Trabajo Práctico 2: Críticas cinematográficas

75.06 / 95.58 Organización de Datos - FIUBA
Ing. Rodríguez - 1°C 2023

Grupo 31 - “Datazo”:
- 106203 - Kisinovsky, Diego Andrés
- 102685 - Mena Giraldo, Michael Gustavo
- 104256 - Brocca, Pablo Martín

Introducción

En este Trabajo Práctico, armamos y entrenamos distintos modelos de clasificación de críticas cinematográficas en formato de texto en lenguaje natural, para predecir si son de carácter positivo o negativo, lo que se conoce como análisis de sentimientos.

Observaciones generales

Entrenamos los siguientes modelos:

- Bayes Naïve
  Elegimos los siguientes hiperparámetros para optimizar:
  'alpha' -> Parámetro de suavizado
  'fit_prior' -> Indica que se deben estimar las probabilidades a priori en función de la frecuencia de las clases en los datos de entrenamiento

- Random Forest
  Elegimos los siguientes hiperparámetros para optimizar:
  'ccp_alpha’ -> Parámetro de poda
  'max_depth' -> Profundidad máxima del árbol
  'min_samples_leaf' -> Controla el número mínimo de muestras requeridas en un nodo hoja
  'min_samples_split' -> Cantidad mínima de muestras requeridas para separar un nodo interno
  'n_estimators' -> Cantidad de estimadores

- XGBoost
  Elegimos los siguientes hiperparámetros para optimizar:
  'colsample_bytree' -> Proporción de columnas por árbol
  'learning_rate' -> Determina el tamaño del paso en cada iteración
  'max_depth' -> Profundidad máxima de cada árbol
  'n_estimators' -> Cantidad de estimadores
  'reg_lambda' -> Valor utilizado en la regularización L2
  'subsample' -> Proporción de submuestras de la data de training

- Redes Neuronales
  Se decidió utilizar una red neuronal recurrente con una única capa GRU de 64 neuronales, en base a lo visto en la clase teórica donde vimos un modelo de procesamiento de lenguaje con una arquitectura similar. No realizamos validación cruzada, pero para evitar el sobreajuste se probó con regularizador L1 y también dropout (apagar neuronas) aunque empeoró las métricas por lo que se decidió desestimarlo.
  Adicionalmente, incorporamos el modelo preentrenado RoBERTuito (Pérez et al., 2022), en particular ‘robertuito-base-deacc’ el cual realiza un proceso de desacentuación, aunque por limitaciones de implementación sólo pudimos llegar a probarlo variando únicamente la cantidad de épocas.

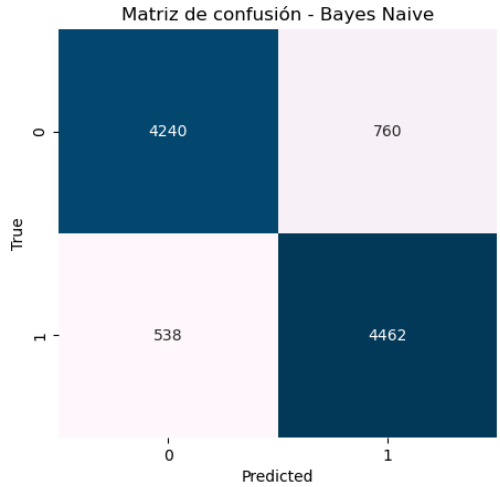
- Ensamble
  Elegimos el modelo Stacking como ensamble. Los modelos utilizados como base son Bayes Naïve, Random Forest y XGBoost. Como modelo final utilizamos un clasificador de Regresión Logística.

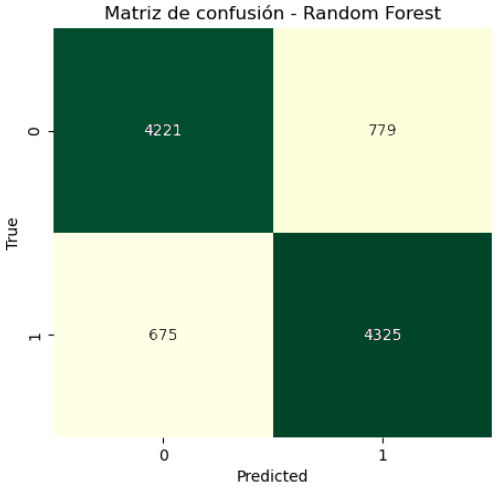
Para todos los modelos en los que los recursos nos lo permitieron, utilizamos K-Fold CV con 5 folds y Bayes Search, que utiliza optimización bayesiana para hallar los mejores hiperparámetros.
Para Bayes Naïve, Random Forest y XGBoost se observó que para el primero el uso de n-gramas mejoró las métricas en particular con n-gram (1,3) a diferencia de los otros dos, a las cuales el uso de n-gramas mayores a (1,1) les empeoraba las métricas por lo que se decidió dejar el valor por defecto generando un modelo más simple y con mejores tiempos de ejecución.
El uso de stop-words mejoraba o empeoraba las métricas dependiendo del modelo. Para Bayes Naïve, Random Forest y XGBoost hubo mejoras, mientras que para redes neuronales empeoraba.
Se probaron además otras alternativas como la lematización o el pasaje a minúsculas de todas las reseñas, pero nada de esto ayudó a mejorar las métricas de ningún modelo.

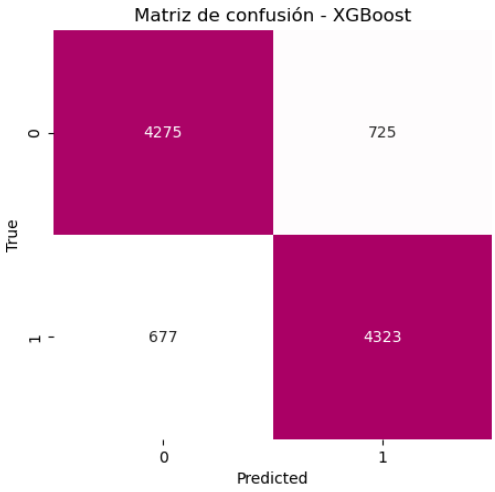
En la siguiente tabla figuran las medidas de Precisión, Recall y F1 Score obtenidas evaluando el modelo con CV en el archivo de entrenamiento, como también el mejor Score público conseguido en la competencia de Kaggle:

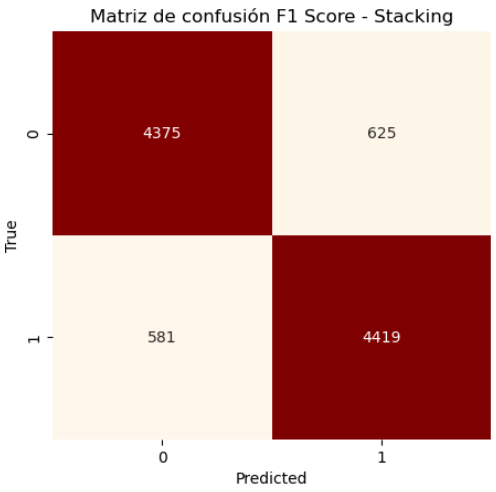
	BN	**RF**	**XGB**	**RN**	**RN (RoBERTuito)**	**Stacking**
Precision	P: 0.85 N: 0.89	P: 0.85 N: 0.86	P: 0.86 N: 0.86	P: 0.87 N: 0.88	P: 0.88 N: 0.87	P: 0.88 N: 0.88
Recall	P: 0.89 N: 0.85	P: 0.86 N: 0.84	P: 0.86 N: 0.85	P: 0.88 N: 0.86	P: 0.87 N: 0.88	P: 0.88 N: 0.88
F1 Score	0.87301	0.85609	0.86046	0.87734	0.87481	0.88
Mejor Public Score Kaggle	0.75499	0.72940	0.72455	0.77515	0.79065	0.73599

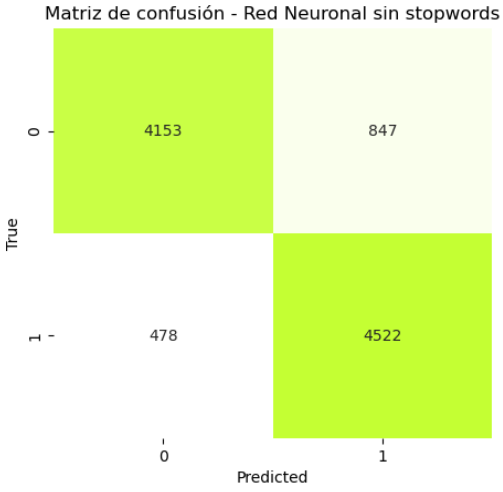
Gráficos

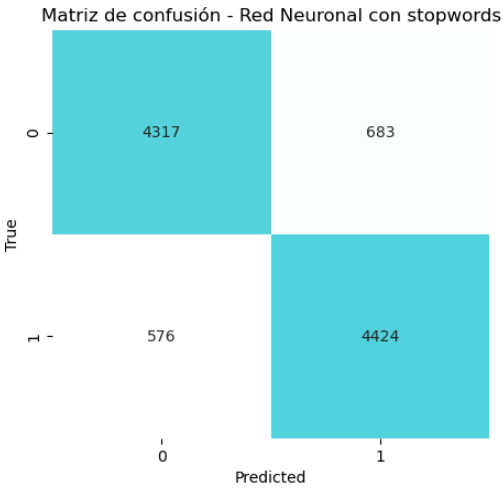


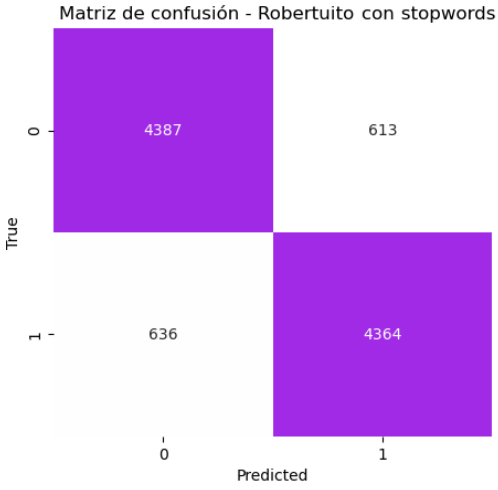


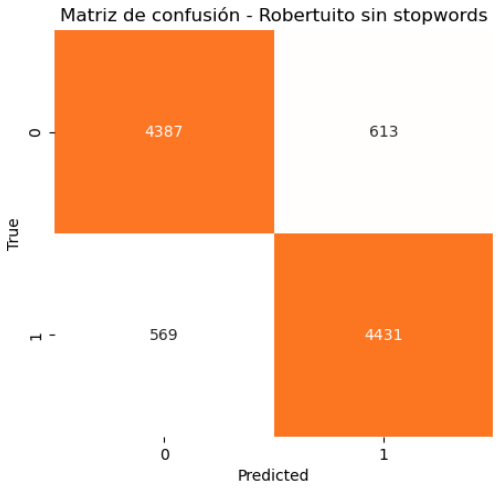

Matriz de confusión - XGBoost
4275
725
677
4323
True
Predicted
0
1


Matriz de confusión F1 Score - Stacking
4375
625
581
4419
True
Predicted
0
1

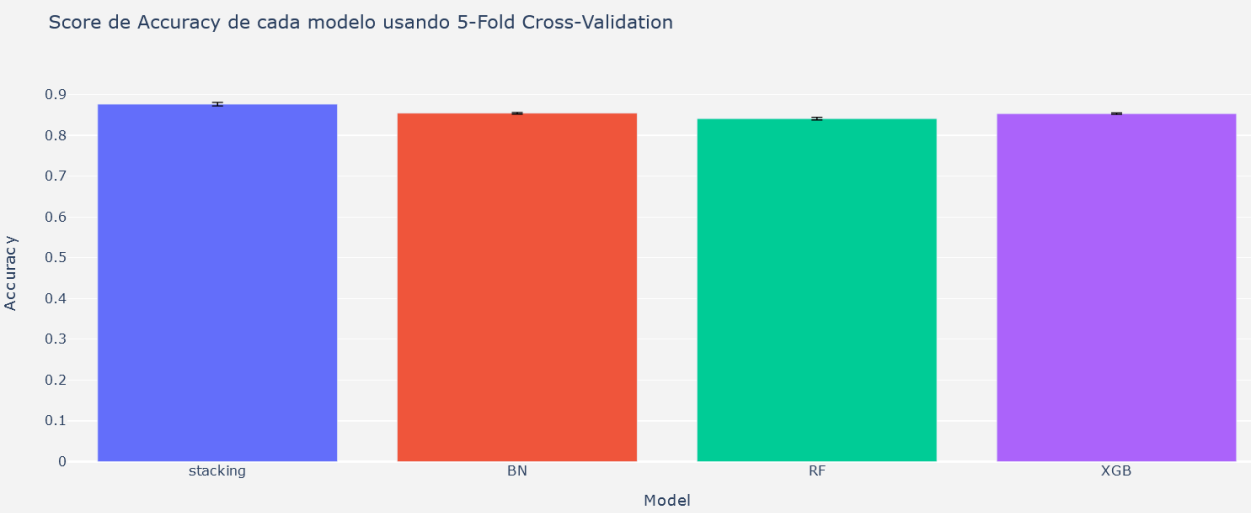

Matriz de confusión - Red Neuronal sin stopwords
4153
847
478
4522
True
Predicted
0
1


Matriz de confusión - Red Neuronal con stopwords
4317
683
576
4424
True
Predicted
0
1


Matriz de confusión - Robertuito con stopwords
4387
613
636
4364
True
Predicted
0
1


Matriz de confusión - Robertuito sin stopwords
4387
613
569
4431
True
Predicted
0
1

Comparación de accuracy para cada modelo del ensamble:



Conclusiones

Pudimos observar que, a diferencia de nuestra experiencia en el Trabajo Práctico anterior donde se utilizaron mayormente datos numéricos, en este Trabajo los modelos de redes neuronales dieron los mejores resultados, junto con Bayes Naïve. Esto deja en evidencia la efectividad que tienen ciertos modelos dependiendo del tipo de problema, en este caso tratándose de procesamiento de lenguaje natural.

Bibliografía

- Pérez, J. M., Furman, D. A., Alonso Alemany, L., Luque, F. M. (2022). *RoBERTuito: a pre-trained language model for social media text in Spanish.* https://aclanthology.org/2022.lrec-1.785.pdf