PRA2

Pablo Morante

4/6/2021

Competencias

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

● 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.

● Capacidad para aplicar las técnicas específicas de tratamiento de datos **(integración, transformación, limpieza y validación) para su posterior análisis.**

Objetivos

Los objetivos concretos de esta práctica son:

● Aprender a aplicar los conocimientos adquiridos y su capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios o multidisciplinares.

● Saber identificar los datos relevantes y los tratamientos necesarios (integración, limpieza y validación) para llevar a cabo un proyecto analítico.

● Aprender a analizar los datos adecuadamente para abordar la información contenida en los datos.

● Identificar la mejor representación de los resultados para aportar conclusiones sobre el problema planteado en el proceso analítico.

● Actuar con los principios éticos y legales relacionados con la manipulación de datos en Tipología y ciclo de vida de los datos Práctica 2 pág 2 función del ámbito de aplicación.

● Desarrollar las habilidades de aprendizaje que les permitan continuar estudiando de un modo que tendrá que ser en gran medida autodirigido o autónomo.

● Desarrollar la capacidad de búsqueda, gestión y uso de información y recursos en el ámbito de la ciencia de datos.

Resolución de la práctica

Descripción del dataset

El dataset contiene datos sobre los pasajeros que iban a bordo del Titanic. El objetivo de la creacion de esta base de datos esta suscritos a la aplicación de algotitmos de Machine Learning para intentar predecir que pasajeros sobrevivieron o no en base a sus características. Aunque el dataset ha sido creado con esa intención, en esta práctica lo utilizaremos para los objetivos mencionados en el apartado anterior, y si se considera necesario se aplicara algun algoritmo.

Los archivos descargados contienen 3 datasets, uno para entrenar el modelo de ML y otro para testearlo y otro que contiene la solucion correcta para la variable regresora (Survived) en la muestra de TEST. /*En esta practica, trataremos de integrar o fusionar estos tres para tener todos los datos en un mismo dataset.*/

Las variables contenidas en estos datasets son:

● PassengerId: Identificador del pasajero.

● Survival: Variable que indica si el pasajero sobrevivió o no. 0 = No, 1= Si.

● pclass: Variable que indica la "clase" del pasajero. 1= 1º clase, 2=2º clase, 3= 3º clase

● Name: Nombre y apellidos del pasajero

● sex: Variable que indica el sexo del pasajero

● Age: Variable que indica la edad en años del pasajero

● sibsp: Número de hermanos, hermanas, hermanastros o hermanastras en el barco

● parch: Número de padres e hijos en el barco

● ticket: Número de ticket

● fare: Precio pagado por el billete.

● cabin: Número de cabina/camarote asignado al pasajero.

● embarked: Puerto de embarcación del pasajero. C = Cherbourg, Q = Queenstown, S = Southampton.

La pregunta a la que pretende responder el dataset es si hubo caracteristicas que hicieran que la gente sobreviviera o no. Es decir, por ejemplo, mujeres y niños sobrevivieron mas que hombres? Se priorizó a gente de clase alta por delante de gente de clase baja?

Integración y selección de los datos de interés

Primero procederemos a importar los datos de los CSV

```
train <-
read.csv("C:/Users/Pablo/Desktop/Master/Tipologia/PRA2/train.csv")
```

```
#head(train)
test <- read.csv("C:/Users/Pablo/Desktop/Master/Tipologia/PRA2/test.csv")
#head(test)
result_test <-
read.csv("C:/Users/Pablo/Desktop/Master/Tipologia/PRA2/gender_submission.
csv")
head(result_test)
```

```
##   PassengerId Survived
## 1         892        0
## 2         893        1
## 3         894        0
## 4         895        0
## 5         896        1
## 6         897        0
```

Ahora uniremos todos los datos en un union dataset para proceder a la limpieza de datos. Si se necesita, mas tarde se volveran a separar en muestras de entrenamiento y muestras de test.

```
titanic <- rbind(train,merge(test, result_test))
summary(titanic)
```

```
##   PassengerId      Survived          Pclass          Name
##  Min.   :   1   Min.   :0.0000   Min.   :1.000   Length:1309
##  1st Qu.: 328   1st Qu.:0.0000   1st Qu.:2.000   Class :character
##  Median : 655   Median :0.0000   Median :3.000   Mode  :character
##  Mean   : 655   Mean   :0.3774   Mean   :2.295
##  3rd Qu.: 982   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :1309   Max.   :1.0000   Max.   :3.000
##
##      Sex                 Age            SibSp            Parch
##  Length:1309        Min.   : 0.17   Min.   :0.0000   Min.   :0.000
##  Class :character   1st Qu.:21.00   1st Qu.:0.0000   1st Qu.:0.000
##  Mode  :character   Median :28.00   Median :0.0000   Median :0.000
##                     Mean   :29.88   Mean   :0.4989   Mean   :0.385
##                     3rd Qu.:39.00   3rd Qu.:1.0000   3rd Qu.:0.000
##                     Max.   :80.00   Max.   :8.0000   Max.   :9.000
##                     NA's   :263
##     Ticket              Fare            Cabin              Embarked
##  Length:1309        Min.   :  0.000   Length:1309        Length:1309
##  Class :character   1st Qu.:  7.896   Class :character   Class
:character
##  Mode  :character   Median : 14.454   Mode  :character   Mode
:character
##                     Mean   : 33.295
##                     3rd Qu.: 31.275
##                     Max.   :512.329
##                     NA's   :1
```

```
#nrow(titanic)  #Filas
#length(titanic) # Columnas
```

Vemos que el dataset creado contiene 1309 observaciones con 12 variables. Utilizando la funcion summary vemos de que tipo son las variables asi como unos descriptivos basicos que pueden ser de gran ayuda, como el numero de NA's.

Limpieza de datos

Missings

Hemos visto en el apartado anterior que algunas variables contienen datos nulos. Vamos a comprobarlo mejor.

```
sapply(titanic,function(x) sum(is.na(x)))
```

```
## PassengerId    Survived      Pclass        Name         Sex
Age
##           0           0           0           0           0
263
##       SibSp       Parch      Ticket        Fare       Cabin
Embarked
##           0           0           0           1           0
0
```

Vemos que hay muchos Na para la variable de edad, y un unico NA para la variable Fare. Tambien observamos que en las variables de tipo caracter, al no contener un NA y tener un espacio en blanco, no detecta bien los NA's. Ante esto, decidimos eliminar la variable Cabin ya que contiene mucho valor nulo y no aporta información relevante al estudio. Eliminaremos tambien las variables de Pasajero IDn Name y ticket.

Sobre los NA's deberemos decidir que hacemos. Imputar valores manualmente no es una opcion en la variable Age ya que tenemos 263 missing aunque si podria serlo en la variable FAre. Para la variable Age seria aconsejable sustituir los NA's por valores que proviniesen de una regresion realizada, la media o mediana de la edad, o aplicar KNN para conseguir un valor de los individuos que tengan caracteristicas similares.

Despues de razonarlo decidimos aplicar la mediana para sustituir los NA's en Age y Fare. Para Embarked decidimos introducir el puerto del que mas pasajeros proceden.

```
#Eliminamos las variables mencionadas
#titanic = titanic[,c(-4,-9,-11)]

#Cambiamos NA's por mediana de la variable Fare
titanic$Fare[which(is.na(titanic$Fare))] = median(titanic$Fare, na.rm =
T)
#Cambiamos NA's por mediana de la variable AGE
titanic$Age[which(is.na(titanic$Age))] =median(titanic$Age, na.rm = T)
#Cambiamos NA's por mediana de la variable Embarked
table(titanic$Embarked)
```

```
##
##       C   Q   S
##   2 270 123 914

#Vemos que la categoria mas abundante es S, asi que sustituiremos los
missing por este puerto
titanic$Embarked[which(titanic$Embarked=='')] = 'S'
```

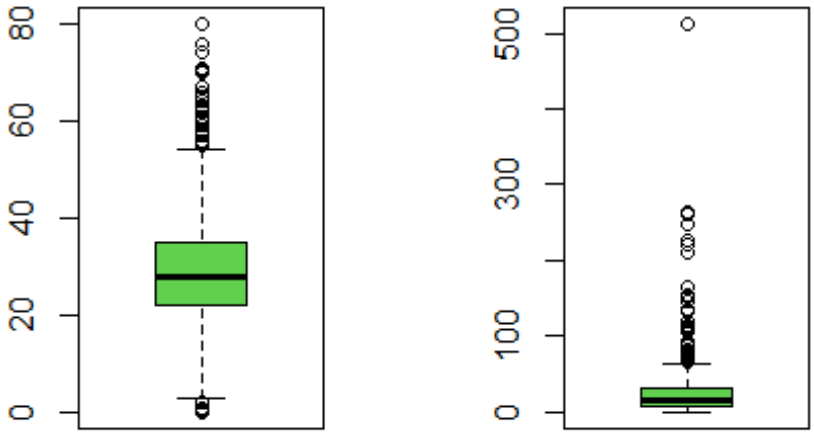
Como ultimo punto en este apartado convertiremos algunas variables. Las variables Survived y Pclass son realmente de tipo factor.

```
titanic$Survived = as.factor(titanic$Survived)
titanic$Pclass = as.factor(titanic$Pclass)
```

Valores Extremos

Las unicas variables puramente numericas que pueden proporcionarnos algun tipo de outlier son las variables Age y Fare. Usamos boxplot para ver sus valores extremos y corregirlos o excluirlos si es necesario aunque a priori no esperamos encontrar muchos.

```
par(mfrow=c(1,2))
boxplot(titanic$Age, col = 3)
boxplot(titanic$Fare, col=3)
```



Tras observar los missings, no decidimos excluir ninguno ya que nos parecen valores viables en ambas variables.

Análisis de datos

En este punto de la práctica, nos plantemos que analisis estadistico hacer. Realizaremos un Random Forest para intentar predecir la variable Survival. Adicionalmente realizaremos tambien una regresión logística. Evitaremos analizar la normalidad de los datos y la homogeneidad de la varianza ya que en modelos de ML problemas de heterocedasticidad y de normalidad quedan subsanados ya que el algoritmo no es dependiente de estos supuestos.

Empezaremos seleccionando variables y dividiendo el dataset limpio en muestras de entrenamiento y de test.

```
train <- titanic[1:891,]
test <- titanic[892:1309,]
```

Ahora crearemos nuestro modelo de regresión logística en base a los datos de entrenamiento y lo aplicaremos a los datos de test para ver si predicen bien nuestra variable Survived. Para el modelo de entrenamiento utilizaremos las mismas filas que las que habian en el csv inicial de TRAIN, y para el test las mismas del csv TEST.

Seleccionaremos las variables Pclass, sex, Age, SibSp, Parch, Fare como explicativas y como especificativa queremos Survived.

```
library("caret")

## Warning: package 'caret' was built under R version 4.0.5

## Loading required package: lattice

## Loading required package: ggplot2

#Entrenamos el modelo
titanic_RL <- glm(Survived ~ Pclass + Sex + Age + SibSp + Parch + Fare,
data = train, family = binomial(link='logit'))
summary(titanic_RL)

##
## Call:
## glm(formula = Survived ~ Pclass + Sex + Age + SibSp + Parch +
##     Fare, family = binomial(link = "logit"), data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7024  -0.6063  -0.4231   0.6130   2.4059
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.810955   0.443359   8.596  < 2e-16 ***
## Pclass2     -1.011926   0.293357  -3.449 0.000562 ***
## Pclass3     -2.151933   0.290012  -7.420 1.17e-13 ***
## Sexmale     -2.758449   0.199048 -13.858  < 2e-16 ***
```

```
## Age         -0.039159   0.007837  -4.997 5.83e-07 ***
## SibSp       -0.348059   0.109056  -3.192 0.001415 **
## Parch       -0.108447   0.117480  -0.923 0.355947
## Fare         0.003018   0.002447   1.234 0.217367
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1186.66  on 890  degrees of freedom
## Residual deviance:  789.09  on 883  degrees of freedom
## AIC: 805.09
##
## Number of Fisher Scoring iterations: 5
```

```
#Predecimos valores
check <- predict(titanic_RL, test)

#Ajustamos los valores a 1 o 0
check_RL <- ifelse(check > 0.5,1,0)
```

Realizaremos la matriz de confusion para evaluar los resultados

```
confusionMatrix(as.factor(check_RL),test$Survived)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 264  27
##          1   2 125
##
##                Accuracy : 0.9306
##                  95% CI : (0.9019, 0.953)
##     No Information Rate : 0.6364
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.8446
##
##  Mcnemar's Test P-Value : 8.324e-06
##
##             Sensitivity : 0.9925
##             Specificity : 0.8224
##          Pos Pred Value : 0.9072
##          Neg Pred Value : 0.9843
##              Prevalence : 0.6364
##          Detection Rate : 0.6316
##    Detection Prevalence : 0.6962
##       Balanced Accuracy : 0.9074
##
```

```
##        'Positive' Class : 0
##
```

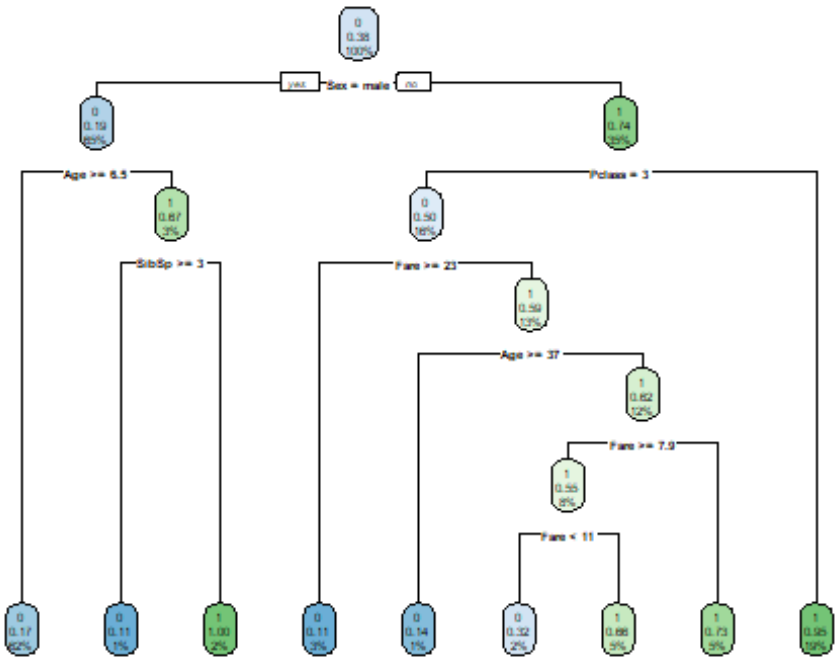
Vemos que tenemos un 93% de precision, es decir que el modelo ha acertado el 93% de los datos de test evaluados.

Ahora, realizaremos un random forest para ver que tal clasifica este algoritmo nuestra variable. El funcionamiento sera el mismo que el apartado anterior.

```
library('rpart')
library(rpart.plot)

## Warning: package 'rpart.plot' was built under R version 4.0.5

titanic_arbol <- rpart(Survived ~ Pclass + Sex + Age + SibSp + Parch +
Fare, data = train)
#summary(titanic_arbol)
pred_arbol <- predict(titanic_arbol, test, type='class')
rpart.plot(titanic_arbol)
```




```
#printcp(titanic_arbol)

confusionMatrix(pred_arbol,test$Survived)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
```

```
##          0 260  24
##          1   6 128
##
##                Accuracy : 0.9282
##                  95% CI : (0.8991, 0.9511)
##     No Information Rate : 0.6364
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.8409
##
##  Mcnemar's Test P-Value : 0.001911
##
##             Sensitivity : 0.9774
##             Specificity : 0.8421
##          Pos Pred Value : 0.9155
##          Neg Pred Value : 0.9552
##              Prevalence : 0.6364
##          Detection Rate : 0.6220
##    Detection Prevalence : 0.6794
##       Balanced Accuracy : 0.9098
##
##        'Positive' Class : 0
##
```

Con el uso de este algoritmo tenemos un 93% de precision.

Conclusiones

Como conclusiones, vemos que la regresion logistica ajusta mejor las personas que sobrevivieron o no. Como parametros significativos de este tenemos el sexo, laedad y la clase.

En el arbol de decision podemos ver que tambien tiene una precion alta 92,8% siendo ligeramente inferior a la de la regresion logistica. Ademas graficamente nos hacemos una idea de como discrimina el organitmo gracias al rpart.plot. Vemos que la variable que mas discrimina es el sexo, y vemos que el hecho de ser hombre aumenta la probabilidad de no sobrevivir.

Hemos visto en la matriz de confusión que ambos algoritmos clasifican bastante bien. Se ha valorado el hecho de crear variables dummies para la ejecucion del algoritmo pero finalmente se ha deshechado.

Previamente al análisis, hemos visto que la variable de edad tenía valores nulos y los hemos decidido sustituir por la mediana. Esta sustitución puede provocar que en caso de utilizar otro metodo (imputacion knn, eliminacion de observaciones) los resultados den diferentes.