Checkpoint 3 - Grupo 39

Introducción

En el preprocesamiento hicimos unas funciones para mostrar métricas y para exportar, así no repetimos código. También hicimos normalización y balanceo del dataset para mejorar los scores que vamos a entrenar. Luego entrenábamos modelos con parámetros por default para ver cual es el score base y luego buscábamos hiperparametros. Con respecto al dataset utilizamos One Hot Encoding para procesar las variables cualitativas.

Construcción del modelo

Para KNN optimizamos:
- Weights: 'distance'
- n_neighbors: 19
- Metric: 'manhattan'
- Algorithm: 'brute'

Para SVM Lineal:
- C: 10

Para SVM Polinomico:
- Gamma: 0.1
- degree: 1
- coef0: 0.5
- C: 1

Para SVM Radial:
- Probability: False
- gamma: 0.1
- C: 10

Para RF:
- min_samples_split: 10
- n_estimators: 200
- min_samples_leaf: 1
- max_features: log2
- max_depth: None

- criterion: gini
- class_weight: balanced
- ccp_alpha: 0.0
- bootstrap: False

Para XGBoost:

- subsample: 0.8
- n_estimators: 200,
- min_child_weight: 2,
- max_depth: 8,
- learning_rate: 0.2,
- gamma: 0.2,
- colsample_bytree: 0.9

Stacking:

- base models: RF, SVM radial, XGBoost
- meta modelo: logistic regression

Voting:

- base models: RF, SVM radial, XGBoost

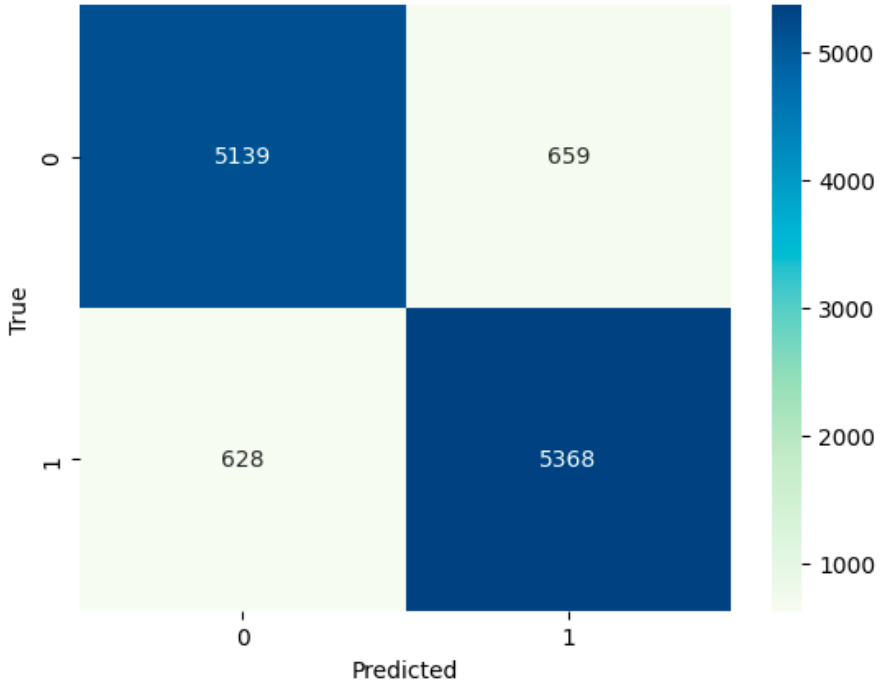
Cuadro de Resultados

Modelo	**F1-Test**	**Presicion Test**	**Recall Test**	**Accuracy**	**Kaggle**
KNN	0.840	0.845	0.840	0.840	0.759
SVM	0.835	0.840	0.835	0.840	0.827
Random Forest	0.910	0.905	0.910	0.910	0.864
XGBoost	0.890	0.885	0.885	0.890	0.777
Voting	0.890	0.890	0.895	0.890	0.866
Stacking	0.900	0.900	0.900	0.900	0.864

Para KNN utilizamos el modelo con los parámetros optimizados ya que el accuracy era mayor que con parámetros default.

Para SVM usamos el Kernel radial que es especialmente útil para conjuntos no linealmente separables y optimizamos sus parámetros para tener el mayor f1 posible.
En cambio para Random Forest utilizamos los parámetros default ya que dio mejor score que con parámetros optimizados.
Para el XGBoost utilizamos optimizados ya que el f1 score era mayor que con parámetros default.
Para Voting y Stacking decidimos utilizar los mismos modelos ya que fueron los que mejores scores tenían.

Matriz de Confusión



Matriz de confusión del modelo Voting que obtuvo el mejor score en Kaggle.
Podemos observar que de las reservas que predijimos que serían canceladas, acertamos en 5368 y fallamos en 659. Y de las que predijimos que no serían canceladas acertamos en 5139 y fallamos en 628.
De esta forma, notamos que hay una distribución muy pareja entre los errores cometidos, por lo que podemos afirmar que no hay problemas para detectar una clase en particular.

Tareas Realizadas

Integrante	Tarea
Cattaneo Ariadna Antonella	Modelos: KNN, SVM Preprocesamiento, Reporte
Agustin Ezequiel Sanchez Decouflet	Modelos: Voting, Stacking Preprocesamiento, Reporte
Franco Dario Mazzaro	Modelos: XGBoost, Random Forest Preprocesamiento, Reporte