

**Modelo predictivo usando Machine Learning para determinar millas por galón**

**Predictive model using Machine Learning to determine miles per gallon**

**Modelo preditivo usando aprendizaje de máquina para determinar milhas por galão**

María Alejandrina Nivelá Cornejo<sup>1</sup>  
Universidad Bolivariana del Ecuador  
[manivela@ube.edu.ec](mailto:manivela@ube.edu.ec)  
<https://orcid.org/0000-0002-0356-7243>



 DOI / URL: <https://doi.org/10.55813/gaea/ccri/v5/n1/467>

**Como citar:**

Nivelá, M. (2024). *Modelo predictivo usando Machine Learning para determinar millas por galón*. *Código Científico Revista de Investigación*, 5(1), 1528-1545.

**Recibido:** 02/05/2024

**Aceptado:** 04/06/2024

**Publicado:** 30/06/2024

---

<sup>1</sup> Doctora en Ciencias Humanas. Magister en Educación Informática, Licenciada en Educación Informática. Docente de Pregrado y Posgrado. Docente investigadora acreditada por la Senescyt. Especialista en Informática Educativa comprometida con la innovación educativa y el desarrollo de estrategias pedagógicas basadas en tecnología.

## Resumen

La contaminación vehicular es la principal causa del cambio climático, entre los gases que generan el efecto invernadero, el CO<sub>2</sub> es el que más contribuye al calentamiento global. Gracias a las técnicas de inteligencia artificial, dentro del subcampo del aprendizaje automático o machine Learning, se han dado avances significativos que permiten agilizar el análisis de grandes volúmenes de datos. Este estudio tiene como objetivo, determinar el problema de, construir un modelo para predecir la eficiencia de vehículos de 1970 y 1980. Para hacer esto proveeremos el modelo con una descripción de muchos automóviles de ese periodo. Esta descripción incluye atributos como: Cilindros, desplazamiento, potencia y peso. con el lenguaje de programación Python. En la metodología usada se crearon los modelos, que luego fueron definidos y entrenados y después de realizar las pruebas y predicciones, las métricas de validación determinaron la precisión y efectividad de estos modelos. De los resultados obtenidos se concluye que el modelo es bueno dado que proporcionó una precisión del 85% por lo que finalmente con estos modelos se podrán realizar predicciones confiables en cuanto a la efectividad.

**Palabras clave:** Machine Learning, millas por galón, lenguaje de programación.

## Abstract

Vehicle pollution is the main cause of climate change, among the gases that generate the greenhouse effect, CO<sub>2</sub> is the one that contributes the most to global warming. Thanks to artificial intelligence techniques, within the subfield of automatic learning or machine learning, significant advances have been made that allow speeding up the analysis of large volumes of data. This study aims to determine the problem of building a model to predict the efficiency of vehicles from 1970 and 1980. To do this we will provide the model with a description of many cars from that period. This description includes attributes such as: Cylinders, Displacement, Power and Weight. with the Python programming language. In the methodology used, the models were created, which were then defined and trained and after performing the tests and predictions, the validation metrics determined the accuracy and effectiveness of these models. From the results obtained, it is concluded that the model is good since it provided an accuracy of 85%, so finally, with these models, reliable predictions can be made in terms of effectiveness.

**Keywords:** Machine Learning, miles per gallon, Programming Language.

## Resumo

A poluição veicular é a principal causa das mudanças climáticas; entre os gases que geram o efeito estufa, o CO<sub>2</sub> é o que mais contribui para o aquecimento global. Graças às técnicas de inteligência artificial, no subcampo do aprendizado de máquina, foram alcançados avanços significativos que permitem agilizar a análise de grandes volumes de dados. Este estudo tem como objetivo determinar o problema de construção de um modelo para prever a eficiência dos veículos das décadas de 1970 e 1980. Para isso forneceremos ao modelo uma descrição de muitos carros desse período. Esta descrição inclui atributos como: Cilindros, cilindrada, potência e peso. com a linguagem de programação Python. Na metodologia utilizada foram criados os modelos, que foram então definidos e treinados e após a realização dos testes e

previsões, as métricas de validação determinaram a precisão e eficácia destes modelos. Dos resultados obtidos conclui-se que o modelo é bom pois proporcionou uma precisão de 85%, pelo que finalmente com estes modelos podem ser feitas previsões fiáveis quanto à eficácia.

**Palavras-chave:** Aprendizado de máquina, milhas por galão, linguagem de programação.

## **Introducción**

El deterioro del medio ambiente está ligado con el aumento de consumo de energía, mayor volumen de energía en países industrializados alcanzan tasas elevadas de contaminación, es necesario economizar energía, aumentar eficiencia de aprovechamiento y por motivos ecológicos se, ha desarrollado con mejor tecnología los vehículos, la reformulación de combustibles y el uso de combustibles variados.

En todo el mundo las necesidades de transporte derivadas del crecimiento acelerado de las grandes ciudades evidencian el aumento de problemas de movilidad urbana (CAF, 2017).

[1] La preferencia por el automóvil, como principal medio de transporte en las ciudades en todo el mundo, causa cada vez más contratiempos.

(Aldas, Reyes, 2018) Morales, Nuñez, y Toaza (desarrollan una investigación que analiza los impactos de modelos de transporte, selecciona la mejor alternativa tomando cuatro ciudades con alto índice de movilidad urbana y condiciones óptimas de desarrollo sostenible, establecen los criterios de prioridad como tráfico, impacto ambiental, impacto social e impacto económico y evalúan.

## **Desarrollo**

Los algoritmos que usan como modelo de regresión lineal o regresión logística ( Linear Regression, Decision Tree Regresor, Support Vector Regression-SVR, La regresión lineal permitió crear un modelo de predicción basado en las variables, (Chanchí Golondrino, Campo Muñoz, & Sierra Martínez, 2020); aquí se va predecir cuántos kilómetros puede correr un automóvil por galón de combustible.

**Primero se lee el conjunto de datos**

El lenguaje que se utiliza en Python un lenguaje de alto nivel, de código abierto denominada Python software foundation license está orientado a objetos, se puede realizar cualquier tipo de programas.

**Un concepto referente a Python nos dice:**

Es un lenguaje multiparadigma en el que conviven de forma nativa aspectos imperativos, funcionales y orientados a objetos. Estos paradigmas están muy bien desacoplados, lo que permite que la entrada al lenguaje se pueda hacer de forma progresiva empezando, por ejemplo, con un estilo imperativo e incluyendo posteriormente elementos funcionales y orientados a objetos (Troyano et al., 2018, p. 224).

Python es realmente un lenguaje de propósito general y ha ido ganando popularidad en varios ámbitos como el desarrollo rápido de aplicaciones web, administración de sistemas, ciencia de datos, computación científica (donde domina con diferencia), inteligencia artificial, internet de las cosas, etc. (Carlos & Monsálvez, 2017).

Srinath (2017) nos dice que, Python es un lenguaje de programación que tiene muchas herramientas de aprendizaje que brinda diversos recursos que hacen que su uso sea más sencillo animando a las personas que lo usan seguir practicándolo. Ya que es un software muy sencillo y sofisticado de utilizar. (p.355).

En los estudios comparativos (Grandell, 2006; Ateeq, 2014; Koulori, 2014; Mészárosóvá, 2015; Peña, 2015b), las principales características por las que destacan a Python son: Sintaxis simple, alta legibilidad (sagrado obligatorio), entorno amigable de desarrollo (intérprete interactivo), abstracciones de más alto nivel (mayor nivel de expresividad), potente librería estándar y gran cantidad de módulos de terceros (actualmente son más de 100.000), multiparadigma (imperativo, POO y funcional), disponibilidad de recursos educativos abiertos, software libre y comunidad entusiasta.

## **Que es Dataset**

Se utilizo Dataset, es un término extranjero que se ha incorporado en la lengua hispana que es conjuntos de datos siendo una colección de datos que puede tabular, aquí se obtiene todos los valores que se van a utilizar como son la altura y el peso de un objeto, que corresponden a cada miembro del conjunto de datos. Cada uno de estos valores se conoce con el nombre de dato. El conjunto de datos puede incluir datos para uno o más miembros en función de su número de filas (Tandon et al., 2020).

## **Google Colaboratory**

Es un entorno que facilita al investigador ejecutar programas en Python desde el navegador de preferencia, también se lo conoce como cuaderno de trabajo que permite escribir código y ejecutarlos.

Talens (2020) hace la comparación entre Kaggle y Google Colab donde asegura que las dos plataformas disponen de servidores para ejecutar, pero la característica que resalta la comparación realizada es que los cuadernos que se crean se quedan almacenados en la cuenta Google, estando a disposición de manera instantánea.

## **Google Drive**

El concepto de Google Drive dice que: Es una aplicación que forma parte de la plataforma gratuita Google Apps; está disponible en la red, localizada en la nube y tiene un espacio de almacenamiento variable desde 5 GB, con la posibilidad de aumentarlo a 15 GB si se dispone de una cuenta de Gmail, e incluso se puede obtener más espacio si se abona una cantidad mensual. (Ceballos, 2020, p. 16).

En este problema de regresión se busca predecir la salida de un valor continuo como la probabilidad de un precio.

## **Machine Learning**

Dando a conocer la definición es un brazo de la inteligencia artificial donde su propósito

es crear técnicas de desarrollo para que las computadoras aprendan, obteniendo experiencia involucrándose con el manejo de los datos de gran magnitud. Un concepto de, Sandoval (2018) dice que “Es una rama de la Inteligencia Artificial que se encarga de generar algoritmos que tienen la capacidad de aprender y no tener que programarlos de manera explícita” (p. 37).

Usando machine Learning en el campo de la medicina en las últimas décadas se han adaptado con la tardea de aprendizaje y predicción para encontrar soluciones en diversos casos siendo precedentes de la inteligencia artificial aplicando esta herramienta. Una definición en base a machine Learning dice que:

Es la ciencia de los algoritmos que se encarga de darle sentido a los datos. Este concepto es adecuado indicarlo pues cada vez estamos rodeados de más información de la que creemos, algo que si sabemos procesarlo y hacemos un adecuado uso de los algoritmos de autoaprendizaje nos puede ayudar a identificar patrones y con ello clasificar o predecir partiendo de nuevos conjuntos de datos que no han sido tratados previamente. (Zamorano, 2018, p. 14

Su finalidad como herramienta es perfeccionar el análisis de datos, en base a predicciones, acentuando implementación es de sistemas o sencillamente mejorarlos aplicando el uso de los diversos algoritmos existentes evaluando los que se acoplen de manera optimo con el sistema, modelo que se desea trabajar.

Hay varias definiciones que describen el aprendizaje automatizado en este caso tomando en cuenta lo que nos dice López (2018) que este “Se basa en conceptos y resultados de muchos campos, incluidas la estadística, la inteligencia artificial, la filosofía, la teoría de la información, la biología, la ciencia cognitiva, la complejidad computacional y la teoría del control” (p. 10).

### **Historia de machine Learning**

Machine Learning aparece en el año 1943 donde el matemático Walter Pitts y Warren

McCulloch, neurofisiólogo dieron a conocer su trabajo que en la actualidad se conoce como inteligencia artificial. Ellos proponían en su teoría investigar y analizar el mecanismo funcional de un cerebro de computadora y en cuanto a creación de ordenadores que se ejecuten de igual manera que la red neuronal propia.

Es ahí donde se demostró que se podía conocer e incentivar al conocimiento de que existía algo inteligente, pero desde una máquina, mostrando algunas de sus repercusiones. En el año 1950 Alan Mathison Turing, científico informático, matemático, deportista y filósofo nacido en Maida Vale del Reino Unido, reconocido por crear “Test de Turing”, que básicamente lo que hacía era medir el nivel de inteligencia de una computadora, respondiendo una conversación de manera natural simulando al ser humano en su conducta.

En 1952, Arthut Samuel docente e informático mostró el primer programa que podía aprender era un juego de damas, podía guardar información y los diferentes estilos del juego permitía mejorar la respuesta de juego. En 1956 entre un grupo de profesionales que dieron una conferencia científica en Darthmouth dieron el nombre “Artificial Inteligencia”, liderado por Martin Minsky y John McCarthy. Luego de esto fue olvidada esta rama de la ciencia ya que era muy costosa, a principio de la década de los 70.

En 1979 lograron crear un robot llamado “Stanford Car” que evade obstáculos en una habitación. Luego en 1981 Gerald Dejong presento modelos de software que trabajaba en concepto a “Explanación Base Learning” este permitía el acceso a variables que se ingresaban y empezaron a tomar fuerza en la industria.

## **Metodología**

### **Preparación del modelo**

Se usa el conjunto de datos Auto MPG, que registra datos reales de varios indicadores de rendimiento del vehículo y otros factores como la cantidad de cilindros, peso, potencia, etc. Vea los 5 elementos principales del conjunto de datos, como se muestra en la Tabla 6.1, donde

se enumera el significado de cada campo En la tabla 6.2. Además del campo numérico de origen que indica la categoría, los otros campos son tipos numéricos. Para el lugar de origen, 1 significa Estados Unidos, 2 significa Europa y 3 significa Japón.

Tabla 1  
Datos de auto mpg

MPG	Cylinders	Displacement	Horsepower	Weight	Acceleration	Model Year	Origin
393	27.0	4	140.0	86.0	2790.0	15.6	82 1
394	44.0	4	97.0	52.0	2130.0	24.6	82 2
395	32.0	4	135.0	84.0	2295.0	11.6	82 1
396	28.0	4	120.0	79.0	2625.0	18.6	82 1
397	31.0	4	119.0	82.0	2720.0	19.4	82

Datos de la investigación

La categoría y el número de la función se muestran como un valor nulo (NaN).

Dataset. Isna(). Sum() # Cuente el número de valores perdidos

MPG 0

Cylinders 0

Displacement 0

Horsepower 6

Weight 0

Acceleration 0

Model Year 0

Origin 0

dtype: int64

1535atasset = 1535atasset.dropna() # Eliminar valores perdidos

1535atasset.tail()

Tabla 2  
Eliminar valores perdidos

MPG	Cylinders	Displacement	Horsepower	Weight	Acceleration	Model Year	Origin
393	27.0	4	140.0	86.0	2790.0	15.6	82 1
394	44.0	4	97.0	52.0	2130.0	24.6	82 2
395	32.0	4	135.0	84.0	2295.0	11.6	82 1
396	28.0	4	120.0	79.0	2625.0	18.6	82 1
397	31.0	4	119.0	82.0	2720.0	19.4	82

Se utiliza la biblioteca seaborn para dibujar una imagen y dibujar la imagen de la variable especificada. `Diag_kind = 'kde'` es la configuración de la imagen. Si es un gráfico univariado (en comparación con usted), es un gráfico lineal, y si es una comparación de diferentes variables, es un gráfico de dispersión.

### Resultados

Figura 1  
Regresión lineal

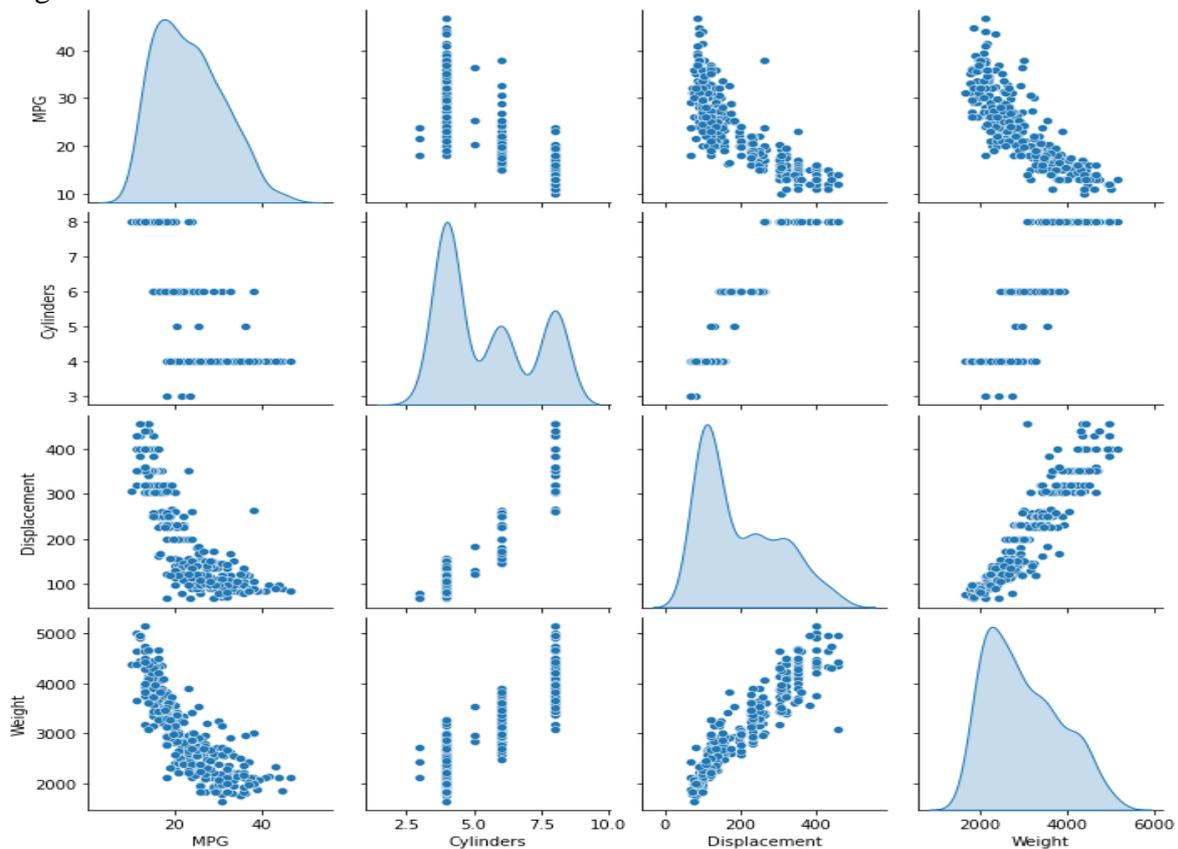


Tabla 3  
Parámetros de estadística

	count	mean	std	min	25%	50%	75%	max
<b>Cylinders</b>	314.0	5.477707	1.699788	3.0	4.00	4.0	8.00	8.0
<b>Displacement</b>	314.0	195.318471	104.331589	68.0	105.50	151.0	265.75	455.0
<b>Horsepower</b>	314.0	104.869427	38.096214	46.0	76.25	94.5	128.00	225.0
<b>Weight</b>	314.0	2990.251592	843.898596	1649.0	2256.50	2822.5	3608.00	5140.0
<b>Acceleration</b>	314.0	15.559236	2.789230	8.0	13.80	15.5	17.20	24.8
<b>Model Year</b>	314.0	75.898089	3.675642	70.0	73.00	76.0	79.00	82.0
<b>USA</b>	314.0	0.624204	0.485101	0.0	0.00	1.0	1.00	1.0
<b>Europe</b>	314.0	0.178344	0.383413	0.0	0.00	0.0	0.00	1.0
<b>Japan</b>	314.0	0.197452	0.398712	0.0	0.00	0.0	0.00	1.0

Como se muestra en la tabla 3 se usa la función describe para ver el conjunto de datos para ver los parámetros de algunos conjuntos de datos con más detalle, de modo que no necesitemos llamar a la función para calcular por nosotros mismos.

**Inspeccione el modelo**

Model: "sequential\_12"

Layer (type)	Output Shape	Param #
dense_42 (Dense)	(None, 64)	640
dense_43 (Dense)	(None, 64)	4160
dense_44 (Dense)	(None, 1)	65

Total params: 4,865  
Trainable params: 4,865  
Non-trainable params: 0

```
Ahora tome otro modelo
example_batch = normed_train_data[:10]
example_result = model.predict(example_batch)
print(example_result)
```

[-0.26191565]  
[-0.06376351]  
[ 0.5379154 ]  
[-0.16877869]  
[-0.05534531]  
[-0.06515738]  
[-0.08105768]  
[-0.07248831]  
[-0.01380318]  
[-0.09675659]]

**Entrenar el modelo**

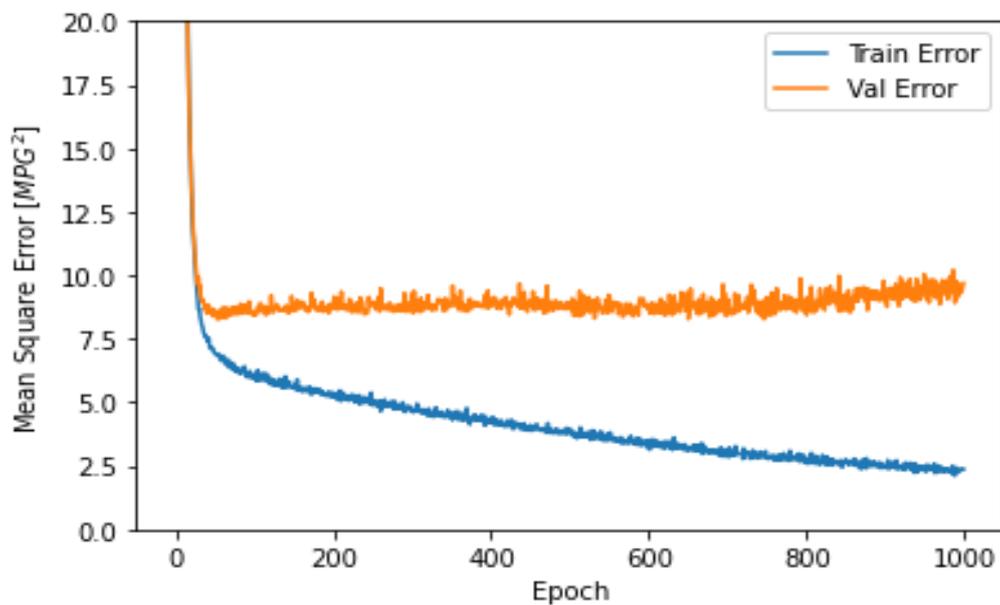
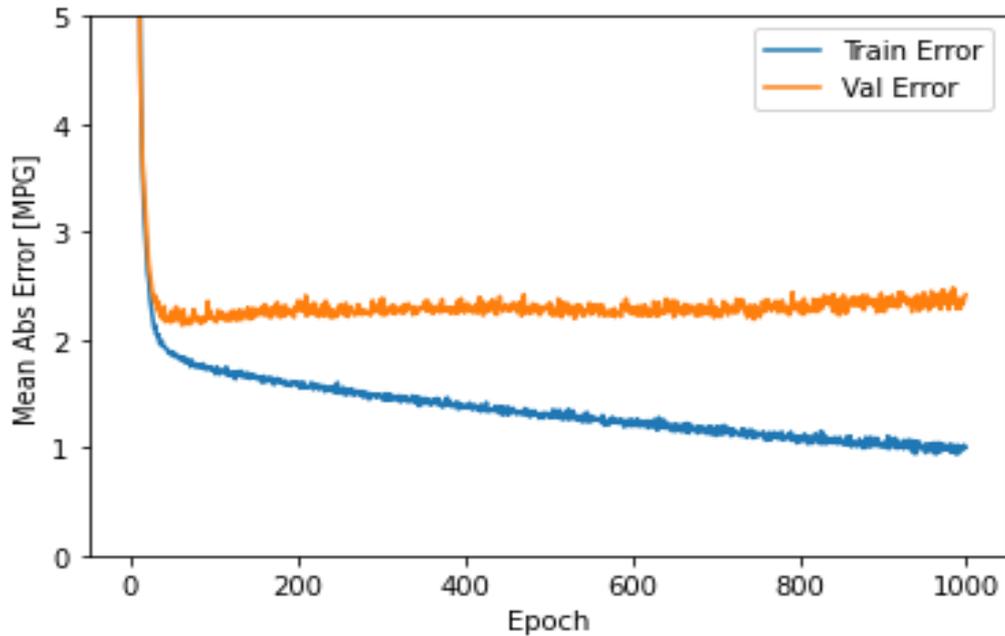
```
history=model.fit(normed_train_data,train_labels,  
epochs=1000,  
validation_split=0.2, # Utilice el 20% del conjunto de formación como conjunto de validación  
verbose=0, # No muestra la barra de progreso  
callbacks=[PrintDot()]) # La función de devolución de llamada es PrintDot
```

**Visualice el proceso**

hist.tail()

loss	mae	mse	val_loss	val_mae	val_mse	epoch	
995	2.595970	0.978913	2.595970	11.760169	2.524347	11.760169	995
996	2.479838	0.976182	2.479838	11.067865	2.483008	11.067865	996
997	2.674930	1.019684	2.674930	11.104455	2.580837	11.104455	997
998	2.610440	0.966319	2.610440	10.906663	2.532035	10.906663	998
999	2.501746	1.019541	2.501746	10.737952	2.528607	10.737952	999

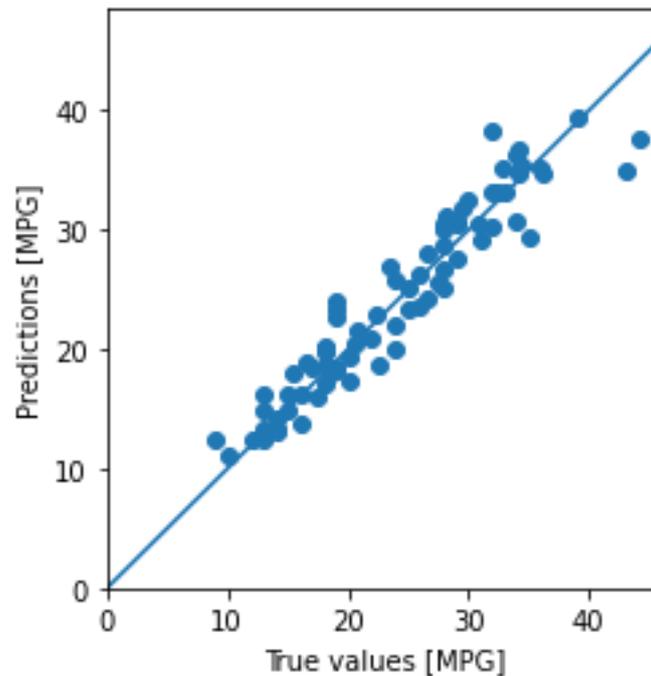
```
ef plot_history(history):  
    hist = pd.DataFrame(history.history)  
    hist['epoch'] = history.epoch  
    plt.figure()  
    plt.xlabel('Epoch')  
    plt.ylabel('Mean Abs Error [MPG]')  
    plt.plot(hist['epoch'], hist['mae'], # (x,y  
    label='Train Error') # El nombre del segmento de línea es el nombre en la tarjeta de etiqueta  
    It.plot(hist['epoch'], hist['val_mae'],  
    label = 'Val Error')  
    plt.ylim([0,5])  
    plt.legend()  
    plt.figure()  
    plt.xlabel('Epoch')  
    plt.ylabel('Mean Square Error [MPG^2$]')  
    plt.plot(hist['epoch'], hist['mse'],  
    label='Train Error')  
    plt.plot(hist['epoch'], hist['val_mse'],  
    label = 'Val Error')  
    plt.ylim([0,20])  
    plt.legend() # Imprimir tarjeta de etiqueta  
    plt.show()  
    plot_history(history) # El gráfico muestra que después de aproximadamente 100 épocas, el er  
    ror no ha mejorado, sino que ha empeorado.
```



Este gráfico muestra poca mejora, o incluso degradación en el error de validación después de aproximadamente 100 épocas. Actualicemos la llamada `model.fit` para detener automáticamente el entrenamiento cuando el puntaje de validación no mejore. Utilizaremos una \* devolución de llamada de `EarlyStopping` \* que pruebe una condición de entrenamiento para cada época. Si transcurre una cantidad determinada de épocas sin mostrar mejoría, entonces detiene automáticamente el entrenamiento.

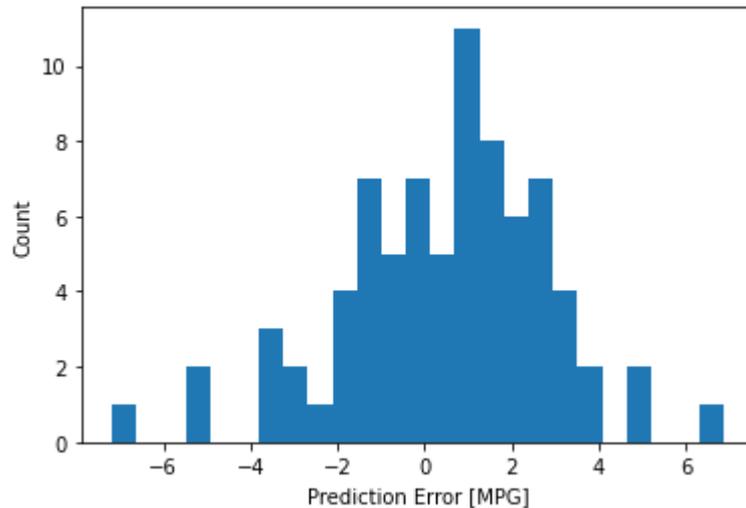
```
plt.scatter(test_labels, test_predictions)
```

```
plt.xlabel("True values [MPG]")
plt.ylabel("Predictions [MPG]")
plt.axis('equal') # proporción
plt.axis('square')
plt.xlim([0,plt.xlim()[1]])
plt.ylim([0,plt.ylim()[1]])
_ = plt.plot([-100,100],[-100,100]) # Dibuja un segmento de línea que pase por (-100, -
100) y (100, 100)
```



Parece que nuestro modelo predice razonablemente bien. Echemos un vistazo a la distribución de errores.

```
error=test_predictions-test_labels
plt.hist(error,bins=25)
plt.xlabel("PredictionError[MPG]")
_ = plt.ylabel("Count")
```



## Conclusiones

En esta investigación se introdujo algunas técnicas para manejar un problema de regresión.

- El error cuadrático medio (MSE) es una función de pérdida común utilizada para problemas de regresión (se utilizan diferentes funciones de pérdida para problemas de clasificación).
- Del mismo modo, las métricas de evaluación utilizadas para la regresión difieren de la clasificación. Una métrica de regresión común es el error absoluto medio (MAE).
- Cuando las características de datos de entrada numéricos tienen valores con diferentes rangos, cada característica debe escalarse independientemente al mismo rango.
- Si no hay muchos datos de entrenamiento, una técnica es preferir una red pequeña con pocas capas ocultas para evitar el sobreajuste.
- La detención temprana es una técnica útil para evitar el sobreajuste.

## Referencias bibliográficas

Álvarez Cedillo, J. A., Álvarez Sánchez, T., Sandoval Gómez, R. J., & Aguilar Fernández, M. (2019). La exploración en el desarrollo del aprendizaje profundo / The Exploration in the Develop Deep Learning. RIDE Revista Iberoamericana Para La Investigación y El Desarrollo Educativo, 9(18). <https://doi.org/10.23913/ride.v9i18.474>

Aldas, D., Reyes, J., Morales, L., Nuñez, J., & Toaza, B. (2018). Impacts Analysis Towards a Sustainable Urban Public Transport System. International Conference on Operations Research and Enterprise Systems (ICORES).

API design for machine learning software: experiences from the scikit-learn project, Buitinck et al., 2013.

Árboles de decisión con Python: regresión y clasificación by Joaquín Amat Rodrigo, available under a Attribution 4.0 International (CC BY 4.0) at [https://www.cienciadedatos.net/documentos/py07\\_arboles\\_decision\\_python.html](https://www.cienciadedatos.net/documentos/py07_arboles_decision_python.html)

Cabanelas Omil, J. (2019). Inteligencia artificial ¿Dr. Jekyll o Mr. Hyde? Mercados y Negocios: Revista de Investigación y y Análisis, 0(40).

CAF, B. d. (2017). Banco de Desarrollo de América Latina. Recuperado el 23 de Febrero de 2018, de <https://www.caf.com/es/actualidad/noticias/2017/10/mejorar-lamovilidad-en-las-ciudades-coadyuvara-al-desarrollo-de-americalatina/?parent=14062>

Campos, Á. (2019). Análisis de datos financieros con técnicas de Machine Learning. <https://repositorio.comillas.edu/xmlui/bitstream/handle/11531/32146/TFGCamposMartAn%20Alvaro.pdf?sequence=1&isAllowed=y>

Carlos, J., & Monsálvez, G. (2017). Python como primer lenguaje de programación textual en la Enseñanza Secundaria Python as First Textual Programming Language in Secondary Education. 18, 147–162.

Ceballos, J. (2020). Google Drive como entorno virtual de investigación científica para el desarrollo de la escritura académica. Universidad Privada Del Estado de México Instituto Americano Cultural S. C., 75, 14–21.

Chanchí Golondrino, G. E., Campo Muñoz, W. Y., & Sierra Martínez, L. M. (2020). Aplicación de la regresión polinomial para la caracterización de la curva del COVID-19, mediante técnicas de machine Learning.

Dividamos la data de entrenamiento y prueba. [https://www.tensorflow.org/tutorials/keras/regression?hl=es-419#dividamos\\_la\\_data\\_en\\_entrenamiento\\_y\\_prueba](https://www.tensorflow.org/tutorials/keras/regression?hl=es-419#dividamos_la_data_en_entrenamiento_y_prueba)

El Modelo [https://www.tensorflow.org/tutorials/keras/regression?hl=es-419#construye\\_el\\_modelo](https://www.tensorflow.org/tutorials/keras/regression?hl=es-419#construye_el_modelo)

Google, «Google Drive: almacenamiento de archivos en línea para empresas | Google Workspace», 2012. [https://workspace.google.com/intl/es419/products/drive/?utm\\_source=google&utm\\_medium=m=cpc&utm\\_campaign=latam-T1-all-es-dr-bkws-allall-trial-e-dr-1009897-LUAC0012558&utm\\_content=text-ad-none-anyDEV\\_c-CRE\\_479487543818-ADGP\\_Hybrid%7CBKWS-MIX%7C Txt~](https://workspace.google.com/intl/es419/products/drive/?utm_source=google&utm_medium=m=cpc&utm_campaign=latam-T1-all-es-dr-bkws-allall-trial-e-dr-1009897-LUAC0012558&utm_content=text-ad-none-anyDEV_c-CRE_479487543818-ADGP_Hybrid%7CBKWS-MIX%7C Txt~) (accedido feb. 27, 2021).

- Grandell, L., Peltomäki, M., Back, R. J., & Salakoski, T. (2006). Why complicate things?: introducing programming in high school using Python. In Proceedings of the 8th Australasian Conference on Computing Education-Volume 52 (pp. 71-80). Australian Computer Society, Inc.  
<https://doi.org/10.29057/icbi.v7iEspecial.4785>
- Kampelis, N., Tsekeri, E., Kolokotsa, D., Kalaitzakis, K., Isidori, D., & Cristalli, C. (2018). Development of demand response energy management optimization at building and district levels using genetic algorithm and artificial neural network modelling power predictions. *Energies*, 11(11). <https://doi.org/10.3390/en11113012>
- López, A. (2018). Fundamentos Matemáticos de los Métodos Kernel para Aprendizaje Supervisado. 73.
- Perez, D. P., Bustillos, R. S., Botto-Tobar, M., & Mora, C. M. (2021). X-Ray Images Analysis by Medium Artificial Neural Network. *Ecuadorian Science Journal*, 5(1), 55–60. <https://doi.org/10.46480/esj.5.1.50>
- Salvador, V. L., Mamaqi, X., & Bordes, J. V. (2020). Artificial intelligence: Theoretical, formative and communicative challenges of datification. In *Icono14* (Vol. 18, Issue 1). <https://doi.org/10.7195/RI14.V18I1.1434>
- Sandoval, L. (2018). Algoritmos de aprendizaje automático para análisis y predicción de datos. 11, 36–40.  
[http://www.redicces.org.sv/jspui/bitstream/10972/3626/1/Art6\\_RT2018.pdf](http://www.redicces.org.sv/jspui/bitstream/10972/3626/1/Art6_RT2018.pdf)
- Sepúlvera, G., Vega-Alvarado, E., & Portilla-Flores, E. A. (2019). Machine Learning para Robots, del Entrenamiento Virtual a la Tarea Real. *Pädi Boletín Científico de Ciencias Básicas e Ingenierías Del ICBI*, 7(Especial), 14–18.
- R. Joaquín Amat, «Correlación lineal y Regresión lineal simple», 2016.  
[https://www.cienciadedatos.net/documentos/24\\_correlacion\\_y\\_regresion\\_lineal.html](https://www.cienciadedatos.net/documentos/24_correlacion_y_regresion_lineal.html) (accedido feb. 26, 2021).
- Talens, R. (2020). La relación entre los casos de Covid-19 y su impacto en Twitter.
- Tandon, N., Mishra, B. D., Sakaguchi, K., Bosselut, A., & Clark, P. (2020). Wiqa: A dataset for “What if...” reasoning over procedural text. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*. <https://doi.org/10.18653/v1/d19-1629>
- Tu, Y. (2019). Machine learning. In *EEG Signal Processing and Feature Extraction*. [https://doi.org/10.1007/978-981-13-9113-2\\_15](https://doi.org/10.1007/978-981-13-9113-2_15)

- Xin, Y., 2018. Machine Learning and Deep Learning Methods for Cybersecurity, IEEE Access.
- Zamorano, J. (2018). Comparativa Y Análisis De Algoritmo
- Zheng, B., & Zhang, J. (2019). Finite Element Analysis of the Brake Drum Based on ANSYS APDL. IOP Conference Series: Materials Science and Engineering, 472(1). <https://doi.org/10.1088/1757-899X/472/1/012081>
- Zou, D., Cao, Y., Zhou, D., & Gu, Q. (2020). Gradient descent optimizes overparameterized deep ReLU networks. Machine Learning, 109(3). <https://doi.org/10.1007/s10994-019-05839>