Animal Shelter Outcomes - Capstone

Michelle Turovsky

December 14, 2019

1 Introduction

Animal shelters throughout the U.S. house over 6 million animals every year [9]. Animal centers provide temporary homes for strays, lost pets, and surrendered animals. Their ultimate goal is to appropriately re-home every animal if possible. With so many cats and dogs moving through the animal shelter system, it would be advantageous to understand the intricacies of what makes a certain cat more adoptable, or what makes a certain lost dog more likely to be returned to their owner. I will attempt to use pet intake data provided by the Austin Animal Center (AAC) to predict how an animal will leave the shelter.

2 Dataset Overview

The Austin Animal Center (AAC) is one of the largest "no-kill" shelters in the United States. AAC provides a temporary home to over 18,000 animals every year. The shelter is interested in predicting the outcome of animals that pass through their shelter. Understanding the outcome of an animal ahead of time can help animal shelters better utilize their resources, or advertise for particular animals with different nuances. Having volunteered at AAC many times in the past, I hope to bring some knowledge of the animal shelter system to my data analysis!

The data provided by AAC is collected for cats and dogs when they first enter the animal center, and when they leave the animal center. There are 26,729 observations and 10 variables in the dataset provided. Variables include:

- AnimalID
- Name

- DateTime
- OutcomeType
- OutcomeSubtype
- AnimalType
- SexuponOutcome
- AgeuponOutcome
- Breed
- Color

The data was collected between October 2013 and March 2016. The final variable I will predict is **OutcomeType**. All data is kindly hosted on Kaggle and can be found at: https://www.kaggle.com/c/shelter-animal-outcomes/data.

Unfortunately it looks like the dataset is rather imbalanced. When looking at our prediction variable **OutcomeType** we can clearly see that Adoptions and Transfers outweigh Return to Owner, Euthanasia, and Death in the dataset. Animals with an outcome of Died only account for less than 1% of observations.

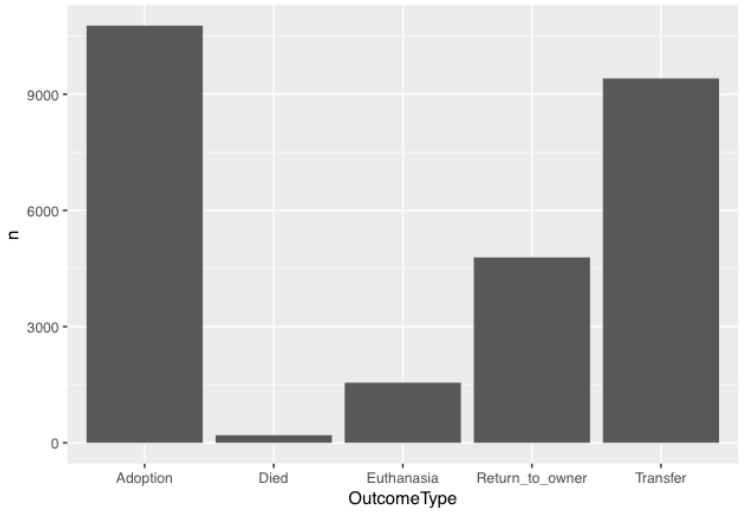

Figure 1: OutcomeType Distribution

3 Related Work

There has been a recent uptick in using analytics to predict the outcomes of animals in shelter. One example is the paper entitled "Animal Shelter Dogs: Factors Predicting Adoption Versus Euthanasia" [2] by Jamie DeLeeuw. DeLeeuw found that the factors that contributed most to an animal's outcome included the reason the animal entered the shelter, the breed of the animal, and how small the animal was. DeLeeuw used a pet database system that included information on a dog health, weight, coat, breed, and more. DeLeeuw's dataset was comprised of 7,602 dogs. While DeLeeuw had a dataset with slightly more attributes than I have for predicting animal outcomes, I would guess that some of the same variables will also affect my model as well, such as coat color, being a purebred, and being a small animal breed. A few major differences between DeLeeuw's study and my own, is that DeLeeuw only looked at dog outcomes, and only had two levels for the outcome variable: adopted vs euthanized. My own analysis will evaluate both cats and dogs, and will include five different outcome types.

4 Data Exploration and Preprocessing

4.1 Feature Removal

Upon inspecting the data, there are two variables that stand out as columns that can be removed from the dataset altogether: **AnimalID** and **OutcomeSubtype**.

AnimalID will probably not be very useful for prediction purposes, as it is just a randomized placeholder assigned to an animal within the shelter.

The variable **OutcomeSubtype** will also be removed. Upon inspecting **OutcomeSubtype**, I can see that it is essentially a more granular variable of **OutcomeType**. Each **OutcomeSubtype** maps perfectly to an **OutcomeType**. Since this variable would be a perfect predictor of **OutcomeType**, I will remove it to ensure I am not 'cheating' when creating a model.

OutcomeType	OutcomeSubtype	n
Adoption	*NA*	8803
Adoption	Barn	1
Adoption	Foster	1800
Adoption	Offsite	165
Died	*NA*	16
Died	At Vet	4
Died	Enroute	8
Died	In Foster	52
Died	In Kennel	114
Died	In Surgery	3
Euthanasia	*NA*	1
Euthanasia	Aggressive	320
Euthanasia	Behavior	86
Euthanasia	Court/Investigation	6
Euthanasia	Medical	66
Euthanasia	Rabies Risk	74
Euthanasia	Suffering	1002
Return_to_owner	*NA*	4786
Transfer	*NA*	6
Transfer	Barn	1
Transfer	Partner	7816
Transfer	SCRP	1599

Figure 2: OutcomeType by OutcomeSubType Distribution

4.2 Missing Data

Next I will clean any missing data. There are several missing instances in the **Name, SexuponOutcome**, and **AgeuponOutcome** variables.

```
> summary(is.na(outcomes))
  AnimalID          Name           DateTime       OutcomeType     OutcomeSubtype
 Mode :logical   Mode :logical   Mode :logical   Mode :logical   Mode :logical
 FALSE:26729     FALSE:19038     FALSE:26729     FALSE:26729     FALSE:13117
                 TRUE :7691                                      TRUE :13612
 AnimalType      SexuponOutcome  AgeuponOutcome    Breed           Color
 Mode :logical   Mode :logical   Mode :logical   Mode :logical   Mode :logical
 FALSE:26729     FALSE:26728     FALSE:26711     FALSE:26729     FALSE:26729
                 TRUE :1         TRUE :18
```

Figure 3: Nulls in the Dataset

The **Name** variable has quite a few missing instances. Let us assume that the specific animal name has no bearing on an animal's outcome (i.e. Buddy is no more likely to get adopted than Spot). However, I have heard that when animals have a name assigned to them, people will be more likely to be attracted to their online adoption profile. Furthermore, if an animal has a name, it is more likely that it may be a lost pet than a stray, and this will likely affect the **OutcomeType**. I will transform this **Name** variable into a boolean variable stating whether or not the animal has been assigned a name.

SexuponOutcome appears to be missing in only 1 row of data, and **AgeuponOutcome** is missing in only 18 rows of data. Since these are both fairly small numbers compared to our large dataset, I will remove these rows of data altogether, and I will likely not suffer from any information loss. The AAC has apparently done a great job in documenting most of the fields for every pet that passes through their shelter!

4.3 Feature Engineering

The **DateTime** variable is probably too specific, and I can group the days and times together. By evaluating the distribution of **OutcomeType** by a given month, I can see that there is generally a lot of activity around the end of year. This corresponds with a major holiday season in the US, and perhaps people adopt pets as gifts? I can try grouping month into quarters. Next, taking a look at the hours in which outcomes occur, I can see that there is a huge spike in adoptions in the afternoon/evening. Transfers seem to have a spike first thing in the morning. I can group hours into

a bucketed time of day variable, with buckets of Morning, Afternoon, Evening, and Night.

Biplots can be informative for visualizing data patterns for similarity. The Time of Day Biplot (below) illustrates that most adoptions happen in the evening, while euthansia is typically performed in the morning. By plotting features in this manner we can get a sense for similarity between data points across various columns.

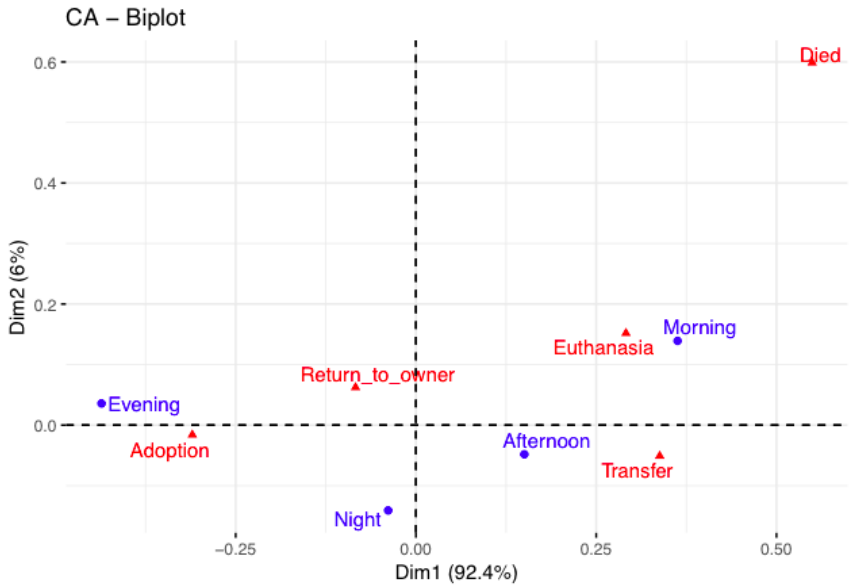

Figure 4: Time of Day Biplot

Upon inspecting **AgeuponOutcome**, I can see that it is fairly messy. It has every type of unit to describe the animal age, including weeks, months, and years. To make everything consistent, I will parse the numeric value from the character, and transform all animal ages into days. Next, based on boxplots (below) of **Age** vs **OutcomType** I will group these consistent ages into buckets of 'Month', 'TwoMonth', 'HalfYear', 'Year', 'TwoYear' and 'Adult'. These buckets will group together animals around the same ages, with more granularity for younger animals, as I know that very small animals under one month are more likely to die unexpectedly, and animals around two months are more likely to be adopted. We can see from Figure 6 below that TwoYear old animals get Adopted and Transferred most frequently, while surprisingly OneMonth old animals get transferred quite frequently.

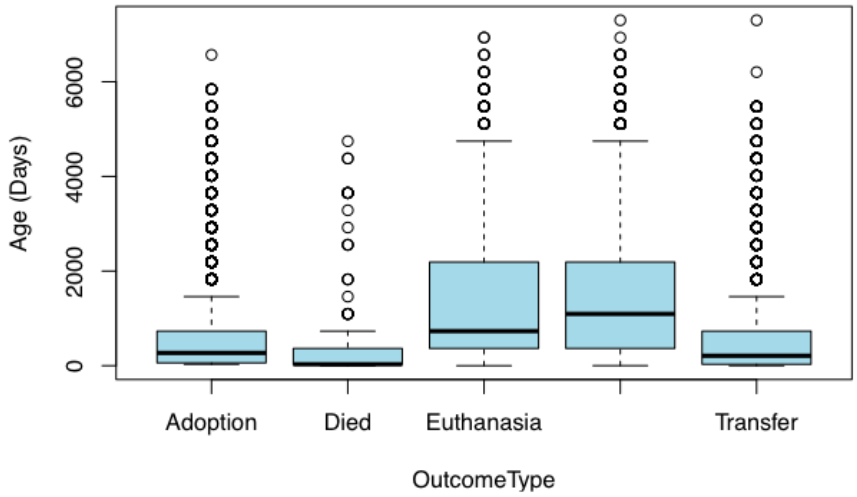

Figure 5: Age Distribution

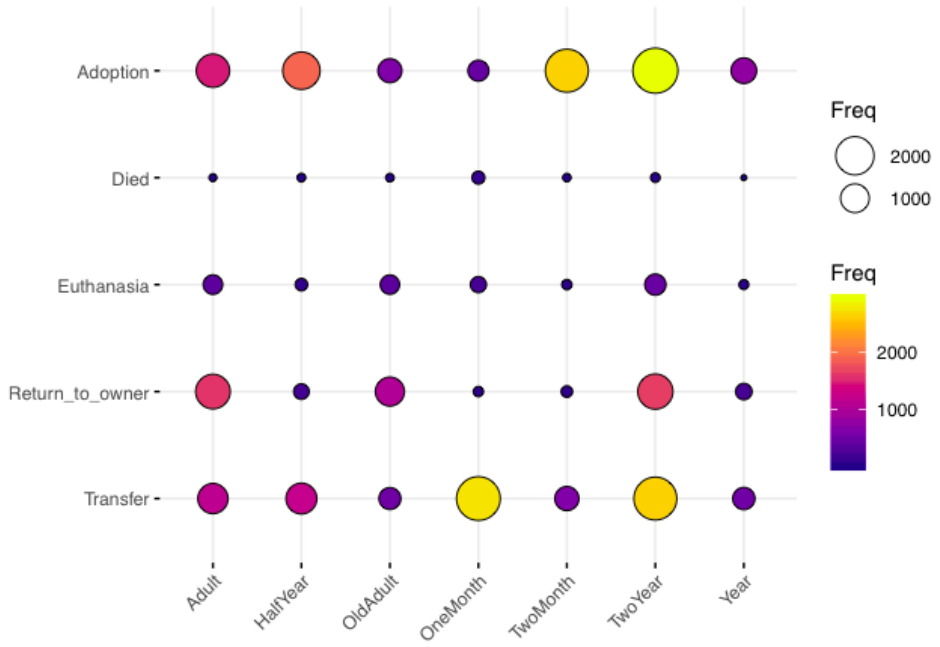

Figure 6: Age Distribution Bucketed

The **SexuponOutcome** variable is really two different variables; gender and whether or not the animal is neutered/spayed. I will split apart the gender portion and the information on neutering into two separate variables. I will consider 'spayed' to be the same thing as 'neutered', because these are gender specific words that we will no longer need, given we will have a gender variable on its own.

Looking at the **Color** variable, I can see that there are 366 unique value. For animals, there is a difference between coating and color, but it looks like a lot of the entries have both a coat type and color merged together. I can create a series of coat variables including TabbyTiger, Brindle, Merle, Tick, and Point from these coat characters. Next, I will split apart the remaining colors so that there is one color listed per column. Assuming that the animal shelter listed the animals' most prominent color first, I will discard the secondary color in an attempt to reduce some of the attribute dimensionality.

The **Breed** variable has a whopping 1,380 unique values. It also is a little misleading- some animals have their breed listed as 'Mix' and other animals simply have two distinct breeds listed but do not specify 'Mix'. I will consider both of these types of animals mixes, and create a new variable **Mix** indicating more than one breed. Next I will split apart the **Breed** variable to get a single breed per col-

umn. I know that oftentimes dog breeds are considered similar if they fall into the same breed group. The American Kennel Club [7] curates a list of dog breeds and groups. I have copied this information from their website into a CSV file and added a few more dog breeds into the list based on quick google searches. By doing all of this I can bucket the 1,380 number of distinct dog breeds into under 10 groups.

2	3	4
Group	Breed	
Herding	Australian Cattle	
Herding	Queensland Heeler	
Herding	Australian Cattle Dog	
Herding	Australian Shepherd	
Herding	Bearded Collie	
Herding	Beauceron	
Herding	Belgian Malinois	
Herding	Belgian Sheep	

Figure 7: Dog Breed to Group Mapping

While all of the features and buckets that I created are not perfect and may cause some specificity loss, I need to reduce the number of variables I will end up with, especially after I create dummy variables for attributes that have many unique values. This may even help prevent some overfitting.

4.4 Evaluate Feature Significance

Throughout feature engineering and the exploratory process I performed multiple chisq.test's on the features related to **OutcomeType**. All engineered attributes have an extremely low p-value for Pearson's Chi-squared test, and therefore are likely going to be good predictors of our **OutcomeType** variable. Additionally I created a few biplots of the features I engineered to get a sense for which variables and features will likely help contribute in my **OutcomeType** predictions.

5 Data Analysis and Results

5.1 Models

Since all of our attributes are now categorical variables, I will create dummy variables for them using caret's dummyVars function. I will specify that I want a full rank

matrix here, meaning that I will remove any collinearity in the data by having one less attribute than the number of unique categories per attribute.

I will evaluate several models; decision tree (RPART), random forest (RF), K Nearest Neighbors (KNN), Naive Bayes (NB), gradient boosting machine (GBM), and a Multilayer Perceptron Artificial Neural Network (ANN).

5.2 Method of Evaluation

Since the animal shelter dataset is rather imbalanced, I will not only look at accuracy but also evaluate the sensitivity and specificity of the models as the evaluation measure for this classification problem. In classification cases where a dataset is imbalanced it is important to evaluate more than just accuracy. If an imbalanced dataset is 80% class A and 20% class B, a model predicting class A for every instance would have 80% accuracy. Clearly this type of model does not provide any valuable predictive power. Therefore by evaluating sensitivity and specificity we can with greater confidence identify a model with good predictive power. Sensitivity indicates the portion of positive instances that are identified as positive, while specificity indicates the portion of negative instances of a class that are identified as negative.

The "no information rate" is the accuracy a model would have by just guessing the most common class in the dataset for every instance. For this project the no information rate would have been 40.38%. We will aim to beat the no information rate with all of our models.

5.3 Dataset Imbalance

While iterating through this project I attempted to implement a few methods to combat the imbalance in my dataset. I first tried upsampling my dataset to take random samples of my minority classes (mainly the **OutcomeType**s Died and Euthanized) and add these instances back into the original dataset. I also tried a more advanced method of creating synthetic data called Synthetic Minority Oversampling Technique (SMOTE). SMOTE creates synthetic data based on a nearest neighbor algorithm. SMOTE therefore can add additional instances of minority classes to a dataset to balance out the classes.

After separately implementing both of these methods to fix class imbalance and running the new balanced datasets separately through my models below, my accuracy scores actually decreased significantly. I suspect this may be have happened because upsampling or creating synthetic data based on so few instances introduced more noise into my models. This leads me to believe that the current set of features in my

data is not highly predictive of **OutcomeType**. Other features that characterize an animals health or behavior may be more indicative of what their **OutcomeType** would be. Perhaps in the future Kaggle and the AAC can talk about collecting additional metrics on animals in the shelter to augment the current dataset. For now, given the lowered accuracy rates using imbalanced dataset correction techniques, I will proceed with the original imbalanced dataset.

5.4 Train-Test Split

I will split the data into 80% training data and 20% test data. When I run the algorithms I will use 5-fold cross validation. Given better computing power, I could increase this to 10 or higher, but for now 5 will allow me to run these algorithms in a reasonable amount of time.

5.5 Hyperparameters

I will allow the caret package in R to autotune most of the hyperparameters for my models. In a few instances, after running the models several times I selected my own gridsearch values for certain hyperparemeters in various models. In general this yielded me slightly better results.

5.6 Results

- RPART: I used RPART as a decision tree baseline model. We can see from the confusion matrix that RPART had a hard time with the sparse classes and did not predict any instances of Died or Euthanasia. So while accuracy is not bad, the classes we are most interested in looking at never get predicted. RPART uses gini impurity to split along different variables, and by doing so it selected the Neutered variable as well as several Age related variables as those having the most "pure" splits for the purposes of our classification problem.

```
## CART
##
## 21424 samples
##    88 predictor
##     5 classes: 'Adoption', 'Died', 'Euthanasia', 'Return_to_owner', 'Transfer'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 17139, 17138, 17140, 17140, 17139
## Resampling results across tuning parameters:
##
##   cp           Accuracy   Kappa
##   0.002322062  0.6276141  0.4058795
##   0.006778334  0.6130983  0.3735898
##   0.339699436  0.5228194  0.2100117
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.002322062.
```

Figure 8: RPART Accuracy

```
## Confusion Matrix and Statistics
##
##                  Reference
## Prediction        Adoption Died Euthanasia Return_to_owner Transfer
##   Adoption            1868    5         72             625      469
##   Died                   0    0          0               0        0
##   Euthanasia             0    0          0               0        0
##   Return_to_owner       58    1         14             111       45
##   Transfer             236   33        225             223     1369
##
## Overall Statistics
##
##                Accuracy : 0.6253
##                  95% CI : (0.6122, 0.6383)
##     No Information Rate : 0.4038
##     P-Value [Acc > NIR] : < 0.00000000000000022
##
##                   Kappa : 0.4016
##
##  Mcnemar's Test P-Value : NA
##
```

Figure 9: RPART Confusion Matrix

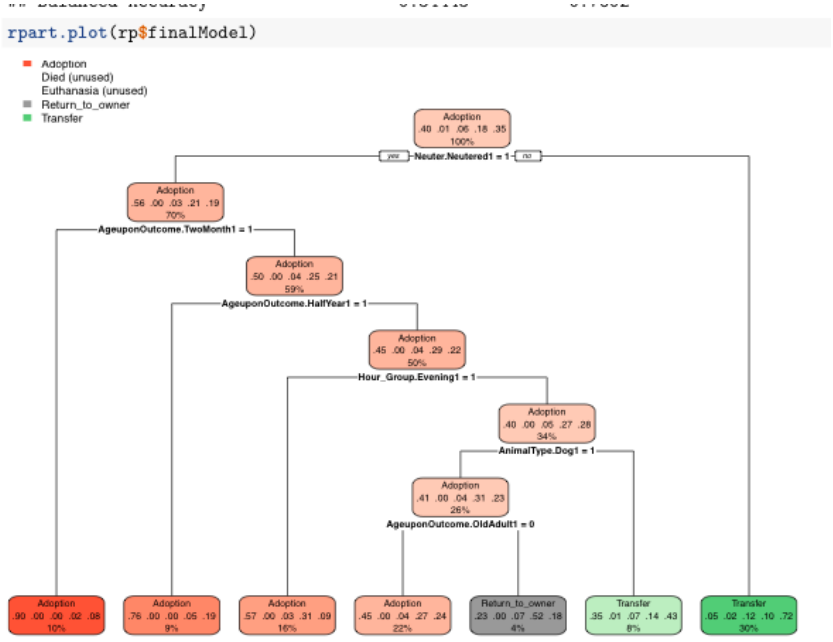

Figure 10: RPART Decision Tree

- Random Forest: Random Forests are slightly more complicated model but in general perform very well on both classification and regression problems. They allow evaluation of feature importance which is really interesting for this project. Unsurprisingly, whether an animal is neutered or not is a very important variable to the random forest model just like it was in RPART. This could be because animal shelters do not allow dogs and cats to be adopted without being spayed or neutered, and perhaps these animals only have other types of outcomes. Other important variables include whether or not the animal has been given a name, which we did not see in RPART. It looks like the animal breed and color are less important in judging an outcome. The sensitivity and specificity of the random forest is a mixed bag. For the **OutcomeType** of Adoption and Transfer, the sensitivity and specificity are fairly high. This means the probability of the model predicting the **OutcomeType** being Adoption or Transfer is high among animals whose true **OutcomeType** is Adoption or Transfer, and the probability of the model predicting the **OutcomeType** not being Adoption or Transfer is high among animals whose true **OutcomeType** is not Adoption or Transfer. However, the sensitivities for **OutcomeType** Died and Euthanasia are near zero. This means the model is

not correctly identifying very many animals with these **OutcomeTypes**, but is still doing a better job than RPART.

mtry	Accuracy	Kappa
5	0.6450715	0.4423385
15	0.6510923	0.4682272
25	0.6371365	0.4504902

Figure 11: RF Model Accuracies

```
## Statistics by Class:
##
##                      Class: Adoption Class: Died Class: Euthanasia
## Sensitivity                   0.8016   0.0000000          0.147910
## Specificity                   0.7212   0.9994356          0.992465
## Pos Pred Value                0.6607   0.0000000          0.547619
## Neg Pred Value                0.8429   0.9927116          0.949715
## Prevalence                    0.4038   0.0072843          0.058087
## Detection Rate                0.3237   0.0000000          0.008592
## Detection Prevalence          0.4899   0.0005603          0.015689
## Balanced Accuracy             0.7614   0.4997178          0.570187
##                      Class: Return_to_owner Class: Transfer
## Sensitivity                         0.42127          0.6845
## Specificity                         0.89192          0.8629
## Pos Pred Value                      0.45961          0.7303
## Neg Pred Value                      0.87598          0.8345
## Prevalence                          0.17912          0.3517
## Detection Rate                      0.07546          0.2408
## Detection Prevalence                0.16418          0.3297
## Balanced Accuracy                   0.65660          0.7737
```

Figure 12: RF Sensitivity/Specificy

```
## rf variable importance
##
##   only 20 most important variables shown (out of 88)
##
##                                  Overall
## Neuter.Neutered1                 100.000
## HasName.11                        41.052
## AgeuponOutcome.TwoMonth1          28.122
## Hour_Group.Evening1               24.710
## AgeuponOutcome.OneMonth1          20.837
## AnimalType.Dog1                   16.733
## AgeuponOutcome.OldAdult1          14.807
## AgeuponOutcome.HalfYear1          13.555
## Gender.Male1                      12.758
## Month_Group.Q41                   12.408
## AgeuponOutcome.TwoYear1           11.348
## Month_Group.Q21                   10.892
## Final_Group.Domestic.Shorthair1   10.808
## Hour_Group.Morning1               10.375
## Month_Group.Q31                   10.330
## Color.Black1                       9.927
## Color.Brown1                       8.777
## Color.White1                       8.594
## Final_Group.Non.Sporting1          6.906
## Mix.11                             6.759
```

Figure 13: RF Variable Importance

- KNN: This algorithm performed best with **k**=15. This means that the model used 15 nearest neighbor data observations when classifying **OutcomeType**. KNN performed worse than the random forest in terms of sensitivity and specificity in almost every **OutcomeType** value. For this reason, I would not consider KNN a strong contender for a final model of animal shelter outcomes classification.

```
## k-Nearest Neighbors
##
## 21424 samples
##    88 predictor
##     5 classes: 'Adoption', 'Died', 'Euthanasia', 'Return_to_owner', 'Transfer'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 17139, 17138, 17140, 17140, 17139
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    2  0.5879861  0.3787579
##   10  0.6230869  0.4210119
##   15  0.6256071  0.4227978
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 15.
```

Figure 14: KNN Model

```
## Statistics by Class:
##
##                      Class: Adoption Class: Died Class: Euthanasia
## Sensitivity                   0.8302    0.000000          0.0160772
## Specificity                   0.6507    1.000000          0.9992068
## Pos Pred Value                0.6168         NaN          0.5555556
## Neg Pred Value                0.8498    0.992716          0.9427502
## Prevalence                    0.4038    0.007284          0.0580874
## Detection Rate                0.3353    0.000000          0.0009339
```



```
## Detection Prevalence          0.5435    0.000000          0.0016810
## Balanced Accuracy             0.7405    0.500000          0.5076420
##                      Class: Return_to_owner Class: Transfer
## Sensitivity                         0.37331          0.6352
## Specificity                         0.89829          0.8750
## Pos Pred Value                      0.44472          0.7337
## Neg Pred Value                      0.86788          0.8155
## Prevalence                          0.17912          0.3517
## Detection Rate                      0.06687          0.2234
## Detection Prevalence                0.15035          0.3044
## Balanced Accuracy                   0.63580          0.7551
```

Figure 15: KNN Sensitivity/Specificity

- Naive Bayes: This model performed the poorest out of all models in terms of accuracy, and again it struggled with **OutcomeType**s of Died and Euthanized. I was surprised to find Naive Bayes performing so poorly, given it generally performs well even on limited data. However perhaps this dataset is violating an important assumption of my predictors all being independent from one another. Due to the poor generalizability of Naive Bayes I will also not consider it for selection as our final model.

```
## Naive Bayes
##
## 21424 samples
##    88 predictor
##     5 classes: 'Adoption', 'Died', 'Euthanasia', 'Return_to_owner', 'Transfer'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 17139, 17138, 17140, 17140, 17139
## Resampling results across tuning parameters:
##
##   usekernel  Accuracy   Kappa
##   FALSE      0.5827574  0.4016996
##    TRUE      0.5827574  0.4016996
##
## Tuning parameter 'fL' was held constant at a value of 0
## Tuning
##  parameter 'adjust' was held constant at a value of 1
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were fL = 0, usekernel = FALSE
##  and adjust = 1.
```

Figure 16: Naive Bayes Model

```
##
## Statistics by Class:
##
##                      Class: Adoption Class: Died Class: Euthanasia
## Sensitivity                   0.6008   0.0000000          0.041801
## Specificity                   0.8271   0.9996237          0.995638
## Pos Pred Value                0.7018   0.0000000          0.371429
## Neg Pred Value                0.7536   0.9927130          0.943974
## Prevalence                    0.4038   0.0072843          0.058087
## Detection Rate                0.2426   0.0000000          0.002428
## Detection Prevalence          0.3457   0.0003736          0.006537
## Balanced Accuracy             0.7140   0.4998119          0.518719
##                      Class: Return_to_owner Class: Transfer
## Sensitivity                          0.7518          0.5916
## Specificity                          0.7265          0.8764
## Pos Pred Value                       0.3749          0.7220
## Neg Pred Value                       0.9306          0.7982
## Prevalence                           0.1791          0.3517
## Detection Rate                       0.1347          0.2081
## Detection Prevalence                 0.3592          0.2882
## Balanced Accuracy                    0.7392          0.7340
```

Figure 17: Naive Bayes Sensitivity/Specificity

- GBM: This model performed the best with **interaction.depth**=3, and **n.trees**=150. It looks like the feature importance is extremely similar to that of the random forest. The animal being neutered, having a name, being young, and having an outcome occur in the evening were all very important features to the model. The accuracy of GBM is just slightly higher than the random forest accuracy. Inspecting the sensitivity and specificity of GBM and comparing it to that of the random forest, I'm not convinced that the values indicate GBM is necessarily the superior model. However, when looking at the 95% accuracy confidence interval for GBM, it does look like this interval is higher than the confidence interval for the random forest accuracy.

```
## Confusion Matrix and Statistics
##
##                 Reference
## Prediction       Adoption Died Euthanasia Return_to_owner Transfer
##   Adoption           1798    7         41            448      441
##   Died                  0    0          0              0        1
##   Euthanasia            1    1         36              1       18
##   Return_to_owner     195    1         67            378      127
##   Transfer            168   30        167            132     1296
##
## Overall Statistics
##
##                Accuracy : 0.6552
##                  95% CI : (0.6423, 0.6679)
##     No Information Rate : 0.4038
##     P-Value [Acc > NIR] : < 0.00000000000000022
##
##                   Kappa : 0.4693
##
##  Mcnemar's Test P-Value : < 0.00000000000000022
##
```

Figure 18: GBM Model

```
""
## Statistics by Class:
##
##                      Class: Adoption Class: Died Class: Euthanasia
## Sensitivity                   0.8316   0.0000000          0.115756
## Specificity                   0.7065   0.9998119          0.995836
## Pos Pred Value                0.6574   0.0000000          0.631579
## Neg Pred Value                0.8610   0.9927144          0.948084
## Prevalence                    0.4038   0.0072843          0.058087
## Detection Rate                0.3358   0.0000000          0.006724
## Detection Prevalence          0.5108   0.0001868          0.010646
## Balanced Accuracy             0.7690   0.4999059          0.555796
##                      Class: Return_to_owner Class: Transfer
## Sensitivity                          0.3942          0.6883
## Specificity                          0.9113          0.8568
## Pos Pred Value                       0.4922          0.7228
## Neg Pred Value                       0.8733          0.8352
## Prevalence                           0.1791          0.3517
## Detection Rate                       0.0706          0.2421
## Detection Prevalence                 0.1434          0.3349
## Balanced Accuracy                    0.6527          0.7725
```

Figure 19: GBM Sensitivity/Specificity

- Artificial Neural Net: A Multilayer Perceptron will allow us to predict a multi-class classification problem. To use keras and tensorflow in R took some setup and merging with Python using the "reticulate" package. I will set up a simple neural net model with the first layer having 5 neurons, and the second softmax layer having 5 possible probability scores for the 5 **OutcomeType** options. This model is fairly simple with only 10 epochs, and is trained using an ADAM optimizer. By the 5th or 6th epoch, accuracy appears to asymptote. This ANN did fairly well in terms of accuracy with nearly the highest accuracy score. We can also see from the confusion matrix that a lot of classes were predicted correctly, however no instances of Death were predicted.

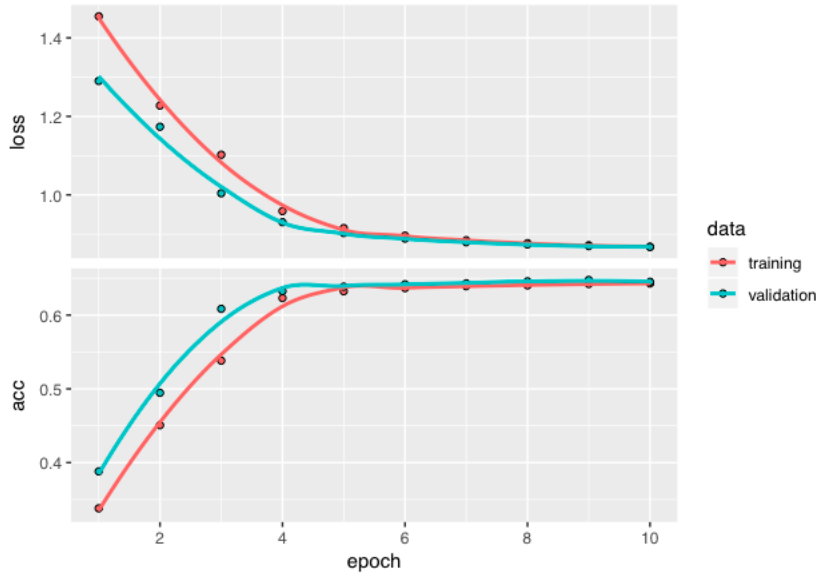

Figure 20: ANN Loss

```
##
##                  Adoption Died Euthanasia Return_to_owner Transfer
##   Adoption            1784    8         33            433      424
##   Euthanasia             1    1         35              7       21
##   Return_to_owner      236    1         78            397      160
##   Transfer             141   29        165            122     1278
```

```
score <- model %>% evaluate(as.matrix(outcomes_test[,-1]), as.matrix(test
print(score)
```

```
## $loss
## [1] 0.8578746
##
## $acc
## [1] 0.6525962
```

Figure 21: ANN Accuracy

6 Conclusion

Overall it would appear that GBM was the best model for our task. It had high scores for accuracy and decent sensitivity/specificity compared to some of the other models. The ANN also performed very well, but I think one of the strengths of using GMB is the ability to see feature importance. While accuracy was not extremely high even in the best model, I was still able to achieve some gain over the no information rate. In the future, better feature engineering (or maybe even less feature engineering), and using more data could improve upon the accuracy rate.

The variable importance output from the GBM (and random forest) was very interesting to inspect. It appears that some of the features I engineered, such as **HasName** and **Neuter** are very important to the model. Times of day and times of year are also very important surprisingly.

Processing this dataset was an excellent reminder that real world data is often extremely messy and difficult to wrangle. Tough decisions need to be made regarding missing values, categorical attributes with too many unique values, and general feature engineering. There is a huge trade off between having large datasets with high explanatory power, and algorithm computation time.

There are several items to consider as future improvements to this project. First would be enhancing the volume of data collected to overcome issues with imbalanced data. The **OutcomeType** class has relatively few instances of animal death or euthanasia. While this is great news in the real world, for the purposes of this analysis it makes predicting rare animal outcomes harder. One sure-fire way to combat this data imbalance issue would be to obtain the most updated version of the Austin Animal Center data, and perhaps obtain data on even more characteristics about each animal.

Another possibility to augment this analysis includes implementing entity embedding for categorical variables with a high number of unique entries. Entity embedding essentially feeds one-hot encoded data into a neural network. The weights from the hidden layer of the neural network would be used as a representation of the categorical variable but weights would be of low-dimensionality. This could potentially be a more robust method to process the **Breed** and **Color** variables. I merely grouped these variables into higher-order categories, but perhaps entity embedding would help the predictions.

7 References:

1. Ali, Saqib. "Recursive Partitioning / Decision Trees Using CARET." RPubs, rpubs.com/saqib/rpart.

2. DeLeeuw , Jamie L. ANIMAL SHELTER DOGS: FACTORS PREDICTING ADOPTION VERSUS EUTHANASIA. Wichita State University, Dec. 2010, soar.wichita.edu/bitstream/handle/10057/3647/d10022_DeLeeuw.pdf.

3. Fabio. "CA - Correspondence Analysis in R: Essentials." STHDA, 24 Sept. 2017, www.sthda.com/english/articles/31-principal-component-methods-in-r-practical-guide/113-ca-correspondence-analysis-in-r-essentials/#visualization-and-interpretation.

4. Facts + Statistics: Pet Statistics. www.iii.org/fact-statistic/facts-statistics-pet-statistics.

5. "Keras: Deep Learning in R." DataCamp Community, www.datacamp.com/community/tutorials/r-deep-learning.

6. Kuhn, Max. The Caret Package. 27 Mar. 2019, topepo.github.io/caret/pre-processing.html#creating-dummy-variables.

7. "List of Breeds by Group – American Kennel Club." American Kennel Club, www.akc.org/public-education/resources/general-tips-information/dog-breeds-sorted-groups/.

8. Maulik, Patel. "Naive Bayes." RPubs, rpubs.com/maulikpatel/224581.

9. "Pet Statistics." ASPCA, www.aspca.org/animal-homelessness/shelter-intake-and-surrender/pet-statistics.

10. "Tutorial: Basic Classification." Keras-RStudio, keras.rstudio.com/articles/tutorial-basic-classification.html.

11. "Understanding Medical Tests: Sensitivity, Specificity, and Positive Predictive Value." HealthNewsReview.org, www.healthnewsreview.org/toolkit/tips-for-understanding-studies/understanding-medical-tests-sensitivity-specificity-and-positive-predictive-value/.
