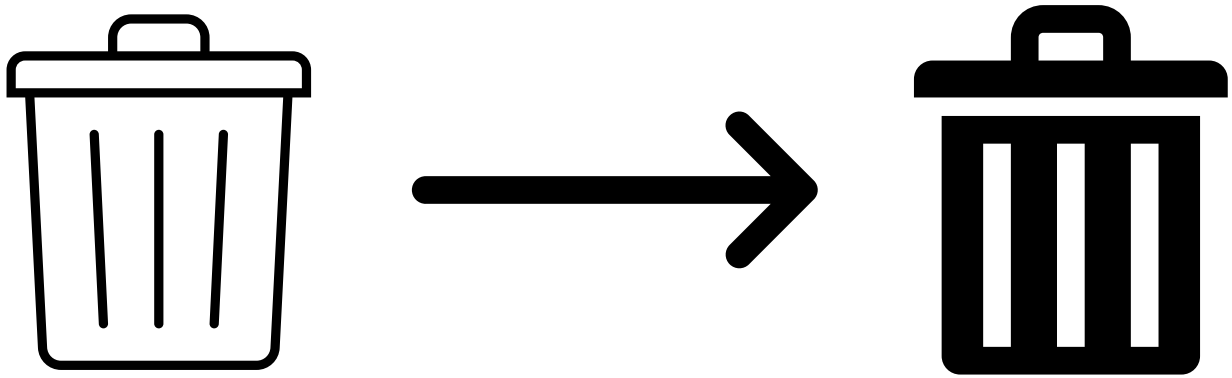
Generación de variables

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

José Ramón Sánchez Leo
rsanchez@afi.es

Febrero 2022
Afi Escuela

Garbage in, garbage out



***Es así siempre, no lo olvides cuando trabajes con datos ;)**

Índice

1. Introducción y ejemplos generales

¿Qué es la generación de variables?

La **generación de variables** en aprendizaje automático es el proceso destinado a construir los mejores predictores posibles para la construcción de un buen modelo predictivo.

Este proceso suele considerarse como una parte del preprocesado de datos, y existen técnicas comunes:

- Suavizado de variables ya existentes.
- Discretización de variables numéricas.
- Agrupación de niveles en variables categóricas.
- Normalización de variables numéricas.
- Combinación de variables.

¿Cómo proceder?

1. *Brainstorming* de variables.
2. Generar variables surgidas en el punto anterior.
3. Entrenar modelos con las variables generadas y comprobar su funcionamiento.
4. Mejorar las variables si fuese necesario.
5. Repetir el proceso hasta que se alcance el rendimiento deseado.

Algunas ideas comunes

- Descomposición de atributos de variables categóricas.
  - Color: Rojo, Azul, Desconocido → Tiene_Color, Es_Rojo, Es_Azul: sí/no.
- Agrupación de niveles de variables categóricas.
- Replantearse las variables numéricas: discretizar, redondear, fijar fronteras, …
- Componentes principales.
- Aplicar funciones a las variables existentes: logaritmos, potencias o interacciones entre ellas.
- Descomposición de variables temporales.
- Variables temporales: cambiar una fecha por el tiempo transcurrido. Aplicar seno y coseno para trabajar con métricas temporales.
- Convertir variables categóricas con orden en numéricas.

- **El conocimiento del negocio es muy importante.**

2. Construcción automática de variables

Métodos populares

- **Deep feature synthesis**: algoritmo de fuerza bruta que genera variables de forma automática para conjuntos de datos de tipo relacional. Al enlazar a este algoritmo algunos modelos de *machine learninng* no demasiado complicados, se ha conseguido un rendimiento más que aceptable en Kaggle.

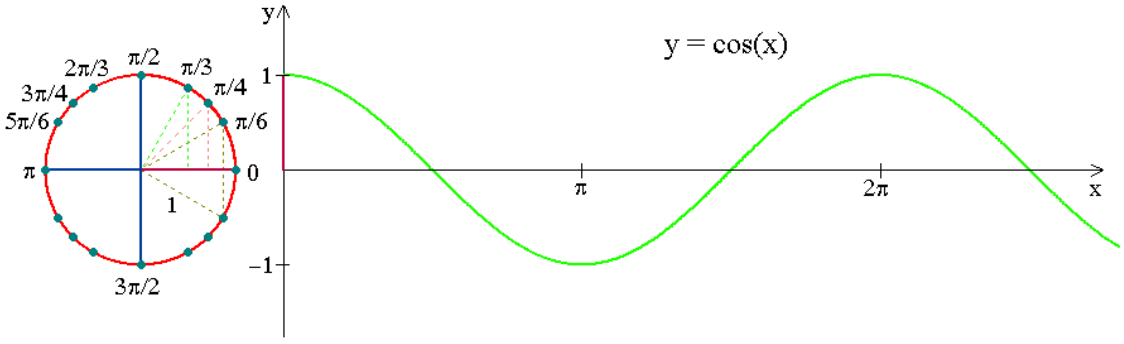
- **Autoencoders:** red neuronal empleada para aprender representaciones eficientes de un determinado conjunto de datos. Históricamente, se han empleado en reducción de dimensiones, pero recurrentemente se emplean para aprender *data generation processes*.

- Emplear salidas de modelos sencillos como predictores.

3. Variables temporales

Descomposición de variables temporales

- Podemos extraer las diferentes características de una variable temporal: año, mes, semana, día, día de la semana, hora, …
- Generar variables transformando por tiempo transcurrido para calcular: antigüedad, días hasta, días desde, …
- Utilizar seno y coseno para conservar distancias temporales.



$$X' = 2\pi \frac{X}{\max(X)}$$

Ejemplo

01_Seleccion_Generación__Temporales.ipnyb

4. Variables categóricas

Métodos populares

- Agrupación de niveles.
- One-Hot-Encoder: generar una variable binaria por cada categoría de la variable.
- Label Encoder: asignar un número entero a cada categoría.
- Target Encoder:
  - Objetivo categórico: se sustituyen por la probabilidad del objetivo de dado un valor categórico particular respeto a la probabilidad a priori sobre todos los datos de entrenamiento.
  - Objetivo continuo: se sustituyen por el valor esperado del objetivo dado un valor categórico específico y el valor esperado del objetivo sobre todos los datos de entrenamiento.
- Imputar por proxy numérico
- **Conocimiento de negocio**
- Podéis encontrar más encoders categóricos en: https://contrib.scikit-learn.org/category_encoders/index.html

Micci-Barreca, D. (2001) A preprocessing scheme for high-cardinality categorical attributes in classification and predictions problems.

Ejemplo

02_Seleccion_Generación__Categoricas.ipnyb

5. Datos secuenciales

Estructuras secuenciales

Cuando las observaciones del set de datos sobre el que se quiere entrenar un modelo tienen estructura temporal (e.j., cada observación corresponde a un día).

En este tipo de situaciones, es más que recomendable aprovecharse de la estructura temporal de la información (emplear la información del pasado para predecir el futuro):

- Utilizar retardos (¿cuántos?) de las variables (y de la salida) como predictores.
- Usar promedios de algunas de las variables en una cierta ventana temporal y sus variaciones como predictores.
- Ajustar modelos agregados para reducir el ruido.

The sliding window method

El método de la ventana deslizante convierte un problema de aprendizaje supervisado con datos secuenciales en un problema "sin estructura temporal".

Si $(x_t, y_t), t = 1, \dots, T$, son patrones disponibles, fijado un ancho de ventana p, se convierte el problema de predecir y_t en términos de x_t, en predecir y_t en términos de $x_t, x_{t-1}, \dots, x_{t-p+1}$.

Hecho esto, se puede emplear cualquiera de los algoritmos de aprendizaje automático para ajustar el modelo predictivo.

x	y
x_1	y_1
x_2	y_2
x_3	y_3
...	...
x_{t-1}	y_{t-1}
x_t	y_t

x_t	x_{t-1}	x_{t-2}	y_t
x_3	x_2	x_1	y_3
x_4	x_3	x_2	y_4
x_5	x_4	x_3	y_5
...	...	...	...
x_{t-1}	x_{t-2}	x_{t-3}	y_{t-1}
x_t	x_{t-1}	x_{t-2}	y_t

The recurring sliding window method

El método de la ventana deslizante recurrente es una modificación del anterior en el que, además, de emplear $x_t, x_{t-1}, \dots, x_{t-p+1}$ como predictores de y_t, se incluye en valor $\hat{y}_{t-1}$ predicho por el modelo para y_{t-1}.

Otras modificaciones son:

- Alargar la ventana temporal en y.
- Emplear valores y_{t-1} en lugar de $\hat{y}_{t-1}$.
- Incluir predicciones de $\hat{y}_{t-1}$ arrojadas por un modelo de predicción de series temporales.

x	y
x_1	y_1
x_2	y_2
x_3	y_3
...	...
x_{t-1}	y_{t-1}
x_t	y_t

x_t	x_{t-1}	x_{t-2}	$\tilde{y}_{t-1}$	y_t
x_3	x_2	x_1	$\tilde{y}_2$	y_3
x_4	x_3	x_2	$\tilde{y}_3$	y_4
x_5	x_4	x_3	$\tilde{y}_4$	y_5
...	...	...	...	...
x_{t-1}	x_{t-2}	x_{t-3}	$\tilde{y}_{t-2}$	y_{t-1}
x_t	x_{t-1}	x_{t-2}	$\tilde{y}_{t-1}$	y_t

Ejemplo

03_Seleccion_Generación__Secuenciales.ipnyb

6. Datos no estructurados

Cuando no existe estructura…

Generar variables también quiere decir dar estructura a datos no estructurados. Si cada una de nuestras observaciones es una imagen, un texto o un audio, generar variables es realizar las mediciones oportunas para convertirlas en una fila de una tabla.

En la inmensa mayoría de los casos, esto se traduce en:

- Realizar recuentos.
- Calcular proporciones.
- Determinar la presencia o no presencia de objetos concretos.

Imágenes

Una imagen es una matriz $m\ x\ n$ a cuyas celdas se denomina **píxeles**. Cada píxel es un cuadradito de la imagen, y su valor en la matriz sirve para especificar su contenido.

Una imagen RGB es una superposición de tres matrices $m\ x\ n$, con entradas en $[0,1]$, que especifican la intensidad de los canales rojo (R), verde (G) y azul (B) para cada píxel, de modo que la imagen es una suma de las tres. En ocasiones, las imágenes podrán tener una matriz adicional, siendo RGBA, donde A hace referencia al coeficiente *alpha*.

Al convertir la imagen en estas tres matrices, cualquier propiedad de las distribuciones de los tres canales se puede emplear como variable, pero se pierde información espacial.

Imágenes

El análisis de los tres canales sirve para reducir el ruido de la imagen: la eliminación de los *outliers* en las distribuciones de cualquiera de los colores suele producir imágenes más puras.

Hay otro tipo de técnicas que se emplean con cierta frecuencia:

- **Mean filtering:** sustituir el valor de cada punto por la media de los que le rodean.
- **Gaussian filtering**: sustituir el valor de cada punto por una media de los que le rodean respecto de un núcleo gaussiano (es decir, ponderando respecto a e^{-d^2} , donde d es la distancia de cada punto al centro).

$$\frac{1}{16} \times \begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array} \qquad \frac{1}{273} \times \begin{array}{|c|c|c|c|c|}\hline 1 & 4 & 7 & 4 & 1 \\ \hline 4 & 16 & 26 & 16 & 4 \\ \hline 7 & 26 & 41 & 26 & 7 \\ \hline 4 & 16 & 26 & 16 & 4 \\ \hline 1 & 4 & 7 & 4 & 1 \\ \hline \end{array}$$

Imágenes

Para tener en cuenta propiedades espaciales de la imagen (qué cosas están próximas, rotadas o reescaladas pueden ser iguales, etc.) se emplean multitud de técnicas. Se pueden destacar las siguientes:

- Detección de bordes.
- Detección de blobs.
- Template matching.

Detección de bordes

Las técnicas de detección de bordes se dedican a identificar puntos de una imagen donde el brillo cambia drásticamente o presenta discontinuidades.

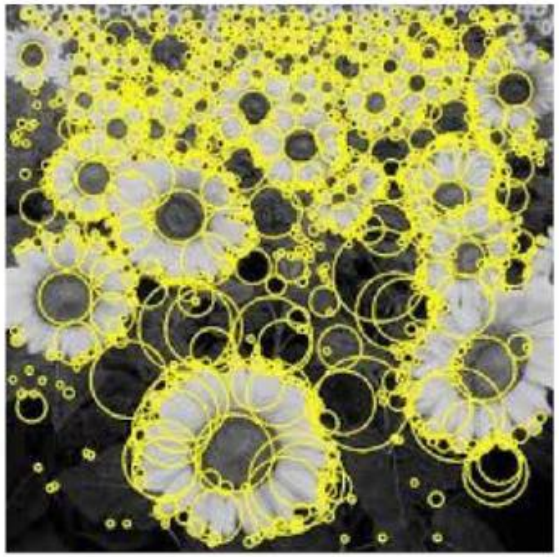
Estas discontinuidades suelen organizaciones en curvas que delimitan zonas distinguidas de la imagen, cuyo comportamiento puede estudiarse por separado.

Una vez detectadas estas, si el objetivo del proceso es una zona en concreto, suelen aplicarse mascaras para aislarla.

Detección de blobs

Las técnicas de detección de blobs (regiones) identifican zonas que se distinguen de los puntos de su alrededor en propiedades como brillo o color, de tal modo que algunas de estas propiedades son constantes (o aproximadamente constantes) a lo largo de la región.

Se emplean, generalmente, para detector aquellas regiones que no son captadas por algoritmos de detección de bordes.



Template matching

Es un algoritmo de fuerza bruta que busca los trozos de una imagen que son más parecidos a un patrón dado. Se coloca el centro del patrón por encima de cada píxel de la imagen y se elige aquel que maximice alguna medida de similitud.

Estas técnicas no tienen en cuenta que imágenes muy parecidas pueden parecer muy diferentes si no se les aplican anteriormente algunas transformaciones (rotaciones, reescalado, etc.).

https://docs.opencv.org/4.x/d4/dc6/tutorial_py_template_matching.html

Deep Learning

Las redes neuronales profundas (*deep learning*) han demostrado gran potencia en problemas de reconocimiento de imágenes, precisamente porque el usuario no necesita realizar la generación de variables.

La red acepta como entrada cada imagen tal cual junto con multitud de transformaciones de ella, y detecta variables para discriminar (aunque no serán interpretables).

Texto

Un texto es una sucesión (finita) de palabras.

Seguramente, la forma más natural de convertir cada texto en una fila de una tabla sea identificar la n-ésima palabra del texto con la n-ésima variable, pero esto no funciona (ni por asomo).

Las alternativas razonables que si funcionan en muchos casos son las representación *Bag of Words* y el tf-idf. El punto de partida es similar: dadas todas las palabras p del lenguaje, tendremos una variable para cada palabra p.

Bag of Words

En la representación *Bag of Words*, para cada palabra p se define de la variable V(p) como "el texto contiene la palabra p", es decir, para cada texto i,

$$V_i(p) = \begin{cases} 1 \; si \; el \; texto \; i \; contiene \; la \; palabra \; p \\ 0 \; en \; caso \; contrario \end{cases}$$

En la representación $tf - idf$ (*term frequency – inverse document frequency*) se mide el número de apariciones de cada palabra en cada texto sobre el número de apariciones de cada palabra en el total de textos.

Estas técnicas vectorizan los textos y los hacen susceptibles de servir como entrada a algoritmos de *machine learning*, pero tienen problemas relativos a la inmensa cantidad de ruido, y no tienen en cuenta posiciones relativas de las palabras.

Textos

Análisis léxico y normalización:

- Mayúsculas y minúsculas.
- Signos de puntuación y siglas.
- Números.

Eliminación de stopwords: las stopwords (artículos, preposiciones, conjunciones, …) son palabras que no tienen mucho significado y cuya presencia es tan frecuente en los textos que será raro que discriminen. Suelen eliminarse.

Para determinadas aplicaciones (análisis exploratorio de *corpus*), las palabras de frecuencia muy bajan suelen eliminarse también, pero en problemas de clasificación no suelen ser útiles.

Textos

Stemming: normalizar palabras de manera que todas las de la misma familia se identifiquen:

- Mediante búsqueda en diccionario.
- Eliminando prefijos y sufijos.

Uso de n-gramas: un n-grama es una tupla de n-palabras consecutivas. Si se alimentan las representaciones *Bag of Words* o tf-idf con n-gramas en lugar de palabras se consigue incorporar información de posiciones relativas de palabras en los textos.

Textos

Otros variables que se generan con cierta frecuencia y que no pasan por vectorizar los textos son:

- Propiedades estructurales del texto: longitud media de palabras, frases, …
- Densidad de signos de puntuación.
- *Part-of-specch tagging*: proporción de sustantivos, verbos , …

Referencias

Referencias

1. Cohen, I. (2019) Optimizing Feature generation. Link
2. Geigey, A. (2016) Machine Learning is Fun! Part 3: Deep Learning and Convolutional Neural Networks. Link
3. Hu, Y.-J.; Kiebller, D. (1996). Generation of attribues for learning algorithms. In Proc. 13th International Conference on Machine Learning. Morgan Kaufmann.
4. Markovitch, S.; Rosentein, D. (2002). Feature Generation Using General Constructor Functions. Machine Learning. 49, pp. 59-98.
5. Micci-Barreca, D. (2001) A preprocessing scheme for high-cardinality categorical attributes in classification and predictions problems.

Afi Escuela