404 error

November 21, 2017

0.1 The preprocessing of the dataset

```
In [143]: import pandas as pd
          import sklearn as skn
          import matplotlib.pyplot as plt
          from bs4 import BeautifulSoup as bs
          import re
          import os
          import numpy as np
```

```
In [144]: # Remove all the tags, scripts in the html
          def clean_html(html):
              soup = bs(html)
              try:
                  title = soup.title.contents[0]
              except (AttributeError, IndexError):
                  title = ''
              for s in soup(['script','style']):
                  s.decompose()
              return ' '.join(soup.stripped_strings), title
```

```
In [145]: html_folder = "/Users/lichenle/Desktop/lri/dataset/from-newslist/"
          train = []
          # This step I will transfer all the data original in to titles and bodies
          for html_part in os.listdir(html_folder):
              #print(html_part)
              if not html_part.startswith('.'):
                  if html_part == 'ok':
                      mark = True
                  elif html_part == '404':
                      mark = False
                  else:
                      mark = 'NaN'
                  html_part_path = html_folder+html_part
                  for html in os.listdir(html_part_path):
                      html_path = html_part_path+"/"+html
                      #print(html_path)
                      temp = {}
```

```
                        temp['body'], temp['title'] = clean_html(open(html_path))
                        temp['type'] = mark
                        temp['path'] = html_part+html
                        train.append(temp)
```

```
/Users/lichenle/anaconda/lib/python3.6/site-packages/bs4/__init__.py:181: UserWarni

The code that caused this warning is on line 193 of the file /Users/lichenle/anacon

 BeautifulSoup([your markup])

to this:

 BeautifulSoup([your markup], "lxml")

  markup_type=markup_type))
```

```
In [146]: train_df = pd.DataFrame(train, columns=['title', 'body', 'type'])
```

```
In [147]: train_df.info()
          train_df.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 310 entries, 0 to 309
Data columns (total 3 columns):
title    310 non-null object
body     310 non-null object
type     310 non-null bool
dtypes: bool(1), object(2)
memory usage: 5.2+ KB
```

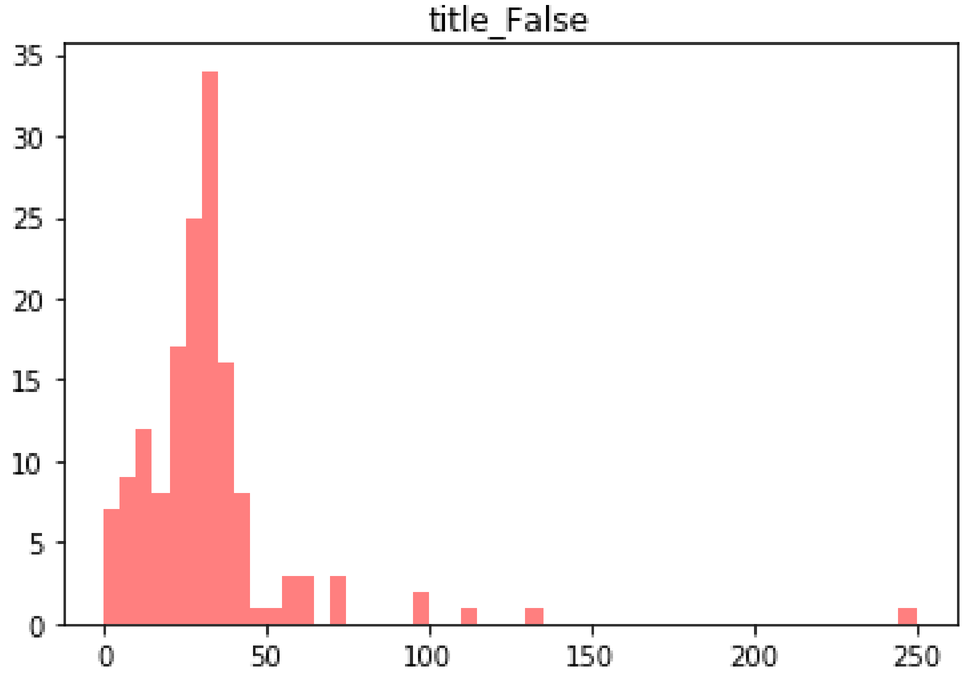
```
Out[147]:                                                title  \
          0  Obama officials treated "special relationship ...
          1  Texas church shooter once escaped from mental ...
          2  Harvey Weinstein is finding that few in Hollyw...
          3  Massive fire erupts at Moscow market causing c...
          4                                       Bangkok Post

                                                          body  type
          0  Obama officials treated "special relationship ...  True
          1  Texas church shooter once escaped from mental ...  True
          2  Harvey Weinstein is finding that few in Hollyw...  True
          3  Massive fire erupts at Moscow market causing c...  True
          4  Bangkok Post <img src="//b.scorecardresearch.c...  True
```

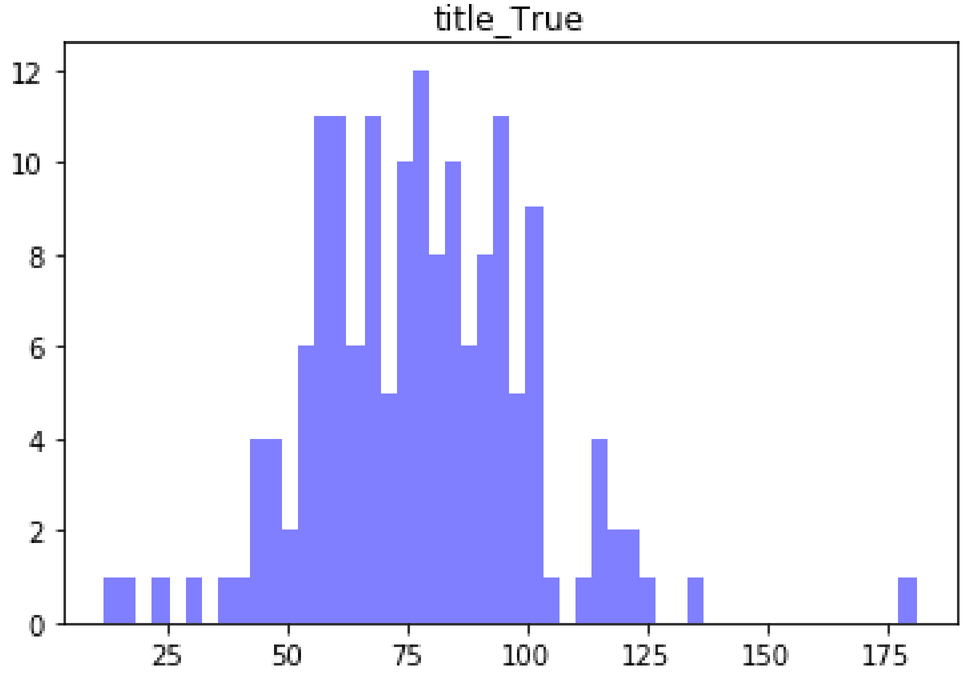
0.2 Data Analysis

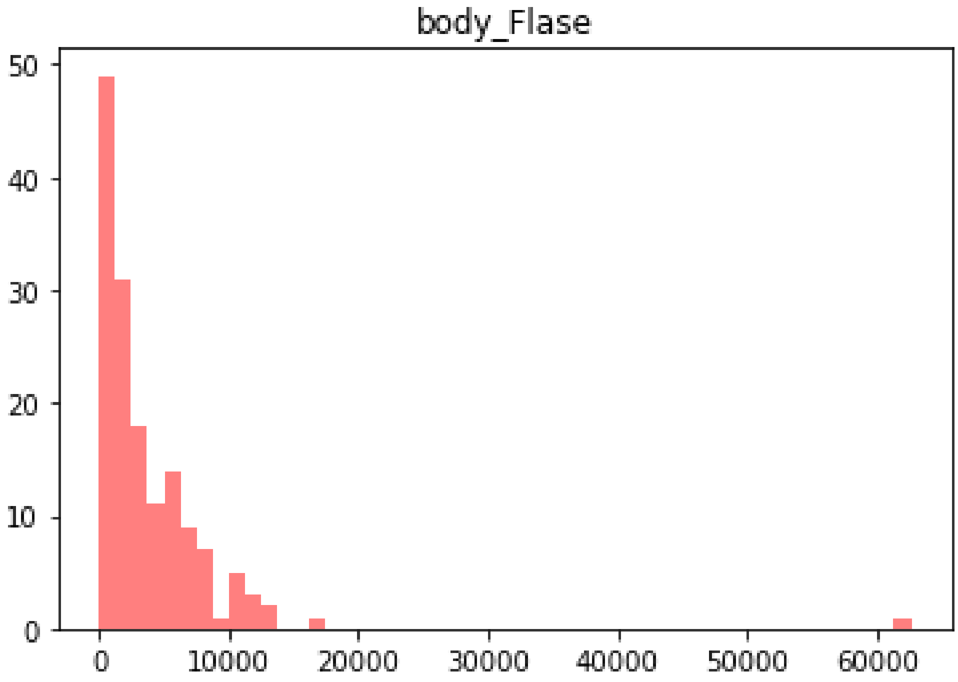
0.2.1 Length Analysis

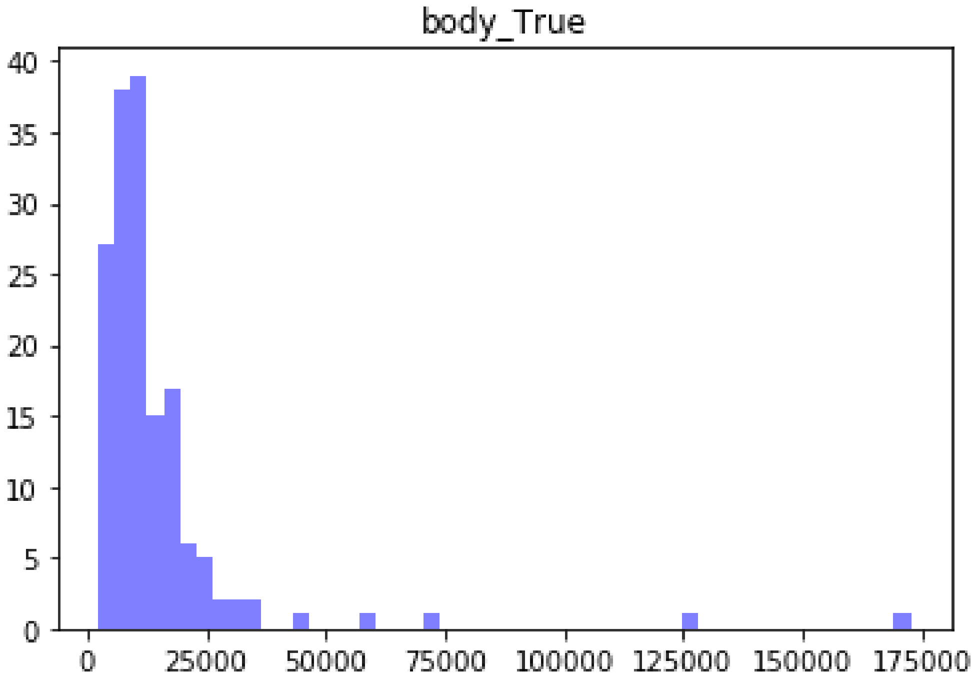
```
In [148]: # Illustre Respectivement The Length Variation of words of title and bod
          length_title_True = []
          length_body_True = []
          length_title_False = []
          length_body_False = []
          for row in train_df.itertuples():
              #print(row)
              if row[3] == False:
                  length_title_False.append(len(row[1]))
                  length_body_False.append(len(row[2]))
              elif row[3] == True:
                  length_title_True.append(len(row[1]))
                  length_body_True.append(len(row[2]))
```

```
In [149]: plt.hist(length_title_False, bins=50, color='red', label='title_False', a
          plt.title('title_False')
          plt.show()
          plt.hist(length_title_True, bins=50, color='blue', label='title_True', al
          plt.title('title_True')
          plt.show()
          plt.hist(length_body_False, bins=50, color='red', label='body_False', alp
          plt.title('body_Flase')
          plt.show()
          plt.hist(length_body_True, bins=50, color='blue', label='title_True', alp
          plt.title('body_True')
          plt.show()
```


title_False
35
30
25
20
15
10
5
0
0
50
100
150
200
250

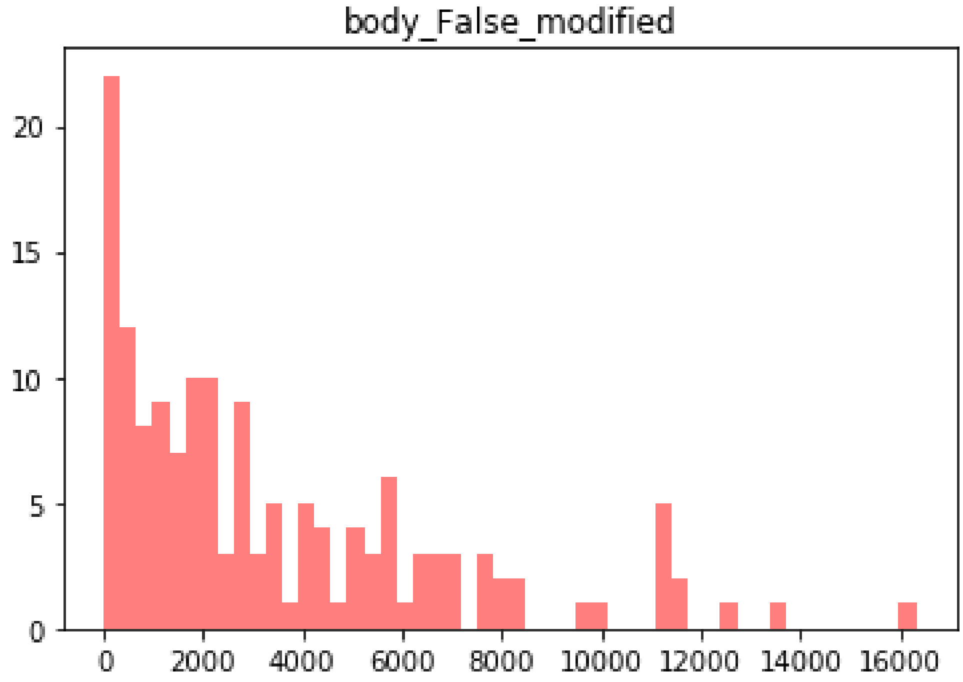

title_True
12
10
8
6
4
2
0
25
50
75
100
125
150
175

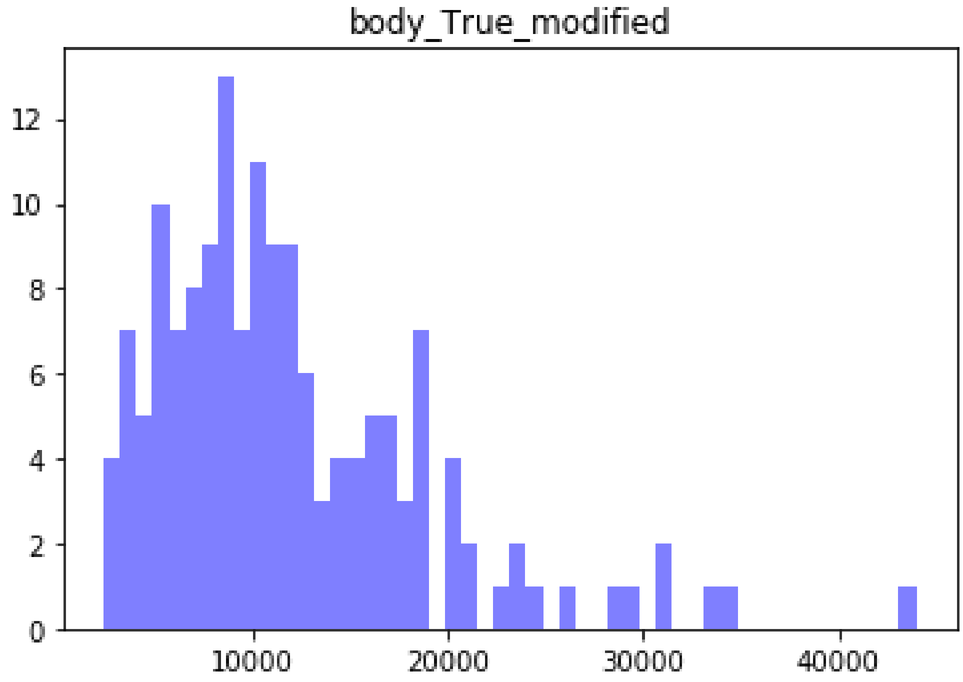




```
In [150]: #Seeing the graph 'body_True', we'd like to remove the outliers, so we ar
          def lessthan1(element):
```

```
    return element < 50000
def lessthan2(element):
    return element < 20000
length_body_True_modified = list(filter(lessthan1, length_body_True))
length_body_False_modified = list(filter(lessthan2, length_body_False))
plt.hist(length_body_False_modified, bins=50, color='red', label='body_Fa
plt.title('body_False_modified')
plt.show()
plt.hist(length_body_True_modified, bins=50, color='blue', label='title_T
plt.title('body_True_modified')
plt.show()
```





Conclusion: As we can see from these graphs, these two features, length of titles and length of bodies, are interesting.

0.2.2 NLP Analysis

```
In [ ]:
```