Daniel Andrade
Matthew Bundas
Shahriar Dipon

Motivation and Proposal
Austin is a unique city because it maintains a 90% survival rate of animals which enter its shelters. Our team proposes to use public data sets from the Austin Animal Shelter to investigate the time an animal will spend in the shelter. This investigation will be based on the animal's attributes provided in the data sets, including: age, color, breed, sex, and outcome type. Investigating these attributes by using machine learning methods may be helpful for other cities or other shelters with occupancy planning, or it may provide guidance in determining which animals to transfer to sanctuaries, no kill cities/ shelters, foster homes. etc. The motivation for this project was to choose a meaningful and unique public data set that has not been analyzed excessively by platforms such as Kaggle or UCI Machine learning Repository. Another motivation for studying this data is its potential to reduce the unnecessary killing of animals in shelters.

Problem - Use machine learning algorithms and data mining techniques to in an effort to predict the time an animal will spend in a shelter based on its physical attributes.

Solution - Acquire, clean, and merge the two data sets described below and apply various machine learning algorithms. Compare and contrast the algorithms performance including accuracy and computational costs.

Data
Both data sets are provided by the City of Austin Open Data Portal which is designed to provide high value city data to users interested in finding out more about the city. For this project, we will use two data sets. Animal Center Intakes which includes data from Oct, 1st 2013 to March 20th 2021. This data set represents the status of animals as they arrive at the Animal Center. All animals receive a unique Animal ID during intake. Annually over 90% of animals entering the center, are adopted, transferred to rescue or returned to their owners. This data set contains 124,492 rows and 12 columns which are Animal ID, Name, Date Time, Month Year, Found Location, Intake Type, Intake Condition, Animal Type, Sex Upon Intake, Age Upon Intake, Breed, and Color.
The second data set is the Animal Center Outcomes from Oct, 1st 2013 to March 20th 2021. Outcomes represent the status of animals as they leave the Animal Center. The Outcomes data set reflects that Austin, TX. is the largest "No Kill" city in the country. This data set contains 124,880 rows and 12 columns which are Animal ID, Names, Date Time, Month Year, Date of Birth, Outcome Type, and outcome Sub-type, Animal Type, Sex upon Outcome, Age upon Outcome, Breed, Color.

Data Preparation
Before analyzing the data, the first step is to investigate and prepare the data. As described in the previous paragraph, the Austin Animal Center Intake consists of the following attributes: Animal ID, Name, Date Time, Month Year, Found Location, Intake Type, Intake Condition, Animal Type, Sex upon Intake, Age upon Intake, Breed, and Color. Animal ID is the unique key between both data sets.

The ‘Name’ column was mostly blank and it is an arbitrary value, so this attribute was dropped, it will not be considered for our analysis. ‘Date Time’ and ‘Month Year’ were found to contain the exact same information, so ‘Month Year’ was dropped. The ‘Found Location’ consisted of specific street addresses inside the City of Austin and almost all the entries were unique, this column was dropped. ‘Intake Type’ consists of Stray, Owner Surrender, Public Assist, Euthanasia Request, and Abandoned. All these categories will be kept and used for analysis. Figure 1 below shows a histogram of this column.

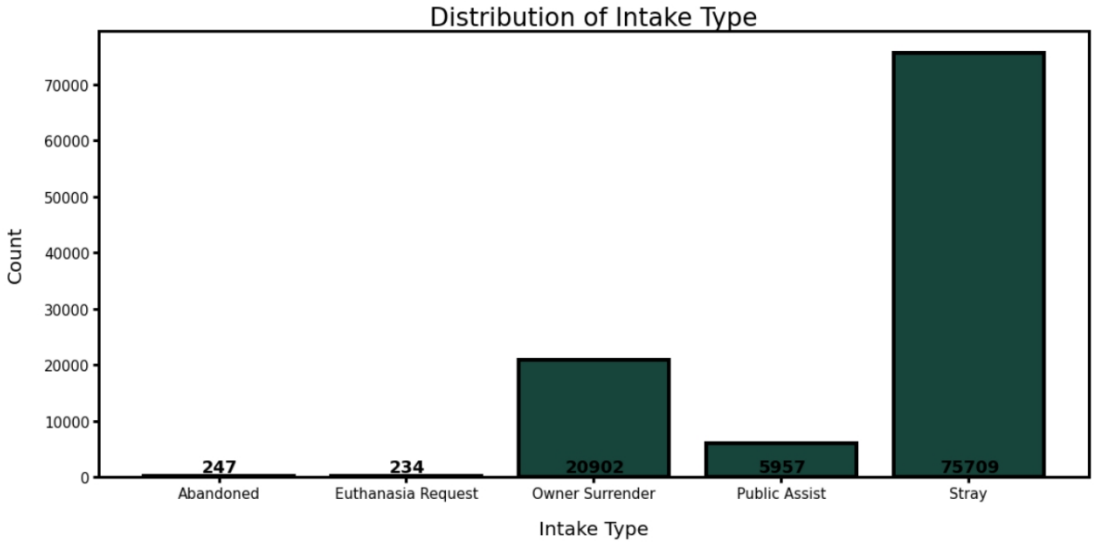

Figure 1. Histogram of Intake Type

‘Intake Condition’ consists of the following attributes: Normal, Sick, Injured, Nursing, Aged, Other, Feral, Medical, Pregnant, and Behavior. All these categories will be kept and used for analysis. Figure 2 provides a visual representation of the distribution.

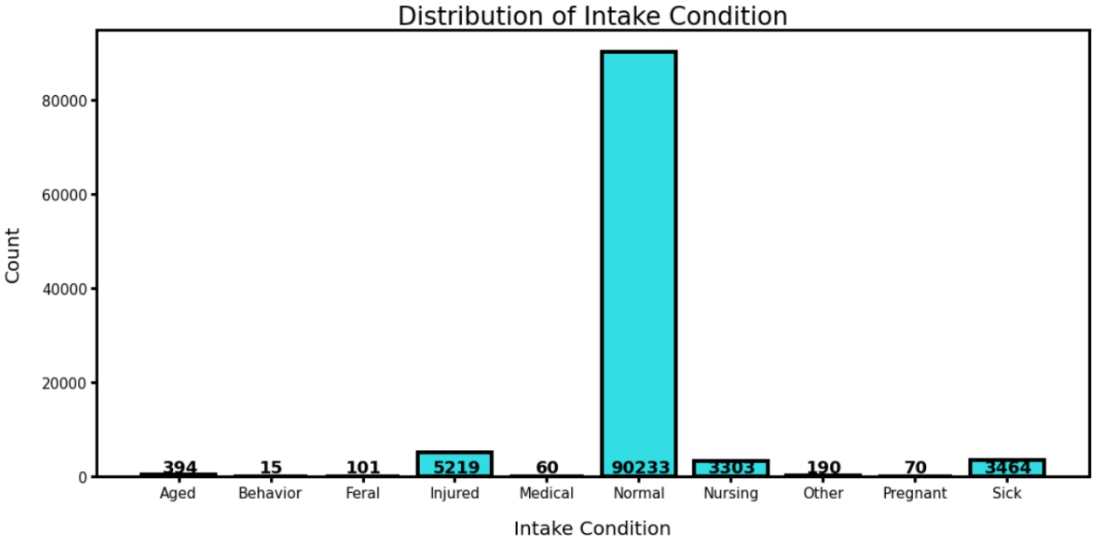

Figure 2. Histogram of Intake Condition

The ‘Animal Type’ attribute was found to contain data for ‘Dog’, ‘Cat’, ‘Other’, ‘Livestock’, and ‘Bird’, the instances of ‘Other’, ‘Livestock’, and ‘Bird’ had very few instances, less than 0.01% of the data, so they were dropped. Figure 3 is a histogram showing the quantity of each animal type.

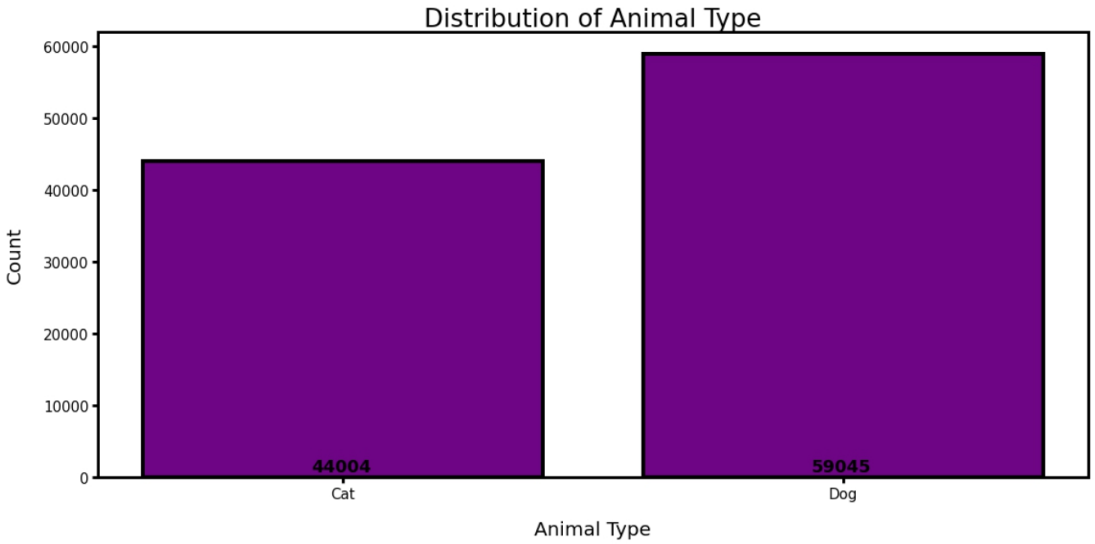

Figure 3. Histogram of Animal Type

‘Sex upon Intake’ consists of Intact Female, Intact Male, Spayed Female, Neutered Male, and Unknown. All categories will be kept for analysis. ‘Age upon Intake’ was dropped because the animal’s age was also present in the Austin Animal Center Outcomes data set. The ‘Breed’ and ‘Color’ were kept, but they contain a large number of unique values. Reducing the number of unique values may be considered later depending on the accuracy of the models.

Austin Animal Center Outcomes consists of the following attributes: Animal ID, Name, Date Time, Month Year, Date of Birth, Outcome Type, Outcome Sub-type, Animal Type, Sex upon Outcome, Age upon Outcome, Breed, and Color. Some of these attributes appear in the Austin Animal Intake data set so redundant columns were dropped which included ‘Name’, ‘Month Year’, ‘Animal Type’, ‘Breed’, and ‘Color’. The attribute ‘Date of Birth’ is not relevant for our analysis because the animal’s age is already listed, so this attribute was deleted. ‘Outcome Sub-type’ was mostly blank, so this column was omitted from our analysis. ‘Sex upon Outcome’ consists of Intact Female, Intact Male, Spayed Female, Neutered Male, and Unknown. ‘Age upon Outcome’ had 49 unique values and they were string values with a combination of years, months, or days. To keep units consistent, all values were converted to a numeric value of days. Animals that were listed with an age of “0 years” or any negative value were dropped. Finally, both data sets were merged based on the ‘Animal ID’ attribute to create a working data set for our analysis. While investigating this data set, it was discovered that there were multiple instances with the same Animal ID. Since the outcome of all the instances was “Return to Owner”, only the most recent instance was kept. Figure 4 shows the distribution of Outcome Type.

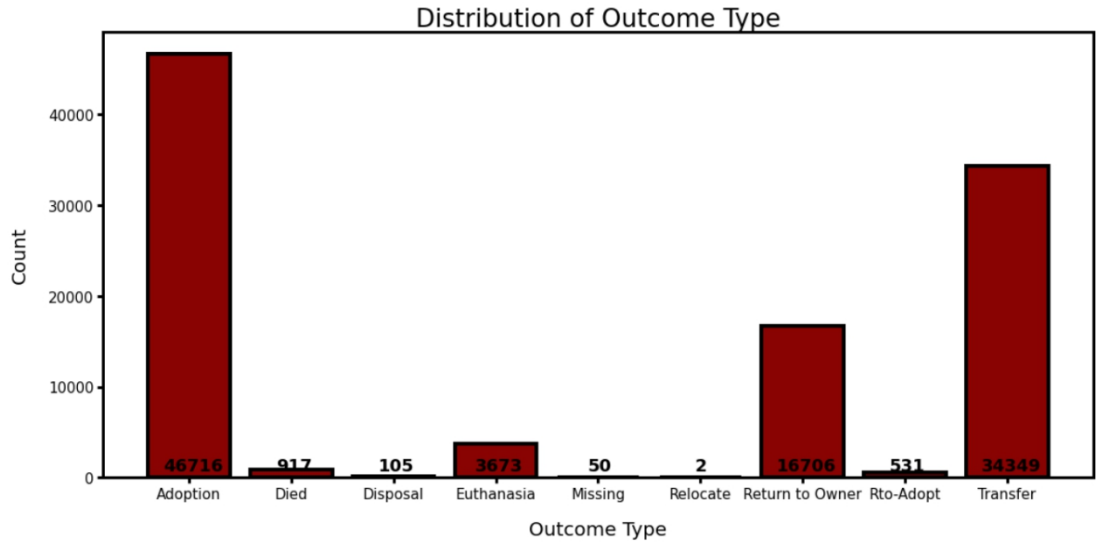

Figure 4. Histogram of Outcome Type

Analysis

As described above, the majority of the data is categorical in the form of text strings. Since most the algorithms in Sci-Kit Learn require numeric data, the data set had to first be encoded with numeric values. Fortunately, the Sci-Kit Learn package has multiple methods to encode data, and two popular approaches are Label Encoder and One-Hot-Encoding. However, each approach has benefits and drawbacks.

The first method used to encode the categorical data was Label Encoder. This method converts each value in a column to an integer. One of the drawbacks of this encoder is that the numeric values could potentially cause an algorithm to misinterpret the data as having some sort of hierarchy or order to the categories/ values. An example of how this works can be seen with the 'Intake Type' column. The figure below shows the contents of the attribute and the output when the Label Encoder Fit Transform is used. It is seen that the value of Wildlife might be weighted higher than any other value in the column. The advantage of this method is that it does not create additional columns like One Hot Encoder. Due to this fact, this method is less computationally expensive than One Hot Encoder.

Original Value	Label Encoder Assigned Value
Abandoned	0
Euthanasia Request	1
Owner Surrender	2
Public Assist	3
Stray	4
Wildlife	5

Figure 5. Label Encoder Values

To avoid the issue of weight or hierarchy, another common approach is One-Hot Encoding. Using this strategy, each value of an attribute is converted into a new

column and assigned with a binary value 1 or 0 (notation for True or False) to the column. Using the same attribute above and using the One Hot Encoding fit and transform function, the result would be as shown in Figure 6. Though this approach eliminates the hierarchy/order issues, it does have the downside of adding more columns to the data set. In our data set the attributes for 'Animal Breed' and 'Animal Color' contain 2391 and 555 unique values respectively. This method of encoding will increase the size of our data set enormously and will be very computationally expensive.

Original Value	OHE1	OHE2	OHE3	OHE4	OHE5	OHE6
Abandoned	1	0	0	0	0	0
Euthanasia Request	0	1	0	0	0	0
Owner Surrender	0	0	1	0	0	0
Public Assist	0	0	0	1	0	0
Stray	0	0	0	0	1	0
Wildlife	0	0	0	0	0	1

Figure 6. One Hot Encoder Values

Now that the methods for encoding the data have been determined, the next step is to choose the algorithms used in our analysis. As with data encoding, there are benefits and drawbacks related to each method.

The target for this project is time, since this is a continuous variable, regression techniques will be employed for analysis. The first method for analysis was the Decision Tree Regression algorithm. Decision trees are very popular and practical for supervised learning due to their quick run time and interpretability. The algorithm starts at the root of the tree and continuously splits the data on features that result in the largest information gain. The goal of the algorithm, is to have "pure" leaves at the end. Sci-Kit Learn has a built-in function, Decision Tree Regressor, which was used along with Label Encoding. Preliminary results show that this algorithm, with its default parameters, result in an R2 score of 0.799. Training this model takes approximately 0.244s, and predictions 0.026s. While these results are encouraging, more will need to be done to access the validity of these results, including performing more in-depth train/test splits as well as parameter tuning, as default parameters were used in obtaining these results.

A similar approach was also used with another model, a Random Forest Regressor. This is an ensemble approach, making use of many individual Decision Trees. This ensemble method has the benefit of combining several, smaller models to hopefully increase regression performance. Compared to individual Decision Trees, Random Forest Regressors are better able to generalize, and are less sensitive to outliers. Similar to the Decision Tree model, label encoding was used with default hyperparameters, and an R2 score was calculated from the training data. This model obtained an R2 score of 0.720, worse than the individual Decision Tree Regressor. Training this model takes approximately 15.33s and predictions take 1.87s. We would expect the R2 score to actually be higher than an individual Decision Tree Regressor, but more analysis will need to be performed to conclude as to why this is not the case. Perhaps with further hyperparameter tuning, and introduction of train/test splits, the

Random Forest Regressor will prove to have better performance.
Except for one, the rest of the features of our dataset are of nominal datatype. Age was the only non-nominal (ordinal) feature present in our dataset.

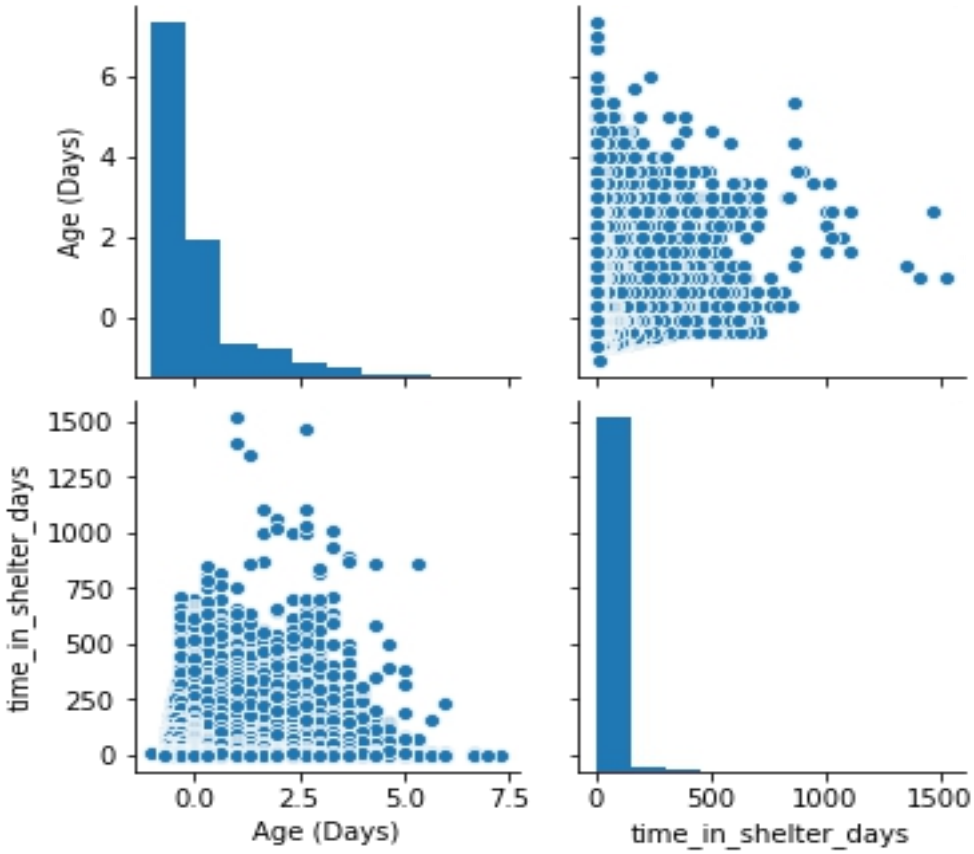

Figure 7. Scatterplot Matrix to View Pair-wise Correlation

We see that the standardized Age (Days) feature has little correlation with the target variable. This is further supported by the correlation matrix below.

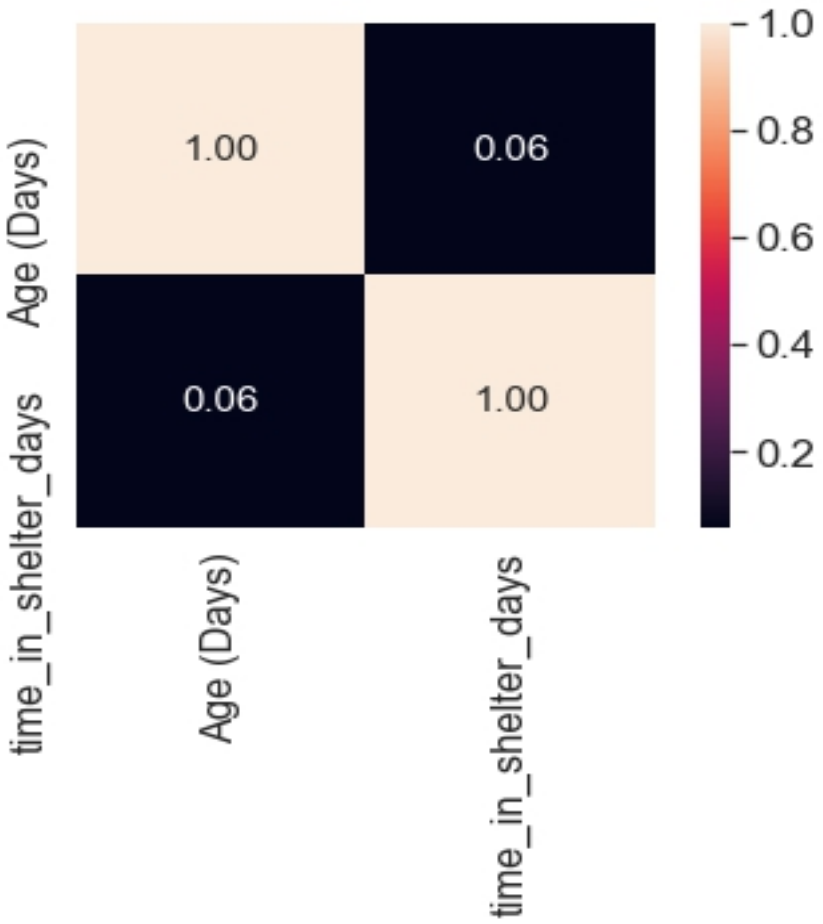

Figure 8. Correlation Matrix

We see that the correlation between Age and the target column is at 0.06. It should be noted that Age is positively skewed- most animals in the shelter are young. We have performed Linear Regression offered by Sci-Kit Learn Library.

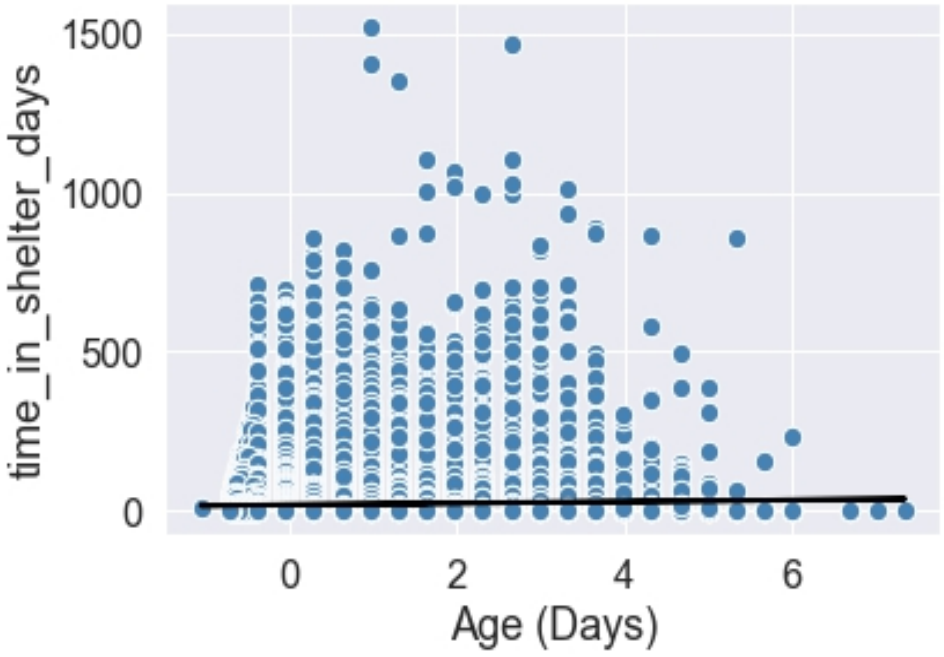

Figure 9. Data Points and Regression Line

The R2 score for the linear regression model was low at 0.003.

Deep learning methods were also attempted, particularly a Neural Network Regressor. In this model, many layers of individual neurons are used to continuously predict the Time in Shelter. With this model, one-hot encoding was used, rather than label encoding. This is essential, as with Deep Learning, it is essential that the neural network does not attempt to resolve relationships between encoded data that do not actually exist. However, because one-hot encoding requires individual columns for each level in each attribute, the Breed and Color attributes were not included in the data used. These attributes have hundreds to thousands of levels, and would not be practical to one-hot encode. Many configurations of layers, both in terms of number of layers and layer size were tried, but with poor results. The neural network was unable to learn from the data, showing a steady loss, with random fluctuation, rather than steady decrease in loss as would be expected with a neural network that is learning well. More analysis will need to be performed, and methods implemented to form a neural network which is able to learn from the data and make reasonable predictions. With more hyperparameter tuning and other techniques, perhaps increased performance will be found. In the future, we aim to work on establishing relationship between nominal variables and the target label.