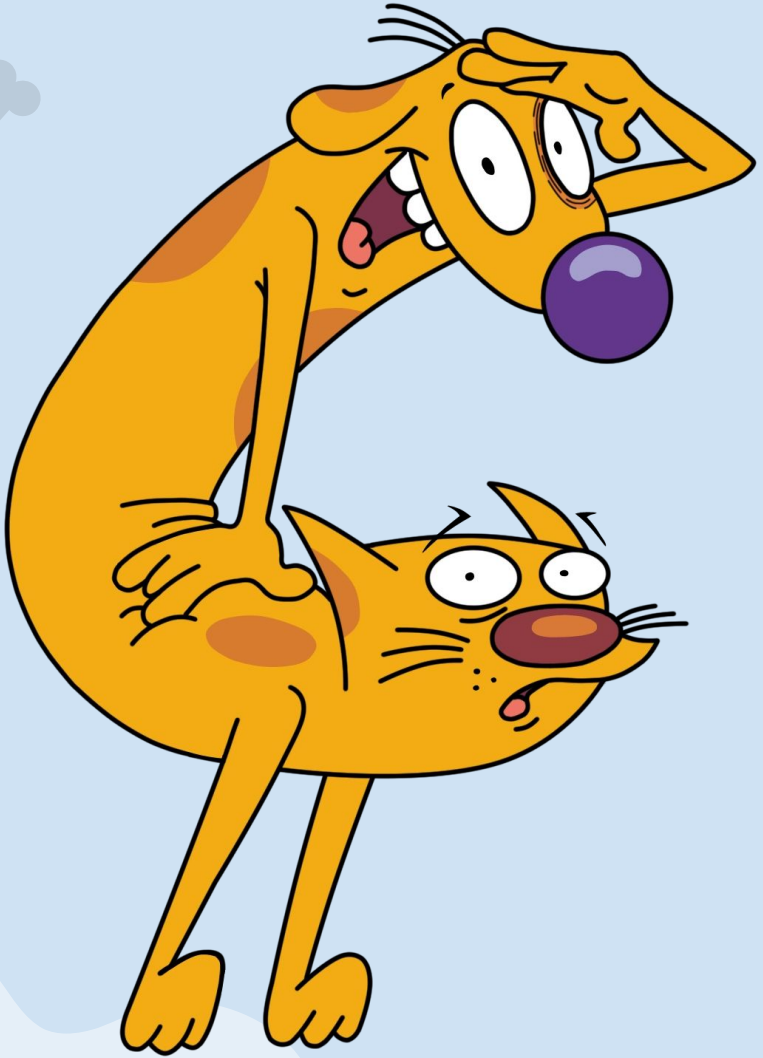


The Best Business Model

Problem Statement

We want to help you identify:

1) Topics of interests and needs of these new dog and cat owners

2) Suggest a viable business model for pet lovers to venture into the pet industry

3) Provide relevant suggestions on creating new services for dog and cat owners and;

Problem Statement

Reddit is a massive collection of forums where people can share social news and content. Essentially, posts are organised according to subject into user-created ‘subreddits’.

Members submit content (such as images, texts, and links) to subreddits, which can then be voted up (‘upvote’) or down (‘downvote’) by other members.

Problem Statement

Our team aims to engineer selective supervised classification models namely:

1. Random Forest Classifier
2. Multinomial Naive Bayes
3. Logistics Regression Classifier

Data Collection

1) Scrap 20,000 posts (body text)
   10,000 for each subreddit

2) Preliminary Analysis
   Selftext from Cats: 8593 of which Blanks (85% of Data)

3) Reason for Blanks???
   Cat owners like to post pictures only!

```
df_cats['selftext'].value_counts().head(3)
```

```
              8593
[removed]      180
[deleted]       14
```

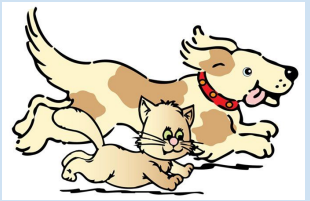
```
df_dogs['selftext'].value_counts().head(3)
```

```
[removed]     792
               68
[deleted]      43
```

SOLUTION

Use Post Title instead

Data Collection - Time Taken



1) **Time to scrap 10,000 posts/subreddit?**

Cats	**Dogs**
Total: 10.7 min	Total: 9.1 min
Mean: 6.5s	Mean: 5.5s

2) **Why scraping posts r/cats take longer???**

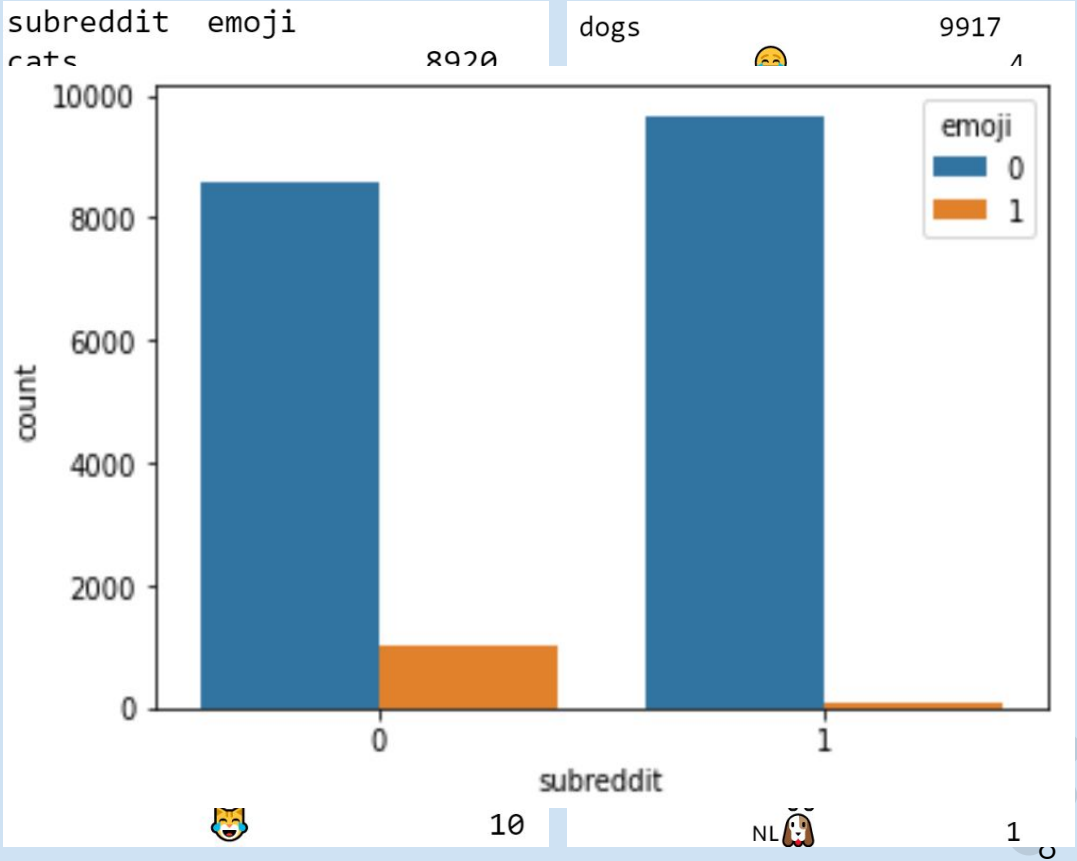
Emojis!!! Cat owners/lovers just ❤ them.

Cleaning Emoji

- Extracting Emoji to a new column

Emoji
Cats: 1072
Dogs: 79

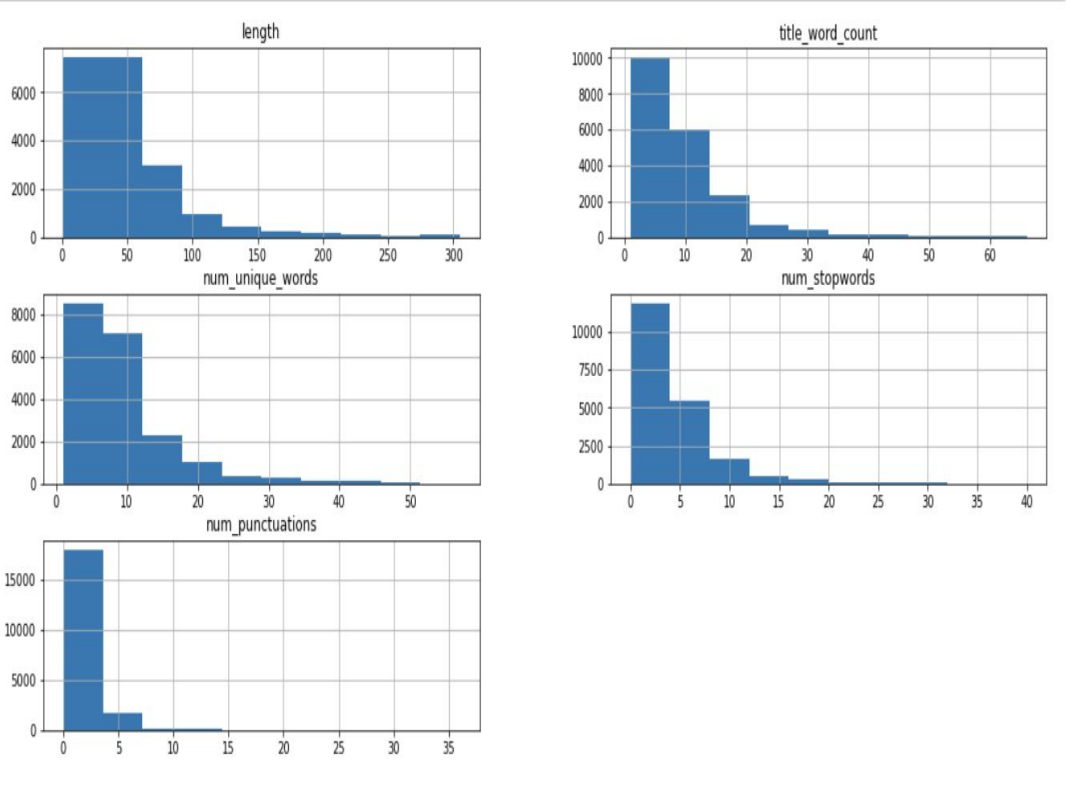
- Insightful Feature Created



Cleaning the Rest

- Lower Case
- Remove Punctuations
- Remove Foreign Languages
- Remove Hyperlinks
- Remove Numbers

EDA



Features engineered:

- Length (i.e. no. of characters)
- Word count of Title
- No. of Unique Words
- No. of Stopwords
- No. of Punctuation

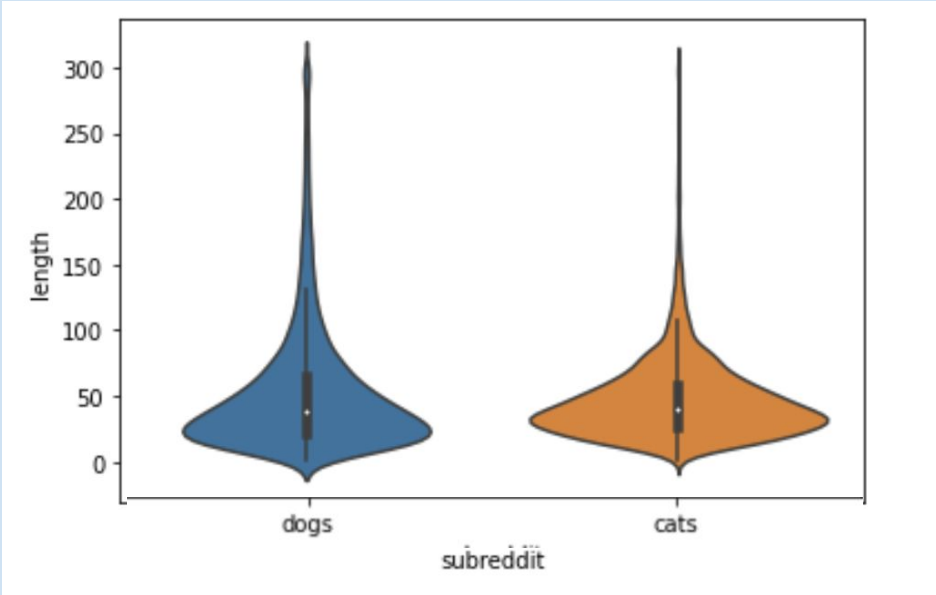
EDA

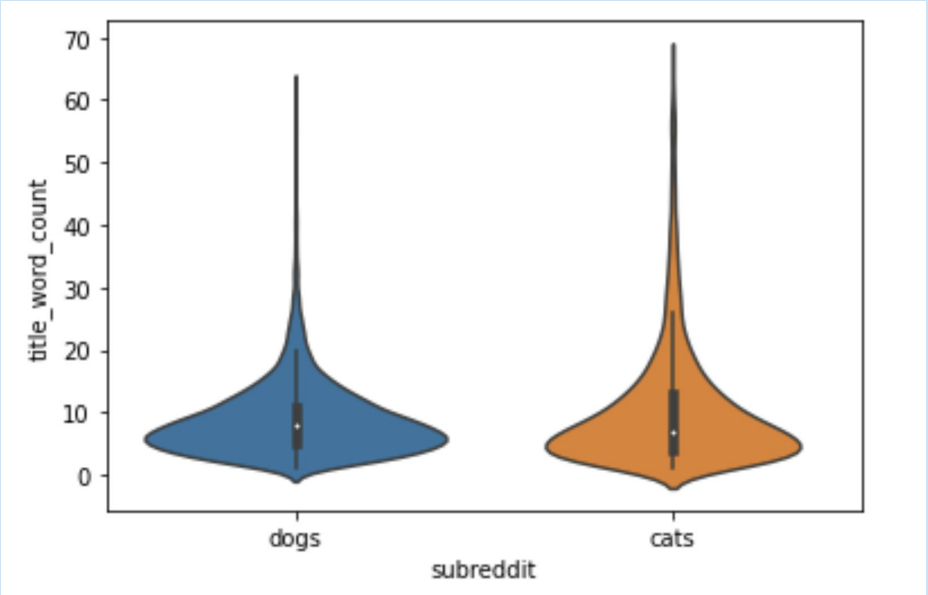
subreddit		0	1
length	count	9937.000000	9975.000000
	mean	53.619704	47.549574
	std	49.837263	32.240927
	min	2.000000	1.000000
	25%	22.000000	27.000000
	50%	38.000000	40.000000
	75%	66.000000	59.000000
	max	304.000000	305.000000
title_word_count	count	9937.000000	9975.000000
	mean	10.442890	8.934236
	std	9.762765	6.269840
	min	1.000000	1.000000
	25%	4.000000	5.000000
	50%	7.000000	8.000000
	75%	13.000000	11.000000
	max	66.000000	62.000000

num_unique_words	count	9937.000000	9975.000000
	mean	9.943645	8.652130
	std	8.643843	5.675194
	min	1.000000	1.000000
	25%	4.000000	5.000000
	50%	7.000000	7.000000
	75%	13.000000	11.000000
	max	57.000000	54.000000
num_stopwords	count	9937.000000	9975.000000
	mean	4.311160	3.455639
	std	4.981325	3.462540
	min	0.000000	0.000000
	25%	1.000000	1.000000
	50%	3.000000	3.000000
	75%	6.000000	5.000000
	max	40.000000	32.000000

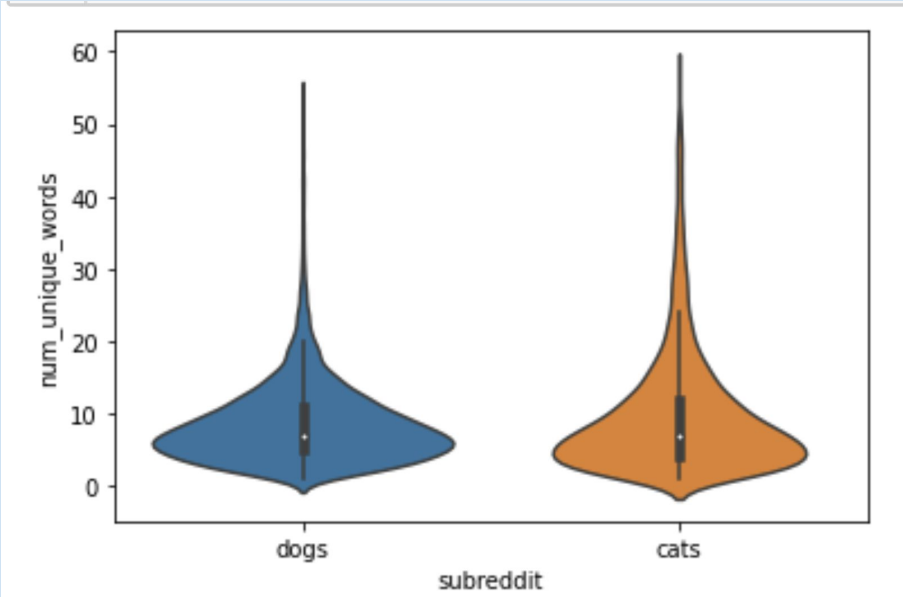
num_punctuations	count	9937.000000	9975.000000
	mean	1.544128	1.269173
	std	2.076847	1.507560
	min	0.000000	0.000000
	25%	0.000000	0.000000
	50%	1.000000	1.000000
	75%	2.000000	2.000000
	max	36.000000	30.000000

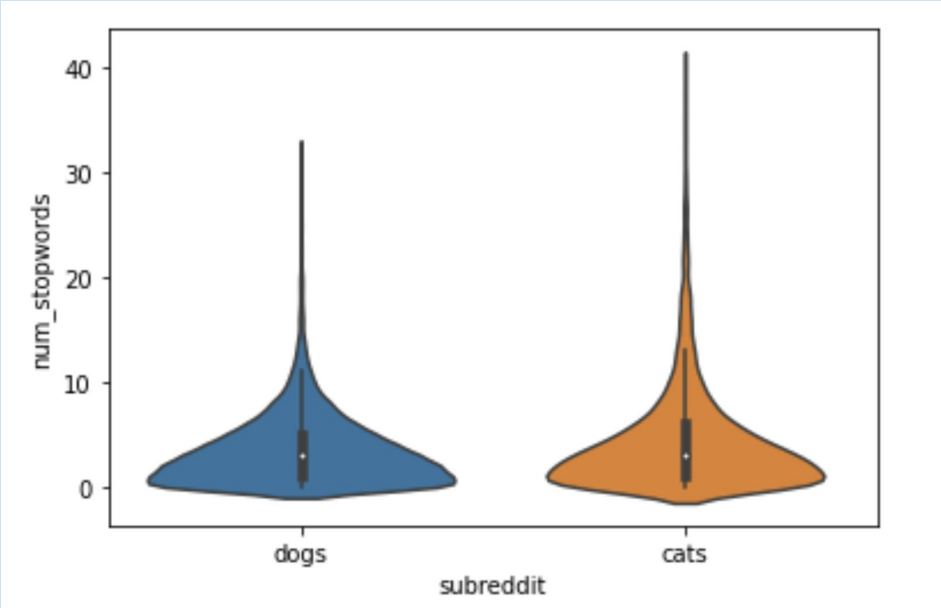
EDA


300
250
200
150
100
50
0
length
dogs
cats
subreddit

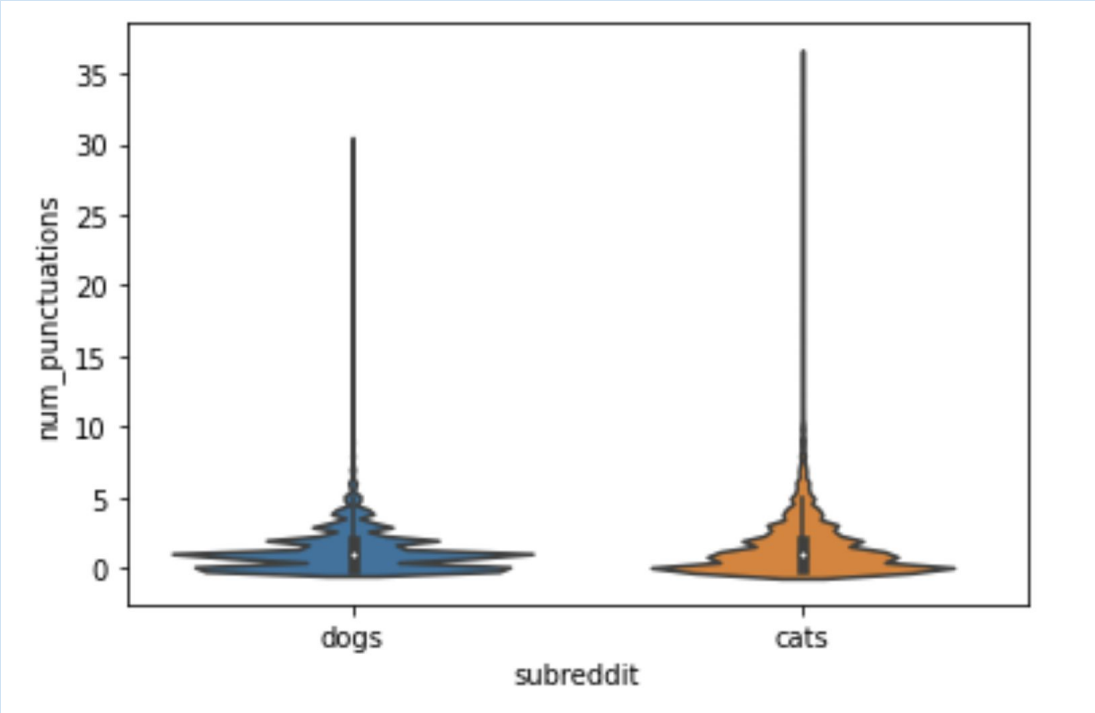

70
60
50
40
30
20
10
0
title_word_count
dogs
cats
subreddit

EDA


num_unique_words
60
50
40
30
20
10
0
dogs
cats
subreddit


num_stopwords
40
30
20
10
0
dogs
cats
subreddit

EDA



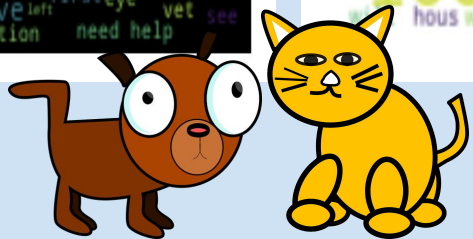
EDA

subreddit	length	title_word_count	num_unique_words	num_stopwords	num_punctuations
cats	53.342003	10.391517	9.895069	4.288086	1.537461
dogs	47.534514	8.927471	8.644558	3.449080	1.273109

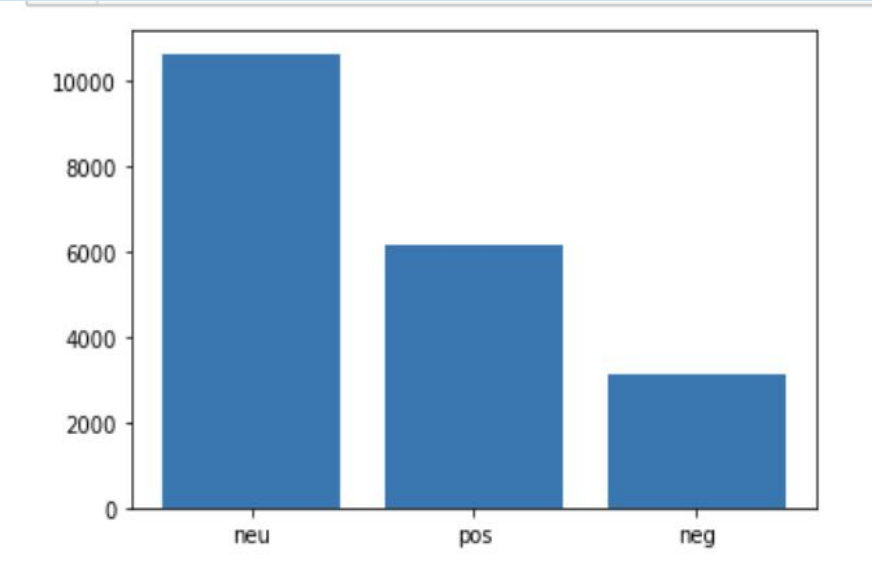
EDA

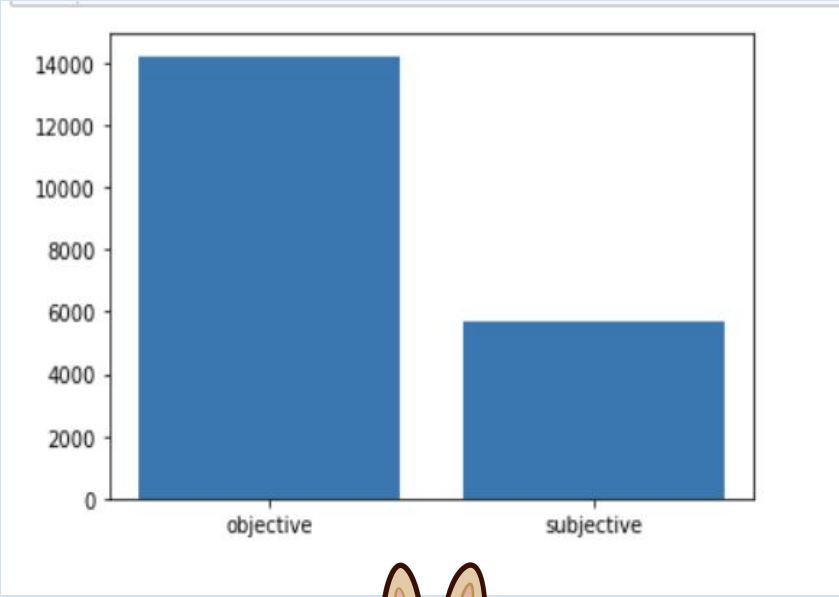
Word Cloud before clean up

Word Cloud after clean up



EDA

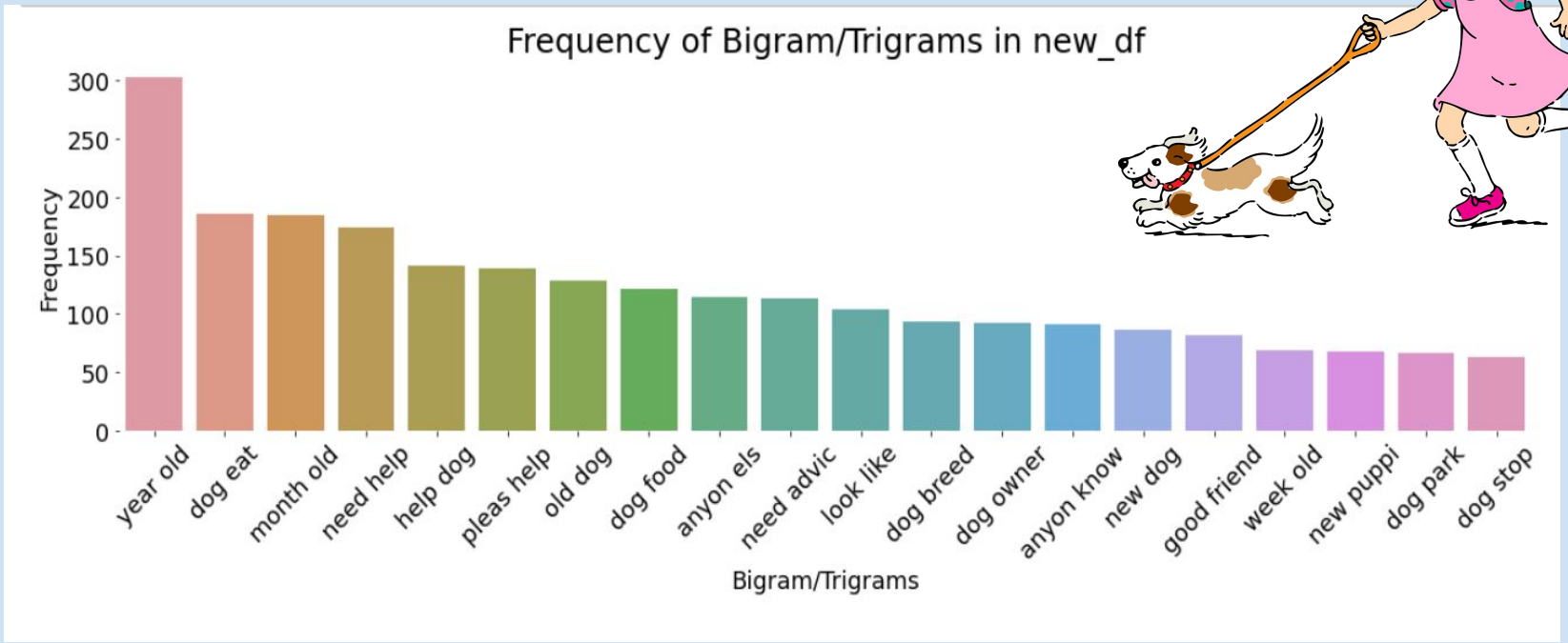

10000
8000
6000
4000
2000
0
neu
pos
neg


14000
12000
10000
8000
6000
4000
2000
0
objective
subjective

EDA

Frequency of Bigram/Trigrams in full_df
Frequency
250
200
150
100
50
0
year old
need help
don know
does dog
month old
help dog
dog food
need advice
dog ate
old dog
new dog
years old
dog breed
dog won
new puppy
months old
best friend
dog park
does know
separation anxiety
Bigram/Trigrams

EDA

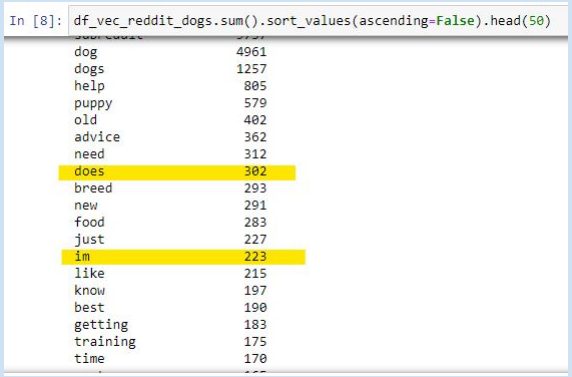

Frequency of Bigram/Trigrams in new_df
Frequency
300
250
200
150
100
50
0
year old
dog eat
month old
need help
help dog
pleas help
old dog
dog food
anyon els
need advic
look like
dog breed
dog owner
anyon know
new dog
good friend
week old
new puppi
dog park
dog stop
Bigram/Trigrams

Pre-processing

- CountVectorizer (stop_words = ‘english’)
- Total columns = **13,472**
- Increase stopwords
- Stemming or Lemmatization

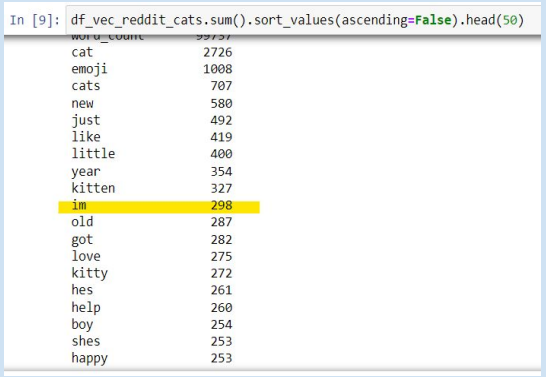
Pre-processing

- Increase Stopwords (e.g. does and im)
- Stemming (reduce 3294 words)
- Lemmatize (reduce 1276 words)



```
In [8]: df_vec_reddit_dogs.sum().sort_values(ascending=False).head(50)
```

dog	4961
dogs	1257
help	805
puppy	579
old	402
advice	362
need	312
does	302
breed	293
new	291
food	283
just	227
im	223
like	215
know	197
best	190
getting	183
training	175
time	170



```
In [9]: df_vec_reddit_cats.sum().sort_values(ascending=False).head(50)
```

cat	2726
emoji	1008
cats	707
new	580
just	492
like	419
little	400
year	354
kitten	327
im	298
old	287
got	282
love	275
kitty	272
hes	261
help	260
boy	254
shes	253
happy	253

Modelling

Model (w cvec)	Train Accuracy	Test Accuracy
Baseline	0.5039	0.5039
Random Forest	0.9964	0.9115
Multinomial Naive Bayes	0.9526	0.9192
Logistic Regression	**0.9617**	**0.9197**

Modelling

MultinomialNB

```
gs2.best.score_: 0.9074151050409771
gs2.best.params_: {'cvec__max_df': 0.35, 'cvec__max_features': 11
000, 'cvec__min_df': 1, 'cvec__ngram_range': (1, 1), 'nb__alpha':
2}
Wall time: 42.5 s

Out[177]: Pipeline(steps=[('cvec',
                           CountVectorizer(max_df=0.35, max_features=11000,
                                           stop_words='english')),
                          ('nb', MultinomialNB(alpha=2))])
```

Logistic Regression

```
gs3.best.score_: 0.9139949127296247
gs3.best.params_: {'cvec__max_df': 0.35, 'cvec__max_features':
11000, 'cvec__min_df': 1, 'cvec__ngram_range': (1, 2), 'lg__C':
1.7575106248547894, 'lg__max_iter': 110}
Wall time: 35min 23s

Out[215]: Pipeline(steps=[('cvec',
                           CountVectorizer(max_df=0.35, max_features=1100
0,
                                           ngram_range=(1, 2), stop_words
='english')),
                          ('lg', LogisticRegression(C=1.7575106248547894,
max_iter=110))])
```

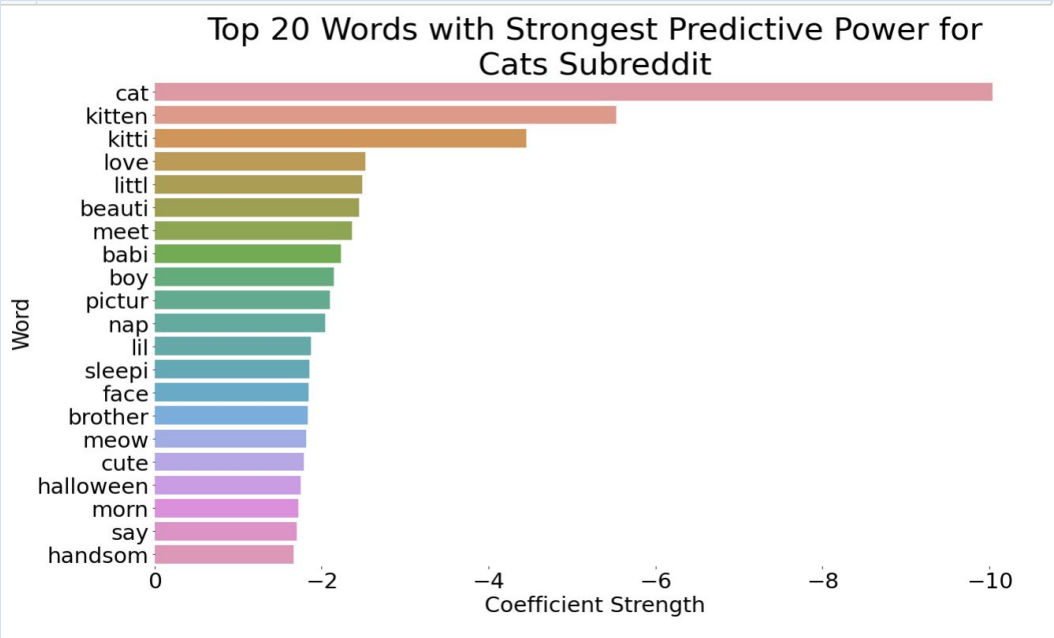
Model (w cvec)	Train Accuracy	Test Accuracy
Multinomial Naive Bayes (GridSearch CV)	0.9486	0.9194
Logistic Regression (GridSearch CV)	0.9725	0.9199

Modelling #2

Why we choose this - logreg - RF vs logreg

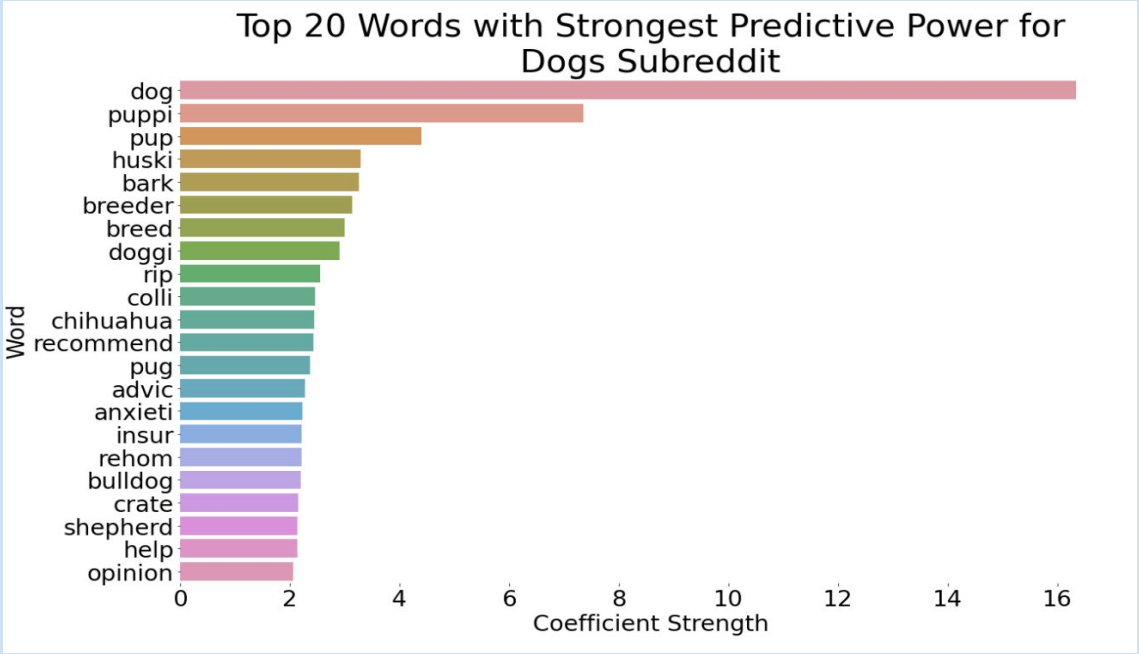
Confusion Matrix

Recommendations: Top Words in r/cats



1) Words such as ‘cute’, ‘sleeping’, ‘little’, ‘kitten’ and ‘beautiful’ reappear more often
2) The r/cats subreddit community may appreciate the aesthetics of cats
3) A possible business venture is to provide grooming services or a cat café which aligns more on these keywords

Recommendations: Top Words in r/dogs



1) Words stemming from dog breeds, ‘advic’ (advice), ‘anxieti’ (anxiety), ‘help’ and ‘opinion’ appear more frequently
2) The r/dogs subreddit community are more specific in asking for advice/help/opinions
3) Topics may seem more targeted towards dog breeds

Recommendations

- r/dogs - more practical and 'discussion' driven

1) Create consultancy and onboarding services for new dog owners
2) If interested to bring in breeds, the top words can signify the popular ones among dog lovers

- r/cats - sharing of digital media

3) Grooming services or venture into F&B (e.g. cat café)

- Usage of Words for Impactful Marketing

4) Unique words for r/dogs < 8 and r/cats < 5

Improvements

1) Include dog, cat, and the variations related to these two words e.g. puppy, kitten in the stopword removal process.
2) Having posts with more unique words helps to distinguish between spam and ham but;
3) Not all posts with unique words are relevant, so we have to focus more on the coefficient of frequently appearing words
4) Explore other forums that are more local e.g. Hardwarezone because Reddit posts are more global and may not attune well to local context