Preprocessament de dades

PID_00284568

Raúl Montoliu Colás

Temps mínim de dedicació recomanat: 2 hores

Universitat Oberta de Catalunya

Raúl Montoliu Colás

Enginyer en Informàtica per la Universitat Jaume I (UJI) de Castelló. Doctor en Mètodes Avançats Informàtics per la mateixa universitat. Actualment treballa com a docent al departament d'Enginyeria i Ciència de les Computadores de l'UJI i com a investigador al grup d'investigació Machine Learning for Smart Environments de l'Instituto de Nuevas Tecnologías de la Imagen (INIT). Des del 2017 col·labora com a docent a la Universitat Oberta de Catalunya (UOC).

L'encàrrec i la creació d'aquest recurs d'aprenentatge UOC han estat coordinats pel professor: Julià Minguillón Alfonso

Primera edició: setembre 2021

Av. Tibidabo, 39-43, 08035 Barcelona
Autoria: Raúl Montoliu Colás
Producció: FUOC

Índex

Introducció

En pràcticament tots els processos de mineria de dades sol ser necessari que les dades, una vegada seleccionades, hagin de ser modificades i preparades. D'aquesta manera, serà possible crear models de mineria de dades que funcionin de manera òptima.

En aquest mòdul es comentaran les tècniques de preparació de dades més habituals:

- La transformació de dades: en concret, la normalització i la discretització.
- El tractament dels valors no observats.
- La reducció de la dimensionalitat de les dades.

Les tècniques que es presenten en aquest mòdul són bastant independents, fins a cert punt, del tipus de model que es farà servir.

1. Transformacions de valors

Per transformacions de valors entenem les modificacions dins del tipus de valors que poden adoptar tots o alguns dels atributs. Les operacions més habituals són la normalització i la discretització de dades.

1.1. Normalització

La normalització consisteix a situar les dades sobre una escala de valors equivalents que permeti la comparació d'atributs que prenen valors en dominis o rangs diferents.

La normalització és útil, o necessària, per a diversos mètodes de construcció de models, com per exemple les xarxes neuronals o alguns mètodes basats en distàncies, com el dels veïns més pròxims. En efecte, si no hi ha normalització prèvia, els mètodes esmentats tendeixen a quedar esbiaixats per la influència dels atributs amb valors més alts, fet que distorsiona el resultat.

Per exemple, imaginem que a la taula 1 hi tenim les dades d'un conjunt de comercials d'una empresa fictícia. D'una banda, la diferència d'edat entre el comercial més jove (comercial 1) i el més gran (comercial 4) és de $|34-64| = 30$ anys. De l'altra, la diferència de salari anual entre aquests comercials és de $|34300 - 34000| = 300$ euros. Mentre que la diferència d'edat entre els dos comercials és bastant elevada, la diferència en el salari anual és raonablement baixa. No obstant això, el valor de la diferència en el cas del salari és 10 vegades superior al de l'edat (300 enfront de 30). Com es pot comprovar, la diferència de salari pot arribar a influenciar més que la diferència d'edat, per una qüestió de les magnituds que es fan servir per representar aquests conceptes.

Taula 1. Exemple hipotètic de les dades d'un conjunt de comercials d'una empresa

Comercial	Edat	Salari	Vendes	Quilòmetres
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 normalització evitarà aquests problemes perquè permet comparar tots els atributs en igualtat de condicions.

1.1.1. Normalització pel màxim

La normalització pel màxim consisteix a trobar el valor màxim x_{max} de l'atribut que s'ha de normalitzar X i dividir tots els valors de l'atribut per x_{max}. Formalment, es defineix d'aquesta manera:

$$x_{max} = \max_{i=1,\ldots,N} x_i \quad (1)$$

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

en què x_i és cadascun dels valors de l'atribut $X = \{x_1, x_2, \ldots, x_N\}$, N és el total de mostres i x'_i és cadascun dels nous valors de l'atribut després del procés de normalització, és a dir, $X' = \{x'_1, x'_2, \ldots, x'_N\}$.

Amb aquesta mena de normalització, ens assegurem que els valors de l'atribut estaran en el rang $[\frac{\min_{i=1,\ldots,N} x_i}{x_{max}}, 1]$.

Per exemple, donat l'atribut *Edat* de la taula 1, que té els valors 34, 54, 39, 64 i 48, el valor màxim és 64. Per tant, els valors normalitzats seran 34/64, 54/64, 39/64, 64/64 i 48/64, del que en resulta (aproximant amb tres decimals) 0.531, 0.844, 0.609, 1.000 i 0.750.

De manera similar, podem normalitzar l'atribut *Salari*, que té els valors 34300, 24000, 45000, 34000 i 28000. El màxim serà 45000 i els valors normalitzats 34300/45000, 24000/45000, 45000/45000, 34000/45000 i 28000/45000, del que en resulta (aproximant amb tres decimals): 0.762, 0.533, 1.000, 0.756 i 0.622.

Després del procés de normalització, la diferència entre el primer i el quart comercial en l'atribut *Edat* és de $|0.531 - 1.00| = 0.469$, mentre que en l'atribut *Salari* és de $|0.762 - 0.755| = 0.007$. És a dir, ara el primer i quart comercial estan més a prop en l'atribut *Salari* que en el d'*Edat*.

1.1.2. Normalització per la diferència

La normalització per la diferència mira de compensar l'efecte de la distància del valor que tractem respecte al màxim dels valors observats. La normalització per la diferència consisteix a dur a terme la transformació següent:

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

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

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

Amb aquesta mena de normalització, ens assegurem que els valors de l'atribut estaran en el rang [0,1].

Per exemple, normalitzarem els atributs *Edat* i *Salari* mostrats a la taula 1. El mínim i màxim de l'atribut *Edat* és 34 i 64, respectivament. Per tant, els nous valors després del procés de normalització seran $\frac{34-34}{64-34}$ = 0.00, $\frac{54-34}{64-34}$ = 0.67, $\frac{39-34}{64-34}$ = 0.17, $\frac{64-34}{64-34}$ = 1.00 i $\frac{48-34}{64-34}$ = 0.47. En el cas de l'atribut *Salari*, els nous valors, tenint en compte que el mínim i el màxim són 24000 i 45000, respectivament, seran $\frac{34300-24000}{45000-24000}$ = 0.49, $\frac{24000-24000}{45000-24000}$ = 0.00, $\frac{45000-24000}{45000-24000}$ = 1.00, $\frac{34000-24000}{45000-24000}$ = 0.48 i $\frac{28000-24000}{45000-24000}$ = 0.19.

1.1.3. Escalat decimal

L'escalat decimal permet reduir el valor d'un atribut a un cert nombre de potències de 10. Aquesta transformació resulta especialment útil per tractar amb valors elevats, com els atributs *Salari* i *Vendes* de la taula 1. La transformació es durà a terme tal com es mostra a continuació:

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

en què *j* ha de ser tal que mantingui el màxim valor que pot adoptar l'atribut per sota d'1.

Per exemple, per a l'atribut *Salari*, *j* seria igual a 5. D'aquesta manera, els nous valors per a aquest atribut, després del procés de normalització, serien $\frac{34300}{10^5}$ = 0.343, $\frac{24000}{10^5}$ = 0.240, $\frac{45000}{10^5}$ = 0.450, $\frac{34000}{10^5}$ = 0.340 i $\frac{28000}{10^5}$ = 0.280. No obstant això, per a l'atribut *Vendes*, el valor correcte de *j* seria 6.

En fer servir aquest tipus de normalització, el nou rang de valors es trobarà a l'interval següent: $[\frac{\min_{i=1,\ldots,N} x_i}{10^j}, 1]$.

1.1.4. Normalització basada en la desviació estàndard

Els mètodes anteriors no tenen en compte la distribució dels valors existents. El mètode de normalització basada en la desviació estàndard assegura que s'obtenen valors que tenen com a propietat que la seva mitjana és 0 i la seva

desviació estàndard val 1.

Aquesta tècnica consisteix a fer la transformació següent sobre els valors dels atributs:

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

en què μ és la mitjana de tots els valors de l'atribut i σ és la seva desviació estàndard. El nou rang de possibles valors és $[\frac{x_{min}-\mu}{\sigma}, \frac{x_{max}-\mu}{\sigma}]$.

Per exemple, per a l'atribut *Edat*, la mitjana és $\mu = 47.8$ i la seva desviació estàndard $\sigma = 11.92$ (calculada amb $N - 1$ al denominador). Per tant, els nous valors d'aquest atribut, després de fer la transformació, seran $\frac{34-47.8}{11.92} = -1.16$, $\frac{59-47.8}{11.92} = 0.52$, $\frac{39-47.8}{11.92} = -0.74$, $\frac{64-47.8}{11.92} = 1.36$ i $\frac{48-47.8}{11.92} = 0.02$. En el cas de l'atribut *Salari*, la mitjana és $\mu = 33060$ i la seva desviació estàndard és $\sigma = 7947.83$. Per tant, els nous valors d'aquest atribut, després de fer la transformació, seran $\frac{34300-33060}{7947.83} = 0.16$, $\frac{24000-33060}{7947.83} = -1.14$, $\frac{45000-33060}{7947.83} = 1.50$, $\frac{34000-33060}{7947.83} = 0.12$ i $\frac{28000-33060}{7947.83} = -0.64$.

1.2. Discretització

La discretització consisteix bàsicament a establir un criteri per mitjà del qual es puguin dividir els valors d'un atribut en dos o més conjunts disjunts.

Ara bé, aquests no són els únics motius existents per discretitzar dades. A continuació n'esmentem d'altres:

1) Cost computacional. Tenint en compte que el conjunt de valors sobre el qual es treballarà després de discretitzar implica una reducció dels valors per tractar, el nombre de comparacions i càlculs que haurà de dur a terme el mètode de mineria de dades corresponent serà inferior.

2) Velocitat en el procés d'aprenentatge. S'ha demostrat empíricament que el temps necessari per dur a terme un procés d'entrenament d'un mètode de mineria de dades és més curt si es fa ús de dades discretitzades.

3) Emmagatzematge. En general, els valors discrets necessiten menys memòria per ser emmagatzemats.

4) Grandària del model resultant. Quan es treballa amb dades contínues, els models classificatoris que s'obtenen són comparativament més elevats. Per

exemple, els arbres de decisió acostumen a tenir un factor de ramificació més alt quan es treballa amb dades contínues que quan es treballa amb dades discretes.

5) Comprensió. La comprensió d'alguns models millora en bona part quan es descriu l'element utilitzant menys termes.

Evidentment, tot mètode de discretització està obligat a mantenir o millorar les característiques del model que ajuda a construir. Per exemple, si introduïm un procés de classificació i obtenim taxes d'error més altes o de predicció més baixes que sense discretitzar, poc hem guanyat. Per tant, el que busquen la majoria dels mètodes de discretització és mantenir la informació associada a l'atribut que es discretitza.

Un inconvenient que normalment s'esmenta quan es parla de mètodes de discretització és precisament la pèrdua d'informació sobre els valors continus. Aquest tipus d'efecte pot tenir una influència notable en la precisió dels mètodes d'aprenentatge.

A continuació es descriuen tres mètodes per discretitzar.

1.2.1. Mètode de partició en intervals de la mateixa amplitud

El mètode de partició en intervals de la mateixa amplitud és una tècnica de discretització en què, donat un atribut X numèric, es duen a terme els passos següents:

1) Calcular el valor mínim x_{min} i el valor màxim x_{max}.

2) Fixar el número k d'intervals que es vol aconseguir.

3) Dividir el rang de valors $[x_{min},x_{max}]$ en k intervals en què la distància entre el màxim i el mínim de cada interval sigui la mateixa i igual a $(x_{max} - x_{min})/k$.

Partint de les dades de la taula 1, podem discretitzar l'atribut *Edat* en 3 intervals ($k = 3$). Sabent que el mínim de l'atribut és 34 i el màxim 64, la grandària de cada interval serà $(64 – 34)/3 = 10$. Per tant, el primer interval contindrà els valors [34,44), el segon [44,54) i el tercer i últim [54,64]. A la base de dades, podríem modificar el valor de cada mostra per a aquest atribut per 1, 2 i 3, indicant-hi el número de l'interval. D'aquesta manera, a la primera mostra (34) li assignaríem el valor 1, a la segona (54) el valor 3, a la tercera (39) el valor 1, a la quarta (64) el valor 3 i a l'última (48) el valor 2.

Per facilitar la compressió dels valors introduïts en aquesta base de dades, hauríem d'acompanyar la base de dades amb una descripció per explicar el significat d'aquests valors. Per exemple, podríem indicar que 1 significa *jove*, 2 *mitjana edat* i 3 *veterà*.

Aquest mètode té alguns inconvenients:

1) Dona la mateixa importància a tots els valors, independentment de la seva freqüència d'aparició.

2) En cas que vulguem utilitzar el mètode com a punt de partida de classificació, ens podem trobar que es barregin dins d'un mateix interval valors que corresponen a classes diferents.

3) Amb aquest mètode no hi ha manera de trobar un valor de k que sigui prou bo.

1.2.2. Obtenció d'intervals de discretització d'igual freqüència

Un dels problemes que hem esmentat en comentar el mètode anterior és que pot generar intervals en els quals les diferents classes o valors es distribueixin amb freqüències diferents. Es pot introduir informació sobre la freqüència requerida de la manera següent:

- indicant el nombre d'intervals que cal obtenir,
- indicant la freqüència que es vol obtenir per als intervals.

L'algorisme 1 presenta els passos que cal seguir.

Algorisme 1 Algorisme per obtenir intervals de discretització d'igual freqüència. $X = \{x_1, x_2, \ldots, x_N\}$ és el conjunt de valors de l'atribut, k el nombre d'intervals buscat i $|\bullet|$ indica el nombre d'elements.

1: Ordenar X
2: $f \leftarrow N/k$
3: $i \leftarrow 1$
4: $j \leftarrow 1$
5: Assignar el valor x_1 a l'interval I_1
6: **per a** $j \leftarrow 2,N$ **fer**
7: **si** $x_j \neq x_{j-1}$ **i** $|I_i| \geq f$ **llavors**
8: $i \leftarrow i + 1$
9: **fi si**
10: Assignar el valor x_j a l'interval I_i
11: **fi per a**

Suposem que l'atribut X té els valors següents (ja ordenats): {22,22,27,27,28,28, 31,31,31,40,50,50} i el volem discretitzar en 3 intervals ($k = 3$). En total, hi ha 12 valors ($N = 12$); per tant, $f = N/k = 4$. L'algorisme 1 assignaria cada mostra a un interval tal com es descriu a continuació:

- $x_1 = 22$: s'assigna al primer interval.
- $x_2 = 22$: s'assigna al primer interval, ja que $x_2 == x_1$.
- $x_3 = 27$: s'assigna al primer interval, ja que, encara que $x_3 \neq x_2$, el nombre d'elements del primer interval és més petit que f.
- $x_4 = 27$: s'assigna al primer interval, ja que $x_4 == x_3$.
- $x_5 = 28$: s'assigna al segon interval, ja que $x_5 \neq x_4$ i el nombre d'elements del primer interval ja té el valor màxim permès.
- $x_6 = 28$: s'assigna al segon interval, ja que $x_6 == x_5$.
- $x_7 = 31$: s'assigna al segon interval, ja que, encara que $x_7 \neq x_6$, el nombre d'elements del segon interval és més petit que f.
- $x_8 = 31$: s'assigna al segon interval, ja que $x_8 == x_7$.
- $x_9 = 31$: s'assigna al segon interval, ja que $x_9 == x_8$.
- $x_{10} = 40$: s'assigna al tercer interval, ja que $x_{10} \neq x_9$ i el nombre d'elements del segon interval ha superat el valor màxim permès.
- $x_{11} = 50$: s'assigna al tercer interval, ja que, encara que $x_{11} \neq x_{10}$, el nombre d'elements del tercer interval és més petit que f.
- $x_{12} = 50$: s'assigna al tercer interval, ja que $x_{12} == x_{11}$.

1.2.3. Mètode de partició basat en l'algorisme *k-means*

Un altre mètode possible és fer servir algunes de les idees de l'algorisme no supervisat d'agregació *k-means*.

La millor manera d'explicar aquest mètode és mitjançant un exemple. Suposem que el volem discretitzar en 3 intervals i que tenim el conjunt de valors següent per a l'atribut X (ja ordenats): {20,21,28,29,30,31,31,39,40,51, 52,52}.

En el primer pas, assignem 4 valors a cadascun dels 3 intervals, ja que 12/3 = 4. Per tant, els intervals inicials contindran els valors:

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

El pas següent consisteix a calcular el valor mitjà de cada interval, de manera que obtenim 24.5 per a I_1, 32.75 per a I_2 i 48.75 per a I_3. Ara calculem per a cada mostra si la seva distància al centroide de l'interval en què es troba és més petita que la distància al centroide de l'interval veí. Si la distància és més petita o igual, la mostra es manté en el mateix interval. Si és més elevada, la mostra es canvia d'interval.

Per exemple, la mostra amb valor 28 que es troba al primer interval està a 28 – 24.5 = 3.5 del centroide de I_1 i a 32.75 – 28 = 4.75 del centroide de I_2. Per tant, es manté a l'interval I_1. No obstant això, la mostra amb valor 29 es troba més lluny del centroide del primer interval (29 – 24.5 = 4.5) que del segon (32.75 – 29 = 3.75), per la qual cosa es mou a l'interval I_2. De manera similar, la mostra amb valor 40 es troba més a prop del centroide de l'interval I_2 que del I_3, per la qual cosa es mou al I_2.

Després d'aquesta primera iteració, els intervals contindran els valors:

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

Ara els centroides dels tres intervals seran 23, 33.3 i 51.6, respectivament. Totes les mostres estan ben etiquetades per la qual cosa el procés finalitza amb els intervals anteriors.

2. Reducció de la dimensionalitat

Una vegada tenim les dades en el format adequat per a la mena de model que es vol obtenir i el mètode per construir-lo, encara és possible aplicar una sèrie nova d'operacions amb la finalitat de complir dos objectius: reduir el nombre d'atributs que cal tenir en compte i reduir el nombre de mostres que cal tractar, assegurant, així mateix, que es mantindrà o es millorarà la qualitat del model resultant.

El motiu per efectuar la reducció acostuma a ser triple:

- El programa de construcció del model triat no pot tractar la quantitat de dades de què disposem. En aquest cas, haurem d'obtenir un subconjunt que sí que es pugui tractar, però intentant que la qualitat del model no es vegi compromesa.

- El programa les pot tractar, però el temps requerit per construir el model és inacceptablement llarg. De manera similar al cas anterior, l'objectiu serà obtenir un subconjunt de les dades que requereixi un temps inferior de construcció del model, però mantenint un model amb un nivell de qualitat alt.

- La presència d'algunes mostres i/o atributs, que lluny de beneficiar l'eficàcia del model, la perjudica.

2.1. Reduir atributs

La reducció del nombre d'atributs consisteix a trobar un subconjunt dels atributs originals que permeti obtenir models de la mateixa qualitat (o fins i tot superior) que els que s'obtindrien utilitzant tots els atributs. Aquest problema també es coneix com a *selecció de característiques*.

Un algorisme trivial seria aquell en el qual construïm el model amb tots els atributs, i per a cada atribut, es construeix un nou model eliminant l'atribut en qüestió. Si el model sense un atribut és millor o comparable amb l'original, llavors podem eliminar l'atribut. Una vegada eliminat el primer atribut, continuaríem el procés tractant d'eliminar un nou atribut d'entre els atributs supervivents. El procés continuaria fins que no fos possible trobar un atribut que, en eliminar-lo, millorés el model o que permetés obtenir uns resultats comparables.

Existeixen algorismes més eficaços i eficients per abordar aquest problema.

2.2. Eliminar *outliers* mitjançant tècniques d'edició

En termes generals, els models de mineria de dades funcionen millor com més mostres tinguin. No obstant això, a vegades algunes mostres, lluny de beneficiar-los, perjudiquen els models. Això es pot deure a molts factors, i el més freqüent és un error en el procés de captura dels valors dels atributs o fins i tot un error en etiquetar la mostra en problemes supervisats.

Imaginem, per exemple, que estem entrenant un model perquè donada una imatge ens digui si hi apareix o no una persona. Per entrenar el model, tenim moltíssimes imatges. Com que estem amb un problema supervisat, cal etiquetar les imatges, de manera que un conjunt de voluntaris s'hi dedica. Podria passar que un dels voluntaris no hagués entès bé les instruccions i s'equivoqués a l'hora d'etiquetar algunes de les imatges. Aquest error faria que el model creat no fos tan bo com podria ser.

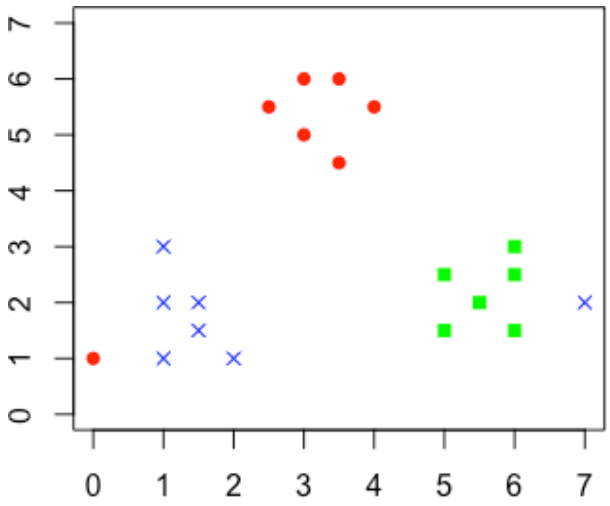
Una altra font de problemes podria ser un malentès en l'escala dels valors que s'ha de fer servir en un determinat atribut. Imaginem que volem tenir una base de dades amb els resultats d'una competició d'atletisme. Existeix un atribut *Longitud* en què s'emmagatzema el resultat obtingut pels atletes participants en aquesta prova. S'espera que els resultats s'introdueixin en metres. Però una de les persones que introdueix les dades creu que s'han d'introduir en centímetres. Aquest malentès introdueix valors en una escala diferent de la resta.

Una altra font d'error podria ser un funcionament erroni d'un sensor usat per capturar els valors d'algun atribut de la base de dades.

Les mostres que se surten del que es considera *normal*, ja que presenten valors que no són normals per a la seva classe o perquè tenen un valor fora del rang esperat, es coneixen com a *outliers*. És molt important detectar aquestes mostres i eliminar-les abans de crear els models.

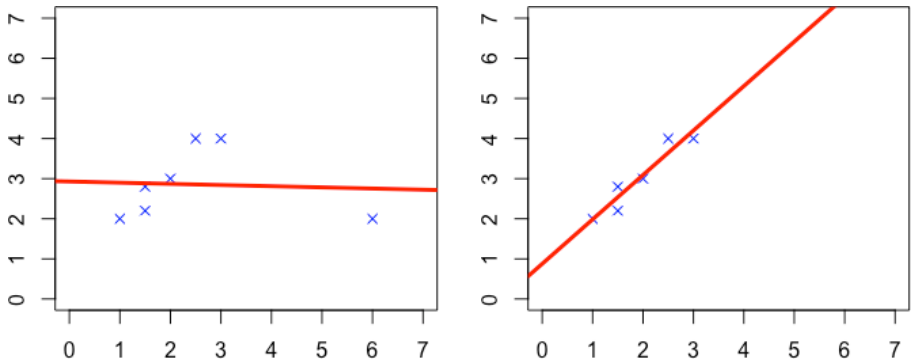
La figura 1 mostra un exemple d'una base de dades amb *outliers*. Tant la mostra vermella que està a les coordenades [0,1] com la mostra blava localitzada a les coordenades [7,2] es poden considerar *outliers*. Tal com es pot comprovar, totes dues mostres estan lluny d'on haurien d'estar tenint en compte la resta de mostres de la seva classe.

Figura 1. Base de dades d'un hipotètic problema supervisat amb dos *outliers*



La figura 2 mostra un exemple de fins a quin punt la presència d'*outliers* pot ser perjudicial a l'hora de construir un model. La gràfica esquerra de la figura 2 mostra un conjunt de dades en què hi ha el punt [6,2], que clarament és un *outlier*. Es pot comprovar que, en calcular la recta de regressió, el model resultant (la recta de color vermell) no és el més correcte. A la gràfica dreta de la figura 2 s'ha eliminat l'*outlier* i s'ha calculat la recta de regressió sense tenir-lo en compte. En aquest cas, la recta resultant sí que és correcta.

Figura 2. Exemple de com la presència d'un únic *outlier* pot reduir dràsticament la qualitat d'un model. A l'esquerra, resultat d'un model de regressió tenint en compte l'*outlier* situat a les coordenades [6,2]. A la dreta, resultat d'un model de regressió sense tenir-lo en compte.



Una de les tècniques més corrents per tractar la presència d'*outliers* en problemes supervisats és l'edició de Wilson. Aquesta tècnica consisteix a crear un model sense la mostra que volem analitzar i aplicar-lo fent servir la mostra eliminada com a entrada, per veure a quina classe pertany. Si la classe obtinguda és la correcta, llavors la mostra es manté en el conjunt de dades. Si, per contra, la classe és diferent a la real, llavors s'assumeix que és un *outlier* i s'elimina de la base de dades. A l'exemple de la figura 1 es pot comprovar que la mostra localitzada a les coordenades [0,1] seria amb tota probabilitat etiquetada com a pertanyent a la classe blava. Com que realment és de la classe vermella, s'assumiria que és un *outlier* i s'eliminaria de la base de dades. De manera similar,

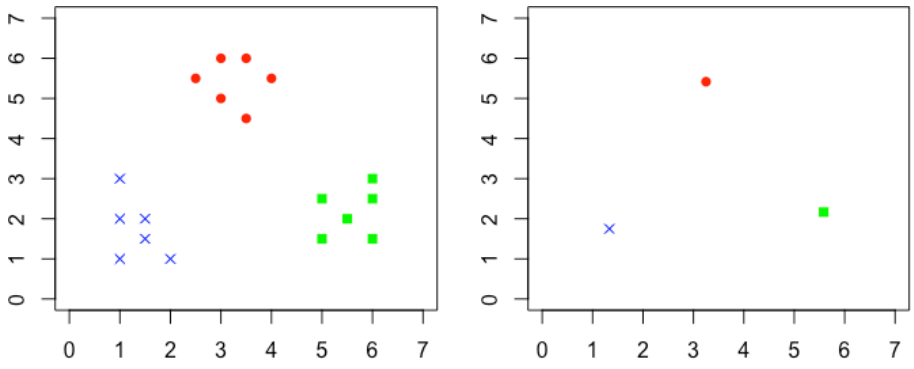
la mostra localitzada a les coordenades [7,2] es classificaria com a pertanyent a la classe verda, quan realment és de la classe blava. Per tant, es consideraria com un *outlier* i s'eliminaria de la base de dades.

2.3. Reduir mostres mitjançant tècniques de condensació

En altres casos, el problema és que hi ha massa mostres i el nostre equip de processament no és capaç de crear un model o triga massa temps a fer-ho. Tot i que sempre hi ha la possibilitat d'adquirir un equip millor (si econòmicament es pot), és recomanable comprovar si podem reduir el nombre de mostres sense que la qualitat del model es vegi greument afectada. Hi ha tècniques que permeten reduir el nombre de mostres mantenint un nivell acceptable de funcionament del model. El conjunt d'aquestes tècniques es coneix com a *tècniques de condensació*.

Una tècnica de condensació molt senzilla d'implementar és buscar grups de mostres molt semblants i reemplaçar-los pel seu centroide. En el cas dels problemes supervisats, també caldrà comprovar que tinguin la mateixa etiqueta. La figura 3 mostra un exemple de l'aplicació d'aquesta tècnica en un problema supervisat amb tres classes. La gràfica esquerra de la figura mostra el conjunt de dades original. Tal com es pot comprovar, els tres conjunts estan clarament separats. Per tant, la substitució de totes les mostres de cada classe pel seu centroide (gràfica dreta de la figura 3) no empitjoraria la qualitat del model. En aquest cas, s'ha reduït en aproximadament un 80 % el nombre de mostres.

Figura 3. Exemple de l'aplicació d'una tècnica de condensació. A l'esquerra, base de dades original. A la dreta, base de dades després del procés de condensació.



3. Valors absents

Un dels problemes més habituals en el tractament previ de les dades és l'absència de valors per a un atribut determinat. Existeixen tres possibilitats a l'hora d'enfrontar-se a aquest problema, depenent de com de freqüent sigui l'absència de dades:

- Si per a un determinat atribut hi ha moltes mostres amb valors absents, el més correcte és no tenir en compte aquest atribut per crear el nostre model.
- Si el nombre de mostres amb valors absents és baix o moderat, es poden fer servir tècniques per assignar un valor adequat.
- També és possible que hi hagi una mostra amb molts valors absents. En aquest cas, seria convenient no fer-la servir en el nostre model.

El conjunt de tècniques que permeten assignar valors adequats en els casos de valors absents es coneixen com a *tècniques d'imputació*. Les tècniques d'imputació més comunes són les següents:

- Assignar un valor fix, com el valor mínim que pot tenir aquest atribut.
- Assignar un valor calculat a partir dels valors existents per a aquest atribut a la resta de mostres del conjunt de dades. Per exemple, la mitjana, la moda, etc.
- Fer servir un model supervisat de regressió en què la variable dependent sigui l'atribut amb el valor absent i les variables independents, la resta d'atributs.

Suposem que tenim el conjunt de dades de la taula 2, que mostra l'edat, el salari, les vendes i els quilòmetres recorreguts per 5 comercials d'una empresa. Com es pot comprovar, la mostra corresponent al comercial 3 té un valor absent a l'atribut *Salari* i la corresponent al comercial 5 el té a l'atribut *Quilòmetres*.

La primera acció possible és assignar un valor fix als valors absents, com per exemple assignar el valor 0 als dos valors absents. En aquest cas, es pot comprovar que el valor 0 als dos atributs no té gaire sentit, per la qual cosa no seria una bona idea.

Taula 2. Dades fictícies de cinc comercials d'una empresa en què hi ha valors absents.

Comercial	Edat	Salari	Vendes	Quilòmetres
1	34	34300	120000	3400
2	54	24000	80000	1400
3	39		20000	1300
4	64	34000	130000	5400
5	48	28000	220000	

La segona acció possible és imputar el valor absent fent servir un valor calculat amb els valors existents de la resta de mostres del conjunt. Per exemple, si fem la mitjana, el camp *Salari* del comercial 3 s'imputaria amb el valor $(34300+24000+34000+28000)/4 = 30075$ i el camp *Quilòmetres* del comercial 5 s'imputaria amb el valor $(3400+1400+1300+5400)/4 = 2875$.

La tercera possibilitat és fer servir un mètode supervisat de regressió. Per imputar el camp *Salari*, utilitzaríem com a variables independents els atributs *Edat*, *Vendes* i *Quilòmetres*, i com a variable dependent, l'atribut *Salari*. Usaríem per entrenar el model les mostres dels comercials 1, 2 i 4. La mostra del comercial 5 no s'utilitzaria, perquè té valors absents a l'atribut *Quilòmetres*. D'aquesta manera, obtindríem un model capaç de predir el salari a partir dels altres tres atributs. Una vegada creat el model, el podem usar per predir el valor que volem imputar a l'atribut *Salari*, donats els tres valors d'*Edat*, *Vendes* i *Quilòmetres* coneguts per al comercial 3.

De manera similar, crearíem un altre model de regressió per imputar el valor de l'atribut *Quilòmetres* per al comercial 5. En aquest cas, usaríem com a variables independents els atributs *Edat*, *Vendes* i *Salari*, i com a variable dependent, l'atribut *Quilòmetres*. Per entrenar, usaríem les mostres 1, 2 i 4, tot i que també podríem fer servir la mostra 3 amb el valor imputat prèviament per a l'atribut *Salari*. El model obtingut seria capaç de predir el valor de l'atribut *Quilòmetres*, donats els tres valors d'*Edat*, *Vendes* i *Salari* coneguts per al comercial 5.

L'ús de models de regressió per imputar valors absents sol ser el mètode més efectiu, tot i que hi ha casos particulars en què no es compleix aquesta afirmació. Depenent del problema de mineria de dades que es vulgui resoldre, un mètode serà més efectiu que un altre. En un cas real, s'han de provar diversos mètodes per validar quin funciona millor.