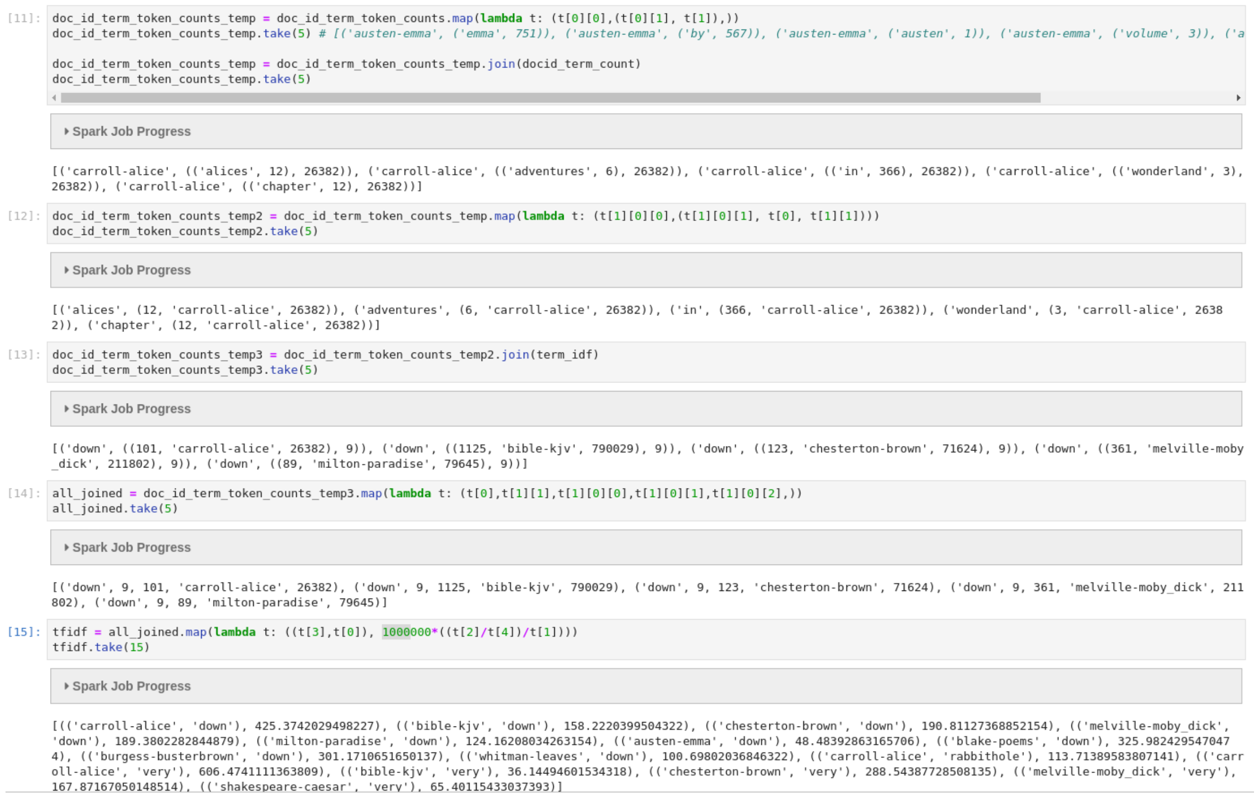
1. Harsha Thulasi
2. I spent between 6-8 hours on this one. Lot of it was searching spark related commands and re-watching some of your videos. I made one mistake of converting all the data into dataframes and trying to make that work. As I was about to submit I re-read your lab6.txt and realized I need to submit with RDDs instead of DFs. Beyond that, my notebook/cluster kept freezing many times. I had to start/stop the notebook and cluster (tried doing it individually a few times) to get the spark env to work consistently.
3. Trying to work in AWs hindered me a bit. Beyond that I wasn't sure if my solution was correct. Looking at the formula we used for lab5 it's a bit different. In lab5, I got the sum of the tfidf of each word per docid and divided it by the number of words in string (more on it below). Trying to understand it took around 2-3 hours. The step to make the tfidf table (i think cell 55 but not sure if it updated), is combined of all the joins and maps. Here's a broken down version of it:



```
[11]: doc_id_term_token_counts_temp = doc_id_term_token_counts.map(lambda t: (t[0][0],(t[0][1], t[1]),))
      doc_id_term_token_counts_temp.take(5) # [('austen-emma', ('emma', 751)), ('austen-emma', ('by', 567)), ('austen-emma', ('austen', 1)), ('austen-emma', ('volume', 3)), ('a

      doc_id_term_token_counts_temp = doc_id_term_token_counts_temp.join(docid_term_count)
      doc_id_term_token_counts_temp.take(5)
```

▸ Spark Job Progress

```
[('carroll-alice', (('alices', 12), 26382)), ('carroll-alice', (('adventures', 6), 26382)), ('carroll-alice', (('in', 366), 26382)), ('carroll-alice', (('wonderland', 3),
26382)), ('carroll-alice', (('chapter', 12), 26382))]
```

```
[12]: doc_id_term_token_counts_temp2 = doc_id_term_token_counts_temp.map(lambda t: (t[1][0][0],(t[1][0][1], t[0], t[1][1])))
      doc_id_term_token_counts_temp2.take(5)
```

▸ Spark Job Progress

```
[('alices', (12, 'carroll-alice', 26382)), ('adventures', (6, 'carroll-alice', 26382)), ('in', (366, 'carroll-alice', 26382)), ('wonderland', (3, 'carroll-alice', 2638
2)), ('chapter', (12, 'carroll-alice', 26382))]
```

```
[13]: doc_id_term_token_counts_temp3 = doc_id_term_token_counts_temp2.join(term_idf)
      doc_id_term_token_counts_temp3.take(5)
```

▸ Spark Job Progress

```
[('down', ((101, 'carroll-alice', 26382), 9)), ('down', ((1125, 'bible-kjv', 790029), 9)), ('down', ((123, 'chesterton-brown', 71624), 9)), ('down', ((361, 'melville-moby
_dick', 211802), 9)), ('down', ((89, 'milton-paradise', 79645), 9))]
```

```
[14]: all_joined = doc_id_term_token_counts_temp3.map(lambda t: (t[0],t[1][1],t[1][0][0],t[1][0][1],t[1][0][2],))
      all_joined.take(5)
```

▸ Spark Job Progress

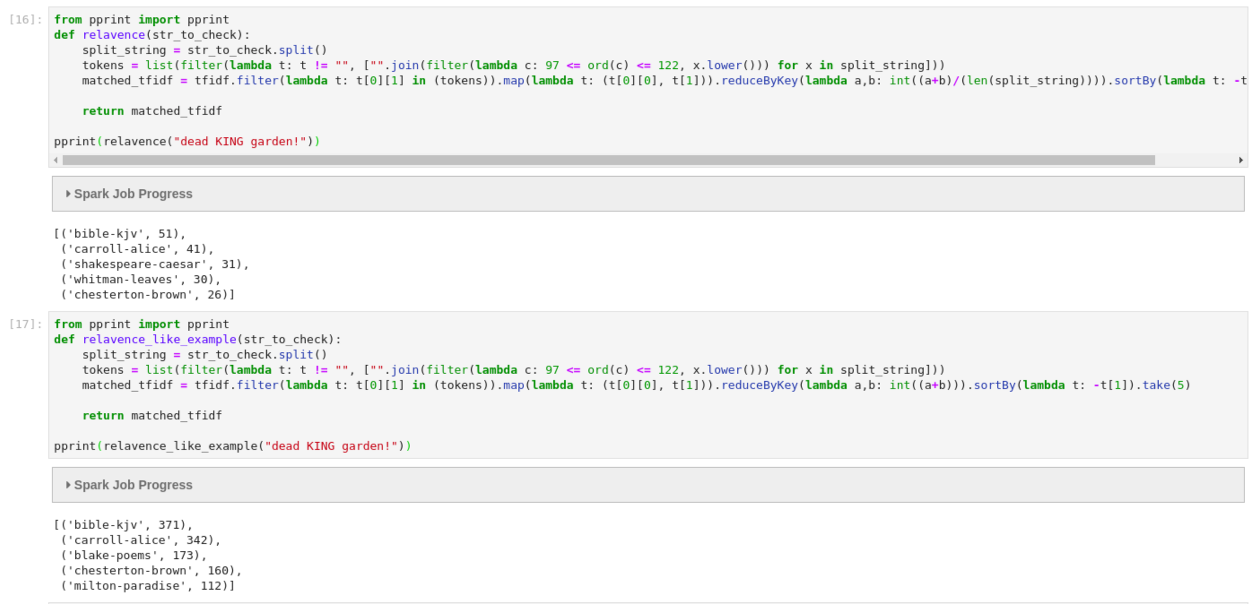
```
[('down', 9, 101, 'carroll-alice', 26382), ('down', 9, 1125, 'bible-kjv', 790029), ('down', 9, 123, 'chesterton-brown', 71624), ('down', 9, 361, 'melville-moby_dick', 211
802), ('down', 9, 89, 'milton-paradise', 79645)]
```

```
[15]: tfidf = all_joined.map(lambda t: ((t[3],t[0]), 1000000*((t[2]/t[4])/t[1])))
      tfidf.take(15)
```

▸ Spark Job Progress

```
[(('carroll-alice', 'down'), 425.3742029498227), (('bible-kjv', 'down'), 158.2220399504322), (('chesterton-brown', 'down'), 190.81127368852154), (('melville-moby_dick',
'down'), 189.3802282844879), (('milton-paradise', 'down'), 124.16208034263154), (('austen-emma', 'down'), 48.48392863165706), (('blake-poems', 'down'), 325.982429547047
4), (('burgess-busterbrown', 'down'), 301.1710651650137), (('whitman-leaves', 'down'), 100.69802036846322), (('carroll-alice', 'rabbithole'), 113.71389583807141), (('carr
oll-alice', 'very'), 606.4741111363809), (('bible-kjv', 'very'), 36.14494601534318), (('chesterton-brown', 'very'), 288.54387728508135), (('melville-moby_dick', 'very'),
167.87167050148514), (('shakespeare-caesar', 'very'), 65.40115433037393)]
```

4. Yes, I tried the extra credit. I am not sure if it is correct as I tried with the formula from lab5 and the results didn't quite match the example:



```
[16]:
from pprint import pprint
def relavence(str_to_check):
    split_string = str_to_check.split()
    tokens = list(filter(lambda t: t != "", ["".join(filter(lambda c: 97 <= ord(c) <= 122, x.lower())) for x in split_string]))
    matched_tfidf = tfidf.filter(lambda t: t[0][1] in (tokens)).map(lambda t: (t[0][0], t[1])).reduceByKey(lambda a,b: int((a+b)/(len(split_string)))).sortBy(lambda t: -t

    return matched_tfidf

pprint(relavence("dead KING garden!"))
```

▸ Spark Job Progress

```
[('bible-kjv', 51),
 ('carroll-alice', 41),
 ('shakespeare-caesar', 31),
 ('whitman-leaves', 30),
 ('chesterton-brown', 26)]
```

```
[17]:
from pprint import pprint
def relavence_like_example(str_to_check):
    split_string = str_to_check.split()
    tokens = list(filter(lambda t: t != "", ["".join(filter(lambda c: 97 <= ord(c) <= 122, x.lower())) for x in split_string]))
    matched_tfidf = tfidf.filter(lambda t: t[0][1] in (tokens)).map(lambda t: (t[0][0], t[1])).reduceByKey(lambda a,b: int((a+b))).sortBy(lambda t: -t[1]).take(5)

    return matched_tfidf

pprint(relavence_like_example("dead KING garden!"))
```

▸ Spark Job Progress

```
[('bible-kjv', 371),
 ('carroll-alice', 342),
 ('blake-poems', 173),
 ('chesterton-brown', 160),
 ('milton-paradise', 112)]
```

The first one is following the formula and second is trying without but getting close to the values. I believe step15 needs the values to be multiplied by 100000 instead of 1000000 to get the values like the example.