
- Min_samples_split
- Min_samples_leaf
- Max_features
- Max_depth
- Bootstrap

Hiperparámetros optimizados para XGBoost

- subsample
- reg_lambda

- reg_alpha
- n_estimators
- max_depth
- learning_rate
- gamma
- colsample_bytree

Modelos utilizados para el ensamble tipo voting

- KNN
- SVM con kernel polinómico
- SVM con kernel radial
- Random Forest
- XGBoost

Modelos y meta modelo utilizados para el ensamble tipo stacking

- Idem modelos utilizados para Voting
- Meta modelo: Cross Validation con Regresión logistica

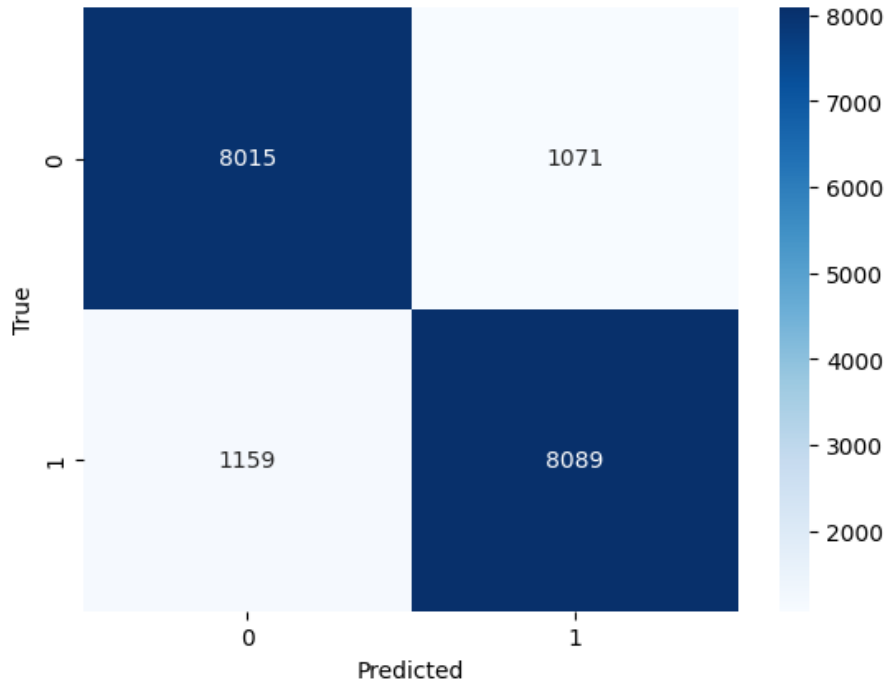
Cuadro de Resultados

Medidas de rendimiento en el conjunto de TEST:

Modelo	F1-Test	Presicion Test	Recall Test	Accuracy	Kaggle
KNN	0.8399	0.835	0.84	0.84	0.83693
SVM radial	0.844	0.845	0.84	0.84	0.8466
SVM polinómico	0.845	0.845	0.845	0.84	0.84346
Random Forest	0.8789	0.875	0.875	0.88	**0.87208**
XGBoost	0.8778	0.875	0.875	0.88	0.86896
Voting	0.8657	0.865	0.865	0.87	0.86807
Stacking	0.8788	0.88	0.88	0.88	0.87068

- **Modelo KNN:** Algoritmo de clasificación y regresión, en el cual, para clasificar un punto de datos más cercanos en el conjunto de entrenamiento y toma una decisión en función de la mayoría de etiquetas.
- **SVM Radial:** Algoritmo de clasificación que busca encontrar el hiperplano que mejor separa dos clases de datos
- **SVM polinomico:** Similar al SVM Radial, solo que en este caso utiliza un kernel polinomico y no uno radial
- **Random Forest:** Algoritmo de conjunto que combina múltiples árboles de decisión. Cada árbol se entrena con una muestra aleatoria del conjunto de datos, y luego se combinan la salida de los árboles para tomar decisiones. Fue el modelo que mejor resultados nos dio, con un score 0.87208 en Kaggle.
- **XGBoost:** Es un método de aprendizaje automático supervisado para clasificación y regresión basado en el Boosting, en este caso usado para clasificación. Esto es una forma de aprendizaje de los modelos basada en generar predicciones y asignar un mayor peso a aquellas mal clasificadas, así iterativamente hasta un límite determinado por los hiperparámetros.
- **Voting:** Método de conjunto en el que múltiples modelos se utilizan para tomar una decisión colectiva. La decisión final se basa en la mayoría votada.
- **Stacking:** Técnica de conjunto en la que se apilan modelos y la decisión final la toma otro modelo entrenado por los datos output de los apilados..

Matriz de Confusion



Matriz de confusión del modelo Random Forest. Podemos observar que tiene un rendimiento bueno en general y que realizó un poco mejor la predicción sobre los verdaderos

Tareas Realizadas

Integrante	Tarea
Daniel Agustin Marianetti	Construcción de modelos Optimización de hiperparametros Armado de reporte
Ezequiel Lazarte	Preprocesamiento de datos Optimización de hiperparametros Armado de reporte
Franco Ezequiel Rodriguez	Optimización de hiperparametros Armado de reporte