

## Use of AI

-----

[1]. Tool: ChatGPT

Prompt: Convert “Age upon Outcome” to numeric days

Output:

```
def age_to_days(s):
    if pd.isna(s): return np.nan
    m = re.match(r'^\s*(\d+(?:\.\d+)?)\s*([A-Za-z]+)\s*$', str(s))
    if not m: return np.nan
    v = float(m.group(1)); u = m.group(2).lower()
    mult = {'day':1, 'days':1, 'week':7, 'weeks':7, 'month':30, 'months':30, 'year':365, 'years':365}.get(u)
    return v*mult if mult else np.nan

if 'Age upon Outcome' in df.columns:
    df['age_outcome_days'] = df['Age upon Outcome'].map(age_to_days)
    df.drop(columns=['Age upon Outcome'], inplace=True)
```

[2]. Tool: ChatGPT

Prompt: Convert high cardinal features into numerical type with frequency-encoding

Output:

```
high_card_cols = ['Breed', 'Color', 'Name']

for col in high_card_cols:
    freq_map = df[col].value_counts().to_dict()
    df[col + '_freq'] = df[col].map(freq_map)
    df.drop(columns=[col], inplace=True)
```

[3] Tool ChatGPT

Prompt: Make any other features with object type into categorical type

```
obj_cols = df.select_dtypes(include='object').columns
df[obj_cols] = df[obj_cols].apply(lambda s: s.astype('category'))
```