The bulk of the work for Project 1 involved preparation of the dataset. The first thing that I did after loading the dataset and dropping duplicate rows was print the dataset head, shape, info, and data types of each column. This gave a solid initial understanding of what the dataset looked like, how large it was, and which of the columns needed to be converted into usable data types for my ML models. I immediately noticed that the data in the MonthYear column was redundant to the data in the DateTime column and that the Animal ID gave no valuable information when it came to predicting the animal's outcome, so I dropped these two columns. Additionally, I dropped the Outcome Subtype columns, as I realized that they were redundant to the Outcome Type column, and I dropped the Breed column as recommended in the instructions.

The next step was converting continuous columns with string data types into integer values that could be used by an ML model. I first converted the Date of Birth and DateTime columns, which were initially strings, into datetime objects, and dropped the timestamp component of DateTime, as I figured that an event's time of day was likely a very noisy data point and that shelter workers likely logged events later than they occurred. This would be useful for data analysis, but I later converted the DateTime column into an integer representing the number of days since the first recorded DateTime that the animal was checked in and dropped the Date of Birth column, as I realized that it was redundant to the animal's Age Upon Outcome. I next converted the Age Upon Outcome column from a string value into integer values representing the number of days that each animal had been alive at the time of their outcome.

Once I had done this, I began converting categorical columns into one-hot-encoded columns. I did this for the Animal Type, Color, Sex Upon Outcome, and Name columns. The Animal Type and Sex Upon Outcome conversions ran very smoothly, as there were only 4 possible Type values and 5 possible Sex values. Color and Name were a bit trickier, as there were 60 different possible Color values and almost no two animals shared the same name (although many animals were either unnamed or had no recorded name). To remedy the Color situation, I decided to drop colors that provided little additional information compared to the rest of the dataset using an analysis that I'll discuss later. Additionally, to create useful value from the Name column, I decided to create a new column that simply indicated whether the animal had a name or not. I noted that this could be covariate with an animal's age—perhaps puppies were more likely to be unnamed—and I later confirmed this suspicion but found that the named status still provided unique insight into the probable outcome of an animal.

Now that I had prepared my data, I began investigating relationships between different feature columns and the outcome column, and the relationships between a few feature columns, with the goal of identifying useless data that could be dropped. To visualize this, I plotted a bar chart where each possible feature value was its own bar that was split into two sections sized proportionally to the number of animals belonging to that category that were adopted vs. transferred. I was able to apply this method of analysis to continuous variables by grouping those variables into discrete categories that I could then plot as bars. I noted that this could be a poor decision in the event that continuous variables were grouped too broadly or too granularly to capture all possible trends. To correct for this, I made sure to investigate differently sized groupings for each continuous variable. For example, when investigating the relationship between DateTime and Outcome, I grouped DateTime into four 3-year periods and forty eight 3-month periods. This gave me an understanding of both the large-scale (3-year) and seasonal shifts in the shelter systems adoption patterns.

I also performed some analysis to investigate the value of color, as mentioned earlier. For color, I first dropped any column that had fewer than 100 samples, as I figured that these data were likely very noisy. Next, I noted that some colors likely only occurred in particular animals. For example, gray tabby is a color that only occurs in cats. I used this observation by computing a predicted adoption rate based on the weighted average of adoption rates in animal types that included a particular color and then comparing the predicted adoption rate with the true adoption rate per color. For example, if 96% of buff-colored animals were dogs and 4% were cats, then I calculated a predicted adoption rate of 0.96**adoption rate of dogs + 0.04*adoption rate of cats* and subtracted this from the true adoption rate of buff animals. Once I had done this for all columns, I dropped the color columns with less than a 2% difference between the actual and expected adoption rates. This was based on the assumption that the marginal difference could easily be the result of a noisy dataset and that those colors gave me very little information that wasn't already portrayed by the animal type. I used a similar calculation to investigate the value of the named feature, where I compared the predicted outcome based on the weighted average of adoption rates among the age demographics of named/unnamed animals. I found that while the named status of an animal did depend on its age, the named status still provided additional information about the probability of adoption.

Aside from the color categories that I found redundant, my analysis proved that every feature was useful in predicting an animal's outcome. The main benefit that I got from my analysis was the discovery that the named status of an animal strongly depended on its age and that outcome type was most strongly indicated by an animal's sex and age. I was able to use these discoveries by filling in missing age values with the average of the ages of all other animals with the same named status and filling in missing outcome types with the mode of the outcome types of all other animals with the same sex and age. These were the last adjustments that my dataset needed to be ready for machine learning.

Once my dataset was prepared and analyzed, training my ML models was relatively straightforward. I split my dataset into training and test sets using a 25% test size and stratifying the data by outcome type. I then trained a basic KNN Classifier using 3 nearest neighbors. This model achieved the best results when evaluated for recall, scoring 0.85 on the test set and 0.93 on the train set. Next, I trained 4 more KNN Classifiers with GridSearchCV using 5 folds to choose the optimal number of neighbors between 1 and 99. These models were optimized for accuracy, recall, precision, and F1, respectively. I found that the KNN model that was optimized for accuracy used just the single nearest neighbor and achieved 77% accuracy on the test set and 99% accuracy on the train set (highly overfit). The model that was optimized for precision selected just the 2 nearest neighbors and scored 100% precision on the train set and 85% precision on the test set (again, highly overfit). The model that was optimized for F1 used the 85 nearest neighbors and scored 82% F1 on the train set and 81% F1 on the test set. Finally, the model that was optimized for recall selected the 99 nearest neighbors and scored 92% recall on both the test and train sets. Because this model selected the upper bound of neighbors, I ran it again, allowing it to choose from [100, 120, 140, 160, 180, 200] nearest neighbors. This time it chose 200 nearest neighbors, but it scored the same as before. Therefore, I decided to keep the original result. My last model was a linear classifier that I trained using perceptron loss and an alpha of 0.05. This model scored poorly on accuracy, recall, and F1 but achieved 95% precision on both the train and test sets. This result tells me that the model likely predicted that most samples belonged to the adopted class and, as a result, had very few false negatives. For the full results, including how well each model performed on metrics other than they were optimized for, please view my notebook.

To determine the most valuable metric for classifying these models, I will assume that a shelter is using a model to predict which animals will be transferred and euthanize them in order to clear space for adoptable dogs. In this case, we should grade models on recall in order to reduce false negatives—we don't want to predict that an adoptable dog won't be adopted and then accidentally kill them.