Titanic survivors classification prediction

Data sourced from Kaggle https://www.kaggle.com/c/titanic/data (https://www.kaggle.com/c/titanic/data)

Sarah Hall

3/11/2018

```
#Load librarys
library(caret)
library(randomForest)
library (tidyverse)
library(fields)
library(s20x)

#Load data files (downloaded from Kaggle)
TitanicTrain.df <- read.csv("train.csv", header = TRUE, stringsAsFactors = FALSE)
TitanicTest.df <- read.csv("test.csv", header = TRUE, stringsAsFactors = FALSE)
```

View data sets:

```
head(TitanicTrain.df)
```

```
##   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
```

```
head(TitanicTest.df)
```

```
##   PassengerId Pclass                                          Name    Sex
## 1         892      3                              Kelly, Mr. James   male
## 2         893      3              Wilkes, Mrs. James (Ellen Needs) female
## 3         894      2                     Myles, Mr. Thomas Francis   male
## 4         895      3                              Wirz, Mr. Albert   male
## 5         896      3 Hirvonen, Mrs. Alexander (Helga E Lindqvist) female
## 6         897      3                    Svensson, Mr. Johan Cervin   male
##    Age SibSp Parch  Ticket    Fare Cabin Embarked
## 1 34.5     0     0  330911  7.8292              Q
## 2 47.0     1     0  363272  7.0000              S
## 3 62.0     0     0  240276  9.6875              Q
## 4 27.0     0     0  315154  8.6625              S
## 5 22.0     1     1 3101298 12.2875              S
## 6 14.0     0     0    7538  9.2250              S
```

Quick data quality assessment:

```
summary(TitanicTrain.df)
```

```
##   PassengerId       Survived          Pclass          Name
##  Min.   :  1.0   Min.   :0.0000   Min.   :1.000   Length:891
##  1st Qu.:223.5   1st Qu.:0.0000   1st Qu.:2.000   Class :character
##  Median :446.0   Median :0.0000   Median :3.000   Mode  :character
##  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
##
##      Sex                 Age            SibSp           Parch
##  Length:891         Min.   : 0.42   Min.   :0.000   Min.   :0.0000
##  Class :character   1st Qu.:20.12   1st Qu.:0.000   1st Qu.:0.0000
##  Mode  :character   Median :28.00   Median :0.000   Median :0.0000
##                     Mean   :29.70   Mean   :0.523   Mean   :0.3816
##                     3rd Qu.:38.00   3rd Qu.:1.000   3rd Qu.:0.0000
##                     Max.   :80.00   Max.   :8.000   Max.   :6.0000
##                     NA's   :177
##     Ticket               Fare           Cabin             Embarked
##  Length:891         Min.   :  0.00   Length:891         Length:891
##  Class :character   1st Qu.:  7.91   Class :character   Class :character
##  Mode  :character   Median : 14.45   Mode  :character   Mode  :character
##                     Mean   : 32.20
##                     3rd Qu.: 31.00
##                     Max.   :512.33
##
```

Age contains NA's, will need to impute these if using age as a feature.

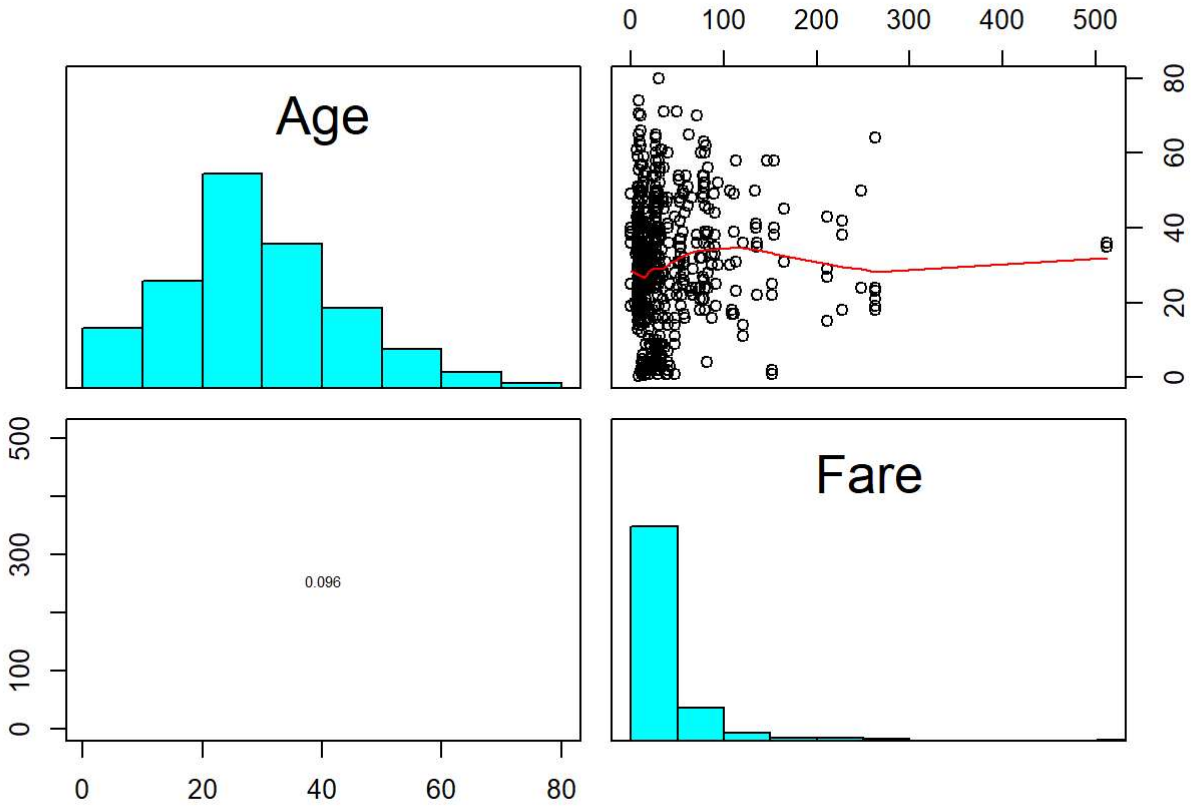
Exploratory data analysis

Test for useful features:

```
names(TitanicTrain.df)
```

```
##  [1] "PassengerId" "Survived"    "Pclass"      "Name"        "Sex"
##  [6] "Age"         "SibSp"       "Parch"       "Ticket"      "Fare"
## [11] "Cabin"       "Embarked"
```

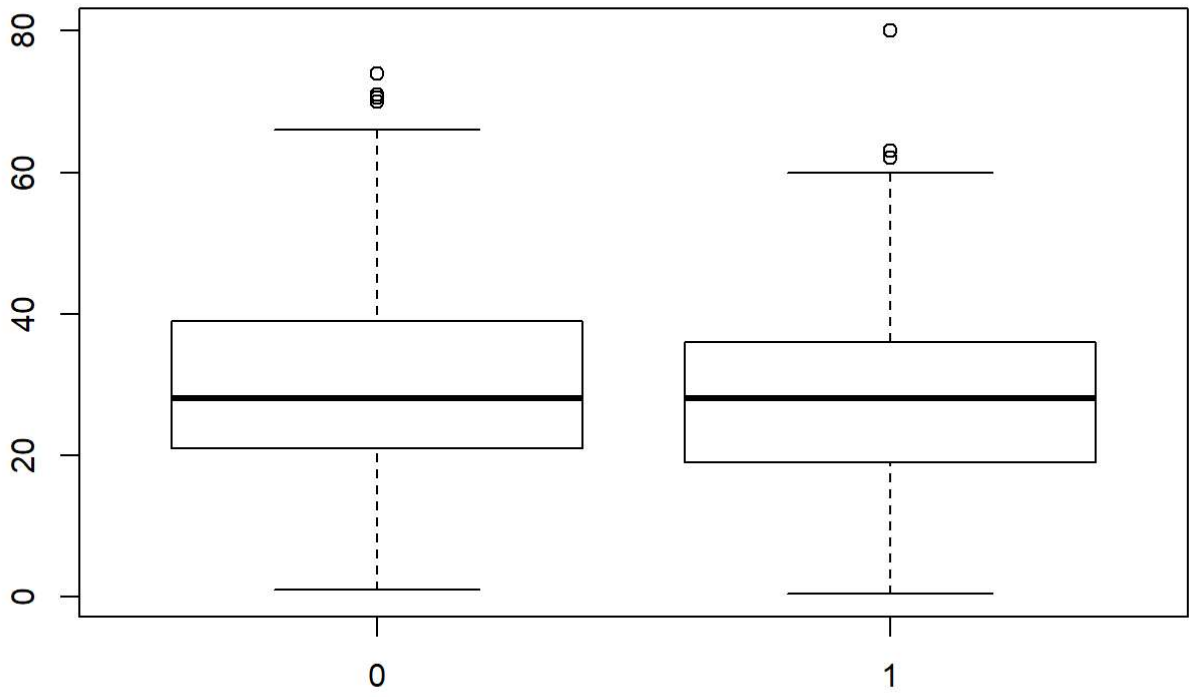
```
qt.df <- TitanicTrain.df %>%
  dplyr::select(Age,Fare)

pairs20x(qt.df)
```



Age and Fare don't seem particualrly correlated.

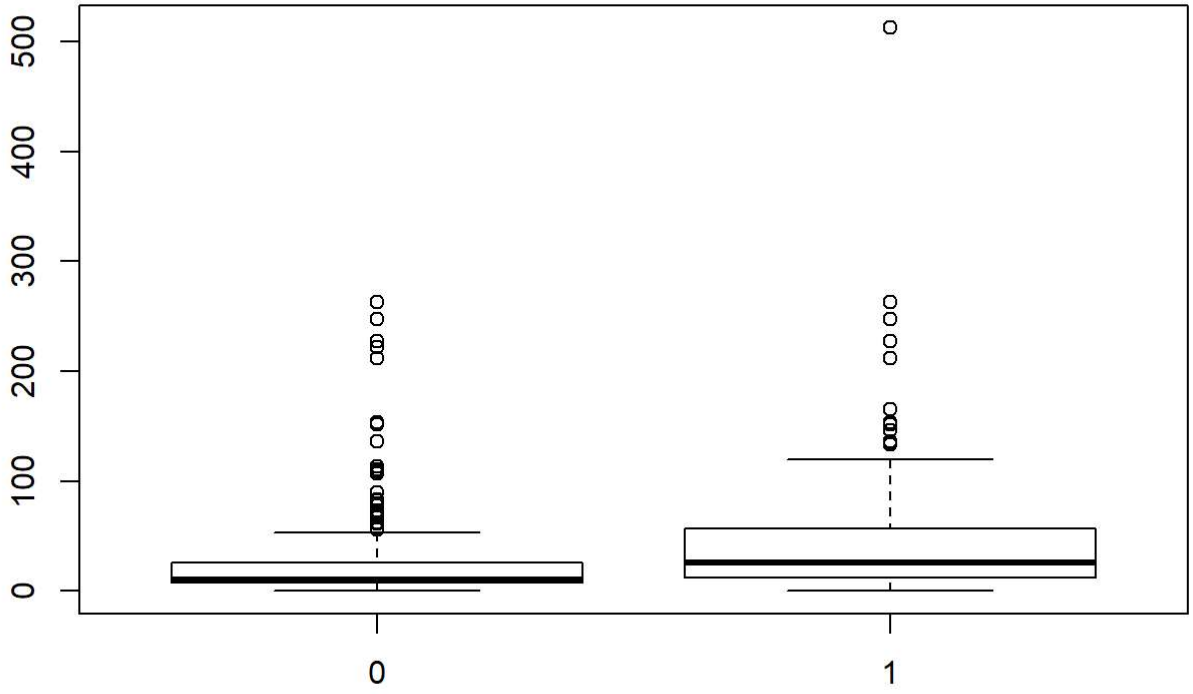
Explore continuious variables:

```
boxplot(TitanicTrain.df$Age ~ TitanicTrain.df$Survived)
```



Age unlikely to be a useful predictor, will not use as feature.

```
boxplot(TitanicTrain.df$Fare ~ TitanicTrain.df$Survived)
```



Fare could be useful, will use as feature.

Explore categorical variables:

```
prop.table(table(TitanicTrain.df[,c("Survived", "Pclass")]),2)
```

```
##          Pclass
## Survived         1         2         3
##        0 0.3703704 0.5271739 0.7576375
##        1 0.6296296 0.4728261 0.2423625
```

Pclass could be a useful predictor due to the survival ratio of each class, will use as a feature.

```
prop.table(table(TitanicTrain.df[,c("Survived", "Sex")]),2)
```

```
##          Sex
## Survived    female      male
##        0 0.2579618 0.8110919
##        1 0.7420382 0.1889081
```

Sex could be a useful predictor, higher % of women survived than men, will use as a feature.

```
prop.table(table(TitanicTrain.df[,c("Survived", "SibSp")]),2)
```

```
##          SibSp
## Survived         0         1         2         3         4         5
##        0 0.6546053 0.4641148 0.5357143 0.7500000 0.8333333 1.0000000
##        1 0.3453947 0.5358852 0.4642857 0.2500000 0.1666667 0.0000000
##          SibSp
## Survived         8
##        0 1.0000000
##        1 0.0000000
```

SibSp could be useful, though unlikely. Will use as a feature initially.

```
prop.table(table(TitanicTrain.df[,c("Survived", "Parch")]),2)
```

```
##          Parch
## Survived         0         1         2         3         4         5
##        0 0.6563422 0.4491525 0.5000000 0.4000000 1.0000000 0.8000000
##        1 0.3436578 0.5508475 0.5000000 0.6000000 0.0000000 0.2000000
##          Parch
## Survived         6
##        0 1.0000000
##        1 0.0000000
```

Parch could be useful, though unlikely. Will use as a feature initially.

```
prop.table(table(TitanicTrain.df[,c("Survived", "Embarked")]),2)
```

```
##          Embarked
## Survived                    C         Q         S
##        0 0.0000000 0.4464286 0.6103896 0.6630435
##        1 1.0000000 0.5535714 0.3896104 0.3369565
```

Embarked looks useful, will use as a feature. Name, Cabin, and Ticket will not be used as features.

Initial model train: random forest with six features

```
# Convert Survived to factor
TitanicTrain.df$Survived = factor(TitanicTrain.df$Survived)
# Set a random seed
set.seed(123)
# Train the model - random forest
model <- train(Survived ~ Pclass + Sex + SibSp +
                Parch + Embarked + Fare,
               data = TitanicTrain.df,
               method = "rf",
               trControl = trainControl(method = "cv", number = 5))
print(model)
```

```
## Random Forest
##
## 891 samples
##   6 predictor
##   2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 713, 712, 713, 713, 713
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##   2     0.7990835  0.5573193
##   5     0.7901387  0.5491407
##   8     0.7822798  0.5346626
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 2.
```

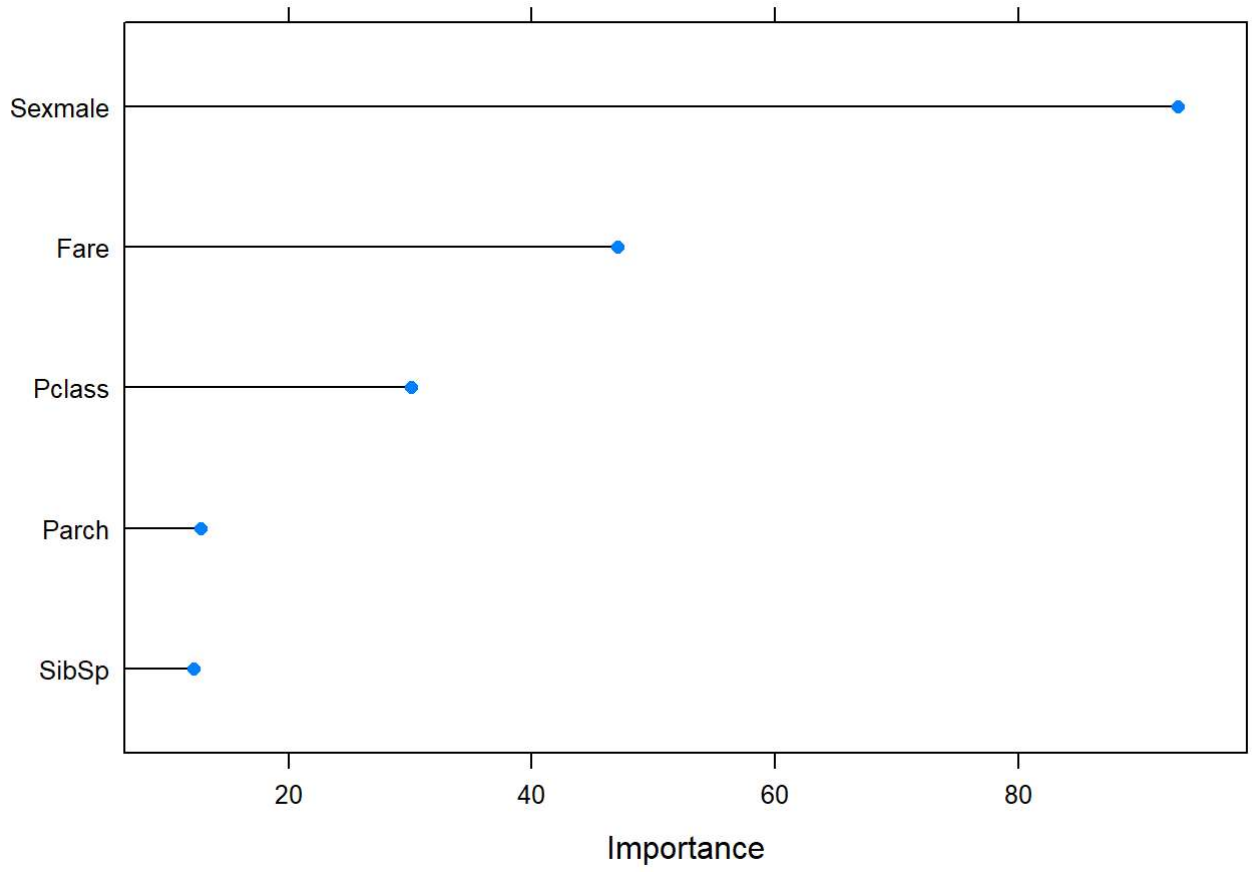
79% accuracy looks good for a first stab.

Test model specific variable importance

```
varImp(model, useModel = TRUE, scale = TRUE)
```

```
## rf variable importance
##
##           Overall
## Sexmale   100.000
## Fare       49.042
## Pclass     30.349
## Parch      11.214
## SibSp      10.518
## EmbarkedC   2.638
## EmbarkedS   2.516
## EmbarkedQ   0.000
```

```
plot(varImp(model, useModel = TRUE, scale = FALSE), top = 5)
```



Sex is a very important feature, Fare and Pclass are important and Parch and SibSp may be important (about the same impact as each other).

Test two new models, both random forest. One with Sex, Fare and Pclass as features, the other with SibSp and Parch as well.

Train random forest model with Sex, Fare,Pclass, SibSp and Parch

```
model.FiveFt <- train(Survived ~ Pclass + Sex + SibSp +
                        Parch + Fare,
                      data = TitanicTrain.df,
                      method = "rf",
                      trControl = trainControl(method = "cv", number = 5))
print(model.FiveFt)
```

```
## Random Forest
##
## 891 samples
##   5 predictor
##   2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 713, 713, 713, 713, 712
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##   2     0.8058942  0.5779740
##   3     0.8215806  0.6177332
##   5     0.8070052  0.5870401
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 3.
```

Accuracy improved to 82%, good improvement.

Train random forest model with Sex, Fare and Pclass

```
model.ThreeFt <- train(Survived ~ Pclass + Sex + Fare,
                       data = TitanicTrain.df,
                       method = "rf",
                       trControl = trainControl(method = "cv", number = 5))
```

```
## note: only 2 unique complexity parameters in default grid. Truncating the grid to 2 .
```

```
print(model.ThreeFt)
```

```
## Random Forest
##
## 891 samples
##   3 predictor
##   2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 712, 714, 713, 713, 712
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##   2     0.8126462  0.5890232
##   3     0.8148745  0.6031973
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 3.
```

Accuracy of 81% is a slight improvement on initial 6 feature model, but the 5 feature model is best so far.

Test with different model: recursive partitioning and regressive trees model

```
model.rpart <- train(Survived ~ Pclass + Sex + SibSp +
                       Parch + Embarked + Fare,
                     data = TitanicTrain.df,
                     method = "rpart",
                     trControl = trainControl(method = "cv", number = 5))
print (model.rpart)
```

```
## CART
##
## 891 samples
##   6 predictor
##   2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 713, 713, 712, 713, 713
## Resampling results across tuning parameters:
##
##   cp          Accuracy   Kappa
##   0.01023392  0.8114619  0.5809705
##   0.03070175  0.7901513  0.5480892
##   0.44444444  0.7114996  0.3104391
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.01023392.
```

Accuracy of 81% about on par with three feature random forest model.

Five feature random forest is still best.

Predict all four models on the test set of data

```
summary(TitanicTest.df)
```

```
##   PassengerId         Pclass          Name              Sex
##  Min.   : 892.0   Min.   :1.000   Length:418         Length:418
##  1st Qu.: 996.2   1st Qu.:1.000   Class :character   Class :character
##  Median :1100.5   Median :3.000   Mode  :character   Mode  :character
##  Mean   :1100.5   Mean   :2.266
##  3rd Qu.:1204.8   3rd Qu.:3.000
##  Max.   :1309.0   Max.   :3.000
##
##       Age            SibSp            Parch           Ticket
##  Min.   : 0.17   Min.   :0.0000   Min.   :0.0000   Length:418
##  1st Qu.:21.00   1st Qu.:0.0000   1st Qu.:0.0000   Class :character
##  Median :27.00   Median :0.0000   Median :0.0000   Mode  :character
##  Mean   :30.27   Mean   :0.4474   Mean   :0.3923
##  3rd Qu.:39.00   3rd Qu.:1.0000   3rd Qu.:0.0000
##  Max.   :76.00   Max.   :8.0000   Max.   :9.0000
##  NA's   :86
##       Fare            Cabin             Embarked
##  Min.   :  0.000   Length:418         Length:418
##  1st Qu.:  7.896   Class :character   Class :character
##  Median : 14.454   Mode  :character   Mode  :character
##  Mean   : 35.627
##  3rd Qu.: 31.500
##  Max.   :512.329
##  NA's   :1
```

NA value in “Fare” column. Fix this by imputing with mean of “Fare” column.

```
TitanicTest.df$Fare <- ifelse(is.na(TitanicTest.df$Fare), mean(TitanicTest.df$Fare, na.rm = T
RUE), TitanicTest.df$Fare)

# predict on test set
TitanicTest.df$Survived.RF_6FT <- predict(model, newdata = TitanicTest.df)
TitanicTest.df$Survived.RPRT_6FT <- predict(model.rpart, newdata = TitanicTest.df)
TitanicTest.df$Survived.RF_5FT <- predict(model.FiveFt, newdata = TitanicTest.df)
TitanicTest.df$Survived.RF_3FT <- predict(model.ThreeFt, newdata = TitanicTest.df)
```

Create output files for Kaggle

```
output_RF_6FT <- TitanicTest.df %>%
  dplyr::select(PassengerId,Survived.RF_6FT)
colnames(output_RF_6FT)[2] <- "Survived"
write.csv(output_RF_6FT, file = "output_RF_6FT.csv", row.names = FALSE)

output_RPRT_6FT <- TitanicTest.df %>%
  dplyr::select(PassengerId,Survived.RPRT_6FT)
colnames(output_RPRT_6FT)[2] <- "Survived"
write.csv(output_RPRT_6FT, file = "output_RPRT_6FT.csv", row.names = FALSE)

output_RF_5FT <- TitanicTest.df %>%
  dplyr::select(PassengerId,Survived.RF_5FT)
colnames(output_RF_5FT)[2] <- "Survived"
write.csv(output_RF_5FT, file = "output_RF_5FT.csv", row.names = FALSE)

output_RF_3FT <- TitanicTest.df %>%
  dplyr::select(PassengerId,Survived.RF_3FT)
colnames(output_RF_3FT)[2] <- "Survived"
write.csv(output_RF_3FT, file = "output_RF_3FT.csv", row.names = FALSE)
```

Summary

Results from Kaggle:

- Random forest with 6 features: 77.9% accuracy
- Random forest with 5 features: 79.4% accuracy
- Random forest with 3 features: 77.5% accuracy
- Recursive partitioning and regressive trees model with 6 features: 78.4% accuracy

As expected, the random forest model with 5 features (Sex, Pclass, Fare, SibSp and Parch) performed best.

No model provided a significant imporvement on the initial model though, so next steps will be required to improve my score.

Next steps to test:

- Is random forest the best model?
- Is there any interraction between Embarked and Pclass or Fare?
- Can overfitting be reduced?
    - Are Pclass and Fare correlated?
    - Are SibSp and Parch correlated?