Regresión Lineal

M.Sc. William Caicedo Torres

Universidad Tecnológica de Bolívar

caicedo77@gmail.com

5 de octubre de 2016

Un problema

- Ud. es contratado por un banco para automatizar el proceso de asignación de cupos de crédito a los usuarios.

Un problema

- Ud. es contratado por un banco para automatizar el proceso de asignación de cupos de crédito a los usuarios.
- La idea es que ud. escriba un programa basado en Inteligencia Artificial, que sea capaz de asignar automáticamente un cupo de crédito adecuado a un cliente determinado.

Un problema

- Ud. es contratado por un banco para automatizar el proceso de asignación de cupos de crédito a los usuarios.
- La idea es que ud. escriba un programa basado en Inteligencia Artificial, que sea capaz de asignar automáticamente un cupo de crédito adecuado a un cliente determinado.
- Hasta el momento, en el banco se asigna el cupo de crédito teniendo en cuenta los siguientes criterios:

Un problema

- Ud. es contratado por un banco para automatizar el proceso de asignación de cupos de crédito a los usuarios.
- La idea es que ud. escriba un programa basado en Inteligencia Artificial, que sea capaz de asignar automáticamente un cupo de crédito adecuado a un cliente determinado.
- Hasta el momento, en el banco se asigna el cupo de crédito teniendo en cuenta los siguientes criterios:
  - Edad.

Un problema

- Ud. es contratado por un banco para automatizar el proceso de asignación de cupos de crédito a los usuarios.
- La idea es que ud. escriba un programa basado en Inteligencia Artificial, que sea capaz de asignar automáticamente un cupo de crédito adecuado a un cliente determinado.
- Hasta el momento, en el banco se asigna el cupo de crédito teniendo en cuenta los siguientes criterios:
  - Edad.
  - Salario anual.

Un problema

- Ud. es contratado por un banco para automatizar el proceso de asignación de cupos de crédito a los usuarios.
- La idea es que ud. escriba un programa basado en Inteligencia Artificial, que sea capaz de asignar automáticamente un cupo de crédito adecuado a un cliente determinado.
- Hasta el momento, en el banco se asigna el cupo de crédito teniendo en cuenta los siguientes criterios:
    - Edad.
    - Salario anual.
    - Tiempo en el trabajo actual (años).

Universidad Tecnológica de Bolívar
CARTAGENA DE INDIAS

Un problema

- Ud. es contratado por un banco para automatizar el proceso de asignación de cupos de crédito a los usuarios.
- La idea es que ud. escriba un programa basado en Inteligencia Artificial, que sea capaz de asignar automáticamente un cupo de crédito adecuado a un cliente determinado.
- Hasta el momento, en el banco se asigna el cupo de crédito teniendo en cuenta los siguientes criterios:
  - Edad.
  - Salario anual.
  - Tiempo en el trabajo actual (años).
  - Total de deuda actual.

- Además de los criterios por los cuales los analistas toman la decisión, se cuenta con datos históricos (ejemplos) sobre los cupos aprobados:

$$\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$$

donde x_n es un vector que representa la información del cliente n (edad, salario, tiempo de trabajo y total de deuda), y $y_n \in \mathbb{R}$ es el cupo aprobado para dicho cliente.

- Además de los criterios por los cuales los analistas toman la decisión, se cuenta con datos históricos (ejemplos) sobre los cupos aprobados:

$$\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$$

  donde x_n es un vector que representa la información del cliente n (edad, salario, tiempo de trabajo y total de deuda), y $y_n \in \mathbb{R}$ es el cupo aprobado para dicho cliente.
- Entonces, ¿Cómo escribimos un programa que intente replicar las decisiones tomadas por los expertos humanos en este caso?

- En Machine Learning, un algoritmo de regresión produce salidas que pertenecen a $\mathbb{R}$.

- En Machine Learning, un algoritmo de regresión produce salidas que pertenecen a $\mathbb{R}$.
- A través de un proceso de entrenamiento, busca modelar lo mejor posible el comportamiento de un sistema real.

Regresión Lineal

- En Machine Learning, un algoritmo de regresión produce salidas que pertenecen a $\mathbb{R}$.
- A través de un proceso de entrenamiento, busca modelar lo mejor posible el comportamiento de un sistema real.
- En el caso de la regresión lineal, la hipótesis usada para conseguirlo tiene la siguiente forma:

$$h(x) = \theta_0 + \theta_1 x_1 + \ldots + \theta_n x_n$$

Regresión Lineal

- En Machine Learning, un algoritmo de regresión produce salidas que pertenecen a $\mathbb{R}$.
- A través de un proceso de entrenamiento, busca modelar lo mejor posible el comportamiento de un sistema real.
- En el caso de la regresión lineal, la hipótesis usada para conseguirlo tiene la siguiente forma:

$$h(x) = \theta_0 + \theta_1 x_1 + \ldots + \theta_n x_n$$

Donde θ_n representa cada uno de los parámetros del modelo.

Universidad Tecnológica de Bolívar
CARTAGENA DE INDIAS

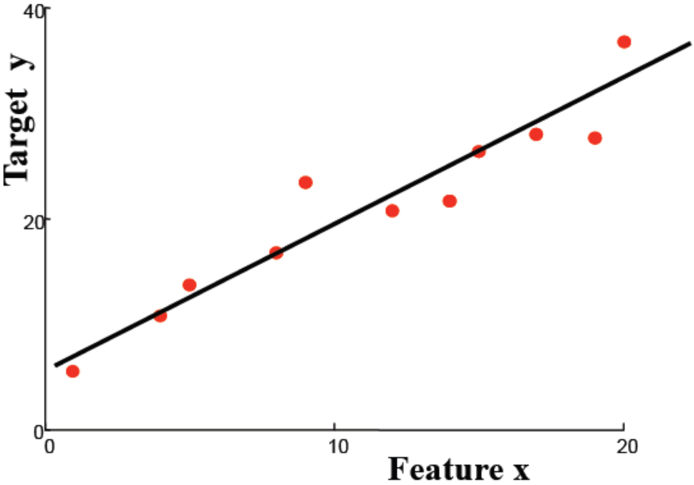
- En Machine Learning, un algoritmo de regresión produce salidas que pertenecen a $\mathbb{R}$.
- A través de un proceso de entrenamiento, busca modelar lo mejor posible el comportamiento de un sistema real.
- En el caso de la regresión lineal, la hipótesis usada para conseguirlo tiene la siguiente forma:

$$h(x) = \theta_0 + \theta_1 x_1 + \ldots + \theta_n x_n$$

  Donde θ_n representa cada uno de los parámetros del modelo.
- Nótese que la expresión anterior es lineal (una recta) en θ, de ahí el nombre Regresión Lineal.

Universidad Tecnológica de Bolívar
CARTAGENA DE INDIAS

Regresión Lineal: Representación gráfica



Universidad Tecnológica de Bolívar
CARTAGENA DE INDIAS

Regresión Lineal: Aprendizaje

- Queremos obtener la recta que mejor capture el comportamiento de los datos de entrenamiento.

Regresión Lineal: Aprendizaje

- Queremos obtener la recta que mejor capture el comportamiento de los datos de entrenamiento.
- El primer paso para lograrlo es definir una medida de error que nos permita elegir cual es la mejor recta de todas.

- Queremos obtener la recta que mejor capture el comportamiento de los datos de entrenamiento.
- El primer paso para lograrlo es definir una medida de error que nos permita elegir cual es la mejor recta de todas.
- En este caso queremos una recta que pase lo más cerca posible de todos los puntos de entrenamiento, por lo que introducimos la siguiente medida de error:

$$J(\theta) = \frac{1}{N}\sum_{n=1}^{N}(h(x_n) - y_n)^2$$

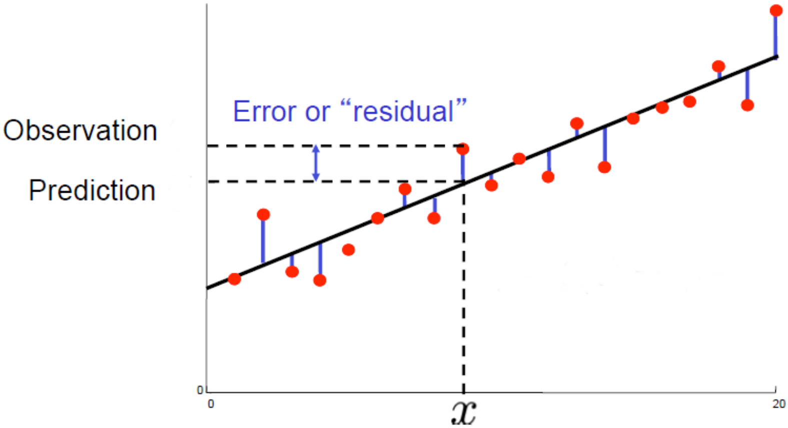
- Queremos obtener la recta que mejor capture el comportamiento de los datos de entrenamiento.
- El primer paso para lograrlo es definir una medida de error que nos permita elegir cual es la mejor recta de todas.
- En este caso queremos una recta que pase lo más cerca posible de todos los puntos de entrenamiento, por lo que introducimos la siguiente medida de error:

$$J(\theta) = \frac{1}{N}\sum_{n=1}^{N}(h(x_n) - y_n)^2$$

- Vale destacar que la medida error es una función de los parámetros θ. De aquí podemos concluir que el proceso de entrenamiento es realmente buscar un conjunto de θs que minimice el error $J(\theta)$

Universidad Tecnológica de Bolívar
CARTAGENA DE INDIAS

Regresión Lineal: Aprendizaje



- Ahora, para encontrar el conjunto de θs que minimice el error, planteamos el siguiente problema de optimización:

- Ahora, para encontrar el conjunto de θs que minimice el error, planteamos el siguiente problema de optimización:

$$min_{\theta} J(\theta) = \frac{1}{N}\sum_{n=1}^{N}(h(x_n) - y_n)^2$$

Regresión Lineal: Aprendizaje

- Ahora, para encontrar el conjunto de θs que minimice el error, planteamos el siguiente problema de optimización:

$$min_{\theta} J(\theta) = \frac{1}{N} \sum_{n=1}^{N} (h(x_n) - y_n)^2$$

- Observando detenidamente la estructura de la función de error, podremos notar que es una función cuadrática en θ - Recordemos que

$$h(x) = \theta_0 + \theta_1 x_1 + \ldots + \theta_n x_n$$

Universidad Tecnológica de Bolívar
CARTAGENA DE INDIAS

Regresión Lineal: Aprendizaje

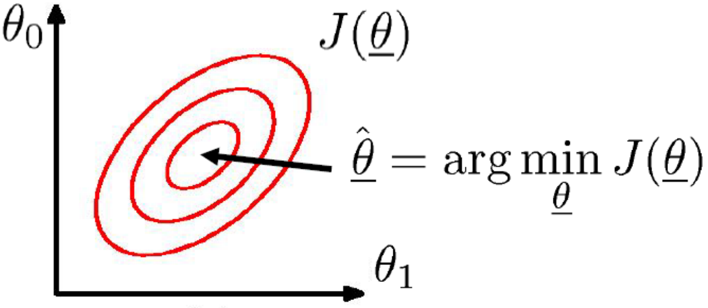
- Ahora, para encontrar el conjunto de θs que minimice el error, planteamos el siguiente problema de optimización:

$$min_\theta J(\theta) = \frac{1}{N}\sum_{n=1}^{N}(h(x_n) - y_n)^2$$

- Observando detenidamente la estructura de la función de error, podremos notar que es una función cuadrática en θ - Recordemos que

$$h(x) = \theta_0 + \theta_1 x_1 + \ldots + \theta_n x_n$$

- El hecho de que sea cuadrática es de gran ayuda: Nos garantiza la existencia de un único mínimo global!


θ_0
$J(\underline{\theta})$
$\hat{\underline{\theta}} = \arg\min_{\underline{\theta}} J(\underline{\theta})$
θ_1

- Comencemos por expresar nuestra hipótesis $h(x)$ de una forma más compacta.

- Comencemos por expresar nuestra hipótesis $h(x)$ de una forma más compacta.
  Si agregamos una nueva variable artificial, $x_0 = 1$, nuestra hipótesis quedaría de la siguiente manera:

$$h(x) = \theta_0 x_0 + \theta_1 x_1 + \ldots + \theta_n x_n$$

- Comencemos por expresar nuestra hipótesis $h(x)$ de una forma más compacta.
  Si agregamos una nueva variable artificial, $x_0 = 1$, nuestra hipótesis quedaría de la siguiente manera:

$$h(x) = \theta_0 x_0 + \theta_1 x_1 + \ldots + \theta_n x_n$$

$$h(x) = \sum_{i=0}^{n} \theta_i x_i$$

Regresión Lineal: Solución al problema de optimización

- Comencemos por expresar nuestra hipótesis $h(x)$ de una forma más compacta.
  Si agregamos una nueva variable artificial, $x_0 = 1$, nuestra hipótesis quedaría de la siguiente manera:

$$h(x) = \theta_0 x_0 + \theta_1 x_1 + \ldots + \theta_n x_n$$

$$h(x) = \sum_{i=0}^{n} \theta_i x_i$$

$$h(x) = \theta^T x$$

- Ahora, procederemos a derivar la solución al problema de aprendizaje.

Tenemos entonces que

$$J(\theta) = \frac{1}{N}\sum_{n=1}^{N}(\theta^T \mathbf{x}_n - y_n)^2$$

Regresión Lineal: Solución al problema de optimización

Tenemos entonces que

$$J(\theta) = \frac{1}{N}\sum_{n=1}^{N}(\theta^{T}x_n - y_n)^2$$

Lo anterior puede ser expresado en notación vectorial:

Tenemos entonces que

$$J(\theta) = \frac{1}{N}\sum_{n=1}^{N}(\theta^T x_n - y_n)^2$$

Lo anterior puede ser expresado en notación vectorial:

$$J(\theta) = \frac{1}{N}||X\theta - y||^2$$

Tenemos entonces que

$$J(\theta) = \frac{1}{N}\sum_{n=1}^{N}(\theta^T x_n - y_n)^2$$

Lo anterior puede ser expresado en notación vectorial:

$$J(\theta) = \frac{1}{N}||X\theta - y||^2$$

donde $X = \begin{bmatrix} - - x_1^T - - \\ - - x_2^T - - \\ \vdots \\ - - x_N^T - - \end{bmatrix}$, $y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$

Universidad Tecnológica de Bolívar
CARTAGENA DE INDIAS

- Ahora, procedemos a minimizar $J(\theta)$:

- Ahora, procedemos a minimizar $J(\theta)$:

$$J(\theta) = \frac{1}{N}||X\theta - y||^2$$

- Ahora, procedemos a minimizar $J(\theta)$:

$$J(\theta) = \frac{1}{N}||X\theta - y||^2$$

$$\nabla J(\theta) = \frac{2}{N}X^T(X\theta - y) = 0$$

- Ahora, procedemos a minimizar $J(\theta)$:

$$J(\theta) = \frac{1}{N}||X\theta - y||^2$$

$$\nabla J(\theta) = \frac{2}{N}X^T(X\theta - y) = 0$$

$$X^T X\theta = X^T y$$

Regresión Lineal: Solución al problema de optimización

- Ahora, procedemos a minimizar $J(\theta)$:

$$J(\theta) = \frac{1}{N}||X\theta - y||^2$$

$$\nabla J(\theta) = \frac{2}{N}X^T(X\theta - y) = 0$$

$$X^TX\theta = X^Ty$$

$$\theta = (X^TX)^{-1}X^Ty$$

Universidad Tecnológica de Bolívar
CARTAGENA DE INDIAS

- Ahora, procedemos a minimizar $J(\theta)$:

$$J(\theta) = \frac{1}{N}||X\theta - y||^2$$

$$\nabla J(\theta) = \frac{2}{N}X^T(X\theta - y) = 0$$

$$X^T X\theta = X^T y$$

$$\theta = (X^T X)^{-1} X^T y$$

A la matriz $(X^T X)^{-1} X^T$ se le denomina la matriz **Pseudoinversa** de X.

- Vemos entonces, que existe un vector θ que minimiza el error $J(\theta)$.

- Vemos entonces, que existe un vector θ que minimiza el error $J(\theta)$.
- Para encontrarlo debemos:

- Vemos entonces, que existe un vector θ que minimiza el error $J(\theta)$.
- Para encontrarlo debemos:
  1. Construir la matriz X (entradas) y el vector y (salidas esperadas) con los datos de entrenamiento.

Regresión Lineal: Solución al problema de optimización

- Vemos entonces, que existe un vector θ que minimiza el error $J(\theta)$.
- Para encontrarlo debemos:
  1. Construir la matriz X (entradas) y el vector y (salidas esperadas) con los datos de entrenamiento.
  2. Calcular la matriz pseudoinversa de X:

$$X^{\dagger} = (X^T X)^{-1} X^T$$

Regresión Lineal: Solución al problema de optimización

- Vemos entonces, que existe un vector θ que minimiza el error $J(\theta)$.
- Para encontrarlo debemos:
  1. Construir la matriz X (entradas) y el vector y (salidas esperadas) con los datos de entrenamiento.
  2. Calcular la matriz pseudoinversa de X:

$$X^{\dagger} = (X^T X)^{-1} X^T$$

  3. Obtener el vector de parámetros θ:

Universidad Tecnológica de Bolívar
CARTAGENA DE INDIAS

Regresión Lineal: Solución al problema de optimización

- Vemos entonces, que existe un vector θ que minimiza el error $J(\theta)$.
- Para encontrarlo debemos:
  1. Construir la matriz X (entradas) y el vector y (salidas esperadas) con los datos de entrenamiento.
  2. Calcular la matriz pseudoinversa de X:

$$X^{\dagger} = (X^T X)^{-1} X^T$$

  3. Obtener el vector de parámetros θ:

$$\theta = X^{\dagger} y$$

Universidad Tecnológica de Bolívar
CARTAGENA DE INDIAS

- Vemos entonces, que existe un vector θ que minimiza el error $J(\theta)$.
- Para encontrarlo debemos:
  1. Construir la matriz X (entradas) y el vector y (salidas esperadas) con los datos de entrenamiento.
  2. Calcular la matriz pseudoinversa de X:

$$X^\dagger = (X^T X)^{-1} X^T$$

  3. Obtener el vector de parámetros θ:

$$\theta = X^\dagger y$$

- Luego de esto, estamos listos para hacer predicciones para nuevas entradas, de la siguiente forma:

$$h(x) = \theta^T x$$

Universidad Tecnológica de Bolívar
CARTAGENA DE INDIAS

- El método de gradiente descendiente es un algoritmo de optimización que utiliza el valor del gradiente de la función a optimizar para dirigirse hacia un mínimo local.

- El método de gradiente descendiente es un algoritmo de optimización que utiliza el valor del gradiente de la función a optimizar para dirigirse hacia un mínimo local.
- Es un método iterativo de primer orden.

Un método alternativo de solución: Gradiente Descendente

- El método de gradiente descendiente es un algoritmo de optimización que utiliza el valor del gradiente de la función a optimizar para dirigirse hacia un mínimo local.
- Es un método iterativo de primer orden.
- Si la función a optimizar es convexa, este método siempre convergerá al óptimo local dada una correcta escogencia de parámetros.

- El método de gradiente descendiente es un algoritmo de optimización que utiliza el valor del gradiente de la función a optimizar para dirigirse hacia un mínimo local.
- Es un método iterativo de primer orden.
- Si la función a optimizar es convexa, este método siempre convergerá al óptimo local dada una correcta escogencia de parámetros.
- La intuición tras este método es que siendo el gradiente el vector que apunta en la dirección de mayor crecimiento de la función a optimizar, es lógico entonces moverse en la **dirección contraria a este**, para alcanzar un óptimo local.

- Fórmula general:

- Fórmula general:

$$\theta = \theta - \alpha \frac{\partial J(\theta)}{\partial \theta}$$

Un método alternativo de solución: Gradiente Descendente

- Fórmula general:

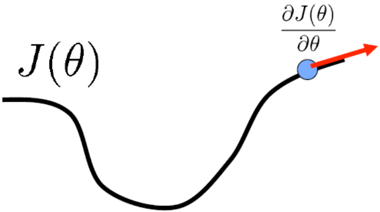
$$\theta = \theta - \alpha \frac{\partial J(\theta)}{\partial \theta}$$

- Captura la intuición de moverse en la dirección contraria al gradiente, al anteponer un signo negativo a este.

- Fórmula general:

$$\theta = \theta - \alpha \frac{\partial J(\theta)}{\partial \theta}$$

- Captura la intuición de moverse en la dirección contraria al gradiente, al anteponer un signo negativo a este.

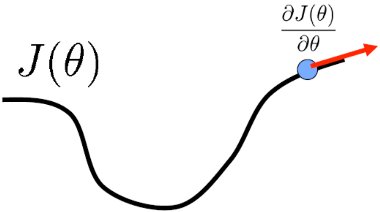


Un método alternativo de solución: Gradiente Descendente

- Fórmula general:

$$\theta = \theta - \alpha \frac{\partial J(\theta)}{\partial \theta}$$

- Captura la intuición de moverse en la dirección contraria al gradiente, al anteponer un signo negativo a este.



- Para poder aplicarlo, necesitamos conocer el gradiente de la función a optimizar ($\frac{\partial J(\theta)}{\partial \theta}$).

Gradiente Descendente aplicado a la Regresión Lineal

- Entonces, tenemos que la función de error en la Regresión Lineal es:

- Entonces, tenemos que la función de error en la Regresión Lineal es:

$$J(\theta) = \frac{1}{2N} \sum_{n=1}^{N} (h(x_n) - y_n)^2$$

Nótese que ahora la sumatoria está multiplicada por $\frac{1}{2N}$. Este es un pequeño artificio para facilitar la derivación de la regla de actualización de Gradiente Descendente, y no afecta el resultado.

Universidad Tecnológica de Bolívar
CARTAGENA DE INDIAS

- Entonces, tenemos que la función de error en la Regresión Lineal es:

$$J(\theta) = \frac{1}{2N}\sum_{n=1}^{N}(h(x_n) - y_n)^2$$

  Nótese que ahora la sumatoria está multiplicada por $\frac{1}{2N}$. Este es un pequeño artificio para facilitar la derivación de la regla de actualización de Gradiente Descendente, y no afecta el resultado.

- El gradiente entonces es:

Universidad Tecnológica de Bolívar
CARTAGENA DE INDIAS

- Entonces, tenemos que la función de error en la Regresión Lineal es:

$$J(\theta) = \frac{1}{2N} \sum_{n=1}^{N} (h(x_n) - y_n)^2$$

Nótese que ahora la sumatoria está multiplicada por $\frac{1}{2N}$. Este es un pequeño artificio para facilitar la derivación de la regla de actualización de Gradiente Descendente, y no afecta el resultado.

- El gradiente entonces es:

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{N} \sum_{n=1}^{N} (h(x_n) - y_n) x_{nj}$$

Universidad Tecnológica de Bolívar
CARTAGENA DE INDIAS

- Por lo anterior, la regla de actualización de Gradiente Descendente sería:

- Por lo anterior, la regla de actualización de Gradiente Descendente sería:

$$\theta j = \theta_j - \alpha \frac{1}{N} \sum_{n=1}^{N} (h(x_n) - y_n) x_{jn}$$

- Teniendo la regla, el algoritmo sería el siguiente:

- Por lo anterior, la regla de actualización de Gradiente Descendente sería:

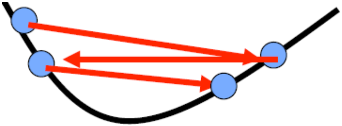
$$\theta j = \theta_j - \alpha \frac{1}{N} \sum_{n=1}^{N} (h(x_n) - y_n) x_{jn}$$

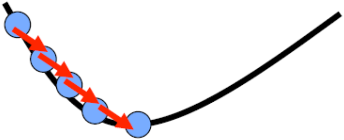
- Teniendo la regla, el algoritmo sería el siguiente:
  1. Hacer, para toda j
  2. $$\theta j = \theta_j - \alpha \frac{1}{N} \sum_{n=1}^{N} (h(x_n) - y_n) x_{nj}$$
  3. Mientras que $\alpha || \frac{\partial J(\theta)}{\partial \theta_j} || > \epsilon$

  Nota: La actualización de los θ_j debe ser simultánea (primero se calculan todos los θ_j y luego se reemplazan los anteriores).

Gradiente Descendente aplicado a la Regresión Lineal

- α se conoce como la tasa de aprendizaje (el tamaño de los pasos que da el algoritmo), y debe ser escogido cuidadosamente puesto que un valor muy alto puede hacer que el algoritmo diverja. Si es muy pequeño el algoritmo puede demorar demasiado tiempo en converger.





Escalamiento y normalización de los datos

- Al usar Gradiente Descendente para entrenar un modelo de Regresión Lineal, es conveniente escalar y normalizar las entradas.
- Escalando y normalizando usualmente acelera el proceso de aprendizaje, al poder usarse una tasa de aprendizaje mayor.
- Una forma de normalizar la media de los datos ($\mu = 0$) y llevarlos aproximandamente al intervalo $[-1, 1]$ es la siguiente:

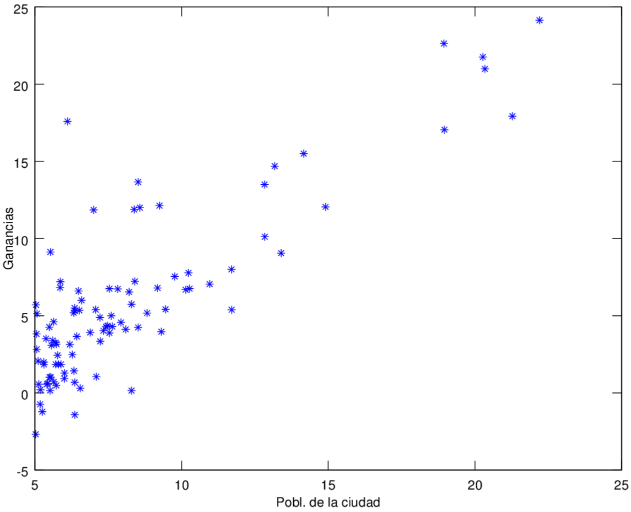
$$x_i = \frac{x_i - \mu_i}{max_i - min_i}$$

para cada variable de entrada x.

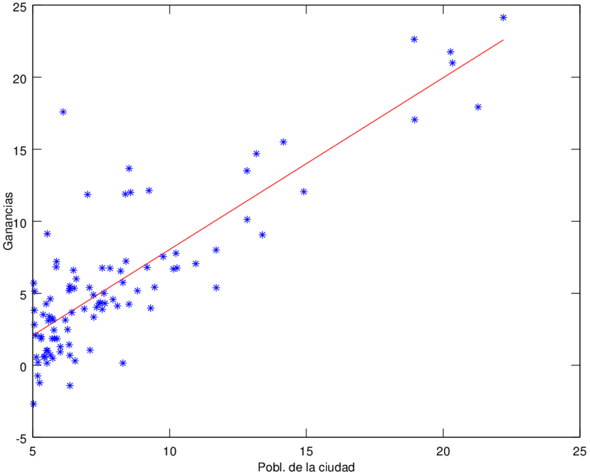
Ejemplo

- Usted es el CEO de una franquicia de restaurantes y está considerando varias ciudades para abrir un nuevo local.
- Se cuenta con información sobre las ciudades donde ya se han abierto restaurantes de la franquicia (población de la ciudad y ganancias obtenidas.
- Queremos utilizar la información disponible para decidir en que ciudad abrir un nuevo restaurante.

Datos Ciudades vs Ganancias



Modelo de Regresión entrenado



$$Gan = -3,8958 + 1,1930 * Pob$$

Universidad Tecnológica de Bolívar
CARTAGENA DE INDIAS

Muchas gracias!

Preguntas?