

INTRODUCCIÓ

L'objectiu d'aquesta activitat serà el tractament d'un dataset lliure disponible a <https://www.kaggle.com/c/titanic>. Es seguiran les principals etapes d'un projecte analític, descrits en els següents apartats.

PRÀCTICA

1. Descripció del dataset. Perquè és important i quina pregunta/problema pretén respondre?

Com s'ha introduït, per a la realització de la pràctica es treballarà amb un dataset relacionat amb l'enfonsament del Titànic. El seu contingut és molt simple: un conjunt de registres de passatgers, amb diferents dades personal que s'explicaran més endavant. La informació més important, i en la que es basa l'estudi és que per cada passatger, es descriu si ha sobreviscut o no.

Per tant, l'estudi actual consisteix en esbrinar, a partir de diferents dades dels passatgers, si han sobreviscut o no. Dit d'una altra forma; ens trobem davant d'un problema de classificació binària.

De totes maneres, abans d'endinsar-nos en el treball de classificació, cal tractar les diferents variables, i per això, procedirem a descriure-les:

VARIABLE	DESCRIPCIÓ
<i>PassengerId</i>	Identificador del passatger.
<i>Pclass</i>	Classe en la que viatja el passatger: 1 = primera. 2 = segona. 3 = tercera.
<i>Name</i>	Nom i Cognom del passatger.
<i>Sex</i>	Sexe.
<i>Age</i>	Edat.
<i>SibSp</i>	Numero de familiars: germans, marits i mullers.
<i>Parch</i>	Numero de familiars: pare, mare,...
<i>Ticket</i>	Número del ticket.
<i>Fare</i>	Tarifa del ticket.
<i>Cabin</i>	Número de la cabina.
<i>Embarked</i>	On ha embarcat el passatger: C = Cherbourg. Q = Queenstown. S = Southampton.
<i>Survived</i>	Si el passatger ha sobreviscut: 0 = mor. 1 = viu.

A l'enllaç esmentat a la introducció hi trobem realment dos datasets: "train" i "test". El primer, serveix per a crear el model de classificació, mentre que el segon s'utilitza per a posar el model a prova. Com és lògic, el dataset "test" NO conté la variable "Survived"; ja que és la que haurem de trobar. Durant els següents apartats treballaré amb el primer dataset, ja que hauré de generar el model. A l'apartat 6 **Resolució del problema. A partir dels resultats obtinguts, quines són les conclusions? Els resultats permeten resoldre el problema?**, a banda d'explicar les conclusions trobades del dataset "train", també es procedirà a aplicar el model sobre el dataset "test" per esbrinar quins passatgers van sobreviure.

2. Integració i selecció de les dades d'interès a analitzar.

En total hi ha 11 variables (sense comptar la variable classificatòria "Survived"). De totes aquestes, hi ha algunes que no són rellevant per a l'estudi, o bé que tenen relacions entre elles.

Òbviament, la variable "Survived" és necessària per generar el model.

Les variables "Pclass", "Sex", "Fare", "Age" i "Embarked" són variables que ens permeten desgranar la tipologia del passatger. La classe, la tarifa i la localitat del embarcament, ens dona una idea de l'estatus del passatger.

```
data_vars = data[["Pclass", "Sex", "Age", "Embarked", "Fare", "Survived"]]
```

3. Neteja de les dades.

3.1. Les dades contenen zeros o elements buits? Com gestionaries aquests casos?

La variable "Sex" conté un total de 177 d'elements nuls i "Embarked" només 2.

```
for var in data_vars:
    print("La variable", var, "conté", data_vars[var].isna().sum(), "valors nuls")

La variable Pclass conté 0 valors nuls
La variable Sex conté 0 valors nuls
La variable Age conté 177 valors nuls
La variable Embarked conté 2 valors nuls
La variable Fare conté 0 valors nuls
La variable Survived conté 0 valors nuls
```

Hi ha un total de 891 registres i per tant, podem eliminar les dues observacions on tenim un element buit de la variable "Embarked". En canvi, 177 valors nuls són molts per tal d'obviar-los; per tant, es reompliran de la següent forma: Es calcularà la mediana de les edats agrupades per sexe i s'assignarà aquest valor.

```
# Reomplim els NA a partir de la mediana de les edats per Sexe.
data_vars['Age'].fillna(data_vars.groupby('Sex')['Age'].transform("median"), inplace=True)
# Elimino les dues observacions amb NA de "Embarked"
data_vars = data_vars.dropna()
```

3.2. Conversió de valors.

Aquest apartat s'ha afegit a la pràctica per tal de convertir variables categòriques en numèriques i així, facilitar el seu anàlisi.

Es faran un total de tres tipus de conversions:

- Sexe: Als homes s'assignarà un 1 i a les dones un 2.
- Embarcament: El valor C = 1, el Q = 2 i el S = 3.
- Edat: L'edat està definida com a "float". La convertiré en un enter. Aquest procés arrodoneix els valors i per tant és possible que apareixin valors = 0. En aquests casos no realitzaré cap acció ja que consideraré al passatger com a un nadó.

```
#Conversió "male" - "female" en enters:
data_clean = data_vars.replace("female", 2)
data_clean = data_clean.replace("male", 1)

# Conversió dels tres valors d'embarcament:
data_clean["Embarked"] = data_clean["Embarked"].replace("C", 1)
data_clean["Embarked"] = data_clean["Embarked"].replace("Q", 2)
data_clean["Embarked"] = data_clean["Embarked"].replace("S", 3)

# Conversió de "Age" a enters:
data_clean["Age"] = data_clean["Age"].round().astype(int)
```

3.3. Identificació i tractament de valors extrems.

Si revisem els valors *max* i *min* de la taula *describe()* del dataframe, veiem que les variables tenen valors acotats i no hi ha cap que s'escapi de la lògica:

```
print(data_clean.describe())
```

	Pclass	Sex	Age	Embarked	Fare	Survived
count	889.000000	889.000000	889.000000	889.000000	889.000000	889.000000
mean	2.311586	1.350956	29.390326	2.535433	32.096681	0.382452
std	0.834700	0.477538	12.982384	0.792088	49.697504	0.486260
min	1.000000	1.000000	0.000000	1.000000	0.000000	0.000000
25%	2.000000	1.000000	22.000000	2.000000	7.895800	0.000000
50%	3.000000	1.000000	29.000000	3.000000	14.454200	0.000000
75%	3.000000	2.000000	35.000000	3.000000	31.000000	1.000000
max	3.000000	2.000000	80.000000	3.000000	512.329200	1.000000

4. Anàlisi de les dades.

4.1. Selecció dels grups de dades que es volen analitzar/comparar (planificació dels anàlisis a aplicar).

Com ja s'ha explicat, es procedirà a seleccionar únicament les 5 variables descriptives i la variable classificatòria "Survived".

En els següents apartats es realitzarà un estudi de correlacions per tal de veure la influència de cada una de les 5 variables sobre “Survived”. De moment, repassem la tipologia i els possibles valors que poden adquirir totes les variables:

- Pclass: 3, 2, 1.
- Sex: 1, 2.
- Age: enters 0 – 80.
- Fare: float 0 -512.3292.
- Survivec: 0, 1.

4.2. Comprovació de la normalitat i homogeneïtat de la variància.

Per revisar la normalitat s’aplicarà la prova de normalitat Anderson-Darling. Aquesta prova consisteix parteix d’un hipòtesi nul·la on es considera una distribució normal; i una hipòtesi alternativa que considera una distribució no normal. Mitjançant la prova *Anderson-Darling* s’obté un p-valor per a cada variable. En el supòsit que $p \leq \alpha$, es refusa la hipòtesi nul·la i per tant, es considera que la distribució NO és normal. En cas contrari on $p > \alpha$, es confirma la hipòtesi nul·la que la distribució és normal. El valor de α o de significació definit és 0.05.

```
# Revisem normalitat (Prova de normalitat Anderson-Darling).
# Si p-value > 0.5, implica distribució normal.
for var in data:
    res = scipy.stats.anderson(data[var])
    print("El p-value de la variable",var,"és",res[1][2], "i el estadístic és"
, res[0])
```

```
El p-value de la variable Pclass és 0.783 i el estadístic és 110.07359556722486
El p-value de la variable Sex és 0.783 i el estadístic és 178.57044801344568
El p-value de la variable Age és 0.783 i el estadístic és 15.177390367849057
El p-value de la variable Embarked és 0.783 i el estadístic és 182.60119840206448
El p-value de la variable Fare és 0.783 i el estadístic és 122.80816188550375
El p-value de la variable Survived és 0.783 i el estadístic és 171.3628439667541
```

Veiem que en tots els casos, el p-valor és inferior al nivell de significació 0.05; per tant, acceptem la hipòtesi nul·la que indica que la distribució és normal.

Un cop descoberta la normalitat, procedim a esbrinar la homogeneïtat de la variància. En aquest cas s’utilitzarà el test de Levene. De nou, partim de dues hipòtesis. La nul·la, confirma la homogeneïtat de la variància; en canvi, la hipòtesi nul·la, indica un diferència entre variàncies. El p-valor retornat en el test de Levene servirà per triar la hipòtesis. En el cas que $p \leq \alpha$, es refusa la hipòtesi nul·la; i en cas contrari on $p > \alpha$, es confirma la hipòtesi nul·la.

En aquest cas s’agafen les tres variables categòriques “Sex”, “Pclass” i “Embarked” o es comparen els seus valors possibles amb la variables “Survived”.

```
hom_sex = scipy.stats.levene(data["Survived"][data["Sex"]==1],
                             data["Survived"][data["Sex"]==2])

hom_class = scipy.stats.levene(data["Survived"][data["Pclass"]==1],
                               data["Survived"][data["Pclass"]==2],
                               data["Survived"][data["Pclass"]==3])
```

```
hom_emb = scipy.stats.levene(data["Survived"][data["Embarked"]==1],
                             data["Survived"][data["Embarked"]==2],
                             data["Survived"][data["Embarked"]==3])

print(hom_sex)
print(hom_class)
print(hom_emb)
```

```
LeveneResult(statistic=6.052073198971666, pvalue=0.014079700102452174)
LeveneResult(statistic=18.743249946154936, pvalue=1.0651350809216878e-08)
LeveneResult(statistic=3.6101502536808128, pvalue=0.027446419795460055)
```

En aquest cas hem obtingut uns p-valor molt inferior al nivell de significació 0.05; per tant, refusem la hipòtesi nul·la i confirmem que hi ha diferència entre les variàncies.

4.3. Aplicació de proves estadístiques per comparar els grups de dades. En funció de les dades i de l'objectiu de l'estudi, aplicar proves de contrast d'hipòtesi, correlacions, regressions,...

En aquest apartat es farà un estudi de la correlació entre les diferents variables per observar quines tenen més pes sobre "Survived". Un cop aplicada la correlació, es crearan tres models de regressió lineal que classificarà als passatgers entre supervivents i no supervivents.

- Correlació entre variables.

```
corr = data.corr()
print(corr)
```

	Pclass	Sex	Age	Embarked	Fare	Survived
Pclass	1.000000	-0.127741	-0.335053	0.164681	-0.548193	-0.335549
Sex	-0.127741	1.000000	-0.100592	-0.110320	0.179958	0.541585
Age	-0.335053	-0.100592	1.000000	-0.020347	0.091372	-0.077904
Embarked	0.164681	-0.110320	-0.020347	1.000000	-0.226311	-0.169718
Fare	-0.548193	0.179958	0.091372	-0.226311	1.000000	0.255290
Survived	-0.335549	0.541585	-0.077904	-0.169718	0.255290	1.000000

De la taula anterior, veiem que les tres variables que tenen més força sobre "Survived" són, per ordre: "Sex", "Fare" i "Age".

- Regressió lineal.

Es crearan tres models diferents:

- Model 1: Amb les 5 variables (Pclass, Sex, Age, Embarked i Fare).
- Model 2: Amb les tres variables més fortes sobre "Survived" trobades a partir de l'estudi de correlació (Sex, Fare i Age).
- Model 3: Únicament amb la variable que té més força (Sex).

Per a no repetir tres cops el mateix codi, creo una funció `red_lin()` que a partir de les dades d'entrada, retorna tres valors: Error Quadràtic Mig (MSE), Precisió (Accuracy) i la matriu de confusió.

```
def red_lin(data_X, data_Y):
```

```

data_X_train = data_X[:-200]
data_Y_train = data_Y[:-200]

data_X_test = data_X[-200:]
data_Y_test = data_Y[-200:]

regr = linear_model.LinearRegression()
regr.fit(data_X_train, data_Y_train)

survived_pred = regr.predict(data_X_test)

# Càlculs:
MSE = mean_squared_error(data_Y_test, survived_pred)
acc = accuracy_score(data_Y_test, survived_pred.round())
cm = confusion_matrix(data_Y_test, survived_pred.round())
return(MSE, acc, cm)

data_X_m1 = data[["Pclass", "Sex", "Age", "Embarked", "Fare"]]
data_X_m2 = data[["Sex", "Age", "Fare"]]
data_X_m3 = data[["Sex"]]
data_Y = data[["Survived"]]

MSE1, acc1, cm1 = reg_lin(data_X_m1, data_Y)
MSE2, acc2, cm2 = reg_lin(data_X_m2, data_Y)
MSE3, acc3, cm3 = reg_lin(data_X_m3, data_Y)

```

En els següents apartats es mostraran els gràfics i resultats.

5. Representació dels resultats a partir de taules i gràfiques.

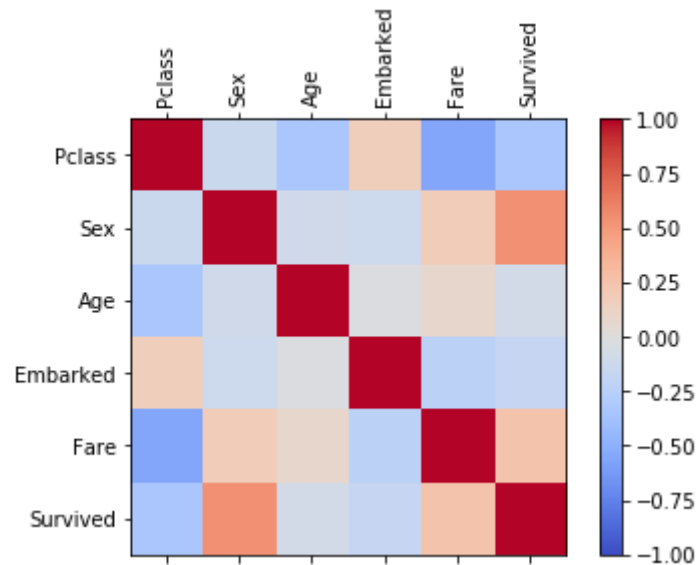
- Correlacions.

Mostrem primerament la correlació entre variables. Anteriorment hem obtingut una taula/matriu numèrica amb els valors de correlació entre les variables. Per a millorar la visualització, procedim a graficar amb diferents colors els pesos de les variables. Un color vermell implica una relació més forta que un color blavós.

```

# Correlació de les variables:
fig = plt.figure()
ax = fig.add_subplot(111)
cax = ax.matshow(corr, cmap='coolwarm', vmin=-1, vmax=1)
fig.colorbar(cax)
ticks = np.arange(0, len(data.columns), 1)
ax.set_xticks(ticks)
plt.xticks(rotation=90)
ax.set_yticks(ticks)
ax.set_xticklabels(data.columns)
ax.set_yticklabels(data.columns)
plt.show()

```



Podem confirmar doncs que les tres variables que tenen més pes sobre “Survived” són, per ordre: “Sex”, “Fare” i “Age”.

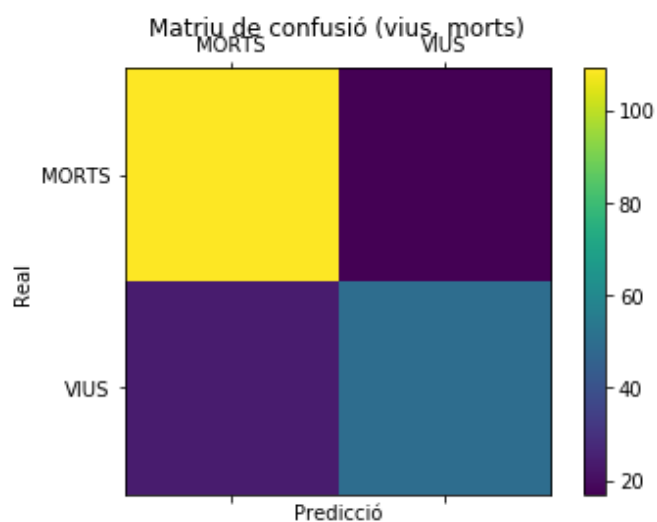
- Matrius confusió dels models.

Per a cada un dels tres models creats, s’ha graficat la seva matriu de confusió (a més dels seus valors numèrics):

```
# Matrius de Confusió dels models:
def representa_mtx(cm):
    labels = ['MORTS', 'VIUS']
    fig = plt.figure()
    ax = fig.add_subplot(111)
    cax = ax.matshow(cm)
    plt.title("Matriu de confusió (vius, morts)")
    fig.colorbar(cax)
    ax.set_xticklabels([''] + labels)
    ax.set_yticklabels([''] + labels)
    plt.xlabel('Predicció')
    plt.ylabel('Real')
    plt.show()
    print("La taula amb els valors és la següent:\n", cm)

representa_mtx(cm1)
representa_mtx(cm2)
representa_mtx(cm3)
```

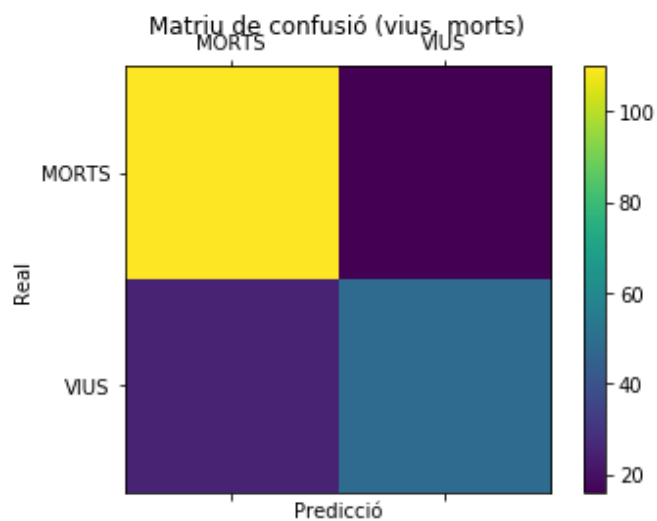
- Model 1 (Amb totes les variables triades):



La taula amb els valors és la següent:

```
[[109 17]
 [ 24 50]]
```

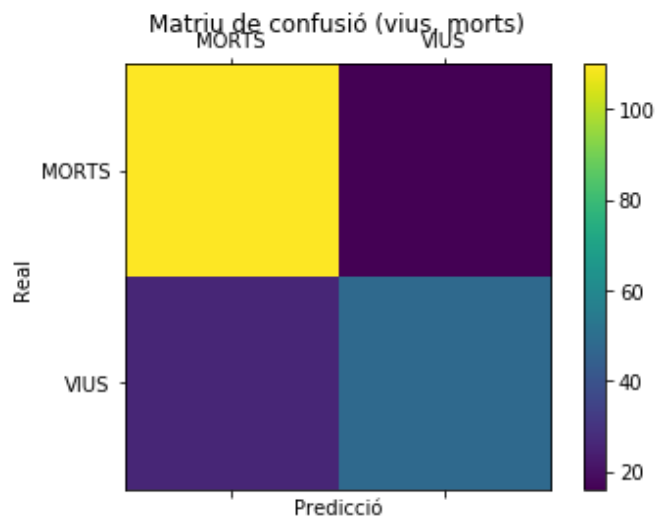
- Model 2 (Amb les variables "Sex", "Fare" i "Age"):



La taula amb els valors és la següent:

```
[[110 16]
 [ 25 49]]
```


- Model 3 (Amb la variable "Sex"):



La taula amb els valors és la següent:

```
[[110 16]
 [ 26 48]]
```

Veiem que entre els tres models, la diferència és mínima. Les matrius de confusió quan triem 5 variables o quan només triem "Sex" varien en 1 o 2 valors; per tant, podem reduir dràsticament la dimensionalitat. Confirmem visualitzant els resultat de MSE i la precisió:

```
# Valors de MES i precisió:
print("MODEL 1:\n      Error quadràtic mig (MSE) =",MSE1," \n      Precisió =",acc
1)
print("MODEL 2:\n      Error quadràtic mig (MSE) =",MSE2," \n      Precisió =",acc
2)
print("MODEL 3:\n      Error quadràtic mig (MSE) =",MSE3," \n      Precisió =",acc
3)
```

```
MODEL 1:
      Error quadràtic mig (MSE) = 0.1367640225295661
      Precisió = 0.795
MODEL 2:
      Error quadràtic mig (MSE) = 0.1540867313916442
      Precisió = 0.795
MODEL 3:
      Error quadràtic mig (MSE) = 0.1651998745924483
      Precisió = 0.79
```

La única variació que observem és la de l'error quadràtic mig.

6. Resolució del problema. A partir dels resultats obtinguts, quines són les conclusions? Els resultats permeten resoldre el problema?

Recordem que el problema, inclou un segon dataframe de test el qual no conte la variable classificada "Survived". Un cop generats els tres model anteriors, procedirem a aplicar un d'ells per tal d'esbrinar dins d'aquest dataframe de test, quins passatgers han sobreviscut i quins no.

Sembla que la variable que acaba definint la supervivència del passatger, correspon al sexe. De totes maneres, per tal de no simplificar radicalment l'exercici, s'optarà per utilitzar el segon model on a banda del sexe, també es tenen en compte aspectes com l'edat i la tarifa del ticket.

Carreguem el dataframe de test, i visualitzem si hi ha cap valor nul en les tres variables: "Sex", "Age" i "Fare":

```
# Càrrega del dataset.
data_test= pd.read_csv("test_titanic.csv")

data_test_vars = data_test[["Sex", "Age", "Fare"]]

print ("El dataset conté:", len(data_test_vars), "registres")

for var in data_test_vars:
    print("La variable", var, "conté", data_test_vars[var].isna().sum(), "valors nuls")
```

```
El dataset conté: 418 registres
La variable Sex conté 0 valors nuls
La variable Age conté 86 valors nuls
La variable Fare conté 1 valors nuls
```

En aquest cas, a banda de tenir valors perduts en la variable "Age", també en trobem un en la variable "Fare". Procediré a reomplir aquests valors amb la mateixa metodologia que amb el dataframe de traint: calculant la mediana per sexes:

```
#### NETEJA
# Reomplim els NA a partir de la mediana de les edats per Sexe.
data_test_vars['Age'].fillna(data_test_vars.groupby('Sex')['Age'].transform("median"), inplace=True)
data_test_vars['Fare'].fillna(data_test_vars.groupby('Sex')['Fare'].transform("median"), inplace=True)

#### CONVERSIÓ
# Conversió "male" - "female" en enters:
data_test_clean = data_test_vars.replace("female", 2)
data_test_clean = data_test_clean.replace("male", 1)
# Conversió de "Age" a enters:
data_test_clean["Age"] = data_test_clean["Age"].round().astype(int)
```

A més, també s'han modificat la variable "Sex" i arrodonit "Age".

Finalment, s'aplica el segon model creat anteriorment "regr2" i es recompten el nombre de morts i de supervivents:

```
pred = regr2.predict(data_test_clean)
resultat = data_test.assign(Survived_PRED = pred.round())

print(resultat[0:10])

# Recompte de morts i supervivents.
sumari = resultat["Survived_PRED"].value_counts()
print("En total moren", sumari[0], "passatgers i sobreviuen", sumari[1], "passa
tgers")

# Emmagatzemo
resultat.to_csv("resultat.csv")
```

\	PassengerId	Pclass	Name	Sex
0	892	3	Kelly, Mr. James	male
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female
2	894	2	Myles, Mr. Thomas Francis	male
3	895	3	Wirz, Mr. Albert	male
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female
5	897	3	Svensson, Mr. Johan Cervin	male
6	898	3	Connolly, Miss. Kate	female
7	899	2	Caldwell, Mr. Albert Francis	male
8	900	3	Abraham, Mrs. Joseph (Sophie Halaut Easu)	female
9	901	3	Davies, Mr. John Samuel	male

	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Survived_PRED
0	34.5	0	0	330911	7.8292	NaN	Q	0.0
1	47.0	1	0	363272	7.0000	NaN	S	1.0
2	62.0	0	0	240276	9.6875	NaN	Q	0.0
3	27.0	0	0	315154	8.6625	NaN	S	0.0
4	22.0	1	1	3101298	12.2875	NaN	S	1.0
5	14.0	0	0	7538	9.2250	NaN	S	0.0
6	30.0	0	0	330972	7.6292	NaN	Q	1.0
7	26.0	1	1	248738	29.0000	NaN	S	0.0
8	18.0	0	0	2657	7.2292	NaN	C	1.0
9	21.0	2	0	A/4 48871	24.1500	NaN	S	0.0

En total moren 264 passatgers i sobreviuen 154 passatgers

7. Codi: Cal adjuntar el codi, preferiblement en R, amb el que s'ha realitzat la neteja, anàlisi i representació de les dades. Si ho preferiu també podeu treballar en Python.

Tot el codi utilitzat, s'ha adjuntat a mida que s'anava explicant aquesta PRAC. A més, a github s'ha pujat l'arxiu notebook.