Afi Escuela de Finanzas

Técnicas de remuestreo

Máster en Data Science y Big Data en Finanzas (MDSF)
Máster en Data Science y Big Data(MDS)

Rocío Parrilla
rocio.parrilla@atresmedia.com

Diciembre de 2021

En 1987 *The New York Times* (ver aquí).

Se hizo un estudio para ver si pequeñas dosis de aspirina podrín prevenir los ataques al corazón en personas de mediana edad. Los datos del estudio se tomaron de manera eficiente mediante un estudio controlado, aleatorizado y doble ciego.
La mitad de las personas recibió una sustancia placebo y las personas se asignaron de manera aleatoria a los tratamientos.

HEART ATTACK RISK FOUND TO BE CUT BY TAKING ASPIRIN

LIFESAVING EFFECTS SEEN

Study Finds Benefit of Tablet Every Other Day Is Much Greater Than Expected

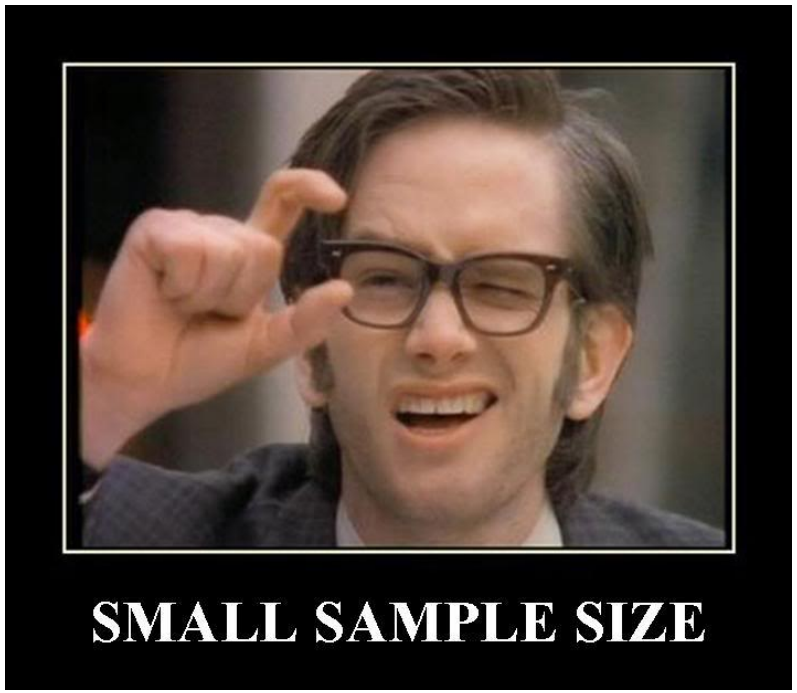
Figura: The New York Times, 1987

Tanto los sujetos como las personas que trabajaban en el estudio no sabían la identificación del tipo de las dosis ni de los pacientes.
Los estadísticos de resumen del artículo eran muy simples.

	Ataques al corazón	**No ataque**
Aspirina	104	11037
Placebo	189	11034

La razón de odds (la razón o ratio entre ambos ratios de ataques la corazón) es

$$\hat{\theta} = \frac{104/11037}{189/11034} = 0,55.$$



Según este estudio, las personas que toman aspirinas tienen casi la mitad de riesgo de sufrir un ataque al corazón.

Realmente $\hat{\theta}$ es solo un estimador del valor poblacional desconocido. La muestra parece suficientemente grande en el estudio: 22071, pero la conclusión de que la aspirina funciona bien se basa en solo 293 casos observados de ataques al corazón.

¿Se puede asegurar que se obtendría el mismo resultado si tomamos otra muestra distinta?

Para casos como el anterior, podemos usar **técnicas de remuestreo**.

Según este estudio, las personas que toman aspirinas tienen casi la mitad de riesgo de sufrir un ataque al corazón.

Realmente $\hat{\theta}$ es solo un estimador del valor poblacional desconocido. La muestra parece suficientemente grande en el estudio: 22071, pero la conclusión de que la aspirina funciona bien se basa en solo 293 casos observados de ataques al corazón.

¿Se puede asegurar que se obtendría el mismo resultado si tomamos otra muestra distinta?

Para casos como el anterior, podemos usar **técnicas de remuestreo**.

El remuestreo es el método que consiste en extraer muestras repetidas de la muestra de datos originales.

El remuestreo es el método que consiste en extraer muestras repetidas de las muestras de datos originales.

En estadística, se considera **remuestreo** a toda una variedad de métodos para:

- Estimar la precisión de estadísticos muestrales utilizando subconjuntos de la muestra disponible (*jacknife*) o tomando submuestras con reemplazamiento de tal muestra (*bootstrap*).
- Validar modelos empleando subconjuntos aleatorios de la muestra disponible (*bootstrap* y *cross-validation*).

El remuestreo es el método que consiste en extraer muestras repetidas de las muestras de datos originales.

En estadística, se considera **remuestreo** a toda una variedad de métodos para:

- Estimar la precisión de estadísticos muestrales utilizando subconjuntos de la muestra disponible (*jacknife*) o tomando submuestras con reemplazamiento de tal muestra (*bootstrap*).
- Validar modelos empleando subconjuntos aleatorios de la muestra disponible (*bootstrap* y *cross-validation*).

En otras ocasiones empleamos técnicas de remuestreo para modificar la distribución inicial de un set de datos con clases desbalanceadas.

Algunas de las más importantes:

- **Oversampling**: Consiste en modificar la distribución de los datos incrementando el número de casos de la clase minoritaria.
- **Undersampling**: Consiste en modificar la distribución de los datos reduciendo el número de casos de la clase mayoritaria.
- **Algoritmos híbridos**: Se combinan las técnicas de undersampling y oversampling.

En otras ocasiones empleamos técnicas de remuestreo para modificar la distribución inicial de un set de datos con clases desbalanceadas.

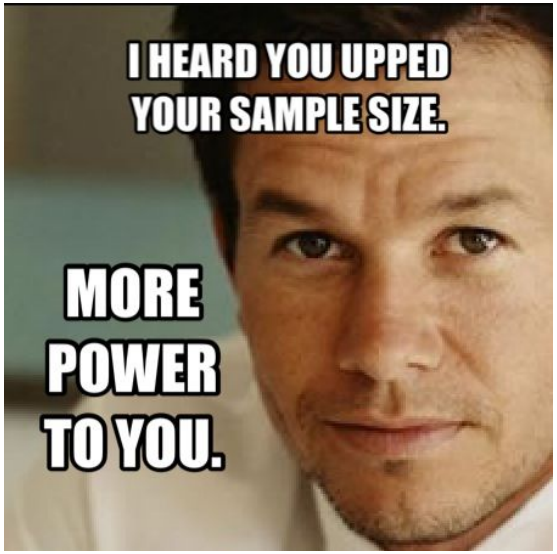
Algunas de las más importantes:

- **Oversampling**: Consiste en modificar la distribución de los datos incrementando el número de casos de la clase minoritaria.
- **Undersampling**: Consiste en modificar la distribución de los datos reduciendo el número de casos de la clase mayoritaria.
- **Algoritmos híbridos**: Se combinan las técnicas de undersampling y oversampling.

En otras ocasiones empleamos técnicas de remuestreo para modificar la distribución inicial de un set de datos con clases desbalanceadas.

Algunas de las más importantes:

- **Oversampling**: Consiste en modificar la distribución de los datos incrementando el número de casos de la clase minoritaria.
- **Undersampling**: Consiste en modificar la distribución de los datos reduciendo el número de casos de la clase mayoritaria.
- **Algoritmos híbridos**: Se combinan las técnicas de undersampling y oversampling.



Tradicionalmente el remuestreo era muy costoso, pero hoy en día ya no tanto.

El **bootstrap** se emplea para estimar la distribución muestral de un estimador mediante *remuestreos* con reemplazo de la muestra original.

El objetivo, habitualmente, es conseguir estimaciones robustas o intervalos de confianza para parámetros cuando no se puede emplear la inferencia paramétrica clásica

- porque no se dispone de hipótesis sobre la población (o, disponiendo de ellas, su veracidad es dudosa).
- porque el cálculo de las desviaciones estándar es muy complicado.

El **jacknife** se emplea para estimar el sesgo y la varianza de un cierto estadístico a partir de una muestra, recalculando la estimación de tal estadístico dejando fuera una o varias observaciones de la muestra cada vez.

En multitud de ocasiones, la estimación jacknife del estadístico converge hacia el verdadero valor del estadístico en algún sentido, lo que justifica su utilidad.

La validación cruzada (**cross-validation**) es una variación del jacknife que se emplea para validar el ajuste de un modelo predictivo.

Un subconjunto de la muestra de datos disponible se emplea como conjunto de validación: se ajusta un modelo sobre el resto de los datos (el conjunto de *training*) y se emplea para calcular valores predichos sobre el conjunto de test. La comparación de valores reales y valores predichos sirve para estimar el rendimiento *real* del modelo.

Nota: Para fijar ideas, nos centramos en problemas de regresión, aunque todas las técnicas que vamos a ver son aplicables en problemas de clasificación.

Cuando se construye un modelo $\hat{f}$ para predecir una cierta variable Y en función de un conjunto de predictores $X = (X_1, \ldots, X_p)$, el objetivo es, casi siempre, minimizar una cierta función de error

$$E(Y, \hat{Y}) = E(Y, \hat{f}(X)).$$

Más precisamente, lo que guía la construcción del modelo (elección de parámetros, estructura, etc.) es la búsqueda de un mínimo de tal función **sobre la muestra**.

Al error que produce el modelo sobre la muestra de datos disponible se le llama **error de training**.

Naturalmente, el fin último de la construcción de un modelo no es (o no debe ser) minimizar el error de training, sino cometer el mínimo error posible sobre la población (es decir, sus predicciones sobre datos *nuevos* deben ser todo lo precisas que sea posible). A este error sobre datos nuevos se le llama **error de test**.

Minimizar el error de training no garantiza que el error de test vaya a ser pequeño. De hecho, alcanzar errores de training muy pequeños suele ser señal de sobreajuste (**overfitting**): el modelo puede quedar ajustado a unas características muy específicas de los datos de entrenamiento que no tienen relación con la función objetivo. Con esto se logra que el rendimiento del modelo sobre la muestra mejore mientras que su actuación con muestras nuevas empeora.

Para estimar el error de test, como sabemos, hay dos alternativas posibles:

- Ajustar el error de training, asumiendo que la muestra es representativa y suficientemente grande y que tiene determinadas propiedades estadísticas, para estimar el error de test.
- *Reservar* un cierto subconjunto de los datos de training, que no intervengan en la construcción del modelo, y aplicar el modelo sobre ese subconjunto.

Nos centraremos en la segunda alternativa (*cross-validation*).

La primera forma (o, más bien, un antecedente) de *cross-validation* es el uso de un único conjunto de validación.

1. Se divide, de forma aleatoria, el conjunto de datos en un conjunto de training y un conjunto de test.
2. Se ajusta el modelo sobre el conjunto de training y se mide su rendimiento sobre el conjunto de test.

Un parámetro a tener en cuenta es el tamaño relativo de las muestras de entrenamiento y test, que dependerá de la aplicación (y de factores externos).

La primera forma (o, más bien, un antecedente) de *cross-validation* es el uso de un único conjunto de validación.

1. Se divide, de forma aleatoria, el conjunto de datos en un conjunto de training y un conjunto de test.
2. Se ajusta el modelo sobre el conjunto de training y se mide su rendimiento sobre el conjunto de test.

Un parámetro a tener en cuenta es el tamaño relativo de las muestras de entrenamiento y test, que dependerá de la aplicación (y de factores externos).

La primera forma (o, más bien, un antecedente) de *cross-validation* es el uso de un único conjunto de validación.

1. Se divide, de forma aleatoria, el conjunto de datos en un conjunto de training y un conjunto de test.
2. Se ajusta el modelo sobre el conjunto de training y se mide su rendimiento sobre el conjunto de test.

Un parámetro a tener en cuenta es el tamaño relativo de las muestras de entrenamiento y test, que dependerá de la aplicación (y de factores externos).

Esta forma de validación es fácil de entender y sencilla de implementar, pero presenta dos problemas potenciales:

1. Las estimaciones del error de test pueden variar mucho dependiendo, precisamente, de qué observaciones *caigan* en el conjunto de training y cuáles en el de test.

2. Dado que sólo se emplea un subconjunto de los datos disponibles para entrenar el modelo, el error de test estimado será, probablemente, una sobreestimación del verdadero error de test del modelo.

Esta forma de validación es fácil de entender y sencilla de implementar, pero presenta dos problemas potenciales:

1. Las estimaciones del error de test pueden variar mucho dependiendo, precisamente, de qué observaciones *caigan* en el conjunto de training y cuáles en el de test.

2. Dado que sólo se emplea un subconjunto de los datos disponibles para entrenar el modelo, el error de test estimado será, probablemente, una sobreestimación del verdadero error de test del modelo.

La ***leave-one-out cross-validation*** (en adelante, `LOOCV`) intenta resolver algunos de los problemas que presenta la forma de validación anterior.

Si (x_1, y_1), (x_2, y_2), $\ldots$, (x_n, y_n) es la muestra disponible, podemos incluir una sola observación (x_i, y_i) en el conjunto de test, y emplear el resto de observaciones como conjunto de training. Si ajustamos un modelo $\hat{f}_i$ utilizando ese conjunto de training y, con él, se predice $\hat{y}_i = \hat{f}(x_i)$, se puede considerar que

$$\texttt{MSE}_i = (y_i - \hat{y}_i)^2$$

es una aproximación del `MSE` de test del modelo.

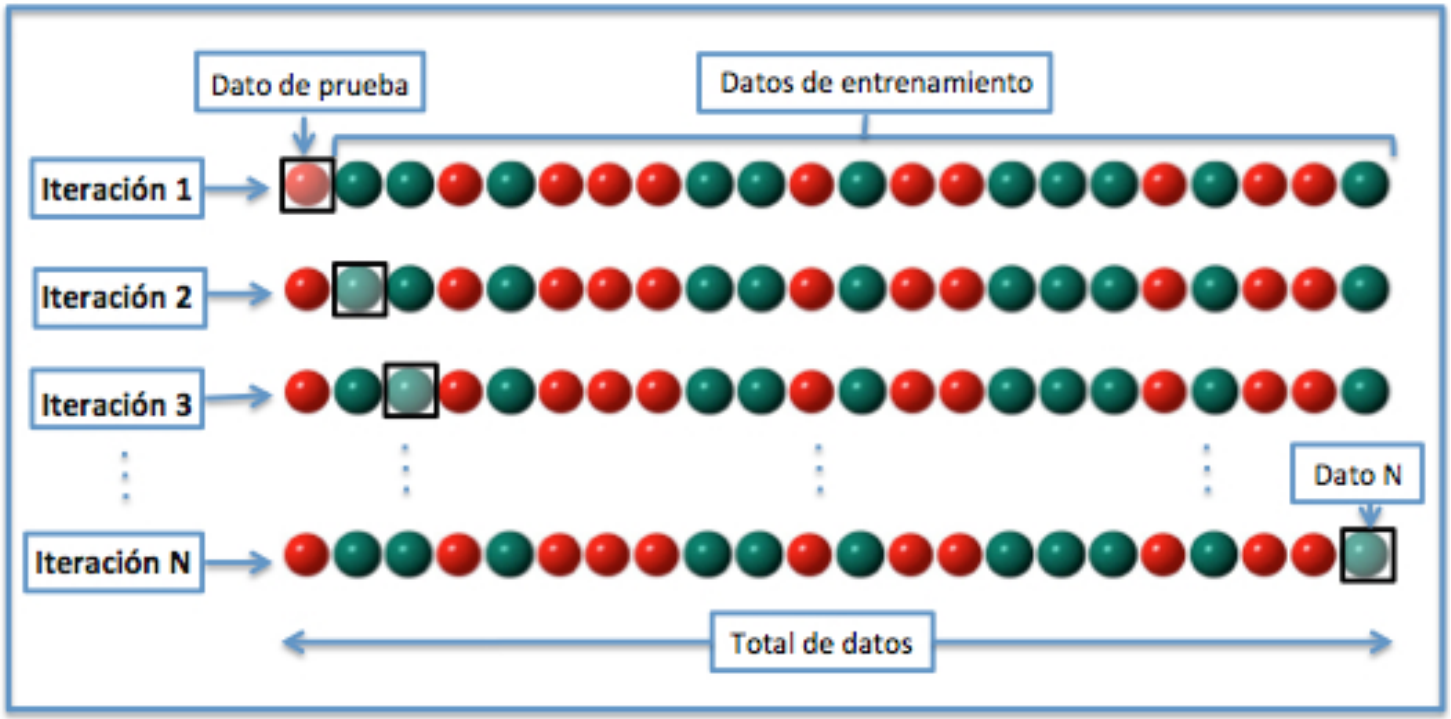


Figura: ***leave-one-out cross-validation***

De hecho, se puede probar (bajo hipótesis razonables) que $\texttt{MSE}_i$ es un estimador insesgado del $\texttt{MSE}$ de test del modelo, pero tiene una varianza muy alta (ya que depende de una única observación).

La solución natural en este caso es tomar medias: repetimos el proceso anterior para todas las observaciones (x_i, y_i), y estimamos el error de test mediante

$$\texttt{MSE}_{\texttt{LOOCV}} = \frac{1}{n}\sum_{i=1}^{n} \texttt{MSE}_i.$$

Con esto resolvemos los problemas de la primera estrategia de validación:

- La estimación del error de test no es estocástica (no incluye componente de incertidumbre, ya que se realiza con todos los puntos de la muestra).
- Los modelos se ajustan con un conjunto de datos de tamaño muy similar al total, así que el error estimado no deber sobreestimar (mucho) el error de test real. Aunque, por otro lado, podríamos hacer overfitting, ya que se acaban empleando todos los datos para ajustar el modelo. Aunque no sobreestima tanto el error de test como la validación simple.

El problema fundamental que se presenta en este caso es que esta forma de validación es muy costosa desde el punto de vista computacional.

Con esto resolvemos los problemas de la primera estrategia de validación:

- La estimación del error de test no es estocástica (no incluye componente de incertidumbre, ya que se realiza con todos los puntos de la muestra).
- Los modelos se ajustan con un conjunto de datos de tamaño muy similar al total, así que el error estimado no deber sobreestimar (mucho) el error de test real. Aunque, por otro lado, podríamos hacer overfitting, ya que se acaban empleando todos los datos para ajustar el modelo. Aunque no sobreestima tanto el error de test como la validación simple.

El problema fundamental que se presenta en este caso es que esta forma de validación es muy costosa desde el punto de vista computacional.

Con esto resolvemos los problemas de la primera estrategia de validación:

- La estimación del error de test no es estocástica (no incluye componente de incertidumbre, ya que se realiza con todos los puntos de la muestra).
- Los modelos se ajustan con un conjunto de datos de tamaño muy similar al total, así que el error estimado no deber sobreestimar (mucho) el error de test real. Aunque, por otro lado, podríamos hacer overfitting, ya que se acaban empleando todos los datos para ajustar el modelo. Aunque no sobreestima tanto el error de test como la validación simple.

El problema fundamental que se presenta en este caso es que esta forma de validación es muy costosa desde el punto de vista computacional.

Una generalización sencilla de la LOOCV que es menos costosa consiste en tomar, en cada iteración, un conjunto más grande de observaciones en lugar de una única observación en el conjunto de test.

En k***-fold cross-validation*** (en adelante, kFCV)

1. Se reparten los datos disponibles en k submuestras (*folds*) de tamaño comparable.
2. Para cada fold i, se entrena el modelo utilizando las $k-1$ submuestras restantes y se valida con ese fold como conjunto de test, y se anota el error $\texttt{MSE}_i$.
3. Se estima el error de test mediante

$$\texttt{MSE}_{k\texttt{FCV}} = \frac{1}{k}\sum_{i=1}^{k}\texttt{MSE}_i.$$

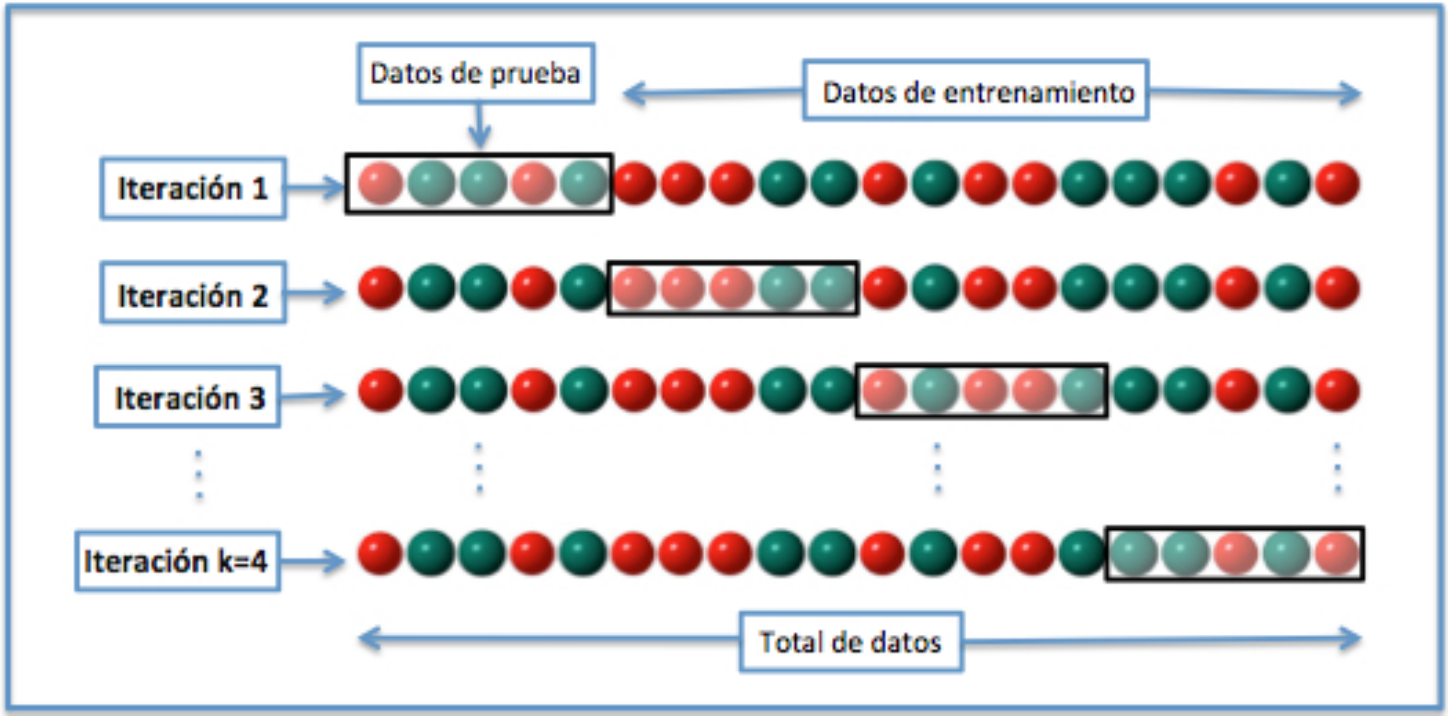


Figura: k-***fold cross-validation***

Una última alternativa, muy similar a la anterior, es la ***leave-group-out cross-validation*** (en adelante, `LGOCV`): fijados p (la proporción de datos de training) y k (el número de iteraciones):

1. Para cada $i = 1, 2, \ldots, k$, separamos de forma aleatoria los datos disponibles en una muestra de training de tamaño relativo p frente al total, y utilizamos el resto como muestra de test.
2. Ajustamos un modelo utilizando la muestra de training y validamos con la muestra de test, anotando el error $\texttt{MSE}_i$.
3. Se estima el error de test mediante

$$\texttt{MSE}_{\texttt{LGOCV}} = \frac{1}{k} \sum_{i=1}^{k} \texttt{MSE}_i .$$

Ventaja frente a LOOCV: es menos costoso computacionalmente, ya que no tiene que ajustar n modelos.

Una última alternativa, muy similar a la anterior, es la ***leave-group-out cross-validation*** (en adelante, LGOCV): fijados p (la proporción de datos de training) y k (el número de iteraciones):

1. Para cada $i = 1, 2, \ldots, k$, separamos de forma aleatoria los datos disponibles en una muestra de training de tamaño relativo p frente al total, y utilizamos el resto como muestra de test.
2. Ajustamos un modelo utilizando la muestra de training y validamos con la muestra de test, anotando el error $\texttt{MSE}_i$.
3. Se estima el error de test mediante

$$\texttt{MSE}_{\texttt{LGOCV}} = \frac{1}{k} \sum_{i=1}^{k} \texttt{MSE}_i .$$

Ventaja frente a LOOCV: es menos costoso computacionalmente, ya que no tiene que ajustar n modelos.

El método **Bootstrap** fue descrito por Efron (1979).

Normalmente, no podemos acceder a la distribución real de un proceso, pero disponemos de distintas muestras para conseguir una estimación.

La idea es usar datos para generar más datos, es decir, ser autosuficiente.

En estadística, ***bootstrapping*** se refiere a cualquier técnica o métrica que se base en muestreo aleatorio con reemplazo. Más precisamente, para estimar propiedades de un cierto estadístico dada una cierta muestra de la población (de la que no se conoce *gran cosa*):

1. Se consideran B muestras tomadas de la muestra original, del mismo tamaño que ella, y con reemplazo.
2. Se calcula el valor del estadístico sobre cada una de esas B muestras.
3. Se *estima* la distribución muestral del estadístico mediante la distribución del estadístico de las muestras de bootstrap.

Recomendación: Leer esto y esto.

En estadística, ***bootstrapping*** se refiere a cualquier técnica o métrica que se base en muestreo aleatorio con reemplazo. Más precisamente, para estimar propiedades de un cierto estadístico dada una cierta muestra de la población (de la que no se conoce *gran cosa*):

1. Se consideran B muestras tomadas de la muestra original, del mismo tamaño que ella, y con reemplazo.
2. Se calcula el valor del estadístico sobre cada una de esas B muestras.
3. Se *estima* la distribución muestral del estadístico mediante la distribución del estadístico de las muestras de bootstrap.

Recomendación: Leer esto y esto.

En estadística, ***bootstrapping*** se refiere a cualquier técnica o métrica que se base en muestreo aleatorio con reemplazo. Más precisamente, para estimar propiedades de un cierto estadístico dada una cierta muestra de la población (de la que no se conoce *gran cosa*):

1. Se consideran B muestras tomadas de la muestra original, del mismo tamaño que ella, y con reemplazo.
2. Se calcula el valor del estadístico sobre cada una de esas B muestras.
3. Se *estima* la distribución muestral del estadístico mediante la distribución del estadístico de las muestras de bootstrap.

Recomendación: Leer esto y esto.

En estadística, ***bootstrapping*** se refiere a cualquier técnica o métrica que se base en muestreo aleatorio con reemplazo. Más precisamente, para estimar propiedades de un cierto estadístico dada una cierta muestra de la población (de la que no se conoce *gran cosa*):

1. Se consideran B muestras tomadas de la muestra original, del mismo tamaño que ella, y con reemplazo.
2. Se calcula el valor del estadístico sobre cada una de esas B muestras.
3. Se *estima* la distribución muestral del estadístico mediante la distribución del estadístico de las muestras de bootstrap.

Recomendación: Leer esto y esto.

Referencias:

- A. Luntz and V. Brailovsky. On estimation of characters obtained in statistical procedure of recognition (in russian). Technicheskaya Kibernetica, 3, 1969.
- James, G., Witten, D., Hastie, T., Tibshirani, R. (2013). *An Introduction to Statistical Learning with Applications in R.* Springer. ISBN: 978-1-4614-7138-7.
- Hastie, T., Tibshirani, R., Friedman, J. (2009). *The elements of Statistical Learning. Data Mining, Inference and Prediction.* Springer. ISBN: 978-0-387-84858-7.