Titanic dataset: Data cleaning and validation

Jesús Ros Solé

9 de diciembre, 2018

Contents

1. Dataset description

This dataset has been obtained from the Kaggle competition Titanic: Machine Learning from Disaster and contains a set of atributes for each of the passengers boarding the Titanic in the day of its accident.

It contains two datasets: a training dataset to build models with and a test dataset to perform predictions on. The training dataset has information on 891 passengers (rows) and 12 attributes (columns) while the test dataset has 418 passengers and 11 attributes (minus the target attribute `Survived`).

The attribute descriptions are:

- *PassengerID*: an integer identifier for each passenger.
- *Survived*: target class encoded as `1` (survived) or `0` (not survived). Missing in test set.
- *Pclass*: boarding class encoded as `1` (first class), `2` (second class) or `3` (third class).
- *Name*: name of the passenger as a string.
- *Sex*: sex of the passenger encoded as `male` or `female`.
- *Age*: age of the passenger,
- *SibSp*: number of siblings/spouse aboard.
- *Parch*: number of parent/child aboard.
- *Ticket*: identifier of the boarding pass.
- *Fare*: ticket fare amount.
- *Cabin*: identifier of the passenger cabin.

- *Embarked*: identifier for the port in which the passenger embarked encoded as `C` (Cherbourg), `S` (Southampton) and `Q` (Queenstown).

The objective of this work is study the groups of people that are more likely to survive according to the given attributes and to predict the survival chances of the people in the test set.

2. Data integration and selection

First we will load the training and test set using `read.csv` indicating the corresponding data types. We then extract the target attribute from the test set and merge the training and test datasets for the next steps.

```
train <- read.csv("../data/raw/train.csv",
                  colClasses=c("integer", "factor" ,"factor" ,"character" ,"factor"
                               ,"numeric" ,"integer" ,"integer" ,"character" ,"numeric"
                               ,"character" ,"factor"),
                  na.strings = c("NA", ""))
test <- read.csv("../data/raw/test.csv",
                 colClasses=c("integer" ,"factor" ,"character" ,"factor" ,"numeric"
                              ,"integer" ,"integer" ,"character" ,"numeric"
                              ,"character" ,"factor"),
                 na.strings = c("NA", ""))

y_train <- train["Survived"]
train$Survived <- NULL
test_id <- test["PassengerId"]

all <- rbind(train, test)
```

Now that we have our data properly loaded, we will select the attributes that are going to be useful for the posterior analysis. First note that the `PassengerID` and `Ticket` attributes will not yield much information to our analysis since they are mostly unique identifiers and can thus be ignored. The `Name` attribute by itself will not be very informative either, but we can extract the passenger title from it.

```
all$Title <- gsub("^.*, (.*?)\\..*$","\\1",all$Name)
table(all$Title)
```

```
## 
##         Capt          Col          Don         Dona           Dr 
##            1            4            1            1            8 
##     Jonkheer         Lady        Major       Master         Miss 
##            1            1            2           61          260 
##         Mlle          Mme           Mr          Mrs           Ms 
##            2            1          757          197            2 
##          Rev          Sir the Countess 
##            8            1            1 
```

By looking at the obtained titles we can see that `Master`, `Miss`, `Mr` and `Mrs` are the most common while the others are quite rare. We will try to aggregate some of them to the most common ones (for instance `Ms` is a different spelling from `Miss`) and create an additional class for the rest.

```
all$Title[all$Title %in% c("Mlle", "Ms")] <- "Miss"
all$Title[!(all$Title %in% c('Master', 'Miss', 'Mr', 'Mrs'))] <- "Other"
table(all$Title)
```

```
## 
## Master   Miss     Mr    Mrs  Other 
##     61    264    757    197     30 
```

```
all$Title <- as.factor(all$Title)
```

Additinally, we can create a new variable indicating the family size aboard the Titanic from the `SibSp` and `Parch` attributes.

```
all$FamilySize <- all$SibSp+all$Parch+1 #includes self
```

Now we can drop the variables that yield no information `PassengerId`, `Name` and `Ticket`.

```
drop <- c("PassengerId", "Name", "Ticket")
all <- all[, !(names(all) %in% drop)]
```

3. Data cleaning

3.1. Empty values

At this point, we have selected the attributes that will be useful for our posterior analysis and have created a couple new derived attributes from our dataset. Now we will inspect the remaining attributes for empty values that need to be taken care of.

```
count_empty <- function(attr){sum(is.na(attr))}
sapply(all,count_empty)
```

```
##     Pclass        Sex        Age      SibSp      Parch       Fare
##          0          0        263          0          0          1
##      Cabin   Embarked      Title FamilySize
##       1014          2          0          0
```

We observed that the `Cabin` attribute contains mostly empty elements, in fact, about 77% of the rows contain missing data. Therefore, this attribute can be ignored since it will not yield much information. The other attributes that contain missing data are `Age`, `Fare` and `Embarked`. We will use kNN imputation with default settings after dropping the `Cabin` column.

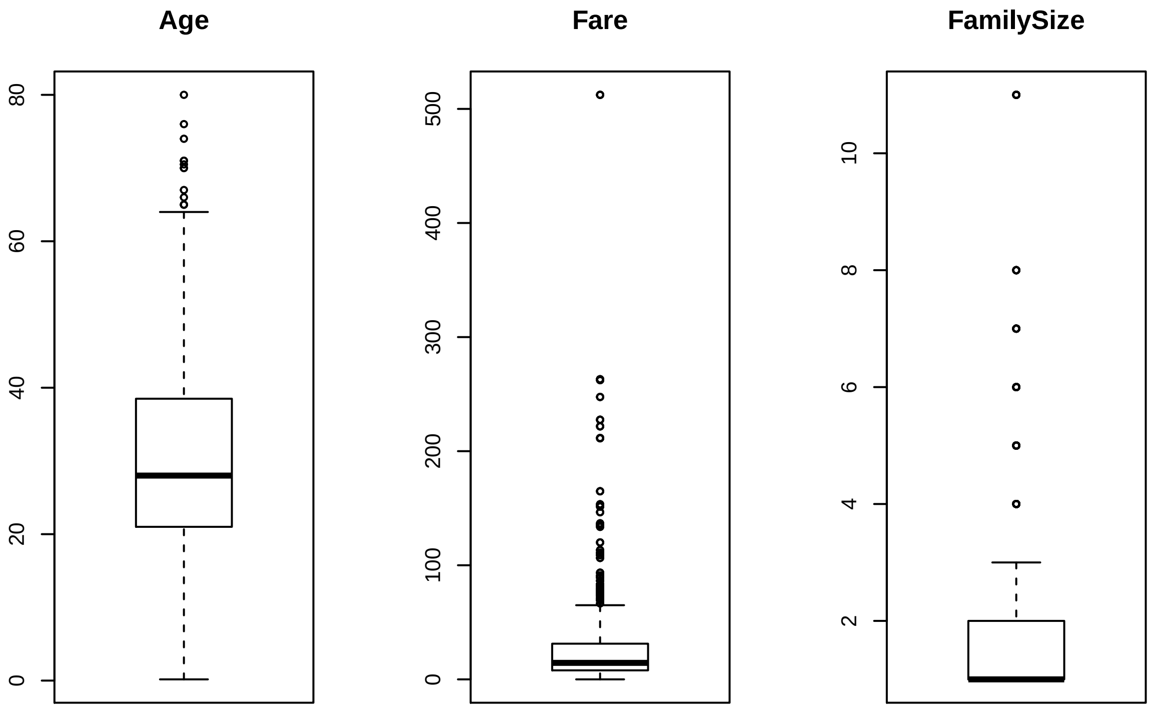
```
all$Cabin <- NULL
all_clean <- suppressWarnings(kNN(all,imp_var=FALSE))
sapply(all_clean,count_empty)
```

```
##     Pclass        Sex        Age      SibSp      Parch       Fare
##          0          0          0          0          0          0
##   Embarked      Title FamilySize
##          0          0          0
```

3.2. Extreme scores

Next we will investigate the extreme values in the dataset for numeric attributes. Numeric attributes are `Age`, `SibSp`, `Parch`, `Fare` and `FamilySize`. Since `FamilySize` is derived from `SibSp` and `Parch` we will only investigate `Age`, `Fare` and `FamilySize` using a boxplot.

```
par(mfrow=c(1,3))
boxplot(all_clean$Age, main="Age")
boxplot(all_clean$Fare, main="Fare")
boxplot(all_clean$FamilySize, main="FamilySize")
```



We observe some outliers in all 3 variables but their values seem to be realistic. We have ages up to 80 years, fares up to 500 dollars and families of up to 11 members. Therefore we decide to leave the outliers as is.

Finally, we have a clean dataset for further analysis. We will export the clean dataset to a datafile, respecting the initial traning and test partition and adding the target attribute to the training file.

```r
train <- cbind(all_clean[1:nrow(train),],y_train)
test <- all_clean[(nrow(train)+1):nrow(all_clean),]
write.csv(train, "../data/clean/train.csv", row.names=F)
write.csv(test, "../data/clean/test.csv", row.names=F)
```

4. Data analysis

4.1. Objective

The objective of the analysis is two-fold: first we will investigate associations between the variables with the target attribute `Survived`. This analysis will help us decide which models to build afterwards.

Since we have both numerical and categorical attributes and the target attribute is categorical, we will use the corresponding p-value from a One-Way ANOVA test to test signifiance between continuous variables and the target attribute and the p-value from a chi-squared test to test signifiance between categorical variables and the target attribute.

The null hypothesis for the One-Way ANOVA test is that all group produce the same variation on the response, on average. The null hypothesis for the chi-squared test is that the two variables are independent.

4.2. Normality and homogeneity of variance

First of all, we will test the numerical attributes for normality using the Shapiro-Wilk test and the homogeneity of the variance for the different groups where we apply the One-Way ANOVA test using Levene's Test.

```
numeric_attr <- c("Age", "SibSp", "Parch", "Fare", "FamilySize")
#Shapiro-Wilk test
shapiro.wilk <- sapply(train[numeric_attr],function(x){shapiro.test(x)$p.value})
#Levene's Test for homogeneity of variance
compute_levene <- function(attr_name, dataset, target_name){
        leveneTest(as.formula(paste(attr_name,target_name,sep="~")),
                              data=dataset)[["Pr(>F)"]][[1]]
}
levene <- sapply(numeric_attr, compute_levene, dataset=train,target_name="Survived")
rbind(shapiro.wilk,levene)
```

```
##                         Age        SibSp        Parch         Fare
## shapiro.wilk 2.017559e-08 5.750831e-44 2.386622e-43 1.084045e-43
## levene       1.652388e-01 2.922439e-01 1.479925e-02 3.337353e-11
##                FamilySize
## shapiro.wilk 1.567118e-40
## levene       9.360352e-01
```

We can see that none of the variables appear to be normally distributed and that the attributes `Parch` and `Fare` do not have equal variances when grouped by the target value.

4.3. Association with target variable

Now we can finally explore the association between variables, specifically which variables are associated with the target attribute `Survived`. This will hint on which variables can be more informative during model building.

Since some of the groups have unequal variances we will be using the Welch ANOVA test that does not assume equal variances between groups. Additionally, we can use this parametric test because we have large sample sizes, even though we have seen they are not normally distributed. An alternative would be to use a non-parametric test such as the Kruskal-Wallis test.

```
compute_corr <- function(attr_name, dataset, target_name){
        if(is.numeric(dataset[[attr_name]])){
                pvalue <- oneway.test(as.formula(paste(attr_name, target_name, sep="~")), data=dataset)$
        } else { #it is factor
                pvalue <- chisq.test(dataset[[attr_name]], dataset[[target_name]])$p.value
        }
        pvalue
}

sapply(names(train[-ncol(train)]),compute_corr,dataset=train,target_name="Survived")
```

```
##       Pclass          Sex          Age        SibSp        Parch
## 4.549252e-23 1.197357e-58 1.190510e-01 2.326626e-01 1.339484e-02
##         Fare     Embarked        Title   FamilySize
## 2.699332e-11 2.300863e-06 8.511041e-61 5.853351e-01
```

We can see that for all variables except `Age`, `SibSp` and `FamilySize` the results are significant at a 95% confidence level, that is those variables are associated with the target attribute `Survived`.

4.4. Model building

Now that we understand the association between the atributes and the response variable, we will propose several models to predict the test set. Since we are interested in models that output the probability of survival, we will use logistic regression models with different attributes as models.

First we will inspect a model with all variables that turned out significant in the previous analysis. That is, a model that related `Survived` with `Pclass`, `Sex`, `Parch`, `Fare`, `Embarked` and `Title`.

```
m1 <- glm(Survived ~ Pclass + Sex + Parch + Fare + Embarked + Title, family=binomial(link='logit'), data
summary(m1)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Sex + Parch + Fare + Embarked +
##     Title, family = binomial(link = "logit"), data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4623  -0.6341  -0.3660   0.6170   2.3451
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  18.257723 437.522408   0.042 0.966714
## Pclass2      -0.835196   0.297943  -2.803 0.005060 **
## Pclass3      -1.993777   0.286341  -6.963 3.33e-12 ***
## Sexmale     -15.257644 437.522134  -0.035 0.972181
## Parch        -0.461983   0.129339  -3.572 0.000354 ***
## Fare          0.001793   0.002334   0.768 0.442382
## EmbarkedQ    -0.269478   0.384326  -0.701 0.483197
## EmbarkedS    -0.603872   0.241278  -2.503 0.012321 *
## TitleMiss   -15.327244 437.522307  -0.035 0.972054
## TitleMr      -3.086177   0.405979  -7.602 2.92e-14 ***
## TitleMrs    -14.844836 437.522339  -0.034 0.972934
## TitleOther   -3.460938   0.679286  -5.095 3.49e-07 ***
## ---
## 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:  755.39  on 879  degrees of freedom
## AIC: 779.39
##
## Number of Fisher Scoring iterations: 13
```

From the coefficients of the obtained model, we see that the coefficients for the atributes `Fare`, `Sex` and some of the categories in `Title` and `Embarked` are not significant at a 95% confidence level. We will compare this model with another without these variables.

```
m2 <- glm(Survived ~ Pclass + Parch + Embarked + Title, family=binomial(link='logit'), data=train)
summary(m2)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Parch + Embarked + Title, family = binomial(link = "logit"),
##     data = train)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5125  -0.6270  -0.3639   0.6235   2.3442
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.18197    0.47073   6.760 1.38e-11 ***
## Pclass2     -0.96862    0.27106  -3.573 0.000352 ***
## Pclass3     -2.12311    0.24648  -8.614  < 2e-16 ***
## Parch       -0.44173    0.12477  -3.540 0.000399 ***
## EmbarkedQ   -0.31693    0.38215  -0.829 0.406920
## EmbarkedS   -0.63296    0.23661  -2.675 0.007469 **
## TitleMiss   -0.06907    0.39210  -0.176 0.860174
## TitleMr     -3.10724    0.40614  -7.651 2.00e-14 ***
## TitleMrs     0.40068    0.42385   0.945 0.344483
## TitleOther  -2.97072    0.60704  -4.894 9.89e-07 ***
## ---
## 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:  763.22  on 881  degrees of freedom
## AIC: 783.22
##
## Number of Fisher Scoring iterations: 5
```

Comparing both models we can see that the second model has most of its attributes significant, except the same categories in `Title` and `Embarked` attributes. COmparing the AIC values for both models suggest that the first model is slightly better since it has a lower value of AIC.

AIC stands for Akaine Information Criterion and is a measure of the quality of a given model. This estimator balances the goodness of fit of the model with its complexity. More complex models tend to have a higher capacity of better explaining the data at the cost of overfitting it. Therefore, this estimator introduces a penalisation for higher complexity.

Since we have obtained a lower AIC value for our more complex model (it introduces more regressors), it means that this addicional complexity is compensated by a better goodness of fit.

Finally, we will try a third model from the first one just dropping the `Fare` attribute. This corresponds to the idea that in a disaster such as the Titanic, the rescue teams would prioritize women and children and, therefore, `Age` might be important even though its coefficient turned out not significative.

```r
m3 <- glm(Survived ~ Pclass + Sex + Parch + Embarked + Title, family=binomial(link='logit'), data=train
summary(m3)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Sex + Parch + Embarked + Title,
##     family = binomial(link = "logit"), data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5013  -0.6340  -0.3640   0.6155   2.3441
##
## Coefficients:
```

```
##             Estimate Std. Error z value Pr(>|z|)    
## (Intercept)  18.4133   437.2343   0.042 0.966409    
## Pclass2      -0.9270     0.2730  -3.396 0.000685 ***
## Pclass3      -2.1058     0.2468  -8.533  < 2e-16 ***
## Sexmale     -15.2634   437.2341  -0.035 0.972152    
## Parch        -0.4368     0.1246  -3.505 0.000457 ***
## EmbarkedQ    -0.2901     0.3836  -0.756 0.449450    
## EmbarkedS    -0.6313     0.2385  -2.647 0.008115 ** 
## TitleMiss   -15.3298   437.2342  -0.035 0.972031    
## TitleMr      -3.0938     0.4061  -7.619 2.55e-14 ***
## TitleMrs    -14.8616   437.2343  -0.034 0.972885    
## TitleOther   -3.5014     0.6786  -5.160 2.47e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 1186.7  on 890  degrees of freedom
## Residual deviance:  756.0  on 880  degrees of freedom
## AIC: 778
## 
## Number of Fisher Scoring iterations: 13
```

Indeed it turns out to be a slightly better model according to the AIC values obtained. We will therefore select the third model to make the final prediction on the test set. Before that, we will inspect the significant coefficients of the model to gain some insight on how they affect the response variable.

- Being of a lower class reduces the chances of survival so passengers of class 1 are expected to survive more than passengers of class2 and even more than passengers of class 3.
- Having more parent/children reduces the chances of survival. Since this attribute will be at most 2 for children (they can at most have their 2 parent aboard) while parents of large families will have larger values of this attribute, it is expected that parents of large families will have less chances of survival.
- Passengers who embarked from Southampton also have lower chances of survival. It could be explored if those passengers have different demographics as others.
- Having a title of Mr. or Other. reduces chances of survival. This is in line with the fact the women survived more than man. As for the Other class, there were a few passengers with this class and it might be interesting to explore if they also had different demographics as other passengers.

Finally, we will use the third model to make predictions on the test set.

```r
y_pred <- predict(m3, test, type="response")
head(cbind(test,y_pred))
```

```
##     Pclass    Sex  Age SibSp Parch    Fare Embarked Title FamilySize
## 892      3   male 34.5     0     0  7.8292        Q    Mr          1
## 893      3 female 47.0     1     0  7.0000        S   Mrs          2
## 894      2   male 62.0     0     0  9.6875        Q    Mr          1
## 895      3   male 27.0     0     0  8.6625        S    Mr          1
## 896      3 female 22.0     1     1 12.2875        S   Mrs          3
## 897      3   male 14.0     0     0  9.2250        S    Mr          1
##         y_pred
## 892 0.08787450
## 893 0.69307856
## 894 0.23847482
## 895 0.06410067
## 896 0.59332957
## 897 0.06410067
```

From the predictions we can see that passenger 892 has a predicted probability of survival of less than 9% while passenger 893 has almost 70% chances of survival. To determine the predicted class we we will use a probability of 50% as a threshold to determine which passengers we think survived. We can finally make a prediction of the test set classes.

```
y_test <- as.factor(ifelse(y_pred > 0.5, 1, 0))
submission <- data.frame(PassengerId = test_id, Survived=y_test)
write.csv(submission, "../data/submission/submission.csv", row.names=F, quote=F)
```

5. Conclusion

In this report we have explored the association of the different attributes with the survival chance of passengers of the Titanic using the dataset from the Kaggle competition Titanic: Machine Learning from Disaster which has helped in building logistic regression models to predict the class for unseen data.

Our analysis determine that all variables except `Age`, `SibSp` and `FamilySize` significantly associated with `Survived` at a 95% confidence level. Furthermore, the model that uses the attributes `Pclass`, `Sex`, `Parch`, `Embarked` and `Title` as regressors has been the chosen model to predict with according to resulting AIC values amongst the tested models.

This model allows us to explore the relationships between regressors and the target attribute as well as predict probability of survival for passengers. In particular we have detemrined that lower classes, large families, passengers with a title of *Mr* or *Other* and passengers who embarked at Southampton have a significantly lower chance of survival, according to those coeficients from the regression model that turned out to be significant.

Finally, we have used this model to label the instances from the test dataset with their predicted class values.

6. Resources

1. Peter Dalgaard (2008). Introductory statistics with R. Springer Science & Business Media.
2. Jiawei Han, Micheine Kamber, Jian Pei (2012). Data mining: concepts and techniques. Morgan Kaufmann. Chapter 3.
3. Jason W. Osborne (2010). Data Cleaning Basics: Best Practices in Dealing with Extreme Scores. Newborn and Infant Nursing Reviews; 10 (1): pp. 1527-3369.
4. R Documentation. Shapiro-Wilk Normality Test. http://finzi.psych.upenn.edu/R/library/stats/html/shapiro.test.html.
5. R Documentation. Levene's Test. http://finzi.psych.upenn.edu/R/library/car/html/leveneTest.html
6. R Documentation. Test for Equal Means in a One-Way Layout. http://finzi.psych.upenn.edu/R/library/stats/html/oneway.test.html
7. R Documentation. Pearson's Chi-squared Test for Count Data. http://finzi.psych.upenn.edu/R/library/stats/html/chisq.test.html