Tipología y ciclo de vida de los datos. Práctica 2

Baltasar Boix / Yago Ezcurra

13/5/2021

Contents

1 Titanic - Machine Learning from Disaster. Kaggle competition.

```
require(tidyverse)
require(lares)
require(GGally)
require(knitr)
require(kableExtra)
require(gridExtra)
require(DescTools)
```

2 Descripción del dataset. ¿Por qué es importante y qué pregunta/problema pretende responder?

2.1 Lectura y analisis previo del dataset.

Data Dictionary

Variable	Definition	Key
survival	Survival	0 = No, 1 = Yes
pclass	Ticket class	1 = 1st, 2 = 2nd, 3 = 3rd
sex	Sex	
Age	Age in years	
sibsp	# of siblings / spouses aboard the Titanic	
parch	# of parents / children aboard the Titanic	
ticket	Ticket number	
fare	Passenger fare	
cabin	Cabin number	
embarked	Port of Embarkation	C = Cherbourg, Q = Queenstown, S = Southampton

Variable Notes

pclass: A proxy for socio-economic status (SES)
1st = Upper
2nd = Middle
3rd = Lower

age: Age is fractional if less than 1. If the age is estimated, is it in the form of xx.5

sibsp: The dataset defines family relations in this way...
Sibling = brother, sister, stepbrother, stepsister
Spouse = husband, wife (mistresses and fiancés were ignored)

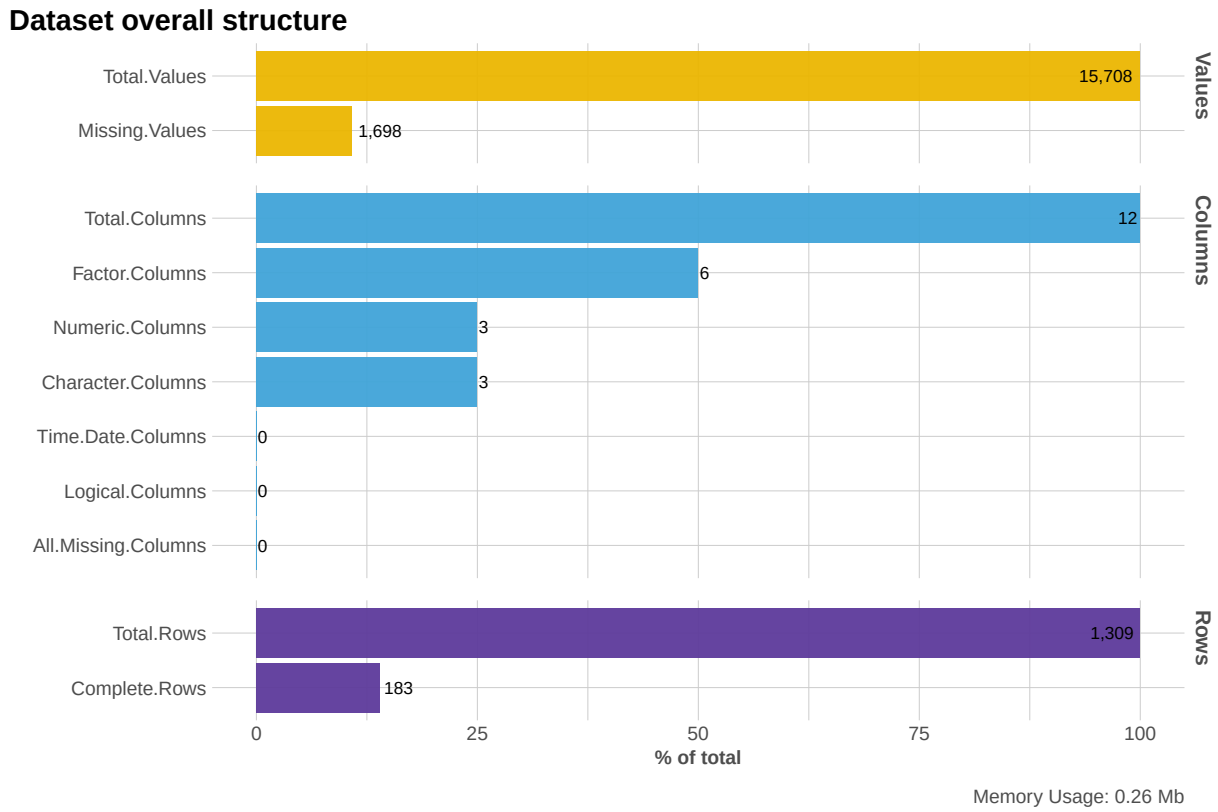
parch: The dataset defines family relations in this way...
Parent = mother, father
Child = daughter, son, stepdaughter, stepson
Some children travelled only with a nanny, therefore parch=0 for them.

Figure 1: Descripción del dataset obtenido en Kaggle

```r
df <- read_csv('../data/train.csv')
df_t <- read_csv('../data/test.csv')

df_t$Survived <- NA
df <- bind_rows(df, df_t)

df$Survived <- factor(df$Survived)
df$Pclass <- factor(df$Pclass)
df$Sex <- factor(df$Sex)
df$SibSp <- factor(df$SibSp)
df$Parch <- factor(df$Parch)
df$Embarked <- factor(df$Embarked)

summary(df)
```

```
##   PassengerId   Survived   Pclass      Name              Sex
##  Min.   :   1   0   :549   1:323   Length:1309        female:466
##  1st Qu.: 328   1   :342   2:277   Class :character   male  :843
##  Median : 655   NA's:418   3:709   Mode  :character
##  Mean   : 655
##  3rd Qu.: 982
##  Max.   :1309
##
##       Age         SibSp       Parch         Ticket               Fare
##  Min.   : 0.17   0:891   0      :1002   Length:1309        Min.   :  0.000
##  1st Qu.:21.00   1:319   1      : 170   Class :character   1st Qu.:  7.896
##  Median :28.00   2: 42   2      : 113   Mode  :character   Median : 14.454
##  Mean   :29.88   3: 20   3      :   8                      Mean   : 33.295
##  3rd Qu.:39.00   4: 22   4      :   6                      3rd Qu.: 31.275
##  Max.   :80.00   5:  6   5      :   6                      Max.   :512.329
##  NA's   :263     8:  9   (Other):   4                      NA's   :1
##     Cabin           Embarked
##  Length:1309        C   :270
##  Class :character   Q   :123
##  Mode  :character   S   :914
##                     NA's:  2
##
##
##
```

```r
df_str(df)
```



Creamos la variable dicotómica `Child` para diferenciar los niños de los adultos (>12años).

Creamos la variable `n_ticket` con el número de peraonas que viajan con el mismo ticket.

Separamos del `Name` el título (`title_name`) y el primer apellido (`first_name`).

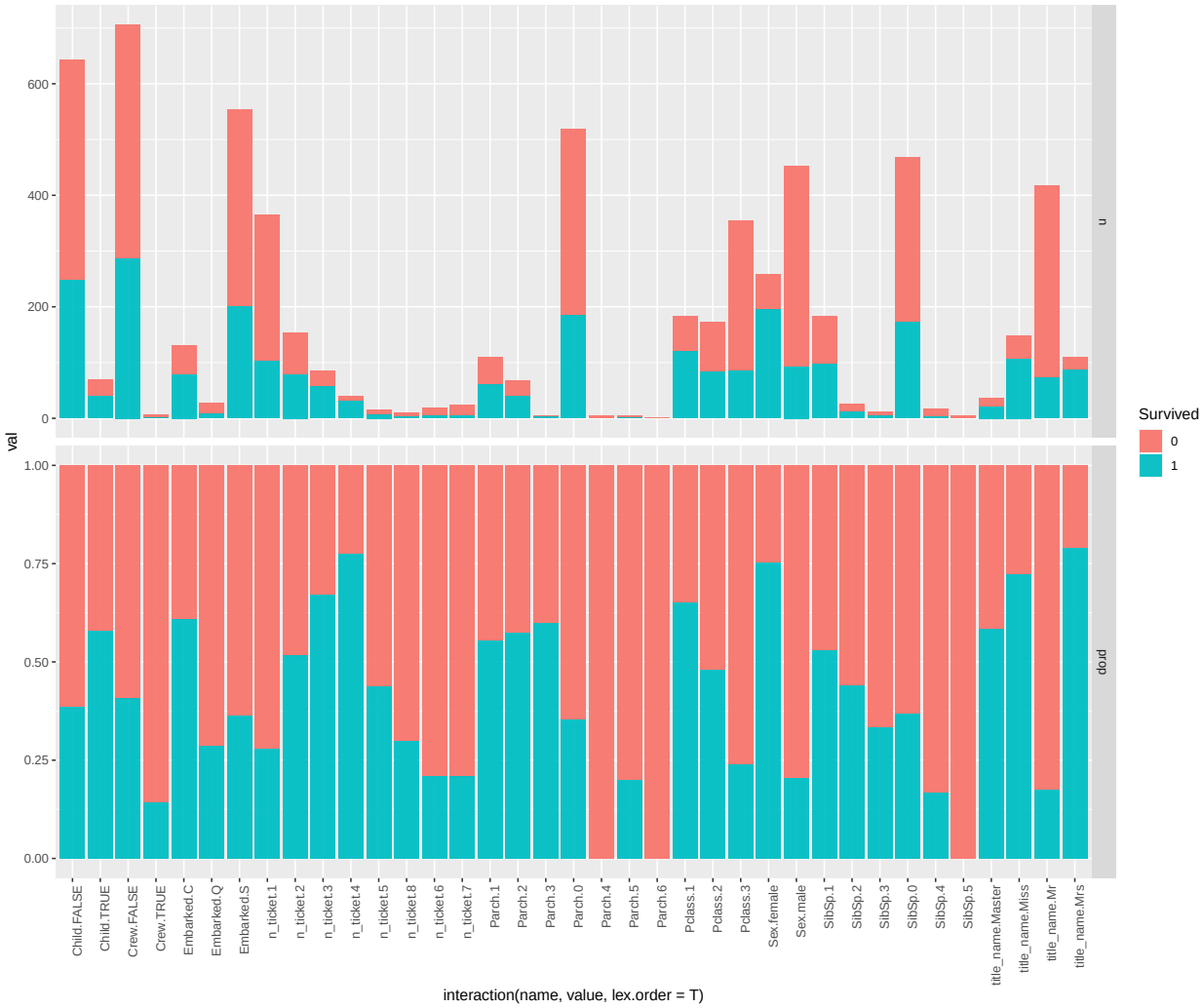
Simplificamos `title_name` en cuatro niveles.

Creamos la variable `Crew`. TRUE para la personas con Fare==0 que consideramos que viajan como tripulación.

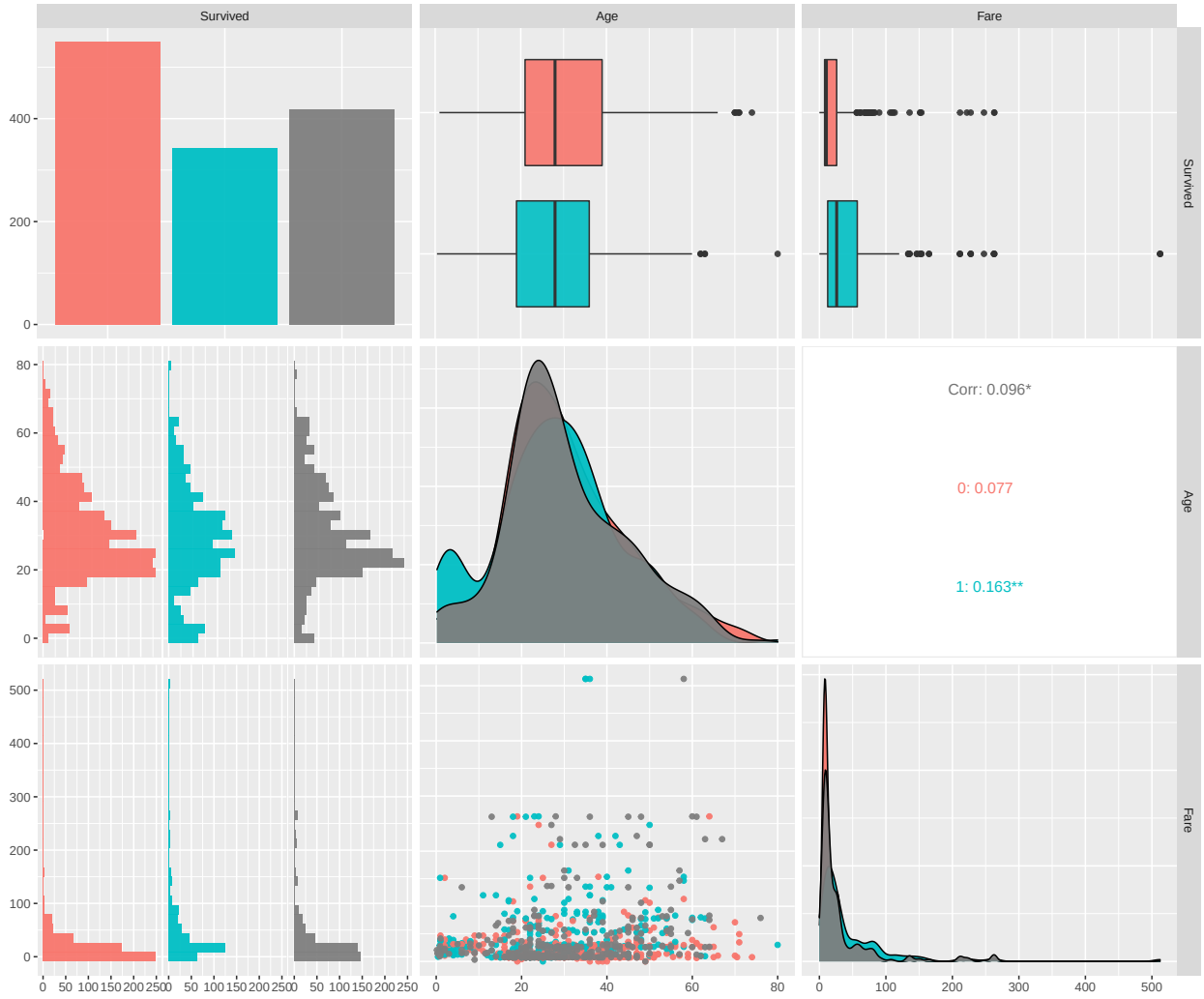
3 Integración y selección de los datos de interés a analizar.

```
df <- df %>%
  mutate(Child=factor(Age<=12))

df <- left_join(df, df %>%
                  group_by(Ticket) %>%
                  summarize(n_ticket=n())) %>%
  mutate(n_ticket=factor(n_ticket))

df <- df %>%
  separate(Name, c('first_name', 'rest_name'), sep=', ', remove=F) %>%
  separate(rest_name, c('title_name', 'rest_name'), sep='\\.') %>%
  select(-rest_name)

df$title_name[df$title_name %in% c('Capt', 'Col', 'Don', 'Dr', 'Jonkheer',
                                   'Major', 'Rev', 'Sir')] <- 'Mr'
df$title_name[df$title_name %in% c('Lady', 'Mme','the Countess', 'Dona')] <- 'Mrs'
df$title_name[df$title_name %in% c('Mlle', 'Ms')] <- 'Miss'
df$title_name <- factor(df$title_name)

df <- df %>%
```

```
  mutate(Crew=factor(if_else(Fare==0, TRUE, FALSE)))


df %>%
  select(where(is.factor)) %>%
  na.omit() %>%
  pivot_longer(-Survived) %>%
  group_by(name, value, Survived) %>%
  summarize(n=n()) %>%
  mutate(prop=prop.table(n)) %>%
  pivot_longer(c(n,prop), names_to='tipo', values_to='val') %>%
  ggplot(aes(x=interaction(name,value, lex.order = T), y=val, fill=Survived)) +
    geom_bar(stat='identity', position='stack') +
    facet_grid(tipo ~ ., scale='free_y') +
    theme(axis.text.x = element_text(angle = 90, vjust = 1, hjust=1))
```




```
df %>%
  select(Survived, where(is.numeric), -PassengerId) %>%
  ggpairs(aes(color=Survived))
```



4 Limpieza de los datos.

4.1 ¿Los datos contienen ceros o elementos vacíos? ¿Cómo gestionarías cada uno de estos casos?

Sustituimos los NA's de la variable Child con los siguientes criterios:

Asumimos que los viajeros con ticket unipersonal con SibSp==0 (sin hermanos o esposa a bordo) y Parch==0 (sin hermanos o padres a bordo) no son niños.

Asumimos que las personas con title_name=='Mrs' (mujeres casadas) no son niños.

Asumimos que las personas con title_name=='Master' son niños.

Asumimos que las personas con SibSp>0 y Parch>0 son niños.

```
df <- df %>%
  mutate(Child = if_else(SibSp == 0 & Parch == 0 & is.na(Child) & n_ticket == 1,
                         FALSE, as.logical(Child) )) %>%
  mutate(Child = if_else(title_name == 'Master' & is.na(Child),
                         TRUE, as.logical(Child))) %>%
  mutate(Child = if_else(title_name == 'Mrs' & is.na(Child),
                         FALSE, as.logical(Child))) %>%
  mutate(Child = if_else(SibSp != '0' & Parch != '0' & is.na(Child),
```

```
                        TRUE, as.logical(Child))) %>%
  mutate(Child = if_else(is.na(Child), FALSE, as.logical(Child))) %>%
  mutate(Child=factor(Child))



prop.table(table(df$Survived, df$Child, dnn = c('Survived', 'Child')), margin = 2)
```

```
##          Child
## Survived     FALSE      TRUE
##        0 0.6290323 0.4941176
##        1 0.3709677 0.5058824
```

4.2 Identificación y tratamiento de valores extremos.

5 Análisis de los datos.

5.1 Selección de los grupos de datos que se quieren analizar/comparar (planificación de los análisis a aplicar).

5.2 Comprobación de la normalidad y homogeneidad de la varianza.

5.3 Aplicación de pruebas estadísticas para comparar los grupos de datos. En función de los datos y el objetivo del estudio, aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc. Aplicar al menos tres métodos de análisis diferentes.

Para determinar la asociación de la variable `Survived` con el resto de variables categóricas utilizamos el test χ^2 y el coeficiente CramerV.

Para determinar la asociación de la variable `Survived` con las variables numéricas utilizaremos el t.test (H_0: la media del subset con `Survived==1` es igual a la media con `Survived==0`).

Ajustamos un modelo GLM y minimizamos el Aic con la función `step` (búsqueda de las variables más significativas).

Los dos primeros métodos identifican la asociación de cada variable individualmente. GLM tiene en cuenta la colinearidad de las variables.

Las más significativas son: `Pclass`, `title_name`, `Age` y `n_ticket`.

```
df %>%
  select(Survived, where(is.factor)) %>%
  filter(!is.na(Survived)) %>%
  pivot_longer(-Survived) %>%
  group_by(name) %>%
  summarize(cramerv=CramerV(x=Survived, y=factor(value)),
            chisq.pvalue=chisq.test(Survived, factor(value))$p.value) %>%
  mutate(`signif 95%`=chisq.pvalue < 0.05) %>%
  arrange(chisq.pvalue) %>%
  kable(format='latex', digits=4, caption='CramerV y chisq.test') %>%
  kable_styling(full_width = F, latex_options = "HOLD_position")
```

Table 1: CramerV y chisq.test

name	cramerv	chisq.pvalue	signif 95%
title_name	0.5708	0.0000	TRUE
Sex	0.5434	0.0000	TRUE
Pclass	0.3398	0.0000	TRUE
n_ticket	0.3388	0.0000	TRUE
SibSp	0.2045	0.0000	TRUE
Embarked	0.1726	0.0000	TRUE
Parch	0.1770	0.0001	TRUE
Child	0.0815	0.0206	TRUE
Crew	0.0853	0.0226	TRUE

```
df %>%
  select(Survived, where(is.numeric), -PassengerId) %>%
  filter(!is.na(Survived)) %>%
  pivot_longer(-Survived) %>%
  group_by(name, Survived) %>%
  summarize(value_list = list(value)) %>%
  pivot_wider(names_from=Survived, values_from=value_list) %>%
  mutate(vartest.pval=var.test(unlist(`0`), unlist(`1`))$p.val) %>%
  mutate(ttest.pval=t.test(unlist(`0`), unlist(`1`),
                           var.equal=vartest.pval>0.05)$p.val) %>%
  mutate(`signif 95%`=ttest.pval < 0.05) %>%
  select(-`0`, -`1`)  %>%
  kable(format='latex', digits=4, caption='t.test') %>%
  kable_styling(full_width = F, latex_options = "HOLD_position")
```

Table 2: t.test

name	vartest.pval	ttest.pval	signif 95%
Age	0.317	0.0391	TRUE
Fare	0.000	0.0000	TRUE

```
# GLM
df0 <- df %>%
  select(-PassengerId, -Name, -first_name, -Ticket, -Cabin) %>%
  na.omit()

fit <- glm(data=df0, Survived ~ ., family=binomial(link = "logit"))

step_fit <- step(fit, direction='both', trace=0)

summary(step_fit)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + title_name + Age + SibSp +
##     Fare + n_ticket, family = binomial(link = "logit"), data = df0)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4976  -0.4888  -0.3271   0.5291   2.4838
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      5.009233   0.853296   5.870 4.35e-09 ***
## Pclass2         -1.488484   0.384893  -3.867  0.00011 ***
## Pclass3         -2.527913   0.412596  -6.127 8.96e-10 ***
## title_nameMiss  -1.133159   0.720060  -1.574  0.11556
```

```
## title_nameMr    -4.069607   0.754773  -5.392 6.97e-08 ***
## title_nameMrs   -0.307542   0.777244  -0.396  0.69234
## Age             -0.033613   0.010319  -3.258  0.00112 **
## SibSp1          -0.203700   0.310747  -0.656  0.51214
## SibSp2          -0.283934   0.687729  -0.413  0.67971
## SibSp3          -0.048285   0.917546  -0.053  0.95803
## SibSp4          -1.023801   1.135404  -0.902  0.36721
## SibSp5         -19.614635 552.948800  -0.035  0.97170
## Fare             0.007836   0.004780   1.639  0.10113
## n_ticket2       -0.572295   0.347432  -1.647  0.09951 .
## n_ticket3       -0.428813   0.452726  -0.947  0.34355
## n_ticket4        0.376895   0.682009   0.553  0.58052
## n_ticket5       -2.480800   0.809389  -3.065  0.00218 **
## n_ticket6       -4.699915   1.107800  -4.243 2.21e-05 ***
## n_ticket7       -2.425401   1.065202  -2.277  0.02279 *
## n_ticket8        2.181463   1.041160   2.095  0.03615 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 960.90  on 711  degrees of freedom
## Residual deviance: 552.66  on 692  degrees of freedom
## AIC: 592.66
##
## Number of Fisher Scoring iterations: 14
```

6 Representación de los resultados a partir de tablas y gráficas.

7 Resolución del problema. A partir de los resultados obtenidos, ¿cuáles son las conclusiones? ¿Los resultados permiten responder al problema?