Exploración y Curación de Datos

EyCD 2024

Datos Ruidosos: Datos Erróneos y datos faltantes


MISSING DATA
MISSING DATA EVERYWHERE


Remove rows with NaN values
Replacing NaN Values with zeros
GFG


KNNIMPUTER FILLING THE MISSING DATA WITH A SINGLE VALUE

¿Qué vimos?

- ✓ Probabilidad y Estadística
- ✓ Análisis y Visualización de la información.
- ✓ Base de datos

¿Qué vamos a ver?

- Datos ruidosos ¿Qué hacer con los datos faltantes?
- Sesgos ¿Qué puede pasar si no tengo la muestra de datos suficiente?
- Preparamos nuestros datos para lo que viene ¿Qué tipos de transformaciones vamos hacer a los datos? y por qué?

Indice de temas

Indice de temas

1.

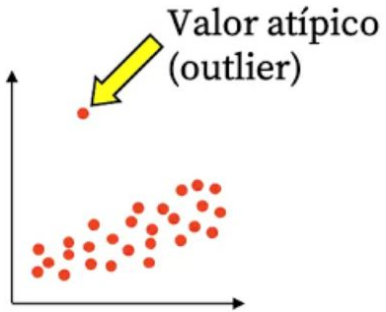
DATOS RUIDOSOS

En cualquier tipo de comunicación, el ruido es algo que hay que evitar, ya que ensucia/contamina el mensaje que se está transmitiendo.

¿Qué puede pasar si no hacemos curación de los datos?

Tipos de datos ruidosos

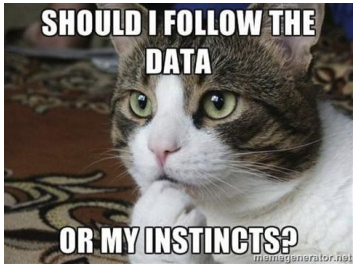
- Datos Ruidosos
  - Dato Erróneo
    - Atípico
    - Mal codificado



Ejemplo: un campo fecha → campo de texto

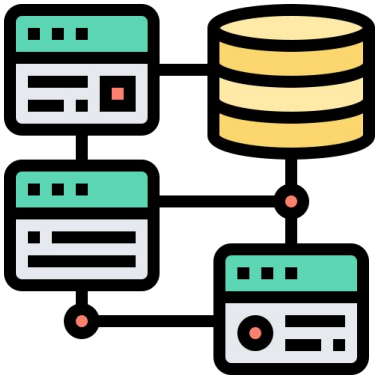
Tipos de datos ruidosos

- Datos Ruidosos
  - Dato Faltante
    - Perdidos
    - Inexistentes



Dato Erróneo ¿qué hacer?

- Ruido
  - Corregir si Mal codificado
  - Drop
- Dato Atípico
  - Modelar
  - Drop



¿Cómo trabajamos con datos erróneos?

Inspeccionamos los datos

+

Separamos datos atípicos de datos erróneamente codificados

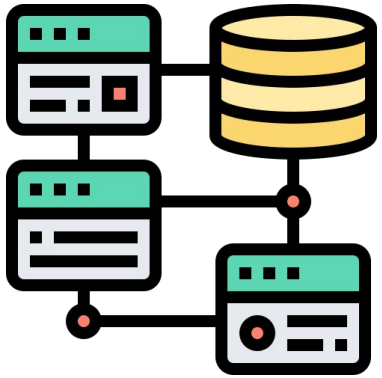
→

Decidimos

- Retirar los datos atípicos
- Retirar los erróneamente codificados
- Registrar los problemas y no tomamos acción

Dato faltante

- Perdido
  - Modelar para Imputar
  - Drop
- Inexistente
  - Nan
  - Codificación atípica



¿Cómo trabajamos con datos faltantes?

Predecir

Predecir es estimar un valor a un dato que todavía no ha sido muestreado.

+

Imputar

Imputar es es la sustitución de valores no informados en una observación por otros.

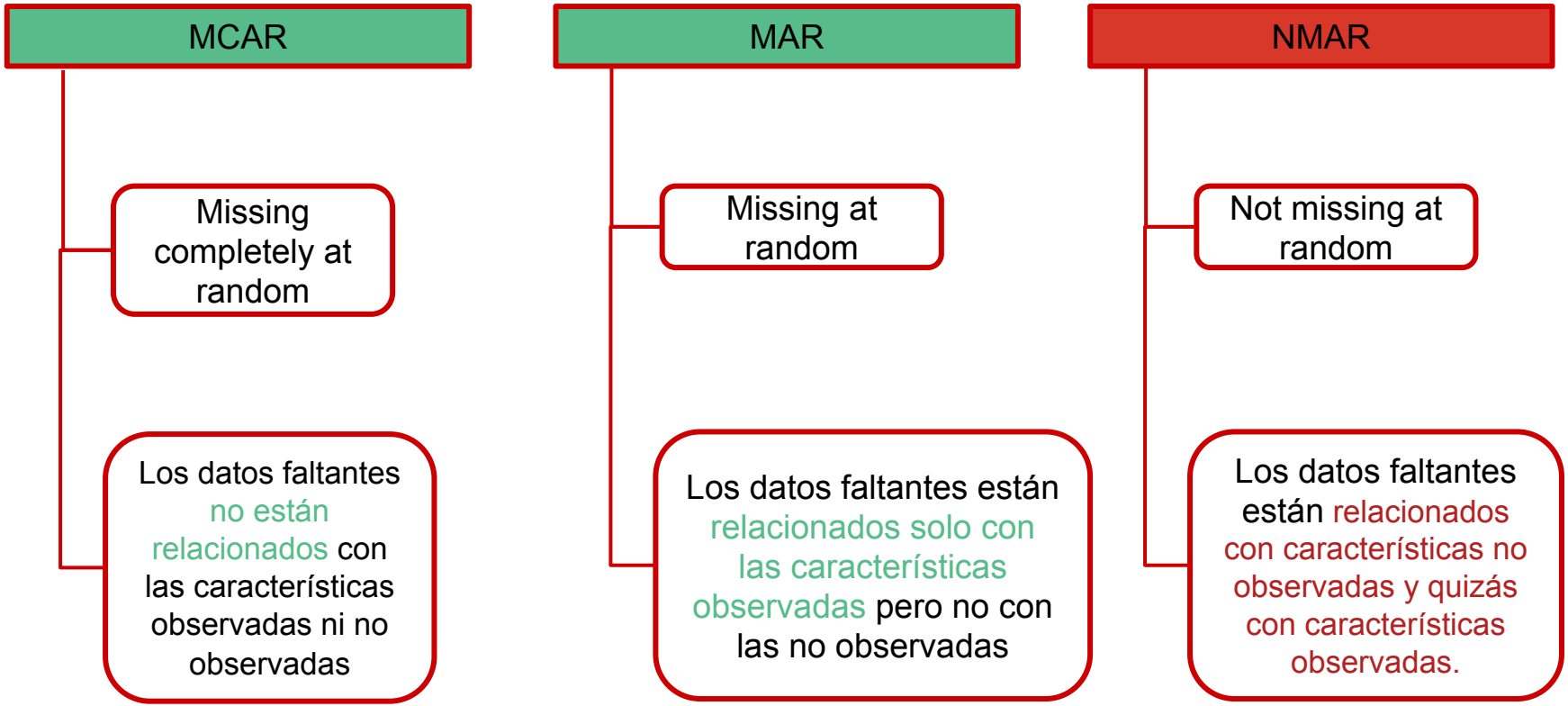
→

Si se logra realizar un modelo de predicción basado en los datos que no tienen problemas, imputar es predecir esos datos

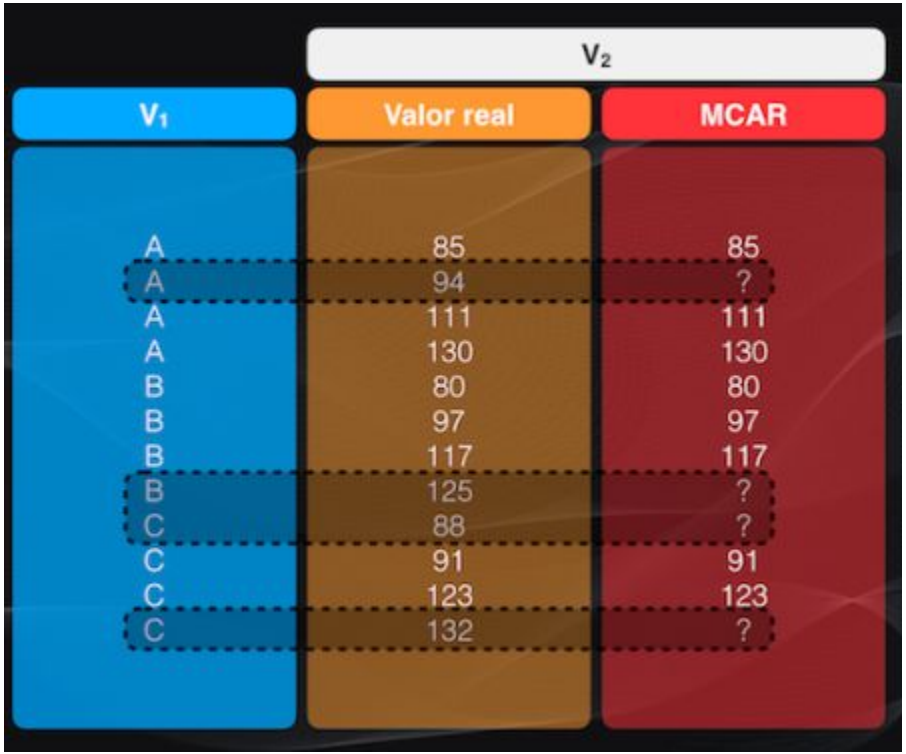
Identificar el mecanismo de los datos faltantes para ver cómo proceder!

Rubin et al. http://www.asasrms.org/Proceedings/papers/1978_004.pdf

Clasificación de los mecanismos de datos faltantes



Missing completely at random (MCAR)



Tenemos un set de datos y los datos faltantes aparecen tanto en la categoría A como en la B o en la C, y los valores faltantes pueden ser altos o bajos. Esto quiere decir que esos datos faltantes no dependen ni de la categoría ni del valor mismo de los datos, por lo que podemos decir que el mecanismo es completamente aleatorio

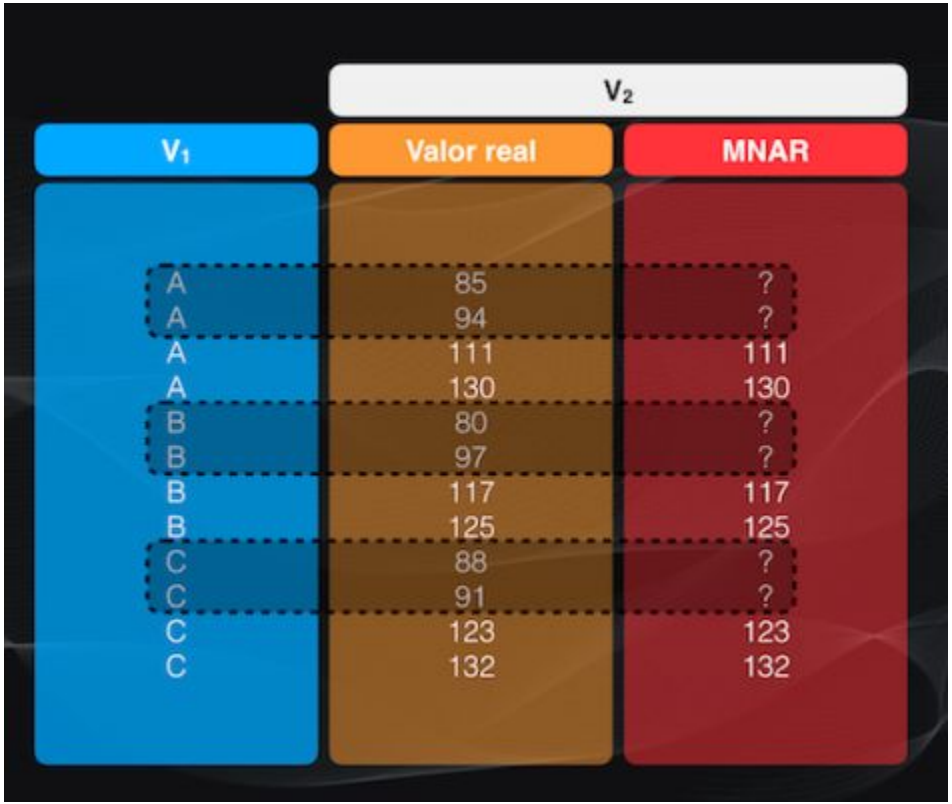
Insesgado

Missing at random (MAR)

	V_2	
V_1	Valor real	MAR
A	85	85
A	94	94
A	111	111
A	130	130
B	80	?
B	97	?
B	117	?
B	125	?
C	88	88
C	91	91
C	123	123
C	132	132

En el ejemplo vemos que los datos faltantes corresponden únicamente a datos en la categoría B, y que estos datos faltantes van desde los más pequeños a los más grandes. Esto quiere decir que los valores faltantes dependen sólo de la variable “V1” (la categoría) y no de la propia variable “V2”.

Not missing at random (NMAR)



	V_2	
V_1	Valor real	MNAR
A	85	?
A	94	?
A	111	111
A	130	130
B	80	?
B	97	?
B	117	117
B	125	125
C	88	?
C	91	?
C	123	123
C	132	132

Los datos con valores menores a 100 faltan, tanto para las categorías A, B como C. Es decir que los valores faltantes dependen de la variable “V2”, y por tanto la razón de la falta de datos NO es aleatoria

sistematicidad

Missingno: librería para explorar datos faltantes

- `pip install missingno`

https://github.com/ResidentMario/missingno

- **Bar Chart :**
  - Este gráfico de barras dá una idea de cuántos valores faltantes hay en cada columna.
- **Matrix :**
  - Con este gráfico de barras especial se puede encontrar muy rápidamente el patrón de datos faltantes en el conjunto de datos.
- **Heatmap :**
  - Este mapa visualiza la correlación de los datos faltantes entre dos columnas con un heatmap.

¿Qué vemos en estos gráficos?

A poner en práctica y probar

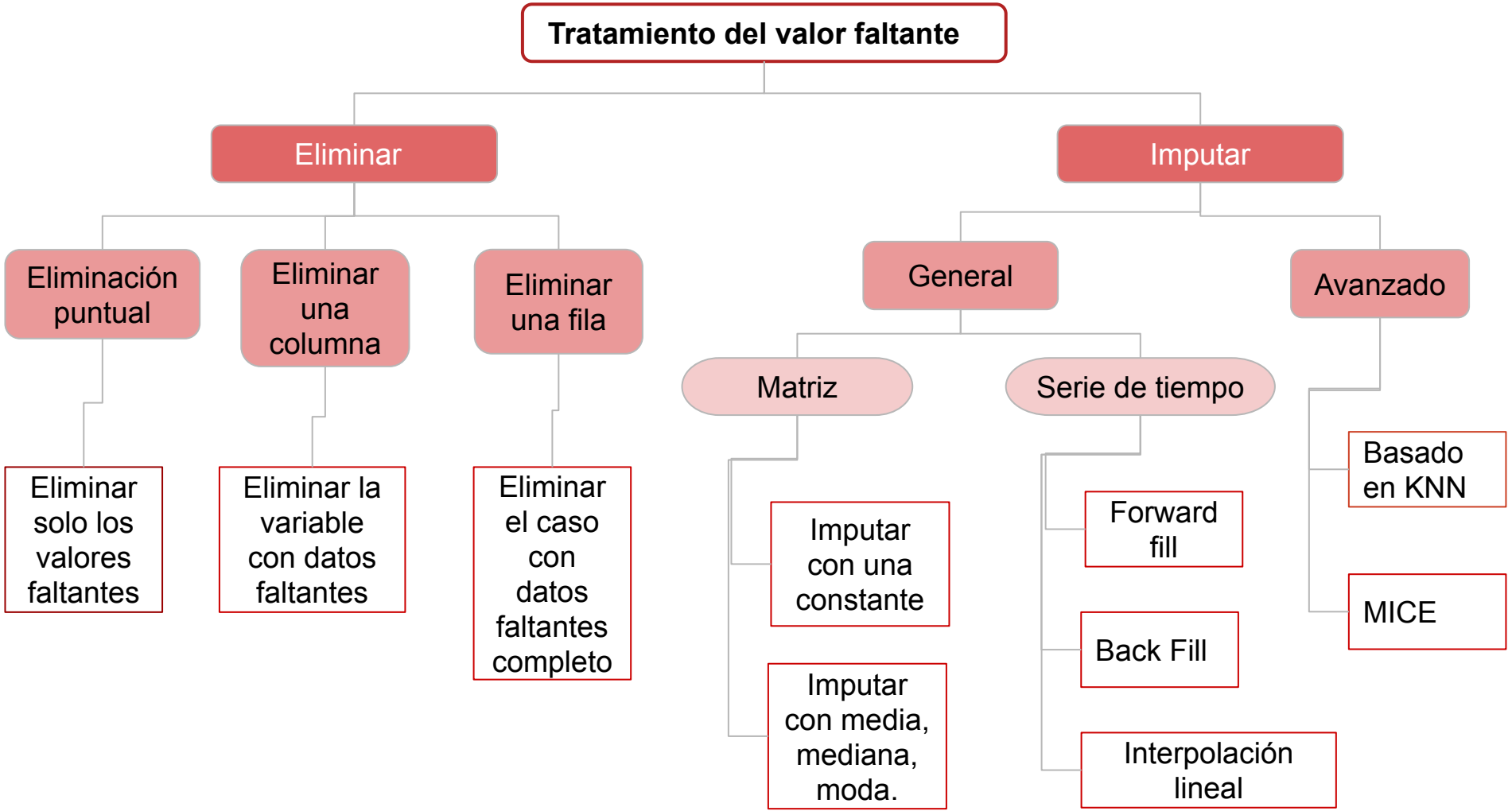
Parte 1

03_Datos_faltantes.ipynb

03_Datos_faltantes_Ejercicios.ipynb

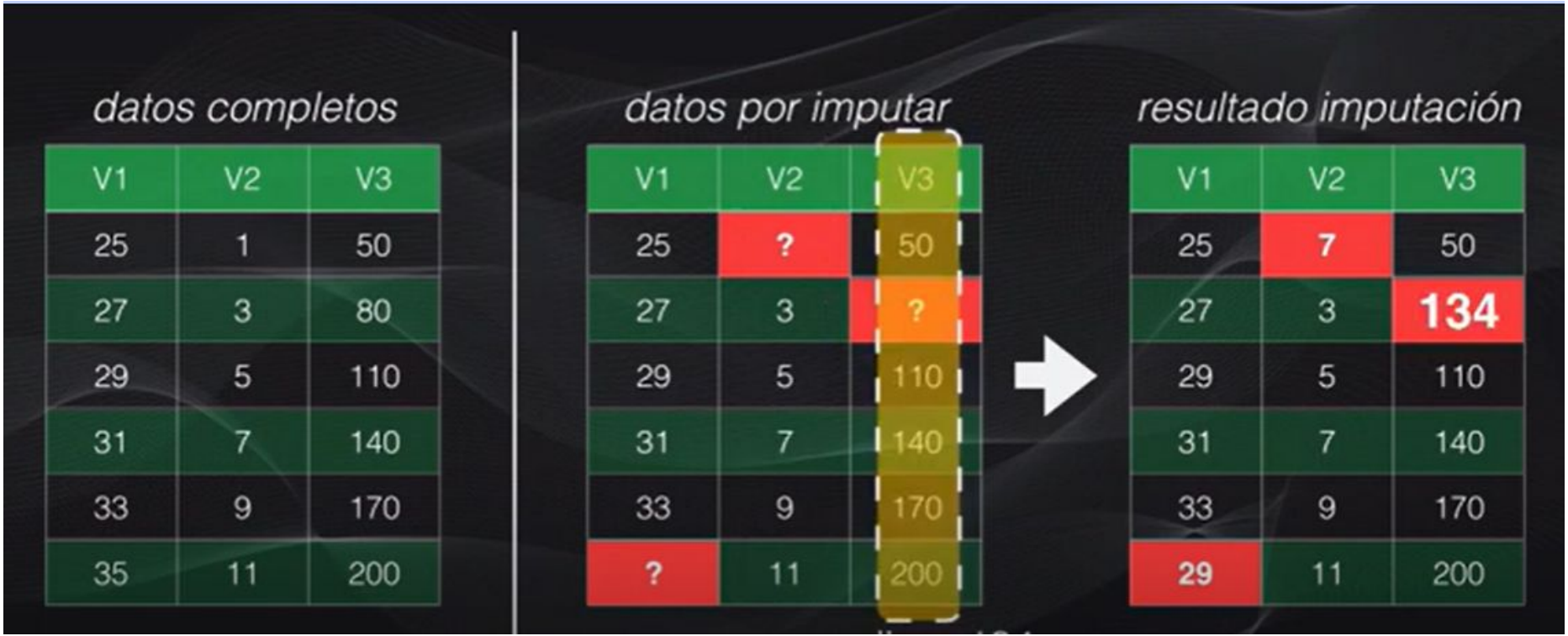
Referencia para leer:

Python: https://jakevdp.github.io/PythonDataScienceHandbook/03.04-missing-values.html
R : https://rpubs.com/dataintelligence/reemplazo_de_NAs


Tratamiento del valor faltante
Eliminar
Imputar
Eliminación puntual
Eliminar una columna
Eliminar una fila
General
Avanzado
Matriz
Serie de tiempo
Eliminar solo los valores faltantes
Eliminar la variable con datos faltantes
Eliminar el caso con datos faltantes completo
Imputar con una constante
Imputar con media, mediana, moda.
Forward fill
Back Fill
Interpolación lineal
Basado en KNN
MICE

La imputación simple - por la media o la mediana

La imputación simple - por la media o la mediana- resultado



La imputación simple - por regresión

La imputación simple - por regresión-resultado

La imputación simple - por k-vecinos más cercanos

datos completos

V1	V2	V3
25	1	50
27	3	80
29	5	110
31	7	140
33	9	170
35	11	200

datos por imputar

V1	V2	V3
25	?	50
27	3	?
29	5	110
31	7	140
33	9	170
?	11	200

La imputación simple - por k-vecinos más cercanos

KNN imputation

- K-Nearest Neighbor es un algoritmo muy utilizado para una clasificación simple.

- El algoritmo utiliza "similitud de características" para predecir los valores de cualquier nuevo punto de datos. Esto significa que al nuevo punto se le asigna un valor en función de su parecido con los puntos del conjunto de entrenamiento.

- Esto es muy útil para hacer predicciones sobre valores faltantes al encontrar los k-vecinos más cercanos a la observación con datos perdidos y luego imputarlos en función de los valores no perdidos en el vecindario.

Hay varias librerías que tienen este algoritmo

- **Fancyimpute**
- **impyute**
- **sklearn.impute**

Imputación Múltiple

La imputación múltiple es un método para llenar los valores faltantes en un conjunto de datos que considera la relación entre las diferentes variables, lo que permite realizar estimaciones más precisas y completas de los valores faltantes.

Imputación Múltiple quiebra el problema de inferencia de los valores faltantes en tres pasos:

- imputación a partir de varios subconjuntos aleatorios
- análisis
- agrupación

Imputación Múltiple

La imputación y el análisis puede realizarse siguiendo el análisis estándar, pero la combinación debe realizarse siguiendo la regla de Rubin que da la fórmula para estimar la varianza total que se compone de la varianza dentro de la imputación y la varianza entre las diferentes imputaciones.

Hay varias librerías que tienen este algoritmo, la de sklearn **sklearn.impute** permite el uso de diferentes predictores

- BayesianRidge(),
- DecisionTreeRegressor(max_features='sqrt', random_state=0),
- ExtraTreesRegressor(n_estimators=10, random_state=0),
- KNeighborsRegressor(n_neighbors=15)

MICE

- **Imputación múltiple por ecuaciones encadenadas (MICE)** es una estrategia para imputar valores faltantes modelando cada característica con valores perdidos como una función de otras características en forma rotatoria.

- Realiza regresiones múltiples sobre una muestra aleatoria de los datos, luego toma el promedio de los valores de regresión múltiple y usa ese valor para imputar el valor faltante.

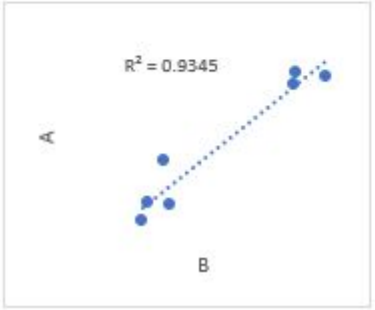
https://www.kaggle.com/code/shilongzhuang/intro-to-mice-an-imputation-strategy

La imputación múltiple: el algoritmo MICE

MICE Forest

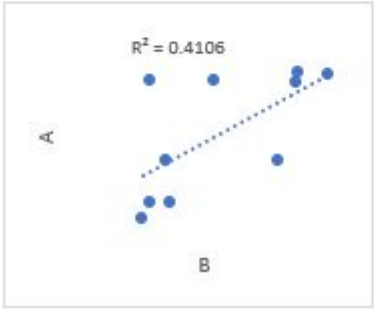
Missing data is in red. There is a strong correlation between A and B, so let's try to impute A using B and C.

A	B	C
0.93	1.40	1.53
0.24	0.46	0.76
	0.80	
0.95	1.24	1.46
0.23	0.57	
0.90		1.28
0.15	0.42	
0.47	0.54	0.63
	1.14	
0.89	1.23	1.45



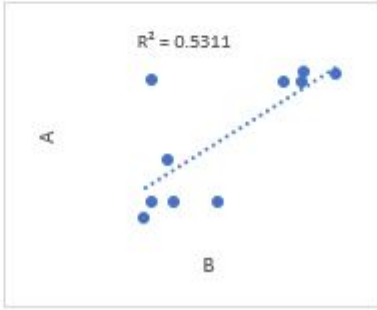
Missing data is filled in randomly. This dillutes the correlations, but allows us to impute using all available data.

A	B	C
0.93	1.40	1.53
0.24	0.46	0.76
0.90	0.80	1.53
0.95	1.24	1.46
0.23	0.57	1.28
0.90	0.46	1.28
0.15	0.42	1.53
0.47	0.54	0.63
0.47	1.14	1.28
0.89	1.23	1.45



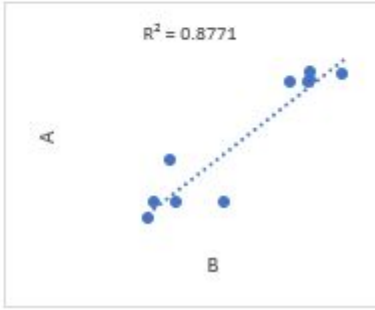
A random forest is used to predict A with B and C. Notice the correlation between A and B improved.

A	B	C
0.93	1.40	1.53
0.24	0.46	0.76
0.24	0.80	1.53
0.95	1.24	1.46
0.23	0.57	1.28
0.90	0.46	1.28
0.15	0.42	1.53
0.47	0.54	0.63
0.89	1.14	1.28
0.89	1.23	1.45



After Imputing B using A and C, we have achieved a correlation between A and B much closer to the original data.

A	B	C
0.93	1.40	1.53
0.24	0.46	0.76
0.24	0.80	1.53
0.95	1.24	1.46
0.23	0.57	1.28
0.90	1.24	1.28
0.15	0.42	1.53
0.47	0.54	0.63
0.89	1.14	1.28
0.89	1.23	1.45

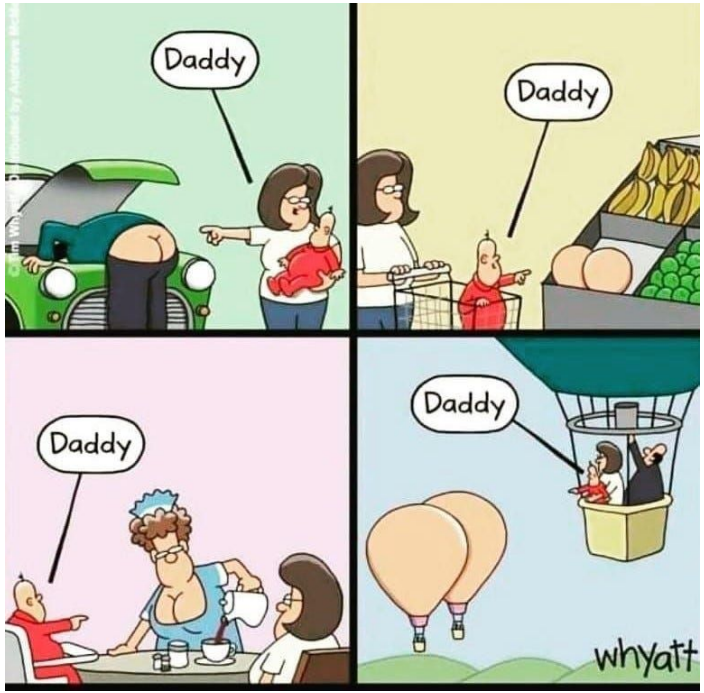


Conclusión

- No existe una forma perfecta de compensar los valores perdidos en un conjunto de datos.

- Cada estrategia puede funcionar mejor para ciertos conjuntos de datos y tipos de datos faltantes, pero puede funcionar mucho peor en otros tipos de conjuntos de datos.

- Hay algunas reglas establecidas para decidir qué estrategia usar para tipos particulares de valores perdidos, pero más allá de eso, **debe experimentar** y verificar qué modelo funciona mejor para su conjunto de datos.



A poner en práctica y probar

Parte 2

03_Datos_faltantes.ipynb

03_Datos_faltantes_Ejercicios.ipynb

Referencias:

https://www.codificandobits.com/blog/manejo-datos-faltantes/#datos-faltantes-completamente-aleatorios-o-mcar-missing-completely-at-random
https://www.codificandobits.com/blog/analisis-exploratorio-de-datos/