

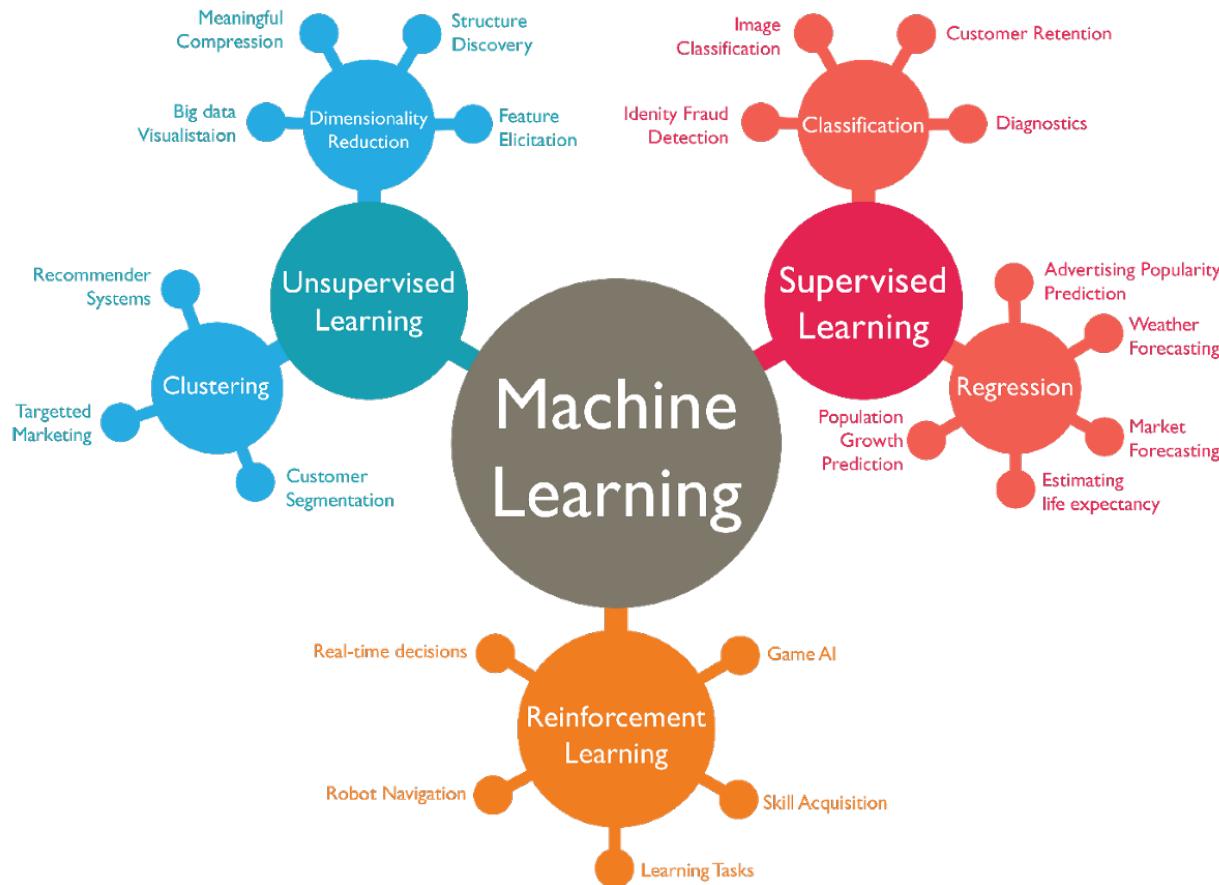
ALGORITMOS DE APRENDIZAJE DE MÁQUINA

CLASE 19

ALGORITMOS DE APRENDIZAJE DE MÁQUINA

- **Algoritmos** → procedimiento, método o conjunto de pasos o reglas para lograr una tarea.
- **Aprendizaje** → proceso de convertir experiencia (i.e. data) en habilidad o conocimiento (i.e. un modelo o programa capaz de realizar una tarea)
- **Algoritmos de aprendizaje de máquina (ML)** → métodos computacionales que utilizan data anterior (i.e. experiencia) para generar modelos o programas capaces de realizar tareas como predecir, clasificar, agrupar, ordenar o reducir dimensionalidad.
- Para una tarea dada, pueden proponerse múltiples algoritmos posibles
 - El éxito de un algoritmo de ML se evalúa en base a métricas de precisión, eficiencia y tiempo computacional.
 - La elección del algoritmo a usar dependerá de: contexto y complejidad del problema, suposiciones de base, tamaño y variedad de la data disponible.
 - Implementarlo

ALGORITMOS DE APRENDIZAJE DE MÁQUINA: EJEMPLOS



TIPOS DE APRENDIZAJE Y TAREAS COMUNES

Aprendizaje supervisado

predicción de un cierto output para un input dado, a partir de un conjunto de **datos de entrenamiento** o pares de input/outputs conocidos.

Regresión: predicción de un **valor real** para cada ítem

Clasificación: predicción de una **categoría** para cada ítem de un conjunto. La clasificación puede ser *binaria* o *multiclas*.

Ranking: aprender a **ordenar** ítems de acuerdo a algún criterio

Measuring the Gross Domestic Product (GDP): The Ultimate Data Science Project

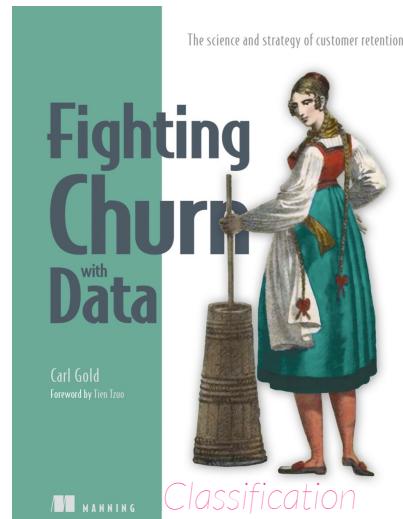
by Brian C. Moyer and Abe Dunn

Published on Jan 31, 2020

<https://hdsr.mitpress.mit.edu/pub/5pkkan15/release/3?readingCollection=a133a0a2>

Regression

Otros ejemplos: precios, temperaturas, indicadores económicos, contagios, etc...



Otros ejemplos:
Spam filter,
paga/ no paga,
especies,
yes/no,
usos suelo (construido/área
verde/habitacional/comercial),
... etc

TIPOS DE APRENDIZAJE Y TAREAS COMUNES

Aprendizaje no supervisado

algoritmos en los cuáles sólo se conoce la [data de entrada no etiquetada](#), no hay outputs previos conocidos.

Es más difícil evaluar la precisión de las predicciones.

Unsupervised Content-Based Identification of Fake News Articles with Tensor Decomposition Ensembles

Seyedmehdi Hosseiniomotagh
University of California Riverside
shoss007@ucr.edu

Evangelos E. Papalexakis
University of California Riverside
epapalex@cs.ucr.edu

Article | Published: 20 January 2020

Clustering of health, crime and social-welfare inequality in 4 million citizens from two nations

Leah S. Richmond-Rakerd , Stephanie D'Souza, Signe Hald Andersen, Sean Hogan, Renate M. Houts, Richie Poulton, Sandhya Ramrakha, Avshalom Caspi, Barry J. Milne & Terrie E. Moffitt

Nature Human Behaviour 4, 255–264 (2020) | [Cite this article](#)

Clustering: particionar un set de ítems en subconjuntos homogéneos

Reducción de dimensionalidad: transformar una representación de ítems en una representación con menos dimensiones, pero preservando algunas propiedades de la representación inicial.

Novel Clustering-Based Web Service Recommendation Framework

Priya Bhaskar Pandharbale (School of Computer Engineering, KIIT University (Deemed), Bhubaneswar, India), Sachin Nandan Mohanty (Department of Computer Sc & Engineering, Vardhaman College of Engineering (Autonomous), Hyderabad, India) and Alok Kumar Jagadev (School of Computer Engineering, KIIT University (Deemed), Bhubaneswar, India)

Source Title: [International Journal of System Dynamics Applications \(IJSDA\)](#) 11(5)
Copyright: © 2022 | Pages: 15
DOI: [10.4018/IJSDA.20220901.oa1](https://doi.org/10.4018/IJSDA.20220901.oa1)



Journal of Cleaner Production
Volume 275, 1 December 2020, 122942



Article | Open Access | Published: 05 January 2021

Benchmarking joint multi-omics dimensionality reduction approaches for the study of cancer

Laura Cantini , Pooya Zakeri, Celine Hernandez, Aurelien Naldi, Denis Thieffry, Elisabeth Remy & Anaïs Baudot 

Nature Communications 12, Article number: 124 (2021) | [Cite this article](#)

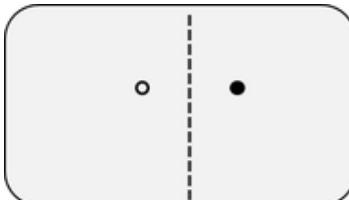
A multi-stage method to predict carbon dioxide emissions using dimensionality reduction, clustering, and machine learning techniques

Abbas Mardani , Huchang Liao , Mehrbakhsh Nilashi , Melfi Alrasheedi , Fausto Cavallaro 

TIPOS DE APRENDIZAJE Y TAREAS COMUNES

Aprendizaje semi-supervisado

predicción de un cierto output para un input dado, a partir de un conjunto de datos de entrenamiento que incluyen **datos etiquetados y no etiquetados**

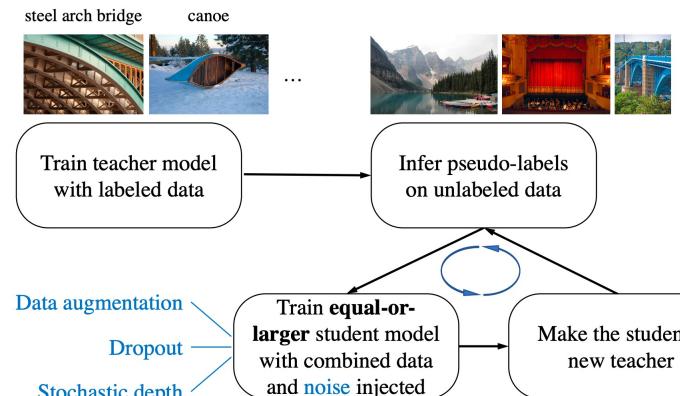


Clasificación con datos etiquetados

Clasificación con datos etiquetados y no etiquetados

Común en situaciones donde la data sin etiquetar es fácilmente accesible, pero el etiquetado es muy costoso (por falta de expertise, tiempo, acceso a datos).

Auto-entrenamiento (self-training), Modelos generativos, Mixtura gaussiana, Métodos basados en gráficos... etc



Clasificación de imágenes y audio

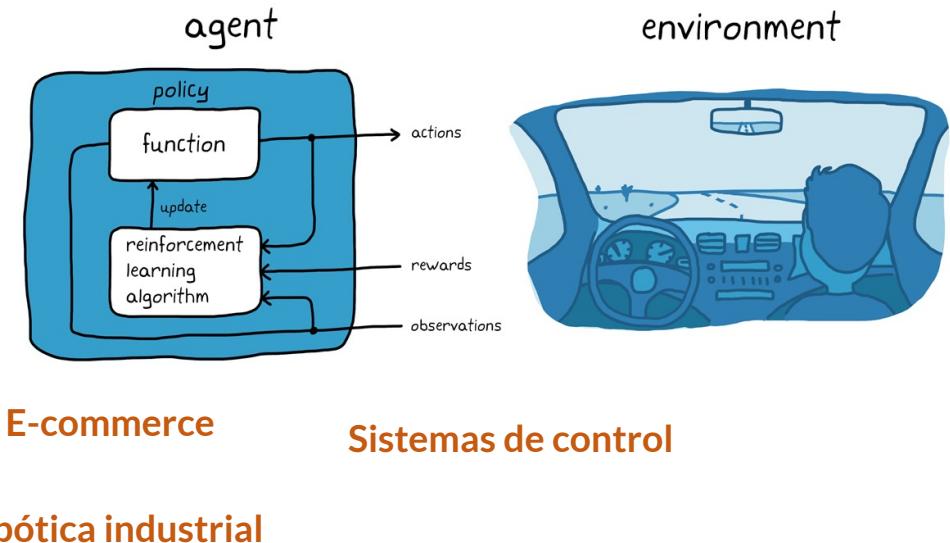
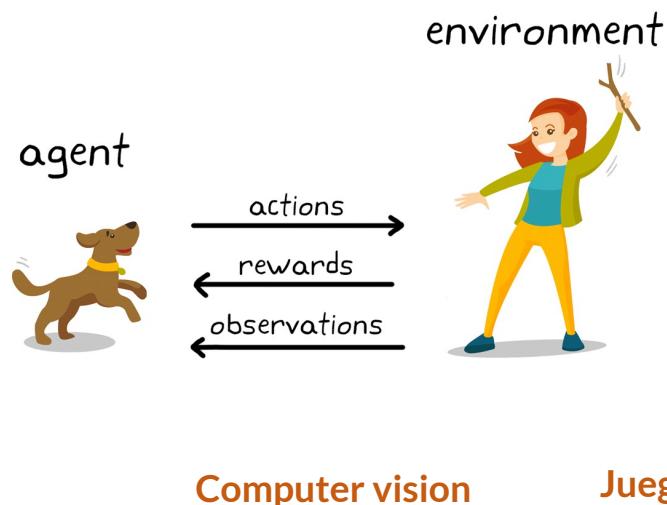
Procesamiento de lenguajes

Clasificación de páginas web

Secuenciación de proteínas

TIPOS DE APRENDIZAJE Y TAREAS COMUNES

Aprendizaje reforzado: Tipo de algoritmos de ML en los cuales el agente (software) aprenden a realizar ciertas acciones que maximizan una cierta “**recompensa**”, mediante una repetición de interacciones con un ambiente dinámico (prueba-error).



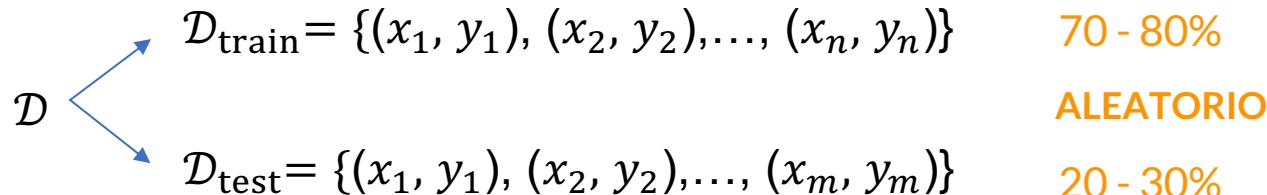
¿QUÉ TIPO DE ALGORITMO USAR?

- ¿Cuál es la pregunta que se trata de responder?
- ¿Cómo puedo enunciar mi pregunta en términos de un problema de aprendizaje de máquina?
- ¿Tengo suficientes datos para representar el problema que busco resolver?
- ¿Qué features extraje de la data, y son adecuadas para las predicciones?
- ¿Cómo mido el éxito de mi aplicación?

APRENDIZAJE SUPERVISADO

APRENDIZAJE SUPERVISADO

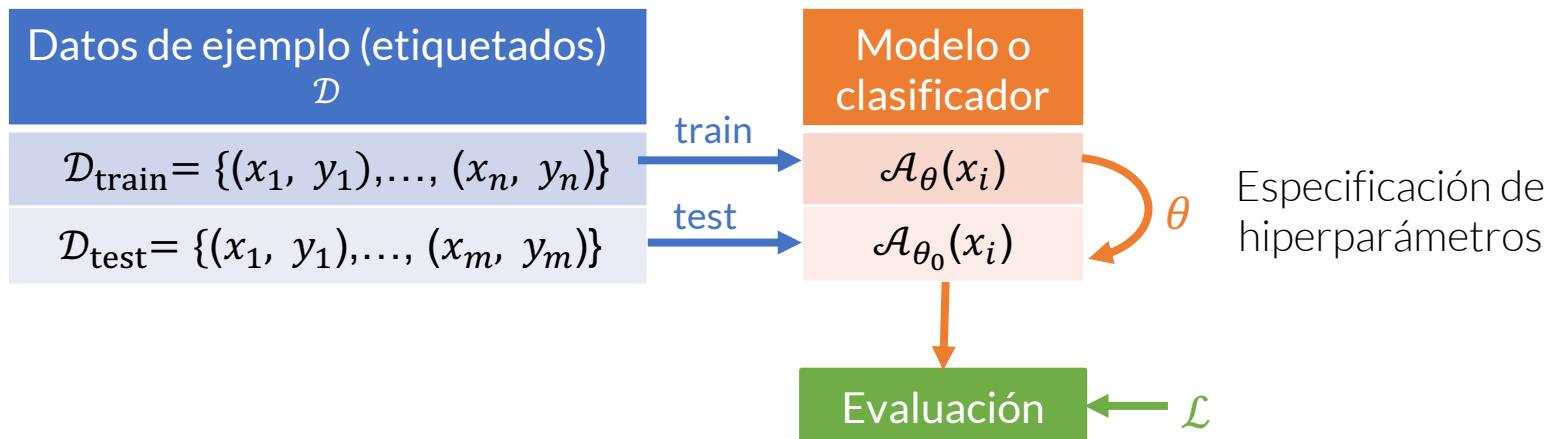
- El objetivo es realizar predicciones precisas para *nuevos datos* con características similares a los datos usados para construir el modelo → **generalización**
- **Entrenamiento y testeо:** el conjunto de datos de ejemplo \mathcal{D} se divide aleatoriamente en dos
 - **Datos de entrenamiento ($\mathcal{D}_{\text{train}}$)** → para entrenar el modelo
 - **Datos de prueba ($\mathcal{D}_{\text{test}}$)** → para evaluar qué tan bien funciona el modelo frente a datos que no conoce. No se usa para el entrenamiento.
 - **Etiquetas (y)** → valores o categorías asignadas a los datos de ejemplo



- **Función de pérdida (\mathcal{L})** → función que mide la diferencia o pérdida entre la predicción y el valor real de una etiqueta o valor.

APRENDIZAJE SUPERVISADO

- El objetivo es realizar predicciones precisas para *nuevos datos* con características similares a los datos usados para construir el modelo → **generalización**
- **Entrenamiento y testeo:**
 - **Hiperparámetros (θ)** → parámetros libres del modelo que no son determinados por el algoritmo, sino entregados como input



APRENDIZAJE SUPERVISADO

MODELOS DE REGRESIÓN

REGRESIÓN EN sklearn

<https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsRegressor.html>

https://scikit-learn.org/stable/modules/linear_model.html

Procedimiento general:

- Dividir datos en conjuntos train/test
- Definir modelo o estimador (regresión, u otro)
- Fit → predict
- Evaluar métricas (score, R², etc)

Methods

<code>fit(X, y[, sample_weight])</code>	Fit linear model.
<code>get_params([deep])</code>	Get parameters for this estimator.
<code>predict(X)</code>	Predict using the linear model.
<code>score(X, y[, sample_weight])</code>	Return the coefficient of determination of the prediction.
<code>set_params(**params)</code>	Set the parameters of this estimator.

`fit(X, y, sample_weight=None)`

Fit linear model.

Parameters:	X : {array-like, sparse matrix} of shape (n_samples, n_features) Training data.
y : array-like of shape (n_samples,) or (n_samples, n_targets)	Target values. Will be cast to X's dtype if necessary.
sample_weight : array-like of shape (n_samples,), default=None	Individual weights for each sample.
Returns:	self : object Fitted Estimator.

PREDICTORES Y OUTCOMES

- En un problema de regresión, buscamos predecir el valor de una variable a partir del valor de otras variables.
- Ejemplo:
 - Predecir el consumo de combustible de un auto a partir de sus características de diseño.

n observaciones
i=1,2,...n

	car_name	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
1	Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
2	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
3	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
4	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2

$$Y = y_1, \dots, y_n$$

outcome / variable dependiente
respuesta

$$X = X_1, \dots, X_p$$

$$X_j = x_{1j}, \dots, x_{ij}, \dots, x_{nj}$$

predictores /variable independiente
covariates / features

REGRESIÓN

- Para predecir Y , asumimos que se relaciona con X mediante una función desconocida f .
- Problema de inferencia \Rightarrow encontrar \hat{f} , la estimación de f
 - Ej: análisis exploratorio
 - ¿cuáles son las variables que determinan Y ?
 - ¿cuál es la contribución de cada una de ellas?
 - ¿qué función analítica puede representar a f ? ¿lineal?
- Problema de **predicción** \Rightarrow usar la data de entrenamiento para predecir o generalizar a situaciones no observadas.
 - No nos interesa la forma de f , sino sólo las **predicciones** \hat{y}_i



$$Y = f(X)$$

$f \rightarrow$ función desconocida
 $\hat{f} \rightarrow$ estimación de f

Predicciones $\rightarrow \hat{y}_i = \hat{f}(x_{i,1}, \dots, x_{i,p})$

REGRESIÓN - kNN

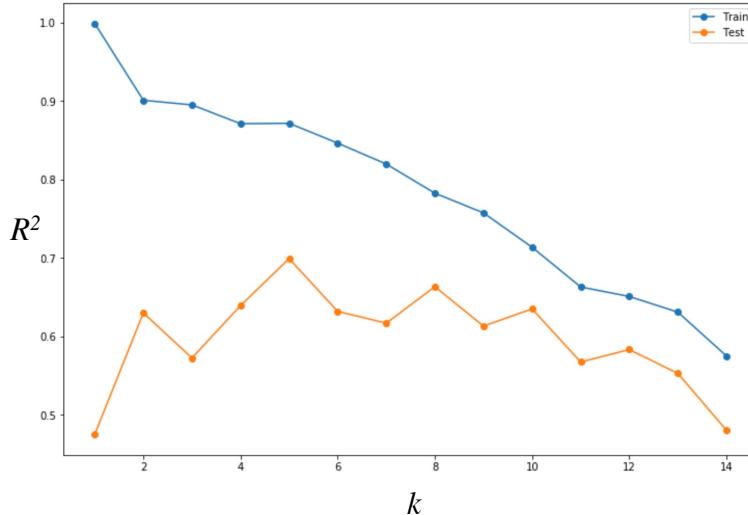
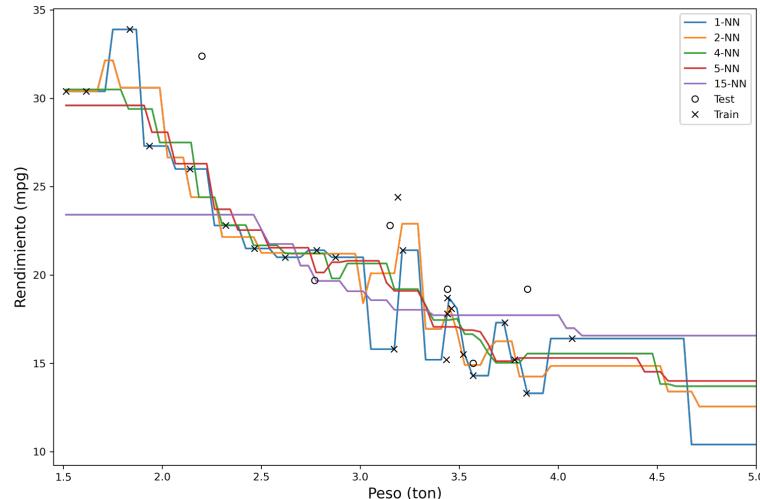
¿Cuál es la forma más simple de predecir \hat{y}_i ?

Usamos el promedio de las respuestas a otras observaciones más cercanas a ella

→ los “ k vecinos más cercanos” o **k -nearest neighbors**:

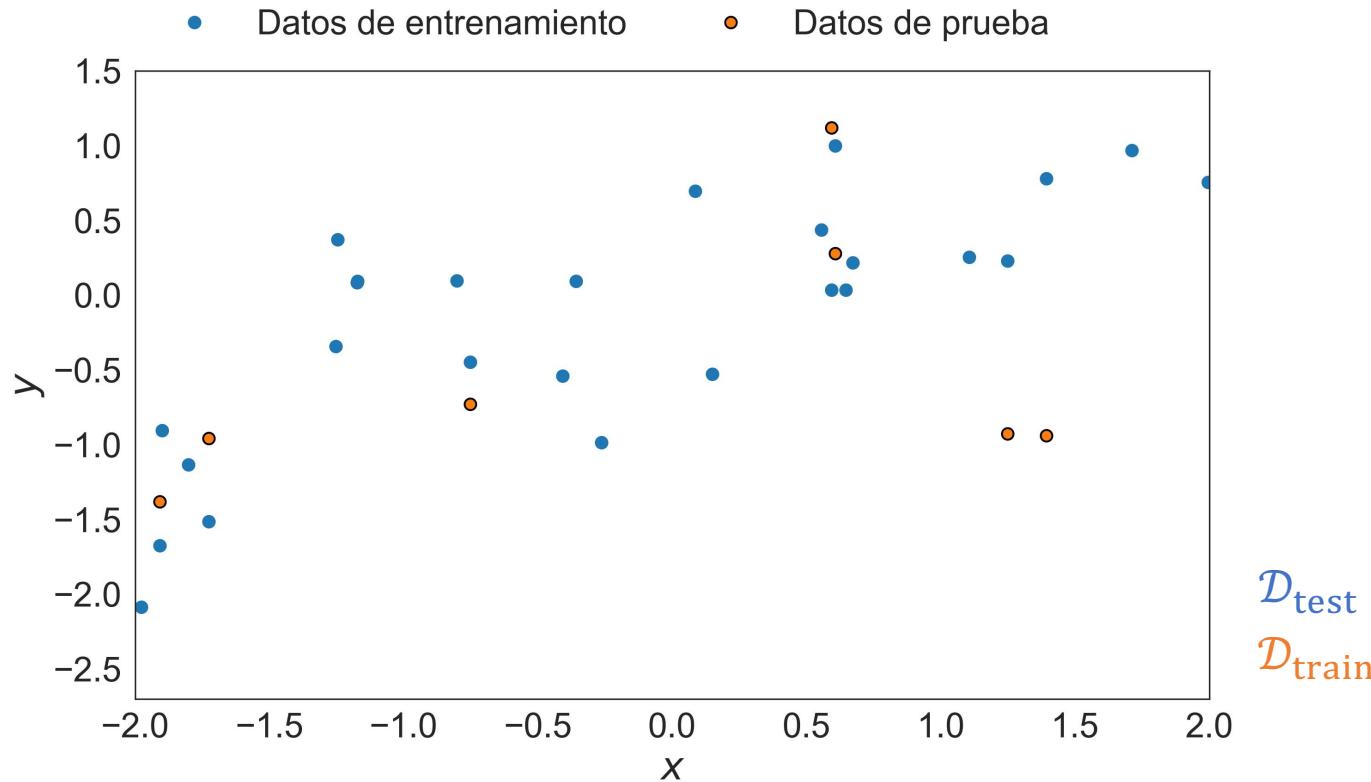
$$\hat{y}_i = \frac{1}{k} \sum_{j=1}^k y_{ij}$$

Donde $\{x_{i1}, \dots, x_{ik}\}$ son las k observaciones más similares (cercanas) a x_i



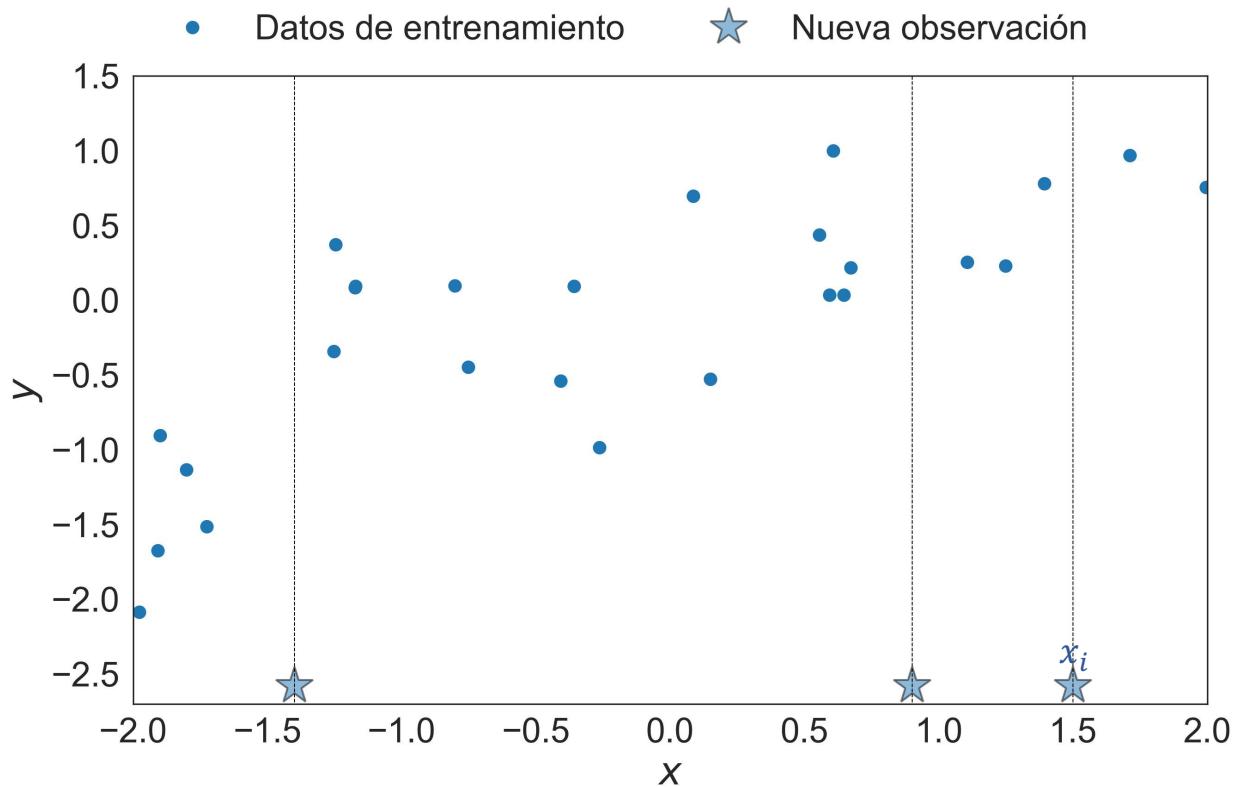
REGRESIÓN kNN

Datos etiquetados (\mathcal{D})



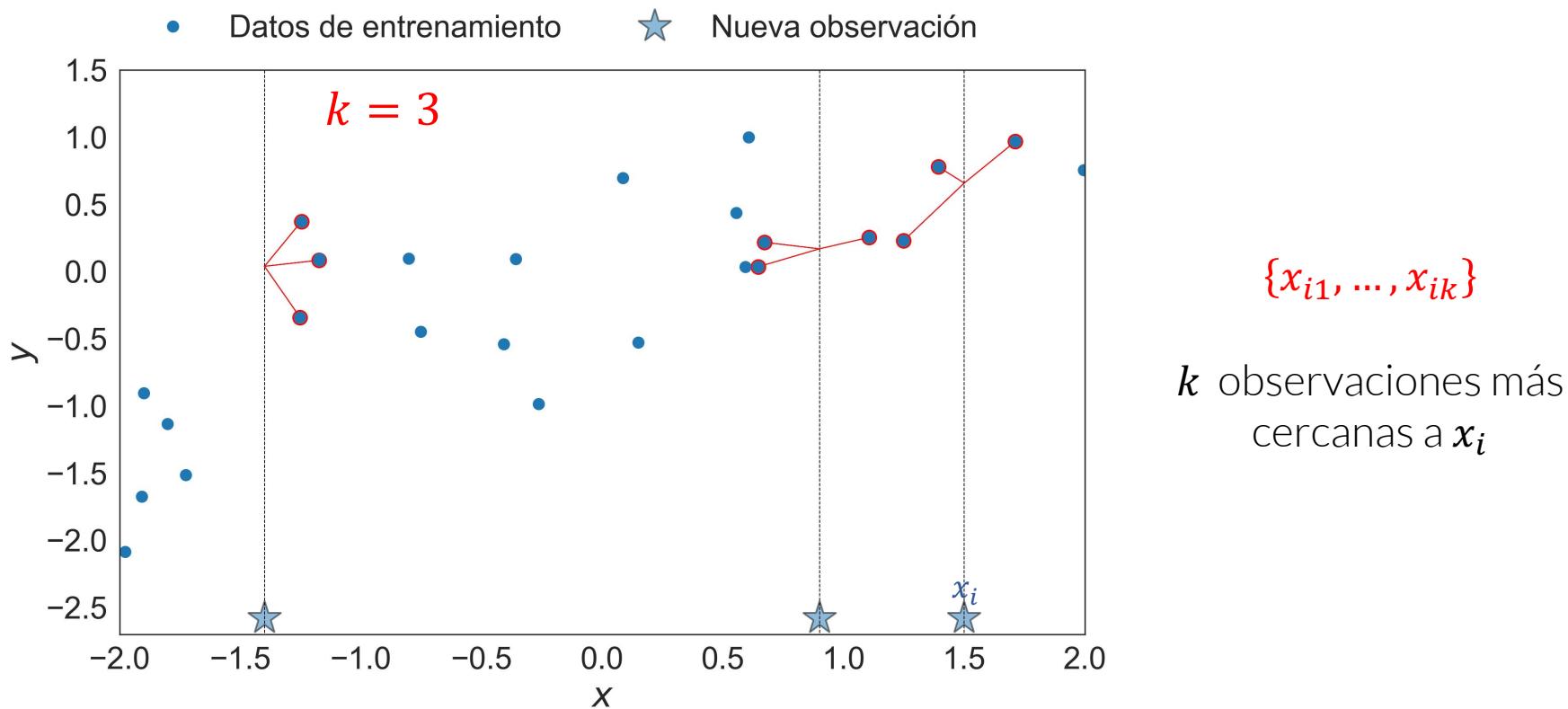
REGRESIÓN kNN

¿Cómo estimar y_i para una nueva observación x_i ?



REGRESIÓN kNN

Identificar los k vecinos más cercanos a cada valor de x



REGRESIÓN kNN

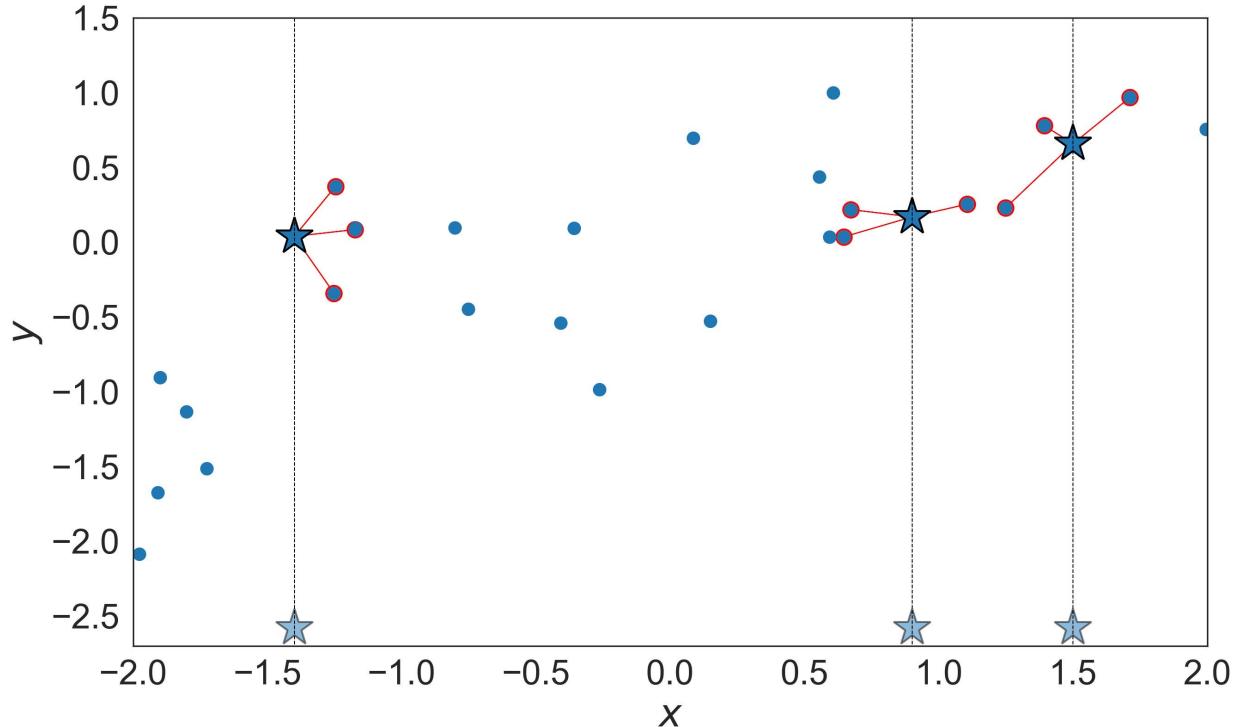
- Datos de entrenamiento



- Nueva observación



- Predicción

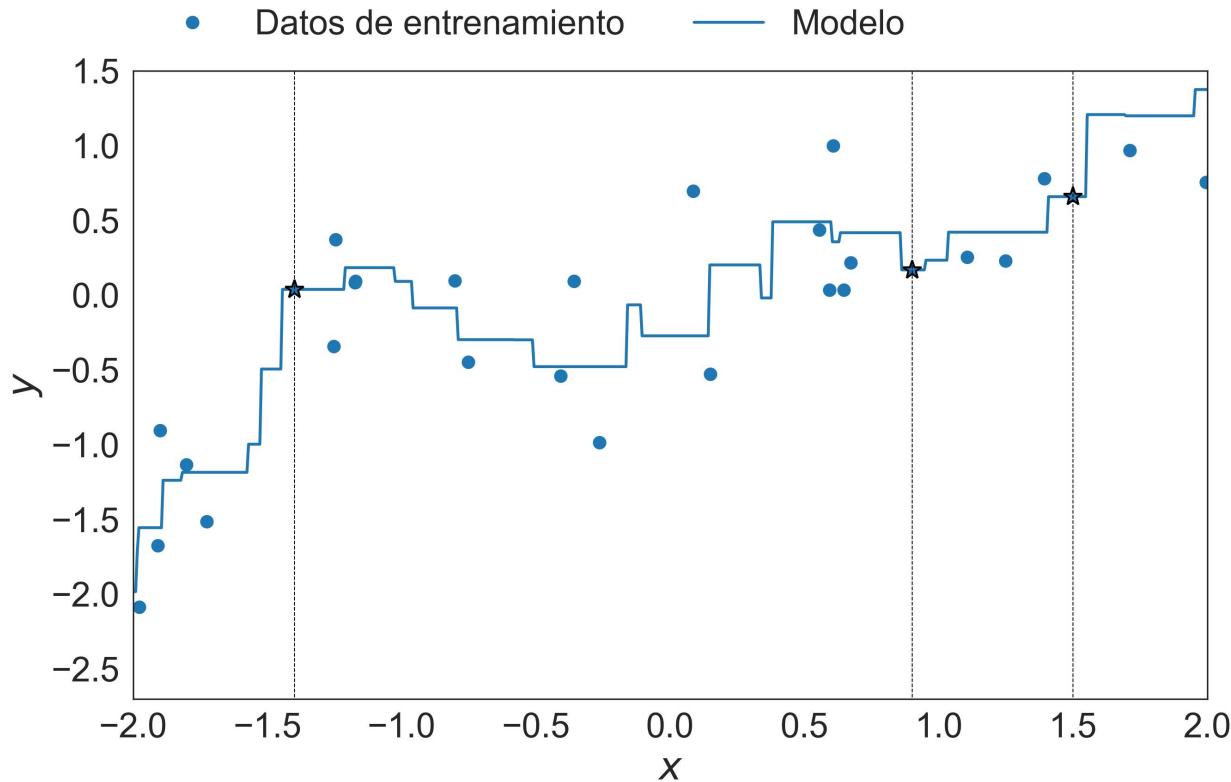


Predicción:

$$\hat{y}_i = \frac{1}{k} \sum_{j=1}^k y_{i_j}$$

REGRESIÓN kNN

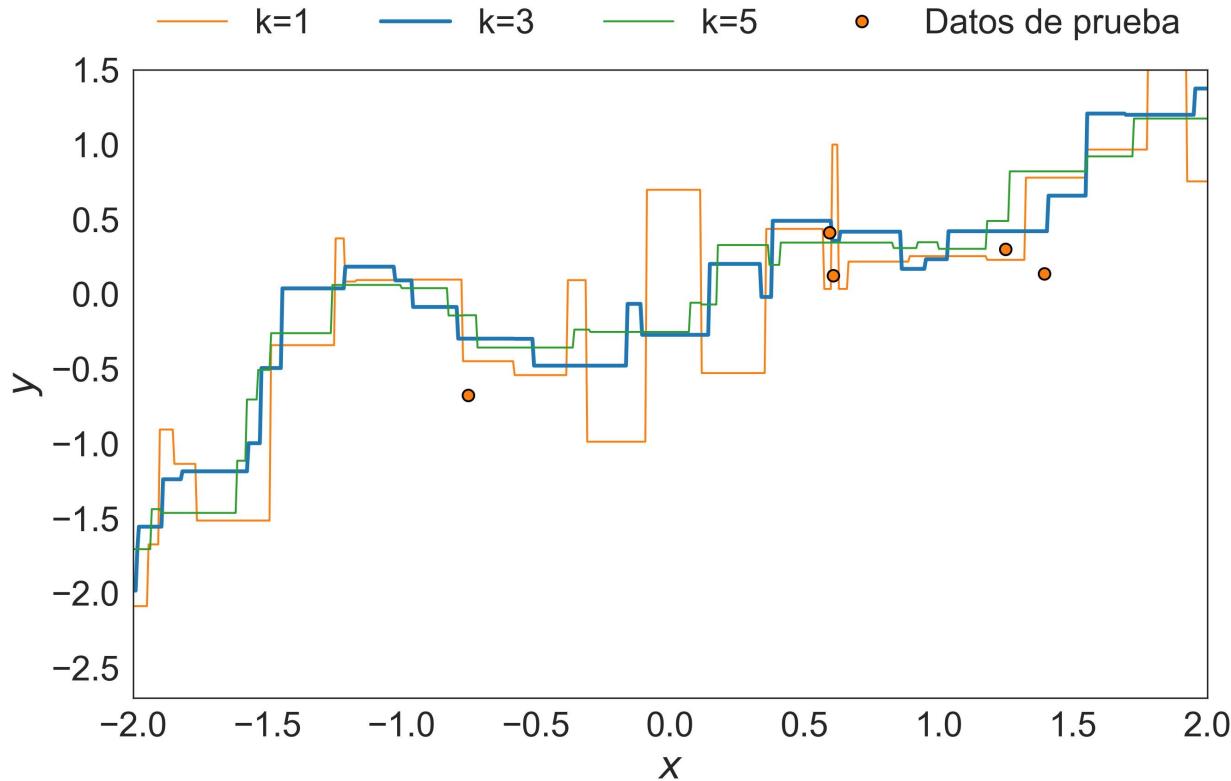
Modelo entrenado para $k = 3$



¿Qué pasa si cambiamos k ?

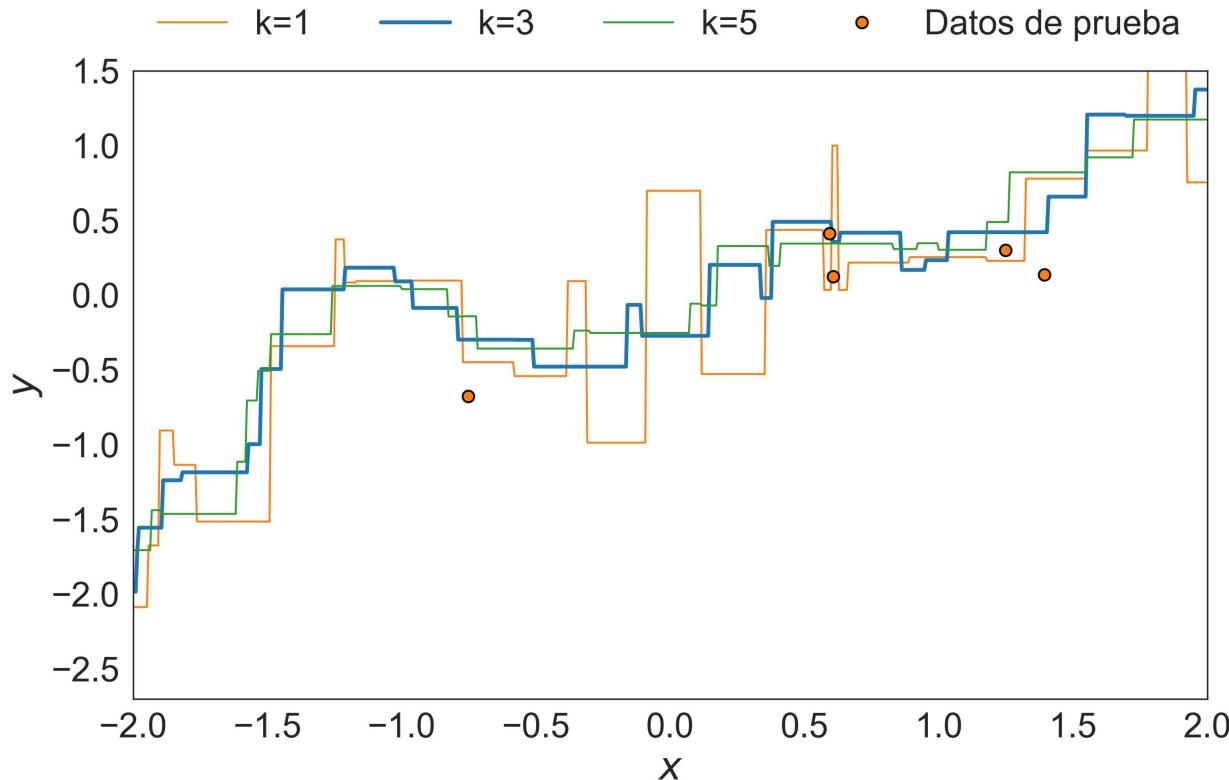
REGRESIÓN kNN

Ajuste de hiperparámetro: ¿cuál es la mejor elección de k ?



REGRESIÓN kNN

Ajuste de hiperparámetro: ¿cuál es la mejor elección de k ?



Funciones de pérdida:

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

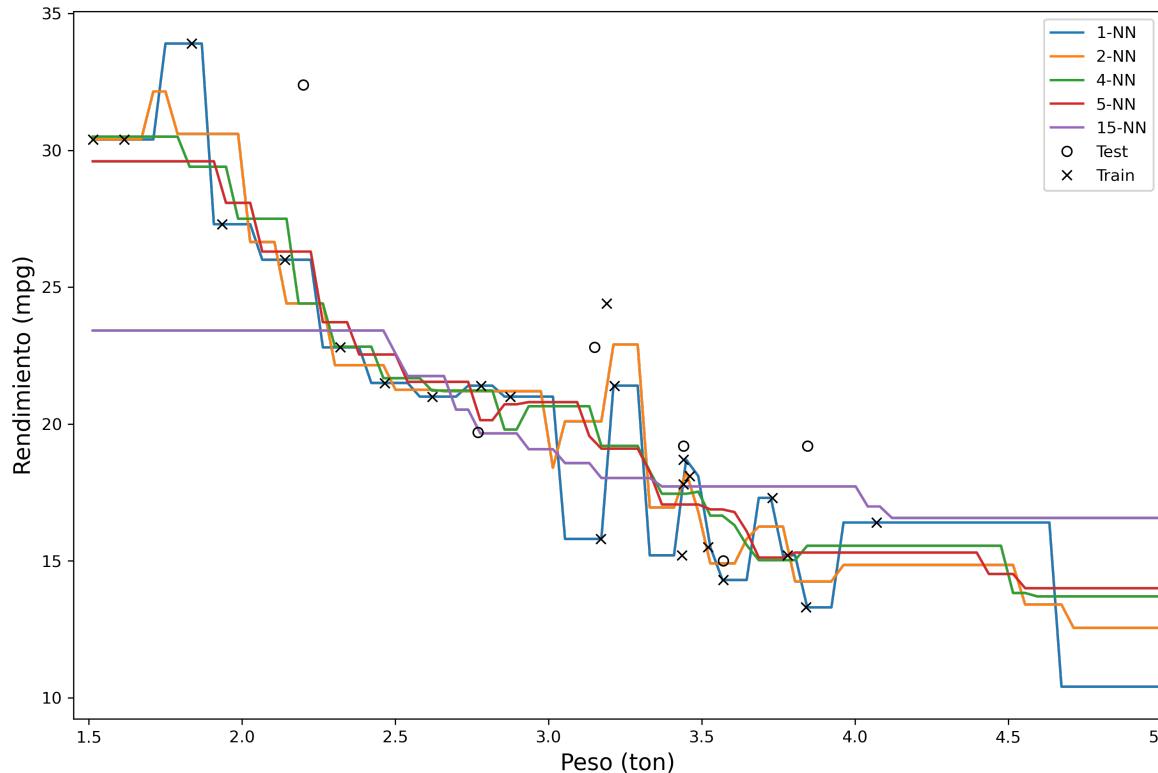
$$\text{RMSE} = \sqrt{\text{MSE}}$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

$\Rightarrow k = 3$

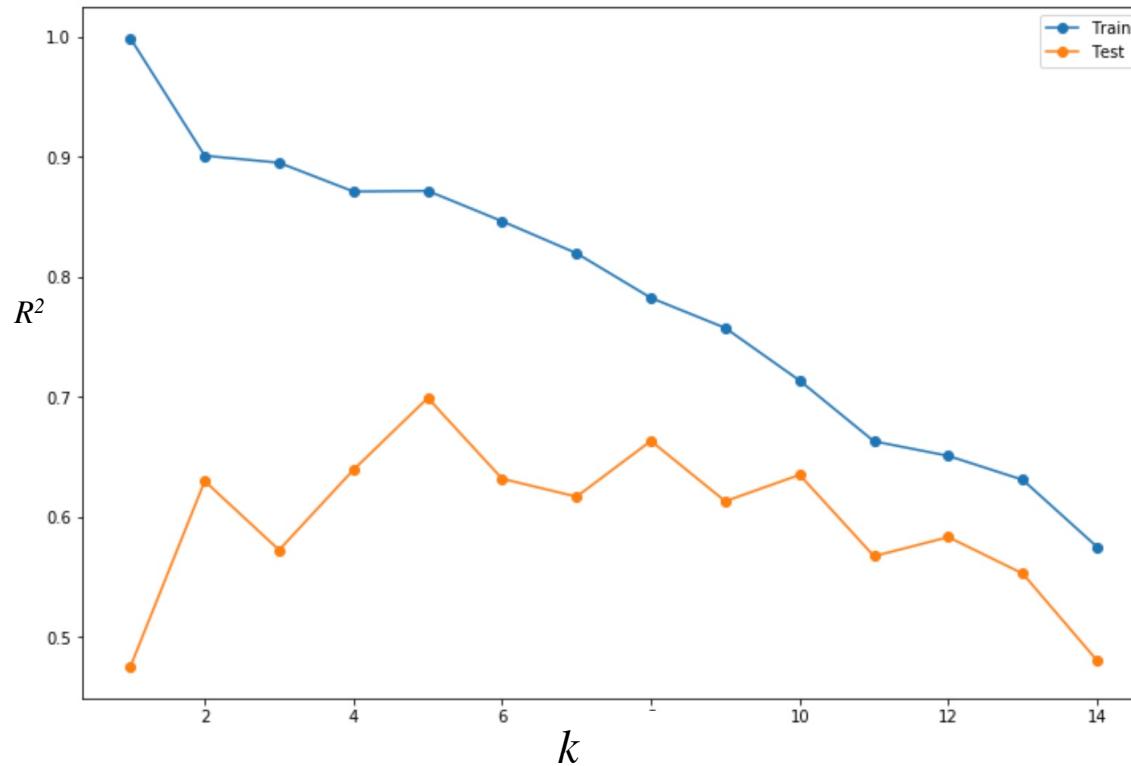
REGRESIÓN kNN

Ejemplo: predicción de rendimiento de vehículos



REGRESIÓN kNN

Ejemplo: predicción de rendimiento de vehículos



RREGRESIÓN kNN: NORMALIZACIÓN

Si hay múltiples predictores: se define una medida de distancia multidimensional para identificar las observaciones más similares o “vecinos”.

- Distancia euclídea: $D(\mathbf{x}_i, \mathbf{x}_0) = \sqrt{\sum_{j=1}^P (x_{i,j} - x_{0,j})^2}$
- Si los predictores tienen diferentes escalas y variabilidad → se introducen efectos de escala en la medición de distancia.
- Por lo tanto, para $p > 1$, es necesario estandarizar los predictores.
- **Normalización z:** se resta la media, y se divide por la desviación estándar.

$$\rightarrow x_{scaled} = \frac{x - \mu}{\sigma}$$

REGRESIÓN LINEAL

- En la regresión kNN, obtenemos predicciones pero no la forma de la función f .
 - ¿qué pasa con y si duplico el valor de x?
- Otra opción, es construir un modelo asumiendo una forma simple para f :

$$Y = f(X) + \epsilon = \beta_0 + \beta_1 X + \epsilon$$

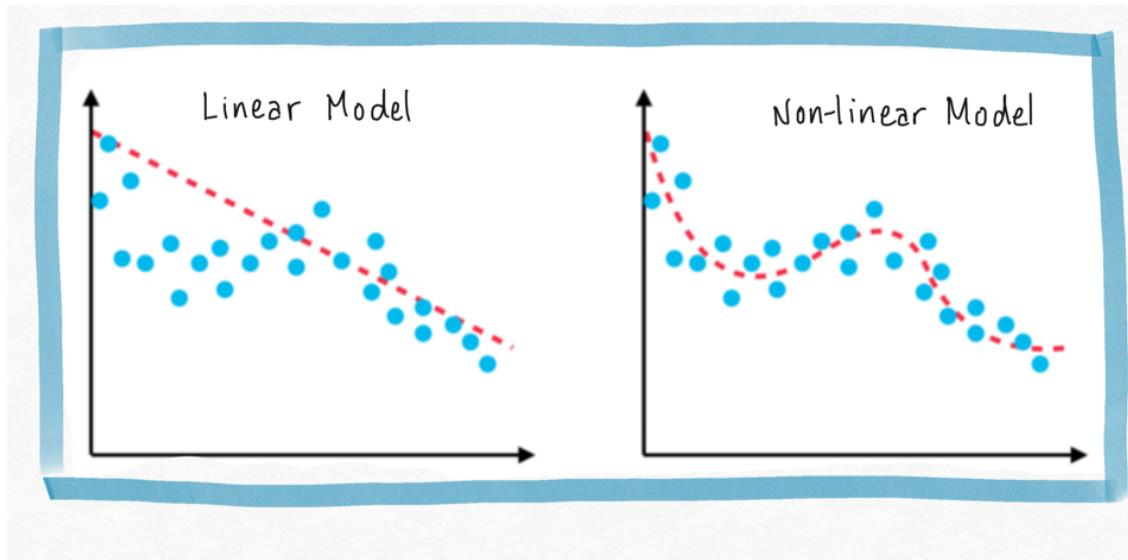
- ¿Cómo estimamos los coeficientes de regresión?

$$\mathcal{L}(\beta_0, \beta_1) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \frac{1}{n} \sum_{i=1}^n (y_i - (\beta_0 + \beta_1 X))^2$$

$$\hat{\beta}_0, \hat{\beta}_1 = \operatorname{argmin} \mathcal{L}(\beta_0, \beta_1) \Rightarrow \begin{aligned} \frac{\partial \mathcal{L}}{\partial \beta_0} &= 0 \\ \frac{\partial \mathcal{L}}{\partial \beta_1} &= 0 \end{aligned}$$

REGRESIÓN POLINOMIAL

- Los modelos lineales pueden entrenarse para conjuntos de datos con muchos predictores. Pero, la relación entre predictores y outcomes no siempre es lineal.
- En este caso, buscamos un modelo de la forma $Y = f_{\beta}(X)$, donde f es una función no-lineal y β es el vector de parámetros de f .



REGRESIÓN POLINOMIAL

- La forma no-lineal más simple que podemos considerar es un polinomio de grado M :

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_M x^M$$

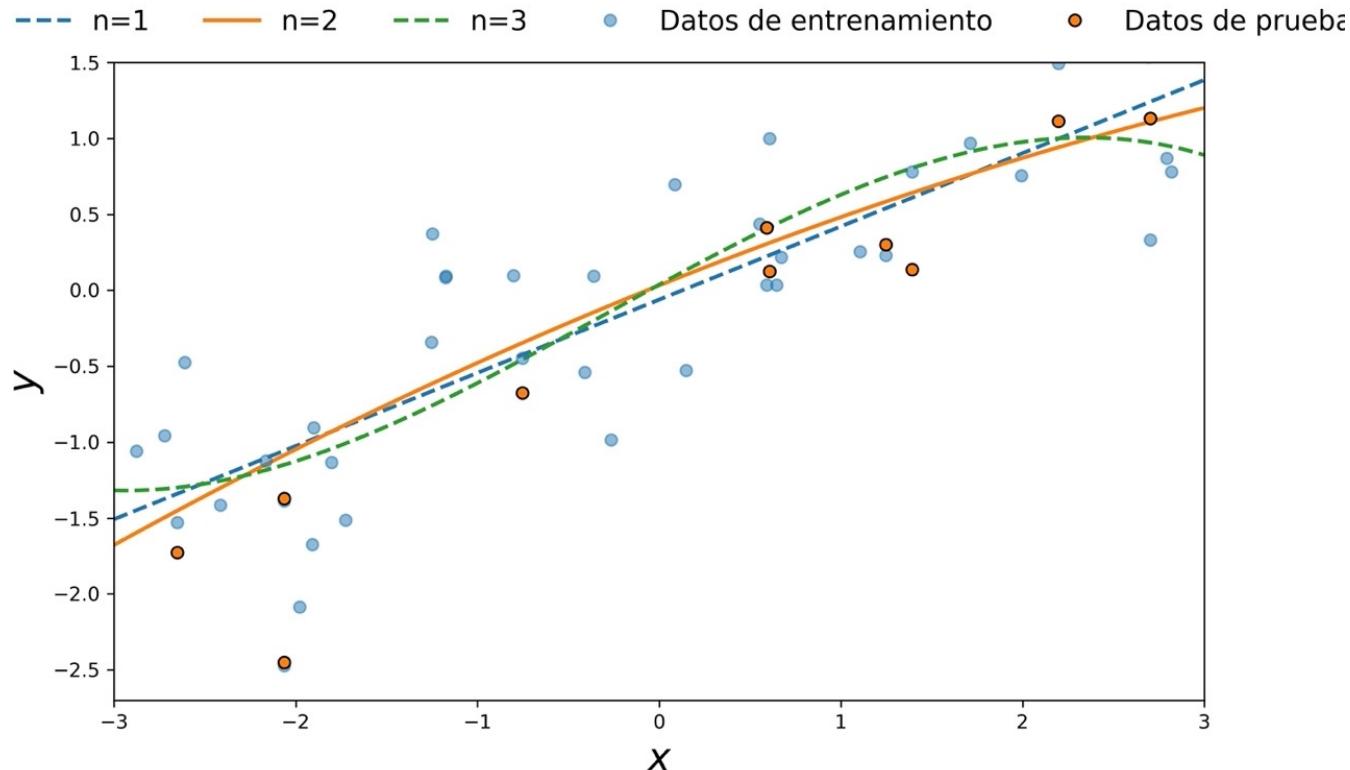
- En este caso, tratamos cada variable x^m como un nuevo predictor, y escribimos el problema como un modelo lineal de la forma:

$$\mathbf{Y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} 1 & x_1^1 & \dots & x_1^M \\ 1 & x_2^1 & \dots & x_2^M \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n^1 & \dots & x_n^M \end{pmatrix}, \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_M \end{pmatrix}.$$

- Una vez construidos los vectores (Y, X) , podemos resolver el problema de la misma forma que una regresión lineal.

$$\Rightarrow \mathbf{Y} = \boldsymbol{\beta} \mathbf{X}$$

REGRESIÓN LINEAL Y POLINOMIAL - EJEMPLO

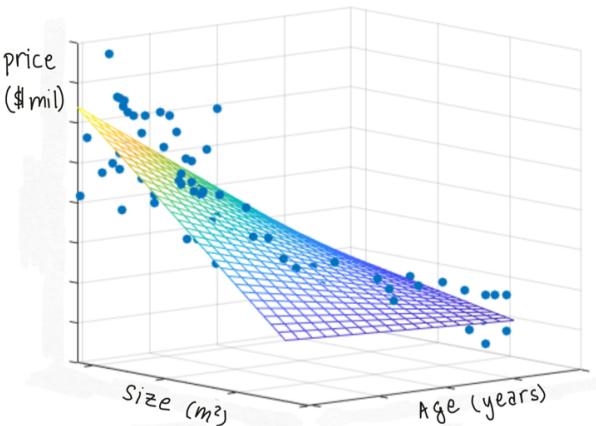


REGRESIÓN MULTILINEAL

- Usualmente, la variable de respuesta Y no depende sólo de una variable predictora, sino de un conjunto de ellas: $f(X_1, \dots, X_J)$.

$$Y = y_1, \dots, y_n, \quad X = X_1, \dots, X_J$$

- Por lo tanto, el modelo a entrenar tiene la forma: $f(X_1, \dots, X_J) = \beta_0 + \beta_1 X_1 + \dots + \beta_J X_J$

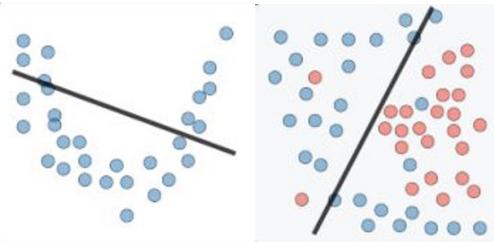


$$\mathbf{Y} = \begin{pmatrix} y_1 \\ \vdots \\ y_y \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} 1 & x_{1,1} & \dots & x_{1,J} \\ 1 & x_{2,1} & \dots & x_{2,J} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n,1} & \dots & x_{n,J} \end{pmatrix}, \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_J \end{pmatrix},$$

$$\Rightarrow \mathbf{Y} = \boldsymbol{\beta} \mathbf{X}$$

APRENDIZAJE SUPERVISADO: OVERFITTING

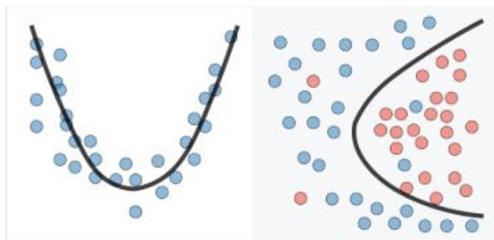
- Si el modelo se ajusta muy cercanamente a la data de entrenamiento, pero falla al generalizar o predecir la data de prueba → **overfitting**



Underfitting

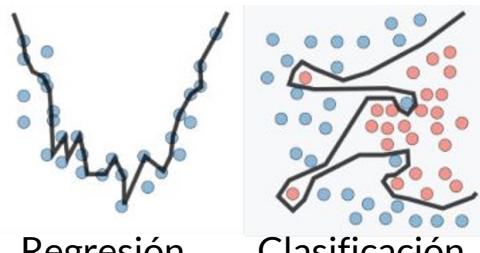
Alto error de entrenamiento

Error de prueba similar a
error de entrenamiento



Óptimo

Error de entrenamiento levemente
más bajos que
error de prueba



Overfitting

Muy bajo error de entrenamiento
Error de prueba mucho mayor a
error de entrenamiento

Regresión

Clasificación

APRENDIZAJE SUPERVISADO: OVERFITTING

- Si el modelo se ajusta muy cercanamente a la data de entrenamiento, pero falla al generalizar o predecir la data de prueba → **overfitting**
- Factores que influyen en overfitting:
 - **Complejidad del modelos (d)**
 - Demasiado simple → underfitting
 - Demasiado complejo → overfitting
 - **Nº de datos de entrenamiento (N)**
 - A mayor cantidad y variedad de datos, más complejo puede ser el modelo sin caer en overfitting
 - **Magnitud del ruido**

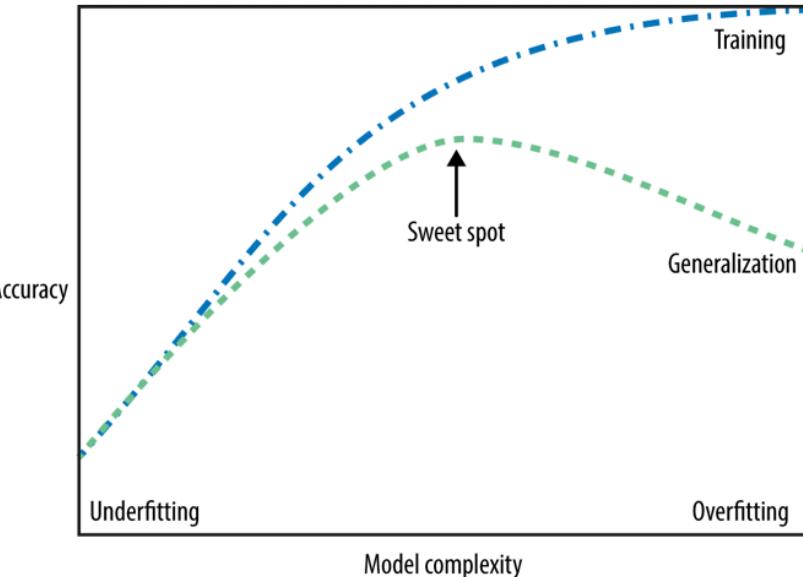
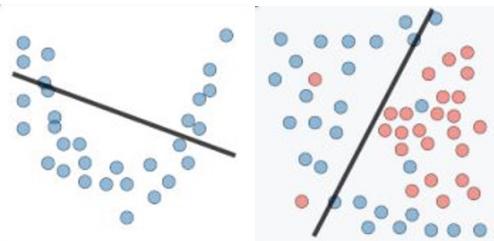


Figure 2-1. Trade-off of model complexity against training and test accuracy

APRENDIZAJE SUPERVISADO: OVERFITTING

- Si el modelo se ajusta muy cercanamente a la data de entrenamiento, pero falla al generalizar o predecir la data de prueba → **overfitting**

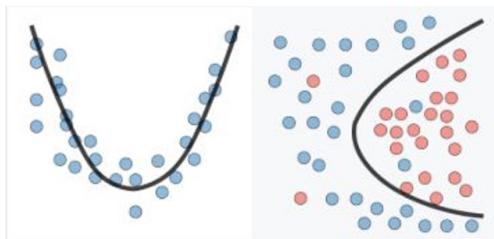


Underfitting

Alto error de entrenamiento
Error de prueba similar a
error de entrenamiento

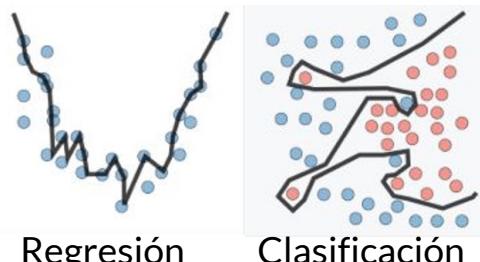


Complejizar modelo
Aregar features



Óptimo

Error de entrenamiento levemente
más bajos que
error de prueba



Overfitting

Muy bajo error de entrenamiento
Error de prueba mucho mayor a
error de entrenamiento



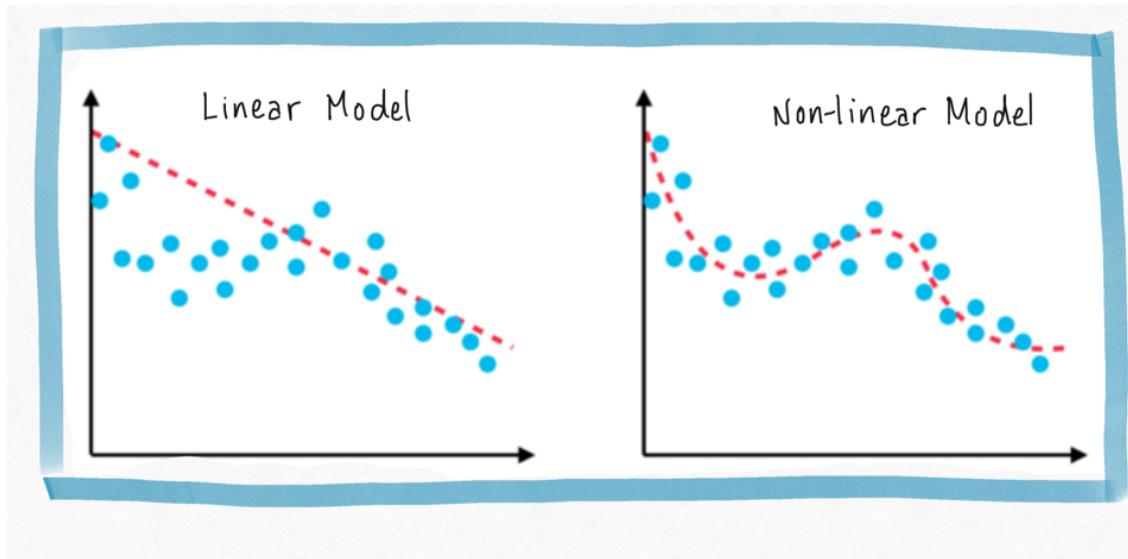
Simplificar modelo
Buscar más datos
Regularización

Regresión

Clasificación

REGRESIÓN POLINOMIAL

- Los modelos lineales pueden entrenarse para conjuntos de datos con muchos predictores. Pero, la relación entre predictores y outcomes no siempre es lineal.
- En este caso, buscamos un modelo de la forma $Y = f_{\beta}(X)$, donde f es una función no-lineal y β es el vector de parámetros de f .



REGRESIÓN POLINOMIAL

- La forma no-lineal más simple que podemos considerar es un polinomio de grado M :

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_M x^M$$

- En este caso, tratamos cada variable x^m como un nuevo predictor, y escribimos el problema como un modelo lineal de la forma:

$$\mathbf{Y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} 1 & x_1^1 & \dots & x_1^M \\ 1 & x_2^1 & \dots & x_2^M \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n^1 & \dots & x_n^M \end{pmatrix}, \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_M \end{pmatrix}.$$

- Una vez construidos los vectores (\mathbf{Y}, \mathbf{X}) , podemos resolver el problema de la misma forma que una regresión lineal.

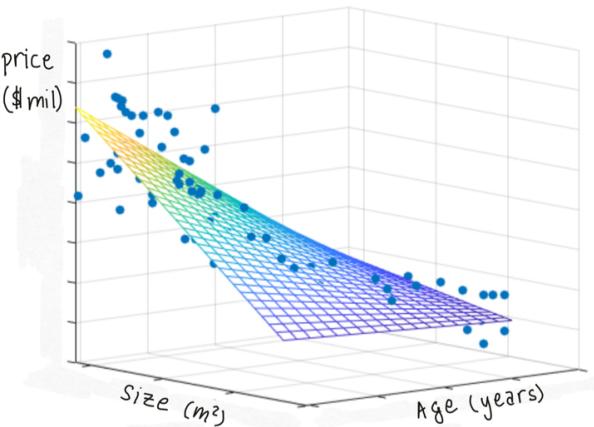
$$\Rightarrow \mathbf{Y} = \boldsymbol{\beta} \mathbf{X}$$

REGRESIÓN MULTILINEAL

- Usualmente, la variable de respuesta Y no depende sólo de una variable predictora, sino de un conjunto de ellas: $f(X_1, \dots, X_J)$.

$$Y = y_1, \dots, y_n, \quad X = X_1, \dots, X_J$$

- Por lo tanto, el modelo a entrenar tiene la forma: $f(X_1, \dots, X_J) = \beta_0 + \beta_1 X_1 + \dots + \beta_J X_J$

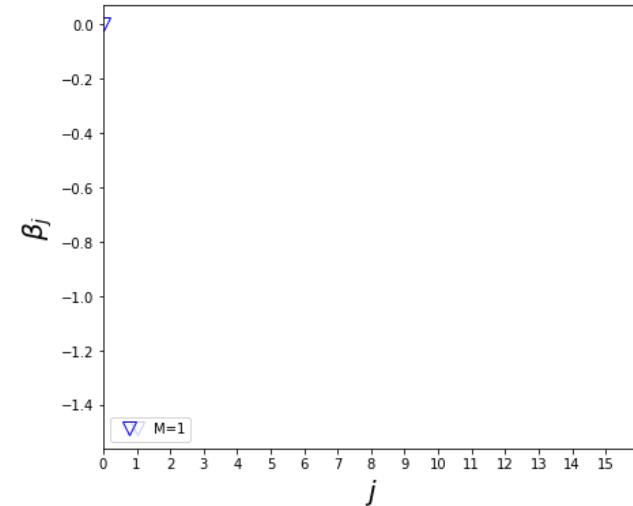
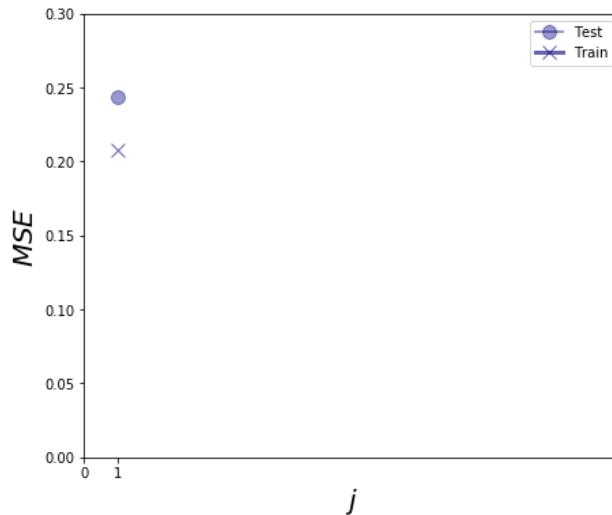
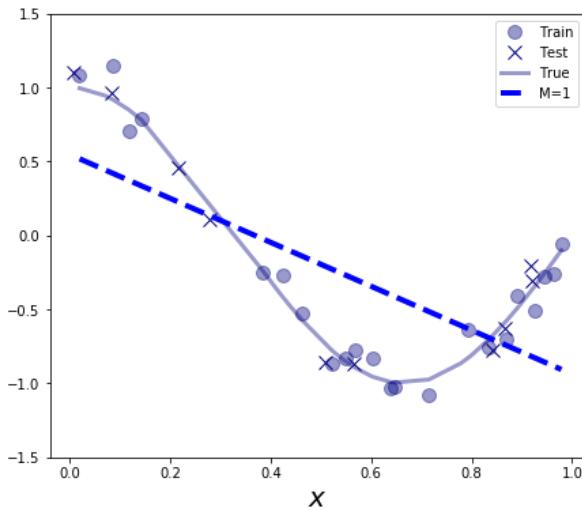


$$\mathbf{Y} = \begin{pmatrix} y_1 \\ \vdots \\ y_y \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} 1 & x_{1,1} & \dots & x_{1,J} \\ 1 & x_{2,1} & \dots & x_{2,J} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n,1} & \dots & x_{n,J} \end{pmatrix}, \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_J \end{pmatrix},$$

$$\Rightarrow \mathbf{Y} = \boldsymbol{\beta} \mathbf{X}$$

REGRESIÓN Y OVERFITTING

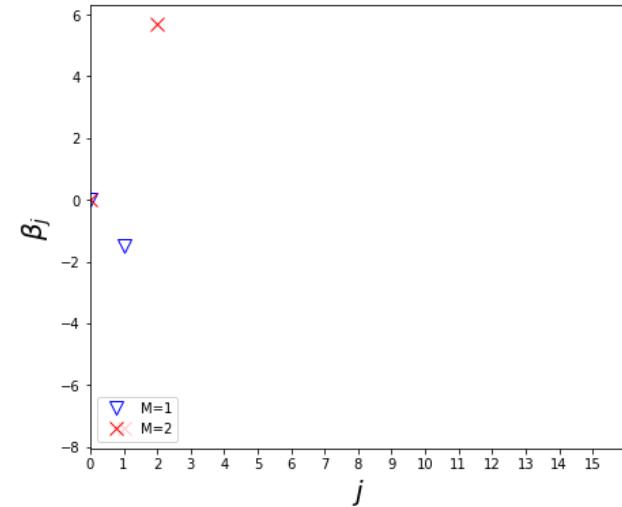
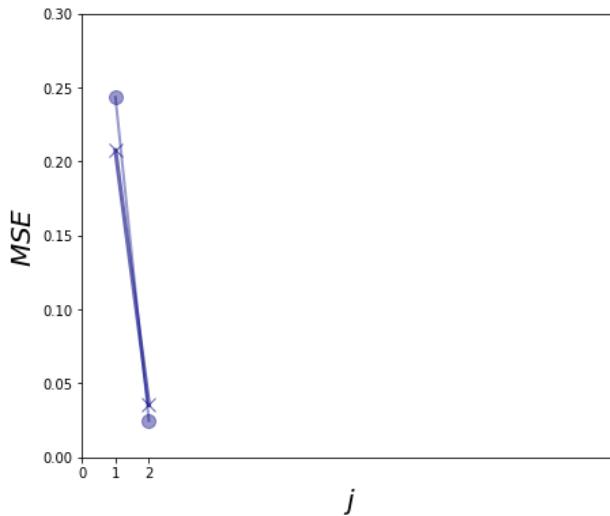
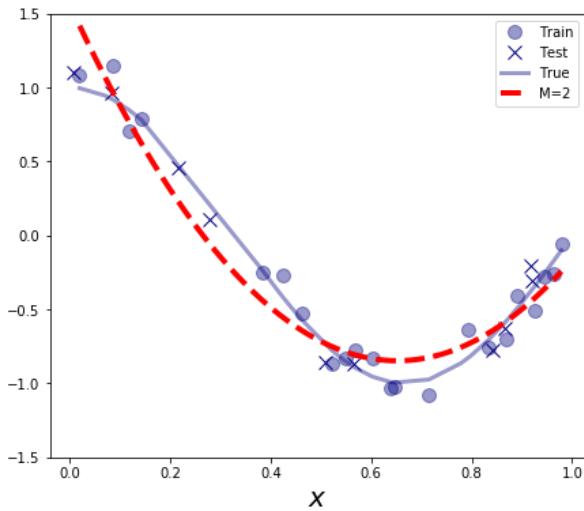
- Supongamos un modelo de regresión polinomial para un conjunto de $n=30$ datos



Ajuste lineal → underfitting

REGRESIÓN Y OVERFITTING

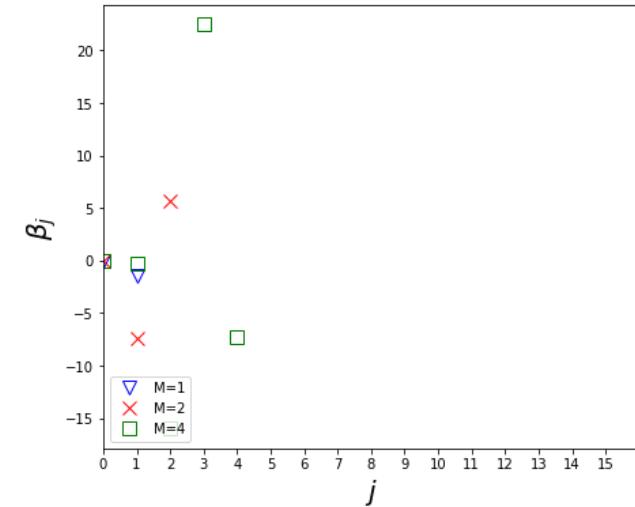
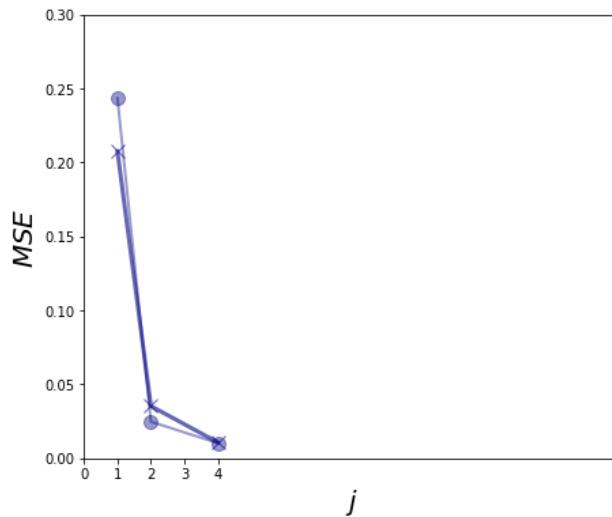
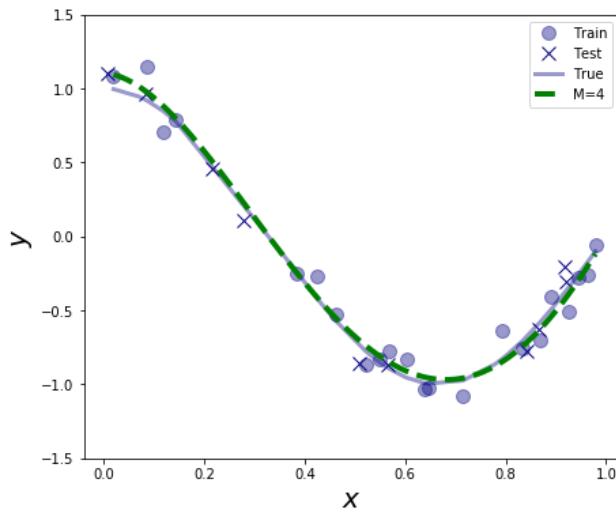
- Supongamos un modelo de regresión polinomial para un conjunto de $n=30$ datos



Disminuye el error 😊

REGRESIÓN Y OVERFITTING

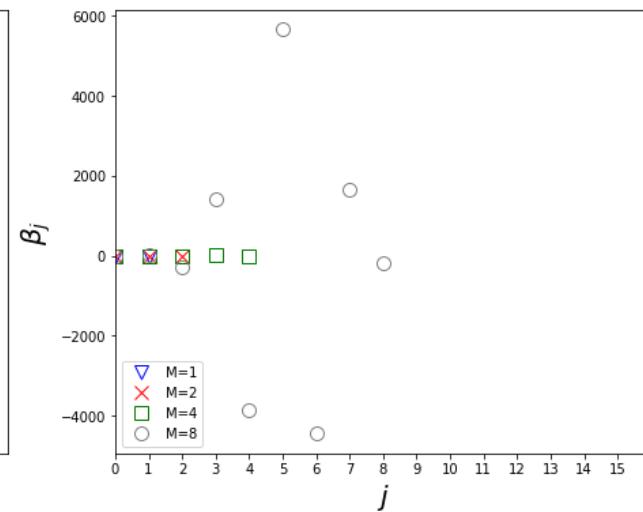
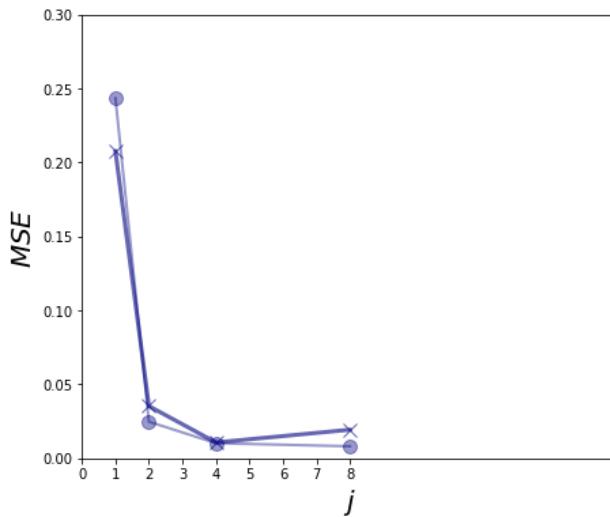
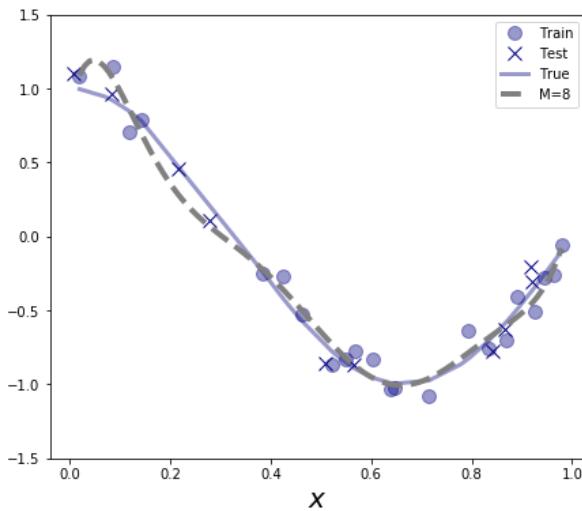
- Supongamos un modelo de regresión polinomial para un conjunto de $n=30$ datos



Disminuye el error 😊

REGRESIÓN Y OVERFITTING

- Supongamos un modelo de regresión polinomial para un conjunto de $n=30$ datos

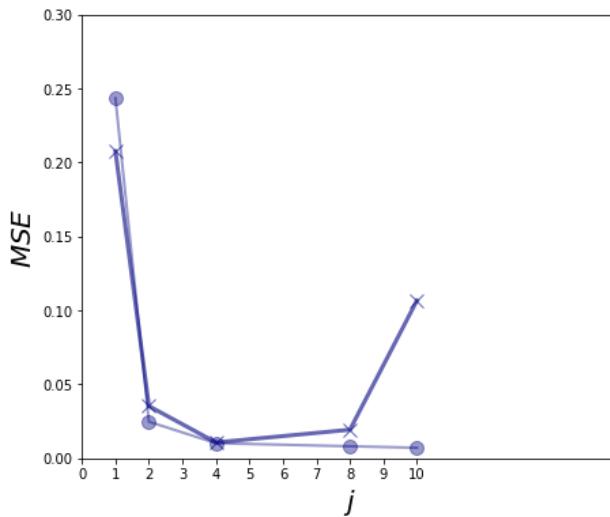
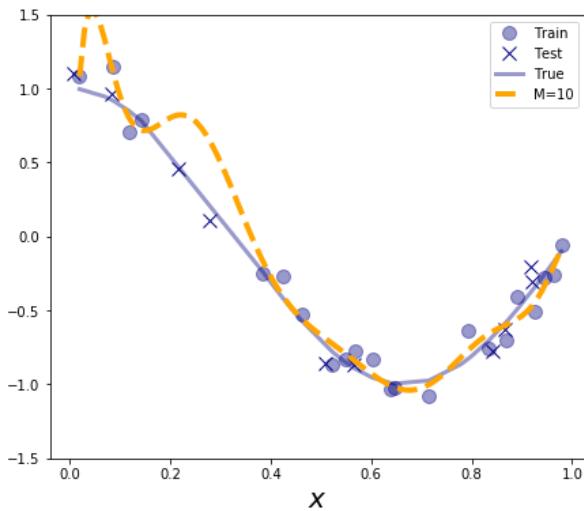


Aumenta el error de validación → comienza el overfitting... (M=8) 😞

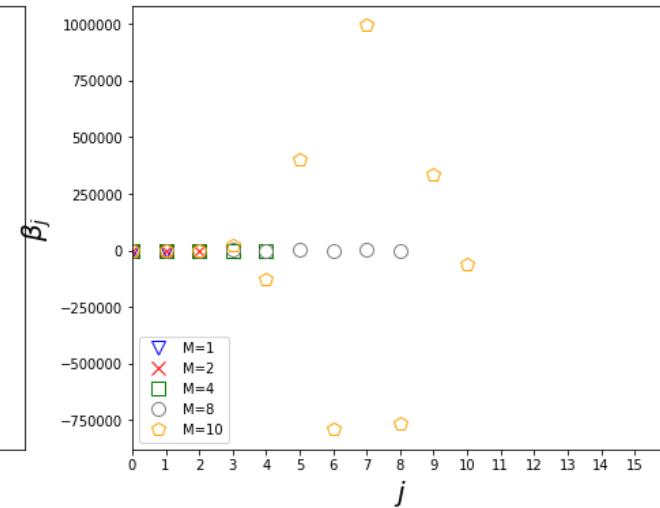
Valores de los coeficientes se hacen extremos 😞

REGRESIÓN Y OVERFITTING

- Supongamos un modelo de regresión polinomial para un conjunto de $n=30$ datos



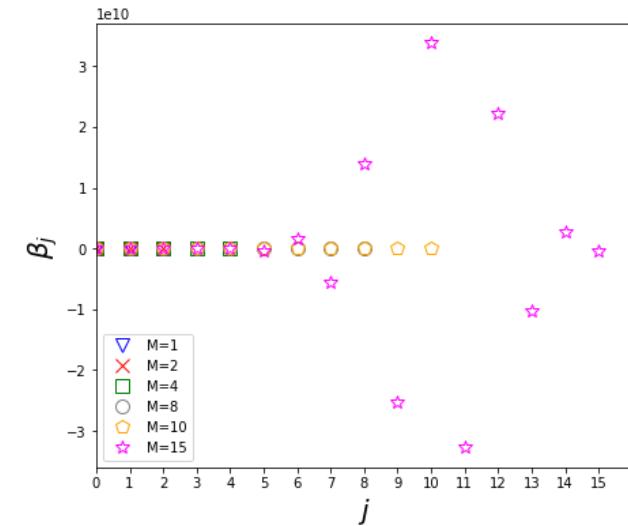
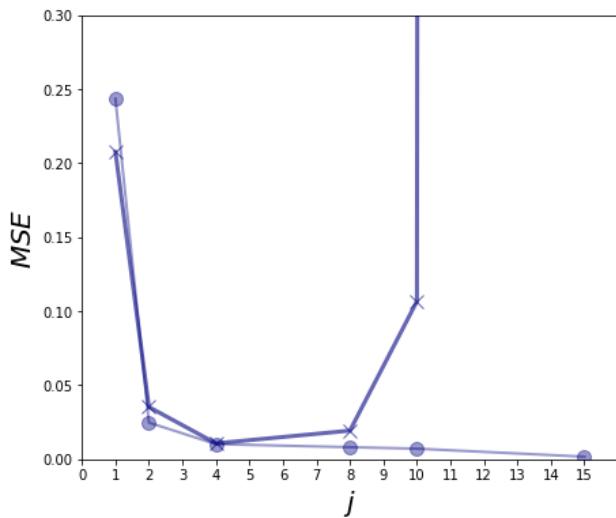
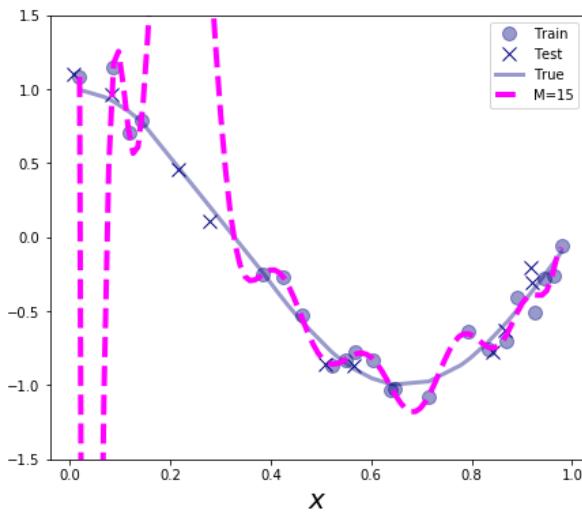
Overfitting!! 😥 😥



Valores de los
coeficientes se hacen
más extremos 😥

REGRESIÓN Y OVERFITTING

- Supongamos un modelo de regresión polinomial para un conjunto de $n=30$ datos



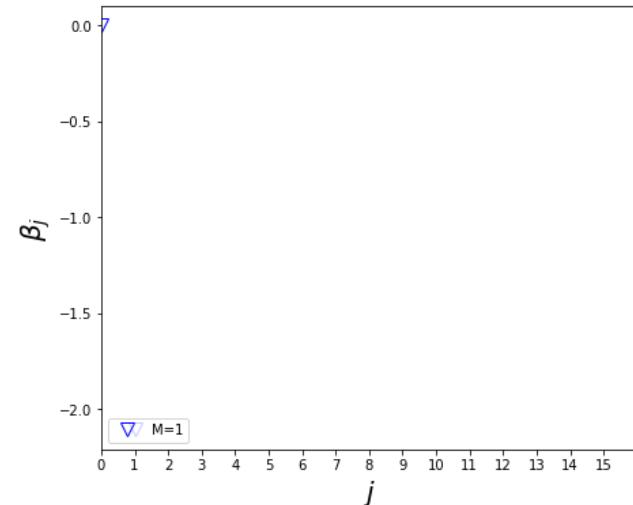
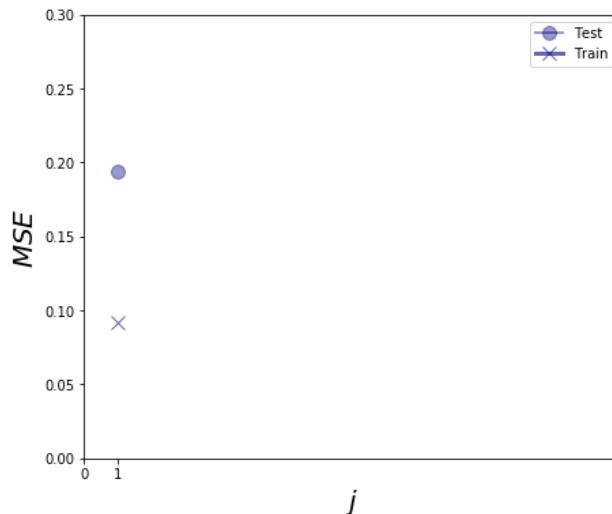
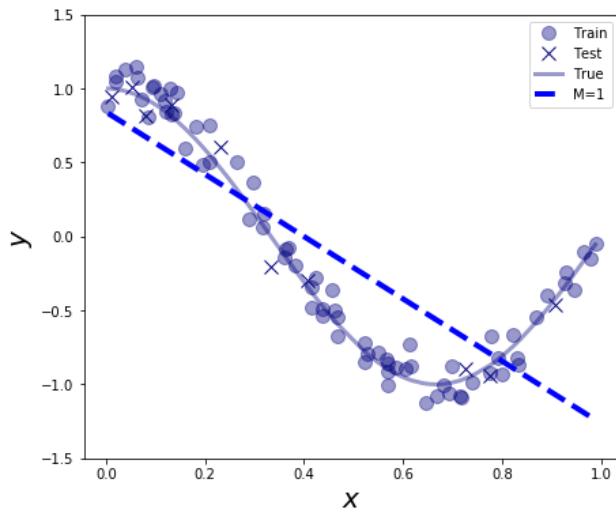
Mega overfitting!! 😞 😞 😞

Valores de los
coeficientes se hacen
muy extremos 😞

¿Qué pasa si agregamos más datos?

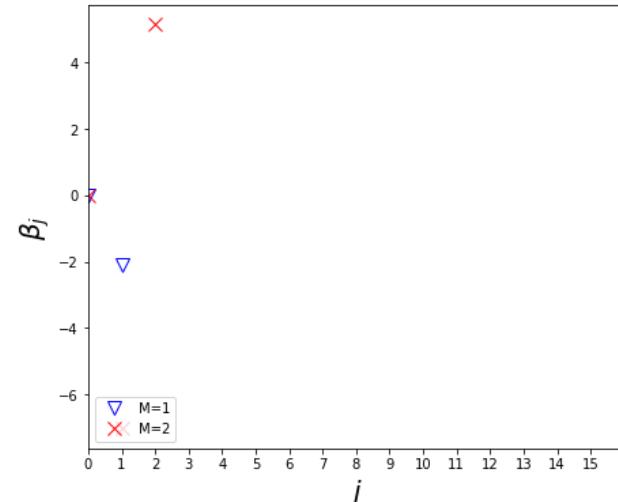
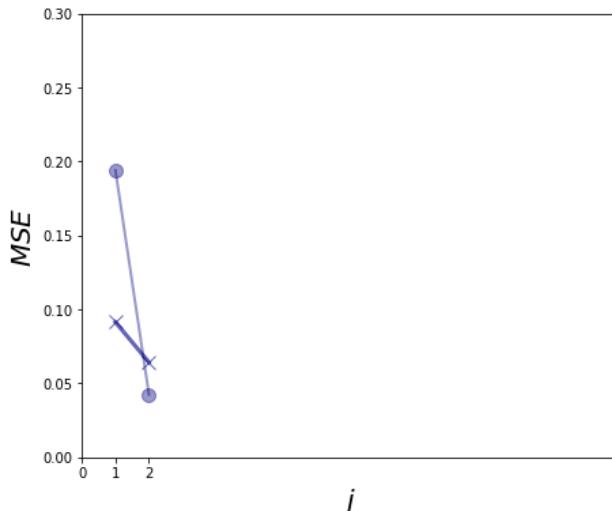
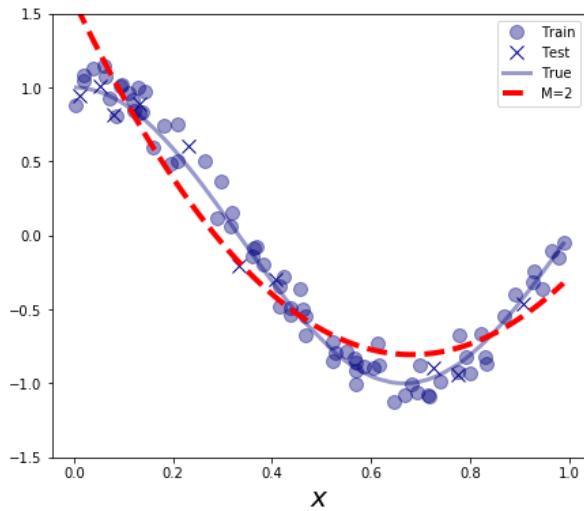
REGRESIÓN Y OVERFITTING

- Supongamos un modelo de regresión polinomial para un conjunto de **n=100 datos**



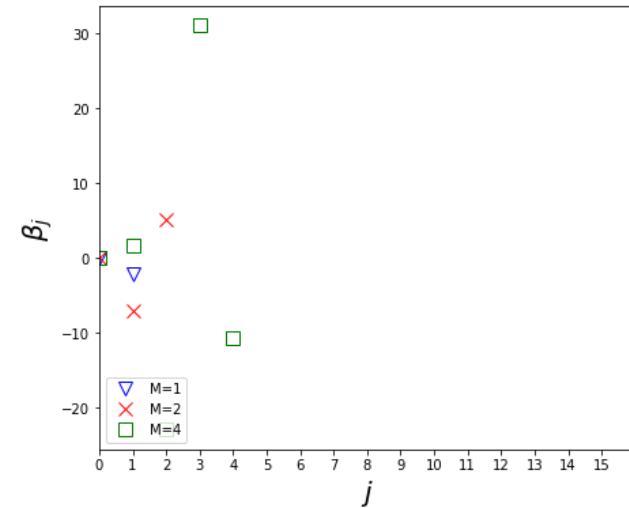
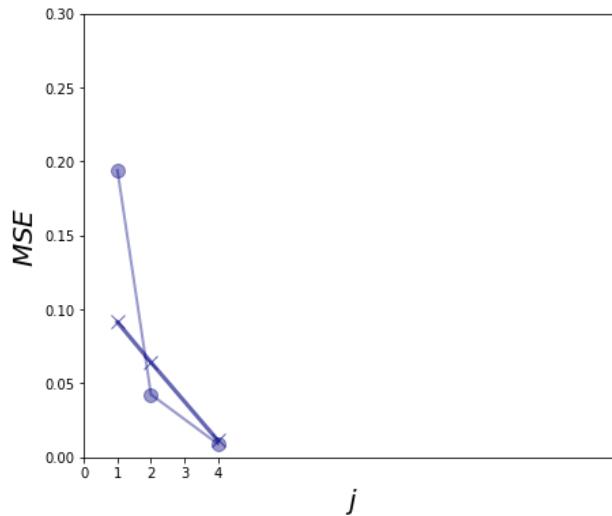
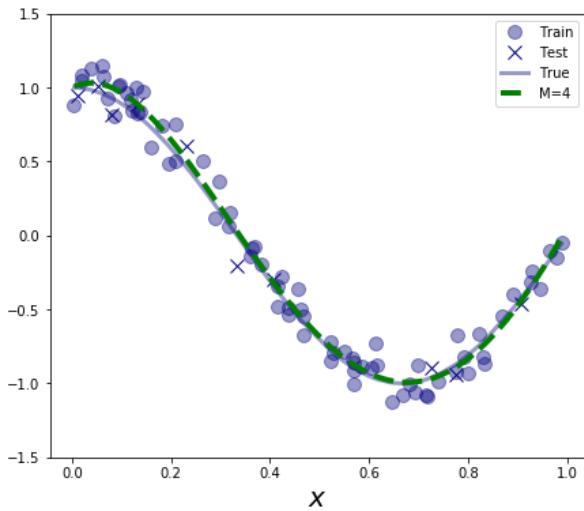
REGRESIÓN Y OVERFITTING

- Supongamos un modelo de regresión polinomial para un conjunto de **n=100 datos**



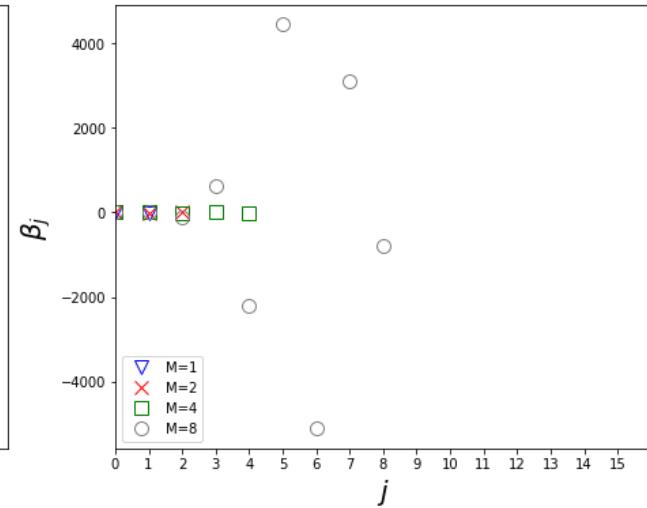
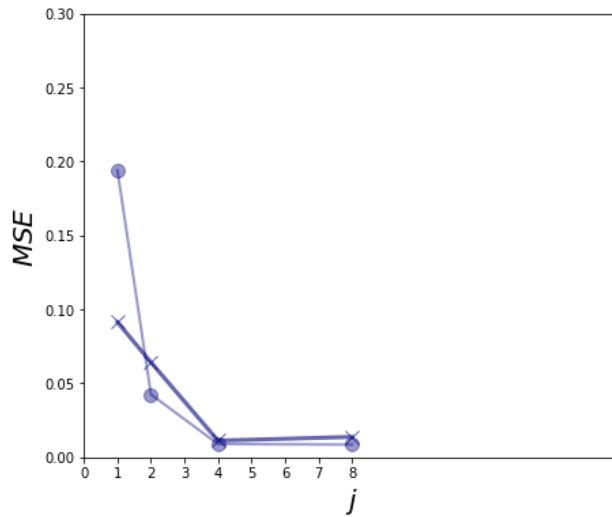
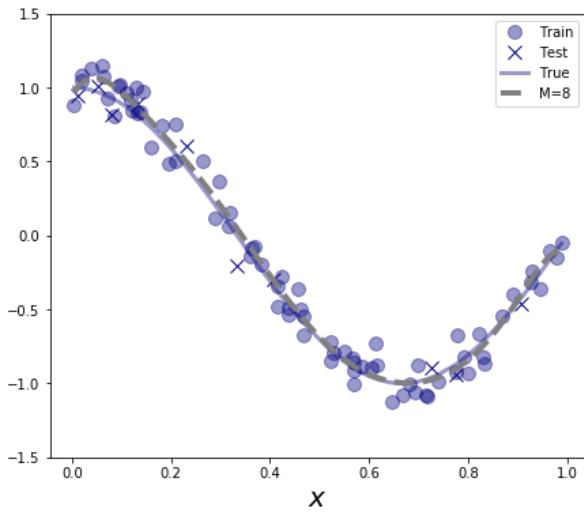
REGRESIÓN Y OVERFITTING

- Supongamos un modelo de regresión polinomial para un conjunto de **n=100 datos**



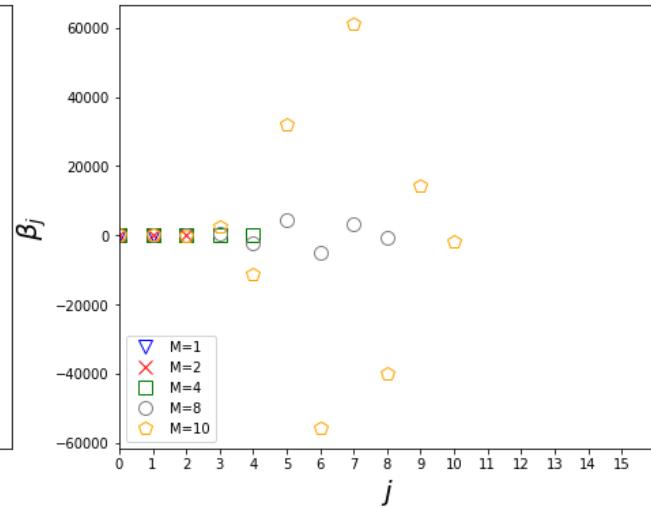
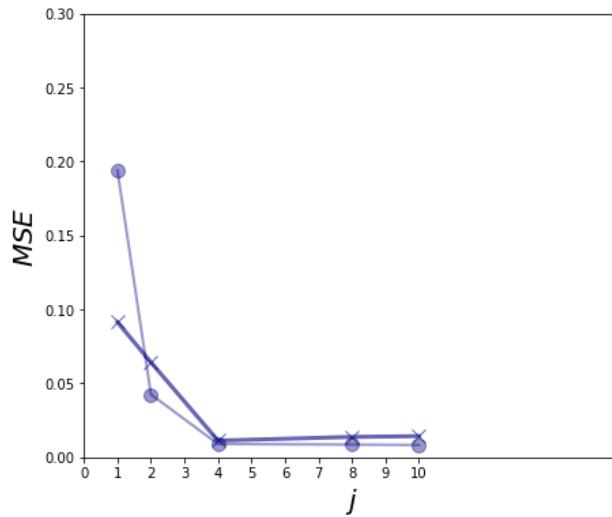
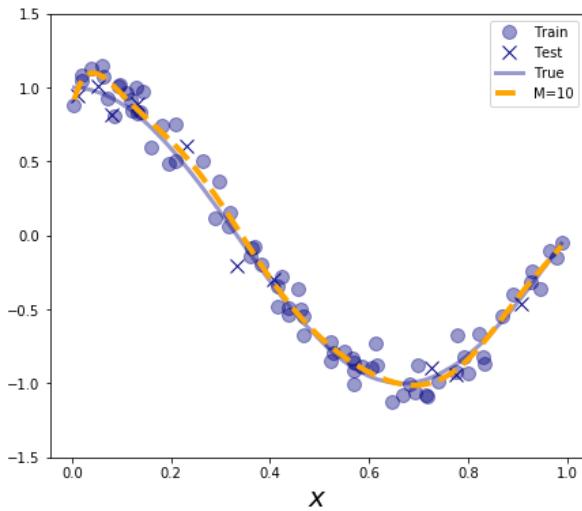
REGRESIÓN Y OVERFITTING

- Supongamos un modelo de regresión polinomial para un conjunto de **n=100 datos**



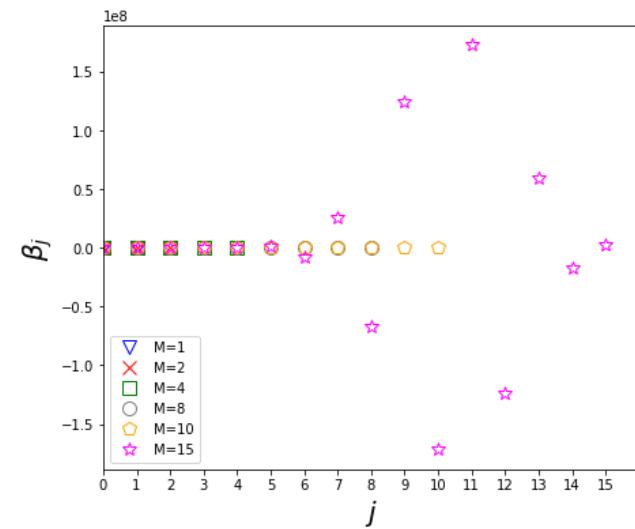
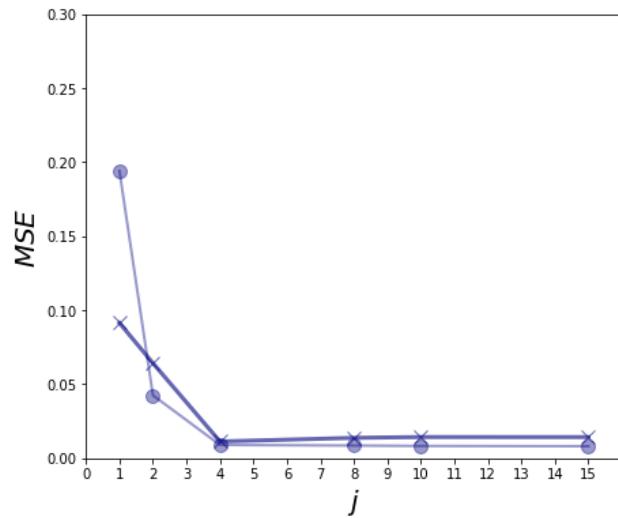
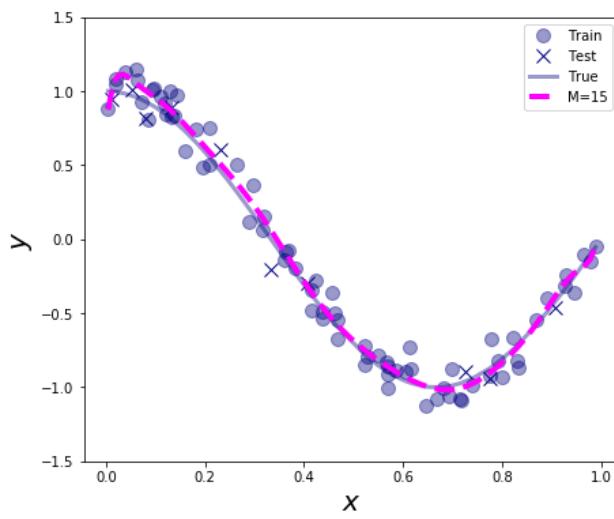
REGRESIÓN Y OVERFITTING

- Supongamos un modelo de regresión polinomial para un conjunto de **n=100 datos**



REGRESIÓN Y OVERFITTING

- Supongamos un modelo de regresión polinomial para un conjunto de **n=100 datos**



→ Con más datos, puedo entrenar un modelo más complejo, con menor riesgo de caer en overfitting.