ANÁLISIS DE DATOS

Clase 5. Taller de preparación de datos (final)

Temario

1. Discretización
2. Outliers
3. Feature Scaling
4. Cadenas de procesamiento.
5. Transformación de variables.
6. Conceptos básicos de Teoría de la información
   - Entropía y entropía conjunta
   - Entropía relativa
   - Información mutua
7. Test estadísticos

DISCRETIZACIÓN

Definición

- Es el proceso de transformar variables continuas en variables con valores discretos mediante la creación de intervalos contiguos que cubren el espectro de valores que toma la variable.
- También se le llama **binning**, donde “bin” (en inglés “cesto”) es un nombre alternativo para un intervalo.

Motivación

- En preparación de datos para modelos de ML, es de interés discretizar por una o más de las siguientes razones:
  - algunas técnicas para discretizar permiten convertir distribuciones con oblicuidad en distribuciones normales o uniformes.
  - manejo de outliers: se pueden agrupar valores extremos en el primer y último intervalo.
  - al obtener una cantidad finita de valores posibles para una variable, puede combinarse con otros métodos, por ejemplo, codificación de variables categóricas.

Métodos de discretización

- No supervisados:
  - Ancho fijo
  - Frecuencia fija
  - K-means
- Supervisados:
  - Árboles de decisión

Discretización por intervalos de ancho fijo.
Definición y características.

- Divide el espectro de valores posible en N bins del mismo ancho.
- El tamaño de cada intervalo está dado por:
  - ancho = (max-min)/N
- Características:
  - No mejora la distribución.
  - Permite manejar valores extremos (dejan de ser extremos, y quedan asignados a los intervalos de de los extremos).
  - Crea una variable discreta.
  - Puede combinarse con codificaciones categóricas.

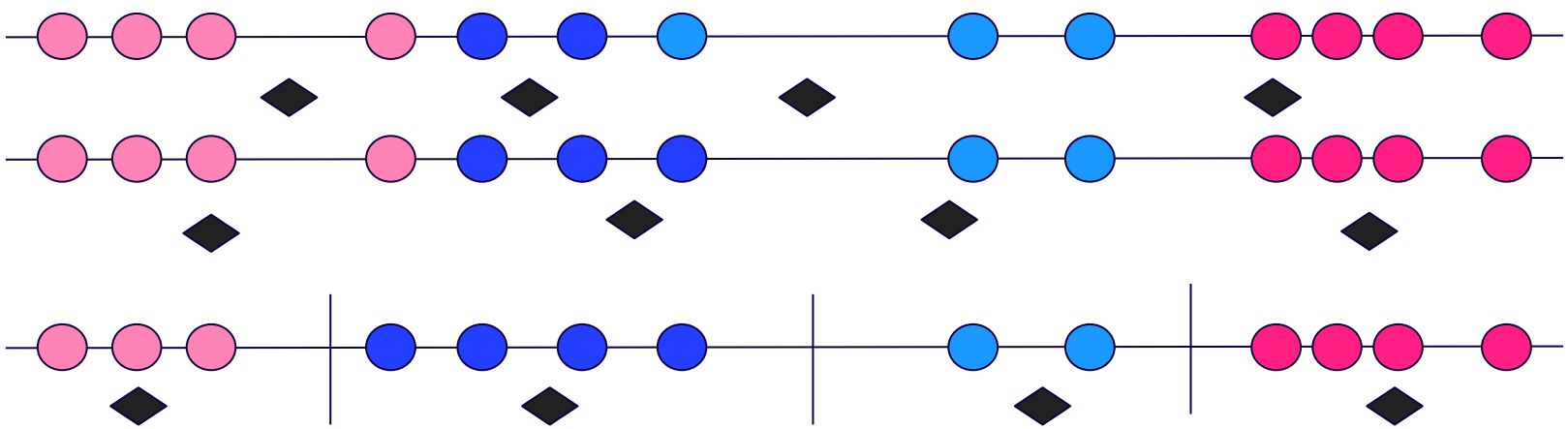
Discretización por intervalos de frecuencia fija.

Definición y características.

- Divide el espectro de valores posible en N bins, donde cada bin tiene aproximadamente la misma cantidad de observaciones.
- La definición de cada intervalo se realiza a partir de los cuantiles.
- Características:
  - Mejora la distribución.
  - Permite manejar valores extremos.
  - Crea una variable discreta.
  - Puede combinarse con codificaciones categóricas.

Discretización por K-means. Definición.

- Consiste en aplicar k-means para hallar n intervalos.



Discretización por K-means. Características.

- No mejora la distribución.
- Permite manejar valores extremos, si bien pueden influenciar la ubicación de los centroides.
- Crea una variable discreta.
- Puede combinarse con codificaciones categóricas.

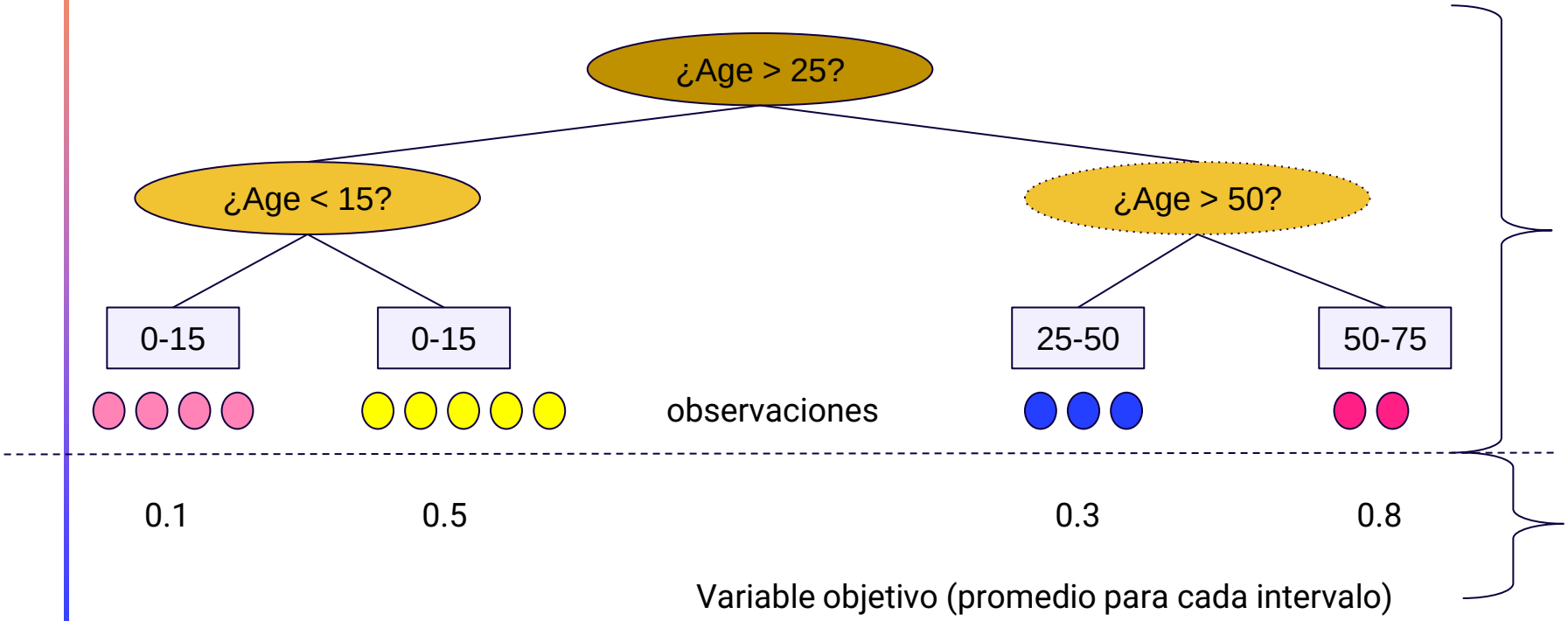
Discretización + codificación

- Mencionamos que la discretización es el proceso de representar un intervalo continuo como una variable de k intervalos (de 0 a k-1).
- Si se está construyendo un modelo no-lineal, entonces puede utilizarse esta nueva variable directamente.
- Si en cambio, se está trabajando con un modelo lineal se pueden interpretar los bins como si fueran categorías.
- En este último caso es posible aplicar las mismas técnicas vistas en codificación categórica para obtener una relación monotónica entre la variable independiente y la variable objetivo.

Discretización con árboles de decisión. Definición.

- Consiste en utilizar un árbol de decisión para hallar los bins óptimos.
- Una decisión de un árbol asigna una observación a una de sus N hojas.
- Los árboles generan una salida discreta, cuyos valores son las predicciones para cada una de sus N hojas.

Discretización con árboles de decisión. Ejemplo.



Tratamiento de valores extremos

- No mejora la distribución.
- Maneja outliers.
- Crea una variable discreta.
- Crea una relación monotónica.

Ejemplos en jupyter

Clase 4.6 - Preparación de datos - Discretización.ipynb

TRATAMIENTO DE VALORES EXTREMOS

Tratamiento de valores extremos. Motivación.

- En la sección 2 se introdujo el concepto de “outlier” y cómo identificarlo para distribuciones con y sin oblicuidad.
- Dado que los valores extremos pueden tener un efecto negativo en la capacidad de generalizar de algunos algoritmos, se expondrán algunas técnicas para su tratamiento y sus ventajas y desventajas.

Tratamiento de valores extremos. Métodos.

Poda

- Remover los valores extremos del dataset.

Datos faltantes

- Tratar los valores extremos como datos faltantes y aplicar métodos de imputación.

Discretización

- Ubicar los valores extremos en primer y último bin.

Censura

- Limitar valor (capping)
- Codificación extremo inferior/superior.
- Winsorization

Vistos en secciones 3 y 6.

Poda (trimming). Definición y características.

- Consiste en eliminar los outliers del dataset.
- Ventajas:
  - Facilidad de implementación, rapidez.
- Desventajas:
  - Pérdida de información.
  - En la mayoría de los casos se debe eliminar una observación completa aún cuando sólo falten valores en alguna de las variables.

Censura/limitar valor (capping). Definición y características.

- Consiste en reemplazar los outliers por el valor límite más cercano.
- Ventajas:
  - Facilidad de implementación, rapidez.
  - No es necesario eliminar una observación completa.
- Desventajas:
  - Distorsiona la distribución original.

Detección de valores extremos.

- Distribución normal → Media y desvío estándar.
- IQR, regla de proximidad.
- Cuantiles.

Ejemplos en jupyter

Clase 4.7 - Preparación de datos - Valores extremos.ipynb

FEATURE SCALING

Motivación. Cómo influye la magnitud de las variables de entrada (1 / 2)

- El coeficiente de regresión está directamente influenciado por la escala de la variable.
- Las variables con mayor rango de magnitud/valor dominan sobre las que tienen un menor rango de magnitud/valor.
- Los algoritmos de la familia de Gradient Descent convergen más rápidamente cuando las variables de entrada están en la misma escala.
- Las distancias euclídeas son sensibles a la magnitud de las variables.

Motivación Cómo influye la magnitud de las variables de entrada (2 / 2)

- Algoritmos afectados
  - Regresión lineal y logística.
  - Redes neuronales
  - Support Vector Machines
  - KNN
  - K-means
  - Principal Component Analysis (PCA)
- No afectados (por ejemplo, los basados en árboles)
  - Árboles de clasificación y regresión.
  - Random Forest
  - Gradient Boosted Trees.

Feature scaling (escalado de variables)

- En ML, se refiere a los métodos utilizados para normalizar el rango de valores que puede tomar una variable independiente.
- Suele ser el último paso en una cadena de procesamiento de datos (realizado justo antes del entrenamiento de algoritmos).

Feature scaling. Métodos

- Los más utilizados:
  - Estandarización
  - Escalado a mínimo-máximo.
- Otros métodos:
  - Normalización de media.
  - Escalado a valor absoluto.
  - Escalado a mediana y cuantiles.
  - Escalado a norma unitaria.

Estandarización

- Centra la variable en cero y establece su varianza a 1.

$$Z = \frac{x - \mu}{\sigma}$$

Estandarización. Efecto

- Centra la media en 0.
- Escala la varianza a 1.
- Preserva la forma de la distribución original.
- Los mínimos y máximos pueden variar.
- Preserva los valores extremos (outliers).

Escalado a mínimo-máximo.

- Esta transformación restringe el valor de la variable a un rango (típicamente x_min=0,x_max=1).

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

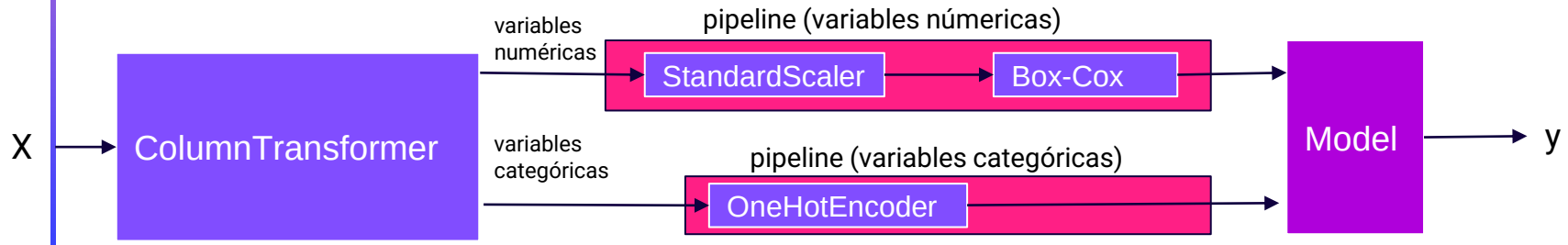
Escalado a mínimo-máximo. Efecto.

- La media puede variar.
- La varianza puede variar.
- Puede modificar la forma de la distribución original.
- Los mínimos y máximos quedan restringidos.
- Preserva los valores extremos (outliers).

CADENAS DE PROCESAMIENTO

Implementación de cadenas de procesamiento con SKLearn

- Arquitectura de SKlearn:
  - **Transformer**: clase para transformar datos. Debe implementar fit() y transform().
  - **Predictor:** clase para realizar predicciones. Debe implementar fit() y predict().
  - **Pipeline:** clase que ejecuta secuencia de transformers y/o predictors.
    - Cada elemento de la lista es un paso (step).
    - El único elemento de la lista que puede ser predictor es el último (los predecesores deben ser transformers).
  - **Columntransformer**: clase que permite aplicar transformaciones específicas para cada columna de un dataset.



Implementación de cadenas de procesamiento con SKLearn

- Ejemplos de uso de clases **Pipeline** y **ColumnTransformer** para problemas de clasificación y regresión.
- Referencias:
  - https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html
  - https://scikit-learn.org/stable/modules/compose.html#columntransformer-for-heterogeneous-data
  - Sección de trransformaciones de datos de documentación de SKLearn: https://scikit-learn.org/stable/data_transforms.html

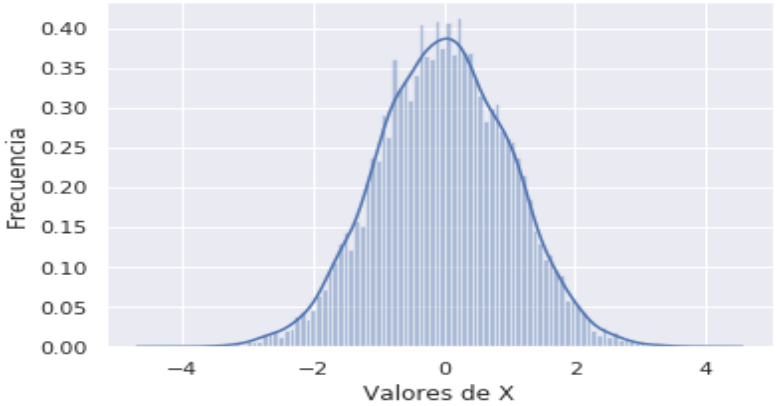
Ejemplos en jupyter

Clase 4.8 - Preparación de datos - Cadenas de procesamiento.ipynb

TRANSFORMACIÓN DE VARIABLES

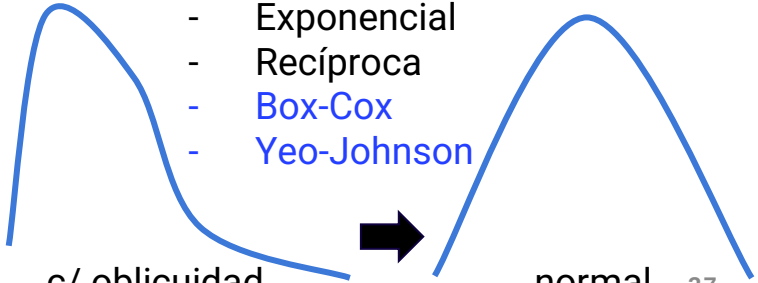
Distribución normal en modelos lineales

- Es deseable que los valores de cada variable independiente (X) tengan una distribución normal.



- Es muy común que esto no se cumpla en los datos originales. En este caso, puede intentar obtenerse una distribución más semejante a una normal luego de aplicar una transformación.

- Logarítmica
- Exponencial
- Recíproca
- Box-Cox
- Yeo-Johnson



Transformaciones matemáticas

- Logarítmica: $\log(x)\,, x > 0$
- Recíproca: $\frac{1}{x}, \forall x \in \mathbb{R} - \{0\}$
- Potencia/exponencial
  - $X\ e^{\lambda}$
  - $X^{1/2}/X^{3}$
  - No definido para todo X.
- Exponencial (casos especiales):
  - *Box-Cox*, X>0
  - *Yeo-Johnson*

Transformación de Box Cox

- La transformación de Box-Cox estima un valor de lambda que minimiza la desviación estándar de una variable transformada estandarizada.
- El método de Box-Cox busca entre muchos tipos de transformaciones.

$$y_i^{(\lambda)} = \begin{cases} \dfrac{y_i^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0, \\ \ln(y_i) & \text{if } \lambda = 0, \end{cases}$$

L	Y'
-2	$Y^{-2} = 1/Y^2$
-1	$Y^{-1} = 1/Y^1$
-0.5	$Y^{-0.5} = 1/(\text{Sqrt}(Y))$
0	log(Y)
0.5	$Y^{0.5} = \text{Sqrt}(Y)$
1	$Y^1 = Y$
2	Y^2

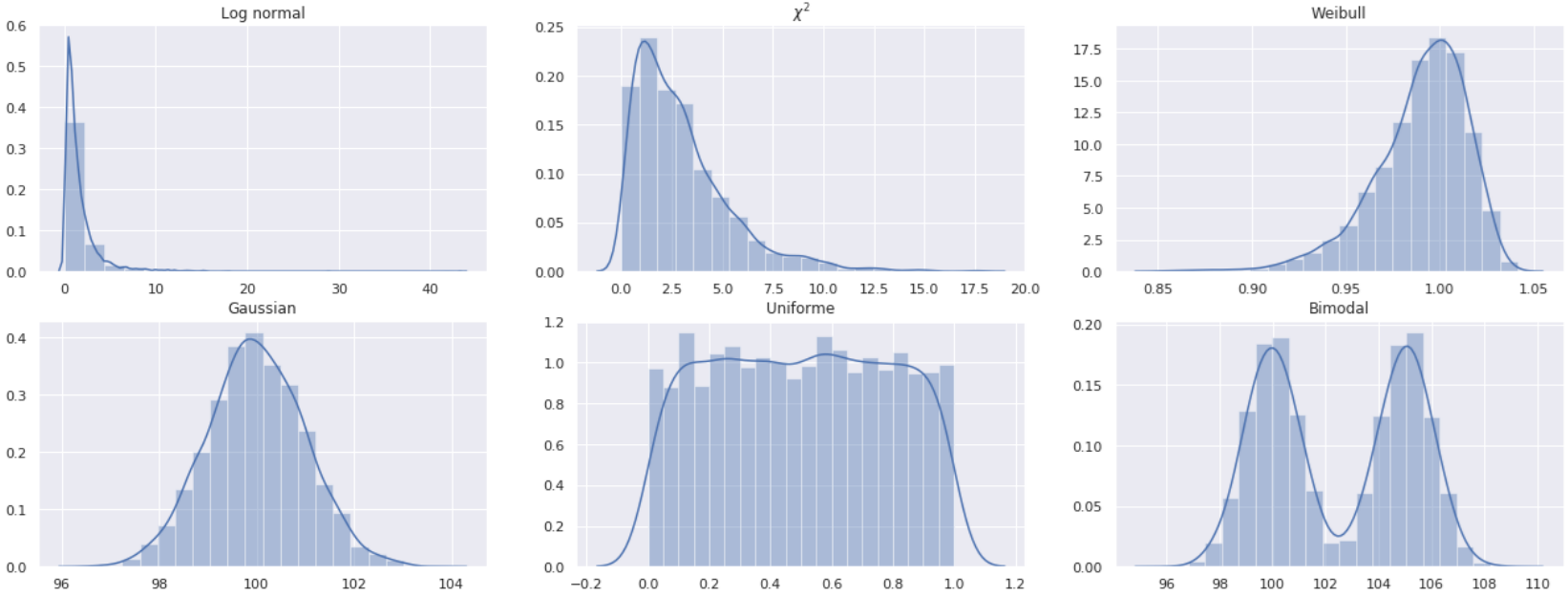
Source: Box and Cox (1964). Where, 'Y' is the transformation of t
Note that for Lambda = 0, the transformation is NOT Y^0 (because t
every

Transformación de Yeo-Johnson

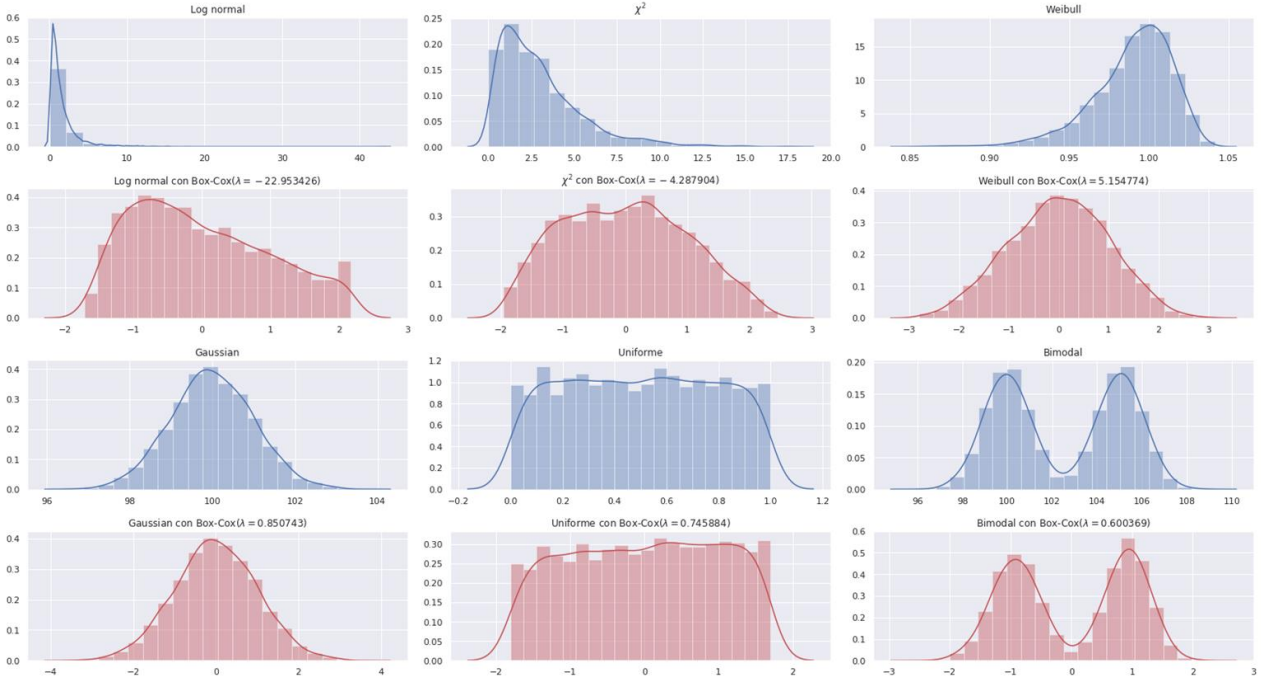
- Junto con Box-Cox, es otra de las transformaciones más utilizadas.
- Admite que X tome valores negativos.

$$y_i^{(\lambda)} = \begin{cases} ((y_i + 1)^{\lambda} - 1)/\lambda & \text{if } \lambda \neq 0, y \geq 0 \\ \log(y_i + 1) & \text{if } \lambda = 0, y \geq 0 \\ -[(-y_i + 1)^{(2-\lambda)} - 1]/(2 - \lambda) & \text{if } \lambda \neq 2, y < 0 \\ -\log(-y_i + 1) & \text{if } \lambda = 2, y < 0 \end{cases}$$

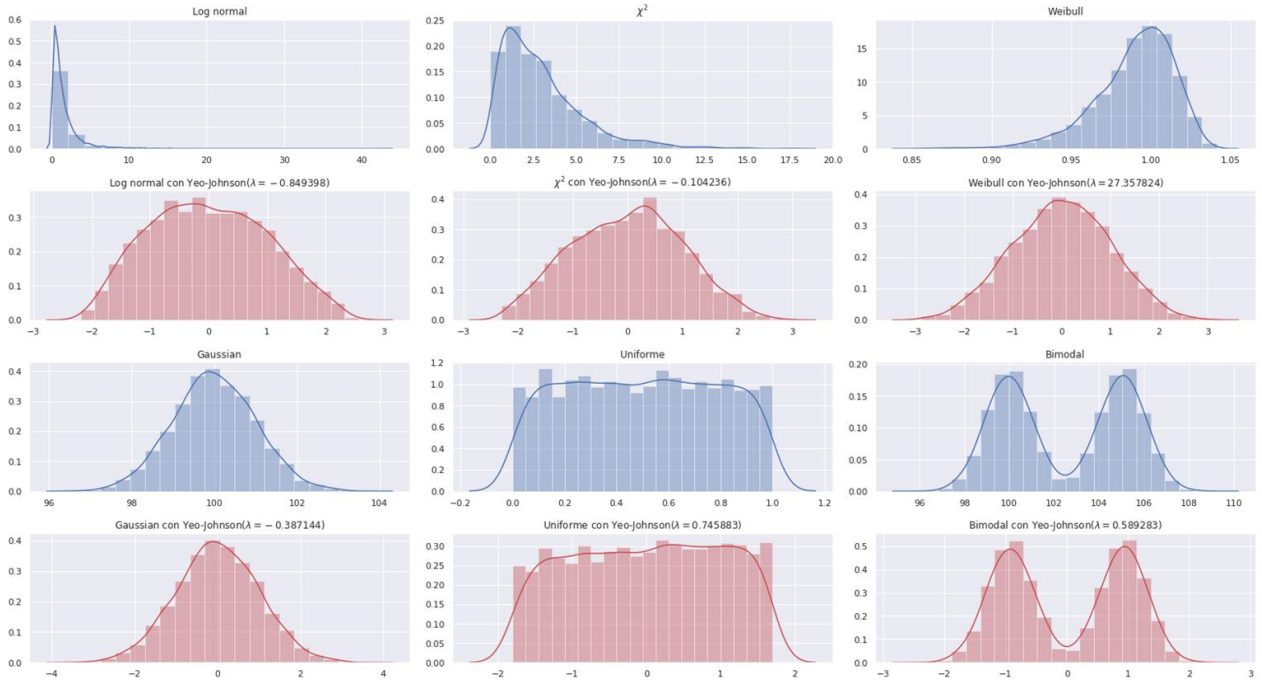
Ejemplos



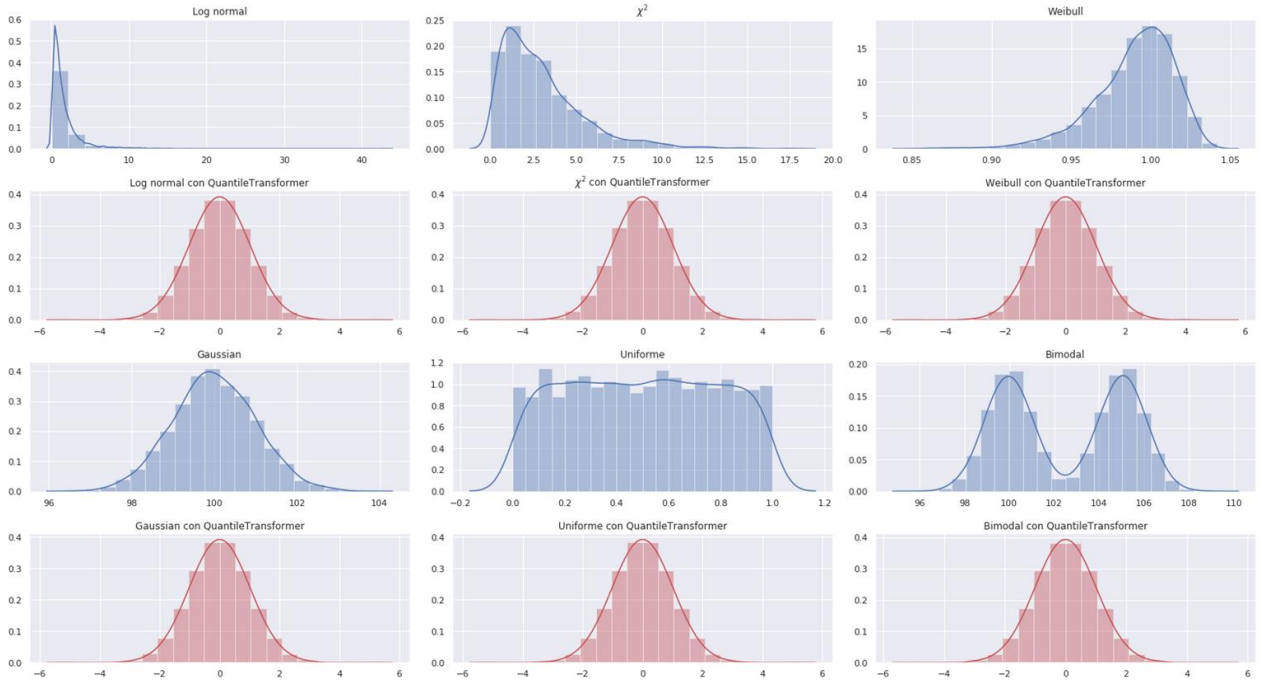
Ejemplos. Box-Cox



Ejemplos. Yeo-Johnson



Ejemplos. QuantileTransformer



Ejemplos en jupyter

Clase 4.5 - Preparación de datos - Transformación de variables.ipynb

Resumen

El objetivo de estas clases fue presentar algunas de las técnicas más utilizadas para la preparación de datasets y cómo combinarlas para construir cadenas de procesamiento en SKLearn.

En las siguientes clases, se estudiarán otros aspecto de la **preparación de datos**:

- La **selección de variables** de entrada: ¿Qué variables son más relevantes para el modelo de AA?
- **Ingeniería de variables** (como generar variables que aporten mayor información, a partir de las existentes).
- **Reducción de dimensiones**. Creación de un nuevo espacio de variables entrada, a partir de proyecciones compactas.

DUDAS?

ENCUESTA