## I. COE 379L; Project 1 - Use of AI; Mahin Naveen - mn27995

## II. Prompts & Responses:

- [1] Tool: ChatGPT

Prompt: I’m trying to parse the “Age Upon Outcome” column, but the dataset uses mixed units like “2 years,” “4 weeks,” and “5 months.” Can you show me how to convert all those into a single numeric column in days using Python?

Output: ChatGPT provided a short function using regex and conditional checks to standardize all age values into days. I slightly modified it to fit my variable names and testing.

```
def parse_age_to_days(text):
    if pd.isna(text): return np.nan
    s = str(text).strip().lower()
    m = re.match(r"\s*(\d+(?:\.\d+)?)\s*(year|month|week|day)s?", s)
    if not m: return np.nan
    val = float(m.group(1))
    unit = m.group(2)
    if "year" in unit: return val * 365
    if "month" in unit: return val * 30
    if "week" in unit: return val * 7
    return val
```

(Note: used to guide my implementation; code in notebook includes my edited version)

- [2] Tool: ChatGPT

Prompt: I want to extract gender and fixed status from the “Sex Upon Outcome” column. For example, “Neutered Male” should become Male and Fixed = 1, “Intact Female” should be Female and Fixed = 0. How can I do this cleanly without using complex one-liners?

Output: The response suggested defining a helper function that returns a tuple (sex, fixed) and applying it with pandas .apply(). This clarified how to structure a readable solution.

```
def split_sex_upon_outcome(text):
    if pd.isna(text): return ("Unknown", 0)
    s = text.title()
    sex = "Male" if "Male" in s else "Female" if "Female" in s else "Unknown"
    fixed = 1 if "Neutered" in s or "Spayed" in s else 0
    return (sex, fixed)
```

- [3] Tool: ChatGPT

Prompt: When I train my KNN model, it takes forever to run. Is there a way to test my code faster without changing the overall logic of the project?

Output: ChatGPT explained that KNN is slow for large datasets because of distance calculations, and suggested testing the pipeline on a smaller sample first to confirm correctness before running the full dataset. I used this advice to validate my preprocessing and model steps efficiently.

(No direct code copied, just made adjustments as needed.)

- [4] Tool: ChatGPT

Prompt: I’m unsure whether to use accuracy or F1 score as my main metric for this problem. Can you explain which one makes more sense for predicting “Adoption” vs “Transfer”?

Output: The answer explained that since both classes are important and the dataset may be slightly imbalanced, F1 score better captures the balance between precision and recall. I used this to justify my report discussion in Part 3.

- [5] Tool: ChatGPT

Prompt: My model training stopped with a KeyError for “Age Upon Outcome,” but the column name exists. Could this be a formatting issue in the CSV?

Output: ChatGPT helped me identify that the column headers had inconsistent spacing and capitalization. It provided a short snippet to normalize and rename columns automatically by stripping spaces and replacing variants like “ageuponoutcome” → “Age Upon Outcome.”

```
df.columns = [c.strip() for c in df.columns]
renameMap = {}
for c in df.columns:
    if c.lower().replace(" ", "") == "ageuponoutcome":
        renameMap[c] = "Age Upon Outcome"
df = df.rename(columns=renameMap)
```

This fixed the KeyError and allowed the rest of the notebook to run smoothly.