# Regularización Paramétrica

# Conceptos básicos de las máquinas de aprendizaje

# El escenario inicial

**Un conjunto de datos**

Existe una serie de pasos estándar a seguir en la implementación de una solución de Machine Learning:

# El escenario inicial

**Un conjunto de datos**

Existe una serie de pasos estándar a seguir en la implementación de una solución de Machine Learning:

- Dividir

# El escenario inicial

**Un conjunto de datos**

Existe una serie de pasos estándar a seguir en la implementación de una solución de Machine Learning:

- Dividir
- Entrenar

# El escenario inicial

**Un conjunto de datos**

Existe una serie de pasos estándar a seguir en la implementación de una solución de Machine Learning:

- Dividir
- Entrenar
- Predecir

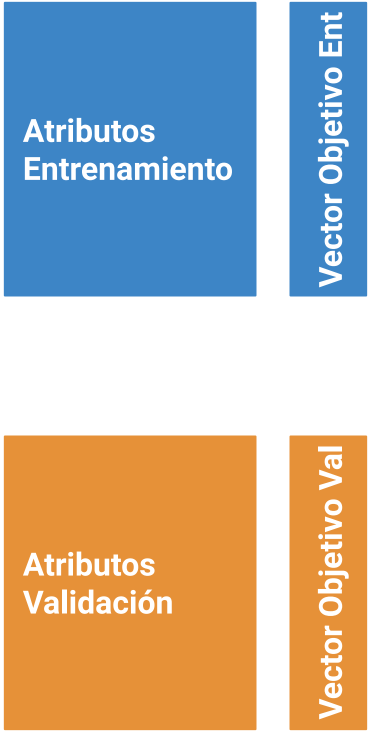
# El escenario inicial

**Un conjunto de datos**

Existe una serie de pasos estándar a seguir en la implementación de una solución de Machine Learning:

- Dividir
- Entrenar
- Predecir
- Evaluar

# La división de muestras




Dividimos para “replicar” el comportamiento de nuestro modelo en un nuevo conjunto de datos.

Por lo general dividimos en 4 objetos:

- Atributos (X) de entrenamiento
- Atributos (X) de validación
- Vector objetivo de entrenamiento
- Vector objetivo de validación

# El entrenamiento de un modelo

$$y = \beta_0 + \beta_1 x_i + \varepsilon_i$$

Atributos Entrenamiento

Vector Objetivo Ent

Atributos Validación

# El entrenamiento de un modelo

$$y = \beta_0 + \beta_1 x_i + \varepsilon_i$$

Atributos Entrenamiento

Vector Objetivo Ent

Atributos Entrenamiento

Vector Objetivo Ent

Atributos Validación

# La predicción de un modelo

$$y = \beta_0 + \beta_1 x_i + \varepsilon_i$$

Atributos Entrenamiento

Vector Objetivo Ent

Atributos Entrenamiento

Vector Objetivo Ent

$$\hat{y} = \beta_0 + \beta_1 X$$

Atributos Validación

Vector Objetivo Val

# La predicción de un modelo

$$y = \beta_0 + \beta_1 x_i + \varepsilon_i$$

Atributos Entrenamiento

Vector Objetivo Ent

Atributos Entrenamiento

Vector Objetivo Ent

$$\hat{y} = \beta_0 + \beta_1 X$$

Atributos Validación

Vector Objetivo Val

Atributos Validación

# La predicción de un modelo

$$y = \beta_0 + \beta_1 x_i + \varepsilon_i$$


$$\hat{y} = \beta_0 + \beta_1 X$$

# La evaluación del modelo

$$y = \beta_0 + \beta_1 x_i + \varepsilon_i$$

Atributos Entrenamiento

Vector Objetivo Ent

Atributos Entrenamiento

Vector Objetivo Ent

$$\hat{y} = \beta_0 + \beta_1 X$$

Atributos Validación

Vector Objetivo Val

Atributos Validación

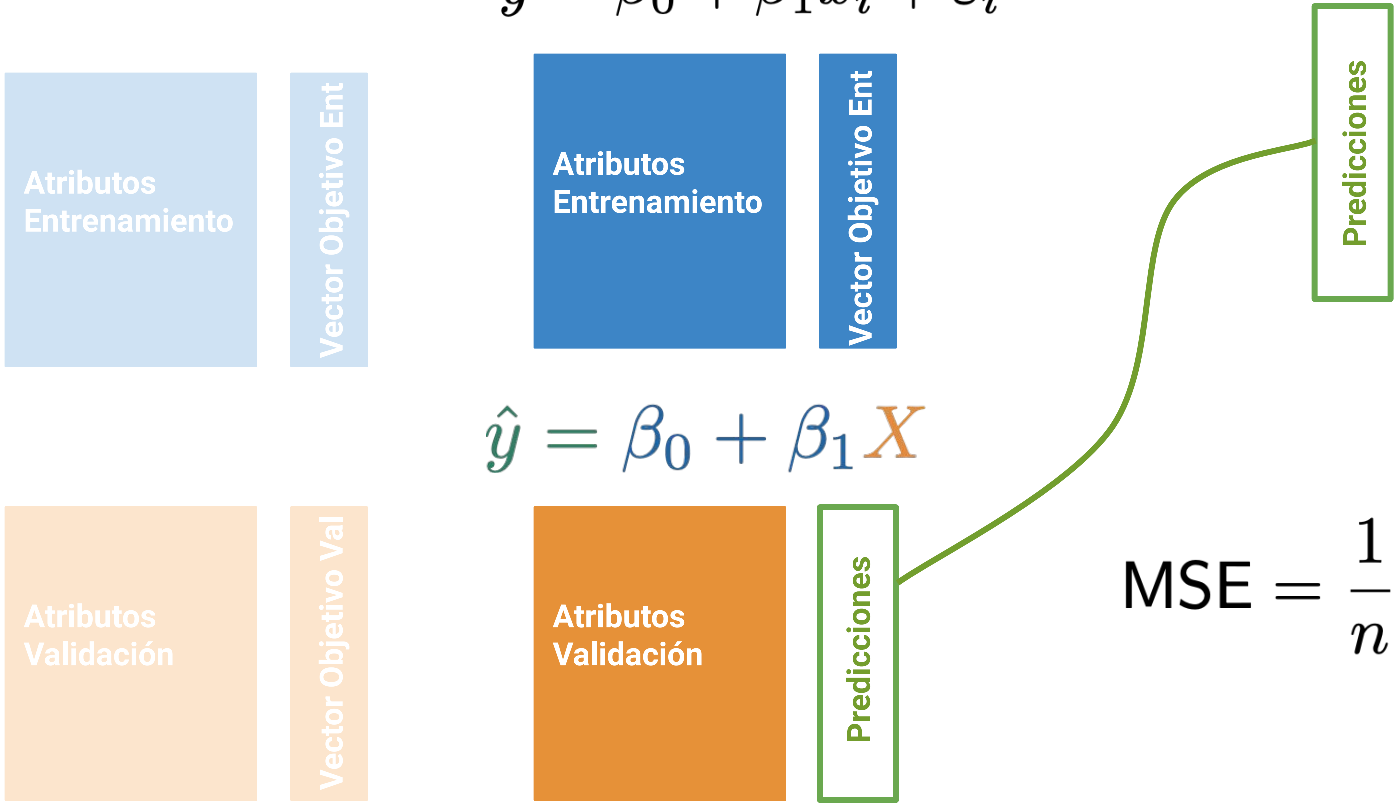
Predicciones

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

# La evaluación del modelo

$$y = \beta_0 + \beta_1 x_i + \varepsilon_i$$

$$\hat{y} = \beta_0 + \beta_1 X$$

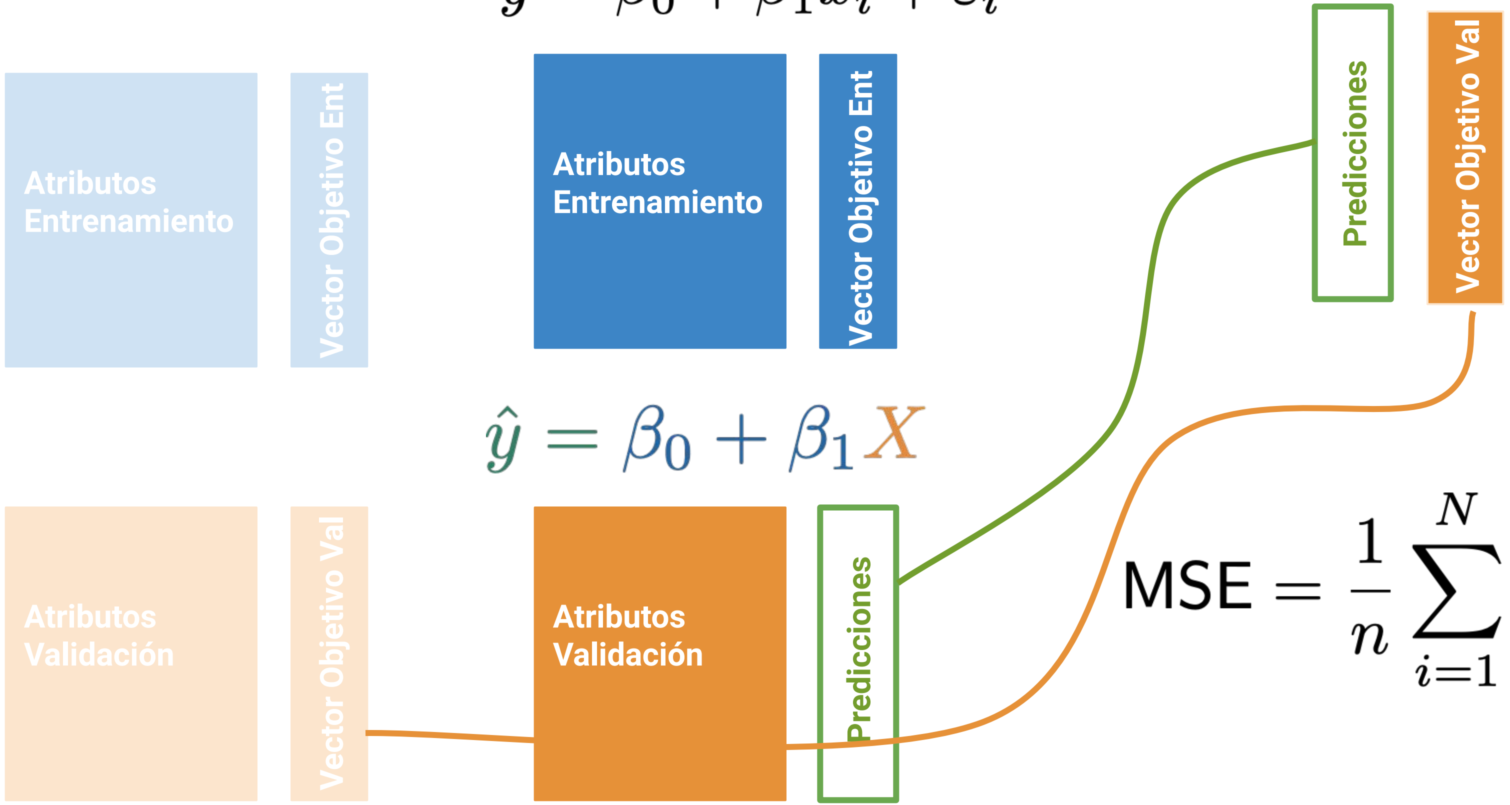



$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

# La evaluación del modelo

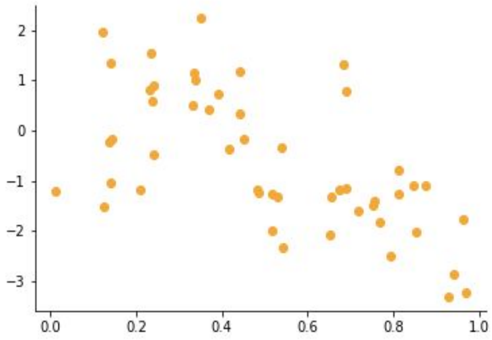
$$y = \beta_0 + \beta_1 x_i + \varepsilon_i$$

$$\hat{y} = \beta_0 + \beta_1 X$$

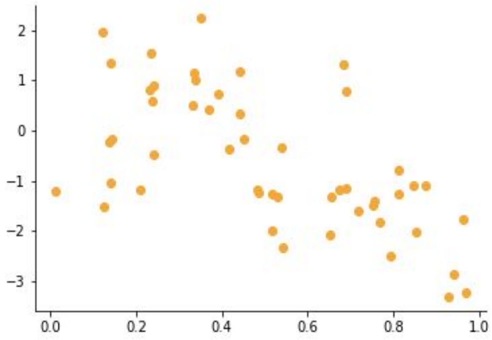


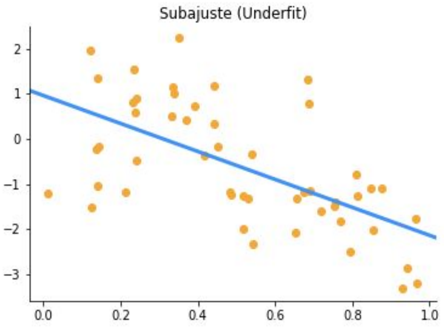

$$\mathrm{MSE} = \frac{1}{n} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

# Subajuste/Sobreajuste

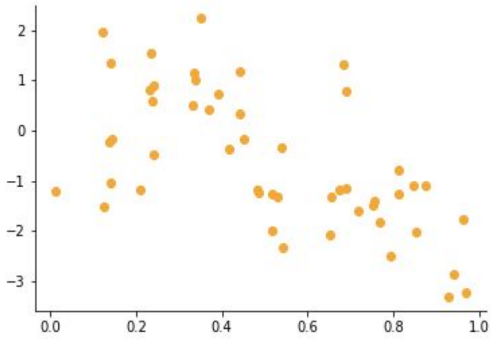


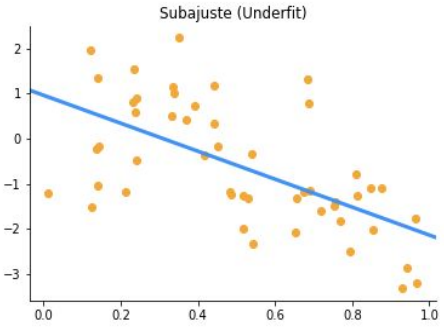
# Subajuste/Sobreajuste



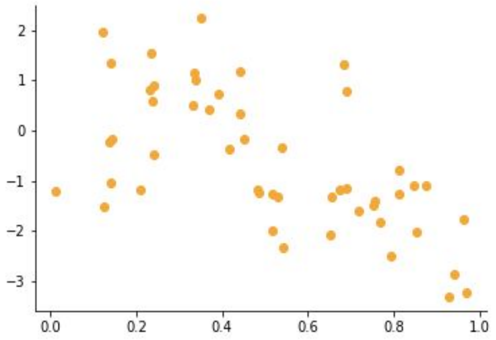
# Subajuste/Sobreajuste

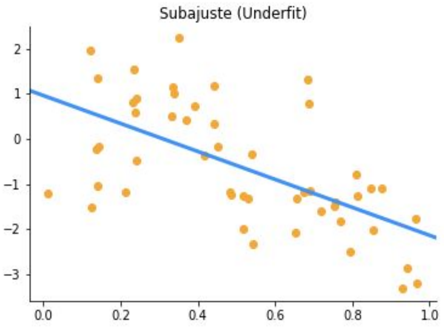





**Alto Sesgo (forma funcional inflexible)**

# Subajuste/Sobreajuste

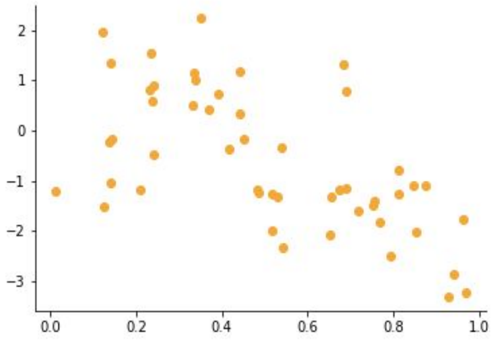


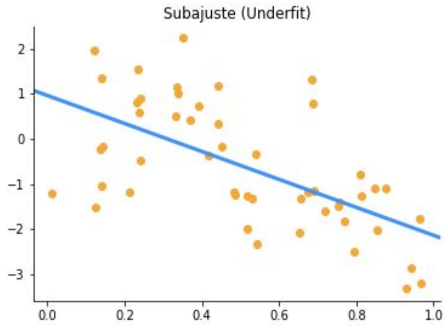



**Alto Sesgo (forma funcional inflexible)**

**Menor capacidad explicativa en muestra**

# Subajuste/Sobreajuste



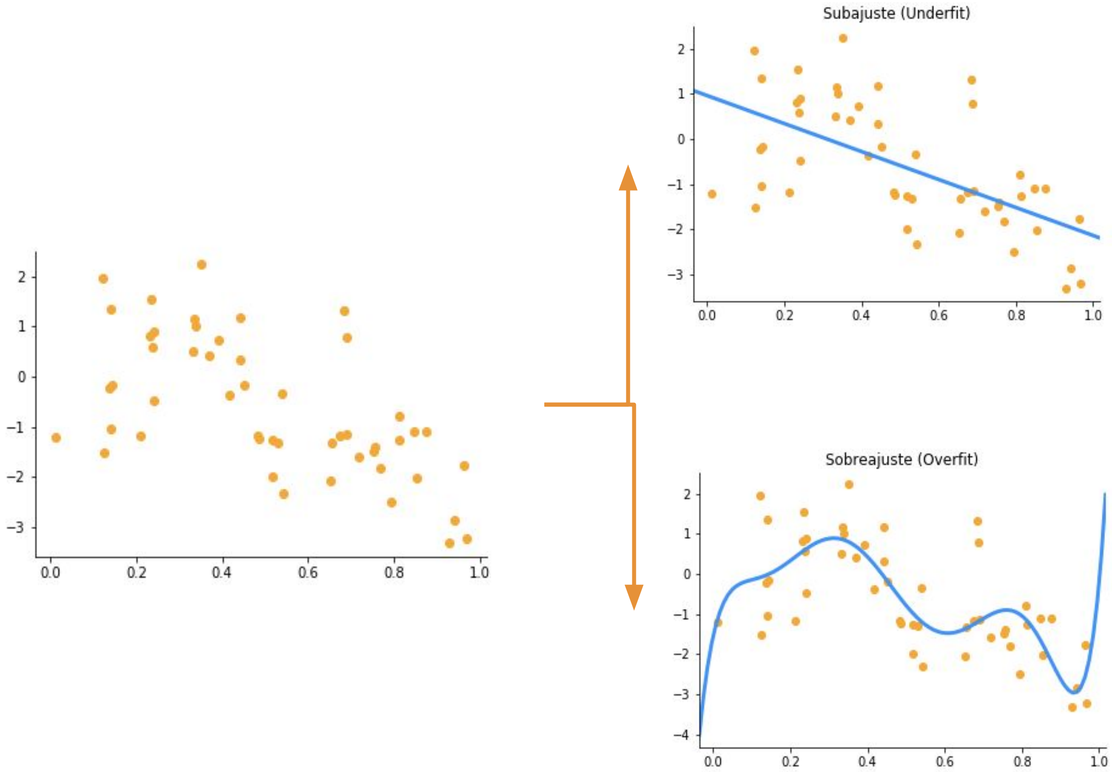




**Alto Sesgo (forma funcional inflexible)**

**Menor capacidad explicativa en muestra**

**Mejores chances de ser generalizable**

# Subajuste/Sobreajuste

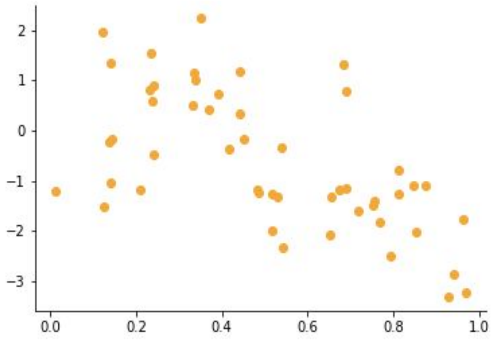


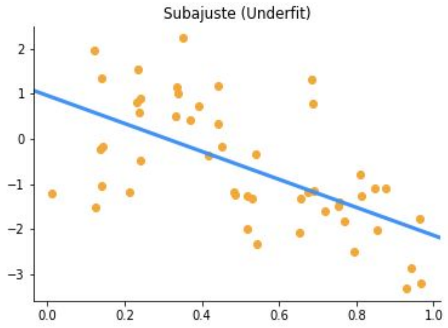

**Alto Sesgo (forma funcional inflexible)**

**Menor capacidad explicativa en muestra**

**Mejores chances de ser generalizable**

# Subajuste/Sobreajuste

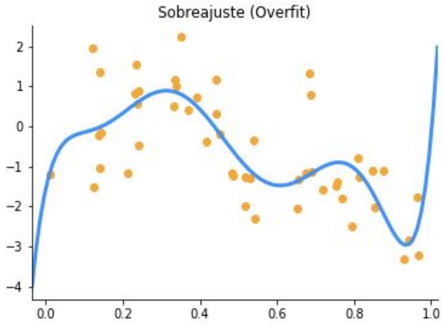





Alto Sesgo (forma funcional inflexible)

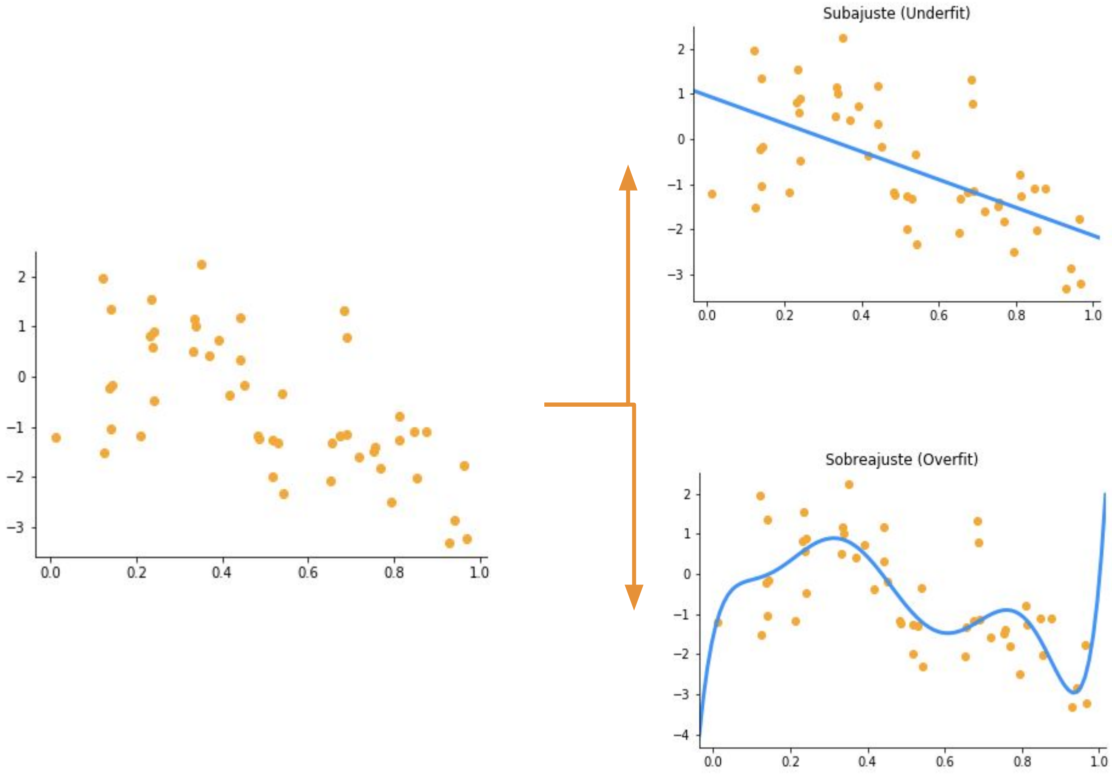
Menor capacidad explicativa en muestra

Mejores chances de ser generalizable




Alta Varianza ( Forma funcional acoplada)

# Subajuste/Sobreajuste




Alto Sesgo (forma funcional inflexible)

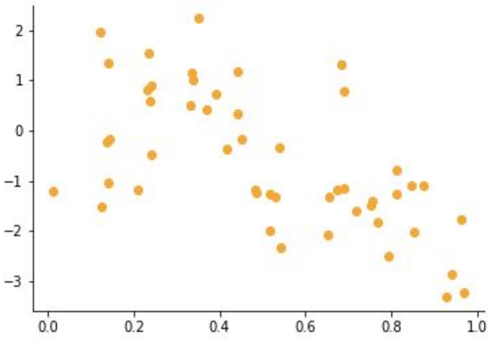
Menor capacidad explicativa en muestra

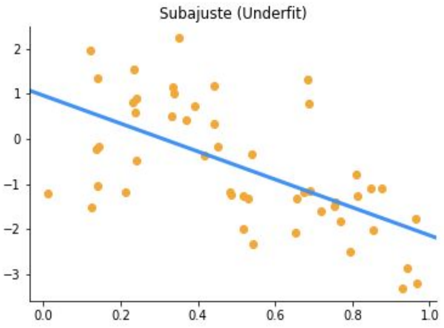
Mejores chances de ser generalizable

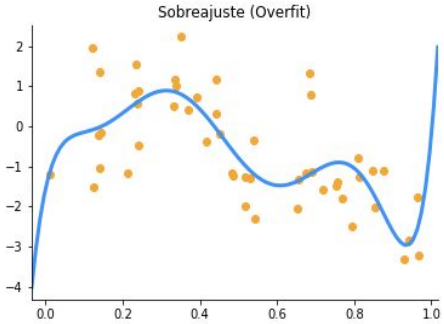
Alta Varianza ( Forma funcional acoplada)

Mayor capacidad explicativa en muestra

# Subajuste/Sobreajuste




**Alto Sesgo (forma funcional inflexible)**

**Menor capacidad explicativa en muestra**

**Mejores chances de ser generalizable**

**Alta Varianza ( Forma funcional acoplada)**

**Mayor capacidad explicativa en muestra**

**Menores chances de ser generalizable**

# Elementos básicos de la regularización

# Punto de partida: La regresión lineal

$$\beta_{\text{OLS}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Para obtener la ecuación de la recta de un conjunto de datos, implementamos MCO.
De esta manera nos aseguramos que las estimaciones sean ELIO.

# Punto de partida: La regresión lineal

$$\beta_{\mathrm{OLS}} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

- Para obtener la ecuación de la recta de un conjunto de datos, implementamos MCO.
- De esta manera nos aseguramos que las estimaciones sean ELIO.
- El método de mínimos cuadrados busca encontrar el argumento que minimice la siguiente función.

# Punto de partida: La regresión lineal

$$\beta_{\text{OLS}} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

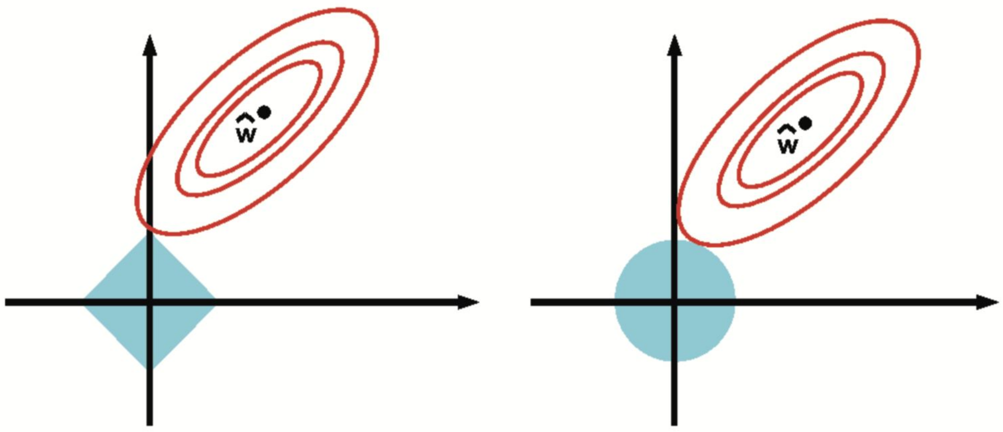
- Para obtener la ecuación de la recta de un conjunto de datos, implementamos MCO.
- De esta manera nos aseguramos que las estimaciones sean ELIO.
- El método de mínimos cuadrados busca encontrar el argumento que minimice la siguiente función.
- Esta función es la diferencia entre lo predicho y lo observado en cada observación de un conjunto de datos.

# ¿Y por qué debo regularizar?

Existen parámetros estimables que pueden tener un peso exagerado en nuestro entrenamiento.

- **Complejidad computacional:** En la medida que agregamos más parámetros, hacemos más costosa de estimar nuestra ecuación.
- Regularizar (en versiones específica), permite seleccionar de mejor manera los atributos de un conjunto de datos.
- Regularizar permite una evaluación agnóstica de los parámetros inferidos, dependiendo de elementos estrictamente ajenos a los producidos por el modelo.

# Nomenclatura necesaria




**Norma L1**: Sintetiza la distancia entre dos vectores mediante la norma absoluta. Se conoce como Lasso.

**Norma L2**: Sintetiza la distancia entre dos vectores mediante la norma euclídea. Se conoce como Ridge.

# Ridge

# Ridge

$$\beta_{\mathsf{OLS}} = \underset{\boldsymbol{\beta}}{\mathsf{argmin}} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

- Un modelo OLS puede sufrir de coeficientes inflados, conllevando a overfit en la muestra de entrenamiento.

# Ridge

$$\beta_{\text{Ridge}} = \underset{\boldsymbol{\beta}}{\text{argmin}} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

- Un modelo OLS puede sufrir de coeficientes inflados, conllevando a overfit en la muestra de entrenamiento.
- Ridge modifica la superficie de penalización de los coeficientes mediante el **hiperparámetro** lambda.

# Ridge

$$\beta_{\text{Ridge}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

- Un modelo OLS puede sufrir de coeficientes inflados, conllevando a overfit en la muestra de entrenamiento.
- Ridge modifica la superficie de penalización de los coeficientes mediante el **hiperparámetro** lambda.
- Lambda gobierna la superficie de penalización que está determinada por la cantidad de parámetros inferidos en el modelo.
- Dado que tiene una forma cuadrática, suaviza pero no elimina atributos irrelevantes.

# Elección de Hiperparámetros

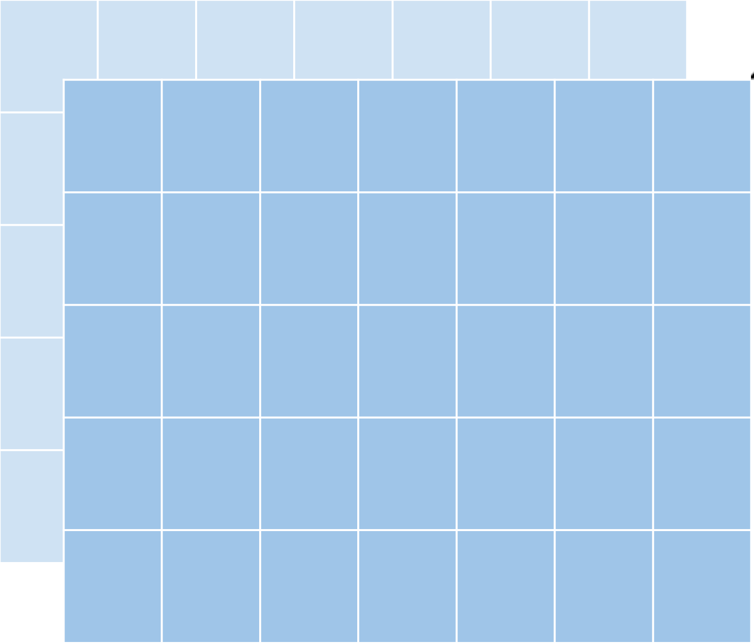
# Elección de lambda

$$\lambda = 0.001$$

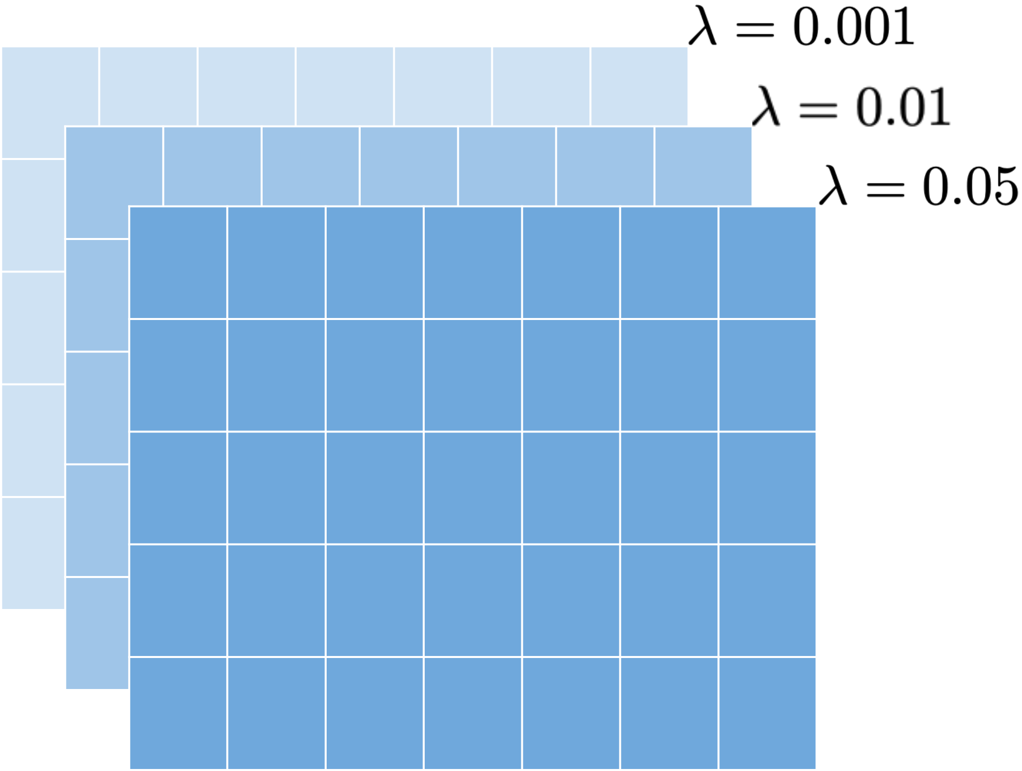
# Elección de lambda

$\lambda = 0.001$

$\lambda = 0.01$



# Elección de lambda



# Elección de lambda

$\lambda = 0.001$

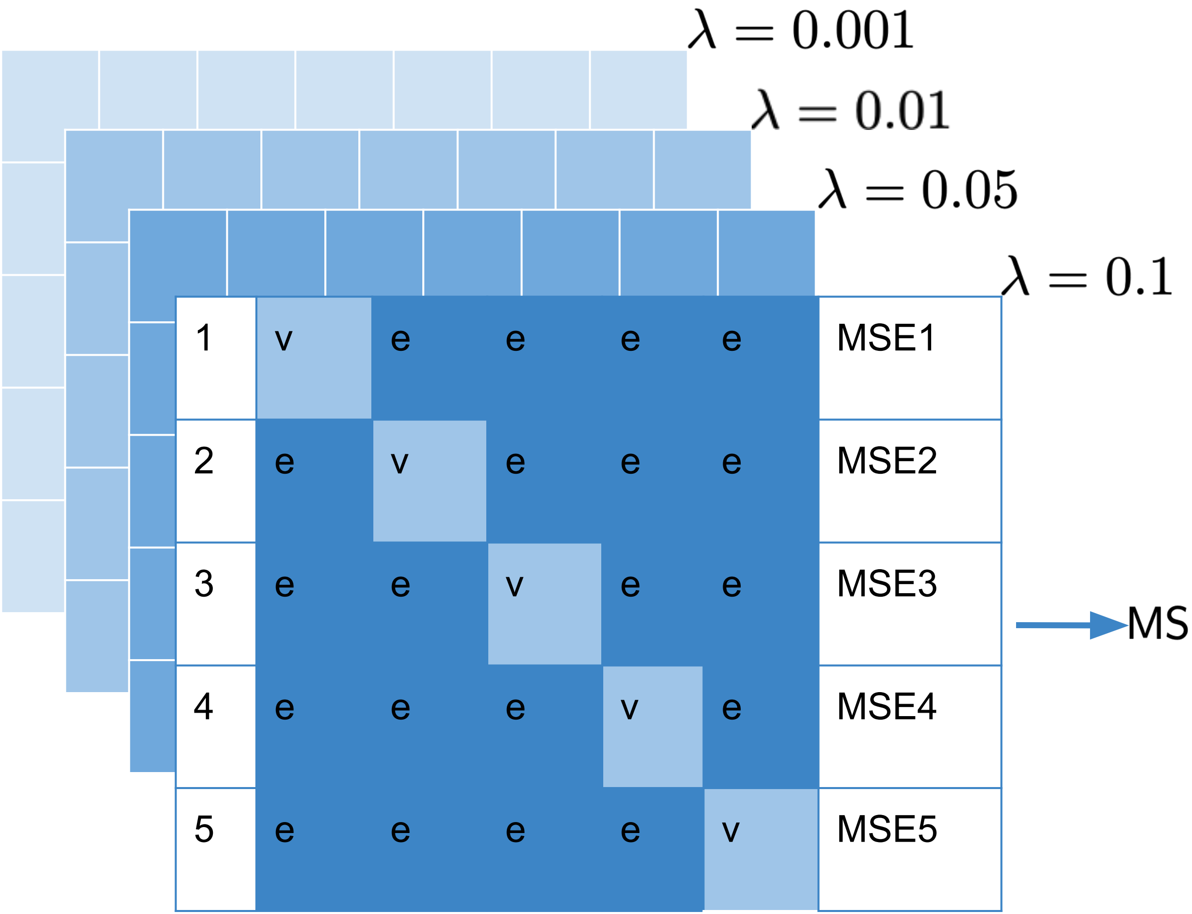
$\lambda = 0.01$

$\lambda = 0.05$

$\lambda = 0.1$

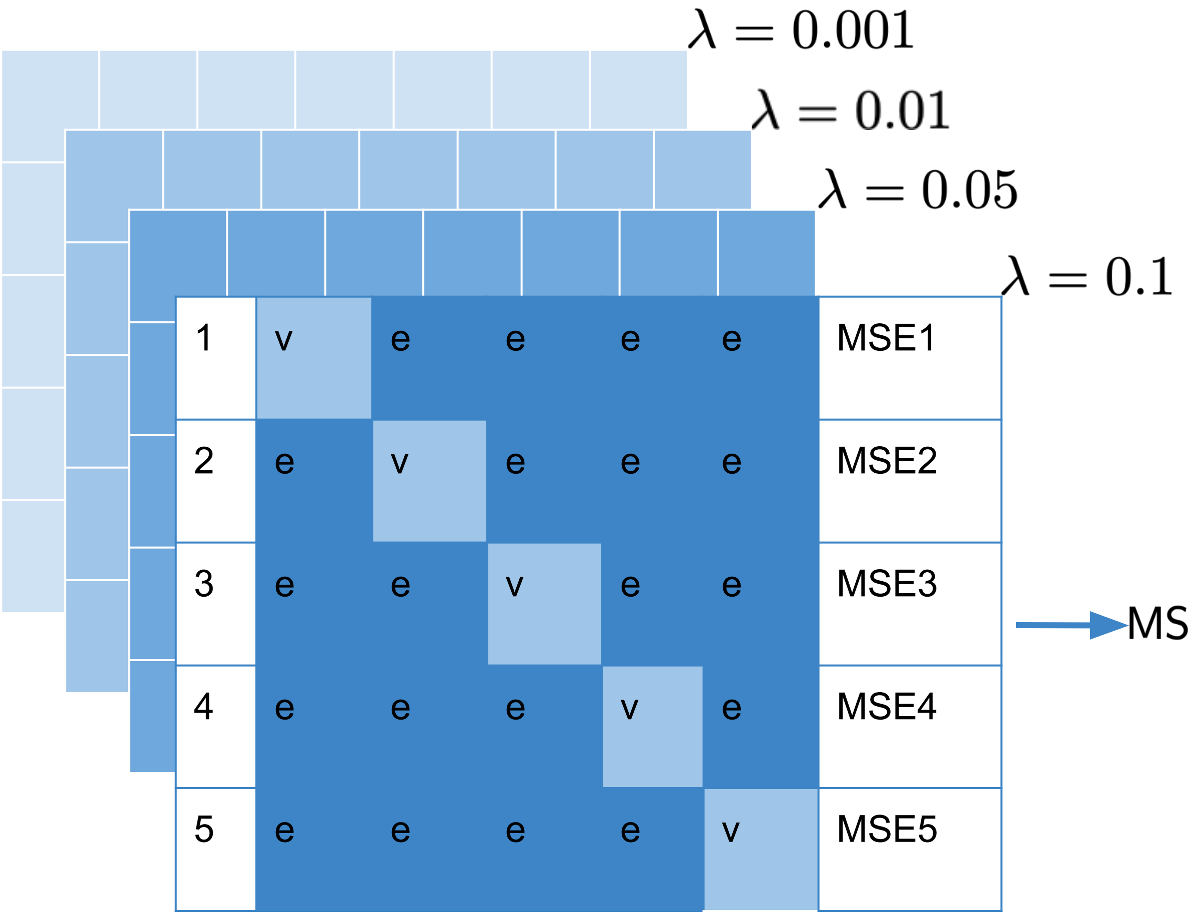
| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | v | e | e | e | e | MSE1 |
| 2 | e | v | e | e | e | MSE2 |
| 3 | e | e | v | e | e | MSE3 |
| 4 | e | e | e | v | e | MSE4 |
| 5 | e | e | e | e | v | MSE5 |

# Elección de lambda




$$\text{MSE}_{\text{cv}} = \frac{1}{\#\text{cv}} \sum_{i=1}^{CV} \text{MSE}$$

# Elección de lambda




| Lambda | MSEcv |
|---|---|
| $\lambda = 0.001$ | 1 |
| $\lambda = 0.01$ | 2 |
| $\lambda = 0.05$ | 3 |
| $\lambda = 0.1$ | 4 |

$$\mathrm{MSE}_{\mathrm{cv}} = \frac{1}{\#\mathrm{cv}} \sum_{i=1}^{CV} \mathrm{MSE}$$

# Lasso

# Lasso

$$\beta_{\mathsf{OLS}} = \underset{\boldsymbol{\beta}}{\mathsf{argmin}} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

- **Lasso**: Least Absolute Shrinkage and Selection Operator.

# Lasso

$$\beta_{\text{Lasso}} = \underset{\boldsymbol{\beta}}{\text{argmin}} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

- **Lasso**: Least Absolute Shrinkage and Selection Operator.
- **Principal diferencia con Ridge**: **Permite seleccionar y eliminar atributos irrelevantes del modelo**
- De igual manera que en Ridge, el hiperparámetro lambda define el área de la superficie de penalización.

# Lasso

$$\beta_{\text{Lasso}} = \underset{\boldsymbol{\beta}}{\text{argmin}} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

- **Lasso**: Least Absolute Shrinkage and Selection Operator.
- **Principal diferencia con Ridge**: **Permite seleccionar y eliminar atributos irrelevantes del modelo**.
- De igual manera que en Ridge, el hiperparámetro lambda define el área de la superficie de penalización.
- La diferencia radica en la norma de penalización.
- Dado que la superficie de penalización es absoluta, tenderá a entregar soluciones dispersas.

# Elastic Net

# Elastic Net

$$\beta_{\text{OLS}} = \underset{\boldsymbol{\beta}}{\text{argmin}} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

- Elastic Net combina ambas normas de penalización.

# Elastic Net

$$\beta_{\text{ElasticNet}} = \underset{\beta}{\operatorname{argmin}} \sum_{i}^{n} (y_i - \hat{y}_i)^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} \beta_j^2$$

- Elastic Net combina ambas normas de penalización.
- L1 nos asegura una selección de atributos.

# Elastic Net

$$\beta_{\text{ElasticNet}} = \underset{\beta}{\text{argmin}} \sum_{i}^{n} (y_i - \hat{y}_i)^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} \beta_j^2$$

- Elastic Net combina ambas normas de penalización.
- L1 nos asegura una selección de atributos
- L2 nos asegura una penalización parsimoniosa de los coeficientes de los atributos.

# Elastic Net

$$\beta_{\text{ElasticNet}} = \underset{\beta}{\operatorname{argmin}} \sum_{i}^{n} (y_i - \hat{y}_i)^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} \beta_j^2$$

- Elastic Net combina ambas normas de penalización.
- L1 nos asegura una selección de atributos.
- L2 nos asegura una penalización parsimoniosa de los coeficientes de los atributos.
- Existe un parámetro que gobierna la dominancia entre ambas formas de penalización.

{desafío} latam_

Academia de talentos digitales

www.desafiolatam.com