# Lecture 6: Machine Learning
# Paradigma Predictivo

## Big Data and Machine Learning en el Mercado Inmobiliario
## Educación Continua

Ignacio Sarmiento-Barbieri

Universidad de los Andes

March 23, 2022

# Agenda

1. Recap
2. Precio de Propiedades
3. Predicción: estadística clásica vs la máquina de aprender
   - Tipos de Aprendizaje
   - Prediction vs Estimation
   - Overfit y Predicción fuera de Muestra
4. Error de predicción y métodos de remuestreo
   - Enfoque de conjunto de validación
   - LOOCV
   - Validación cruzada en K-partes
5. Para seguir leyendo
6. Break

# Modelo Monocéntrico

- Planteamos un modelo monocentrico
- Nos ilustraba de manera sencilla la importancia de ciertos elementos a considerar:
  - Distancia al empleo
  - Costos de transporte
  - Costos de construcción (elasticidad)
  - Amenidades

# Precio de Propiedades

- Con estas consideraciones, podemos pensar una casa como un bien diferenciado: H que tiene distintas características $(x_1, x_2, \ldots, x_n)$

$$H = (x_1, x_2, \ldots, x_n) \tag{1}$$

- El precio de este producto (y) es función de estos atributos

$$y = f(x_1, x_2, \ldots, x_n) \tag{2}$$

- El precio de equilibrio para cada variedad del bien diferenciado (por ejemplo una casa en particular) es una función de los atributos de la misma.

# Estadística clásica vs la máquina de aprender

$$y = f(X) + u \tag{3}$$

- Estadística Clásica
  - Inferencia
  - $f()$ "correcta" el interes es en entender como $y$ afecta $X$
  - modelos surge de la teoria/experimentos
  - Interés es en test de hipótesis (std. err., ci's)
- Maquina de Aprender
  - Interés es predecir $y$
  - El $f()$ correcto es el que predice mejor
  - Modelo?

# ¿Qué es la máquina de aprender?

- El aprendizaje de máquinas es una rama de la informática y la estadística, encargada de desarrollar algoritmos para predecir los resultados $y$ a partir de las variables observables $X$.
- La parte de aprendizaje proviene del hecho de que no especificamos cómo exactamente la computadora debe predecir $y$ a partir de $X$.
- Esto queda como un problema empírico que la computadora puede ”aprender”.
- En general, esto significa que nos abstraemos del modelo subyacente, el enfoque es muy pragmático

# ¿Qué es la máquina de aprender?

- El aprendizaje de máquinas es una rama de la informática y la estadística, encargada de desarrollar algoritmos para predecir los resultados $y$ a partir de las variables observables $X$.
- La parte de aprendizaje proviene del hecho de que no especificamos cómo exactamente la computadora debe predecir $y$ a partir de $X$.
- Esto queda como un problema empírico que la computadora puede ”aprender”.
- En general, esto significa que nos abstraemos del modelo subyacente, el enfoque es muy pragmático

**“Lo que sea que funciona, funciona...”**

# “Lo que sea que funciona, funciona...”

# Tipos de Aprendizaje

- ML se divide en dos (¿?) ramas principales:
  1. Aprendizaje supervisado: Tenemos datos tanto sobre un resultado $y$ como sobre las variables explicativas $X$.
     - Esto es lo más cercano al análisis de regresión que conocemos.
     - Si $y$ es discreto, también podemos ver esto como un problema de clasificación.
     - Es el enfoque de este curso.
  2. Aprendizaje no supervisado: No tenemos datos sobre $y$, solo sobre $X$.
     - Queremos agrupar estos datos (sin especificar qué agrupar).
     - Permite reducir la dimensionalidad y explorar datos
     - Algunos algoritmos destacados: PCA, y K-medias

# Predicción y Error Predictivo

- El objetivo es predecir $y$ dadas otras variables $X$. Ej: precio vivienda dadas las características
- Asumimos que el link entre $y$ and $X$ esta dado por el modelo:

$$y = f(X) + u \tag{4}$$

- donde $f(X)$ es cualquier función,
- $u$ una variable aleatoria no observable $E(u) = 0$ and $V(u) = \sigma^2$

# Como medimos: "Lo que sea que funciona, funciona..."

- En la práctica no conocemos $f(X)$
- Es necesario estimarla $\hat{y} = \hat{f}(X)$
- La medida de cuán bien funciona nuestro modelo es

$$MSE(\hat{y}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{5}$$

$$= \frac{1}{n}\sum_{i=1}^{n}(f_i - \hat{f}_i)^2 \tag{6}$$

# Como medimos: "Lo que sea que funciona, funciona..."

- Podemos descomponer el *MSE* en dos partes

$$MSE(\hat{y}) = MSE(\hat{f}) + \sigma^2 \tag{7}$$

- el error de estimar $f$ con $\hat{f}$. (*reducible*)
- el error de no observar $u$. (*irreducible*)

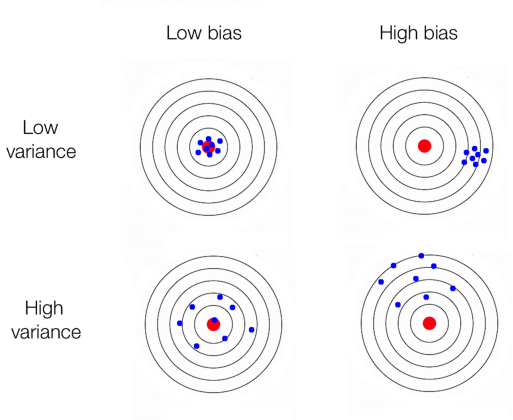
# Predicción y Error Predictivo

- Descomponiendo un poco más:

$$Err(Y) = MSE(\hat{f}) + \sigma^2 \tag{8}$$
$$= Bias^2(\hat{f}) + V(\hat{f}) + Error\ Irreducible \tag{9}$$

- Este resultado es muy importante,
  - Aparece el dilema entre sesgo y varianza

# Dilema sesgo/varianza



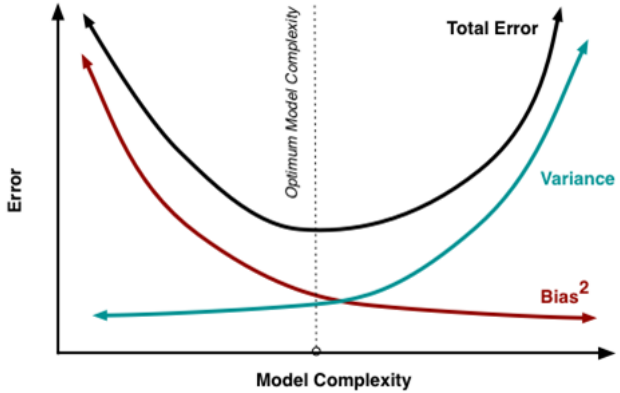

Source: https://tinyurl.com/y4lvjxpc

# Dilema sesgo/varianza

- El secreto de ML: admitiendo un poco de sesgo podemos tener ganancias importantes en varianza

# Dilema sesgo/varianza

- El secreto de ML: admitiendo un poco de sesgo podemos tener ganancias importantes en varianza




Source: https://tinyurl.com/y4lvjxpc

# Predicción y regresión lineal

- El problema es:

$$y = f(X) + u \tag{10}$$

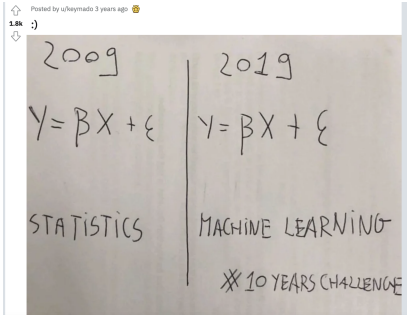
# Predicción y regresión lineal

- El problema es:

$$y = f(X) + u \tag{10}$$

- proponemos :

$$f(X) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p \tag{11}$$




Fuente: https://www.reddit.com/r/datascience/comments/ab0g69/ /

# Predicción y regresión lineal

- Y el dilema sesgo varianza?

# Predicción y regresión lineal

- Y el dilema sesgo varianza?
- Bajo los supuestos clásicos (Gauss-Markov) el estimador de OLS es insesgado:

$$
\begin{aligned}
E(X\hat{\beta}) &= E(\hat{\beta}_1 + \hat{\beta}_2 X_2 + \cdots + \hat{\beta}_p X_p) && (12)\\
&= E(\hat{\beta}_1) + E(\hat{\beta}_2) X_2 + \cdots + E(\hat{\beta}_p) X_p && (13)\\
&= X\beta && (14)
\end{aligned}
$$

- $MSE(\hat{y})$ se reduce a $V(\hat{\beta})$

# Complejidad y compensación de varianza/sesgo

- En la econometría clásica, la elección de modelos se resume a elegir entre modelos más pequeños y más grandes.
- Considere los siguientes modelos para estimar $y$:

$$y = \beta_1 X_1 + u_1$$

- $\hat{\beta}_1^{(1)}$ el estimador de OLS $y$ on $X_1$
- La predicción es:

$$\hat{y}^{(1)} = \hat{\beta}_1^{(1)} X_1$$

$$y = \beta_1 X_1 + \beta_2 X_2 + u_2$$

- $\hat{\beta}_1^{(2)}$ y $\hat{\beta}_2^{(2)}$ con $\beta_1$ y $\beta_2$ los el estimador de OLS de $y$ en $X_1$ y $X_2$.
- La predicción es:

$$\hat{y}^{(2)} = \hat{\beta}_1^{(2)} X_1 + \hat{\beta}_2^{(2)} X_2$$

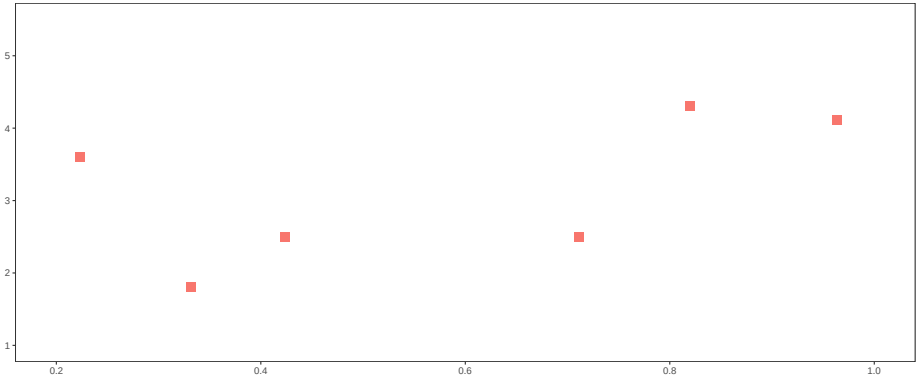
# Complejidad y compensación de varianza/sesgo

- Una discusión importante en la econometría clásica es la de la omisión de variables relevantes frente a la inclusión de variables irrelevantes.
  - Si el modelo (1) es verdadero entonces estimar el modelo más grande (2) conduce a estimadores ineficientes aunque no sesgados debido a que incluyen innecesariamente $X_2$.
  - Si el modelo (2) se verdadero, estimar el modelo más pequeño (1) conduce a una estimación de menor varianza pero sesgada si $X_1$ también se correlaciona con el regresor omitido $X_2$.
- Esta discusión de pequeño vs grande siempre es con respecto a un modelo que se supone es verdadero.
- Pero en la práctica el modelo verdadero es desconocido!!!

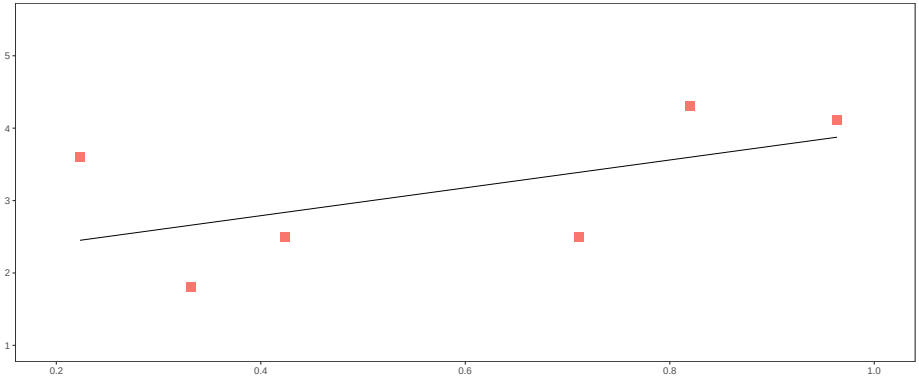
# Complejidad y compensación de varianza/sesgo

- Elegir entre modelos implica un dilema *sesgo/varianza*
- La econometría clásica tiende a resolver este dilema abruptamente,
  - requiriendo una estimación no sesgada y, por lo tanto, favoreciendo modelos más grandes para evitar sesgos
- En esta configuración simple, los modelos más grandes son "más complejos", por lo que los modelos más complejos están menos sesgados pero son más ineficientes.
- Por lo tanto, en este marco muy simple, la complejidad se mide por el número de variables explicativas.
- Una idea central en el aprendizaje automático es generalizar la idea de complejidad,
  - Nivel óptimo de complejidad, es decir, modelos cuyo sesgo y varianza conducen al menor MSE.

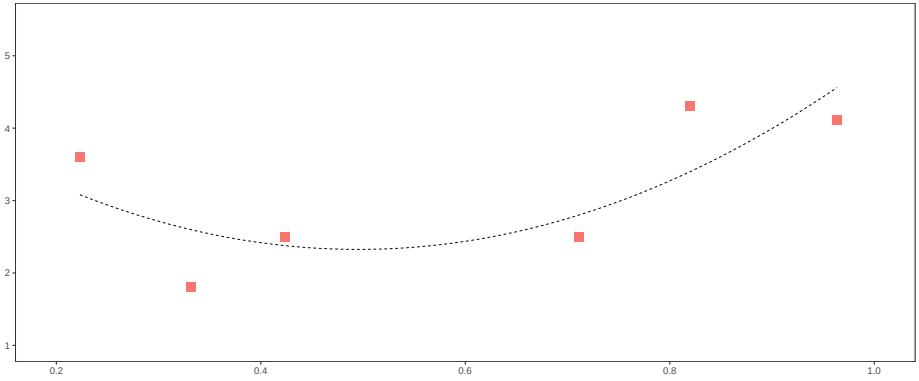
# Overfit



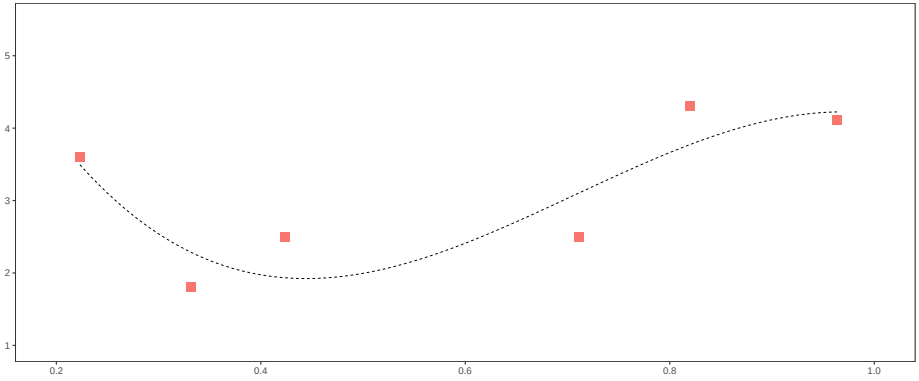
# Overfit



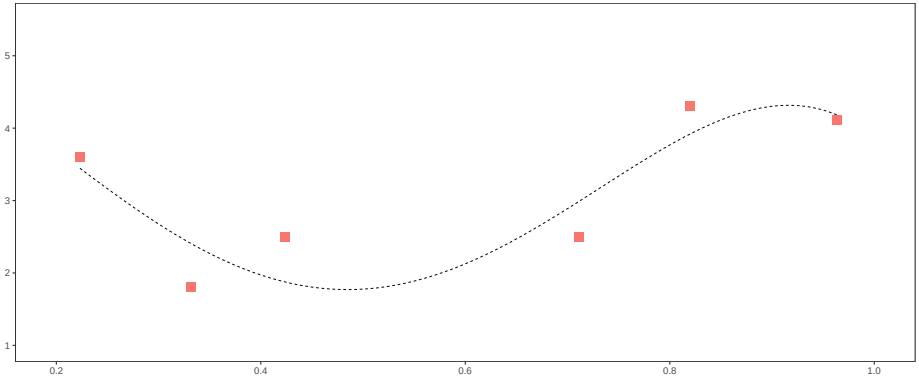
# Overfit



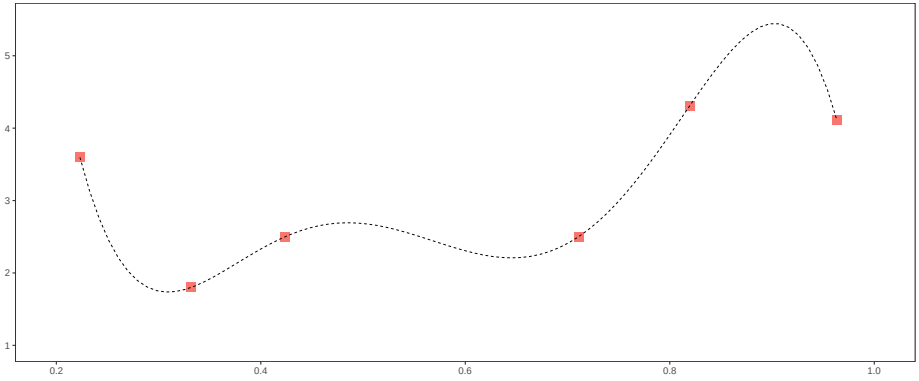
# Overfit



# Overfit



# Overfit



# Overfit

- En efecto si el modelo verdadero es $y = f(x) + u$
- donde $f$ es un polinomio de grado $p^*$, with $E(u) = 0$ and $V(u) = \sigma^2$
- con $p^*$ finito pero desconocido
- podemos ajustar polinomios de grados crecientes $p = 1, 2, ....$

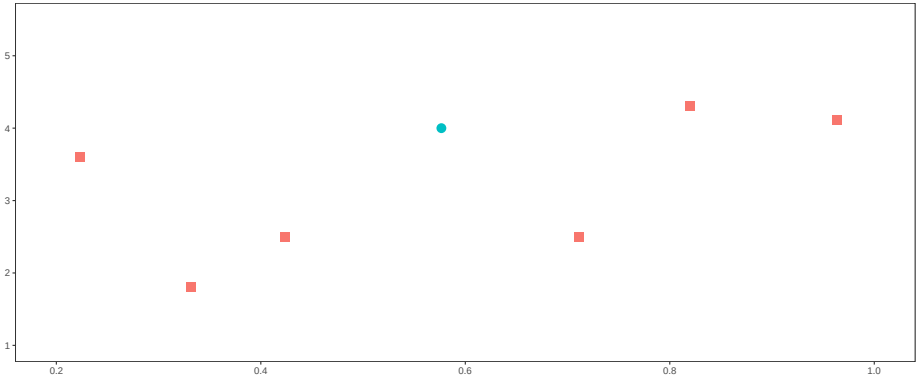
$$Err(Y) = MSE(\hat{f}) + \sigma^2 \tag{15}$$

$$= Bias^2(\hat{f}) + V(\hat{f}) + Irreducible\ Error \tag{16}$$

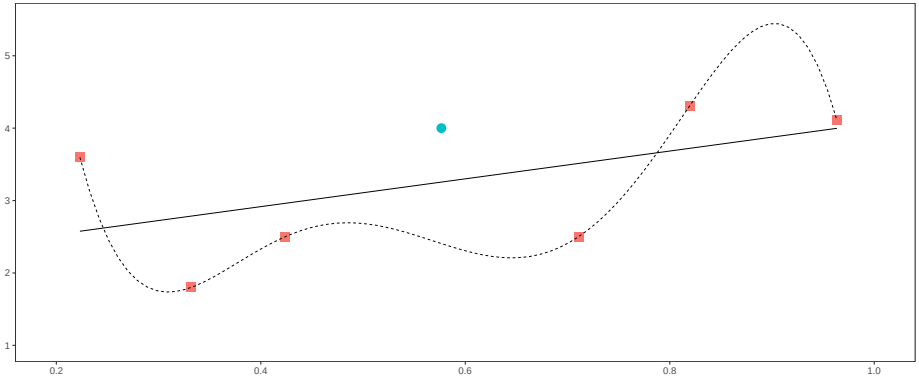
# Overfit y Predicción fuera de Muestra

- ML nos interesa la predicción fuera de muestra
- Overfit: modelos complejos predicen muy bien dentro de muestra, pero tienden a hacer un trabajo fuera de muestra
- Hay que elegir el nivel adecuado de complejidad
- Como medimos el error de predicción fuera de muestra?
- $R^2$ no funciona: se concentra en la muestra y es no decreciente en complejidad

# Overfit



# Overfit



# Métodos de remuestreo

- Los métodos de resampleo son una herramienta indispensable de la estadística moderna.
- Estos envuelven sacar muestras aleatorias de nuestra muestra y reajustar el modelo de interés en cada muestra para obtener información adicional del modelo.
- Quizás el método más conocido por ustedes es el de bootstrap.
- Nosotros vamos a discutir la validación cruzada (cross-validation)

# Error de Prueba y de Entrenamiento

- Dos conceptos importantes
  - *Test Error*: es el error de predicción en la muestra de prueba (test)

$$Err_{\mathcal{T}est} = MSE[(y, \hat{y})|\mathcal{T}est] \tag{17}$$

  - *Training error*:es el error de predicción en la muestra de entrenamiento (training)

$$Err_{\mathcal{T}rain} = MSE[(y, \hat{y})|\mathcal{T}rain] \tag{18}$$

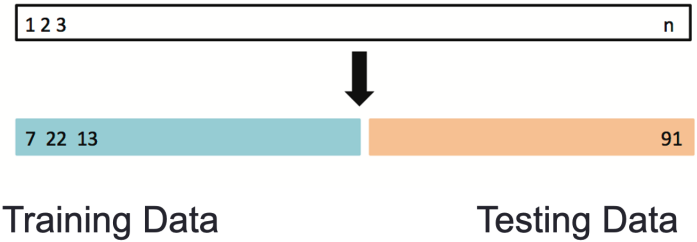
- Cómo elegimos $\mathcal{T}est$?

# Qué son los Métodos de Remuestreo?

- Herramientas que implican extraer repetidamente muestras de un conjunto de entrenamiento y reajustar el modelo de interés en cada muestra para obtener más información sobre el modelo.
- Evaluación del modelo: estimar el error de predicción en la muestra de prueba
- Selección de modelo: seleccione el nivel apropiado de flexibilidad del modelo
- ¡Son computacionalmente costosos! Pero en estos días tenemos computadoras poderosas

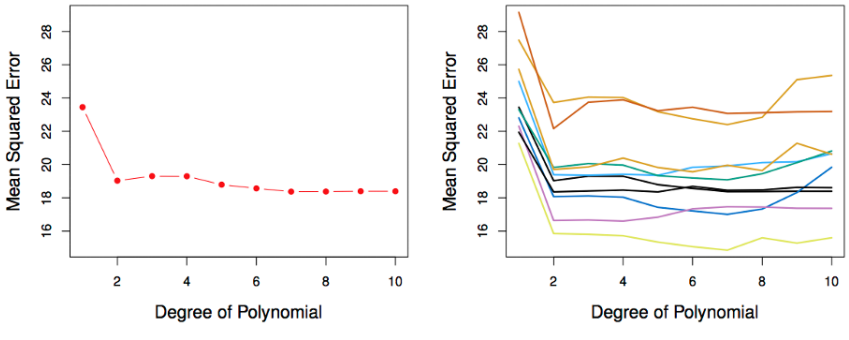
# Enfoque de conjunto de validación

- Suponga que nos gustaría encontrar un conjunto de variables que den el menor error de predicción en la muestra de prueba (no de entrenamiento)
- Si tenemos muchos datos, podemos lograr este objetivo dividiendo aleatoriamente los datos en partes de entrenamiento y validación (prueba)
- Luego usaríamos la parte de entrenamiento para construir cada modelo posible (es decir, las diferentes combinaciones de variables) y elegimos el modelo que dio lel menor error de predicción en la muestra de prueba



# Enfoque de conjunto de validación

- Modelo $y = f(x) + u$ donde $f$ es un polinomio de grado $p^*$.
  - Izquierda: error de predicción en la muestra de prueba para una sola partición
  - Derecha: error de predicción en la muestra de prueba para varias particiones
  - Hay un montón de variabilidad. (Necesitamos algo mas estable)

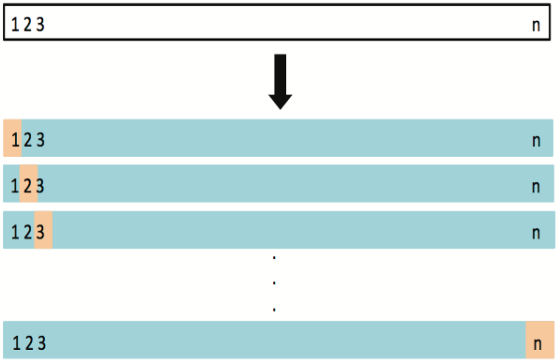


# Enfoque de conjunto de validación

- Ventajas:
  - Simple
  - Fácil de implementar
- Desventajas:
  - El MSE de validación (prueba) puede ser altamente variable
  - Solo se utiliza un subconjunto de observaciones para ajustar el modelo (datos de entrenamiento). Los métodos estadísticos tienden a funcionar peor cuando se entrenan con pocas observaciones

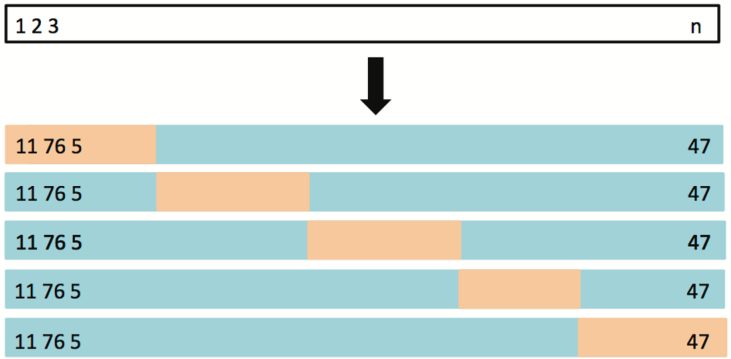
# Leave-One-Out Cross Validation (LOOCV)

- Este método es similar al enfoque de validación, pero trata de abordar las desventajas de este último.



# Validación cruzada en K-partes

- LOOCV es computacionalmente intensivo, por lo que podemos ejecutar k-fold Cross Validation



# Validación cruzada en K-partes

- Dividir los datos en K partes ($N = \sum_{j=1}^{K} n_j$)
- Ajustar el modelo dejando afuera una de las partes (folds) $\rightarrow f_{-k}(x)$
- Calcular el error de predicción en la parte (fold) que dejamos afuera

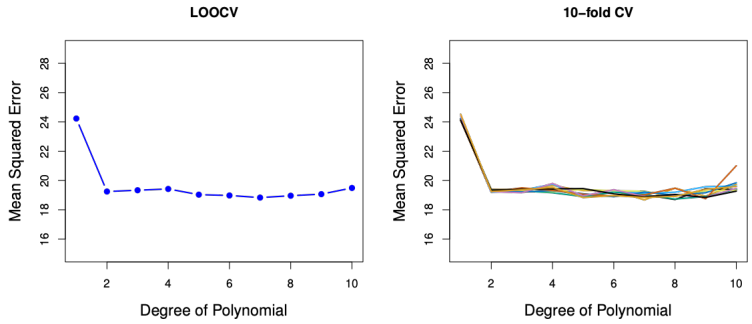
$$MSE_j = \frac{1}{n_j} \sum (y_j^k - \hat{y}_{-j})^2 \tag{19}$$

- Promediar

$$CV_{(k)} = \frac{1}{k} \sum_{j=1}^{k} MSE_j \tag{20}$$

# Validación cruzada en K-partes

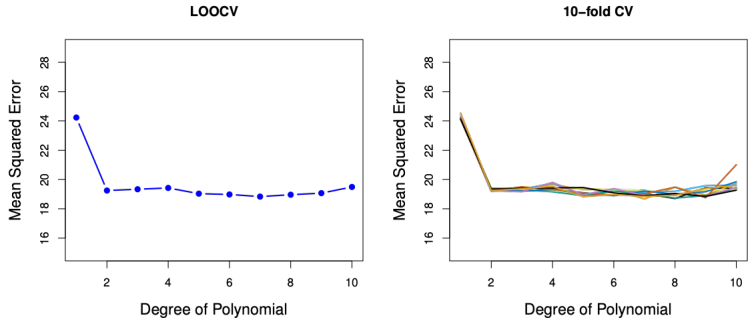
- Izquierda: LOOCV error
- Derecha: 10-fold CV
- LOOCV es caso especial de k-fold, donde k = n
- Ambos son estables, pero LOOCV (generalmente) es mas intensivo computacionalmente!



# Validación cruzada en K-partes para selección de modelos

- Supongamos que $\alpha$ parametriza la complejidad del modelo (en nuestro ejemplo el grado del polinomio)
- Primero calculamos el CV error para un grupo de modelos ($\alpha$), y elegimos el mínimo

$$\min_{\alpha} CV_{(k)}(\alpha) \tag{21}$$



# Trade-off Sesgo-Varianza para validación cruzada en K-partes

- Sesgo:
  - El enfoque del conjunto de validación tiende a sobreestimar el error de predicción en la muestra de prueba (menos datos, peor ajuste)
  - LOOCV, agrega más datos $\rightarrow$ menos sesgo
  - K-fold un estado intermedio
- Varianza:
  - LOOCV promediamos los resultados de n modelos ajustados, cada uno está entrenado en un conjunto casi idéntico de observaciones $\rightarrow$ altamente correlacionado
  - K partes esta correlación es menor, estamos promediando la salida de k modelo ajustado que están algo menos correlacionados
- Por lo tanto, existe un trade-off
  - Tendemos a usar k-fold CV con (K = 5 y K = 10)
  - Se ha demostrado empíricamente que producen estimaciones del error de prediccion que no sufren ni de un sesgo excesivamente alto ni de una varianza muy alta Kohavi (1995)

# Para seguir leyendo


- Davidson, R., & MacKinnon, J. G. (2004). Econometric theory and methods (Vol. 5). New York: Oxford University Press.
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112, p. 18). New York: springer.
- Friedman, J., Hastie, T., & Tibshirani, R. (2001). The elements of statistical learning (Vol. 1, No. 10). New York: Springer series in statistics.
- Murphy, K. P. (2012). Machine learning: a probabilistic perspective. MIT press.
- Rosen, S. (1974). Hedonic prices and implicit markets: product differentiation in pure competition. Journal of political economy, 82(1), 34-55.

# Volvemos en 5 min con `Python y Webscraping`