Evaluación de estrategias de sobremuestreo utilizando SMOTE para mejorar problemas de clasificación supervisada

Autor: Juan Manuel Natello.

Resumen. La presencia de clases desbalanceadas se convierte en un desafío para los modelos que asumen una distribución de datos relativamente equilibrada. Frente a este panorama, uno de los enfoques existentes en el aprendizaje automático es la utilización de técnicas de sobremuestreo. Asimismo, debido a que esta generación de nuevos ejemplos crea ruido y solapamiento, se aplica una posterior limpieza de datos, principalmente en la frontera de decisión. El objetivo de este trabajo es precisamente trabajar sobre este tipo de conjuntos desbalanceados y aplicarles estas técnicas de sobremuestreo y limpieza en pos de mejorar la performance de clasificadores como bosques aleatorios y clasificador bayesiano ingenuo.

1 Introducción

En aprendizaje automático la clasificación supervisada, tiene por objetivo el etiquetado de datos que responden a cuestiones tales como, ¿se dará de baja el cliente? ¿Tendrá cáncer un paciente? ¿Este cliente ha cometido algún fraude con tarjetas de crédito? ¿Este auto puede considerarse como aceptable? A este tipo de objetivo se lo define como del tipo categórico, ya que se le asigna una clase o categoría entre un número limitado de clases. A partir de allí, se define el conjunto con los parámetros de entrada que se establezcan, las cuales van a representar a las clases. Es entonces, donde se busca definir un patrón o lo que es lo mismo, la función que pueda representar mejor a cada instancia, esta depende en gran medida del volumen con que se la entrene, como así también de la calidad de las mismas. Una vez definida esta función, al ingresar una nuevo ejemplo sin etiquetar, el modelo es capaz de asignarle la clase adecuada. Sin embargo, en la realidad los conjuntos de datos que más nos interesan son aquellos que ocurren con poca frecuencia. Este es el conjunto de datos que se analizó en el presente trabajo y se los conoce como conjuntos de datos con clases desbalanceadas.

Dicha distribución de clases desbalanceada trae consecuencias en las predicciones que realiza el algoritmo de clasificación, las que pueden ser erróneas, ya que al tener tan pocos ejemplos de una, carece de antecedentes suficientes, para clasificar correctamente una instancia cuando la misma se le presenta. Asimismo, las métricas con que se mida al modelo pueden resultar engañosas y se tiene que prestar especial atención, para entender cuál usar, buscando que la misma sea lo suficientemente representativa del modelo de clasificación que se está evaluando.

Por último, y no menos importante, es necesario resaltar que un algoritmo que carezca de una predicción precisa y adecuada reduce drásticamente la confianza en el mismo, ya que en un ambiente donde la tolerancia a fallas es muy baja, su bajo rendimiento puede llevar a pérdidas significativas.

Una solución a este problema, es mediante el uso de técnicas de sobremuestreo que consiste en generar muestras sintéticas para emparejar las diferencias de la clase minoritaria con la que tiene mayor cantidad de ejemplos. Una de las herramientas más utilizadas para esta tarea es el sobremuestreo de minorías sintéticas conocido como smote [1] (por su sigla en inglés). este tipo de algoritmos permiten generar instancias sintéticas a partir de la incorporación de un error aleatorio y además existen variantes para aquellos ejemplos de la clase minoritaria más cercanos a la frontera de decisión borderline-smote [5] o también generar instancias sintéticas de forma adaptativa según una función de distribución. adasyn (por sus siglas, adaptative sampling approach for imbalance learning)[4].

Finalmente con estas herramientas es posible generar instancias sintéticas con diferentes variantes del algoritmo smote.
El objetivo de este trabajo es comparar el impacto de estas técnicas cuando se las combina en problemas de clasificación supervisada, comparando los resultados de la métrica Puntaje F1 con una evaluación cuantitativa y con datos independientes.

2 Materiales y Métodos

2.a Conjuntos de datos

Se utilizaron tres conjuntos de datos, de diferentes temáticas que presentan un marcado desbalanceo en sus variables objetivo (Tabla 2a). En cada conjunto de datos se observa que la clase minoritaria está presente en distinta magnitud que va desde 1.7% al 30%.

Tabla 2a Conjunto de Datos

Conjunto	Items / Clase	# Items / Clase	N
2a-1) Enfermedad de la Tiroides	2	0 = 3481 / 1 = 291	3772
2a-2) Evaluación del Automóvil	2	0 = 1210 / 1 = 518	1728
2a-3) Fraude Crediticio	2	0 = 24712 / 1 = 422	25134

Los datos de los datasets se encuentran referenciados en el Anexo del presente informe

2.a.1 LIMPIEZA DE DATOS

1) *Para el dataset de Enfermedad de la Tiroides*, convertimos variables nominales a numéricas 0 y 1; se realizó una imputación por la media a las variables con valores nulos;
2) *Para el dataset de Evaluación del Automóvil*, de la columna de clases se optó por reducir a valores de aceptable e inaceptable, y transformarlos a 0 y 1 respectivamente; Se convirtieron las variables Categóricas a Ordinales; Verificamos si hay variables con valores nulos.
3) *Para el dataset de Fraude Crediticio*, se realizó una transformación de los atributos nominales con solo dos estados, a valores de 0 y 1; Tenemos descriptores categóricos, por lo que podemos recodificar las variables categóricas en variables Dummy o Flags; Eliminamos variables no importantes; y Normalizamos por el método Min-Max, solo aquellas variables con valores mayores a 1.

2.b Herramientas de muestreo

En la etapa de sobremuestreo de datos se dividió en dos partes: creación de ejemplos artificiales mediante la técnica de sobremuestreo de SMOTE (Sobre-muestreo de minorías sintéticas), y una segunda etapa de limpieza. Para la etapa de creación de ejemplos sintéticos, el algoritmo de SMOTE

[4] sigue los siguientes pasos: sobre cada uno de los ejemplos de la clase minoritaria, se identifican los vecinos más cercanos (el cual se suele establecer en 3), mediante el uso de una medida de distancia (ejemplo distancia euclidiana); seguidamente se elige de manera aleatoria uno de estos vecinos, se calcula la diferencia entre ambos puntos y se multiplica esta diferencia por un número aleatorio entre 0 y 1. Finalmente, se suma este valor al ejemplo inicial, y se devuelve el nuevo punto siendo esta la muestra artificial.

Ejemplo de su uso (repo):

Imagen-2b.1

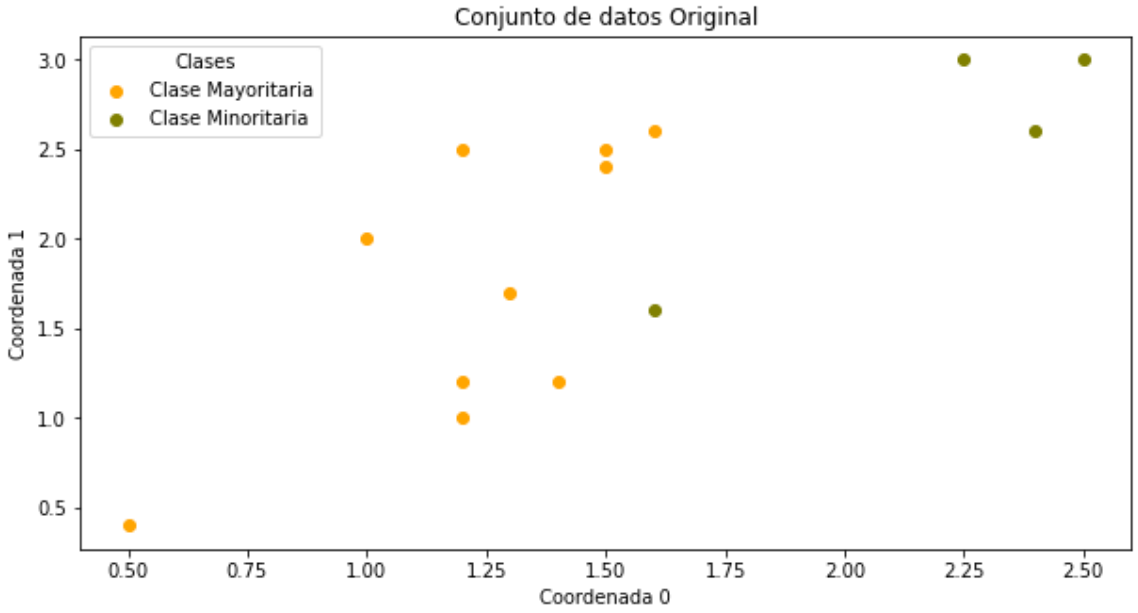
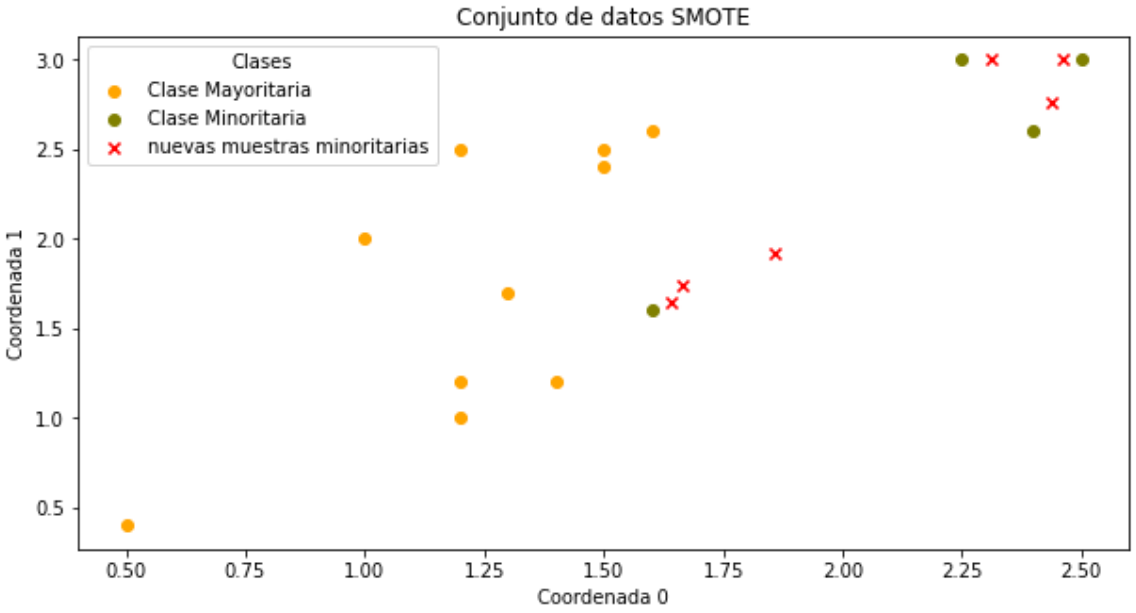


Imagen-2b.2



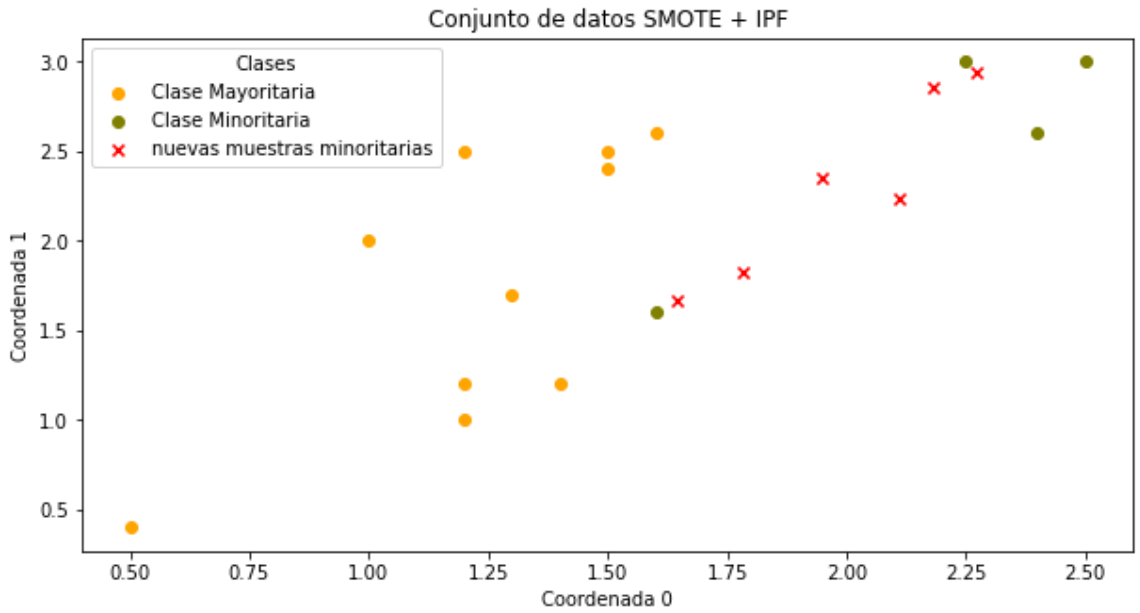
En la imagen-2b.2 se puede observar cómo se generan nuevos ejemplos sintéticos pertenecientes a la clase minoritaria (marcados con x). Asimismo es dable destacar que iguala ambos conjuntos en 10 ejemplos cada clase, generando un total de 6 nuevos ejemplos.

Ahora bien en la segunda etapa se usaron tres técnicas de eliminación de ruido, solapamiento y división mejorada de los conjuntos, eliminando aquellos cercanos al límite:

I. *IPF (Filtro de Partición Iterativa)[8]: El cual es un Filtro basado en conjuntos, donde se realizan n iteraciones con un criterio de parada n=3. Su algoritmo se puede describir en 4 pasos:*

1. Dividir el conjunto de datos de entrenamiento actual E en n subconjuntos de igual tamaño.
2. Crear un clasificador con el algoritmo C4.5 sobre cada uno de estos n subconjuntos de datos de entrenamiento actual E.
3. Agregar a A los ejemplos ruidosos identificados en E, usando un esquema de votación (consenso o mayoría).
4. Eliminar los ejemplos ruidosos $E <= E - A$.

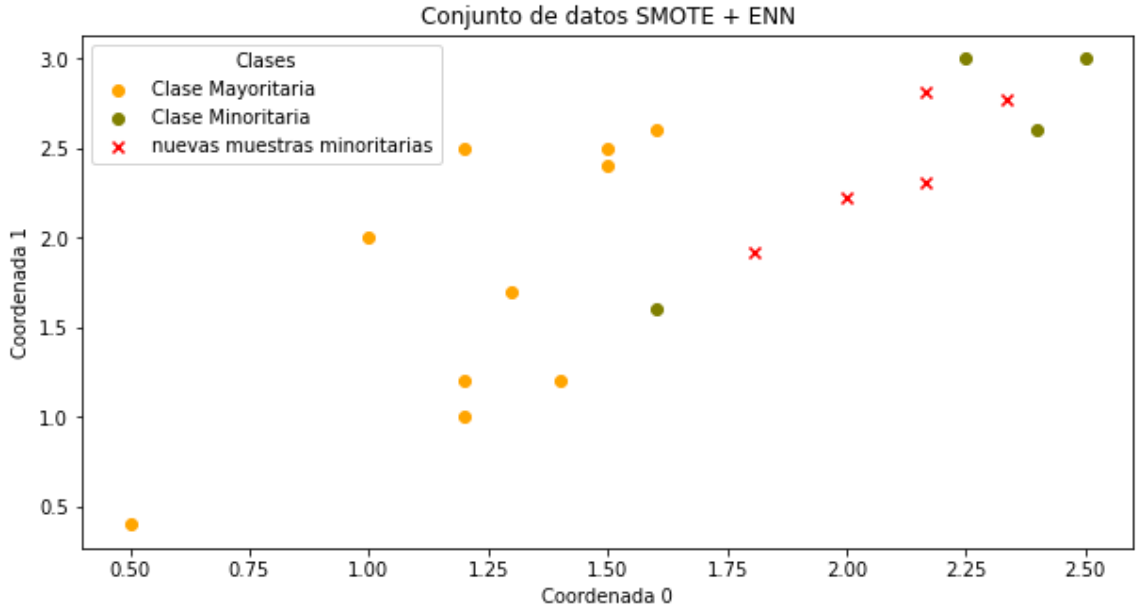
Imagen-2b.3



De la imagen 2b.3, es dable remarcar que, la función de limpieza *IPF* va a eliminar solo aquellos ejemplos donde SMOTE haya creado ejemplos artificiales que sean ruidosos, es por eso que en la imagen no se ven eliminados ninguno de los creados, quedando en igual proporción ambas clases.

II. ***ENN (Vecino más Cercano Editado)[5]:*** *consiste en eliminar instancias mal clasificadas de un conjunto de datos mediante la regla k-NN. Para identificar estas instancias el algoritmo elimina aquellas, cuya clase no coincide con la clase de la mayoría de sus vecinos.*

Imagen-2b.4

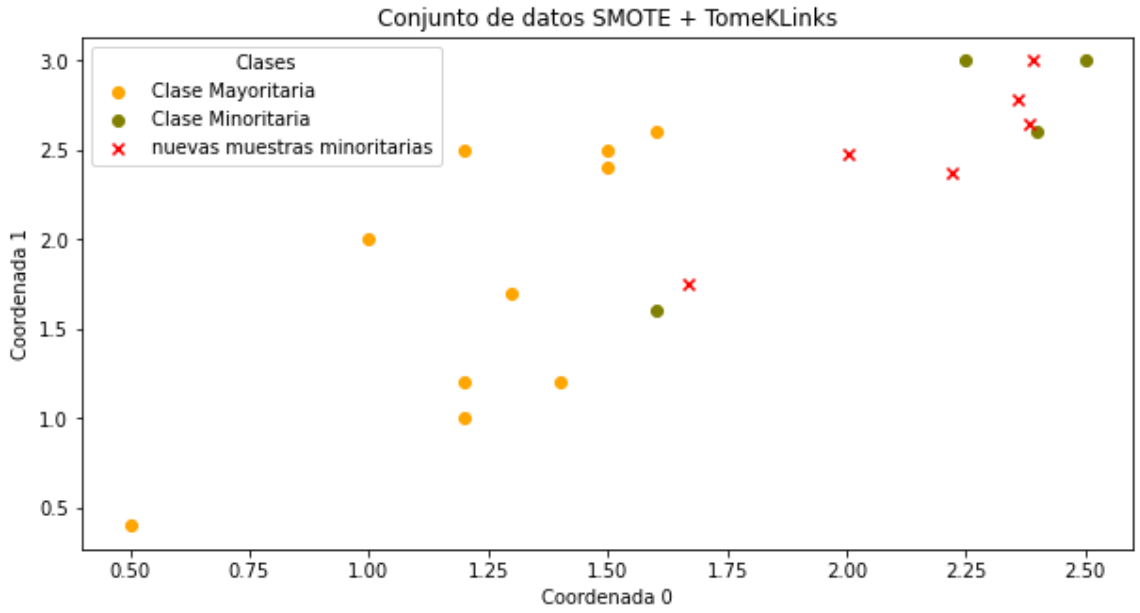


En este caso de la Imagen-2b.4, se observa que las clases no tienen las mismas cantidades, es decir, la clase minoritaria cuenta con 5 ejemplos artificiales, y eso es porque *ENN* eliminó 1 ejemplo donde sus vecinos eran en su mayoría de la clase contraria.

III. ***Tomek Links (Enlaces Tomek)[4]:*** *Es la distancia mínima entre parejas de clases opuestas. Se da cuando teniendo dos instancias Xi y Xj, perteneciendo cada una respectivamente a la clase minoritaria y a la clase mayoritaria, donde existe una distancia entre ambas d(Xi, Xj) y donde no existe una instancia Xk que cumpla con la regla: d(Xi, Xk) < d(Xi, Xj) o d(Xj, Xk) < d(Xi, Xj). En otras palabras: "el enlace tomek ocurre cuando dos instancias de clases contrarias tienen una conexión directa, sin otros elementos entre medias".*

Las instancias que cumplan con estas características y forman un enlace tomek pueden ser clasificadas como ruido o como instancias en el borde de decisión. Una vez eliminadas, el conjunto de datos resultante estará compuesto por instancias cuyos vecinos con distancia mínima pertenecen a su misma clase.

Imagen-2b.



Por último en el filtrado con TomekLinks, no hubo eliminaciones por parte del algoritmo de limpieza, por lo que las clases se mantuvieron niveladas, ya que ninguna de las instancias creadas, formó un enlace TomeK.

2.c Algoritmos de aprendizaje supervisado

2.c-1 Bosques Aleatorios. [7]

La elección de su uso, se basa en la robustez que presentan al evitar los sobreajustes, en comparación con los árboles de decisión los cuales son propensos a tal error de medición, si no se controla la profundidad con que se los entrena. En este sentido es que los bosques aleatorios promedian un conjunto de árboles de decisión, consiguiendo la mejor configuración. Siendo una de las desventajas notables el recurso computacional, si se decide usar una gran cantidad de árboles, lo que después de cierto valor ya no tiene sentido, porque deja de obtenerse una mejora. El recurso computacional, también es influenciado por la medida de selección de atributos que se elija, es decir, si elegimos una medida de desigualdad como ganancia de información, en contraposición a un índice GINI.

Asimismo, es dable destacar que este algoritmo puede manejar la dimensionalidad y las grandes cantidades de datos. Sin embargo este clasificador, también sufre del desequilibrio de clases, siendo que los árboles se vuelven expertos en clasificar la clase mayoritaria, no sucediendo lo mismo con la clase minoritaria de la cual no hay suficientes ejemplos, en consecuencia en la mayoría de los casos por votación siempre va a ganar la clase mayoritaria. Una mejora a este clasificador la propone Wang et al [6], donde se presenta una mejora de muestreo por reemplazo, donde se extraen de forma independiente y aleatoria un subconjunto de ejemplos de la clase mayoritaria, siendo este subconjunto sustancialmente igual al número de ejemplos de la clase minoritaria. Seguidamente se crea un nuevo conjunto con las clases equilibradas y se vuelve a entrenar un Random Forest, logrando de esta forma un mejor rendimiento en el modelo.

Atento a lo mencionado en el párrafo anterior, en el presente trabajo solo se implemento el modelo de Random Forest standard.

2.c-2 Clasificador Bayesiano Ingenuo. [7]

Este tipo de clasificador, tiene como característica básica la independencia entre las variables de un conjunto de datos, esto trae aparejado un inconveniente cuando, en la etapa de testeo aparece una variable no observada en el conjunto de entrenamiento, que en la etapa de predicción del modelo resulta en una probabilidad de cero a esa instancia;

En esta fórmula nos encontramos con una productoria, que para evitar resultados nulos en las instancias con probabilidad cero, se la reduce a un número muy bajo, logrando realizar la clasificación de cada ejemplo, sin pérdida de información.

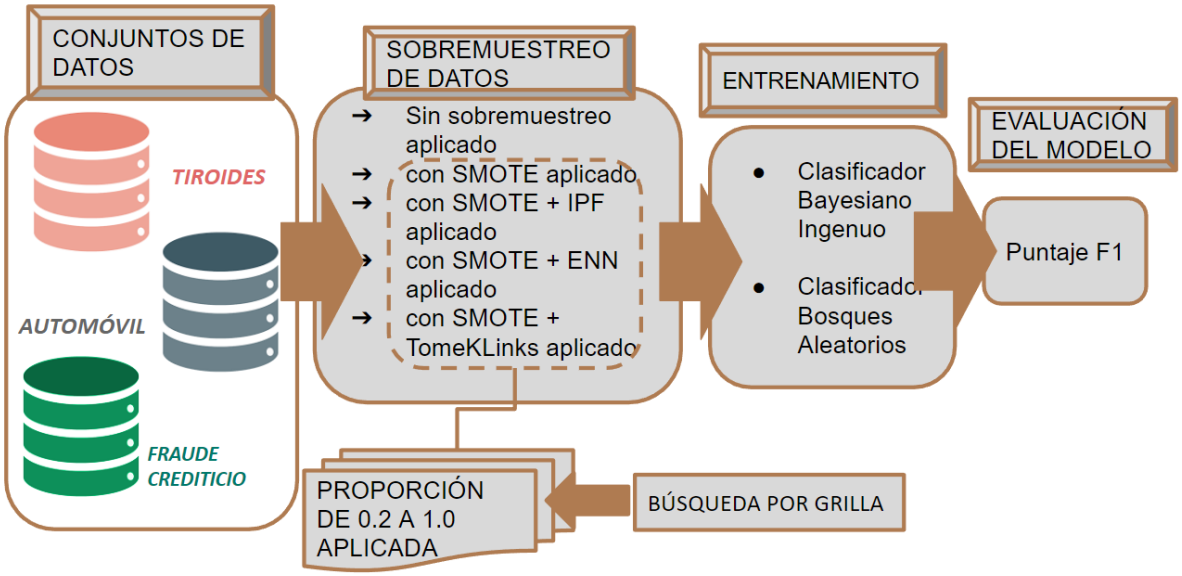
En contraposición a la técnica de aprendizaje automático de RandomForest, Bayes consume muy poco recursos, lo que lo hace más rápido al momento de la ejecución.

Dentro de este marco de los conjuntos desbalanceados, el algoritmo se va a haber afectado en el cálculo de la probabilidad a priori de cada clase, donde va a inclinarse hacia la clase mayoritaria, aumentando la tasa de error en las predicciones.

2.d Experimentos realizados

Los experimentos realizados sobre los distintos datasets, se muestran diferenciados en dos partes, una con la combinación de Dataset, Tipo de Muestreo y Algoritmo de Machine Learning elegido (Gráfico 2.d-1) y la proporción aplicada a cada combinación.

Gráfico 2.d-1



Para la calibración de los hiper parámetros se utilizó un esquema de búsqueda por grillas con validación cruzada, donde la idea principal es la de utilizar las distintas proporciones de creación de instancias artificiales de SMOTE, de la clase minoritaria en comparación a la clase mayoritaria, es decir, se probaron generación de instancias de porcentajes que van del 0.2 a 1.0, donde 1.0

equilibrará el conjunto de datos. De cada proporción, se evalúan dos modelos “Árboles Aleatorios” y “Clasificador bayesiano ingenuo”.

2.d.1 Enlace a Repositorio

Los experimentos realizados fueron subidos al repositorio correspondiente de Github. Allí se pueden observar los datasets mencionados, y las etapas de limpieza, preprocesamiento y entrenamiento de los modelos. [Github - smote_tpfinal]

2.e Herramientas

Para el diseño de los experimentos, se utilizó como lenguaje de programación python, sobre el editor de código de Visual Studio Code, en una Jupyter Notebook. En el mismo, se utilizó la clase GridSearchCV disponible en scikit-learn que permite evaluar y seleccionar de forma sistemática los parámetros de un modelo y los parámetros a probar.

Para los experimentos con SMOTE y las variantes, se utilizó la librería de *Smote-variant*

2.f Métrica de evaluación

El uso de la métrica de evaluación “*puntaje F1*” [wiki], nos permite tener un promedio ponderado entre precisión y exhaustividad (recall). teniendo en cuenta tanto los falsos positivos como los falsos negativos.

Asimismo F1 es más robusta que otras métricas como sensibilidad o recall, ya que en situaciones donde las clases están desbalanceadas corremos el riesgo de evaluar mal el modelo. Por definición esta métrica cuenta la cantidad de instancias relevantes seleccionadas, lo que nos daría una puntuación alta, debido a la gran cantidad de instancias de la clase mayoritaria, sin embargo la medición busca analizar que tan bien identificó el modelo las instancias de la clase minoritaria.

Respecto de la métrica de Precisión no se la tuvo en cuenta debido a que en presencia de clases desbalanceadas el modelo de aprendizaje mostrará un desempeño errado, ya que clasificará correctamente la mayoría de los ejemplos mayoritarios, no así con los ejemplos minoritarios, sin embargo en el resultado final el porcentaje del modelo de desempeño va a ser alto, ante lo cual se podría interpretar como que el modelo funciona bien, pero no es así.

3 Resultados

Los resultados obtenidos para los experimentos realizados con el clasificador bayesiano ingenuo destaca lo siguiente: con bajos requerimientos computacionales como es este clasificador aplicado al dataset de Evaluación del automóvil, se logra una gran mejora del 10,53% con una puntuación F1 de 0.8687, con la implementación de la técnica de sobremuestreo de SMOTE + ENN y una proporción de 1.0. En el caso de Enfermedad de la Tiroides, con una puntuación F1 de 0.1728, fue la mejor puntuación inclusive sin aplicar sobremuestreo; se observó que hubo disminución en la métrica y en el mejor de los casos con una puntuación F1 de 0.1652 reducción de -4.6% con una proporción de 0.4 mediante el uso de la técnica de SMOTE + ENN (Tabla 3.1).
Y para el caso del dataset de Fraude Crediticio, el rendimiento es totalmente deficiente, el mejor caso sucede sin aplicar SMOTE (F1= 0.0579) y el mejor caso es SMOTE+ENN (F1=0.0450) la disminución es del 28.66%.
En la Tabla 3.1 y 3.2, se pueden observar los resultados de aplicar la búsqueda por grillas con validación cruzada con las configuraciones de proporción y clasificador específico (RandomForest, y GaussianNB).

Tabla 3.1: Resultados de rendimiento con GaussianNB - con y sin SMOTE y sus variantes

DATASET´S	Medidas de Performance con F1 SIN aplicar OVERSAMPLERS Clasificador: Bayesiano Ingenuo	Medidas de Performance con F1 aplicando OVERSAMPLERS Clasificador: GaussianNB			
		SMOTE	SMOTE + IPF	SMOTE + ENN	SMOTE + TOMEKLINKS
´Enfermedad de la Tiroides´[*]	**0.1728**	0.1649 (0.7)	0.1645 (1.0)	0.1652 (0.4)	0.1636 (0.8)
´Evaluación del Automóvil´ [*]	0.7859	0.8682 (0.6)	**0.8656 (0.8)**	0.8687 (1.0)	0.8668 (0.7)
´Fraude Crediticio´[*]	0.0579	0.0432 (0.2)	0.0405 (0.2)	**0.0450 (0.2)**	0.0447 (0.2)

Tabla 3.2: Resultados de rendimiento con RandomForest - *con y sin SMOTE y sus variantes*

DATASET´S	Medidas de Performance con F1 SIN aplicar OVERSAMPLERS Clasificador: RandomForest	Medidas de Performance con F1 aplicando OVERSAMPLERS Clasificador: RandomForest			
		SMOTE	SMOTE + IPF	SMOTE + ENN	SMOTE + TOMEKLINKS
´Enfermedad	**0.9861**	0.9796	0.9813	0.9846	0.9813

de la Tiroides´[*]		(0.2)	(0.2)	(1.0)	(0.2)
´Evaluación del Automóvil´[*]	0.9798	0.9797 (0.8)	**0.9867 (1.0)**	0.9788 (0.4)	0.9812 (0.7)
´Fraude Crediticio´[*]	0.1612	0.2060 (0.4)	0.2157 (0.3)	**0.2166 (0.7)**	0.2081 (0.4)

Luego pasando al clasificador de Bosques aleatorios, se pueden observar mejoras y rendimientos muy distintos al anterior. En el caso del dataset de Enfermedad de la Tiroides el mejor caso sucede con la técnica SMOTE + ENN y una proporción de 1.0, pero no alcanza al mejor caso sin aplicar SMOTE (F1=0.9861) la disminución es del -0.15 %; En el caso del conjunto de Evaluación del automóvil se puede observar un leve incremento en la eficiencia de un +0.7% y una proporción de 1.0. Y finalmente en el último conjunto de Fraude crediticio es el que mejores resultados muestra con una mejora del +34.36%, con una proporción de 0.7 usando solo técnica de SMOTE+ENN, siendo que las demás técnicas donde se aplicó la limpieza si bien son menores, la mejora sigue siendo notable.

Comentarios Finales

Del análisis en el rendimiento de ambos clasificadores se puede concluir que, el clasificador bayesiano tuvo un mal rendimiento para las predicciones, sin embargo las mejoras de base con la aplicación de SMOTE + los filtros de limpieza IPF-ENN-TomeKLinks hace que el modelo aumente su rendimiento solo en conjuntos donde las clases no estén demasiado desbalanceadas. En contraposición al clasificador de Bosques Aleatorios, donde aumentó el rendimiento con la generación de nuevas muestras artificiales, en este se destacó la técnica de limpieza de Vecinos más Cercano Editado (ENN), dentro del dataset de Fraude Crediticio y la técnica de Filtro de Partición Iterativa (IPF) en el conjunto de Evaluación del automóvil. Si bien en el conjunto de Enfermedad de la Tiroides disminuye el rendimiento, eso no quita que el rendimiento siguió siendo bueno.

Finalmente se puede concluir que el uso de técnicas de sobremuestreo en conjuntos desbalanceados mejora el rendimiento y aumenta en combinación a técnicas de limpieza posterior a fin de corregir la creación de ejemplos artificiales ruidosos o solapados.

Anexo

2a-1) 'Enfermedad de la Tiroides':

1) AGE
2) SEX
3) ON THYROXINE
4) QUERY ON THYROXINE
5) ON ANTITHYROID MEDICATION
6) SICK
7) PREGNANT
8) THYROID SURGERY
9) I131 TREATMENT
10) QUER1Y HYPOTHYROID
11) QUERY HYPERTHYROID
12) LITHIUM
13) GOITRE
14) TUMOR
14) HYPOPITUITARY
15) PSYCH
16) TSH MEASURED
17) TSH
18) T3 MEASURED
19) T3
20) TT4 MEASURED
21) TT4
22) T4U MEASURED
23) T4U
24) FTI MEASURED
25) FTI
26) TBG MEASURED
27) BINARYCLASS

2a-2) 'Evaluación del Automóvil':

1) BUYING
2) MAINT
3) DOORS
4) PERSONS
5) LUG_BOOT
6) SAFETY
7) CLASS

2a-3) 'Fraude Crediticio':

1) CAR
2) REALITY
3) NO_OF_CHILD
4) INCOME
5) WORK_PHONE
6) PHONE
7) E_MAIL
8) FAMILY
9) BEGIN_MONTH
10) AGE
11) YEARS_EMPLOYED
12) INC_TYPE_PENSIONER
13) INC_TYPE_STATE
14) INC_TYPE_STUDENT
15) INC_TYPE_WORKING
16) EDU_TYPE_HIGHER
17) EDU_TYPE_INCOMPLETE
18) EDU_TYPE_LOWER
19) EDU_TYPE_SECONDARY
20) FAM_TYPE_MARRIED
21) FAM_TYPE_SEPARATED
22) FAM_TYPE_SINGLE
23) FAM_TYPE_WIDOW
24) HOUSE_TYPE_HOUSE
25) HOUSE_TYPE_MUNICIPAL
26) HOUSE_TYPE_OFFICE
27) HOUSE_TYPE_RENTED
28) HOUSE_TYPE_WITH
29) TARGET

Referencias

1. *Elreedy, D., & Atiya, A. F. (2019). A Comprehensive Analysis of Synthetic Minority Oversampling Technique (SMOTE) for handling class imbalance. Information Sciences, 505, 32–64. https://doi.org/10.1016/j.ins.2019.07.070*

2. *Kovács, G. (2019). Smote-variants: A python implementation of 85 minority oversampling techniques. Neurocomputing, 366, 352–354. https://doi.org/10.1016/j.neucom.2019.06.100*

3. *Qian, Y., Liang, Y., Li, M., Feng, G., & Shi, X. (2014). A resampling ensemble algorithm for classification of imbalance problems. Neurocomputing, 143, 57–67. https://doi.org/10.1016/j.neucom.2014.06.021*

4. *Yan, Y. T., Wu, Z. B., Du, X. Q., Chen, J., Zhao, S., & Zhang, Y. P. (2019). A three-way decision ensemble method for imbalanced data oversampling. International Journal of Approximate Reasoning: Official Publication of the North American Fuzzy Information Processing Society, 107, 1–16. https://doi.org/10.1016/j.ijar.2018.12.011*

5. *Fernández, A., Garcia, S., Herrera, F., & Chawla, N. V. (2018). SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. Journal of artificial intelligence research, 61, 863-905.*

6. *Wang, L., Han, M., Li, X., Zhang, N., & Cheng, H. (2021). Review of classification methods on unbalanced data sets. IEEE Access, 9, 64606-64628.*

7. *Han, J., Kamber, M., & Pei, J. (2011). Data mining: concepts and techniques: concepts and techniques. Elsevier.*

8. *Luengo, J., Stefanowski, J. & Herrera, F. (2015, enero). SMOTE–IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. Information Sciences, 291, 184-203. https://doi.org/10.1016/j.ins.2014.08.051*