we're going to be compiling feature lists of words from positive reviews and words from the negative reviews to hopefully see trends in specific types of words in positive or negative reviews.

To start, our code:

import nltk

import random

from nltk.corpus import movie\_reviews

documents = [(list(movie\_reviews.words(fileid)), category)

for category in movie\_reviews.categories()

for fileid in movie\_reviews.fileids(category)]

random.shuffle(documents)

all\_words = []

for w in movie\_reviews.words():

all\_words.append(w.lower())

all\_words = nltk.FreqDist(all\_words)

word\_features = list(all\_words.keys())[:3000]

Mostly the same as before, only with now a new variable, word\_features, which contains the top 3,000 most common words. Next, we're going to build a quick function that will find these top 3,000 words in our positive and negative documents, marking their presence as either positive or negative:

def find\_features(document):

words = set(document)

features = {}

for w in word\_features:

features[w] = (w in words)

return features

Next, we can print one feature set like:

print((find\_features(movie\_reviews.words('neg/cv000\_29416.txt'))))

Then we can do this for all of our documents, saving the feature existence booleans and their respective positive or negative categories by doing:

featuresets = [(find\_features(rev), category) for (rev, category) in documents]

Awesome, now that we have our features and labels.