

UCUENCA

Universidad de Cuenca

Facultad de Ciencias Químicas

Carrera de Ingeniería Industrial

Identificación de defectos en textiles utilizando algoritmos de inteligencia artificial


Trabajo de titulación previo a la obtención del título de Ingeniero Industrial

Autor:

Josue David Arce Castro

Director:

Juan Carlos Llivisaca Villazhañay

ORCID:  0000-0003-2154-3277

Cuenca, Ecuador

2025-03-20

Resumen

La detección oportuna y precisa de defectos en productos textiles es crucial para garantizar la calidad, reducir costos y satisfacer las demandas de los consumidores. Sin embargo, los métodos tradicionales de inspección visual presentan limitaciones en cuanto a consistencia, escalabilidad y objetividad. Esta investigación evaluó el potencial de varios algoritmos de detección de anomalías no supervisados basados en técnicas de inteligencia artificial para mejorar la identificación de defectos en textiles. Se implementaron y compararon los desempeños de Isolation Forest, Local Outlier Factor (LOF), Elliptic Envelope, One-Class SVM, DBSCAN y K-Means. Los resultados mostraron que el modelo de Isolation Forest se destacó por obtener los mejores valores en accuracy, precisión y F1-score, indicando su capacidad para detectar defectos con pocos falsos positivos y negativos. Por otro lado, LOF tuvo la mayor sensibilidad, mientras que DBSCAN presentó una especificidad relativamente alta. La implementación de estos algoritmos de IA en entornos de producción textil podría automatizar y optimizar los procesos de inspección, reduciendo costos y mejorando la calidad de los productos. Esta investigación contribuye al campo de la inteligencia artificial aplicada a la manufactura, demostrando el potencial de los modelos de detección de anomalías para abordar desafíos específicos de la industria textil.

Palabras clave del autor: anomalías, detección anómala, isolation forest, no supervisado



El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Cuenca ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por la propiedad intelectual y los derechos de autor.

Repositorio Institucional: <https://dspace.ucuenca.edu.ec/>

Abstract

Timely and accurate detection of defects in textile products is crucial to ensure quality, reduce costs and meet consumer demands. However, traditional visual inspection methods have limitations in terms of consistency, scalability and objectivity. This research evaluated the potential of several unsupervised anomaly detection algorithms based on artificial intelligence techniques to improve textile defect identification. The performances of Isolation Forest, Local Outlier Factor (LOF), Elliptic Envelope, One-Class SVM, DBSCAN and K-Means were implemented and compared. The results showed that the Isolation Forest model stood out for obtaining the best values in accuracy, precision and F1-score, indicating its ability to detect defects with few false positives and negatives. On the other hand, LOF had the highest sensitivity, while DBSCAN presented a relatively high specificity. The implementation of these AI algorithms in textile production environments could automate and optimize inspection processes, reducing costs and improving product quality. This research contributes to the field of artificial intelligence applied to manufacturing, demonstrating the potential of anomaly detection models to address specific challenges in the textile industry.

Author keywords: anomalies, anomaly detection, isolation forest, unsupervised



The content of this work corresponds to the right of expression of the authors and does not compromise the institutional thinking of the University of Cuenca, nor does it release its responsibility before third parties. The authors assume responsibility for the intellectual property and copyrights.

Institutional Repository: <https://dspace.ucuenca.edu.ec/>

Índice de contenido

1. Introducción.....	10
1.1. Objetivo general	10
1.2. Objetivos específicos:.....	10
1.1. Hipótesis	11
2. Marco teórico.....	11
2.1. Inteligencia artificial	12
2.2. Anomalías	12
2.3. Clasificación de anomalías.....	13
2.3.1. Anomalía puntual o simple	13
2.3.2. Anomalía contextual o condicional	13
2.3.3. Anomalía colectiva	13
2.4. Técnicas de aprendizaje automático de detección de anomalías	13
2.4.1. Detección de anomalías supervisada	14
2.4.2. Detección de anomalías semi supervisada.....	14
2.4.3. Detección de anomalías no supervisadas	14
2.5. Algoritmos de detección de anomalías	14
2.6. Métricas para su evaluación y eficacia	16
2.6.1. ROC.....	16
2.6.2. AUC: Área bajo la curva ROC	16
2.6.3. Matriz de confusión	16
2.6.4. Sensibilidad (Recall).....	17
2.6.5. Especificidad (Specificity)	17
2.6.6. Precisión (Precision)	17
2.6.7. Exactitud (Accuracy)	17
2.6.8. F1.....	17
2.7. Casos de éxito.....	17
3. Metodología	18
3.1. Enfoque de investigación	18
3.2. Fase 1: Selección de variables y parámetro de entrada	19
3.3. Fase 2: Estructurar la base de datos de entrenamiento y validación:	22
3.4. Fase 3: Entrenamiento de los algoritmos de inteligencia artificial	22
3.5. Fase 4: Ajuste y validación de los algoritmos utilizados:.....	23
3.6. Fase 5: Análisis de resultados mediante indicadores:	23
4. Resultados	24

4.1.	Análisis Exploratorio de datos	24
4.2.	Selección de variables y parámetros de entrada.	30
4.3.	Estructuración de base de datos de entrenamiento y validación	31
4.4.	Parámetros utilizados para el entrenamiento.....	32
4.4.1.	Algoritmo Isolation Forest.....	32
4.4.2.	Algoritmo Elliptic Envelope	32
4.4.3.	Algoritmo LOF (Local Outlier Factor	32
4.4.4.	Algoritmo DBSCAN	33
4.4.5.	Algoritmo KMeans	33
4.5.	Comparación de métricas de desempeño	33
4.5.1.	Matriz de confusión de Isolation Forest	33
4.5.2.	Matriz de confusión de LOF	34
4.5.3.	Matriz de confusión de Elliptic Envelope	35
4.5.4.	Matriz de confusión de One-Class SVM.....	36
4.5.5.	Matriz de confusión de DBSCAN.....	37
4.5.6.	Matriz de confusión de K-means	38
4.5.7.	Análisis de métricas	38
4.5.8.	Índice de Silueta y Chi-cuadrado.....	40
5.	Discusión.....	42
6.	Conclusiones.....	43
	Referencias	44
7.	Anexos	47

Índice de figuras

Ilustración 3.1 Metodología.....	19
Ilustración 4.1 Historial de revisión de calidad.....	24
Ilustración 4.2 Variables con datos perdidos.....	26
Ilustración 4.3 Datos imputados.....	26
Ilustración 4.4 Correlación de variables.....	27
Ilustración 4.5 Matriz de dispersión.....	28
Ilustración 4.6: Análisis de la variable tipo tela y calidad.....	29
Ilustración 4.7 Calidad por tipo de tela.....	29
Ilustración 4.8 Análisis de tipos de defectos.....	30
Ilustración 4.9 Código de escalado.....	31
Ilustración 4.10 Variable respuesta.....	31
Ilustración 4.11 División de datos en entrenamiento y validación.....	31
Ilustración 4.12 Matrix de confusión del modelo Isolation Forest.....	33
Ilustración 4.13 Matriz de confusión de LOF.....	35
Ilustración 4.14 Matriz de confusión del modelo Elliptic Envelope.....	36
Ilustración 4.15 Matriz de confusión del modelo One Class SVM.....	37
Ilustración 4.16 Matriz de confusión del modelo DBSCAN.....	37
Ilustración 4.17 Matriz de confusión de K-means. Fuente: Elaboración propia.....	38
Ilustración 4.18 Métricas de desempeño.....	40
Ilustración 4.19 Métricas de modelos Fuente: Elaboración propia.....	41
Ilustración 4.20 Comparación de métricas. Fuente: Elaboración propia.....	42
Ilustración 7.1 Disposición de variables.....	47
Ilustración 7.2 Código de carga librerías.....	48
Ilustración 7.3 Depuración y selección de variables.....	48
Ilustración 7.4 División de datos en entrenamiento y validación, creación de función de almacenamiento de métricas.....	49
Ilustración 7.5 Entrenamiento de modelos parte 1.....	49
Ilustración 7.6 Entrenamiento de modelos parte 2.....	50
Ilustración 7.7 Código de los resultados de los modelos.....	50
Ilustración 7.8 Código grafico tipo radar de métricas.....	51

Índice de tablas

Tabla 2.1 Tipos de algoritmos para detección de anomalías..... 14
Tabla 4.1 Identificación de valores faltantes..... 25
Tabla 4.3 Métricas de desempeño 38
Tabla 4.4 índice de silueta y Chi cuadrado..... 40

Dedicatoria

A mi madre Cristina Castro que, con todo el amor y cariño me educó constantemente para ser siempre una persona de bien, quién con sus palabras de aliento no me dejó decaer para seguir adelante que siempre sea perseverante y cumpla con mis ideales siempre dándome su apoyo incondicional en mi nivel académico y moral. Agradecido con todo lo brindado Mamá este logro es tuyo.

A mi padre Patricio Arce cada palabra escrita está impregnada de sabiduría tu paciencia me ha impulsado a superar los diferentes desafíos y cumplir mis metas tu infinita comprensión ha sido fundamental para mantenerme enfocado y motivado en cada paso que doy.

Dedico de manera especial a mis hermanos quienes son un pilar fundamental en la construcción de mi vida profesional con valores y principios, por su apoyo en cada decisión que he tomado a lo largo de mi vida y brindarme día a día sus deseos que me motivaron constantemente para seguir adelante y alcanzar mis metas.

A mis compañeros expreso mi gratitud que en todo este trayecto compartimos experiencias y demostrando cada una de nuestras fortalezas y debilidades que a cada uno nos ayudó a crecer de manera oportuna.

Agradecimiento

Quiero expresar mi más sincero agradecimiento a todas las personas que han contribuido y apoyado en la realización de esta investigación.

En primer lugar, agradezco profundamente a mi tutor, el Ingeniero Juan Llivisaca, por su invaluable guía, su dedicación y su constante apoyo a lo largo de este proceso. Sus valiosos conocimientos, su paciencia y su retroalimentación oportuna han sido fundamentales para el desarrollo de este trabajo.

Asimismo, extendiendo mi gratitud a los docentes de la Universidad de Cuenca, quienes han compartido sus conocimientos y experiencias, enriqueciendo mi formación académica y brindándome las herramientas necesarias para llevar a cabo esta investigación.

A mis compañeros y compañeras de la Universidad de Cuenca, les agradezco por su compañerismo, sus aportes y su apoyo durante esta etapa. El intercambio de ideas y la colaboración han sido invaluable.

Finalmente, agradezco a la Universidad de Cuenca por brindarme la oportunidad de realizar esta investigación y por proporcionar los recursos y el entorno adecuado para su desarrollo.

A todos ustedes, mi más sincero agradecimiento.

1. Introducción

La detección de anomalías es una técnica utilizada para determinar la existencia de valores con un comportamiento diferente. Su importancia radica en la posibilidad de detectar cualquier característica anormal de un conjunto de datos. Diversos estudios se han realizado en diferentes áreas de la ciencia, en el ámbito textil se ha utilizado para detectar anomalías en el producto textil a partir de procesamiento de imágenes, los resultados muestran que los algoritmos reconocen los patrones anormales del producto, destacando los defectos presentes. (Ferreira et al., 2024) destaca la necesidad imperante de las organizaciones por contar con un buen sistema de control de calidad para detección oportuna de defectos, permitiendo garantizar calidad, por tal motivo emplea una comparativa de diferentes algoritmos de inteligencia artificial evaluando ventajas y desventajas, así como costos asociados a su implementación. Por otra parte, (Ferro et al., 2021) en su estudio muestra la eficacia de emplear algoritmos de detección de anomalías logrando entrenar diferentes algoritmos con base de datos de imágenes que contienen defectos. Se analiza la variación de los datos presentados identificando cambios respecto a las condiciones de forma del tejido creando patrones normales del tejido y anormales validando la efectividad del modelo. Para complementar (Tsai & Jen, 2021) destaca la detección de anomalías como una manera eficaz para diferenciar muestras normales de defectuosa entrenando algoritmos de autoencoders ajustando los parámetros de acuerdo a criterios de minimización de errores de reconstrucción resultando en una alternativa al método de autoencoders tradicional recomendando ampliamente en la detección de defectos en la industria textil. Con estos antecedentes, en el presente estudio se va a evaluar diferentes algoritmos de detección no supervisada de anomalías identificando su capacidad de detectar comportamientos anormales de un conjunto de datos resultado del control de calidad en textiles de una MiPyme textil.

1.1. Objetivo general

- Identificar defectos en producto en proceso de textiles utilizando algoritmos de detección de defectos basados en el Isolation Forest.

1.2. Objetivos específicos:

- Determinar e identificar los parámetros y variables de entrada necesarios para la construcción de los algoritmos de algoritmos de detección de anomalías.
- Generar y estructurar la base de datos de entrenamiento y validación que se va a emplear.

- Entrenar y probar los algoritmos de inteligencia artificial para predecir anomalías de los defectos en tela plana.
- Evaluar y validar los algoritmos utilizados para la predicción de defectos.
- Analizar los resultados obtenidos a partir de indicadores de desempeño

1.1. Hipótesis

La implementación de algoritmos de inteligencia artificial puede mejorar la detección de anomalías presentes en defectos de productos textiles en comparación con métodos convencionales de inspección.

2. Marco teórico

MiPymes textil

La industria textil representa hoy un aporte importante a la economía mundial representando en las organizaciones MiPymes el 90% del PIB, su aporte se ha visto disminuido por exceso de importaciones aumentando la brecha de avance tecnológico (Larios-Francia & Ferasso, 2023).

En el Ecuador el panorama de aporte ha mejorado con mayor número de empresas registradas, aumentando en un 0,5% con respecto al 2022, sin embargo, en relación de aporte de empresas manufactureras ha decrecido en un 0,5%. En la provincia del Azuay se destaca un crecimiento del 1,77% para el 2023, registrándose más de 77000 empresas (INEC, 2023).

(Ibujés Villacís & Benavides Pazmiño (2018) sugiere en su estudio contribución de la tecnología a la productividad de las pymes de la industria textil del Ecuador que esta se ha visto influenciada por el apoyo de la tecnología, mano de obra y capital, estableciendo una proporción de crecimiento moderada. Por lo que se demuestra que, al incrementar el uso de tecnología, el efecto sobre la productividad es positivo favoreciendo al desarrollo industrial.

Sin embargo, un factor importante a considerar para integrar la tecnología al sector empresarial para economías en desarrollo radica en la inversión que se debe realizar. En un estudio realizado por (Aslam et al., 2024) demuestra que la incorporación de tecnología específicamente inteligencia artificial contribuye a la optimización de los procesos productivos, reducción de desperdicios, y mejoras en la calidad del producto.

2.1. Inteligencia artificial

La inteligencia artificial se basa en la capacidad de una máquina para realizar una o múltiples tareas empleando algoritmos que utilizan datos y generando respuestas que son almacenadas para un autoaprendizaje y tomar decisiones, con la diferencia que tienen la capacidad de procesar mucha información en menor tiempo sin requerir de descanso (Rouhiainen, 2018).

Existen diversos algoritmos utilizados de los cuales destacan en algoritmos supervisado y no supervisado, la elección del mejor algoritmo va a depender de la precisión, sensibilidad y especificidad para entrenar el algoritmo (Valero et al., 2022).

El aprendizaje de los algoritmos dependerá de la finalidad del estudio, para el caso propuesto se empleará algoritmos de detección de anomalías para determinar eventos o valores atípicos de una muestra representativa de datos que resulta complejo para el ser humano (Trilles et al., 2024).

2.2. Anomalías

Una anomalía es un comportamiento desviado de una distribución ordinaria de un conjunto de datos (Nassif et al., 2021). Se distinguen las anomalías o valores atípicos ya que presentan desviaciones de un conjunto de observaciones producto de un resultado no esperado por lo que se destaca la necesidad de analizar cuál es su comportamiento (Arciba, 2023).

Los procesos, o comportamientos de un suceso que se estudia es el resultado del planteamiento de hipótesis de una observación realizada que describe un comportamiento, el cual puede ser normal o variable, existiendo así datos anormales, por tanto, las anomalías pretenden descubrir las variaciones en la normalidad de los datos observados, sin embargo, no hay una regla única para detectar las variaciones (Mehrotra et al., 2017).

Las anomalías u observaciones incorrectas tienen diferentes clasificaciones las cuales se destacan a continuación.

2.3. Clasificación de anomalías

2.3.1. Anomalía puntual o simple

Una anomalía se considera puntual cuando al menos uno de los datos se desvía del comportamiento ordinario del conjunto de datos analizado (Nassif et al., 2021). Es la forma más sencilla de anomalía y está asociada a una observación alejada de sus vecinos.

2.3.2. Anomalía contextual o condicional

Para detectar una anomalía contextual, el detector debe considerar el contexto en el que se produjo el punto de datos. Este colectivo de datos en el conjunto anómalo puede ser normales, pero juntos representan una anomalía (Arnot, 2024).

Este tipo de anomalía está presente en un conjunto de particular de un fenómeno pero que no corresponde o forma parte de otro fenómeno o contexto (Nassif et al., 2021). Existen dos tipos de atributos que definen el contexto de la anomalía, el primero se aplica para determinar el contexto de una instancia y el segundo define el atributo de comportamiento.

2.3.3. Anomalía colectiva

Se analiza una secuencia de observaciones para conocer el comportamiento selectivo del flujo de datos. Cualquier desviación del patrón normal puede dar lugar a anomalías colectivas con respecto a patrones de datos completos en intervalos de tiempo consecutivos (Arnot, 2024).

El análisis de anomalías se realiza mediante técnicas de detección las cuales permiten identificar de un conjunto de datos cuales son las observaciones atípicas para lo cual se clasifica en diferentes formas. Para ello se parte de dos consideraciones se considera una variable respuesta que será evaluada para establecer si es normal o tienen un comportamiento anómalo.

2.4. Técnicas de aprendizaje automático de detección de anomalías

Los diferentes estudios sobre técnicas de detección de anomalías se destacan por crear nuevas herramientas que adquieran la capacidad de distinguir entre clases ordinarias y anormales. Esta clasificación o detección de anomalías se puede dividir en tres grandes categorías basadas en la función de datos de entrenamiento utilizada para construir.

2.4.1. Detección de anomalías supervisada

La detección de anomalías supervisada describe la configuración en la que los datos se componen de datos completamente etiquetados, estos datos se dividen en conjuntos de datos de entrenamiento y prueba (Goldstein & Uchida, 2016).

Uno de los problemas de estos algoritmos radica en la existencia de que las clases suelen estar muy desequilibradas. Los datos con clases desequilibradas son aquellos que presentan una desviación amplia o ruido por lo que no todos los algoritmos de clasificación se adaptan perfectamente esta tarea (Goldstein & Uchida, 2016).

Estos algoritmos funcionan con conjunto de datos etiquetados por lo que no se recomienda emplear para la detección de anomalías, ya que de una estructura de datos las anomalías no se conocen.

2.4.2. Detección de anomalías semi supervisada

La detección de anomalías semi supervisada también utiliza conjuntos de datos de entrenamiento y de prueba. de entrenamiento consisten únicamente en datos normales sin anomalías (Goldstein & Uchida, 2016). La idea básica es que se aprende un modelo de la clase normal y las anomalías pueden detectarse después desviándose de ese modelo.

2.4.3. Detección de anomalías no supervisadas

Los datos de la variable respuesta no están etiquetados, la representación se considera un modelo de los datos, teniendo como objetivo descubrir las estructuras ocultas a partir de datos no etiquetados o inferir un modelo que tenga la densidad de probabilidad de los datos de entrada (Rincy & Roopam, 2021).

2.5. Algoritmos de detección de anomalías

Tabla 2.1 Tipos de algoritmos para detección de anomalías.

Algoritmo	Uso	Fuente
Isolation Forest	Este algoritmo utiliza un bosque de árboles de decisión para identificar observaciones anómalas. Cada árbol se construye aleatoriamente y las observaciones más difíciles de aislarse se consideran anómalas.	(Chater et al., 2022)

Local Outlier Factor (LOF)	Este algoritmo utiliza una métrica de LOF para medir la anomalía de cada observación. Las observaciones con un LOF más alto son consideradas anómalas.	(Adesh et al., 2024)
Elliptic Envelope	Ajusta un modelo de covarianza mínima determinante (covMcd) a los datos de entrenamiento. Luego, calcula la distancia de Mahalanobis de cada observación de prueba al centro del modelo. Las observaciones con una distancia más alta son consideradas anómalas.	(Vishwakarma & Kesswani, 2023)
One-Class SVM	Utiliza un modelo de máquina de vectores de soporte de una clase para identificar observaciones anómalas. El modelo se entrena utilizando solo las observaciones normales y clasifica las observaciones de prueba como anómalas si no se ajustan bien al modelo.	(Barbado et al., 2022)
DBSCAN	Utiliza un enfoque de agrupación basado en la proximidad para identificar observaciones anómalas. Las observaciones que están más cerca de un número mínimo de puntos (minPts) se agrupan en clústeres. Las observaciones que no se agrupan en clústeres son consideradas anómalas.	(Chen & Li, 2011)
K-Means	Emplea un enfoque de agrupación basado en la distancia Euclidiana para identificar observaciones anómalas. Las observaciones que se agrupan en clústeres más pequeños son consideradas anómalas.	(Nowak - Brzezińska & Horyń, 2020)

2.6. Métricas para su evaluación y eficacia

2.6.1. ROC

Una curva ROC (Receiver Operating Characteristic Curve) es un gráfico que muestra el rendimiento de un modelo de clasificación en todos los umbrales de clasificación. Esta curva representa la relación entre parámetros de sensibilidad y especificidad (Vujovic, 2021).

Para calcular los puntos en una curva ROC, podríamos evaluar un modelo de regresión logística muchas veces con diferentes umbrales de clasificación, pero esto es ineficiente. Afortunadamente, existe un algoritmo eficiente basado en ordenamiento que puede brindarnos esta información, denominado AUC.

2.6.2. AUC: Área bajo la curva ROC

El AUC (Area Under the ROC Curve) proporciona una medición agregada del rendimiento en todos los umbrales de clasificación posibles. Una forma de interpretar el AUC es como la probabilidad de que el modelo clasifique un ejemplo positivo aleatorio más alto que un ejemplo negativo aleatorio (Richardson et al., 2024).

2.6.3. Matriz de confusión

La matriz de confusión es una herramienta que nos muestra el desempeño de un algoritmo de clasificación, describiendo cómo se distribuyen los valores reales y las predicciones mediante 4 distintos casos basados en 4 variables que se comentan a continuación (Vujovic, 2021).

Valores actuales: Verdadero o Falso.

Valores predichos: Positivo o Negativo.

Verdadero positivo (VP): Se predice que es positivo y es verdad, fue clasificado correctamente.

Verdadero negativo (VN): Se predice que es negativo y es verdad, fue clasificado correctamente.

Falso positivo (FP): Se predice que es positivo y es falso, fue clasificado incorrectamente.

Falso negativo (FN): Se predice que es negativo y es falso, fue clasificado incorrectamente.

2.6.4. Sensibilidad (Recall)

Es la proporción de casos positivos que fueron correctamente identificadas por el algoritmo.

2.6.5. Especificidad (Specificity)

Se trata de los casos negativos que el algoritmo ha clasificado correctamente. Expresa cuan bien puede el modelo detectar esa clase.

2.6.6. Precisión (Precision)

Es la cantidad de predicciones positivas que fueron correctas. La mejor precisión viene dada por el valor 1 mientras que la peor precisión viene dada por el valor 0.

2.6.7. Exactitud (Accuracy)

Se refiere al porcentaje de casos positivos detectados.

2.6.8. F1

Esta métrica combina precisión y sensibilidad en un sólo valor, es de gran utilidad cuando la distribución de las clases es desigual. Respecto a las métricas se dan cuatro posibles casos:

Alta exactitud y sensibilidad: el modelo maneja perfectamente esa clase.

Alta exactitud y baja sensibilidad: el modelo no detecta la clase muy bien, pero cuando lo hace es altamente confiable.

Baja exactitud y alta sensibilidad: El modelo detecta bien la clase, pero también incluye muestras de otras clases.

Baja exactitud y sensibilidad: El modelo no logra clasificar la clase correctamente.

Maximizar el valor de F1, conlleva a un mejor rendimiento del algoritmo.

2.7. Casos de éxito

En un estudio en la industria de Chihuahua se busca reducir riesgos y mejorar rendimientos de todos los componentes de un horno en la industria del acero explorando

algoritmos de Isolation Forest y Autoencoders para la detección de anomalías, y la programación dinámica como método seleccionado para segmentar los gases, logrando una efectividad por encima del 85%, para que el umbral utilizado para la detección de segmentos sea el mismo y funcione de manera adecuada se ha ajustado el umbral de los parámetros (Arciba, 2023).

En otro estudio se utilizó un algoritmo para detectar anomalías en datos de tipo nominal, se trata de un algoritmo global de “k” vecinos más cercanos, un algoritmo de agrupamiento denominado k-means y un método estadístico denominado chi-cuadrado, se ha implementado sobre una base de datos de clientes que han solicitado un crédito financiero, se analizó datos de 1180 tuplas, en donde al alterar datos y generar valores atípicos se demostró que el método propuesto es capaz de detectar todos los valores atípicos introducidos (Orellana & Cedillo, 2020).

La Inteligencia Artificial crea una potencia de trabajo totalmente novedosa, que puede replicar las actividades laborales a mayor escala y velocidad e incluso realizar tareas superiores a los humanos (Ruano et al., 2019). La implementación de soluciones innovadoras a los problemas sociales, económicos y ambientales están generando una gran oportunidad para que los países de la región desarrollen ecosistemas de innovación con estándares y herramientas confiables basadas en IA (Barragán-Martínez, 2023).

3. Metodología

La metodología que se estableció mediante fases se definió la secuencia de pasos a seguir para la implementación y validación de los algoritmos de detección de anomalías, estos algoritmos de detección de anomalías son claves para detectar patrones anómalos en la estructura de los datos; para lo cual, se realizó un análisis inicial para determinar los parámetros que constituyen la base de datos, selección de las variables correctas para el entrenamiento de los algoritmos, ajuste de parámetros para mejorar el rendimiento y visualización de los resultados a partir de indicadores o métricas claves que permitieron mostrar el comportamiento de los mejores algoritmos de detección de anomalías en la base de datos probada.

3.1. Enfoque de investigación

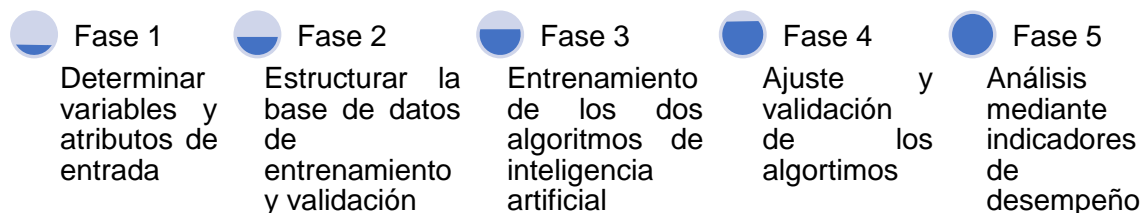
La investigación busca un enfoque de uso de herramientas de inteligencia artificial para determinar mediante algoritmos de detección de anomalías evaluar un conjunto de datos obtenidos de un control de calidad en el área de producción y poder comparar los

resultados numéricos a través de métricas clave y determinar la medida o sensibilidad de emplear estos algoritmos. El objetivo de este análisis es aprovechar las herramientas disponibles en la actualidad que permitan obtener resultados certeros de entrenar algoritmos de inteligencia artificial.

La recolección de datos se llevó a cabo en el área de producción de una empresa de MIPyme textil dedicada a la confección de tela plana de algodón y polímeros sintéticos. En el área se cuenta con máquinas tejedoras industriales capaces de producir grandes volúmenes de tela por día aumentando su producción.

La metodología que se establece a continuación define las fases a seguir para la implementación y validación de los algoritmos de detección de anomalías.

Ilustración 3.1 Metodología.



Fuente: Elaboración propia.

3.2. Fase 1: Selección de variables y parámetro de entrada

Para el estudio en cuestión se empleó una base de datos de una MiPyme de la ciudad de Cuenca, que consta de datos etiquetados con diferentes variables y observaciones. Las características de los datos corresponden a 17 variables que describen el control de calidad de textiles. Las variables son obtenidas mediante muestras que permiten visualizar el comportamiento del producto defectuoso y su grado de incidencia sobre el producto terminado.

Las variables están relacionadas directamente con el proceso de producción, como el tipo de defecto, longitud del defecto, cantidad de defectos y metros de tela inspeccionada, estas variables son críticas afectando a la calidad del producto terminado y rendimiento del sistema productivo (Santillán, 2020).

Por otro lado, el sistema de calificación cuatro puntos establecido en la norma técnica ASTM D5430 - 07 (2011) (American Society for Testing and Materials) establece los

parámetros y variables para clasificar defectos textiles visibles en las telas, este sistema es ampliamente utilizado por las industrias para asegurar un control estandarizado y minimiza productos defectuosos (Hanbay et al., 2016) (Hanbay et al., 2016b).

Se distinguen diferentes tipos de variables entre categóricas y numéricas, destacando variables nominales, discretas y continuas. El estudio cuenta con 6000 observaciones que han sido analizados y depuradas para evitar error en la corrida del código creado en R. Se eligió este programa para realizar el proceso de detección de anomalías ya que es un programa ágil y flexible al momento de analizar un amplio volumen de datos.

La depuración consiste en identificar valores faltantes, no codificados establecer si existe datos que no corresponde a la naturaleza normal de la variable, ya que es muy común que en procesos de control de calidad no se registren adecuadamente una observación (Orellana & Cedillo, 2020).

Una vez identificado los valores faltantes estos deben ser separados o corregidos de acuerdo a criterios establecidos por la experiencia estadística, ya sea reemplazar con la mediana, media u otro indicador estadístico, o a su vez deberá ser eliminado, sin embargo, si se elimina una observación indicaría que no se considera como relevante en el estudio (Ji et al., 2024).

Una vez depurada la base de datos se determina la variable que contiene las etiquetas anómalas, ya que con la misma se verificará al final si el algoritmo determinó bien o no la clasificación. Definido esta variable se codificó la variable predictora en formato binario ya que es una característica de los algoritmos de detección de anomalías., para lo cual se considera una observación anómala con 0 y 1 a una variable normal.

Las variables consideradas adecuadas para el entrenamiento del algoritmo son numéricas ya que muestran los aspectos más representativos. Es importante notar que los algoritmos de detección de anomalías funcionan mejor si son entrenados con variables cuantitativas (Rincy & Roopam, 2021). Estas variables aun no son adecuadas para entrenar a los algoritmos ya que no tienen una escala de medida que sea correspondiente entre todas las variables, para ello se debe realizar un proceso de normalización.

Este paso permite llevar a todas las variables de una escala diferente a tener una escala de misma dimensión, pero sin alterar el comportamiento o distribución de los datos. Una base de datos normalizada se caracteriza en contener datos con escala de máximos y

mínimos para cada una de las variables, comparando la media, desviación típica y otros factores representando una etiqueta en una escala de 1 y -1. (Zaragoza, 2020) menciona que se lleva a cabo la normalización de los datos para que las muestras de entrada tengan media cero y varianza unitaria.

Limpieza de datos

En las organizaciones se manejan grandes volúmenes de datos y normalmente se encuentran problemas en las variables y observaciones mal registradas, para corregir estos errores se realiza una depuración minuciosa tomando en cuenta recomendaciones de buenas prácticas.

Si los datos no tienen el nivel de calidad adecuado, deben realizarse tareas de limpieza para transformarlos en datos consistentes, corrigiendo datos incorrectos, corruptos, con formato incorrecto, duplicados o incompletos (Orellana & Cedillo, 2020).

En R, existen varios paquetes para llevar a cabo estos trabajos de los que se han empleado tidyverse para la manipulación de ficheros y variables; caret se utilizó para imputar los datos faltantes o perdidos.

Se debe seguir un proceso general de limpieza identificando el estado actual de los datos y realizando actividades necesarias de limpieza. Primero se analizó si falta encabezados, codificación de caracteres desconocida, categorías incorrectas, desconocidas o inesperadas.

Se destacan algunas técnicas para la limpieza de datos; una vez establecidas las correcciones, los datos pueden ser leídos por un dataframe en el programa asignando correctamente nombres, tipos y etiquetas.

- Ajustar los errores estructurales.
- Eliminación de registros innecesarios o duplicados.
- Manejo de valores inusuales.
- Manejo de valores inexistentes.
- Control y validación de la calidad de los datos.

La tipificación de las variables categóricas y la imputación de los valores insuficientes con la mediana se consideraron soluciones adecuadas.

3.3. Fase 2: Estructurar la base de datos de entrenamiento y validación:

En el estudio de problemas de ciencia de datos es crucial encontrar el modelo o algoritmo que mejor resuelva y proporcione mejores resultados (Montero, 2024).

En otras palabras, un algoritmo que no solo ajuste bien los datos pasados sino, lo que es más importante, que proporcione predicciones futuras acertadas y precisas. Para ello, inicialmente, se dividen los datos en dos subconjuntos.

Los algoritmos de detección de anomalías se deben entrenar a partir de datos proporcionados de la base de datos original, se separa un porcentaje específico para entrenamiento y un porcentaje para validación. Este porcentaje corresponde según la revisión de literatura a un 80% de división de datos para entrenamiento y un 20% para validación, en ocasiones se evidencia que puede llegar hasta una división de 70 a 30. La proporción de datos considerados debe tener el mismo comportamiento en distribución tanto para entrenamiento y validación, evitando sesgos en la elección.

Se utilizaron funciones de distribución que permitirá elegir los datos con las mismas características para entrenamiento y validación para evitar caer en errores de sensibilidad, ya que sesgar a la muestra es entrenar al algoritmo con cierta característica y forma y validar con otra distribución de los datos obteniendo resultados sobre ajustados o sin ajuste.

3.4. Fase 3: Entrenamiento de los algoritmos de inteligencia artificial

Se utilizaron diferentes algoritmos para detectar anomalías, tales como Isolation Forest o bosques de aislamiento, k-means, variaciones de Isolation Forest, entre otros. Todos los algoritmos empleados se caracterizan ya que corresponden a algoritmos de inteligencia artificial no supervisado y son empleados para detección de anomalías (Rincy & Roopam, 2021)

En el programa R se cuentan con los códigos con toda la programación de cada algoritmo y se realiza el entrenamiento. Los algoritmos empleados son los siguientes:

- **Local Outlier Factor (LOF):** Detecta anomalías basadas en la densidad local.
- **One-Class SVM:** Análisis de discriminante vs Clasifica una clase en función de los datos de entrenamiento.
- **DBSCAN (Density-Based Spatial Clustering of Applications with Noise):** Identifica puntos que no pertenecen a ningún clúster.

- **Elliptic Envelope:** Asume que los datos utilizan una distribución gaussiana y detecta anomalías basadas en esta suposición.
- **K-Means Clustering:** Identifica anomalías basadas en la distancia de los puntos a sus centroides más cercanos.

Los algoritmos empleados destacan por su capacidad para identificar patrones inusuales en diferentes tipos de datos lo que permite cubrir diversos enfoques para garantizar un análisis robusto de las anomalías (Ferreira et al., 2024).

3.5. Fase 4: Ajuste y validación de los algoritmos utilizados:

En esta etapa se visualiza los resultados preliminares del entrenamiento de los algoritmos y se establece ajuste de parámetros, (Mehrotra et al., 2017) sugiere establecer selección del mejor valor para el parámetro de contaminación, con distintas ejecuciones y una selección manual posterior del mejor valor para cada conjunto de datos.

Se seleccionó manualmente el mejor valor para cada parámetro mediante análisis de las métricas de Sensibilidad, Especificidad y Exactitud y se ejecutó el algoritmo elegido sobre el conjunto de prueba con parámetros adaptados y modificados manualmente para obtener un mejor rendimiento (Goldstein & Uchida, 2016)

Para un enfoque más adecuado se consideró desde el principio de los algoritmos establecer un porcentaje de datos para entrenamiento y validación para lograr estimar su exactitud en el conjunto de validación.

En el ajuste los parámetros se consideraron lo recomendado por defecto en el programa de R para cada uno de los modelos, esto para optimizar el rendimiento computacional, y son conocido como conjunto de buenas prácticas en el campo de la investigación. Estos hiper parámetros se establecen en la creación del código del modelo respectivo.

Sin embargo, se destacan el uso de técnicas para determinar los hiper parámetros diferentes tales como técnicas bayesianas, reducción del gradiente, búsqueda en cuadrícula, muestra aleatoria y optimización secuencial; permitiendo obtener parámetros óptimos proporcionando una configuración óptima que dará mejores resultados.

3.6. Fase 5: Análisis de resultados mediante indicadores:

Se determinó la matriz de confusión de las variables clasificadas correctamente respecto a las variables que no fueron clasificadas correctamente. Ese estableció una comparación

de todos los modelos para determinar cuál de los modelos se ajusta mejor a la estructura de los datos.

Se determinó los indicadores de sensibilidad, precisión y exactitud, métricas que indican como es el comportamiento de cada modelo y permite identificar cual tuvo mejor rendimiento.

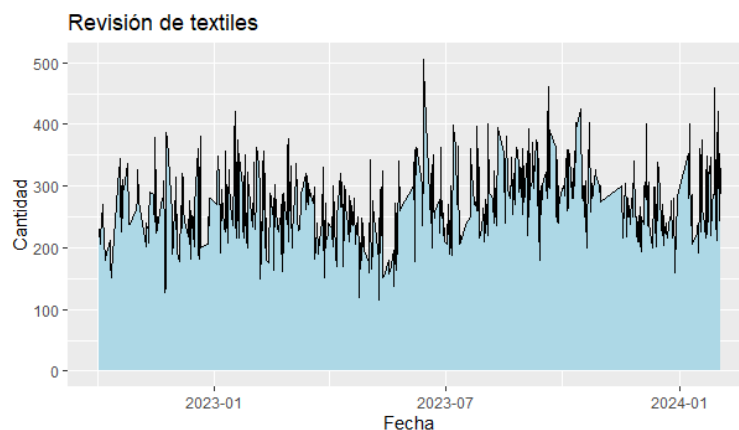
Mediante graficas con la ayuda de la librería ggplot2 se graficó las métricas determinadas en donde se visualizó el comportamiento de los modelos y permitió una interpretación visual del mejor algoritmo de detección de anomalías. Las métricas determinadas corresponden a sensibilidad, precisión exactitud, especificidad, f1, AUC, y el área bajo la curva AUC – ROC.

4. Resultados

4.1. Análisis Exploratorio de datos

La base de datos utilizada corresponde un conjunto de registros tomados de la inspección de calidad de una organización MiPyme textil, cuenta con 6766 observaciones en 17 variables, estos datos se han registrado desde el periodo del 2022 hasta la fecha, como se muestra en la ilustración 4.1 y siguen registrándose ya que permite tomar decisiones de mejora en el proceso productivo.

Ilustración 4.1 Historial de revisión de calidad



Fuente: Elaboración propia.

Se realizó la identificación de valores faltantes, codificaciones incorrectas de manera minuciosa en cada una de las variables de forma visual y general. Los resultados obtenidos muestran en la tabla 4.1. Existen dos variables que presentan registros vacíos.

Tabla 4.1 Identificación de valores faltantes

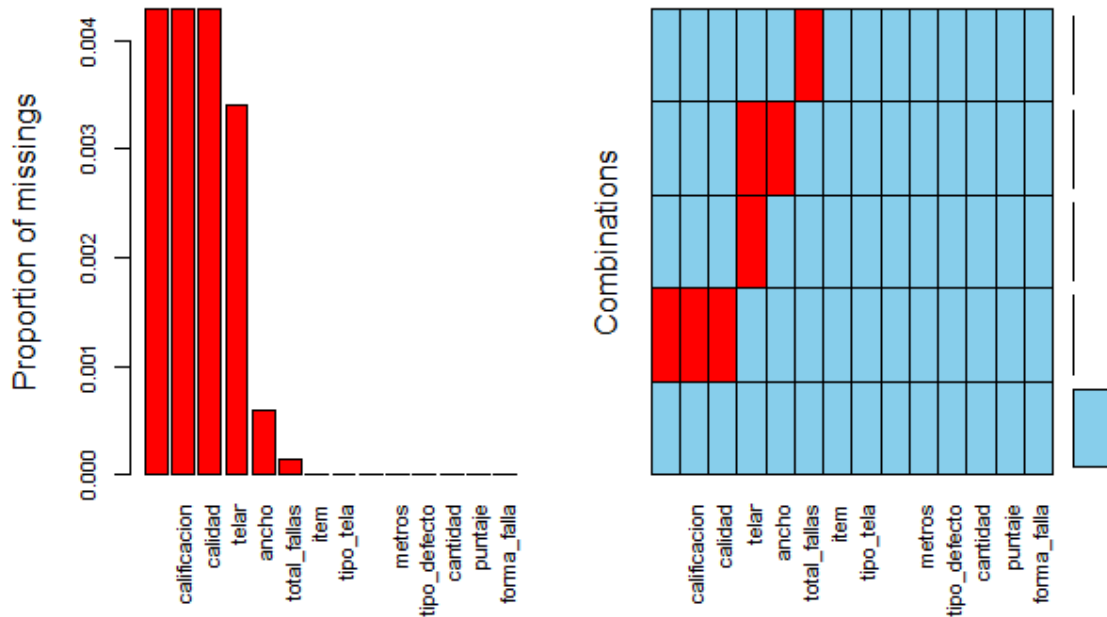
Variable	Valores faltantes	Valores Totales	Valores existentes
Ítem	0	6766	6766
tipo tela	0	6766	6766
fecha revisión	0	6766	6766
Telar	23	6766	6743
Metros	0	6766	6766
Ancho	4	6766	6762
tipo defecto	0	6766	6766
Cantidad	0	6766	6766
Puntaje	0	6766	6766
forma falla	0	6766	6766
total fallas	0	6766	6766
defectos totales	0	6766	6766
Calificación	0	6766	6766
Calidad	0	6766	6766
Semana	0	6766	6766
Mes	0	6766	6766
Año	0	6766	6766
Anomalía	0	6766	6766

Fuente: Elaboración propia.

Se tipificó correctamente las variables eliminando tildes, espacios y caracteres como puntos y comas. Se realizó estas depuraciones ya que el programa R detecta a cada carácter como único en el caso de categorías que contengan minúsculas son diferentes a mayúsculas aún si signifique lo mismo. En el caso del idioma inglés que es la base del programa, este no admite tildes, tampoco espacios entre dos palabras y así demás consideraciones en el análisis de base de datos.

Este tratamiento de datos se realizó a las variables que se muestran en la ilustración 4.2, estos datos faltantes o perdidos se dan comúnmente en el manejo de base de datos en producción. En este sentido se observa que las variables con mayor proporción de valores perdidos no supera el 0,4%, destacando 4 variables. La variable con mayor proporción de valores perdidos es la variable de defectos totales, calificación y calidad con 0,428% seguido de 0,33% de telar, además de un porcentaje muy pequeño representado en las variables de ancho y total fallas.

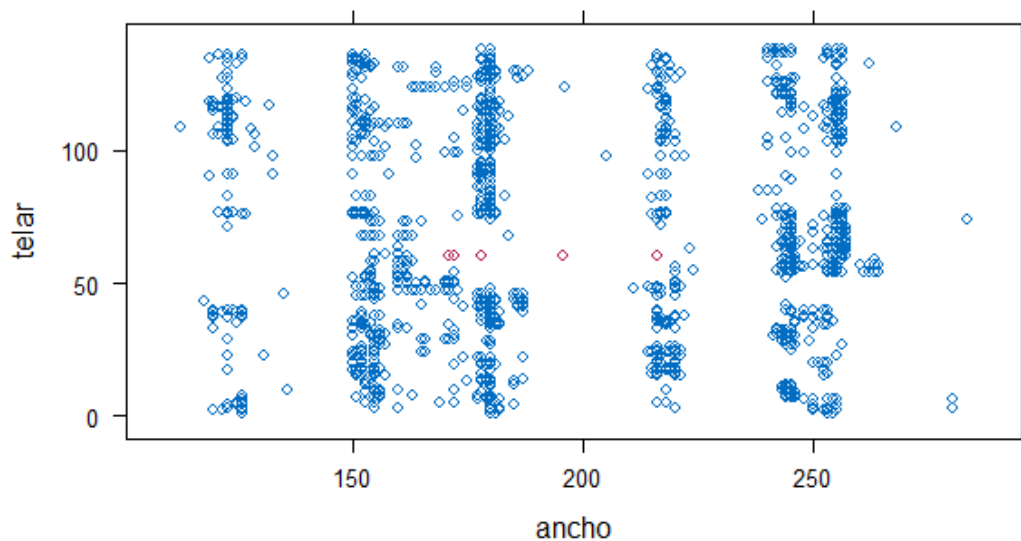
Ilustración 4.2 Variables con datos perdidos



Fuente: Elaboración propia.

Por tanto, se realiza un análisis de estos valores faltantes y mediante imputación de la media de datos se realiza la imputación de los mismos mostrando el resultado en la ilustración, para tomar como ejemplo en la ilustración 4.3 se puede observar en un diagrama de dispersión de color rojo los valores imputados.

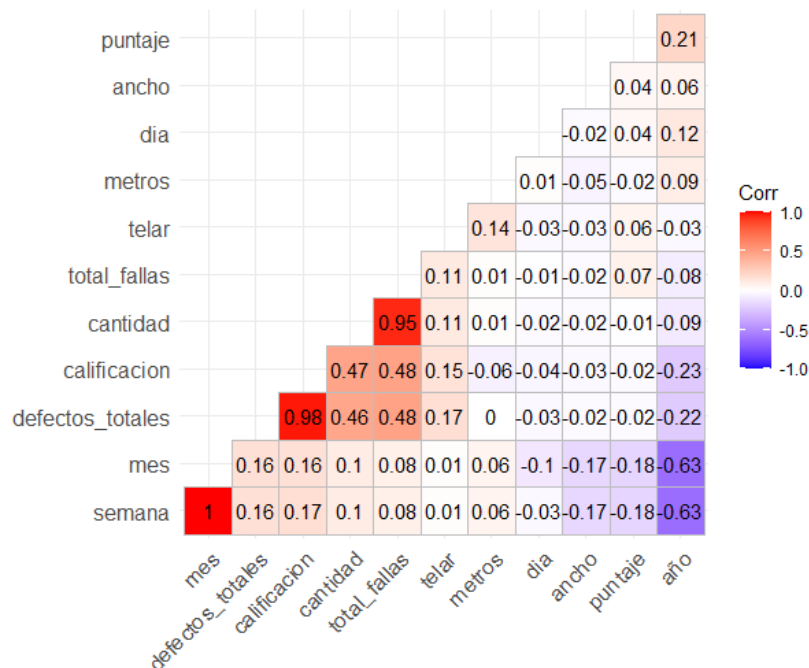
Ilustración 4.3 Datos imputados



Fuente: Elaboración propia.

En la ilustración 4.4 se puede observar un diagrama de correlación de variables utilizadas en el tratamiento de los datos. En general se muestra una correlación entre las variables, este método indica el grado de correlación entre las variables, aquellos valores positivos y cercanos a uno se visualiza de color rojo tienen correlación positiva, y de color azul violeta es el indicativo a correlación negativa.

Ilustración 4.4 Correlación de variables

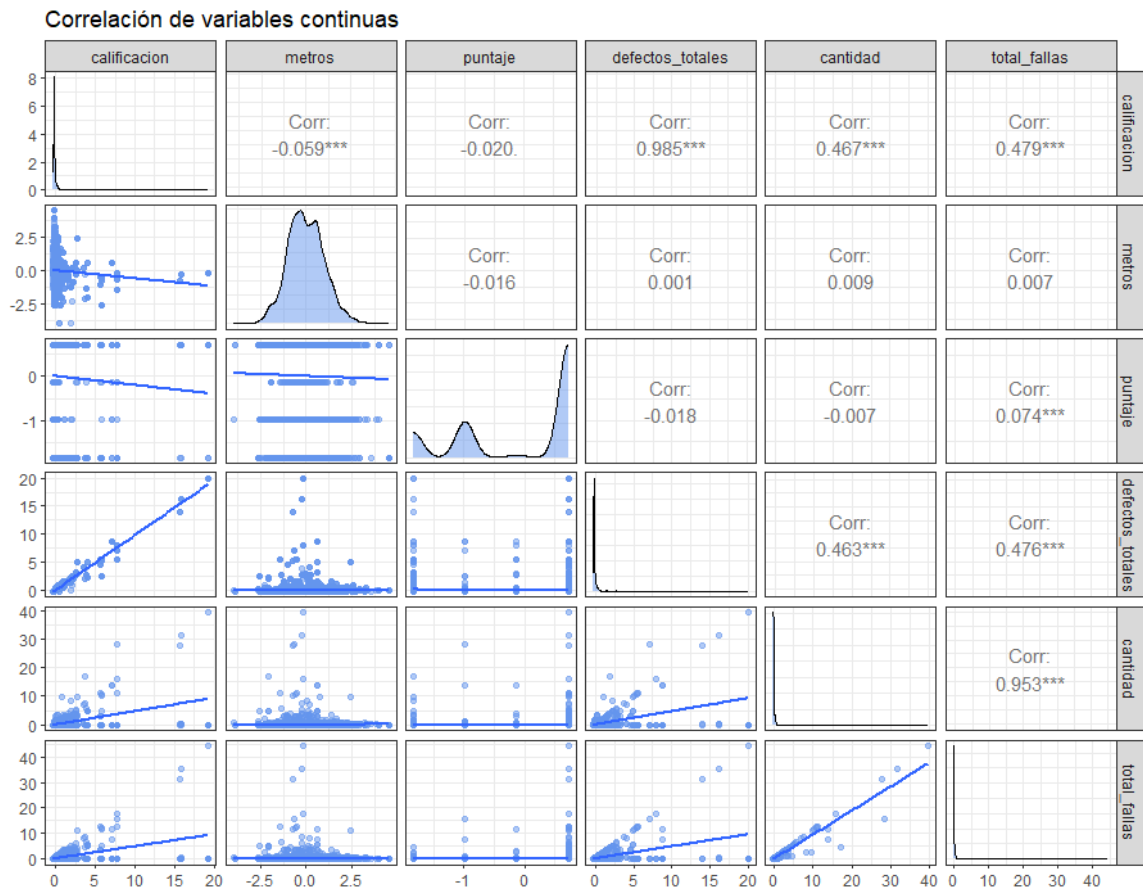


Fuente: Elaboración propia.

Los resultados muestran que algunas variables muestran correlación fuerte, destacando total fallas y cantidad defectos, mientras que otras variables tienen una débil correlación con valores cercanos a 0. La ilustración 4.5 a continuación muestra la distribución de las variables y su grado de correlación entre las mismas destacando defectos totales y calificación mostrando que la una variable tiene incidencia en la otra.

El análisis entre las variables de calificación y metros es muy débil, indicando una relación negativa (-0.0020) sugiriendo que no existe una relación, lo mismo ocurre al comparar las variables de calificación y puntaje con una correlación de 0.001 indicando ausencia de relación, sin embargo, al comparar los defectos totales y cantidad se encuentra una correlación positiva, indicando que, si cantidad total aumenta, también lo hace defectos totales.

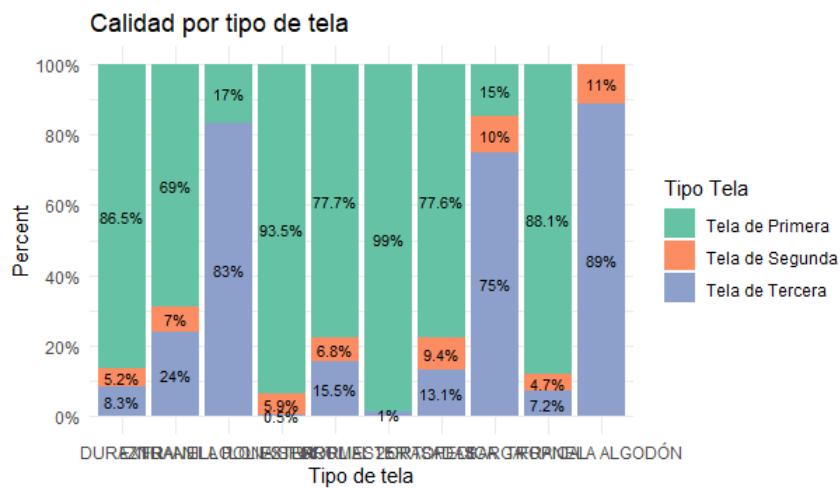
Ilustración 4.5 Matriz de dispersión



Fuente: Elaboración propia.

En el análisis de la ilustración 4.6 se establece que de estas dos variables comparadas se obtiene que 7 de 10 tipos de tela en promedio registran Tela de Primera con un valor superior al 70%, sin embargo, se muestra que el restante de telas registra con un valor superior al 75% tela de tercera, la concentración en tela de segunda muestra un valor promedio que no supera el 11%.

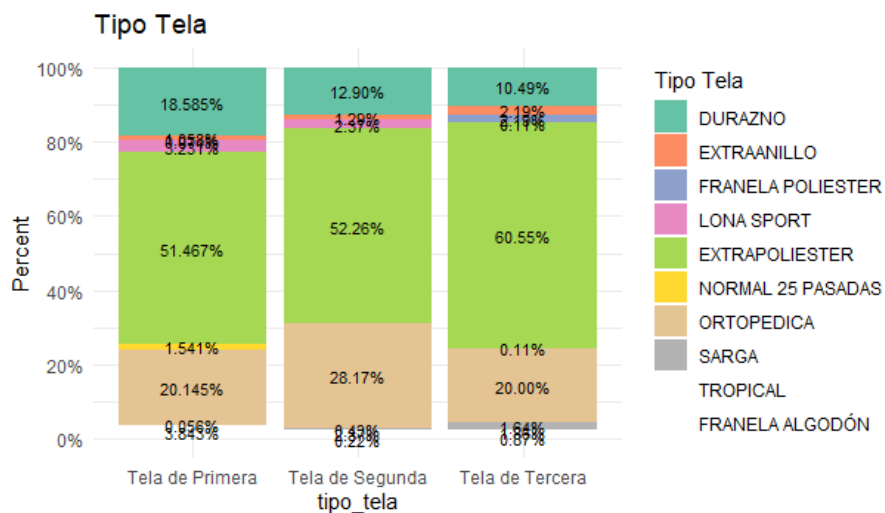
Ilustración 4.6: Análisis de la variable tipo tela y calidad



Fuente: Elaboración propia.

El análisis mostrado en la ilustración 4.7 el tipo de tela con mayor inspección y está presente en tres categorías es extra poliéster con un promedio de 54%, seguido de tela ortopédica con un 22%, el tercer lugar se destaca con 13% a tela piel de durazno, el restante de tela inspeccionada no supera el 5%, estos resultados muestran que tipo de tela se han revisado con más frecuencia.

Ilustración 4.7 Calidad por tipo de tela

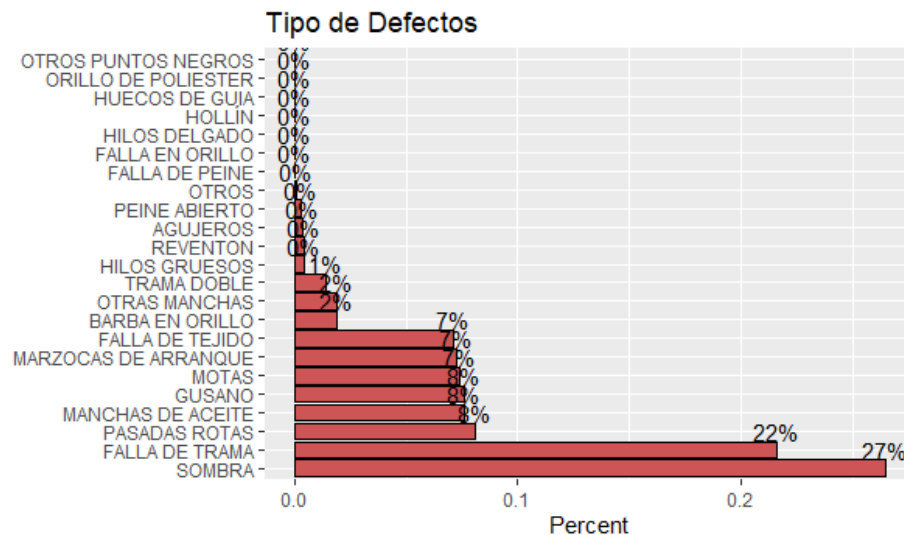


Fuente: Elaboración propia.

En la ilustración 4.8 se analizó la variable de tipos de defectos mostrando que la mayor concentración en defectos es debido a fallas en la trama del tejido con un 27% y sombras

con un 22%, esta concentración de defectos supera el 40% de defectos encontrados, en este caso los problemas de productos defectuosos corresponden a las características del hilo empleado mostrados. Por otro lado, pasadas rotas y manchas de aceite que representan un 8% del total de defectos corresponden a fallas de máquina, por tanto, se destaca la necesidad de establecer un procedimiento de control y mantenimiento preventivo.

Ilustración 4.8 Análisis de tipos de defectos



Fuente: Elaboración propia.

Una vez conocida la distribución de los datos y su relación se procedió a determinar variables y parámetros a emplear.

4.2. Selección de variables y parámetros de entrada.

El análisis de algoritmos de detección de anomalías se realizó a partir de datos numéricos, por ende, la base de datos inicial que contenía 18 variables se redujo el número de variables a 6 de tipo numérico mostrándose su comportamiento en la ilustración 4.5. Posteriormente se realizó un proceso de escalado o bien denominado normalización de variables. En la ilustración 4.9 la función scale permite normalizar la base de datos y crear un nuevo dataframe.

Ilustración 4.9 Código de escalado

```
####escalado####  
data_scaled <- scale(data_numeric)  
data_scaled_df <- as.data.frame(data_scaled)
```

Fuente: Elaboración propia.

En la ilustración 4.10 con la función mutate se creó y añadió a la base de datos una nueva variable, la variable anomalía que contienen la variable respuesta a ser comprobada, esta es binaria de acuerdo a la condición de que si los valores de la variable de calidad corresponden a tela de tercera se considera una observación codificada con 1 y 0 corresponden a observaciones normales.

Ilustración 4.10 Variable respuesta

```
# Preprocesamiento  
data <- data %>% mutate(anomalia = ifelse(calidad == "Tela de Tercera", 1, 0))
```

Fuente: Elaboración propia.

4.3. Estructuración de base de datos de entrenamiento y validación

Se decidió emplear la división de la base de datos en un porcentaje 70/30, el 70% de los datos se utilizaron para entrenar los modelos y el restante se utilizó para validación. Este proceso se llevó a cabo en cada uno de los modelos de detección de anomalías.

En la ilustración 4.11 se dividió la base de datos original en base de entrenamiento (trainData) y base de validación (testData) en una relación de los datos del 70% para entrenamiento y 30% en validación.

Ilustración 4.11 División de datos en entrenamiento y validación

```
# Dividir los datos en conjuntos de entrenamiento y validación  
set.seed(123)  
trainIndex <- createDataPartition(data_scaled_df$anomia, p = 0.7, list = FALSE)  
trainData <- data_scaled_df[trainIndex, ]  
testData <- data_scaled_df[-trainIndex, ]
```

Fuente: Elaboración propia.

4.4. Parámetros utilizados para el entrenamiento

4.4.1. Algoritmo Isolation Forest

Se inicializó el modelo con los parámetros de tamaño de muestra para construir el árbol de 256, y número de árbol para el bosque de 100. Posteriormente se ajustó el modelo para que aprenda a patrones de normalidad y anomalías. En cada observación se realizó la predicción de anomalías con la función de anomaly score. Basándose en el umbral (0,65 para el caso) de puntaje de anomalía el modelo clasificó observaciones anómalas y normales.

4.4.2. Algoritmo Elliptic Envelope

El procedimiento ajustado del modelo (Elliptic Envelope) para detectar anomalías, calculó las distancias de Mahalanobis para encontrar observaciones anómalas y se evaluó la efectividad del modelo, las predicciones se compararon con las etiquetas reales y el umbral para clasificar las anomalías se estableció un percentil 95 de las distancias de Mahalanobis.

El algoritmo ajustó un modelo de covarianza mínima determinante (covMcd) a los datos de entrenamiento. Si la distancia excede un umbral basado en una distribución Chi-cuadrado, se clasifica como anomalía.

4.4.3. Algoritmo LOF (Local Outlier Factor)

Para la implementación del algoritmo Local Outlier Factor (LOF), primero se instalaron y cargaron los paquetes necesarios para el análisis. Luego, se ajustó el modelo LOF a los datos escalados, calculando un puntaje LOF para cada observación, lo que mide qué tan anómalo es cada punto respecto a sus vecinos. Posteriormente, se definió un umbral basado en el percentil 95 de los puntajes LOF para clasificar las observaciones como anomalías o no.

Se generaron predicciones clasificando cada punto como anomalía o no, según si su puntaje LOF superaba el umbral establecido. A continuación, se creó una matriz de confusión para comparar las predicciones con las etiquetas reales de anomalías, y se evaluó el rendimiento del modelo calculando varias métricas de rendimiento, como la exactitud, sensibilidad y especificidad. Este proceso permitió identificar y evaluar las anomalías en el conjunto de datos, proporcionando una visión clara del rendimiento del modelo a través de la matriz de confusión y las métricas asociadas.

4.4.4. Algoritmo One-Class SVM

El One-Class SVM entrenó un modelo de máquina de vectores de soporte de una clase utilizando los datos de entrenamiento. Se utilizó un kernel radial y el tipo de clasificación fue 'one-classification'. Las predicciones se generaron aplicando el modelo a los datos de prueba y convirtiendo los resultados a 0 (normal) o 1 (anomalía).

4.4.5. Algoritmo DBSCAN

Para el algoritmo DBSCAN se aplicaron a los datos de entrenamiento, agrupando las observaciones cercanas basadas en un radio (eps) de 0.5 y un número mínimo de puntos (minPts) de 5. Las predicciones se generaron asignando 1 a las observaciones clasificadas como ruido (cluster -1) y 0 a las demás.

4.4.6. Algoritmo KMeans

Por último, K-Means agrupó los datos de entrenamiento en 2 clústeres utilizando el algoritmo k-means. Para cada observación de prueba, se asignó el cluster más cercano basado en la distancia Euclidiana a los centroides. Si el cluster asignado corresponde al cluster con el menor número de observaciones, se predice 1 (anomalía), de lo contrario 0.

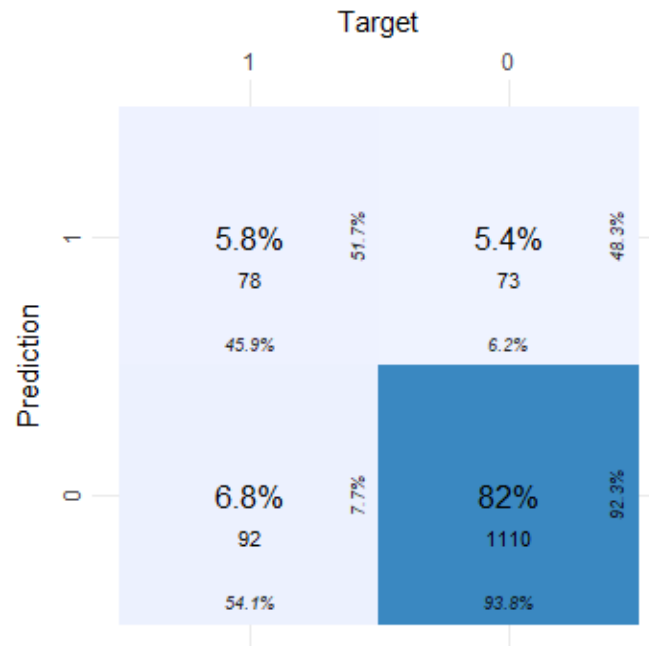
4.5. Comparación de métricas de desempeño

En esta investigación, se evaluó el desempeño de diversos algoritmos de detección de anomalías no supervisados para identificar defectos en muestras de tela plana. Los modelos analizados incluyen Isolation Forest, Local Outlier Factor (LOF), Elliptic Envelope, One Class SVM, DBSCAN y K-Means.

4.5.1. Matriz de confusión de Isolation Forest

Los resultados de la matriz de confusión de la ilustración 4.12 muestran el resultado del entrenamiento del modelo de Isolation Forest, en este sentido se tiene que el modelo es capaz de detectar anomalías con pocos falsos positivos y negativos.

Ilustración 4.12 Matrix de confusión del modelo Isolation Forest



Fuente: Elaboración propia.

El modelo muestra que el 82% fueron clasificados correctamente como positivos o normales, mientras que el 5,8% fueron clasificados correctamente como anomalías. No obstante, el 12,2 % de los datos fueron clasificados incorrectamente por el modelo, 6,8% de las observaciones fueron incorrectamente clasificadas como normales y 5,4% de las observaciones se clasificaron como anómalas.

Las instancias correctamente clasificadas por el modelo (Precisión) representan el 92% de los datos de validación, indicando un porcentaje positivo para el caso del modelo de Isolation Forest. Las instancias positivas se muestran que el modelo logra un rendimiento muy bueno con el 93,8% de los datos son clasificados correctamente. En el caso de anomalías el modelo refleja solo un 45,9% de los datos clasificados correctamente.

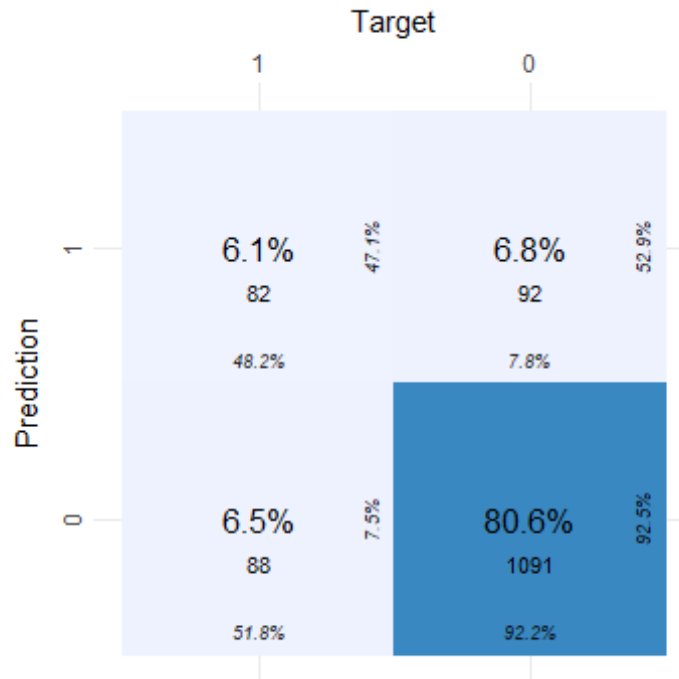
En una visión más completa del modelo con el indicador F1 que indica un equilibrio entre sensibilidad y precisión muestra un 93,08% de estabilidad.

4.5.2. Matriz de confusión de LOF

Para el caso del modelo LOF, se muestra en la ilustración 4.13, un 80.6% de los datos clasificados correctamente como normales y un 6.1% clasificados correctamente como anómalos. Por el contrario, el porcentaje clasificaciones incorrectas este modelo obtuvo

13.3% de los cuales 6.5% de las observaciones fueron incorrectamente clasificadas como normales y 6.8% de las variables normales erróneamente se clasificó como anomalías.

Ilustración 4.13 Matriz de confusión de LOF

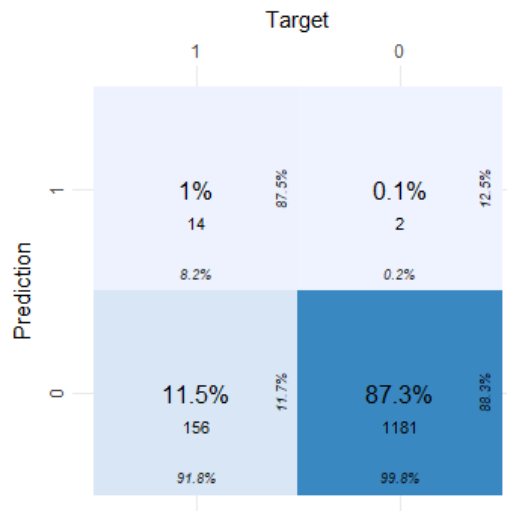


Fuente: Elaboración propia.

4.5.3. Matriz de confusión de Elliptic Envelope

Para el caso del algoritmo de la ilustración 4.14 se observa que existe una correcta clasificación de observaciones normales con una tasa verdaderos positivos de 87,2%, mientras que una tasa de verdaderos negativos muy baja con solo el 1%, es decir este modelo no clasifica anomalías de acuerdo a las observaciones reales. Esto se contrasta con una tasa de 11,5% de observaciones clasificadas erróneamente y con una tasa de error de clasificación a anomalías representando solo el 0,1%. Este modelo considera mejor la clasificación no reconoce valores atípicos del conjunto de datos después de realizar la validación.

Ilustración 4.14 Matriz de confusión del modelo *Elliptic Envelope*



Fuente: Elaboración propia.

4.5.4. Matriz de confusión de One-Class SVM

Los resultados obtenidos en la matriz de confusión de la ilustración 4.15 muestran que los valores correctamente clasificados como normales representan 41,2%, y las observaciones correctamente clasificadas como anomalías representan solo el 3,8%. Por el contrario, los resultados del modelo muestran una tasa elevada de clasificación incorrecta de datos normales en anomalías con una tasa de 46,3%, y un 8,8% de observaciones incorrectamente clasificadas como normales, esto muestra que el modelo tiene una elevada tasa de clasificación incorrecta, es decir, tiende a cometer errores de clasificación.

Ilustración 4.15 Matriz de confusión del modelo One Class SVM

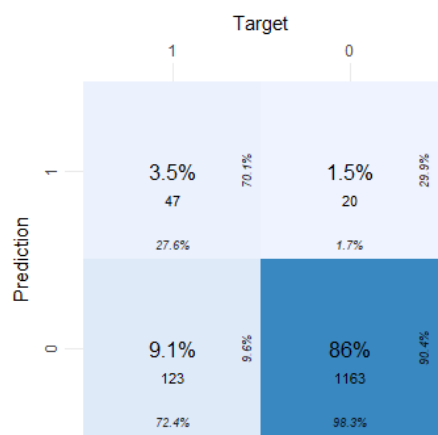


Fuente: Elaboración propia.

4.5.5. Matriz de confusión de DBSCAN

La tasa de clasificación correcta de verdaderos positivos representa el 86%, indicando que el modelo logra clasificar correctamente 1163 observaciones como normales, por otro lado, la tasa de verdaderos negativos el modelo logra clasificar observaciones como anómalas un 3,5%. En contraste, el modelo tiene una tasa elevada de clasificación incorrecta representando el 9,15% de las observaciones anómalas como normales. El restante 1,5% de las observaciones se clasificaron erróneamente como normales cuando no lo son. Estos resultados se pueden contrastar en la ilustración 4.16.

Ilustración 4.16 Matriz de confusión del modelo DBSCAN

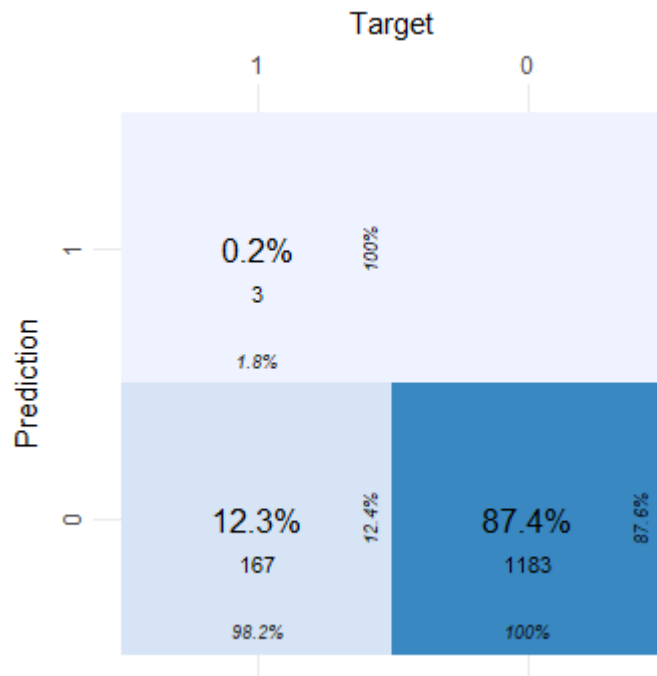


Fuente: Elaboración propia.

4.5.6. Matriz de confusión de K-means

En el último modelo los resultados obtenidos se muestran en la ilustración 4.17 indicando una tasa de clasificación de verdaderos positivos elevado correspondiendo al 87,4%. Asimismo, la tasa de clasificación de verdaderos negativos, es decir, valores clasificados como anómalos corresponde el 3,5% del total de observaciones. A diferencia del resto de modelos, la tasa de clasificación de falsos positivos representa el 12,3% indicando que el modelo clasificó 167 observaciones en su variable respuesta como normales cuando no lo son. Otro aspecto importante a resaltar es que no existe observaciones clasificadas como falsos negativos.

Ilustración 4.17 Matriz de confusión de K-means. Fuente: Elaboración propia.



Fuente: Elaboración propia.

4.5.7. Análisis de métricas

En la tabla 4.3 se establecieron el valor obtenido de las métricas de cada uno de los modelos empleados. Se destaca el modelo de Isolation Forest, ya que tiene una tasa de precisión del 92,35%,

Tabla 4.2 Métricas de desempeño

Métrica	Isolation Forest	Elliptic Envelope	LOF	One Class SVM	DBSCAN	KMeans
Sensitivity	0,9383	0,9983	0,8487	0,4708	0,0008	1,0000

F1	0,9308	0,9373	0,8746	0,5992	0,0017	0,9344
Precision	0,9235	0,8833	0,9021	0,8240	1,0000	0,8769
Accuracy	0,8780	0,8832	0,7871	0,4494	0,1264	0,8773
AUC	0,6986	0,5403	0,6038	0,6146	0,5004	0,5118
Specificity	0,4588	0,0824	0,3588	0,3000	1,0000	0,0235

Fuente: Elaboración propia.

Los resultados indican la precisión del algoritmo Isolation Forest indicando una precisión del 92,35% y F1 score de 93,08% indicando una tasa elevada de detección de anomalías con pocos falsos positivos y negativos, sin embargo, su especificidad (0.4588) y AUC (0.4588) son relativamente bajas en comparación con otros modelos.

Se destaca a los algoritmos Isolation Forest (0.8780), Elliptic Envelope (0.8832), KMeans (0.8773) e Isolation Forest (0.8780) como los más efectivos. Estos modelos muestran un rendimiento robusto en la clasificación general, indicando su capacidad para identificar correctamente tanto anomalías como puntos normales en los datos.

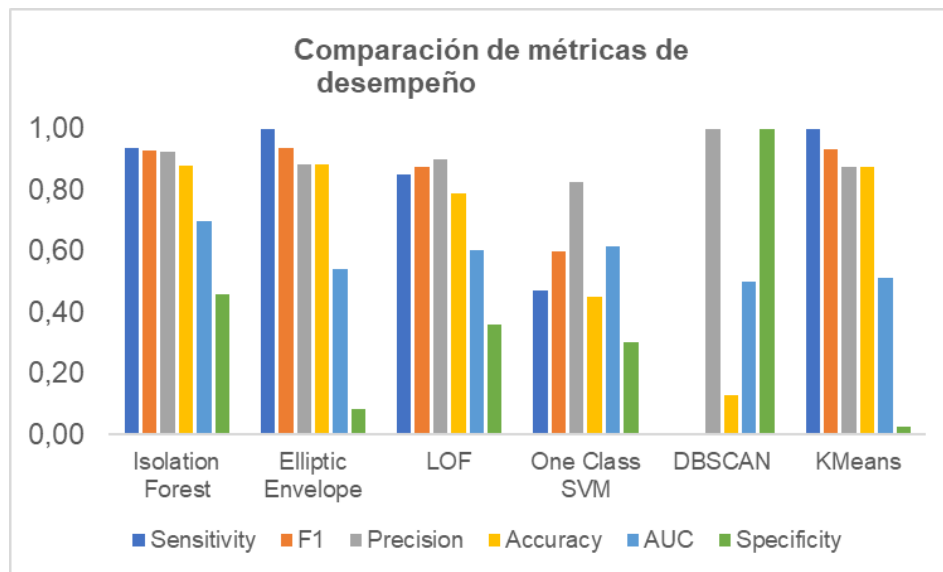
KMeans (1.0000) y Elliptic Envelope (0.9983) sobresalieron, demostrando una casi perfecta identificación de anomalías. Isolation Forest también mostró una alta sensibilidad (0.9383), posicionándose como una opción fuerte para detectar anomalías en el conjunto de datos.

La proporción de puntos normales correctamente identificados, fue más alta para DBSCAN (1.000), lo que indica su efectividad en minimizar falsos positivos. Isolation Forest (0.4588) mostró una especificidad moderada, mientras que One Class SVM (0.3000) tuvo un rendimiento más bajo, sugiriendo que este último es menos adecuado para este conjunto de datos específico.

La proporción de predicciones de anomalías que realmente son anomalías, fue notablemente alta en DBSCAN (1.000) e Isolation Forest (0.9235). Estos resultados indican que la mayoría de las predicciones de anomalías hechas por estos modelos son correctas, lo que es crucial cuando la precisión en la identificación de anomalías es vital.

Una evaluación equilibrada entre precisión y sensibilidad, y la capacidad de diferenciación del modelo resalta que Elliptic Envelope (0.9373) y KMeans (0.9344) obtuvieron los mejores F1 Scores, reflejando un buen equilibrio. En este sentido Isolation Forest demuestra con un AUC moderado (0.6986) sugiriendo su excelente capacidad para diferenciar entre anomalías y puntos normales, lo que lo hace muy confiable para la detección de anomalías.

Ilustración 4.18 Métricas de desempeño.



Fuente: Elaboración propia.

4.5.8. Índice de Silueta y Chi-cuadrado

El valor de chi-cuadrado determina la independencia de las variables, mientras que el índice de silueta determina la cohesión de los clústeres. En la tabla 4.4 se muestra que el mejor índice de silueta (0.6261) de Elliptic Envelope señaló clústeres bien definidos. DBSCAN (242.8667) mostró un valor más moderado, equilibrando la efectividad y la consistencia de las predicciones; sin embargo, su alto valor de chi-cuadrado indica una mayor discrepancia entre las predicciones y las observaciones reales.

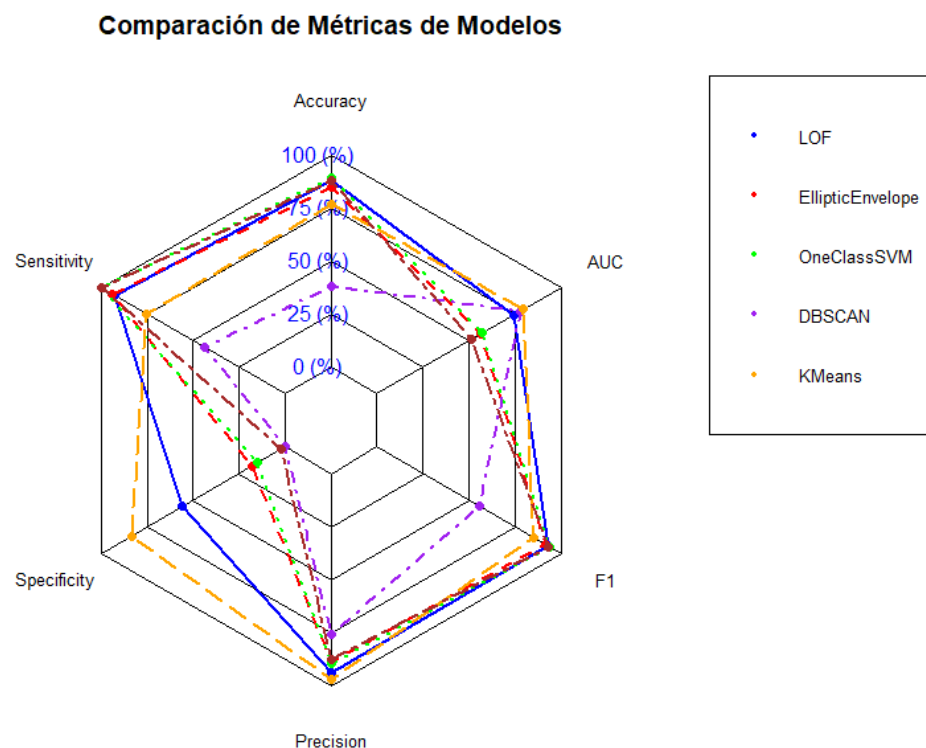
Tabla 4.3 Índice de silueta y Chi cuadrado

Modelo	Chi-squared	Silhouette
IsolationForest	181,9160	0,2777
LOF	96,9697	0,2366
EllipticEnvelope	13,2300	0,6261
OneClassSVM	191,8632	0,0735
DBSCAN	242,8667	0,4147
KMeans	21,9822	0,2647

Fuente: Elaboración propia.

Para detectar anomalías en la ilustración 4.18, los algoritmos Isolation Forest, Elliptic Envelope y KMeans son los más adecuados debido a su equilibrio entre sensibilidad y precisión. Elliptic Envelope se distinguen por su habilidad para diferenciarse y formar clústeres claros. Por su equilibrio entre sensibilidad y precisión, los algoritmos Elliptic Envelope y KMeans son los más apropiados para detectar anomalías, mientras que su habilidad para diferenciarse y crear clústeres bien definidos es lo que distingue a KMeans.

Ilustración 4.19 Métricas de modelos Fuente: Elaboración propia.

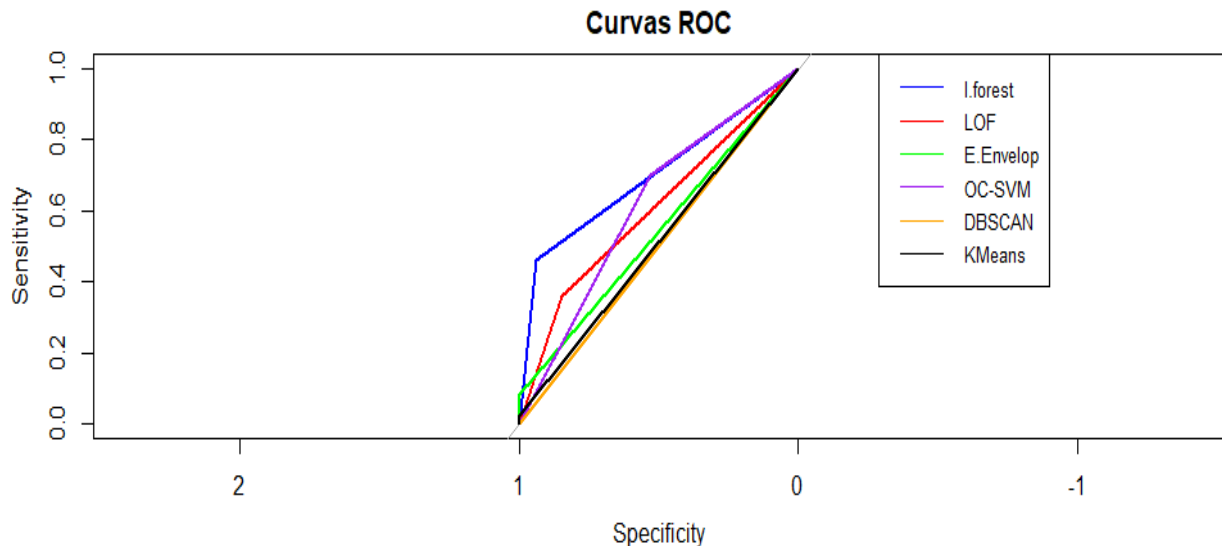


Fuente: Elaboración propia.

Asimismo, Isolation Forest en la ilustración 4.19 brinda un excelente rendimiento general, aunque su especificidad más elevada podría limitar su aplicación en algunos casos. Para este conjunto de datos, el modelo One Class SVM no es adecuada debido a su

desempeño significativamente inferior. Estos hallazgos ofrecen una base sólida para seleccionar modelos adecuados para detectar anomalías en aplicaciones similares.

Ilustración 4.20 Comparación de métricas. Fuente: Elaboración propia.



Fuente: Elaboración propia.

5. Discusión

Los resultados obtenidos en esta investigación demuestran que los algoritmos de detección de anomalías no supervisados, en particular el modelo de Isolation Forest, pueden mejorar significativamente la identificación de defectos en productos textiles en comparación con los métodos convencionales de inspección visual.

Los métodos tradicionales de inspección visual suelen depender de la experiencia y subjetividad de los operadores, lo que puede llevar a una detección inconsistente de defectos. En contraste, los algoritmos de IA desarrollados en este estudio han demostrado ser capaces de identificar defectos de manera más precisa y objetiva, lo que podría optimizar los procesos de control de calidad en la industria textil.

La implementación de estos modelos de detección de anomalías en entornos de producción textil podría automatizar y agilizar los procesos de inspección, reduciendo los costos asociados a la mano de obra y mejorando la consistencia en la identificación de defectos. Esto, a su vez, permitiría a las empresas del sector textil mejorar la calidad de sus productos y la satisfacción de los clientes.

Una limitación de este estudio es que el conjunto de datos utilizado se basó únicamente en defectos identificados mediante inspección visual, lo que puede introducir sesgos en los resultados. En el futuro, sería importante complementar estos datos con muestras de telas sin defectos, a fin de mejorar la capacidad de generalización de los modelos.

Además, explorar la combinación de varios algoritmos de detección de anomalías, como Isolation Forest y Elliptic Envelope, podría resultar en un desempeño aún más robusto y preciso para la identificación de defectos en textiles.

6. Conclusiones

El análisis y selección de variables demostró ser un paso esencial para determinar el mejor algoritmo en detección de anomalías, puesto que variables con mejor relación entre sí permitieron mejorar los modelos. Asimismo, fue importante determinar una estructura de entrenamiento y validación de los modelos mostrando así resultados más reales de la capacidad de los modelos en realizar predicciones acertadas con nuevos datos.

Se determinó la mejor estructuración de los modelos para obtener resultados de acuerdo al comportamiento de los datos que se verificaron con la validación de del 80% de los datos, destacando así la necesidad de mejorar el modelo con nuevos datos.

Esta investigación ha demostrado el potencial de los algoritmos de inteligencia artificial, específicamente los modelos de detección de anomalías no supervisados, para mejorar la identificación de defectos en productos textiles. Los resultados obtenidos con el modelo de Isolation Forest son particularmente prometedores y sugieren que la implementación de estas técnicas puede optimizar los procesos de control de calidad en la industria.

Referencias

- Adesh, A., G, S., Shetty, J., & Xu, L. (2024). Local outlier factor for anomaly detection in HPC systems. *Journal of Parallel and Distributed Computing*, 192, 104923. <https://doi.org/https://doi.org/10.1016/j.jpdc.2024.104923>
- Arciba, M. (2023). *Data Science aplicado en la industria acerera*. Universidad Autónoma de Chihuahua.
- Arnot, I. (2024). *Análisis y detección de anomalías usando técnicas de deep learning: Caso de estudio en cotizaciones de Pfizer moderna*. 2024.
- Aslam, B., Abbas, S., Ahmad, Y., Shah, S. T. H., & Raza, M. (2024). Interplay of emerging industrial technologies, ambidexterity, and sustainability: The case of the textile sector. *Sustainable Production and Consumption*, 49, 372–386. <https://doi.org/10.1016/j.spc.2024.07.006>
- Barbado, A., Corcho, Ó., & Benjamins, R. (2022). Rule extraction in unsupervised anomaly detection for model explainability: Application to OneClass SVM. *Expert Systems with Applications*, 189, 116100. <https://doi.org/10.1016/j.eswa.2021.116100>
- Barragán-Martínez, X. (2023). Situación de la Inteligencia Artificial en el Ecuador en relación con los países líderes de la región del Cono Sur. *FIGEMPA: Investigación y Desarrollo*, 16(2), 23–38. <https://doi.org/10.29166/revfig.v16i2.4498>
- Chater, M., Borgi, A., Slama, M. T., Sfar-Gandoura, K., & Landoulsi, M. I. (2022). Fuzzy Isolation Forest for Anomaly Detection. *Procedia Computer Science*, 207, 916–925. <https://doi.org/10.1016/j.procs.2022.09.147>
- Chen, Z., & Li, Y. F. (2011). Anomaly Detection Based on Enhanced DBScan Algorithm. *Procedia Engineering*, 15, 178–182. <https://doi.org/10.1016/j.proeng.2011.08.036>
- Ferreira, H. M., Carneiro, D. R., Guimarães, M. Â., & Oliveira, F. V. (2024). Supervised and unsupervised techniques in textile quality inspections. *Procedia Computer Science*, 232, 426–435. <https://doi.org/https://doi.org/10.1016/j.procs.2024.01.042>
- Ferro, F. F., Rampazzo, M., & Beghi, A. (2021). Elastic Shape Analysis for Anomaly Detection in Fabric Images. *IFAC-PapersOnLine*, 54(7), 67–72. <https://doi.org/https://doi.org/10.1016/j.ifacol.2021.08.336>
- Goldstein, M., & Uchida, S. (2016). A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data. *PLOS ONE*, 11(4), e0152173. <https://doi.org/10.1371/journal.pone.0152173>
- Ibujés Villacís, J. M., & Benavides Pazmiño, M. A. (2018). Contribución de la tecnología a la productividad de las pymes de la industria textil en Ecuador. *Cuadernos de Economía*, 41(115), 140–150. <https://doi.org/https://doi.org/10.1016/j.cesjef.2017.05.002>
- INEC. (2023). *Registro Estadístico de Empresas (REEM)*.

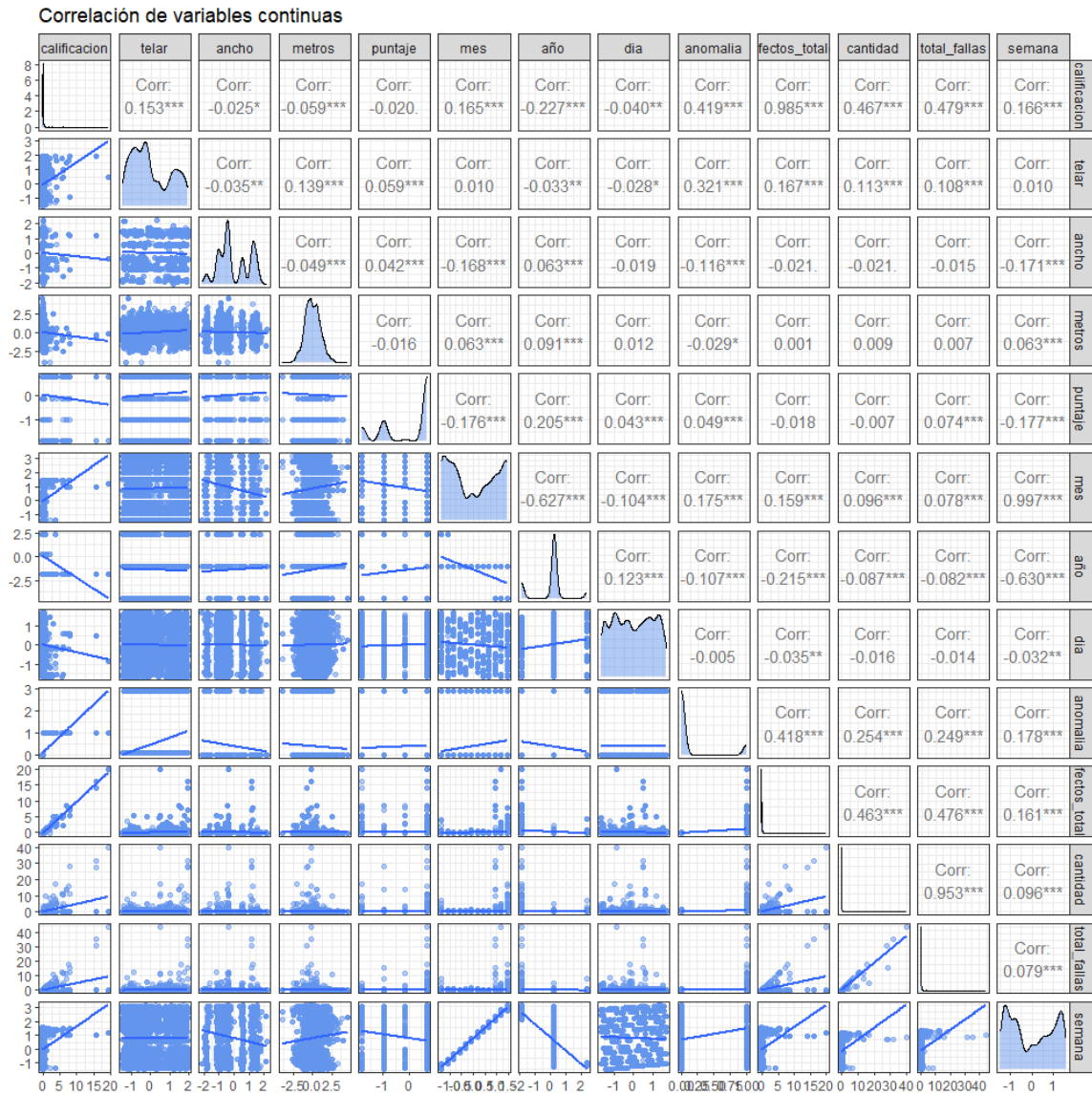
- Larios-Francia, R. P., & Ferasso, M. (2023). The relationship between innovation and performance in MSMEs: The case of the wearing apparel sector in emerging countries. *Journal of Open Innovation: Technology, Market, and Complexity*, 9(1), 100018. <https://doi.org/https://doi.org/10.1016/j.joitmc.2023.100018>
- Mehrotra, K. G., Mohan, C. K., & Huang, H. (2017). *Anomaly Detection* (pp. 21–32). https://doi.org/10.1007/978-3-319-67526-8_2
- Nassif, A. B., Talib, M. A., Nasir, Q., & Dakalbab, F. M. (2021). Machine Learning for Anomaly Detection: A Systematic Review. *IEEE Access*, 9, 78658–78700. <https://doi.org/10.1109/ACCESS.2021.3083060>
- Nowak - Brzezińska, A., & Horyń, C. (2020). Outliers in rules - the comparison of LOF, COF and KMEANS algorithms. *Procedia Computer Science*, 176, 1420–1429. <https://doi.org/10.1016/j.procs.2020.09.152>
- Orellana, M., & Cedillo, P. (2020). Detección de valores atípicos con técnicas de minería de datos y métodos estadísticos. *Enfoque UTE*, 11(1), 56–67. <https://doi.org/10.29019/enfoque.v11n1.584>
- Richardson, E., Trevizani, R., Greenbaum, J. A., Carter, H., Nielsen, M., & Peters, B. (2024). The receiver operating characteristic curve accurately assesses imbalanced datasets. *Patterns*, 5(6), 100994. <https://doi.org/10.1016/j.patter.2024.100994>
- Rincy, T., & Roopam, G. (2021). A Survey of Network Intrusion Detection Using Machine Learning Techniques. In H. Chiroma, S. M. Abdulhamid, P. Fournier-Viger, & N. M. Garcia (Eds.), *Machine Learning and Data Mining for Emerging Trend in Cyber Dynamics: Theories and Applications* (pp. 81–122). Springer International Publishing. https://doi.org/10.1007/978-3-030-66288-2_4
- Rouhiainen, L. (2018). *Inteligencia artificial: 101 cosas que debes saber hoy sobre nuestro futuro*. Alienta.
- Ruano, E., Montenegro, J., & León, R. (2019). Inteligencia artificial e innovación: campos de aplicación para la industria del Ecuador. *VISIÓN EMPRESARIAL*, N° 9, 1–10. <https://orcid.org/0000-0002-9441-0910>
- Trilles, S., Hammad, S. S., & Iskandaryan, D. (2024). Anomaly detection based on Artificial Intelligence of Things: A Systematic Literature Mapping. *Internet of Things*, 25, 101063. <https://doi.org/10.1016/j.iot.2024.101063>
- Tsai, D.-M., & Jen, P.-H. (2021). Autoencoder-based anomaly detection for surface defect inspection. *Advanced Engineering Informatics*, 48, 101272. <https://doi.org/https://doi.org/10.1016/j.aei.2021.101272>
- Valero, J., Navarro, Á., Larios, A., & Julca, J. (2022). Deserción universitaria: Evaluación de diferentes algoritmos de Machine Learning para su predicción. *Dialnet*, XXVIII(3), 362–375.

Vishwakarma, M., & Kesswani, N. (2023). A new two-phase intrusion detection system with Naïve Bayes machine learning for data classification and elliptic envelop method for anomaly detection. *Decision Analytics Journal*, 7, 100233. <https://doi.org/10.1016/j.dajour.2023.100233>

Vujovic, Ž. Đ. (2021). Classification Model Evaluation Metrics. *International Journal of Advanced Computer Science and Applications*, 12(6). <https://doi.org/10.14569/IJACSA.2021.0120670>

7. Anexos

Ilustración 7.1 Disposición de variables



Fuente: Elaboración propia.

Código de ejecución

Ilustración 7.2 Código de carga librerías

```
# Carga de las librerías
library(dplyr)
library(caret)
library(e1071)
library(ggplot2)
library(pROC)
library(cluster)
library(fpc)
library(rstatix)
library(reshape2)
library(knitr)
library(kableExtra)
library(mclust)
library(keras)
library(stats)
library(solitude)
library(isotree)
library(dbSCAN)
library(MASS)
library(lattice)
```

Fuente: Elaboración propia.

Ilustración 7.3 Depuración y selección de variables

```
# Cargar los datos
data <- read_excel("datos.xlsx")
data <- as.data.frame(data)

# Imputación con la mediana
data <- data %>%
  mutate(across(where(is.numeric), ~ifelse(is.na(.), median(., na.rm = TRUE), .)))

# Crear la variable de anomalía
data <- data %>% mutate(anomalía = ifelse(calidad == "Tela de Tercera", 1, 0))

# seleccionar solo variables numéricas
data_numeric <- data %>% dplyr::select(metros, ancho, cantidad, puntaje,
                                     total_fallas, defectos_totales, calificación)

# Escalado de variables
scaled_data <- scale(data_numeric)

# Crear un dataframe con los datos escalados y la variable de anomalía
data_scaled_df <- as.data.frame(scaled_data)
data_scaled_df$anomalía <- data$anomalía
```

Fuente: Elaboración propia.

Ilustración 7.4 División de datos en entrenamiento y validación, creación de función de almacenamiento de métricas

```
# Dividir en conjunto de entrenamiento y validación (80/20)
set.seed(123)
trainIndex <- createDataPartition(data$anomalia, p = .80, list = FALSE)
train_data <- data_numeric[trainIndex, ]
test_data <- data_numeric[-trainIndex, ]
train_labels <- data$anomalia[trainIndex]
test_labels <- data$anomalia[-trainIndex]

# Función para calcular métricas
calcular_metricas <- function(predictions, test_labels) {
  conf_matrix <- confusionMatrix(as.factor(predictions), as.factor(test_labels))
  roc_curve <- roc(test_labels, predictions)
  auc <- auc(roc_curve)
  chi_square <- chisq.test(table(predictions, test_labels))
  silhouette_score <- silhouette(predictions, dist(test_data))
  list(
    Accuracy = conf_matrix$overall["Accuracy"],
    sensitivity = conf_matrix$byClass["sensitivity"],
    specificity = conf_matrix$byClass["specificity"],
    Precision = conf_matrix$byClass["Precision"],
    F1 = conf_matrix$byClass["F1"],
    AUC = auc,
    chi_squared = chi_square$statistic,
    silhouette = mean(silhouette_score[, 3])
  )
}

# Resultados
resultados <- data.frame(Metrica = c("Accuracy", "Sensitivity", "specificity",
  "Precision", "F1", "AUC", "Chi-squared", "Silhouette"))
```

Fuente: Elaboración propia.

Ilustración 7.5 Entrenamiento de modelos parte 1

```
# Isolation Forest
model_if <- isolation.forest(train_data, ntree = 100)
scores_if <- predict(model_if, test_data)
threshold_if <- mean(scores_if) + sd(scores_if)
predictions_if <- ifelse(scores_if > threshold_if, 1, 0)
metricas_if <- calcular_metricas(predictions_if, test_labels)
resultados <- cbind(resultados, IsolationForest = unlist(metricas_if))

# Local Outlier Factor (LOF) utilizando el paquete dbscan
minPts <- 21
lof_train <- dbscan::lof(train_data, minPts = minPts)
lof_test <- dbscan::lof(test_data, minPts = minPts)
threshold_lof <- mean(lof_train) + sd(lof_train)
predictions_lof <- ifelse(lof_test > threshold_lof, 1, 0)
metricas_lof <- calcular_metricas(predictions_lof, test_labels)
resultados <- cbind(resultados, LOF = unlist(metricas_lof))

# Elliptic Envelope
model_ee <- cov.rob(train_data)
mahal_dist <- mahalanobis(test_data, colMeans(train_data), cov(train_data))
threshold_ee <- qchisq(0.975, df = ncol(train_data))
predictions_ee <- ifelse(mahal_dist > threshold_ee, 1, 0)
metricas_ee <- calcular_metricas(predictions_ee, test_labels)
resultados <- cbind(resultados, EllipticEnvelope = unlist(metricas_ee))
```

Fuente: Elaboración propia.

Ilustración 7.6 Entrenamiento de modelos parte 2

```
# One-class SVM
model_svm <- svm(train_data, type = 'one-classification', kernel = 'radial')
predictions_svm <- predict(model_svm, test_data)
predictions_svm <- ifelse(predictions_svm == TRUE, 1, 0)
metricas_svm <- calcular_metricas(predictions_svm, test_labels)
resultados <- cbind(resultados, oneClassSVM = unlist(metricas_svm))

# DBSCAN
db <- dbscan(train_data, eps = 0.5, minPts = 5)
test_clusters <- predict(db, newdata = test_data, data = train_data)
predictions_db <- ifelse(test_clusters == 0, 1, 0)
metricas_db <- calcular_metricas(predictions_db, test_labels)
resultados <- cbind(resultados, DBSCAN = unlist(metricas_db))

# K-means
model_km <- kmeans(train_data, centers = 2)
test_clusters_km <- apply(test_data, 1, function(x) {
  which.min(colsums((t(model_km$centers) - x)^2))
})
predictions_km <- ifelse(test_clusters_km == which.min(table(model_km$cluster)), 1, 0)
metricas_km <- calcular_metricas(predictions_km, test_labels)
resultados <- cbind(resultados, KMeans = unlist(metricas_km))
```

Fuente: Elaboración propia.

Ilustración 7.7 Código de los resultados de los modelos

```
# Presentar los resultados
kable(resultados, format = "html") %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive"))
# Especificar el nombre del archivo CSV
archivo_csv <- "resultados.csv"
# Guardar el dataframe de resultados en un archivo CSV
write.csv(resultados, archivo_csv, row.names = FALSE)
# Mensaje de confirmación
cat("Resultados guardados en", archivo_csv, "\n")
# Visualización de resultados
# Excluir la fila con "chi-cuadrado"
resultados <- resultados[resultados$Metrica != "Chi-squared", ]
resultados <- resultados[resultados$Metrica != "silhouette", ]
# Convertir los resultados a formato largo para ggplot
results_long <- resultados %>%
  pivot_longer(-Metrica, names_to = "Modelo", values_to = "valor")
# Gráfico de las métricas
ggplot(results_long, aes(x = Modelo, y = valor, fill = Metrica)) +
  geom_bar(stat = "identity", position = position_dodge()) +
  theme_minimal() +
  labs(title = "Comparación de Métricas de Rendimiento", y = "valor", x = "Modelo") +
  theme(plot.title = element_text(hjust = 0.5))
```

Fuente: Elaboración propia.

Ilustración 7.8 Código gráfico tipo radar de métricas

```
# Convertir los datos en el formato adecuado para fmsb

metrics_radar <- as.data.frame(t(metrics[-1]))
colnames(metrics_radar) <- metrics$Metric
metrics_radar <- rbind(rep(1, ncol(metrics_radar)), rep(0, ncol(metrics_radar)), metrics_radar)

# Crear gráfico radar
library(fmsb)
radarchart(metrics_radar,
  axistype = 1,
  pcol = c("blue", "red", "green", "purple", "orange", "brown"),
  #pfcol = alpha(c("blue", "red", "green", "purple", "orange", "brown"), 0.5),
  plwd = 2,
  cglcol = "black",
  cglty = 1,
  axislabels = c("0", "0.5", "1"),
  vlce = 0.8,
  title = "Comparación de Métricas de Modelos"
)
legend("topright", legend = colnames(resultados)[-c(1, 2)],
  col = c("blue", "red", "green", "purple", "orange", "brown"), pch = 22, cex = 0.8)
```

Fuente: Elaboración propia.