Score: 0.78469 Model: randomForest

The first thing I did was try to clean the data a bit. In both the "train" data and the "test" there are a number of 'NA' fields (fields missing data). I decided to disregard things like passenger name and fare. Passenger name because it's very noisy, fare because it's missing a lot of data and looks like it means "balance paid" rather than individual fare. So for a family of four, if one person paid for everything, their fare might look really high even for 3rd class passage. So the two columns with missing data are "age" and "embarked."

I looked for the median age of the passengers and used this median to fill in any NA ages. I then looked for the most common embarkation code and used that to fill in any NA embarked fields.

After the massaging, here's the code I used.

> library(randomForest)

#now I load in both the "train" and "test" data frames

> train <- read.csv("C:/train.csv")

> test <- read.csv("C:/test.csv")

#Now, randomForest can either model a regression (continuous number) or a classification (yes/no).

#This competition is to predict survival. Either the person survived or they didn't: yes/no.

#So, I want to make sure randomForest models a classification. To do this, I tell the program

#that the "Survived" column (our dependent variable) is a factor, not a number.

> train$Survived <- as.factor(train$Survived)

#Next, I decide what predictors (independant variables) I want to train my model with.

#I decided to use Pclass, Sex, Age, SibSp, Parch, and Embarked as predictors.

#Additionally, I decide to include the relationships between Pclass:Sex and Pclass:Age and Age:Sex

#just for kicks. I think women are more likely to survive than men ("Women and children first!")

#but I think it's possible that a rich woman (Pclass = 1) might be more likely to survive than a

#poor woman. So we'll model things like this in and see what we get.

#Before I do anything else, I want to set a "seed" so that others can reproduce my results.

#randomForest is RANDOM. If I don't set a seed value, then each time I run my model I might

#get a slightly different result.

> set.seed(107)

#Now I build my model

> model <- randomForest(Survived ~ Pclass + Sex + Age + SibSp + Parch + Embarked + Pclass:Sex + Pclass:Age + Age:Sex, data=train, ntree=20000)

#Now I use the model to predict against test data. This enters the prediction as a new column

#called “Survived” in the test data frame.

> test$Survived <- predict(model, newdata=test, type=’class’)

#Finally I save my updated test data frame as a .csv file

> write.csv(test, file=”c:/R/predictions.csv”, row.names=F)

#Now, outside of R I go into my newly saved "predictions.csv" file and delete all the columns

#except for passenger ID and Survived. I then upload this to Kaggle to see what my score is.

Score for this model: 0.78469