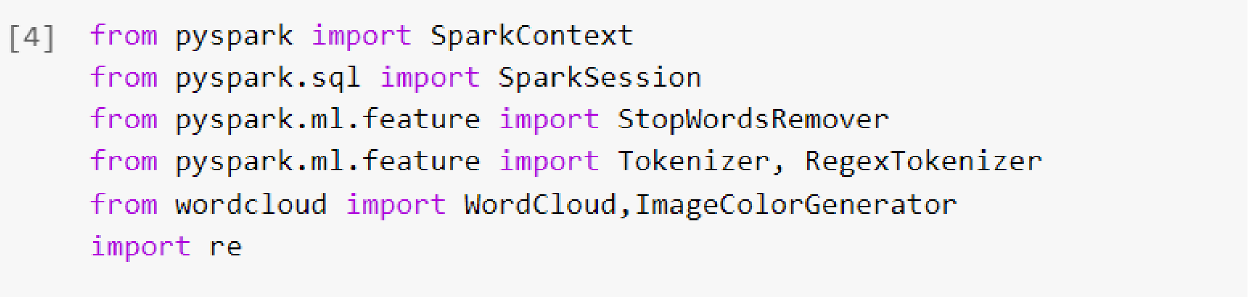
Warm Up Project task 1

Question

Build a word cloud for NY Times articles using Apache Hadoop or Spark. (either platform is fine)-List top 100 words used in all articles-Drop stop words [a, the, in, for, ...]

1. Importing libraries

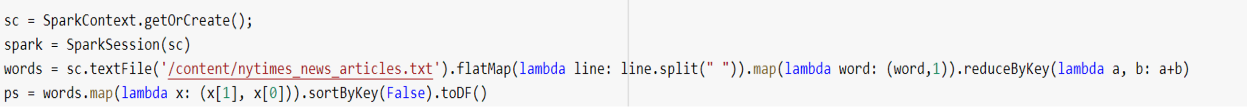
I have used four libraries Pyspark, wordcloud, re, ImageColorGenerator



```
[4]  from pyspark import SparkContext
     from pyspark.sql import SparkSession
     from pyspark.ml.feature import StopWordsRemover
     from pyspark.ml.feature import Tokenizer, RegexTokenizer
     from wordcloud import WordCloud,ImageColorGenerator
     import re
```

2. Spark session and reading into RDD

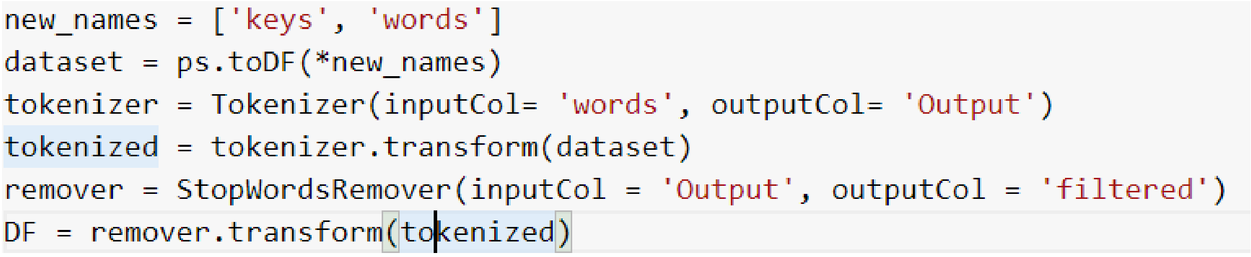
Creating spark session, importing txt file, the article in the form of textile is loaded as RDD, where each line is first split on the basis of presence of space between two words using flat map and further using a lambda function and appoint a key using map to each word, and then keys are added if two words are the same, this is done using reduce by key. Then using map, we sort the words based upon their key values.



```
sc = SparkContext.getOrCreate();
spark = SparkSession(sc)
words = sc.textFile('/content/nytimes_news_articles.txt').flatMap(lambda line: line.split(" ")).map(lambda word: (word,1)).reduceByKey(lambda a, b: a+b)
ps = words.map(lambda x: (x[1], x[0])).sortByKey(False).toDF()
```

3. Stop Words removal

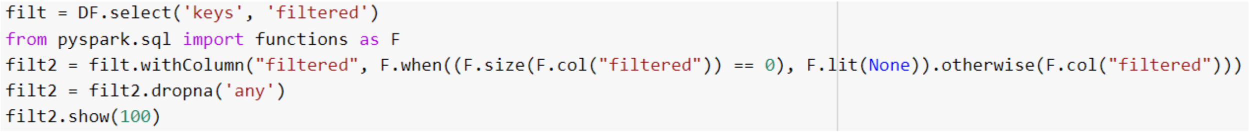
The RDD is converted to a data frame, in order for it to be used by tokenizer because stop word remover package of pyspark.ml requires tokenized words, and filtered column is obtained, the words that were removed had empty string with key. In order to remove that, we set convert empty string to 0. And drop them using dropna('any'). Further we remove any special characters from our column also



```
new_names = ['keys', 'words']
dataset = ps.toDF(*new_names)
tokenizer = Tokenizer(inputCol= 'words', outputCol= 'Output')
tokenized = tokenizer.transform(dataset)
remover = StopWordsRemover(inputCol = 'Output', outputCol = 'filtered')
DF = remover.transform(tokenized)
```

4. Conversion of empty string into NA

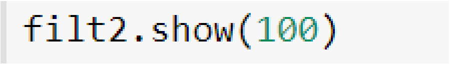
Filtered column is obtained, the words that were removed had empty string with key. In order to remove that, we set convert empty string to 0. And drop them using dropna('any').



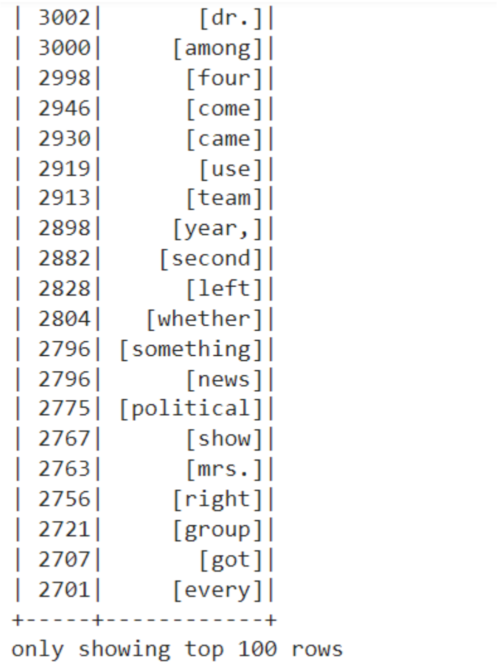
```
filt = DF.select('keys', 'filtered')
from pyspark.sql import functions as F
filt2 = filt.withColumn("filtered", F.when((F.size(F.col("filtered")) == 0), F.lit(None)).otherwise(F.col("filtered")))
filt2 = filt2.dropna('any')
filt2.show(100)
```

5. Top 100 words

Since cannot show all showing the last of them, as you can see they are special characters they are removed further



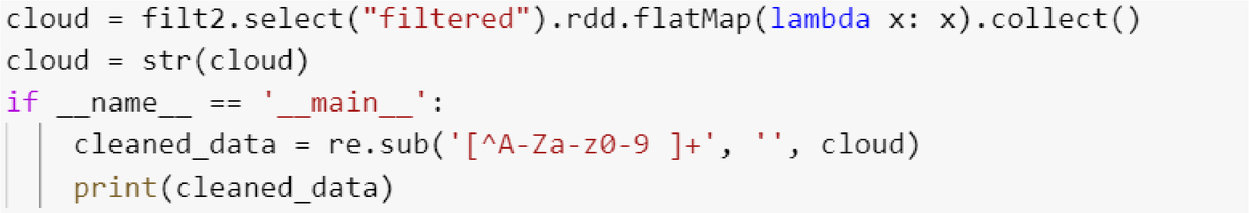
```
filt2.show(100)
```



```
| 3002|       [dr.]|
| 3000|     [among]|
| 2998|      [four]|
| 2946|      [come]|
| 2930|      [came]|
| 2919|       [use]|
| 2913|      [team]|
| 2898|     [year,]|
| 2882|    [second]|
| 2828|      [left]|
| 2804|   [whether]|
| 2796| [something]|
| 2796|      [news]|
| 2775| [political]|
| 2767|      [show]|
| 2763|      [mrs.]|
| 2756|     [right]|
| 2721|     [group]|
| 2707|       [got]|
| 2701|     [every]|
+-----+------------+
only showing top 100 rows
```

6. Removal of Special character

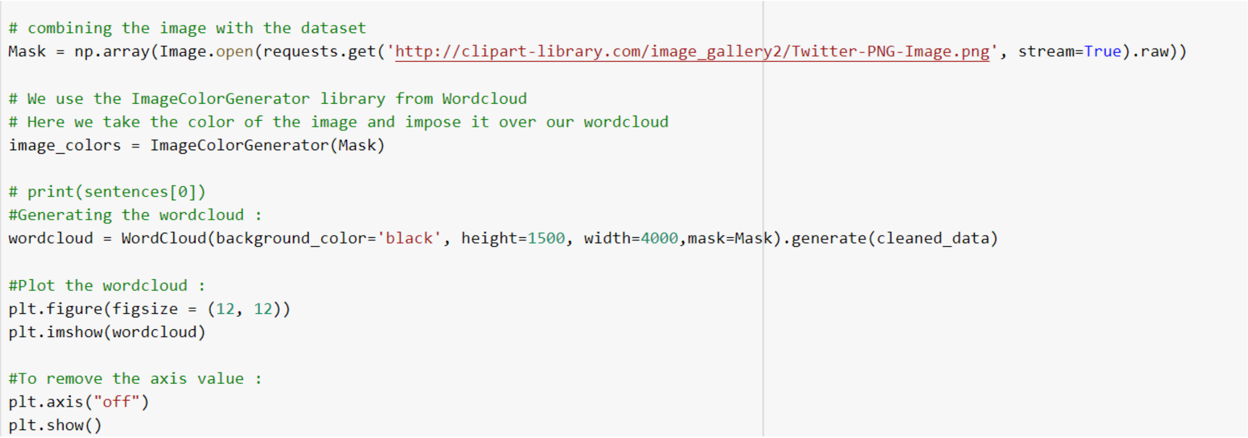
Even after removal of stopwords we have special characters, we remove them as follows



```
cloud = filt2.select("filtered").rdd.flatMap(lambda x: x).collect()
cloud = str(cloud)
if __name__ == '__main__':
    cleaned_data = re.sub('[^A-Za-z0-9 ]+', '', cloud)
    print(cleaned_data)
```

7. Word Cloud

For word cloud we use library word cloud and we create a mask using NumPy array utilizing an image from web. And project the words on to the image using Matplotlib



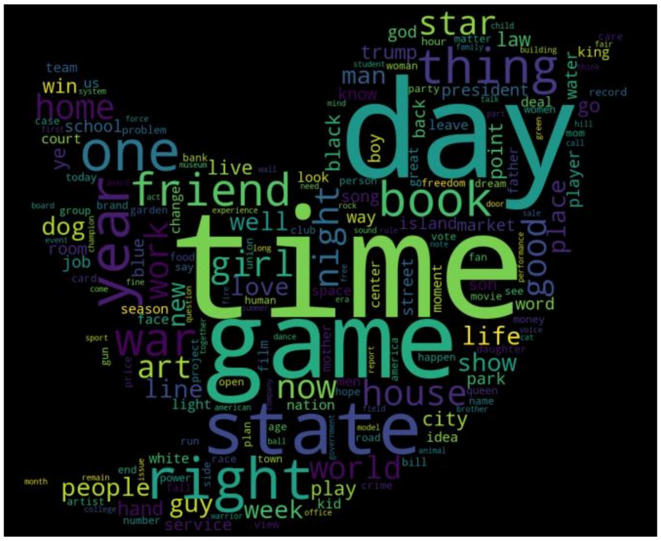
```
# combining the image with the dataset
Mask = np.array(Image.open(requests.get('http://clipart-library.com/image_gallery2/Twitter-PNG-Image.png', stream=True).raw))

# We use the ImageColorGenerator library from Wordcloud
# Here we take the color of the image and impose it over our wordcloud
image_colors = ImageColorGenerator(Mask)

# print(sentences[0])
#Generating the wordcloud :
wordcloud = WordCloud(background_color='black', height=1500, width=4000,mask=Mask).generate(cleaned_data)

#Plot the wordcloud :
plt.figure(figsize = (12, 12))
plt.imshow(wordcloud)

#To remove the axis value :
plt.axis("off")
plt.show()
```


time
game
day
state
right
thing
friend
book
one
year
star
home
war
world
house
people
guy
night
place
good
girl
love
art
now
life
show
play
week
work
new
black
boy
man
president
point
street
center
moment
law
king
god
trump
water
deal
go
win
team
us
system
school
problem
court
dog
job
room
group
brand
season
face
line
light
nation
city
idea
park
word
song
island
market
well
way
live
look
father
player
back
great
leave
dream
freedom
person
kid
hand
service
white
end
power
remain
artist
college
month
number
plan
age
ball
road
bill
run
race
town
office
open
american
hope
film
dance
union
change
today
card
sport
gun
price
project
blue
food
say
fine
human
space
era
free
movie
see
son
fan
vote
note
money
voice
daughter
happen
queen
name
brother
model
animal
crime
view