Checkpoint 3 - Grupo 12

Introducción

En este checkpoint, buscamos mejorar la performance (de predicción) a través de la optimización de hiperparámetros de algunos de los modelos de aprendizaje no supervisados: KNN, SVM, así como los modelos de ensamble: RandomForest, XGBoost. Por otro lado, buscamos seleccionar los mejores modelos base para los modelos de ensamble híbridos, siendo estos: Stacking, Voting.
Volvimos a efectuar los mismos cambios sobre nuestro dataset, como en el checkpoint anterior, y no hicimos modificaciones nuevas.

Construcción del modelo

Vamos a explicar la importancia de los hiperparámetros que buscamos optimizar para cada modelo, para luego construirlos:

- KNN: Para KNN se buscó primero un número de vecinos que de la mayor cantidad de aciertos para luego buscar los otros hiperparámetros a través de Random Search Cross Validation.
- SVM: Para SVM lo que hicimos fue buscar el mejor kernel y el mejor valor del hiperparámetro “c”, que controla la regularización del modelo, donde la regularización es un proceso que se utiliza para prevenir el sobreajuste (overfitting) en el entrenamiento de un modelo de SVM. Luego se buscó un coeficiente “degree” óptimo para el kernel polinómico.
- Random Forest (RF): Se optimizaron sus hiperparámetros a través de k-fold cross validation.
- XGBoost: Primero buscamos un learning_rate óptimo para el resto de los hiperparámetos por defecto, luego se realizó Random search cross validation para encontrar los mejores hiperparámetros restantes.

Para los modelos de tipo Voting y Stacking, usamos como modelos base (los optimizados) de: KNN, XGBoost, RandomForest, y como meta-modelo usamos a la regresión logística, por ser un problema de clasificación.

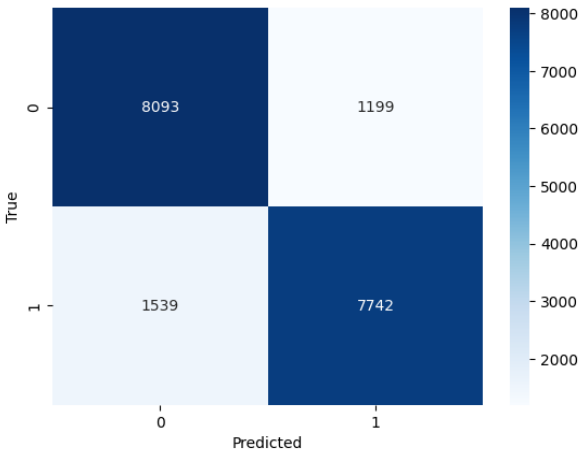
Cuadro de Resultados

Medidas de rendimiento en el conjunto de TEST:

Modelo	F1-Test	Precision Test	Recall Test	Accuracy	Kaggle
KNN	0.74	0.75	0.74	0.75	0.56234
SVM	0.74	0.74	0.75	0.75	0.70845
Random Forest	0.87	0.83	0.85	0.86	0.84591
XGBoost	0.84	0.85	0.84	0.85	0.82653
Voting	0.85	0.87	0.83	0.85	0.83459
Stacking	0.86	0.84	0.87	0.86	0.84374

KNN: Se utilizó el método de clasificación supervisada k-vecinos más cercanos, con 8 vecinos y distancia Manhattan.
SVM: Se utilizó support-vector machines con un kernel polinómico de grado 5 y un grado de normalización C = 10.
Random forest: Se utilizó el modelo clasificador Random Forest optimizando sus hiperparámetros con k-fold cross validation. (Este fue el mejor modelo en Kaggle)
XGBoost: Su utilizó el modelo clasificador XGBoost optimizando sus hiperparámetros con k-fold cross validation.
Voting: Ensamble del tipo voting de los mejores modelos que obtuvimos de KNN, Random Forest y XGBoost.
Stacking: Ensamble del tipo stacking de los mejores modelos que obtuvimos de RandomForest y XGBoost.

Matriz de Confusión (Del modelo Random Forest)



Lo que se puede observar es que el modelo tiene alta precisión para ambas etiquetas, cuya fórmula es: TP / (TP + FP)),y tiene un recall alto para ambas etiquetas, con su fórmula: TP / (TP + FN), lo que sugiere que el modelo es preciso y efectivo en la identificación de instancias positivas.

Tareas Realizadas

Integrante	Tarea
Gaston Sabaj	Construí los clasificadores de RF, XGBoost, Voting, Stacking, y contribuí al informe.
Juan Yago Pimenta	Construí los clasificadores de KNN, SVM ,Stacking y contribuí al informe.