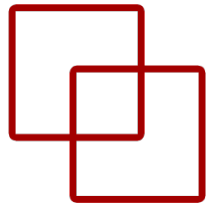
Análisis y preprocesamiento de datos

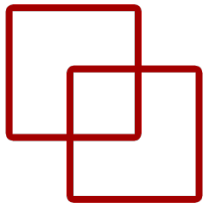
Dr. Manuel Castillo-Cara
www.manuelcastillo.eu
Departamento de Inteligencia Artificial
Escuela Técnica Superior de Ingeniería Informática
Universidad Nacional de Educación a Distancia (UNED)

Preliminar



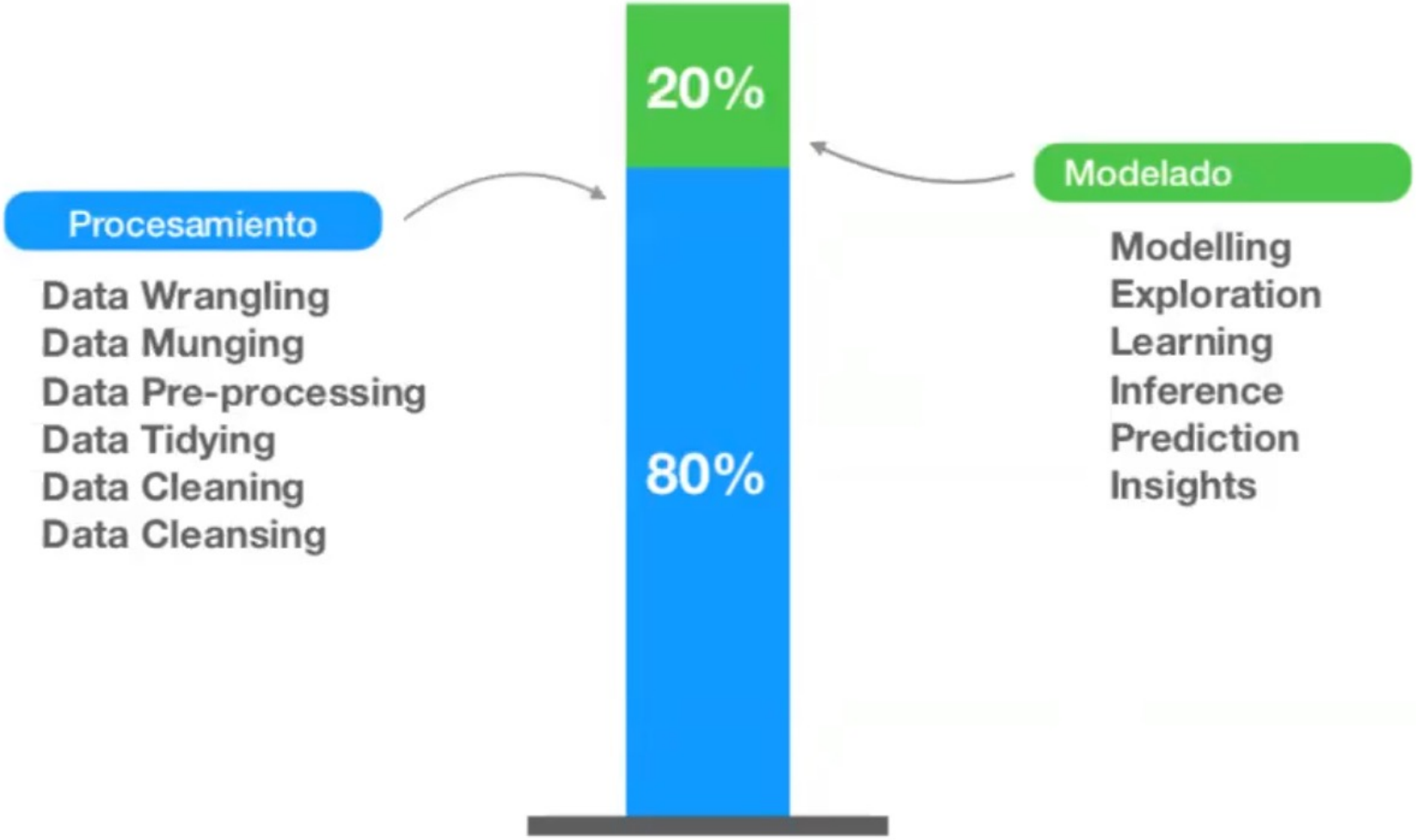
-

Índice



Fuentes de datos

Procesamiento de datos – Tarea que más tiempo consume



Formato de los datos

- Es un problema real, los datos originales no vendrán en un formato propicio para su análisis directo (estructurado)

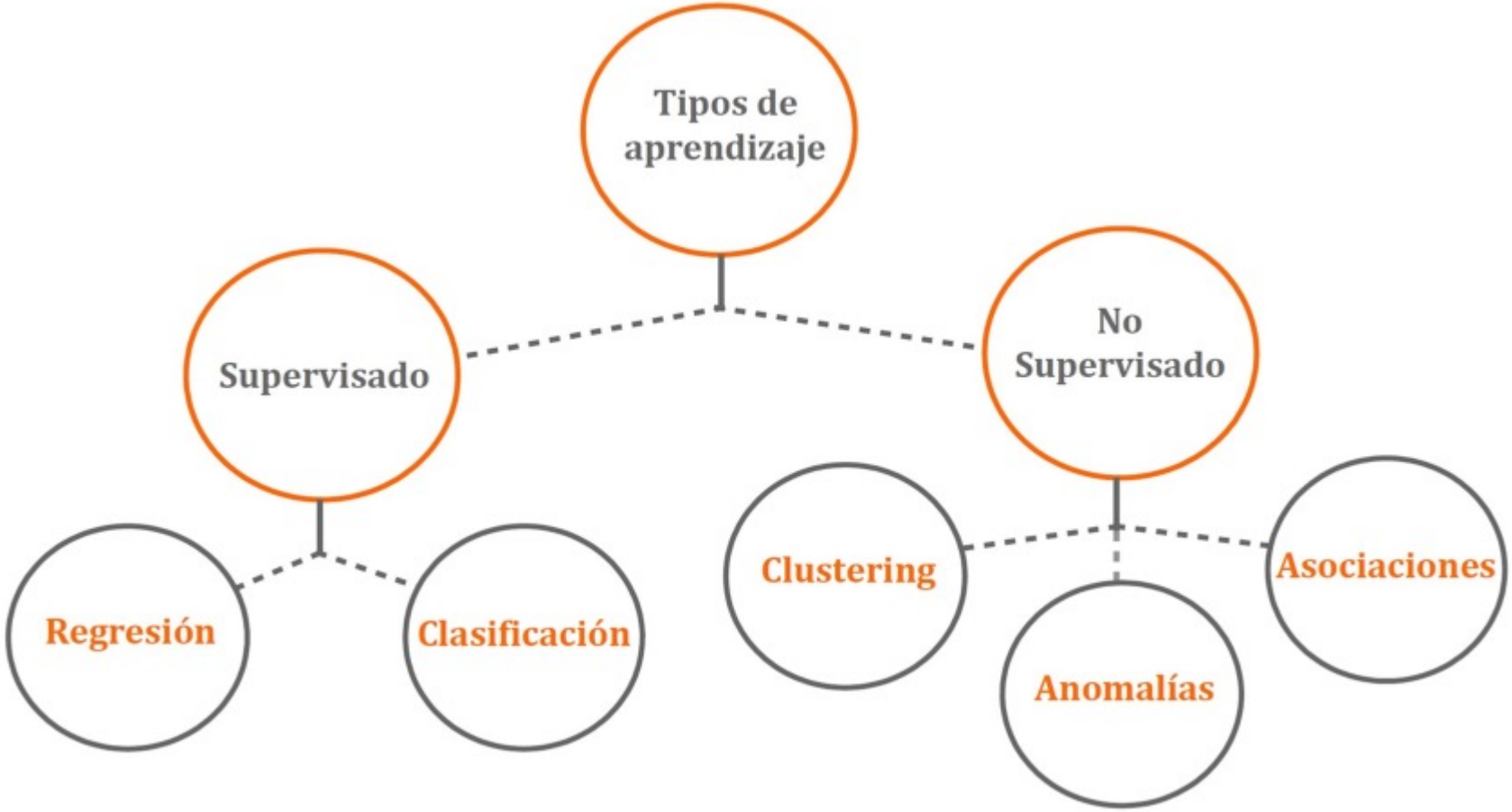
Sepal LengthCM	Sepal WidthCm	Petal LengthCm	Petal WidthCm	Species
5.1	3.5	1.4	0.2	setosa
4.9	3	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
5.4	3.7	1.5	0.2	setosa
4.8	3.4	1.6	0.2	setosa
4.8	3	1.4	0.1	setosa
4.3	3	1.1	0.1	setosa

Tidy data

> "Tidy data is a standard way of mapping the meaning of a dataset to its structure. A dataset is messy or tidy depending on how rows, columns and tables are matched up with observations, variables and types"
>
> Hadley Wickham 2014 (Tidy Data - Journal of Statistical Software)

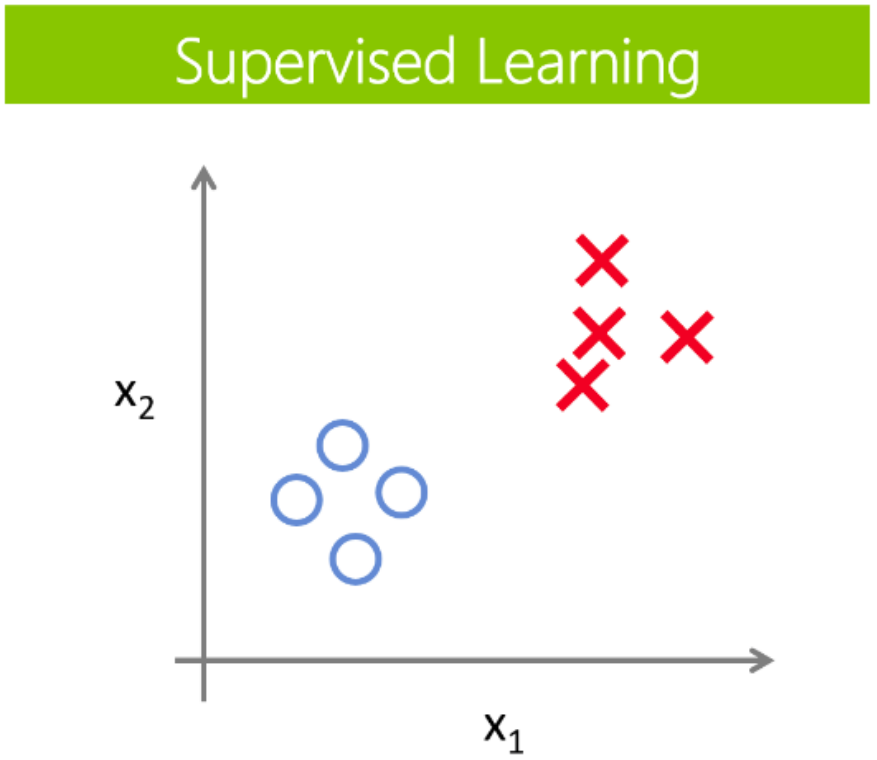
- Un dataset "Tidy" mantiene las siguientes **propiedades**:
  - Cada variable representa una columna
  - Cada observación representa una fila
  - Cada unidad observacional representa una tabla
- Permite definir objetivos, estrategias y herramientas **estandarizadas** para la limpieza y transformación de datos.
- Permite definir un vocabulario y operadores de transformación desde un punto de vista **agnóstico a cualquier lenguaje**.
- Artículo: *Wickham, H. (2014). Tidy data. Journal of Statistical Software*

Tipos de aprendizaje

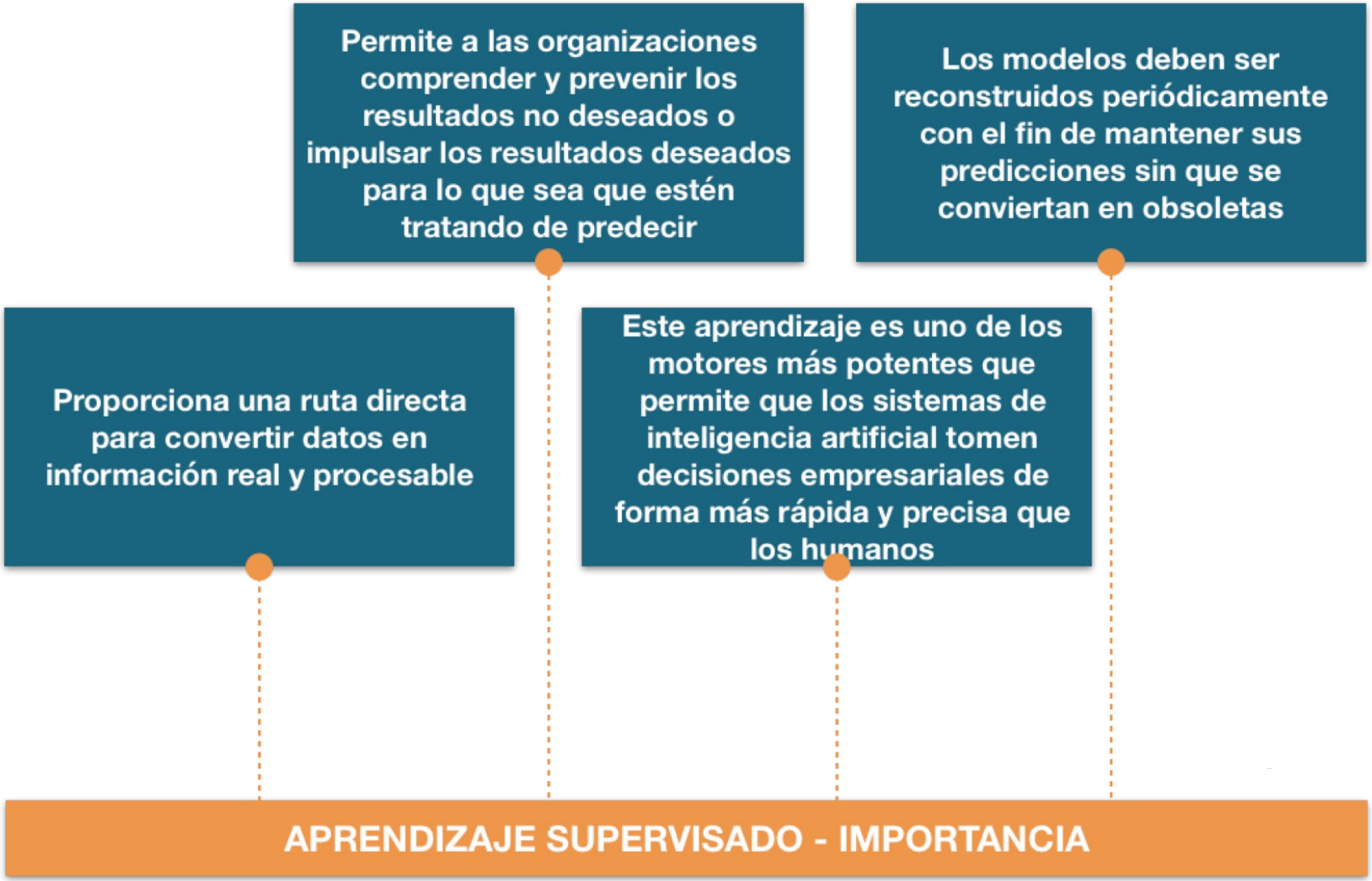


Aprendizaje Supervisado

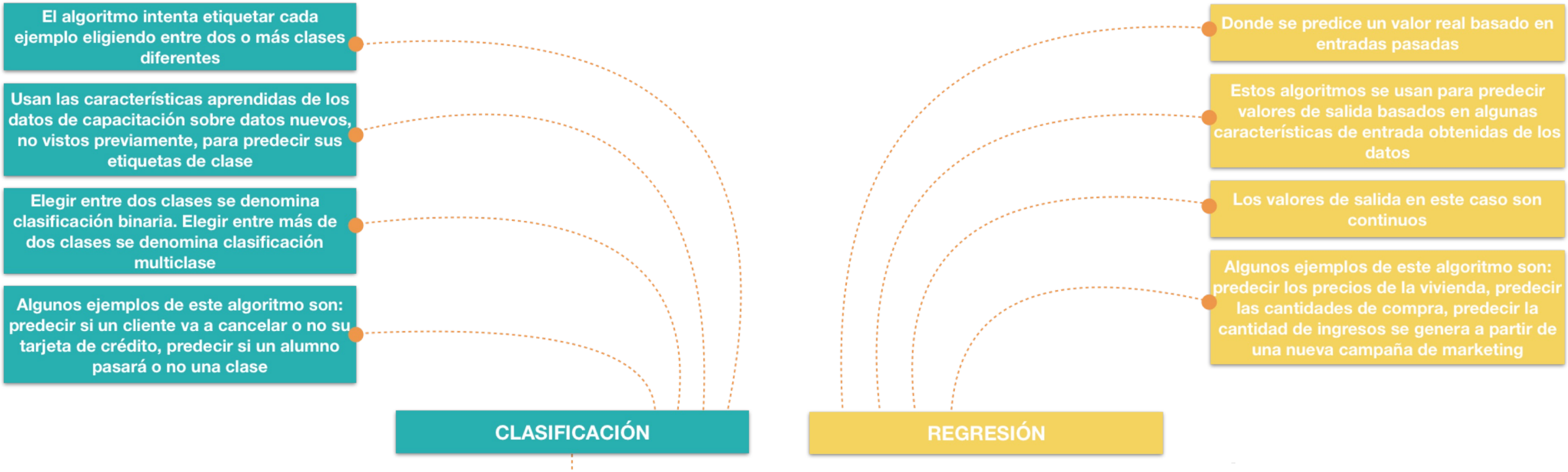
- Los algoritmos trabajan con datos "etiquetados" (*labeled data*).
- **Objetivo**: encontrar una función que, dadas las variables de entrada (*input data*), les asigne la etiqueta de salida adecuada.
- Entrenamiento con un "histórico" de datos para "aprender" a asignar la etiqueta de salida.
- **Función final**: predecir el valor de salida.



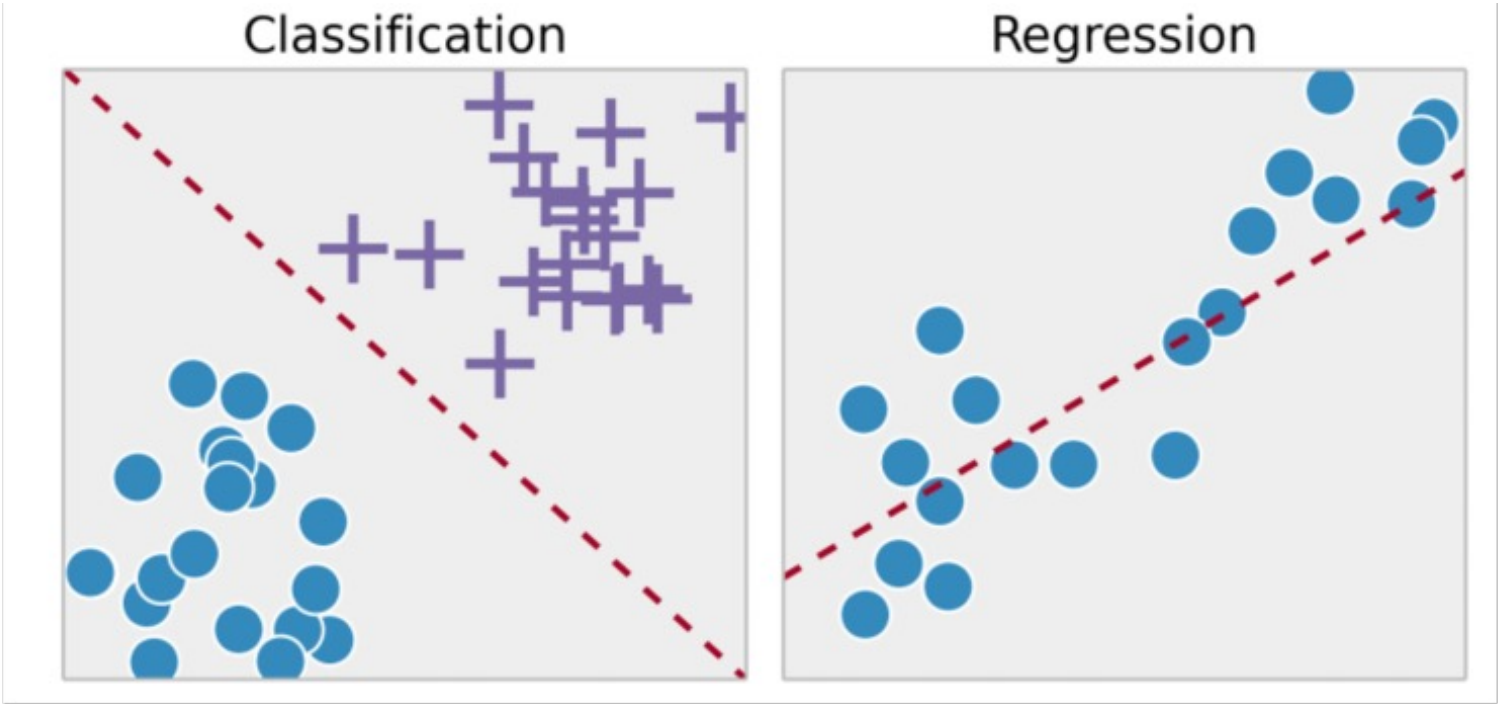
Aprendizaje supervisado

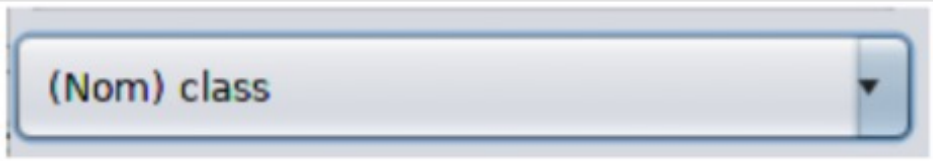


Clasificación Vs. Regresión

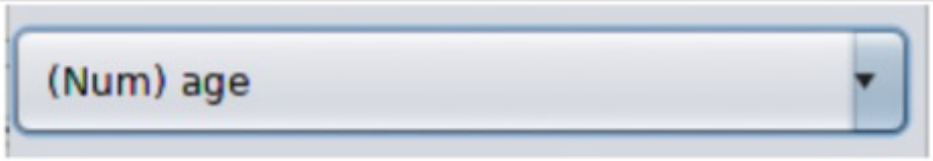


Clasificación Vs. Regresión





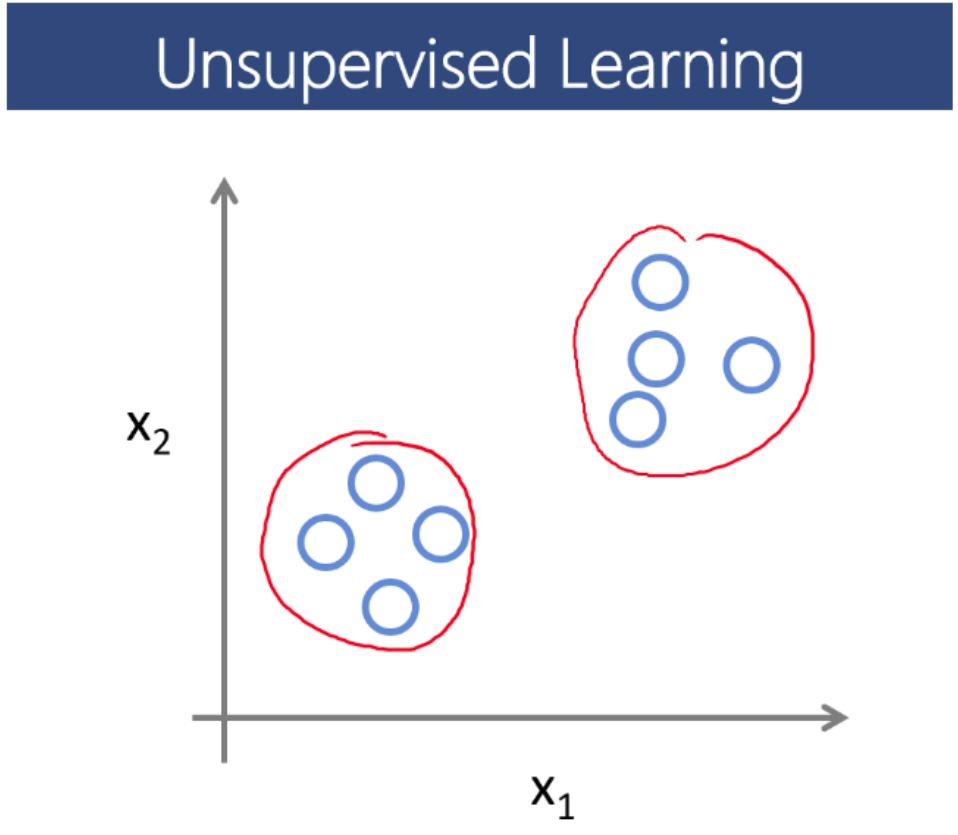
(a) Muestra de un atributo nominal.



(b) Muestra de un atributo numérico.

Aprendizaje No Supervisado

- No se dispone de datos “etiquetados” para el entrenamiento.
- Sólo se conocen los datos de entrada, pero no tienen atributo clase (dato de salida).
- Sólo pueden describirse la estructura de los datos.
- **Objetivo**: Encontrar algún tipo de organización que simplifique el análisis.
- Por ello, tienen un carácter exploratorio.
  - *(Ejemplo de un txt de Weka el atributo clase).*



SLAs Vs. ULAs

	A	B	C	D	E	F	G	H	I
1	preg	plas	pres	skin	insu	mass	pedi	age	class
2	1	85	66	29	0	26.6	351	31	tested_negative
3	5	116	74	0	0	25.6	201	30	tested_negative
4	10	115	0	0	0	35.3	134	29	tested_negative
5	4	110	92	0	0	37.6	191	30	tested_negative
6	10	139	80	0	0	27.1	1441	57	tested_negative
7	8	99	84	0	0	35.4	388	50	tested_negative
8	5	117	92	0	0	34.1	337	38	tested_negative
9	5	109	75	26	0	36	546	60	tested_negative

¿Tipo problema?

	A	B	C	D	E	F	G	H
1	preg	plas	pres	skin	insu	mass	pedi	age
2	1	85	66	29	0	26.6	351	31
3	5	116	74	0	0	25.6	201	30
4	10	115	0	0	0	35.3	134	29
5	4	110	92	0	0	37.6	191	30
6	10	139	80	0	0	27.1	1441	57
7	8	99	84	0	0	35.4	388	50
8	5	117	92	0	0	34.1	337	38
9	5	109	75	26	0	36	546	60

¿Tipo problema?

Regresión Vs. Clasificación

	A	B	C	D	E	F	G	H	I
1	preg	plas	pres	skin	insu	mass	pedi	age	class
2	1	85	66	29	0	26.6	351	31	tested_negative
3	5	116	74	0	0	25.6	201	30	tested_negative
4	10	115	0	0	0	35.3	134	29	tested_negative
5	4	110	92	0	0	37.6	191	30	tested_negative
6	10	139	80	0	0	27.1	1441	57	tested_negative
7	8	99	84	0	0	35.4	388	50	tested_negative
8	5	117	92	0	0	34.1	337	38	tested_negative
9	5	109	75	26	0	36	546	60	tested_negative

¿Tipo problema?

¿Tipo problema?

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	class
2	0.00632	18	2.31	0	538	6575	65.2	4.09	1	296	15.3	396.9	4.98	24
3	0.02731	0	7.07	0	469	6421	78.9	4.9671	2	242	17.8	396.9	9.14	21.6
4	0.02729	0	7.07	0	469	7185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
5	0.03237	0	2.18	0	458	6998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
6	0.06905	0	2.18	0	458	7147	54.2	6.0622	3	222	18.7	396.9	5.33	36.2
7	0.02985	0	2.18	0	458	6.43	58.7	6.0622	3	222	18.7	394.12	5.21	28.7

Regresión Vs. Clasificación

	A	B	C	D	E	F	G	H	I
1	preg	plas	pres	skin	insu	mass	pedi	age	class
2	1	85	66	29	0	26.6	351	31	tested_negative
3	5	116	74	0	0	25.6	201	30	tested_negative
4	10	115	0	0	0	35.3	134	29	tested_negative
5	4	110	92	0	0	37.6	191	30	tested_negative
6	10	139	80	0	0	27.1	1441	57	tested_negative
7	8	99	84	0	0	35.4	388	50	tested_negative
8	5	117	92	0	0	34.1	337	38	tested_negative
9	5	109	75	26	0	36	546	60	tested_negative

Clasificación

Regresión

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	class
2	0.00632	18	2.31	0	538	6575	65.2	4.09	1	296	15.3	396.9	4.98	24
3	0.02731	0	7.07	0	469	6421	78.9	4.9671	2	242	17.8	396.9	9.14	21.6
4	0.02729	0	7.07	0	469	7185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
5	0.03237	0	2.18	0	458	6998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
6	0.06905	0	2.18	0	458	7147	54.2	6.0622	3	222	18.7	396.9	5.33	36.2
7	0.02985	0	2.18	0	458	6.43	58.7	6.0622	3	222	18.7	394.12	5.21	28.7

Tipo de problema en datasets

Iris

- Dimensiones: 150 instancias, 5 atributos.
- Entradas: Numéricas.

```
> data(iris)
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

CÓDIGO 2.25: Resumen del conjunto de datos Iris

Económica de Longley

- Descripción: Predecir el número de personas empleadas a partir de variables económicas.
- Dimensiones: 16 instancias, 7 atributos.
- Entradas: Numéricas.

```
> data(longley)
> head(longley)
     GNP.deflator     GNP Unemployed Armed.Forces Population Year
1947         83.0 234.289      235.6        159.0    107.608 1947
1948         88.5 259.426      232.5        145.6    108.632 1948
1949         88.2 258.054      368.2        161.6    109.773 1949
1950         89.5 284.599      335.1        165.0    110.929 1950
1951         96.2 328.975      209.9        309.9    112.075 1951
1952         98.1 346.999      193.2        359.4    113.270 1952
```

CÓDIGO 2.26: Resumen del conjunto de datos Longley

Boston Housing

- Descripción: Predecir el precio medio de una casa en los suburbios de Boston.
- Dimensiones: 506 instancias, 14 atributos.
- Entradas: Numéricas.

```
> data(BostonHousing)
> head(BostonHousing)
     crim zn indus chas   nox    rm  age    dis rad tax ptratio      b
1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3 396.90
2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8 396.90
3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8 392.83
4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7 394.63
5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7 396.90
6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7 394.12
  lstat medv
1  4.98 24.0
2  9.14 21.6
3  4.03 34.7
4  2.94 33.4
5  5.33 36.2
6  5.21 28.7
```

BreastCancer Wisconsin

- Descripción: Predecir si una muestra de tejido es maligna o benigna dadas propiedades sobre la muestra de tejido.
- Dimensiones: 699 instancias, 11 atributos.
- Entradas: Entero (Nominal).

```
> data(BreastCancer)
> head(BreastCancer)
      Id Cl.thickness Cell.size Cell.shape Marg.adhesion Epith.c.size
1 1000025            5         1          1             1            2
2 1002945            5         4          4             5            7
3 1015425            3         1          1             1            2
4 1016277            6         8          8             1            3
5 1017023            4         1          1             3            2
6 1017122            8        10         10             8            7
  Bare.nuclei Bl.cromatin Normal.nucleoli Mitoses     Class
1           1           3                1       1    benign
2          10           3                2       1    benign
3           2           3                1       1    benign
4           4           3                7       1    benign
5           1           3                1       1    benign
6          10           9                7       1 malignant
```

Identificación de vidrio

- Descripción: Predecir el tipo de vidrio a partir de propiedades químicas.
- Dimensiones: 244 instancias, 10 atributos.
- Entradas: Numérico.

```
> data(Glass)
> head(Glass)
       RI    Na   Mg   Al    Si    K   Ca Ba   Fe Type
1 1.52101 13.64 4.49 1.10 71.78 0.06 8.75  0 0.00    1
2 1.51761 13.89 3.60 1.36 72.73 0.48 7.83  0 0.00    1
3 1.51618 13.53 3.55 1.54 72.99 0.39 7.78  0 0.00    1
4 1.51766 13.21 3.69 1.29 72.61 0.57 8.22  0 0.00    1
5 1.51742 13.27 3.62 1.24 73.08 0.55 8.07  0 0.00    1
6 1.51596 12.79 3.61 1.62 72.97 0.64 8.07  0 0.26    1
```

Ionosphere

- Descripción: Predecir estructuras de alta energía en la atmósfera a partir de datos de antena.
- Dimensiones: 351 instancias, 35 atributos.
- Entradas: Numérico.

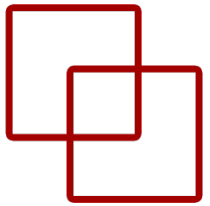
```
> data(Ionosphere)
> head(Ionosphere)
  V1 V2      V3       V4       V5       V6       V7       V8      V9      V34 Class
1  1  0 0.99539 -0.05889  0.85243  0.02306  0.83398 -0.37708 1.00000  0.45300  good
2  1  0 1.00000 -0.18829  0.93035 -0.36156 -0.10868 -0.93597 1.00000  0.02447   bad
3  1  0 1.00000 -0.03365  1.00000  0.00485  1.00000 -0.12062 0.88965  0.38238  good
4  1  0 1.00000 -0.45161  1.00000  1.00000  0.71216 -1.00000 0.00000  1.00000   bad
5  1  0 1.00000 -0.02401  0.94140  0.06531  0.92106 -0.23255 0.77152  0.65697  good
6  1  0 0.02337 -0.00592 -0.09924 -0.11949 -0.00763 -0.11824 0.14706  0.12011   bad
```

Diabetes de Pima Indians

- Descripción: Predecir el inicio de la diabetes en mujeres indias pima a partir de datos de registros médicos.
- Dimensiones: 768 instancias, 9 atributos.
- Entradas: Numérico.

```
> data(PimaIndiansDiabetes)
> head(PimaIndiansDiabetes)
  pregnant glucose pressure triceps insulin mass pedigree age diabetes
1        6     148       72      35       0 33.6    0.627  50      pos
2        1      85       66      29       0 26.6    0.351  31      neg
3        8     183       64       0       0 23.3    0.672  32      pos
4        1      89       66      23      94 28.1    0.167  21      neg
5        0     137       40      35     168 43.1    2.288  33      pos
6        5     116       74       0       0 25.6    0.201  30      neg
```

Sonar, Mines vs. Rocks



- Descripción: Predice los retornos de metal o roca a partir de los datos de retorno del sonar.
- Dimensiones: 208 instancias, 61 atributos.
- Entradas: Numérico.

```
> data(Sonar)
> head(Sonar)
     V1     V2     V3     V4     V5     V6     V7     V8     V9    V10 Class
1 0.0200 0.0371 0.0428 0.0207 0.0954 0.0986 0.1539 0.1601 0.3109 0.2111     R
2 0.0453 0.0523 0.0843 0.0689 0.1183 0.2583 0.2156 0.3481 0.3337 0.2872     R
3 0.0262 0.0582 0.1099 0.1083 0.0974 0.2280 0.2431 0.3771 0.5598 0.6194     R
4 0.0100 0.0171 0.0623 0.0205 0.0205 0.0368 0.1098 0.1276 0.0598 0.1264     R
5 0.0762 0.0666 0.0481 0.0394 0.0590 0.0649 0.1209 0.2467 0.3564 0.4459     R
6 0.0286 0.0453 0.0277 0.0174 0.0384 0.0990 0.1201 0.1833 0.2105 0.3039     R
```

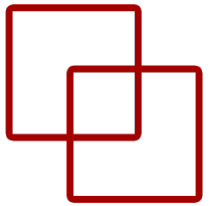
Base de datos Soya

- Descripción: Predecir problemas con cultivos de soja a partir de datos de cultivos.
- Dimensiones: 683 instancias, 26 atributos.
- Entradas: Entero (Nominal).

```
> data(Soybean)
> head(Soybean)
                  Class date plant.stand precip temp hail crop.hist
1 diaporthe-stem-canker    6           0      2    1    0         1
2 diaporthe-stem-canker    4           0      2    1    0         2
3 diaporthe-stem-canker    3           0      2    1    0         1
4 diaporthe-stem-canker    3           0      2    1    0         1
5 diaporthe-stem-canker    6           0      2    1    0         2
6 diaporthe-stem-canker    5           0      2    1    0         3
```

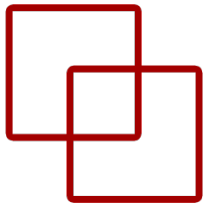
Estadística descriptiva

Entender nuestros datos



- Error típico: Trabajar directamente algoritmos sin conocer nuestros datos.
- Una buena cocina de datos (análisis y procesamiento) nos lleva a obtener mejores y más confiables resultados, entendiendo nuestro conjunto de datos.
- Las tareas a realizar son:
  - **Limpieza de datos:** Descubrimos datos *missing* o corruptos y/o marcamos o imputamos datos faltantes.
  - **Transformación de datos**: Descubrir distribuciones como la gaussiana o exponencial que nos genera una idea de qué técnica utilizar, p.e., escalamiento o transformación de datos.
  - **Modelado de datos**: A través de la estadística descriptiva obtenemos una idea de qué algoritmo poder utilizar.

Conocer nuestros datos

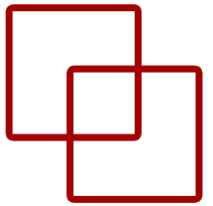


- La función "*head()*" nos permite ver nuestras instancias.

```
# View first 20 rows
data.head(10)
```

	preg	plas	pres	skin	test	mass	pedi	age	class
0	6	148	72	35	0	33.6	627.0	50	1
1	1	85	66	29	0	26.6	351.0	31	0
2	8	183	64	0	0	23.3	672.0	32	1
3	1	89	66	23	94	28.1	167.0	21	0
4	0	137	40	35	168	43.1	2288.0	33	1
5	5	116	74	0	0	25.6	201.0	30	0

Dimensiones del dataset



- Especifica el número de atributos e instancias.
  - Muchas instancias entonces es convenientes disminuirlas (capacidad computacional).
  - Muchos atributos entonces problema de la dimensionalidad
- Utilizaremos la propiedad "*shape*".

```
# Dimensions of your data
data.shape
```

```
(768, 9)
```

Resumen de los datos

- Muestra las estadísticas primarias para cada atributo. Muestra:
  - Mínimo (Min.) / Valor del 25 % Percentil (1st Qu.) / Mediana (Median) - Media (Mean) / Valor del 75 % Percentil (3st Qu.) / Máximo (Max.).
- Utilizaremos la función "*describe()*".

```
# Statistical Summary
data.describe()
```

	preg	plas	pres	skin	test	mass	pedi	age	class
count	768.000	768.000	768.000	768.000	768.000	768.000	768.000	768.000	768.000
mean	3.845	120.895	69.105	20.536	79.799	31.993	428.235	33.241	0.349
std	3.370	31.973	19.356	15.952	115.244	7.884	340.486	11.760	0.477
min	0.000	0.000	0.000	0.000	0.000	0.000	0.100	21.000	0.000
25%	1.000	99.000	62.000	0.000	0.000	27.300	205.000	24.000	0.000
50%	3.000	117.000	72.000	23.000	30.500	32.000	337.000	29.000	0.000
75%	6.000	140.250	80.000	32.000	127.250	36.600	591.500	41.000	1.000
max	17.000	199.000	122.000	99.000	846.000	67.100	2329.000	81.000	1.000

Distribución de clase

- Muchas veces un conjunto de datos para un problema de clasificación puede encontrarse desbalanceado:
  - No existe un equilibrio entre el número total de instancias para cada clase.
  - En este caso, se tendrán que aplicar técnicas de rebalanceo de clases.
- Función: "*groupby('class').size()*".

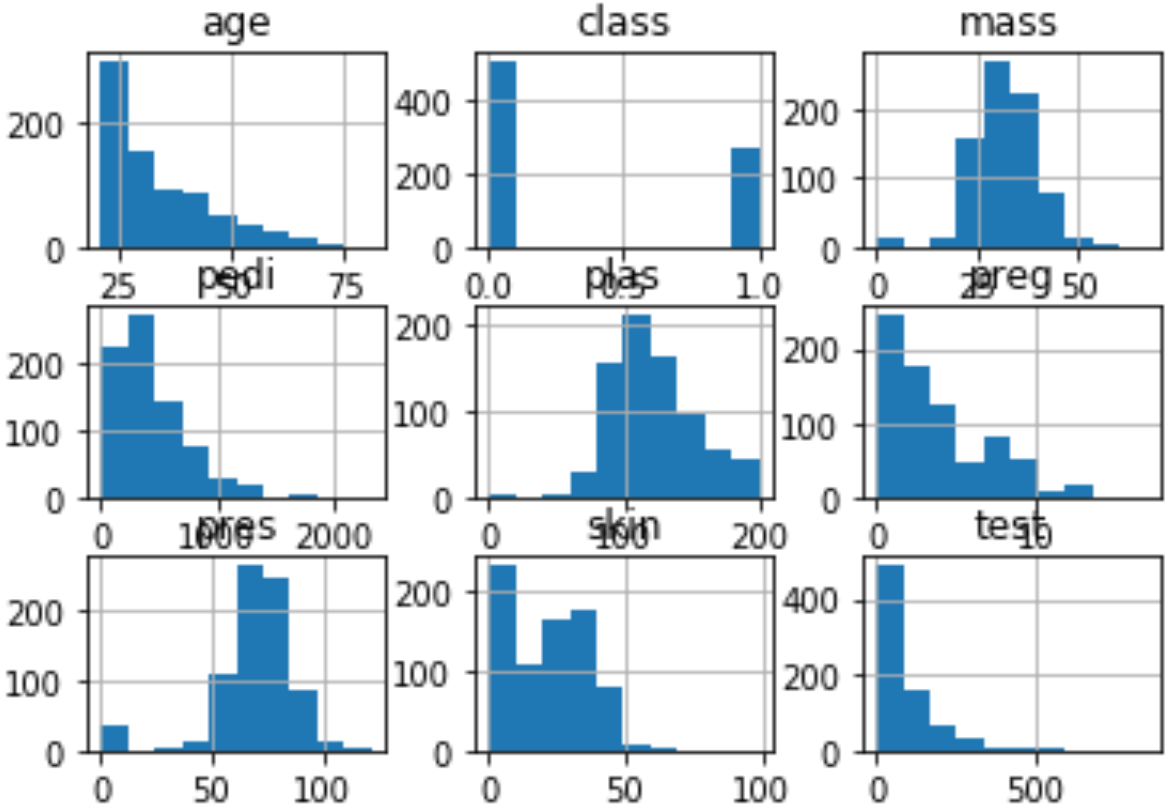
```
# Class Distribution
data.groupby('class').size()
```

```
class
0    500
1    268
dtype: int64
```

Visualización Univariable

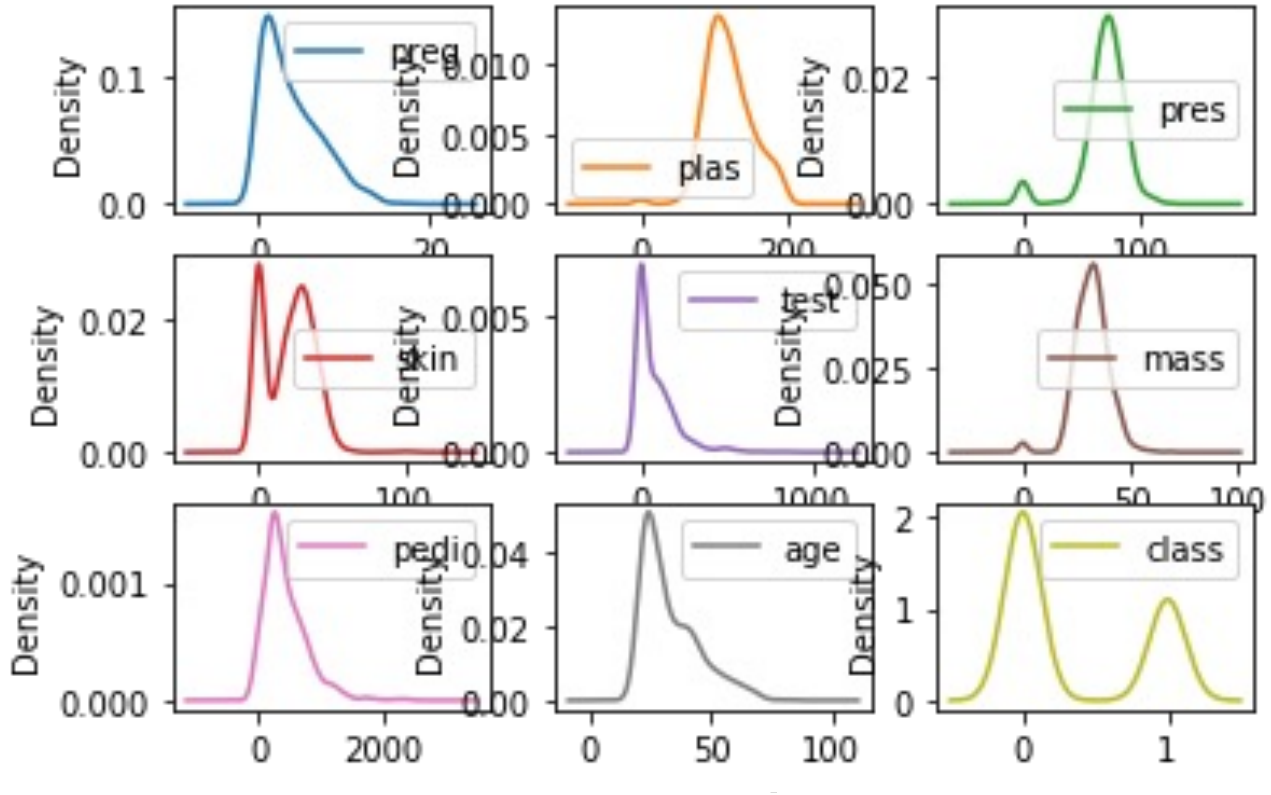
Histogramas

- Muestran un gráfico de barras a un atributo numérico.
- Su altura muestra el número de instancias
- Utilizaremos la función "*hist()*".



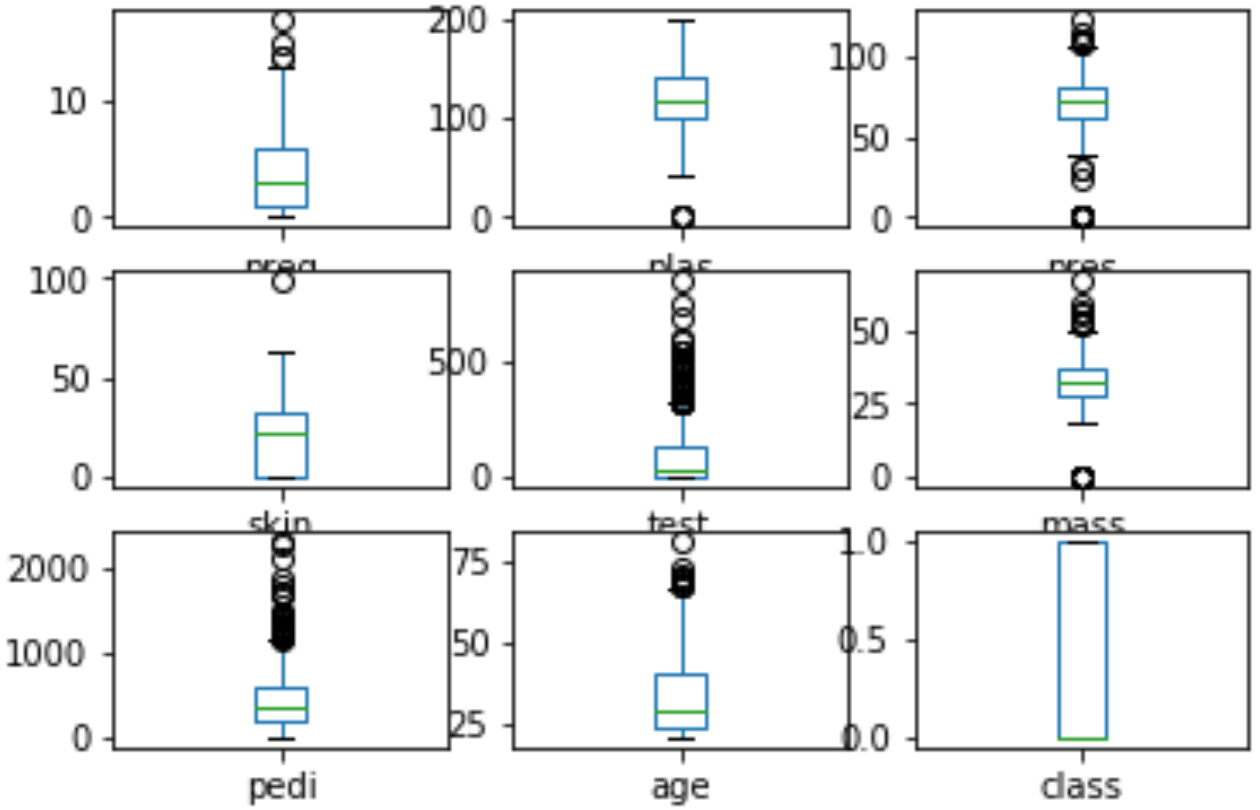
Densidad

- Visualizar los histogramas a líneas.
- Utilizaremos la función "*plot()*" con el parámetro *kind='density'*.
  - Puede observarse la la doble distribución gaussiana de *skin*.
  - Una posible distribución exponencial (similar a Lapacian) de *age*.
  - Un posible sesgo en *plas*.



Boxplot

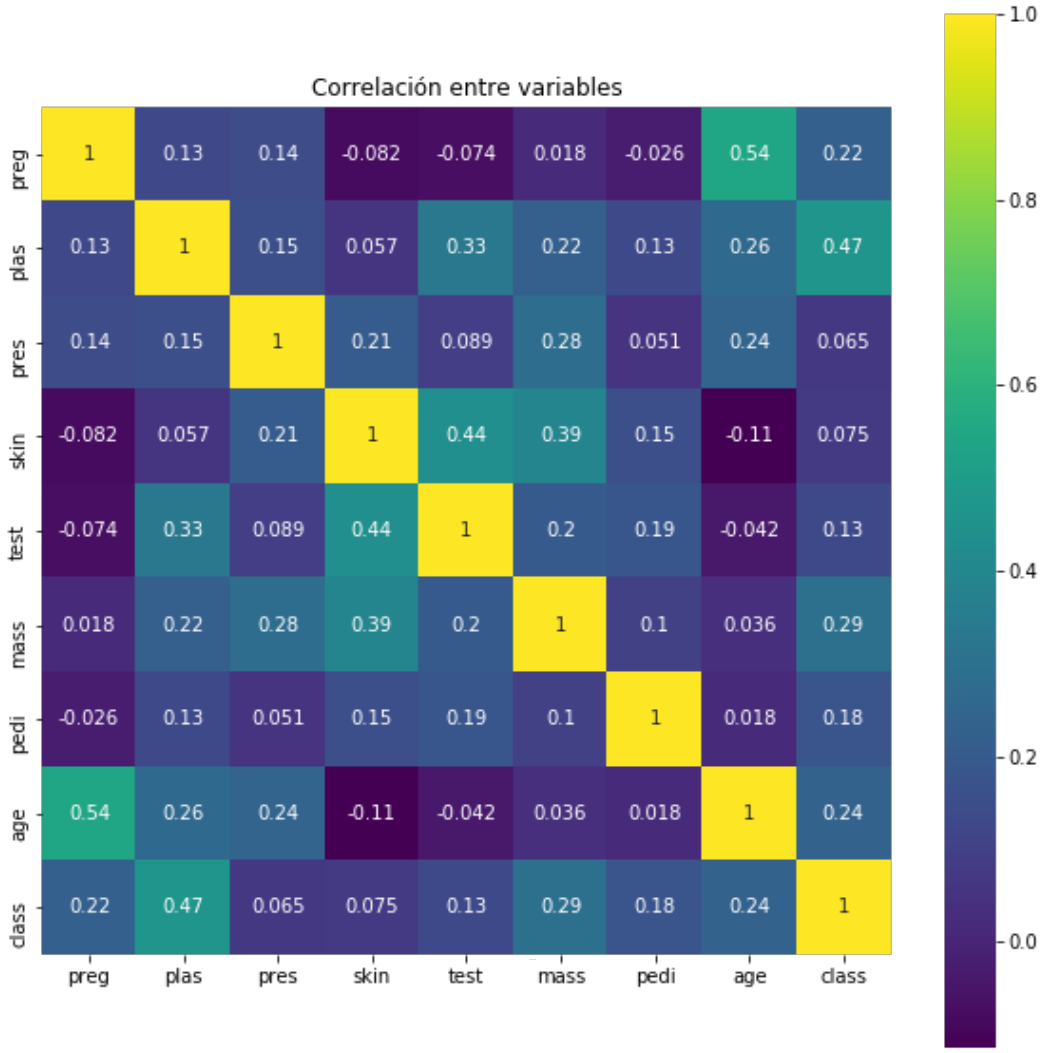
- La representación de este tipo de gráfico es como sigue:
  - La caja captura el 50% medio de los datos.
  - La línea muestra la mediana.
  - Los bordes de los gráficos muestran la extensión razonable de los datos.
  - Cualquier punto fuera de los bordes es un buen candidato para los valores atípicos (*outlayers*).
- Utilizaremos la función "*plot()*" con el parámetro *kind='box'*.



Visualización Multivariable

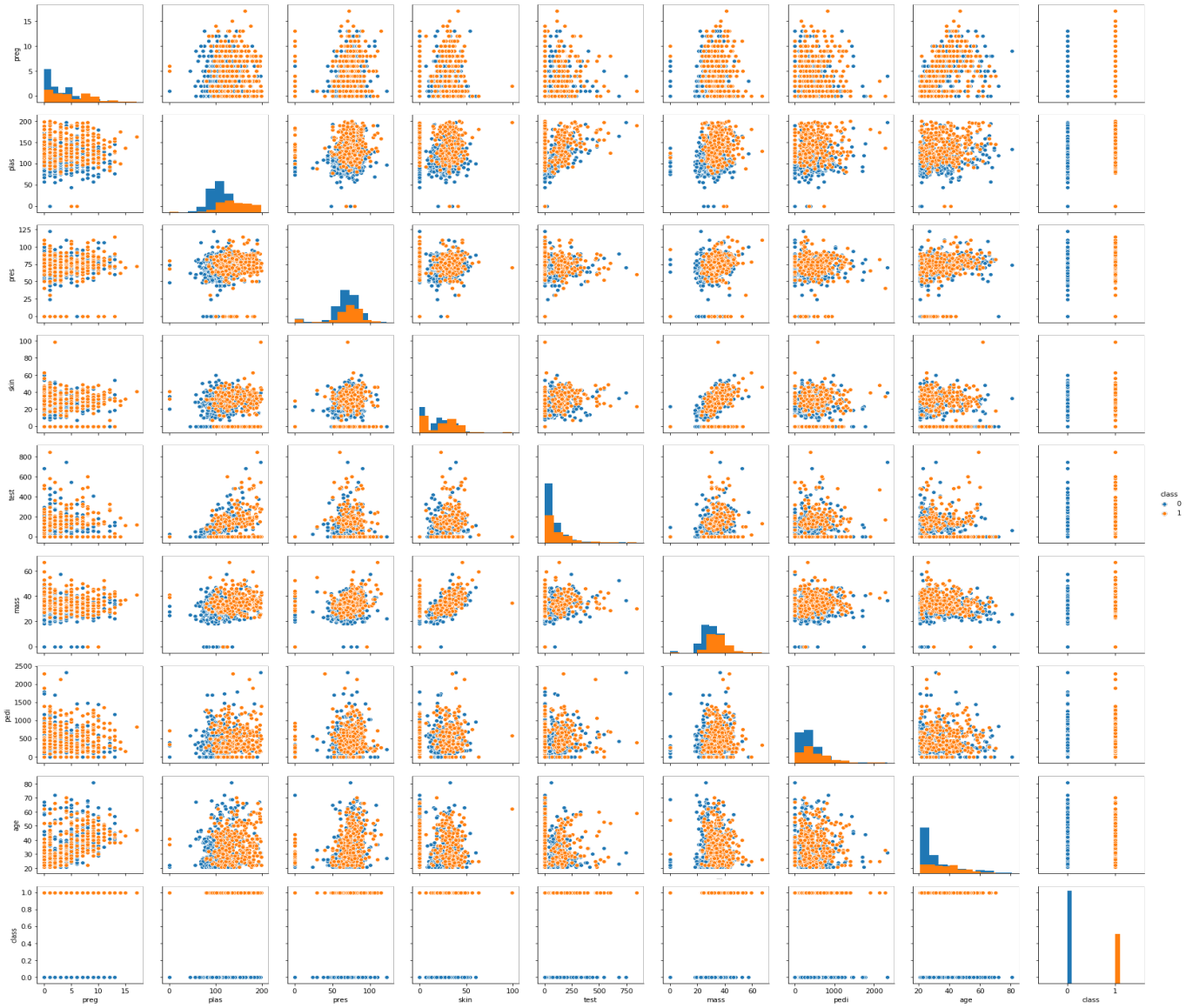
Correlación

- Evaluar la correlación entre nuestros atributos de tipo numérico, esto es, evaluar que atributos cambian juntos.
- Se puede observar:
  - Poca correlación entre atributos.
  - Casi siempre correlación positiva.



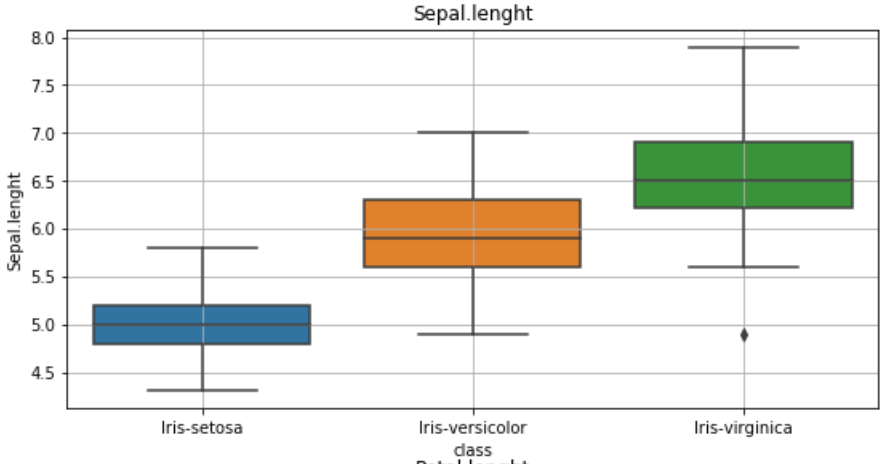
Matriz de dispersión por clase

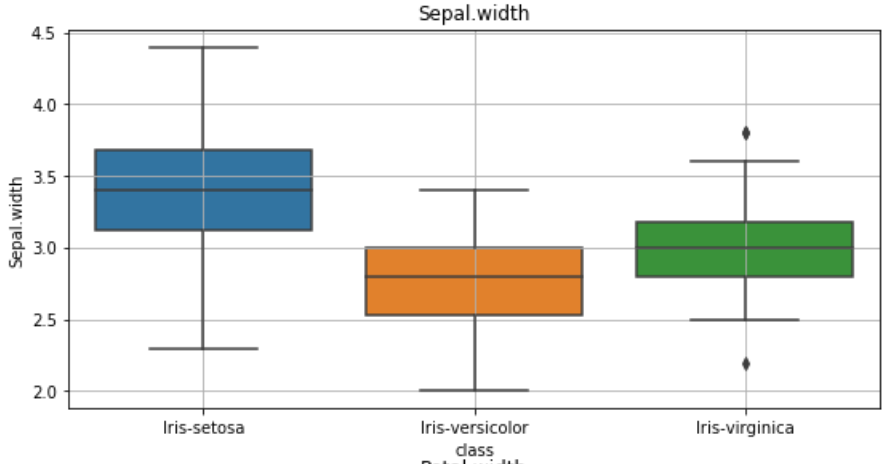
- Los puntos en una matriz de diagrama de dispersión pueden ser coloreados por la etiqueta de clase en problemas de clasificación.
  - Nos ayuda a detectar una **separación clara (o no clara) de las clases** y quizás a darnos una idea de lo difícil que puede ser el problema.
- Los diagramas están dados por la interacción de los pares de atributos, pero teniendo en cuenta la etiqueta clase.

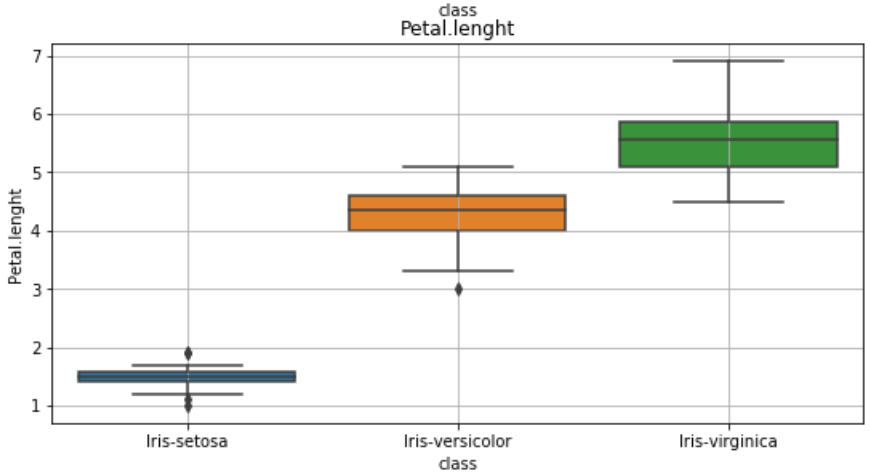


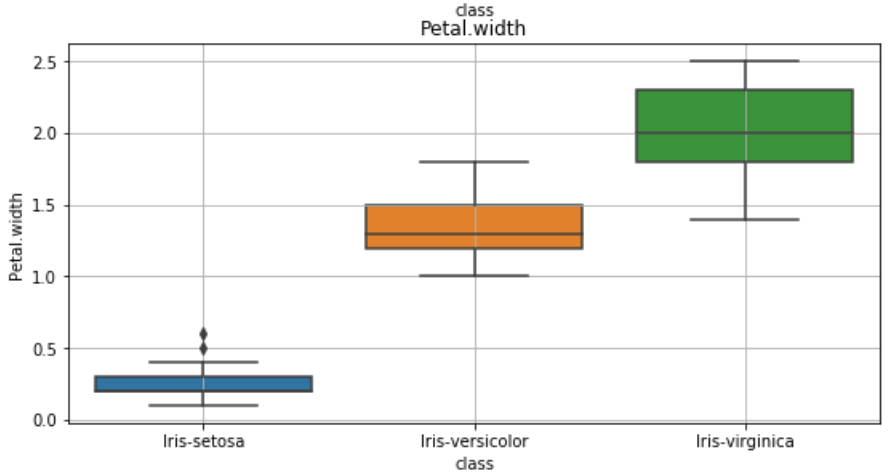
Boxplot por clase

- Nos permite comprender cómo se relaciona cada atributo con el valor de la clase.









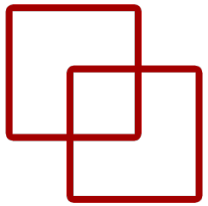
Métodos de transformación de datos

Métodos de transformación

preprocessing.Binarizer([threshold, copy])	Binarize data (set feature values to 0 or 1) according to a threshold
preprocessing.FunctionTransformer([func, …])	Constructs a transformer from an arbitrary callable.
preprocessing.KBinsDiscretizer([n_bins, …])	Bin continuous data into intervals.
preprocessing.KernelCenterer()	Center a kernel matrix
preprocessing.LabelBinarizer([neg_label, …])	Binarize labels in a one-vs-all fashion
preprocessing.LabelEncoder	Encode target labels with value between 0 and n_classes-1.
preprocessing.MultiLabelBinarizer([classes, …])	Transform between iterable of iterables and a multilabel format
preprocessing.MaxAbsScaler([copy])	Scale each feature by its maximum absolute value.
preprocessing.MinMaxScaler([feature_range, copy])	Transform features by scaling each feature to a given range.
preprocessing.Normalizer([norm, copy])	Normalize samples individually to unit norm.
preprocessing.OneHotEncoder([categories, …])	Encode categorical features as a one-hot numeric array.
preprocessing.OrdinalEncoder([categories, dtype])	Encode categorical features as an integer array.
preprocessing.PolynomialFeatures([degree, …])	Generate polynomial and interaction features.
preprocessing.PowerTransformer([method, …])	Apply a power transform featurewise to make data more Gaussian-like.
preprocessing.QuantileTransformer([…])	Transform features using quantiles information.
preprocessing.RobustScaler([with_centering, …])	Scale features using statistics that are robust to outliers.
preprocessing.StandardScaler([copy, …])	Standardize features by removing the mean and scaling to unit variance

preprocessing.add_dummy_feature(X[, value])	Augment dataset with an additional dummy feature.
preprocessing.binarize(X[, threshold, copy])	Boolean thresholding of array-like or scipy.sparse matrix
preprocessing.label_binarize(y, classes[, …])	Binarize labels in a one-vs-all fashion
preprocessing.maxabs_scale(X[, axis, copy])	Scale each feature to the [-1, 1] range without breaking the sparsity.
preprocessing.minmax_scale(X[, …])	Transform features by scaling each feature to a given range.
preprocessing.normalize(X[, norm, axis, …])	Scale input vectors individually to unit norm (vector length).
preprocessing.quantile_transform(X[, axis, …])	Transform features using quantiles information.
preprocessing.robust_scale(X[, axis, …])	Standardize a dataset along any axis
preprocessing.scale(X[, axis, with_mean, …])	Standardize a dataset along any axis
preprocessing.power_transform(X[, method, …])	Power transforms are a family of parametric, monotonic transformations that are applied to make data more Gaussian-like.

Escalamiento (I)

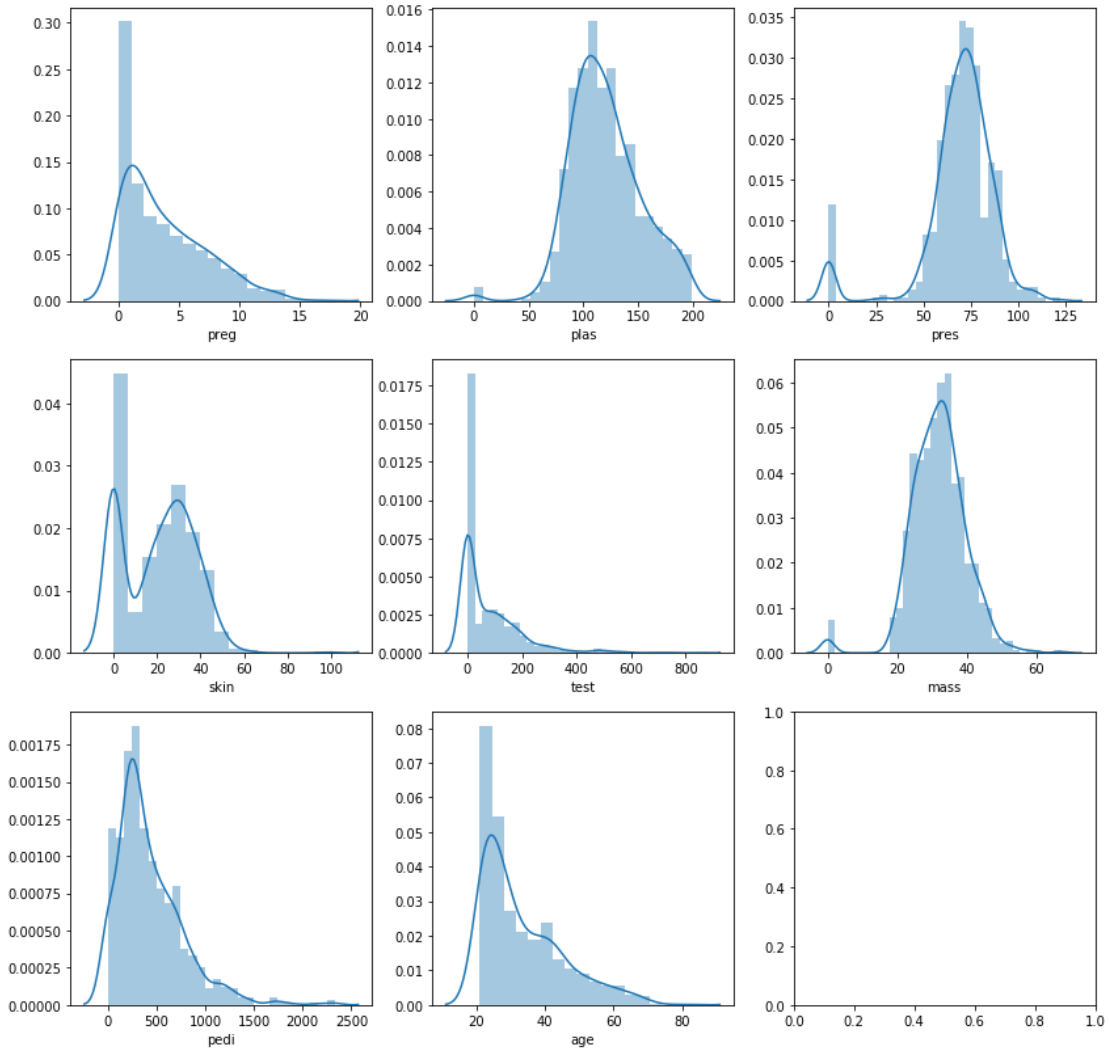


- Esta transformación es útil para los algoritmos de optimización utilizados en el núcleo de los algoritmos de aprendizaje automático como Gradiente Descendiente.
- También es útil para algoritmos que ponderan entradas como Regression y Neural Networks y algoritmos que usan medidas de distancia como k-Nearest Neighbours.
- Puede reescalar sus datos usando la clase *MinMaxScaler*.
- Después de reescalar puede ver que todos los valores están en el **rango [0,1]**.

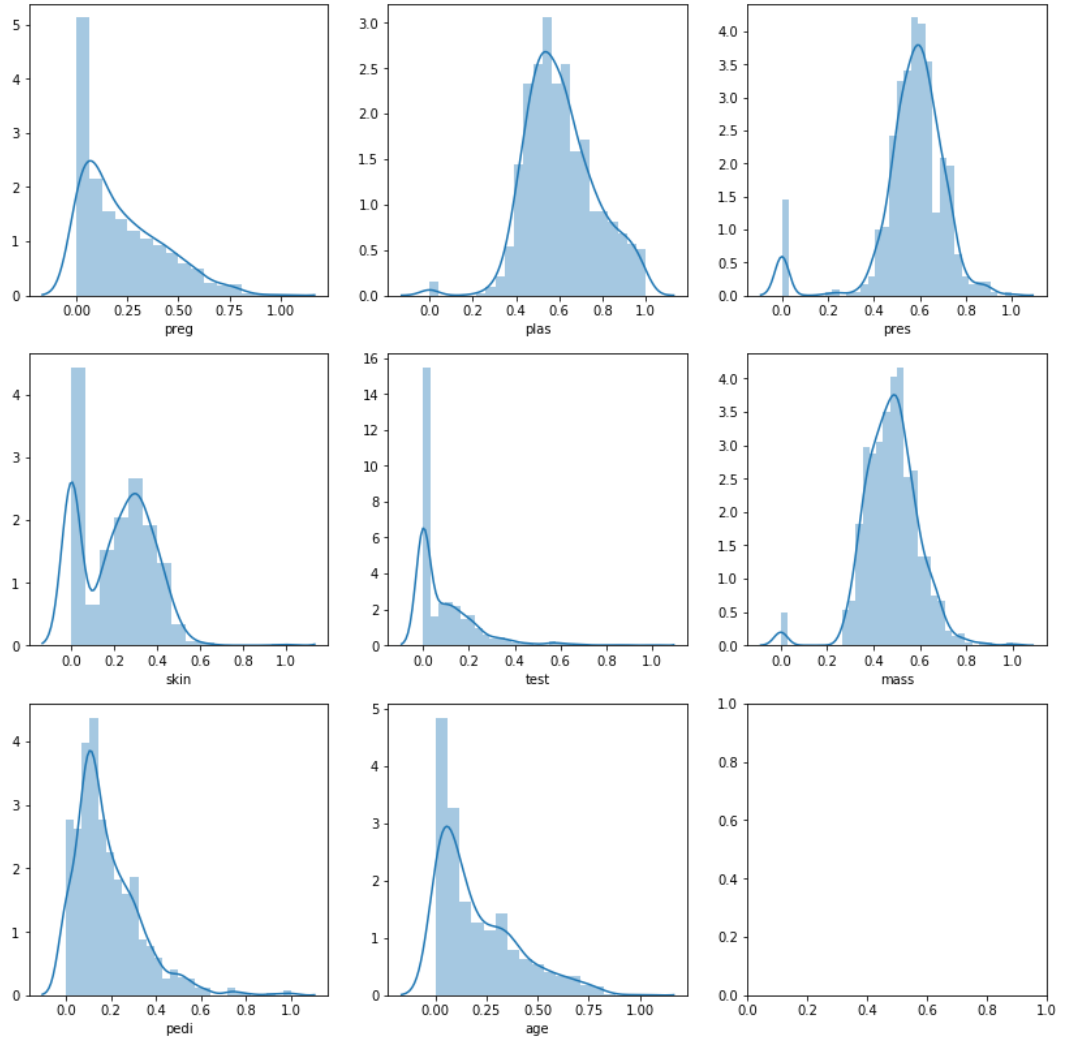
```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range=(0, 1))
rescaledX = scaler.fit_transform(X)
# summarize transformed data
np.set_printoptions(precision=3)
print(names)
print(rescaledX[0:5,:])
print(type(rescaledX))
```

```
['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
[[0.353 0.744 0.59  0.354 0.    0.501 0.269 0.483]
 [0.059 0.427 0.541 0.293 0.    0.396 0.151 0.167]
 [0.471 0.92  0.525 0.    0.    0.347 0.289 0.183]
 [0.059 0.447 0.541 0.232 0.111 0.419 0.072 0.   ]
 [0.    0.688 0.328 0.354 0.199 0.642 0.982 0.2  ]]
<class 'numpy.ndarray'>
```

Escalamiento (II)

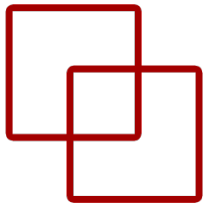


ANTES



DESPUÉS

Estandarización de datos (I)

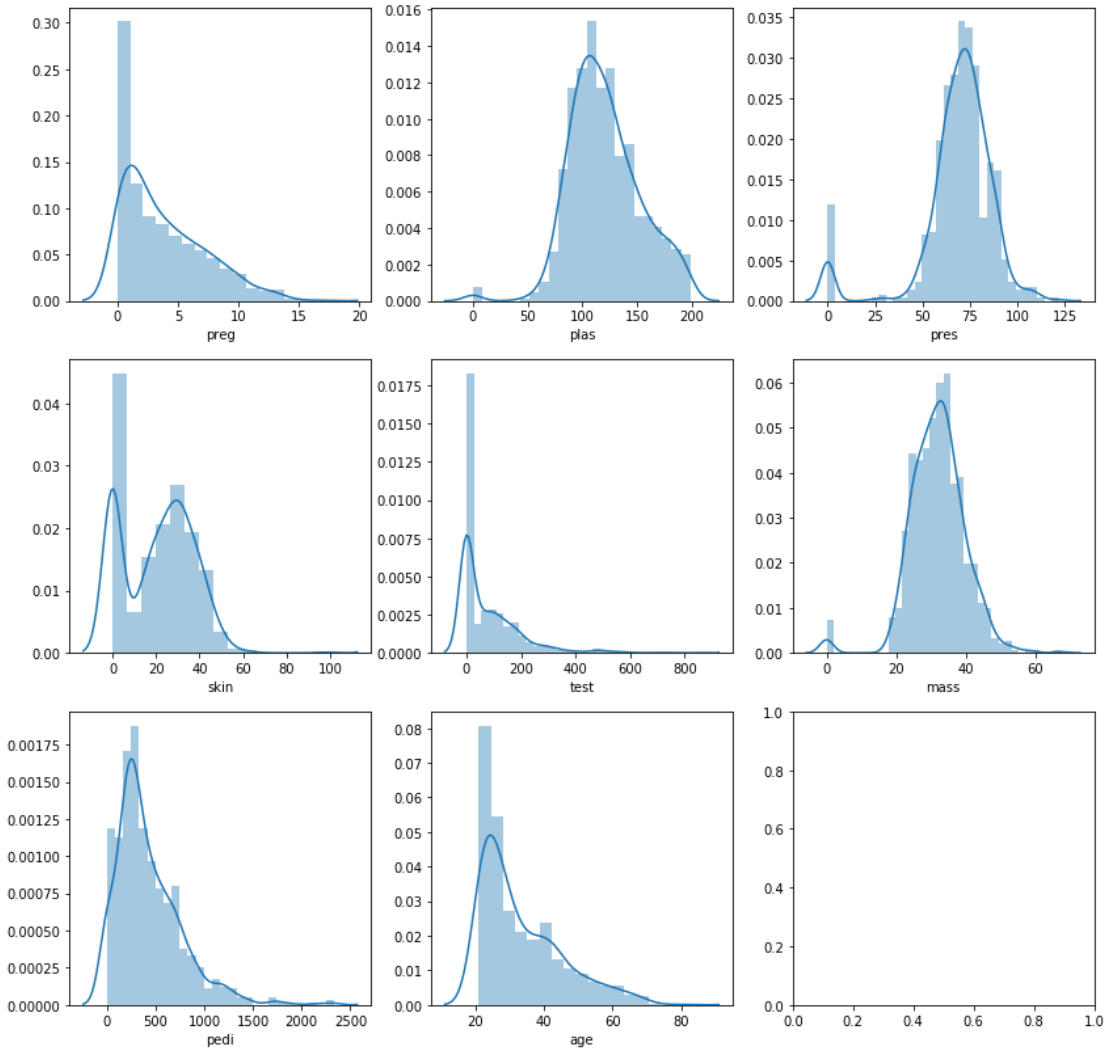


- Es más adecuada para técnicas que asumen una distribución gaussiana en las variables de entrada y funcionan mejor con datos reescalados, como LiR, LoR y LDA.
- Puede estandarizar datos utilizando la clase *StandardScaler*.
- Los valores **para cada atributo** ahora tienen un valor medio de 0 y una desviación estándar de 1.

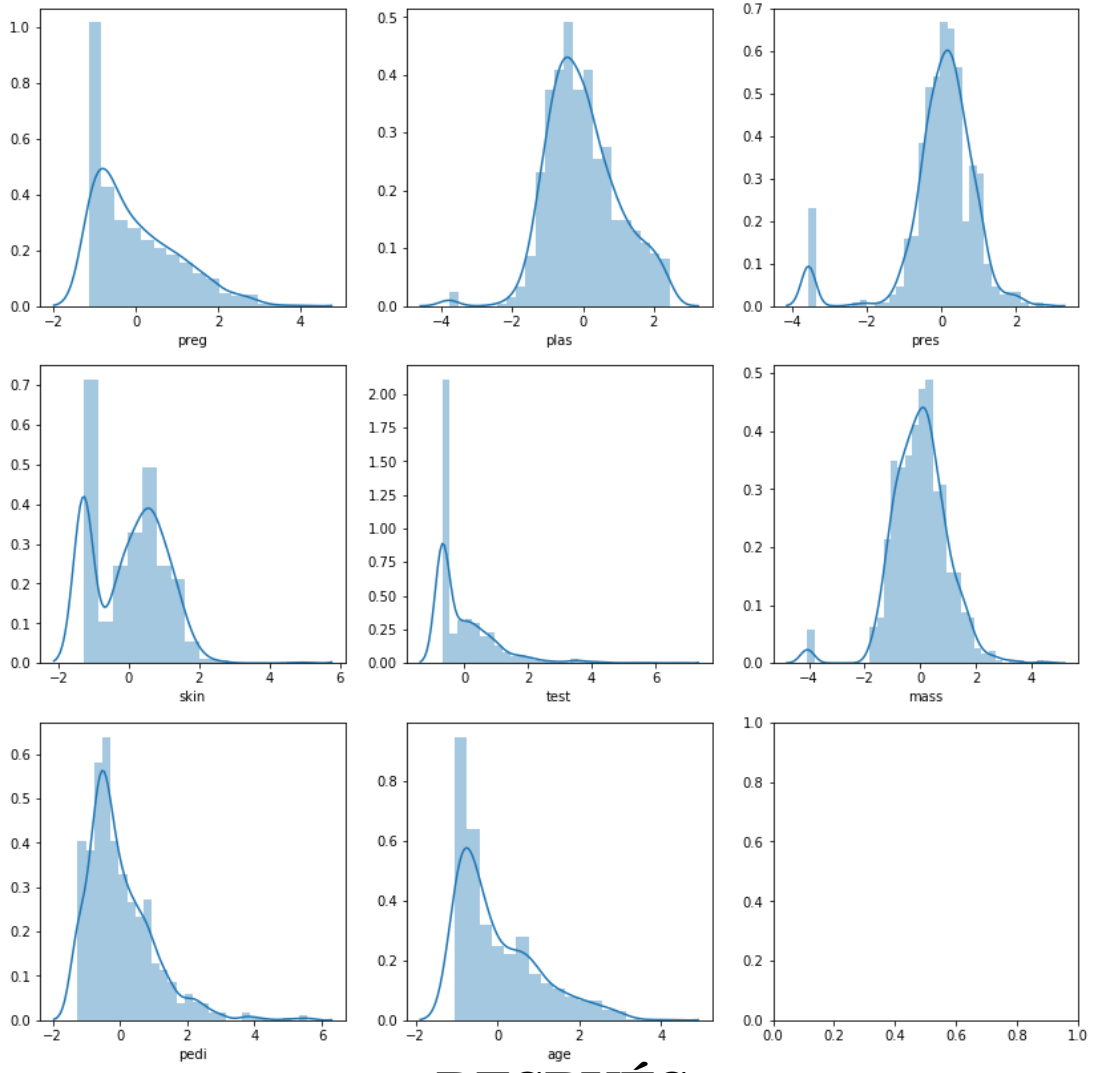
```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler().fit(X)
rescaledX = scaler.transform(X)
# summarize transformed data
np.set_printoptions(precision=3)
print(names)
print(rescaledX[0:5,:])
```

```
['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
[[ 0.64   0.848  0.15   0.907 -0.693  0.204  0.584  1.426]
 [-0.845 -1.123 -0.161  0.531 -0.693 -0.684 -0.227 -0.191]
 [ 1.234  1.944 -0.264 -1.288 -0.693 -1.103  0.716 -0.106]
 [-0.845 -0.998 -0.161  0.155  0.123 -0.494 -0.768 -1.042]
 [-1.142  0.504 -1.505  0.907  0.766  1.41   5.466 -0.02 ]]
```

Estandarización de datos (II)

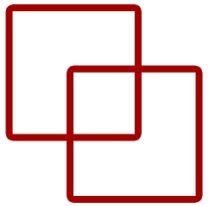


ANTES



DESPUÉS

Box-Cox (I)

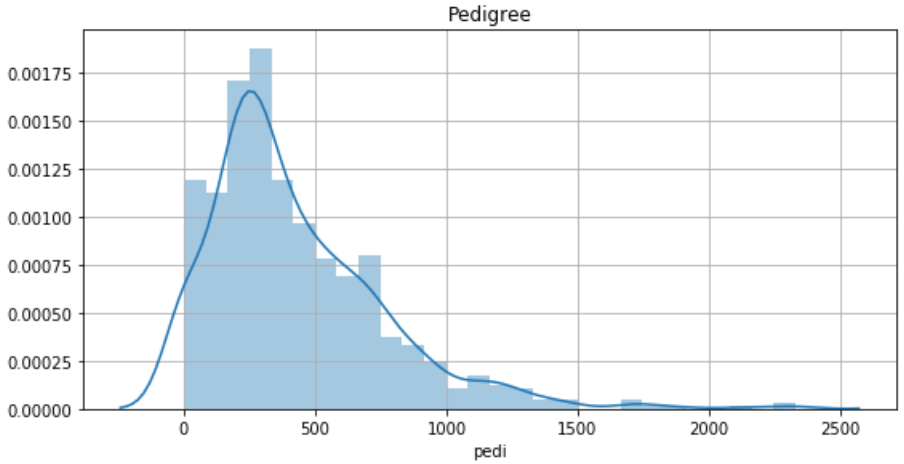


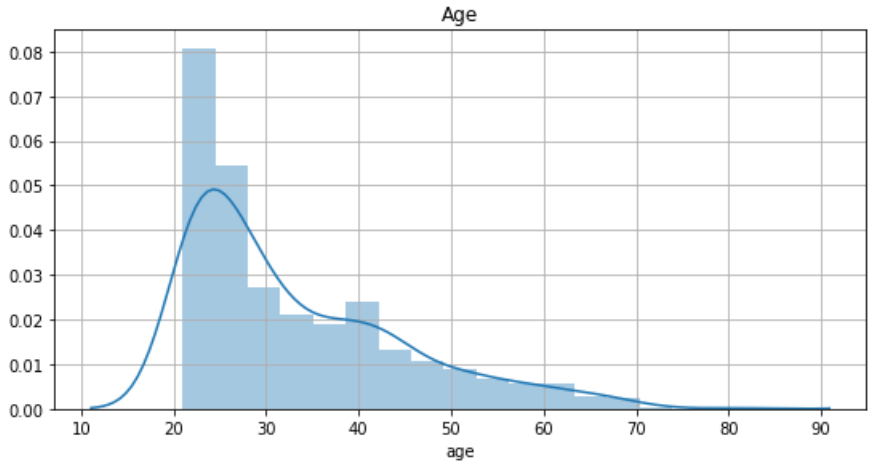
- Los atributos representan un sesgo o inclinación (Gaussiana desplazada).
  - Box-Cox asume todos los atributos **positivos**.
  - Aplica la transformación a los **atributos que parecen tener sesgo**.
  - Corrige la no linealidad en la relación (mejorar correlación entre las variables).

```
from sklearn.preprocessing import PowerTransformer
# extract features with skew
features = data[['pedi', 'age']]
#instatiate
pt = PowerTransformer(method='box-cox', standardize=True,)
#Fit the data to the powertransformer
skl_boxcox = pt.fit(features)
#Lets get the Lambdas that were found
#print (skl_boxcox.lambdas_)
calc_lambdas = skl_boxcox.lambdas_
#Transform the data
skl_boxcox = pt.transform(features)
#Pass the transformed data into a new dataframe
df_features = pd.DataFrame(data=skl_boxcox, columns=['pedi', 'age'])
# Pass to the original dataframe the transform columns
data.drop(['age'], axis=1, inplace=True)
data.drop(['pedi'], axis=1, inplace=True)
# Concatenar ambos dataframes
df_data = pd.concat([data, df_features], axis=1)
cols = df_data.columns.tolist()
# Pasa e útimo elemento al primero de la lista (2 veces)
cols = cols[-1:] + cols[:-1]
cols = cols[-1:] + cols[:-1]
# Sobreescrimos
df_data = df_data[cols]
df_data
```

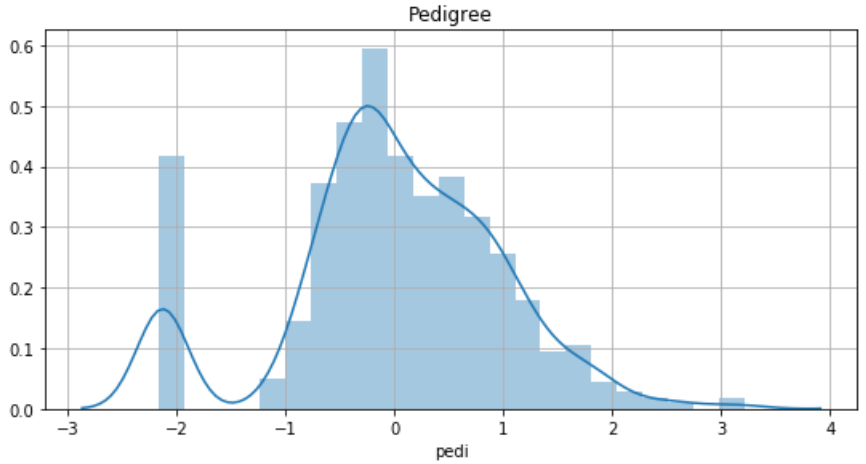
Box-Cox (II)

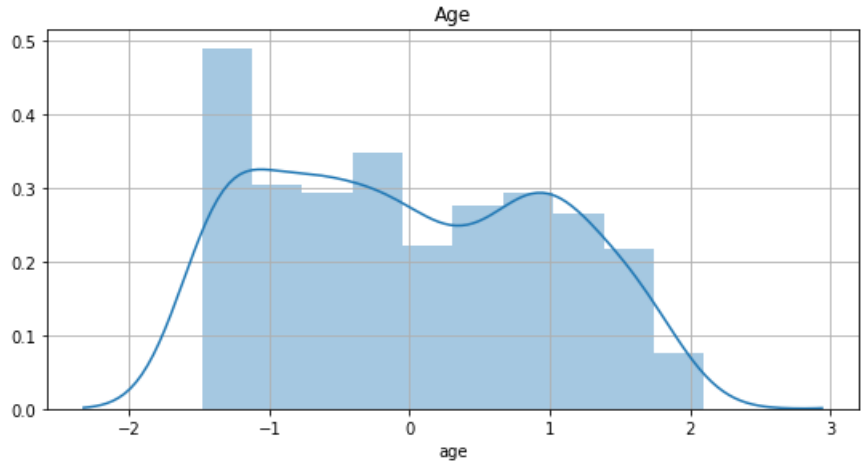
ANTES



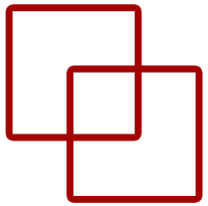


DESPUÉS





Yeo-Johnson (I)



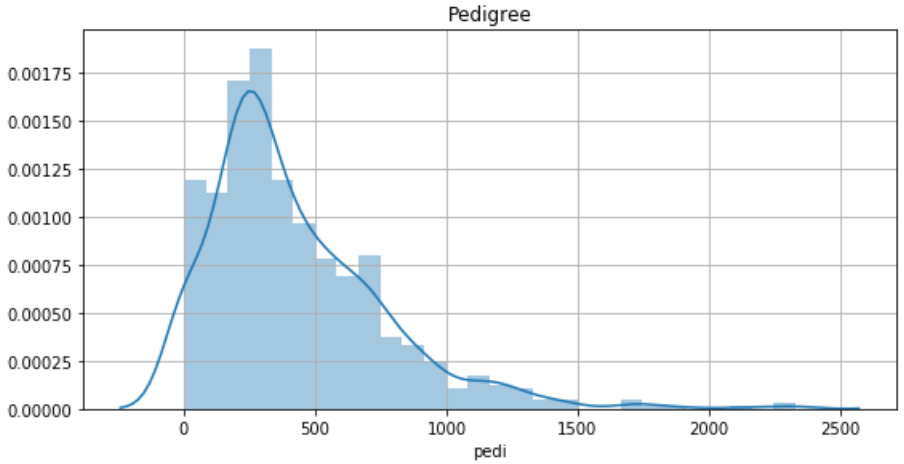
- Igual que Box-Cox pero soporta valores en bruto que son iguales a cero y negativos.

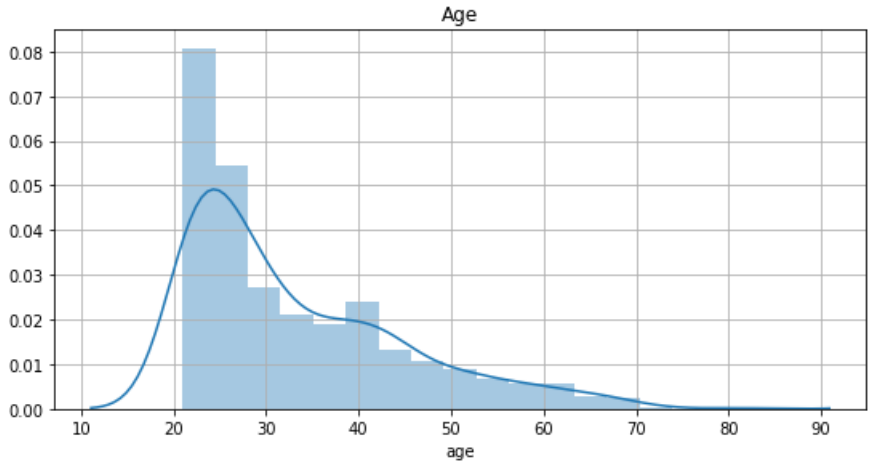
```
# extract features with skew
features = data[['pedi', 'age']]
#instatiate
pt = PowerTransformer(method='yeo-johnson', standardize=True,)
#Fit the data to the powertransformer
skl_boxcox = pt.fit(features)
#Lets get the Lambdas that were found
#print (skl_boxcox.lambdas_)
calc_lambdas = skl_boxcox.lambdas_
#Transform the data
skl_boxcox = pt.transform(features)
#Pass the transformed data into a new dataframe
df_features = pd.DataFrame(data=skl_boxcox, columns=['pedi', 'age'])

# Pass to the original dataframe the transform columns
data.drop(['age'], axis=1, inplace=True)
data.drop(['pedi'], axis=1, inplace=True)
# Concatenar ambos dataframes
df_data = pd.concat([data, df_features], axis=1)
cols = df_data.columns.tolist()
```

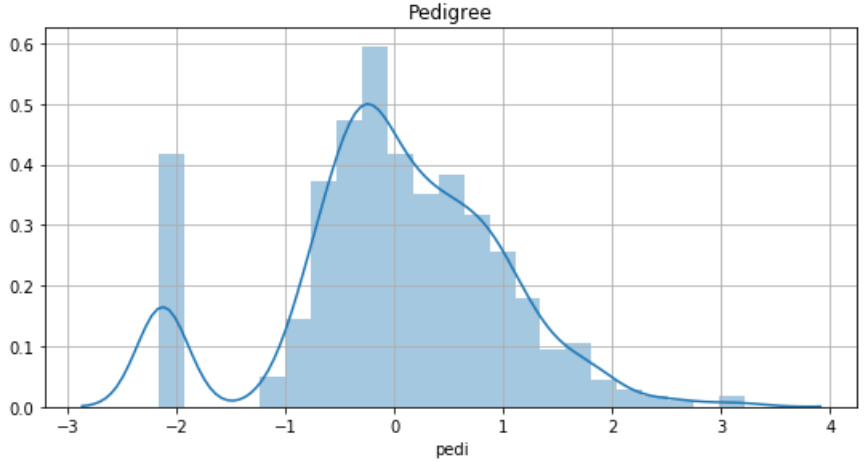
Yeo-Johnson (II)

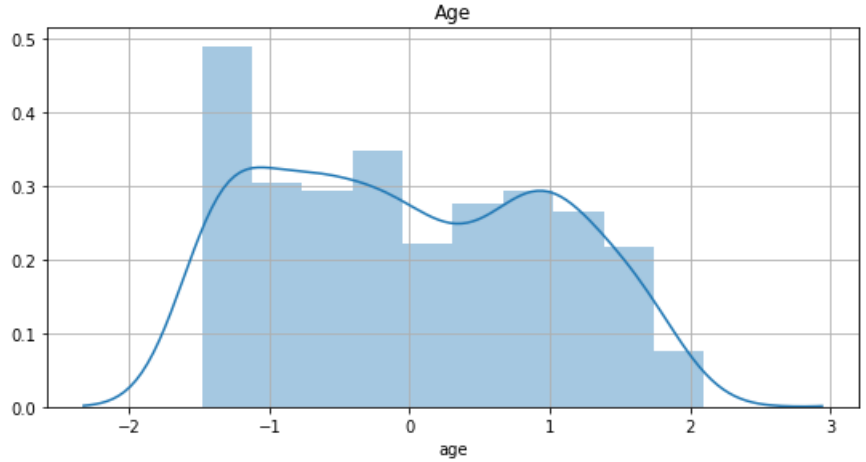
ANTES





DESPUÉS





¡GRACIAS!

Dr. Manuel Castillo-Cara
www.manuelcastillo.eu
Departamento de Inteligencia Artificial
Escuela Técnica Superior de Ingeniería Informática
Universidad Nacional de Educación a Distancia (UNED)