
UNIVERSIDAD DE GRANADA

MASTER PROFESIONAL EN INGENIERÍA INFORMÁTICA

PRÁCTICA 1

Competición en Kaggle sobre Clasificación Binaria

Autor:

Manuel Jesús García Manday
(nickter@correo.ugr.es)

Master en Ingeniería Informática

20 de abril de 2017

Índice

1. Exploración de datos.	3
2. Procesamiento de datos.	5
3. Técnicas de clasificación.	15
4. Presentación y discusión de resultados.	15
5. Conclusiones y trabajo futuro.	21
6. Listado de soluciones.	21
7. Bibliografía.	22

1. Exploración de datos.

Para esta práctica disponemos de dos conjuntos de datos, uno para entrenamiento (**train**) y otro para pruebas (**test**). Ambos comparten la misma estructura en cuanto al número y tipos de variables, a excepción de la variable objetivo **Survived** que no se encuentra en el dataset de prueba y con una notable diferencia en el número de observaciones. Para conocer con un nivel mayor de detalle estos dataset vamos a pasar a cargar el dataset de entrenamiento para analizar la estructura y naturaleza de sus datos.

```

> str(train)
Classes 'tbl_df', 'tbl' and 'data.frame':      891 obs. of  12 variables:
 $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
 $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
 $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
 $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)" "Heikkinen, Miss. Laina" "Futrelle, Mrs. Jacques Heath (Lily May Peel)" ...
 $ Sex        : chr  "male" "female" "female" "female" ...
 $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
 $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
 $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
 $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
 $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
 $ Cabin      : chr  NA "C85" NA "C123" ...
 $ Embarked   : chr  "S" "C" "S" "S" ...

```

Figura 1: Estructura de los datos.

Como se puede apreciar en la imagen, principalmente son tres los tipos de datos que aparecen en este dataset, **int**, **num** y **chr**, aunque este último puede ser cambiado a **factor** para que se traten como categorías en lugar de cadenas de texto. El dataset de entrenamiento consta de 891 observaciones y 12 variables, siendo la variable **Survived** el campo objetivo como se ha mencionado antes.

Haciendo una primera previsualización de los datos en base a la variable objetivo (**Survived**) obtenemos la siguiente información:

```

> table(train$Survived)

 0    1 
549 342

```

Figura 2: Info. datos variable objetivo (I).

```

> prop.table(table(train$Survived))

 0          1 
0.6161616 0.3838384

```

Figura 3: Info. datos variable objetivo (II).

Según los datos obtenidos que se muestran en las imágenes podemos ver como el **61,61%** de las personas que viajaban en el Titanic murieron. Esta información arroja poca claridad sobre los datos, pero tomando la famosa frase de "las mujeres y los niños primero" podemos afinar un poco mas el resultado.

```

Console ~/Desktop/Master/SIGE/Practicas/Practica1/dataset/
> table(train$Sex)

female  male
   314    577

```

Figura 4: Info. datos variable **Sex** (I).

Se puede ver en la anterior imagen como la mayoría de los pasajeros eran de sexo masculino, por lo que podemos usar esta variable para conocer que género tiene mayor índice de supervivencia frente al otro.

```

Console ~/Desktop/Master/SIGE/Practicas/Practica1/dataset/
> prop.table(table(train$Sex, train$Survived),1)

           0           1
female 0.2579618 0.7420382
male   0.8110919 0.1889081

```

Figura 5: Info. datos variable **Sex** (II).

En esta última figura se muestra como el porcentaje de personas que no sobrevivieron es mucho mayor en el género masculino que en el femenino, superando el primero el **80 %** y obteniendo el segundo un **25 %**.

Es conveniente hacer un resumen sobre cada uno de los campos del dataset de entrenamiento para así poder ver que propiedades y valores nos arroja cada uno de ellos. De esta forma podemos identificar circunstancias en las variables como la cantidad de valores perdidos, el número de categorías, así como la ausencia de asignación a una categoría. Para ver esto ejecutamos el comando **summary(train)** junto con el nombre del dataset para que nos muestre la siguiente salida:

```

Console ~/Desktop/Master/SIGE/Practicas/Practica1/dataset/
> summary(train)
 PassengerId   Survived  Pclass                               Name
Min.   : 1.0   Min.   :0.0000 Min.   :1.000   Abbing, Mr. Anthony           : 1
1st Qu.:223.5 1st Qu.:0.0000 1st Qu.:2.000   Abbott, Mr. Rossmore Edward  : 1
Median :446.0 Median :0.0000 Median :3.000   Abbott, Mrs. Stanton (Rosa Hunt) : 1
Mean   :446.0 Mean   :0.3838 Mean   :2.309   Abelson, Mr. Samuel          : 1
3rd Qu.:668.5 3rd Qu.:1.0000 3rd Qu.:3.000   Abelson, Mrs. Samuel (Hannah Wizosky): 1
Max.   :891.0 Max.   :1.0000 Max.   :3.000   Adahl, Mr. Mauritz Nils Martin : 1
                                     (Other)                :885

```

Figura 6: Resumen de las variables del dataset de entrenamiento (I).

```

Sex      Age      SibSp      Parch      Ticket      Fare
female:314 Min.   : 0.42 Min.   :0.000 Min.   :0.0000 1601 : 7 Min.   : 0.00
male :577 1st Qu.:20.12 1st Qu.:0.000 1st Qu.:0.0000 347082 : 7 1st Qu.: 7.91
          Median :28.00 Median :0.000 Median :0.0000 CA. 2343: 7 Median :14.45
          Mean   :29.70 Mean   :0.523 Mean   :0.3816 3101295 : 6 Mean   :32.20
          3rd Qu.:38.00 3rd Qu.:1.000 3rd Qu.:0.0000 347088 : 6 3rd Qu.:31.00
          Max.   :80.00 Max.   :8.000 Max.   :6.0000 CA 2144 : 6 Max.   :512.33
          NA's   :177                                     (Other) :852

```

Figura 7: Resumen de las variables del dataset de entrenamiento (II).

```

      Cabin      Embarked
      :687      : 2
B96 B98      : 4      C:168
C23 C25 C27: 4      Q: 77
G6           : 4      S:644
C22 C26      : 3
D            : 3
(Other)      :186

```

Figura 8: Resumen de las variables del dataset de entrenamiento (III).

Podemos ver como las imágenes nos arroja mucha información acerca de las variables del dataset. Vemos como el campo **Age** tiene 177 valores perdidos, lo que resulta una cantidad considerable. También nos podemos dar cuenta de como hay dos pasajeros que no tienen valor asignado en el atributo **Embarked** y de como existen muchas categorías en atributos como **Cabin** y **Ticket**. Para las variables que son continuas nos muestra valores como la media, el valor máximo, el valor mínimo, la mediana, etc.

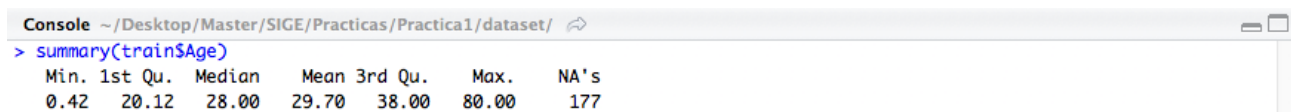
Esta información nos muestra una mayor claridad sobre las variables, lo que se convierte en un punto de comienzo para empezar a preprocesarlas como se verá en el siguiente apartado.

2. Procesamiento de datos.

Una vez que se ha realizado la exploración de los datos del dataset de entrenamiento en el apartado anterior, se ha podido comprobar que existen valores perdidos en algunas de sus variables como el campo **Age** y casos en los que no tienen valor asignado como las variables **Embarked**. Este tipo de circunstancia es muy común que se presente en un dataset debido a la complejidad y cantidad de observaciones que contiene, aunque existen diversos mecanismos que se suelen emplear para paliar estos tipos de problemas.

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
1	1	0	3	Braund, Mr. Owen Harris	male	22.00	1	0	A/5 21171
2	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38.00	1	0	PC 17599
3	3	1	3	Heikkinen, Miss. Laina	female	26.00	0	0	STON/O2. 3101282
4	4	1	1	Futelle, Mrs. Jacques Heath (Lily May Peel)	female	35.00	1	0	113803
5	5	0	3	Allen, Mr. William Henry	male	35.00	0	0	373450
6	6	0	3	Moran, Mr. James	male	NA	0	0	330877
7	7	0	1	McCarthy, Mr. Timothy J	male	54.00	0	0	17463
8	8	0	3	Palsson, Master. Gosta Leonard	male	2.00	3	1	349909
9	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.00	0	2	347742
10	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.00	1	0	237736
11	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.00	1	1	PP 9549
12	12	1	1	Bonnell, Miss. Elizabeth	female	58.00	0	0	113783
13	13	0	3	Saunderscock, Mr. William Henry	male	20.00	0	0	A/5. 2151
14	14	0	3	Andersson, Mr. Anders Johan	male	39.00	1	5	347082
15	15	0	3	Vestrom, Miss. Hulda Amanda Adolfina	female	14.00	0	0	350406
16	16	1	2	Hewlett, Mrs. (Mary D Kingcome)	female	55.00	0	0	248706
17	17	0	3	Rice, Master. Eugene	male	2.00	4	1	382652
18	18	1	2	Williams, Mr. Charles Eugene	male	NA	0	0	244373

Figura 9: Valores perdidos en la variable **Age** (I).

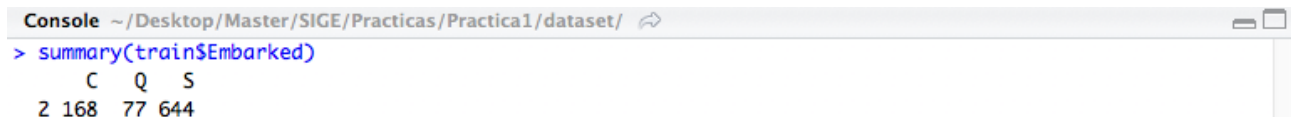


```

Console ~/Desktop/Master/SIGE/Practicas/Practica1/dataset/
> summary(train$Age)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's
  0.42  20.12   28.00   29.70   38.00   80.00   177

```

Figura 10: Valores perdidos en la variable **Age** (II).



```

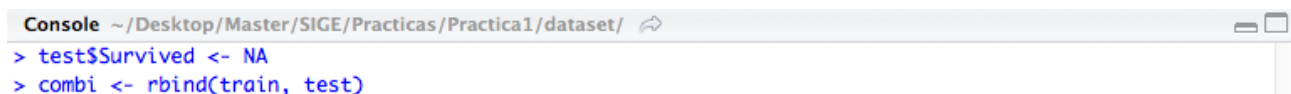
Console ~/Desktop/Master/SIGE/Practicas/Practica1/dataset/
> summary(train$Embarked)
  C    Q    S
2 168  77 644

```

Figura 11: Valores sin asignación en la variable **Embarked**.

Realizando un análisis mas detallado sobre dichas variables, podemos ver como son 177 observaciones las que tienen valor perdido en la variable **Age** como muestra la **Figura 7**, una cantidad elevada que podría acarrear problemas de precisión a la hora de realizar la predicción. Existen diferentes técnicas que se pueden aplicar para completar esos valores perdidos de la variable, para este caso vamos a utilizar la predicción empleando para ello un árbol de decisión.

Con los datasets de entrenamiento y prueba cargados, lo primero que vamos a hacer es añadirle al dataset de prueba el campo **Survived** para que de esta forma tenga el mismo número de variables que el dataset de entrenamiento y podamos unir ambos. Esta unión nos facilitará el trabajo a la hora de querer interaccionar con un dataset o con otro, ya que en este nuevo tenemos la fusión de los dos. Señalar que la variable **Survived** añadida al dataset de prueba será completada con valores perdidos como se muestra en los siguientes comandos de la imagen.



```

Console ~/Desktop/Master/SIGE/Practicas/Practica1/dataset/
> test$Survived <- NA
> combi <- rbind(train, test)

```

Figura 12: Unión de los dataset.

Antes de ponernos a completar los valores perdidos que hemos encontrado en algunas variables, necesitamos realizar una serie de ajustes previos que nos permitan luego predecir esos valores con una mayor exactitud. El primer ajuste que vamos a realizar viene relacionado con el atributo **Name**, del que vemos que podemos extraer una información adicional referida al título que la persona tenía adoptado (si era soltero, casado, con una buena situación económica, títulos nobiliarios, etc). Esta generación adicional de característica va a crear un nuevo atributo que nos puede aportar información relevante, ya que el título de una persona viene reflejado por su estatus económico, por el cual podemos obtener una primera intuición en la que a mayor nivel económico mas cerca de los botes salvavidas podían encontrarse debido a que esas zonas eran mas caras de adquirir.

Lo primero es convertir dicho atributo a cadena de texto para poder tratarlo y substraer del nombre la parte del título solamente. Una vez obtenido el título, agrupamos los que son de la misma índole para posteriormente convertirlo a **factor**.

```

Console ~/Desktop/Master/SIGE/Practicas/Practica1/dataset/
> combi$Name <- as.character(combi$Name)
> combi$Title <- sapply(combi$Name, FUN=function(x) {strsplit(x, split='[,.]')[[1]][2]})
> combi$Title <- sub(' ', '', combi$Title)
> combi$Title[combi$Title %in% c('Mme', 'Mlle')] <- 'Mlle'
> combi$Title[combi$Title %in% c('Capt', 'Don', 'Major', 'Sir')] <- 'Sir'
> combi$Title[combi$Title %in% c('Dona', 'Lady', 'the Countess', 'Jonkheer')] <- 'Lady'
> combi$Title <- factor(combi$Title)

```

Figura 13: Generación de característica adicional **Title** (I).

Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Title
male	22.00	1	0	A/5 21171	7.2500		S	Mr
female	38.00	1	0	PC 17599	71.2833	C85	C	Mrs
female	26.00	0	0	STON/O2. 3101282	7.9250		S	Miss
female	35.00	1	0	113803	53.1000	C123	S	Mrs
male	35.00	0	0	373450	8.0500		S	Mr
male	NA	0	0	330877	8.4583		Q	Mr
male	54.00	0	0	17463	51.8625	E46	S	Mr
male	2.00	3	1	349909	21.0750		S	Master
female	27.00	0	2	347742	11.1333		S	Mrs
female	14.00	1	0	237736	30.0708		C	Mrs
female	4.00	1	1	PP 9549	16.7000	G6	S	Miss
female	58.00	0	0	113783	26.5500	C103	S	Miss
male	20.00	0	0	A/5. 2151	8.0500		S	Mr
male	39.00	1	5	347082	31.2750		S	Mr

Figura 14: Generación de característica adicional **Title** (II).

Siguiendo en el mismo hilo para la generación de una nueva característica adicional, podemos observar que hay dos atributos que nos dan información sobre el número de miembros de la familia que viajaban con el pasajero. Generar un atributo que nos diga el tamaño total de la familia de cada pasajero puede ser útil ya que dentro del pánico creado en el momento del accidente una familia grande lo tendría mas complicado para reunirse todos los miembros frente a una más pequeña. Es por eso por lo que creamos este atributo adicional en base a los atributos **SibSp** y **Parch**.

```

Console ~/Desktop/Master/SIGE/Practicas/Practica1/dataset/
> combi$FamilySize <- combi$SibSp + combi$Parch + 1

```

Figura 15: Generación de característica adicional **FamilySize** (I).

Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Title	FamilySize
male	22.00	1	0	A/5 21171	7.2500		S	Mr	2
female	38.00	1	0	PC 17599	71.2833	C85	C	Mrs	2
female	26.00	0	0	STON/O2. 3101282	7.9250		S	Miss	1
female	35.00	1	0	113803	53.1000	C123	S	Mrs	2
male	35.00	0	0	373450	8.0500		S	Mr	1
male	NA	0	0	330877	8.4583		Q	Mr	1
male	54.00	0	0	17463	51.8625	E46	S	Mr	1
male	2.00	3	1	349909	21.0750		S	Master	5
female	27.00	0	2	347742	11.1333		S	Mrs	3
female	14.00	1	0	237736	30.0708		C	Mrs	2
female	4.00	1	1	PP 9549	16.7000	G6	S	Miss	3
female	58.00	0	0	113783	26.5500	C103	S	Miss	1
male	20.00	0	0	A/5. 2151	8.0500		S	Mr	1
male	39.00	1	5	347082	31.2750		S	Mr	7

Figura 16: Generación de característica adicional **FamilySize** (II).

Continuando con el atributo **Name** vemos que podemos sacarle aún mas partido. Vamos a generar un nuevo atributo que nos muestre el apellido de familia al que pertenece cada pasajero al que denominaremos **Surname**, siendo el comando muy similar al utilizado para la generación de la anterior variable.

```

Console ~/Desktop/Master/SIGE/Practicas/Practica1/dataset/
> combi$Surname <- sapply(combi$Name, FUN=function(x) {strsplit(x, split='[,.]')[[1]][1]})

```

Figura 17: Generación de característica adicional **Surname** (I).

Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Title	FamilySize	Surname
22.00	1	0	A/5 21171	7.2500		S	Mr	2	Braund
38.00	1	0	PC 17599	71.2833	C85	C	Mrs	2	Cumings
26.00	0	0	STON/O2. 3101282	7.9250		S	Miss	1	Heikkinen
35.00	1	0	113803	53.1000	C123	S	Mrs	2	Futrelle
35.00	0	0	373450	8.0500		S	Mr	1	Allen
NA	0	0	330877	8.4583		Q	Mr	1	Moran
54.00	0	0	17463	51.8625	E46	S	Mr	1	McCarthy
2.00	3	1	349909	21.0750		S	Master	5	Palsson
27.00	0	2	347742	11.1333		S	Mrs	3	Johnson
14.00	1	0	237736	30.0708		C	Mrs	2	Nasser
4.00	1	1	PP 9549	16.7000	G6	S	Miss	3	Sandstrom
58.00	0	0	113783	26.5500	C103	S	Miss	1	Bonnell
20.00	0	0	A/5. 2151	8.0500		S	Mr	1	Saunderscock
39.00	1	5	347082	31.2750		S	Mr	7	Andersson

Figura 18: Generación de característica adicional **Surname** (II).

Dándole una vuelta, este nuevo atributo no terminaría de clasificar bien a cada pasajero ya que es muy probable que entre tantas personas hubiese algún apellido común que se repitiese, por lo que no se puede averiguar a que familia pertenece compartiendo el mismo apellido. Para solucionar esta circunstancia vamos a definir un nuevo atributo llamado **FamilyID** en el que identificaremos a cada familia por la combinación de los atributos **Surname** y **FamilySize**. De esta forma es menos probable que dos pasajeros que compartan el mismo valor para la variable **Surname** tengan también el mismo número de miembros en la familia. Además de eso, las familias que tengan 2 o menos miembros serán identificadas como pequeñas (**Small**).

```

Console ~/Desktop/Master/SIGE/Practicas/Practica1/dataset/
> combi$FamilyID <- paste(as.character(combi$FamilySize), combi$Surname, sep="")
> combi$FamilyID[combi$FamilySize <= 2] <- 'Small'

```

Figura 19: Generación de característica adicional **FamilyID** (I).

SibSp	Parch	Ticket	Fare	Cabin	Embarked	Title	FamilySize	Surname	FamilyID
1	0	A/5 21171	7.2500		S	Mr	2	Braund	Small
1	0	PC 17599	71.2833	C85	C	Mrs	2	Cumings	Small
0	0	STON/O2. 3101282	7.9250		S	Miss	1	Heikkinen	Small
1	0	113803	53.1000	C123	S	Mrs	2	Futrelle	Small
0	0	373450	8.0500		S	Mr	1	Allen	Small
0	0	330877	8.4583		Q	Mr	1	Moran	Small
0	0	17463	51.8625	E46	S	Mr	1	McCarthy	Small
3	1	349909	21.0750		S	Master	5	Palsson	5Palsson
0	2	347742	11.1333		S	Mrs	3	Johnson	3Johnson
1	0	237736	30.0708		C	Mrs	2	Nasser	Small
1	1	PP 9549	16.7000	G6	S	Miss	3	Sandstrom	3Sandstrom
0	0	113783	26.5500	C103	S	Miss	1	Bonnell	Small
0	0	A/5. 2151	8.0500		S	Mr	1	Saunderscock	Small
1	5	347082	31.2750		S	Mr	7	Andersson	7Andersson

Figura 20: Generación de característica adicional **FamilyID** (II).

Analizando esta nueva variable podemos ver como se muestran familias de 1 o 2 miembros, algo que vamos a descartar ya que como hemos comentado antes, para considera una familia debe tener al menos tres miembros.

```

Console ~/Desktop/Master/SIGE/Practicas/Practica1/dataset/
> famIDs <- data.frame(table(combi$FamilyID))

```

Figura 21: Analizando atributo **FamilyID** (I).

Var1	Freq
11Sage	11
3Abbott	3
3Appleton	1
3Beckwith	2
3Boulos	3
3Bourke	3
3Brown	4
3Caldwell	3
3Christy	2
3Collyer	3
3Compton	3
3Cornell	1
3Coutts	3
3Crosby	3

Figura 22: Analizando atributo **FamilyID** (II).

```

Console ~/Desktop/Master/SIGE/Practicas/Practica1/dataset/
> famIDs <- famIDs[famIDs$Freq <= 2,]
> combi$FamilyID[combi$FamilyID %in% famIDs$Var1] <- 'Small'
> combi$FamilyID <- factor(combi$FamilyID)

```

Figura 23: Identificando familias pequeñas por **FamilyID** (I).

Ticket	Fare	Cabin	Embarked	Title	FamilySize	Surname	FamilyID
A/5 21171	7.2500		S	Mr	2	Braund	Small
PC 17599	71.2833	C85	C	Mrs	2	Cumings	Small
STON/O2. 3101282	7.9250		S	Miss	1	Heikkinen	Small
113803	53.1000	C123	S	Mrs	2	Futrelle	Small
373450	8.0500		S	Mr	1	Allen	Small
330877	8.4583		Q	Mr	1	Moran	Small
17463	51.8625	E46	S	Mr	1	McCarthy	Small
349909	21.0750		S	Master	5	Palsson	5Palsson
347742	11.1333		S	Mrs	3	Johnson	3Johnson
237736	30.0708		C	Mrs	2	Nasser	Small
PP 9549	16.7000	G6	S	Miss	3	Sandstrom	3Sandstrom
113783	26.5500	C103	S	Miss	1	Bonnell	Small
A/5. 2151	8.0500		S	Mr	1	Saunderscock	Small
347082	31.2750		S	Mr	7	Andersson	7Andersson

Figura 24: Identificando familias pequeñas por **FamilyID** (II).

Una vez realizados los ajustes necesarios en el dataset, vamos ahora a crear el árbol de decisión que nos ayudará a predecir los valores perdidos del atributo **Age** en el mismo. Para ello cargamos previamente la librería **rpart** y posteriormente seleccionamos el conjunto de las variables que crearán dicho árbol, **Pclass**, **Sex**, **Parch**, **Fare**, **Embarked**, **Title** y **FamilySize**. Con el árbol de decisión creado predecimos los valores perdidos de la variable **Age** que haya en el dataset.

```

Console ~/Desktop/Master/SIGE/Practicas/Practica1/dataset/
> summary(combi$Age)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's
  0.17  21.00   28.00   29.88  39.00   80.00   263

```

Figura 25: Analizando dataset.

```

Console ~/Desktop/Master/SIGE/Practicas/Practica1/dataset/
> Agefit <- rpart(Age ~ Pclass + Sex + SibSp + Parch + Fare + Embarked + Title + FamilySize,
+               data=combi[!is.na(combi$Age),], method="anova")
> combi$Age[is.na(combi$Age)] <- predict(Agefit, combi[is.na(combi$Age),])

```

Figura 26: Creando árbol de decisión y predicción de valores (I).

PassengerId	Survived	Pclass	Name	Sex	Age
1	0	3	Braund, Mr. Owen Harris	male	22.000000
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38.000000
3	1	3	Heikkinen, Miss. Laina	female	26.000000
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.000000
5	0	3	Allen, Mr. William Henry	male	35.000000
6	0	3	Moran, Mr. James	male	28.862881
7	0	1	McCarthy, Mr. Timothy J	male	54.000000
8	0	3	Palsson, Master. Gosta Leonard	male	2.000000
9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.000000
10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.000000
11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.000000
12	1	1	Bonnell, Miss. Elizabeth	female	58.000000
13	0	3	Saunderscock, Mr. William Henry	male	20.000000

Figura 27: Creando árbol de decisión y predicción de valores (II).

Como se muestran en las anteriores imágenes, hemos conseguido predecir el valor de la edad de todos los pasajeros que carecían de ella. Con esto limpiamos un poco mas el dataset a la vez que gana en calidad.

Continuando con dicha limpieza, se hizo mención antes de los espacios en blanco que había del atributo **Embarked** en algunas observaciones. Esta variable nos dice cual fue el puerto en el que embarcó cada pasajero del barco, siendo tres las posibles localizaciones como se muestra en la imagen.

```

Console ~/Desktop/Master/SIGE/Practicas/Practica1/dataset/
> summary(combi$Embarked)
  C   Q   S
2 270 123 914

```

Figura 28: Analizando variable **Embarked**.

Hemos visto que son dos los registros que están sin valor, y que la mayoría de ellos embarcaron desde Southampton (S), por lo que vamos a rellenar esas dos observaciones en blanco con esa misma localización, tomando como medida desde donde partieron la mayoría.

```

Console ~/Desktop/Master/SIGE/Practicas/Practica1/dataset/
> which(combi$Embarked == '')
[1] 62 830
> combi$Embarked[c(62,830)] = "S"
> combi$Embarked <- factor(combi$Embarked)

```

Figura 29: Completando espacios en blanco en la variable **Embarked**.

Otra de las variables que necesitamos limpiar de los valores predidos que presenta es la que nos dice que tarifa tenía cada pasajero (**Fare**). Al tratarse de una variable continua y de existir un único registro con valor

perdido, he decido tomar la mediana de los restantes valores para obtener el mismo.

```

Console ~/Desktop/Master/SIGE/Practicas/Practica1/dataset/
> summary(combi$Fare)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
 0.000  7.896  14.450  33.300  31.280  512.300     1

```

Figura 30: Analizando variable **Fare**.

```

Console ~/Desktop/Master/SIGE/Practicas/Practica1/dataset/
> which(is.na(combi$Fare))
[1] 1044
> combi$Fare[1044] <- median(combi$Fare, na.rm=TRUE)

```

Figura 31: Completando valores perdidos en la variable **Embarked**.

Analizando la nueva variable de categoría creada **FamilyID** con la que se identifica a cada pasajero dentro de una familia en función del apellido y número de miembros de la misma, he comprobado que el número total de categorías que obtenemos supera el número máximo de las que permite el algoritmo **Random Forest**, por lo que es necesario reducir el número de categorías para poder usar esta técnica si se ve oportuno, tomando ahora como familia pequeña aquella que tenga tres o menos miembros.

```

Console ~/Desktop/Master/SIGE/Practicas/Practica1/dataset/
> summary(combi$FamilyID)
 11Sage    3Abbott    3Boulos    3Bourke    3Brown    3Caldwell    3Collyer
   11         3         3         3         4         3         3
 3Compton  3Coutts    3Crosby    3Danbom    3Davies    3Dodge      3Drew
   3         3         3         3         5         3         3
 3Elias    3Goldsmith  3Hart     3Hickman   3Johnson   3Klasen    3Mallet
   3         3         3         3         3         3         3
 3McCoy    3Moubarek    3Nakid    3Navratil  3Peacock    3Peter     3Quick
   3         3         3         3         3         3         3
 3Rosblom  3Samaan    3Sandstrom  3Spedden   3Taussig    3Thayer    3Touma
   3         3         3         3         3         3         3
 3van Billiard  3Van Impe    3Wells     3Wick     3Widener    4Allison    4Baclini
   3         3         3         3         3         4         4
 4Becker    4Carter     4Dean     4Herman   4Johnston   4Laroche    4West
   4         4         4         4         4         4         4

```

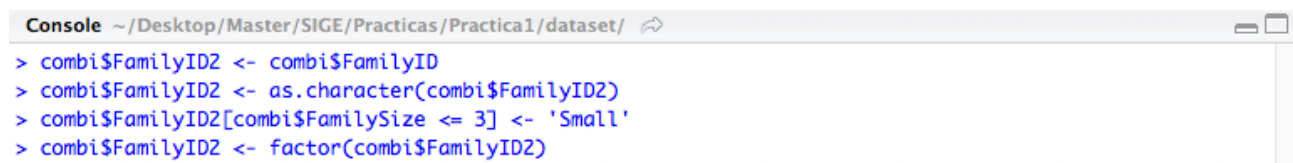
Figura 32: Analizando variable **FamilyID** (I).

```

 5Ford     5Lefebre    5Palsson    5Ryerson    6Fortune    6Panula     6Rice
   5         5         5         5         6         6         6
 6Skoog    7Andersson    7Asplund    8Goodwin    Small      1074
   6         9         7         8

```

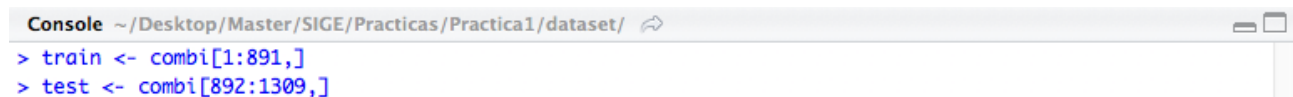
Figura 33: Analizando variable **FamilyID** (II).



```
Console ~/Desktop/Master/SIGE/Practicas/Practica1/dataset/
> combi$FamilyID2 <- combi$FamilyID
> combi$FamilyID2 <- as.character(combi$FamilyID2)
> combi$FamilyID2[combi$FamilySize <= 3] <- 'Small'
> combi$FamilyID2 <- factor(combi$FamilyID2)
```

Figura 34: Creando nueva variable **FamilyID2**.

Lo último que vamos a realizar es dividir el dataset **combi** en los dos que lo conforma, es decir, el de entrenamiento y el de prueba pero en este caso ambos con el mismo número de variables y con el mismo preprocesamiento realizado.



```
Console ~/Desktop/Master/SIGE/Practicas/Practica1/dataset/
> train <- combi[1:891,]
> test <- combi[892:1309,]
```

Figura 35: Cargando datasets de prueba y entrenamiento actualizados.

Todo lo desarrollado en este apartado es dedicado a la limpieza de los datos para conseguir de este modo un dataset de mayor calidad que no permita obtener una mejor precisión en la clasificación. Se ha detallado todas las técnicas empleadas para el preprocesamiento así como los resultados obtenidos en cada una de ellas y que será de gran utilidad en los apartados posteriores.

3. Técnicas de clasificación.

Con los datos de dataset limpios del apartado anterior, es momento de emplear las técnicas correspondientes para realizar la clasificación. Principalmente son dos las técnicas que se han utilizado para elaborar las posibles soluciones de esta competición, **árboles de decisión** y **Random Forest**.

Analizando la naturaleza del problema y observando la competición en la plataforma **Kaggle**, he optado por utilizar estas dos técnicas ya que son las que pueden dar un mejor resultado de clasificación viendo los datos que forman el dataset y el conjunto de clases del mismo, que básicamente es un problema de clasificación binaria.

El optar por otro tipo de técnica en un problema como este pienso que sería como "matar a una mosca a cañonazos", ya que la mayoría de las veces se emplea este tipo de técnicas para resolver los problemas de clasificación de esta índole.

4. Presentación y discusión de resultados.

Una vez decididas las técnicas que se van a emplear para este problema en el anterior apartado, vamos a pasar a mostrar el proceso y resultado de cada una de ellas para poder compararlos y debatirlos.

Los **árboles de decisión** ha sido la primera técnica que he empleado, creandome para ello un modelo basado en las variables **Pclass**, **Sex**, **Age**, **SibSp**, **Parch**, **Fare** y **Embarked** para la construcción del mismo. Antes de crear la predicción en base al árbol construido vamos a visualizarlo. Para ello es necesario instalar y cargar una serie de paquetes que facilitan esa labor.

```

Console ~/Desktop/Master/SIGE/Practicas/Practica1/dataset/
> library(rpart.plot)
> library(RColorBrewer)
> library(rpart)
> library(rattle)
> library(rpart.plot)
> library(RColorBrewer)

```

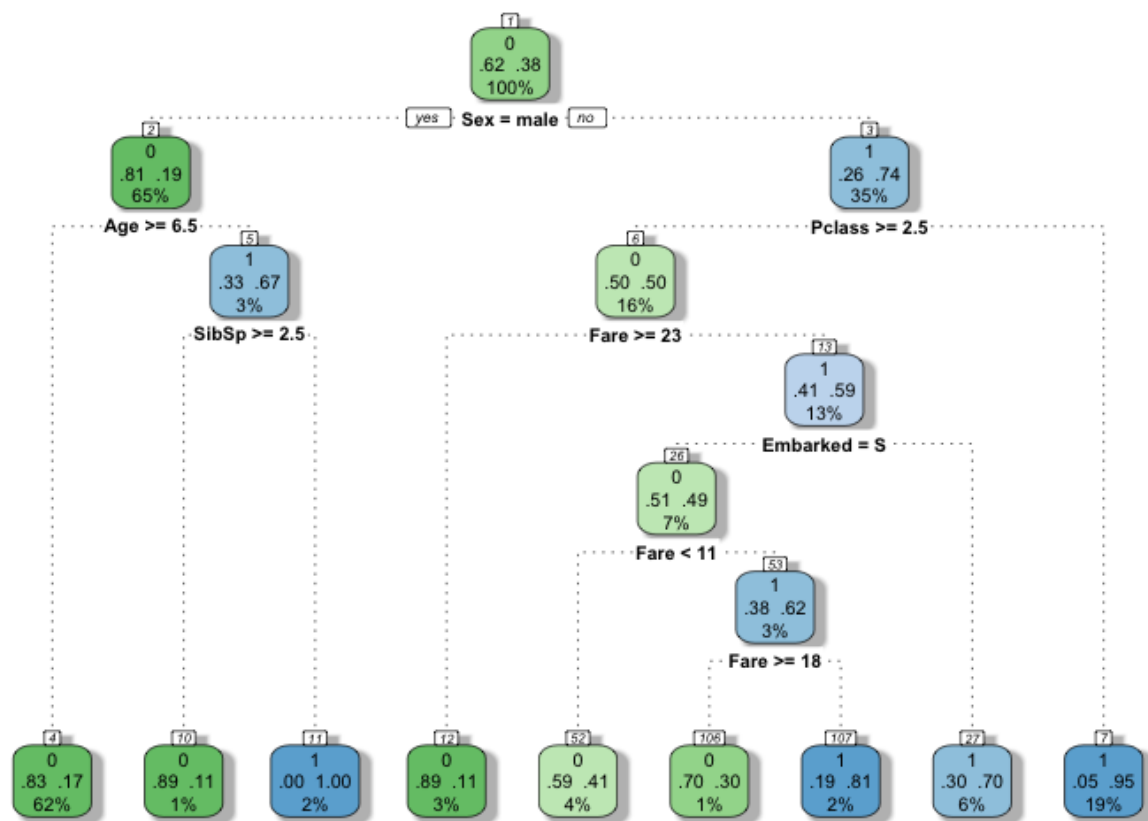
Figura 36: Cargando librerías.

```

Console ~/Desktop/Master/SIGE/Practicas/Practica1/dataset/
> fit <- rpart(Survived ~ Pclass + Sex + Age + SibSp + Parch + Fare + Embarked,
+             data=train,
+             method="class")

```

Figura 37: Creando árbol de decisión.



Rattle 2017-abr-25 11:49:24 jesusgarciamanday

Figura 38: Visualizando árbol de decisión.

Vemos en la anterior figura como se han encontrado decisiones para variables como **SibSp** e incluso el puerto de embarcación. Observamos también como del lado masculino los niños menores de 6 años tienen más oportunidad de sobrevivir, cumpliéndose de este modo la famosa regla de "los niños y las mujeres primero".

Habiendo evaluado el resultado del árbol vamos a calcular el porcentaje de predicción presentando dicho resultado en la plataforma Kaggle.

```
Console ~/Desktop/Master/SIGE/Practicas/Practica1/dataset/
> Prediction <- predict(fit, test, type = "class")
> submit <- data.frame(PassengerId = test$PassengerId, Survived = Prediction)
> write.csv(submit, file = "decisiontree.csv", row.names = FALSE)
```

Figura 39: Creando predicción y fichero csv.

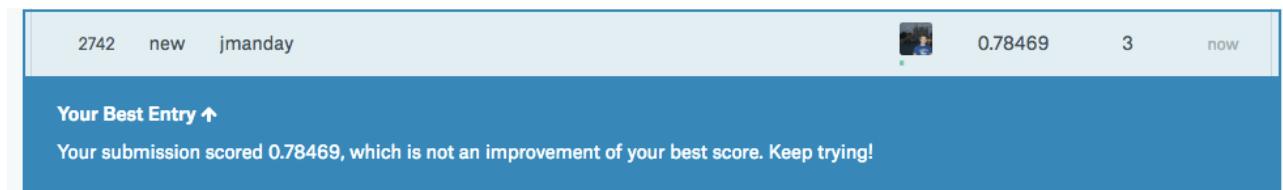


Figura 40: Resultado predicción en Kaggle.

Con esta primera técnica hemos obtenido una predicción del **78,469 %** sobre el conjunto de prueba. Para la segunda técnica se ha empleado **Random Forest**, que no es más que un conjunto de árboles de decisión creados en base a un subconjunto del modelo inicial de variables para cada árbol que votan sobre un subconjunto de las observaciones para decidir su clase en función a la mayoría de votos de la misma. El proceso de aleatoriedad tanto en la selección de las observaciones como en las variables para cada árbol, permite obtener un mayor grado de dinamismo frente al complemento estático que presenta el clásico árbol de decisión.

Vamos a construir el primer conjunto de árboles en base al modelo formado por las variables **Pclass**, **Sex**, **Age**, **SibSp**, **Parch**, **Fare**, **Embarked**, **Title**, **FamilySize** y **FamilyID2**. Para este tipo de árboles no le indicamos el parámetro **method=class** para la clasificación binaria, sino que obligamos a predecir cambiando temporalmente la variable objetivo a **factor**. Otro aspecto a comentar es el número de árboles que le especificamos a crear, en este caso 2000.

```
Console ~/Desktop/Master/SIGE/Practicas/Practica1/dataset/
> set.seed(415)
> fit <- randomForest(as.factor(Survived) ~ Pclass + Sex + Age + SibSp + Parch + Fare + Embarked + Title
+ FamilySize + FamilyID2,
+ data=train, importance=TRUE, ntree=2000)
```

Figura 41: Creando **Random Forest**.

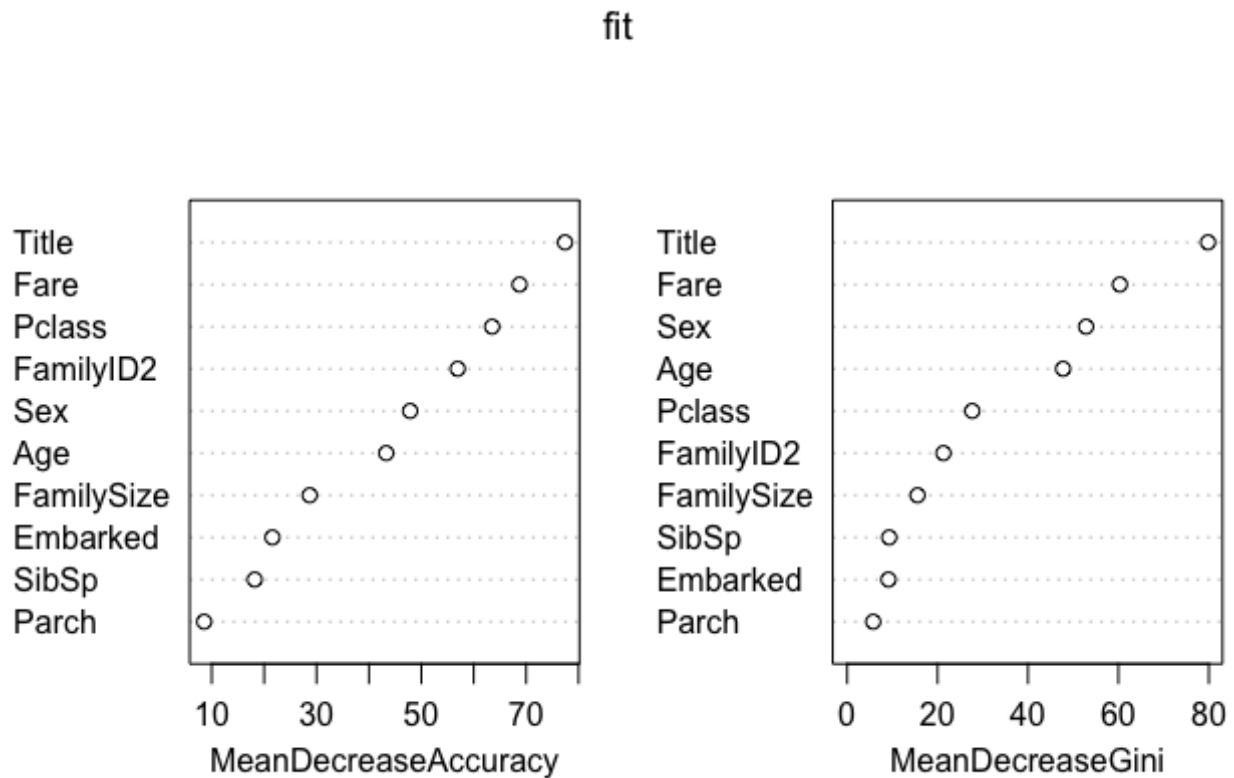


Figura 42: Analizando **RandomForest**.

Una vez que hemos creado el conjunto de árboles mediante dicho algoritmo, pasamos a ver la importancia que tienen las variables dentro del mismo. En la figura anterior se muestra con claridad como coinciden en el campo **Title** como el más importante y el atributo **Parch** como el menos relevante en ambos, variando un poco el resto de variables.

Para comprobar la fiabilidad de esta nueva clasificación vamos a crear una predicción en base a la misma para comprobar que porcentaje de acierto nos arroja **Kaggle**.

```

Console ~/Desktop/Master/SIGE/Practicas/Practica1/dataset/
> Prediction <- predict(fit, test)
> submit <- data.frame(PassengerId = test$PassengerId, Survived = Prediction)
> write.csv(submit, file = "reasuretree.csv", row.names = FALSE)

```

Figura 43: Creando predicción y fichero csv.

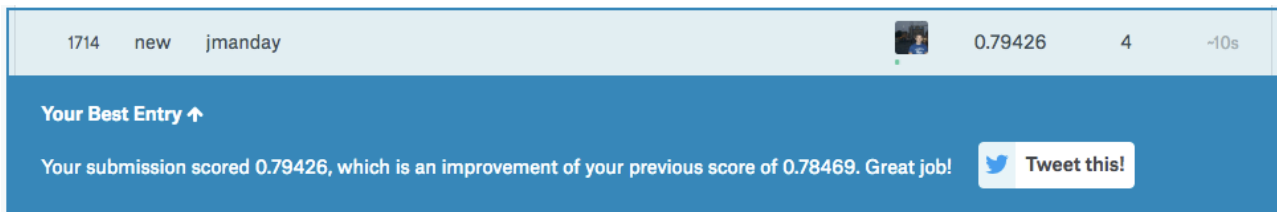


Figura 44: Resultado predicción en Kaggle.

El resultado de **79,426 %** de acierto en la clasificación obtenido con esta segunda técnica vemos que mejora al anterior, lo que nos dice que utilizando un conjunto de árboles podemos aumentar el porcentaje de acierto con respecto a un único árbol de decisión. Esto parece evidente ya que con la primera técnica nos basamos en el resultado de un único árbol creado en base a un único modelo, mientras que con **Random Forest** son muchos árboles que diferentes modelos de variables los que clasifican las diferentes observaciones, lo que nos da una mejora de conocimiento.

Vamos a probar a ajustar un poco mas el resultado utilizando la técnica de inferencia condicional dentro de los árbol de decisión. De este modo cada árbol toma su decisión en base a pruebas estadísticas en vez de utilizar una medida como en el realizado antes. Para usar esta alternativa hay que instalar y cargar el paquete **party**. Se especificarán el mismo número de árboles.

```
Console ~/Desktop/Master/SIGE/Practicas/Practica1/dataset/
> set.seed(415)
> fit <- cforest(as.factor(Survived) ~ Pclass + Sex + Age + SibSp + Parch + Fare + Embarked + Title + FamilySize + FamilyID,
+               data = train, controls=cforest_unbiased(ntree=2000, mtry=3))
> Prediction <- predict(fit, test, OOB=TRUE, type = "response")
> submit <- data.frame(PassengerId = test$PassengerId, Survived = Prediction)
> write.csv(submit, file = "randomforest.csv", row.names = FALSE)
```

Figura 45: Creando predicción y fichero csv.

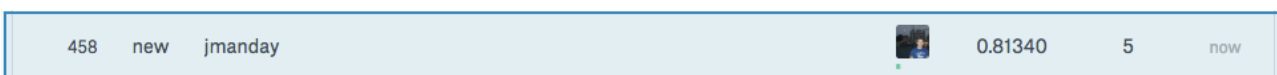


Figura 46: Resultado predicción en Kaggle.

Se ha podido mejorar la clasificación obtenida antes con esa única modificación obteniendo un **81,340 %** de acierto, lo que hace que haya conseguido el mejor de mis resultados hasta el momento y que sea el que se usará como puntuación final en la plataforma **Kaggle**.

Analizando todos los resultados obtenidos empleando ambas técnicas de clasificación, se ha podido comprobar como el algoritmo **Random Forest** obtiene mejores resultados que el clásico **árbol de decisión** como he mencionado antes. El que se utilice un conjunto de árboles diferentes aporta un grado de dinamismo y aprendizaje que se reflejan en la mejora obtenida de la clasificación, y como en esta última técnica, el variar el parámetro de configuración para que los árboles utilicen pruebas estadísticas en vez de las medidas tradicionales hace que mejoren mas. Por todo eso pienso que el emplear la técnica de **Random Forest** dará la mayoría de la veces mejor resultado.

5. Conclusiones y trabajo futuro.

Después de todas las técnicas de preprocesamiento utilizadas así como las diferentes técnicas de clasificación empleadas en todas las pruebas, se puede decir que el resultado obtenido es aceptable ya que supera el **80 %** de acierto, lo que lo coloca en una buena posición en la competición. No obstante, esto no quita que puedan realizarse posibles ajustes que puedan dar lugar a una mejora en la predicción.

Como posibles propuestas estaría la de explorar algunas variables más como **Cabin** o **Ticket** que en esta ocasión no se han analizado y que podrían aportar algo más de información sobre los pasajeros del viaje que ayuden a ajustar más el modelo. El probar diferentes configuraciones de los árboles de decisión sería otra de las propuestas a realizar, ya que aunque en la última predicción se modificó los parámetros del algoritmo **Random Forest** con respecto al anterior, quedan aún más parámetros como el número de árboles a crear, el número del subconjunto de variables, etc. que se pueden ir configurando para observar el comportamiento que va teniendo en los resultados.

6. Listado de soluciones.

En la siguiente tabla se muestran las diferentes soluciones obtenidas:

Solución	Preproc.	Alg. y soft.	% aciertos	Pos. Kaggle
1	Femenino sobrevive y masculino muere	Ninguno	76,555	4845
2	Child menor 18 y agrupar en Fare	Ninguno	77,990	3724
3	Los del apartado de la memoria	árbol de decisión/rpart	78,469	2742
4	Los del apartado de la memoria	random forest/cforest	79,426	1714
5	Los del apartado de la memoria	random forest + parámetros/cforest	81,340	458

Cuadro 1: Lista de soluciones.

7 submissions for **jmanday**

Sort by

Most recent

All Successful **Selected**

You can select up to 5 submissions to be used to calculate your final leaderboard score. If 5 submissions are not selected, they will be chosen based on your best submission scores on the public leaderboard.

Your final score will not be based on the same exact subset of data as the public leaderboard, but rather a different private data subset of your full submission —your public score is only a rough indication of what your final score is.

You should thus choose submissions that will most likely be best overall, and not necessarily on the public subset.

Submission and Description	Public Score	Use for Final Score
randomforest.csv 5 days ago by jmanday Using random forest	0.81340	<input checked="" type="checkbox"/>

No more submissions to show 🙄(ツ)🙄

Figura 47: Mejor solución obtenida.

7. Bibliografía.

<http://trevorstephens.com/kaggle-titanic-tutorial/getting-started-with-r/>