Predicting the amount of time a dog will spend at a shelter before being adopted

Amanda White

Problem Statement

Annually an estimated 3.3 million dogs enter an animal shelter in the United States, and of that 3.3 million only 1.6 million dogs are adopted (ASPCA). Of the unadopted dogs, 670,000 are euthanized and the rest remain in shelters. Animal shelters are often stretched thin caring for these unadopted dogs. The goal of this project is to develop a model that can estimate how long it will take for a dog to be adopted. With an accurate prediction of how long a dog will be in their care, a shelter can better advertise and promote these dogs to improve adoption rates, as well as help shelter staff better estimate what resources they will need.

Data Exploration and Cleaning

Using data from the Austin Animal Center, I created a model that can estimate how long a dog will spend in the shelter before getting adopted. The initial dataset contained 7 years worth of information about all the animals who entered the shelter, but this project only focuses on dogs. Each row of the data set is a dog and some of their attributes, including: color, breed, age, intake condition (healthy, pregnant, aged, etc..), intake type (stray, owner surrender, public assist), sex, and the days in the shelter. I narrowed the data set down to only dogs who were adopted and dropped all other outcome types. I did not want to include dogs who were returned to their owners or transferred because there is a chance these dogs were never eligible for adoption. Only outcome_sutype had many missing values and so I dropped the entire column. Beyond that, only three rows had missing values, and since it was so few I just dropped these rows.

Exploring the data further, I found that there were 1,377 unique breeds most of the dogs had two breeds listed or included the word “mix”. To reduce dimensionality I aggregated the breeds by only keeping the first breed listed which reduced the number of unique breeds to 186. I used the same approach with the color column that had 258 unique color combinations and reduced it to 35. I did this under the assumption that the first breed and color listed was the dog’s primary breed/color. To further reduce noise in the data I dropped the least common breeds that make up 20% of the data and the colors that make up ~4% of the data. I ended with the most frequent 27 breeds and 12 colors.

Preprocessing and Modeling

To prepare the data for modeling, I encoded the categorical variables and created dummy variables for these columns. The final dataframe was made of the following columns: intake condition, intake type, breed, color, sex, age, and time in shelter.

For the target variable (time in shelter) I created four categories (<4 days, 4-7 days, 7-23 days, and 23+ days) I based my categories on the data quantiles to make them as even as possible.

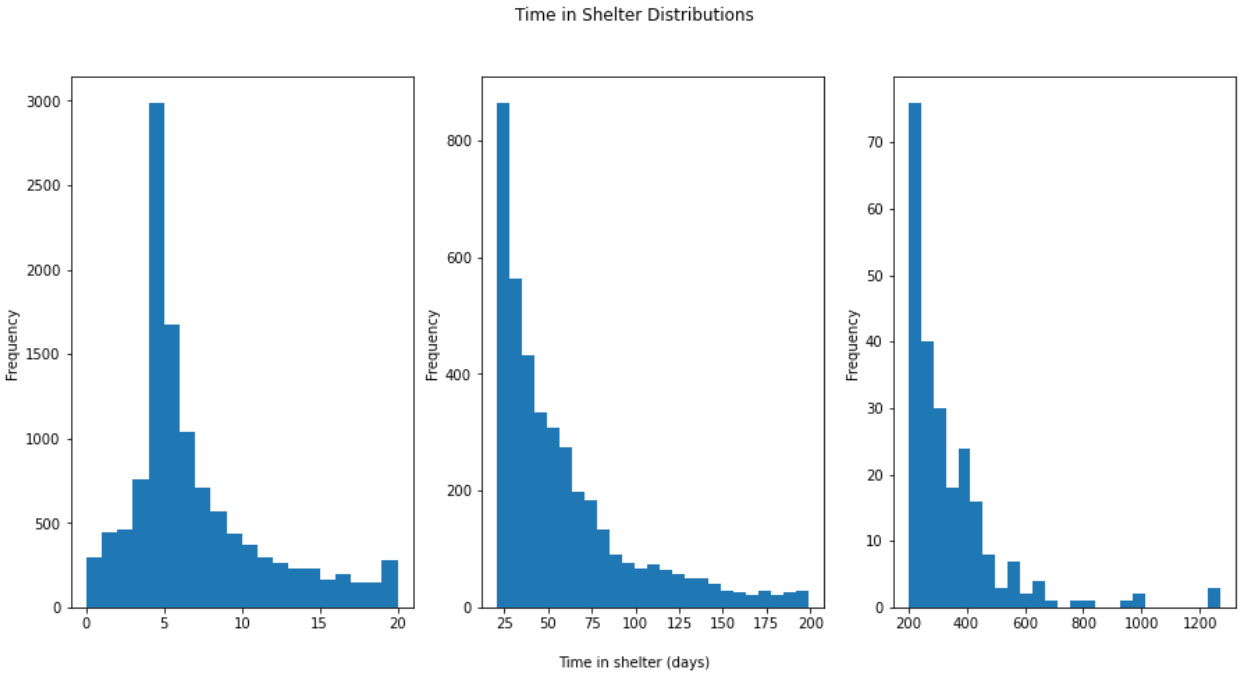

Figure 1. Histograms of the time spent in the shelter, nearly 50% of dogs spend less than one week in the shelter.

Then I split my data with a 70/30 train/test split using Sklearn's train-test-split. I chose to try K-Nearest Neighbors, Random Forest, and Logistic Regression. Using all columns (with dummy variables) KNN predicted with a 43.6% accuracy using an ideal K value found via GridSearchCV. The confusion matrix indicated that the model was favoring < 4 days over the other categories.
Random Forest performed similarly out of the box with 42% accuracy. Hyperparameter tuning did not change the accuracy score much, but did change the confusion matrix (Figure 2). Running the Random Forest with tuned hyperparameters increased the amount of dogs predicted to be at the shelter for fewer than four days. And very few dogs were predicted to be at the shelter for 4-7 days.

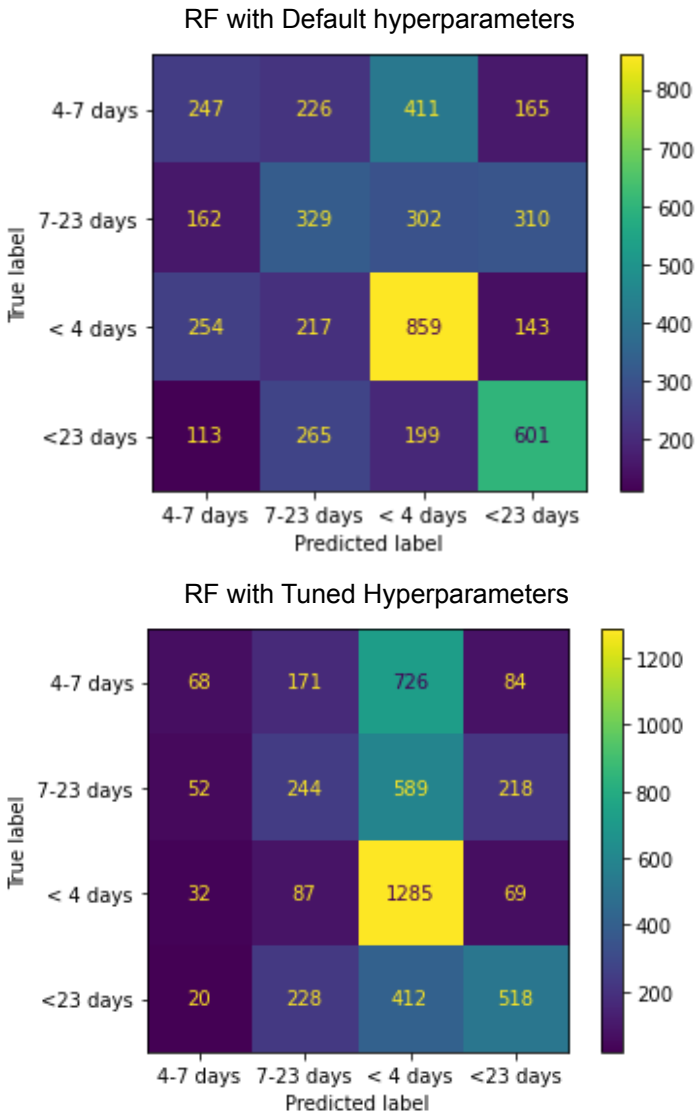

Figure 2. Confusion matrices of random forest results. The top image is the random forest run with all default hyper parameters and the bottom is with “best parameters” as determined by running GridSearchCV.

I also ran Random Forest with encoded variables instead of the dummy variables. Here, the accuracy score was similar, but the confusion matrix was more balanced. Ultimately, after some trial and error, I found removing the color column and running a Random Forest with entropy criterion had the highest accuracy score and the most balanced confusion matrix. I suspect removing the color column improved the model because color is correlated with breed and color was adding extra noise. Random forest entropy, no max depth and without color included turned out to be the best performing model and is my model selection for this project.
The last algorithm I tried was Logistic Regression. In order to run this I had to convert the target variable labels to integers using sklearn’s label encoder. This model produced similar results as the first two (43% accuracy and a similarly unbalanced confusion matrix).

Limitations and Use

This model is limited and not to be used by itself. It is to be used as a baseline estimate and in conjunction with the estimates of shelter staff. Additional characteristics like dog size and temperament would likely improve predictive power.

Future Considerations and Research

One of the best ways to improve this model is to simply add more data and implement a standardized method of gathering data. More data could include how large the dog is by size (small, medium, large) or by weight, information on the dog's temperament may also improve model performance. Standardizing the way data are gathered will help reduce noise and make sure the assumptions I made in the process of cleaning the data are correct. For example, when listing a dog's breed only the breed that the dog looks most like is listed and for dogs who are purebred that can be specified as its own column, or if a dog does not appear to be any one breed it can be listed as a 'mix'.