

WARM UP PROJECT MODIFIED FILE

FOR TASK 1

It states that generate wordcloud of NY articles, so I created the wordcloud of entire news article and words on getting top 100 words separately.

Generate Top 100 words of NY article

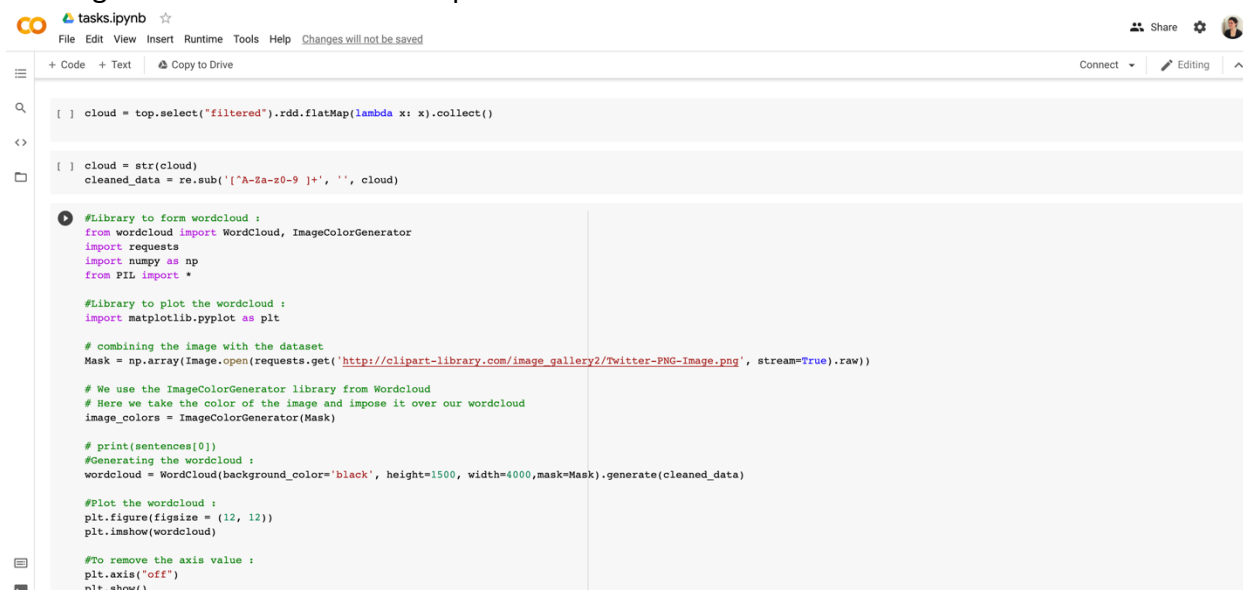


```
from pyspark import SparkContext
from pyspark.sql import SparkSession
from pyspark.ml.feature import StopWordsRemover
from pyspark.ml.feature import Tokenizer, RegexTokenizer
from wordcloud import WordCloud, ImageColorGenerator
import re
import pyspark.sql.functions as f

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()
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)
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)
top = filt2.limit(100)
```

4045	[company]
3995	[work]
3948	[part]
3916	["we]
3914	[take]

Plotting the wordcloud of these top 100 words



```
[ ] cloud = top.select("filtered").rdd.flatMap(lambda x: x).collect()

[ ] cloud = str(cloud)
cleaned_data = re.sub('[^A-Za-z0-9 ]+', '', cloud)

#Library to form wordcloud :
from wordcloud import WordCloud, ImageColorGenerator
import requests
import numpy as np
from PIL import *

#Library to plot the wordcloud :
import matplotlib.pyplot as plt

# 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()
```

Output of Word cloud.

