# Cats vs Dogs

Sarah Scolnik
DSI-US-06

photo credits: https://www.reddit.com/r/cats/comments/a7umba/the_best_christmas_pictures/ https://www.instagram.com/jenbeecardco/

# Project Goal:
# Classification of comments from two subreddits

Process:

- Data collection: Reddit and pushshift.io APIs
- Data cleaning and EDA
- Preprocessing and Modeling
- Evaluation
- Conclusions

# Data Collection

Reddit API:

- 100 posts per request & limited to 1,000 most recent posts total
- more difficult to download comments
- more cleaning of text for html tags, emoji, etc. may be required

pushshift.io:
(open data initiative to make social media data available for researchers and academic institutions)

- 500 posts per request & no limit to requests
- submissions and comments available
- searchable by various parameters

# Data Collection

Using the pushshift.io Reddit API, I downloaded:

- 20,000 submissions (10,000 each from /r/cats and /r/dogs subreddits)
- 20,000 comments (10,000 each from /r/cats and /r/dogs subreddits)

I analyzed comments for this project, as cat submissions are mostly photos and dog submissions are mostly text (/r/dogs doesn't allow photo posts, only links to photos).

# Data Cleaning / Preprocessing

Cleaning:

- dropped duplicates (mod bot messages, etc.)
- re.sub() to remove: html, hyperlinks, punctuation, words with 2 or fewer letters, whitespace including line returns, non-standard characters (emoji)
- after cleaning: 20,000 -> 18,000 records

Preprocessing:

- lemmatization (dogs -> dog, cats -> cat)
- added to stop words: ‘ha’, ‘wa’, ‘did’, ‘doe’, ‘don’, ‘got’, ‘doesn’, ‘getting’, ‘going’
- train/test split (used default 0.25 test, stratify, shuffle)
- classes are balanced, each approx. 50%

# EDA: most frequent words

| cats | | | | dogs | | |
|---|---|---|---|---|---|---|
| beautiful | kitty | really | | breed | like | sure |
| best | know | sorry | | breeder | look | thing |
| cat | life | sure | | day | lot | think |
| cute | like | thank | | dog | love | time |
| day | little | thing | | food | make | training |
| food | lol | think | | good | need | vet |
| good | look | time | | help | people | walk |
| home | love | vet | | home | pet | want |
| just | make | want | | just | puppy | work |
| kitten | old | year | | know | really | year |

# Data Preprocessing

CountVectorizer:
Baseline logistic regression model train/test scores:
0.9290 / 0.8528

| | coef_ | abs_coef |
|---|---|---|
| **dog** | -15.559583 | 15.559583 |
| **cat** | 10.286536 | 10.286536 |
| **kitty** | 6.147446 | 6.147446 |
| **pup** | -5.449378 | 5.449378 |
| **puppy** | -5.426464 | 5.426464 |
| **kitten** | 4.118441 | 4.118441 |
| **mix** | -3.050064 | 3.050064 |
| **crate** | -3.009038 | 3.009038 |
| **breed** | -2.830657 | 2.830657 |

Tf-idf:
Baseline logistic regression model train/test scores:
0.9018 / 0.8484

| | coef_ | abs_coef |
|---|---|---|
| **cropping** | -15.702207 | 15.702207 |
| **buyer** | 10.405316 | 10.405316 |
| **housebreaking** | 6.195553 | 6.195553 |
| **petroleum** | -5.441822 | 5.441822 |
| **phenobarbital** | -5.368938 | 5.368938 |
| **hound** | 4.131172 | 4.131172 |
| **launch** | -3.063162 | 3.063162 |
| **columbia** | -2.986207 | 2.986207 |
| **brachy** | -2.824862 | 2.824862 |
| **questionable** | -2.768942 | 2.768942 |

# Data Preprocessing

Stop Words:

Baseline logistic regression model using standard English stop words:  train/test scores: 0.9290 / 0.8528, using additional stop words: 0.9290 / 0.8543

Baseline random forest model using standard English stop words: features with highest feature importance values included “wa”, “don”, “ha”, “isn”
Adding these to stop words didn’t have much effect on this model - before train/test scores: 0.9819 / 0.8115, after: 0.9819 / 0.8113

| | coef_ | abs_coef |
|---|---|---|
| dog | -15.559583 | 15.559583 |
| cat | 10.286536 | 10.286536 |
| kitty | 6.147446 | 6.147446 |
| pup | -5.449378 | 5.449378 |
| puppy | -5.426464 | 5.426464 |
| kitten | 4.118441 | 4.118441 |
| mix | -3.050064 | 3.050064 |
| crate | -3.009038 | 3.009038 |
| breed | -2.830657 | 2.830657 |

| | feature_importances_ |
|---|---|
| dog | 0.102354 |
| cat | 0.040045 |
| kitty | 0.010908 |
| puppy | 0.009497 |
| really | 0.009427 |
| just | 0.009030 |
| pup | 0.007590 |
| breed | 0.007191 |
| walk | 0.006820 |
| know | 0.006277 |

# Data Preprocessing

n-grams: (1 - 3):

CountVectorizer & Logistic regression: top 50 features all 1-grams except ‘just need’, train/test scores: 0.9351 / 0.8492

TfidfVectorizer & Logistic regression: top 50 features all 1-grams except ‘sound like’, train/test scores: 0.9048 / 0.8521

# Models / Tuning

Logistic regression:
Gridsearch best params: C = 1.0, penalty: l2 (ridge)
Train / test scores: 0.8481 / 0.8543

Random forest:
Gridsearch best params: max depth: None, n_estimators: 30
Train / test scores: 0.8269 / 0.8227

Multinomial naive Bayes:
Gridsearch best params: alpha: 0.5
Train / test scores: 0.8086 / 0.8192

# Conclusions

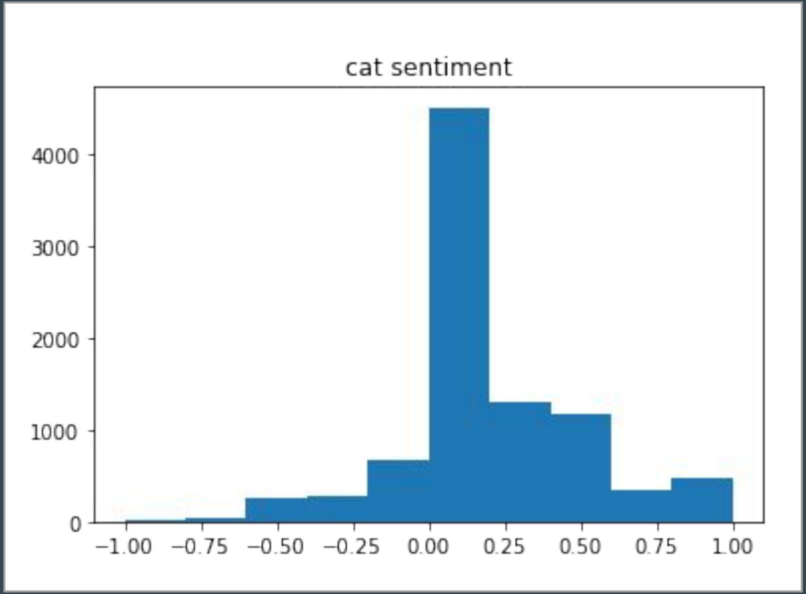
Cats vs Dogs: The differences outweigh the similarities for NLP and classification modeling

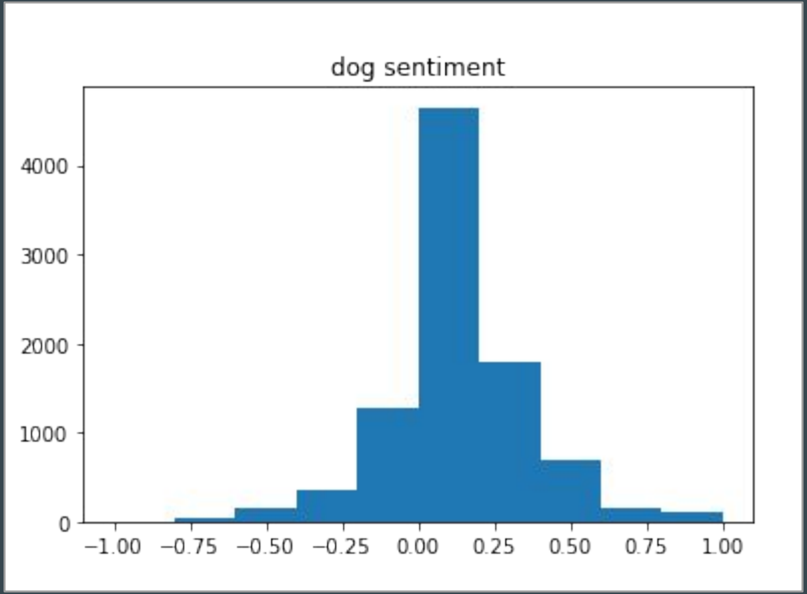
Best scoring model: Logistic regression, Train / test scores: 0.8481 / 0.8543

Potential improvements: collect more training data, do more data cleaning and preprocessing (remove more stop words i.e. numbers, stem/lemmatize i.e. -ing verbs), more intensive gridsearching to optimize models, try more models (boosting, SVM)

# Bonus: Sentiment Analysis

sentiment analysis with TextBlob.sentiment.polarity



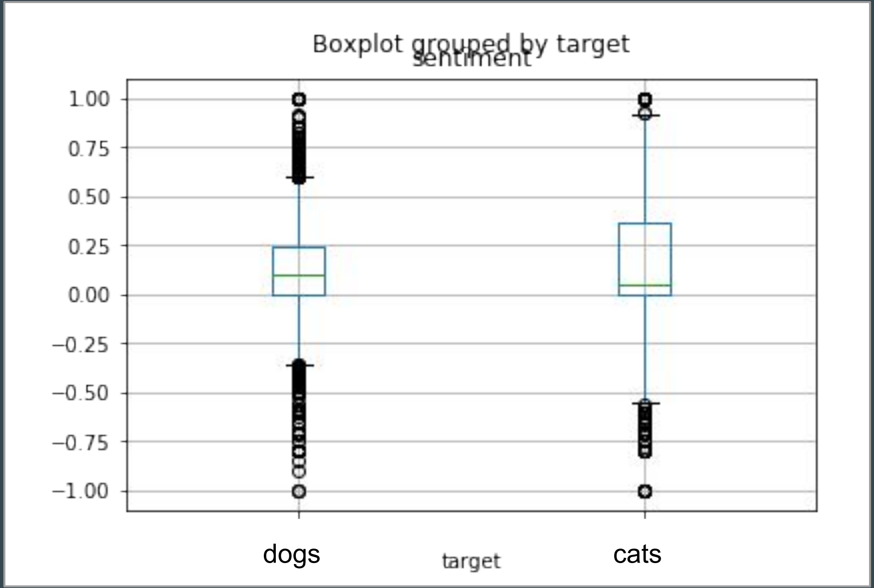



# Bonus: Sentiment Analysis

mean sentiment:
cats: 0.168
dogs: 0.120

median sentiment:
cats: 0.050
dogs: 0.097

cats have more comments 0.5 and above



# Any questions?