Preparing the data:

I dropped all duplicate rows. I filled in all of the missing name values with “No Name”. I took out all rows missing both “Outcome Type” and “Outcome Subtype.” I filled in rows missing “Outcome Type” that had “Outcome Subtype” based on the Outcome subtype. I filled in all missing “Outcome Subtype” values with the Outcome Type “Adoption” with “Adopted.” This got rid of all of the missing values.

Then, I found that the DateTime column had different formats (some didn’t include time zones), so I made them all the same to go to a single format (UTC). I then turned all of the “Date of Birth” entries into the datetime entries. I then calculated the age upon outcome based on the “DateTime” and “Date of Birth” columns in days. I then encoded the “MonthYear” Category as the number of months in the 21st century instead of months and the year. I also found that the “Age upon Outcome” category had a few negative ages (meaning they were adopted before they were born) and removed the small amount of those. I then found duplicates where only the “Animal ID” was different, and dropped those duplicates as well.

Once all of the data was cleaned, I started dropping categories that I felt weren’t super relevant. I dropped:

- Date of Birth: Redundant with Age upon Outcome and MonthYear
- DateTime: Redundant with MonthYear and I didn’t think time of day mattered.
- Breed and Color: Too many categories and couldn’t do one-hot encoding for them.
- Name: Too many categories and probably not too relevant.
- AnimalID: Probably just not very relevant

Then, I did one-hot encoding on “Animal Type”, “Sex upon Outcome”, “Outcome Type”, and “Outcome Subtype”. After that, all of my data was either numerical or categorial.

Finally, I also went through univariate analysis on the dataset before training. I did histograms for the “MonthYear” and “Age upon outcome”, splitting “MonthYear” appropriately into each month. I did these plots separately for the transferred and adopted animals. What I got from the data was that certain months out of a year do tend to have higher rates. Also, when animals have their birthdays, there is a big spike in rates as well. The issue is that both of these rates are roughly the same for both transferred and adopted animals, so it didn’t seem like it was a function of the outcome, but more a trend of animals being moved during those events.

Unfortunately it wasn’t super possible to look at count plots for categorial data because the variables either had too many categories (breed, color) or were basically the Outcome Type (Outcome Subtype).

In terms of general dataset insight, I learned that data can be very unclean and that cleaning up the dataset if often most of the work in comparison to the training aspect. You have to really get a good picture of the data to clean it up, which often takes a lot of work.

In terms of this dataset, I did find trends, but none that were related to the outcome type. I just found a lot of small exceptions that I needed to clean up.

Training the model:

For the procedure, I split the data into training and testing sets, with the Outcome Type being y, or the dependent variable, making sure to use stratify based on the Outcome Type to ensure that it maintains roughly the proportion of each class of the dependent variable. I also fixed the seed to ensure that the results are reproducible.

Then for the regular Linear Classifier and the K-Nearest Neighbor Classifier, I loaded the model with an arbitrary parameter (the ones used in class) and trained it on the training dataset. I then ran it a few times with different learning rates or different n-values to see if I could roughly guess the best hyperparameter;

For the K-Nearest Neighbors Classifier using Grid search CV, I first looked at a really wide range of values (1 – 1000), jumping from each hundred so I only tried 10 values. Then when I got that the best one was 1, I did 1-100 for 10 values, doing 1, 11, 21, and so on until 101. I then got 1 again as the best value of n and did 1-21 just to be safe, trying every value. In the end it seemed like n=2 was the best value. (The scoring was based on macro f1-score.)

The model doesn't perform too great, but it performs decently well. For the Linear Classifier, it has an accuracy of 79% while the best value for the K-Nearest Neighbors Classifier also had an accuracy of 79%. It's not terrible, but it's definitely not great.

I thought that macro f1-scoring was the most important metric for this problem as is just give the number of correct classifications. The reason I say this is that I wouldn't say that classifying either animals either as adopted or transferred has more weight than the other and I feel like the consequences of false negatives and positives are pretty much the same for both. But since the classes are mildly imbalanced, I feel that the macro f1-score does the best job of representing how good the model does.

Honestly, I don't feel very confident in the model since it's only 80%. I feel confident enough using it in this scenario since the stakes aren't very high in my opinion, but if it were higher stakes, such as in the medical field, I wouldn't want to use or depend on this model.