

**UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
SEDE VIÑA DEL MAR - JOSÉ MIGUEL CARRERA**

**PROPUESTA DE MODELO DE PREDICCIÓN DE FALLOS PARA COMPONENTE
CRÍTICO DE CAMIÓN MINERO, UTILIZANDO MACHINE LEARNING.**

Trabajo de Titulación para optar al Título
de Ingeniera en Mantenimiento Industrial
Licenciada en Ingeniería
Alumna:
Karem Solange Lastarria Monardes
Profesor Guía:
Mg. Ing. Andrés Eduardo Aránguiz Garrido

2024

DEDICATORIA

El cansancio, cada desvelo, e incluso los momentos en familia que quedaron en pausa, quedan solo inscritos en la memoria, este documento es el hito que cierra un ciclo y marca el comienzo de un nuevo camino.

Este es un logro compartido, como todos los que se han obtenido en mi vida, es un premio a la paciencia, comprensión y tolerancia de mi amado esposo Rubén y es el ejemplo de vida que les heredo a mis adorados hijos, Pablo, Javiera y Rubén.

Fue un sacrificio de toda una familia, que por años postergaron muchas cosas para que pudiese llegar a terminar este proceso, por ello, y por ser el pilar motivador del día a día, les doy gracias. No puedo dejar de mencionar a Dios, que me concedió la gracia de siempre iluminar mi sendero incluso cuando tempestades decidieron visitar en forma imprevista mi rutina.

Gracias también al cuerpo docente de mi alma mater que no solo se limitaron a transmitir conocimiento técnico, sino también experiencias de vida las cuales son invaluable.

Solo me queda mirar atrás con conformidad y satisfacción del camino recorrido y sentirme feliz y agradecida de las herramientas que me acompañaran en el futuro, ahora es mi responsabilidad saber utilizarlas para bien de mi familia y para ser un gran aporte en nuestra sociedad.

RESÚMEN

KEYWORDS: Machine Learning, Aprendizaje Supervisado, Mantenimiento Predictivo
Regresión logística, Máquina de Vectores de Soporte, Random Forest.
Fallas, Camión Minero de Extracción.

El desarrollo de este documento toma como base un conjunto de datos históricos de fallas de camiones de extracción minera, para realizar la propuesta de un modelo de Machine Learning.

Apunta a proponer un modelo de predicción de fallos para el componente crítico identificado, "Motor Diesel" de un camión minero, para ello se seleccionan modelos que son evaluados a través de métricas de calidad cuantitativas, permitiendo seleccionar aquel modelo que presenta una mejor evaluación.

Se emplean diferentes métodos para estimar la probabilidad de falla y el tiempo de vida útil del componente, tales como el modelo de vida Weibull, el modelo de Regresión Logística (RL), Máquina de Soporte de Vectores (SVM) y Random Forest (RF). Se comparan los resultados de cada método con los datos reales, y se evalúa el desempeño de cada uno mediante métricas estadísticas como el error cuadrático medio (RMSE), y la medida F1.

Las medidas de calidad obtenidas para los modelos de Clasificación (F1) fueron de 77,5% y 46,6% para Regresión Logística y Máquinas de Soporte de Vectores respectivamente; mientras que para los modelos de regresión la medida de calidad (RMSE) fue de 1519,351 y 76,915 para Weibull y Random Forest respectivamente

Se concluye que, para el caso de modelos de clasificación, el que presenta el menor error y la mayor precisión es el de Regresión Logística prediciendo falla del motor Diesel a las 4.200 horas de funcionamiento y en el caso de aplicación de modelo de regresión Random Forest, que logra captar la complejidad y la no linealidad de los datos. Sin embargo, es necesario precisar que es posible mejorar el rendimiento ante lo cual se sugiere realizar un nuevo entrenamiento con conjunto de datos menos dispares, de modo de mejorar las medidas de calidad.

ÍNDICE DE MATERIAS

DEDICATORIA 2

RESÚMEN 3

SIGLA Y SIMBOLOGÍA 7

OBJETIVO GENERAL 2

OBJETIVOS ESPECÍFICOS 2

1. ¿CÓMO LOGRAR EL KNOW-HOW? 4

1.1. DESCRIPCIÓN DE LA PROBLEMÁTICA Y contexto OPERACIONAL. 4

1.2. DESCRIPCIÓN GENERAL DEL EQUIPO en análisis 4

1.3. CONSISTENCIA DE LA DATA 6

1.4. IDENTIFICACIÓN DEL EQUIPO CRÍTICO 7

1.5. DESCRIPCION GENERAL DE LOS MODO DE FALLA DEL COMPONENTE CRÍTICO 8

1.6. COSTOS POR INEFICIENCIA 9

1.7. PLANTEAMIENTO DE LA PROPUESTA DE SOLUCIÓN 11

2. CONCEPTUALIZACIÓN de inteligencia artificial 13

2.1. DESCRIPCIÓN DE LAS PRINCIPALES TÉCNICAS 13

2.1.1. Estructura de la Inteligencia Artificial 13

2.1.2. Inteligencia Artificial Predictiva o Machine Learning 14

2.2. SELECCIÓN DE TECNICA DE MACHINE LEARNING PARA LA PREDICCIÓN DE FALLOS 15

2.2.1. Antecedentes bases 16

2.3. DESCRIPCIÓN DE LOS MODELOS SELECCIONADOS 17

2.3.1. Regresión Logística 17

2.3.2. Máquinas de vectores de soporte 21

2.3.3. Random Forest 24

2.4. ANÁLISIS DE CALIDAD 25

2.4.1. Error Medio Absoluto 26

2.4.2. Error medio cuadrático 26

2.4.3. Raíz Error medio cuadrático 26

2.4.4. Coeficiente de determinación R^2 26

2.4.5. Matriz de confusión 27

3. ESTRUCTURA DE MODELAMIENTO Y PROCESAMIENTO DE DATOS 31

3.1. TRATAMIENTO DE LA DATA 31

3.1.1. Categorización de datos de entrenamiento 31

3.1.2. División de datos para entrenamiento, validación y test 31

3.1.3. Programación de algoritmo con los datos de entrenamiento 31

3.1.4. Programación de algoritmo de predicción 31

3.1.5. Test de algoritmo de predicción. 31

3.1.6. Análisis de resultados 32

3.2. CÁLCULO BASE CON WEIBULL 32

3.2.1. Importación de datos 32

3.2.2. Estimación de Parámetros Dist.Weibull 32

3.2.3. Creación de números aleatorios y cálculo de RMSE..... 34

3.2.4. Análisis Exploratorio de Datos..... 35

3.2.5. Determinación de errores..... 36

3.2.6. Interpretación de los resultados..... 36

3.3. MODELO REGRESIÓN LOGÍSTICA..... 36

3.3.1. Importación de datos 36

3.3.2. Exploración gráfica 37

3.3.3. Entrenamiento y ajuste del modelo 37

3.3.4. Predicción del modelo 38

3.3.5. Representación gráfica del modelo. 39

3.3.6. Medidas de Calidad. 40

3.4. MODELO MÁQUINA DE VECTORES SOPORTES 40

3.4.1. Importación de datos 40

3.4.2. Exploración gráfica 40

3.4.3. Entrenamiento y ajuste del modelo 41

3.4.4. Predicción del modelo y representación gráfica 42

3.4.5. Medidas de Calidad. 42

3.5. MODELO RANDOM FOREST 42

3.5.1. Importación de datos 42

3.5.2. Entrenamiento y ajuste del modelo 43

3.5.3. Predicción del modelo y representación gráfica 43

3.5.4. Medidas de Calidad. 44

3.6. CUADRO COMPARATIVO DE MODELOS..... 45

CONCLUSIONES 46

BIBLIOGRAFÍA 48

ANEXO 1 49

ANEXO 2 53

ANEXO 3 77

ÍNDICE DE FIGURAS

Figura 1-1: Camión de extracción, modelo 930E. 5

Figura 1-2: Jack Knife Componente CAEX 930E3 7

Figura 1-3: Jack Knife CAEX 930E3 8

Figura 2-1 Mapa conceptual de Machine Learning 15

Figura 2-2 Ranking de uso de Modelos Machine Learning 17

Figura 2-3 Dispersión de Datos binario 18

Figura 2-4 Función Sigmoidal 19

Figura 2-5: Representación de ecuaciones de Coste. 19

Figura 2-6: Predicción mediante iteraciones Modelo Regresión Logística..... 20

Figura 2-7: Esquematización de datos en el modelo SVM.	21
Figura 2-8: Enfoque Hard SVM.	23
Figura 2-9: Enfoque Soft SVM.	23
Figura 2-10: Árbol de decisión de 3 niveles	24
Figura 2-11: Relación entre exactitud y precisión	29
Figura 3-1: Exportación de datos	32
Figura 3-2: Cálculo de parámetros Distribución Weibull	33
Figura 3-3: Gráfico de dispersión de Weibull.....	33
Figura 3-4: Tasa de Fallas y confiabilidad de datos reales.....	34
Figura 3-5: y_predictivo de datos aleatorios.....	34
Figura 3-6: Histograma de Frecuencia Absoluta y TEF	35
Figura 3-7: Comparativa de datos reales versus predichos	35
Figura 3-8: Medidas de Calidad	36
Figura 3-9: Gráfico de Violín para cada clase	37
Figura 3-10: Resultados de optimización del modelo	38
Figura 3-11: Predicción de probabilidades	38
Figura 3-12: Predicción de probabilidades	39
Figura 3-13: Predicción de probabilidades de fallo en primer y segundo periodo ...	39
Figura 3-14: Medidas de Calidad Modelo RL	40
Figura 3-15: Gráfico de distribución de fallas de componente crítico.....	41
Figura 3-16: Medidas de Calidad Modelo SVM.....	42
Figura 3-17: Evolución del error versus número de árboles RF.....	43
Figura 3-18: Distribución de datos de test	44
Figura 3-19: Predicción de falla de Componente crítico con RF.....	44
Figura 3-19: Medidas de Calidad de Modelo RF	45

ÍNDICE DE TABLAS

Tabla 1-1: Indicadores de flota de Camiones modelo 930E3	8
Tabla 1-2: Modos de Fallos indicados en data de CAEX 930E3.....	9
Tabla 1-3: Estimación de costos por ineficiencia de CAEX 930E3	11
Tabla 2-1: Matriz de confusión.....	27
Tabla 2-2: Interpretación de Medidas de Calidad.....	28
Tabla 3-1: Cuadro comparativo de medidas de calidad RF y Weibull	45
Tabla 3-2: Cuadro comparativo de medidas de calidad RL y SMV	45

SIGLA Y SIMBOLOGÍA

SIGLA

CAEX	:	Camión de extracción
USD	:	dólar americano
Null	:	Nulo
CIF	:	Costos por Ineficiencia
HH	:	Hora Hombre
IEEE Xplore	:	Institute of Electrical and Electronics Engineers

SIMBOLOGÍA

Σ	:	Sumatoria
∂	:	Derivada parcial
%	:	Porcentaje
Cost	:	Coste
$\geq$	:	Mayor que
$\approx$	:	Equivalente

ECUACIONES

Ecuación 1: $CIF = FF * TPFS * CSHA + CP + CC \text{ USD/año}$

Ecuación 2: $yx = b + wx1 + wx2$

Ecuación 3: $cost_{hx,y} = -ylog_{hx} - 1 - ylog(1 - hx)$

Ecuación 4: $\partial J(\theta) / \partial \theta_j = 1/n \sum_i (h_{xi} - y_i) * x_j$

Ecuación 5: $wx + b \geq 1 \rightarrow x \in class1$

Ecuación 6: $wx + b \leq -1 \rightarrow x \in class2$

Ecuación 7: $wx + b \geq 1 \rightarrow y = 1$

Ecuación 8: $wx^2 + b = 1$

Ecuación 9: $wx1 + b = -1$

Ecuación 10: $margin = x2 - x1 = 2w$

Ecuación 11: $max_2 w \approx min_2 w$

Ecuación 12: $loss = min_2 w^2$, sujeto a $ywx + b > 0$

Ecuación 13: $loss = min_2 w^2 + C \sum_i \xi_i$, sujeto a $y_i w x_i + b \geq 1 - \xi_i, \xi_i \geq 0$

Ecuación 14: $MAEX,y = 1/N \sum_i (1 - p(X_i))^2$

Ecuación 15: $MSEX,y = 1/N \sum_i (1 - p(X_i))^2$

Ecuación 16: $RMSEX,y = 1/N \sum_i (1 - p(X_i))^2$

Ecuación 17: $R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$

Ecuación 20: $Precisión = \frac{TP}{TP + FP}$

Ecuación 21: $Recall = \frac{TP}{TP + FN}$

Ecuación 22: $F1 = 2 \frac{Precisión * Calidad}{Precisión + Calidad}$

Ecuación 23: $Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$

Ecuación 24: $TPR = \frac{TP}{TP + FN}$

Ecuación 25: $FPR = \frac{FP}{FP + TN}$

INTRODUCCIÓN

Para una organización, el mantenimiento es considerado un gasto y existe el desafío continuo de minimizarlo sin afectar la confiabilidad de los equipos. De las políticas de mantenimiento existentes, aquella que destaca por los altos gastos en los que se incurren es el Mantenimiento Predictivo, dado que requiere de una fuerte inversión en análisis técnico, en la determinación de las variables a medir, en los sensores a implementar y el manejo de data en grandes volúmenes, que con herramientas tradicionales no es posible procesar.

Es en la gran industria, en donde el alto costo de sus activos y el impacto que puede provocar una falla en el ámbito económico, en la integridad de sus colaboradores, la seguridad operacional o deterioro en el medio ambiente, son las principales organizaciones que implementan las políticas de mantenimiento predictivo, dado que el costo versus beneficio que se genera con la capacidad de detectar fallas tempranas permite prevenir una falla total o parcial de los equipos, que puede llegar a ser catastrófica.

Por otra parte, existe un acelerado crecimiento de aplicaciones de Inteligencia Artificial (IA), en forma transversal, y que ha venido con fuerza ganando terreno en el mantenimiento predictivo, y la pregunta que se plantea es ¿cómo interactúan la Inteligencia Artificial y Mantenimiento predictivo?

Si se considera un equipo al cual se le desea predecir un fallo, observando su comportamiento, registrando un conjunto de datos de distintas variables (Temperatura, presión, vibraciones, etc.) además de los históricos de fallos en condiciones particulares de estas variables, se puede generar un algoritmo que realice un aprendizaje automático, o profundo o una red neuronal que a partir de los datos de entrada puedan obtener una salida, que es la predicción temprana de la falla, es así como estos últimos conceptos que son parte de la Inteligencia Artificial se cohesionan con el mantenimiento predictivo y va resultando interesante ahondar no solo en la teoría sino también en la aplicación de estos modelos en casos reales, más aún, con el crecimiento acelerado de tecnología en el tiempo actual y con la masificación de la puesta en práctica de técnicas de la Industria 4.0.

Todo lo anteriormente descrito sienta las bases para motivar un análisis y consiguiente modelación de algoritmos de IA en Mantenimiento Industrial, que es lo que se desarrollará en el presente documento.

OBJETIVOS

OBJETIVO GENERAL

Proponer modelo de Machine Learning, para la predicción de fallas de componentes críticos de camiones de extracción, en industria Minera, seleccionando la metodología que presente la mejor métrica de calidad.

OBJETIVOS ESPECÍFICOS

- Identificar equipo y componente críticos de un conjunto de Camiones de extracción minera, a través de metodología de Análisis de Regresión Logarítmica, bajo la problemática existente en su contexto operacional
- Contrarrestar técnicas de Machine Learning a través de análisis cualitativo, seleccionando aquella(s) que se aplique con mayor éxito en la predicción de fallos.
- Analizar los resultados de la aplicación de los modelos de Machine Learning sobre los históricos de falla, a través de métricas de evaluación cuantitativas, determinando el modelo que presenta una predicción más certera en el conjunto de datos.

**CAPÍTULO 1: PROBLEMÁTICA DE ESTUDIO E IDENTIFICACION DE COMPONENTE Y
EQUIPO CRÍTICO**

1. ¿CÓMO LOGRAR EL KNOW-HOW?

Para lograr los objetivos planteados, es necesario aplicar el "Know-How" o "Saber Hacer", y aplicar una metodología de trabajo. Se opta por la resolución de problemas a través de la Metodología de Portones (Front End Loaded, FEL), es decir proyectar un esquema escalonado de metas del conocimiento que permitirán apuntar a una solución técnicamente viable.

1.1. DESCRIPCIÓN DE LA PROBLEMÁTICA Y CONTEXTO OPERACIONAL.

La disponibilidad de los camiones de extracción es parte de un requerimiento de alto nivel de las empresas mineras, para ello éstas, externalizan el mantenimiento principalmente con sus vendedores, y las exigencias de operatividad de las flotas, exigen estrategias de mantenimiento que aseguren la confiabilidad de los equipos, para ello, se ha recurrido a análisis de fallas de causa raíz, mejoras en planes de mantenimiento, y otras herramientas que minimicen la indisponibilidad.

Un caso concreto es lo ocurrido en una compañía minera con su flota de CAEX, camiones eléctricos modelo 930E3 (8 unidades) y modelo 930E4 (46 unidades), en donde se ha detectado que uno de los mayores problemas es la indisponibilidad que se genera en sus componentes críticos, por eventos no previstos, la cual produce un incumplimiento de las metas planteadas de disponibilidad y pone en riesgo la continuidad operacional. Particularmente en el presente trabajo se realizará análisis sobre el modelo 930E3, dado que todo cálculo y modelamiento es posible en un futuro replicarlo con los datos del modelo 930E4.

Podemos indicar que el contexto operacional en el que desarrollan las actividades de los camiones en estudio (CAEX), corresponde al carguío de material de extracción hasta el sector de chancado primario, con camiones de alto tonelaje con capacidad de transporte de 280 toneladas y a lo más 320 toneladas cortas (290 toneladas) cumpliendo un 86% de las horas nominales de disponibilidad de los camiones de extracción (CAEX).

1.2. DESCRIPCIÓN GENERAL DEL EQUIPO EN ANÁLISIS

Los camiones eléctricos 930E3 de la marca Japones Komatsu, cuentan con una tolva fabricada con aceros anti-desgaste y de bajo peso que permite incrementar la cantidad de materia, transportando sin exceder el peso máximo

bruto permitido por fabrica 498,960 Ton. Poseen un motor Diesel de 16 cilindros que proveen una potencia neta al volante de 2550 Hp.

Respecto a la propulsión de estos equipos se puede mencionar que es a través de energía eléctrica proporcionada por un módulo de potencia, el cual está compuesto por un Motor Diesel, radiador, alternador principal y un soplador del sistema de propulsión.

El módulo de potencia, en el cual se encuentran en serie el Alternador principal y el Motor Diesel, proporciona el movimiento necesario al alternador principal y a su vez permite el funcionamiento del sistema eléctrico de alto voltaje.

Para el control de velocidad o desaceleración estos camiones cuentan con un sistema de retardo dinámico que permite disminuir la velocidad del camión de forma controlada.

El sistema hidráulico disponible para las aplicaciones de dirección, frenos y levante cuenta con un tanque de capacidad de 974 litros de aceite 10W y bombas que para el flujo necesario para cada aplicación.

Para mayor detalle ver Anexo 1 fichas técnicas de los CAEX 930E3.



Figura 1-1: Camión de extracción, modelo 930E.

Fuente: Specifications 930E-3 Electric Drive Truck.

1.3. CONSISTENCIA DE LA DATA

Se procede a realizar un análisis de la consistencia de data, de modo de asegurar que el procesamiento de esta no realice sesgos o presente error por la existencia de variables sin valor. Los datos son estructurados con etiquetas y reúnen la siguiente información

- a. En listado de equipos existen 27 etiquetados de los cuales solo se considerarán para efecto de análisis los siguientes parámetros:
 - i. Equipo (RMC)
 - ii. Número Interno
 - iii. Modelo
 - iv. Horómetro Inicio de Mes
 - v. Horómetro Fin de Mes
 - vi. Tiempo Operativo en horas
- b. Histórico de fallas de ambos modelos con 16 etiquetados, de los cuales se considerarán solo los siguientes:
 - i. Equipo
 - ii. Fecha Caída
 - iii. Horas
 - iv. Descripción
 - v. Tarea
 - vi. Código Componente
- c. Para el modelo MLP 930E-3, no se considerarán 24 registros de un total de 6.702 equivalente a un 0,36%, por corresponder a datos de Tiempo fuera de servicio (TFS) iguales a 0.
- d. Existe un tercer registro, que contiene la clasificación de sistemas y subsistemas que permitirá analizar Taxonomía según norma ISO 14224:2016.

En conclusión, se puede asegurar que la data es consistente, se eliminaron los valores Null, lo que permitirá alimentar sin inconvenientes los algoritmos a desarrollar. Para mayor detalle ver Anexo 2, el cual contiene data a procesar.

Camiones modelo 930E3	Nº de Fallas	TFS	MTTR
CA-52 A30432	46	309,8	6,73478261
CA-56 A30453	55	535,4	9,73454545
CA-57 A30458	45	619,6	13,76888889
CA-51 A30431	46	646,5	14,0543478
CA-53 A30433	72	846,3	11,7541667
CA-55 A30450	105	973,3	9,26952381
CA-54 A30435	92	1036,2	11,2630435
CA-50 A30398	95	1366,2	14,3810526

Tabla 1-1: Indicadores de flota de Camiones modelo 930E3

Fuente: Elaboración Propia en base a desarrollo en Excel

En Figura 1.3 se observa que el camión crítico es el CA-50 A30398, por lo cual todos los modelamientos se harán en base a dicho camión de extracción.

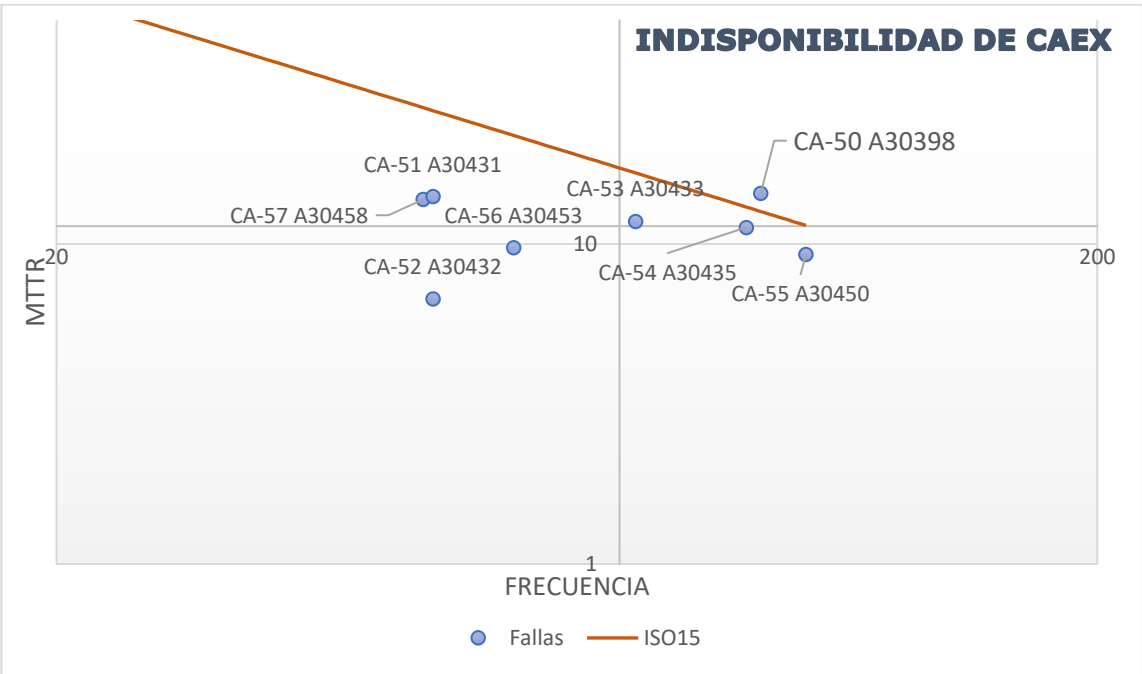


Figura 1-3: Jack Knife CAEX 930E3

Fuente: Elaboración Propia en base a desarrollo en Excel

1.5. DESCRIPCIÓN GENERAL DE LOS MODO DE FALLA DEL COMPONENTE CRÍTICO

De los antecedentes proporcionados, es necesario indicar que los registros no son totalmente categóricos con la identificación de los modos de falla. No existen

modos de fallas descritos correctamente. En Tabla 1-2 se identifican los enunciados con los que se categorizan los modos de fallas en registro entregado, es posible apreciar en misma tabla, por ejemplo, en la etiqueta denominada “Falla Motor Diesel” no describe un evento único que causa una falla potencial, (The Engineering Society For Advancing Mobility Land Sea Air and Space, 1999), por lo cual es un poco complejo identificar los principales modos de fallos, sin embargo, como no es uno de los objetivos de este trabajo ahondar en estos, solo se identifica este hallazgo para considerar mejoras futuras.

Etiquetas de fila	Nº de Fallos	TFS
Falla motor Diesel	58	644,1
Cambio módulo Diesel por condición	3	622,6
Luz roja motor	89	611,4
Baja potencia	71	438
Sin partida	21	307,7
Cambio módulo de potencia por condición	1	293,4
Luz roja motor // cambio módulo de potencia	1	270,3
Luz roja motor, cambio modulo Diesel por condición	1	264,1
Cambio de modulo Diesel (espera de traslado)	1	245,8
Falla módulo de potencia	1	213,3
Cambio de módulo Diesel	1	138,8
Cambio motor Diesel por condición	1	135,5
Aceite motor, cambio módulo Diesel por condición	1	128,7
Humo negro	3	83,6
Adv propulsión	1	77,5
Potencia	17	75,7
Ruido en motor Diesel	1	56,7
Termostato de refrigerante	1	51,8
Perdida de potencia	2	51,2
Propulsión	8	46,5
Luz roja motor Diesel	1	45

Tabla 1-2: Modos de Fallos indicados en data de CAEX 930E3

Fuente: Elaboración propia en base a data proporcionada.

1.6. **COSTOS POR INEFICIENCIA**

Se muestra interesante poder identificar cual es el costo asociado al no cumplimiento contractual de un mínimo de 86% de las horas nominales de disponibilidad de los camiones de extracción. Para poder calcular el costo por ineficiencia anualizado es necesario contar con información específica como se indica en ecuación 1.

Ecuación 1: $CIF = FF * TPFS * (CSHA + CP + CC)$ [USD/año]

- FF* = Frecuencia de Fallos [Fallos/año]
TPFS = Tiempo promedio Fuera de Servicio [horas/Fallo]
CSHA= Costo de penalización por Seguridad, Higiene y Ambiente [horas/Fallo]
CP = Costos de penalización por producción por hora [USD/hora]
CC = Costos de mantenimiento correctivo por hora [USD/hora]

Como no se disponen los datos de los Costos de penalización por seguridad, higiene y ambiente, ni los Costos de penalización por producción por hora ni los de mantenimiento correctivo, no es posible calcular el Costo de Ineficiencia por Fallos al año, pero si es posible aproximar este valor con los antecedentes propuestos en un estudio realizado con anterioridad, (Ismael, 2021), el cual indica un costo de 703 USD/mensual por Hora fuera de servicio por camión, éste no considera HH de mantenimiento correctivo.

Con los datos proporcionados es posible estimar el costo por camión el cual se desarrolla en tabla 2-3. Como resultado el Costo por ineficiencia del sistema de 8 camiones es de **USD 3.793.551 al año**, y considerando que el costo de inversión de un CAEX es de aproximadamente USD 5.000.000, el CIF calculado, representa el 79% de éste.

Además, se puede inferir que si se considerase la flota del modelo 930E4 cuyo número de camiones es de 45, este costo por ineficiencia estaría muy superior a los USD 5.000.000 al año.

EQUIPO	INDICADOR	UNIDAD	AÑO 2021
CA-50 A30398	TFS	horas/año	991,50
	TPO	horas	50,92
	FF	fallos/año	156,00
	TPFS	horas	6,36
	A		0,89
	CIF		USD 697.025
CA-51 A30431	TFS	horas/año	432,20
	TPO	horas	68,20
	FF	fallos/año	122,00
	TPFS	horas	3,54
	A		95,06
	CIF		USD 303.837
CA-52 A30432	TFS	horas/año	552,90
	TPO	horas	60,33
	FF	fallos/año	137,00
	TPFS	horas	4,04
	A		93,73
	CIF		USD 388.711
CA-53 A30433	TFS	horas/año	936,50
	TPO	horas	66,21
	FF	fallos/año	118,00

EQUIPO	INDICADOR	UNIDAD	AÑO 2021
	TPFS	horas	7,94
	A		89,30
	CIF		USD 658.360
CA-54 A30435	TFS	horas/año	648,00
	TPO	horas	56,74
	FF	fallos/año	140,00
	TPFS	horas	4,63
	A		92,46
	CIF		USD 455.544
CA-55 A30450	TFS	horas/año	514,60
	TPO	horas	48,53
	FF	fallos/año	169,00
	TPFS	horas	3,04
	A		94,10
	CIF		USD 361.764
CA-56 A30453	TFS	horas/año	847,10
	TPO	horas	60,12
	FF	fallos/año	131,00
	TPFS	horas	6,47
	A		90,29
	CIF		USD 595.511
CA-57 A30458	TFS	horas/año	473,40
	TPO	horas	62,53
	FF	fallos/año	132,00
	TPFS	horas	3,59
	A		94,58
	CIF		USD 332.800
CIF Total sistema		USD/año	USD 3.793.551

Tabla 1-3: Estimación de costos por ineficiencia de CAEX 930E3

Fuente: Elaboración propia en base a calculo en data

1.7. PLANTEAMIENTO DE LA PROPUESTA DE SOLUCIÓN

En la empresa minera en cuestión, se han desarrollado distintas estrategias de mantenimiento preventivo y correctivo, sin embargo, la propuesta a desarrollar consiste en presentar un modelo de predicción de fallos a través de herramientas tecnológicas y de información como lo es la Inteligencia Artificial. Este modelo generado será en base a entrenamiento con la data existente y será capaz de predecir posibles fallos en componentes críticos de los camiones mineros en función de nuevos datos de operación.

CAPÍTULO 2: ANÁLISIS CUALITATIVO DE TECNICAS DE INTELIGENCIA
ARTIFICIAL

2. CONCEPTUALIZACIÓN DE INTELIGENCIA ARTIFICIAL

La Inteligencia Artificial es una disciplina centrada en la creación de máquinas o software, que pueden tomar decisiones en base a la incertidumbre. Ha sido mal concebida como el remplazo de las funciones del humano, pero esto, al menos a mediano plazo, no es posible, ya que sólo puede llevar a cabo una tarea definida, pero si se ve enfrentada a un patrón que se escapa de los parámetros de entrenamiento, su tarea no puede ser ejecutada o entrega un resultado no coherente, es decir no entiende las causalidades, o conceptos abstractos, no puede comprender contextos de información, ni emocionalidad ni hacer juicios de valor.

Según la definición de la Comisión Europea, existen tres tipos de IA (Tebar, 2023):

- **IA débil:** es aquella que trata de imitar las habilidades humanas, principalmente asociado a reconocimiento de patrones o toma de decisiones
- **IA Fuerte:** ésta trata de imitar la inteligencia humana en su totalidad, incluyendo la capacidad de aprender y resolver problemas.
- **IA súper:** ésta busca superar la inteligencia humana, desarrollando sistemas más rápidos, precisos y más capaces que los humanos.

2.1. DESCRIPCIÓN DE LAS PRINCIPALES TÉCNICAS

De modo de entender la composición de la Inteligencia Artificial en los siguientes puntos se darán a conocer generalidades de éstas, destacando aquellas que tienen aplicación en el mantenimiento predictivo.

2.1.1. Estructura de la Inteligencia Artificial

En el contexto particular, la Inteligencia Artificial se divide en dos grandes técnicas de aplicación, la Inteligencia Artificial Generativa, que es aquella que crea contenidos informáticos de un modelo base, es decir, aprende patrones y estructuras de datos y crea nuevos ejemplos con características similares a los originales. Las técnicas utilizadas comúnmente son redes Neuronales Adversarias (GAN), o Autocodificadores variacionales generativas.

Por otra parte, se tiene a la Inteligencia Artificial Predictiva, es aquella que permiten crear pronósticos de eventos futuros en base a datos actuales y pasados, entrenando un modelo para generar resultados predictivos, ésta requiere de datos etiquetados con resultados conocidos para ser entrenados. Las técnicas utilizadas

corresponden a regresión análisis de series temporales, arboles de decisión o redes neuronales y con ellas se identifican patrones y luego realizan predicciones. (Miguel, 2023).

Ambos tipos de inteligencias artificiales tienen un punto de partida en común. los "Datos", los cuales pueden ser, imágenes, textos, audios, videos, también llamados "Datos no estructurados" o valores etiquetados, comúnmente tablas con información numérica y alfanumérica, que son conocidos como "Datos Estructurados". Es necesario complementar que la ciencia que extrae ideas a partir de los datos y se presenta con resultados y recomendaciones es conocida como "Data Science"; por lo tanto, se puede inferir que, para los fines del presente trabajo, el área de interés está en la Inteligencia Artificial Predictiva la cual se desarrolla a continuación en el punto 2.1.2

2.1.2. Inteligencia Artificial Predictiva o Machine Learning

Como se indicó en el punto 2.1.1, su foco se centra en respuestas predictivas alimentado de datos estructurados, se divide en tres, "Aprendizaje supervisado", "Aprendizaje de reforzamiento", y "Aprendizaje no supervisado" y; cada una de ellas posee varias las técnicas de aplicación y no todas presentan la utilidad que se requiere para el presente proyecto. (Jesus, 2020) A continuación, una breve descripción las herramientas que aplican:

a. Aprendizaje Supervisado.

Corresponde al procesamiento de datos estructurados con etiquetas y solución, realiza predicción en base a experiencia, para lo cual se parcela la data en un 2 o 3 considerando parte de ella para el aprendizaje y el restante para validación del modelo, alguno de sus modelos son:

- i. Regresión Lineal
- ii. K-vecinos más cercanos
- iii. Máquinas de vectores de soportes
- iv. Regresión Logística.

b. Redes Neuronales.

Corresponden a modelos que realizan el procesamiento de información e identificación de patrones. Está compuesto de varios elementos de computación simples conocidos como neuronas o nodos, los cuales se encuentran interconectados y organizados en capas operan de modo que cada neurona realiza la tarea de procesar información mediante la conversión de entradas recibidas en salidas procesadas, se mencionan los principales modelos:

- i. Redes Generativas Adversativas (GAN)
- ii. Redes Neuronales Recurrentes (RNN)
- iii. Perceptrón Multicapa

c. Métodos de ensamble.

Corresponde a una agrupación de modelos de Machine Learning. Cada modelo genera una predicción diferente. Las predicciones de los distintos modelos se combinan para obtener una única predicción. Entre los más usados se encuentran:

- i. Random Forest
- ii. Adaboost
- iii. Lightgbm

En figura 2-1, se puede observar gráficamente la estructura de los principales modelos de Machine Learning.

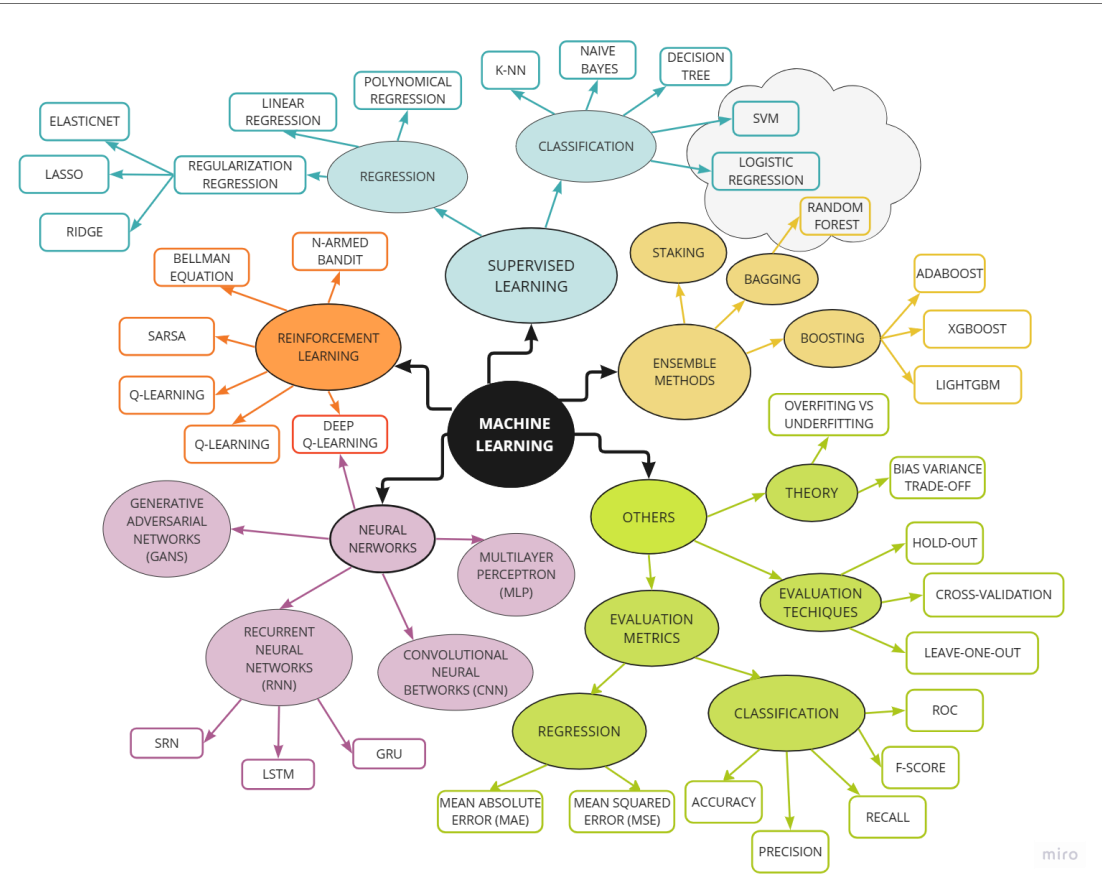


Figura 2-1 Mapa conceptual de Machine Learning

Fuente: <https://vitalflux.com/great-mind-maps-for-learning-machine-learning/>

2.2. SELECCIÓN DE TÉCNICA DE MACHINE LEARNING PARA LA PREDICCIÓN DE FALLOS.

En la presente sección se recurre literatura de publicaciones de especialistas para realizar un análisis cualitativo de los modelos a utilizar en el caso en estudio.

2.2.1. Antecedentes bases

Según la bibliografía de investigación, (Carvalho, 2019), se realizó una Revisión Sistemática de la Literatura (SLR), el cual permite identificar, evaluar e interpretar aspectos de investigación para un tema, área o fenómeno.

Esta investigación desarrolló 5 preguntas:

- Pregunta N°1: ¿Cuáles son los métodos de ML que se utilizan en Mantenimiento Predictivo?
- Pregunta N°2: ¿Qué equipos están siendo sometidos a técnicas de Mantenimiento Predictivo?
- Pregunta N°3: ¿Cuáles son los datos utilizados para aplicar Mantenimiento Predictivo?
- Pregunta N°4: ¿Los datos son reales o sintéticos?
- Pregunta N°5: ¿Cómo se emplean los métodos de Machine Learning en aplicaciones de Mantenimiento predictivo?

Para fines del presente acápite solo se enfocará la atención en la pregunta N°1.

El estudio en comento realizó la investigación en bases de datos con alcance científico tales como IEEE Xplore y ScienceDirect, realizando motores de búsqueda combinados como "mantenimiento predictivo" Y "Machine Learning o Machine Learning Technique", "mantenimiento predictivo" Y "Técnicas de aprendizaje automático", respectivamente, arrojando como resultado que los métodos más utilizados para realizar Mantenimiento Predictivo son Random Forest, Redes Neuronales, Máquinas de vectores de soporte y K- Vecinos más cercanos.

Adicionalmente (Baptista, y otros, 2017), comparan varias técnicas de Inteligencia Artificial contra técnicas estadísticas para predecir cuándo un equipo estará en riesgo de fracaso futuro, dando como resultado que el aprendizaje automático supera al enfoque estadístico.

Si bien, no se ha podido identificar en la literatura una jerarquización de las técnicas de Machine Learning aplicadas al mantenimiento predictivo, esto es principalmente porque dependerá de diversos factores, como la naturaleza de los datos, el contexto específico del sistema, la disponibilidad de datos históricos y la complejidad del equipo o sistema bajo consideración.

Todas las investigaciones apuntan siempre al mismo grupo de modelos utilizados en mantenimiento, por lo cual, considerando el porcentaje de utilización

las más recurrentes son coincidentes con los análisis tanto de Carvallo y Baptista y los cuales se ordenan según preferencia en la Figura 2-2.

Para los objetivos planteados en el presente trabajo se decide realizar una comparativa con las técnicas de Regresión logística, Random Forest y Máquina de vectores de soporte y no considerar Regresión lineal dado que esta última es la más simple de las técnicas de Machine Learning, la cual consiste en aproximar mediante la técnica de los mínimos cuadrados los valores de los ensayos realizados.

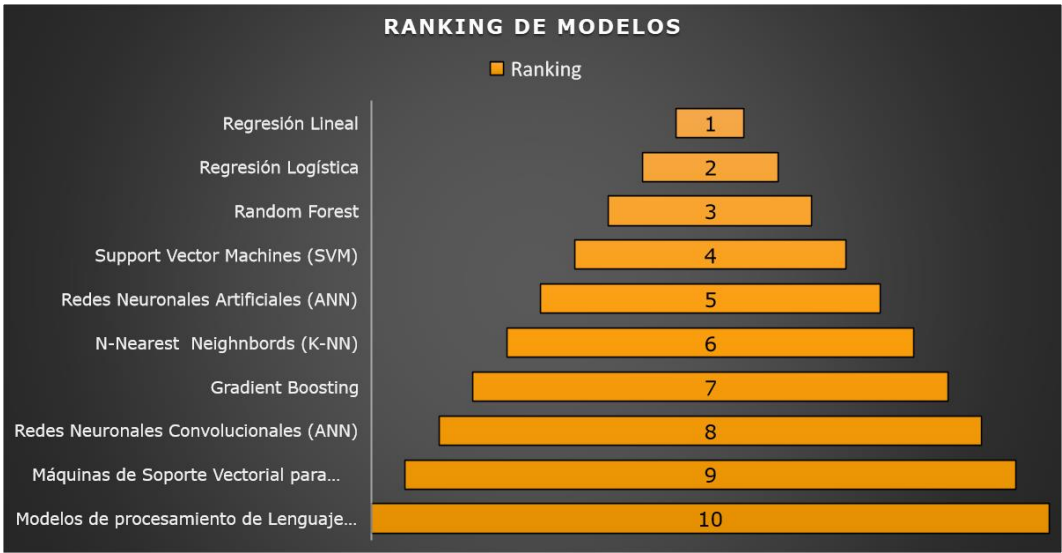


Figura 2-2 Ranking de uso de Modelos Machine Learning

Fuente: Elaboración propia en base a literatura analizada y motores de búsqueda IA

2.3. DESCRIPCIÓN DE LOS MODELOS SELECCIONADOS

En los siguientes puntos se desarrolla conceptualmente cada uno de los modelos a utilizar

2.3.1. Regresión Logística

La Logistic Regression (LR), es utilizada por lo general para casos de clasificación, el modelado y la analítica predictiva, con más de una dimensión encuentra las relaciones entre dos factores de datos. luego, utiliza esta relación para predecir el valor de uno de esos factores basándose en el otro. Normalmente, la predicción tiene un número finito de resultados, como un sí o un no sobre un grupo objetivo de datos el cual puede ser binarios o multiclase.

Para entender cómo se desarrolla este modelo, se considera un conjunto de datos de la forma:

Ecuación 2: $y(x) = b + wx_1 + wx_2$

En donde:

x_1 se toman de la característica 1,

x_2 corresponden a la característica 2

W corresponde a la pendiente

b offset (distancia vertical al eje)

si este conjunto de datos se grafica se verá cómo se indica en figura 2-3.

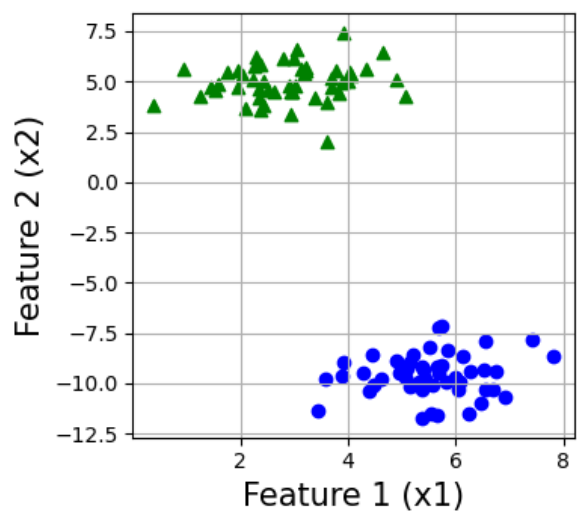


Figura 2-3 Dispersión de Datos binario

Fuente: Elaboración propia Python en base a código Machine Learning y Deep Learning, Jesús Bobadilla.

La regresión logística genera una etiqueta de clase en donde $h(x)=0$ corresponde a la primera clase identificada por puntos y $h(x)=1$ es la segunda clase asociada a los triángulos. Para convertir los valores “y” lineales a binarios “h(x)”, la regresión logística utiliza una función del tipo sigmoideal de la forma $\frac{1}{(1+e^{-t})}$, que se comporta no linealmente asociando el valor de 0 cuando su entrada es menor que 0 y devuelve +1 cuando su entrada es mayor que 0, tal como la se grafica en Figura 2-4.

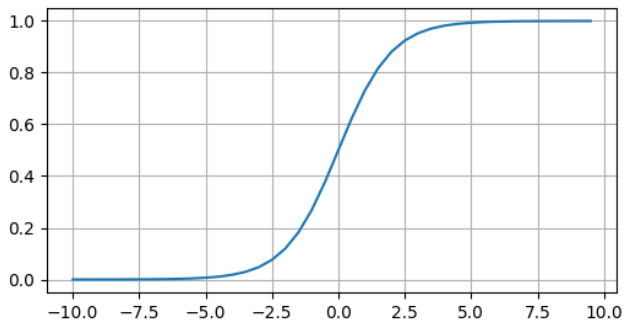


Figura 2-4 Función Sigmoidal

Fuente: Elaboración propia Python en base a código Machine Learning y Deep Learning, Jesús Bobadilla.

El fundamento de predicción de este modelo se basa en la definición de una ecuación derivable de coste (ecuación 3) la cual busca minimizar los errores entre las entradas "x" y las salidas "y".

Ecuación 3: $cost(h(x), y) = -y \log(h(x)) - (1 - y) \log(1 - h(x))$

Se puede observar que en la ecuación antes mencionada cuando y=1 toma el valor de $-\log(h(x))$ y mientras $h(x) = 1$ o cercano a él, el coste será bajo por lo cual se premia la predicción correcta del objetivo $y=1$. En el caso contrario si $y=0$ lo función coste queda equivalente a $-\log(1-h(x))$, y similar al análisis anterior con $h(x)=0$ o cercano, se premia la predicción del objetivo $y=0$, ver figura 2-5.

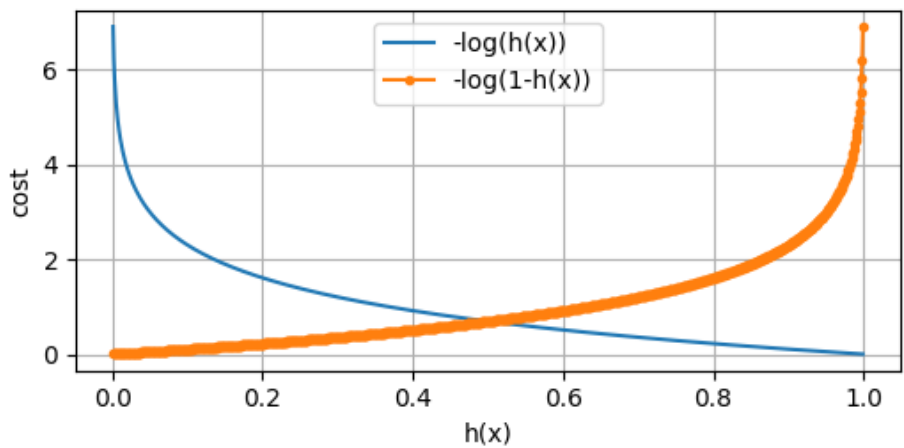


Figura 2-5: Representación de ecuaciones de Coste.

Fuente: Elaboración propia Python en base a código Machine Learning y Deep Learning, Jesús Bobadilla.

El modelo trabaja con la función gradiente descendente (ecuación 4), correspondiente a la derivada parcial de la forma discreta de la ecuación (3), codificándola y devolviendo los valores de θ después de procesar una iteración (epoch).

Ecuación 4: $\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{n} \sum_{i=1}^n [(h(x_i) - y_i) * x_j]$

Para ajustar el modelo es necesario realizar varias iteraciones de la función hasta que el parámetro θ no presente cambios significativos o hasta un número determinado de iteraciones, ver Figura 2-6. Si se desarrolla este principio en los datos del ejemplo, es posible observar lo siguiente:

- a. Si no se ajusta el modelo (0 epochs) la mitad de la muestra se etiqueta erróneamente.
- b. Ajustando el modelo con 2 iteraciones la cantidad de etiquetas erróneas se reduce en la mitad de los obtenidos con 0 iteraciones.
- c. Usando sólo 3 iteraciones todas las muestras se etiquetan correctamente.

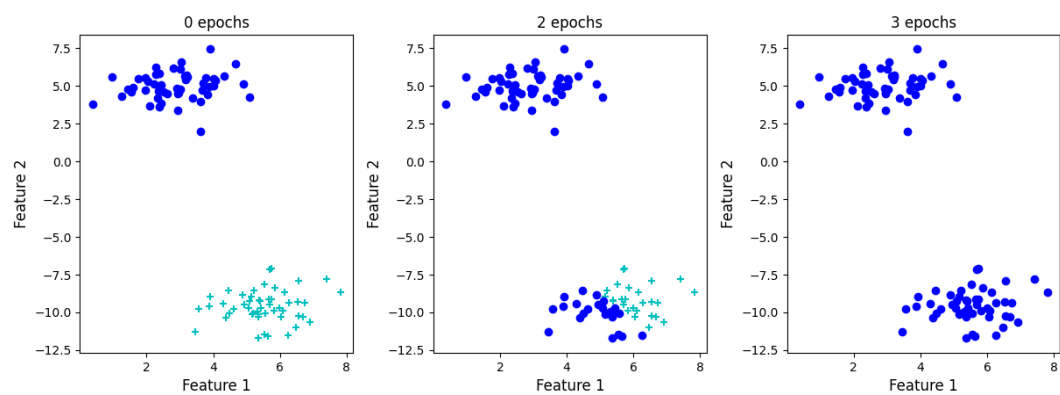


Figura 2-6: Predicción mediante iteraciones Modelo Regresión Logística.

Fuente: Elaboración propia Python en base a código Machine Learning y Deep Learning, Jesús Bobadilla.

El ejemplo antes desarrollado, es bajo un escenario de 2 características, es decir, $n_features = 2$, pero tambien es aplicable para el caso de $n_features = 3$, sin embargo, en este caso no se detallará el modelamiento en 3 dimensiones.

En lo que respecta a la aplicación en mantenimiento predictivo, existe un estudio en el cual, en la utilización de este modelo, se dieron muy buenos

resultados en la predicción de fallos en maquinaria Industrial, pero con un número considerable de falsos positivos. (Battifarano, DeSmet, Madabhushi, & Nabar, 2018)

2.3.2. Máquinas de vectores de soporte

También es conocido como SVM (Support Vector Machines), es un método que consiste en la representación de los datos en un espacio cuya dimensión la define el número de variables, encuentra el hiperplano que maximiza el margen entre clases. En este modelo las muestras o vectores definen el hiperplano (decisión boundary) son los llamado vectores de soporte. En Figura 2-7 a) se observa dos clases separadas en forma linea, existen infinitas líneas que pasan por ambos grupos, pero se desea encontrar el mejor plano de separación. En la Figura b) muestra el hiperplano más conveniente ya que maximiza la anchura del margen entre las fronteras o vectores de soportes encontrados, en otras palabras SVM es la que busca estas fronteras.

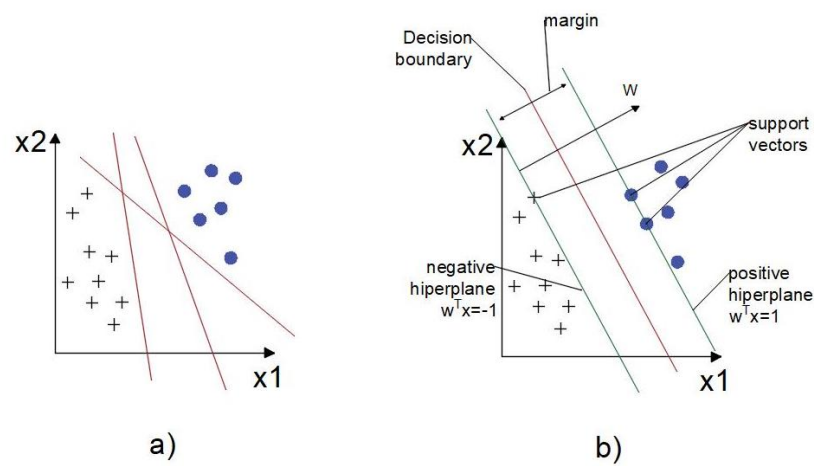


Figura 2-7: Esquematización de datos en el modelo SVM.

Fuente: Elaboración propia CAD en base a Machine Learning y Deep Learning, Jesús Bobadilla.

El principio de este modelo es muy similar al de Regresión logística, en donde las muestras ubicadas a un lado de la frontera de decisión son clasificadas con valores positivos iguales o mayores a 1, en el caso contrario las que quedasen al lado opuesto son clasificadas con valores negativos iguales o menores que 1. Matemáticamente hablando lo anterior se expresa de la siguiente forma

Ecuación 5: $w x + b \geq 1 \rightarrow x \in class1$

Ecuación 6: $w x + b \leq -1 \rightarrow x \in class2$

$$\text{Ecuación 7: } wx + b \geq 1 \rightarrow y = 1$$

Adicionalmente los hiperplanos Positivos y Negativos quedan definidos por la ecuación 8 y 9 respectivamente

$$\text{Ecuación 8: } wx_2 + b = 1$$

$$\text{Ecuación 9: } wx_1 + b = -1$$

Para calcular el margen es la resta entre ecuación 8 y 9 que da como resultado

$$\text{Ecuación 10: } \text{margin} = x_2 - x_1 = \frac{2}{\|w\|}$$

En el modelo SVM se busca maximizar el margen, lo cual es equivalente a lo indicado en la ecuación 11.

$$\text{Ecuación 11: } \max\left(\frac{2}{\|w\|}\right) \approx \min\left(\frac{1}{2} \|w\|^2\right)$$

Esta última expresión se transforma en la ecuación de coste denominada también Hard SVM, porque no permite muestras no clasificadas (ecuación 12).

$$\text{Ecuación 12: } \text{loss} = \min\left(\frac{1}{2} \|w\|^2\right), \text{ sujeto a } y(wx + b) > 0$$

Para levantar la restricción antes descrita se suele utilizar el parámetro "slack" ξ_i , que permite trabajar con datos no clasificados y valores aislados, pasando de un enfoque Hard a Soft SVM quedando la ecuación de costo formulada de la siguiente forma:

$$\text{Ecuación 13: } \text{loss} = \min\left(\frac{1}{2} \|w\|^2\right) + C \sum_i \xi_i, \text{ sujeto a } y_i(wx_i + b) \geq 1 - \xi_i, \xi_i \geq 0$$

El parámetro C permite mantener el equilibrio entre el ancho del margen y las clasificaciones erróneas, es decir, cuando C toma valores pequeños el modelo se regulariza (se hace mas soft).

En la Figura 2-8 muestra 3 vectores de soporte y su hiperplano bajo el enfoque Hard SVM.

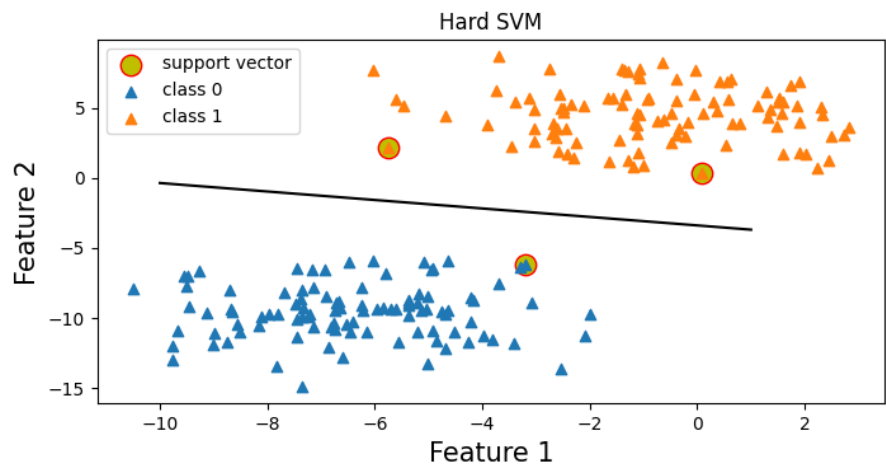


Figura 2-8: Enfoque Hard SVM.

Fuente: Elaboración propia Python en base a código Machine Learning y Deep Learning, Jesús Bobadilla.

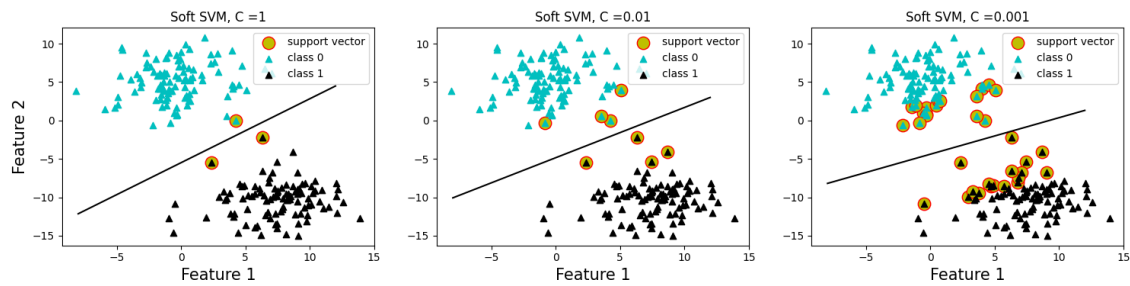


Figura 2-9: Enfoque Soft SVM.

Fuente: Elaboración propia Python en base a código Machine Learning y Deep Learning, Jesús Bobadilla.

Por otra parte, en la figura 2-9 se observa que mientras más se minimiza el parámetro C el modelo va aumentando el margen y por ende incrementa también el número de vectores de soporte, dejando mejor identificadas las regiones que definen la predicción.

Este método fue utilizado en la Detección de fallas incipientes en rodamientos de Generadores Sincrónicos en donde los resultados obtenidos fueron superiores a 85% para clasificación multiclase y superiores a 96% para clasificación binaria. (Sparano, Ramírez, Guerra, & Terán , 2023).

2.3.3. Random Forest

Este método es del tipo ensamble basado en los árboles de decisión sobre un conjunto de datos elegidos al azar.

Antes de seguir desarrollando el concepto de Random Forest se dará a entender que es un árbol de decisión y para ello se desarrolla un ejemplo con una base de datos, generando el antes mencionado con una restricción de 3 niveles. En figura 2-10 se puede observar que en el nodo raíz que indica un tamaño de muestra de 569 unidades; en donde el parámetro de consulta es la condición “si es menor a 16.795” se sigue el camino de la izquierda y en caso contrario hacia la derecha; cada casilla muestra la cantidad de elementos que conforman los tipos de clase y la intensidad del color es proporcional a la probabilidad de la clase objetivo y cada nivel va agrupando las clases según el parámetro de consulta.

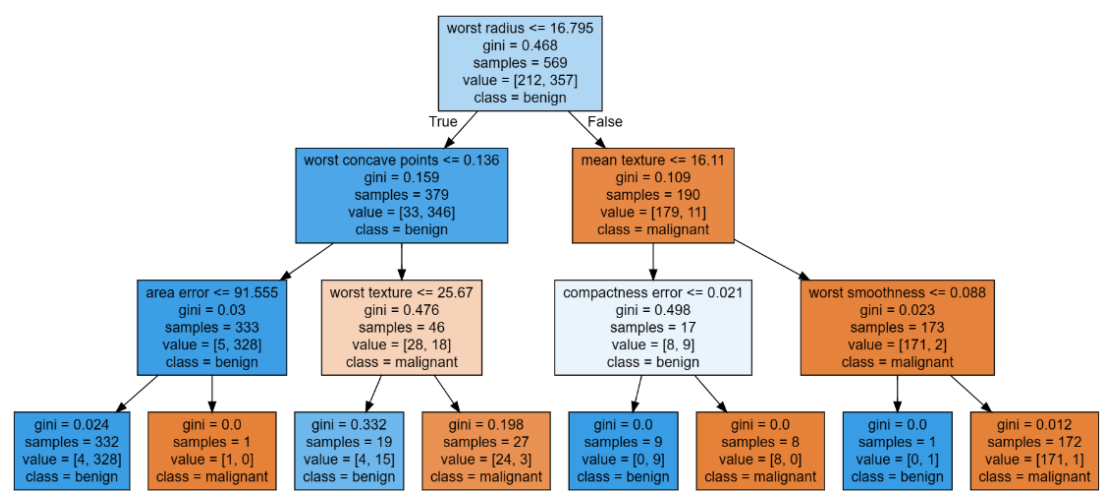


Figura 2-10: Árbol de decisión de 3 niveles

Fuente: Elaboración propia Graphiiz Online en base a código Machine Learning y Deep Learning, Jesús Bobadilla.

El enfoque de Random Forest (RF) es distinto, es decir, en lugar de combinar varios algoritmos de Machine Learning para mejorar la exactitud, utiliza el mismo algoritmo, pero en distintos subconjuntos aleatorios de datos, es decir utiliza varios árboles de decisión y cada uno de estos árboles se desarrolla con un conjunto de datos aleatorios de entrenamiento.

Random Forest posee tres hiperparámetros principales, que deben configurarse antes del entrenamiento estos son, tamaño del nodo, cantidad de árboles y cantidad de características muestreadas.

Por otra parte, del total de la muestra se recomienda que de la muestra de entrenamiento, un tercio se reserve como datos de prueba. Luego, se genera una nueva aleatoriedad a los subconjuntos a través del agrupamiento de características, con esto se logra más diversidad al conjunto de datos y reduce la correlación entre los árboles de decisión y de esta forma se evita el overfitting (sobreajuste). Random Forest funciona tanto como método de clasificación como de regresión en donde la predicción estará dada por variable categórica más frecuente, arrojará la clase predicha en el caso de clasificación, mientras que para el de regresión se promediarán los árboles de decisión individuales,

Finalmente, la muestra reservada de prueba se utiliza para la validación cruzada, finalizando esa predicción.

La desventaja de este modelo es principalmente la aleatoriedad de los árboles que se pueden generar ya que se produce cierta contradicción entre los resultados obtenidos sobre un mismo conjunto de datos. Existe antecedentes en su desarrollo en la predicción de fallos en sistemas de refrigeración con un acierto del 89%.

2.4. ANÁLISIS DE CALIDAD

Es necesario identificar y evaluar exactitud de los resultados, indistintamente del modelo utilizado. Machine Learning emplea diferentes medidas de calidad, algunos son aplicables a los resultados de clasificación y otros a los de regresión, sin embargo, hay que tener en cuenta que no se debe medir la calidad con el mismo conjunto de datos empleados para el entrenamiento.

En el modelo se debe evitar situaciones en la que aprenda del conjunto de datos de entrenamiento, pero sea incapaz de tratar datos nuevos para su predicción, esto es lo que se conoce como sobre ajuste overfitting

Se debe dividir los datos y los valores objetivos en varios conjuntos tales como **entrenamiento, validación y test** el conjunto de los datos de entrenamiento es el de mayor volumen, por lo general corresponde al 80% del total de la muestra y es esta la que se utiliza para entrenar el modelo; por otra parte el conjunto de validación es aquel que contiene las muestras utilizadas para mejorar el modelo y realizar un ajuste final de los hyperparámetros, es decir, de los valores que controlan diferentes variaciones en el funcionamiento de los algoritmos; y por último tenemos el conjunto de test o prueba que es aquel que permitirá medir la calidad del modelo utilizando las muestras que no hayan sido utilizadas para

entrenar o mejorar el modelo, por lo tanto, se debe asegurar que los 3 conjuntos tienen una distribución similar de datos es decir que son datos representativos.

A continuación, se indican las medidas de calidad más utilizadas para los modelos de regresión:

2.4.1. Error Medio Absoluto

También conocido como MAE (Mean Absolute Error), corresponde a una medida de calidad que penaliza la distancia entre el valor objetivo y cada predicción

$$\text{Ecuación 14: } MAE(X, y) = \frac{1}{N} \sum_{i=1}^N (\hat{y} - p(X_i))^2$$

2.4.2. Error medio cuadrático

Corresponde al MSE (Mean Squared Error), y tiene el mismo principio que el MAE con la diferencia que éste es más severo al elevarlo al cuadrado.

$$\text{Ecuación 15: } MSE(X, y) = \frac{1}{N} \sum_{i=1}^N (\hat{y} - p(X_i))^2$$

2.4.3. Raíz Error medio cuadrático

Corresponde al RMSE (Root Mean Squared Error), corresponde a la raíz cuadrada del MSE.

$$\text{Ecuación 16: } RMSE(X, y) = \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{y} - p(X_i))^2}$$

2.4.4. Coeficiente de determinación R^2

Esta medida de calidad a diferencia del MAE y MSE es del tipo relativo, en donde su valor está entre 0 y 1, cuando es 1 indica que el modelo tiene una predicción perfecta, y si llega a ser 0 es una regresión simple.

$$\text{Ecuación 17: } R^2 = 1 - \frac{\sum_{i=0}^{n-1} (y_i - f(x_i))^2}{\sum_{i=0}^{n-1} (y_i - \bar{y})^2}$$

Por otra parte, los algoritmos de clasificación también generan predicciones y las medidas de calidad correspondientes son:

2.4.5. Matriz de confusión

Medida de calidad que corresponde a una clasificación binaria entre aciertos y no aciertos en la predicción.

En tabla 2.1 en cuadrante TP se indican los aciertos o predicciones correctas, ídem en el cuadrante TN, pero asociados al valor objetivo contrario, mientras que, en el cuadrante FN y FP, estos últimos corresponden a los errores de la predicción. 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.

Predicción	Positivos en clase 0	Negativos en clase 1
Clase 1	TP Predicho Verdadero y Verdadera en realidad	FN Predicción de falso y verdadero en la realidad
Clase 0	FP Predicción de verdadero y falso en la realidad	TN Predicho Falsa y Falsa en realidad

Tabla 2-1: Matriz de confusión

Fuente: Elaboración propia en base a resumen de punto 2.4.5

2.4.5.1. Precisión

Corresponde a la medida de calidad que relaciona los valores de predicción de la clase objetivo, es decir, cuantas predicciones son correctas, es decir, se refiere a la dispersión del conjunto de valores obtenidos a partir de mediciones repetidas de una magnitud. Cuanto menor es la dispersión mayor la precisión y corresponde a la siguiente proporción:

Ecuación 18: Precisión $= \frac{TP}{TP+FP}$

2.4.5.2. Calidad Recall

Esta medida de calidad tambien llamada sensibilidad, como lo indica su nombre se enfoca en el número relativo de aciertos positivos, es decir, cuantas predicciones son correctas en una clase en relación con el total de muestras positivas.

Ecuación 19: Recall $= \frac{TP}{TP+FN}$

Es importante mencionar que se recomienda que cuando el número de predicciones aumenta el valor de la precisión debe ser mayor; por otra parte, cuando el número de muestras positivas es bajo el recall debe ser mayor.

2.4.5.3. F1

Corresponde a la medida de calidad que relaciona una Precisión y Recall. Es de gran utilidad cuando la distribución de las clases es desigual.

Ecuación 20: $F1 = 2 \frac{Precisión * Calidad}{Precisión + Calidad}$

2.4.5.4. Calidad de exactitud

Medida utilizada en el caso que todas las clases sean iguales y se refiere a lo cerca que está el resultado de una medición del valor verdadero. En términos estadísticos, la exactitud está relacionada con el sesgo de una estimación.

Ecuación 21: $Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$

2.4.5.5. ROC

Medida de calidad que relaciona los valores positivos con los falsos positivos

Ecuación 22: $TPR = \frac{TP}{TP + FN}$

Ecuación 23: $FPR = \frac{FP}{FP + TN}$

En tabla resumen 2-2 se desarrolla la interpretación de las métricas antes descritas:

Precisión	Recall	Interpretación
Alta	Alto	Modelo de Machine Learning escogido se maneja perfectamente esa clase
Alta	Bajo	Modelo de Machine Learning escogido no detecta la clase muy bien, pero cuando lo hace es altamente confiable
Baja	Alto	El modelo de Machine Learning escogido detecta bien la clase, pero también incluye muestras de la otra clase
Baja	Bajo	El modelo de Machine Learning escogido no logra clasificar la clase correctamente

Tabla 2-2: Interpretación de Medidas de Calidad.

Fuente: Elaboración propia en base a <https://www.juanbarrios.com/la-matriz-de-confusion-y-sus-metricas/>

En figura 2-11 se grafica como es la relación entre exactitud y precisión

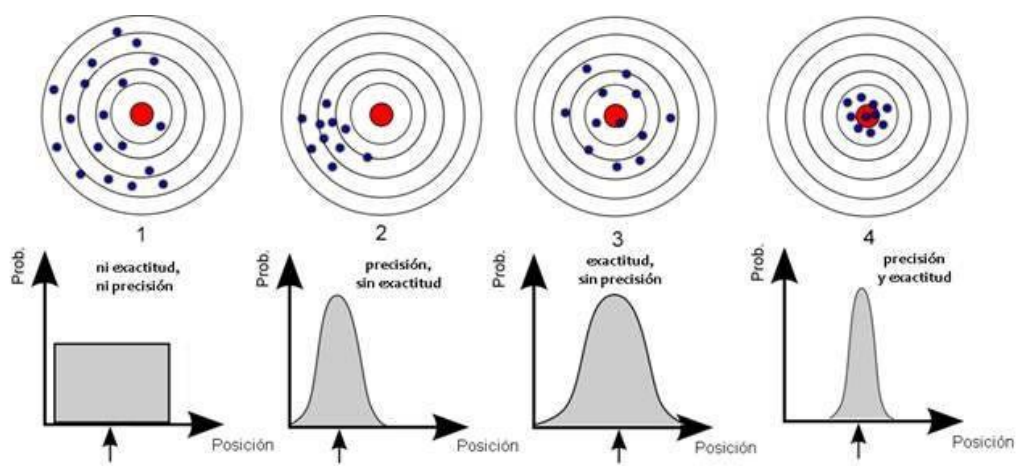


Figura 2-11: Relación entre exactitud y precisión

<https://www.juanbarrios.com/la-matriz-de-confusion-y-sus-metricas/>

**CAPÍTULO 3: APLICACIÓN DE LOS MODELOS DE MACHINE LEARNING Y ANALISIS
DE RESULTADOS**

3. ESTRUCTURA DE MODELAMIENTO Y PROCESAMIENTO DE DATOS

Con la base teórica y entendimiento de la lógica de los modelos de Machine Learning desarrollados en el capítulo 2, en la presente sección se desarrollan los pasos a seguir en el tratamiento de los datos y como se realizan los algoritmos, así como también los indicadores de mantenimiento que sean calculados con las metodologías de aprendizaje supervisado.

3.1. TRATAMIENTO DE LA DATA

3.1.1. Categorización de datos de entrenamiento

Se categorizará los modos de falla identificados en el componente crítico y se le asignará un valor de importancia. Una vez realizado este análisis de criticidad, se seleccionará un volumen de datos de entrenamiento a los cuales se les asignará dicha categoría.

3.1.2. División de datos para entrenamiento, validación y test

Una vez se tienen las variables de entrada preprocesadas, se realizará la parcelación de los datos en datos de entrenamiento, validación y test.

3.1.3. Programación de algoritmo con los datos de entrenamiento

Se realiza la programación a través de código Python en consola PyCharm Community Edition 2022.3.3 y CVS de entrada al algoritmo utilizando los modelos identificados en el capítulo 2. El entrenamiento se realiza con la fracción de datos seleccionados, se debe tener en consideración que mientras mayor sea el volumen de datos de entrenamiento mejores serán los resultados del aprendizaje del algoritmo.

3.1.4. Programación de algoritmo de predicción

Se definen las variables y se procede a la programación del modelo de predicción con el grupo de datos de validación para mejorar el modelo y realizar un ajuste final.

3.1.5. Test de algoritmo de predicción.

Con el conjunto de datos de test o prueba se mide la calidad del modelo

para lo cual se utilizarán las métricas de calidad vistos en el capítulo 2

Todos los códigos de programación se encuentran consignados en el Anexo 3.

3.1.6. Análisis de resultados

Para culminar se realizará el análisis de los resultados obtenidos, evaluando ventajas y desventajas.

3.2. CÁLCULO BASE CON WEIBULL

Adicionalmente al desarrollo de Machine learning, se presentan los resultados de evaluación de los datos con técnicas tradicionales como la Distribución de Weibull, la finalidad es identificar cual es el aporte de los modelos de aprendizaje por sobre la predicción con Weibull.

3.2.1. Importación de datos

En archivo 'CAEX930E3WB.csv', en anexo 2 se adjunta archivo, se encuentra la data con el tratamiento ya mencionado en los capítulos anteriores del cual se exportan los datos de descripción de falla, fecha, TFS y TEF, ver figura

```
C:\Users\klast\pythonProjectTT\Scripts\python.exe C:\Users\klast\PycharmProjects\pythonProjectTT\Weibull.py
```

	Descripcion	Fecha	TFS	TEF
0	BAJA POTENCIA	21-09-2019	7.8	1.43
1	FILTROS DE AIRE DE MOTOR DIESEL	15-05-2022	0.9	1.50
2	MODULO DIESEL(ESPERA DE TRASLADO)	24-12-2021	245.8	1.78
3	BAJA POTENCIA	24-06-2021	6.0	1.82
4	SIN PARTIDA	26-01-2022	24.1	3.23

Figura 3-1: Exportación de datos

Fuente. Elaboración propia en base a modelación en Python

3.2.2. Estimación de Parámetros Dist.Weibull

Se realiza el cálculo de Weibull con la función "fweibull(a,b,t), sin ranking ajustado y obteniendo además los valores de confiabilidad; para cálculo de pendiente e intersección se utilizó el método de Mínimos Cuadrados y tambien se estimó el Coeficiente de correlación de Pearson o coeficiente para determinar el coeficiente de determinación R².

Lo resultados se indican en Figura 3-2, en donde además se muestran el cálculo de los parámetros de Weibull. De los datos arrojados se puede afirmar:

- Con el cálculo del Factor de Forma β equivalente a 0,6218 se tiene una tasa de falla descendente, en donde el activo se encuentra en etapa de Mortalidad infantil.
- Por otra parte, se puede afirmar que a las 286 [hrs/falla] se espera que 63,2% de los camiones habrán fallado. ((IEC), 2008).
- Se estima que el Tiempo medio entre fallas es de 411,6 [hrs/falla].

```
RESULTADOS DEL AJUSTE WEIBULL
La ecuación de la Recta es:  0.621807154344182 X + -3.517224920563246
Coeficiente de Determinación R^2:  0.9852444656842967
El valor de Alpha es:  286.1328402448173
El valor de Beta es:  0.621807154344182
El valor de MTBF es:  411.6008864320373
```

Figura 3-2: Cálculo de parámetros Distribución Weibull
Fuente. Elaboración propia en base a modelación en Python

En la Figura 3-3 se desarrolla el grafico de dispersión de Weibull en donde se observa que los datos se ajustan a una linea recta.

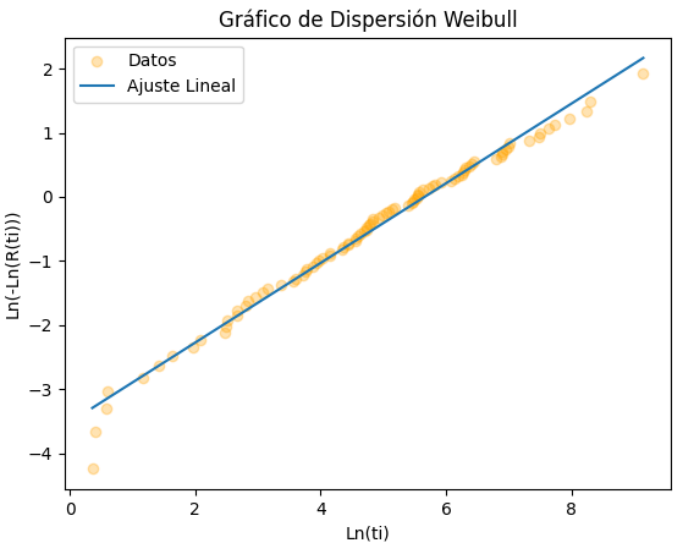


Figura 3-3: Gráfico de dispersión de Weibull
Fuente. Elaboración propia en base a modelación en Python

Por otra parte, en figura 3-4 se puede observar que, según la distribución de Weibull para el componente crítico en estudio, el MTBF calculado es de 411.6 [Horas], en dicho tiempo se tiene una confiabilidad aproximada cercana al 30%.

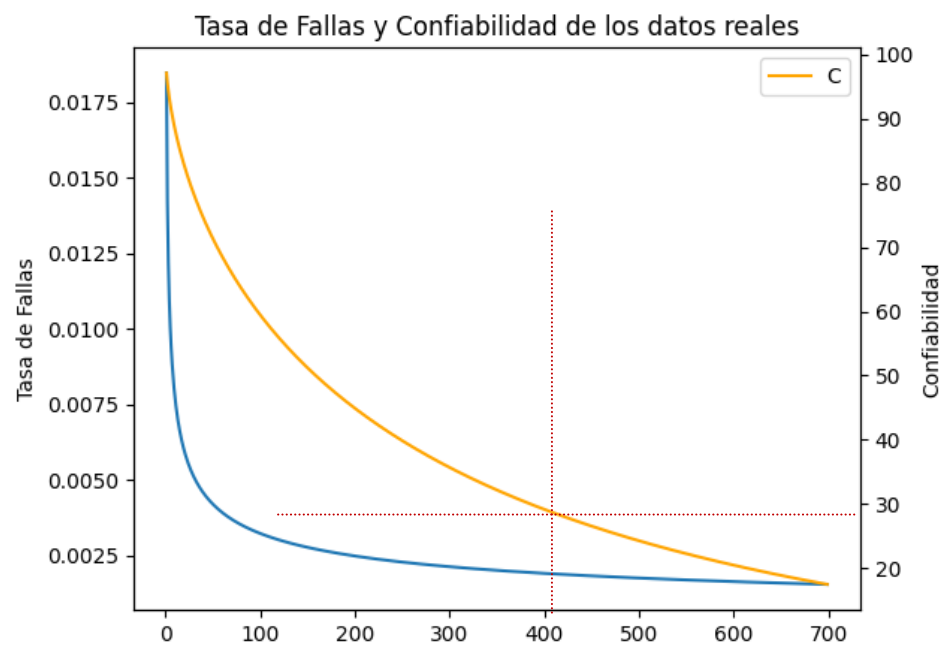


Figura 3-4: Tasa de Fallas y confiabilidad de datos reales
Fuente. Elaboración propia en base a modelación en Python

3.2.3. Creación de números aleatorios y cálculo de RMSE

En la siguiente etapa, se crean eventos aleatorios, un total de 91, con la función np.random.rand de numpy en Python que se usa para generar números aleatorios con una distribución uniforme entre 0 y 1. Esta función crea un arreglo de la forma dada y lo llena con muestras aleatorias de la distribución uniforme. Posteriormente genera la predicción del TEF (Y_predicted), ver figura 3-5.

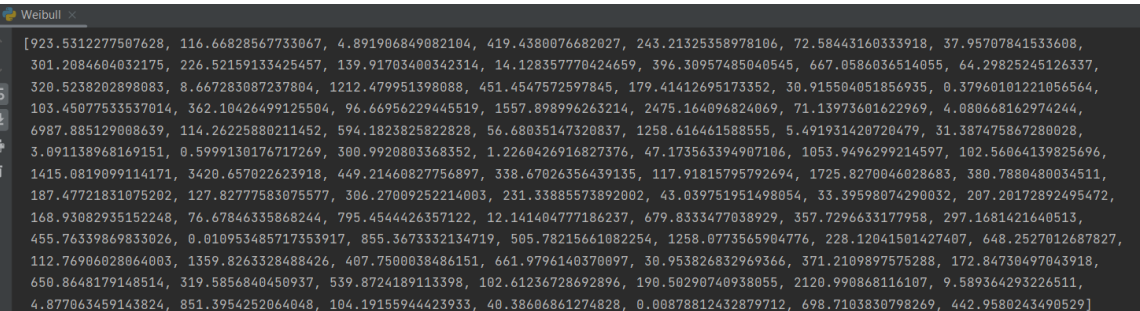


Figura 3-5: y_predictivo de datos aleatorios
Fuente. Elaboración propia en base a modelación en Python

3.2.4. Análisis Exploratorio de Datos

Con los datos aleatorios, los datos simulados se representan en Figura 3-6, en dicho gráfico se puede observar una mayor frecuencia absoluta para los datos presenta un TEF entre 0 y cercano a las 200 [horas/falla], lo que hace estimar que para el componente crítico en estudio hay altas posibilidades que falle antes de las 200 de operación posterior la última falla presentada.

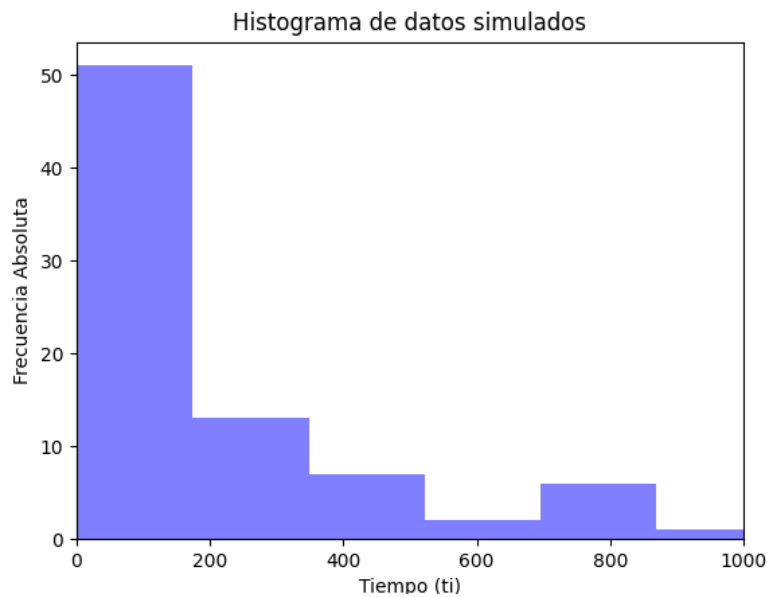


Figura 3-6: Histograma de Frecuencia Absoluta y TEF

Fuente. Elaboración propia en base a modelación en Python

Con respecto a la comparativa entre los datos reales y los predichos se observa en grafico 3-7, la línea roja representa las predicciones perfectas y no se la existencia de datos que cumplan esta condición

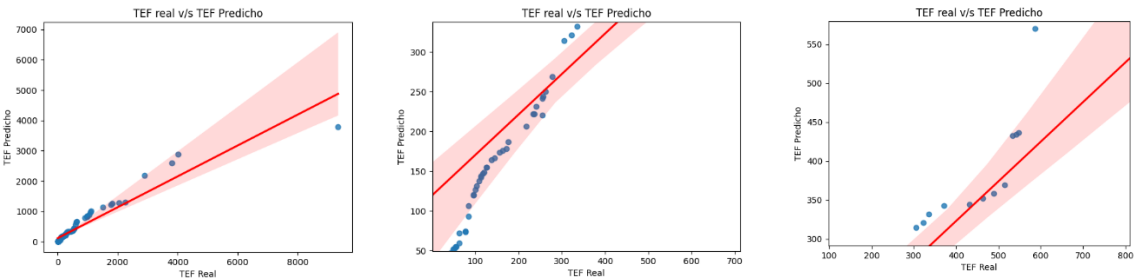


Figura 3-7: Comparativa de datos reales versus predichos

Fuente. Elaboración propia en base a modelación en Python

3.2.5. Determinación de errores

Con los datos estimados anteriormente, y los valores reales de TEF se calcula las medidas de calidad y se indican en Figura 3-8. Hay que tener en consideración que cada vez que se haga correr el algoritmo este valor de error cambiará, ya que los datos aleatorios de entrada se modificarán, sin embargo, el rango en el que se mueve el error no variará más de un 10%

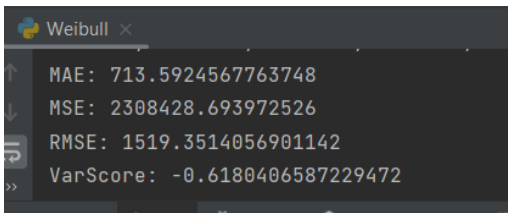


Figura 3-8: Medidas de Calidad

Fuente. Elaboración propia en base a modelación en Python

3.2.6. Interpretación de los resultados

De las medidas de calidad se observa que al tener un RMSE excesivamente alto, esto implica que las predicciones no se acercan en absoluto al valor objetivo, por ende, se puede afirmar que la predicción por medio de Distribución de Weibull no es el modelo más adecuado de predicción para los datos seleccionados.

3.3. **MODELO REGRESIÓN LOGÍSTICA**

Para realizar el modelamiento en Python es necesario realizar la importación de la función LogisticRegression y biblioteca sklearn con distintas métricas

3.3.1. Importación de datos

En archivo 'CAEX930Ecritico.csv', en anexo 2 se adjunta archivo, se encuentra la data con el tratamiento ya mencionado en los capítulos, los datos corresponden al CAEX crítico del modelo CAEX-930E, CA-50 A30398, el cual presenta un total de 875 registros con fechas de inicio y termino de fallas del 20-03-2016 y 18-08-2022 respectivamente, en este conjunto se encuentran 94 eventos de falla de componente crítico es decir “falla en Motor Diesel”

3.3.2. Exploración gráfica

En figura 3-9 se puede observar la distribución de los datos en el cual la clase 1 corresponde a la ocurrencia de falla en componente crítico, "Motor Diesel", mientras que la clase 0, muestra cuando la ocurrencia de falla es en cualquier otro componente. Se puede apreciar que las fallas del componente critico en el percentil 75 están concentradas aproximadamente 4.500 y 5200 [horas] de operación del camión.

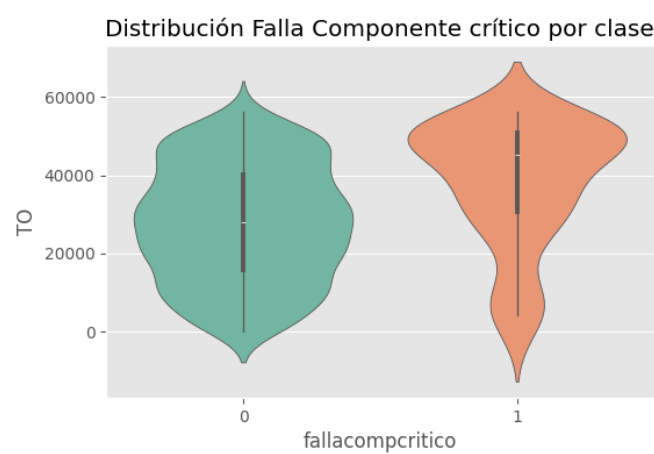


Figura 3-9: Gráfico de Violín para cada clase

Fuente. Elaboración propia en base a modelación en Python

3.3.3. Entrenamiento y ajuste del modelo

Se ajusta un modelo empleando como variable respuesta Falla de componente crítico y como predictor Horas de Funcionamiento. Como en todo estudio predictivo, no solo es importante ajustar el modelo, sino también cuantificar su capacidad para predecir nuevas observaciones. Para poder hacer esta evaluación, se dividen los datos en dos grupos, uno de entrenamiento y otro de test.

Posteriormente se procede a la creación del modelo, para ello se utiliza Statsmodels, es más completa que la de Scikitlearn ya que, además de ajustar el modelo, permite calcular los tests estadísticos y análisis necesarios para verificar que se cumplen las condiciones sobre las que se basa este tipo de modelos, el entrenamiento se realizara generando dos matrices, una con los predictores y otra con la variable respuesta. Esta es igual a la empleada por Scikitlearn con la diferencia de que a la matriz de predictores hay que añadirle una primera columna de 1s.

En figura 3-10 se observan los resultados del modelo después de la optimización que se logró después de 7 iteraciones.

```
Optimization terminated successfully.
Current function value: 0.294631
Iterations 7

Logit Regression Results
=====
Dep. Variable: y No. Observations: 700
Model: Logit Df Residuals: 698
Method: MLE Df Model: 1
Date: Sun, 21 Jan 2024 Pseudo R-squ.: 0.06633
Time: 18:46:29 Log-Likelihood: -206.24
converged: True LL-Null: -220.89
Covariance Type: nonrobust LLR p-value: 6.186e-08
=====
               coef      std err          z      P>|z|      [0.025      0.975]
-----
const      -3.8962      0.392      -9.929      0.000      -4.665      -3.127
x1          4.931e-05   9.86e-06       5.001      0.000       3e-05   6.86e-05
=====
```

Figura 3-10: Resultados de optimización del modelo

Fuente. Elaboración propia en base a modelación en Python

3.3.4. Predicción del modelo

Una vez entrenado el modelo, se pueden obtener predicciones para nuevos datos. Los modelos de regresión logística de statsmodels devuelven la probabilidad de pertenecer a la clase de referencia. En figura 3-11 se observan las predicciones que posteriormente son interpolados a 0 si esta probabilidad es menor a 0,5 y a e1 en caso contrario

```
[0.0641249 0.10668252 0.05029892 0.1474254 0.05087682 0.1871367
0.05231004 0.07552253 0.04395893 0.02654473 0.15756203 0.02001228
0.09877682 0.10191514 0.13186896 0.10863908 0.11631088 0.02403185
0.07123513 0.08731672 0.07889853 0.09420332 0.03915578 0.03105434
0.03627392 0.17510879 0.02330937 0.0615437 0.09123336 0.09098018
0.03798911 0.05240791 0.08899344 0.09312329 0.11586559 0.0287633
0.15717622 0.03623945 0.0851217 0.02303811 0.04737009 0.03381518
0.05190086 0.22537158 0.06983529 0.11391492 0.22104525 0.08925366
0.03890423 0.02673911]
```

Figura 3-11: Predicción de probabilidades

Fuente. Elaboración propia en base a modelación en Python

3.3.5. Representación gráfica del modelo.

En figura 3-12, el modelo muestra para la predicción, existe probabilidad de fallo del componente crítico en ciertas horas de funcionamiento bien definidas. En figura 3-13 se realiza un zoom del primer periodo en el cual se puede ver que existe probabilidad de falla del componente crítico, “Motor Diesel”, en las primeras horas de funcionamiento, antes de las 4.200 horas de funcionamiento y una nueva falla un poco más de 1000 horas de operación posterior.

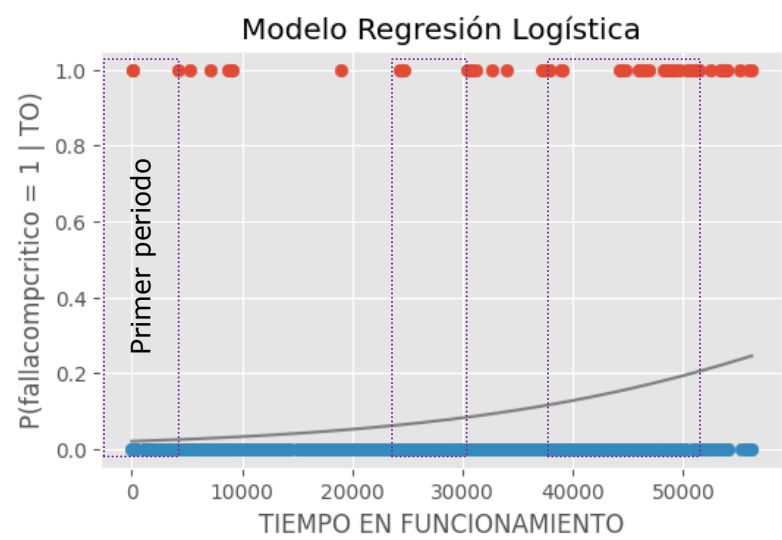


Figura 3-12: Predicción de probabilidades

Fuente. Elaboración propia en base a modelación en Python

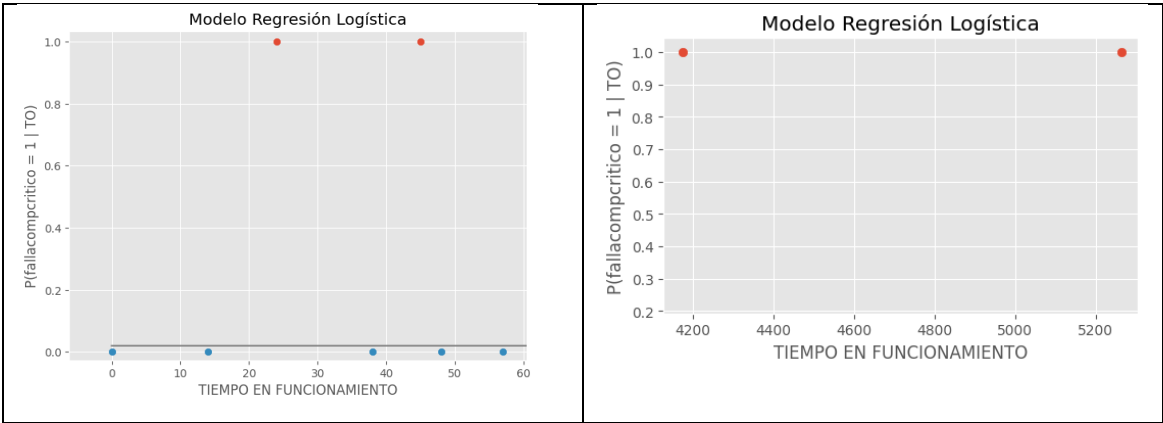


Figura 3-13: Predicción de probabilidades de fallo en primer y segundo periodo

Fuente. Elaboración propia en base a modelación en Python

3.3.6. Medidas de Calidad.

En figura 3-15 es posible observar las medidas de calidad del modelo. Las cuáles se puede indicar que el conjunto de predicciones presenta un F1 de 77.5% lo que es conservador, ya que, mientras más cercano a 100% el modelo es más preciso. Se utiliza esta medida de calidad dado que la distribución de las clases es desigual.

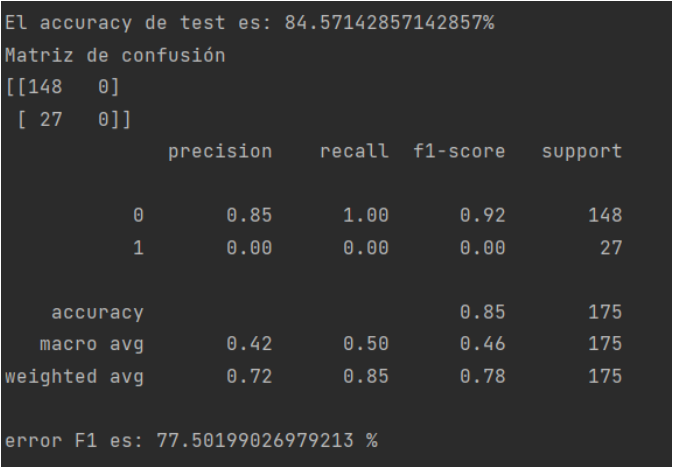


Figura 3-14: Medidas de Calidad Modelo RL

Fuente. Elaboración propia en base a modelación en Python

3.4. **MODELO MÁQUINA DE VECTORES SOPORTES**

En el desarrollo del presente modelo se utilizará las clases sklearn.svm.SVC que permiten crear modelos SVM de clasificación empleando kernel lineal, polinomial, radial o sigmoide. Con este algoritmo se tiene la ventaja que SVC controla la regularización a través del hiperparámetro C.

3.4.1. Importación de datos

Como se indicó en los apartados anteriores se sigue trabajando con el archivo 'CAEX930Ecritico.csv'.

3.4.2. Exploración gráfica

La figura 3-15 muestra la distribución de fallas categorizadas por dos características, en el eje X se indican las fallas ocurridas en el tiempo de operación del camión CAEX en análisis, mientras que el eje Y las clasifica por el tiempo Fuera en servicio que incurre cada falla, en amarillo se presentan aquellas fallas asociadas

al componente crítico, “Motor Diesel”, que evidencias altas horas fuera inoperativos por falla, mientras que las moradas son las fallas de cualquier otro componente, en imagen de la derecha, que corresponde a una ampliación de la izquierda, se puede apreciar que la gran densidad de datos a lo largo de la operación del CAEX muestran sus tiempos Fuera de Servicio son menores a 14 [horas]

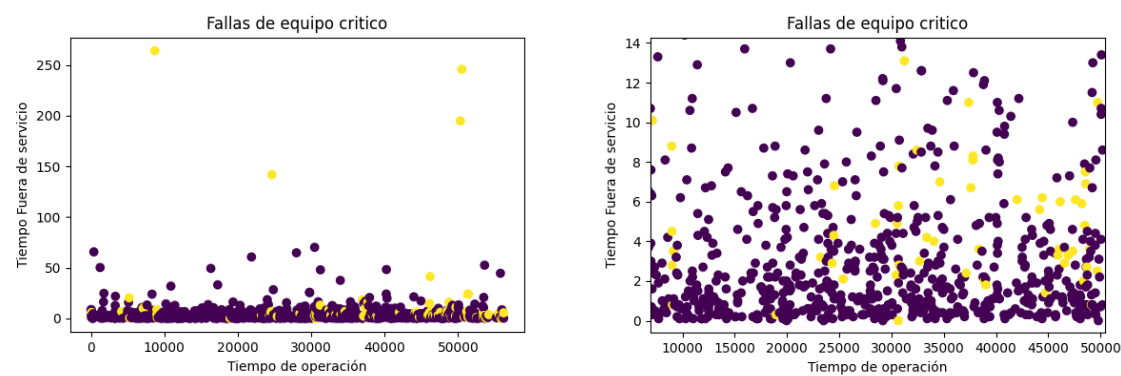


Figura 3-15: Gráfico de distribución de fallas de componente crítico

Fuente. Elaboración propia en base a modelación en Python

3.4.3. Entrenamiento y ajuste del modelo

Del mismo modo que se ha estado efectuando el desarrollo de los modelos anteriores se procede a dividir la muestra en los datos de train y test, para mayor detalle ver anexo 2 con códigos desarrollados.

Además, se realiza el ajuste del modelo y se calcula el valor del coeficiente del hiperplano o “valor C” el cual, es muy cercano a cero ($5,68 \times 10^{-7}$), este valor representa la penalización por clasificar erróneamente los datos de entrenamiento, al ser casi cero implica que SVM no les da mucha importancia a los errores de clasificación y busca un hiperplano con un margen amplio, que separe las clases de forma más suave. Esto puede hacer que el modelo sea más generalizado, pero también más propenso a ignorar algunos casos que podrían ser importantes. Por el contrario, si C es muy grande, la SVM intenta minimizar los errores de clasificación y busca un hiperplano con un margen estrecho, que separe las clases de forma más ajustada. Esto puede hacer que el modelo sea más preciso, pero también más sensible al ruido y a los datos atípicos. Por lo tanto, el valor de C afecta al equilibrio entre el sesgo y la varianza del modelo SVM.

3.4.4. Predicción del modelo y representación gráfica

La predicción se desarrolla con la función `clf.predict`, el algoritmo arrojó los datos predichos para ambas clases, (1= falla en componente crítico; 0= falla en otro componente), sin embargo, no fue posible graficar las predicciones, para poder realizar el mismo análisis realizado en los dos modelos anteriores, por lo cual se sugiere realizar un análisis más profundo de las causas que no permitieron dicho desarrollo.

3.4.5. Medidas de Calidad.

En figura 3-16 es posible observar las medidas de calidad del modelo. Se puede observar que el valor de F1 es bastante bajo lo que indica, que para la muestra seleccionada y las características de análisis este modelo no cumple con una predicción adecuada a los nuevos datos de entrada.

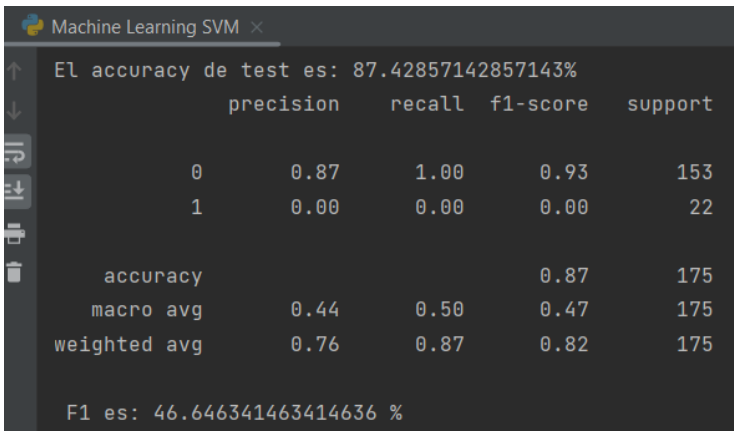


Figura 3-16: Medidas de Calidad Modelo SVM

Fuente. Elaboración propia en base a modelación en Python

3.5. MODELO RANDOM FOREST

Este modelo se desarrolla con la función `RandomForestRegressor` de la biblioteca `scikit-learn` del grupo de herramientas de ensamble.

3.5.1. Importación de datos

Como se indicó en los apartados anteriores se sigue trabajando con el archivo 'CAEX930Ecritico.csv'.

3.5.2. Entrenamiento y ajuste del modelo

Este modelo es del tipo de regresión por lo cual el tratamiento del entrenamiento y ajuste va dirigido a considerar mismas variables desarrolladas en la predicción con Weibull, es decir, se busca predecir el tiempo entre falla para datos de entrada. Para los datos de entrenamiento se calcula el RMSE, cuyo valor es de RMSE= 76.91494706029677, considerando un numero de estimadores igual a 150

Es necesario tener en cuenta que en el proceso de entrenamiento se debe identificar el mejor estimador ya que esto reducirá los errores. Cada árbol de decisión es entrenado con una muestra aleatoria y cada uno genera una desviación del promedio de las predicciones (sesgo) y además el modelo cambia en función de los datos utilizados para el entrenamiento (varianza). Para ello en figura 3-17 muestra la evolución del error según los estimadores o números de árboles generados por el modelo, identificando que el numero recomendado es de 85, haciendo que disminuya levemente el RSME a 76.9109095064208.

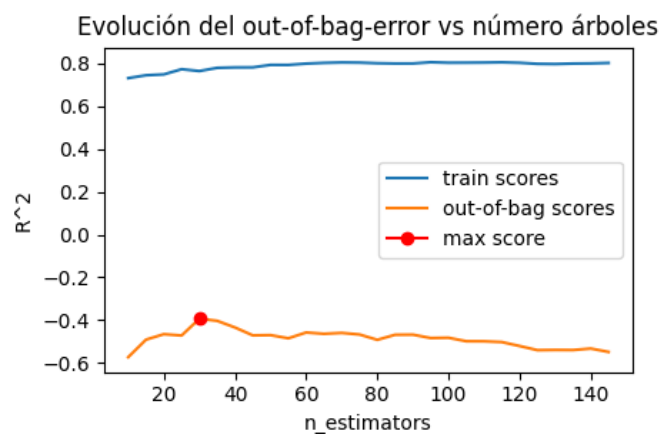


Figura 3-17: Evolución del error versus número de árboles RF

Fuente. Elaboración propia en base a modelación en Python
Python

3.5.3. Predicción del modelo y representación gráfica

La predicción se desarrolla con la función rf_model.predict. En figura 3-19 se logra identificar que existen datos correctamente predichos, (zona roja), asociados a la clase objetivo, los cuales indican que el componente crítico, Motor Diesel. Además, cabe destacar que con los datos de predichos el modelo categoriza la gran mayoría de fallos presentan TEF menores a 125 [horas].

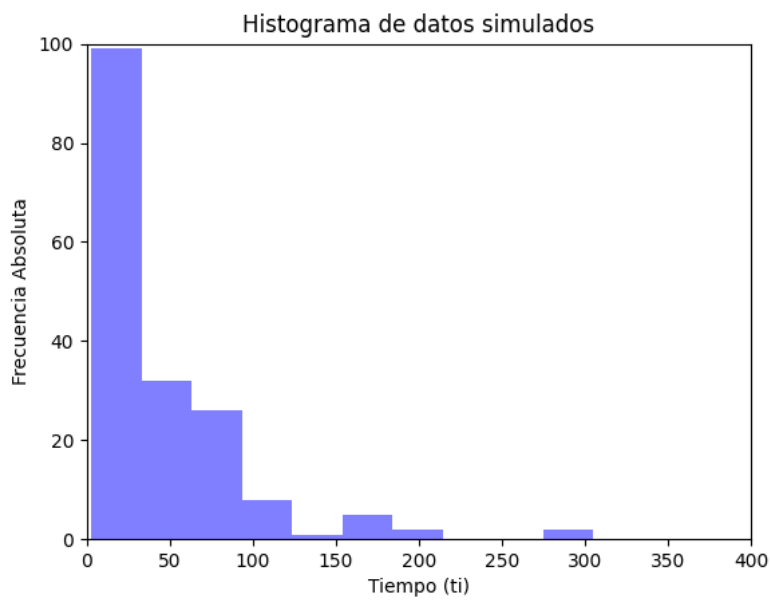


Figura 3-18: Distribución de datos de test

Fuente. Elaboración propia en base a modelación de RF.

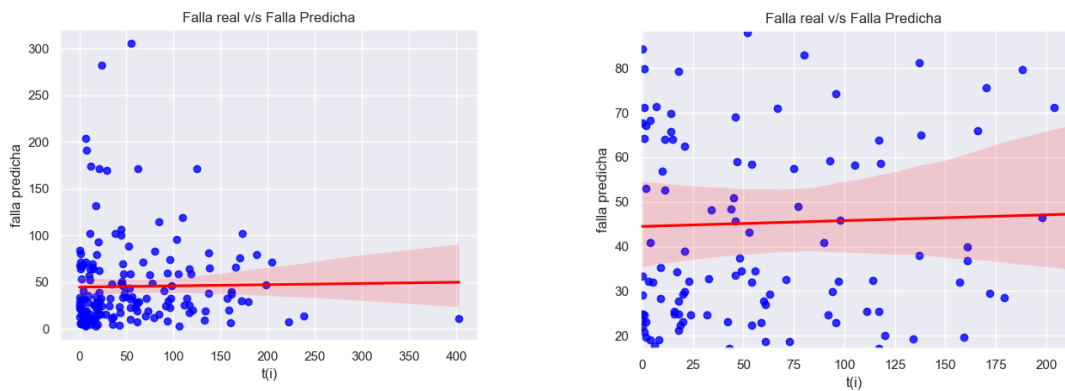


Figura 3-19: Predicción de falla de Componente crítico con RF

Fuente. Elaboración propia en base a modelación en Python

3.5.4. Medidas de Calidad.

En figura 3-20 es posible observar las medidas de calidad del modelo. Este modelo del tipo regresión se evalúa con la medida de calidad RMSE el cual sustancialmente bajo en comparación al obtenido en la predicción por medio de Weibull

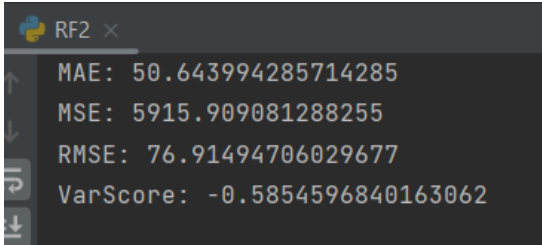


Figura 3-20: Medidas de Calidad de Modelo RF

Fuente. Elaboración propia en base a modelación en Python

3.6. CUADRO COMPARATIVO DE MODELOS

En tablas 3-1 y 3-2 se consignan los valores obtenidos en las medidas de calidad de los modelos agrupados según tipo de ML. Estos cuadros comparativos permitirán generar las conclusiones a exponer al final de este desarrollo.

Regresión	Modelo	Predicción	Medidas de calidad			
			MAE	MSE	RMSE	VarScore
	Distribución de Weibull	TEF	713,592	2308428,693	1519,351	-0,618
	Random Forest	TEF	50,643	5915,909	76,915	-0,585

Tabla 3-1: Cuadro comparativo de medidas de calidad RF y Weibull

Fuente: Elaboración propia en base a resultados de modelamiento

Clasificación	Modelo	Predicción	Clase	Medidas de calidad				
				Precisión	Recall	f1-score	Accuracy	F1
	Regresión Logística	TO	0	0,85	1	0,92	84,571%	77,502%
			1	0	0	0		
	Máquina de Soportes de Vectores	TO	0	0,87	1	0,93	87,429%	46,646%
			1	0	0	0		

Tabla 3-2: Cuadro comparativo de medidas de calidad RL y SMV

Fuente: Elaboración propia en base a resultados de modelamiento

CONCLUSIONES

- De la base de datos histórico se logró identificar el componente crítico, la metodología que permitió cumplir con el objetivo fue el Diagrama de Dispersión Logarítmico. El resultado de este análisis indica que Componente Motor Diesel es el que presenta una condición más aguda-crónica, ya que, es quien aporta con más imprevistos (555) y genera mayor indisponibilidad en términos de detenciones no programadas, con más de 6300 horas de detención en el periodo en estudio. Por otra parte, el camión crítico corresponde al CA-5 A30398, el cual presenta el mayor MTTR del sistema con 14,38 [horas].
- Por medio de antecedentes recabados de estudios, artículos, y publicaciones fue posible realizar un análisis cualitativo, de la aplicación de técnicas de Machine Learning en el Mantenimiento Predictivo, concluyendo que las principales técnicas corresponden a Random Forest, Redes Neuronales, Máquinas de vectores de soporte y K- Vecinos más cercanos. Adicionalmente se utilizaron motores de búsqueda de inteligencia artificial, tales como Bing, los cuales arrojaron similares resultados posicionando en los primeros lugares como herramientas de predicción de fallos a Regresión Lineal, Regresión Logística Random Forest y Maquinas de Vectores de Soporte. No se logró identificar en la literatura un desarrollo o estudio que generara una jerarquización de las técnicas de Machine Learning aplicadas a la predicción de fallos, por esta razón se apuntó al objetivo de seleccionar una o más técnica de aprendizaje Supervisado.
- Se realiza el cálculo base de confiabilidad con Distribución de Weibull, el cual determinó una tasa de falla descendente en el equipo antes mencionado con componente crítico "Motor Diesel", además de identificar que el activo se encuentra en etapa de Mortalidad infantil, por otra parte, se puede afirmar que a las 286 [hrs/falla] se espera que 63,2% de los camiones habrán fallado y el cálculo determinó que el Tiempo medio entre fallas es de 411,6 [hrs/falla]. Se realiza una predicción con datos aleatorios arrojando que para el Motor Diesel más del 50% de los datos predichos, tienen tiempos entre falla (TEF), menores a 200[horas]. Esta modelación obtuvo un RMSE igual a 1.519,351, esta medida de precisión también conocida como Raíz del Error Medio cuadrático al ser muy elevado implica que el modelo no es preciso, ya que los datos observados presentan un mal ajuste de los valores predichos del modelo. A diferencia del Modelo de Random Forest en donde el modelamiento realizado más del 50% de los datos simulados presentaron valores de TEF menores a 125[horas] y El RMSE obtenido es de 76,915, sustancialmente menor que el obtenido en el modelamiento de Weibull, equivalente a un poco más del 5% de éste último, lo cual evidencia que este modelo presenta un mejor ajuste, ya que la

interpretación de esta medida implica que las predicciones del modelo final se alejan en promedio 76,9 unidades del valor real, por lo tanto, se concluye que para el conjunto de datos seleccionados las predicciones de Weibull no presentan resultados favorables, siendo una buena alternativa de predicción para fallos del componente crítico el modelo de Random Forest .

- La modelación con la técnica de Regresión Logística, del tipo de Machine learning de Clasificación, presento probabilidad de falla del componente crítico, "Motor Diesel" en las primeras horas de funcionamiento, antes de las 4.200 horas de funcionamiento y una nueva falla un poco más de 1000 horas de operación posterior. Las medidas de calidad no son favorables para la clase 1, esto implica que el modelo no logra clasificar la clase correctamente, esto suele ocurrir cuando la distribución de las clases es desigual, es en estos casos en donde el parámetro de calidad de evaluación del modelo es el valor F1 que la precisión con la sensibilidad es del 77,5%. Para el caso de modelamiento con Máquina de Vectores de Soporte el resultado obtenido del valor de F1 es bastante bajo lo que indica, que para la muestra seleccionada y las características de análisis este modelo no cumple con una predicción adecuada a los nuevos datos de entrada. Y en comparación al modelo de Regresión Logística, no es la técnica más apropiada de predicción

- Como conclusión general de las técnicas de Machine Learning considerados se puede indicar que los modelos que presentaron mejores resultados en la predicción de fallos del componente crítico, es decir, Motor Diesel, corresponde al modelo de Regresión Logística, que es capaz de predecir en que tiempo de funcionamiento del motor este presentara una falla y sus resultados son con bastante precisión por sobre Maquina de Soporte de Vectores con un F1 de 77,5% y si bien no es muy cercano a 100% que es el mejor escenario de predicción, esto es principalmente por porque las clases consideradas (1=falla componente crítico y 0=falla en otro componente) es muy desigual. Por otra parte, en el caso de predicción de falla del componente critico prediciendo el TEF del éste el modelo más adecuado es Random Forest el cual presenta un RSME 95% menor que el Weibull, siendo este modelo la mejor opción predicción. Con lo antes descrito se puede dar por logrado el objetivo de este trabajo, que era "Proponer modelo de Machine Learning, para la predicción de fallas de componentes críticos de camiones de extracción, en industria Minera, seleccionando la metodología que presente la mejor métrica de calidad".

Se sugiere considerar ejecutar los modelos con un conjunto de datos más homogéneos o parcelar la data en periodos de tiempo más cortos permitiendo realizar entrenamiento constante con pequeños grupos de datos, para analizar el comportamiento de los modelos.

BIBLIOGRAFÍA

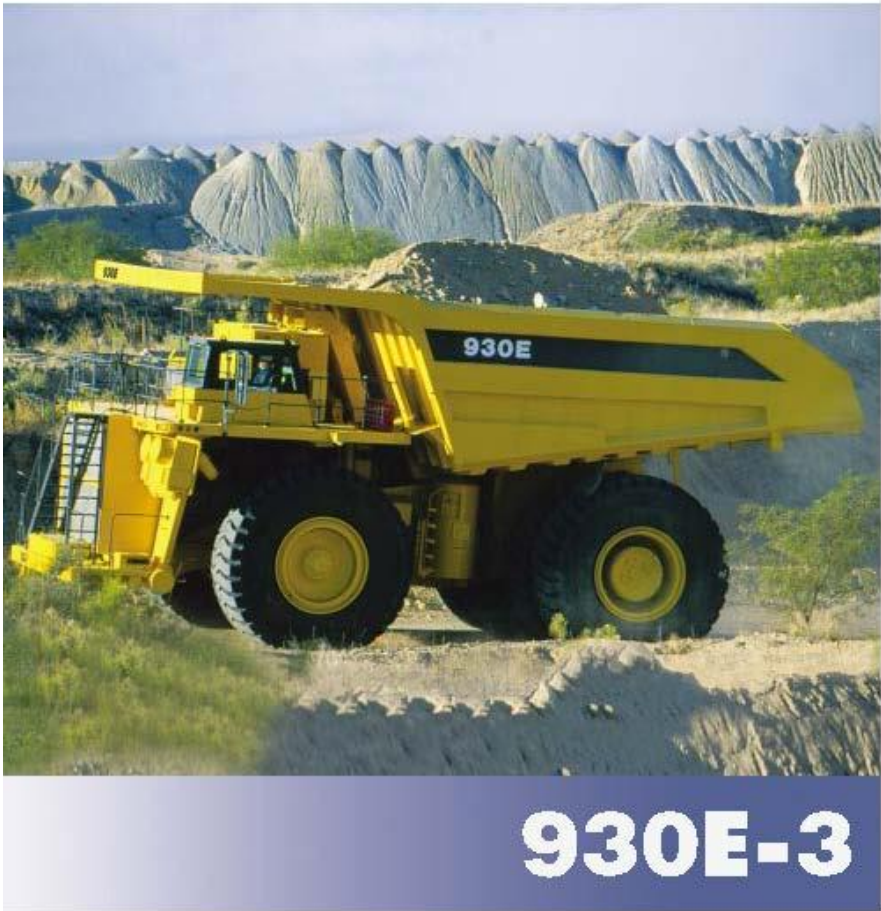
- (n.d.). Retrieved from <https://www.ibm.com/mx-es/topics/random-forest>
- (n.d.). Retrieved from <https://cienciadedatos.net/documentos/py24-svm-python>
- (n.d.). Retrieved from https://colab.research.google.com/drive/164nCeA_jj1u86KakWgFZDjX5LE004u3o#scrollTo=objEv-dgcQYM
- (IEC), T. 6. (2008). *INTERNATIONAL STANDARD WEIBULL ANALYSIS* (2.0 ed.). Geneva, Switzerland. Retrieved from [https://standards.iteh.ai/catalog/standards/sist/8f778a3f-f6e1-47e8-b78a-](https://standards.iteh.ai/catalog/standards/sist/8f778a3f-f6e1-47e8-b78a-baptista, M., Sankararaman, S., P. de Medeiros, I., P. Malere, J., Nascimento Jr, C., Prendinger, H., & M.P. Henriques, E. (2017). Forecasting Fault Events for Predictive Maintenance using Data-driven Techniques and ARMA Modeling.)
- Baptista, M., Sankararaman, S., P. de Medeiros, I., P. Malere, J., Nascimento Jr, C., Prendinger, H., & M.P. Henriques, E. (2017). Forecasting Fault Events for Predictive Maintenance using Data-driven Techniques and ARMA Modeling.
- Battifarano, M., DeSmet, D., Madabhushi, A., & Nabar, P. (2018). Predicting Future Machine Failure from Machine. *Research Gate*.
- Carvalho, T. S. (2019, 08 27). A systematic literature review of machine learning methods applied to predictive. *Journal Pre-proofs*.
- Cristian, Z. O. (n.d.). Retrieved from <https://es.slideshare.net/PaulaTapiaAguilar/prsentacion-caex-930e-3y4-czopdf>
- Gálvez Bernal, L. F. (2021). *PROPUESTA DE MEJORA A GESTIÓN DE MANTENIMIENTO UTILIZANDO HERRAMIENTAS DE APRENDIZAJE SUPERVISADO EN EQUIPO CRÍTICO DE PLANTA CELULOSA*. Trabajo de Titulación para optar a l Título de Ingeniero de Mantenimiento Industrial Licenciado en Ingeniería.
- <https://aws.amazon.com/>. (n.d.). Retrieved from <https://aws.amazon.com/es/what-is/generative-ai/>
- Ismael, C. A. (2021). *PROPUESTAS DE MEJORA A LA GESTIÓN DE MANTENIMIENTO DE SISTEMA CRÍTICO DE CAMIONES EXTRACCIÓN 930E DE MINERA LOS PELAMBRES*. Trabajo de Titulación para optar al Título profesional de Ingeniero Industrial, Universidad Técnica Federico Santa María.
- Jesus, B. (2020). *Machine Learning y Deep Learning Usando Python, Scikit y Keras*. Bogotá Colombia: Ra-ma Editorial.
- Miguel, Z. R. (2023). <https://www.researchgate.net/>. doi:10.35542/osf.io/86ykm
- Sparano, A., Ramiírez, J., Guerra, L., & Terán , R. (2023). Detección de fallas incipientes en rodamientos de Generadores. *Revista Ingeniería UC*, 28. doi:10.54139/revinguc
- Tebar, S. M. (2023). *Qué es la Inteligencia Artificial y Cómo se Usa.docx*. doi:10.13140/RG.2.2.19496.85761
- The Engineering Society For Advancing Mobility Land Sea Air and Space, S. I. (1999). NORMA PARA VEÍCULOS AEROESPACIALES Y DE SUPERFICIE SAE JA1011.

ANEXO 1

KOMATSU®

930E-3

MAXIMUM GVW
498952 kg 1,100,000 lb
PAYLOAD
290 m ton 320 U.S. ton
GROSS HORSEPOWER
2014 kW 2,700 HP



ELECTRIC DRIVE TRUCK

SPECIFICATIONS



ENGINE

Make and model	Komatsu SSDA16V160
Fuel	Diesel
Number of cylinders	16
Operating cycle	4 cycle
*Rated brake power	2014 kW 2,700 HP @ 1900 rpm
**Flywheel power	1902 kW 2,550 HP @ 1900 rpm
Weight (wet)	9608 kg 21,182 lb

*Rated brake power is the output of the engine as installed in this machine, at governed rpm and with engine manufacturer's approved fuel setting. Accessory losses included are water pump, fuel pump and oil pump
**Flywheel power is the rated power at the engine flywheel minus the average accessory losses. Accessories include fan and charging alternator. Rating(s) represent gross engine performance in accordance with SAE J1349 conditions.



ELECTRIC DRIVE

AC/DC CURRENT	
Alternator	GTA-41
Dual impeller in-line blower	340 m³/min 12,000 cfm
Control	AC Torque Control System
*Motorized wheels	GDY106 Induction Traction Motors
*Ratio	32.62:1
Speed (maximum)	64.5 km/h 40 mph

*Wheel motor application depends upon gross vehicle weight, haul road grade, haul road length, rolling resistance and other parameters. Komatsu and G.E. must analyze each job condition to assure proper application.
*Optional ratios available.



TIRES AND RIMS

Rock service, tubeless, radial tires	
*Standard tire	53/80 R63
Standard five (5) piece rim	
914 mm x 1600 mm x 127 mm 36" x 63" x 5.0" patented Phase I rim assembly with forged center bolt flange and 1626 mm 64" centerline dual spacing. Rated at 758 kPa 110 psi cold inflation pressure for rims.	

*Tires should meet application requirements for **tightness**, tread, compound, inflation pressure, ply rating or equivalent, etc.



BODY

All-welded steel flat floor body with horizontal bolsters and full canopy. Eyebrow, rear wheel rock ejectors, body tilt cable, and rubber mounts on frame are standard. Pivot exhaust heating optional.	
Floor sheet	16 mm 0.63" 1379 mPa 200,000 psi tensile strength steel (two-piece)
Front sheet	9 mm 0.35" 1379 mPa 200,000 psi tensile strength steel
Side sheet	8 mm 0.31" 1379 mPa 200,000 psi tensile strength steel
Canopy sheet	5 mm 0.19" 690 mPa 100,000 psi
Struck	171 m³ 224 yd³
Standard SAE heaped 2:1	211 m³ 276 yd³

*Komatsu must approve all bodies through a Body Application Worksheet.



CAB

Advanced Operator Environment with integral 4-post ROPS/FOPS structure (meets J1040 Apr88), adjustable air suspension seat w/lumbar support and arm rests, passenger seat, maximum R-value insulation, tilt and telescoping steering wheel, electric windshield wipers w/washer, tinted safety glass, power windows, Komatsu Payload Weighing System, 55,000 Btu/hr heater and defroster, 21,600 Btu/hr air conditioning (HFC - 134A refrigerant).



FRAME

Advanced technology, full butt-welded box sectional ladder-type frame with integral ROPS supports, integral front bumper, rear tubular cross members, steel castings at all critical stress transition zones, rugged continuous horsecollar.	
Plate material	482.6 mPa 70,000 psi tensile strength steel
Casting material	620.5 mPa 90,000 psi tensile strength steel
Rail width	305 mm 12"
Rail depth (minimum)	864 mm 34"
Top and bottom plate thickness	45 mm 1.77"
Side plate thickness	25 mm 0.98"
Drive axle mounting	Pin and spherical bushing
Drive axle alignment	Swing link between frame and axle



BRAKING SYSTEM

Service brakes: oil-cooled, hydraulic-actuated, multiple disc brakes at each wheel. Traction system wheel slip/slide control.	
Max. service apply pressure	17237 kPa 2500 psi
Total friction area per brake	97025 cm² 15,038 in²
Emergency brakes	Automatically applied prior to hydraulic system pressure dropping below level required to meet secondary stopping requirements.
Wheel brake locks	Switch activated
Parking brakes	Multiple disc, spring-applied, hydraulically-released, dry brakes on inboard end of each wheel motor rotor shaft. Rated to hold on ±15% grade @ maximum gross vehicle weight.
Electric dynamic retarder	Max 4026 kW 5400 hp Continuous 2460 kW 3300 hp
Continuously rated (16 element) high-density blown grids w/retard at engine idle and retard in reverse propulsion.	



SUSPENSION

HYDRAIR® II	
Variable rate hydro-pneumatic with integral rebound control	
Max. front stroke	328 mm 12.92"
Max. rear stroke	239 mm 9.40"
Max. rear axle oscillation	+6.5°

*Rear suspensions are inverted.



COOLING SYSTEM

L&M radiator assembly, split-flow, with deaeration-type top tank.	
Radiator frontal area	7.02 m² 75.5 ft²



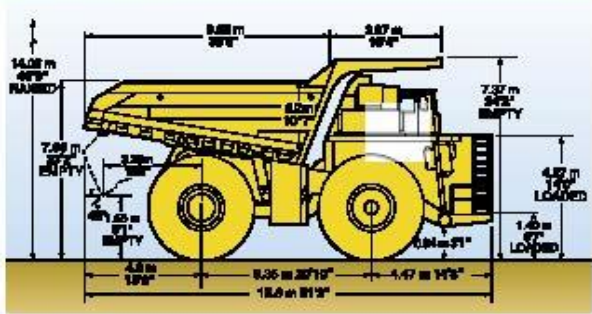
SERVICE CAPACITIES

Cooling System	594 L	157 U.S. gal
Crankcase*	280 L	74 U.S. gal
Hydraulic system	1325 L	350 U.S. gal
Motor gear box (each)	95 L	25 U.S. gal
Fuel	4542 L	1200 U.S. gal

*Includes lube oil filters

930E-3
ELECTRIC DRIVE TRUCK

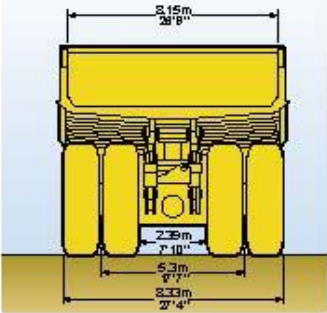
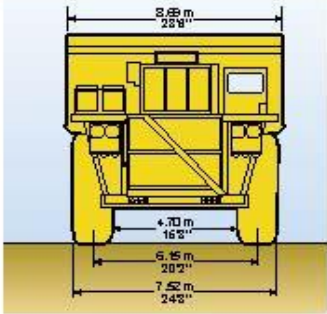
DIMENSIONS



All dimensions are with 211 m³ 276 yd³ body.

Bodies	Struck	2:1 Heap	*Loading Height
Standard	17.1 m ³ 224 yd ³	211 m ³ 276 yd ³	7.06 m 23'2"

*Tool box 1.43 m (4'8") high, 0.61 m (2'0") wide, 0.61 m (2'0") deep.



HYDRAULIC SYSTEM

- Steering.....Accommodator assisted with double acting cylinder provides constant rate steering. Emergency steering automatically applied by accommodator.
- Trailing circle diameter (SAB).....30.4 m 97'7"
- Reservoir.....947 L 250 U.S. gal
- Filtration.....In-line replaceable element
- Steering.....Single, full flow, 100 mesh
- Hoist and steering.....Dual, in-line, high pressure, $p_{12} = 200$.
- Compensable lift.....Above deck, easily accessible with diagnostic test connections.
- Hoist.....Two 3-stage dual-acting on-board cylinders, internal cushion valve, over-center dampening.
- Hoist times
 - Power-up loaded.....21 sec
 - Power-down.....23 sec
 - Float-down empty.....24 sec
- Pumps.....Two pumps, single package, end of alternator
- Hoist and brake cooling.....Tandem gear pump with output of 1022 ltr @ 246 gpm at 1900 rpm and 17.237 kPa 2,500 psi
- Steering and brake.....Pressure-compensating piston pump with output of 246 ltr @ 65 gpm at 1900 rpm and 18.961 kPa 2,750 psi
- System relief pressures
 - Hoist and brake cooling.....17.237 kPa 2,500 psi
 - Steering and brake.....27.579 kPa 4,000 psi
- Quick disconnects standard for powering disabled truck and for systems diagnostics.

ELECTRICAL SYSTEM

- 4 x 80 1450 CCA, 12 volt, in series (parallel), and 2 x 30H 800 CCA, 220-ampere-hour batteries, bumper mounted with disconnect switch.
- Alternator.....24-volt, 240 amp
- Lighting.....24-volt
- Starters.....Two 24-volt

WEIGHT DISTRIBUTION

Empty Vehicle	kg	lb	%
Front axle	102830	226,699	49.3
Rear axle	105760	233,137	50.7
Total (net, 50% tie b)	208590	459,836	
Loaded Vehicle at maximum GVW rating			
Front axle	164655	363,000	33
Rear axle	334301	737,000	67
Total	498957	1,100,000	

NOTE: GVW shall not exceed 498957 kg / 1100000 lb including options, taxes, fuel and payload, subject to application approval by Komatsu.

NOTE: Komatsu trucks comply with SAE specifications for cab noise, ROPS, steering and braking. Cover photos and illustrations may show optional equipment. Models and specifications subject to change without notice.



STANDARD EQUIPMENT

- Air cleaners, dry type SRG
 - Alternator (24V/240A)
 - Auto lubrication system w/ground level fill & level indicator
 - Batteries—4 x 8D and 2 x 30H (1450 CCA's)
 - Battery charging cable and socket
 - Body over center device
 - Brakes: oil-cooled, multiple disc front & rear
 - Control cabinet
 - Electric start
 - Eliminator, sentinel, reserve
 - Filters, high pressure hydraulic
 - Gate valve on hydraulic tank
 - Ground level radiator fill
 - Komatsu Payload Weighing System—PLM-III
 - Mirrors, LH flat and RH rectangular convex
 - Mud flaps
 - Muffled exhaust—deck-mounted
 - Power supply for 2-way radio
 - Quick disconnects (hoist and diagnostics)
 - Radiator sight gauge
 - Removable power module unit (radiator, engine, alternator, blower)
 - Retard grids, 18-element, blown
 - Retard speed control
 - Reverse retarding
 - Rock ejectors
 - Thermostatic Fan Clutch
 - Fast-Fill Fuel System (in tank and left side remote)
 - Service Center—LH
 - Body Impact Plate

SAFETY:

 - All hydraulic service brakes with emergency auto apply
 - Battery disconnect switch
 - Body prop cable
 - Brake lock and drive system interlock
 - Diagonal staircase across grill
 - Dynamic retarding with continuous rated 18-element grids
 - Engine shutdown at ground level
 - Hoist propulsion interlock
 - Horns (electric—front and back-up)
 - Integral ROPS/FOPS Cab
 - Maintenance and power lockout
- Parking brakes with warning light & speed application protection
 - Power steering w/auto emergency steering
 - Protective deck handrails
 - Pump driveline protector
 - Radiator fan guard
 - Seat belts (3" wide, retractable)
 - Skid-resistant coating on walkways
 - 24-volt circuit breakers

CAB:

 - AC Drive Interface Display
 - Air cleaner vacuum gauges
 - Air conditioner HFC-134A
 - Alarm System warning lights (red)
 - Accumulator pre-charge
 - Coolant low level w/alarm
 - Drive system temperature w/alarm
 - Electric system fault w/alarm
 - Hydraulic oil level w/alarm
 - Hydraulic oil temperature w/alarm
 - Low engine oil pressure w/alarm
 - Low brake pressure w/alarm
 - No propel/retard power w/alarm
 - Stop engine no propel w/alarm
 - Steering pressure w/alarm
 - AM/FM radio cassette player
 - Column-mounted retarder control
 - Digital tachometer and speedometer
 - Dome light
 - Engine hourmeter, oil pressure gauge, coolant temperature gauge, hydraulic oil temperature gauge
 - Engine shutdown w/5 minute delay timer
 - Floor mat (double barrier)
 - Fuel gauge in cab and on tank
 - Fuel low level light and buzzer
 - Gauges (w/backlight)
 - Headlight switch
 - Heater and defroster (heavy-duty)
 - Heater switch
 - High beam selector and indicator
 - Horn (center of steering wheel)
 - Indicator lights (amber):
 - Check engine w/alarm
 - Circuit breaker tripped
 - Coolant high temperature w/alarm
- Drive system reset switch
 - Drive system temperature w/alarm
 - Dump body up
 - Dynamic retarding
 - Manual back-up lights
 - Park brakes applied
 - Propel system not ready
 - Reduced propel system mode
 - Retard at continuous level
 - Retard speed control
 - Service brakes
 - Voltmeter (battery output)
 - Insulation (Max R-Value)
 - Operator seat, adjustable w/air suspension, lumbar support and arm rests
 - Panel lighting (adjustable)
 - Passenger seat
 - Power windows
 - Pressurized cab air system w/fan on
 - Single brake/retarder pedal
 - Starter key switch
 - Sunvisor (adjustable)
 - Tilt & telescoping steering wheel (adjustable)
 - Windshield (tinted safety plate)
 - Windshield wiper (dual) and washer (electric)

LIGHTING:

 - Back-up lights—rear mount (2) halogen
 - Back-up lights—R and L - deck mount (2) halogen
 - Brake and retard lights on top of cab
 - Clearance lights
 - Control cabinet service light
 - Dynamic retarding, rear (2) halogen
 - Engine compartment service lights
 - Fog lights (2) halogen
 - Headlights (8) halogen
 - Stairway lights
 - Manual back-up light switch & indicator
 - Payload lights R and L
 - Platform lights R, L and Center
 - Stop & tail lights (2) halogen
 - Turn signals halogen
 - Under-hood service lights



OPTIONAL EQUIPMENT

- (Optional equipment may change operating weight).
- Air filter evacuators
 - Body Liners*
 - Fire extinguisher
 - Heated Body
 - High altitude grid
 - Hot start engine coolant (220V 2-2500W)
 - Hot start engine oil (220V 2-600W)
 - Modular Mining Systems (MMS) ready
 - Motorized wheel ratios (higher-lower)
 - Mufflers between frame rails
 - Service Center—RH
 - Special language markings
 - Suspensions, cold weather
- *Available factory installed or non-installed. All other options and accessories listed are available factory installed only.

440 N. Fairway Drive
P.O. Box 8112
Vernon Hills, IL 60061-8112

ANEXO 2**A. *BASE DE DATOS PARA EXPORTAR A PYTHON CAEX930E3WB.CVS***

Descripcion,Fecha,TFS,TEF
 BAJA POTENCIA,21-09-2019,7.8,1.43
 FILTROS DE AIRE DE MOTOR DIESEL,15-05-2022,0.9,1.5
 MODULO DIESEL(ESPERA DE TRASLADO),24-12-2021,245.8,1.78
 BAJA POTENCIA,24-06-2021,6,1.82
 SIN PARTIDA,26-01-2022,24.1,3.23
 BAJA_POTENCIA,12-07-2020,8.3,4.12
 FALLA MOTOR DIESEL,29-01-2022,6.5,5.13
 FALLA MOTOR DIESEL,28-01-2022,6.7,7.17
 RPM,29-03-2017,8.8,8.02
 STALL MOTOR,30-01-2022,24.1,11.72
 ADV_PROPULSION,29-03-2017,4.5,12.12
 ADV_PROPULSION,30-03-2017,2.8,12.22
 ROJA MOTOR,22-03-2016,1.6,14.32
 ROJA MOTOR,13-09-2019,4.9,14.4
 DETENCION MOTOR DIESEL,03-10-2021,6.9,16.53
 SIN PARTIDA,29-01-2022,1.3,17.13
 ROJA MOTOR,26-04-2022,0.8,19.43
 ADV_PROPULSION,31-03-2017,0.8,21.83
 BAJA POTENCIA,27-04-2022,2.6,23.33
 ROJA MOTOR,23-03-2016,5.9,29.27
 ROJA MOTOR,09-01-2022,2.4,35.35
 BAJA POTENCIA,28-10-2021,0.8,36.38
 BAJA POTENCIA,08-01-2022,1.7,41.55
 BAJA POTENCIA,02-10-2021,7.5,43.05
 BAJA POTENCIA,26-06-2021,41.3,43.77
 FALLA MOTOR DIESEL,20-11-2021,11,47.8
 BAJA POTENCIA,20-09-2019,2.8,51
 Cummins,06-01-2019,6.8,53.53
 FILTROS DE AIRE DE MOTOR DIESEL,25-10-2016,0.4,56.07
 BAJA POTENCIA,11-07-2020,8.1,63.22
 FUGA REFRIGERANTE,06-01-2022,1.4,63.25
 BAJA POTENCIA,12-10-2021,16,77.2
 BAJA POTENCIA,01-05-2022,3,77.78
 ROJA MOTOR,04-04-2017,3.5,84.05
 FALLA MOTOR DIESEL,24-06-2021,14.7,84.1
 BAJA POTENCIA,15-05-2022,1.1,95.37
 DETENCION MOTOR DIESEL,18-09-2019,5.8,95.63
 FALLA MOTOR DIESEL,11-01-2019,141.8,99.52

BAJA POTENCIA,08-10-2021,2.7,102.83
 BAJA POTENCIA,30-09-2021,4.8,108.73
 BAJA POTENCIA,18-08-2022,5.6,112.63
 SIN PARTIDA,26-01-2022,4,113.22
 BAJA RPM Y POTENCIA,08-07-2020,16,117.62
 BAJA POTENCIA,20-06-2021,3.6,120.52
 ROJA MOTOR,14-06-2020,2.4,125.32
 SIN PARTIDA,01-09-2020,1.8,125.37
 ROJA MOTOR,26-10-2021,0.8,137.33
 RPM,04-01-2019,4.3,144.68
 BAJA POTENCIA,25-09-2021,2,155.92
 BAJA POTENCIA,20-10-2021,3.6,162.5
 BAJA POTENCIA,11-05-2022,0.2,172.02
 MECANICA IMPREVISTA,03-07-2020,6.7,176.28
 FILTROS DE AIRE DE MOTOR DIESEL,14-08-2022,1,218.12
 TERMOSTATO,24-06-2020,11,255.08
 FUGA REFRIGERANTE,13-04-2021,6.2,234.17
 MECANICA IMPREVISTA,21-03-2016,5.5,236.8
 ALTERNADOR,12-12-2019,3,240.33
 BAJA POTENCIA,21-01-2022,7,254.37
 ADV_PROPULSION,26-05-2022,3.8,256.88
 BAJA POTENCIA,24-08-2021,6.1,262.17
 BAJA POTENCIA,04-08-2022,0.3,277.83
 ROJA MOTOR,27-04-2021,1.4,305.85
 BAJA POTENCIA,12-07-2021,2.9,322.53
 ROJA MOTOR,26-07-2021,3.3,335.6
 FUGA REFRIGERANTE,21-07-2022,4.7,370.93
 ADV_PROPULSION,12-10-2019,13.1,430.98
 FALLA MOTOR DIESEL,18-11-2021,2.5,462.37
 FUGA REFRIGERANTE,26-08-2020,2,488.93
 ROJA MOTOR,04-08-2020,3.6,513.95
 FUGA ACEITE MOTOR,01-03-2020,7,533.07
 RPM,10-02-2019,2.1,541.55
 ROJA MOTOR,16-12-2021,194.9,546.97
 ADV_PROPULSION,19-09-2021,5.9,586.6
 ADV_PROPULSION,07-02-2020,4,611.57
 FILTROS DE AIRE DE MOTOR DIESEL,09-01-2020,4.2,623.92
 BAJA POTENCIA,25-04-2022,6,900.68
 HUMO MOTOR DIESEL,22-10-2016,20.4,969.9
 BAJA POTENCIA,06-07-2022,1.5,978
 RPM,28-12-2018,2.9,992.83
 RPM,17-03-2022,1.1,1039.95
 FALLA DE ACTUADOR,01-12-2019,8.6,1088.55

BAJA POTENCIA,15-06-2021,3.3,1112.07
ROJA MOTOR,17-03-2017,264.1,1506.12
FUGA ACEITE MOTOR,10-01-2017,10.1,1769.28
BAJA TEMPERATURA,09-09-2019,2.3,1809.48
RPM,03-04-2021,5.6,2051.82
ROJA MOTOR,08-06-2020,18.4,2258.4
DETENCION MOTOR DIESEL,19-06-2019,4.9,2891.42
BAJA TEMPERATURA,10-09-2016,6.9,3811.28
ROJA MOTOR,12-11-2018,3.2,4014.67
FALLA MOTOR DIESEL,18-05-2018,0.3,9349.23

B. BASE DE DATOS PARA EXPORTAR A PYTHON CAEX930E3critico.CVS

FECHA,TFS,TO,TEF,fallacompcritico

20-03-2016,7.9,0,0,0
 20-03-2016,8.8,14,6,0
 21-03-2016,5.5,24,1,1
 21-03-2016,1.3,38,8,0
 22-03-2016,1.6,45,5,1
 22-03-2016,3.2,48,1,0
 22-03-2016,8.3,57,5,0
 23-03-2016,5.9,87,21,1
 24-03-2016,1.1,101,7,0
 26-03-2016,3,144,42,0
 27-03-2016,1.1,181,33,0
 29-03-2016,0.1,221,38,0
 30-03-2016,0.8,250,29,0
 31-03-2016,0.2,262,10,0
 01-04-2016,4.7,293,30,0
 02-04-2016,3.5,317,19,0
 05-04-2016,65.7,389,68,0
 05-05-2016,3.6,1100,645,0
 06-05-2016,1.4,1125,21,0
 08-05-2016,1.5,1171,45,0
 10-05-2016,50.3,1239,65,0
 16-05-2016,1.2,1364,74,0
 16-05-2016,0.3,1382,17,0
 21-05-2016,2,1499,116,0
 21-05-2016,2.7,1502,1,0
 22-05-2016,3.5,1514,8,0
 23-05-2016,7.9,1534,16,0
 26-05-2016,0.3,1622,80,0
 28-05-2016,2.7,1663,39,0
 29-05-2016,4.6,1684,18,0
 30-05-2016,3.8,1703,14,0
 30-05-2016,16.4,1710,3,0
 01-06-2016,24.8,1761,34,0
 12-06-2016,1.1,2009,223,0
 14-06-2016,0.9,2074,63,0
 18-06-2016,2.8,2160,84,0
 29-06-2016,3.1,2424,261,0
 11-07-2016,0.7,2706,278,0
 13-07-2016,0.6,2768,60,0
 15-07-2016,0.6,2800,32,0

18-07-2016,1.1,2874,72,0
21-07-2016,1.5,2956,80,0
24-07-2016,0.9,3019,61,0
24-07-2016,1.6,3021,2,0
03-08-2016,0.2,3262,238,0
04-08-2016,0.1,3286,23,0
04-08-2016,7,3286,0,0
04-08-2016,12.8,3297,3,0
05-08-2016,21.9,3323,13,0
09-08-2016,1.4,3410,64,0
29-08-2016,0.3,3886,474,0
29-08-2016,2,3896,10,0
03-09-2016,2.1,4003,104,0
10-09-2016,6.9,4174,169,1
17-09-2016,0.2,4342,160,0
06-10-2016,1.4,4796,453,0
08-10-2016,3,4853,55,0
14-10-2016,7.8,4993,137,0
19-10-2016,15.9,5113,112,0
20-10-2016,3.3,5147,17,0
21-10-2016,3.8,5160,9,0
22-10-2016,20.4,5187,23,1
25-10-2016,0.4,5263,56,1
03-11-2016,0.9,5474,210,0
07-11-2016,6.2,5578,103,0
09-11-2016,0.8,5625,41,0
23-11-2016,2.5,5965,339,0
30-11-2016,0.2,6117,149,0
07-12-2016,6,6290,172,0
07-12-2016,1.4,6297,1,0
12-12-2016,0.3,6411,112,0
13-12-2016,0.6,6435,23,0
20-12-2016,0.9,6603,167,0
21-12-2016,0.2,6633,29,0
21-12-2016,2.7,6638,4,0
22-12-2016,2.4,6649,8,0
22-12-2016,0.5,6658,6,0
23-12-2016,1.5,6686,26,0
23-12-2016,6.6,6687,0,0
27-12-2016,3.7,6761,66,0
02-01-2017,2.4,6925,160,0
03-01-2017,10.7,6931,3,0
04-01-2017,6.4,6963,20,0

04-01-2017,2.6,6970,1,0
04-01-2017,3.9,6973,0,0
05-01-2017,3,6982,5,0
05-01-2017,7.6,6985,0,0
07-01-2017,0.7,7030,37,0
07-01-2017,0.3,7042,10,0
08-01-2017,6.3,7059,16,0
09-01-2017,1.6,7093,28,0
10-01-2017,10.1,7116,20,1
19-01-2017,2.7,7332,205,0
21-01-2017,0.5,7364,29,0
26-01-2017,2.3,7496,131,0
28-01-2017,0.3,7534,35,0
01-02-2017,13.3,7639,104,0
04-02-2017,1.9,7698,44,0
05-02-2017,0.1,7721,21,0
13-02-2017,0.4,7917,196,0
18-02-2017,2.9,8043,125,0
19-02-2017,1,8067,21,0
26-02-2017,0.6,8230,162,0
01-03-2017,2.1,8306,75,0
01-03-2017,24,8309,0,0
02-03-2017,8.1,8342,9,0
08-03-2017,0.4,8467,117,0
13-03-2017,4.2,8592,124,0
13-03-2017,0.8,8601,4,0
17-03-2017,1.1,8682,80,0
17-03-2017,264.1,8699,15,1
29-03-2017,8.8,8971,8,1
29-03-2017,4.5,8992,12,1
30-03-2017,2.8,9008,12,1
30-03-2017,0.9,9014,2,0
31-03-2017,0.8,9034,19,1
04-04-2017,3.5,9119,84,1
06-04-2017,1.1,9165,42,0
09-04-2017,0.8,9241,75,0
15-04-2017,3.2,9400,157,0
17-04-2017,2.4,9445,42,0
18-04-2017,0.7,9468,20,0
19-04-2017,0.2,9487,18,0
20-04-2017,3.8,9506,18,0
22-04-2017,2.3,9553,43,0
27-04-2017,0.8,9676,120,0

29-04-2017,0.1,9722,44,0
02-05-2017,6.2,9807,85,0
06-05-2017,0.6,9898,84,0
17-05-2017,0.6,10144,245,0
17-05-2017,0.6,10160,15,0
19-05-2017,14.4,10207,46,0
22-05-2017,1.2,10286,64,0
23-05-2017,0.1,10311,24,0
26-05-2017,0.4,10383,71,0
28-05-2017,7.1,10416,32,0
29-05-2017,0.7,10448,25,0
31-05-2017,0.3,10498,48,0
03-06-2017,1.5,10574,76,0
05-06-2017,0.2,10611,35,0
09-06-2017,10.6,10714,103,0
10-06-2017,2,10735,9,0
16-06-2017,8.7,10870,133,0
16-06-2017,31.9,10880,1,0
19-06-2017,11.2,10936,23,0
19-06-2017,2.5,10948,0,0
20-06-2017,2.2,10963,11,0
22-06-2017,0.1,11026,60,0
24-06-2017,1.2,11062,35,0
26-06-2017,1,11108,45,0
01-07-2017,1.6,11245,136,0
06-07-2017,1.8,11361,114,0
09-07-2017,12.9,11420,56,0
11-07-2017,5.4,11464,31,0
11-07-2017,1.5,11474,3,0
11-07-2017,4.3,11483,8,0
15-07-2017,0.7,11582,94,0
16-07-2017,0.5,11595,12,0
20-07-2017,0.8,11697,101,0
25-07-2017,0.1,11810,111,0
27-07-2017,0.1,11848,38,0
31-07-2017,2.7,11965,116,0
01-08-2017,1.1,11977,9,0
06-08-2017,4.5,12097,118,0
10-08-2017,0.1,12188,86,0
10-08-2017,6.7,12197,8,0
10-08-2017,0.8,12206,2,0
14-08-2017,1.1,12295,88,0
17-08-2017,4.2,12364,67,0

20-08-2017,0.2,12433,64,0
22-08-2017,0.6,12481,48,0
23-08-2017,0.5,12501,19,0
23-08-2017,5.1,12502,0,0
23-08-2017,1.4,12518,10,0
24-08-2017,1.6,12520,0,0
25-08-2017,0.8,12554,31,0
03-09-2017,6.8,12771,216,0
05-09-2017,3.3,12823,45,0
09-09-2017,3.9,12907,80,0
12-09-2017,1.2,12986,75,0
20-09-2017,1,13179,191,0
22-09-2017,2,13225,45,0
22-09-2017,0.3,13232,5,0
25-09-2017,3.4,13295,62,0
29-09-2017,0.4,13401,102,0
30-09-2017,2.3,13420,18,0
01-10-2017,0.9,13446,24,0
02-10-2017,0.6,13474,27,0
03-10-2017,0.6,13493,17,0
04-10-2017,0.3,13507,13,0
05-10-2017,0.3,13541,34,0
22-10-2017,0.6,13951,409,0
23-10-2017,0.8,13976,24,0
28-10-2017,2.2,14089,111,0
29-10-2017,7.5,14112,21,0
01-11-2017,0.3,14181,60,0
01-11-2017,1.6,14188,6,0
02-11-2017,0.1,14218,29,0
08-11-2017,7.7,14364,145,0
11-11-2017,0.4,14434,62,0
06-12-2017,0.2,15031,596,0
09-12-2017,0.2,15093,62,0
10-12-2017,10.5,15131,37,0
15-12-2017,1.3,15242,100,0
16-12-2017,0.9,15274,30,0
17-12-2017,1,15285,9,0
17-12-2017,4.6,15286,0,0
31-12-2017,1.4,15618,326,0
31-12-2017,6.5,15628,9,0
02-01-2018,0.2,15682,47,0
06-01-2018,2,15769,86,0
14-01-2018,13.7,15955,184,0

22-01-2018,4.1,16157,188,0
 25-01-2018,6.3,16228,66,0
 29-01-2018,49.4,16323,88,0
 06-02-2018,2.1,16511,138,0
 08-02-2018,2.4,16560,47,0
 11-02-2018,2.4,16626,63,0
 13-02-2018,10.7,16693,65,0
 15-02-2018,0.2,16738,34,0
 16-02-2018,5.5,16763,24,0
 17-02-2018,2.4,16780,11,0
 25-02-2018,2.9,16969,186,0
 02-03-2018,0.1,17081,109,0
 04-03-2018,0.3,17143,61,0
 07-03-2018,5.8,17217,73,0
 08-03-2018,0.1,17226,3,0
 08-03-2018,4.9,17238,12,0
 09-03-2018,33.2,17265,21,0
 19-03-2018,1.7,17498,199,0
 21-03-2018,1.7,17553,53,0
 28-03-2018,1.5,17709,154,0
 29-03-2018,1,17731,21,0
 31-03-2018,8.7,17779,46,0
 04-04-2018,1.4,17882,94,0
 04-04-2018,0.4,17894,11,0
 08-04-2018,3.8,17982,87,0
 09-04-2018,1,18008,21,0
 14-04-2018,2.7,18122,112,0
 14-04-2018,0.6,18125,1,0
 15-04-2018,0.6,18148,21,0
 15-04-2018,2.2,18149,0,0
 18-04-2018,1.4,18220,69,0
 23-04-2018,2.2,18350,129,0
 24-04-2018,1.6,18356,3,0
 26-04-2018,0.1,18412,54,0
 26-04-2018,0.6,18418,5,0
 27-04-2018,1.1,18442,23,0
 02-05-2018,0.1,18550,107,0
 02-05-2018,0.1,18550,0,0
 04-05-2018,7.3,18609,58,0
 05-05-2018,5.7,18635,18,0
 06-05-2018,2.9,18642,1,0
 13-05-2018,1.3,18817,172,0
 13-05-2018,5.2,18820,1,0

15-05-2018,8.8,18862,36,0
 15-05-2018,2.2,18872,1,0
 18-05-2018,0.3,18945,70,1
 19-05-2018,5.6,18961,15,0
 25-05-2018,2.4,19105,138,0
 01-06-2018,0.4,19277,170,0
 04-06-2018,16.3,19357,79,0
 07-06-2018,3.7,19427,53,0
 13-06-2018,0.9,19566,134,0
 14-06-2018,1,19587,20,0
 15-06-2018,0.1,19613,25,0
 15-06-2018,0.4,19621,8,0
 16-06-2018,2,19634,12,0
 17-06-2018,1.4,19651,15,0
 17-06-2018,0.3,19655,2,0
 17-06-2018,0.6,19671,16,0
 23-06-2018,4.1,19801,129,0
 25-06-2018,0.3,19857,51,0
 28-06-2018,4.6,19914,57,0
 28-06-2018,5.8,19920,0,0
 30-06-2018,0,19962,36,0
 30-06-2018,2,19962,0,0
 30-06-2018,6.5,19981,17,0
 02-07-2018,1.6,20024,36,0
 02-07-2018,4.2,20027,1,0
 04-07-2018,7.4,20066,34,0
 04-07-2018,0.4,20078,5,0
 05-07-2018,0.5,20097,18,0
 05-07-2018,0.1,20103,4,0
 11-07-2018,1.8,20245,142,0
 12-07-2018,1.5,20265,18,0
 13-07-2018,0.8,20285,18,0
 14-07-2018,2.8,20305,19,0
 14-07-2018,13,20320,11,0
 16-07-2018,1.6,20354,21,0
 26-07-2018,7.3,20588,232,0
 30-07-2018,0.3,20703,107,0
 05-08-2018,2.2,20845,141,0
 06-08-2018,0.4,20861,13,0
 06-08-2018,2.4,20871,10,0
 20-08-2018,1.8,21184,310,0
 20-08-2018,3.2,21193,7,0
 23-08-2018,5.6,21262,65,0

26-08-2018,2.3,21338,70,0
27-08-2018,3.1,21374,34,0
28-08-2018,3.2,21397,19,0
02-09-2018,0.5,21497,96,0
04-09-2018,3.4,21558,60,0
06-09-2018,1.7,21608,46,0
07-09-2018,4.7,21616,5,0
11-09-2018,8.6,21727,106,0
15-09-2018,1.8,21808,72,0
17-09-2018,60.7,21864,54,0
21-09-2018,0.6,21958,33,0
21-09-2018,7.5,21970,11,0
24-09-2018,6.6,22041,63,0
26-09-2018,0.6,22082,34,0
05-10-2018,0.2,22299,216,0
06-10-2018,1.2,22315,15,0
10-10-2018,0.6,22420,103,0
11-10-2018,1.9,22444,23,0
20-10-2018,5.8,22649,203,0
25-10-2018,0.8,22789,134,0
27-10-2018,0.7,22833,43,0
27-10-2018,0.5,22839,5,0
31-10-2018,7.1,22923,83,0
04-11-2018,1.1,23014,83,0
04-11-2018,2.4,23018,2,0
04-11-2018,4.1,23022,1,0
04-11-2018,9.6,23029,2,0
08-11-2018,1.6,23125,86,0
12-11-2018,3.2,23220,93,1
16-11-2018,5.9,23304,80,0
16-11-2018,0.1,23313,3,0
19-11-2018,0.8,23380,66,0
21-11-2018,5.4,23437,56,0
24-11-2018,0,23506,63,0
24-11-2018,0.7,23506,0,0
26-11-2018,1.7,23545,37,0
28-11-2018,7.9,23590,43,0
29-11-2018,2.5,23612,13,0
30-11-2018,5.4,23640,25,0
04-12-2018,0.1,23728,83,0
04-12-2018,11.2,23748,19,0
05-12-2018,3,23760,1,0
07-12-2018,2,23816,53,0

11-12-2018,4.4,23898,79,0
11-12-2018,5.3,23903,0,0
11-12-2018,3.9,23910,1,0
12-12-2018,0.3,23931,17,0
12-12-2018,3.2,23943,11,0
13-12-2018,1.9,23955,8,0
13-12-2018,1.6,23966,8,0
15-12-2018,18.6,24015,47,0
22-12-2018,13.7,24171,137,0
24-12-2018,3.6,24222,36,0
26-12-2018,0.6,24278,52,0
27-12-2018,1.4,24301,22,0
28-12-2018,4.4,24314,11,0
28-12-2018,2.9,24326,8,1
30-12-2018,0.1,24370,41,0
01-01-2019,4.7,24409,38,0
04-01-2019,3.5,24475,61,0
04-01-2019,4.3,24482,3,1
05-01-2019,1.8,24501,15,0
05-01-2019,1.1,24506,3,0
06-01-2019,6.8,24543,35,1
11-01-2019,141.8,24649,99,1
17-01-2019,1.1,24801,9,0
19-01-2019,28.4,24844,42,0
20-01-2019,3.9,24875,2,0
21-01-2019,1.1,24883,3,0
31-01-2019,1.3,25141,257,0
01-02-2019,0.7,25155,12,0
04-02-2019,1,25217,61,0
06-02-2019,0.2,25268,50,0
07-02-2019,0.5,25306,37,0
08-02-2019,7,25319,11,0
10-02-2019,2.1,25378,52,1
23-02-2019,8,25676,296,0
25-02-2019,2.6,25737,52,0
26-02-2019,0.4,25757,17,0
27-02-2019,0.3,25769,11,0
28-02-2019,0.5,25811,41,0
04-03-2019,3.6,25889,77,0
04-03-2019,2.9,25909,16,0
09-03-2019,0.5,26011,98,0
14-03-2019,5.1,26142,130,0
15-03-2019,3.8,26159,11,0

19-03-2019,1.2,26261,98,0
22-03-2019,0.2,26332,69,0
28-03-2019,2.6,26485,152,0
29-03-2019,2.5,26503,15,0
29-03-2019,2.9,26508,2,0
30-03-2019,7.1,26517,5,0
30-03-2019,3,26535,11,0
03-04-2019,6.3,26623,84,0
04-04-2019,15.6,26641,12,0
06-04-2019,9.5,26683,26,0
09-04-2019,1,26774,80,0
16-04-2019,1.2,26936,161,0
17-04-2019,3.6,26958,21,0
26-04-2019,0.3,27177,214,0
30-04-2019,0.3,27266,88,0
05-05-2019,0.3,27378,111,0
06-05-2019,1.1,27423,44,0
08-05-2019,0.8,27469,45,0
25-05-2019,0.4,27873,403,0
26-05-2019,1.2,27885,10,0
26-05-2019,3.2,27900,14,0
27-05-2019,1.1,27916,12,0
29-05-2019,2.5,27960,42,0
29-05-2019,1.5,27964,2,0
30-05-2019,0.6,27978,12,0
30-05-2019,1.8,27986,7,0
30-05-2019,64.8,27989,1,0
02-06-2019,8.3,28068,13,0
06-06-2019,1.1,28157,81,0
08-06-2019,0.4,28215,56,0
09-06-2019,2.7,28226,10,0
14-06-2019,2.6,28346,117,0
15-06-2019,3.9,28370,21,0
15-06-2019,0.1,28377,2,0
15-06-2019,3.2,28380,3,0
17-06-2019,0.5,28426,42,0
19-06-2019,4.9,28458,31,1
21-06-2019,11.1,28506,43,0
23-06-2019,0.5,28567,49,0
25-06-2019,0.3,28600,32,0
26-06-2019,2.2,28634,33,0
26-06-2019,1.3,28645,8,0
28-06-2019,1.2,28694,47,0

02-07-2019,0.6,28790,94,0
 02-07-2019,3.8,28791,0,0
 07-07-2019,0.3,28892,97,0
 08-07-2019,8.8,28927,34,0
 10-07-2019,0.5,28970,34,0
 11-07-2019,3.6,28994,23,0
 13-07-2019,3.3,29033,35,0
 13-07-2019,4.2,29044,7,0
 17-07-2019,3.1,29148,99,0
 18-07-2019,12.2,29153,2,0
 18-07-2019,0,29170,4,0
 18-07-2019,12.1,29176,6,0
 21-07-2019,7.5,29242,54,0
 22-07-2019,0.6,29252,2,0
 23-07-2019,3.1,29274,20,0
 24-07-2019,5.4,29307,30,0
 28-07-2019,0.6,29395,82,0
 28-07-2019,2.6,29397,1,0
 31-07-2019,0.2,29486,85,0
 05-08-2019,1.5,29601,115,0
 05-08-2019,4.4,29605,2,0
 09-08-2019,0.8,29687,77,0
 11-08-2019,5.3,29740,52,0
 12-08-2019,25.8,29763,17,0
 13-08-2019,25.5,29793,3,0
 15-08-2019,4,29836,17,0
 17-08-2019,0.3,29884,44,0
 17-08-2019,5.2,29892,7,0
 20-08-2019,0.8,29950,52,0
 23-08-2019,3.8,30032,80,0
 24-08-2019,0.9,30046,10,0
 25-08-2019,0.3,30070,22,0
 09-09-2019,4.4,30439,368,0
 09-09-2019,2.3,30445,1,1
 10-09-2019,11.7,30449,2,0
 10-09-2019,70.2,30463,1,0
 13-09-2019,4.9,30543,10,1
 14-09-2019,0.6,30566,17,0
 15-09-2019,5.2,30575,8,0
 17-09-2019,0,30638,57,1
 17-09-2019,3.1,30638,0,0
 18-09-2019,5.8,30653,11,1
 18-09-2019,1.9,30662,3,0

20-09-2019,2.8,30711,47,1
21-09-2019,7.8,30716,1,1
22-09-2019,9.1,30745,21,0
26-09-2019,14.1,30853,98,0
30-09-2019,2.7,30939,71,0
30-09-2019,16.5,30949,7,0
02-10-2019,13.8,30985,19,0
02-10-2019,7.7,30999,0,0
05-10-2019,1.4,31072,64,0
06-10-2019,2.6,31081,8,0
07-10-2019,4.4,31105,21,0
08-10-2019,2.3,31132,22,0
12-10-2019,13.1,31229,95,1
13-10-2019,48,31255,13,0
20-10-2019,1.1,31418,114,0
06-11-2019,2.4,31837,418,0
07-11-2019,1.7,31841,1,0
08-11-2019,1.8,31874,30,0
09-11-2019,1.2,31900,24,0
09-11-2019,0.9,31906,4,0
11-11-2019,1.7,31947,39,0
11-11-2019,0.2,31950,1,0
14-11-2019,0.3,32026,75,0
15-11-2019,0.5,32050,22,0
16-11-2019,0.6,32073,23,0
16-11-2019,8.4,32077,3,0
19-11-2019,4.7,32151,64,0
20-11-2019,5.8,32162,6,0
25-11-2019,1.6,32281,113,0
26-11-2019,1.6,32309,25,0
27-11-2019,4.5,32333,23,0
30-11-2019,0.9,32393,54,0
30-11-2019,0.8,32395,1,0
01-12-2019,8.6,32419,23,1
02-12-2019,5,32459,30,0
05-12-2019,5.2,32533,69,0
09-12-2019,2.1,32617,78,0
10-12-2019,1.1,32633,14,0
12-12-2019,3,32682,47,1
16-12-2019,0.4,32799,114,0
18-12-2019,8.5,32843,43,0
19-12-2019,12.6,32853,1,0
19-12-2019,12.6,32868,2,0

27-12-2019,0.7,33058,176,0
 28-12-2019,0.5,33083,24,0
 29-12-2019,2.4,33111,27,0
 30-12-2019,3.4,33133,19,0
 05-01-2020,5.2,33257,120,0
 09-01-2020,4.2,33355,92,1
 10-01-2020,1,33392,32,0
 11-01-2020,0.1,33409,16,0
 13-01-2020,2.7,33462,53,0
 14-01-2020,9.7,33477,12,0
 15-01-2020,1.4,33499,12,0
 16-01-2020,2.9,33540,39,0
 25-01-2020,8.8,33752,209,0
 27-01-2020,1,33785,24,0
 27-01-2020,0.8,33794,8,0
 27-01-2020,0.2,33795,0,0
 31-01-2020,9.6,33890,95,0
 31-01-2020,2.1,33902,2,0
 01-02-2020,0.6,33922,17,0
 03-02-2020,37.6,33964,41,0
 07-02-2020,4,34049,47,1
 11-02-2020,7.8,34151,97,0
 11-02-2020,0.5,34168,9,0
 12-02-2020,0.1,34175,6,0
 14-02-2020,2.8,34227,52,0
 17-02-2020,1.7,34305,75,0
 19-02-2020,0.4,34342,34,0
 23-02-2020,8.5,34436,93,0
 23-02-2020,3,34448,3,0
 23-02-2020,5.3,34452,0,0
 01-03-2020,7,34616,159,1
 01-03-2020,0.4,34624,0,0
 02-03-2020,2.9,34627,2,0
 04-03-2020,1.8,34688,58,0
 05-03-2020,2,34720,29,0
 07-03-2020,3.3,34746,24,0
 07-03-2020,1.6,34753,3,0
 11-03-2020,2.1,34855,100,0
 11-03-2020,4.7,34858,0,0
 12-03-2020,1.4,34880,16,0
 13-03-2020,0.4,34898,16,0
 17-03-2020,4.6,34997,98,0
 18-03-2020,3.7,35009,7,0

30-03-2020,0.1,35301,287,0
 30-03-2020,1.7,35310,8,0
 31-03-2020,11.1,35334,23,0
 01-04-2020,16.5,35347,1,0
 11-04-2020,1.6,35596,232,0
 11-04-2020,0.4,35602,4,0
 13-04-2020,6.1,35642,39,0
 14-04-2020,0.3,35660,11,0
 16-04-2020,1.1,35727,66,0
 23-04-2020,1.2,35890,161,0
 24-04-2020,11.6,35915,23,0
 27-04-2020,8.8,35989,62,0
 02-05-2020,0.4,36111,113,0
 04-05-2020,2.6,36137,24,0
 07-05-2020,0.4,36225,86,0
 12-05-2020,0.7,36328,102,0
 12-05-2020,4.1,36330,0,0
 16-05-2020,1.9,36446,112,0
 22-05-2020,1.8,36585,136,0
 28-05-2020,2.2,36734,147,0
 31-05-2020,1.3,36797,60,0
 02-06-2020,0.5,36848,49,0
 04-06-2020,1.4,36889,40,0
 06-06-2020,0.4,36950,59,0
 07-06-2020,0.2,36975,24,0
 08-06-2020,2.2,36977,1,0
 08-06-2020,18.4,36993,13,1
 14-06-2020,2.4,37137,125,1
 15-06-2020,14.7,37160,20,0
 24-06-2020,11,37379,204,1
 26-06-2020,17,37410,20,0
 03-07-2020,6.7,37583,156,1
 08-07-2020,16,37707,117,1
 11-07-2020,8.1,37787,63,1
 12-07-2020,8.3,37799,4,1
 14-07-2020,12.5,37841,33,0
 15-07-2020,0.7,37887,33,0
 18-07-2020,20.7,37952,64,0
 19-07-2020,0.9,37978,4,0
 24-07-2020,4.8,38093,114,0
 28-07-2020,2.8,38192,93,0
 01-08-2020,0.4,38291,96,0
 04-08-2020,3.6,38364,72,1

06-08-2020,4.9,38401,33,0
 16-08-2020,0.5,38642,235,0
 16-08-2020,3.5,38654,11,0
 20-08-2020,2.2,38748,90,0
 22-08-2020,11.9,38789,39,0
 25-08-2020,0.1,38852,51,0
 26-08-2020,2.2,38873,20,0
 26-08-2020,2,38882,6,1
 26-08-2020,12.1,38887,2,0
 29-08-2020,0.8,38947,48,0
 30-08-2020,0.3,38978,29,0
 30-08-2020,0.4,38980,1,0
 31-08-2020,2.1,39001,20,0
 01-09-2020,1.8,39025,21,1
 01-09-2020,8.6,39031,4,0
 09-09-2020,0.2,39219,179,0
 09-09-2020,1,39221,2,0
 14-09-2020,5.2,39338,115,0
 16-09-2020,0.2,39397,53,0
 17-09-2020,0.7,39422,24,0
 26-09-2020,1.1,39631,207,0
 01-10-2020,1.1,39745,113,0
 04-10-2020,2.7,39810,63,0
 11-10-2020,5.2,39978,166,0
 15-10-2020,1.3,40081,97,0
 15-10-2020,1.8,40084,1,0
 15-10-2020,9.5,40093,7,0
 16-10-2020,4.9,40109,6,0
 16-10-2020,3.8,40117,2,0
 17-10-2020,11,40121,0,0
 17-10-2020,3.4,40135,2,0
 17-10-2020,8.1,40141,2,0
 18-10-2020,1.6,40157,7,0
 19-10-2020,7.4,40174,16,0
 20-10-2020,4,40194,12,0
 20-10-2020,8.2,40208,10,0
 21-10-2020,24.1,40221,4,0
 22-10-2020,48.3,40247,1,0
 24-10-2020,10.6,40296,1,0
 24-10-2020,8,40308,0,0
 25-10-2020,1.1,40321,4,0
 25-10-2020,19.9,40329,7,0
 12-11-2020,5.9,40751,401,0

12-11-2020,0.3,40760,3,0
14-11-2020,9.4,40793,32,0
14-11-2020,9.8,40813,11,0
16-11-2020,0.1,40845,21,0
23-11-2020,0.3,41014,169,0
27-11-2020,1.4,41107,91,0
02-12-2020,0.3,41234,125,0
09-12-2020,10.3,41413,178,0
10-12-2020,0.4,41424,0,0
27-12-2020,0.6,41835,410,0
01-01-2021,1.9,41947,111,0
04-01-2021,6.1,42016,67,1
08-01-2021,1.5,42127,104,0
10-01-2021,11.2,42175,46,0
18-01-2021,0.4,42367,180,0
22-01-2021,1.6,42470,102,0
23-01-2021,3.7,42487,15,0
23-01-2021,3,42493,2,0
24-01-2021,3.2,42505,9,0
02-02-2021,1,42715,206,0
02-02-2021,1,42718,2,0
03-02-2021,2.7,42745,25,0
04-02-2021,1.7,42784,35,0
05-02-2021,2.8,42786,0,0
05-02-2021,4.1,42791,2,0
07-02-2021,0.3,42847,51,0
08-02-2021,5.2,42868,20,0
13-02-2021,0.9,42999,125,0
17-02-2021,1.3,43076,75,0
21-02-2021,0.3,43173,95,0
02-03-2021,4.6,43406,232,0
03-03-2021,0.8,43417,7,0
03-03-2021,1.9,43421,2,0
03-03-2021,2.2,43429,6,0
04-03-2021,1.9,43434,2,0
05-03-2021,4.3,43477,41,0
06-03-2021,1.1,43489,7,0
23-03-2021,1.9,43897,406,0
23-03-2021,0.2,43903,4,0
26-03-2021,21.4,43981,77,0
03-04-2021,5.6,44166,163,1
05-04-2021,3.9,44213,42,0
09-04-2021,0.4,44312,95,0

10-04-2021,0.3,44343,29,0
 13-04-2021,0.1,44397,54,0
 13-04-2021,6.2,44410,12,1
 16-04-2021,1.2,44477,60,0
 19-04-2021,3.3,44557,78,0
 20-04-2021,4,44561,0,0
 25-04-2021,2,44683,117,0
 27-04-2021,1.4,44733,47,1
 02-05-2021,4.8,44867,132,0
 05-05-2021,0.4,44929,57,0
 06-05-2021,17.3,44953,23,0
 08-05-2021,2.5,45012,41,0
 09-05-2021,0.9,45032,17,0
 10-05-2021,2.3,45059,26,0
 11-05-2021,1.7,45074,13,0
 11-05-2021,2.1,45077,0,0
 11-05-2021,4.9,45085,5,0
 16-05-2021,1.1,45196,106,0
 16-05-2021,4.6,45198,0,0
 17-05-2021,3.5,45210,7,0
 03-06-2021,3.4,45638,424,0
 05-06-2021,0.2,45685,43,0
 06-06-2021,0.3,45689,4,0
 06-06-2021,1.3,45693,2,0
 11-06-2021,7.2,45815,121,0
 12-06-2021,2.3,45841,18,0
 15-06-2021,3.3,45907,63,1
 19-06-2021,2.6,46023,112,0
 20-06-2021,3.6,46033,7,1
 24-06-2021,14.7,46121,84,1
 24-06-2021,6,46138,1,1
 26-06-2021,41.3,46187,43,1
 02-07-2021,1.1,46323,94,0
 02-07-2021,1.2,46325,1,0
 05-07-2021,0.9,46394,67,0
 11-07-2021,3.7,46537,142,0
 12-07-2021,2.9,46558,17,1
 18-07-2021,2,46717,155,0
 20-07-2021,0.5,46755,36,0
 26-07-2021,3.3,46902,146,1
 31-07-2021,7.3,47026,120,0
 02-08-2021,1.7,47065,31,0
 09-08-2021,1.3,47228,161,0

10-08-2021,1.1,47269,39,0
12-08-2021,4.6,47306,35,0
13-08-2021,3.5,47323,12,1
13-08-2021,10,47329,2,0
17-08-2021,1.4,47417,77,0
19-08-2021,0.9,47482,64,0
23-08-2021,2.8,47580,96,0
24-08-2021,6.1,47604,21,1
25-08-2021,0.4,47627,16,0
27-08-2021,0.3,47674,46,0
29-08-2021,0.2,47716,41,0
30-08-2021,1.1,47751,35,0
07-09-2021,1.2,47943,190,0
09-09-2021,0.5,47979,35,0
10-09-2021,0.3,47999,18,0
10-09-2021,1,48001,2,0
10-09-2021,1.3,48013,11,0
10-09-2021,0.6,48016,1,0
11-09-2021,0.7,48022,5,0
12-09-2021,1.8,48044,21,0
12-09-2021,0.1,48046,0,0
12-09-2021,0.4,48046,0,0
12-09-2021,0.2,48064,17,0
13-09-2021,1.2,48082,18,0
19-09-2021,5.9,48208,124,1
23-09-2021,4.1,48311,97,0
25-09-2021,2,48374,58,1
27-09-2021,0.8,48421,45,0
29-09-2021,7.9,48464,42,0
30-09-2021,4.8,48493,21,1
02-10-2021,7.5,48541,43,1
03-10-2021,6.9,48565,16,1
04-10-2021,1.7,48585,12,0
05-10-2021,0.8,48604,18,0
06-10-2021,2.7,48625,20,0
06-10-2021,3.1,48638,9,0
07-10-2021,0.7,48653,11,0
08-10-2021,2.7,48684,30,1
09-10-2021,0.8,48707,19,0
10-10-2021,2.3,48716,8,0
10-10-2021,4.5,48723,5,0
11-10-2021,1.8,48757,29,0
11-10-2021,0.3,48759,0,0

12-10-2021,16,48774,13,1
 18-10-2021,3.2,48905,115,0
 20-10-2021,3.6,48955,47,1
 20-10-2021,7.7,48963,4,0
 23-10-2021,4,49032,61,0
 24-10-2021,3.5,49057,21,0
 26-10-2021,0.8,49111,50,1
 27-10-2021,1.3,49130,17,0
 28-10-2021,0.8,49150,18,1
 29-10-2021,0.8,49185,34,0
 29-10-2021,11.5,49186,0,0
 30-10-2021,6.7,49198,0,0
 31-10-2021,0.8,49225,20,0
 01-11-2021,13,49254,28,0
 10-11-2021,0.4,49467,199,0
 10-11-2021,0.5,49468,0,0
 11-11-2021,1.4,49485,16,0
 11-11-2021,3.8,49487,0,0
 11-11-2021,4.4,49498,7,0
 13-11-2021,0.3,49545,42,0
 13-11-2021,8.1,49546,0,0
 18-11-2021,2.5,49665,110,1
 20-11-2021,11,49715,47,1
 23-11-2021,0,49781,54,0
 25-11-2021,3.1,49837,56,0
 28-11-2021,2.2,49895,54,0
 01-12-2021,0.7,49970,73,0
 03-12-2021,4.1,50029,58,0
 04-12-2021,10.4,50046,12,0
 05-12-2021,10.7,50057,0,0
 06-12-2021,13.4,50080,12,0
 09-12-2021,0.8,50173,79,0
 10-12-2021,8.6,50185,11,0
 16-12-2021,194.9,50327,133,1
 24-12-2021,245.8,50524,1,1
 06-01-2022,1.4,50833,63,1
 08-01-2022,1.7,50876,41,1
 08-01-2022,3.5,50888,10,0
 09-01-2022,2.4,50916,25,1
 12-01-2022,5.3,50969,50,0
 12-01-2022,4.7,50982,7,0
 12-01-2022,1.5,50989,2,0
 15-01-2022,4.6,51051,60,0

17-01-2022,1.7,51093,37,0
21-01-2022,7,51191,96,1
24-01-2022,2.5,51279,80,0
26-01-2022,4,51314,32,1
26-01-2022,24.1,51321,3,1
28-01-2022,6.7,51352,7,1
29-01-2022,1.3,51376,17,1
29-01-2022,6.5,51382,5,1
30-01-2022,24.1,51401,11,1
06-02-2022,2.6,51578,153,0
08-02-2022,8.2,51625,44,0
11-02-2022,0.7,51696,63,0
11-02-2022,7.8,51697,0,0
11-02-2022,4.1,51708,2,0
12-02-2022,0.8,51728,15,0
19-02-2022,0.3,51902,173,0
20-02-2022,0.8,51913,11,0
26-02-2022,0.4,52070,155,0
08-03-2022,14.2,52293,222,0
09-03-2022,1.4,52312,4,0
09-03-2022,4.5,52314,0,0
09-03-2022,5.7,52321,2,0
10-03-2022,3.7,52355,28,0
13-03-2022,0.3,52429,70,0
14-03-2022,0.9,52444,14,0
17-03-2022,1.1,52521,76,1
19-03-2022,0.2,52574,51,0
25-03-2022,0.5,52709,134,0
30-03-2022,0.2,52838,128,0
06-04-2022,1.5,52998,160,0
09-04-2022,1,53077,77,0
16-04-2022,1.1,53245,166,0
20-04-2022,1,53330,83,0
21-04-2022,12.3,53351,20,0
25-04-2022,6,53441,77,1
26-04-2022,0.8,53466,19,1
26-04-2022,10.8,53484,17,0
27-04-2022,2.6,53501,5,1
27-04-2022,20.8,53505,1,0
30-04-2022,0.2,53570,43,0
01-05-2022,3,53602,32,1
02-05-2022,52.6,53620,14,0
11-05-2022,0.2,53830,157,1

11-05-2022,2.5,53834,3,0
15-05-2022,1.1,53928,91,1
15-05-2022,0.9,53931,1,1
17-05-2022,1.8,53972,40,0
21-05-2022,0.9,54079,105,0
26-05-2022,3.8,54191,111,1
06-07-2022,1.5,55173,977,1
07-07-2022,0.8,55215,40,0
21-07-2022,4.7,55546,330,1
30-07-2022,0.7,55750,198,0
30-07-2022,1.8,55767,16,0
31-07-2022,44.6,55778,9,0
04-08-2022,0.3,55876,53,1
08-08-2022,2.2,55965,88,0
08-08-2022,8.6,55979,11,0
14-08-2022,1,56105,117,1
18-08-2022,0.3,56218,111,0
18-08-2022,5.6,56219,1,1

ANEXO 3

A. CÓDIGO DISTRIBUCIÓN DE WEIBULL

```

from typing import List, Any

import numpy as np
import statistics
import matplotlib.pyplot as plt
import scipy.special
from sklearn.metrics import mean_squared_error
from sklearn.metrics import confusion_matrix
from sklearn.metrics import f1_score
from sklearn.metrics import classification_report
import statsmodels.api as sm
import seaborn as sns
import numpy as np
import pandas as pd
def fweibull(a, b, t):
    f = (b / a) * ((t / a) ** (b - 1)) * np.exp(-(t / a) ** b)
    return f
def Fweibull(a, b, t):
    F = 1 - np.exp(-(t / a) ** b)
    return F
def TFweibull(a, b, t):
    TF = (fweibull(a, b, t)) / (1 - Fweibull(a, b, t))
    return TF
# Importar Datos:
df = pd.read_csv ( 'CAEX930E3WB.csv' ).sort_values(by='TEF')
df.head ( )
print ( df.head ( ) )
# Estimación de Parámetros Dist.Weibull:
n = len(df) # Numero muestral
df['kj'] = range(1,n+1) # SIN ranking ajustado
df['Ft_b'] = (df['kj']+0.3)/(n+0.4) # F(t): Rango Mediana de Benard
df['Rt_b'] = 1-df['Ft_b'] # R(t): Confiabilidad
# Listas con la Info Necesaria:
kj = df['kj'].values.tolist()
Ti_l = df['TEF'].values.tolist()
Rt_b = df['Rt_b'].values.tolist()
# Ejes del ploteo
Ejex = [np.log(Ti_l[i]) for i in range(len(kj))]
Ejey = [np.log(-(np.log(Rt_b[i]))) for i in range(len(kj))]

```

```

# Pendiente e Intersección: Método Mínimos Cuadrados
nr = len(Ejex)
sumx,sumy = sum(Ejex), sum(Ejey)
XY = [Ejex[i]*Ejey[i] for i in range(nr)]
X2 = [Ejex[i]**2 for i in range(nr)]
Y2 = [Ejey[i]**2 for i in range(nr)]
sumxy, sumx2 = sum(XY), sum(X2)
# Cuadrado del Coeficiente de correlación de Pearson (rxy)^2:
destx = statistics.stdev(Ejex)
desty = statistics.stdev(Ejey)
rxy = (sumxy-nr*(sumx/nr)*(sumy/nr))/((nr-1)*destx*desty)
R2 = rxy**2
# Intersección (A) y Pendiente (B):#
B = (nr * sumxy - sumx * sumy) / (nr * sumx2 - (sumx ) ** 2)
A = (sumy - B * sumx ) / nr
print('RESULTADOS DEL AJUSTE WEIBULL')
print('La ecuación de la Recta es: ',B,'X +' ,A)
print('Coeficiente de Determinación R^2: ',R2)
Ec = [A+B*Ejex[i] for i in range(len(Ejex))]
# Estimación de Parámetros Weibull:
Alpha = np.exp(-A / B)
Beta = B
MTBF= Alpha*scipy.special.gamma(1+(1/Beta))
print('El valor de Alpha es: ',Alpha)
print('El valor de Beta es: ', Beta)
print('El valor de MTBF es: ', MTBF)
# Ploteo del Grafico Dispersion Weibull:
plt.scatter(Ejex,Ejey,alpha=0.3,label='Datos',c='orange')
plt.plot(Ejex,Ec,label='Ajuste Lineal')
plt.title('Gráfico de Dispersión Weibull')
plt.xlabel('Ln(ti)')
plt.ylabel('Ln(-Ln(R(ti)))')
plt.legend()
plt.show()
plt.clf()
Equis = np.arange(1, 700)
plt.title('Tasa de Fallas y Confiabilidad de los datos reales')
plt.plot(Equis, TFweibull(Alpha, Beta, Equis), label='Tasa de Fallas')
plt.ylabel('Tasa de Fallas')
plt.twinx()
plt.plot(Equis, (1 - Fweibull(Alpha, Beta, Equis)) * 100, 'orange',
label='Confiabilidad')
plt.ylabel('Confiabilidad')

```

```

plt.legend(('C'), loc='best')
plt.show()
plt.clf()

#Creación de números aleatorios:
from numpy import random
datos = np.random.rand(91)
alpha = Alpha
beta = Beta
y_predicted: list[Any] = (list(alpha * (-np.log(1 - datos)) ** (1 / beta)))
print(y_predicted)
df2 = pd.read_csv('CAEX930E3WB.csv')
y_actual = df2['TEF'].values.tolist()
print(y_actual)
# Print de los errores
from sklearn import metrics
import seaborn as sns
print('MAE:', metrics.mean_absolute_error(y_actual, y_predicted))
print('MSE:', metrics.mean_squared_error(y_actual, y_predicted))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_actual, y_predicted)))
print('VarScore:', metrics.explained_variance_score(y_actual, y_predicted))
y_predicted.sort()

# Análisis Exploratorio de Datos:
plt.title('Histograma de datos simulados')
plt.hist( y_predicted, bins=18, color="blue", alpha=0.5 )
plt.ylabel('Frecuencia Absoluta')
plt.xlabel('Tiempo (ti)')
plt.xlim(0,1000)
# plt.ylim(0,250)
plt.show()
plt.clf()

# Predicciones perfectas en linea roja
plt.title('TEF real v/s TEF Predicho')
sns.regplot(x=y_actual, y=y_predicted,
            line_kws = {"color": "red"})
plt.xlabel('TEF Real')
plt.ylabel('TEF Predicho')
plt.show()

```

B. *CÓDIGO MODELO 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
from sklearn.metrics import confusion_matrix
import statsmodels.api as sm
import statsmodels.formula.api as smf
from statsmodels.stats.weightstats import ttest_ind
from sklearn.metrics import classification_report
from sklearn.metrics import f1_score
# 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')
datos = pd.read_csv ( 'CAEX930E3critico.csv' )
datos.head ( )
datos2 = pd.DataFrame({'fallacompcritico': datos.fallacompcritico, 'TO': datos.TO})
datos2.head(3)
print(datos2)
print ( 'Time period start: {} \n Time period end: {}'.format ( datos.FECHA.min ( ),
datos.FECHA.max ( ) ) )
# Número de obsercaciones por clase
#
=====
=====
print(datos2.fallacompcritico.value_counts().sort_index())

# Gráfico
#
=====
=====
fig, ax = plt.subplots(figsize=(6, 3.84))

sns.violinplot(
    x    = 'fallacompcritico',
    y    = 'TO',
    data = datos,
    color = "blue",
    ax    = ax, palette="Set2"
)

ax.set_title('Distribución Falla Componente crítico por clase');
plt.show ( )
# T-test entre clases
#
=====
=====
res_ttest = ttest_ind(
    x1 = datos2.TO[datos.fallacompcritico == 0],
    x2 = datos2.TO[datos.fallacompcritico == 1],
    alternative='two-sided'
)
print(f"t={res_ttest[0]}, p-value={res_ttest[1]}")
# División de los datos en train y test

```

```

#
=====
=====
X = datos2[['TO']]
y = datos['fallacompcritico']

X_train, X_test, y_train, y_test = train_test_split(
    X.values.reshape(-1,1),
    y.values.reshape(-1,1),
    train_size = 0.8,
    random_state = 1234,
    shuffle = True
)

# Creación del modelo utilizando matrices como en scikitlearn
#
=====
=====
# A la matriz de predictores se le tiene que añadir una columna de 1s para el
intercept del modelo
X_train = sm.add_constant(X_train, prepend=True)
modelo = sm.Logit(endog=y_train, exog=X_train,)
modelo = modelo.fit()
print(modelo.summary())
# Intervalos de confianza para los coeficientes del modelo
#
=====
=====
intervalos_ci = modelo.conf_int(alpha=0.05)
intervalos_ci = pd.DataFrame(intervalos_ci)
intervalos_ci.columns = ['2.5%', '97.5%']
print(intervalos_ci)

# Predicción de probabilidades
#
=====
=====
predicciones = modelo.predict(exog = X_train)
print(predicciones[:50])

# Clasificación predicha

```

```

#
=====
=====
clasificacion = np.where(predicciones<0.5, 0, 1)
print(clasificacion)

# Predicciones en todo el rango de X
#
=====
=====
# Se crea un vector con nuevos valores interpolados en el rango de observaciones.
grid_X = np.linspace(
    start = min(datos.TO),
    stop  = max(datos.TO),
    num   = 200
).reshape(-1,1)

grid_X = sm.add_constant(grid_X, prepend=True)
predicciones = modelo.predict(exog = grid_X)
# Gráfico del modelo
#
=====
=====
fig1, ax = plt.subplots(figsize=(6, 3.84))

ax.scatter(
    X_train[(y_train == 1).flatten(), 1],
    y_train[(y_train == 1).flatten()].flatten()
)
ax.scatter(
    X_train[(y_train == 0).flatten(), 1],
    y_train[(y_train == 0).flatten()].flatten()
)
ax.plot(grid_X[:, 1], predicciones, color = "gray")
ax.set_title("Modelo Regresión Logística")
ax.set_ylabel("P(fallacompcritico = 1 | TO)")
ax.set_xlabel("TIEMPO EN FUNCIONAMIENTO")
plt.show()

# Accuracy de test del modelo
#
=====
=====

```

```

X_test = sm.add_constant(X_test, prepend=True)
predicciones = modelo.predict(exog = X_test)
clasificacion = np.where(predicciones<0.5, 0, 1)
accuracy = accuracy_score(
    y_true = y_test,
    y_pred = clasificacion,
    normalize = True
)
print("")

print(f"El accuracy de test es: {100*accuracy}%")

# DETERMINACION DE MEDIDAS DE CALIDAD#
cutoff = 0.5 # decide on a cutoff limit
y_pred_classes = np.zeros_like(predicciones) # initialise a matrix full with zeros
y_pred_classes[predicciones > cutoff] = 1 # add a 1 if the cutoff was breached

y_test_classes = np.zeros_like(y_test)
y_test_classes[y_test > cutoff] = 1
print("Matriz de confusión")
print(confusion_matrix(y_test_classes, y_pred_classes))

print(classification_report(y_test_classes, y_pred_classes))
F1= f1_score(y_test_classes, y_pred_classes ,average='weighted')
print( " F1 es:", F1*100, "%")

```

C. CÓDIGO MODELO MÁQUINA DE VECTORES DE SOPORTE

```

import pandas as pd
import numpy as np
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, recall_score, f1_score,
classification_report, confusion_matrix
import statsmodels.api as sm
# Gráficos
#
=====

import matplotlib.pyplot as plt
from matplotlib import style
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
from sklearn import svm, datasets
from sklearn.svm import LinearSVC
import sklearn.svm
#
=====

import warnings
warnings.filterwarnings('ignore')

datos = pd.read_csv ('CAEX930E3critico.csv' )
datos.head ( )
datos2 = pd.DataFrame({'fallacompcritico': datos.fallacompcritico, 'TO': datos.TO,
'TFS': datos.TFS})
datos2.head(3)
print(datos2)
print ( 'Time period start: {} \nTime period end: {}'.format ( datos.FECHA.min ( ),
datos.FECHA.max ( ) ) )

# Número de observaciones por clase
#
=====

print(datos2.fallacompcritico.value_counts().sort_index())

fig, ax = plt.subplots(figsize=(6,4))

```

```

ax.scatter(datos.TO, datos.TFS, c=datos.fallacompcritico);
ax.set_title("Fallas de equipo crítico");
plt.xlabel('Tiempo de operación')
plt.ylabel('Tiempo Fuera de servicio')
plt.show()
plt.clf()

# Separar las características y la variable objetivo
X = datos2[['TO']]
y = datos2['fallacompcritico']

# Dividir los datos en conjuntos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=123)

# Crear un objeto SVM con kernel gaussiano, gamma=0.1 y C=1
svm = SVC(kernel="rbf", gamma=0.1, C=1)

# Entrenar el modelo SVM con los datos de entrenamiento
svm.fit(X_train, y_train)

# Predecir las etiquetas de los datos de prueba
y_pred = svm.predict(X_test)
print(y_pred)

# Generar una malla de puntos
x_min, x_max = X.iloc[:, 0].min() - 1, X.iloc[:, 0].max() + 1
y_min, y_max = X.iloc[:, 1].min(), X.iloc[:, 1].max() + 1
h = (x_max / x_min) / 100
xx, yy = np.meshgrid(np.arange(x_min, x_max, h), np.arange(y_min, y_max, h))

# Create the model
svc = LinearSVC()
# Fit the model
svc.fit(X.iloc[:, 0:1], y)

# Evaluar el rendimiento del modelo con diferentes métricas
acc = accuracy_score(y_test, y_pred)
rec = recall_score(y_test, y_pred, average="macro")

```

```
f1 = f1_score(y_test, y_pred, average="macro")  
# Imprimir los resultados  
print(f"El accuracy de test es: {100*acc}%")
```

```
print(classification_report(y_test, y_pred))  
F1= f1_score(y_test, y_pred ,average="macro")  
print( " F1 es:", F1*100, "%")
```

D. CÓDIGO MODELO RANDOM FOREST

Tratamiento de datos

```
#
=====

import numpy as np
import pandas as pd

# Gráficos
#
=====

import matplotlib.pyplot as plt

# Preprocesado y modelado
#
=====

from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
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
import seaborn as sns
import statsmodels.api as sm
# Configuración warnings
#
=====

import warnings

df = pd.read_csv ( 'CAEX930E3critico.csv' ).sort_values(by='TEF')
df.head()
df.info()

# División de los datos en entrenamiento y test
```

```

#
=====
# Dividir los datos en conjuntos de entrenamiento y prueba
X = df[['TO']]
y = df['TEF']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=123)

# Inicializar y entrenar el modelo Random Forest
rf_model = RandomForestRegressor(n_estimators=85, random_state=123)
rf_model.fit(X_train, y_train)

#70 Error de test del modelo inicial
#
=====
predicciones = rf_model.predict(X = X_test)

rmse = mean_squared_error(
    y_true = y_test,
    y_pred = predicciones,
    squared = False)
print(f"El error (RMSE) de test es: {rmse}")

#Optimización de hiperparámetros#
# 71 Validación empleando el Out-of-Bag error
#
=====
warnings.filterwarnings ( 'ignore' )
train_scores = []
oob_scores = []

# Valores evaluados
estimator_range = range ( 10, 150, 5 )

# Bucle para entrenar un modelo con cada valor de n_estimators y extraer su error
# de entrenamiento y de Out-of-Bag.
for n_estimators in estimator_range:
    modelo = RandomForestRegressor (
        n_estimators=n_estimators,

```

```

        criterion='squared_error',
        max_depth=None,
        max_features=1,
        oob_score=True,
        n_jobs=-1,
        random_state=123)
modelo.fit ( X_train, y_train )
train_scores.append ( modelo.score ( X_train, y_train ) )
oob_scores.append ( modelo.oob_score_ )

# Gráfico con la evolución de los errores
fig, ax = plt.subplots ( figsize=(5, 3) )
ax.plot ( estimator_range, train_scores, label="train scores" )
ax.plot ( estimator_range, oob_scores, label="out-of-bag scores" )
ax.plot ( estimator_range[np.argmax ( oob_scores )], max ( oob_scores ),
        marker='o', color="red", label="max score" )
ax.set_ylabel ( "R^2" )
ax.set_xlabel ( "n_estimators" )
ax.set_title ( "Evolución del out-of-bag-error vs número árboles" )
plt.legend ( );
print ( f"Valor óptimo de n_estimators: {estimator_range[np.argmax ( oob_scores
)]}" )
warnings.filterwarnings ( 'default' )
plt.show()

# Realizar predicciones en el conjunto de prueba
y_pred = rf_model.predict(X_test)

# Análisis Exploratorio de Datos:
plt.title('Histograma de datos simulados')
plt.hist( y_pred, bins=10, color="blue", alpha=0.5 )
plt.ylabel('Frecuencia Absoluta')
plt.xlabel('Tiempo (ti)')
plt.xlim(0,400)
plt.ylim(0,100)
plt.show()
plt.clf()

# Predicciones perfectas en linea roja

sns.set(style="darkgrid")

tips = sns.load_dataset("tips")

```

```

plt.title('Falla real v/s Falla Predicha')
sns.regplot(x=y_test, y=y_pred, scatter_kws={"color": "blue"},
            line_kws = {"color": "red"}, )
plt.xlabel('t(i)')
plt.ylabel('falla predicha')
plt.show()
print( "valores reales ", y_test)
print(y_pred)

# Print de los errores
from sklearn import metrics
import seaborn as sns
print('MAE:', metrics.mean_absolute_error(y_test, y_pred))
print('MSE:', metrics.mean_squared_error(y_test, y_pred))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
print('VarScore:', metrics.explained_variance_score(y_test, y_pred))
y_pred.sort()

```