Predicción de errores en producción industrial de piezas mediante clasificación supervisada con desbalanceo de clases.

José Ahias Lopez Portillo
Master Universitario en Ciencia de Datos
Minería de datos y Machine Learning

Consultor: Jerónimo Hernández González
Profesor responsable de la asignatura: Jordi Casas Roma

Fecha Entrega: 9 de junio de 2019

Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada 3.0 España de Creative Commons

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Predicción de errores en producción industrial de piezas mediante clasificación supervisada con desbalanceo de clases*
Nombre del autor:	*José Ahias Lopez Portillo*
Nombre del consultor/a:	*Jerónimo Hernández González*
Nombre del PRA:	Jordi Casas Roma
Fecha de entrega (mm/aaaa):	06/2019
Titulación:	Master Universitario en Ciencia de Datos
Área del Trabajo Final:	Minería de datos y Machine Learning
Idioma del trabajo:	*Español*
Palabras clave	*Aprendizaje Automático, Clases desbalanceadas, Algoritmos clasificatorios*

Resumen del Trabajo (máximo 250 palabras):

El objetivo de este trabajo es lograr implementar un algoritmo de aprendizaje automático que permita la mejor clasificación sobre el conjunto de datos de información recolectado por diferentes sensores de la fábrica Bosch. El conjunto de datos constante de un archivo con 980 dimensiones y un millo de observaciones con una clasificación dicotómica. Al realizar diferentes investigaciones de soluciones para el procesamiento de conjuntos de datos desbalanceados, se implementaron 26 experimentos con 2 conjuntos de datos de diferente tamaño obteniendo el mejor resultado con técnicas de remuestreo y bosques aleatorios.

Abstract (in English, 250 words or less):

The objective of this work is to implement an automatic learning algorithm that allows the best classification on the set of information data collected by different sensors of the Bosch factory. The constant data set of a file with 980 dimensions and one million observations with a dichotomous classification. When carrying out different investigations of solutions for the processing of unbalanced data sets, 26 experiments with 2 sets of data of different sizes were implemented obtaining the best result with techniques of resampling and random forests.

Índice

Lista de figuras

Lista de tablas

1. Introducción

1.1 Contexto y justificación del Trabajo

Los modelos de producción sufren constantemente cambios asociados a la evolución y descubrimientos humanos, la primera revolución industrial marcó un antes y después en los métodos de fabricación. Más de 200 años han transcurrido y la única verdad es que los cambios continúan. Estamos sufriendo una cuarta revolución industrial que está basada en el conocimiento y el uso de datos. Muchos expertos la llaman transformación digital por su enfoque en el uso de tecnologías emergentes que logran potenciar los negocios y transformar los modelos productivos.

Uno de los hitos más importantes de la revolución industrial es la mejora continua en los procesos de fabricación. Hoy en día es común el uso de sensores en procesos industriales donde se producen una gran cantidad de bienes a través de procesos repetitivos donde se almacenan cada uno de los datos recolectados. Un fenómeno común en la mayoría de las empresas es que no procesan y no análisis los datos, el problema anterior es un reto empresarial para transformar esos datos en información que potencie y mejore continuamente la detección de errores en la producción. Las empresas invierten en diferentes tipos de tecnologías que generen alguna ventaja competitiva. La clasificación y predicción de eventos que pueden causar problemas en la cadena de producción son retos comunes que pueden ser resueltos con técnicas de aprendizaje automático. Dando origen a nuevos proyectos de ciencia de datos los cuales combinan diferentes técnicas de matemáticas e informáticas que permiten generar un valor adicional en los procesos industriales.

En empresas industriales uno de los principales retos es el control de calidad y la detección automática de piezas defectuosas. Este problema puede ser abordado de diferentes formas. Lo cual implica un reto para encontrar un

mecanismo óptimo y confiable. Los algoritmos de aprendizaje automático de clasificación son una de las soluciones que pueden utilizarse, siempre y cuando se aborde desde una perspectiva de clases altamente desbalanceadas.

Así, el propósito de este trabajo es explorar algoritmos de aprendizaje automático de clasificación para clases altamente desbalanceadas, para encontrar la mejor solución predictiva aplicada al conjunto de datos de una antigua competición de Kaggle (https://www.kaggle.com/c/bosch-production-line-performance/) donde la predicción es determinar si una pieza producida en línea no superara el control de calidad.

1.2 Objetivos del Trabajo

Objetivo General:

- Crear un modelo de clasificación para predecir cuándo una pieza producida en línea no superara el control de calidad mediante la selección del mejor algoritmo de aprendizaje automático de clasificación supervisada aplicada sobre el conjunto de datos de Kaggle con información de producción de la empresa Bosch.

Objetivos específicos:

- Aplicar un análisis exploratorio y reducción de dimensiones para generar un conjunto de datos adecuados para la implementación de los algoritmos de aprendizaje automático.
- Implementar algoritmos de clasificación supervisados especializados en clases desbalanceadas optimizados paramétricamente para generar el mejor modelo predictivo.
- Evaluar si un modelo que combina diferentes técnicas y algoritmos pueden mejorar la predicción de los modelos de clasificación supervisados individuales realizados en la fase anterior.

- Seleccionar el mejor modelo en base al análisis de las métricas de rendimiento de los algoritmos de clasificación supervisados.
- Generar las conclusiones del modelo a recomendar para la predicción de piezas producidas en línea que no superará el control de calidad para el conjunto de datos de entrenamiento y pruebas.

1.3 Enfoque y método seguido

Para la consecución de los objetivos, se aplica cada una de las técnicas de aprendizaje automático a los conjuntos de datos de entrenamiento con las variables numéricas y tiempo. Será necesario realizar un preprocesamiento y reducción de dimensiones para generar un conjunto de datos con las dimensiones más explicativas del modelo. Para ello se realizará un proceso de optimización de parámetros para cada uno de los algoritmos de clasificación supervisados para poder encontrar la mejor configuración que permita que los modelos puedan generar la mejor predicción siguiendo la siguiente estrategia:

a) Carga de datos y análisis exploratorio.
b) Aplicación de algún método de reducción de dimensiones.
c) Optimización individual de cada uno de los algoritmos de aprendizaje automático implementados sobre el conjunto de datos de entrenamiento reprocesado.
d) Validar los resultados de los algoritmos utilizados sobre el conjunto de datos de test reprocesado.
e) Crear un modelo de predicción combinando las diferentes salidas de los algoritmos predictivos generado en la primera fase.
f) Evaluar las métricas de los algoritmos de predicción para seleccionar el mejor modelo predictivo individual o combinado.
g) Generar las conclusiones y publicar la solución óptima para el conjunto de datos de entrenamiento y test utilizado para la clasificación de piezas defectuosas.

1.4 Planificación del Trabajo

Para el desarrollo del trabajo, utilizaremos las bases de datos de entrenamiento y pruebas que provee la competición de Kaggle que contiene información de producción de la empresa Bosch.

Las tareas planificadas para la realización del trabajo han sido las siguientes:

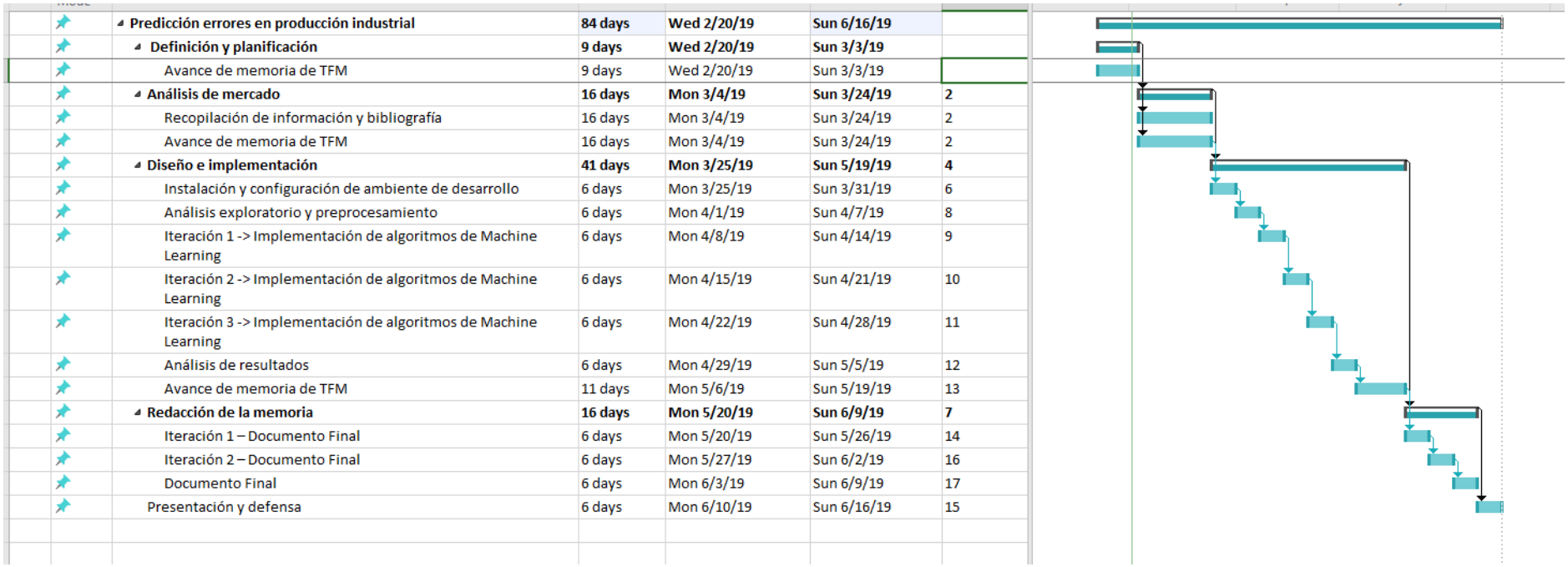
- **PEC1-Definición y planificación:** El objetivo principal de esta fase será establecer un plan inicial donde se realice una introducción a la problemática, objetivos y definición de metodología básica para el desarrollo del proyecto.

- **PEC2-Análisis de mercado:** El objetivo principal de esta fase es investigar publicaciones relacionadas a la problemática y mejorar el documento de memoria al agregar elementos científicos de investigación.

- **PEC3-Diseño e implementación:** El objetivo principal de esta fase es desarrollar el producto final que será un algoritmo de aprendizaje automático para la clasificación de piezas defectuosas, mediante 3 iteraciones de refinamiento.

- **PEC4-Redacción de la memoria:** El objetivo principal de esta fase es general el documento final del trabajo de fin de master.

- **PEC5-Presentación y defensa:** El objetivo principal de esta fase es general los recursos didácticos para realizar la defensa final del trabajo de fin de master**.**



Predicción errores en producción industrial	**84 days**	**Wed 2/20/19**	**Sun 6/16/19**	
Definición y planificación	**9 days**	**Wed 2/20/19**	**Sun 3/3/19**	
Avance de memoria de TFM	9 days	Wed 2/20/19	Sun 3/3/19	
Análisis de mercado	**16 days**	**Mon 3/4/19**	**Sun 3/24/19**	**2**
Recopilación de información y bibliografía	16 days	Mon 3/4/19	Sun 3/24/19	2
Avance de memoria de TFM	16 days	Mon 3/4/19	Sun 3/24/19	2
Diseño e implementación	**41 days**	**Mon 3/25/19**	**Sun 5/19/19**	**4**
Instalación y configuración de ambiente de desarrollo	6 days	Mon 3/25/19	Sun 3/31/19	6
Análisis exploratorio y preprocesamiento	6 days	Mon 4/1/19	Sun 4/7/19	8
Iteración 1 -> Implementación de algoritmos de Machine Learning	6 days	Mon 4/8/19	Sun 4/14/19	9
Iteración 2 -> Implementación de algoritmos de Machine Learning	6 days	Mon 4/15/19	Sun 4/21/19	10
Iteración 3 -> Implementación de algoritmos de Machine Learning	6 days	Mon 4/22/19	Sun 4/28/19	11
Análisis de resultados	6 days	Mon 4/29/19	Sun 5/5/19	12
Avance de memoria de TFM	11 days	Mon 5/6/19	Sun 5/19/19	13
Redacción de la memoria	**16 days**	**Mon 5/20/19**	**Sun 6/9/19**	**7**
Iteración 1 – Documento Final	6 days	Mon 5/20/19	Sun 5/26/19	14
Iteración 2 – Documento Final	6 days	Mon 5/27/19	Sun 6/2/19	16
Documento Final	6 days	Mon 6/3/19	Sun 6/9/19	17
Presentación y defensa	6 days	Mon 6/10/19	Sun 6/16/19	15

Figura 1. Diagrama de Gantt. Planificación de tareas

2. Fundamentos de Aprendizaje Automático

El objetivo de esta sección es proporcionar una introducción a conceptos de aprendizaje automático asociados a la resolución de problemas de clasificación.

2.1 Aprendizaje automático

Las empresas almacenan grandes cantidades de información relacionada a sus operaciones diarias. Esta información proviene de diferentes fuentes como sistemas relacionales, sensores y redes sociales. Estos datos se vuelven útil solo cuando logramos convertirlos en información que genera valor para la toma de decisiones, este proceso se describe de la siguiente forma: Datos→Información→ Conocimiento→Estrategias→Predicción.

Si una empresa de fabricación requiere mejorar sus procesos en la identificación de piezas defectuosas, podría utilizar diferentes técnicas de control de calidad, aunque el objetivo principal debería ser, crear un proceso automatizado que pueda predecir si una pieza es defectuosa. Esto es posible si utilizamos datos recopilados por sensores que se encuentren en la cadena de producción, esperando encontrar patrones y desviaciones que permitan crear un modelo matemático para la predicción.

El aprendizaje automático es un programa computacional que optimiza algún criterio utilizando datos de experiencias pasadas [1]. En este proceso se utilizarán técnicas estadísticas y algún modelo de algoritmo que genere una solución predictiva.

Los algoritmos de aprendizaje automático pueden ser agrupados dependiendo su naturaleza y aplicación, la clasificación que utilizaremos en esta investigación es la realizada por Kononenko [2] esta se puede observar en la Figura 2.1

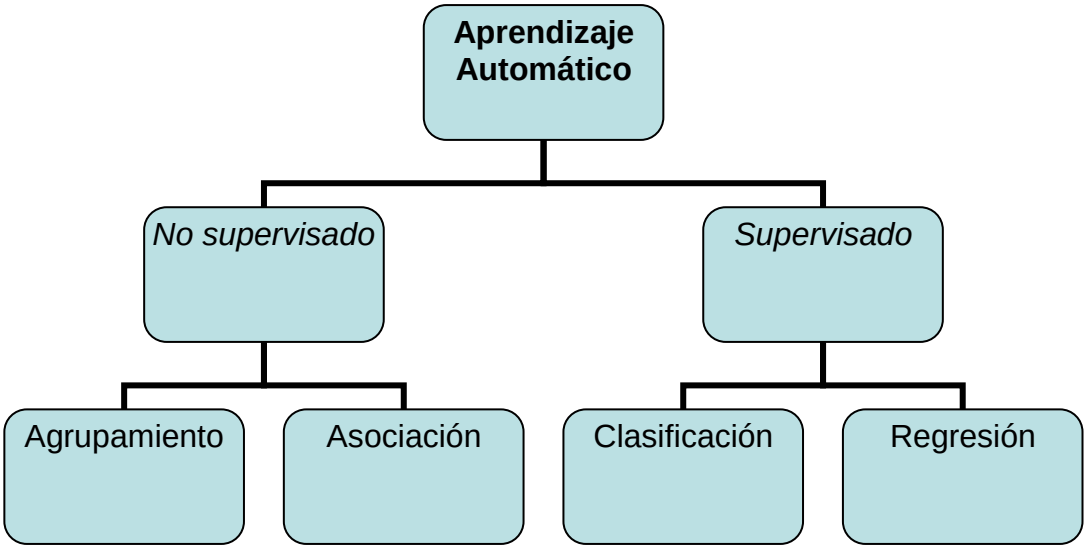


Figura 2.1: Clasificación de algoritmos de Aprendizaje automático

Aunque existe un gran número de algoritmos de aprendizaje automático las técnicas más utilizadas son las de clasificación. Para esta investigación nos centraremos en: Árboles de decisiones, Bosques aleatorios, Clasificadores Bayesianos, Regresión logística y Máquina de vectores de soporte que se muestran en la Figura 2.2.

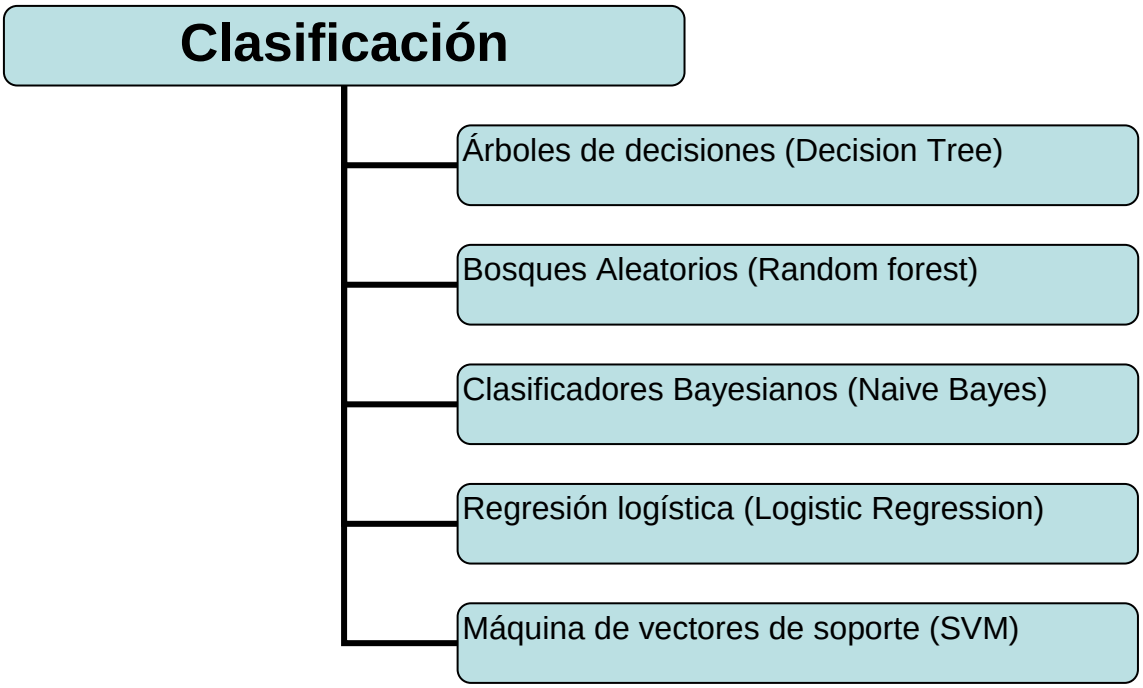


Figura 2.2: Algoritmos de Clasificación de Aprendizaje automático

2.1.1 Aprendizaje no supervisado

El aprendizaje no supervisado es sinónimo de agrupamiento. La principal característica de estos algoritmos es la utilización de datos de entrada que no están etiquetados [3]. El objetivo principal es encontrar patrones de agrupamiento que ocurren frecuentemente en los datos generando una clasificación que en muchas ocasiones son el punto de partida de un análisis exploratorio.

Uno de los algoritmos que más se utilizan en el aprendizaje no supervisado es el agrupamiento o clustering cuyo objetivo es agrupar los datos en conjuntos con atributos similares dentro del mismo grupo, mientras que las observaciones con atributos diferentes serán asociadas a otro grupo [4]. Esta función permite identificar anomalías o patrones que pueden ser útiles en la toma de decisiones.

Otro algoritmo de aprendizaje no supervisado es reglas de asociación donde el objetivo principal es poder identificar una implicación de la forma X→Y donde X es antecedente e Y es el consecuente de la regla [1]. Esta función permite identificar qué elementos están relacionados.

2.1.2 Aprendizaje supervisado

El aprendizaje supervisado es sinónimo de clasificación. La principal característica de estos algoritmos es la utilización de datos que se dividen en dos subconjuntos llamados entrenamiento y prueba, cada uno de los subconjuntos tendrán atributos de entrada y un dato de salida llamada clase [3]. La salida de cada uno de los algoritmos será un único valor continuo (regresión) o la predicción de una clase (clasificación).

2.1.2.1 Clasificación

La clasificación es uno de los problemas más demandados de las soluciones de aprendizaje automático estos problemas tienen como objetivo predecir una clase basada en sus atributos o propiedades. Los atributos son variables independientes que determinan la clase a que pertenece la observación.

Para determinar la clase de la observación el algoritmo necesita definir una función discreta que relaciona el espacio de los atributos con el espacio de la clase. Esta función es definida por un proceso de entrenamiento utilizando observaciones que son representativas con respecto al problema que se quiere solucionar mediante un modelo predictivo.

2.1.2.2 Regresión

La regresión no es un proceso exclusivo del aprendizaje automático, ramas como economía, estadística, ciencias sociales y otras disciplinas suelen utilizar modelos de regresión para poder explicar y predecir fenómenos.

En el aprendizaje automático la regresión parte de un conjunto de observaciones que son utilizadas para el entrenamiento. Las variables independientes o atributos explicativos, son las que determinaran una función predictiva para el cálculo de un valor continúo asociado a las características propias de observación analizada [4].

Las variables independientes son datos de entradas que se utilizaran como predictores en la función de regresión. Esta función puede ser definida previamente en el modelo o ser generada en el entrenamiento. El problema consiste en determinar o generar una función para predecir nuevos valores [2].

En la regresión lineal hay que tomar consideraciones con respecto a los datos:

- Los datos deben venir de una distribución normal.
- El método asume que existe una relación entre las variables dependiente e independiente.
- El algoritmo debe determinar la mejor función en base al menor error cuadrático medio.

2.2 Algoritmos de clasificación

2.2.1 Árboles de decisiones (Decision Tree)

Los árboles de decisiones, son métodos basados en la estrategia “divide y vencerás” [5][6]. Su estructura consta de nodos de decisiones que evalúan uno o más atributos, generando un efecto de programación similar al **“IF-THEN”** gráficamente podemos ver su funcionamiento en la Figura 2.3.

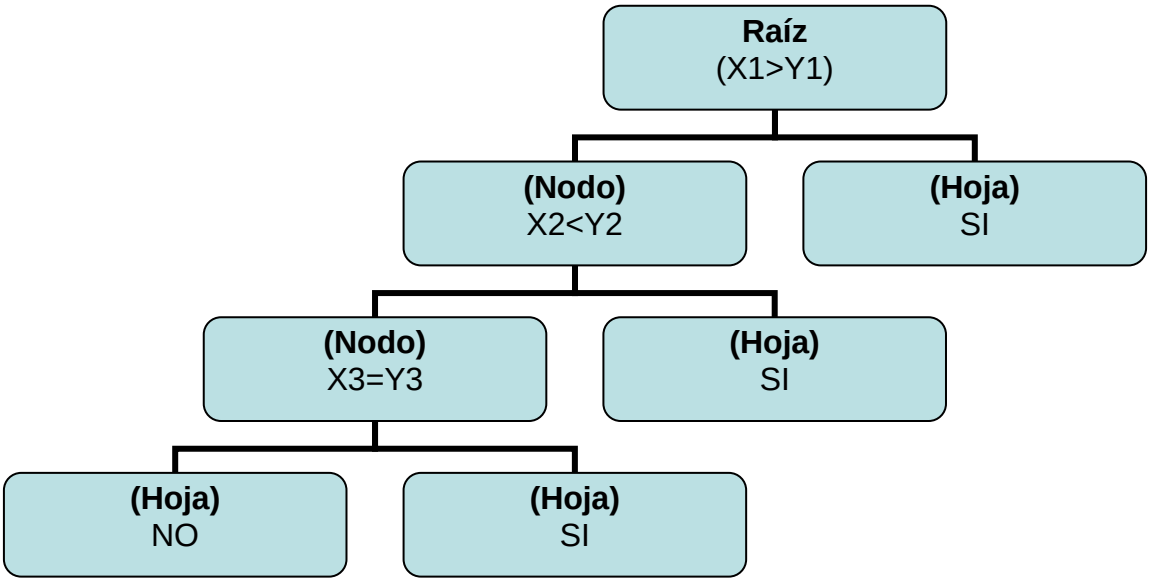


Figura 2.3: Ejemplo de Árbol de clasificación de dos clases

La implementación de árboles de decisiones implica dos etapas de funcionamiento: aprendizaje y clasificación. En la etapa de aprendizaje, se lleva a cabo la construcción del árbol que define las reglas de clasificación entre nodos de forma recursiva a partir de un conjunto de entrenamiento, en la

segunda etapa es clasificar utilizando una etiqueta. La clasificación inicia desde el nodo raíz, siguiendo el camino definido por la característica de la observación hasta llegar a la hoja que define la clase.

La construcción del árbol de decisiones constituye la fase mas compleja y es la que determina el resultado final.

2.2.2 Bosques Aleatorios (Random forest)

Los algoritmos de clasificación de bosques aleatorios (Random Forest) son clasificadores que utilizan arboles de decisiones como base [7]. Este algoritmo utiliza una técnica que consiste en crear diferentes modelos usando muestras aleatorias con reemplazo para luego combinar los resultados(Bagging). Por esto los Bosques aleatorios se ajustan perfectamente a los datos de entrenamiento que al promediar los resultados de cada uno de los árboles de decisiones ofrecen buenos resultados.

2.2.3 Clasificadores Bayesianos (Naive Bayes)

Los clasificadores bayesianos se basan en probabilidades condicionales del teorema de Bayes que asume el supuesto de independencia entre los predictores [8]. El clasificador minimiza el error esperando, asumiendo independencia de los atributos de la clase.

Un clasificador bayeasiano, está compuesto por nodos y arcos entre los nodos. Cada nodo corresponde a una variable o atributo llamado X que tiene una probabilidad P(X). Si existe un arco entre el nodo “X” y el nodo “Y” entonces esto indica una influencia condicional en términos de la probabilidad P(Y|X) [8].

La probabilidad de los nodos subsecuentes se puede calcular a partir del teorema de Bayes que especifica:

$$P(C|x) = \frac{P(C)p(x|C)}{P(x)}$$

Como método de clasificación, los clasificadores bayesianos se implementan como discriminadores entre clases, donde el espacio se divide en k regiones de decisiones donde las características están ligadas a las clases, considerando únicamente las observaciones que entran en la región de clasificación de la clase y rechazando aquellas que no [5].

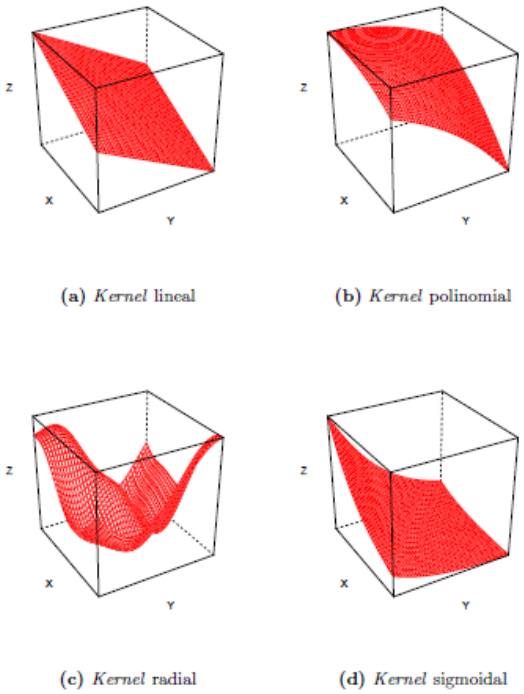
2.2.4 Regresión logística (Logistic Regression)

La regresión logística es un modelo estadístico que trata de explicar con que probabilidad ocurre un evento en base a atributos predictores [9]. El objetivo del modelo será crear una ecuación matemática la cual genera una salida binaria, la función de predicción viene dada por la siguiente expresión:

$$pi = \frac{1}{1 + e^{-(\beta 0 + \beta 1 X 1 + \cdots \beta z X z)}}$$

2.2.5 Máquinas de vectores de soporte (SVM)

La implementación de Maquinas de vectores de soporte en soluciones de ciencia de datos es una práctica común. Este algoritmo trata de dividir el espacio de características mediante hiperplanos de tal forma que estos generen una frontera entre clases, creando así tantos subespacios como etiquetas existan [10]. El proceso para determinar estas fronteras se basa en trasladar las instancias al espacio de características F y buscar el hiperplano que separe a estas. El cambio de espacio se realiza utilizando un kernel (polinomial, esférico, lineal) [11].



(a) *Kernel* lineal (b) *Kernel* polinomial

(c) *Kernel* radial (d) *Kernel* sigmoidal

Figura 2.4: Ejemplos de las distintas formas en el espacio transformado que podemos gestionar con las funciones kernel. Jordi Girones. Jordi Casas. Minería de datos modelos y algoritmos. UOC. 2017. p161

2.3 Evaluación de los modelos de clasificación

La implementación de un modelo de clasificación requiere una evaluación que determine numéricamente que tan bueno es el modelo. Las métricas más empleadas para esta tarea son: la tasa de errores y la tasa de aciertos. Un ejemplo para un problema de clasificación de dos clases utilizamos una matriz de confusión que se muestra en la Tabla 2.1.

Clase/ Predicción	**Positiva**	**Negativa**
Positiva	Verdadero Positivo (TP)	Falso Negativo (FN)
Negativa	Falso Positivo (FP)	Verdadero Negativo (TN)

Tabla 2.1: Matriz de confusión

La matriz de confusión no provee los siguientes datos:

- Verdadero Positivo (TP): número de clasificaciones correctas en la clase positiva.
- Verdadero negativo (TN) número de clasificaciones correctas en la clase negativa.
- Falso negativo (FN): número de clasificaciones incorrectas de clase positiva como clase negativa.
- Falso positivo (FP): número de clasificaciones incorrectas de la clase negativa en como clase positiva.

Utilizando la matriz de confusión podemos calcular las siguientes métricas:

- Error de clasificación (misclassification error, ERR)

$$ERR = \frac{FP + FN}{FP + FN + TP + TN}$$

- Exactitud (accuracy, ACC)

$$ACC = \frac{TP + TN}{FP + FN + TP + TN} = 1 - ERR$$

- Tasa de verdaderos positivos (True Positive Rate, TPR)

$$TPR = \frac{TP}{FN + TP}$$

- Tasa de verdaderos negativos (False Positive Rate, FPR)

$$FPR = \frac{FP}{FP + TN}$$

- Precisión (precision, PRE)

$$PRE = \frac{TP}{TP + FP}$$

- Recall (REC) y Sensibilidad (sensitivity, SEN)

$$REC = SEN = TPR = \frac{TP}{FN + TP}$$

- Especificidad (specificty, SPE)

$$SPE = \frac{TN}{TN + FP} = 1 - FPR$$

- F1

$$F1 = 2\ x\ \frac{PREC\ x\ REC}{PREC + REC}$$

2.4 Curvas ROC

La curva ROC es un gráfico bidimensional que permite visualizar, organizar y seleccionar clasificadores basados en su efectividad. La curva ROC mide el rendimiento respecto a los falsos positivos (FP) y verdaderos positivos (TP). La diagonal de la curva ROC se interpreta como un modelo generado aleatoriamente, mientras que valores inferiores se considera peores que una estimación aleatoria de un nuevo dato [12]. un ejemplo de curva ROC está ilustrado en la figura 2.5

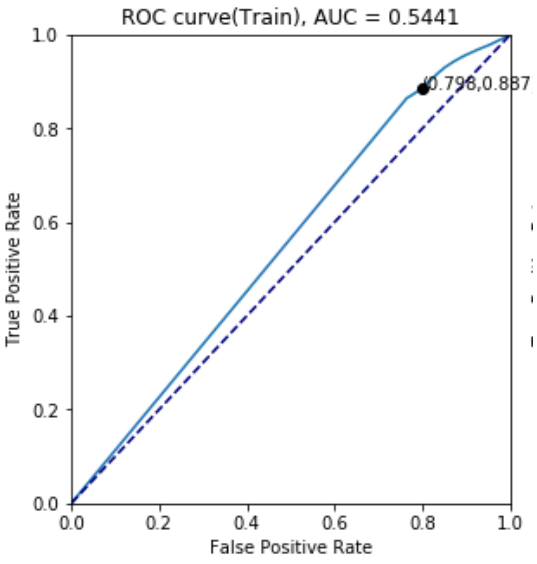


Figura 2.5: Ejemplo de curva ROC

3. Estado del Arte

3.1 Descripción del problema

El problema de distribuciones no balanceadas de datos entre clase está generando gran interés en disciplinas de Aprendizaje Automático y Minería de datos. El desbalanceo de clases es una característica que se genera cuando al menos una o más clases (clases minoritarias) se encuentra representada significativamente en una menor cantidad con respecto a las otras (clases mayoritarias). En este contexto, las clases minoritarias o mayoritarias suelen representar fenómenos que pueden ocurrir en diferentes niveles y no existe un umbral que nos indique cuando las clases de una muestra o una población está desbalanceada.

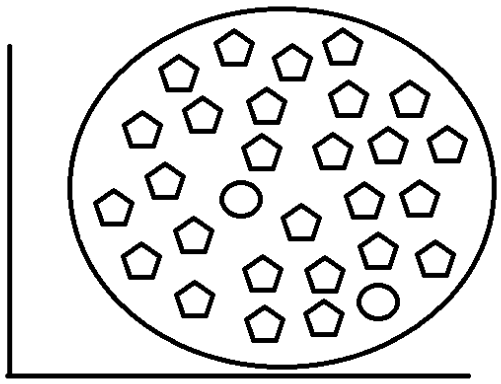


Figura 3.1: Ejemplo de una distribución desbalanceada

En la Figura 3.1 se muestra de una manera gráfica, el problema del desbalanceo de datos, nos encontramos con un número reducidos de círculos (clase minoritaria) en comparación con el pentágono (clase mayoritaria).

Algunos ejemplos de este tipo de caso son:

- Detección de fraude en llamadas telefónicas [13].
- Diagnósticos de fallos en equipos de telecomunicaciones [14]
- En finanzas [15].

- Prevención de intrusiones [16].

En desbalanceo de clases minoritarias suelen representar el mayor interés de clasificación esto es debido a que las clases minoritarias suelen ser valores atípicos asociados a un problema que se quiere solucionar.

Los escenarios de desbalanceo generan grandes problemas para los algoritmos de clasificación tradicionales los cuales son sensibles a generar valores sesgados a favor de la clase mayoritaria, lo que implica que la predicción de la clase minoritaria presenta bajo rendimiento en la clasificación. Esto es debido a que la mayoría de algoritmos de clasificación asumen una distribución balanceada en las clases. En la investigación realizadas existen diferentes literaturas de estudios realizados sobre algoritmos de clasificación para clases desbalanceadas, como las máquinas de vectores de soporte [17], vecino más cercano [18], árboles de decisiones [19], redes neuronales [20], entre otras investigaciones [21][22].

Las investigaciones anteriores concuerdan en:

- La mayoría de algoritmos de clasificación tiene por objetivo maximizar la tasa de aciertos.
- Los algoritmos de clasificación asumen que los datos de entrenamiento y prueba las clases se distribuyen uniformemente.
- Los algoritmos de clasificación asumen que los errores de clasificación tienen el mismo peso.

3.2 Origen del desbalanceo

Los investigadores concuerdan que es importante identificar el origen y los tipos de desbalanceo que existen entre clases y como afectan al conjunto de datos que los contienen.

El origen del desbalanceo puede ser producidos por:

a) Los datos están naturalmente desbalanceados.
b) Los datos no están naturalmente desbalanceados.

El literal “a”, se refiere a escenarios donde existen valores atípicos que no deberían existir en el conjunto de datos, un ejemplo: errores de fabricación, el objetivo de cualquier empresa industrial es poder generar una producción sin ningún error, la realidad es que las empresas tratan de minimizar esos errores de esta forma una pieza con error forma parte de un conjunto de datos minoritarios que debe analizarse.

Por otra parte, en el literal “b” son errores que pueden ser asociadas a la recolección de datos o una muestra no es representativa.

3.3 Métodos de tratamiento de clases desbalanceadas

3.3.1 Métodos de ajuste de la muestra

Existen diferentes métodos estadísticos para poder transforman un conjunto de datos desbalanceado a balanceado con el objetivo de poder implementar algoritmos de clasificación orientados a clases balanceadas, las técnicas que utilizaremos en esta investigación son:

- Sobremuestreo (oversampling)
- Submuestreo (undersampling)
- Sobremuestreo sintentetico (SMOTE)

3.3.1.1 Sobremuestreo (oversampling)

Consiste en obtener más observaciones de la clase minoritaria de la que originalmente existen, dejando intacta la cantidad de observaciones de la clase mayoritaria. Las observaciones adicionadas serán seleccionadas

aleatoriamente con reemplazo hasta lograr el efecto que ambas clases sean del mismo tamaño

Esta técnica permite generar un nuevo conjunto de datos sin realizar ninguna perdida de información, dado que el sobremuestreo simplemente agrega datos replicando, generando una desventaja que el conjunto de datos contiene información redundante y que en muchos casos los modelos de aprendizaje automático tienden a generar sobreajuste en sus predicciones [23].

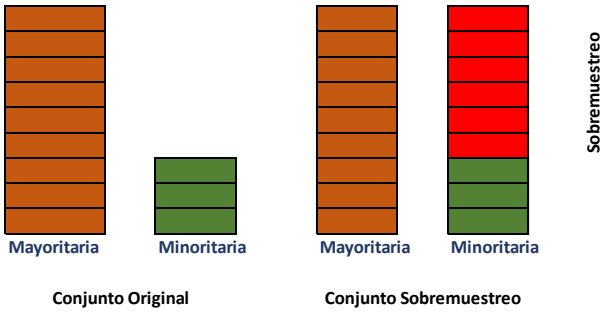


Figura 3.2: Ejemplo de Sobremuestreo

3.3.1.2 Submuestreo (oversampling)

Esta técnica permite generar un nuevo conjunto de datos mediante un muestreo aleatorio seleccionando una cantidad de observaciones limitadas de la clase mayoritaria hasta igualar las observaciones de la clase minoritaria, la ventaja de esta técnica es que reduce el numero de observaciones que ingresa al modelo de entrenamiento sin embargo el reducir observaciones se corre el riesgo de perder información pudiendo generar un sesgo en las predicciones [23].

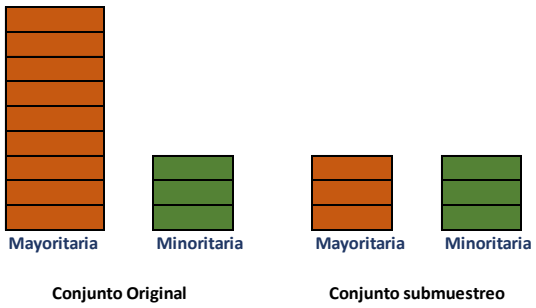


Figura 3.3: Ejemplo de Submuestreo

3.3.1.3 Sobremuestreo sintentetico (SMOTE)

La técnica de SMOTE es una propuesta para combatir las desventajas de las técnicas de sobremuestreo y submuestreo. SMOTE crea datos artificiales basados en la extrapolación de individuos que previamente existían, buscando para cada observación un individuo cercano para generar una nueva observación que es el resultado de unos o varios individuos existente [23].

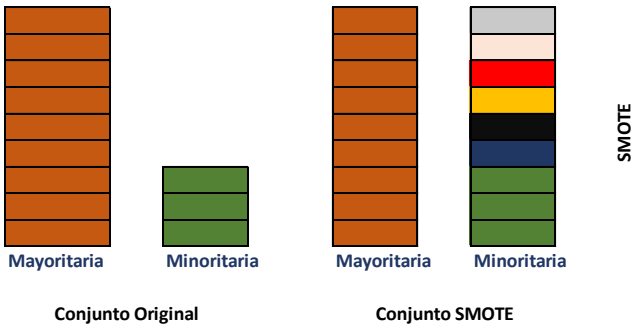


Figura 3.4: Ejemplo de SMOTE

3.3.2 Métodos Clasificación de una clase

Los algoritmos de clasificación de una clase (One-Class Classification OCC) pretenden construir modelos de clasificación cuando la clase negativa está ausente, existe un mal muestreo. Estos algoritmos son considerados para soluciones de detección de valores atípico [22]. Los clasificadores de una clase

(OCC) se entrenan para rechazar un objeto y etiquetarlo como salida falsa o valor atípico, siempre que no pueda asignarse el valor positivo.

3.3.2.1 SVM una clase de clasificación

Es una variante de SVM únicamente consiste en evaluar una de las clases con objetivo de generar la frontera de decisión [22].

4. Presentación del problemas y análisis descriptivo

En este apartado se expondrán los resultados obtenidos, así como la metodología utilizada en el análisis de ciencia de datos sobre el conjunto de datos de una antigua competición de Kaggle (https://www.kaggle.com/c/bosch-production-line-performance/) donde el objetivo es la predicción si una pieza producida en línea no superara el control de calidad.

4.1 Presentación del problema

En este proyecto se ha utilizado el conjunto de datos asociado al archivo ***"train_numeric.csv***", el cual contiene información de las lecturas de los diferentes sensores de la línea de producción de Bosch. Cada pieza está asociada a un identificador único con un resultado posible.

El conjunto de datos contiene una gran cantidad de observaciones con un alto grado de dimensiones, cada dimensión indica la línea de producción, la estación y el numero de la función. Ejemplo L3_S36_F3939 es una dimensión asociada a la línea de producción 3, estación 36 y es la característica 3939.

Al realizar un primer un primer análisis nos encontramos con un archivo que contiene más de un millón de observaciones y 970 dimensiones con la siguiente clasificación:

Clase	Valor
Clase Positiva (+)	0 (pieza sin defectos)
Clase Negativa (-)	1 (pieza con defectos)

Tabla 4.1: Matriz de confusión

La distribución de clase esta definida en la Figura 4.1.

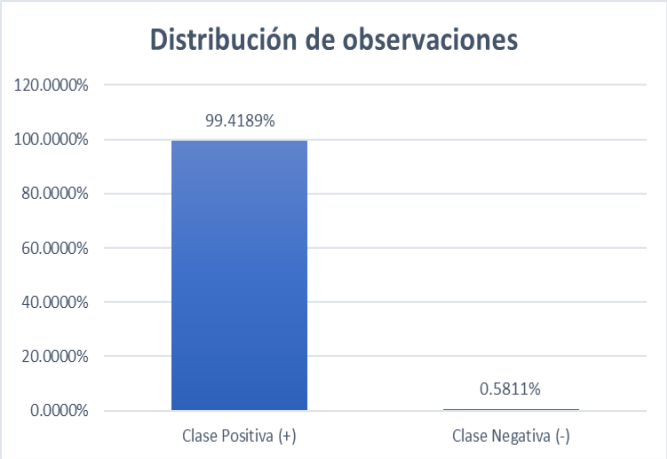


Figura 4.1: Distribución de observaciones por clases

Al analizar la Figura 4.1 observamos que nos encontramos con conjunto de datos altamente desbalanceado.

Es importante realizar un análisis exploratorio a nivel de observaciones y dimensiones. Iniciamos analizando las Figuras 4.2 y 4.3 que contiene los primeros y últimos 5 registros contenidos en el conjunto de datos.

	Id	L0_S0_F0	L0_S0_F2	L0_S0_F4	L0_S0_F6	L0_S0_F8	L0_S0_F10	L0_S0_F12	L0_S0_F14	L0_S0_F16	...	L3_S50_F42
0	4	0.030	-0.034	-0.197	-0.179	0.118	0.116	-0.015	-0.032	0.020	...	Na
1	6	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	Na
2	7	0.088	0.086	0.003	-0.052	0.161	0.025	-0.015	-0.072	-0.225	...	Na
3	9	-0.036	-0.064	0.294	0.330	0.074	0.161	0.022	0.128	-0.026	...	Na
4	11	-0.055	-0.086	0.294	0.330	0.118	0.025	0.030	0.168	-0.169	...	Na

5 rows × 970 columns

Figura 4.2: Visualización de primeras 5 filas

50_F4247	L3_S50_F4249	L3_S50_F4251	L3_S50_F4253	L3_S51_F4256	L3_S51_F4258	L3_S51_F4260	L3_S51_F4262	Response
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0

Figura 4.3: Visualización de ultimas 5 filas

Las Figuras 4.2 y 4.3 muestran que existe una gran cantidad de observaciones que en sus dimensiones contienen valores **“NaN”**, esto pueden ser errores de lecturas o datos válidos, para ellos vamos a realizar un análisis de como se distribuyen estos valores en cada una de las dimensiones es necesario validar la Figura 4.4.

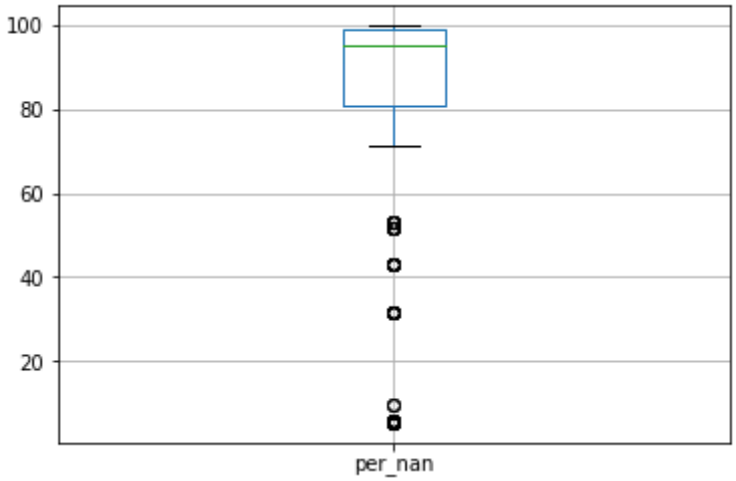


Figura 4.4: Grafico de cajas con la distribución de **“NaN”** en cada dimensión

Al generar un análisis de valores **“NaN”** versus valores numéricos observamos:

a) Que de las 968 dimensiones un bajo número poseen un alto porcentaje de valores numéricos, siendo consideradas como valores atípicos en la Figura 4.4.
b) Si analizamos la distribución, el primer cuartil (25%) de las dimensiones contendrán el 80% de las observaciones con valores **“NaN”**.

c) El 50% con un 95% de valores **“NaN”** y en casos extremos existen dimensiones donde 99.89% de sus observaciones registran valores **“NaN”**.

En la Figura 4.5 detallamos un análisis estadístico descriptivo de la distribución de los porcentajes de **“NaN”** (El porcentaje se calcula en base per_nan= Observaciones con NaN / Total de observaciones del conjunto de datos).

	per_nan
count	968.000000
mean	81.084969
std	30.681883
min	5.350000
25%	80.940000
50%	95.330000
75%	98.990000
max	99.890000

Figura 4.5: Distribución de % de valores **“NaN”** en dimensiones

Con este primer análisis nos encontramos con un conjunto de datos con serios problemas para ser utilizado en un análisis de ciencia de datos, al consultar la documentación proporcionada por la competición, no existe ningún detalle que nos ayude a comprender el proceso de generación de los datos.

Para continuar con este proyecto asumiremos que todos datos proporcionados en el conjunto de entrenamiento son significativos y que pueden utilizarse para la implementación de un modelo de aprendizaje automático.

El principal reto es, que no podemos descartar ninguna dimensión, porque no existe un patrón que nos indique cuales son las dimensiones mas significativas, porque al hacer una exploración aleatoria, observamos que una pieza puede registrar valores numéricos en diferentes dimensiones, lo que significa que

existe un alto número de combinaciones posibles, sumando el reto de un conjunto de datos altamente desbalanceado, asumiremos que debemos utilizar las 968 dimensiones.

4.1 preprocesamiento

Para poder iniciar con la implementación de los algoritmos de aprendizaje automático debemos solucionar el problema de los valores no definidos, existe un gran número de técnicas de imputación de datos.

Para el desarrollo del este proyecto utilizaremos la media para reemplazar estos valores no definidos, asumiendo que cualquier valor diferente a la media nos ayude a identificar atípicos. Estamos conscientes que el utilizar la media como valor de imputación alteramos la distribución de los datos y que puede generar un sesgo.

Una vez solucionamos el problema de los valores no definidos, debemos tomar la decisión si utilizar las 968 dimensiones para entrenar el modelo. Al tener un alto número de dimensiones.

Las mejores prácticas recomiendan utilizar alguna técnica de reducción como PCA. Al implementar PCA se recomienda un análisis de correlación, pero tenemos un alto número de dimensiones lo cual implica que la matriz correlación es enorme y difícil de interpretar. Para evitar este problema utilizaremos una matriz de colores que en base al valor de la correlación se pinta un punto dentro de la matriz, el objetivo al generar esta visualización es ver si los colores pintados están asociados a correlaciones altas o bajas, el resultado se muestra en la Figura 4.6.

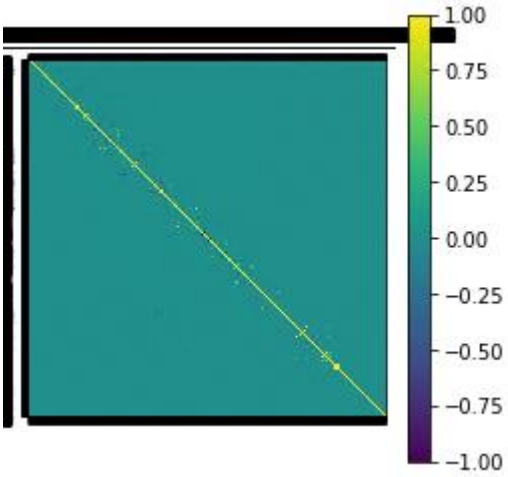


Figura 4.6: Matriz de correlación

Al analizar la Figura 4.6 observamos que los colores están asociado a correlaciones muy baja, lo cual nos indica que si implementamos PCA, la reducción de dimensiones podría ser no significativa, el objetivo principal al utilizar PCA es generar el menor números de dimensiones que explique el modelo en este caso corremos el riesgo que el número de dimensiones reducidas pueda ser cero, para validar esta problemática, vamos a generar un proceso para probar diferentes números de dimensiones del PCA el resultado de este proceso se puede validar en la figura 4.7.

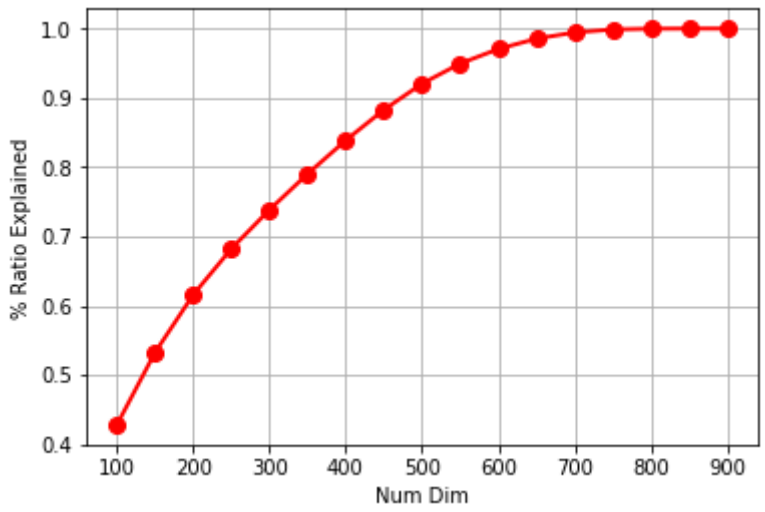


Figura 4.7: Grafico de experimentos con PCA

Dimensiones	Explicación	Dimensiones	Explicación
100	42.83%	**500**	92.04%
150	53.25%	**550**	94.95%
200	61.58%	**600**	97.05%
250	68.22%	**650**	98.54%
300	73.84%	**700**	99.53%
350	79.02%	**750**	99.86%
400	83.87%	**800**	99.97%
450	88.27%	**850**	99.99%

Tabla 4.2: Resultados de experimento PCA

Al analizar la Figura 4.6 y la Tabla 4.2, tenemos múltiples opciones del número óptimo de dimensiones que puede utilizar el PCA. Para este proyecto se utilizará 650 dimensiones para tener una explicación del 98.54% reduciendo el número de dimensiones en un 32%.

Una vez hemos ingresado los valores perdidos por medio de la media y finalizado el preprocesamiento del conjunto de datos con un PCA de 650 dimensiones, realizamos experimentos con todo el conjunto de datos. La sorpresa fue que la cantidad de datos y dimensiones utilizadas demandaban un alto tiempo de procesamiento y memoria con ANACONDA (Python 3.7), generando errores como falta de memoria, luego de muchas pruebas se reproceso diferentes conjuntos de datos más pequeños tomando la siguiente cantidad de observaciones para entrenamiento y pruebas ver tabla 4.3:

Experimento	Entrenamiento (Train)	Pruebas (Test)
1	169576	72677
2	251957	107982

Tabla 4.3: Distribución de observaciones para entrenamiento y pruebas.

5. Análisis predictivo

El objetivo del proyecto es implementar algoritmos de aprendizaje automático para la predicción de piezas defectuosa para ello se han realizado diferentes experimentos con algoritmos de clasificación.

En el problema de análisis sobre clasificación desbalanceada existen diferentes líneas de investigación y solución, en este proyecto se ha enfocado a:

- Transformar el conjunto de datos desbalanceado a balanceado con métodos de balanceos de clases como:
    - Sobremuestreo (Oversampling)
    - Submuestreo (Undersampling)
    - Sobremuestreo sintético (SMOTE).

- Algoritmos de clasificación para clases balanceadas como:
    - Arboles de decisiones (Decision Tree)
    - Bosques Aleatorios (Random forest)
    - Regresión logística binaria (Logistic Regression)
    - Clasificador Bayesiano (Naive Bayes)

- Algoritmos de clasificación orientados a una clase como:
    - Maquina de soporte vectorial de una clase (One-Support Vector Machine).

5.1 Algoritmos de clasificación de clases balanceadas

5.1.1 Sobremuestreo (oversampling)

Nomenclatura	Descripción
Odcexp1	Resultados del experimento 1 utilizando arboles de decisiones y sobremuestreo.
Odcexp2	Resultados del experimento 2 utilizando arboles de decisiones y sobremuestreo.
Orfexp1	Resultados del experimento 1 utilizando bosque aleatorios y sobremuestreo.
Orfexp2	Resultados del experimento 2 utilizando bosque aleatorios y sobremuestreo.
Olrexp1	Resultados del experimento 1 utilizando regresión lineal y sobremuestreo.
Olrexp2	Resultados del experimento 2 utilizando regresión lineal y sobremuestreo.
Obyexp1	Resultados del experimento 1 utilizando Bayes y sobremuestreo.
Obyexp2	Resultados del experimento 2 utilizando Bayes y sobremuestreo.

Tabla 5.1: Tabla de nomenclatura para experimentos de sobremuestreo.

Métricas	Odcexp1	Odcexp2	Orfexp1	Orfexp2	Olrexp1	Olrexp2	Obyexp1	Obyexp2
Accuracy	94.60%	96.21%	97.36%	**98.20%**	63.56%	63.94%	65.65%	77.25%
F-measure	14.30%	12.18%	19.69%	**18.26%**	7.80%	5.45%	6.56%	4.83%
Precision	96.91%	97.81%	99.88%	**99.91%**	63.83%	63.73%	66.32%	78.17%
False Positive Rate (FPR)	86.98%	89.08%	27.24%	**30.67%**	95.80%	97.13%	96.45%	97.38%
True Positive Rate (sensibility)	97.52%	98.31%	97.47%	**98.28%**	97.95%	98.63%	97.53%	98.29%
True Negative Rate (specificity)	15.84%	13.76%	11.39%	**10.51%**	54.31%	54.36%	42.44%	30.18%

Tabla 5.2: Resumen de métricas de experimentos con sobremuestreo.

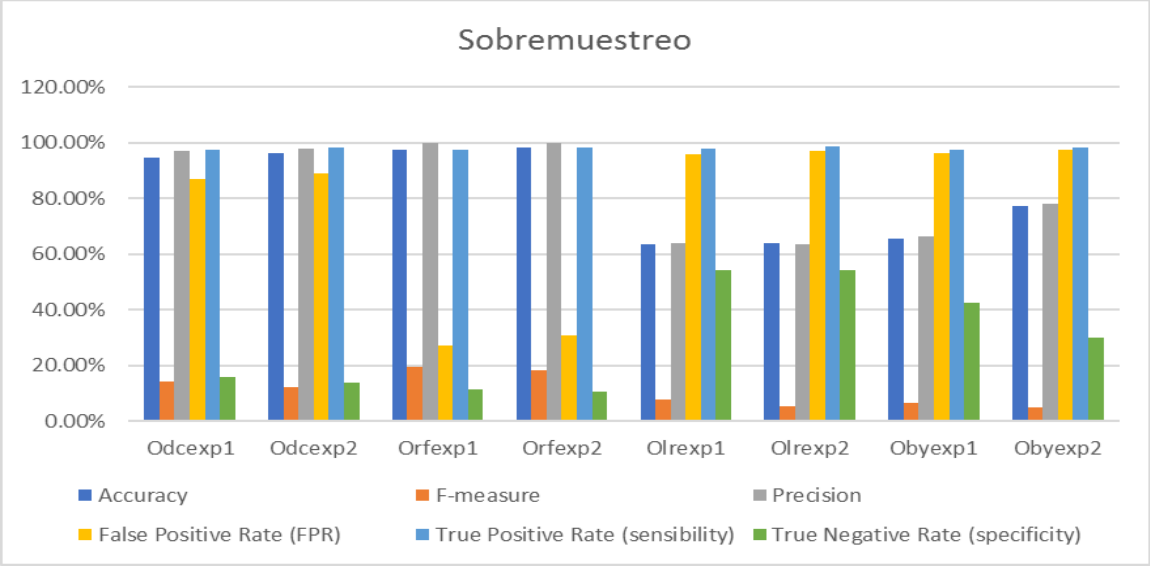


Figura 5.1: Resumen de métricas de experimentos con sobremuestreo.

5.1.2 Submuestreo (Undersampling)

Nomenclatura	Descripción
Udcexp1	Resultados del experimento 1 utilizando arboles de decisiones y submuestreo.
Udcexp2	Resultados del experimento 2 utilizando arboles de decisiones y submuestreo.
Urfexp1	Resultados del experimento 1 utilizando bosque aleatorios y submuestreo.
Urfexp2	Resultados del experimento 2 utilizando bosque aleatorios y submuestreo.
Ulrexp1	Resultados del experimento 1 utilizando regresión lineal y submuestreo.
Ulrexp2	Resultados del experimento 2 utilizando regresión lineal y submuestreo.
Ubyexp1	Resultados del experimento 1 utilizando Bayes y submuestreo.
Ubyexp2	Resultados del experimento 2 utilizando Bayes y submuestreo.

Tabla 5.3: Tabla de nomenclatura para experimentos de submuestreo.

Métricas	Udcexp1	Udcexp2	Urfexp1	Urfexp2	Ulrexp1	Ulrexp2	Ubyexp1	Ubyexp2
Accuracy	55.70%	55.32%	68.38%	70.45%	59.00%	60.78%	**85.30%**	84.99%
F-measure	6.74%	4.70%	9.08%	6.32%	7.35%	5.20%	**8.46%**	5.19%
Precision	55.68%	55.27%	68.76%	70.81%	59.05%	60.87%	**87.09%**	86.22%
False Positive Rate (FPR)	96.41%	97.55%	95.05%	96.64%	96.07%	97.28%	**94.86%**	97.05%
True Positive Rate (sensibility)	97.76%	98.53%	98.15%	98.70%	97.93%	98.62%	**97.51%**	98.26%
True Negative Rate (specificity)	56.40%	5.77%	55.62%	52.13%	57.27%	56.25%	**23.93%**	21.51%

Tabla 5.4: Resumen de métricas de experimentos con submuestreo.

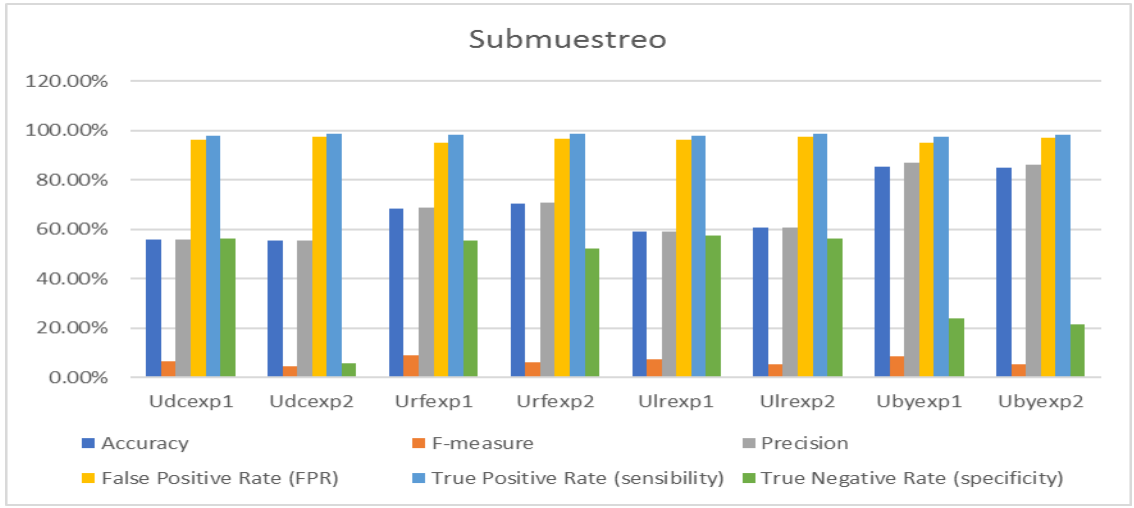


Figura 5.2: Resumen de métricas de experimentos con submuestreo.

5.1.3 Sobremuestreo sintético (SMOTE).

Nomenclatura	Descripción
Sdcexp1	Resultados del experimento 1 utilizando arboles de decisiones y SMOTE.
Sdcexp2	Resultados del experimento 2 utilizando arboles de decisiones y SMOTE.
Srfexp1	Resultados del experimento 1 utilizando bosque aleatorios y SMOTE.
Srfexp2	Resultados del experimento 2 utilizando bosque aleatorios y SMOTE.
Slrexp1	Resultados del experimento 1 utilizando regresión lineal y SMOTE.
Slrexp2	Resultados del experimento 2 utilizando regresión lineal y SMOTE.
Sbyexp1	Resultados del experimento 1 utilizando Bayes y SMOTE.
Sbyexp2	Resultados del experimento 2 utilizando Bayes y SMOTE.

Tabla 5.5: Tabla de nomenclatura para experimentos de SMOTE.

Métricas	Sdcexp1	Sdcexp2	Srfexp1	Srfexp2	Slrexp1	Slrexp2	Sbyexp1	Sbyexp2
Accuracy	85.04%	88.20%	97.20%	**98.11%**	63.51%	63.95%	21.78%	21.95%
F-measure	8.36%	6.69%	19.58%	**19.22%**	7.87%	5.36%	5.54%	3.78%
Precision	86.82%	89.48%	99.69%	**99.79%**	63.77%	64.16%	20.06%	20.82%
False Positive Rate (FPR)	94.94%	96.06%	47.12%	**47.74%**	95.76%	97.18%	97.13%	98.07%
True Positive Rate (sensibility)	97.51%	98.33%	97.49%	**98.31%**	97.97%	98.60%	97.32%	98.17%
True Negative Rate (specificity)	24.03%	22.14%	12.02%	**11.77%**	54.89%	53.39%	80.72%	80.14%

Tabla 5.6: Resumen de métricas de experimentos con SMOTE.

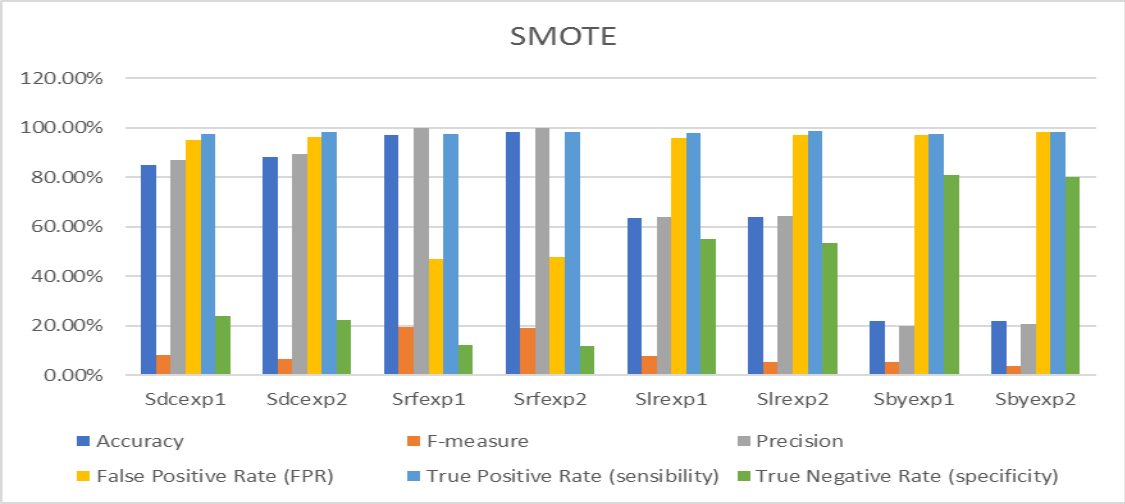


Figura 5.3: Resumen de métricas de experimentos con SMOTE.

5.2 Algoritmos de clasificación basados en una clase

Soporte de maquina vectorial de una clase (one-support vector machine)

Nomenclatura	Descripción
OSVMexp1	Resultados del experimento 1 utilizando arboles de decisiones y OSVM.
OSVMexp2	Resultados del experimento 2 utilizando arboles de decisiones y OSVM.

Tabla 5.7: Tabla de nomenclatura para experimentos de OSVM.

Métricas	OSVMexp1	OSVMexp2
Accuracy	96.90%	**97.70%**
F-measure	1.66%	**1.27%**
Precision	99.70%	**99.59%**
False Positive Rate (FPR)	91.67%	**96.42%**
True Positive Rate (sensibility)	97.18%	**98.10%**
True Negative Rate (specificity)	0.92%	**0.78%**

Tabla 5.8: Tabla de métricas para OSVM.

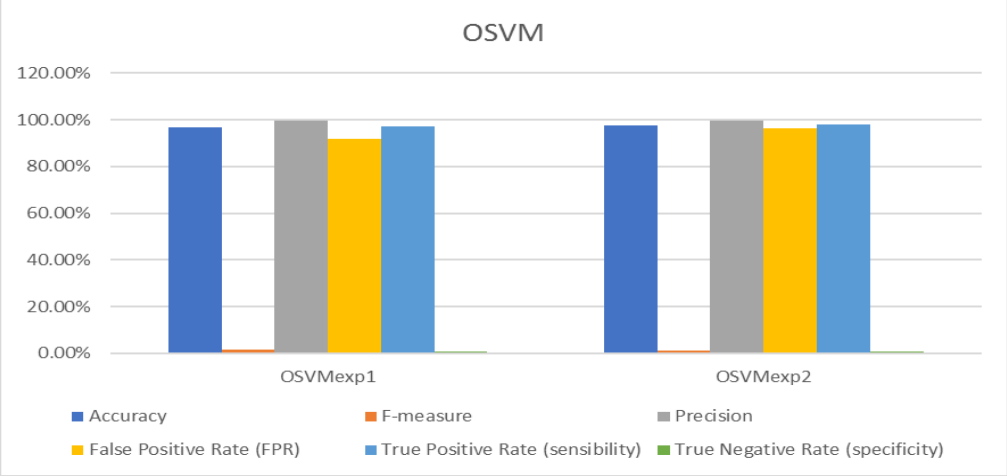


Figura 5.4: Resumen de métricas de experimentos con OSVM.

5.3 Resumen de experimentos

Métricas	Orfexp2	Ubyexp1	Srfexp2	OSVMexp2
Accuracy	**98.20%**	85.30%	**98.11%**	97.70%
F-measure	**18.26%**	8.46%	**19.22%**	1.27%
Precision	**99.91%**	87.09%	**99.79%**	99.59%
False Positive Rate (FPR)	**30.67%**	94.86%	**47.74%**	96.42%
True Positive Rate (sensibility)	**98.28%**	97.51%	**98.31%**	98.10%
True Negative Rate (specificity)	**10.51%**	23.93%	**11.77%**	0.78%

Tabla 5.9: Mejores resultados de experimentos 1 y 2

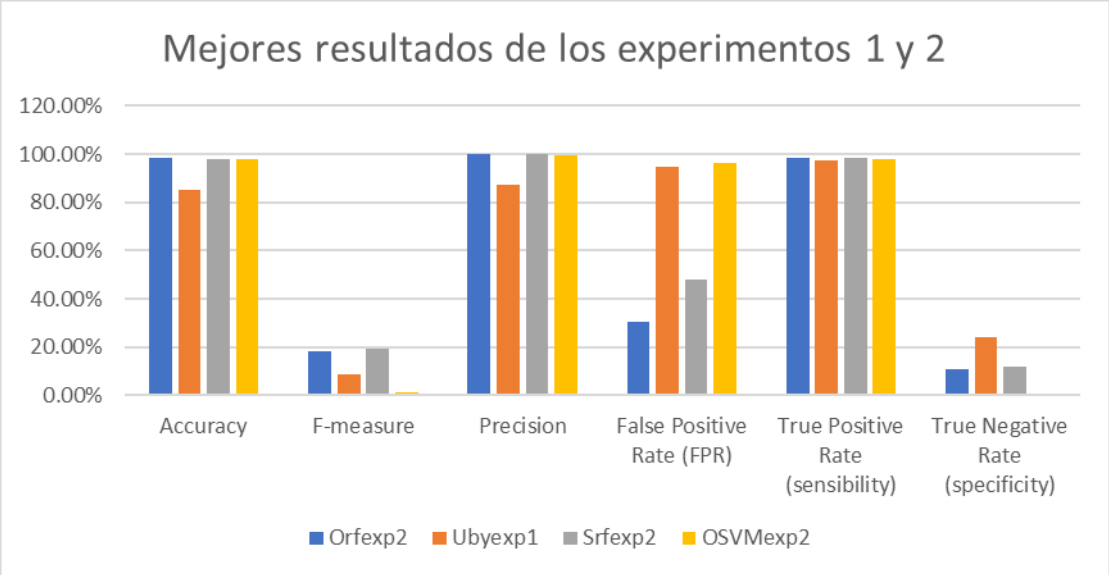


Figura 5.5: Mejores modelos de aprendizaje automático.

a) Se implementaron 26 modelos de aprendizaje automático utilizando diferentes técnicas de muestreo y algoritmos especializados para clases desbalanceadas.
b) Todas las implementaciones muestran una deficiencia en el cálculo de los verdaderos negativos lo que implica un gran número de Falsos positivos.

c) La peor técnica utilizada para el conjunto de datos es utilizar submuestreo, el descartar observaciones de la clase mayoritaria afecta la predicción de todos los modelos.
d) Aunque OSVM es un algoritmo especializado para escenarios de clases desbalanceadas no ha generado el mejor resultado en sus métricas.
e) En la tabla 5.9 se han seleccionado con verde los mejores 2 modelos que podrían utilizarse en un entorno productivo, estos modelos están relacionados al experimento 2 lo que implica que aumentar la muestra de entrenamiento ayuda a mejorar los resultados de los modelos.
f) El algoritmo que mejor resultado ofrece es Bosque aleatorios si se utiliza técnicas de sobremuestreo y sobremuestreo sintético (SMOTE)

6. Conclusiones y Trabajo futuro

6.1 Conclusiones

Existe una diferencia marcada entre el mundo académico y el mundo real empresarial, especialmente los proyectos de ciencia de datos que tienen como objetivo potenciar los datos hasta un nivel predictivo.

El presente proyecto sobre la clasificación con desbalanceo de clases, se convirtió en un reto profesional y personal, al ser un tema de mucho interés y, sobre todo ser un tema relacionado a una competición mundial con datos reales de una de las más grandes empresas “Bosch”.

Uno de los principales retos al inicio del proyecto fue la gran demanda de recursos que requiere analizar un conjunto de datos con un alto número de dimensiones y observaciones. La primera gran sorpresa fue que un único conjunto de datos utilice 11 gb de RAM solo para cargarlo en memoria y que los procesos preprocesamiento e implementación de modelos de aprendizaje automático llegaran utilizar más de 64gb de memoria RAM. El conseguir un recurso con poder de procesamiento adecuado implico un gran reto, pero gracias a la nube se logro adquirir un entorno para realizar los experimentos. El principal problema es que cada minuto significa dinero y que a medida se desarrollaba la solución la demanda de tiempo de procesamiento eran elevados, algunos experimentos implicaban 1 semana de procesamiento lo que aumentaba los costos por la utilización de la infraestructura en la nube, lo cual si se ve desde un punto de vista personal es costoso económicamente.

Una vez solventado el problema de infraestructura surgían preguntas y preguntas como resolver el problema del desbalanceo. El conjunto de datos implicaba 2 clases donde: La clase positiva está asociada al resultado de una pieza sin errores y la clase negativa está asociada a una pieza con errores. Al realizar un análisis de distribución de datos, las observaciones de la clase negativa o minoritaria solo significaban apenas un 0.58% del total

observaciones lo que implicaba que el conjunto de datos sufría un alto desbalanceo.

Una vez preprocesado el conjunto de datos, se desarrollaron 26 diferentes experimentos utilizando técnicas de remuestreo con algoritmos de clasificación orientados a clases balanceadas y algoritmos de clasificación orientados a una clase. De los 26 experimentos sobre el conjunto de datos utilizados, los mejores resultados se obtienen con: sobremuestreo, SMOTE y bosques aleatorios.

La mayor sorpresa en la investigación es comprobar que cada conjunto de datos es completamente diferente y que no existe una mejor solución específica. La motivación de la investigación, era comprobar que los algoritmos de clasificación orientados a una clase es la mejor solución, pero que la experimentación demostró todo lo contrario.

6.2 Trabajo futuro

En cuanto al trabajo futuro a desarrollar, como primer punto generar una serie de nuevos experimentos con más observaciones para el conjunto de entrenamiento, con el objetivo de mejorar la variación presentada entre los experimentos 1 y 2 donde la única diferencia es el tamaño de observaciones que tienen cada conjunto. Los experimentos deben estar enfocados en utilizar técnicas de sobremuestreo, SMOTE y bosques aleatorios tratando de realizar validaciones cruzadas y búsquedas de parámetros aleatorios para encontrar la mejor configuración para el modelo a implementar en un entorno de producción.

Como segundo punto es recomendable realizar una nueva serie de experimentos con otro algoritmo orientado a una clase como **“Isolation Forest”** o **“Redes neuronales”**

7. Bibliografía

[1] Ethem. Alpaydin. Introduction to Machine Learning, Second Edition. The MIT Press, 2010.

[2] Igor, Kononenko. Matjazkukar, Kukar. Machine Learning and Data Mining, Introduction to principles and Algorithms. Horwood Publishing, 2007.

[3] Han, Jiawei. Kamber, Micheline. Pei, Jian. Data Mining Concepts and Techniques, Third Edition, Morgan Kaufmann Publishing, 2012.

[4] Bishop, Christopher M. Pattern Recognition and Machine Learning, Springer Science+Business Media, LLC, 2006.

[5] Lopez, Arturo. Tesis: Algoritmos de balanceo de clases en problemas de clasificación binaria de conjuntos altamente desproporcionados. Instituto tecnológico y de estudios superiores de monterrey, 2008

[6] Dunham, Margaret. Data Mining: Introductory and Advanced Topics. Pearson Education. 2008

[7] Breiman, Leo. Random forest. University of California Berkeley, CA. 2001

[8] Torgo, Luis. Data Mining with R: Learning with case Studies. Chapman & Hall/ CRC. 2011.

[9] Cox, David Roxbee. The regression análisis of binary sequences. Journal of the Royal Statistical Society. Series B (Methodological) Vol. 20, No. 2. (1958). pp. 215-242

[10] Boser, B.E., Guyon, I.M., y Vapnik, V.N. (1992). A training algorithm for optimal margin classifiers. En Proceedings of the fifth annual workshop on Computational learning theory (pp. 144-152).

[11] Lopez, Ander. Tesis: Detección de sucesos raros con machine Learning. Universidad politécnica de Madrid. 2017

[12] Girones, Jordi. Casas, Jordi. Minguillon, Julian. Caihuelas, Ramon. Mineria de datos modelos y algoritmos. Editorial UOC.2017

[13] Fawcett, Tom. y Provost, Foster.: Adaptive Fraud Detection. Data Minining and Knowledge Discovery, 1997. pp 1-28

[14] Weiss, Gary M. Hrish, Haym. Learning to predict rare events in event sequences. Appears in Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining. 1998. pp 359-363

[15] Korn Sue.The Opportunity for Predictive Analytics in Finance. 2011

[16] De la Hoz Emiro. De la Hoz Eduardo., Ortiz Andrés. Ortega Julio. Modelo de detección de intrusiones en sistemas de red, realizando selección de características con FDR y entrenamiento y clasificación con SOM. 2012.

[17] Wu, Gang. Chang,Edward Y. Class-Boundary Alignment for Imbalanced Dataset Learning. Department of Electrical & Computer Engineering, University of California, Santa Barbara, 2003

[18] Hand, David J. Vinciotti, Veronica. Choosing k for two-class nearest neighbour classifiers with unbalanced classes. Pattern Recognition Letters, 2003, pp. 1555-1562.

[19] Philip, K. Chan. Salvatore, J. Stolfo. Toward Scalable Learning with Non-Uniform Class and Cost Distributions: A Case Study in Credit Card Fraud Detection. Knowledge Discovery and Data Mining. 1998,

[20] Perez, M.D. Rivera Antonio j. Carmona, C.J.. De Jesus, M.J. Training algorithms for Radial Basis Function Networks to tackle learning processes with imbalanced data-sets. Applied Soft Computing. 2014

[21] Arrieta, José. Mera, Carlos. Estudio Comparativo de Técnicas de Balanceo de Datos en el Aprendizaje de Múltiples Instancias. 2015

[22] Shehroz S. Khan. Michael G. Madden. One-class classification: taxonomy of study and review of techniques. 2014.

[23] Chakkrit, Tantithamthavorn. Kenichi,Matsumoto. The Impact of Class Rebalancing Techniques on the Performance and Interpretation of Defect Prediction Models. IEEE. 2018

8. Anexos

8.1 Métricas de Sobremuestreo (oversampling)

Arboles de decisiones (Decision Tree)

		Predicción		
		Clase + (0)	**Clase – (1)**	**Total**
Real	**Clase + (0)**	68429	2184	70613
	Clase – (1)	1737	327	2064
	Total	70166	2511	72677

Tabla 8.1: Matriz de confusión de árboles de decisiones (Experimento 1)

		Predicción		
		Clase + (0)	**Clase – (1)**	**Total**
Real	**Clase + (0)**	103602	2316	105918
	Clase – (1)	1780	284	2064
	Total	105382	284	107982

Tabla 8.2: Matriz de confusión de árboles de decisiones (Experimento 2)

Métricas	**Experimento 1**	**Experimento 2**
Accuracy	94.60%	96.21%
F-measure	14.30%	12.18%
Precision	96.91%	97.81%
False Positive Rate (FPR)	86.98%	89.08%
True Positive Rate (sensibility)	97.52%	98.31%
True Negative Rate (specificity)	15.84%	13.76%

Tabla 8.3: Tabla de métricas de evaluación de árboles de decisiones

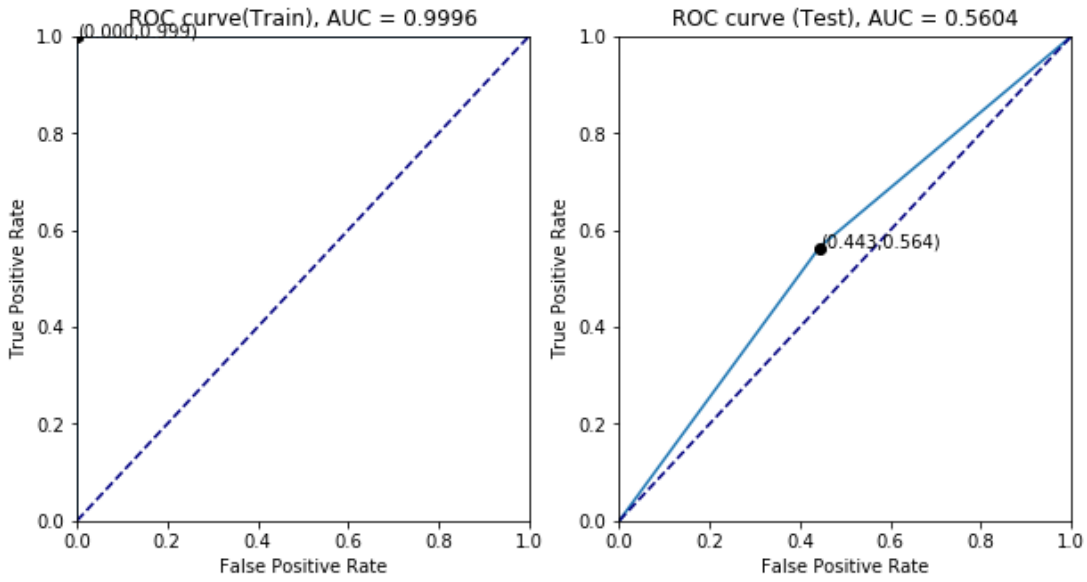


Figura 8.1: Curva ROC de árboles de decisiones para experimento 1

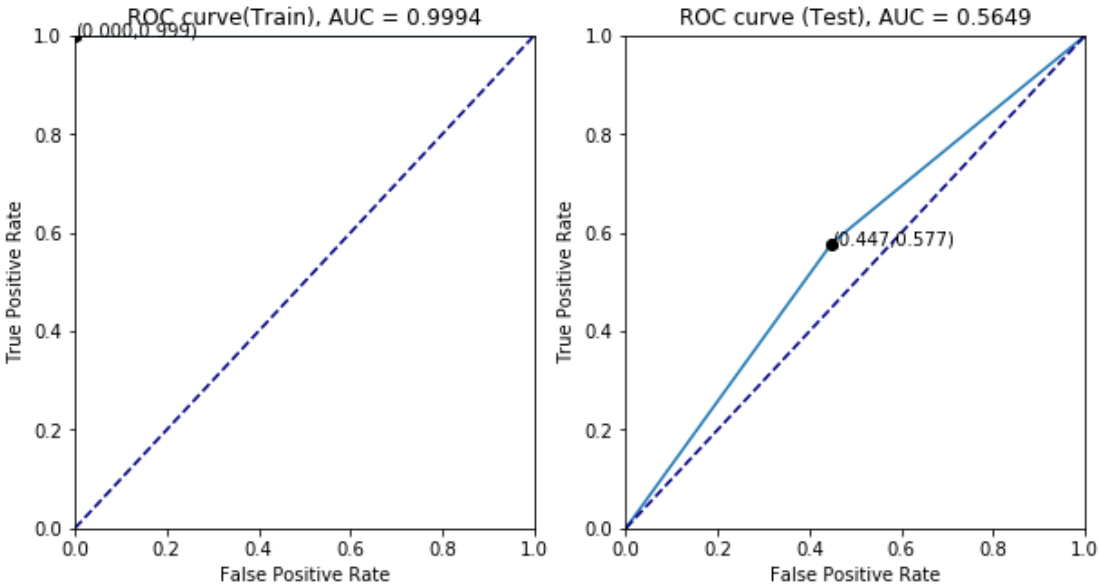


Figura 8.2: Curva ROC de árboles de decisiones para experimento 2

La ejecución del modelo de árboles de decisión ofrece un buen resultado para los experimentos 1 y 2, en cuanto a sus métricas “f-measure” y “specificity” presentan valores extremadamente bajos. Esto hace que el algoritmo presente un alto número de Falsos Negativos, un punto importante entre ambas ejecuciones es que su “accuracy” es mayor a 94%. Lo que significa que el número de errores en la predicción es bajo, tomando en cuenta que estamos

analizando un problema de fabricación es más barato generar una pieza mala que descartar 100 buenas.

Bosques aleatorios (Random forest)

		Predicción		
		Clase + (0)	**Clase – (1)**	**Total**
Real	**Clase + (0)**	70525	88	70613
	Clase – (1)	1829	235	2064
	Total	72356	323	72677

Tabla 8.4: Matriz de confusión de bosques aleatorios (Experimento 1)

		Predicción		
		Clase + (0)	**Clase – (1)**	**Total**
Real	**Clase + (0)**	105822	96	105918
	Clase – (1)	1847	217	2064
	Total	107669	313	107982

Tabla 8.5: Matriz de confusión de bosques aleatorios (Experimento 2)

Métricas	Experimento 1	Experimento 2
Accuracy	97.36%	98.20%
F-measure	19.69%	18.26%
Precision	99.88%	99.91%
False Positive Rate (FPR)	27.24%	30.67%
True Positive Rate (sensibility)	97.47%	98.28%
True Negative Rate (specificity)	11.39%	10.51%

Tabla 8.6: Tabla de métricas de evaluación de bosques aleatorios

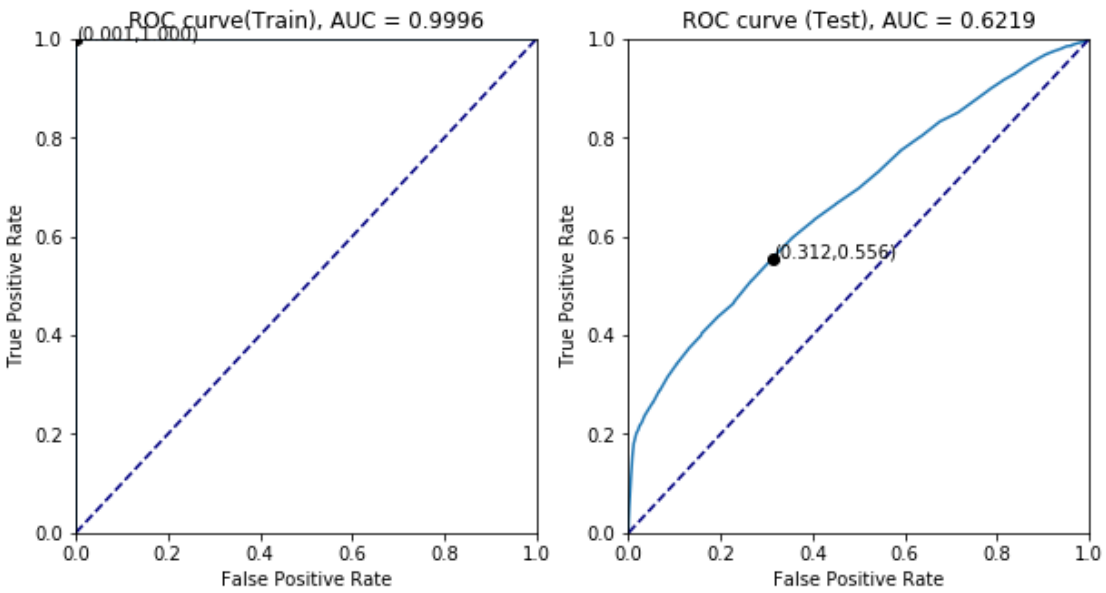


Figura 8.3: Curva ROC de bosques aleatorios para experimento 1

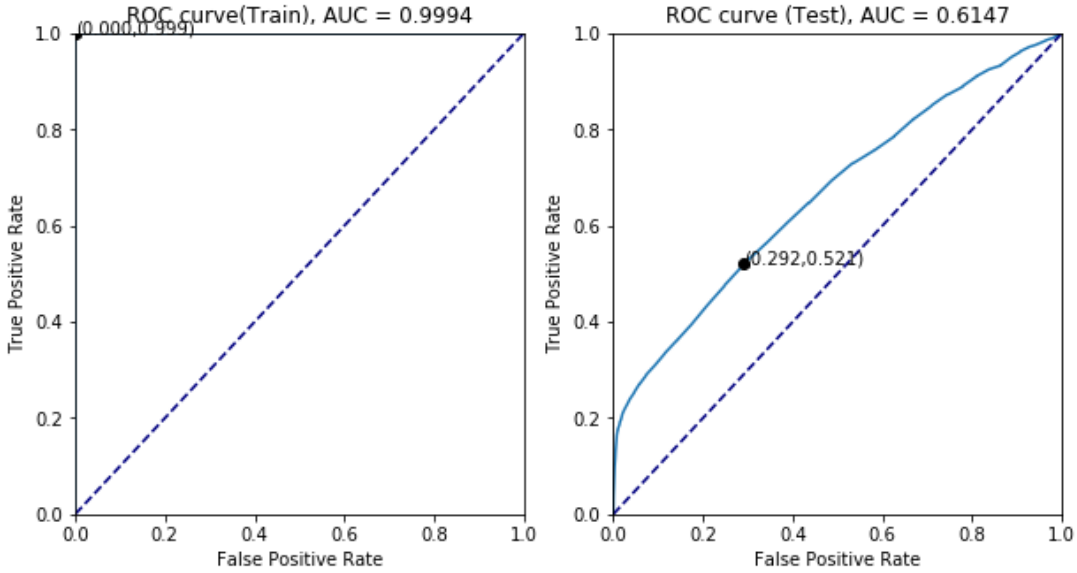


Figura 8.4: Curva ROC de bosques aleatorios para experimento 2

El algoritmo de bosques aleatorios ofrece mejores resultados para los experimentos 1 y 2, mejorando las métricas generada por los árboles de decisiones, aunque se observa que la tendencia en la clasificación de los falso

negativos continua notablemente, aunque “Accuracy” y “Precision” presenta una mejora significativa de más de 1% con respecto al algoritmo anterior.

Regresión logística (Logistic Regression)

		Predicción		
		Clase + (0)	**Clase – (1)**	**Total**
Real	**Clase + (0)**	45070	25543	70613
	Clase – (1)	943	1121	2064
	Total	46013	26664	72677

Tabla 8.7: Matriz de confusión de regresión logística (Experimento 1)

		Predicción		
		Clase + (0)	**Clase – (1)**	**Total**
Real	**Clase + (0)**	67924	37994	105918
	Clase – (1)	942	1122	2064
	Total	68866	39116	107982

Tabla 8.8: Matriz de confusión de regresión logística (Experimento 2)

Métricas	**Experimento 1**	**Experimento 2**
Accuracy	63.56%	63.94%
F-measure	7.80%	5.45%
Precision	63.83%	63.73%
False Positive Rate (FPR)	95.80%	97.13%
True Positive Rate (sensibility)	97.95%	98.63%
True Negative Rate (specificity)	54.31%	54.36%

Tabla 8.9: Tabla de métricas de evaluación de regresión logística

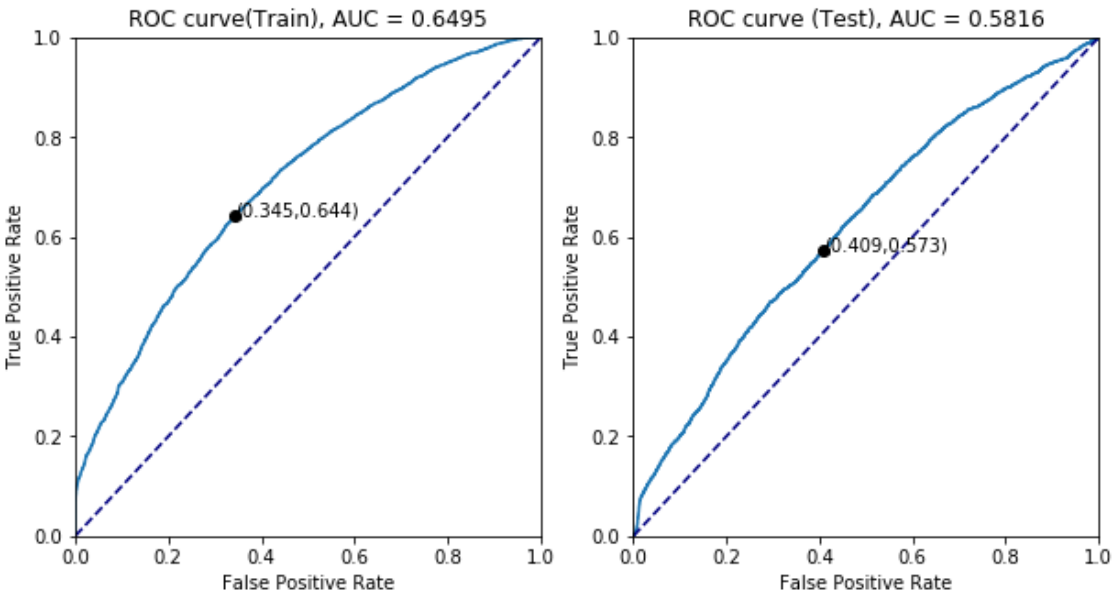


Figura 8.5: Curva ROC de regresión logística para experimento 1

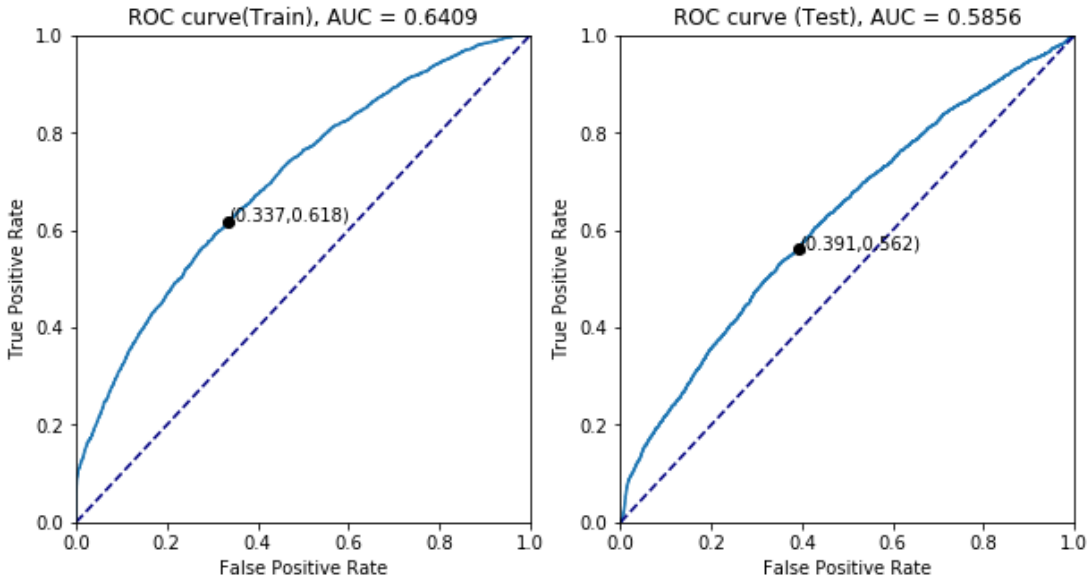


Figura 8.6: Curva ROC de regresión logística para experimento 2

Con un 63% de predicciones correctas el modelo de regresión logista ofrece un pésimo rendimiento para ser considerado un buen modelo, el utilizar sobremuestreo sobre el conjunto de datos de entrenamiento no genera un beneficio positivo con respecto a las predicciones correctas, aunque se observa

una mejora en la predicción de los verdaderos negativos lo que implica que es un método más sensible para predecir valores atípicos.

Clasificador Bayesiano (Naive Bayes)

		Predicción		
		Clase + (0)	**Clase – (1)**	**Total**
Real	**Clase + (0)**	46833	23780	70613
	Clase – (1)	1188	876	2064
	Total	48021	24656	72677

Tabla 8.10: Matriz de confusión de clasificador bayesiano (Experimento 1)

		Predicción		
		Clase + (0)	**Clase – (1)**	**Total**
Real	**Clase + (0)**	82791	23127	105918
	Clase – (1)	1441	623	2064
	Total	84232	23750	107982

Tabla 8.11: Matriz de confusión de clasificador bayesiano (Experimento 2)

Métricas	**Experimento 1**	**Experimento 2**
Accuracy	65.65%	77.25%
F-measure	6.56%	4.83%
Precision	66.32%	78.17%
False Positive Rate (FPR)	96.45%	97.38%
True Positive Rate (sensibility)	97.53%	98.29%
True Negative Rate (specificity)	42.44%	30.18%

Tabla 8.12: Tabla de métricas de evaluación de clasificador bayesiano

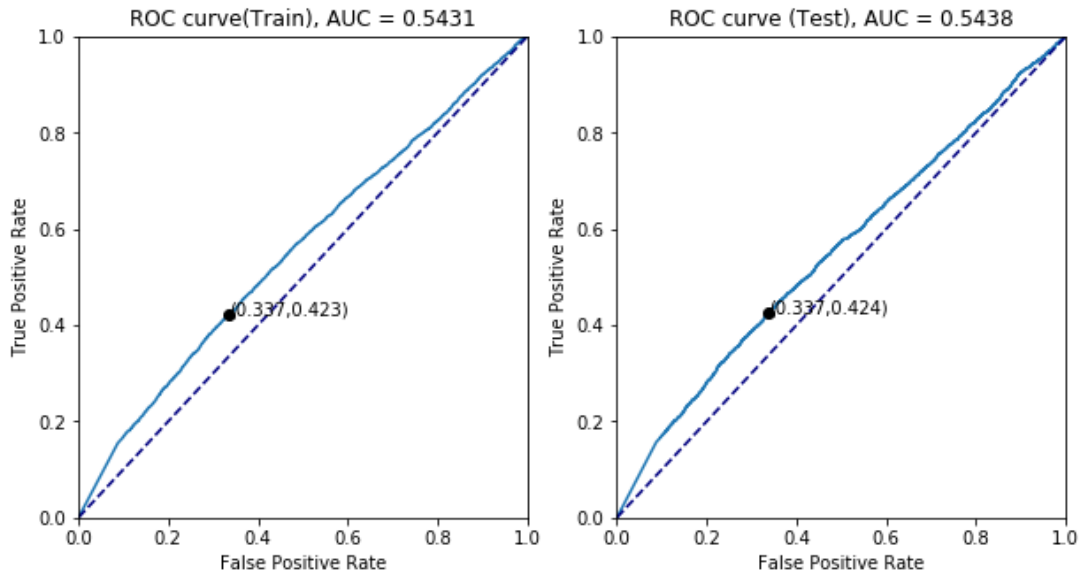


Figura 8.7: Curva ROC de clasificador bayesiano para experimento 1

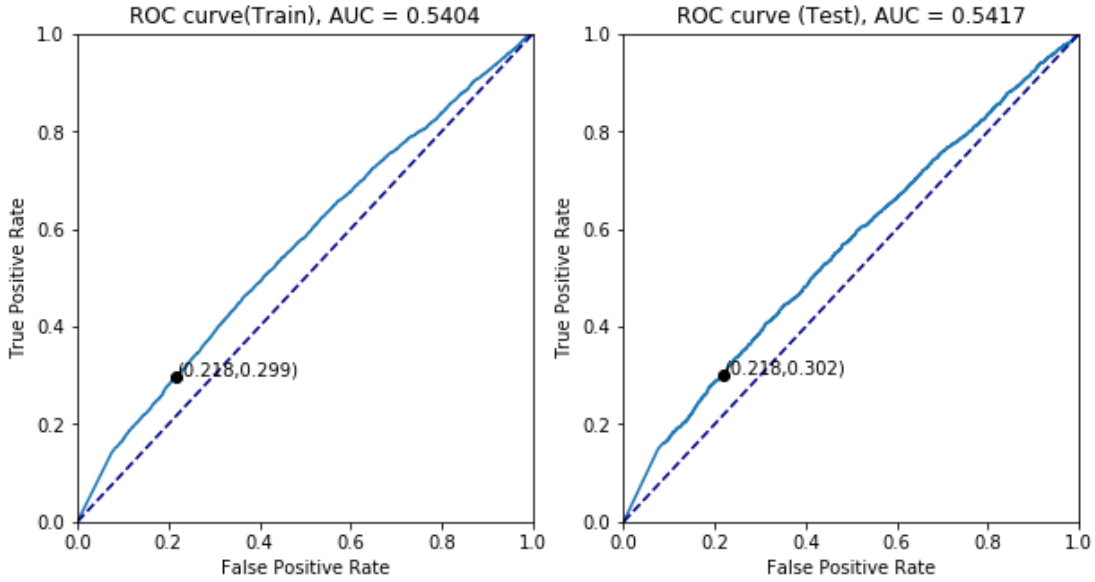


Figura 8.8: Curva ROC de clasificador bayesiano para experimento 2

El clasificador bayesiano ofrece mejores predicciones con respecto a los atípicos con un 77% de “Accuracy” el modelo ofrece un bajo rendimiento con respecto a bosques aleatorios.

8.2 Métricas de Submuestreo (Undersampling)

Árboles de decisiones (Decision Tree)

		Predicción		
		Clase + (0)	**Clase – (1)**	**Total**
Real	**Clase + (0)**	39317	31296	70613
	Clase – (1)	900	1164	2064
	Total	40217	32460	72677

Tabla 8.13: Matriz de confusión de árboles de decisiones (Experimento 1)

		Predicción		
		Clase + (0)	**Clase – (1)**	**Total**
Real	**Clase + (0)**	58540	47378	105918
	Clase – (1)	873	1191	2064
	Total	59413	48569	107982

Tabla 8.14: Matriz de confusión de árboles de decisiones (Experimento 2)

Métricas	**Experimento 1**	**Experimento 2**
Accuracy	55.70%	55.32%
F-measure	6.74%	4.70%
Precision	55.68%	55.27%
False Positive Rate (FPR)	96.41%	97.55%
True Positive Rate (sensibility)	97.76%	98.53%
True Negative Rate (specificity)	56.40%	5.77%

Tabla 8.15: Tabla de métricas de evaluación de árboles de decisiones

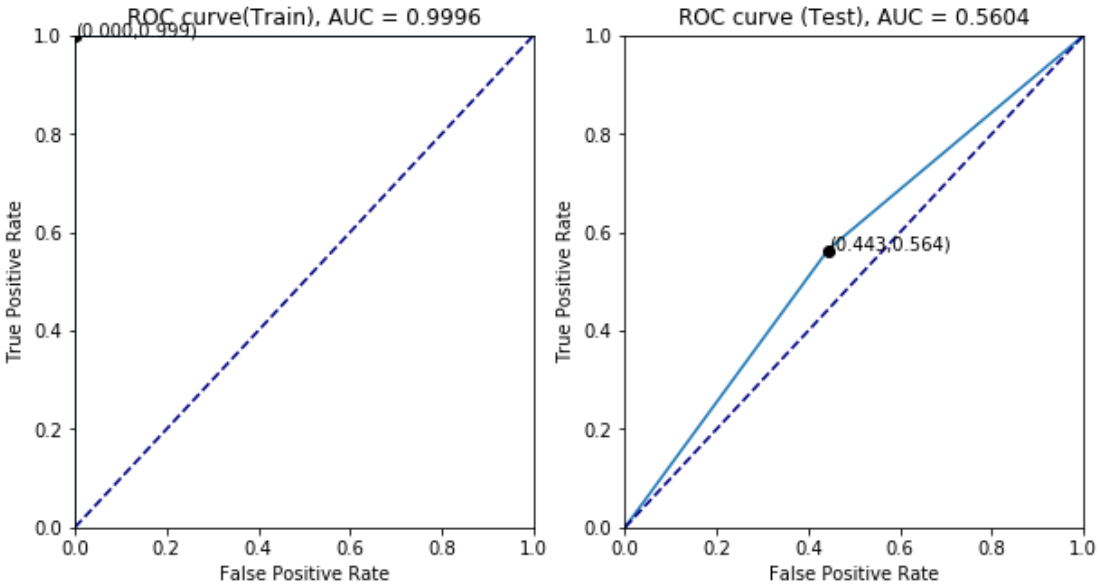


Figura 8.9: Curva ROC de árboles de decisiones para experimento 1

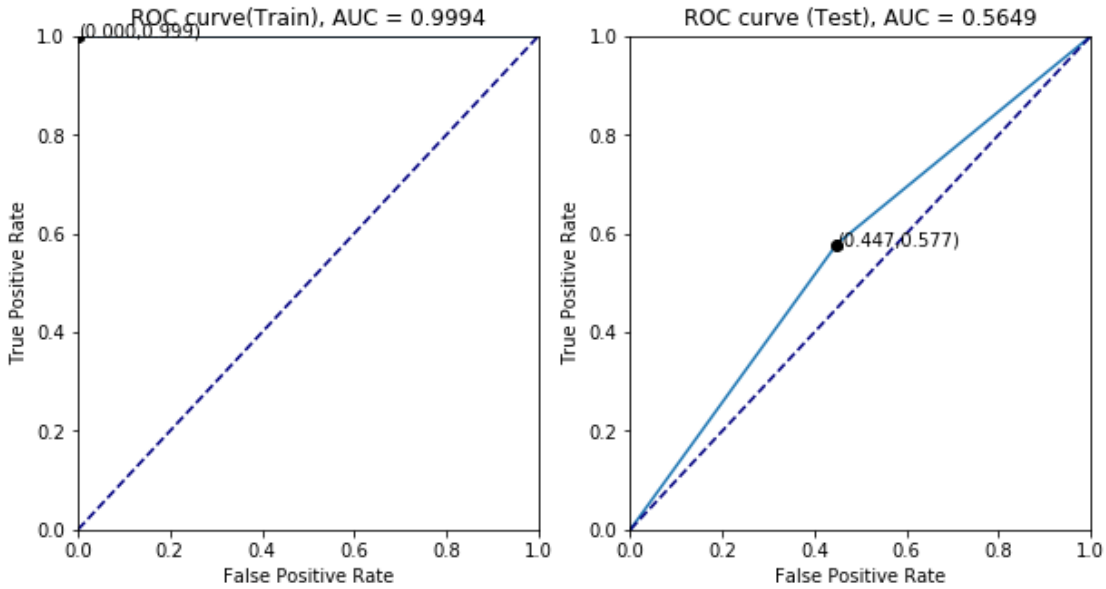


Figura 8.10: Curva ROC de árboles de decisiones para experimento 2

El utilizar la técnica de submuestreo está generando un efecto desastroso en el rendimiento de predicción, esto se puede observar en la métrica “Accuracy” que apenas es de 55% en los experimentos 1 y 2, esto nos indica que el tamaño de la muestra influye significativamente en el resultado del modelo generado.

Bosques aleatorios (Random forest)

		Predicción		
		Clase + (0)	**Clase – (1)**	**Total**
Real	**Clase + (0)**	48551	22062	70613
	Clase – (1)	916	1148	2064
	Total	49467	23210	72677

Tabla 8.16: Matriz de confusión de bosques aleatorios (Experimento 1)

		Predicción		
		Clase + (0)	**Clase – (1)**	**Total**
Real	**Clase + (0)**	75002	30916	105918
	Clase – (1)	988	1076	2064
	Total	75990	31992	107982

Tabla 8.17: Matriz de confusión de bosques aleatorios (Experimento 2)

Métricas	**Experimento 1**	**Experimento 2**
Accuracy	68.38%	70.45%
F-measure	9.08%	6.32%
Precision	68.76%	70.81%
False Positive Rate (FPR)	95.05%	96.64%
True Positive Rate (sensibility)	98.15%	98.70%
True Negative Rate (specificity)	55.62%	52.13%

Tabla 8.18: Tabla de métricas de evaluación de bosques aleatorios

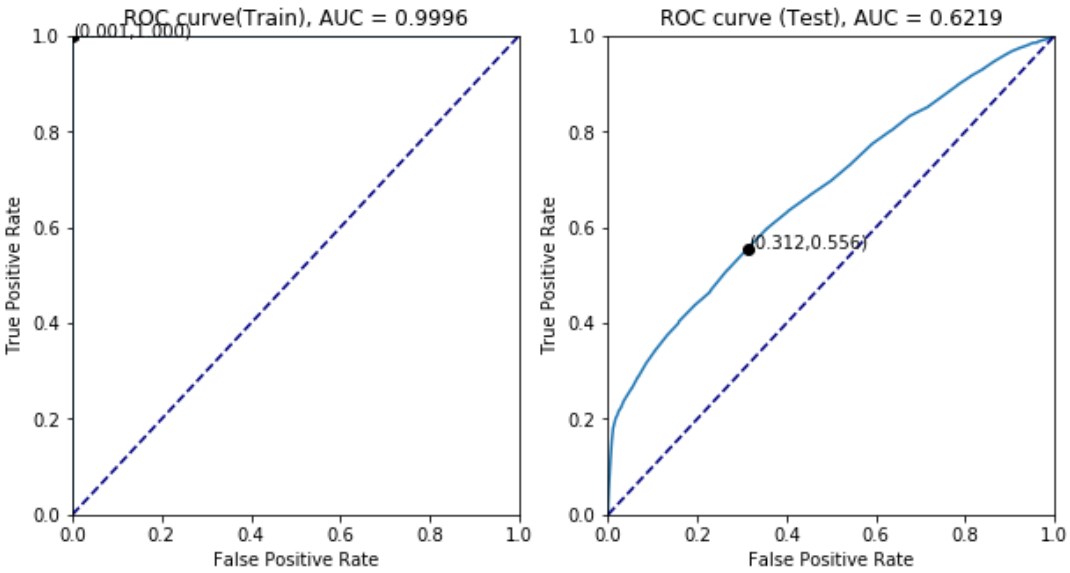


Figura 8.11: Curva ROC de bosques aleatorios para experimento 1

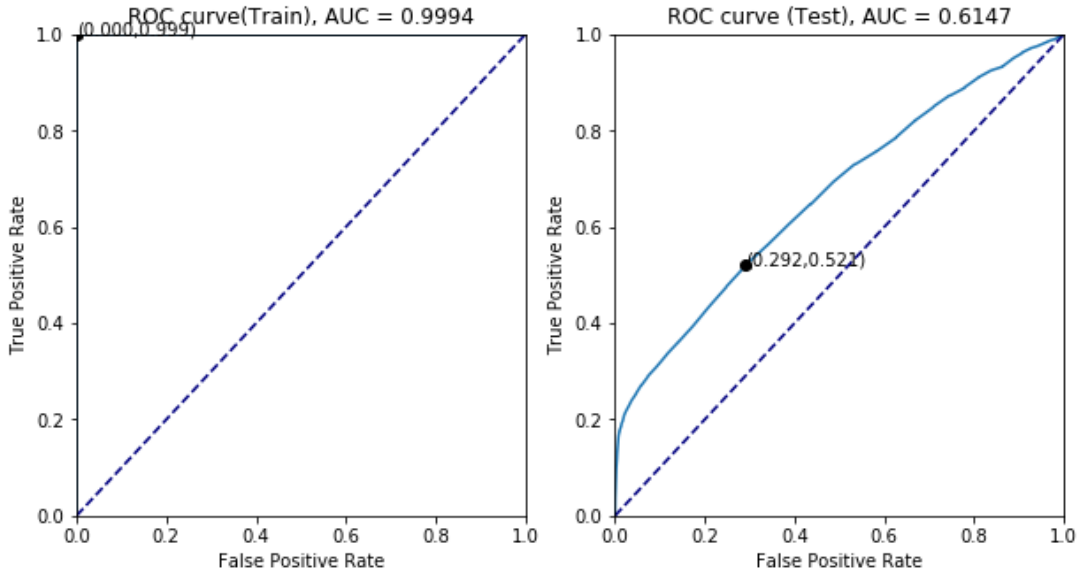


Figura 8.12: Curva ROC de bosques aleatorios para experimento 2

El disminuir la cantidad de observaciones de clase mayoritaria afecta significativa el resultado de las métricas de los modelos para los experimentos 1 y 2.

Regresión logística (Logistic Regression)

		Predicción		
		Clase + (0)	**Clase – (1)**	**Total**
Real	**Clase + (0)**	41697	28916	70613
	Clase – (1)	882	1182	2064
	Total	42579	30098	72677

Tabla 8.19: Matriz de confusión de regresión logística (Experimento 1)

		Predicción		
		Clase + (0)	**Clase – (1)**	**Total**
Real	**Clase + (0)**	64471	41447	105918
	Clase – (1)	903	1161	2064
	Total	65374	42608	107982

Tabla 8.20: Matriz de confusión de regresión logística (Experimento 2)

Métricas	**Experimento 1**	**Experimento 2**
Accuracy	59.00%	60.78%
F-measure	7.35%	5.20%
Precision	59.05%	60.87%
False Positive Rate (FPR)	96.07%	97.28%
True Positive Rate (sensibility)	97.93%	98.62%
True Negative Rate (specificity)	57.27%	56.25%

Tabla 8.21: Tabla de métricas de evaluación de regresión logística

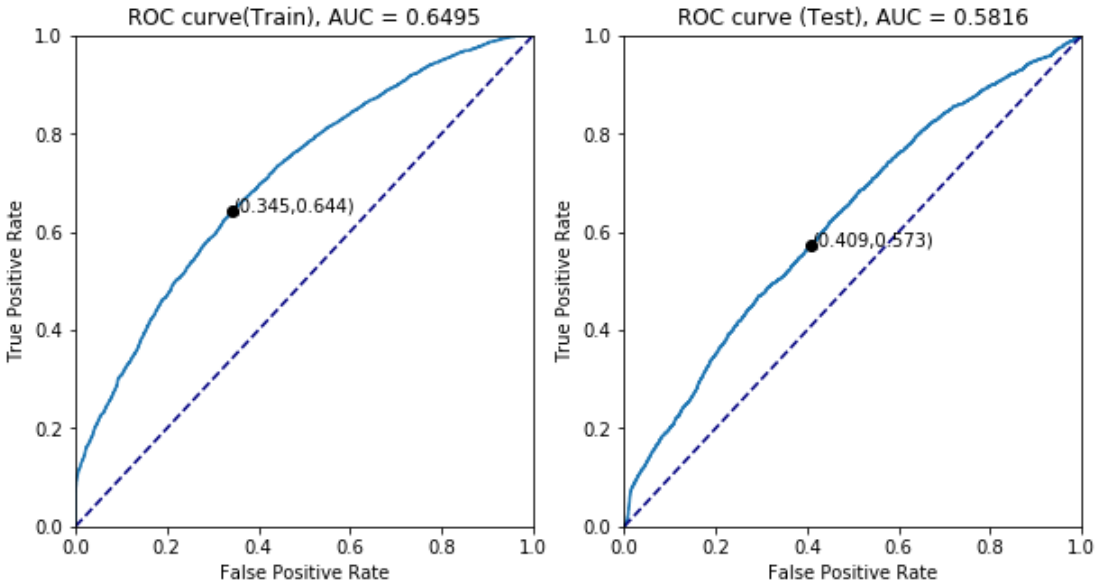


Figura 8.13: Curva ROC de regresión logística para experimento 1

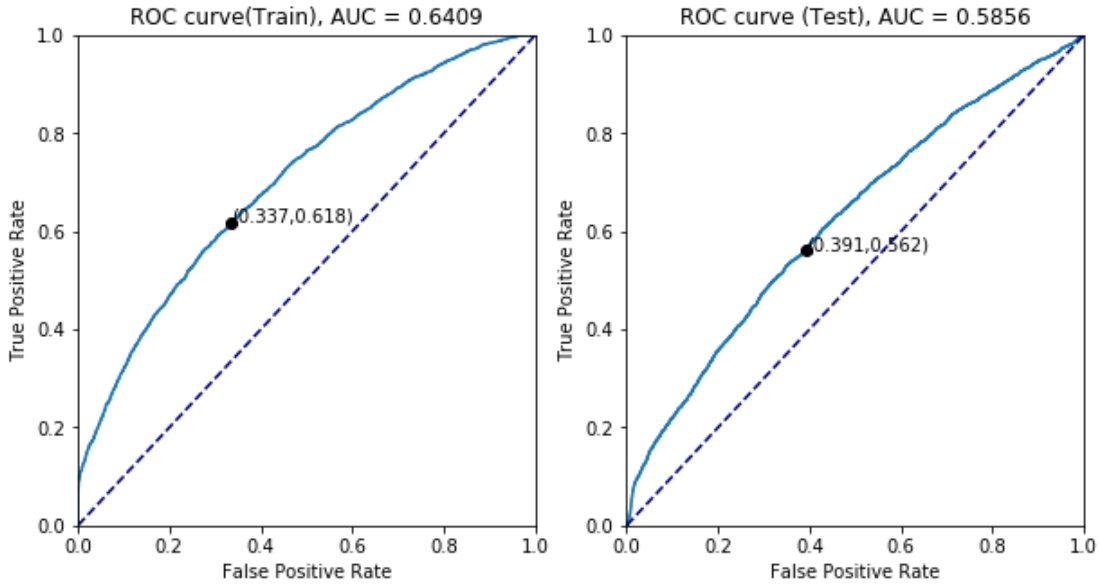


Figura 8.14: Curva ROC de regresión logística para experimento 2

La tendencia del submuestreo a generar un bajo rendimiento continua con un “Accuracy” 60% el modelo ofrece un bajo rendimiento con respecto a las predicciones correctas.

Clasificador Bayesiano (Naive Bayes)

		Predicción		
		Clase + (0)	**Clase – (1)**	**Total**
Real	**Clase + (0)**	61498	9115	70613
	Clase – (1)	1570	494	2064
	Total	63068	9609	72677

Tabla 8.22: Matriz de confusión de clasificador bayesiano (Experimento 1)

		Predicción		
		Clase + (0)	**Clase – (1)**	**Total**
Real	**Clase + (0)**	91327	14591	105918
	Clase – (1)	1620	444	2064
	Total	92947	15035	107982

Tabla 8.23: Matriz de confusión de clasificador bayesiano (Experimento 2)

Métricas	**Experimento 1**	**Experimento 2**
Accuracy	85.30%	84.99%
F-measure	8.46%	5.19%
Precision	87.09%	86.22%
False Positive Rate (FPR)	94.86%	97.05%
True Positive Rate (sensibility)	97.51%	98.26%
True Negative Rate (specificity)	23.93%	21.51%

Tabla 8.24: Tabla de métricas de evaluación de clasificador bayesiano

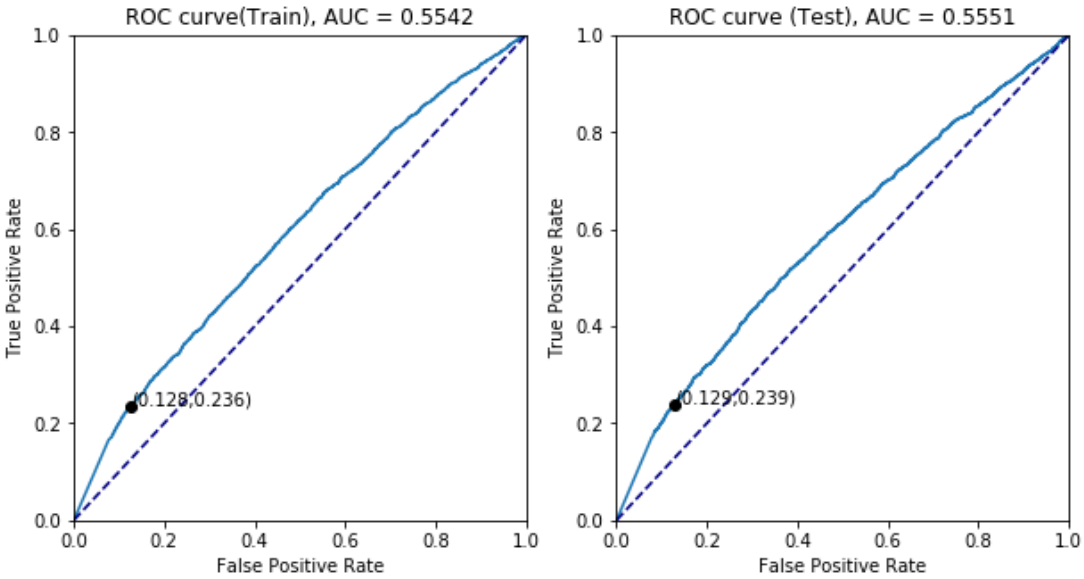


Figura 8.15: Curva ROC de clasificador bayesiano para experimento 1

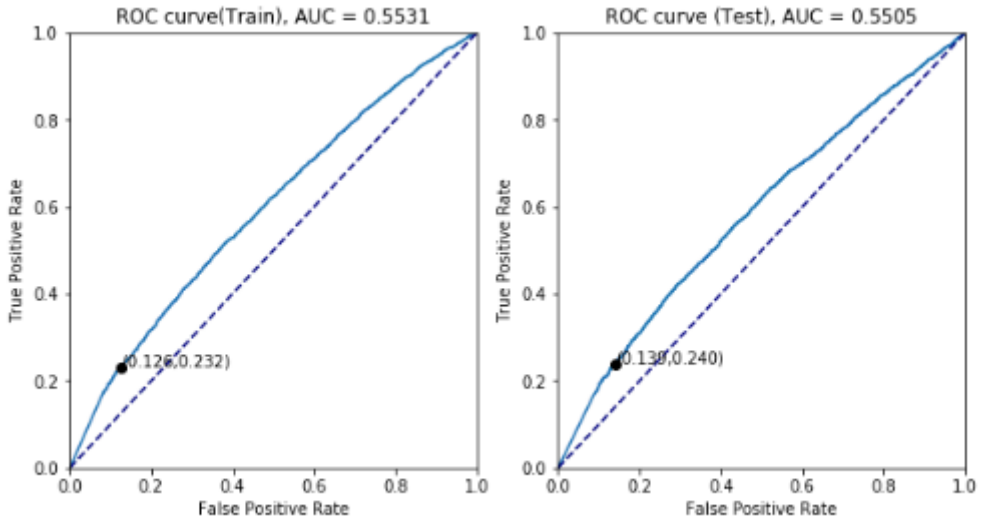


Figura 8.16: Curva ROC de clasificador bayesiano para experimento 2

El modelo de clasificación bayesiano se ve beneficiado en los experimentos 1 y 2, con un “Accuracy” de 85% el número de predicciones correctas aumenta.

8.3 Métricas de Sobremuestreo sintético (SMOTE).

Árboles de decisiones (Decision Tree)

		Predicción		
		Clase + (0)	**Clase – (1)**	**Total**
Real	**Clase + (0)**	61307	9306	70613
	Clase – (1)	1568	496	2064
	Total	62875	9802	72677

Tabla 8.25: Matriz de confusión de árboles de decisiones (Experimento 1)

		Predicción		
		Clase + (0)	**Clase – (1)**	**Total**
Real	**Clase + (0)**	94780	11138	105918
	Clase – (1)	1607	457	2064
	Total	96387	11595	107982

Tabla 8.26: Matriz de confusión de árboles de decisiones (Experimento 2)

Métricas	**Experimento 1**	**Experimento 2**
Accuracy	85.04%	88.20%
F-measure	8.36%	6.69%
Precision	86.82%	89.48%
False Positive Rate (FPR)	94.94%	96.06%
True Positive Rate (sensibility)	97.51%	98.33%
True Negative Rate (specificity)	24.03%	22.14%

Tabla 8.27: Tabla de métricas de evaluación de árboles de decisiones

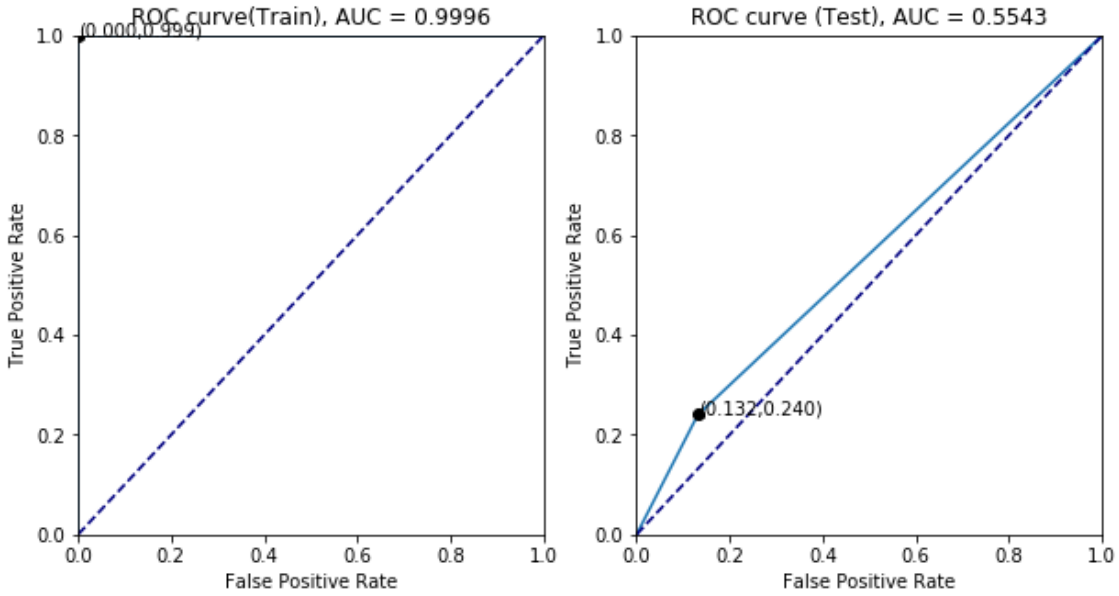


Figura 8.17: Curva ROC de árboles de decisiones para experimento 1

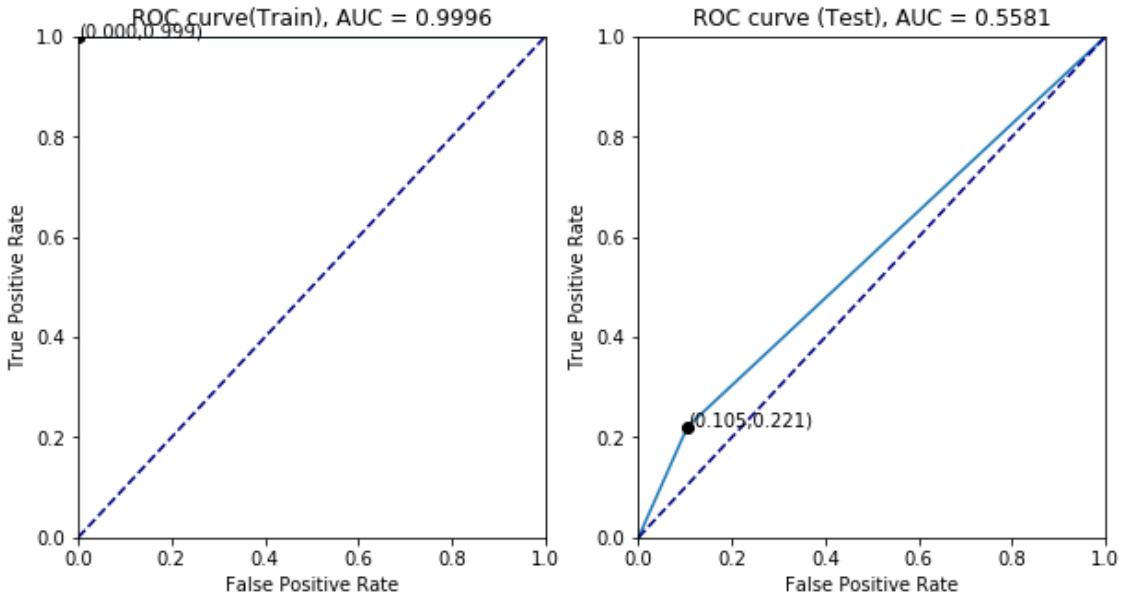


Figura 8.18: Curva ROC de árboles de decisiones para experimento 2

La generación artificial de observaciones de la clase minoritaria afecta el rendimiento del algoritmo disminuyendo “Accuracy” significativamente comparado a los resultados de sobremuestreo.

Bosques aleatorios (Random forest)

		Predicción		
		Clase + (0)	**Clase – (1)**	**Total**
Real	**Clase + (0)**	70392	221	70613
	Clase – (1)	1816	248	2064
	Total	72208	469	72677

Tabla 8.28: Matriz de confusión de bosques aleatorios (Experimento 1)

		Predicción		
		Clase + (0)	**Clase – (1)**	**Total**
Real	**Clase + (0)**	105696	222	105918
	Clase – (1)	1821	243	2064
	Total	107517	465	107982

Tabla 8.29: Matriz de confusión de bosques aleatorios (Experimento 2)

Métricas	**Experimento 1**	**Experimento 2**
Accuracy	97.20%	98.11%
F-measure	19.58%	19.22%
Precision	99.69%	99.79%
False Positive Rate (FPR)	47.12%	47.74%
True Positive Rate (sensibility)	97.49%	98.31%
True Negative Rate (specificity)	12.02%	11.77%

Tabla 8.30: Tabla de métricas de evaluación de bosques aleatorios

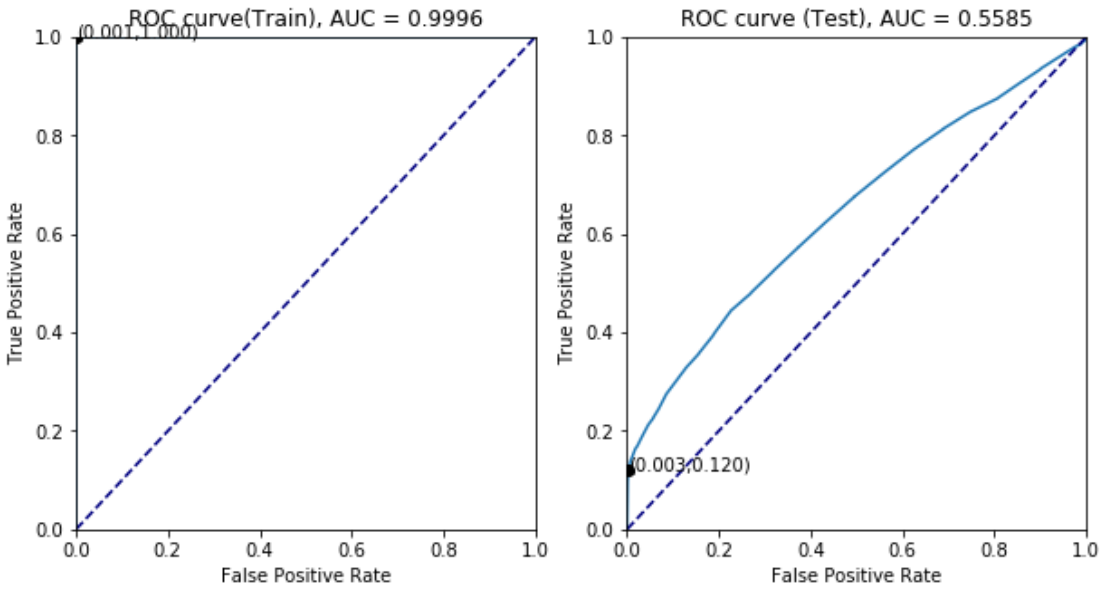


Figura 8.19: Curva ROC de bosques aleatorios para experimento 1

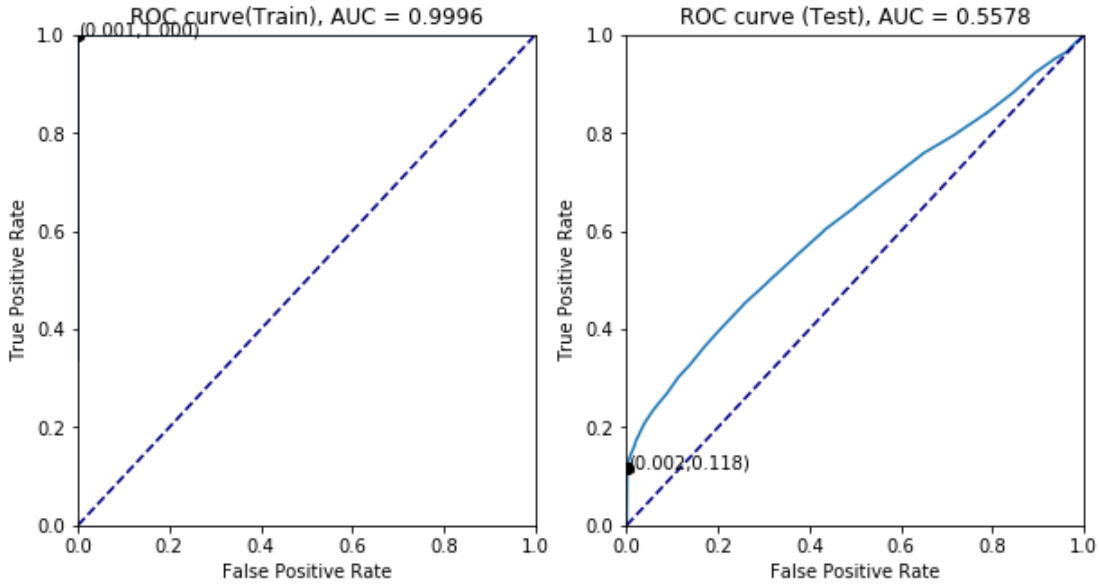


Figura 8.20: Curva ROC de bosques aleatorios para experimento 2

SMOTE genera un gran beneficio a las métricas de Bosques aleatorios siendo esta implementación la candidata a mejor solución.

Regresión logística (Logistic Regression)

		Predicción		
		Clase + (0)	**Clase – (1)**	**Total**
Real	**Clase + (0)**	45027	25586	70613
	Clase – (1)	931	1133	2064
	Total	46195	26719	72677

Tabla 8.31: Matriz de confusión de regresión logística (Experimento 1)

		Predicción		
		Clase + (0)	**Clase – (1)**	**Total**
Real	**Clase + (0)**	67952	37966	105918
	Clase – (1)	962	1102	2064
	Total	68914	39068	107982

Tabla 8.32: Matriz de confusión de regresión logística (Experimento 2)

Métricas	**Experimento 1**	**Experimento 2**
Accuracy	63.51%	63.95%
F-measure	7.87%	5.36%
Precision	63.77%	64.16%
False Positive Rate (FPR)	95.76%	97.18%
True Positive Rate (sensibility)	97.97%	98.60%
True Negative Rate (specificity)	54.89%	53.39%

Tabla 8.33: Tabla de métricas de evaluación de regresión logística

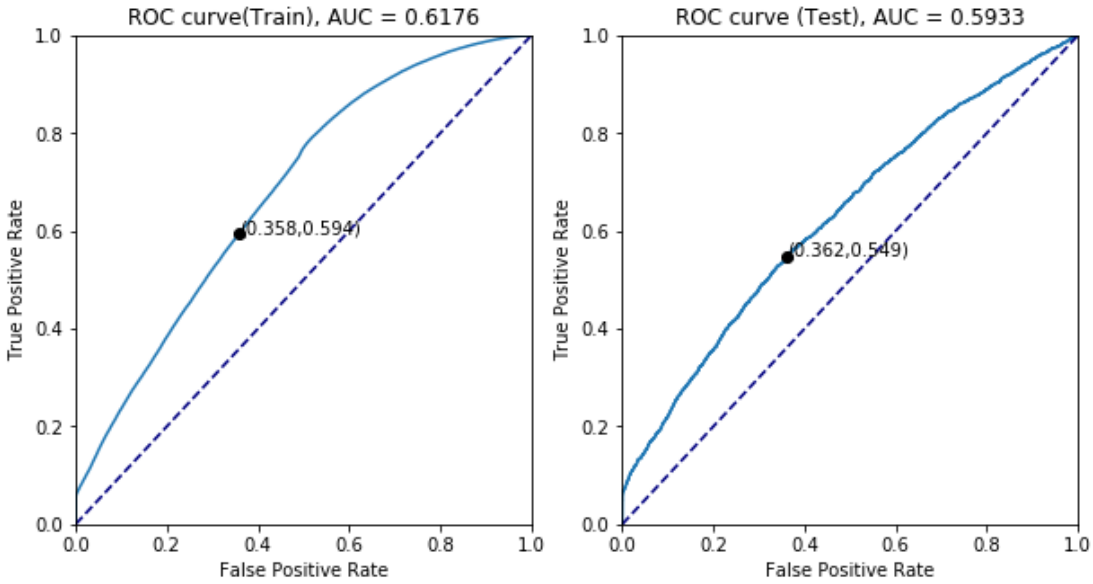


Figura 8.21: Curva ROC de regresión logística para experimento 1

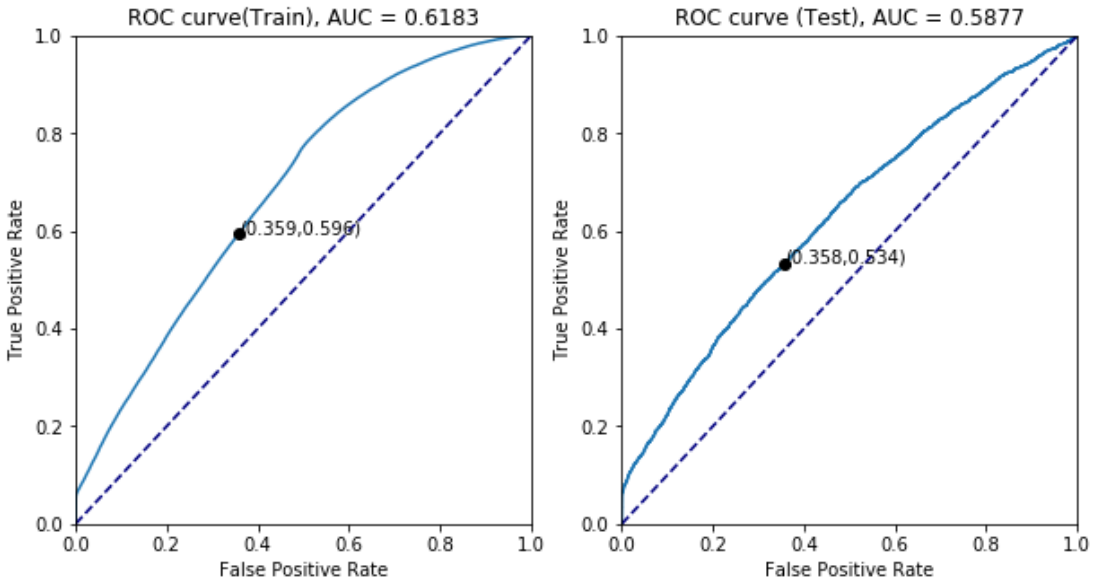


Figura 8.22: Curva ROC de regresión logística para experimento 2

El resultado de los experimentos 1 y 2 con la utilización de un conjunto de datos con observaciones artificiales no mejora el rendimiento del modelo ofreciendo apenas “Accuracy” del 63% siendo este resultado muy malo si lo comparamos a los otros modelos.

Clasificador Bayesiano (Naive Bayes)

		Predicción		
		Clase + (0)	**Clase – (1)**	**Total**
Real	**Clase + (0)**	14165	56448	70613
	Clase – (1)	398	1666	2064
	Total	14563	58114	72677

Tabla 8.34: Matriz de confusión de clasificador bayesiano (Experimento 1)

		Predicción		
		Clase + (0)	**Clase – (1)**	**Total**
Real	**Clase + (0)**	22052	83866	105918
	Clase – (1)	410	1654	2064
	Total	21472	85520	107982

Tabla 8.35: Matriz de confusión de clasificador bayesiano (Experimento 2)

Métricas	**Experimento 1**	**Experimento 2**
Accuracy	21.78%	21.95%
F-measure	5.54%	3.78%
Precision	20.06%	20.82%
False Positive Rate (FPR)	97.13%	98.07%
True Positive Rate (sensibility)	97.32%	98.17%
True Negative Rate (specificity)	80.72%	80.14%

Tabla 8.36: Tabla de métricas de evaluación de clasificador bayesiano

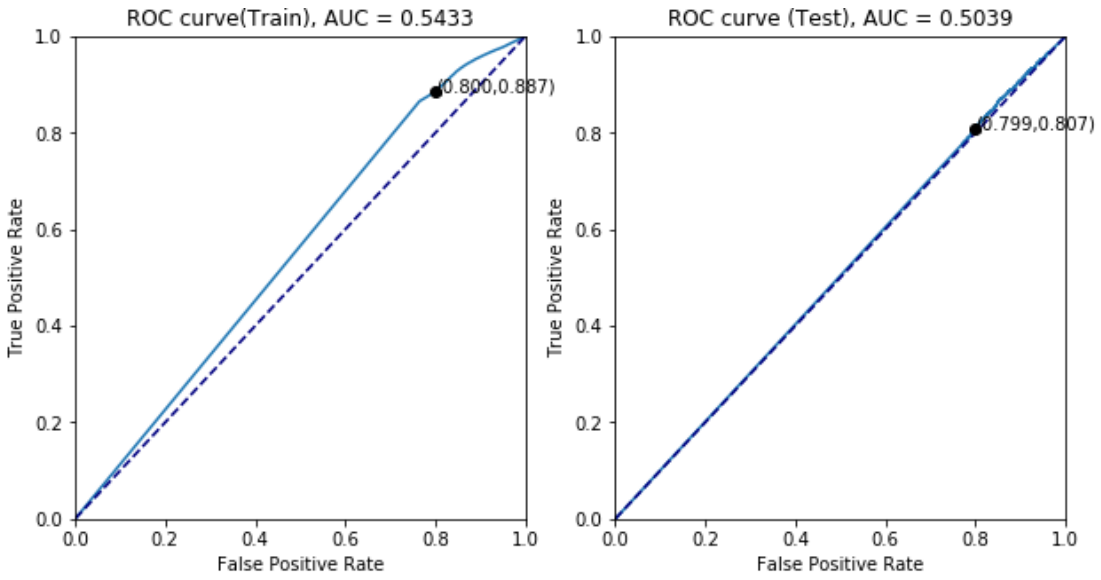


Figura 8.23: Curva ROC de clasificador bayesiano para experimento 1

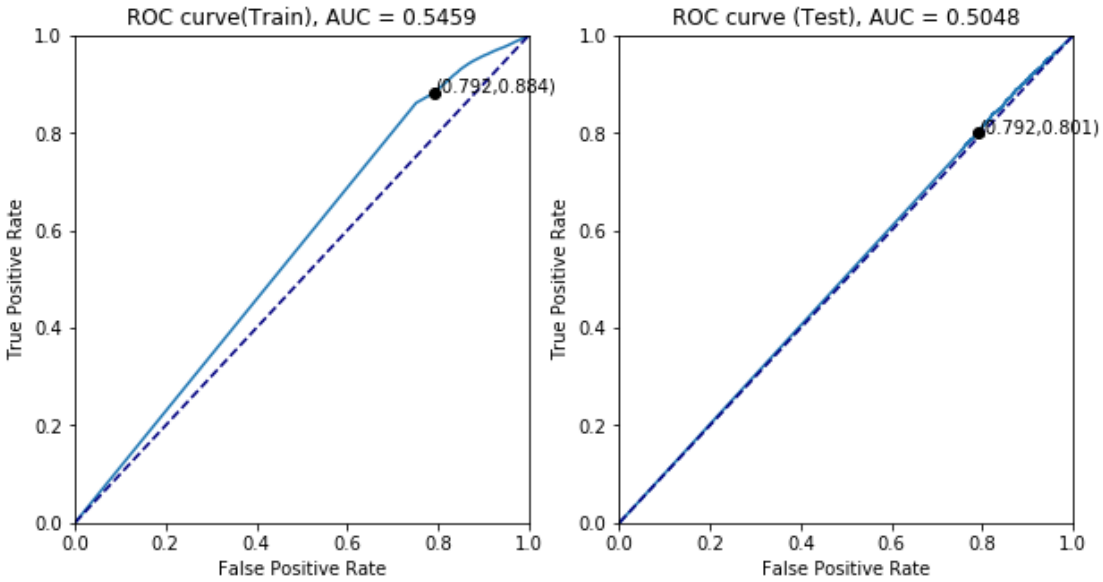


Figura 8.24: Curva ROC de clasificador bayesiano para experimento 2

El resultado final de este experimento con SMOTE los clasificadores bayesianos ofrecen el peor resultado con “Accuracy” de 22% de predicciones correctas es considerado el peor resultado.

8.4 Métricas de soporte de maquina vectorial de una clase (one-support vector machine)

		Predicción		
		Clase + (0)	**Clase – (1)**	**Total**
Real	**Clase + (0)**	70404	209	70613
	Clase – (1)	2045	19	2064
	Total	72449	228	72677

Tabla 8.37: Matriz de confusión de OSVM (Experimento 1)

		Predicción		
		Clase + (0)	**Clase – (1)**	**Total**
Real	**Clase + (0)**	105487	431	105918
	Clase – (1)	2048	16	2064
	Total	107535	447	107982

Tabla 8.38: Matriz de confusión de OSVM (Experimento 2)

Métricas	**Experimento 1**	**Experimento 2**
Accuracy	96.90%	97.70%
F-measure	1.66%	1.27%
Precision	99.70%	99.59%
False Positive Rate (FPR)	91.67%	96.42%
True Positive Rate (sensibility)	97.18%	98.10%
True Negative Rate (specificity)	0.92%	0.78%

Tabla 8.39: Tabla de métricas de evaluación de OSVM