In this assignment, we wrangle thee data from controvertial comments data set.

We will perfomr initial cleanup of the data and create new columns to work with

We will thenvectorize the comments using CountVectorizer and TF-IDF models.

we will then calculate the jaccard distance beween two samples of the data set to show similairy

Using keywords deemed toward 'liberal', 'conservative', 'Positive1','Positive2', and 'negative', we will quantify them and show scatter plot and stats

In [83]:

```
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
import numpy as np
import collections
import matplotlib.pyplot as plt
```

In [60]:

```
def jaccard(a, b):
    c = a.intersection(b)
    #print("this is intersection", c)

    d=a.union(b)
    #print("this is union", d)
    _a = collections.Counter(a)
    _b = collections.Counter(b)
    c = (_a - _b) + (_b - _a)
    #print("c", c)
    n = sum(c.values())
    #print(_a)
    #print(_b)
    #print("_a and _b")
    #print ("n", n)
    return float(len(c)) / (len(a) + len(b) - len(c))
```

In [144]:

```
file ='controversial-comments.jsonl'
# possible orient value: split, records, index, columns, and values)
# The following file is a subset of above file with the 1st 233537 lines.
dataframe = pd.read_json("controversial-comments_small.jsonl",orient="columns",lines=True)
```

In [5]:

```
dataframe.shape
```

Out[5]:

```
(233538, 2)
```

In [145]:

```
# take a 1000 record sample of data with no duplicate records
df1 = dataframe.sample(n=1000, replace=True, random_state=10)
df2 = dataframe.sample(n=1000, replace=True, random_state=20)
```

In [146]:

```
df1.head()
```

Out[146]:

	con	txt
83209	0	I don't know. But you're right, she certainly ...
94735	0	Young-earth creationists, evolution deniers, a...
181568	0	Literally every thing you have said is incorre...
93553	0	There's a huge difference between violence don...
105595	0	It's not. \n\nReally though any Democrat with ...

In [147]:

```
df2.head()
```

Out[147]:

	con	txt
92634	0	Your post broke my heart. Please don't despair...
37135	0	That is simply not how evidence functions.
23775	0	I agree, I think both are true. We tend to deh...
220060	0	[removed]
31962	0	If you are implying that we are fucked no matt...

In [197]:

```
# convert to lower case
def ToLower(string: str) -> str:
    return string.lower()
```

In [198]:

```
df1['lower_txt'] = [ToLower(string) for string in df1['txt']]
df2['lower_txt'] = [ToLower(string) for string in df2['txt']]
df1.head(5)
```

Out[198]:

	con	txt	lower_txt	Tokenized	CleanText
83209	0	I don't know. But you're right, she certainly ...	i don't know. but you're right, she certainly ...	[i don't know., but you're right, she certainl...	i don't know. but you're right, she certainly ...
94735	0	Young-earth creationists, evolution deniers, a...	young-earth creationists, evolution deniers, a...	[young-earth creationists, evolution deniers, ...	young-earth creationists, evolution deniers, a...
181568	0	Literally every thing you have said is incorre...	literally every thing you have said is incorre...	[literally every thing you have said is incorr...	literally every thing you have said is incorre...
93553	0	There's a huge difference between violence don...	there's a huge difference between violence don...	[there's a huge difference between violence do...	there's a huge difference between violence don...
105595	0	It's not. \n\nReally though any Democrat with ...	it's not. \n\nreally though any democrat with ...	[it's not., really though any democrat with an...	it's not. \n\nreally though any democrat with ...

In [199]:

```
from nltk import sent_tokenize

# Tokenize words
df1['Tokenized'] = [sent_tokenize(string) for string in df1['lower_txt']]
df2['Tokenized'] = [sent_tokenize(string) for string in df2['lower_txt']]
df1.head(5)
```

Out[199]:

	con	txt	lower_txt	Tokenized	CleanText
83209	0	I don't know. But you're right,	i don't know. but you're right,	[i don't know., but you're right,	i don't know. but you're right,

	con	txt	lower_txt	Tokenized	CleanText
		she certainly ...	she certainly ...	she certainl...	she certainly ...
94735	0	Young-earth creationists, evolution deniers, a...	young-earth creationists, evolution deniers, a...	[young-earth creationists, evolution deniers, ...	young-earth creationists, evolution deniers, a...
181568	0	Literally every thing you have said is incorre...	literally every thing you have said is incorre...	[literally every thing you have said is incorr...	literally every thing you have said is incorre...
93553	0	There's a huge difference between violence don...	there's a huge difference between violence don...	[there's a huge difference between violence do...	there's a huge difference between violence don...
105595	0	It's not. \n\nReally though any Democrat with ...	it's not. \n\nreally though any democrat with ...	[it's not., really though any democrat with an...	it's not. \n\nreally though any democrat with ...

In [200]:

```
df1['Tokenized']
```

Out[200]:

```
83209     [i don't know., but you're right, she certainl...
94735     [young-earth creationists, evolution deniers, ...
181568    [literally every thing you have said is incorr...
93553     [there's a huge difference between violence do...
105595    [it's not., really though any democrat with an...
                                ...
81173     [up and downvoting is the way that the entiret...
219185    [how hard is it to vote when you have weeks av...
141377    [almost everything around you has something in...
123782    [i'm used to that aspect of a lot of trumpers....
53634     [on my facebook today i saw someone say that w...
Name: Tokenized, Length: 1000, dtype: object
```

In [201]:

```
import re
#remove everything that has numbers in it
def removeInvalidWords(text):
    return re.sub(r"\d+", '',text)
```

In [202]:

```
df1['CleanText'] = [removeInvalidWords(string) for string in df1['lower_txt']]
df1.head(5)
```

Out[202]:

	con	txt	lower_txt	Tokenized	CleanText
83209	0	I don't know. But you're right, she certainly ...	i don't know. but you're right, she certainly ...	[i don't know., but you're right, she certainl...	i don't know. but you're right, she certainly ...
94735	0	Young-earth creationists, evolution deniers, a...	young-earth creationists, evolution deniers, a...	[young-earth creationists, evolution deniers, ...	young-earth creationists, evolution deniers, a...
181568	0	Literally every thing you have said is incorre...	literally every thing you have said is incorre...	[literally every thing you have said is incorr...	literally every thing you have said is incorre...
93553	0	There's a huge difference between violence don...	there's a huge difference between violence don...	[there's a huge difference between violence do...	there's a huge difference between violence don...
105595	0	It's not. \n\nReally though any Democrat with ...	it's not. \n\nreally though any democrat with ...	[it's not., really though any democrat with an...	it's not. \n\nreally though any democrat with ...

In [203]:

```
df2['CleanText'] = [removeInvalidWords(string) for string in df2['lower_txt']]
df2.head(5)
```

Out[203]:

	con	txt	CleanText	lower_txt	Tokenized
92634	0	Your post broke my heart. Please don't despair...	your post broke my heart. please don't despair...	your post broke my heart. please don't despair...	[your post broke my heart., please don't despa...
37135	0	That is simply not how evidence	that is simply not how evidence	that is simply not how evidence	[that is simply not how evidence

		functions.	functions.	functions.	functions.]
	con	txt	CleanText	lower_txt	Tokenized
23775	0	I agree, I think both are true. We tend to deh...	i agree, i think both are true. we tend to deh...	i agree, i think both are true. we tend to deh...	[i agree, i think both are true., we tend to d...
220060	0	[removed]	[removed]	[removed]	[[removed]]
31962	0	If you are implying that we are fucked no matt...	if you are implying that we are fucked no matt...	if you are implying that we are fucked no matt...	[if you are implying that we are fucked no mat...

Use CountVectorizer to create list of words and feature matrix

In [205]:

```
vectorizer = CountVectorizer()
X1 = vectorizer.fit_transform(df1['CleanText'])
list1 = vectorizer.get_feature_names()
print(list1[:30])
```

```
['_lfxpiobm', '_r', '_states_by_poverty_rate', 'abandoned', 'abased', 'abba', 'abcnews', 'abe', 'a
bility', 'able', 'abortion', 'abortions', 'about', 'above', 'abraham', 'abroad', 'absolute', 'abso
lutely', 'absurd', 'absurdity', 'absurdly', 'abuse', 'abused', 'abusing', 'abysmal', 'aca',
'accelerate', 'accept', 'acceptable', 'acceptance']
```

In [181]:

```
print( X1.toarray())
X2.shape
```

```
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
```

Out[181]:

```
(1000, 5352)
```

In [206]:

```
vectorizer = CountVectorizer()
X2 = vectorizer.fit_transform(df2['CleanText'])
list2 = vectorizer.get_feature_names()
print(list2[:30])
```

```
['_r', '_x', 'abcnews', 'ability', 'able', 'abolish', 'abomination', 'abort', 'abortion',
'abortions', 'about', 'above', 'abroad', 'absolute', 'absolutely', 'abstain', 'absurd', 'abuser',
'abysmal', 'aca', 'academics', 'accept', 'accepted', 'accepting', 'accident', 'accomodating', 'acc
omplished', 'accordance', 'according', 'account']
```

In [180]:

```
print( X2.toarray())
X2.shape
```

```
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
```

Out[180]:

```
(1000, 5352)
```

jaccard distance show the two samples are 90% similar

In [210]:

```
words1 = set(list1)
words2 = set(list2)

z = jaccard(words1, words2)
print("this is percent similarity", z*100)
```

```
this is percent similarity 90.47790154509522
```

Use TfidVectorizer to create list of words and feature matrix

In [211]:

```
#TfidVectorizer Example
#Provide a list of all words in the corpus

from sklearn.feature_extraction.text import TfidfVectorizer
corpus = df1['CleanText']
vectorizer = TfidfVectorizer()
X1 = vectorizer.fit_transform(corpus)
list1 = vectorizer.get_feature_names()[:30]
print(list1)

print(X1.shape)
```

```
['_lfxpiobm', '_r', '_states_by_poverty_rate', 'abandoned', 'abased', 'abba', 'abcnews', 'abe', 'a
bility', 'able', 'abortion', 'abortions', 'about', 'above', 'abraham', 'abroad', 'absolute', 'abso
lutely', 'absurd', 'absurdity', 'absurdly', 'abuse', 'abused', 'abusing', 'abysmal', 'aca',
'accelerate', 'accept', 'acceptable', 'acceptance']
(1000, 5250)
```

In [212]:

```
#TfidVectorizer Example
#Provide a list of all words in the corpus

from sklearn.feature_extraction.text import TfidfVectorizer
corpus = df2['CleanText']
vectorizer = TfidfVectorizer()
X2 = vectorizer.fit_transform(corpus)
list2 = vectorizer.get_feature_names()[:30]
print(list2)

print(X2.shape)
```

```
['_r', '_x', 'abcnews', 'ability', 'able', 'abolish', 'abomination', 'abort', 'abortion',
'abortions', 'about', 'above', 'abroad', 'absolute', 'absolutely', 'abstain', 'absurd', 'abuser',
'abysmal', 'aca', 'academics', 'accept', 'accepted', 'accepting', 'accident', 'accomodating', 'acc
omplished', 'accordance', 'according', 'account']
(1000, 5352)
```

In [214]:

```
words1 = set(list1)
words2 = set(list2)

z = jaccard(words1, words2)
print("this is percent similarity", z*100)
```

```
this is percent similarity 100.0
```

TfidVectorizer Vectorization yield 100% similarity

sentiment analyis

In [215]:

```
# Take a 10,000 sample
df = dataframe.sample(n=10000, replace=True, random_state=2)
```

In [216]:

```
df['lower_txt'] = [ToLower(string) for string in df['txt']]
df['CleanText'] = [removeInvalidWords(string) for string in df['lower_txt']]
```

In [217]:

```
df['Liberal'] = df.CleanText.str.count('hillary') +  df.CleanText.str.count('clinton') +  df.CleanT
ext.str.count('liberal') +  df.CleanText.str.count('wellware')
df['Conservative']= df.CleanText.str.count('wall') +  df.CleanText.str.count('trump') +  df.CleanTe
xt.str.count('activits') + df.CleanText.str.count('taxes')
```

In [218]:

```
df[['Liberal','Conservative']].head()
```

Out[218]:

	Liberal	Conservative
89256	0	0
100879	0	0
203245	0	0
95816	0	0
175638	0	0

In [219]:

```
NumberofLiberalComments = df['Liberal'][df.Liberal != 0].size
NumberofConservativeComments = df['Conservative'][df.Conservative != 0].size
```

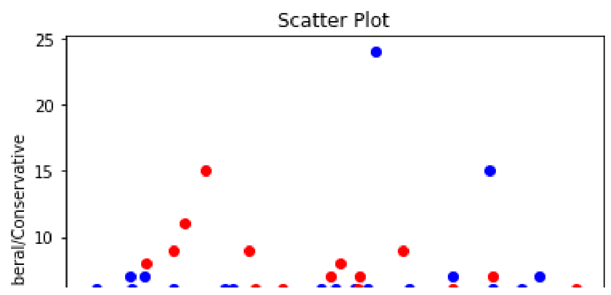
In [220]:

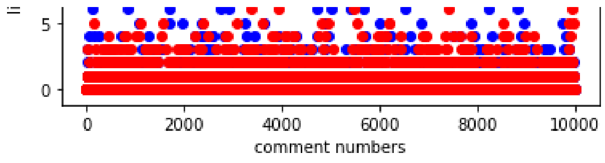
```
print(NumberofLiberalComments)
print(NumberofConservativeComments)
```

```
1006
1766
```

In [222]:

```
plt.scatter(range(0,df['Liberal'].size),df['Liberal'],c='blue')
plt.scatter(range(0,df['Conservative'].size),df['Conservative'],c='red')
plt.title('Scatter Plot')
plt.xlabel('comment numbers')
plt.ylabel('liberal/Conservative')
plt.show()
```





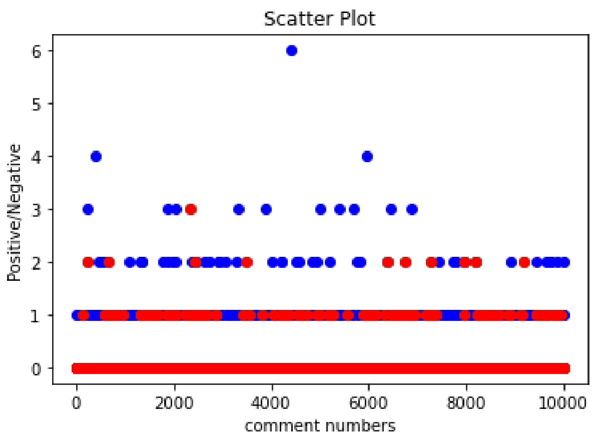
positive/negatve sentiment

In [223]:

```
df['positive1'] = df.CleanText.str.count('good')
df['positive2']= df.CleanText.str.count('special')
df['negative'] = df.CleanText.str.count('bad')
df['TotScore'] = df.positive1 + df.positive2 - df.negative
```

In [224]:

```
plt.scatter(range(0,df['positive1'].size),df['positive1'],c='blue')
plt.scatter(range(0,df['positive2'].size),df['positive2'],c='green')
plt.scatter(range(0,df['negative'].size),df['positive2'],c='red')
plt.title('Scatter Plot')
plt.xlabel('comment numbers')
plt.ylabel('Positive/Negative')
plt.show()
```



In [ ]: