Surviving on the Titanic

Charles Westby

12/9/2017

Synopsis

Almost everyone is familiar with the sinking of the Titanic. When the large ship sunk in the Atlantic, killing a majority of its passengers, it stunned the world. This report explores a dataset containing information on passengers on the Titanic. Some survived and some passed away. In the end, a machine learning model will be built using this information, in order to predict who will survive the Titanic given a different set of passengers.

Exploratory Analysis

Loading Packages and Data

```
#loading the library packages and dataset
library(dplyr)
library(ggplot2)
library(caret)
library(gridExtra)
library(VIM)

titanic <- read.csv("train.csv")
```

Previewing the Data

```
str(titanic)
```

```
## '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 16 559 520 629 417 581
##  $ 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 ...
```

```
head(titanic)
```

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

Summary

```
summary(titanic)
```

```
##   PassengerId       Survived          Pclass
##  Min.   :  1.0   Min.   :0.0000   Min.   :1.000
##  1st Qu.:223.5   1st Qu.:0.0000   1st Qu.:2.000
##  Median :446.0   Median :0.0000   Median :3.000
##  Mean   :446.0   Mean   :0.3838   Mean   :2.309
##  3rd Qu.:668.5   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :891.0   Max.   :1.0000   Max.   :3.000
##
##                                     Name         Sex           Age
##  Abbing, Mr. Anthony                   :  1   female:314   Min.   : 0.42
##  Abbott, Mr. Rossmore Edward           :  1   male  :577   1st Qu.:20.12
##  Abbott, Mrs. Stanton (Rosa Hunt)      :  1                Median :28.00
##  Abelson, Mr. Samuel                   :  1                Mean   :29.70
##  Abelson, Mrs. Samuel (Hannah Wizosky):  1                3rd Qu.:38.00
##  Adahl, Mr. Mauritz Nils Martin        :  1                Max.   :80.00
##  (Other)                               :885                NA's   :177
##      SibSp           Parch             Ticket         Fare
##  Min.   :0.000   Min.   :0.0000   1601    :  7   Min.   :  0.00
##  1st Qu.:0.000   1st Qu.:0.0000   347082  :  7   1st Qu.:  7.91
##  Median :0.000   Median :0.0000   CA. 2343:  7   Median : 14.45
##  Mean   :0.523   Mean   :0.3816   3101295 :  6   Mean   : 32.20
##  3rd Qu.:1.000   3rd Qu.:0.0000   347088  :  6   3rd Qu.: 31.00
##  Max.   :8.000   Max.   :6.0000   CA 2144 :  6   Max.   :512.33
##                                   (Other) :852
##          Cabin     Embarked
##             :687    :  2
##  B96 B98    :  4   C:168
##  C23 C25 C27:  4   Q: 77
##  G6         :  4   S:644
##  C22 C26    :  3
##  D          :  3
```

```
##  (Other)    :186
```

The dataset contains records of 891 passengers. These records contain passenger ID, their class on the ship, age, sex, name, ticket number, cabin number, fare, where they boarded the ship and other measurements of family members on the ship. Some of these variables can be removed, which will happen in the next section.

Manipulating Dataset

Creating Factor Variables

```
#Creating Factor Variables out of Survived and Pclass columns
titanic$Survived <- factor(titanic$Survived, labels = c("Passed", "Survived"))
titanic$Pclass <- factor(titanic$Pclass, labels = c("First", "Second", "Third"))
```

Creating Subset of Titanic Dataset

```
#Subsetting and viewing dataset
titanic_sub <- titanic %>%
  select(-PassengerId, -Name, -Ticket, -Cabin, -SibSp, -Parch)
glimpse(titanic_sub)
```

```
## Observations: 891
## Variables: 6
## $ Survived <fctr> Passed, Survived, Survived, Survived, Passed, Passed...
## $ Pclass   <fctr> Third, First, Third, First, Third, Third, First, Thi...
## $ Sex      <fctr> male, female, female, female, male, male, male, male...
## $ Age      <dbl> 22, 38, 26, 35, 35, NA, 54, 2, 27, 14, 4, 58, 20, 39,...
## $ Fare     <dbl> 7.2500, 71.2833, 7.9250, 53.1000, 8.0500, 8.4583, 51....
## $ Embarked <fctr> S, C, S, S, S, Q, S, S, S, C, S, S, S, S, S, S, Q, S...
```

NA Values

Finding NA Values

```
#Finding total NA values in Age and Fare columns
sum(is.na(titanic_sub$Age))
```

```
## [1] 177
```

```
sum(is.na(titanic_sub$Fare))
```

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

Replacing NA Values

```
#Imputing NA values
titanic_sub <- kNN(titanic_sub)
titanic_sub <- titanic_sub %>%
  select(-(Survived_imp:Embarked_imp))
glimpse(titanic_sub)
```

```
## Observations: 891
## Variables: 6
```

```
## $ Survived <fctr> Passed, Survived, Survived, Survived, Passed, Passed...
## $ Pclass   <fctr> Third, First, Third, First, Third, Third, First, Thi...
## $ Sex      <fctr> male, female, female, female, male, male, male, male...
## $ Age      <dbl> 22.0, 38.0, 26.0, 35.0, 35.0, 32.0, 54.0, 2.0, 27.0, ...
## $ Fare     <dbl> 7.2500, 71.2833, 7.9250, 53.1000, 8.0500, 8.4583, 51....
## $ Embarked <fctr> S, C, S, S, S, Q, S, S, S, C, S, S, S, S, S, S, Q, S...
```

Initially the variables, Survived and Pclass are stored as numeric variables. However, they needed to be converted to factor variables. In order to create clarity when looking at the dataset, labels were added to the levels of the factor variables. The 0 and 1 in the Survived column were converted to Passed and Survived respectively. Also the 1, 2, and 3 in the Pclass column were converted to First, Second and Third. The variables Name, PassengerID and Cabin were removed because they are attributes that will be unique to each passenger. These variables will not be useful in the analysis. Also there are 177 NA values in the Age variable of the dataset. These NA values will cause problems when trying to build a machine learning model. For this analysis, a suitable replacement for these NA values will be knn imputation. Knn imputation will replace missing values with a guess based on people with similar attributes.

Finding Relationships

Tables

```
#Creating Tables to See Who Survived and Who Didn't
table(titanic$Survived, titanic$Sex)
```

```
##
##            female male
##   Passed       81  468
##   Survived    233  109
```

```
table(titanic$Survived, titanic$Pclass)
```

```
##
##            First Second Third
##   Passed      80     97   372
##   Survived   136     87   119
```

```
table(titanic$Survived, titanic$Embarked)
```

```
##
##                C   Q   S
##   Passed     0  75  47 427
##   Survived   2  93  30 217
```

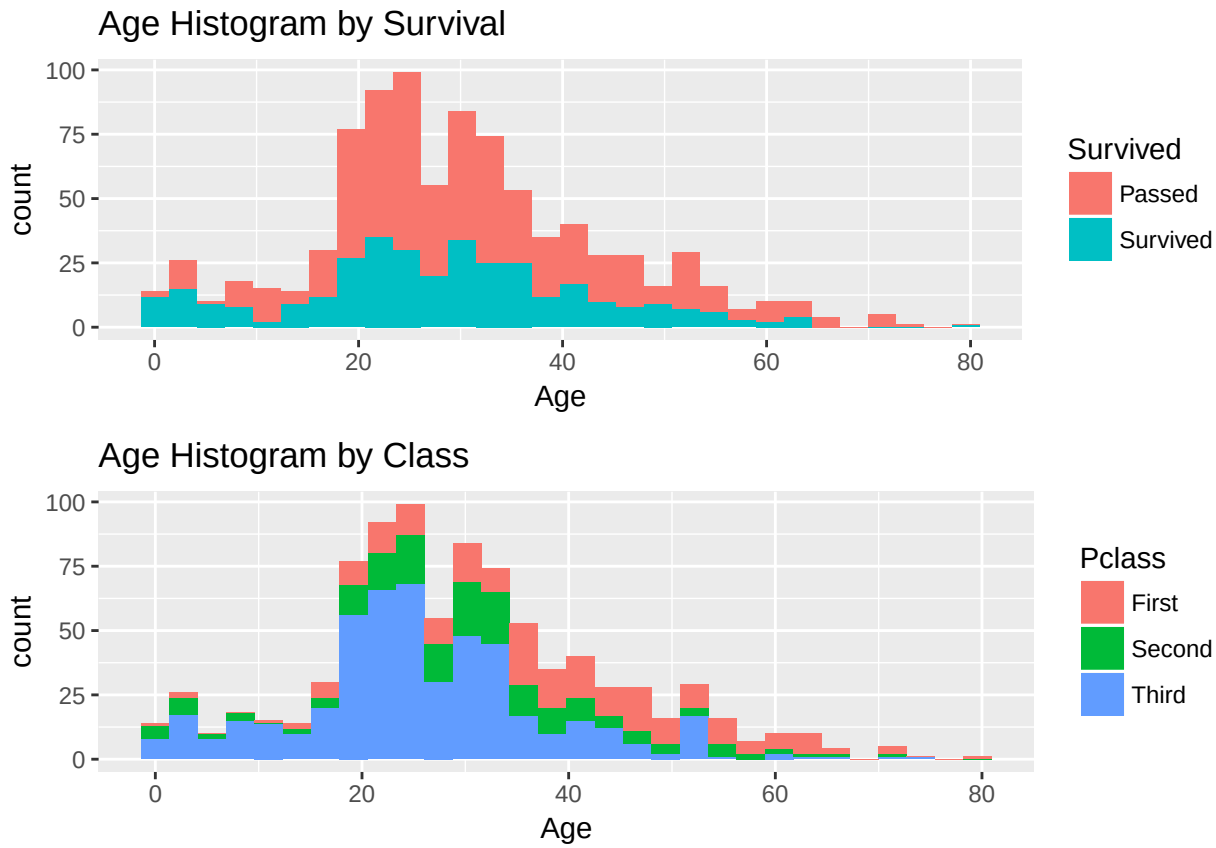
These tables show that the majority of survivors were female. It also shows that many who passed were male, from third class and who embarked from Southampton. Many of the survivors were also from Southampton, since it appears that this dock was where most passengers boarded.

Visualizing Data

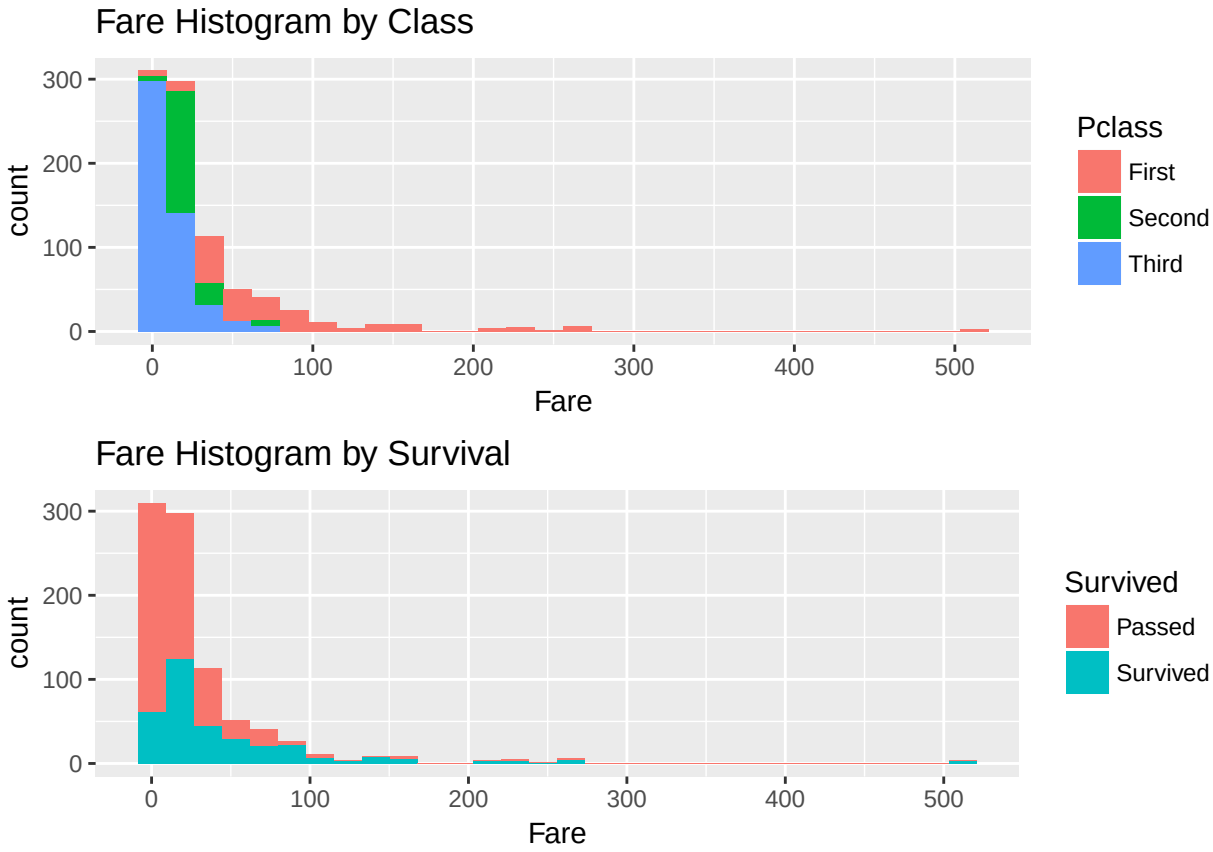
```
#Creating Scatter Plots of Fare vs Age
fa_class <- ggplot(titanic_sub, aes(x= Age, y = Fare, col = Pclass)) +
  geom_point() +
  labs(title = "Fare vs Age (Class)")

fa_survival <- ggplot(titanic_sub, aes(x=Age, y = Fare, col = Survived)) +
```

```
  geom_point() +
  labs(title = "Fare vs Age (Survival)")

grid.arrange(fa_class, fa_survival, ncol = 1, nrow = 2)
```


```
#Creating Histograms of Age
age_s_hist <- ggplot(titanic_sub, aes(x=Age, fill = Survived)) +
  geom_histogram() +
  labs(title = "Age Histogram by Survival")

age_c_hist <- ggplot(titanic_sub, aes(x=Age, fill = Pclass)) +
  geom_histogram() +
  labs(title = "Age Histogram by Class")
grid.arrange(age_s_hist, age_c_hist, ncol = 1, nrow = 2)
```




```
#Creating histograms of Fare
fare_s_hist <- ggplot(titanic_sub, aes(x=Fare, fill = Survived)) +
  geom_histogram() +
  labs(title = "Fare Histogram by Survival")

fare_c_hist <- ggplot(titanic_sub, aes(x=Fare, fill = Pclass)) +
  geom_histogram() +
  labs(title = "Fare Histogram by Class")

grid.arrange(fare_c_hist, fare_s_hist, ncol = 1, nrow = 2)
```



These graphs show that there is no relationship between the `Age` and `Fare` variables. However it does show that those who survived often paid a higher fare. In addition, the *Age Histogram by Survival* graph shows that older people passed away at a higher proportion than younger people. The age with the highest rate of survival occurs on the low end of the graph, suggesting that children survived at a higher rate. Additionally, most of the riders of the Titanic were between the ages of 20 and 40. The graph *Fare Histogram by Class* shows that those who paid the lowest fares died at the highest rates. Now that the Exploratory Analysis is concluded, the Machine Learning portion of the paper can begin.

Machine Learning Models

Partitioning The Data

```
set.seed(366284)
#Partitioning the data and subsetting data into a Train Set and a Test Set
inTrain <- createDataPartition(y = titanic_sub$Survived, p = 0.7, list=FALSE)
train <- titanic_sub[inTrain, ]
test <- titanic_sub[-inTrain, ]
```

During this step the dataset is split into a train set and a test set. The train set contains 70% of the data that was selected using random sampling. The test set contains the other 30% of the data. The `set.seed` function was called in order to ensure reproducibility for this paper.

Random Forest Model

```
#Creating Random Forest Model
model_rf <- train(Survived ~ ., train, method = "ranger", preProcess = c("center", "scale"), weights = 
model_rf
```

```
## Random Forest
##
## 625 samples
##   5 predictor
##   2 classes: 'Passed', 'Survived'
##
## Pre-processing: centered (8), scaled (8)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 562, 563, 562, 563, 563, 563, ...
## Resampling results across tuning parameters:
##
##   mtry  splitrule   Accuracy   Kappa
##   2     gini        0.8160522  0.5869918
##   2     extratrees  0.8143881  0.5825573
##   3     gini        0.8240143  0.6099831
##   3     extratrees  0.8159754  0.5844420
##   4     gini        0.8271889  0.6254364
##   4     extratrees  0.8143625  0.5833607
##   5     gini        0.8320789  0.6396720
##   5     extratrees  0.8127240  0.5889617
##   6     gini        0.8304659  0.6365018
##   6     extratrees  0.8191244  0.6122004
##   7     gini        0.8336662  0.6422804
##   7     extratrees  0.8288018  0.6345698
##   8     gini        0.8304916  0.6365859
##   8     extratrees  0.8271633  0.6302253
##
## Accuracy was used to select the optimal model using  the largest value.
## The final values used for the model were mtry = 7 and splitrule = gini.
```

This table shows the results of creating the Random Forest Model. The algorithm used accuracy to select the optimal model using the largest value. The final values for the model were 7 for mtry and gini for splitrule. This model's accuracy was about 83.37%.

```
#Testing Accuracy of Model
predictions_rf <- predict(model_rf, test)
confusionMatrix(predictions_rf, test$Survived)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Passed Survived
##   Passed      148       25
##   Survived     16       77
##
##                Accuracy : 0.8459
##                  95% CI : (0.7968, 0.8871)
##     No Information Rate : 0.6165
##     P-Value [Acc > NIR] : <2e-16
```

```
##
##                   Kappa : 0.6685
##  Mcnemar's Test P-Value : 0.2115
##
##             Sensitivity : 0.9024
##             Specificity : 0.7549
##          Pos Pred Value : 0.8555
##          Neg Pred Value : 0.8280
##              Prevalence : 0.6165
##          Detection Rate : 0.5564
##    Detection Prevalence : 0.6504
##       Balanced Accuracy : 0.8287
##
##        'Positive' Class : Passed
##
```

The model predicted with 84.59% accuracy when used to compare the predicted values for the test set with the actual values in the test set. This result is in line with the model which predicted that it would predict values at about 83.37% accuracy. Although this result is satisfactory, it is beneficial to test other models. The Sensitivity or True Positive Rate was 90.24%. The Specificity or True Negative Rate was 75.49%. Thus, the model was better at predicting who passed correctly than it was at predicting who survived.

Ada Model

```
#Creating Ada Model
model_ada <- train(Survived ~ ., train, method = "ada", weights = train$Fare, trControl = trainControl(
model_ada
```

```
## Boosted Classification Trees
##
## 625 samples
##   5 predictor
##   2 classes: 'Passed', 'Survived'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 563, 563, 563, 563, 562, 562, ...
## Resampling results across tuning parameters:
##
##   maxdepth  iter  Accuracy   Kappa
##   1          50   0.7869688  0.5415727
##   1         100   0.7869688  0.5408629
##   1         150   0.7885817  0.5444646
##   2          50   0.7903226  0.5438846
##   2         100   0.7919099  0.5487324
##   2         150   0.7966718  0.5588023
##   3          50   0.8238607  0.6146299
##   3         100   0.8142601  0.5998318
##   3         150   0.8126216  0.5968771
##
## Tuning parameter 'nu' was held constant at a value of 0.1
## Accuracy was used to select the optimal model using  the largest value.
## The final values used for the model were iter = 50, maxdepth = 3 and nu
```

```
##  = 0.1.
```

For this model accuracy was used to select the optimal model using the largest value. The final values used for the model were 50 for iter, 3 for maxdepth, and 0.1 for nu. This model should predict with about 82.39% accuracy. When testing the model, it should yield slightly less accurate results than the Random Forest Model.

```
#Testing Ada Model
predictions_ada <- predict(model_ada, test)
confusionMatrix(predictions_ada, test$Survived)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Passed Survived
##   Passed      152       33
##   Survived     12       69
##
##                Accuracy : 0.8308
##                  95% CI : (0.7803, 0.8738)
##     No Information Rate : 0.6165
##     P-Value [Acc > NIR] : 2.211e-14
##
##                   Kappa : 0.6277
##  Mcnemar's Test P-Value : 0.002869
##
##             Sensitivity : 0.9268
##             Specificity : 0.6765
##          Pos Pred Value : 0.8216
##          Neg Pred Value : 0.8519
##              Prevalence : 0.6165
##          Detection Rate : 0.5714
##    Detection Prevalence : 0.6955
##       Balanced Accuracy : 0.8016
##
##        'Positive' Class : Passed
##
```

This model predicted with about 83.08% accuracy. Althouh its predictions were more successful than the 82.39% shown in the model, its accuracy was still close to what was predicted. In addition, the Sensitivity or True Positive Rate was 92.68% and the Specificity or True Negative Rate was 67.65%. This Ada Model, like the Random Forest Model, is better at predicting who would pass away on the Titanic than it is at predicting who would survive.

Conclusion

Since the Random Tree Model estimated that it would predict with higher accuracy than the Ada Model and showed that it actually predicted with better accuracy than the Ada model when tested, the Random Tree Model is the model that will be used on our outside data. Although there were many trends that may have ensured survival on the Titanic, more than likely a person would die if they were a passenger. Women survived more but still many died. The same is true for those who paid higher fares and were young. Still we will test how this model performs when given data from another dataset.

Testing Random Forest Model

Loading Data Set

```
#Loading Data Set
#Turning Pclass to Factor Variable
#Imputing NA Values
final_test  <- read.csv("test.csv", header = TRUE)
final_test$Pclass <- factor(final_test$Pclass,
    labels = c("First", "Second", "Third"))
final_test <- kNN(final_test)
final_test <- final_test %>%
  select(-(PassengerId_imp:Embarked_imp))

dim(final_test)
```

```
## [1] 418  11
```

Here the dataset to be used for predictions is loaded and the `Pclass` variable is converted to a factor for testing. In addition, knn imputation is used for all missing values in the `titanic` dataset. This imputation was used mostly in the Age column and for one passenger whose Fare was missing. Finally, the dimensions for the dataset are shown. So there will be 418 predictions for this dataset.

```
#Making Predictions
#Adding Survived Column
#Selecting Columns for Submission
#Changing Survived Column back to Factors of 0 and 1
#Writing CSV file for submission. Removing row names
predictions_rf <- predict(model_rf, final_test)
final_test$Survived <- predictions_rf
submission_rf <- final_test[, c("PassengerId", "Survived")]
submission_rf$Survived <- factor(submission_rf$Survived, labels = c(0, 1))
write.csv(submission_rf, "titanic_rf_predictions.csv", row.names = FALSE)
length(predictions_rf)
```

```
## [1] 418
```

There were 418 predictions made and stored in the variable `predictions_rf`. For the purpose of submission to the Kaggle competition, a heading was added to the dataframe and it was saved as a csv file.