Práctica 2 - Data cleaning

Patricia Ferreiro Alonso

June 11, 2018

Presentación

En esta práctica se elabora un caso práctico orientado a aprender a identificar los datos relevantes para un proyecto analítico y usar las herramientas de integración, limpieza, validación y análisis de las mismas.

Competencias

En esta práctica se desarrollan las siguientes competencias del Máster de Data Science:

1. Capacidad de analizar un problema en el nivel de abstracción adecuado a cada situación y aplicar las habilidades y conocimientos adquiridos para abordarlo y resolverlo.
2. Capacidad para aplicar las técnicas específicas de tratamiento de datos (integración, transformación, limpieza y validación) para su posterior análisis.

Respuestas

Siguiendo las principales etapas de un proyecto analítico, las diferentes tareas a realizar (y justificar) son las siguientes:

1. Descripción del dataset

Como fuente de datos de este estudio se ha utilizado el repositorio de datos del "Center for Machine Learning and Intelligent Systems" de la Universidad de California, Irvine,, disponible en la siguiente url: https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients.

En concreto, se analizará el siguiente dataset, que contiene datos sobre deudas en pagos con tarjetas de crédito: https://archive.ics.uci.edu/ml/machine-learning-databases/00350/

Dichos datos se utilizaron en una investigación de gestión de riesgos del sector bancario, concretamente para la predicción del riesgo de morosidad de los nuevos clientes de un banco Taiwanés. Para ello, se aplicaron seis algoritmos distintos de inteligencia artifical, con el objetivo de identificar cuál de ellos producía los resultados más precisos y eficientes. No obstante, en este trabajo nos centraremos tan sólo en el preprocesado de los datos, esto es, su adaptación, limpieza y preparación para posibilitar la posterior utilización en un algoritmo explicativo/predictivo.

Disponemos de 30000 entradas con 24 atributos cada una, correspondientes a clientes de una entidad bancaria. Éstos incluyen datos personales como edad, género, estado civil y nivel educativo, así como la interacción del cliente con el banco, en términos de créditos e historial de pagos/impagos.

2. Integración y selección de los datos de interés

Para empezar, exportamos los datos en R nombrando al dataset resultante "default" ("impago" en inglés) y la librería "readxl", que posibilita la lectura de formato .xsl. El comando skip = 1 fuerza que se ignore la primera fila durante la importación, ya que tan sólo contiene la numeración de las columnas. En su lugar, utilizaremos la segunda fila, que sí contiene los nombres descriptivos de las variables:

```
library(readxl)
default <- read_excel("~/data/default of credit card clients.xls",
    col_names = FALSE, skip = 1)
View(default)
```

El fichero consta de 30000 observaciones o registros, esto es, los datos correspondientes a 30000 clientes de una entidad bancaria.

La investigación utilizó una variable binaria "default payment" o indicando el riesgo de impago del cliente (Sí=1, No=0) como variable de respuesta. Las 23 variables restantes se utilizaron con variables explicatorias:

- ID: identificador número único de cliente.
- LIMIT_BAL: límite crédito concedido al cliente.
- SEX: sexo del cliente (1=hombre, 2=mujer).
- EDUCATION: educación del cliente (1=formación profesional, 2=universidad, 3=instituto, 4=otros).
- MARRIAGE: estado civil del cliente (1=casado, 2=soltero, 3=otros).
- AGE: edad del cliente.
- PAY_0-5: estado de los pagos mensuales del cliente desde Abril hasta Septiembre de 2005 (-1= pagado a tiempo, 1=retraso de 1 mes en el pago; ... ; 9=retraso de 9 meses en el pago o más).
- BILL_AMT1-6: historial de facturas previas desde Abril hasta Septiembre de 2005.
- PAY_AMT1-5: historial de pagos previos desde Abril hasta Septiembre de 2005.

3. Limpieza de los datos

3.1 Detección y tratamiento de datos nulos o vacíos

Nos interesa comprobar la calidad de los datos. Primeramente, editamos el nombre de la columna de control a fin de agilizar el tratamiento del dataset:

```
> names(default)[names(default) == 'default payment next month'] <- 'DEFAULT_RES'
```

Seguidamente, procedemos a detectar si una columna contiene algún valor nulo mediante la función "is.na":

```
> sum(is.na(default$PAY_0))
[1] 0
> sum(is.na(default$PAY_AMT6))
[1] 0
> sum(is.na(default$EDUCATION))
[1] 0
```

que devuelve 0 en caso contrario.

De forma más general, podemos aplicar esa misma función al set de datos completo:

```
> apply(default, 2, function(x) any(is.na(x)))
```

Mediante el uso de "apply", se nos devuelve TRUE si is.na(x) se cumle para x, esto es, si algún elemento es nulo y FALSE en caso contrario. Al ejecutar el comando anterior se muestran los resultados para todas las columnas del dataset, resultado que se ha omitido en este documento por brevedad. Observamos que devuelve FALSE para todos los atributos, por lo que nuestro dataset es consistente y no contiene valores nulos.

De forma análoga, la siguiente función nos devolverá TRUE si existe algún elemento en la columna especificada que sea igual a cero:

```
> apply(default, 2, function(x) any(is.na(x)))
```

La mayoría de los atributos devuelven TRUE, por lo que contienen al menos un valor igual a cero. Los atributos LIMIT_BAL, SEX y AGE son los únicos que devuelven FALSE. Este resultado tiene sentido, ya que el crédito concedido para una tarjeta siempre es mayor que 0, el atributo sexo está binarizado como (1 = hombre; 2 = mujer) y la edad de un propietario de tarjeta de crédito es claramente mayor que 0.

Examinemos pues, la consistencia de los demás atributos.

- EDUCATION: atributo discretizado del 1 al 4. Un valor 0 indica desconocimiento del valor real o errores en la inserción u obtención de los datos.
- MARRIAGE: atributo discretizado del 1 al 3. Un valor 0 indica desconocimiento del valor real o errores en la inserción u obtención de los datos.
- PAY_01-06: atributos correspondiente al historial de pagos de los últimos 6 meses. del 1 al 3. Pueden darse meses en los que no se realiza ningún pago, por lo que un valor 0 resulta coherente.
- BILL_AMT1-6: atributos correspondiente al historial de facturas de los últimos 6 meses. del 1 al 3. Pueden darse meses en los que no se emita ninguna factura, por lo que un valor 0 resulta coherente.
- DEFAULT: atributo binario que determina la predicción de si existe riesgo (1) o no (0) de impago.

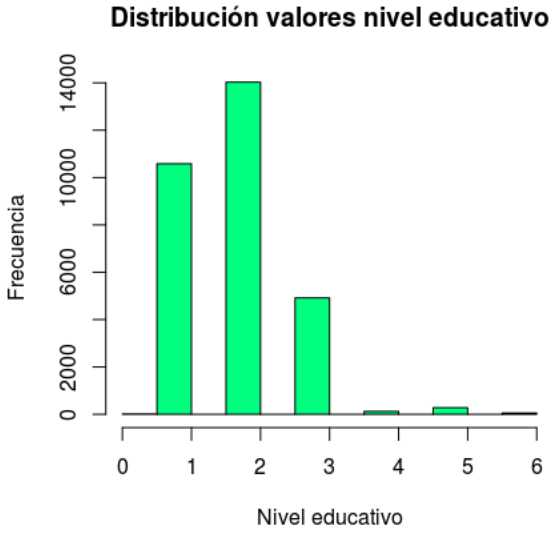
Los únicos atributos que podemos afirmar presentan valores nulos equivocadamente son EDUCATION y MARRIAGE. Calculamos cómo se distribuyen los valores para el atributo EDUCATION:

```
> table(default$EDUCATION)
```

```
    0     1     2     3     4     5     6
   14 10585 14030  4917   123   280    51
```

Gráficamente, podemos visualizarlo como un histograma:

```
> hist(default$EDUCATION,
      main = "Distribucion valores nivel educativo",
      ylab = "Frecuencia",
      xlab = "Nivel educativo",
      col = "springgreen")
```



Los valores 0 tan sólo corresponden al 0.0466% de las muestras:

```
> (sum(default$EDUCATION==0)/sum(default$EDUCATION!=0))*100
[1] 0.04668845
```

Como también hemos detectado otros valores no previstos (5 y 6) decidimos agrupar todos los valores desconocidos dentro de la clase genérica 4 "otros". Por tanto, cualquier valor de la columna EDUCATION que sea igual a 0,5 ó 6, valdrá ahora 4:

```
> default[default$EDUCATION %in% c(0,5,6),]<-4
```

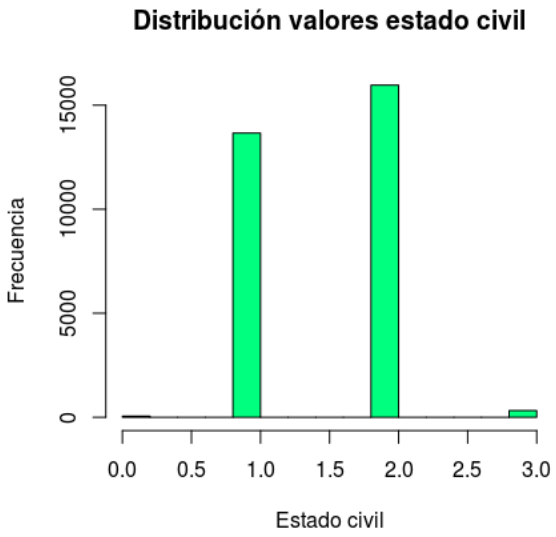
Análogamente, paraa el atributo MARRIAGE:

```
> table(default$MARRIAGE)

    0     1     2     3
   54 13659 15964   323
```

Gráficamente, podemos visualizarlo como un histograma:

```
> hist(default$MARRIAGE,
      main = "Distribucion valores estado civil",
      ylab = "Frecuencia",
      xlab = "Estado civil",
      col = "springgreen")
```



Los valores 0 tan sólo corresponden al 0.180% de las muestras:

```
> (sum(default$MARRIAGE==0)/sum(default$MARRIAGE!=0))*100
[1] 0.1803246
```

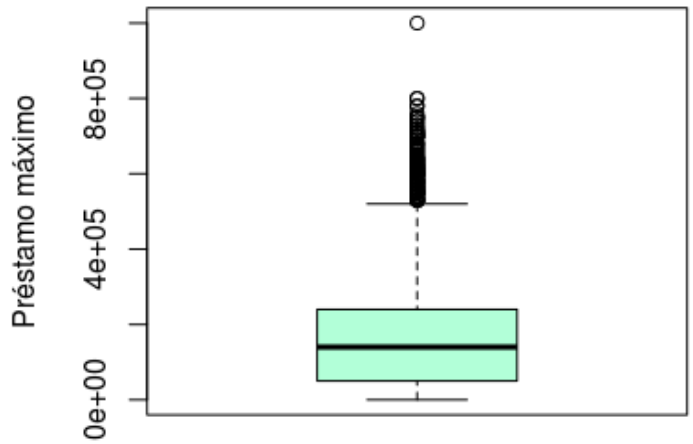
Por lo que decidimos eliminar las entradas correspondientes:

```
> default <- default[which(MARRIAGE!=0),]
```

3.2. Identificación y tratamiento de valores extremos

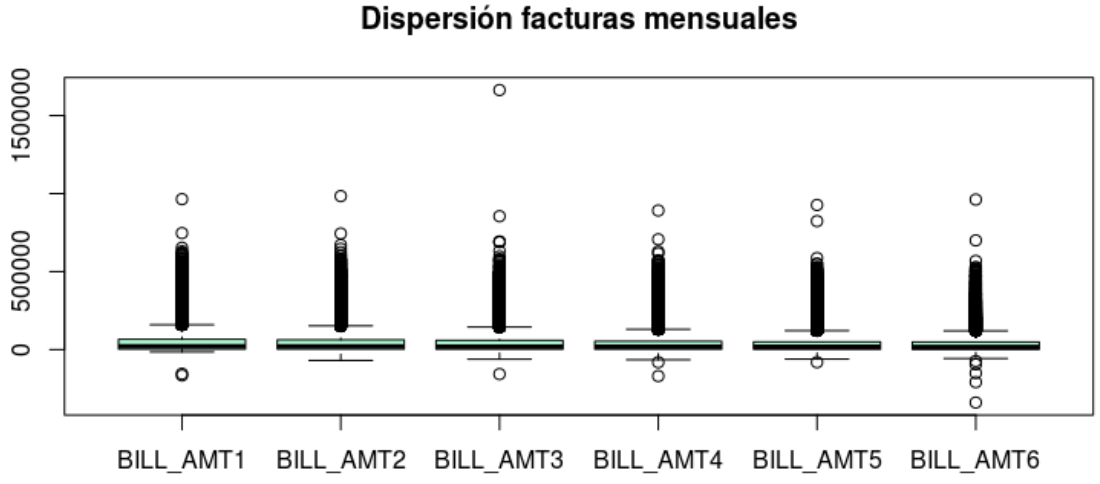
A fin de identificar valores extremos, graficaremos la distribución de las variables númericas no discretas. Comenzamos por el valor máximo de préstamo:

```
> boxplot.stats(default$LIMIT_BAL,
col=alpha("springgreen", 0.3))
```



El histórico de facturas mensuales:

```
> boxplot(default[,13:18],
col=alpha("springgreen", 0.3),
main="Dispersion facturas mensuales")
```



A pesar de que se observan valores que se desvían más de tres veces de la media, debido al contexto de banca en el que nos encontramos, deben mantenerse, ya que es perfectamente posible la adjudicación de un préstamo desproporcionado o la acumulación repentina de facturas.

4. Análisis de los datos

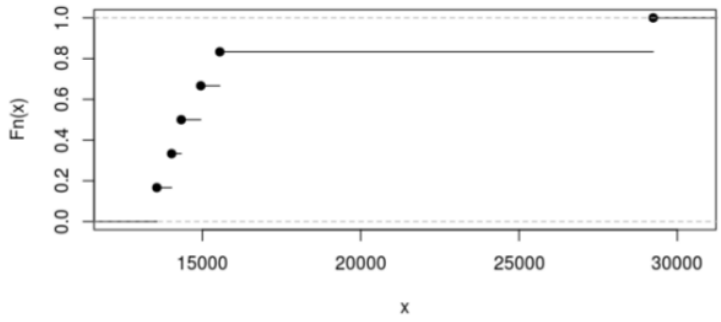
4.1 Selección de los grupos de datos que se quieren analizar/comparar

Nos interesa diferenciar entre morosos y no morosos, por lo que creamos dos datasets, default_true y default_false, a partir del original y en función de dicha característica:

```
> default_true <- subset(default, DEFAULT_RES == "1")
> default_false <- subset(default, DEFAULT_RES == "0")
```

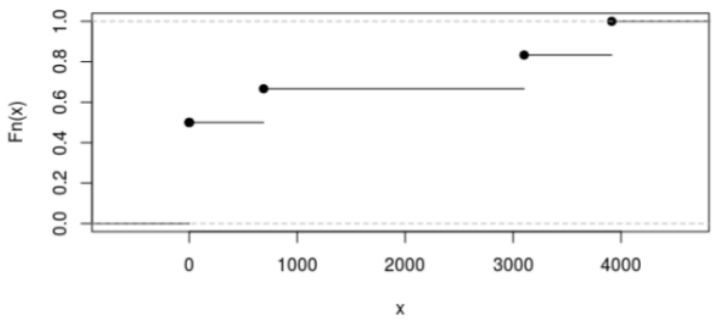
Ahora pasamos a explorar sus atributos para intentar explicar qué caracteriza el comportamiento de ambos grupos. Seleccionamos una muestra del grupo general, conteniendo los seis meses de facturas y otra conteniendo los seis meses de pagos. Además, observamos que una gráfica de tipo función cumulativa de probabilidad podría ser interesante para entender el patrón de pagos/deudas a lo largo de los meses.
Veamos cómo se comportan utilizando dos muestras de cada subgrupo y utilizando la función ecdf(). Para los no morosos:

```
> ecdf(default_false[1,c(13:18)])
Empirical CDF
Call: ecdf(default_false[1, c(13:18)])
x[1:6] = 13559, 14027, 14331, ..., 15549, 29239
> plot(ecdf(default_false[1,c(13:18)]))
```



Y para los morosos:

```
> ecdf(default_true[1,c(13:18)])
Empirical CDF
Call: ecdf(default_true[1, c(13:18)])
x[1:6] = 0, 0, 0, ..., 3102, 3913
```



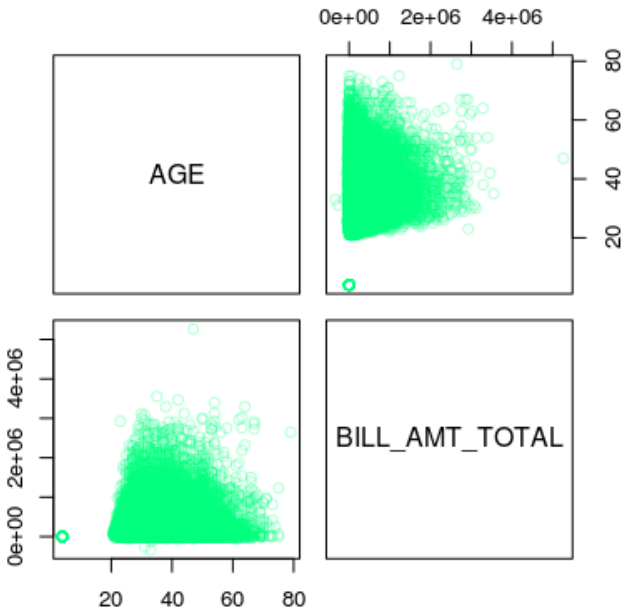
Observamos que la función densidad de probabilidad de los clientes morosos es mucho más plana, indicando el acumulo de deuda constante, sin la realización de pagos relevantes.

Nos interesaría saber, para cada cliente, el valor total de facturas acumuladas en los seis meses de datos que comprende el dataset. Para ello, sumaremos los valores de los campos BILL_AMT# con # entre 1 y 6, añadiendo un nuevo campo con el resultado de la operación, ATM_SUM al data frame original. Definimos la siguiente transformación sobre 6 columnas de default:

```
> default <- transform(default,
AMT_SUM=rowSums(default[,c(13:18)]))
```

Utilizamos ahora una matriz de scattering, implementada en R mediante la función "pair". En este caso, pretendemos analizar la influencia de los atributos de edad y suma de pagos totales:

```
> pairs(default[,c(6,26)], col=alpha("springgreen", 0.3))
```



Podemos extraer varias conclusiones, como que a partir de los 60 la cantidad y la cuantía de los pagos desciende enormemente.
La normalización de valores basada b la desviación estándar genera valores de entre 0 y 1 respetando la distribución de los valores iniciales. El resultado es un conjunto de media 0 y variancia 1 y resulta especialmente adecuada para aplicar algoritmos basados en el cálculo de distancias.

5. Código

El código en R utilizado a lo largo de este documento se encuentra disponible en la carpeta /src del repositorio de Github.

Bibliografía

Yeh, I. C., & Lien, C. H. (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. Expert Systems with Applications, 36(2), 2473-2480.