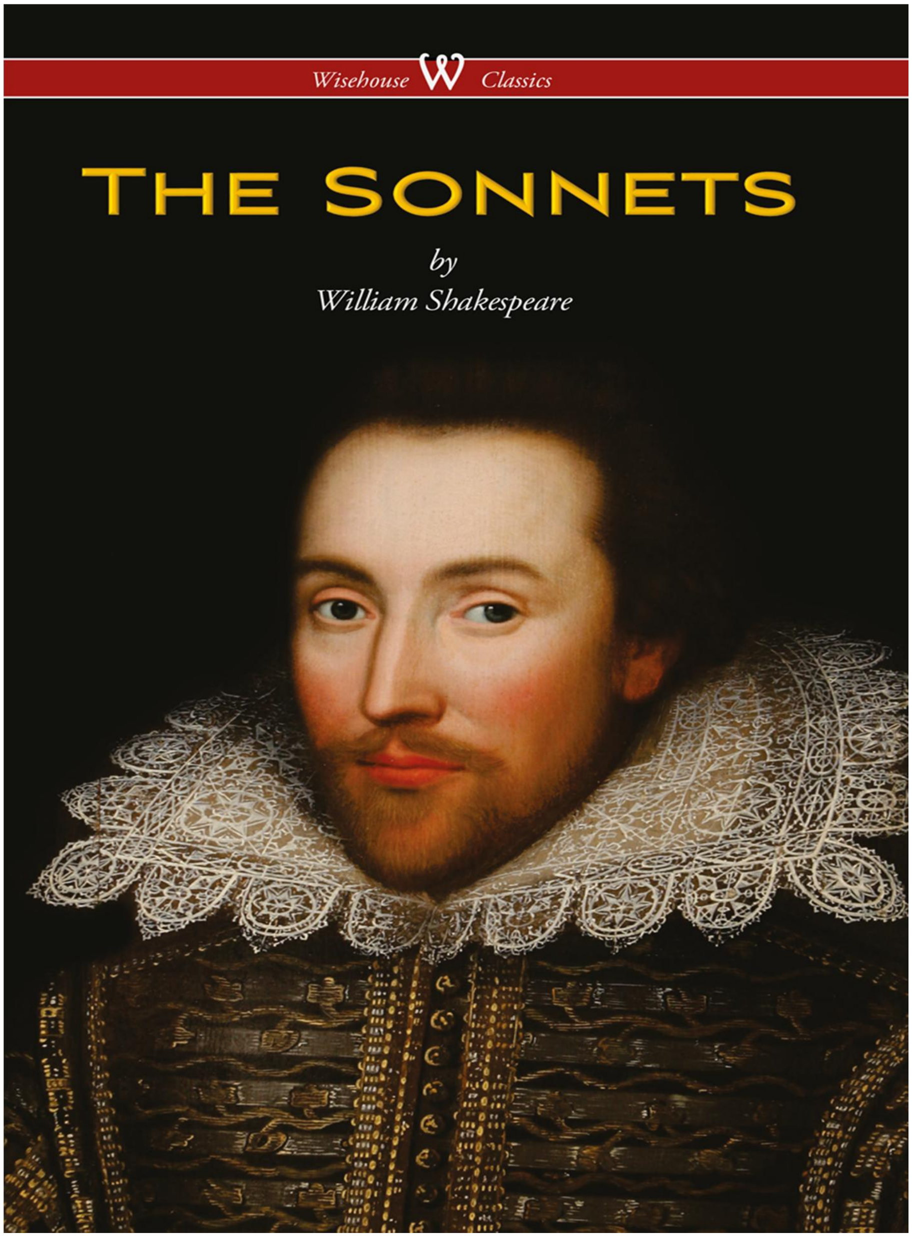
BOOK RECOMMENDATION MODEL USING PYSPARK



ASSIGNMENT IN ADVANCE BIGDATA TECHNOLOGIES

Megha M | Post Graduate Diploma in Data Science, MAHE Dubai |Roll no: 190114015

Objective: The main objective is to build a book recommendation model to which age group a particular book is suitable using pyspark

i) First count all words in the book.
ii) Create a separate list manually of in-appropriate words (foul words, explicit content, violent words)
iii) Then predict whether this book is suitable for children or its adults only.

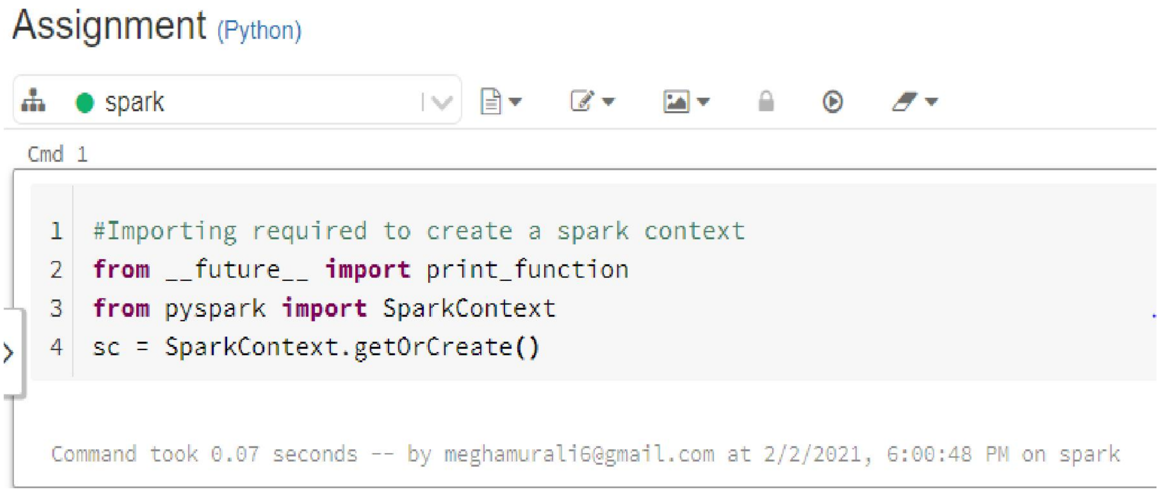
We have to build a book recommendation model using pyspark by the following steps:

1. Create cluster in data bricks and attach to the pyspark notebook through the detach option. Once the cluster dot is in green color we can start to code

2. Creating the Spark Context

Codes:

```
from __future__ import print_function
from pyspark import SparkContext
sc = SparkContext.getOrCreate()
```

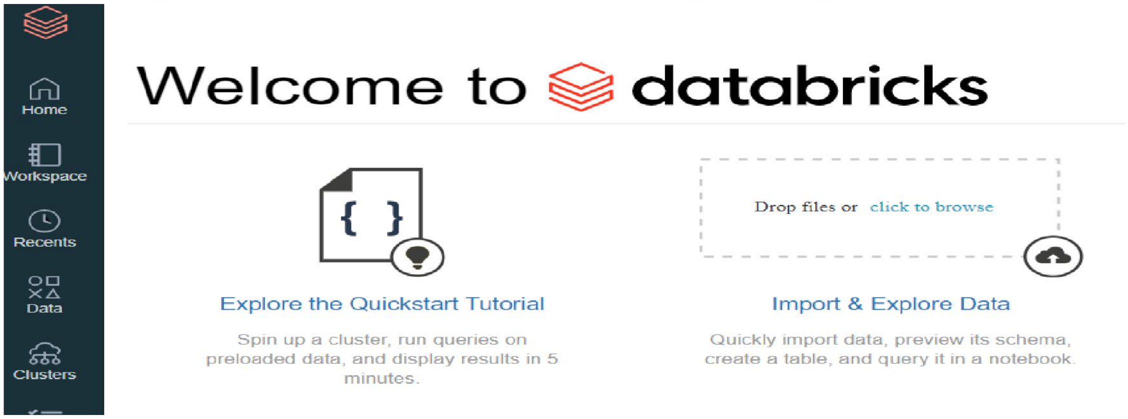


Assignment (Python)

```
#Importing required to create a spark context
from __future__ import print_function
from pyspark import SparkContext
sc = SparkContext.getOrCreate()
```

```
Command took 0.07 seconds -- by meghamurali6@gmail.com at 2/2/2021, 6:00:48 PM on spark
```

3. Download text file from Google and convert it to text file.In databricks there is a option to load the file to the cluster (Import & Export Data)



Welcome to databricks

Explore the Quickstart Tutorial

Spin up a cluster, run queries on preloaded data, and display results in 5 minutes.

Import & Explore Data

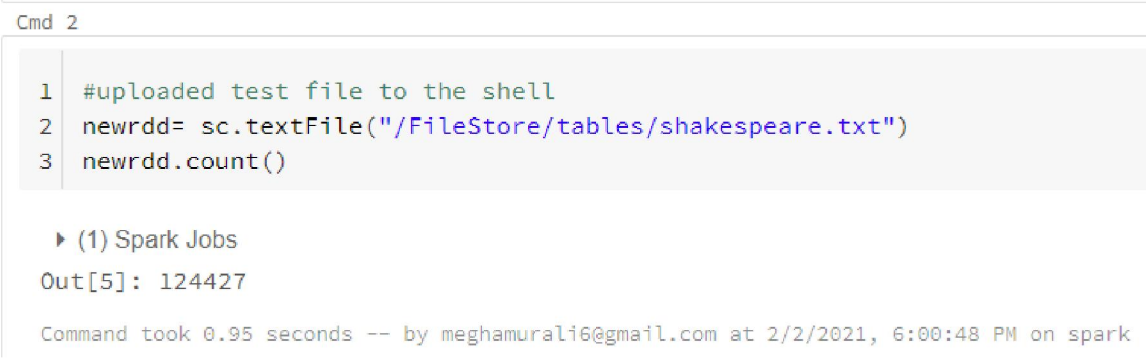
Quickly import data, preview its schema, create a table, and query it in a notebook.

4. Uploading test file to the spark shell

Codes:

newrdd= sc.textFile("/FileStore/tables/shakespeare.txt")
newrdd.count()

Screen shot:



```
#uploaded test file to the shell
newrdd= sc.textFile("/FileStore/tables/shakespeare.txt")
newrdd.count()
```

```
Out[5]: 124427

Command took 0.95 seconds -- by meghamurali6@gmail.com at 2/2/2021, 6:00:48 PM on spark
```

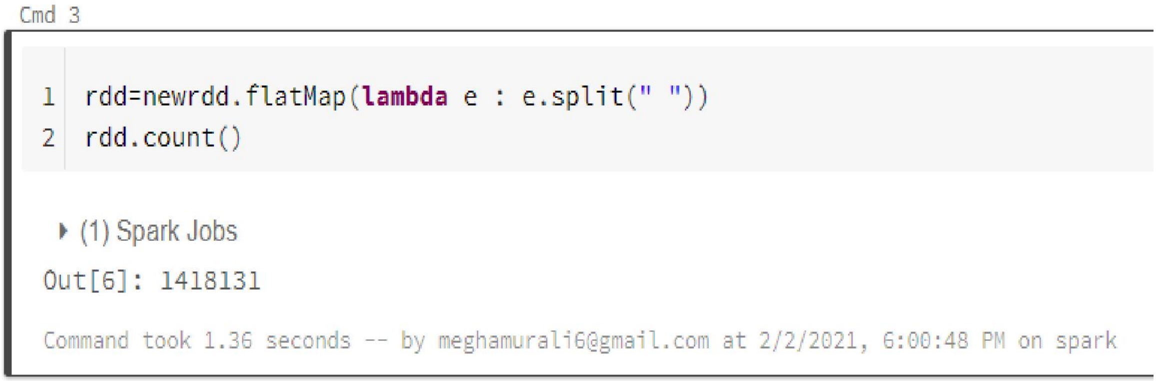
5. Converting all lines into list of words & counting the number of words present in the entire text file.

Codes :

rdd=newrdd.flatMap(lambda e : e.split(" "))

rdd.count()

Screenshot:



```
rdd=newrdd.flatMap(lambda e : e.split(" "))
rdd.count()
```

```
Out[6]: 1418131

Command took 1.36 seconds -- by meghamurali6@gmail.com at 2/2/2021, 6:00:48 PM on spark
```

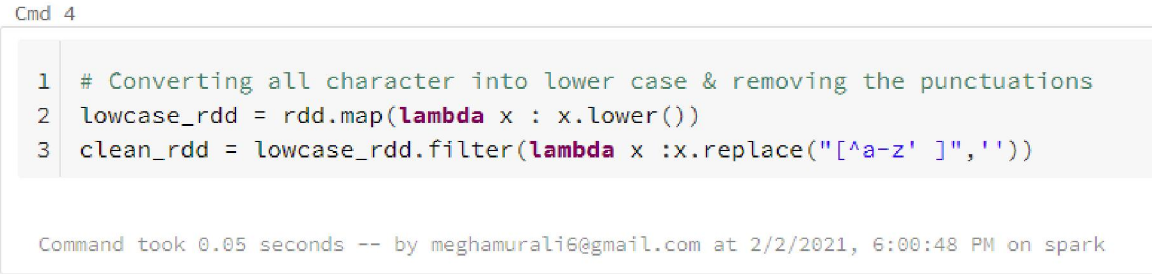
There are 1418131 words in this book

6. Converting all character into lower case and removing punctuation.

Codes:

```
lowcase_rdd = rdd.map(lambda x : x.lower())

clean_rdd = lowcase_rdd.filter(lambda x :x.replace("[^a-z' ]",''))
```

Screenshot:



Cmd 4

```
# Converting all character into lower case & removing the punctuations
lowcase_rdd = rdd.map(lambda x : x.lower())
clean_rdd = lowcase_rdd.filter(lambda x :x.replace("[^a-z' ]",''))
```

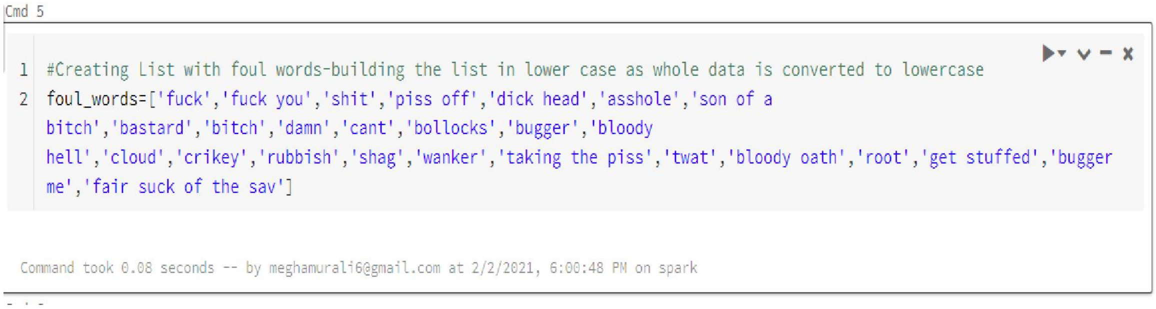
Command took 0.05 seconds -- by meghamurali6@gmail.com at 2/2/2021, 6:00:48 PM on spark

7. Creating List with foul words-building the list in lower case as whole data is converted to lowercase

Codes:

foul_words=['fuck','fuck you','shit','piss off','dick head','asshole','son of a bitch','bastard','bitch','damn','cant','bollocks','bugger','bloody hell','cloud','crikey','rubbish','shag','wanker','taking the piss','twat','bloody oath','root','get stuffed','bugger me','fair suck of the sav']

Screenshot:



Cmd 5

```
#Creating List with foul words-building the list in lower case as whole data is converted to lowercase
foul_words=['fuck','fuck you','shit','piss off','dick head','asshole','son of a
bitch','bastard','bitch','damn','cant','bollocks','bugger','bloody
hell','cloud','crikey','rubbish','shag','wanker','taking the piss','twat','bloody oath','root','get stuffed','bugger
me','fair suck of the sav']
```

Command took 0.08 seconds -- by meghamurali6@gmail.com at 2/2/2021, 6:00:48 PM on spark

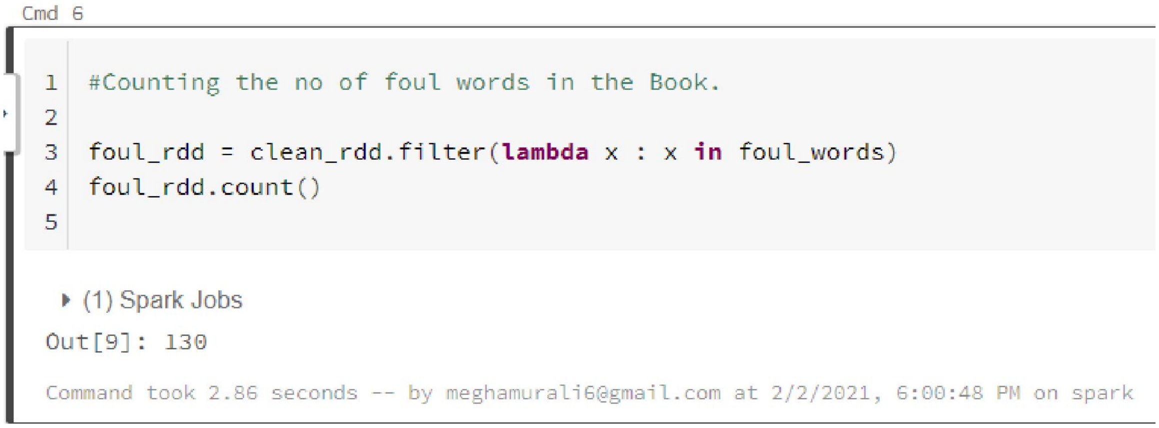
8. Counting the no of foul words in the Book.

Codes:

foul_rdd = clean_rdd.filter(lambda x : x in foul_words)

foul_rdd.count()

Screenshot:



```
Cmd 6

#Counting the no of foul words in the Book.

foul_rdd = clean_rdd.filter(lambda x : x in foul_words)
foul_rdd.count()

▸ (1) Spark Jobs
Out[9]: 130

Command took 2.86 seconds -- by meghamurali6@gmail.com at 2/2/2021, 6:00:48 PM on spark
```

9. Counting the frequency of each Foul Words.

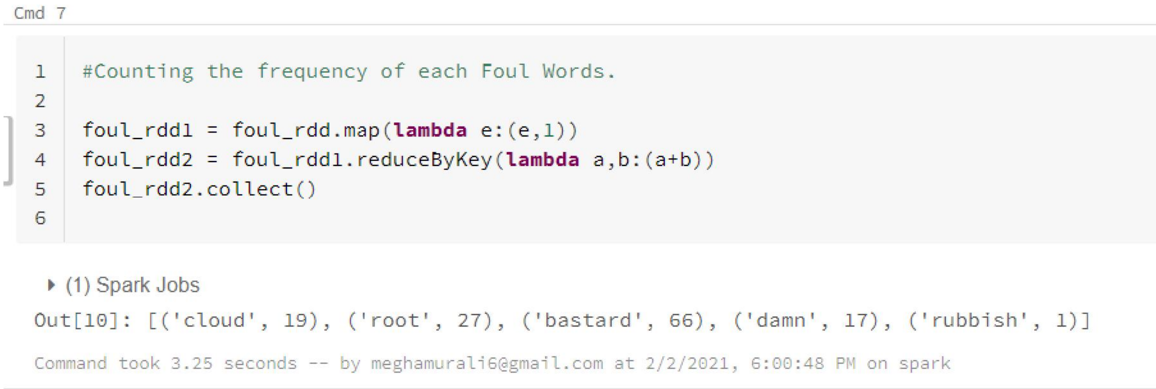
Codes:

foul_rdd1 = foul_rdd.map(lambda e:(e,1))

foul_rdd2 = foul_rdd1.reduceByKey(lambda a,b:(a+b))

foul_rdd2.collect()

Screenshot:



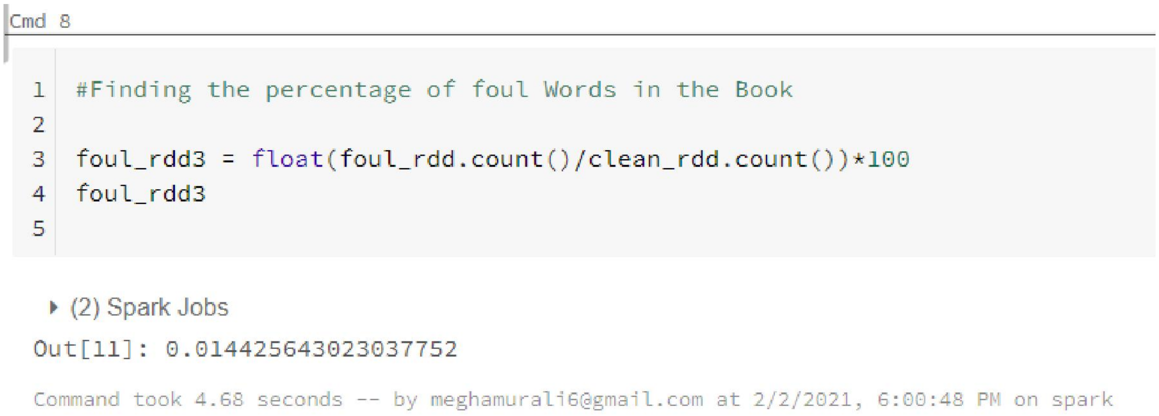
```
Cmd 7

#Counting the frequency of each Foul Words.

foul_rdd1 = foul_rdd.map(lambda e:(e,1))
foul_rdd2 = foul_rdd1.reduceByKey(lambda a,b:(a+b))
foul_rdd2.collect()

▸ (1) Spark Jobs
Out[10]: [('cloud', 19), ('root', 27), ('bastard', 66), ('damn', 17), ('rubbish', 1)]

Command took 3.25 seconds -- by meghamurali6@gmail.com at 2/2/2021, 6:00:48 PM on spark
```

10. Finding the percentage of foul Words in the Book

Codes:

```
foul_rdd3 = float(foul_rdd.count()/clean_rdd.count())*100

foul_rdd3
```

Screenshot:



Cmd 8

```
#Finding the percentage of foul Words in the Book

foul_rdd3 = float(foul_rdd.count()/clean_rdd.count())*100
foul_rdd3

```

▸ (2) Spark Jobs

```
Out[11]: 0.014425643023037752
```

Command took 4.68 seconds -- by meghamurali6@gmail.com at 2/2/2021, 6:00:48 PM on spark

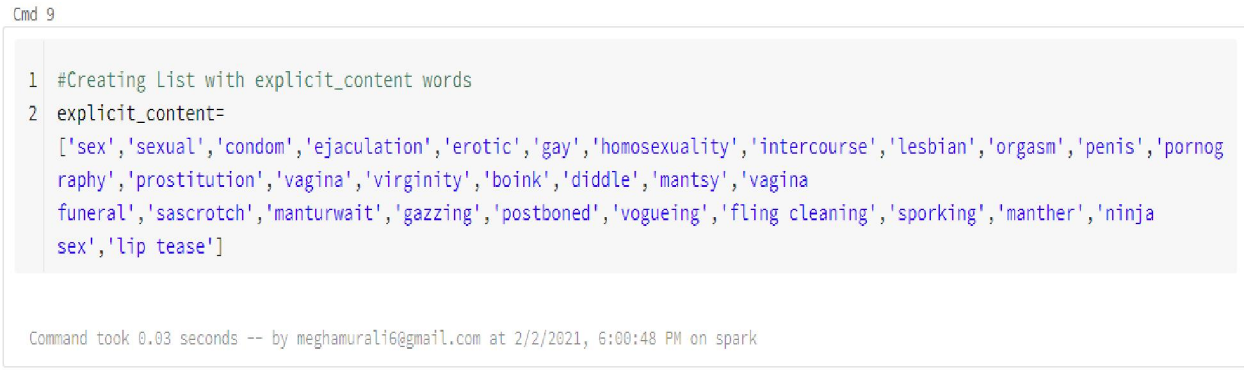
#There is only one percentage of foul words in the entire book on the bases of list what we have created

11. Creating List with explicit content words

Codes:

```
explicit_content=['sex','sexual','condom','ejaculation','erotic','gay','homosexuality','intercourse','lesbian','orgasm','penis','pornography','prostitution','vagina','virginity','boink','diddle','mantsy','vaginafuneral','sascrotch','manturwait','gazzing','postboned','vogueing','fling cleaning','sporking','manther','ninja sex','lip tease']
```

Screenshot:



Cmd 9

```
#Creating List with explicit_content words
explicit_content=['sex','sexual','condom','ejaculation','erotic','gay','homosexuality','intercourse','lesbian','orgasm','penis','pornography','prostitution','vagina','virginity','boink','diddle','mantsy','vagina funeral','sascrotch','manturwait','gazzing','postboned','vogueing','fling cleaning','sporking','manther','ninja sex','lip tease']
```

Command took 0.03 seconds -- by meghamurali6@gmail.com at 2/2/2021, 6:00:48 PM on spark

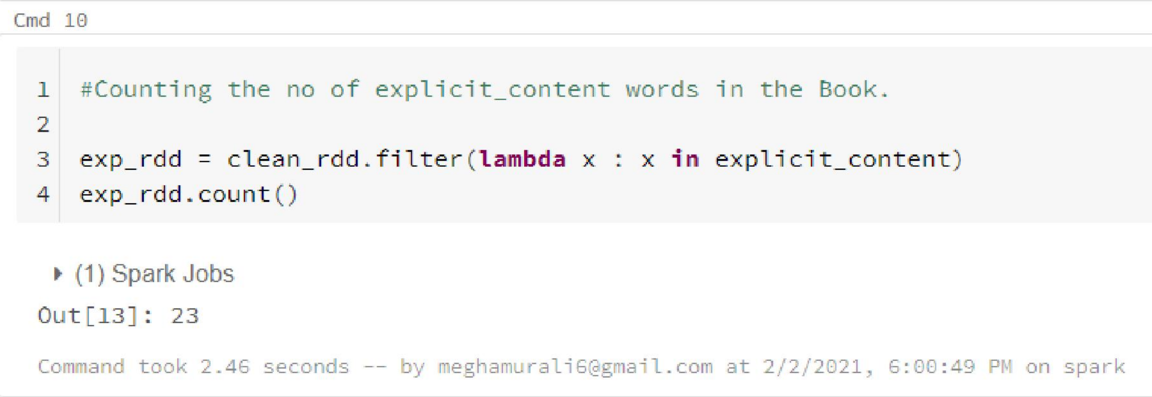
12.Counting the no of explicit content words in the Book.

Codes:

exp_rdd = clean_rdd.filter(lambda x : x in explicit_content)

exp_rdd.count()

Screenshot:



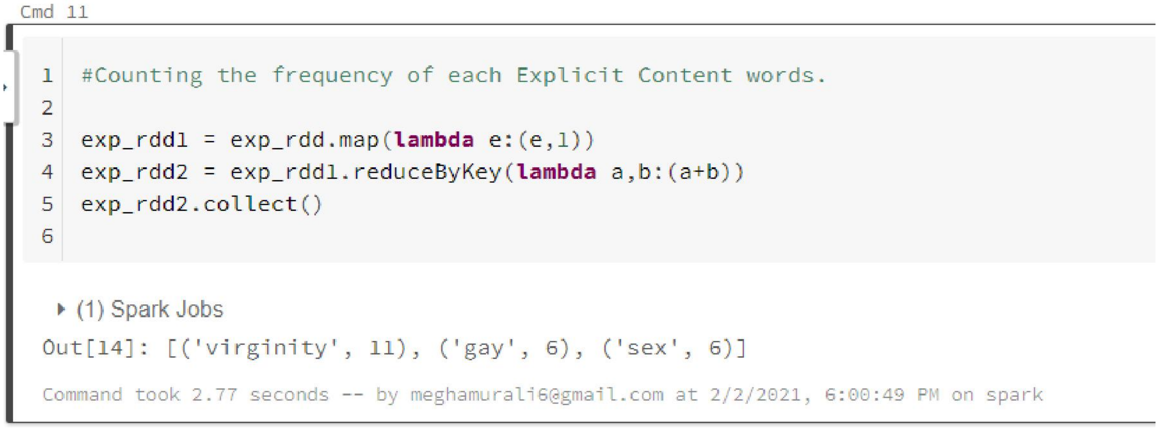
```
Cmd 10

#Counting the no of explicit_content words in the Book.

exp_rdd = clean_rdd.filter(lambda x : x in explicit_content)
exp_rdd.count()

▸ (1) Spark Jobs
Out[13]: 23

Command took 2.46 seconds -- by meghamurali6@gmail.com at 2/2/2021, 6:00:49 PM on spark
```

13. Counting the frequency of each Explicit Content words.

Codes:

exp_rdd1 = exp_rdd.map(lambda e:(e,1))
exp_rdd2 = exp_rdd1.reduceByKey(lambda a,b:(a+b))
exp_rdd2.collect()

Screenshot:



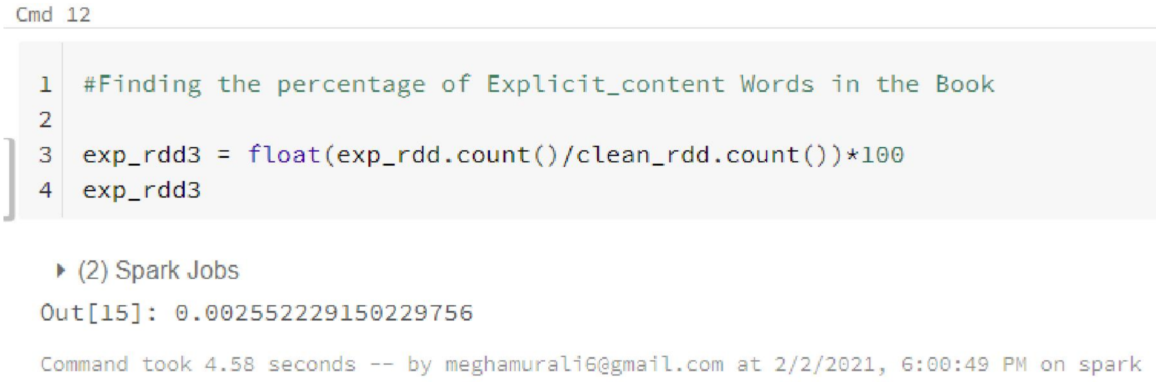
```
Cmd 11

#Counting the frequency of each Explicit Content words.

exp_rdd1 = exp_rdd.map(lambda e:(e,1))
exp_rdd2 = exp_rdd1.reduceByKey(lambda a,b:(a+b))
exp_rdd2.collect()


▸ (1) Spark Jobs
Out[14]: [('virginity', 11), ('gay', 6), ('sex', 6)]

Command took 2.77 seconds -- by meghamurali6@gmail.com at 2/2/2021, 6:00:49 PM on spark
```

14. Finding the percentage of Explicit_content Words in the Book

Codes:

```
exp_rdd3 = float(exp_rdd.count()/clean_rdd.count())*100
exp_rdd3
```

Screenshot:



Cmd 12

```
#Finding the percentage of Explicit_content Words in the Book

exp_rdd3 = float(exp_rdd.count()/clean_rdd.count())*100
exp_rdd3
```

▸ (2) Spark Jobs

```
Out[15]: 0.002552229150229756
```

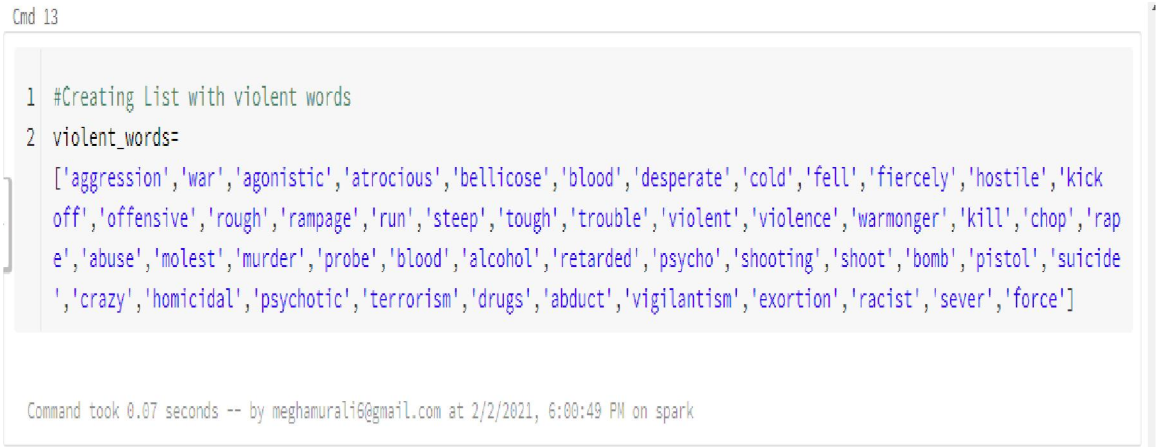
Command took 4.58 seconds -- by meghamurali6@gmail.com at 2/2/2021, 6:00:49 PM on spark

15.Creating List with violent words

Codes:

violent_words=['aggression','war','agonistic','atrocious','bellicose','blood','desperate','cold', 'fell','fiercely','hostile','kick off','offensive','rough','rampage','run','steep','tough' ,'trouble','violent','violence','warmonger','kill','chop','rape','abuse','molest','murder','probe','blood','alcohol','retarded','psycho','shooting','shoot','bomb','pistol','suicide','crazy','homicidal','psychotic','terrorism','drugs','abduct','vigilantism','exortion','racist','sever','force']

Screen Shot:



Cmd 13

```
#Creating List with violent words
violent_words=['aggression','war','agonistic','atrocious','bellicose','blood','desperate','cold','fell','fiercely','hostile','kick off','offensive','rough','rampage','run','steep','tough','trouble','violent','violence','warmonger','kill','chop','rape','abuse','molest','murder','probe','blood','alcohol','retarded','psycho','shooting','shoot','bomb','pistol','suicide','crazy','homicidal','psychotic','terrorism','drugs','abduct','vigilantism','exortion','racist','sever','force']
```

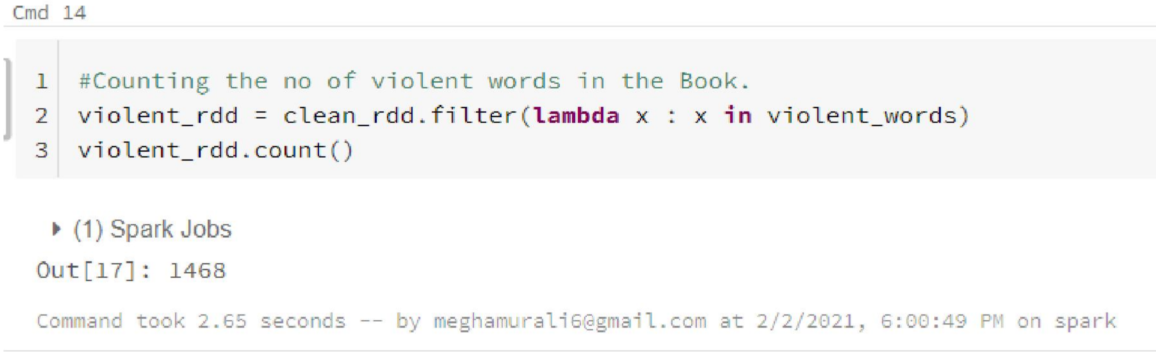
Command took 0.07 seconds -- by meghamurali6@gmail.com at 2/2/2021, 6:00:49 PM on spark

16.Counting the no of violent words in the Book.

Codes:

```
violent_rdd = clean_rdd.filter(lambda x : x in violent_words)
violent_rdd.count()
```

Screenshot:



Cmd 14

```
#Counting the no of violent words in the Book.
violent_rdd = clean_rdd.filter(lambda x : x in violent_words)
violent_rdd.count()
```

▸ (1) Spark Jobs

```
Out[17]: 1468
```

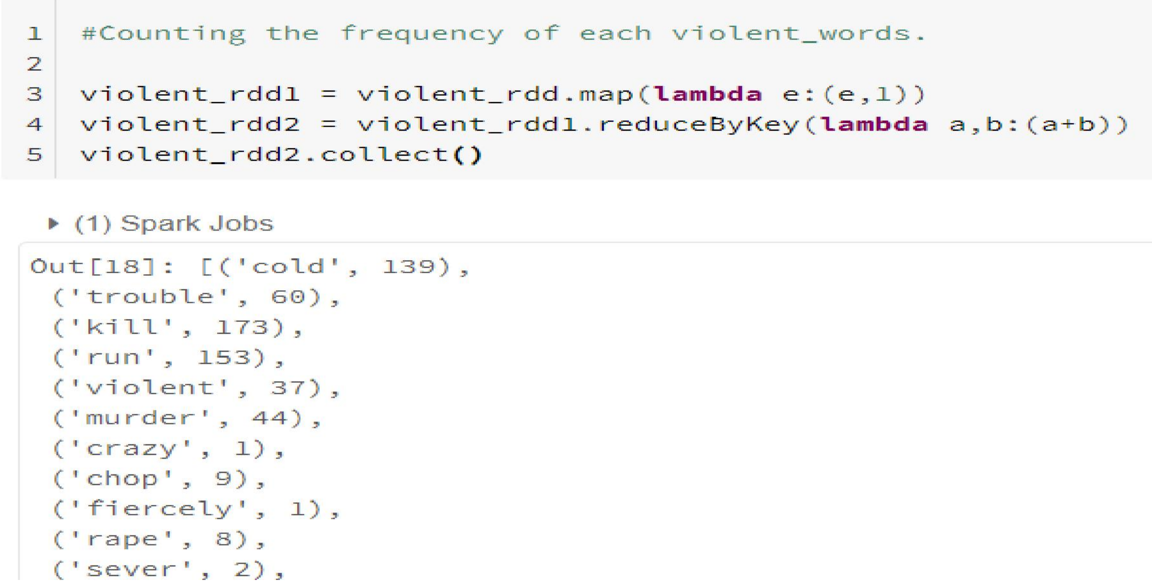
Command took 2.65 seconds -- by meghamurali6@gmail.com at 2/2/2021, 6:00:49 PM on spark

17.Counting the frequency of each violent_words.

Codes:

```
violent_rdd1 = violent_rdd.map(lambda e:(e,1))
violent_rdd2 = violent_rdd1.reduceByKey(lambda a,b:(a+b))
violent_rdd2.collect()
```

Screenshot:



```
#Counting the frequency of each violent_words.

violent_rdd1 = violent_rdd.map(lambda e:(e,1))
violent_rdd2 = violent_rdd1.reduceByKey(lambda a,b:(a+b))
violent_rdd2.collect()
```

▸ (1) Spark Jobs

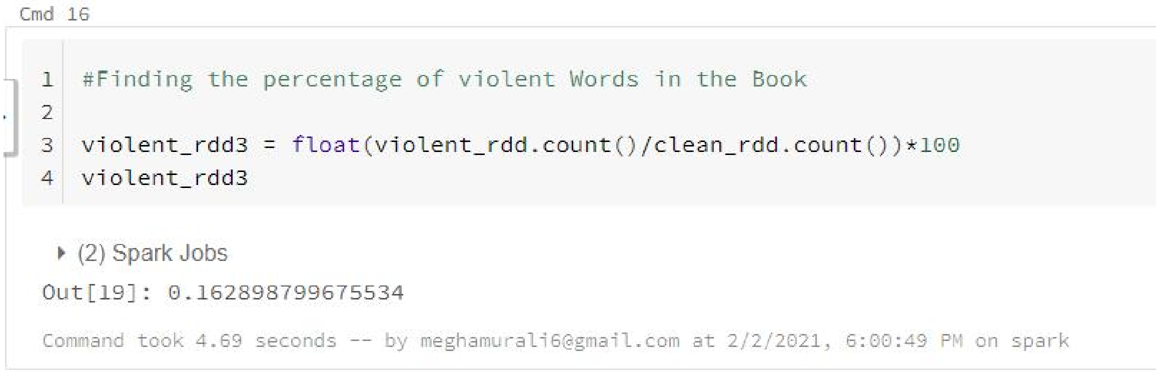
```
Out[18]: [('cold', 139),
 ('trouble', 60),
 ('kill', 173),
 ('run', 153),
 ('violent', 37),
 ('murder', 44),
 ('crazy', 1),
 ('chop', 9),
 ('fiercely', 1),
 ('rape', 8),
 ('sever', 2),
```

18.Finding the percentage of violent Words in the Book

Codes:

```
violent_rdd3 = float(violent_rdd.count()/clean_rdd.count())*100
violent_rdd3
```

Screenshot:



Cmd 16

```
#Finding the percentage of violent Words in the Book

violent_rdd3 = float(violent_rdd.count()/clean_rdd.count())*100
violent_rdd3
```

```
Out[19]: 0.162898799675534
```

```
Command took 4.69 seconds -- by meghamurali6@gmail.com at 2/2/2021, 6:00:49 PM on spark
```

Conclusion

The percentage of foul, violent and explicit content words are very less so that we can highly recommend this book to all age group especially children less than 15 years.

The Sonnets has been specially prepared to help all students in schools and colleges. Each sonnet is presented with accompanying material which aims to enrich your own experience of the poem, whilst leaving you to make your own mind up about the sonnet rather than having someone else's interpretation and judgment handed down to you.