Using a Neural Network to Predict Titanic Survival

Jarred Priester

12/28/2021

1. Overview

1.1 description of dataset

This is my first attempt at a Kaggle competition. The competition I decided to give a try was the "Titanic - Machine Learning from Disaster". This competition will we be using a data set on the passengers of the Titanic. The following is from Kaggle's competition webpage:

The Challenge

The sinking of the Titanic is one of the most infamous shipwrecks in history.

On April 15, 1912, during her maiden voyage, the widely considered "unsinkable" RMS Titanic sank after colliding with an iceberg. Unfortunately, there weren't enough lifeboats for everyone onboard, resulting in the death of 1502 out of 2224 passengers and crew.

While there was some element of luck involved in surviving, it seems some groups of people were more likely to survive than others.

In this challenge, we ask you to build a predictive model that answers the question: "what sorts of people were more likely to survive?" using passenger data (ie name, age, gender, socio-economic class, etc).

We are given two data sets. One being titled train and the other being titled test. The train data set consist of the following variables:

- PassendgerId
- Survived
- Pclass
- Name
- Sex
- Age
- SibSp
- Parch
- Ticket
- Fare
- Cabin
- Embarked

The test data set consist of all the same variables except for the Survived variables. That missing information is what we will be trying to predict in this project.

1.2 goal of the project

The first goal of this project is to simply get started with Kaggle by entering into a competition and successfully submitting predictions. Second, I am wanting to share this code in order to help others like myself write their own projects in R and get started with Kaggle. Third, I am wanting to practice using a neural network to make some predictions. Fourth, after looking at other scores on the leaderboard a score of 80% accuracy look like a good score so we will use that as our benchmark for this project.

1.3 steps to achieve the goal

We will be applying a neural network using the nnet method from the caret library. First we will download the data given to us from Kaggle. Then we will clean up the data set by replacing missing information as well as creating a few new variables. Then we will analyze the data through visualization to help determine which variables are important. Then we will apply the neural network to get our predictions.

2. Data Cleaning

2.1 downloading the data

```
#loading libraries
if(!require(tidyverse)) install.packages("tidyverse")
if(!require(caret)) install.packages("caret")
if(!require(ggplot2)) install.packages("ggplot2")
```

```
if(!require(dplyr)) install.packages("dplyr")
if(!require(rpart)) install.packages("rpart")
if(!require(mice)) install.packages("mice")
if(!require(Rcpp)) install.packages("Rcpp")
if(!require(ggthemes)) install.packages("ggthemes")

library(tidyverse)
library(caret)
library(ggplot2)
library(dplyr)
library(rpart)
library(mice)
library(Rcpp)
library(ggthemes)

#Loading the data
train <- read.csv("train.csv",stringsAsFactors = F)
test <- read.csv("test.csv",stringsAsFactors = F)
```

Now we will combine both data sets and call this "all". We will use this for data exploration and feature engineering, then we will split ""all"" back into train and test for the model.

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

Now let's take a look at the data set

```
head(all)
```

```
##   PassengerId Survived Pclass
## 1           1        0      3
## 2           2        1      1
## 3           3        1      3
## 4           4        1      1
## 5           5        0      3
## 6           6        0      3
##                                                 Name    Sex Age SibSp Parch
## 1                            Braund, Mr. Owen Harris   male  22     1     0
## 2 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38     1     0
## 3                             Heikkinen, Miss. Laina female  26     0     0
## 4       Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35     1     0
## 5                           Allen, Mr. William Henry   male  35     0     0
## 6                                   Moran, Mr. James   male  NA     0     0
##             Ticket    Fare Cabin Embarked
## 1        A/5 21171  7.2500              S
## 2         PC 17599 71.2833   C85        C
## 3 STON/O2. 3101282  7.9250              S
## 4           113803 53.1000  C123        S
## 5           373450  8.0500              S
## 6           330877  8.4583              Q
```

```
str(all)
```

```
## 'data.frame':    1309 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" ...
```

```
summary(all)
```

```
##   PassengerId      Survived          Pclass          Name
##  Min.   :   1   Min.   :0.0000   Min.   :1.000   Length:1309
##  1st Qu.: 328   1st Qu.:0.0000   1st Qu.:2.000   Class :character
##  Median : 655   Median :0.0000   Median :3.000   Mode  :character
##  Mean   : 655   Mean   :0.3838   Mean   :2.295
##  3rd Qu.: 982   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :1309   Max.   :1.0000   Max.   :3.000
##                 NA's   :418
##      Sex                 Age            SibSp            Parch
##  Length:1309        Min.   : 0.17   Min.   :0.0000   Min.   :0.000
##  Class :character   1st Qu.:21.00   1st Qu.:0.0000   1st Qu.:0.000
##  Mode  :character   Median :28.00   Median :0.0000   Median :0.000
##                     Mean   :29.88   Mean   :0.4989   Mean   :0.385
##                     3rd Qu.:39.00   3rd Qu.:1.0000   3rd Qu.:0.000
##                     Max.   :80.00   Max.   :8.0000   Max.   :9.000
##                     NA's   :263
##     Ticket               Fare            Cabin             Embarked
##  Length:1309        Min.   :  0.000   Length:1309        Length:1309
##  Class :character   1st Qu.:  7.896   Class :character   Class :character
##  Mode  :character   Median : 14.454   Mode  :character   Mode  :character
##                     Mean   : 33.295
##                     3rd Qu.: 31.275
##                     Max.   :512.329
##                     NA's   :1
```

Having looked at the feature classes, we are going to factor a few of them.

```
all <- all %>% mutate(Survived = factor(Survived),
                      Pclass = factor(Pclass),
                      Sex = factor(Sex),
                      Embarked = factor(Embarked))
```

2.2 feature engineering

There is not much we can do with the names but we can extract the titles and group them

```
all$Title <- gsub('(.*, )|(\\..*)', '', all$Name)

#showing a tibble of the titles and the count for each of #them
all %>% group_by(Title)%>%
  summarize(count = n())
```

```
## # A tibble: 18 x 2
##    Title        count
##    <chr>        <int>
##  1 Capt             1
##  2 Col              4
##  3 Don              1
##  4 Dona             1
##  5 Dr               8
##  6 Jonkheer         1
##  7 Lady             1
##  8 Major            2
##  9 Master          61
## 10 Miss           260
## 11 Mlle             2
## 12 Mme              1
## 13 Mr             757
## 14 Mrs            197
## 15 Ms               2
## 16 Rev              8
## 17 Sir              1
## 18 the Countess     1
```

```
#changing a few titles to miss
all$Title[all$Title == 'Ms'] <- 'Miss'
all$Title[all$Title == 'Mlle'] <- 'Miss'

#changing Mme to Mrs
all$Title[all$Title == 'Mme'] <- 'Mrs'

#most of these titles only show up once or a few times in #order to avoid over fitting we will group th
other <- c('Capt','Col','Don','Dona','Jonkheer','Lady','Major',
           'Rev','Sir','the Countess')

all$Title[all$Title %in% other]  <- 'Other'

#factoring the Titles
all$Title <- factor(all$Title)
```

Next we are going to create a feature called the family size

```
all$Family_size <- all$SibSp + all$Parch + 1

#factoring Family size
all$Family_size <- factor(all$Family_size)
```

2.3 Cleaning missing values

From looking at the data we can see that two observations are blank for embarked. Let's look at the average fare cost per embarked and make an estimation.

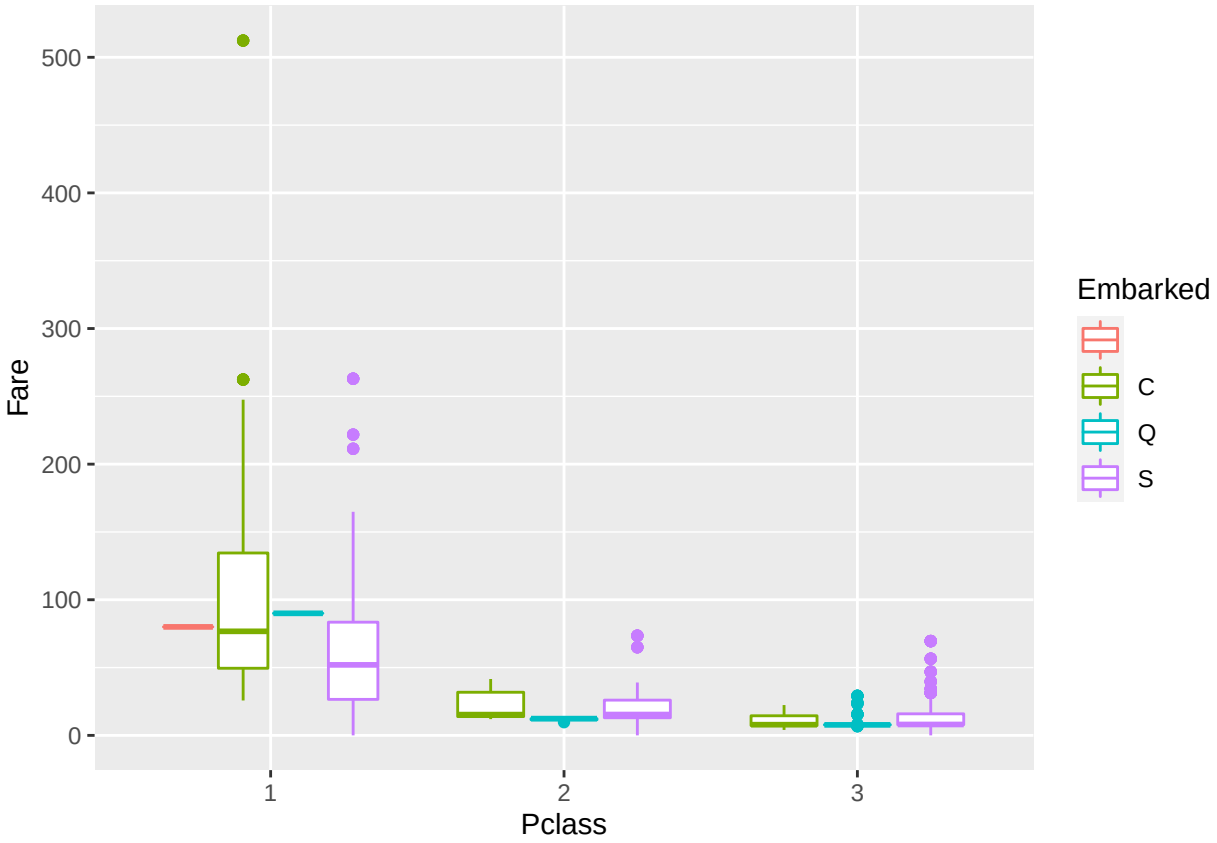
```
which(all$Embarked == "")
```

```
## [1]  62 830
```

```
view(all[c(62,830),])

#we know that both of these observations have the same pclass and fare, lets
#look at a boxplot to visualize this
all %>% ggplot(aes(Pclass,Fare)) +
  geom_boxplot(aes(color = Embarked))
```

```
## Warning: Removed 1 rows containing non-finite values (stat_boxplot).
```




```
#Both of these observations look like they should be Embarked from C
#changing the missing observations to C
all$Embarked[c(62,830)] <- "C"

#factoring the Embarked feature, we should only have 3 levels now
all$Embarked <- factor(all$Embarked)
```

Finding the remaining blank observations.

```
colSums(all == "")
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0          NA           0           0           0          NA
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0          NA        1014           0
##       Title Family_size
##           0           0
```

Cabin has too many that are black so we will leave that column alone.

Looking at the number of NA for each column

```
colSums(is.na(all))
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0         418           0           0           0         263
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           1           0           0
##       Title Family_size
##           0           0
```

Fare has one NA in the test set. We will change that to the average fare.

```
which(is.na(test$Fare))
```

```
## [1] 153
```

```
#taking a look at the row that has the NA
test[153,]
```

```
##     PassengerId Pclass               Name  Sex  Age SibSp Parch Ticket Fare
## 153        1044      3 Storey, Mr. Thomas male 60.5     0     0   3701   NA
##     Cabin Embarked
## 153              S
```

```
#changing the NA to the avg Fare
all <- all %>%
  mutate(Fare = ifelse(is.na(Fare),median(Fare, na.rm = TRUE),Fare))

#checking that the NA was changed
sum(is.na(all$Fare))
```

```
## [1] 0
```

263 NA in total for Age. Using the mice function to fill in those NAs with predictions

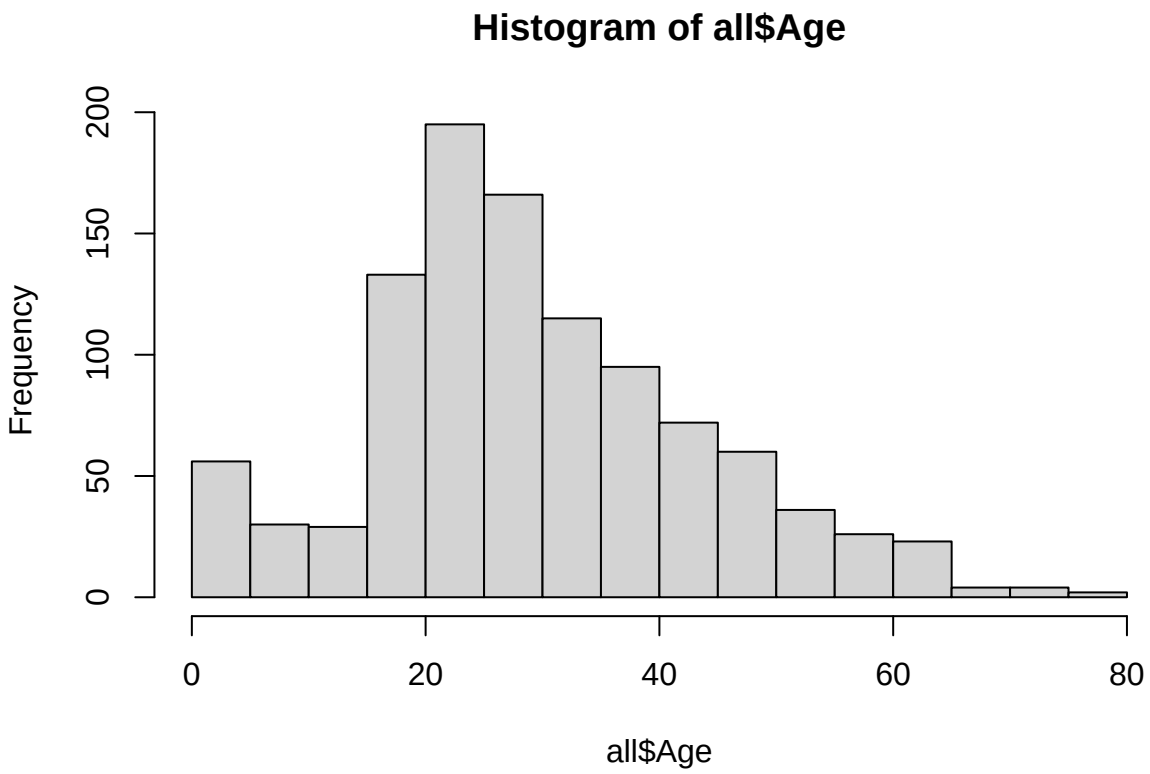
```r
temp <- all %>% select(Pclass,Sex,Age)

set.seed(1)
mice_input <- mice(temp, method = 'rf')
```

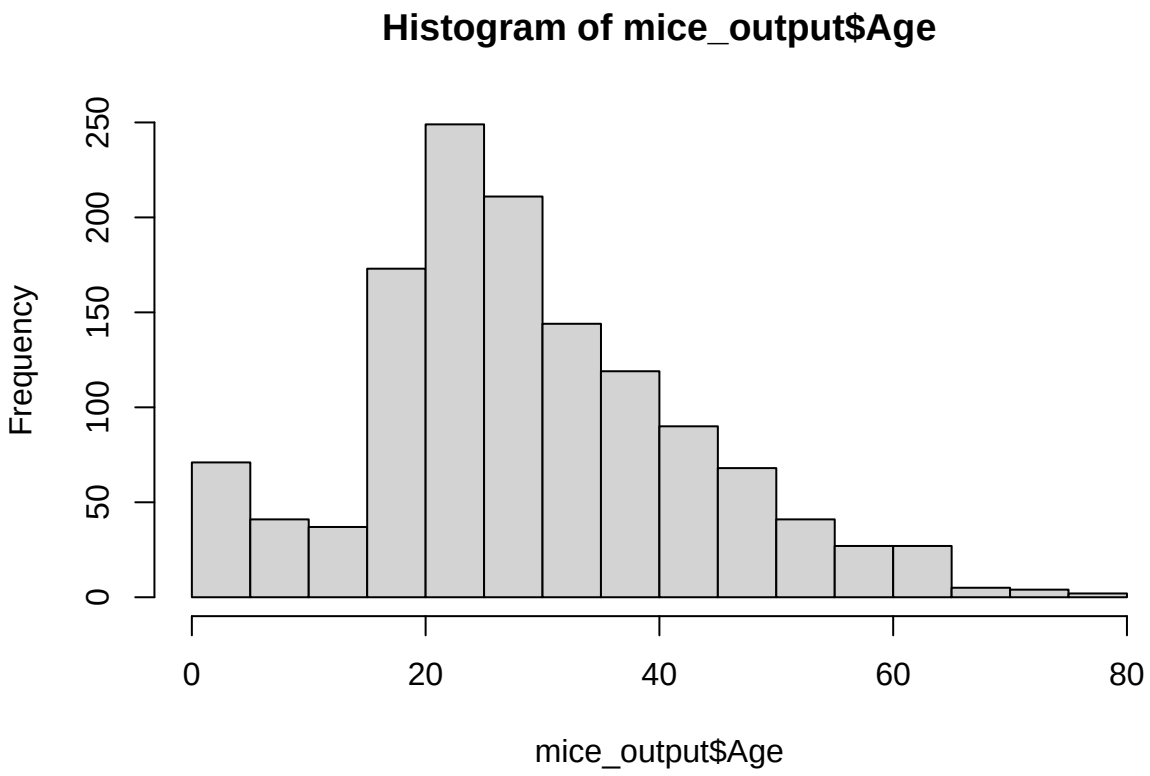
```
##
##  iter imp variable
##   1   1  Age
##   1   2  Age
##   1   3  Age
##   1   4  Age
##   1   5  Age
##   2   1  Age
##   2   2  Age
##   2   3  Age
##   2   4  Age
##   2   5  Age
##   3   1  Age
##   3   2  Age
##   3   3  Age
##   3   4  Age
##   3   5  Age
##   4   1  Age
##   4   2  Age
##   4   3  Age
##   4   4  Age
##   4   5  Age
##   5   1  Age
##   5   2  Age
##   5   3  Age
##   5   4  Age
##   5   5  Age
```

```r
mice_output <- complete(mice_input)

#using histograms to make sure the new predictions match the distribution of all
hist(all$Age)
```




```
hist(mice_output$Age)
```




```
#replacing age variable with new age predictions
all$Age <- mice_output$Age

#checking to see if there are any NA in train$Age
sum(is.na(all$Age))
```

```
## [1] 0
```

We now have 0 NAs in the data set expect for the 418 Survived values we are trying to predict.

```
colSums(is.na(all))
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0         418           0           0           0           0
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           0           0           0
##       Title Family_size
##           0           0
```

3. Exploratory Data Analysis

3.1 exploring the data

What percentage of the data set are male

```
mean(all$Sex == "male")
```

```
## [1] 0.6440031
```

What percentage of the data set are female

```
mean(all$Sex == "female")
```

```
## [1] 0.3559969
```

What percentage of the data set survived

```
mean(train$Survived == 1)
```

```
## [1] 0.3838384
```

What percentage of the data set died

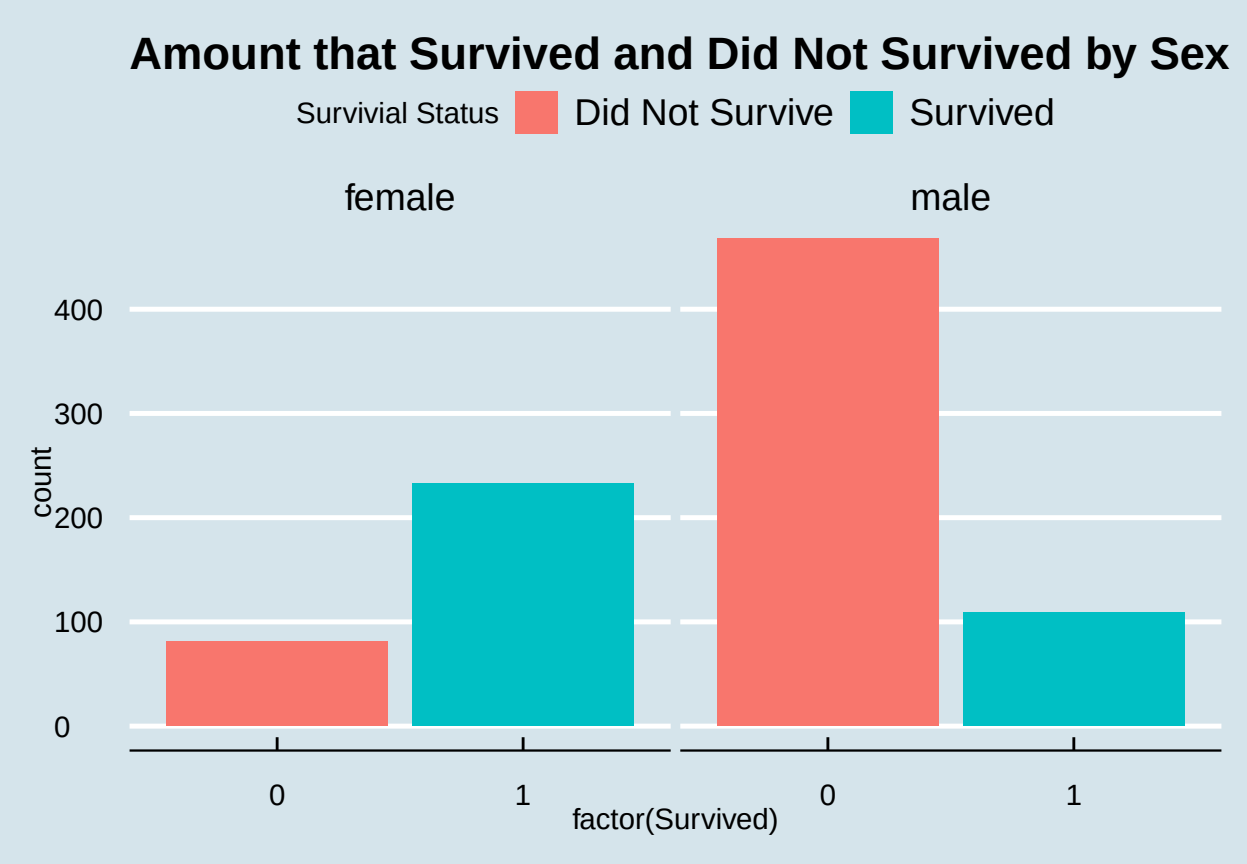
```
mean(train$Survived == 0)
```

```
## [1] 0.6161616
```

3.2 visualization

Graph showing the amount survived and not survived split by Sex

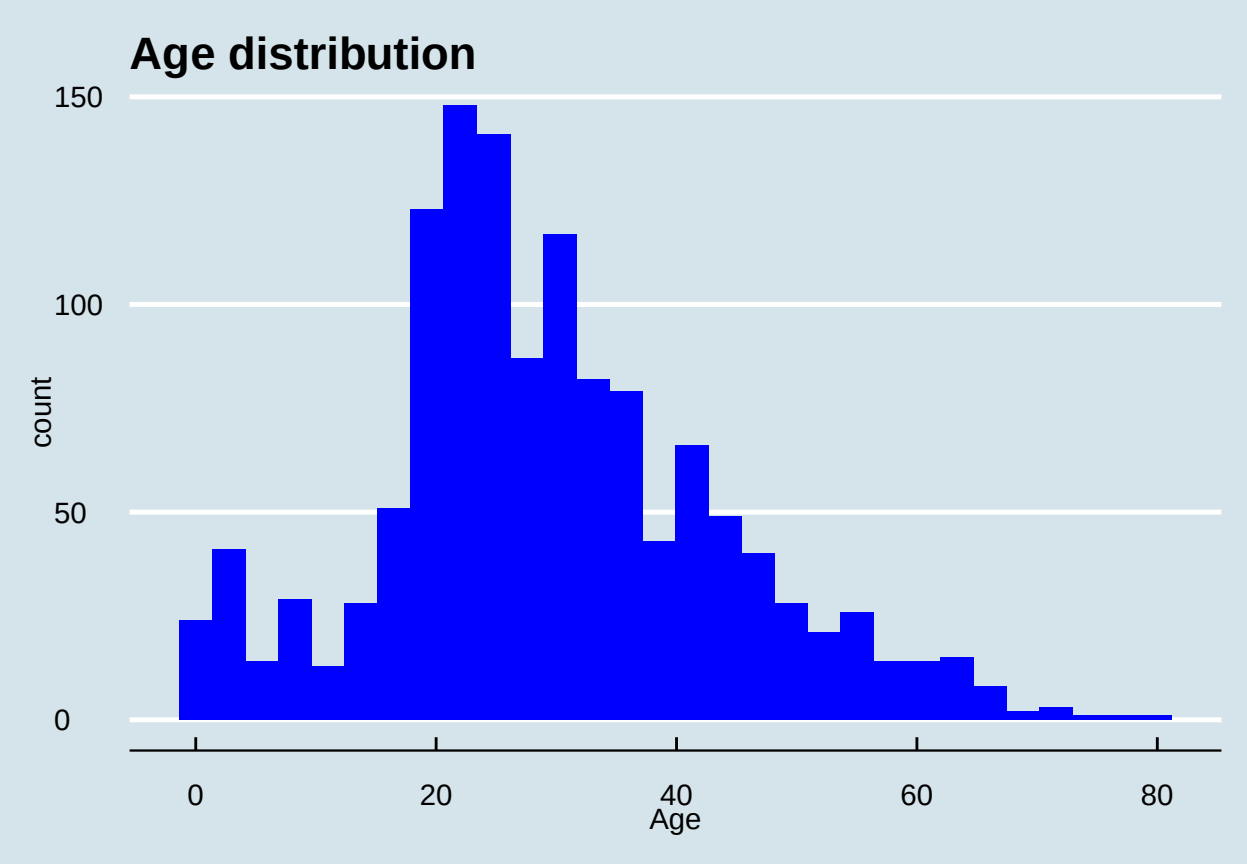
```
train %>% ggplot(aes(factor(Survived))) +
  facet_grid(.~Sex) +
    geom_bar(aes(fill=factor(Survived))) +
  ggtitle("Amount that Survived and Did Not Survived by Sex") +
  scale_fill_discrete(name = "Survivial Status",
                      labels = c("Did Not Survive", "Survived")) +
  theme_economist()
```



histogram of age distribution in the data

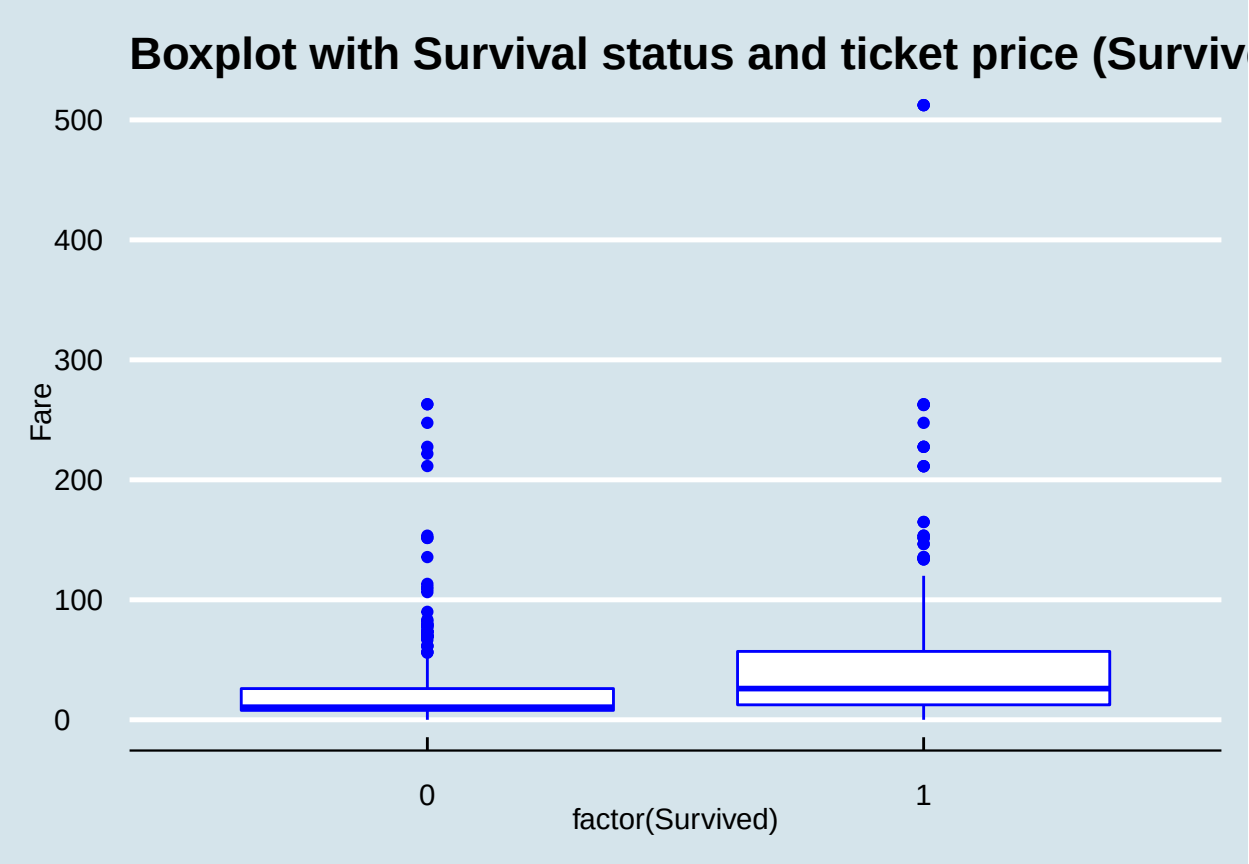
```
all %>% ggplot(aes(Age)) +
  geom_histogram(fill = "blue") +
  ggtitle("Age distribution") +
  theme_economist()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



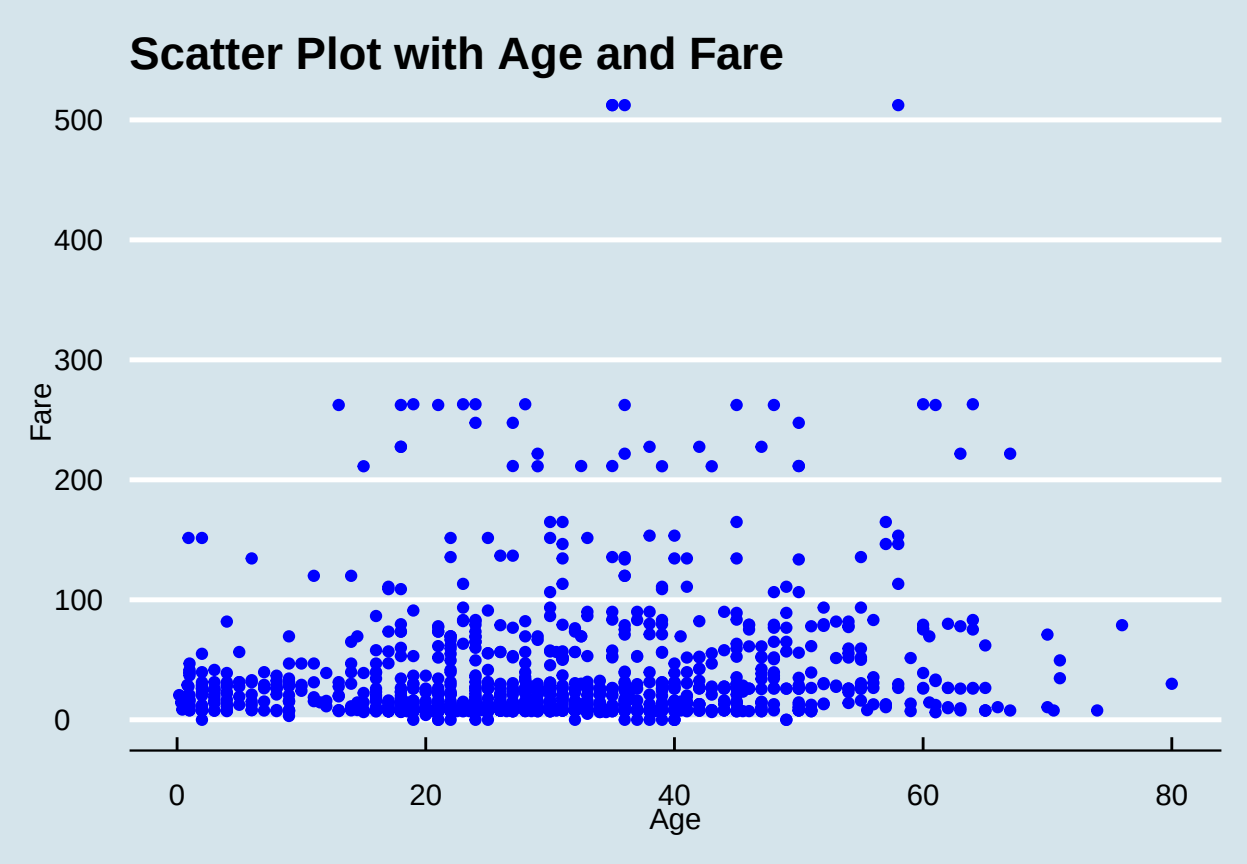
Boxplot with Survival status and ticket price

```
train %>% ggplot(aes(factor(Survived),Fare)) +
  geom_boxplot(color = "blue") +
  ggtitle("Boxplot with Survival status and ticket price (Survived = 1)") +
theme_economist()
```



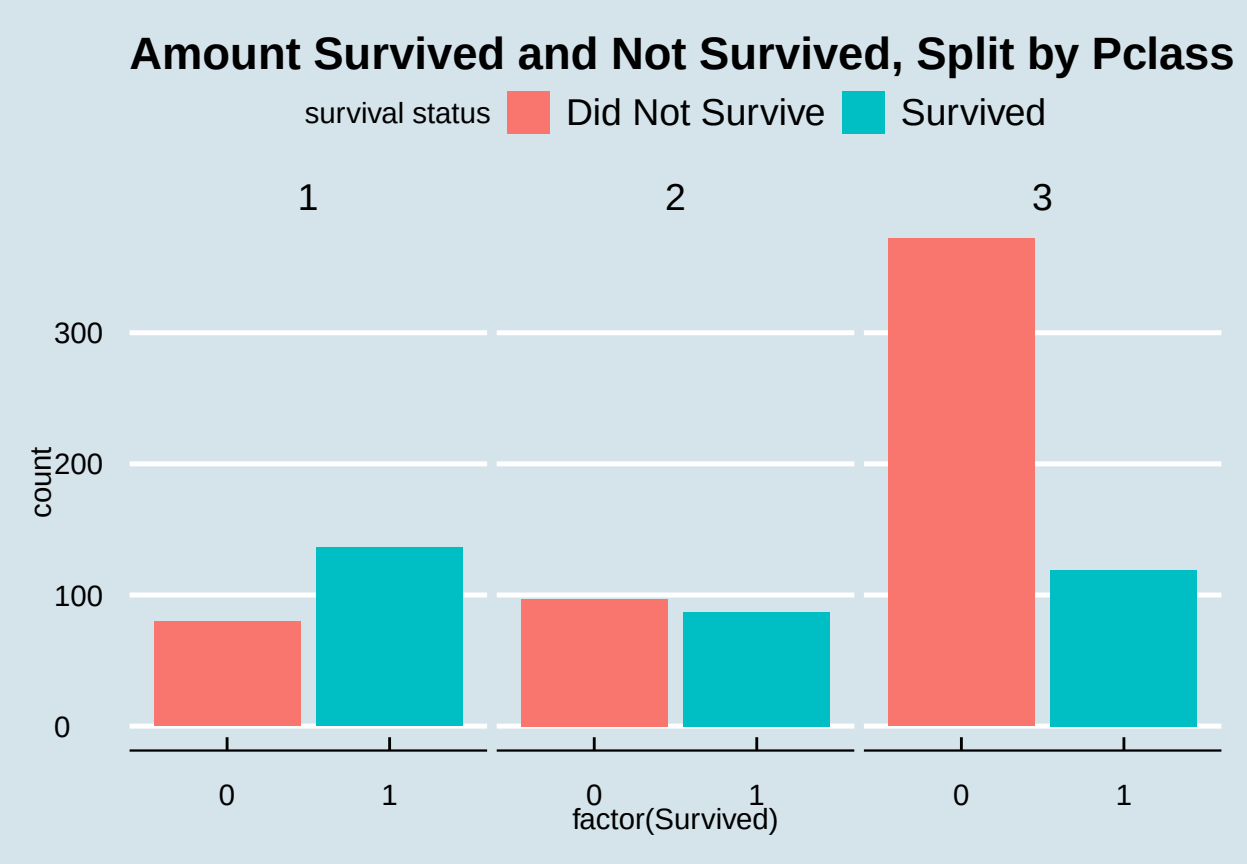
Scatter plot with Age and Fare with Survival Status

```
all %>% ggplot(aes(Age,Fare)) +
  geom_point(color = "blue") +
  ggtitle("Scatter Plot with Age and Fare") +
  xlab("Age") +
  ylab("Fare") +
  theme_economist()
```



Bar graph with Pclass and Survival Status

```
train %>% ggplot(aes(factor(Survived))) +
  facet_grid(.~Pclass) +
  geom_bar(aes(fill=factor(Survived))) +
  ggtitle("Amount Survived and Not Survived, Split by Pclass") +
  scale_fill_discrete(name = "survival status",
                      labels = c("Did Not Survive","Survived")) +
  theme_economist()
```



4. Neural Network

4.1 Neural Network setup

splitting the all dataset back into train and test

```
train <- all[1:891,]
test <- all[892:1309,]
```

Will be using k-fold cross validation on all the algorithms creating the k-fold parameters, k is 10

```
set.seed(1, sample.kind = "Rounding")
```

```
## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used
```

```
control <- trainControl(method = "cv", number = 10, p = .9)
```

setting the parameters for the neural network

```
tuning <- data.frame(size = seq(100), decay = seq(.01,1,.1))
```

creating the x and y for the model. X is the data that will be used as input. Y is what we will be trying to predict as the output.

```
train_x <- train %>% select(Pclass, Sex, Age, SibSp, Parch, Fare, Embarked,Title,Family_size)

train_y <- train$Survived
```

4.2 Neural Network

```
#Predicting survival by using a neural network
set.seed(1, sample.kind = "Rounding")
train_nn <- train(train_x, train_y,
                  method = "nnet",
                  tuneGrid = tuning,
                  trControl = control)
```

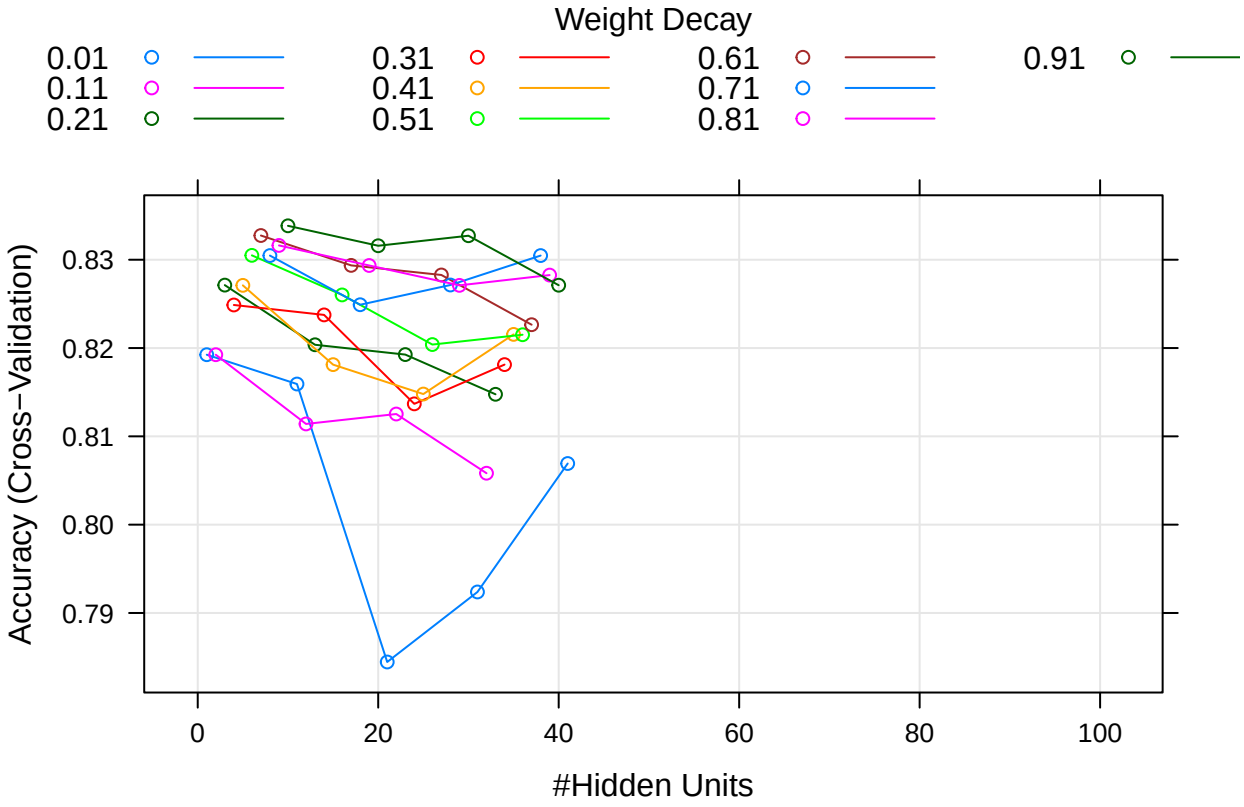
best tune

```
train_nn$bestTune
```

```
##    size decay
## 10   10  0.91
```

4.3 Results

```
#plotting results
plot(train_nn)
```



creating the predictions

```
nn_preds <- predict(train_nn, test)

solution <- data.frame(PassengerID = test$PassengerId,
                       Survived = nn_preds)

write.csv(solution, file = 'nn.titanic.preds.cvs', row.names = FALSE)
```

These predictions were submitted to the Kaggle Titanic competition and received a score of 77% accuracy. This was just below our target goal of 80%.

5. Conclusion

To recap, we downloaded the data from Kaggle, cleaned and analyzed the data, and successfully created predictions by using a neural network. As mentioned, this did not meet our goal of 80% but if we wanted to improve this score we could add more algorithms and combine them to make an ensemble. A neural network may not be the most effective algorithm for this problem and by combining different algorithms we may achieve the goal of 80%. All in all, I believed this was good start in practicing using neural networks and I hope that you were able to learn something from this notebook. This is my first Kaggle notebook so please feel free to leave feedback, I am always wanting to learn and improve.