Andrew Hautau

Deriving Knowledge From Data At Scale

Kaggle Titanic Competition

GitHub Repo for project codebase: <https://github.com/CrossTheStreams/kaggle_titanic>

Part 1

My initial days working on this assignment focused exclusively on data preparation and casual analysis. I converted categorical variables such as “Sex”, “Cabin”, and “Pclass” to logical values and tried to gain an appreciation for which variables might be of the most immediate use. Throughout this project, I stuck with R as my preferred tool.

During my first passes at analysis, it became clear which variables would require more effort in order to contribution to any predictive model. Many observations in the data lacked “Age” values, so imputation would be required in order to use the variable. On the other hand, it wasn’t readily apparent what I could do with variables such as “name” and “ticket”.

After this data prep, I had a grab bag of logical variables to work with— my next goal was to find an efficient way to detect signal among them. To that end, I created a function that took in a sample of the training data observations and plotted R2 scores and p values returned from chi squared tests, along with a Pearson correlation of each variable and the target “survived” attribute.

After several iterations with this method, it appeared to be a safe assumption that “Sex”, “first.class”, and “third.class” carried signal for a first predictive model. To complement this feature selection, I also found positive Pearson correlation between “Survived” and “Fare” of 0.26. A simply boxplot reveals that surviving passengers have a much wider distribution of fare values that leaned towards higher values.

The predictive model I chose to use was an SVM model. Using the “female”,”first.class”,”third.class”, and “Fare” variables, my first submission to Kaggle achieved a score of 0.77512 on the test set.



Part 2

Finding improvement in my score would turn out to be difficult. All but a few of my following submissions using the SVM model failed to score higher on the leaderboard, and improvement was marginal.

At first, I tried to impute Age and use it to directly inform predictions. Analysis of observations with age values indicated that there was some correlation with survival among young crewmembers. My method for imputing age used the “Parch” and “SibSp” variables as proxies, iterating over observations without age values and taking the mean ages of the observations with the same “Parch” and “SibSp” values.

In an attempt at feature creation, I used kmeans clustering to create features based on the other features I had previously made predictions on. While it turned into an interesting investigation, I think this simply made my model over-fit the training set slightly (although, the model did not performance significantly worse, either).

Eventually, I did achieve a slight improvement in my score by creating a “Young” categorical feature. I gave a true value for this attribute in any observation that either had an age below 16 or had a “Name” attribute that contained the title “Master” (consistently used for boys in the data). While my imputed “Age” attribute did not improve predictions directly, it acted as a useful tool nonetheless.

Part 3

I found another minor improvement in my prediction on the test set from binning the “Fare” variable and assigning to logical variable, boosting my score to 0.78469 (which does manage to edge out a “benchmark” on the contest leaderboard). I binned the Fare based on quantiles, creating four new variables.

I attempted to use a random forest model with the same set of feature settings. Random forest using the same attributes for prediction performed slightly worse than the SVM model on the test set. Further, I made a simple ensembling method the incorporated both the SVM model and the random forest to see if they could compliment each other on the test set. My ensemble first trained both models and made a decision on which to use on the test and training sets depending on their performance during training. I took the mean values of rows that predicted survival for each model and then selected a model for the ensemble output based on the least error between the means. Interestingly, training the ensemble and it’s models on a sample of the training set improved performance. Further nuance in this approach might yet yield results. Further, I’ve started to research other, more automated approaches to ensembling in R, like using the [caretEnsemble](https://github.com/zachmayer/caretEnsemble) package.

Kaggle Submissions

