Detección de Plagio con IA

Enfocado en código de programación

Junio 2024

Andrés Magaña Pérez,
Flavio Ruvlacaba Lejia

Contenidos

Introducción

↘

El plagio no solo es un problema en trabajos escritos, sino también en el ámbito de la computación, manifestándose como clones de código. Este proyecto se enfoca en la duplicación de código y presenta nuestra propuesta para la detección de plagio, un desafío persistente tanto en el ámbito académico como profesional.

Es esencial contar con herramientas efectivas para detectar el plagio y evitar que este problema se agrave en las instituciones educativas.

Tipos de Plagio en Código

Tipo I

Fragmentos de código que son idénticos exceptuando espacios en blanco y comentarios

Tipo II

Idénticos en estructura o sintaxis, exceptuando literales, identificadores, etc.

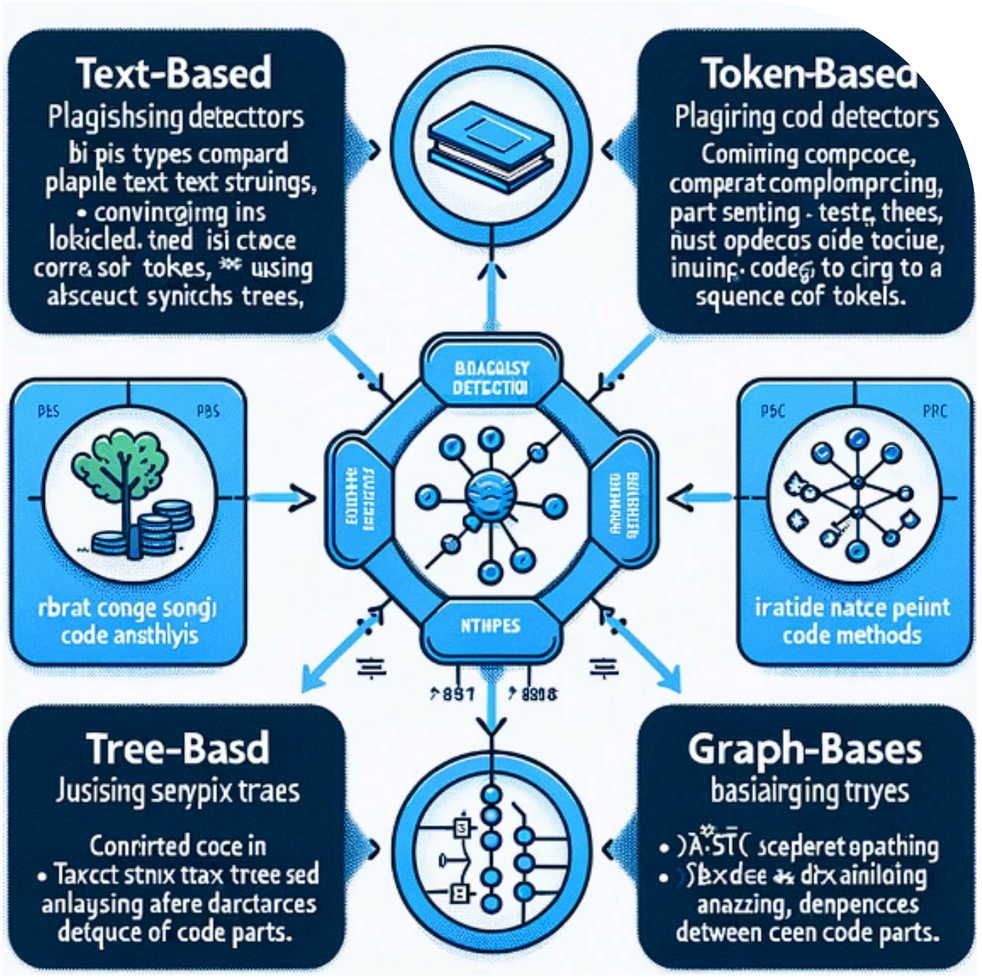
Tipo III

Incluyen variaciones gramaticales o estructurales, pero se consideran copiados

Tipo IV

Misma funcionalidad computacional, pero difieren en implementación del código

Tipos de Detectores de Plagio



Text-Based

Este método compara el código fuente a nivel de texto plano.

Token-Based

Este método convierte el código fuente en una secuencia de tokens

Graph-Based

Crea un grafo que representa la lógica y el flujo de datos.

Datos

Los datos que utilizamos en el proyecto fueron brindados por nuestros profesores

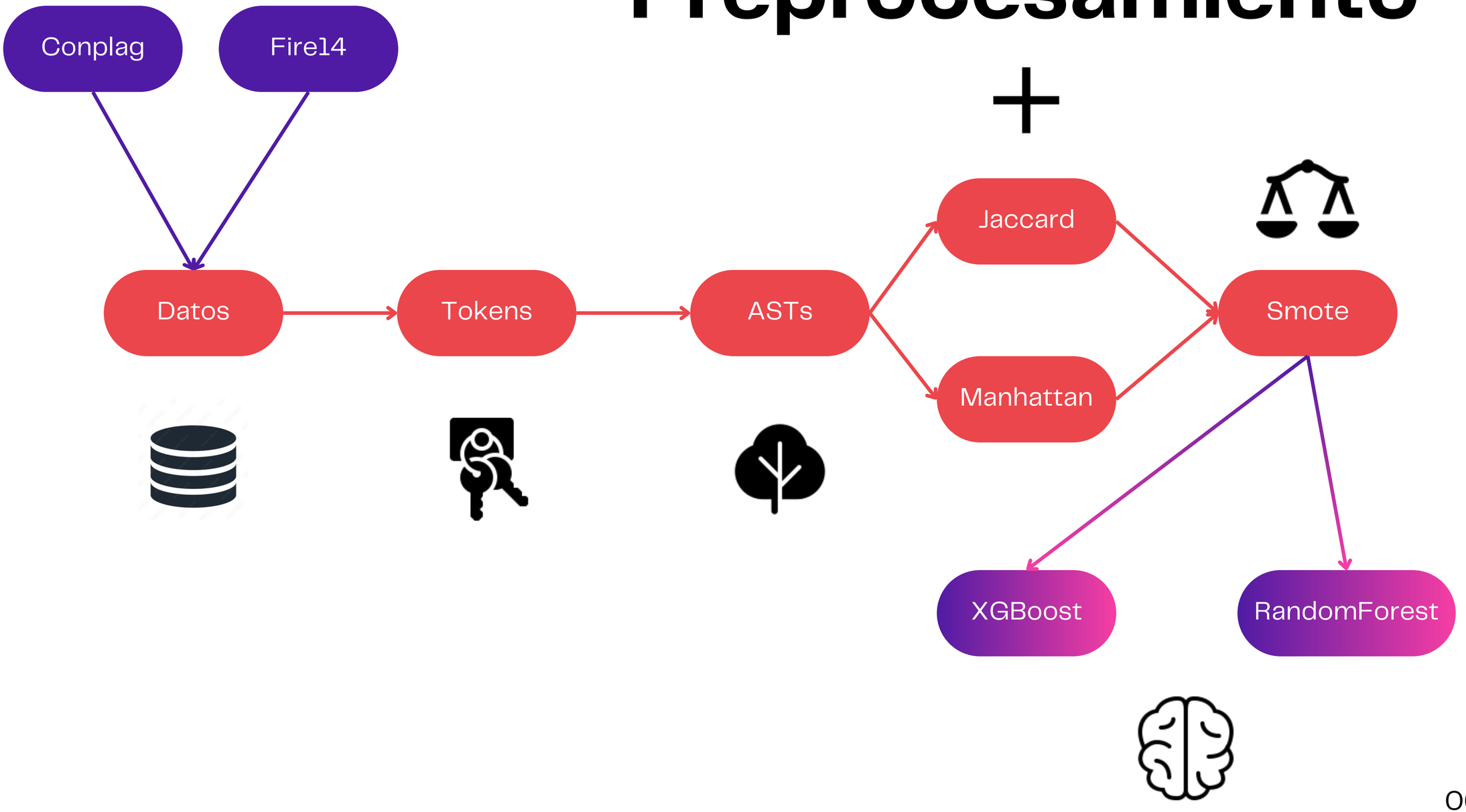
Conplag

- Archivos java
- 911 parejas, 251 plagio, 660 no
- 2 versiones de archivos
- Archivo csv con labels
- Veredicto en los labels

Fire14

- Archivos java y c
- 84 parejas de plagio
- 1 sola version de archivos
- Archivo qrel con las parejas
- No tiene veredictos

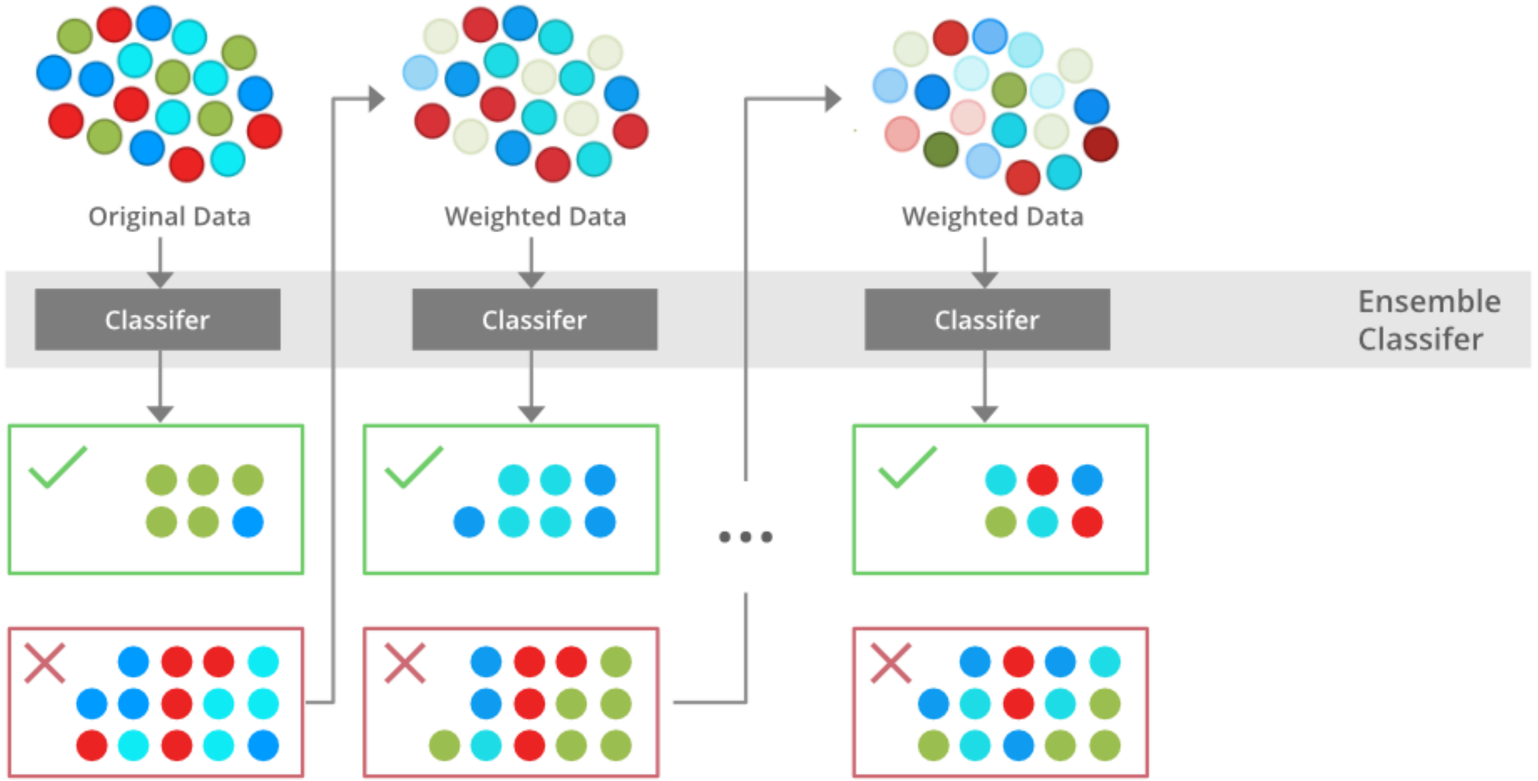
Preprocesamiento



Nuestros modelos de aprendizaje

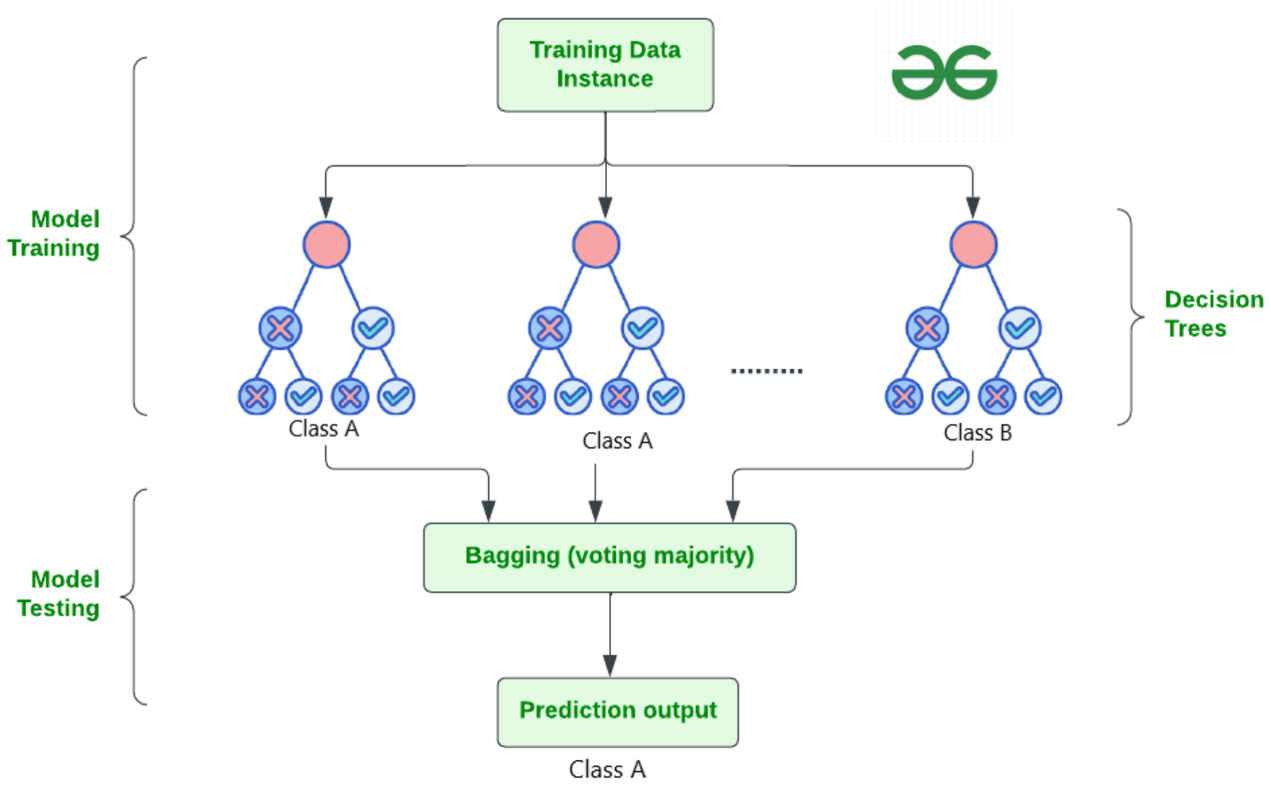
XGBOOST & RANDOMFOREST

Modelo XGBoost



XGBoost es una versión mejorada de Gradient Boosting, una técnica de aprendizaje supervisado que refuerza modelos débiles como los árboles de decisión. En cada iteración, XGBoost mejora las predicciones de los árboles anteriores.

Modelo Random Forest



El modelo de Random Forest es una técnica de aprendizaje supervisado ampliamente utilizada en tareas de clasificación y regresión. Este enfoque combina múltiples árboles de decisión para mejorar la precisión.

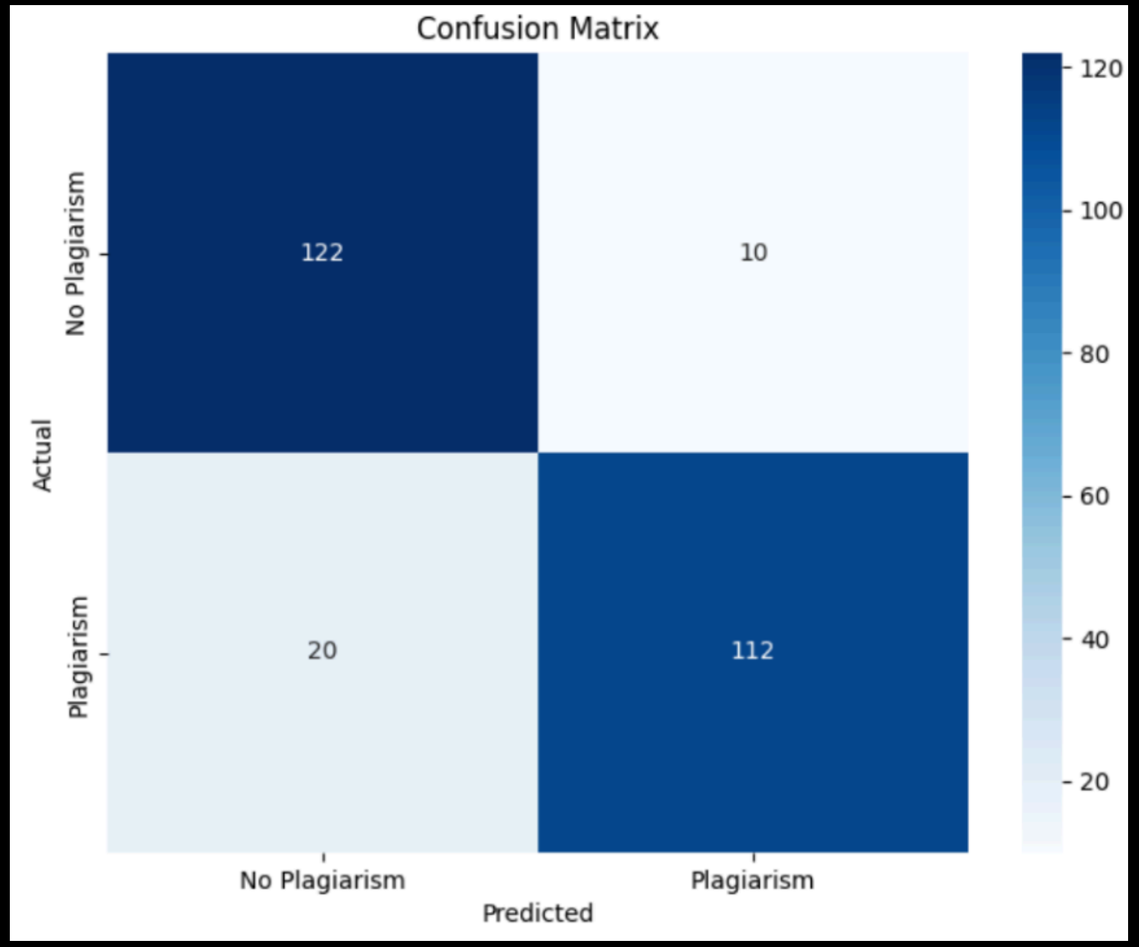
Configuramos el modelo con una búsqueda de hiperparámetros utilizando GridSearchCV para encontrar los mejores valores de los mismos. Entrenamos el modelo y evaluamos de igual manera la accuracy, perdida, precision, f1 score, recall y una matriz de confusion.

Resultados

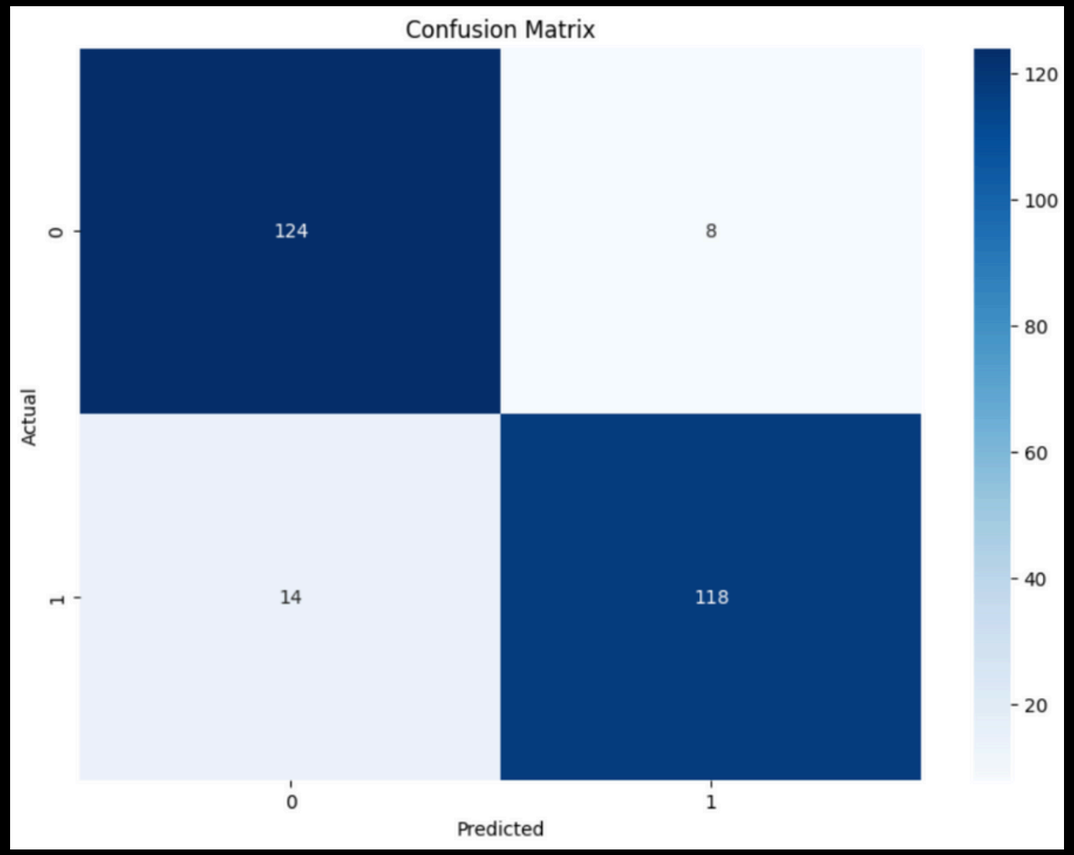
Metric	XGBoost	Random Forest	Descripción
Accuracy	91%	88%	Medida de error del modelo.
Loss	0.2974	0.2806	Medida de error del modelo; cuanto menor, mejor.
Precision	0.9365	0.918	Proporción de verdaderos positivos sobre el total de predicciones positivas.
Recall	0.8939	0.8485	Proporción de verdaderos positivos sobre el total de casos reales positivos.
F1 Score	0.9147	0.8862	Promedio armonico entre precisión y recall, balanceando ambos.

Matriz de confusion Random Forest

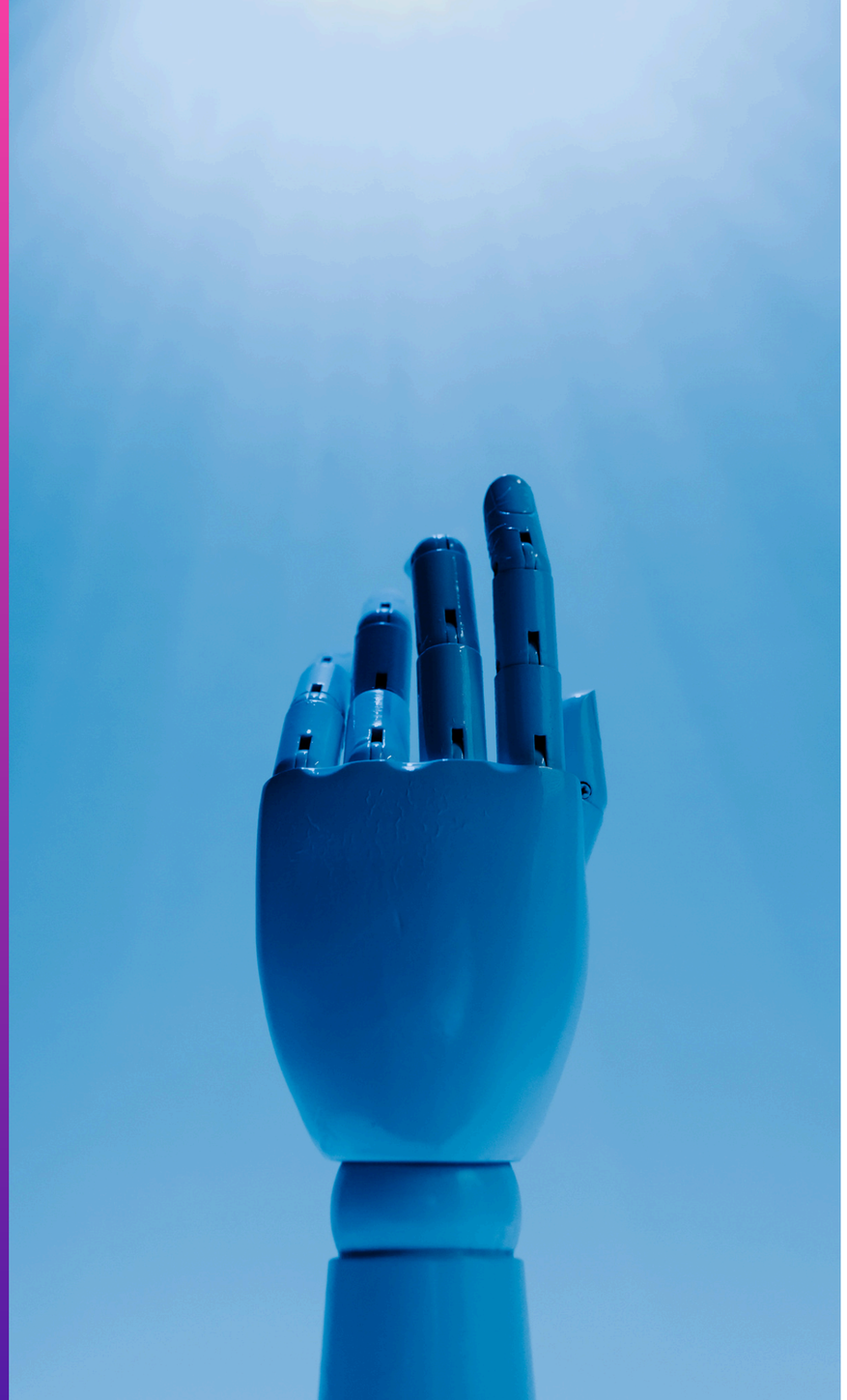
La matriz de confusión del modelo Random Forest muestra un desempeño con precisión del 88%. El modelo tiene una alta precisión en la identificación de casos de plagio, con un 92% de precisión en predicciones positivas y una tasa de verdaderos positivos del 85%. Sin embargo, la tasa de falsos negativos es del 15%. Además, la tasa de falsos positivos es baja, del 8%, demostrando que pocos documentos no plagiados son erróneamente etiquetados como plagiados



Matriz de confusion XGBoost



La matriz de confusión del modelo XGBoost demuestra un desempeño notable en la detección de plagio, con una precisión del 91%. El modelo identificó correctamente el 94% de los casos predichos como plagio y logró una tasa de verdaderos positivos del 89%. Sin embargo, existe un 6% de falsos positivos y un 11% de falsos negativos, lo que sugiere que aún algunos casos de plagio no son detectados.



Conclusion

El modelo XGBoost muestra un mejor rendimiento general en comparación con el modelo Random Forest. XGBoost tiene una precisión de prueba del 91.67%, mientras que Random Forest tiene una precisión del 88%. Además, XGBoost presenta una tasa de verdaderos positivos del 89%, superior al 85% de Random Forest, y una menor tasa de falsos negativos (11% frente a 15%).

En conclusión, el modelo XGBoost es superior en términos de precisión y capacidad de detección de plagio, haciendo que sea una elección más efectiva y confiable para la detección de plagio.

Referencias:

[1] G. Lee, J. Kim, M.-s. Choi, R.-Y. Jang, and R. Lee, "Review of Code Similarity and Plagiarism Detection Research Studies," Appl. Sci., vol. 13, no. 20, p. 11358, Oct. 2023, doi: 10.3390/app132011358.

[2] M. Duracik, P. Hrkut, E. Krsak, and S. Toth, "Abstract Syntax Tree Based Source Code Antiplagiarism System for Large Projects Set," IEEE Access, vol. 8, pp. 178750-178763, 2020, doi: 10.1109/ACCESS.2020.3026422.

[3] S. E. Robertson, "Understanding Inverse Document Frequency: On Theoretical Arguments for IDF," Journal of Documentation, vol. 60, no. 5, pp. 503-520, Oct. 2004, doi: 10.1108/00220410410560582.

[4] A. K. Dipongkor, R. Islam, M. Shafiuzzaman, M. A. Nashiry, S. M. Galib, and K. M. Mazumder, "AcPgChecker: Detection of Plagiarism among Academic and Scientific Writings," in 2021 Joint 10th International Conference on Informatics, Electronics & Vision (ICIEV) and 2021 5th International Conference on Imaging, Vision & Pattern Recognition (icIVPR), Kitakyushu, Japan, 2021, pp. 1-5. doi: 10.1109/ICIEVicIVPR52578.2021.9534100.

[5] K. Li, W. Zhang, Q. Lu, and X. Fang, "An Improved SMOTE Imbalanced Data Classification Method Based on Support Degree," in 2014 International Conference on Identification, Information and Knowledge in the Internet of Things, Beijing, China, 2014, pp. 35-38. doi: 10.1109/IIKI.2014.14.

[6] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," presented at the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16), San Francisco, CA, USA, Aug. 2016, pp. 785-794. doi: 10.1145/2939672.2939785.

[7] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," arXiv preprint, arXiv:1603.02754, Mar. 2016, doi: 10.48550/arXiv.1603.02754.

[8] R. S. Mehsen y H. D. Joshi, "Source Code Plagiarism Detection using Random Forest Classifier," en 4th International Conference on Communication Engineering and Computer Science (CIC-COCOS'2022), 2022, pp. 69-75. doi: 10.24086/cocos2022/paper.732.

[9] A. Gholamy, V. Kreinovich, and O. Kosheleva, "Why 70/30 or 80/20 Relation Between Training and Testing Sets: A Pedagogical Explanation," Departmental Technical Reports (CS), University of Texas at El Paso, TX, USA, Feb. 2018.

¡Muchas gracias!