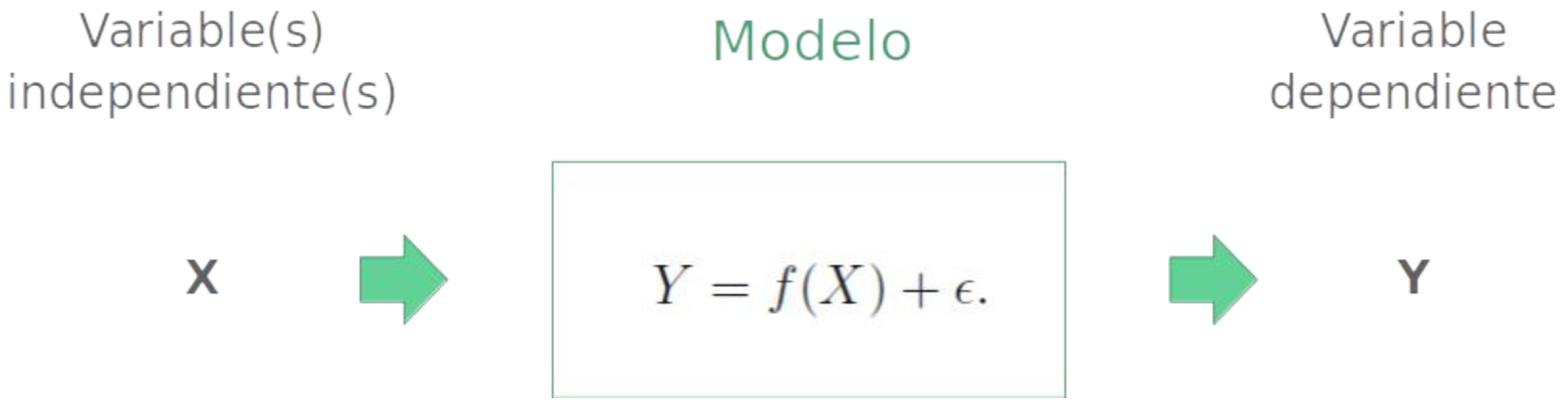
Machine Learning aplicado a las Ciencias Sociales

Clase 4. Análisis supervisado. Train, test, etc.

factor~data
EIDAES_UNSAM

Repaso… ¿qué era un modelo (supervisado)?

- Modelizar es construir una función **f(x)** que relacione variable(s) independiente(s) con variable dependiente



¿Para qué modelizar?

Predecir

Tenemos valores de un conjunto de variables independientes (X_1, X_2, etc.) y queremos un modelo que ***prediga*** el valor de la variable dependiente (Y).

Inferir

Queremos ***comprender la relación*** entre la variable dependiente (Y) y el conjunto de variables independientes (X_1, X_2, etc.).

Predicción

- **f(x)** funciona como una caja negra

- El error de nuestras predicciones se puede descomponer en dos partes:
  Reducible: mejorar **f(x)**
  Irreducible: ϵ (variables que no incluimos o no podemos medir)

Inferencia

- Énfasis en **f(x)**. El modelo se postula en base a supuestos sobre **f(x)**
- **f(x)** NO puede funcionar como una caja negra
- La calidad de nuestros resultados dependen de una serie de supuestos acerca de la distribución de los datos.

- Nuestro interés principal son preguntas cómo:
  ¿Qué variables X están relacionadas con Y?
  ¿Qué dirección tienen estas relaciones?
  ¿Qué efecto tiene cada variable X en la variable Y?
  ¿Las relaciones son lineales o no lineales?

Discusión…

¿Qué pasa si nuestro modelo de inferencia no predice bien?

\+ problemas para predecir por fuera del intervalo de entrenamiento

(e.g.: predecir el futuro con datos del pasado)

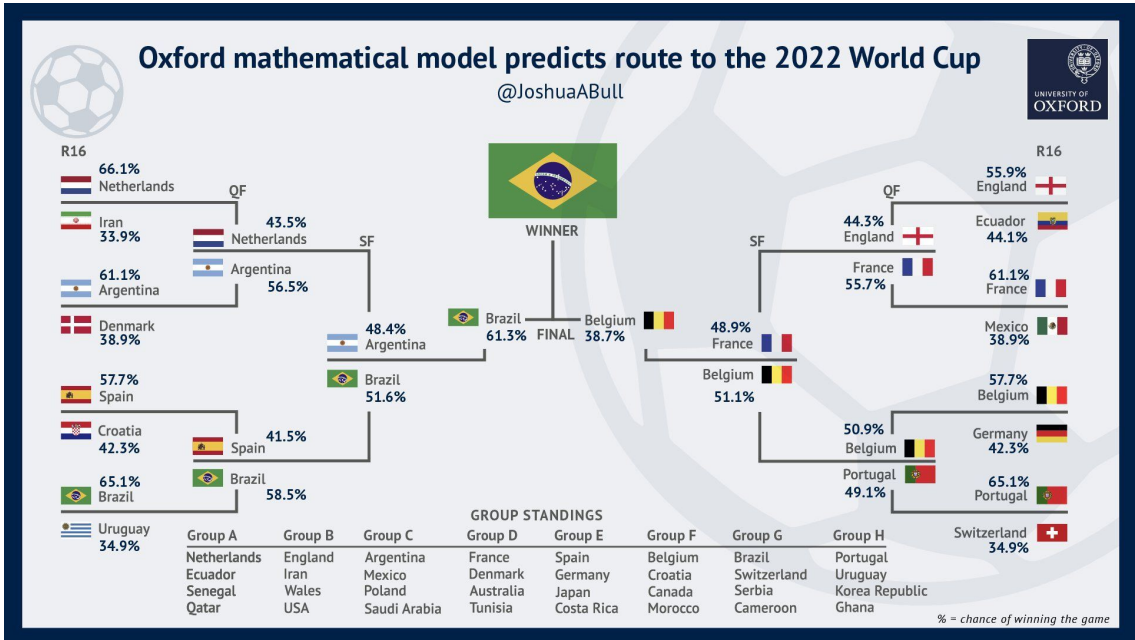
INNOVATION

The Real Reason Why Google Flu Trends Got Big Data Analytics So Wrong

Martin Willcox Brand Contributor
Teradata BRANDVOICE | Paid Program

Mar 4, 2016, 12:46pm EST

Unless you have just returned to Earth after a short break on Mars, you will have noted that some of the shine has come off the big data bandwagon lately.



factor~data
EIDAES_UNSAM

Tipos de problemas

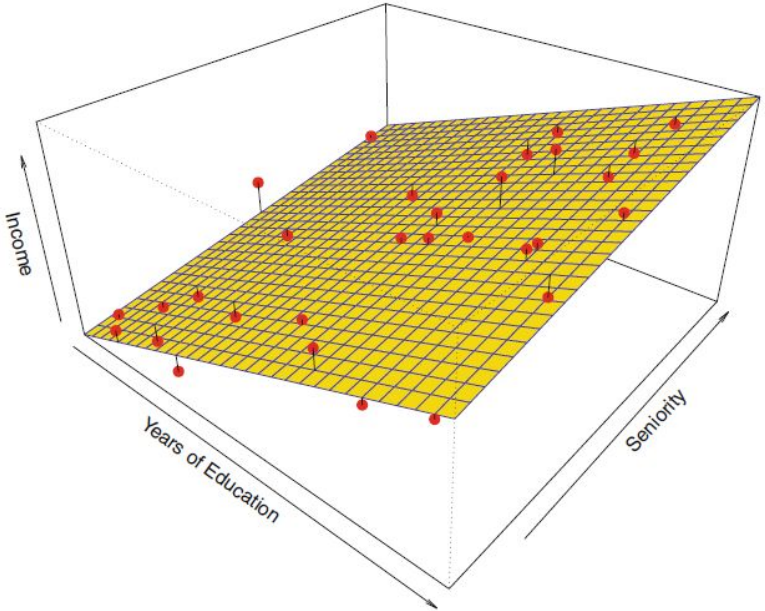
- Aprendizaje Supervisado
  - Variable dependiente => Y, resultado, target
  - Matriz de predictores (p), X, features, variables independientes, etc.
  - Problemas de regresión: Y es cuantitativa
  - Problemas de clasificación: Y es cualitativa
  - Tenemos datos de entrenamiento (conjuntos de Xi,Yi), observaciones
  - Podemos definir una (o varias) métricas para evaluar los modelos

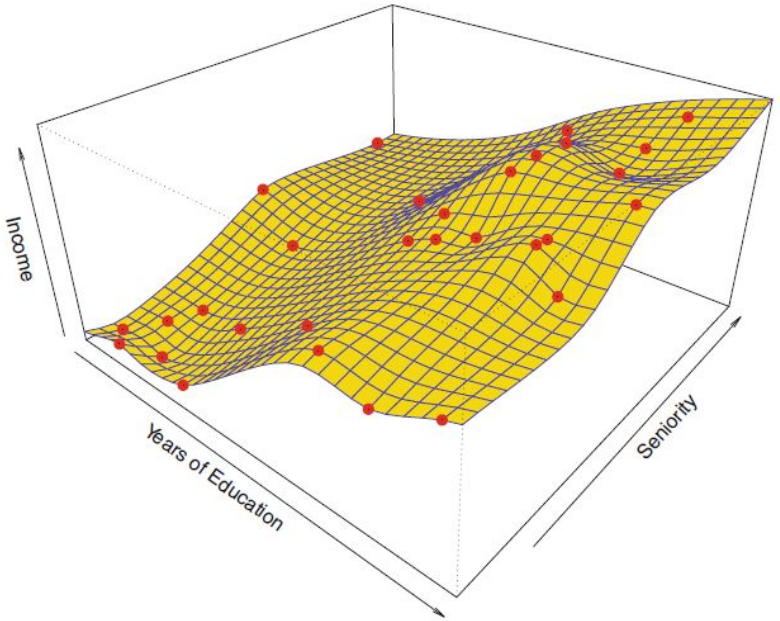
- **Aprendizaje no Supervisado**
  - **No hay variable target (Y)**
  - **Solo hay X**
  - **Es más difícil evaluar qué tan bien funciona el modelo**

Paramétrico vs. no paramétrico

Asumir la forma de f o no asumirla

Los modelos **paramétricos** asumen la forma de la función. Los métodos **no paramétricos**, no, y pueden tener diferentes formas para diferentes valores de X.





Modelos paramétricos

- **Pasos** a seguir:
  1. Asumimos la forma funcional de *f*.
  2. Ajustamos el modelo.

- **Ventajas**
  - Asumir una forma funcional para *f* simplifica la estimación de los parámetros.

- **Desventajas**
  - La forma funcional que elegimos difícilmente se ajusta a la forma real de *f* .

Modelos no paramétricos

- **Ventajas**
  - Al no asumir una forma funcional para f, tienen el potencial de ajustarse con **precisión** a una gama más amplia de formas posibles para *f*.

- **Desventajas**
  - Necesitamos una **cantidad de observaciones** mayor que en el caso de los modelos paramétricos.
  - Peligro de **overfitting**.

Trade-off precisión-interpretabilidad

¿Precisión a cualquier costo?

Modelos más precisos y menos restrictivos (no paramétricos, no lineales o que incorporen un alto número de variables independientes) reducen la interpretabilidad de las relaciones. Es necesario evaluar si la ganancia marginal en precisión compensa ese costo (además de la mayor posibilidad de *overfitting*).

Trade-off sesgo-varianza

Buscar el mejor equilibrio para cada modelo

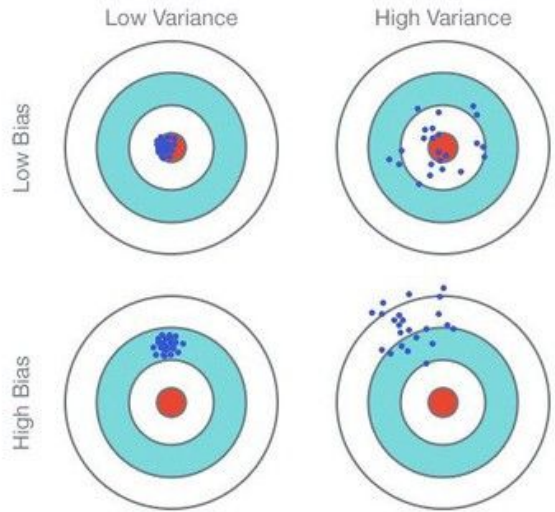


Fig. 1: Graphical Illustration of bias-variance trade-off , Source: Scott Fortmann-Roe., Understanding Bias-Variance Trade-off

Varianza: cantidad que variaría la estimación de *f* si usarámos otro test de entrenamiento. Métodos **más flexibles** tienen **más varianza**.

Sesgo: cualidad del modelo de sistemáticamente subestimar o sobreestimar el valor a predecir. Métodos **menos flexibles** tienen **más sesgo**.

¿Cuán bueno es el modelo?

Medidas de la calidad del fit: set de training y set de testing

Mean Squared Error (MSE)

- Una primera medida para **evaluar nuestro modelo**:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$$

El error cuadrático medio será **menor** cuanto más **cercanas** sean nuestras predicciones a los valores reales.

Error Rate

- Una primera medida para **evaluar nuestro modelo de clasificación**:

$$Error\ Rate = \frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i)$$

Donde:

I = 1 cuando nuestro modelo haya clasificado bien $(y_i = \hat{y}_i)$

I = 0 cuando nuestro modelo haya clasificado mal $(y_i \neq \hat{y}_i)$

Muestra que % de los casos nuestro modelo pifia en la clasificación

Predicción

		Positivo	Negativo	
Valor real	Positivo	True Positive (TP)	False Negative (FN)	
	Negativo	False Positive (FP)	True Negative (TN)	

Train-Test

Lo importante: MSE en el set de testing

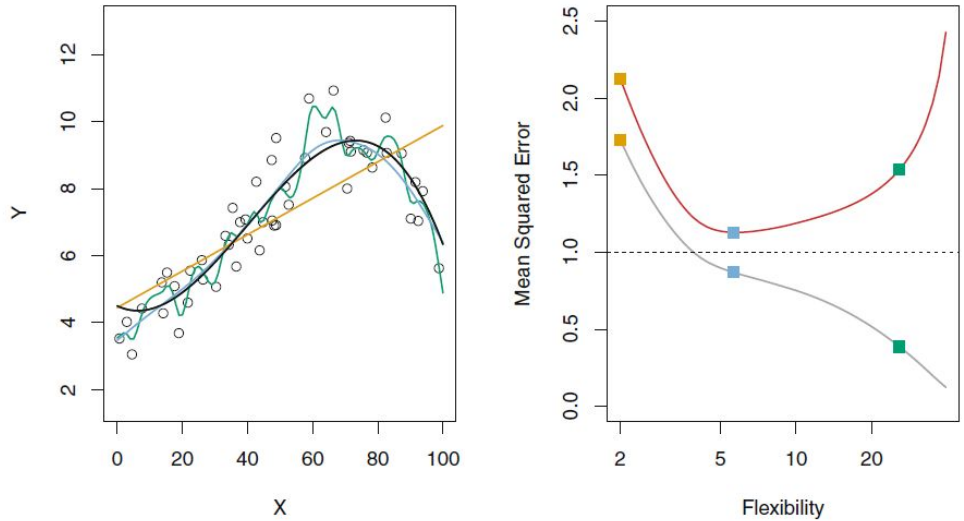
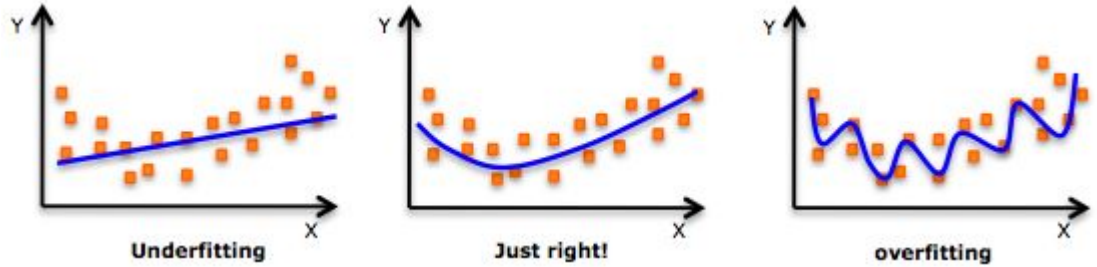


FIGURE 2.9. Left: *Data simulated from f, shown in black. Three estimates of f are shown: the linear regression line (orange curve), and two smoothing spline fits (blue and green curves).* Right: *Training MSE (grey curve), test MSE (red curve), and minimum possible test MSE over all methods (dashed line). Squares represent the training and test MSEs for the three fits shown in the left-hand panel.*

Entendamos juntes este gráfico de MSE (mean squared error) en training y testing set.

Overfitting (Repaso)



- El problema está centralmente ligado a la **predicción.**
- Nuestro interés no es que el modelo tenga buenas métricas en los datos con los que se entrenó, sino que sea bueno para predecir datos nuevos
- Overfitting se puede producir por varias causas. Ej: polinomios de muy alto grado, utilización de una gran cantidad de variables, valores de parámetros que otorgan mucha flexibilidad (como un K bajo en KNN)

Problema al usar un solo dataset en testing

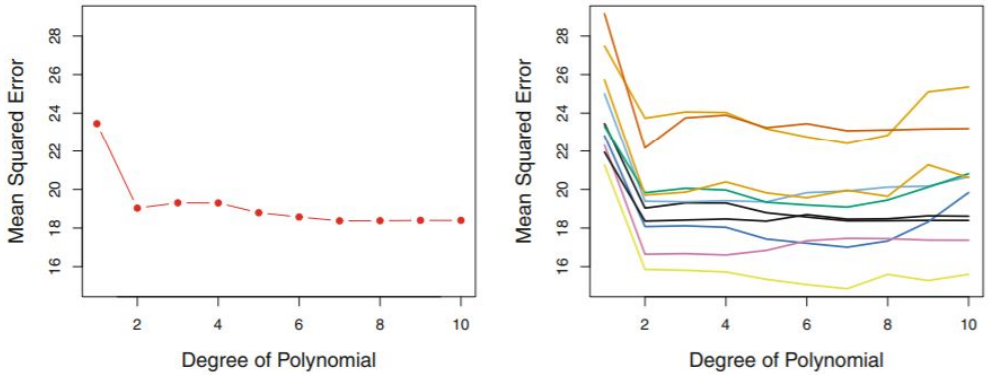
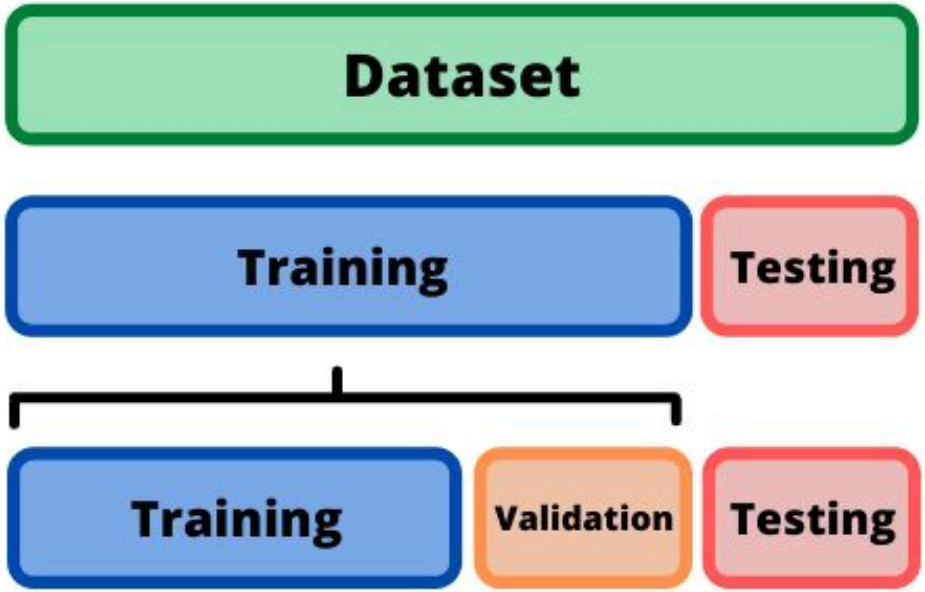


FIGURE 5.2. *The validation set approach was used on the* `Auto` *data set in order to estimate the test error that results from predicting* `mpg` *using polynomial functions of* `horsepower`. Left: *Validation error estimates for a single split into training and validation data sets.* Right: *The validation method was repeated ten times, each time using a different random split of the observations into a training set and a validation set. This illustrates the variability in the estimated test MSE that results from this approach.*

- Las métricas calculadas sobre el data set de testing estarán muy ligadas a los datos que aleatoriamente quedaron seleccionados en él

- El gráfico de la derecha muestra que, para un mismo dataset, al cambiar el ***subset*** de testing se pueden obtener diferencias significativas en el MSE (se muestra el error alcanzado con modelos que utilizan polinomios de distintos grados)

Train-test split



- Técnica para lidiar con el overfitting: Se evalua la calidad de las predicciones sobre datos no utilizados para entrenar el modelo. ¿En qué consiste?
- Se divide el dataset (usualmente 80% - 20%) utilizando una parte para entrenamiento del modelo y otra para testeo
- Se estiman las métricas de calidad de las predicciones en la base de testeo (es decir sobre casos que no fueron utilizados para entrenar el modelo)
- En algunos casos, se recomienda dejar parte del *subset* de training para realizar operaciones de validación. Esto es para ajustar algún parámetro si es que el modelo lo tiene (Ejemplo el valor **K** en KNN)

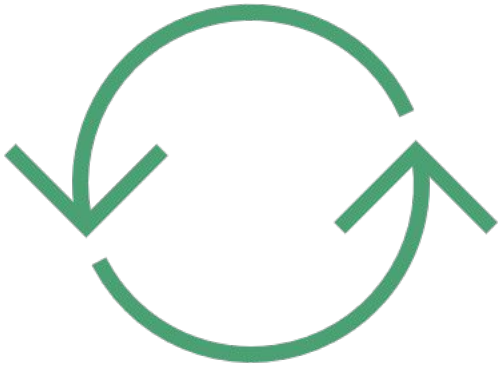
K-fold cross validation

- Se define un **K** (usualmente 5 o 10, por convención)
- Se divide el dataset en **K** particiones.
- Se realizan **K** iteraciones, usando en cada iteración una partición distinta como *subset de* testing para las restantes
- Se calcula un promedio sobre las métricas de error (u otras) que arroja cada iteración

tidymodels

Flujo de trabajo

1. Data cleaning
2. Understanding the data (exploratory data analysis)
3. Feature engineering
4. Model tuning and selection
5. Model evaluation



¿Por qué tidymodels y no R-Base?

- Fácil de **entender**: condición necesaria para trabajar en equipo.
- **Continuidad** con la sintaxis tidyverse.
- Mantener **estructuras de datos** (dataframe).

- Fundamental: **Pipes** (%>%) para encadenar secuencias complejas

Ejemplo. Dos formas de escribir:

```
small_mtcars <- slice(arrange(mtcars, gear), 1:10)

small_mtcars <- mtcars %>%
    arrange(gear) %>%
    slice(1:10)
```

Vamos al Notebook