Preprocesado de datos

PID_00284571

Raúl Montoliu Colás

Tiempo mínimo de dedicación recomendado: 2 horas

Universitat Oberta de Catalunya

Raúl Montoliu Colás

Ingeniero en Informática por la Universidad Jaume I (UJI) de Castellón. Doctor en Métodos Avanzados Informáticos por la misma universidad. Actualmente trabaja como docente en el departamento de Ingeniería y Ciencia de los Computadores de la UJI y como investigador en el grupo de investigación Machine Learning for Smart Environments del Instituto de Nuevas Tecnologías de la Imagen (INIT). Desde el 2017 colabora como docente en la Universitat Oberta de Catalunya (UOC).

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

Primera edición: septiembre 2021

Av. Tibidabo, 39-43, 08035 Barcelona
Autoría: Raúl Montoliu Colás
Producción: FUOC

Índice

Introducción 5

1 Transformaciones de valores 7
1.1 Normalización 7
1.1.1 Normalización por el máximo 8
1.1.2 Normalización por la diferencia 8
1.1.3 Escalado decimal 9
1.1.4 Normalización basada en la desviación estándar 9
1.2 Discretización 10
1.2.1 Método de partición en intervalos de la misma amplitud 11
1.2.2 Obtención de intervalos de discretización de igual frecuencia 12
1.2.3 Método de partición basado en el algoritmo *k-means* . 13

2 Reducción de la dimensionalidad 15
2.1 Reducir atributos 15
2.2 Eliminar *outliers* mediante técnicas de edición 16
2.3 Reducir muestras mediante técnicas de condensación 18

3 Valores ausentes 19

Introducción

En prácticamente todos los procesos de minería de datos suele ser necesario que los datos, una vez seleccionados, tengan que ser modificados y preparados. De este modo, será posible crear modelos de minería de datos que funcionen de forma óptima.

En este módulo, se comentarán las técnicas de preparación de datos más frecuentes:

- La transformación de datos: en concreto, la normalización y la discretización.
- El tratamiento de los valores no observados.
- La reducción de la dimensionalidad de los datos.

Las técnicas que se presentan en este módulo son bastante independientes, hasta cierto punto, del tipo de modelo que se va a usar.

1. Transformaciones de valores

Por transformación de valores, entendemos modificaciones dentro del tipo de valores que pueden adoptar todos o algunos de los atributos. Las operaciones más habituales son la normalización y la discretización de datos.

1.1 Normalización

La normalización consiste en situar los datos sobre una escala de valores equivalentes que permita la comparación de atributos que toman valores en dominios o rangos diferentes.

La normalización es útil, o necesaria, para varios métodos de construcción de modelos, como, por ejemplo, las redes neuronales o algunos métodos basados en distancias, como el de los vecinos más próximos. En efecto, si no hay normalización previa, los mencionados métodos tienden a quedar sesgados por la influencia de los atributos con valores más altos, hecho que distorsiona el resultado.

Por ejemplo, imaginemos que tenemos los datos de un conjunto de comerciales de una empresa ficticia mostrados en la tabla 1. Por un lado, la diferencia de edad entre el comercial más joven (comercial 1) y el menos (comercial 4) es de $|34 - 64| = 30$ años. Por otro lado, la diferencia de salario anual entre los citados comerciales es de $|34300 - 34000| = 300$ euros. Mientras que la diferencia de edad entre los dos comerciales es bastante elevada, la diferencia en el salario anual es razonablemente baja. Sin embargo, el valor de la diferencia en el caso del salario es 100 veces superior al de la edad (300 frente a 30). Como puede comprobarse, la diferencia de salario puede llegar a influenciar más que la diferencia en edad por una cuestión de las magnitudes que se usan para representar dichos conceptos.

Tabla 1. Ejemplo hipotético de los datos de un conjunto de comerciales de una empresa

Comercial	Edad	Salario	Ventas	Kilómetros
1	34	34300	120000	3400
2	54	24000	80000	1400
3	39	45000	20000	1300
4	64	34000	130000	5400
5	48	28000	220000	3400

La normalización evitará estos problemas porque permite comparar todos los atributos en igualdad de condiciones.

1.1.1 Normalización por el máximo

La normalización por el máximo consiste en encontrar el valor máximo x_{max} del atributo que se debe normalizar X y dividir todos los valores del atributo por x_{max}. Formalmente se define como:

$$x_{max} = \max_{i=1,\ldots,N} x_i \tag{1}$$

$$x'_i = \frac{x_i}{x_{max}}, \forall i \in [1,N] \tag{2}$$

donde x_i es cada uno de los valores del atributo $X = \{x_1, x_2, \ldots, x_N\}$, N es el total de muestras y x'_i es cada uno de los nuevos valores del atributo tras el proceso de normalización, es decir $X' = \{x'_1, x'_2, \ldots, x'_N\}$.

Con este tipo de normalización, nos aseguramos que los valores del atributo estarán en el rango $[\frac{\min_{i=1,\ldots,N} x_i}{x_{max}}, 1]$.

Por ejemplo, dado el atributo *Edad* de la tabla 1 que tiene los valores: 34, 54, 39, 64 y 48, el valor máximo es 64. Por lo tanto, los valores normalizados serán: 34/64, 54/64, 39/64, 64/64 y 48/64, de lo que resulta (aproximando con tres decimales): 0.531, 0.844, 0.609, 1.000 y 0.750.

De forma similar, podemos normalizar el atributo *Salario*, que tiene los valores: 34300, 24000, 45000, 34000, 28000. El máximo será 45000 y los valores normalizados: 34300/45000, 24000/45000, 45000/45000, 34000/45000, 28000/45000, lo que resulta (aproximando con tres decimales): 0.762, 0.533, 1.000, 0.756 y 0.622.

Tras el proceso de normalización, la diferencia entre el primer y el cuarto comercial en el atributo *Edad* es de $|0.531 - 1.00| = 0.469$, mientras que en el atributo *Salario* es de $|0.762 - 0.755| = 0.007$. Es decir, ahora el primer y cuarto comercial están más cerca en el atributo *Salario* que en el de *Edad*.

1.1.2 Normalización por la diferencia

La normalización por la diferencia trata de compensar el efecto de la distancia del valor que tratamos respecto al máximo de los valores observados. La normalización por la diferencia consiste en realizar la transformación siguiente:

$$x_{max} = \max_{i=1,\ldots,N} x_i \tag{3}$$

$$x_{min} = \min_{i=1,\ldots,N} x_i \tag{4}$$

$$x'_i = \frac{x_i - x_{min}}{x_{max} - x_{min}}, \forall i \in [1,N] \tag{5}$$

Con este tipo de normalización nos aseguramos que los valores del atributo estarán en el rango [0,1].

Por ejemplo, vamos a normalizar los atributos *Edad* y *Salario* mostrados en la tabla 1. El mínimo y máximo del atributo *Edad* es: 34 y 64, respectivamente. Por lo tanto, los nuevos valores tras el proceso de normalización serán: $\frac{34-34}{64-34} = 0.00$, $\frac{54-34}{64-34} = 0.67$, $\frac{39-34}{64-34} = 0.17$, $\frac{64-34}{64-34} = 1.00$ y $\frac{48-34}{64-34} = 0.47$. En el caso del atributo *Salario*, los nuevos valores, teniendo en cuenta que el mínimo y el máximo son 24000 y 45000, respectivamente, serán: $\frac{34300-24000}{45000-24000} = 0.49$, $\frac{24000-24000}{45000-24000} = 0.00$, $\frac{45000-24000}{45000-24000} = 1.00$, $\frac{34000-24000}{45000-24000} = 0.48$ y $\frac{28000-24000}{45000-24000} = 0.19$.

1.1.3 Escalado decimal

El escalado decimal permite reducir en un cierto número de potencias de diez el valor de un atributo. Esta transformación resulta especialmente útil al tratar con valores elevados, como por ejemplo los atributos *Salario* y *Ventas* mostrados en la tabla 1. La transformación se realizará tal como se muestra a continuación:

$$x'_i = \frac{x_i}{10^j}, \forall i \in [1,N] \tag{6}$$

donde *j* tiene que ser tal que mantenga el máximo valor que puede adoptar el atributo por debajo de uno.

Por ejemplo, para el atributo *Salario*, *j* sería igual a 5. De esta forma, los nuevos valores para este atributo, tras el proceso de normalización, serían: $\frac{34300}{10^5} = 0.343$, $\frac{24000}{10^5} = 0.240$, $\frac{45000}{10^5} = 0.450$, $\frac{34000}{10^5} = 0.340$ y $\frac{28000}{10^5} = 0.280$. Sin embargo, para el atributo *Ventas*, el valor correcto de *j* sería 6.

Al usar este tipo de normalización, el nuevo rango de valores estará en el siguiente intervalo: $[\frac{\min_{i=1,\ldots,N} x_i}{10^j}, 1]$.

1.1.4 Normalización basada en la desviación estándar

Los métodos anteriores no tienen en cuenta la distribución de los valores existentes. El método de normalización basada en la desviación estándar asegura

que se obtienen valores que tienen como propiedad que su media es cero y su desviación estándar vale uno.

Esta técnica consiste en hacer la transformación siguiente sobre los valores de los atributos:

$$x'_i = \frac{x_i - \mu}{\sigma}, \forall i \in [1,N] \quad (7)$$

donde μ es la media de todos los valores del atributo y σ es su desviación estándar. El nuevo rango de posible valores es: $[\frac{x_{min}-\mu}{\sigma}, \frac{x_{max}-\mu}{\sigma}]$.

Por ejemplo, para el atributo *Edad*, la media es μ = 47.8 y su desviación estándar σ = 11.92. Por lo tanto, los nuevos valores de este atributo tras realizar la transformación serán: $\frac{34-47.8}{11.92}$ = –1.16, $\frac{59-47.8}{11.92}$ = 0.52, $\frac{39-47.8}{11.92}$ = –0.74, $\frac{64-47.8}{11.92}$ = 1.36 y $\frac{48-47.8}{11.92}$ = 0.02. En caso del atributo *Salario*, la media es μ = 33060 y su desviación estándar σ = 7947.83. Por lo tanto, los nuevos valores de este atributo tras realizar la transformación serán: $\frac{34300-33060}{7947.83}$ = 0.16, $\frac{24000-33060}{7947.83}$ = –1.14, $\frac{45000-33060}{7947.83}$ = 1.50, $\frac{34000-33060}{7947.83}$ = 0.12 y $\frac{28000-33060}{7947.83}$ = –0.64.

1.2 Discretización

La discretización consiste básicamente en establecer un criterio por medio del cual se puedan dividir los valores de un atributo en dos o más conjuntos disjuntos.

Aunque no son estos los únicos motivos existentes para discretizar datos, citamos estos otros:

1) Coste computacional. Teniendo en cuenta que el conjunto de valores sobre el cual se trabajará después de discretizar implica una reducción de los valores por tratar, el número de comparaciones y cálculos que tendrá que realizar el correspondiente método de minería de datos será menor.

2) Velocidad en el proceso de aprendizaje. Se ha demostrado empíricamente que el tiempo necesario para llevar a cabo un proceso de entrenamiento de un método de minería de datos es más corto si se hace uso de datos discretizados.

3) Almacenamiento. En general, los valores discretos necesitan menos memoria para ser almacenados.

4) Tamaño del modelo resultante. Cuando se trabaja con datos continuos, los modelos clasificatorios que se obtienen son comparativamente mayores. Por ejemplo, los árboles de decisión que se obtienen acostumbran a tener un

factor de ramificación más alto cuando se trabaja con datos continuos que cuando se trabaja con datos discretos.

5) Comprensión. La comprensión de algunos modelos mejora en gran medida al describir el elemento utilizando menos términos.

Evidentemente, todo método de discretización está obligado a mantener o mejorar las características del modelo a cuya construcción sirve. Por ejemplo, si introducimos un proceso de clasificación y obtenemos tasas de error más altas o de predicción más bajas que sin discretizar, poco hemos ganado. Por lo tanto, lo que buscan la mayoría de los métodos de discretización es mantener la información asociada al atributo que se discretiza.

Un inconveniente que normalmente se cita al hablar de métodos de discretización es precisamente la pérdida de información sobre los valores continuos. Este tipo de efectos puede tener una influencia notable en la precisión de los métodos de aprendizaje.

A continuación se describen tres métodos para discretizar.

1.2.1 Método de partición en intervalos de la misma amplitud

El método de partición en intervalos de la misma amplitud es una técnica de discretización donde, dado un atributo X numérico, se siguen los siguientes pasos:

1) Calcular el valor mínimo x_{min} y el valor máximo x_{max}.

2) Fijar el número k de intervalos que se desea alcanzar.

3) Dividir el rango de valores $[x_{min},x_{max}]$ en k intervalos donde la distancia entre el máximo y el mínimo de cada intervalo sea la misma e igual a $(x_{max} - x_{min})/k$.

Partiendo de los datos mostrados en la tabla 1, podemos discretizar el atributo *Edad* en 3 intervalos (k = 3). Sabiendo que el mínimo del atributo es 34 y el máximo 64, el tamaño de cada intervalo será: (64 – 34)/3 = 10. Por lo tanto, el primer intervalo contendrá los valores [34,44), el segundo [44,54) y el tercero y último [54,64]. En la base de datos, podríamos modificar el valor de cada muestra para este atributo por 1, 2 y 3, indicando el número del intervalo. De esta forma, a la primera muestra (34) le asignaríamos el valor 1, a la segunda (54) el valor 3, a la tercera (39) el valor 1, a la cuarta (64) el valor 3 y a la última (48) el valor 2.

Para facilitar la compresión de los valores introducidos en esta base de datos, deberíamos acompañar a la base de datos con una descripción para explicar el significado de esos valores. Por ejemplo, podríamos indicar que 1 significa *joven*, 2 *mediana edad* y 3 *veterano*.

Este método tiene sus inconvenientes:

1) Da la misma importancia a todos los valores, independientemente de su frecuencia de aparición.

2) En caso de que queramos utilizar el método como punto de partida de clasificación, podemos encontrarnos con que se mezclen dentro de un mismo intervalo valores que corresponden a clases diferentes.

3) Con este método no hay manera de encontrar un valor de k que sea lo suficientemente bueno.

1.2.2 Obtención de intervalos de discretización de igual frecuencia

Uno de los problemas que hemos citado al comentar el método anterior, es que puede generar intervalos en los que las distintas clases o valores se distribuyan con frecuencias diferentes. Puede introducirse información sobre la frecuencia requerida de la manera siguiente:

- indicando el número de intervalos que hay que obtener,
- indicando la frecuencia que desea obtenerse para los intervalos.

El algoritmo 1 presenta los pasos que hay que seguir.

Algoritmo 1 Algoritmo para la obtención de intervalos de discretización de igual frecuencia. $X = \{x_1, x_2, \ldots, x_N\}$ es el conjunto de valores del atributo, k el número de intervalos buscado y $|\bullet|$ indica número de elementos.

1: Ordenar X
2: $f \leftarrow N/k$
3: $i \leftarrow 1$
4: $j \leftarrow 1$
5: Asignar el valor x_1 al intervalo I_1
6: **para** $j \leftarrow 2, N$ **hacer**
7: **si** $x_j \neq x_{j-1}$ **y** $|I_i| \geq N$ **entonces**
8: $i \leftarrow i + 1$
9: **fin si**
10: Asignar el valor x_j al intervalo I_i
11: **fin para**

Supongamos que el atributo X tiene los siguientes valores (ya ordenados): {22,22,27,27,28,28,31,31,31,40,50,50} y queremos discretizar en 3 intervalos ($k = 3$). En total hay 12 valores ($N = 12$), por lo tanto $f = N/k = 4$. El algoritmo 1 asignaría cada muestra a un intervalo tal como se describe a continuación:

- $x_1 = 22$: se asigna al primer intervalo.
- $x_2 = 22$: se asigna al primer intervalo, puesto que $x_2 == x_1$.
- $x_3 = 27$: se asigna al primer intervalo, puesto que, aunque $x_3 \neq x_2$, el número de elementos del primer intervalo es menor a f.
- $x_4 = 27$: se asigna al primer intervalo, puesto que $x_4 == x_3$.
- $x_5 = 28$: se asigna al segundo intervalo, puesto que $x_5 \neq x_4$ y el número de elementos del primer intervalo ya tiene el valor máximo permitido.
- $x_6 = 28$: se asigna al segundo intervalo, puesto que $x_6 == x_5$.
- $x_7 = 31$: se asigna al segundo intervalo, puesto que aunque $x_7 \neq x_6$, el número de elementos del segundo intervalo es menor a f.
- $x_8 = 31$: ae asigna al segundo intervalo, puesto que $x_8 == x_7$.
- $x_9 = 31$: se asigna al segundo intervalo, puesto que $x_9 == x_8$.
- $x_{10} = 40$: se asigna al tercer intervalo, puesto que $x_{10} \neq x_9$ y el número de elementos del segundo intervalo ha superado el valor máximo permitido.
- $x_{11} = 50$: se asigna al tercer intervalo, puesto que, aunque $x_{11} \neq x_{10}$, el número de elementos del tercer intervalo es menor a f.
- $x_{12} = 50$: se asigna al tercer intervalo, puesto que $x_{12} == x_{11}$.

1.2.3 Método de partición basado en el algoritmo *k-means*

Otro posible método es usar algunas de las ideas del algoritmo no supervisado de agregación *k-means*.

La mejor forma de explicar este método es mediante un ejemplo. Supongamos que queremos discretizar en 3 intervalos y que tenemos el siguiente conjunto de valores para el atributo X (ya ordenados): {20,21,28,29,30,31,31,39,40,51, 52,52}.

En el primer paso, asignamos 4 valores a cada uno de los tres intervalos, puesto que 12/3 = 4. Por lo tanto, los intervalos iniciales contendrán los valores:

- $I_1 = \{20,21,28,29\}$
- $I_2 = \{30,31,31,39\}$
- $I_3 = \{40,51,52,52\}$

El siguiente paso consiste en calcular el valor medio de cada intervalo, obteniendo 24.5 para I_1, 32.75 para I_2 y 48.75 para I_3. Ahora calculamos para cada muestra si su distancia al centroide del intervalo donde se encuentra es menor a la distancia al centroide del intervalo vecino. Si la distancia es menor o igual, la muestra permanece en el mismo intervalo. Si es mayor, la muestra se cambia de intervalo.

Por ejemplo, la muestra con valor 28 que se encuentra en el primer intervalo está a 28–24.5 = 3.5 del centroide de I_1 y a 32.75–28 = 4.75 del centroide de I_2. Por lo tanto, permanecerá en el intervalo I_1. Sin embargo, la muestra con valor 29 se encuentra más lejos del centroide del primer intervalo (29 – 24.5 = 4.5) que del segundo (32.75 – 29 = 3.75), por lo que se moverá al intervalo I_2. De forma similar, la muestra con valor 40 se encuentra más cerca del centroide del intervalo I_2 que del I_3, por lo que se moverá al I_2.

Tras esta primera iteración, los intervalos contendrán los valores:

- $I_1 = \{20,21,28\}$
- $I_2 = \{29,30,31,31,39,40\}$
- $I_3 = \{51,52,52\}$

Ahora los centroides de los tres intervalos serán: 23, 33.3 y 51.6, respectivamente. Todas las muestras están bien etiquetadas por lo que el proceso finaliza con los intervalos anteriores.

2. Reducción de la dimensionalidad

Una vez tenemos los datos en el formato adecuado para el tipo de modelo que se quiere obtener y el método para construirlo, todavía es posible aplicar una serie nueva de operaciones con el fin de cumplir dos objetivos: la reducción del número de atributos por considerar y la reducción del número de muestras que hay que tratar, asegurando, asimismo, que se mantendrá o se mejorará la calidad del modelo resultante.

El motivo para efectuar la reducción acostumbra a ser triple:

- El programa de construcción del modelo elegido no puede tratar la cantidad de datos de los que disponemos. En este caso, deberemos obtener un subconjunto que sí pueda ser tratado, pero intentando que la calidad del modelo no se vea comprometida.
- El programa puede tratarlos, pero el tiempo requerido para construir el modelo es inaceptablemente largo. De forma similar al caso anterior, el objetivo será obtener un subconjunto de los datos que requiera un tiempo menor de construcción del modelo, pero manteniendo un modelo con un nivel de calidad alto.
- La presencia de algunas muestras y/o atributos que, lejos de beneficiar la eficacia del modelo, la perjudica.

2.1 Reducir atributos

La reducción del número de atributos consiste en encontrar un subconjunto de los atributos originales que permita obtener modelos de la misma calidad, o incluso superior, que los que se obtendrían utilizando todos los atributos. A este problema también se le conoce como selección de características.

Un algoritmo trivial sería aquel en el que construimos el modelo con todos los atributos y, para cada atributo, se construye un nuevo modelo eliminando el atributo en cuestión. Si el modelo sin un atributo es mejor o comparable con el original, entonces podemos eliminar el atributo. Una vez eliminado el primer atributo, continuaríamos el proceso tratando de eliminar un nuevo atributo de entre los atributos sobrevivientes. El proceso continuaría hasta que no es posible encontrar un atributo que, al eliminarlo, mejore el modelo o los resultados sean comparables.

Existen algoritmos más eficaces y eficientes para tratar este problema.

2.2 Eliminar *outliers* mediante técnicas de edición

En términos generales, los modelos de minería de datos funcionan mejor cuantas más muestras tengan. Sin embargo, hay ocasiones donde algunas muestras, lejos de beneficiarlo, perjudican al modelo. Esto puede ser debido a muchos factores, y el más frecuente es un error en el proceso de captura de los valores de los atributos o incluso un error al etiquetar la muestra en problemas supervisados.

Imaginemos, por ejemplo, que estamos entrenando un modelo que dada una imagen nos diga si aparece o no una persona. Para entrenar el modelo tenemos muchísimas imágenes. Como estamos ante un problema supervisado, es necesario etiquetar las imágenes y, para ello, un conjunto de voluntarios se dedica a etiquetar cada una de ellas. Podría suceder que uno de los voluntarios no hubiese entendido bien las instrucciones y se equivocase a la hora de etiquetar algunas de las imágenes. Este error haría que el modelo creado no fuera todo lo bueno que podría ser.

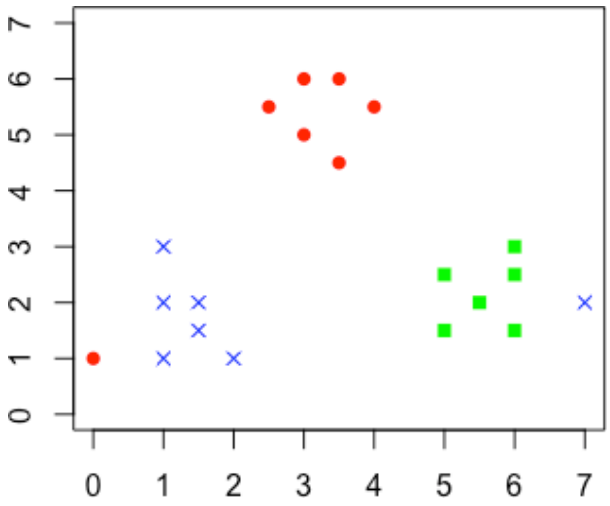
Otra fuente de problemas podría ser un malentendido en la escala de los valores que se debe usar en un determinado atributo. Imaginemos que queremos tener una base de datos con los resultados de una competición de atletismo. Existe un atributo *longitud* donde se almacena el resultado obtenido por los atletas participantes en esta prueba. Se espera que los resultados se introduzcan en metros. Pero una de las personas que introduce los datos cree que se deben introducir en centímetros. Este malentendido introduce valores en una escala diferente al resto.

Otra fuente de error podría ser un funcionamiento erróneo de un sensor usado para capturar los valores de algún atributo de la base de datos.

A las muestras que se salen de lo *normal*, puesto que presentan valores que no son normales para su clase o porque tienen un valor fuera del rango esperado, se les conoce como *outliers*. Es muy importante detectar estas muestras y eliminarlas antes de crear los modelos.

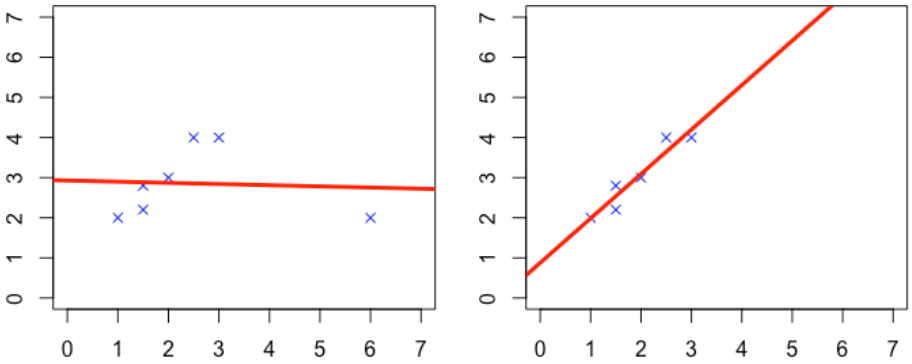
La figura 1 muestra un ejemplo de una base de datos con *outliers*. Tanto la muestra roja que está en las coordenadas [0,1] como la muestra azul localizada en las coordenadas [7,2] pueden considerarse como *outliers*. Tal como se puede comprobar, ambas muestras están lejos de donde deberían estar teniendo en cuenta el resto de muestras de su clase.

Figura 1. Base de datos de un hipotético problema supervisado con dos *outliers*



La figura 2 muestra un ejemplo de hasta qué punto la presencia de *outliers* puede ser perjudicial a la hora de construir un modelo. La gráfica izquierda de la figura 2 muestra un conjunto de datos donde hay punto [6,2] que claramente es un *outlier*. Puede comprobarse que al calcular la recta de regresión, el modelo resultante (la recta de color rojo) no es el más correcto. En la gráfica derecha de la figura 2 se ha eliminado el *outlier* y se ha calculado la recta de regresión sin tenerlo en cuenta. En este caso, la recta resultante sí es correcta.

Figura 2. Ejemplo de cómo la presencia de un único *outlier* puede reducir drásticamente la calidad de un modelo. A la izquierda el resultado de un modelo de regresión teniendo en cuenta el *outlier* situado en las coordenadas [6,2]. A la derecha el resultado de un modelo de regresión sin tenerlo en cuenta



Una de las técnicas más comunes para tratar la presencia de *outliers* en problemas supervisados es la edición de Wilson. Esta técnica consiste en crear un modelo sin la muestra que queremos analizar y aplicar el modelo, usando la muestra eliminada como entrada, para ver a qué clase pertenece. Si la clase obtenida es la correcta, entonces la muestra permanece en el conjunto de datos. Si, por el contrario, la clase es diferente a la real, entonces se asume que es un *outlier* y se elimina de la base de datos. En el ejemplo mostrado en la figura 1 se puede comprobar que la muestra localizada en las coordenadas [0,1] sería con toda probabilidad etiquetada como perteneciente a la clase azul. Como

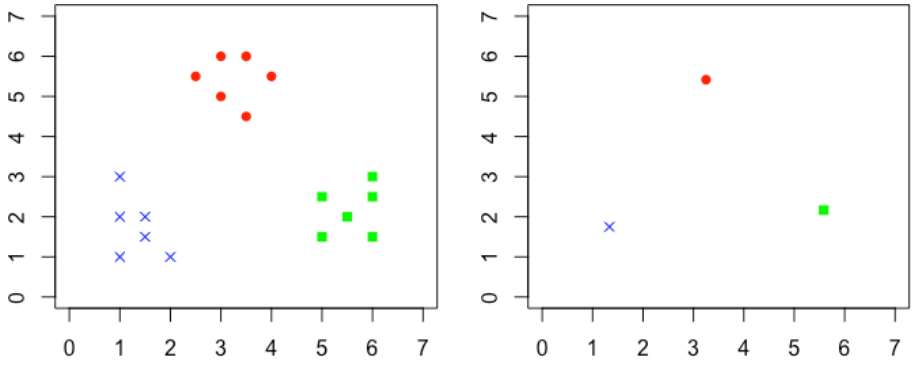
realmente es de la clase roja, se asumiría que es un *outlier* y se eliminaría de la base de datos. De forma similar, la muestra localizada en las coordenadas [7,2] sería clasificada como perteneciente a la clase verde, cuando realmente es de la clase azul. Por lo tanto, se consideraría como un *outlier* y se eliminaría de la base de datos.

2.3 Reducir muestras mediante técnicas de condensación

En otras ocasiones, el problema es que hay demasiadas muestras y nuestro equipo de procesamiento no es capaz de crear un modelo o tarda demasiado tiempo en hacerlo. Aunque siempre existe la posibilidad de adquirir un equipo mejor (si económicamente es posible), es recomendable comprobar si podemos reducir el número de muestras sin que la calidad del modelo se vea seriamente afectada. Existen técnicas que permiten reducir el número de muestras manteniendo un nivel de funcionamiento del modelo aceptable. A un conjunto de estas técnicas se las conoce como técnicas de condensación.

Una técnica de condensación muy sencilla de implementar es buscar grupos de muestras muy parecidas y reemplazarlas por su centroide. En el caso de problemas supervisados, deberemos además comprobar que tengan la misma etiqueta. La figura 3 muestra un ejemplo de la aplicación de esta técnica en un problema supervisado con tres clases. La gráfica izquierda de la figura muestra el conjunto de datos original. Tal como puede comprobarse, los tres conjuntos están claramente separados. Por lo tanto, la sustitución de todas las muestras de cada clase por su centroide (gráfica derecha de la figura 3) no empeoraría la calidad del modelo. En este caso, se ha reducido en aproximadamente un 80 % el número de muestras.

Figura 3. Ejemplo de la aplicación de una técnica de condensación. A la izquierda la base de datos original. A la derecha la base de datos tras el proceso de condensación



3. Valores ausentes

Uno de los problemas más habituales en el tratamiento previo de los datos es la ausencia de valores para un atributo determinado. Existen tres posibilidades a la hora de enfrentarse a este problema, dependiendo de lo frecuente que sea la ausencia de datos:

- Si para un determinado atributo existen muchas muestras con valores ausentes, lo más correcto es no tener en cuenta dicho atributo para crear nuestro modelo.
- Si el número de muestras con valores ausentes es bajo o moderado, se pueden usar técnicas para asignar un valor adecuado.
- También es posible que exista una muestra con muchos valores ausentes. En ese caso, sería conveniente no usar dicha muestra en nuestro modelo.

Al conjunto de técnicas que permiten asignar valores adecuados en los casos de valores ausentes se les conoce como técnicas de imputación. Las técnicas de imputación más comunes son las siguientes:

- Asignar un valor fijo, como por ejemplo el valor mínimo que puede tener este atributo.
- Asignar un valor calculado a partir de los valores existentes para ese atributo en el resto de muestras del conjunto de datos. Por ejemplo, se puede usar la media, la moda, etc.
- Usar un modelo supervisado de regresión donde la variable dependiente sea el atributo con el valor ausente y las variables independientes el resto de atributos.

Supongamos que tenemos el conjunto de datos mostrado en la tabla 2 que muestra la edad, salario, ventas y kilómetros realizados por 5 comerciales de una empresa. Como se puede comprobar, la muestra correspondiente al comercial 3 tiene un valor ausente en el atributo *Salario* y la correspondiente al comercial 5 lo tiene en el atributo *Kilómetros*.

La primera posible acción es asignar un valor fijo a los valores ausentes, como, por ejemplo, asignar el valor 0 a ambos valores ausentes. En este caso, se puede comprobar que el valor 0 en ambos atributos no tiene mucho sentido, por lo que no sería una buena idea.

Tabla 2. Datos ficticios de cinco comerciales de una empresa donde existen valores ausentes

Comercial	Edad	Salario	Ventas	Kilómetros
1	34	34300	120000	3400
2	54	24000	80000	1400
3	39		20000	1300
4	64	34000	130000	5400
5	48	28000	220000	

La segunda posible acción es imputar el valor ausente usando un valor calculado con los valores existentes del resto de muestras del conjunto. Por ejemplo, si usamos la media, el campo *Salario* del comercial 3 se imputaría con el valor $(34300+24000+35000+28000)/4 = 30075$ y el campo *kilómetros* del comercial 5 se imputaría con el valor $(3400 + 1400 + 1300 + 5400)/4 = 2875$.

La tercera posibilidad es usar un método supervisado de regresión. Para imputar el campo *Salario* usaríamos como variables independientes los atributos: *Edad*, *Ventas* y *Kilómetros*, y como variable dependiente el atributo *Salario*. Usaríamos para entrenar el modelo las muestras de los comerciales 1, 2 y 4. La muestra del comercial 5 no se usaría, pues tiene valores ausentes en el atributo *Kilómetros*. De esta forma, obtendríamos un modelo capaz de predecir el salario, a partir de los otros tres atributos. Una vez el modelo esté creado, lo podemos usar para predecir el valor a imputar del atributo *Salario*, dados los tres valores de *Edad*, *Ventas* y *Kilómetros* conocidos para el comercial 3.

De forma similar, crearíamos otro modelo de regresión para imputar el valor del atributo *Kilómetros* para el comercial 5. En este caso, usaríamos como variables independientes los atributos *Edad*, *Ventas* y *Salario* y como variable dependiente el atributo *Kilómetros*. Para entrenar, usaríamos las muestras 1, 2 y 4, aunque también podríamos usar la muestra 3 usando el valor imputado previamente para el atributo *Salario*. El modelo obtenido sería capaz de predecir el valor del atributo *Kilómetros* dados los tres valores de *Edad*, *Ventas* y *Salario* conocidos para el comercial 5.

El uso de modelos de regresión para imputar valores ausentes suele ser el método más efectivo, aunque existen casos particulares donde no se cumple esta afirmación. Dependiendo del problema de minería de datos que se quiere resolver, un método será más efectivo que otro. En un caso real, se tienen que probar varios métodos para validar cuál funciona mejor.