Predicting Survival on the Titanic

Andrew Rosa

12/10/2016

Background

The Titanic was the largest ship to have been built when it had entered service. For it's maden voyage it left the port of Southhampton England, made stops in Cherbourg France and what was known at the time as Queenstown Ireland, now Cobh then set it's site on America. In the early morning, of April 12th, 1912 the RMS Titanic colided with an iceberg in the North Atlantic Ocean. The damage caused it to sink in about a 2 hour timespan. Over 2,200 passengers were estimated to be on board, 1,502 of them died in the disaster.

This project seeks to explore a data set of the passengers who were recorded to be onboard the Titanic to create a predictive model of passanger's fate.

First

We'll need to load in the data. The titanic dataset is on CRAN(Comprehensive R Arichive Network). To obtaine it we simply need to run `istall.packages("titanic")`, following this by `library(titanic)` to load it into our global enviroment. The package not only contains the full titanic dataset, but a training sample and a test sample set as well. For now I'll load in the training sample as a dataframe called train_df. Let's take glance at the data.

```
str(train_df)
```

```
## '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)"
##  $ 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  "" "C85" "" "C123" ...
##  $ Embarked   : chr  "S" "C" "S" "S" ...
```

Formatting

Right off the bat we can spot a few formatting discrepencies that we should address before doing further analysis. Notice that the Survive and Pclass(Passanger's Class) variables have numbers. These numbers actually represent catigorys. 0 marks died and 1 marks survived for the Survive variable. The 1, 2, and 3 represent first, second, and third class respectivly. Sex and Embarked currently have a character class and should be made catigorigal as well. We'll make them have a factor class instead, so that each variable has a level determined by catigory.

```
var_names <- names(train_df)
var_factors <- c(var_names[1:3], var_names[5], var_names[11:12])
for(i in var_factors){
        train_df[, i] <- factor(train_df[, i])
}
```

Now that the formatting is fixed lets call a summary on the data frame to get an idea of some basic statistics.

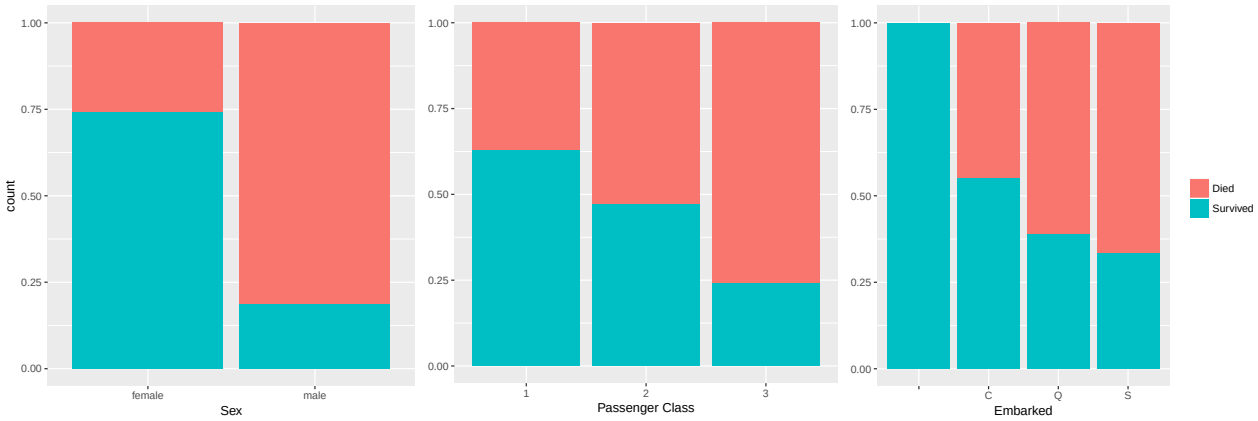
```
summary(train_df)
```

```
##   PassengerId  Survived Pclass      Name              Sex
##  1      :  1   0:549    1:216   Length:891         female:314
##  2      :  1   1:342    2:184   Class :character   male  :577
##  3      :  1            3:491   Mode  :character
##  4      :  1
##  5      :  1
##  6      :  1
##  (Other):885
##       Age            SibSp           Parch           Ticket
##  Min.   : 0.42   Min.   :0.000   Min.   :0.0000   Length:891
##  1st Qu.:20.12   1st Qu.:0.000   1st Qu.:0.0000   Class :character
##  Median :28.00   Median :0.000   Median :0.0000   Mode  :character
##  Mean   :29.70   Mean   :0.523   Mean   :0.3816
##  3rd Qu.:38.00   3rd Qu.:1.000   3rd Qu.:0.0000
##  Max.   :80.00   Max.   :8.000   Max.   :6.0000
##  NA's   :177
##       Fare                Cabin     Embarked
##  Min.   :  0.00             :687    :  2
##  1st Qu.:  7.91   B96 B98   :  4   C:168
##  Median : 14.45   C23 C25 C27:  4   Q: 77
##  Mean   : 32.20   G6        :  4   S:644
##  3rd Qu.: 31.00   C22 C26   :  3
##  Max.   :512.33   D         :  3
##                   (Other)   :186
```

We see here that out of 891 of the tottal passengers in this sample that 342 passengers survived(38.4%). We also see that there is some missing data. For now we wont worry about it, but it's good to take note of.

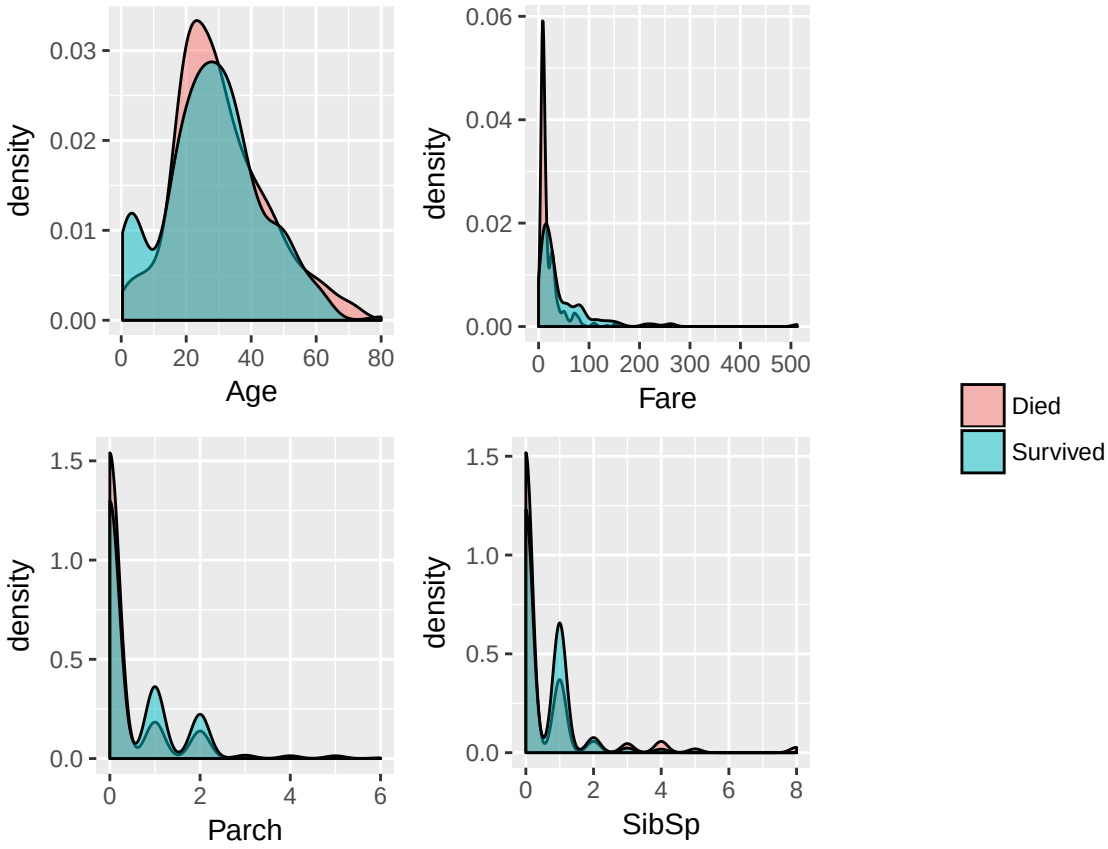
Exploritory Visualizations

Now let's use visualizations to take a look how different variables compare to the Survived variable. We'll start with examining a few catigorical variables. In these first three visualizations we can see the porportion of which passengers survived determind by Sex, Passenger Class, and City of Embarkment with the use of histograms. The sex chart shows that if you were female you had a significantly better chance of living than if you were male. The passenger class also shows someone of wealth having a better chance of surving. We see that those who embarked from Cherbourg had a better chance of surviving than those who embarked from the other two ports. A few quick calculations show that 50.6% of the passengers from Cherbourg are part of the wealthiest class. A much higher concentration than passangers who embarked from the other ports, which helps to explaine the correlation.



Next we'll look at some of the numeric data with the use of density plots. Taking a look at the first density plot, you can easily see a noticable difference in the amount of survivers verses the amount of passengers that died below the age of 15. Clearly Age will help as an indicator for our predictions. The amount of fare payed for a ticket to board the ship seems to be a good indicator as well. The last two varaibles visualized here are Parch, and SibSp. These two vairables represent family aspects. Parch represents the number of parents and children of the on board with the passanger. The SibSp represents the number of siblings and spouses are on board with the passanger.

```
plot_hidden <- ggplot(train_df, aes(x = SibSp, fill = factor(Survived, labels = c("Died", "Survived"))))
  geom_density(alpha = 0.5) +
  theme(legend.title = element_blank())
```

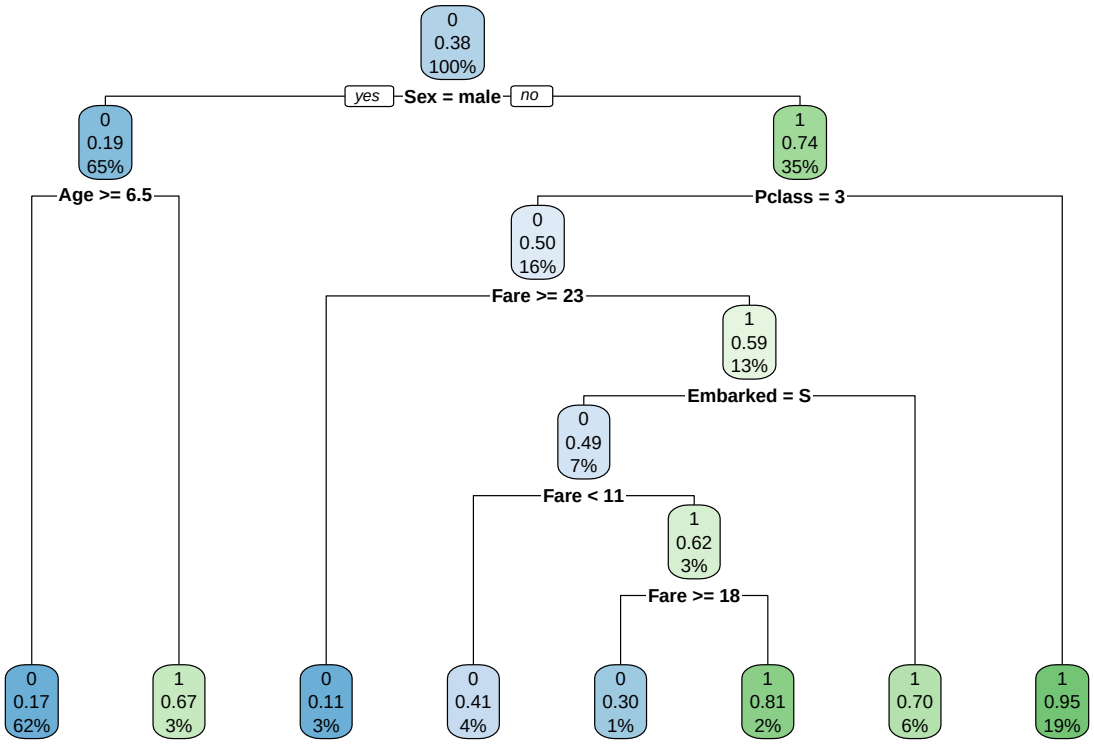


Predictive Modeling

Now that we have a good understanding of what the data looks like lets create a model to predict if a passenger died or survived the sinking of the Titanic. Since we are creating a model to predict a catigory as oppose to a quanititative measurement we'll use a tree model. The tree model we'll use in preticular is a recursive partition model. A recursive Partition model splits the data into subsets based on the independent variables provided to calculate the probibilty of an occoance hapening within each subset. The partitioning contitnues the subsetting process untill it reaches an appropraite solution. We'll use the Rpart package to create the model.

```
library(rpart)
model <- rpart(Survived ~ Fare + Sex + Age + Pclass + Embarked + Parch, data = train_df, method = "clas
```

```
library(rpart.plot)
rpart.plot(model)
```



From the visualization of the model above, we can see the likely hood of surviving given certian conditions. For example we see that if you are a male over the age of six and half years old you would have a high chance of dieing(83%). On the opposite end we see that if you are female and not part of the lowest passenger class you had a 95% chance of surviving. At this point we infer that this model created from a sample is represintitive of the overall data set, and can be used to make predictions on a sample with the survival variable missing. Now it's time to test or cross validate the model to get a sense of how accurate it is.

Model Testing

The cross validation process will require us to use a different sample to test our model on. Luckily the titanic package already has a set for testing made, we just have to load it in, and make the same formating changes as we did earlier on.

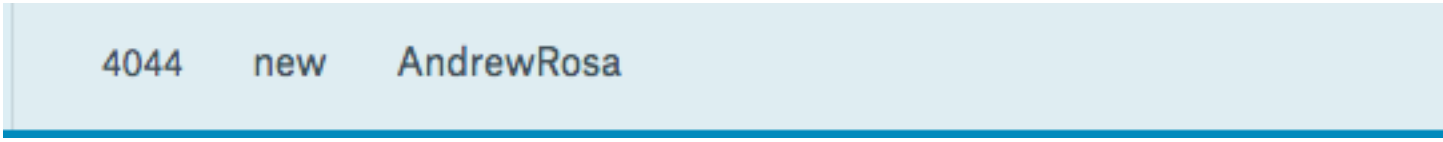
```
test_df <- titanic_test
var_factors2 <- c(var_names[1], var_names[3], var_names[5], var_names[11:12])
for(i in var_factors2){
        test_df[, i] <- factor(test_df[, i])
}

prediction <- predict(model, newdata = test_df, type = "class")

solution_1 <- data.frame(PassengerId = test_df$PassengerId, Survived = prediction)
write.csv(solution_1, file = "solution_1.csv", row.names = FALSE)
```

Normally from here we would create a confusion matrix, and calculate the accuracy of the prediction model. We would need to have the actual Survival data for the testing sample, but because this data is provided by on CRAN by Kaggle.com for a contest they held we can't do this. We can still upload the model to Kaggle.com to see the accuracy of the model though. The accuracy of this model is recorded as 77.5%.

```
library(knitr)
include_graphics("/Users/andrewrosa/Desktop/Titanic_Project/image_1.png")
```



Edit Model/ Test again

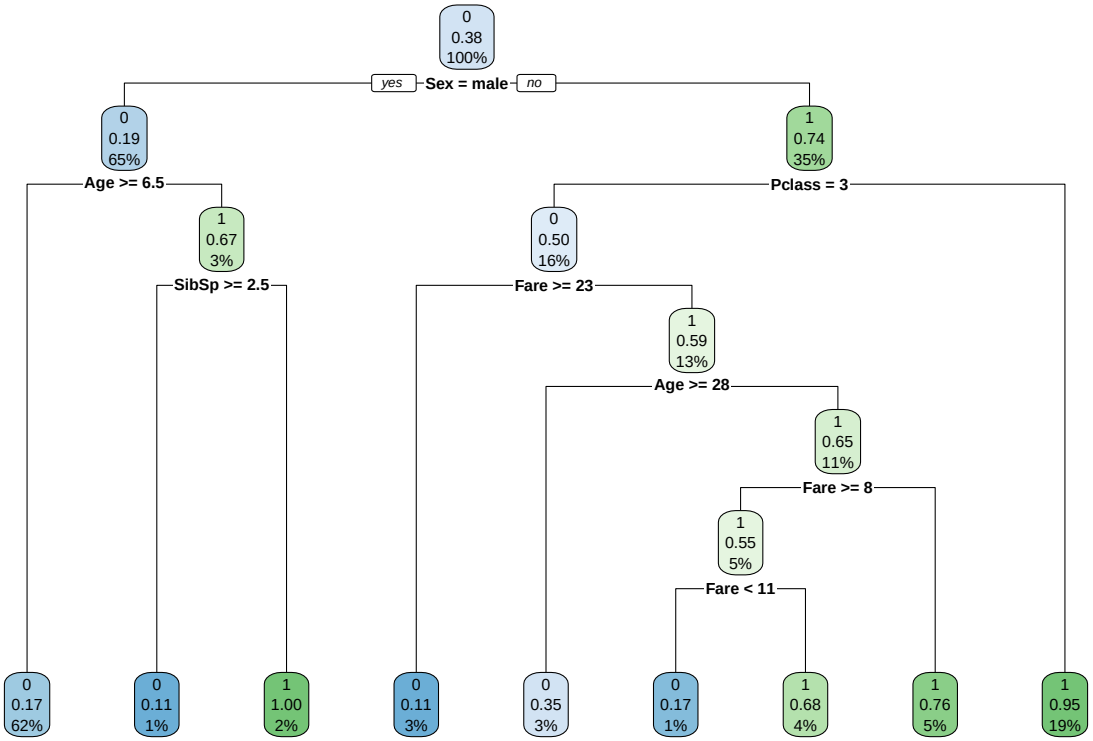
This first model is pretty basic, with some work the accuracy can be improved. The rpart function by defualt excludes any observations with missing data (ie "NA"). By doing this the sample size that the model is based on is smaller. Increasing that sample size would hopefully help make a better prediction. The Age variable has 177 NA's, so what do we do to replace them? There are a few different options, such as just using the average age, or making a linear regression madel that predicts age from a sample that excludes the NA's and then replacing the NA's with the prediction. Another method that I will use, is to choose a recursive partition model to predict age.

```
age_model <- rpart(Age ~ Pclass + Sex + SibSp + Parch + Fare + Embarked, train_df[!is.na(train_df$Age),
train_df$Age[is.na(train_df$Age)] <- predict(age_model, newdata = train_df[is.na(train_df$Age),])
```

We can also engineer additional variables ourselves that would help further define the recursive partition model. An easy variable to creat would be the size of the family for each passenger by combining the parent/child variable with the sibling/spouse variable and adding 1 to account for the said passenger.

```
train_df$FamilySize <- train_df$SibSp + train_df$Parch + 1
model_2 <- rpart(Survived ~ Fare + Sex + Age + Pclass + Embarked + Parch + SibSp + FamilySize, train_df

rpart.plot(model_2)
```




```
test_df$FamilySize <- test_df$SibSp + test_df$Parch + 1
prediction_2 <- predict(model_2, newdata = test_df, type = "class")
solution_2 <- data.frame(PassengerId = test_df$PassengerId, Survived = prediction_2)
write.csv(solution_2, file = "solution_2.csv", row.names = FALSE)
```

Conclusion

Unfortanitly this solution did not improve the model providing again an accuracy of 77.5%. This is just the start to creating a simple recursive partision model though. The next steps would be to further re-engineer the data by analizing the name variable for titles like Dr, or Mrs, and turning that data into a seperate variable of it's oown. Adjusting the fit of the model by setting the number of splits and size of leafs could possibly help as well. Further cross validation would be needed to figure which model would produce the most accurate results.