1. Data Preparation

The raw dataset contains 131,165 rows and 12 columns that each describe various characteristics of animals in Austin shelters.

The first observation I had was that all the columns were stored as object types, so I had to do quite a few preprocessing steps to format the data properly.

1. **Date Conversion:**
    - Converted Date of Birth, DateTime, and MonthYear to datetime64 to allow for chronological analysis based on the actual date.
2. **Duplicate Removal:**
    - A total of 17 duplicates were found and removed them using animals.drop_duplicates(inplace=True).
3. **Missing Values:**
    - Replaced NaN values in *Name* and *Outcome Subtype* with "Unknown".
    - Filled Outcome Type missing entries using the mode, which was (‘Adoption’).
    - After cleaning, the dataset had 131,148 valid records.

4. **Age Normalization:**
    - Converted Age upon Outcome to a consistent numeric format (days) using a function that:
        - Days = 1 × value
        - Weeks = 7 × value
        - Months = 30.44 × value
        - Years = 365.25 × value
    - I created a new numeric column:
        - Age upon Outcome (days) with a mean of ~577 days (~1.6 years).

5. **Irrelevant Columns:**
    - Dropped columns that were non-predictive or redundant:
        - Animal ID, Name, Date of Birth, DateTime, MonthYear, Outcome Subtype
    - Final dataset after removal: 131,148 rows × 7 columns.

6. **Categorical Conversion & Encoding:**
    - Converted these to categorical types:
        - Outcome Type, Animal Type, Sex upon Outcome, Breed, Color.

After reformatting and cleaning, the dataset had 131,148 rows and 3,150 columns (~400 MB). Finally, per instructions, I dropped the Breed column before modeling.

2. Exploratory Insights

Animal Type:

- Dogs (~68 K) and Cats (~63 K) dominate the dataset while birds and livestock are much rarer.

Outcome Type:

- Most records are Adoptions (~73 K), followed by Transfers (~48 K).

Sex upon Outcome:

- Neutered Male(~43 K) and Spayed Female(~41 K) animals form the majority, and intact animals are uncommon(~17 combined).

Age Distribution:

- Most animals are under 2 years old, mainly “2 months”, “1 year”, and “2 years”.

Monthly Trends:

- Peaks around 2017–2019.
- Sharp decline after 2020, Maybe due to the COVID-19 pandemic impact on shelter operations.

Breed & Color:

- Most common breeds are Domestic Shorthair Mix and Labrador Retriever Mix.
- Most common colors: Black/White, Brown Tabby, and Brown/White.

These distributions show the dataset is slightly skewed toward adopted dogs and cats that are already neutered/spayed with the rest being much more uncommon.

3. More Data Preperation

Simplifying Categories

To reduce dimensionality after one hot encoding I collapsed rare breeds and colors into "Other" and only kept top 20 breeds and top 10 colors.

One-Hot Encoding

- Applied pd.get_dummies() on:
  ['Animal Type', 'Sex upon Outcome', 'Breed', 'Color']
  using dtype='int8' for memory efficiency.
- This generated ~45 columns in total and these new dummy variables were numeric and ready for my modeling in step 2. .

Dropped all breed dummies after to avoid complication as instructed in the beginning of step 2.

4. Model Training Setup

To predict Outcome Type (Adoption or Transfer):

1. **Target Variable:** Outcome Type converted to binary (1 = Adoption, 0 = Transfer)

2. **Train/Test Split:** Used *train_test_split* with 30% test data and *stratify=y* to preserve class ratios.

3. **Feature Selection:** Included *Age upon Outcome (days)* and a subset of dummy variables *(Animal Type_, Sex upon Outcome_)* to avoid memory overload where this wasn't done.

4. **Models Trained:**

    - Baseline KNN (k = 5)

    - Optimized KNN using GridSearchCV (*k ∈ [1 … 15], 5-fold CV*)

- Linear Classification (Logistic Regression)

All models used *accuracy, precision, recall,* and *f1-score* as metrics, computed with *classification_report()*.

5. Model Performance

Model	Accuracy	Precision (macro)	Recall (macro)	F1 (macro)
KNN (k = 5)	0.8248	0.8142	0.8010	0.8066
KNN (Grid Search k = 14)	0.8473	0.8470	0.8171	0.8281
Perceptron	**0.8576**	**0.8751**	**0.8175**	**0.8349**

Observations:

- Baseline KNN (k = 5) performed pretty well with ~82% accuracy. This solid for a simple non-parametric classifier so it is reasonable.
- GridSearchCV optimization (k = 14) improved KNN's accuracy to ~85%, showing that tuning neighborhood size does significantly benefit model balance.
- Perceptron achieved the highest accuracy of ~86% and the best F1-score of ~0.83, showing that a linear model generalized well and effectively separated the classes.

6. Model Confidence & Interpretation

- The F1-score was prioritized in my analysis since it balances precision and recall, reducing both false positives and false negatives.
- Perceptron's consistent performance across both shows it generalizes effectively and doesn't overfit like KNN might when the k is small.
- The high accuracy (~86%) and strong F1 values indicate that the model is reliable and confident in predicting animal outcomes based on features such as age, animal type, and sex upon outcome.