

# ***COMP 6315 Minería de Datos***

---

## CLASE 3: Preprocesamiento de Datos

Dr. Edgar Acuna  
Departamento de Matematicas

Universidad de Puerto Rico-  
Mayaguez

website: [academic.uprm.edu/eacuna](http://academic.uprm.edu/eacuna)

# Preparacion de datos-1

---

Preparacion de Datos = Limpieza de Datos+ Feature Engineering

La limpieza de datos prepara correctamente a los datos para el análisis.

Incluye las siguientes tareas:

- Seleccionar, Filtrar y remoción de duplicados
- Muestreo
- Particionamiento de la data para crear los conjuntos de entrenamiento de validacion y de prueba.
- Normalizacion de los datos.
- Reducción de Datos
- Integración de datos.
- Discretización (Binning)
- Imputación de los valores “missings”

# Preparacion de datos-2

---

Feature Engineering:

Mediante este proceso se selecciona los atributos correctos para aplicar el metodo de Machine learning Incluye:

- El uso de “domain knowledge” de los datos para seleccionar los mejores atributos
- Selección de Variables (Feature selection)
- Validacion de que los attributes elegidos funcionen bien con el modelo.
- Feature Extraction

# Preparacion de datos-3

---

Preparacion de Datos: Preprocesamiento de Datos + Data Wrangling (Data Munging)

De acuerdo a en que etapa del proceso del analisis se lleva a cabo la preparacion de los datos, ésta puede ser clasificada en dos:

Pre-procesamiento de datos: Es la preparación de los datos que se lleva a cabo inmediatamente despues de haber accesado a los datos. Es llevado a cabo antes de aplicar el algoritmo de Machine Learning. Es ejecutado una sola vez.

Data Wrangling: Es la preparacion de los datos durante el proceso de la construccion del modelo. En este proecso se cambia iterativamente la forma del conjunto de datos hasta que sea posible construir un buen modelo a partir del mismo

# Por qué preprocesar los datos?

---

- Los datos en el mundo real son “sucios”:
  - **incompletos**: falta valores en los atributos, carecen de algunos atributos de interés, o contienen sólo totalizaciones.
  - **ruidosos**: contienen errores o valores anómalos.
  - **inconsistentes**: contienen discrepancias en códigos o nombres ( Notas: A,AB,B,C,D,F,W).
  - **datos duplicados**.
- Si no hay calidad en los datos, no hay calidad en los resultados!
  - Las decisiones de calidad deben estar basadas en datos de calidad.
  - Para hacer Data Warehouse se necesita integrar consistentemente datos de calidad.

# Ruido

---

- El ruido se refiere a la modificación de valores originales.
- Ejemplos: distorsión de la voz de una persona cuando habla por teléfono, “nieve” en la pantalla de televisión.

**Two Sine Waves**

**Two Sine Waves + Noise**

# “Outliers”

---

- Los “outliers” son casos con características que difieren considerablemente de la mayoría de los casos en el conjunto de datos.



# Principales Tareas de Preprocesamiento de Datos

---

- Limpieza de datos
  - Completar valores faltantes, suavizar datos ruidosos, identificar o eliminar “outliers”, y resolver inconsistencias.
- Integración de datos
  - Integración de múltiples bases de datos.
- Transformación de datos
  - Normalización y totalización.
- Reducción de datos
  - Se obtiene una representación más reducida en volumen pero que produce los mismos o similares resultados analíticos.
- Discretización de datos
  - Parte de la reducción de datos pero que es particularmente importante para datos numéricos.



# Limpieza de Datos

---

- Tareas de la limpieza de datos:
  - Completar valores faltantes.
  - Identificar “outliers” y suavizar datos ruidosos.
  - Corregir datos inconsistentes.

# Preprocesamiento - Datos Faltantes

---

- Los datos no siempre están disponibles.
  - E.g., muchas filas no tienen registrados valores para muchos atributos, tales como los ingresos del cliente en datos de ventas.
- La falta de valores se puede deber a:
  - mal funcionamiento de equipos.
  - inconsistencia con otros datos registrados y por lo tanto han sido eliminados y considerados faltantes.
  - datos no ingresados debido a equivocaciones o malos entendidos.
  - algunos datos pudieron no considerarse importantes al momento de ingresar datos y fueron omitidos.
- Puede ser necesario estimar los valores faltantes.

## Datos faltantes (cont)

---

- Los valores faltantes son un problema común en análisis estadístico.
- Se ha propuesto muchos métodos para el tratamiento de valores faltantes. Muchos de estos métodos fueron desarrollados para el tratamiento de valores faltantes en encuestas por muestreo.
- Bello (1995), tratamiento de valores faltantes in regression
- Troyanskaya et al (2001), tratamiento de datos faltantes en clasificacion no supervisada.

# Datos faltantes (cont)

- Estudios relacionados con clasificación supervisada:
  - Chan and Dunn (1972) – Imputation en LDA para problemas con dos clases.
  - Dixon (1975) – Imputacion k-nn para lidiar con valores faltantes en clasificacion supervisada.
  - Tresp (1995)- el problema de valores faltantes en aprendizaje supervisado usando redes neurales.
- Impacto de valores faltantes:
  - 1% datos faltantes – trivial
  - 1-5% - manejable
  - 5-20% - requiere métodos sofisticados
  - 20% o mas- efecto perjudica las interpretaciones



# Mecanismos de valores faltantes

---

- Hay tres mecanismos de valores que tratan de explicar las razones que causan que hayan datos faltantes:
- i) Valores faltantes completamente al azar (*MCAR*): La probabilidad que una instancia tenga un valor faltante para un atributo es la misma para todas las instancias. Es decir, esta probabilidad no depende ni de los valores observados ni de los valores faltantes. La mayoría de los valores faltantes no son MCAR.

Este mecanismo es mas adecuado para datos a ser usados en clasificacion no supervisada.

Asimismo es adecuado para conjuntos de datos con un bajo porcentaje de valores faltantes.

# Mecanismos de valores faltantes

---

- ii) *Valores faltantes al azar (MAR)*: La probabilidad que una instancia tenga un valor faltante en un atributo depende de los valores observados, como por ejemplo la clase a la cual pertenece la instancia, pero no depende de los valores faltantes. Este mecanismo es mas adecuado para datos usados en clasificacion supervisada. Asimismo para simular datos faltantes.
- iii) *Valores faltantes no al azar o no ignorables(NMAR)*: La probabilidad de que una instancia tenga un valor faltante en un atributo depende de los valores faltantes en el conjunto de datos. Ocurre cuando las personas entrevistadas no quieren revelar algo muy personal acerca de ellas. Este tipo de valores faltantes es el mas dificil de tratar y es el que ocurre mas frecuentemente. Se considera cuando hay una gran cantidad de datos faltantes

# Conjunto de datos Census

---

Tambien conocido como **Adult**.

48842 instancias, contiene variables continuas , ordinales y nominales (entrenamiento=32561, prueba=16281).

Cuando se eliminan las instancias con valores faltantes quedan 45222 (entrenamiento=30162, prueba=15060).

Tamano=3.8MB (entrenamiento), 1.9MB(prueba)

Disponible en: <http://archive.ics.uci.edu/ml/>

Donantes: Ronny Kohavi y Barry Becker (1996).

# Variables en Census

---

- 1- age: continua.
- 2- workclass: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked. Nominal
- 3- fnlwgt (final weight) : Continua.
- 4- education: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool. Ordinal.
- 5- education-num: continua.
- 6- marital-status: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse. Nominal
- 7- occupation: Nominal



# Variables en Census

---

- 8-relationship: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried. Nominal
- 9-race: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black. Nominal
- 10-sex: Female[0], Male[1]. Nominal-Binaria
- 11-capital-gain: continua.
- 12-capital-loss: continua.
- 13-hours-per-week: continua.
- 14-native-country: nominal
- 15 Salary: >50K [2], <=50K [1].

## Ejemplo: Conjunto de datos census

age	employe	education	edun	marital	...	job	relation	race	gender	hour	country	wealth
					...							
39	State_gov	Bachelors	13	Never_mar	...	Adm_cleric	Not_in_fam	White	Male	40	United_States	poor
51	Self_emp	Bachelors	13	Married	...	Exec_man	Husband	White	Male	13	United_States	poor
39	Private	HS_grad	9	Divorced	...	Handlers_c	Not_in_fam	White	Male	40	United_States	poor
54	Private	11th	7	Married	...	Handlers_c	Husband	Black	Male	40	United_States	poor
28	Private	Bachelors	13	Married	...	Prof_speci	Wife	Black	Female	40	Cuba	poor
38	Private	Masters	14	Married	...	Exec_man	Wife	White	Female	40	United_States	poor
50	Private	9th	5	Married_sp	...	Other_serv	Not_in_fam	Black	Female	16	Jamaica	poor
52	Self_emp	HS_grad	9	Married	...	Exec_man	Husband	White	Male	45	United_States	rich
31	Private	Masters	14	Never_mar	...	Prof_speci	Not_in_fam	White	Female	50	United_States	rich
42	Private	Bachelors	13	Married	...	Exec_man	Husband	White	Male	40	United_States	rich
37	Private	Some_coll	10	Married	...	Exec_man	Husband	Black	Male	80	United_States	rich
30	State_gov	Bachelors	13	Married	...	Prof_speci	Husband	Asian	Male	40	India	rich
24	Private	Bachelors	13	Never_mar	...	Adm_cleric	Own_child	White	Female	30	United_States	poor
33	Private	Assoc_acc	12	Never_mar	...	Sales	Not_in_fam	Black	Male	50	United_States	poor
41	Private	Assoc_voc	11	Married	...	Craft_repai	Husband	Asian	Male	40	*MissingVar	rich
34	Private	7th_8th	4	Married	...	Transport	Husband	Amer_India	Male	45	Mexico	poor
26	Self_emp	HS_grad	9	Never_mar	...	Farming_fi	Own_child	White	Male	35	United_States	poor
33	Private	HS_grad	9	Never_mar	...	Machine_c	Unmarried	White	Male	40	United_States	poor
38	Private	11th	7	Married	...	Sales	Husband	White	Male	50	United_States	poor
44	Self_emp	Masters	14	Divorced	...	Exec_man	Unmarried	White	Female	45	United_States	rich
41	Private	Doctorate	16	Married	...	Prof_speci	Husband	White	Male	60	United_States	rich
:	:	:	:	:	:	:	:	:	:	:	:	:

# Leyendo los datos en Python

---

```
import pandas as pd
df = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data',
header=None, sep=',',na_values=" ?")
df.columns=['v1', 'v2', 'v3', 'v4',v5,'v6','v7','v8','v9','v10','v11','v12','v13','v14','class']
```

Otra forma:

```
import pandas as pd
names=['v1','v2','v3','v4','v5','v6','v7','v8','v9','v10','v11','v12','v13','v14','clase']
data=pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data',
names=names, na_values=" ?")
print(data.shape)
(32562, 15)
data.describe()
```

# Python functions dealing with missing values

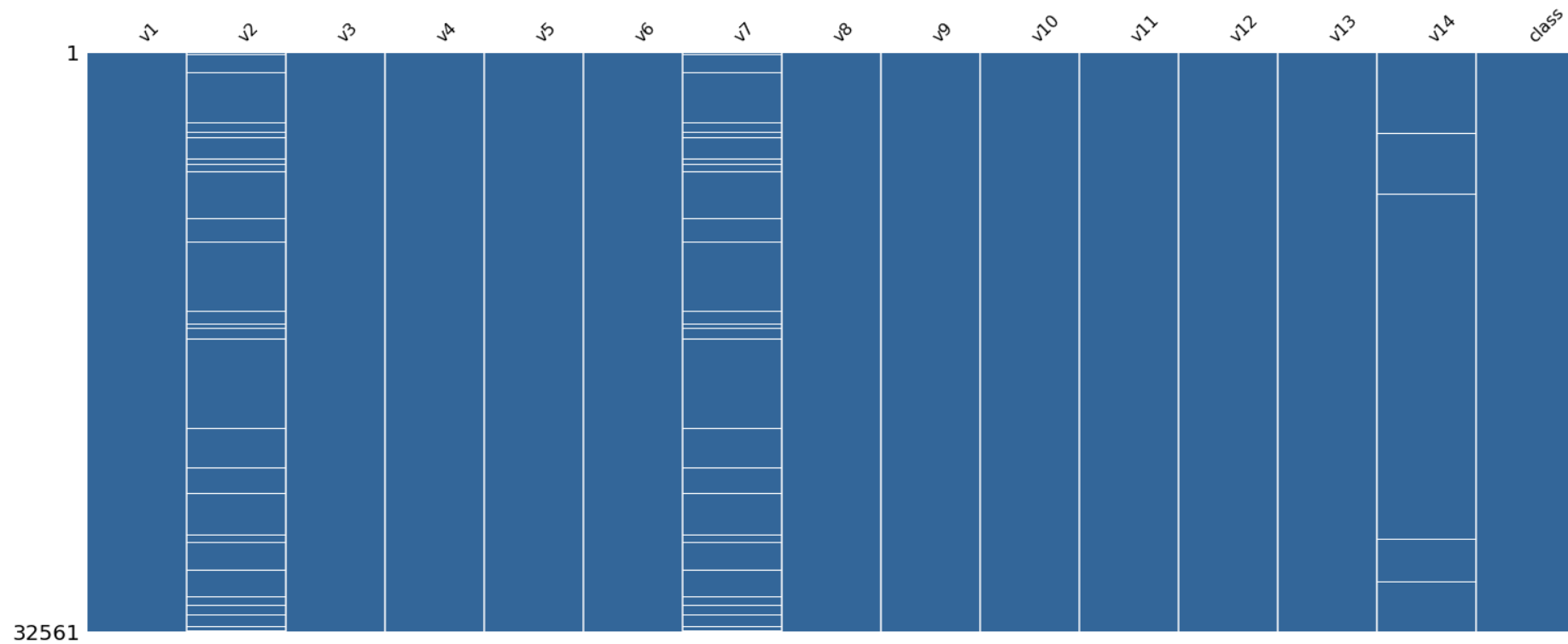
---

Assuming that df is a Pandas dataframe

- To find out columns containing missing values  
**`colmiss=df.columns[df.isnull().any()].tolist()`**
- To find out rows containing missing values  
**`rowmiss=df.index[df.isnull().T.any()].tolist()`**
- To find out the percentage of rows with missing values  
**`df.isnull().T.any().sum()*100/len(df)`**
- To find out the percentage of missing values per column  
**`df[colmiss].isnull().sum()*100/len(df)`**
- Deleting all the rows containing missing values  
`dfclean=df.dropna()`  
`print dfclean.shape`  
`(30162, 15)`

## *Visualizing the missing values*

---



## A “clean” function

---

- We can build a function for deleting columns and rows with a large amount of missing values. Something like:

**`clean(df,tol.col=.5,tol.row=.3)`**

In here, from the dataframe df, we are deleting columns with more than 50 % of missing values and rows with more than 30% of missings

# La función clean

---

- Esta función elimina columnas y filas que tienen un gran número de valores faltantes.

```
census.cl=clean(adult,tol.col=.5,tol.row=.3,name="cl.adult")
```

	Variables	Percent.of.missing
● 1	V2	5.6386474616873
● 2	V6	5.66014557292466
● 3	V13	1.79048555019809

Maximum number of values to be imputed: 4262

# Modules in Python to impute missing values

---

The module scikit-learn has a class named `Imputer` to carry out imputation.

The function `dropna` deletes rows with at least one missing value.

The function `fillna` in pandas does imputation.

Also, there are two other modules for imputation

`FancyImpute`

`impyte`

Both of them run only in Python 3.



# Tratamiento de valores faltantes

---

- **Eliminación de casos.** Ignorar la fila que contiene datos faltantes. Usualmente es aplicado cuando el valor que falta es el de la clase (asumiendo que se esta haciendo clasificación). No es efectiva cuando el porcentaje de valores faltantes por atributo varía considerablemente.
- **Técnicas de Imputación,** donde los valores faltantes son reemplazados con valores estimados basados en la información disponible en el conjunto de datos.
- **Estimación de parámetros,** donde los procedimientos de Máxima Verosimilitud que usan variantes del algoritmo EM (Expectation-Maximization) pueden manejar la estimación de parámetros simultaneamente a lcon la sustitucion de valores faltantes.

# Tratamiento de valores faltantes en clasificacion

---

Eliminacion de casos (CD) – Este método consiste en descartar todas las instancias (casos) con valores perdidos en por lo menos un atributo. Una variante de este método consiste en determinar el grado de valores faltantes en cada instancia y atributo, y eliminar las instancias y/o atributos con altos niveles de valores faltantes. Antes de eliminar cualquier atributo es necesario evaluar su relevancia en el análisis.

# Tratamiento de valores faltantes

---

- Imputacion usando la media (MI) – Reemplazar los valores faltantes de un atributo dado por la media de todos los valores conocidos de ese atributo en la clase a la que la instancia con el valor faltante pertenece. Si la variable es nominal se usa la moda en lugar de la mediana
- Imputacion usando la mediana (MDI). Como la media se ve afectada por la presencia de outliers, parece natural usar la mediana en su lugar para asegurar robustez. En este caso los valores faltantes para un atributo dado es reemplazado por la mediana de todos los valores conocidos de ese atributo en la clase a la que la instancia con el valor faltante pertenece.

`adult.mimp=ce.mimp(adult,"mean",1:14,c(2,7,14))`

`adult.mdimp=ce.mimp(adult,"median",atr=1:14,nomatr=c(2,7,14))`

# Imputación con los k-vecinos mas cercanos (KNNI)

---

- Dividir el conjunto de datos  $D$  en dos partes. Sea  $D_m$  el conjunto que contiene las instancias en las cuales falta por lo menos uno de los valores. Las demás instancias con información completa forman un conjunto llamado  $D_c$ .
  - Para cada vector  $x$  en  $D_m$ :
    - A) Dividir el vector en dos partes: una la de información observada y otra la de información faltante,  $x = [x_o; x_m]$ .
    - B) Calcular la distancia entre  $x_o$  y todos los vectores del conjunto  $D_c$ . Usar solo aquellos atributos en los vectores de  $D_c$  que están observados en el vector  $x$ .
    - C) Usar los  $K$  vectores más cercanos (K-nearest neighbors) y considerar la moda como un estimado de los valores faltantes para los atributos nominales. Para atributos continuos, reemplazar el valor faltante por la media del atributo en la vecindad de los  $k$  vecinos mas cercanos (k-nearest neighborhood).

# Imputación con los k-vecinos mas cercanos (KNNI)[2]

A1	A2	A3	A4	Clase
4	5	NA	6	1
5	1	4	1	1
7	9	5	2	1
8	2	5	8	1
6	4	6	2	1

$D(r1,r2)= 6.48$   $D(r1,r3)=6.40$ ,  $D(r1,r4)=5.38$ ,  $D(r1,r5)= 4.58$   
Luego, si se usa  $k=1$  vecinos cercanos ( $r5$ ), NA debería ser reemplazado por 6, si se usa  $k=3$  vecinos cercanos ( $r3, r4$  y  $r5$ ), NA debería ser reemplazado por el promedio de 5, 5 y 6 que es 5.33

# Imputación con los k vecinos mas cercanos

## KNNI[3]

---

- El metodo no se podria aplicar si todas las filas de la matriz contienen al menos un valor faltante.
- Usualmente, se toma k igual a 10. Pero el numero de vecinos a usar es a lo sumo igual al numero de filas completas de la matriz.
- Cuando hay variables cuantitativas y categóricas se usa la distancia Gower en lugar de la distancia euclideana

# Imputación con los k vecinos mas cercanos (KNNI)[4]

---

- Cuando la base de datos tiene atributos de distintos tipos, ya no se puede usar las distancias usuales como Euclideana y Manhattan que solo sirven para atributos numericos.
- Hay muchas alternativas para medir la distancia entre los registros (ver Wilson y Martinez). Sin embargo, la distancia mas usada es la distancia Gower. Knn-imputación usando la distancia Gower esta implementada en dprep

`data(crx)`

`crx.knn=ec.knnimp(crx,nomatr=c(1,4:7,9:10,12:13),k=5)`

imputacion por KNN también esta implementada en RapidMiner.

# Distancia Gower(disponible en la librería StatMatch)

---

- Supongamos que tenemos  $p$  atributos algunos de los cuales son cuantitativos y otros nominales no ordinales.

La distancia Gower entre los registros  $X_i$  y  $X_j$

$X_i=(x_{i1},x_{i2},\dots,x_{ip})$  y  $X_j=(x_{j1},x_{j2},\dots,x_{jp})$

Estaa data por  $D_G(X_i,X_j)=\sum_{k=1}^p d(x_{ik},x_{jk})/p$

Donde  $d(x_{ik},x_{jk})=|x_{ik}-x_{jk}|/\text{rango}(X_k)$  si la variable es continua y

$d(x_{ik},x_{jk})=1$  si la variable  $X_k$  es nominal y sus valores son distintos y  $d(x_{ik},x_{jk})=0$  si sus valores son iguales.

El valor de la distancia Gower esta estandarizado entre 0 y 1.



# Insertando aleatoriamente valores faltantes[1]

---

```
> mat1=cbind(c("a","b","ab","b","b","ab"),c("ac","ad","ac","ab","ac","ab"))
> mat1 # formando una matriz 6x12 a partir del vector mat1
      [,1] [,2]
[1,] "a"  "ac"
[2,] "b"  "ad"
[3,] "ab" "ac"
[4,] "b"  "ab"
[5,] "b"  "ac"
[6,] "ab" "ab"
> dim(mat1)
[1] 6 2
```

## Insertando aleatoriamente datos faltantes[2]

---

```
>#convirtiendo la matriz en un arreglo vectorial
>mat2=as.vector(mat1)
> mat2
[1] "a" "b" "ab" "b" "b" "ab" "ac" "ab" "ac" "ab" "ac" "ab"
>#insertando al azar 6 valores faltantes
>mat2[sample(1:12,6)]=NA
> mat2
[1] NA  "b" "ab" NA  NA  "ab" "ac" NA  "ac" NA  NA  "ab"
>#reconstruyendo la matriz
>mat2=matrix(mat2,ncol=2)
```

# Insertando aleatoriamente datos faltantes[3]

---

```
>#anadiendole una columna ficticia de clases, porque la imputación
>en dprep solo es para datos supervisados
>mat3=cbind(mat3,rep(1,6))
>#convirtiendo mat3 en dataframe
>mat3=data.frame(mat3)
> mat3
  X1  X2 X3
1 <NA> ac 1
2  b <NA> 1
3 ab  ac  1
4 <NA> <NA> 1
5 <NA> <NA> 1
6 ab  ab  1
```

# Imputando los datos faltantes

---

```
>ce.mimp(mat3,"mean",1:2,nomatr=c(1,2))
```

Summary of imputations using substitution of mean (mode for nominal features):

Row Column Class Imput.value

```
[1,] "1" "1"  "1"  "ab"  
[2,] "2" "2"  "1"  "ac"  
[3,] "4" "1"  "1"  "ab"  
[4,] "4" "2"  "1"  "ac"  
[5,] "5" "1"  "1"  "ab"  
[6,] "5" "2"  "1"  "ac"
```

Total number of imputations per class:

1

6

## Otros métodos de imputación.

---

- **Hot deck and Cold deck.** [[nces.ed.gov/statprog](https://nces.ed.gov/statprog)]. En Cold deck se usan valores de estudios similares para reemplazar valores perdidos en el estudio actual. En Hot deck se usa valores de atributos correlacionados con el atributo que contiene el valor faltante para sustituirlos.
- **Modelo predictivo:** Regresión Lineal (atributos continuos), Regresión Logística (atributos binarios), logística Polychotomous (atributos nominales). El atributo con valor faltante es usado como la variable de respuesta y los demás atributos son considerados predictoras. En general, se puede usar modelos de Classification and Regression Trees.
- **Desventajas:** Puede crear sesgo, requiere correlación alta entre predictoras. Cómputo lento.

## Otros métodos de imputación.

---

- Imputación Múltiple. Se imputan varias veces los valores faltantes con valores simulados de una distribución que se asume para cada variable.
- Metodo de la SVD ( $X=U\Sigma V'$ ). Iterativamente, considerando inicialmente a los missings como las medias de cada columna. Y sustituyendo los missings en cada iteración, hasta que la norma de la matriz converja.
- Algoritmo EM. Asumir una distribución para las predictoras y estimando los parámetros iterativamente hasta alcanzar convergencia.
- Los árboles de decisión tienen su propio enfoque para tratar valores faltantes.

# Imputacion en Python

---

La libreria pandas tiene funciones que permiten hacer imputacion por la media y/o mediana en caso de que los atributos sean cuantitativos y por la moda en el caso de atributos categoricos . La function imputer de la libreria sklearn solo hace imputacion por vecinos mas cercanos con datos cuantitativos. Si hubieran variables categoricas estan deberian convertirse en numericas antes de imputar por knn. Ademias, predice el valor missing usando regresion no-parametrica por k-nn

# Efecto de valores faltantes en clasificacion supervisada

---

- Para conjuntos de datos con una pequena cantidad de valores faltantes se observa poca diferencia entre la eliminacion de casos y otros metodos de imputacion.
- Cuando se usa eliminacion de casos la variabilidad del estimado del error de clasificacion aumenta.
- Casi no hay diferencia entre usar imputacion por la media e imputacion por la mediana.
- El efecto de los valores faltantes depende de la forma que se distribuyen en la matriz de datos y en su localizacion con respecto a las variables mas importantes.
- El porcentaje de instancias con valores faltantes tiene mayor efecto en el proceso de clasificacion que el porcentaje total de valores faltantes en la matriz de datos
- El tratamiento de los valores faltantes en el procesos de clasificacion depende del clasificador que esta siendo usado.



# Preprocesamiento-Normalización

---

- La normalización de datos consiste en re-escalar los valores de los atributos dentro de un rango especificado, tal como -1 a 1 o 0 a 1.
- También es conocido como “normalizacion del rango”.

# Razones para normalizar

---

- Normalizar los datos de entrada ayudará a acelerar la fase de aprendizaje.
- Los atributos con rangos grandes de valores tendrán más peso que los atributos con rangos de valores más pequeños, y entonces dominarán la medida de distancia. Por ejemplo, el clasificador K-nn usando la medida de distancia euclídeana depende de que todas las dimensiones de los valores de entrada estén en la misma escala.
- También puede ser necesario aplicar algún tipo de normalización de datos para evitar problemas numéricos tales como pérdida de precisión y desbordamientos aritméticos (overflows).

# Conjunto de datos Bupa

---

Número de instancias: 345

Número de atributos: 7

Descripción de los atributos:

1. Mcv volumen corpuscular
2. alkphos fosfatasa alcalina
3. sgpt alamine aminotransferase
4. sgot aspartate aminotransferase
5. gammagt gamma-glutamyl transpeptidase
6. drinks numero de bebidas alcoholicas
7. Class: 1 (higado enfermo) y 2 (higado sano)

**> bupa[1:20,]**

	<b>V1</b>	<b>V2</b>	<b>V3</b>	<b>V4</b>	<b>V5</b>	<b>V6</b>	<b>V7</b>
<b>1</b>	<b>85</b>	<b>92</b>	<b>45</b>	<b>27</b>	<b>31</b>	<b>0</b>	<b>1</b>
<b>2</b>	<b>85</b>	<b>64</b>	<b>59</b>	<b>32</b>	<b>23</b>	<b>0</b>	<b>2</b>
<b>3</b>	<b>86</b>	<b>54</b>	<b>33</b>	<b>16</b>	<b>54</b>	<b>0</b>	<b>2</b>
<b>4</b>	<b>91</b>	<b>78</b>	<b>34</b>	<b>24</b>	<b>36</b>	<b>0</b>	<b>2</b>
<b>5</b>	<b>87</b>	<b>70</b>	<b>12</b>	<b>28</b>	<b>10</b>	<b>0</b>	<b>2</b>
<b>6</b>	<b>98</b>	<b>55</b>	<b>13</b>	<b>17</b>	<b>17</b>	<b>0</b>	<b>2</b>
<b>7</b>	<b>88</b>	<b>62</b>	<b>20</b>	<b>17</b>	<b>9</b>	<b>0.5</b>	<b>1</b>
<b>8</b>	<b>88</b>	<b>67</b>	<b>21</b>	<b>11</b>	<b>11</b>	<b>0.5</b>	<b>1</b>
<b>9</b>	<b>92</b>	<b>54</b>	<b>22</b>	<b>20</b>	<b>7</b>	<b>0.5</b>	<b>1</b>
<b>10</b>	<b>90</b>	<b>60</b>	<b>25</b>	<b>19</b>	<b>5</b>	<b>0.5</b>	<b>1</b>
<b>11</b>	<b>89</b>	<b>52</b>	<b>13</b>	<b>24</b>	<b>15</b>	<b>0.5</b>	<b>1</b>
<b>12</b>	<b>82</b>	<b>62</b>	<b>17</b>	<b>17</b>	<b>15</b>	<b>0.5</b>	<b>1</b>
<b>13</b>	<b>90</b>	<b>64</b>	<b>61</b>	<b>32</b>	<b>13</b>	<b>0.5</b>	<b>1</b>
<b>14</b>	<b>86</b>	<b>77</b>	<b>25</b>	<b>19</b>	<b>18</b>	<b>0.5</b>	<b>1</b>
<b>15</b>	<b>96</b>	<b>67</b>	<b>29</b>	<b>20</b>	<b>11</b>	<b>0.5</b>	<b>1</b>
<b>16</b>	<b>91</b>	<b>78</b>	<b>20</b>	<b>31</b>	<b>18</b>	<b>0.5</b>	<b>1</b>
<b>17</b>	<b>89</b>	<b>67</b>	<b>23</b>	<b>16</b>	<b>10</b>	<b>0.5</b>	<b>1</b>
<b>18</b>	<b>89</b>	<b>79</b>	<b>17</b>	<b>17</b>	<b>16</b>	<b>0.5</b>	<b>1</b>
<b>19</b>	<b>91</b>	<b>107</b>	<b>20</b>	<b>20</b>	<b>56</b>	<b>0.5</b>	<b>1</b>
<b>20</b>	<b>94</b>	<b>116</b>	<b>11</b>	<b>33</b>	<b>11</b>	<b>0.5</b>	<b>1</b>

# Normalización Z-score

---

Los valores  $V$  son normalizados en base a la media y desviación estándar.

$$V' = (V - \text{mean}) / \text{std}$$

Este método trabaja bien en los casos en que no se conoce el máximo y mínimo de los datos de entrada pero no cuando existen outliers que tienen un gran efecto en el rango de los datos.

```
> zbupa=rangenorm(bupa,'znorm',superv=T)
```

# Normalización Min-Max

---

Este método realiza una transformación lineal de los datos originales  $V$  en el intervalo especificado  $[\text{newmin}, \text{newmax}]$

$$V' = (V - \text{min}) * (\text{newmax} - \text{newmin}) / (\text{max} - \text{min}) + \text{newmin}$$

La ventaja de este método es que preserva exactamente todas las relaciones entre los datos. No introduce ningún potencial sesgo en los datos. La desventaja es que se encontrará un error “fuera del límite” (“out of bounds”) si un futuro ingreso de datos cae fuera del rango original.

- `> mmbupa=rangenorm(bupa,mmnorm,superv=TRUE)`

## Normalización por escalamiento decimal

---

Este método realiza la normalización moviendo el punto decimal de los valores. El número de puntos decimales movidos depende del máximo valor absoluto.

$$V' = V / 10^j$$

donde  $j$  es el entero mas pequeño tal que  $\text{Max}(|V'|) < 1$ . Sólo es útil cuando los valores de los atributos son mayores que 1 en valor absoluto.

- `> dsbupa=rangenorm(bupa,'dscale',superv=T)`

# Normalización Sigmoidal

---

Este método realiza una transformación no lineal de los datos de entrada en el rango -1 a 1, usando una función sigmoidal.

$$V' = (1 - e^{(-a)}) / (1 + e^{(-a)}) \text{ donde } a = (V - \text{mean}) / \text{std}$$

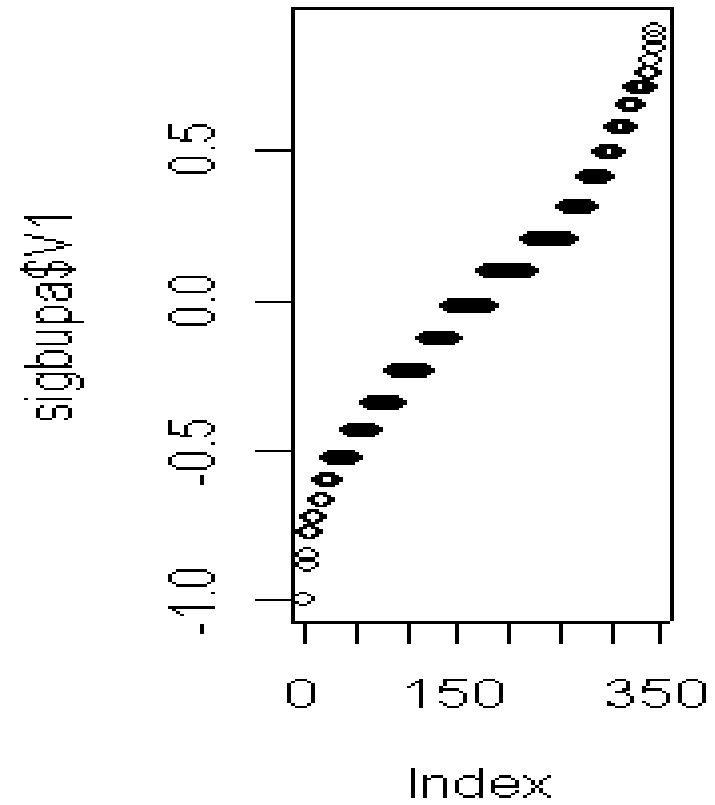
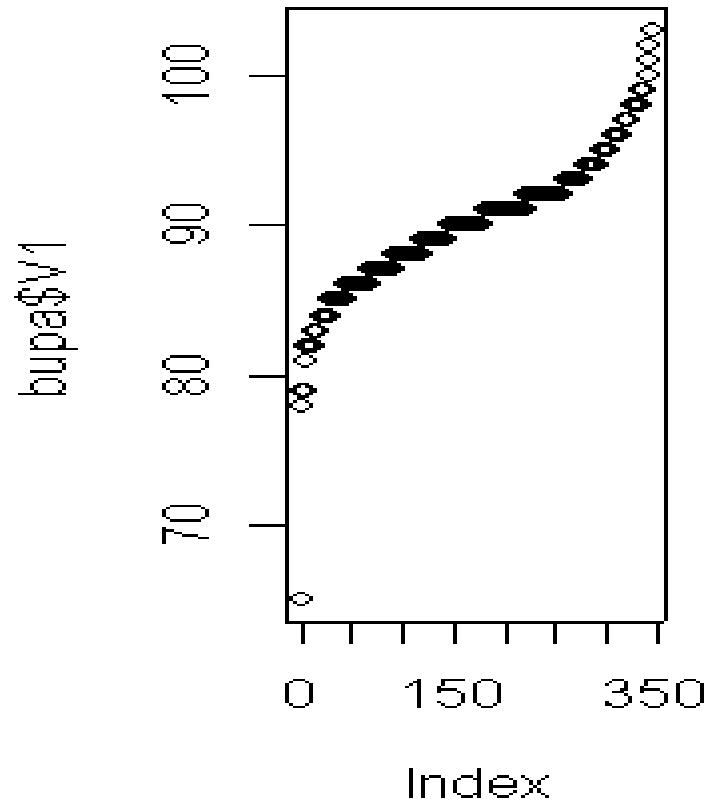
Los datos dentro de una desviación estándar de la media son mapeados a la región casi linear del sigmoide. Los puntos anómalos son comprimidos a lo largo de las colas de la función sigmoidal.

La normalización sigmoidal es especialmente apropiada cuando se tienen datos anómalos que se desean incluir en el conjunto de datos. Este previene que los valores que ocurren más comúnmente sean comprimidos en los mismos valores, sin perder la habilidad de representar grandes valores anómalos.

```
> sigbupa=rangenorm(bupa, 'signorm',superv=T)
> plot(sort(bupa$V1))
> plot(sort(sigbupa$V1))
```



## Visualización del efecto de la transformación sigmoidal



# Normalización Softmax

---

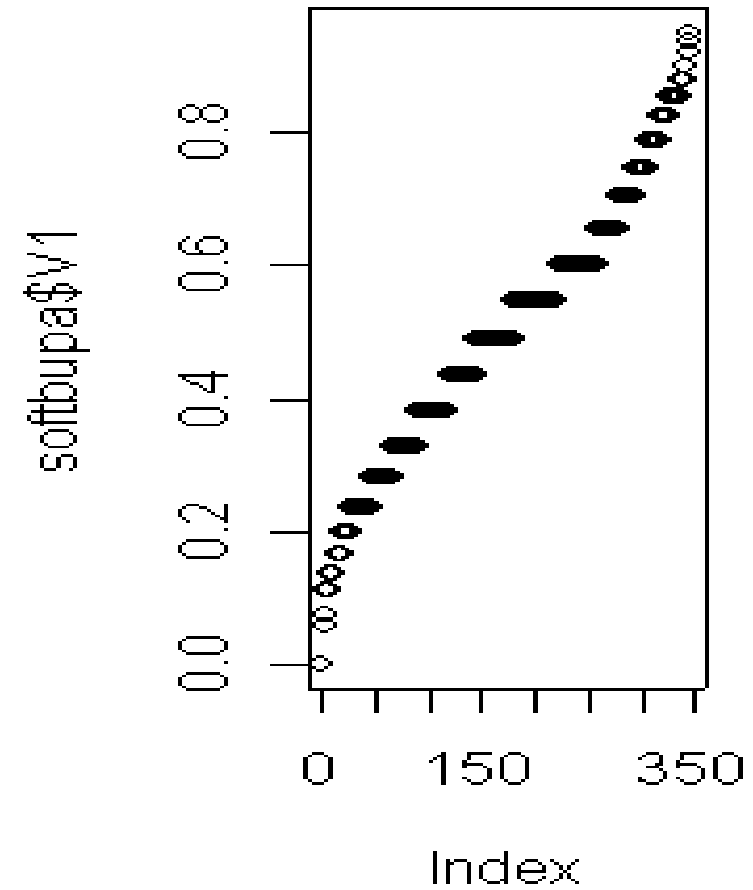
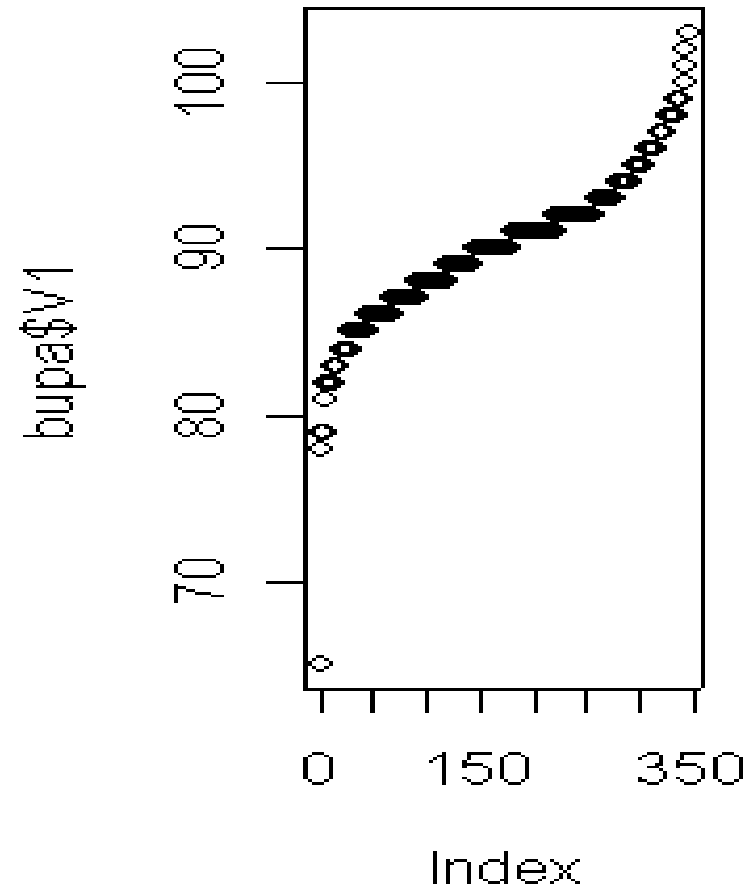
Se llama así porque llega suavemente hacia su valor máximo y mínimo. La transformación es mas o menos lineal en el rango medio, y tiene una ligera no linealidad a ambos extremos. El rango total cubierto es 0 a 1 y la transformación asegura que no ocurran valores futuros que caigan fuera del rango.

$$V' = 1 / ( 1 + e^{(-a)} )$$

donde  $a = ( V - \text{mean} ) / \text{std}$

- `softbupa=softmaxnorm(bupa)`
- `zbupa=rangenorm(bupa,method="znorm")`

## Visualizacion del efecto de la transformacion softmax



---

Boxplots que muestran el efecto de la normalización

```
> par(mfrow=c(2,3))  
> boxplot(bupa[,1:6],main="bupa")  
> boxplot(zbupa[,1:6],main="znorm bupa")  
> boxplot(mmbupa[,1:6],main="min-max bupa")  
> boxplot(dsbupa[,1:6],main="dec scale bupa")  
> boxplot(sigbupa[,1:6],main="signorm bupa")  
> boxplot(softbupa[,1:6],main="softmax bupa")
```

