Prediction of Illicit Bidding in the Spanish Railway Sector

by

Félix Sánchez Miqueleiz

A thesis submitted in conformity with the requirements
for the MSc in Economics, Finance and Computer Science

University of Huelva & International University of Andalusia

September 2023

Predicción de licitaciones ilícitas en el Sector Ferroviario español

Félix Sánchez Miqueleiz

Máster en Economía, Finanzas y Computación

Supervisado por:

Diego Marín Santos y **Manuel Emilio Gegúndez Arias**

Universidad de Huelva y Universidad Internacional de Andalucía

2023

Abstract

The competition in the Spanish railway industry is a clear and recent example of a cartel in many of its subsectors. In this work, the competitive analysis has enable to set up a supervised machine learning model to predict illicit bids. All publicly available information on illicit bids in the railway industry has been gathered. After applying various machine learning techniques, it has been concluded that Random Forest offers the most efficient results with a 97% accuracy rate.

The analysis concludes that there is not enough data to confirm that a bid classified as illicit is truly illicit, primarily due to the limited separability between classes. The publication of more data to enrich analyses focused on industry competition will enable to make prediction models more efficient and verify if it is a suitable mechanism for detecting illicit activities before companies apply for the Leniency Programme.

JEL classification: H4, L13, L14, L44.

Keywords: Machine Learning, Supervised Learning, Binary Classification, AUC, Imbalanced Classes, Cartel, Bid rigging.

Resumen

La competencia del sector ferroviario español es un ejemplo de cártel claro y reciente en muchos de sus subsectores. En este trabajo, el análisis de la competencia ha permitido configurar un modelo de aprendizaje automático supervisado para predecir licitaciones ilícitas. Se ha recopilado toda la información pública disponible de las licitaciones ilícitas del sector ferroviario. Tras la aplicación de diferentes técnicas de aprendizaje automático, se ha concluido que el *Random Forest* es la que ofrece los resultados más eficientes con un 97% de tasa de aciertos.

El análisis concluye que no hay suficientes datos para confirmar que una licitación clasificada como ilícita sea realmente ilícita debido principalmente a la poca separabilidad entre clases. La publicación de más datos para enriquecer análisis enfocados en la competencia de un sector permitirá hacer más eficientes los modelos de predicción y comprobar si es un mecanismo adecuado para la detección de actividades ilícitas antes de que las empresas se acojan al Programa de Clemencia.

Índice de contenido

Índice de figuras

Índice de tablas

1 Propuesta TFM

1.1 Motivación

Mejorar la eficiencia del gasto público es uno de los elementos cruciales para el crecimiento económico de un país. El sector ferroviario español es público y 6 expedientes resueltos de la CNMC han evidenciado la existencia de cárteles en licitaciones ofertadas por ADIF[1] y RENFE[2].

La teoría económica demuestra que los cárteles suponen una pérdida significativa del bienestar social generado por la competencia causado por el excedente del consumidor y la pérdida irrecuperable de eficiencia. Esta pérdida se expande a todos los contribuyentes en el caso de que el sector donde ocurra este fenómeno sea público. Por este motivo, es crucial la regulación y el arbitraje en la competencia de los sectores donde se utilizan recursos públicos.

García-Verdugo, Merino y Miren Martín afirman: "Los cárteles están entre las prácticas anticompetitivas más perjudiciales para la sociedad. Los cárteles aumentan los precios, disminuyen la calidad de los productos y servicios ofrecidos por las empresas, y con frecuencia están relacionados con una reducción en las opciones disponibles para los consumidores." (2019)

De acuerdo con Cerdá Martínez-Pujalte (2018), el 20% del PIB español es contratación pública. Las prácticas anticompetitivas suponen un incremento del 20% en el precio los bienes y servicios. Por ejemplo, en el año 2015, las prácticas anticompetitivas han supuesto un sobrecoste de 47.500 millones de €.

El desmantelamiento de los cárteles se produce principalmente a través del Programa de Clemencia de la UE. Este Programa permite que aquellas empresas que formen parte de un cártel consigan una exención parcial o incluso total en la sanción correspondiente por las actividades económicas ilícitas realizadas. Si las autoridades consiguieran anticiparse y detectar un cártel antes de que las empresas acudan al Programa de Clemencia, se conseguiría reducir aún más los

[1] Administrador de Infraestructuras Ferroviarias

[2] Red Nacional de Ferrocarriles Españoles

incentivos que tienen las empresas a formar un cártel y, en definitiva, evitar que se produzca este fallo de mercado.

1.2 Objetivos

El objetivo general es diseñar un modelo de aprendizaje automático supervisado para predecir actividades económicas anticompetitivas del Sector Ferroviario.

Los objetivos específicos son:

- Construir la base de datos que recopile toda la información disponible sobre las licitaciones en las que se ha demostrado la existencia de prácticas competitivas. Además, los datos han de ser compatibles con los datos de aquellas licitaciones en las que no se ha demostrado tal fenómeno.

- Análisis de variables y su influencia para predecir conductas anticompetitivas.

- Diseño, aplicación y evaluación de modelos de aprendizaje automático: Naïve Bayes Gaussiano, Máquinas de Vector Soporte, *Random Forest*, *Gradient Tree Boosting*.

- Aplicación de técnicas de desbalanceo de datos: Muestreo Aleatorio Simple con Reemplazamiento y *Cluster Centroids.*

- Proponer predictores que puedan enriquecer el análisis para futuras aplicaciones en la política *antitrust*.

1.3 Estructura del documento

El documento se divide en tres partes:

1. Introducción y desarrollo del contexto económico tanto de la colusión explícita en general, como en la competencia del sector ferroviario. Explicación del marco teórico de las técnicas de aprendizaje automático utilizadas.

2. Explicación de las clases y predictores del problema y de la base de datos utilizada. Exposición de las estadísticas descriptivas de la base de datos.

3. Fundamentos de los algoritmos de Aprendizaje Automático utilizados y análisis de las métricas obtenidas en los distintos escenarios en los que se aplican.

2 Introducción. Estado del arte

2.1 Marco teórico económico

Estimar la probabilidad de detección de un cártel es fundamental para determinar de forma óptima aquella sanción que desincentive el reparto de mercado entre dos o más empresas. Allain et al. (2013) consideran que la condición suficiente para disuadir estos fenómenos es la siguiente:

$$F \geq \frac{\Delta\pi}{\lambda} \equiv DDF$$

Siendo F la sanción a imponer, $\Delta\pi$ el beneficio anual ilícito por pertenecer al cártel, λ la probabilidad anual de detectar el cártel y DDF[3] es la Sanción Dinámica Disuasoria. Por lo tanto, si la probabilidad de desmantelamiento de cártel es igual a ($\lambda = 0{,}5$) entonces la multa a imponer (F) ha de ser, como mínimo, el doble que el beneficio anual ilícito por pertenecer al cártel. Cuanto menor es λ, F es mayor suponiendo que $\Delta\pi$ permanezca constante.

El Programa de Clemencia causa dos efectos en la ecuación. El primer efecto es el aumento de λ, ya que las empresas que forman parte del cártel tienen incentivos extra para desmantelar el cártel al que pertenecen. Dicho incentivo es cuantificable y es igual a la cuantía correspondiente a la exención de la multa que las autoridades les imponga. El segundo efecto es la consecuente deflación de F, ya que en términos netos solo se acaba sancionando a aquellas empresas que formaron parte del cártel y que no acudieron al Programa de Clemencia.

Si las autoridades consiguen predecir con suficiente tiempo la existencia de un cártel, el Programa de Clemencia no sería necesario para desmantelar los cárteles.

[3] Son las siglas en inglés de *Dynamic Deterrence Fine*

2.2 Descripción de la competencia en el Sector Ferroviario

Desde el punto de vista de la oferta y la demanda, el sector ferroviario funciona de la siguiente forma:

1. Desde el punto de vista de la oferta: Son empresas privadas que ofertan sus servicios para el correcto funcionamiento y expansión del sector ferroviario. Bajo condiciones de competencia, aquella empresa que ofrezca la mejor relación calidad – precio en la licitación en la que participe, será la empresa adjudicataria de la licitación.
2. Desde el punto de vista de la demanda: Principalmente ADIF y RENFE publican licitaciones mediante distintas vías. Estas licitaciones son tramitadas por toda la burocracia correspondiente resumida en la Figura 1.

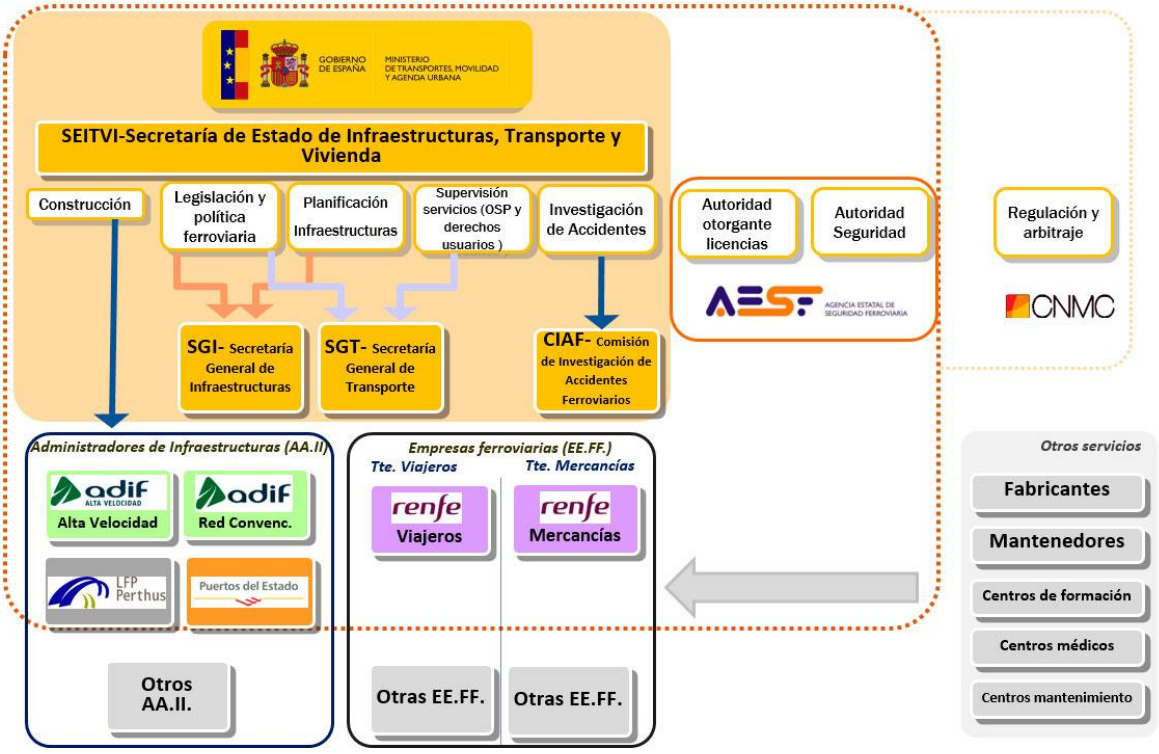

Figura 1: Esquema de los órganos públicos involucrados en contrataciones del sector público en el sector ferroviario.[4]

[4] Fuente y elaboración: CNMC

Hay evidencias que demuestran que el comportamiento que tienen las empresas oferentes que coluden explícitamente en las licitaciones del sector ferroviario son distintas de las empresas que compiten realmente. Dichas evidencias se pueden resumir en los siguientes fenómenos:

- Se pactan las ofertas previamente para simular competencia y repartirse las licitaciones.
- Los licitadores aceptan abstenerse o retirarse de una o varias ofertas a cambio de una comisión pagada por el propio cártel.
- Se establecen turnos para ganar las licitaciones y se reparten el mercado por zonas geográficas o por distintos tipos de mercado. Tienen el suficiente poder de mercado para hacerlo.
- Utilizan la UTE[5] de forma injustificada, aunque sea aceptada legalmente en la licitación en curso.
- El precio adjudicado, que suele ser inferior al precio presupuestado, se aproxima al precio presupuestado o incluso lo supera en las licitaciones que han sido objeto de reparto en términos medios. Este fenómeno es notable también comparando las pujas con el precio presupuestado.

Estas evidencias justifican la realización de este trabajo cuya motivación fundamental es construir una base de datos con predictores adecuados para valorar la viabilidad del desarrollo de un modelo de predicción para la detección de conductas anticompetitivas.

3 Marco teórico.

En esta sección se detalla todo el marco teórico que tiene que ver con las técnicas de aprendizaje automático que se aplica para predecir las clases del problema que se plantea.

[5] Una Unión Temporal de Empresas (UTE) es un sistema utilizado en España por el cual dos o más empresas se unen para realizar una obra o prestar un servicio determinado. Se constituyen como una única empresa temporalmente mientras dure la obra, normalmente de gran porte.

3.1 Técnicas de desbalanceo seleccionadas.

La base de datos que se ha creado para la realización de este trabajo se describe en el Capítulo 4.1. De forma resumida, se han extraído 2.216 instancias correspondientes a licitaciones objeto de reparto y licitaciones no objeto de reparto. Por lo tanto, tal y como se representa en la (Figura 2), las clases del problema presentan un importante desbalanceo.

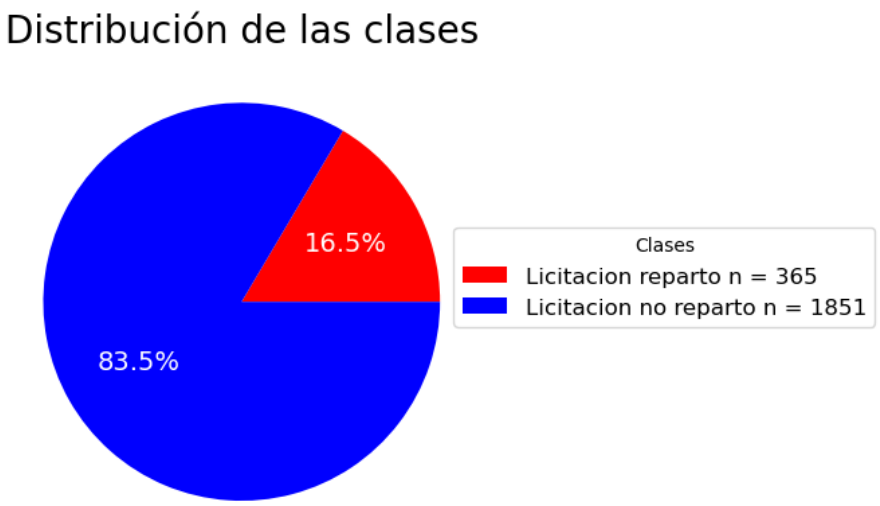

Figura 2: Distribución de las clases del problema de clasificación binaria.

Las técnicas utilizadas para tratar el desbalanceo de clases que se han seleccionado son las siguientes:

- Para el sobre muestreo se utiliza el Muestreo Aleatorio Simple con Reemplazamiento: Se replica muestras obtenidas mediante el muestreo aleatorio simple con reemplazamiento[6] hasta conseguir que el número de instancias de la clase minoritaria iguale el de la clase mayoritaria tal y como se representa en la (Figura 3).

[6] Más información en https://imbalanced-learn.org/stable/over_sampling.html#random-over-sampler

- Para el infra muestreo se utiliza el *Cluster Centroids*[7]: Se selecciona un subconjunto de los centroides calculados mediante un estimador K-*Means* que segmenta el espacio de características. Esta técnica permite captar toda la variabilidad de la clase mayoritaria para que los clasificadores se entrenen correctamente. El número de centroides seleccionados será igual al número de instancias de la clase minoritaria tal y como se representa en la (Figura 4).

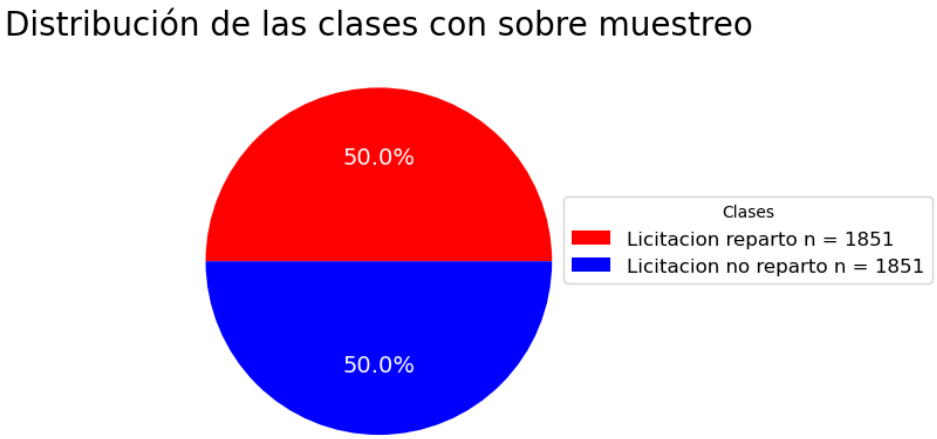

Figura 3: Distribución de las clases del problema de clasificación binaria tras el sobre muestreo.

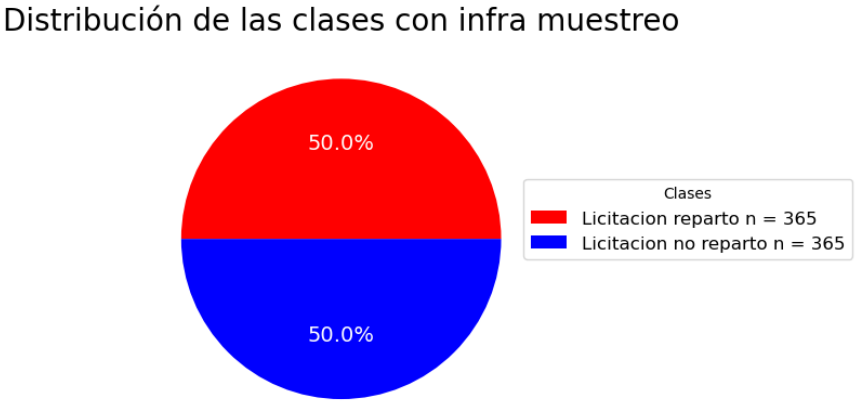

Figura 4: Distribución de las clases del problema de clasificación binaria tras el infra muestreo.

[7] Más información en https://imbalanced-learn.org/stable/under_sampling.html#cluster-centroids

3.2 Técnicas de clasificación seleccionadas.

3.2.1 Clasificador Naïve Bayes Gaussiano.[8]

El clasificador Naïve Bayes (clasificador Bayesiano “ingenuo”), se fundamenta en el teorema de Bayes. Este enfoque recibe su nombre de "ingenuo" debido a su suposición de que las variables utilizadas para la predicción son mutuamente independientes. Además, se supone que todos los predictores siguen una distribución normal o gaussiana.

El teorema de Bayes ofrece un método para calcular la probabilidad a posteriori P(c|x) mediante el uso de P(c), P(x), y P(x|c). El clasificador Bayesiano ingenuo asume que el impacto del valor de un predictor x en una clase determinada c es independiente de los valores de otros predictores, lo que se conoce como independencia condicional de clase. La fórmula de este algoritmo se expresa de la siguiente manera:

Este método se utiliza para calcular la probabilidad de asignar un objeto a una clase específica sin necesidad de estimar ningún parámetro. En ocasiones, este clasificador puede producir resultados superiores a métodos más avanzados, especialmente en dos situaciones: cuando los atributos son completamente independientes o cuando son completamente dependientes entre sí. Esto suele ocurrir cuando existe una marcada diferencia entre las medias de las variables. Sin embargo, en situaciones intermedias entre estos dos escenarios, los resultados generados por este algoritmo pueden no ser tan efectivos.

En *Scikit-learn* se pueden configurar los siguientes hiperparámetros para este clasificador:

- *priors*: Este hiperparámetro permite proporcionar una lista de probabilidades *a priori* para las clases en el conjunto de datos. Las probabilidades a priori son las estimaciones iniciales de la probabilidad de que un ejemplo pertenezca a una clase antes de observar cualquier evidencia (características). Si no se proporciona ninguna, el modelo calculará las probabilidades a priori a partir de los datos de entrenamiento de manera automática.

[8] Mateo Vázquez J. D. (2017). Competición de Kaggle.com: Santander Costumer Satisfaction pp 13-14

- *var_smoothing*: Es un hiperparámetro que controla la suavización de las varianzas de las características (atributos) en el modelo Naïve Bayes. Este proceso se llama suavización de Laplace o suavización de aditivo. Se utiliza para evitar que la probabilidad de una característica sea cero para una clase si esa característica no aparece en absoluto en las muestras de entrenamiento de esa clase. Esto es importante para prevenir problemas de probabilidad nula que podrían hacer que el modelo sea demasiado confiado o inestable.

3.2.2 Clasificador Máquinas de Vector Soporte (SVM).[9]

Las Máquinas de Vector Soporte son algoritmos de aprendizaje supervisado utilizados en tareas de clasificación y regresión. Estos sistemas operan en un espacio hiperdimensional y se entrenan mediante un algoritmo basado en la teoría de optimización que incorpora un sesgo de aprendizaje derivado de distintos principios estadísticos.

El SVM emplea un conjunto de datos de entrenamiento para construir un modelo que busca separar las clases de datos en dos espacios de la manera más amplia posible, mediante uno o varios hiperplanos de separación. La solución del hiperplano se basa en la combinación de puntos de entrada llamados vectores de soporte. El objetivo del algoritmo es encontrar el hiperplano óptimo que logre la mejor separación de datos, lo que se traduce en un problema de optimización donde se busca maximizar el margen entre las clases. Una vez se dispone de nuevas muestras, estas pueden clasificarse según su posición relativa a los hiperplanos de separación.

El SVM se desempeña bien cuando existe una clara separación entre las categorías y es eficiente en espacios de alta dimensionalidad. Sin embargo, puede ser ineficiente en bases de datos grandes debido a los tiempos de entrenamiento relativamente largos que requiere.

En *Scikit-learn* se pueden configurar los siguientes hiperparámetros para este clasificador:

[9] Mateo Vázquez J. D. (2017). Competición de Kaggle.com: Santander Costumer Satisfaction pp 14

- C: El hiperparámetro C controla el parámetro de regularización en una SVM. Un valor más alto de C implica una regularización más débil, lo que significa que el modelo intentará clasificar todos los ejemplos de entrenamiento correctamente, incluso si eso significa tener un margen de decisión más pequeño. Un valor más bajo de C implica una regularización más fuerte, lo que dará como resultado un margen más grande pero posiblemente algunos ejemplos de entrenamiento clasificados incorrectamente. Ajustar C permite equilibrar la precisión en el conjunto de entrenamiento con la capacidad de generalización del modelo.
- *break*_ties: El hiperparámetro break_ties se utiliza cuando hay empate en la decisión del hiperplano en SVM. Cuando se establece en True, el clasificador romperá los empates de manera predecible, lo que puede ser útil en algunas aplicaciones donde se requiere una decisión determinista en caso de empate. Cuando está en False, el comportamiento en caso de empate es menos predecible.
- *cache_size*: Controla el tamaño de la memoria caché utilizada por el clasificador SVM para almacenar información intermedia durante el entrenamiento. Un valor más grande de *cache_size* puede acelerar el entrenamiento si se tiene suficiente memoria RAM disponible.
- *class_weight*: Permite asignar diferentes pesos a las clases en función de su importancia relativa. Si se establece en *None*, todas las clases tienen el mismo peso. Si se elige un valor diferente, se pueden dar pesos mayores a clases menos frecuentes para equilibrar el impacto de las clases en el proceso de entrenamiento.
- *coef0*: Es un hiperparámetro utilizado en algunos kernels, como el kernel polinómico y el kernel sigmoide. Controla la influencia de los términos independientes en la función de decisión.
- *decisión_function_shape*: Determina la estrategia para generar las puntuaciones de decisión en un problema de clasificación multiclase. *'ovr'* significa "uno contra el resto", lo que genera una puntuación de decisión por clase en un enfoque de clasificación binaria. Otras opciones incluyen 'ovo' para "uno contra uno" y 'raw' para puntuaciones de decisión sin procesar.
- *degree*: Es el grado del kernel polinómico. Este hiperparámetro se utiliza solo cuando el kernel es polinómico.

- *kernel*: Especifica el tipo de *kernel* utilizado en la Máquina de Vector Soporte. 'rbf' se refiere al kernel de base radial, pero también se puede elegir otros como 'linear', 'poly' (para kernel polinómico) o 'sigmoid'.
- *max_iter*: Establece el número máximo de iteraciones para el solucionador de optimización interno. Si se establece en -1, se utilizará un valor predeterminado interno.
- *probability*: Si *probability* se establece en True, el clasificador SVM calculará las probabilidades de clase. Esto es útil si deseas estimaciones de probabilidad en lugar de solo las etiquetas de clase.
- *random_state*: Controla la semilla utilizada por el generador de números aleatorios para garantizar la reproducibilidad de los resultados.
- *shrinking*: El hiperparámetro *shrinking* controla si se debe utilizar la técnica de encogimiento para acelerar el entrenamiento. Cuando está en *True,* el modelo intentará eliminar ejemplos de entrenamiento que no son vectores de soporte para acelerar el proceso de entrenamiento.
- *tol*: Es la tolerancia para el criterio de convergencia del entrenamiento. El entrenamiento se detendrá cuando la diferencia entre los valores de pérdida en dos iteraciones consecutivas sea menor o igual a *tol*.
- *verbose*: Controla si se muestran mensajes informativos durante el entrenamiento. Cuando se establece en *True*, el modelo imprimirá información sobre el progreso del entrenamiento.

3.2.3 Random Forest[10]

El algoritmo *Random Forest* emplea la técnica de agregación llamada Bootstrap o *Bagging* para entrenar árboles. En el proceso de *Bagging*, se toma un conjunto de entrenamiento compuesto por datos y sus correspondientes respuestas. Luego, se realiza el muestreo con reemplazamiento de este conjunto de datos de entrenamiento repetidamente un número B de veces. A continuación, se entrena un árbol de clasificación o regresión utilizando estas muestras. En el caso de un problema de clasificación, se elige la clase predicha que sea más frecuente entre los resultados. Este

[10] Mateo Vázquez J. D. (2017). Competición de Kaggle.com: Santander Costumer Satisfaction pp 15

procedimiento mejora el rendimiento del modelo al reducir la variabilidad sin aumentar el sesgo. La cantidad de árboles entrenados es un parámetro ajustable.

El *Random Forest* es similar al *Bagging*, pero difiere en que selecciona un subconjunto aleatorio de las características del conjunto de datos durante el proceso de aprendizaje. Esto significa que, si ciertas características tienen una alta capacidad predictiva de la variable dependiente, entonces serán seleccionadas en varios de los conjuntos de árboles entrenados. En general, para problemas de clasificación, se suelen usar aproximadamente la raíz cuadrada del número total de características en cada división, mientras que, para problemas de regresión, se suelen utilizar alrededor de un tercio de las características.

El *Random Forest* es un algoritmo eficaz para hacer predicciones, ya que generaliza bien y tiende a evitar el sobreajuste a los datos de entrenamiento. Además, es adecuado para grandes volúmenes de datos, problemas desequilibrados y conjuntos de datos con muchas características. También se utiliza comúnmente para evaluar la importancia de las características en un conjunto de datos.

En *Scikit-learn* se pueden configurar los siguientes hiperparámetros para este clasificador:

- *Bootstrap*: El hiperparámetro *bootstrap* controla si se debe utilizar el muestreo con reemplazamiento al construir cada uno de los árboles en el bosque. Cuando está configurado en True, cada árbol se construye utilizando una muestra aleatoria con reemplazo de los datos de entrenamiento. Esto introduce variabilidad en el proceso de construcción del árbol, lo que puede mejorar la generalización del modelo.
- *ccp_alpha*: Es un hiperparámetro que controla la complejidad del árbol de decisión en cada estimador del bosque. Un valor más alto de ccp_alpha lleva a la poda más agresiva de las ramas del árbol, lo que resulta en árboles más pequeños y simplificados.
- *class_weight*: Permite equilibrar las clases en el conjunto de datos proporcionando pesos a las clases. 'gini' significa que los pesos se calculan de manera automática para que sean inversamente proporcionales a la frecuencia de las clases en el conjunto de entrenamiento. También se puede establecerlo en *None* o proporcionar pesos personalizados.

- *criterion*: Especifica la función utilizada para medir la calidad de una división en cada nodo del árbol. Si se establece en *None*, se utiliza 'gini' para problemas de clasificación.
- *max_depth*: controla la profundidad máxima de cada árbol en el bosque. 'auto' significa que los árboles se expandirán hasta que todas las hojas sean puras o contengan menos ejemplos que *min_samples_split*.
- *max_features*: Especifica el número máximo de características a considerar al buscar la mejor división en cada nodo del árbol. 'auto' significa que se consideran todas las características.
- *max_leaf_nodes*: Controla el número máximo de nodos hoja en cada árbol. Si se establece en *None*, no hay restricción en el número de nodos hoja.
- *min_impurity_decrease*: Establece un umbral para dividir un nodo en función de la reducción mínima de impureza requerida. Si la reducción de impureza en un nodo es menor que este valor, el nodo no se dividirá.
- *min_samples_split*: Establece el número mínimo de ejemplos requeridos para dividir un nodo interno en el árbol.
- *min_weight_fraction_leaf* Controla la fracción mínima del total de ejemplos que deben estar en una hoja. Es similar a *min_samples_leaf*, pero se expresa como una fracción en lugar de un número absoluto.
- *n_estimators*: Especifica el número de árboles en el bosque. Cuantos más árboles se tenga, mayor será la capacidad del modelo para generalizar.
- *n_jobs*: controla el número de núcleos de CPU a utilizar para entrenar los árboles en paralelo. Si se establece en *None*, se utilizarán todos los núcleos disponibles.
- *oob_score:* Si se establece en True, se calcula el error de clasificación fuera de la bolsa (out-of-bag error), que es una estimación de la precisión del modelo en datos no utilizados en el entrenamiento.
- *random_state:* Controla la semilla utilizada por el generador de números aleatorios para garantizar la reproducibilidad de los resultados.
- *Verbose:* Controla la cantidad de información que se muestra durante el entrenamiento. Un valor mayor que 0 mostrará información detallada sobre el proceso de entrenamiento.

- *warm_start*: Permite continuar el entrenamiento desde donde se dejó en una llamada anterior al método *fit* de *Scikit-learn*. Esto puede ser útil en ciertos escenarios de entrenamiento incremental.

3.2.4 *Gradient Tree Boosting* (GTB)[11]

Cuando se trabaja con un conjunto de datos para predecir una variable objetivo, es necesario definir una función objetivo que evalúe el desempeño de un modelo con parámetros específicos. La función objetivo típicamente consta de dos componentes: la función de pérdida y la regularización.

Donde el primer sumando de la función objetivo es la función de pérdida utilizada durante el entrenamiento, y el segundo sumando representa el término de regularización. La función de pérdida mide la capacidad del modelo para predecir los datos de entrenamiento, siendo un ejemplo de ello el error cuadrático medio. El término de regularización controla la complejidad del modelo y ayuda a prevenir el sobreajuste, siguiendo el principio general de buscar un modelo sencillo y predictivo.

En el caso de *Gradient Tree Boosting*, se emplea un conjunto de árboles de clasificación y regresión (CART). A diferencia de los árboles de decisión tradicionales, en los modelos CART, cada hoja contiene una puntuación en lugar de una decisión única. Las puntuaciones de predicción de cada árbol individual se suman para obtener la puntuación final, permitiendo que los árboles se complementen entre sí

En resumen, *Gradient Tree Boosting* es un algoritmo de ensamblaje de árboles que mejora gradualmente sus predicciones combinando árboles secuencialmente, con el objetivo de abordar las deficiencias de los árboles anteriores y obtener un modelo más preciso. Para entrenar el modelo, se adopta una estrategia aditiva en la que se fija lo que se ha aprendido previamente y se agrega un nuevo árbol en cada paso. En cada uno de estos pasos, se agrega el árbol que mejor optimiza el objetivo.

En *Scikit-learn* se pueden configurar los siguientes hiperparámetros para este clasificador:

[11] Mateo Vázquez J. D. (2017). Competición de Kaggle.com: Santander Costumer Satisfaction pp 16 - 17

- *ccp_alpha*: Es el mismo hiperparámetro que se utiliza para el *Random Forest.*

- *criterion*: Especifica la función utilizada para medir la calidad de una división en cada nodo del árbol. 'friedman mse' se refiere a la métrica de error cuadrático medio de Friedman, que es adecuada para algoritmos de aumento de gradiente.
- *init*: Es un estimador que se utiliza para inicializar el modelo. Se puede proporcionar un estimador pre entrenado, de lo contrario, se utilizará el valor predeterminado. *None*.
- learning_rate: Controla la tasa de aprendizaje en el algoritmo en este clasificador. Un valor más bajo hace que el modelo sea más robusto, pero puede requerir más estimadores (árboles) para alcanzar un rendimiento similar.
- *loss:* Especifica la función de pérdida utilizada para medir la discrepancia entre las predicciones y los valores reales. 'deviance' se refiere a la pérdida de deviance, que es adecuada para problemas de clasificación.
- *max_depth*: Es el mismo hiperparámetro que se utiliza para el *Random Forest.*
- *max_features*: Es el mismo hiperparámetro que se utiliza para el *Random Forest.*
- *max_leaf_nodes*: Es el mismo hiperparámetro que se utiliza para el *Random Forest.*
- *min_impurity_decrease*: Es el mismo hiperparámetro que se utiliza para el *Random Forest.*
- *min_samples_leaf*: Es el mismo hiperparámetro que se utiliza para el *Random Forest.*
- *min_samples_split*: Es el mismo hiperparámetro que se utiliza para el *Random Forest.*
- *min_weight_fraction_leaf*: Es el mismo hiperparámetro que se utiliza para el *Random Forest.*
- *n_estimators*: Es el mismo hiperparámetro que se utiliza para el *Random Forest.*
- *n_iter_no_change*: Controla cuántas iteraciones se pueden realizar sin mejorar la puntuación de validación antes de detener el entrenamiento. Si se establece en *None*, no se aplica esta detención temprana.
- *random_state*: Controla la semilla utilizada por el generador de números aleatorios para garantizar la reproducibilidad de los resultados.
- *subsample*: Controla la fracción de datos de entrenamiento a utilizar para ajustar cada árbol base.

- *tol*: Establece un umbral de convergencia para este clasificador. El entrenamiento se detiene cuando la mejora en la función de pérdida es menor que este valor.
- *validation_fraction*: Especifica la fracción de datos que se debe reservar para la validación temprana durante el entrenamiento. Esto es útil para el control de la detención temprana.
- *verbose*: Controla la cantidad de información que se muestra durante el entrenamiento. Un valor mayor que 0 mostrará información detallada sobre el proceso de entrenamiento.
- *warm_start*: Permite continuar el entrenamiento desde donde se dejó en una llamada anterior al método fit. Esto puede ser útil en ciertos escenarios de entrenamiento incremental.

3.3 Metodologías y métricas de evaluación.

El preprocesamiento de los datos ha consistido en codificar las variables categóricas convirtiendo las variables multinomiales en variables binomiales añadiendo tantas columnas como sean necesarias. Después se ha estandarizado y normalizado las variables numéricas.

Para cada clasificador en cada posible escenario del problema, es decir, el problema sin tratar el desbalanceo y el problema tratando el desbalanceo mediante las técnicas seleccionadas tanto para sobre muestreo como para infra muestreo, se hacen los siguientes pasos:

1. Se adopta la estrategia *Leave One Out* debido al número reducido de instancias disponibles. *Leave One Out*[12] es una validación simple cruzada. Cada conjunto de aprendizaje se crea tomando todas las observaciones excepto una, siendo el conjunto de prueba la observación que se deja fuera. Por lo tanto, para n observaciones, hay n conjuntos de entrenamiento diferentes y n conjuntos de prueba diferentes. Este procedimiento de validación cruzada no desperdicia muchos datos, ya que solo se elimina una observación del conjunto de entrenamiento. Es lo mismo que utilizar el método de validación simple cruzada *k-fold* con k = n.

[12] https://scikit-learn.org/stable/modules/cross_validation.html#leave-one-out

2. La curva ROC[13] es una técnica desarrollada para evaluar y analizar el rendimiento de un sistema de toma de decisiones. Esta metodología se basa en una representación gráfica que compara dos medidas clave, la sensibilidad y la especificidad, en el contexto de un sistema de clasificación binario. La sensibilidad se refiere a la capacidad del sistema para identificar correctamente los casos positivos, y se calcula dividiendo el número de casos verdaderamente positivos entre el número de casos clasificados como positivos. En contraste, la especificidad se relaciona con la capacidad del sistema para identificar correctamente los casos negativos, y se calcula dividiendo el número de casos verdaderamente negativos entre el número de casos clasificados como negativos. En la curva ROC, el eje de abscisas representa los falsos positivos (uno menos la especificidad), mientras que el eje de ordenadas representa la sensibilidad. Por lo tanto, la posición ideal en esta curva se encuentra cerca del vértice superior izquierdo, lo que indica un alto nivel tanto de sensibilidad como de especificidad. La curva ROC proporciona una visión completa del rendimiento de un clasificador en su rango operativo completo. Se utiliza para visualizar y seleccionar el mejor clasificador, aquel que maximiza los verdaderos positivos y negativos al tiempo que minimiza los falsos positivos. El criterio para evaluar un clasificador puede variar según el tipo de problema que se esté abordando. El área bajo la curva ROC (AUC[14]) es una métrica comúnmente utilizada que representa la probabilidad de que un clasificador coloque aleatoriamente una instancia positiva por encima de una instancia negativa. El AUC se utiliza para comparar diferentes clasificadores y evaluar cuál tiene un rendimiento superior.

3. Se obtiene la matriz de confusión (Figura 5) y se calculan las métricas correspondientes en el punto óptimo de la curva ROC, que será aquella probabilidad que maximiza las métricas de la especificidad y sensibilidad.

[13] Receiver Operating Characteristic

[14] Area Under the Curve

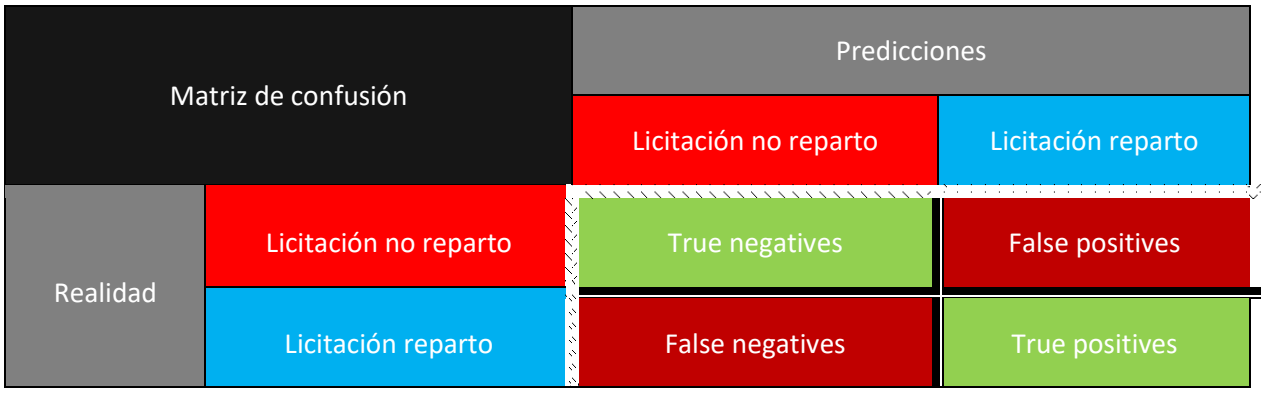

Figura 5: Matriz de confusión del problema de clasificación binaria.

Las métricas[15] que se calculan y almacenan para realizar el paso 3) además del AUC son las siguientes:

- Tasa de aciertos:

$$\frac{True\ positives + True\ Negatives}{\mathrm{N}}$$

- Sensibilidad:

$$1 + \frac{True\ positives}{False\ Negatives}$$

- Especificidad:

$$1 + \frac{True\ negatives}{False\ Positives}$$

[15] *Positives* son las instancias de la clase positiva (en el problema analizado, las licitaciones objeto de reparto) y *Negatives* son las instancias de la clase negativa (en el problema analizado, las licitaciones que no han sido objeto de reparto). *True* indica si la predicción coincide con la clase real y *False* indica si la predicción no coincide con la clase real.

- Precisión:

$$1 + \frac{True\ positives}{False\ Positives}$$

- F1:

$$\frac{2 \times Sensibilidad \times Precisión}{Sensibilidad + Precisión}$$

4 Materiales.

En esta sección se describe cada una de las clases del problema y cada uno de los predictores de la base de datos. Además, se analizan los datos mediante una serie de estadísticas descriptivas. También se visualizan los predictores cuantitativos por cada uno de los predictores cualitativos.

4.1 Descripción de la base de datos.

El Ministerio de Hacienda y Función Pública y la CNMC han sido el origen desde el que se ha recopilado todos los datos. A partir del año 2012 se han empezado a publicar las licitaciones públicas en la página web del Ministerio de Hacienda y Función Pública[16]. Las primeras licitaciones publicadas de ADIF son desde el año 2014.

Sin embargo, las licitaciones que se encuentran en los PDF disponibles en la página web de la CNMC son desde el año 2002 hasta el año 2017 sin contar con las observaciones que tienen valores nulos en alguno de los predictores que se han utilizado para realizar el análisis. Se considera que utilizar técnicas de sustitución de valores nulos sesgaría el análisis debido a dos motivos:

[16] https://www.hacienda.gob.es/es-ES/GobiernoAbierto/Datos%20Abiertos/Paginas/LicitacionesContratante.aspx

1) No se descartan del análisis un porcentaje mayoritario de las observaciones disponibles para la clase licitación reparto. 27 sobre 392 observaciones son valores nulos. Es decir, un 6.9%.

2) No hay suficiente separabilidad de entre las clases. Las técnicas de sustitución de valores nulos que se utilizarían podrían convertir observaciones nulas en observaciones cuyos valores sean más parecidos a los de la clase contraria.

En el Anexo se puede encontrar más información con respecto a cómo se ha construido la base de datos. Sorprende los datos tan limitados y tan poco accesibles que hay sobre las licitaciones públicas ilícitas que hay publicadas. Las columnas de la base de datos son las siguientes:

Hay un total de 2 clases:

1. Licitación objeto de reparto: Sí se ha demostrado la existencia de prácticas anticompetitivas en la licitación.

2. Licitación no objeto de reparto: No se ha demostrado la existencia de prácticas anticompetitivas en la licitación.

Hay un total de 5 predictores:

3. Presupuesto base con impuestos licitación / lote (Predictor continuo): Es una estimación que hace ADIF de entorno a cuánto costaría realizar la obra o servicio licitado.

4. Importe adjudicación con impuestos licitación/lote (Predictor continuo): Es el importe que acaba pagando ADIF a la empresa adjudicataria o ganadora de la licitación.

5. Mercado Producto o MP (Predictor categórico).

5.1 Es igual a 1 si la licitación se corresponde con las descripciones del mercado de producto del siguiente informe: Resolución Expte. Electrificación y electromecánica ferroviaria. S/DC/0598/16 (2019).

5.2 Es igual a 4 si la licitación se corresponde con las descripciones del mercado de producto del siguiente informe: Resolución Expte, S/DC/0519/14 Infraestructuras ferroviarias (2016).

5.3 Es igual a 6 si la licitación se corresponde con las descripciones del mercado de producto del siguiente informe: ADIF. Resolución Expte. S/DC/0614/17 Seguridad y Comunicaciones Ferroviarias (2021).

6. Empresa sancionada (Predictor categórico).

6.1 Si es igual a 0 indica que la empresa adjudicataria no fue sancionada (independientemente de si la licitación ocurrió antes o después de la sanción)

6.2 Si es igual a 1 Indica que la empresa adjudicataria fue sancionada (independientemente de si la licitación ocurrió antes o después de la sanción)

Las empresas que han sido sancionadas son: Alstom, Amurrio Ferrocarril y Equipos, Bombardier, Cafs, Citracc, Cobra, Comsa, Cymi, Duro Felguera Rail, Elecnor, Electren, Eym, Inabensa, Indra, Isolux, Jez Sistemas Ferroviarios, Neopul, Nokia, Semi, Siemens, Siemens Rail, Talleres Alegría, Telice y Thales.

Las empresas que se acogieron al Programa de Clemencia en alguno de los expedientes consiguiendo una exención parcial o total de la sanción son: Alstom, Siemens y Siemens Rail.

7 UTE (Predictor categórico).

7.1 Si es igual a 0 Indica el número de empresas que conforman la parte adjudicataria es igual a 1

7.2 Si es igual a 1 Indica el número de empresas que conforman la parte adjudicataria es mayor que 1.

4.2 Análisis de datos.

4.2.1 Predictores numéricos

Las estadísticas globales para los predictores numéricos importe presupuestado e importe adjudicado contienen los resultados de los estadísticos recuento, media, desviación estándar, mínimo, percentil 25, percentil 50 o mediana, percentil 75 y el máximo. Los resultados de estas estadísticas están representados en la Tabla 1 y en la Tabla 2.

Importe presupuestado	*No licitación reparto*	*Licitación reparto*
Recuento	1851	365
Media	7.554.080,81 €	20.783.089,35 €
Desviación estándar	31.615.363,26 €	63.228.651,56 €
Mínimo	21.780,00 €	23.284,77 €
Percentil 25	158.456,80 €	245.824,00 €
Mediana	498.085,69 €	843.023,56 €
Percentil 75	3.136.504,33 €	9.384.990,09 €
Máximo	640.794.710,72 €	609.019.766,00 €

Tabla 1: Estadísticas globales del predictor Importe presupuestado.

Importe adjudicado	*No licitación reparto*	*Licitación reparto*
Recuento	1851	365
Media	6.672.706,26 €	18.812.981,61 €
Desviación típica	26.756.131,21 €	56.574.260,16 €
Mínimo	20.379,58 €	19.629,06 €
Percentil 25	143.598,65 €	221.088,00 €
Mediana	445.418,21 €	764.618,00 €
Percentil 75	2.903.285,95 €	9.362.302,59 €
Máximo	511.925.990,98 €	504.244.803,00 €

Tabla 2: Estadísticas globales del predictor Importe adjudicado.

Se aprecia que en ambos predictores, la media, la mediana y la desviación típica es mayor para la clase licitación reparto que para la clase licitación no reparto. Sin embargo, debido a la variación que tiene las observaciones, estas tablas no muestran suficiente información para encontrar la separabilidad entre las clases del problema.

Por lo tanto, para analizar si hay separabilidad entre las clases del problema, se representa en un gráfico X-Y los predictores continuos. En primer lugar, sin filtrar (Figura 5 y Figura 6) y en segundo lugar filtrando por cada valor único de cada predictor categórico (de la Figura 7 a la Figura 20). Se añade un diagrama de caja con los mismos filtros junto a cada gráfico X-Y. Se observa que cada diagrama de caja tiene un porcentaje asociado. Estos porcentajes representan la proporción de instancias que son *fliers*[17]. Además, se muestran las matrices de covarianza por clase y por valor único de los predictores categóricos. Estas matrices están representadas desde la Tabla 3 hasta la Tabla 10.

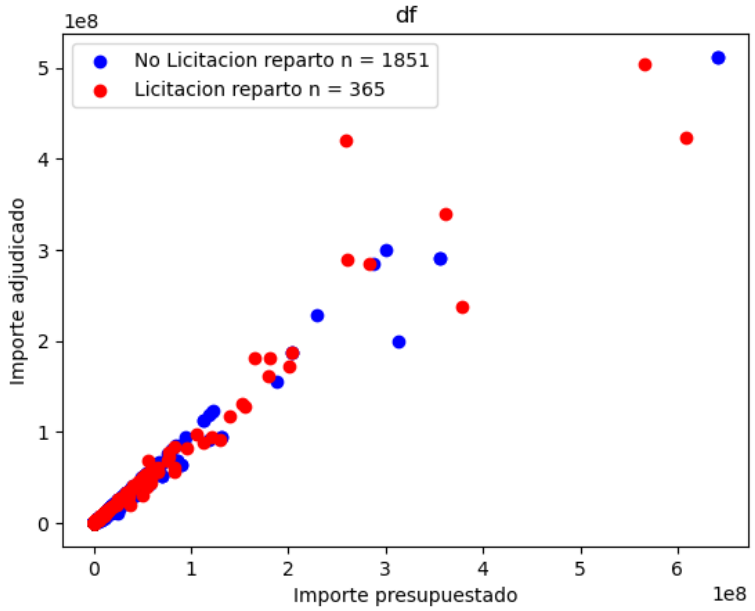

Figura 6: Gráfico X-Y de la base de datos.

[17] Un *flier* es una observación que es 1,5 veces mayor o menor que el Rango Intercuartil, que se calcula como el percentil 75 menos el percentil 25.

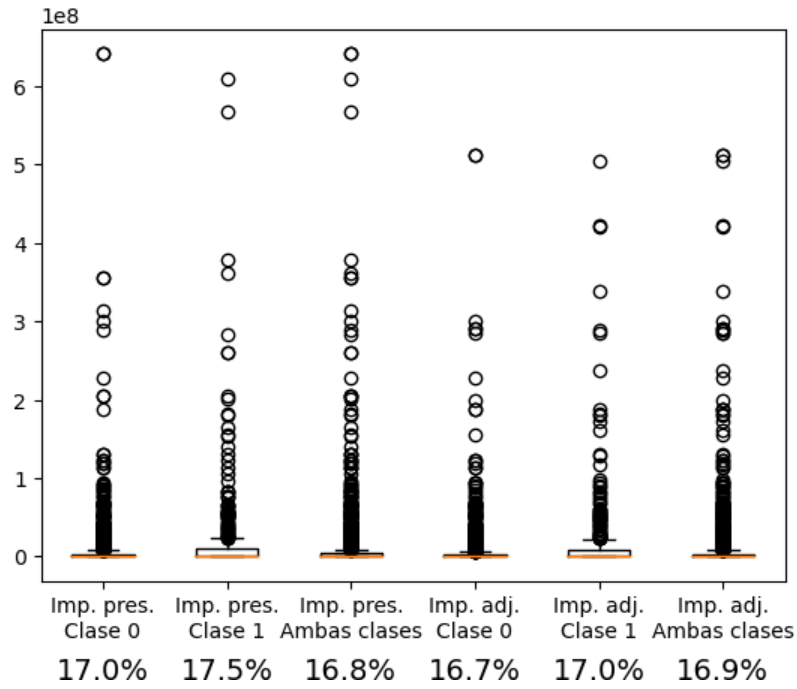

Figura 7: Diagrama de caja de la base de datos.

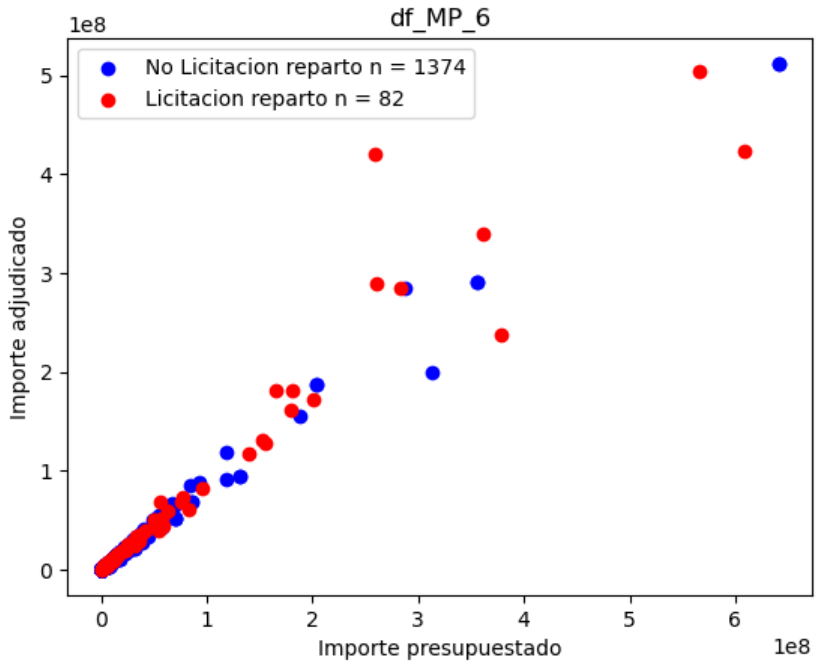

Figura 8: Gráfico X-Y de la base de datos cuando el predictor Mercado – Producto es igual a 6.

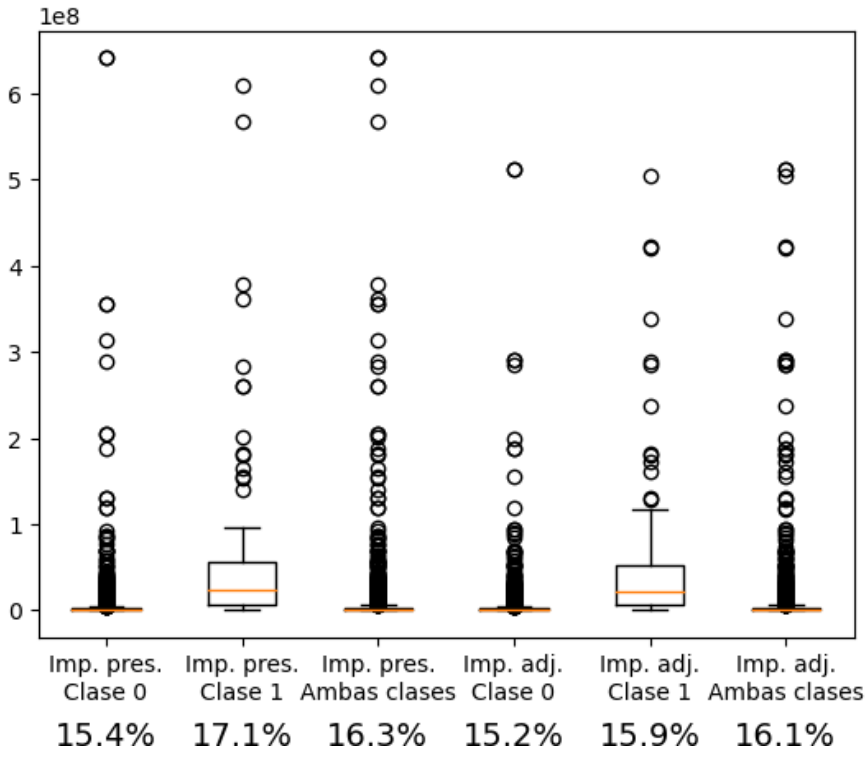

Figura 9: Diagrama de caja de la base de datos cuando el predictor Mercado – Producto es igual a 6.

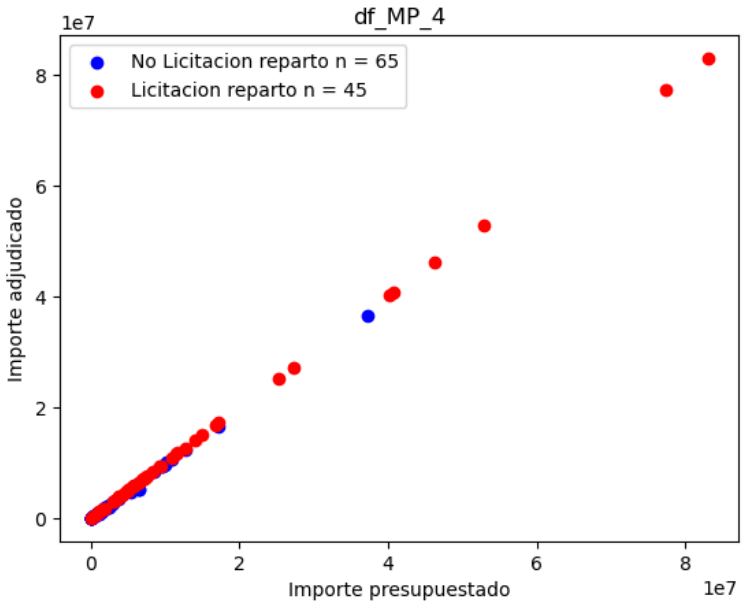

Figura 10: Gráfico X-Y de la base de datos cuando el predictor Mercado – Producto es igual a 4.

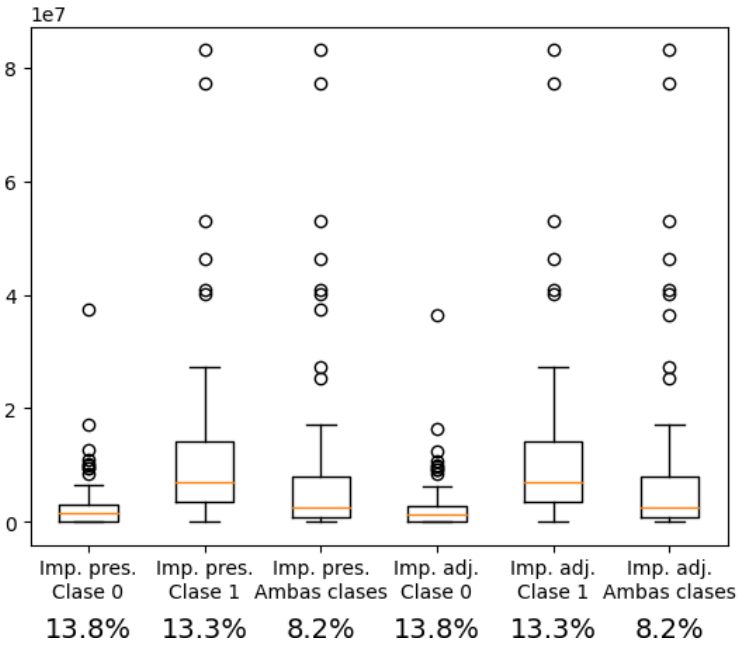

Figura 11: Diagrama de caja de la base de datos cuando el predictor Mercado – Producto es igual a 4.

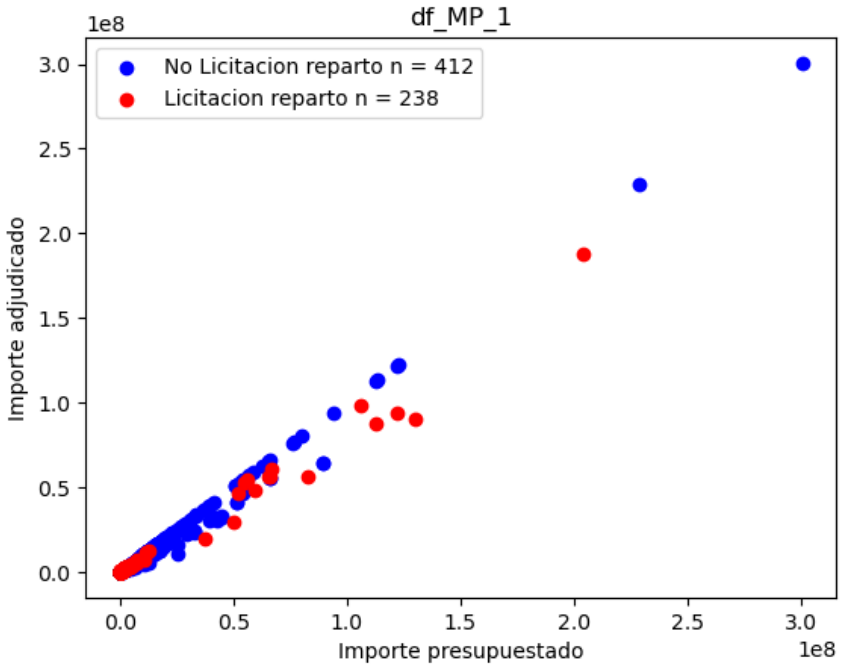

Figura 12: Gráfico X-Y de la base de datos cuando el predictor Mercado – Producto es igual a 1.

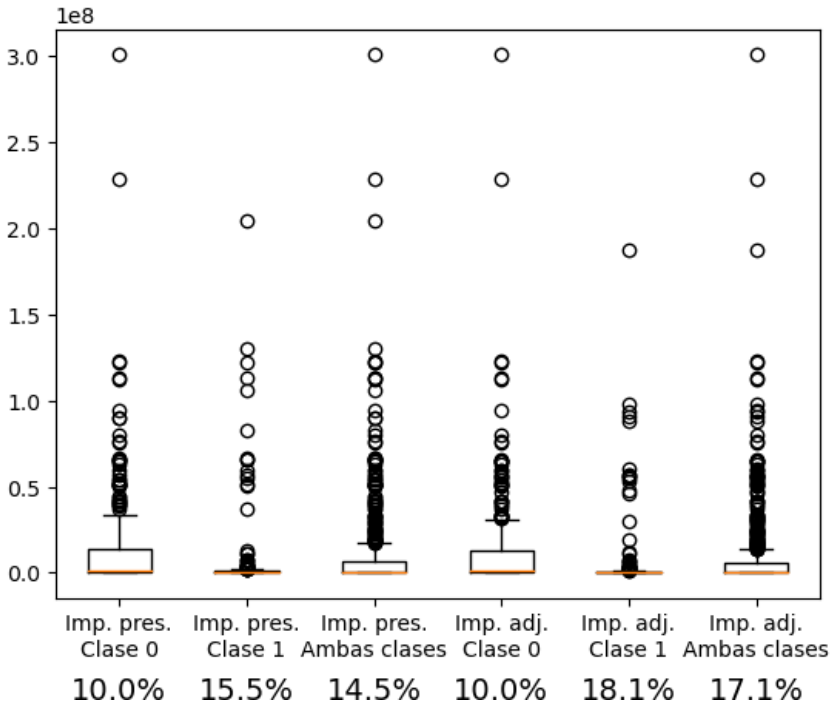

Figura 13: Diagrama de caja de la base de datos cuando el predictor Mercado – Producto es igual a 4.

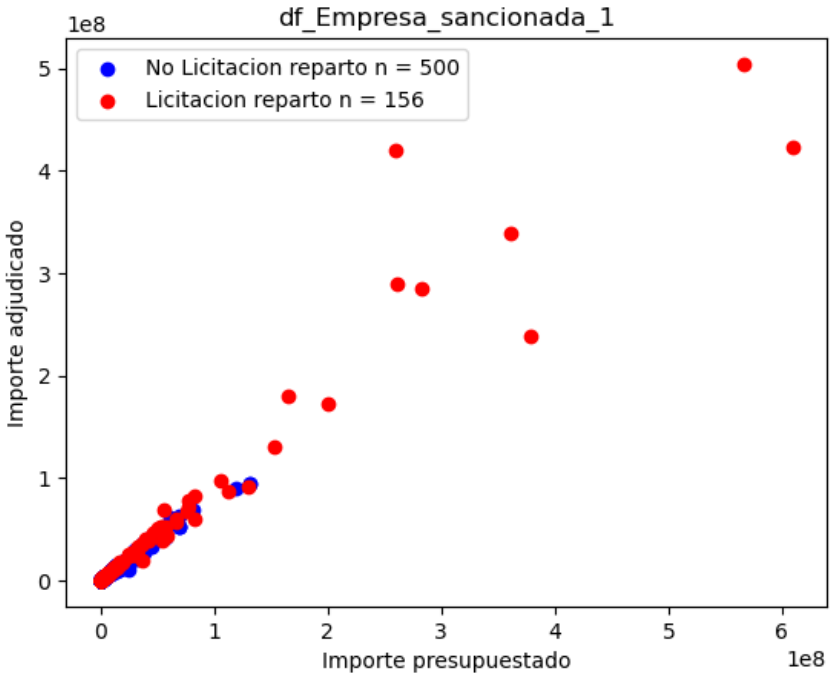

Figura 14: Gráfico X-Y de la base de datos cuando el predictor Empresa Sancionada es igual a 1.

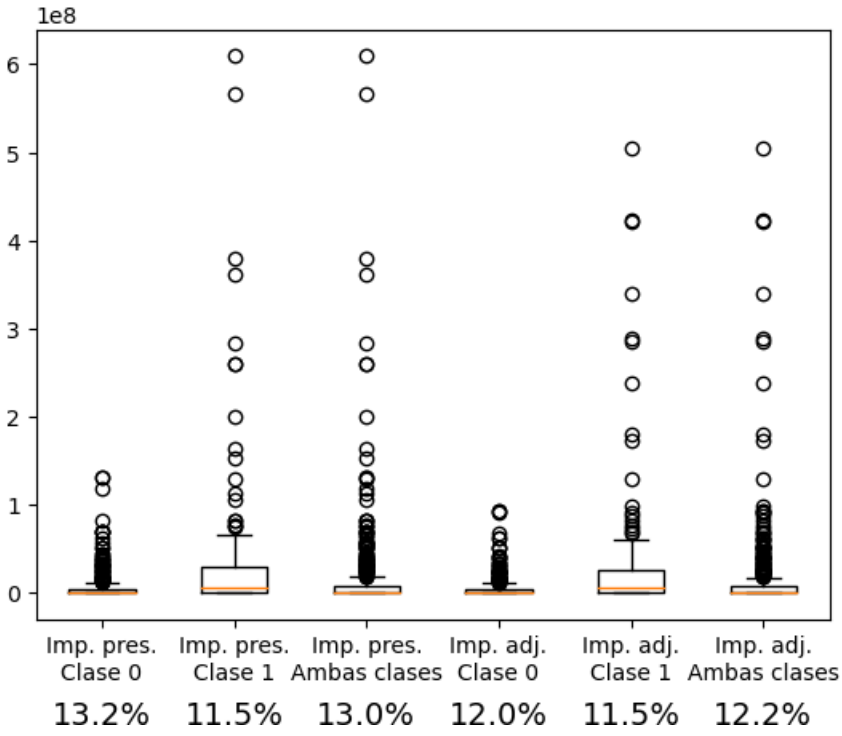

Figura 15: Diagrama de caja de la base de datos cuando el predictor Empresa Sancionada es igual a 1.

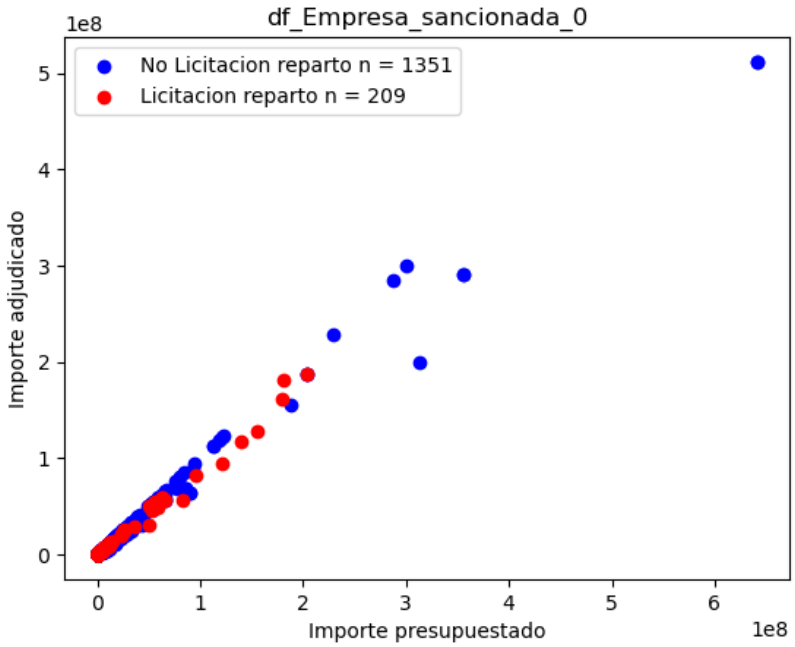

Figura 16: Gráfico X-Y de la base de datos cuando el predictor Empresa Sancionada es igual a 0.

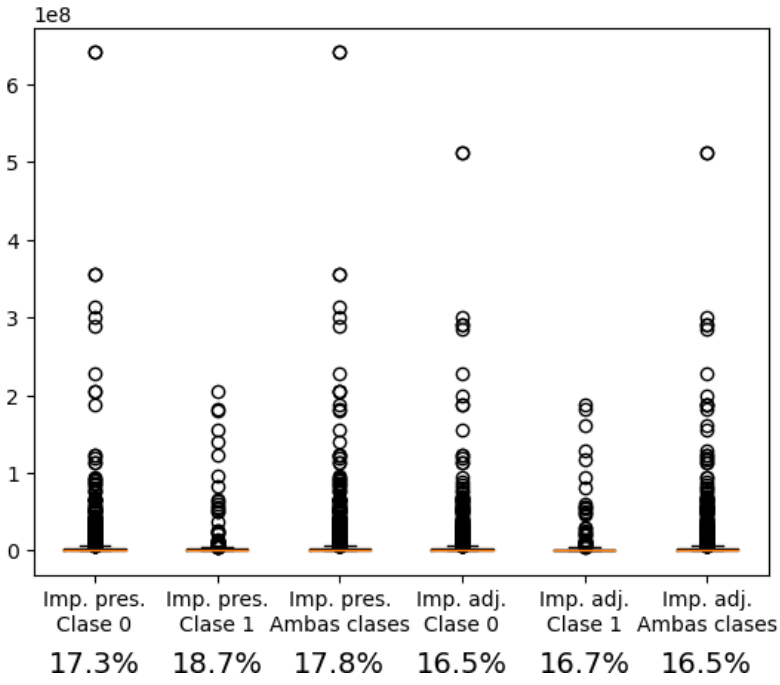

Figura 17: Diagrama de caja de la base de datos cuando el predictor Empresa Sancionada es igual a 1.

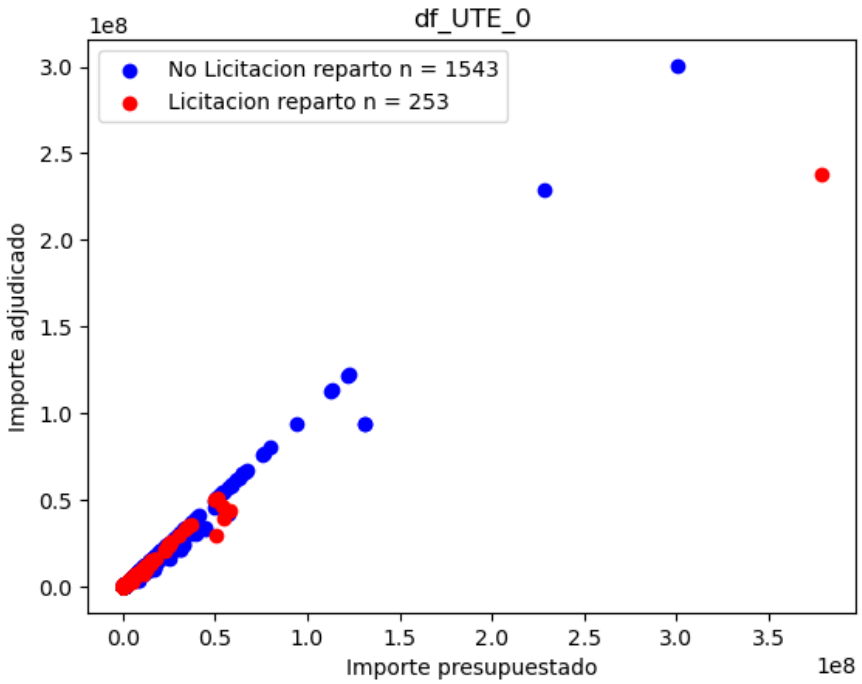

Figura 18: Gráfico X-Y de la base de datos cuando el predictor UTE es igual a 0.

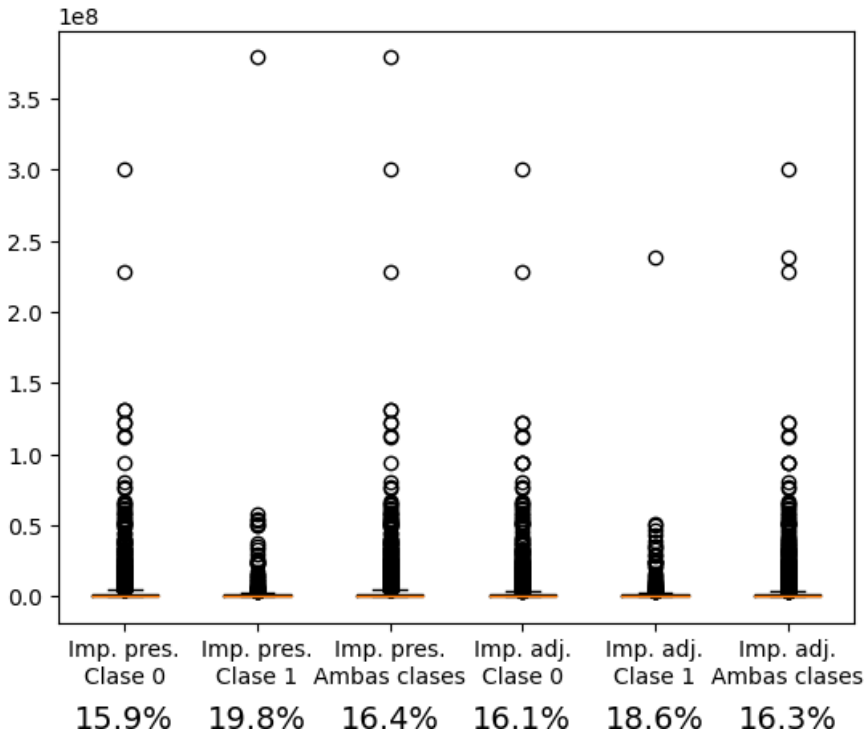

Figura 19: Diagrama de caja de la base de datos cuando el predictor el predictor UTE es igual a 0.

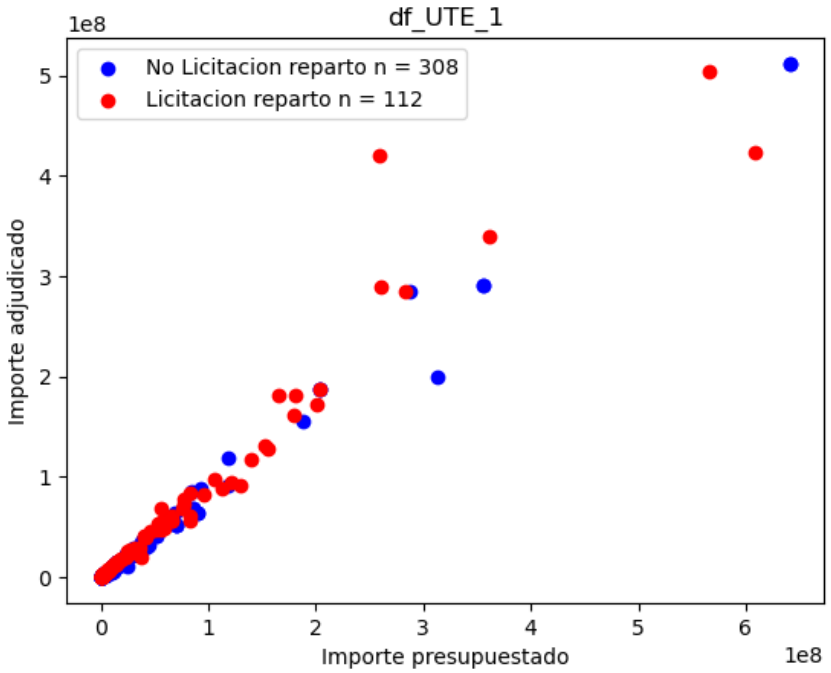

Figura 20: Gráfico X-Y de la base de datos cuando el predictor UTE es igual a 1.

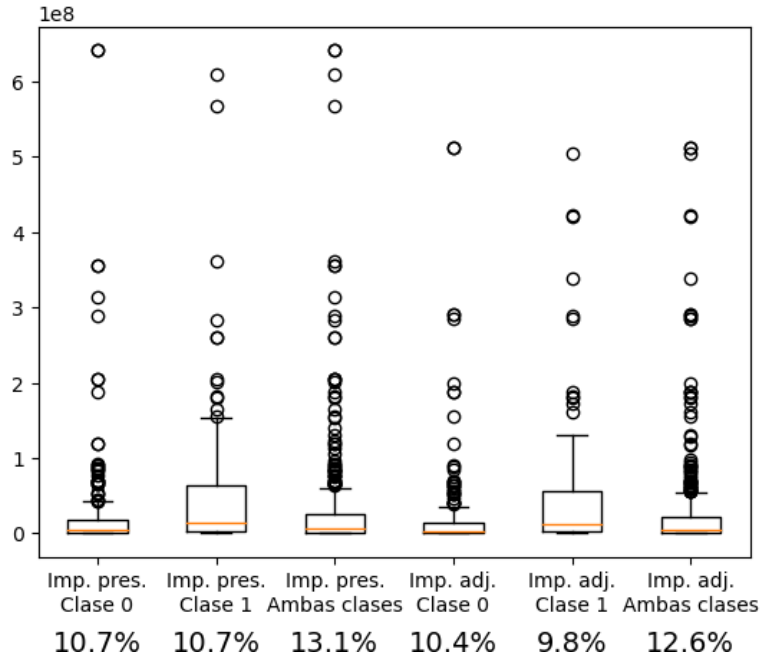

Figura 21: Diagrama de caja de la base de datos cuando el predictor UTE es igual a 1.

Ambas clases	*Importe presupuestado*	*Importe adjudicado*
Importe presupuestado	1,52E+15	1,29E+15
Importe adjudicado	1,29E+15	1,14E+15
No licitación reparto	Importe presupuestado	Importe adjudicado
Importe presupuestado	1E+15	8,4E+14
Importe adjudicado	8,4E+14	7,16E+14
Licitacion reparto	Importe presupuestado	Importe adjudicado
Importe presupuestado	4E+15	3,47E+15
Importe adjudicado	3,47E+15	3,2E+15

Tabla 3: Matrices de covarianzas por clase sin filtrar la base de datos.

Ambas clases	*Importe presupuestado*	*Importe adjudicado*
Importe presupuestado	2,00E+15	1,68E+15
Importe adjudicado	1,68E+15	1,47E+15
No licitación reparto	Importe presupuestado	Importe adjudicado
Importe presupuestado	1,12E+15	9,14E+14
Importe adjudicado	9,14E+14	7,52E+14
Licitacion reparto	Importe presupuestado	Importe adjudicado
Importe presupuestado	1,34E+16	1,15E+16
Importe adjudicado	1,15E+16	1,07E+16

Tabla 4: Matrices de covarianzas por clase cuando el predictor MP es igual a 6.

Ambas clases	*Importe presupuestado*	*Importe adjudicado*
Importe presupuestado	1,93E+14	1,93E+14
Importe adjudicado	1,93E+14	1,93E+14
No licitación reparto	Importe presupuestado	Importe adjudicado
Importe presupuestado	3,17E+13	3,09E+13
Importe adjudicado	3,09E+13	3,02E+13
Licitacion reparto	Importe presupuestado	Importe adjudicado
Importe presupuestado	3,62E+14	3,62E+14
Importe adjudicado	3,62E+14	3,62E+14

Tabla 5: Matrices de covarianzas por clase cuando el predictor MP es igual a 4.

Ambas clases	*Importe presupuestado*	*Importe adjudicado*
Importe presupuestado	6,55E+14	6,09E+14
Importe adjudicado	6,09E+14	5,75E+14
No licitación reparto	Importe presupuestado	Importe adjudicado
Importe presupuestado	7,16E+14	6,93E+14
Importe adjudicado	6,93E+14	6,78E+14
Licitacion reparto	Importe presupuestado	Importe adjudicado
Importe presupuestado	5,29E+14	4,42E+14
Importe adjudicado	4,42E+14	3,74E+14

Tabla 6: Matrices de covarianzas por clase cuando el predictor MP es igual a 1.

Ambas clases	*Importe presupuestado*	*Importe adjudicado*
Importe presupuestado	2,16E+15	1,86E+15
Importe adjudicado	1,86E+15	1,71E+15
No licitación reparto	Importe presupuestado	Importe adjudicado
Importe presupuestado	2,14E+14	1,67E+14
Importe adjudicado	1,67E+14	1,32E+14
Licitacion reparto	Importe presupuestado	Importe adjudicado
Importe presupuestado	7,74E+15	6,68E+15
Importe adjudicado	6,68E+15	6,19E+15

Tabla 7: Matrices de covarianzas por clase cuando el predictor Empresa_sancionada es igual a 1.

Ambas clases	*Importe presupuestado*	*Importe adjudicado*
Importe presupuestado	1,24E+15	1,05E+15
Importe adjudicado	1,05E+15	9,03E+14
No licitación reparto	Importe presupuestado	Importe adjudicado
Importe presupuestado	1,29E+15	1,09E+15
Importe adjudicado	1,09E+15	9,31E+14
Licitacion reparto	Importe presupuestado	Importe adjudicado
Importe presupuestado	9,20E+14	8,13E+14
Importe adjudicado	8,13E+14	7,26E+14

Tabla 8: Matrices de covarianzas por clase cuando el predictor Empresa_sancionada es igual a 0.

Ambas clases	*Importe presupuestado*	*Importe adjudicado*
Importe presupuestado	2,96E+14	2,56E+14
Importe adjudicado	2,56E+14	2,31E+14
No licitación reparto	Importe presupuestado	Importe adjudicado
Importe presupuestado	2,37E+14	2,28E+14
Importe adjudicado	2,28E+14	2,21E+14
Licitacion reparto	Importe presupuestado	Importe adjudicado
Importe presupuestado	6,53E+14	4,32E+14
Importe adjudicado	4,32E+14	2,93E+14

Tabla 9: Matrices de covarianzas por clase cuando el predictor UTE es igual a 0.

Ambas clases	*Importe presupuestado*	*Importe adjudicado*
Importe presupuestado	6,15E+15	5,23E+15
Importe adjudicado	5,23E+15	4,62E+15
No licitación reparto	Importe presupuestado	Importe adjudicado
Importe presupuestado	4,55E+15	3,69E+15
Importe adjudicado	3,69E+15	3,02E+15
Licitacion reparto	Importe presupuestado	Importe adjudicado
Importe presupuestado	9,86E+15	8,74E+15
Importe adjudicado	8,74E+15	8,28E+15

Tabla 10: Matrices de covarianzas por clase cuando el predictor UTE es igual a 1.

Gráficamente no se encuentra una separabilidad suficientemente significativa. Ni siquiera tras haber filtrado la base de datos por cada valor único de cada predictor categórico. Se observa dos tipos de dispersiones en los gráficos:

1. Para valores altos de ambos predictores, hay más dispersión en los datos y pocas observaciones.

2. Para el resto de valores, las observaciones se concentran entorno a la bisectriz.

Los datos son muy dispersos. El 17 % de las observaciones son fliers. Este porcentaje disminuye filtrando el conjunto de datos hasta en un 7%. Las matrices de covarianzas muestran una mayor dispersión para todos las particiones realizadas en las licitaciones objeto de reparto, excepto en las observaciones donde se excluyen las empresas sancionadas. Hay pocos registros de licitaciones en el sector de desvíos ferroviarios. 65 licitaciones no objeto de reparto y 45 objeto de reparto.

4.2.2 Predictores categóricos

Se muestran gráficos de barras (Figura 21 a Figura 23) que miden el recuento de las observaciones por valor único de predictor categórico y clase.

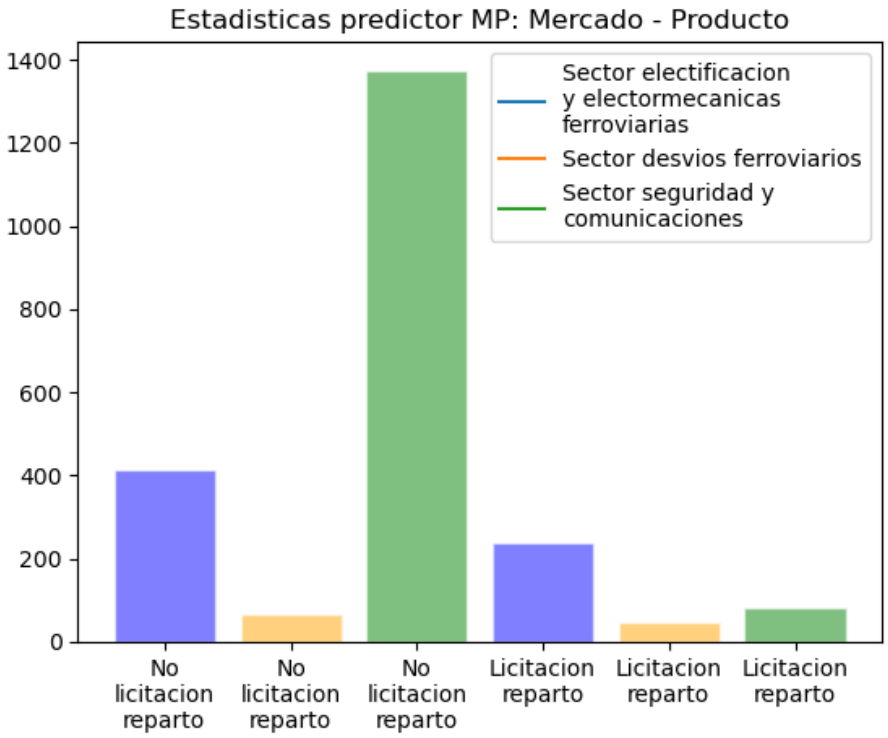

Figura 22: Gráfico de barras de recuento del predictor Mercado Producto.

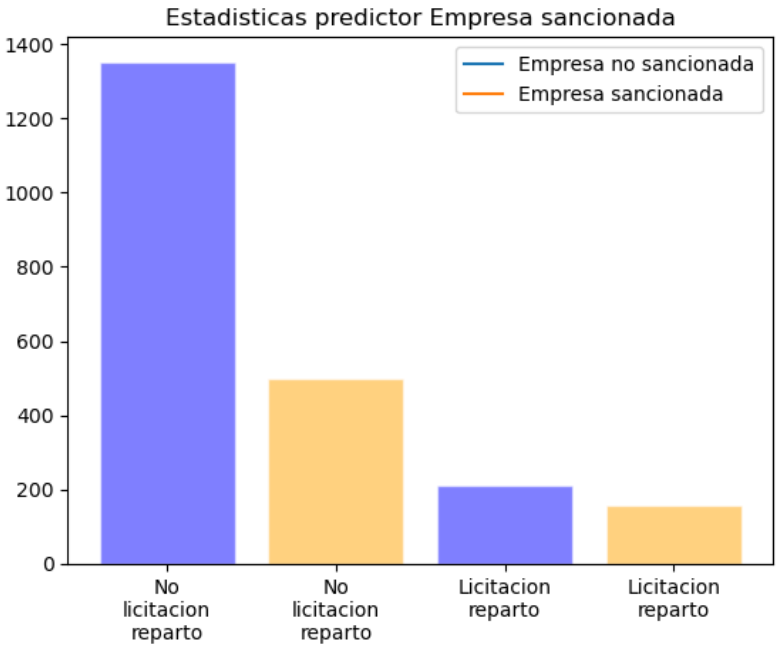

Figura 23: Gráfico de barras de recuento del predictor Empresa sancionada.

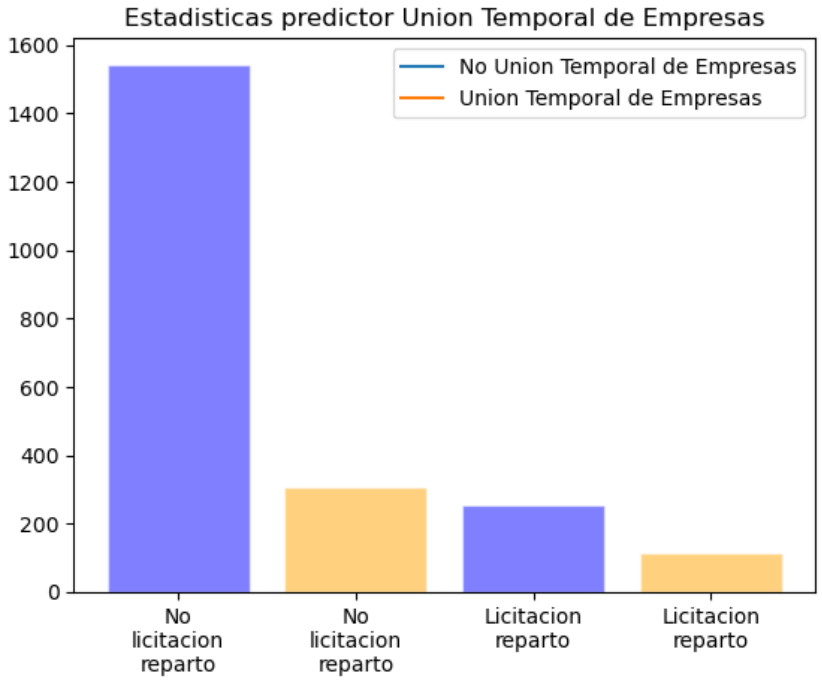

Figura 24: Gráfico de barras de recuento del predictor UTE.

Los registros de la base de datos se concentran en la clase no licitación reparto. Destaca que el número de licitaciones ilícitas que han sido adjudicadas a empresas no sancionadas es mayor que el número de licitaciones ilícitas si la parte adjudicataria hubiera sido sancionada.

Para justificar estadísticamente la aplicación de estos predictores categóricos en el análisis, se justifica mediante la prueba Chi – Cuadrado de las tablas de contingencia asociadas a cada clase (Tabla 11 a Tabla 13) que todos y cada uno de ellos son estadísticamente significativos. La hipótesis nula de la prueba chi – cuadrado es que hay una relación estadísticamente significativa entre la clase y los predictores categóricos. Se rechaza la hipótesis nula para todos los predictores categóricos con un nivel de significatividad de 0.01.

MP	*No licitacion reparto*	*Licitacion reparto*	*Total*
Sector electrificación y electromecanicas ferroviarias	412	238	650
Sector desvíos ferroviarios	65	45	110
Sector seguridad y comunicaciones	1374	82	1456
Total	1851	365	2216

Tabla 11: Tabla de contingencia del predictor categórico MP.

Empresa sancionada	*No licitacion reparto*	*Licitacion reparto*	*Total*
Empresa no sancionada	1351	209	1560
Empresa sancionada	500	156	656
Total	1851	365	2216

Tabla 12: Tabla de contingencia del predictor categórico Empresa sancionada.

UTE	*No lictiacion reparto*	*Licitacion reparto*	*All*
No Union Temporal de Empresas	1543	253	1796
Union Temporal de Empresas	308	112	420
Total	1851	365	2216

Tabla 13: Tabla de contingencia del predictor categórico UTE.

5 Experimentación

En[18], se ofrece un enlace de Github donde se puede examinar todo el código escrito para la elaboración de este apartado, así como la base de datos utilizada.

18 https://github.com/Felixxx00/TFM_Prediccion_licitaciones_ilicitas_en_el_sector_ferroviario/tree/main/Experimentacion

Los hiperparámetros que se han utilizado son los que se muestran en las siguientes tablas. (Tablas 14 a 17).

Clasificador	*Hiperparámetro*	*Valor*
Naïve Bayes Gaussiano	priors	None
	var_smoothing	1.00E-09

Tabla 14: Hiperparámetros aplicados en el clasificador Naïve Bayes Gaussiano

Clasificador	*Hiperparámetro*	*Valor*
Máquinas de Vector Soporte	C	1.0
	break_ties	False
	cache_size	200
	class_weight	None
	coef0	0.0
	decision_function_shape	'ovr'
	degree	3
	gamma	'scale'
	kernel	'rbf'
	max_iter	-1
	probability	False
	random_state	123
	shrinking	True
	tol	0.001
	verbose	False

Tabla 15: Hiperparámetros aplicados en el clasificador Máquinas de Vector Soporte

Clasificador	*Hiperparámetro*	*Valor*
Random Forest	bootstrap	True
	ccp_alpha	0.0
	class_weight	'gini'
	criterion	None
	max_depth	'auto'
	max_features	'auto'
	max_leaf_nodes	None
	max_samples	None
	min_impurity_decrease	0.0
	min_samples_leaf	1
	min_samples_split	2
	min_weight_fracftion_leaf	0.0
	n_estimators	100
	n_jobs	None
	oob_score	False
	random_state	123
	verbose	0
	warm_start	False

Tabla 16: Hiperparámetros aplicados en el clasificador *Random Forest*

Clasificador	*Hiperparámetro*	*Valor*
Gradient Tree Boosting	ccp_alpha	0.0
	criterion	'friedman mse'
	init	None
	learning rate	0.1
	loss	'deviance'
	max_depth	3
	max features	None
	max_leaf_nodes	None
	min_impurity_decrease	0.0
	min_samples_leaf	1
	min_samples_split	2
	min_wight_fraction_leaf	0.0
	n_estimators	100
	n_iter_no_change	None
	random_state	123
	subsample	1.0
	tol	0.0001
	validation_fraction	0.1
	verbose	0
	warm_start	False

Tabla 17: Hiperparámetros aplicados en el clasificador *Gradient Tree Boosting*

Se ha controlado la aleatoriedad de los clasificadores mediante la semilla 123. No se ha realizado un ajuste exhaustivo de los hiperparámetros debido al extenso tiempo de ejecución que hubiera supuesto. Hay que recordar que por cada clasificador se está utilizando el método *Leave One Out* para que se utilice toda la información de la base de datos para clasificar cada una de las instancias. Este método hace que la cantidad de veces que se ejecuta cada clasificador para obtener

los resultados de las métricas pertinentes sea igual al número de observaciones totales de la Base de Datos, es decir, 2216 veces en el caso de no tratar el desbalanceo de clases, 3702 veces en el caso de tratar el desbalanceo de clases por sobre muestreo y 730 veces en el caso de tratar el desbalanceo de clases por infra muestreo. Se puede consultar los tiempos de ejecución que ha conllevado cada clasificador en cada escenario de balanceo o desbalanceo de las clases en el Anexo II.

La información pública disponible es escasa[19] y aunque se haya obtenido el rendimiento óptimo para cada clasificador, no habría suficientes evidencias estadísticas como para concluir que una licitación es o no ilícita en términos realistas.

Se representa las curvas ROC en donde se señala la métrica AUC y el *Threshold* óptimo para cada uno de los clasificadores en cada escenario de balanceo o desbalanceo de clases. El *Threshold* óptimo es aquella probabilidad (o valor de confianza en el caso de la Máquina de Vector Soporte) que maximiza la sensibilidad y minimiza 1 - especificidad. Es decir, aquel valor que representado en el gráfico ROC esté más cerca del punto (0,1).

[19] Destaca que en el TFM que realiza Villa Pedroza (2021) haya mucha más información disponible, en términos de número de predictores, en un país que en términos generales es más corrupto que España de acuerdo con Transparency International.

5.1 ROC por conjunto de datos: Clasificador Naïve Bayes Gaussiano

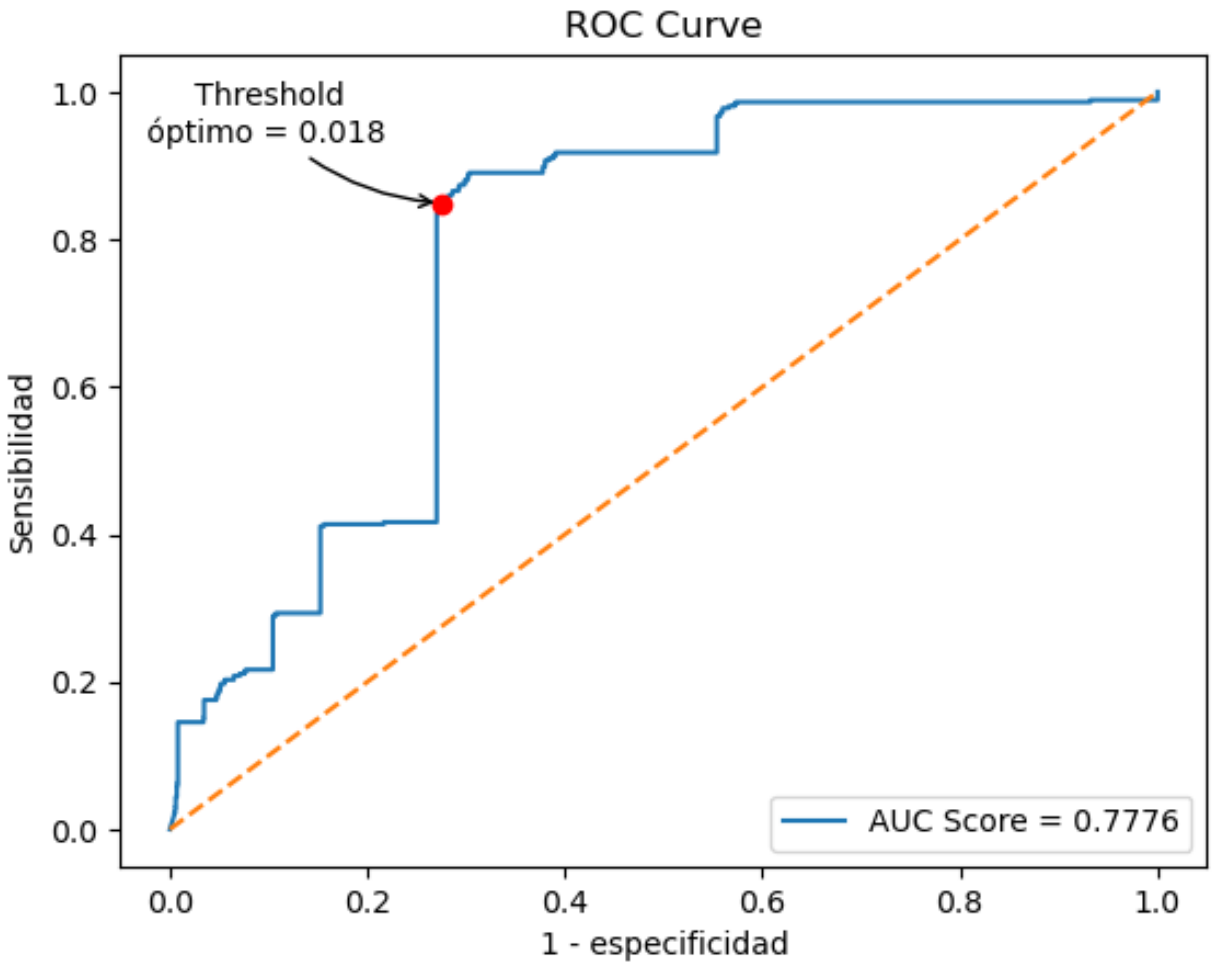

Figura 25: Curva ROC del Clasificador Naïve Bayes Gaussiano en el conjunto de datos original.

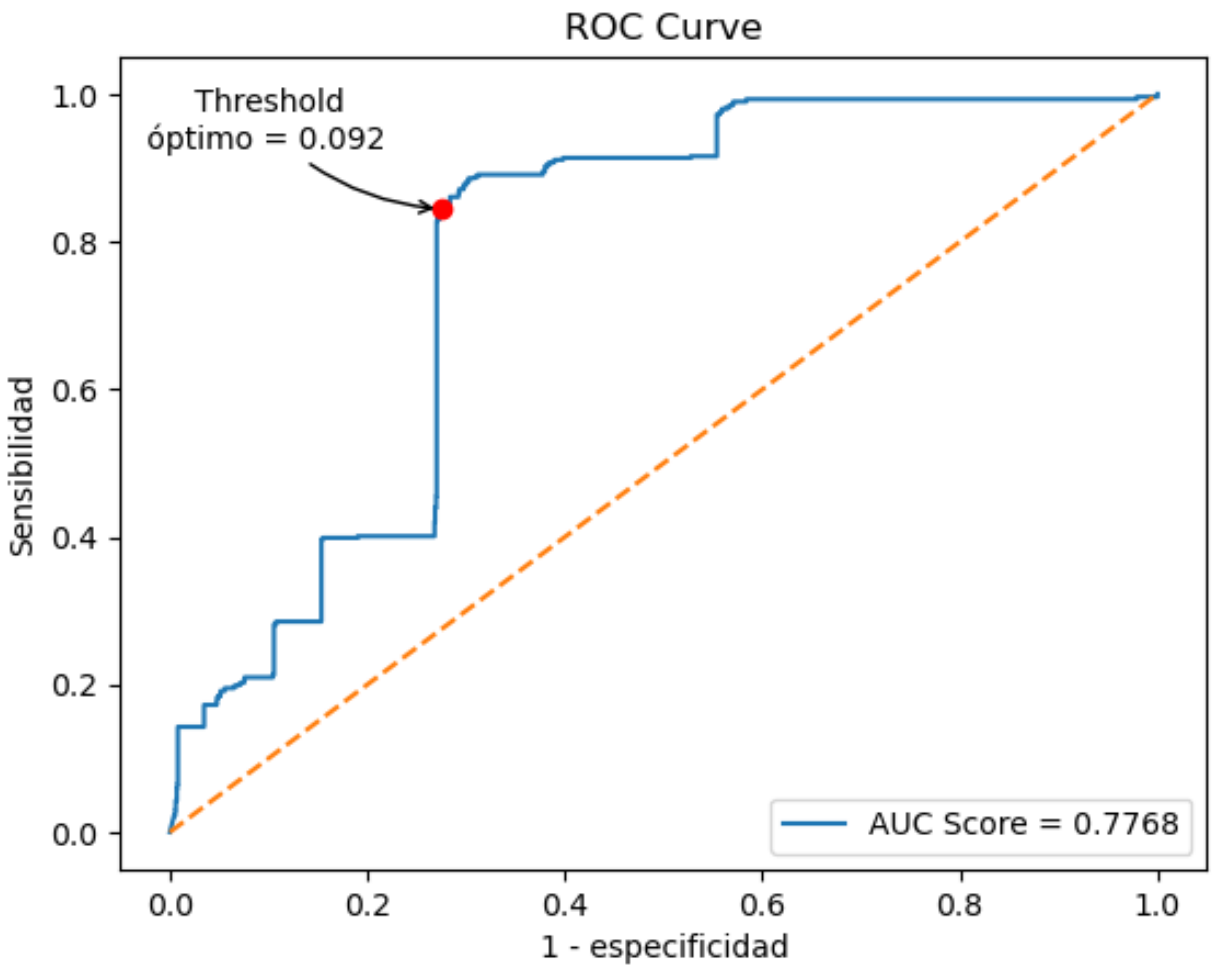

Figura 26: Curva ROC del Clasificador Naïve Bayes Gaussiano en el conjunto de datos balanceado por sobre muestreo.

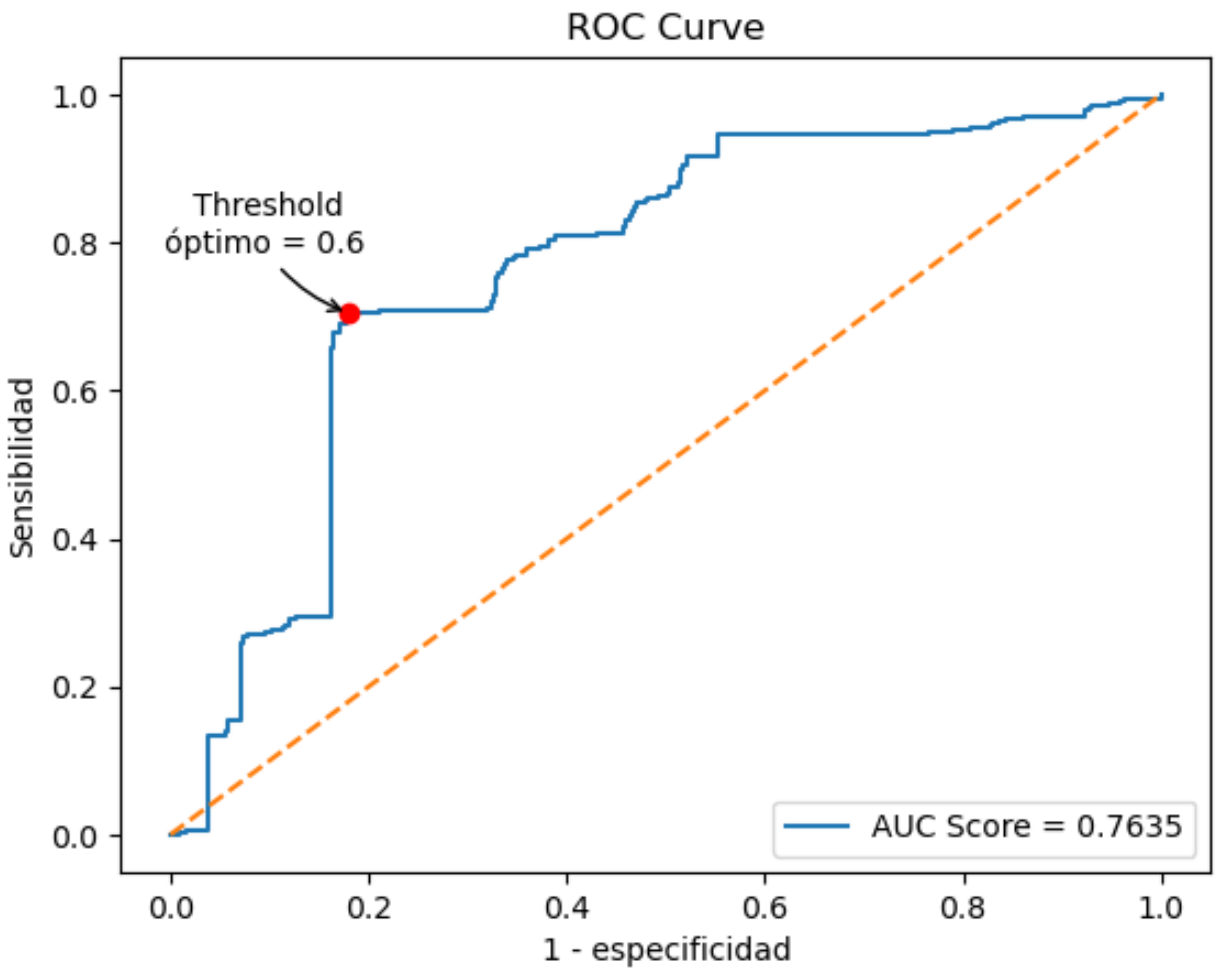

Figura 27: Curva ROC del Clasificador Naïve Bayes Gaussiano en el conjunto de datos balanceado por infra muestreo.

5.2 ROC por conjunto de datos: Clasificador Máquinas de Vector Soporte

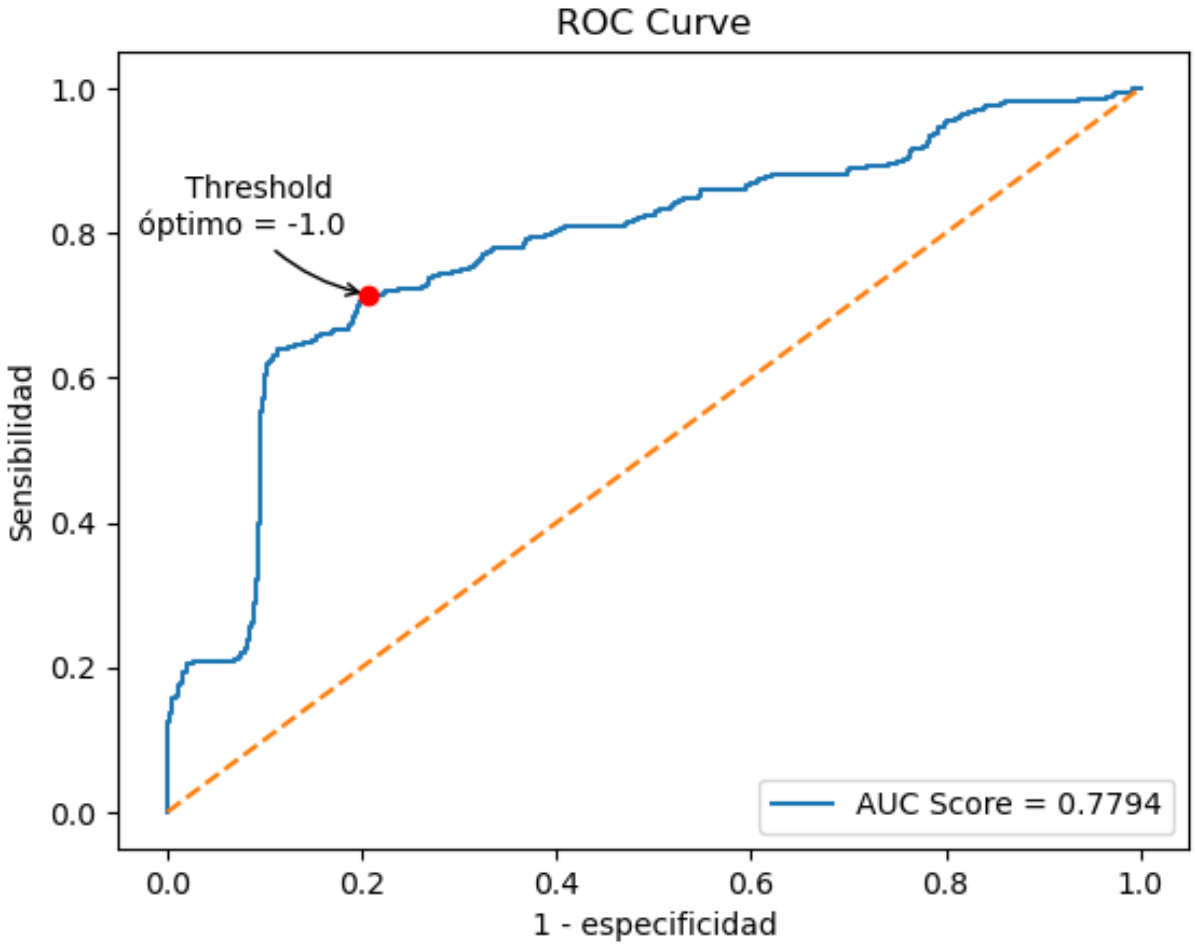

Figura 28: Curva ROC del Clasificador Máquinas de Vector Soporte en el conjunto de datos original.

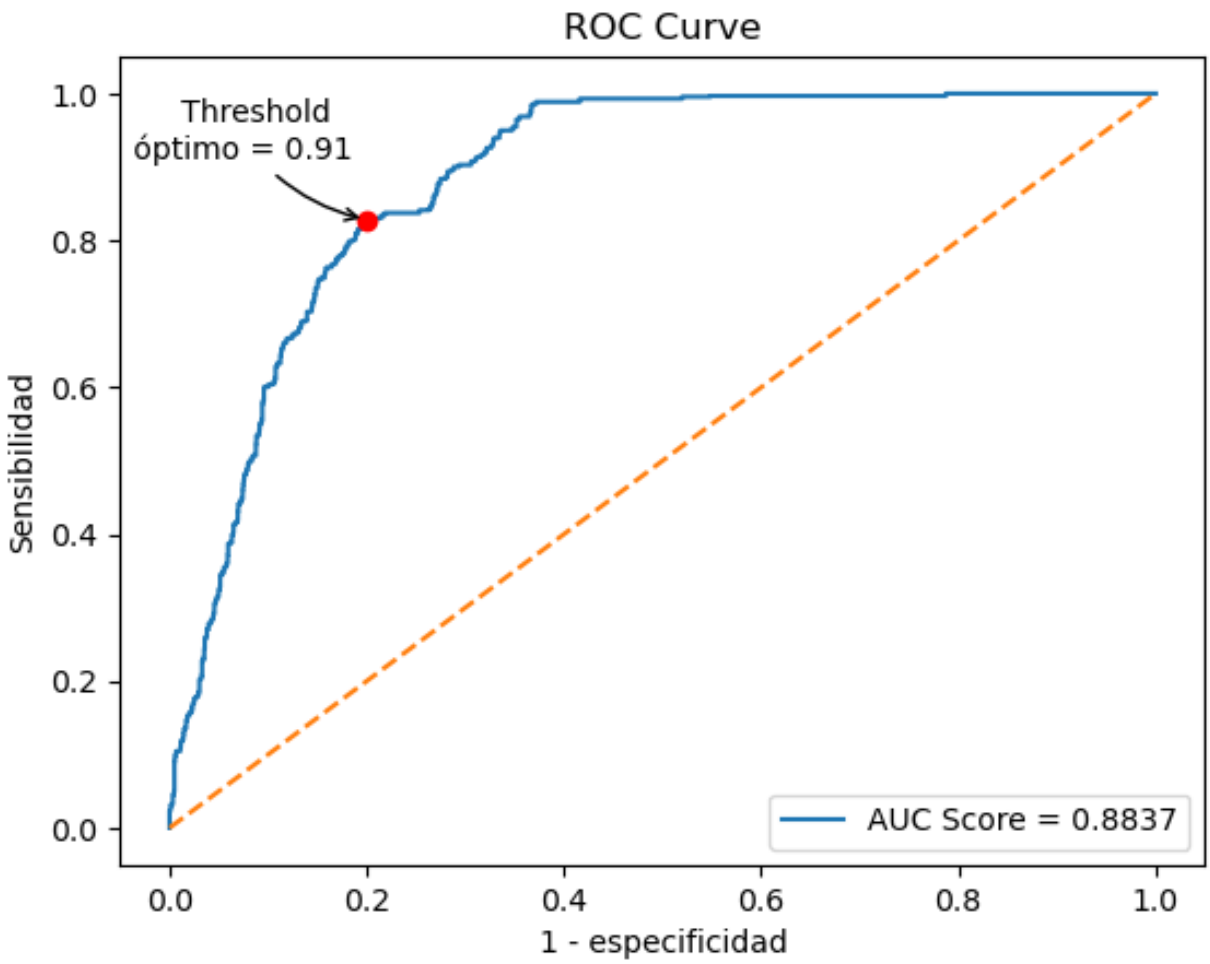

Figura 29: Curva ROC del Clasificador Máquinas de Vector Soporte en el conjunto de datos balanceado por sobre muestreo.

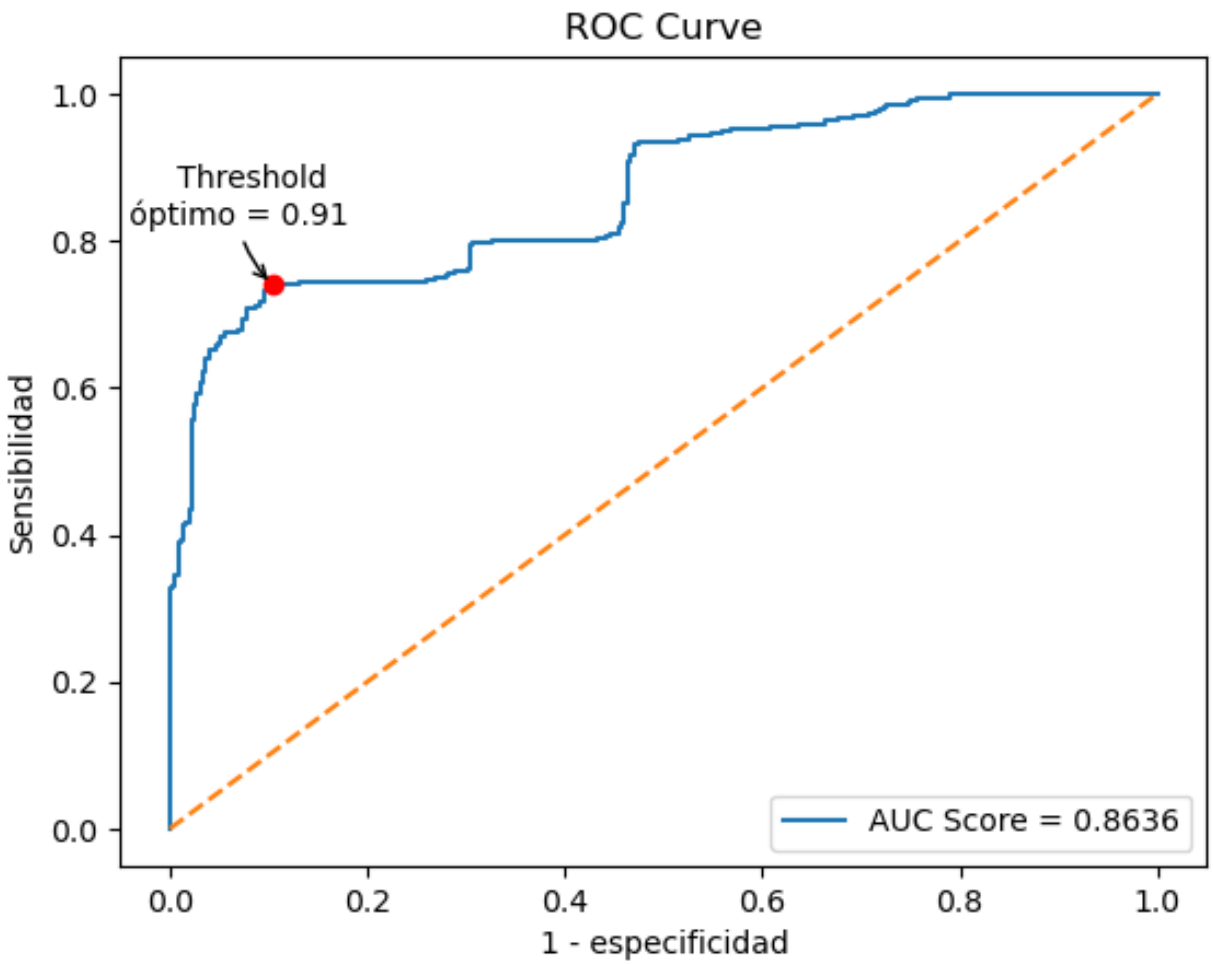

Figura 230: Curva ROC del Clasificador Máquinas de Vector Soporte en el conjunto de datos balanceado por infra muestreo.

5.3 ROC por conjunto de datos: Clasificador *Random Forest*

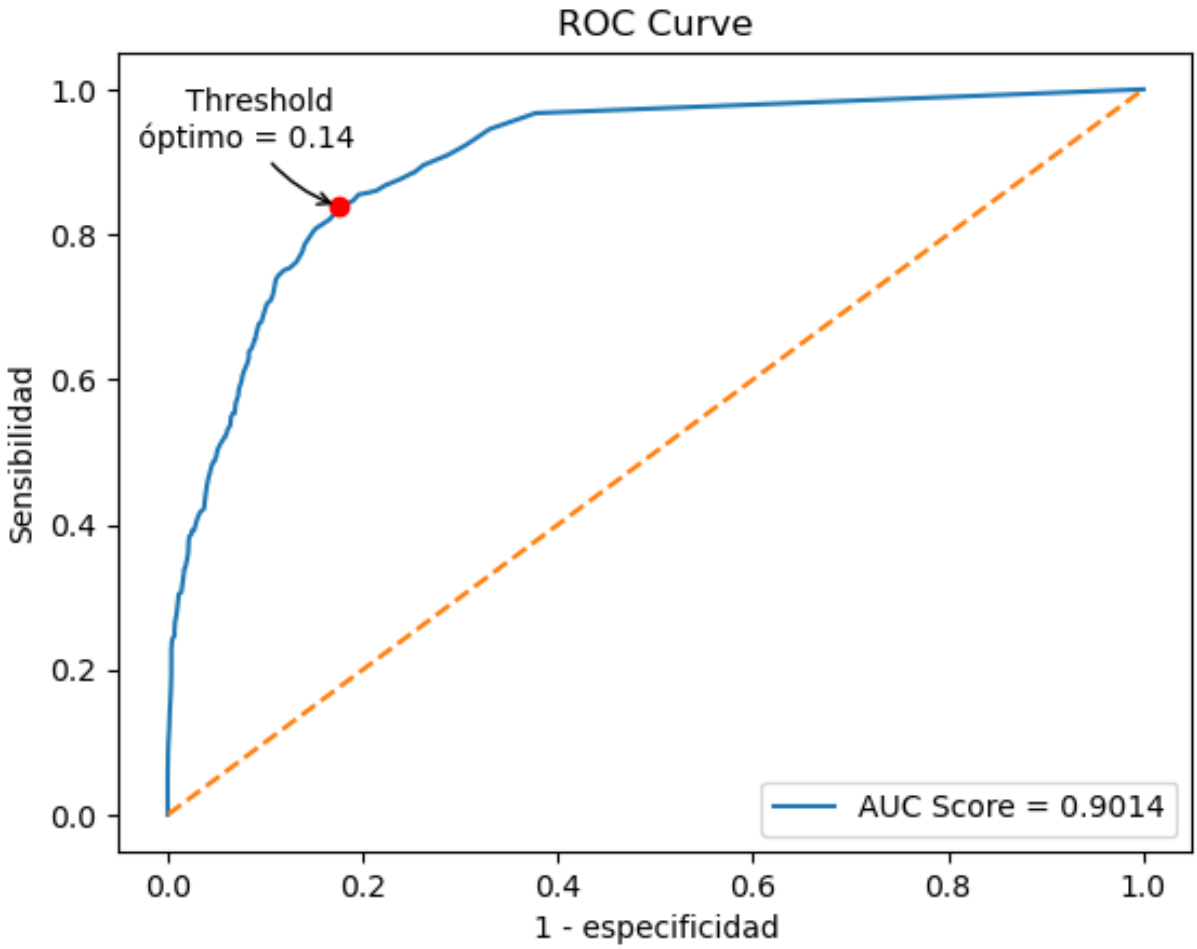

Figura 31: Curva ROC del Clasificador *Random Forest* en el conjunto de datos original

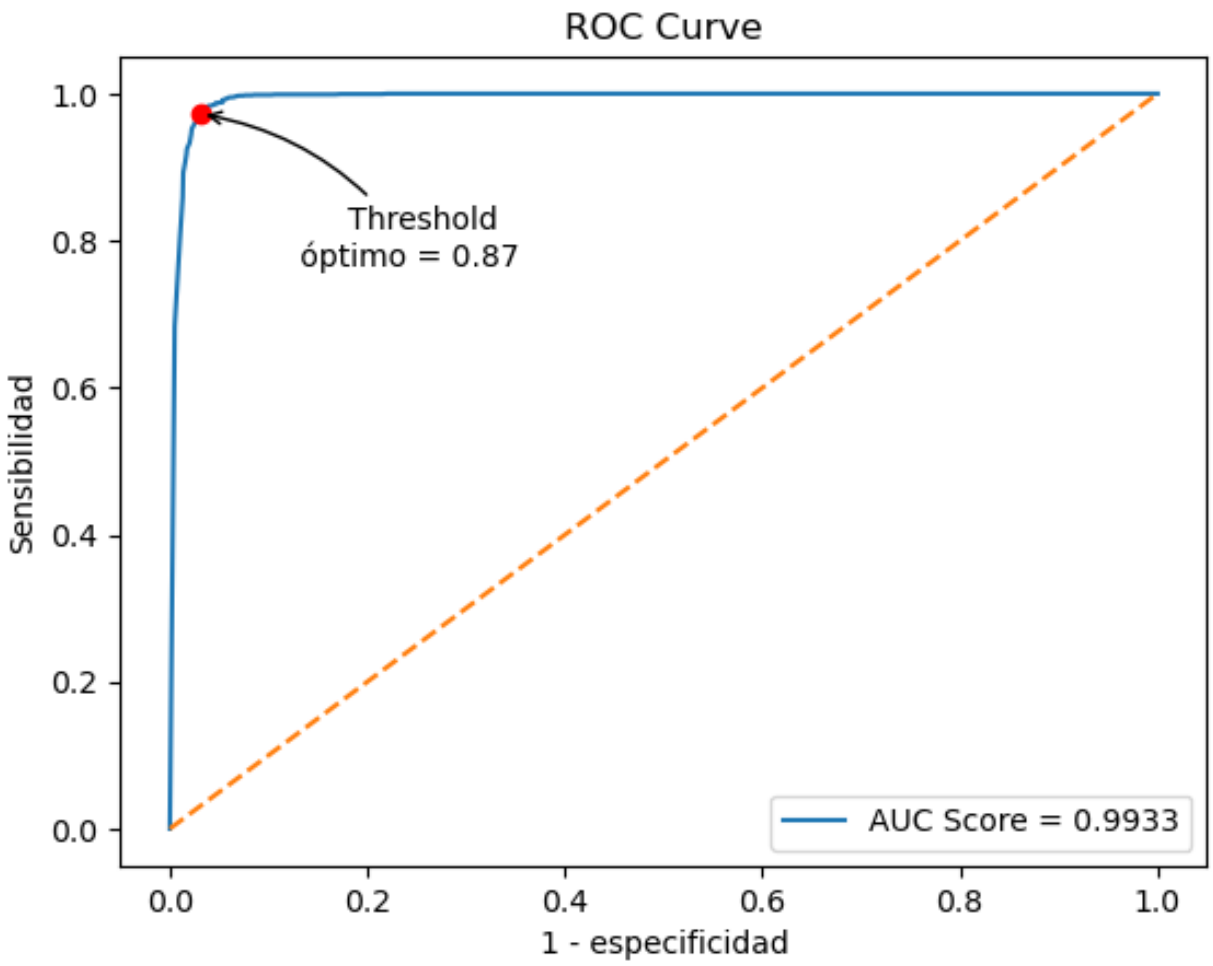

Figura 32: Curva ROC del Clasificador *Random Forest* en el conjunto de datos balanceado por sobre muestreo.

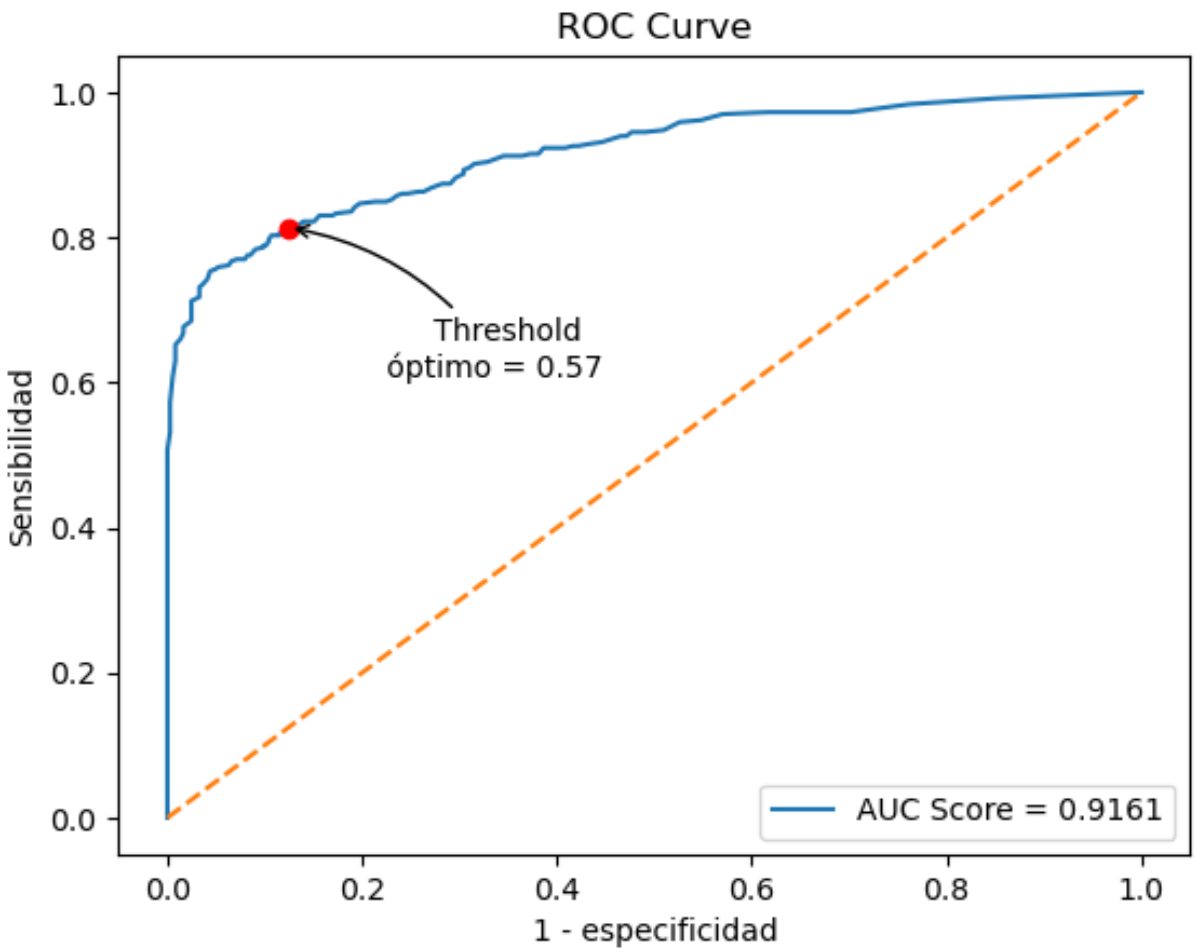

Figura 33: Curva ROC del Clasificador *Random Forest* en el conjunto de datos balanceado por infra muestreo.

5.4 ROC por conjunto de datos: Clasificador *Gradient Boosting Tree*

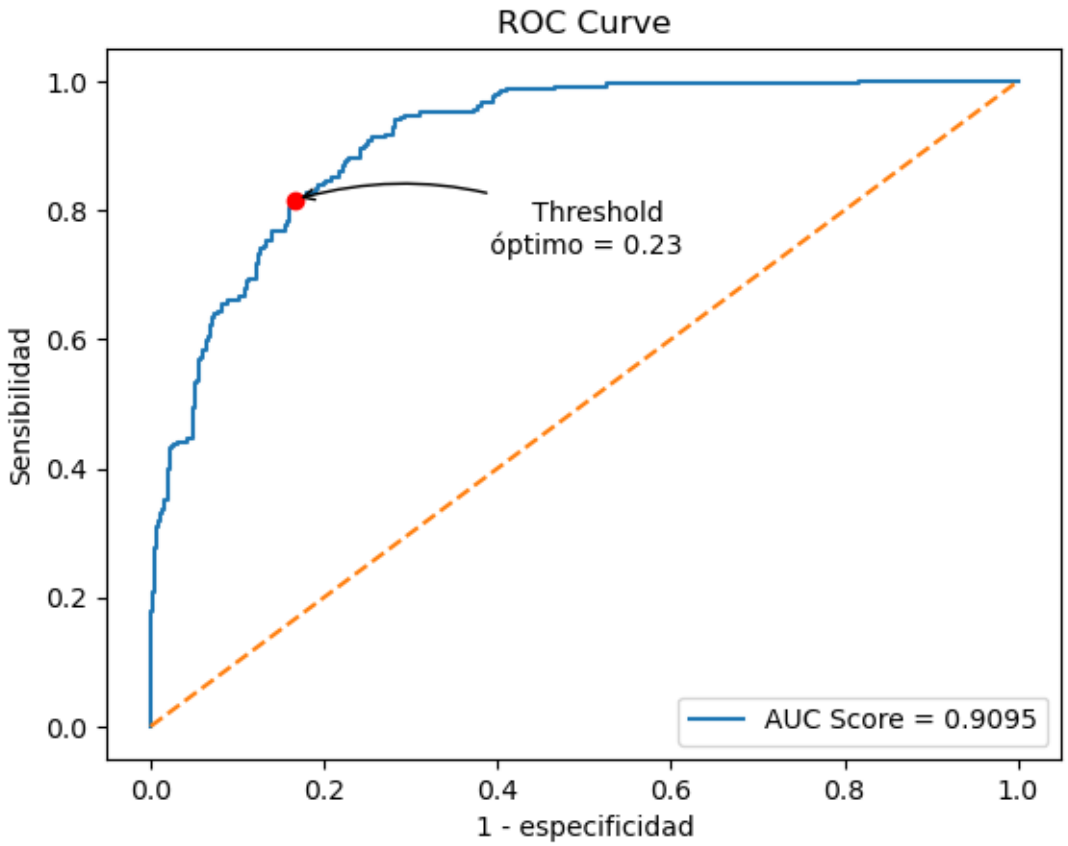

Figura 34: Curva ROC del Clasificador *Gradient Boosting Tree* en el conjunto de datos original

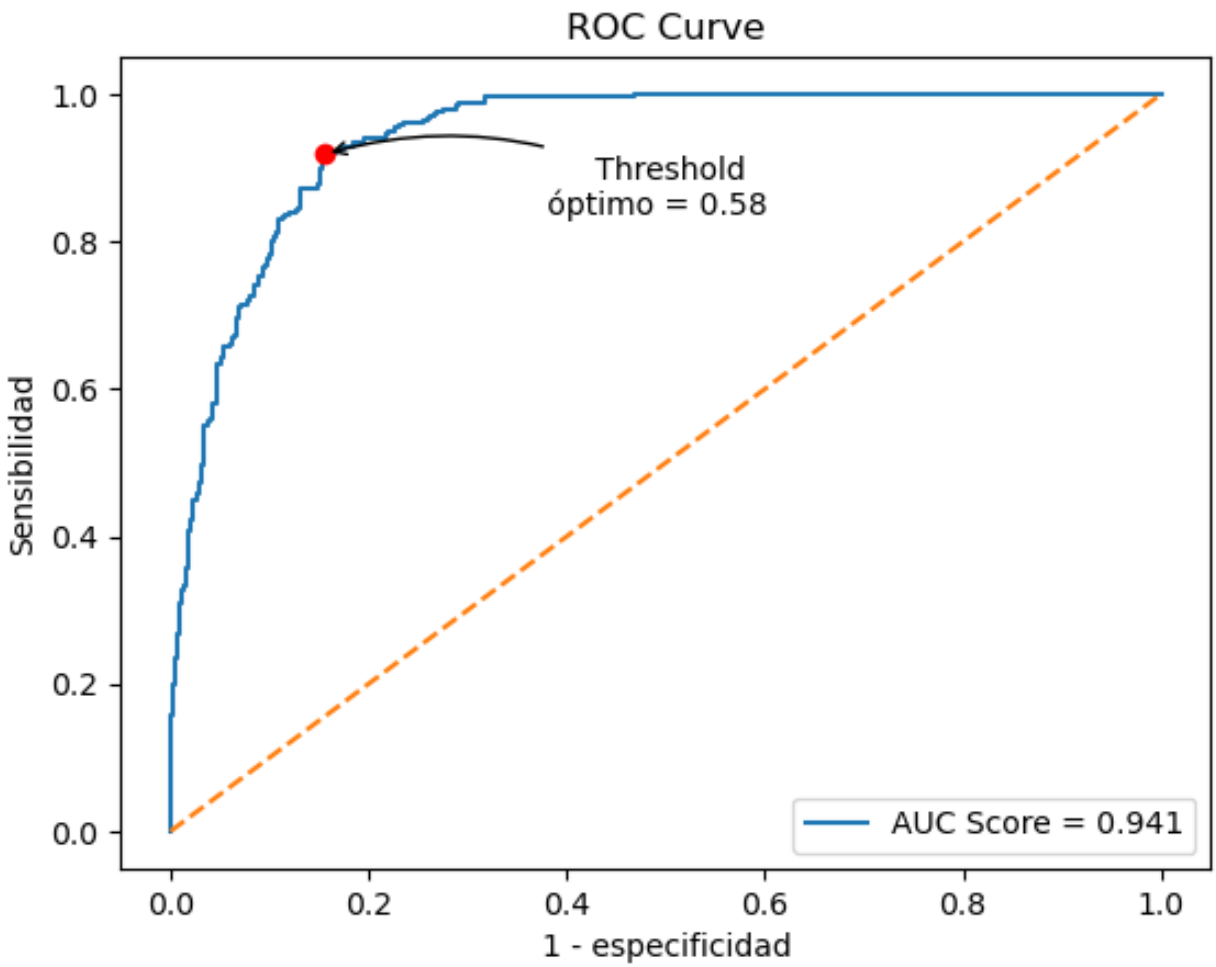

Figura 35: Curva ROC del Clasificador *Gradient Boosting Tree* en el conjunto de datos balanceado por sobre muestreo.

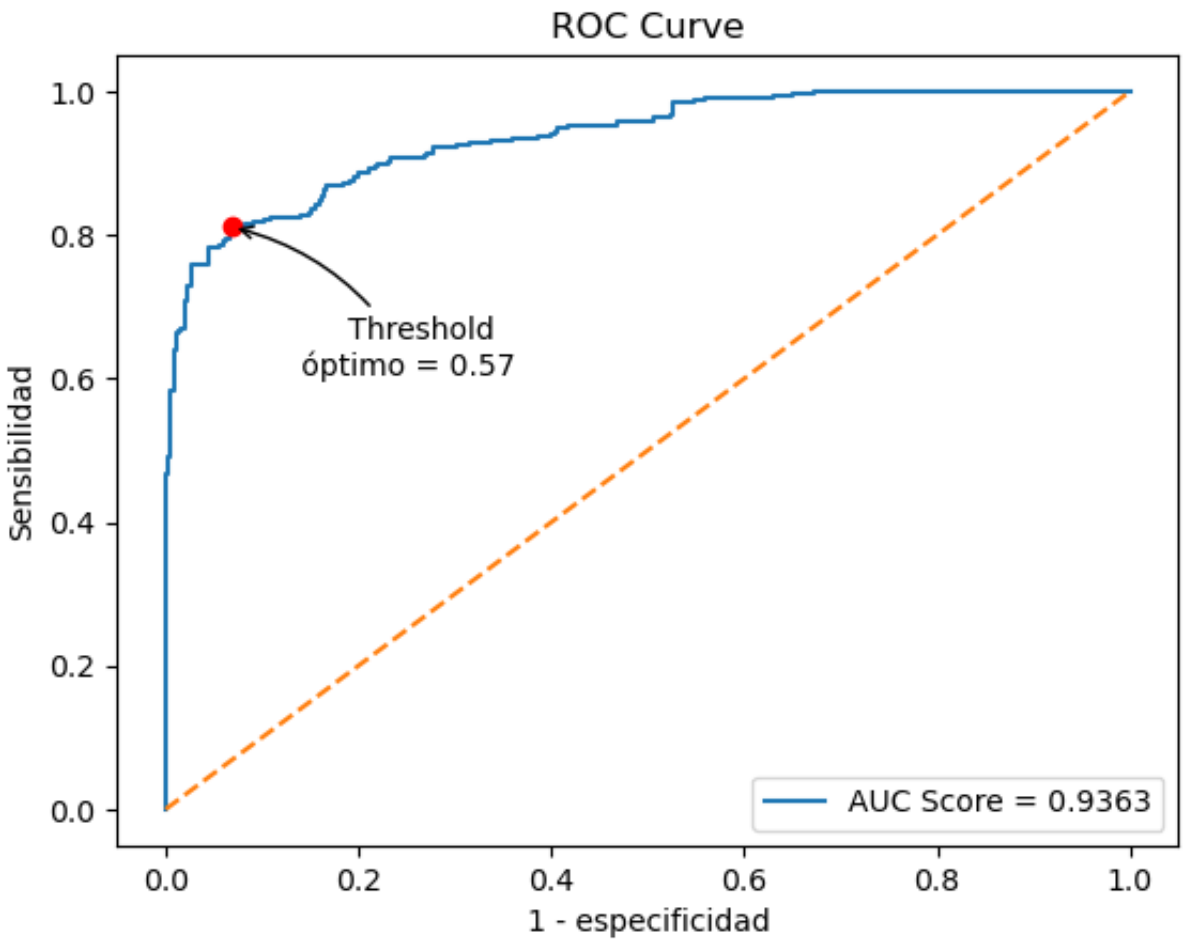

Figura 36: Curva ROC del Clasificador *Gradient Boosting Tree* en el conjunto de datos balanceado por infra muestreo.

5.5 Resumen de las métricas del problema de clasificación binaria y tiempos de ejecución

En la (Tabla 18) se observa los resultados que toman las métricas utilizadas para medir el rendimiento de cada clasificador en los tres escenarios descritos de la base de datos para corregir el desbalanceo de clases.

	Métricas	*Naïve Bayes Gaussiano*	*Máquinas de Vector Soporte*	*Random Forest*	*Gradient Tree Boosting*
Clases desbalanceadas	Tasa de aciertos	0,7459	0,7811	0,8267	0,8303
	Sensibilidad	0,8493	0,7151	0,8384	0,8164
	Especificidad	0,7256	0,7942	0,8244	0,8331
	Precisión	0,3790	0,4065	0,4849	0,4909
	F1	0,5241	0,5184	0,6145	0,6132
	AUC	0,7776	0,7794	0,9014	0,9095
Clases balanceadas por sobre muestreo	Tasa de aciertos	0,7850	0,8142	0,9711	0,8814
	Sensibilidad	0,8444	0,8282	0,9724	0,9184
	Especificidad	0,7256	0,8001	0,9697	0,8444
	Precisión	0,7547	0,8056	0,9698	0,8551
	F1	0,7970	0,8167	0,9711	0,8856
	AUC	0,7768	0,8837	0,9933	0,9410
Clases balanceadas por infra muestreo	Tasa de aciertos	0,7616	0,8178	0,8452	0,8712
	Sensibilidad	0,7041	0,7397	0,8137	0,8110
	Especificidad	0,8192	0,8959	0,8767	0,9315
	Precisión	0,7957	0,8766	0,8684	0,9221
	F1	0,7471	0,8024	0,8402	0,8630
	AUC	0,7635	0,8636	0,9161	0,9363

Tabla 18: Métricas de los clasificadores en los distintos escenarios del conjunto de datos.

Se observa que cuando no se trata el desbalanceo de clases, el rendimiento de los clasificadores es menor. La Precisión en este caso es inferior a 0,5. Sin embargo, llama la atención que las métricas Sensibilidad y Especificidad son mayores al no tratar el desbalanceo de clases porque el clasificador funciona mejor al clasificar instancias como ‘Licitación no reparto’. Esto se

debe a que esta clase es la mayoritaria, de hecho, la probabilidad *a priori* de esta clase es de 0,835. También hay que tener en cuenta que se ha calculado las métricas en el punto óptimo de la curva AUC y no de la curva PR[20]. Por lo tanto, no se está maximizando la eficiencia de los clasificadores en estos términos.

Al tratar el desbalanceo de clases por sobre muestreo e infra muestreo, los resultados de las métricas mejoran excepto para el clasificador Naïve Bayes Gaussiano, que solamente mejora en la Precisión y consecuentemente en la métrica F1. *Random Forest* es el que más mejora, especialmente al sobre muestrear la base de datos. El F1 en este caso es igual a 0,9711 con una proporción de aciertos del 97,1%. *Gradient Tree Boosting* es el clasificador que mejores resultados ofrece cuando se infra muestrea la base de datos, con un F1 igual a 0,8630 y una tasa de aciertos del 87,1%.

6 Conclusiones y líneas futuras

Tras haber realizado todo el análisis se puede concluir los siguientes puntos:

1) No hay suficiente información como para lograr afirmar que una instancia clasificada como una licitación ilícita en este trabajo, sea realmente ilícita por ser objeto de reparto de un cártel o por cualquier otra conducta anticompetitiva a pesar de las evidencias económicas explicadas. Por lo tanto, no hay suficientes datos para evidenciar que una licitación clasificada como ilícita, sea realmente ilícita.
2) La base de datos que se ha diseñado no muestra una separabilidad clara entre clases. Si la CNMC o el organismo público correspondiente hubiera publicado información clara sobre los siguientes puntos, muy probablemente se habría conseguido obtener una base de datos suficientemente rica como para encontrar una separabilidad entre clases y cumplir el objetivo general de este TFM satisfactoriamente.
    a. Número de empresas que pujaron en la licitación,
    b. Tiempo que tardó una licitación en adjudicarse desde que se publicó,
    c. Cuantía de las pujas ofertadas

[20] *Precision-Recall*

d. Atributos cualitativos que indiquen el tipo de contrato que se realiza
e. Número total de empresas que ofrecen los bienes y servicios demandados por licitación

Las líneas futuras para aplicar dependerán de la disponibilidad de información. Si se consigue una mayor separabilidad entre clases, no solamente habrá evidencias económicas que la justifique, sino que también habrá justificaciones estadísticas suficientes como para que la autoridad competente pueda aprovecharse de materias como el Aprendizaje Automático o Aprendizaje Profundo para conseguir desmantelar conductas anticompetitivas en cualquier sector. Esto implicaría que no se tenga que desmantelar los cárteles a través del Programa de Clemencia y, por tanto, se evitaría tener que imponer una exención total o parcial sobre la sanción que habría que aplicar a una empresa que, a sabiendas, no cumple con la Ley *anti trust*.

7 Referencias

Cabral L.M.B. (2017) Introduction to Industrial Organization

Cerdá Martínez-Pujalte C. M. (2018) La manipulación fraudulenta de las licitaciones públicas (Bid rigging) *CNMC*

García-Verdugo J., Merino C. y Miren Martín A. (2019). Probability of cartel detection in Spain: an assessment

James G., Witten D.,Hastie T. y Tibshirani R. (2017). An Introduction to Statistical Learning

Marín Quemada J. M. et al, (2014). Resolución Expte. S/0453/12 Rodamientos ferroviarios *CNMC*

Lemaître G., Nogueira F., Aridas C. K. (2017). Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning

Marín Quemada J. M. et al, (2016). Resolución Expte. S/DC/0519/14 Infraestructuras ferroviarias *CNMC*

Marín Quemada J. M. et al, (2017). Resolución Expte. S/DC/0511/14 Renfe Operadora *CNMC*

Marín Quemada J. M. et al, (2017). Resolución Expte. S/DC/0511/14 NOKIA S/DC/0557/15 *CNMC*

Marín Quemada J. M. et al, (2019). Resolución Expte. S/DC/0598/16 Electrificación y electromecánica ferroviaria.*CNMC*

Mateo Vázquez J. D. (2017). Competición de Kaggle.com: Santander Costumer Satisfaction

McHugh, M.L. (2013). The Chi-square test of independence. Biochemia Medica, 23 (2), 143-149. https://doi.org/10.11613/BM.2013.018

Müller,A. y Guido, S. (2018). Introduction to Machine Learning with Python. O'Reilly Media Inc

Pedregosa et al. (2011) Scikit-learn: Machine Learning in Python

Sánchez Miqueleiz, F. (2021). Análisis de la competencia del sector de electrificación y electromecánicas ferroviarias.

Transparencia Internacional (2021). Corruption Perceptions Index

Villa Pedroza F. I. (2021). Métodos de aprendizaje automático para clasificación de riesgo de corrupción por adición en contratación de obra públicas mediante licitación. Caso de estudio en Colombia.

Anexo I: Construcción de la base de datos

Las fuentes de información y los pasos seguidos para el diseño de la base de datos son las siguientes:

1. Ministerio de Hacienda y Función Pública[21]: Se extraen todas las licitaciones disponibles hasta el 29 de marzo de 2023. Se filtran las licitaciones por la columna DIR 3 para obtener todas aquellas licitaciones en cuyo órgano de contratación sea:

    - ADIF, ya sea ADIF – Presidencia
    - ADIF Alta Velocidad - Consejo de Administración
    - ADIF Alta Velocidad - Presidencia
    - ADIF - Consejo de Administración
    - Administrador de Infraestructuras Ferroviarias (Dirección General Desarrollo de Infraestructura)
    - Dirección Comunicación – ADIF
    - Dirección General Explotación y Construcción – ADIF
    - Dirección General Financiera y Corporativa – ADIF
    - Dirección General RR.HH y Secretaría General y del Consejo – ADIF
    - Dirección General Servicios Clientes y Patrimonio – ADIF.

[21] https://www.hacienda.gob.es/es-ES/GobiernoAbierto/Datos%20Abiertos/Paginas/LicitacionesContratante.aspx

Se crea una nueva columna que concatene las columnas "CPV[22]" y "CPV licitaciones/lote" para clasificar de forma manual[23] cada una de las licitaciones para que el objeto de la licitación sea del mismo subsector que el objeto de la licitación de aquellas licitaciones en las que se ha demostrado colusión explícita. Se selecciona todas aquellas columnas que sean compatibles con las columnas disponibles en la resolución de los expedientes de la CNMC que se describen en los siguientes puntos.

2. RENFE. Resolución Expte. S/0453/12 Rodamientos ferroviarios (2014). No interesa porque no hay datos claros sobre las licitaciones. Se comparan licitaciones del año 2001 con otras del año 2004 y solamente se detalla el incremento entre precios por matrícula (las matrículas son rodamientos ferroviarios específicos). No hay datos disponibles que sea compatible con los predictores correspondientes.

3. ADIF. Resolución Expte, S/DC/0519/14 Infraestructuras ferroviarias (2016). Hay que concluir que Baja = (1 - Importe adjudicado / Importe presupuestado) *100. En base a esta fórmula y a las tablas que se muestra en el informe, es posible anexar los datos a la base de datos.

4. RENFE Operadora. Resolución Expte S/DC/0511/14 Renfe Operadora (2017) Es un caso de abuso de posición de dominio de RENFE Operadora. No hay licitaciones objeto de reparto.

5. ADIF. Resolución Expte. S/DC/0557/15 NOKIA (2017) Es un caso de abuso de posición de dominio de Nokia. Solamente se trata de una licitación y no tiene los suficientes predictores como para poder incluirlo en el análisis.

6. ADIF. Resolución Expte. S/DC/0614/17 Seguridad y Comunicaciones Ferroviarias (2021). Se extrae toda la información disponible respecto a aquellas licitaciones que son objeto de reparto para anexarlas a la base de datos. No hay información disponible sobre el número

[22] Vocabulario Común de Contratación Pública. En inglés *Common Procurement Vocabulary*

[23] Se clasificaron 10.180 licitaciones.

de participantes en las licitaciones. Teniendo en cuenta el número de observaciones que hay proporcionalmente respecto a la clase ‘Licitación objeto de reparto’, se decide eliminar el predictor ‘Número de participantes que pujaron en la licitación’, ya que proporcionalmente representarían un 95% de las instancias, teniendo que sacrificar del análisis predictor Mercado – Producto.

7. ADIF. Resolución Expte. Electrificación y electromecánica ferroviaria. S/DC/0598/16 (2019). Se extrae toda la información disponible respecto a aquellas licitaciones que son objeto de reparto para anexarlas a la base de datos.

Por lo tanto, el análisis se ha realizado solo sobre las licitaciones que ha publicado ADIF, ya que no hay suficiente información de las licitaciones objeto de reparto de RENFE como para poder hacer un análisis mínimamente coherente con los principios del Aprendizaje Automático.

Anexo II: Tiempo de ejecución de los ficheros notebook utilizados para la experimentación

El tiempo de ejecución[24] de los 12 ficheros notebook .ipynb que se elaborado para la sección 5. Experimentación ha sido el siguiente:

Algoritmo	*Fichero notebook*	*Tiempo*
Naïve Bayes Gaussiano	Clasificacion binaria gnb.ipynb	10 segundos
	Clasificacion binaria Sobre_Muestreo gnb.ipynb	20 segundos
	Clasificacion binaria Infra_Muestreo gnb.ipynb	6 segundos
Clasificador Máquinas de Vector Soporte	Clasificacion binaria svm.ipynb	3 minutos, 58 segundos
	Clasificacion binaria Sobre_Muestreo svc.ipynb	21 minutos, 58 segundos
	Clasificacion binaria Infra_Muestreo svm.ipynb	20 segundos
Random Forest	Clasificacion binaria rfc.ipynb	10 minutos, 8 segundos
	Clasificacion binaria Sobre_Muestreo rfc.ipynb	16 minutos, 48 segundos
	Clasificacion binaria Infra_Muestreo rfc.ipynb	2 minutos, 16 segundos
Gradient Tree Boosting	Clasificacion binaria gbt.ipynb	8 minutos, 32 segundos
	Clasificacion binaria Sobre_Muestreo gbt.ipynb	17 minutos, 44 segundos
	Clasificacion binaria Infra_Muestreo gbt.ipynb	1 minutos, 10 segundos

Tabla 19: Tiempo de ejecución de los ficheros. ipynb

[24] Procesador: 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz. RAM instalada: 16,0 GB (15,7 GB usable)