```
#!/usr/bin/python
# -*- coding: utf-8 -*-

# analyze_terms.py

import sys, json
import collections
from string import punctuation
from hcde.utils.stop_words import remove_stops, STOPLIST
custom_stoplist = ['hes', 'shes']

# data filenames
#data_file = "dog_rates_timeline_2018-03-04.json"
#data_output_top = "dog_rates_top_terms.json"
#data_output_bottom = "dog_rates_bottom_terms.json"

data_file = "dog_feelings_timeline_2018-03-04.json"
data_output_top = "dog_feelings_top_terms.json"
data_output_bottom = "dog_feelings_bottom_terms.json"

# clean tweet text
# removes punctuation and stopwords
def clean_text(text):
    # remove capitalization and punctuation
    clean_tweet = ''.join(c for c in text if c not in punctuation)
    clean_tweet = clean_tweet.lower()
    # split into tokens
    return clean_tweet
```

```
# remove stopwords from a list
# inputs: list of tokens
# remove_stopwords(s, stopwords)
def remove_custom_stops(str):
    # first pass with hcde module
    str = remove_stops(str)

    tokens = str.split()
    for word in tokens:
        if word in custom_stoplist or word.isdigit():
            #print word
            tokens.remove(word)
    #print tokens
    return " ".join(tokens)

# ***get most common words***
def get_most_common(list, n):
    counter=collections.Counter(list)
    output=counter.most_common(n)
    top_terms = []
    for item in output:
        term = {
            'text': item[0],
            'count': item[1]
        }
        top_terms.append(term)
        #print item[1]
        #pass
    return top_terms
```

```
def search_word(list, term):
    retVal = []
    for tweet in list:
        splitText = tweet['full_text'].split()
        foundWord = False
        for word in splitText:
            withoutPunc = ''.join(c for c in word if c not in string.punctuation)
            if withoutPunc.lower() == term.lower():
                foundWord = True
            if foundWord:
                retVal.append(tweet)
    return retVal


def search_word(list, term):
    retVal = []
    for tweet in list:
        splitText = tweet['full_text'].split()
        foundWord = False
        for word in splitText:
            withoutPunc = ''.join(c for c in word if c not in string.punctuation)
            if withoutPunc.lower() == term.lower():
                foundWord = True
            if foundWord:
                retVal.append(tweet)
    return retVal
```

```
# read in data
data = []
with open(data_file) as json_file:
    data = json.load(json_file)
# sanity check
#print "hello?"
#print len(data)

# sort data by favorites and split into two lists
#data = sorted(data, key=lambda k: k['retweet_count'], reverse=True)
data = sorted(data, key=lambda k: k['favorite_count'], reverse=True)
mid = len(data)/2
top_half = data[:mid]
bottom_half = data[mid:]

# get a list of just the tweet text and remove
top_half_tokens = []   # all the tokens in all the tweets
for tweet in top_half:
    s1 = clean_text(tweet['full_text'])
    s2 = remove_custom_stops(s1)
    tok = s2.split()
    top_half_tokens += tok

bottom_half_tokens = []   # all the tokens in all the tweets
for tweet in bottom_half:
    s1 = clean_text(tweet['full_text'])
    s2 = remove_custom_stops(s1)
    tok = s2.split()
    bottom_half_tokens += tok
```

```
#print len(tweet_tokens)

top_half_terms = get_most_common(top_half_tokens, 20)
bottom_half_terms = get_most_common(bottom_half_tokens, 20)
#print json.dumps(top_terms, indent=True).encode('utf-8')

# these are our lists!!!!!!!
#print top_terms
#print top_counts

# save this list into a json file
with open(data_output_top, "w") as f:
    json.dump(top_half_terms, f, sort_keys=True, indent=4)

with open(data_output_bottom, "w") as f:
    json.dump(bottom_half_terms, f, sort_keys=True, indent=4)
```