Aprendizaje supervisado

El aprendizaje supervisado es una subcategoría del machine learning y la inteligencia artificial que se basa en el uso de conjuntos de datos etiquetados para entrenar algoritmos que clasifican datos o predicen resultados con precisión.

El aprendizaje supervisado utiliza un conjunto de datos de entrenamiento que incluye datos de entrada y salidas correctas, que permiten que el modelo aprenda con el tiempo. El algoritmo mide su precisión a través de la función de pérdida, ajustándose hasta que el error se haya minimizado lo suficiente.

El aprendizaje supervisado se puede dividir en dos tipos de problemas: regresión y clasificación.

La regresión se utiliza para comprender la relación entre variables dependientes e independientes. Se usa comúnmente para realizar proyecciones, como ingresos por ventas para una empresa determinada.

La clasificación utiliza un algoritmo para asignar con precisión datos de prueba a categorías específicas. Reconoce entidades específicas dentro del conjunto de datos e intenta sacar algunas conclusiones sobre cómo se deben etiquetar o definir esas entidades.

El aprendizaje supervisado utiliza varios algoritmos y técnicas de cálculo, como regresión lineal, regresión logística, máquinas de vectores de soporte (SVM), redes neuronales, naive bayes, k vecinos más cercanos (KNN) y bosques aleatorios.

El aprendizaje supervisado permite resolver una amplia variedad de problemas del mundo real a escala, como la clasificación de spam, el reconocimiento de imágenes, el diagnóstico médico, la detección de fraudes, entre muchos otros.

1. Regresión lineal

La regresión lineal es un modelo de aprendizaje supervisado que busca encontrar una relación lineal entre una variable dependiente y una o más variables independientes. La forma general de una regresión lineal simple es:

$$y = \theta_0 + \theta_1 x_1$$

Y de una regresión lineal múltiple:

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n + \varepsilon$$

Donde:

- Y, es la variable dependiente o respuesta, que se quiere predecir o explicar, se la denomina también como etiqueta o label.
- $x_1, x_2, \ldots, x_n$_son las variables independientes o predictoras, que usamos para estimar o influir en y. Se la denomina también variable de entrada, feature, característica o variable de un ejemplo o dato de entrada.
- $\theta_0, \theta_1, \ldots, \theta_n$, son los coeficientes o parámetros del modelo, que indican la contribución de cada variable independiente a la variable dependiente.
- ε, épsilon es el término de error o residuo, que representa la diferencia entre el valor observado y el valor estimado de y.

De manera más general una regresión lineal multiple se puede representar como:

$$y = \theta_0 + \sum_{i=0}^{n} \theta_i x_i$$

El objetivo de la regresión lineal es estimar los coeficientes θ del modelo a partir de un conjunto de datos de entrenamiento, que contiene observaciones de las variables dependiente e independientes. Para ello, se suele utilizar el método de mínimos cuadrados ordinarios (OLS), que consiste en minimizar la suma de los cuadrados de los errores:

$$\min_{\theta_0, \theta_1, \ldots, \theta_n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Donde:

- n, es el número de observaciones en el conjunto de datos de entrenamiento.
- y_i, es el valor observado de la variable dependiente para la i-ésima observación.
- $\hat{y}_i$, es el valor estimado de la variable dependiente para la i-ésima observación, que se obtiene sustituyendo los valores de las variables independientes en el modelo.

La solución de este problema de optimización se puede obtener mediante álgebra matricial, derivadas parciales o métodos numéricos. El resultado es un vector de coeficientes estimados, que se pueden usar para hacer predicciones o inferencias sobre la variable dependiente.

Para evaluar la calidad del ajuste del modelo, se pueden utilizar diferentes medidas, como el coeficiente de determinación (R^2), el error cuadrático medio (MSE), el error absoluto medio (MAE), el criterio de información de Akaike (AIC) o el criterio de información bayesiano (BIC). Estas medidas permiten comparar diferentes modelos y elegir el más adecuado según el contexto y el objetivo del análisis.

Ejemplo 1 de regresión lineal en Python

Para ilustrar cómo se puede implementar una regresión lineal en Python, vamos a usar un conjunto de datos sintético (Datos generados bajo ciertos criterios de aleatoriedad y consistencia) que contiene el salario anual de 30 empleados de una empresa en función de su experiencia laboral en años. El objetivo es construir un modelo de regresión lineal que prediga el salario de un empleado a partir de su experiencia.

Primero, importamos las librerías que vamos a necesitar:

```
import numpy as np # para operaciones matriciales
import pandas as pd # para manipulación de datos
import matplotlib.pyplot as plt # para visualización de datos
import statsmodels.api as sm # para regresión lineal
```

Luego, cargamos el conjunto de datos desde un archivo CSV y lo guardamos en un objeto de tipo DataFrame:

```
df = pd.read_csv("salario.csv")
df.head() # mostramos las primeras filas del DataFrame
```

	Experiencia	**Salario**
0	1.1	39343
1	1.3	46205
2	1.5	37731
3	2.0	43525
4	2.2	39891

A continuación, identificamos que la variable dependiente corresponde con el salario y la variable independiente con la experiencia y asignamos a dos listas (objetos de tipo array):

```
y = df["salario"] # variable dependiente
x = df["experiencia"] # variable independiente
```

Para ajustar un modelo de regresión lineal, usamos la función OLS de la librería statsmodels, que recibe como argumentos la variable dependiente y la variable independiente con una constante añadida. La constante representa el término independiente del modelo, que se puede omitir si se asume que es cero. El resultado es un objeto de tipo RegressionResults, que contiene toda la información del modelo.

```
model = sm.OLS(y, sm.add_constant(x)) # ajustamos el modelo
results = model.fit() # guardamos los resultados
```

Para ver un resumen del modelo, usamos el método summary del objeto results, que muestra los coeficientes estimados, sus errores estándar, sus valores p, el coeficiente de determinación, el criterio de información de Akaike, entre otras medidas.

```
results.summary() # mostramos el resumen del modelo
```

| | coef | std err | t | P>|t| | [0.025 | 0.975] | |---------------------------|---------|---------|--------|------|--------|--------| | const | 25792.2 | 2273.05 | 11.347 | 0.000 | 21223.1 | 30361.3 | | experiencia | 9449.96 | 378.76 | 24.950 | 0.000 | 8684.34 | 10215.6 | | | | | | | | | | Omnibus | 2.191 | | | | | | | Prob(Omnibus) | 0.335 | | | | | | | Skew | -0.379 | | | | | | | Kurtosis | 2.086 | | | | | | | Durbin-Watson | 1.283 | | | | | | | Jarque-Bera (JB) | 1.959 | | | | | | | Prob(JB) | 0.376 | | | | | | | Cond. No. | 14.4 | | | | | | | | | | | | | | | | Dep. Variable | salario | | | | | | | Model | OLS | | | | | | | Method | Least Squares | | | | | | | Date | Tue, 23 Jan 2024 | | | | | | | Time | 05:35:51 | | | | | | | No. Observations | 30 | | | | | | | Df Residuals | 28 | | | | | | | Df Model | 1 | | | | | | | R-squared | 0.957 | | | | | | | Adj. R-squared | 0.955 | | | | | | | F-statistic | 622.5 | | | | | | | Prob (F-statistic) | 1.14e-20 | | | | | | | Log-Likelihood | -301.44 | | | | | | | AIC | 608.9 | | | | | | | BIC | 612.9 | | | | | | | | | | | | | | | | Covariance Type | nonrobust | | | | | |

El resumen del modelo nos muestra que los coeficientes estimados son:

Theta_0 = 25792.2

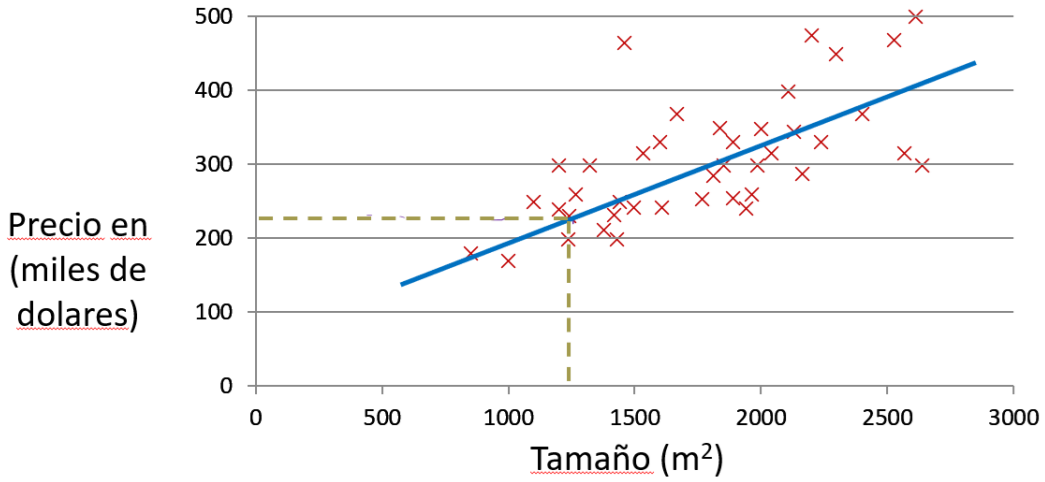
theta_1 = 9449.96

Esto significa que el modelo ajustado es:

y = 25792.2 + 9449.96 x

Ejemplo 2 de regresión lineal para la predicción del precio de una vivienda

Se disponen de datos que representan el costo de un inmueble en relación al área de su terreno, a partir de esto se procederá a generar un modelo que permita predecir el costo de inmuebles con tamaños de terrenos que no están contemplados en el dataset.



El dataset dispone de la siguiente información:

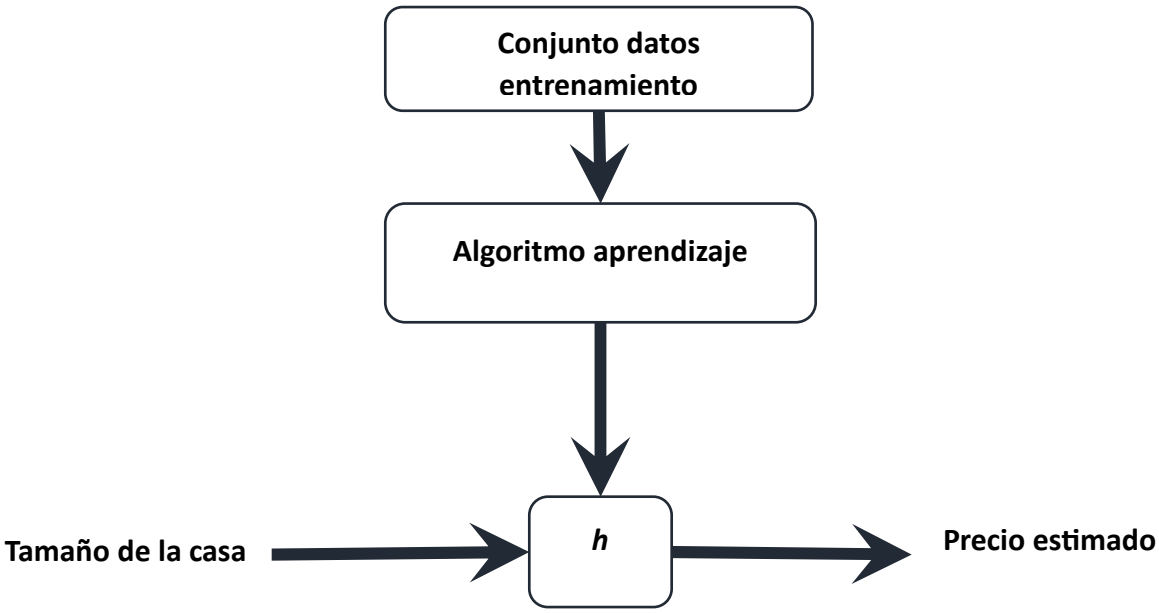
Tamaño en pies² (x)	**Precio ($) en 1000's (y)**
2104	460
1416	232
1534	315
852	178
…	…

Donde:

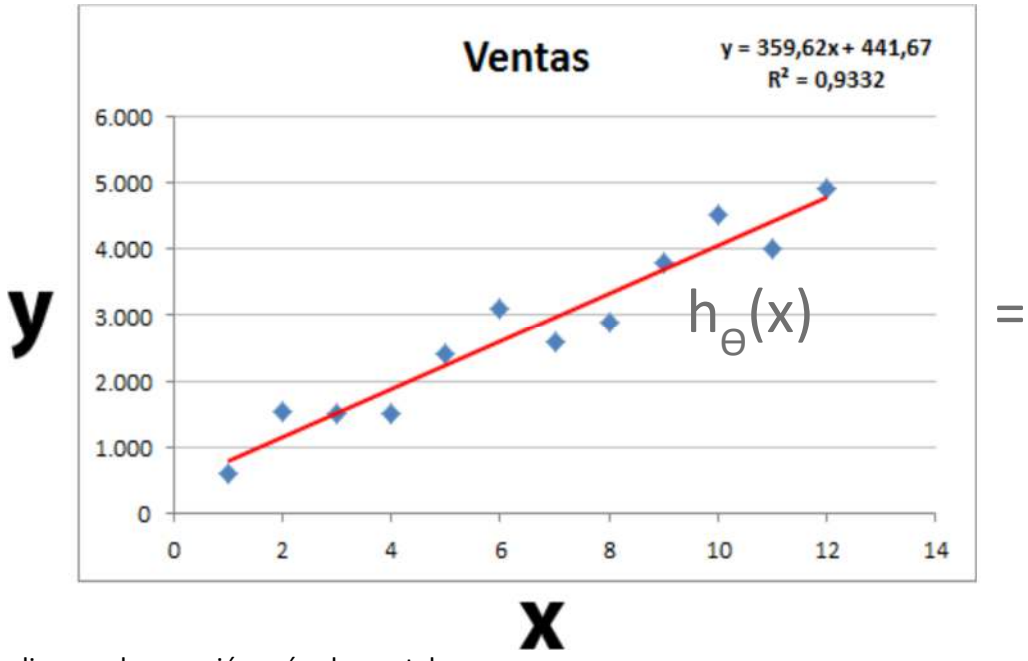
- m = Numero de ejemplos de entrenamiento.
- x's = variable "entradas" / caracteristicas
- y's = variable "salida" / variable "objetivo"
- (x, y) = un ejemplo de entrenamiento
- (x(i), y(i)) = iesimo ejemplo de entrenamiento

1.1 Hipótesis (H)

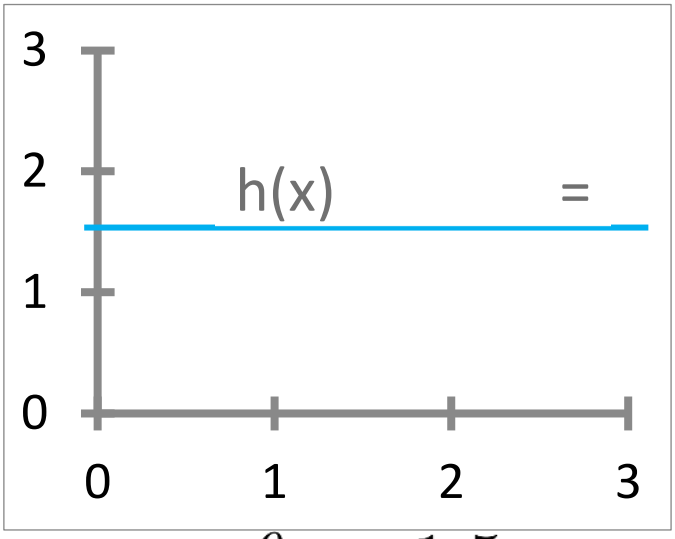
La hipótesis se constituye en la ecuación final, que considera los valores de theta calculados y reemplazados en la ecuación de la regresión, donde el resultado que se genera al aplicar con valores de x, corresponde con el resultado de y, que se estima, tratando que esta sea lo más cercana a la y generada en un ejemplo de los datos de entrenamiento.



La hipótesis se genera a partir de un proceso de aprendizaje, que considera características y valores de y conocidos para calcular los valores de theta, que puedan lograr representar efectivamente una relación que mejor represente a todos los ejemplos del dataset simultáneamente.

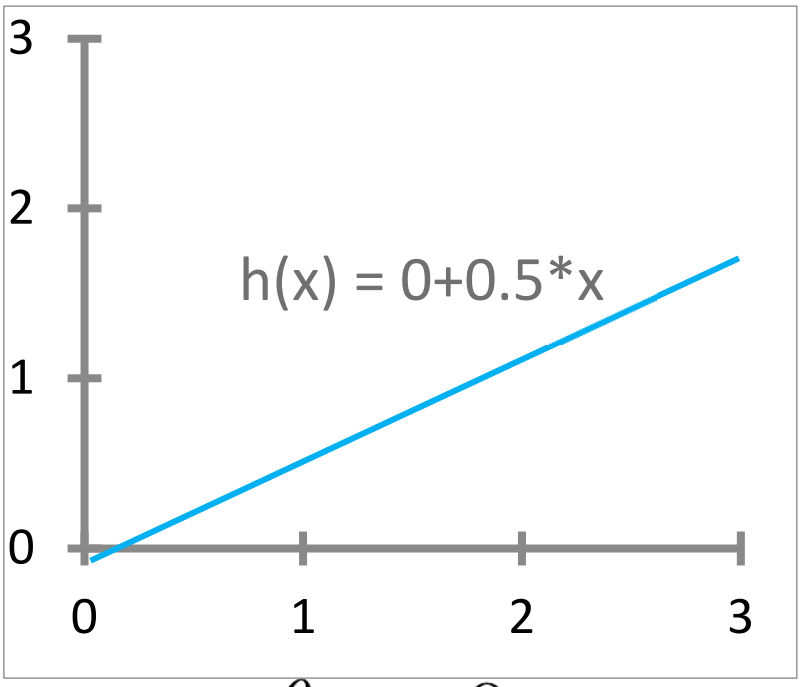


Analicemos la ecuación más elemental:



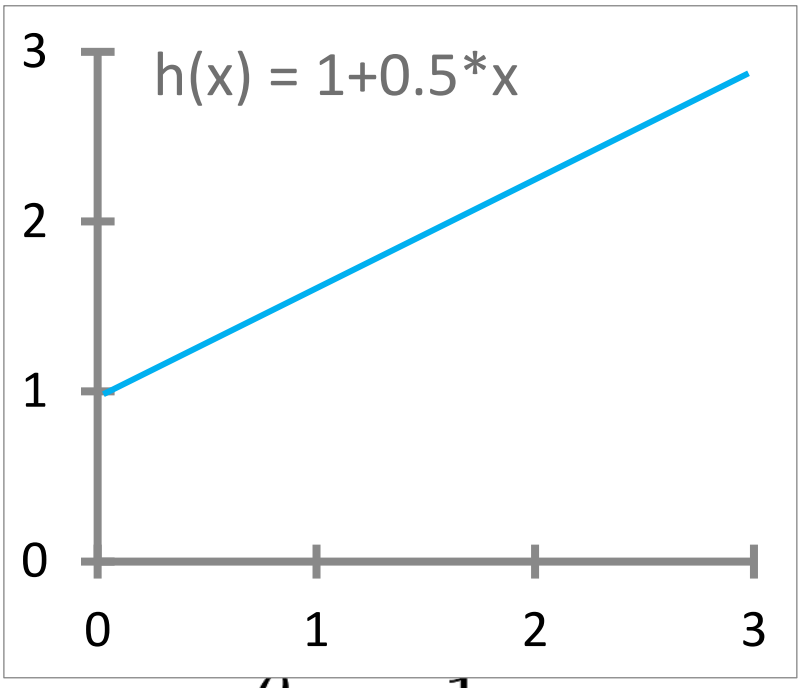
$$\theta_0 = 1.5$$

$$\theta_1 = 0$$



$$\theta_0 = 0$$

$$\theta_1 = 0.5$$



$$\theta_0 = 1$$

$$\theta_1 = 0.5$$

1.2 Función de costo (J)

Bajo el procedimiento que asume la construcción de un modelo, el paso más importante es el calcular los valores de theta que satisfagan de la mejor manera el cálculo de y a partir de x's, utilizando estos parámetros, sin embargo el cálculo de theta requiere que se ajusten estos, considerando la variación o diferencia que se produce con un determinado ejemplo de entrenamiento, para determinar esa diferencia se utiliza la función de costo.

Se busca Θ_0, Θ_1 de manera que se h(Θ) sea cercano a y, con los m datos de entrenamiento x, y.

Se utiliza la fórmula de diferencia de cuadrados:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

Hasta este punto las formulas mas importantes que se deben considerar son:

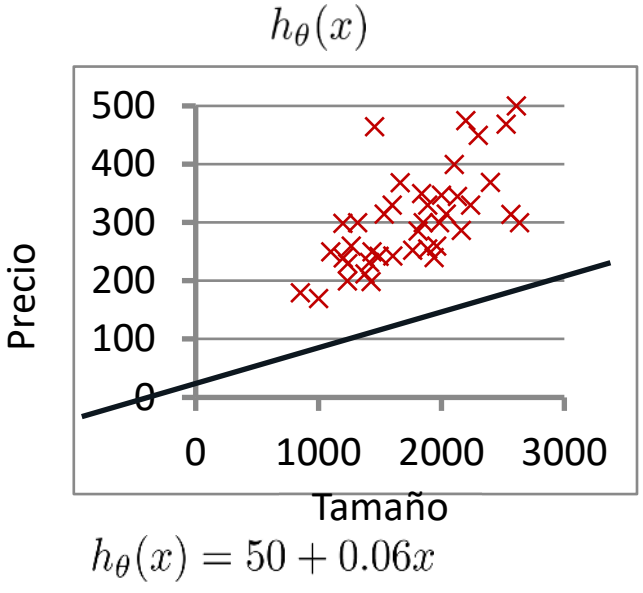
Hipótesis:
- $h_\theta(x) = \theta_0 + \theta_1 x$
- Si $\theta_0 = 0$; $h_\theta(x) = \theta_1 x$ (Simplificado)

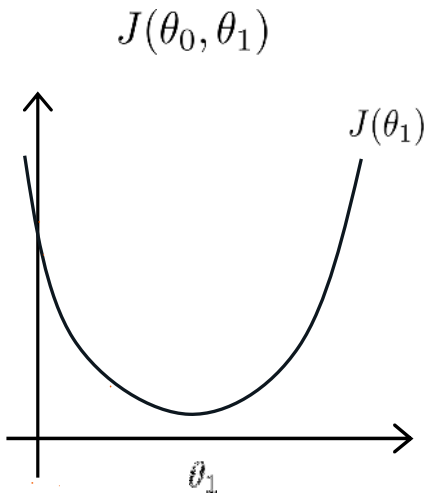
Parámetros:
- θ_0 , θ_1

Función de costo:
- $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$

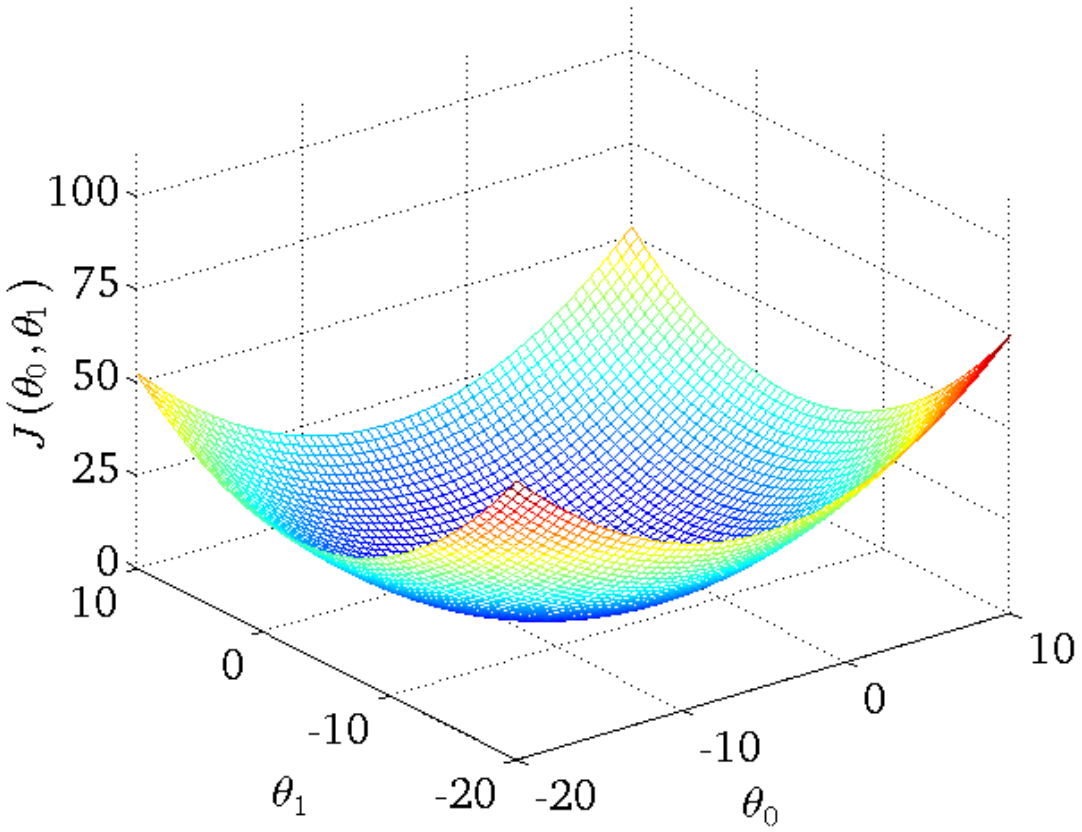
Objetivo:
- Minimizar $J(\theta_0, \theta_1)$

Se puede establecer una relación entre la función de hipótesis y la función de costo, a partir de los siguientes gráficos.


$h_\theta(x)$
500
400
300
200
100
0
Precio
0
1000
2000
3000
Tamaño
$h_\theta(x) = 50 + 0.06x$


$J(\theta_0, \theta_1)$
$J(\theta_1)$
θ_1

Considerando (θ_0, θ_1):



Permite observar que la función de costo ira decreciendo a medida que los valores de theta sean los más adecuados para la hipótesis. Sin embargo, la pregunta o desafío es determinar esos valores de theta que nos permitan lograr alcanzar el mínimo global en la función de costo. Si bien se pueden recurrir a métodos recursivos o numéricos, lo ideal es utilizar la técnica del descenso por el gradiente que se explica a continuación.

1.3 Descenso por el gradiente

Esta técnica lo que busca es tratar de encontrar los valores de theta que generen el menor valor posible para la función de costo.

Es decir:

Dada la función $J(\theta_0, \theta_1)$, se quiere minimizar $\min_{\theta_0,\theta_1} J(\theta_0, \theta_1)$

La estrategia consiste en:

- Asignar valores iniciales a θ_0, θ_1
- Realizar y almacenar cambios en θ_0, θ_1

- Hasta alcanzar un mínimo para $J(\theta_0, \theta_1)$

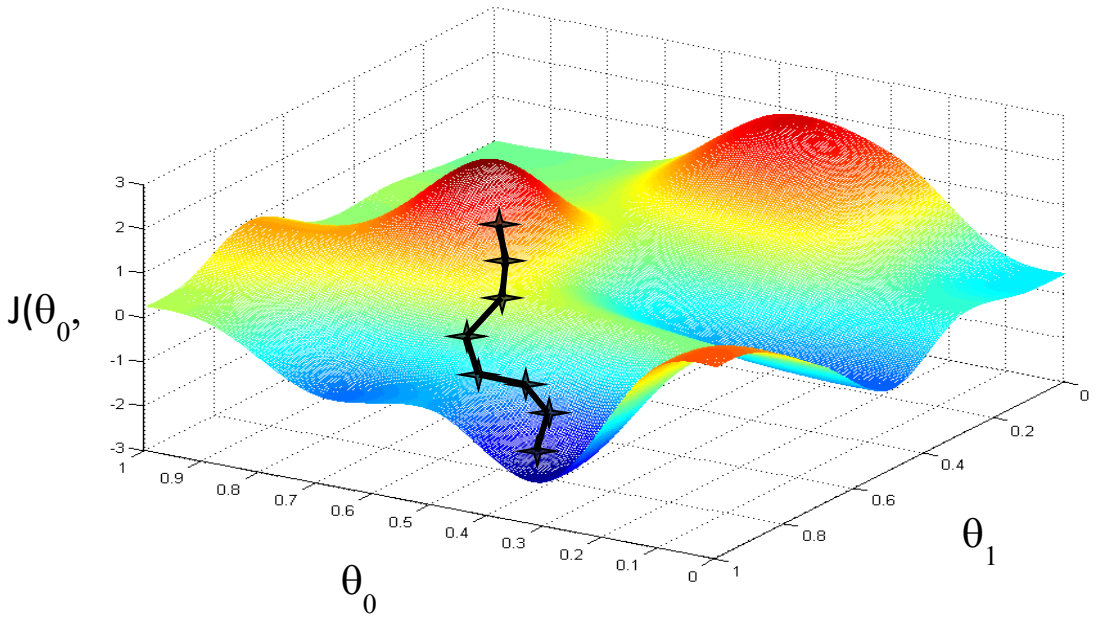
Sin embargo, no convendría hacer esto de manera arbitraria y aleatoria, por esto es que se asume la utilización de las siguientes formulas:

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)$$

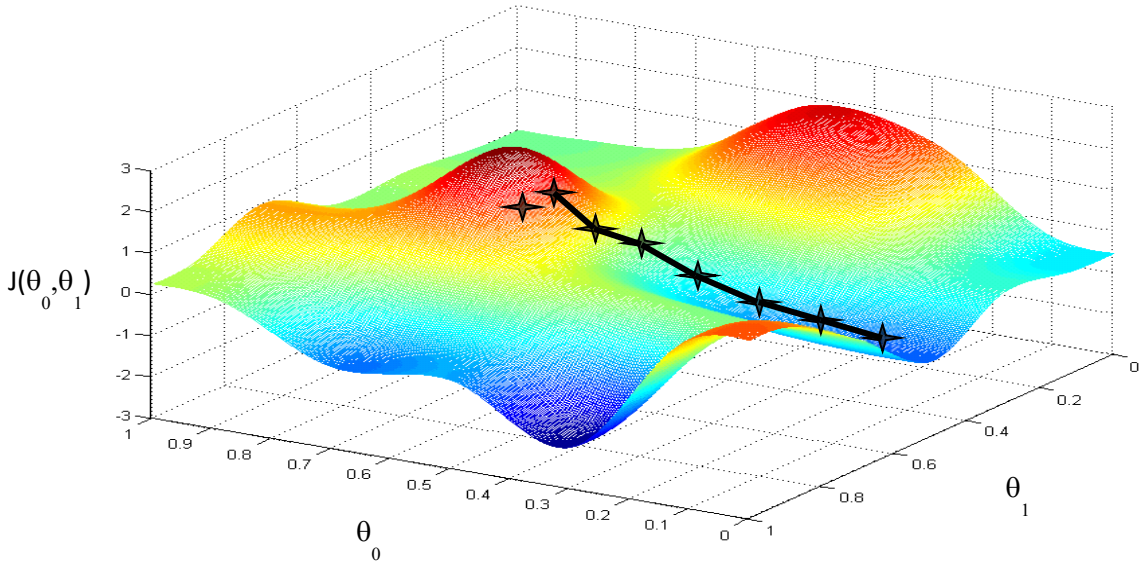
$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) \cdot x^{(i)}$$

Permitiendo calcular un nuevo valor para thetas, que reducen el error integral de todos los elementos asumidos para x hasta un determinado momento. Es decir se debe repetir este calculo considerando todos los valores de x y y que se dispongan y por un numero de veces que permita lograr una convergencia a cero de la función de costo.

La grafica del descenso por el gradiente es algo parecido a lo siguiente:



O también podría ser:



Podemos formalizar el algoritmo del descenso por el gradiente de la siguiente manera:

Repetir hasta que converja (Para j = 0 y j = 1) {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

}

Se debe considerar siempre lo siguiente:

Forma correcta de actualización de los valores de theta	Forma incorrecta de actualización de los valores de theta
$\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$ $\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$ $\theta_0 := \text{temp0}$ $\theta_1 := \text{temp1}$	$\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$ $\theta_0 := \text{temp0}$ $\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$ $\theta_1 := \text{temp1}$

1.4 Coeficiente de aprendizaje alfa (α)

Si bien la ecuación presentada anteriormente es la que permite determinar los valores de theta utilizando la ecuación y algoritmo del descenso por el gradiente, es muy importante comprender la importancia y efecto que tienen los diferentes valores que se pueden asignar al coeficiente alfa.

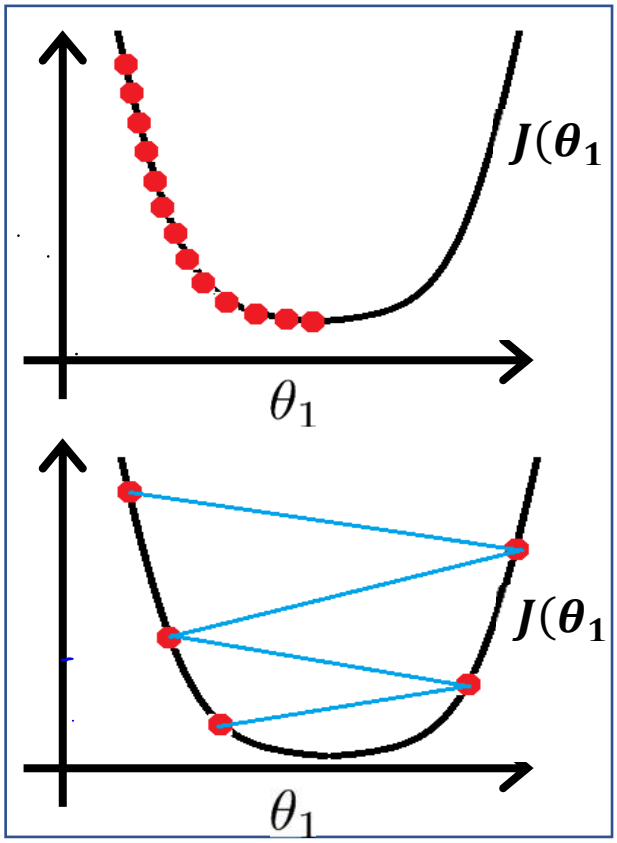
Para realizar este análisis asumiremos la siguiente ecuación:

$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

Donde se producirán las siguientes circunstancias:

- SI α es muy pequeño, el descenso por el gradiente es lento.
- Si α es muy grande, el descenso por el gradiente puede saltar el mínimo. Y fallar en su propósito de converger a un mínimo global.
- El valor de α puede cambiar a conveniencia, sin embargo aun cuando este estuviese fijo, lograra converger.

Como se puede apreciar en las siguientes graficas:



En el primer caso si Alpha es demasiado pequeño y el segundo si Alpha es demasiado grande.

1.5 Regresión lineal múltiple

Sin embargo, es muy difícil que cuando se trabaje con dataset reales, estos solo dispongan de una sola característica, al contrario, los dataset que se utilizan en trabajos serios y reales disponen de bastantes características, como ejemplo presentamos la siguiente tabla:

x_1 Tamaño (m^2)	x_2 Número dormitorios	x_3 Numero pisos	x_4 Antiguedad de la casa (años)	y Precio ($1000)
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
…	…	…	…	…

Lo que establecerá que se conforme la siguiente ecuación:

$$\boldsymbol{h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4}$$

Donde se debe asumir la siguiente notación:

- n = número de características $\quad x_1, x_2, x_3, \dots, x_n$
- $x^{(i)}$= entradas (catacterísticas) i^{esimo} de ejemplo de entrenamiento.
- $x_j^{(i)}$= valor de la característica j en el i^{esimo} ejemplo de entrenamiento.

La hipótesis para una regresión lineal múltiple asume la siguiente forma:

Por conveniencia asumiremos: $\boldsymbol{x_0 = 1}$

Donde:

$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ . \\ . \\ . \\ x_n \end{bmatrix}$$

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ . \\ . \\ . \\ \theta_n \end{bmatrix}$$

$$\theta^T x = [\theta_0 \quad \theta_1 \quad \theta 2_1 \ldots \theta_n] \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ . \\ . \\ . \\ x_n \end{bmatrix}$$

Lo que permite formular la hipótesis como:

$$\boldsymbol{h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots \theta_n x_n}$$

O de forma resumida:

$$\boldsymbol{h_\theta(x) = \theta^T x}$$

A partir de esto se pueden establecer las siguientes formulas:

Hipótesis: $h_\theta(x) = \theta^T x = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$

Donde los parámetros son: $\theta_0, \theta_1, \ldots, \theta_n$

La función de costo: $J(\theta_0, \theta_1, \ldots, \theta_n) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$

Y el descenso por el gradiente:

Repetir $\{$

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \ldots, \theta_n)$$

$\}$ (actualiza simultaneamente cada $j = 0, \ldots, n$

Detallando mas el descenso por el gradiente:

Nuevo algoritmo: $(n \geq 1)$

Repetir $\{$

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

(actualiza simultaneamente θ_j para $j = 0, \ldots, n$)

$\}$

Desglosando:

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_0^{(i)}$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_1^{(i)}$$

$$\theta_2 := \theta_2 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_2^{(i)}$$

1.6 Escalado de características

Cuando se trabaja con n características estas pueden responder a diferentes escalas de valores, lo cual, si bien no impide que el algoritmo del descenso por el gradiente tienda a converger, el tiempo requerido es muy amplio o en su caso se presentaras valores demasiado grandes o pequeños para ser representados por un computador lo cual genera un error de cómputo que impide la culminación exitosa del algoritmo, por lo cual una estrategia obligada que se debe asumir es el escalado de los valores de las diferentes características.

Si bien establecer una escala general entre 0 y 1 es bueno, y es utilizado con mucha regularidad, también se puede utilizar una escala entre -1 y 1, sin embargo, si alguna característica estuviera en un rango cercano al rango asumido, no debería existir dificultad alguna en su convergencia, por lo que no sería necesario aplicar una técnica de escalado .

Para establecer una escala entre 0 y 1, se puede simplemente realizar una división de cada valor de una característica entre su valor mayor.

Una técnica mas común es de escalado es la normalización media, que consiste en aplicar la siguiente formula a todas las características excepto a x_0.

$$x_i = \frac{x_i - \mu_i}{\sigma_i}$$

Donde μ_i es el promedio de los valores de la característica i, σ_i es la desviación estándar, considerada como la diferencia del mayor valor – menor valor entre los valores de la característica

Reemplazar x_i con $\frac{x_i - \mu_i}{\sigma_i}$ para que las funciones tengan una media de aproximadamente cero. (No aplicar a x_0).

Considerando el ejemplo:

$$x_1 = \frac{size - 1000}{2000}$$

$$x_2 = \frac{\#bedrooms - 2}{5}$$

Aplicando la fórmula de normalización media se logrará que:

$$-0.5 \leq x_1 \leq 0.5, -0.5 \leq x_2 \leq 0.5$$

Es importante recordar que la normalización garantiza la convergencia del descenso por el gradiente, su aplicación es de carácter obligatoria en la mayoría de los casos.

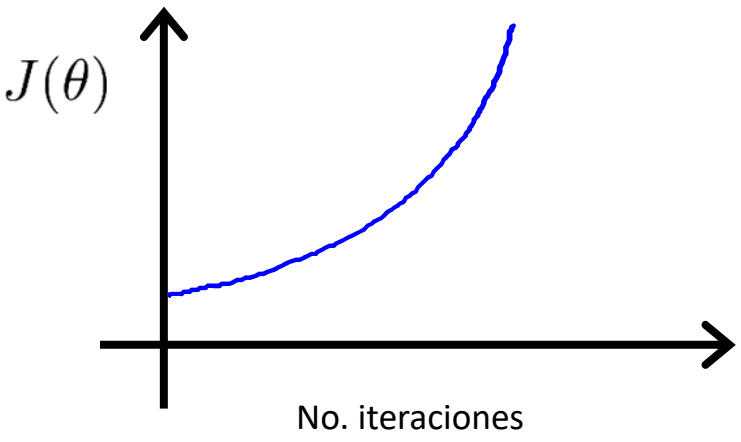
1.7 Sugerencia para el descenso por el gradiente

La forma general del descenso por el gradiente es:

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

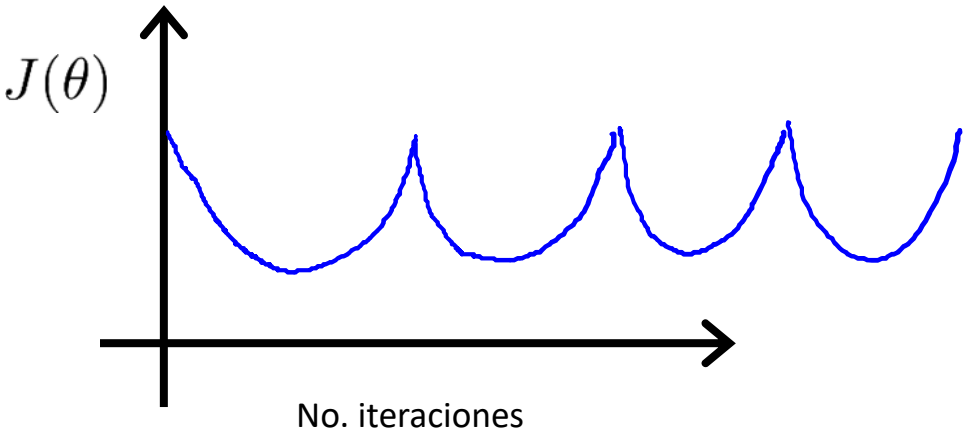
Un desafío interesante es la elección del valor adecuado de alfa.

Si la gráfica de la función del costo se parece a:

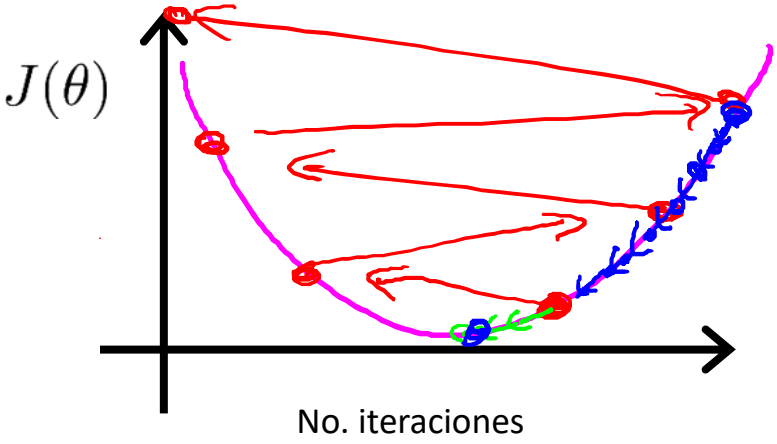


Significa que el descenso por el gradiente no está funcionando.

Si tendría una forma como esta:



O una gráfica similar a esta:



Se debe utilizar un Alpha más pequeño.

Donde:

Para un alfa suficiente pequeño, la función de costo debe decrecer en cada iteración, si alfa es muy pequeño, el descenso por el gradiente puede converger lentamente.

Si alfa es muy pequeño: convergencia lenta.

Si alfa es muy grande la función de costo no decrece en cada iteración, no converge

Para elegir alfa, ensayar con: …, 0.001, 0.003, 0.01, 0.03, 0.1, 0,3. 1, …

1.8 Regresión Polinómica

La regresión polinomial te permite utilizar la maquinaria de la regresión lineal para ajustar funciones muy complicadas, incluso no lineales. Tomemos el ejemplo de predecir el precio de una casa. Supongamos que tienes dos variables, la fachada de la casa y la profundidad de la casa. Por lo que, aquí está la imagen de la casa que estamos tratando de vender. Así que, la fachada se define como esta distancia y es básicamente el ancho o la longitud del ancho de tu lote si esta es tu propiedad, y la profundidad de la casa se refiere a que tan profunda es tu propiedad, así que hay una fachada, hay una profundidad. Así tienes dos variables llamadas fachada y profundidad. Es posible construir un modelo de regresión lineal como este en donde la fachada es tu primera variable x1 y la profundidad es tu segunda variable x2, pero cuando estás aplicando la regresión lineal, no necesariamente tienes que usar solamente las variables x1 y x2 que te dan. Lo que puedes hacer es crear nuevas variables por ti mismo. Así, si quiero predecir el precio de una casa, lo que podría hacer en su lugar es decidir que lo que realmente determina el tamaño de la casa es el área o el área de la tierra que me pertenece. Así, podría crear una nueva variable. Voy a llamar a esta función "x" que es la fachada, multiplicada por la profundidad. Este es un símbolo de multiplicación. Es la fachada multiplicada por la profundidad porque esta es el área de la tierra que me pertenece y entonces puedo seleccionar mi hipótesis así, utilizando solamente una variable que es el área de mi tierra, ¿correcto? Ya que el área de un rectángulo es como sabes, el producto de la longitud de sus lados.

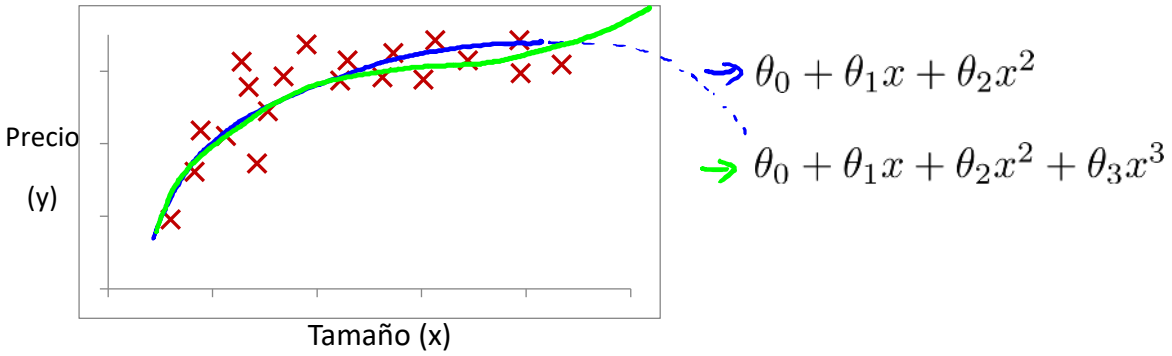
$$h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2$$

$$h_\theta(x) = \theta_0 + \theta_1 * frente + \theta_2 * fondo$$

$$x = frente * fondo$$

$$h_\theta(x) = \theta_0 + \theta_1 x$$

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



$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$$
$$= \theta_0 + \theta_1 (size) + \theta_2 (size)^2 + \theta_3 (size)^3$$
$$x_1 = (size)$$
$$x_2 = (size)^2$$
$$x_3 = (size)^3$$

Así es que, dependiendo del entendimiento que puedas tener sobre un problema particular, en lugar de simplemente tomar las variables fachada y profundidad que son las que nos han dado para comenzar, a veces mediante la definición de nuevas variables en realidad podrías conseguir un mejor modelo.

Estrechamente relacionada con la idea de elegir tus variables está la idea llamada regresión polinomial. Digamos que tienes un conjunto de datos de precios de vivienda que tienen este aspecto. Entonces hay algunos modelos diferentes que podrías ajustar a esto. Una cosa que podrías hacer es ajustar un modelo cuadrático así. No parece que una línea recta se ajuste muy bien a estos datos. Así que tal vez quieras ajustar un modelo cuadrático como éste en donde piensas que el tamaño, en donde piensas que el precio es una función cuadrática y tal vez eso te puede dar, como sabes, un ajuste a los datos que se ve así. Pero entonces puedes decidir que tu modelo cuadrático no tiene sentido con una función cuadrática, porque eventualmente esta función vuelve a bajar y bien, no creemos que los precios de vivienda deban bajar mientras que el tamaño sube tan alto. Entonces tal vez podamos elegir un modelo polinomial diferente y optar por utilizar en su lugar una función cúbica, en donde tenemos ahora un término de tercer orden y ajustamos eso, tal vez obtenemos este tipo de modelo, y tal vez la línea verde se ajusta un poco mejor a los datos ya que no volverá a bajar eventualmente. Así que ¿cómo podemos ajustar un modelo como este a nuestros datos?

Aplicando diferentes modificaciones de las características, y aplicarlas en la hipótesis como:

$$h_\theta(x) = \theta_0 + \theta_1 (size) + \theta_2 (size)^2$$

$$h_\theta(x) = \theta_0 + \theta_1(size) + \theta_2\sqrt{(size)}$$

Sólo queda señalar una cosa más, y es que si se eligen variables de esta forma, entonces el escalamiento de variables se hace cada vez más importante. Así que si el tamaño de la casa está dentro del rango de uno a mil, entonces, como sabes, de uno a mil pies cuadrados, digamos, entonces el tamaño al cuadrado de la casa estará en el rango de uno a un millón, el cuadrado de mil, y tu tercera variable x al cubo, tu tercera variable x3 que es el tamaño al cubo de la casa, estará en el rango de uno a diez a la novena potencia, y así estas tres variables adquieren muy diferentes rangos de valor, y es importante aplicar el escalamiento de variables si estás usando el gradiente de descenso para ponerlos en rangos de valores comparables. Para terminar, aquí hay un último ejemplo de cómo tienes realmente amplias opciones en las funciones que utilizas. Anteriormente hablamos de cómo un modelo cuadrático como este podría no ser lo ideal porque, como sabes, tal vez un modelo cuadrático se ajusta bien a los datos, pero la función cuadrática vuelve a bajar y realmente no queremos, ¿correcto? que los precios de la vivienda bajen, para predecir eso, mientras el tamaño de la vivienda se congela. Pero en lugar de usar un modelo cúbico ahí, tienes, tal vez, otras opciones de variables y hay muchas opciones posibles.

La regresión polinomial, se debe considerar como la técnica de ajustar un polinomio, como una función cuadrática, o una función cúbica, a tus datos. También te di esta idea, de que tienes la elección de qué variables usar, como en vez de utilizar la fachada y la profundidad de la casa, tal vez, puedes multiplicarlas juntas para obtener una variable que capturé el área del terreno de una casa. En caso de que esto parezca un poco desconcertante, con todas esas opciones de variables diferente, ¿cómo decido qué variables utilizar, ahora solo tienes que estar consciente de que tienes opciones en cuanto a qué variables utilizar, y mediante el diseño de diferentes variables puedes ajustar funciones más complejas a tus datos, solamente ajustando una línea recta a los datos y en particular puedes poner funciones polinomiales también y a veces con el conocimiento apropiado de la variable te permite obtener un modelo mucho mejor para tus datos.

El resto de ecuación se aplican de manera similar a la regresión lineal multivariable.

1.9 Ecuación de la normal

Existe una forma alternativa de calcular los valores adecuados de theta para minimizar la función de costo, es decir los valores optimos, este método es analítico y se denomina la ecuación de la normal, en este método, minimizaremos J tomando explícitamente sus derivadas con respecto a los θj 's, y poniéndolas a cero. Esto nos permite encontrar el theta óptimo sin iteración. La fórmula de la ecuación normal se da a continuación:

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

Para explicar la aplicación del método utilizaremos los siguiente datos:

	Size (feet^2)	Number of bedrooms	Number of floors	Age of home (years)	Price ($1000)
1	2104	5	1	45	460
1	1416	3	2	40	232
1	1534	3	2	30	315
1	852	2	1	36	178

De donde se tiene:

$$X = \begin{bmatrix} 1 & 2104 & 5 & 1 & 45 \\ 1 & 1416 & 3 & 2 & 40 \\ 1 & 1534 & 3 & 2 & 30 \\ 1 & 852 & 2 & 1 & 36 \end{bmatrix}$$

Y

$$y = \begin{bmatrix} 460 \\ 232 \\ 315 \\ 178 \end{bmatrix}$$

Una vez que se tiene identificadas las x's e y, se sustituye en la ecuación y se obtiene los valores óptimos de theta.

Sin embargo, es importante considerar lo siguiente antes de aplicar y sea el descenso por el gradiente o la ecuación de la normal para calcular los valores de theta.

Descenso por el gradiente	Ecuación de la Normal
• Se requiere elegir el coeficiente de aprendizaje alpha • Se requiere muchas iteraciones • Trabaja muy bien aun cuando n es grande	• No requiere elegir el coeficiente de aprendizaje alpha • No necesita iterar. • Necesita calcular: $(X^T X)^{-1}$ • Lento si n es muy grande • La dificultad es cuando no se puede realizar la inversión: $X^T X$

	• Esto ocurre muy rara vez. Generalmente hay dos casos comunes: ○ La primera causa es que, de alguna manera, en el problema de aprendizaje tiene características redundantes. Por ejemplo si trata de predecir precios de casa y si x1 es el tamaño de un casa en metros cuadrados y x2 es el tamaño en pies cuadrados. Como 1 m cuadrado es igual a 3.28 pies, x1 = (3.28)*x2, de manera que si dos características están relacionadas por una ecuación como la anterior esa matriz no es invertible. ○ La segunda causa se debe a que cuando se entrena un algoritmo con muchas características (m < n)

La recomendación es que se aplique la ecuación de la normal siempre que se pueda, por la rapidez y precisión que otorga a los valores de theta. Otra limitante que tiene este método es que esta limitado por la cantidad de memoria disponible, pues la cantidad de parámetros y ejemplos pueden generar una matriz demasiado grande que no pueda ser calculada o el procesador tarde demasiado en realizar el cálculo.