UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA DE MINAS

MODELO PREDICTIVO DE FALLAS PARA CHANCADORES DE LA PLANTA DE CHANCADO SECUNDARIO Y TERCIARIO DE DIVISIÓN CHUQUICAMATA DE CODELCO

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL DE MINAS

FRANCISCO FELIPE VILCHES MEDINA

PROFESOR GUÍA:
YERKO FERNÁNDEZ ÁNGEL

MIEMBROS DE LA COMISIÓN:
NADIA MERY GUERRERO
GONZALO MONTES ATENAS

SANTIAGO DE CHILE
2023

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE INGENIERO CIVIL DE MINAS
POR: FRANCISCO FELIPE VILCHES MEDINA
FECHA: 2023
PROF. GUÍA: YERKO FERNÁNDEZ ÁNGEL.

MODELO PREDICTIVO DE FALLAS PARA CHANCADORES DE LA PLANTA DE CHANCADO SECUNDARIO Y TERCIARIO DE DIVISIÓN CHUQUICAMATA DE CODELCO

A partir del uso de algoritmos de *Machine Learning* se busca crear un modelo predictivo de fallas para los chancadores de la planta de chancado secundario y terciario de la división Chuquicamata de Codelco. Para determinar el momento en que fallarán los chancadores se utilizan los algoritmos de regresión logística, Naïve-Bayes y K vecinos más cercanos (K-NN) empleando como variable independiente sólo los Tiempo entre Falla (TEF) que se calculan a partir de una base de datos entregada por la Superintendencia de Mantenimiento de Chuquicamata, Codelco. La base de datos es analizada y purificada para obtener distribuciones de TEF fiables lo que se comprueba verificando que las distribuciones obtenidas satisfacen los tests de bondad de ajuste de Kolmogorov-Smirnov, Anderson-Darling y χ^2 (chi-cuadrado) imponiendo que estas distribuciones corresponden a distribuciones Weibull, por otro lado, se verifica que estas distribuciones siguen una distribución Weibull utilizando el método de mínimos cuadrados y analizando su coeficiente de determinación (R^2). Para las tareas anteriores se utiliza el complemento *Easy Fit* y una planilla *Excel* facilitada por la misma Superintendencia de Mantenimiento. Se comprueba que a un nivel de confianza del 2 % los tests de bondad nunca son rechazados, además, el coeficiente de determinación es siempre mayor o igual a 0,92. Los resultados de las técnicas de *Machine Learning* indican que la exactitud promedio al considerar los 5 chancadores secundarios y 10 chancadores terciaros son de 83,4 % para la regresión logística, 80,8 % para el Naïve-Bayes y 78,1 % para el K-NN. En cuanto a la precisión a la hora de determinar fallas estas corresponden a 91,7 %, 90,2 % y 83,1 % respectivamente. Los mejores resultados se observan en el chancador secundario MP1000, el chancador terciario 8 y el chancador terciario 9 donde las precisiones fluctúan entre un 90 % y 95 % para todos los algoritmos, por el contrario, para el chancador secundario Symons 7" de la sección B de la planta, el algoritmo K-NN muestra una precisión de tan sólo 48 %. Con el objetivo de determinar el "cómo" va a fallar el chancador se realiza un *Random Forest* donde se utilizan los modos de falla como única variable independiente. Debido a las disímiles frecuencias de los modos de falla se utilizan técnicas de balanceo de datos, junto a lo anterior se utilizan técnicas para optimizar los hiper-parámetros del modelo. Al utilizar la información del chancador MP1000 se alcanza una exactitud del 58 % lo que se puede interpretar como un discreto resultado el cual se explica por la utilización de tan sólo una variable independiente, sin embargo, los resultados del *Random Forest* brindan información para realizar una pauta de inspección de equipos enfocándose en las fallas por el sistema de lubricación que es la de mayor frecuencia y el modo de falla entregado por el propio modelo de *Random Forest*. Finalmente, se sugiere realizar un análisis económico para determinar los beneficios de considerar los resultados entregados y para determinar el momento óptimo de realización de mantenciones preventivas y su periodicidad.

ABSTRACT OF THE THESIS TO OBTAIN '
THE GRADE OF MINING ENGINEER
BY: FRANCISCO FELIPE VILCHES MEDINA
DATE: 2023
THESIS ADVISOR: YERKO FERNÁNDEZ ÁNGEL

PREDICTIVE FAILURE MODEL FOR CRUSHERS OF THE DIVISION SECONDARY AND TERTIARY CRUSHING PLANT CHUQUICAMATA FROM CODELCO

Based on the use of *Machine Learning* algorithms, the aim is to create a predictive failure model for the crushers of the secondary and tertiary crushing plant of the Chuquicamata mine, Codelco. Logistic regression, Naïve-Bayes and K-NN algorithms are used to determine the time when the crushers will fail, using only the Time Between Failures (TBF) as the independent variable that are calculated from a database provided by the Maintenance Superintendence of Chuquicamata, Codelco. The database is analyzed and purified to obtain reliable TEF distributions, which is verified by verifying that the obtained distributions satisfy the Kolmogorov-Smirnov, Anderson-Darling and χ^2 (chi-square) goodness-of-fit tests, imposing that these distributions correspond to Weibull distributions, on the other hand, it is verified that these distributions follow a Weibull distribution using the method of least squares and analyzing its coefficient of determination (R^2). For the above tasks, the *Easy Fit* complement and a *Excel* spreadsheet provided by the Maintenance Superintendence itself are used. It is verified that at a confidence level of 2 % the goodness tests are never rejected, furthermore, the coefficient of determination is always greater than or equal to 0.92. The results of the *Machine Learning* techniques indicate that the average accuracy when considering the 5 secondary crushers and 10 tertiary crushers are 83.4 % for the logistic regression, 80.8 % for the Naïve-Bayes and 78.1 % for the K-NN. In terms of precision for determining when the crushers will fail, these correspond to 91.7 % and 90.2 % and 83.1 % respectively. The best results are observed in the secondary crusher MP1000, the tertiary crusher 8 and the tertiary crusher 9 where the accuracies fluctuate between 90 % and 95 % for all the algorithms, on the contrary, the most discreet results are those that are obtained for the Symons 7" secondary crusher in section B of the plant where the K-NN algorithm shows an accuracy of only 48 %. In order to define how the crusher will fail, a *Random Forest* is performed where the failure modes are used as the only independent variable. Due to the dissimilar frequencies of the failure modes, the SMOTE technique is used to balance the data, together to the above, a "grid-searchïs used to optimize the hyper-parameters of the model. When using the information from the MP1000 crusher, an accuracy of 58 % is reached, which can be interpreted as a discrete result which is explained for the use of only an independent variable, however, the results of the *Random Forest* provide information to carry out an equipment inspection guideline focusing on failures due to the lubrication system, which is the most frequent, and the failure mode delivered by the *Random Forest* model itself. Finally, it is suggested to perform an economic analysis to determine the economic profits (maintenance cost savings) of considering the results delivered and to determine the optimal time to perform preventive maintenance and its periodicity.

Quien espera paciente
y prudentemente será
recompensado en el
momento adecuado.

...

T.L.

Agradecimientos

En primer lugar quiero agradecer a mi familia por el apoyo permanente que he recibido a lo largo de mi estadía en la universidad. En particular a mi mamá, mi hermana, a la Coquita, al Zippito, al Luckito, al Gatón, a mis tíos, mi abuela y también a mi abuelita y abuelo que están en el cielo junto a la Coquita y al Gatón.

A partir de ahora mencionaré a todas las personas que deseo agradecer en orden alfabético para que no sientan ninguna diferencia en caso de que algún día lean estas palabras. Quiero agradecer a mis amigos de toda la vida: Francisco Espinoza, Javier Soto, Sebastián Sáez, y tambien a todos los amigos que hice durante mi paso por la universidad con los que viví gratos y muchos inolvidables momentos: Álvaro Mardones, Bastián Urrutia, Bayron Navarrete, Darío Toro, Elías Riveros, Francisco Arévalo, Ignacio Henríquez, Madeline Valdivia, Matías Muñoz, Moisés Miranda, Nicolás Lepiman, Sebastián Caballero. Pido las disculpas correspondientes en caso de haber olvidado citar a alguno.

Quiero agradecer también a todos los donantes de sangre que aparecieron a fines del 2021 e inicios del 2022 dado que viví complicados momentos de salud mientras realizaba este trabajo de memoria. Fueron bastantes y por eso no los puedo nombrar debido a que como dijo Pierre de Fermat este margen no me alcanza para contenerlos.

Finalmente quiero agradecer a la Universidad de Chile por haberse convertido realmente en mi segundo hogar en todos los sentidos en los que se puede interpretar esta palabra y también a Codelco por permitirme realizar este trabajo de memoria en una de sus faenas, en especial a mi tutor de memoria Don Yerko Fernández que tuvo la mejor de las disposiciones para enseñarme y comprender la situación que viví en mis meses en Calama. Mis agradecimientos también van a mi comisión, la profesora Nadia Mery y el profesor Gonzalo Montes por haber aceptado la responsabilidad de guiar mi trabajo.

Gracias totales!!!

Tabla de Contenido

Índice de Tablas

Índice de Ilustraciones

Capítulo 1

Introducción

1.1. Motivacion

La mantención de los equipos que operan en la producción de las plantas de minería es una actividad crítica en el proceso productivo minero, debido a que éstos son requeridos en forma intensiva, por este motivo la mantención resulta ser un aspecto fundamental en la actividad minera. Los programas de mantención deben considerar la disminución de los tiempos de detención y la seguridad de los trabajadores que deben realizar esta tarea con equipos complejos y en gran cantidad de veces de gran tamaño. A lo anterior se añade el escenario actual, donde las compañías mineras están focalizadas en mejorar productividad y menores costos en todo lo que involucra su operación.

El complejo entorno vinculado a la alta volatilidad en el precio del cobre, la disminución en la ley media de los yacimientos y las cada vez más exigentes condiciones operacionales, obligan a mejorar la confiabilidad y el desempeño de los activos/equipos con el objetivo de dar cumplimiento a los planes de producción establecidos por la corporación.

Una baja confiabilidad de los activos conlleva a eventos no programados, lo cual impacta negativamente en la producción al no encontrarse los equipos disponibles para ser operados, sumado al hecho de que una mantención de equipo correctiva implica un desembolso de capital aproximadamente 5 veces mayor al de una mantención programada.

La planta de chancado secundario y terciario de la división Chuquicamata de Codelco actualmente cuenta con un plan de mantenimiento programado para evitar la avería de equipos durante la operación. Lo anterior implica que cada uno de los equipos deja de estar disponible en la operación en determinado día del mes con el objetivo de evitar fallas mayores que impacten en los costos y en la producción, sin embargo, puede existir el caso donde el mantenimiento programado sea incluso innecesario.

Por otro lado, el *Machine Learning* puede ser definido como un método de análisis de datos que automatiza la construcción de modelos analíticos. El resurgimiento del interés en el aprendizaje basado en máquina (*Machine Learning*) se debe a los mismos factores que

han hecho a la minería de datos y el análisis Bayesiano más populares que nunca, es decir volúmenes y variedades crecientes de datos disponibles, procesamiento computacional más económico y poderoso, y almacenaje de datos cada vez más asequible. Lo anterior implica que es posible producir modelos de modo rápido y automático capaces de analizar datos más grandes y complejos y así producir resultados más rápidos y precisos (incluso en escalas muy grandes). Un buen modelo permite a una organización identificar oportunidades rentables como también evitar riesgos desconocidos.

Con la intención de mejorar los KPI (indicadores clave de rendimiento), es decir, aumentar tanto la disponibilidad como la utilización de los chancadores, y junto a ello ahorrar costos en mantenimiento (los que pueden incluso llegar a un 60 % de los costos operacionales de una mina) se propone utilizar técnicas de *Machine Learning* con la finalidad de predecir los momentos en que los chancadores fallarán en la planta. Lo anterior es posible debido a bases de datos creadas por la Gerencia de Mantenimiento de Concentradora de la División Chuquicamata de Codelco (DCH) donde se registra información concerniente a los inconvenientes que sufre cada equipo. El análisis de estas bases de datos junto a los algoritmos propios del *Machine Learning* permite dar un acercamiento probabilístico al concepto de falla de equipos.

1.2. Objetivos

1.2.1. General

- Diseñar y evaluar un modelo supervisado en el ámbito de *Machine Learning* para la predicción de fallas en chancadores de la planta de chancado secundario/terciario en división Chuquicamata, Codelco.

1.2.2. Especificos

- Analizar base de datos proporcionada para identificar fallas y eliminar redundacias.
- Proponer algoritmos de *Machine Learning* supervisado para predicción de fallas.
- Crear set de datos a utilizar por los algoritmos de *Machine Learning*.
- Determinar la exactitud y precisión de los modelos utilizados y comparar resultados obtenidos.
- Realizar un *Random Forest* para un chancador crítico de la planta para predecir el modo de falla del equipo.

1.3. Alcances

El modelo supervisado de *Machine Learning* se realiza sólo para chancadores pertenencientes a la planta de chancado secundario y terciario de la División Chuquicamata de Codelco. Lo

anterior incluye: chancadores secundarios y chancadores terciarios.

Para realizar el modelo supervisado se utiliza una base de datos de registro de fallas recopilada desde el 01 de enero de 2019 hasta el 30 de septiembre de 2021. Debido a la base de datos utilizada, y a las condiciones propias de la planta de chancado secundario y terciario, los resultados obtenidos no son replicables a plantas de procesamiento de otras divisiones ni las técnicas empleadas resultan ser válidas para predecir fallas de chancadores en otro tipo de planta.

Se examinan como modelos supervisados la Regresión Logística, Naïve-Bayes y k-NN ($K - nearest\ neighbors$) y se utiliza como variable independiente únicamente los Tiempo entre Fallas (TEF). Los equipos analizados corresponden a 5 chancadores secundarios y 10 chancadores terciarios.

El modelo supervisado creado asume todo tipo de mantenciones realizadas durante el perìodo donde se registran fallas en la base de datos. Además, a partir del uso de probabilidades, determina la existencia o no de falla de algún equipo en determinada hora o día de la operación.

El *Random Forest* utiliza como variable independiente sólamente los modos de falla del equipo. El objetivo del problema es generar una pauta de inspección de chancadores que permita enfocar los recursos en las más probables causas de falla de los equipos.

1.4. Estructura

- En el segundo capítulo se entregan antecedentes de la planta de chancado secundario y terciario de la División Chuquicamata de Codelco y se realiza un marco teórico para comprender principalmente los fundamentos del mantenimiento de equipos, de la teoría de confiabilidad, de distribuciones Weibull y de modelos de *Machine Learning*.
- En el tercer capítulo se explica en detalle la metodología utilizada para obtener los resultados que entrega cada uno de los algoritmos a utilizar.
- En el cuarto capítulo se presentan los resultados de cada modelo para cada uno de los chancadores de la planta.
- En el quinto capítulo se analizan los resultados obtenidos.
- En el sexto capítulo se realizan las conclusiones del trabajo de memoria.
- En Anexos se entregan los códigos en *Python* utilizados.

Capítulo 2

Marco Teórico

2.1. Generalidades

El yacimiento Chuquicamata es conocido desde el siglo XIX, este corresponde al principal depósito de cobre y molibdeno de la División Codelco Norte, ubicado en la precordillera del Norte de Chile. Durante los primeros años fueron explotados óxidos de alta ley como también vetas de óxido por pequeños mineros, llegado el año 1915 se inicia la explotación de los óxidos a escala industrial dando origen al rajo Chuquicamata. Al día de hoy se cuenta con dos métodos de explotación para el mismo depósito Chuquicamata, el rajo, cuya progresiva profundización y expansión lateral ha resultado en un incremento en la relación estéril : mineral y también la mina subterránea (abreviada CHS) que actualmente trabaja con un nivel de producción y tiene contemplado trabajar con 3 niveles a medida que se desarrolla la obra. CHS se explota por medio de macro bloques mediante el proceso de extracción de 'Block Caving'. CHS tiene cuantificadas unas reservas cercanas a 1.700 millones de toneladas de cobre con una ley promedio de 0,7 % de cobre y de 502 ppm (partes por millón) de molibdeno. En estos momentos CHS se encuentra trabajando para poder llegar lo antes posible a régimen (Tasa de producción de 140.000 [tpd]). En cuanto al rajo, este se encuentra actualmente en Fase de cierre, se le proyecta una vida útil de 5 años más, se ha decidido desde hace algunos años (2017) aumentar el ángulo de Talud de la pared Norte-Este para así reducir la relación estéril : mineral, este ángulo se planea aumentar en 3°, pasando de los 55° actuales a 58°, este aumento en el ángulo de talud permite extraer más mineral y de mejor ley, además pemite ahorrar u optimizar el desarrollo en 13 millones de toneladas de material que ya no será necesario extraer, teóricamente esto permite aumentar la ley de cobre de 14 millones de toneladas de sulfuros de cobre al llegar a una ley de 0,83 %, mientras que antes del plan implementado, se estimaba una ley del 0,79 % para este material.

Figura 2.1: A la izquierda el rajo Chuquicamata, a la derecha Chuquicamata subterránea, Fuente: Sitio web de Codelco.

2.2. Ubicación

El yacimiento Chuquicamata se conoce como el principal depósito del Distrito Zona Norte de CODELCO. Se ubica en la precordillera de la Región de Antofagasta, Provincia del Loa, a cerca de 13 [km] al Norte de la ciudad de Calama, a 232 [km] al Noroeste de la ciudad de Antofagasta y a 1.650 [km] al Norte de Santiago (capital de la República de Chile). Sus coordenadas geográficas son 22,27° de latitud Sur y 68,54° de longitud Oeste y su altitud media corresponde a 2.870 [m.s.n.m].

Figura 2.2: Ubicación Mina Chuquicamata de Codelco. Fuente: Codelco, Vicepresidencia corporativa de proyectos (2009).

2.3. Antecedentes planta de chancado secundario y terciario

Previo al proceso de molienda y flotación que se realizan en las plantas concentradoras, el mineral debe pasar por un proceso de chancado secundario y terciario. El área de la planta de chancado secundario y terciario de la división Chuquicamata de Codelco recibe el mineral proveniente del chancador primario, el cual es chancado (capacidad de 104 [ktpd] húmedas) para posteriormente ser transportado hacia los molinos existentes en las 3 plantas concentradoras de la división.

La figura 2.3 entrega un diagrama de la planta de chancado secundario y terciario de la división Chuquicamata de Codelco.

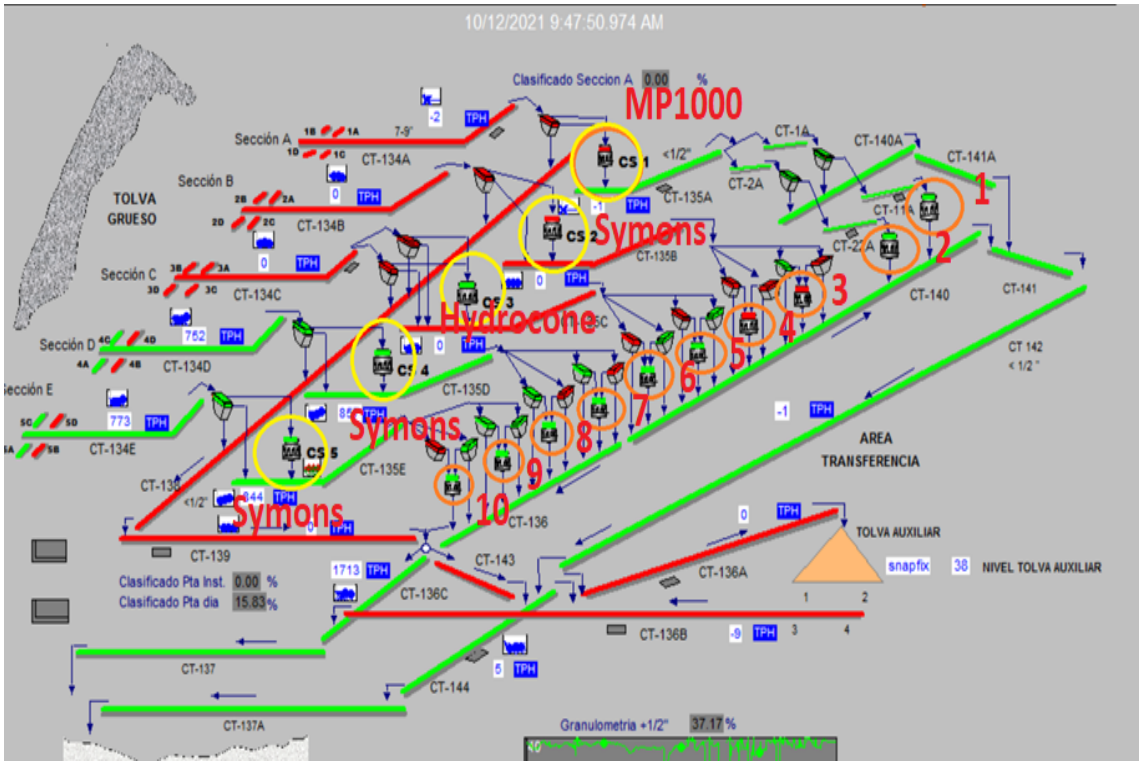


Figura 2.3: Planta de chancado secundario y terciario de división Chuquicamata de Codelco. Fuente: Programa Sip Mobile.

Encerrados en círculos amarillos en la figura 2.3 se encuentran los chancadores secundarios de la planta. Se cuenta con 5 de ellos. La sección A de chancado secundario utiliza un chancador de cono *Nordberg* MP1000 con un rango de capacidad de 615-2420 [tph]. La figura 2.4 muestra el chancador de cono *Nordberg* MP1000.

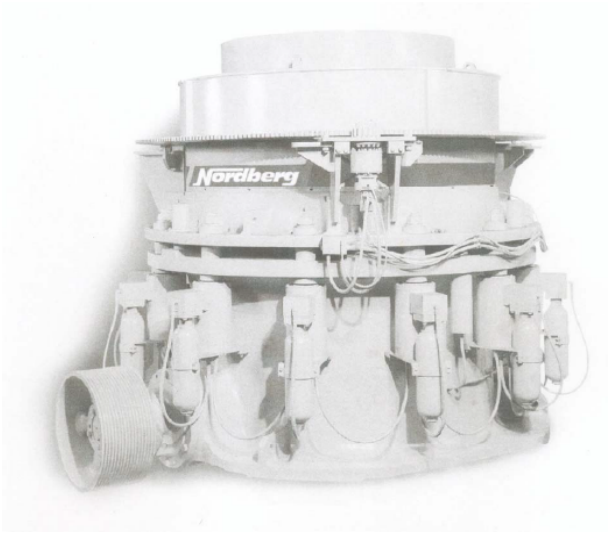


Figura 2.4: Chancador de cono utilizado en la sección A de la planta de chancado secundario y terciario. Fuente: Manual de instrucciones, Chancador de conos Nordberg de la Serie MP, Metso Minerals.

Las secciones B, D y E de la línea de chancado secundario utilizan chancadores de cono *Symons* de 7' de cabeza estándar cuya capacidad varía entre 789 [tph] y 1257 [tph] como el que se muestra en la figura 2.5.

Figura 2.5: Chancador de cono utilizado en las secciones B, D y E de la planta de chancado secundario y terciario. Fuente: Manual de servicio, Chancador de Cono Symons $4\frac{1}{4}'$, $5\frac{1}{2}'$ & $7'$.

Las sección C de la línea de chancado secundario utiliza un chancador Hydrocone H8800 de *Sandvik* que cuenta con una capacidad máxima de 350 [tph] como el que se muestra en la figura 2.6.

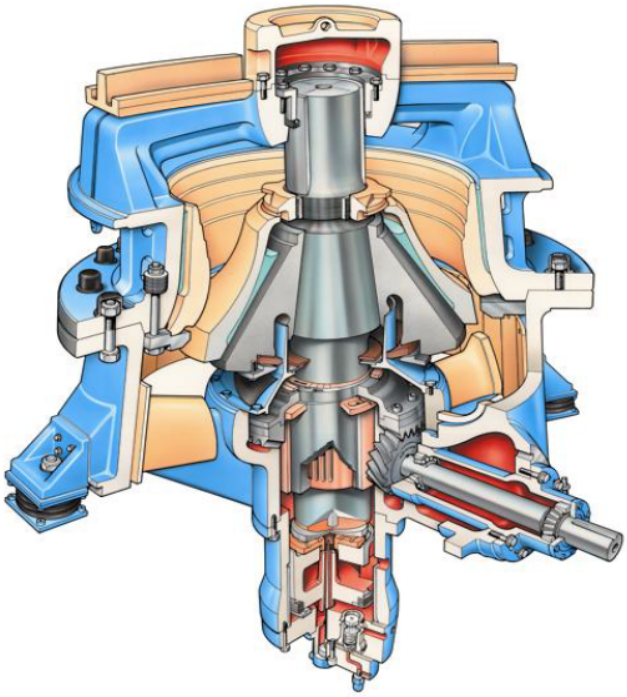


Figura 2.6: Chancador de cono utilizado en la sección C de la planta de chancado secundario y terciario. Fuente: H8800 Instrucción de mantenimiento con lista de piezas de recambio.

Vale decir que cada chancador secundario se comunica mediante el uso de correas transportadoras a 2 chancadores terciarios Symons 7' SH (short head), por tanto, la planta cuenta con 10 de este tipo de chancadores.

Esta área cuenta con tres stockpiles para el almacenamiento de minerales:

- Tolva para mineral grueso procedente de las trituradoras primarias (E4 y F3). Su capacidad nominal es de 35.000 [ton].
- Tolva auxiliar para el almacenamiento del producto de trituración 2 º / 3 º. Su capacidad nominal es de 33.000 [ton].
- Tolva convencional para el almacenamiento del producto de trituración 2 º / 3 º que se alimenta de las secciones de molienda de la A-0 y A-1. Su capacidad nominal es de 28.000 [ton].

2.4. Mantenimiento de equipos

Según Torres (2007), el término mantenimiento incluye varias acciones y tareas que tienden a incrementar o mantener la fiabilidad de las máquinas. Un término ampliamente presentado

corresponde a la llamada política de mantenimiento, que hace alusión a las categorías de acciones de mantenimiento aplicadas a los equipos. Dependiendo del tipo de industria (en particular minería) y al tipo de paradigma de mantenimiento empleado, los términos de estas categorías varían. Dichos términos incluyen al mantenimiento basado en estados, al mantenimiento basado en fallas, al mantenimiento preventivo, al mantenimiento predictivo, al mantenimiento reactivo, al mantenimiento correctivo y el mantenimiento oportunista. Todas estas categorías hacen referencia a la programación o metodología de la gestión de mantenimiento.

La principal categorización la brindan las actividades de mantenimiento que ocurren antes y después de la falla. Luego, se pueden utilizar los términos de mantenimiento proactivo y mantenimiento reactivo. Los tipos de mantenimiento proactivo y reactivo pueden ser a su vez planificados o no planificados. Lo anterior indica que los procedimientos de trabajo de la acción de mantenimiento y los recursos que esta requiere pueden planificarse como no planificarse previo a la necesidad de ejecutar una acción de mantenimiento. Una acción de mantenimiento realizada para a una falla que nunca antes había sucedido es compleja de ser planificada con anticipación. Por tanto, el mantenimiento reactivo es usualmente pensado como un sinónimo de mantenimiento no planificado. Por otro lado, la planificación del mantenimiento reactivo puede ser llevada adelante si los procedimientos de mantenimiento son conocidos.

Una acción de mantenimiento proactivo puede ser del tipo preventivo como predictivo. El mantenimiento preventivo intenta prevenir la falla antes de su ocurrencia con actividades tales como la lubricación, limpieza y cambio de componentes desgastados en una máquina. Por otro lado, el mantenimiento predictivo, también conocido como mantenimiento basado en estados, intenta detectar la condición de la máquina, de modo automático o manual, para ejecutar la acción de mantenimiento basada en el estado actual de la máquina. El mantenimiento programado es una acción de mantenimiento proactivo que ha sido programada de antemano según un plan. El mantenimiento proactivo puede ser programado, pero el mantenimiento reactivo jamás puede ser programado con anticipación.

Por otro lado, según González (2005), el mantenimiento durante el siglo XX se puede dividir en 3 grandes etapas que abarcan todos los medios y objetivos de los tipos de mantenimiento mencionados previamente. Estas etapas muestran cómo ha sido la evolución de las técnicas y organizaciones que se han ido implementando con el paso del tiempo. La figura 2.7 resume la evolución de las técnicas de mantenimiento durante el siglo XX.

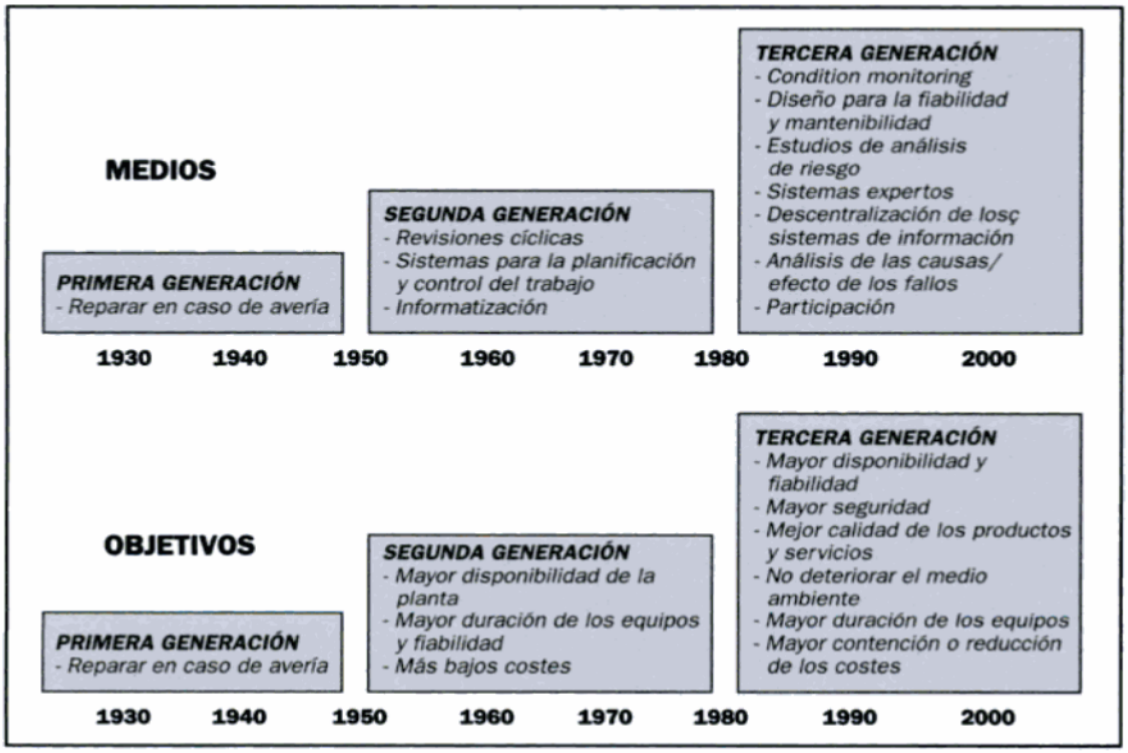


Figura 2.7: Evolución de los tipos de mantenimiento durante el siglo XX. Fuente: González (2005).

2.5. Teoría de la confibialidad

Según Torres (2007), la ingeniería de confiabilidad es una rama de las matemáticas y de la ciencia de diseño de máquinas bastante bien establecida y conocida.

La relación entre ingeniería de confiabilidad y mantenimiento está definida por la "Asociación Americana de Transporte Aéreo" en la nota adicional vinculada a los objetivos del programa de mantenimiento. Estos objetivos admiten que los programas de mantenimiento, como tales, no pueden corregir deficiencias en la seguridad inherente y los niveles de confiabilidad de los equipos. El programa de mantenimiento puede sólamente prevenir el deterioro de dichos niveles inherentes. Si los niveles inherentes son encontrados poco satisfactorios, modificaciones en el diseño son necesarias para obtener mejoras.

El mantenimiento no puede mejorar la capacidad ni la confiabilidad de una máquina por encima de los niveles inherentes de esta. Entonces, el diseño de la máquina así como los materiales empleados son los factores más significativos en el momento de determinar la máxima confiabilidad de los sistemas.

Como una metodología, la ingeniería de confiabilidad se enfoca en la identificación de las causas, probabilidades y consecuencias de las fallas para planificar las acciones relevantes para reducirlas. Esto se lleva a cabo gracias a la modelación de sistemas físicos y sus características de confiabilidad y falla con modelos matemáticos o mediante el uso de análisis

de decisión de mayor calidad. Los sistemas físicos son usualmente modelados con redes de confiabilidad o árboles de falla, los cuales describen la relación entre los componentes y cómo las fallas de un componente afectan la operación de otros componentes. La ocurrencia de falla y la probabilidad de duración de los sistemas y componentes son modelados mediante el uso de distribuciones de probabilidad, como la de Weibull, exponencial y distribución Gamma entre otras. Mediante el uso de estos modelos es posible estimar el tiempo de vida y las probabilidades de falla de componentes y de los sistemas que comprenden estos componentes.

La ingeniería de confiabilidad hace algunas suposiciones respecto a las fallas. La definición formal de la palabra "falla" es usualmente excluida de la literatura de ingeniería de confiabilidad y de los modelos matemáticos. Con frecuencia, las fallas son consideradas como eventos binarios que ocurren estocásticamente durante el ciclo de vida de la máquina. La máquina puede estar o no en estado de falla. Las metodologías de la ingeniería de confiabilidad raramente presentan metodologías para manipular fallas parciales. La razón principal es aparentemente que resulta difícil modelar la propagación de fallas parciales hacia otras fallas parciales en el sistema de interés.

Como una ciencia matemática, el interés radica en los modelos de falla y su aplicabilidad a diferentes sistemas. Ya que la aproximación es probabilística y se enfoca en la estimación de la confiabilidad promedio o asintótica de los sistemas, este enfoque es utilizado en la actualidad por fabricantes de máquinas que tienen interés en incrementar el mantenimiento y la inherente confiabilidad de sus productos.

2.6. Términos importantes

Según Torres (2007):

2.6.1. Fallas en Mantenimiento:

Según Mancuzo (2020), las fallas en el mantenimiento son eventos inesperados que implican el funcionamiento defectuoso o el propio cese en las funciones de los equipos, lo que se traduce directamente en la merma de la productividad de una empresa.

Los tipos de fallas más comunes pueden ser evitadas con un plan de mantenimiento adecuado basado en acciones preventivas y la ayuda de un Software de Mantenimiento que permita monitorear y prever todo tipo de riesgos en los equipos.

Nunca se puede anticipar el nivel de gravedad de una falla inesperada: la falla puede generar días de nula productividad e incluso posibles accidentes laborales con operarios e insumos.

2.6.2. Estados de falla:

Según Torres (2007), la teoría de sistemas indica que una función de sistema se encuentra definida por sus estados y los estados de cambio. Esto se relaciona con la idea del mantenimiento basado en la confiabilidad (RCM) y el mantenimiento productivo total (TPM),

donde las fallas son estados de sistema. Luego, se asume que la falla es un estado. A modo de ejemplo, el estado de falla $S_f \in S$, donde $S = \{S_0, S_1, \ldots, S_n\}$ es un conjunto de posibles estados del sistema.

En la práctica, un sistema de producción es un sistema diseñado por el hombre con un propósito. Por tanto, el estado de falla siempre es definido por el usuario. Un sistema de producción puede contar con mùltiples usuarios, cada uno con diferentes ideas y actitudes que incumben al propósito del sistema. Por ende, el usuario es meramente un rol de una organización, y la definición de fallas no son necesariamente específicas de una persona sino más bien específicas de un rol.

2.6.3. Teoría de Fallas:

Según Salazar et al. (2005), las fallas ocurren de modo incierto y se ven influenciadas por el diseño, manufactura o construcción, mantenimiento y operación, como también por factores humanos, algunas de ellas pueden ser catastróficas significando incluso pérdidas humanas. No existe manera en que las fallas puedan ser erradicadas del todo, por ende, cualquier producto u objeto fallará independiente de qué tan bien haya sido diseñado, sin embargo, se puede llegar a reducir la tasa de incidencia de fallas dentro de cierto límite de tiempo, con la integración efectiva de buena ingeniería y manejo de dicho producto u objeto.

2.6.4. Curva típica de fallas:

Según Salazar et al. (2005), la curva típica de fallas es una curva que representa los diferentes tipos de falla que un equipo o componente del mismo sufre durante el período de tiempo desde su puesta en operación hasta el fin de su ciclo de vida útil. La figura 2.8 representa los tres componentes que forman la curva típica de flujo de fallas:

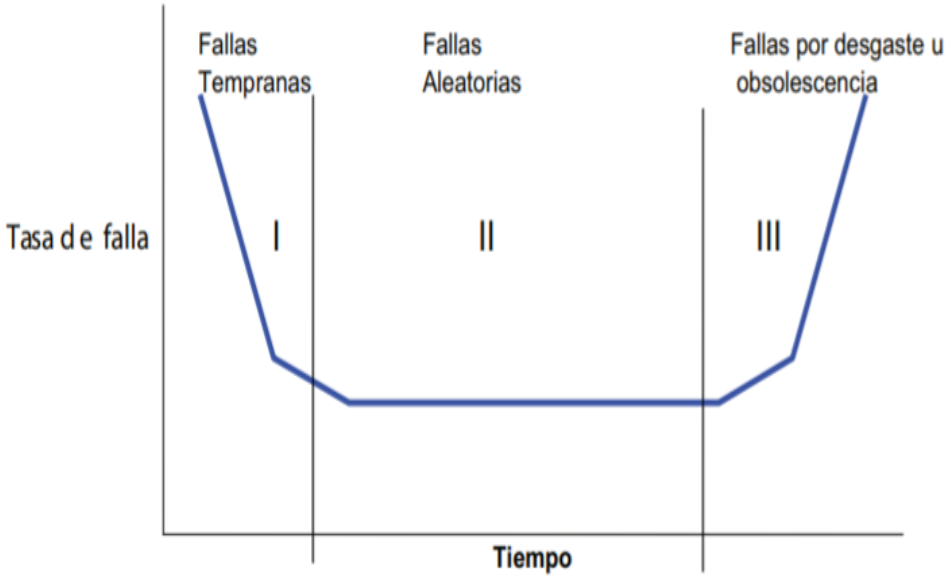


Figura 2.8: Curva típica de flujo de fallas. Fuente: Salazar et al. (2005)

De la curva anterior aparecen tres nuevos conceptos:

- **Fallas tempranas:** Son representadas por la primera parte de la curva (zona izquierda), las tasas de falla se relacionan con un equipo nuevo y pueden ser provocadas por partes faltantes, falta de capacitación de personas que instalan el equipo, daño causado a los aparatos o dispositivos o fallas por defecto de fabricación de las máquinas y por insuficiente asentamiento de las piezas y uniones.
- **Fallas aleatorias:** Este tipo de fallas son inesperadas y suelen aparecer por sobrecargas o averías, causadas por factores externos que provocan las fallas aún en las piezas mejor construidas (región II de la figura 2.8). A este tipo de fallas se les conoce como fallas aleatorias y se representan por una línea horizontal en la curva, lo anterior implica que cada miembro de la población de componentes tiene la misma probabilidad de sufrir una falla.
- **Fallas por desgaste u obsolescencia:** Se representan por la tercera parte de la gráfica de la figura 2.8. Son las fallas debido a obsolescencia, por la edad, fatiga, corrosión, deterioro mecánico, eléctrico, hidráulico o por el bajo nivel de mantenimiento y reparación del equipo. Destaca el hecho de que existe cierta similitud entre la curva típica de flujo de fallas y las tasas de mortalidad y supervivencia humana.

2.6.5. Tiempo medio entre fallas (MTBF):

Según Arróspide (2008), el tiempo medio entre fallas (o MTBF, del inglés: Mean Time Between Failures) indica el intervalo de tiempo más probable entre un arranque y la aparición de una falla; es decir, es el tiempo medio transcurrido hasta la llegada del evento "falla". Mientras mayor sea su valor, mayor es la confiabilidad del componente o equipo. Uno de los parámetros más importantes utilizados en el estudio de la Confiabilidad constituye el MTBF, es por esta razón que debe ser tomado como un indicador más que represente de alguna manera el comportamiento de un equipo específico. Asimismo, para determinar el valor de este indicador se deberá utilizar la data primaria histórica almacenada en los sistemas de información.

De este modo, el MTBF puede determinarse calculando la diferencia entre el tiempo total de trabajo del equipo (que es el número de horas que habría funcionado si no se hubiese averiado) y su tiempo de avería, dividido por el número total de fallas que ha sufrido. Matemáticamente

$$MTBF = \frac{Tiempo\ Total\ de\ Trabajo - Tiempo\ de\ avería}{Número\ de\ fallas} \tag{2.1}$$

2.6.6. Tiempo medio de reparación (MTTR):

Según Arróspide (2008), el tiempo medio de reparación (MTTR, del inglés: Mean Time to Repair) es la medida de la distribución del tiempo de reparación de un equipo o sistema. Este indicador mide la efectividad en restituir el equipo a condiciones óptimas de operación una vez que el equipo se encuentra fuera de servicio por un fallo, dentro de un período de tiempo determinado. El tiempo promedio para reparar es un parámetro de medición

asociado a la mantenibilidad, es decir, a la ejecución del mantenimiento. La mantenibilidad, definida como la probabilidad de devolver el equipo a condiciones operativas en un cierto tiempo utilizando procedimientos prescritos, es una función del diseño del equipo (factores tales como accesibilidad, modularidad, estandarización y facilidades de diagnós tico, facilitan enormemente el mantenimiento). Para un diseño dado, si las reparaciones se realizan con personal calificado y con herramientas, documentación y procedimientos prescritos, el tiempo de reparación depende de la naturaleza del fallo y de las mencionadas características de diseño.

Este indicador se calcula dividiendo el tiempo total de mantenimiento correctivo durante un determinado período de tiempo por el número de intervenciones de mantenimiento realizadas. Matemáticamente:

$$MTTR = \frac{Tiempo\ Total\ de\ Mantenimiento}{N\acute{u}mero\ de\ intervenciones} \tag{2.2}$$

2.7. Distribución Weibull

La información presentada sobre la distribución de Weibull es extraida de Salazar et al. (2005). En 1930, el ingeniero y matemático suizo Waloddi Weibull propuso una función de distribución de 3 parámetros cuyas características para su época eran difíciles de manejar. En un inicio, la expectativa de dicha propuesta, presentaba dificultades que radicaban principalmente en el manejo de las no linealidades de la función, para poder así encontrar los parámetros de ajuste. Sin embargo, con el tiempo el desarrollo tecnológico que ha propiciado la creación de diversos softwares ha permitido hacer un uso intensivo de dicha distribución en especial empezando por el medio industrial; y ahora también ha habido un crecimiento explosivo de aplicaciones en el área de las ciencias agrícolas.

La versatilidad de la distribución de Weibull radica en las diferentes formas que puede tomar dependiendo de los valores que toman sus parámetros. Las implicancias físicas, teóricas, algebraicas, y gráficas son algunos aspectos interesantes que generan y dan lugar a una gran cantidad de trabajos diversos.

Físicamente, los valores extremos de la función Weibull se vinculan a la vida útil de los productos en estudio, y han desarrollado toda una filosofía o iniciativas de perfeccionamiento relacionado con los círculos de calidad, o el concepto de cero-falla entre otros. Teóricamente, se puede observar que los cambios en los parámetros generan una familia de distribución cuyos casos específicos coinciden con otras distribuciones como la exponencial, gaussiana, o chi-cuadrada (χ^2), entre muchas otras.

Algebraicamente, la función de distribución Weibull, así como su distribución acumulada son formas cerradas desde el punto de vista matemático, sin embargo hallar sus parámetros requiere estrategias algebraicas no tan triviales, de índole no lineal que propicia el uso de algoritmos especializados del tipo Newton. Gráficamente, también se han formulado estrategias para encontrar los parámetros de la función de Weibull haciendo uso de escalas logarítmicas,

sin embargo, poco a poco los métodos computacionales han ido ganando más terreno en el ajuste de curvas (Wallace et al. (2000)).

2.7.1. Características generales de la Distribución Weibull

La función Weibull de densidad viene dada por:

$$f(T) = \frac{\beta}{\eta}\left(\frac{T-\gamma}{\eta}\right)^{\beta-1} e^{-\left(\frac{T-\gamma}{\eta}\right)^{\beta}} \tag{2.3}$$

donde:

- β=Parámetro de forma (indicador del mecanismo de falla).
- η=Parámetro de escala (vida característica).
- γ=Parámetro de localización (vida mínima).

La figura 2.9 muestra el comportamiento de la distribución Weibull para diversos valores del parámetro de forma β:

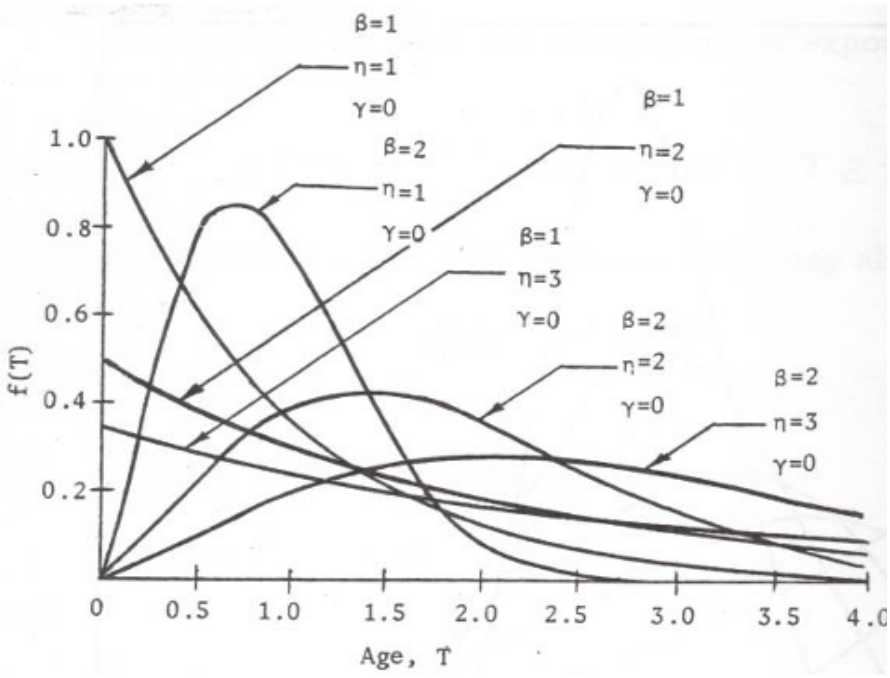


Figura 2.9: Función de densidad de densidad de probabilidad de Weibull para diversos valores de β y η manteniendo γ=0. Fuente: Kececioglu (1991).

A continuación se muestran algunas propiedades de la función Weibull de densidad obtenidas de [6]:

1. Para $0 < \beta < 1$, $f(T)$ decrece monótamente y es convexa.

2. Para $\beta = 1$ se obtiene la función exponencial con dos parámetros.

$$f(T) = \frac{1}{\eta} e^{-\left(\frac{T-\gamma}{\eta}\right)} \qquad donde \qquad \eta = \frac{1}{\lambda} \tag{2.4}$$

$$f(T) = \lambda e^{-\lambda(T-\gamma)} \quad \gamma \geq 0, t \geq \gamma, \eta > 0 \tag{2.5}$$

3. Para $\beta > 1$, $f(T)$ asume formas similares a la distribución normal siempre que $\eta = 1$ y $\gamma = 0$.

4. Un cambio en el parámetro de escala η tiene el mismo efecto en la distribución que un cambio de escala de la abscisa, es decir, para un mismo valor de β y γ en una distribución normal, si η se incrementa, la distribución se contrae y si η disminuye, la distribución se expande.

5. El parámetro de localización γ sirve para ubicar el inicio de la distribución a lo largo del eje x.

6. La media $\bar{T}$ es:

$$\bar{T} = \eta\Gamma\left(\frac{1}{\beta} + 1\right) \tag{2.6}$$

7. La desviación estándar σ corresponde a:

$$\sigma = \eta\left(\Gamma\left(\frac{2}{\beta} + 1\right) - \Gamma\left(\frac{1}{\beta} + 1\right)^2\right)^{\frac{1}{2}} \tag{2.7}$$

8. Respecto a las unidades, el parámetro β es un número adimensional, pero los parámetros η y γ tienen las mismas unidades que T, tales como horas, ciclos, miles, etc. Por otro lado, γ con valor negativo indica fallas antes del inicio de la prueba.

2.7.2. Función de confiabilidad

Según Madrigal (2004), la función de distribución acumulada para una población es llamada distribución de vida y se denota como $F(t)$. La $F(t)$ se interpreta como la proporción de componentes, equipos o sistemas que fallan antes o hasta el tiempo t.

$$F(t) = \int_0^t f(x)dx, \quad t \in [0, \infty] \tag{2.8}$$

donde t es la variable aleatoria que indica el tiempo de fallas.

La función de distribución acumulada se puede interpretar de dos maneras:

1. La probabilidad o seguridad de que una unidad de la población falle antes de t unidades de tiempo.

2. Fracción de la población que falla antes de t unidades de tiempo (incluye el tiempo t).

La función de confiabilidad $R(t)$ se define de la siguiente manera:

$$R(t) = 1 - F(t) \tag{2.9}$$

Esta función se puede interpretar de la siguiente forma:

1. La probabilidad de que una unidad de la población no haya fallado antes del tiempo t.
2. Fracción de la población que sobreviva al tiempo t.

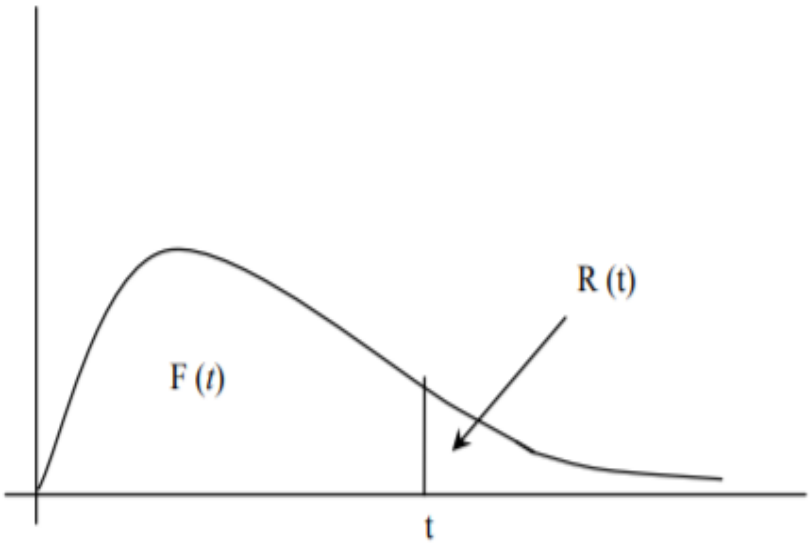


Figura 2.10: Gráfica de la función de confiabilidad. Fuente: Madrigal (2004)

2.7.3. Características de la función de confiabilidad Weibull

La función de confiabilidad Weibull se expresa en la siguiente ecuación:

$$R(t) = e^{-\left(\frac{T-\gamma}{\eta}\right)^{\beta}} \tag{2.10}$$

Según Salazar et al. (2004), la función de confiabilidad Weibull se inicia en 1, dado que se supone que al iniciar operación todos los equipos se encuentran en buenas condiciones y conforme pasa el tiempo la confiabilidad va disminuyendo como se muestra en la figura 2.10, para valores de β menores de 1 la función de confiabilidad disminuye de manera asintótica. Al igual que la función de densidad, para valores de $\beta = 1$ la función de confiabilidad asume la forma exponencial.

2.7.4. Características generales de la función de confiabilidad de Weibull

1. Para $\beta = 1$ la curva decrece monótamente más rápido que para $0 < \beta < 1$.

2. La confiabilidad para una operación de duración $(\gamma + \eta)$ empezando esta a la edad 0 es:

$$R(T) = e^{-\left(\frac{\gamma+\eta-T}{\eta}\right)^{\beta}} = e^{-1} = 0,368 \tag{2.11}$$

Lo anterior significa que para una operación de duración $(\gamma + \eta)$, únicamente el 36,8 % de los equipos sobrevivirán.

2.7.5. Función Tasa de Falla de Weibull

La función de tasa de falla de Weibull se muestra en la siguiente ecuación:

$$\lambda(t) = \frac{\beta}{\eta}\left(\frac{T-\gamma}{\eta}\right)^{\beta-1} = \frac{f(T)}{R(t)} \tag{2.12}$$

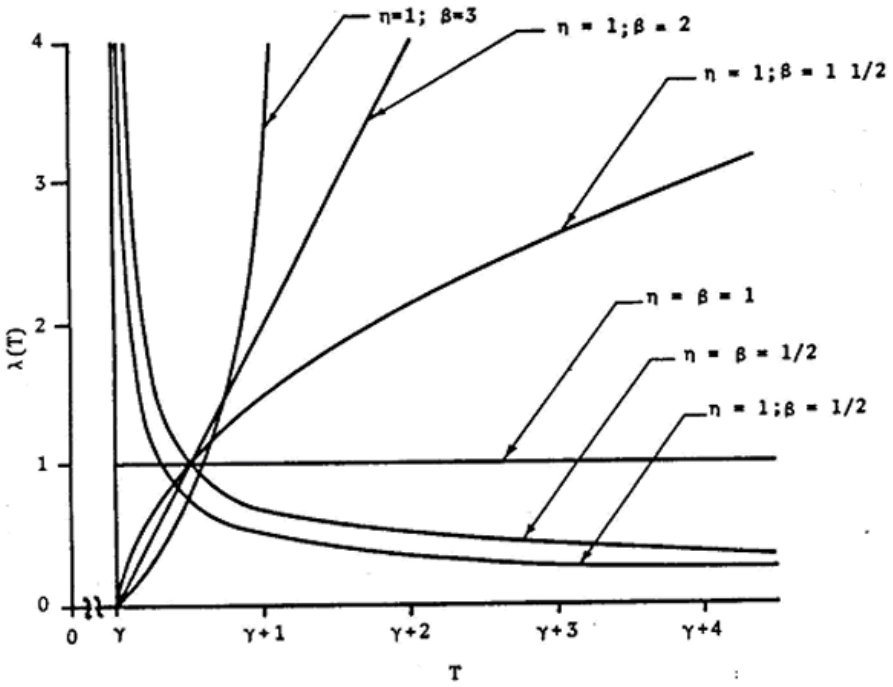


Figura 2.11: Función tasa de falla para varios valores de β. Fuente: Salazar et al. (2004)

2.7.6. Estimación de parámetros de la distribución Weibull:

Según Serrano (2013), Justus et. al (1978), Seguro Lambert (2000) y Christofferson Gillette (1987) existen diferentes métodos para determinar los parámetros de forma y escala de una distribución Weibull. Estos parámetros pueden ser calculados por cualquiera de los siguientes 5 métodos presentados:

1. **Método de máxima probabilidad:** En este método se utilizan iteraciones numéricas para determinar los parámetros de la distribución de Weibull y son determinados

mediante las siguientes ecuaciones:

$$\beta = \left(\frac{\sum_{i=1}^{n} y_i^k ln(y_i)}{\sum_{i=1}^{n} y_i^k} - \frac{\sum_{i=1}^{n} ln(y_i)}{N} \right) \tag{2.13}$$

$$\eta = \left(\frac{1}{N} \sum_{i=1}^{N} y_i^k \right)^{\frac{1}{k}} \tag{2.14}$$

Donde N representa el número de observaciones e y_i la variable dependiente registradas en ese intervalo de tiempo.

2. **Método de máxima probabilidad modificada:** Cuando los datos de la variable independiente se encuentran en formato de distribución de frecuencia, los parámetros de Weibull se pueden determinar mediante las expresiones:

$$\beta = \left(\frac{\sum_{i=1}^{n} y_i^k ln(y_i) P(y_i)}{\sum_{i=1}^{n} y_i^k P(y_i)} - \frac{\sum_{i=1}^{n} ln(y_i) P(y_i)}{P(y \geq 0)} \right) \tag{2.15}$$

$$\eta = \left(\frac{1}{P(y \geq 0)} \sum_{i=1}^{N} y_i^k P(y_i) \right)^{\frac{1}{k}} \tag{2.16}$$

Donde y_i es el valor central del intervalo i de la variable dependiente, $P(y_i)$ es la frecuencia de la variable dependiente que ocurre dentro del intervalo i.

Cook (2001) sugiere el método de orden estadístico para estimar la distribución acumulada; en el cual los N datos son ordenados en forma ascendente y la distribución acumulada para cada rango r es determinado por:

$$P = \frac{r}{N+1} \tag{2.17}$$

3. **Método de Momentos:** El primer momento y el segundo momento corregido de los parámetros β, η de la distribución de Weibull son:

$$E(v) = c\Gamma\left(1 + \frac{1}{k}\right) \tag{2.18}$$

$$\sigma^2 = c^2 \left[\Gamma\left(1 + \frac{2}{k}\right) - \Gamma^2\left(1 + \frac{1}{k}\right) \right] \tag{2.19}$$

Además, conocido el promedio de la variable dependiente, $\bar{y}$, y la desviación estándar de los datos, s; Dorvlo Atsu (2014) y E.K. Akpinar S. Akpinar (2004) indican que el

parámetro de forma, β, se puede determinar mediante iteración de:

$$\frac{s^2}{\bar{y}^2} = \frac{\Gamma\left(1+\frac{2}{k}\right) - \Gamma^2\left(1+\frac{1}{k}\right)}{\Gamma^2\left(1+\frac{1}{k}\right)} \tag{2.20}$$

Y el parámetro de escala, η, puede ser calculado por:

$$c = \frac{\bar{y}}{\Gamma\left(1+\frac{1}{k}\right)} \tag{2.21}$$

4. **Método mínimos cuadrados (método gráfico):** El método gráfico se utiliza mediante la función de distribución acumulada. En este método, los registros de la variable dependiente son interpolados por una línea recta usando el concepto de mínimos cuadrados. La función de distribución acumulada después de algunas operaciones matemáticas se puede escribir como sigue:

$$ln(-ln(1-F(y))) = \beta ln(y) - \beta ln(\eta) \tag{2.22}$$

La ecuación anterior es básicamente la ecuación de una línea recta, $y = ax + b$. Graficando $ln(-ln(1-F(y)))$ versus $ln(y)$ genera una línea recta con pendiente β y con intersección con al eje y, en $kln(\eta)$.

5. **Método de Christofferson y Gillette:** Un procedimiento desarrollado por Christofferson y Gillette en el cual el parámetro de forma (β) es determinado a través de la ecuación:

$$\beta = \frac{\pi}{\sqrt{6}}\left[\frac{N(N-1)}{N\left(\sum_{i=1}^{N} ln^2(y_i)\right) - \left(\sum_{i=1}^{N} ln(y_i)\right)^2}\right]^{\frac{1}{2}} \tag{2.23}$$

Donde N corresponde al número de observaciones y el parámetro de escala puede ser calculado mediante:

$$\eta = \left(\frac{1}{N}\sum_{i=1}^{N} y_i^k\right)^{\frac{1}{k}} \tag{2.24}$$

2.7.7. Tests de bondad de ajuste

Para comprobar que cierta distribución de observaciones sigue determinada distribución de probabilidad se hace uso de los llamados tests de bondad de ajuste. Las medidas de bondad en general resumen la discrepancia entre los valores observados y los valores esperados en el

modelo de estudio. Formalmente, se puede indicar que un test de bondad de ajuste corresponde a una pruebe estadística donde se deconoce alguna propiedad de la forma funcional de la distribución muestreada. Para un intervalo de confianza fijado, la hipótesis nula (premisa que se asume verdadera) será rechazada si existe una diferencia suficiente entre las frecuencias observadas (datos muestreados) y las esperadas (E.K. Akpinar S. Akpinar (2004)). Ejemplos de tests de bondad de ajuste se mencionan a continuación:

- Test de Kolmogorov-Smirnov.
- Test de Anderson-Darling.
- Test χ^2.

2.8. Machine Learning

El aprendizaje automático o aprendizaje de máquinas (*Machine Learning*) es un campo de la informática que surgió a partir de la investigación realizada en inteligencia artificial. La ventaja que presenta el aprendizaje automático sobre otros tipos de análisis radica en la capacidad que tiene para descubrir relaciones entre variables no sencillas de distinguir y predecir con ello resultados de interés. A diferencia de los algoritmos iterativos en los cuales las operaciones se declaran explícitamente, los algoritmos de Machine Learning toman prestados conceptos de la teoría de la probabilidad para seleccionar, evaluar y mejorar los modelos estadísticos ya existentes.

2.8.1. Aprendizaje supervisado

Según Vallalta (2021), los algoritmos de aprendizaje supervisado basan su aprendizaje en un juego de datos de entrenamiento previamente etiquetados. Por etiquetado se entiende que para cada ocurrencia del juego de datos de entrenamiento se conoce el valor de su atributo objetivo o variable dependiente. Esto le permite al algoritmo poder aprender una función capaz de predecir el atributo objetivo para un juego de datos nuevo. Las dos grandes familias de algoritmos supervisados son:

- Los algoritmos de regresión cuando el resultado a predecir es un atributo numérico.
- Los algoritmos de clasificación cuando el resultado a predecir es un atributo categórico.

Ejemplos de este tipo de algoritmos son los modelos de regresión lineal y logística, clasificándose en regresión simple y múltiple, los árboles de decisión, clasificación Naïve-Bayes (NBC), las redes neuronales y K-NN ($k-$ *nearest neighbor*).

Los algoritmos anteriormente citados se resumen a continuación:

- **Árboles de decisión:** Los árboles de decisión son algoritmos estadísticos o técnicas de *Machine Learning* que permiten construir modelos predictivos en análisis de datos para el Big Data basados en su clasificación según determinadas características o

propiedades, o en la regresión mediante la relación entre distintas variables para predecir el valor de otra.

El árbol de decisión es una estructura que está formada por ramas y nodos de distintos tipos:

- Los nodos internos representan cada una de las características o propiedades a considerar para tomar una decisión.
- Las ramas representan la decisión en función de una determinada condición (por ejemplo probabilidad de ocurrencia).
- Los nodos finales representan el resultado de la decisión.

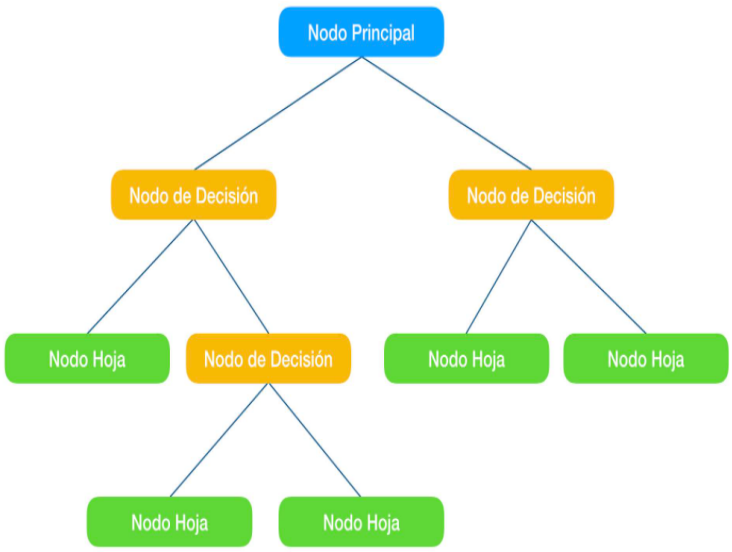


Figura 2.12: Esquema de árbol de decisión con su nomenclatura. Fuente: Rodríguez (2018)

Según Ferrero (2020), la creación de un árbol de decisión en un problema de clasificación se lleva a cabo aplicando el algoritmo de Hunt el cual se basa en la división en sub-conjuntos que buscan una separación óptima. Dado un conjunto de registros de entrenamiento de un nodo, si pertenecen a la misma clase este se considera un nodo terminal, pero si pertenecen a varias clases, se dividen los datos en sub-conjuntos más pequeños en función de una variable y se repite el proceso. Considerando lo anterior se definen métricas para determinar el grado de pureza de un nodo entre las cuales se encuentran:

- Índice Gini: Mide la probabilidad de no sacar dos registros de la misma clase del nodo. A mayor índice de Gini menor pureza, por lo que seleccionaremos la variable con menor Gini ponderado. Suele seleccionar divisiones desbalanceadas, donde normalmente aísla en un nodo una clase mayoritaria y el resto de clases los clasifica en otros nodos. Matemáticamente:

$$GINI = \sum_{i=1}^{n} (P_i)^2 \tag{2.25}$$

– Entropía: La entropía es una medida que se aplica para cuantificar el desorden de un sistema. Si un nodo es puro su entropía es 0 y sólo tiene observaciones de una clase, pero si la entropía es igual a 1, existe la misma frecuencia para cada una de las clases de observaciones. La entropía tiende a crear nodos balanceados en el número de observaciones. Matemáticamente:

$$Entropía = \sum_{i=1}^{n} P_i \times \log_2 P_i \tag{2.26}$$

El módulo *tree* de la librería *sklearn* de *Python* permite graficar árboles de decisión. La figura 2.13 muestra un árbol de decisión generado por este complemento. Dentro de cada nodo (caja) de este árbol se tiene primero el nombre de la variable utilizada, luego el criterio de impureza de los nodos utilizado, posteriormente la cantidad de muestras que llegan a tal nodo y finalmente, dentro del paréntesis cuadrado, la cantidad de instancias de cada clase recordando que las clases toman valores numéricos entre 0 y $n-1$, siendo n el número total de clases del problema, además, estos se ordenan numéricamente dentro del paréntesis.

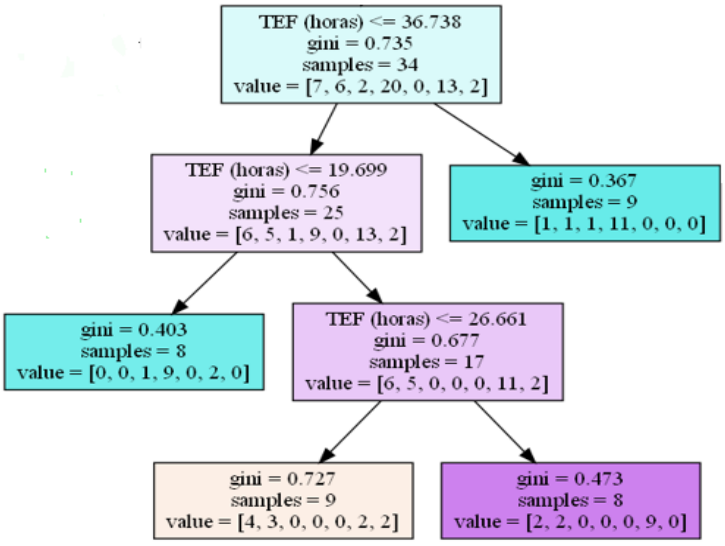


Figura 2.13: Árbol de decisión entregado por Python. Fuente: Elaboración propia.

- **Random Forest:** Según Yiu (2019), un *Random Forest* (Bosque Aleatorio), como su nombre lo indica, consiste en una gran cantidad de árboles de decisión individuales que operan como un conjunto. Cada árbol individual en el bosque aleatorio arroja una predicción de clase y la clase con más votos se convierte en la predicción del modelo. El concepto fundamental detrás del modelo *Random Forest* es simple, pero a la vez poderoso: la sabiduría de las multitudes o también llamada democracia de los árboles

de decisión. En términos de ciencia de datos, la razón por la que el modelo de Bosque Aleatorio funciona tan bien es que una gran cantidad de modelos relativamente no correlacionados (árboles) que funcionan como un conjunto superarán a cualquiera de los modelos constituyentes individuales y constará de una menor varianza a la hora de predecir.

Para asegurar que existan diferencias entre los árboles de decisión que conforman el *Random Forest* se utiliza una técnica llamada *Bagging* o *Bootstrapping* que consiste en que el Bosque Aleatorio permita que cada árbol individual que lo conforma tome muestras aleatorias del conjunto de datos considerando reemplazo de los mismos, esta particularidad permite que se formen árboles de decisión diferentes. Dado que los árboles de decisión son muy sensibles a los datos con los que se entrenan; pequeños cambios en el conjunto de entrenamiento pueden dar como resultado estructuras de árbol significativamente diferentes.

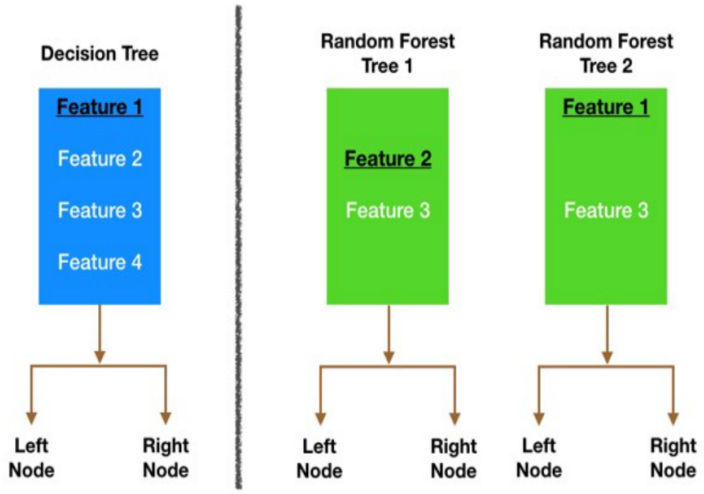


Figura 2.14: Esquema que muestra la construcción de un Random Forest como árboles de clasificación de un conjunto de datos original. Fuente: Yiu (2019)

- **Clasificación Naïve-Bayes:** El algoritmo clasificador Naïve-Bayes (NBC), es un clasificador probabilístico simple el cual incluye una fuerte suposición de independencia de las variables utilizadas. Aunque la suposición de independencia de las variables es generalmente una suposición pobre y a menudo no se respeta para los conjuntos de datos verdaderos. Generalmente proporciona una mejor precisión de clasificación en conjuntos de datos en tiempo real que cualquier otro clasificador. También requiere una pequeña cantidad de datos de entrenamiento. El clasificador Naïve-Bayes aprende de los datos de entrenamiento y luego predice la clase de la instancia de prueba con la mayor probabilidad posterior. Este algoritmo también es útil para datos dimensionales altos dado que la probabilidad de cada atributo se estima independientemente (Chandra et al., 2007). Los algoritmos Naïve-Bayes son conocidos por ser pobres estimadores, por ello, no se deben tomar muy en serio las probabilidades que se obtienen (Lizarazo 2021).

- **K-NN:** La idea de este tipo de algoritmos es simple: el algoritmo clasifica cada dato o

registro nuevo en el grupo que corresponda, según tenga k vecinos más cerca de un grupo o de otro. Es decir, calcula la distancia del elemento nuevo a cada uno de los existentes, y ordena dichas distancias de menor a mayor para ir seleccionando el grupo al cual pertenecer. Este grupo será, por tanto, el de mayor frecuencia con menores distancias. Merkle (2020).

Toda la información presentada sobre regresión lineal y logística es entregada por Amat (2020). Se decide dejarla en una sub-sección diferente debido a que es éste el tipo de algoritmo de Machine Learning que se utilizará para generar un modelo predictivo de fallas para los equipos de la planta de chancado secundario y terciario de DCH.

2.8.2. Regresión Logística Simple

La Regresión Logística Simple, desarrollada por David Cox (1958), es un método de regresión que permite estimar la probabilidad de una variable cualitativa binaria en función de una variable cuantitativa. Una de las principales aplicaciones de la regresión logística es la de clasificación binaria, en el que las observaciones se clasifican en un grupo u otro dependiendo del valor que tome la variable empleada como predictor. Por ejemplo, clasificar a un individuo desconocido como hombre o mujer en función del tamaño de la mandíbula.

Es importante tener en cuenta que, aunque la regresión logística permite clasificar, se trata de un modelo de regresión que modela el logaritmo de la probabilidad de pertenecer a cada grupo. La asignación final se hace en función de las probabilidades predichas.

La existencia de una relación significativa entre una variable cualitativa con dos niveles y una variable continua se puede estudiar mediante otros test estadísticos tales como t-test o ANOVA (un ANOVA de dos grupos es equivalente al t-test). Sin embargo, la regresión logística permite además calcular la probabilidad de que la variable dependiente pertenezca a cada una de las dos categorías en función del valor que adquiera la variable independiente.

2.8.3. Relación entre regresión logística y lineal

Si una variable cualitativa con dos niveles se codifica como 1 y 0, matemáticamente es posible ajustar un modelo de regresión lineal por mínimos cuadrados $\beta_0 + \beta_1 x$. El problema de esta aproximación es que, al tratarse de una recta, para valores extremos del predictor, se obtienen valores de Y menores que 0 o mayores que 1, lo que entra en contradicción con el hecho de que las probabilidades siempre están dentro del rango $[0, 1]$.

Para evitar el problema anterior, la regresión logística transforma el valor devuelto por una regresión lineal ($\beta_0 + \beta_1 x$) empleando una función cuyo resultado está siempre comprendido entre 0 y 1. Existen varias funciones que cumplen esta descripción, una de las más utilizadas es la función logística (también conocida como función sigmoide) la cual se define como sigue:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2.27}$$

Notar que para valores de x muy grandes positivos, el valor de e^x es aproximadamente 0 por lo que el valor de la función sigmoide es 1. Para valores de x muy grandes negativos, el valor e^x tiende a infinito por lo que el valor de la función sigmoide es 0.

Sustituyendo la x de la ecuación anterior por la función lineal: $\beta_0 + \beta_1 x$ (asumiendo que se realiza una regresión lineal previa con la variable independiente y su resultado asociado (variable depenediente), se obtiene que:

$$\begin{aligned} P(Y = k|X = x) &= \frac{1}{1 + e^{-(\beta_0+\beta_1 X)}} \\ &= \frac{1}{\frac{e^{\beta_0+\beta_1 X}}{e^{\beta_0+\beta_1 X}} + \frac{1}{e^{\beta_0+\beta_1 X}}} \\ &= \frac{1}{\frac{1+e^{\beta_0+\beta_1 X}}{e^{\beta_0+\beta_1 X}}} \\ &= \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}} \end{aligned}$$

Donde $P(Y = k|X = x)$ puede interpretarse como: la probabilidad de que la variable cualitativa Y adquiera el valor k (el nivel de referencia, codificado como 1), dado que el predictor X tiene el valor x.

Esta función, puede ajustarse de forma sencilla con métodos de regresión lineal si se emplea su versión logarítmica, obteniendo lo que se conoce como LOG of ODDs:

$$ln\left(\frac{P(Y = k|X = x)}{1 - P(Y = k|X = x)}\right) = \beta_0 + \beta_1 X \tag{2.28}$$

2.8.4. Odds o razón de probabilidad

En la regresión lineal simple, se modela el valor de la variable dependiente Y en función del valor de la variable independiente X. Sin embargo, en la regresión logística, tal como se ha descrito en la sección anterior, se modela la probabilidad de que la variable respuesta Y pertenezca al nivel de referencia 1 en función del valor que adquieran los predictores, mediante el uso de LOG of ODDs.

Los ODDs o razón de probabilidad del evento verdadero se define como el ratio entre la probabilidad de evento verdadero y la probabilidad de evento falso $\frac{p}{q}$.

La trasformación de probabilidades a ODDs es monotónica, si la probabilidad aumenta también lo hacen los ODDs, y viceversa. El rango de valores que pueden tomar los ODDs es de $[0, \infty]$. Dado que el valor de una probabilidad está acotado entre $[0, 1]$ se recurre a una trasformación logit (existen otras) que consiste en el logaritmo natural de los ODDs. Esto permite convertir el rango de probabilidad previamente limitado a $[0, 1]$ a $[\infty, +\infty]$.

2.8.5. Ajuste del modelo

Una vez obtenida la relación lineal entre el logaritmo de los ODDs y la variable predictora X, se tienen que estimar los parámetros β_0 y β_1. La combinación óptima de valores será aquella que tenga la máxima verosimilitud (maximum likelihood, en inglés), es decir el valor de los parámetros β_0 y β_1 con los que se maximiza la probabilidad de obtener los datos observados.

El método de máxima verosimilitud está ampliamente extendido en la estadística aunque su implementación no siempre es trivial.

Otra forma para ajustar un modelo de regresión logística es empleando descenso de gradiente. Si bien este no es el método de optimización más adecuado para resolver la regresión logística, está muy extendido en el ámbito del Machine Learning para ajustar otros modelos.

2.8.6. Evaluación del modelo

Existen diferentes técnicas estadísticas para calcular la significancia de un modelo logístico en su conjunto (p-value del modelo). Todos ellos consideran que el modelo es útil si es capaz de mostrar una mejora respecto a lo que se conoce como modelo nulo, el modelo sin predictores, sólo con β_0. Dos de los más empleados son:

- Wald chi-square: Está muy expandido, pero pierde precisión con tamaños muestrales pequeños.
- Likelihood ratio: Usa la diferencia entre la probabilidad de obtener los valores observados con el modelo logístico creado y las probabilidades de hacerlo con un modelo sin relación entre las variables. Para ello, calcula la significancia de la diferencia de residuos entre el modelo con predictores y el modelo nulo (modelo sin predictores). El estadístico tiene una distribución chi-cuadrado con grados de libertad equivalentes a la diferencia de grados de libertad de los dos modelos comparados. Si se compara respecto al modelo nulo, los grados de libertad equivalen al número de predictores del modelo generado.

Para determinar la significancia individual de cada uno de los predictores introducidos en un modelo de regresión logística se emplea el estadístico Z y el test Wald chi-test. El estadístico Z se define como:

$$Z = \frac{b}{SE_b} \tag{2.29}$$

donde b corresponde al valor del coeficiente en la regresión lineal para determinada variable y SE_b su correspondiente error estándar.

Para evaluar la calidad del modelo de predicción utilizado se suele emplear uno o más de los 5 conceptos que a continuación se presentan:

1. **Probabilidad obtenida a clasificación:** Una de las principales aplicaciones de un modelo de regresión logística es clasificar la variable cualitativa en función de valor que tome el predictor. Para conseguir esta clasificación, es necesario establecer un umbral de probabilidad a partir de la cual se considera que la variable pertenece a uno de los niveles. A modo de ejemplo, se puede asignar una observación al grupo 1 si $P(Y = 1|X) > 0,5$ y al grupo 0 si de lo contrario.

2. **Matriz de confusión:** Según lo expuesto por López (2021), en el campo de la inteligencia artificial y en especial en el problema de la clasificación estadística (regresión logística, Naïve-Bayes y K-NN), una matriz de confusión es una herramienta que permite visualizar de modo sencillo el desempeño de un algoritmo que se utiliza con aprendizaje supervisado. Cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa a las instancias en la clase real. Uno de los beneficios de las matrices de confusión es que facilitan ver si el sistema está confundiendo dos clases.

   Se definen:

   - **Verdaderos positivos (VP):** Número de casos que la prueba declara positivos y que son verdaderamente positivos.
   - **Falsos positivos (FP):** Número de casos que la prueba declara positivos y que en realidad son negativos.
   - **Verdaderos negativo (VN):** Número de casos que la prueba declara negativos y que son realmente negativos.
   - **Falsos negativos (FN):** Número de casos que la prueba declara negativos y que en realidad son positivos.

   La figura 2.15 muestra una representación gráfica de una matriz de confusión.

Figura 2.15: Matriz de confusión. Fuente: Barrios (2019)

3. **Curva ROC:** De lo presentado por Swets (1995), en la teoría de detección de señales, una curva ROC (acrónimo de Receiver Operating Characteristic, o Característica Operativa del Receptor) corresponde a una representación gráfica de la sensibilidad frente a la especificidad para un sistema clasificador binario según se varía el umbral de discriminación. Otra interpretación de este gráfico es la representación de la razón o proporción de verdaderos positivos (VPR = Razón de Verdaderos Positivos) frente a la razón o proporción de falsos positivos (FPR = Razón de Falsos Positivos) también según se varía el umbral de discriminación (valor a partir del cual se decide que un caso es un positivo). ROC también puede significar Relative Operating Characteristic (Característica Operativa Relativa) porque es una comparación de dos características operativas (VPR y FPR) según se cambia el umbral para la decisión.

   Formalmente, se tiene que:

   - Razón de falsos positivos: Se calcula como el número de positivos verdaderos divididos entre el número de positivos verdaderos y de falsos negativos. Describe qué tan bueno es un modelo prediciendo las clases positivas cuando la salida real es positiva. También se conoce esta tasa como sensibilidad.
   - Razón de Falsos Positivos: Se calcula como el número de falsos positivos dividido entre la suma de falsos positivos con los verdaderos negativos. Se considera como la tasa de "falsa alarma" y resume qué tan común es que una clase negativa sea determinada por el modelo como positiva.
   - La especificidad es la inversa de la tasa de falsos positivos. Se obtiene dividiendo el número total de verdaderos negativos entre la suma de los verdaderos negativos y

los falsos positivos.

La matriz de confusión puede proporcionar varias medidas de evaluación (ver caja de terminología). Para dibujar una curva ROC sólo son necesarias las razones de Verdaderos Positivos (VPR) y de Falsos Positivos (FPR). La VPR mide hasta qué punto un clasificador o prueba diagnóstica es capaz de detectar o clasificar los casos positivos correctamente, de entre todos los casos positivos disponibles durante la prueba. La FPR define cuántos resultados positivos son incorrectos de entre todos los casos negativos disponibles durante la prueba.

Un espacio ROC se define por FPR y VPR como ejes x e y respectivamente, y representa los intercambios entre verdaderos positivos (en principio, beneficios) y falsos positivos (en principio, costes). Dado que VPR es equivalente a sensibilidad y FPR es igual a $1 - especificidad$, el gráfico ROC también es conocido como la representación de sensibilidad frente a $(1 - especificidad)$. Cada resultado de predicción o instancia de la matriz de confusión representa un punto en el espacio ROC.

La figura 2.16 muestra un espacio ROC.

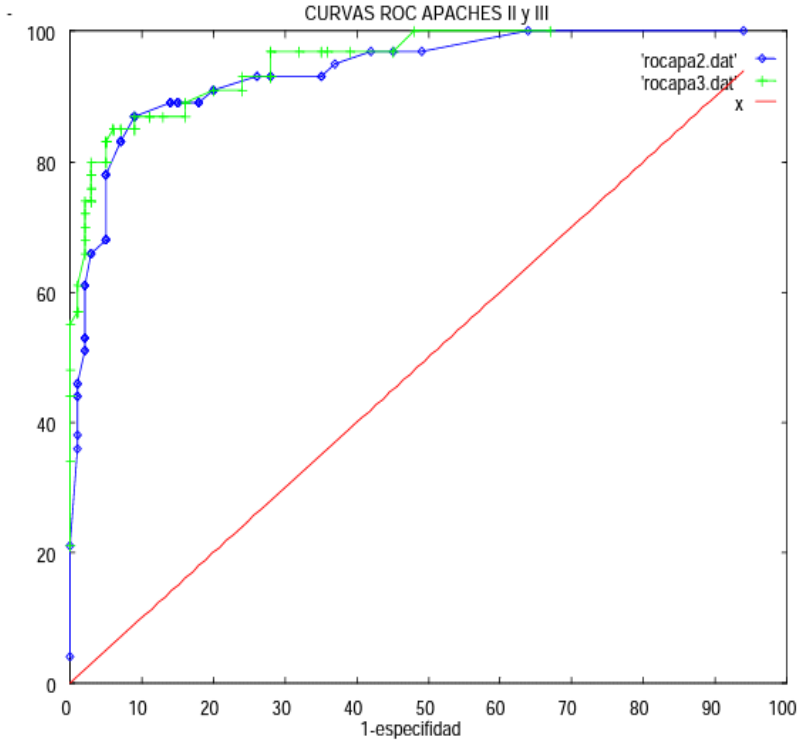


Figura 2.16: Ejemplo de curvas ROC en una prueba diagnóstica de detección rápida en la UCI. Fuente: Concejero (2004)

El mejor método posible de predicción se situaría en un punto en la esquina superior izquierda, o coordenada $(0, 1)$ del espacio ROC, representando un 100 % de sensibilidad (ningún falso negativo) y un 100 % también de especificidad (ningún falso positivo). A este punto $(0, 1)$ también se le llama una clasificación perfecta. Por el contrario, una

clasificación totalmente aleatoria (o adivinación aleatoria) daría un punto a lo largo de la línea diagonal, que se llama también línea de no-discriminación, desde el extremo inferior izquierdo hasta la esquina superior derecha (independientemente de los tipos de base positivas y negativas). Un ejemplo típico de adivinación aleatoria sería decidir a partir de los resultados de lanzar una moneda al aire, a medida que el tamaño de la muestra aumenta, el punto de un clasificador aleatorio de ROC se desplazará hacia la posición $(0,5, 0,5)$.

La diagonal divide el espacio ROC. Los puntos por encima de la diagonal representan los buenos resultados de clasificación (mejor que el azar), puntos por debajo de la línea de los resultados pobres (peor que al azar). Notar que la salida de un predictor consistentemente pobre simplemente podría ser invertida para obtener un buen predictor.

4. **Curva precisión-sensibilidad:** Según lo expuesto en The Machine Learners (2022) la precisión se calcula como el número de verdaderos positivos entre la suma de verdaderos positivos y de falsos positivos. Describe cuán bueno es el modelo para predecir las salidas de la clase positiva (habitualmente designada con un 1). La precisión también es llamada poder predictivo positivo.

   Por otro lado, la sensibilidad (recall), se define como Verdaderos positivos divididos entre la suma de verdaderos positivos y de falsos positivos.

   Por tanto, la curva de precisión-sensibilidad enfrenta la precisión (eje y) con la sensibilidad (eje x) para diferentes umbrales.

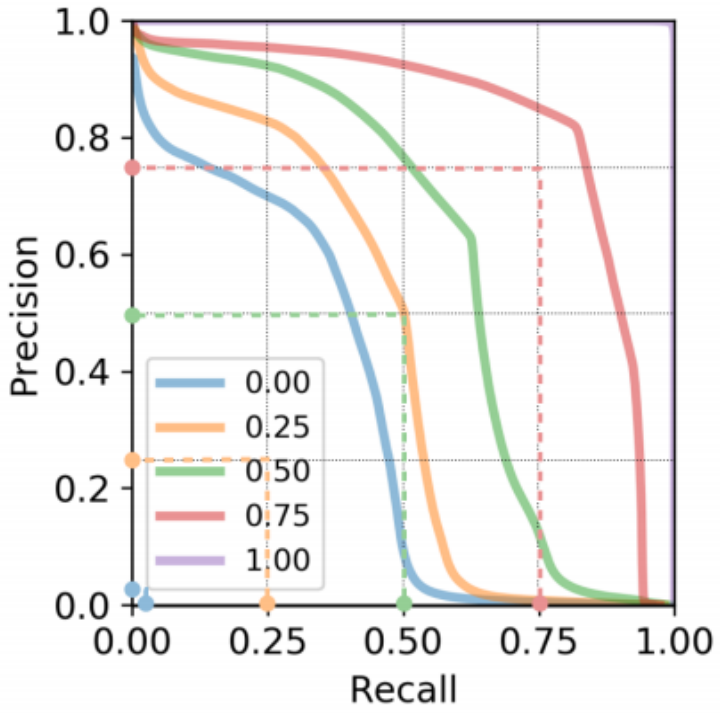


Figura 2.17: Ejemplo de curva de precisión-sensibilidad. Fuente: The Machine Learners (2022)

5. **Área bajo la curva (AUC) de curva ROC:** El valor AUC se utiliza como resumen del rendimiento del modelo. Cuanto más esté hacia la izquierda la curva ROC, más área habrá contenida bajo ella y por ende, mejor será el clasificador. A modo de ejemplo, un clasificador aleatorio cuenta con una AUC de 0,5 (línea punteado de color rojo en 2.18). A modo de ejemplo, la curva donde se ubica el punto A cuenta con una AUC superior a 0,5.

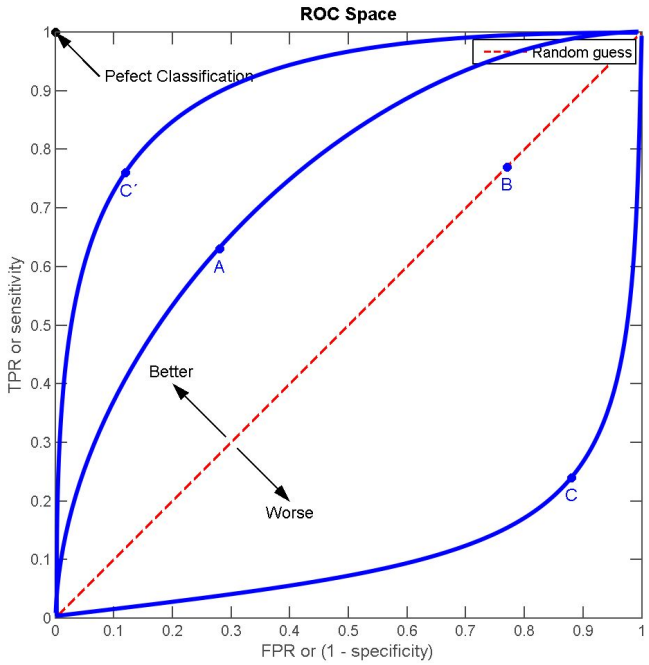


Figura 2.18: Curvas ROC y AUC de clasificador aleatorio. Fuente: Melillanca (2018)

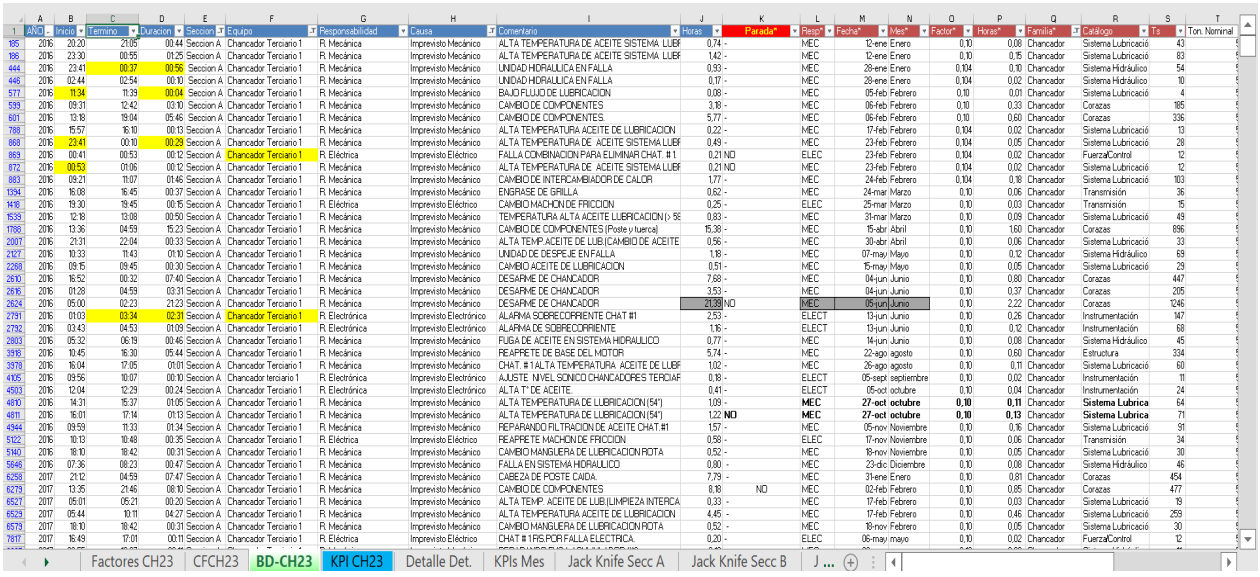
Capítulo 3

Metodología

3.1. Datos a utilizar

Se cuenta con una base de datos que incluye registros de falla e inconvenientes en los equipos de la planta de chancado secundario y terciario de la división Chuquicamata de Codelco (DCH). Esta base de datos tiene información recopilada desde enero de 2016 a septiembre de 2021. La figura 3.1 muestra la base de datos a utilizar con todos los campos que contiene. Para poder predecir fallas a partir del uso de algoritmos de *Machine Learning* es necesario alimentar este programa con los datos adecuados que se extraen a partir de cálculos hechos con la información que provee esta base.

Debido al nivel de confianza que se tiene con los registros tomados año a año, se decide considerar sólo las fallas desde el 01 enero de 2019 al 30 septiembre de 2021.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	AÑO	Inicio	Termino	Duracion	Seccion	Equipo	Responsabilidad	Causa	Comentario	Horas	Parada*	Resp*	Fecha*	Mes*	Factor*	Horas*	Familia*	Catálogo	Ts	Ton. Nominal
185	2016	20:20	21:05	00:44	Seccion A	Chancador Terciario 1	R. Mecánica	Imprevisto Mecánico	ALTA TEMPERATURA DE ACEITE SISTEMA LUBR	0,74	-	MEC	12-ene	Enero	0,10	0,08	Chancador	Sistema Lubricació	43	
186	2016	23:30	00:55	01:25	Seccion A	Chancador Terciario 1	R. Mecánica	Imprevisto Mecánico	ALTA TEMPERATURA DE ACEITE SISTEMA LUBR	1,42	-	MEC	12-ene	Enero	0,10	0,15	Chancador	Sistema Lubricació	83	
444	2016	23:41	00:37	00:56	Seccion A	Chancador Terciario 1	R. Mecánica	Imprevisto Mecánico	UNIDAD HIDRAULICA EN FALLA	0,93	-	MEC	28-ene	Enero	0,104	0,10	Chancador	Sistema Hidráulico	54	
446	2016	02:44	02:54	00:10	Seccion A	Chancador Terciario 1	R. Mecánica	Imprevisto Mecánico	UNIDAD HIDRAULICA EN FALLA	0,17	-	MEC	28-ene	Enero	0,104	0,02	Chancador	Sistema Hidráulico	10	
577	2016	11:34	11:39	00:04	Seccion A	Chancador Terciario 1	R. Mecánica	Imprevisto Mecánico	BAJO FLUJO DE LUBRICACION	0,08	-	MEC	05-feb	Febrero	0,10	0,01	Chancador	Sistema Lubricació	4	
599	2016	09:31	12:42	03:10	Seccion A	Chancador Terciario 1	R. Mecánica	Imprevisto Mecánico	CAMBIO DE COMPONENTES	3,18	-	MEC	06-feb	Febrero	0,10	0,32	Chancador	Corazas	185	
601	2016	13:18	19:04	05:46	Seccion A	Chancador Terciario 1	R. Mecánica	Imprevisto Mecánico	CAMBIO DE COMPONENTES.	5,77	-	MEC	06-feb	Febrero	0,10	0,60	Chancador	Corazas	336	
788	2016	15:57	16:10	00:13	Seccion A	Chancador Terciario 1	R. Mecánica	Imprevisto Mecánico	ALTA TEMPERATURA ACEITE DE LUBRICACION	0,22	-	MEC	17-feb	Febrero	0,104	0,02	Chancador	Sistema Lubricació	13	
868	2016	23:41	00:10	00:29	Seccion A	Chancador Terciario 1	R. Mecánica	Imprevisto Mecánico	ALTA TEMPERATURA DE ACEITE SISTEMA LUBR	0,49	-	MEC	23-feb	Febrero	0,104	0,05	Chancador	Sistema Lubricació	28	
869	2016	00:41	00:53	00:12	Seccion A	Chancador Terciario 1	R. Eléctrica	Imprevisto Eléctrico	FALLA COMBINACION PARA ELIMINAR CHAT. # 1	0,21	NO	ELEC	23-feb	Febrero	0,104	0,02	Chancador	Fuerza/Control	12	
872	2016	00:53	01:06	00:12	Seccion A	Chancador Terciario 1	R. Mecánica	Imprevisto Mecánico	ALTA TEMPERATURA DE ACEITE SISTEMA LUBR	0,21	NO	MEC	23-feb	Febrero	0,104	0,02	Chancador	Sistema Lubricació	12	
883	2016	09:21	11:07	01:46	Seccion A	Chancador Terciario 1	R. Mecánica	Imprevisto Mecánico	CAMBIO DE INTERCAMBIADOR DE CALOR	1,77	-	MEC	24-feb	Febrero	0,104	0,18	Chancador	Sistema Lubricació	103	
1394	2016	16:08	16:45	00:37	Seccion A	Chancador Terciario 1	R. Mecánica	Imprevisto Mecánico	ENGRASE DE GRILLA	0,62	-	MEC	24-mar	Marzo	0,10	0,06	Chancador	Transmisión	36	
1416	2016	19:30	19:45	00:15	Seccion A	Chancador Terciario 1	R. Eléctrica	Imprevisto Eléctrico	CAMBIO MACHON DE FRICCION	0,25	-	ELEC	25-mar	Marzo	0,10	0,03	Chancador	Transmisión	15	
1539	2016	12:18	13:08	00:50	Seccion A	Chancador Terciario 1	R. Mecánica	Imprevisto Mecánico	TEMPERATURA ALTA ACEITE LUBRICACION (> 55	0,83	-	MEC	31-mar	Marzo	0,10	0,09	Chancador	Sistema Lubricació	49	
1788	2016	13:36	04:59	15:23	Seccion A	Chancador Terciario 1	R. Mecánica	Imprevisto Mecánico	CAMBIO DE COMPONENTES (Poste y tuerca)	15,38	-	MEC	15-abr	Abril	0,10	1,60	Chancador	Corazas	896	
2007	2016	21:31	22:04	00:33	Seccion A	Chancador Terciario 1	R. Mecánica	Imprevisto Mecánico	ALTA TEMP. ACEITE DE LUB.) CAMBIO DE ACEITE	0,56	-	MEC	30-abr	Abril	0,10	0,06	Chancador	Sistema Lubricació	33	
2127	2016	10:33	11:43	01:10	Seccion A	Chancador Terciario 1	R. Mecánica	Imprevisto Mecánico	UNIDAD DE DESPEJE EN FALLA	1,18	-	MEC	07-may	Mayo	0,10	0,12	Chancador	Sistema Hidráulico	69	
2288	2016	09:15	09:45	00:30	Seccion A	Chancador Terciario 1	R. Mecánica	Imprevisto Mecánico	CAMBIO ACEITE DE LUBRICACION	0,51	-	MEC	15-may	Mayo	0,10	0,05	Chancador	Sistema Lubricació	29	
2610	2016	16:52	00:32	07:40	Seccion A	Chancador Terciario 1	R. Mecánica	Imprevisto Mecánico	DESARME DE CHANCADOR	7,68	-	MEC	04-jun	Junio	0,10	0,80	Chancador	Corazas	447	
2616	2016	01:28	04:59	03:31	Seccion A	Chancador Terciario 1	R. Mecánica	Imprevisto Mecánico	DESARME DE CHANCADOR	3,53	-	MEC	04-jun	Junio	0,10	0,37	Chancador	Corazas	205	
2624	2016	05:00	02:23	21:23	Seccion A	Chancador Terciario 1	R. Mecánica	Imprevisto Mecánico	DESARME DE CHANCADOR	21,38	NO	MEC	05-jun	Junio	0,10	2,22	Chancador	Corazas	1246	
2791	2016	01:03	03:34	02:31	Seccion A	Chancador Terciario 1	R. Electrónica	Imprevisto Electrónico	ALARMA SOBRECORRIENTE CHAT #1	2,53	-	ELECT	13-jun	Junio	0,10	0,26	Chancador	Instrumentación	147	
2792	2016	03:43	04:53	01:09	Seccion A	Chancador Terciario 1	R. Electrónica	Imprevisto Electrónico	ALARMA DE SOBRECORRIENTE	1,16	-	ELECT	13-jun	Junio	0,10	0,12	Chancador	Instrumentación	68	
2803	2016	05:32	06:19	00:46	Seccion A	Chancador Terciario 1	R. Mecánica	Imprevisto Mecánico	FUGA DE ACEITE EN SISTEMA HIDRAULICO	0,77	-	MEC	14-jun	Junio	0,10	0,08	Chancador	Sistema Hidráulico	45	
3918	2016	10:45	16:30	05:44	Seccion A	Chancador Terciario 1	R. Mecánica	Imprevisto Mecánico	REAPRETE DE BASE DEL MOTOR	5,74	-	MEC	22-ago	agosto	0,10	0,60	Chancador	Estructura	334	
3978	2016	16:04	17:05	01:01	Seccion A	Chancador Terciario 1	R. Mecánica	Imprevisto Mecánico	CHAT. # 1 ALTA TEMPERATURA ACEITE DE LUBR	1,02	-	MEC	26-ago	agosto	0,10	0,11	Chancador	Sistema Lubricació	60	
4105	2016	09:56	10:07	00:10	Seccion A	Chancador terciario 1	R. Electrónica	Imprevisto Electrónico	AJUSTE NIVEL SONICO CHANCADORES TERCIAR	0,18	-	ELECT	05-sept	septiembre	0,10	0,02	Chancador	Instrumentación	11	
4503	2016	12:04	12:29	00:24	Seccion A	Chancador Terciario 1	R. Electrónica	Imprevisto Electrónico	ALTA T° DE ACEITE.	0,41	-	ELECT	05-oct	octubre	0,10	0,04	Chancador	Instrumentación	24	
4810	2016	14:31	15:37	01:05	Seccion A	Chancador Terciario 1	R. Mecánica	Imprevisto Mecánico	ALTA TEMPERATURA DE LUBRICACION (54°)	1,09	-	**MEC**	**27-oct**	**octubre**	**0,10**	**0,11**	Chancador	**Sistema Lubrica**	64	
4811	2016	16:01	17:14	01:13	Seccion A	Chancador Terciario 1	R. Mecánica	Imprevisto Mecánico	ALTA TEMPERATURA DE LUBRICACION (54°)	1,22	**NO**	**MEC**	**27-oct**	**octubre**	**0,10**	**0,13**	Chancador	**Sistema Lubrica**	71	
4944	2016	09:59	11:33	01:34	Seccion A	Chancador Terciario 1	R. Mecánica	Imprevisto Mecánico	REPARANDO FILTRACION DE ACEITE CHAT.#1	1,57	-	MEC	05-nov	Noviembre	0,10	0,16	Chancador	Sistema Lubricació	91	
5122	2016	10:13	10:48	00:35	Seccion A	Chancador Terciario 1	R. Eléctrica	Imprevisto Eléctrico	REAPRETE MACHON DE FRICCION	0,58	-	ELEC	17-nov	Noviembre	0,10	0,06	Chancador	Transmisión	34	
5140	2016	18:10	18:42	00:31	Seccion A	Chancador Terciario 1	R. Mecánica	Imprevisto Mecánico	CAMBIO MANGUERA DE LUBRICACION ROTA	0,52	-	MEC	18-nov	Noviembre	0,10	0,05	Chancador	Sistema Lubricació	30	
5646	2016	07:36	08:23	00:47	Seccion A	Chancador Terciario 1	R. Mecánica	Imprevisto Mecánico	FALLA EN SISTEMA HIDRAULICO	0,80	-	MEC	23-dic	Diciembre	0,10	0,08	Chancador	Sistema Hidráulico	46	
6258	2017	21:12	04:59	07:47	Seccion A	Chancador Terciario 1	R. Mecánica	Imprevisto Mecánico	CABEZA DE POSTE CAIDA.	7,79	-	MEC	31-ene	Enero	0,10	0,81	Chancador	Corazas	454	
6273	2017	13:35	21:46	08:10	Seccion A	Chancador Terciario 1	R. Mecánica	Imprevisto Mecánico	CAMBIO DE COMPONENTES	8,18	NO	MEC	02-feb	Febrero	0,10	0,85	Chancador	Corazas	477	
6527	2017	05:01	05:21	00:20	Seccion A	Chancador Terciario 1	R. Mecánica	Imprevisto Mecánico	ALTA TEMP. ACEITE DE LUB.) LIMPIEZA INTERCA	0,33	-	MEC	17-feb	Febrero	0,10	0,03	Chancador	Sistema Lubricació	19	
6529	2017	05:44	10:11	04:27	Seccion A	Chancador Terciario 1	R. Mecánica	Imprevisto Mecánico	ALTA TEMPERATURA ACEITE DE LUBRICACION	4,45	-	MEC	17-feb	Febrero	0,10	0,46	Chancador	Sistema Lubricació	259	
6573	2017	18:10	18:42	00:31	Seccion A	Chancador Terciario 1	R. Mecánica	Imprevisto Mecánico	CAMBIO MANGUERA DE LUBRICACION ROTA	0,52	-	MEC	19-nov	Febrero	0,10	0,05	Chancador	Sistema Lubricació	30	
7817	2017	16:49	17:01	00:11	Seccion A	Chancador Terciario 1	R. Eléctrica	Imprevisto Eléctrico	CHAT # 1 RS POR FALLA ELECTRICA.	0,20	-	ELEC	06-may	mayo	0,10	0,02	Chancador	Fuerza/Control	12	

Factores CH23 | CFCH23 | BD-CH23 | KPI CH23 | Detalle Det. | KPIs Mes | Jack Knife Secc A | Jack Knife Secc B | J ...

Figura 3.1: Base de datos a utilizar.

Como menciona Salazar et al. (2005) existe un indicador de suma relevancia en cuanto a

Tabla 3.1: Causas de imprevistos en Base de datos proporcionada.

Causa de imprevistos
Aseo
Atollo
Cambio de circuito
Elemento extraño
Imprevisto Eléctrico
Imprevisto Electrónico
Imprevisto Mecánico
Inchancable
Inspecciones
Mantención Mecánica
Protección Equipos
Responsabilidad Eléctrica
Responsabilidad Mecánica
Responsabilidad Operacional
Reducciones
Silos llenos

mantenimiento de equipos, este indicador es el llamado Tiempo medio entre fallas (MTBF) que se calcula sumando todos los tiempos entre falla del equipo y dividiéndolos por la cantitad total de fallas. El algoritmo de *Machine Learning* se debe nutrir con los tiempos entre fallas del equipo considerado y luego considerar horas en las cuales el equipo no va a fallar.

3.1.1. Filtros

1. **Causa de Fallo**

   La base de datos es en primera instancia filtrada por la *Causa* del imprevisto. Las "causas" en cada registro de la base de datos se muestran en la Tabla 3.1. De las causas que se muestran en la Tabla 3.1 sólo los "imprevistos mecánicos", "imprevistos eléctricos" e "imprevistos electrónicos" pueden realmente ser considerados como fallas de un determinado equipo.

2. **Equipo**

   El segundo filtro corresponde al de seleccionar el equipo de la planta de chancado secundario y terciario a analizar, es por este motivo que se filtra por cada uno de los chancadores de la planta (5 chancadores secundarios y 10 chancadores terciarios) y se almacena toda la información concerniente a cada chancador en una nueva hoja de cálculo.

   Es importante destacar que la Base de Datos no se encuentra completamente limpia y en muchas oportunidades es necesario juntar registros de falla porque combinadas re-

presentan sólo una falla, en tal caso, se requiere sumar los tiempos de detención de cada registro que aparece en la columna llamada "Duración" de la Base de datos. Una vez se realiza lo anterior se procede a eliminar "outliers" relacionados a prolongados lapsos de tiempo donde no se efectuaron registros de estos imprevistos, en particular, se observa que no hubo registros de falla entre octubre y diciembre de 2020 y 2021. Con la Base de datos ya depurada se procede a calcular los tiempo entre fallas (TEF) que posteriormente se registran en un archivo que alimenta los algoritmos de *Machine Learning* supervisado.

Los tiempos entre fallas se ordenan de menor a mayor para identificar posibles quiebres en estos tiempos y utilizar ese tiempo como ciclo de evaluación del equipo. Una vez determinado el ciclo de evaluación se rellena el archivo con múltiplos de la duración del ciclo de evaluación y se etiquetan/categorizan con un 0 mostrando que corresponden a registros donde no ocurren fallas a esas horas. Lógicamente los TEF se etiquetan con un 1 y se obtiene un archivo ".csv" de clasificación binaria.(0 ó 1). Con los TEF y los tiempos equiespaciados donde se supone que el equipo no falla se obtiene un conjunto de datos que puede ser utilizado en los 3 algoritmos de *Machine Learning*, es decir, en la regresión logística, Naïve-Bayes y K-NN. Por cada equipo se prueban diversos conjunto de datos los cuales se obtienen modificando el ciclo de evaluación del chancador lo que implica que los 0 del archivo se distribuyen equiespaciadamente en el tiempo de distintas maneras, el objetivo de lo anterior consiste en encontrar el algoritmo de mejor desempeño en términos de exactitud y precisión.

3.2. Validación de datos

Como menciona Salazar et al. (2005), los tiempos entre fallas siguen una distribución Weibull. Debido a que la base de datos suministrada es manipulada con el objetivo de obtener estos tiempos resulta necesario comprobar que los TEF calculados siguen este tipo de distribución y si sus parámetros asociados guardan lógica con las características propias del equipo en la planta.

Para verificar lo anterior se calculan los parámetros de la distribución con el uso de una regresión lineal y se obtiene la curva de confiabilidad del equipo. Por otro lado, se utiliza el complemento *EasyFit* para comprobar que esta distribución cumplen con diversos tests de bondad de ajuste, en particular, el test de Kolmogorov-Smirnov (KS), Anderson-Darling (AD) y χ^2.

3.3. Predicción de momento de falla

Se decide utilizar Regresión Logística, Naïve-Bayes (Gaussiano) y K-NN ($k-nearest\ neighbor$) con el fin de compararlos y determinar cuál algoritmo consta con mayor exactitud y precisión al momento de predecir fallas. Haciendo uso del lenguaje *Python* se programan estos algoritmos. Se hace hincapié en la precisión que tiene cada uno en acertar la ocurrencia de fallas considerando que en mantenimiento de equipos es de mayor gravedad tener un falso negativo que un falso positivo debido a que al obedecer el resultado de la predicción, un falso negativo impide tomar decisiones adecuadas en la planificación del

mantenimiento.

3.3.1. Regresión Logística

El código en *Python* a utilizar se encuentra en la sección de Anexos A.

Se dividen los datos con los que se alimenta el algoritmo en dos grupos. El primero corresponde al conjunto de entrenamiento (80 % de los datos) y el segundo al conjunto de testeo (20 % de los datos).

Se extrae la matriz de confusión correspondiente a las instancias predichas por el algoritmo.

Además, la regresión logística cuenta con un híper-parámetro denominado "umbral" que permite graficar la llamada curva ROC y calcular el área bajo la curva (AUC). La métrica anterior permite decidir qué tan buen predictor es la regresión logística al momento de clasificar fallas como tal. Haciendo uso de *Python* se grafica esta curva. Se considera un umbral de 0,5 para registrar los resultados de la regresión logística en términos de exactitud y precisión.

Dado que los resultados de la regresión logística dependen de cómo se dividen los datos en los conjuntos de entrenamiento y testeo se realizan 30 repeticiones con los datos de cada chancador con el objetivo de obtener estadísticas representativas al considerar la media.

Los resultados de cada ejecución del algoritmo se guardan en un archivo de *Excel* que representa el entregable del modelo.

3.3.2. Naïve-Bayes

El código en *Python* a utilizar se encuentra en la sección de Anexos B.

Se extrae la matriz de confusión correspondiente a las instancias predichas por el algoritmo.

Los resultados de cada ejecución del algoritmo se guardan en un archivo de *Excel* que representa el entregable del modelo.

3.3.3. K-NN

El código en *Python* a utilizar se encuentra en la sección de Anexos C.

Se realizan múltiples ensayos para determinar la cantidad de vecinos adecuada para predecir fallas. Una vez determinado este híper-parámetro, se extrae la matriz de confusión correspondiente a las instancias predichas por el algoritmo.

Los resultados de cada ejecución del algoritmo se guardan en un archivo de *Excel* que representa el entregable del modelo.

3.4. Selección de modelo y equipo

A partir de los resultados entregados por cada algoritmo en términos de exactitud y precisión se decide escoger el modelo con mejor desempeño considerando como métrica de evaluación la propia matriz de confusión y la información que se extrae de ella.

Para la elección de equipo resulta importante los resultados entregados por las técnicas de *Machine Learning* como también la producción que entrega cada equipo en la planta e indicadores de confiabilidad como es el tiempo medio entre fallas (MTBF).

Una vez seleccionado el equipo crítico se planea predecir el cómo va a fallar el equipo. Para lo anterior se recurre a la construcción de un "Random Forest " que permite predecir el modo de falla del chancador. El código en *Python* a utilizar se encuentra en la sección de Anexos D.

3.5. Random Forest

De la base de datos citada con anterioridad se extrae la columna *Catálogo* que representa los modos de falla del equipo. La tabla 3.2 muestra los modos de falla de los chancadores en estudio.

3.5.1. Limpieza de datos

Dado que la columna *Catálogo* de la base de datos presenta algunas inconsistencias en cuanto a los nombres asignados a cada modo de falla, se procede a normalizarla en el sentido de asignar un único nombre o identificador a cada posible modo de falla. Considerando lo anterior, se eliminan los nombres duplicados que resultan de un mal tipeo o el uso o no uso de tildes en la escritura.

3.5.2. Filtros

Dado que se cuenta con 17 modos de falla y la cantidad de registros de falla utilizados con los algoritmos de *Machine Learning* es inferior a 500 es de esperar que existan modos de falla que se repiten mínimamente a lo largo de los casi 3 años considerados en los registros. Dada la imposibilidad de predecir adecuadamente con estos pocos registros, se consideran sólo los modos de falla que ocurren como mínimo 10 veces en el horizonte de tiempo considerado en la base de datos.

3.5.3. Balanceo de datos

Debido a que existen modos de falla que se repiten con mayor frecuencia que otros resulta necesario aplicar técnicas de balanceo de datos que hacen uso de técnicas de sobre muestreo y

Tabla 3.2: Modos de falla de los chancadores.

Catálogo
Base
Buzón Alimentación
Buzón Móvil
Cinta
Compuerta
Contraeje
Corazas
Estructuras
Excéntrica/Quicionera
Fuerza/Control
Instrumentación
Módulos/Parrillas
Motor
Silos llenos
Sistema Hidráulico
Sistema Lubricación
Transmisión

sub muestreo. Se utiliza SMOTE-Tomek Links para obtener un conjunto de datos balanceado que permite al algoritmo de Random Forest mostrar un verdadero poder predictivo y no entregar como resultado sólo la clase mayoritaria.

3.5.4. Ajuste de híper-parámetros

La técnica SMOTE requiere fijar una cantidad de registros para crear muestras sintéticas de las clases minoritarias. Se prueba utilizando desde 1 a 7 vecinos más cercanas para esto y se escoge la cantidad que entrega los mejores resultados en términos de exactitud.

Se hace uso de una "grid search" para determinar la cantidad de árboles de decisión, la profundidad de estos árboles y el criterio de pureza (gini o entropía) a utilizar.

3.5.5. Resultados

Se guardan los resultados del *Random Forest* sin considerar balanceo de datos y optimización de híper-parámetros con el objetivo de comparar los resultados cuando sí se utiliza una o ambas técnicas.

3.5.6. Entregables

A partir de librerías de *Python* se grafican todos los árboles de decisión del "Random Forest" y se guardan en un archivo *pdf*.

Se construye una matriz de confusión multiclase que permite extraer información relacionada a la precisión y exactitud del algoritmo.

La figura 3.2 resume gráficamente la metodología utilizada para generar los entregables.

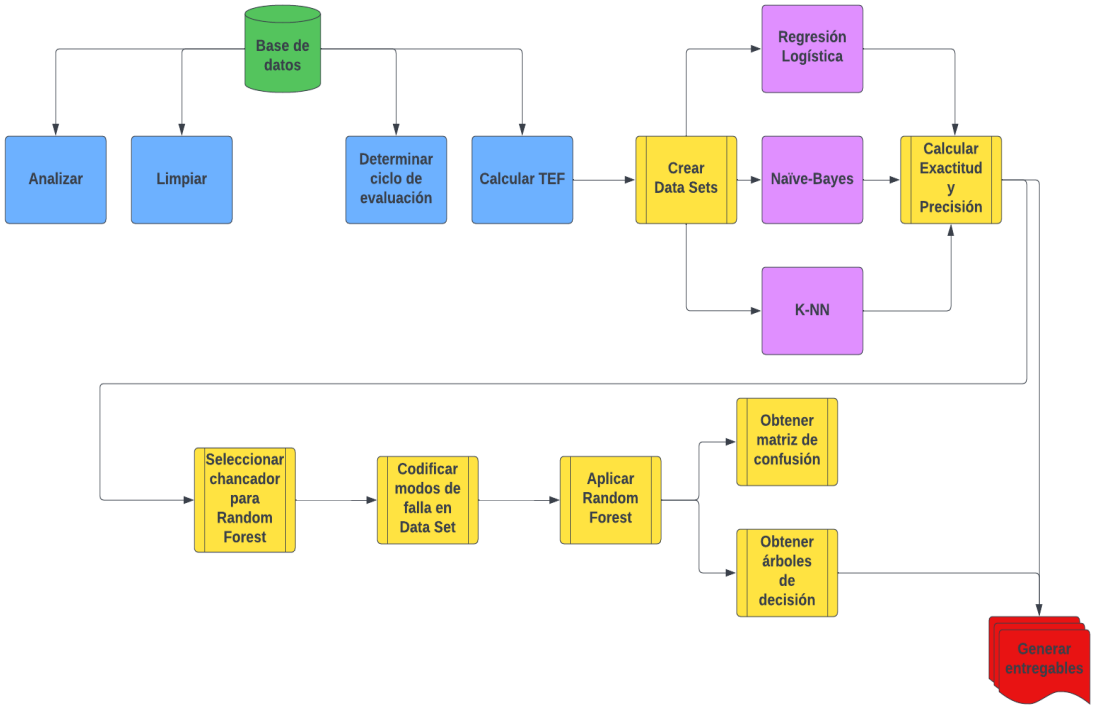


Figura 3.2: Diagrama de metodología a utilizar.

Capítulo 4

Resultados

A continuación se presentan los resultados extraídos para cada chancador de la planta de chancado secundario y terciario de la división Chuquicamata de Codelco. La figura 2.3 muestra la ubicación de cada uno de ellos al interior de la planta. En primer lugar se entraga el MTBF y MTTR, luego la regresión lineal de los TEF para determinar los parámetros de su distribución Weibull, luego los resultados de los tests de confianza aplicados, para luego analizar los resultados de las técnicas de *Machine Learning* en términos de su matriz de confusión considerando su respectivo ciclo de evaluación, finalmente se entregan los resultados de exactitud y precisión en determinar fallas. La tabla 4.1 referencia la sección donde se encuentran los resultados para cada chancador.

Tabla 4.1: Sección donde se encuentran los resultados para cada chancador.

Chancador	**Sección**
Chancador secundario MP1000	4.1
Chancador secundario Symons 7' sección B	4.2
Chancador secundario Hydrocone H8800	4.3
Chancador secundario Symons 7' sección D	4.4
Chancador secundario Symons 7' sección E	4.5
Chancador terciario 1	4.6
Chancador terciario 2	4.7
Chancador terciario 3	4.8
Chancador terciario 4	4.9
Chancador terciario 5	4.10
Chancador terciario 6	4.11
Chancador terciario 7	4.12
Chancador terciario 8	4.13
Chancador terciario 9	4.14
Chancador terciario 10	4.15

4.1. Chancador secundario MP1000

Tras filtrar y limpiar los datos se obtienen 246 registros de falla y por ende 245 TEF calculados. La figura 4.1 muestra la regresión lineal que se realiza con los TEF. Se observa que el coeficiente de determinación (R^2) es de 0,991.

Además, considerando los 245 TEF calculados se tiene que:

$$MTBF = 65,4\ [hrs]$$

y el tiempo medio de reparación corresponde a:

$$MTTR = 3\ [hrs]$$

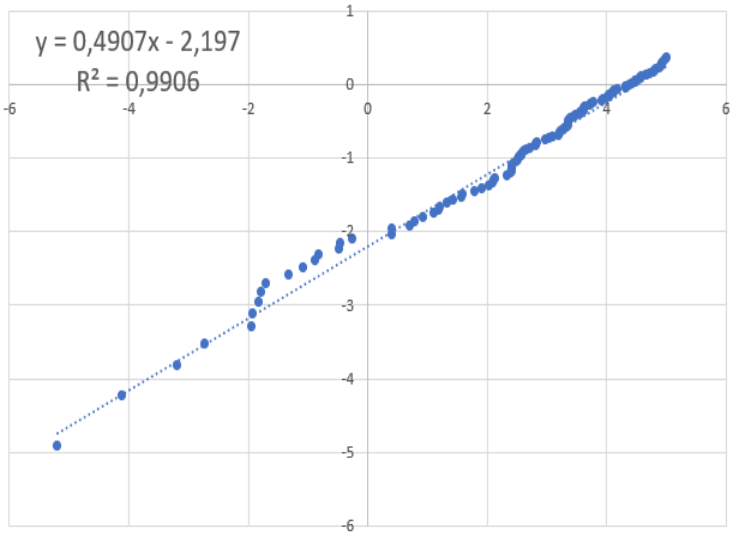


Figura 4.1: Regresión lineal.

En la figura 4.2 se observa el ajuste de los datos a una distribución Weibull. Por otro lado, en la tabla 4.2 se muestran los estadísticos de prueba que se obtienen para cada test de confianza aplicado y en la tabla 4.3 se observan los niveles de confianza en donde la hipótesis nula que indica que los datos siguen una distribución Weibull es rechazada o no.

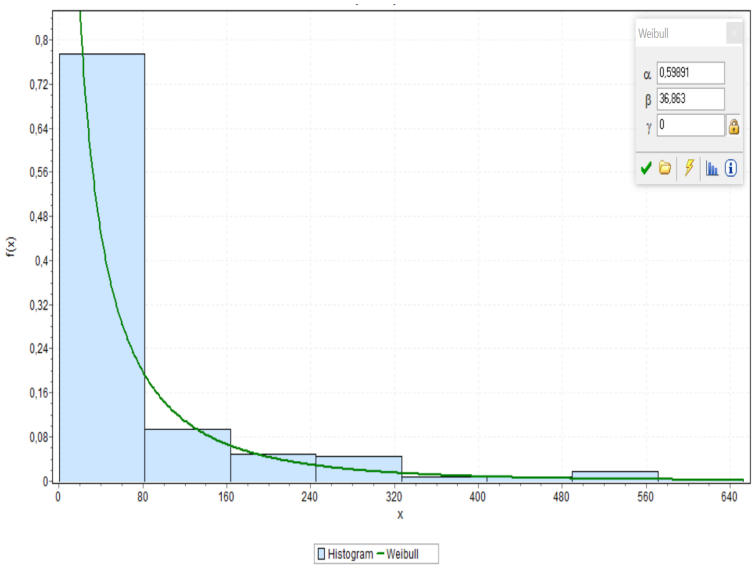


Figura 4.2: Ajuste distribución Weibull de los registros de falla del chancador MP1000.

Tabla 4.2: Estadísticos de prueba para tests de confianza de tiempos entre fallas de chancador MP1000.

Test	**Kolmogorov-Smirnov**	**Anderson-Darling**	χ^2
Estadístico	0,087	2,880	17,654

Tabla 4.3: Resultados de tests de bondad de ajuste para tiempos entre fallas de chancador MP1000.

Kolmogorov-Smirnov					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	0,069	0,078	0,087	0,097	0,104
¿Rechazar?	Sí	Sí	No	No	No
Anderson-Darling					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	1,378	1,929	2,502	3,289	3,907
¿Rechazar?	Sí	Sí	Sí	No	No
χ^2					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	9,803	12,017	14,067	16,622	18,475
¿Rechazar?	Sí	Sí	No	No	No

La tabla 4.4 muestra los parámetros de la distribución Weibull que siguen los tiempos entre falla determinados para el chancador secundario MP1000.

Se utilizan 3 ciclos de evaluación de 4, 6 y 12 horas. Los ciclos de evaluación indican la forma

Tabla 4.4: Parámetros distribución Weibull para tiempos entre fallas de chancador MP1000.

Parámetro	**Valor**
β	0,599
η	36,863
γ	0

en que se distribuirán los ceros en el conjunto de datos que será cargado en los algoritmos de *Machine Learning*.

4.1.1. Ciclo de evaluación de 4 horas

Se obtienen 164 registros los cuales se etiquetan con un 0 y corresponden a tiempos en los cuales el equipo no fallará. La figura 4.3 muestra cómo se distribuyen los 0 y 1 en el conjunto de datos.

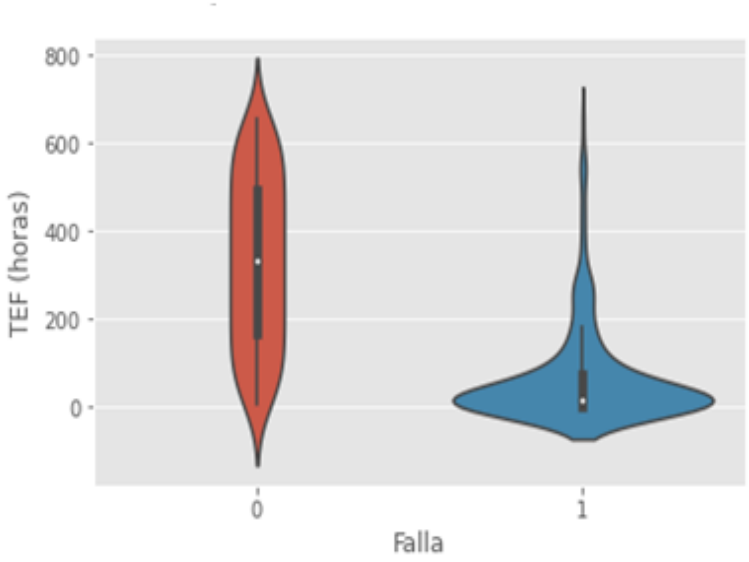


Figura 4.3: Diagrama de violín para chancador secundario MP1000 usando ciclo de evaluación de 4 horas.

La figura 4.4 muestra la matriz de confusión resultante al aplicar una regresión logística a los datos considerando un ciclo de evaluación de 4 horas.

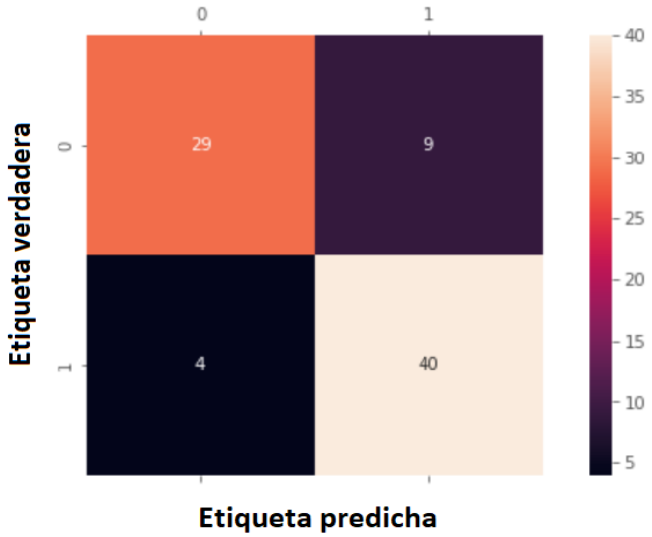


Figura 4.4: Matriz de confusión para chancador secundario MP1000 usando ciclo de evaluación de 4 horas.

$$Exactitud = \frac{29 + 40}{29 + 40 + 9 + 4} = \frac{49}{62} = 0,79 = 79\,\%$$

$$Precisión = \frac{40}{44} = 0,91 = 91\,\%$$

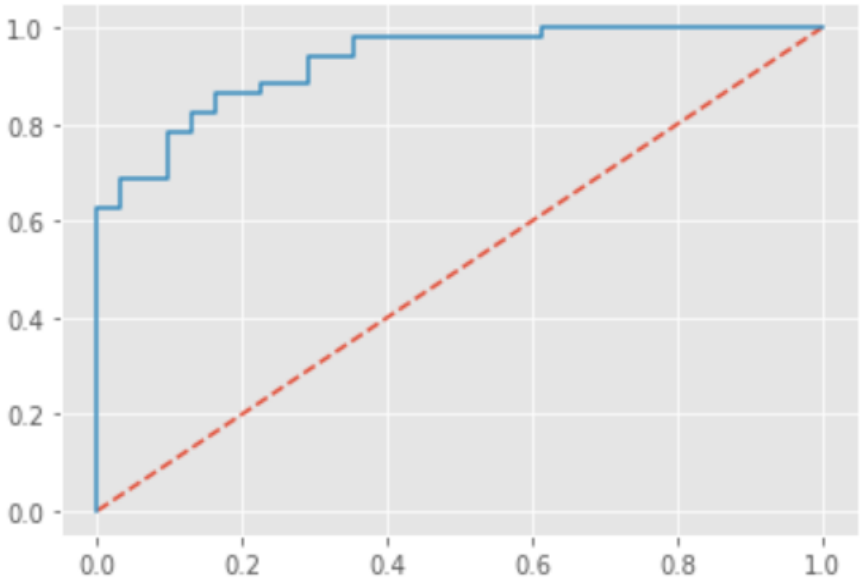


Figura 4.5: Curva ROC para chancador secundario MP1000 usando ciclo de evaluación de 4 horas.

$$AUC = 0,93$$

La figura 4.6 muestra la matriz de confusión resultante al aplicar el algoritmo Naïve-Bayes a los datos considerando un ciclo de evaluación de 4 horas.

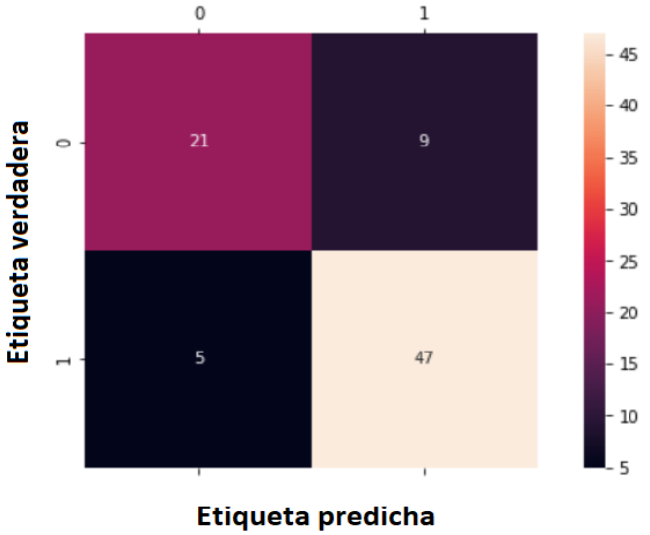


Figura 4.6: Matriz de confusión para chancador secundario MP1000 usando ciclo de evaluación de 4 horas.

$$Exactitud = \frac{21+47}{21+47+9+5} = \frac{68}{82} = 0,83 = 83\,\%$$

$$Precisión = \frac{47}{52} = 0,90 = 90\,\%$$

La figura 4.7 muestra la matriz de confusión resultante al aplicar el algoritmo K-NN a los datos considerando un ciclo de evaluación de 4 horas.

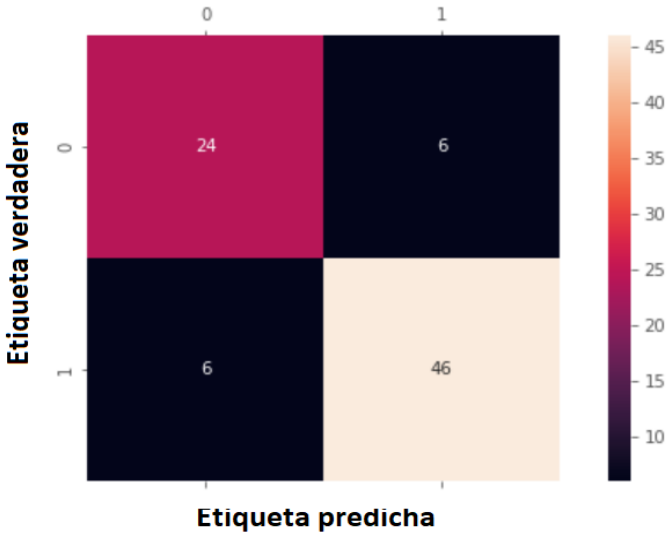


Figura 4.7: Matriz de confusión para chancador secundario MP1000 usando ciclo de evaluación de 4 horas.

$$Exactitud = \frac{24 + 46}{24 + 46 + 6 + 6} = \frac{70}{82} = 0,85 = 85\,\%$$

$$Precisión = \frac{46}{52} = 0,88 = 88\,\%$$

4.1.2. Ciclo de evaluación de 6 horas

Se obtienen 109 registros los cuales se etiquetan con un 0 y corresponden a tiempos en los cuales el equipo no fallará. La figura 4.8 muestra cómo se distribuyen los 0 y 1 en el conjunto de datos.

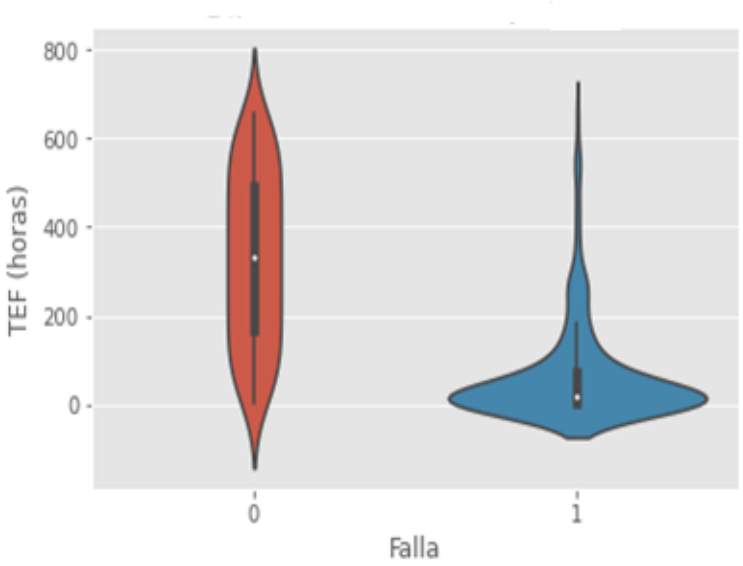


Figura 4.8: Diagrama de violín para chancador secundario MP1000 usando ciclo de evaluación de 6 horas.

La figura 4.9 muestra la matriz de confusión resultante al aplicar una regresión logística a los datos considerando un ciclo de evaluación de 6 horas.

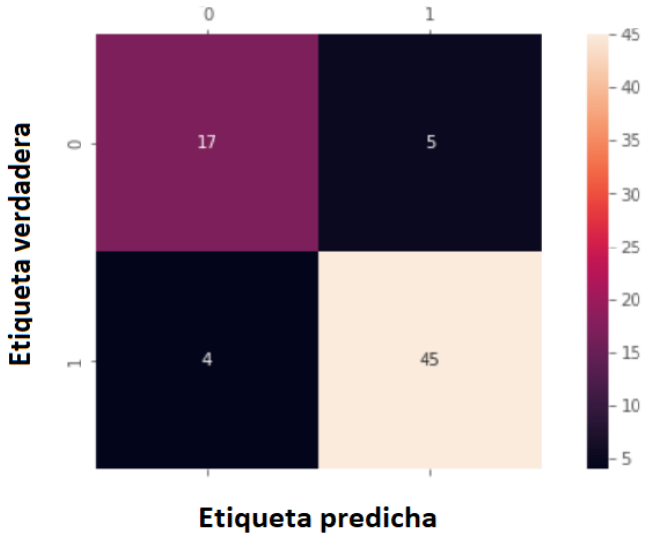


Figura 4.9: Matriz de confusión para chancador secundario MP1000 usando ciclo de evaluación de 6 horas.

$$Exactitud = \frac{17 + 45}{17 + 45 + 5 + 4} = \frac{62}{71} = 0,87 = 87\,\%$$

$$Precisión = \frac{45}{49} = 0,92 = 92\,\%$$

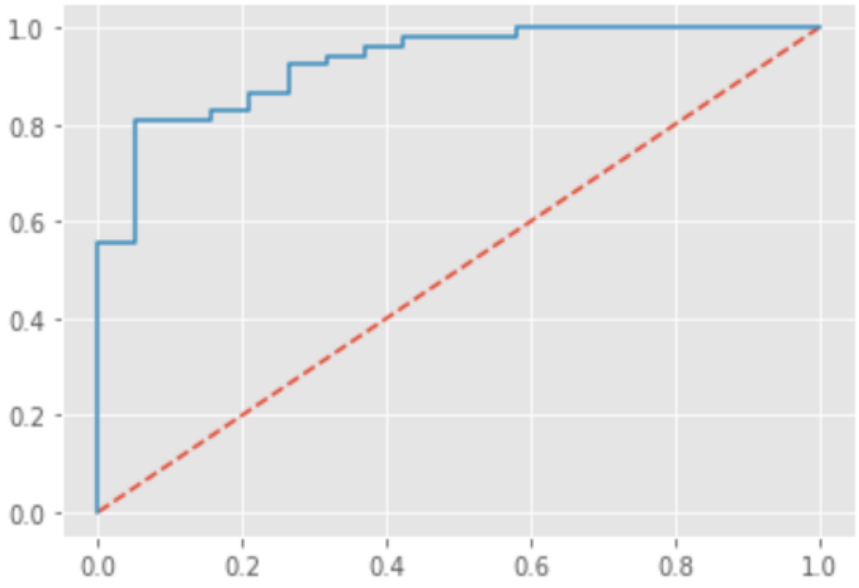


Figura 4.10: Curva ROC para chancador secundario MP1000 usando ciclo de evaluación de 6 horas.

$$AUC = 0,93$$

La figura 4.11 muestra la matriz de confusión resultante al aplicar el algoritmo Naïve-Bayes a los datos considerando un ciclo de evaluación de 6 horas.

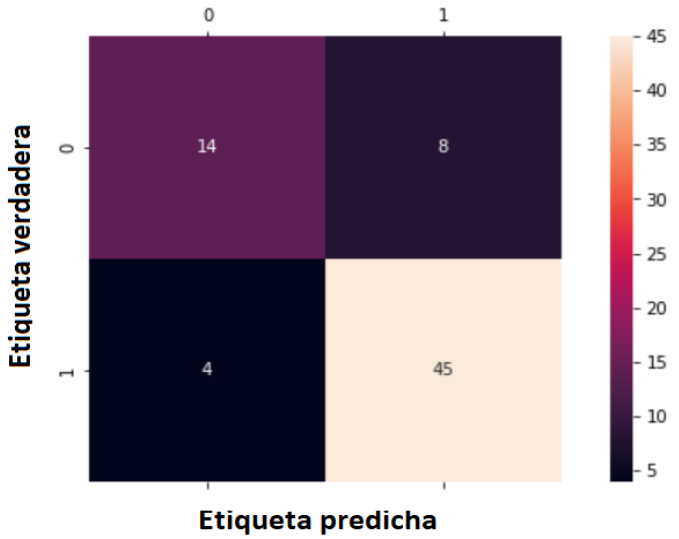


Figura 4.11: Matriz de confusión para chancador secundario MP1000 usando ciclo de evaluación de 6 horas.

$$Exactitud = \frac{14+45}{14+45+8+4} = \frac{59}{71} = 0,83 = 83\,\%$$

$$Precisión = \frac{45}{49} = 0,92 = 92\,\%$$

La figura 4.12 muestra la matriz de confusión resultante al aplicar el algoritmo K-NN a los datos considerando un ciclo de evaluación de 6 horas.

Figura 4.12: Matriz de confusión para chancador secundario MP1000 usando ciclo de evaluación de 6 horas.

$$Exactitud = \frac{17+38}{17+38+5+11} = \frac{55}{71} = 0,77 = 77\,\%$$

$$Precisión = \frac{38}{49} = 0,78 = 78\,\%$$

4.1.3. Ciclo de evaluación de 12 horas

Se obtienen 55 registros los cuales se etiquetan con un 0 y corresponden a tiempos en los cuales el equipo no fallará. La figura 4.13 muestra cómo se distribuyen los 0 y 1 en el conjunto de datos.

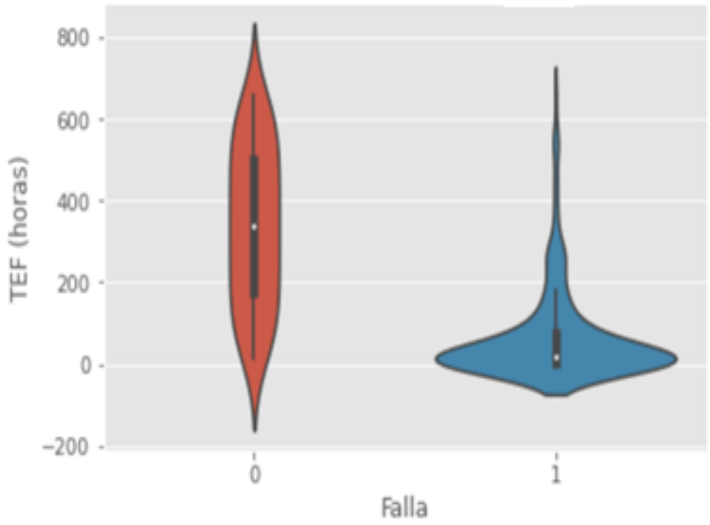


Figura 4.13: Diagrama de violín para chancador secundario MP1000 usando ciclo de evaluación de 12 horas.

La figura 4.14 muestra la matriz de confusión resultante al aplicar una regresión logística a los datos considerando un ciclo de evaluación de 12 horas.

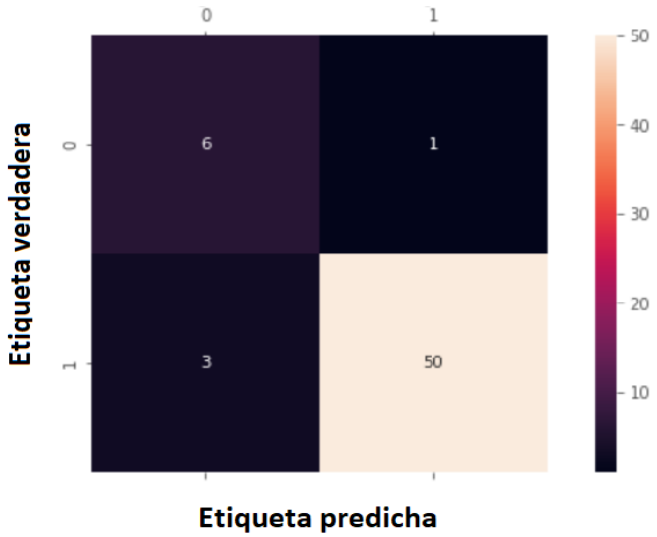


Figura 4.14: Matriz de confusión para chancador secundario MP1000 usando ciclo de evaluación de 12 horas.

$$Exactitud = \frac{6+50}{6+50+1+3} = \frac{56}{60} = 0,93 = 93\,\%$$

$$Precisión = \frac{50}{53} = 0,94 = 94\,\%$$

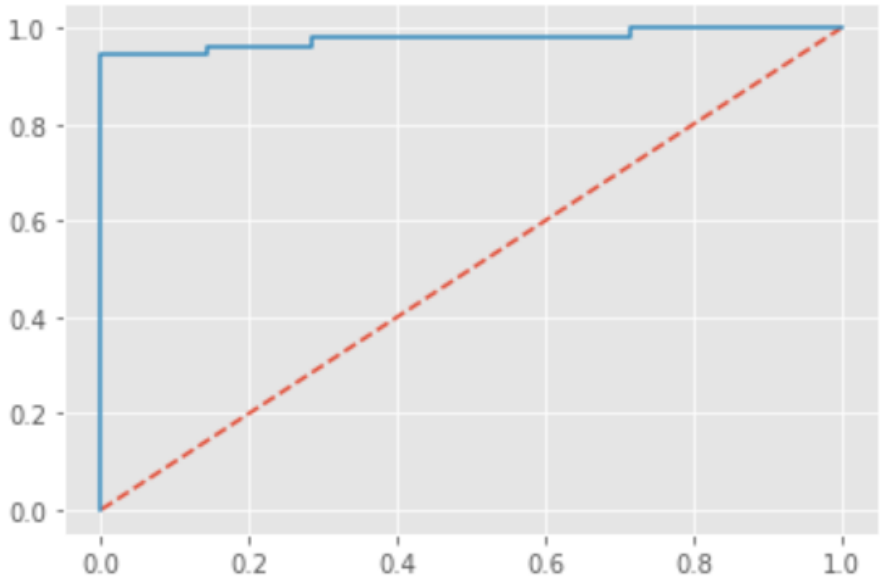


Figura 4.15: Curva ROC para chancador secundario MP1000 usando ciclo de evaluación de 12 horas.

$$AUC = 0,98$$

La figura 4.16 muestra la matriz de confusión resultante al aplicar el algoritmo Naïve-Bayes a los datos considerando un ciclo de evaluación de 12 horas.

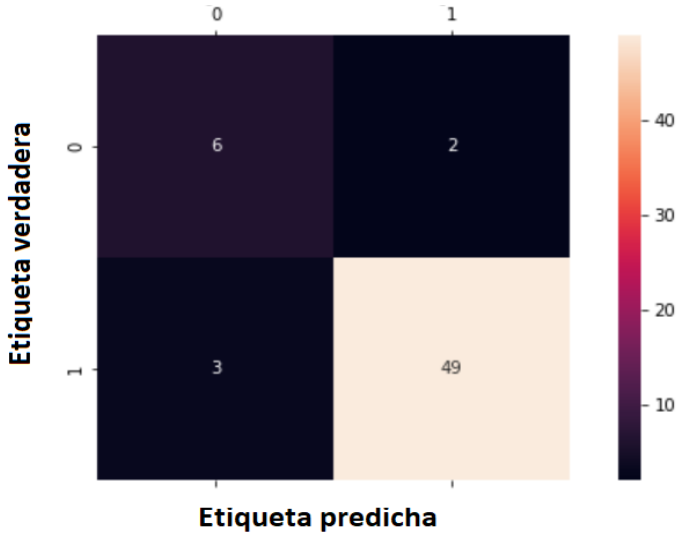


Figura 4.16: Matriz de confusión para chancador secundario MP1000 usando ciclo de evaluación de 12 horas.

$$Exactitud = \frac{6+49}{6+49+2+3} = \frac{55}{60} = 0,92 = 92\,\%$$

$$Precisión = \frac{49}{52} = 0,94 = 94\,\%$$

La figura 4.17 muestra la matriz de confusión resultante al aplicar el algoritmo K-NN a los datos considerando un ciclo de evaluación de 12 horas.

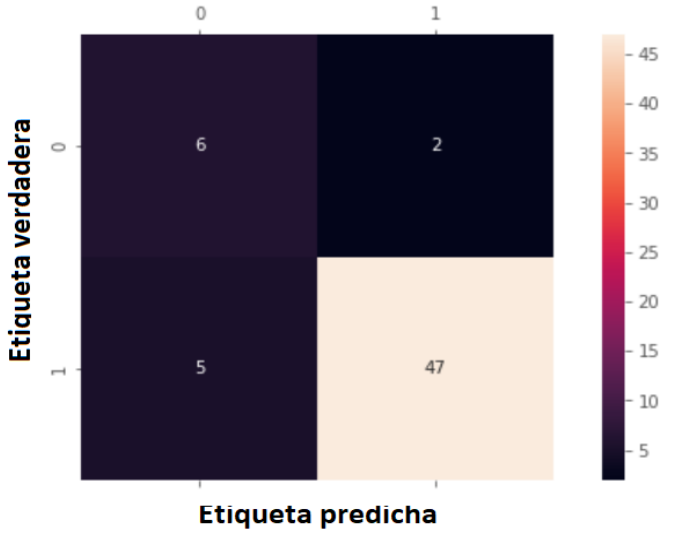


Figura 4.17: Matriz de confusión para chancador secundario MP1000 usando ciclo de evaluación de 12 horas.

$$Exactitud = \frac{6+47}{6+47+2+5} = \frac{53}{60} = 0,88 = 88\,\%$$

$$Precisión = \frac{47}{52} = 0,90 = 90\,\%$$

4.2. Chancador secundario Symnons 7' Sección B

Tras filtrar y limpiar los datos se obtienen 108 registros de falla y por ende 107 TEF calculados. La figura 4.18 muestra la regresión lineal que se realiza con los TEF. Se observa que el coeficiente R^2 es de 0,968.

Además, considerando los 107 TEF calculados se tiene que:

$$MTBF = 188,3\ [hrs]$$

y el tiempo medio de reparación corresponde a:

$$MTTR = 2,1\ [hrs]$$

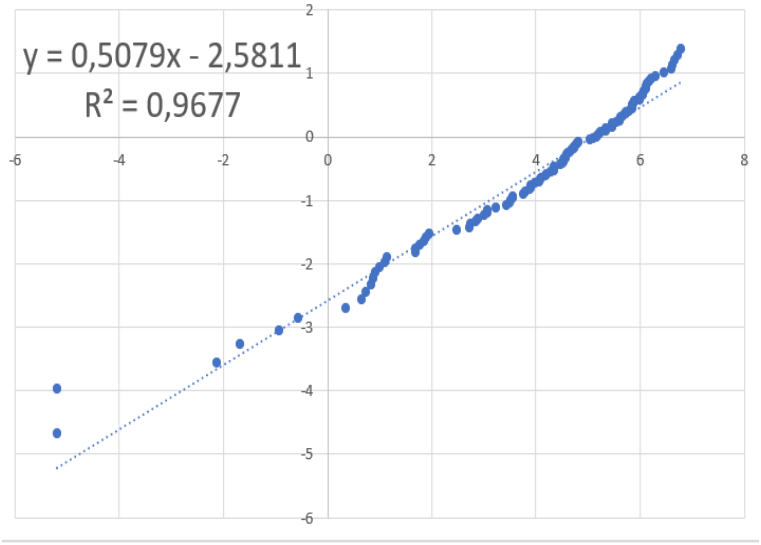


Figura 4.18: Regresión lineal.

En la figura 4.19 se observa el ajuste de los datos a una distribución Weibull. Por otro lado, en la tabla 4.5 se muestran los estadísticos de prueba que se obtienen para cada test de confianza aplicado y en la tabla 4.6 se observan los niveles de confianza en donde la hipótesis nula que indica que los datos siguen una distribución Weibull es rechazada o no.

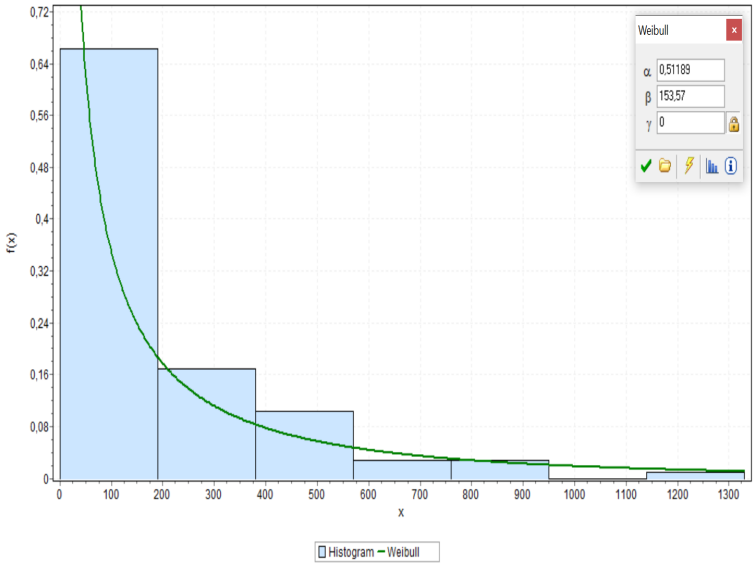


Figura 4.19: Ajuste distribución Weibull de los registros de falla del chancador Symons de sección B.

Tabla 4.5: Tests de confianza para tiempos entre fallas de chancador MP1000.

Test	**Kolmogorov-Smirnov**	**Anderson-Darling**	χ^2
Estadístico	0,085	1,396	6,606

Tabla 4.6: Resultados de tests de bondad de ajuste para tiempos entre fallas de chancador Symons 7' de sección B.

Kolmogorov-Smirnov					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	0,104	0,118	0,131	0,147	0,157
¿Rechazar?	No	No	No	No	No
Anderson-Darling					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	1,375	1,929	2,502	3,289	3,907
¿Rechazar?	Sí	No	No	No	No
χ^2					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	8,558	10,645	12,592	15,033	16,812
¿Rechazar?	No	No	No	No	No

La tabla 4.7 muestra los parámetros de la distribución Weibull que siguen los tiempos entre falla determinados para el chancador secundario Symons 7' de la sección B de la planta.

Dada la distribución de los TEF se utiliza un ciclo de evaluación de 8 horas.

Tabla 4.7: Parámetros distribución Weibull para tiempos entre fallas de chancador MP1000.

Parámetro	**Valor**
β	0,512
η	153,570
γ	0

4.2.1. Ciclo de evaluación de 8 horas

Se obtienen 167 registros los cuales se etiquetan con un 0 y corresponden a tiempos en los cuales el equipo no fallará. La figura 4.20 muestra cómo se distribuyen los 0 y 1 en el conjunto de datos.

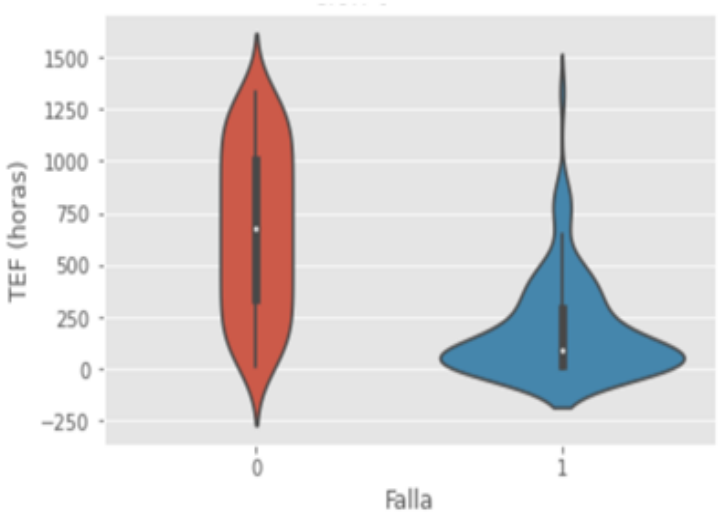


Figura 4.20: Diagrama de violín para chancador secundario Symons de sección B.

La figura 4.21 muestra la matriz de confusión resultante al aplicar una regresión logística a los datos considerando un ciclo de evaluación de 8 horas.

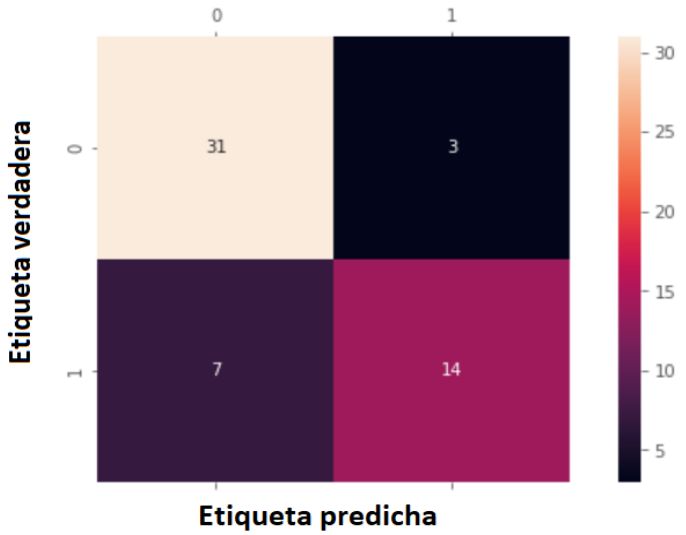


Figura 4.21: Matriz de confusión para chancador secundario Symons de sección B.

$$Exactitud = \frac{31+14}{31+14+3+7} = \frac{45}{55} = 0,82 = 82\,\%$$

$$Precisión = \frac{14}{21} = 0,67 = 67\,\%$$

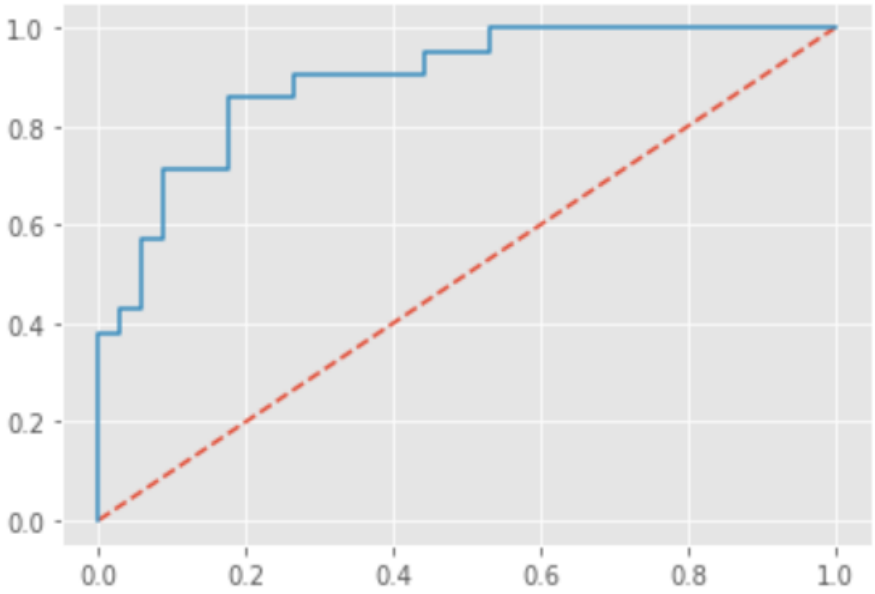


Figura 4.22: Curva ROC para chancador secundario Symons de sección B.

$$AUC = 0,89$$

La figura 4.23 muestra la matriz de confusión resultante al aplicar el algoritmo Naïve-Bayes a los datos considerando un ciclo de evaluación de 8 horas.

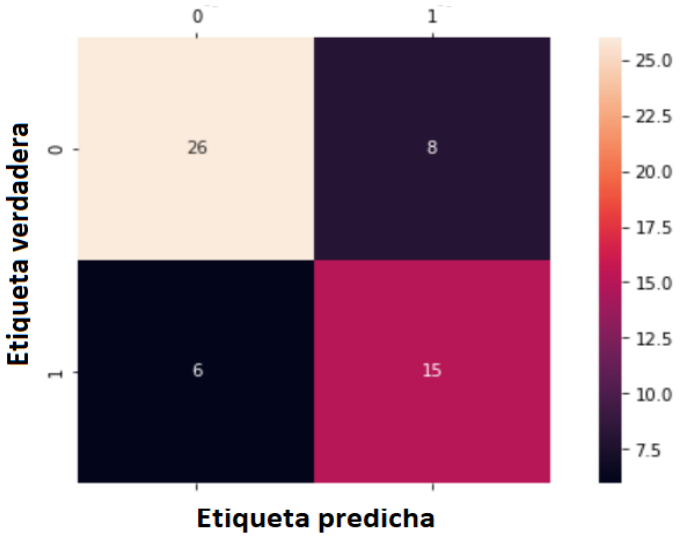


Figura 4.23: Matriz de confusión para chancador secundario Symons de sección B.

$$Exactitud = \frac{26+15}{26+15+8+6} = \frac{41}{55} = 0,75 = 75\,\%$$

$$Precisión = \frac{15}{21} = 0,71 = 71\,\%$$

La figura 4.24 muestra la matriz de confusión resultante al aplicar el algoritmo K-NN a los datos considerando un ciclo de evaluación de 8 horas.

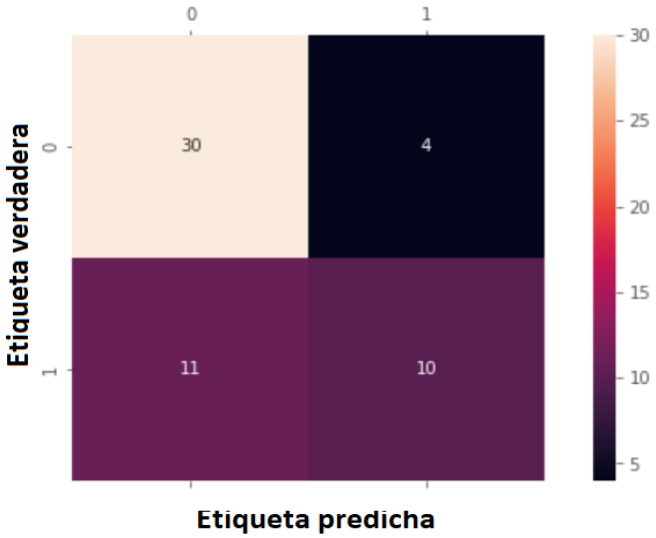


Figura 4.24: Matriz de confusión para chancador secundario Symons de sección B.

$$Exactitud = \frac{30+10}{30+10+4+11} = \frac{40}{55} = 0,73 = 73\,\%$$

$$Precisión = \frac{10}{21} = 0,48 = 48\,\%$$

4.3. Chancador secundario Hydrocone H8800

Tras filtrar y limpiar los datos se obtienen 190 registros de falla y por ende 189 TEF calculados. La figura 4.25 muestra la regresión lineal que se realiza con los TEF. Se observa que el coeficiente R^2 es de 0,980.

Además, considerando los 189 TEF calculados se tiene que:

$$MTBF = 99,4\ [hrs]$$

y el tiempo medio de reparación corresponde a:

$$MTTR = 1,7\ [hrs]$$

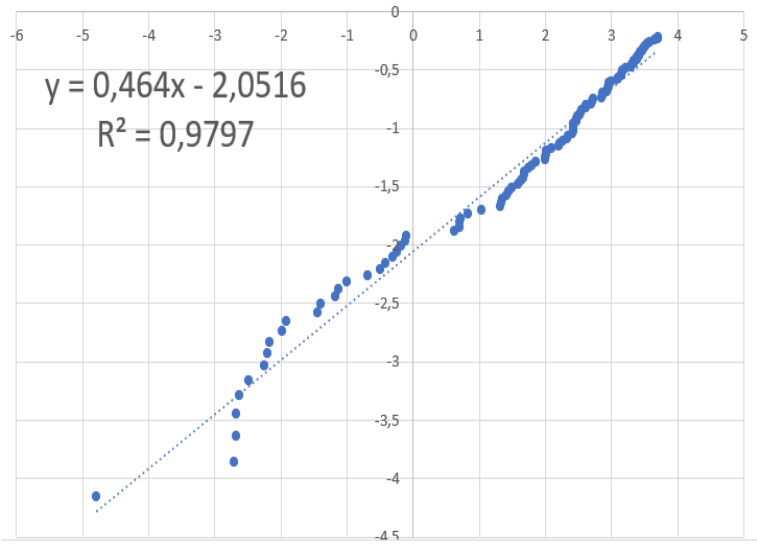


Figura 4.25: Regresión lineal.

En la figura 4.26 se observa el ajuste de los datos a una distribución Weibull. Por otro lado, en la tabla 4.8 se muestran los estadísticos de prueba que se obtienen para cada test de confianza aplicado y en la tabla 4.9 se observan los niveles de confianza en donde la hipótesis nula que indica que los datos siguen una distribución Weibull es rechazada o no.

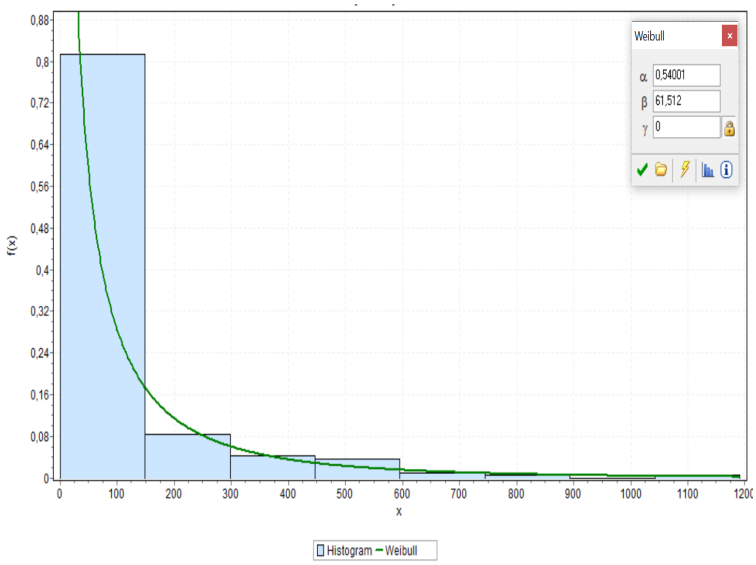


Figura 4.26: Ajuste distribución Weibull de los registros de falla del chancador Hydrocone H8800.

Tabla 4.8: Tests de confianza para tiempos entre fallas de chancador Hydrocone H8800.

Test	**Kolmogorov-Smirnov**	**Anderson-Darling**	χ^2
Estadístico	0,043	0,361	4,030

Tabla 4.9: Resultados de tests de bondad de ajuste para tiempos entre fallas de chancador Hydrocone H8800.

Kolmogorov-Smirnov					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	0,078	0,089	0,099	0,110	0,118
¿Rechazar?	No	No	No	No	No
Anderson-Darling					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	1,375	1,929	2,502	3,289	3,907
¿Rechazar?	No	No	No	No	No
χ^2					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	9,803	12,017	14,067	16,622	18,475
¿Rechazar?	Sí	Sí	No	No	No

La tabla 4.10 muestra los parámetros de la distribución Weibull que siguen los tiempos entre falla determinados para el chancador secundario Symons 7' de la sección B de la planta.

Dada la distribución de los TEF se utiliza un ciclo de evaluación de 16 horas.

Tabla 4.10: Parámetros distribución Weibull para tiempos entre fallas de chancador MP1000.

Parámetro	Valor
β	0,540
η	61,512
γ	0

4.3.1. Ciclo de evaluación de 16 horas

Se obtienen 75 registros los cuales se etiquetan con un 0 y corresponden a tiempos en los cuales el equipo no fallará. La figura 4.27 muestra cómo se distribuyen los 0 y 1 en el conjunto de datos.

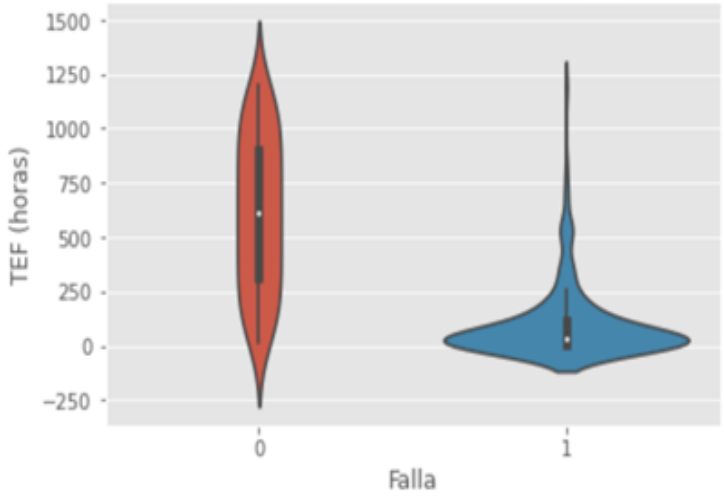


Figura 4.27: Diagrama de violín para chancador secundario Hydrocone H8800.

La figura 4.28 muestra la matriz de confusión resultante al aplicar una regresión logística a los datos considerando un ciclo de evaluación de 16 horas.

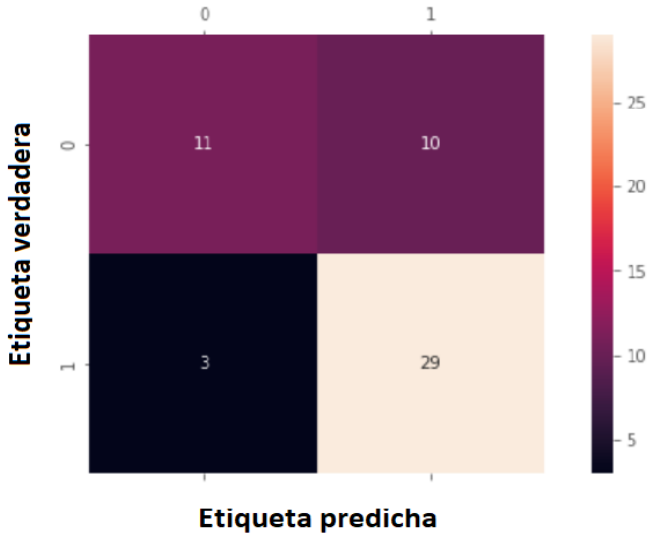


Figura 4.28: Matriz de confusión para chancador secundario Hydrocone H8800.

$$Exactitud = \frac{11 + 29}{11 + 29 + 10 + 3} = \frac{40}{53} = 0,75 = 75\,\%$$

$$Precisión = \frac{29}{32} = 0,91 = 91\,\%$$

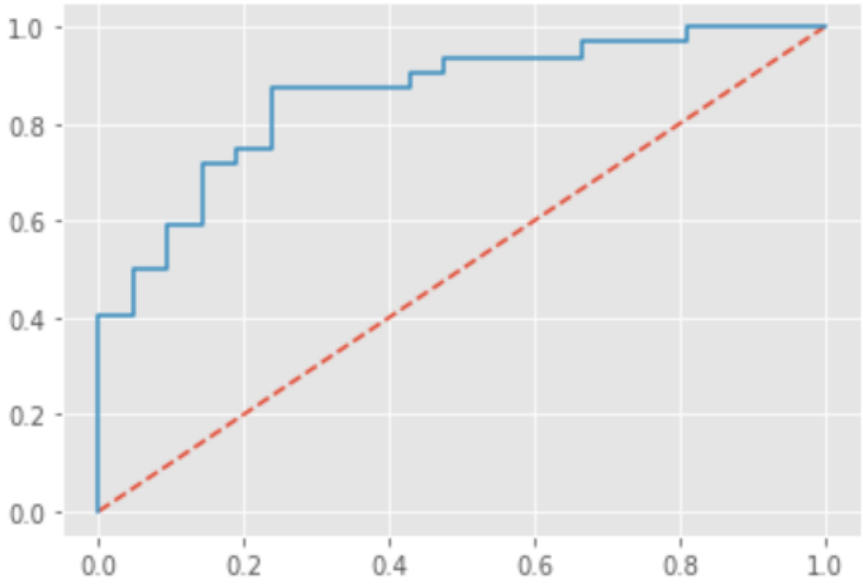


Figura 4.29: Curva ROC para chancador secundario Hydrocone H8800.

$$AUC = 0,86$$

La figura 4.30 muestra la matriz de confusión resultante al aplicar el algoritmo Naïve-Bayes a los datos considerando un ciclo de evaluación de 16 horas.

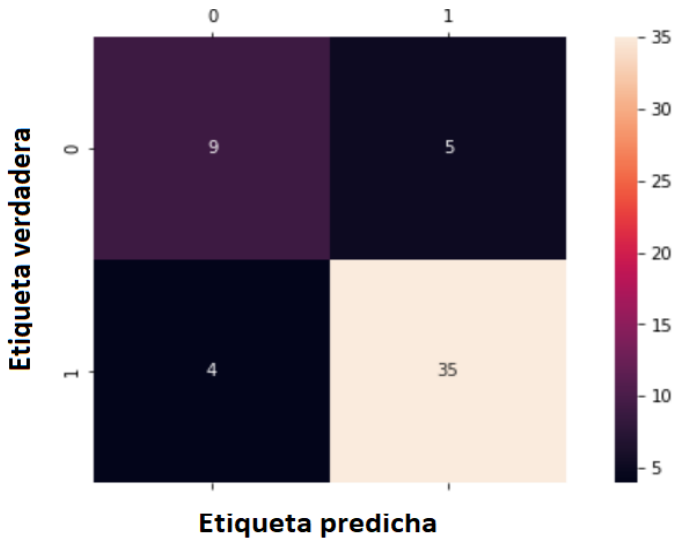


Figura 4.30: Matriz de confusión para chancador secundario Hydrocone H8800.

$$Exactitud = \frac{9+35}{9+35+5+4} = \frac{44}{53} = 0,83 = 83\,\%$$

$$Precisión = \frac{35}{39} = 0,90 = 90\,\%$$

La figura 4.31 muestra la matriz de confusión resultante al aplicar el algoritmo K-NN a los datos considerando un ciclo de evaluación de 16 horas.

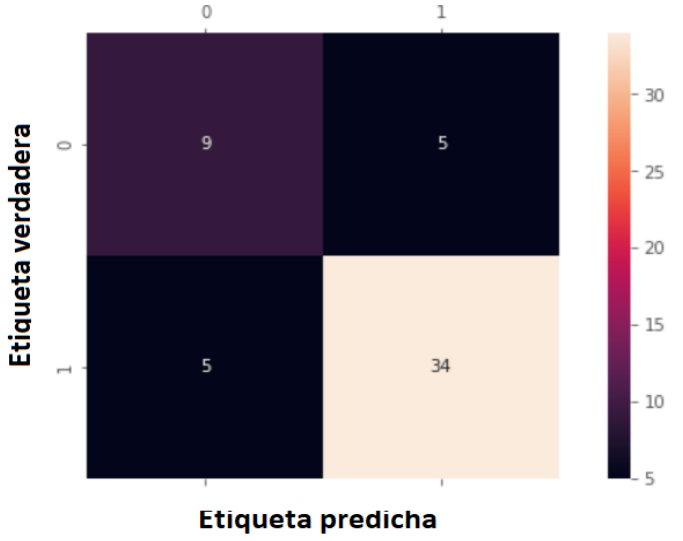


Figura 4.31: Matriz de confusión para chancador secundario Hydrocone H8800.

$$Exactitud = \frac{9+34}{9+34+5+5} = \frac{43}{53} = 0,81 = 81\,\%$$

$$Precisión = \frac{34}{39} = 0,87 = 87\,\%$$

4.4. Chancador secundario Symons 7' Sección D

Tras filtrar y limpiar los datos se obtienen 176 registros de falla y por ende 175 TEF calculados. La figura 4.32 muestra la regresión lineal que se realiza con los TEF. Se observa que el coeficiente R^2 es de 0,984.

Además, considerando los 175 TEF calculados se tiene que:

$$MTBF = 117,1\ [hrs]$$

y el tiempo medio de reparación corresponde a:

$$MTTR = 1,7\ [hrs]$$

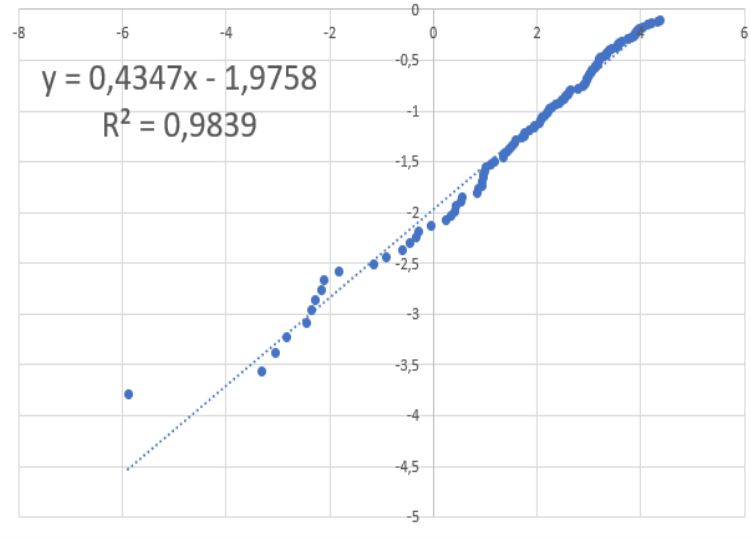


Figura 4.32: Regresión lineal.

En la figura 4.33 se observa el ajuste de los datos a una distribución Weibull. Por otro lado, en la tabla 4.11 se muestran los estadísticos de prueba que se obtienen para cada test de confianza aplicado y en la tabla 4.12 se observan los niveles de confianza en donde la hipótesis nula que indica que los datos siguen una distribución Weibull es rechazada o no.

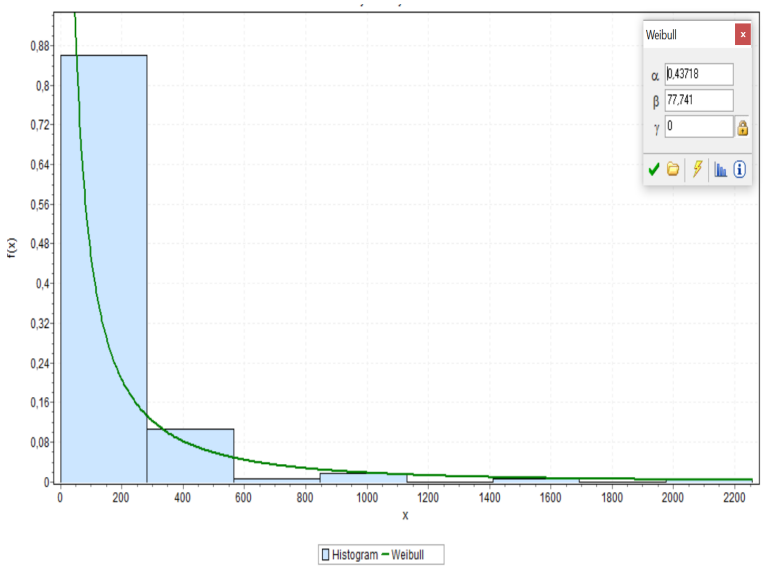


Figura 4.33: Ajuste distribución Weibull de los registros de falla del chancador Symons de sección D.

Tabla 4.11: Tests de confianza para tiempos entre fallas de chancador Symons de sección D.

Test	**Kolmogorov-Smirnov**	**Anderson-Darling**	χ^2
Estadístico	0,061	1,080	12,818

Tabla 4.12: Resultados de tests de bondad de ajuste para tiempos entre fallas de chancador Symons 7' de sección D.

Kolmogorov-Smirnov					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	0,080	0,091	0,101	0,113	0,121
¿Rechazar?	No	No	No	No	No
Anderson-Darling					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	1,375	1,929	2,502	3,289	3,907
¿Rechazar?	No	No	No	No	No
χ^2					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	9,803	12,017	14,067	16,622	18,475
¿Rechazar?	Sí	Sí	No	No	No

La tabla 4.13 muestra los parámetros de la distribución Weibull que siguen los tiempos entre falla determinados para el chancador secundario Symons 7' de la sección D de la planta.

Dada la distribución de los TEF se utiliza un ciclo de evaluación de 15 horas.

Tabla 4.13: Parámetros distribución Weibull para tiempos entre fallas de chancador Symons de sección D.

Parámetro	Valor
β	0,437
η	77,741
γ	0

4.4.1. Ciclo de evaluación de 15 horas

Se obtienen 73 registros los cuales se etiquetan con un 0 y corresponden a tiempos en los cuales el equipo no fallará. La figura 4.34 muestra cómo se distribuyen los 0 y 1 en el conjunto de datos.

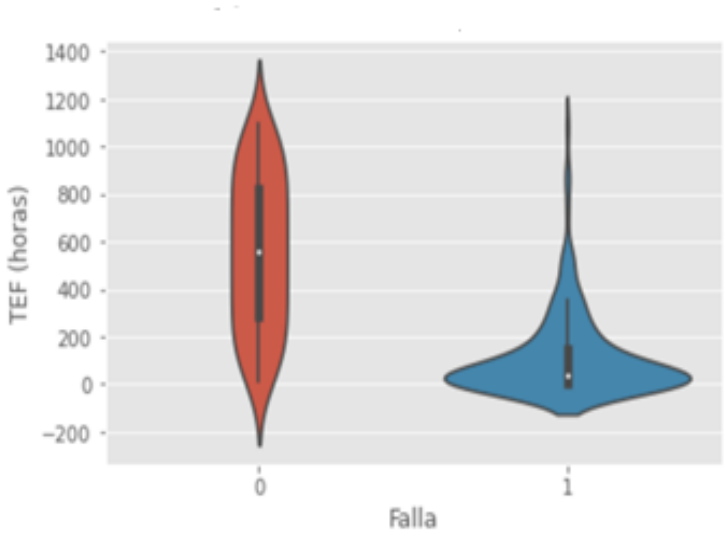


Figura 4.34: Diagrama de violín para chancador secundario Symons sección D.

La figura 4.35 muestra la matriz de confusión resultante al aplicar una regresión logística a los datos considerando un ciclo de evaluación de 15 horas.

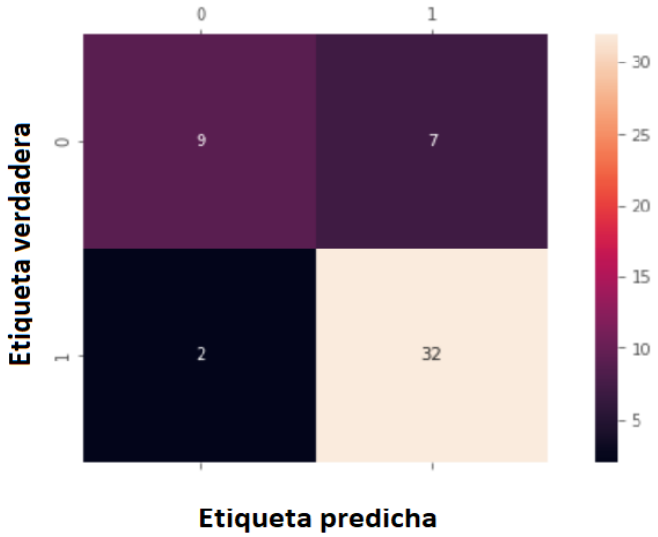


Figura 4.35: Matriz de confusión para chancador secundario Symons sección D.

$$Exactitud = \frac{9+32}{9+32+7+2} = \frac{41}{50} = 0,82 = 82\,\%$$

$$Precisión = \frac{32}{34} = 0,94 = 94\,\%$$

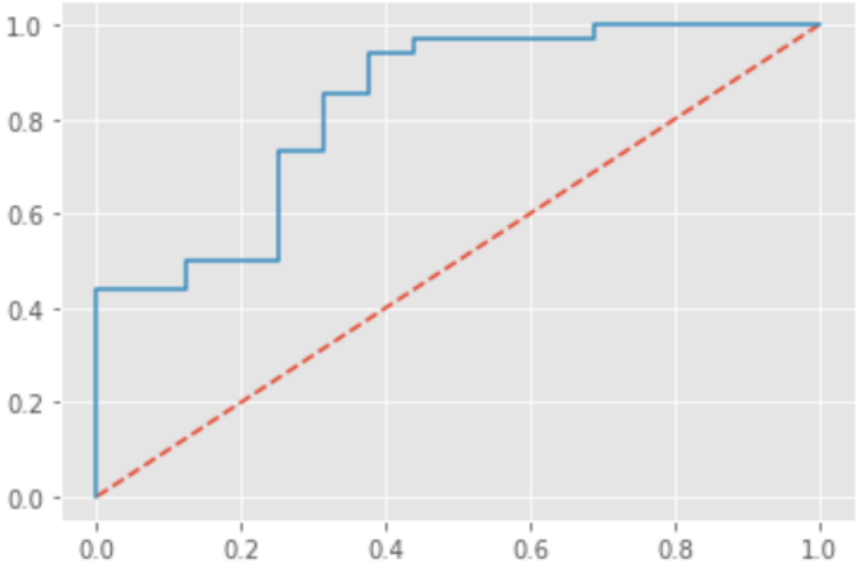


Figura 4.36: Curva ROC para chancador secundario Symons sección D.

$$AUC = 0,83$$

La figura 4.37 muestra la matriz de confusión resultante al aplicar el algoritmo Naïve-Bayes a los datos considerando un ciclo de evaluación de 16 horas.

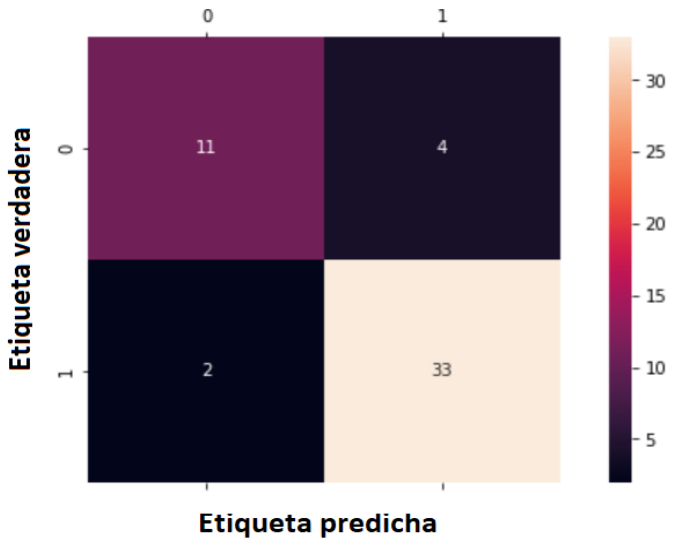


Figura 4.37: Matriz de confusión para chancador secundario Symons sección D.

$$Exactitud = \frac{11+33}{11+33+4+2} = \frac{44}{50} = 0,88 = 88\,\%$$

$$Precisión = \frac{33}{35} = 0,94 = 94\,\%$$

La figura 4.38 muestra la matriz de confusión resultante al aplicar el algoritmo K-NN a los datos considerando un ciclo de evaluación de 16 horas.

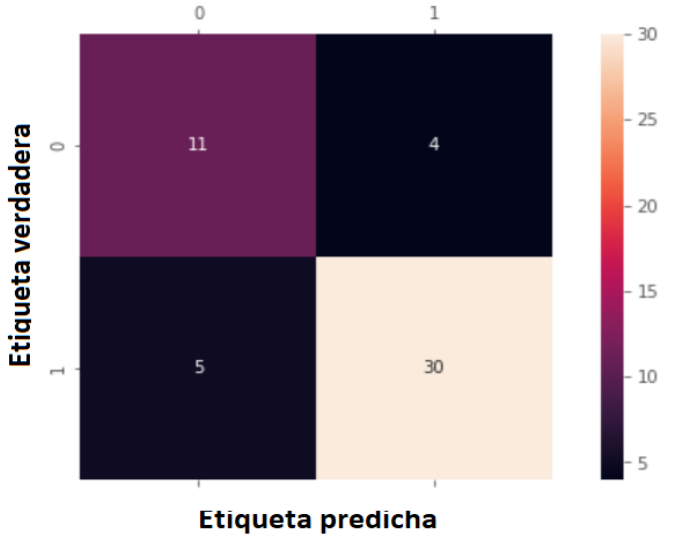


Figura 4.38: Matriz de confusión para chancador secundario Symons sección D.

$$Exactitud = \frac{11+30}{11+30+4+5} = \frac{41}{50} = 0,82 = 82\,\%$$

$$Precisión = \frac{30}{35} = 0,86 = 86\,\%$$

4.5. Chancador secundario Symons 7' Sección E

Tras filtrar y limpiar los datos se obtienen 154 registros de falla y por ende 153 TEF calculados. La figura 4.39 muestra la regresión lineal que se realiza con los TEF. Se observa que el coeficiente R^2 es de 0,991.

Además, considerando los 153 TEF calculados se tiene que:

$$MTBF = 118,9\ [hrs]$$

y el tiempo medio de reparación corresponde a:

$$MTTR = 1,5\ [hrs]$$

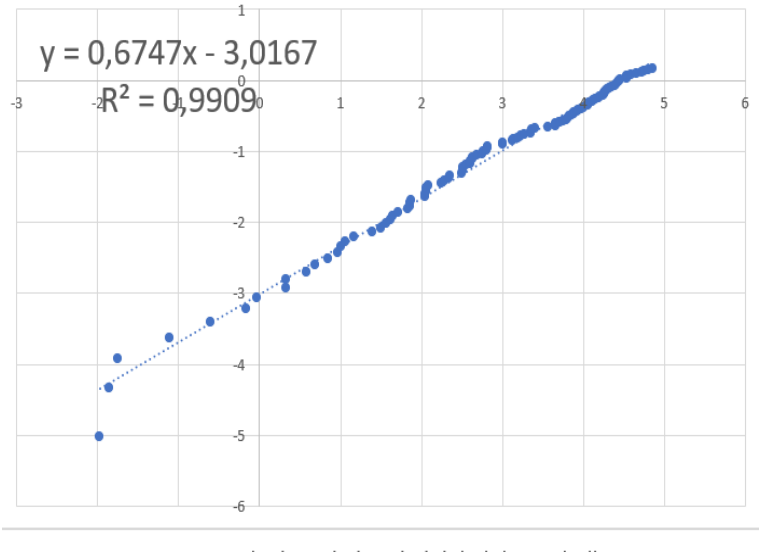


Figura 4.39: Regresión lineal.

En la figura 4.40 se observa el ajuste de los datos a una distribución Weibull. Por otro lado, en la tabla 4.14 se muestran los estadísticos de prueba que se obtienen para cada test de confianza aplicado y en la tabla 4.15 se observan los niveles de confianza en donde la hipótesis nula que indica que los datos siguen una distribución Weibull es rechazada o no.

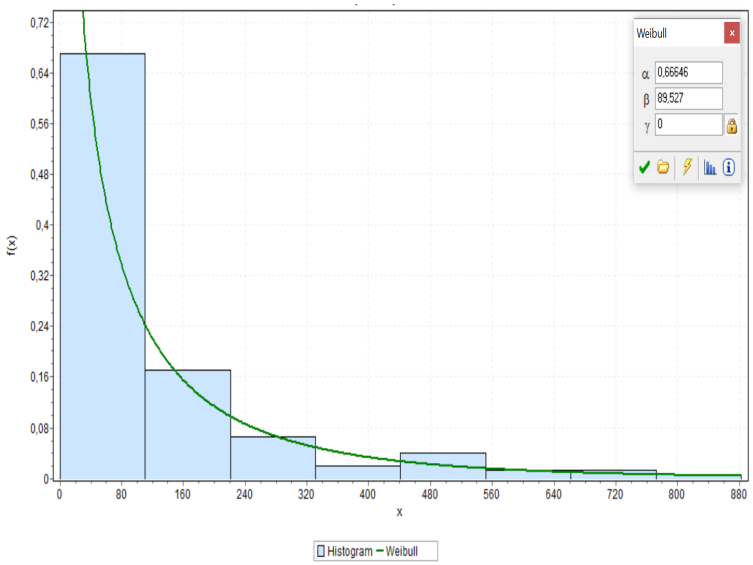


Figura 4.40: Ajuste distribución Weibull de los registros de falla del chancador Symons de sección E.

Tabla 4.14: Tests de confianza para tiempos entre fallas de chancador Symons de sección E.

Test	**Kolmogorov-Smirnov**	**Anderson-Darling**	χ^2
Estadístico	0,050	0,263	7,892

Tabla 4.15: Resultados de tests de bondad de ajuste para tiempos entre fallas de chancador Symons 7' de sección E.

Kolmogorov-Smirnov					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	0,087	0,099	0,110	0,123	0,132
¿Rechazar?	No	No	No	No	No
Anderson-Darling					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	1,375	1,929	2,502	3,289	3,907
¿Rechazar?	No	No	No	No	No
χ^2					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	9,803	12,017	14,067	16,622	18,475
¿Rechazar?	No	No	No	No	No

La tabla 4.16 muestra los parámetros de la distribución Weibull que siguen los tiempos entre falla determinados para el chancador secundario Symons 7' de la sección E de la planta.

Dada la distribución de los TEF se utiliza un ciclo de evaluación de 11 horas.

Tabla 4.16: Parámetros distribución Weibull para tiempos entre fallas de chancador Symons de sección E.

Parámetro	Valor
β	0,666
η	89,527
γ	0

4.5.1. Ciclo de evaluación de 11 horas

Se obtienen 81 registros los cuales se etiquetan con un 0 y corresponden a tiempos en los cuales el equipo no fallará. La figura 4.41 muestra cómo se distribuyen los 0 y 1 en el conjunto de datos.

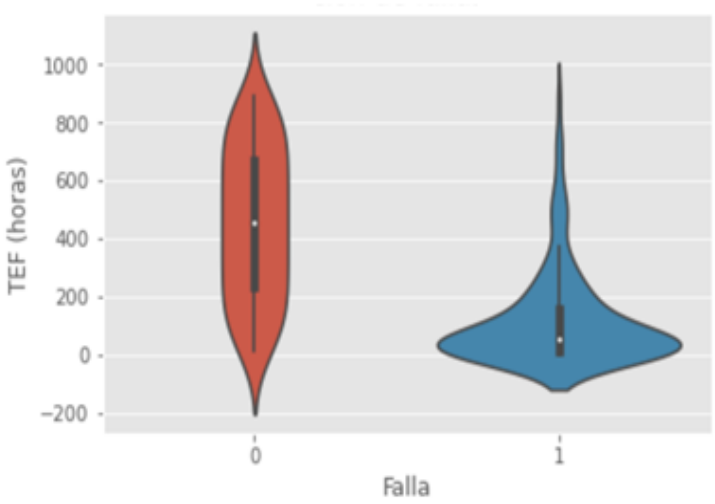


Figura 4.41: Diagrama de violín para chancador secundario Symons sección E.

La figura 4.42 muestra la matriz de confusión resultante al aplicar una regresión logística a los datos considerando un ciclo de evaluación de 11 horas.

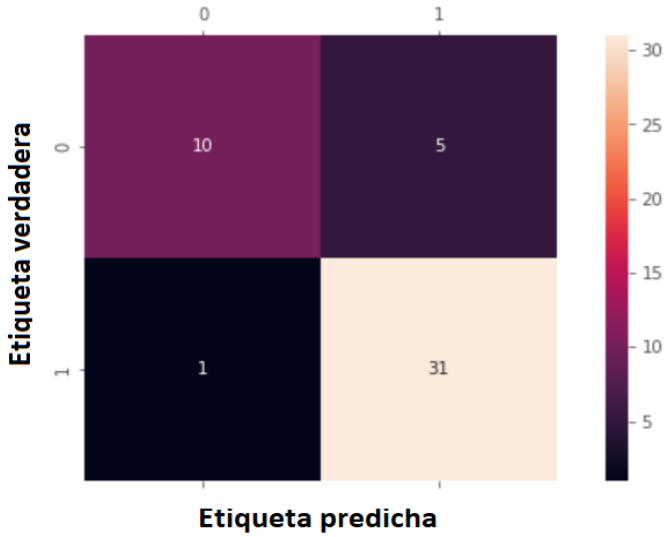


Figura 4.42: Matriz de confusión para chancador secundario Symons sección E.

$$Exactitud = \frac{10+31}{10+31+5+1} = \frac{41}{47} = 0,87 = 87\,\%$$

$$Precisión = \frac{31}{32} = 0,97 = 97\,\%$$

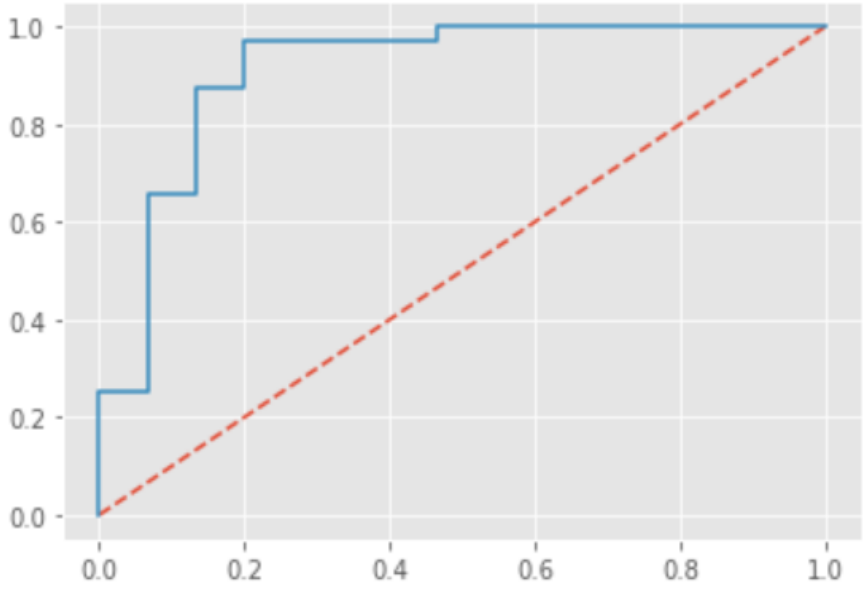


Figura 4.43: Curva ROC para chancador secundario Symons sección E.

$$AUC = 0,92$$

La figura 4.44 muestra la matriz de confusión resultante al aplicar el algoritmo Naïve-Bayes a los datos considerando un ciclo de evaluación de 11 horas.

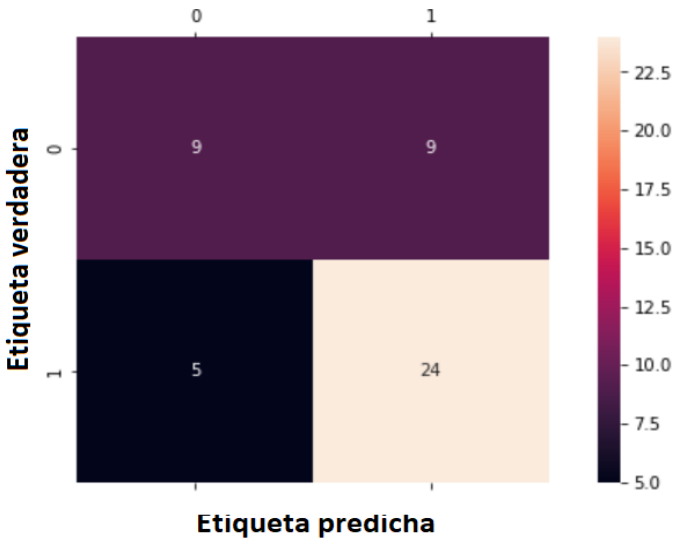


Figura 4.44: Matriz de confusión para chancador secundario Symons sección E.

$$Exactitud = \frac{9+24}{9+24+9+5} = \frac{33}{47} = 0,70 = 70\,\%$$

$$Precisión = \frac{24}{29} = 0,83 = 83\,\%$$

La figura 4.45 muestra la matriz de confusión resultante al aplicar el algoritmo K-NN a los datos considerando un ciclo de evaluación de 11 horas.

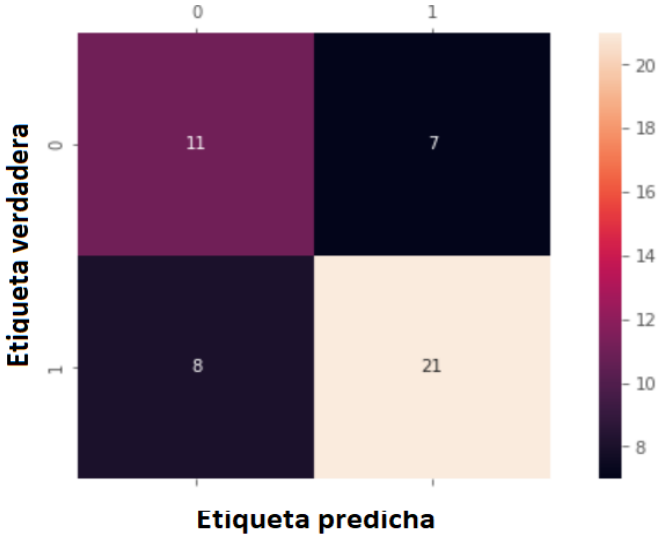


Figura 4.45: Matriz de confusión para chancador secundario Symons sección E.

$$Exactitud = \frac{11+21}{11+21+7+8} = \frac{32}{47} = 0,68 = 68\,\%$$

$$Precisión = \frac{21}{29} = 0,72 = 72\,\%$$

4.6. Chancador terciario 1

Tras filtrar y limpiar los datos se obtienen 140 registros de falla y por ende 139 TEF calculados. La figura 4.46 muestra la regresión lineal que se realiza con los TEF. Se observa que el coeficiente R^2 es de 0,991.

Además, considerando los 139 TEF calculados se tiene que:

$$MTBF = 123,7\ [hrs]$$

y el tiempo medio de reparación corresponde a:

$$MTTR = 3,2\ [hrs]$$

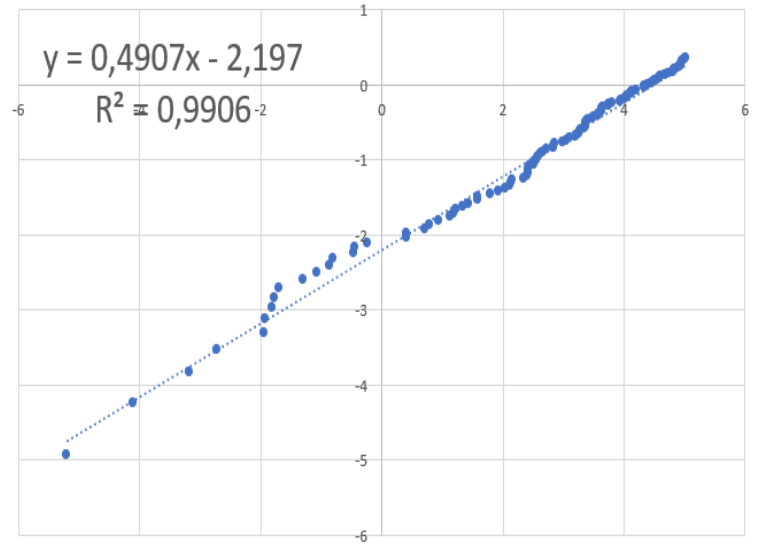


Figura 4.46: Regresión lineal.

En la figura 4.47 se observa el ajuste de los datos a una distribución Weibull. Por otro lado, en la tabla 4.17 se muestran los estadísticos de prueba que se obtienen para cada test de confianza aplicado y en la tabla 4.18 se observan los niveles de confianza en donde la hipótesis nula que indica que los datos siguen una distribución Weibull es rechazada o no.

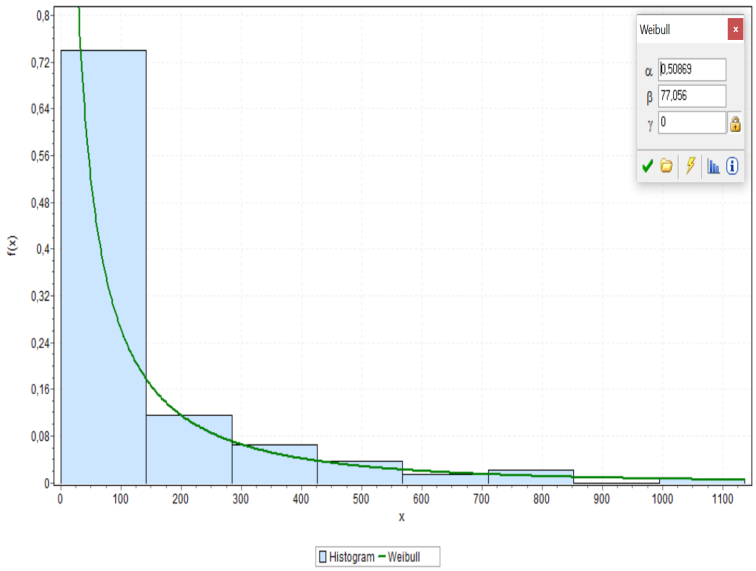


Figura 4.47: Ajuste distribución Weibull de los registros de falla del chancador terciario 1.

Tabla 4.17: Tests de confianza para tiempos entre fallas de chancador terciario 1.

Test	**Kolmogorov-Smirnov**	**Anderson-Darling**	χ^2
Estadístico	0,061	0,385	4,707

Tabla 4.18: Resultados de tests de bondad de ajuste para tiempos entre fallas de chancador terciario 1.

Kolmogorov-Smirnov					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	0,091	0,104	0,115	0,129	0,138
¿Rechazar?	No	No	No	No	No
Anderson-Darling					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	1,375	1,929	2,502	3,289	3,907
¿Rechazar?	No	No	No	No	No
χ^2					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	9,803	12,017	14,067	16,622	18,475
¿Rechazar?	No	No	No	No	No

La tabla 4.19 muestra los parámetros de la distribución Weibull que siguen los tiempos entre falla determinados para el chancador terciario 1.

Dada la distribución de los TEF se utiliza un ciclo de evaluación de 14 horas.

Tabla 4.19: Parámetros distribución Weibull para tiempos entre fallas de chancador terciario 1.

Parámetro	Valor
β	0,509
η	77,056
γ	0

4.6.1. Ciclo de evaluación de 14 horas

Se obtienen 82 registros los cuales se etiquetan con un 0 y corresponden a tiempos en los cuales el equipo no fallará. La figura 4.48 muestra cómo se distribuyen los 0 y 1 en el conjunto de datos.

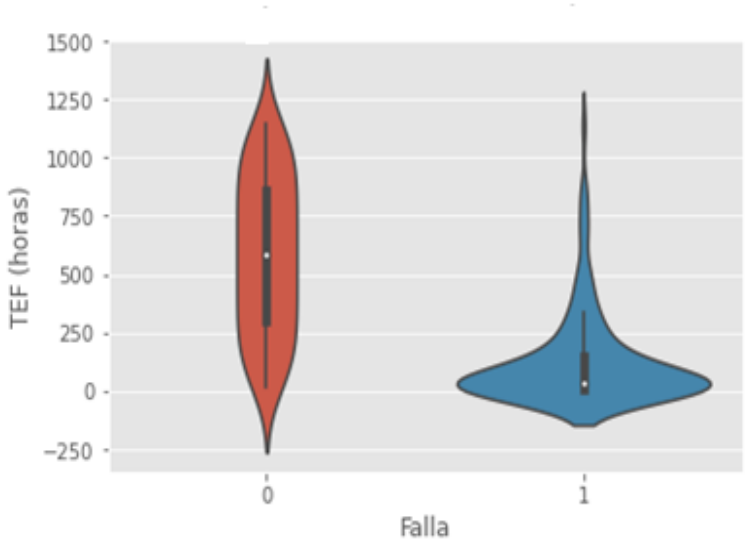


Figura 4.48: Diagrama de violín para chancador terciario 1.

La figura 4.49 muestra la matriz de confusión resultante al aplicar una regresión logística a los datos considerando un ciclo de evaluación de 11 horas.

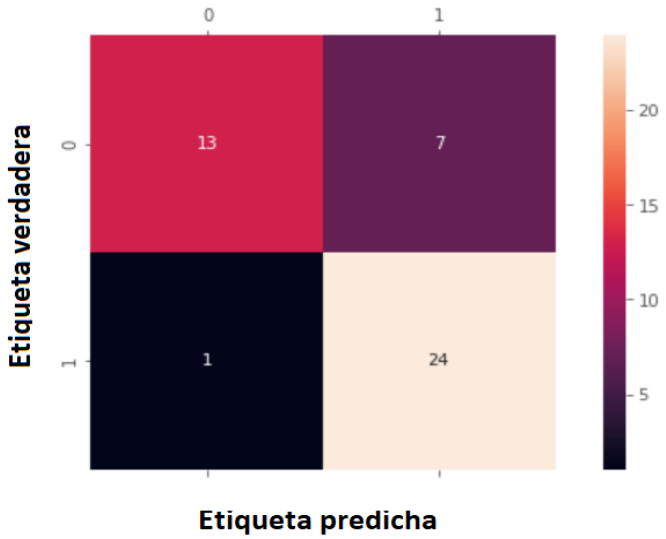


Figura 4.49: Matriz de confusión para chancador terciario 1.

$$Exactitud = \frac{13+24}{13+24+7+1} = \frac{37}{45} = 0,82 = 82\,\%$$

$$Precisión = \frac{24}{25} = 0,96 = 96\,\%$$

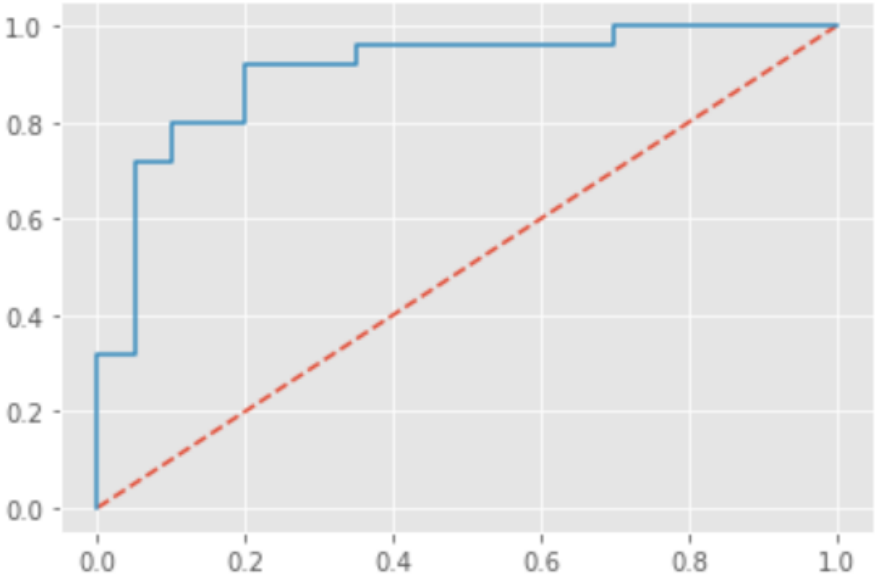


Figura 4.50: Curva ROC para chancador terciario 1.

$$AUC = 0,92$$

La figura 4.51 muestra la matriz de confusión resultante al aplicar el algoritmo Naïve-Bayes

a los datos considerando un ciclo de evaluación de 14 horas.

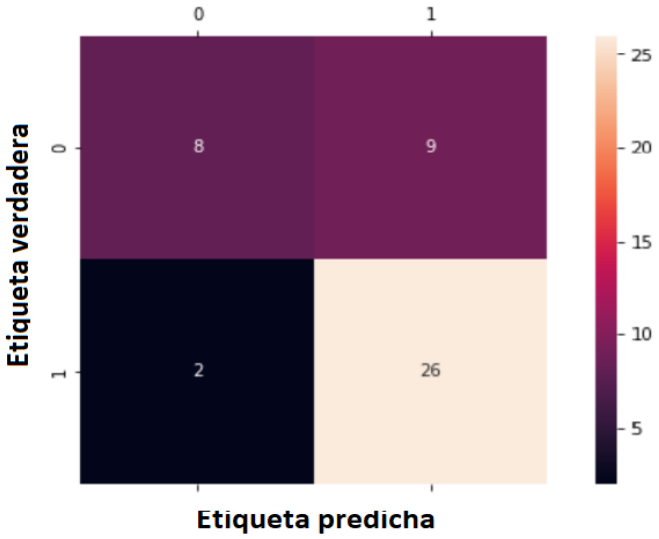


Figura 4.51: Matriz de confusión para chancador terciario 1.

$$Exactitud = \frac{8+26}{8+26+9+2} = \frac{34}{45} = 0,76 = 76\,\%$$

$$Precisión = \frac{26}{28} = 0,93 = 93\,\%$$

La figura 4.52 muestra la matriz de confusión resultante al aplicar el algoritmo K-NN a los datos considerando un ciclo de evaluación de 14 horas.

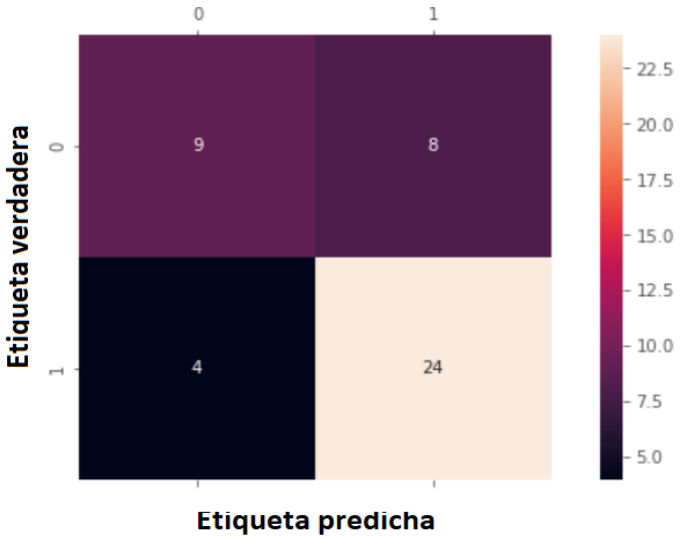


Figura 4.52: Matriz de confusión para chancador terciario 1.

$$Exactitud = \frac{9+24}{9+24+8+4} = \frac{33}{45} = 0,73 = 73\,\%$$

$$Precisión = \frac{24}{28} = 0,86 = 86\,\%$$

4.7. Chancador terciario 2

Tras filtrar y limpiar los datos se obtienen 121 registros de falla y por ende 120 TEF calculados. La figura 4.53 muestra la regresión lineal que se realiza con los TEF. Se observa que el coeficiente R^2 es de 0,976.

Además, considerando los 120 TEF calculados se tiene que:

$$MTBF = 94,5\ [hrs]$$

y el tiempo medio de reparación corresponde a:

$$MTTR = 3,7\ [hrs]$$

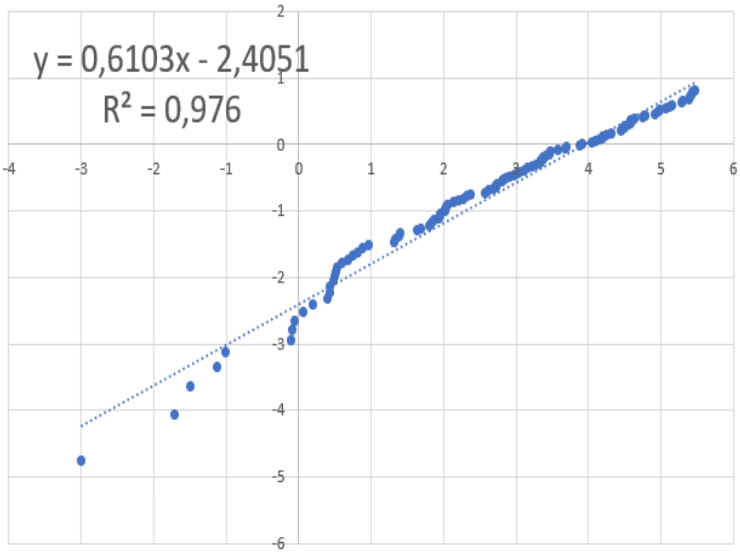


Figura 4.53: Regresión lineal.

En la figura 4.54 se observa el ajuste de los datos a una distribución Weibull. Por otro lado, en la tabla 4.20 se muestran los estadísticos de prueba que se obtienen para cada test de confianza aplicado y en la tabla 4.21 se observan los niveles de confianza en donde la hipótesis nula que indica que los datos siguen una distribución Weibull es rechazada o no.

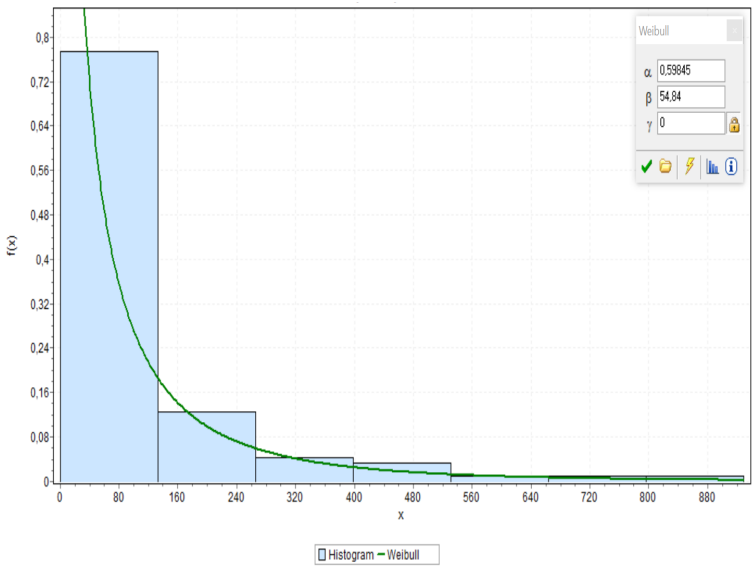


Figura 4.54: Ajuste distribución Weibull de los registros de falla del chancador terciario 2.

Tabla 4.20: Tests de confianza para tiempos entre fallas de chancador terciario 2.

Test	**Kolmogorov-Smirnov**	**Anderson-Darling**	χ^2
Estadístico	0,062	0,831	6,870

Tabla 4.21: Resultados de tests de bondad de ajuste para tiempos entre fallas de chancador terciario 2.

Kolmogorov-Smirnov					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	0,098	0,112	0,124	0,139	0,149
¿Rechazar?	No	No	No	No	No
Anderson-Darling					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	1,375	1,929	2,502	3,289	3,907
¿Rechazar?	No	No	No	No	No
χ^2					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	8,559	10,645	12,592	15,033	16,812
¿Rechazar?	No	No	No	No	No

La tabla 4.22 muestra los parámetros de la distribución Weibull que siguen los tiempos entre falla determinados para el chancador terciario 2.

Dada la distribución de los TEF se utiliza un ciclo de evaluación de 13 horas.

Tabla 4.22: Parámetros distribución Weibull para tiempos entre fallas de chancador terciario 2.

Parámetro	Valor
β	0,598
η	54,840
γ	0

4.7.1. Ciclo de evaluación de 13 horas

Se obtienen 72 registros los cuales se etiquetan con un 0 y corresponden a tiempos en los cuales el equipo no fallará. La figura 4.55 muestra cómo se distribuyen los 0 y 1 en el conjunto de datos.

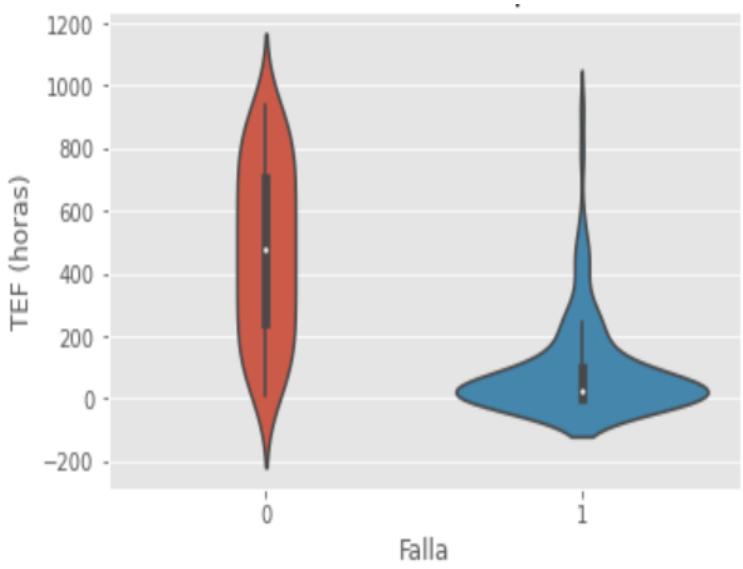


Figura 4.55: Diagrama de violín para chancador terciario 2.

La figura 4.56 muestra la matriz de confusión resultante al aplicar una regresión logística a los datos considerando un ciclo de evaluación de 13 horas.

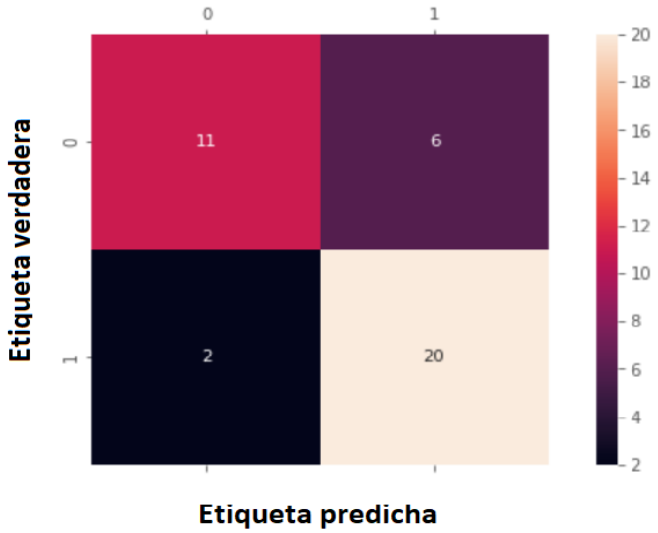


Figura 4.56: Matriz de confusión para chancador terciario 2.

$$Exactitud = \frac{11 + 20}{11 + 20 + 6 + 2} = \frac{31}{39} = 0,79 = 79\,\%$$

$$Precisión = \frac{20}{22} = 0,91 = 91\,\%$$

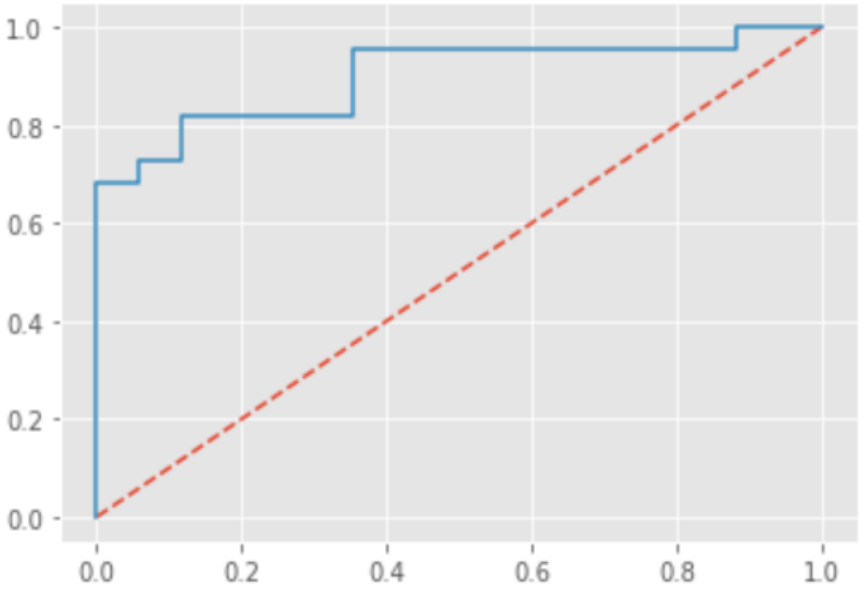


Figura 4.57: Curva ROC para chancador terciario 2.

$$AUC = 0,90$$

La figura 4.58 muestra la matriz de confusión resultante al aplicar el algoritmo Naïve-Bayes

a los datos considerando un ciclo de evaluación de 13 horas.

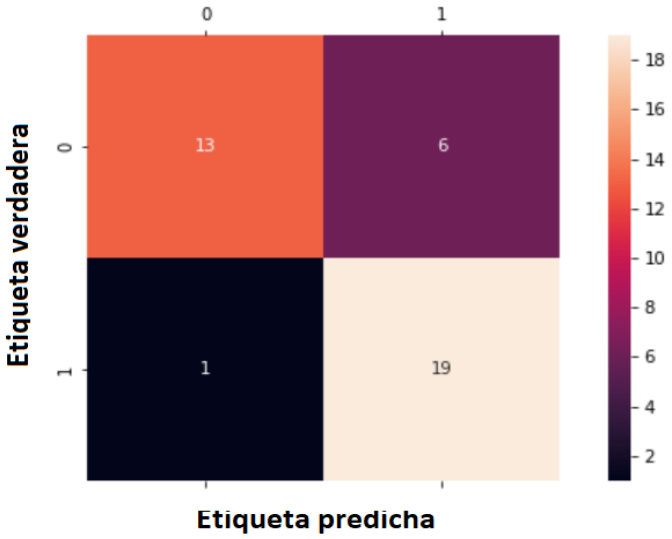


Figura 4.58: Matriz de confusión para chancador terciario 2.

$$Exactitud = \frac{13+19}{13+19+6+1} = \frac{32}{39} = 0,82 = 82\,\%$$

$$Precisión = \frac{19}{20} = 0,95 = 95\,\%$$

La figura 4.59 muestra la matriz de confusión resultante al aplicar el algoritmo K-NN a los datos considerando un ciclo de evaluación de 13 horas.

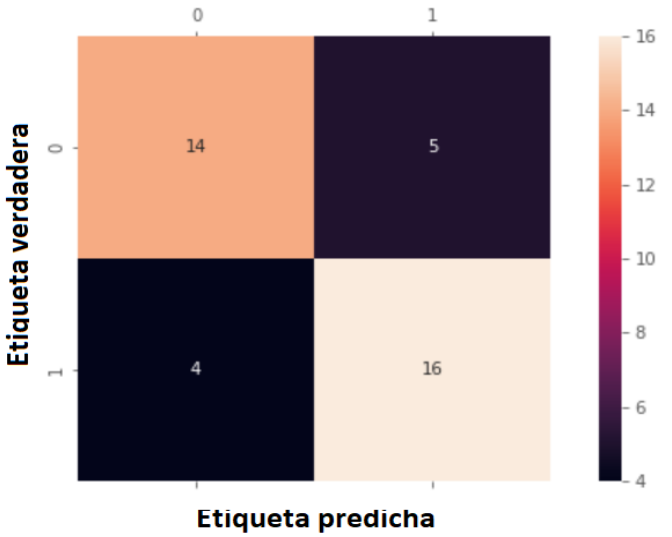


Figura 4.59: Matriz de confusión para chancador terciario 2.

$$Exactitud = \frac{14+16}{14+16+5+4} = \frac{30}{39} = 0,77 = 77\,\%$$

$$Precisión = \frac{16}{20} = 0,80 = 80\,\%$$

4.8. Chancador terciario 3

Tras filtrar y limpiar los datos se obtienen 126 registros de falla y por ende 125 TEF calculados. La figura 4.60 muestra la regresión lineal que se realiza con los TEF. Se observa que el coeficiente R^2 es de 0,970.

Además, considerando los 125 TEF calculados se tiene que:

$$MTBF = 76,5\ [hrs]$$

y el tiempo medio de reparación corresponde a:

$$MTTR = 1,4\ [hrs]$$

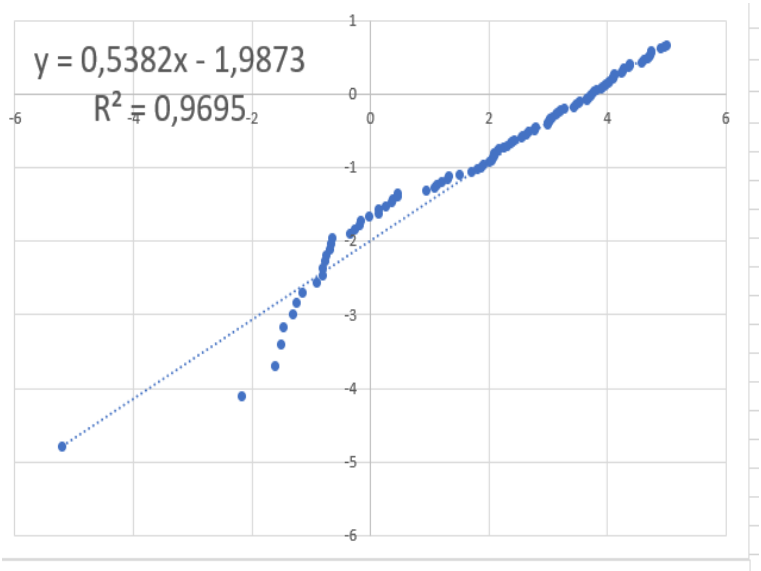


Figura 4.60: Regresión lineal.

En la figura 4.61 se observa el ajuste de los datos a una distribución Weibull. Por otro lado, en la tabla 4.23 se muestran los estadísticos de prueba que se obtienen para cada test de confianza aplicado y en la tabla 4.24 se observan los niveles de confianza en donde la hipótesis nula que indica que los datos siguen una distribución Weibull es rechazada o no.

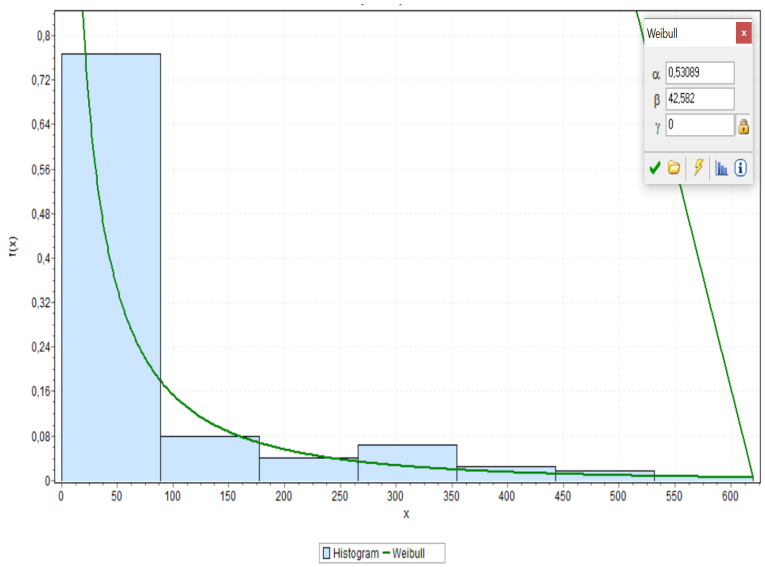


Figura 4.61: Ajuste distribución Weibull de los registros de falla del chancador terciario 3.

Tabla 4.23: Tests de confianza para tiempos entre fallas de chancador terciario 3.

Test	**Kolmogorov-Smirnov**	**Anderson-Darling**	χ^2
Estadístico	0,062	0,532	4,528

Tabla 4.24: Resultados de tests de bondad de ajuste para tiempos entre fallas de chancador terciario 3.

Kolmogorov-Smirnov					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	0,096	0,109	0,121	0,136	0,146
¿Rechazar?	No	No	No	No	No
Anderson-Darling					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	1,375	1,929	2,502	3,289	3,907
¿Rechazar?	No	No	No	No	No
χ^2					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	8,558	10,645	12,592	15,033	16,812
¿Rechazar?	No	No	No	No	No

La tabla 4.25 muestra los parámetros de la distribución Weibull que siguen los tiempos entre falla determinados para el chancador terciario 3.

Dada la distribución de los TEF se utiliza un ciclo de evaluación de 10 horas.

Tabla 4.25: Parámetros distribución Weibull para tiempos entre fallas de chancador terciario 3.

Parámetro	Valor
β	0,531
η	42,582
γ	0

4.8.1. Ciclo de evaluación de 10 horas

Se obtienen 62 registros los cuales se etiquetan con un 0 y corresponden a tiempos en los cuales el equipo no fallará. La figura 4.62 muestra cómo se distribuyen los 0 y 1 en el conjunto de datos.

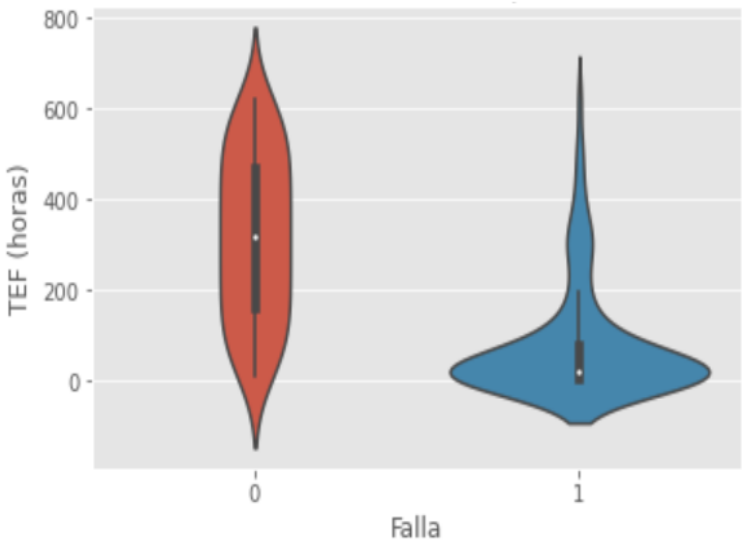


Figura 4.62: Diagrama de violín para chancador terciario 3.

La figura 4.63 muestra la matriz de confusión resultante al aplicar una regresión logística a los datos considerando un ciclo de evaluación de 10 horas.

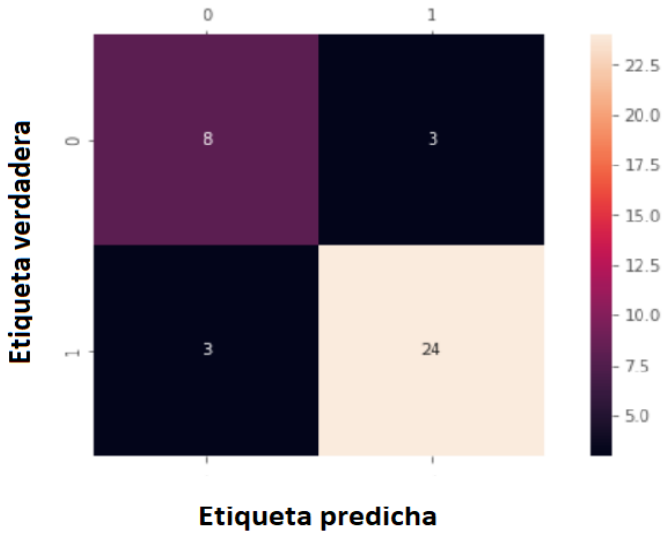


Figura 4.63: Matriz de confusión para chancador terciario 3.

$$Exactitud = \frac{8+24}{8+24+3+3} = \frac{32}{38} = 0,84 = 84\,\%$$

$$Precisión = \frac{24}{27} = 0,89 = 89\,\%$$

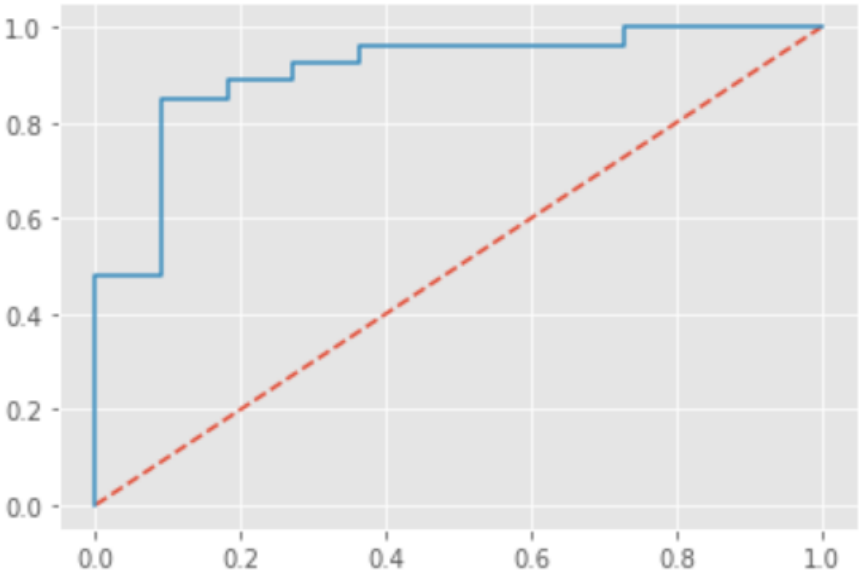


Figura 4.64: Curva ROC para chancador terciario 3.

$$AUC = 0,91$$

La figura 4.65 muestra la matriz de confusión resultante al aplicar el algoritmo Naïve-Bayes

a los datos considerando un ciclo de evaluación de 10 horas.

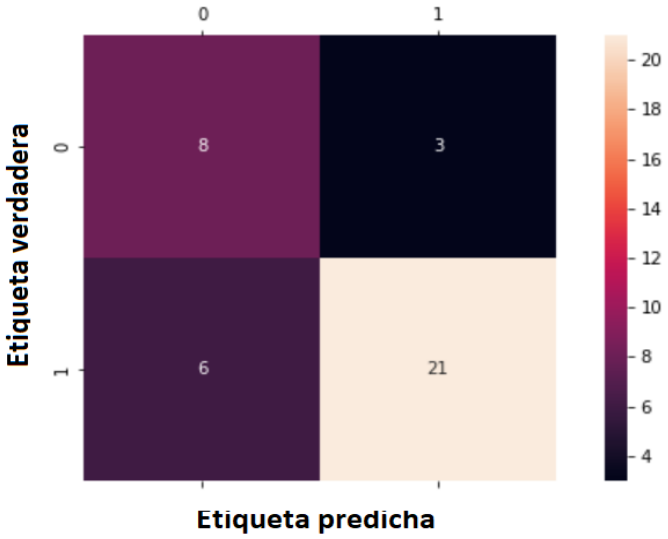


Figura 4.65: Matriz de confusión para chancador terciario 3.

$$Exactitud = \frac{8+21}{8+21+3+6} = \frac{29}{38} = 0,76 = 76\,\%$$

$$Precisión = \frac{21}{27} = 0,78 = 78\,\%$$

La figura 4.66 muestra la matriz de confusión resultante al aplicar el algoritmo K-NN a los datos considerando un ciclo de evaluación de 10 horas.

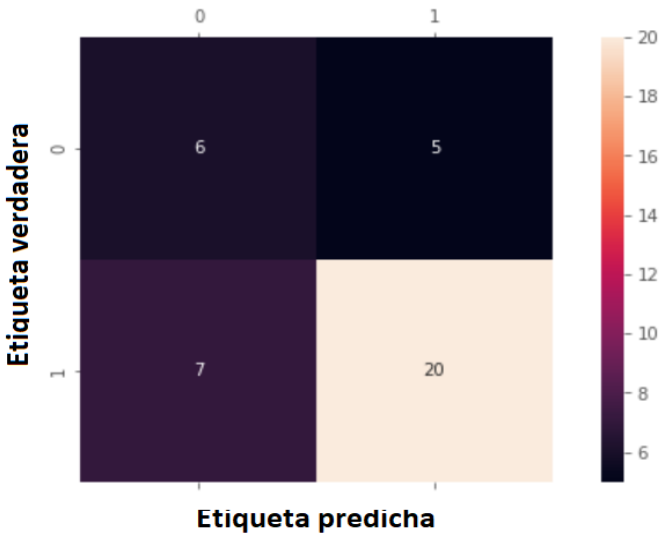


Figura 4.66: Matriz de confusión para chancador terciario 3.

$$Exactitud = \frac{6+20}{6+20+5+7} = \frac{26}{38} = 0,68 = 68\,\%$$

$$Precisión = \frac{20}{26} = 0,77 = 77\,\%$$

4.9. Chancador terciario 4

Tras filtrar y limpiar los datos se obtienen 187 registros de falla y por ende 186 TEF calculados. La figura 4.67 muestra la regresión lineal que se realiza con los TEF. Se observa que el coeficiente R^2 es de 0,990.

Además, considerando los 186 TEF calculados se tiene que:

$$MTBF = 105,0\ [hrs]$$

y el tiempo medio de reparación corresponde a:

$$MTTR = 3,1\ [hrs]$$

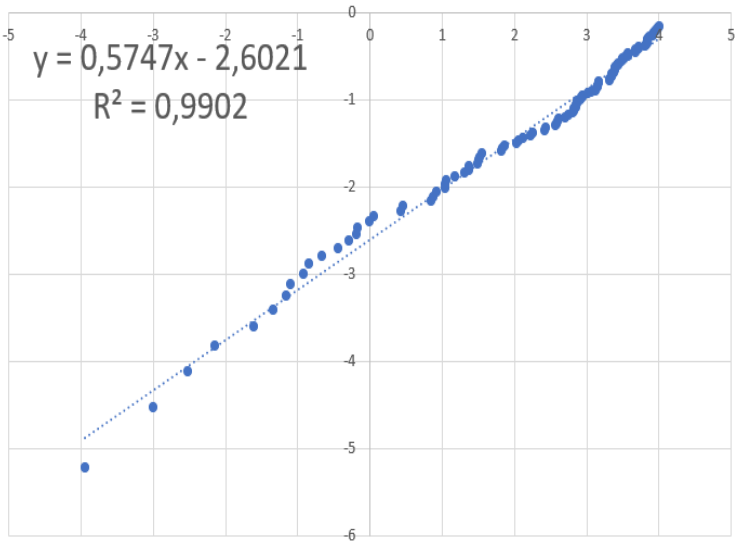


Figura 4.67: Regresión lineal.

En la figura 4.68 se observa el ajuste de los datos a una distribución Weibull. Por otro lado, en la tabla 4.26 se muestran los estadísticos de prueba que se obtienen para cada test de confianza aplicado y en la tabla 4.27 se observan los niveles de confianza en donde la hipótesis nula que indica que los datos siguen una distribución Weibull es rechazada o no.

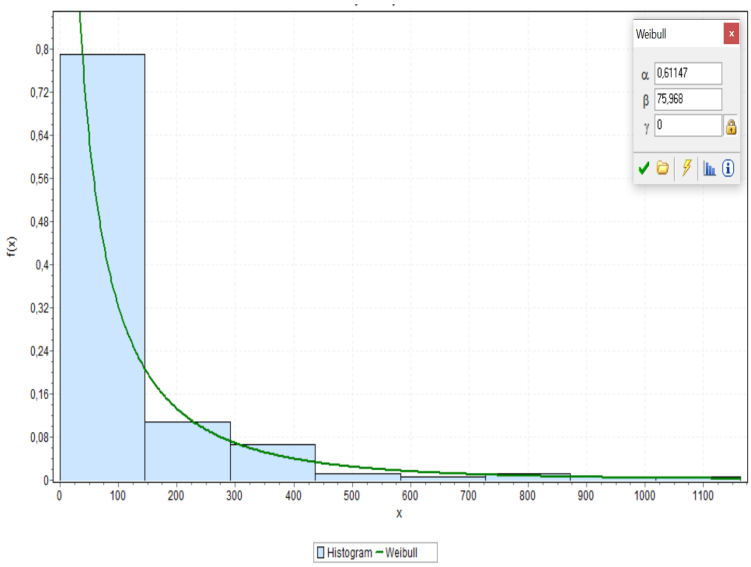


Figura 4.68: Ajuste distribución Weibull de los registros de falla del chancador terciario 4.

Tabla 4.26: Tests de confianza para tiempos entre fallas de chancador terciario 4.

Test	**Kolmogorov-Smirnov**	**Anderson-Darling**	χ^2
Estadístico	0,064	0,561	8,242

Tabla 4.27: Resultados de tests de bondad de ajuste para tiempos entre fallas de chancador terciario 4.

Kolmogorov-Smirnov					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	0,079	0,090	0,100	0,111	0,119
¿Rechazar?	No	No	No	No	No
Anderson-Darling					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	1,375	1,929	2,502	3,289	3,907
¿Rechazar?	No	No	No	No	No
χ^2					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	9,803	12,017	14,067	16,622	18,475
¿Rechazar?	No	No	No	No	No

La tabla 4.28 muestra los parámetros de la distribución Weibull que siguen los tiempos entre falla determinados para el chancador terciario 4.

Dada la distribución de los TEF se utiliza un ciclo de evaluación de 12 horas.

Tabla 4.28: Parámetros distribución Weibull para tiempos entre fallas de chancador terciario 4.

Parámetro	Valor
β	0,611
η	75,968
γ	0

4.9.1. Ciclo de evaluación de 12 horas

Se obtienen 97 registros los cuales se etiquetan con un 0 y corresponden a tiempos en los cuales el equipo no fallará. La figura 4.69 muestra cómo se distribuyen los 0 y 1 en el conjunto de datos.

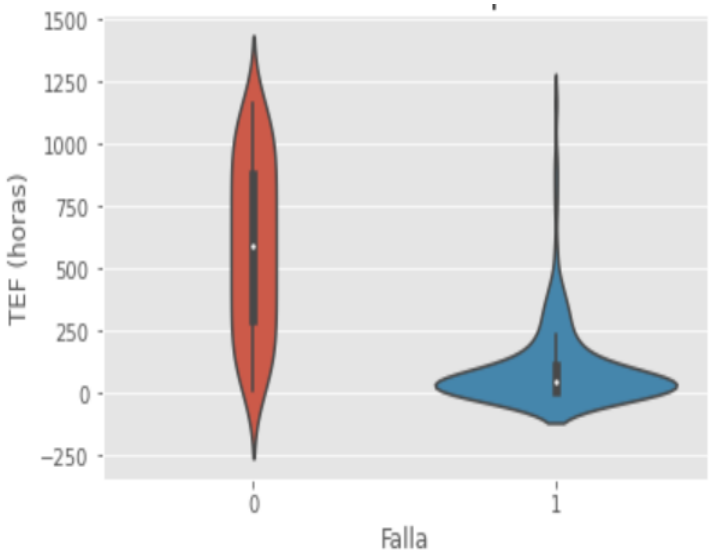


Figura 4.69: Diagrama de violín para chancador terciario 4.

La figura 4.70 muestra la matriz de confusión resultante al aplicar una regresión logística a los datos considerando un ciclo de evaluación de 12 horas.

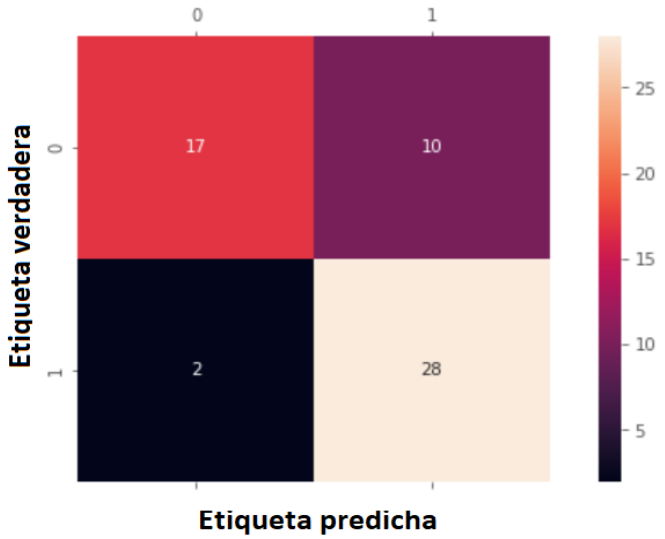


Figura 4.70: Matriz de confusión para chancador terciario 4.

$$Exactitud = \frac{17+28}{17+28+10+2} = \frac{45}{57} = 0,79 = 79\,\%$$

$$Precisión = \frac{28}{30} = 0,93 = 93\,\%$$

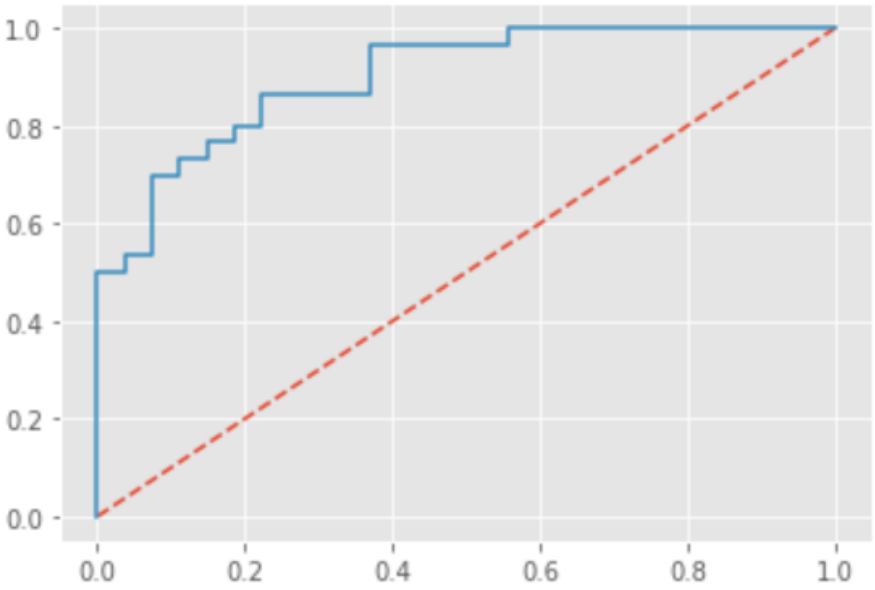


Figura 4.71: Curva ROC para chancador terciario 4.

$$AUC = 0,97$$

La figura 4.72 muestra la matriz de confusión resultante al aplicar el algoritmo Naïve-Bayes

a los datos considerando un ciclo de evaluación de 12 horas.

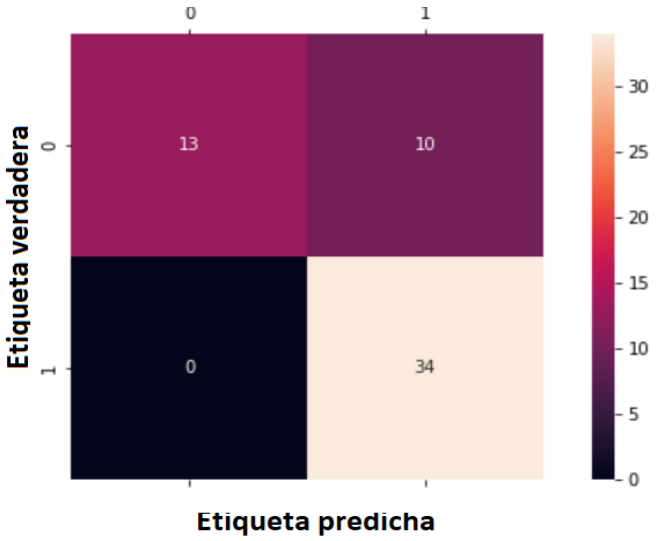


Figura 4.72: Matriz de confusión para chancador terciario 4.

$$Exactitud = \frac{13 + 34}{13 + 34 + 10 + 0} = \frac{47}{57} = 0,82 = 82\,\%$$

$$Precisión = \frac{34}{34} = 1,00 = 100\,\%$$

La figura 4.73 muestra la matriz de confusión resultante al aplicar el algoritmo K-NN a los datos considerando un ciclo de evaluación de 12 horas.

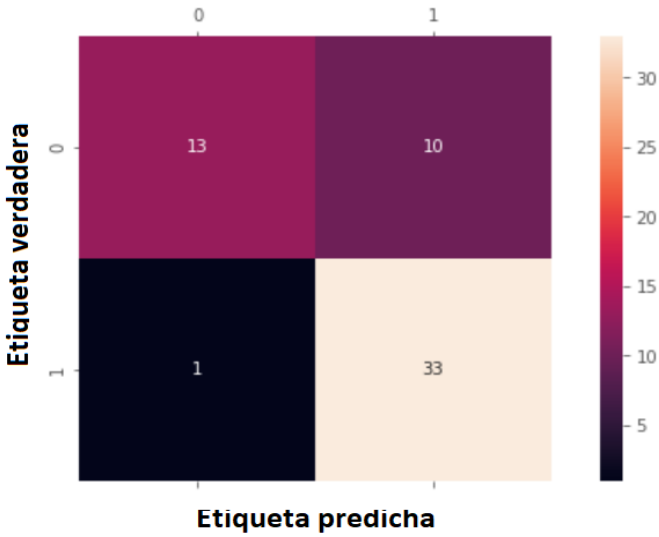


Figura 4.73: Matriz de confusión para chancador terciario 4.

$$Exactitud = \frac{13+33}{13+33+10+1} = \frac{46}{57} = 0,81 = 81\,\%$$

$$Precisión = \frac{33}{34} = 0,97 = 97\,\%$$

4.10. Chancador terciario 5

Tras filtrar y limpiar los datos se obtienen 144 registros de falla y por ende 143 TEF calculados. La figura 4.74 muestra la regresión lineal que se realiza con los TEF. Se observa que el coeficiente R^2 es de 0,989.

Además, considerando los 143 TEF calculados se tiene que:

$$MTBF = 118,5\ [hrs]$$

y el tiempo medio de reparación corresponde a:

$$MTTR = 6,6\ [hrs]$$

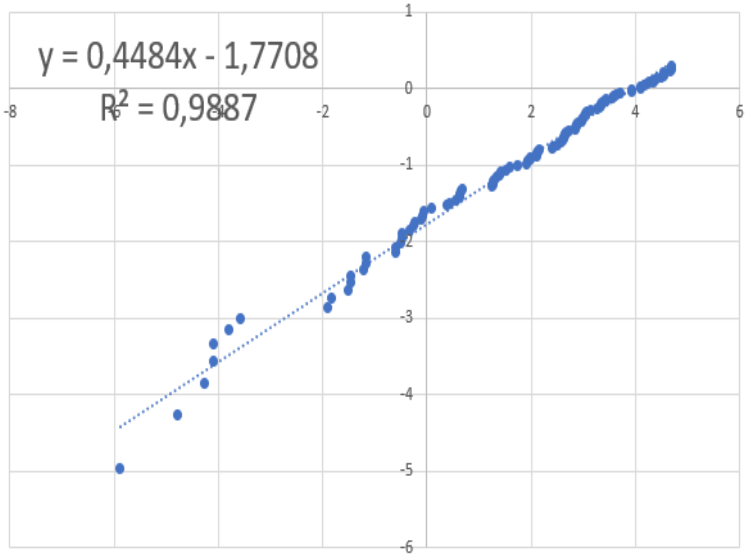


Figura 4.74: Regresión lineal.

En la figura 4.75 se observa el ajuste de los datos a una distribución Weibull. Por otro lado, en la tabla 4.29 se muestran los estadísticos de prueba que se obtienen para cada test de confianza aplicado y en la tabla 4.30 se observan los niveles de confianza en donde la hipótesis nula que indica que los datos siguen una distribución Weibull es rechazada o no.

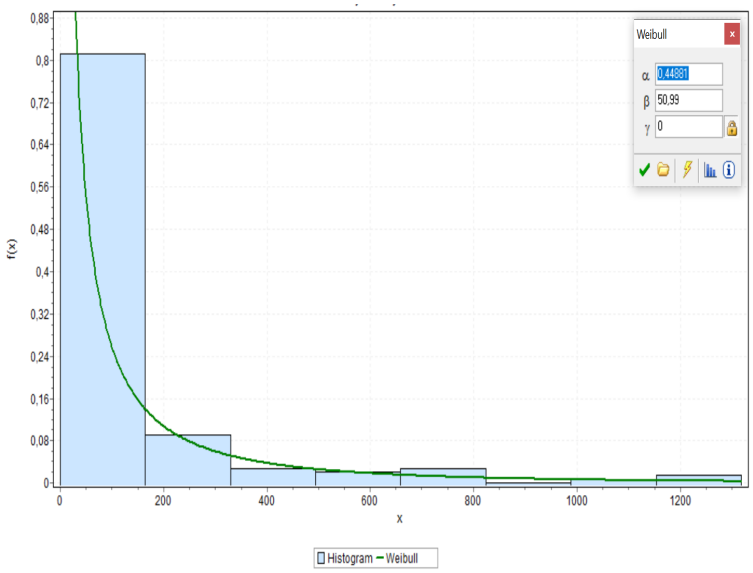


Figura 4.75: Ajuste distribución Weibull de los registros de falla del chancador terciario 5.

Tabla 4.29: Tests de confianza para tiempos entre fallas de chancador terciario 5.

Test	**Kolmogorov-Smirnov**	**Anderson-Darling**	χ^2
Estadístico	0,039	0,254	4,832

Tabla 4.30: Resultados de tests de bondad de ajuste para tiempos entre fallas de chancador terciario 5.

Kolmogorov-Smirnov					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	0,090	0,102	0,114	0,127	0,136
¿Rechazar?	No	No	No	No	No
Anderson-Darling					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	1,375	1,929	2,502	3,289	3,907
¿Rechazar?	No	No	No	No	No
χ^2					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	9,803	12,017	14,067	16,622	18,475
¿Rechazar?	No	No	No	No	No

La tabla 4.31 muestra los parámetros de la distribución Weibull que siguen los tiempos entre falla determinados para el chancador terciario 5.

Dada la distribución de los TEF se utiliza un ciclo de evaluación de 13 horas.

Tabla 4.31: Parámetros distribución Weibull para tiempos entre fallas de chancador terciario 5.

Parámetro	Valor
β	0,449
η	50,990
γ	0

4.10.1. Ciclo de evaluación de 13 horas

Se obtienen 102 registros los cuales se etiquetan con un 0 y corresponden a tiempos en los cuales el equipo no fallará. La figura 4.76 muestra cómo se distribuyen los 0 y 1 en el conjunto de datos.

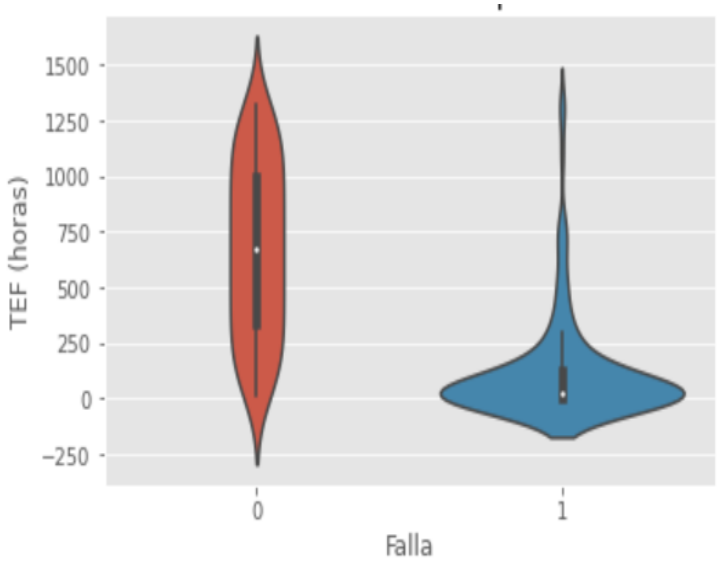


Figura 4.76: Diagrama de violín para chancador terciario 5.

La figura 4.77 muestra la matriz de confusión resultante al aplicar una regresión logística a los datos considerando un ciclo de evaluación de 13 horas.

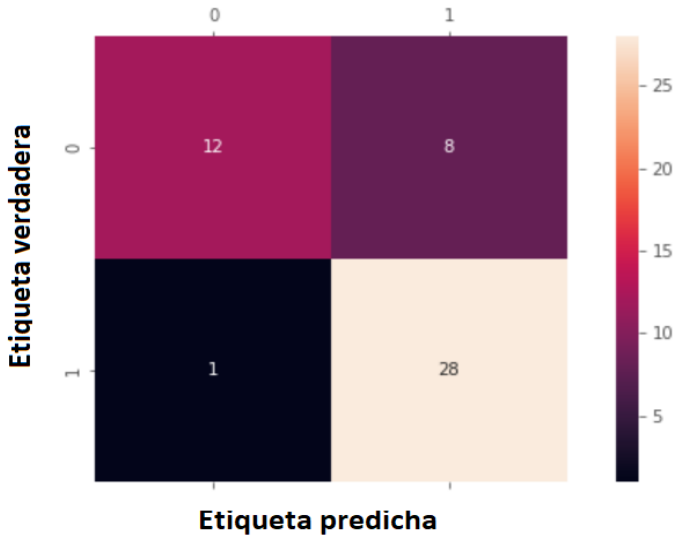


Figura 4.77: Matriz de confusión para chancador terciario 5.

$$Exactitud = \frac{12+28}{12+28+8+1} = \frac{40}{49} = 0,82 = 82\,\%$$

$$Precisión = \frac{28}{29} = 0,97 = 97\,\%$$

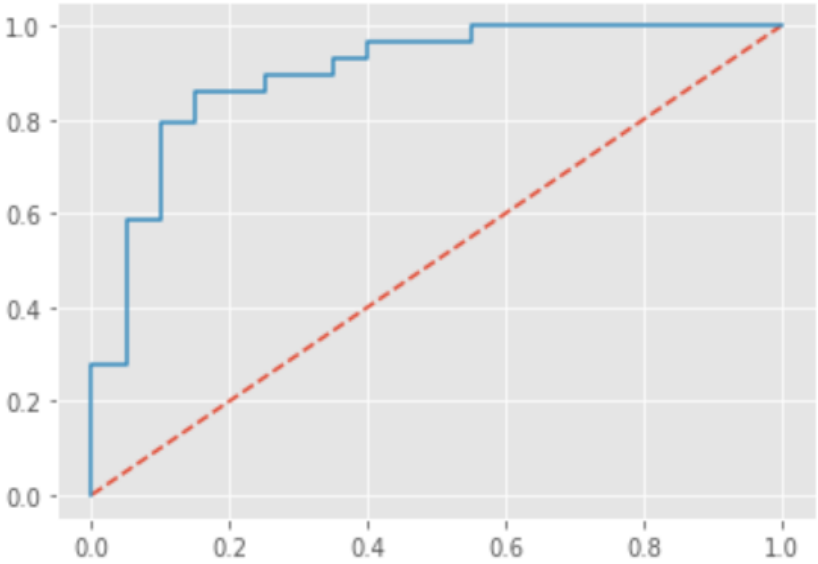


Figura 4.78: Curva ROC para chancador terciario 5.

$$AUC = 0,92$$

La figura 4.79 muestra la matriz de confusión resultante al aplicar el algoritmo Naïve-Bayes

a los datos considerando un ciclo de evaluación de 13 horas.

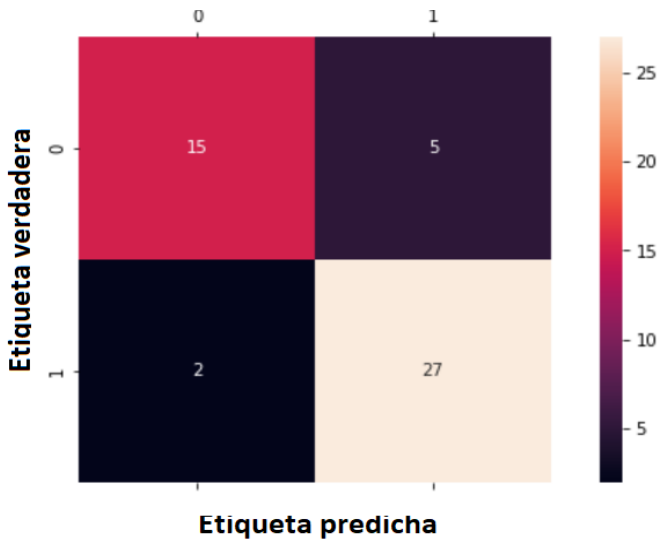


Figura 4.79: Matriz de confusión para chancador terciario 5.

$$Exactitud = \frac{15 + 27}{15 + 27 + 5 + 2} = \frac{42}{49} = 0,86 = 86\,\%$$

$$Precisión = \frac{27}{29} = 0,93 = 93\,\%$$

La figura 4.80 muestra la matriz de confusión resultante al aplicar el algoritmo K-NN a los datos considerando un ciclo de evaluación de 13 horas.

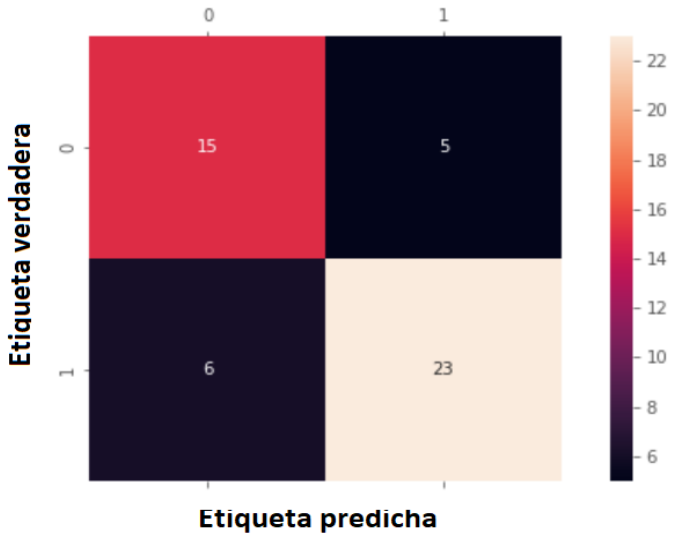


Figura 4.80: Matriz de confusión para chancador terciario 5.

$$Exactitud = \frac{15+23}{15+23+5+6} = \frac{38}{49} = 0,78 = 78\,\%$$

$$Precisión = \frac{23}{29} = 0,79 = 79\,\%$$

4.11. Chancador terciario 6

Tras filtrar y limpiar los datos se obtienen 229 registros de falla y por ende 228 TEF calculados. La figura 4.81 muestra la regresión lineal que se realiza con los TEF. Se observa que el coeficiente R^2 es de 0,982.

Además, considerando los 228 TEF calculados se tiene que:

$$MTBF = 75,8\ [hrs]$$

y el tiempo medio de reparación corresponde a:

$$MTTR = 3,9\ [hrs]$$

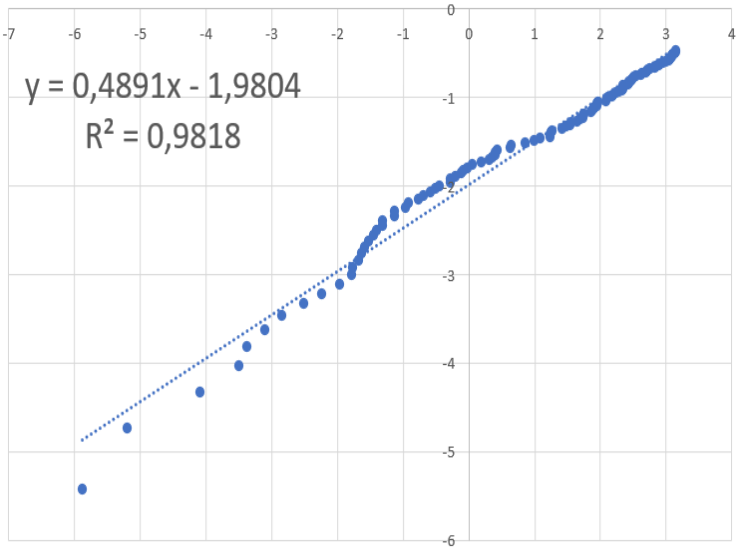


Figura 4.81: Regresión lineal.

En la figura 4.82 se observa el ajuste de los datos a una distribución Weibull. Por otro lado, en la tabla 4.32 se muestran los estadísticos de prueba que se obtienen para cada test de confianza aplicado y en la tabla 4.33 se observan los niveles de confianza en donde la hipótesis nula que indica que los datos siguen una distribución Weibull es rechazada o no.

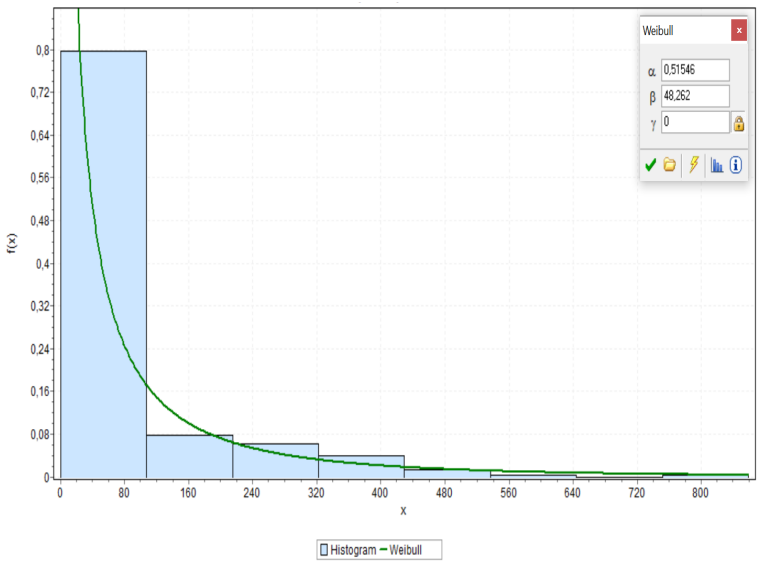


Figura 4.82: Ajuste distribución Weibull de los registros de falla del chancador terciario 6.

Tabla 4.32: Tests de confianza para tiempos entre fallas de chancador terciario 6.

Test	**Kolmogorov-Smirnov**	**Anderson-Darling**	χ^2
Estadístico	0,055	0,873	9,597

Tabla 4.33: Resultados de tests de bondad de ajuste para tiempos entre fallas de chancador terciario 6.

Kolmogorov-Smirnov					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	0,071	0,081	0,090	0,101	0,108
¿Rechazar?	No	No	No	No	No
Anderson-Darling					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	1,375	1,929	2,502	3,289	3,907
¿Rechazar?	No	No	No	No	No
χ^2					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	9,803	12,017	14,067	16,622	18,475
¿Rechazar?	No	No	No	No	No

La tabla 4.34 muestra los parámetros de la distribución Weibull que siguen los tiempos entre falla determinados para el chancador terciario 6.

Dada la distribución de los TEF se utiliza un ciclo de evaluación de 12 horas.

Tabla 4.34: Parámetros distribución Weibull para tiempos entre fallas de chancador terciario 6.

Parámetro	Valor
β	0,515
η	48,262
γ	0

4.11.1. Ciclo de evaluación de 12 horas

Se obtienen 102 registros los cuales se etiquetan con un 0 y corresponden a tiempos en los cuales el equipo no fallará. La figura 4.83 muestra cómo se distribuyen los 0 y 1 en el conjunto de datos.

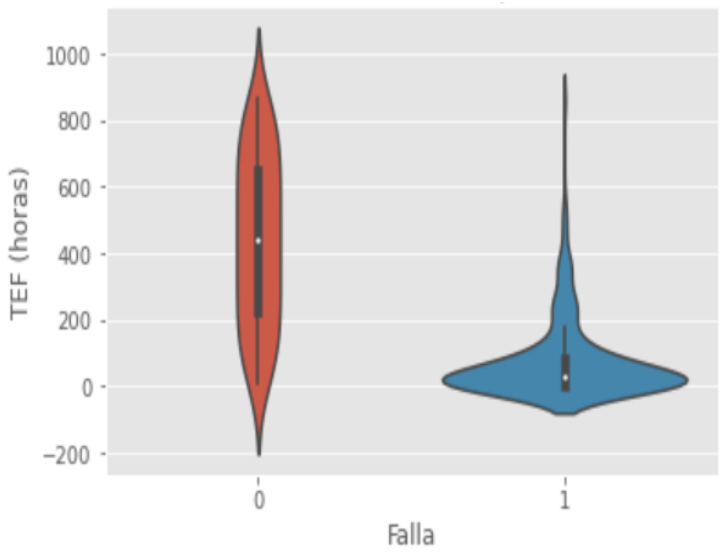


Figura 4.83: Diagrama de violín para chancador terciario 6.

La figura 4.84 muestra la matriz de confusión resultante al aplicar una regresión logística a los datos considerando un ciclo de evaluación de 12 horas.

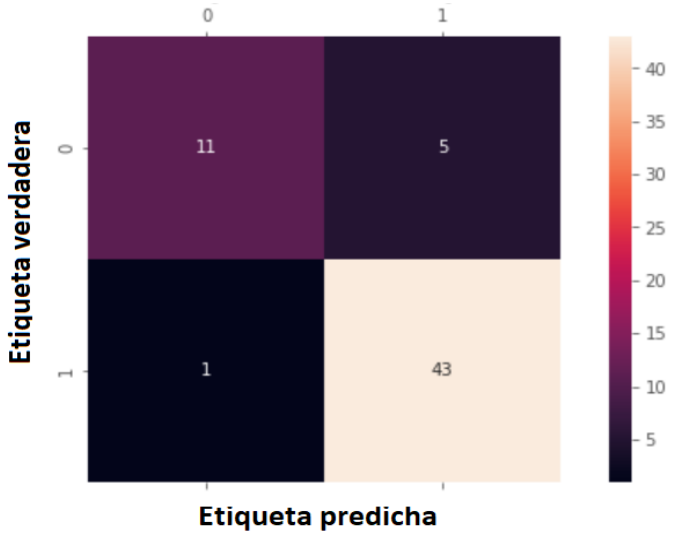


Figura 4.84: Matriz de confusión para chancador terciario 6.

$$Exactitud = \frac{11+43}{11+43+5+1} = \frac{54}{60} = 0,90 = 90\,\%$$

$$Precisión = \frac{43}{44} = 0,98 = 98\,\%$$

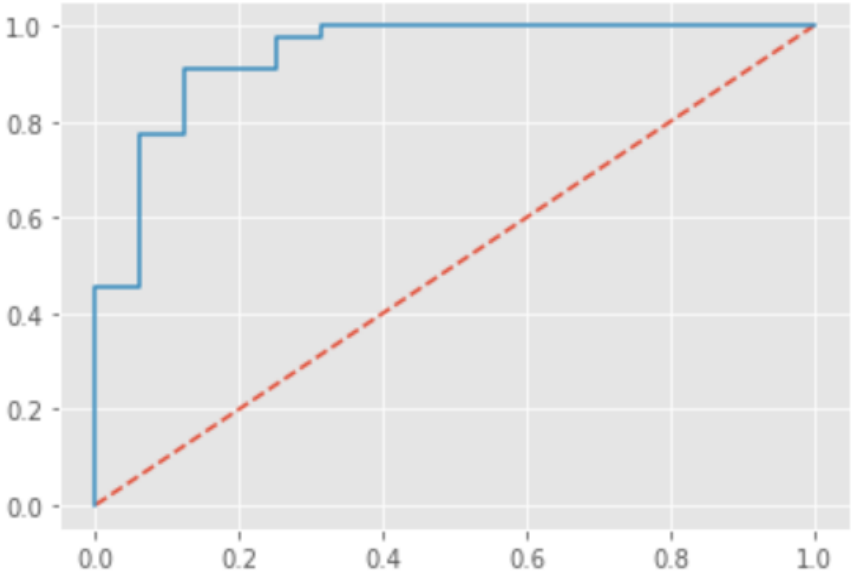


Figura 4.85: Curva ROC para chancador terciario 6.

$$AUC = 0,94$$

La figura 4.86 muestra la matriz de confusión resultante al aplicar el algoritmo Naïve-Bayes

a los datos considerando un ciclo de evaluación de 12 horas.

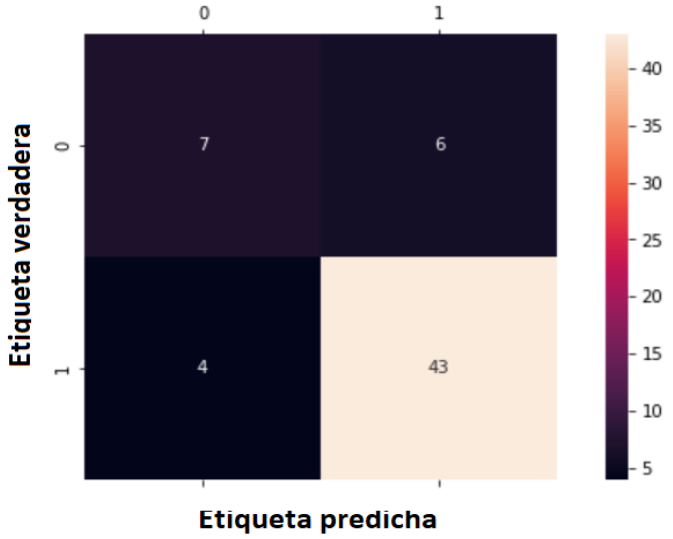


Figura 4.86: Matriz de confusión para chancador terciario 6.

$$Exactitud = \frac{7+43}{7+43+6+4} = \frac{50}{60} = 0,83 = 83\,\%$$

$$Precisión = \frac{43}{47} = 0,91 = 91\,\%$$

La figura 4.87 muestra la matriz de confusión resultante al aplicar el algoritmo K-NN a los datos considerando un ciclo de evaluación de 12 horas.

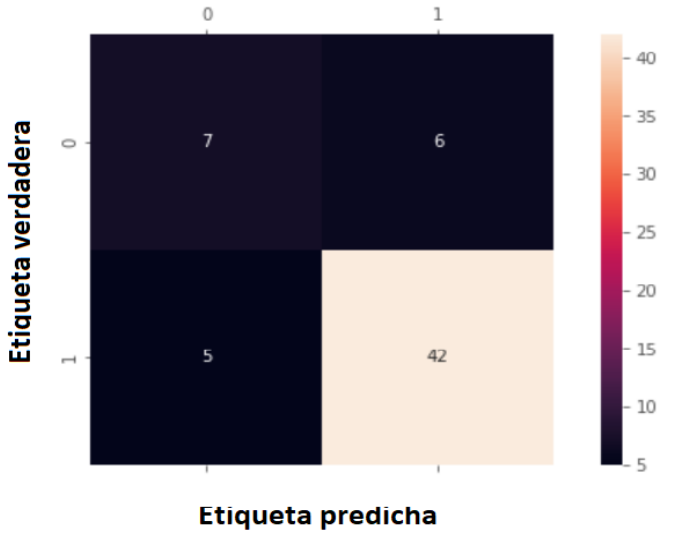


Figura 4.87: Matriz de confusión para chancador terciario 6.

$$Exactitud = \frac{7+42}{7+42+6+5} = \frac{49}{60} = 0,82 = 82\,\%$$

$$Precisión = \frac{42}{47} = 0,89 = 89\,\%$$

4.12. Chancador terciario 7

Tras filtrar y limpiar los datos se obtienen 201 registros de falla y por ende 200 TEF calculados. La figura 4.88 muestra la regresión lineal que se realiza con los TEF. Se observa que el coeficiente R^2 es de 0,978.

Además, considerando los 200 TEF calculados se tiene que:

$$MTBF = 94,3\ [hrs]$$

y el tiempo medio de reparación corresponde a:

$$MTTR = 3,1\ [hrs]$$

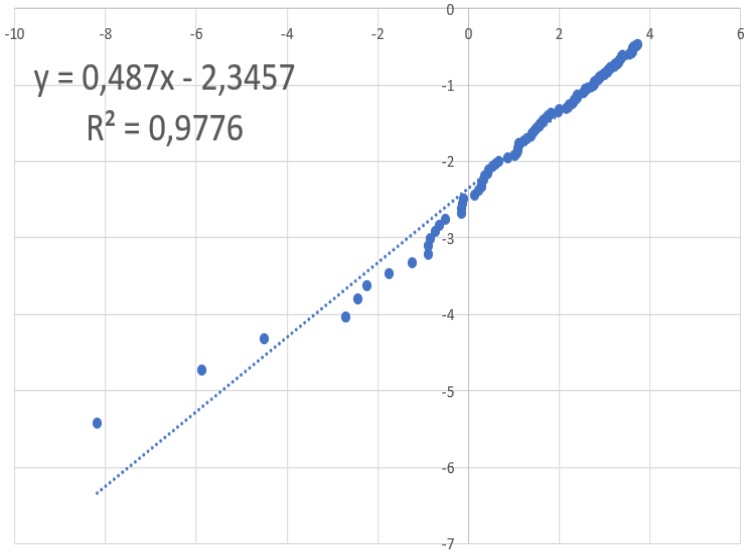


Figura 4.88: Regresión lineal.

En la figura 4.89 se observa el ajuste de los datos a una distribución Weibull. Por otro lado, en la tabla 4.35 se muestran los estadísticos de prueba que se obtienen para cada test de confianza aplicado y en la tabla 4.36 se observan los niveles de confianza en donde la hipótesis nula que indica que los datos siguen una distribución Weibull es rechazada o no.

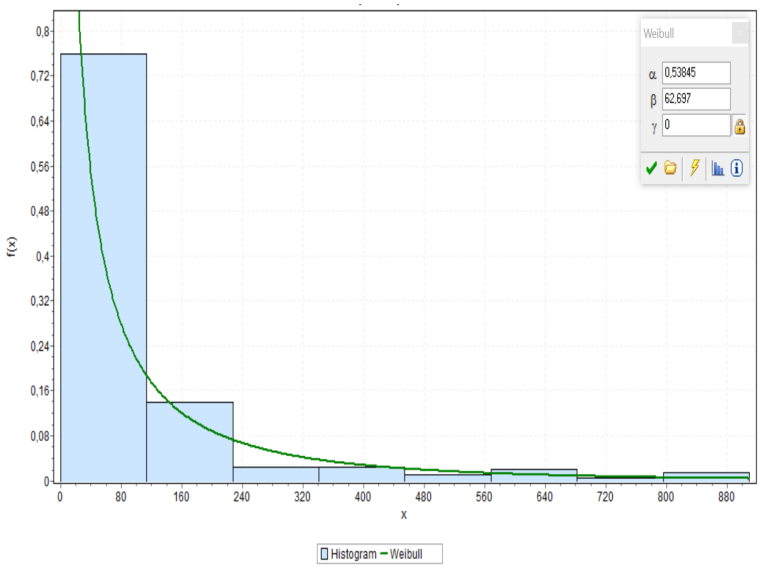


Figura 4.89: Ajuste distribución Weibull de los registros de falla del chancador terciario 7.

Tabla 4.35: Tests de confianza para tiempos entre fallas de chancador terciario 7.

Test	**Kolmogorov-Smirnov**	**Anderson-Darling**	χ^2
Estadístico	0,053	0,562	5,418

Tabla 4.36: Resultados de tests de bondad de ajuste para tiempos entre fallas de chancador terciario 7.

Kolmogorov-Smirnov					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	0,076	0,086	0,096	0,107	0,115
¿Rechazar?	No	No	No	No	No
Anderson-Darling					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	1,375	1,929	2,502	3,289	3,907
¿Rechazar?	No	No	No	No	No
χ^2					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	9,803	12,017	14,067	16,622	18,475
¿Rechazar?	No	No	No	No	No

La tabla 4.37 muestra los parámetros de la distribución Weibull que siguen los tiempos entre falla determinados para el chancador terciario 7.

Dada la distribución de los TEF se utiliza un ciclo de evaluación de 12 horas.

Tabla 4.37: Parámetros distribución Weibull para tiempos entre fallas de chancador terciario 7.

Parámetro	Valor
β	0,538
η	62,697
γ	0

4.12.1. Ciclo de evaluación de 12 horas

Se obtienen 102 registros los cuales se etiquetan con un 0 y corresponden a tiempos en los cuales el equipo no fallará. La figura 4.90 muestra cómo se distribuyen los 0 y 1 en el conjunto de datos.

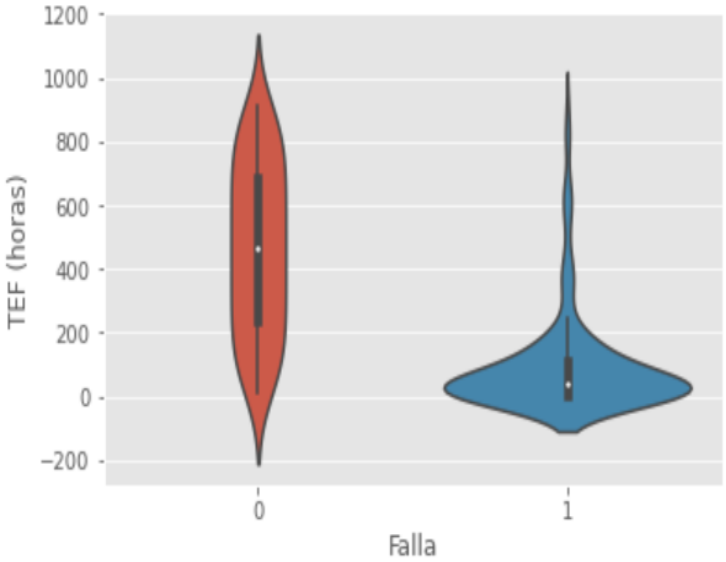


Figura 4.90: Diagrama de violín para chancador terciario 7.

La figura 4.91 muestra la matriz de confusión resultante al aplicar una regresión logística a los datos considerando un ciclo de evaluación de 12 horas.

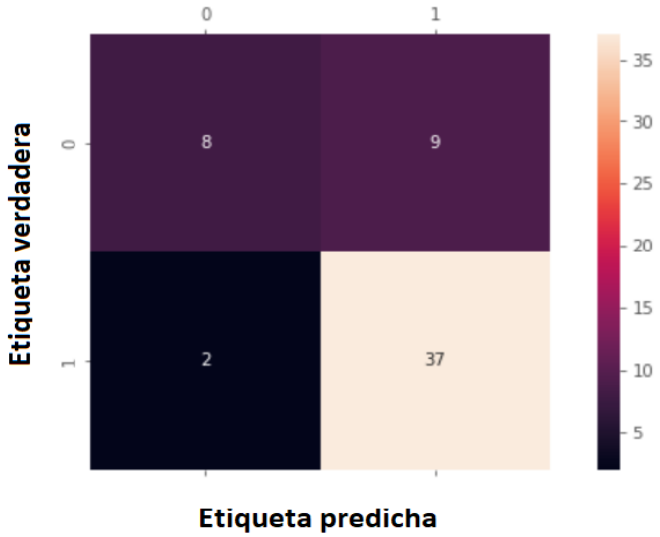


Figura 4.91: Matriz de confusión para chancador terciario 7.

$$Exactitud = \frac{8+37}{8+37+9+2} = \frac{45}{56} = 0,80 = 80\,\%$$

$$Precisión = \frac{37}{39} = 0,95 = 95\,\%$$

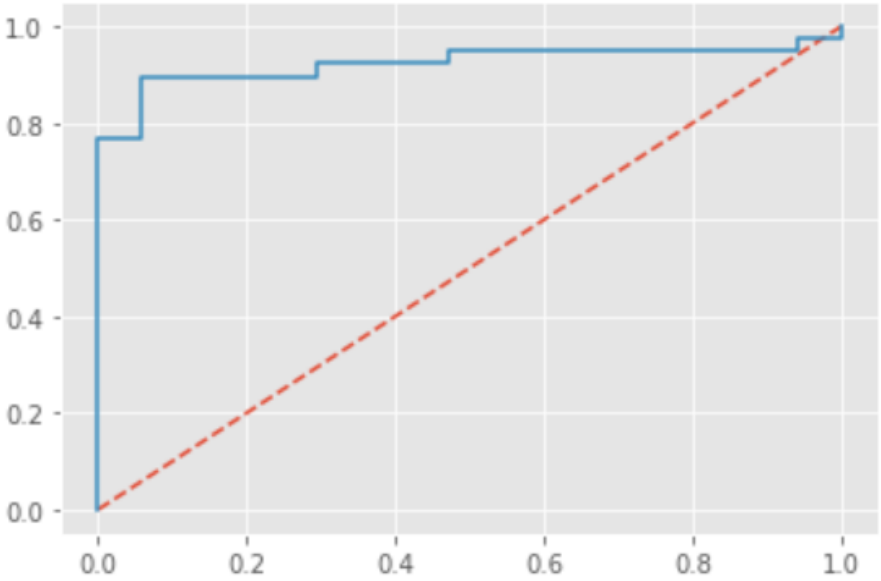


Figura 4.92: Curva ROC para chancador terciario 7.

$$AUC = 0,92$$

La figura 4.93 muestra la matriz de confusión resultante al aplicar el algoritmo Naïve-Bayes

a los datos considerando un ciclo de evaluación de 12 horas.

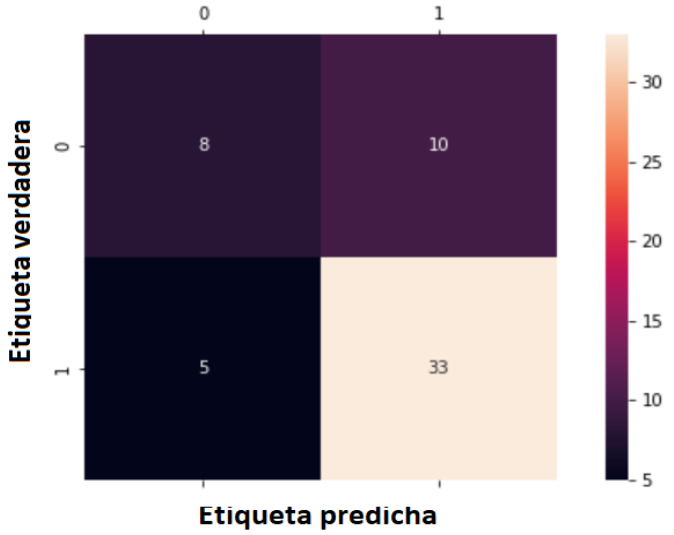


Figura 4.93: Matriz de confusión para chancador terciario 7.

$$Exactitud = \frac{8+33}{8+33+10+5} = \frac{41}{56} = 0,73 = 73\,\%$$

$$Precisión = \frac{33}{38} = 0,87 = 87\,\%$$

La figura 4.94 muestra la matriz de confusión resultante al aplicar el algoritmo K-NN a los datos considerando un ciclo de evaluación de 12 horas.

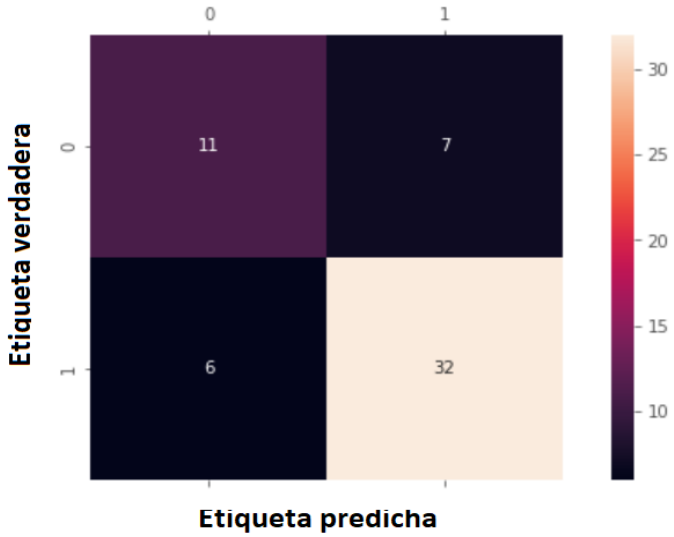


Figura 4.94: Matriz de confusión para chancador terciario 7.

$$Exactitud = \frac{11+32}{11+32+7+6} = \frac{43}{56} = 0,77 = 77\,\%$$

$$Precisión = \frac{32}{38} = 0,84 = 84\,\%$$

4.13. Chancador terciario 8

Tras filtrar y limpiar los datos se obtienen 211 registros de falla y por ende 210 TEF calculados. La figura 4.95 muestra la regresión lineal que se realiza con los TEF. Se observa que el coeficiente R^2 es de 0,983.

Además, considerando los 210 TEF calculados se tiene que:

$$MTBF = 87,9\ [hrs]$$

y el tiempo medio de reparación corresponde a:

$$MTTR = 5,7\ [hrs]$$

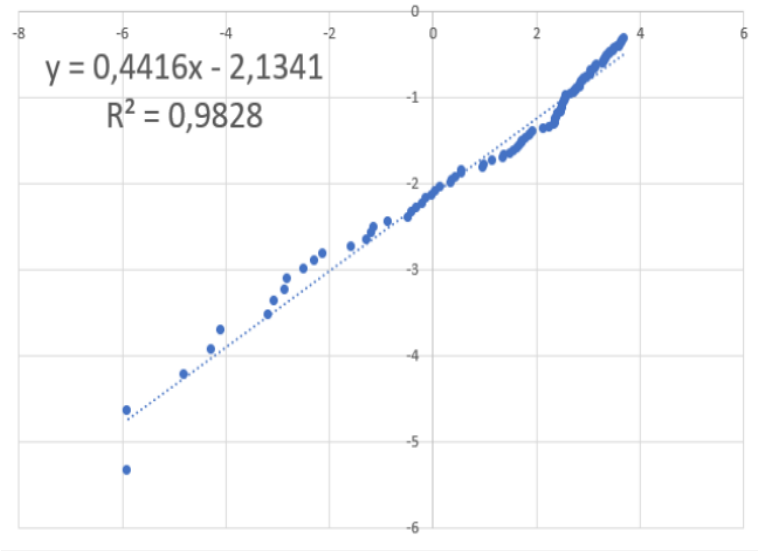


Figura 4.95: Regresión lineal.

En la figura 4.96 se observa el ajuste de los datos a una distribución Weibull. Por otro lado, en la tabla 4.38 se muestran los estadísticos de prueba que se obtienen para cada test de confianza aplicado y en la tabla 4.39 se observan los niveles de confianza en donde la hipótesis nula que indica que los datos siguen una distribución Weibull es rechazada o no.

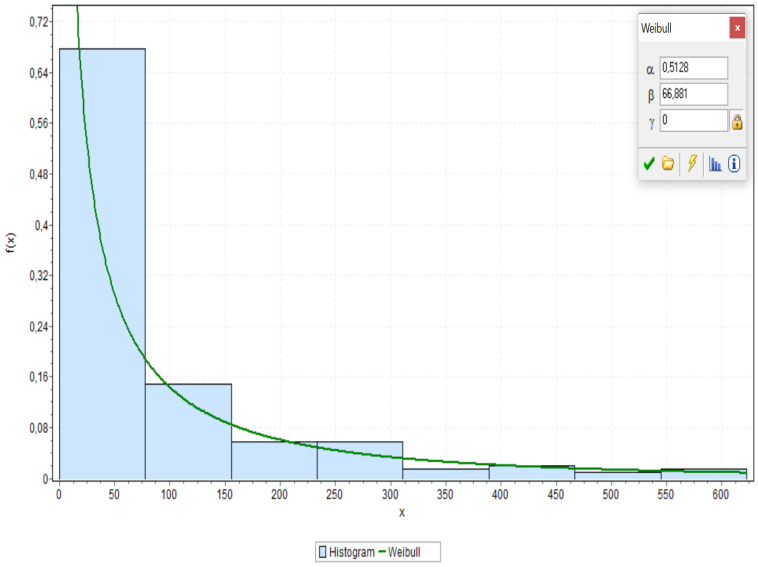


Figura 4.96: Ajuste distribución Weibull de los registros de falla del chancador terciario 8.

Tabla 4.38: Tests de confianza para tiempos entre fallas de chancador terciario 8.

Test	**Kolmogorov-Smirnov**	**Anderson-Darling**	χ^2
Estadístico	0,094	1,986	12,937

Tabla 4.39: Resultados de tests de bondad de ajuste para tiempos entre fallas de chancador terciario 8.

Kolmogorov-Smirnov					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	0,074	0,085	0,094	0,105	0,113
¿Rechazar?	Sí	Sí	No	No	No
Anderson-Darling					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	1,375	1,929	2,502	3,289	3,907
¿Rechazar?	Sí	Sí	No	No	No
χ^2					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	9,803	12,017	14,067	16,622	18,475
¿Rechazar?	Sí	Sí	No	No	No

La tabla 4.40 muestra los parámetros de la distribución Weibull que siguen los tiempos entre falla determinados para el chancador terciario 8.

Dada la distribución de los TEF se utiliza un ciclo de evaluación de 12 horas.

Tabla 4.40: Parámetros distribución Weibull para tiempos entre fallas de chancador terciario 8.

Parámetro	Valor
β	0,513
η	66,881
γ	0

4.13.1. Ciclo de evaluación de 12 horas

Se obtienen 102 registros los cuales se etiquetan con un 0 y corresponden a tiempos en los cuales el equipo no fallará. La figura 4.97 muestra cómo se distribuyen los 0 y 1 en el conjunto de datos.

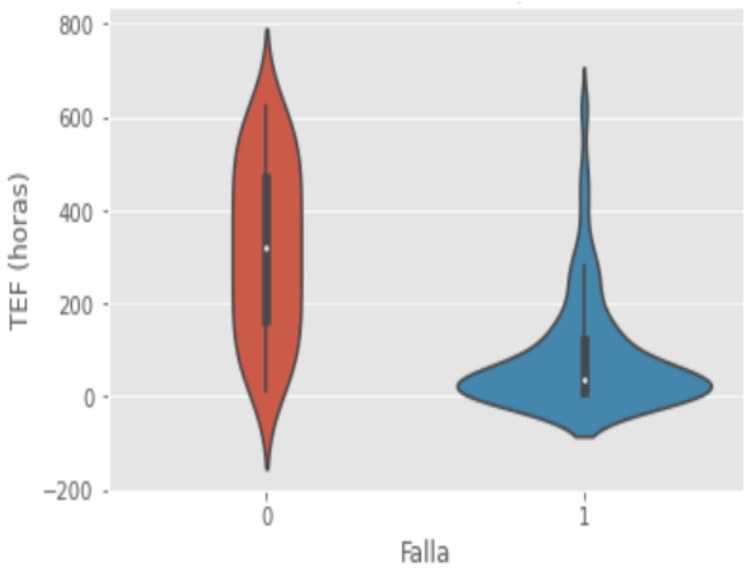


Figura 4.97: Diagrama de violín para chancador terciario 8.

La figura 4.98 muestra la matriz de confusión resultante al aplicar una regresión logística a los datos considerando un ciclo de evaluación de 12 horas.

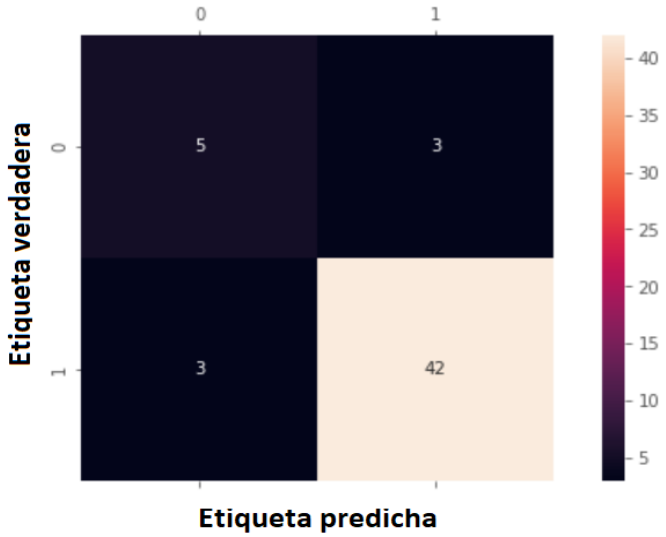


Figura 4.98: Matriz de confusión para chancador terciario 8.

$$Exactitud = \frac{5+42}{5+42+3+3} = \frac{47}{53} = 0,89 = 89\,\%$$

$$Precisión = \frac{42}{45} = 0,93 = 93\,\%$$

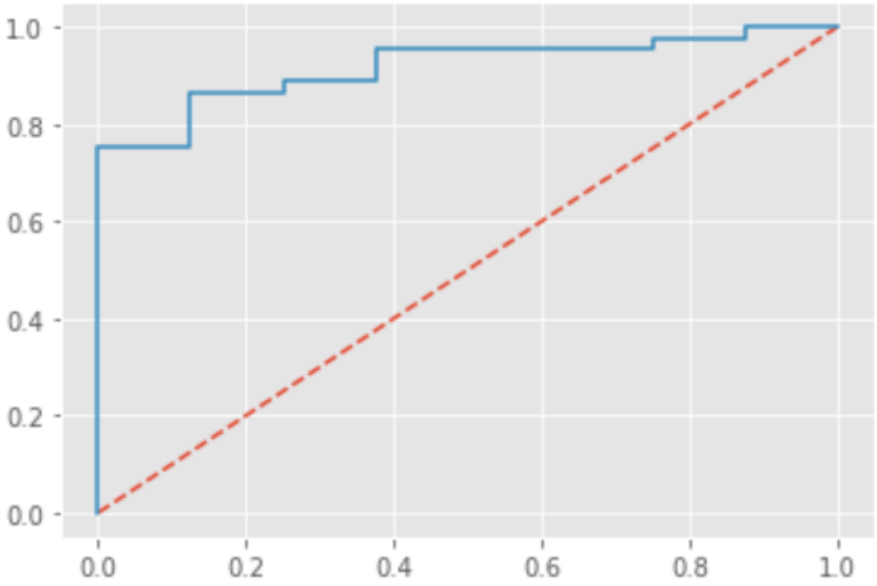


Figura 4.99: Curva ROC para chancador terciario 8.

$$AUC = 0,92$$

La figura 4.100 muestra la matriz de confusión resultante al aplicar el algoritmo Naïve-Bayes

a los datos considerando un ciclo de evaluación de 12 horas.

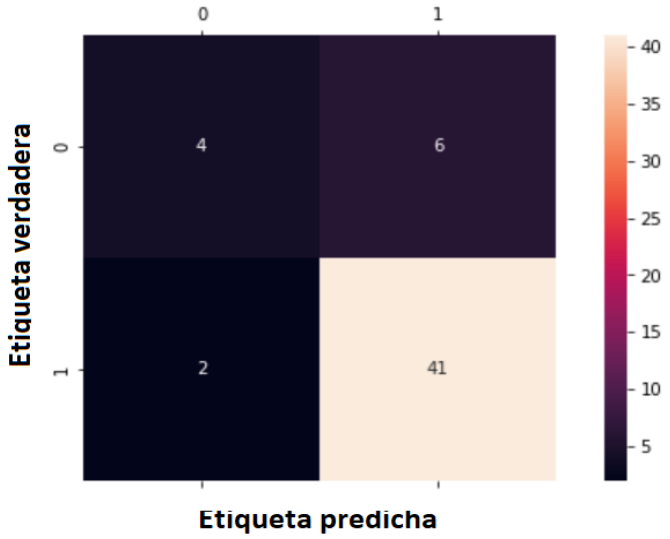


Figura 4.100: Matriz de confusión para chancador terciario 8.

$$Exactitud = \frac{4+41}{4+41+6+2} = \frac{45}{53} = 0,85 = 85\,\%$$

$$Precisión = \frac{41}{43} = 0,95 = 95\,\%$$

La figura 4.101 muestra la matriz de confusión resultante al aplicar el algoritmo K-NN a los datos considerando un ciclo de evaluación de 12 horas.

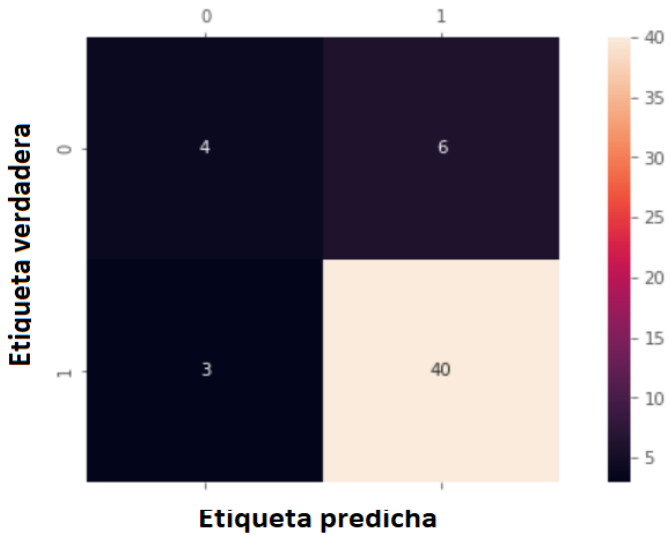


Figura 4.101: Matriz de confusión para chancador terciario 8.

$$Exactitud = \frac{4+40}{4+40+6+3} = \frac{44}{53} = 0,83 = 83\,\%$$

$$Precisión = \frac{40}{43} = 0,93 = 93\,\%$$

4.14. Chancador terciario 9

Tras filtrar y limpiar los datos se obtienen 268 registros de falla y por ende 267 TEF calculados. La figura 4.102 muestra la regresión lineal que se realiza con los TEF. Se observa que el coeficiente R^2 es de 0,985.

Además, considerando los 267 TEF calculados se tiene que:

$$MTBF = 71,4\ [hrs]$$

y el tiempo medio de reparación corresponde a:

$$MTTR = 3,6\ [hrs]$$

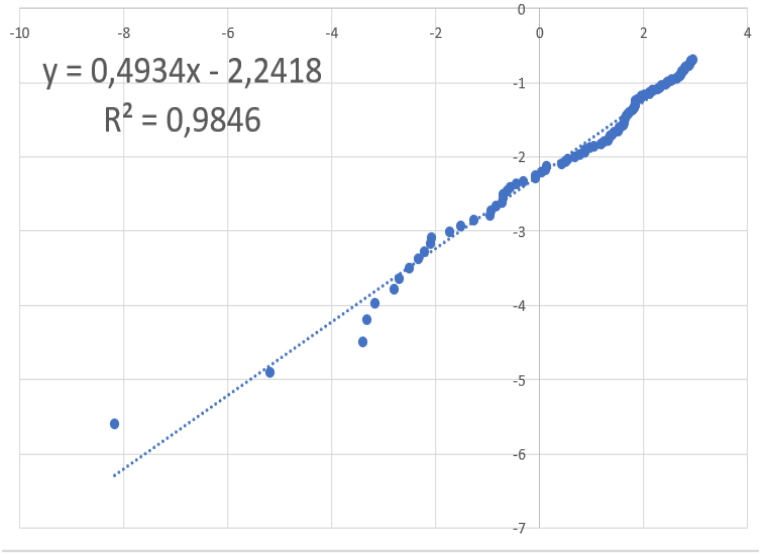


Figura 4.102: Regresión lineal.

En la figura 4.103 se observa el ajuste de los datos a una distribución Weibull. Por otro lado, en la tabla 4.41 se muestran los estadísticos de prueba que se obtienen para cada test de confianza aplicado y en la tabla 4.42 se observan los niveles de confianza en donde la hipótesis nula que indica que los datos siguen una distribución Weibull es rechazada o no.

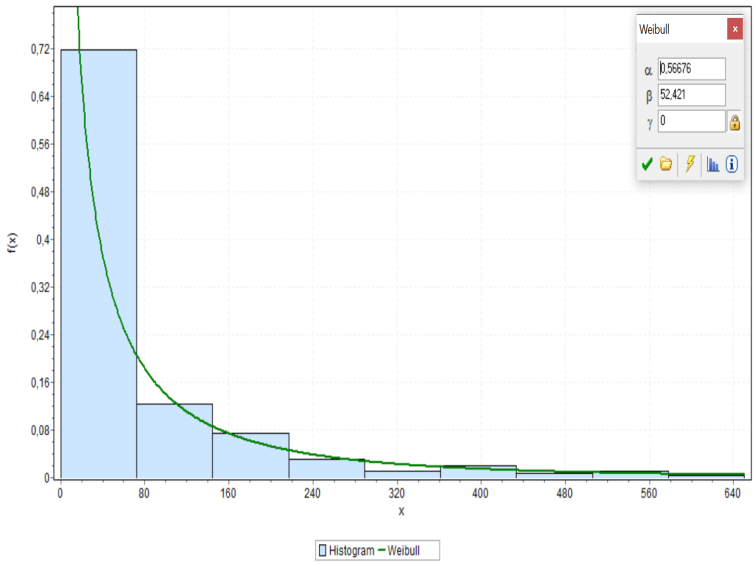


Figura 4.103: Ajuste distribución Weibull de los registros de falla del chancador terciario 9.

Tabla 4.41: Tests de confianza para tiempos entre fallas de chancador terciario 9.

Test	**Kolmogorov-Smirnov**	**Anderson-Darling**	χ^2
Estadístico	0,055	0,922	9,942

Tabla 4.42: Resultados de tests de bondad de ajuste para tiempos entre fallas de chancador terciario 9.

Kolmogorov-Smirnov					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	0,066	0,075	0,083	0,093	0,100
¿Rechazar?	No	No	No	No	No
Anderson-Darling					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	1,375	1,929	2,502	3,289	3,907
¿Rechazar?	No	No	No	No	No
χ^2					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	11,030	13,362	15,507	18,168	20,090
¿Rechazar?	No	No	No	No	No

La tabla 4.43 muestra los parámetros de la distribución Weibull que siguen los tiempos entre falla determinados para el chancador terciario 9.

Dada la distribución de los TEF se utiliza un ciclo de evaluación de 12 horas.

Tabla 4.43: Parámetros distribución Weibull para tiempos entre fallas de chancador terciario 9.

Parámetro	Valor
β	0,567
η	52,421
γ	0

4.14.1. Ciclo de evaluación de 12 horas

Se obtienen 55 registros los cuales se etiquetan con un 0 y corresponden a tiempos en los cuales el equipo no fallará. La figura 4.104 muestra cómo se distribuyen los 0 y 1 en el conjunto de datos.

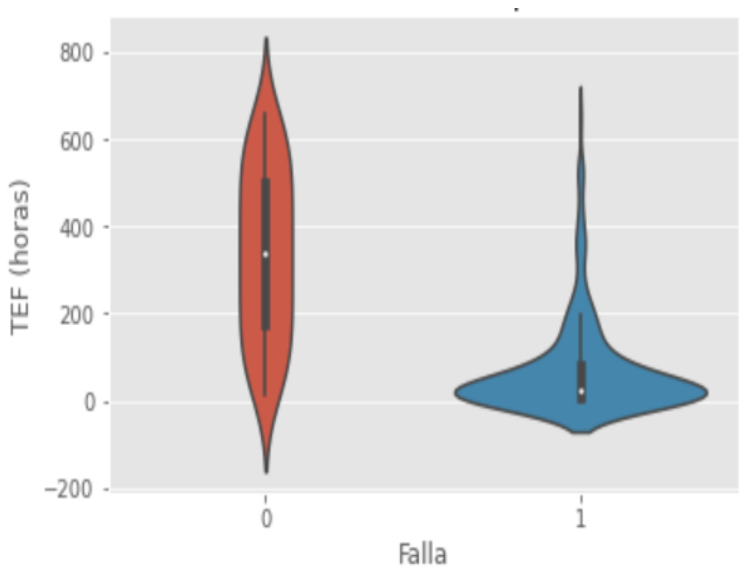


Figura 4.104: Diagrama de violín para chancador terciario 9.

La figura 4.105 muestra la matriz de confusión resultante al aplicar una regresión logística a los datos considerando un ciclo de evaluación de 12 horas.

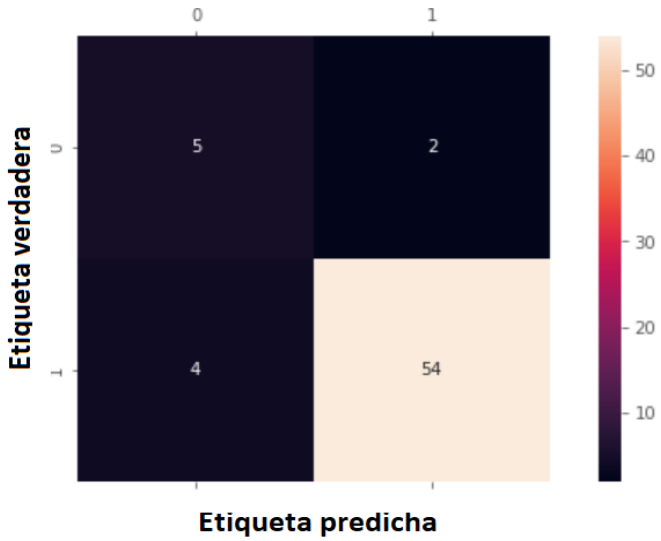


Figura 4.105: Matriz de confusión para chancador terciario 9.

$$Exactitud = \frac{5+54}{5+54+2+4} = \frac{59}{65} = 0,91 = 91\,\%$$

$$Precisión = \frac{54}{58} = 0,93 = 93\,\%$$

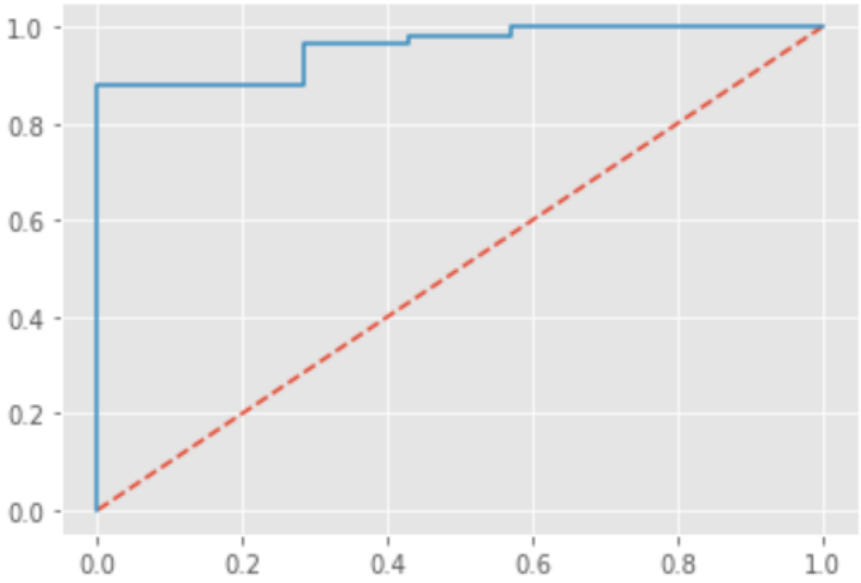


Figura 4.106: Curva ROC para chancador terciario 9.

$$AUC = 0,96$$

La figura 4.107 muestra la matriz de confusión resultante al aplicar el algoritmo Naïve-Bayes

a los datos considerando un ciclo de evaluación de 12 horas.

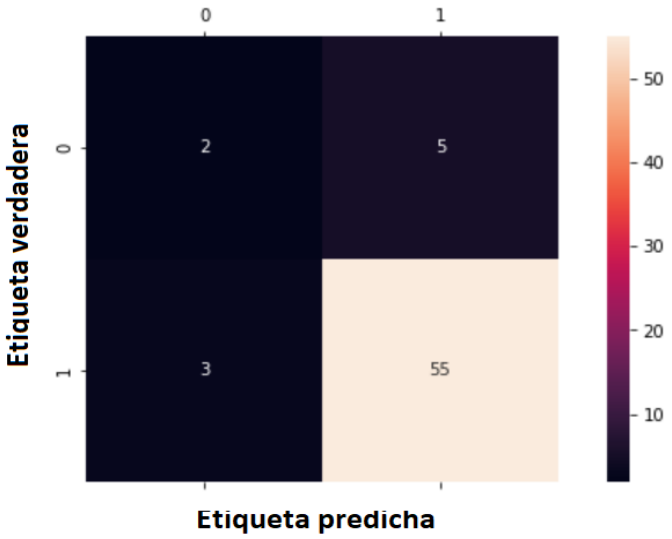


Figura 4.107: Matriz de confusión para chancador terciario 9.

$$Exactitud = \frac{2+55}{2+55+5+3} = \frac{57}{65} = 0,88 = 88\,\%$$

$$Precisión = \frac{55}{58} = 0,95 = 95\,\%$$

La figura 4.108 muestra la matriz de confusión resultante al aplicar el algoritmo K-NN a los datos considerando un ciclo de evaluación de 12 horas.

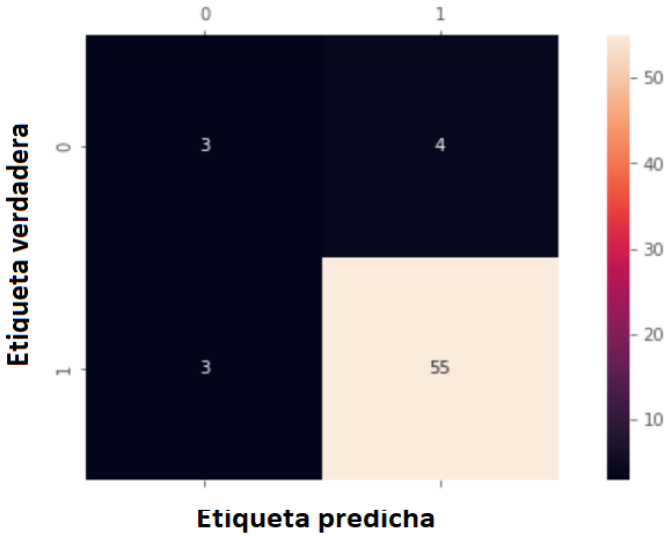


Figura 4.108: Matriz de confusión para chancador terciario 9.

$$Exactitud = \frac{3+55}{3+55+4+3} = \frac{58}{65} = 0,89 = 89\,\%$$

$$Precisión = \frac{55}{58} = 0,95 = 95\,\%$$

4.15. Chancador terciario 10

Tras filtrar y limpiar los datos se obtienen 169 registros de falla y por ende 168 TEF calculados. La figura 4.109 muestra la regresión lineal que se realiza con los TEF. Se observa que el coeficiente R^2 es de 0,924.

Además, considerando los 168 TEF calculados se tiene que:

$$MTBF = 105,2\ [hrs]$$

y el tiempo medio de reparación corresponde a:

$$MTTR = 5,1\ [hrs]$$

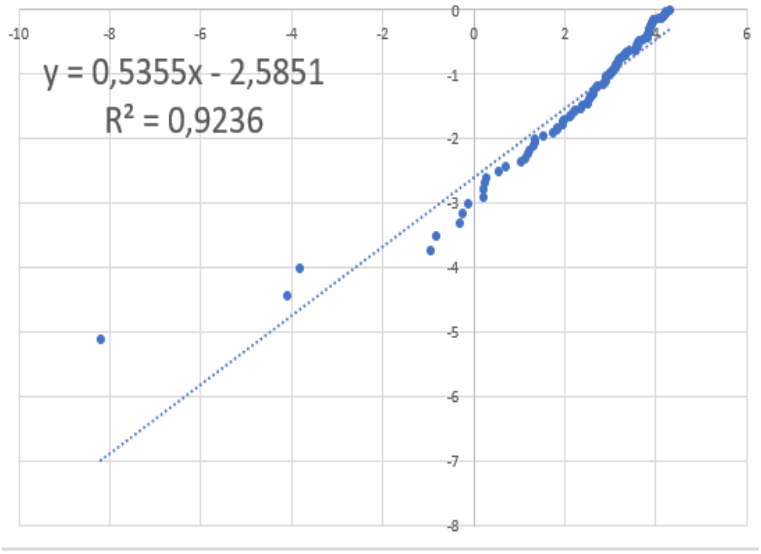


Figura 4.109: Regresión lineal.

En la figura 4.110 se observa el ajuste de los datos a una distribución Weibull. Por otro lado, en la tabla 4.44 se muestran los estadísticos de prueba que se obtienen para cada test de confianza aplicado y en la tabla 4.45 se observan los niveles de confianza en donde la hipótesis nula que indica que los datos siguen una distribución Weibull es rechazada o no.

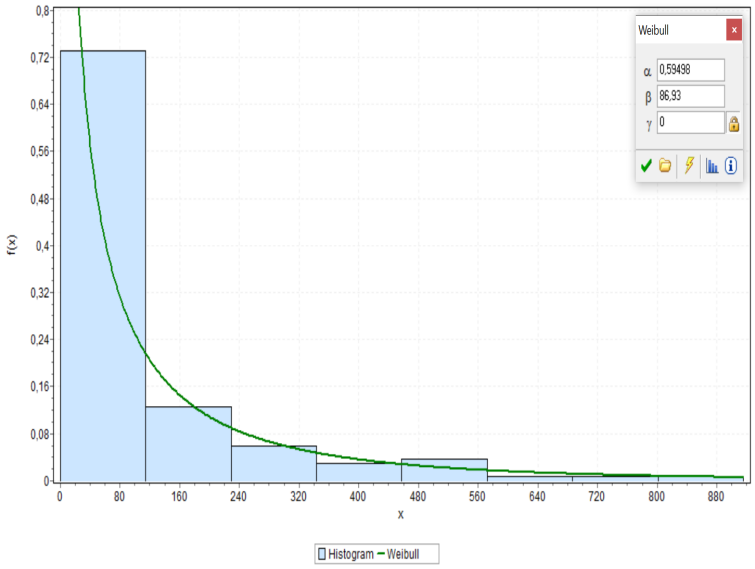


Figura 4.110: Ajuste distribución Weibull de los registros de falla del chancador terciario 10.

Tabla 4.44: Tests de confianza para tiempos entre fallas de chancador terciario 10.

Test	**Kolmogorov-Smirnov**	**Anderson-Darling**	χ^2
Estadístico	0,065	1,139	7,962

Tabla 4.45: Resultados de tests de bondad de ajuste para tiempos entre fallas de chancador terciario 10.

Kolmogorov-Smirnov					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	0,083	0,094	0,105	0,117	0,126
¿Rechazar?	No	No	No	No	No
Anderson-Darling					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	1,375	1,929	2,502	3,289	3,907
¿Rechazar?	No	No	No	No	No
χ^2					
Nivel de confianza	0,2	0,1	0,05	0,02	0,01
Valor crítico	9,803	12,017	14,067	16,622	18,475
¿Rechazar?	No	No	No	No	No

La tabla 4.46 muestra los parámetros de la distribución Weibull que siguen los tiempos entre falla determinados para el chancador terciario 10.

Dada la distribución de los TEF se utiliza un ciclo de evaluación de 11 horas.

Tabla 4.46: Parámetros distribución Weibull para tiempos entre fallas de chancador terciario 10.

Parámetro	Valor
β	0,595
η	86,930
γ	0

4.15.1. Ciclo de evaluación de 11 horas

Se obtienen 84 registros los cuales se etiquetan con un 0 y corresponden a tiempos en los cuales el equipo no fallará. La figura 4.111 muestra cómo se distribuyen los 0 y 1 en el conjunto de datos.

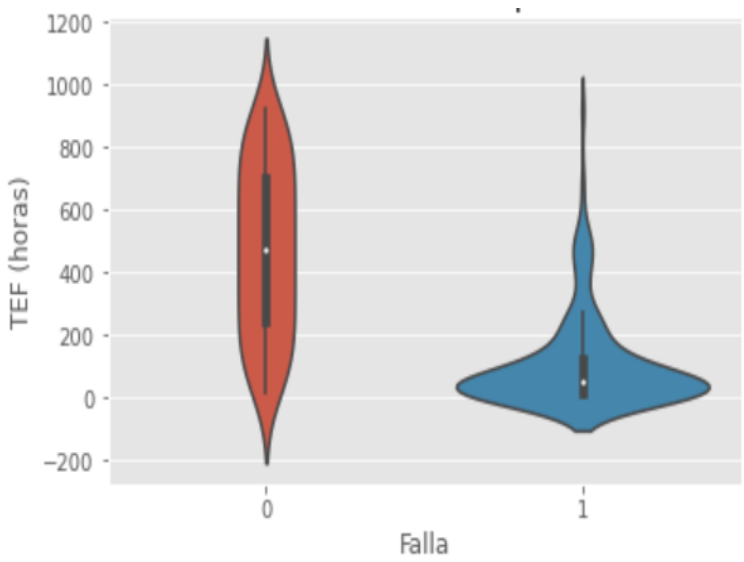


Figura 4.111: Diagrama de violín para chancador terciario 10.

La figura 4.112 muestra la matriz de confusión resultante al aplicar una regresión logística a los datos considerando un ciclo de evaluación de 11 horas.

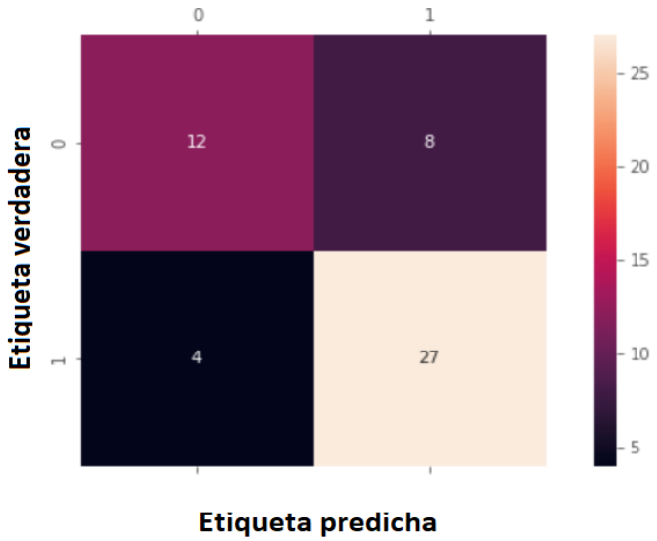


Figura 4.112: Matriz de confusión para chancador terciario 10.

$$Exactitud = \frac{12+27}{12+27+8+4} = \frac{39}{51} = 0,76 = 76\,\%$$

$$Precisión = \frac{27}{31} = 0,87 = 87\,\%$$

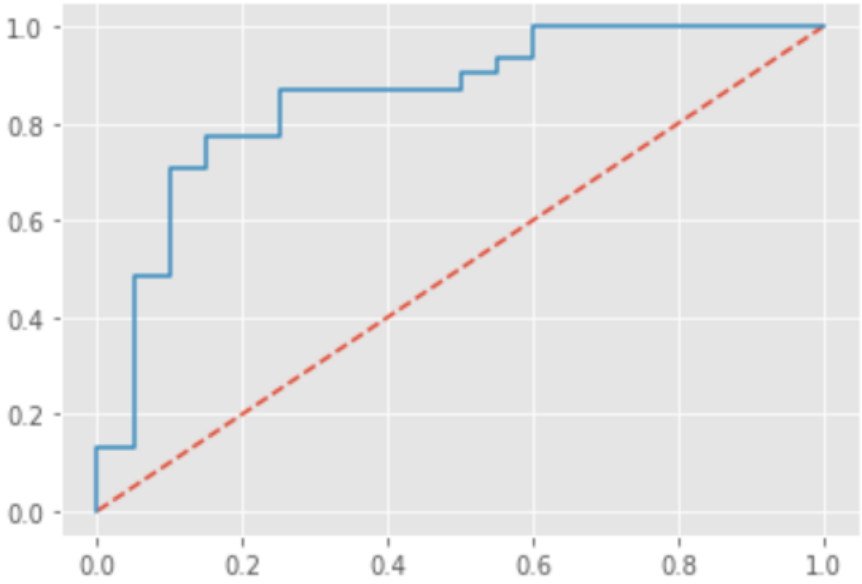


Figura 4.113: Curva ROC para chancador terciario 10.

$$AUC = 0,85$$

La figura 4.114 muestra la matriz de confusión resultante al aplicar el algoritmo Naïve-Bayes

a los datos considerando un ciclo de evaluación de 11 horas.

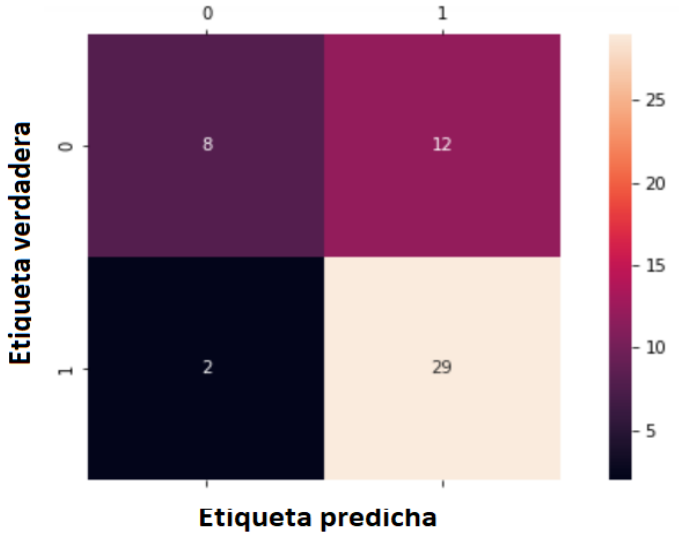


Figura 4.114: Matriz de confusión para chancador terciario 10.

$$Exactitud = \frac{8+29}{8+29+12+2} = \frac{37}{51} = 0,73 = 73\,\%$$

$$Precisión = \frac{29}{31} = 0,94 = 94\,\%$$

La figura 4.115 muestra la matriz de confusión resultante al aplicar el algoritmo K-NN a los datos considerando un ciclo de evaluación de 11 horas.

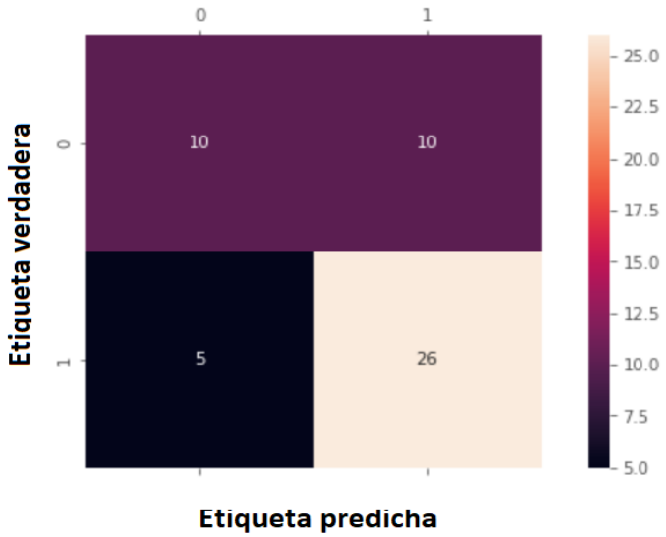


Figura 4.115: Matriz de confusión para chancador terciario 10.

$$Exactitud = \frac{10 + 26}{10 + 26 + 10 + 5} = \frac{36}{51} = 0,71 = 71\,\%$$

$$Precisión = \frac{26}{31} = 0,84 = 84\,\%$$

4.16. Resumen de resultados

La tabla 4.47 muestra un resumen de las precisiones y exactitudes de cada modelo aplicado a cada equipo, por otro lado, la figura 4.116 muestra la misma información de modo gráfico.

Tabla 4.47: Tabla con exactitud y precisión de cada modelo de *Machine Learning* por equipo.

Chancador	**Exactitud [%]**			**Precisión [%]**		
	Regresión Logística	**Naive-Bayes**	**KNN**	**Regresión Logística**	**Naive-Bayes**	**KNN**
MP1000	93	92	88	94	94	90
Symons B	82	75	73	67	71	48
Hydrocone H8800	75	83	81	91	90	87
Symons D	82	88	82	94	94	86
Symons E	87	70	68	97	83	72
Ch. terciario 1	82	76	73	96	93	86
Ch. terciario 2	79	82	77	91	95	80
Ch. terciario 3	84	76	68	89	78	77
Ch. terciario 4	79	82	81	93	100	97
Ch. terciario 5	82	86	78	97	93	79
Ch. terciario 6	90	83	82	98	91	89
Ch. terciario 7	80	73	77	95	87	84
Ch. terciario 8	89	85	83	93	95	93
Ch. terciario 9	91	88	89	93	95	95
Ch. terciario 10	76	73	71	87	94	84
Promedio	**83,4**	**80,8**	**78,1**	**91,7**	**90,2**	**83,1**

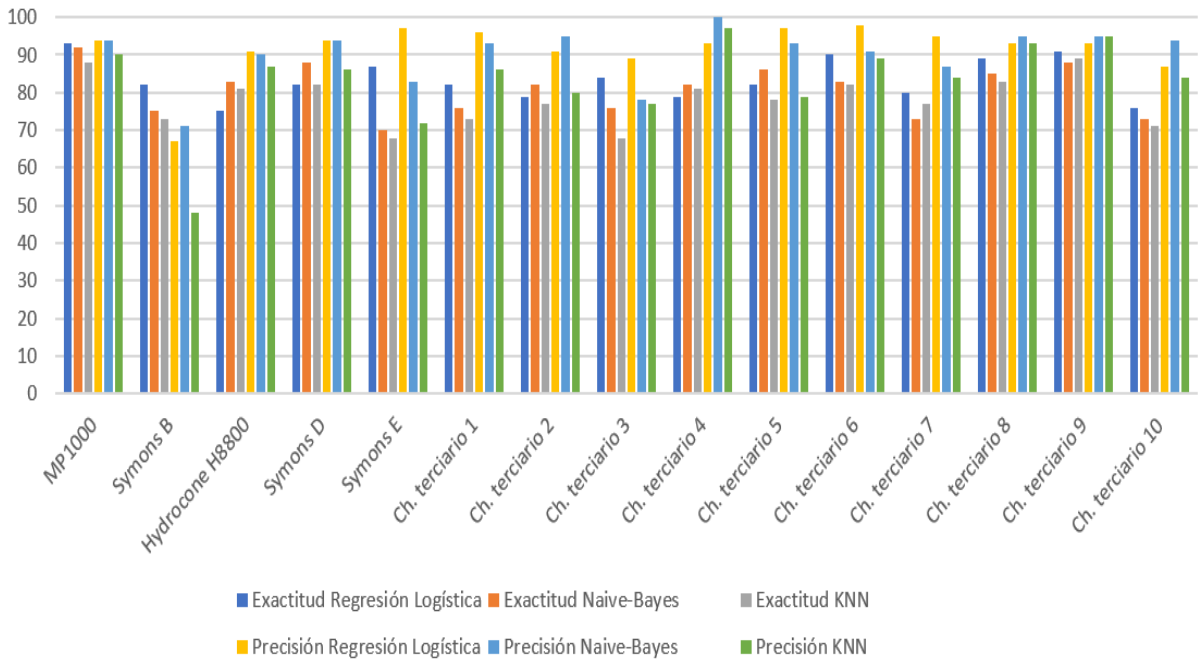


Figura 4.116: Exactitud y precisión de cada modelo de *Machine Learning* por equipo.

La tabla 4.48 muestra la producción de cada chancador y el costo asociado a la detención por hora de cada uno de ellos. Se asume una ley de cobre del mineral de 0,5 %, una recuperación de cobre en la planta de flotación de 83,5 % y un precio de cobre de 4 $[USD/lb]$ debido al alto valor del metal rojo en los últimos semestres.

Tabla 4.48: Producción de cobre y costo de detención por hora de cada chancador.

Chancador	**Producción [tph]**	**Cobre [tph]**	**Costo de detención por hora [USD/hr]**
MP1000	1.400	5,845	51.544
Symons B	850	3,549	31.295
Hydrocone H8800	1.100	4,593	40.499
Symons D	950	3,966	34.976
Symons E	950	3,966	34.976
Ch. terciario 1	560	2,338	20.618
Ch. terciario 2	560	2,338	20.618
Ch. terciario 3	340	1,420	12.518
Ch. terciario 4	340	1,420	12.518
Ch. terciario 5	440	1,837	16.200
Ch. terciario 6	440	1,837	16.200
Ch. terciario 7	380	1,587	13.991
Ch. terciario 8	380	1,587	13.991
Ch. terciario 9	380	1,587	13.991
Ch. terciario 10	380	1,587	13.991

4.17. Resultados Random Forest

Considerando la elevada exactitud y precisión de los tres modelos de *Machine Learning* para el chancador secundario MP1000 junto a su bajo $MTBF$ y la mayor producción que tiene con relación a los demás chancadores, lo que implica una mayor pérdida económica al tenerlo detenido, se propone realizar un *Random Forest* con este chancador para predecir el modo de falla del equipo.

Al analizar los Modos de Falla que se extraen de la columna "Catálogo" de la base de datos se observa que muchos de estos modos cuentan con pocos registros. La tabla 4.49 muestra la cantidad de instancias para cada uno de los modos de falla del equipo.

Tabla 4.49: Frecuencia de modos de falla de chancador MP1000.

Modo de falla	**Frecuencia**
Base	1
Buzón Alimentación	1
Buzón Móvil	1
Cinta	19
Compuerta	1
Contraeje	2
Corazas	4
Estructuras	6
Excéntrica/Quicionera	5
Fuerza/Control	30
Instrumentación	19
Módulos/Parrillas	0
Motor	18
Silos llenos	0
Sistema Hidráulico	13
Sistema Lubricación	130
Transmisión	10

Debido a la presencia de modos de falla que se repiten con una muy baja frecuencia se considera realizar el *Random Forest* con aquellos modos de falla que se repiten como mínimo 10 veces. Considerando lo anterior, el *Random Forest* se realiza con las siguientes variables categóricas y sus TEF respectivos:

- Cinta
- Fuerza/Control
- Instrumentación
- Motor

- Sistema Hidráulico
- Sistema de Lubricación
- Transmisión

Dado que los modos de falla eliminados corresponden a pocos registros en comparación al total, se supone que no hay pérdida de información relevante al eliminar estos TEF del conjunto de datos a utilizar por el algoritmo de *Random Forest*.

Con la intención de evaluar las técnicas de balanceo de datos (SMOTE) y la optimización de híper-parámetros con un "grid search" en *Python* se realiza el *Random Forest* considerando 3 casos los cuales se mencionan a continuación:

1. Sin aplicar balanceo de datos ni optimización de híper-parámetros.
2. Sin aplicar balanceo de datos, pero optimizando híper-parámetros.
3. Aplicando balanceo de datos y optimizando híper-parámetros.

La figura 4.117 muestra la frecuencia con la que aparece cada modo de falla considerado.

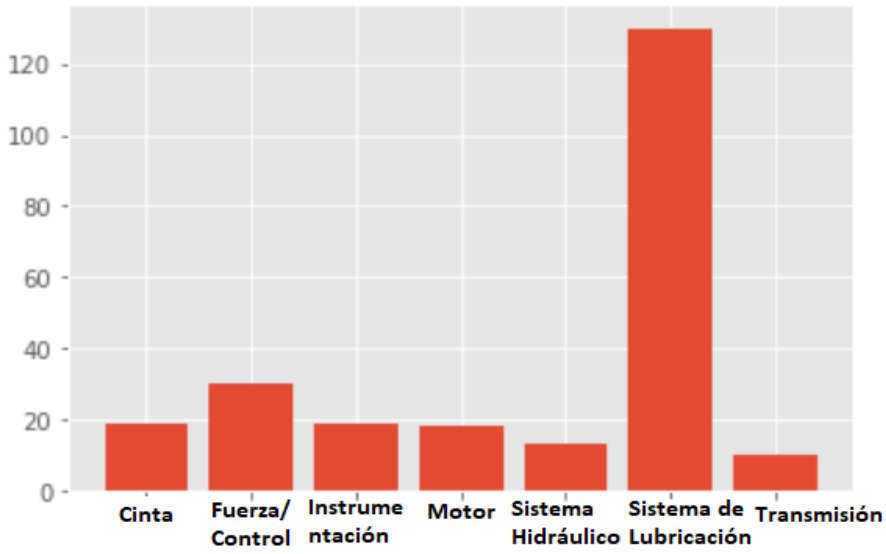


Figura 4.117: Histograma de los modos de falla considerados en *Random Forest* de chancador MP1000.

Dada la forma de trabajar de *Python* se requiere una codificación de las variables categóricas (modos de falla). Por lo anterior se le asignan números a cada modo de falla. La tabla 4.50 resume los códigos utilizados. Las matrices de confusión multiclase utilizan esta codificación para presentar sus resultados.

Tabla 4.50: Codificación de modos de falla de chancador MP1000.

Modo de falla	Codificación
Cinta	0
Fuerza/Control	1
Instrumentación	2
Motor	3
Sistema Hidráulico	4
Sistema Lubricación	5
Transmisión	6

4.17.1. Caso 1

Considerando este conjunto de datos y sin aplicar ninguna técnica que suponga mejores predicciones se obtiene la matriz de confusión multiclase que se muestra en la figura 4.118.

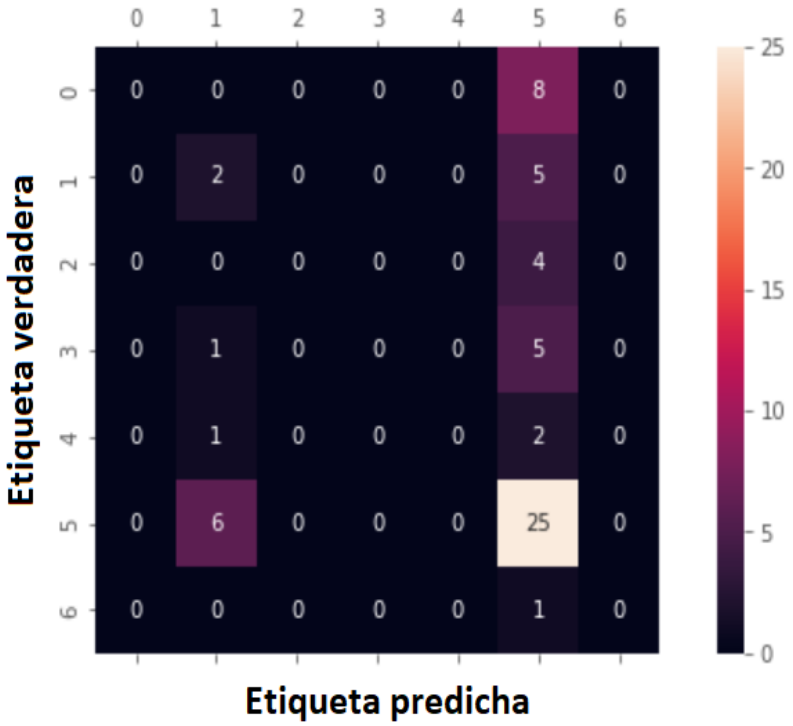


Figura 4.118: Matriz de confusión sin considerar balanceo de datos ni optimización de híper-parámetros.

$$Exactitud = \frac{2+25}{2+25+8+5+4+1+5+1+2+6+1} = \frac{27}{60} = 0,45 = 45\,\%$$

$$Precisión_{Cinta} = \frac{0}{8} = 0,00 = 0\,\%$$

$$Precisión_{Fuerza/Control} = \frac{2}{7} = 0,29 = 29\,\%$$

$$Precisión_{Instrumentación} = \frac{0}{4} = 0,00 = 0\,\%$$

$$Precisión_{Motor} = \frac{0}{6} = 0,00 = 0\,\%$$

$$Precisión_{Sistema\ Hidráulico} = \frac{0}{3} = 0,00 = 0\,\%$$

$$Precisión_{Sistema\ de\ Lubricación} = \frac{25}{31} = 0,81 = 81\,\%$$

$$Precisión_{Transmisión} = \frac{0}{1} = 0, = 0\,\%$$

4.17.2. Caso 2

Tras aplicar un "grid search" que considera la siguiente grilla de parámetros:

- Número de árboles: 20, 30, 40, 50, 100, 150, 200.
- Profundidad del árbol: Indefinida, 3, 5, 10.
- Criterio de pureza: Gini, Entropía.

Se obtiene que el mejor conjunto de parámetros corresponde a:

- Número de árboles: 100
- Profundidad del árbol: 3
- Criterio de impureza: Entropía

La figura 4.119 muestra la matriz de confusión multiclase resultante.

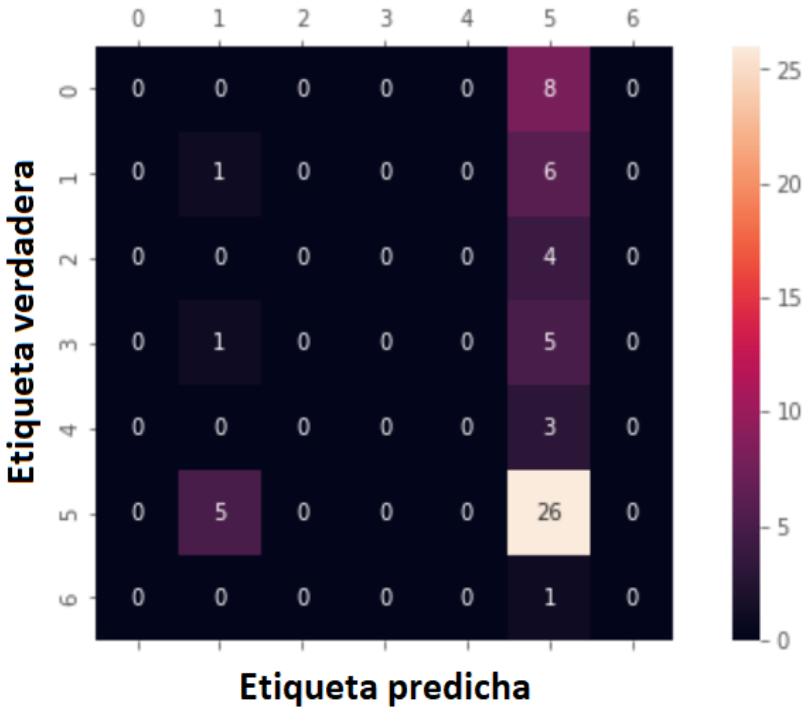


Figura 4.119: Matriz de confusión sin considerar balanceo de datos, pero optimizando híper-parámetros.

$$Exactitud = \frac{1+26}{1+26+8+6+4+1+5+3+5+1} = \frac{27}{60} = 0,45 = 45\,\%$$

$$Precisión_{Cinta} = \frac{0}{8} = 0,00 = 0\,\%$$

$$Precisión_{Fuerza/Control} = \frac{1}{7} = 0,14 = 14\,\%$$

$$Precisión_{Instrumentación} = \frac{0}{4} = 0,00 = 0\,\%$$

$$Precisión_{Motor} = \frac{0}{6} = 0,00 = 0\,\%$$

$$Precisión_{Sistema\ Hidráulico} = \frac{0}{3} = 0,00 = 0\,\%$$

$$Precisión_{Sistema\ de\ Lubricación} = \frac{26}{31} = 0,84 = 84\,\%$$

$$Precisión_{Transmisión} = \frac{0}{1} = 0,00 = 0\,\%$$

4.17.3. Caso 3

Dado que existen modos de falla con frecuencias bastante más elevadas que otras se requiere utilizar técnicas de balanceo de datos para que las predicciones tengan el menor sesgo posible

y el algoritmo cuente con un verdadero poder predictivo y no sólo prediga todo como la clase mayoritaria.

Se utiliza la técnica SMOTE para balancear los datos. La técnica SMOTE requiere fijar una cantidad de vecinos más cercanos para crear muestras sintéticas. Para decidir la cantidad de vecinos se examinan los resultados "score" y "oob_score_" (out of bag score) que indican respectivamente la proporción de predicciones acertadas considerando tanto los datos que se utilizan para crear el modelo *Random Forest* como aquellos que quedaron fuera y cuando sólo se utilizan los datos que no se utilizaron para crear los árboles de decisión del *Random Forest*. Estas métricas las entrega la ejecución del código en *Python* que se presenta en Anexos D. La tabla 4.51 entrega los resultados correspondientes.

Tabla 4.51: Score y out of bag score (predicciones acertadas/predicciones totales) para distina cantidad de vecinos más cercanos al crear muestras sintéticas mediante el uso de la técnica SMOTE.

Número de vecinos	**score**	**oob score**
1	0,57	0,55
2	0,46	0,49
3	0,44	0,37
4	0,35	0,37
5	0,37	0,37
6	0,33	0,28
7	0,28	0,29

Considerando los resultados obtenidos se decide utilizar sólo 1 vecino más cercano para crear muestras sintéticas mediante el uso de la técnica SMOTE. Se decide crear muestras sintéticas de las clases minoritarias para evitar pérdida de información al eliminar instancias de la clase mayoritaria (Sistema de Lubricación), por lo cual, cada modo de falla cuenta con 130 registros. La figura 4.120 muestra cómo se distribuyen porcentualmente los datos originales y cómo se distribuyen tras aplicar la técnica SMOTE.

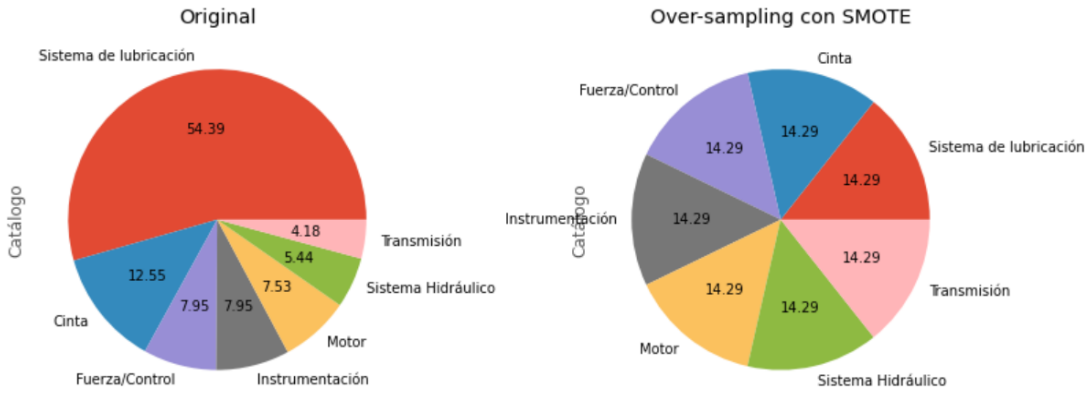


Figura 4.120: Gráficos de torta de los modos de falla de chancador MP1000. Derecha: Datos originales de entrada, Izquierda: Datos balanceados tras aplicar SMOTE.

El “grid search” ocupado entrega los siguientes valores para los híper-parámetros del modelo:

- Número de árboles: 150
- Profundidad del árbol: 10
- Criterio de pureza: Gini

La figura 4.121 muestra la matriz de confusión multiclase considerando tanto balanceo de datos efectuado con la técnica SMOTE como optimización de híper-parámetros realizado con un “grid search”.

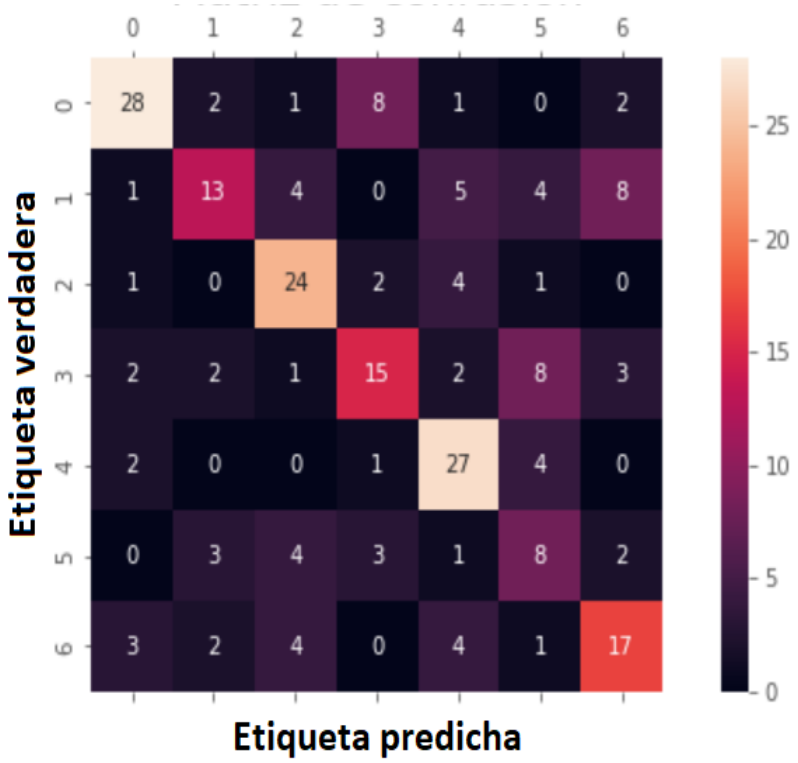


Figura 4.121: Matriz de confusión considerando balanceo de datos y optimización de híper-parámetros.

$$Exactitud = \frac{28+13+24+15+27+8+17}{228} = \frac{132}{228} = 0,58 = 58\,\%$$

$$Precisión_{Cinta} = \frac{28}{42} = 0,67 = 67\,\%$$

$$Precisión_{Fuerza/Control} = \frac{13}{35} = 0,37 = 37\,\%$$

$$Precisión_{Instrumentación} = \frac{24}{32} = 0,75 = 75\,\%$$

$$Precisión_{Motor} = \frac{15}{33} = 0,45 = 45\,\%$$

$$Precisión_{Sistema\ Hidráulico} = \frac{27}{34} = 0,79 = 79\,\%$$

$$Precisión_{Sistema\ de\ Lubricación} = \frac{8}{21} = 0,38 = 38\,\%$$

$$Precisión_{Transmisión} = \frac{17}{31} = 0,55 = 55\,\%$$

Dado el uso de la técnica SMOTE se crean muestras sintéticas las cuales no son registros reales de modos de falla. La figura 4.122 muestra la matriz de confusión correspondiente sólo a registros originales de modos de falla. Vale indicar que para la obtención de esta matriz de confusión se utiliza el modelo predictivo que se crea con los datos ya balanceados, y luego se evalúan los resultados considerando únicamente registros originales de TEF con sus respectivos modos de falla.

La figura 4.123 muestra uno de los 100 árboles de decisión utilizados por el *Random Forest* creado.

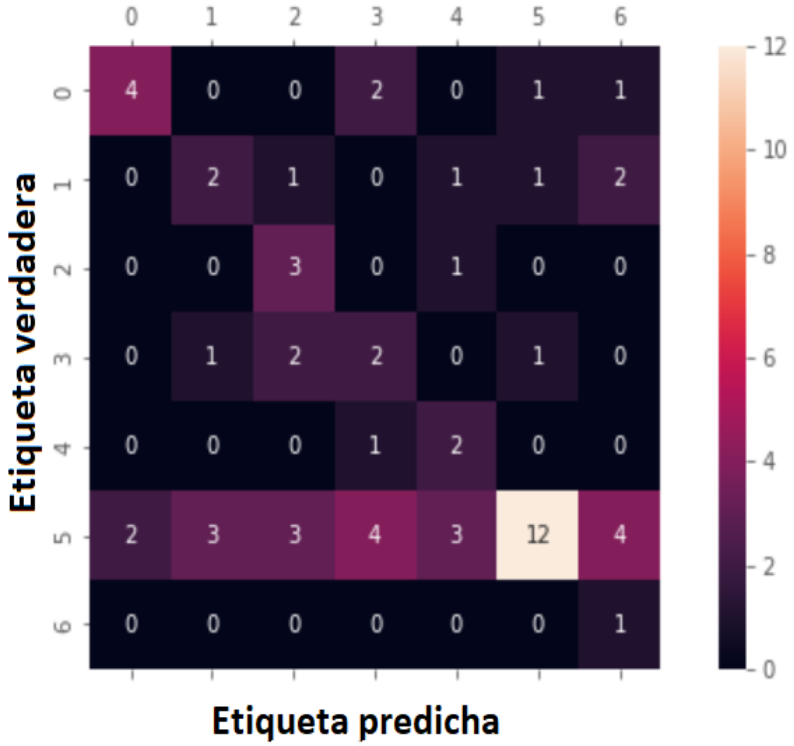


Figura 4.122: Matriz de confusión de registros originales considerando balanceo de datos y optimización de híper-parámetros.

$$Exactitud = \frac{4+2+3+2+2+12+1}{60} = \frac{26}{60} = 0,43 = 43\,\%$$

$$Precisión_{Cinta} = \frac{4}{9} = 0,50 = 50\,\%$$

$$Precisión_{Fuerza/Control} = \frac{2}{7} = 0,29 = 29\,\%$$

$$Precisión_{Instrumentación} = \frac{3}{4} = 0,75 = 75\,\%$$

$$Precisión_{Motor} = \frac{2}{6} = 0,33 = 33\,\%$$

$$Precisión_{Sistema\ Hidráulico} = \frac{2}{3} = 0,67 = 67\,\%$$

$$Precisión_{Sistema\ de\ Lubricación} = \frac{12}{31} = 0,39 = 39\,\%$$

$$Precisión_{Transmisión} = \frac{1}{1} = 1,00 = 100\,\%$$

La figura 4.123 muestra uno de los 150 árboles de decisión que conforman el *Random Forest*.

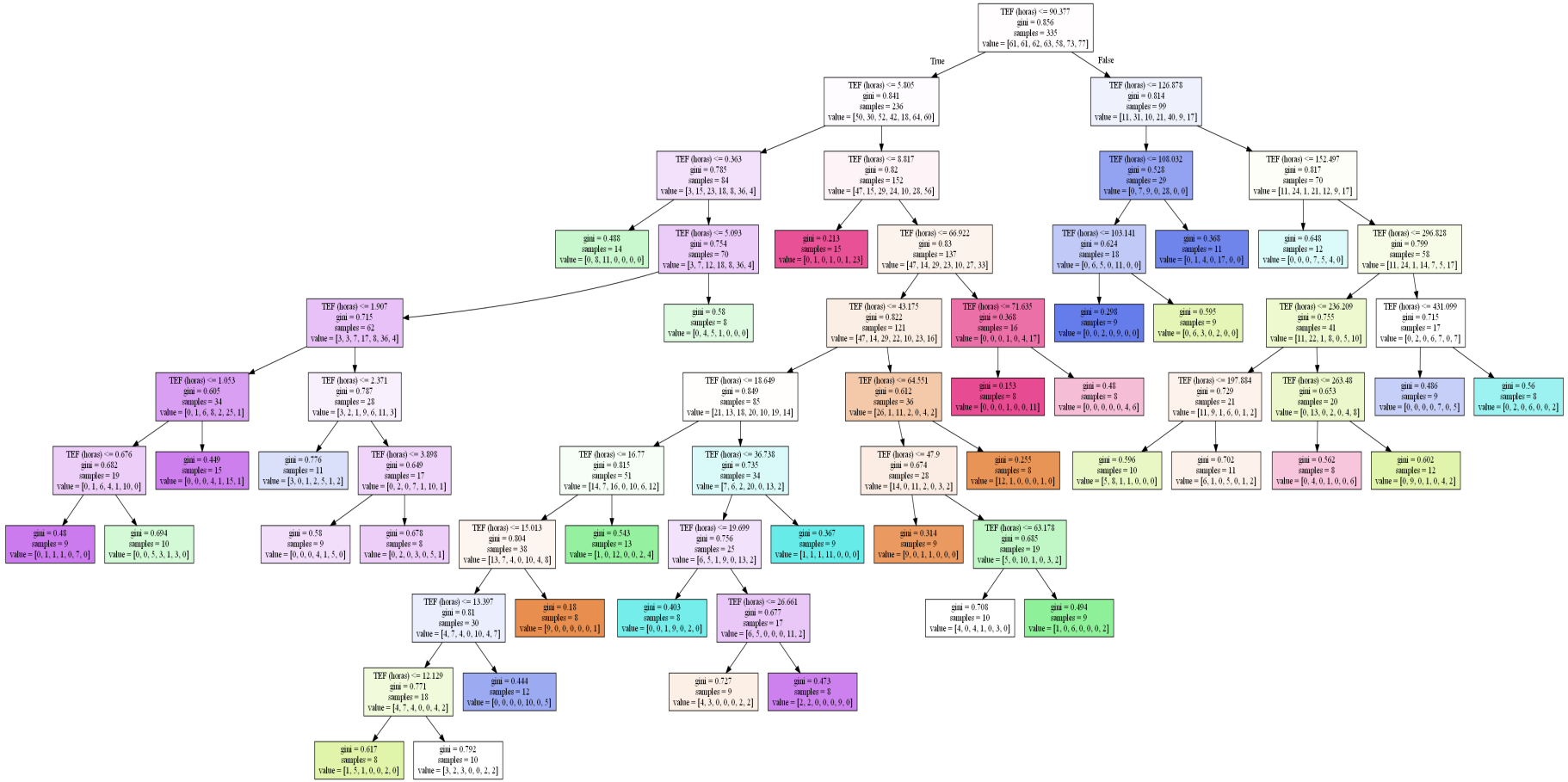


Figura 4.123: Árbol de decisión de *Random Forest* para el Caso 3 considerando balanceo de datos y optimización de híper-parámetros.

Capítulo 5

Análisis de resultados

Los tiempo medios entre falla (MTBF) varían entre 65,4 [hrs] para el chancador MP1000 y 188,3 [hrs] para el chancador Symnos de 7' de la sección B de la planta. Un menor MTBF implica que las fallas en el equipo son más frecuentes, particularmente, para el chancador MP1000 es posible asumir una falla cada 3 días en promedio. A partir de las figuras que muestran las regresiones lineales realizadas como son las figuras 4.1, 4.18, 4.25, entre otras, se observa que el tratamiento que recibe la base de datos es adecuado debido a que los coeficientes de determinación en cada regresión lineal suelen ser muy cercanos a 1. Los diagramas de violín que se muestran en los resultados de cada chancador son prácticamente idénticos, lo anterior se debe a que los TEF de cada equipo sigue una distribución Weibull cuyos coeficientes β y η son del mismo orden haciendo que la distribución Weibull tome una forma exponencial, es decir, se está en presencia de equipos que presentan fallas tempranas, además, los registros de no falla siguen siempre una distribución equiespaciada por el ciclo de evaluación utilizado. Resulta importante destacar el hecho de que los tests de bondad de ajuste de Kolmogorov-Smirnov, Anderson-Darling y χ^2 se satisfacen para cada chancador a un nivel de confianza del 2 %, siendo para la mayoría satisfechos en todos los niveles de confianza calculados, lo que indica que la curva de distribución de los TEF de cada uno de los equipos está correctamente realizada.

En cuanto a la regresión logística se observa que los valores del AUC son por lo general mayores a 0,90 para cada chancador bajo estudio. El valor mínimo de 0,83 corresponde al chancador secundario Symons 7' de la sección D de la planta como se observa en la figura 4.36. Lo anterior indica que existe como mínimo un 83 % de probabilidad de que el modelo pueda distinguir entre fallas y no fallas.

Con relación a los resultados de los algoritmos de *Machine Learning* para el chancador MP1000 se observa que en los ciclos de evaluación de 4 y 6 horas los resultados en cuanto a precisión y exactitud son bastante positivos para los 3 modelos, sin embargo, se observa que los resultados obtenidos para el ciclo de evaluación de 12 horas son aún mejores. Esta mejoría se puede atribuir a que los datos se encuentran más desbalanceados (55 registros de no falla v/s 245 de falla) por lo que los 3 algoritmos tienden a entregar como resultado la clase mayoritaria, de todos modos, la precisión al determinar la clase minoritaria, es decir los 0, sigue siendo bastante alta como se observa en las figuras 4.14, 4.16 y 4.17.

Considerando los resultados del resto de chancadores se observa que en líneas generales la regresión logística y Naïve-Bayes cumplen mejor su tarea como predictores de falla en comparación al algortimo k-NN. Lo anterior se observa en la tabla 4.47 y en la figura 4.116. Los resultados para el chancador secundario Symons 7' de la sección B de la planta entrega claramente los resultados más discretos en términos de precisión y exactitud de cada uno de los 3 algoritmos utilizados. Por el contrario, los resultados del chancador secundario MP1000 y los chancadores terciarios 8 y 9 son los mejores al utilizar estas mismas métricas. La regresión logística tiende a tener mejores resultados en cuanto a exactitud y Naïve-Bayes en cuanto a precisión al determinar los tiempo entre falla para estos chancadores.

Se observa una clara dependencia en la exactitud y precisión de cada algoritmo con el ciclo de evaluación que se utiliza, no obstante, los conjuntos de datos a utilizar pueden seguir considerándose balanceados dado que los registros de falla y los de no falla a lo sumo tienen una relación de 5:1 lo que provoca que los algoritmos mantengan una correcta capacidad predictiva de la clase minoritaria. Los peores resultados en cuanto a predicción de la clase minoritaria vienen dados por el algoritmo Naïve-Bayes en el chancador terciario 1, los algoritmos de regresión logística y Naïve-Bayes en el chancador terciario 7, Naïve-Bayes y K-NN en el chancador terciario 8 y 9 y por último Naïve-Bayes en el chancador terciario 10, en todos estos casos la predicción de la clase minoritaria es levemente inferior a un 50 %. Por otro lado, la predicción de fallas (1's), ya sea clase minoritario o mayoritaria, siempre mantiene valores notablemente altos lo que resulta fundamental en la predicción de fallas debido al alto costo que tiene asociado predecir que el equipo no fallará cuando realmente el equipo sí va a fallar (falsos negativos).

Pese a que se utiliza sólo los TEF como variable independiente en los 3 algoritmos supervisados, los resultados del problema binario de determinar si existirá o no falla en los chancadores son bastante positivos. Lo anterior se explica por el hecho de que la distribución de TEF de cada equipo esconde amplia información sobre el estado del mismo equipo e incluso es posible calcular una curva de confiabilidad con esta distribución y con ella determinar la probabilidad de que el equipo falle en determinada ventana de tiempo.

La mayor capacidad productiva del chancador MP1000 junto a tener el más bajo MTBF justifica el hecho de querer determinar el modo de falla del equipo. En la figura 4.117 se observa cómo algunos modos de falla se repiten más que otros, en particular, las fallas debido al sistema de lubricación cuentan con una frecuencia evidentemente mayor que otras incluso llegando a tener una relación de 10:1 con respecto a otro tipo de fallas como por ejemplo las fallas debido al sistema hidráulico. Lo anterior justifica el hecho de usar técnicas de balanceo de datos como SMOTE. En la tabla 4.51 se observa cómo los resultados de "score" y "oob_score_" tienden a disminuir a medida que se utiliza una mayor cantidad de vecinos más cercanos para crear muestras sintéticas. Lo anterior es posible explicarlo debido a que al utilizar más vecinos, los registros de modos de falla tienden a distribuirse de un modo más homogéneo en la ventana de tiempo que considera los TEF, lo que finalmente se traduce en una pérdida de información de los TEF existentes cuando se considera un determinado modo de falla. Por otro lado, al utilizar sólo un vecino más cercano, se crean muestras sintéticas

con TEF muy similar al original lo que posteriormente le indica al *Random Forest* que en determinados tiempos debiese ocurrir determinado tipo de falla. Al comparar los resultados de la matriz de confusión de los 3 casos estudiados se observa que al no utilizar técnicas de balanceo de datos se tiende a predecir gran parte de las instancias como la clase mayoritaria (Sistema de lubricación) debido a que 25 de 60 predicciones hechas del conjunto de testeo son clasificadas como la clase mayoritaria. Al utilizar un "grid search" y mantener el conjunto de datos desbalanceado no se observan mejoras en el poder predictivo del *Random Forest* siendo los resultados bastante similares a los que se obtienen el caso previo. Una vez se utiliza SMOTE para balancear los datos y se utiliza un "grid search" para optimizar los híper-parámetros los resultados dados por el algoritmo cambian radicalmente debido a que a pesar de que la exactitud se mantiene medianamente discreta con un 58 % se observa en la figura 4.121 que las instancias son predichas de distintas maneras y no se tiende a sobrecargar una clase como ocurre en los casos anteriores con la clase mayoritaria, además, en la figura 4.122 se observa que las predicciones con los datos reales, es decir, sin considerar las muestras sintéticas creadas, se mantienen en valores similares a los anteriores por lo cual se puede inferir que la creación de muestras sintéticas no aporta nueva información al modelo que tienda a conducir las predicciones realizadas por el *Random Forest* en determinado sentido.

La figura 4.123 muestra uno de los 150 árboles de decisión utilizados en el *Random Forest*. Se observa que en los nodos terminales de este árbol no se alcanza la pureza deseada. Una forma de enmendar este inconveniente corresponde a aumentar la produndidad de los árboles, sin embargo, estos árboles se construyen con los híper-parámetros optimizados entregados por el "grid search". Considerando lo anterior es posible indicar que la impureza de los nodos terminales en cada árbol de decisión se debe a falta de información del modelo para predecir de modo fehaciente cada instancia.

Capítulo 6

Conclusiones y recomendaciones

Tras analizar la base de datos se determina que los TEF son la única variable independiente que puede ser utilizada por algoritmos de *Machine Learning* para la predicción de fallas de los chancadores de la planta. Dado que esta variable no se encuentra en la base de datos de modo explícito se hace necesario calcular los TEF lo que implica una limpieza de la base de datos la que se realiza sin inconveniente alguno. Debido a la tarea a realizar se proponen algoritmos supervisados de *Machine Learning* ampliamente difundidos y de sencilla implementación en *Python*. Una vez se fijan los ciclos de evaluación de cada chancador se logra construir los conjuntos de datos a utilizar por los algoritmos. Se observa un mejor desempeño al predecir el cuándo va a fallar un chancador por los algoritmos de Regresión Logística y Naïve-Bayes presentando el primero de ellos una menor variabilidad en cuanto a sus resultados, sin embargo, se observa una clara dependencia de los resultados con la razón de instancias de falla y no falla. A pesar de lo anterior los algoritmos utilizados presentan un buen comportamiento entregando una baja cantidad tanto de falsos negativos como de falsos positivos. En cuanto a los resultados del *Random Forest* se hace necesaria la utilización de técnicas de balanceo de datos para dotar al algoritmo de un verdadero poder predictivo debido a que la no utilización de estas técnicas implica muchas veces predecir una instancia como la clase mayoritaria. En el caso del chancador MP1000 se observa cómo las fallas por el sistema de lubricación son mucho más frecuentes que las otras. De todas formas se observan discretos resultados en cuanto a exactitud del algoritmo lo que se atribuye al uso de tan sólo una variable independiente en el proceso, no obstante, es posible utilizar los resultados del *Random Forest* para determinar el cómo va a fallar el chancador (modo de falla) y sumado a la predicción de los modelos anteriores es posible elaborar pautas de inspección de los equipos considerando que si se determina que ocurrirá una falla en determinado momento existe una alta probabilidad de que la falla sea por el sistema de lubricación o el modo de falla entregado por el *Random Forest* en caso de ser distinto al sistema de lubricación. Lo anterior permite un ahorro sustancial en costos de mantenimiento dado que una corrección preventiva resulta bastante más barata que una correctiva. Se sugiere realizar un análisis económico detallado para determinar la cuantía de ahorros en mantenimiento y también fijar el momento adecuado de labores de mantenimiento y su periodicidad. Por otro lado, se sugiere incorporar mayor número de variables independientes como son el tipo de roca, dureza y work index (W_i) de la roca a chancar, y también parámetros asociados a los equipos como son su tiempo de uso, temperatura, presión en los descansos, entre otros, a los modelos para hallar relaciones más poderosas entre estas variables y así contar con técnicas de predicción más adecuadas.

Bibliografía

[1] Akpinar E. K. and Akpinar S. (2004). Determination of the wind energy potential for Maden-Elazig, Turkey, Energy Conversion and Management Vol. 45, pp. 2901–2914. Obtenido de: <https://doi.org/10.1016/j.enconman.2003.12.016>.

[2] Amat, J. (2020). Regresión logística con Python. Obtenido de Ciencia de Datos: <https://www.cienciadedatos.net/documentos/py17-regresion-logistica-python.html>.

[3] Arróspide, C. (2008). Glosario de términos en la gestión de mantenimiento, modelo de mantención y reparación, Santiago, Chile.

[4] Barrios, J.C. (2019). La matriz de confusión y sus métricas. Obtenido de juanbarrios.com: <https://www.juanbarrios.com/la-matriz-de-confusion-y-sus-metricas/>.

[5] Bravo, G. (2014). Evaluación técnica-económica de depósito de relleno en el rajo de la mina Chuquicamata Proyecto Mina Chuquicamata Subterránea. Memoria para al título de Ingeniera civil de Minas, Santiago, Universidad de Chile. Obtenido de: <https://repositorio.uchile.cl/handle/2250/131816>.

[6] Chandra, B., Gupta, M. y Gupta, M.P. (2007). Robust Approach for Estimating Probabilities in Naive-Bayes Classifier. In International Conference on Pattern Recognition and Machine Intelligence (pp. 11-16), Springer, Berlin, Heidelberg. Obtenido de: <https://link.springer.com/chapter/10.1007/978-3-540-77046-6_2>.

[7] Christofferson R. D., Gillette D.A. (1987). A simple estimator of the shape factor of the two- parameter Weibull distribution. Journal of Climate and Applied Meteorology. Vol26, pp 323–5. Obtenido de: <https://journals.ametsoc.org/view/journals/apme/26/2/1520-0450_1987_026_0323_aseots_2_0_co_2.xml>.

[8] Codelco, Vicepresidencia corporativa de proyectos. (2009). Estudio de prefactibilidad Proyecto Mina Chuquicamata Subterránea, Pricipales Decisiones. Obtenido de docplayer.es: <https://docplayer.es/10309086-Codelco-chile-vicepresidencia-corporativa-de-proyectos-estudio-de-prefactibilidad-proyecto-mina-chuquicamata-subterranea-principales-decisiones.html>.

[9] Concejero, P. (2005). Comparación de modelos de curvas ROC para la evaluación de procedimientos estadísticos de predicción en investigación de mercados, Capítulo 4 El análisis de curvas ROC, Tesis Doctoral. Universidad Complutense de Madrid, Madrid, España.

[10] Cook N.J. (2001). Discussion on modern estimation of the parameters of the Weibull wind speed distribution for wind speed energy analysis by J.V. Seguro, T.W. Lambert. J Wind Eng Ind Aerodyn;892001–9.

[11] Dorvlo Atsu S.S. (2012). Estimating wind speed distribution. Energy Conversion and Ma-

nagement Vol. 43 pp. 2311–2318. Obtenido de: <https://doi.org/10.1016/S0196-8904(01)00182-0>.

[12] Ferrero, R. (2020). Qué son los árboles de decisión y para qué sirven. Obtenido de Máxima Formación: <https://www.maximaformacion.es/blog-dat/que-son-los-arboles-de-decision-y-para-que-sirven/#:~:text=El%20%C3%ADndice%20de%20Gini%20mide,variable%20con%20menor%20Gini%20ponderado.>.

[13] González, F. (2005). Teoría y Práctica del Mantenimiento Industrial Avanzado, España, Fundación Confemetal.

[14] Justus, C.G., Hargraves W.R., Mikhail, A. and Graber, D. (1978). Methods for Estimating Wind Speed Frequency Distributions, J Appl Meteorol, Vol. 17, pp. 350–353. Obtenido de: <https://doi.org/10.1175/1520-0450(1978)017<0350:MFEWSF>2.0.CO;2>.

[15] Kececioglu, D. (1991). Reliability Engineering Handbook. Prentice Hall Inc, Englewood Cliffs.

[16] Lizarazo, A (2021), Entiende Naive-Bayes y sus fundamentos de implementación, Obtenido de Platzi: <https://platzi.com/tutoriales/2081-ds-probabilidad/9010-entiende-naive-bayes-y-sus-fundamentos-de-implementacion/?gclid=Cj0KCQjwz7uRBhDRARIsAFqjulne2-MdH93GvcYGuxZo9mwDurTJXQGqPJmE8XAVgd-OhzgMXPDkrzsaAu3iEALw_wcB&gclsrc=aw.ds>.

[17] López, A. (2021). Cómo Saber Si Tu Modelo Está Aprendiendo: Métricas De Evaluación De Modelos. Obtenido de Machine Learning En Espanol.com: <https://machinelearningenespanol.com/2021/01/05/como-saber-si-tu-modelo-esta-aprendiendo-metricas-de-evaluacion-de-modelos/>.

[18] Madrigal, E. (2004). Cap. 2, Conceptos básicos de confiabilidad en estimación e inferencia de los parámetros de la distribución Hockey Stick.(pp.24-31), Departamento de Ingeniería Industrial y Textil, Escuela de Ingeniería, Universidad de las Américas, Puebla, México.

[19] Mancuzo, G. (2020). Fallas en el Mantenimiento | Definición y Análisis, Argentina, ComparaSoftware Santiago Recicla. Obtenido de: <https://blog.comparasoftware.com/fallas-en-el-mantenimiento/>.

[20] Melillanca, E. (2018). Evaluación de modelos de clasificación: Matriz de Confusión y Curva ROC, Obtenido de: <http://www.ericmelillanca.cl/content/evaluaci-n-modelos-clasificaci-n-matriz-confusi-n-y-curva-roc>.

[21] Merkle (2020). El algoritmo K-NN y su importancia en el modelado de datos, Obtenido de Merkle: <https://www.merkleinc.com/es/es/blog/algoritmo-knn-modelado-datos>.

[22] Metso, Minerals (2001). Manual de servicio Chancador de cono Symons $4\frac{1}{4}'$, $5\frac{1}{2}'$ & $7'$. Obtenido de: <https://www.academia.edu/33635789/Manual_Symons>.

[23] Metso, Minerals (2004). Manual de instrucciones Chancador de conos Nordberg de la Serie MP. Obtenido de: <https://1library.co/document/zwr6xnly-chancador-nordberg-series-cone-crushers-instuction-manual-spanish.html>.

[24] Rodríguez, V. (2018). Decision trees / Árboles de decisión para clasificar en python. Obtenido de vincentblogxyz: <https://vincentblog.xyz/posts/decision-trees-arboles-de-decision-para-clasificar-en-python>.

[25] Salazar, M. & Rojano, A. & Figueroa, E. & Pérez, F. Aplicaciones de la distribución Weibull en Ingeniería, México, COLMEME UAN.

[26] Sandvik (2002). H8800 Instrucción de mantenimiento con lista de piezas de recambio. Obtenido de: <https://dokumen.tips/documents/75327883-chancador-sandvik-h8800-manual-s223-420-00-es.html?page=1>.

[27] Seguro J. V. and Lambert T. W. (2000). Modern estimation of the parameters of the Weibull wind speed distribution for wind energy analysis. J Wind Eng. Ind. Aerodyn., vol. 85, pp 75-84. Obtenido de: <https://doi.org/10.1016/S0167-6105(99)00122-1>.

[28] Serrano, J. (2013). Comparación de métodos para determinar los parámetros de Weibull para la generación de energía eólica, Pamplona, España, Scientia et Technica Año XVIII, Vol. 18, No 2. Obtenido de: <https://www.redalyc.org/articulo.oa?id=84929153004>.

[29] Swets, J.A. (1995). Signal detection theory and ROC analysis in psychology and diagnostics: Collected papers. Lawrence Erlbaum Associates. Obtenido de: <https://doi.org/10.4324/9781315806167>.

[30] The Machine Learners (2022). ¿Qué funciona mejor para evaluar un modelo de clasificación en Python? Obtenido de: <https://www.themachinelearners.com/curva-roc-vs-prec-recall/#Que_es_la_curva_de_precision-sensibilidad>.

[31] Torres, J. (2007). Diseño de un sistema de apoyo para la administración de capacidad en la industria de arriendo de equipos industriales. Tesis para optar al grado de Magíster en Gestión de Operaciones, Santiago, Universidad de Chile. Obtenido de: <https://repositorio.uchile.cl/handle/2250/102941>.

[32] Vallalta, J. (2021). Aprendizaje supervisado y no supervisado. Obtenido de IA Health Data Miner: <https://healthdataminer.com/data-mining/aprendizaje-supervisado-y-no-supervisado/>.

[33] Wallace, R., Blischke, D. N., Prabhakar,M. (2000) Reliability Modeling Prediction and Optimization. California, Estados Unidos, John Wiley Sons Inc.

[34] Yiu, T. (2019). Understanding Random Forest. Obtenido de towardsdatascience.com: <https://towardsdatascience.com/understanding-random-forest-58381e0602d2>.

Anexos

Anexo A

Código en *Python* de Regresión Logística

[ ]:
```
# Tratamiento de datos
# ==============================================================
import pandas as pd
import numpy as np

# Gráficos
# ==============================================================
import matplotlib.pyplot as plt
from matplotlib import style
import seaborn as sns

# Preprocesado y modelado
# ==============================================================
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
import statsmodels.api as sm
import statsmodels.formula.api as smf
from statsmodels.stats.weightstats import ttest_ind

# Configuración matplotlib
# ==============================================================
plt.rcParams['image.cmap'] = "bwr"
#plt.rcParams['figure.dpi'] = "100"
plt.rcParams['savefig.bbox'] = "tight"
style.use('ggplot') or plt.style.use('ggplot')

# Configuración warnings
# ==============================================================
import warnings
```

```
warnings.filterwarnings('ignore')
```

[ ]:
```
dataframe = pd.read_csv(r"C:
↪\Users\pxo_t\OneDrive\Desktop\data_set_ch_3_9.csv")
dataframe.head(10)
```

[ ]:
```
dataframe.Falla.value_counts().sort_index()
```

[ ]:
```
eje_x = dataframe[['TEF (horas)']].values.reshape(-1, 1)
eje_y = dataframe[['Falla']].values.reshape(-1, 1)

fig, ax = plt.subplots()
ax.scatter(eje_x, eje_y)
plt.show()
```

[ ]:
```
# Gráfico
# ==============================================================
fig, ax = plt.subplots(figsize=(6, 3.84))

sns.violinplot(
      x     = 'Falla',
      y     = 'TEF (horas)',
      data  = dataframe,
      #color = "white",
      ax    = ax
   )

ax.set_title('Distribución de fallas por hora');
```

[ ]:
```
# Defino algunas variables a utilizar
X = dataframe[['TEF (horas)']]
y = dataframe['Falla']
w = dataframe.drop(['Falla'],1)
```

[ ]:
```
# T-test entre clases
# ==============================================================
res_ttest = ttest_ind(
            x1 = dataframe[['TEF (horas)']].values.reshape(-1,␣
↪1)[dataframe[['Falla']] == 0],
            x2 = dataframe[['TEF (horas)']].values.reshape(-1,␣
↪1)[dataframe[['Falla']] == 1],
            alternative='two-sided'
         )
print(f"t={res_ttest[0]}, p-value={res_ttest[1]}")
```

[ ]:
```
v = dataframe

# División de los datos en train y test
# ==============================================================
train, test = train_test_split(v, test_size = 0.20)

# Creación del modelo
# ==============================================================
# Para no incluir ningún tipo de regularización en el modelo se indica
# penalty='none'
modelo = LogisticRegression(penalty='none')
modelo.fit(X = train['TEF (horas)'].values.reshape(-1, 1), y = train['Falla'].values.
 ↪reshape(-1,1))
```

[ ]:
```
# Información del modelo
# ==============================================================
print("Intercept:", modelo.intercept_)
print("Coeficiente:", list(zip(X.columns, modelo.coef_.flatten(), )))
print("Accuracy de entrenamiento:", modelo.score(X, y))

#Extra que se puede borrar (AUC)
testy = test['Falla'].values.reshape(-1, 1)
```

[ ]:
```
# Predicciones probabilísticas
# ==============================================================
# Con .predict_proba() se obtiene, para cada observación, la probabilidad predicha
# de pertenecer a cada una de las dos clases.
columna_1=['No Falla','Falla predictiva']
predicciones = modelo.predict_proba(X = test['TEF (horas)'].values.reshape(-1, 1))
#Valores de .predict_proba entran para el cálculo de auc
predicciones_auc = predicciones
#Transformación de tipo de dato para crear tabla y poder imprimir resultados en␣
 ↪Excel
predicciones = pd.DataFrame(predicciones, columns = columna_1)
predicciones.head(20)

#Datos Extras para AUC
predictions = predicciones['Falla predictiva'].values.reshape(-1, 1)
type(predictions)
```

[ ]:
```
clasificacion=np.where(predicciones['Falla predictiva']<0.5,0,1)
clasificacion_df=pd.DataFrame(clasificacion, columns=['Prediccion'])
clasificacion_df.head(40)
```

[ ]:
```
test.reset_index(inplace=True)
clasificacion_df.reset_index(inplace=True)
predicciones.reset_index(inplace=True)
```

```
test_1 = pd.concat([test,clasificacion_df],axis=1)

test_2 = pd.concat([test_1,predicciones],axis=1)

test_3=test_2.drop(['index'], axis=1)

print(test_3)
```

[ ]:
```
for j in range(len(test_3)):
    if test_3.iloc[j,2] == 1:
        print("El equipo " + "MP1000" + " fallará a las " + str(test_3.iloc[j,0]) + "␣
 ↪horas de uso")
```

[ ]:
```
for i in range(len(test_3)):
    if test_3.iloc[i,2] == 1:
        dt = test_3.iloc[i,0]
        dt_2=dt*60
        print("El chancador " + " MP1000 " + " fallará a los " + str(dt_2) + "␣
 ↪minutos. ")
```

[ ]:
```
predicciones_1 = modelo.predict(test['TEF (horas)'].values.reshape(-1, 1))
print(predicciones_1)
print(test)
print(clasificacion)
accuracy = accuracy_score(
          y_true = test['Falla'].values.reshape(-1, 1),
          y_pred = clasificacion,
          normalize = True
         )
print("")
print(f"El accuracy de test es: {100*accuracy}%")
```

[ ]:
```
print(predicciones_1)
print(clasificacion)
```

[ ]:
```
from sklearn.metrics import confusion_matrix, classification_report
import seaborn as sn
cm = confusion_matrix(test['Falla'].ravel(), clasificacion)
fig, ax = plt.subplots(figsize=(10,5))
ax.matshow(cm)
plt.title('Matriz de confusion', fontsize=20)
plt.ylabel('Etiqueta Verdadera', fontsize=15)
plt.xlabel('Etiqueta Predicha', fontsize=15)
sn.heatmap(cm, annot=True)
plt.show()
```

```
for (i, j), z in np.ndenumerate(cm):
    ax.text(j, i, '{:0.1f}'.format(z), ha='center', va='center')
```

[ ]:
```
# Matriz de confusión de las predicciones de test
# ================================================
confusion_matrix = pd.crosstab(
    test['Falla'].ravel(),
    clasificacion,
    rownames=['Real'],
    colnames=['Predicción']
)

confusion_matrix
```

[ ]:
```
test_index=test_3[test_3["Falla predictiva"]<=0.7].index
horas_de_falla=test_3.drop(test_index)
horas_de_falla=horas_de_falla.drop(['Falla'], axis=1)
horas_de_falla=horas_de_falla.drop(['No Falla'], axis=1)
horas_de_falla=horas_de_falla.drop(['Prediccion'], axis=1)

print(horas_de_falla)
```

[ ]:
```
horas_de_falla.to_excel("Fallas.xlsx")
```

[ ]:
```
# roc curve
from sklearn.datasets import make_classification
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import roc_curve
from matplotlib import pyplot
from sklearn.metrics import roc_auc_score

# keep probabilities for the positive outcome only
# calculate roc curve
fpr, tpr, thresholds = roc_curve(testy, predictions)
# plot no skill
pyplot.plot([0, 1], [0, 1], linestyle='--')
# plot the roc curve for the model
pyplot.plot(fpr, tpr)
# show the plot
pyplot.show()
auc = roc_auc_score(testy, predicciones_auc[:, 1])
print(auc)
```

Anexo B

Código en *Python* de algoritmo Naïve-Bayes

[ ]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import sklearn
```

[ ]:
```
dataframe = pd.read_csv(r"C:
↪\Users\pxo_t\OneDrive\Desktop\data_set_ch_3_4.csv")
dataframe.head(10)
```

[ ]:
```
dataframe.Falla.value_counts().sort_index()
```

[ ]:
```
eje_x = dataframe[['TEF (horas)']].values.reshape(-1, 1)
eje_y = dataframe[['Falla']].values.reshape(-1, 1)

fig, ax = plt.subplots()
ax.scatter(eje_x, eje_y)
plt.show()
```

[ ]:
```
# Defino algunas variables a utilizar
X = dataframe[['TEF (horas)']]
y = dataframe[['Falla']]
print(X)
print(y)
```

[ ]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20,␣
↪random_state = 0)
```

[ ]:
```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
```

```
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

[ ]:
```
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(X_train, y_train)
```

[ ]:
```
y_pred  =  classifier.predict(X_test)
```

[ ]:
```
 y_pred
```

[ ]:
```
y_test
```

[ ]:
```
from sklearn.metrics import confusion_matrix, classification_report
import seaborn as sn
cm = confusion_matrix(y_test, y_pred)
fig, ax = plt.subplots(figsize=(10,5))
ax.matshow(cm)
plt.title('Matriz de confusion', fontsize=20)
plt.ylabel('Etiqueta Verdadera', fontsize=15)
plt.xlabel('Etiqueta Predicha', fontsize=15)
sn.heatmap(cm, annot=True)
plt.show()


for (i, j), z in np.ndenumerate(cm):
    ax.text(j, i, '{:0.1f}'.format(z), ha='center', va='center')
```

[ ]:
```
from sklearn.metrics import confusion_matrix,accuracy_score
cm = confusion_matrix(y_test, y_pred)
ac = accuracy_score(y_test,y_pred)
print(cm)
ac
```

Anexo C

Código en *Python* de algoritmo K-NN

[ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.colors import ListedColormap
import matplotlib.patches as mpatches
import seaborn as sb

%matplotlib inline
plt.rcParams['figure.figsize'] = (16, 9)
plt.style.use('ggplot')

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
```

[ ]:
```
dataframe = pd.read_csv(r"C:
↪\Users\pxo_t\OneDrive\Desktop\data_set_ch_3_10.csv")
dataframe.head(10)
```

[ ]:
```
dataframe.Falla.value_counts().sort_index()
```

[ ]:
```
eje_x = dataframe[['TEF (horas)']].values.reshape(-1, 1)
eje_y = dataframe[['Falla']].values.reshape(-1, 1)

fig, ax = plt.subplots()
ax.scatter(eje_x, eje_y)
plt.show()
```

[ ]:
```
# Defino algunas variables a utilizar
X = dataframe[['TEF (horas)']]
```

```
y = dataframe[['Falla']]
```

[ ]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20,␣
↪random_state = 0)
scaler = MinMaxScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

[ ]:
```
n_neighbors = 7

knn = KNeighborsClassifier(n_neighbors)
knn.fit(X_train, y_train)
print('Accuracy of K-NN classifier on training set: {:.2f}'
     .format(knn.score(X_train, y_train)))
print('Accuracy of K-NN classifier on test set: {:.2f}'
     .format(knn.score(X_test, y_test)))
```

[ ]:
```
pred = knn.predict(X_test)
from sklearn.metrics import confusion_matrix, classification_report
import seaborn as sn
cm = confusion_matrix(y_test, pred)
fig, ax = plt.subplots(figsize=(10,5))
ax.matshow(cm)
plt.title('Matriz de confusion', fontsize=20)
plt.ylabel('Etiqueta Verdadera', fontsize=15)
plt.xlabel('Etiqueta Predicha', fontsize=15)
sn.heatmap(cm, annot=True)
plt.show()


for (i, j), z in np.ndenumerate(cm):
    ax.text(j, i, '{:0.1f}'.format(z), ha='center', va='center')
```

[ ]:
```
pred = knn.predict(X_test)
print(confusion_matrix(y_test, pred))
print(classification_report(y_test, pred))
```

Anexo D

Código en *Python* de *Random Forest*

[ ]:
```
# Tratamiento de datos
# ==============================================================
import pandas as pd
import numpy as np

# Gráficos
# ==============================================================
import matplotlib.pyplot as plt
from matplotlib import style
import seaborn as sns

# Preprocesado y modelado
# ==============================================================
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.datasets import make_classification
from sklearn.model_selection import cross_validate
from sklearn.model_selection import RepeatedStratifiedKFold
import statsmodels.api as sm
import statsmodels.formula.api as smf
from statsmodels.stats.weightstats import ttest_ind

# Configuración matplotlib
# ==============================================================
plt.rcParams['image.cmap'] = "bwr"
#plt.rcParams['figure.dpi'] = "100"
plt.rcParams['savefig.bbox'] = "tight"
style.use('ggplot') or plt.style.use('ggplot')

# Balanceo de datos
# ==============================================================
```

```
from imblearn.pipeline import Pipeline
from imblearn.combine import SMOTETomek
from imblearn.under_sampling import TomekLinks

# Configuración warnings
# ===============================================================
import warnings
warnings.filterwarnings('ignore')
```

[ ]:
```
dataframe = pd.read_csv(r"C:\Users\pxo_t\OneDrive\Desktop\bd_rf_codificada.
↪csv",encoding='latin-1')
dataframe.head(10)
```

[ ]:
```
from sklearn.preprocessing import LabelEncoder
from collections import Counter
from matplotlib import pyplot
index =["Cinta","Fuerza/Control","Instrumentación","Motor","Sistema␣
↪Hidráulico","Sistema Lubricación","Transmisión"]
# Defino algunas variables a utilizar
X = dataframe[['TEF (horas)']]
y = dataframe['Catálogo']
y_encode = LabelEncoder().fit_transform(y)
# summarize distribution
counter = Counter(y_encode)
for k,v in counter.items():
	per = v / len(y_encode) * 100
	print('Class=%d, n=%d (%.3f%%)' % (k, v, per))
# plot the distribution
pyplot.bar(counter.keys(), counter.values())
pyplot.show()
```

[ ]:
```
dataframe.Catálogo.value_counts().sort_index()
```

[ ]:
```
histogram = y.plot.hist(bins=30)
print(histogram)
plt.show()
```

[ ]:
```
dataframe.info()
```

[ ]:
```
from imblearn.under_sampling import TomekLinks
from imblearn.under_sampling import RandomUnderSampler
from imblearn.over_sampling import RandomOverSampler
from imblearn.datasets import make_imbalance
import matplotlib.pyplot as plt

fig, axs = plt.subplots(ncols=2, figsize=(10, 5))
autopct = "%.2f"
```

```
sampling_strategy = {4: 19, 10: 30, 11: 19, 12: 18, 13: 13, 14: 130, 15: 10}
rus = RandomUnderSampler(sampling_strategy=sampling_strategy)
X_res, y_res = rus.fit_resample(X, y)

y_res.value_counts().plot.pie(autopct=autopct, ax=axs[0])
axs[0].set_title("Original")

sampling_strategy = {4: 70, 10: 70, 11: 70, 12: 70, 13: 70, 14: 130, 15: 70}

from pandas import read_csv
from imblearn.over_sampling import SMOTE
from collections import Counter
from matplotlib import pyplot
from sklearn.preprocessing import LabelEncoder

oversample = SMOTE(sampling_strategy='not majority', k_neighbors=1)
X_res_os, y_res_os = oversample.fit_resample(X_res, y_res)
# summarize distribution
print(X_res_os)
print(y_res_os)

y_res_os.value_counts().plot.pie(autopct=autopct, ax=axs[1])
_ = axs[1].set_title("Over-sampling con SMOTE")
```

[ ]:
```
# summarize distribution
counter = Counter(y_res_os)
for k,v in counter.items():
      per = v / len(y_res_os) * 100
      print('Class=%d, n=%d (%.3f%%)' % (k, v, per))
# plot the distribution
pyplot.bar(counter.keys(), counter.values())
pyplot.show()
```

[ ]:
```
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error
```

[ ]:
```
# Dividimos los datos en entrenamiento y prueba
from sklearn.model_selection import train_test_split
# u son nuestras variables independientes
u = dataframe.drop(["Catálogo"],axis = 1)
# v es nuestra variable dependiente
v = dataframe.Catálogo

# División 75% de datos para entrenamiento, 25% de daatos para test
```

```
X_train, X_test, y_train, y_test = train_test_split(X_res_os, y_res_os,␣
↪random_state=0)
```

[ ]:
```
# Creamos el modelo de Bosques Aleatorios (y configuramos el número de␣
↪estimadores (árboles de decisión))
from sklearn.ensemble import RandomForestClassifier
BA_model = RandomForestClassifier(n_estimators = 15,
                                  criterion = "gini",
                                  max_features="sqrt",
                                  max_depth=5,
                                  bootstrap=True,
                                  random_state = 2016,
                                  min_samples_leaf = 8,
                                  max_samples = 2/3 ,
                                  oob_score=True)
```

[ ]:
```
from sklearn.datasets import load_boston
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import plot_confusion_matrix
from sklearn.metrics import classification_report
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import train_test_split
from sklearn.model_selection import RepeatedKFold
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import ParameterGrid
from sklearn.inspection import permutation_importance
import multiprocessing

# Grid de hiperparámetros evaluados
# ==============================================================
param_grid = {'n_estimators': [20, 30, 40, 50, 100, 150, 200],
              'max_features': ["sqrt"],
              'max_depth'   : [None, 3, 10, 20],
              'criterion'   : ['gini', 'entropy']
             }

# Búsqueda por grid search con validación cruzada
# ==============================================================
grid = GridSearchCV(
       estimator  = RandomForestClassifier(random_state = 123),
       param_grid = param_grid,
       scoring    = 'accuracy',
```

```
        n_jobs     = multiprocessing.cpu_count() - 1,
        cv         = RepeatedKFold(n_splits=5, n_repeats=3, random_state=123),
        refit      = True,
        verbose    = 0,
        return_train_score = True
       )

grid.fit(X = X_train, y = y_train)

# Resultados
# ==============================================================
resultados = pd.DataFrame(grid.cv_results_)
resultados.filter(regex = '(param*|mean_t|std_t)') \
    .drop(columns = 'params') \
    .sort_values('mean_test_score', ascending = False) \
    .head(4)
```

[ ]:
```
# Mejores hiperparámetros por validación cruzada
# ==============================================================
print("----------------------------------------")
print("Mejores hiperparámetros encontrados (cv)")
print("----------------------------------------")
print(grid.best_params_, ":", grid.best_score_, grid.scoring)
```

[ ]:
```
# Redefino parámetros de Random Forest
# ==============================================================
from sklearn.metrics import mean_squared_error

modelo_final = RandomForestClassifier(n_estimators = 150,
                                      criterion = "gini",
                                      max_features="sqrt",
                                      max_depth=10,
                                      bootstrap=True,
                                      random_state = 2016,
                                      min_samples_leaf = 8,
                                      max_samples = 2/3 ,
                                      oob_score=True)
```

[ ]:
```
modelo_final.fit(X_train, y_train)
```

[ ]:
```
print(modelo_final.score(X_test, y_test))
print(modelo_final.oob_score_)
```

[ ]:
```
y_pred = modelo_final.predict(X_test)
```

[ ]:
```
np.equal(y_pred,y_test.values)
```

[ ]:
```
from sklearn.metrics import confusion_matrix, classification_report
import seaborn as sn
cm = confusion_matrix(y_test, y_pred)
fig, ax = plt.subplots(figsize=(10,5))
ax.matshow(cm)
plt.title('Matriz de confusion', fontsize=20)
plt.ylabel('Etiqueta Verdadera', fontsize=15)
plt.xlabel('Etiqueta Predicha', fontsize=15)
sn.heatmap(cm, annot=True)
plt.show()

for (i, j), z in np.ndenumerate(cm):
    ax.text(j, i, '{:0.1f}'.format(z), ha='center', va='center')
```

[ ]:
```
#Guardar árboles de decisión en pdf
import matplotlib.pyplot as plt
plt.figure(figsize=(15,10))
from sklearn import tree

for arbol in modelo_final.estimators_:
    tree.plot_tree(arbol, feature_names=dataframe.columns[:-1])
    plt.savefig("arboles.pdf",
            dpi = 10000,
            bbox_inches ="tight",
            pad_inches = 1,
            transparent = True,
            facecolor ="w",
            edgecolor ='b',
            orientation ='landscape')
    plt.show()
```

[ ]:
```
import graphviz
from matplotlib import pyplot as plt
from sklearn import datasets
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree
# DOT data
for arbol in modelo_final.estimators_:
    dot_data = tree.export_graphviz(arbol, out_file=None,
                                    feature_names=dataframe.columns[:-1],
                                    filled=True)# Draw graph
    graph = graphviz.Source(dot_data, format="pdf")
    graph
    graph.render("decision_tree_graphivz")
```

[ ]:
```
from matplotlib import pyplot as plt
from sklearn.ensemble import RandomForestClassifier
from sklearn import tree
import numpy as np
from matplotlib.backends.backend_pdf import PdfPages
import pandas as pd
import matplotlib

samples=10
features = 20
names = ["f_"+str(feature) for feature in range(features)]

pdf = matplotlib.backends.backend_pdf.PdfPages("output.pdf")

for i, arbol in enumerate(modelo_final.estimators_):
    tree.plot_tree(arbol, feature_names=dataframe.columns[:-1])
    fig = plt.gcf()
    pdf.savefig(fig) #los guarda todos juntos
pdf.close()
```

[ ]:
```
# Datos originales, no considera oversampling
# División 75% de datos para entrenamiento, 25% de datos para test
X_train_2, X_test_2, y_train_2, y_test_2 = train_test_split(X, y,␣
 ↪random_state=0)
```

[ ]:
```
y_pred_original = modelo_final.predict(X_test_2)
```

[ ]:
```
np.equal(y_pred_original,y_test_2.values)
```

[ ]:
```
from sklearn.metrics import confusion_matrix, classification_report
import seaborn as sn
cm = confusion_matrix(y_test_2, y_pred_original)
fig, ax = plt.subplots(figsize=(10,5))
ax.matshow(cm)
plt.title('Matriz de confusion', fontsize=20)
plt.ylabel('Etiqueta Verdadera', fontsize=15)
plt.xlabel('Etiqueta Predicha', fontsize=15)
sn.heatmap(cm, annot=True)
plt.show()

for (i, j), z in np.ndenumerate(cm):
    ax.text(j, i, '{:0.1f}'.format(z), ha='center', va='center')
```

[ ]:
```
print(modelo_final.score(X_test_2, y_test_2))
print(modelo_final.oob_score_)
```