Titantic Survival

Alex Thom

03/12/2019

```
train2 <- train %>% select(Survived, Pclass, Sex, Embarked)
```

Introduction

The Titanic was a major catastrophe in 1912 with an estimated 832 passengers and 685 crew members perishing in the disaster. This review of machine learning techniques looks at the best methodologies for predicitng whether a passenger would have survied or died. In order to accomplish this task there is a dataset provided below:

```
str(train)
```

```
## Classes 'tbl_df', 'tbl' and '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       : Factor w/ 891 levels "Abbing, Mr. Anthony",..: 109 191 358 277 16 559 520 629 417 58
##  $ Sex        : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
##  $ 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     : Factor w/ 681 levels "110152","110413",..: 524 597 670 50 473 276 86 396 345 133 ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : Factor w/ 148 levels "","A10","A14",..: 1 83 1 57 1 1 131 1 1 1 ...
##  $ Embarked   : Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
```

The variable that will be predicted is the Survived variable. Although it is currently an integer, it is really a factor as 0 is died and 1 is survied. This will effect the type of model that is used as the target variable being a facotr means this is a classification problem. The missing data is summarised below

```
sapply(train, function(x) {sum(is.na(x))})
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0           0           0           0           0         177
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           0           0           0
```

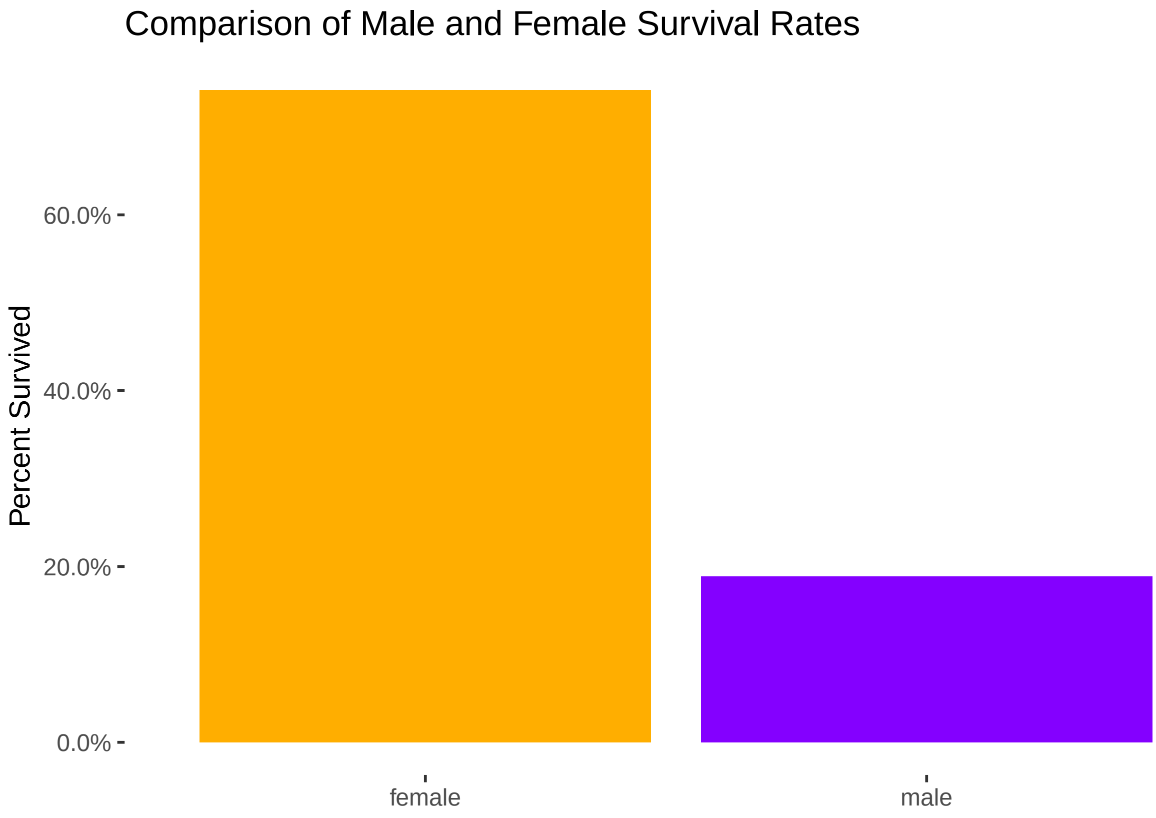
In the training set it looks like there is a lot of missing data for the age variable. This could be something that could be improved on in feature engineering part of this project. Also cabine has a lot of missing data which might be harder to develope into a variabel.

```
library(scales)
```

```
##
## Attaching package: 'scales'
## The following object is masked from 'package:purrr':
##
##     discard
```

```
## The following object is masked from 'package:readr':
##
##     col_factor
```

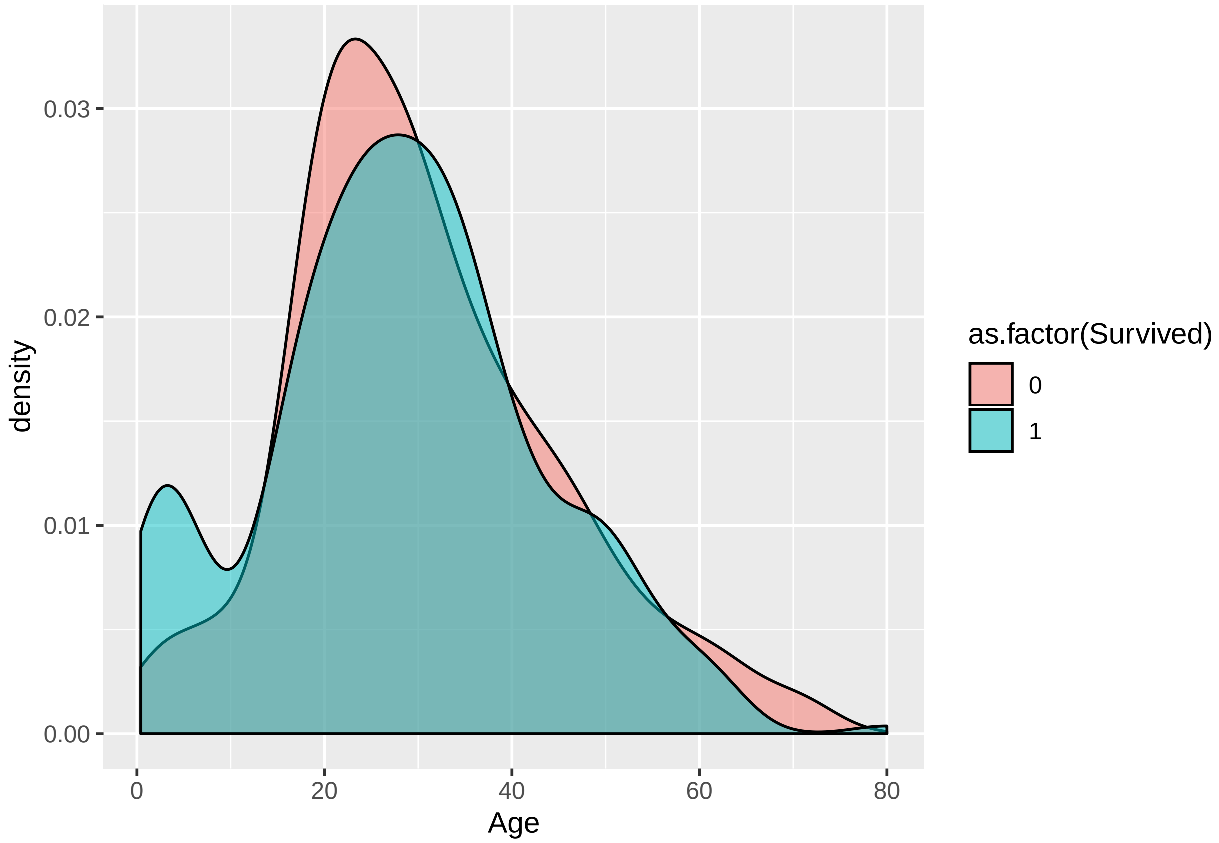
```
vis1 <- train %>% group_by(Sex) %>%
                    summarise(tot = sum(Survived), n = n()) %>%
                      mutate(per = tot/n)


cols <- c("female" = "#ffae00", "male" = "#8400ff")

ggplot(vis1, aes(x = Sex, y = per, fill = Sex)) +
                                          geom_col() +
                                            scale_y_continuous(labels = percent_format()) +
                                              scale_fill_manual(values = cols) +
                                                labs(x = "", y = "Percent Survived", title = "Compariso
                                                          guides(fill = F) +
                                                    theme(panel.background = element_blank())
```



Its clear that females had a significant higher chance of survivl and therefore the Sex variable will have signifancat influneces in the models predictions.

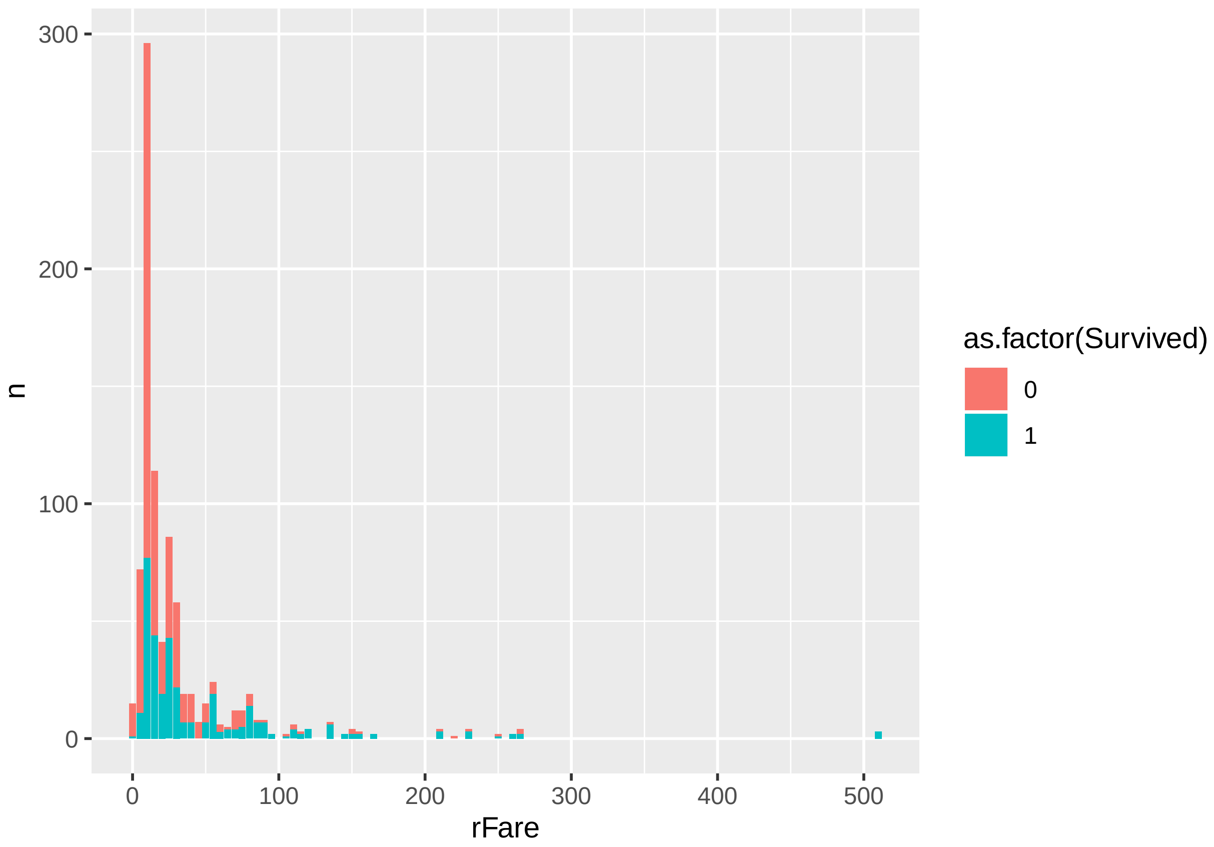
```
ggplot(train, aes(x = Age, group = Survived, fill = as.factor(Survived))) + geom_density(alpha = 0.5)
```

```
## Warning: Removed 177 rows containing non-finite values (stat_density).
```



COmparing how ages effects survival the major feature visible is at the younger ages. Above 10 years old the rates of survival are pretty similar. Howerver, less then 10 years old it looks like there are significantly more likelly to be able to survive. This variable has some missing data however it looks like it has interesting features so will have to create a method in order to geat more details from this feature.

```
mround <- function(x,base){
        base*round(x/base)
}


vis2 <- train %>% mutate(rFare = mround(Fare, 5)) %>%
                    group_by(rFare, Survived) %>%
                        summarise(n = n())

ggplot(vis2, aes(x = rFare, y = n, fill = as.factor(Survived))) + geom_col()
```



The lower fares are dominated by people who didnt survive the catastophe. This will be another feature that will have a storng preidctive power.

Methods

```
rf <- train(as.factor(Survived)~., data = train2,  method = "rf", trControl = trainControl(method = "cv
```

```
rf_pred <- predict(rf, testdata)


confusionMatrix(rf_pred, as.factor(testdata$Survived))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 237  70
##          1  23  88
##
##                Accuracy : 0.7775
##                  95% CI : (0.7346, 0.8165)
##     No Information Rate : 0.622
##     P-Value [Acc > NIR] : 7.115e-12
##
```

```
##                   Kappa : 0.4975
##
##  Mcnemar's Test P-Value : 1.842e-06
##
##             Sensitivity : 0.9115
##             Specificity : 0.5570
##          Pos Pred Value : 0.7720
##          Neg Pred Value : 0.7928
##              Prevalence : 0.6220
##          Detection Rate : 0.5670
##    Detection Prevalence : 0.7344
##       Balanced Accuracy : 0.7343
##
##        'Positive' Class : 0
##
```

```
svm <- train(as.factor(Survived)~., data = train2,  method = "svmLinear", trControl = trainControl(meth
```

```
svm_pred <- predict(svm, testdata)


confusionMatrix(svm_pred, as.factor(testdata$Survived))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 214  52
##          1  46 106
##
##                Accuracy : 0.7656
##                  95% CI : (0.7219, 0.8054)
##     No Information Rate : 0.622
##     P-Value [Acc > NIR] : 2.705e-10
##
##                   Kappa : 0.4977
##
##  Mcnemar's Test P-Value : 0.6135
##
##             Sensitivity : 0.8231
##             Specificity : 0.6709
##          Pos Pred Value : 0.8045
##          Neg Pred Value : 0.6974
##              Prevalence : 0.6220
##          Detection Rate : 0.5120
##    Detection Prevalence : 0.6364
##       Balanced Accuracy : 0.7470
##
##        'Positive' Class : 0
##
```

```
svm <- train(as.factor(Survived)~., data = train2,  method = "svmLinear", trControl = trainControl(metho
```