Giovanni Briggs
4/11/15
CSCI 183

HW2 – Titanic Predictions

Overview

This assignment asked us to predict who survives the Titanic based on known factors such as age, fare, gender and socio-economic status.

My first solution attempted to use a cosine similarity model to predict who survived. The idea behind the solution is that similar passengers should be scored similarly. Passengers are evaluated as vectors, and if the angle between two vectors determines their similarity. The close the angle is to 1, the more similar they are. This solution scored a 0.65 on Kaggle, but took almost 20 minutes to execute. The code for it is saved in *hw2.r*, and its predictions are saved in *CosineSimilarityModel.csv*

The second solution uses the *glm()* function in R to create a linear regression model of the training data. It takes that model and uses the *predict()* function to get the probability of survival in the test data. The code for it is saved in *hw2_glm.R* and the predictions are saved in *RoundedModel.csv*.

Formatting Data

This part will only examine the solution created by using the linear regression model to predict survival. This solution also is based on the exploratory data analysis in these links:

https://github.com/wehrley/wehrley.github.io/blob/master/SOUPTONUTS.md

http://www.anesi.com/titanic.htm

Using the *glm()* function is easy and fast, however, it produces a lot of NA values due to missing data in the test dataset. The model I used is as follows:

```
>>>  fit <- glm(Survived ~ Pclass + Sex + Age + SibSp + Parch + Fare,
                    data = df.train, family=binomial)
```

This uses the training data to get the chances of survival based on a series of factors. These factors were all chosen because they easily translate to numeric values. Factors such as a passenger’s cabin are not so easy to convert into numeric values that can be easily compared.

We then call:

```
>>> survived.glm <- predict(fit, df.test, type='response')
```

This gives us the probability of survival for each passenger in the test dataset, but the matrix is not complete. About 20% of the matrix contains NA values, which means that we have no prediction for 20% of the passengers in the training set. This NA value arises because of NA values in the test dataset. About 20% of all ages in the test dataset are NA.

In order to solve this, we can fill the NA values with the mean of all non-NA values. This will stop the predict function from producing NA values.

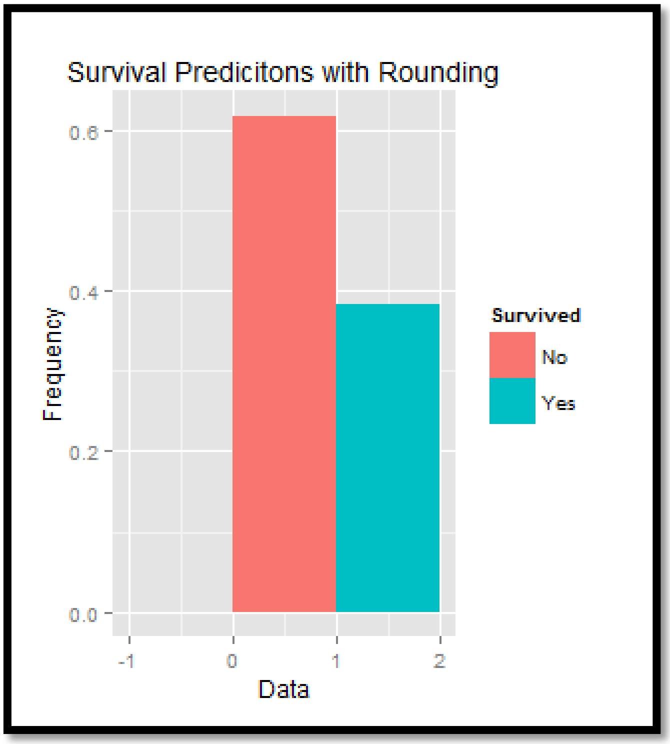
After doing this, there is only 1 NA value left I the predicted values. Passenger 1044 is the only passenger in the test set for which the fare value is NA. We do know that this passenger is in Pclass 3, which means he is in the lower socio-economic group. We can fill his NA value with the average fare of all other Pclass 3 passengers (which turns out to be 12.45968).

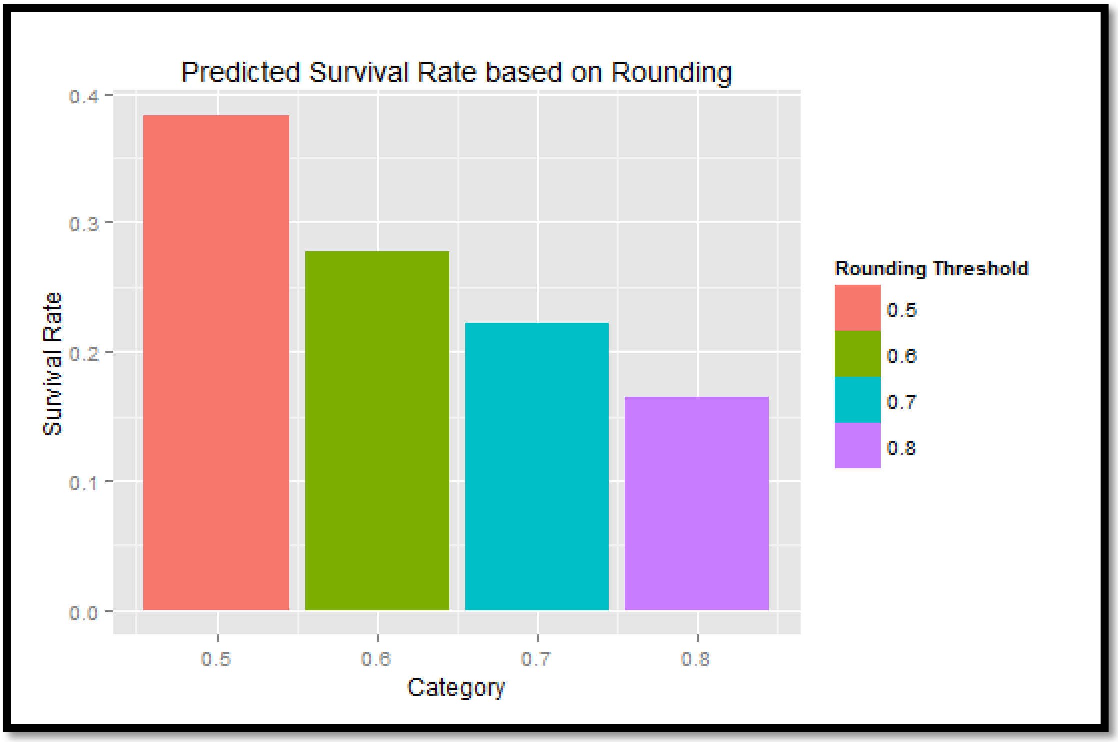
Solution

Survived.glm contains the probability that a particular passenger survived the Titanic. We want to convert this probability into an actual value indicating survival. 0 is used for did not survive, while 1 is used for survived.

One way to accomplish this is to simply round the probability values. Since a value of 0.9 indicates that the passenger has a 90% chance of survival, then rounding that value translates into a 1 (which means survived). Thus, any passenger with a 50% chance of survival or higher is given the benefit of the doubt and is predicted to survive.

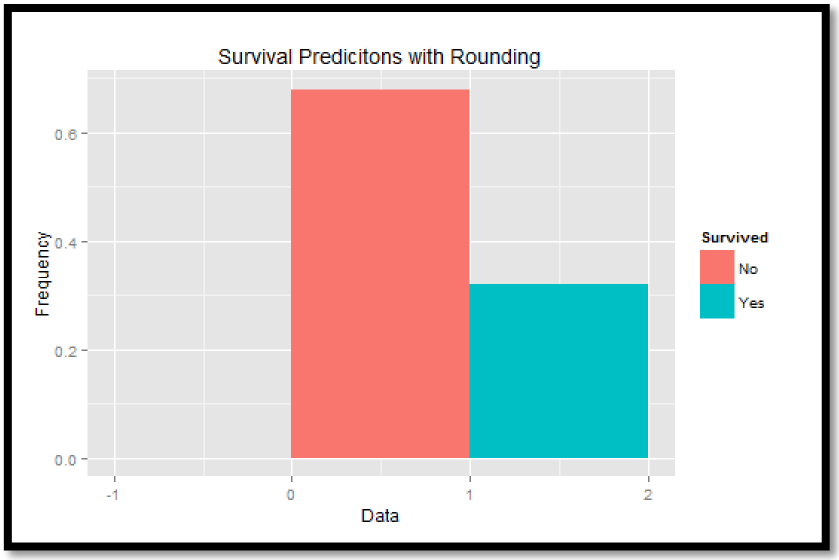
This solution estimates that 38% of passengers survived the Titanic. We know that the actual rate is closer to 31%. So, we can create a function that moves the threshold for round. If a value falls above that threshold, we can round the value up, and if it is below the threshold, round it down. The larger the threshold value, the lower the predicted survival rate.





A rounding threshold of 0.5 is too high, but a rounding threshold of 0 .6 is too low. There is a steady relationship between the rounding threshold and the survival rate. We can then apply a linear regression to this data in order to find the rounding threshold that will get us a survival rate of 31%. That magic number turns out to be 0.5766054. We can then apply this value and get the final prediction. While the prediction did not land as perfectly on 31% survival rate as expected, it is close enough. Using this data gives us a Kaggle score of **0.77990.**

Survived	No	Yes
Frequency	0.6794	0.3206


Survival Predicitons with Rounding
Frequency
Data
Survived
No
Yes