Clasificación Multiclase - Glass Classification

Gabriela Gallegos Rubio - ggallegosr@unal.edu.co
2024 - II
Universidad Nacional de Colombia - Introducción a Sistemas Inteligentes

Abstract

El proyecto presenta diferentes algoritmos de aprendizaje supervisado para la tarea de clasificación, se realiza el análisis a partir de los resultados de cada técnica para definir cual es el mejor, de acuerdo al conjunto de datos, sin embargo, el conjunto de datos *Glass* está desbalanceado por lo que se realiza la normalización para todo el conjunto y oversampling en las clases con menor número de muestras, para evidenciar la importancia del procesamiento de los datos y como repercute en las previsiones de las técnicas, se desarrolla el contraste de los resultados cuando los datos no se preprocesan, normalizados y normalizados con oversampling.

Palabras clave: Aprendizaje supervisado,clasificación multiclase, oversampling, matriz de confusión.

1 Introducción

El crecimiento exponencial de los datos a nivel global ha generado que se consideran activos al permitir la toma de decisiones mediante la predicción de su comportamiento. El Machine Learning se enfoca en el entrenamiento de algoritmos de aprendizaje para lograr que se adapte y pueda cambiar con relación a los datos que se le van suministrando con el histórico de datos.[1] Los datos pueden ser etiquetados o desconocidos, en el mismo orden hacen referencia a el aprendizaje supervisado y aprendizaje no supervisado.

El proyecto maneja un enfoque hacia el aprendizaje supervisado, que tiene dos subcategorías: regresión y clasificación; el aprendizaje supervisado con clasificación compete a la relación funcional entre los input y output, es decir se convierte en un modelo predictivo con relación a que clase (etiquetas) pertenecen esos datos. El modelo se entrena con los datos designados al *"training"* normalmente el 80 % y se evalúa con los datos de prueba antes de realizar previsiones con el conjunto de datos reservados que no han sido vistos.

Las técnicas empleadas en el proyecto se eligen de acuerdo al tipo de tarea (Clasificación) por medio de aprendices ansiosos y vagos. El conjunto de datos asignado es Glass que contiene información para la clasificación de diferentes tipos de vidrio según su composición (elementos), el conjunto de datos tiene 214 muestras, que a su vez cada una contiene 9 atributos que reflejan sus composición, además de el índice de refracción del vidrio [2].

2 Preprocesamiento de datos

El conjunto de datos Glass es utilizado para la clasificación de diferentes tipos de vidrio, con base en su composición química,cada muestra contiene la concentración de varios elementos como sodio, magnesio y silicio, los cuales permiten distinguir entre diferentes clases de vidrio (6 clases). El dataset Glass cuenta con 214 instancias y 10 atributos continuos que representan la concentración de elementos químicos. La variable objetivo está representada por la columna Type, la cual indica la clase de vidrio; es la variable importante a predecir para los modelos.

#	Columna	Descripción	Not-Null Count	Dtype
0	RI	Índice de Refracción	214 non-null	float64
1	Na	Sodio	214 non-null	float64
2	Mg	Magnesio	214 non-null	float64
3	Al	Aluminio	214 non-null	float64
4	Si	Silicio	214 non-null	float64
5	K	Potasio	214 non-null	float64
6	Ca	Calcio	214 non-null	float64
7	Ba	Bario	214 non-null	float64
8	Fe	Hierro	214 non-null	float64
9	Type	Clase de vidrio	214 non-null	int64

Table 1: Descripción de las columnas del dataset Glass

De la tabla anterior se destaca que no se presentan valores nulos en el conjunto de datos, sin embargo es un dataset muy pequeño. Se presentan 6 tipos de vidrio que son las variables objetivo, mediante los modelos se desea obtener una matriz de confusión que presente los valores estimados de manera correcta y los equivocados, con ello se consigue la exactitud (accuracy) del modelo de acuerdo a la técnica.

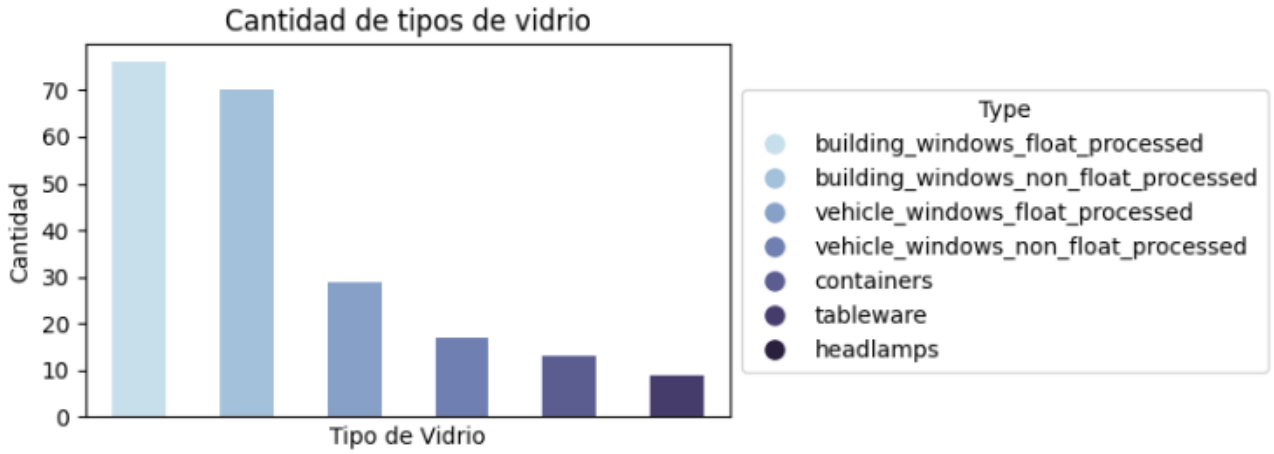


Figure 1: Cantidad de tipos de vidrio

La figura 1 evidencia que existen clases dominantes en el dataset por lo cual el patrón de la clase dom-

inante supera de manera drástica al tipo con menor frecuencia, es decir, que el conjunto de datos está desbalanceado.

2.1 Análisis exploratorio

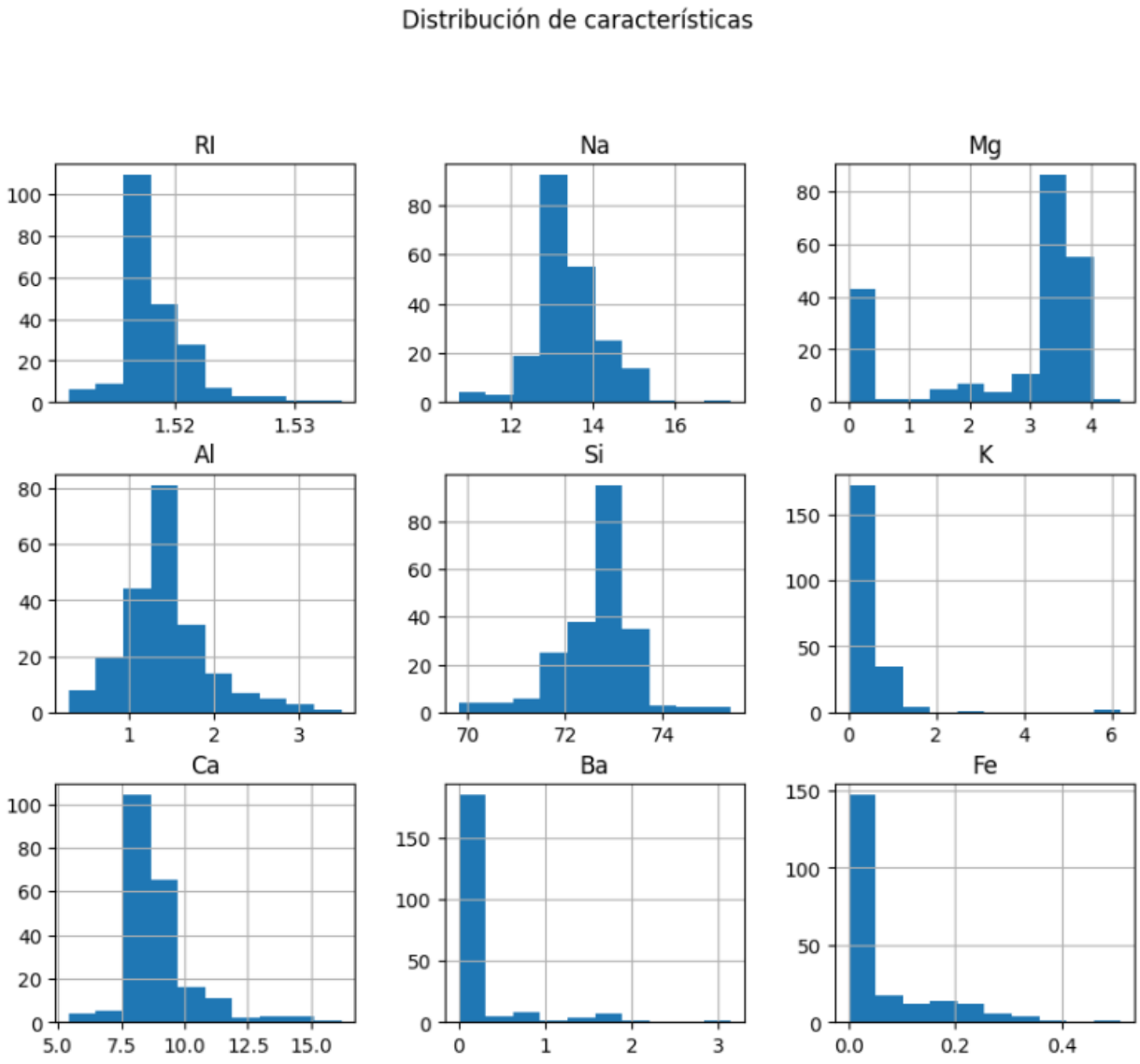


Figure 2: Distribución de características - Glass dataset

Los histogramas de la distribución de las características en la figura 2 presenta distribuciones sesgadas a izquierda o derecha , además las características Fe, K y Ba tienen sesgos hacia valores muy bajos, otras características presentan concentraciones medias significativas; por ejemplo, Si, Na y Ca. Lo anterior permite plantear el supuesto de que por la variabilidad de los datos se pueden presentar outliers en ciertas características, más predominantes en unas que en otras.

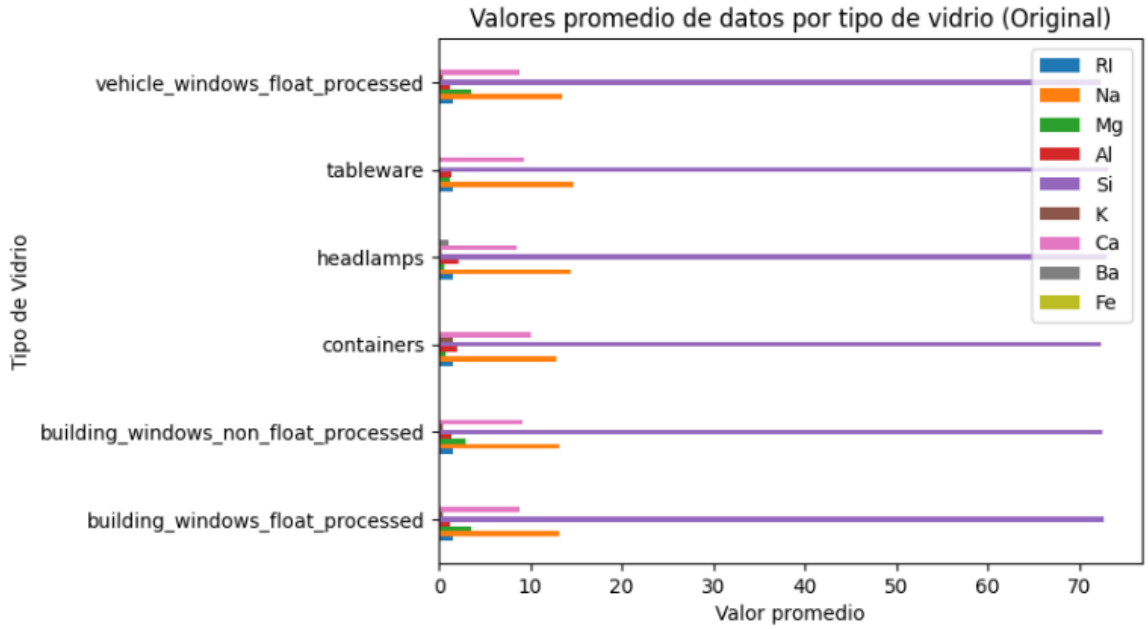


Figure 3: Valores Promedios dataset original

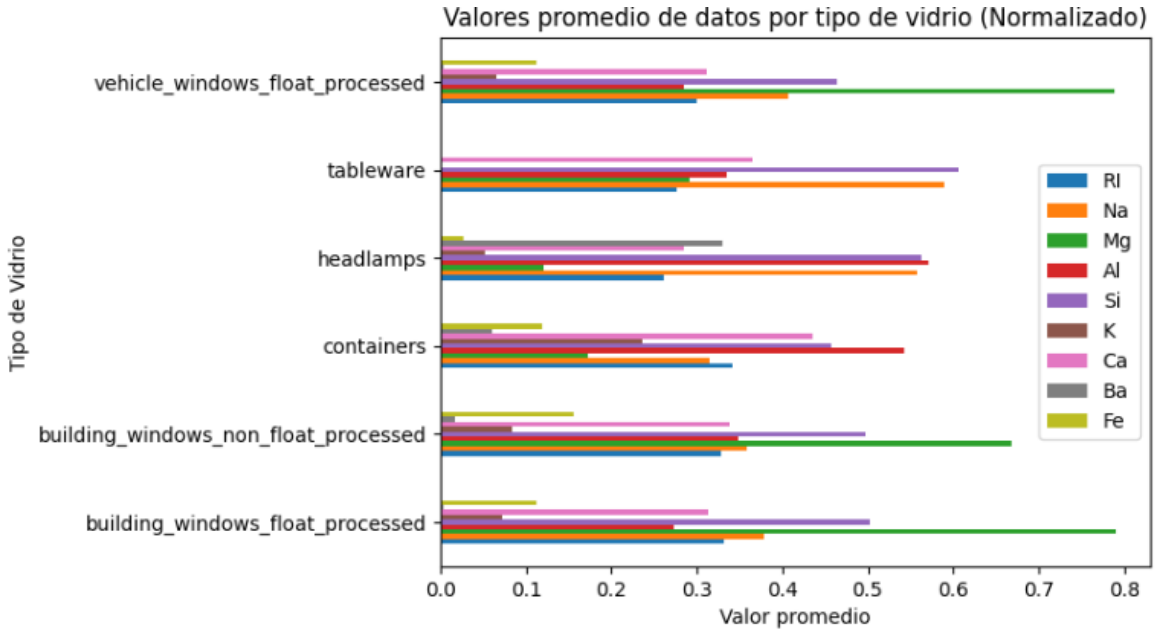


Figure 4: Valores Promedios dataset normalizado

Las figuras 3 y 4 se confirma que existen características predominantes en la composición de los tipos de vidrio, se normalizan los datos con el propósito de realizar un análisis en un conjunto de datos más equilibrado (a través de MinMaxScaler() que escala los datos al rango [0, 1]), por lo tanto, equitativo sin perder el comportamiento entre clases. Como estrategia para balancear las clases minoritarias, se decide implementar *Oversampling*, esta técnica reside en aumentar el número de muestras de las clases con menor frecuencia mediante datos sintéticos, estos datos se crean a partir de la información existente sin que sean idénticos pero si cercanos, es una buena técnica por lo que no se pierde o reduce la información.

En la revisión de la técnica de Oversampling se decide implementar *Borderline-SMOTE* para minimizar el problema del ruido en el dataset mediante la creación de datos más difíciles," Los puntos de datos "más difíciles" son puntos de datos cercanos al límite de decisión y, por lo tanto, más difíciles de clasificar. Estos puntos más difíciles son más útiles para que el modelo aprenda."[3]. Con lo cual permite clasificar ciertos datos de la clase minoritaria en una zona de peligro al estar cerca de datos de la clase mayoritaria. Se decide aumentan las clases minoritarias en un 50% para minimizar sesgos y sobreajuste, obteniendo un dataset mejor balanceado sin exceso de datos sintéticos y siguiendo la estructura original del conjunto, el conjunto de datos con oversampling cuenta con 365 instancias.

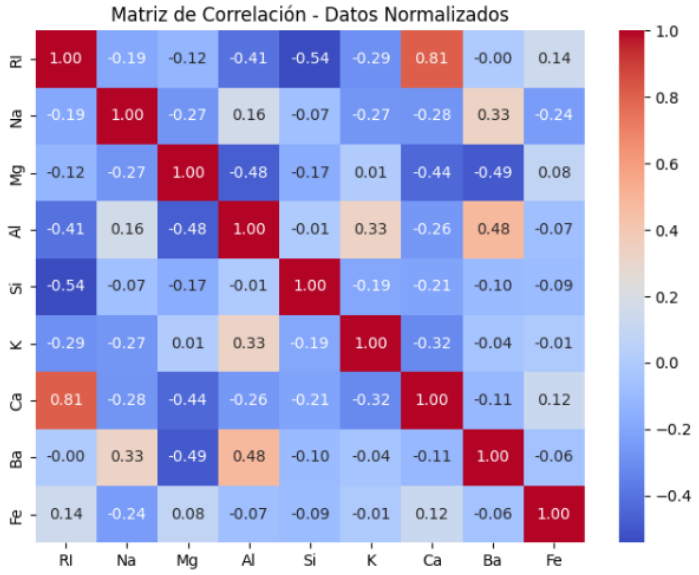


Figure 5: Matriz de correlación de Datos Normalizados

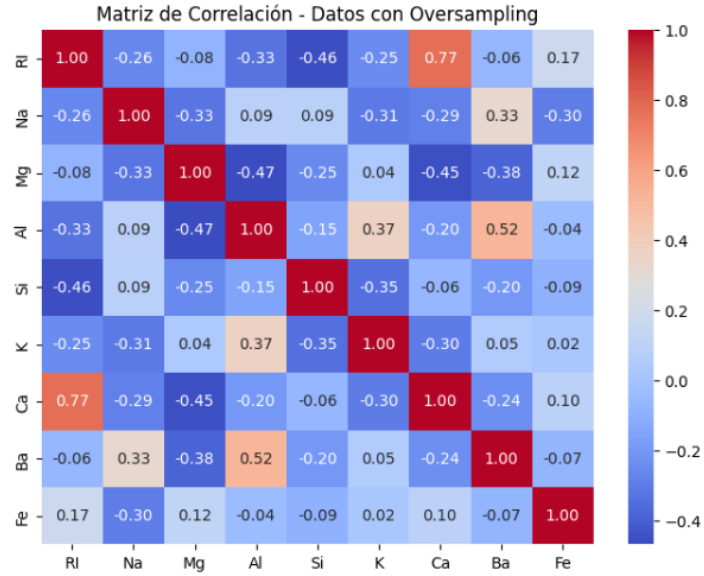


Figure 6: Matriz de correlación de Datos Normalizados con Oversampling

Se aplica un análisis de correlaciones mediante la correlación de Pearson al ser el modelo estándar, el

cual permite mediar la relación lineal entre las características y con ello identificar las correlaciones altas y bajas, la matriz de correlación de los datos en crudo y los datos normalizados es la misma por lo que MinMaxScaler() preserva las relaciones lineales por lo que no cambian tras normalizar.

Por ejemplo, las características RI y Ca están altamente correlacionadas positivamente (o negativamente, según el valor específico), lo que indica una relación lineal fuerte. Los valores de correlación cercanos a cero reflejan una menor fuerza de relación lineal, mientras que un signo negativo en el valor de la correlación indica una pendiente negativa en la relación lineal, es decir, que cuando una característica aumenta, la otra tiende a disminuir[4]. Sin embargo, una correlación alta no implica necesariamente que sean redundantes, ya que ambas podrían aportar información complementaria para clasificar los tipos de vidrio, dependiendo de su contribución a las diferencias entre clases.

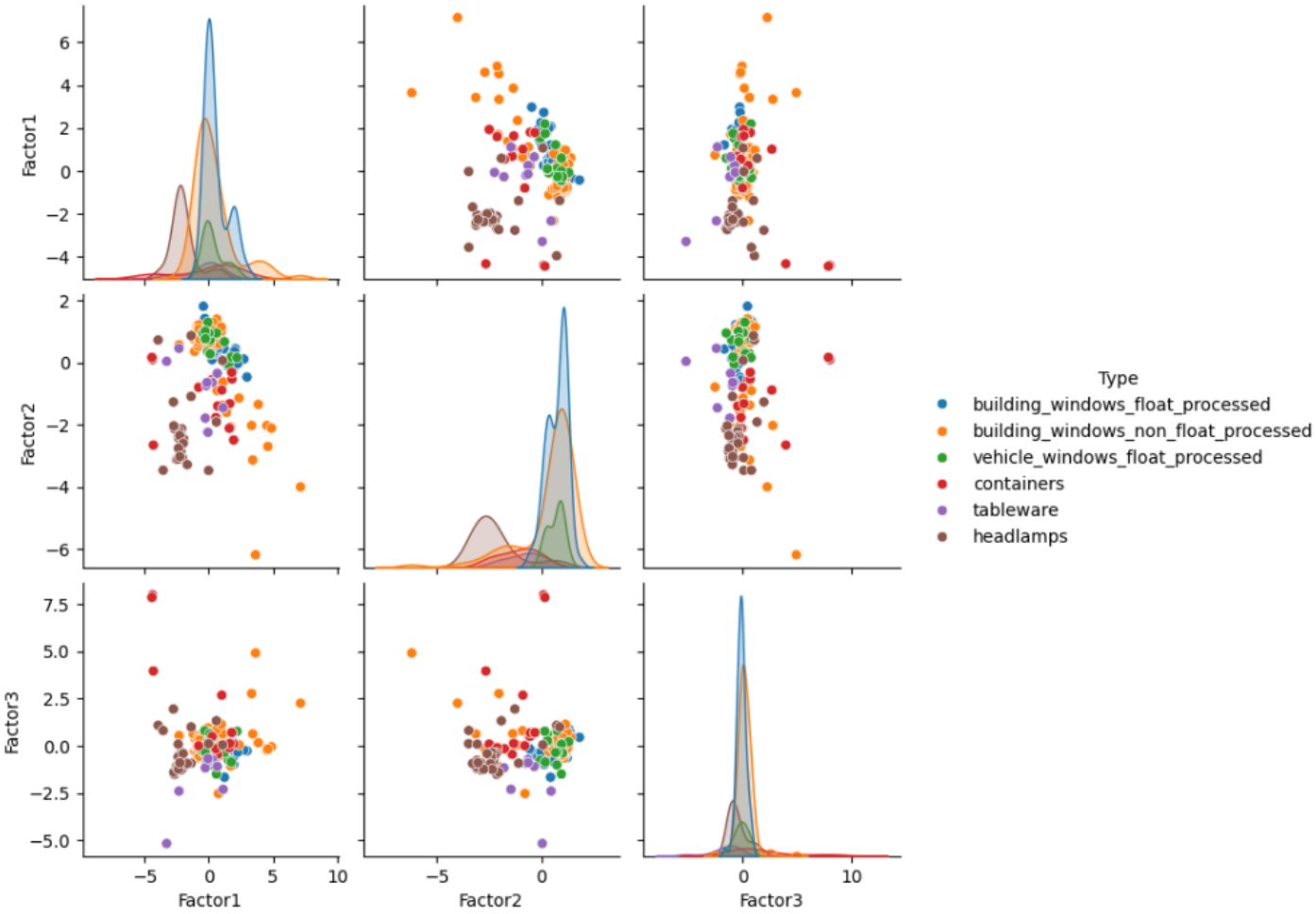


Figure 7: Análisis de componentes principales (PCA)

Ahora bien, el análisis de componentes principales (PCA) reduce la dimensión para reconocer los componentes que son capaces de explicar la mayor varianza en los datos, en las clases con mayor número de muestras como "building_windows_float_processed" y "building_windows_non_float_processed" están distanciadas en el Factor1 y Factor2, por el contrario el resto de clases son minoritarias y se solapan bastante por lo que evidencia que carecen de variabilidad. Se podrían destacar los siguientes aspectos:

- Los puntos aislados y las distribuciones sesgadas a izquierda o derecha podrían ser un supuesto de outliers.

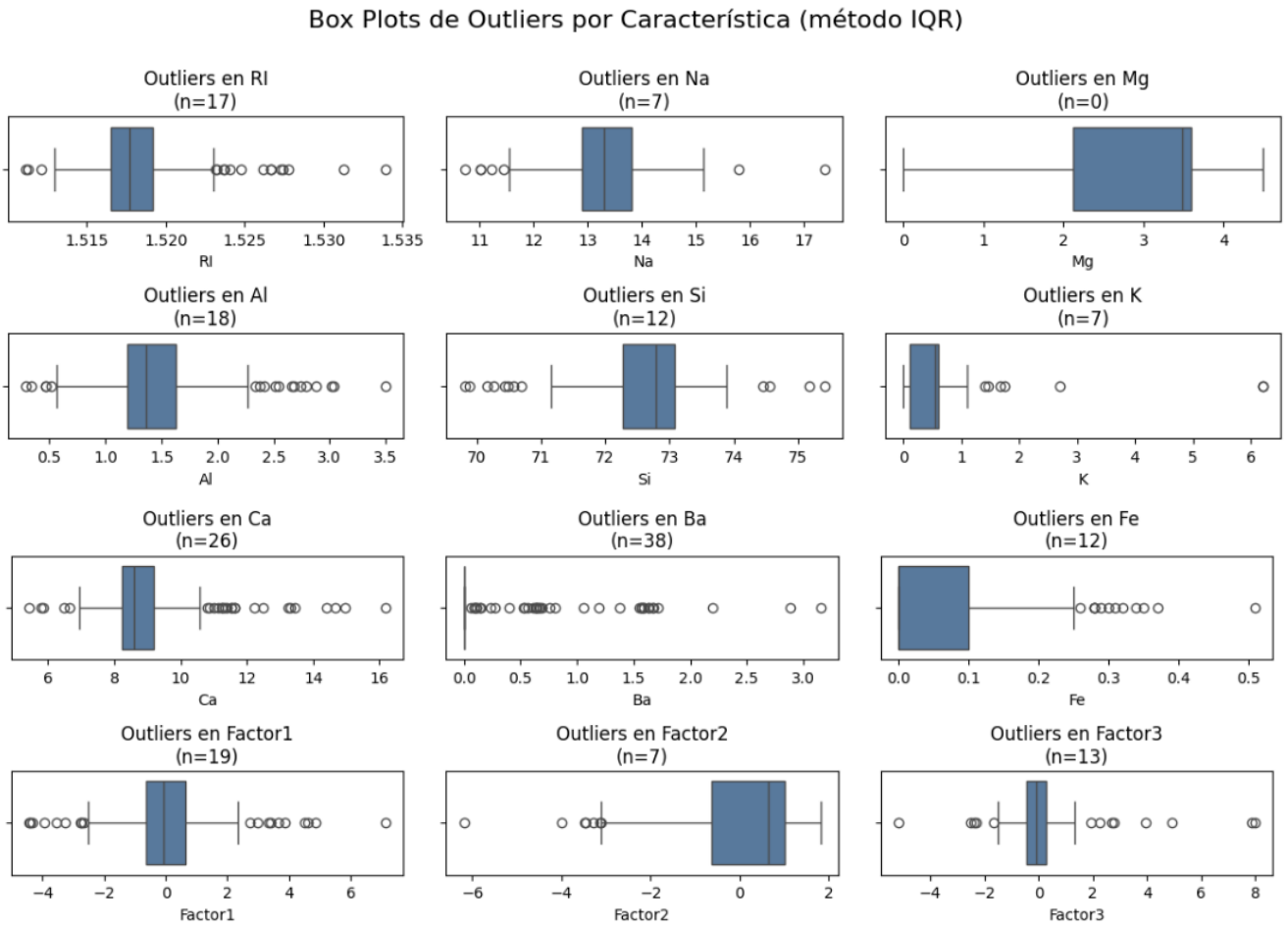


Figure 8: Box Plots de Outliers por característica (PCA)

La mayoría de la características de los box plots anteriores presentan valores atípicos excepto Mg, la variable Ba presenta 38 outliers con lo cual podría inferirse que es una variable que produce bastante ruido en el modelo por lo que podría desviar de forma significativa las predicciones.

2.2 Enfoque propuesto

Se implementan 7 técnicas de clasificación para el dataset Glass, se preprocesan los datos pero no se genera limpieza por lo que se presenta baja cantidad de muestras en el conjunto, por lo que quitar información que tenemos podría resultar en un sesgo. Para resaltar la necesidad del preprocesamiento de los datos, a cada técnica se le pasan 3 dataset que son:

- Datos en Crudo : original_dataframe, Datos Normalizados : data_normalized y Datos Normalizados y con Oversampling : data_oversampled

Con este enfoque se logra realizar el contraste entre cada conjunto de datos, además de realizar el contraste entre las técnicas implementadas y con eso saber cual es la que proporciona mejor clasificación del tipo

de vidrio.

3 Experimentación

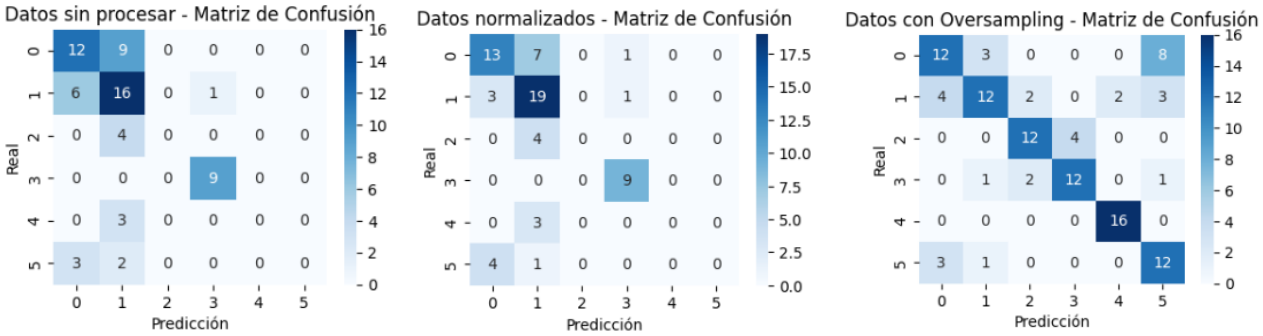
Por cada técnica de clasificación se describe su enfoque y principales características, se presenta la matriz de confusión, se elige mostrar la matriz para saber que tan preciso fue el modelo para cada uno de los conjuntos de datos.

- **Matriz de confusión:** "Una matriz de confusión (o matriz de error) es un método de visualización para los resultados del algoritmo clasificador. Más específicamente, es una tabla que desglosa el número de instancias de verdad fundamental de una clase específica frente al número de instancias de clase previstas"[5].

3.1 Árbol de decisión simple

3.1.1 Descripción:

El árbol de decisión simple, implementado mediante *DecisionTreeClassifier* de scikit-learn, es un modelo basado en reglas jerárquicas que divide recursivamente los datos en subgrupos según las características más informativas, seleccionadas mediante criterios como la ganancia de información o el índice Gini. Su enfoque es intuitivo y no necesita que se le pasen los datos normalizados.



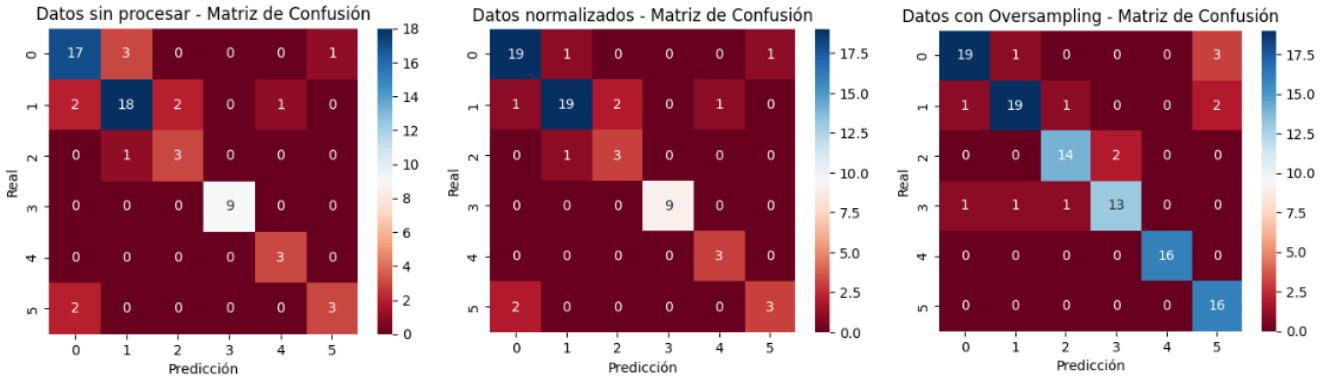
((a)) Datos sin procesar ((b)) Datos Normalizados ((c)) Datos con Oversampling

Figure 9: Conjuntos de datos para árbol de decisión simple - Matriz de confusión

3.2 Random Forest SKlearn

3.2.1 Descripción:

El Random Forest de scikit-learn, basado en*RandomForestClassifier*, es un ensamblado de múltiples árboles de decisión que mejora la robustez y reduce el sobreajuste mediante bagging y selección aleatoria de características.Es menos sensible a outliers que un árbol simple,se configura con parámetros como *n_estimators* y *max_depth* para optimizar el rendimiento en datasets desbalanceados.



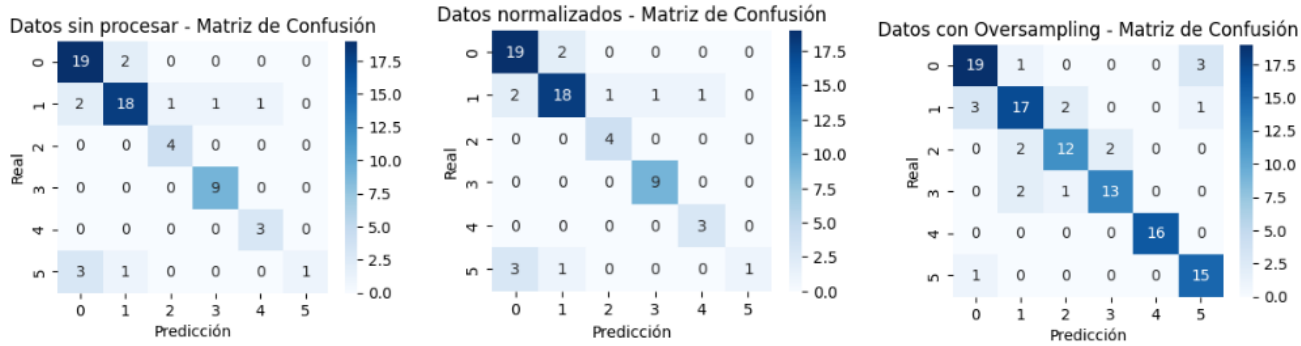
((a)) Datos sin procesar ((b)) Datos Normalizados ((c)) Datos con Oversampling

Figure 10: Conjuntos de datos para Random Forest SKlearn - Matriz de confusión

3.3 Random Forest XGBoost

3.3.1 Descripción:

El Random Forest en XGBoost, basado en *XGBRFClassifier*, es una implementación optimizada de Random Forest que utiliza árboles de decisión con gradiente boosting. Clasifica clases optimizando funciones de pérdida con regularización (parámetros *lambda* y *alpha*), manejando desbalance mediante pesos de clase o *scale_pos_weight*.



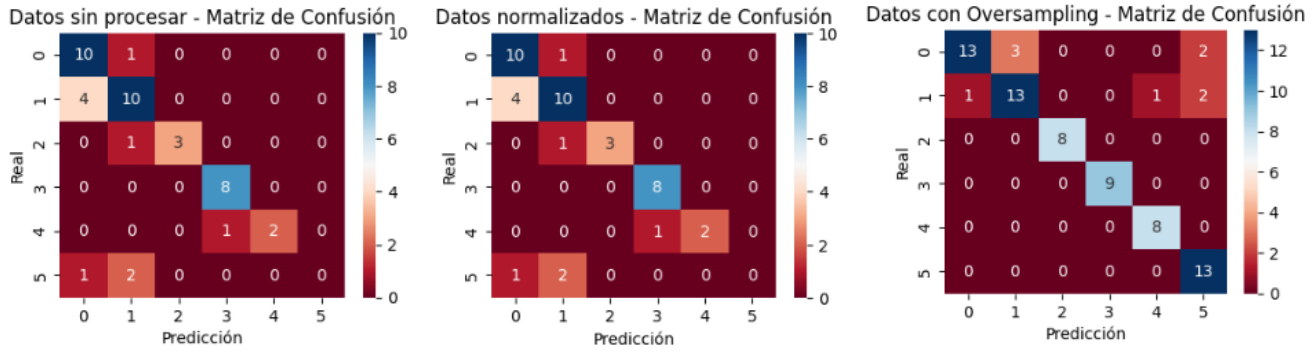
((a)) Datos sin procesar ((b)) Datos Normalizados ((c)) Datos con Oversampling

Figure 11: Conjuntos de datos para Random Forest XGBoost - Matriz de confusión

3.4 Red Neuronal

3.4.1 Descripción:

La red neuronal, implementada con *MLPClassifier* de scikit-learn, es un modelo basado en capas de neuronas que aprende patrones no lineales . Utiliza activaciones como ReLU y optimización con gradiente descendente. Es efectiva para capturar relaciones complejas, pero puede sobreajustar sin regularización.



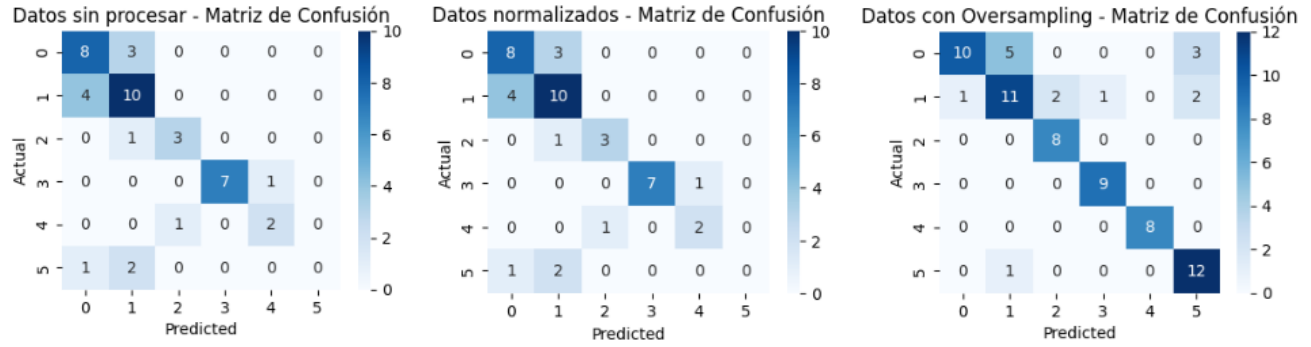
((a)) Datos sin procesar ((b)) Datos Normalizados ((c)) Datos con Oversampling

Figure 12: Conjuntos de datos para Red Neuronal - Matriz de confusión

3.5 KNN (k-Nearest Neighbors)

3.5.1 Descripción:

Esta técnica se implementa con *KNeighborsClassifier* de scikit-learn, clasifica de acuerdo en la distancia euclidiana (o Manhattan) a los k vecinos más cercanos en el espacio de características. No requiere entrenamiento explícito, pero es sensible a la escala y resulta efectivo para conjuntos pequeños.



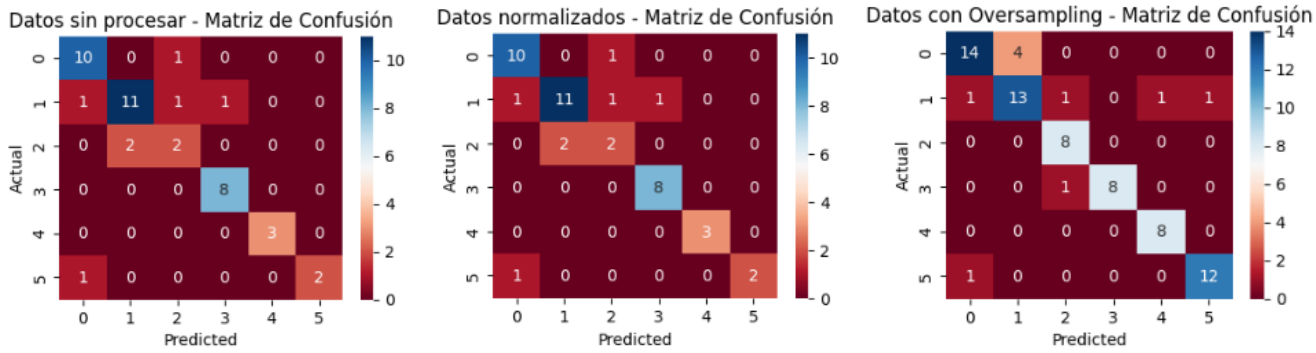
((a)) Datos sin procesar ((b)) Datos Normalizados ((c)) Datos con Oversampling

Figure 13: Conjuntos de datos para KNN - Matriz de confusión

3.6 Classifier XGBoost

3.6.1 Descripción:

El *XGBClassifier* de XGBoost es un modelo de gradient boosting que combina árboles de decisión secuenciales, optimizando funciones de pérdida con regularización (parámetros *learning_rate*, *max_depth*, *lambda*). Clasifica tipos de vidrio manejando desbalance con *scale_pos_weight* o pesos de clase, siendo eficiente para datasets como Glass tras normalización y oversampling.



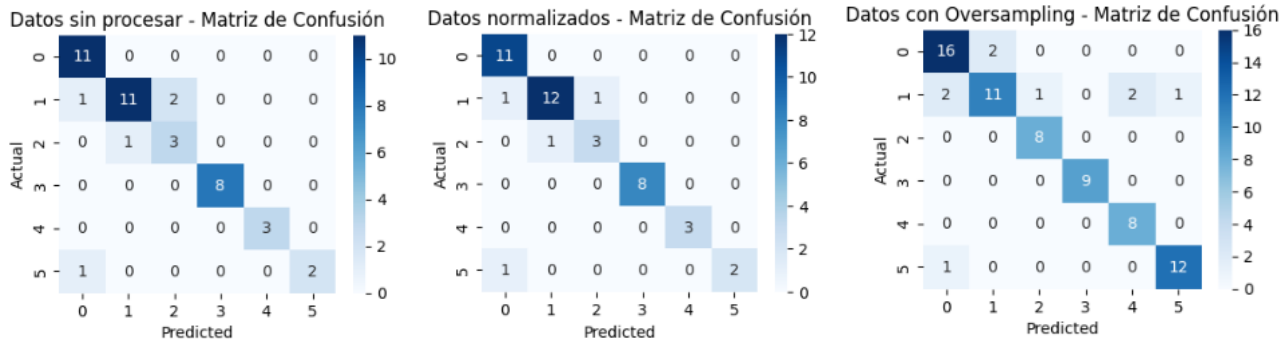
((a)) Datos sin procesar ((b)) Datos Normalizados ((c)) Datos con Oversampling

Figure 14: Conjuntos de datos para Classifier XGBoost - Matriz de confusión

3.7 CatBoost

3.7.1 Descripción:

La técnica de CatBoost, implementado con *CatBoostClassifier*, es un modelo de gradient boosting optimizado para datasets con características categóricas y numéricas, combinando árboles con regularización y manejo automático de desbalance, es robusto a outliers.



((a)) Datos sin procesar ((b)) Datos Normalizados ((c)) Datos con Oversampling

Figure 15: Conjuntos de datos para CatBoost - Matriz de confusión

4 Resultados y análisis

4.1 Resultados

De cada una de las técnicas utilizadas se escoge el dataset que tuvo mayor rendimiento y exactitud en la predicción de los tipos de vidrio (se remarca el conjunto destacado), para evidenciar el contraste entre las técnicas se presenta el reporte de clasificación de las mismas y con ello poder hacer inferencias.

Árbol de decisión simple

Métrica	Datos sin procesar	Datos normalizados	Datos con Oversampling
Accuracy	0.63	0.63	**0.69**
Macro avg f1-score	0.37	0.37	**0.70**
Recall promedio	0.41	0.41	**0.72**

Table 2: Comparación de Métricas para Árbol de Decisión Simple

Classification Report	Precision	Recall	F1-score	Support
building_windows_float_processed	0.63	0.52	0.57	23
building_windows_non_float_processed	0.71	0.52	0.60	23
containers	0.75	0.75	0.75	16
headlamps	0.75	0.75	0.75	16
tableware	0.89	1.00	0.94	16
vehicle_windows_float_processed	0.50	0.75	0.60	16
accuracy			0.69	110
macro avg	0.70	0.72	0.70	110
weighted avg	0.70	0.69	0.69	110

Table 3: Reporte de clasificación árbol de decisión simple - Datos normalizados con oversampling

Random Forest SKlearn

Métrica	Datos sin procesar	Datos normalizados	Datos con Oversampling
Accuracy	0.82	0.82	**0.88**
Macro avg f1-score	0.75	0.75	**0.88**
Recall promedio	0.76	0.76	**0.88**

Table 4: Comparación de Métricas para Random Forest con SKLearn

Classification Report	Precision	Recall	F1-score	Support
building_windows_float_processed	0.90	0.83	0.86	23
building_windows_non_float_processed	0.90	0.83	0.86	23
containers	0.88	0.88	0.88	16
headlamps	0.87	0.81	0.84	16
tableware	1.00	1.00	1.00	16
vehicle_windows_float_processed	0.76	1.00	0.86	16
accuracy			0.88	110
macro avg	0.89	0.89	0.88	110
weighted avg	0.89	0.88	0.88	110

Table 5: Reporte de clasificación Random Forest SK - Datos normalizados con oversampling

Random Forest XGBoost

Métrica	Datos sin procesar	Datos normalizados	Datos con Oversampling
Accuracy	0.83	0.83	**0.84**
Macro avg f1-score	0.78	0.78	**0.84**
Recall promedio	0.81	0.81	**0.84**

Table 6: Comparación de Métricas para Random Forest con XGBoost

Classification Report	Precision	Recall	F1-score	Support
building_windows_float_processed	0.83	0.83	0.83	23
building_windows_non_float_processed	0.77	0.74	0.76	23
containers	0.80	0.75	0.77	16
headlamps	0.87	0.81	0.84	16
tableware	1.00	1.00	1.00	16
vehicle_windows_float_processed	0.79	0.94	0.86	16
accuracy			0.84	110
macro avg	0.84	0.84	0.84	110
weighted avg	0.84	0.84	0.84	110

Table 7: Reporte de clasificación Random Forest XGBoost - Datos normalizados con oversampling

Red Neuronal

Métrica	Datos sin procesar	Datos normalizados	Datos con Oversampling
Accuracy	0.77	0.77	**0.81**
Macro avg f1-score	0.76	0.76	**0.83**
Recall promedio	0.80	0.80	**0.83**

Table 8: Comparación de Métricas para la Red Neuronal

KNN (k-Nearest Neighbors)

Métrica	Datos sin procesar	Datos normalizados	Datos con Oversampling
Accuracy	0.70	0.70	**0.79**
Macro avg f1-score	0.61	0.61	**0.83**
Recall promedio	0.62	0.62	**0.85**

Table 9: Comparación de Métricas para KNN

Classification Report	Precision	Recall	F1-score	Support
building_windows_float_processed	0.91	0.56	0.69	18
building_windows_non_float_processed	0.65	0.65	0.65	17
containers	0.80	1.00	0.89	8
headlamps	0.90	1.00	0.95	9
tableware	1.00	1.00	1.00	8
vehicle_windows_float_processed	0.71	0.92	0.80	13
accuracy			0.79	73
macro avg	0.83	0.85	0.83	73
weighted avg	0.81	0.79	0.79	73

Table 10: Reporte de clasificación KNN - Datos normalizados con oversampling

Classifier XGBoost

Métrica	Datos sin procesar	Datos normalizados	Datos con Oversampling
Accuracy	0.84	0.84	**0.86**
Macro avg f1-score	0.82	0.82	**0.88**
Recall promedio	0.81	0.81	**0.89**

Table 11: Comparación de Métricas para XGBoost Classifier

Classification Report	Precision	Recall	F1-score	Support
building_windows_float_processed	0.88	0.78	0.82	18
building_windows_non_float_processed	0.76	0.76	0.76	17
containers	0.80	1.00	0.89	8
headlamps	1.00	0.89	0.94	9
tableware	0.89	1.00	0.94	8
vehicle_windows_float_processed	0.92	0.92	0.92	13
accuracy			0.86	73
macro avg	0.88	0.89	0.88	73
weighted avg	0.87	0.86	0.86	73

Table 12: Reporte de clasificación XGBoost Classifier - Datos normalizados con oversampling

CatBoost

Métrica	Datos sin procesar	Datos normalizados	Datos con Oversampling
Accuracy	0.88	**0.91**	0.88
Macro avg f1-score	0.87	**0.89**	0.89
Recall promedio	0.87	0.88	**0.91**

Table 13: Comparación de Métricas para CatBoost

Classification Report	Precision	Recall	F1-score	Support
building_windows_float_processed	0.85	1.00	0.92	11
building_windows_non_float_processed	0.92	0.86	0.89	14
containers	0.75	0.75	0.75	4
headlamps	1.00	1.00	1.00	8
tableware	1.00	1.00	1.00	3
vehicle_windows_float_processed	1.00	0.67	0.80	3
accuracy			0.91	43
macro avg	0.92	0.88	0.89	43
weight avg	0.88	0.88	0.87	73

Table 14: Reporte de clasificación CatBoost - Datos normalizados

4.2 Análisis

A partir de los experimentos realizados con las distintas técnicas de clasificación, se pueden resaltar los siguientes puntos:

- **Análisis exploratorio:** Es una etapa fundamental por lo que permite conocer el conjunto de datos, visualizar los comportamientos de las características que componen los tipos de vidrio, a su vez detectar los sesgos presentes y con eso conocer si el conjunto está o no desbalanceado. El análisis de los valores atípicos permite conocer como es la tendencia de las muestras sobre la característica.

- **Importancia del preprocesamiento de datos:** Al conocer los comportamientos predominantes en el dataset se decide si se aplican técnicas de normalizar o de submuestreo; se eligen de acuerdo a la información disponible del data set, no se aplicó undersampling por lo que el número de muestras es muy pequeño (201) y no conviene eliminar información vital en el modelo.

- **Impacto del sobremuestreo:** La técnica de sobremuestreo aplicada con *SMOTE* demostró ser una estrategia efectiva para mejorar las métricas de clasificación, por lo que en la mayoría de los resultados de las técnicas fue el conjunto de datos elegido. Por lo que se sugiere que la generación de datos sintéticos le permite al modelo aprender mejor las clases minoritarias, equilibrando el sesgo presente en la distribución original del dataset y obteniendo muy buenos porcentajes de predicción y exactitud.

- **Sesgo de la Distribución:** A pesar de las mejoras con el sobremuestreo, las clases con mayor cantidad de instancias como*building_windows_float_processed* y *building_windows_non_float_processed* siguen siendo las mejor clasificadas (clases mayoritarias). Las clases minoritarias, como *tableware* y *containers*, aunque mejoraron, mantienen un desempeño inferior debido a la naturaleza sesgada del dataset original. Esto refleja que el sesgo de la distribución inicial aún tiene incidencia en el modelo, aunque se haya mitigado parcialmente con el sobremuestreo, sin embargo, también se puede inferir que se realizó de manera correcta el sobremuestreo porque se conservó la tendencia del comportamiento del dataset.

- **Técnicas a destacar:**Aunque tanto Bagging como Boosting demostraron mejorar significativamente el rendimiento de los modelos, los resultados experimentales evidencian que el Boosting, especialmente con algoritmos como XGBoost y CatBoost, ofrece un mejor desempeño para este conjunto de datos. Sin embargo, también se evidencia que combinar ambas técnicas en un mismo modelo no garantiza una mejora adicional y, en algunos casos, puede generar redundancias o sobreajuste. Esta observación resalta la importancia de una selección cuidadosa de métodos en función del problema específico y la naturaleza de los datos.

5 Conclusiones

Los resultados obtenidos en este trabajo permiten concluir que la calidad y el balance del conjunto de datos influyen directamente en el rendimiento de los clasificadores. La aplicación de técnicas de sobremuestreo como SMOTE contribuyó a mitigar el sesgo de clase, mejorando la estabilidad de las métricas de evaluación. No obstante, se identificó que el sesgo original en la distribución de clases continúa afectando la capacidad de generalización de algunos modelos, es decir que se conservó la naturaleza del dataset.

Este trabajo investigativo permitió no solo aplicar técnicas convencionales de clasificación (Supervisadas), sino también explorar nuevos enfoques y herramientas, enriqueciendo el análisis y aportando evidencia comparativa valiosa. La implementación de diversos algoritmos logró cumplir con el objetivo principal: predecir los tipos de vidrio presentes en el conjunto de datos, reduciendo el sesgo sin comprometer la eficiencia ni sobrecargar el modelo. Además, se obtuvieron contrastes útiles entre métodos tradicionales y avanzados, reafirmando la relevancia del preprocesamiento y la selección algorítmica en problemas de clasificación real.

Referencias

[1] Acevedo.N, Vargas.C.(2017).Machine Learning: algoritmos de clasificación y sus aplicaciones en el análisis de datos. UNAM - UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO. Recuperado el 22 de febrero dehttps://ru.dgb.unam.mx/bitstream/20.500.14330/TES01000767007/3/0767007.pdf

[2] Anónimo. (2017).Glass Classification.UCI MACHINE LEARNING. Recuperado el 22 de febrero de https://www.kaggle.com/datasets/uciml/glass/data

[3] Murel J.(2024) Muestreo superior - ¿Qué es el sobremuestreo?. IBM Corporation. Recuperado el 22 de febrero de https://www.ibm.com/es-es/topics/upsampling

[4] DATAtab Team (2025). DATAtab: Online Statistics Calculator. DATAtab e.U. Graz, Austria. Recuperado el 25 de febrero dehttps://datatab.es/tutorial/correlation

[5] Murel J, Kavlakoglu, E.(2024) Matriz de confusión - ¿Qué es una matriz de confusión?. IBM Corporation.Recuperado el 25 de febrero dehttps://www.ibm.com/eses/topics/confusionmatrix: :text=Una