

# Elements of Machine Learning

## Introducción

MSc. Diego Porres



Enero 2019

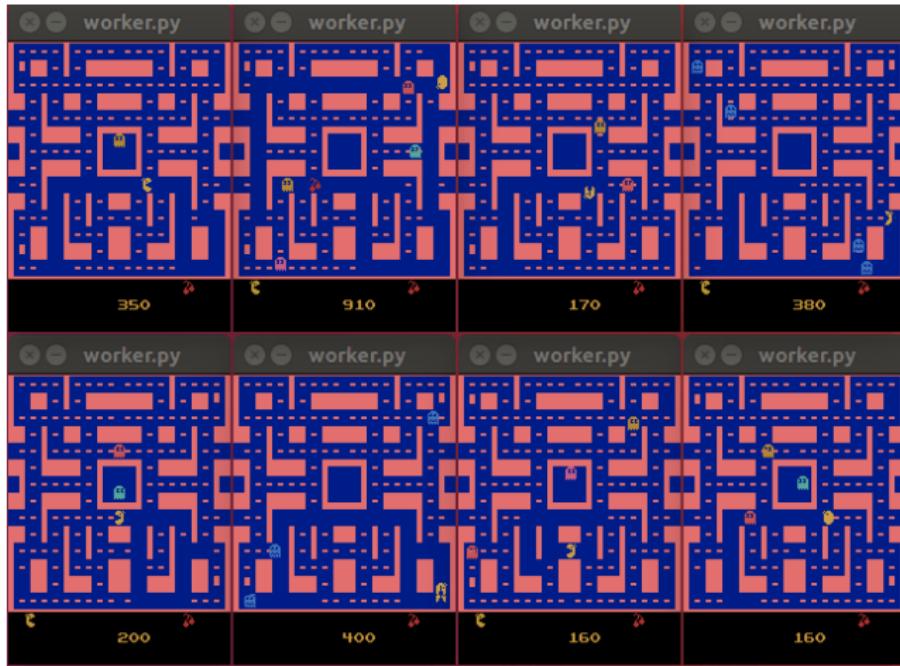
Some of the figures in this presentation are taken from *An Introduction to Statistical Learning, with applications in R* (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani

# Introducción

## ■ Acerca de mí:

- Licenciatura en Física, UVG, Guatemala
- MSc. in Mathematical Engineering, Universidad de L'Aquila, Italia
- Master of Mathematics and Interactions, Universidad Nice-Sophia Antipolis, Francia
  - ▶ Pasantía en el Centre de Visió per Computador, Barcelona, España
  - ▶ [Deep] Reinforcement Learning for Self-Driving Cars
- Aplicaciones de drones a la agricultura de precisión.
- ¿?

# Pasantía en el CVC - MsPacman



# Pasantía en el CVC - DuskDrive



# NeurIPS 2018: *Latent Fabrics*



<http://www.aiartonline.com/community/diego-porres/>

# Objetivos del Curso

- Curso introductorio al *Aprendizaje Estadístico (SL)* y *Aprendizaje de Máquinas (ML)*.
- Específicamente, buscamos ahondar en el rigor matemático necesario para la realización de investigación en éstas disciplinas.
- Queremos que el alumno no vea los distintos algoritmos utilizados en SL y ML como *cajas negras*.
- Buscamos que los alumnos tengan una buena base de Cálculo, Probabilidad, Estadística y Álgebra lineal.

# Evaluación

Procedimiento	Ponderación
Laboratorios (GitHub)	30%
Tareas	15%
Portafolio de Algoritmos (GitHub)	20%
Publicación de artículo	15%
Proyecto Final (Paper)	20%

- No se aceptará tarde la entrega de tareas ni se repondrán trabajos con ponderación (salvo con excusa).
- Para aprobar el curso, es necesario realizar la publicación de un artículo o entrada de blog en cualquier medio.
- El proyecto final consistirá en la explicación e implementación de algún algoritmo avanzado en datos obtenidos por el estudiante (véase Kaggle).

# Laboratorios

- Lab. 1: Intro. a Python  
(Jupyter, Colab, etc.)
- Lab. 2: Regresión lineal
- Lab. 3: Regresión Logística  
(LDA, KNN, QDA)
- Lab. 4: Validación cruzada y  
Bootstrap
- Lab. 5: Árboles de Decisión
- Lab. 6: Redes Neuronales (NN)
- Lab. 7: Máquinas de Vectores  
de soporte (SVMs)
- Lab. 8: Clasificadores de  
k-vecinos cercanos
- Lab. 9: Clustering
- Lab. 10: Análisis de  
Componentes Principales (PCA)

# Bibliografía de Referencia

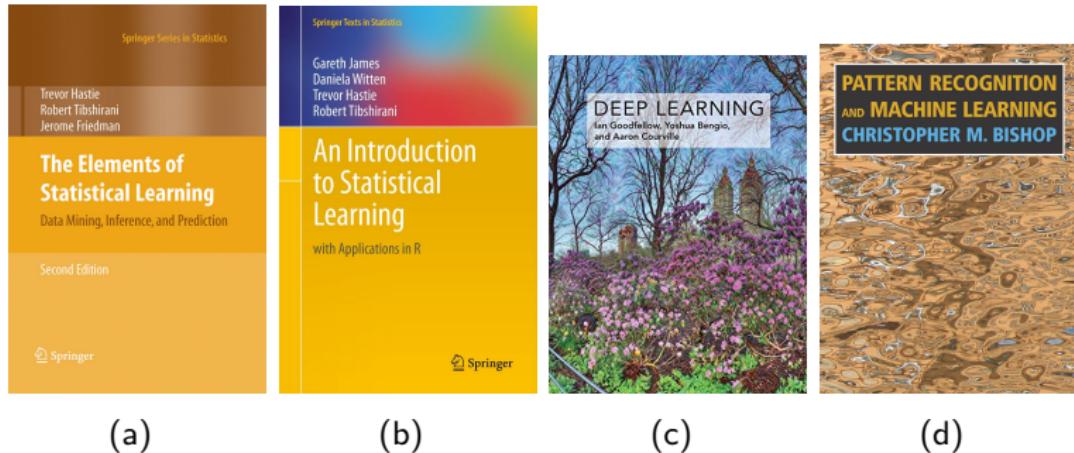


Figura 1: (a) Elements of Statistical Learning (b) Introduction to Statistical Learning (c) Deep Learning (d) Pattern Recognition and Machine Learning

# Machine Learning vs. Statistical Learning

- *Machine Learning (ML)* surgió como un subcampo de la Inteligencia Artificial.
- *Statistical Learning (SL)* surgió como un subcampo de la Estadística.
- **Existe mucho traslapo**, ya que ambos campos se concentran en problemas Supervisados y No Supervisados:
  - ML se concentra más en *aplicaciones a gran escala* y en la *precisión de la predicción*.
  - SL se concentra más en los *modelos*, su interpretabilidad, en la *precisión e incertidumbre*.
- **La distinción es cada vez más borrosa.**

# Introducción: Una Visión General de SL

- *Aprendizaje estadístico* se refiere a una serie de herramientas para entender datos.
- Podemos clasificar a dichas herramientas como **supervisadas** y **no supervisadas**.
  - Tercera area: *aprendizaje por refuerzo (RL)*, el cual no veremos.

# Aprendizaje Supervisado (SL)

- En el *aprendizaje supervisado (SL)*, construimos un modelo estadístico para estimar un resultado medible  $Y$  a partir de mediciones  $X$ .
- $Y$  también es llamado la variable dependiente, *variable de respuesta*, *variable de salida* o variable objetivo.
- $X$  es un vector de  $p$  *mediciones predictoras*, también llamadas *variable de entrada*, regresores, covariables, características o variables independientes.
- El objetivo de SL entonces es de predecir el valor de  $Y$  dado  $X$ , con (bastantes) ejemplos o instancias de estas medidas:

$$\{(x_1, y_1), \dots, (x_n, y_n)\}$$

# SL - Tipos de variables

- Por lo tanto,  $Y$  puede ser:
  - **Discreta** (cualitativa o categórica)
  - **Continua** (cuantitativa)
  - **Ordenada categórica** (el orden es importante)
- Si estamos prediciendo una variable discreta, nos referimos al problema como de *clasificación*.
- Si estamos prediciendo una variable continua, nos referimos al problema como de *regresión*.

# Aprendizaje No Supervisado (UL)

- En el *aprendizaje no supervisado (UL)*, no tenemos variable de salida, solamente un conjunto de predictores (características) medidos en un conjunto de muestras.
- El objetivo final está más difuso y depende de la aplicación: encontrar subgrupos (clusters), subconjunto de predictores que se comporten de igual manera, o bien una combinación lineal de éstos que tengan la mayor variación.
- Puede ser útil para preprocesamiento de datos a utilizar en SL.
- No hay una métrica *per se*, por lo que es difícil medir qué tan bien nos va.

# Notación (1)

- $n$  representará el número de puntos de datos distintos u observaciones en nuestra muestra.
- $p$  denotará el número de variables que tenemos para realizar predicciones.
  - En algunos casos,  $p > n$  e inclusive  $p \gg n$ , lo cual es común para muestras biológicas.
- Sea  $x_{ij}$  el valor de la variable  $j$  para la observación  $i$ , donde  $i = 1, \dots, n$  y  $j = 1, \dots, p$ . Esto quiere decir que podemos ordenar a nuestros datos en una forma matricial:

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{pmatrix}$$

## Notación (2)

- Denotaremos a  $x_i$  como el  $i$ -ésimo valor observado de  $X$ , donde  $x_i$  es un vector de longitud  $p$ , el cual contiene las mediciones de las  $p$  variables para la observación  $i$ :

$$x_i = \begin{pmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{ip} \end{pmatrix}$$

- Las columnas de  $\mathbf{X}$  serán denotadas por  $\mathbf{x}_j$ , un vector de longitud  $n$  que contiene los valores de la variable  $j$  en los  $n$  datos:

$$\mathbf{x}_j = \begin{pmatrix} x_{1j} \\ x_{2j} \\ \vdots \\ x_{nj} \end{pmatrix}$$

## Notación (3)

- Por lo tanto, podemos escribir a  $\mathbf{X}$  de las siguientes maneras:

$$\mathbf{X} = \begin{pmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_n^T \end{pmatrix} \quad \text{ó} \quad \mathbf{X} = (x_1 \ x_2 \ \dots \ x_p)$$

- $X$  denotará una *variable de entrada*.
- $Y$  denotará una *variable de salida cuantitativa* (continua).
- $G$  denotará una *variable de salida cualitativa* (discreta).
- Utilizaremos a  $x_i$ ,  $y_i$  y  $g_i$ ,  $i = 1, \dots, n$  como los  $i$ -ésimas instancias de  $X$ ,  $Y$  y  $G$ , respectivamente.

## Notación (4)

- $\mathcal{G}$  denotará el conjunto que contiene todas las clases que  $G$  puede tomar.
- Denotaremos a  $\hat{Y}$  como la *predicción de salida* para un  $X$  dado.
- Se presume que se tiene una *serie de datos de entrenamiento etiquetados* para los problemas de regresión:

$$\mathcal{T} = \{(x_1, y_1), \dots, (x_n, y_n)\}$$

donde  $x_i \in \mathbb{R}^p$  y  $y_i \in \mathbb{R}$ .

- Se presume que se tiene una *serie de datos de entrenamiento etiquetados* para los problemas de clasificación:

$$\mathcal{T} = \{(x_1, g_1), \dots, (x_n, g_n)\}$$

donde  $x_i \in \mathbb{R}^p$  y  $g_i \in \mathcal{G}$ .

## Ejemplo - SL

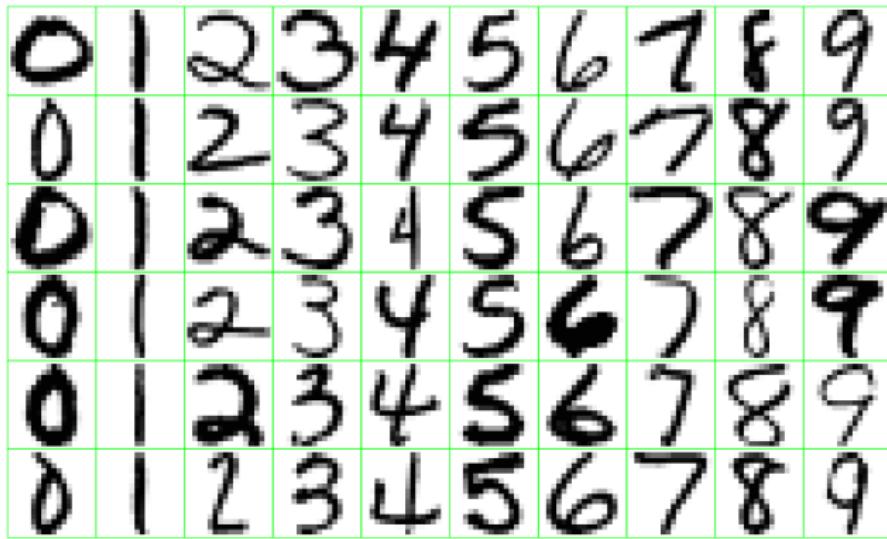


Figura 2: Ejemplos de imágenes de dígitos escritos a mano de sobres postales de los EE.UU. Cada imagen es de  $16 \times 16$  pixeles de ocho bits, y cada pixel tiene una intensidad de 0 a 255. Por lo tanto,  $\mathcal{G} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$  y queremos clasificar correctamente cada imagen.

# Ejemplo - UL

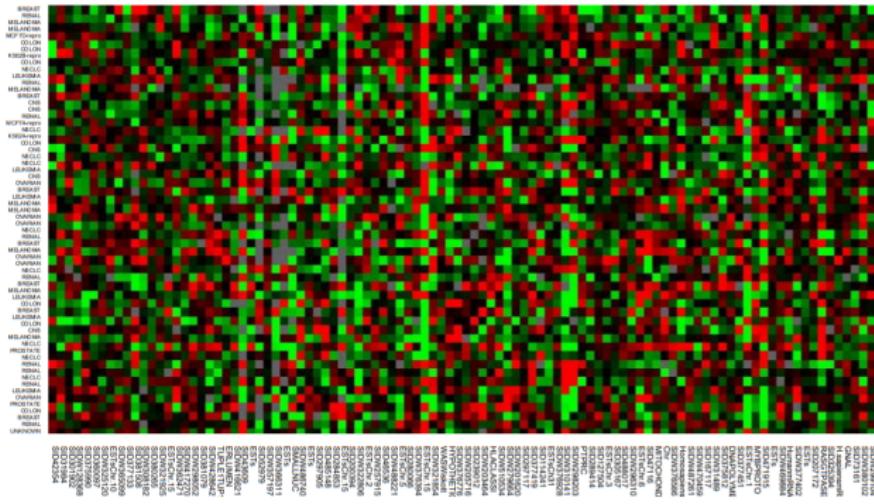


Figura 3: Datos de ADN: matriz de expresión de  $p = 6830$  genes (columnas) y  $n = 64$  muestras (filas) de tumores de distintos pacientes. Se muestran 100 columnas únicamente (de manera aleatoria). Los valores van de verde claro (negativo) a rojo (positivo). Estamos interesados en si hay grupos o *clusters*.

# Una motivación (1)

- Suponga que es lo contratan para analizar los datos de publicidad de una empresa.
- La empresa tiene datos de cuánto gasta en tres distintos medios TV, radio y periódico, así como las ventas =  $Y$  en más de 200 mercados.
- Podemos asignar a  $X_1$  al presupuesto asignado a TV,  $X_2$  al presupuesto asignado a radio y  $X_3$  al presupuesto asignado a periódico (aleatoriamente).
- Si la empresa desea establecer cuál medio es más importante para invertir en publicidad e incrementar sus ventas, debemos de tomar a  $X = (X_1, X_2, X_3)$  y  $Y$  (i.e.,  $\mathcal{T}$ ) y generar un **modelo**:

$$\text{ventas} \approx f(\text{TV, radio, periódico})$$

## Una motivación (2)

- En otras palabras, asumimos que existe una relación del tipo:

$$Y = f(X) + \epsilon \quad (1)$$

donde  $X = (X_1, \dots, X_p)$  y  $\epsilon$  es un *término de error* aleatorio (o *error irreducible*), independiente de  $X$  y con media 0.

$$\mathbb{E} [\epsilon] = 0$$

- Desconocemos a  $f$ , por lo que debemos de *estimarla* utilizando los datos observados.

## Una motivación (3)

- ¿Para qué queremos encontrar a  $f$ ?

- Deseamos realizar **predicciones** de  $Y$  cuando  $X = x$ :

$$\hat{Y} = \hat{f}(X) \quad (2)$$

- Podremos entender cuáles componentes de  $X = (X_1, \dots, X_p)$  son importantes para poder explicar a  $Y$  (**inferencia**).
  - Dependiendo de la complejidad de  $f$ , podremos entender qué tanto afecta cada  $X_j \in X$  a  $Y$ .
- ★ Por lo que, en esencia SL se refiere a un conjunto de métodos para estimar a  $f$ .

## Una motivación (4)

- Si asumimos que tanto  $\hat{f}$  y  $X$  están fijos, entonces (usando a las Ecuaciones 1 y 2):

$$\begin{aligned}\mathbb{E} [(Y - \hat{Y})^2] &= \mathbb{E} [(f(X) + \epsilon - \hat{f}(X))^2] \\&= \mathbb{E} [(f(X) - \hat{f}(X))^2] + 2\mathbb{E} [(f(X) - \hat{f}(X))\epsilon] + \mathbb{E} [\epsilon^2] \\&= \mathbb{E} [(f(X) - \hat{f}(X))^2] + 2\mathbb{E} [f(X) - \hat{f}(X)] \mathbb{E} [\epsilon] + \\&\quad + \mathbb{E} [\epsilon^2]\end{aligned}$$

- Es decir:

$$\mathbb{E} [(Y - \hat{Y})^2] = (f(X) - \hat{f}(X))^2 + \mathbb{V}(\epsilon) \quad (3)$$

- **Error reducible**
- **Error irreducible**

## Una motivación (4)

- Si asumimos que tanto  $\hat{f}$  y  $X$  están fijos, entonces (usando a las Ecuaciones 1 y 2):

$$\begin{aligned}\mathbb{E} [(Y - \hat{Y})^2] &= \mathbb{E} [(f(X) + \epsilon - \hat{f}(X))^2] \\&= \mathbb{E} [(f(X) - \hat{f}(X))^2] + 2\mathbb{E} [(f(X) - \hat{f}(X))\epsilon] + \mathbb{E} [\epsilon^2] \\&= \mathbb{E} [(f(X) - \hat{f}(X))^2] + 2\mathbb{E} [f(X) - \hat{f}(X)] \mathbb{E} [\epsilon] + \\&\quad + \mathbb{E} [(\epsilon - 0)^2]\end{aligned}$$

- Es decir:

$$\mathbb{E} [(Y - \hat{Y})^2] = (f(X) - \hat{f}(X))^2 + \mathbb{V}(\epsilon) \quad (3)$$

- **Error reducible**
- **Error irreducible**

## Una motivación (4)

- Si asumimos que tanto  $\hat{f}$  y  $X$  están fijos, entonces (usando a las Ecuaciones 1 y 2):

$$\begin{aligned}\mathbb{E} [(Y - \hat{Y})^2] &= \mathbb{E} [(f(X) + \epsilon - \hat{f}(X))^2] \\&= \mathbb{E} [(f(X) - \hat{f}(X))^2] + 2\mathbb{E} [(f(X) - \hat{f}(X))\epsilon] + \mathbb{E} [\epsilon^2] \\&= \mathbb{E} [(f(X) - \hat{f}(X))^2] + 2\mathbb{E} [f(X) - \hat{f}(X)] \mathbb{E} [\epsilon] + \\&\quad + \mathbb{E} [(\epsilon - 0)^2]\end{aligned}$$

- Es decir:

$$\mathbb{E} [(Y - \hat{Y})^2] = \boxed{(f(X) - \hat{f}(X))^2} + \boxed{\mathbb{V}(\epsilon)} \quad (3)$$

- **Error reducible**
- **Error irreducible**

## Una motivación (4)

- Si asumimos que tanto  $\hat{f}$  y  $X$  están fijos, entonces (usando a las Ecuaciones 1 y 2):

$$\begin{aligned}\mathbb{E} [(Y - \hat{Y})^2] &= \mathbb{E} [(f(X) + \epsilon - \hat{f}(X))^2] \\&= \mathbb{E} [(f(X) - \hat{f}(X))^2] + 2\mathbb{E} [(f(X) - \hat{f}(X))\epsilon] + \mathbb{E} [\epsilon^2] \\&= \mathbb{E} [(f(X) - \hat{f}(X))^2] + 2\mathbb{E} [f(X) - \hat{f}(X)] \mathbb{E} [\epsilon] + \\&\quad + \mathbb{E} [(\epsilon - 0)^2]\end{aligned}$$

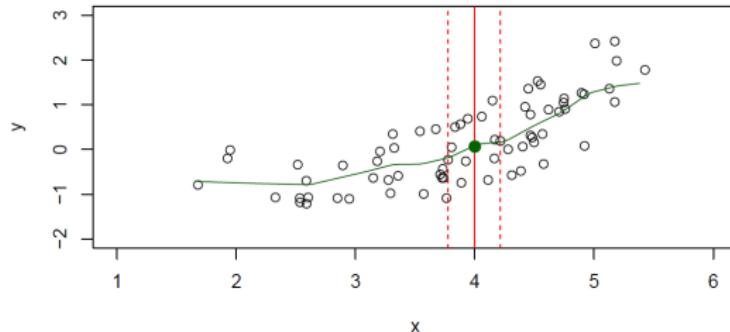
- Es decir:

$$\mathbb{E} [(Y - \hat{Y})^2] = \text{(f}(X) - \hat{f}(X))^2 + \mathbb{V}(\epsilon) \quad (3)$$

- **Error reducible**
- **Error irreducible**

# ¿Cómo estimamos a $f$ ? (1)

- Imaginen que tenemos una serie de datos:



- $f(x) = \mathbb{E}[Y|X=x]$  será la *función de regresión*.
- ¿Cuál es un buen valor de  $f(X)$  cuando  $X=4$ ?
- No siempre podremos encontrar dicho valor, por lo que podemos *relajar* la definición:

$$\hat{f}(x) = \text{Ave}(Y|X \in \mathcal{N}(x))$$

donde  $\mathcal{N}(x)$  es un *vecindario* de  $x$ .

## ¿Cómo estimamos a $f$ ? (2)

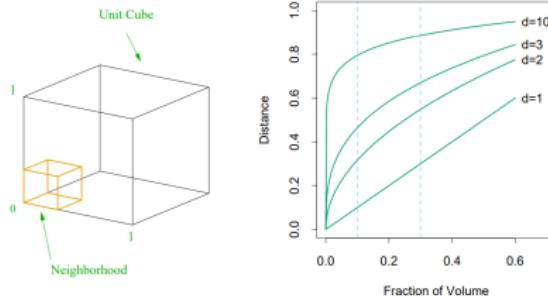
- El promedio puede funcionar muy bien para  $p \leq 4$  y un  $n$  (mas o menos) grande.
- Todo se rompe en dimensiones altas (**maldición de la dimensión**).
- ¿Qué proporción de puntos se encuentran en la *frontera* de un hipercubo unitario de dimensión  $d = 50$ ?
  - Nuestro hipercubo se define como  $[0, 1]^d = [0, 1]^{50}$ .
  - La frontera se definirá como el conjunto de todos los puntos para el que existe un  $j$  tal que  $0 \leq x_j \leq 0.05$  ó  $0.95 \leq x_j \leq 1$ .
  - Entonces, la proporción de puntos que *no* se encuentra en la frontera será:

$$\frac{(1 - 0.05 - 0.05)^{50}}{(1 - 0)^{50}} = \left(\frac{0.9}{1}\right)^{50} \approx 0.005$$

es decir, el 99.5% de los puntos se encontrará en la frontera.

## ¿Cómo estimamos a $f$ ? (3)

- Si los datos están repartidos uniformemente, vemos en el gráfico a la derecha a la longitud  $a$  de la arista necesaria para que el subcubo capture una fracción  $r$  del volumen de datos:  $a^p = r \iff a = r^{1/p}$ .



- Para  $p = 10$ , para poder capturar 1% de los datos ( $r = 0.01$ ), la arista esperada del subcubo medirá  $a = 0.01^{1/10} = 0.63$ , mientras que para capturar el 10% de los datos, la arista del subcubo medirá  $a = 0.1^{1/10} = 0.80$ .
- El estimado del vecindario  $\mathcal{N}(x)$  o *vecinos cercanos* para  $x$  ya no será un método local.

# Modelos Paramétricos: Modelo Lineal (1)

- Empezemos por el modelo paramétrico más sencillo (pero no necesariamente el más simple): un *model lineal*.
- Un modelo lineal predice el *output*  $Y$  de la siguiente manera:

$$\hat{Y} = \hat{\beta}_0 + \sum_{j=1}^p X_j \hat{\beta}_j \quad (4)$$

donde  $\hat{\beta}_0$  es el *intercepto* o *sesgo*.

- Un modelo lineal es especificado en términos de sus  $p + 1$  parámetros  $\hat{\beta}_0, \dots, \hat{\beta}_p$ .

## Modelos Paramétricos: Modelo Lineal (2)

- Si bien casi nunca es acertado, el modelo lineal nos sirve para tener una aproximación interpretable de  $f$ .
- Si  $X = (1, X_1, \dots, X_p)^\top$  y  $\hat{\beta} = (\hat{\beta}_0, \dots, \hat{\beta}_p)^\top$ , entonces podemos escribir a la Ecuación 4 como:

$$\hat{Y} = X^\top \hat{\beta} \quad (5)$$

- Estimamos a los parámetros al ajustar a nuestro modelo a los datos de entrenamiento  $\mathcal{T}$ .
- La forma más común es utilizando **mínimos cuadrados** para así escoger a  $\hat{\beta}$  que minimize la *suma residual de cuadrados (RSS)*:

$$\text{RSS}(\beta) = \sum_{i=1}^n (y_i - x_i^\top \beta)^2$$

## Modelos Paramétricos: Modelo Lineal (3)

- Como RSS es una función cuadrática, existirá un mínimo mas puede no ser único.
- En notación matricial, podemos reescribir a RSS como:

$$\text{RSS}(\beta) = (\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta) \quad (6)$$

donde  $\mathbf{X} \in \mathbb{R}^{n \times (p+1)}$  y  $\mathbf{y} \in \mathbb{R}^n$ .

- Si  $\mathbf{X}^T\mathbf{X}$  es no singular (tiene inversa), entonces la solución a

$$\hat{\beta} = \arg \min_{\beta} (\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta)$$

esta dada por

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \quad (7)$$

# Modelos Paramétricos: Modelo Lineal (4)

- La ecuación 7 se llama la *ecuación normal*  $\Rightarrow p + 1$  parámetros .
- Se encuentra fácilmente al derivar a RSS respecto de  $\beta$ .
- Podemos encontrar numéricamente a  $\hat{\beta}$  mediante la ecuación normal o utilizando el algoritmo de **descenso gradiente**.

Descenso Gradiente	Ecuación Normal
Necesitamos escoger $\alpha$ Necesita muchas iteraciones $\mathcal{O}(kp^2)$ Funciona bien aunque $p$ sea grande	No necesitamos escoger $\alpha$ No hay necesidad de iteración $\mathcal{O}(p^3)$ Lento cuando $p$ es grande

# Modelos Paramétricos: Modelo Lineal (5)

- ¿Qué pasa si aplicamos el modelo de regresión lineal a un conjunto de datos categóricos?
- $\mathcal{T} = \{(x_i, y_i)\}_{i=1}^n$ , con  $y_i \in \{\text{NARANJA}, \text{AZUL}\}$ .
- Un modelo de regresión lineal nos dará a  $\hat{\beta}$  tal que:

$$\hat{G} = \begin{cases} \text{NARANJA} & \text{si } X^\top \hat{\beta} > 0.5 \\ \text{AZUL} & \text{si } X^\top \hat{\beta} \leq 0.5 \end{cases}$$

# Modelos Paramétricos: Modelo Lineal (6)

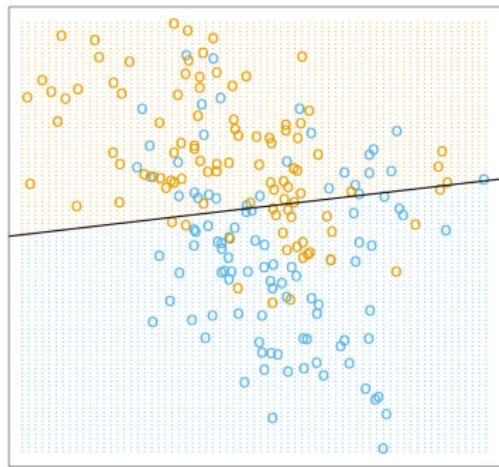


Figura 4: Generamos datos en dos dimensiones con clases AZUL y NARANJA. Para realizar el modelo, codificamos a las clases como una variable binaria ( $\text{AZUL} = 0$  y  $\text{NARANJA} = 1$ ). Vemos que el modelo lineal es demasiado rígido y, por inspección visual, las dos clases no son linealmente separables.  $X^T \hat{\beta} = 0.5$  define a la *frontera de decisión*.

# Modelos No Paramétricos: Vecinos Más Cercanos (1)

- Definimos al modelo de *k vecinos más cercanos (KNN)* como:

$$\hat{Y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i \quad (8)$$

o en palabras: encuentre las  $k$  observaciones  $x_i$  más cercanas a  $x$  y calcule el promedio de sus respectivas respuestas  $y_i$ .

- ¿Cómo definimos *cercano*?

- Usaremos (en general) la **distancia Euclídea** o **norma L2**. Para el vector  $\mathbf{x} = (x_1, \dots, x_p)$ :

$$\|\mathbf{x}_i\|_2 = \sqrt{\sum_{i=1}^p x_i^2}$$

- Existen más normas que iremos definiendo cuando sea pertinente.

- $k$  es un **hiperparámetro**.
- Los parámetros de los modelos no crecen conforme  $\mathcal{T}$  crece.
- **Ventaja principal:**
  - No asumimos la forma de  $f$ , por lo que cubren una mayor cantidad de formas de  $f$ .
- **Desventaja principal:**
  - Necesitamos más datos para conseguir un estimado correcto de  $f$ .

## Modelos No Paramétricos: Vecinos Más Cercanos (3)

- Para la clasificación binaria,  $\mathcal{T} = \{(x_i, y_i)\}_{i=1}^n$ , donde  $y_i \in \{0, 1\}$  (de regreso a nuestro ejemplo anterior.)
- El estimado de  $G$ ,  $\hat{G}$  dado por KNN es:

$$\hat{G} = \begin{cases} \text{NARANJA} & \text{si } \frac{1}{k} \sum_{x_i \in N_k(x)} y_i > 0.5 \\ \text{AZUL} & \text{si } \frac{1}{k} \sum_{x_i \in N_k(x)} y_i \leq 0.5 \end{cases}$$

- Es decir, encontramos las  $k$  observaciones  $x_i$  más cercanas a  $x$  y estimamos la clase de  $x$  como la que pertenece la *mayoría* de sus vecinos.

## KNN, $k = 15$

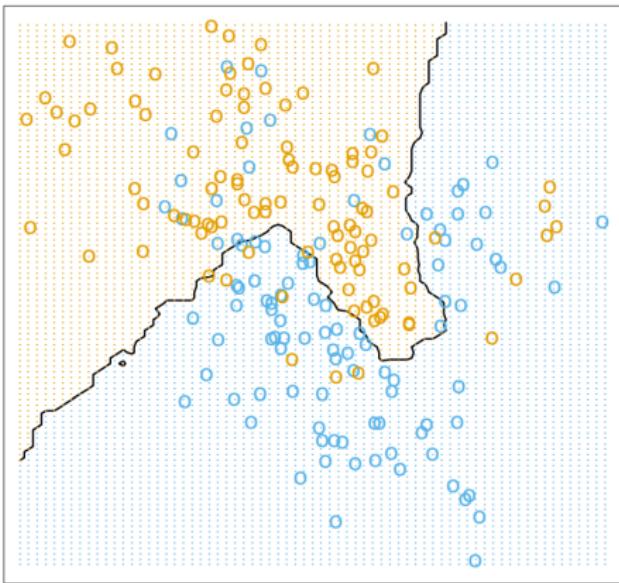
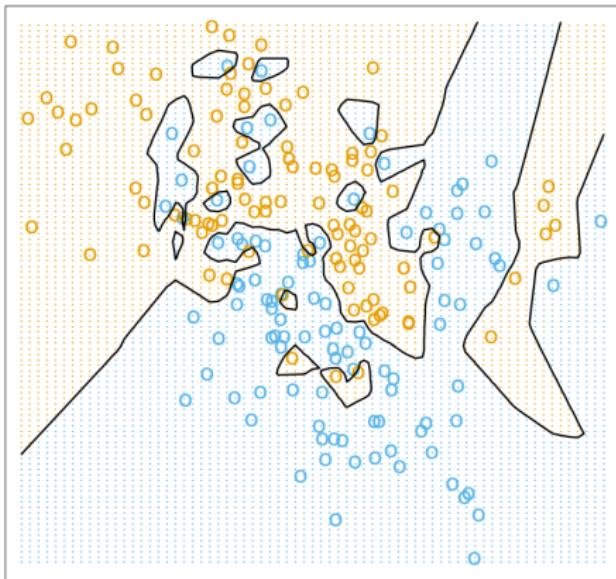


Figura 5: Usamos los mismos datos que en la Figura 34; codificamos a las clases como una variable binaria (**AZUL** = 0 y **NARANJA** = 1) y lo ajustamos a un modelo de KNN con  $k = 15$ . Por ende, la clase predicha es escogida por voto mayoritario entre los 15 vecinos más cercanos.

# KNN, $k = 1$



**Figura 6:** Usamos los mismos datos que en la Figura 34; codificamos a las clases como una variable binaria (**AZUL** = 0 y **NARANJA** = 1) y lo ajustamos a un modelo de KNN con  $k = 1$ . Llegamos a lo que comúnmente se denomina como *Tesselación de Voronoi*.

- Nótese que para  $k = 1$ , nuestro clasificador no comete errores.
- ¿Qué tan bien le irá clasificando datos que no ha visto antes?
- Definimos al *conjunto de entrenamiento*  $\mathcal{T}_{\text{Tr}}$  como los datos que utilizamos para entrenar nuestro algoritmos.
- Definimos a *conjunto de prueba*  $\mathcal{T}_{\text{Te}}$  como los datos que utilizamos para realizar predicciones utilizando a nuestro algoritmo.
- Es importante que no hayan datos cruzados, ni que se escogen a mano cuál irá en cada conjunto.
  - Para  $|\mathcal{T}| \sim 100000$ ,  $|\mathcal{T}_{\text{Tr}}|/|\mathcal{T}| \approx 0.7$  y  $|\mathcal{T}_{\text{Te}}|/|\mathcal{T}| \approx 0.3$ .
  - Para  $|\mathcal{T}| > 1000000$ ,  $|\mathcal{T}_{\text{Tr}}|/|\mathcal{T}| \approx 0.98$  y  $|\mathcal{T}_{\text{Te}}|/|\mathcal{T}| \approx 0.02$ .

```
import numpy as np
from sklearn.model_selection import train_test_split

X = np.random.randn(10).reshape((5, 2))

y = np.random.choice(2, 5)

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42, shuffle=True)
```

# ¿Qué modelo elegimos?

- Debemos de escoger una métrica con la cual medir a nuestros modelos que hemos entrenado con el conjunto de entrenamiento  $\mathcal{T}_{\text{Tr}}$ .
- Específicamente, nos interesa saber qué tan precisos son nuestros modelos tanto en  $\mathcal{T}_{\text{Tr}}$  como en  $\mathcal{T}_{\text{Te}}$ .
- El *error cuadrático medio (MSE) de entrenamiento* es:

$$\text{MSE}_{\mathcal{T}_{\text{Tr}}} = \text{Ave}_{i \in \mathcal{T}_{\text{Tr}}} (y_i - \hat{f}(x_i))^2 \quad (9)$$

- El *error cuadrático medio (MSE) de prueba* calculado sobre  $\mathcal{T}_{\text{Te}}$  es:

$$\text{MSE}_{\mathcal{T}_{\text{Te}}} = \text{Ave}_{i \in \mathcal{T}_{\text{Te}}} (y_i - \hat{f}(x_i))^2 \quad (10)$$

# Eligiendo a $k$ en KNN

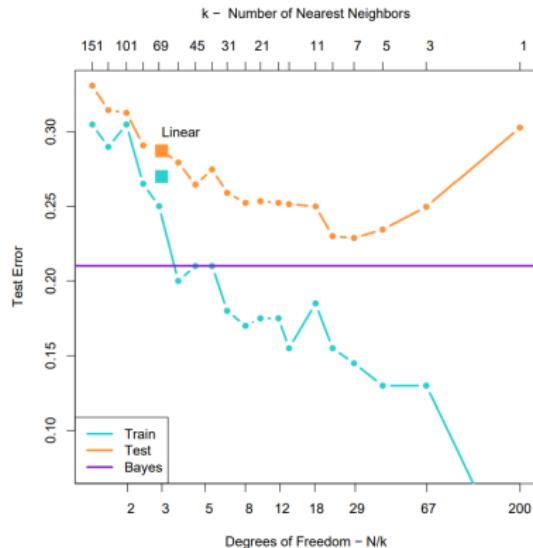


Figura 7: Curvas de MSE de entrenamiento y de prueba para distintos valores de  $k$ . Se presentan además los mismos valores para nuestro modelo lineal. El error para la frontera de decisión óptima se conoce como el *ritmo de error de Bayes*.

# Frontera de Decisión de Bayes

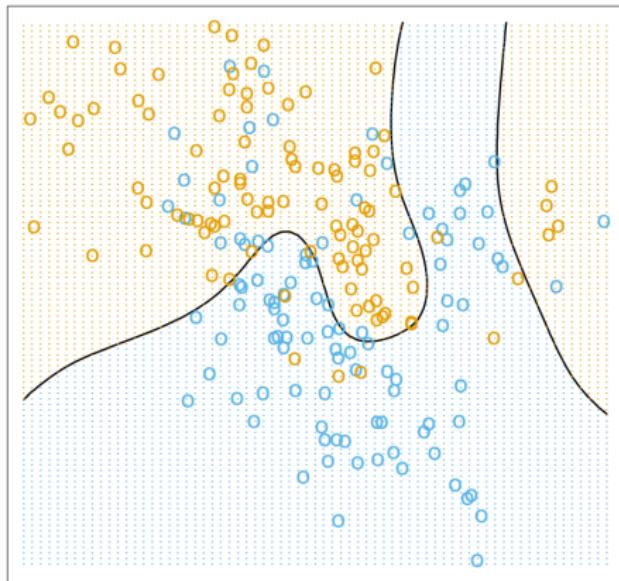
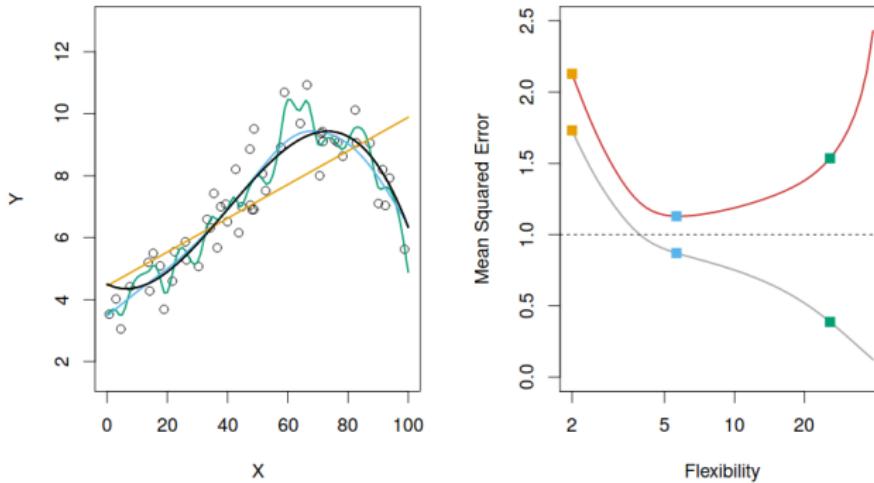


Figura 8: La frontera de decisión óptima para los datos anteriores. La podemos calcular ya que se conocen las densidades de probabilidad para con las cuales se generaron las dos series de datos.

# MSE para Tres Curvas (1)



**Figura 9:** La curva negra indica la función *real*  $f$  con la cual generamos los datos. Se muestran tres aproximaciones y sus respectivos  $MSE_{\mathcal{T}_{Tr}}$  (en gris) y  $MSE_{\mathcal{T}_{Te}}$  (en rojo): regresión lineal en naranja y dos *curvas spline* en azul y verde. La línea punteada representa el error de Bayes o  $\mathbb{V}[\epsilon]$ .

## MSE para Tres Curvas (2)

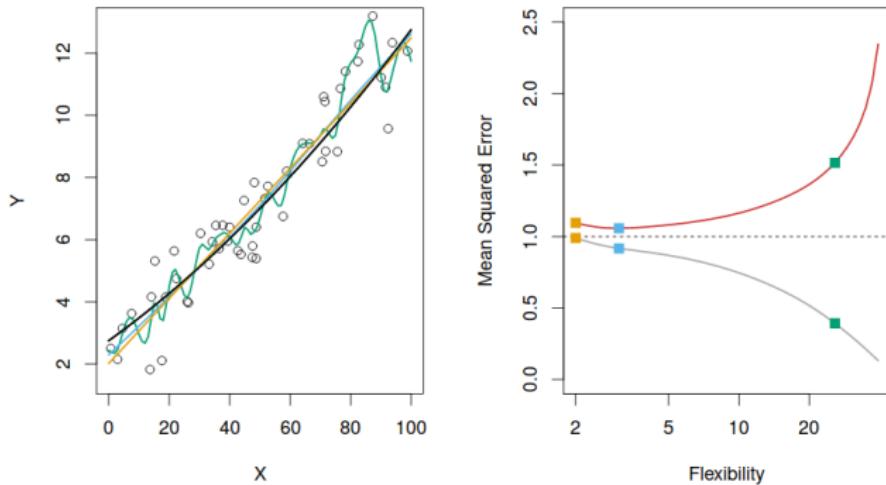


Figura 10: Los detalles son los mismos que para la Figura 9. Nótese que ya que  $f$  es más cercano a ser lineal, la regresión lineal realizará un buen ajuste a los datos.

## MSE para Tres Curvas (3)

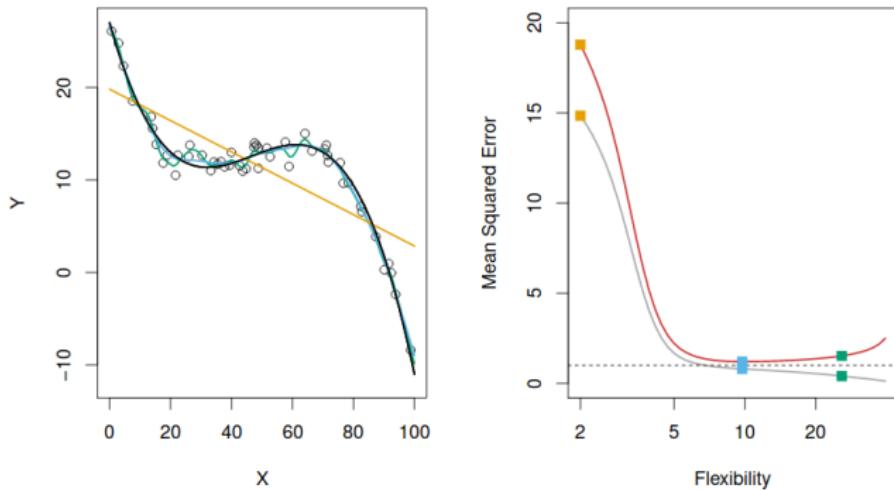


Figura 11: Los detalles son los mismos que para la Figura 9. Ahora  $f$  está lejos de ser lineal, por lo que la regresión lineal realizará un ajuste muy pobre a los datos.

# Intercambio entre Varianza y Sesgo (1)

- Hacemos la siguiente observación:
  - Mientras más flexible sea un modelo, menor será su  $MSE_{\mathcal{T}_{Tr}}$ , pero mayor será su  $MSE_{\mathcal{T}_{Te}}$ .
  - Por lo tanto, un  $MSE_{\mathcal{T}_{Tr}}$  bajo **no es un buen indicador** de un  $MSE_{\mathcal{T}_{Te}}$  bajo.
- En general, podemos descomponer al  $MSE_{\mathcal{T}_{Te}}$  de la siguiente manera para una observación de prueba  $(x_0, y_0)$ :

$$MSE_{\mathcal{T}_{Te}}(x_0) = \mathbb{E}_{\mathcal{T}_{Te}}[(y_0 - \hat{f}(x_0))^2] \quad (11)$$

$$= (\text{Bias}[\hat{f}(x_0)])^2 + \mathbb{V}[\hat{f}(x_0)] + \mathbb{V}[\epsilon] \quad (12)$$

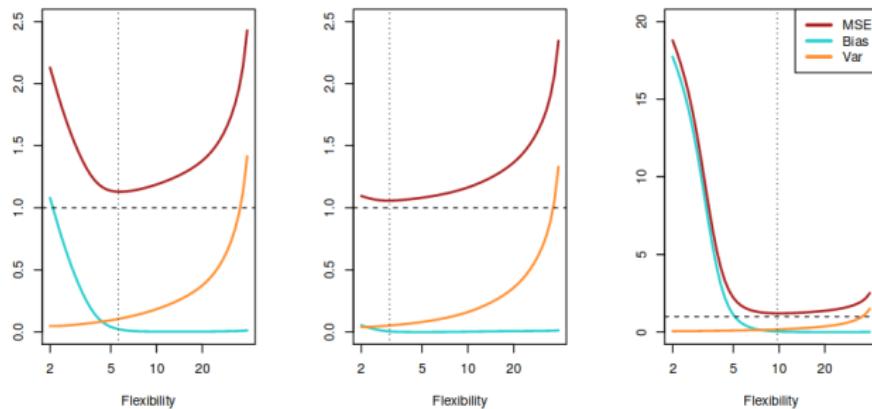
donde

$$\text{Bias}[\hat{f}(x)] = \mathbb{E}_{\mathcal{T}_{Te}}[\hat{f}(x)] - f(x) \quad \mathbb{V}[\hat{f}(x)] = \mathbb{E}_{\mathcal{T}_{Te}}[\hat{f}(x)^2] - \mathbb{E}_{\mathcal{T}_{Te}}[\hat{f}(x)]^2$$

## Intercambio entre Varianza y Sesgo (2)

- Queremos entonces que nuestra estimación  $\hat{f}$  obtenga un sesgo (Bias) bajo y una varianza baja.
- Error irreducible :(
- *Varianza* de un modelo se refiere a la cantidad que  $\hat{f}$  cambiaría si usaramos un conjunto de entrenamiento distinto para ajustarlo.
- *Bias* (o *Sesgo*) se refiere al error introducido al tratar de aproximar un problema de la vida real.
- El *intercambio entre varianza y sesgo* se refiere al hecho de que los modelos con sesgo bajo tienen varianza alta y viceversa  $\Rightarrow$  **dilemma**.

# Intercambio entre Varianza y Sesgo (3)



**Figura 12:** Descomposición del  $MSE_{T_e}$  para los datos de las Figuras 9, 10 y 11. La línea vertical punteada indica el nivel de flexibilidad correspondiente al valor mínimo de  $MSE_{T_e}$ . En problemas reales, solo tendremos que estimar a  $MSE_{T_e}$ , usando, por ejemplo, **validación cruzada**.

## Intercambio entre Varianza y Sesgo (4)

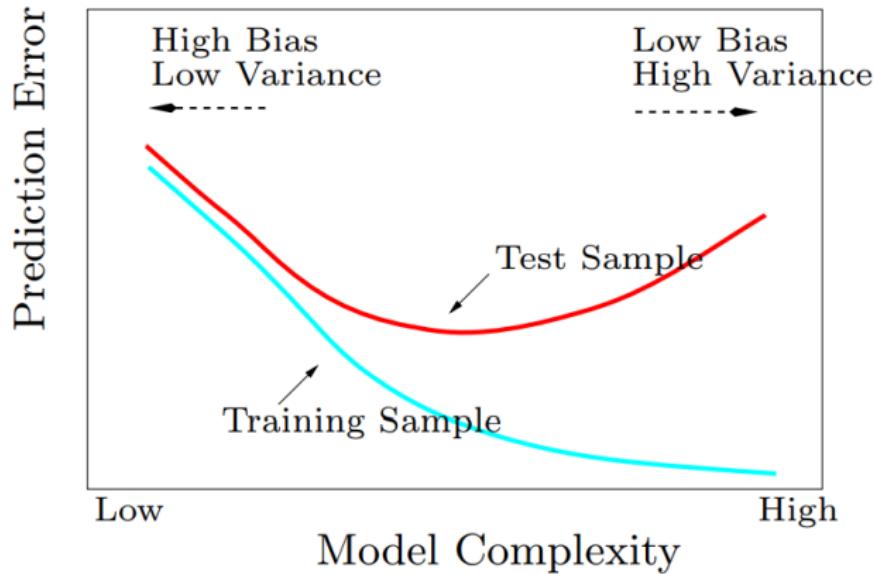
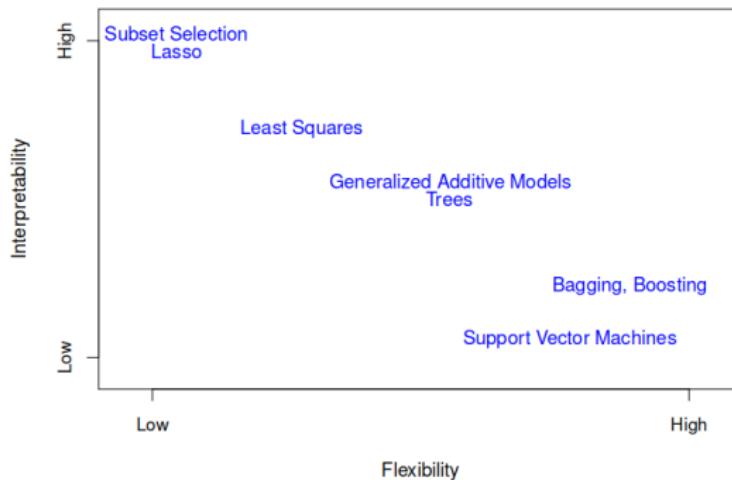


Figura 13:  $MSE_{T_e}$  y  $MSE_{T_r}$  en función de la complejidad del modelo. Si tenemos Bias alto y Varianza baja, tenemos un problema de *underfitting*. Si tenemos el Bias bajo y la Varianza alta, tenemos un problema de *overfitting*.

# Intercambio entre Precisión e Interpretabilidad



- Mientras mas *flexible* un modelo, será mayor su capacidad de generar funciones para aproximar a  $f$ .
- Preferimos modelos mas simples que involucren menos variables sobre un predictor de **caja negra** que involucre todas las variables.