Titanic: Machine learning from disaster

Sami Chakroun – 201305849

Data Mining Project – Final Report

Dr. Georges Khazen

Fall 2016

Table of Contents

Abstract

This report explains the findings of my data analysis performed on the Titanic dataset from Kaggle.com. All the work was done using RStudio. The source code is available on Github on this link: https://github.com/SamiChakroun/TitanicMachineLearning

60% of the work is focused on exploring the data, cleaning it and understanding the relations between the different features we have the outcome ("Survived").

The remaining is building models, cross validating them and submitting the results to Kaggle for validation. We build a Random Forest model and compare its performance to a Logistical Regression model.

Introduction

Background information

In 1912, the Titanic sank after colliding with an iceberg, killing 1502 out of 2224 passengers and crew. One of the reasons that the shipwreck led to such loss of life was that there were not enough lifeboats for the passengers and crew. Although there was some element of luck involved in surviving the sinking, some groups of people were more likely to survive than others, such as women, children, and the upper-class. (Kaggle.com, 2016)

In this project, I will be analyzing the dataset available on the Kaggle website using RStudio in order to build the most accurate predictive model possible. I chose to work in RStudio since it provides a better working environment than the R console where you can see all your variables, explore your data, and navigate between plots. It also makes source control much easier to handle.

This project was not only an opportunity to apply the knowledge we acquired throughout the semester in our Data Mining class, but also, to explore some more advanced tools and techniques that we briefly mentioned in class such as the "ggplot2" library for plotting or the Random Forests modelling for classification.

The dataset

The dataset is available for download from Kaggle.com and is also available on the project main folder. The data is split into two files "train.csv" and "test.csv". The Kaggle website also provides the meta data shown below which appears to be particularly useful in understanding the dataset we're working with.

VARIABLE DESCRIPTIONS:

survival Survival

(0 = No; 1 = Yes)

pclass Passenger Class

(1 = 1st; 2 = 2nd; 3 = 3rd)

name Name

sex Sex

age Age

sibsp Number of Siblings/Spouses Aboard

parch Number of Parents/Children Aboard

ticket Ticket Number

fare Passenger Fare

cabin Cabin

embarked Port of Embarkation

(C = Cherbourg; Q = Queenstown; S = Southampton)

SPECIAL NOTES:

Pclass is a proxy for socio-economic status (SES)

1st ~ Upper; 2nd ~ Middle; 3rd ~ Lower

Age is in Years; Fractional if Age less than One (1)

If the Age is Estimated, it is in the form xx.5

With respect to the family relation variables (i.e. sibsp and parch)

some relations were ignored. The following are the definitions used

for sibsp and parch.

Sibling: Brother, Sister, Stepbrother, or Stepsister of Passenger Aboard Titanic

Spouse: Husband or Wife of Passenger Aboard Titanic (Mistresses and Fiances Ignored)

Parent: Mother or Father of Passenger Aboard Titanic

Child: Son, Daughter, Stepson, or Stepdaughter of Passenger Aboard Titanic

Other family relatives excluded from this study include cousins, nephews/nieces, aunts/uncles, and in-laws. Some children travelled only with a nanny, therefore parch=0 for them. As well, some travelled with very close friends or neighbors in a village, however, the definitions do not support such relations.

Expected outcomes

Analyzing this dataset is more of a learning experience than a mean to uncover something new.

However, we dive into the process expecting to:

- find out which people were more likely to survive.
- Predict which passengers survived the tragedy by applying Machine Learning tools (using predictive models).

Anyone who watched the movie "Titanic" would go into this process with some expected results such as the fact that 1st class passengers were more likely to have survived than 2nd or 3rd class passengers. Also, since women and children were prioritized to embark on the lifeboats, they are more likely to have been saved. The exploratory analysis should show if these assumptions are correct or not.

How to use this document?

This document is an explanation of the steps performed and the interpretation of the results of each phase of the data analysis process achieved using RStudio.

Each section has a corresponding script file with comments explaining what we did and how we got the results we are looking at.

IMPORTANT: You can run all scripts in order except for "step4.R" which should be run before "step3.R".

Getting Started

Loading the data

Our data is split into train and test sets which are in separate comma separated files (.csv).

After running the code in "step1.R", we can start looking at the data which we loaded into two data frames "test" and "train".

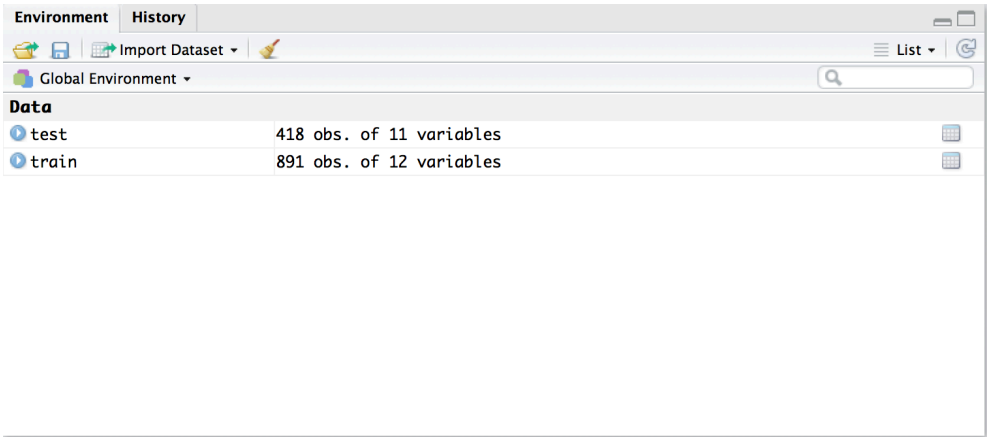
The train data frame contains rows having passenger id values from 1 till 891. The test dataset contains elements with passenger id from 892 till 1309. The outcome is given by the variable "Survived" which can take a value of 0 for died or 1 for survived.

However, as the snapshot below shows, we found out that the "test" data doesn't have the outcome "Survived" since it will be only used for making predictions.

In order for us to look at the whole data, we need to combine both data frames into one. But, before doing so, we change the row names of test to match the passenger id column. Then, we remove the passenger id column from both test and train since row.names is now representing it.

Now, we create a new test.survived data frame with an extra column for the feature "Survived" having all "NA" values.

Finally, using the "rbind()" function we combine the two data frames having matching columns.



We can see that in the data.combined data frame, we now have values of "1" or "0" for all rows before 892. The rest have a value of "NA".

Structure and Summary statistics

Now that we have our data combined, we start looking at some summary statistics using the functions "summary()" and "str()". For that, run the code in "step2.R".

We start by looking at the head and tail of data.combined to see what kind of data we're dealing with. We have some striking observations from here. The "Name" is in a complex form but there seems to be a title always showing up such as "Mr.". Also, "Ticket" seems like it is not well structured and it is not something we could possibly make sense out of. Finally, "Age" and "Cabin" have many missing values from what we see in head and tail.

Now, we check the structure of our data frame using the str() function. Weirdly, the "Survived" feature is of type character but we would like it to be of type factor. We use the as.factor() function to change the type of this feature to factor. The feature "Pclass" should also be of type factor so perform the same action.

Now that the types of features are good, we move to checking the summary statistics of our data. The summary() function gives us a clearer idea about our dataset now. We can see that from the train data 549 passengers died and 342 survived. The summary also tells us that we have 466 females and 843 males in our data frame.

In addition to that, summary() tells us that we have 263 NA values in "Age", 1 in "Fare" and the 418 NA's we created in "Survived" for the test data. The NA's in Cabin do not show as NA's since they have an empty string value.

The most interesting observation in the summary statistics is that we found two duplicates in "Name" for our 1309 observed rows. So let's see if these are truly duplicates or not.

Cleaning the data

Since we already know there are duplicates in "Name", let's check if these are really duplicates.

The two names seem to correspond to different persons since they have different ages and ticket numbers.

Our biggest problem when it comes to cleaning the data is the number of missing values in the variable "Age" which is about 1/5 of the total data. We cannot simply infer this value by replacing it with a mean or median since that would have a huge impact on our prediction. Instead we should find a better way of estimating the age for each passenger with a missing age value. We will do that based on the names since these have interesting "titles" in them.

After looking at quantitative variables from the summary, our data doesn't seem to have any outliers. Except for "Fare" has a max of 512 while the average and median are around 31 ~ 33. We won't consider this an outlier since it is possible that this person actually paid that amount for a luxurious trip. Also, in our case every row matters so we don't want to remove data that could have a huge impact on our predictions.

Extracting features

When looking at the "Name" column of our data, we noticed that each name has a title "Mr.", "Mrs.", "Master.", or "Miss" and other titles that are less abundant.

This title could be very meaningful and have some predictive power to whether the person survived or not since it is indirectly implying the age and gender.

Therefore, we decided to extract this part of each name and make it as a feature by itself. The following snapshot was taken in the middle of the process of extraction of titles.

```
> summary(data.combined$Title)
   Col.     Dr.  Major. Master.   Miss.    Mme.     Mr.    Mrs.      NA    Rev.
      4       8       2      61     260       1     758     199       8       8
```

The 8 NA values happen to correspond to very specific titles such as "Jonkheer", "Don", or "Dona" which we inserted within the main categories "Mr" or Mrs".

Going back to our meta data, we see that we have two variables that could be more predictive if merged together. These variables are the sibsp (Number of Siblings/Spouses Aboard) and parch (Number of Parents/Children Aboard).

These variables should give us a good approximation of family size which would impact the chances of survival a lot. In case of an emergency, you would certainly expect a family to stick together and try to save each other. We will therefore create a family size variable following this approach:

Family size = (SibSp + Parch + 1) #1 for the person considered itself. By plotting this feature with respect by Pclass we can see that there is a pattern where big families of 3rd class with more than 5 people tend to perish.

We see how this result is achieved in “script4.R”.

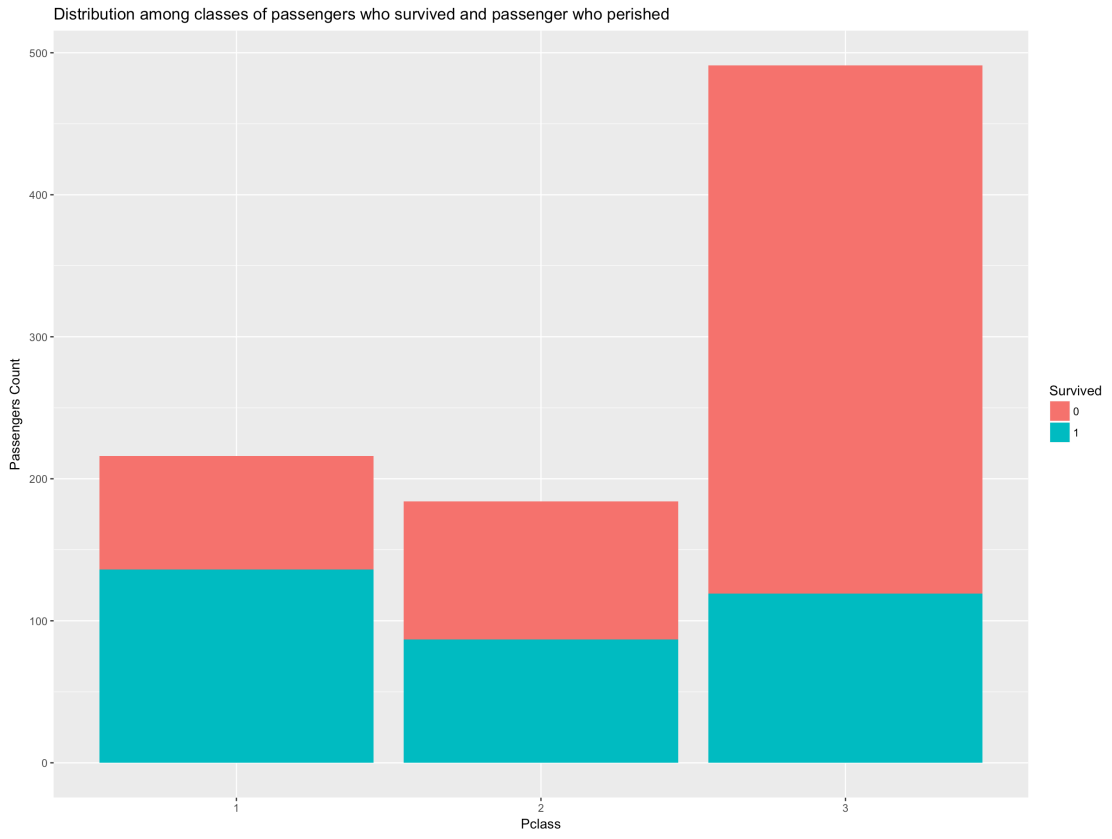
Exploratory Analysis

Plots with "ggplot2"

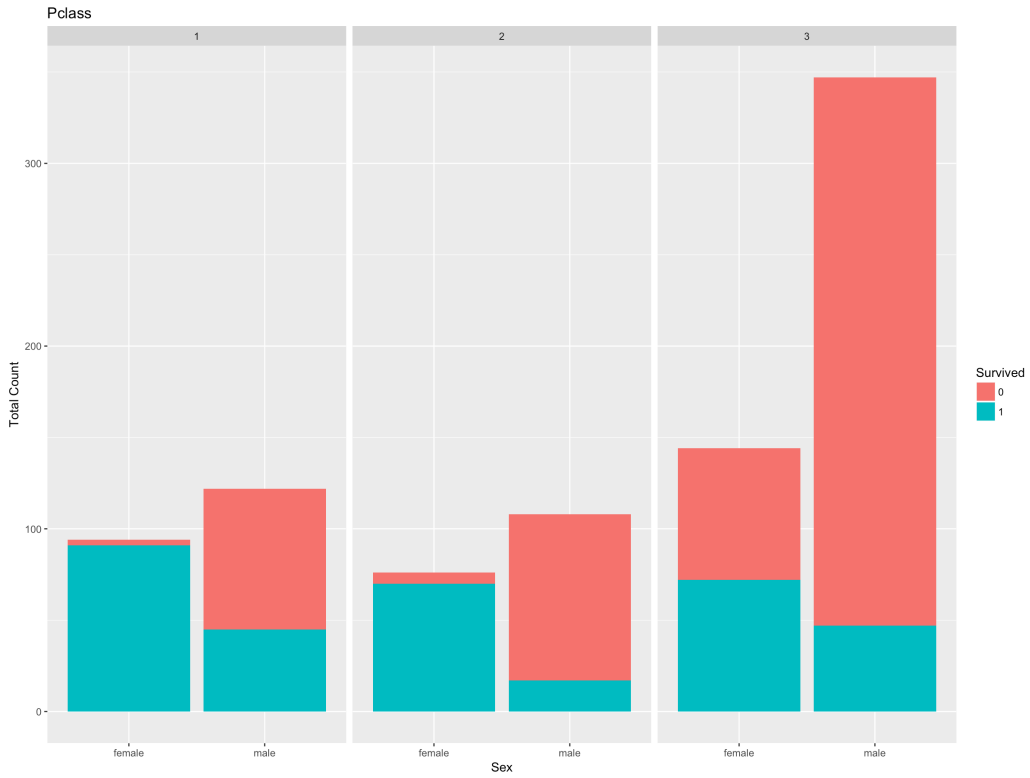
1. The first plot shows the distribution among classes of passengers who survived and passengers who perished.

Interpretation:

- We can clearly see that first class passengers were very likely to survive (around 2/3 survive)
- Second class passengers had around the same chances of surviving or perishing.
- In third class, things get more interesting as the proportion of passengers that die is way higher than those who survive.



2. Now let's try to do a similar plotting but by dividing the passengers based on gender. Here is what we get.

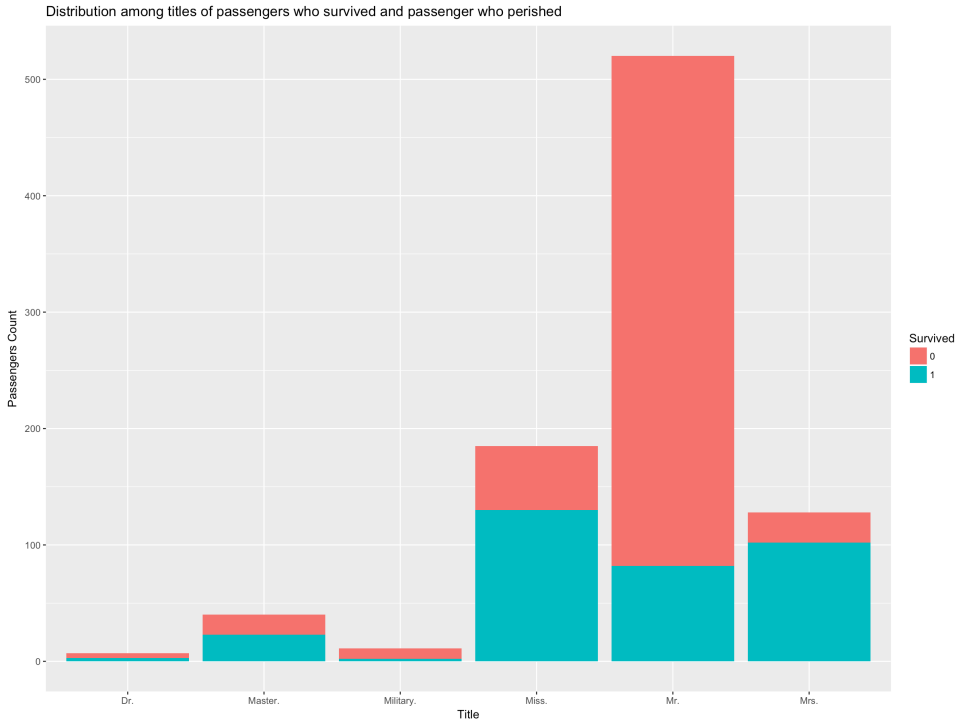


This plot definitely proves our hypothesis on women being prioritized to be saved though this distribution varies from class to class.

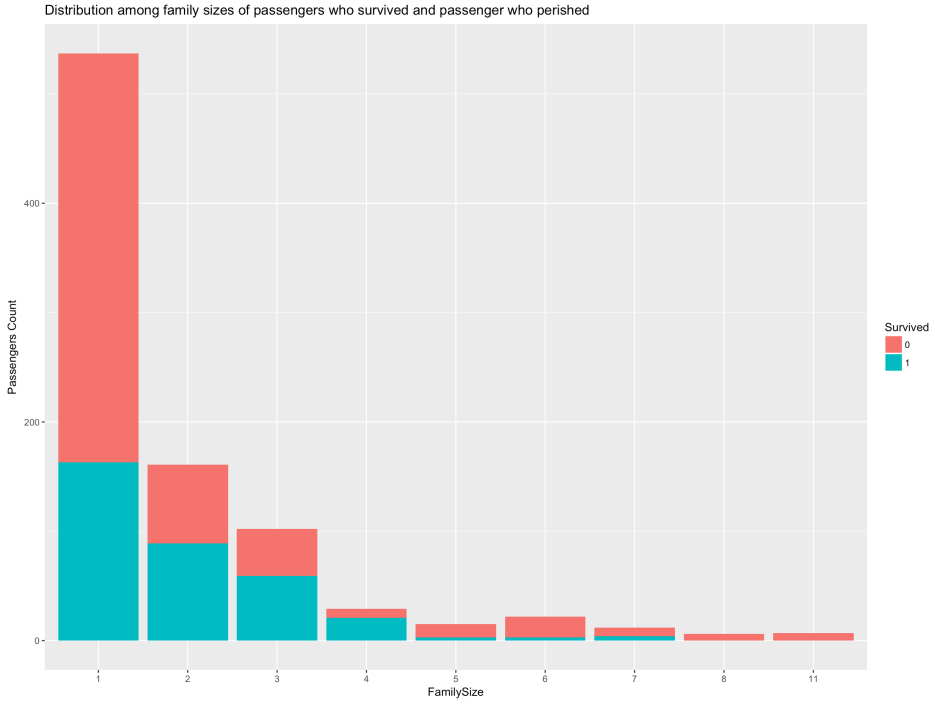
Interpretation:

- Females in first class were very unlikely to die. Same for females in second class. In third class things are different since they have almost a 50/50 chance of survival.
- Males in first class have around 1/3 chance of survival which is much better than the chances of men surviving in second or third classes.

3. Distribution among titles of passengers who survived and passenger who perished:



4. Distribution among family sizes of passengers who survived and passenger who perished:



Why not PCA, Hierarchical Clustering or K-means clustering?

In this project, we care a lot about interpretability of results. We didn't go directly into exploratory modeling since we don't have that many features.

It would have been good to use PCA on a big dataset with many quantitative predictors. PCA is mainly used with qualitative variables that would have some correlations in between them. In our data that is not the case.

We tried performing Hierarchical Clustering but since most of the features are qualitative the results do not really make much sense. Plus, with 891 rows on our train data, the dendrogram in very hard to interpret.

Exploratory Analysis Results

The Pclass feature seems to be the most significant factor for survival on the Titanic data.

Gender is also very significant since females have much higher survival rates than men.

Age is probably going to lead to overfitting since after we replaced the NA values with the medians of their categories we got a very high peak at age 29 predicted to perish.

Family size has an impact on survival since large families will likely die together while people traveling alone (Family size = 1) have a higher chance of survival.

Subset Selection

Feature Selection

We start by doing some cleaning to our data. We remove the features that have unorganized data and that will not have an impact on predicting survival.

Ticket for example is a ticket number that doesn't seem to follow any specific format plus it clearly has missing values.

Cabin also has a lot of missing values therefore we will omit it.

As for name, we already extracted the meaningful part of it which is the title. Therefore, we can now omit the full name.

We will use regsubsets() for best subset selection from our clean data as show in "step6.R".

This is what our data looks like now:

```
> str(data.clean)
'data.frame':   1309 obs. of  10 variables:
 $ Survived  : Factor w/ 3 levels "0","1","NA": 1 2 2 2 1 1 1 1 2 2 ...
 $ Pclass    : Factor w/ 3 levels "1","2","3": 3 1 3 1 3 3 1 3 3 2 ...
 $ Sex       : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
 $ Age       : num  22 38 26 35 35 29 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 ...
 $ Fare      : num  7.25 71.28 7.92 53.1 8.05 ...
 $ Embarked  : Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
 $ Title     : Factor w/ 6 levels "Dr.","Master.",..: 5 6 4 6 5 5 5 2 6 6 ...
 $ FamilySize: Factor w/ 9 levels "1","2","3","4",..: 2 2 1 2 1 1 1 5 3 2 ...
>
```

The results of best subset selection confirm our assumptions but also are a bit surprising.

The best features seem to be by far Pclass and Title. Gender also is significant but not that much which is understandable since Title gives an idea about the gender.

SibSp is surprisingly significant. But we see that FamilySize is even more significant.

The plotting suggests that we should change our FamilySize feature in the following way: FamilySize should be a factor of 3 levels.

Passengers traveling alone (FamilySize=1)
Passengers of small family size (FamilySize <= 4)
Passengers of big family size (FamilySize>=5)

This is due to the fact that larger families tend to perish together.

Cross validation

In order to make our results more accurate, we need to perform cross validation to choose the best subsets. Unfortunately, we couldn't achieve this.

Building Models

Random Forests

Building different models

In "step7.R" we create multiple Random Forest models and compare them in order to select our Features.

The purpose of this step is to validate our assumptions about the importance of certain features for predicting the survival of passengers. This is especially important for the features that we engineered.

Cross Validation of best models

In order to perform Cross Validation on our Random Forest model we will use the Caret package. The detailed implementation is explained in "step7.R".

The cross validation performed on the Random Forest model gives us about the same accuracy as seen before therefore we conclude that we don't have much overfitting.

If we had a huge difference in accuracy that would indicate that we are overfitting our data.

However, when we submit the results of our predictions for validation on Kaggle, we see that the difference between our expected accuracy and the actual accuracy is huge (around 6% accuracy difference). Therefore, we actually have overfitting.

Logistical Regression

Building the model

We now build a Logistic Regression model to predict survival. The results are not bad but also not as good as Random Forest.

The implementation of logistical regression is shown in "step8.R".

Support Vector Machine

Building the model

Using SVM is good in this case since it provides greater robustness to individual observations, and better classification of most of the training observations.

Unfortunately, since our model has many features, we couldn't do the plotting and understand our model to better tune it.

The detailed implementation is explained in the script "step9.R".

Results and possible improvements

Kaggle submissions

We made 4 different submissions to Kaggle in order to validate our models. The score in Kaggle represents the accuracy of the prediction we made on the train dataset. The results are as follows:

1. Random Forest Model 1
    a. This submission gave a score of 0.77990.
    b. 77.99% is different from our accuracy we predicted from the training data.
    c. This means we have some overfitting in our modelling.
2. Random Forest Model 2
    a. This submission improved the score by 0.00957 simply by adding the feature Age to our model.
    b. The score is now 0.78947
3. Logistic Regression Model
    a. This submission scored 0.77512
    b. Logistic regression is fast and gives a result close to the one we got using our first random forest.
4. Support Vector Machine Model
    a. This submission gave a score of 0.78469
    b. This score is very close to what we obtained using random forests
    c. Possibly more tuning could result in a better model and better predictions.

Possible improvements

We saw that Random Forest gives an approximate accuracy of 83%. However, Logistic Regression performs really badly with an accuracy of around 40%.

Random Forest seems to be the most suitable model for this type of data. Cleaning the data and engineering features more precisely would probably be the best way to improve the accuracy of our predictions.

Conclusion

The data will eventually speak to you when you spend enough time digging into it. That is exactly what happened since we proved based on this data the following assumptions:

- Women and Children were survived in bigger proportions than Men.
- 3rd class passengers perished in bigger proportions than 2nd and 1st class passengers.
- We can predict whether a person survived or died based on simple information such as the title of the person, the class they were in, their gender, age, and size of their family.

Even though the results are not very convincing and certainly could be achieved in better ways, this project was a wonderful learning experience. It was a great opportunity to apply the theories and knowledge we have been acquiring throughout the semester.

Resources

1. https://www.kaggle.com/c/titanic/data
2. https://www.kaggle.com/c/titanic/forums
3. https://www.youtube.com/watch?v=eKD5gxPPeY0
4. http://datascienceplus.com/perform-logistic-regression-in-r/
5. http://docs.ggplot2.org/0.9.3.1/geom_point.html
6. http://www.cookbook-r.com/Graphs/Scatterplots_(ggplot2)/
7. http://ahmedbesbes.com/how-to-score-08134-in-titanic-kaggle-challenge.html
8. https://cran.r-project.org/web/packages/ggdendro/vignettes/ggdendro.html