

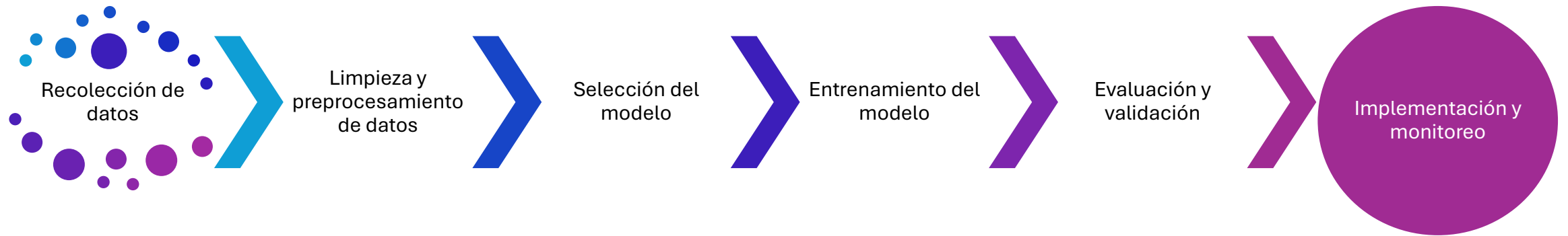
# Machine Learning



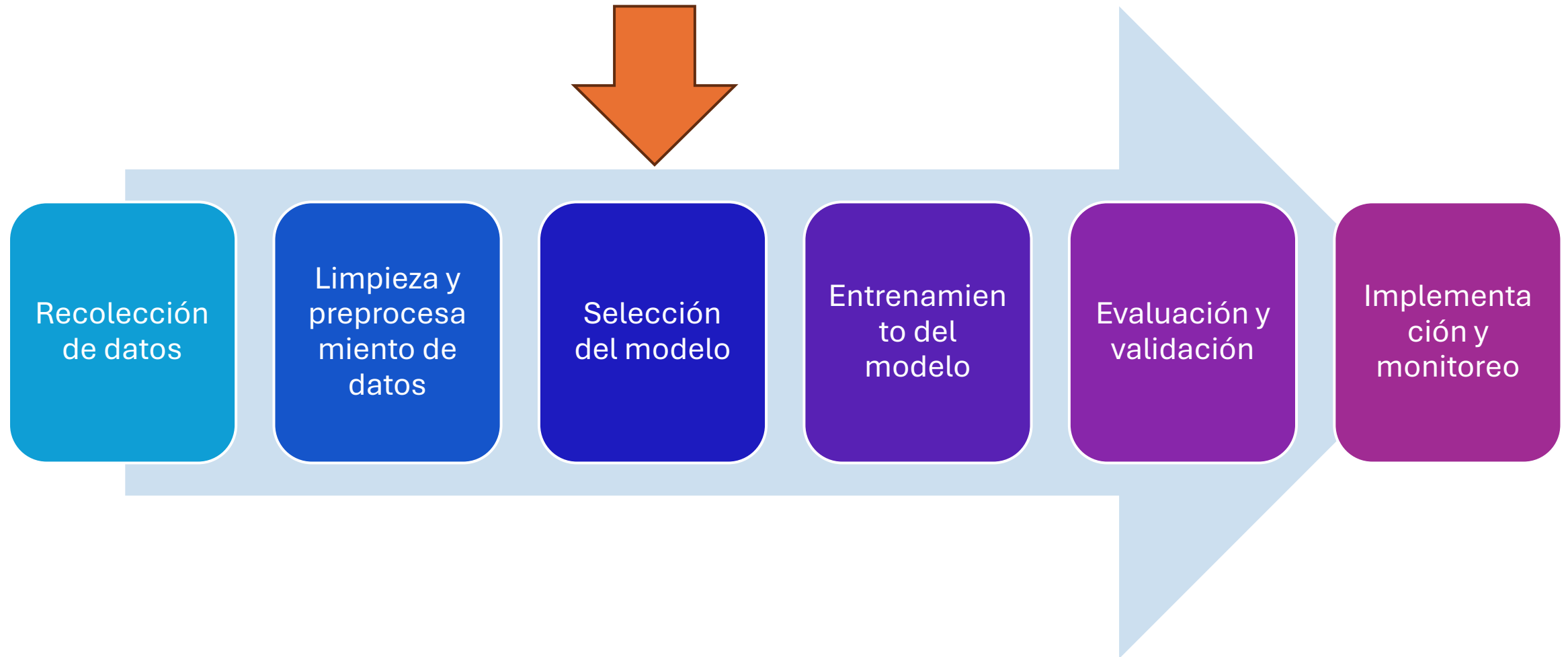
**Susana Medina Gordillo**

[susana.medina@correounivalle.edu.co](mailto:susana.medina@correounivalle.edu.co)

# Flujo de trabajo en Machine Learning



# Flujo de trabajo en Machine Learning



# **Técnicas de Aprendizaje Supervisado**

# Regresión Lineal

*Breve introducción al concepto de regresión y su importancia en el aprendizaje supervisado*

# Regresión Lineal Simple: definición

La regresión lineal es una **técnica estadística** que se usa para encontrar la **relación** entre las **variables**.

En un contexto de ML, la regresión lineal encuentra la relación entre **atributos** y una **etiqueta**.



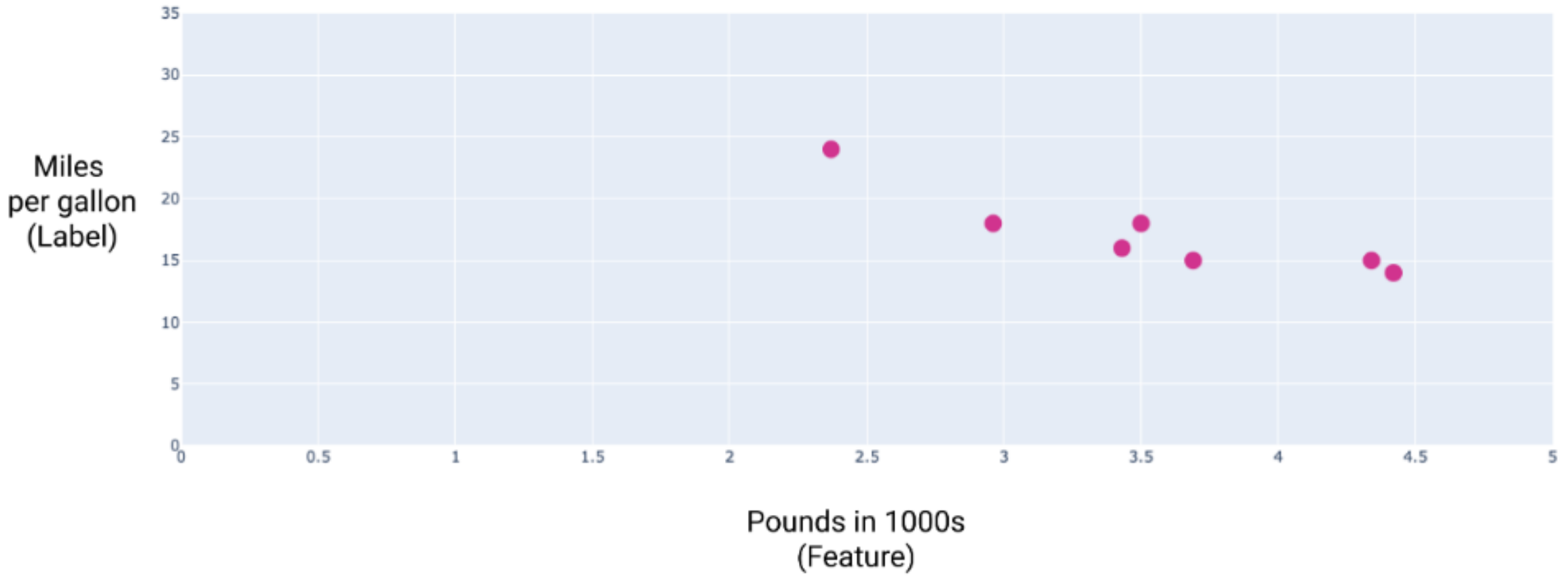
# Regresión Lineal Simple

## Ejemplo

Queremos **predecir** la eficiencia de combustible de un automóvil en millas por galón según su peso y tenemos el siguiente conjunto de datos:

| Libras en miles (función) | Millas por galón (etiqueta) |
|---------------------------|-----------------------------|
| 3.5                       | 18                          |
| 3.69                      | 15                          |
| 3.44                      | 18                          |
| 3.43                      | 16                          |
| 4.34                      | 15                          |
| 4.42                      | 14                          |
| 2.37                      | 24                          |

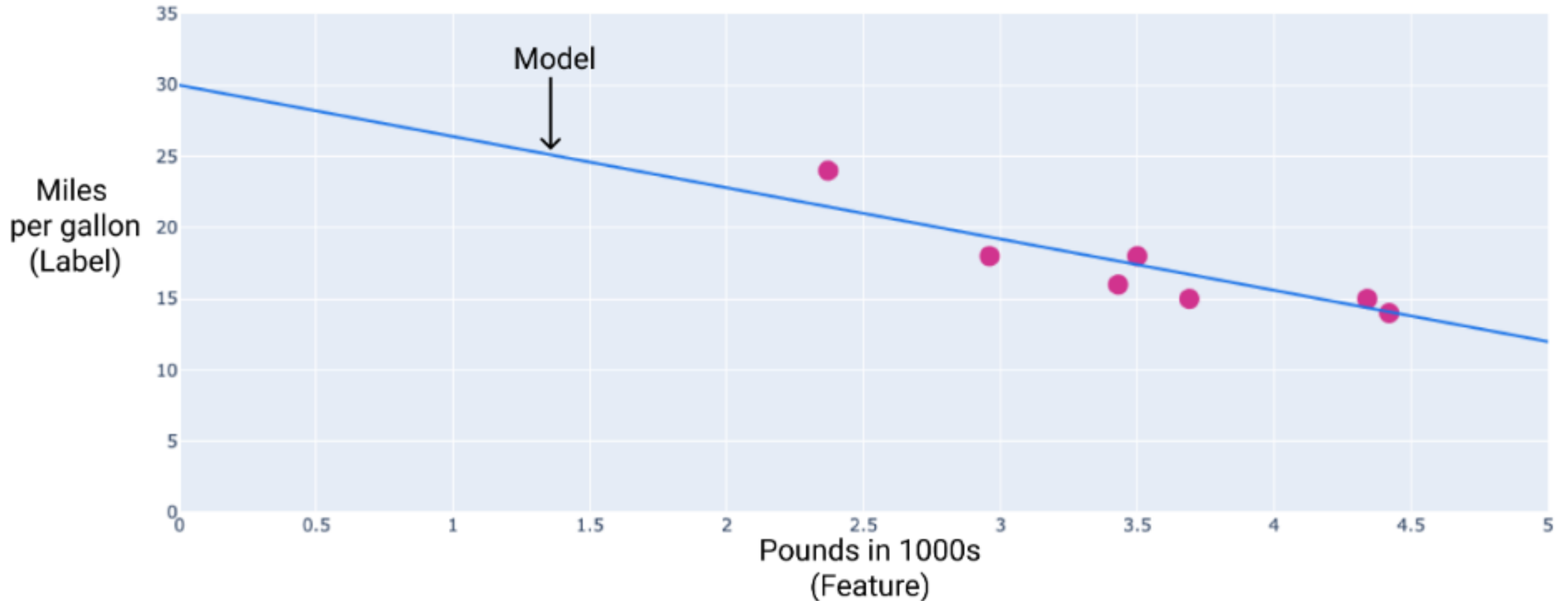
# Regresión Lineal Simple



**Figura 1.** Peso del automóvil (en libras) en comparación con la clasificación de millas por galón A medida que un automóvil se hace más pesado, su clasificación de millas por galón suele disminuir.



# Regresión Lineal Simple



**Figura 2.** Una línea de mejor ajuste trazada a través de los datos de la figura anterior.

# Ecuación de la regresión lineal

En términos algebraicos, el modelo se definiría como  $y = mx + b$ , donde:

$y$  son millas por galón, el valor que queremos predecir.

$m$  es la pendiente de la línea.

$x$  son libras, nuestro valor de entrada.

$b$  es la intersección en  $y$ .

# Ecuación de la regresión lineal

$$y' = b + w_1 x_1$$

Prediction      Bias      Weight      Feature value

Calculated from training

# Ecuación de la regresión lineal en Aprendizaje Automático

Donde:

$y'$  es la etiqueta predicha, el resultado.

$b$  es el **sesgo** del modelo. El sesgo es el mismo concepto que el eje Y en la ecuación algebraica de una línea. En el AA, a veces se hace referencia al sesgo como  $w_0$ . El sesgo es un **parámetro** del modelo y se calcula durante el entrenamiento.

$w_1$  es el peso de la función. El peso es el mismo concepto que la pendiente  $m$  en la ecuación algebraica de una línea. El peso es un **parámetro** del modelo y se calcula durante el entrenamiento.

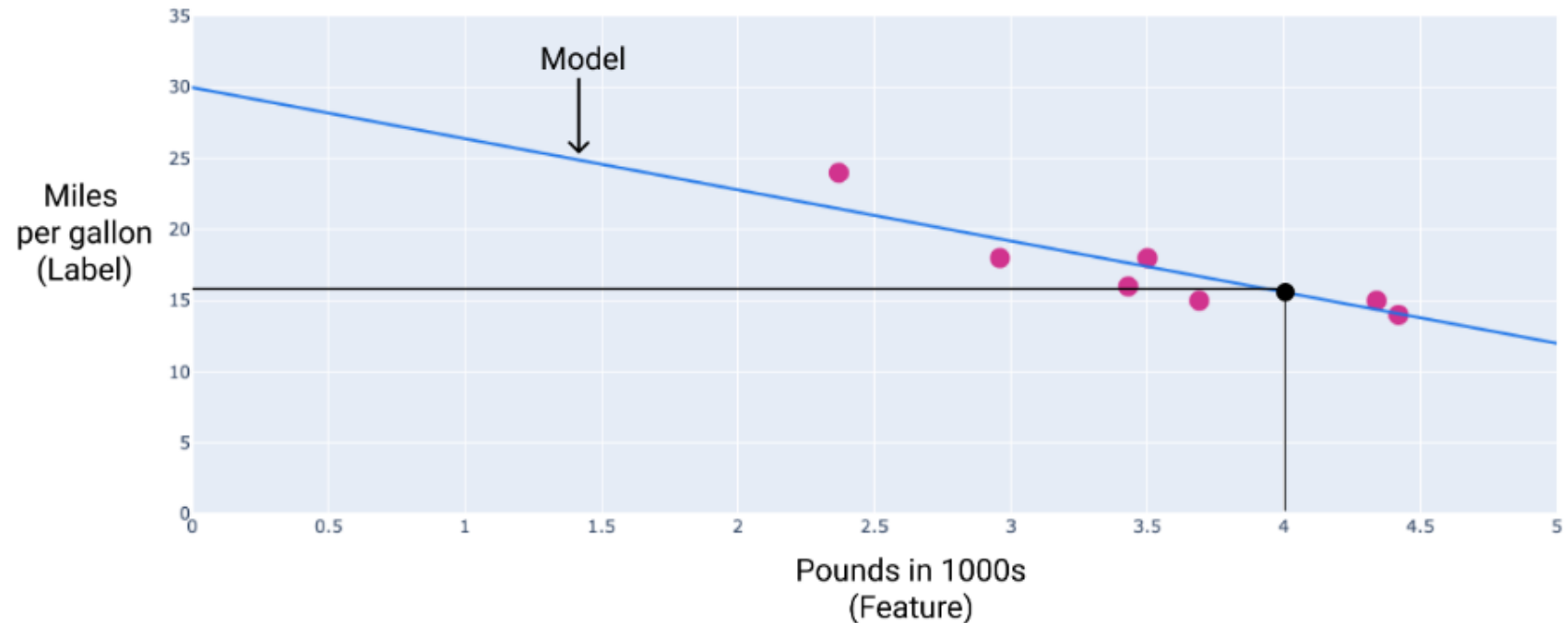
$x_1$  es un **atributo**: la entrada.

# Regresión Lineal Simple

Calculamos el peso y el sesgo de la línea que dibujamos. El sesgo es 30 (donde la línea se cruza con el eje Y) y el peso es -3.6 (la pendiente de la línea).

El **modelo** se definiría como  
$$y' = 30 + (-3.6)(x_1)$$
y podríamos usarlo para hacer predicciones.

Por ejemplo, con este modelo, un automóvil de 1,814 kg tendría una eficiencia de combustible prevista de 25.1 km por galón.



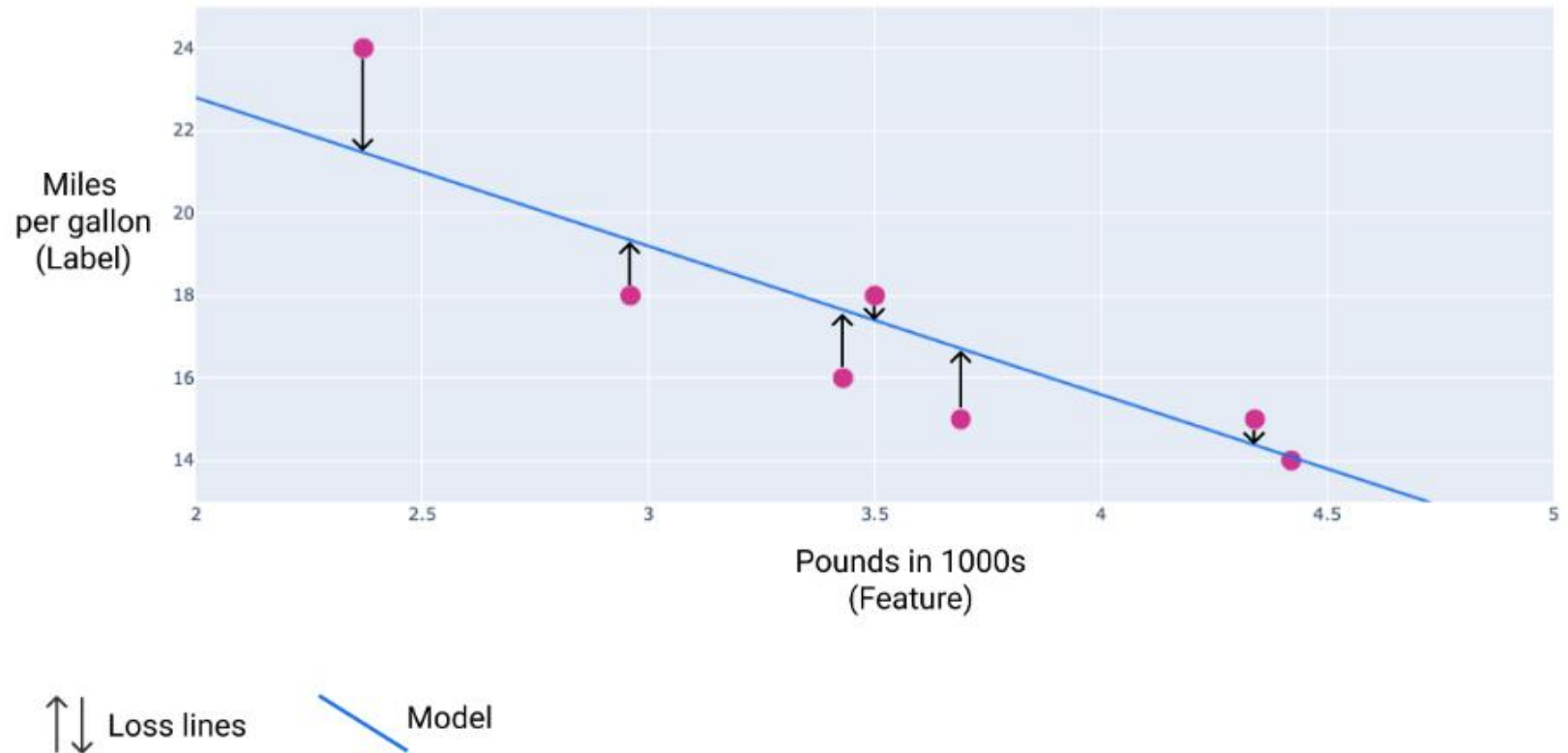
**Figura 3.** Con el modelo, un automóvil de 1,814 kg tiene una eficiencia de combustible prevista de 25.1 km por galón.

# Regresión Lineal: Pérdida

La **Pérdida** es una métrica numérica que describe cuán incorrectas son las predicciones de un modelo

La pérdida mide la **distancia** entre las predicciones del modelo y las etiquetas reales.

El objetivo de entrenar un modelo es **minimizar la pérdida** y reducirla al valor más bajo posible.



**Figura 4.** La pérdida se mide del valor real al valor previsto.

# Tipos de Pérdidas

| Tipo de pérdida              | Definición   | Ecuación   |
|------------------------------|--|--|
| Pérdida de $L_1$             | Es la suma de los valores absolutos de la diferencia entre los valores predichos y los reales. | $\sum  valor\ real - valor\ previsto $               |
| Error absoluto medio (MAE)   | Es el promedio de las pérdidas de $L_1$ en un conjunto de ejemplos.                            | $\frac{1}{N} \sum  valor\ real - valor\ previsto $   |
| Pérdida de $L_2$             | Es la suma de la diferencia al cuadrado entre los valores predichos y los reales.              | $\sum (valor\ real - valor\ previsto)^2$             |
| Error cuadrático medio (ECM) | Es el promedio de las pérdidas de $L_2$ en un conjunto de ejemplos.                            | $\frac{1}{N} \sum (valor\ real - valor\ predicho)^2$ |



# Ejemplo de cálculo de Pérdidas

De la línea de ajuste óptimo, obtuvimos los siguientes valores para el peso y el sesgo:

$$\text{Peso} = -3.6$$

$$\text{Sesgo} = 30$$

Si el modelo predice que un automóvil de 1,070 kilogramos (2,370 libras) recorre 34.6 kilómetros (21.5 millas) por galón, pero en realidad recorre 38.6 kilómetros (24 millas) por galón, calcularíamos la pérdida de  $L_2$  de la siguiente manera:

| Valor         | Ecuación   | Resultado |
|---------------|--|-----------|
| Predicción    | $\text{bias} + (\text{weight} * \text{feature value})$<br>$30 + (-3.6 * 2.37)$ | 21.5      |
| Valor real    | $\text{label}$   | 24        |
| Pérdida $L_2$ | $(\text{valor real} - \text{valor previsto})^2$<br>$(24 - 21.5)^2$             | 6.25      |

**En este ejemplo, la pérdida de  $L_2$  para ese único dato es 6.25.**

# Ejercicio práctico #1

Responder  
pregunta final de la  
sección:  
Regresión Lineal –  
Pérdida

¿Cuál de los dos  
conjuntos de datos  
que se muestran en  
las representaciones  
anteriores tiene el  
error cuadrático  
medio (ECM) **más  
alto?**

## developers.google.com



### Curso intensivo de aprendizaje automático

Un curso práctico para explorar los conceptos básicos del  
aprendizaje automático.



# Ejercicio práctico #2

Regresión lineal:  
Ejercicio de  
parámetros

**Tu tarea:** Ajusta los  
controles  
deslizantes **Peso** y **S**  
**esgo** sobre el gráfico  
para encontrar el  
modelo lineal que  
minimice la pérdida  
de MSE en los datos.

developers.google.com



Curso intensivo de  
aprendizaje automático

Un curso práctico para explorar los conceptos básicos del  
aprendizaje automático.



# Hiperparámetros de la Regresión Lineal

*Define y describe cómo el descenso del gradiente encuentra los parámetros óptimos del modelo.*

*Describe cómo ajustar los hiperparámetros para entrenar un modelo lineal de manera eficiente.*

# Regresión Lineal : Hiperparámetros

Los **hiperparámetros** son variables que controlan diferentes aspectos del **entrenamiento**. Estos son tres hiperparámetros comunes:

- **Tasa de aprendizaje (learning rate)**
- **Tamaño del lote (batch size)**
- **Épocas (epoch)**

Por el contrario, los **parámetros** son las variables, como los pesos y la ordenada al origen, que forman parte del modelo. En otras palabras, los hiperparámetros son valores que controlas; los parámetros son valores que el modelo calcula durante el entrenamiento.



# Regresión Lineal : Tasa de aprendizaje

La tasa de aprendizaje es un **número de punto flotante** que estableces y que influye en la **rapidez** con la que **converge** el modelo.

Si la tasa de aprendizaje es demasiado baja, el modelo puede tardar mucho tiempo en converger.

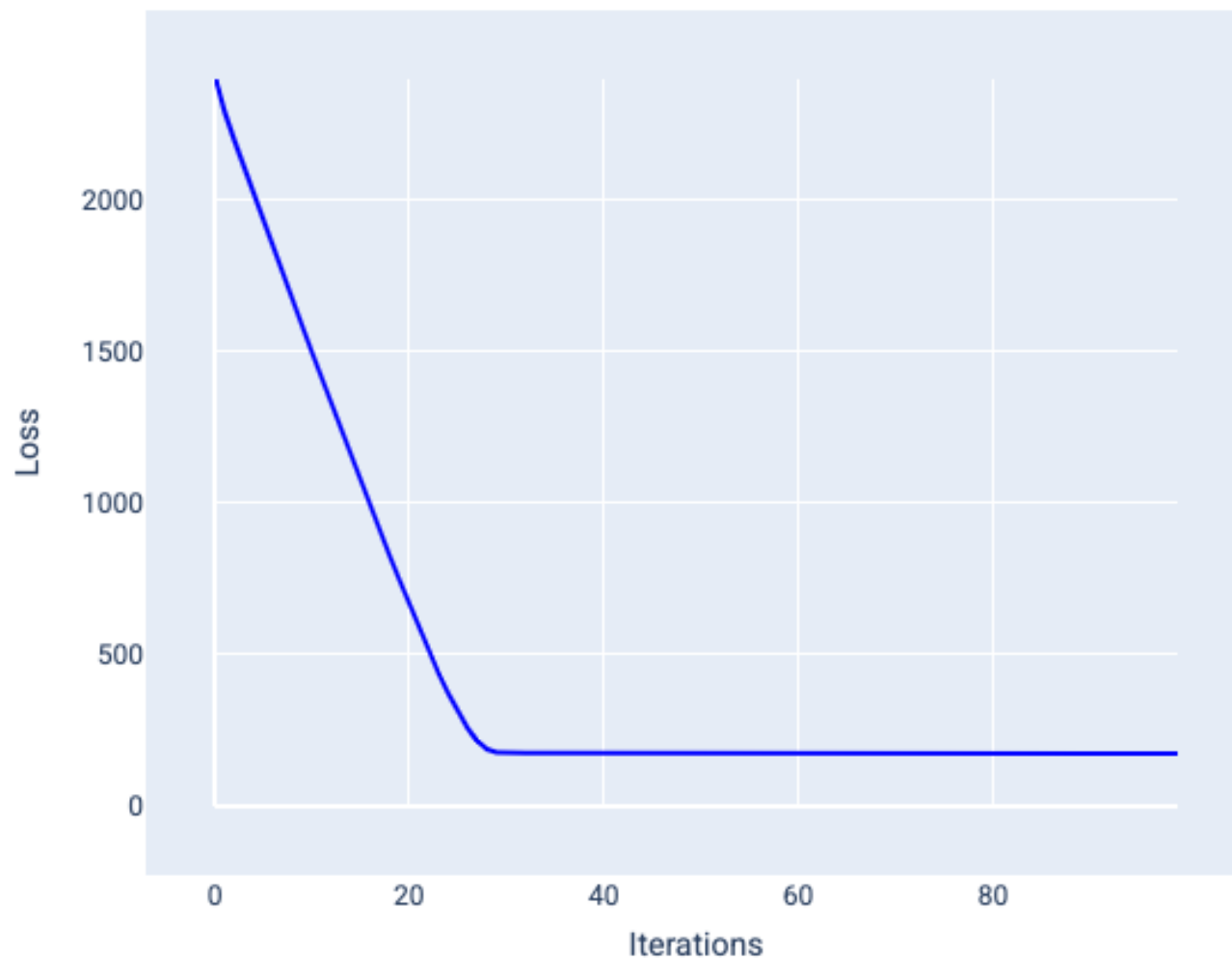
Sin embargo, si la tasa de aprendizaje es demasiado alta, el modelo nunca converge, sino que rebota alrededor de los pesos y la sesgo que minimizan la pérdida.

El objetivo es elegir una tasa de aprendizaje que no sea demasiado alta ni demasiado baja para que el **modelo converja rápidamente**.

# Tasa de aprendizaje: convergencia rápida

La tasa de aprendizaje ideal ayuda al modelo a converger dentro de una cantidad razonable de iteraciones.

En la Figura, la **curva de pérdida** muestra que el modelo mejora significativamente durante las primeras 20 iteraciones antes de comenzar a converger:

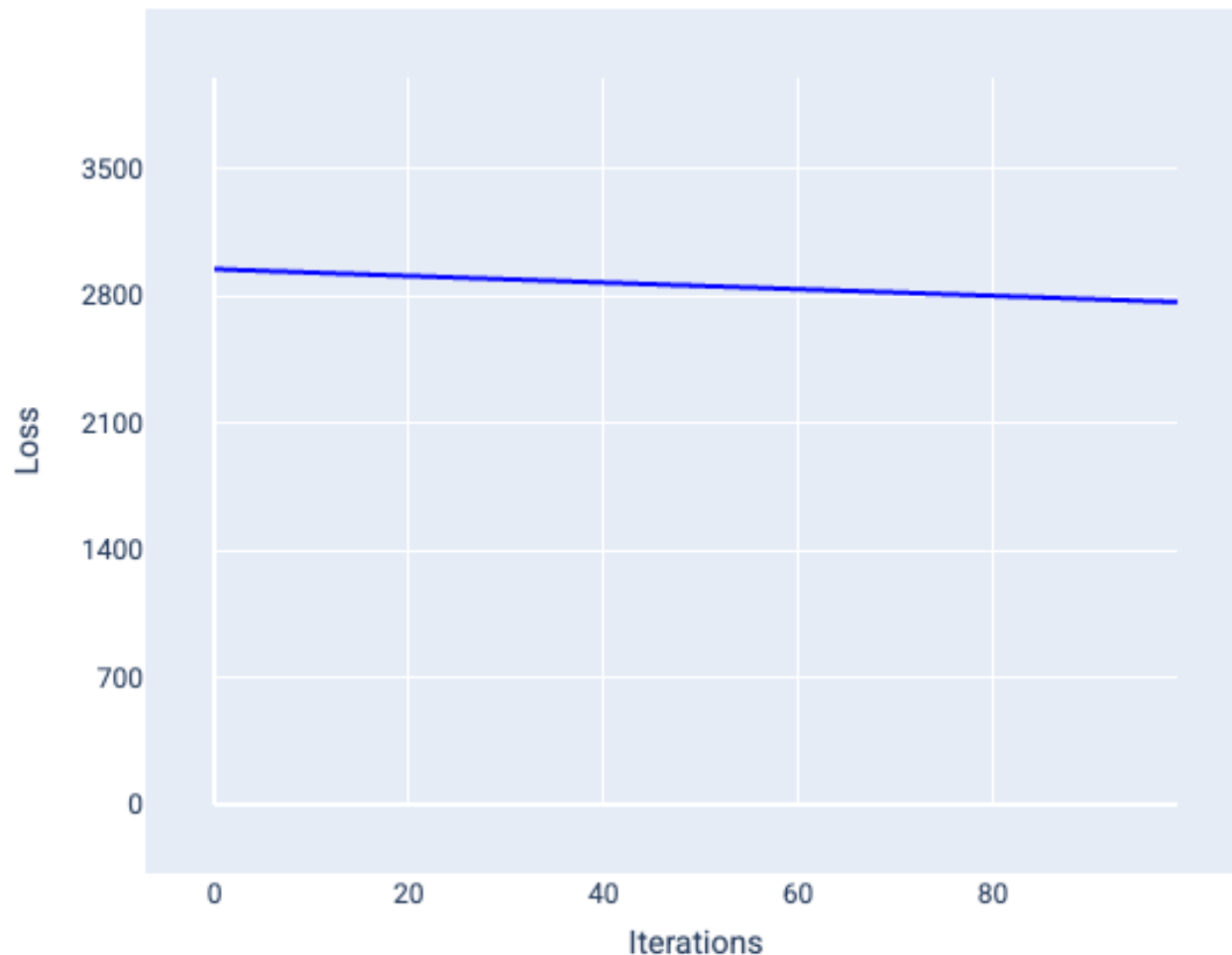




# Tasa de aprendizaje: convergencia lenta

Por el contrario, una tasa de aprendizaje demasiado **pequeña** puede tardar demasiado en converger.

En la Figura, la **curva de pérdida** muestra que el modelo solo realiza mejoras menores después de cada iteración:



# Regresión Lineal : Tamaño del lote

**Tamaño del lote** es un **hiperparámetro** que se refiere a la **cantidad de ejemplos** que el modelo procesa antes de actualizar sus pesos y sesgos.

El modelo debería calcular la pérdida para cada ejemplo del conjunto de datos antes de actualizar los pesos y la polarización. Sin embargo, cuando un conjunto de datos contiene cientos de miles o incluso millones de ejemplos, no es práctico usar el lote completo.

Dos técnicas comunes para obtener el gradiente correcto en el promedio sin necesidad de ver todos los ejemplos del conjunto de datos antes de actualizar los pesos y la polarización son:

- ✓ **El descenso estocástico del gradiente** (lote = 1)
- ✓ **El descenso estocástico del gradiente en minilotes** (lote = N)

# Regresión Lineal : Tamaño del lote (batch)

Full batch: an entire dataset

|      | Feature | Label |
|------|---------|-------|
| 0    |         |       |
| .    |         |       |
| .    |         |       |
| .    |         |       |
| .    |         |       |
| 500  |         |       |
| .    |         |       |
| .    |         |       |
| .    |         |       |
| .    |         |       |
| 1000 |         |       |

Single batch: a mini batch

|     | Feature | Label |
|-----|---------|-------|
| 1   |         |       |
| .   |         |       |
| .   |         |       |
| 100 |         |       |

One epoch: a full batch composed of ten mini batches

|     | Feature | Label |      | Feature | Label |
|-----|---------|-------|------|---------|-------|
| 1   |         |       | 501  |         |       |
| .   |         |       | .    |         |       |
| .   |         |       | .    |         |       |
| 100 |         |       | 600  |         |       |
|     |         |       |      |         |       |
| 101 |         |       | 601  |         |       |
| .   |         |       | .    |         |       |
| .   |         |       | .    |         |       |
| 200 |         |       | 700  |         |       |
|     |         |       |      |         |       |
| 201 |         |       | 701  |         |       |
| .   |         |       | .    |         |       |
| .   |         |       | .    |         |       |
| 300 |         |       | 800  |         |       |
|     |         |       |      |         |       |
| 301 |         |       | 801  |         |       |
| .   |         |       | .    |         |       |
| .   |         |       | .    |         |       |
| 400 |         |       | 900  |         |       |
|     |         |       |      |         |       |
| 401 |         |       | 901  |         |       |
| .   |         |       | .    |         |       |
| .   |         |       | .    |         |       |
| 500 |         |       | 1000 |         |       |

# Regresión Lineal : Época

Recorrido de entrenamiento completo por todo el **conjunto de entrenamiento** de manera que cada ejemplo se haya procesado una vez.

Un **ciclo de entrenamiento** representa  $N/\text{tamaño del lote}$  iteraciones, donde  $N$  es la cantidad total de ejemplos.

Por ejemplo, supongamos lo siguiente:

El conjunto de datos consta de 1.000 ejemplos.

El tamaño del lote es de 50 ejemplos.

Por lo tanto, una sola época requiere 20 iteraciones:

$$1 \text{ epoca} = (N / \text{tamaño lote}) = (1000 / 50) = 20 \text{ iteraciones}$$

# Ejercicio práctico #3

## Regresión lineal: Ejercicio de descenso de gradientes

En este ejercicio, repasarás el gráfico de datos de eficiencia de combustible del ejercicio de parámetros. Pero esta vez, usarás el descenso de gradientes para aprender los valores óptimos de peso y sesgo para un modelo lineal que minimice la pérdida.

# developers.google.com



## Curso intensivo de aprendizaje automático

Un curso práctico para explorar los conceptos básicos del aprendizaje automático.



# Regresión Múltiple

*Breve introducción al concepto de regresión y su importancia en el aprendizaje supervisado*

# Modelos con varias funciones

El ejemplo anterior solo usa un atributo (el peso del automóvil), un modelo más sofisticado podría usar varios atributos, cada uno con un peso distinto ( $w_1$ ,  $w_2$ , etc.). Por ejemplo, un modelo que se basa en **cinco atributos** se escribiría de la siguiente manera:

$$y' = b + w_1 X_1 + w_2 X_2 + w_3 X_3 + w_4 X_4 + w_5 X_5$$

Pounds   Displacement   Acceleration   Number of cylinders   Horsepower



# Regresión Múltiple

La Regresión Múltiple es una técnica de aprendizaje supervisado que extiende la Regresión Lineal Simple para modelar la relación entre una variable dependiente (o de respuesta) y dos o más variables independientes (o predictoras).

Su objetivo es **predecir** el valor de la variable dependiente basándose en los valores de las **variables independientes**.

# Regresión Múltiple: interpretación de coeficientes

**Magnitud:** La magnitud del coeficiente indica la fuerza de la relación entre la variable independiente y la variable dependiente.

**Signo:** El signo del coeficiente indica la dirección de la relación:

- Un coeficiente **positivo** significa que un aumento en la variable independiente se asocia con un aumento en la variable dependiente.
- Un coeficiente **negativo** significa que un aumento en la variable independiente se asocia con una disminución en la variable dependiente.

# Métricas de Evaluación para Regresión

**Error Cuadrático Medio  
(MSE)**

- Promedio de los errores al cuadrado.

**Raíz del Error Cuadrático  
Medio (RMSE)**

- Raíz cuadrada del MSE.

**Error Absoluto Medio  
(MAE)**

- Promedio de los errores absolutos.

**R-cuadrado ( $R^2$ )**

- Proporción de la varianza explicada por el modelo.

# scikit-learn

Machine Learning in Python

Getting Started

Release Highlights for 1.6

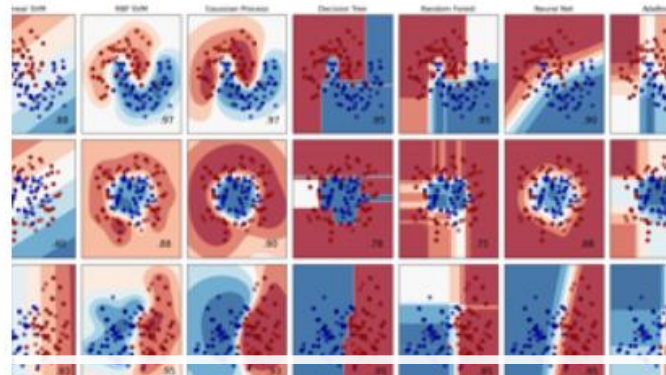
- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

## Classification

Identifying which category an object belongs to.

**Applications:** Spam detection, image recognition.

**Algorithms:** [Gradient boosting](#), [nearest neighbors](#), [random forest](#), [logistic regression](#), and [more...](#)



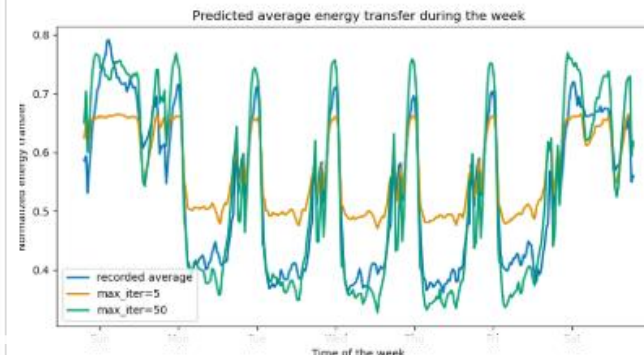
Examples

## Regression

Predicting a continuous-valued attribute associated with an object.

**Applications:** Drug response, stock prices.

**Algorithms:** [Gradient boosting](#), [nearest neighbors](#), [random forest](#), [ridge](#), and [more...](#)



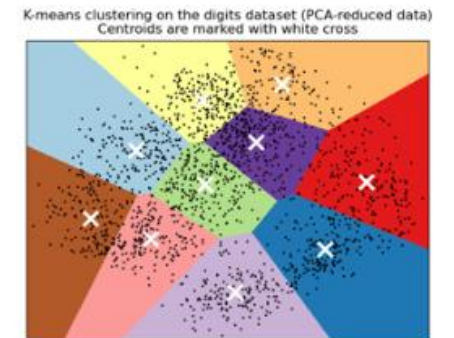
Examples

## Clustering

Automatic grouping of similar objects into sets.

**Applications:** Customer segmentation, grouping experiment outcomes.

**Algorithms:** [k-Means](#), [HDBSCAN](#), [hierarchical clustering](#), and [more...](#)



Examples

## Dimensionality reduction

Reducing the number of random variables to consider.

**Applications:** Visualization, increased efficiency.

## Model selection

Comparing, validating and choosing parameters and models.

## Preprocessing

Feature extraction and normalization.

**Applications:** Transforming input data such as text for

# Conclusiones...

La regresión lineal es una herramienta poderosa y ampliamente utilizada en el aprendizaje automático y la estadística. Su simplicidad y capacidad de interpretación la hacen valiosa para una variedad de aplicaciones de predicción.

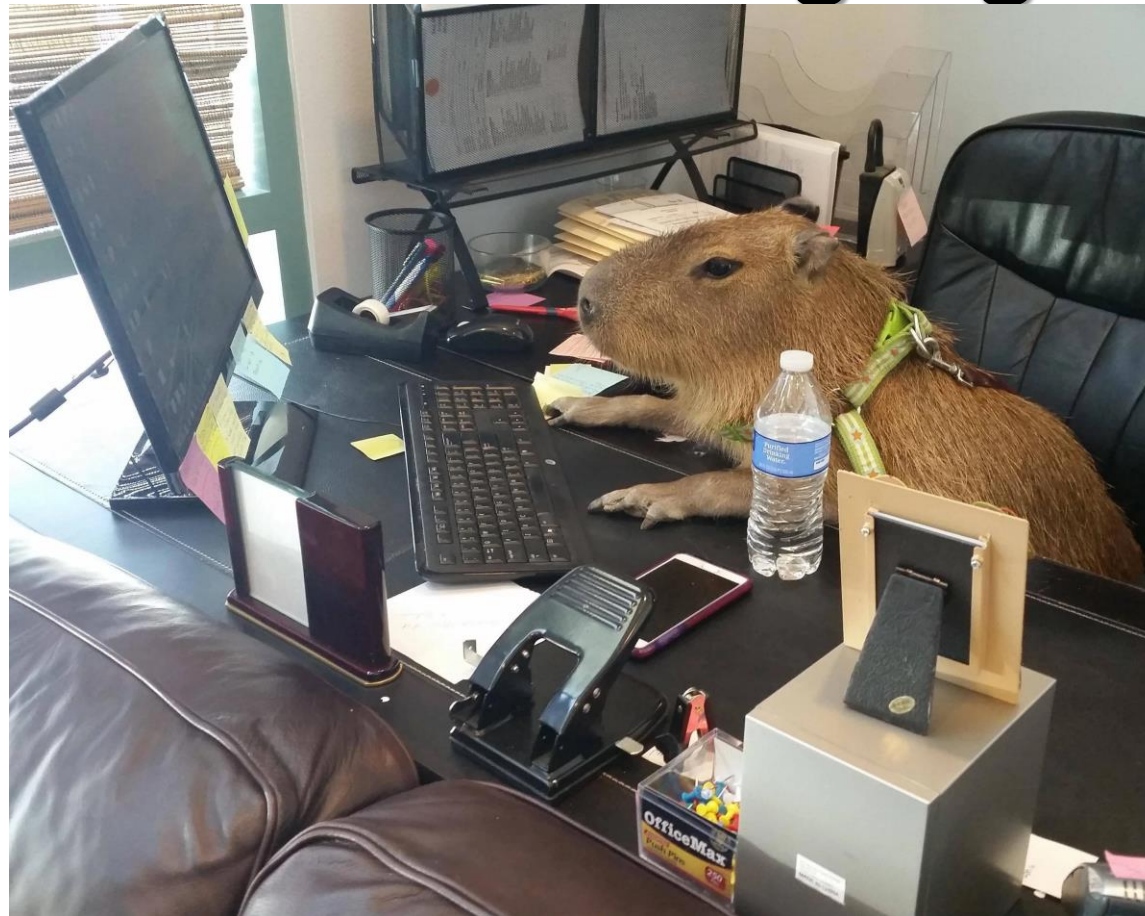
La Regresión Múltiple es una herramienta poderosa para modelar relaciones complejas entre variables.

Permite predecir una variable dependiente basándose en múltiples variables independientes, y los coeficientes proporcionan información valiosa sobre la fuerza y la dirección de estas relaciones.



# Ejercicio práctico

[colab.research.google.com](https://colab.research.google.com)



# Ejercicio práctico: Google Colaboratory (*Colabs*)

- Página oficial: <https://colab.google/>
- Abrir Colab (incluye tutorial): <https://colab.research.google.com/>
- Guía para EDA: [https://colab.research.google.com/github/Tanu-N-Prabhu/Python/blob/master/Exploratory\\_data\\_Analysis.ipynb](https://colab.research.google.com/github/Tanu-N-Prabhu/Python/blob/master/Exploratory_data_Analysis.ipynb)
- Guía / tutorial para Selección de características con **scikit-learn**: <https://www.datacamp.com/tutorial/feature-selection-python>

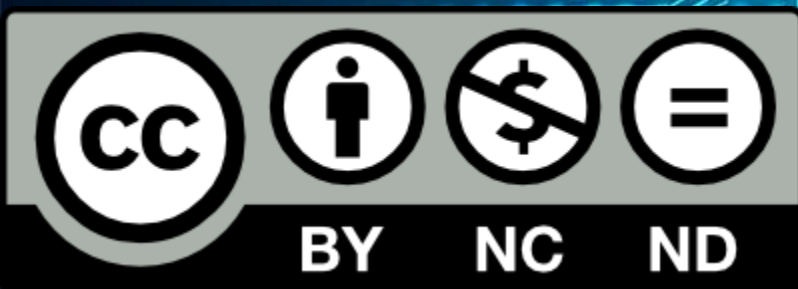




# Referencias

- “Regresión lineal | Machine Learning | Google for Developers”. Consultado: el 26 de marzo de 2025. [En línea]. Disponible en: <https://developers.google.com/machine-learning/crash-course/linear-regression?hl=es-419>
- “scikit-learn: Machine Learning in Python”. Consultado: el 20 de febrero de 2025. [En línea]. Disponible en: <https://scikit-learn.org/stable/>
- Información e ideas presentadas basadas en el conocimiento general de modelos de lenguaje de IA. Gemini 2.9 Flash. Consultado: el 25 de marzo de 2025. [En línea].

# Machine Learning



**Susana Medina Gordillo**

[susana.medina@correounivalle.edu.co](mailto:susana.medina@correounivalle.edu.co)

