

Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación
Maestría en Ciencias de la Computación

**Detección de Patrones de Fallas de
Automóviles Basada en Técnicas de Aprendizaje
automático y Bases de Conocimiento**

Septiembre 2024

Tesis para obtener el grado de
Maestro en Ciencias de la Computación

Presenta

Antonio Pérez Vázquez

Director

Dr Mario Anzures García

Co director

Dr Pedro Bello López

Resumen

El siguiente trabajo de tesis de maestría propone un enfoque para el diagnóstico de fallas en motores de vehículos utilizando un árbol de decisión, bases de conocimiento, bases de datos y lógica difusa. El objetivo es identificar patrones que determinen la causa de comportamientos irregulares en motores y establecer un sistema práctico en talleres mecánicos. El estudio utiliza técnicas de aprendizaje automático, en especial árboles de decisión, para representar el diagnóstico de fallas de manera ramificada según las características o condiciones de manejo. Las bases de conocimiento y bases de datos con códigos OBDII alimentan el diagnóstico. El árbol de decisión se poda para mejorar la precisión, y la propuesta se valida con diversas métricas de evaluación para obtener un diagnóstico preciso en el ámbito automotriz.

Agradecimientos

A Dios por acompañarme y guiarme a lo largo de mi vida

A mi madre padres por el amor constante que me han brindado a lo largo de mi educación.

A mi padre por el ejemplo de conducta ante la vida en situaciones adversas.

Por ser un excelente ejemplo de vida a seguir. Gracias siempre.

A mis amigos, por sus aportaciones, vivencias, experiencias y consejos.

A mis asesores de tesis. Dr. Mario Anzures García, Dr. Pedro Bello López, por brindarme tiempo, su dedicación, interés y consejos, por hacer esto posible. Gracias por todo.

A mis alumnos de IMCC que me han animado a mejorar como docente y persona.

Contenido

| | |
|---|----|
| Capítulo 1. Introducción..... | 1 |
| 1.1 Problema | 2 |
| 1.2 Objetivos..... | 2 |
| 1.3 Propuesta | 3 |
| Capítulo 2. Marco teórico..... | 5 |
| 2.1 Aprendizaje automático..... | 5 |
| 2.1.1 Modelos de aprendizaje automático | 6 |
| 2.2 Árboles de decisión..... | 7 |
| 2.2.1 Recorrido del árbol | 8 |
| 2.2.2 Entropía..... | 9 |
| 2.2.3 Gini..... | 10 |
| 2.2.4 Poda de árboles | 10 |
| 2.2.5 Métricas..... | 11 |
| 2.3 Sistemas expertos..... | 16 |
| 2.4 Motor de inferencia | 16 |
| 2.5 Base de datos | 17 |
| 2.6 Base de conocimiento..... | 18 |
| 2.6.1 Herramientas para crear una base de conocimiento..... | 19 |
| 2.7 Lógica difusa..... | 20 |
| 2.8 Python..... | 21 |
| 2.8.1 Bibliotecas | 22 |
| 2.9 Chatbot | 23 |
| 2.9.1 Clasificación | 24 |
| 2.10 Códigos de avería | 25 |
| 2.10.1 OBD-II (On-Board Diagnostics II):..... | 26 |
| 2.11 Metodología para el desarrollo de software | 26 |
| Capítulo 3. Estado del arte | 28 |
| 3.1 Trabajos de investigación..... | 28 |
| 3.1.1 Case Based Reasoning (CBR) methodology for Car Fault Diagnosis System (CFDS) using decision tree and Jaccard Similarity Method..... | 28 |
| 3.1.2 The application of rough set theory in vehicle transmission system fault diagnosis | 29 |

| | | |
|-----------------------------|--|----|
| 3.1.3 | Ensembles of neural networks with generalization capabilities for vehicle fault diagnostics | 29 |
| 3.1.4 | Automotive signal fault diagnostics—part I: signal fault analysis, signal segmentation, feature extraction and quasi-optimal feature selection | 29 |
| 3.1.5 | Immune systems inspired approach to anomaly detection and fault diagnosis for engines | 30 |
| 3.1.6 | An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, Boosting, and Randomization..... | 31 |
| 3.1.7 | Car failure fuzzy fault diagnostic system based on fault tree analysis | 31 |
| 3.1.8 | Decision tree methods: applications for classification and prediction | 32 |
| 3.1.9 | Study and analysis of decision tree based classification algorithms.. | 32 |
| 3.1.10 | A decision tree classifier for vehicle failure isolation..... | 33 |
| 3.1.11 | Sistemas expertos basados en reglas | 33 |
| 3.1.12 | Desarrollo de un sistema experto para el diagnóstico de fallas automotrices | 34 |
| 3.1.13 | Sistema experto para el diagnóstico automotriz..... | 34 |
| 3.1.14 | Sistema experto para el diagnóstico de fallas en motores a inyección electrónica de vehículos | 34 |
| 3.1.15 | Agente para el diagnóstico de motores de automóviles. | 35 |
| 3.2 | Foros web | 35 |
| 3.2.1 | Opinautos | 35 |
| 3.2.2 | CarComplaints..... | 36 |
| 3.3 | Herramientas de diagnóstico..... | 36 |
| 3.3.1 | ScanatorPC | 36 |
| 3.4 | Conclusión del estado del arte | 36 |
| Capítulo 4. Desarrollo..... | | 37 |
| 4.1 | Base de datos con códigos OBDII..... | 37 |
| 4.2 | Construcción de la base de conocimiento | 40 |
| 4.2.1 | Hechos: | 41 |
| 4.2.2 | Reglas: | 41 |
| 4.3 | Desarrollo de un árbol de decisión | 42 |
| 4.3.1 | Creación y poda de los árboles de decisión..... | 43 |
| 4.4 | Clasificación por medio de lógica difusa..... | 45 |
| 4.5 | Asistente personalizado inteligente: Chatbot..... | 47 |
| 4.5.1 | Módulo de atención a preguntas frecuentes | 47 |

| | | |
|---|--|----|
| 4.5.2 | Asistente de diagnóstico de averías en el bloque motor | 49 |
| Capítulo 5. Resultados | | 60 |
| 5.1 | Correlación de Pearson entre variables | 60 |
| 5.2 | Correlación de Spearman entre variables | 61 |
| 5.3 | Resultados obtenidos después de realizar la poda | 62 |
| 5.4 | Observaciones de la clasificación de temperatura..... | 69 |
| 5.5 | Generación del diagnóstico..... | 70 |
| Capítulo 6. Conclusiones y trabajo futuro | | 71 |
| Bibliografía | | 72 |

Índice Imágenes

| | |
|--|----|
| Figura 2.1. Estructura de un árbol de decisión | 7 |
| Figura 2.2. Matriz de confusión..... | 11 |
| Figura 2.3. Flexibilidad de la clasificación de la lógica difusa..... | 20 |
| Figura 4.1. Esquema de la propuesta | 37 |
| Figura 4.2. Base de datos de códigos OBDII | 38 |
| Figura 4.3. Consulta de códigos de avería desde el Chatbot | 39 |
| Figura 4.4. Panel de códigos desconocidos..... | 39 |
| Figura 4.5. Panel de actualización de códigos | 40 |
| Figura 4.6. Base de conocimiento..... | 41 |
| Figura 4.7. Métodos de poda de árboles de decisión..... | 43 |
| Figura 4.8. Árbol de decisión CART con poda CCP 2..... | 44 |
| Figura 4.9. Árbol de decisión CART con poda CCP 2..... | 45 |
| Figura 4.10. Universo de temperatura..... | 46 |
| Figura 4.11. Funciones de membresía..... | 46 |
| Figura 4.12. Clasificación de la temperatura..... | 46 |
| Figura 4.13. Asistente personalizado Chatbot Internauta..... | 48 |
| Figura 4.14. Función con AJAX del Chatbot | 49 |
| Figura 4.15. Barras de rango para voltajes. | 51 |
| Figura 4.16. Formulario de problema de encendido con <i>checkboxes</i> | 52 |
| Figura 4.17. Menú de consulta de códigos OBDII y alertas o testigos..... | 53 |
| Figura 4.18. Carrusel de alertas..... | 54 |
| Figura 4.19. Formulario de problemas de desempeño con <i>checkboxes</i> | 55 |
| Figura 4.20. Formulario de temperatura del anticongelante. | 56 |
| Figura 4.21. Formulario de sistema de enfriamiento. | 57 |
| Figura 4.22. Formulario de sistema de emisiones contaminantes..... | 58 |
| Figura 5.1. Matriz de correlación de Pearson..... | 60 |
| Figura 5.2. Matriz de correlación de Pearson..... | 62 |
| Figura 5.3. Matriz de confusión con CCP..... | 63 |
| Figura 5.4. Precisión, exactitud, tasa de error y <i>Novelty rejection percentage</i> de CCP | 64 |
| Figura 5.5. Precisión, exactitud, tasa de error y <i>Novelty rejection percentage</i> de Max Depth..... | 64 |

| | |
|---|----|
| Figura 5.6. Matriz de confusión con Max Depth | 65 |
| Figura 5.7. Matriz de confusión con <i>Min Samples Leaf</i> | 66 |
| Figura 5.8. Precisión, exactitud, tasa de error y <i>Novelty rejection percentage</i> de <i>Min Samples Leaf</i> | 66 |
| Figura 5.9. Matriz de confusión con <i>Min Samples Split</i> | 67 |
| Figura 5.10. Precisión, exactitud, tasa de error y <i>Novelty rejection percentage</i> de <i>Min Samples Split</i> | 68 |
| Figura 5.11. Promedio de validación cruzada por modelo..... | 68 |
| Figura 5.12. Valores de membresía..... | 69 |
| Figura 5.13. Diagnóstico generado. | 70 |

Índice Ecuaciones

| | |
|--|----|
| Ecuación 1: Fórmula de la entropía | 10 |
| Ecuación 2: Fórmula de la impureza de Gini | 10 |
| Ecuación 3: Fórmula de la precisión | 12 |
| Ecuación 4: Fórmula de la exactitud | 12 |
| Ecuación 5: Tasa de error | 13 |
| Ecuación 6: Fórmula del <i>Novelty rejection percentage</i> | 13 |
| Ecuación 7: Fórmula de la validación cruzada | 14 |
| Ecuación 8: Fórmula del coeficiente de correlación de Pearson | 15 |
| Ecuación 9: Fórmula del coeficiente de correlación de Spearman | 15 |

Capítulo 1. Introducción

Los automóviles han estado presentes por cerca de 139 años, cuando el 3 de julio de 1886, Karl Benz realizó su primer viaje público en el Patent-Motorwagen por las calles de Mannheim, Alemania, lo que marcó un hito en la historia de la automoción. Y desde entonces surgió la necesidad de brindar servicio a estas unidades, así como también, este nuevo invento pasó de ser un lujo a una necesidad.

Es aquí donde nace el servicio automotriz, para ofrecer a los dueños de los automóviles una asistencia preventiva y correctiva a los motores, los cuales, al ser impulsados por combustión interna, generan calor; y como es bien conocido, el enemigo de todo mecanismo motriz es el calor, que a su vez genera desgaste en sus componentes.

Estos servicios han ido cambiando a lo largo del tiempo, y han evolucionado a la par de los motores. Desde la implementación de aditivos que ayuden a mejorar el desempeño del automóvil pasando por motores más eficientes y económicos en cuanto a gasto de combustible se refiere, hasta la introducción de la electrónica para mejorar la eficiencia y facilitar el diagnóstico.

La introducción de la electrónica en los automóviles fue un proceso gradual que comenzó en las décadas de 1960 y 1970. En esta última década, se introdujeron sistemas electrónicos de control de emisiones para cumplir con las regulaciones ambientales. Estos sistemas utilizaban sensores y unidades de control electrónico para monitorear y ajustar el funcionamiento del motor, permitiendo un control preciso de la mezcla de combustible y aire, reduciendo las emisiones contaminantes [1]. A medida que avanzaba la electrónica, se agregaron más sistemas de este tipo a los automóviles, como sistemas de control de tracción, sistemas de frenos antibloqueo (ABS, *Anti-lock Braking System*), sistemas de control de estabilidad, sistemas de gestión del motor, sistemas de infoentretenimiento, entre otros.

Con ello, se desarrollaron los códigos de diagnóstico a bordo (OBD, *On Board Diagnostics*) para poder interpretar las fallas de los sensores y actuadores con los que ahora contaba el vehículo. Se introdujeron por primera vez en la década de 1980 con el sistema OBD-I. Sin embargo, el estándar OBD-II, que es el sistema de diagnóstico utilizado en la mayoría de los vehículos modernos, fue implementado de manera generalizada a partir de mediados de la década de 1990.

La Agencia de Protección Ambiental de los Estados Unidos (EPA, *Environmental Protection Agency*) estableció los estándares para el sistema OBD-II con el objetivo de mejorar el monitoreo y diagnóstico de los sistemas de emisiones de los vehículos. La implementación gradual del OBD-II comenzó en 1994 para los vehículos fabricados en los Estados Unidos, haciéndose obligatorio para todos los automóviles y camiones ligeros nuevos a partir de 1996 [2]. Convirtiéndose en el sistema de diagnóstico estándar utilizado

por la mayoría de los fabricantes de automóviles, ya que permite la detección y el registro de fallas en diferentes sistemas del vehículo, lo que facilita el proceso de diagnóstico y reparación por parte de los técnicos automotrices.

1.1 Problema

Es común hoy en día contar con un scanner para obtener un código de diagnóstico a bordo OBD-II (*On Board Diagnostics*) [3] en el teléfono inteligente y acceder —por medio de una interfaz bluetooth— a los códigos que son generados por alguna falla en el vehículo, así como también visitar una refaccionaria cercana, ya que algunas de ellas ofrecen el servicio de escaneo de la unidad. Lo cual ofrece más ventajas que tener como referencia una luz de alerta en el tablero, que puede tener cientos de causas y respectivas soluciones.

Sin embargo, el correcto diagnóstico de una avería es complejo, por lo que un testigo en el tablero o un código en el scanner no es suficiente para tomar una decisión. Así que, una vez obtenido un código, se debe buscar con ayuda de un manual o de un experto del dominio la descripción de este para analizar los sensores o actuadores involucrados en dicho bloque para proceder a revisarlos de manera individual y encontrar al responsable de la falla.

Algunas necesidades y problemas que se presentan actualmente al realizar un diagnóstico automotriz son:

- El significado y gravedad de las alertas presentes en el clúster del automóvil
- La interpretación de los códigos OBDII que almacena la ECU del vehículo.
- Los horarios de atención de servicio en un taller mecánico.
- El acceso a manuales de fallas automotrices o guías de reparación del bloque motor.
- La experiencia necesaria para realizar las pruebas de funcionamiento pertinentes.
- El proceso para seguir al revisar los principales sistemas del motor de un vehículo

1.2 Objetivos

El objetivo general del proyecto de tesis es: Especificar un conjunto de patrones para diagnosticar de manera automática las fallas de motor presentes en un automóvil a través de reglas y hechos, sustentado en árboles de decisión, bases de conocimiento y lógica difusa.

Los objetivos específicos se centran en:

- Crear una base de datos con los códigos del sistema OBDII que cuente con la descripción y las causas de la falla de motor para proporcionar una respuesta inmediata.
- Elaborar una base de conocimiento con los hechos y reglas observadas en el automóvil para alimentar el algoritmo del árbol de decisión.
- Desarrollar un sistema de clasificación de temperaturas utilizando lógica difusa y funciones de pertenencia, para mejorar la precisión del diagnóstico de fallas del motor en función de las variaciones y la incertidumbre en las lecturas de temperatura.
- Generar un árbol de decisión para determinar una estructura simple que permita inferir el diagnóstico correcto a través de la base de conocimiento.
- Implementar un asistente personalizado, un Chatbot, para proporcionar sugerencias y/o diagnosticar la falla del automóvil de un cliente y probar la viabilidad del árbol de decisión.

1.3 Propuesta

En consecuencia, este trabajo de tesis de maestría plantea el diagnóstico de fallas de motor por medio de patrones que permitan identificar la causa del comportamiento errático, basado en técnicas de aprendizaje automático y bases de conocimiento para que pueda ser implementado en un taller mecánico.

Las técnicas de aprendizaje automático a utilizar son, principalmente, el aprendizaje supervisado y árboles de decisión, debido a que el diagnóstico de fallas de motor se representa de manera ramificada dependiendo del conjunto de características presentadas donde los nodos hoja o terminales simbolizan la falla en cuestión.

Los patrones se generan a través del árbol de decisión al momento de recorrer los nodos (Que contienen hechos). Las características que denotan los patrones de fallas de motor se almacenan en una base de conocimiento, de tal manera, que establecen ciertas condiciones que se cumplen y que permiten realizar el diagnóstico apropiado.

Un asistente personalizado, se presenta para probar la viabilidad de la propuesta de investigación, que toma como base la aplicación desarrollada en la tesis de licenciatura denominada “Aplicación Coq’s Tuning Performance, para mejorar la Atención al Cliente mediante el Procesado de Solicitudes en Segundo Plano” [4]; se trata de un sitio web basado en AJAX (*Asynchronous Javascript And XML*) [5] para agendar y gestionar las citas de clientes del taller Coq’s Tuning Performance. En este trabajo para mejorar y simplificar la atención al cliente se incorpora un Chatbot inteligente como asistente a dicha aplicación, el cual responde a las dudas de los clientes. Desde casos básicos como agendar citas de servicio, orientar para llegar al taller por medio de Google Maps hasta identificar patrones

de comportamiento errático del vehículo para diagnosticar una falla por medio de hechos y reglas [6].

De esa manera, se incide en el modelo de negocio de este tipo de pequeñas o medianas empresas; teniendo como finalidad cubrir las necesidades de un mercado creciente y súper desarrollado como es hoy en día “la industria automotriz”. El Chatbot [72] está basado en el aprendizaje automático [7], suponiendo un enorme paso en la mejora y automatización de los procesos; ofreciendo un servicio inmediato, con disponibilidad absoluta, y resolviendo uno de los grandes problemas históricos de los centros de atención al cliente, especialmente en horas pico. Simplemente gestionan la automatización de las preguntas que los usuarios hacen más frecuentemente, ya que tienden a coincidir sobre cuestiones muy similares. En este trabajo, se innova al proporcionar el diagnóstico de fallas de motor identificando los patrones asociados a la misma, así como haciendo uso de aprendizaje supervisado, arboles de decisión, bases de conocimiento y lógica difusa.

El documento de tesis se organiza de la siguiente forma: Sección 2 presenta el marco teórico que sustenta la Detección de Patrones de Fallas de Automóviles basada en Técnicas de aprendizaje automático y Bases de Conocimiento. Sección 3 expone el estado del arte de este trabajo de tesis. Sección 4 explica el desarrollo del Chatbot como asistente personalizado en el sistema Web de Coq’s Tuning Performance. Sección 5 describe los resultados obtenidos. Sección 6 detalla las conclusiones y el trabajo futuro.

Capítulo 2. Marco teórico.

La inteligencia artificial (IA) es un campo de la informática que se centra en el desarrollo de sistemas y máquinas capaces de realizar tareas que normalmente requieren inteligencia humana. La idea principal es crear sistemas que puedan aprender, razonar, resolver problemas, reconocer patrones y tomar decisiones de manera autónoma. Las tecnologías basadas en la IA ya están siendo utilizadas para ayudar a los humanos a beneficiarse de mejoras significativas y disfrutar de una mayor eficiencia en casi todos los ámbitos de la vida. Pero el gran crecimiento de la IA también nos obliga a estar atentos para prevenir y analizar las posibles desventajas directas o indirectas que pueda generar la proliferación de la IA [8].

Los sistemas inteligentes se aplican en una amplia gama de campos, desde la atención médica, la industria automotriz, las finanzas hasta la gestión de la cadena de suministro, entre muchos otros. Su objetivo es automatizar tareas, mejorar la precisión y eficiencia, y en algunos casos, ofrecer soluciones innovadoras para problemas complejos [9].

2.1 Aprendizaje automático

El aprendizaje automático es un campo de la inteligencia artificial que se centra en el desarrollo de algoritmos y modelos que permiten a las computadoras aprender y tomar decisiones basadas en datos sin ser explícitamente programadas para cada tarea. Donde el desarrollador no tendrá que sentarse a programar por horas tomando en cuenta todos los escenarios posibles ni todas las excepciones posibles. Lo único que hay que hacer es alimentar el algoritmo con un volumen gigantesco de datos para que el algoritmo aprenda y sepa qué hacer en cada uno de estos casos [10].

Existen tres tipos principales de aprendizaje en aprendizaje automático:

- **Aprendizaje Supervisado:** Se entrena a los algoritmos con datos etiquetados, es decir, datos que tienen una salida conocida. El algoritmo aprende a hacer predicciones o tomar decisiones basadas en ejemplos pasados. Por ejemplo, predecir el precio de una casa según sus características.
- **Aprendizaje No Supervisado:** Los algoritmos aprenden a partir de datos no etiquetados. Aquí, el objetivo es encontrar patrones o estructuras dentro de los datos. Algunos ejemplos incluyen la segmentación de clientes en grupos similares o la detección de anomalías.
- **Aprendizaje por Refuerzo:** Los algoritmos aprenden observando y actuando en un entorno para maximizar una recompensa. El agente de

aprendizaje toma decisiones y recibe retroalimentación en forma de recompensa o penalización según sus acciones.

2.1.1 Modelos de aprendizaje automático

Los algoritmos de Aprendizaje automático, se pueden agrupar en tres modelos:

2.1.1.1 Modelos lineales

Estos tratan de encontrar una línea que se “ajuste” bien a la nube de puntos que se disponen. Aquí destacan desde modelos muy conocidos y usados como la regresión lineal (también conocida como la regresión de mínimos cuadrados), la logística (adaptación de la lineal a problemas de clasificación -cuando son variables discretas o categóricas-). Estos dos modelos tienen el problema del “*overfit*”, esto significa que se ajustan “demasiado” a los datos disponibles, con el riesgo que esto tiene para nuevos datos que pudieran llegar. Al ser modelos relativamente simples, no ofrecen resultados muy buenos para comportamientos más complicados [10].

2.1.1.2 Modelos de árbol

Son modelos precisos, estables y más sencillos de interpretar básicamente porque construyen unas reglas de decisión que se pueden representar como un árbol. A diferencia de los modelos lineales, pueden representar relaciones no lineales para resolver problemas. En estos modelos, destacan los árboles de decisión y los *random forest* (una media de árboles de decisión). Al ser más precisos y elaborados, obviamente ganamos en capacidad predictiva, pero perdemos en rendimiento [10].

2.1.1.3 Redes neuronales:

Las redes artificiales de neuronas tratan, en cierto modo, de replicar el comportamiento del cerebro, donde tenemos millones de neuronas que se interconectan en red para enviarse mensajes unas a otras. Esta réplica del funcionamiento del cerebro humano es uno de los “modelos de moda” por las habilidades cognitivas de razonamiento que adquieren. El reconocimiento de imágenes o vídeos, por ejemplo, es un mecanismo complejo y una red neuronal es lo mejor para realizarlo. El problema, como ocurre con el cerebro humano, es que son lentas de entrenar y necesitan mucha capacidad de cómputo [10].

2.2 Árboles de decisión

Dentro del aprendizaje automático, existen diferentes enfoques y técnicas. Uno de ellos son los "árboles de decisión". Estos son modelos de aprendizaje automático que imitan la forma en que toma decisiones una persona. Imagina un árbol con ramas que representan las posibles decisiones basadas en las características de los datos [11]. Cada "nodo" del árbol representa una característica o atributo, y las "hojas" del árbol representan la salida, es decir, la decisión o la predicción, tal como se aprecia en la figura 2.1.

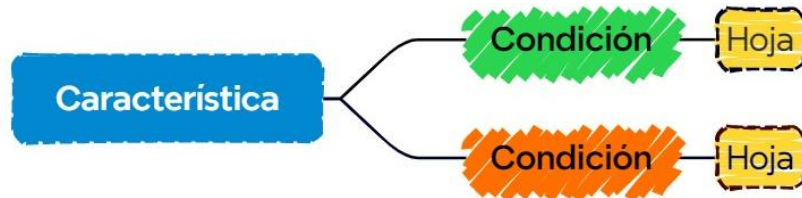


Figura 2.1. Estructura de un árbol de decisión

Los árboles de decisión dividen el conjunto de datos en subconjuntos cada vez más pequeños en función de diferentes características, con el objetivo de clasificar o predecir una variable objetivo. Son fáciles de entender e interpretar, lo que los hace populares en aplicaciones donde se necesita transparencia en el proceso de toma de decisiones [12].

Algunos de los algoritmos o variantes en la construcción y el funcionamiento de los árboles de decisión son:

- **ID3 (*Iterative Dichotomiser 3*)**: Fue uno de los primeros algoritmos para la construcción de árboles de decisión. Utiliza la ganancia de información y la entropía para tomar decisiones sobre cómo dividir los datos en cada nodo [12].
- **C4.5**: Es una evolución del ID3 y es capaz de manejar tanto características categóricas como numéricas [12]. Además de utilizar la ganancia de información, el C4.5 utiliza la ganancia de información normalizada (IG) y puede manejar conjuntos de datos incompletos.
- **C5.0**: Es una versión comercial del C4.5 y ofrece mejoras en términos de rendimiento y precisión en comparación con su predecesor [13].
- **GBT (*Gradient Boosting Trees*)**: Aunque no es un algoritmo de árbol de decisión en sí mismo, el *Gradient Boosting Trees* es una técnica que construye árboles de forma secuencial para corregir los errores de los árboles anteriores [13]. Utiliza el principio de *boosting* para mejorar la precisión del modelo. Los árboles individuales en un GBT se denominan árboles débiles.
- **CART (*Classification and Regression Trees*)**: es un algoritmo de aprendizaje supervisado que se utiliza para construir árboles de decisión tanto para problemas de clasificación como de regresión [14]. Fue propuesto por Leo Breiman en 1984.

Cada uno de estos algoritmos tiene sus propias características, ventajas y desventajas, y puede ser más adecuado para ciertos tipos de datos o contextos específicos. Por ejemplo, mientras que el ID3 es más simple y fácil de interpretar, el C4.5 y sus variantes manejan mejor la mayoría de los tipos de datos, incluidos los valores faltantes, el GBT es conocido por su capacidad para generar modelos altamente precisos a costa de mayor complejidad, por otra parte, CART puede manejar tanto problemas de clasificación como de regresión, además es robusto ante datos con mezcla de características categóricas y numéricas.

En este trabajo de tesis de maestría, se utiliza el algoritmo CART para la construcción del árbol de decisión por las ventajas y características que ofrece, tales como:

- **Interpretabilidad:** Los árboles de decisión CART son fáciles de interpretar y visualizar. Permite seguir las ramas del árbol para entender cómo se toma cada decisión, lo que facilita la explicación del modelo a personas no técnicas.
- **Manejo automático de características categóricas:** También pueden manejar automáticamente características categóricas sin necesidad de codificación adicional, como el uso de *one-hot encoding*. Esto simplifica el preprocesamiento de datos.
- **No linealidad:** Los árboles CART son capaces de capturar relaciones no lineales en los datos. Pueden modelar patrones complejos y no lineales en la relación entre las características y la variable objetivo.
- **No requiere escalar las características:** A diferencia de algunos modelos, como las máquinas de soporte vectorial, los árboles de decisión CART no requieren que las características se escalen, lo que puede simplificar el proceso de preprocesamiento de datos.
- **Rápido tiempo de entrenamiento:** En general, este tipo de árboles de decisión pueden entrenarse rápidamente, lo que los hace útiles para prototipos y desarrollo rápido de modelos.

2.2.1 Recorrido del árbol

Algunos de los algoritmos de recorrido de árboles más comunes incluyen la búsqueda en profundidad (DFS) y la búsqueda en amplitud (BFS) [12]. La elección entre ellos y su eficiencia depende de varios factores:

- **Profundidad del árbol:** Si el árbol es muy profundo, pero tiene un número limitado de hijos en cada nodo, DFS puede ser más eficiente, ya que no necesita mantener una gran cantidad de nodos en memoria. Si el árbol es ancho, pero poco profundo, BFS podría ser más eficiente.
- **Objetivos de búsqueda:** Si estás buscando un elemento específico en el árbol, DFS podría ser más eficiente si el elemento suele encontrarse cerca de la raíz del

árbol. Por otro lado, BFS sería más eficiente si estás buscando el elemento en el nivel más bajo posible.

- **Espacio en memoria disponible:** BFS tiende a utilizar más memoria porque mantiene una cola de nodos por procesar. Si la memoria es limitada, DFS podría ser preferible.
- **Comportamiento recursivo:** DFS se presta de manera natural a la implementación recursiva. Si estás trabajando con un lenguaje de programación que admite la recursión de manera eficiente, esto puede ser un factor en tu elección.
- **Estructura del árbol:** Algunos árboles, como los árboles binarios de búsqueda (BST), se pueden recorrer de manera más eficiente con algoritmos específicos, como el *Inorder*, *Preorder* o *Postorder traversal*. Estos algoritmos se adaptan a la estructura de los árboles específicos y su objetivo.

Para este trabajo de Tesis de maestría se utiliza un árbol CART, el cual utiliza un recorrido de árbol conocido como recorrido en profundidad primero.

En este recorrido, el algoritmo CART sigue el siguiente orden:

1. Visita el nodo actual (nodo de decisión).
2. Explora el subárbol izquierdo.
3. Explora el subárbol derecho.

Este recorrido en profundidad primero es adecuado para la construcción de árboles de decisión, ya que se toman decisiones en cada nodo y se exploran las ramas de manera recursiva. Cabe destacar que CART no utiliza recorridos de árbol específicos para su fase de construcción. La elección de las divisiones se realiza según la impureza de Gini en el caso de árboles de clasificación o la reducción de varianza en el caso de árboles de regresión.

2.2.2 Entropía

El concepto de entropía se basa en la teoría de la información. Esta teoría fue desarrollada inicialmente por Claude Shannon⁴ a mediados del siglo XX. Se usa para medir la homogeneidad de los datos y seleccionar la mejor característica para dividir el conjunto de datos en ramas más homogéneas en términos de sus clases. Esto ayuda a tomar decisiones óptimas para la clasificación de los datos durante la construcción del árbol [15].

La entropía se calcula matemáticamente tal como se aprecia en la ecuación 1 y puede variar entre 0 y 1. Un valor de entropía de 0 indica que el conjunto de datos está completamente ordenado o todas las muestras pertenecen a la misma categoría, mientras que un valor de 1 significa que el conjunto de datos es completamente aleatorio o caótico, sin patrones discernibles [10].

$$\text{Entropía} = -p_1 \log_2(p_1) - p_0 \log_2(p_0)$$

2.2.3 Gini

La impureza de Gini es una medida de cuán impuros son los nodos en un árbol de decisión. En el contexto de un árbol de clasificación, la impureza de Gini mide la probabilidad de clasificar incorrectamente una muestra aleatoria [16].

Se utiliza la impureza de Gini para evaluar cómo se deben dividir los nodos durante el proceso de construcción del árbol. El algoritmo de árbol de clasificación CART (*Classification and Regression Trees*) de *scikit-learn* utiliza la impureza de Gini como criterio por defecto para realizar divisiones.

Si el índice de Gini vale 0, significa que el nodo es totalmente puro. Se determina de la siguiente manera tal como se aprecia en la ecuación 2:

$$Gini(t) = 1 - \sum_{i=1}^j p(i|t)^2$$

Ecuación 2: Fórmula de la impureza de Gini

2.2.4 Poda de árboles

La poda de árboles de decisión es un proceso que implica eliminar ciertas ramas del árbol con el objetivo de mejorar la generalización del modelo y prevenir el sobreajuste. Una manera de corregir el sobre entrenamiento es el llamado podado de un árbol. Normalmente los criterios de parada suelen ser laxos, o incluso inexistentes dejando solamente el criterio de nodo terminal, pues al ser una búsqueda ávida y no exhaustiva, es posible que un corte no tan bueno desde el punto de vista del error, permita otro posterior mucho mejor. De ahí la importancia de dejar crecer el árbol para su posterior podado, e imponer criterios de parada laxos en la fase de crecimiento del árbol [17].

Existen diferentes enfoques para podar árboles de decisión, y algunos de los métodos comunes incluyen:

- **Poda por Complejidad del Costo (CCP):** También conocida como poda de mínima complejidad, este método utiliza un parámetro de complejidad del costo (CCP) para regular la cantidad de poda que se aplica al árbol. A medida que se incrementa el valor de CCP, se eliminan las ramas del árbol que proporcionan una mejora marginal en términos de precisión [17].
- **Poda por Altura (Max Depth):** Limitar la profundidad máxima del árbol es otra forma de controlar el sobreajuste. Al especificar la profundidad máxima del árbol, se cortan las ramas más profundas, lo que ayuda a evitar que el árbol se ajuste demasiado a los datos de entrenamiento [18].

- **Poda por Número Mínimo de Muestras en Hojas (*Min Samples Leaf*):** Este método implica establecer un umbral mínimo para la cantidad de muestras que debe haber en una hoja del árbol. Si después de una división una hoja tiene menos muestras que el umbral especificado, se poda esa rama [19].
- **Poda por Número Mínimo de Muestras para Dividir un Nodo (*Min Samples Split*):** Similar a la poda por número mínimo de muestras en hojas, esta técnica establece un umbral mínimo para la cantidad de muestras requeridas para realizar una nueva división. Si un nodo tiene menos muestras que este umbral, la división no se realiza [20].

Para este proyecto de Tesis de maestría se implementan estos métodos de poda al entrenar el árbol de decisión, con la finalidad de comparar los resultados de la predicción generada por el árbol y con sus respectivas métricas de evaluación elegir el más adecuado.

2.2.5 Métricas

La disponibilidad de volúmenes masivos de datos ha impulsado el desarrollo de nuevos algoritmos de *Machine and Deep Learning* que buscan continuamente mejorar el rendimiento en problemas de clasificación multiclase con datos no balanceados. Para resolver un problema de clasificación es necesario seleccionar el algoritmo apropiado mediante una o varias métricas de evaluación del rendimiento en varios algoritmos candidatos [21].

2.2.5.1 Matriz de confusión

La matriz de confusión o tabla de confusión es útil para calcular diversas métricas de evaluación del rendimiento de un modelo, como la precisión, la sensibilidad, la especificidad, la tasa de falsos positivos, la tasa de falsos negativos, entre otras. Donde los elementos diagonales representan las predicciones correctas, mientras que los elementos fuera de la diagonal representan errores de predicción. Las filas pertenecen a la clase predicha y las columnas a la clase real, representando el número de predicciones de cada clase mutuamente excluyentes [21].

La matriz de confusión se puede expresar de la siguiente manera tal como se aprecia en la figura 2.2:

| | |
|-----------|-----------|
| <i>TN</i> | <i>FP</i> |
| <i>FN</i> | <i>TP</i> |

Figura 2.2. Matriz de confusión

- **TN:** Es el número de verdaderos negativos.

- **FP:** Es el número de falsos positivos.
- **FN:** Es el número de falsos negativos.
- **TP:** Es el número de verdaderos positivos.

2.2.5.2 Precisión

La precisión (*precision* en inglés) es una métrica de evaluación utilizada en problemas de clasificación. Representa la proporción de instancias positivas correctamente clasificadas con respecto al total de instancias clasificadas como positivas por el modelo [21].

Es importante tener en cuenta que, en algunos casos, la precisión por sí sola puede no ser suficiente para evaluar completamente el rendimiento de un modelo, especialmente si hay un desequilibrio en la distribución de clases.

La fórmula de precisión se expresa tal como se aprecia en la ecuación 3:

$$\text{Precisión} = \frac{\text{Verdaderos Positivos}}{\text{Verdaderos Positivos} + \text{Falsos Positivos}}$$

Ecuación 3: Fórmula de la precisión

2.2.5.3 Accuracy

Mide la fracción de predicciones correctas (tanto positivas como negativas) con respecto al total de predicciones realizadas por el modelo. Es una métrica útil para problemas de clasificación balanceados, donde las clases positivas y negativas están aproximadamente igualmente distribuidas. Es la métrica más utilizada por investigadores en el caso dicotómico y multiclase, debido a su facilidad de cálculo y comprensión para evaluar la efectividad general del algoritmo. Sin embargo, una de sus principales desventajas es que produce menos valores discriminatorios y distintivos para el caso multiclase desbalanceado [21].

Es importante considerar otras métricas como precisión, para obtener una evaluación más completa del rendimiento del modelo.

La fórmula de la exactitud se expresa tal como se aprecia en la ecuación 4:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Ecuación 4: Fórmula de la exactitud

2.2.5.4 Tasa de error

La tasa de error es una métrica que se utiliza para evaluar el rendimiento de un modelo de clasificación. Representa la proporción de ejemplos que el modelo clasifica incorrectamente sobre el total de ejemplos en el conjunto de datos de prueba [22].

Matemáticamente, la tasa de error se calcula tal como se observa en la ecuación 5:

$$Tasa\ de\ error = \frac{\text{Número de predicciones incorrectas}}{\text{Total de ejemplos en el conjunto de datos}}$$

Ecuación 5: Tasa de error

Esta métrica es complementaria a la precisión (*accuracy*), que representa la proporción de ejemplos clasificados correctamente. Una tasa de error baja indica un buen rendimiento del modelo, mientras que una tasa de error alta indica que el modelo está clasificando incorrectamente una gran cantidad de ejemplos.

2.2.5.5 Novelty rejection percentage

Novelty rejection percentage es una métrica que indica la proporción de ejemplos en los datos de prueba que el modelo ha decidido abstenerse de predecir debido a que se consideran novedosos o poco probables según los datos de entrenamiento. Un *Novelty rejection percentage* alto indica que el modelo está siendo cauteloso al evitar hacer predicciones en casos inciertos, mientras que una tasa baja podría indicar que el modelo está haciendo predicciones en casos difíciles de manera indiscriminada. La combinación del *Novelty rejection percentage* con métricas tradicionales de evaluación del rendimiento del modelo permite una comprensión más completa de cómo el modelo se comporta en diferentes situaciones y su capacidad para manejar datos novedosos [22].

La fórmula para calcular el *Novelty rejection percentage* (NR, por sus siglas en inglés) se puede expresar de la siguiente manera (ver ecuación 6):

$$NR = \frac{N_{rechazos}}{N_{total}}$$

Ecuación 6: Fórmula del *Novelty rejection percentage*

Donde:

- N rechazos es el número total de ejemplos que el modelo ha decidido abstenerse de predecir debido a que se consideran novedosos.
- N total es el número total de ejemplos en el conjunto de datos de prueba.

Esta métrica proporciona una medida de la proporción de ejemplos que el modelo ha rechazado debido a su percepción de que son novedosos o inciertos en comparación con el conjunto total de ejemplos en el conjunto de datos de prueba.

2.2.5.6 Validación cruzada

La validación cruzada es una técnica utilizada en el aprendizaje automático para evaluar el rendimiento de un modelo predictivo. En lugar de dividir los datos en un conjunto de entrenamiento y un conjunto de prueba una sola vez, la validación cruzada divide los datos en múltiples subconjuntos (llamados "*fold*s"). Luego, se entrena el modelo en varios de estos subconjuntos y se evalúa en el resto. Esto se repite varias veces, de modo que cada subconjunto se utiliza tanto para entrenamiento como para prueba en diferentes momentos [23].

El propósito de la validación cruzada es proporcionar una estimación más precisa del rendimiento del modelo, especialmente cuando se dispone de un conjunto de datos limitado. Al evaluar el modelo en múltiples divisiones de los datos, se reduce la posibilidad de que el rendimiento del modelo esté sesgado por una partición específica de los datos de entrenamiento y prueba [23].

Uno de los métodos más comunes de validación cruzada es la validación cruzada *k-fold*, donde los datos se dividen en *k* subconjuntos de aproximadamente el mismo tamaño. El modelo se entrena *k* veces, utilizando un subconjunto diferente como datos de prueba en cada iteración y los restantes como datos de entrenamiento.

Si se está utilizando la validación cruzada *k-fold* para evaluar la precisión de un modelo de clasificación, la fórmula es (ver ecuación 7):

$$\text{Precisión} = \frac{\text{Número de predicciones correctas en el fold } i}{\text{Número total de muestras en el fold } i}$$

Ecuación 7: Fórmula de la validación cruzada

Donde:

- *k* es el número de *fold*s en la validación cruzada.

La validación cruzada es una técnica fundamental en la evaluación y selección de modelos de aprendizaje automático, ya que proporciona una evaluación más robusta y confiable del rendimiento del modelo.

2.2.5.7 Correlación.

El concepto de correlación implica contar con un par de observaciones (X y Y), es decir, el valor que toma Y para determinado valor de X; la correlación permite examinar la tendencia de dos variables a ir juntas, por ejemplo, sabemos que al incrementar la edad también aumentan las cifras de presión arterial, por lo tanto, si queremos responder una pregunta de investigación como ¿cuál es la relación entre edad y presión arterial?, la prueba estadística pertinente es una prueba de correlación. Esta prueba permite cuantificar la magnitud de la correlación entre dos variables y ayuda a predecir valores. Si estas variables

tuvieran una correlación perfecta se podría inferir el valor de la variable Y conociendo el valor de X [24].

2.2.5.7.1 Correlación de Pearson

El coeficiente de correlación de Pearson fue introducido por Galton en 1877 y desarrollado más adelante por Pearson. Es un indicador usado para describir cuantitativamente la fuerza y dirección de la relación entre dos variables cuantitativas de distribución normal y ayuda a determinar la tendencia de dos variables a ir juntas, a lo que también se denomina covarianza [24].

La fórmula para calcular el coeficiente de correlación de Pearson, la cual considera en el numerador la covarianza (suma de productos xy) y en el denominador, la raíz del producto de las sumas de cuadrados de ambas variables se observa en la ecuación 8.

$$r_s = \frac{n\sum xy - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}}$$

Ecuación 8: Fórmula del coeficiente de correlación de Pearson.

2.2.5.7.2 Correlación de Spearman.

La correlación de Spearman o también conocida como rho de Spearman es el análogo no paramétrico de la correlación de Pearson. Se utiliza para variables cuantitativas de libre distribución o con datos ordinales. La correlación de Spearman se basa en la sustitución del valor original de cada variable por sus rangos, tal como se puede observar en su fórmula (Ver ecuación 9). Para calcularla se requiere que se ordenen los valores de cada sujeto para cada variable X, Y, además de que se asigne un rango. Si existe una correlación fuerte, los rangos deben ser consistentes: bajos rangos de X se correlacionarán con bajos rangos de Y [24].

$$r_s = 1 - \frac{n\sum d^2}{n(n^2 - 1)}$$

Ecuación 9: Fórmula del coeficiente de correlación de Spearman.

Donde:

$\sum d^2$ = sumatoria de la diferencia de rangos

n = número de pares (X, Y)

2.3 Sistemas expertos

El nombre de Sistema Experto deriva del término “sistema experto basado en conocimiento”. Un Sistema Experto es un sistema que emplea conocimiento humano capturado en una computadora para resolver problemas que normalmente requieran de expertos humanos [25].

Las principales características clave de los sistemas expertos son las siguientes:

- **Base de conocimiento:** Contienen una base de conocimiento que almacena información específica del dominio en forma de reglas, hechos o relaciones.
- **Motor de inferencia:** Utiliza un motor de inferencia para procesar la información almacenada en la base de conocimiento y derivar conclusiones o recomendaciones.
- **Interfaz de usuario:** Proporciona una interfaz de usuario para interactuar con los usuarios y recibir consultas o datos de entrada.
- **Capacidad explicativa:** Puede explicar el razonamiento detrás de sus decisiones, lo que se conoce como capacidad explicativa.
- **Aprendizaje:** Algunos sistemas expertos pueden incluir capacidades de aprendizaje, actualizando su base de conocimiento con nueva información a lo largo del tiempo.
- **Dominio específico:** Se aplican a un dominio específico de conocimiento y no son generalistas en naturaleza.

2.4 Motor de inferencia

Los motores de inferencia, también llamados razonadores, son aplicaciones de software que derivan nuevos hechos o asociaciones a partir de información existente. La inferencia y las reglas de inferencia permiten derivar nuevos datos a partir de datos que ya se conocen. De esta manera, se pueden agregar nuevos conocimientos basados en los anteriores. Al crear un modelo de información y relaciones, permitimos a los razonadores sacar conclusiones lógicas basadas en el modelo [26].

El motor de inferencia está conformado por un algoritmo de encadenamiento hacia atrás (árbol de decisión) que refleja un tipo de inferencia [27]. Además, es una parte fundamental de un sistema experto y se utiliza para procesar las reglas lógicas y deducir conclusiones o resultados a partir de un conjunto de hechos o premisas. En el contexto de sistemas expertos, el motor de inferencia simula el proceso de razonamiento humano, aplicando reglas lógicas para llegar a conclusiones basadas en la información disponible.

2.5 Base de datos

Una base de datos es un sistema organizado para almacenar, gestionar y recuperar información de manera estructurada. En esencia, es un conjunto de datos interrelacionados que se almacenan de manera que se puedan acceder, modificar o eliminar fácilmente. Estas bases de datos pueden ser tan simples como una hoja de cálculo o tan complejas como los sistemas utilizados por grandes empresas o instituciones.

Las bases de datos se utilizan en una amplia gama de aplicaciones y entornos, desde sistemas empresariales hasta aplicaciones web, servicios en la nube, aplicaciones móviles y mucho más. Su función principal es permitir el almacenamiento y la gestión eficiente de grandes cantidades de datos, facilitando la recuperación y manipulación de la información cuando sea necesario [28].

Clasificar las bases de datos según su naturaleza y estructura nos lleva a estas tres categorías principales:

- Bases de Datos Relacionales:
 - Utilizan un modelo relacional para organizar los datos en tablas con filas y columnas.
 - Están basadas en el álgebra relacional y utilizan el lenguaje SQL (*Structured Query Language*) para realizar consultas.
 - Garantizan la integridad de los datos mediante restricciones y relaciones clave entre tablas.
 - Ejemplos: MySQL, PostgreSQL, Oracle, SQL Server.

- Bases de Datos No Relacionales (NoSQL):
 - Permiten una estructura de datos más flexible que no está limitada por tablas, filas y columnas.
 - Pueden manejar grandes volúmenes de datos no estructurados o semi-estructurados.
 - Incluyen varios tipos, como las bases de datos de documentos, de grafos, clave-valor y de columnas.
 - Ejemplos: MongoDB, Cassandra, Redis, Neo4j.

- Bases de Datos Temporales:
 - Diseñadas para almacenar y gestionar datos temporales o datos que cambian con el tiempo.
 - Permiten recuperar datos tal como estaban en un momento específico del pasado.

- Se utilizan en aplicaciones que necesitan manejar datos temporales como registros de cambios, seguimiento de versiones o análisis de tendencias a lo largo del tiempo.
- A menudo se integran con sistemas de gestión de bases de datos relacionales o no relacionales.
- Ejemplos: Teradata, TimeScaleDB, sistemas de bases de datos temporales específicos construidos sobre otras bases de datos.

La construcción de una base de datos es un proceso iterativo. Es posible que se deba ajustar y optimizar el diseño a medida que evolucionan los requisitos del sistema y se acumulan datos.

Cada tipo de base de datos tiene su propia forma de almacenar, organizar y recuperar los datos, y se elige según las necesidades específicas de la aplicación y los requisitos de almacenamiento y manipulación de datos.

Para este proyecto de tesis se utiliza una base de datos relacional, ya que es una excelente elección debido a varias razones clave. En primer lugar, la organización tabular de una base de datos relacional proporciona una estructura clara y coherente para almacenar los códigos de avería junto con sus respectivas causas y soluciones, facilitando así la gestión y consulta de los datos. Además, el lenguaje SQL utilizado en las bases de datos relacionales permite realizar operaciones CRUD (crear, leer, actualizar y eliminar) de manera eficiente y segura, lo que es crucial para la interacción dinámica a través del panel AJAX desarrollado. Estas características aseguran la consistencia, integridad y escalabilidad del sistema, cumpliendo con los requisitos funcionales y técnicos del proyecto.

2.6 Base de conocimiento

Una base de conocimiento es una colección organizada de información, datos, hechos, procedimientos, reglas y experiencias que se utilizan para almacenar, gestionar y compartir conocimientos. Se estructura de manera que sea accesible y útil para aquellos que necesitan información específica sobre un tema determinado [29].

Las bases de conocimiento son vitales en muchos entornos, desde empresas y organizaciones hasta instituciones académicas y comunidades en línea. Facilitan la gestión y el acceso a información crucial, permitiendo a las personas obtener conocimiento relevante, tomar decisiones informadas y resolver problemas de manera más efectiva. Además, estas bases de conocimiento pueden evolucionar con el tiempo a medida que se agregan nuevos conocimientos y se actualizan los existentes [30].

La creación de una base de conocimiento implica reunir, organizar y estructurar información relevante para un dominio específico. Los pasos generales para crear una base de conocimiento son los siguientes:

1. Definir el dominio:

- Identifica el área o tema específico para el cual deseas crear la base de conocimiento.

2. Recopilar información:

- Reúne información relevante y confiable relacionada con el dominio. Se puede utilizar fuentes como libros, artículos, documentos técnicos, expertos en el campo, entre otros.

3. Organizar la información:

- Estructura la información de manera lógica y coherente. Se puede utilizar categorías, subcategorías y etiquetas para organizar los datos.

2.6.1 Herramientas para crear una base de conocimiento

Existen varias herramientas y plataformas que se pueden utilizar para crear y gestionar bases de conocimiento, las siguientes son las mejor valoradas del 2023 [31]:

1. Guru:

- Enfoque: *Guru* se centra en la gestión del conocimiento y la colaboración del equipo.

2. Freshservice:

- Enfoque: *Freshservice* es una solución completa de servicio de TI que incluye funciones de gestión de incidentes, problemas, cambios y activos, además de un portal de autoservicio.

3. Zoho Desk:

- Enfoque: *Zoho Desk* es una solución de mesa de ayuda y soporte al cliente.

4. SQLite:

- Enfoque: No necesita un servidor de base de datos separado, lo que facilita la implementación y administración.

5. Google Sheets:

- Enfoque: Diseñado para facilitar la colaboración en tiempo real entre múltiples usuarios en línea.

Si bien las 3 primeras herramientas ofrecen un enfoque especializado en soporte y atención al cliente, con características avanzadas, tienen un modelo de pago asociado. Más apropiados para empresas con necesidades específicas de servicio técnico.

Por otra parte, SQLite y Google Sheets son más adecuadas para aplicaciones locales o en la nube que requieren eficiencia y simplicidad en la gestión y creación de bases de conocimiento

2.7 Lógica difusa

La lógica difusa es un enfoque que permite manejar la incertidumbre y la imprecisión en el razonamiento y la toma de decisiones. A diferencia de la lógica clásica, donde las variables son verdaderas o falsas, en la lógica difusa las variables pueden tener valores intermedios, representando grados de verdad o pertenencia [32].

Básicamente la Lógica Difusa es una lógica multivaluada que permite representar matemáticamente la incertidumbre y la vaguedad, proporcionando herramientas formales para su tratamiento [33].

Como indica Zadeh [34], "Cuando aumenta la complejidad, los enunciados precisos pierden su significado y los enunciados útiles pierden precisión.", que puede resumirse como que "los árboles no te dejan ver el bosque".

En la lógica difusa, se utilizan conjuntos difusos para representar estas variables, donde los elementos tienen grados de pertenencia. Por ejemplo, en lugar de decir que Una temperatura es fría o caliente, se puede expresar su temperatura como "fría en un 70%" o "cálida en un 30%" tal como se aprecia en la figura 2.3.

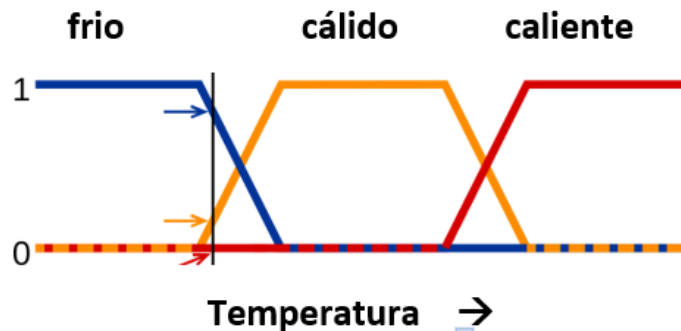


Figura 2.3. Flexibilidad de la clasificación de la lógica difusa

Algunos conceptos clave de cómo trabaja la lógica difusa se describen a continuación:

1. Conjuntos difusos:

- En lugar de tener conjuntos nítidos (definidos de manera precisa), la lógica difusa utiliza conjuntos difusos, donde los elementos tienen grados de pertenencia. Cada elemento puede pertenecer al conjunto en diferentes grados, en lugar de estar completamente dentro o fuera del conjunto.

2. Funciones de pertenencia:

- Cada conjunto difuso está asociado con una función de membresía que describe cómo los elementos pertenecen al conjunto en términos de grados. Estas funciones pueden tener formas diversas, como triángulos, trapecios o curvas más complejas.

3. Operaciones difusas:

- Las operaciones difusas permiten combinar conjuntos difusos para obtener nuevos conjuntos difusos. Algunas operaciones comunes incluyen la unión, la intersección y la complementación difusas.

4. Reglas difusas:

- En lógica difusa, las reglas se expresan de manera más flexible. Cada regla implica una combinación de proposiciones con grados de verdad difusos. Estas reglas se utilizan en sistemas de inferencia difusa para tomar decisiones.

5. Sistemas de inferencia difusa:

- Estos sistemas utilizan reglas difusas y operaciones difusas para inferir resultados basados en la información difusa proporcionada. Los sistemas de inferencia difusa son ampliamente utilizados en control difuso, toma de decisiones y modelado de sistemas complejos.

6. Defuzzificación:

- La salida de un sistema difuso es a menudo un conjunto difuso. La defuzzificación es el proceso de convertir ese conjunto difuso en un valor numérico concreto para tomar decisiones o realizar acciones.

Este enfoque es útil en situaciones donde los límites son difusos o subjetivos, ya que permite modelar conceptos que no son completamente verdaderos ni completamente falsos. Se aplica en campos como sistemas de control, inteligencia artificial, reconocimiento de patrones, toma de decisiones, entre otros, donde la ambigüedad y la imprecisión son comunes.

2.8 Python

Python es un lenguaje de programación creado por Guido van Rossum a principios de los años 90 cuyo nombre está inspirado en el grupo de cómicos ingleses “Monty Python”. Es un lenguaje similar a Perl, pero con una sintaxis muy limpia y que favorece un código legible. Se trata de un lenguaje interpretado o de script, con tipado dinámico, fuertemente tipado, multiplataforma y orientado a objetos. Ha emergido como una fuerza dominante en

el panorama de la programación, ganando adeptos en diversas comunidades y sectores. Su popularidad se debe en gran medida a su sintaxis clara, versatilidad y a una comunidad activa que ha contribuido al desarrollo de bibliotecas y *frameworks* poderosos [35].

Python ha encontrado un nicho especial en el ámbito de la ciencia de datos e inteligencia artificial. Bibliotecas como *NumPy*, *Pandas*, *Scikit-Learn*, *DecisionTreeClassifier*, *CSV*, *Graphviz* *Matplotlib*, *Seaborn* y *skfuzzy*. Y han convertido a Python en la elección preferida para profesionales de estas áreas. Su facilidad para realizar análisis de datos, aprendizaje automático y desarrollo de modelos ha consolidado su posición en la vanguardia de la revolución digital.

2.8.1 Bibliotecas

Las bibliotecas mencionadas, como *NumPy*, *Pandas*, *Scikit-Learn* y *TensorFlow*, desempeñan roles específicos y complementarios en el ámbito de la ciencia de datos, aprendizaje automático e inteligencia artificial [35]. Aquí hay una breve descripción de cómo trabajan algunas de estas librerías:

1. **NumPy:**

- Propósito: *NumPy* es una biblioteca fundamental para realizar operaciones numéricas en Python.

2. **Pandas:**

- Propósito: *Pandas* se utiliza para manipulación y análisis de datos estructurados.

3. **Cómo Trabaja:** Introduce las estructuras de datos *DataFrame* y *Series*. Los **Scikit-Learn:**

- Propósito: *Scikit-Learn* es una biblioteca para aprendizaje automático y minería de datos.

4. **DecisionTreeClassifier:**

- Propósito: El propósito principal del *DecisionTreeClassifier* es realizar tareas de clasificación, donde se busca asignar una etiqueta a cada instancia de entrada basándose en las características de esa instancia.

5. **CSV (Comma-Separated Values):**

- Propósito: La biblioteca *CSV* en Python se utiliza para trabajar con archivos *CSV*, que son archivos de texto plano que representan datos tabulares utilizando comas para separar las columnas. Su propósito principal es facilitar la lectura y escritura de datos en formato *CSV*.

6. **Graphviz:**

- Propósito: La biblioteca *graphviz* se utiliza para visualizar gráficos, incluidos los árboles de decisión generados por modelos como *Decision Trees*. En el contexto de árboles de decisión, *graphviz* se utiliza junto con la función `export_graphviz` de *Scikit-Learn* para generar archivos DOT que representan el árbol de decisión.

7. **Matplotlib:**

- Propósito: *Matplotlib* es una biblioteca de trazado en 2D de Python que produce figuras de alta calidad en una variedad de formatos y entornos. Es muy flexible y se puede utilizar para crear una amplia gama de gráficos, desde simples trazados de líneas hasta gráficos complejos.

8. **Seaborn:**

- Propósito: *Seaborn* es una biblioteca de visualización de datos en Python que se basa en *Matplotlib* y proporciona una interfaz de alto nivel para crear gráficos estadísticos atractivos y informativos. Está diseñado específicamente para funcionar bien con estructuras de datos de Pandas y es ideal para explorar y visualizar conjuntos de datos complejos.

9. **skfuzzy:**

- Propósito: Es una biblioteca en Python que proporciona herramientas para trabajar con lógica difusa y sistemas difusos. La lógica difusa es una extensión de la lógica clásica que maneja la incertidumbre y la imprecisión, permitiendo representar y procesar información que no es necesariamente precisa o booleana.

En este trabajo de tesis de maestría se hace uso de este conjunto de módulos y librerías para entrenar, evaluar y visualizar modelos de árboles de decisión para clasificación utilizando varias técnicas de poda y visualizar los resultados de manera efectiva.

2.9 Chatbot

Un Chatbot es un programa informático diseñado para simular una conversación con seres humanos. Está basado en inteligencia artificial y se utiliza para interactuar con personas a través de texto o voz, respondiendo preguntas, proporcionando información, realizando tareas específicas o simplemente manteniendo una conversación.

2.9.1 Clasificación

Los Chatbots se pueden clasificar utilizando diferentes parámetros: el dominio del conocimiento, el servicio prestado, los objetivos, el método de procesamiento de entrada y generación de respuesta, la ayuda humana y el método de construcción [37,38].

La clasificación basada en los objetivos considera el objetivo principal que los Chatbots pretenden alcanzar. Los Chatbots informativos están diseñados para proporcionar al usuario información almacenada de antemano o disponible en una fuente fija, como los Chatbots de preguntas frecuentes [37, 41].

La clasificación basada en el método de procesamiento de entradas y generación de respuestas tiene en cuenta el método de procesamiento de entradas y generación de respuestas. Hay tres modelos utilizados para producir las respuestas adecuadas: modelo basado en reglas, modelo basado en recuperación y modelo generativo [39].

Los Chatbots modelo basados en reglas son el tipo de arquitectura con el que se construyeron la mayoría de los primeros Chatbots, al igual que numerosos Chatbots en línea. Eligen la respuesta del sistema basándose en un conjunto fijo de reglas predefinidas, basadas en el reconocimiento de la forma léxica del texto de entrada sin crear ninguna respuesta de texto nueva. [42].

Un poco diferente del modelo basado en reglas es el modelo basado en recuperación, que ofrece más flexibilidad al consultar y analizar los recursos disponibles utilizando API [40]. Un Chatbot basado en recuperación recupera algunas respuestas candidatas de un índice antes de aplicar el enfoque de coincidencia a la selección de respuestas [42].

El modelo generativo genera respuestas mejor que los otros tres modelos, basándose en los mensajes de los usuarios actuales y anteriores. Estos Chatbots son más parecidos a los humanos y utilizan algoritmos de aprendizaje automático y técnicas de aprendizaje profundo. Sin embargo, existen dificultades para construirlos y capacitarlos [40].

En este trabajo de tesis de maestría se construyó un Chatbot de dominio cerrado especializado en el área de fallas del bloque motor, con un enfoque informativo y basado en modelos de Aprendizaje automático. Utiliza un formulario estructurado para procesar las entradas del usuario y emplea un árbol de decisión CART para realizar predicciones de fallas. Adoptando una estrategia más específica y controlada para proporcionar respuestas precisas a consultas relacionadas con fallas del bloque motor.

Existen plataformas para crear un Chatbot, tales como [43]:

- **Zendesk:** Es una plataforma de servicio al cliente que ofrece una variedad de herramientas, incluyendo Chatbots. Ayuda a las empresas a gestionar y optimizar sus interacciones con los clientes a través de diferentes canales,

como chat en vivo, correo electrónico y redes sociales [44]. Se ha convertido en una herramienta popular para empresas de diversos tamaños que buscan mejorar sus operaciones de atención al cliente [45].

- **Tars:** (Anteriormente conocido como *ChatGrape*) es una plataforma de construcción de Chatbots centrada en la creación de *bots* de conversación específicos para la generación de leads y la interacción con clientes [46]. Ofrece una interfaz de arrastrar y soltar para la creación de flujos de conversación y no requiere habilidades de programación para su utilización.
- **Flow XO:** Es una plataforma de automatización de flujos de trabajo y Chatbots que permite crear *bots* conversacionales para diversas aplicaciones, incluyendo Facebook Messenger y *Slack* [46]. Ofrece una interfaz intuitiva basada en la lógica de arrastrar y soltar para construir flujos de conversación y procesos automatizados [48].
- **Chatfuel:** Es una plataforma que permite crear Chatbots para Facebook Messenger. Está diseñada para facilitar la creación de *bots* sin necesidad de programación [49]. Es una herramienta basada en arrastrar y soltar que simplifica el proceso de desarrollo de Chatbots, lo que la hace accesible incluso para personas sin experiencia en programación [50].

Sin embargo, las aplicaciones antes mencionadas pueden ser costoso, especialmente para pequeñas empresas. Además, que las opciones gratuitas tienen algunas limitaciones, y es posible que se necesite una suscripción a un plan de pago para acceder a funciones más avanzadas.

Se optó por desarrollar el Chatbot desde cero con AJAX por diversas ventajas, principalmente:

- **Personalización total:** Crear un Chatbot desde cero con AJAX brinda un control total sobre la lógica y el comportamiento del *bot*, permitiendo adaptarlo exactamente a las necesidades.
- **Flexibilidad:** Se puede integrar el Chatbot con cualquier tecnología o sistema que se desee, ya que se cuenta con el control total sobre el código.
- **Escalabilidad:** Al desarrollar desde cero, permite escalar el Chatbot según sea necesario, ajustando y optimizando el rendimiento según el crecimiento de la demanda.

2.10 Códigos de avería

Los sistemas OBD (On-Board Diagnostics) son estándares utilizados en vehículos para monitorear y diagnosticar el rendimiento del motor y los sistemas de emisiones. La

diferencia principal entre OBD-I y OBD-II radica en la complejidad y capacidad de diagnóstico [51].

2.10.1 OBD-II (On-Board Diagnostics II):

OBD-II representó un avance significativo en la estandarización de los sistemas de diagnóstico a bordo, brindando una mayor capacidad de monitoreo y diagnóstico, así como una mayor accesibilidad a la información de diagnóstico para los técnicos y propietarios de vehículos.

Para este trabajo de tesis de maestría se almacena en una base de datos este conjunto de códigos para integrar su respectiva causa, así como también las soluciones pertinentes a la avería presente. Todo esto mediante un gestor presente en el sitio web Coq's Tuning Performance, para su ingreso, actualización, consulta y manejo de la información.

2.11 Metodología para el desarrollo de software

Para el desarrollo de proyectos de software (en este caso de proyectos web) existen diversas metodologías que ofrecen diferentes ventajas y trabajan de diferente forma de acuerdo al tipo de proyecto a realizar.

La metodología hace referencia al conjunto de procedimientos racionales utilizados para alcanzar un objetivo que requiera habilidades y conocimientos específicos. La metodología es una de las etapas específicas de un trabajo o proyecto que parte de una posición teórica y conlleva a una selección de técnicas concretas o métodos acerca del procedimiento para el cumplimiento de los objetivos. Es el conjunto de métodos que se utilizan en una determinada actividad con el fin de formalizarla y optimizarla. Determina los pasos a seguir y cómo realizarlos para finalizar una tarea. [52].

La metodología para el desarrollo de software es un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito. Una metodología para el desarrollo de software comprende los procesos a seguir sistemáticamente para idear, implementar y mantener un producto software desde que surge la necesidad del producto hasta que se cumple el objetivo por el cual fue creado [53].

Los pasos que se llevarán a cabo para desarrollar la propuesta son:

1. Investigar el estado del arte de fallas de motor utilizando técnicas de inteligencia artificial.
2. Revisar *datasets* de códigos OBD-II y seleccionar el más adecuado a la propuesta.
3. Analizar técnicas de ML para aprendizaje automático, patrones, árboles de decisión, bases de conocimiento y reglas.

4. Crear una base de conocimiento para establecer las fallas de motor.
5. Desarrollar un árbol de decisión basado en la base de conocimiento y reglas.
6. Especificar patrones de fallas de motor de automóviles para proporcionar un diagnóstico correcto.
7. Realizar un caso de estudio a través de un *ChatBot* para probar la viabilidad de la propuesta.

Capítulo 3. Estado del arte

En este capítulo se presentan trabajos de investigación relacionados con árboles de decisión aplicado a problemas de clasificación, de diagnóstico de fallas por medio de sistemas expertos y agentes, sitios web que incluyen la experiencia de usuarios que comparten experiencias y síntomas de falla, así como aplicaciones de escritorio que proporcionan códigos de avería y síntomas que muestra un vehículo.

3.1 Trabajos de investigación

El siguiente apartado se centra en explorar diversos trabajos de investigación acerca de diagnóstico de fallas en automóviles, aplicaciones de árboles de decisión para problemas de clasificación y sistemas expertos en el ámbito del diagnóstico automotriz. Estas aplicaciones son cruciales para mejorar la eficiencia y precisión en la clasificación e identificación de fallas en vehículos, contribuyendo así a la seguridad y el mantenimiento adecuado de los automóviles.

3.1.1 Case Based Reasoning (CBR) methodology for Car Fault Diagnosis System (CFDS) using decision tree and Jaccard Similarity Method

Este artículo [54] presenta una metodología de Razonamiento Basado en Casos (CBR, por sus siglas en inglés) para el Sistema de Diagnóstico de Fallas en Automóviles (CFDS). Este sistema utiliza un árbol de decisión para almacenar casos y el método de similitud de Jaccard para calcular la similitud entre casos nuevos y casos almacenados. el artículo no proporciona detalles específicos sobre el tipo exacto de árboles de decisión utilizados. No obstante, dado el contexto de la aplicación y la descripción proporcionada, es posible que se estén utilizando árboles de decisión estándar, como los árboles de decisión binarios. Estos árboles de decisión se utilizan para representar las relaciones entre las características del automóvil y las posibles fallas, lo que facilita la recuperación eficiente de casos similares durante el proceso de diagnóstico. Si los árboles de decisión utilizados para organizar y recuperar casos no se interpretan correctamente, pueden conducir a resultados de diagnóstico erróneos o sesgados.

Menciona el uso de la metodología de árbol de decisiones y el método de similitud de Jaccard para calcular la similitud entre nuevos casos y casos almacenados en la base de casos. Sin embargo, no proporciona resultados numéricos específicos derivados de estas métricas en términos de precisión, *recall*, *F1-score* u otras métricas típicas de evaluación de sistemas de diagnóstico de fallas en automóviles. En su lugar, describe cómo se utilizan estas técnicas en el proceso de diagnóstico de fallas, pero no proporciona valores numéricos específicos asociados con su rendimiento.

3.1.2 The application of rough set theory in vehicle transmission system fault diagnosis

Este artículo [55] trata sobre la aplicación de la teoría de conjuntos aproximados (*rough set theory*) en el diagnóstico de fallas en el sistema de transmisión de vehículos. Se explica cómo se minimiza el vector de características de falla mediante el cálculo de la importancia de cada atributo y su dependencia. Esto permite simplificar los datos y mejorar la eficiencia del diagnóstico en tiempo real. Además, presenta un método para diagnosticar rápidamente fallas paroxísticas mediante la eliminación de vectores de características redundantes. Finalmente discute la consistencia de las decisiones y se propone una reducción de reglas de decisión. Se detalla cómo se aplica la teoría de conjuntos aproximados específicamente en el diagnóstico de fallas en la caja de engranajes del vehículo. Se analiza la importancia de los atributos y se calcula su dependencia.

Sin embargo, el artículo no proporciona detalles específicos sobre las métricas utilizadas o la precisión del método propuesto.

3.1.3 Ensembles of neural networks with generalization capabilities for vehicle fault diagnostics

El artículo [56] trata sobre un enfoque de ensamblaje de redes neuronales para la detección de fallas en vehículos. Se presenta un enfoque de ensamblaje de dos pasos que utiliza un algoritmo de selección de ensamblaje (BFES) y una función de decisión de ensamblaje analógica basada en Bayes (*A-Bayesian-Entropy*). Además, presenta resultados experimentales que muestran el rendimiento de los diferentes métodos propuestos en la detección de fallas en motores de vehículos. Se utiliza un conjunto de datos de entrenamiento de un modelo de vehículo para generar y validar las funciones de decisión del ensamblaje y luego probar su rendimiento en modelos de vehículos no vistos. Los resultados muestran que el enfoque propuesto, especialmente el BFES acompañado de la función de decisión *A-Bayesian-Entropy*, tiene una buena capacidad de generalización en modelos de vehículos no vistos.

El artículo utiliza la métrica de "error de prueba" para evaluar el rendimiento de los algoritmos en los datos de prueba de diferentes modelos de vehículos. Sin embargo, los valores específicos de rendimiento pueden variar según el conjunto de datos, el enfoque algorítmico utilizado y otras consideraciones del estudio.

3.1.4 Automotive signal fault diagnostics—part I: signal fault analysis, signal segmentation, feature extraction and quasi-optimal feature selection

El artículo [57] trata sobre el diagnóstico de fallos en señales automotrices, específicamente en el contexto del análisis de señales de vehículos. Discuten tareas

importantes como el análisis de comportamiento de señales defectuosas, la segmentación automática de señales, la extracción de características y la selección de características importantes. También se discuten diferentes tipos de fallas comunes en señales de vehículos, como magnitudes anormales, rолlos, ruido, fallas de dependencia, entre otras. Además, se presenta un algoritmo de segmentación de señales basado en la detección de bordes usando coeficientes de ondícula.

Se enfoca específicamente en el proceso de extracción de características importantes para el diagnóstico de fallos. Se discuten heurísticas y métodos para calcular características de señales que caracterizan comportamientos de fallos. Se presentan diferentes familias de características, como características básicas, concavidad/convexidad, coeficientes de ondaleta y otros. Además, se describe un algoritmo híbrido para seleccionar un conjunto cuasi óptimo de características utilizando medidas estadísticas y el método *Bayesian EM*. No menciona explícitamente las métricas de evaluación utilizadas para medir el rendimiento del diagnóstico de fallos en señales automotrices. Sin embargo, describe que se evalúa el rendimiento de los algoritmos y métodos propuestos utilizando conjuntos de datos de entrenamiento y prueba sin mencionar algún valor.

3.1.5 Immune systems inspired approach to anomaly detection and fault diagnosis for engines

El artículo [58] trata sobre la aplicación de un sistema de detección de anomalías basado en el Sistema de Modelos Múltiples de Estructura en Crecimiento (GSMMS, por sus siglas en inglés) para monitorear la dinámica del cigüeñal en un motor. El GSMMS combina redes auto-organizadas en crecimiento con algoritmos eficientes de estimación de parámetros de modelos lineales para crear un marco integrado para modelar e identificar sistemas dinámicos no lineales. describe cómo se utiliza este enfoque para monitorear y detectar desviaciones en la dinámica del cigüeñal durante el funcionamiento normal y anormal del motor, utilizando datos de entrenamiento y algoritmos de control específicos para cada región operativa del motor.

Utiliza varias métricas de evaluación para medir el rendimiento del sistema de detección de anomalías:

1. **Gráficos de Control de Media Móvil Exponencialmente Ponderada (EWMA):** Estos gráficos se utilizan en cada región operativa para detectar errores residuales fuera de los límites establecidos.
2. **Curvas Características Operativas del Receptor (ROC):** Estas curvas se utilizan para evaluar el rendimiento de la detección de anomalías en diferentes regiones y con diferentes aumentos en los coeficientes de amortiguación viscosa. Las ROC permiten analizar la tasa de verdaderos positivos frente a la tasa de falsos positivos a medida que se ajustan los parámetros del sistema de detección de anomalías.

- 3. Análisis de superposición de distribuciones de errores:** Se evalúa la superposición entre las distribuciones de errores actuales y las distribuciones que representan el comportamiento normal. Esto se utiliza para cuantificar la gravedad de la anomalía al observar cómo cambian las distribuciones de errores cuando los coeficientes de amortiguación viscosa se desvían de sus valores nominales.

Se menciona que el rendimiento del sistema de detección de anomalías mejora a medida que se utilizan más muestras y que casi se logra una detección perfecta cuando se alcanza una longitud de datos de alrededor de 1000 revoluciones. También menciona que el tiempo medio de detección es sustancialmente más corto para lograr una precisión de detección similar cuando se detectan aumentos del 20% y 30% en los coeficientes de amortiguación viscosa.

3.1.6 An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, Boosting, and Randomization

El artículo [59] explora y compara la eficacia de tres métodos para mejorar el rendimiento del algoritmo de árbol de decisión C4.5. Los métodos analizados son *Bagging*, *Boosting* y Randomización. El estudio se centra en construir un conjunto (ensemble) de clasificadores diversos y precisos. *Bagging* y *Boosting* son técnicas populares para construir conjuntos, mientras que la randomización propone una alternativa al manipular las decisiones internas del algoritmo base. La investigación compara estos métodos en diversas situaciones, tanto con poco ruido de clasificación como con ruido significativo. Los resultados muestran que en situaciones con poco o ningún ruido de clasificación, la randomización es competitiva con *Bagging* y ligeramente superior, pero no tan precisa como *Boosting*. Sin embargo, en situaciones con un ruido significativo de clasificación, *Bagging* supera a *Boosting* y, a veces, incluso a la randomización.

Se mencionan las tasas de error para diferentes dominios de aplicación, pero no se proporcionan valores específicos. Además, El artículo describe la diferencia en la tasa de error entre los diferentes métodos de ensemble (*Randomized C4.5*, *Bagged C4.5* y *Adaboosted C4.5*) y el clasificador base C4.5 en varios dominios de aplicación. Pero no incluye valores numéricos específicos para estas diferencias en la tasa de error.

3.1.7 Car failure fuzzy fault diagnostic system based on fault tree analysis

Este artículo [60] presenta un sistema de diagnóstico de fallas en automóviles basado en árboles de fallas difusas y análisis de árboles de fallas para tener en cuenta la imprecisión y la incertidumbre en la información y los datos relacionados con el componente defectuoso en el motor del automóvil. Dicho sistema utiliza la lógica difusa para identificar las fallas en los componentes y las perturbaciones en el proceso que pueden provocar el mal funcionamiento del sistema. Evalúa factores de certeza para verificar la secuencia de diagnóstico de las causas positivas. Está centrado en un contexto específico (un sistema

de diagnóstico de fallas en motores de automóviles) y utiliza un lenguaje técnico y especializado que puede resultar difícil de entender para personas que no estén familiarizadas con los conceptos de ingeniería de confiabilidad y sistemas de diagnóstico de fallas en vehículos.

Sin embargo, no proporciona métricas de precisión específicas en términos de valores numéricos como la precisión, sensibilidad, especificidad, etc.

3.1.8 Decision tree methods: applications for classification and prediction

El artículo *Decision tree methods* [61] proporciona una visión general de la aplicación de árboles de decisión en el contexto de la bioestadística y la psiquiatría, destacando sus ventajas y desventajas, así como sus aplicaciones prácticas en la investigación médica. Presenta algoritmos frecuentemente utilizados para desarrollar árboles de decisión, tales como CART, C4.5 entre otros.

Finalmente, el artículo destaca algunas aplicaciones comunes de los modelos de árboles de decisión en la investigación médica, como la selección de variables, la evaluación de la importancia relativa de las variables y la predicción. Es importante recalcar las limitaciones observadas, como el riesgo de sobreajuste y sub ajuste, así como la posibilidad de selección de variables correlacionadas. Además, que no especifica las métricas exactas de precisión o evaluación utilizadas en los ejemplos presentados.

3.1.9 Study and analysis of decision tree based classification algorithms

El artículo [62] explora el uso de algoritmos de árboles de decisión en el contexto de aprendizaje automático. Se centra en tres algoritmos específicos de árboles de decisión: ID3, C4.5 y CART. El objetivo principal del estudio es comparar estos algoritmos en términos de precisión, tiempo de ejecución y otras métricas utilizando un conjunto de datos de automóviles.

En la sección de experimentos, se utiliza un conjunto de datos de automóviles y se aplican los tres algoritmos para evaluar su desempeño en términos de tiempo de ejecución y precisión. Los resultados prácticos muestran que, aunque CART tiene el tiempo de ejecución más lento, tiene la mayor precisión en comparación con ID3 y C4.5.

Se destaca que el algoritmo CART es considerado el más preciso y adecuado para el conjunto de datos de automóviles utilizado en el estudio. También se mencionan posibles direcciones futuras para la investigación, como la implementación en servidores web y la integración de sistemas de calificación en línea para mejorar las predicciones.

Utiliza la matriz de confusión, *accuracy* y *precisión* para evaluar los algoritmos propuestos, obteniendo una exactitud (*accuracy*) de 97.11 y una precisión (*precision*) de .972

3.1.10A decision tree classifier for vehicle failure isolation

El artículo [63] se centra en la aplicación de un clasificador de árbol de decisión para detectar y aislar fallas en vehículos en el ámbito de la posventa de la industria automotriz. A medida que la industria automotriz ha experimentado grandes evoluciones con la introducción de sistemas electrónicos a bordo y el crecimiento de funciones basadas en la electrónica, la reparabilidad para los técnicos de posventa se ha vuelto más difícil.

El método propuesto utiliza el algoritmo ID3 para construir el árbol y aplica un método de poda basado en errores (EBP) para reducir la complejidad del árbol y evitar el sobreajuste. Además, se introduce un sistema de decisión adaptativo que combina las salidas de cada árbol para clasificar una instancia en una o varias clases conocidas o considerarla como perteneciente a una clase desconocida. Si bien se resalta la mejora gracias a la poda de árboles, existen otros algoritmos de poda con lo que podrían generarse nuevas comparativas.

El artículo utiliza métricas como la tasa de error y el *Novelty rejection percentage* para evaluar el rendimiento del clasificador. Se aplicó la validación cruzada de N-fold para obtener estimaciones de las métricas de rendimiento. Se encontró que el clasificador tenía un error promedio del orden del 3%, lo que es muy aceptable en tareas de clasificación, y es eficiente para detectar instancias de clases desconocidas.

3.1.11 Sistemas expertos basados en reglas

El trabajo Sistemas Expertos Basados en Reglas [64] aborda el uso de sistemas basados en reglas en situaciones deterministas en nuestra vida cotidiana, tales como el control de tráfico, la seguridad, las transacciones bancarias, etc. En este contexto, las reglas deterministas son una metodología simple utilizada en sistemas expertos. Hay que tener en cuenta que los sistemas basados en reglas evalúan todas las reglas aplicables para tomar una decisión y pueden volverse ineficientes en situaciones donde hay muchas reglas y la mayoría de ellas no son relevantes. Además de que en este trabajo se hace uso de un árbol de refutación, el cual puede volverse muy complejo, especialmente en conjuntos de datos grandes. Esto puede dificultar la interpretación del modelo y requerir un esfuerzo considerable para ajustar y optimizar los parámetros del árbol.

Se centra en discutir conceptos y principios relacionados con la coherencia de reglas, la coherencia de hechos y el tratamiento de la incertidumbre en sistemas expertos basados en reglas. Estas métricas son más cualitativas y conceptuales, y su evaluación generalmente se realiza mediante análisis lógico y conceptual en lugar de valores numéricos cuantitativos.

3.1.12 Desarrollo de un sistema experto para el diagnóstico de fallas automotrices

En este artículo [65] los autores hacen uso de un motor de inferencia y Prolog para crear el sistema DIMv3 como auxiliar en el proceso del diagnóstico. Además del hecho de que el sistema pueda ser utilizado por técnicos experimentados y personas con conocimientos básicos de mecánica automotriz lo hace accesible para una amplia gama de usuarios. Por otra parte, DIMv3 es una aplicación de escritorio donde se debe tener en cuenta la compatibilidad del sistema operativo, las actualizaciones manuales y la familiarización que se tenga al momento de instalar el software.

Evalúa el sistema experto desde diferentes perspectivas, incluyendo su desempeño, facilidad de uso, fiabilidad, capacidad de enriquecimiento continuo de la base de conocimiento y capacidad de actualización para mantenerse relevante en un entorno automotriz en constante evolución.

3.1.13 Sistema experto para el diagnóstico automotriz

En el trabajo de tesis Sistema Experto Para El Diagnostico Automotriz [66] se encuentra un prototipo de sistema experto basado en reglas para el diagnóstico automatizado exclusivamente del sistema de frenos hidráulicos del automóvil. Es relevante dado que los frenos son una parte crítica de la seguridad de un automóvil y su correcto funcionamiento es fundamental. Sin embargo, presenta dificultades de mantenimiento actualizar la base de conocimiento del sistema experto requiriendo tiempo y recursos para incorporar nuevas reglas, eliminar reglas obsoletas y garantizar que la información sea precisa y actualizada. Además, que no involucra métricas de evaluación.

3.1.14 Sistema experto para el diagnóstico de fallas en motores a inyección electrónica de vehículos

Este trabajo de tesis [67] se centra en desarrollar un prototipo de sistema experto probabilístico para el diagnóstico de fallas en motores a inyección electrónica de vehículos, que puede ayudar a los usuarios a diagnosticar problemas en motores de este tipo de vehículos de manera confiable y educativa. No obstante, como en cualquier sistema de diagnóstico, existe la posibilidad de que el sistema experto produzca falsos positivos (indicando una falla cuando no la hay) o falsos negativos (no detectar una falla real).

Es evaluada la interfaz del prototipo en una escala de "Mala" a "Excelente", con un 85% de aceptación para la interfaz. Además de la exactitud de los resultados proporcionados por el sistema experto: Evaluada en una escala de "Mala" a "Excelente", con un 90% de aceptación para los resultados proporcionados. Los resultados son evaluados por los expertos en el campo de la mecánica automotriz, específicamente por

los estudiantes de último año de la carrera técnica de Mecánica Automotriz del Instituto Técnico (ITAB)

3.1.15 Agente para el diagnóstico de motores de automóviles.

En este trabajo [68] se diseña un modelo de agente basado en conocimiento experto de Mecánica Automotriz para el diagnóstico de fallas mecánicas en motores de automóviles, utilizando lógica difusa y basado en reglas. Sin embargo, para que el sistema siga siendo relevante y preciso, debe actualizarse de manera regular a medida que cambian las tecnologías automotrices y se introducen nuevos modelos de vehículos. Esto puede ser una tarea continua y costosa.

Es evaluada la cantidad de información proporcionada: Con un 85% de aceptación para la cantidad de información proporcionada. Dicha información fue recopilada mediante encuestas o cuestionarios que contienen las preguntas relacionadas con la calidad y eficacia del sistema experto. Los resultados de estas evaluaciones fueron realizados por los estudiantes de último año de la carrera técnica de Mecánica Automotriz del Instituto Técnico (ITAB).

3.2 Foros web

Se muestra un análisis de dos plataformas de opinión en línea centradas en las fallas automotrices. Ambas valiosas para recopilar información sobre problemas automotrices. La popularidad de estas comunidades en línea destaca la necesidad de evaluar críticamente la fiabilidad de la información compartida.

3.2.1 Opinautos

Es un sitio web [69] y una comunidad en línea dedicada a compartir opiniones, comentarios y reseñas sobre fallas de automóviles. Proporciona una plataforma donde los usuarios describen sus problemas, para que otros miembros de la comunidad puedan ayudar. En la actualidad cuenta con más de 600.000 casos reportados. Desafortunadamente la información proporcionada por los usuarios puede variar en términos de precisión y veracidad. Algunos usuarios pueden proporcionar diagnósticos o soluciones incorrectas, lo que podría llevar a malentendidos o a la toma de decisiones erróneas.

3.2.2 CarComplaints

Es un sitio web [70] donde los propietarios de automóviles pueden informar y compartir problemas y fallas mecánicas que han experimentado con sus vehículos. También proporciona clasificaciones de problemas comunes por marca y modelo. A pesar de ello, la información proporcionada por los usuarios no siempre se verifica de manera exhaustiva, así que pueden surgir situaciones en las que los informes de problemas se basen en malentendidos o diagnósticos incorrectos.

3.3 Herramientas de diagnóstico

Dentro de las herramientas de diagnóstico automotriz se encuentra ScanatorPC, empresa 100% mexicana dedicada al diagnóstico automotriz con funcionalidades multimarca y específicas, basado en Windows.

3.3.1 ScanatorPC

Es un sistema de diagnóstico automotriz [71] basado en Windows con capacidades multimarca y específicas. Actualmente soporta diagnóstico de Motor, Transmisión, ABS, Dirección asistida, BCM, Inmovilizador y muchos sistemas más, para marcas como Opel, Peugeot, Renault, Ford, Honda, Nissan, Matiz, Mercedes, Daewoo y algunos más que día a día se van sumando. No obstante, las limitaciones en la cobertura y el horario de asistencia del soporte técnico son limitados.

3.4 Conclusión del estado del arte

En general, después de analizar trabajos de investigación, sitios web y de escritorio se demuestra la importancia de la tecnología y la colaboración en la industria automotriz. Todos proporcionan recursos valiosos para problemas de clasificación, para propietarios de vehículos y profesionales de la mecánica automotriz, pero es esencial reconocer sus limitaciones antes mencionadas, resaltando las reglas relevantes e irrelevantes, las actualizaciones del sistema, posibles necesidades de capacitación, presencia de sesgos en las reseñas de los usuarios, limitaciones en la cobertura y el horario del soporte técnico brindado.

Capítulo 4. Desarrollo

En este trabajo se proponen patrones para la detección de averías de motor basado en árboles de decisión, bases de datos, bases de conocimiento y lógica difusa. De tal manera que se dé un diagnóstico correcto a un cliente de un taller automotriz; mejorando las propuestas previamente abordadas en el estado del arte. Integrando la propuesta a la aplicación web Coq's Tuning Performance de un asistente inteligente (Chatbot), tal como se aprecia en la figura 4.1.

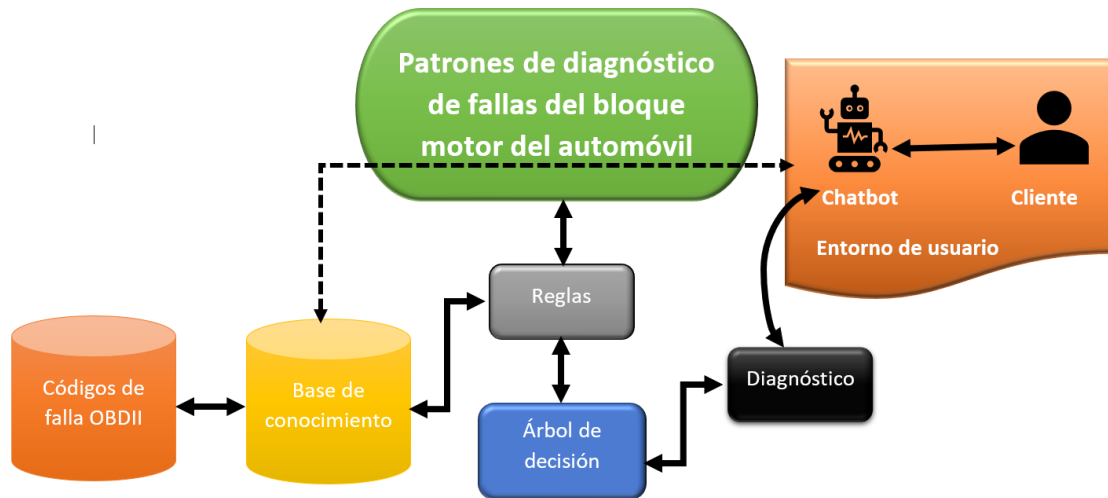


Figura 4.1. Esquema de la propuesta

En este capítulo, se describe el desarrollo de una solución para la detección de averías de motor en un taller automotriz, utilizando árboles de decisión, bases de datos, lógica difusa, y un asistente inteligente en forma de Chatbot. Se presenta una descripción detallada de cómo se estructuraron y utilizaron estas herramientas para construir una base de conocimiento sólida, capaz de ofrecer diagnósticos precisos y mejorar la experiencia del cliente. Además, se presentan los métodos implementados para la clasificación de datos y la interacción eficiente con los usuarios a través de una aplicación web personalizada.

4.1 Base de datos con códigos OBDII

Para gestionar el conjunto de códigos de avería OBDII se crea una base de datos no relacional, incluyendo la causa del problema presente, así como también posible solución incluyendo el o los elementos que generan dicha alerta, con la finalidad de que después de obtener un código de avería de la computadora del motor por medio de software, se pueda consultar directamente.

Dicha base de datos se desarrolló desde el panel de control creado con AJAX, el cual funciona de la siguiente manera:

Los códigos son consultados por medio del Chatbot, los cuales se encuentran en la base de datos y son desplegados con su respectiva causa y solución, tal como se aprecia en la figura 4.2 y 4.3. Se almacenan separados por “|” para que al consultar, el sistema pueda identificar y devolver el mismo resultado, ya sea que el usuario ingrese directamente el “código” o escriba la palabra “Código” seguido de los caracteres que lo componen.

| id | codigo | respuesta | solucion |
|-----|---------------------|--|---|
| 265 | P0262 Código P0262 | Circuito de Inyector en el cilindro 1 (Señal alta) | Cortocircuito a positivo en el cableado, inyector,... |
| 266 | P0263 Código P0263 | Contribución/Balance cilindro 1 | Cableado, sistema de combustible, módulo de contro... |
| 267 | P0264 Código P0264 | Circuito de Inyector en el cilindro 2 (Señal baja) | Cortocircuito a masa en el cableado, inyector, mód... |
| 268 | P0265 Código P0265 | Circuito de Inyector en el cilindro 2 (Señal alta) | Cortocircuito a positivo en el cableado, inyector,... |
| 269 | P0266 Código P0266 | Contribución/Balance cilindro 2 | Cableado, sistema de combustible, módulo de contro... |
| 270 | P0267 Código P0267 | Circuito de Inyector en el cilindro 3 (Señal baja) | Cortocircuito a masa en el cableado, inyector, mód... |
| 271 | P0268 Código P0268 | Circuito de Inyector en el cilindro 3 (Señal alta) | Cortocircuito a positivo en el cableado, inyector,... |
| 272 | P0269 Código P0269 | Contribución/Balance cilindro 3 | Cableado, sistema de combustible, módulo de contro... |
| 273 | P0270 Código P0270 | Circuito de Inyector en el cilindro 4 (Señal baja) | Cortocircuito a masa en el cableado, inyector, mód... |
| 274 | P0271 Código P0271 | Circuito de Inyector en el cilindro 4 (Señal alta) | Cortocircuito a positivo en el cableado, inyector,... |
| 275 | P0272 Código P0272 | Contribución/Balance cilindro 4 | Cableado, sistema de combustible, módulo de contro... |
| 276 | P0273 Código P0273 | Circuito de Inyector en el cilindro 5 (Señal baja) | Cortocircuito a masa en el cableado, inyector, mód... |
| 277 | P0274 Código P0274 | Circuito de Inyector en el cilindro 5 (Señal alta) | Cortocircuito a positivo en el cableado, inyector,... |
| 278 | P0275 Código P0275 | Contribución/Balance cilindro 5 | Cableado, sistema de combustible, módulo de contro... |
| 279 | P0276 Código P0276 | Circuito de Inyector en el cilindro 6 (Señal baja) | Cortocircuito a masa en el cableado, inyector, mód... |
| 280 | P0277 Código P0277 | Circuito de Inyector en el cilindro 6 (Señal alta) | Cortocircuito a positivo en el cableado, inyector,... |
| 281 | P0278 Código P0278 | Contribución/Balance cilindro 6 | Cableado, sistema de combustible, módulo de contro... |
| 282 | P0279 Código P0279 | Circuito de Inyector en el cilindro 7 (Señal baja) | Cortocircuito a masa en el cableado, inyector, mód... |

Figura 4.2. Base de datos de códigos OBDII

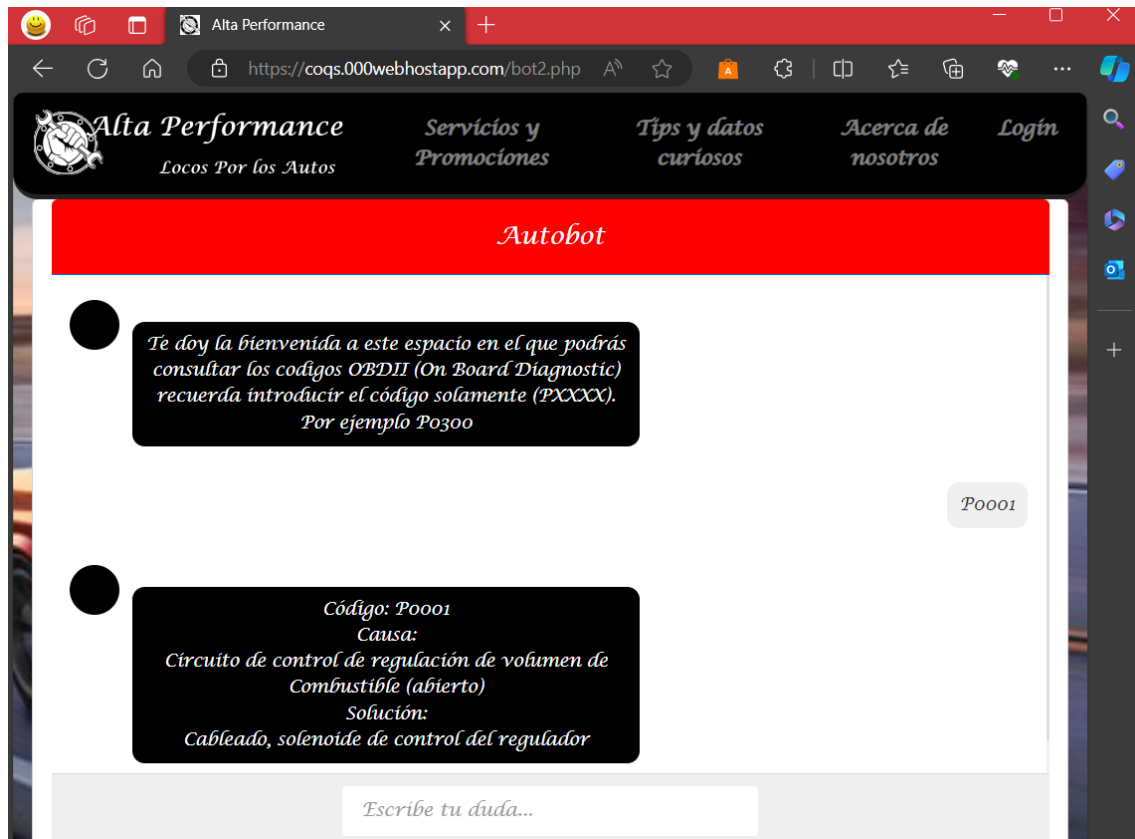


Figura 4.3. Consulta de códigos de avería desde el Chatbot

Si dicho código no se encuentra, es almacenado en una tabla temporal en la base de datos, para que el técnico especializado pueda darle atención, Así como también la capacidad de actualizar la información ya existente. Tal como se aprecia en la figura 4.4.

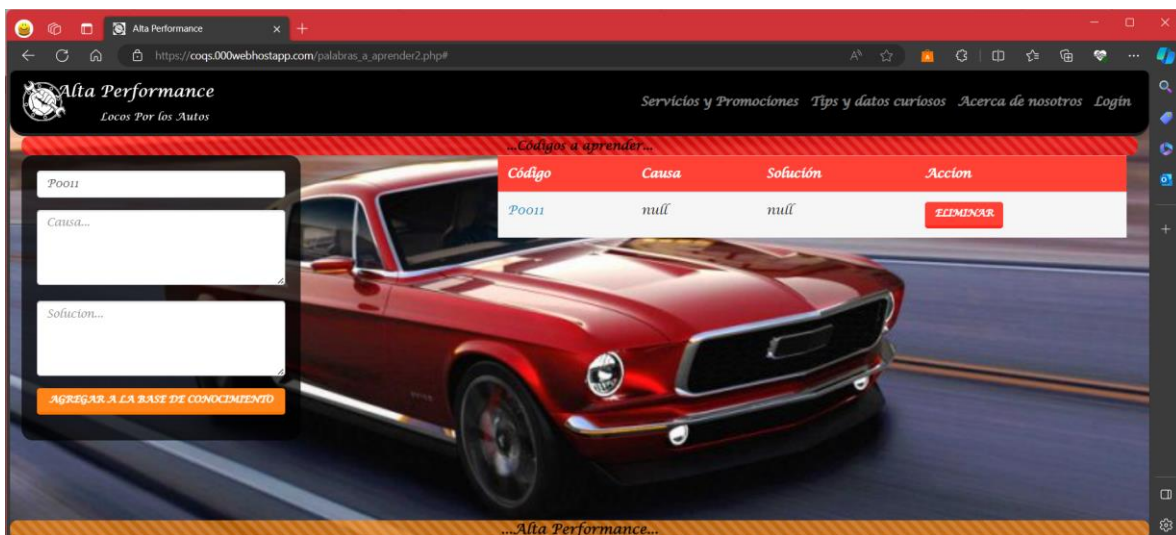


Figura 4.4. Panel de códigos desconocidos

Por lo que se desarrolló un panel de códigos desconocidos, donde se muestran aquellas averías que aún no se encuentran en la base de datos y un panel de actualización de averías para brindar actualización o corrección a los registros previamente ingresados, tal como se aprecia en la figura 4.5.

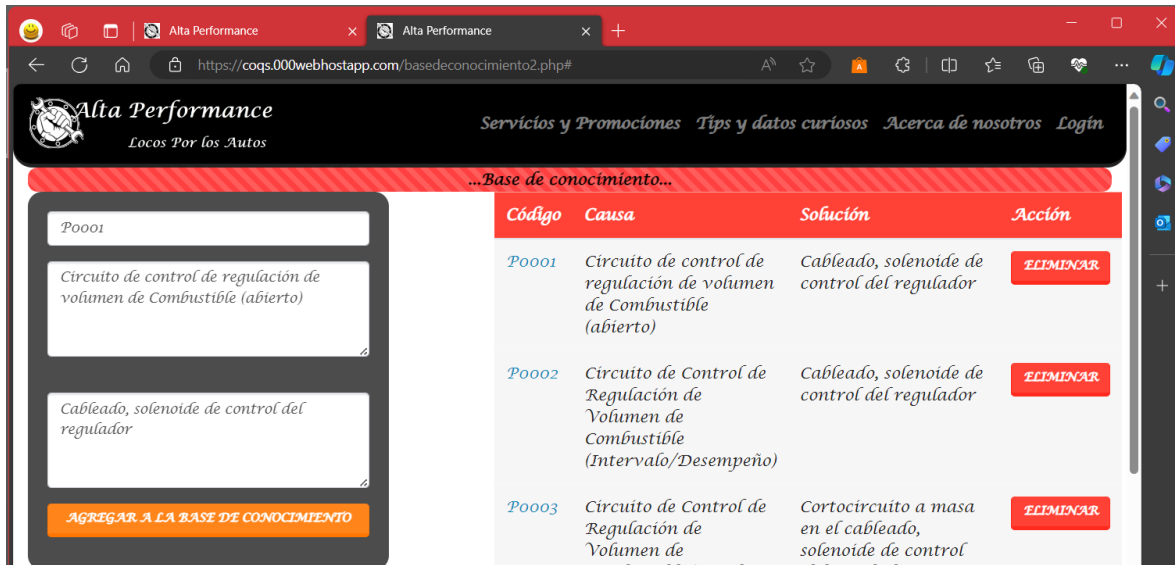


Figura 4.5. Panel de actualización de códigos

Ya que, año con año siguen saliendo nuevos modelos de vehículos automotrices los cuales incorporan un conjunto más amplio de sensores y actuadores para mejorar su rendimiento. Y es sumamente importante mantener actualizado el conjunto de datos de códigos de averías, porque pueden experimentar cambios y adiciones a lo largo del tiempo como resultado de la evolución de la tecnología automotriz y las necesidades de diagnóstico.

A pesar de que las tablas no están actualmente relacionadas, la capacidad de definir relaciones en el futuro añade flexibilidad para ampliar el sistema según sea necesario.

4.2 Construcción de la base de conocimiento

Para el diagnóstico de averías del bloque motor, además de considerar los códigos OBD-II se deben tener en cuenta ciertas características o condiciones de manejo, un conjunto de reglas y hechos que alimenten al algoritmo del árbol de decisión.

Dichas condiciones de averías se dividen en 4 grupos:

- Fallas al momento de encender la unidad.
- Falla durante el desempeño
- Falla de calentamiento
- Falla del sistema de emisiones contaminantes

Cada grupo engloba diversos elementos del bloque motor (Sensores, actuadores y partes mecánicas) que pueden estar trabajando fuera de rango y dependiendo de los valores obtenidos es como se identifica al responsable de la falla.

Para facilitar el acceso a los datos, estos son almacenados en un archivo CSV, para su posterior aplicación como conjunto de entrenamiento del modelo, tal como se aprecia en la figura 4.6.

| sistema_con_Voltaje_ | encendido | Voltaje_apagado | Marcha | Chispa | Inyeccion_coi | Tiempo | Compresion | Codigos_OBD | Sist_aire_libre | Presion_com | Pulso | Iny_desemp | Nivel_ant | Fuga | Termost_abre | Ventil_act |
|----------------------|-----------|-----------------|---------------|------------------|---------------|-------------|------------|-------------|-----------------|-------------|-------------|------------|-----------|---------|--------------|------------|
| Alternador mi | V>12 | Da marcha | Hay chispa | Si hay inyección | A tiempo | Compresion> | no | Libre | Alta | Hay pulso | Buen desemp | A nivel | No fuga | abre | Si_activa | |
| Motor de arranque | V>12 | No da marcha | Hay chispa | Si hay inyección | A tiempo | Compresion> | no | Libre | Alta | Hay pulso | Buen desemp | A nivel | No fuga | abre | Si_activa | |
| Batería mala | V<12 | Da marcha | Hay chispa | Si hay inyección | A tiempo | Compresion> | no | Libre | Alta | Hay pulso | Buen desemp | A nivel | No fuga | abre | Si_activa | |
| Desuso | V>12 | Da marcha | Hay chispa | Si hay inyección | A tiempo | Compresion> | no | Libre | Alta | Hay pulso | Buen desemp | A nivel | No fuga | abre | Si_activa | |
| Problema en el | V>12 | Da marcha | No hay chispa | Si hay inyección | A tiempo | Compresion> | no | Libre | Alta | Hay pulso | Buen desemp | A nivel | No fuga | abre | Si_activa | |
| Bomba de aceite | V>12 | Da marcha | Hay chispa | No hay inyección | A tiempo | Compresion> | no | Libre | Alta | Hay pulso | Buen desemp | A nivel | No fuga | abre | Si_activa | |
| Poner a tiempo | V>12 | Da marcha | Hay chispa | Si hay inyección | A tiempo | Compresion> | no | Libre | Alta | Hay pulso | Buen desemp | A nivel | No fuga | abre | Si_activa | |
| Abrir motor | V>12 | Da marcha | Hay chispa | Si hay inyección | A tiempo | Compresion> | no | Libre | Alta | Hay pulso | Buen desemp | A nivel | No fuga | abre | Si_activa | |
| Ir al consultorio | V>12 | Da marcha | Hay chispa | Si hay inyección | A tiempo | Compresion> | si | Libre | Alta | Hay pulso | Buen desemp | A nivel | No fuga | abre | Si_activa | |
| Revisar sistema | V>12 | Da marcha | Hay chispa | Si hay inyección | A tiempo | Compresion> | no | obstruido | Alta | Hay pulso | Buen desemp | A nivel | No fuga | abre | Si_activa | |
| Bomba y línea | V>12 | Da marcha | Hay chispa | Si hay inyección | A tiempo | Compresion> | no | Libre | baja | Hay pulso | Buen desemp | A nivel | No fuga | abre | Si_activa | |
| Bujías, cables, | V>12 | Da marcha | No hay chispa | Si hay inyección | A tiempo | Compresion> | no | Libre | Alta | Hay pulso | Buen desemp | A nivel | No fuga | abre | Si_activa | |
| Sensor CKP o | V>12 | Da marcha | Hay chispa | Si hay inyección | A tiempo | Compresion> | no | Libre | Alta | sin pulso | Buen desemp | A nivel | No fuga | abre | Si_activa | |
| Inyector deficiente | V>12 | Da marcha | Hay chispa | Si hay inyección | A tiempo | Compresion> | no | Libre | Alta | Hay pulso | mal desemp | A nivel | No fuga | abre | Si_activa | |
| Rellenar a nivel | V>12 | Da marcha | Hay chispa | Si hay inyección | A tiempo | Compresion> | no | Libre | Alta | Hay pulso | Buen desemp | Bajo | No fuga | abre | Si_activa | |
| Cambiar empuje | V>12 | Da marcha | Hay chispa | Si hay inyección | A tiempo | Compresion> | no | Libre | Alta | Hay pulso | Buen desemp | A nivel | Fuga | abre | Si_activa | |
| Cambiar Termostato | V>12 | Da marcha | Hay chispa | Si hay inyección | A tiempo | Compresion> | no | Libre | Alta | Hay pulso | Buen desemp | A nivel | No fuga | No abre | Si_activa | |
| Cambiar ventilador | V>12 | Da marcha | Hay chispa | Si hay inyección | A tiempo | Compresion> | no | Libre | Alta | Hay pulso | Buen desemp | A nivel | No fuga | abre | No_activa | |
| Sensor O2 1 banco 1 | V>12 | Da marcha | Hay chispa | Si hay inyección | A tiempo | Compresion> | no | Libre | Alta | Hay pulso | Buen desemp | A nivel | No fuga | abre | Si_activa | |
| Sensor O2 2 banco 1 | V>12 | Da marcha | Hay chispa | Si hay inyección | A tiempo | Compresion> | no | Libre | Alta | Hay pulso | Buen desemp | A nivel | No fuga | abre | Si_activa | |
| Sensor O2 1 banco 2 | V>12 | Da marcha | Hay chispa | Si hay inyección | A tiempo | Compresion> | no | Libre | Alta | Hay pulso | Buen desemp | A nivel | No fuga | abre | Si_activa | |
| Sensor O2 2 banco 2 | V>12 | Da marcha | Hay chispa | Si hay inyección | A tiempo | Compresion> | no | Libre | Alta | Hay pulso | Buen desemp | A nivel | No fuga | abre | Si_activa | |
| Alternador mi | V>12 | Da marcha | Hay chispa | Si hay inyección | A tiempo | Compresion> | no | Libre | Alta | Hay pulso | Buen desemp | A nivel | No fuga | abre | Si_activa | |
| Motor de arranque | V>12 | No da marcha | Hay chispa | Si hay inyección | A tiempo | Compresion> | no | Libre | Alta | Hay pulso | Buen desemp | A nivel | No fuga | abre | Si_activa | |
| Batería mala | V>12 | Da marcha | Hay chispa | Si hay inyección | A tiempo | Compresion> | no | Libre | Alta | Hay pulso | Buen desemp | A nivel | No fuga | abre | Si_activa | |
| Desuso | V>12 | Da marcha | Hay chispa | Si hay inyección | A tiempo | Compresion> | no | Libre | Alta | Hay pulso | Buen desemp | A nivel | No fuga | abre | Si_activa | |
| Problema en el | V>12 | Da marcha | No hay chispa | Si hay inyección | A tiempo | Compresion> | no | Libre | Alta | Hay pulso | Buen desemp | A nivel | No fuga | abre | Si_activa | |

Figura 4.6. Base de conocimiento

4.2.1 Hechos:

Los hechos en este contexto son las condiciones observadas en el sistema. Estos hechos son los valores que se encuentran en las diferentes columnas para cada fila de la tabla. Tales como:

- Voltaje encendido: V>12
- Voltaje apagado: V>12
- Marcha: Da marcha
- Chispa: Hay chispa
- Inyección de combustible: Si hay inyección
- Tiempo: A tiempo

Estos son los hechos que describen el estado del sistema en un momento dado.

4.2.2 Reglas:

Las reglas se pueden inferir de las relaciones entre los hechos y las acciones o diagnósticos sugeridos. Por ejemplo:

- Regla 1: "Si el voltaje encendido es mayor a 12V, el voltaje apagado es mayor a 12V, hay marcha, hay chispa, hay inyección de combustible, el

tiempo es correcto, la compresión es mayor a 120, y no hay códigos OBDII, entonces el sistema está funcionando correctamente o el problema no está relacionado con estos parámetros."

- Regla 2: "Si el voltaje encendido es mayor a 12V, el voltaje apagado es mayor a 12V, no hay chispa, y todo lo demás es correcto, entonces hay un problema en el sistema de encendido."

Las reglas son la lógica que se utiliza para hacer diagnósticos o tomar decisiones basadas en los hechos observados.

4.3 Desarrollo de un árbol de decisión

Se implementa CART (*Classification and Regression Trees*) para crear un árbol de decisión, el cual funciona dividiendo el conjunto de datos en subconjuntos más pequeños de manera recursiva. Cada división se realiza seleccionando la característica y el umbral que mejor separan los datos, como la ganancia de información en el caso de la clasificación.

El conjunto de datos proporcionado en formato CSV se utiliza como conjunto de entrenamiento. Este conjunto de entrenamiento es esencialmente una colección de ejemplos que se utiliza para enseñar al modelo de árbol de decisión cómo realizar predicciones basadas en ciertas características o condiciones de manejo observadas.

Se realiza la separación de atributos (X) y etiquetas (y) facilitar el proceso de entrenamiento del modelo, donde:

- X se refiere a todas las columnas en data excepto 'sistema_con_averia'. Estas son las características que el modelo utilizará para hacer predicciones.
- y se refiere a la columna 'sistema_con_averia'. Estos son los valores que el modelo intentará predecir.

Muchos algoritmos de aprendizaje automático, incluidos los árboles de decisión, trabajan mejor con datos numéricos. Sin embargo, en conjuntos de datos del mundo real, a menudo encuentras características categóricas, es decir, variables que pueden tomar un conjunto limitado de categorías o etiquetas.

El *One-Hot Encoding* es una técnica que convierte las variables categóricas en un formato que puede ser proporcionado a algoritmos de aprendizaje automático para mejorar su rendimiento. La idea principal es crear nuevas columnas binarias para cada categoría única presente en la característica categórica original.

Este proceso asegura que todas las características, incluso las categóricas, estén representadas de una manera numérica que puede ser interpretada por el algoritmo de árbol de decisión durante el entrenamiento y la predicción.

4.3.1 Creación y poda de los árboles de decisión

Una vez realizado el *One-Hot Encoding* se crean y entrenan cuatro modelos de árboles de decisión con diferentes métodos de poda: CCP (poda por complejidad del costo), *Max Depth* (poda por altura), *Min Samples Leaf* (poda por número mínimo de muestras en hojas) y *Min Samples Split* (poda por número mínimo de muestras para dividir un nodo). Cada modelo se entrena utilizando los datos codificados *X_encoded* y las etiquetas y tal como se aprecia en la figura 10.

```
# Método de poda por complejidad del costo (CCP)
clf_ccp = DecisionTreeClassifier(ccp_alpha=0.04)
clf_ccp = clf_ccp.fit(X_encoded, y)

# Método de poda por altura
clf_max_depth = DecisionTreeClassifier(max_depth=7)
clf_max_depth = clf_max_depth.fit(X_encoded, y)

# Método de poda por número mínimo de muestras en hojas
clf_min_samples_leaf = DecisionTreeClassifier(min_samples_leaf=3)
clf_min_samples_leaf = clf_min_samples_leaf.fit(X_encoded, y)

# Método de poda por número mínimo de muestras para dividir un nodo
clf_min_samples_split = DecisionTreeClassifier(min_samples_split=10)
clf_min_samples_split = clf_min_samples_split.fit(X_encoded, y)
```

Figura 4.7. Métodos de poda de árboles de decisión

Se crea una instancia de la clase *DecisionTreeClassifier* proporcionada por la biblioteca *scikit-learn*. Esta instancia es el clasificador del árbol de decisión que se utilizará para entrenar el modelo y realizar predicciones.

Se codifica la nueva instancia utilizando el mismo proceso de codificación que se utilizó para los datos de entrenamiento. Se asegura que las columnas de la nueva instancia coincidan con las columnas del conjunto de entrenamiento. Se realizan predicciones con cada uno de los modelos entrenados para la nueva instancia utilizando el método *predict* sobre nuevas instancias (conjuntos de características). El modelo utiliza las relaciones y patrones aprendidos durante el entrenamiento para predecir si una nueva instancia está asociada a un sistema con avería o no.

No se trabaja directamente con datos históricos, sino con una nueva instancia capturada por el usuario para generar una inferencia o predicción.

Con esto se aplica el modelo aprendido a un ejemplo práctico. Esto es similar al proceso de aprendizaje, pero en lugar de utilizar el conjunto de datos completo para entrenar el modelo, se aplica el conocimiento adquirido por el modelo a una instancia

específica para hacer una predicción sobre esa instancia, pero en un contexto más específico y práctico.

Posteriormente se evalúan los modelos utilizando una matriz de confusión, se obtienen los verdaderos positivos, verdaderos negativos y se calculan la precisión (*precision*), la exactitud (*accuracy*), la tasa de error (*error rate*), *Novelty rejection percentage* y puntaje promedio de validación cruzada (*Mean Cross-Validation Score*) de cada modelo.

El árbol CART con poda CCP generado se muestra a continuación en la figura 4.8 y 4.9. El resultado final se encuentra en las hojas del árbol, donde se asigna una clase o categoría específica a una instancia de entrada basada en las características proporcionadas.

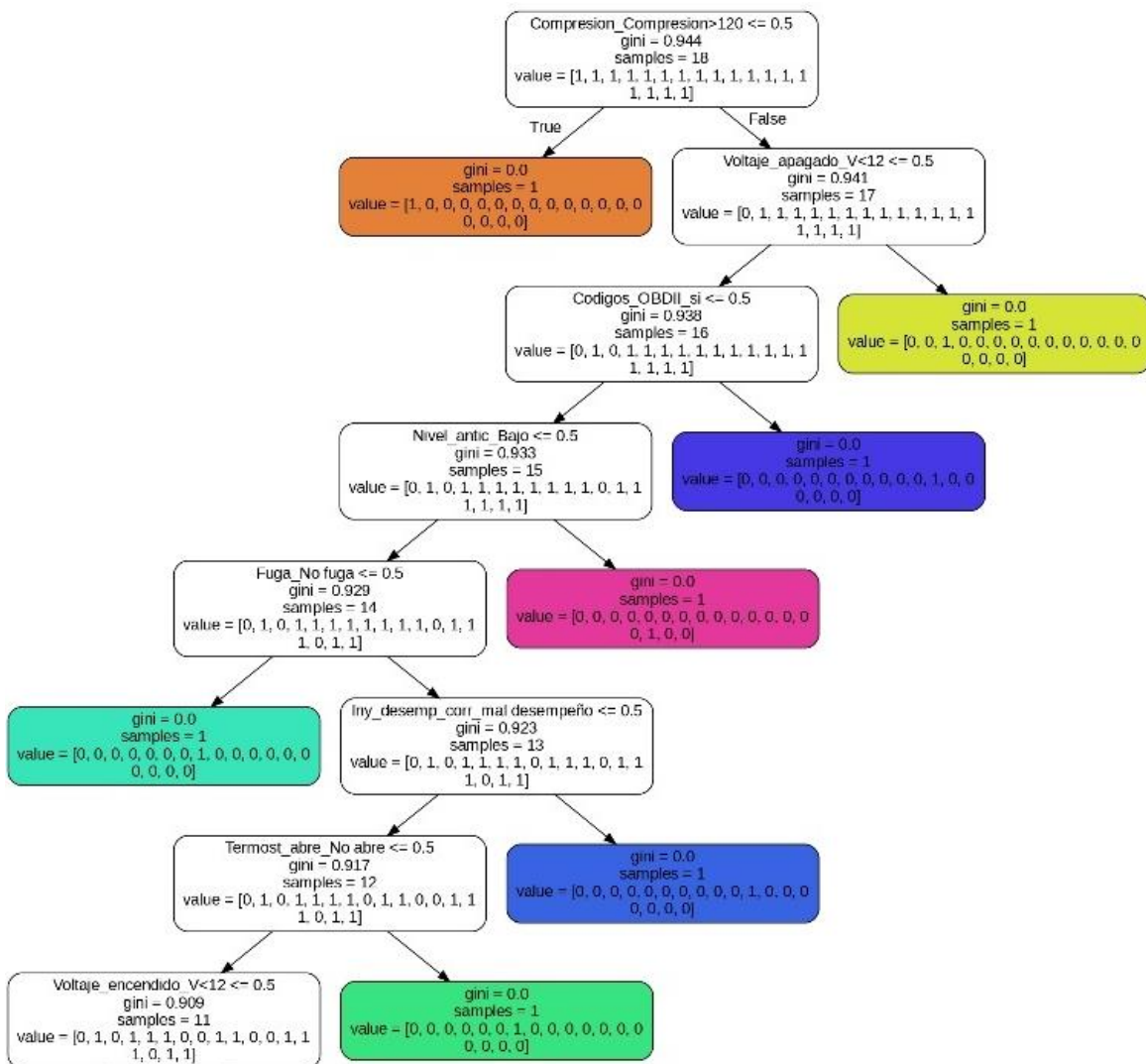


Figura 4.8. Árbol de decisión CART con poda CCP 2.

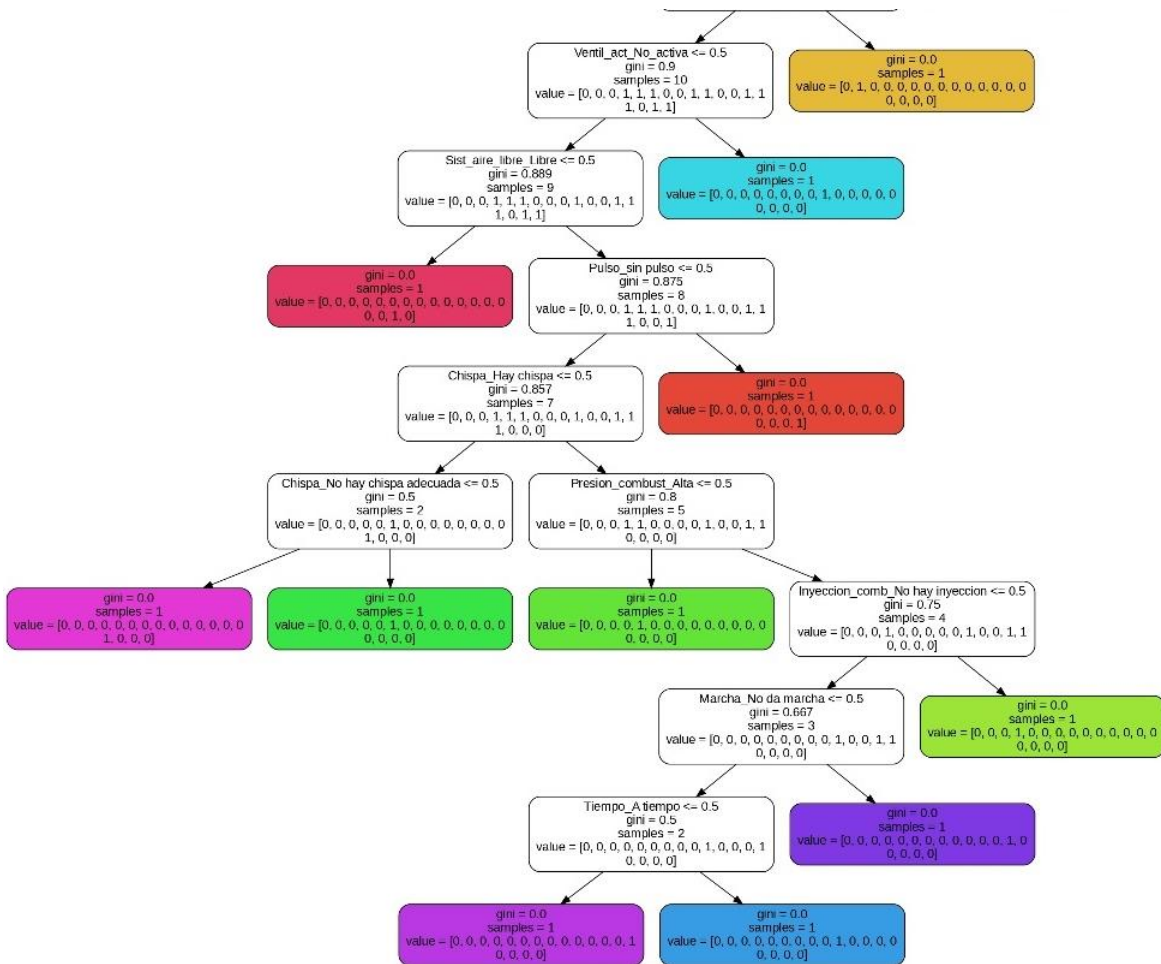


Figura 4.9. Árbol de decisión CART con poda CCP 2.

4.4 Clasificación por medio de lógica difusa

Controlar y conocer la temperatura del refrigerante del motor de un automóvil es fundamental por varias razones. Principalmente para prevenir el sobrecalentamiento del motor, ya que la temperatura del refrigerante del motor es crucial para mantener el motor funcionando dentro de un rango seguro. Si la temperatura del refrigerante aumenta demasiado, puede provocar la evaporación del líquido refrigerante o incluso la ebullición, lo que puede causar daños en el sistema de enfriamiento, como la formación de burbujas de aire, la pérdida de presión y la corrosión de los componentes además de provocar daños graves, como la fusión de componentes internos, deformación de piezas metálicas e incluso la avería completa del motor.

Por otra parte, la optimización del rendimiento y la eficiencia, pues el motor funciona de manera óptima dentro de ciertos rangos de temperatura. Si la temperatura del refrigerante es demasiado baja, el motor puede operar de manera menos eficiente, lo que resulta en un menor rendimiento y mayor consumo de combustible. Por otro lado, si la

temperatura es demasiado alta, puede provocar pérdida de potencia y un aumento en las emisiones contaminantes.

Y es aquí donde la lógica difusa juega un papel crucial, ya que nos permite clasificar la temperatura de un vehículo no solo como baja o alta. Sino con más valores para poder clasificar las temperaturas en categorías utilizando funciones de pertenencia difusa.

Lo primero que se hace es definir el universo de temperatura (*temperatura_universe*) como un rango de valores de 0 a 150 grados Celsius con incrementos de 1 grado tal como se aprecia en la figura 4.10.

```
temperatura_universe = np.arange(0, 150, 1)
```

Figura 4.10. Universo de temperatura

Esto se hace ya que los motores de combustión interna actuales están fabricados en su mayoría con cabezas de aluminio y no soportan valores por encima de esa temperatura. Antiguamente eran hechos de fierro colado o hierro fundido, por lo que aguantaban temperaturas de funcionamiento de hasta aproximadamente 600-700°C sin experimentar deformación permanente.

Se definen las funciones de membresía difusa para las categorías de baja, media y alta temperatura utilizando *fuzz.trimf*. Estas funciones de membresía asignan grados de pertenencia difusa a cada temperatura en el universo de discurso tal como se aprecia en la figura 4.11.

```
baja = fuzz.trimf(temperatura_universe, [0, 40, 80])  
media = fuzz.trimf(temperatura_universe, [75, 90, 110])  
alta = fuzz.trimf(temperatura_universe, [105, 130, 150])
```

Figura 4.11. Funciones de membresía

Se realiza la clasificación de la temperatura, se calcula su grado de pertenencia a las categorías de baja, media y alta temperatura utilizando *fuzz.interp_membership* tal como se aprecia en la figura 4.12.

```
pertenencia_baja = fuzz.interp_membership(temperatura_universe, baja, temp)  
pertenencia_media = fuzz.interp_membership(temperatura_universe, media, temp)  
pertenencia_alta = fuzz.interp_membership(temperatura_universe, alta, temp)
```

Figura 4.12. Clasificación de la temperatura

Los grados de pertenencia se imprimen junto con la temperatura, lo que proporciona una clasificación difusa de la temperatura en las categorías de baja, media y alta.

4.5 Asistente personalizado inteligente: Chatbot

En este trabajo de tesis de maestría se desarrolló un Chatbot por medio de técnicas AJAX (*Asynchronous Javascript And XML*) como asistente personalizado inteligente con la finalidad de mejorar la atención del cliente del taller automotriz Coq's Tuning Performance a través de 2 módulos:

- El módulo de dudas o preguntas frecuentes acerca de los servicios automotrices.
- El módulo de diagnóstico de averías presentes en el bloque motor como experto del dominio.

4.5.1 Módulo de atención a preguntas frecuentes

Una de las principales ventajas de un Chatbot es la disponibilidad las 24 horas del día, los 7 días de la semana. Es por ello que se crea como un asistente virtual. Si bien el sitio web de Coq's Tuning performance cuenta con un chat entre los clientes y el administrador del Taller, el cual depende de los horarios de atención para responder las peticiones.

Y es aquí donde el Chatbot mejora la atención brindada no solo a los internautas, sino también a los clientes del taller mecánico, despejando dudas y preguntas frecuentes, hasta responder acerca de estatus de citas o dirigir al cliente hasta el taller por medio de Google Maps. Tal como se aprecia en la figura 4.13.

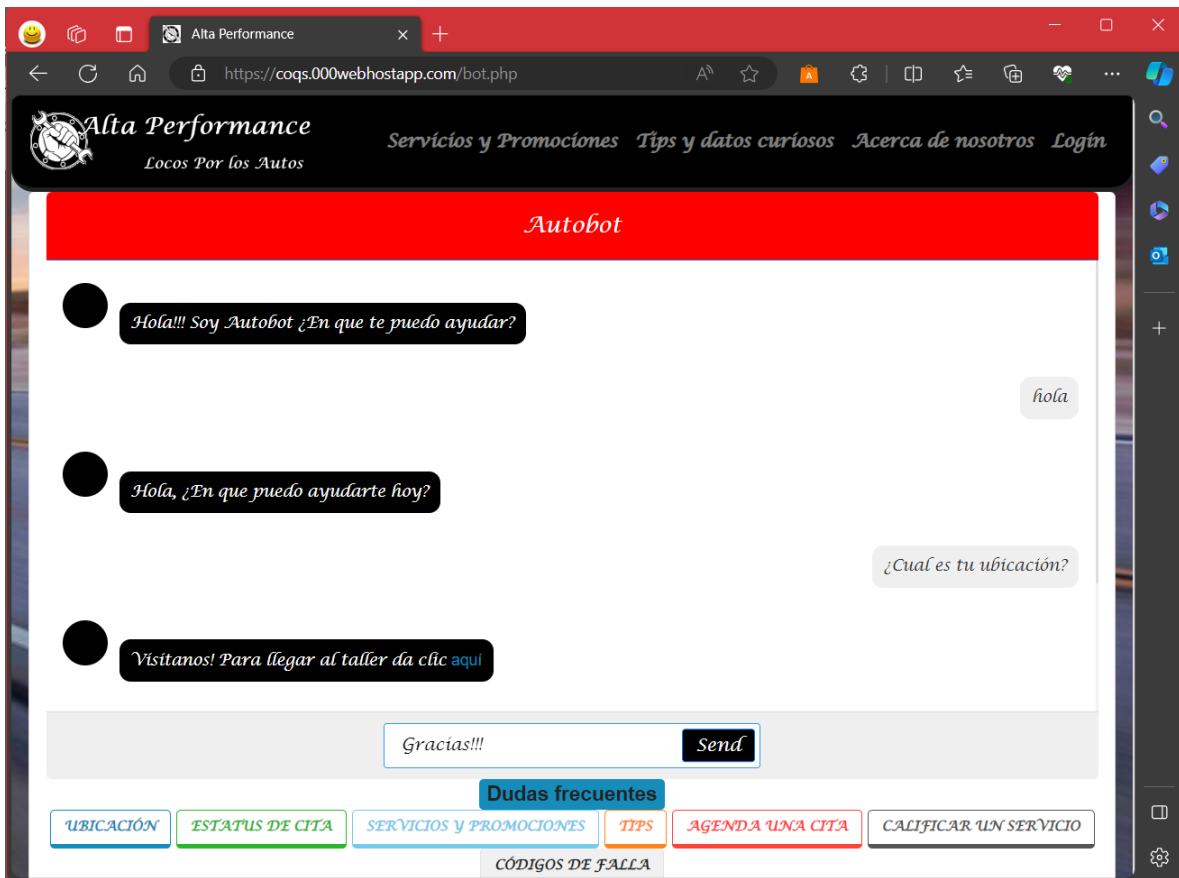


Figura 4.13. Asistente personalizado Chatbot Internauta

Con AJAX, se actualiza el contenido de la página web sin necesidad de recargarla por completo. Esto significa que los mensajes del Chatbot pueden aparecer instantáneamente en la ventana del chat a medida que el usuario interactúa con él, proporcionando una experiencia de usuario más fluida y en tiempo real. Ya que AJAX permite que el navegador web envíe y reciba datos del servidor en segundo plano, sin interferir con otras acciones que pueda estar realizando el usuario en la página web. Esto es ideal para un Chatbot, ya que puede enviar mensajes al servidor y recibir respuestas sin interrumpir la conversación en curso.

Con ayuda de AJAX se envían los mensajes del usuario al servidor a través de un botón, para el que hay una función a la escucha de que suceda el evento de ser presionado. Cuando el usuario hace clic en este botón, se ejecuta el código dentro de la función de la siguiente manera:

- Se obtiene el valor del elemento con el id "data". Este elemento generalmente es un input donde el usuario escribe su mensaje.
- Se construye el HTML para mostrar el mensaje del usuario en la ventana del chat. Aquí, el valor del mensaje del usuario se inserta en el HTML.
- Se agrega el mensaje del usuario a la ventana del chat, para finalmente limpiar el input después de que el usuario ha enviado el mensaje.

- Se inicia una solicitud AJAX al servidor. Aquí es donde se envía el mensaje del usuario al servidor y se recibe la respuesta del servidor.
 - *url*: Especifica la URL del script PHP (en este caso, 'message.php') que procesará la solicitud.
 - *type*: Especifica el método de la solicitud (en este caso, 'POST').
 - *data*: Especifica los datos que se enviarán al servidor. En este caso, se envía el texto del mensaje del usuario.
 - *success*: Una función que se ejecuta si la solicitud AJAX es exitosa. Dentro de esta función, se maneja la respuesta del servidor y se agrega la respuesta a la ventana del chat.

Finalmente se desplaza la ventana del chat hacia abajo para mostrar el último mensaje enviado o recibido. Dicho código se aprecia en la figura 4.14.

```

<script>
$(document).ready(function(){
  $("#send-btn").on("click", function(){
    $value = $("#data").val();
    $msg = '<div class="user-inbox inbox"><div class="msg-header"><p>'+ $value +'</p></div></div>';
    $(".form").append($msg);
    $("#data").val('');

    // start ajax code
    $.ajax({
      url: 'message.php',
      type: 'POST',
      data: 'text='+$value,
      success: function(result){
        $replay = '<div class="bot-inbox inbox"><div class="icon" style="background-color:black;"><i class="fas fa-user"></i></div><div class="msg-header" ><p style="background-color:black;">'+ result +'</p></div></div>';
        $(".form").append($replay);
        // when chat goes down the scroll bar automatically comes to the bottom
        $(".form").scrollTop($(".form")[0].scrollHeight);
      }
    });
  });
});
</script>

```

Figura 4.14. Función con AJAX del Chatbot

4.5.2 Asistente de diagnóstico de averías en el bloque motor

Para la implementación del asistente de diagnóstico de averías se toma en cuenta principalmente la forma en que el usuario introduce la información. Ya que la elección de los elementos de entrada de datos influye significativamente en la experiencia del usuario y en la calidad de los datos recopilados para generar el diagnóstico de la falla del vehículo. Es importante seleccionar los tipos de entrada que sean intuitivos, fáciles de usar y que minimicen la posibilidad de sesgos o errores por parte del usuario.

Para recopilar la información acerca de los síntomas o condiciones presentes en la unidad el Chatbot solicita información haciendo uso de opciones predefinidas como *checkboxes*, botones de opción o barras de rango para ser más efectivo que simplemente solicitar al usuario que escriba sus respuestas en un campo de texto abierto. Estos elementos de entrada estructurados ayudan significativamente a guiar al usuario y reducir

la ambigüedad en sus respuestas, lo que a su vez mejora la precisión y relevancia de la información recopilada.

En este trabajo los problemas del bloque motor se dividen en 4 grupos:

- Problemas al encender el auto
- Problemas durante el desempeño
- Problemas de calentamiento
- Problemas con el sistema de emisiones contaminantes

Dentro de los problemas antes mencionados se encuentra la opción de acceder a la base de datos de códigos de avería. Ya que, si bien un código de avería OBDII almacenado en la ECU no indica la solución de la falla, es capaz de indicar el conjunto de elementos donde los sensores o actuadores están trabajando fuera de rango para hacer las pruebas y revisiones pertinentes tal cual se indican en el apartado de posible causa y solución en la base de datos. Además, la presencia de un testigo o alerta en el *cluster* de instrumentos (tablero) del vehículo es de vital importancia y conocer la gravedad o significado del mismo es crucial. Dichos testigos, así como su significado y sugerencia se pueden consultar directamente en el apartado de alertas o testigos.

Por otra parte, la principal problemática en el diagnóstico de una avería es la ausencia de códigos OBDII, teniendo únicamente la falla presente. Es aquí donde entra la interacción con la base de conocimiento y el modelo de inteligencia artificial que es el árbol de decisión para poder analizar las observaciones o lecturas presentes y realizar una predicción para generar un diagnóstico.

Para no saturar al usuario con tantas preguntas y pueda perderse fácilmente, Estos pequeños formularios se agrupan en tarjetas, las cuales tienen botones, barras de rango y *checkboxes* como se mencionó anteriormente, con la finalidad de evitar sesgos en la información introducida y mejorar la interacción humano-computadora.

4.5.2.1 Problemas para poner en marcha el vehículo

Para el primer grupo de problemas se toma la condición inicial en donde el vehículo enciende o no enciende.

4.5.2.1.1 Condición inicial: Vehículo enciende

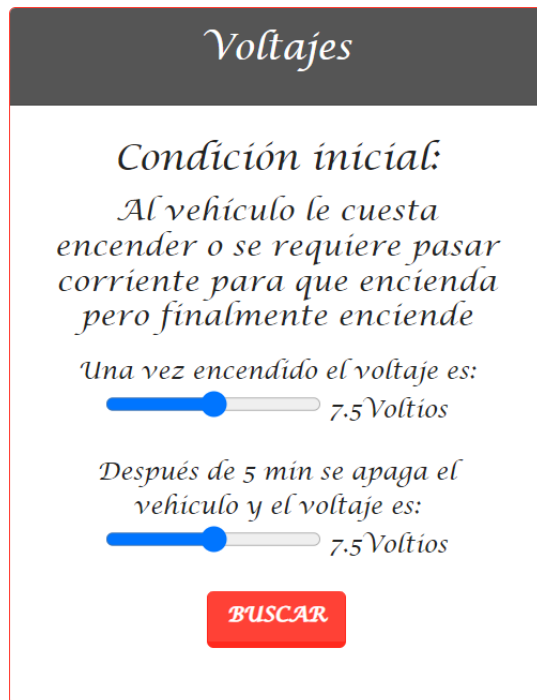
En esta situación el vehículo presenta dificultades para encender o se requiere pasar batería, pero finalmente enciende.

Para ello el técnico automotriz deberá asegurarse de que se cumpla la condición inicial de un voltaje mayor o igual a 12 Volts. Teniendo esta condición verificada se realizan estas 2 acciones:

- Se procede a dar arranque al vehículo y al encender se mide el voltaje y se registra en el formulario.

- Después de 5 minutos se apaga el vehículo y se toma una nueva lectura del voltaje de batería y se registra en el formulario.

Con dichos datos capturados en el formulario a través de unas barras de rango (ver figura 4.15), se realiza la consulta y se genera la predicción o diagnóstico.



The image shows a digital form titled "Voltajes" with a dark header. The main content area is white and contains the following text and elements:

- Condición inicial:**
Al vehículo le cuesta encender o se requiere pasar corriente para que encienda pero finalmente enciende
- Una vez encendido el voltaje es:*
A slider bar with a blue handle positioned at approximately 25% of the range, followed by the text "7.5 Voltios".
- Después de 5 min se apaga el vehículo y el voltaje es:*
A slider bar with a blue handle positioned at approximately 25% of the range, followed by the text "7.5 Voltios".
- A red button with the text "BUSCAR" in white capital letters.

Figura 4.15. Barras de rango para voltajes.

4.5.2.1.2 Condición inicial: Vehículo no enciende

En esta situación el vehículo presenta dificultades para encender o se requiere pasar batería, pero no es posible que encienda o se mantenga encendido.

Para ello el técnico automotriz deberá asegurarse de que se cumplan ciertas características necesarias para que el vehículo encienda. Tales como:

- La marcha o motor de arranque active
- Existencia de chispa en las bujías
- Presión de combustible adecuada
- Motor a tiempo
- Compresión en la cámara de combustión de cada cilindro

Todo esto a través de un formulario creado con *checkboxes* tal como se aprecia en la figura 4.16.

Motor no arranca

*Condición inicial:
El vehículo no enciende y si
tiene batería*

- Da marcha o intento por arrancar*
 - Chispa en las bujías*
- presión correcta de combustible*
 - El tiempo es correcto*
- Tiene compresión suficiente*

BUSCAR

Figura 4.16. Formulario de problema de encendido con *checkboxes*.

Dichas características se encuentran ordenadas por el grado de complejidad, dejando hasta el final la compresión del motor, la cual, si estuviese fuera de rango, indicaría una avería interna donde el paso siguiente de diagnóstico involucra que el motor debe ser desarmado y sometido a pruebas de fugas, fisuras o hermeticidad de las válvulas y estado de los anillos de compresión entre otros elementos.

4.5.2.2 Problemas de desempeño

Para el segundo grupo de averías se contempla la condición de que el vehículo enciende, pero presenta comportamiento anormal de operación durante la marcha. Esto puede deberse a varios factores, ya que en este segundo grupo se encuentran varios sistemas involucrados, tales como:

- Sistema de alimentación de aire
- Sistema de inyección de combustible
- Sistema de encendido electrónico
- Sistema de distribución y tiempo

No obstante, antes de proceder a realizar alguna comprobación a cualquiera de los sistemas antes mencionados, se debe verificar la existencia de un testigo o alerta en el tablero o *cluster* de instrumentos para proceder a consultar el significado de la alerta. A través del formulario de la figura 4.17.

Testigos, alertas en el tablero o códigos OBDII

Condición inicial:

El vehículo presenta testigo en el tablero o código de avería

Los códigos de avería pueden ser consultados directamente en la base de datos. Para consultar dicho código da clic en el botón códigos OBDII

CÓDIGOS OBDII

Las alertas presentes en el tablero pueden deberse a algún sistema en operación o a averías presentes en la unidad. Para consultar el significado de una alerta en el tablero da clic al botón Alertas o testigos

ALERTAS O TESTIGOS

Figura 4.17. Menú de consulta de códigos OBDII y alertas o testigos.

En el apartado de la figura 4.18, se puede consultar el significado de los testigos de alerta en el tablero mediante un carrusel de imágenes, con el propósito de comprender su significado y la gravedad del problema. Esto permite determinar si es necesario detener la marcha o si solo se requiere una inspección del sistema. En caso de ser necesario, se debe utilizar el equipo de diagnóstico para obtener un código de avería de la ECU y consultarlo en la base de datos a través del Chatbot.

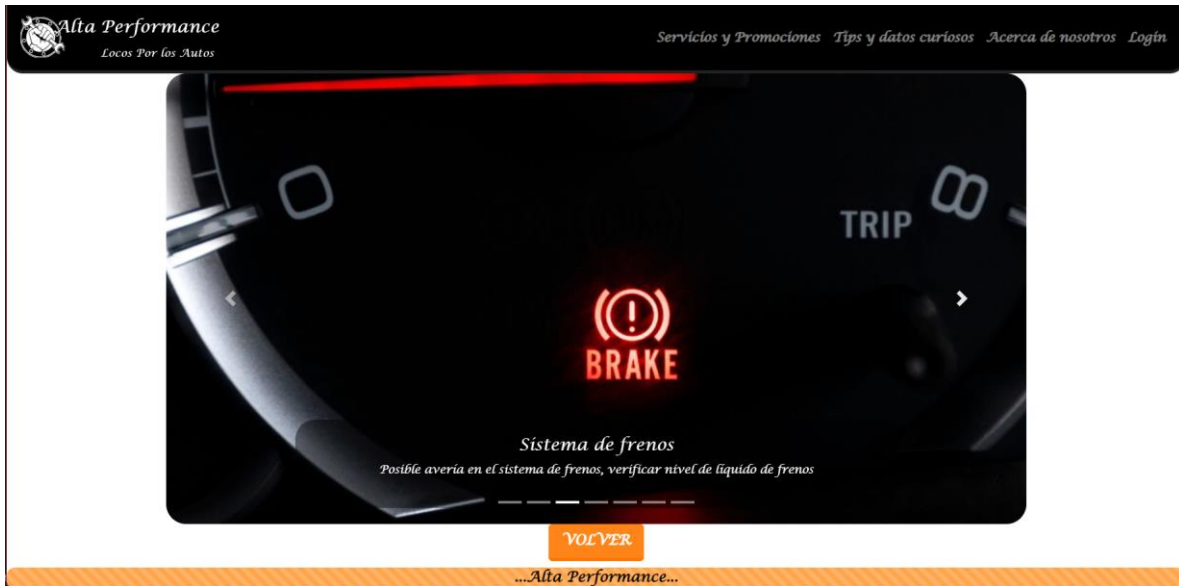


Figura 4.18. Carrusel de alertas.

Por otra parte, no siempre se enciende una luz de advertencia en el tablero o se encuentra algún código de avería como indicio almacenado en la computadora de la unidad.

En esos casos se procede a realizar una serie de comprobaciones de los elementos inmersos en los sistemas antes mencionados (sensores y actuadores). Para ello el técnico automotriz deberá asegurarse de que se cumplan ciertas características necesarias para que el vehículo trabaje de forma correcta. Tales como:

- Sistema de alimentación de aire sin obstrucciones o fisuras.
- Presión de combustible en rango
- Chispa presente en bobinas y bujías
- Pulso de inyección electrónica presente
- Funcionamiento en rango de inyectores
- Compresión en la cámara de combustión superior a 120 PSI
- Sistema de distribución a tiempo

Para ello se despliega el siguiente formulario con *checkboxes* (ver figura 4.19).

Problema de desempeño

*Condición inicial:
El vehículo enciende y
presenta comportamiento
errático durante la
conducción*

- Sistema de alimentación de aire sin
obstrucciones o fisuras*
- Presión de combustible*
- Chispa con la misma intensidad en
las bujías*
 - Pulso en los inyectores*
 - Desempeño adecuado de los
inyectores*
- Compresión mayor a 120 PSI*
- Motor sincronizado a tiempo*

BUSCAR

Figura 4.19. Formulario de problemas de desempeño con *checkboxes*.


4.5.2.3 Problemas de calentamiento

Para este sistema de refrigerante del automóvil se consideran 2 formularios. El primero es mediante una barra de rango donde se debe introducir la temperatura que arroja la aguja en el tablero de la unidad o, ya que en algunos vehículos modernos dicho indicador no se encuentra presente, por medio de la herramienta de diagnóstico se toma la lectura de datos en vivo para obtener el valor y enviarlo en el formulario de la figura 4.20.

Temperatura del refrigerante del motor

Condición inicial:
El vehículo presenta un valor de temperatura en el tablero fuera de lo común o sale vapor del cofre

La lectura en el tablero o con los datos en vivo en el scanner es:



BUSCAR

Figura 4.20. Formulario de temperatura del anticongelante.

Además, existen algunos elementos importantes que deben ser monitoreados para detectar problemas adicionales. Tales como:

- Nivel de anticongelante correcto
- Posibles fugas
- Apertura del *termoswitch* o termostato
- Ventiladores operando por intervalos

Esto se realiza mediante el formulario de la figura 4.21, en el cual el técnico especializado debe realizar las pruebas pertinentes a cada uno de los elementos involucrados para capturar la información y proceder al diagnóstico.

Sistema refrigerante

Condición inicial:

Presenta problemas de calentamiento o pérdida de líquido refrigerante

- Nivel de anticongelante correcto*
- No existe presencia de fuga*
- Termoswitch o termostato activa*
- Ventiladores activan y desactivan*

BUSCAR

Figura 4.21. Formulario de sistema de enfriamiento.

4.5.2.4 Sistema de emisiones contaminantes

El sistema que controla las emisiones contaminantes del vehículo es principalmente el convertidor catalítico y los sensores de oxígeno, forman parte del sistema de control de emisiones de los automóviles modernos y son esenciales para cumplir con las regulaciones ambientales [49].

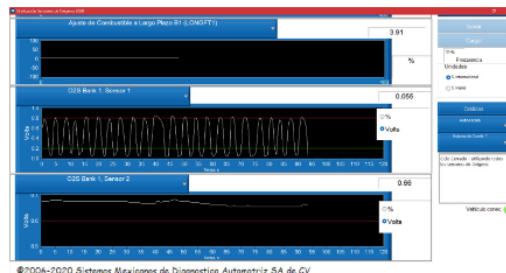
Los sensores de oxígeno, también conocidos como sondas lambda, son componentes clave en el sistema de control de emisiones de los vehículos modernos con motor de combustión interna. Estos sensores monitorean el contenido de oxígeno en los gases de escape del motor y proporcionan retroalimentación en tiempo real al sistema de gestión del motor para ajustar la mezcla aire-combustible.

Para poder generar un diagnóstico se toma la lectura del sensor 1 y 2 de cada banco de la unidad por medio de la herramienta de diagnóstico automotriz y se captura en el formulario, para determinar si se encuentra trabajando en el rango ideal tal como se aprecia en la figura 4.22.

Sistema de emisiones contaminantes

Condición inicial: Problemas para pasar la prueba de Verificación vehicular

Los sensores de oxígeno trabajan
ocilando el voltaje de señal entre
cero y uno tal como se aprecia en la
iguiente imagen



- Sensor O₂ 1 Banco 1 oscilando entre cero y uno
- Sensor O₂ 1 Banco 1 oscilando entre cero y uno
- Sensor O₂ 1 Banco 1 oscilando entre cero y uno
- Sensor O₂ 1 Banco 1 oscilando entre cero y uno

BUSCAR

Figura 4.22. Formulario de sistema de emisiones contaminantes.

Dichos sensores producen una señal eléctrica que oscila entre cero y uno voltio. Esta oscilación se debe a las variaciones en la concentración de oxígeno en los gases de escape del motor.

Cuando la relación aire-combustible es rica (exceso de combustible en comparación con el oxígeno), la señal del sensor de oxígeno tiende a ser baja, cerca de cero voltios. Esto indica que hay poco oxígeno presente en los gases de escape.

Por otro lado, cuando la relación aire-combustible es pobre (exceso de oxígeno en comparación con el combustible), la señal del sensor de oxígeno tiende a ser alta, cerca de un voltio. Esto indica que hay mucho oxígeno presente en los gases de escape.

Cuando la relación aire-combustible es cercana a la relación estequiométrica (la cantidad teórica de aire requerida para quemar completamente el combustible), la señal del sensor de oxígeno oscila entre cero y uno voltio. Esta oscilación indica que el sistema está alternando entre una mezcla rica y una mezcla pobre para mantener la relación aire-combustible lo más cercana posible a la relación estequiométrica.

Capítulo 5. Resultados

Los resultados obtenidos de este trabajo de tesis de maestría son el modelo de aprendizaje automático basado en árboles de decisión CART, así como la clasificación de temperatura por medio de lógica difusa.

5.1 Correlación de Pearson entre variables

La matriz de correlación de Pearson muestra las relaciones entre las variables codificadas de tu conjunto de datos (Ver figura 5.1).

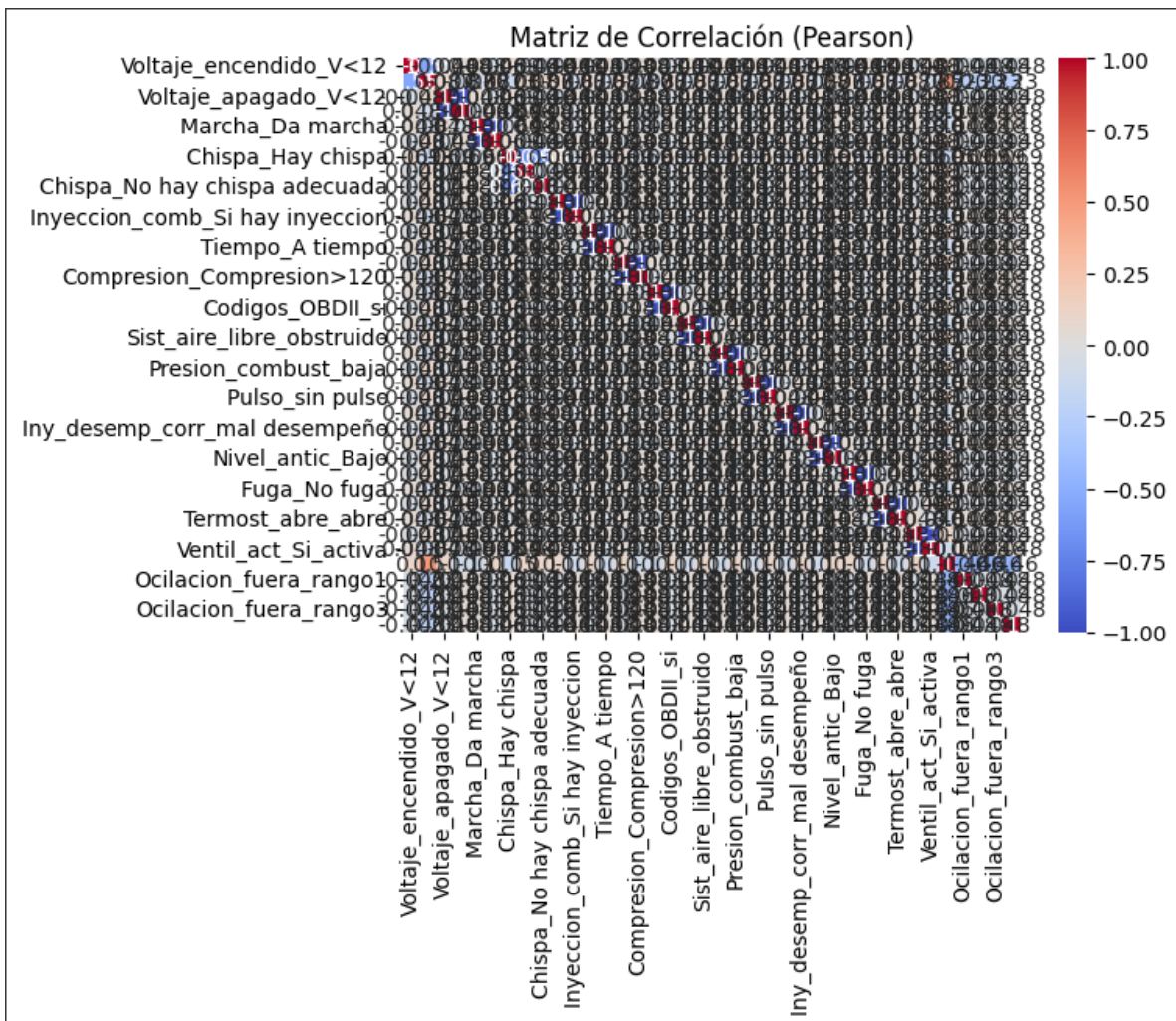


Figura 5.1. Matriz de correlación de Pearson.

Algunos puntos clave que se pueden interpretar a partir de esta matriz:

- **Correlaciones Perfectas o Altas:** Los valores en la diagonal de la matriz son 1, lo que es esperado porque cada variable tiene una correlación perfecta consigo misma. Como Oscilación (Oscilación)_fuera_rango, donde se aprecian diversas variables que indican si la oscilación está fuera de un rango específico. Estas variables están altamente correlacionadas entre sí (correlación cercana a 1), lo cual sugiere que pueden estar midiendo aspectos similares o estar relacionadas a un mismo fenómeno.
- **Correlaciones Inversas:** Las correlaciones entre pares de variables como Voltaje_encendido_V<12 y Voltaje_encendido_V>12, que tienen una correlación negativa fuerte (-0.549170). Esto sugiere que cuando una está activada (o tiene un valor alto), la otra tiende a no estarlo, lo cual es lógico dado que estas variables representan condiciones opuestas. La variable Voltaje_apagado_V<12 tiene una correlación inversa perfecta (-1) con Voltaje_apagado_V>12, lo que es natural ya que una condición excluye a la otra. Las variables "Ventil_act_Si_activa" y "Ventil_act_No_activa" tienen una correlación perfectamente inversa también (-1 y 1), lo cual es esperado porque una variable es simplemente la negación de la otra.

5.2 Correlación de Spearman entre variables

La matriz de correlación de Spearman muestra en la figura 5.2, la relación entre diferentes variables codificadas en el conjunto de datos. Cada valor en la matriz representa el coeficiente de correlación de Spearman entre dos variables, que varía de -1 (correlación perfectamente negativa) a 1 (correlación perfectamente positiva), pasando por 0 (sin correlación).

Algunos puntos clave que se pueden interpretar a partir de esta matriz:

- **Valores Altos y Bajos:** Los coeficientes que se acercan a 1 o -1 indican una fuerte relación monótonica entre dos variables. Por ejemplo: Voltaje_encendido_V<12 y Voltaje_encendido_V>12 tienen una correlación de -0.549, indicando una relación negativa moderada entre estas dos variables.
- **Simetría:** La matriz es simétrica con respecto a la diagonal principal (donde la correlación de una variable consigo misma es 1), lo que es una característica típica de las matrices de correlación.

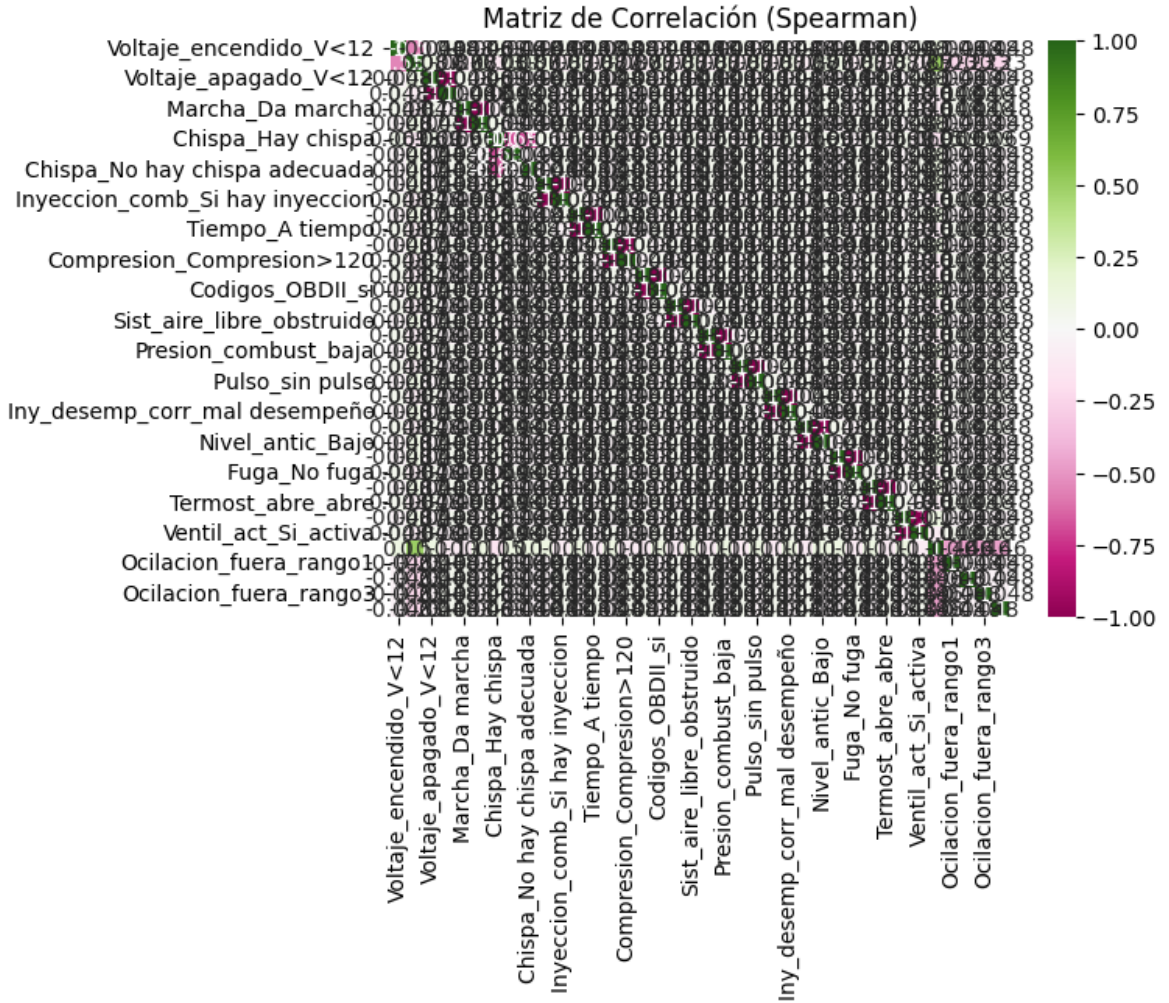


Figura 5.2. Matriz de correlación de Pearson.

5.3 Resultados obtenidos después de realizar la poda

Para el modelo de aprendizaje automático abordado en este trabajo de tesis de maestría, el árbol de decisión CART, se realizaron 4 métodos de poda. Estos métodos de poda incluyeron el uso de *CCP (Cost Complexity Pruning)*, *Max Depth (Profundidad Máxima)*, *Min Samples Leaf (Mínimo de Muestras por Hoja)* y *Min Samples Split (Mínimo de Muestras por División)*, cada uno con sus propias configuraciones y resultados.

Los resultados obtenidos muestran que el método de poda utilizando CCP logró una precisión y una exactitud del 100%, una tasa de error y *Novelty rejection percentage* del 0%. En la figura 5.3 se aprecia la matriz de confusión.

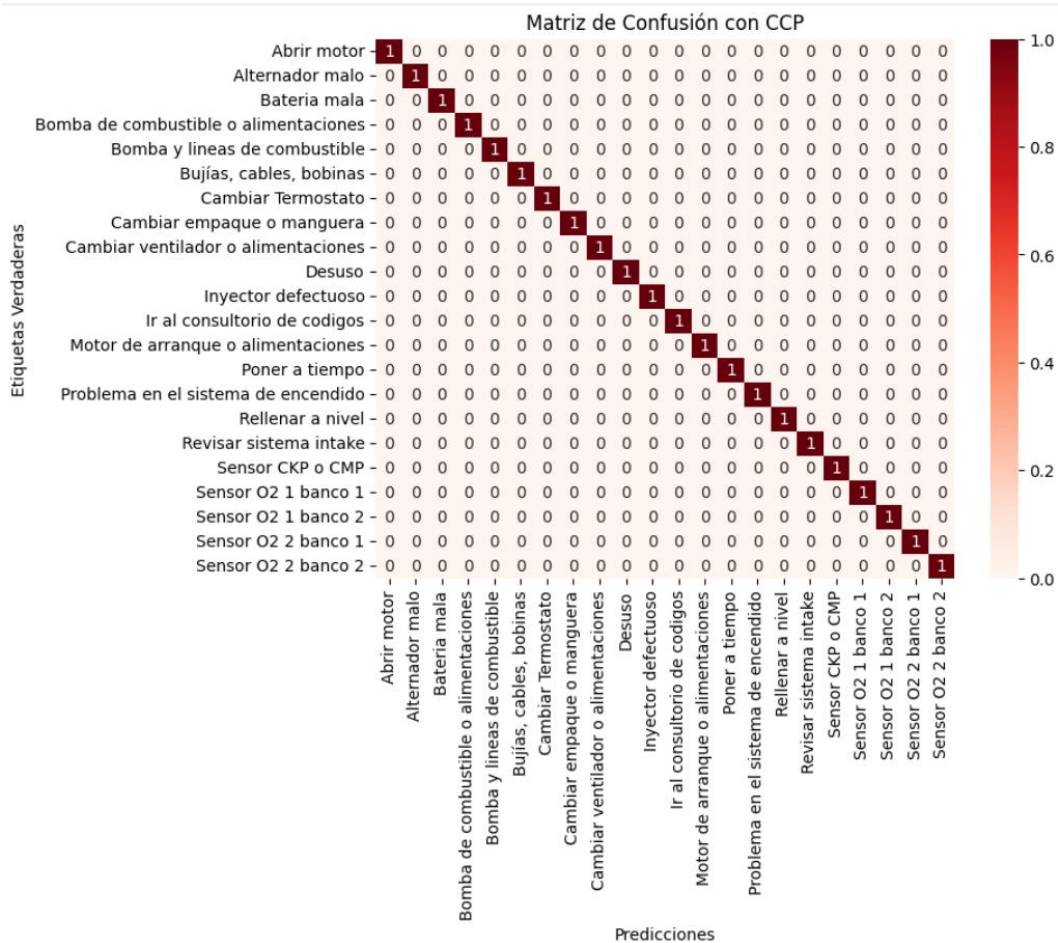


Figura 5.3. Matriz de confusión con CCP

Esto se evidencia en la matriz de confusión, donde se observa que todos los valores pronosticados coinciden con los valores reales. Esto indica una excelente capacidad del modelo para predecir con precisión las clases de salida. En la figura 5.4 se muestra la precisión, exactitud, tasa de error y *Novelty rejection percentage*.

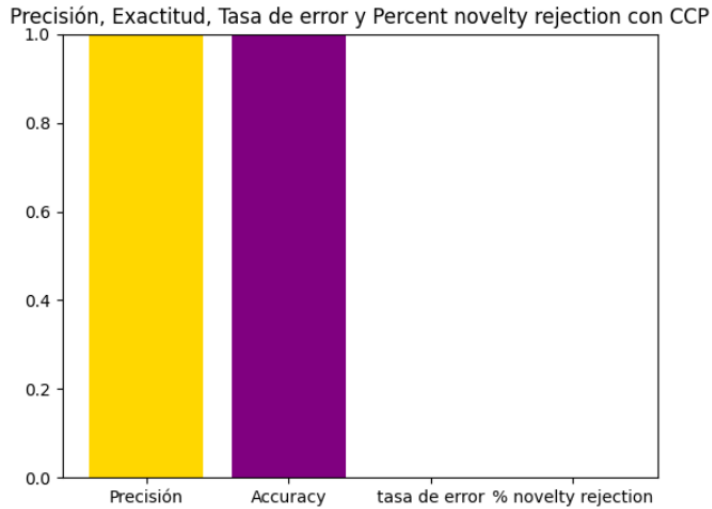


Figura 5.4. Precisión, exactitud, tasa de error y *Novelty rejection percentage* de CCP

En contraste, los métodos de poda *Max Depth*, *Min Samples Leaf* y *Min Samples Split* mostraron resultados variados en términos de precisión y exactitud. Por ejemplo, el método de poda *Max Depth* alcanzó una precisión del 41.25%, una exactitud del 45.45%, una tasa de error del 54.54% y un *Novelty rejection percentage* del 0%. Tal como se aprecia en la figura 5.5.

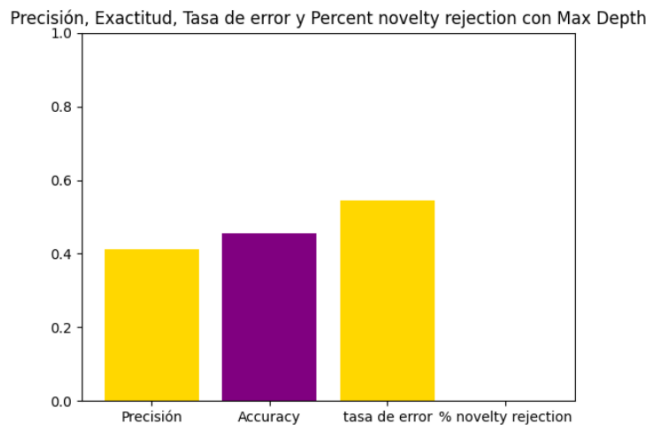


Figura 5.5. Precisión, exactitud, tasa de error y *Novelty rejection percentage* de Max Depth

Contando en la matriz de confusión con 10 verdaderos positivos y 12 verdaderos negativos. Esto sugiere que la limitación en la profundidad del árbol de decisión puede haber afectado negativamente la capacidad del modelo para generalizar los datos de prueba. Restando confiabilidad a la predicción de la avería detectada en la unidad. La matriz de confusión de este método de poda se aprecia en la figura 5.6.

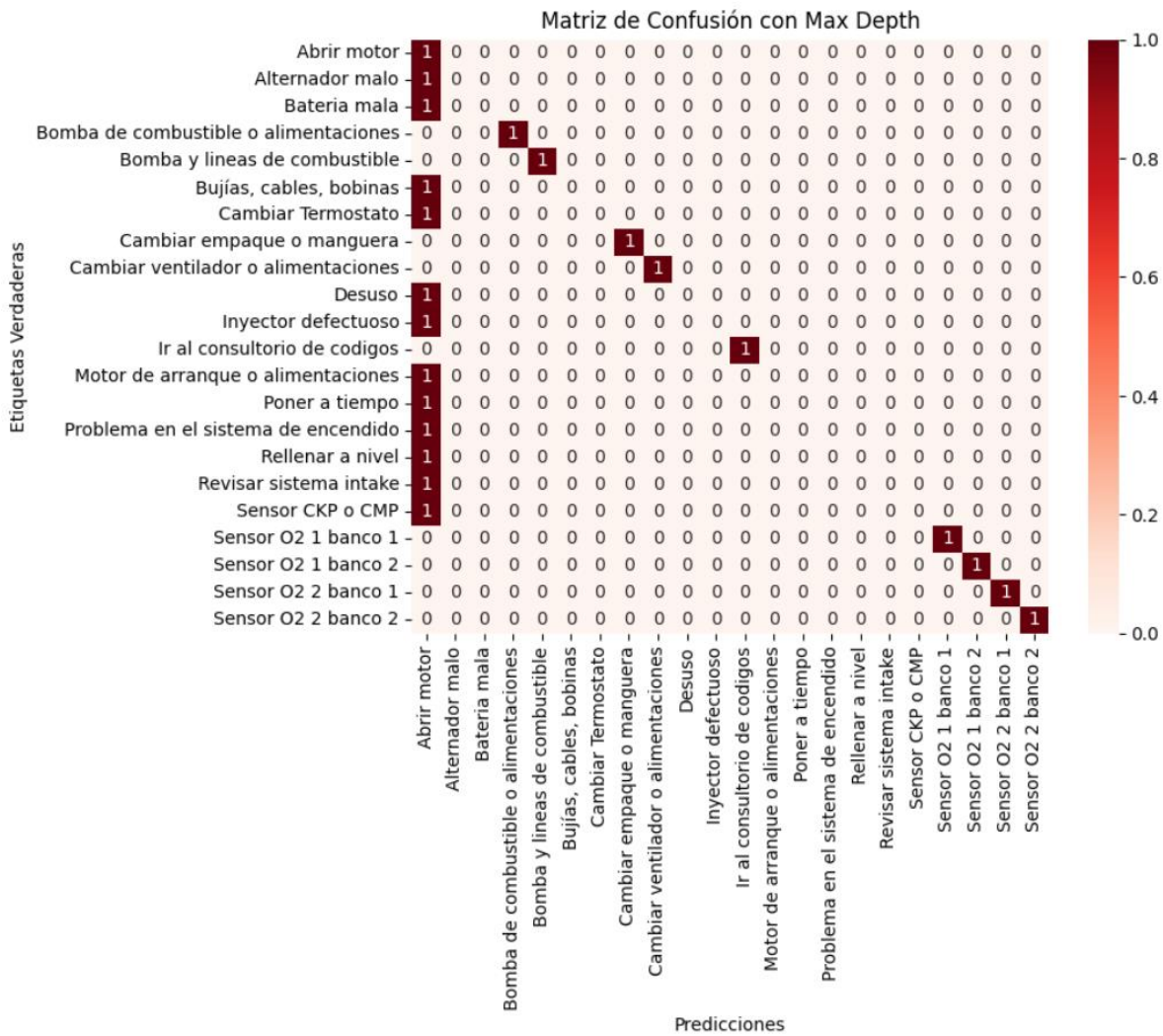


Figura 5.6. Matriz de confusión con Max Depth

Similarmente, el método de poda *Min Samples Leaf* mostró resultados menos favorables en términos de precisión y exactitud, con una precisión del 0.01%, una exactitud del 0.09%, una tasa de error del 90% y un *Novelty rejection percentage* del 0%. Estos resultados indican que ajustar el número mínimo de muestras por hoja puede no haber sido efectivo para mejorar el rendimiento del modelo de árbol.

Los resultados observados de este método de poda en el modelo de árbol de decisión CART destacan la importancia de seleccionar cuidadosamente el número mínimo de muestras por hoja para optimizar el rendimiento del modelo y garantizar una predicción precisa de las clases de salida para generar un buen diagnóstico de averías del bloque motor.

La matriz de confusión de *Min Samples Leaf* se visualiza en la figura 5.7 y la gráfica de la precisión y exactitud se muestra en la figura 5.8.

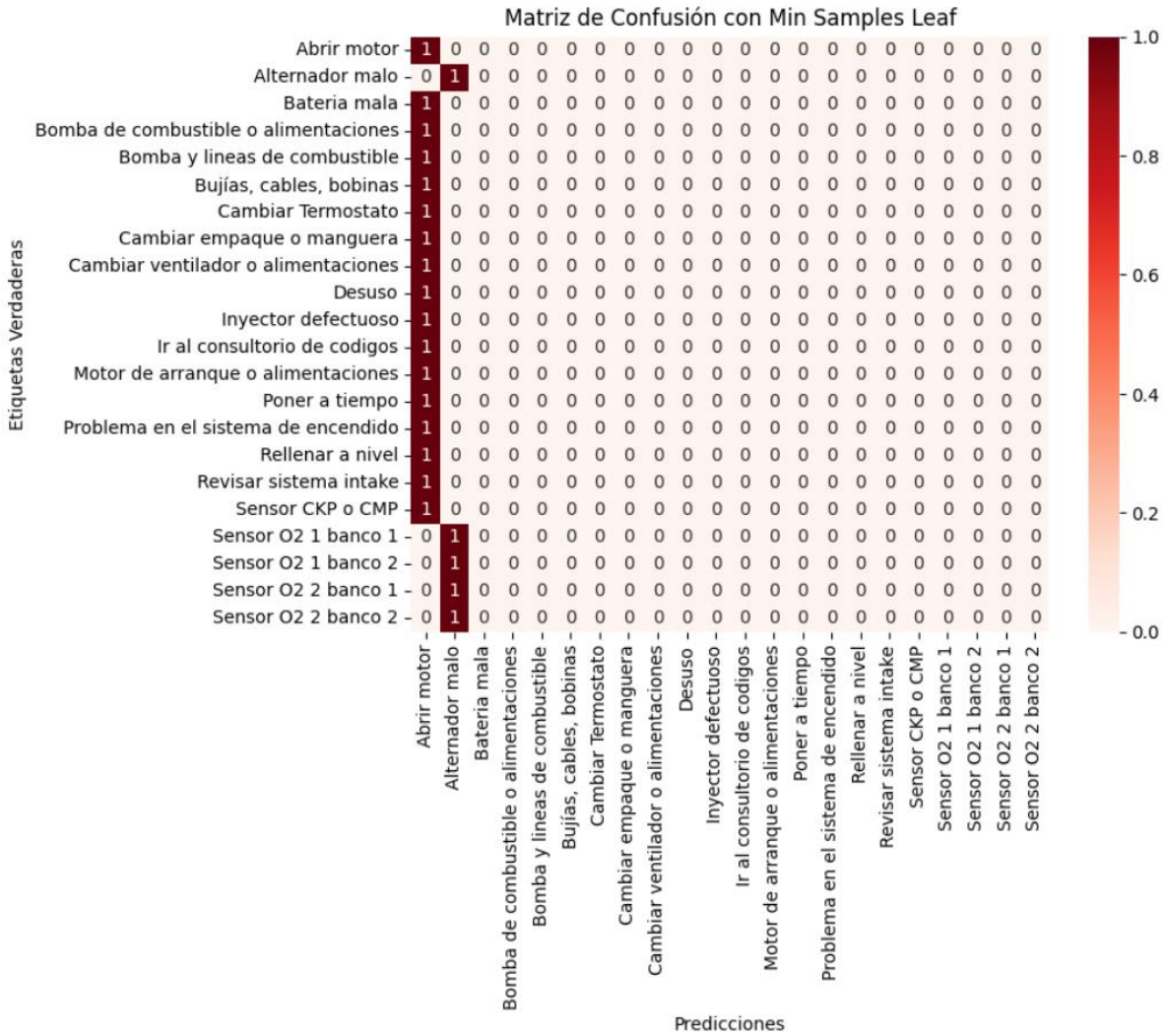


Figura 5.7. Matriz de confusión con *Min Samples Leaf*

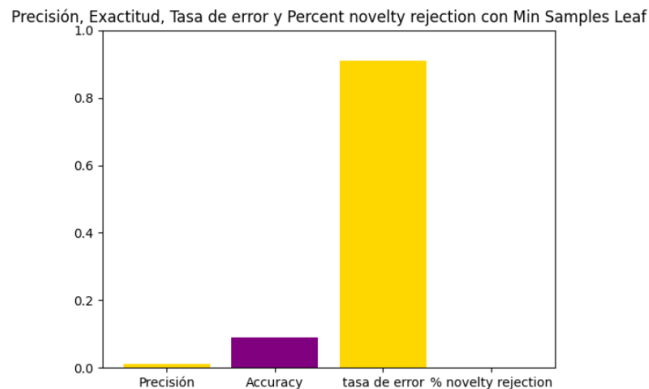


Figura 5.8. Precisión, exactitud, tasa de error y *Novelty rejection percentage* de *Min Samples Leaf*

Los resultados obtenidos con el método de poda utilizando *Min Samples Leaf* muestran una precisión y exactitud muy bajas. Estos valores indican que este método de

poda no fue efectivo para mejorar el rendimiento del modelo de árbol de decisión CART en este contexto específico.

La matriz de confusión revela que todas las predicciones realizadas por el modelo utilizando este método de poda fueron consistentemente incorrectas, con una clasificación errónea de todas las instancias de prueba.

Del mismo modo el método de poda *Min Samples Split* mostró resultados un poco más favorables, Estos resultados indican que ajustar el número mínimo de muestras por división puede no haber sido efectivo para mejorar el rendimiento del modelo. La matriz de confusión se visualiza en la figura 5.9.

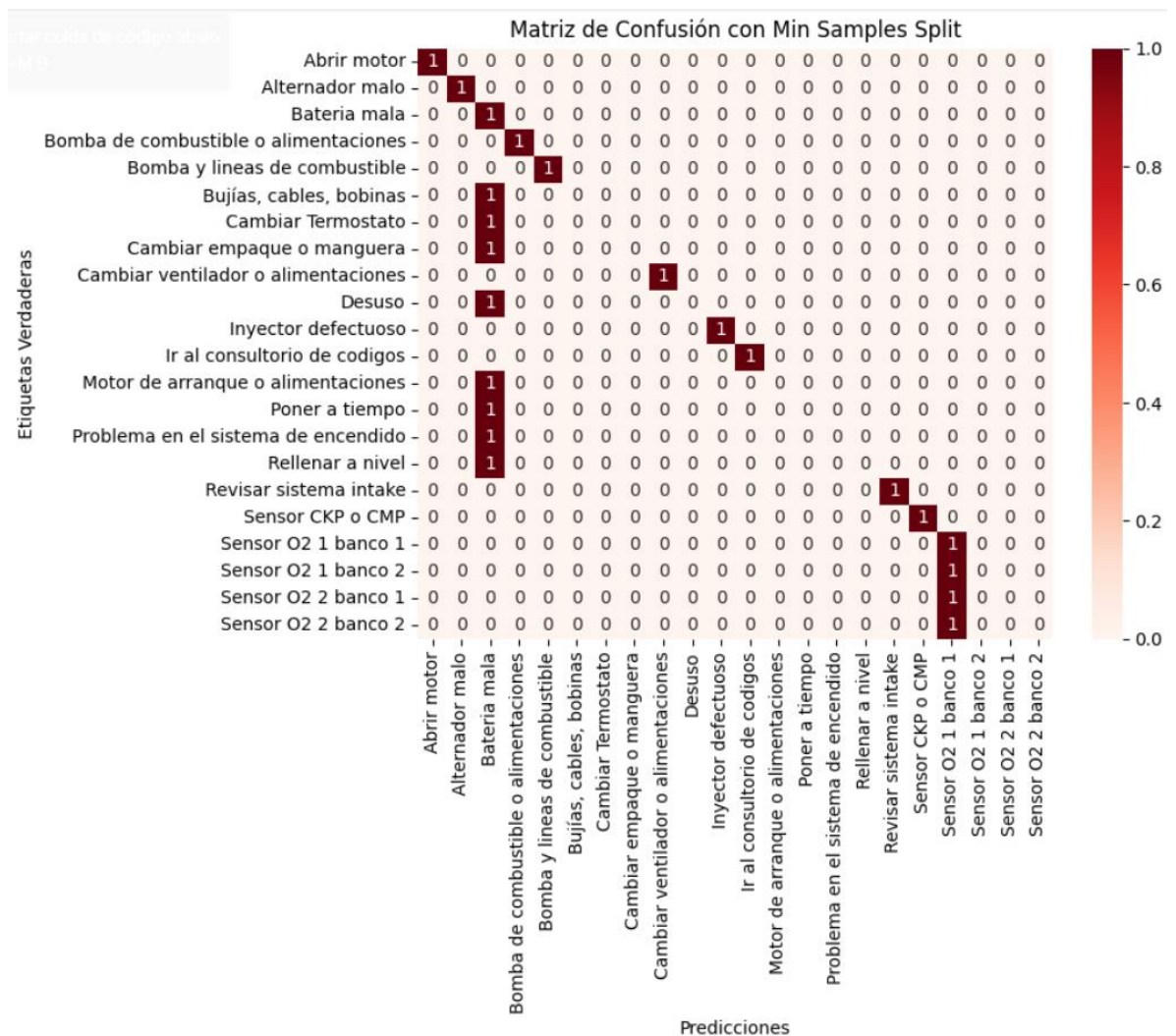


Figura 5.9. Matriz de confusión con *Min Samples Split*

La matriz de confusión revela que el modelo logró clasificar correctamente algunas instancias, pero también cometió errores significativos en otras. Esto sugiere que el método de poda *Min Samples Split* permitió una mejor generalización de los datos en comparación

con *Min Samples Leaf* y *Max Depth*, pero aun así no fue suficiente para obtener un rendimiento robusto en todas las clases.

En términos de precisión y exactitud, con una precisión del 42.55%, una exactitud del 50%, una tasa de error del 50% y un *Novelty rejection percentage* del 0%. posiciona a este método de poda en segundo lugar después de CCP. Sin embargo, estos valores muestran que el rendimiento del modelo no es óptimo. Tal como se aprecia en la figura 5.10.

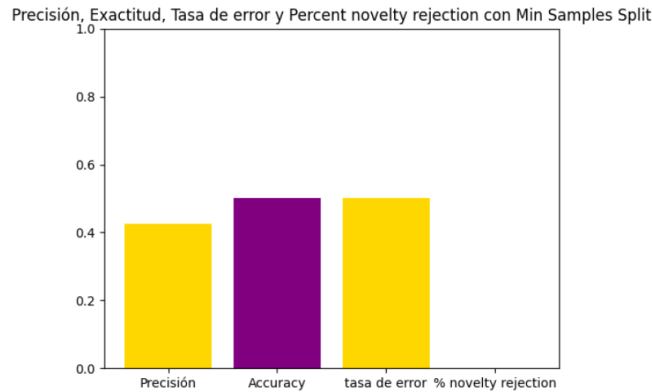


Figura 5.10. Precisión, exactitud, tasa de error y *Novelty rejection percentage* de *Min Samples Split*

El promedio de validación cruzada para cada método de poda muestra una puntuación del 100% para CCP, 36% para *Max Depth*, 09% para *Min Samples Leaf* y 54% para *Min Samples Split* (Ver figura 5.11).

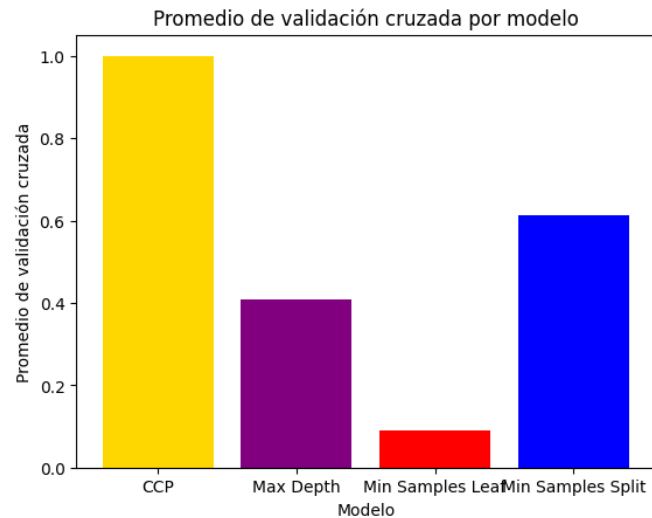


Figura 5.11. Promedio de validación cruzada por modelo

En resumen, los resultados de los diferentes métodos de poda en el modelo de árbol de decisión CART destacan la importancia de seleccionar cuidadosamente las técnicas de poda adecuada para optimizar el rendimiento del modelo y garantizar una predicción precisa de las clases de salida.

Finalmente, el mejor método de poda para este trabajo de tesis de maestría es CCP y es con el que se realiza la predicción para generar un diagnóstico y detectar al responsable de la avería mecánica de la unidad, ya que reduce el tiempo de respuesta al generar un árbol de decisión sin sobre ajustarse, en los otros algoritmos al reducir la profundidad, número de muestras o la división de las ramas baja considerablemente la precisión y la exactitud a menos del 50%. Y si bien la solución de bajar el motor e inspeccionar las piezas mecánicas podría solucionar el problema, esa es la solución más costosa y no precisamente lo que la unidad necesita.

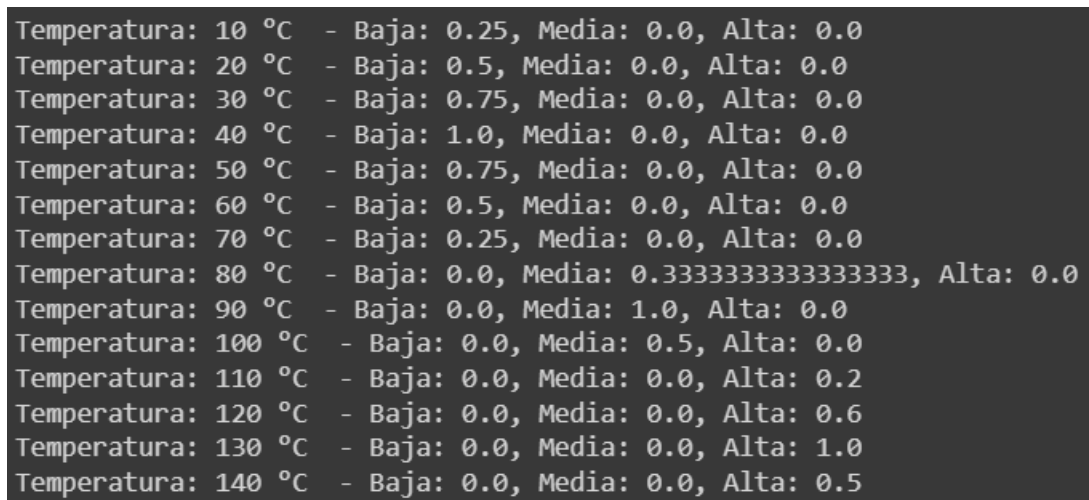
Esta precisión se logra porque las reglas o el conocimiento experto son específicos y se centran en un área particular del problema. Dado que las reglas están diseñadas específicamente para el bloque del motor basándose en los procedimientos de diagnóstico establecidos en los manuales de mantenimiento y la información almacenada en la base de datos de códigos de avería

Los árboles de decisión CART son una excelente elección cuando estás trabajando con datos que no muestran una correlación lineal clara, ya que son capaces de manejar relaciones no lineales y capturar interacciones entre las variables.

No requieren que los datos estén correlacionados linealmente. Pueden capturar relaciones complejas que otros modelos lineales podrían pasar por alto.

5.4 Observaciones de la clasificación de temperatura

Para la clasificación de temperatura con lógica difusa se muestran los resultados de los valores de membresía para un conjunto de valores, tal como se aprecia en la figura 5.12.



| | | | | |
|---------------------|---|-------------|----------------------------|-----------|
| Temperatura: 10 °C | - | Baja: 0.25, | Media: 0.0, | Alta: 0.0 |
| Temperatura: 20 °C | - | Baja: 0.5, | Media: 0.0, | Alta: 0.0 |
| Temperatura: 30 °C | - | Baja: 0.75, | Media: 0.0, | Alta: 0.0 |
| Temperatura: 40 °C | - | Baja: 1.0, | Media: 0.0, | Alta: 0.0 |
| Temperatura: 50 °C | - | Baja: 0.75, | Media: 0.0, | Alta: 0.0 |
| Temperatura: 60 °C | - | Baja: 0.5, | Media: 0.0, | Alta: 0.0 |
| Temperatura: 70 °C | - | Baja: 0.25, | Media: 0.0, | Alta: 0.0 |
| Temperatura: 80 °C | - | Baja: 0.0, | Media: 0.3333333333333333, | Alta: 0.0 |
| Temperatura: 90 °C | - | Baja: 0.0, | Media: 1.0, | Alta: 0.0 |
| Temperatura: 100 °C | - | Baja: 0.0, | Media: 0.5, | Alta: 0.0 |
| Temperatura: 110 °C | - | Baja: 0.0, | Media: 0.0, | Alta: 0.2 |
| Temperatura: 120 °C | - | Baja: 0.0, | Media: 0.0, | Alta: 0.6 |
| Temperatura: 130 °C | - | Baja: 0.0, | Media: 0.0, | Alta: 1.0 |
| Temperatura: 140 °C | - | Baja: 0.0, | Media: 0.0, | Alta: 0.5 |

Figura 5.12. Valores de membresía.

Cada línea representa un valor de temperatura seguido de sus valores de pertenencia a las categorías baja, media y alta, respectivamente. Estos valores representan

el grado de pertenencia de cada temperatura a cada categoría en función de las funciones de pertenencia difusas definidas.

5.5 Generación del diagnóstico

Una vez que el usuario captura las condiciones observadas en el manejo en del automóvil a través de la aplicación web en los formularios, se envía para generar la predicción y devolver el diagnóstico de la unidad. Tal como se aprecia en la figura 5.13.



Figura 5.13. Diagnóstico generado.

Capítulo 6. Conclusiones y trabajo futuro

En este trabajo de tesis de maestría se presentó un enfoque integral para el diagnóstico de fallas del motor donde se combinaron técnicas de aprendizaje automático, bases de conocimiento, árboles de decisión y lógica difusa. Este enfoque ha demostrado mejorar significativamente la precisión y eficiencia en la detección de comportamientos irregulares en motores de automóviles.

Durante la investigación, se desarrolló una base de datos para códigos de falla OBD-II y una base de conocimiento para el diagnóstico de averías, junto con la implementación de árboles de decisión utilizando técnicas de poda. Esta combinación resultó en una alta precisión del 100% con el método de poda CPP en la identificación de fallas, lo que subraya la efectividad de nuestra metodología.

Además, al aplicar lógica difusa para la clasificación de temperaturas, se mejoró la capacidad del sistema para evaluar con precisión la temperatura del motor, lo que contribuye a una detección más precisa de problemas relacionados con el sobrecalentamiento.

Asimismo, se llevó a cabo el desarrollo de un Chatbot como sistema inteligente ya que genera una respuesta en tiempo real para la consulta de averías mecánicas. El asistente inteligente fue implementado dentro de la aplicación web Coq's Tuning Performance; la cual proporciona un entorno sencillo, usable, robusto, seguro y responsivo, permitiendo actualizaciones constantes sin interrumpir la experiencia del usuario. La aplicación de tecnologías como AJAX ha mejorado la capacidad de respuesta y la experiencia del usuario según los requisitos del proyecto.

Durante el desarrollo de este trabajo de tesis, se establecieron varios objetivos específicos que fueron alcanzados de acuerdo a la metodología, por lo tanto, se consiguió el objetivo general. El éxito en el logro de estos objetivos no solo valida la viabilidad de la propuesta, sino que también demuestra el potencial de estas tecnologías para mejorar significativamente los procesos de diagnóstico en talleres mecánicos, ofreciendo un servicio más rápido y preciso a los propietarios de vehículos. Este sistema representa un avance importante en la industria automotriz, proporcionando herramientas innovadoras para el diagnóstico y mantenimiento de vehículos.

En el futuro, se planea expandir la aplicación de la lógica difusa para evaluar otros parámetros de funcionamiento del vehículo. Además, se explorará la integración de sistemas de emisiones contaminantes del automóvil utilizando las condiciones, reglas y hechos observados del conjunto de sensores y actuadores del bloque del motor, con el objetivo de reducir el impacto ambiental generado por los vehículos.

Bibliografía

- [1] White, C. (1998). *Automotive Diagnostic Fault Codes Techbook*. Reino Unido: Haynes.
- [2] México Digital Comunicación S.A. de C.V. (Ed.). (2014). *Manual de computadoras y módulos automotrices: Ford y Volkswagen*.
- [3] Pérez Darquea, D. G. (2018). Evolución de los dispositivos electrónicos en un automóvil. *INNOVA Research Journal*, 3(2), 1–7. <https://doi.org/10.33890/innova.v3.n2.2018.616>
- [4] Pérez Vázquez, Antonio. (2021). *Aplicación Coq's Tuning Performance, para mejorar la Atención al Cliente mediante el Procesado de Solicitudes en Segundo Plano*. [Tesis de licenciatura]. Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Computación.
- [5] W3School. (s. f.). *AJAX Introduction*. w3schools.com. Recuperado 19 de septiembre de 2020, de https://www.w3schools.com/xml/ajax_intro.asp
- [6] Fermé, E. (2007). *Revisión de Creencias*. Recuperado el 10 de septiembre de 2023, de <https://www.redalyc.org/pdf/925/92503403.pdf>
- [7] Gupta, A. (2020). *Introduction to AI Chatbots*. *International journal of engineering research and technology*, V9(07). <https://doi.org/10.17577/ijertv9is070143>
- [8] Rouhiainen, L. P. R. (2018). *Inteligencia artificial: 101 cosas que debes saber hoy sobre nuestro futuro*. Editorial Planeta.
- [9] Boden, M. A. (2017). *Inteligencia artificial*. Turner.
- [10] Sandoval Serrano, L. J. (2018). *Algoritmos de aprendizaje automático para análisis y predicción de datos*. *Revista Tecnológica*; no. 11.
- [11] Sempere, J. (2014). *Aprendizaje de árboles de decisión*. *Universidad Politécnica de Valencia, Valencia*.
- [12] García, A., Martínez, G., Nuñez, E., & Guzmán, A. (1998). *Clasificación supervisada, inducción de arboles de decisión, algoritmo kd*. *Proc. Simp. Int. de Comp. CIC*, 98, 602-614.
- [13] Friedman, J. H. (2001). *Greedy function approximation: a gradient boosting machine*. *Annals of statistics*, 1189-1232.
- [14] Ortiz, G. B. *CLASIFICACION MULTICLASE DE DATOS DE CALIDAD DE SOFTWARE DESARROLLADO EN IBM RPG USANDO ARBOLES DE DECISION*.
- [15] Quinlan, J. R. *Induction of Decision Trees*. *Aprendizaje automático*. P 81-106, 1986
- [15] Quinlan, J. R. *C4.5: Programs for aprendizaje automático*. Morgan Kaufmann Publishers, 1993.
- [16] Suguiura, F. O. R. (2022). *Árbol de Decisión en Aprendizaje Automático*. *REVISTA VARIANZA*, 39-46.
- [17] Carbonell, D. G. E. *Árboles de Regresión. Algunos algoritmos y extensiones a métodos de consenso*.
- [18] McTavish, H., Zhong, C., Achermann, R., Karimalis, I., Chen, J., Rudin, C., & Seltzer, M. (2022, June). *Fast sparse decision tree optimization via reference ensembles*. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 36, No. 9, pp. 9604-9613).
- [19] Zhong, M., Georgiopoulos, M., & Anagnostopoulos, G. C. (2008). *A k-norm pruning algorithm for decision tree classifiers based on error rate estimation*. *Aprendizaje automático*, 71, 55-88.
- [20] Tong, L., Liu, Z., Jiang, Z., Zhou, F., Chen, L., Lyu, J., ... & Zhou, H. (2022). *Cost-sensitive boosting pruning trees for depression detection on Twitter*.

- [21] Borja-Robalino, R., Monleon-Getino, A., & Rodellar, J. (2020). Estandarización de métricas de rendimiento para clasificadores Machine y Deep Learning. *Revista Ibérica de Sistemas e Tecnologías de Informação*, (E30), 184-196.
- [22] Hendrickx, K., Perini, L., Van der Plas, D., Meert, W., & Davis, J. (2021). Aprendizaje automático with a reject option: A survey. *arXiv preprint arXiv:2107.11277*.
- [23] Berrar, D. (2019). Cross-validation.
- [24] Roy-García, I., Rivas-Ruiz, R., Pérez-Rodríguez, M., & Palacios-Cruz, L. (2019). Correlación: no toda correlación implica causalidad. *Revista Alergia México*, 66(3), 354-360.
- [25] Márquez, J. S. (2004). Introducción a los sistemas expertos. *Consultado en Revista Bibliotecas [En línea] Disponible en*, 6(03), 2004.
- [26] Badaró, S., Ibañez, L. J., & Agüero, M. J. (2013). Sistemas expertos: fundamentos, metodologías y aplicaciones. *Ciencia y tecnología*, (13), 349-364.
- [27] Mejia, J. C. G., Quintero, D. M. M., & Charry, W. P. Modelo conceptual para la Inferencia de un sistema basado en conocimiento de Auditorías de sistemas de gestión de la calidad en el resultado de hallazgos.
- [28] Marqués-Andrés, M. (2011). *Bases de datos*. Universitat Jaume I.
- [29] Guarino, N., & Giaretta, P. (1995). Ontologies and knowledge bases. Towards very large knowledge bases, 1-2.
- [30] Wu, S., Hsiao, L., Cheng, X., Hancock, B., Rekatsinas, T., Levis, P., & Ré, C. (2018, May). Fonduer: Knowledge base construction from richly formatted data. In *Proceedings of the 2018 international conference on management of data* (pp. 1301-1316).
- [31] Capterra.com. Recuperado el 19 de enero de 2024, de https://www.capterra.com/sem-compare/knowledge-management-software/?utm_source=ps-google&utm_medium=ppc&utm_campaign=:1:CAP:2:COM:3:All:4:INTL:5:BAU:6:SOF:7:Desktop:8:BR:9:Knowledge_Management&network=g&gclid=Cj0KCQiAtaOtBhCwARIsAN_x-3KdRK57urr3eMsF2Z1K904IJ3KGrUM9ucVMRUV5bXdtq8NS_gKn3WUaApnJEALw_wcB
- [32] Trillas, E., & Eciolaza, L. (2015). Fuzzy logic. *Springer International Publishing. DOI*, 10, 978-3.
- [33] González, C. (2015). Lógica difusa una introducción práctica, técnicas de Soft Computing. *Recuperado de: https://www.esi.uclm.es/www/cglez/downloads/docencia/2011_Softcomputing/LogicaDifusa.pdf*.
- [34] Zadeh, L. A. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on systems, Man, and Cybernetics*, (1), 28-44.
- [35] González Duque, R. (2011). Python para todos.
- [36] The Python Standard Library. (s/f). Python documentation. Recuperado el 13 de enero de 2024, de <https://docs.python.org/3/library/index.html>
- [37] Adamopoulou, E. y Moussiades, L. (2020). Una descripción general de la tecnología de Chatbot. En conferencia internacional IFIP sobre aplicaciones e innovaciones de inteligencia artificial (págs. 373-383). Springer, Cham.
- [38] Nimavat, K., & Champaneria, T. (2017). Chatbots: An overview types, architecture, tools and future possibilities. *Int. J. Sci. Res. Dev*, 5(7), 1019-1024.
- [39] Kucherbaev, P., Bozzon, A., & Houben, G. J. (2018). Human-aided bots. *IEEE Internet Computing*, 22(6), 36-43.
- [40] Hien, H. T., Cuong, P. N., Nam, L. N. H., Nhung, H. L. T. K., & Thang, L. D. (2018, December). Intelligent assistants in higher-education environments: the FIT-EBot, a chatbot for administrative and learning support. In *Proceedings of the 9th International Symposium on Information and Communication Technology* (pp. 69-76).

- [41] Ramesh, K., Ravishankaran, S., Joshi, A., & Chandrasekaran, K. (2017, May). A survey of design techniques for conversational agents. In *International conference on information, communication and computing technology* (pp. 336-350). Singapore: Springer Singapore.
- [42] Wu, Y., Wu, W., Xing, C., Zhou, M., & Li, Z. (2016). Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots. En *arXiv [cs.CL]*. <http://arxiv.org/abs/1612.01627>
- [43] 10 herramientas de Chatbot para WhatsApp en el 2023. (2023, junio 29). Zerviz. <https://www.zervizgroup.com/herramientas-de-chatbot-para-whatsapp/>
- [44] Condon, S. (2020, marzo 4). Zendesk launches Sales Suite, bundling Zendesk Sell, Chat and other tools. ZDNET. <https://www.zdnet.com/article/zendesk-launches-sales-suite-bundling-zendesk-sell-chat-and-other-tools/>
- [45] Joshua, S. R., & Mogeia, T. (2019). Work System Framework: Analisis Inmagic Presto dan Zendesk. *CogITo Smart Journal*, 5(2), 266-279.
- [46] Tars - Shaping the future of customer experience and employee productivity with Conversational AI. (2021, febrero 24). Tars. <https://hellotars.com/>
- [47] Premier AI online chatbot software. (2020, abril 7). Flow XO. <https://flowxo.com/>
- [48] Gadyatskyi, IA (2022). Desarrollo del asistente digital virtual "Solicitante" basado en el servicio Flow XO.
- [49] Janarthanam, S. (2017). Hands-on chatbots and conversational UI development: build chatbots and voice user interfaces with Chatfuel, Dialogflow, Microsoft Bot Framework, Twilio, and Alexa Skills. Packt Publishing Ltd.
- [50] About us. (s/f). Chatfuel. Recuperado el 26 de enero de 2024, de <https://chatfuel.com/about-us>
- [51] Manual de tecnología. Documento escolar del Instituto Superior de Ingeniería y Mecánica Automotriz, 2018
- [52] Maida, E. G., & Pacienza, J. (2015). Metodologías de desarrollo de software.
- [53] Rivas, C. I., Corona, V. P., Gutiérrez, J. F., & Hernández, L. (2015). Metodologías actuales de desarrollo de software. *Revista de Tecnología e Innovación*, 2(5), 980-986.
- [54] De, S., & Chakraborty, B. (2018, April). Case based reasoning (cbr) methodology for car fault diagnosis system (cfd) using decision tree and jaccard similarity method. In 2018 3rd International Conference for Convergence in Technology (I2CT) (pp. 1-6). IEEE.
- [55] Xiaolei, L., & Xiaobing, W. (1999, September). The application of rough set theory in vehicle transmission system fault diagnosis. In *Proceedings of the IEEE International Vehicle Electronics Conference (IVEC'99)*(Cat. No. 99EX257) (pp. 240-242). IEEE.
- [56] Murphey, Y. L., Chen, Z., Abou-Nasr, M., Baker, R., Feldkamp, T., & Kolmanovsky, I. (2009, June). Ensembles of neural networks with generalization capabilities for vehicle fault diagnostics. In 2009 International Joint Conference on Neural Networks (pp. 2188-2194). IEEE.
- [57] Crossman, J. A., Guo, H., Murphey, Y. L., & Cardillo, J. (2003). Automotive signal fault diagnostics-part I: signal fault analysis, signal segmentation, feature extraction and quasi-optimal feature selection. *IEEE Transactions on Vehicular Technology*, 52(4), 1063-1075.
- [58] Djurdjanovic, D., Liu, J., Marko, K. A., & Ni, J. (2007, August). Immune systems inspired approach to anomaly detection and fault diagnosis for engines. In 2007 International Joint Conference on Neural Networks (pp. 1375-1382). IEEE.
- [59] Dietterich, T. G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Aprendizaje automático*, 40, 139-157.

- [60] Sharma, M., Bhardwaj, R., & Shrivastava, R. (2010). Car failure fuzzy fault diagnostic system based on fault tree analysis. *International Journal of Theoretical & Applied Sciences*, 2(1), 18-21.
- [61] Song, Y. Y., & Ying, L. U. (2015). Decision tree methods: applications for classification and prediction. *Shanghai archives of psychiatry*, 27(2), 130.
- [62] Patel, H. H., & Prajapati, P. (2018). Study and analysis of decision tree based classification algorithms. *International Journal of Computer Sciences and Engineering*, 6(10), 74-78.
- [63] Charkaoui, N., Dubuisson, B., Ambroise, C., & Millemann, S. (2005). A decision tree classifier for vehicle failure isolation. *WIT Transactions on Information and Communication Technologies*, 35.
- [64] Gutierrez, J. M. (s/f). *Sistemas Expertos Basados en Reglas*. Unican.es. Recuperado el 30 de septiembre de 2023, de <https://personales.unican.es/gutierjm/cursos/expertos/reglas.pdf>
- [65] Covarrubias, R. F., & Covarrubias, A. G. F. (2013). Desarrollo de un sistema experto para el diagnóstico de fallas automotrices. *DOAJ (DOAJ: Directory of Open Access Journals)*. <https://doaj.org/article/312d5343fda54345b636e3557ede16a5>
- [66] Tapia Barrientos, J. C. (2007). *Sistema Experto Para El Diagnostico Automotriz [Tesis de Licenciatura]*. Universidad Mayor De San Andrés. Facultad De Ciencias Puras Y Naturales Carrera De Informática.
- [67] Soria Mamani, J. L. (2013) *Sistema Experto Para El Diagnóstico De Fallas En Motores A Inyección Electrónica De Vehículos. [Tesis de Licenciatura]*. Universidad Mayor De San Andrés. Facultad De Ciencias Puras Y Naturales Carrera De Informática.
- [68] Ayala Ramos, A. R. (2000). *Agente para el diagnóstico de motores de automóviles [Tesis de licenciatura]*. Universidad Mayor de San Andrés. Facultad De Ciencias Puras Y Naturales Carrera De Informática.
- [69] Opinautos. (s/f). Opinautos.com. Recuperado el 27 de junio de 2023, de <https://www.opinautos.com/mx>
- [70] Carcomplaints.com. (s/f). Carcomplaints.com. Recuperado el 27 de junio de 2023, de <https://www.carcomplaints.com/>
- [71] Scanator PC. (s/f). *Sistemas de Autodiagnóstico*. Recuperado el 27 de junio de 2023, de <https://scanator.com.mx/>