UNIVERSIDAD TÉCNICA DE AMBATO

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

MAESTRÍA EN MATEMÁTICA APLICADA

COHORTE 2021

TEMA: DISEÑO DE UN MODELO PARA EL CONTROL DEL CONSUMO DE FILTROS Y LUBRICANTES DEL EQUIPO CAMINERO Y MAQUINARIA PESADA DEL GAD DEL CANTÓN LA MANÁ MEDIANTE ALGORITMOS DE INTELIGENCIA ARTIFICIAL

Trabajo de titulación previo a la obtención del Título de Cuarto Nivel de Magister en Matemática Aplicada

Modalidad De Titulación: Proyectos de Desarrollo

Autor: Ing. Juan Carlos Ortiz Reyes

Director: Ing. Edison Fernando Loza Aguirre, PhD

Ambato – Ecuador

2023

A la unidad académica de titulación de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

El Tribunal receptor del Trabajo de Titulación presidido por la Ing Elsa Pilar Urrutia Urrutia Mg., e integrado por la señora Ing Maritza Elizabeth Castro Mayorga Mg., y el Ing Fabian Rodrigo Salazar Escobar PhD., designados por la Unidad de Titulación de la Universidad Técnica de Ambato, para receptar el Trabajo de Titulación con el tema: "Diseño de un modelo para el control del consumo de filtros y lubricantes del equipo caminero y maquinaria pesada del GAD del Cantón La Maná mediante algoritmos de inteligencia artificial", elaborado y presentado por el señor Ing Juan Carlos Ortiz Reyes, para optar por el Grado Académico de Magister en Matemática Aplicada; una vez escuchada la defensa oral del Trabajo de Titulación el Tribunal aprueba y remite el trabajo para uso y custodia en las bibliotecas de la Universidad Técnica de Ambato.

Firmado electrónicamente por:
ELSA PILAR URRUTIA URRUTIA

Ing. Elsa Pilar Urrutia Urrutia Mg
Presidente y Miembro del Tribunal de Defensa

Firmado electrónicamente por:
MARITZA ELIZABETH CASTRO MAYORGA

Ing. Maritza Elizabeth Castro Mayorga Mg
Miembro del Tribunal de Defensa

Firmado electrónicamente por:
FABIAN RODRIGO SALAZAR ESCOBAR

Ing. Fabian Rodrigo Salazar Escobar PhD.
Miembro del Tribunal de Defensa

AUTORÍA DEL TRABAJO DE TITULACIÓN

La responsabilidad de las opiniones, comentarios y críticas emitidas en el trabajo de titulación presentado con el tema: **"Diseño de un modelo para el control del consumo de filtros y lubricantes del equipo caminero y maquinaria pesada del GAD del Cantón La Maná mediante algoritmos de inteligencia artificial"**, le corresponde exclusivamente al: Ing Juan Carlos Ortiz Reyes, autor bajo la dirección del Ing Edison Fernando Loza Aguirre PhD, Director del Trabajo de Investigación; y el patrimonio intelectual a la Universidad Técnica de Ambato.

- -

Ing. Juan Carlos Ortiz Reyes
C.C.: 0502254907
AUTOR

- -

Ing Edison Fernando Loza Aguirre PhD
C.C.: 1713425013
DIRECTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que el Trabajo de Titulación, sirva como un documento disponible para su lectura, consulta y procesos de investigación, según las normas de la Institución. Cedo los Derechos de mi trabajo, con fines de difusión pública, además apruebo la reproducción de este, dentro de las regulaciones de la Universidad Técnica de Ambato.

Juan Carlos Ortiz Reyes
C.C.: 0502254907

INDICE GENERAL DE CONTENIDOS

INDICE DE TABLAS

INDICE DE FIGURAS

AGRADECIMIENTO

En primer lugar, quisiera expresar mi más sincero agradecimiento a mi director de tesis, por su invaluable orientación, apoyo y sabiduría durante todo el proceso de investigación. También quiero agradecer a todos los profesores de esta maestría, por sus valiosas contribuciones a mi investigación y por su disposición a discutir y compartir sus conocimientos en el campo de las Matemáticas Asimismo, me gustaría expresar mi gratitud a mi familia y amigos por su apoyo incondicional, paciencia y comprensión durante los momentos de mayor estrés y ansiedad.

DEDICATORIA

Este trabajo está dedicado a mi padre y a toda mi familia, quienes me han inspirado y motivado a perseguir mis sueños y metas académicas. Su amor, apoyo y aliento han sido una fuente constante de fortaleza y motivación para mí. Agradezco sinceramente su presencia en mi vida y su contribución a mi crecimiento personal y profesional.

UNIVERSIDAD TÉCNICA DE AMBATO

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

MAESTRÍA EN MATEMÁTICA APLICADA

COHORTE 2021

TEMA:

"Diseño de un modelo para el control del consumo de filtros y lubricantes del equipo caminero y maquinaria pesada del GAD del Cantón La Maná mediante algoritmos de inteligencia artificial"

MODALIDAD DE TITULACIÓN: Proyectos de desarrollo

AUTOR: Ing. Juan Carlos Ortiz Reyes

DIRECTOR: Ing. Edison Fernando Loza Aguirre PhD

FECHA: 15 de septiembre 2023

RESUMEN EJECUTIVO

El objetivo principal de este estudio fue diseñar un modelo matemático utilizando algoritmos de inteligencia artificial para predecir el consumo de lubricantes en la maquinaria pesada del GAD del Cantón La Mana. Para ello se recopilo información de diferentes tipos de algoritmos de inteligencia artificial que podrían ser útiles para

la predicción de consumo, y se eligió la red neuronal artificial no lineal autorregresiva como la mejor opción. La información utilizada en este estudio fue obtenida de los registros diarios de la Unidad de Transporte y Maquinaria del GAD Municipal La Mana´ en los años 2018 y 2019, donde se registraron los kilometrajes y horómetros diarios al inicio y al final de la jornada laboral. La precisión del modelo se evaluó mediante el cálculo del error medio cuadrático (MSE), que mide la diferencia cuadrática promedio entre los valores predichos y los valores reales. Los resultados mostraron que la red neuronal que utiliza el algoritmo Retropropagación Levenberg-Marquardt con la arquitectura 128, 64 en las capas ocultas de la red neuronal fue el mejor modelo, con un MSE de 0.000301. En resumen, se concluye que el modelo de red neuronal no lineal autorregresiva puede ser una herramienta útil para predecir el kilometraje y horómetro de las maquinaria pesada, y por ende estimar el consumo de lubricantes, lo que podría permitir una mejor planificación y optimización del uso de lubricantes.

Descriptores: Modelos matemáticos, inteligencia artificial, redes neuronales no lineales autorregresivas, error medio cuadrático, predicción.

Capítulo I

PROBLEMA DE INVESTIGACIÓN

El crecimiento económico se ve afectado por la calidad y desarrollo de la infraestructura básica, incluyendo carreteras, acceso a electricidad y agua potable. En un estudio realizado [1], se concluye que la infraestructura vial tiene un mayor impacto en el crecimiento económico en comparación con otras infraestructuras fundamentales. El desarrollo de la infraestructura vial en un país está positivamente correlacionado con el crecimiento económico, ya que un eficiente movimiento de personas, bienes y servicios depende en gran medida del acceso a carreteras y vías. Esto también permite a la sociedad participar en una amplia gama de actividades comerciales y sociales [2].

En [3] se demuestra estadísticamente que la infraestructura vial juega un papel positivo en la promoción del crecimiento económico de los países. Es crucial que la dotación de infraestructura en una región esté vinculada a proyectos y políticas públicas que fomenten el crecimiento económico y el beneficio social, así como el desarrollo urbano. La cantidad de vías en un área afecta directamente su capacidad para abarcar actividades sociales y económicas [4].

Las carreteras y la infraestructura de transporte son fundamentales tambien para el desarrollo de la industria del turismo, ya que facilitan el acceso a destinos turísticos y fomentan la actividad comercial en la zona, mejorando así el nivel de vida de la población local [5, 6].

Además de ayudar a reducir la pobreza y la desigualdad socio territorial, las infraestructuras viales también facilitan la producción de bienes y servicios, lo que tiene un impacto positivo en la productividad [7]. América Latina es una región en de-

sarrollo con las tasas más bajas de crecimiento económico e inversión pública de cualquier región, así como también tiene la peor infraestructura, lo que conduce a que un 62,8 % de su población esté viviendo en la pobreza o en riesgo de pobreza [8, 9]. Esta característica se explica por el sesgo generalizado contra el gasto de capital entre los gobiernos regionales. Por ejemplo, Ecuador se ha vuelto cada vez más dependiente del capital extranjero para la construcción de su infraestructura; según algunas estimaciones, China suministró préstamos para el 61% de las necesidades financieras del país [10]. Sin embargo, el Estado Ecuatoriano a través del Plan de Creación de Oportunidades 2021 – 2025 propone políticas y metas con el afán de mejorar el acceso y calidad de las viviendas, la disponibilidad de servicios públicos, entre otros [11]. Asimismo, el GAD Municipal de La Maná cuenta con el Plan de Desarrollo y Ordenamiento Territorial del Cantón La Maná, mismo que en el Componente de Movilidad, Energía y Conectividad tiene como objetivo mejorar la infraestructura vial urbana, y como política afín el promover el mejoramiento de sistema vial urbano [12].

En este contexto, el equipo de maquinaria pesada constituye una pieza fundamental en la realización de trabajos de apertura y mantenimiento de vías en el Cantón La Maná. Por esta razón, su operatividad se convierte en una de las prioridades para el GAD Municipal de La Maná, siempre con el afán de brindar a la comunidad una red vial en buen estado que garantice el buen vivir de los habitantes. El mantenimiento de la maquinaria pesada juega un rol fundamental en el afán de mantenerla operativa. El mantenimiento predictivo y preventivo ha sido probada en diferentes estudios con el afán de reducir el costo de las operaciones y mejorar la confiabilidad. Por ejemplo, en [13] los autores concluyen que el mantenimiento de equipos rotativos con un man-

tenimiento preventivo cuesta un 30 % menos que con un modelo correctivo o también llamado reactivo. Asimismo, el estudio establece que las tecnologías predictivas agregan un retorno de la inversión adicional significativo.

Los avances tecnológicos han permitido el surgimiento de un nuevo paradigma de mantenimiento, métodos y herramientas innovadores, lo cual denominan como Mantenimiento 4.0 [14]. El término ”Mantenimiento 4.0”, también conocido como ”Mantenimiento inteligente”, se refiere al desarrollo de métodos que aseguran la integridad de piezas, bienes y sistemas al identificar, anticipar o prever problemas provocados por fallas en el desempeño que podrían tener un efecto negativo, al realizar análisis predictivos y hacer recomendaciones viables, particularmente para los aspectos de mantenimiento que se ocupan de la recopilación de datos, el análisis y la toma de decisiones [15]. En este contexto, el uso de métodos basados en datos como el Aprendizaje Automático (Machine Learning - ML) se convierten cada vez más una alternativa viable y confiable para actividades que van desde el mantenimiento predictivo hasta la calidad predictiva [16, 14].

El GAD La Maná dispone de equipo caminero y maquinaria pesada cuya tasa de daños es elevada. Uno de los factores principales de esta problemática es la ausencia de un plan de mantenimiento, por lo cual, es necesario implementar alternativas eficientes que permitan tener un control de las actividades de mantenimiento vehicular, beneficiando a la flota del GAD para el correcto desarrollo de sus actividades cotidianas. En este contexto, la realización de un modelo matemático permitirá implementar un mantenimiento predictivo, el cual tiene como objetivo principal la disminución del número de mantenimientos correctivos realizados en un vehículo, obteniendo así

mayor rendimiento [17].

1.1 Justificación

El mantenimiento de maquinaria es un factor crucial en cualquier industria, ya que puede afectar tanto la eficiencia como el tiempo de operación de los equipos [18]. En general, la práctica de mantenimiento suele ser correctiva, lo que significa que los equipos se reparan o reemplazan solo después de que fallan [19]. La lubricación inadecuada es una de las principales causas de desgaste prematuro en las piezas de maquinaria en movimiento [20]. Por lo tanto, es importante identificar y solucionar las fallas de los equipos para evitar interrupciones en los procesos de producción y optimizar los recursos [18, 19].

En el caso del GAD La Maná, se ha observado una alta tasa de daños en su equipo caminero y maquinaria pesada, principalmente debido a la falta de un plan de mantenimiento adecuado. En este contexto, la inteligencia artificial (IA) se ha establecido como una herramienta potencialmente poderosa para abordar estos problemas. A través de técnicas como el uso de redes neuronales o lógica difusa, la IA permite detectar de manera eficiente fallas en el equipo y maquinaria. El análisis de datos y el ML ofrecen grandes ventajas en la detección automática de anomalías en vehículos, así como en la alerta temprana de posibles fallos. Además, la predicción de fallos es un campo de aplicación clave, ya que permite anticipar problemas y realizar mantenimiento preventivo de manera más eficiente. Estas herramientas tecnológicas tienen el potencial de mejorar significativamente la eficacia y vida útil del equipo, reduciendo los costos de reparación y aumentando la productividad general.

En resumen, la detección y predicción automatizadas de fallos en la maquinaria son enfoques algorítmicos esenciales para un mantenimiento inteligente y predictivo, lo que permite un control efectivo de las actividades de mantenimiento vehicular en el GAD La Maná. Estas soluciones tecnológicas pueden beneficiar significativamente la flota de vehículos del GAD al garantizar el correcto desarrollo de sus actividades diarias. Por lo tanto, la implementación de un modelo matemático puede ser una solución efectiva para la implementación de un mantenimiento predictivo, cuyo principal objetivo es reducir la cantidad de mantenimientos correctivos realizados en los vehículos y lograr un mayor rendimiento [17]. El consumo de lubricantes mensuales del equipo caminero puede ser predicho a partir de los datos de kilometraje y tiempo de uso de los vehículos.

1.2 Objetivos

1.2.1 Objetivo General

Diseñar de un modelo matemático para el control del consumo de filtros y lubricantes del equipo caminero y maquinaria pesada del GAD del Cantón La Maná mediante algoritmos de inteligencia artificial.

1.2.2 Objetivos específicos

- Investigar los tipos de algoritmos basados en inteligencia artificial para aplicar al proyecto.
- Recopilar los datos mediante la utilización de métodos estadísticos para realizar

el dataset.

- Desarrollar un algoritmo basado en inteligencia artificial que pueda controlar el consumo de filtros y lubricantes del equipo caminero y maquinaria pesada del GAD del Cantón La Maná.

- Implementar ajustes de hyperpárametros para determinar la precisión del modelo.

1.3 Antecedentes Investigativos

Como preámbulo de esta investigación, es necesario analizar la situación energética actual del Ecuador. En este contexto, el organismo encargado de proporcionar información relacionada es la Agencia de Regulación y Control de Energía y Recursos Naturales No Renovables, el mismo que proporciona una estadística anual sobre el estado del sector eléctrico ecuatoriano (ARCERNNR).

En las últimas dos décadas se ha realizado una investigación significativa para la gestión de operaciones, con un enfoque en el mantenimiento, y los investigadores han hecho una serie de sugerencias para resolver algunos problemas bien definidos. Estos problemas se presentan en la mayoría de industrias, y todas ellas, incluyendo la industria automotriz han dirigido la atención hacia el mantenimiento preventivo, debido las ventajas claves que ofrecen, tales como: mejorar el rendimiento de los vehículos, diagnosticar los vehículos para prevenir los riesgos a los que el vehículo y el operador puede enfrentarse, entre otros [21].

Hoy en día, los fabricantes de equipos originales de vehículos de transporte suelen

diseñar planes de mantenimiento basados en parámetros simples como el tiempo calendario o el kilometraje. Sin embargo, esto ya no es suficiente en el mercado y existe la necesidad de enfoques más avanzados que proporcionan predicciones de las futuras necesidades de mantenimiento de camiones. En lugar de vender solo vehículos, el sector se encamina hacia la venta de servicios completos de transporte; por ejemplo, una flota de camiones, incluido el mantenimiento, con un nivel de disponibilidad garantizado. Esta predicción de las necesidades futuras de mantenimiento de los equipos se puede abordar de muchas maneras diferentes. Un enfoque consiste en monitorear el equipo y detectar patrones que señalan una falla emergente.

Sin embargo, las soluciones asumen que se dispone de datos precisos, y muchos de ellos son demasiado caros desde el punto de obra para ser prácticos. Los métodos típicos de diagnóstico y mantenimiento predictivo requieren una amplia experimentación y modelado durante el desarrollo. Esto es inviable si se aborda el vehículo completo, ya que requeriría demasiados recursos de ingeniería.

La IA es reconocida por muchos investigadores como una herramienta potencialmente poderosa para resolver estos problemas de mantenimiento, incluido el uso de redes neuronales o lógica difusa para detectar fallas. Una gran promesa del análisis de datos y el aprendizaje automático es detectar anomalías en los vehículos automáticamente y alertar a sus usuarios de las fallas que ocurren. Como extensión, la predicción de fallos en vehículos es un importante campo de aplicación. La detección y predicción automatizadas de fallos en la maquinaria es un enfoque algorítmico clave detrás del mantenimiento inteligente y predictivo.

Capítulo II

MARCO TEÓRICO

Este capítulo presenta la base teórica empleada para la investigación, donde se detallan temas importantes como son el Mantenimiento, sus tipos con las respectivas ventajas y desventajas durante su implementación. Luego se muestra una revisión acerca de la IA, enfocándose principalmente en las redes neuronales. Y, finalmente se presenta una reseña sobre la las métricas de desempeño utilizadas para evaluar modelos de IA.

2.1 Mantenimiento

El objetivo del mantenimiento, desde el punto de vista tradicional, es reparar los componentes dañados. Desde esta perspectiva tan limitada, las tareas solo serán reactivas y consistirán en acciones para reparar o reemplazar elementos defectuosos. Esta estrategia también se conoce como mantenimiento reactivo, mantenimiento por avería o mantenimiento correctivo. Una definición más moderna de mantenimiento se podría establecer como todas las actividades destinadas a mantener o restaurar un artículo en el estado físico considerado necesario para el cumplimiento de su función de producción. Esta perspectiva ampliada también cubre acciones proactivas como mantenimiento regular, inspección periódica, reemplazo preventivo y monitoreo de condición [22].

En [23] se define al mantenimiento como una serie de acciones que tienen como objetivo mantener los equipos en una condición operativa lo más cercana posible a

su estado teórico o nominal, con la menor inversión, de una manera que sea segura para las personas y el medio ambiente, y que apoye el logro de los objetivos de una organización.

En [24] se detalla otra definición de mantenimiento como el conjunto de técnicas destinado a conservar equipos e instalaciones industriales en servicio durante el mayor tiempo posible y con el máximo rendimiento. De acuerdo a [25], los distintos tipos de mantenimiento se los puede resumir como se muestra en la Figura 2.3.

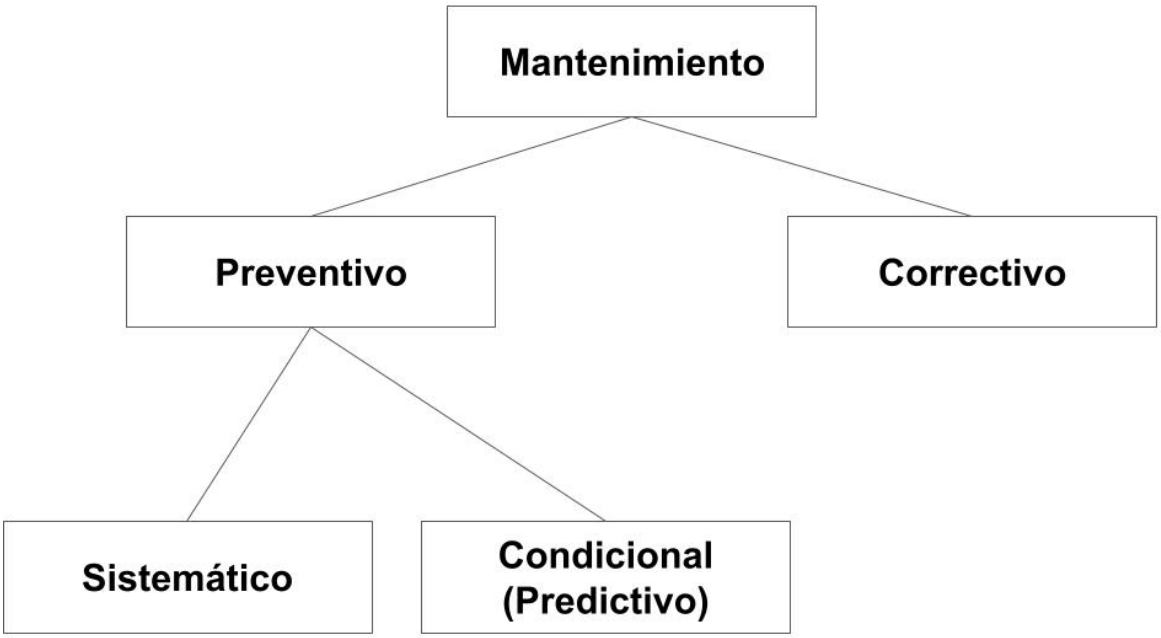


Figure 2.1: Diferentes tipos de mantenimiento [25].

2.1.1 Mantenimiento Correctivo

Es aquel que es efectuado después de que ocurre fallos, con el afán de reparar averías [25].

Ventajas: [25]

- No requiere de gran infraestructura técnica, ni elevada capacidad de análisis.

- Máximo aprovechamiento de la vida útil de los equipos.

Inconvenientes: [25]

- Las averías se presentan de manera imprevista.
- Existe riesgo de fallos en elementos difíciles de adquirir.
- Poco tiempo disponible para reparar.

2.1.2 Mantenimiento Preventivo

Este mantenimiento es realizado con el objetivo de reducir la probabilidad de fallo, de cual se puede tener dos modalidades [25].

Mantenimiento Preventivo Sistemático: Es efectuado durante intervalos regulares de tiempo según un programa establecido, en el cual se toma en cuenta la criticidad de cada máquina.

Mantenimiento Preventivo Condicional: También denominado Mantenimiento Preventivo según condición, el cual está motivado debido a un acontecimiento predeterminado.

Ventajas [25]:

- Importante reducción de paradas imprevistas en equipos.
- Adecuado cuando existe una cierta relación entre probabilidad de fallos y duración de vida.

Inconvenientes [25]:

- La totalidad de la vida útil del equipo no se aprovecha.

- Aumenta el gasto y disminuye la disponibilidad si no se elige convenientemente la frecuencia de las acciones preventivas [25].

2.1.3 Mantenimiento Predictivo

Este tipo de mantenimiento hace referencia a las técnicas de detección temprana de síntomas, lo cual sirve para intervenir antes de la aparición del fallo [25].

Ventajas:

- Determinación óptima del tiempo para realizar el mantenimiento preventivo.
- Ejecución sin interrumpir el funcionamiento normal de equipos e instalaciones.
- Mejora el conocimiento y el control del estado de los equipos.

Inconvenientes:

- Requiere personal mejor formado e instrumentación de análisis costoso.
- No es viable una monitorización de todos los parámetros funcionales significativos, por lo que pueden presentarse averías no detectadas por el programa de vigilancia.
- Se pueden presentar averías en el intervalo de tiempo comprendido entre dos medidas consecutivas [25].

2.2 Inteligencia Artificial

La IA surge como una respuesta del desarrollo tecnológico que atraviesa el mundo, en su deseo de aproximar el comportamiento y el pensamiento humano a diversos

sistemas para la solución de determinadas problemáticas. En la actualidad, existen sistemas altamente desarrollados capaces de imitar ciertos aspectos humanos. No obstante, aún queda mucho por recorrer para poder reproducir otros aspectos [26].

La IA es el estudio de cómo programar computadoras para llevar a cabo funciones cognitivas que, en el pasado, solo podían ser realizadas por personas. El rápido desarrollo de la IA en los últimos años ha alterado la forma de vida de las personas [27]. El enfoque actual de la IA es en realidad una mezcla de múltiples campos de investigación, cada uno con su propio objetivo, métodos, situaciones aplicables entre otros [3].

La inteligencia artificial se puede dividir en siete categorías, según la forma de resolver problemas: detección, agrupación, clasificación, predicción, optimización, capacidad de generalización y capacidad de creación [28, 29].

2.3 Redes Neuronales

Una red neuronal artificial (ANN) es un sistema nervioso biológicamente inspirado. Tiene una gran cantidad de elementos de procesamiento denominados neuronas, que están interconectados en la red [30]. Las redes neuronales artificiales son aproximaciones no lineales de cómo funciona el cerebro; como resultado, no deben compararse directamente con el cerebro, sus principios operativos no deben confundirse con los del cerebro, y las redes neuronales no deben considerarse basadas únicamente en redes biológicas, ya que solo imitan la funcionalidad del cerebro humano de una manera muy simplificada [31].

En general, una ANN es un sistema adaptativo que ajusta su organización en

función de la información que se ejecuta a través de la red durante el proceso de aprendizaje [32]. Una red neural toma como ejemplos problemas resueltos para construir un sistema que toma decisiones y realiza clasificaciones. Los problemas adecuados para la solución neural son aquellos que no tienen solución computacional precisa o que requieren algoritmos muy extensos como en el caso del reconocimiento de imágenes [26]. Las ventajas y desventajas de las redes neuronales se las detalla en la Tabla 2.1.

Table 2.1: Ventajas y desventajas de las redes neuronales.

Ventajas	**Desventajas**
Sintetizan algoritmos a través de un proceso de aprendizaje.	Deben ser entrenadas para cada problema y es necesario realizar múltiples pruebas para determinar la arquitectura adecuada.
No es necesario conocer los detalles matemáticos, sino estar familiarizado con los datos del problema.	El entrenamiento es largo y puede consumir varias horas de la computadora (CPU).
Permiten la solución de problemas no lineales.	Debido a que las redes se entrenan, es necesario la obtención de un conjunto de datos grande.
Son robustas, es decir que pueden fallar algunos elementos de procesamiento, pero la red continúa trabajando.	

Las redes neuronales están compuestas de nodos o unidades (neurona artificial) conectadas a través de conexiones dirigidas. La neurona artificial tiene varias entradas de estímulo ($X = [x_1, x_2, \ldots, x_n]$). que pueden provenir del sistema sensorial externo o de otras neuronas con las cuales posee conexión. La información recibida por la neurona es modificada por un vector de pesos sinápticos ($W = \left[W_{j1}, W_{j2}, \ldots, W_{jn}\right]$) con el afán de emular la sinapsis propia de las neuronas biológicas. El parámetro W0 se conoce como bias. Los valores de entrada de la neurona que son modificados por los pesos sinápticos son combinados mediante una suma básica. La suma de las entradas es modificada a través de una función de activación y el valor de la salida de

esta función es directamente transferida a la salida del elemento [33].

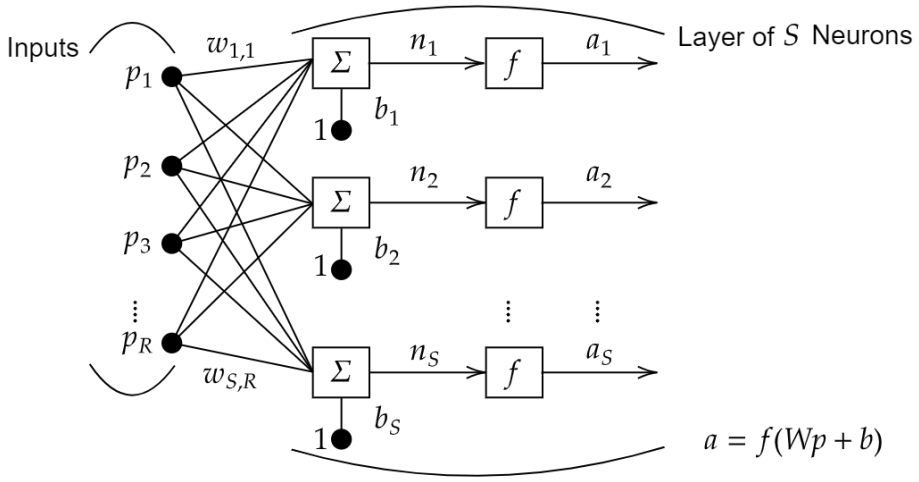


Figure 2.2: Diagrama de una red neuronal artificial [33].

La entrada neta en una neurona artificial se puede calcular mediante la ecuación (2.1), y en forma vectorial se representa con la ecuación (2.2)

$$Net_j = \sum_{i=1}^{N} x_i w_{ij} + \theta_j, \tag{2.1}$$

$$Net_j = w_1 x_{j1} + w_2 x_{j2} + \cdots + w_N x_{jN} + \theta_j \tag{2.2}$$

$$Net_j = w^T X_j + \theta_j$$

La salida de la neurona artificial está determinada por una función de activación (Fact), tal como se aprecia en la ecuación (2.3)

$$y_j = Fact_j(Net_j) \tag{2.3}$$

Existen varias funciones de activación, tales como el escalón, escalón simétrico, logística y la tangente hiperbólica. En la Figura 2.3 se muestran algunas funciones de

activación [34].

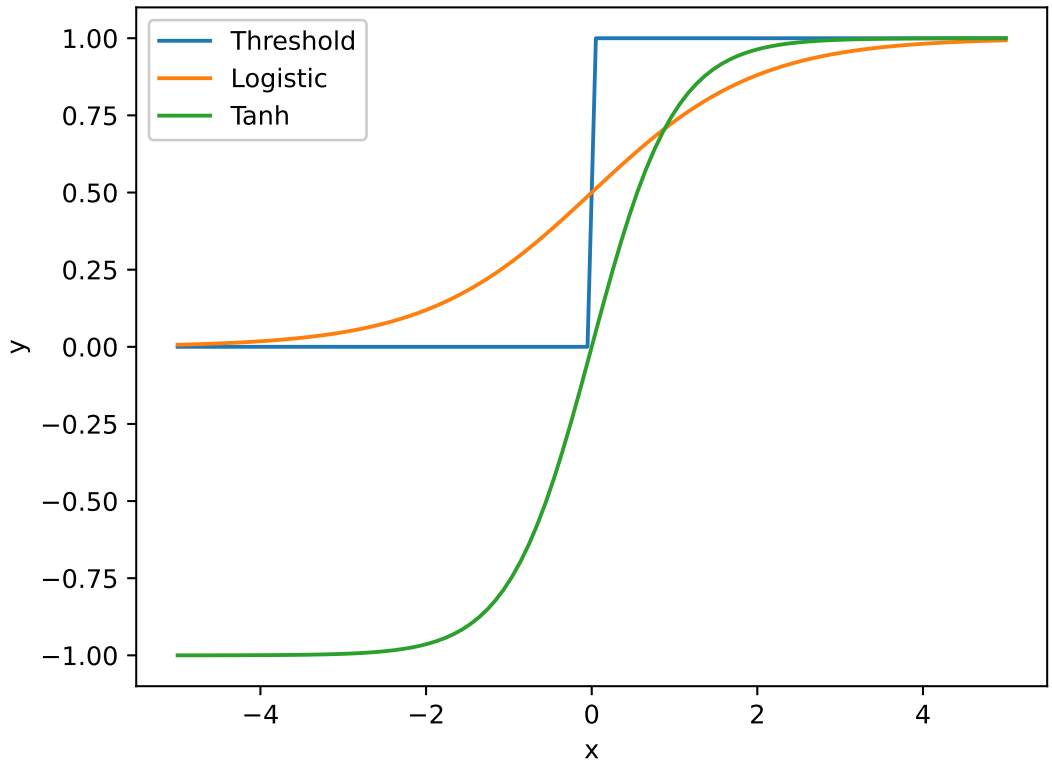


Figure 2.3: Funciones de activación.

El poder de procesamiento y el nivel de aplicabilidad de la neurona artificial como tal son limitados cuando se usan solas; su verdadero potencial radica en la interconexión, tal como ocurre en el cerebro humano. Las redes neuronales artificiales se crearon como resultado de las propuestas de varios investigadores con el afán de proponer diferentes estructuras para conectar las neuronas entre sí [33].

El proceso de aprendizaje de una Red de Neuronas Artificiales es su componente más crucial. Los tipos de problemas que una red neuronal puede resolver están determinados por su estrategia de aprendizaje. El tipo de ejemplos disponibles durante el proceso de aprendizaje tendrá un impacto fundamental en la capacidad de una red

para resolver un problema. Desde la perspectiva de los ejemplos [35], el conjunto de aprendizaje debe tener las siguientes cualidades:

- El número de ejemplos debe ser el suficiente, ya que, si el conjunto de aprendizaje es pequeño, la red no podrá adaptar sus pesos de manera efectiva.
- El conjunto de aprendizaje debe tener diversidad de ejemplos. Esto con el afán de que la red no se especialice en un solo subconjunto de datos y no sea de aplicación general.

En una ANN, el aprendizaje es el proceso de calcular los valores de peso precisos para cada una de sus conexiones, lo que le permite resolver problemas rápidamente. El proceso general de aprendizaje implica la introducción de cada ejemplo del conjunto de aprendizaje de uno en uno y el cambio de los pesos de las conexiones de acuerdo con un esquema de aprendizaje predeterminado [35].

2.3.1 Perceptrón de una capa

Cuando se utilizan más neuronas con la función de activación escalón, tenemos un perceptrón de una sola capa. El perceptrón de una sola capa se puede utilizar para clasificar los datos del vector de entrada en varias clases [34]. Las salidas de las neuronas pueden ser calculadas por:

$$a = f(Wp + b). \tag{2.4}$$

2.3.2 Redes Neuronales No lineales Autorregresivas

La red neuronal autorregresiva (NAR) no lineal es un tipo de ANN apropiado para la estimación de valores futuros de la variable de entrada. La red NAR permite predecir valores futuros de una serie temporal, apoyándose en sus antecedentes históricos, mediante un mecanismo de realimentación, en el que un valor predicho puede servir de entrada para nuevas predicciones en puntos más avanzados del tiempo. La red se crea y se entrena en bucle abierto, utilizando los valores objetivo-reales como realimentación y garantizando así una mayor precisión en el entrenamiento. Tras el entrenamiento, la red se convierte en un bucle cerrado y los valores predichos se utilizan para suministrar nuevas entradas de retroalimentación a la red [36].

Matemáticamente, el modelo pronostica valores futuros de una serie de tiempo $y(t)$, con base en sus valores históricos $y(t-1), y(t-2), \ldots, y(t-d)$ (modelo NAR) utilizando adicionalmente una serie de tiempo externa $x(t-1), x(t-2), \ldots, x(t-d)$, donde d es el parámetro de retardo de tiempo. El entrenamiento de la red se realiza, generalmente, mediante el algoritmo de retro propagación, y utiliza el método de descenso más pronunciado para minimizar el error cuadrático entre los valores reales y los predichos [36].

2.4 Métricas de desempeño

En ML las métricas de desempeño se utilizan para comparar las predicciones del modelo entrenado con los datos reales del conjunto de datos de prueba. La predicción tiene una larga historia de empleo de métricas de rendimiento para medir cuánto se desvían

las predicciones de las observaciones con el fin de evaluar la calidad y elegir métodos de predicción [37]. Los investigadores en estudios de clasificación predictiva utilizan usualmente varias métricas para evaluar el desempeño de sus modelos, tales como la exactitud, la sensibilidad, la matriz de confusión, el puntaje F1, la curva ROC y el área bajo la curva (AUC). La exactitud y la sensibilidad son eficaces cuando se trabaja con datos equilibrados, su confiabilidad disminuye cuando se trabaja con datos desequilibrados. El uso de diversas métricas de rendimiento y la falta de estandarización en su selección a menudo dificultan la comparación de resultados entre investigaciones que no se han evaluado en condiciones similares [38].

Las métricas de desempeño son escogidas dependiendo del tipo de modelo de ML, por ejemplo, si se utiliza un modelo de clasificación binaria, las métricas a utilizarse son curva ROC o matriz de confusión; por otro lado, si se utiliza un modelo de regresión para predecir la respuestas o resultados utilizando variables continuas, las métricas a utilizarse son el error cuadrático medio (MSE), error absoluto medio (MAE) o el coeficiente de determinación (R^2) [39].

La exactitud 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. La sensibilidad es una medida que permite conocer la proporción de casos positivos que fueron correctamente clasificados. En un modelo perfecto la sensibilidad es igual a 1 para cada clase. Desde el punto de vista analítico un investigador busca aumentar la sensibilidad sin afectar el valor de la exactitud [38].

La medida F1-score fusiona las métricas accuracy con recall, presentando diferencias en el rendimiento de un clasificador que no son revelados únicamente con la accuracy. Es directamente proporcional al aumento de las dos medidas, por lo tanto, valores altos de F1-score demuestra que el algoritmo de clasificación predice de mejor manera la clase positiva. Otra métrica utilizada a menudo para la evaluación de rendimiento en clasificadores dicotómicos es la curva ROC, que se trata de una gráfica bidimensional de la sensibilidad versus (1 – especificidad) por cada clase. La medida de comparación en esta gráfica es la AUC que corresponde al área bajo la curva ROC y sus valores se encuentran entre 0 y 1, considerando que en el caso totalmente aleatorio se tiene una AUC igual a 0,5 [38].

El error cuadrático medio es, por construcción, una medida de la precisión predictiva del modelo, denotado MSEP es simplemente la diferencia cuadrática promedio entre la cantidad de interés y la predicción del modelo de esa cantidad. Depende de los valores particulares de los parámetros que se utilizan en el modelo, como indica la notación, se define de la siguiente manera:

$$MSEP(\hat{p}) = E\left[\left(y - f(X, \hat{p})\right)^2 |\hat{p}\right]$$

Capítulo III

MARCO METODOLÓGICO

3.1 Ubicación

El proyecto de investigación está desarrollado en base al equipo caminero del Gobierno Autónomo Descentralizado (GAD) Municipal de La Maná. Este municipio pertenece al cantón La Maná, que esta ubicada en la provincia de Cotopaxi, Ecuador. [40].

3.2 Equipos y materiales

En nuestra metodología, utilizamos varias herramientas informáticas, como Excel, MATLAB, R y Python, en un computador con Windows equipado con un procesador Intel i5-7200U, para llevar a cabo nuestros análisis y procesamiento de datos de manera eficiente.

3.3 Comparación entre Matlab y Python

En la comparación entre MATLAB y Python para el entrenamiento de una red neuronal NAR, ambos tienen sus ventajas y desventajas. MATLAB ofrece una interfaz amigable y optimización de rendimiento, pero puede ser costoso y menos flexible en términos de personalización. Por otro lado, Python proporciona una amplia gama de bibliotecas y flexibilidad, junto con una comunidad activa y es de código abierto, pero

puede requerir un mayor aprendizaje inicial y configuración.

3.4 Metodología

La estimación del consumo de lubricantes del del equipo caminero y maquinaria pesada del GAD del Cantón La Maná se desarrolla mediante metodología experimental apoyándose de simulaciones numéricas realizadas en el software Matlab y Python a través de algoritmos de AI.

En la Figura 3.1 se presenta el diagrama de flujo sobre las actividades que se llevan a cabo para realizar este estudio. Primero se realizan los trabajos diarios propios de la maquinaria pesada del GAD Municipal, entre los que constan reconformación, nivelación y compactación de las diferentes vías urbanas del Cantón La Maná. Luego, la Unidad de Transporte y Maquinaria de la entidad municipal registra diariamente en sus archivos cada uno de los kilometrajes y odómetros de las diferentes unidades vehiculares. Para el presente estudio, se ha recopilado la información descrita anteriormente durante los años 2018 y 2019. Los datos del 2020 no se consideraron por ser atípicos debido a la emergencia nacional.

Posterior, se realiza un preprocesamiento de los datos, puesto que en el registro de los kilometrajes y odómetros solamente cuentan con los días laborables en el sector público, mientras que para el modelo es necesario contar con los datos de todos los días del año. En este paso, se normalizan los datos con el afán de evitar la influencia de las diferentes escalas de medición, lo cual es fundamental para la consecución de resultados favorables por parte del modelo de Red Neuronal implementada en este trabajo. Después, se procede a la implementación de la NAR, la misma que sirve

para predecir valores de series temporales, con lo cual tendríamos la estimación de los kilometrajes y odómetros para un año fiscal posterior.

Una vez que el modelo está entrenado, se puede utilizar para predecir el nivel de consumo de la maquinaria a partir de nuevos valores de odómetro y kilometraje. Estos valores permiten obtener una única salida correspondiente al nivel de consumo estimado de la maquinaria.

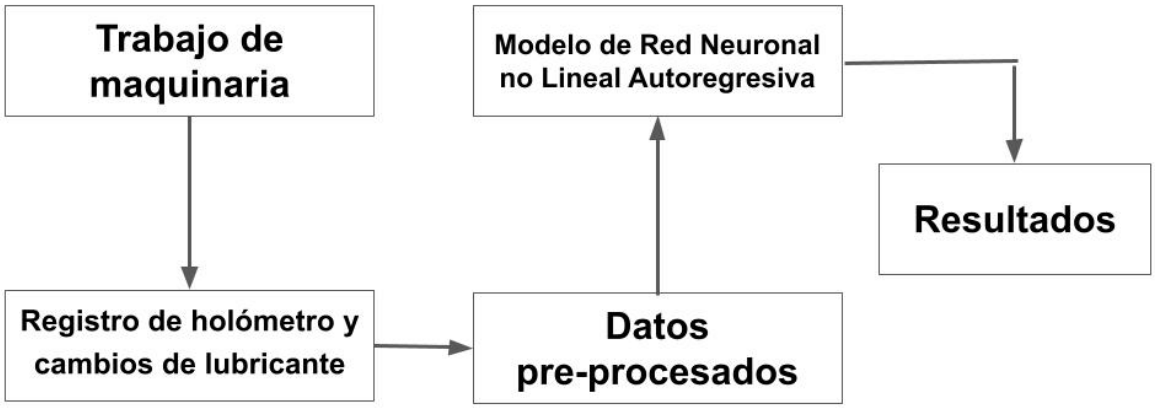


Figure 3.1: Metodología implementada.

3.4.1 Maquinaria analizada

Este estudio abarca diferentes unidades operativas del equipo caminero y maquinaria pesada del GAD del Cantón La Maná, para lo cual se han escogido como caso de estudio las enlistadas en la Tabla 3.1.

Table 3.1: Maquinaria analizada en este estudio

Ítem	Maquinaria	Marca	Número de vehículos	Tipo de aceite
1	Volquete	Volkswagen	4	15W40
2	Volquete	Hino GH	1	15W40
3	Volquete	Hino FF	1	25W50
4	Volquete	KB	1	25W50
5	Camión	Chevrolet NPR	1	25W50
6	Camioneta	Chevrolet D-MAX	2	15W40
7	Camioneta	Chevrolet LUV	1	15W40
8	JEEP	Toyota	1	25W50
9	Camión	Hino	2	25W50

3.4.2 Recolección de Datos

Los datos del presente proyecto son resultado del registro diario recopilado por la Unidad de Transporte y Maquinaria del GAD Municipal La Maná. Estos datos han sido recogidos durante los años 2018 y 2019, constan de los kilometrajes y horómetros diarios con los que se inicia la jornada laboral y con los que se termina la misma.

Podemos describir nuestro dataset de la siguiente manera:

1. **VEHÍCULO - MAQUINARIA**:
    - Descripción del vehículo o maquinaria, indicando su tipo y uso.
    - TIPO: Tipo de vehículo o maquinaria.
    - TIPO de aceite: Tipo de vehículo o maquinaria.
2. **MARCA**:
    - MARCA: Marca del vehículo o maquinaria.
    - COLOR: Color del vehículo o maquinaria.
3. **CÓDIGO**:

- CÓDIGO: Código de identificación del equipo.
- PLACAS: Número de placas del vehículo o maquinaria.
- FECHA: Fecha en la que se registraron los datos.
- Km - Hr POR CAMBIO: Kilometraje o horas de uso acumulados en el equipo desde el último cambio de aceite.

4. **MANT-ANTERIOR**:

    - MANT-ANTERIOR: Información sobre el mantenimiento anterior realizado en el equipo.

5. **PROGRAMACIÓN APROXIMADA EN MESES CON KILOMETRAJES Y HORÓMETROS DURANTE EL AÑO**:

    - Datos de programación de mantenimiento con estimaciones aproximadas en meses, kilómetros y horas durante el año.

En la Figura 3.3 se puede observar el diagrama de flujo desarrollado para la realización de la red neuronal que permitirá predecir los kilómetros y consumos previo de la maquinaria se realizarán durante un año por parte de la maquinaria pesada del GAD Municipal de La Maná. Al tener los datos recolectados, es necesario identificar las variables de entrada y salida para nuestra red, los cuales serían los kilómetros y horas trabajadas a lo largo de cada día, es entonces que se realiza el preprocesamiento de los datos, los mismos que permiten crear el dataset y dividir los datos para el entrenamiento, validación y prueba del modelo. Después, se define las diferentes arquitecturas de la red neuronal, se procede a realizar el entrenamiento con diferentes

algoritmos, para así evaluar el modelo a través del error cuadrático medio. Finalmente, se analiza que modelo presenta el mejor rendimiento y permite la predicción de los datos de manera confiable.

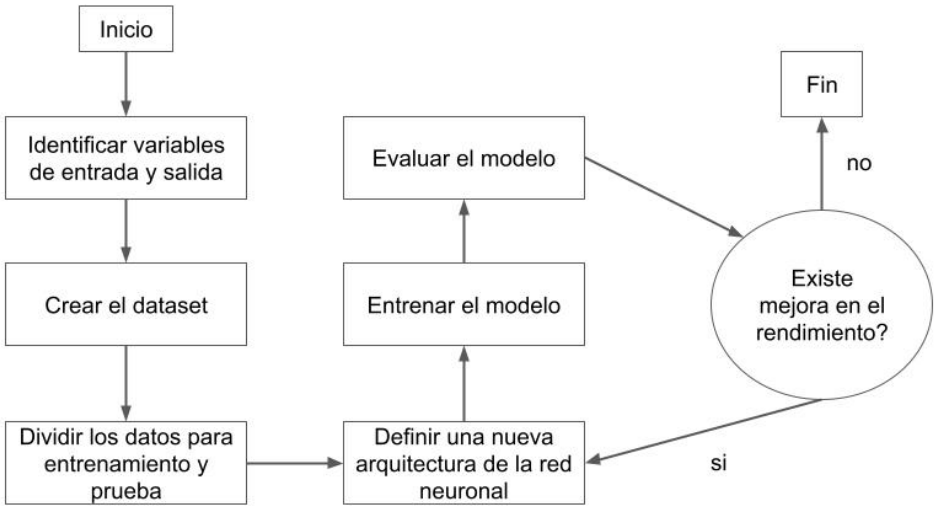


Figure 3.2: Metodología implementada.

3.4.3 Preprocesamiento de Datos

El preprocesamiento de los datos empleado en este estudio consiste en dos actividades. La primera que consiste en rellenar con ceros la serie temporal, con el afán de que existan datos válidos durante todo el año, puesto que los registros diarios la Unidad de Transporte y Maquinaria solo los realiza durante jornadas laborales para el sector público. La segunda actividad hace referencia a la normalización de los datos, lo cual es importante para conseguir que todos los datos tengan aproximadamente la misma escala, lo cual tiene influencia directa en la red neuronal, puesto que:

- Facilita la inicialización de los pesos

- Entrenamiento de la red neuronal más rápido
- Producen mejores resultados

La normalización está definida como sigue:

$$X_n = a + \frac{(X - X_{min})(b - a)}{X_{max} - X_{min}}$$

Donde X_n es el valor normalizado; X es el valor a normalizar; a, b son los valores inferior y superior del rango a reducir; mientras que X_{min} y X_{max} Son los valores máximos y mínimos de los datos de entrada.

3.4.4 Modelo de Red Neuronal No Lineal Autorregresiva

Una (NAR) es un tipo de red neuronal que utiliza una combinación de entradas actuales y anteriores para predecir la salida futura. Este modelo se puede utilizar para predecir los consumos de una maquinaria de transporte pesada. En este caso, la red neuronal NAR utiliza una arquitectura con activaciones sigmoid en sus capas ocultas y activación ReLU en su capa de salida.

La red neuronal NAR consta de una capa de entrada que recibe las entradas actuales y anteriores, tales como el kilometraje y el consumo previo de la maquinaria. Luego, hay una o varias capas ocultas que procesan la información recibida de las capas anteriores utilizando activaciones sigmoid. Estas activaciones sigmoid son funciones no lineales que permiten a la red neuronal aprender patrones complejos en los datos.

Algorithm 1 Red neuronal no lineal autorregresiva

1: Inicializar número de vehículos M

2: Inicializar pesos de la red neuronal

3: **for** maquinaria $\leftarrow 1$ to M **do**

4: Inicializar entradas actuales

5: Inicializar entradas anteriores

6: Dividir los datos en conjuntos de entrenamiento, validación y test.

7: Crear capas ocultas con hiperparametros variables.

8: Añadir no linearidad con la funcion de activación sigmoide en las capas ocultas

9: $t = t + 1$ ▷ MSE loss

10: **end for**

11: **return** predicciones

La capa de salida de la red neuronal NAR utiliza activación ReLU. Esta capa tiene neuronas que producen una salida numérica, que representa la predicción del consumo de la maquinaria en el siguiente periodo. La activación ReLU es otra función no lineal que permite a la red neuronal aprender patrones complejos y producir una salida numérica precisa. La red neuronal aprende con una retropropagación de gradiente conjugado escalado. La red neuronal NAR se entrena utilizando un conjunto de datos de entrenamiento que consiste en los datos de entrada anteriores y los consumos correspondientes de la maquinaria del GAD La Maná. El objetivo del entrenamiento es minimizar el error entre las predicciones de la red y los valores reales de los consumos.

Una vez que la red neuronal NAR está entrenada, se puede utilizar para hacer predicciones en tiempo real del consumo futuro de la maquinaria. La red neuronal

NAR es capaz de aprender y adaptarse a los cambios en los datos de entrada y de salida, lo que permite una mayor precisión en las predicciones.

En resumen, la red neuronal NAR con activaciones sigmoid en sus capas ocultas y activación ReLU en su capa de salida es un modelo de aprendizaje profundo que puede utilizarse para predecir los consumos de una maquinaria de transporte pesada en función de los datos de entrada actuales y anteriores. Esto permite optimizar el uso de la maquinaria y reducir los costos operativos.

3.4.5 Modelo matemático

El modelo matemático es una red neuronal recurrente dinámica que incluye varias capas con conexiones de retroalimentación, según Hayken [8]. Anteriormente, muchos investigadores lo han utilizado para modelar procesos no lineales, como Coruh et al. [4], que aplicaron la red para predecir la eficiencia de adsorción en porcentaje para la eliminación de iones de zinc de aguas residuales. Lin et al. [14] afirmaron que la red es una herramienta poderosa para modelar y validar, con una convergencia mucho más rápida y una generalización mucho mejor que otros modelos de redes neuronales.

La expresión matemática para la salida de la red en el proceso de entrenamiento se puede representar mediante la siguiente ecuación:

$$y(t) = f\left(x(t-1), x(t-2) \ldots x(t-D_x), y(t-1), y(t-2) \ldots y(t-D_y)\right), \quad (3.1)$$

Donde f es una función no lineal aproximada por un Perceptrón Multicapa, $x(t)$ y $y(t)$ son la entrada y la salida del modelo en el paso de tiempo t, respectivamente,

mientras que D_x y D_y son el orden de la entrada y la salida. En nuestro caso, la entrada $x(t)$ corresponde al kilometraje de cada maquinaria. La salida predicha por la red, $y(t)$, se compara con los valores objetivo de entrada y representa los galones consumidos en un período futuro.

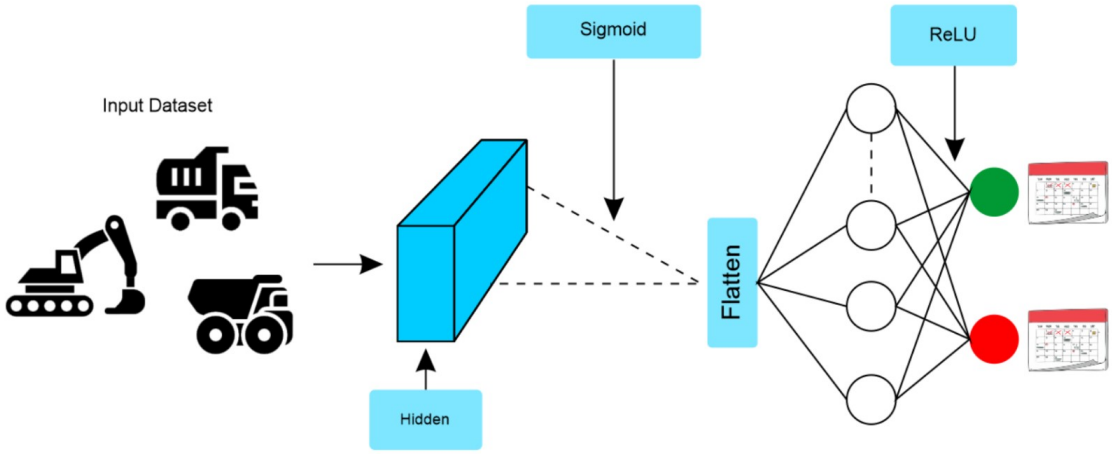


Figure 3.3: Diagrama de la red neuronal implementada en el estudio.

3.4.6 Evaluación del Modelo

En este apartado, se analizarán los resultados de la red neuronal propuesta bajo diferentes arquitecturas de la red neuronal y los diferentes algoritmos de entrenamiento. Entre los parámetros de la arquitectura bajo estudio se encuentran el número de neuronas ocultas y el número de retardos a considerar en el modelo. Mientras que los algoritmos de entrenamiento abordados en este estudio son: Levenberg-Marquardt, y el Gradiente Conjugado Escalado.

Se evaluó la capacidad predictiva utilizando el conjunto de prueba o validación. Para hacer esto, se introduce cada valor de kilometraje y odómetro del conjunto de prueba o validación en la red neuronal y se comparan las predicciones de consumo con los valores reales de consumo. Se utilizó el error medio cuadrático (MSE) para

evaluar la precisión de las predicciones de la red neuronal.

Después de evaluar la capacidad predictiva de la red neuronal, se pueden ajustar los parámetros de la red para mejorar su capacidad predictiva. Este proceso se llama validación cruzada y se realiza mediante la iteración de la división de los datos en conjuntos de entrenamiento y prueba o validación y el entrenamiento y evaluación de la red neuronal en cada iteración. Es importante tener en cuenta que el ajuste de los parámetros de la red neuronal debe realizarse con cuidado para evitar el sobreajuste.

Finalmente, al realizar el análisis comparativo acerca de las diferentes arquitecturas y algoritmos de entrenamiento utilizados, se escoge el modelo que permita obtener mayores resultados en cuanto a la predicción de los datos.

3.4.7 Estimación del consumo de lubricantes

Una vez que se tiene las predicciones de kilometraje y el consumo de aceite durante un año a través de la red neuronal implementada, es necesario estimar el consumo de lubricantes a partir de las recomendaciones del fabricante y la experiencia propia, para establecer cada que funcionamiento de la maquinaria se requiere el cambio de lubricantes.

Capítulo IV

MARCO EXPERIMENTAL Y DISCUSIÓN

Este capitulo presenta los resultados experimentales obtenidos con nuestra metodologia. Los resultados incluyen, el analisis en términos de la funcion de perdida de la red neuronal, calculos de error, tuneo de hiperparametros (hyperparameter tuning en íngles). Se incluye variaciones en las capas intermedias de la red, y su funcion de activacion. Se han experimentado variacion escenarios de predicción durante el tiempo, y ademas la prediccion de consumo de aceites dependiendo el desgaste del motor en cada cambio de aceite.

4.1 Hiperparametros de la red neuronal

Describimos los hiperparámetros de la red neuronal en la Tabla 4.1. Esta tabla muestra los diferentes modelos que se han evaluado durante una búsqueda de hiperparámetros, junto con los valores específicos de los hiperparámetros utilizados en cada modelo y sus métricas de rendimiento correspondientes. La tabla tiene columnas que representan los diferentes hiperparámetros y las métricas de rendimiento, y cada fila representa un modelo diferente. Se ha elegido el primer modelo, como el mejor de todos ya que tiene mejores métricas comparados con otros. Todos se han comparado usando la retropropagración Levenberg-Marquardt.

Neuronas en las capas ocultas	learning rate (lr)	Perdida en conjunto de validación (MSE)
128, 64	0.01	0.000301
96, 48	0.01	0.000400
96, 96	0.01	0.000306
64, 16	0.01	0.000410
32, 80	0.01	0.000302
256, 32	0.01	0.000320
32, 16	0.0001	0.000301
96, 96	0.0001	0.000311
96, 128	0.0001	0.000370
128, 96	0.0001	0.000378

Table 4.1: Las 10 mejores combinaciones de hiperparametros

4.2 Funcion de perdida

La Figura 4.1 muestra la pérdida (loss) de la red neuronal a lo largo del entrenamiento. Configuramos la tasa de aprendizaje (lr) en 0.01. El eje x representa las épocas de entrenamiento, mientras que el eje y representa el valor de la pérdida. La pérdida comienza alta al inicio del entrenamiento, con un valor relativo de 0.8, lo que indica que la red comienza aprender de los datos. Sin embargo, a medida que avanza el entrenamiento, la pérdida disminuye gradualmente, lo que indica que la red está aprendiendo de una mejor manera. Eventualmente, la pérdida converge a un valor bajo de 0 luego de 60 épocas, lo que indica que la red ha aprendido a predecir con precisión los valores óptimos. En general, la figura demuestra que la red está aprendiendo eficazmente de sus datos y mejorando su rendimiento con el tiempo.

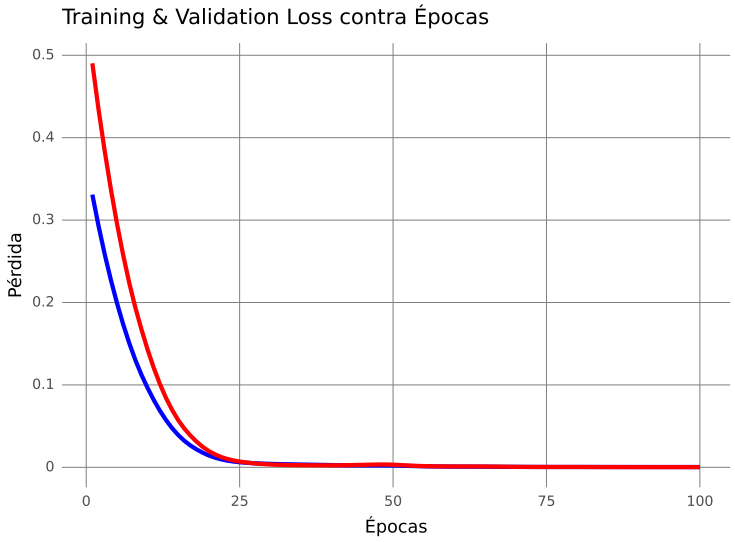


Figure 4.1: Pérdida de entrenamiento de la red neuronal durante 100 épocas.

4.3 Predición consumo de aceite de un vehiculo

Para calcular el consumo de lubricantes de la maquinaria analizada, a partir del kilometraje y horómetro estimado por medio de la red neuronal, se procedió a establecer de manera mensual los cambios de aceite necesarios para cada una de la maquinaria enlistada de acuerdo a las recomendaciones del fabricante.

La Figura 4.2 ilustra la predicción del consumo de aceite en galones de una camioneta. Inicialmente, el consumo es aproximadamente de 1.25 galones, pero a medida que aumenta la cantidad de kilómetros recorridos, también aumenta el consumo de galones. Los puntos rojos representan las mediciones aproximadas con ruido gaussiano, mientras que la línea verde representa el conjunto de validación.

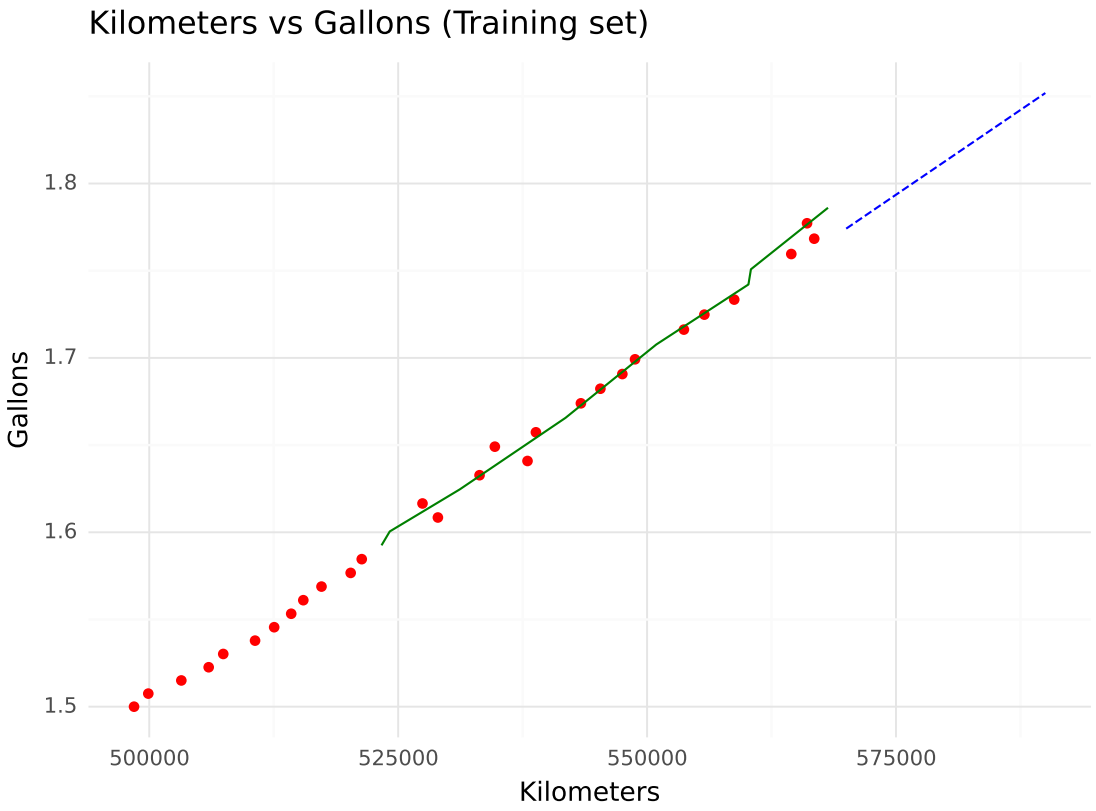


Figure 4.2: Predicción del consumo de aceite

4.3.1 Predicción por tipo de vehiculo

Para mejorar la predicción general por tipo de vehículo, hemos decidido agrupar los vehículos en cuatro categorías distintas en función del desgaste del motor y el tipo de uso. Para cada una de estas categorías, utilizaremos la misma red neuronal para realizar las predicciones. Los grupos se definen de la siguiente manera:

Tipo 1, Figura 4.3 (volquetes): Este grupo incluirá vehículos de carga pesada, como volquetes de gran tonelaje utilizados en proyectos de construcción y transporte de materiales pesados. Estas maquinarias usan aproximadamente 4 galones. Conforme los cambios de aceite de acuerdo el fabricamente y la antiguedad de la maquinaria, se ha realizado la predicció del consumo de aceite, la cual nos da aproxi-

madamente un aumento de uso en un 0.15 galones despues de 25000 galones.

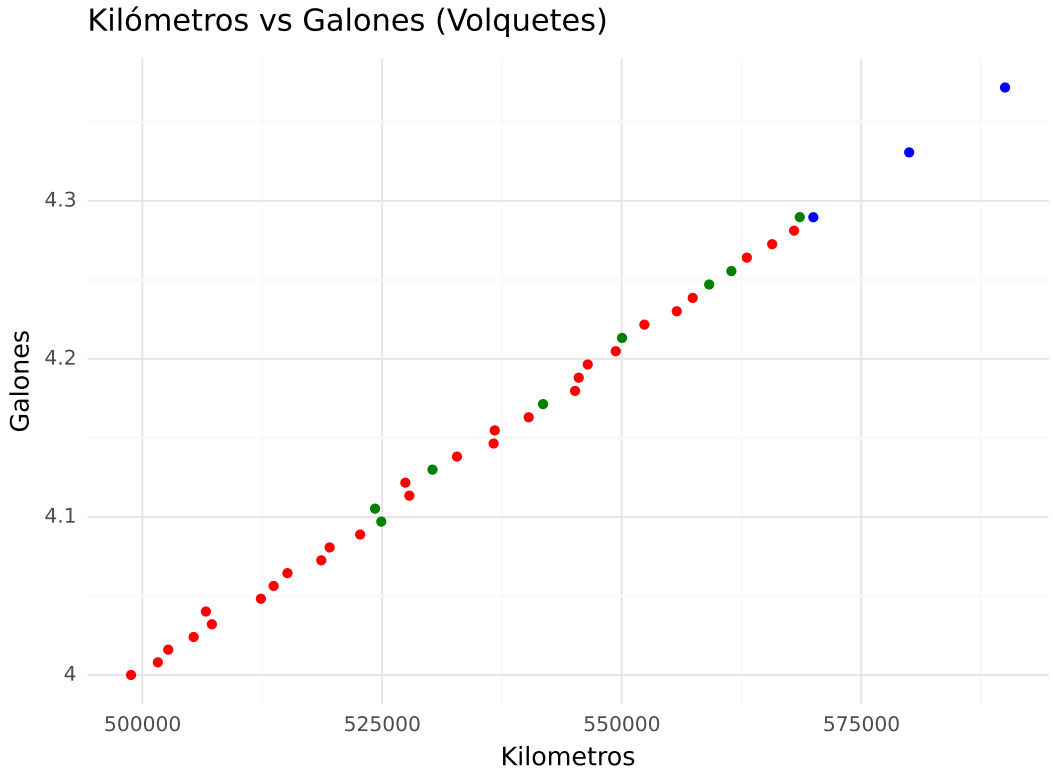


Figure 4.3: Tipo 1 (Volquetes)

Tipo 2, Figura 4.4 (camiones): En esta categoría, se incluyen camiones de carga de mediano tonelaje, los cuales son utilizados para el transporte de mercancías y suministros en distancias intermedias. Se ha observado que algunos de estos camiones presentan un alto consumo de aceite, lo que requiere que se realice un cambio periódico de aceite cada 2500 km.

La figura muestra un análisis de las predicciones realizadas para estos camiones, y se puede apreciar que las predicciones siguen una tendencia lineal con respecto a los datos de entrenamiento. Esto sugiere que existe una relación directa y predecible entre el consumo de aceite y la distancia recorrida en estos camiones.

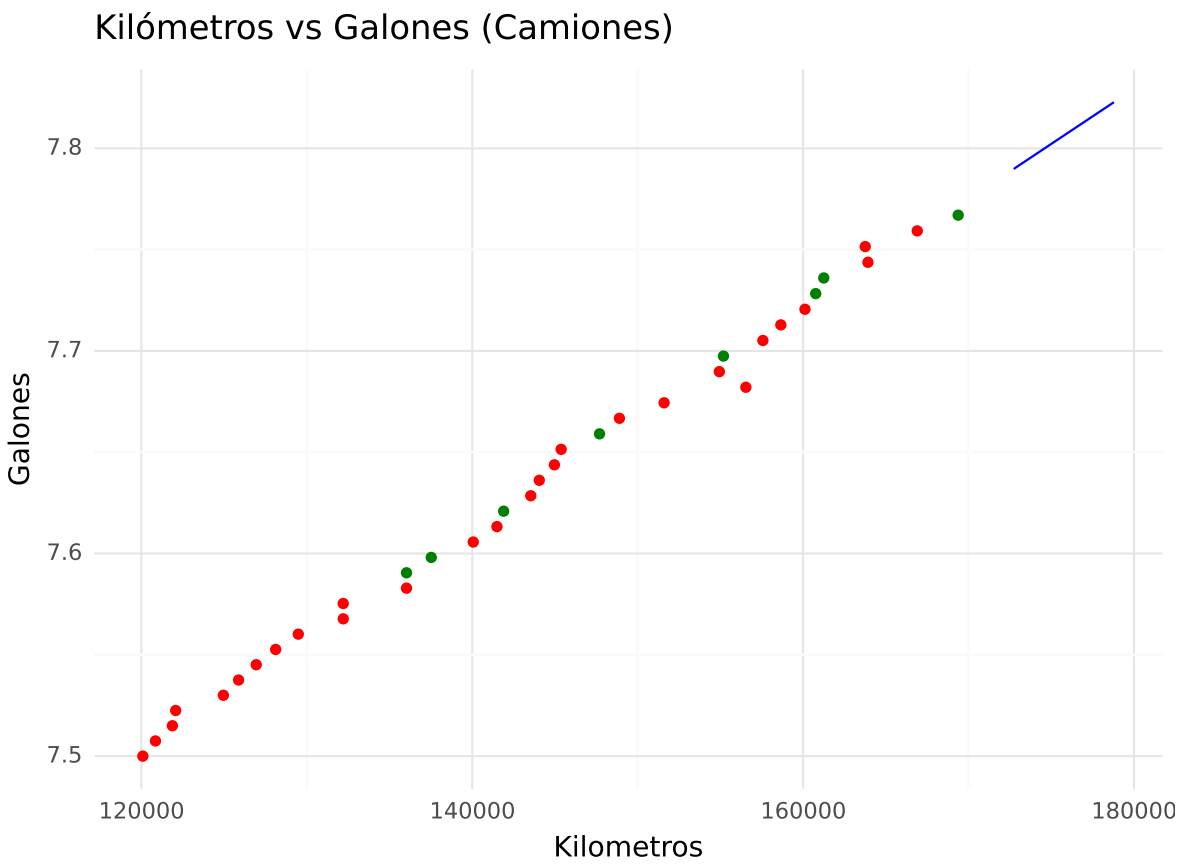


Figure 4.4: Tipo 2 (Camiones)

Tipo 3 (Camionetas), Figura 4.5 : Esta categoría comprenderá vehículos utilitarios ligeros, como camionetas y furgonetas, ideales para el transporte de mercancías más pequeñas y para uso comercial o personal.

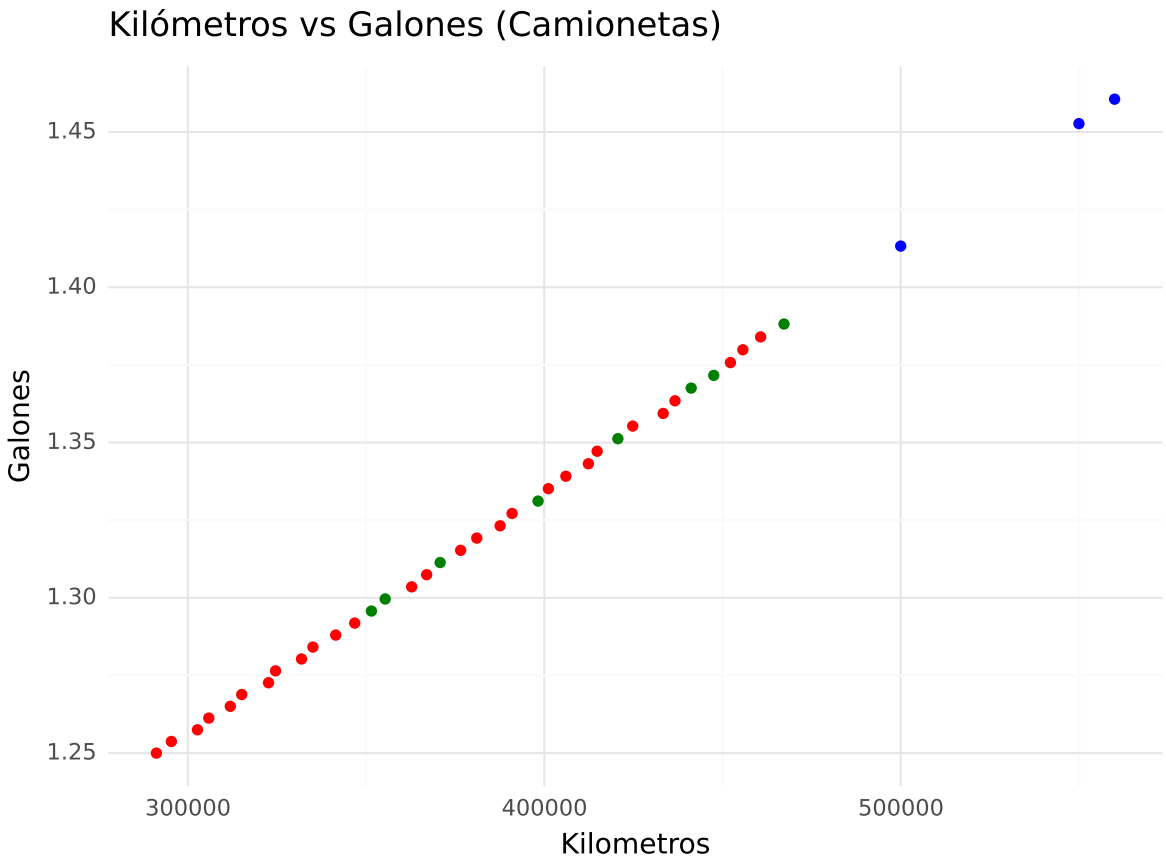


Figure 4.5: Tipo 3 (Camionetas)

Tipo 4 (Jeep), Figura 4.6 : En este grupo incluiremos vehículos todo terreno, como Jeeps y SUV, diseñados para la conducción fuera de carretera y condiciones adversas.

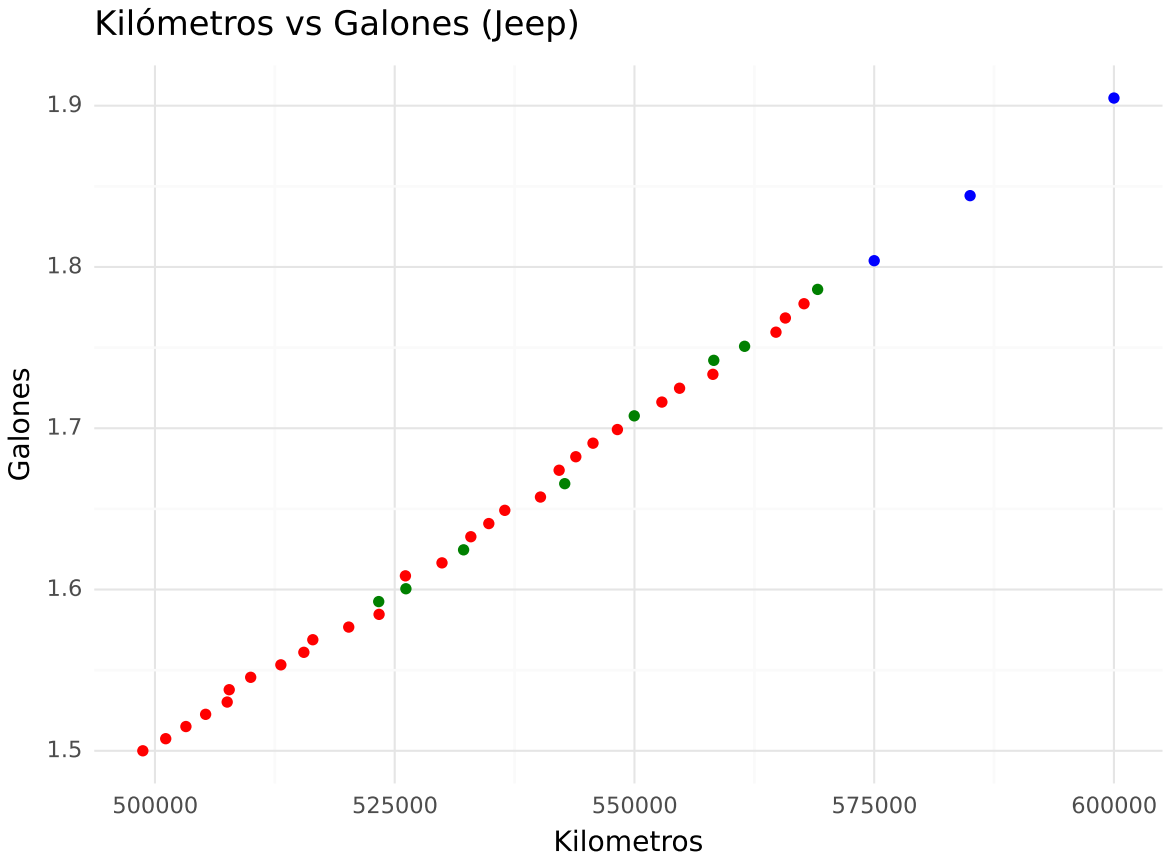


Figure 4.6: Tipo 4 (Jeep)

Al utilizar esta clasificación, podremos ajustar la red neuronal para cada tipo de vehículo específico, lo que nos permitirá mejorar las predicciones y obtener un mejor rendimiento en general para nuestro modelo.

4.4 Métricas de desempeño

La Figura 4.7 muestra las predicciones de galones para los datos de futuros kilómetros, junto con el rango de incertidumbre. En el eje x se encuentran los ”Kilómetros Fu-

turos” y en el eje y se representa la cantidad de galones predicha.

Los puntos azules representan las predicciones de galones para los diferentes puntos de ”Kilómetros Futuros”, basados en el modelo entrenado.

El área gris que rodea a los puntos azules representa el rango de incertidumbre. Este rango se calcula utilizando la desviación estándar de las predicciones de galones para los ”Kilómetros Futuros”. Cuanto más amplia es el área gris, mayor es la incertidumbre en las predicciones, y cuanto más estrecha es, mayor es la confianza en las predicciones.

Es importante considerar la incertidumbre al tomar decisiones basadas en las predicciones del modelo, especialmente cuando las predicciones son cruciales o sensibles. La incertidumbre nos permite comprender la variabilidad y la confianza que podemos tener en los resultados del modelo.

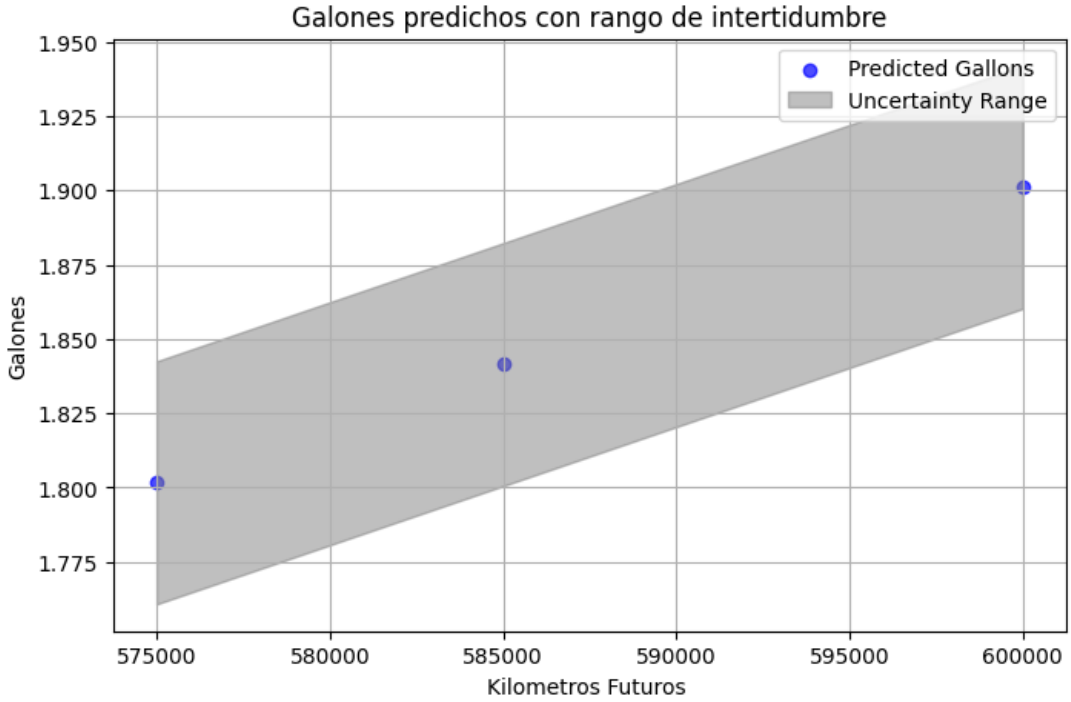


Figure 4.7: Galones predichos con rango de incertidumbre

Capítulo V

5.1 Conclusiones

En la actualidad, alrededor del mundo se ha desplegado grandes esfuerzos en el desarrollo de nuevas tecnologías con el afán de brindar mayor comodidad a la comunidad en general. Es así, que el campo de la IA durante los últimos años ha experimentado un desarrollo bastante grande en las diferentes industrias, en este caso, el mantenimiento preventivo apoyado en técnicas de inteligencia artificial se ha convertido en una herramienta valiosa en la labor de mantener operativa las máquinas. El cambio de lubricantes en los vehículos y maquinaria constituye una labor fundamental en el objetivo de preservarlas en perfectas condiciones. En el caso del sector público, la planificación del consumo de lubricantes juega un rol fundamental, puesto que permite realizar los procedimientos de contratación respectivos de manera oportuna y eficiente.

La investigación de los diferentes tipos de algoritmos basados en inteligencia artificial es un paso fundamental para identificar la mejor aproximación para el control del consumo de filtros y lubricantes. Esta etapa permitirá seleccionar el algoritmo más adecuado en función de las características de los datos y los requisitos del proyecto.

En este estudio se ha desarrollado un algoritmo basado en redes neuronales no lineales autorregresivas (NARs), las mismas que han permitido predecir el kilometraje y su previo consumo de lubricantes durante un año fiscal, y finalmente estimar el consumo de lubricantes a partir de los datos.

La NAR es una herramienta efectiva para predecir el kilometraje y horas de fun-

cionamiento de la maquinaria del GAD Municipal de La Maná, misma que ha sido desarrollada para diferentes maquinarias de las cuales se disponía de datos históricos recogidos por los funcionarios de la entidad. La NAR puede convertirse en una ayuda valiosa para el mantenimiento predictivo y la gestión del inventario de lubricantes, al predecir el consumo futuro de lubricantes en función del kilometraje y las horas de funcionamiento del equipo caminero, se pueden planificar con anticipación los servicios de mantenimiento y el abastecimiento de lubricantes para optimizar el rendimiento del equipo y reducir los costos de mantenimiento.

Al analizar los resultados de los modelos con el tuneo de hiperparámetros, se pudo establecer que la arquitectura con 128, 64 neuronas en cada capa oculta en las capas ocultas ofrece un mejor rendimiento en comparación a las demás analizada, ya que su valor de MSE es de 0.000301. Asimismo, el algoritmo de entrenamiento que mostró mejores resultados es el de Retropropagación de gradiente conjugado escalado. El consumo de lubricantes puede ser estimado a partir de los kilometrajes estimados mediante la predicción realizada por la red neuronal, es importante tener en cuenta que el consumo de lubricantes esta relacionado con el cambio de filtros, el cual es proporcional a los galones usados y kilometros sobre la frecuencia por año. Estas predicciones servirán de gran ayuda en el afán de establecer procesos en el GAD Municipal de La Maná que permitan garantizar la operatividad de la maquinaria. Esto debido a que, en los procedimientos de contratación pública es necesario contar con una planificación adecuada que permita contar con un contrato vigente de cambio de lubricantes para todo el año fiscal.

La recopilación de datos mediante métodos estadísticos es esencial para la con-

strucción del conjunto de datos (dataset) necesario para entrenar y evaluar el modelo de inteligencia artificial. La calidad y cantidad de los datos recopilados desempeñarán un papel crucial en la precisión del modelo final. El desarrollo de un algoritmo basado en inteligencia artificial específicamente diseñado para controlar el consumo de filtros y lubricantes es un logro significativo. Este algoritmo debe ser capaz de analizar datos históricos y tomar decisiones en tiempo real para optimizar el consumo y el mantenimiento de los equipos. La implementación de técnicas estadísticas para evaluar la precisión del modelo es esencial para garantizar su eficacia en la práctica. Estas técnicas permiten medir la capacidad del algoritmo para predecir el consumo de filtros y lubricantes de manera confiable, lo que es fundamental para la toma de decisiones informadas en la gestión de recursos.

Finalmente, como trabajo futuro se podría considerar el establecimiento de un plan de mantenimiento a partir de la predicción del kilometraje y horómetro de la maquinaria, con lo cual las paradas de la misma se realizarán de manera más controlada y eficaz.

5.1.1 Recomendaciones

La precisión de la predicción depende de la calidad de los datos de entrada y la cantidad de datos de entrenamiento. Cuanto más datos de alta calidad se utilicen para entrenar el modelo, más precisas serán las predicciones de la red neuronal. Es importante evaluar la calidad de las predicciones de la red neuronal con datos nuevos y verificar que las predicciones sean precisas y útiles en la práctica. Si las predicciones de la red neuronal no son precisas, se pueden considerar otras técnicas de aprendizaje

automático o métodos de modelado para mejorar la precisión de las predicciones.

Bibliografía

1. Achmad Tjachja Nugraha et al. “The Role Of Infrastructure In Economic Gro\Tth And Income Inequality In Indonesia”. In: *Economics & Sociology* 13.1 (2020), pp. 102–115.

2. C P Ng et al. “Road infrastructure development and economic growth”. In: *IOP Conference Series: Materials Science and Engineering* 512.1 (Apr. 2019), p. 012045. DOI: 10.1088/1757-899X/512/1/012045. URL: https://dx.doi.org/10.1088/1757-899X/512/1/012045.

3. Chao Wang et al. “Railway and road infrastructure in the Belt and Road Initiative countries: Estimating the impact of transport infrastructure on economic growth”. In: *Transportation Research Part A: Policy and Practice* 134 (2020), pp. 288–307. ISSN: 0965-8564. DOI: https://doi.org/10.1016/j.tra.2020.02.009. URL: https://www.sciencedirect.com/science/article/pii/S0965856419307001.

4. Isidro Ortega, Gerardo Angeles Castro, and David Carrillo-Murillo. “Infraestructura carretera y crecimiento económico en México”. In: *Problemas del Desarrollo. Revista Latinoamericana de Economía* 50 (May 2019). DOI: 10.22201/iiec.20078951e.2019.198.66383.

5. Shamsa Kanwal et al. “Road and transport infrastructure development and community support for tourism: The role of perceived benefits, and community satisfaction”. In: *Tourism Management* 77 (2020), p. 104014. ISSN: 0261-5177. DOI: https://doi

.org/10.1016/j.tourman.2019.104014. URL: https://www.sciencedirect.com/science/article/pii/S0261517719302122.

6. Ramadan Mazrekaj. “Impact of road infrastructure on tourism development in Kosovo”. In: *International Journal of Management* 11.4 (2020).

7. Jaime Bojorque Iñeguez and Cristina Chuquiguanga-Auquilla. “Efecto de la infraestructura pública en el precio del suelo urbano. Caso de la ciudad de Cuenca-Ecuador”. In: *Revista Urbano* 24 (May 2021), pp. 74–83. DOI: 10.22320/07183607.2021.24.43.07.

8. Jeannette Lardé. “Latin America’s infrastructure investment situation and challenges”. In: (2016).

9. *Multidimensional progress: well-being beyond income — socialprotection.org — socialprotection.org*. https://socialprotection.org/es/discover/publications/multidimensional-progress-well-being-beyond-income. [Accessed 31-07-2023].

10. Eamon McDermot et al. “Improving performance of infrastructure projects in developing countries: An Ecuadorian case study”. In: *International Journal of Construction Management* 22.13 (2022), pp. 2469–2483.

11. *Plan de CreacixF3;n de Oportunidades 2021-2025 de Ecuador — Observatorio Regional de PlanificacixF3;n para el Desarrollo — observatorioplanificacion.cepal.org*. https://observatorioplanificacion.cepal.org/es/planes/p

lan-de-creacion-de-oportunidades-2021-2025-de-ecuador. [Accessed 31-07-2023].

12. *Plan de Desarrollo y Ordenamiento Territorial del Cantón La Mana — lamana.gob.ec*. https://lamana.gob.ec/download/plan-de-desarrollo-y-ordenamiento-territorial-del-canton-la-mana/. [Accessed 31-07-2023].

13. RJ Hudachek and VR Dodd. "Progress and Payout of a Machinery Surveillance and Diagnostic Program". In: *MECHANICAL ENGINEERING*. Vol. 99. 1. ASME-AMER SOC MECHANICAL ENG 345 E 47TH ST, NEW YORK, NY 10017. 1977, pp. 106–106.

14. Małgorzata Jasiulewicz-Kaczmarek, Stanislaw Legutko, and Piotr Kluk. "Maintenance 4.0 technologies–new opportunities for sustainability driven maintenance". In: *Management and production engineering review* 11 (2020).

15. Andreas Theissler et al. "Predictive maintenance enabled by machine learning: Use cases and challenges in the automotive industry". In: *Reliability engineering & system safety* 215 (2021), p. 107864.

16. Chong Chen et al. "Automobile maintenance prediction using deep learning with GIS data". In: *Procedia CIRP* 81 (2019), pp. 447–452.

17. Alberto Jiménez Cortadi. "Aplicacion de modelos matematicos para el mantenimiento predictivo". In: (2020).

18. Thyago P Carvalho et al. “A systematic literature review of machine learning methods applied to predictive maintenance”. In: *Computers & Industrial Engineering* 137 (2019), p. 106024.

19. Matthias Faes and David Moens. “Cross-dependence between interval fields in Finite Element models: definition and analysis”. In: *Proceedings of ESREL*. 2019.

20. Gavril Grebenişan, Nazzal Salem, and Sanda Bogdan. “The lubricants’ parameters monitoring and data collecting”. In: *MATEC Web of Conferences*. Vol. 184. EDP Sciences. 2018, p. 03008.

21. Muskan Jain et al. “Systematic literature review on predictive maintenance of vehicles and diagnosis of vehicle’s health using machine learning techniques”. In: *Computational Intelligence* 38.6 (2022), pp. 1990–2008.

22. Andrew KS Jardine and Albert HC Tsang. *Maintenance, replacement, and reliability: theory and applications*. CRC press, 2021.

23. Carlos Alberto Montilla Montaña. *Fundamentos de mantenimiento industrial*. Universidad Tecnológica de Pereira, 2016.

24. S. García Garrido. *Manual práctico para la gestión eficaz del mantenimiento industrial*. RENOVETEC, 2012.

25. J.D. Navarro and Calpe Institute of Tecnology. *Técnicas de mantenimiento industrial*. Manuales Calpe de ingenierıa civil, industrial y militar. Calpe Institute of Technology, 2007. ISBN: 9788461162437. URL: https://books.google.co.ve/books?id=5vE6QwAACAAJ.

26. Pedro Ponce. *Inteligencia artificial: con aplicaciones a la ingenieria*. Alpha Editorial, 2010.

27. Caiming Zhang and Yang Lu. "Study on artificial intelligence: The state of the art and future prospects". In: *Journal of Industrial Information Integration* 23 (2021), p. 100224.

28. Jay Lee. *Industrial AI: Applications with Sustainable Performance*. Jan. 2020. ISBN: 978-981-15-2143-0. DOI: 10.1007/978-981-15-2144-7.

29. Andres Quelal et al. "Identifying the Political Tendency of Social Bots in Twitter Using Sentiment Analysis: A Use Case of the 2021 Ecuadorian General Elections". In: *Doctoral Symposium on Information and Communication Technologies*. Springer. 2022, pp. 184–196.

30. B Aruna Devi and M Pallikonda Rajasekaran. "Performance evaluation of MRI pancreas image classification using artificial neural network (ANN)". In: *Smart Intelligent Computing and Applications: Proceedings of the Second International Conference on SCI 2018, Volume 1*. Springer. 2019, pp. 671–681.

31. Peter Norvig Russell. "Artificial intelligence: a modern approach by stuart". In: *Russell and Peter Norvig contributing writers, Ernest Davis...[et al.]* (2010).

32. Ibrahim M Nasser and Samy S Abu-Naser. "Predicting tumor category using artificial neural networks". In: (2019).

33. Eduardo Francisco Caicedo Bravo and Jesús Alfonso López Sotelo. "Una aproximación práctica a las redes neuronales artificiales." In: (2017).

34. Ke-Lin Du and Madisetti NS Swamy. *Neural networks and statistical learning*. Springer Science & Business Media, 2013.

35. PEDRO Isasi Vinuela and IM Galván León. "Redes de neuronas artificiales". In: *Un Enfoque Práctico, Editorial Pearson Educación SA Madrid España* (2004).

36. Fabio Henrique Pereira et al. "Nonlinear autoregressive neural network models for prediction of transformer oil-dissolved gas concentrations". In: *Energies* 11.7 (2018), p. 1691.

37. Alexei Botchkarev. "Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology". In: *arXiv preprint arXiv:1809.03006* (2018).

38. Ricardo Borja-Robalino, Antonio Monleon-Getino, and José Rodellar. "Estandarización de métricas de rendimiento para clasificadores Machine y Deep Learning". In: *Revista Ibérica de Sistemas e Tecnologias de Informação* E30 (2020), pp. 184–196.

39. Guy S Handelman et al. "Peering into the black box of artificial intelligence: Evaluation metrics of machine learning methods." In: *AJR. American journal of roentgenology* 212.1 (2018), pp. 38–43.

40. *La Maná - Datos Generales*. `https://lamana.gob.ec/datos-generales/`. Sitio web oficial de La Maná. Sin fecha de actualización.

Appendices

prediction

August 3, 2023

0.0.1 Importar las librerias

[1]:
```
import math
import numpy as np
import pandas as pd
from pandas import read_csv
import matplotlib.pyplot as plt
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
import tensorflow as tf
from sklearn.model_selection import train_test_split
import plotnine as pn
```

0.0.2 Cargar los datos

[2]:
```
dataframe = read_csv('maquinaria.csv', engine='python',)
```

0.0.3 Visualizar los datos

[36]:
```
dataframe
```

[36]:

	n	maq	frecuency	previous	n1	n2	n3	\
0	1	CAMIONETA5_0321	5000	286142	291142	296142	301142	
1	2	JEEP4_0357	4000	495265	499265	501265	503265	
2	3	VOLQ4_0322	4000	251134	253134	255134	257134	
3	4	VOLQ3_SP	3000	186798	188298	189798	191298	
4	5	VOLQ4_1350	4000	183128	185128	187128	189128	
5	8	VOLQ5_1354	5000	95750	98250	100750	103250	
6	9	VOLQ5_1351	5000	97872	100372	102872	105372	
7	10	VOLQ5_1353	5000	83000	85500	88000	90500	
8	11	VOLQ5_1352	5000	94283	96783	99283	101783	
9	12	CAM5_0059	5000	223779	226279	228779	231279	
10	13	CAM5_1354	5000	285204	290204	295204	300204	
11	14	CAM5_1363	5000	239032	244032	249032	254032	

12	15	CAM25_SP	2500	116601	119101	120351	121601
13	16	CAM30_1373	3000	9530	12530	14030	15530
14	1	CAMIONETA5_0321	5000	346142	351142	356142	361142
15	2	JEEP4_0357	4000	521265	523265	525265	527265
16	3	VOLQ4_0322	4000	275134	277134	279134	281134
17	4	VOLQ3_SP	3000	204798	206298	207798	209298
18	5	VOLQ4_1350	4000	207128	209128	211128	213128
19	8	VOLQ5_1354	5000	125750	130750	133250	135750
20	9	VOLQ5_1351	5000	127872	132872	135372	137872
21	10	VOLQ5_1353	5000	113000	115500	118000	120500
22	11	VOLQ5_1352	5000	124283	126783	129283	131783
23	12	CAM5_0059	5000	253779	256279	258779	261279
24	13	CAM5_1354	5000	345204	350204	355204	360204
25	14	CAM5_1363	5000	299032	304032	309032	314032
26	15	CAM25_SP	2500	134101	136601	137851	139101
27	16	CAM30_1373	3000	30530	33530	35030	36530
28	1	CAMIONETA5_0321	5000	406142	411142	416142	421142
29	2	JEEP4_0357	4000	545265	547265	549265	551265
30	3	VOLQ4_0322	4000	299134	301134	303134	305134
31	4	VOLQ3_SP	3000	222798	224298	225798	227298
32	5	VOLQ4_1350	4000	231128	233128	235128	237128
33	8	VOLQ5_1354	5000	158250	163250	165750	168250
34	9	VOLQ5_1351	5000	160372	165372	167872	170372
35	10	VOLQ5_1353	5000	143000	145500	148000	150500
36	11	VOLQ5_1352	5000	154283	156783	159283	161783
37	12	CAM5_0059	5000	283779	286279	288779	291279
38	13	CAM5_1354	5000	405204	410204	415204	420204
39	14	CAM5_1363	5000	359032	364032	369032	374032
40	15	CAM25_SP	2500	151601	154101	155351	156601
41	16	CAM30_1373	3000	51530	54530	56030	57530

	n4	n5	n6	n7	n8	n9	n10	n11	n12	\
0	306142	311142	316142	321142	326142	331142	336142	341142	346142	
1	505265	507265	509265	511265	513265	515265	517265	519265	521265	
2	259134	261134	263134	265134	267134	269134	271134	273134	275134	
3	192798	194298	195798	197298	198798	200298	201798	203298	204798	
4	191128	193128	195128	197128	199128	201128	203128	205128	207128	
5	105750	108250	110750	113250	115750	118250	120750	123250	125750	
6	107872	110372	112872	115372	117872	120372	122872	125372	127872	
7	93000	95500	98000	100500	103000	105500	108000	110500	113000	
8	104283	106783	109283	111783	114283	116783	119283	121783	124283	
9	233779	236279	238779	241279	243779	246279	248779	251279	253779	
10	305204	310204	315204	320204	325204	330204	335204	340204	345204	
11	259032	264032	269032	274032	279032	284032	289032	294032	299032	
12	122851	124101	125351	126601	127851	129101	130351	131601	134101	
13	17030	18530	20030	21530	23030	24530	26030	27530	30530	
14	366142	371142	376142	381142	386142	391142	396142	401142	406142	

15	529265	531265	533265	535265	537265	539265	541265	543265	545265
16	283134	285134	287134	289134	291134	293134	295134	297134	299134
17	210798	212298	213798	215298	216798	218298	219798	221298	222798
18	215128	217128	219128	221128	223128	225128	227128	229128	231128
19	138250	140750	143250	145750	148250	150750	153250	155750	158250
20	140372	142872	145372	147872	150372	152872	155372	157872	160372
21	123000	125500	128000	130500	133000	135500	138000	140500	143000
22	134283	136783	139283	141783	144283	146783	149283	151783	154283
23	263779	266279	268779	271279	273779	276279	278779	281279	283779
24	365204	370204	375204	380204	385204	390204	395204	400204	405204
25	319032	324032	329032	334032	339032	344032	349032	354032	359032
26	140351	141601	142851	144101	145351	146601	147851	149101	151601
27	38030	39530	41030	42530	44030	45530	47030	48530	51530
28	426142	431142	436142	441142	446142	451142	456142	461142	466142
29	553265	555265	557265	559265	561265	563265	565265	567265	569265
30	307134	309134	311134	313134	315134	317134	319134	321134	323134
31	228798	230298	231798	233298	234798	236298	237798	239298	240798
32	239128	241128	243128	245128	247128	249128	251128	253128	255128
33	170750	173250	175750	178250	180750	183250	185750	188250	190750
34	172872	175372	177872	180372	182872	185372	187872	190372	192872
35	153000	155500	158000	160500	163000	165500	168000	170500	173000
36	164283	166783	169283	171783	174283	176783	179283	181783	184283
37	293779	296279	298779	301279	303779	306279	308779	311279	313779
38	425204	430204	435204	440204	445204	450204	455204	460204	465204
39	379032	384032	389032	394032	399032	404032	409032	414032	419032
40	157851	159101	160351	161601	162851	164101	165351	166601	169101
41	59030	60530	62030	63530	65030	66530	68030	69530	72530

	factor	galons
0	3	1.25
1	5	1.50
2	2	4.00
3	2	3.50
4	2	6.00
5	1	10.00
6	1	10.00
7	1	10.00
8	1	10.00
9	2	2.50
10	3	2.00
11	2	2.00
12	1	7.50
13	0	4.00
14	4	1.25
15	5	1.50
16	2	4.00
17	2	3.50

```
18        2     6.00
19        1    10.00
20        1    10.00
21        1    10.00
22        1    10.00
23        2     2.50
24        4     2.00
25        3     2.00
26        1     7.50
27        0     4.00
28        4     1.25
29        5     1.50
30        3     4.00
31        2     3.50
32        2     6.00
33        1    10.00
34        1    10.00
35        1    10.00
36        1    10.00
37        3     2.50
38        4     2.00
39        4     2.00
40        1     7.50
41        0     4.00
```

[38]:
```
dataframe[" maq"][0]
```

[38]:
```
'CAMIONETA5_0321'
```

[80]:
```
last_km = {}
for i in range(14):
    print(dataframe[" n12"][28 + i: 29 + i].values[0])
```

```
466142
569265
323134
240798
255128
190750
192872
173000
184283
313779
465204
419032
169101
72530
```

[4]:
```
kilometers_columns = dataframe.iloc[:, 4:16]
factor_exp = dataframe.iloc[:, 16:18]
#create an array with the values of the row
kilometers_values = kilometers_columns.values
```

Crear conjuntos X y Y

[5]:
```
kilometers_list = []

for i in range(0, 14):
    kilometers_list.append(np.concatenate((kilometers_values[i],␣
 ↪kilometers_values[i+14], kilometers_values[i+28])))

galones = []
for i in range(0, 14):
    array = factor_exp.iloc[i, 1] * (1 + (factor_exp.iloc[i, 0] / 1000)) ** np.
 ↪arange(36)
    galones.append(array)
```

[8]:
```
maq_number = 1
X = kilometers_list[maq_number] + np.random.normal(0, 1000, 36)
y = galones[maq_number]

scaler_X = MinMaxScaler()
scaler_y = MinMaxScaler()
X_normalized = scaler_X.fit_transform(X.reshape(-1, 1))
y_normalized = scaler_y.fit_transform(y.reshape(-1, 1))

# Splitting the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_normalized, y_normalized,␣
 ↪test_size=0.2, random_state=42)

# Neural network model
model = tf.keras.Sequential([
    tf.keras.layers.Dense(64, activation='relu', input_shape=(1,)),
    tf.keras.layers.Dense(32, activation='relu'),
    tf.keras.layers.Dense(1)
])

# Compile the model with mean squared error loss function
model.compile(optimizer='adam', loss='mean_squared_error')

history = model.fit(X_train, y_train, epochs=100, batch_size=32,␣
 ↪validation_data=(X_test, y_test), verbose=0)

# Access the loss values from the history object
train_loss = history.history['loss']
```

```
val_loss = history.history['val_loss']

df = pd.DataFrame({
    'Epochs': range(1, len(train_loss) + 1),
    'Training Loss': train_loss,
    'Validation Loss': val_loss
})
```

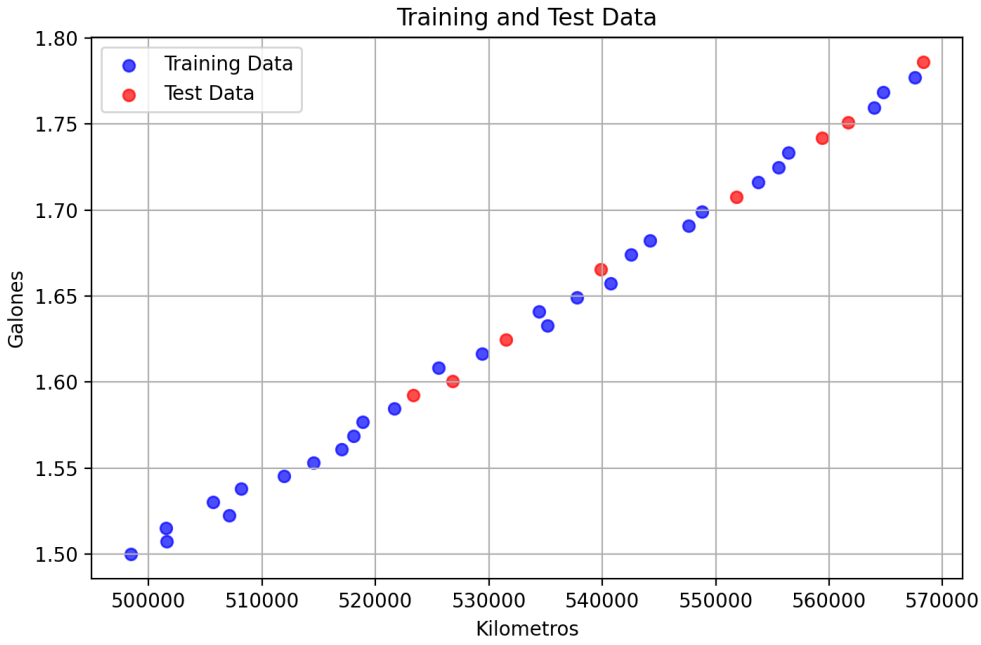
```
2023-08-03 00:56:29.185262: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.
2023-08-03 00:56:29.308823: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.
```

[9]:
```
# Plot the data using Plot Nine
plot = pn.ggplot(df, pn.aes(x='Epochs')) + \
       pn.geom_line(pn.aes(y='Training Loss'), color='blue', size=1.5) + \
       pn.geom_line(pn.aes(y='Validation Loss'), color='red', size=1.5) + \
       pn.xlab('Epochs') + \
       pn.ylab('Loss') + \
       pn.ggtitle('Training and Validation Loss over Epochs') + \
       pn.theme_minimal() + \
       pn.theme(panel_grid_major=pn.element_line(color='gray', size=0.5),
                panel_grid_minor=pn.element_blank(),
                legend_position='top')

print(plot)
#output_file = 'plot_nine_output.pdf'
#pn.ggsave(plot, output_file, dpi=300)
```

0.0.4 Testing data y entrenamiento

[11]:
```
# Invertir la normalizacion
X_train_inv = scaler_X.inverse_transform(X_train)
X_test_inv = scaler_X.inverse_transform(X_test)
y_train_inv = scaler_y.inverse_transform(y_train)
y_test_inv = scaler_y.inverse_transform(y_test)
```

[12]:
```
# Plot the training and test data
plt.figure(figsize=(8, 5))
plt.scatter(X_train_inv, y_train_inv, color='blue', label='Training Data',␣
 ↪alpha=0.7)
plt.scatter(X_test_inv, y_test_inv, color='red', label='Test Data', alpha=0.7)
plt.xlabel('Kilometros')
plt.ylabel('Galones')
plt.title('Training and Test Data')
plt.legend()
plt.grid(True)
plt.show()
```



0.0.5 Prediccion

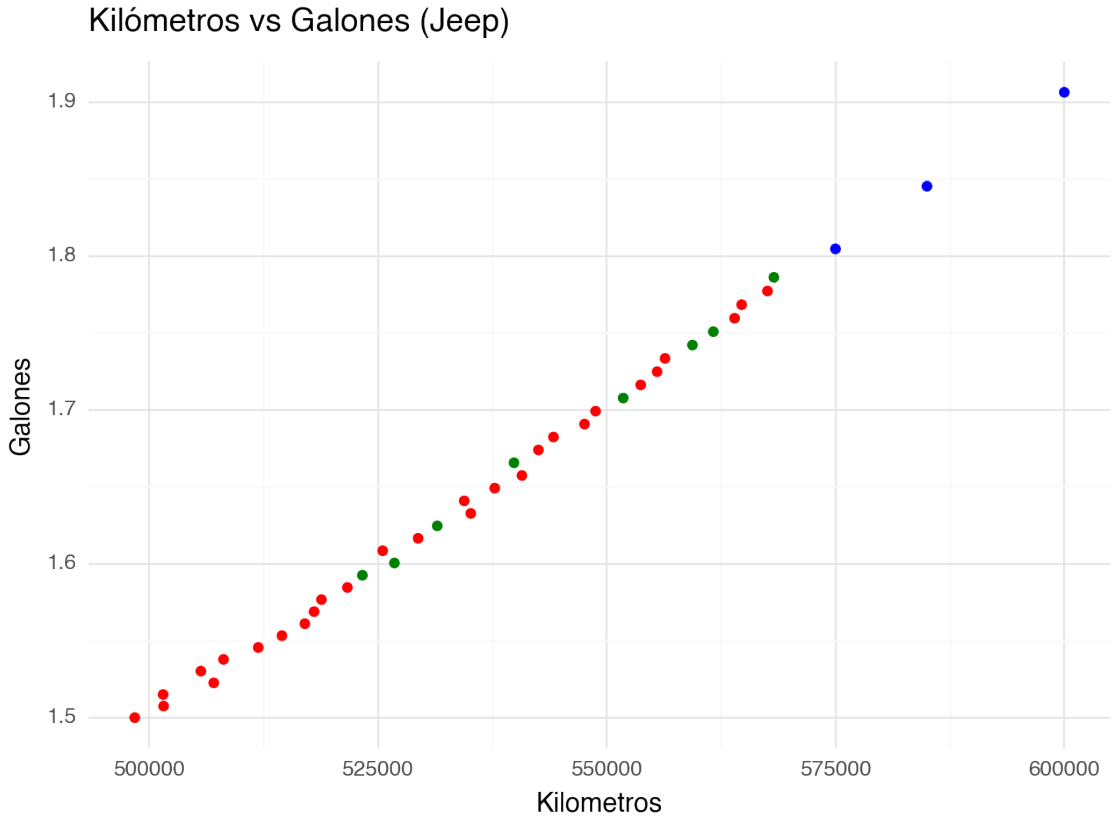
[13]:
```
#Input para la prediccion
future_kilometers = np.array([575000, 585000, 600000])
future_kilometers_normalized = scaler_X.transform(future_kilometers.reshape(-1,␣
↪1))

# Predecir los galones para los kilometros futuros
predicted_gallons_normalized = model.predict(future_kilometers_normalized)
predicted_gallons = scaler_y.inverse_transform(predicted_gallons_normalized)

print("Galones predichos por kilometros:", predicted_gallons)
```

```
1/1 [==============================] - 0s 13ms/step
Galones predichos por kilometros: [[1.8045846]
 [1.8453065]
 [1.9063892]]
```

[20]:
```
from plotnine import ggplot, aes, geom_point, labs, theme_minimal

# Create a scatter plot for training and test data
data_train = pd.DataFrame({'Kilometros': X_train_inv.ravel(), 'Galones':␣
↪y_train_inv.ravel()})
```

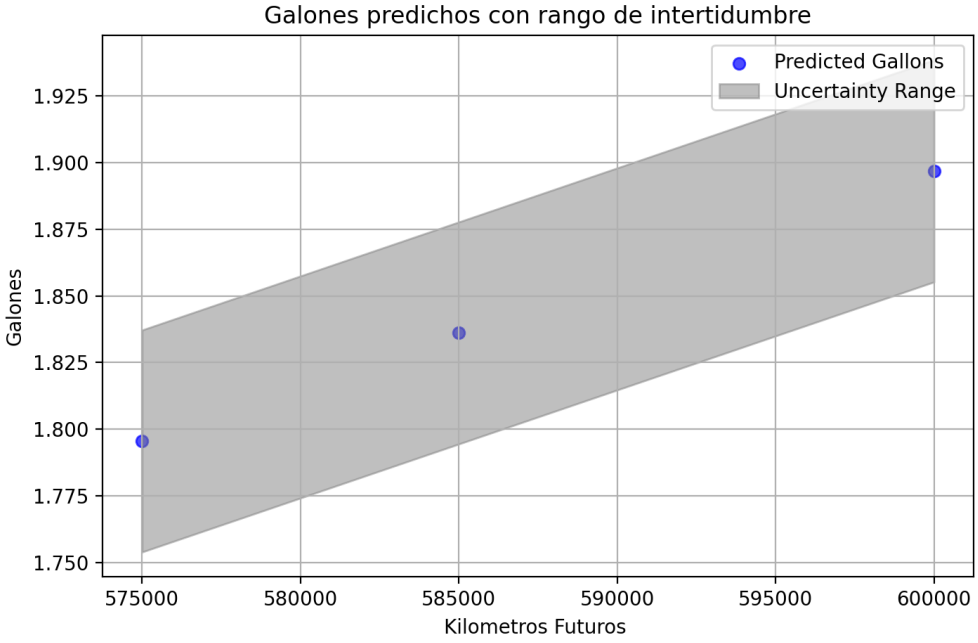
```
data_test = pd.DataFrame({'Kilometros': X_test_inv.ravel(), 'Galones':␣
 ↪y_test_inv.ravel()})
data_predicted = pd.DataFrame({'Kilometros': future_kilometers, 'Galones':␣
 ↪predicted_gallons.ravel()})
```

[21]:
```
p = (
    ggplot() +
    geom_point(data=data_train, mapping=aes(x='Kilometros', y='Galones'),␣
 ↪color='red') +
    geom_point(data=data_test, mapping=aes(x='Kilometros', y='Galones'),␣
 ↪color='green') +
    geom_point(data=data_predicted, mapping=aes(x='Kilometros', y='Galones'),␣
 ↪color='blue') +
    labs(title='Kilómetros vs Galones (Jeep)', x='Kilometros', y='Galones') +
    theme_minimal()
)


# Display the plot
print(p)

#output_file2 = 'jeep.pdf'
#ggsave(p, output_file2, dpi=300)
```



[22]:
```
future_gallons_predicted_normalized = model.
 ↪predict(predicted_gallons_normalized)
future_gallons_predicted = scaler_y.
 ↪inverse_transform(future_gallons_predicted_normalized)

predicted_gallons_std = np.std(future_gallons_predicted, axis=0)

# Calculate upper and lower bounds for uncertainty range
upper_bound = future_gallons_predicted.ravel() + predicted_gallons_std
lower_bound = future_gallons_predicted.ravel() - predicted_gallons_std

# Plot the uncertainty range
plt.figure(figsize=(8, 5))
plt.scatter(future_kilometers, future_gallons_predicted.ravel(), color='b',␣
 ↪label='Predicted Gallons', alpha=0.7)
plt.fill_between(future_kilometers, upper_bound, lower_bound, color='gray',␣
 ↪alpha=0.5, label='Uncertainty Range')
plt.xlabel('Kilometros Futuros ')
plt.ylabel('Galones')
plt.title('Galones predichos con rango de intertidumbre')
plt.legend()
plt.grid(True)
plt.show()
```

```
1/1 [==============================] - 0s 15ms/step
```



0.0.6 Hyperparameter tuning

[23]:
```
import numpy as np
import pandas as pd
import tensorflow as tf
from tensorflow.keras.layers import Dense
from tensorflow.keras.models import Sequential
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
import keras_tuner as kt
```

[46]:
```
# Concatenated data and target values
X = kilometers_list[maq_number] + np.random.normal(0, 1000, 36)
y = galones[maq_number]

scaler_X = MinMaxScaler()
scaler_y = MinMaxScaler()
X_normalized = scaler_X.fit_transform(X.reshape(-1, 1))
y_normalized = scaler_y.fit_transform(y.reshape(-1, 1))

# Splitting the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_normalized, y_normalized,␣
 ↪test_size=0.2, random_state=42)
```

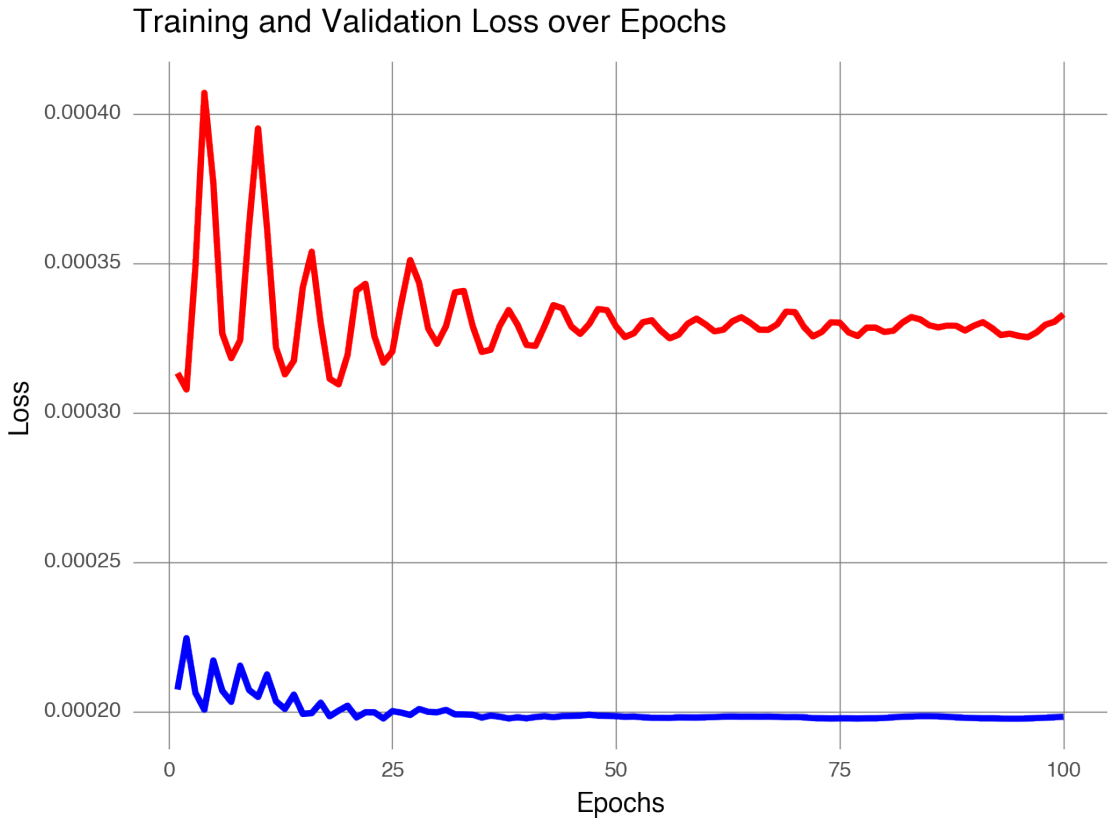
```
def build_model(hp):
    model = Sequential()
    model.add(Dense(units=hp.Int('units_1', min_value=32, max_value=256,␣
 ↪step=32), activation='relu', input_shape=(1,)))
    model.add(Dense(units=hp.Int('units_2', min_value=16, max_value=128,␣
 ↪step=16), activation='relu'))
    model.add(Dense(1))
    model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=hp.
 ↪Choice('learning_rate', values=[1e-2, 1e-3, 1e-4])),
                  loss='mean_squared_error')
    return model

tuner = kt.Hyperband(build_model,
                     objective='val_loss',
                     max_epochs=100,
                     factor=3,
                     directory='my_dir',
                     project_name='hyperparameter_search')

tuner.search(X_train, y_train, epochs=100, batch_size=32,␣
 ↪validation_data=(X_test, y_test), verbose=0)

best_model = tuner.get_best_models(num_models=1)[0]

# Train the best model with the optimal hyperparameters
best_model.fit(X_train, y_train, epochs=100, batch_size=32,␣
 ↪validation_data=(X_test, y_test))

# Access the loss values from the best model history object
train_loss = best_model.history.history['loss']
val_loss = best_model.history.history['val_loss']

df = pd.DataFrame({
    'Epochs': range(1, len(train_loss) + 1),
    'Training Loss': train_loss,
    'Validation Loss': val_loss
})

# Plot the data using Plot Nine
plot = pn.ggplot(df, pn.aes(x='Epochs')) + \
       pn.geom_line(pn.aes(y='Training Loss'), color='blue', size=1.5) + \
       pn.geom_line(pn.aes(y='Validation Loss'), color='red', size=1.5) + \
       pn.xlab('Epochs') + \
       pn.ylab('Loss') + \
       pn.ggtitle('Training and Validation Loss over Epochs') + \
       pn.theme_minimal() + \
       pn.theme(panel_grid_major=pn.element_line(color='gray', size=0.5),
```

```
                panel_grid_minor=pn.element_blank(),
                legend_position='top')

print(plot)

output_file = 'plot_nine_output.pdf'
pn.ggsave(plot, output_file, dpi=300)
```

```
WARNING:absl:At this time, the v2.11+ optimizer `tf.keras.optimizers.Adam` runs
slowly on M1/M2 Macs, please use the legacy Keras optimizer instead, located at
`tf.keras.optimizers.legacy.Adam`.
WARNING:absl:There is a known slowdown when using v2.11+ Keras optimizers on
M1/M2 Macs. Falling back to the legacy Keras optimizer, i.e.,
`tf.keras.optimizers.legacy.Adam`.
WARNING:absl:At this time, the v2.11+ optimizer `tf.keras.optimizers.Adam` runs
slowly on M1/M2 Macs, please use the legacy Keras optimizer instead, located at
`tf.keras.optimizers.legacy.Adam`.
WARNING:absl:There is a known slowdown when using v2.11+ Keras optimizers on
M1/M2 Macs. Falling back to the legacy Keras optimizer, i.e.,
`tf.keras.optimizers.legacy.Adam`.
2023-07-31 20:34:43.561627: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.
2023-07-31 20:34:43.712055: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.
WARNING:absl:At this time, the v2.11+ optimizer `tf.keras.optimizers.Adam` runs
slowly on M1/M2 Macs, please use the legacy Keras optimizer instead, located at
`tf.keras.optimizers.legacy.Adam`.
WARNING:absl:There is a known slowdown when using v2.11+ Keras optimizers on
M1/M2 Macs. Falling back to the legacy Keras optimizer, i.e.,
`tf.keras.optimizers.legacy.Adam`.
2023-07-31 20:34:44.215840: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.
2023-07-31 20:34:44.352352: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.
WARNING:absl:At this time, the v2.11+ optimizer `tf.keras.optimizers.Adam` runs
slowly on M1/M2 Macs, please use the legacy Keras optimizer instead, located at
`tf.keras.optimizers.legacy.Adam`.
WARNING:absl:There is a known slowdown when using v2.11+ Keras optimizers on
M1/M2 Macs. Falling back to the legacy Keras optimizer, i.e.,
`tf.keras.optimizers.legacy.Adam`.
2023-07-31 20:34:44.732062: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.
```

```
val_loss: 3.2743e-04
Epoch 82/100
1/1 [==============================] - 0s 20ms/step - loss: 1.9838e-04 -
val_loss: 3.3012e-04
Epoch 83/100
1/1 [==============================] - 0s 20ms/step - loss: 1.9846e-04 -
val_loss: 3.3199e-04
Epoch 84/100
1/1 [==============================] - 0s 22ms/step - loss: 1.9855e-04 -
val_loss: 3.3120e-04
Epoch 85/100
1/1 [==============================] - 0s 20ms/step - loss: 1.9856e-04 -
val_loss: 3.2924e-04
Epoch 86/100
1/1 [==============================] - 0s 19ms/step - loss: 1.9850e-04 -
val_loss: 3.2853e-04
Epoch 87/100
1/1 [==============================] - 0s 19ms/step - loss: 1.9836e-04 -
val_loss: 3.2912e-04
Epoch 88/100
1/1 [==============================] - 0s 19ms/step - loss: 1.9817e-04 -
val_loss: 3.2908e-04
Epoch 89/100
1/1 [==============================] - 0s 19ms/step - loss: 1.9803e-04 -
val_loss: 3.2747e-04
Epoch 90/100
1/1 [==============================] - 0s 19ms/step - loss: 1.9794e-04 -
val_loss: 3.2911e-04
Epoch 91/100
1/1 [==============================] - 0s 20ms/step - loss: 1.9786e-04 -
val_loss: 3.3028e-04
Epoch 92/100
1/1 [==============================] - 0s 18ms/step - loss: 1.9787e-04 -
val_loss: 3.2832e-04
Epoch 93/100
1/1 [==============================] - 0s 19ms/step - loss: 1.9778e-04 -
val_loss: 3.2598e-04
Epoch 94/100
1/1 [==============================] - 0s 18ms/step - loss: 1.9775e-04 -
val_loss: 3.2643e-04
Epoch 95/100
1/1 [==============================] - 0s 19ms/step - loss: 1.9775e-04 -
val_loss: 3.2568e-04
Epoch 96/100
1/1 [==============================] - 0s 18ms/step - loss: 1.9781e-04 -
val_loss: 3.2524e-04
Epoch 97/100
1/1 [==============================] - 0s 19ms/step - loss: 1.9794e-04 -
```

```
val_loss: 3.2687e-04
Epoch 98/100
1/1 [==============================] - 0s 19ms/step - loss: 1.9805e-04 -
val_loss: 3.2945e-04
Epoch 99/100
1/1 [==============================] - 0s 21ms/step - loss: 1.9820e-04 -
val_loss: 3.3042e-04
Epoch 100/100
1/1 [==============================] - 0s 21ms/step - loss: 1.9838e-04 -
val_loss: 3.3292e-04
```




```
/Users/sebaslomas/Documents/TesisGuido/.venv/lib/python3.11/site-
packages/plotnine/ggplot.py:587: PlotnineWarning: Saving 6.4 x 4.8 in image.
/Users/sebaslomas/Documents/TesisGuido/.venv/lib/python3.11/site-
packages/plotnine/ggplot.py:588: PlotnineWarning: Filename: plot_nine_output.pdf
```

[49]:
```python
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score

# Predicciones en el conjunto de entrenamiento y prueba
y_train_pred = best_model.predict(X_train)
```

```
y_test_pred = best_model.predict(X_test)

# Escalar las predicciones nuevamente para obtener valores reales
y_train_pred_actual = scaler_y.inverse_transform(y_train_pred)
y_test_pred_actual = scaler_y.inverse_transform(y_test_pred)

# Desescalar los valores de y_train y y_test para obtener valores reales
y_train_actual = scaler_y.inverse_transform(y_train)
y_test_actual = scaler_y.inverse_transform(y_test)

# Calcular y mostrar métricas para el conjunto de entrenamiento
mse_train = mean_squared_error(y_train_actual, y_train_pred_actual)
mae_train = mean_absolute_error(y_train_actual, y_train_pred_actual)
r2_train = r2_score(y_train_actual, y_train_pred_actual)
print(f'Métricas de entrenamiento:')
print(f'MSE: {mse_train:.4f}, MAE: {mae_train:.4f}, R^2: {r2_train:.4f}')

# Calcular y mostrar métricas para el conjunto de prueba
mse_test = mean_squared_error(y_test_actual, y_test_pred_actual)
mae_test = mean_absolute_error(y_test_actual, y_test_pred_actual)
r2_test = r2_score(y_test_actual, y_test_pred_actual)
print(f'Métricas de prueba:')
print(f'MSE: {mse_test:.4f}, MAE: {mae_test:.4f}, R^2: {r2_test:.4f}')
```

```
1/1 [==============================] - 0s 38ms/step
1/1 [==============================] - 0s 15ms/step
Métricas de entrenamiento:
MSE: 0.0000, MAE: 0.0032, R^2: 0.9977
Métricas de prueba:
MSE: 0.0000, MAE: 0.0044, R^2: 0.9943
```

```
2023-07-31 20:46:13.985920: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.
```

[50]:
```
tuner.results_summary()
```

```
Results summary
Results in my_dir/hyperparameter_search
Showing 10 best trials
Objective(name="val_loss", direction="min")

Trial 0209 summary
Hyperparameters:
units_1: 128
units_2: 64
learning_rate: 0.01
tuner/epochs: 100
```

```
tuner/initial_epoch: 34
tuner/bracket: 3
tuner/round: 3
tuner/trial_id: 0204
Score: 0.0003052739193663001

Trial 0235 summary
Hyperparameters:
units_1: 96
units_2: 48
learning_rate: 0.01
tuner/epochs: 100
tuner/initial_epoch: 34
tuner/bracket: 2
tuner/round: 2
tuner/trial_id: 0231
Score: 0.0003061496536247432

Trial 0208 summary
Hyperparameters:
units_1: 96
units_2: 96
learning_rate: 0.01
tuner/epochs: 100
tuner/initial_epoch: 34
tuner/bracket: 3
tuner/round: 3
tuner/trial_id: 0206
Score: 0.0003075377317145467

Trial 0147 summary
Hyperparameters:
units_1: 64
units_2: 16
learning_rate: 0.01
tuner/epochs: 100
tuner/initial_epoch: 34
tuner/bracket: 4
tuner/round: 4
tuner/trial_id: 0145
Score: 0.00030818290542811155

Trial 0234 summary
Hyperparameters:
units_1: 32
units_2: 80
learning_rate: 0.01
tuner/epochs: 100
```

```
tuner/initial_epoch: 34
tuner/bracket: 2
tuner/round: 2
tuner/trial_id: 0230
Score: 0.00032353526330552995

Trial 0196 summary
Hyperparameters:
units_1: 256
units_2: 32
learning_rate: 0.01
tuner/epochs: 12
tuner/initial_epoch: 4
tuner/bracket: 3
tuner/round: 1
tuner/trial_id: 0169
Score: 0.00032679986907169223

Trial 0146 summary
Hyperparameters:
units_1: 32
units_2: 16
learning_rate: 0.0001
tuner/epochs: 100
tuner/initial_epoch: 34
tuner/bracket: 4
tuner/round: 4
tuner/trial_id: 0144
Score: 0.00033172083203680813

Trial 0206 summary
Hyperparameters:
units_1: 96
units_2: 96
learning_rate: 0.01
tuner/epochs: 34
tuner/initial_epoch: 12
tuner/bracket: 3
tuner/round: 2
tuner/trial_id: 0189
Score: 0.00033379931119270623

Trial 0239 summary
Hyperparameters:
units_1: 96
units_2: 128
learning_rate: 0.01
tuner/epochs: 34
```

```
tuner/initial_epoch: 0
tuner/bracket: 1
tuner/round: 0
Score: 0.00033648789394646883

Trial 0240 summary
Hyperparameters:
units_1: 128
units_2: 96
learning_rate: 0.01
tuner/epochs: 34
tuner/initial_epoch: 0
tuner/bracket: 1
tuner/round: 0
Score: 0.0003394547966308892
```

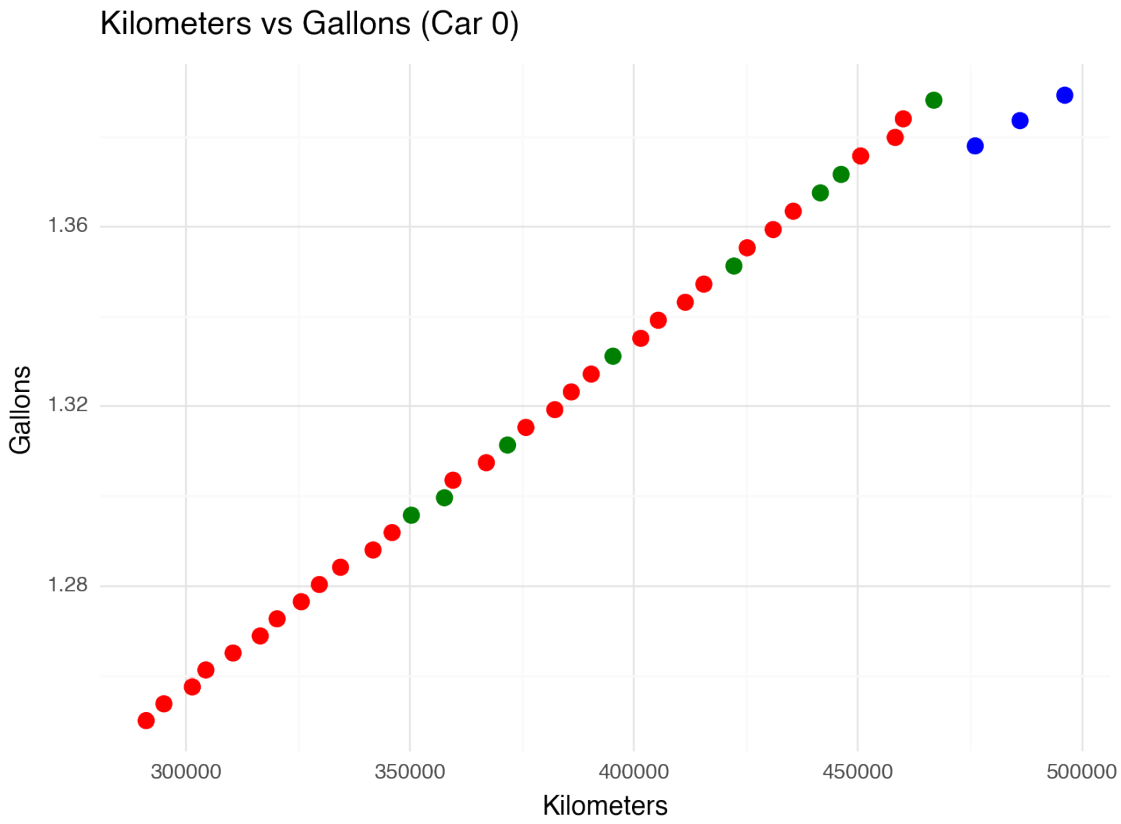
0.0.7 Entremaniento para multiples maquinarias

[81]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
import tensorflow as tf
from plotnine import ggplot, aes, geom_point, labs, theme_minimal, theme

# Generalize the code for multiple cars
car_models = {}       # Dictionary to store models for each car
train_losses_dict = {}  # Dictionary to store training losses for each car
val_losses_dict = {}    # Dictionary to store validation losses for each car

for car_idx in range(len(kilometers_list)):
    print(f'Training model for Car {car_idx}')
    print("-----------------")
    X = kilometers_list[car_idx] + np.random.normal(0, 1000, 36)  # Add some␣
 ↪noise
    y = galones[car_idx]


    # Data normalization
    scaler_X = MinMaxScaler()
    scaler_y = MinMaxScaler()
    X_normalized = scaler_X.fit_transform(X.reshape(-1, 1))
    y_normalized = scaler_y.fit_transform(y.reshape(-1, 1))

    # Splitting the data into training and testing sets
```

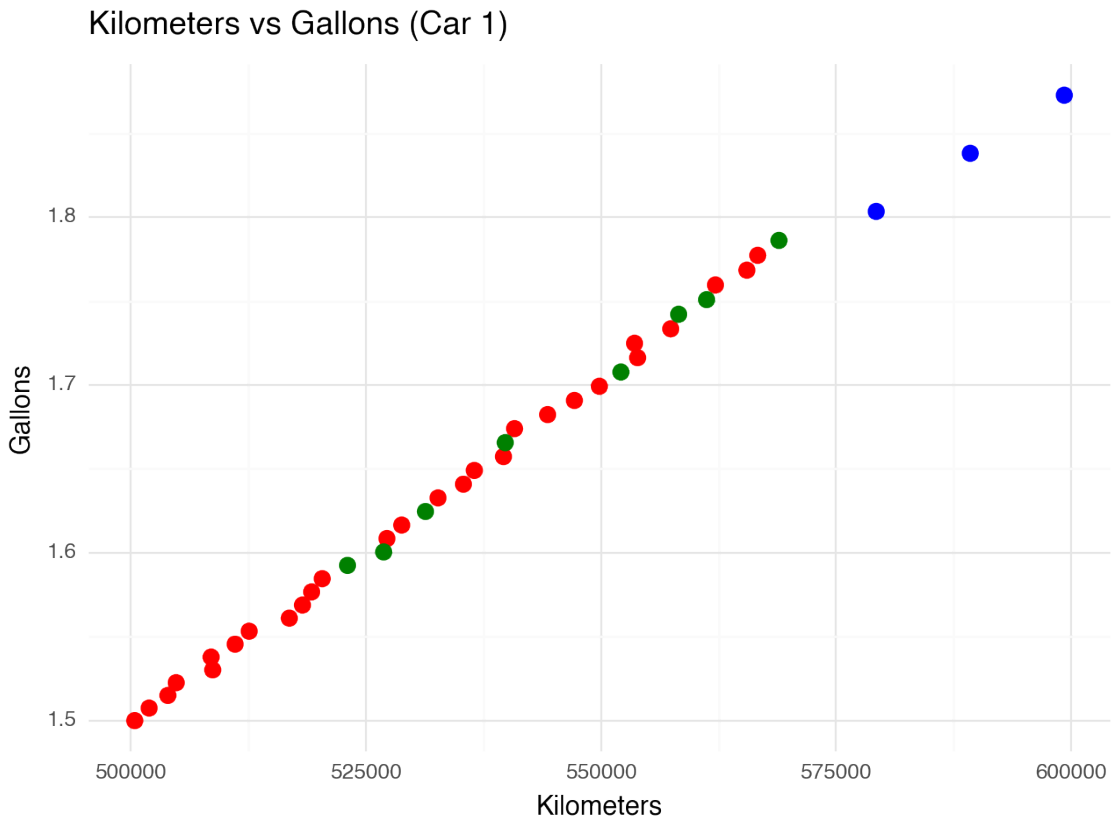
```
  X_train, X_test, y_train, y_test = train_test_split(X_normalized,␣
↪y_normalized, test_size=0.2, random_state=42)

  # Neural network model NAR TWO HIDDEN LAYERS and one neuron of prediction

  model = tf.keras.Sequential([
      tf.keras.layers.Dense(64, activation='softmax', input_shape=(1,)),
      tf.keras.layers.Dense(32, activation='relu'),
      tf.keras.layers.Dense(1)
  ])

  # Compile the model with mean squared error loss function
  model.compile(optimizer='adam', loss='mean_squared_error')

  # Train the model
  history = model.fit(X_train, y_train, epochs=10, batch_size=5,␣
↪validation_data=(X_test, y_test), verbose=0)

  # Store the training and validation losses for the car
  train_losses_dict[f'Car_{car_idx}'] = history.history['loss']
  val_losses_dict[f'Car_{car_idx}'] = history.history['val_loss']


  X_train_inv = scaler_X.inverse_transform(X_train)
  X_test_inv = scaler_X.inverse_transform(X_test)
  y_train_inv = scaler_y.inverse_transform(y_train)
  y_test_inv = scaler_y.inverse_transform(y_test)


  future_kilometers = np.array([
      dataframe[" n12"][28 + car_idx: 29 + car_idx ].values[0] + 10000,
      dataframe[" n12"][28 + car_idx: 29 + car_idx ].values[0] + 20000,
      dataframe[" n12"][28 + car_idx: 29 + car_idx ].values[0] + 30000
  ])

  future_kilometers_normalized = scaler_X.transform(future_kilometers.
↪reshape(-1, 1))

  # Make predictions for the car
  predicted_gallons_normalized = model.predict(future_kilometers_normalized)
  predicted_gallons = scaler_y.inverse_transform(predicted_gallons_normalized)

  # Create DataFrames for training, test, and predicted data points
  data_train = pd.DataFrame({'Kilometros': X_train_inv.ravel(), 'Galones':␣
↪y_train_inv.ravel()})
  data_test = pd.DataFrame({'Kilometros': X_test_inv.ravel(), 'Galones':␣
↪y_test_inv.ravel()})
```

```
    data_predicted = pd.DataFrame({'Kilometros': future_kilometers, 'Galones':␣
↪predicted_gallons.ravel()})


    # Convert pandas DataFrames to plotnine objects
    p = ggplot()

    # Scatter plot for training data
    p += geom_point(data=data_train, mapping=aes(x='Kilometros', y='Galones'),␣
↪color='red', size=3)
    # Scatter plot for test data
    p += geom_point(data=data_test, mapping=aes(x='Kilometros', y='Galones'),␣
↪color='green', size=3)
    # Scatter plot for predicted data
    p += geom_point(data=data_predicted, mapping=aes(x='Kilometros',␣
↪y='Galones'), color='blue', size=3)
    # Other plot aesthetics and settings
    p += aes(x='Kilometros', y='Galones')
    p += labs(x='Kilometers', y='Galones', color='Data')
    p += labs(title=f'Kilometers vs Galones (Car {car_idx})')
    p += theme_minimal() # Minimalistic theme

        # Display the plot for the current car
    print(p)
```

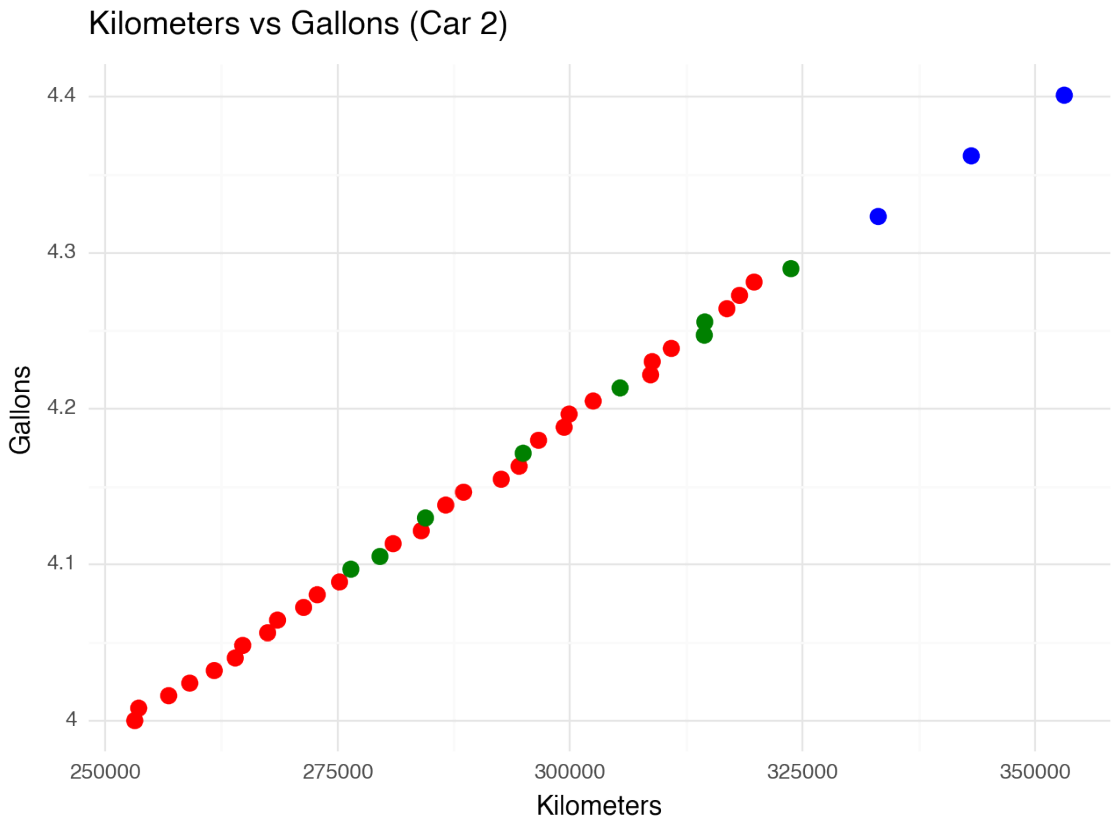
```
Training model for Car 0
-----------------

2023-08-03 02:14:16.042230: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.
2023-08-03 02:14:16.423782: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.

[476142 486142 496142]
1/1 [==============================] - 0s 32ms/step

2023-08-03 02:14:17.097556: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.
```



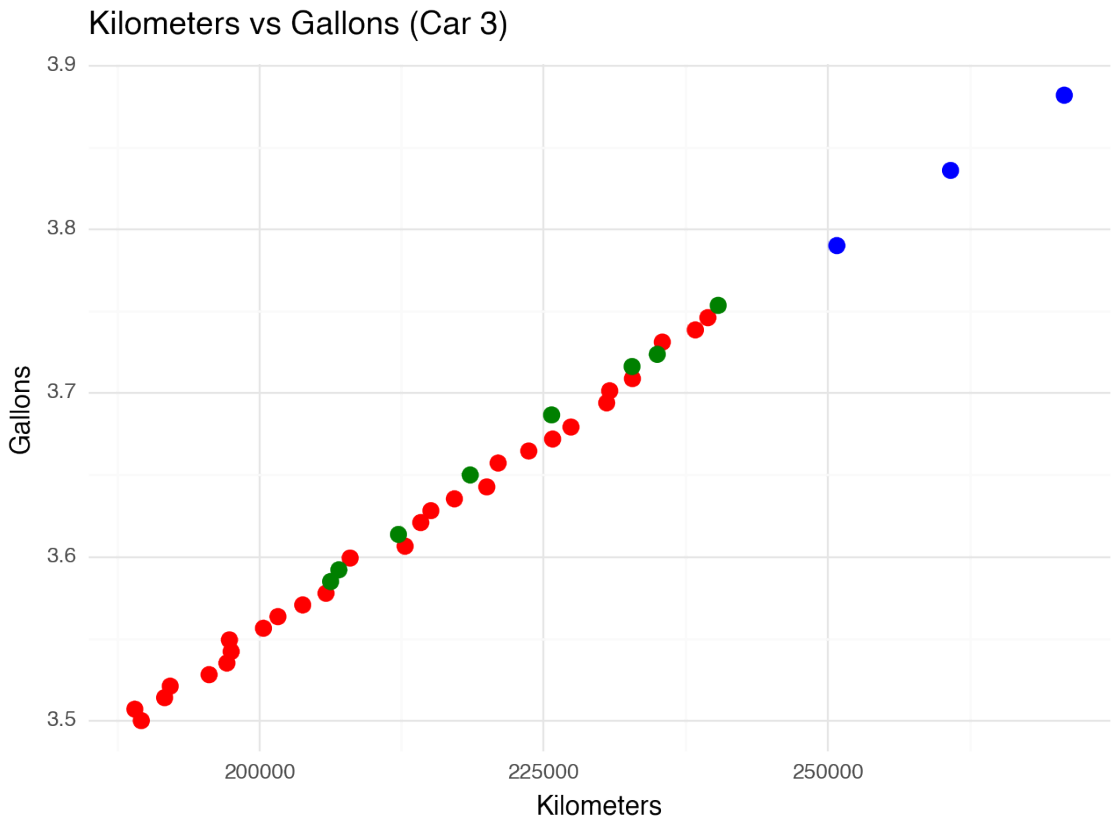

```
Training model for Car 1
-----------------

2023-08-03 02:14:17.409048: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.
2023-08-03 02:14:17.778109: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.

[579265 589265 599265]
1/1 [==============================] - 0s 26ms/step

2023-08-03 02:14:18.399097: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.
```



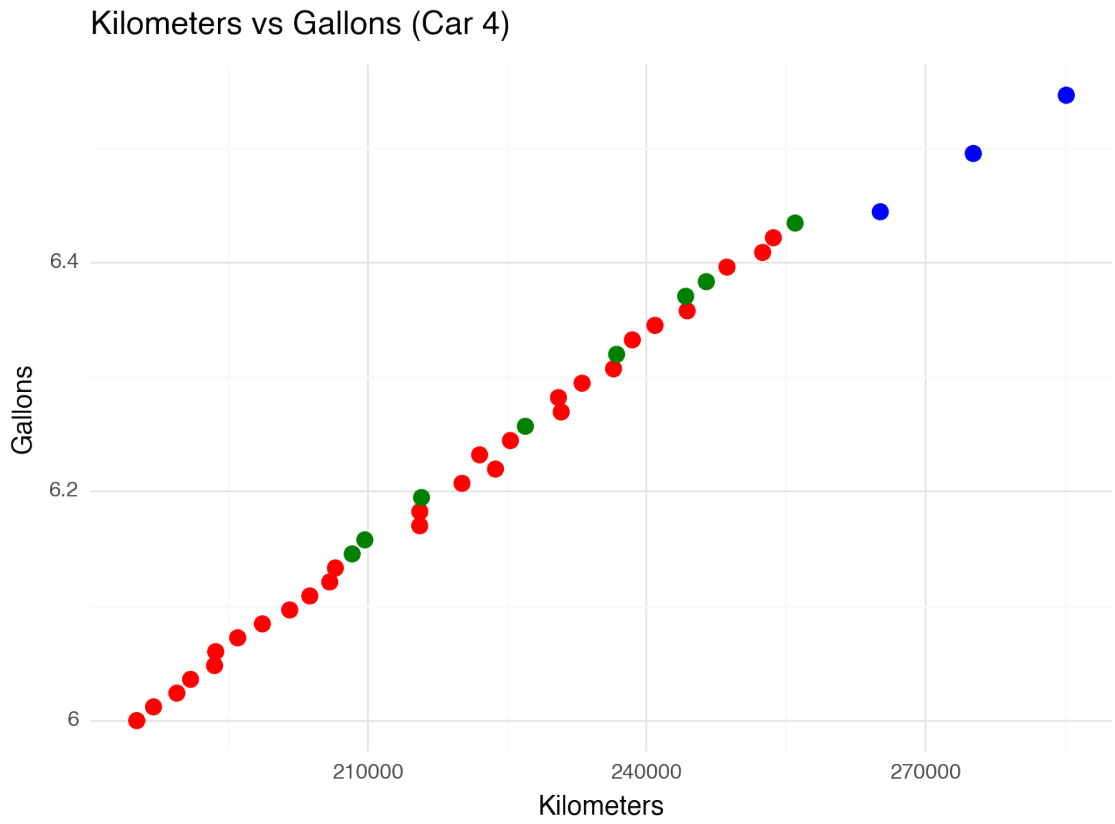

```
Training model for Car 2
-----------------

2023-08-03 02:14:19.142892: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.
2023-08-03 02:14:19.524457: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.

[333134 343134 353134]
1/1 [==============================] - 0s 31ms/step

2023-08-03 02:14:20.126370: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.
```



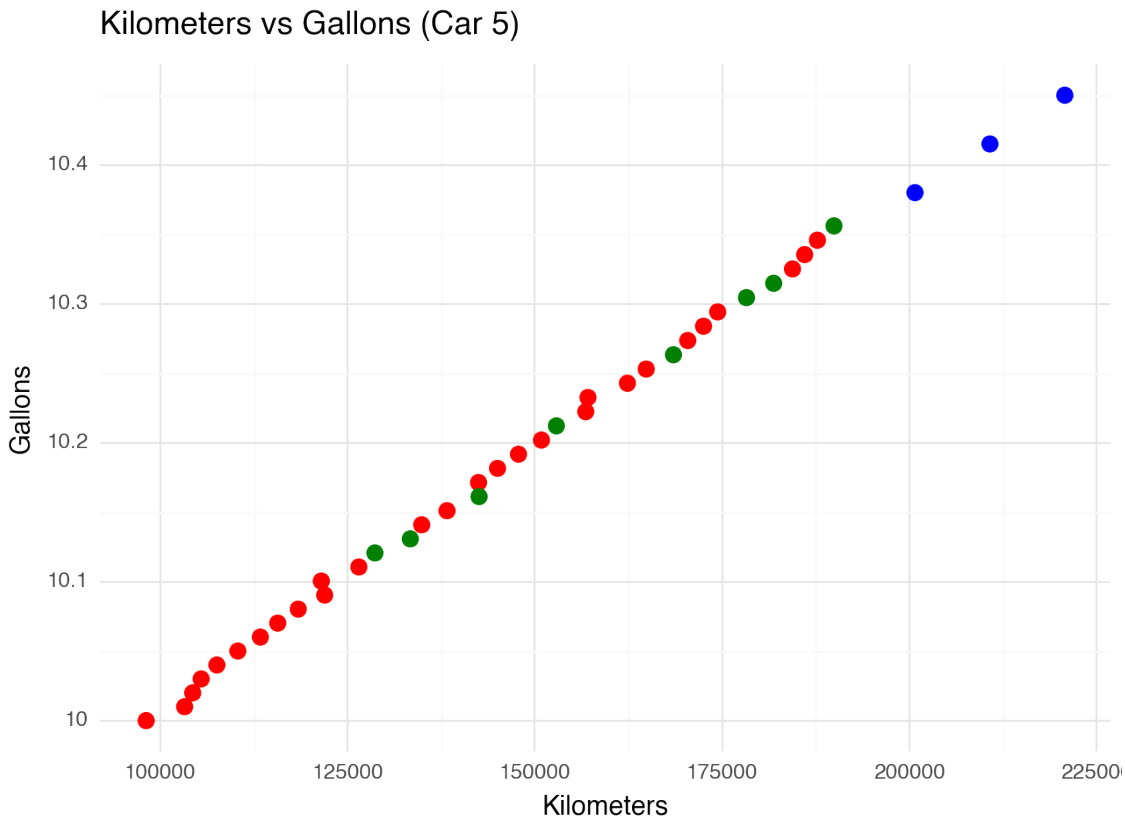

```
Training model for Car 3
-----------------

2023-08-03 02:14:20.430422: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.
2023-08-03 02:14:20.770234: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.

[250798 260798 270798]
1/1 [==============================] - 0s 29ms/step

2023-08-03 02:14:21.434104: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.
```



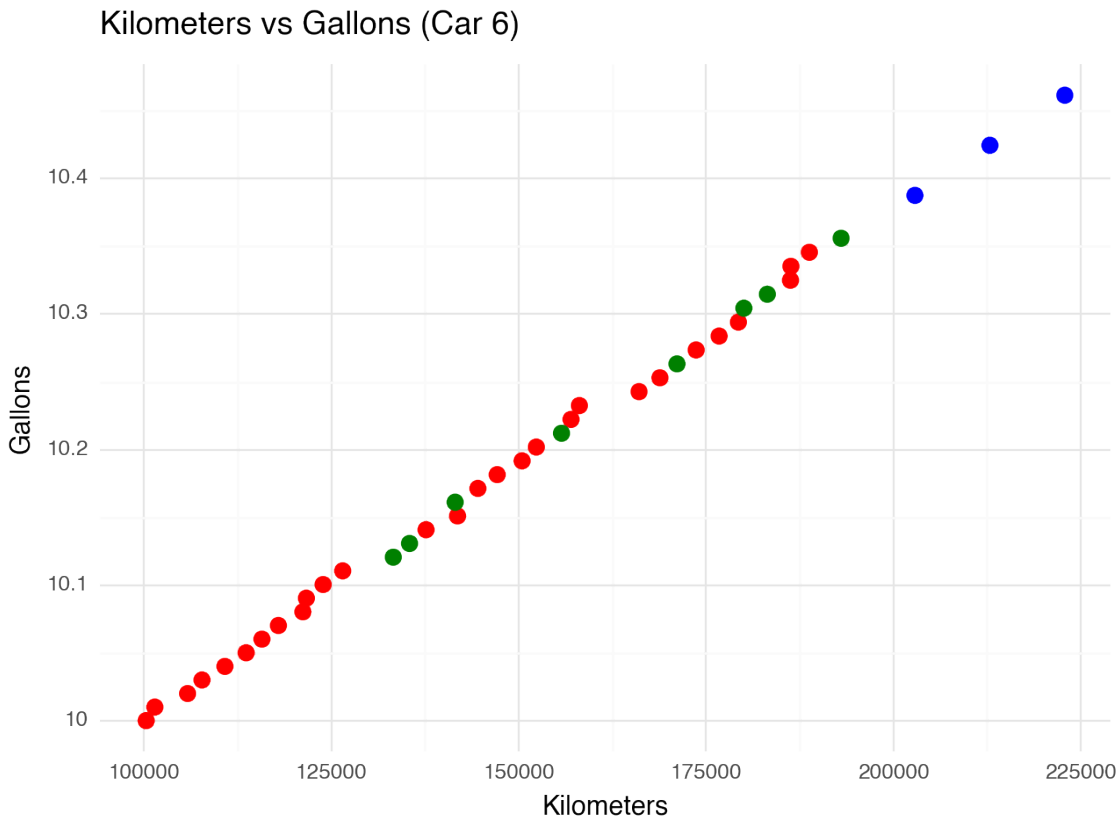

```
Training model for Car 4
-----------------

2023-08-03 02:14:21.746687: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.
2023-08-03 02:14:22.104789: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.

[265128 275128 285128]
1/1 [==============================] - 0s 27ms/step

2023-08-03 02:14:22.736576: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.
```



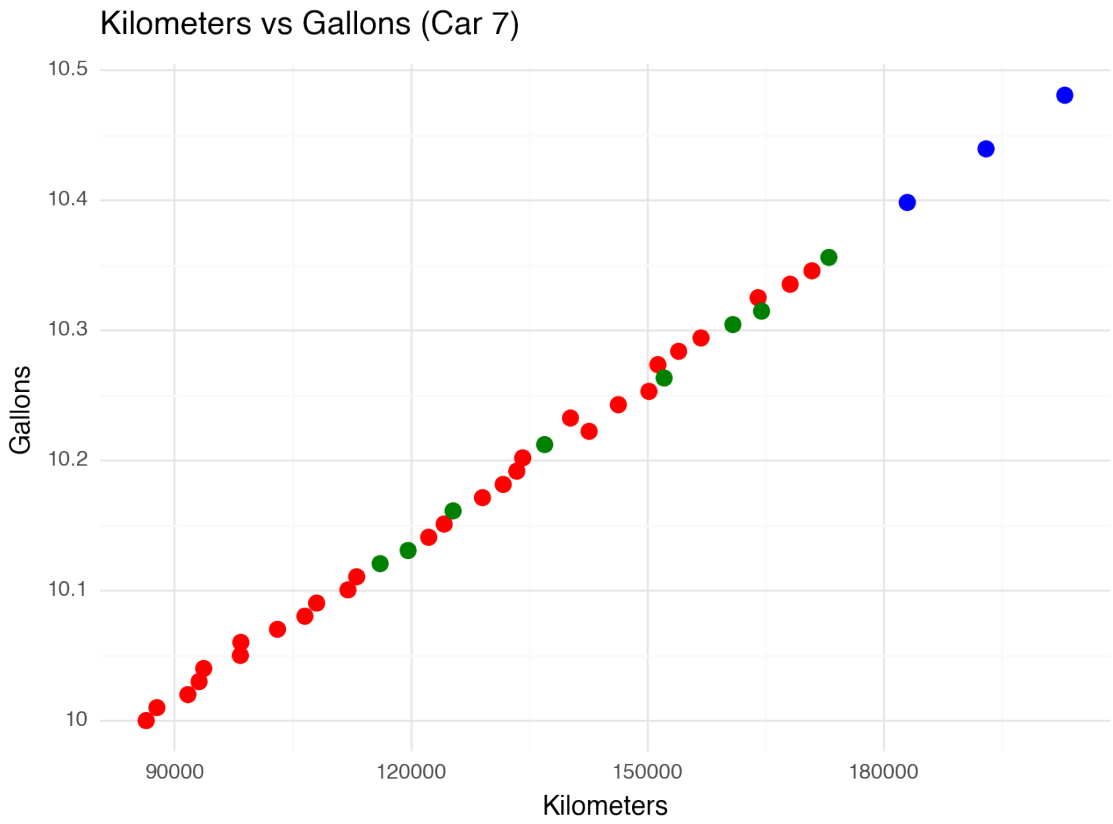

```
Training model for Car 5
-----------------

2023-08-03 02:14:23.028022: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.
2023-08-03 02:14:23.345496: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.

[200750 210750 220750]
1/1 [==============================] - 0s 31ms/step

2023-08-03 02:14:23.994971: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.
```



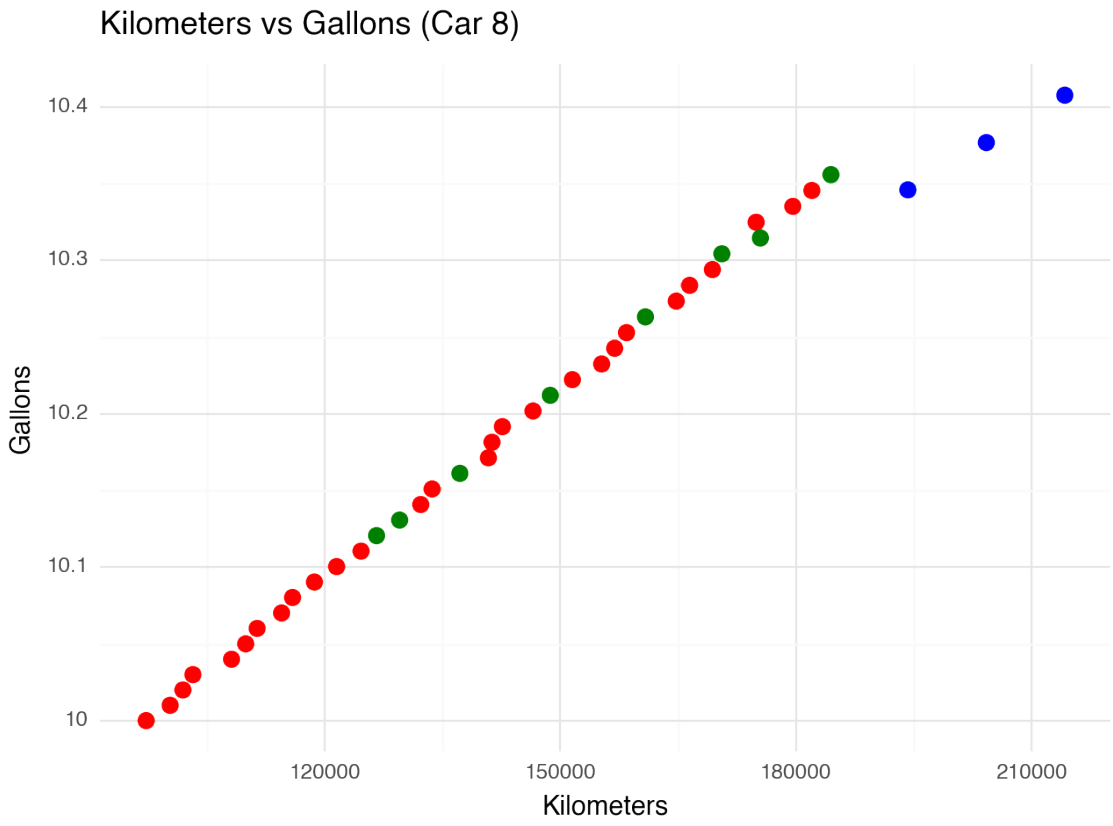

```
Training model for Car 6
-----------------

2023-08-03 02:14:24.295589: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.
2023-08-03 02:14:24.663061: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.

[202872 212872 222872]
1/1 [==============================] - 0s 29ms/step

2023-08-03 02:14:25.318608: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.
```



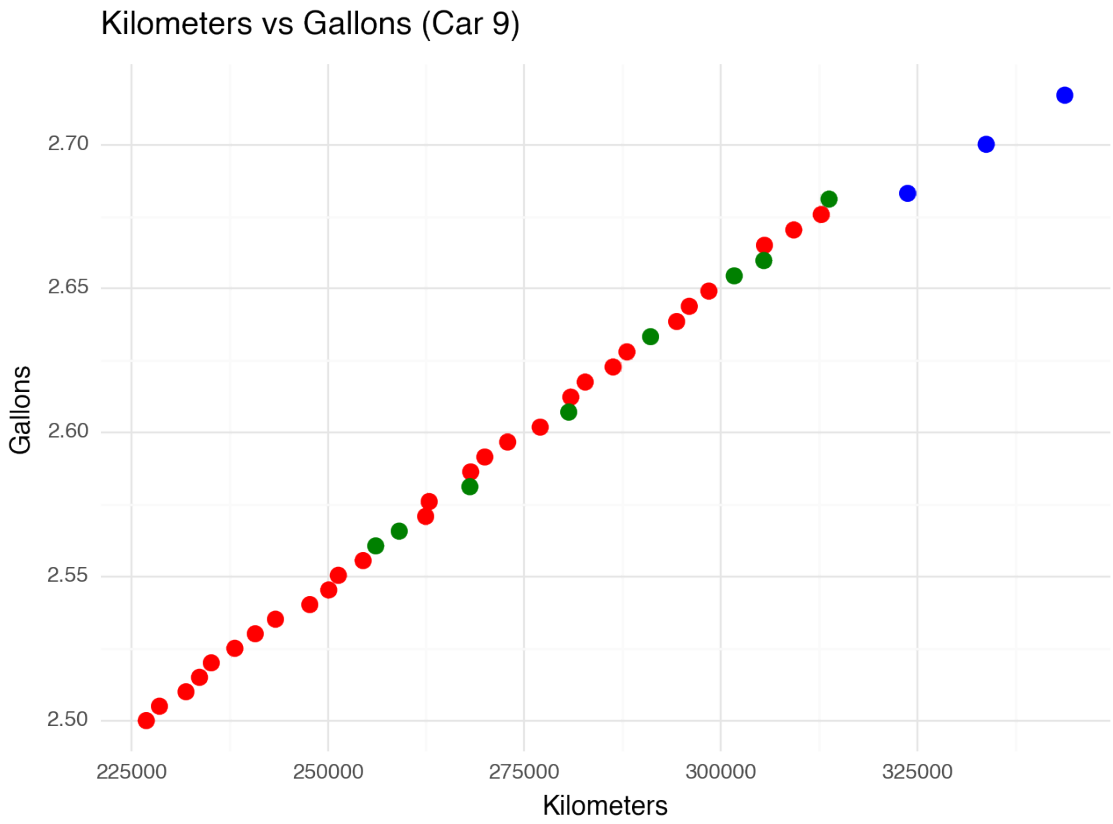

```
Training model for Car 7
------------------

2023-08-03 02:14:25.624385: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.
2023-08-03 02:14:25.963632: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.

[183000 193000 203000]
1/1 [==============================] - 0s 28ms/step

2023-08-03 02:14:26.613659: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.
```



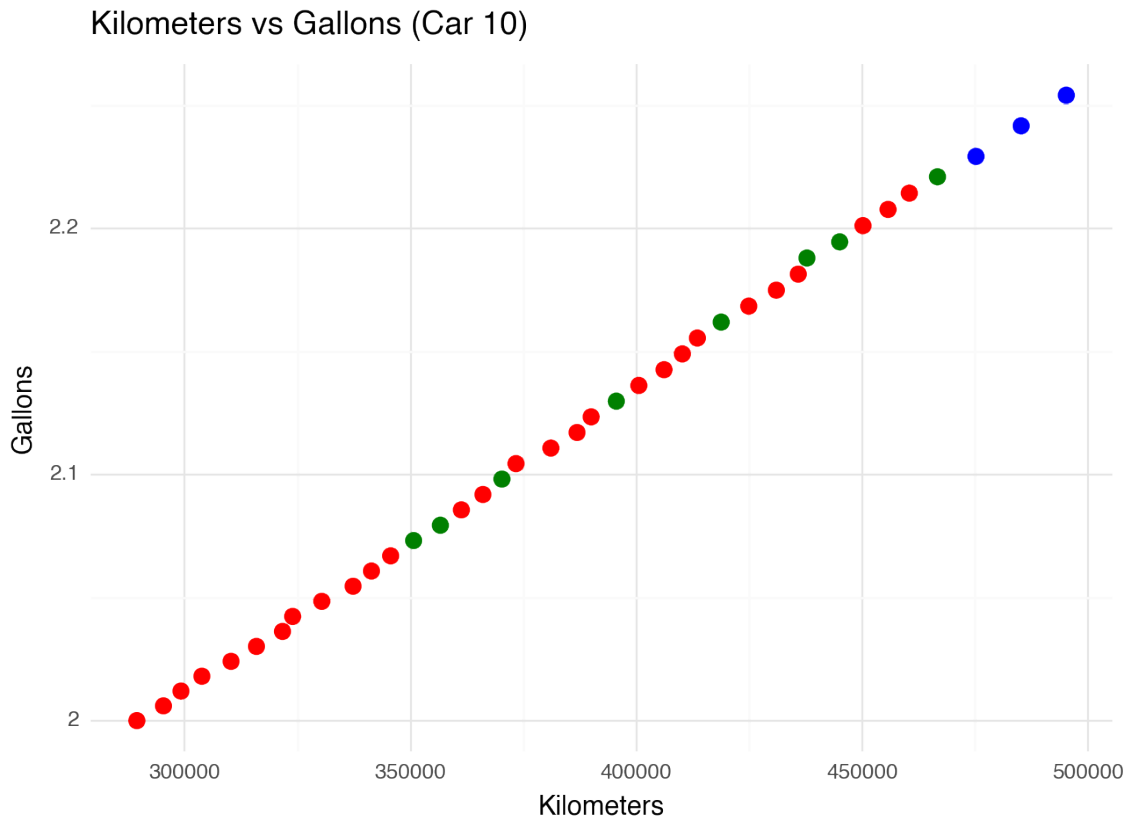

```
Training model for Car 8
-----------------

2023-08-03 02:14:26.908208: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.
2023-08-03 02:14:27.300950: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.

[194283 204283 214283]
1/1 [==============================] - 0s 28ms/step

2023-08-03 02:14:27.923717: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.
```



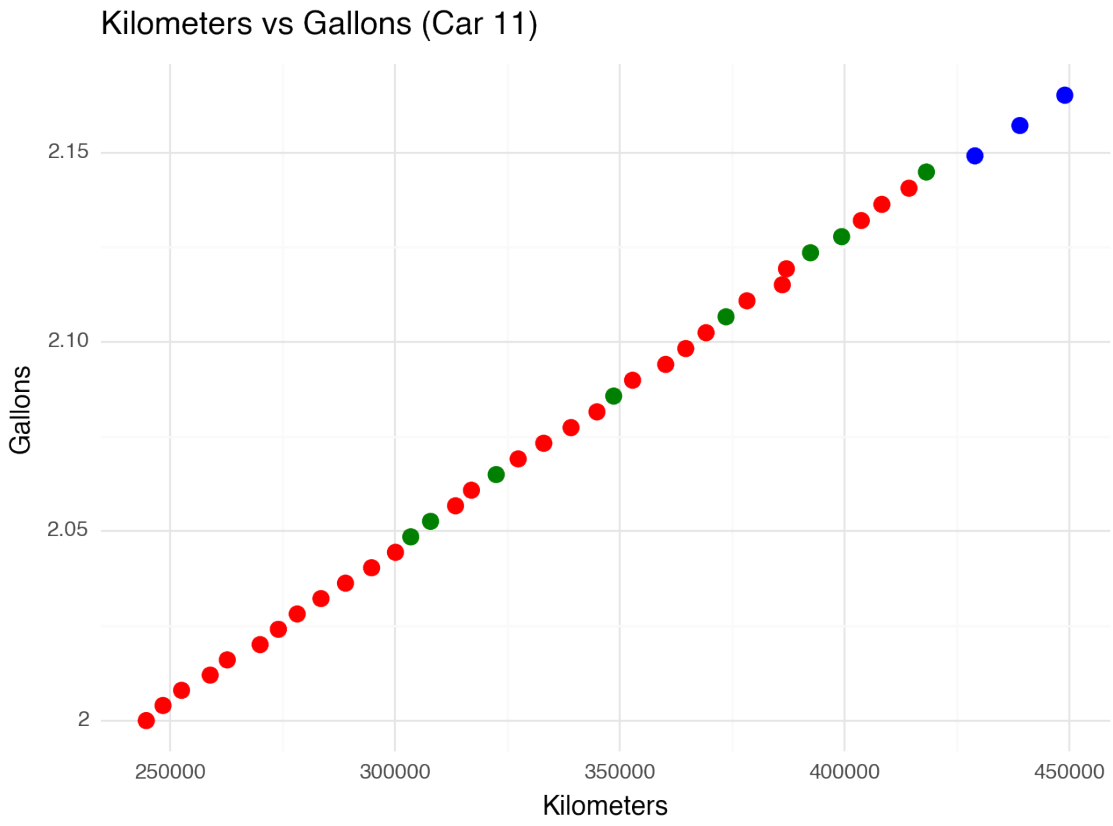

```
Training model for Car 9
-----------------

2023-08-03 02:14:28.219143: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.
2023-08-03 02:14:28.571484: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.

[323779 333779 343779]
1/1 [==============================] - 0s 28ms/step

2023-08-03 02:14:29.183392: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.
```



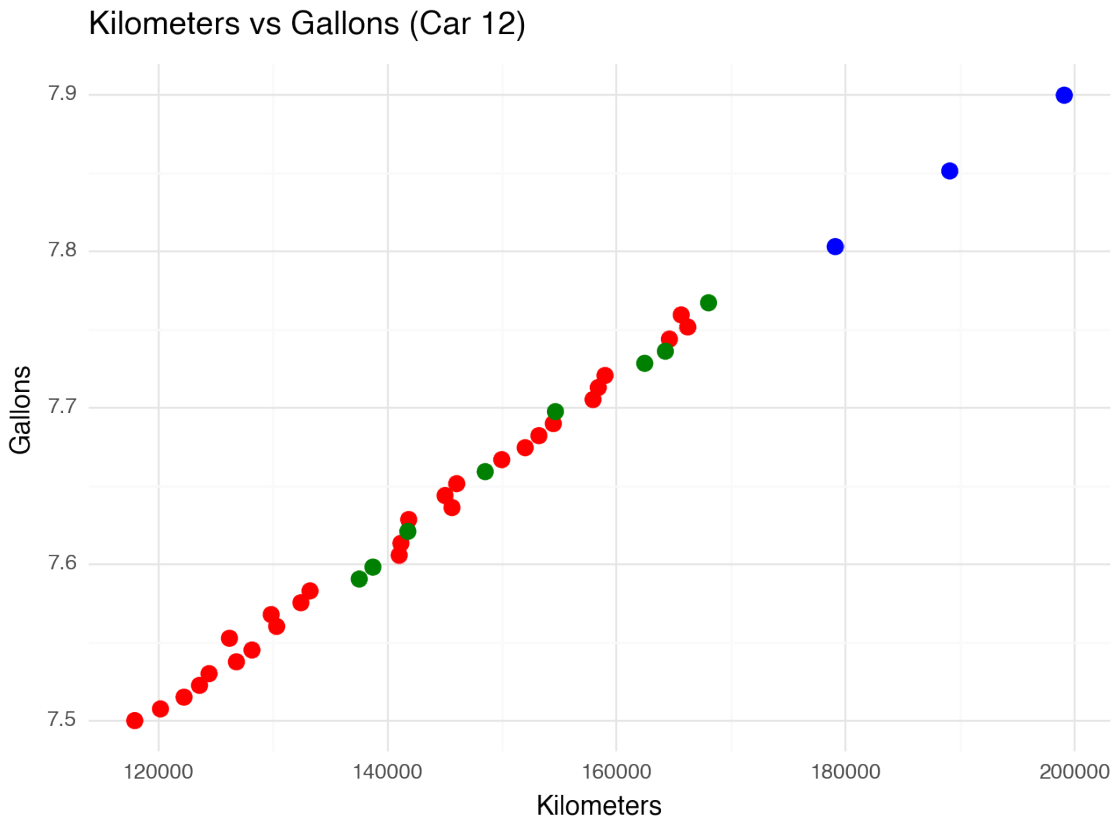

```
Training model for Car 10
-----------------

2023-08-03 02:14:29.474912: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.
2023-08-03 02:14:29.824642: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.

[475204 485204 495204]
1/1 [==============================] - 0s 28ms/step

2023-08-03 02:14:30.447056: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.
```



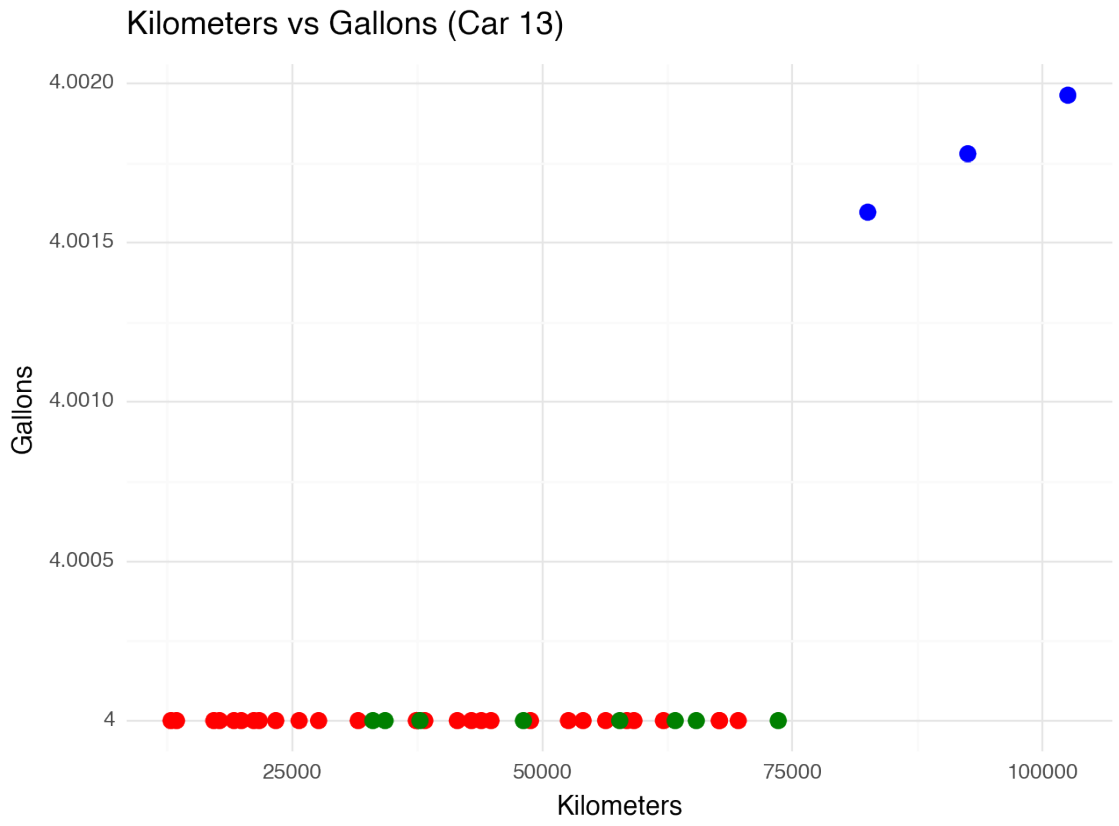

```
Training model for Car 11
-----------------

2023-08-03 02:14:30.731727: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.
2023-08-03 02:14:31.120317: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.

[429032 439032 449032]
1/1 [==============================] - 0s 30ms/step

2023-08-03 02:14:31.698949: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.
```




```
Training model for Car 12
-----------------

2023-08-03 02:14:31.992051: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.
2023-08-03 02:14:32.389304: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.

[179101 189101 199101]
1/1 [==============================] - 0s 29ms/step

2023-08-03 02:14:32.964843: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.
```




```
Training model for Car 13
-----------------

2023-08-03 02:14:33.249397: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.
2023-08-03 02:14:33.565396: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.

[ 82530  92530 102530]
1/1 [==============================] - 0s 25ms/step

2023-08-03 02:14:34.245935: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.
```



{} {'Car_0': [0.5324514508247375, 0.2957758605480194, 0.15508544445037842,
0.08064094930887222, 0.0333794429898262, 0.018656520172953606,
0.013271794654428959, 0.012857562862336636, 0.011859184131026268,
0.010127967223525047], 'Car_1': [0.3867131173610687, 0.21402429044246674,
0.11249673366546631, 0.04635264351963997, 0.016200680285692215,
0.010123049840331078, 0.008893943391740322, 0.007487450260668993,
0.006169444415718317, 0.004674712661653757], 'Car_2': [0.10932408273220062,
0.030590323731303215, 0.004852975718677044, 0.004923746921122074,
0.0049293083138763905, 0.003274735528975725, 0.0026477219071239233,
0.0021083082538098097, 0.0013774849940091372, 0.0011462762486189604], 'Car_3':
[0.10719355195760727, 0.036007724702358246, 0.005792870186269283,
0.0028122770600020885, 0.0023267820943146944, 0.0018187040695920587,
0.001462015206925571, 0.001067887875251472, 0.0008406703709624708,
0.0005290411645546556], 'Car_4': [0.32998713850975037, 0.1791640818119049,
0.08524523675441742, 0.0388869009912014, 0.020056288689374924,
0.011270837858319283, 0.007927254773676395, 0.006401270162314176,
0.004824171774089336, 0.0035840016789734364], 'Car_5': [0.16250260174274445,
0.06548073142766953, 0.015018107369542122, 0.004812707658857107,
0.004885077942162752, 0.0041083674877882, 0.0032569468021392822,
0.0028863009065389633, 0.0018818805692717433, 0.0013648106250911955], 'Car_6':

[0.10723813623189926, 0.029665812849998474, 0.00405769282951951,
0.003268279368057847, 0.0023798574693500996, 0.0017584615852683783,
0.0012061705347150564, 0.0008806927944533527, 0.0005563334561884403,
0.0004156982176937163], 'Car_7': [0.014283218421041965, 0.0016241568373516202,
0.002425593789666891, 0.000753239612095058, 0.0009282446699216962,
0.0003637461340986192, 0.00026429592981003225, 0.0002760483475867659,
0.0002559662680141628, 0.0002575666585471481], 'Car_8': [0.5269155502319336,
0.2954297959804535, 0.12967772781848907, 0.05172329396009445,
0.01587758958339691, 0.008457104675471783, 0.009072734974324703,
0.00855665560811758, 0.007176797837018967, 0.005936585832387209], 'Car_9':
[0.2822337746620178, 0.169057697057724, 0.09438557922840118,
0.04876837879419327, 0.024515053257346153, 0.012361365370452404,
0.006563194096088409, 0.00557751627638936, 0.004722338169813156,
0.0038174260407686234], 'Car_10': [0.07891841977834702, 0.02562515065073967,
0.005248133093118668, 0.0021775239147245884, 0.002080029807984829,
0.0013600714737549424, 0.000910157454200089, 0.0004424861981533468,
0.0001932816085172817, 0.00014302491035778075], 'Car_11': [0.04125220328569412,
0.002411479828879237, 0.004306180868297815, 0.0036959839053452015,
0.0008466496947221458, 0.0015115158166736364, 0.0012395632220432162,
0.0003088130033574998 4, 0.0001563770929351449, 0.00013676760136149824],
'Car_12': [0.19175119698047638, 0.07486188411712646, 0.030672218650579453,
0.008839741349220276, 0.005191106349229813, 0.004108567256480455,
0.003579861018806696, 0.003242390463128686, 0.002114284783601761,
0.0015721680829301476], 'Car_13': [0.00014951782941352576,
0.0001052958468790166 1, 5.716417217627168e-05, 4.3504151108209044e-05,
1.991326644201763e-05, 1.4669487427454442e-05, 1.1363619705662131e-05,
6.947002930246526e-06, 2.524656792957103e-06, 1.3783136409983854e-06]} {'Car_0':
[0.6154011487960815, 0.34253522753715515, 0.1737271398305893,
0.0748276561498642, 0.03035556524991989, 0.01262308843433857,
0.007350103929638863, 0.005715769715607166, 0.0052569676190614 7,
0.004982183687388897], 'Car_1': [0.4563378691673279, 0.2575748562812805,
0.119248166680336, 0.040070049464702606, 0.011888368055224419,
0.005449309945106506, 0.004499029368162155, 0.004594366531819105,
0.003615150228142738 3, 0.002241784008219838], 'Car_2': [0.09073304384946823,
0.01374608464539051, 0.0014764919178560376, 0.002773003187030554,
0.0012877490371465683, 0.0021024630405008793, 0.0021575060673058033,
0.000877707265317 44, 0.00041615634108893573, 0.0003077824949286878], 'Car_3':
[0.0865616500377655, 0.020602256059646606, 0.0035255257971584797,
0.0010975520126521587, 0.0017702281475067139, 0.0030141319148242474,
0.002723696641623974, 0.002097852062433958, 0.0016898149624466896,
0.000891464529559016 2], 'Car_4': [0.36652272939682007, 0.20408117771148682,
0.0949382334947586, 0.043114736676216125, 0.017419906333088875,
0.00532824732363224, 0.0030163307674229145, 0.002768256934359669 7,
0.002859777770936489, 0.0027492274530231953], 'Car_5': [0.16787005960941315,
0.046668242663145065, 0.006816660985350609, 0.002139156684279442,
0.0022184341214597225, 0.0018326486460864544, 0.002748931059613824,
0.0018303608521819115, 0.0008846815908327699, 0.0006470225634984672], 'Car_6':
[0.08633184432983398, 0.009586076252162457, 0.001627518911845982,

0.0015057831769809127, 0.0018071483355015516, 0.0007880475604906678,
0.0005733073921874166, 0.00040721282130107284, 0.00039753070450387895,
0.0002248621399070323], 'Car_7': [0.0011183135211467743, 0.003838691394776106,
0.0007609783788211644, 0.0010957492049783468, 0.0003953751875087619,
0.0001819490280468 0169, 0.00024488952476531267, 0.00027227328973822296,
0.0002052403433481 2313, 0.0001725808106129989], 'Car_8': [0.6196730136871338,
0.30461907386779785, 0.12571436166763306, 0.03935009986162186,
0.008882268331944942, 0.0033595068380236626, 0.003487009322270751,
0.0028919754549860954, 0.0034570421557873487, 0.00430636340752244], 'Car_9':
[0.3354867696762085, 0.1926611214876175, 0.1087983250617981,
0.057206157594919205, 0.023654207587242126, 0.007887884974479675,
0.003512029768899083, 0.0028067256789654493, 0.0024345179554075003,
0.002130510751157999], 'Car_10': [0.07442950457334518, 0.01822330802679062,
0.00364321144297719, 0.0011827144771814346, 0.0010446360101923347,
0.0014568885089829564, 0.0007057313341647387, 0.00027452409267425537,
0.0004202308 1914521754, 0.0001456442550988868], 'Car_11': [0.011331215500831604,
0.0016351256053894758, 0.004503908101469278, 0.0006683199317194521,
0.0021915570832788944, 0.00310343224555254, 0.0010471458081156015,
9.588220564182848e-05, 0.0001508363347966224, 0.0002632818650454283], 'Car_12':
[0.18793055415153503, 0.06835086643695831, 0.01920303888618946,
0.007019301876425743, 0.003223574720323086, 0.002991319168359041,
0.00394720071926713, 0.0026464483235031366, 0.001845603110268712,
0.001353141968138516], 'Car_13': [0.00015385152073577046,
3.1292365747503936e-05, 6.002843747410225e-06, 2.421240060357377e-05,
2.4560627934988588e-05, 1.0915097846009303e-05, 6.467696039180737e-06,
1.6081166904768907e-06, 1.1913335811186698e-06, 1.130052623921074e-06]}