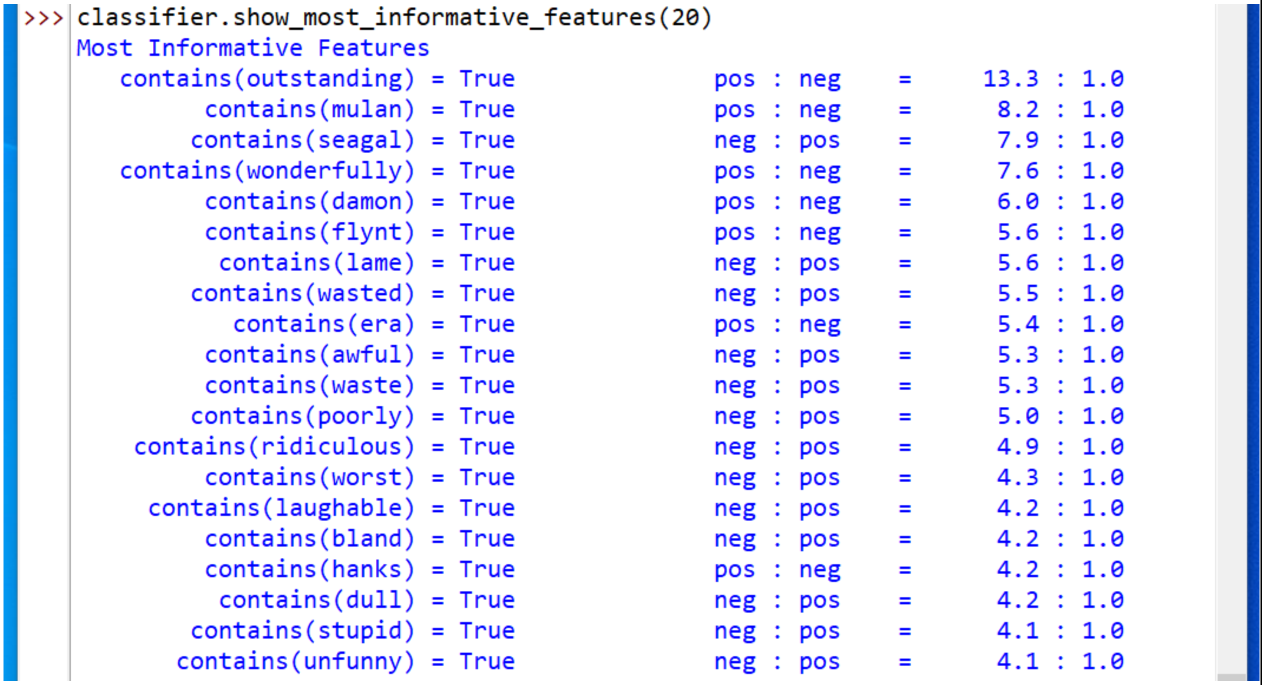
6. Movie review classifier

a.

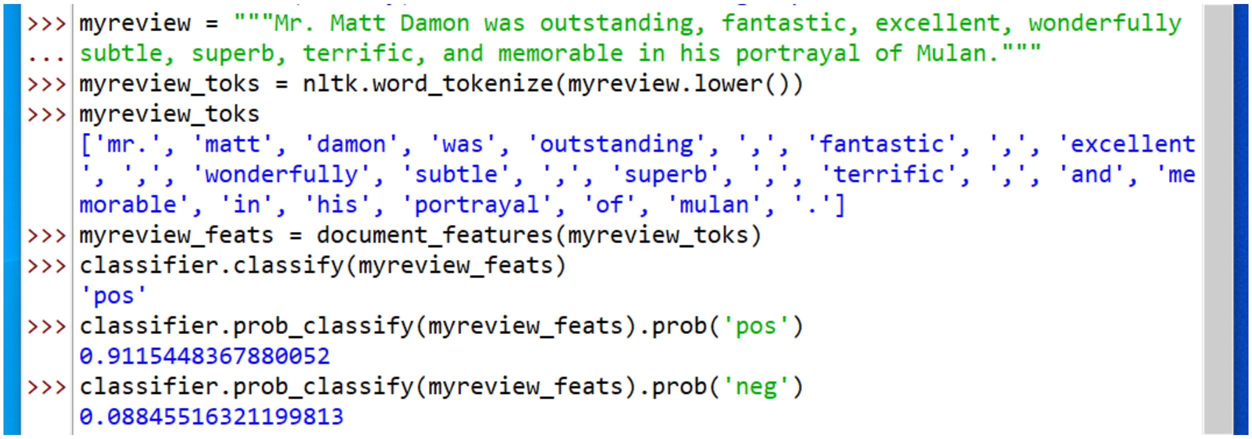


```
>>> classifier.show_most_informative_features(20)
Most Informative Features
   contains(outstanding) = True              pos : neg    =     13.3 : 1.0
         contains(mulan) = True              pos : neg    =      8.2 : 1.0
        contains(seagal) = True              neg : pos    =      7.9 : 1.0
   contains(wonderfully) = True              pos : neg    =      7.6 : 1.0
         contains(damon) = True              pos : neg    =      6.0 : 1.0
         contains(flynt) = True              pos : neg    =      5.6 : 1.0
          contains(lame) = True              neg : pos    =      5.6 : 1.0
        contains(wasted) = True              neg : pos    =      5.5 : 1.0
           contains(era) = True              pos : neg    =      5.4 : 1.0
         contains(awful) = True              neg : pos    =      5.3 : 1.0
         contains(waste) = True              neg : pos    =      5.3 : 1.0
        contains(poorly) = True              neg : pos    =      5.0 : 1.0
    contains(ridiculous) = True              neg : pos    =      4.9 : 1.0
         contains(worst) = True              neg : pos    =      4.3 : 1.0
     contains(laughable) = True              neg : pos    =      4.2 : 1.0
         contains(bland) = True              neg : pos    =      4.2 : 1.0
         contains(hanks) = True              pos : neg    =      4.2 : 1.0
          contains(dull) = True              neg : pos    =      4.2 : 1.0
        contains(stupid) = True              neg : pos    =      4.1 : 1.0
       contains(unfunny) = True              neg : pos    =      4.1 : 1.0
```

“outstanding” is a highly informative feature: it is a strong indicator of the “positive” label, with a “pos” to “neg” ratio of 13.3:1. This means the word is 13 times more likely to occur in a positive movie review than in a negative one.

“seagal” on the other hand is a highly negative feature. Its 7.9:1 “neg”-to-“pos” ratio means it occurs 8 times as many negative reviews as opposed to positive ones.

b.



```
>>> myreview = """Mr. Matt Damon was outstanding, fantastic, excellent, wonderfully
... subtle, superb, terrific, and memorable in his portrayal of Mulan."""
>>> myreview_toks = nltk.word_tokenize(myreview.lower())
>>> myreview_toks
['mr.', 'matt', 'damon', 'was', 'outstanding', ',', 'fantastic', ',', 'excellent
', ',', 'wonderfully', 'subtle', ',', 'superb', ',', 'terrific', ',', 'and', 'me
morable', 'in', 'his', 'portrayal', 'of', 'mulan', '.']
>>> myreview_feats = document_features(myreview_toks)
>>> classifier.classify(myreview_feats)
'pos'
>>> classifier.prob_classify(myreview_feats).prob('pos')
0.9115448367880052
>>> classifier.prob_classify(myreview_feats).prob('neg')
0.08845516321199813
```

My short review contains Matt Damon and a whole lot of positive sounding words. As expected, it was classified as 'positive' with a high, 91% probability.

7. Base probabilities (=priors)

a.	Code/output (copy-paste) Write-up
b.	Code/output (copy-paste) Write-up