Feline Shelter Adoptions

Overview

Every year in the United States, approximately 3.4 million cats enter an animal shelter. This compares to 1.3 million cats that are adopted on annual basis. This means that millions of cats are left in shelters to await being adopted or to face alternative outcomes. With this project, I was hoping to discover what exactly made a feline more "adoptable" than the other cats and if shelters use this information to help find homes for more of their cats.

With only a little over 3,500 brick-and-mortar animal shelters in the United States, the tendencies for these facilities to get overcrowded or filled is very high. If cats are not adopted from a shelter, they may get transferred, go missing, or get euthanized depending on their exact circumstances. Although I do not own a cat, I have many friends and family members who own a cat and I've observed that each of them had something different that stuck out about their cat which inevitably led them to adopt. One person mentioned that they were into red-heads, so when they saw Leland and his bright red fur, they did not hesitate. Another person talked about how they thought kittens were so cute which led to them adopting Chester at the age of 3 months. Another talked about how they were just feeling depressed from the winter and wanted an animal to lift their spirits. While they all had different reasons, is there a common set of variables that would be able to predict at a high accuracy whether a sheltered cat would be adopted?

To try and solve for this, I decided that using a classification model would make the most sense. More specifically, I chose to use a Decision Tree model and experiment with Random Forest as well.

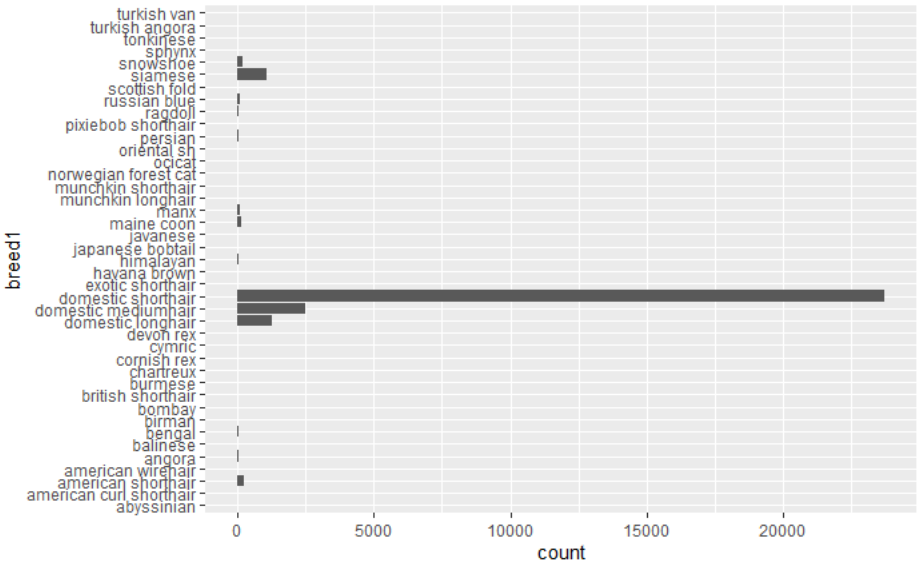
Data

The dataset I chose to use is the Animal Center Shelter Outcomes from the Austin Animal Center. This data was extracted from Kaggle here: https://www.kaggle.com/austin-animal-center-shelter-outcomes. This is a subset of the full dataset, which shows outcomes for more than just cats. The raw data also resides on Austin Center Shelter's Open Data Portal or on Github via the Socrata Open Data Access (SODA) API here https://data.austintexas.gov/resource/hcup-htgu.json.

This dataset contains over 29,000 rows and 37 variables. It includes mostly categorical variables (boolean, nominal, and ordinal). Many of these columns were repetitive in the nature by which they were manipulated after being pulled from the original and larger data source. Below is a list and brief description of the variables without restating any similar types:

Variable	Type	Distinct Values	Other Notes
Breed	Categorical	65	
Outcome Type	Categorical	8	
Outcome Subtype	Categorical	17	Associated with Outcome Type
Sex	Nominal	2	
Spay/Neutered	Nominal	2	
Cat/Kitten	Nominal	2	
DOB Year	Categorical/Ordinal	25	
DOB Month	Categorical/Ordinal	12	
Outcome Year	Categorical/Ordinal	25	
Outcome Month	Categorical/Ordinal	12	
Outcome Day	Categorical/Ordinal	7	
Outcome Hour	Categorical/Ordinal	24	
Outcome Age	Numeric	Min:0 Max: 22	Calculated based on days from DOB
CFA Breed	Boolean	2	CFA determines which cats are competition show eligible
Color	Categorical	40	
Coat Pattern	Categorical	9	

The first thing that struck me regarding these variables were the amount of values in the Breed and Color categories. When taking a closer look, I found that the Color variable had various colors that could arguable fit inside each other. My knowledge of colors exponentially increased as I started googling a few of them, including but not limited to “fawn” and “sable” (both are variation shades of brown). The Breed variable seemed like it would be a bit more of trouble because unlike color, these breeds did not look easy to consolidate. However, when plotting it, I found large volumes in domestic short, medium, and long hair along with Siamese:



When running a check on null values by variable, I found a few instances where a more detailed version of a column showed a large percentage of nulls compared to all values. In many of these instances I had

to decide whether to drop the column or find a grouping strategy to keep them in the dataset. Examples include color2, breed2, outcome_subtype, and coat_pattern. Here is a breakdown of the null counts:

```
 age_upon_outcome          animal_id        animal_type              breed              color      date_of_birth
                0                  0                  0                  0               3626                  0
         datetime          monthyear               name    outcome_subtype       outcome_type  sex_upon_outcome
                0                  0              12774              10780                  3               2528
            count                sex        spay_neuter            periods       period_range  outcome_age_days
                0                  0                  0                  0                  0                  0
outcome_age_years  cat_kitten_outcome    sex_age_outcome          age_group           dob_year         dob_month
                0                  0                  0                  0                  0                  0
     dob_monthyear      outcome_month       outcome_year    outcome_weekday       outcome_hour             breed1
                0                  0                  0                  0                  0                  0
            breed2          cfa_breed     domestic_breed       coat_pattern             color1             color2
            29369                  0                  0              10266                  0              19067
             coat
                0
```

Lastly, since most of my data was categorical, my outlier test did not yield many actionable insights. The one thing I did find though was the distribution of age in years for cats was heavily right skewed:

This data is heavily skewed towards younger cats. Based on the above, I might consider using a logarithmic function to smooth the data or perhaps try binning.

Data Preparation

From my initial exploration of the data, it became clear that many columns were either concatenated or contained similar data. My first step was to remove these columns. I dropped animal id, name, and animal_type since those columns would be of no help to the model. Outcome_subtype was removed due to the number of nulls and since the target variable of “Adopted” is included in the less granular

column called outcome_type. Age_upon_outcome, color, date_of_birth, datetime, monthyear, and many others were dropped due to them having the similar data to other columns.

My next step was to dig further into each columns' nulls. Looking at my target variable, outcome_type, I found 3 rows that had nulls. With the data having >29,000 rows, I decided to drop those three rows after reviewing them to ensure there were no reasons to keep them in. Once I removed the nulls, I decided to make a new column using this data and called it "Adopted" which would only contain two values: True and False. My target variable column was now ready to go.

With many nulls in coat_pattern, I considered dropping it, but decided it was an important variable I wanted to consider in my model. I chose to also switch this to a Boolean to indicate whether a cat had a coat pattern or not. With the breed column, I decided to keep the name for only the top four and bucket the rest into "Other". This decision was made after noticing that these 4 made up over 95% of the data.

As discussed in the data section, age in years was heavily right skewed with most of the cats being aged 0-2 years. I considered using a logarithmic function to smoothen, however I did not want to lose interpretability. After researching, I concluded that most older cats either get transferred out or euthanized as they become a disease risk to the younger cats. I decided to not transform the data and just bucket them into three categories: Young (0-2 years), Adult (2-10), Senior (>10).

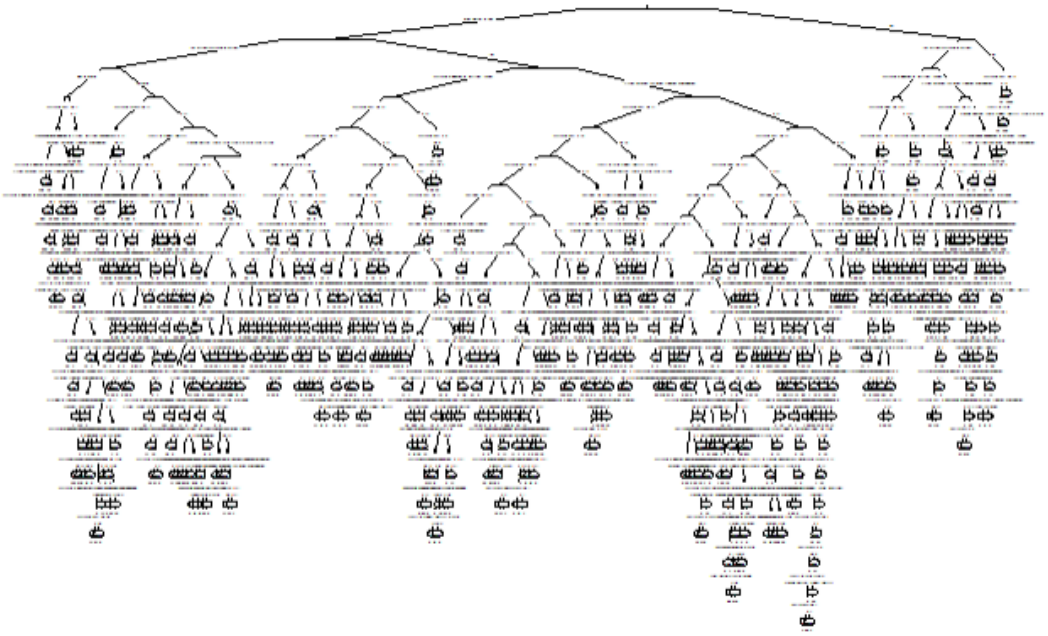
Lastly, I bucketed the color variable into what I thought was the 6 most distinguished categories. This process took me a while as I had to think through which colors fit where and even what shades made up a majority of a color. For example, is the color "flame" more of a yellow, orange, or red? Do they need a separate bucket? After finalizing this, I dropped the remaining columns that I either created a new variable for or did not need anymore.

Modeling

I decided to use a decision tree to build this classification model. My first step was to split the data into training and testing subsets. I chose a 70/30 training/test split. My first run through, I did not tweak any parameter except setting the cost complexity parameter to 0. Here are the results of the confusion matrix:

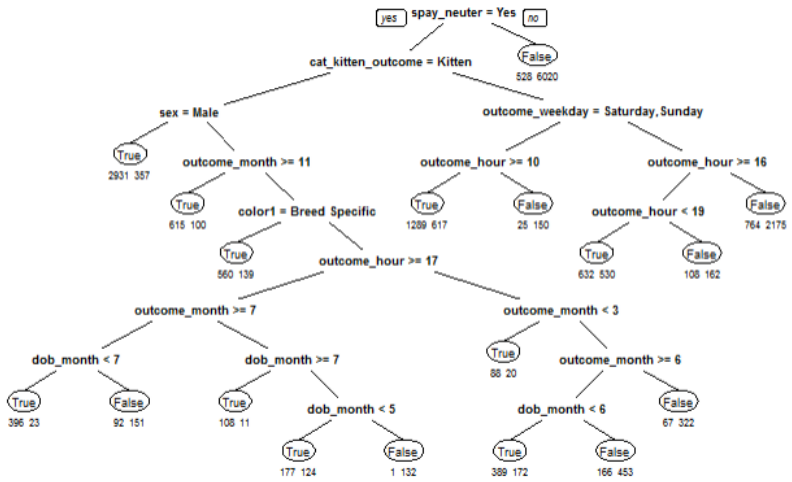
```
          Reference
Prediction True False
     True  3052   725
     False  777  4270

               Accuracy : 0.8298
                 95% CI : (0.8218, 0.8376)
    No Information Rate : 0.5661
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.653

 Mcnemar's Test P-Value : 0.1882

            Sensitivity : 0.7971
            Specificity : 0.8549
         Pos Pred Value : 0.8080
         Neg Pred Value : 0.8460
             Prevalence : 0.4339
         Detection Rate : 0.3459
   Detection Prevalence : 0.4280
      Balanced Accuracy : 0.8260

       'Positive' Class : True
```

There were 11 total variables used in this tree to include age, cat_kitten_outcome, cfa_breed, coat_pattern_bin, color and many others. This signaled to me that the model was a bit too complex and possibly overfitted. The plot of the tree itself confirmed my hypothesis:



From the plot of the complexity of the tree relative to the error rate, I found a bend at right around .005. I pruned my tree using this value for complexity penalty and here is the updated tree and confusion matrix:

```
Confusion Matrix and Statistics

          Reference
Prediction True False
     True  3040   885
     False  789  4110

               Accuracy : 0.8103
                 95% CI : (0.802, 0.8184)
    No Information Rate : 0.5661
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.615

 Mcnemar's Test P-Value : 0.02024

            Sensitivity : 0.7939
            Specificity : 0.8228
         Pos Pred Value : 0.7745
         Neg Pred Value : 0.8389
             Prevalence : 0.4339
         Detection Rate : 0.3445
   Detection Prevalence : 0.4448
      Balanced Accuracy : 0.8084

       'Positive' Class : True
```



This pruning resulted in less complexity and only a short drop in accuracy and sensitivity. I experimented with .006 and .01 as other complexity penalties, but the model worsened. I also noticed that many of the variables used here were time based, such as outcome_month, outcome_hour, dob_month, etc. I went back and attempted to bucket these into categoricals manually (outcome month switching to seasons, for example) but found that this dropped my accuracy a few points. Therefore, I decided to keep the time complexity.

I decided to run a Random Forest model on this same dataset to compare it to my pruned decision tree. It ended up producing even better results. Below is the confusion matrix and variable importances:

```
               Accuracy : 0.8647
                 95% CI : (0.8574, 0.8718)
    No Information Rate : 0.5661
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7231

 Mcnemar's Test P-Value : 1.004e-06

            Sensitivity : 0.8219
            Specificity : 0.8975
         Pos Pred Value : 0.8601
         Neg Pred Value : 0.8680
             Prevalence : 0.4339
         Detection Rate : 0.3566
   Detection Prevalence : 0.4147
      Balanced Accuracy : 0.8597
```

	Overall <dbl>
spay_neuterYes	100.000000
outcome_hour	58.740738
outcome_month	53.285465
dob_month	53.064647
dob_year	24.388575
cat_kitten_outcomeKitten	21.060219
outcome_year	16.943882
sexMale	16.176122
outcome_weekdaySaturday	9.292654

Evaluation

The final accuracy from the decision tree was just over 80%. Due to my initial problem of classifying which cats were more likely to be adopted, accuracy was my most important performance metrics, but it was good to see the specificity and sensitivity right around 80% too. I'm also content on the complexity of the model. Initially it was too complex but pruning helped drill down the variables used from 11 to 8. Prior to pruning, the model was at 82, so while I lost a bit of accuracy, it was worth the exchange for a simpler model.

After running it through the Random Forest model, the accuracy spiked up to 86%. Sensitive and precision also went up to 82% and 89%, respectively. While this model took a while to process on my laptop, the dividends were great since turned out to be my best model.

Discussion and Conclusion

I fully expected spayed/neutered and cat/kitten to be the most significant variables. As we went further down the tree, I found the time factors becoming very interesting. The story I parsed out was that beside a cat being spayed/neutered or being a kitten, seasonality has a strong effect on adoption rates, with higher instances occurring in the fall.

If I were to continue this analysis, I'd want to explore other models, such as KNN to see if that would make any difference in my model. In addition, I ran out of time to experiment with more hyperparameters in the decision tree/random forest so tweaking those would be another route I would want to take. Lastly, It would be fun to explore this entire dataset rather than just feline adoptions. I'd be interested to see how these factors stack up to adoption rates for other animals such as dogs, birds, and horses.

I learned a lot from this project. The first and most important thing was how much time pre-processing the data took and how important it was. I was having a hard time moving onto the actual modeling as I kept thinking of better ways to clean my data. From a project perspective, I was looking for more ways to visualize the data preprocessing steps I took as it was becoming a large part of my story. The second thing I learned was how important it was to understand your data, so you could choose the correct modeling method and performance metrics to key in on. I found myself asking these questions during the pre-processing stage as I became more and more familiar with the data. Lastly, this project made me realize how much fun it is to use a dataset that you might not know a ton about. The knowledge gap was certainly there, but I came out of this learning a lot more about cats and shelters then I did before.