Jason Ballantyne   Text Analytics COMP47600   14/10/2021
13432788   Practical Number 4

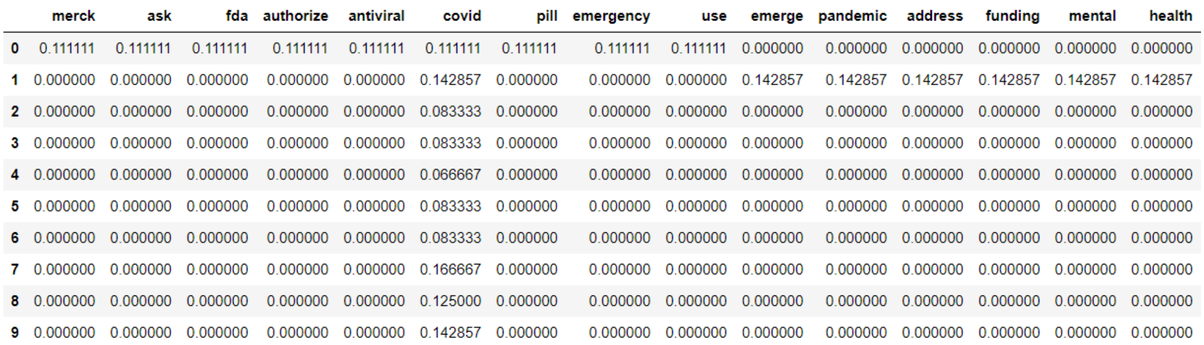
Q1:

The 10 short text items I've chosen are 10 popular tweets that are related under the topic of "Covid".

a) The output after the standard stop-words have been removed is:

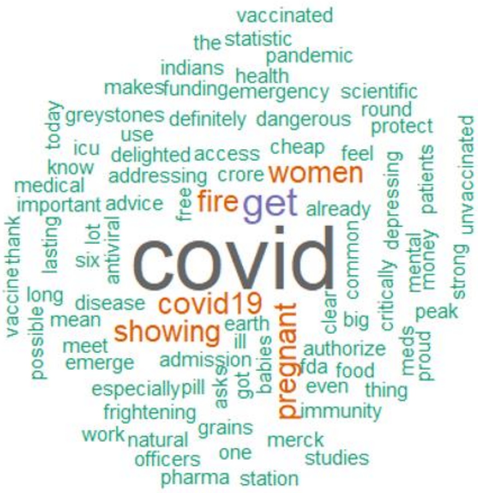
['merck', 'ask', 'fda', 'authorize', 'antiviral', 'covid', 'pill', 'emergency', 'use', 'emerge', 'covid', 'pandemic', 'address', 'funding', 'mental', 'health', 'delighted', 'meet', 'fire', 'officer', 'greystones', 'fire', 'station', 'today', 'thank', 'work', 'especially', 'covid', 'big', 'pharma', 'make', 'lot', 'money', 'med', 'get', 'icu', 'admission', 'cheap', 'covid', 'vaccine', 'scientific', 'study', 'show', 'earth', 'round', 'oh', 'mean', 'show', 'strong', 'long', 'lasting', 'natural', 'immunity', 'covid', 'disease', 'one', 'six', 'critically', 'ill', 'covid', 'patient', 'unvaccinated', 'pregnant', 'woman', 'depress', 'frighten', 'statistic', 'feel', 'proud', 'even', 'peak', 'covid', 'crore', 'indian', 'get', 'access', 'free', 'food', 'grain', 'important', 'thing', 'protect', 'covid', 'get', 'vaccinate', 'covid', 'dangerous', 'pregnant', 'woman', 'baby', 'medical', 'advice', 'clear', 'already', 'covid', 'definitely', 'possible', 'get', 'know', 'common']

b) Please see a snippet of the matrix below.

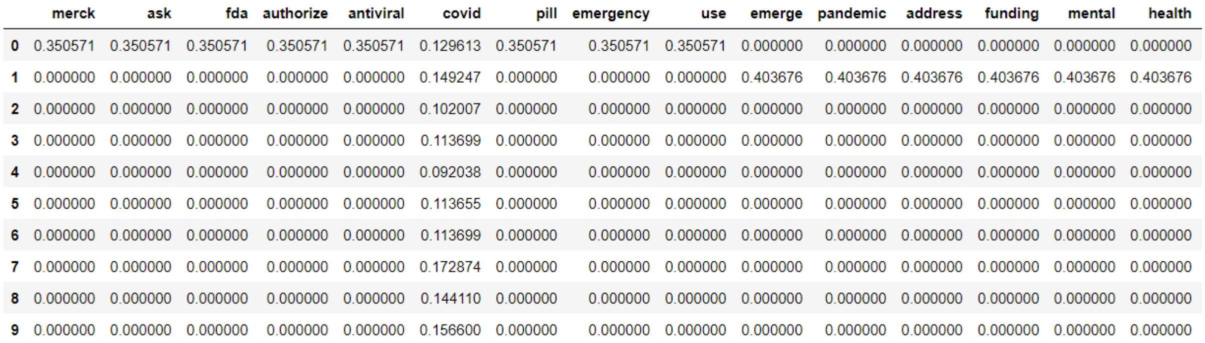


	merck	ask	fda	authorize	antiviral	covid	pill	emergency	use	emerge	pandemic	address	funding	mental	health
0	0.111111	0.111111	0.111111	0.111111	0.111111	0.111111	0.111111	0.111111	0.111111	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
1	0.000000	0.000000	0.000000	0.000000	0.000000	0.142857	0.000000	0.000000	0.000000	0.142857	0.142857	0.142857	0.142857	0.142857	0.142857
2	0.000000	0.000000	0.000000	0.000000	0.000000	0.083333	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
3	0.000000	0.000000	0.000000	0.000000	0.000000	0.083333	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
4	0.000000	0.000000	0.000000	0.000000	0.000000	0.066667	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
5	0.000000	0.000000	0.000000	0.000000	0.000000	0.083333	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
6	0.000000	0.000000	0.000000	0.000000	0.000000	0.083333	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
7	0.000000	0.000000	0.000000	0.000000	0.000000	0.166667	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
8	0.000000	0.000000	0.000000	0.000000	0.000000	0.125000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
9	0.000000	0.000000	0.000000	0.000000	0.000000	0.142857	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

Please see the corresponding word cloud image below.



c) Please see a snippet of the matrix for the TF-IDF scores



	merck	ask	fda	authorize	antiviral	covid	pill	emergency	use	emerge	pandemic	address	funding	mental	health
0	0.350571	0.350571	0.350571	0.350571	0.350571	0.129613	0.350571	0.350571	0.350571	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
1	0.000000	0.000000	0.000000	0.000000	0.000000	0.149247	0.000000	0.000000	0.000000	0.403676	0.403676	0.403676	0.403676	0.403676	0.403676
2	0.000000	0.000000	0.000000	0.000000	0.000000	0.102007	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
3	0.000000	0.000000	0.000000	0.000000	0.000000	0.113699	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
4	0.000000	0.000000	0.000000	0.000000	0.000000	0.092038	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
5	0.000000	0.000000	0.000000	0.000000	0.000000	0.113655	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
6	0.000000	0.000000	0.000000	0.000000	0.000000	0.113699	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
7	0.000000	0.000000	0.000000	0.000000	0.000000	0.172874	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
8	0.000000	0.000000	0.000000	0.000000	0.000000	0.144110	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
9	0.000000	0.000000	0.000000	0.000000	0.000000	0.156600	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

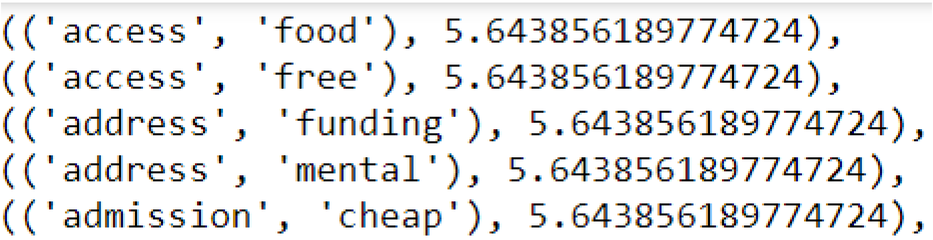
d) If we look at the word “covid” it is boosted in TF-IDF in each of the 10 documents. Similarly, the word “emerge” is boosted from “0.142857” in TF to “0.403676” in TF-IDF. The last example we will look at it is the word “vaccinate” which is boosted from “0.166667” in TF to “0.46758” in TF-IDF.
These changes occur likely due to the IDF not being calculated from a large enough corpus. TF-IDF is seen as being a better/accurate method, independent of classifier. Using only TF we don't really care if a word is common or not. Therefore, common words like e.g. “ask” in our case receive a large weight even if they contribute no real information. Whereas in TF-IDF the more frequent a word is in the corpus, the smaller weight it receives. Thus, common words like “ask” receive small weights but rare words, that it is assumed to carry more information, receive larger weights. The difference in these scores between TF and TF-IDF being whether the corpus-frequencies of words are used or not.

Q2:

PMI – Pointwise mutual information scores ngrams by pointwise mutual information. It is a measure of association between a feature (in our case a word) and a class (category), not between a document (tweet) and a category. The formula can be seen below:

$$PMI(a,b) = \log(\frac{P(a,b)}{\mathrm{P(a)P(b)}})$$

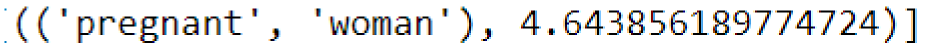
The top-5 pairs based on the PMI scores found in each pair can be seen below.



```
(('access', 'food'), 5.643856189774724),
(('access', 'free'), 5.643856189774724),
(('address', 'funding'), 5.643856189774724),
(('address', 'mental'), 5.643856189774724),
(('admission', 'cheap'), 5.643856189774724),
```

These results do not make a great deal of sense. Nearly all of the scores have been calculated with a PMI of 5.64. The problem that occurred here is a known problem with PMI where it over-estimates the pointwise information that are next to these words even when their frequencies aren't great.

To combat these poor results, we will introduce a minimal cut-off frequency and re-compute the top performing PMI scores. To do this, I have set the "apply_freq_filter" to 2. This means that only bigrams that have a frequency >=2 will be considered in the PMI score calculation process. After the "apply_freq_filter" has been set, please see the result below.



```
(('pregnant', 'woman'), 4.643856189774724)]
```

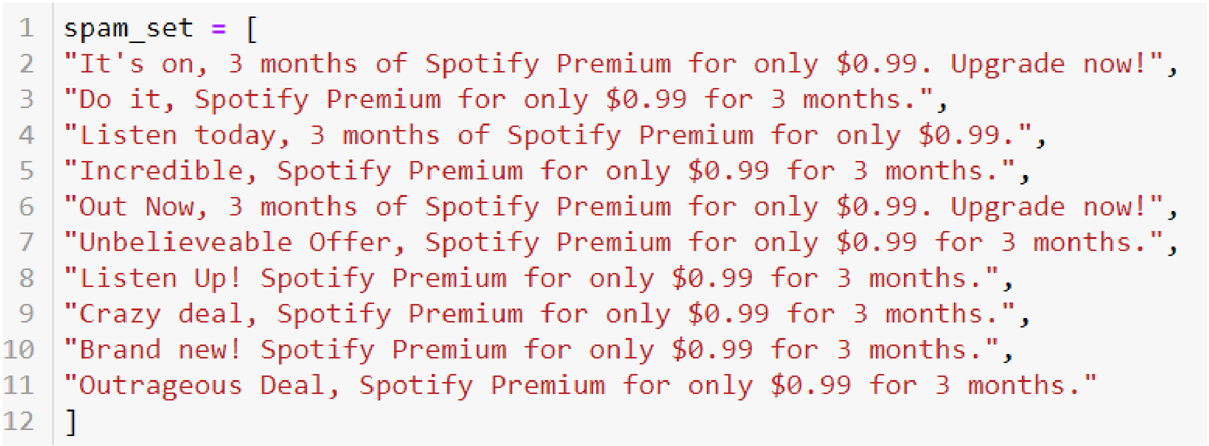
We can see the result above looks to make more sense than our initial results. Although our text items all deal the same toxic of interest which is "covid". Due to the small number of text-items we introduced (10), there are not many frequencies >=2. If we had a larger corpus to work with, we would get more results by setting the "apply_freq_filter" to 2. However, this example this demonstrates how adjusting the "apply_freq_filter" can create more sensible results and prevent the over-estimating we have seen in our initial set of results.

Q3:

Entropy – Entropy is the sum of the probability of each label times the log probability of that same label. It is used as a way to measure how "mixed" a column is. Specifically, entropy is used to measure disorder. In other words, entropy is a measure of uncertainty. The formula for entropy is:

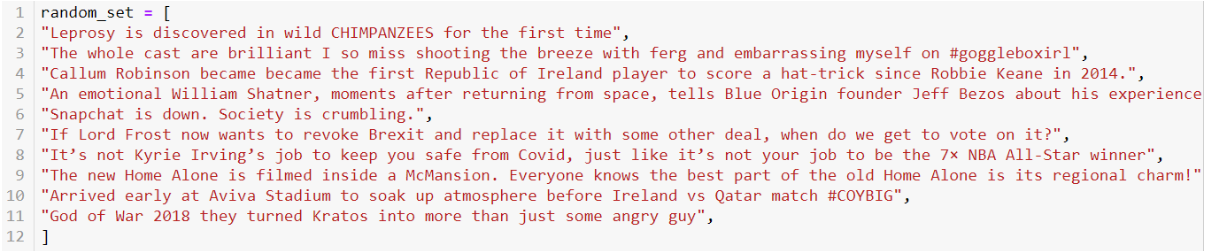
$$-\sum_{i=1}^{C} P(x_i)log_b P(x_i)$$

a) The spam set of tweets is as follows:



```
spam_set = [
"It's on, 3 months of Spotify Premium for only $0.99. Upgrade now!",
"Do it, Spotify Premium for only $0.99 for 3 months.",
"Listen today, 3 months of Spotify Premium for only $0.99.",
"Incredible, Spotify Premium for only $0.99 for 3 months.",
"Out Now, 3 months of Spotify Premium for only $0.99. Upgrade now!",
"Unbelieveable Offer, Spotify Premium for only $0.99 for 3 months.",
"Listen Up! Spotify Premium for only $0.99 for 3 months.",
"Crazy deal, Spotify Premium for only $0.99 for 3 months.",
"Brand new! Spotify Premium for only $0.99 for 3 months.",
"Outrageous Deal, Spotify Premium for only $0.99 for 3 months."
]
```

b) The random set of tweets is as follows:



```
random_set = [
"Leprosy is discovered in wild CHIMPANZEES for the first time",
"The whole cast are brilliant I so miss shooting the breeze with ferg and embarrassing myself on #goggleboxirl",
"Callum Robinson became became the first Republic of Ireland player to score a hat-trick since Robbie Keane in 2014.",
"An emotional William Shatner, moments after returning from space, tells Blue Origin founder Jeff Bezos about his experience
"Snapchat is down. Society is crumbling.",
"If Lord Frost now wants to revoke Brexit and replace it with some other deal, when do we get to vote on it?",
"It’s not Kyrie Irving’s job to keep you safe from Covid, just like it’s not your job to be the 7× NBA All-Star winner",
"The new Home Alone is filmed inside a McMansion. Everyone knows the best part of the old Home Alone is its regional charm!"
"Arrived early at Aviva Stadium to soak up atmosphere before Ireland vs Qatar match #COYBIG",
"God of War 2018 they turned Kratos into more than just some angry guy",
]
```

The entropy values found for the above sets of tweets are as follows:

i.

Spam-Set	4.430285832299606

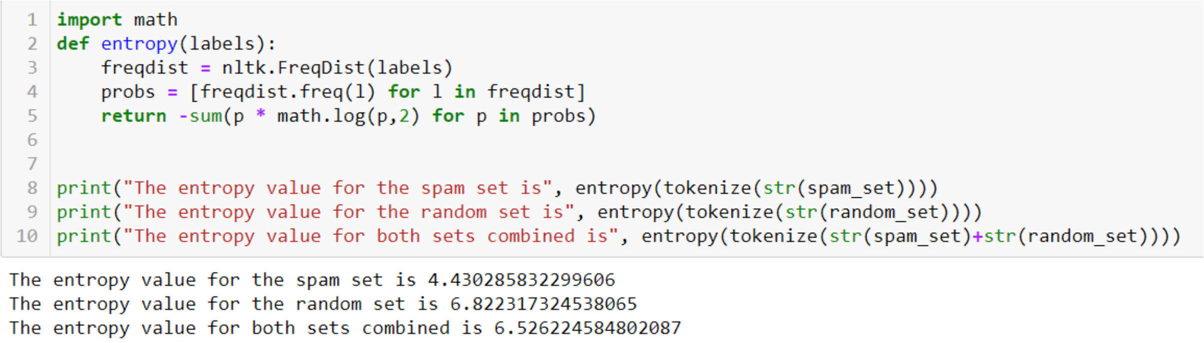
ii.

Random-Set	6.822317324538065

iii.

The two sets combined	6.526224584802087

The program used to calculate the entropy is as follows:



```
import math
def entropy(labels):
    freqdist = nltk.FreqDist(labels)
    probs = [freqdist.freq(l) for l in freqdist]
    return -sum(p * math.log(p,2) for p in probs)


print("The entropy value for the spam set is", entropy(tokenize(str(spam_set))))
print("The entropy value for the random set is", entropy(tokenize(str(random_set))))
print("The entropy value for both sets combined is", entropy(tokenize(str(spam_set)+str(random_set))))
```

```
The entropy value for the spam set is 4.430285832299606
The entropy value for the random set is 6.822317324538065
The entropy value for both sets combined is 6.526224584802087
```

The source for the code is: https://www.nltk.org/book/ch06#fig-entropy

The entropy results obtained show a clearly a lower entropy figure for the spam set than for the random set. This highlights how the random set is varying. High entropy relates to disorder which is visibly present in our random set. The low entropy figure present in the spam set indicates that the set is less ordered.