Preprocessing Data

the cleanup continues…..

Assessing and Dealing with Missingness

df.isnull().sum()

Approaches:

- drop column
- replace NaN values using a thoughtful strategy – what does this mean?
- (considerations - type of data, outliers,

Can use scikit-learn Imputer:

```
from sklearn.preprocessing import Imputer my_imputer = Imputer()
data_with_imputed_values = my_imputer.fit_transform(original_data)
#by default imputes with the mean, other strategies are median, mode
```

http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.Imputer.html

Coding categorical data to be numeric

```
cols_to_transform = ['breed', 'dominant_color', 'borough', 'zip_code']
dogs_dummied = pd.get_dummies(dogs, columns = cols_to_transform)
```

***for binary columns (gender, spayed_or_neutered, guard_or_trained) it's better to simply replace the values so they don't get double counted, for example:**

```
dogs_dummied['gender'] = [0 if gender == 'M' else 1 for gender in dogs_dummied['gender']]
```

https://pandas.pydata.org/pandas-docs/stable/generated/pandas.get_dummies.html

Handling imbalanced classes

- downsample the overrepresented class
- upsample the underrepresented class

```
from sklearn.utils import resample

#separate the classes
yes = dogs_dummied[dogs_dummied.spayed_or_neutered == 1]
no = dogs_dummied[dogs_dummied.spayed_or_neutered == 0]

#downsample the yes group
downsampled_yes = resample(yes, replace = False, n_samples = 18040, random_state =42)

#put the downsampled yes dogs back with the no dogs and check the class counts again
balanced_dogs = pd.concat([downsampled_yes, no])
balanced_dogs.spayed_or_neutered.value_counts())
```

http://scikit-learn.org/stable/modules/generated/sklearn.utils.resample.html

Feature selection, feature extraction, feature engineering

…iterative process

Standardization

Transforming features to a normally distributed shape

```
from sklearn import preprocessing
X_scaled = preprocessing.scale(X)
```

- **StandardScaler creates a transformer that can be *reapplied* to test data so that the transformation is consistent**
- **StandardScaler can be used as part of an sklearn.pipeline**

```
from sklearn.preprocessing import  StandardScaler
scaler = StandardScaler()
```

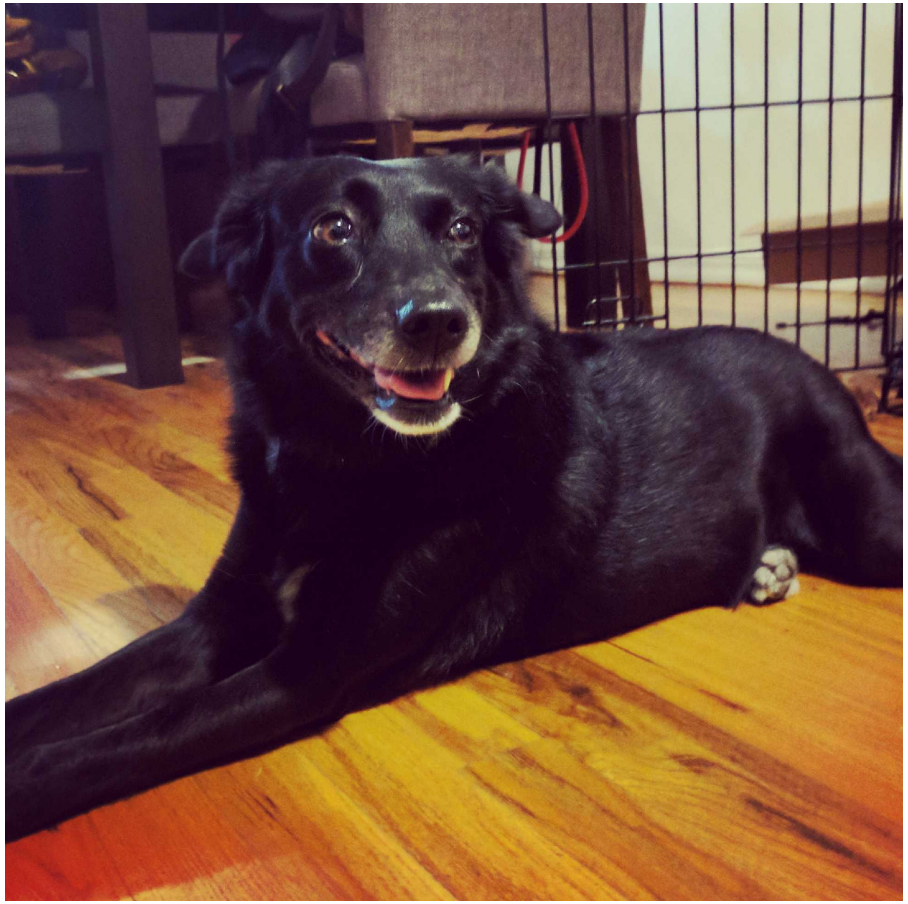
Normalization

Simple definition- transforming features to use a common scale; may involve just scaling a feature, or may involve further transformation, sometimes to align values to a normal distribution.

```
#normalizing a feature set X
X_normalized = preprocessing.normalize(X, norm = '12')    #can use either 11 or 12 norms

#for pipeline
normalizer = preprocessing.Normai\lizer().fit(X)  #fit simply applies the normalizer to the data
normalizer.transform(X)
normalizer.transform(some_other_array)
```

Dogs of New York dataset
2013 - NYC Department of Health & Hygiene



https://fusiontables.google.com/data?docid=1pKcxc8kzJbBVzLu_kgzoAMzqYhZyUhtScXjB0BQ#rows:id=1

Dogs of NYC | WNYC

Source data for WNYC's 'Dogs of NYC' project.
NYC Dept of Health and Mental Hygiene - Edited on 2013 January 24

File Edit Tools Help | Rows 1 | Cards 1 | Map of borough

- Share...
- New table...
- Open...
- Rename...
- Make a copy
- About this table
- Geocode...
- Merge...
- Find a table to merge with...
- Create view...
- Import more rows...
- Download...

Filter: dog_name CONTAINS IGNORING CASE 'skippy' AND birth = 'Dec-01'

1-1 of 1

dog_name	gender	breed	birth	dominant_color	secondary_color	third_color	spayed_or_...	guard_or
Skippy	M	Miniature Pinscher	Dec-01	BLACK	RUST	n/a	Yes	No

Resources

https://elitedatascience.com/imbalanced-classes

https://unsupervisedpandas.com/python/supervised-classification-preprocessing/

http://scikit-learn.org/stable/modules/generated/sklearn.utils.resample.html

https://pandas.pydata.org/pandas-docs/stable/generated/pandas.get_dummies.html

http://scikit-learn.org/stable/modules/preprocessing.html

http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

https://www.kaggle.com/dansbecker/handling-missing-values