DM_Report 2

Chen Min
K20002189
MSc Artificial Intelligence

1. Text Mining

1.1 Tweets sentiment analysis:

Compute the possible sentiments of each tweet:

The possible sentiments: {'Neutral': '0.1874', 'Positive': '0.2775', 'Extremely Negative': '0.1332', 'Negative': '0.2410', 'Extremely Positive': '0.1609'}

Compute the second most popular sentiment in the tweets:

the second most popular sentiment in the tweets: ('Negative', 9917)

Find the date with the most extremely positive tweets:

the date with the greatest number of extremely positive tweets: 25-03-2020

1.2 Tweets words analysis:

Total number of all words (including repetitions):

Words count: 1203227

Total number of all distinct words:

Words count with no repetition: 43011

The 10 most frequent words in the corpus:

```
The 10 most frequent words:
[ 1 ]:   ('the', 44823)
[ 2 ]:   ('to', 38474)
[ 3 ]:   ('and', 24060)
[ 4 ]:   ('covid', 23164)
[ 5 ]:   ('of', 21552)
[ 6 ]:   ('a', 19456)
[ 7 ]:   ('in', 19318)
[ 8 ]:   ('coronavirus', 18116)
[ 9 ]:   ('for', 14049)
[ 10 ]:   ('is', 12248)
```

After remove stop words and words <=2 characters, repeat the above calculation again:

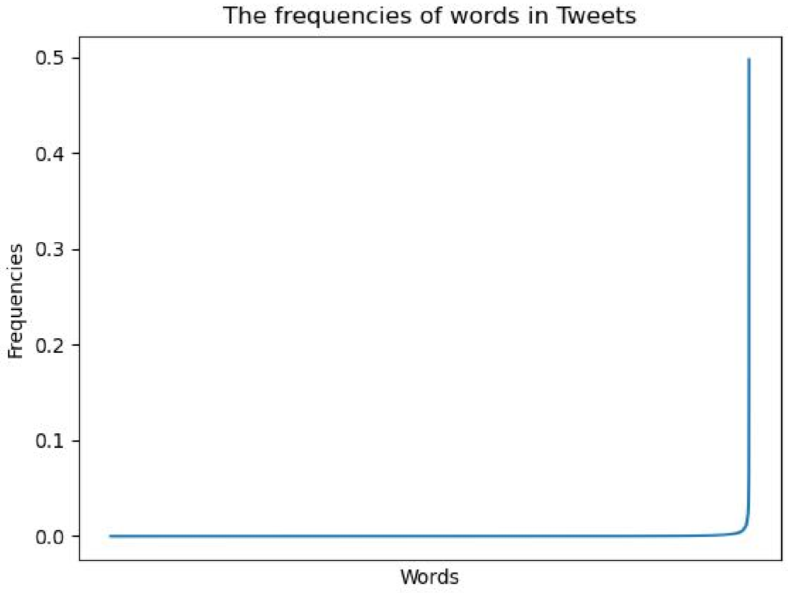
[In this step, i remove stop words, words <= 2 characters and also some other format of words like:'https:\\...' and '@XXX'] ***(I don't know my consideration is correct or not, but it seems like more in line with requirement)***

[Removed]Words count: 700059

[Removed]Words count with no repetition: 42409

```
[Removed]The 10 most frequent words:
[ 1 ]:    ('covid', 23164)
[ 2 ]:    ('coronavirus', 18116)
[ 3 ]:    ('prices', 7870)
[ 4 ]:    ('food', 7134)
[ 5 ]:    ('supermarket', 7035)
[ 6 ]:    ('store', 6852)
[ 7 ]:    ('grocery', 6277)
[ 8 ]:    ('people', 5585)
[ 9 ]:    ('amp', 5191)
[ 10 ]:    ('consumer', 4630)
```

1.3 Plot the line chart of words frequency([x - words][y - words frequencies])



All words frequencies

The shape of CountVectorizer = (41157, 42409) [because i deleted url(“https://”) and @(@XXX)]

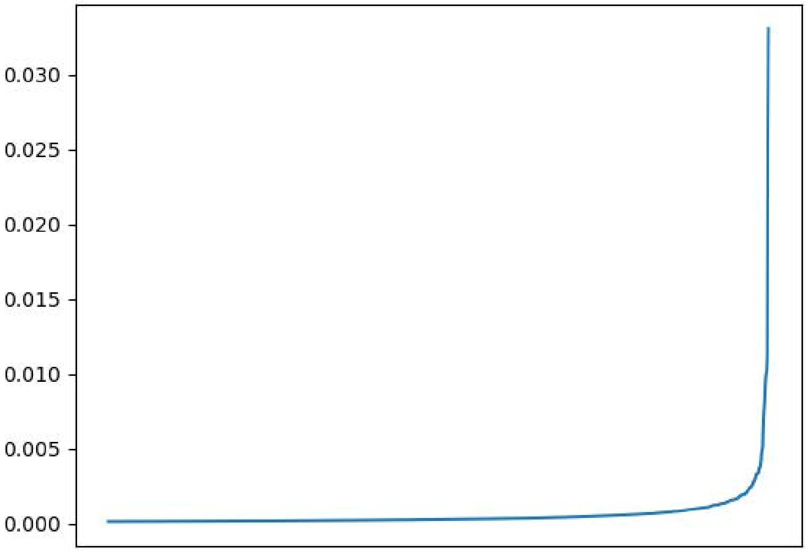
In what way this plot can be useful for deciding the size of the term document matrix?

Since it can be seen from all the words frequency line graphs that most of the words appear frequently (the data generated by the debug query can tell that their number of appearances less than 100 times), so when choosing the size of the document matrix At this time, you can consider selecting only words with more than one occurrence.

How many terms would you add in a term-document matrix for this data set?

After my experiment, I found that the number of occurrences of 100 is a more appropriate number. It not only retains the image of the image without major changes, but also has a great improvement in the speed of image generation.

After build new term-document matrix of data set, the figure like below:



Words frequency > 100

1.4 Naive Bayes classifier for tweets:

The error rate of classifier:

the Error rate of MultinomialNB: 0.28065699637971675

So after accurate to 4 decimal places, the error rate is 0.2807

2. Image Processing

2.1 Avengers Image:

Determine the size of the avengers image:

Size of avengers image is 2268000.

Shape of avengers image is (1200, 630, 3).

So width x height = (1200, 630)

(don’t actually understand the ‘size’ in question, therefore i give two answer)

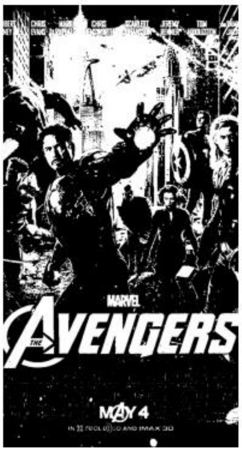
Transform to grayscale and binary:

Grayscale figure:

MARVEL
AVENGERS
MAY 4

avengers_gray.jpg

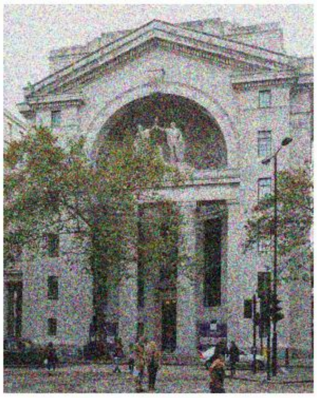
Binary figure:



avengers_binary.jpg

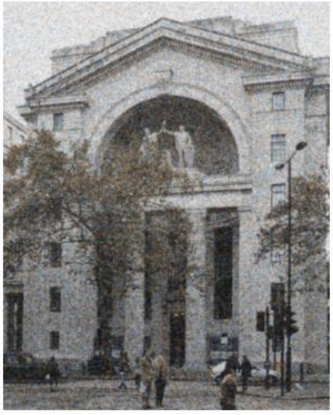
2.2 Bush house Image:

Add Gaussian random noise:



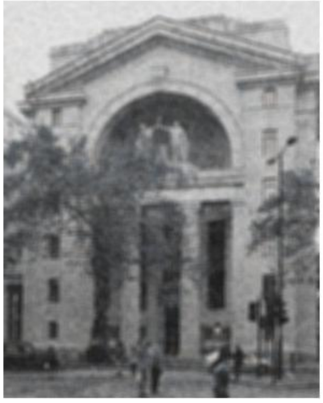
bush_gaussian.jpg

Filter with Gaussian mask(sigma = 1):



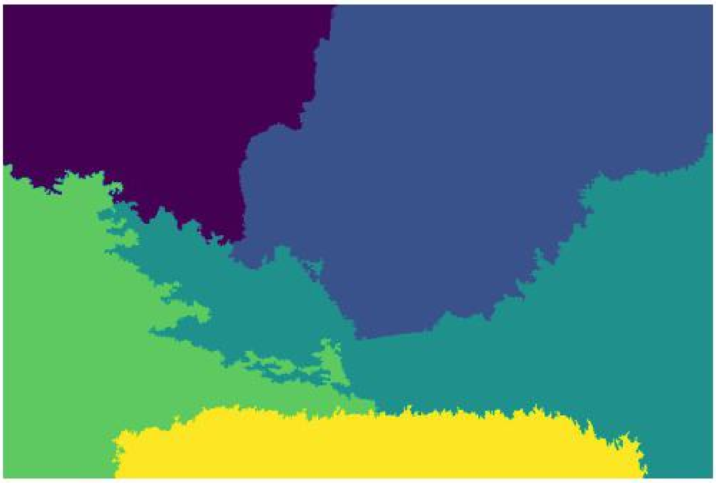
bush_gaussian_filter.jpg

Filter with uniform smoothing mask(size = 9):



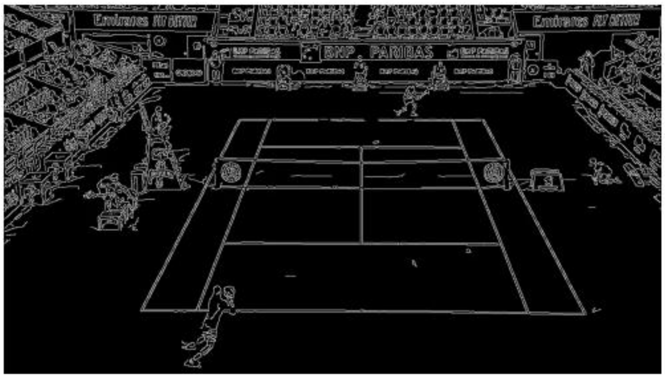
bush_uniform_filter.jpg

2.3 Forestry Image with K-means segmentation:



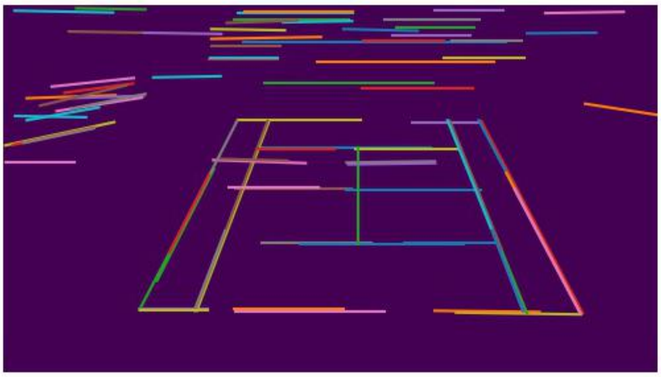
forestry_kmeans.jpg

2.4 Perform Canny edge detection and Hough Transform on rolland Image:
After perform Canny edge detection:



rolland_edges.jpg

Then add Hough Transform:



rolland_hough.jpg