Decision Trees

Cody Frisby

11/5/2017

Quickly looking at the data, it became apparent that some of the predictor variables are highly correlated with each other. Also, some variables are missing MOST of the values in addition to being very messy when reading into R. The **Name** variable appears to contain the title of each passenger. I'd like exclude **Name** from the analysis but perhaps include the title. Parsing out the title from each name (In R that can be done by running the `gsub` on the **Name** variable) I get a new variable called **title**. **Cabin** is another categorical variable that contains way too levels. We could perform a similar operation to **Cabin** that we did to **Name**, extracting out the meaningful/alike levels, reducing the number of levels to a value much less than 186. Additionally it contains 1013 missing values. We would also need handle all these missing values to include it in the analysis so for the purposes of this study I will exclude it.

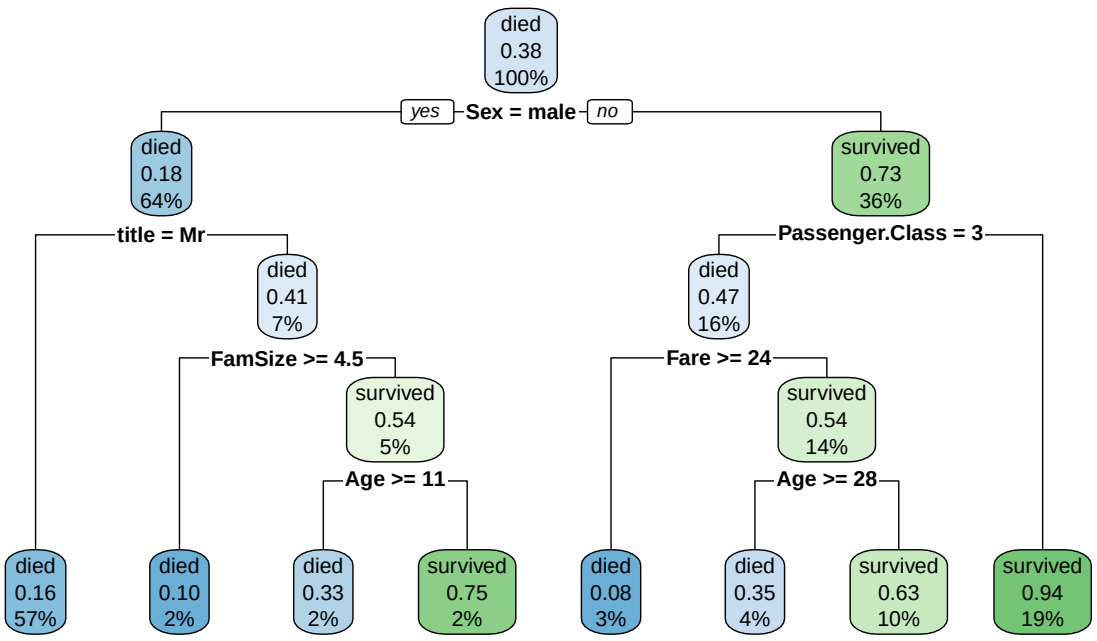
For the **Fare** variable, there appears to be a few extreme outliers. For this analysis, I've chosen to exclude any values that are greater than 300.

Using a simple random sample of 75% of the data, I've split it into two data sets, train and test (train being the larger of the two).

Below is a table showing this new variable, **title**, by **Sex**. As can be seen, title and sex are highly correlated.

	Master	Miss	Mr	Mrs	Rare
female	0	263	0	197	4
male	61	0	754	0	25

The below plot is a visual representation of the model. Each box contains three items: the predicted class, predicted probability, and the percentage of observations in that node.

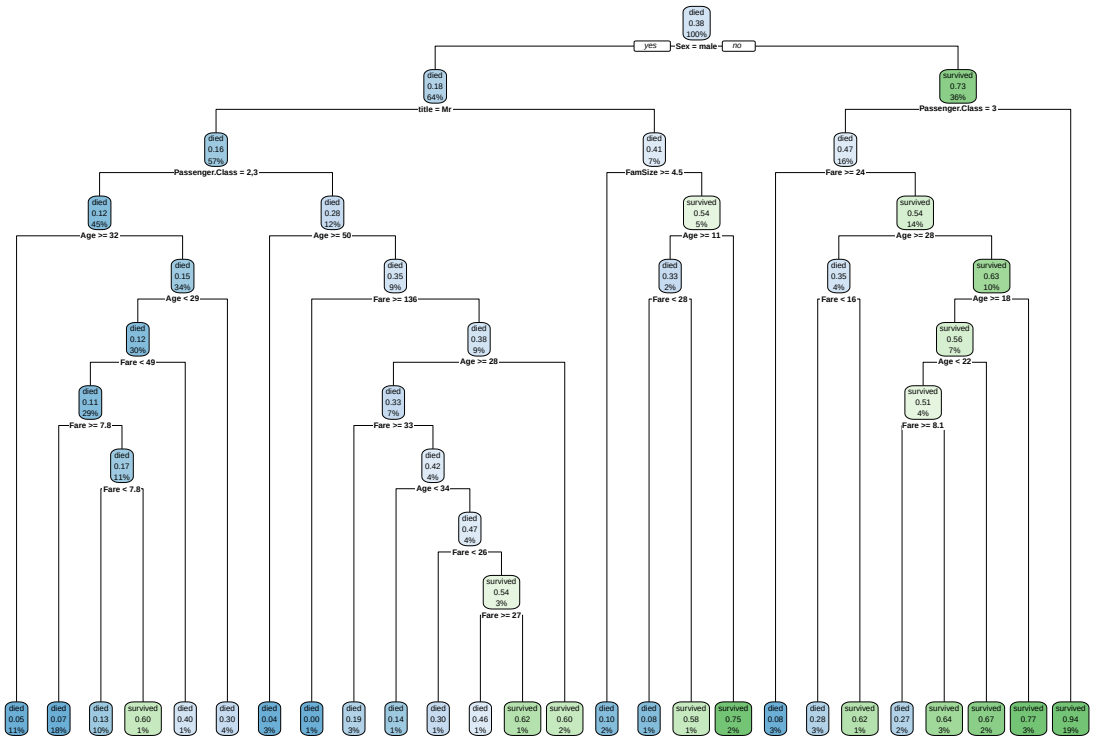


The first decision rule is **Sex** of the passenger (which is correlated to **title** for the most part). Go to the left if "male". For male, the next decision rule is **title** while for female it is **Passanger.Class**. It can be easily seen that if the passenger was female AND from class 1 or 2 they had a 94 percent of survival. This model is fit with the defaults from the `rpart` function and the control arguments for this function, `rpart.control`, were $cp = 0.01$ which is the default.

This model does OK when using it to predict the probabilities of survival on the test data set. Using 0.5 as the cutoff value for whether or not someone survives, we get the following matrix.

	PredictDie	PredictSurvive
died	177	23
survived	43	84

As can be seen, this model isn't AWESOME. We miss-classify $23 + 43 = 66$ of the observations from the test set, for a miss-class error of 0.2018349. At least it's better than a coin flip. Now, if we changed some of the parameters to the model and then testing our model, we might see a contrast to the above result. Changing `cp` argument to 0.001 (basically this means the next attempted split must reduce the overall lack of fit by 0.001 before attempting the split. If you recall, for the first model this number was 0.02. The smaller the number here, the more likely we are to over-fit.) will force the model to include more nodes than before.

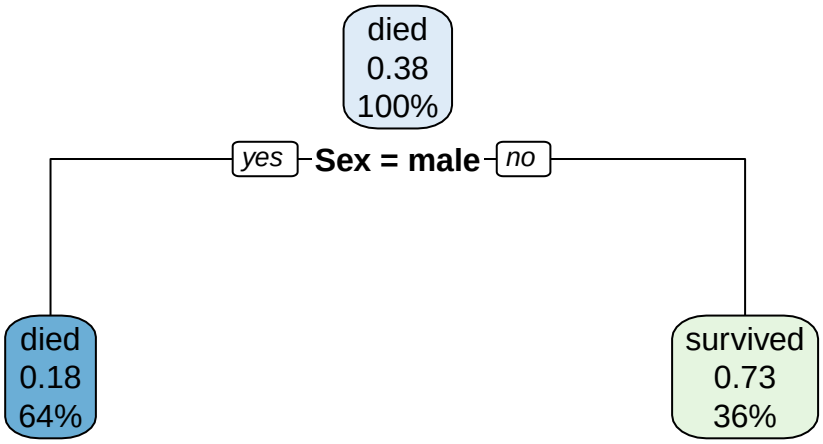


How well does this model perform on the test set?

	PredictDie	PredictSurvive
died	168	32
survived	37	90

And you can see that our miss-class error hasn't moved much. We miss-classify $32+37 = 69$ of the observations from the test set, for a miss-class error of 0.2110092.

On the other extreme, not including enough information, we can also see that will will have a lot of error in our predictions. Setting `cp` $= 0.05$ we get the model that is visualized below. Basically, we are predicting survival based on the *title* of the passanger.

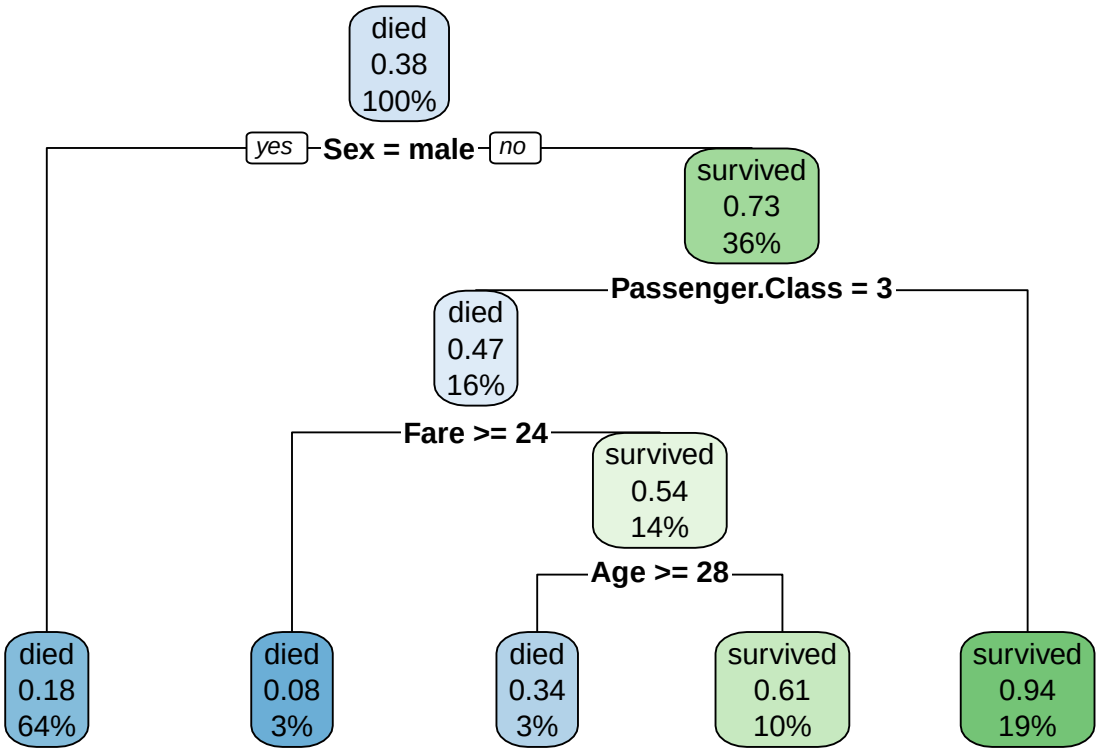


As can be seen, this model is very simple. It simply predicts survival based on the gender of the passanger.

	PredictDie	PredictSurvive
died	168	32
survived	37	90

We miss-classify $32 + 37 = 69$ of the observations from the test set, for a miss-class error of 0.2110092. We do slightly worse with this more simplified model. I'd stick with the first one. It's more simple than the second and does better than the last one.

One last thing. Was the extra effort to create the **title** variable worth it? Well, if it predicts better, then yes. Let's check that by fitting the model without **title**.



Looks like it performs a little worse on the test set when excluding **title**, $error = 0.2110092$. I would have guessed that it would be the same.