

---

# Gestión de características

---

PID\_00284575

Jordi Gironés Roig

---

Tiempo mínimo de dedicación recomendado: 1 hora

---



**Jordi Gironés Roig**

Licenciado en Matemáticas por la Universidad Autónoma de Barcelona y diplomado en Empresariales por la Universitat Oberta de Catalunya. Ha desarrollado la mayor parte de su carrera profesional en torno de la solución SAP, en sus vertientes operativas con S4-HANA y estratégica con SAP-BI. Actualmente trabaja en la industria químico-farmacéutica como responsable de aplicaciones corporativas para Esteve Pharmaceuticals y colabora con la UOC en asignaturas relacionadas con la analítica de datos.

El encargo y la creación de este recurso de aprendizaje UOC han sido coordinados por el profesor: Julià Minguillón Alfonso

Primera edición: septiembre 2021  
© de esta edición, Fundació Universitat Oberta de Catalunya (FUOC)  
Av. Tibidabo, 39-43, 08035 Barcelona  
Autoría: Jordi Gironés Roig  
Producción: FUOC  
Todos los derechos reservados

*Ninguna parte de esta publicación, incluido el diseño general y la cubierta, puede ser copiada, reproducida, almacenada o transmitida de ninguna forma, ni por ningún medio, sea este eléctrico, mecánico, óptico, grabación, fotocopia, o cualquier otro, sin la previa autorización escrita del titular de los derechos.*

# Índice general

<b>Introducción</b>	5
<b>1. Visualización de características</b>	7
<b>2. Selección de características</b>	10
2.1. <i>Undersampling</i>	11
2.1.1. SMOTE	11
2.1.2. ADASYN	11
2.2. Porcentaje de valores nulos	11
2.3. Varianza estadística	12
2.4. Correlación entre variables explicativas	12
2.5. PCA y SVD	13
2.6. Medidas de importancia	13
2.7. Procesos secuenciales	14
2.8. <i>Stability selection</i>	15
<b>3. Ingeniería de características</b>	16
3.1. <i>Feature bucketing</i>	16
3.2. <i>Feature hashing</i>	17
3.3. <i>Feature crossing</i>	18
3.4. Ingeniería sobre datos categoriales	18
3.5. Tipos de codificación	19
3.6. Redondeo	19
3.7. Transformación logarítmica	19



## Introducción

La mitad de la capacidad del cerebro humano es responsable de procesar menos del 5 % del espectro visual al que tenemos acceso. Ni siquiera nuestro sistema nervioso óptico tiene la habilidad de capturar todos los datos que tiene a su alcance.

Y, sin embargo, nuestra capacidad visual es reconocidamente útil, eficaz y eficiente. Ante este escenario, parece lógico plantearse la siguiente pregunta: ¿cómo se consigue con unos recursos tan limitados tales niveles de excelencia?

*Simplificación* es la palabra clave. **Prescindir de lo que no sirve**, focalizarse en lo relevante, identificar patrones y aprender. Estos son los pilares en los que se basa nuestro cerebro para operar ante tanto caos y este va a ser nuestro *leitmotiv* para entender bien la gestión de características en la minería de datos.

Uno de nuestros genios de la pintura universal, Pablo Picasso, describía de este modo el proceso creativo.

«El arte es la eliminación de lo innecesario».

Esta frase encierra la mejor descripción que se puede hacer de un proceso como la **selección de características**. Tanto es así que trabajar con atributos no adecuados puede repercutir negativamente en la construcción de modelos analíticos de las siguientes maneras:

- Ralentizando el proceso de aprendizaje.
- Dificultando la interpretabilidad del modelo generado.
- Disminuyendo drásticamente la capacidad de generalización del modelo.

Podemos tomar una nueva frase prestada del maestro pintor para ilustrar otro aspecto importante de la gestión de características:

«Algunos pintores transforman el sol en una mancha amarilla, otros transforman una mancha amarilla en el sol».

Transformar un atributo en bruto en un atributo capaz de mejorar la capacidad predictiva de un modelo es lo que conocemos como *ingeniería de características* y va a convertir nuestro juego de datos inicial en un juego de datos que definitivamente describe mejor el problema analítico que tratamos de estudiar.

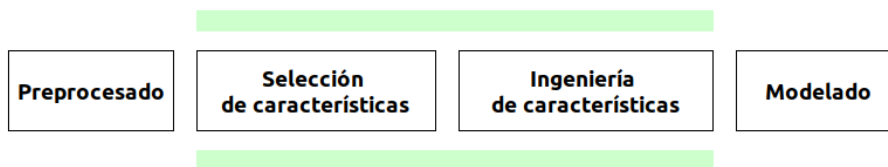
Cuando un modelo pasa de consumir alimentos crudos a alimentos elaborados y mejor adaptados a sus necesidades nutricionales, rinde mejor en todos sus aspectos.

De este modo, la ingeniería de características va a aportar los siguientes beneficios a nuestros proyectos analíticos:

- El incremento de la capacidad de aprendizaje.
- La simplificación de la interpretabilidad del modelo generado.

En la figura 1 ponemos en contexto dentro del proceso analítico las fases de selección e ingeniería de características.

Figura 1. Gestión de características



A modo de resumen, una buena gestión de características, basada en la **selección de atributos** realmente relevantes para nuestro problema analítico y en una **ingeniería de atributos** orientada a optimizar su posterior uso en modelos, va a ser lo que en muchas ocasiones marcará la diferencia. En este material docente introducimos brevemente diferentes métodos para la gestión de características, sin entrar en detalles técnicos.

## 1. Visualización de características

Uno de los consejos más repetidos entre los expertos en inteligencia artificial es este: visualiza tus datos con cierta frecuencia.

La visualización de datos tradicionalmente se ubica en la fase de preparación del juego de datos, pero es también recomendable usar las capacidades gráficas en fases de transformación de características y modelado.

Pero ¿qué buscamos en las visualizaciones? ¿Y qué beneficios pueden aportarnos? Básicamente son dos:

- 1) Nos ayudarán a **corroborar nuestras suposiciones** en cada paso que demos.
- 2) Nos permitirán **valorar los efectos de los cambios** que vayamos introduciendo en el juego de datos.

Los tipos de gráficos más utilizados son los siguientes:

- Gráficos de puntos, en los que podemos visualizar la relación entre dos variables.
- Gráficos de barras, que se usan para visualizar datos categoriales en barras verticales proporcionales a los valores que representan.
- Histogramas de frecuencias, que nos permiten analizar cómo de común es una clase en una característica.

Existen muchos más tipos de gráficos, como los de cajas, los de tarta o los de densidad, y deberemos valorar cuál se adapta mejor a las necesidades de visualización de datos que vayamos detectando.

Gestionar el formato del gráfico también puede contribuir a facilitar la interpretación y la comprensión de su contenido. Algunos aspectos que deben considerarse son:

- Asignar colores y darles un significado relativo a los datos.
- Usar otras figuras geométricas, además de puntos, asignando también significado a cada figura.

- Elaborar las leyendas, explicando el significado de cada símbolo.
- Acompañar el gráfico con la tabla de valores que lo genera, que en ocasiones puede ayudar.

Para mostrar cómo las visualizaciones nos permitirán ir más allá de las medidas estadísticas clásicas, tomaremos el juego de datos conocido como *cuarteto de Anscombe*, disponible en la figura 2.

Figura 2. Juego de datos del cuarteto de Anscombe

I		II		III		IV	
x	y	x	y	x	y	x	y
10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71
9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47
14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04
6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25
4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.50
12.0	10.84	12.0	9.13	12.0	8.15	8.0	5.56
7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91
5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89

Los cuatro juegos de datos comparten las siguientes medidas estadísticas:

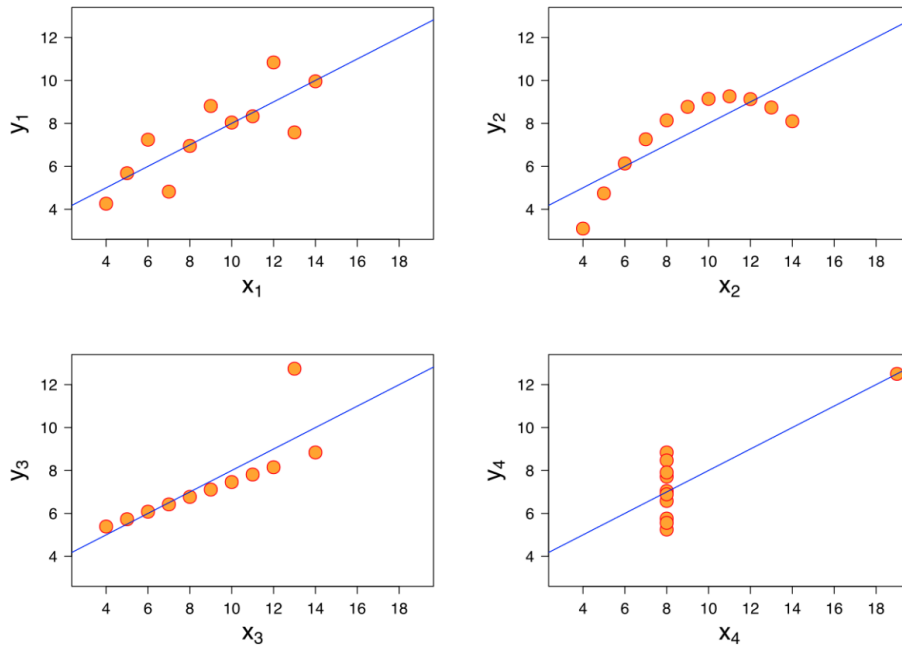
- Media ( $x$ ) = 9
- Varianza ( $x$ ) = 11
- Media ( $y$ ) = 7.50
- Varianza ( $y$ ) = 4.125
- Correlación entre  $x$  e  $y$  = 0.816
- Recta de regresión  $y = 3 + 0.5 x$

En la figura 3 apreciamos cómo, a pesar de que los cuatro juegos de datos comparten los mismos estadísticos, tenemos visualizaciones muy distintas:

- Caso I: apreciamos una relación lineal entre las variables  $x$  e  $y$ .
- Caso II: con relación parabólica.
- Caso III: con relación lineal, pero con un valor extremo significativo.
- Caso IV: en el que un solo valor extremo es capaz de producir un coeficiente de correlación alto a pesar de que el resto de puntos no presentan correlación alguna.



Figura 3. Visualizaciones sobre el cuarteto de Anscombe



En definitiva, las visualizaciones nos darán ese paso atrás que proporciona la perspectiva necesaria para ver lo que *a priori* no es obvio, pero está ahí.

J. A. Baker, en su libro *The Peregrin*, escribía:

«The hardest thing of all to see is what is really there».

En ocasiones, lo más difícil de ver es precisamente lo que está ahí delante nuestro y es por este motivo que a pesar de ser una herramienta irrenunciable, las visualizaciones no son suficientes para captar el conocimiento subyacente en un juego de datos.

## 2. Selección de características

Empecemos por clarificar qué entendemos por característica. Tal y como se aprecia en la figura 4, en un juego de datos encontraremos filas y columnas, donde las primeras representan observaciones o eventos descritos por el juego de datos y las segundas representan las propiedades de estas observaciones.

Figura 4. Definición de *característica*

	Artículo	Unidades	Ventas	
Observaciones →	Alfombra	500	450	← Características
Observaciones →	Silla	800	950	
Observaciones →	Estantería	200	300	

Trasladado al mundo del lenguaje, podemos pensar en las filas como sustantivos y en las columnas como adjetivos que definen y califican los sustantivos. A lo largo de este material didáctico nos referiremos a las columnas también como *atributos*, *características* y *variables*.

La idea generalizada de que cuantos más datos mejor es cuando menos errónea y debería ser sustituida por esta: cuantos más datos relevantes para nuestro objetivo, mejor.

Bajo *feature selection*, en su acepción inglesa, englobamos una serie de técnicas orientadas a seleccionar solo aquellas características en un juego de datos que nos van a aportar y van a hacer que el juego de datos realmente describa el problema analítico que tratamos de resolver.

Veremos a continuación cómo algunas de estas técnicas son extremadamente simples y otras más elaboradas, pero, no debemos olvidar que todas ellas persiguen un objetivo común:

Conseguir que el juego de datos represente lo mejor posible el problema analítico que tratamos de resolver.

## 2.1. Undersampling

La circunstancia del *oversampling* y del *undersampling* impacta especialmente en los problemas de clasificación y sucede cuando una de las clases objetivo está sobrerrepresentada o infrarrepresentada en el juego de datos.

Esto puede provocar que el algoritmo de clasificación solo aprenda sobre la clase de la que tiene más información y en consecuencia genere un modelo que simplemente ignore la clase con menos representación en el juego de datos.

Para evitar este tipo de situaciones disponemos de las técnicas SMOTE y ADASYN.

### 2.1.1. SMOTE

Acrónimo del término inglés *synthetic minority oversampling technique*. Es una técnica estadística que permite generar en un juego de datos observaciones nuevas que ayuden a incrementar la representatividad de aquellas clases minoritarias en el juego de datos.

En lugar de generar observaciones nuevas simplemente como copia de las ya existentes, SMOTE lo que hace es tomar los vecinos más próximos (K-NN) de las observaciones con clase minoritaria y generar una combinación de sus características.

### 2.1.2. ADASYN

Acrónimo del término inglés *adaptive synthetic sampling approach*. Se trata de una evolución del anterior algoritmo SMOTE, de modo que en lugar de tratar de replicar las observaciones con clases minoritarias, solo trata de hacerlo sobre aquellas clases minoritarias con dificultad de aprendizaje alto.

## 2.2. Porcentaje de valores nulos

Los valores nulos, o *missing values* en inglés, pueden ocasionar problemas en el funcionamiento de ciertos algoritmos, por lo que es una situación que debe gestionarse desde diferentes aproximaciones. Las más habituales son:

- Sustituirlos por la media estadística.
- Sustituirlos por la mediana estadística.
- Sustituirlos por valores aleatorios.

- Sustituirlos por cero.
- Eliminar las filas afectadas.

Desde el punto de vista de la gestión de características, deberíamos plantearnos una posibilidad más: eliminar las características que superen un porcentaje preestablecido de valores nulos. Más concretamente, una posible estrategia sería la siguiente:

- Por encima del 95 % de valores nulos, se descarta la característica.
- Entre el 50 % y el 95 % de valores nulos, estos no se rellenan.
- Por debajo del 50 % de valores nulos, estos se rellenan siguiendo alguna de las opciones ya mencionadas.

Esta estrategia, basada en el grado de afectación de los valores nulos sobre cada característica, nos va a permitir una gestión más pormenorizada del problema.

### 2.3. Varianza estadística

En un juego de datos, la cantidad de información contenida en una característica  $X$  con valores  $x_1, \dots, x_n$ , puede medirse mediante la varianza estadística:

$$VAR(X) = \sigma^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2 \quad (1)$$

Donde  $x_i$  son los valores de la característica  $X$  y  $\mu$  es su media estadística.

Deberían descartarse las características con una varianza próxima a cero, puesto que no contienen información al carecer de variabilidad y, en consecuencia, no describen ninguna propiedad de las entradas del juego de datos.

Mantener características con varianza muy baja solo contribuirá a complicar el modelo y a confundirlo.

### 2.4. Correlación entre variables explicativas

Si dos características explicativas están muy correlacionadas entre ellas, significa que ambas aportan información redundante al juego de datos y, en consecuencia, podemos prescindir de una de ellas.

¿Con cuál de las dos características nos quedaremos? Con aquella que tenga más correlación con la variable objetivo, *target* en inglés.

La situación ideal sería que todas las variables explicativas fueran independientes entre ellas, es decir, que hubiera correlación cero entre ellas y solo la tuvieran con la variable objetivo.

Sin embargo, hay que tener mucho cuidado cuando trabajamos con correlaciones porque estas miden la relación lineal entre características y no debemos olvidar que **esta relación podría existir y no ser lineal**.

Una consecuencia de la frase anterior sería que, si una variable explicativa no está correlacionada con la variable objetivo o *target*, no deberíamos llegar a la conclusión de que podemos prescindir de la variable explicativa, ya que aun así, esta podría contener información relevante para predecir la variable objetivo.

## 2.5. PCA y SVD

Tanto el análisis de componentes principales, *principal component analysis* (PCA) en inglés, como la descomposición de valores singulares, *singular value decomposition* (SVD) en inglés, son técnicas que nos permiten trabajar con nuevas características llamadas componentes, que ciertamente son independientes entre sí.

En realidad, estas dos técnicas nos permiten representar el juego de datos en un nuevo sistema de coordenadas que llamamos componentes principales. Este sistema está mejor adaptado a la distribución del juego de datos, de modo que recoge mejor su variabilidad.

Lo que suele hacerse es **incorporar al juego de datos original**, *raw data set* en inglés, los **componentes principales** como características nuevas, seleccionando aquellos componentes que acumuladamente recogen una variabilidad superior al 90%. Esta acción puede actuar de catalizador en un proceso de clasificación.

En contrapartida, tenemos el inconveniente de la interpretabilidad, es decir, dar significado a estas nuevas dimensiones en el juego de datos.

## 2.6. Medidas de importancia

Los árboles de decisión son algoritmos de clasificación que construyen caminos de decisión a partir de las variables del juego de datos. Estas van participando por orden de importancia, de modo que primero intervienen las características más relevantes para el proceso de clasificación.

Esto significa que son capaces de ordenar las características según su importancia o relevancia en el juego de datos. Pero ¿cómo podemos medir la importancia de un atributo en un juego de datos?

Una de las formas más habituales de medir esta importancia es la **ganancia de la información** o pequeñas variaciones sobre este concepto.

A su vez, la ganancia de información descansa sobre otro concepto: la **entropía**.

$$E(X) = \sum_{i=1}^n -p_i \log_2 p_i \quad (2)$$

En este concepto,  $p_i$  es la probabilidad de que se dé el valor  $x_i$  en el atributo  $X$ . De este modo, podemos decir que la entropía mide el desorden o la cantidad de información contenida en un juego de datos.

A partir de aquí, si somos capaces de medir la variación que se produce en la entropía de un juego de datos cada vez que fijamos un valor de un atributo, tendremos un criterio para decidir qué atributo maximiza la ganancia de información, *information gain* en inglés.

$$IG(T) = H(T) - H(T|x) \quad (3)$$

Aquí,  $T$  es un juego de datos con varios atributos y  $H(T|x)$  es la entropía del juego de datos  $T$  cuando fijamos un valor del atributo  $x$ .

Tanto utilizar directamente la ganancia de información como utilizar los diferentes algoritmos de árboles de decisión son buenas opciones para ordenar las características de un juego de datos por su importancia y seleccionar solo aquellas que son más relevantes.

## 2.7. Procesos secuenciales

Podemos usar procesos secuenciales de adición o sustracción de características a un juego de datos a juzgar por su relevancia con el objetivo de maximizar algún criterio preestablecido.

Este criterio podría ser el nivel de precisión de un algoritmo predictivo. De este modo, aplicaríamos procesos secuenciales de selección de características sobre el juego de datos, hasta que el modelo que consume las distintas versiones de juegos de datos que vamos generando alcance un nivel de rendimiento aceptable.

Básicamente, podemos distinguir tres tipos de procesos secuenciales de selección de características:

### 1) *Forward selection*

- Identificar las características más relevantes.
- Añadir la siguiente característica al modelo.
- Iterar el paso anterior hasta alcanzar un nivel de precisión satisfactorio en el modelo.

### 2) *Backward elimination*

- Ejecutar el modelo considerando todas las características.
- Eliminar la característica menos relevante y volver a aplicar el modelo.
- Iterar el paso anterior hasta alcanzar un nivel de precisión satisfactorio en el modelo.

3) *Stepwise selection*. Este proceso es igual al de *forward selection*, excepto por el hecho de que nos plantearemos a lo largo de las iteraciones si merece la pena eliminar alguna característica que haya pasado a ser irrelevante.

## 2.8. *Stability selection*

La selección de características por estabilidad consiste en aplicar un algoritmo de selección de características de forma repetida sobre diferentes subconjuntos o muestras del juego de datos original. La muestra se toma tanto a nivel de observaciones como a nivel de características.

Después de repetir el proceso anterior cierto número de veces, agregamos los resultados de modo que podamos verificar cuántas veces una característica se ha considerado importante.

De este modo podremos distinguir tres tipos de características:

- Características muy relevantes, que identificaremos por tener puntuaciones cercanas al 100 %.
- Características más débiles, pero aun así relevantes, que identificaremos por tener puntuaciones por encima de 0.
- Características irrelevantes, que identificaremos por tener puntuaciones cercanas a 0.

En el artículo «Stability selection, RFE and everything side by side» puede encontrarse un buen ejemplo.

### Referencia bibliográfica

«Stability selection, RFE and everything side by side» (2014). *Diving into data*. <<https://bit.ly/2RLMHZ3>>

### 3. Ingeniería de características

El éxito de muchos algoritmos de minería de datos radica en cómo presentamos los datos al modelo.

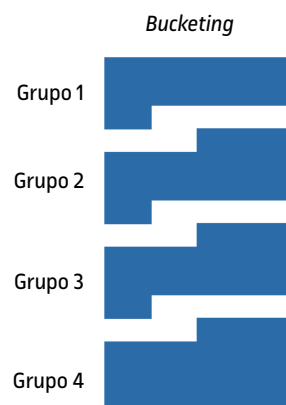
Un problema analítico bien planteado dependerá principalmente de tres factores: disponer de los datos adecuados, presentar las características bien estructuradas y aplicar el modelo correcto.

Incluso si no seleccionamos el mejor de los modelos posibles, si los datos tienen potencial y las características se presentan adecuadamente, podremos obtener buenos resultados. De este modo, la ingeniería de características merece la pena tomársela como un **aspecto estratégico** en todo estudio analítico de datos.

#### 3.1. *Feature bucketing*

La agrupación de valores numéricos en rangos acorde al problema analítico que tratamos de resolver ayuda a mejorar el rendimiento en algoritmos de clasificación como las redes neuronales. En realidad, también puede verse como una transformación de variables numéricas en categoriales.

Figura 5. *Feature bucketing*



Cuando trabajamos con variables continuas como, por ejemplo, ventas de un producto, cantidad de azúcar por litro, ingresos de un cliente... los algoritmos



de clasificación toman cada valor como una clase individual, contribuyendo así al sobreentrenamiento y confundiendo sobre lo que realmente es importante tener en cuenta.

Podemos considerar diferentes criterios de categorización:

- Tomar la distribución por cuartiles.
- Tomar distribuciones equidistantes.

Agrupar por rangos nuestros valores continuos ayudará a la capacidad predictiva y descriptiva de los modelos analíticos.

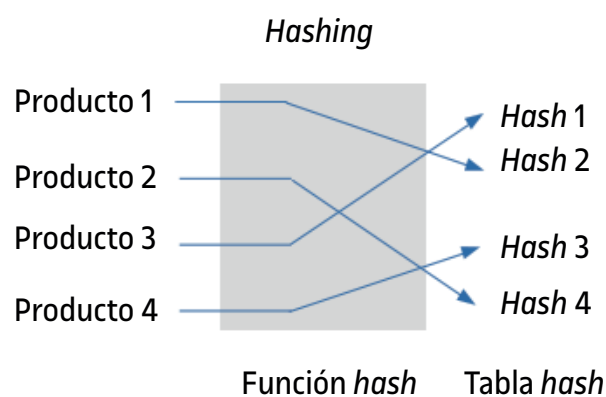
### 3.2. *Feature hashing*

Las funciones *hash* se han convertido en una de las técnicas más usadas dentro del mundo de la inteligencia artificial por las ventajas que ofrecen.

Existen muchos tipos de funciones *hash*, pero todas ellas siguen el mismo principio: mapean un grupo de datos de tamaño arbitrario contra un grupo de datos de tamaño fijo.

Tomando como ejemplo la figura 6, si introducimos un hipotético producto 5, la función *hash* asignará igualmente uno de los valores del diccionario {Hash 1, Hash 2, Hash 3, Hash 4}.

Figura 6. *Feature hashing*



De este modo, como la característica resultante siempre tendrá el mismo diccionario de datos, no será necesario entrenar de nuevo los modelos en los que haya participado en caso que la característica inicial sí que tuviera valores nuevos.

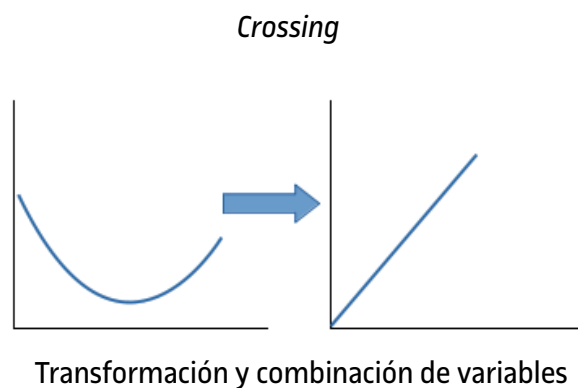
Precisamente gracias a esta habilidad, las técnicas de *feature hashing* están especialmente indicadas para variables categoriales con muchos posibles valores, porque de algún modo son capaces de comprimirlos.

Como inconveniente, merece la pena comentar que las funciones *hash* tienen el riesgo de generar colisiones, es decir, que valores iniciales totalmente dispares acaben siendo asignados mediante el mapeo al mismo valor del diccionario, relacionándolos innecesariamente.

### 3.3. *Feature crossing*

El cruce de características, o *feature crossing* en inglés, es una técnica sencilla, pero que nos va a permitir transformar problemas de clasificación no lineales en problemas linealmente clasificables, tal y como se aprecia en la figura 7.

Figura 7. *Feature crossing*



A modo de ejemplo, en un juego de datos con dos variables numéricas  $\{x_1, x_2\}$  podremos considerar las siguientes tres nuevas variables  $\{x_1^2, x_2^2, x_1 \cdot x_2\}$  con el objetivo de que, tras estas transformaciones y considerando las nuevas variables, nuestro juego de datos pase a ser linealmente separable.

### 3.4. Ingeniería sobre datos categoriales

Puesto que algunos modelos trabajan mejor con atributos numéricos que con atributos categoriales, puede ser de utilidad aplicar un proceso de transformación a valores numéricos.

Este proceso simplemente consiste en asignar un número decimal a cada clase del atributo original. Esta asignación tiene el inconveniente de que no tiene sentido medir distancias o similitudes sobre la variable decimal resultante, pero podemos mitigar este problema considerando también la codificación binaria y la codificación *one-hot*.

### 3.5. Tipos de codificación

Tomando como ejemplo la figura 8, apreciamos cómo el diccionario de medios de transporte urbano formado por las clases {moto, bicicleta, patinete, monociclo, segway} podemos codificarlo de tres modos:

- **Decimal:** asignando un número decimal a cada clase.
- **Binario:** asignando un número expresado de forma binaria.
- **One-hot:** asignando un vector de longitud al número de clases, con 0 en todas las posiciones excepto en la posición de la propia clase, que tomará valor 1.

Figura 8. Tipos de codificación

#### *Encoding*

Etiqueta	Decimal	Binario	<i>One-hot</i>
moto	0	000	00001
bicicleta	1	001	00010
patinete	2	010	00100
monociclo	3	011	01000
segway	4	100	10000

Las codificaciones binaria y *one-hot* son habituales en algoritmos como las redes neuronales.

### 3.6. Redondeo

En ocasiones, características numéricas como porcentajes o proporciones suelen expresarse con precisión decimal. Debemos preguntarnos, en estos casos, si realmente necesitamos para nuestro modelo este nivel de precisión y, en consecuencia, plantearnos la opción de redondearlas.

### 3.7. Transformación logarítmica

La transformación logarítmica nos ayudará especialmente en variables numéricas con valores que pueden tomar magnitudes muy dispares.

Por ejemplo, si seleccionamos una muestra de viviendas de una gran ciudad y consideramos como característica su valor de mercado, veremos mucha dis-

crepancia entre barrios periféricos y barrios céntricos dificultando su representación gráfica. Además, en ocasiones observaremos también distribuciones sesgadas.

$$x \rightarrow \log_2 x \quad (4)$$

La transformación logarítmica al considerar las potencias de las magnitudes contribuye a conseguir una distribución más cerca de la normal y a facilitar su representación gráfica al conseguir magnitudes más uniformes.

Para **reducir el sesgo a la derecha** en una distribución de valores de una variable, usaremos la **transformación logarítmica**.

Para **reducir el sesgo a la izquierda** en una distribución, usaremos la **transformación potencial**, que consistirá en tomar los cuadrados, cubos e incluso potencias superiores. La **transformación Box-Cox** también pertenece a esta familia y es cada vez más usada en procesos de *machine learning*.