Data Science Blog Post – Module 5

Evan Okin

As a data scientist, it is important to understand how to use machine learning algorithms for prediction. It enables us to answer important business problems and create actionable change. Machine learning algorithms can be used to predict a yes/no outcome or other, non-binary, classification outcomes. An example of a problem that I decided to analyze was a problem posted to Kaggle.com, where many data sets exist for aspiring data scientists. The data is about pet adoption. The problem that I wanted to solve was, can I create a machine learning algorithm that can predict whether or not a dog will be adopted?

The first step was to load in data, and preview it to make sure that it looked right. My question focused on dogs, so I removed cats from my analysis. Then, I wanted to test if the data had missing values. The dataset was generally cleaner than I expected. There were 649 missing values for "Name", which is a significant number of missing values and without studying NLP, I decided to drop this column altogether. It is also likely that name is not as indicative of adoption as compared with other traits, because many owners just change the name given to the dog that they adopt. While "Name" had many missing values, no other feature had a significant amount of missing values.

The next step was to explore the data. Regarding adoption rates, 2% of dogs were adopted the same day as they were taken into adoption, 18% were adopted within 1-7 days, 24% were adopted within 8-30 days, 27% were adopted within 31-100 days, and 30% were not adopted after 100 days. Thus, I decided to bucket any adoption within 100 days as "Yes, Adopted" (binary value of 1), which made up 70% of all values, and any not adopted within 100 days as "No, Not Adopted" (binary value of 0), which made up 30% of all values. I found that the most common breed of dogs that get adopted are Yellow Labrador Retrievers, Pekingeses, and Silky Terriers. Dogs that are black or brown have a hard time getting adopted.

As many features were categorical in nature, this increased the amount of features from 24 to 167 (doing so increased accuracy for my best model by 6%). I created a training set to "train" the model on, and a testing set to "test" the model with, to understand model accuracy. I used a 70% training set and a 30% test set in order to give ample size to the test set. The data has enough rows that 70% should be enough to train the data well. I kept a 70%/30% split, consistent for all of the algorithms. I also kept the same random seed for all models to remove variability between the models, so that they could be better compared on an apples-to-apples basis. I used sci-kit learn to run all of the algorithms, an essential, built-in package with built in machine-learning functionality.

The first algorithm that I tested was a logistic regression, which yielded a 76% accuracy. Next, K-Nearest Neighbors yielded 76% and after that, Decision Trees yielded 71%. While the accuracy was low for Decision Trees, I achieved 77% accuracy under the Random Forest (ensemble method) algorithm, which improved accuracy by combining 1,000 decision trees. I iterated on different combinations of estimators and different combinations of max depth. I achieved the best accuracy under 1,000 estimators and a max-depth of 15. This model would

outperform the other ML algorithms as well. I achieved 72% accuracy under Support Vector Machines and 72% accuracy with Principal Component Analysis with 4 features.

I analyzed a confusion matrix for the Random Forest method, our best model. This tells us, in addition to the accuracy, the amount of false positives and false negatives. The rough breakdown for the model was 77% accurate, 19% false positive (predicted adoption but not actually adopted), and 4% false negative (predicted no adoption but actually adopted). From my perspective for this business problem, I would say that a "false positive" is a worse outcome, so we have to be careful in our recommendations, knowing that our model will give a "false positive" result nearly 1 out of every 5 times.

The following are steps we can take to increase the likelihood that dogs will get adopted. We can post more photos and videos to the page. We can make sure that the dog gets vaccinated and sterilized. While out of our control, it's helpful to know that breed makes a big difference, and that younger dogs are more likely to be adopted.

This analysis could be improved going forward in several ways. I could provide a more extensive list of features that influence adoption, such as specific breeds or locations. I can continue to fine-tune our model parameters. For example, I found an optimal outcome for random forests with a max depth of 15, but only looped through 20 possible max depths. There could also be noise associated with the random seed that I used, so I could use more extensive cross validation. I could analyze our dataset for cats to see if it has similar predictive power as for dogs. I could consider "bucketing" adoption speeds to see if it helps prediction – I could even have left the initial 5 buckets instead of performing a binary yes/no prediction.