

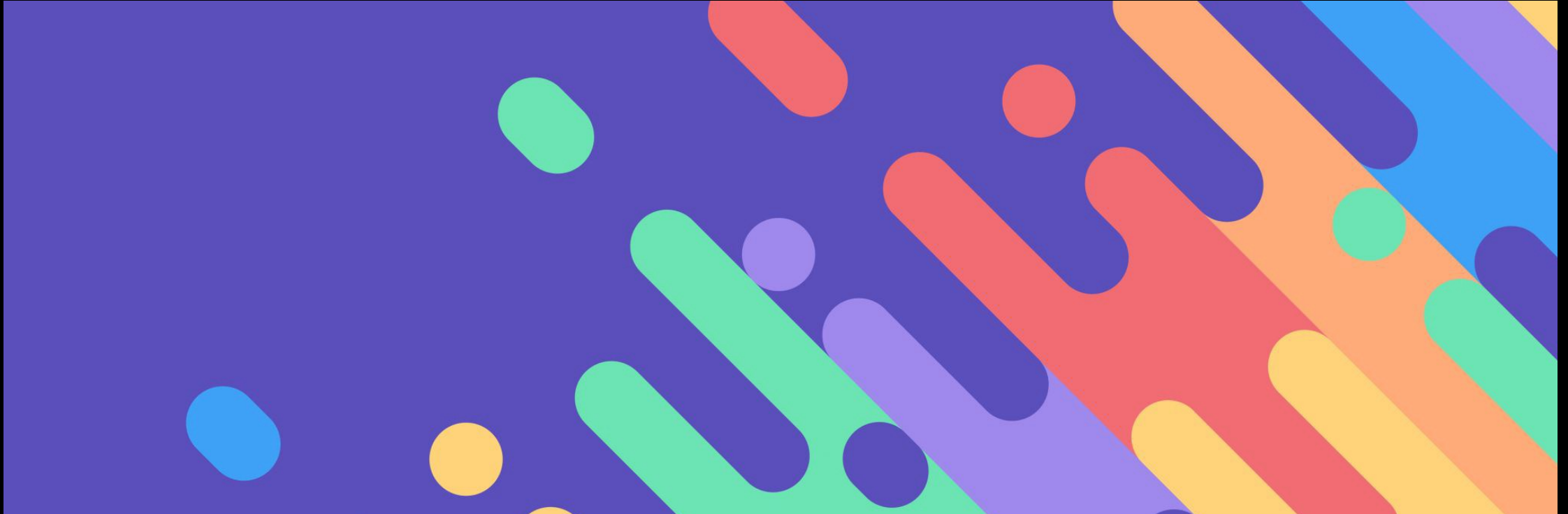
# REGRESIÓN LINEAL



Inteligencia Artificial

CEIA - FIUBA

Dr. Ing. Facundo Adrián  
Lucianna



---

LO QUE VIMOS LA CLASE ANTERIOR...

---

# APRENDIZAJE AUTOMÁTICO

Aprendizaje automático se entiende a:

Una computadora observa algunos datos, construye un modelo basado en estos datos, y usa el modelo como una hipótesis sobre el mundo y como una pieza de software que puede resolver problemas.

*¿Pero porque queremos que una computadora aprenda? ¿Por qué no programar el modelo directamente?*

- *No se puede anticipar todas las posibles situaciones futuras.*
- *No se tiene idea de cómo programar una solución por uno mismo.*



---

# DATOS

Lo más importante en Aprendizaje Automático (y en Data Science en general) son los

## Datos

Nos permite describir un objeto al que podemos llamar *entidad*.

Esta entidad y su información puede ser diferente a pesar de que describa un mismo objeto. La forma en que se elija representar los datos no solo afecta la forma en que se construyen sus sistemas, sino también los problemas que sus sistemas pueden resolver.

Por ejemplo, queremos representar un auto:

- En un problema para compra y venta de autos, podemos representarlo con el fabricante, modelo, año, color, y su precio.
- En un problema de un sistema de seguimiento policial, podemos representarlo por quien es el dueño, patente y su historia de direcciones registradas.

---

# DATOS

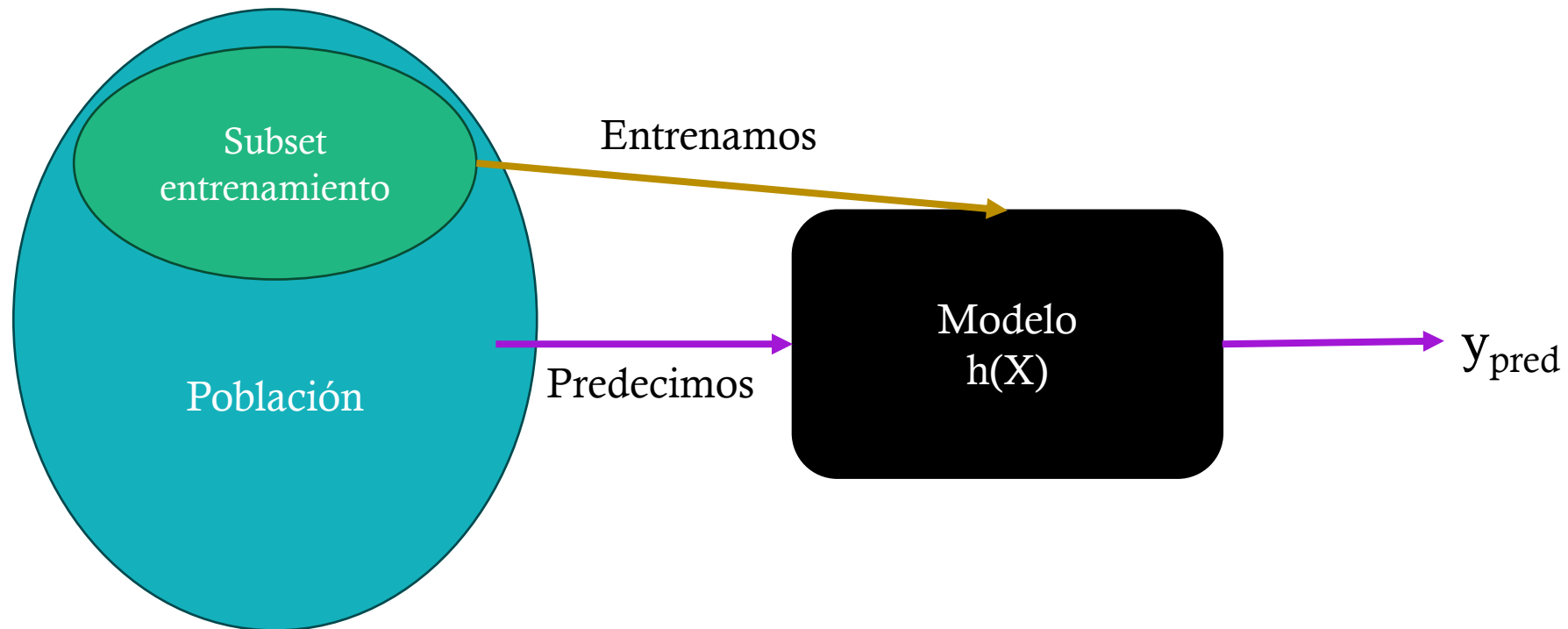
Observación →

Atributos/features					Objetivo
Position	Experience	Skill	Country	City	Salary (\$)
Developer	0	1	Argentina	Buenos Aires	103100
Data Scientist	2	2	Uruguay	Montevideo	104900
Developer	3	1	Argentina	Chivilcoy	106800
QA Eng	2	2	Colombia	Bogotá	108700
Product Manager	1	5	Perú	Lima	110400
Developer	7	5	Paraguay	Asunción	112300
Cloud Eng	5	2	Argentina	Buenos Aires	116100

---

# FORMAS DE APRENDIZAJE

Un esquema de aplicar Aprendizaje Automático nos queda...



---

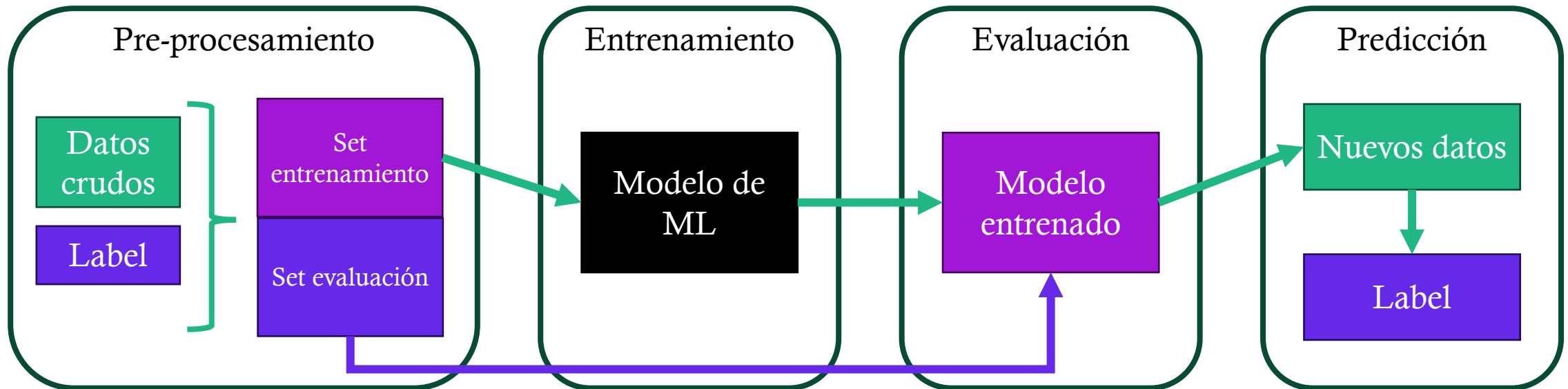
# FORMAS DE APRENDIZAJE

## Tipos de aprendizajes

- **Aprendizaje supervisado:** El modelo observa pares de entradas-salidas y aprende la relación entre ellos. Es decir, en este tipo de aprendizaje, conocemos el valor de  $y$  y se lo enseñamos al modelo.
- **Aprendizaje no supervisado:** El modelo aprende patrones de la entrada sin ninguna retroalimentación. Es decir, no contamos con  $y$  de antemano.
- **Aprendizaje por refuerzo:** El agente aprende con una serie de refuerzos: recompensas y castigos. Depende del agente decidir cuál de las acciones anteriores al refuerzo fue la más responsable de él y modificar sus acciones para apuntar a más recompensas en el futuro.

# APRENDIZAJE SUPERVISADO

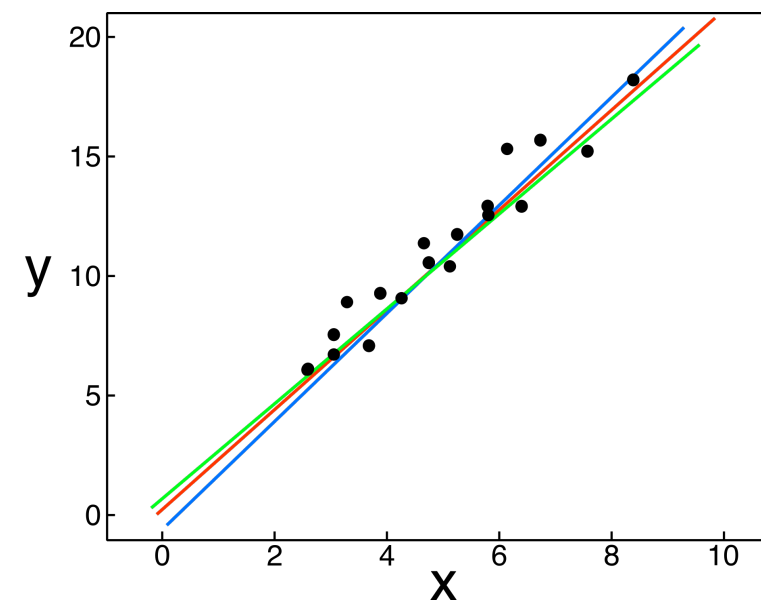
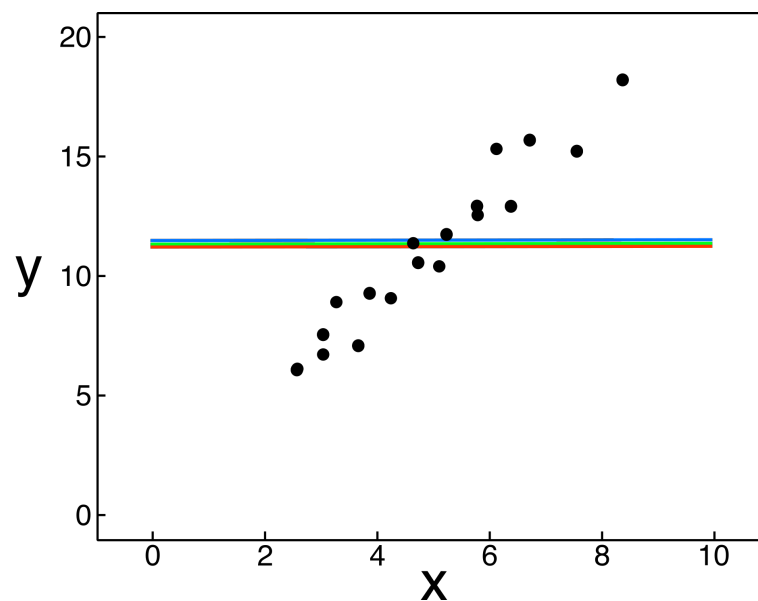
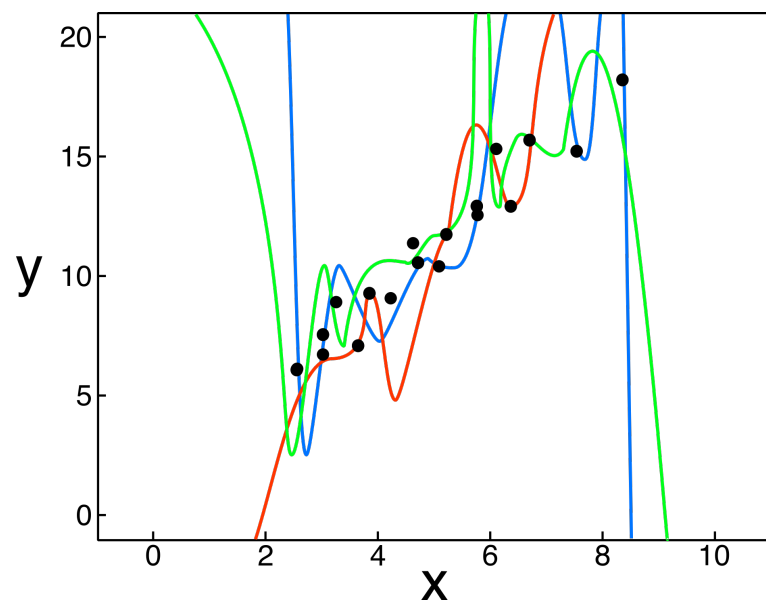
## Generalización





---

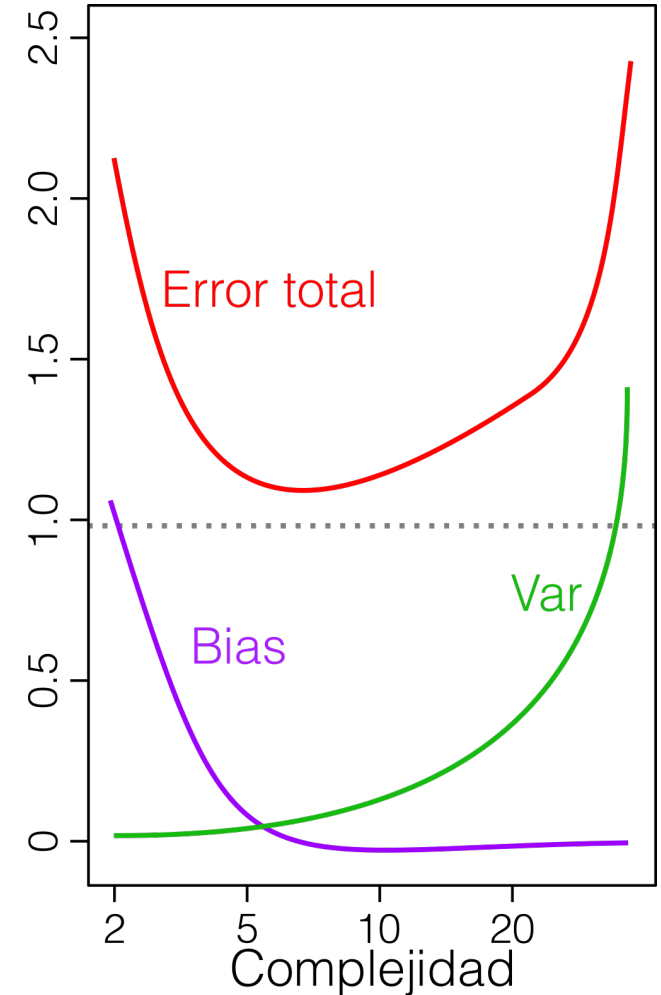
# SESGO Y VARIANZA



# SESGO Y VARIANZA

*Como regla general,*

- Cuando más complejo es el modelo, la varianza va a aumentar y el sesgo va a disminuir.
- Cuando aumentamos la complejidad de este, el sesgo tiende a disminuir más rápido de lo que la variabilidad aumenta, disminuyendo el error.
- Llega un punto en donde el efecto de la variabilidad es apreciable, aumentando el valor del error de nuevo.







---

# MÉTRICAS DE CLASIFICACIÓN

Entonces, es importante para saber si el clasificador binario es bueno o malo, entender cómo se puede equivocar.

Supongamos que, si el clasificador dice que es SPAM, entonces la salida es positiva, y si no es negativo, con eso podemos tener los siguientes casos en comparación con el verdadero valor

		Valor verdadero	
			
Salida del clasificador		Verdadero positivo (TP)	Falso positivo (FP)
		Falso negativo (FN)	Verdadero negativo (TN)

Esta estructura se llama **matriz de confusión**

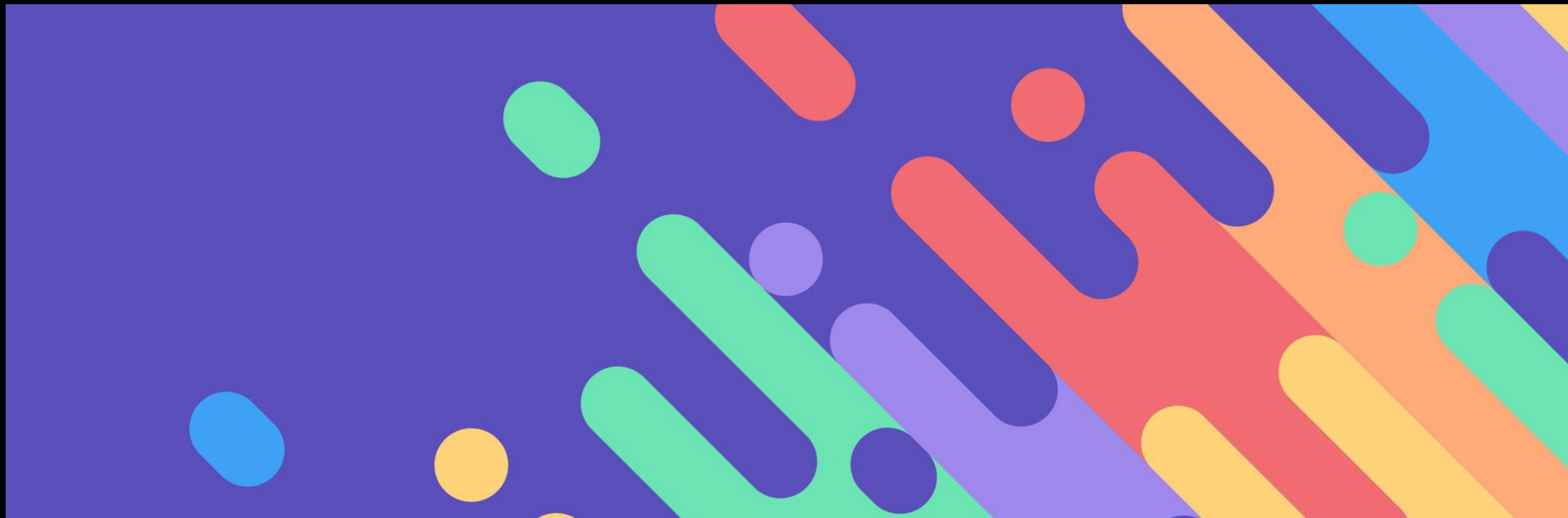
---

---

# MÉTRICAS DE CLASIFICACIÓN

## Matriz de confusión

- **Verdadero positivo:** Es aquellas observaciones que clasificamos como 1 y que realmente eran 1.
- **Verdadero negativo:** Es aquellas observaciones que clasificamos como 0 y que realmente eran 0.
- **Falso positivo:** Es aquellas observaciones que clasificamos como 1 y que realmente eran 0. Este tipo de error se llaman de tipo I.
- **Falso negativo:** Es aquellas observaciones que clasificamos como 0 y que realmente eran 1. Este tipo de error se llaman de tipo II.



---

# REGRESIÓN

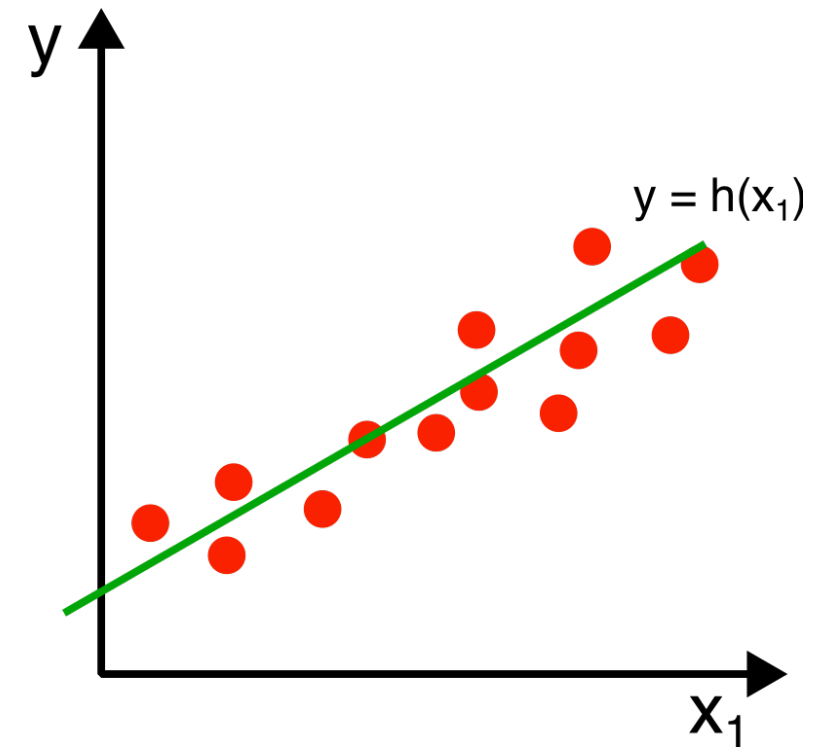
---

# REGRESIÓN

Si tenemos un problema donde el target  $y$  es una *variable numerica*, se llama un **problema de regresión**.

Se centra en estudiar las relaciones entre una variable dependiente de una o más variables independientes.

Es importante notar que, en Aprendizaje Automático, cuando buscamos una  $h(X)$  estamos armando un modelo puramente empírico. Es decir, nos basamos 100% en los datos medidos. En contraste con los modelos basados en propiedades fundamentales.





---

# REGRESIÓN LINEAL

---

# REGRESIÓN LINEAL

El modelo de regresión lineal más simple es el que involucra una combinación lineal de las variables de entradas:

$$\hat{y} = h(X) = b + w_0x_0 + \dots + w_dx_d$$

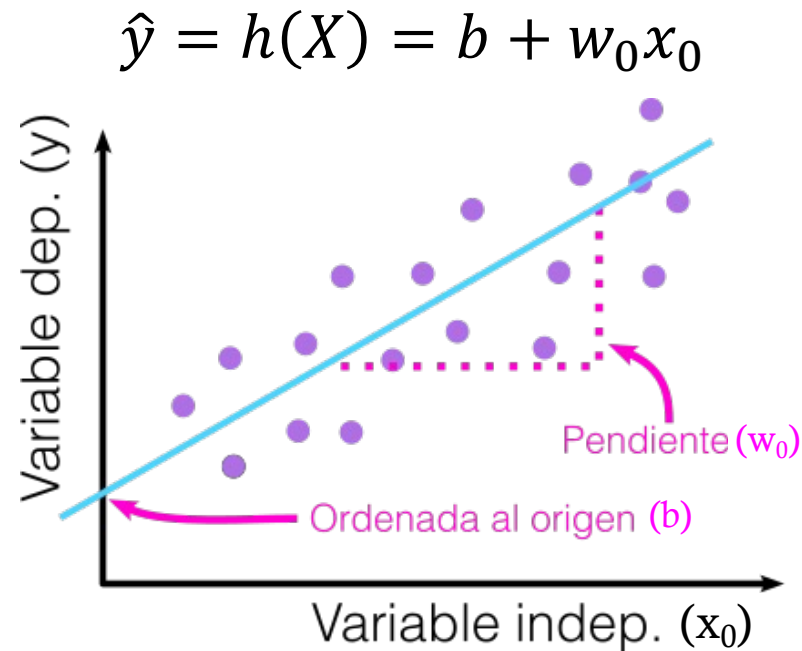
- $X = (x_0, x_1, \dots, x_d)$  Son los *features* de nuestras observaciones. Son todas variables numéricas
- $b, w_0, \dots, w_d$  Son los coeficientes del modelo. Son números reales. Cuanto más cerca de cero, la variable dependiente depende menos del *feature* que multiplica.
- $\hat{y}$  Es la predicción del modelo. Es con quien comparamos con el *Label* de la observación



---

# REGRESIÓN LINEAL

Vamos al caso más sencillo, la regresión lineal de una sola variable independiente:



---

# REGRESIÓN LINEAL

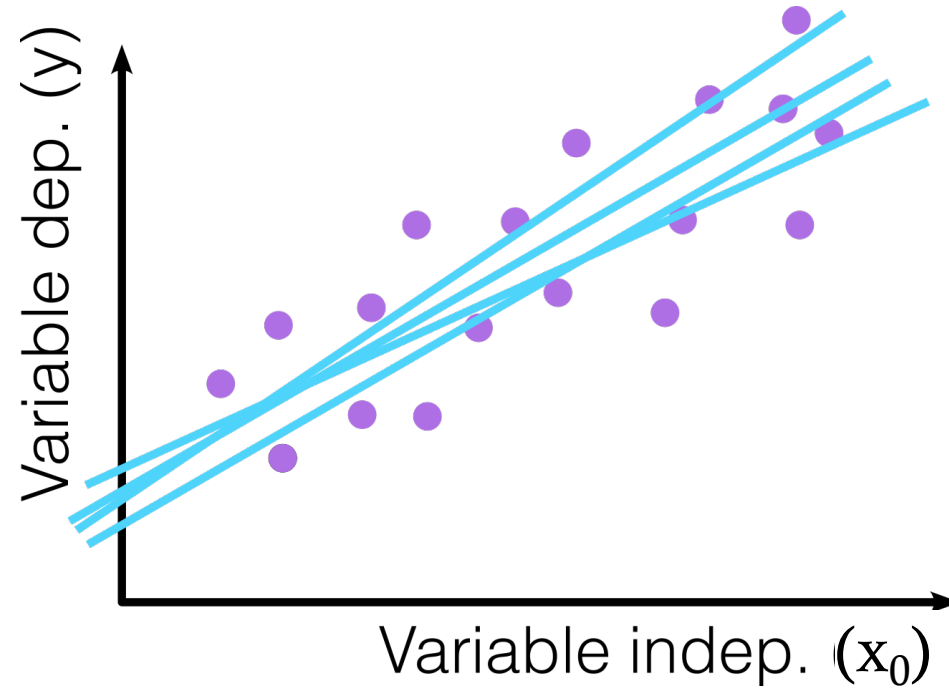
Regresión lineal es parte de una subárea de aprendizaje automático llamada **aprendizaje estadístico**, donde:

- Se enfoca en el estudio de modelos estadísticos y métodos para analizar y entender los datos.
- Utiliza herramientas y técnicas estadísticas clásicas para hacer inferencias y tomar decisiones basadas en los datos.
- El enfoque principal puede ser la estimación de parámetros, la predicción o la inferencia sobre la relación entre variables.

---

# REGRESIÓN LINEAL

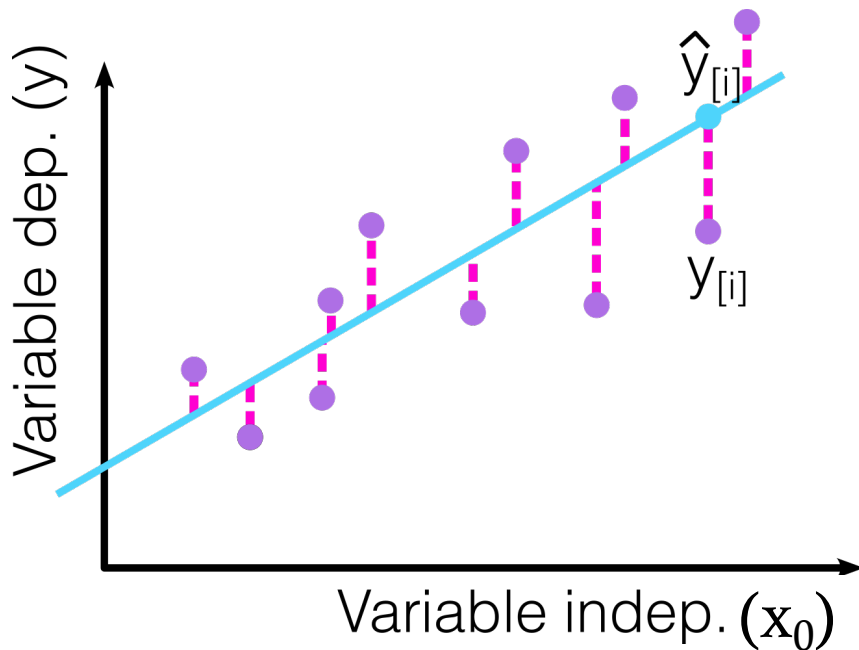
¿Ahora cuál recta?



---

# REGRESIÓN LINEAL

Para encontrarla, medimos la distancia entre la recta y cada punto, que llamamos **residuos**.

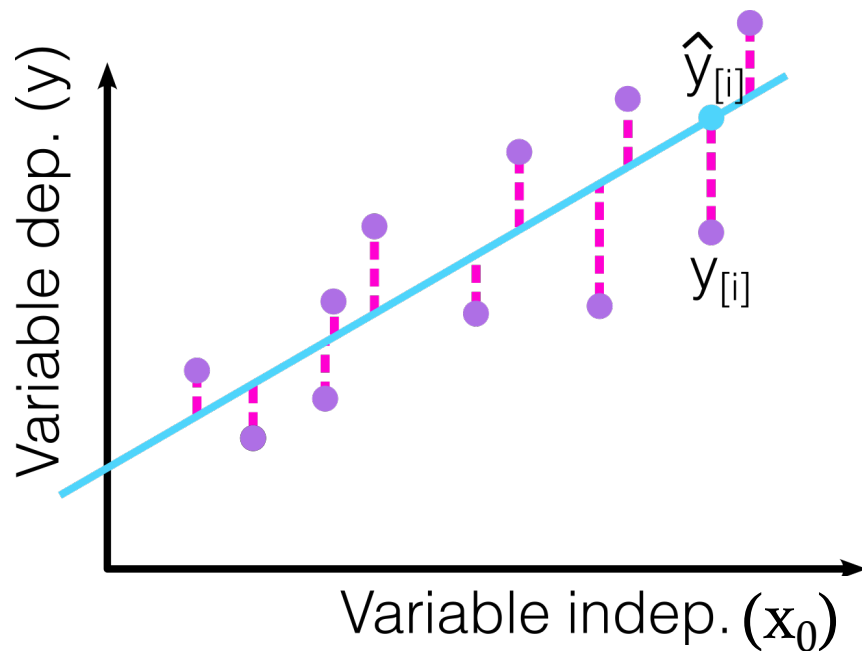


$$e_{[i]} = y_{[i]} - \hat{y}_{[i]}$$

$$y_{[i]} = b + w_0 x_{0[i]} + e_{[i]}$$

# REGRESIÓN LINEAL

Buscamos minimizar el valor de los residuos. Para lograr esto, lo hacemos minimizando la suma de los cuadrados de los **residuos**.



$$S_R = \sum_{i=0}^{N-1} (e_{[i]})^2 = \sum_{i=0}^{N-1} (y_{[i]} - b - w_0 x_{0[i]})^2$$

$$\min(S_R) = \min\left(\sum_{i=0}^{N-1} (e_{[i]})^2\right)$$

Para minimizar, solo podemos tocar los coeficientes. Lo que hacemos es ir por el **gradiente**.

$$\frac{\partial S_R}{\partial b} = 0 \quad \frac{\partial S_R}{\partial w_0} = 0$$

---

# REGRESIÓN LINEAL

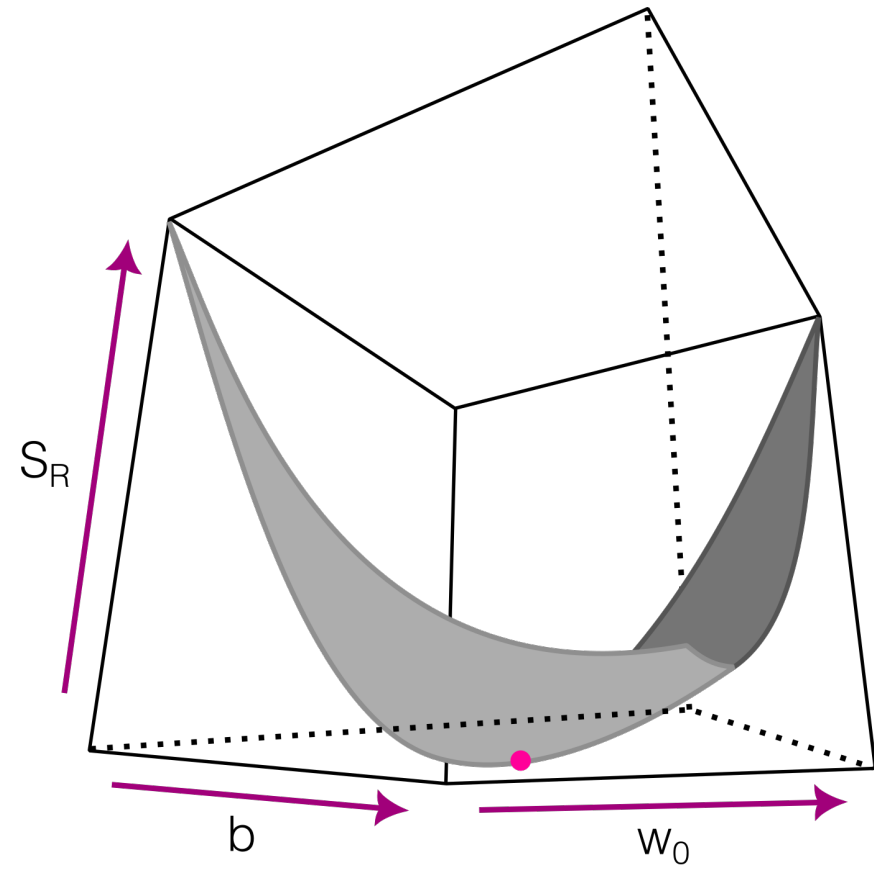
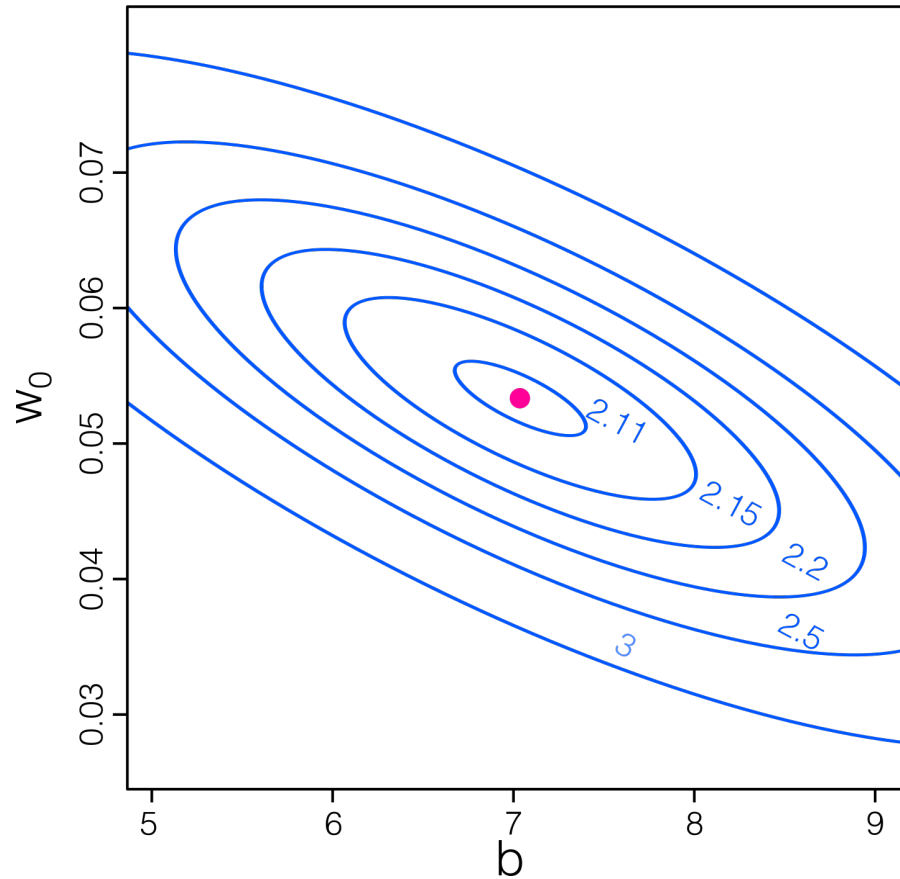
$S_R$  en regresión lineal es siempre convexa, es decir que siempre tiene un solo mínimo. En su forma tradicional,

$$\frac{\partial S_R}{\partial b} = 0 \qquad \frac{\partial S_R}{\partial w_0} = 0$$

Si planteamos las derivadas, obtenemos un sistema de ecuación, llamado ecuaciones normales. Si se resuelve este sistema se encuentra la solución.

El problema es que cuando tenemos muchos datos, resolver el sistema es muy difícil, ¡en esos casos podemos usar **gradiente descendiente**!

# REGRESIÓN LINEAL



---

# REGRESIÓN LINEAL

En todo este proceso, hemos obviado una suposición importante con respecto a los residuos, por el que aplicar mínimos cuadrados funciona,...

$$y = b + w_0x_0 + \dots + w_dx_d + e \quad \text{donde } e \sim N(0, \sigma^2)$$

Para simplificar, pasemos a notación matricial

$$y = b + \mathbf{W}^T \mathbf{X} + e \quad \text{donde } e \sim N(0, \sigma^2)$$

- $\mathbf{W} = (w_0, w_1, \dots, w_d)$
- $\mathbf{X} = (x_0, x_1, \dots, x_d)$



---

# REGRESIÓN LINEAL

Dado que el residuo tiene una distribución normal, la verosimilitud de ver un  $\mathbf{y}$  particular dado un  $\mathbf{X}$  particular:

$$P(y | \mathbf{X}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2} (y - \mathbf{W}^T \mathbf{X} - b)^2\right)$$

Y podemos encontrar los parámetros  $\mathbf{W}$  y  $b$  usando el principio de máxima verosimilitud, buscándolos aquel que maximiza la verosimilitud para todo el dataset:

$$P(\mathbf{y} | \mathbf{X}) = \prod_{i=0}^{N-1} p(y_{[i]} | \mathbf{X}_{[i]})$$

Esto se cumple porque todos los pares  $(\mathbf{X}_{[i]}, y_{[i]})$  se asumen que **independientes e idénticamente distribuidas**.

---

# REGRESIÓN LINEAL

Estimadores elegidos de acuerdo a este principio, se llaman estimadores de máxima verosimilitud. Maximizar una función de productos es realmente difícil, sobre todo si es compuesta de exponenciales. Esto lo podemos simplificar si minimizamos el logaritmo de la verosimilitud:

$$-\ln P(y | \mathbf{X}) = \sum_{i=0}^{N-1} \left( \frac{1}{2} \log(2\pi\sigma^2) + \frac{1}{2\sigma^2} (y_{[i]} - \mathbf{w}^T \mathbf{x}_{[i]} - b)^2 \right)$$

Si asumimos que todos los residuos **tienen la misma varianza**  $\sigma$ , se puede ignorar el primer término. El segundo término es igual al de mínimos cuadrados multiplicado por una constante.

*Por lo que minimizar por cuadrados mínimos es equivalente a usar un estimador de máxima verosimilitud bajo la suposición que el residuo proviene de ruido gaussiano aditivo.*

---

# REGRESIÓN LINEAL

## Ajuste

¿Como hacemos para medir que tan bien se ajusta una regresión a nuestros datos?

Si medimos la varianza de la variable dependiente de los datos:

$$S_T = \sum_{i=0}^{N-1} (y_{[i]} - \bar{y})^2$$

Esta varianza la podemos separar en dos partes, una parte que es **dada por el modelo** y **una que no**:

$$S_T = S_R + S_E$$
$$S_T = \sum_{i=0}^{N-1} (\hat{y}_{[i]} - \bar{y})^2 + \sum_{i=0}^{N-1} (y_{[i]} - \hat{y}_{[i]})^2$$

---

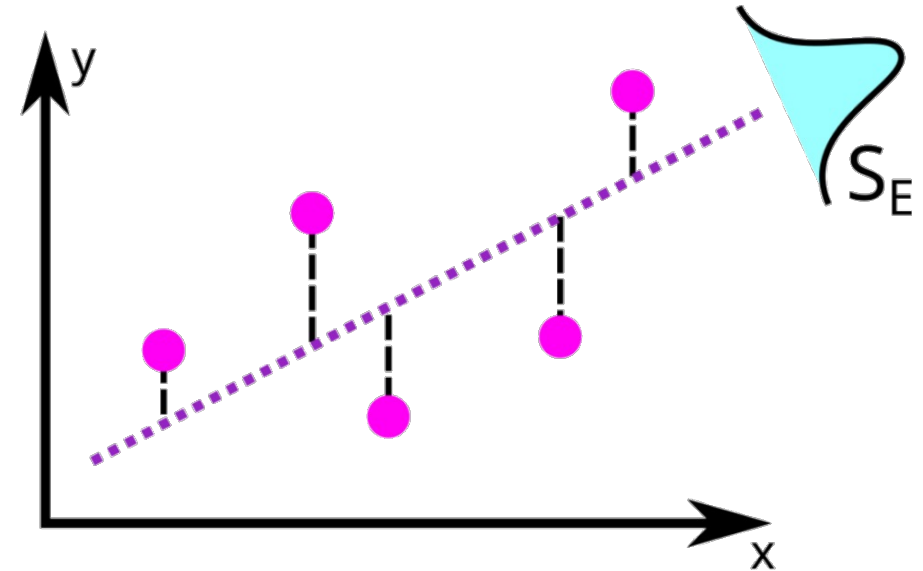
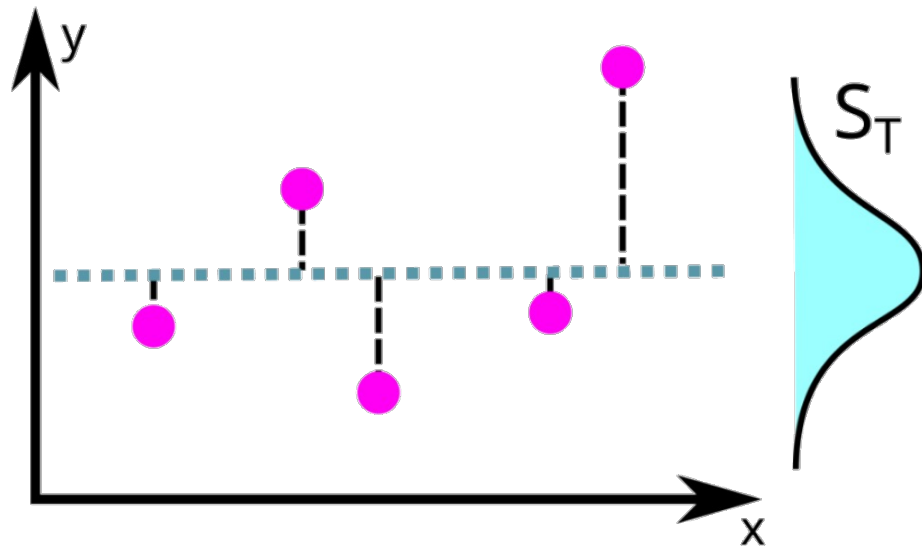
Parte que explica el modelo

Parte que no (residuos)

---

# REGRESIÓN LINEAL

## Ajuste



---

# REGRESIÓN LINEAL

## Ajuste

Como métricas, podemos usar:

- El cálculo del desvío estándar del modelo:

$$s_E = \sqrt{\frac{S_E}{N - d - 1}}$$

Donde  $d$  es la cantidad de features.

- Coeficiente de Pearson (cuando más cerca de 1 mejor ajuste, es decir residuos más chicos):

$$R^2 = \frac{S_R}{S_T} = 1 - \frac{S_E}{S_T}$$

---

# REGRESIÓN LINEAL

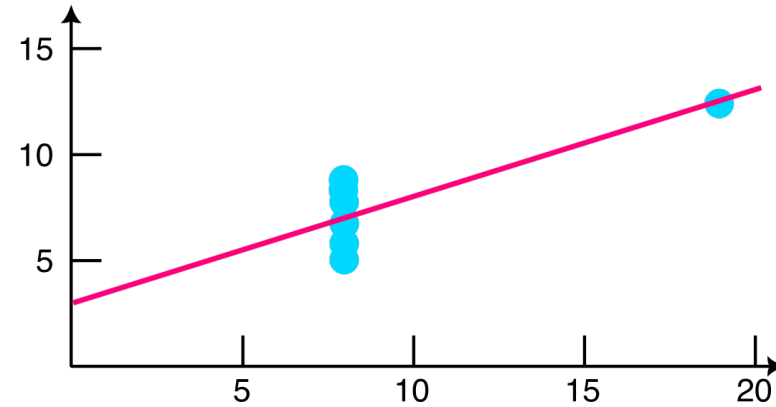
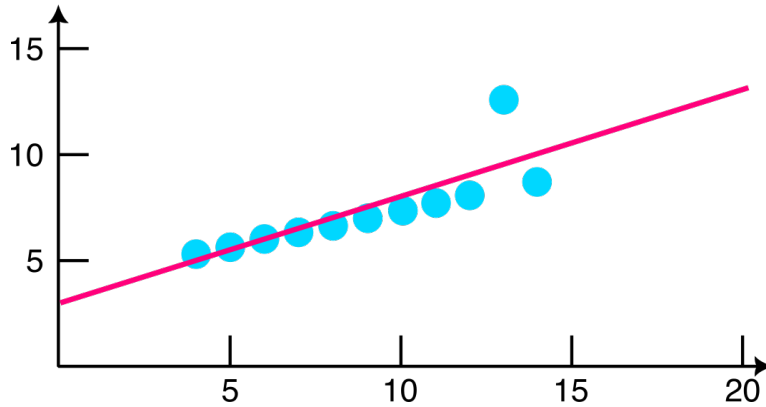
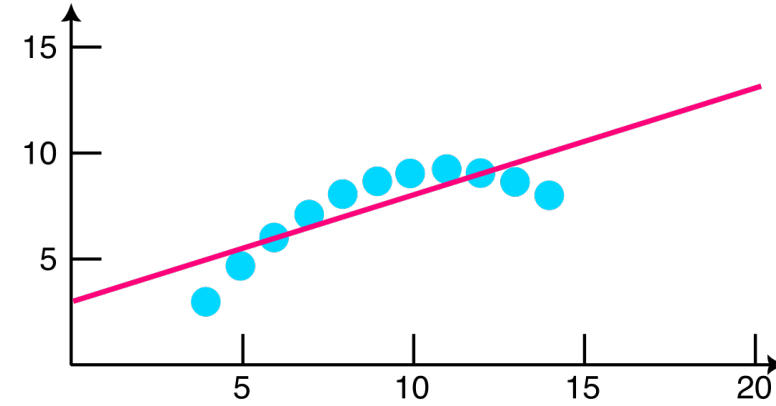
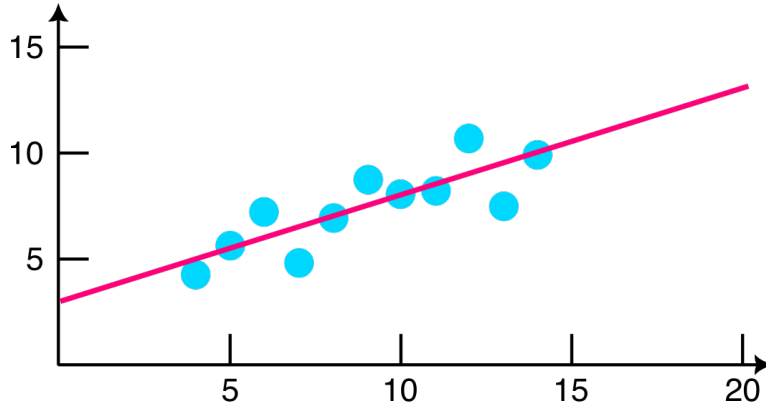
## Ajuste

El valor del coeficiente de Pearson no es una métrica muy buena (a pesar de su popularidad). Todo dependerá de qué hace que genera el valor de residuo:

- Si la relación entre la variable independiente y dependiente es realmente lineal, y se cumplen las suposiciones, el residuo es  $e \sim N(0, \sigma^2)$ . Por lo que depende de cómo se tomaron los datos y la varianza de la distribución normal. *Por ejemplo, en registro de temperatura, si el termómetro es de mala calidad, esperamos tener una mayor variación, por lo que el coeficiente podría ser bajo, pero sin ser culpa del modelo.*
- Si la relación no es lineal, el valor del coeficiente de Pearson nos va a indicar que el modelo es malo.

# REGRESIÓN LINEAL

## Ajuste



---

# REGRESIÓN LINEAL

## Suposiciones

Las suposiciones que usamos para poder aplicar regresión lineal son:

- **Relación lineal:** Lógicamente, y esto muchas veces al aplicar el modelo buscamos validar.
- **Features independientes:** Los features de entrada de la regresión deben ser independientes entre sí.
- **Homocedasticidad:** Es decir, el valor de  $S_E$  se mantiene igual en toda parte de la recta
- **Errores independientes:** Los errores entre si no están correlacionados.





---

# MÉTRICAS DE EVALUACIÓN

---

# MÉTRICAS DE EVALUACIÓN

Como vimos la clase anterior, cuando armamos el modelo, al dataset lo separamos en una parte usada para entrenar el modelo y la parte de evaluación. El conjunto de datos de evaluación se utiliza para evaluar qué tan bien se entrenó el algoritmo con el conjunto de datos de entrenamiento.

¿Pero cómo evaluamos?

- **El coeficiente de Pearson ( $R^2$ ).** Aunque no es el mejor caso.

Podemos usar métricas más generales, métricas que midan error de variables numéricas que se pueda aplicar también a otros tipos de casos, como por ejemplo forecasting en series de tiempo.

---

# MÉTRICAS DE EVALUACIÓN

## Error absoluto medio (MAE)

El **error absoluto medio (MAE)** es el cálculo del valor absoluto del residuo para cada punto de datos, para que los residuos negativos y positivos no se cancelen. Luego tomamos el promedio de todos estos residuos.

$$MAE = \frac{1}{N} \sum_{i=0}^{N-1} |y[i] - \hat{y}[i]|$$

Debido a que utilizamos el valor absoluto del residuo, MAE no indica si el modelo sobreestima o subestima.

---

# MÉTRICAS DE EVALUACIÓN

## Error cuadrático medio (MSE)

El **error cuadrático medio (MSE)** es similar al **MAE**, pero ahora calculamos el cuadrado de los residuos. Esto es similar a lo que se usamos para encontrar los coeficientes.

$$MSE = \frac{1}{N} \sum_{i=0}^{N-1} (y_{[i]} - \hat{y}_{[i]})^2$$

**MSE** siempre es mayor a **MAE**. Un detalle importante son aquellos residuos grandes (**outliers**), en esta métrica aporta más que en **MAE**. En **MAE** el aporte es proporcional al valor del residuo, pero aquí es cuadráticamente más grande.

---

# MÉTRICAS DE EVALUACIÓN

Raíz cuadrada del error cuadrático medio (RMSE)

Si al **MSE** le calculamos la raíz, tendremos una métrica llamada **RMSE** que tiene la misma unidad de la salida original, donde el **MSE** no. El **RMSE** es análogo al desvío estándar.

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=0}^{N-1} (y_{[i]} - \hat{y}_{[i]})^2}$$

---

# MÉTRICAS DE EVALUACIÓN

## Outliers

Valores atípicos es una constante de discusión. *¿Se incluyen o no?*

La respuesta dependerá del problema en particular, de los datos disponibles y las consecuencias que hay si se consideran o no.

Si quiero tenerlo en cuenta a la hora de comparar modelos, me va a convenir usar MSE, en cambio sí quiero reducir su importancia, puedo usar MAE.

Ambas son métricas de error viables, pero describen diferentes matices sobre los errores de predicción.

---

# MÉTRICAS DE EVALUACIÓN

## Error absoluto porcentual medio (MAPE)

El **error absoluto porcentual medio (MAPE)** es el cálculo del error **MAE**, pero escalado al verdadero valor, por lo que el resultado es porcentual

$$MAPE = \frac{100\%}{N} \sum_{i=0}^{N-1} \left| \frac{y[i] - \hat{y}[i]}{y[i]} \right|$$

No es una métrica buena porque es susceptible a errores numéricos. No puede calcularse cuando  $y_{[i]}$  vale cero. Y además tiene sesgo para cuando la predicción subestima:

$$n = 1 \quad \hat{y} = 10 \quad y = 20 \\ MAPE = 50\%$$

$$n = 1 \quad \hat{y} = 20 \quad y = 10 \\ MAPE = 100\%$$

---

# MÉTRICAS DE EVALUACIÓN

Error porcentual medio (MPE)

El **error porcentual medio (MPE)** es el cálculo del error **MAPE**, pero ahora no calculamos el valor absoluto

$$MPE = \frac{100\%}{N} \sum_{i=0}^{N-1} \frac{y[i] - \hat{y}[i]}{y[i]}$$

Aunque la falta de valor absoluto puede ser problemático ya que puede cancelar términos, nos permite saber si:

- El modelo subestima
- El modelo sobreestima





---

# TRATAMIENTO DE VARIABLES

---

# TRATAMIENTO DE VARIABLES

## Normalización o estandarización

En la regresión lineal, tenemos la multiplicación de coeficientes por nuestras entradas:

$$\hat{y} = b + w_0x_0 + w_1x_1$$

Los coeficientes nos dan un **valor de importancia de las entradas**. Pero esto si todas las entradas están en la misma escala.

Si la variable  $x_0$  está en rango de  $[1000, 3000]$  y  $x_1$  en  $[-1, 1]$ , los valores de  $w_0$  y  $w_1$  van a ser de diferentes escalas, y por consiguiente no comparables.

Además, aunque la regresión lineal no presenta problemas de escalas, valores muy diferentes nos pueden introducir **errores numéricos**. Otro tipo de regresiones o clasificadores si o si necesitan escalas, por lo que debemos normalizar o estandarizar.

---

# TRATAMIENTO DE VARIABLES

## Normalización o estandarización

Una forma de **normalizar** es hacer que los valores estén entre 0 y 1, tomando el máximo y el mínimo.

$$\tilde{x} = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Esta fórmula se usa cuando la distribución de datos no es normal. Pero cuando nuestros datos tienen una distribución normal, como la suposición en regresión lineal, se aplica **estandarización**:

$$\tilde{x} = \frac{x - \text{mean}(x)}{\text{std}(x)}$$

Donde ahora hacemos que la distribución tenga medio cero y desvío estándar uno.

Con estos escalados, ahora los parámetros tendrán sentido entre sí.

Mas detalles los verán en: *Análisis de datos*

---

# TRATAMIENTO DE VARIABLES

## Variables Dummies

Como vimos, regresión utiliza variables numéricas para predecir un valor. ¿Como podemos hacer para usar variables categóricas?

Para poder usarlos, debemos transformarlos en numéricas mediante alguna codificación.

Cuando tenemos variables categóricas ordinales, podemos asociar un número.

Por ejemplo, si tenemos que usar casos como : “*me gusta mucho*”, “*me gusta poco*”, “*neutral*”, “*no me gusta poco*”...

Se puede usar números enteros, teniendo en cuenta el orden para darle importancia. Estará en la creatividad de quien lo hace para determinar si las distancias son equidistantes o no.

---

# TRATAMIENTO DE VARIABLES

## Variables Dummies

Ahora si tenemos casos nominales no podemos asociar números, porque al hacerlo, establecemos un orden.

Para este tipo de variable existe **one-hot encoding**.

Al usar esta codificación, creamos nuevos atributos de acuerdo con la cantidad de clases presentes en la variable categórica, es decir, si hay **n** número de categorías, se crearán **n** nuevos atributos. Estos atributos creados se denominan *variables dummies*.

---

# TRATAMIENTO DE VARIABLES

## Variables Dummies

Peso	Altura	País
80	180	Argentina
83	177	Chile
75	169	Chile
68	155	Argentina

---

# TRATAMIENTO DE VARIABLES

## Variables Dummies

Peso	Altura	País	arg	chile
80	180	Argentina	1	0
83	177	Chile	0	1
75	169	Chile	0	1
68	155	Argentina	1	0

---

# TRATAMIENTO DE VARIABLES

## Variables Dummies

Peso	Altura	arg	chile
80	180	1	0
83	177	0	1
75	169	0	1
68	155	1	0



---

# TRATAMIENTO DE VARIABLES

## Variables Dummies

Como vimos, **one-hot encoding** nos genera un nuevo atributo por categoría, pero esto nos genera *una trampa*

Si vemos el ejemplo, las dos variables que estamos usando están 100% correlacionadas entre sí:

$$\begin{aligned} \hat{y} &= b + w_0x_{\text{peso}} + w_1x_{\text{altura}} + w_2x_{\text{arg}} + w_3x_{\text{chile}} \\ x_{\text{chile}} &= 1 - x_{\text{arg}} \\ \hat{y} &= b + w_0x_{\text{peso}} + w_1x_{\text{altura}} + w_2x_{\text{arg}} + w_3(1 - x_{\text{arg}}) \\ \hat{y} &= (b + w_3) + w_0x_{\text{peso}} + w_1x_{\text{altura}} + (w_2 - w_3)x_{\text{arg}} \end{aligned}$$

Para solucionar esto es quitar siempre quitar una columna para romper la trampa.

---

# TRATAMIENTO DE VARIABLES

## Variables Dummies

Peso	Altura	arg
80	180	1
83	177	0
75	169	0
68	155	1

---

# REGRESIÓN LINEAL

*Vamos a practicar un poco...*



---

# CONSTRUCCIÓN DE UN MODELO

---

# CONSTRUCCIÓN DE UN MODELO

¿Cuándo construimos un modelo de regresión múltiple, como hacemos para elegir los features que formarán parte del modelo?

Ver la correlación entre variables es un primer paso, pero surge la pregunta, si dos variables estas correlacionadas, ¿cuál de las dos descarto?

*Por lo que hay diferentes métodos de construir un modelo.*

---

# CONSTRUCCIÓN DE UN MODELO

Podemos mencionar 4 formas:

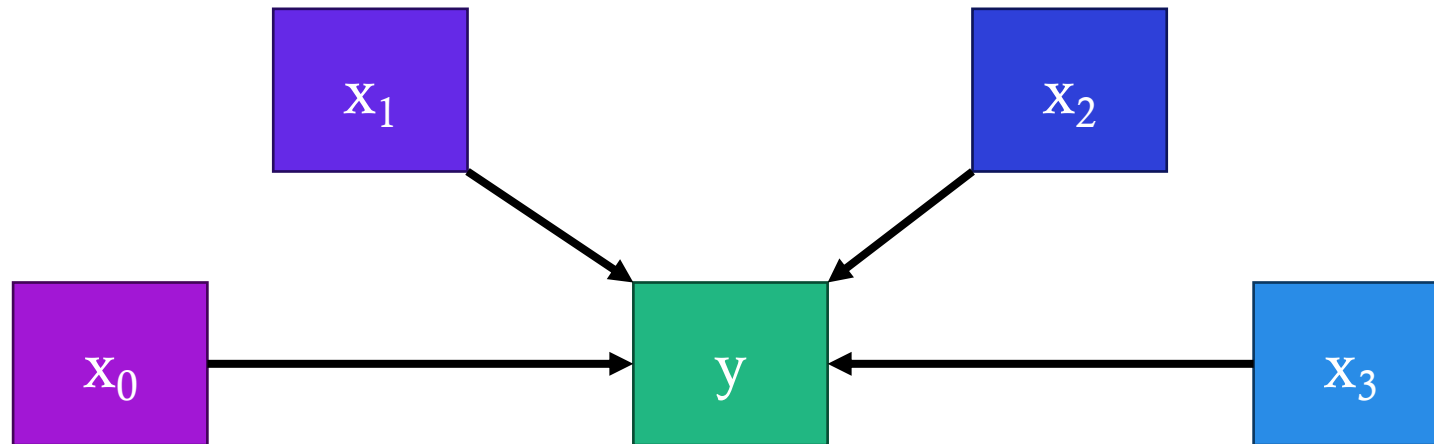
- Exhaustivo
- Eliminación hacia atrás
- Selección hacia adelante
- Eliminación bidireccional

---

# CONSTRUCCIÓN DE UN MODELO

## Exhaustivo (all-in)

Este es el caso más sencillo, usamos todas las variables. En qué casos conviene usar esta forma es cuando tenemos conocimiento a priori, o una necesidad específica.



---

# CONSTRUCCIÓN DE UN MODELO

## Eliminación hacia atrás

- Se arranca con un modelo completo con todas las variables.
- Se va eliminado de forma greedy, variables de entrada que menos “aportan” el modelo de una por vez.
- Se continua hasta que eliminar variables no mejore significativamente el modelo.
- La forma que se puede realizar es con alguna métrica que nos mida la información que aporta cada variable.



---

# CONSTRUCCIÓN DE UN MODELO

## Selección hacia adelante

- Comienza con un modelo *vacío* con solo la ordenada al origen.
- Luego se agrega las variables que más aporta al modelo (usando el criterio de ajuste) de una por vez.
- Se termina una vez que agregar más variables no genera mejor aporte.

---

# CONSTRUCCIÓN DE UN MODELO

## Eliminación bidireccional

- Es en esencia la selección hacia adelante, pero dando la posibilidad de quitar variables en cada iteración cuando se observa correlación entre variables.

---

# CONSTRUCCIÓN DE UN MODELO

Como hacemos para saber el aporte de cada atributo

- **Bondad de ajuste:** Se realiza un test de hipótesis si el coeficiente de una entrada en particular es cero. Luego se evalúa el valor de p. Un valor p bajo ( $< 0,05$ ) indica que se puede rechazar la hipótesis nula, indicando que cambios en esta variable es probable que genere cambios en la respuesta.
- **Coeficiente de Pearson ajustado:** El cual es el  $R^2$  pero penalizando la complejidad del modelo:

$$R_{adj}^2 = 1 - (1 - R^2) \frac{N - 1}{N - d - 1}$$

Donde N es número de observaciones y d es el número de atributos.

---

# CONSTRUCCIÓN DE UN MODELO

Como hacemos para saber el aporte de cada atributo

- **Criterio de Información de Aikake (AIC):** AIC maneja un equilibrio entre la bondad de ajuste del modelo y la complejidad del modelo. En otras palabras, AIC aborda tanto el riesgo de sobre-ajuste como el riesgo sub-ajuste.

$$AIC = 2d - 2\ln(\hat{L})$$

Donde  $-\ln(\hat{L})$  es el logaritmo de la estimación de máxima similitud. Este valor se obtiene como:

$$\begin{aligned} -\ln(\hat{L}) &= \sum_{i=0}^{N-1} \left( \frac{1}{2} \log(2\pi\sigma^2) + \frac{1}{2\sigma^2} (y_{[i]} - \mathbf{w}^T \mathbf{X}_{[i]} - b)^2 \right) \\ -\ln(\hat{L}) &= \frac{N}{2} \ln(2\pi) + \frac{N}{2} \ln(\sigma^2) + \frac{1}{2\sigma^2} \sum_{i=0}^{N-1} (y_{[i]} - \mathbf{w}^T \mathbf{X}_{[i]} - b)^2 \quad \sigma = \sqrt{\frac{S_E}{N}} \quad \text{con } N \gg d \\ -\ln(\hat{L}) &= \frac{N}{2} \ln(2\pi) + \frac{N}{2} \ln\left(\frac{S_E}{N}\right) + \frac{N}{2} \quad S_E = \sum_{i=0}^{N-1} (y_{[i]} - \mathbf{w}^T \mathbf{X}_{[i]} - b)^2 \end{aligned}$$

El valor de AIC cuando es más bajo, es mejor.

---

# CONSTRUCCIÓN DE UN MODELO

Como hacemos para saber el aporte de cada atributo

- **Criterio de información bayesiano (BIC):** Se basa en el principio de la navaja de Occam, que establece que es preferible el modelo más simple que explique los datos. A diferencia del AIC, BIC penaliza más el modelo por su complejidad

$$BIC = d \ln(N) - 2\ln(\hat{L})$$

El valor de BIC cuando es más bajo es mejor.



---

# REGRESIÓN LASSO Y RIDGE

---

# REGRESIÓN DE RIDGE Y LASSO

Como hacemos para saber el aporte de cada atributo

Cuando se trata de entrenar modelos, nos podemos encontrar como problemas de sobreajuste, podemos implementar algún método de regularización.

Con estos métodos de regularización podemos ajustar un modelo que contenga todos los atributos utilizando una técnica que restrinja o regularice las estimaciones de los coeficientes o, de manera equivalente, que reduzca las estimaciones de los coeficientes hacia cero.

Puede que no sea inmediatamente obvio por qué tal restricción debería mejorar el ajuste, pero resulta que **reducir las estimaciones de los coeficientes** puede **reducir significativamente su varianza**.

Las dos técnicas más conocidas para reducir los coeficientes de regresión a cero son la regresión de **Ridge** y la de **Lasso**.

---

# REGRESIÓN DE RIDGE

En la regresión lineal, vimos que se buscaba los coeficientes que minimizaban la suma de los residuos al cuadrado. La regresión de Ridge es muy similar, pero excepto que los coeficientes se estiman minimizando una cantidad ligeramente diferente:

$$\sum_{i=0}^{N-1} (y_{[i]} - b - \mathbf{w}^T \mathbf{X}_{[i]})^2 + \alpha \sum_{j=0}^{d-1} w_j^2$$

Donde  $\alpha$  es un hiper-parámetro de ajuste.



---

# REGRESIÓN DE RIDGE

En esta regresión, se busca los coeficientes que minimizan  $S_E$ . Sin embargo, el segundo termino:

$$\alpha \sum_{j=0}^{d-1} w_j^2$$

Llamado el termino de penalización por encogimiento. Este es pequeño cuando los coeficientes están cerca de cero.

$\alpha$  funciona de control al impacto relativo de ambos términos. Cuando  $\alpha = 0$ , es una regresión normal. En cambio, si  $\alpha$  crece, el impacto de termino de penalización crece, y los coeficientes se acercan a cero.

La **regresión de Ridge** genera un set de coeficientes por cada valor de  $\alpha$ . Seleccionar un valor de  $\alpha$  es difícil, y se deben usar métodos de validación cruzada.

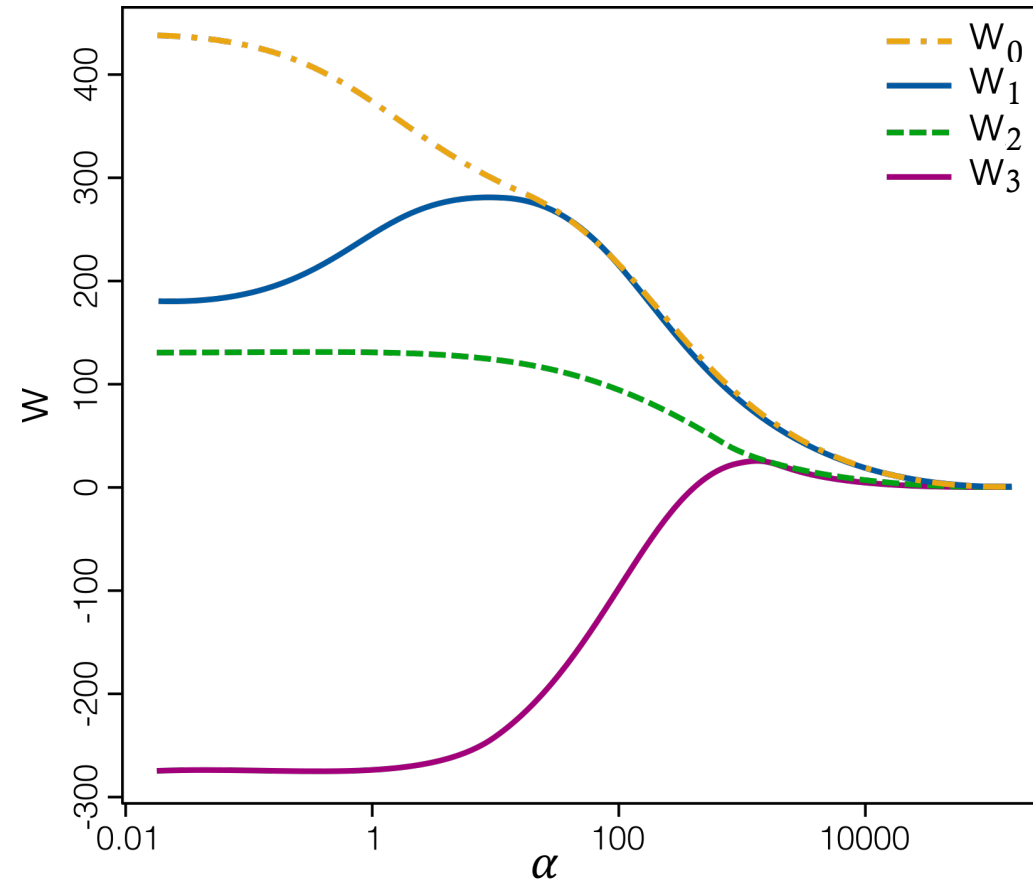
---

# REGRESIÓN DE RIDGE

Notese que la penalización no toca la ordenada al origen  $b$ . Si  $\alpha$  es  $\infty$ , todos los coeficientes son cero y  $b$  nos queda:

$$b = \bar{y} = \frac{1}{N} \sum_{i=0}^{N-1} y[i]$$

# REGRESIÓN DE RIDGE

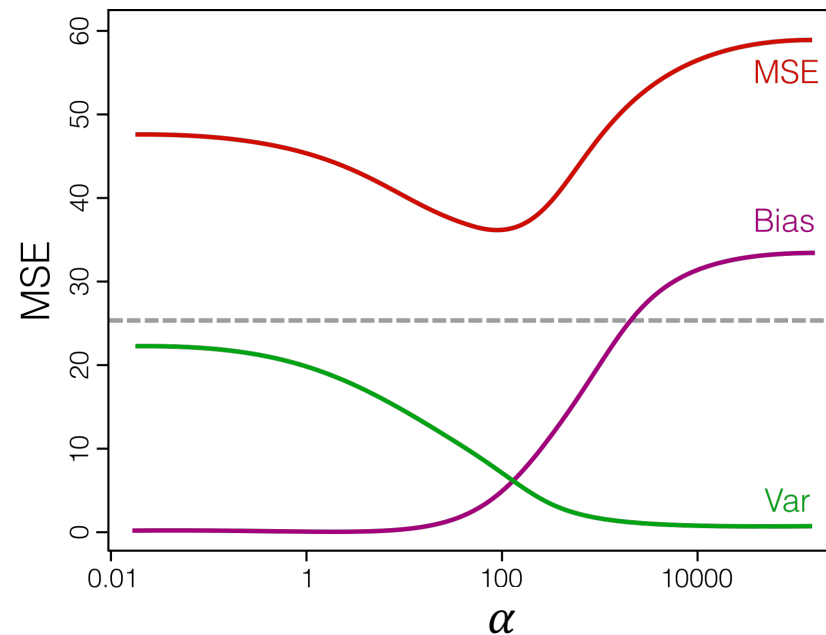


---

# REGRESIÓN DE RIDGE

*¿Para qué nos sirve?*

La ventaja de la regresión de Ridge sobre la regresión lineal de mínimos cuadrados tiene su origen en el equilibrio entre sesgo y varianza. A medida que  $\alpha$  aumenta, la **flexibilidad del ajuste disminuye**, lo que lleva a una **menor varianza**, pero a un **mayor sesgo**.

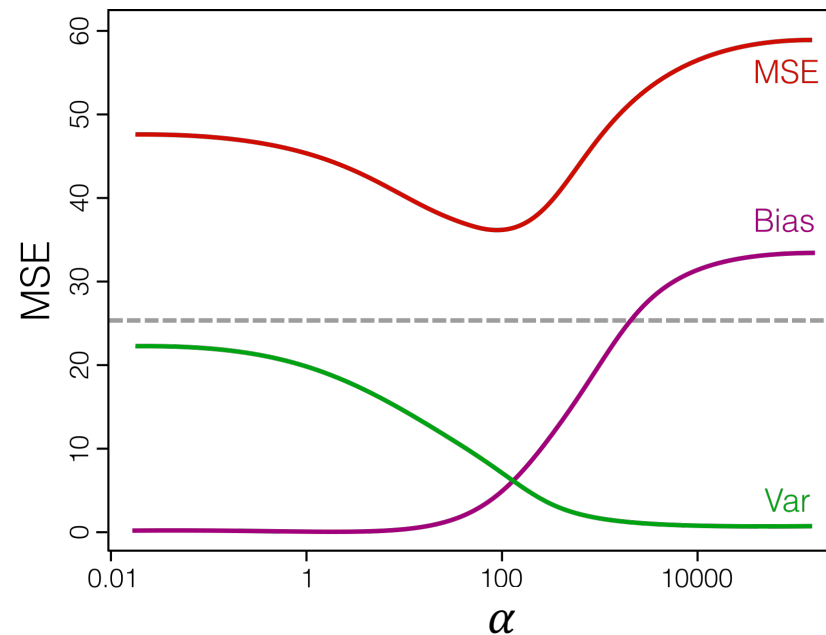


---

# REGRESIÓN DE RIDGE

*¿Para qué nos sirve?*

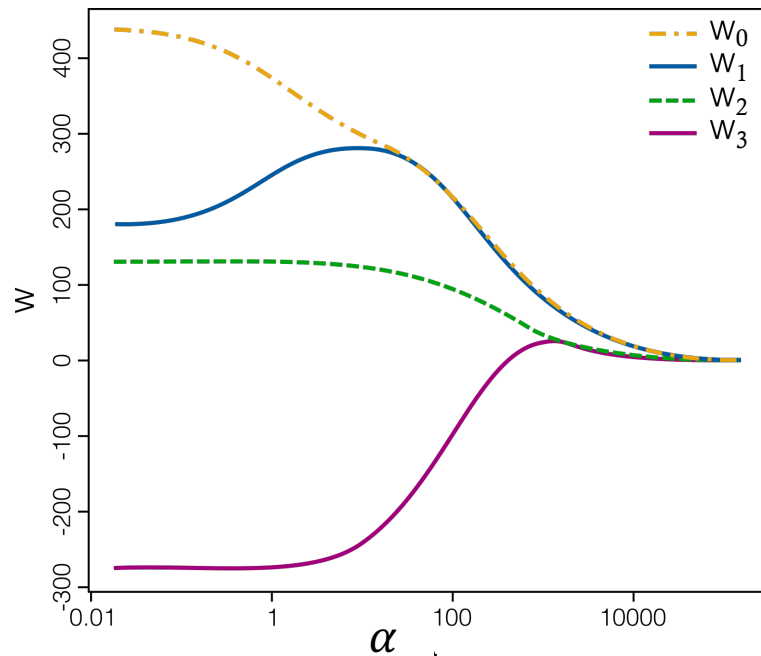
En general, cuando la verdadera relación es lineal, la regresión lineal tiene mucha varianza. Esto principalmente ocurre cuando el **número de observaciones es cercano al número de coeficientes**. En estos casos, la regresión de Ridge funciona mejor.



---

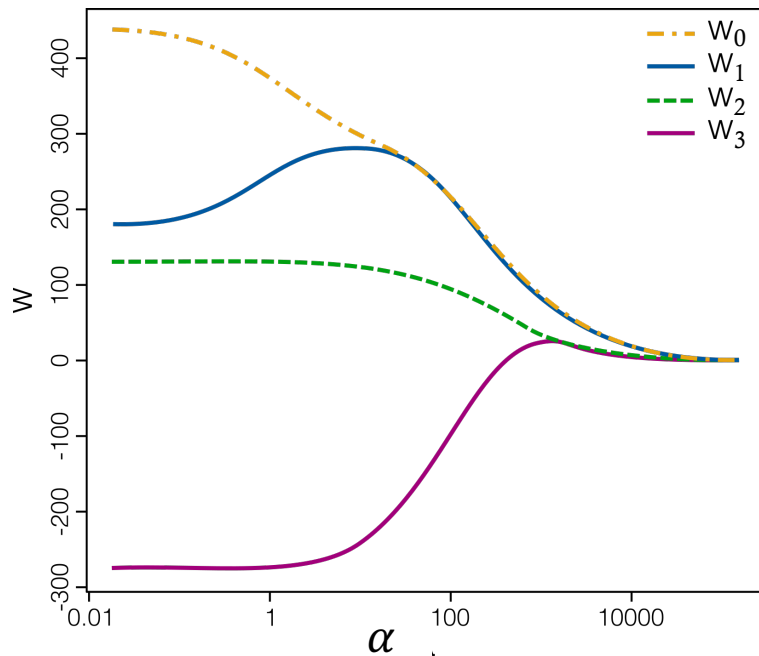
# REGRESIÓN DE LASSO

La regresión de Ridge, a priori, nos parece interesante para hacer una selección de modelo, ya que, jugando con  $\alpha$  podemos ver si algún coeficiente se hace cero:



# REGRESIÓN DE LASSO

La regresión de Ridge, a priori, nos parece interesante para hacer una selección de modelo, ya que, jugando con  $\alpha$  podemos ver si algún coeficiente se hace cero:



El problema es que los coeficientes se achican a cero, pero no se hacen cero, salvo que  $\alpha$  sea infinito. Por lo que **no podemos eliminar atributos**.

---

# REGRESIÓN DE LASSO

La regresión de Lasso cubre esta desventaja:

$$\sum_{i=0}^{N-1} (y_{[i]} - b - \mathbf{w}^T \mathbf{X}_{[i]})^2 + \alpha \sum_{j=0}^{d-1} |w_j|$$

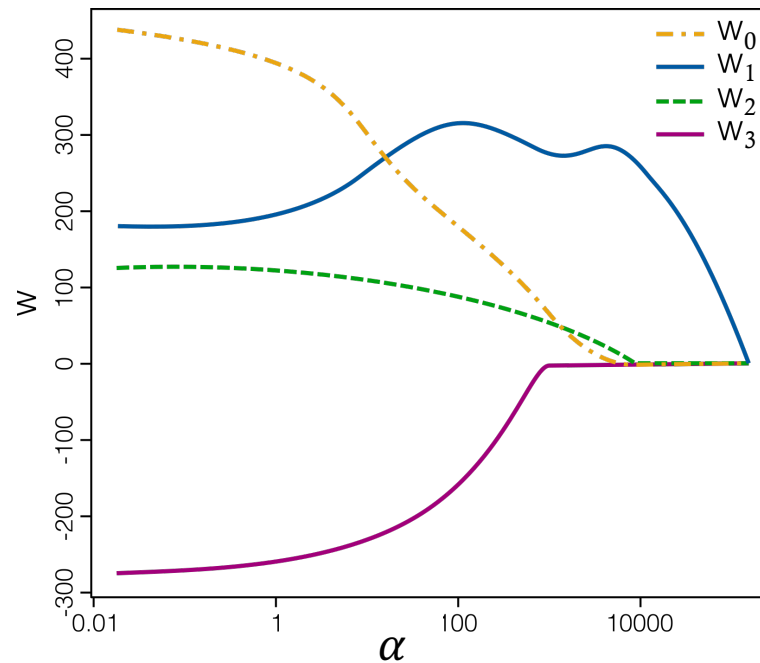
Es decir, la regresión de Lasso usa una penalización L1, mientras que Ridge usa una penalización L2.



---

# REGRESIÓN DE LASSO

Esta regresión cuando  $\alpha$  crece, algunos coeficientes se hacen exactamente cero. Por lo que Lasso realiza una selección de atributos.



---

# REGRESIÓN DE LASSO

*¿Para qué nos sirve?*

Para entender porque esto ocurre, debemos reescribir a las regresiones de otra forma equivalente:

$$\begin{aligned} \text{Regresión Lasso} \quad \underset{w}{\text{minimizar}} &= \left\{ \sum_{i=0}^{N-1} (y_{[i]} - b - \mathbf{w}^T \mathbf{X}_{[i]})^2 \right\} \text{ sujeto a } \sum_{j=0}^{d-1} |w_j| \leq s \\ \text{Regresión Ridge} \quad \underset{w}{\text{minimizar}} &= \left\{ \sum_{i=0}^{N-1} (y_{[i]} - b - \mathbf{w}^T \mathbf{X}_{[i]})^2 \right\} \text{ sujeto a } \sum_{j=0}^{d-1} w_j^2 \leq s \end{aligned}$$

---

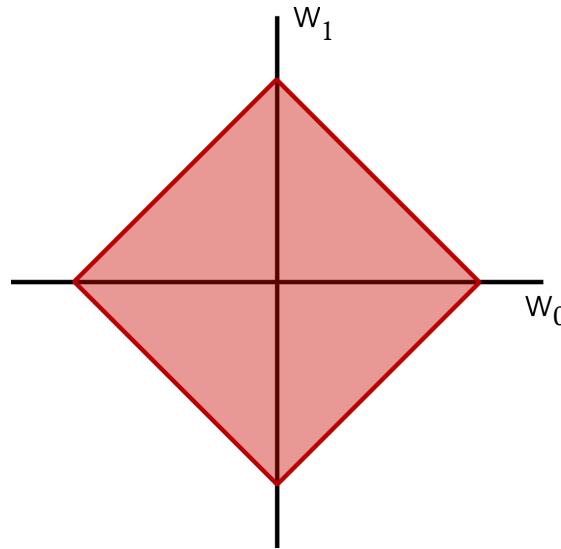
# REGRESIÓN DE LASSO

*¿Para qué nos sirve?*

Veamos el efecto de la penalización en un caso de 2 atributos ( $d=2$ ):

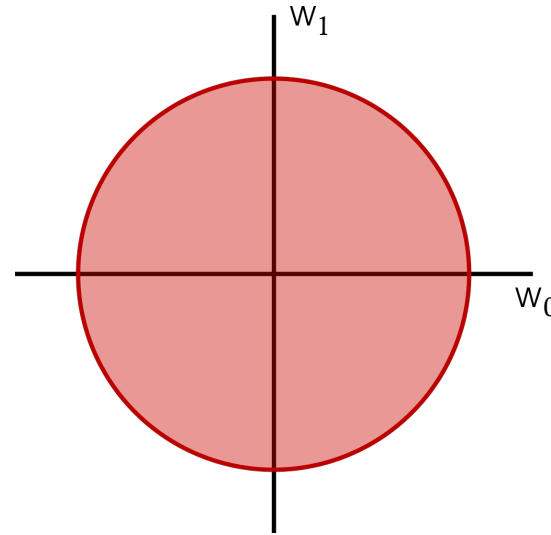
*Regresión Lasso*

$$|w_0| + |w_1| \leq s$$



*Regresión Ridge*

$$w_0^2 + w_1^2 \leq s$$

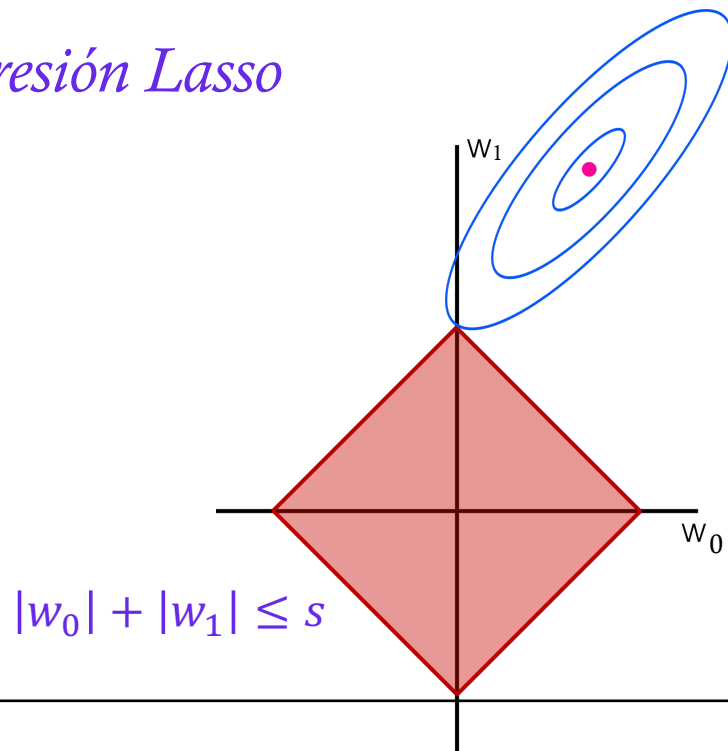


# REGRESIÓN DE LASSO

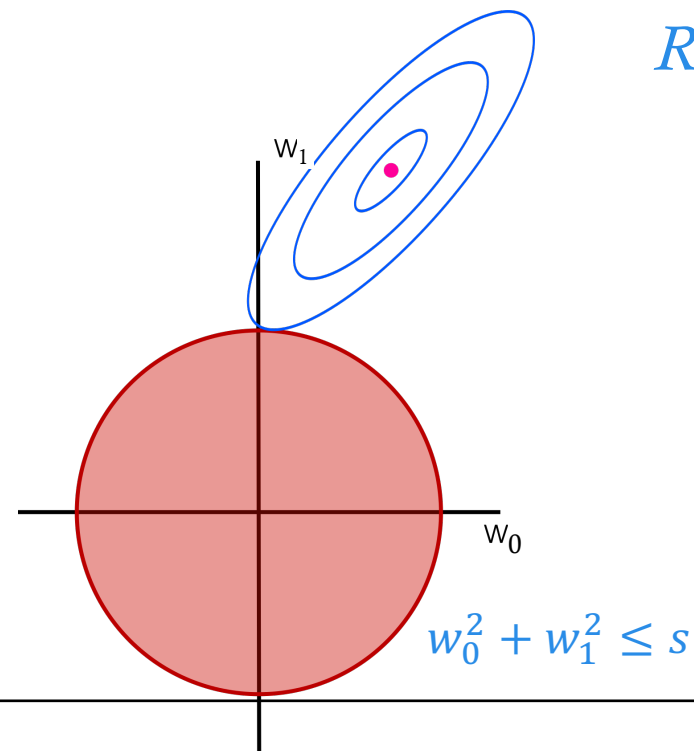
*¿Para qué nos sirve?*

Veamos el efecto de la penalización en un caso de 2 atributos ( $d=2$ ):

*Regresión Lasso*



*Regresión Ridge*



---

# REGRESIÓN DE LASSO

*Vamos a practicar un poco...*