TÍTULO

ALGORITMOS DE APRENDIZAJE AUTOMÁTICO PARA LA PREDICCIÓN DE LA PRODUCTIVIDAD DE LA COFECCIÓN TEXTIL

AUTORA

Lisandra Pérez Zaldívar

Esta edición electrónica ha sido realizada en 2022

Tutores: Dr. D. Diego Marín Santos ; Manuel Emilio Gegúndez Arias

Instituciones: Universidad Internacional de Andalucía ; Universidad de Huelva

Curso: *Máster en Economía, Finanzas y Computación (2020/21)*

Fecha documento: 2021

«Algoritmos de aprendizaje automático para la predicción de la productividad de la confección textil»

by

«Lisandra Perez Zaldivar»

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

University of Huelva & International University of Andalusia

Diciembre 2021

«Algoritmos de aprendizaje automático para la predicción de la productividad de la confección textil»

Lisandra Perez Zaldivar

Máster en Economía, Finanzas y Computación

«Supervisores:»

Dr. Diego Marín Santos

Dr. Manuel Emilio Gegúndez Arias

Universidad de Huelva y Universidad Internacional de Andalucía

2021

Abstract

The precise estimation of productivity is one of the key factors in planning production. Today, apparel companies are focused on forecasting supported by smart decision-making. In this research, a procedure is presented to detect if the productivity of the clothing production lines will be fulfilled, based on previous knowledge. For this, supervised learning techniques are applied, classifying productivity as fulfilled or not fulfilled. These techniques were applied to a set of unbalanced records from the production area of a textile garment company in Bangladesh. After a comparative analysis of the experimental data obtained in the testing process, it is shown that the classification algorithms are more efficient when the training set of data is balanced. Being the Random Forest technique is the one that offers the best results on the defined evaluation criteria (Precision, Sensitivity, and F1-Score). Finally, it can be visualized through the ROC curve that the Random Forest model trained with balanced data obtains the best AUC with 75% at the cut-off point when sensitivity reaches 73%, specificity 76%, and 1 as the optimal threshold. Being favorable for the process of identifying cases when productivity is not met.

JEL classification: I25, I28, J24, J28, O52.

Key words: productivity, knowledge, supervised learning, classifying, unbalanced data.

Resumen

La precisa estimación de la productividad es uno de los factores claves para planificar la producción. Actualmente, las empresas de la confección de prendas de vestir están enfocadas en realizar pronósticos sustentados por la toma de decisiones inteligentes. En esta investigación se presenta un procedimiento para detectar si la productividad de las líneas de producción de la confección será cumplida, en base a conocimientos previos. Para ello se aplican técnicas de aprendizaje supervisado, clasificando la productividad en cumplida o no cumplida. Estas técnicas fueron aplicadas a un conjunto de registros desbalanceados del área de producción de una empresa de confección textil de Bangladesh. Después de un análisis comparativo de los datos experimentales obtenidos en el proceso de pruebas, se demuestra que los algoritmos de clasificación son más eficientes cuando el conjunto de entrenamiento de datos esta equilibrado. Siendo la técnica de Random Forest la que mejor resultados ofrece sobre los criterios de evaluación definidos (Precisión, Sensibilidad y F1-Score). Finalmente, se puede visualizar mediante la curva ROC que el modelo Random Forest entrenado con datos balanceados obtiene el mejor AUC con un 75% en el punto de corte cuando la sensibilidad alcanza un 73%, especificidad un 76%, y como umbral óptimo 1. Siendo favorable para el proceso de identificación de los casos cuando la productividad no es cumplida.

Agradecimientos

Primero que todo quiero agradecer de manera especial a mis tutores Diego y Manuel que, gracias a su orientación y paciencia, me dieron las directrices a seguir para realizar este trabajo de fin de máster.

Agradecer, a mi familia que es mi más grande impulso para aventurarme en esta nueva etapa de mi vida, en especial a mi madre Nirian, mi padre Roberto, mi hermano Robertico, mi novio Arian y su familia, por darme todo el apoyo, fuerza y ánimos durante este año.

Gracias a todos mis buenos amigos Ever, Daimara, Jose Gabriel, Jennifer, Alfredo, Ismaray, Osmar, Ivian, Migue, Claudia y Alberto, que aun estando algunos de ellos lejos su apoyo incondicional nunca me ha faltado, siendo sus palabras de motivación motor impulsor para seguir adelante.

Finalmente, agradecer al claustro de profesores del Máster de economía, finanzas y computación del curso 2020-2021, por su profesionalidad, docencia, empatía y paciencia infinita con los estudiantes a pesar de las circunstancias por las que atravesamos en el momento actual.

Tabla de contenidos

Lista de Tablas

Tabla de figuras

Tabla de gráficas

1 Introducción

En la actual era digital las Tecnologías de la Información y las Comunicaciones (TICs) representan un papel fundamental en el sector industrial. Su uso es de gran utilidad para mejorar la productividad y competitividad de este. Para la gestión de la información de este sector se emplean actualmente técnicas de inteligencia artificial.

En España el sector industrial es clave dentro de la actividad económica, siendo específicamente la segunda rama de actividad más importante de la economía española, después del sector de los servicios. En el informe presentado en el 2019 por el consejo económico y social de España se plantea, que las ramas manufactureras representan el 12,6 % del PIB (Producto interno bruto) a precios corrientes y el 11,5 % del empleo equivalente a tiempo completo en el año 2018 [1].

El sector de la moda también representa un papel significativo en el ámbito económico y social. Conforme a los datos del estudio realizado en el "Informe sector moda en España" generó el 2.8% del PIB nacional y aportó el 4.1% al mercado laboral; siendo la producción textil, confección y comercialización de las prendas factores contribuyentes a generar estos indicadores económicos [2].

También, en el informe [1] se realiza un balance de la actividad económica española, explicando que la industria textil tiene una disminución en cuanto al valor agregado bruto (VAB) durante el periodo 2000-2017. Igualmente, en el informe de industria 2019 [3] se realiza un análisis de la industria manufacturera española. Se explica que el índice de producción industrial (IPI) del sector nacional de confección de prendas de vestir tiene un declive en el año 2015. Sin embargo, a partir del 2018 se logra aumentar el IPI en un 19,7% con respecto al año 2015.

La bibliografía consultada permite asegurar que el sector de la industria manufacturera constituye como factor relevante en la medición de la economía del país. Es por ello que los ingenieros industriales deben definir estratégicamente los objetivos a cumplir. Estos objetivos deben garantizar la continuidad del ritmo productivo, a pesar de los cambios repentinos que pueden surgir durante el desarrollo de la nación.

A nivel global las organizaciones y empresas están enfocadas en alcanzar estos propósitos gracias a las fábricas inteligentes, donde la tecnología puntera es la

inteligencia artificial y, en particular, el Machine Learning (Aprendizaje Automático) [4]. Las técnicas de aprendizaje automático permiten analizar grandes cantidades de datos logrando descubrir conocimientos significativos [5]. Durante el proceso industrial de confección de prendas, se va generando información que puede ser tratada con métodos computacionales de alta complejidad, descubriendo así datos útiles para la toma de decisiones.

Actualmente, son publicadas investigaciones enfocadas en el campo de estudio de la minería de datos y técnicas de aprendizaje automático sobre el sector textil. Estudios que tienen como objetivo mejorar la calidad de los productos, mejorar la producción, reducir los costos y minimizar el tiempo de fabricación.

El presente trabajo se enfoca en predecir si la productividad en el futuro será cumplida o no. Para ello se plantea como objetivo realizar el diseño, entrenamiento y validación de diferentes modelos de algoritmos de predicción, que mediante técnicas de aprendizaje supervisado clasifiquen el cumplimiento de forma efectiva de la productividad partiendo de un conjunto de datos históricos. Lo que supondrá pronósticos acertados, apoyando así la toma de decisiones del proceso productivo de una empresa, dedicada a la confección de prendas de vestir.

2 Problemática

En la industria de la confección de prendas de vestir un aspecto fundamental es el diseño y ejecución eficiente de los procesos productivos. Es de suma importancia trazar estrategias factibles que logren cumplir los objetivos propuestos en el tiempo establecido y con los costos previstos.

Para la producción de prendas de vestir se siguen fases y procesos que siguen un orden secuencial. Entre estos se pueden encontrar: la programación de las tareas, el abastecimiento de materia primas, diseño, selección del material, corte, costura y confección, acabado, inspección y entrega.

Para llevar a cabo el procedimiento de fabricación de prendas de vestir se organiza un plan estratégico que dirige las líneas de producción de los pedidos. En el plan se establece el inicio de la producción, se calcula el consumo de materiales para los productos a elaborar, la capacidad productiva por mano de obra y la fecha de

terminación. Además, se pautan indicadores estratégicos a alcanzar durante la producción, los mismos fomentan la eficiencia entre los trabajadores, con el fin de cumplir las metas establecidas.

Sin embargo, los indicadores reales de la producción que se obtienen no siempre cumplen con los planificados. Entre los factores que influyen en el incumplimiento, es la productividad planificada, donde no se realiza una estimación precisa de los medios o recursos efectivos. Como ejemplo se pueden mencionar: la cantidad de trabajadores destinados a un equipo, la cantidad de prendas a elaborar en un tiempo determinado, o el estímulo monetario para incentivar a los trabajadores. Lo antes expuesto implica afectaciones en el tiempo de producción, provocando retrasos de entregas de pedidos y costos no planificados de inversión en mano de obra.

Por lo tanto, predecir cuándo será cumplida o no la productividad, después de analizar un conjunto de datos históricos con la ayuda de diferentes algoritmos de clasificación, permitirá realizar pronósticos acertados para elaborar un diseño y ejecución eficiente de los procesos productivos basado en conocimientos previos.

3 Revisión de la bibliografía

En los últimos años numerosos trabajos científicos se han centrado en resolver problemáticas respecto a la industria de la confección textil. Entre los estudios realizados destacan las investigaciones respecto a perfeccionar la calidad de las prendas y mejorar el proceso de producción. Para ello se han aplicado distintos métodos basados en la estadística y en la inteligencia artificial, lo cual ha permitido describir y analizar diferentes escenarios respecto a estas situaciones.

Entre las investigaciones que estudian como estandarizar las tallas de las prendas de vestir con la mejor precisión posible, se encuentra el trabajo de [6], donde se explica la selección y extracción de características. El uso de estas características, basadas en medidas de interés objetivas, permitió descubrir subconjuntos relevantes de medidas corporales que requieren un cuidado especial durante el proceso de diseño de las prendas de vestir. Así mismo, los autores [7] haciendo uso del procesamiento digital inteligente de imágenes y redes neuronales, desarrollaron una metodología para la recomendación automática de las tallas de prendas masculinas.

También, se han realizado trabajos que investigan como mejorar los tiempos muertos de las líneas de la producción textil. Entre estos se encuentra la metodología propuesta por [8] donde utilizó la herramienta WEKA (Waikato Environment for Knowledge Analysis) y el método de árbol J48 de clasificación. La metodología logra la identificación y el análisis de las variables que generan las fallas y provocan el mayor intervalo de tiempos muertos en el sistema productivo.

Por otra parte, algunas investigaciones se enfocan en predecir con la mayor precisión los tipos de telas. Ejemplo de ello es el estudio que realizó [9] sobre la clasificación de tipos de telas mediante la comparación de descriptores utilizando imágenes. Mediante la aplicación de los algoritmos de aprendizaje supervisado Máquina de Vector Soporte (SVM) y Redes Neuronales (RNA), llegaron a la conclusión que el algoritmo de aprendizaje RNA fue el mejor método de clasificación junto al descriptor SSFT (Transformada de Fourier corta en espacio) ya que presentó un error de clasificación del 1 %.

4 Marco Teórico

4.1 Inteligencia Artificial

La inteligencia artificial (IA) es definida por [10] como «el área del conocimiento de las ciencias de la computación cuyo objetivo es desarrollar soluciones tecnológicas que posean la capacidad de resolver problemas que requieran inteligencia humana». También [11] define la inteligencia artificial como «la capacidad de un sistema para interpretar correctamente datos externos, para aprender de dichos datos y emplear esos conocimientos para lograr tareas y metas concretas a través de la adaptación flexible».

4.2 Aprendizaje automático

El machine learning (en inglés) o aprendizaje automático, es una rama de la Inteligencia Artificial que permite el desarrollo de sistemas que aprenden automáticamente a partir de datos, identificando y analizando patrones para luego sacar conclusiones que son útiles en la toma de decisiones [12] [13].

Tipos de aprendizajes automático

Los algoritmos de Aprendizaje Automático se clasifican en varias categorías según la variable de salida, entre las que se pueden mencionar:

Aprendizaje supervisado: permite realizar predicciones basadas en datos históricos que están asociados a etiquetas o variables de respuestas conocidas. Su funcionamiento parte cuando se tienen las variables de entrada X y variables de salida Y, se utiliza un algoritmo para inferir la función F(x)=Y. Donde aprende los patrones de entrada que luego entren nuevos datos, se infiera el valor de la salida [14] [15].

- Clasificación: su objetivo es predecir valores de la variable dependiente que son tipo categóricos o nominales. Entre los problemas que resuelven se puede mencionar: el diagnostico de enfermedades, predicción de transacciones fraudulentas o reconocimiento de texto.
- Regresión: tiene como objetivo predecir los valores de salida de la función, pero como un valor numérico real, ejemplo datos monetarios.

Aprendizaje no supervisado: conjunto de algoritmos que permiten diseñar modelos para extraer información, donde solo se tienen las variables con los datos de entrada X, pero no se cuenta con datos previos de salida. Estos algoritmos parten de inferir patrones similares de un conjunto de datos de entrada sin estar previamente etiquetados para relacionarlos entre sí, resolviendo tareas como agrupamiento o asociación de los datos.

La presente investigación está enfocada en la categoría de aprendizaje supervisado, donde se estudian las técnicas de clasificación que se exponen en el epígrafe 4.3.

4.3 Técnicas de clasificación

En este trabajo se emplean las siguientes técnicas de clasificación:

1. **k Vecinos más cercanos (k Nearest Neighbors KNN)**

   El KNN es un método de clasificación no paramétrico basado en instancias para resolver los problemas de clasificación y regresión. En el caso de la clasificación la entrada consiste en k ejemplos de entrenamiento, donde el valor de la salida es asignado según una medida de similitud o distancia de las clases más cercanas a sus k vecinos [16]. Las funciones para calcular la distancia son:

- Distancia Euclidiana
- Distancia Manhattan
- Distancia Minkowski
- Distancia Hamming.

La medida de distancia que se utiliza con mayor frecuencia es la euclidiana, y se calcula utilizando la ecuación (1) [16].

$$d(x_i, x_j) = \sqrt{\sum (x_{ik} - x_{jk})^2} \quad (1)$$

Donde:

- x_i es el dato para clasificar.
- x_j es el dato almacenado.
- x_{ik} es el valor del atributo k para el ejemplo x i.
- x_{jk} es el valor del atributo k para el ejemplo $x\,j$.

Así, la similitud es calculada con la ecuación (2): [16]

$$s(x_i, x_j) = \frac{1}{1+d(xi,xj)} \quad (2)$$

La tarea de clasificación consiste en calcular la similitud entre el ejemplo a clasificar xi y todos los ejemplos de entrenamiento xj que se tienen almacenados. A continuación, se seleccionan los k vecinos más cercanos, es decir, aquellos que tienen un mayor valor en la función s. Por último, se calcula la clase de salida como la clase mayoritaria de los k vecinos más cercanos [16].

2. **Regresión logística**

Es una técnica de clasificación estadística de análisis multivariado. Puede ser utilizada tanto para análisis explicativos como predictivos. Estima los valores discretos de la variable dependiente, dependiendo de un conjunto determinado de variables independientes o predictores. Su objetivo es estimar las probabilidades y predecir la probabilidad de ocurrencia mediante la función logística [17].

3. **Support-vector Machine (SVMs)**

La máquina de vector soporte (SVMs, por sus siglas en inglés: Support Vector Machines), es un clasificador de patrones que se basa en marcos de aprendizaje estadísticos. Las SVMs permiten resolver problemas de regresión y clasificación. Dado un conjunto de entrenamiento que presentan patrones construye un modelo que asigna los nuevos ejemplos a etiquetas o clases.

Clasificación lineal: «Las SVM generan un hiperplano que separa los patrones con clase positiva (yi =+1) de los patrones con clase negativa (yi =-1). Los puntos xi que están en el hiperplano satisfacen $w^T x + b = 0$. El objetivo es encontrar el hiperplano óptimo, en términos de margen máximo. El margen se define como la distancia entre el objeto de cada clase más cercano al hiperplano. Los vectores de soporte son los puntos que tocan el límite del margen», [18] como se muestra en la Figura 1.

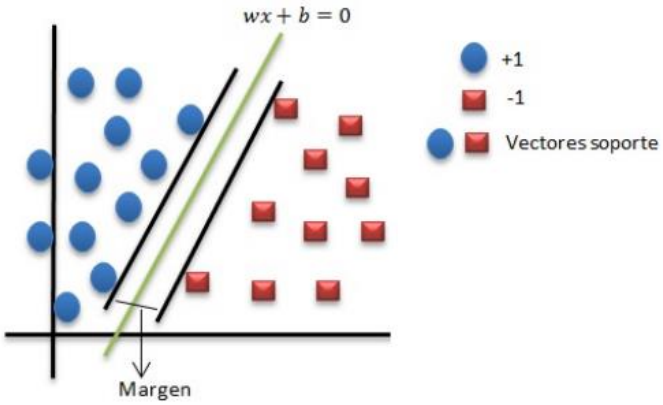


Figura 1 Ejemplo de clasificación lineal en SVM. fuente [18]

Clasificación no lineal: para la clasificación no lineal se realiza una transformación del espacio de entrada en una dimensión más alta, con el objetivo lograr la separabilidad lineal entre los datos. Para ello se utiliza una función kernel que pueden ser de tipo: Gausiano RBF, Polinomial y Lineal [18].

4. Árboles de decisión

Los árboles de clasificación siguen un enfoque de aprendizaje supervisado que se utiliza para resolver problemas de clasificación. Para su ejecución se parte de un nodo raíz y se crean nodos hojas en los extremos que se van conectando mediante las ramas. Esta técnica tiene forma de árbol, las hojas representan las etiquetas de las clases y las ramas son las instancias que conducen a esas clases, enlazando así con los nodos internos que serían los atributos. El árbol se extiende hacia abajo y generalmente se dibuja de izquierda a derecha. El nodo inicial se llama nodo raíz, mientras los nodos en los extremos de la cadena se les conocen como nodos hoja [19].

4.1 Bosques aleatorios (Random Forest)

El algoritmo Random forest es un método de clasificación supervisado que crea un bosque de árboles de decisión independientes probados sobre un conjunto de datos. En la creación de los árboles independientes durante la fase de

entrenamiento, se va seleccionando aleatoriamente un porcentaje de datos y variables para realizar la partición de nodos. Después que se han creado los árboles, se realiza una evaluación a cada uno de ellos, obteniendo, así como predicción la media que han arrojado todos los árboles.

5. **Naïve-Bayes (NBC)**

   Es un método basado en el teorema de Bayes, donde clasifica de forma probabilística basado en un supuesto de independencia entre las variables independientes. Durante el entrenamiento el algoritmo aprende la distribución de probabilidad condicional de los valores de una clase respecto a los valores de los atributos, para predecir la clase a la cual pertenecen los ejemplos posteriores de prueba con la mayor probabilidad.

4.4 Desbalanceo de clases

El desbalanceo de los datos es uno de los problemas que se encuentra con mayor frecuencia en los conjuntos de datos para la clasificación. Esto significa que la distribución de las clases no es homogénea, y están desproporcionadas [20] [21].

Las bases de datos no equilibradas influyen negativamente en el rendimiento predictivo de las técnicas de clasificación, ya que estas se basan en la precisión y al detectar mayor presencia de patrones de la clase mayoritaria, disminuye la eficiencia para detectar los de la clase minoritaria. Siendo en algunas investigaciones los datos que pertenecen a la clase de menor representatividad los de mayor interés en clasificar, lo que puede implicar un inconveniente si no se clasifica correctamente. Durante el preprocesamiento de los datos se aplican técnicas que realizan un remuestreo de los mismos, balanceando así el número de instancias que pertenecen a cada clase [20] [21].

Las técnicas pueden ser de submuestreo, las cuales son efectivas para conjuntos con una gran cantidad de datos, ya que su aplicación es eliminar las instancias mayoritarias. Entre las que se puede mencionar la técnica de Random undersampling que elimina de forma aleatoria muestras de la clase mayoritaria. También es utilizada la técnica Tomek Links [22] donde se eliminan las muestras que más se repiten o las instancias de la clase mayoritaria que tienen la menor distancia con las muestras de la clase minoritaria.

También, se encuentran las técnicas de sobremuestreo que su aplicación es ir generando instancias del conjunto de datos de la clase minoritaria hasta compensar con la cantidad

de la clase mayoritaria [20] [21]. Entre las técnicas que adicionan nuevas muestras a la clase minoritaria con el objetivo de suprimir la diferencia entre los datos, se pueden mencionar: SMOTE, Random oversampling y ADASYN (Adaptative Synthetic Sampling). Donde la técnica que será utilizada para realizar el balanceo de los datos durante el desarrollo de este trabajo es la técnica SMOTE.

Técnicas de balanceo de datos- sobremuestreo

SMOTE (Synthetic Minority Over-Sampling Technique)

Entre las técnicas de sobremuestreo más utilizada se encuentra el algoritmo SMOTE de minorías sintéticas. Este algoritmo va seleccionando las muestras con distancia más cercana y traza una línea entre estas muestras, luego dibuja una nueva instancia en un punto de esa línea. Con la creación de estos nuevos puntos sintéticos se alcanza un equilibrio homogéneo entre las clases mayoritaria y minoritaria [20] [21].

4.5 Métodos de validación

Los métodos de validación tienen como objetivo comprobar que tan precisas pueden llegar a ser las predicciones de los modelos de aprendizaje automático. Luego de obtener el conjunto de datos de entrenamiento, ajustado por el algoritmo de aprendizaje, se ejecuta la validación sobre este conjunto. El procedimiento radica en dividir el conjunto de datos de entrada en subconjuntos de entrenamiento y prueba, para ajustar el grupo de entrenamiento y luego evaluar en el de pruebas. Obteniendo una puntuación en la evaluación respecto a la precisión y error de predicción de este. Esto se realiza un número determinado de veces y los resultados de las puntuaciones se promedian para obtener un resultado final de todas las evaluaciones realizadas.

Para evaluar los modelos de aprendizaje desarrollados en el presente trabajo se hace uso del método de validación cruzada con n particiones [23]. Este método consiste en dividir el conjunto de datos de entrenamiento en subconjuntos distintos. El proceso de ajuste se realiza con n-1 subconjuntos y se evalúa con el subconjunto de datos restante. Esto se repite n veces, dejando siempre un subconjunto de datos diferente para evaluar, lo que genera distintas puntuaciones de exactitud en cuanto a predicción y tasas de error. Finalmente se genera el modelo con el total del conjunto de datos y se calcula la media de las puntuaciones obtenidas.

4.6 Métricas de evaluación

Entre los métodos utilizados para evaluar el desempeño de los modelos de clasificación están la matriz de confusión y la gráfica ROC (característica operativa del receptor).

Matriz de confusión

La matriz de confusión es una herramienta que se utiliza para mostrar un reporte del desempeño de un algoritmo de aprendizaje supervisado. Los resultados se representan como la cantidad de predicciones de cada clase en las columnas, y en las filas se visualizan las instancias de la clase real. En este caso se representa una matriz de confusión para el caso de clasificación binaria.

		Matriz de confusión	
		Predicción	
		0	**1**
Realidad	**0**	**VN**	**FP**
	1	**FN**	**VP**

Figura 1 Matriz de confusión

- Verdadero Negativo: número de clasificaciones correctas de la clase negativa o 0.
- Falso Negativo: número de clasificaciones incorrectas de la clase negativa o 0.
- Verdadero Positivo: número de clasificaciones correctas de la clase positiva o 1.
- Falso Positivo: número de clasificaciones incorrectas de la clase positiva o 1.

Con la ayuda de estos valores se estiman varias métricas que atienden a diferentes aspectos como son:

Precisión: es una métrica que mide la calidad del modelo propuesto. De todas las positivas que se han predicho cuántas son realmente positivas.

- Precisión = Verdadero Positivo / (Verdadero Positivo + Falso Positivo)

Exhaustividad (Recall): es una métrica que brinda la cantidad de casos que el modelo de clasificación es capaz de identificar. De todas las clases positivas cuántas se predijo correctamente.

- Recall = Verdadero Positivo / (Verdadero Positivo + Falso Negativo)

Exactitud (Accuracy): métrica que mide el porcentaje de exactitud de casos que el modelo predice. De todas las clases cuales se predijeron correctamente.

- Accuracy = (Verdadero Positivo + Verdadero Negativo) / Total

Accuracy es una métrica recomendable para evaluar un modelo si las clases están balanceadas, pero no es confiable cuando se trabaja con un conjunto de datos no balanceados. Esto significa que le modelo aprende mejor de la clase mayoritaria, pero no siendo así para la clase minoritaria. Por lo que el valor del accuracy puede verse maximizado por el aprendizaje de la clase mayoritaria, suponiendo una falsa exactitud de clasificación de muestras para ambas clases.

F1-Score (media armónica): métrica que mide el porcentaje de la media armónica de los valores obtenidos de las métricas de precisión y de la sensibilidad. Esta métrica es útil para encontrar una media entre la precisión y la sensibilidad del clasificador. En casos de conjuntos de datos no balanceados es necesaria para identificar qué porcentaje medio de predicciones reales realiza el clasificador.

- F1-score = 2 * ((Precisión + Sensibilidad) / Precisión + Sensibilidad)

Curva ROC

La curva ROC (característica operativa del receptor) es otra herramienta utilizada para conocer el desempeño de los modelos de clasificación. Las métricas que componen los ejes de la curva se calculan con los valores obtenidos en la matriz de confusión [24]:

- Sensibilidad: Tasa de verdaderos positivos en el eje (y): TVP = VP/(VP+FN)

- Especificidad: Tasa de verdaderos negativos en el eje (x): TVN = VN/(FP+VN) La especificidad es la probabilidad de clasificar correctamente un registro cuyo valor sea el definido como negativo.

Se representa mediante un gráfico mostrando la tasa de verdaderos positivos en las ordenadas o sensibilidad frente a la tasa de falsos positivos o 1-especificidad en las abscisas. Un gráfico de curva ROC representa la métrica TVP frente a la Tasa de Falsos Positivos de cada uno de los posibles puntos de corte en diferentes umbrales de la clasificación [25] [24].

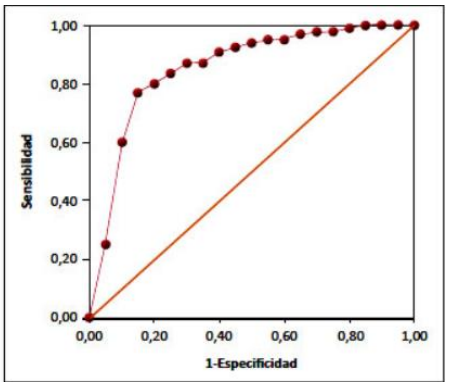


Figura 2 Gráfico de curva ROC. Fuente [24].

Otra métrica utilizada para evaluar el desempeño de las técnicas de clasificación es calcular el área bajo la curva ROC. Esta se obtiene y se determina como parámetro AUC (abreviado de área bajo de la curva), siendo el área comprendida que está en el punto de corte donde se alcanza la mayor sensibilidad y 1-especificidad. Donde el modelo más eficiente es el que ofrece un valor AUC próximo o igual a 1 en el eje correspondiente a la sensibilidad, o menos eficiente si el valor es igual o menor que 0,5 [24].

5 Metodología

En el presente capítulo se define la metodología para desarrollar los modelos predictivos propuestos. Donde se especifican los siguientes objetivos específicos:

1. Seleccionar la base de datos que contiene información de una fábrica de producción textil de Bangladesh.

2. Realizar un análisis exploratorio de los datos para visualizar y detallar las características de las variables, identificar valores perdidos y observar la correlación que existe entre ellas.

3. Preprocesamiento del conjunto de datos.

4. Elaborar el diseño y entrenamiento de los clasificadores.

5. Evaluar los clasificadores mediante métricas

6. Elegir el mejor clasificador según las métricas propuestas para la investigación.

Las muestras de la base de datos no están equilibradas respecto a la variable respuesta, por lo tanto, se realizarán dos experimentos durante el entrenamiento de los modelos. Un primer experimento de entrenar los modelos con el conjunto de datos sin balancear. Seguidamente el segundo experimento será entrenar los modelos con el conjunto de datos balanceados mediante la técnica SMOTE. Luego se tienen los resultados de evaluación sobre el conjunto de pruebas, se comparan de acuerdo con las métricas seleccionadas, donde se decide cual es el más eficiente en cuanto a capacidad de predicción.

Finalmente, para la elección del mejor clasificador se tendrá en cuenta las métricas Precisión, Recall y F1-Score. Luego se realizará la gráfica de la curva ROC-AUC del modelo seleccionado, especificando el punto de corte donde obtiene el mejor valor de sensibilidad y 1-especifidad.

Se utiliza el lenguaje de programación multiparadigma de código abierto Python en su versión 3.7.6, ya que posee numerosas librerías útiles para la exploración y preprocesamiento de los datos. Siendo, entre otras, las librerías de la biblioteca Scikit-learn las utilizadas para el desarrollo del notebook de este proyecto. Esta biblioteca contiene los principales algoritmos y funciones, que son usados en las etapas de preprocesado, entrenamiento, validación y pruebas de los modelos predictivos a crear.

6 Desarrollo y análisis de algoritmos

En este capítulo se aplica la metodología definida, realizando como primer paso la tarea de análisis exploratorio de los datos, luego el preprocesamiento de estos y finalmente la ejecución de los dos experimentos de entrenamiento de los modelos.

6.1 Conjunto de datos

Para la realización de este estudio se seleccionó la base de datos Productivity Prediction of Garment Employees publicado en el repositorio Center for Machine Learning and Intelligent Systems UCI [26]. En este conjunto de datos se almacena información extraída de una fábrica de producción textil de Bangladesh. La información almacenada

es específicamente de los departamentos de Costura-Confección y Terminado. La misma se recopiló en el periodo del 1 de enero del 2015 hasta marzo del mismo año.

El conjunto de datos está formado por 1197 registros y 15 variables explicativas que detallan la información del proceso de producción realizado por día, departamento y equipos de trabajo.

Descripción de las variables

1. date: variable tipo fecha formada por la estructura MM-DD-AAAA. Recoge los registros de producción por fecha.

2. day: variable categórica que contiene el nombre de los días de la semana exceptuando el viernes.

3. quarter: variable categórica que agrupa la información de la producción por semanas.

4. department: variable categórica que almacena los 2 tipos de departamentos que componen el proceso de producción de la confección. Sewing(Costura-Confección) y Finishing(Terminado).

5. team: variable numérica que identifica el número del equipo. En total son 12 equipos que trabajan en los departamentos de producción.

6. no_of_workers: variable numérica que acumula la cantidad de trabajadores por cada equipo.

7. no_of_style_change: variable numérica que almacena la cantidad de cambios que se realizan a una prenda durante el proceso de confección.

8. target_productivity: variable numérica que registra el índice de producción objetivo que debe cumplirse por jornada.

9. smv: variable numérica que supone el tiempo establecido para realizar las tareas.

10. wip: variable numérica que recoge la cantidad de productos que se deben fabricar, además incluye elementos no terminados de los productos.

11. over_time: variable numérica que supone la cantidad de tiempo extra que se le otorga a cada equipo en minutos.

12. incentive: variable numérica que representa la cantidad de dinero asignado para motivar a los trabajadores a cumplir los objetivos de producción.

13. idle_time: variable numérica que representa la cantidad de tiempo que pudo estar interrumpida la producción.

14. idle_men: variable numérica que representa la cantidad de trabajadores que estuvieron inactivos durante las interrupciones de la producción.

15. actual_productivity: variable numérica que recoge los índices de la productividad real de los trabajadores.

6.2 Exploración de los datos

Durante el análisis exploratorio de los datos (EDA) se identifica si existen variables con valores nulos, la distribución de los datos, además de visualizar el comportamiento general de la información recopilada con la ayuda de gráficos y tablas.

Tipos de datos

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1197 entries, 0 to 1196
Data columns (total 15 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   date                   1197 non-null   object
 1   quarter                1197 non-null   object
 2   department             1197 non-null   object
 3   day                    1197 non-null   object
 4   team                   1197 non-null   int64
 5   targeted_productivity  1197 non-null   float64
 6   smv                    1197 non-null   float64
 7   wip                    691 non-null    float64
 8   over_time              1197 non-null   int64
 9   incentive              1197 non-null   int64
 10  idle_time              1197 non-null   float64
 11  idle_men               1197 non-null   int64
 12  no_of_style_change     1197 non-null   int64
 13  no_of_workers          1197 non-null   float64
 14  actual_productivity    1197 non-null   float64
dtypes: float64(6), int64(5), object(4)
memory usage: 140.4+ KB
```

Figura 3 Información del conjunto de datos.

El conjunto de datos contiene 6 variables con valores numéricos de tipo float, 5 variables de tipo entero y 4 variables tipo categóricas. También se puede observar que la variable wip contiene solo 691 observaciones no nulas del total.

Descripción de las variables numéricas.

En la figura 4 se puede observar el comportamiento de los estadísticos descriptivos almacenados en las variables de tipo entero y float de la base de datos.

	count	mean	std	min	25%	50%	75%	max
team	1197.0	6.426901	3.463963	1.000000	3.000000	6.000000	9.000000	12.000000
targeted_productivity	1197.0	0.729632	0.097891	0.070000	0.700000	0.750000	0.800000	0.800000
smv	1197.0	15.062172	10.943219	2.900000	3.940000	15.260000	24.260000	54.560000
wip	691.0	1190.465991	1837.455001	7.000000	774.500000	1039.000000	1252.500000	23122.000000
over_time	1197.0	4567.460317	3348.823563	0.000000	1440.000000	3960.000000	6960.000000	25920.000000
incentive	1197.0	38.210526	160.182643	0.000000	0.000000	0.000000	50.000000	3600.000000
idle_time	1197.0	0.730159	12.709757	0.000000	0.000000	0.000000	0.000000	300.000000
idle_men	1197.0	0.369256	3.268987	0.000000	0.000000	0.000000	0.000000	45.000000
no_of_style_change	1197.0	0.150376	0.427848	0.000000	0.000000	0.000000	0.000000	2.000000
no_of_workers	1197.0	34.609858	22.197687	2.000000	9.000000	34.000000	57.000000	89.000000
actual_productivity	1197.0	0.735091	0.174488	0.233705	0.650307	0.773333	0.850253	1.120437

Figura 4 Estadísticos del conjunto de datos.

Se observa que el valor medio de variable targeted_productivity es de 0.72%. Representando una desviación típica de un 0.097%, y como valor máximo obtuvo un 0.80%.

Por otro lado, el smv o tiempo asignado para una tarea determinada es aproximadamente 15.06 minutos, con un valor mínimo de 2.9 minutos y como máximo 54.56. Además, tiene un total de 10.94 minutos de desviación típica para los valores de esta variable.

La cantidad de trabajo en progreso o elementos por terminar (wip) ha mediado entre los 1190.46 con una desviación típica de 1837.45. Siendo 7 el valor mínimo registrado y 23122 el máximo valor asignado de trabajo por hacer.

La variable que recoge el tiempo extra de cada equipo tiene 0 como valor mínimo en minutos y como máximo 25920 minutos. Además, toma como valor medio 4567.46 minutos y posee una desviación típica de 3348.82 minutos.

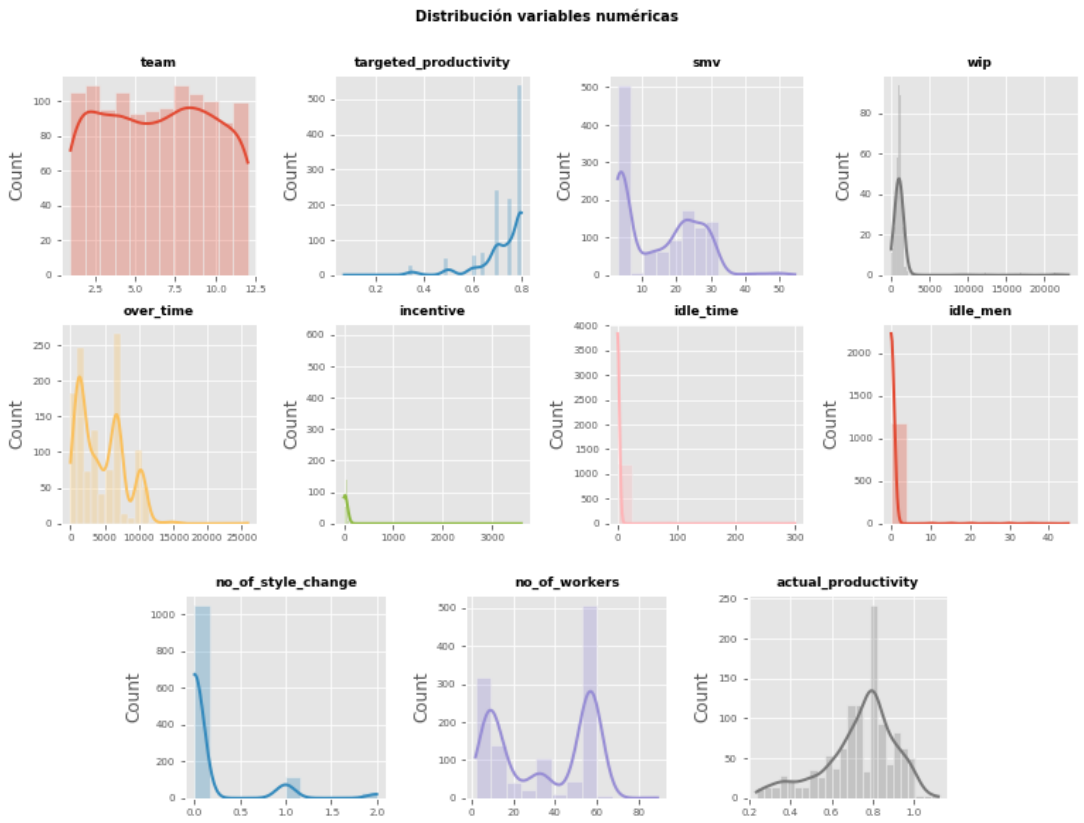
El incentivo que se le ofrece a los trabajadores en valor monetario para motivar el trabajo o cumplimiento de productividad es de 38.21 unidades monetarias aproximadamente, con una desviación típica de 160.182. Donde adquiere como máximo valor un total de 3600.

La variable idle_time que representa la cantidad de tiempo que demora una interrupción de la producción tiene un valor medio de 0.73 minutos, y una desviación típica de 12.70 minutos. Además, registra como valor máximo de tiempo un total de 300 minutos.

El máximo de trabajadores que estuvieron inactivos debido a la interrupción de la producción fue de 45 trabajadores, donde la media fue de 0.36 aproximadamente, y con una desviación típica de 3.26 trabajadores inactivos.

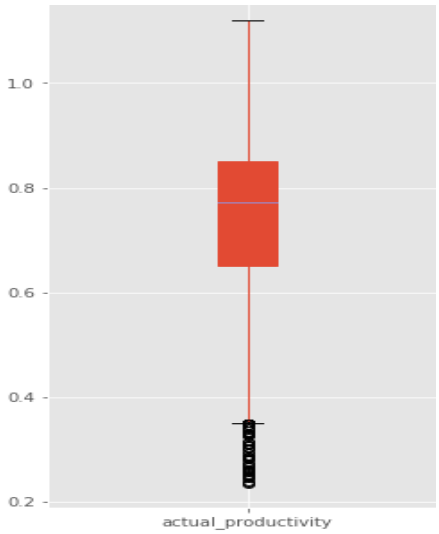
Los cambios de estilos realizados a un producto durante el proceso de producción toman un valor medio de 0.15 cambios, con una desviación típica de 0.42. Además, hay procesos de producción que su producto no sufrió cambios en el estilo y hay otros que arrojan hasta 2 modificaciones como valor máximo.

Por último, se observa que la productividad real obtenida media en un 0.73% con una desviación típica de 0.17%. La baja desviación típica indica que la mayor parte de los valores del conjunto de datos están agrupados cerca de la media y, por lo tanto, no se supera en gran medida la productividad objetivo que marca la empresa siendo esta de 0.8%. Además, se puede observar que el valor máximo de productividad que se ha obtenido es de 1.12%.



Gráfica 1 Distribución de las variables numéricas en el conjunto de datos.

En la gráfica 2 de caja se puede observar que los valores de la productividad real se concentran entre 0.65% y 0.85% aproximadamente. Además, se observan que los valores se centran por debajo del 0.8%.



Gráfica 2 Distribución de los valores de la variable "actual_productivity"

En la figura 5 se muestra la matriz de correlación existente entre las características de la base de datos.

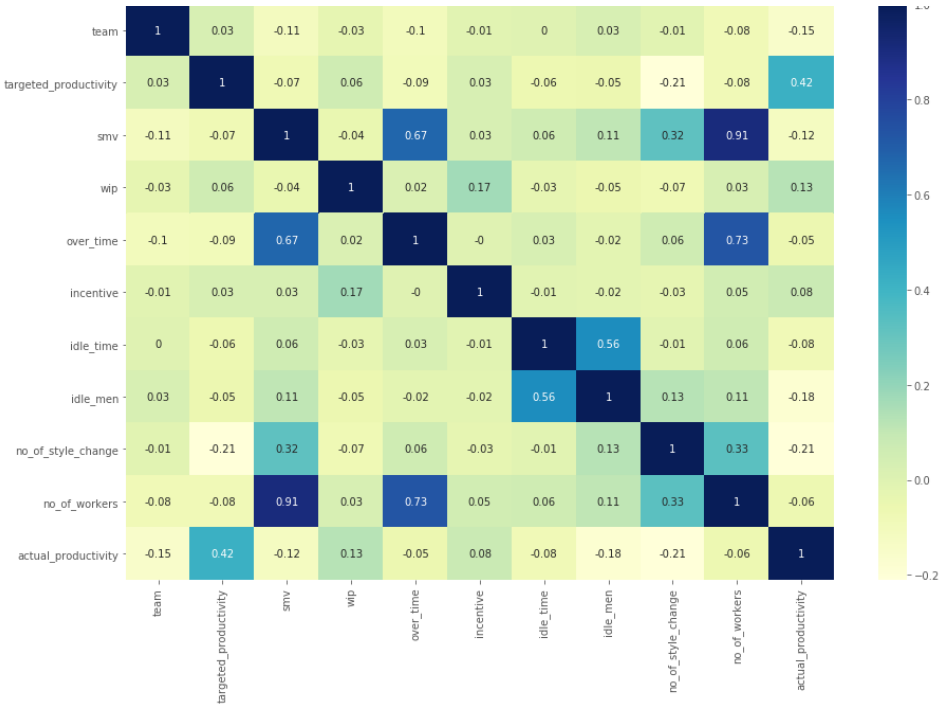


Figura 5 Matriz de correlación por pares de variables.

En la matriz de correlación se observa que existe una correlación significa entre los pares de variables siguientes:

Svm --- no_of_workers 0.91%

over_time ---- no_of_workers 0.73%

Descripción de las variables categóricas

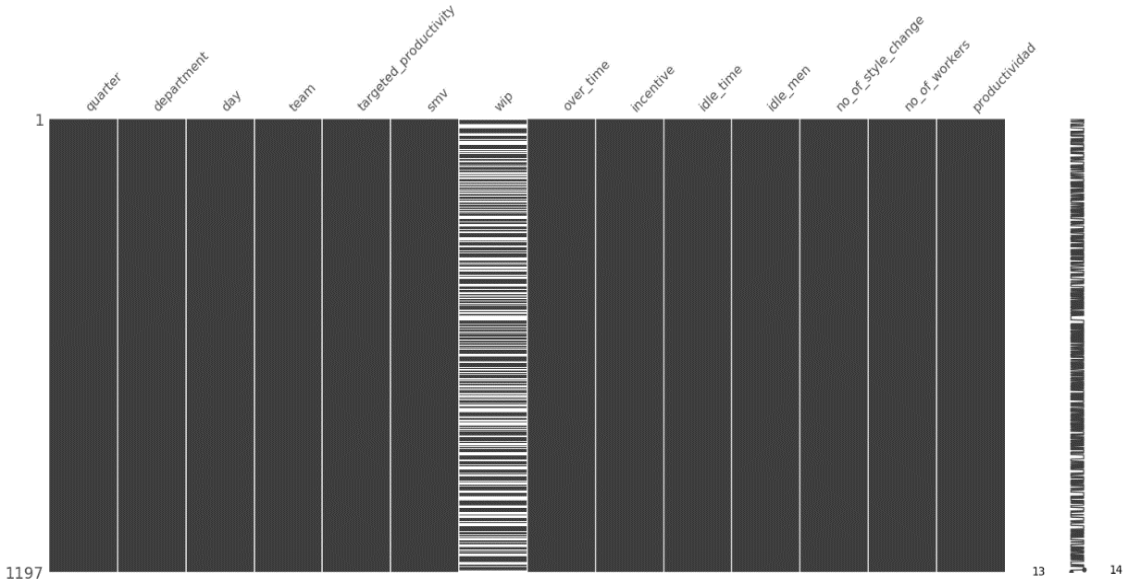
En la tabla 1 se observa la distribución de los datos almacenados en las variables categóricas de la base de datos.

Tabla 1 Distribución de variables categóricas

	quarter	department	day
count	1197	1197	1197
unique	5	2	6
top	Quarter1	sweing	Wednesday
freq	360	691	208

Valores nulos o perdidos

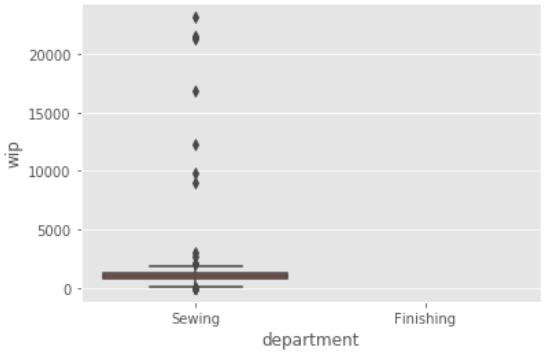
La variable wip contiene 501 valores perdidos, donde se puede observar en la gráfica 3 como se representan en el conjunto de datos.



Gráfica 3 Representación de variables con valores nan.

La variable wip contiene los valores nan agrupados en el departamento de terminado (Finishing), como se muestra en la gráfica 4. Esto es debido a que en el departamento de

terminado no se contabilizan los productos que se deben fabricar, sino que los confeccionados pasan para realizarle las terminaciones.



Gráfica 4 Representación de variable wip por departamento.

Luego de realizar el análisis exploratorio de los datos se procede a la etapa de preprocesamiento de estos para realizar la transformación de las variables, en cuanto a normalización, cambios de scala y eliminación de valores perdidos.

6.3 Preprocesamiento de datos

El proceso de preprocesamiento de los datos es uno de los más importantes antes de ajustar los datos a un modelo de aprendizaje automático, su objetivo principal es preparar el conjunto de observaciones tal que se obtenga un conjunto final de calidad para la fase de extracción del conocimiento [27].

Transformación de variables.

El siguiente paso es transformar las variables categóricas en variables de tipo numéricas y eliminar la característica date, porque las variables quarter y day ya contienen la información significativa respecto la fecha.

Se convierten los valores de las variables categóricas day, quarter y department en valores numéricos con la utilización de la técnica LabelEncoder. La misma codifica las clases de una variable categórica en valores numéricos entre 0 y el número de clases menos 1. Con el método **fit_transform** se realiza simultáneamente el entrenamiento de los datos y la transformación de las etiquetas que se incluyen como argumento en los números correspondientes.

Valores nulos.

Los valores nulos de la variable wip pertenecen al departamento de acabado, el cual recibe el trabajo que ya ha sido contabilizado para producir, pero necesita las terminaciones. Por lo tanto, estos valores nan son remplazados el valor cero.

Variable de salida o respuesta.

La variable inicial de salida es actual_productivity, la cual sigue una distribución continua. En la presente investigación se desea realizar una clasificación binaria para predecir si la productividad será cumplida o no. Para ello se crea una nueva variable productividad con valores de 0 y 1, donde los algoritmos de aprendizaje automático aprenderán para distinguir si se cumple o no. Las clases de etiqueta 0 representan la productividad que no fue cumplida y las que tienen la etiqueta 1 son las que si se ha cumplido.

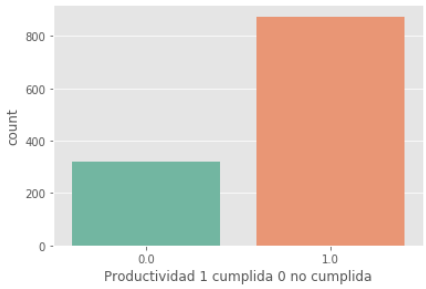
1. La resta entre la variable actual_productividad y la productividad_objetivo, le fue asignada a una variable resto.

2. Luego se creó una nueva variable denominada productividad, donde se asignó el valor de 0 en la posición de la instancia que la variable resto fuera menor que 0, y 1 donde el valor de la instancia de la variable resto fuera mayor o igual a 0.

```
textil_dat['resto']=textil_dat.actual_productivity-textil_dat.targeted_productivity
textil_dat.columns
textil_dat['productividad']=np.nan
textil_dat.loc[textil_dat['resto']<0,'productividad'] = 0
textil_dat.loc[textil_dat['resto']>=0, 'productividad'] = 1
```

Figura 6 Código de creación de variable de salida: productividad

Luego se realiza una descripción de la nueva variable productividad para explorar como están distribuidos los datos. Donde se puede observar en la gráfica 5 que el número de observaciones que cumplieron y no cumplieron con la producción, no está balanceado, significando así que el porcentaje de las clases que representan la producción no cumplida es de un 26.9%, siendo una baja proporción respecto a los registros de la clase que si se cumplió.



Gráfica 5 Número de registros que pertenecen a cada clase de la productividad.

Después de analizar el desequilibrio en el conjunto de datos respecto a la variable productividad, se puede concluir que la mayoría de las observaciones registran que la producción se cumplió. Si se utiliza este conjunto de datos como base en nuestros modelos predictivos y análisis, se podrían obtener errores y los algoritmos se ajustarían en exceso, por lo que asumirían que la mayoría de las veces se cumple con la producción. Pero el objetivo es que el modelo detecte patrones de cuando no se cumple la productividad, para así tomar las decisiones pertinentes.

Por lo tanto, en la presente investigación se realizará una comparación entre modelos de clasificación cuando los datos están balanceados y cuando no. Para ello se aplicará la técnica SMOTE de remuestreo para generar nuevas instancias a partir de las instancias existentes y así equilibrar la clase minoritaria.

División de los datos

Antes de proceder con la ejecución de los algoritmos, se divide el conjunto de datos original en dos subconjuntos para el proceso de entrenamiento y pruebas. El primero se utiliza para realizar el entrenamiento de los modelos y supondrá el 70% de las instancias. El subconjunto de datos para las pruebas contiene el 30% de las observaciones restantes.

Mediante el uso de la técnica train_test_split de scikit-lear se realiza la división de los datos, estableciendo como parámetro el stratify, lo que proporciona de forma equivalente los valores de la variable respuesta en el conjunto de entrenamiento y en el de prueba. Además, se generan los dos subconjuntos seleccionando las instancias de forma aleatoria.

Normalización (estandarización)

El siguiente paso es normalizar las variables independientes de los subconjuntos de datos divididos anteriormente. Para este caso en particular se utiliza la Normalización Z-score usando la librería StandardScaler, para escalar los predictores. Esta transformación de scala de datos radica en la división de cada predictor entre su desviación típica después de haber obtenido su media.

$$Z = \frac{x - \mu}{\sigma}$$

Luego de realizar el preprocesamiento de las variables se obtiene un total de 1197 muestras y 14 variables, las cuales formaran el conjunto de datos para diseñar los modelos de clasificación.

6.4 Entrenamiento y validación de los algoritmos clasificadores para los datos no equilibrados.

En este apartado se procede a diseñar y validar el entrenamiento de los algoritmos de clasificación una vez se ha concluido con la limpieza y transformación de los datos. Después se realiza la comparación entre ellos para determinar cuál es el más efectivo detectando las producciones no cumplidas, para luego evaluarlos en el conjunto de prueba.

En primer lugar, se toma el conjunto de datos seleccionado para el entrenamiento y se procede con la fase de ajustar los modelos. Luego se realiza la validación cruzada [23] con una división de 5 particiones equivalentes, donde 4 subconjuntos de datos se utilizan como conjunto de entrenamiento y 1 para validar el modelo. Este proceso se repite 5 veces donde el conjunto de validación es diferente cada vez. Durante este procedimiento se va obteniendo la puntuación promedio del accuracy de los valores intermedios en cada iteración, mostrando el desempeño del modelo de clasificación sobre el conjunto de datos de entrenamiento.

Las técnicas de clasificación seleccionadas son los siguientes.

1. k Vecinos más cercanos (k Nearest Neighbors KNN)
2. Regresión logística

3. Support-vector Machine (SVMs)
4. Bosques aleatorios (Random Forest)
5. Naïve-Bayes (NBC)

Una vez ejecutadas las técnicas mencionadas, se muestran en la tabla 2 las puntuaciones obtenidas por cada uno de los modelos entrenados y validados.

Tabla 2 Puntuación de validación cruzada, datos no balanceados.

Puntuación de ajuste y validación	
LogisiticRegression	73.0 %
KNeighborsClassifier	75.0 %
Support Vector Classifier	75.0 %
RandomForestClassifier	81.0 %
GaussianNB	74.0 %

Luego de observar los valores ofrecidos después del entrenamiento y validación de los modelos, se puede decir que el RandomForest con un 81% es el que mejor puntuación de exactitud de clasificación ha obtenido sobre el conjunto de entrenamiento. Siendo el algoritmo LogisiticRegression con 73% el que menos puntuación de exactitud ha tenido para clasificar las clases.

Sin embargo, a pesar de obtener buenos resultados en la validación anterior, se desea mejorar el rendimiento del modelo. Este procedimiento se conoce como tunning, donde se aplica el método GridSearchCV [28], el cual realiza una búsqueda del mejor parámetro contenido dentro del conjunto de hiperparámetros definidos en cada algoritmo. Una vez, se entrenan los modelos con el método fit en función de las combinaciones de los hiperparámetros definidos en cada clasificador, se realiza la validación cruzada estratificada dividiendo en 5 particiones, evaluando el modelo con los parámetros óptimos seleccionados en la búsqueda anterior. Donde finalmente, se obtiene la puntuación promedio de la validación cruzada, siendo este un valor más real y ajustado a la precisión de cada uno de los modelos desarrollados.

En la tabla 3 se muestran las puntuaciones obtenidas luego de realizar el tunning y validación de los modelos de clasificación.

Tabla 3 Puntuación de validación cruzada, mejores hiperparámetros

Evaluación por validación cruzada	
LogisiticRegression	73.71%
KNeighborsClassifier	75.39%
Support Vector Classifier	75.27%
RandomForestClassifier	77.9%
GaussianNB	74.31%

Después de realizar el entrenamiento y la validación cruzada, se observa que el algoritmo RandomForestClassifier obtiene una puntuación de 77.9%, la cual disminuyo con respecto a la ofrecida anteriormente sin realizar el tunning. Pero sigue siendo el modelo que mejor puntuación obtiene, con respecto a los demás.

Los hiperparámetros óptimos encontrados para este método fueron los siguientes: criterio de división del árbol utilizado Gini, el máximo de características para crear los árboles independientes fue de 9, y la creación de los árboles fue de hasta el nivel de profundidad 3. En la figura 7 se muestran los mejores hiperparámetros encontrados por cada clasificador.

```
Logistic Regression mejor estimador:  penalty: l2 valor C: 0.01
KNears mejor estimador:  n_neighbors: 3 algorithm: auto
SVC mejor estimador:  C: 1 kernel: rbf
Random forest mejor estimador:  n_estimators: 150 max_features: 9 criterion: gini max_depth: 3
Naive_bayes mejor estimador :  var_smoothing: 0
```

Figura 7 Mejores hiperparámetros

Por lo tanto, el método de Random Forest muestra indicios de ser la técnica que mejor va a clasificar en cuanto a precisión, los casos en productividad cumplida o no del conjunto de pruebas.

6.5 Entrenamiento y validación de los algoritmos clasificadores aplicando la técnica SMOTE.

Luego de realizar el experimento anterior con los datos no balanceados, se procede a realizar el mismo experimento, pero con las clases del conjunto de entrenamiento balanceadas con el interés de que el aprendizaje sea lo más homogéneo posible. En un

principio, el grupo de líneas de producción que cumplieron con la productividad está compuesto por 875 registros, y 322 registros que no cumplieron, representando el 26.9% del total.

Después que se realiza la división del conjunto de datos en subconjuntos de entrenamiento y prueba, se aplica la técnica SMOTE al conjunto de entrenamiento para la corrección del equilibrio de las instancias. Obteniendo así el mismo número de productividad cumplida y no cumplida como se muestra en la figura 7.

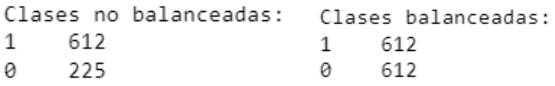


Figura 8 Aplicación de técnica SMOTE (Antes y después)

Una vez se ha corregido el desbalanceo de las clases, se entrenan y validan los 5 algoritmos clasificadores anteriores. Se observan para comprobar cuál es el más exacto durante el entrenamiento, detectando las producciones no cumplidas. Una vez entrenados los algoritmos con las clases balanceadas se muestran las puntuaciones del entrenamiento en la tabla 4.

Tabla 4 Puntuación de validación cruzada, datos balanceados

	Puntuación de ajuste y validación (Accuracy)
LogisiticRegression	67.0 %
KNeighborsClassifier	80.0 %
Support Vector Classifier	78.0 %
RandomForestClassifier	88.0 %
GaussianNB	60.0 %

Los valores ofrecidos después del entrenamiento y validación de los modelos destacan que la técnica de RandomForest es el que mejor puntuación obtiene con un 88% de accuracy. Pero en este caso el algoritmo KNeighborsClassifier le sigue con un 80%, mejorando con respecto al entrenamiento realizado en el epígrafe 6.4. Siendo estos métodos los que mejores resultados obtienen en cuanto a exactitud durante la

validación, se puede ir suponiendo que serán los que mejor clasificación de las clases realicen.

En este apartado se sigue el mismo procedimiento de intentar mejorar el rendimiento de clasificación de los modelos con la utilización del método GridSearchCV [28], donde se definen los mismos hiperparámetros para seleccionar el más optimo, que en el experimento presentado en el epígrafe 6.4. Después de ejecutar la validación cruzada, se obtienen las puntuaciones que se exponen en la tabla 5.

Tabla 5 Puntuación de validación cruzada, mejores hiperparámetros

	Puntuación de validación cruzada
LogisiticRegression	67.08%
KNeighborsClassifier	83.5%
Support Vector Classifier	78.03%
RandomForestClassifier	76.88%
GaussianNB	65.2%

Una vez entrenados y validados los modelos, se observa que el algoritmo KNeighbors obtiene una puntuación de accuracy de un 83.5% lo cual aumenta en comparación al entrenamiento anterior. Por otra parte, el modelo RandomForest disminuye a un 76.88%. Los hiperparámetros óptimos encontrados para estos métodos en cuanto a precisión fueron los que se muestran en la figura 9:

```
Logistic Regression mejor estimador:  penalty: l2 valor C: 1000
KNears mejor estimador:  n_neighbors: 3 algorithm: brute
SVC mejor estimador:  C: 1 kernel: rbf
Random forest mejor estimador:  n_estimators: 100 max_features: 5 criterion: gini max_depth: 3
Naive_bayes mejor estimador :  var_smoothing: 0.3
```

Figura 9 Mejores hiperparámetros

En este caso, el método de KNeighbors ajustado con el conjunto de entrenamiento balanceado es el que muestra indicios de ser la técnica que mejor realizara la clasificación durante la predicción.

7 Experimentación y análisis de resultados

7.1 Test de los algoritmos de clasificación

Una vez se ha realizado el entrenamiento y validación de las 5 técnicas de clasificación sobre el conjunto de datos de entrenamiento, para los casos cuando las muestras están balanceadas y cuando no están balanceadas, se procede a realizar una evaluación de los modelos sobre el conjunto de datos de pruebas. Determinando, mediante una comparación de los resultados obtenidos, cual es el clasificador más efectivo para detectar cuando no será cumplida la productividad. Para realizar esta comparación primero se mostrarán los valores obtenidos en las matrices de confusión de

cada uno de los algoritmos en los dos casos.

Luego se tendrán en cuenta para tomar las decisiones de elegir el mejor clasificador, las métricas de precisión y sensibilidad. Finalmente, se realizará el grafico de la curva ROC que representa la sensibilidad frente a la 1-especificidad. Explicando, donde coincide el punto de corte cuando el método de clasificación alcanza la mejor sensibilidad y tasa de verdaderos negativos durante la predicción, mostrando así el mayor valor AUC que se obtiene para discriminar entre productividades cumplidas y no cumplidas.

Matriz de confusión

La matriz de confusión será representada de forma binaria donde las etiquetas de las clases de 0 y 1 serán identificados como:

- Productividad No cumplida = 0.
- Productividad Cumplida = 1.

		Matriz de confusión	
		Predicción	
		0	**1**
Realidad	**0**	**VN**	**FP**
	1	**FN**	**VP**

Los valores reales serán representados en las filas, mientras que los valores obtenidos en la predicción se representan en las columnas.

- Verdadero Negativo (VN): número de clasificaciones correctas de la clase No Productividad.
- Falso Negativo (FN): número de clasificaciones incorrectas de la clase No Productividad.
- Verdadero Positivo (VP): número de clasificaciones correctas de la clase Productividad.
- Falso Positivo (FP): número de clasificaciones incorrectas de la clase Productividad.

Evaluación: Modelos entrenados con conjunto de datos no balanceados

En la figura 10 se muestra la cantidad de muestras de la clase de salida productividad que utilizaron para las pruebas. Donde se observa que 263 observaciones pertenecen a la clase 1 y 97 a la clase 0.

```
1  y_test.value_counts()

1.0    263
0.0     97
```

Figura 10 Conjunto de datos de prueba

Luego de probar los clasificadores en el conjunto de prueba se obtienen los resultados de la evaluación en las matrices de confusión que se presentan en la tabla 6.

Tabla 6 Matriz confusión de algoritmos de clasificación con datos no balanceados

Logisitic Regression		**KNeighbors**		**Random Forest**		**SVC**		**GaussianNB**	
VN=2	**FP=95**	VN=43	**FP=54**	VN=38	**FP=59**	VN=27	**FP=70**	VN=5	**FP=92**
FN=1	VP=262	FN=35	VP=232	FN=15	VP=252	FN=24	VP=242	FN=7	VP=262

Después de observar las matrices de confusión de la tabla 6, se puede mencionar que los modelos de clasificación han asumido mayor tendencia a clasificar las muestras de Productividad No cumplida como Cumplidas generando así los Falsos positivos (FP),

por lo tanto, presupone un problema para identificar cuando la productividad de la línea de producción no será cumplida y tomar medidas sobre ello. Esto se debe a que las clases están desbalanceadas, y los clasificadores durante el proceso de aprendizaje realizaron una generalización o sobreajuste de la información donde se ha perjudicado la clase minoritaria.

En la figura 11 se muestran los reportes de las matrices de confusión obtenidas, donde se prestará mayor atención a las métricas de precisión, recall y f1-score.

Logistic Regression:

	precision	recall	f1-score	support
0.0	0.67	0.02	0.04	97
1.0	0.73	1.00	0.85	263
accuracy			0.73	360
macro avg	0.70	0.51	0.44	360
weighted avg	0.72	0.73	0.63	360

KNears Neighbors:

	precision	recall	f1-score	support
0.0	0.55	0.44	0.49	97
1.0	0.81	0.87	0.84	263
accuracy			0.75	360
macro avg	0.68	0.66	0.66	360
weighted avg	0.74	0.75	0.74	360

Support Vector Classifier:

	precision	recall	f1-score	support
0.0	0.53	0.28	0.36	97
1.0	0.77	0.91	0.84	263
accuracy			0.74	360
macro avg	0.65	0.59	0.60	360
weighted avg	0.71	0.74	0.71	360

Random Fores:

	precision	recall	f1-score	support
0.0	0.72	0.39	0.51	97
1.0	0.81	0.94	0.87	263
accuracy			0.79	360
macro avg	0.76	0.67	0.69	360
weighted avg	0.78	0.79	0.77	360

GaussianNB:

	precision	recall	f1-score	support
0.0	0.42	0.05	0.09	97
1.0	0.74	0.97	0.84	263
accuracy			0.73	360
macro avg	0.58	0.51	0.46	360
weighted avg	0.65	0.72	0.64	360

Figura 11 Reportes de las matrices de confusión

Luego de observar las métricas de evaluación de los modelos entrenados con el conjunto de muestras desbalanceadas, se puede corroborar que el recall de los clasificadores en general no es bueno. Siendo el algoritmo de Regresión Logística el que peor recall ofrece para detectar los casos de productividad no cumplida. Esto es debido a la baja representatividad de la clase negativa en la muestra que se utilizó para el entrenamiento de los modelos de clasificación. Por lo tanto, la capacidad para detectarlos en el conjunto de pruebas igualmente es baja. Sin embargo, el modelo de clasificación Random Forest ha presentado un recall de 0.39, una precisión 0.72 y f1-score de 0.51, detectando así las muestras de la clase minoritaria, con mejor sensibilidad y precisión.

Evaluación: Modelos entrenados con conjunto de datos balanceados

Después de evaluar los modelos entrenados con las muestras de las clases no balanceadas, se procede a realizar el mismo procedimiento de evaluación en el conjunto de muestras seleccionado para las pruebas, pero siendo los modelos que fueron ajustados con el conjunto de entrenamiento de las muestras con las clases balanceadas.

En la figura 12 se muestra la cantidad de muestras de la clase de salida productividad que se utilizaron para las pruebas. Donde se observa que 263 observaciones pertenecen a la clase 1 y 97 a la clase 0, aclarando que como es el conjunto de pruebas no se aplica la técnica del sobremuestreo de datos para no alterar los valores reales.

```
1  y_test.value_counts()
```

```
1.0    263
0.0     97
```

Figura 12 Conjunto de datos de prueba.

Matriz de confusión

Luego de probar los clasificadores en el conjunto de prueba se obtienen los resultados de la evaluación en las matrices de confusión que se presentan en la tabla 7.

Tabla 7 Matriz confusión de algoritmos de clasificación con datos balanceados

Logisitic Regression	KNeighbors	Random Forest	SVC	GaussianNB

VN=67	FP=30	**VN=60**	FP=37	**VN=74**	FP=23	**VN=69**	FP=28	**VN=38**	FP=59
FN=90	TP=172	FN=69	VP=192	FN=70	VP=192	FN=71	VP=192	FN=41	VP=222

Después de analizar las matrices de confusión de la tabla 7, se puede decir que los modelos de clasificación han mejorado en la clasificación de las muestras de Productividad No cumplida generando así mayor número de verdaderos negativos con respecto a los Falsos positivos (FP), por lo tanto, presupone una mejoría con respecto a la evaluación del caso anterior. Esto se debe a que las clases están balanceadas, y los clasificadores durante el proceso de aprendizaje no realizaron una generalización de la información.

En la figura 13 se muestran los reportes de las matrices de confusión obtenidas, donde se prestará mayor atención a las métricas de precisión, recall y f1-score.

```
Logistic Regression:
              precision    recall  f1-score   support

         0.0       0.43      0.69      0.53        97
         1.0       0.85      0.66      0.74       263

    accuracy                           0.67       360
   macro avg       0.64      0.67      0.64       360
weighted avg       0.74      0.67      0.68       360


KNears Neighbors:
              precision    recall  f1-score   support

         0.0       0.47      0.62      0.53        97
         1.0       0.84      0.74      0.79       263

    accuracy                           0.71       360
   macro avg       0.65      0.68      0.66       360
weighted avg       0.74      0.71      0.72       360


Support Vector Classifier:
              precision    recall  f1-score   support

         0.0       0.49      0.71      0.58        97
         1.0       0.87      0.73      0.80       263

    accuracy                           0.73       360
   macro avg       0.68      0.72      0.69       360
weighted avg       0.77      0.72      0.74       360
```

```
Random Forest:
              precision    recall  f1-score   support

         0.0       0.51      0.76      0.61        97
         1.0       0.89      0.73      0.81       263

    accuracy                           0.74       360
   macro avg       0.70      0.75      0.71       360
weighted avg       0.79      0.74      0.75       360


GaussianNB:
              precision    recall  f1-score   support

         0.0       0.48      0.39      0.43        97
         1.0       0.79      0.84      0.82       263

    accuracy                           0.72       360
   macro avg       0.64      0.62      0.62       360
weighted avg       0.71      0.72      0.71       360
```

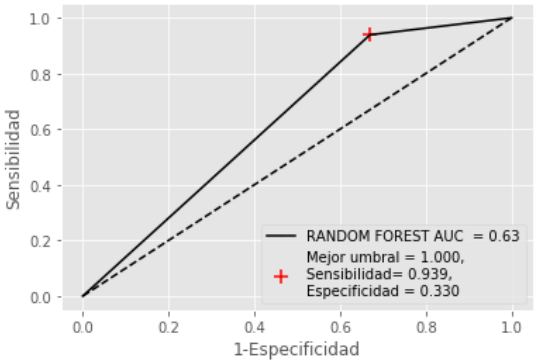
Figura 13 Reportes de las matrices de confusión

Después de observar la figura 13 donde se expone la evaluación de cada matriz de confusión obtenida, se puede concluir que el recall de los clasificadores en general mejoró considerablemente respecto al caso de evaluación anterior. Siendo la representatividad de las muestras de la clase negativa homogénea con respecto a las clases positivas, implicando que la capacidad de los clasificadores para detectarlas en el conjunto de pruebas fuera superior. Siendo el modelo de clasificación Random Forest el que ofrece los mejores resultados en cuanto a la detección de los verdaderos negativos. Donde obtuvo un recall de 0.76 para detectar las muestras negativas, una precisión de 0.51 y un f1-score mejorado con una puntuación de 0.61.

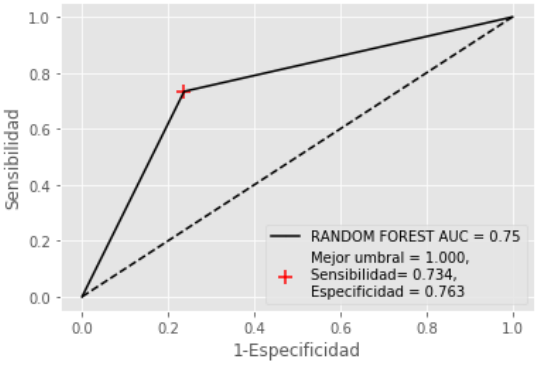
Finalmente, después de analizar los reportes de las métricas de evaluación de las matrices de confusión anteriores, se decide que el algoritmo más eficiente para detectar los casos cuando la productividad no es cumplida es el Random Forests. Realizando predicciones con mayor recall y precisión sobre las muestras de la clase Productividad cuando el conjunto de datos esta equilibrado.

Curva ROC

Después de seleccionar el modelo Random Forest como mejor clasificador, se procede a graficar la curva ROC para los dos casos que se han probado anteriormente en el conjunto de pruebas. La gráfica 6 representa el modelo ajustado a las muestras no balanceadas, y la gráfica 7 representa el modelo ajustado a las muestras balanceadas.



Gráfica 6 Curva ROC-AUC conjunto de datos no balanceados.



Gráfica 7 Curva ROC-AUC conjunto de datos balanceados.

Tabla 8 Comparación de métricas Curva ROC

	Sensibilidad	***Especificidad o Tasa de verdaderos negativos***	***1-Especificidad o Tasa de falsos positivos***	***AUC***
Conjunto no balanceado *Random Forest*	**0.93**	**0.33**	**0.67**	**0.63**
Conjunto balanceado *Random Forest*	**0.73**	**0.76**	**0.24**	**0.75**

Lugo de analizar las métricas, se comparan los puntos de corte en las curvas ROC cuando se obtiene la mayor sensibilidad y menor tasa de falos positivos. Reafirmando que el método Random Forest, predice con mejor desempeño las muestras pertenecientes a la clase negativa, cuando realiza el proceso de aprendizaje sobre un conjunto de datos balanceados.

Por otra parte, el modelo de clasificación muestra una mejoría en cuanto a la métrica AUC con un 75%, cuyo valor se obtiene en el intervalo del punto de corte de mayor detección de tasa de verdaderos positivos y la tasa de verdaderos negativos [0.73 – 0.76], donde toma el umbral óptimo con valor = 1. Presentando así una mayor capacidad de discriminación en cuanto a la precisión para identificar los casos cuando la productividad no es cumplida.

8 Conclusiones

El desarrollo de este trabajo se ejecutó sobre una base de datos que contiene información de una fábrica de producción textil de Bangladesh. Su uso permitió contar con datos fiables para la aplicación de un conjunto de técnicas de aprendizaje supervisado para detectar cuando la productividad no será cumplida. Las técnicas de clasificación utilizadas para desarrollar estos modelos de predicción fueron: Regresión Logística, K-Vecinos más cercanos, Máquina de vector soporte, Random Forests y Naive Bayes Gaussiano.

Las muestras de la base de datos no están equilibradas respecto a la variable respuesta, por lo tanto, se realizaron dos experimentos durante el entrenamiento de los modelos, para comprobar como el equilibrio en los datos mejora los resultados de predicción.

Experimento 1: Ajustar los modelos de aprendizaje con los hiperparámetros óptimos y validar los mismos con la técnica de validación cruzada con 5 particiones, para obtener la puntuación de exactitud promedio que son capaces de predecir. Este procedimiento fue realizado sobre el conjunto de datos para las muestras desbalanceadas. Donde el modelo clasificador Random Forest ofreció un accuracy de 77.9%, siendo la mayor puntuación de validación del aprendizaje para el conjunto de datos no equilibrados respecto a los demás clasificadores.

Experimento 2: Ajustar los modelos de aprendizaje con los hiperparámetros óptimos y validar con la técnica de validación cruzada con 5 particiones los modelos de aprendizaje, obteniendo la puntuación de exactitud promedio que son capaces de predecir. Aplicando la técnica de sobremuestreo SMOTE, para balancear las muestras

con respecto a la clase mayoritaria. Donde el modelo clasificador KNeighbors ofreció un accuracy de 83.5%, siendo la mayor puntuación de validación del aprendizaje para el conjunto de datos no equilibrados respecto a los demás clasificadores.

Por último, se realizó una comparación de los resultados obtenidos de las pruebas de evaluación de los modelos de clasificación desarrollados en el experimento 1 y 2. Donde se pueden realizar las siguientes conclusiones:

1. Las métricas de evaluación obtenidas del reporte de la matriz de confusión del experimento 1, mostraron que el recall o exhaustividad de los clasificadores en general no es buena cuando las muestras no están balanceadas. Por lo tanto, se demostró una deficiente capacidad para detectar las clases pertenecientes al grupo minoritario. Sin embargo, el modelo de clasificación Random Forest presentó un valor de f1-score entre recall y precisión de 0.51, detectando así la cantidad de muestras de la clase minoritaria con mejor precisión que el resto de los modelos.
2. Las métricas de evaluación obtenidas del reporte de la matriz de confusión del experimento 2, mostraron que la sensibilidad de los clasificadores en general mejoró considerablemente respecto al experimento 1, demostrando eficiencia en cuanto a predicción de la clase negativa cuando el conjunto de datos esta equilibrado. Siendo el modelo de clasificación Random Forest el mejor clasificador, respecto a las métricas Recall con un 0.76, precisión de 0.51 y un f1-score de 0.61.

Finalmente se puede concluir después de realizar los experimentos para clasificar las muestras que el mejor clasificador en cuanto a recall y precisión es el modelo Random Forest. Demostrando que realizar el entrenamiento con un conjunto de datos balanceados implica que los modelos aprendan de una manera más eficiente, lo cual presupone una predicción más acertada maximizando la sensibilidad y especificidad de estos. Mediante el análisis de la Curva ROC se observa el rendimiento mejorado que ofrece en cuanto a especificidad obteniendo así un valor AUC del 75%, lo cual se considera un modelo de clasificación aceptable para discriminar entre las clases que cumplieron y no cumplieron la productividad.

Referencias

[1] C. e. y. s. d. España, «INFORME SOBRE LA INDUSTRIA EN ESPAÑA: PROPUESTAS PARA SU DESARROLLO,» Sesión ordinaria del Pleno de 18 de diciembre de 2019, 2019.

[2] EY, «Informe sector moda en España. Análisis del impacto de la crisis del Covid-19,» 2020.

[3] C. d. Industria, «Análisis de la industria manufacturera española. Evolucion sectorial,» Gabinete secretaría general, Madrid, 2019.

[4] Seebo, «Machine Learning and AI in manufacturing. Obtenido de,» Seebo.com, 2019.

[5] D. Hand, «Principles of data mining. Drug Saf.,» 2007.

[6] I. P. a. E. P. Herna L. Viktor, «Who are our clients: consumer segmentation through explorative data mining,» *International Journal of Data Mining, Modelling and Management,* 2012.

[7] R. C. H. C. Joel Miguel Gutierrez Espiritu, «Una metodología para la estimación automática de tallas de polos masculinos empleando principios antropométricos mediante algoritmos de Machine Learning,» 2021.

[8] D. A. GARCÉS y O. D. CASTRILLÓN, «Diseño de una Técnica Inteligente para Identificar y Reducir los Tiempos Muertos en un Sistema de Producción.,» *Información tecnológica,* vol. 28, nº 3, pp. 157-170, 2017.

[9] N. E. A. &. R. J. A. O. Gaviria, « Análisis comparativo de descriptores para la clasificación de telas utilizando imágenes.,» *Universidad Tecnológica de Pereira. Facultad de Ingenierías Eléctrica, Electrónica, Física, y Ciencias de la Computación. Ingeniería Electrónica.,* 2016.

[10] R. L. L.-E. M. Á. M. P. &. M. M. Blanco, «Inteligencia Artificial y Machine Learning en trastornos del movimiento.,» MANUAL SEN DE NUEVAS TECNOLOGÍAS EN TRASTORNOS DEL MOVIMIENTO, 2021.

[11] A. Kaplan y M. Haenlein, «Siri, Siri in my Hand, who's the Fairest in the Land? On the Interpretations, Illustrations and Implications of Artificial Intelligence,» vol. 62, nº 1, 2019.

[12] A. Géron, «Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems.,» *O'Reilly Media.,* 2019.

[13] R. F. &. P. M. A. Mello, «Machine learning: a practical approach on the statistical learning theory,» *Springer,* 2018.

[14] E. H. K. C, «“ONLINE SOCIAL NETWORK ANALYSIS USING MACHINE,» vol. 4, nº 4, pp. 25-40, 2017.

[15] E. S. &. M. M. S. A. Barrientos Mogollon, «Modelos de aprendizaje supervisado como apoyo a la toma de decisiones en las organizaciones basados en datos de redes sociales: Una revisión sistemática de la literatura.,» 2019.

[16] Á. S. Maganto, «ESTUDIO DE TÉCNICAS SUPERVISADAS DE REDUCCIÓN DE DIMENSIONALIDAD PARA PROBLEMAS DE CLASIFICACIÓN,» 2017.

[17] H. Chitarroni, «La regresión logística.,» 2002.

[18] R. I. Flores de la Torre, «Análisis Comparativo de Árboles de Decisión y Máquina de Vectores Soporte para conjuntos de datos de Diabetes y Hepatitis.,» pp. 43-49, 2014.

[19] J. K. R. A. N. y. M. I. Ali, «Random forests and decision trees. IJCSI International,» *Journal of Computer Science Issues,* vol. 9, nº 5, pp. 272-278, 2012.

[20] J. G.-G. D. R.-G. S. G. S. &. H. F. Luengo, «Big data preprocessing: enabling smart data.,» *Springer Nature,* 2020.

[21] N. V. B. K. W. H. L. O. &. K. W. P. Chawla, «SMOTE: synthetic minority over-sampling technique.,» *Journal of artificial intelligence research,* vol. 16, pp. 321-357., 2002.

[22] T. IVAN, «Two modifications of CNN. IEEE transactions on Systems, Man and Communications,» *SMC,* vol. 6, pp. 769-772, 1976.

[23] L. T. H. L. P. Refaeilzadeh, «"Cross-Validation" En Encyclopedia of Database Systems,» *Springer,* 2019.

[24] J. &. C. L. Cerda, «Uso de curvas ROC en investigación clínica: Aspectos teórico-prácticos.,» *Revista chilena de infectología,* vol. 29, nº 2, pp. 138-141., 2012.

[25] T. FAWCETT, «Introduction to receiver operator curves.,» *Pattern Recognit. Lett,* vol. 27, pp. 861-874, 2006.

[26] A. A. Imran, «Data Set Productivity Prediction of Garment Employees,» http://archive.ics.uci.edu/ml/datasets/Productivity+Prediction+of+Garment+Employees, 2020.

[27] S. R.-G. J. L. Salvador García, «Big Data: Preprocesamiento,» *Departamento de Ciencias de la Computación e Inteligencia Artificial, Universidad de Granada,* pp. 17-23, 2016.

[28] F. V. G. G. A. M. V. T. B. G. O. .. &. D. E. Pedregosa, «Scikit-learn: Machine learning in Python.,» *the Journal of machine Learning research,* vol. 12, pp. 2825-2830., 2011.

[29] A. d. D. E. d. P. d. Asturias, «sector-textil-el-sector-en-espana-informacion-general,» 2021.

[30] Nelson Labarca, «Consideraciones teóricas de la competitividad empresarial,» vol. 13, nº 2, pp. 158-184, 2007.

[31] J. L. F. H. D. S. García, «Preprocessing in Data Mining.,» *Springer,* 2015.

[32] 1. scikit learn, «sklearn.preprocessing.LabelEncoder,» *scikit learn.*

[33] F. V. G. G. A. M. V. T. B. G. O. .. &. D. E. Pedregosa, «Scikit-learn: Machine learning in Python.,» *the Journal of machine Learning research,,* vol. 12, pp. 2825-2830., 2011.