# CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERIAS
SEMINARIO DE ALGORITMIA
**Actividad 11 Regresión Lineal:**

Sección: D11
Alumno: Erick Alejandro Carrillo López
Código: 221349828

2023/04/24

## Contents

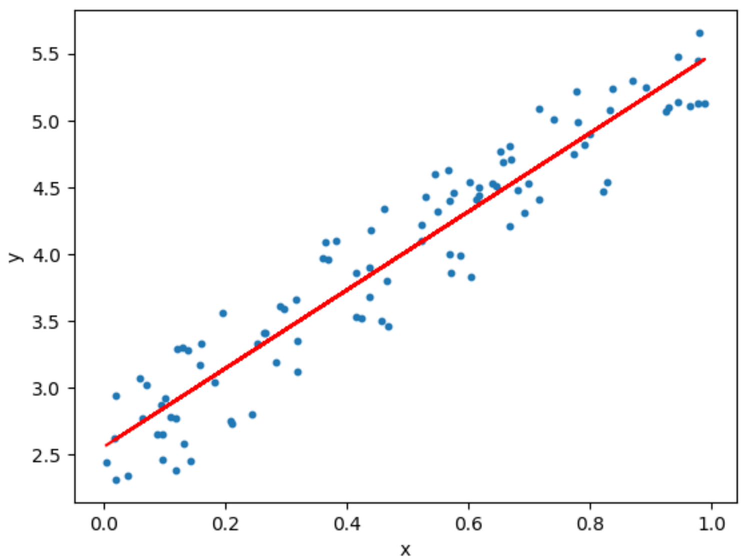
# 1 Introducción:

## 1.1 ¿Que es una regresión lineal?:

Realizar la regresión lineal es paso importante para introducirte en la inteligencia artificial, para mí es como hacer un hello world en un lenguaje de programación, sus conceptos he ideas en los que se fundamenta el algoritmo pueden ser algo complicadas para la gente que no esta familiarizada con las matemáticas.

En estadística, la regresión lineal es un enfoque lineal para modelizar la relación entre una respuesta escalar y una o más variables explicativas, en palabras mas simples busca una relación entre la variable dependiente con la independiente.

En el caso de que solo sea una variable independiente se dice que es una **regresión lineal simple** pero para multiples variables independientes se dice que es una **regresión lineal multiple**, como tal es a travez de esta realación escalar que se busca modelar una **funcion lienal predictora**, que como su nombre lo dice busca predecir o aproximar sus **outcomes** a todas las posibles variables dependientes.




Esta imagen representa perfectamente lo que se quiere logar a travez de la regresión lineal, la linea roja representa la **funcion lineal predictora** que mejor se acomoda o ajusta a los datos , esto a travez de una relación escalar.

## 1.2 ¿Cómo funciona?:

El cómo funciona parte de la conjetura o hipótesis de que exista una función lineal dada en $\vec{x}$ que se ajuste mejor a los datos o puntos por medio de una relación **escalar**, cuando digo relación escalar me refiero a una regla la cual puedes pensar en una función que pase de vectores a escalares.

$$E : \mathbb{R}^j \to \mathbb{R}$$

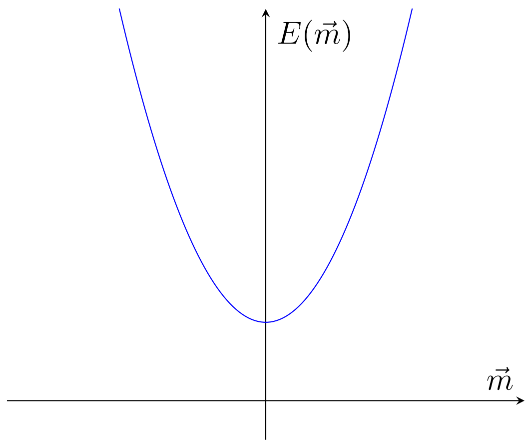
Para ser exactos estoy hablando del **error cuadrático medio**, por la cual fácilmente podemos proponer posibles **funciones lineales predictoras** y medir su grado de a proximidad al ajuste perfecto.

$$MES = \frac{1}{n} \cdot \sum_{i=0}^{n} (y_i - \theta(\vec{x_i}))^2$$

Donde $\theta(\vec{x_i}) = \vec{m} \cdot \vec{x_i}$ esta es la función lineal como puedes ver simplemente es un producto punto, esta recibe un vector con esta forma $\vec{x_i} = [1, x_1, \cdots, x_j] \quad | \quad x_0 = 1$ y este se multiplica por otro vector $\vec{m} = [b, m_1, \cdots, m_j] \quad | \quad m_0 = b$ por lo que sustituimos podemos definir nuestra función como.

$$E(\vec{m}) = \frac{1}{n} \cdot \sum_{i=0}^{n} ((\vec{m} \cdot \vec{x_i}) - y_i)^2$$

Lógicamente, se eleva al cuadrado para evitar generar esas diferencias negativas y se pueda manejar esta como cantidad medible entonces si creamos un gráfico de esta función debería de verse como una parábola, dentro del algoritmo de regresión lineal a esta función se le conoce como función de coste.




## 1.3 El decenso del gradiente:

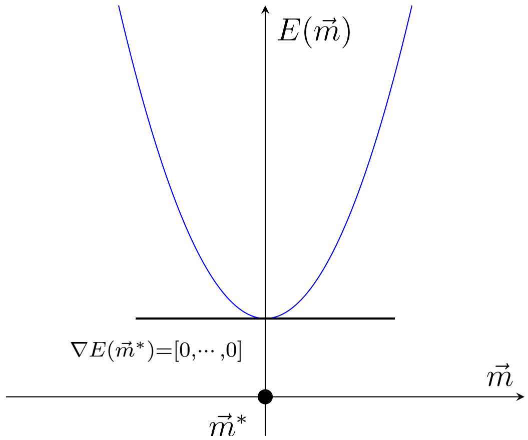
El algoritmo funciona partiendo de un $\vec{m}_0$ con la forma $[b, m_1, \cdots, m_j]$ este se escoge de forma aleatoria dentro del rango de la función $E(\vec{m})$, despues vamos a calcular la derivada de $E(\vec{m})$ sobre $\vec{m}$, esta derivada en realidad es un **gradiente** un simple vector con multiples derivadas parciales.

$$\frac{dE}{d\vec{m}} = \begin{bmatrix} \frac{\partial E}{\partial m_0} \\ \frac{\partial E}{\partial m_1} \\ \vdots \\ \frac{\partial E}{\partial m_j} \end{bmatrix} = \nabla E$$

Usando el **gradiente** podemos encontrar ese vector deseado el cual es $\vec{m}^*$, este debería de ubicarse en el punto donde todas las derivadas parciales da las componentes dan cero.

$$\nabla E(\vec{m}^*) = \begin{bmatrix} \frac{\partial E}{\partial m_0^*} \\ \frac{\partial E}{\partial m_1^*} \\ \vdots \\ \frac{\partial E}{\partial m_j^*} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Y esto se debe a la ubicación del vector ideal $\vec{m}^*$ se encuentra en el punto más bajo de la parábola de múltiples dimensiones.




## 1.4 Calculando las derivadas parciales del gradiente:

Ahora para calcular el gradiente debemos de calcular todas las derivadas parciales a través de las cuales podemos usar la regla de la cadena para simplificar todos nuestros caculos.

Primero quiero desglosar un poco nuestra función de coste para entender mejor las derivadas parciales.

$$E(\vec{m}) = \frac{1}{n} \cdot \sum_{i=0}^{n} (\vec{x_i} \cdot \vec{m} - y_i)^2 = \frac{1}{n} \cdot ((\vec{x_0} \cdot \vec{m} - y_0)^2 + \cdots + (\vec{x_n} \cdot \vec{m} - y_n)^2)$$

Recordemos que por cada $\vec{x_i} \cdot \vec{m}$ es un producto punto por lo que.

$$\vec{x_i} \cdot \vec{m} = \sum_{l=0}^{j} (x_{i,l} \cdot m_l)$$

Sustituyendo nos debería de verse algo como esto.

$$E(\vec{m}) = \frac{1}{n} \cdot ((\sum_{l=0}^{j} (x_{0,l} \cdot m_l) - y_0)^2 + \cdots + (\sum_{l=0}^{j} (x_{n,l} \cdot m_l) - y_n)^2)$$

Por lo que si queremos calcular la derivada parcial para cualquier componente $m_j$ podemos realizar la regla de la cadena con tal de simplificar todo, por ejemplo podemos sustituir por $u_i = \vec{x_i} \cdot \vec{m} - y_i$.

$$u_0 = \vec{x_0} \cdot \vec{m} - y_0 = \sum_{l=0}^{j} (x_{0,l} \cdot m_l) - y_0 = ((x_{0,0} \cdot m_0) + (x_{0,1} \cdot m_1) + \cdots + (x_{0,j} \cdot m_j)) - y_0$$

$$\vdots$$

$$u_n = \vec{x_n} \cdot \vec{m} - y_n = \sum_{l=0}^{j} (x_{n,l} \cdot m_l) - y_n = ((x_{n,0} \cdot m_0) + (x_{n,1} \cdot m_1) + \cdots + (x_{n,j} \cdot m_j)) - y_n$$

Y sacar derivada parcial de cualquier $u_n$ con respecto de cualquier $m_j$ la cual nos debería de quedar algo como esto.

$$\frac{\partial u_n}{\partial m_j} = x_{n,j}$$

Con esta sustitución podemos calcular nuestra derivada parcial $E(\vec{m})$ con respecto cualquier componente $\vec{m_j}$ y poder crear nuestro gradiente, todo a partir de la poderosa chain rule.

$$\frac{\partial E}{\partial m_j} = \frac{1}{n}(\frac{\partial((\vec{x_0} \cdot \vec{m} - y_0)^2)}{\partial m_j} + \frac{\partial((\vec{x_1} \cdot \vec{m} - y_1)^2)}{\partial m_j} + \cdots + \frac{\partial((\vec{x_n} \cdot \vec{m})^2 - y_n)}{\partial m_j})$$

$$\frac{\partial E}{\partial m_j} = \frac{1}{n}(\frac{\partial(u_0^2)}{\partial u_0} \cdot \frac{\partial u_0}{\partial m_j} + \frac{\partial(u_1^2)}{\partial u_1} \cdot \frac{\partial u_1}{\partial m_j} + \cdots + \frac{\partial(u_n^2)}{\partial u_n} \cdot \frac{\partial u_n}{\partial m_j})$$

$$\frac{\partial E}{\partial m_j} = \frac{1}{n}(2u_0 \cdot x_{0,j} + 2u_1 \cdot x_{1,j} + \cdots + 2u_n \cdot x_{n,j})$$

$$\frac{\partial E}{\partial m_j} = \frac{1}{n}(2(\vec{x_0} \cdot \vec{m} - y_0) \cdot x_{0,j} + 2(\vec{x_1} \cdot \vec{m} - y_1) \cdot x_{1,j} + \cdots + 2(\vec{x_n} \cdot \vec{m} - y_n) \cdot x_{n,j})$$

$$\frac{\partial E}{\partial m_j} = \frac{2}{n} \cdot \sum_{i=0}^{n}((\vec{x_i} \cdot \vec{m} - y_i) \cdot x_{i,j})$$

Y podemos generalizar esto para cualquier derivada parcial del componente $j$ para completar lo que viene siendo nuestro gradiente.

$$\nabla E(\vec{m}) = \begin{bmatrix} \frac{\partial E}{\partial m_0} \\ \frac{\partial E}{\partial m_1} \\ \vdots \\ \frac{\partial E}{\partial m_j} \end{bmatrix} = \begin{bmatrix} \frac{2}{n} \cdot \sum_{i=0}^{n}((\vec{x_i} \cdot \vec{m} - y_i) \cdot x_{i,0}) \\ \frac{2}{n} \cdot \sum_{i=0}^{n}((\vec{x_i} \cdot \vec{m} - y_i) \cdot x_{i,1}) \\ \vdots \\ \frac{2}{n} \cdot \sum_{i=0}^{n}((\vec{x_i} \cdot \vec{m} - y_i) \cdot x_{i,j}) \end{bmatrix}$$

## 1.5 El algoritmo:

Entonces nuevamente partiendo de la idea de que empezamos con un vector aleatorio $\vec{m_0} \in \mathbb{R}^j$, con el gradiente podemos computar la siguiente iteración hacía el vector $\vec{m_1}$ usando la dirección opuesta del gradiente osea $-\nabla E(\vec{m_0})$, generalizando eso a $k$ iteraciónes nos debería de quedar algo como esto.

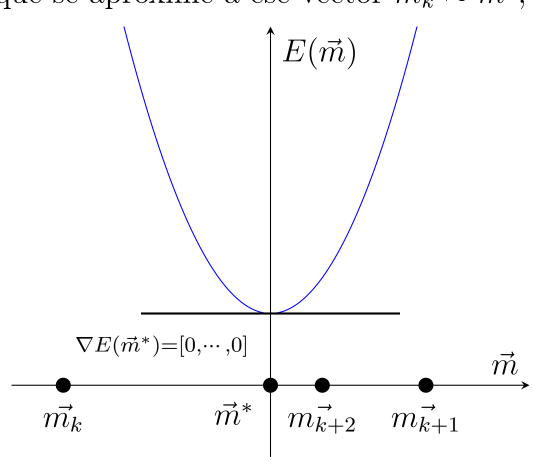
$$\vec{m_{k+1}} = \vec{m_k} - \nabla E(\vec{m_k})$$

Para evitar cambios muy bruscos y que nuestro vector nunca llegue a una parte cóncava o convexa de nuestra función de coste, lo que hacemos es escalar el gradiente a próximo a un vector unitario o a cantidad muy pequeña a este paso se le llama añadirle el **learning rate**, como tal solo estamos interesados en sacar la dirección del gradiente.

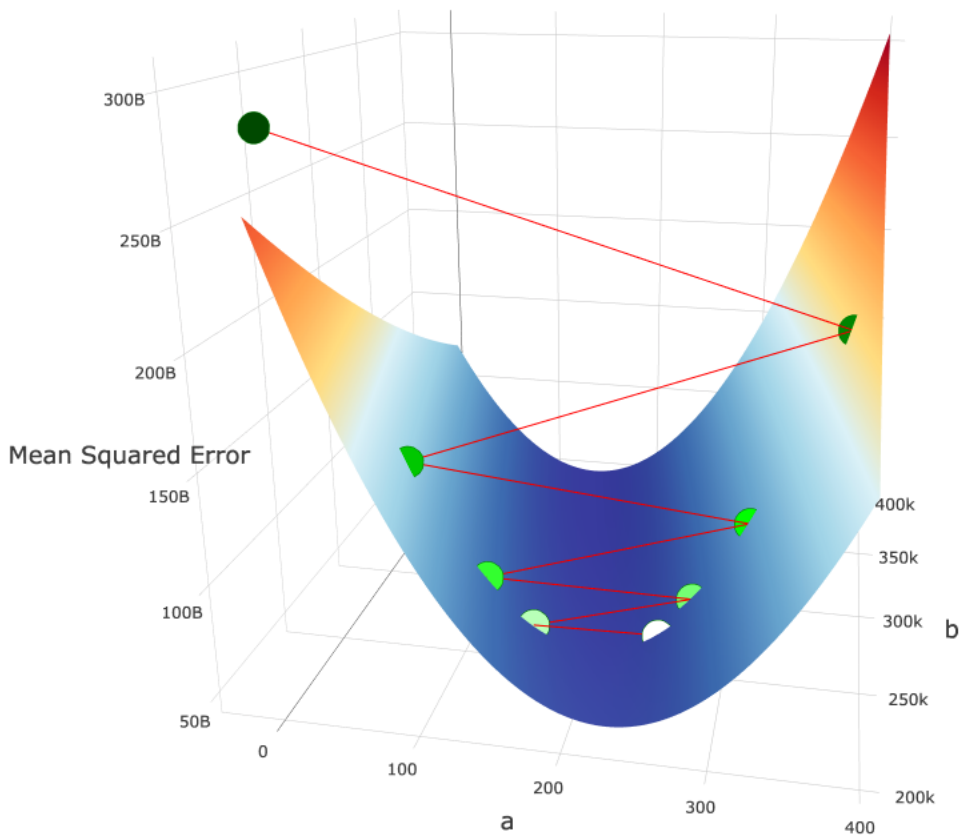
$$\nabla E(\vec{m_k}) \cdot \gamma = \nabla E(\vec{m_k})^* \quad | \quad \gamma \to 0$$

$$\vec{m_{k+1}} = \vec{m_k} - \gamma \cdot \nabla E(\vec{m_k})$$

Con el $-$ podemos provocar que el gradiente escalado tenga una dirección opuesta, y con las restas o sumas vamos ajustando el $\vec{m_k}$ a que se aproxime a ese vector $\vec{m_k} \approx \vec{m}^*$, puedes pensar en este gráfico.



Como puedes ver en el gráfico cada vez se va aproximando hasta el que $\nabla E(\vec{m}_k) \approx [0, \cdots, 0]$ y ya no se realicen cambios o ajustes al $\vec{m}$, otra imagen que explica bien lo que está pasando.




Prácticamente, haces que este rebote hasta que desciende lo suficiente.

# 2 Objetivos:

## 2.1 Objetivo general:

## 2.2 Objetivo particulares:

-

# 3 Desarrollo:

# 4 Conclusión:

# 5 Referencias:


- Wikipedia contributors. (2022). Mean squared error. Wikipedia. `https://en.wikipedia.org/wiki/Mean_squared_error`
- Wikipedia contributors. (2023a). Linear regression. Wikipedia. `https://en.wikipedia.org/wiki/Linear_regression`

- Wikipedia contributors. (2023b). Notation for differentiation. Wikipedia. https://en.wikipedia.org/wiki/Notation_for_differentiation
- Wikipedia contributors. (2023a). Gradient. Wikipedia. https://en.wikipedia.org/wiki/Gradient