Introduction

The purpose of this notebook is running K-Means clustering to see if the algorithm can sucessfully cluster the news in to 'Real' & 'Fake' using just the words in the articles

Imports

In [1]:

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

import matplotlib.pyplot as plt #  plotting and data visualization
import seaborn as sns # improve visuals
sns.set() # Set as default style

import string # python library
import re # regex library

from gensim.parsing.preprocessing import preprocess_string, strip_tags, strip_punctuati
from gensim.models import Word2Vec # Word2vec

from sklearn import cluster # Kmeans clustering
from sklearn import metrics # Metrics for evaluation
from sklearn.decomposition import PCA #PCA
from sklearn.manifold import TSNE #TSNE
```

Data Analysis & Cleanup

In [2]:

```
fake = pd.read_csv('datasets/Fake.csv')
true = pd.read_csv('datasets/True.csv')
```

In [6]:

```
fake.sample(10)
```

Out[6]:

	title	text	subject	date
6274	Trump Claims Hillary Was Involved In Disappea...	Giddy with the news that an Egyptian commercia...	News	May 20, 2016
19248	WOMAN Who Wants To Become DNC Chair: “My job i...	The Democratic Party held a forum for race obs...	left-news	Jan 24, 2017
20862	WATCH: HARVARD STUDENTS Caught On Tape Saying ...	Is there a worse crime than being a white male...	left-news	Mar 16, 2016
21202	EPIC BACKFIRE: The Left Makes Video Warning Fo...	Does anyone remember a time in recent history ...	left-news	Dec 20, 2015
17977	When The View’s WHOOPI GOLDBERG Told Hillary W...	The sympathetic (and borderline communist) wom...	left-news	Sep 14, 2017
14321	The BRUTAL Truth About Why Kids Love Bernie Sa...	At some point, most of us grow up and realize...	politics	Mar 12, 2016
20998	HERE WE GO AGAIN...GRAMMY AWARDS UNDER FIRE For ...	All of a sudden #BlackDeathsMatter Could the G...	left-news	Feb 15, 2016

	title	text	subject	date
17705	APOLOGY ISSUED After LA TIMES and NY TIMES Col...	Wow! What a couple of hypocrites and haters! W...	left-news	Nov 3, 2017
22901	Why Not A Probe of 'Israel-Gate'	21st Century Wire says Investigative reporter ...	Middle-east	April 23, 2017
4261	BREAKING: Trump's Taj Mahal SHUTS DOWN, Thous...	Donald Trump called the Atlantic City casino t...	News	October 10, 2016

In [5]:

```
true.sample(10)
```

Out[5]:

	title	text	subject	date
13223	With new plan, Swiss pin anti-extremism hopes ...	ZURICH (Reuters) - Switzerland released a nati...	worldnews	December 4, 2017
3361	Alaska governor urges budget compromise to avo...	(Reuters) - Alaska Governor Bill Walker urged ...	politicsNews	June 7, 2017
9370	Ohio appeals U.S. court decision in favor of e...	CLEVELAND (Reuters) - The state of Ohio filed ...	politicsNews	May 27, 2016
13056	Three charged in Malta for murder of anti-corr...	VALLETTA (Reuters) - A magistrate on Tuesday c...	worldnews	December 5, 2017
2306	U.S. Vice President Pence's hawkish tone on Ru...	WASHINGTON (Reuters) - When President Donald T...	politicsNews	August 4, 2017
18001	Kenya police use teargas, shoot in air during ...	NAIROBI (Reuters) - Kenyan police fired tearga...	worldnews	October 9, 2017
6411	For Russia, U.S. election meddling claims stri...	MOSCOW (Reuters) - The Kremlin says U.S. intel...	politicsNews	January 11, 2017
4750	Nunes apologized to Democrats after surveillan...	WASHINGTON (Reuters) - The Republican chairman...	politicsNews	March 23, 2017
15147	Self-designed homes could provide sustainable ...	LONDON (Thomson Reuters Foundation) - Self-des...	worldnews	November 10, 2017
14665	Failure of German coalition talks would streng...	BERLIN (Reuters) - The European Union s budget...	worldnews	November 16, 2017

The first issue as seen above is that the True data contains:

1. A reuters disclaimer that the article is a tweet

> "The following statements were posted to the verified Twitter accounts of U.S. President Donald Trump, @realDonaldTrump and @POTUS. The opinions expressed are his own. Reuters has not edited the statements or confirmed their accuracy. @realDonaldTrump"

1. City Name and then publisher at the start

> WASHINGTON (Reuters)

so in the next block of code I remove this from the data

In [7]:

```
# The following is a crude way to remove the @realDonaldTrump tweet disclaimer and Stat

cleansed_data = []
for data in true.text:
    if "@realDonaldTrump : - " in data:
        cleansed_data.append(data.split("@realDonaldTrump : - ")[1])
    elif "(Reuters) -" in data:
        cleansed_data.append(data.split("(Reuters) - ")[1])
    else:
        cleansed_data.append(data)

true["text"] = cleansed_data
true.head(10)
```

Out[7]:

	title	text	subject	date
0	As U.S. budget fight looms, Republicans flip t...	The head of a conservative Republican faction ...	politicsNews	December 31, 2017
1	U.S. military to accept transgender recruits o...	Transgender people will be allowed for the fir...	politicsNews	December 29, 2017
2	Senior U.S. Republican senator: 'Let Mr. Muell...	The special counsel investigation of links bet...	politicsNews	December 31, 2017
3	FBI Russia probe helped by Australian diplomat...	Trump campaign adviser George Papadopoulos tol...	politicsNews	December 30, 2017
4	Trump wants Postal Service to charge 'much mor...	President Donald Trump called on the U.S. Post...	politicsNews	December 29, 2017
5	White House, Congress prepare for talks on spe...	The White House said on Friday it was set to k...	politicsNews	December 29, 2017
6	Trump says Russia probe will be fair, but time...	President Donald Trump said on Thursday he bel...	politicsNews	December 29, 2017
7	Factbox: Trump on Twitter (Dec 29) - Approval ...	While the Fake News loves to talk about my so-...	politicsNews	December 29, 2017
8	Trump on Twitter (Dec 28) - Global Warming	Together, we are MAKING AMERICA GREAT AGAIN! b...	politicsNews	December 29, 2017
9	Alabama official to certify Senator-elect Jone...	Alabama Secretary of State John Merrill said h...	politicsNews	December 28, 2017

In [8]:

```
true.text[7]
```

Out[8]:

```
'While the Fake News loves to talk about my so-called low approval rating, @foxandfriend
s just showed that my rating on Dec. 28, 2017, was approximately the same as President O
bama on Dec. 28, 2009, which was 47%...and this despite massive negative Trump coverage
& Russia hoax! [0746 EST] - Why is the United States Post Office, which is losing many b
illions of dollars a year, while charging Amazon and others so little to deliver their p
ackages, making Amazon richer and the Post Office dumber and poorer? Should be charging
MUCH MORE! [0804 EST] -- Source link: (bit.ly/2jBh4LU) (bit.ly/2jpEXYR) '
```

Some of the text still contains various characters/words such as:

1. Links
2. Timestamps
3. Brackets

4. Numbers

So we will be removing all such characters from the real and fake data using genlib preprocessing and a custom regex for the links in preperation for the Word2Vec

Before that however, the title and text will be merged in to one so that it can all be preprocessed together. I will also add a label for real and fake which will be used later to evaluate our clustering

In [9]:

```python
# Merging title and text
fake['Sentences'] = fake['title'] + ' ' + fake['text']
true['Sentences'] = true['title'] + ' ' + true['text']

# Adding fake and true label
fake['Label'] = 0
true['Label'] = 1

# We can merge both together since we now have labels
final_data = pd.concat([fake, true])

# Randomize the rows so its all mixed up
final_data = final_data.sample(frac=1).reset_index(drop=True)

# Drop columns not needed
final_data = final_data.drop(['title', 'text', 'subject', 'date'], axis = 1)

final_data.head(10)
```

Out[9]:

	Sentences	Label
0	Obama Teams Up With Stephen Colbert For Bruta...	0
1	UKIP leader and Brexit figurehead Farage congr...	1
2	North Korea's Kim says will make 'deranged' Tr...	1
3	‘ANGEL’ MOMS of Sons Killed by Illegals Weigh ...	0
4	THE SMARTEST WOMAN In Politics: “How Trump Can...	0
5	NUT JOB GLENN BECK Joins Liberal, Foul-Mouthed...	0
6	It’s On: GOP Chairman Just Declared WAR On Tr...	0
7	White Las Vegas Teen Caught On Camera Calling...	0
8	[Video] DUMB AND DUMBER Star BASHES TRUMP...Use ...	0
9	Breitbart Caught Praising Melania For Exact T...	0

In [10]:

```python
# Here we preprocess the sentences
def remove_URL(s):
    regex = re.compile(r'https?://\S+|www\.\S+|bit\.ly\S+')
    return regex.sub(r'',s)

# Preprocessing functions to remove lowercase, links, whitespace, tags, numbers, punctu
CUSTOM_FILTERS = [lambda x: x.lower(), strip_tags, remove_URL, strip_punctuation, strip

# Here we store the processed sentences and their label
processed_data = []
processed_labels = []
```

```
for index, row in final_data.iterrows():
    words_broken_up = preprocess_string(row['Sentences'], CUSTOM_FILTERS)
    # This eliminates any fields that may be blank after preprocessing
    if len(words_broken_up) > 0:
        processed_data.append(words_broken_up)
        processed_labels.append(row['Label'])
```

Word2Vec

In [11]:

```
# Word2Vec model trained on processed data
model = Word2Vec(processed_data, min_count=1)
```

In [12]:

```
model.wv.most_similar("country")
```

Out[12]:

```
[('nation', 0.8102749586105347),
 ('america', 0.6123671531677246),
 ('countries', 0.569633960723877),
 ('europe', 0.5472884178161621),
 ('realize', 0.5157971382141113),
 ('world', 0.5116060972213745),
 ('especially', 0.4979727566242218),
 ('path', 0.47543010115623474),
 ('germany', 0.4726695418357849),
 ('american', 0.4709787666797638)]
```

Sentence Vectors

In [13]:

```
# Getting the vector of a sentence based on average of all the word vectors in the sent
# We get the average as this accounts for different sentence lengths

def ReturnVector(x):
    try:
        return model[x]
    except:
        return np.zeros(100)

def Sentence_Vector(sentence):
    word_vectors = list(map(lambda x: ReturnVector(x), sentence))
    return np.average(word_vectors, axis=0).tolist()

X = []
for data_x in processed_data:
    X.append(Sentence_Vector(data_x))
```

```
C:\Users\jarc1\anaconda3\envs\env1\lib\site-packages\ipykernel_launcher.py:6: Deprecatio
nWarning: Call to deprecated `__getitem__` (Method will be removed in 4.0.0, use self.w
v.__getitem__() instead).
```

In [14]:

```
X_np = np.array(X)
X_np.shape
```

Out[14]:

```
(44889, 100)
```

Clustering

In [15]:

```
# Training for 2 clusters (Fake and Real)
kmeans = cluster.KMeans(n_clusters=2, verbose=1)

# Fit predict will return labels
clustered = kmeans.fit_predict(X_np)
```

```
Initialization complete
Iteration 0, inertia 727520.8060985795
Iteration 1, inertia 444601.8157839019
Iteration 2, inertia 444198.8749485893
Iteration 3, inertia 444106.738198857
Iteration 4, inertia 444067.5320110095
Iteration 5, inertia 444054.22201743576
Iteration 6, inertia 444050.0758152132
Iteration 7, inertia 444048.3735076934
Iteration 8, inertia 444047.6695262135
Converged at iteration 8: center shift 5.964693513133669e-06 within tolerance 1.18251043
26848953e-05
Initialization complete
Iteration 0, inertia 694873.9455271196
Iteration 1, inertia 466283.9901389101
Iteration 2, inertia 451117.0384593388
Iteration 3, inertia 445947.3051466154
Iteration 4, inertia 444568.9869460391
Iteration 5, inertia 444203.58595757995
Iteration 6, inertia 444092.7933642296
Iteration 7, inertia 444058.1104155726
Iteration 8, inertia 444050.10025231686
Iteration 9, inertia 444048.4392757061
Iteration 10, inertia 444047.92904216115
Converged at iteration 10: center shift 7.793526651544989e-06 within tolerance 1.1825104
326848953e-05
Initialization complete
Iteration 0, inertia 757542.5909594673
Iteration 1, inertia 500960.3897877762
Iteration 2, inertia 489404.15202963643
Iteration 3, inertia 471582.78791516955
Iteration 4, inertia 453788.0365409462
Iteration 5, inertia 445939.8254213157
Iteration 6, inertia 444356.7108716441
Iteration 7, inertia 444093.7112766059
Iteration 8, inertia 444054.45098416007
Iteration 9, inertia 444048.53459025116
Iteration 10, inertia 444047.55378521356
Converged at iteration 10: center shift 6.842897736040156e-06 within tolerance 1.1825104
326848953e-05
Initialization complete
Iteration 0, inertia 706802.4841913175
Iteration 1, inertia 455130.114407173
Iteration 2, inertia 446528.57087533834
Iteration 3, inertia 444511.805476197
Iteration 4, inertia 444130.7786164206
Iteration 5, inertia 444065.88353173743
Iteration 6, inertia 444051.9301885025
Iteration 7, inertia 444048.7580362772
Iteration 8, inertia 444047.9529572176
Converged at iteration 8: center shift 7.73180636005463e-06 within tolerance 1.182510432
6848953e-05
Initialization complete
Iteration 0, inertia 776131.8069592491
Iteration 1, inertia 503587.3565554795
Iteration 2, inertia 470636.25547178736
Iteration 3, inertia 450869.7340578809
Iteration 4, inertia 445824.51207150216
Iteration 5, inertia 444523.96884628106
Iteration 6, inertia 444174.08507875673
Iteration 7, inertia 444086.3568809182
Iteration 8, inertia 444057.80586423853
Iteration 9, inertia 444050.17508279416
Iteration 10, inertia 444048.42360346665
Iteration 11, inertia 444047.8803475337
Converged at iteration 11: center shift 8.425743016426378e-06 within tolerance 1.1825104
326848953e-05
Initialization complete
Iteration 0, inertia 629955.669630441
Iteration 1, inertia 460087.6933726588
Iteration 2, inertia 448591.87819754495
Iteration 3, inertia 445468.4532904881
Iteration 4, inertia 444487.7582349789
Iteration 5, inertia 444186.5514097214
Iteration 6, inertia 444090.49265178625
Iteration 7, inertia 444059.4536707896
Iteration 8, inertia 444050.5434482273
Iteration 9, inertia 444048.6265348758
Iteration 10, inertia 444047.96836820233
Converged at iteration 10: center shift 7.857622325417152e-06 within tolerance 1.1825104
326848953e-05
Initialization complete
Iteration 0, inertia 761717.9433706887
Iteration 1, inertia 481599.5280748931
Iteration 2, inertia 457480.7212991917
Iteration 3, inertia 446582.1376167262
Iteration 4, inertia 444512.84721470485
Iteration 5, inertia 444136.3203244514
Iteration 6, inertia 444062.8061759696
Iteration 7, inertia 444050.914613307
Iteration 8, inertia 444048.2480113205
Iteration 9, inertia 444047.5858600996
Converged at iteration 9: center shift 6.760486887630382e-06 within tolerance 1.18251043
26848953e-05
Initialization complete
Iteration 0, inertia 615445.5426242244
Iteration 1, inertia 449988.5518006085
Iteration 2, inertia 445127.3283514259
Iteration 3, inertia 444292.5436224275
Iteration 4, inertia 444105.79794715
Iteration 5, inertia 444058.99598001945
Iteration 6, inertia 444049.88631766214
Iteration 7, inertia 444048.1251358142
Iteration 8, inertia 444047.68068915297
Converged at iteration 8: center shift 4.44105149427385e-06 within tolerance 1.182510432
6848953e-05
Initialization complete
Iteration 0, inertia 642932.2559112719
Iteration 1, inertia 449843.553911688
Iteration 2, inertia 446498.31440292107
Iteration 3, inertia 445174.11683985236
Iteration 4, inertia 444500.88250332937
Iteration 5, inertia 444213.48230567476
Iteration 6, inertia 444110.9038201898
Iteration 7, inertia 444068.71698610927
Iteration 8, inertia 444054.50409674476
Iteration 9, inertia 444050.1356362099
Iteration 10, inertia 444048.3995684934
Iteration 11, inertia 444047.6695262135
Converged at iteration 11: center shift 5.964693513134081e-06 within tolerance 1.1825104
326848953e-05
Initialization complete
Iteration 0, inertia 744727.3176324257
Iteration 1, inertia 497710.6752704391
Iteration 2, inertia 458388.1287723366
Iteration 3, inertia 446723.3371366283
Iteration 4, inertia 444650.4787381526
Iteration 5, inertia 444185.0122508677
Iteration 6, inertia 444085.45720773377
Iteration 7, inertia 444057.59115661756
Iteration 8, inertia 444050.2403420266
Iteration 9, inertia 444048.40410434344
Iteration 10, inertia 444047.8434296043
Converged at iteration 10: center shift 7.948656148169681e-06 within tolerance 1.1825104
326848953e-05
```

In [16]:

```
testing_df = {'Sentence': processed_data, 'Labels': processed_labels, 'Prediction': clu
testing_df = pd.DataFrame(data=testing_df)

testing_df.head(10)
```

Out[16]:

	Sentence	Labels	Prediction
0	[obama, teams, stephen, colbert, brutal, trump...	0	0
1	[ukip, leader, brexit, figurehead, farage, con...	1	0
2	[north, korea, kim, says, deranged, trump, pay...	1	1
3	[‘angel’, moms, sons, killed, illegals, weigh,...	0	0
4	[smartest, woman, politics, “how, trump, knock...	0	0
5	[nut, job, glenn, beck, joins, liberal, foul, ...	0	0
6	[it’s, gop, chairman, declared, war, trump, tw...	0	0
7	[white, las, vegas, teen, caught, camera, call...	0	0
8	[video, dumb, dumber, star, bashes, trump...use,...	0	0
9	[breitbart, caught, praising, melania, exact, ...	0	0

The results above show that its correctly clustered them in some cases where 0 is fake news and 1 is real news

In [17]:

```
correct = 0
incorrect = 0
for index, row in testing_df.iterrows():
    if row['Labels'] == row['Prediction']:
        correct += 1
    else:
        incorrect += 1

print("Correctly clustered news: " + str((correct*100)/(correct+incorrect)) + "%")
```

```
Correctly clustered news: 87.442357815946%
```

Visualization

In [18]:

```
# PCA of sentence vectors
pca = PCA(n_components=2)
pca_result = pca.fit_transform(X_np)

PCA_df = pd.DataFrame(pca_result)
```

```
PCA_df['cluster'] = clustered
PCA_df.columns = ['x1','x2','cluster']
```

In [19]:

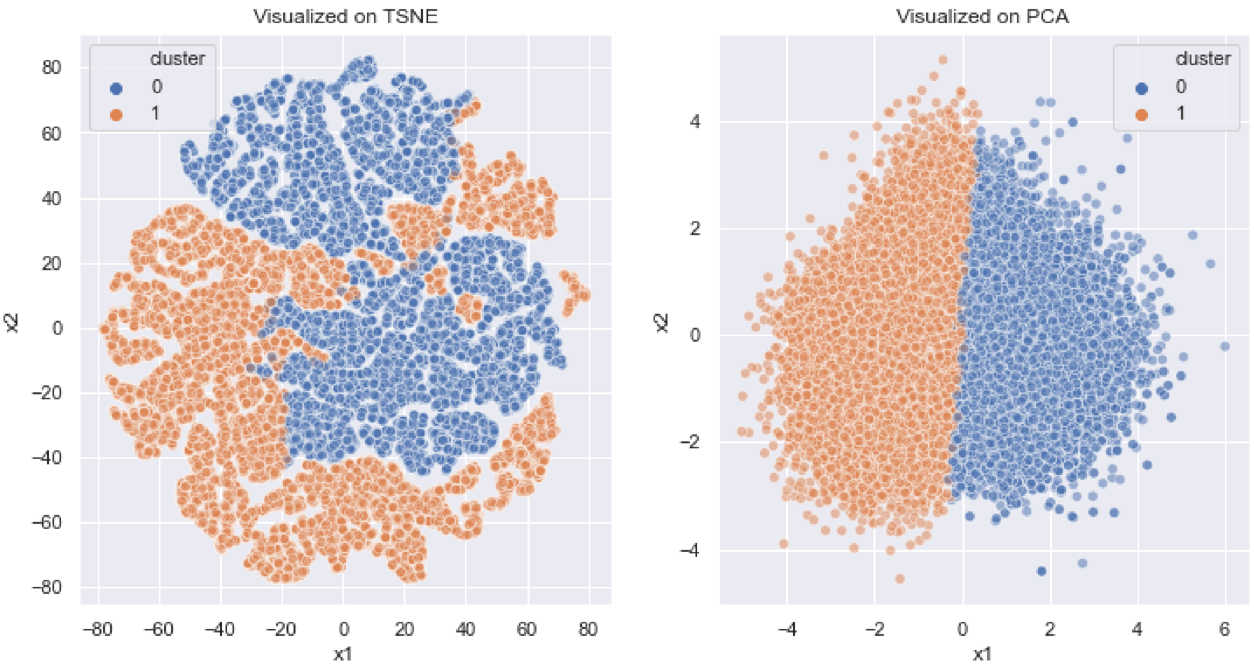
```
# T-SNE
tsne = TSNE(n_components=2)
tsne_result = tsne.fit_transform(pca_result)

TSNE_df = pd.DataFrame(tsne_result)
TSNE_df['cluster'] = clustered
TSNE_df.columns = ['x1','x2','cluster']
```

In [20]:

```
# Plots
fig, ax = plt.subplots(1, 2, figsize=(12,6))
sns.scatterplot(data=PCA_df,x='x1',y='x2',hue='cluster',legend="full",alpha=0.5,ax=ax[1
sns.scatterplot(data=TSNE_df,x='x1',y='x2',hue='cluster',legend="full",alpha=0.5,ax=ax[
ax[0].set_title('Visualized on TSNE')
ax[1].set_title('Visualized on PCA')
```

Out[20]:

```
Text(0.5, 1.0, 'Visualized on PCA')
```



Custom News Tests

In [21]:

```
# Testing with fake news generated from https://www.thefakenewsgenerator.com/
onion_data = "Flint Residents Learn To Harness Superpowers, But Trump Gets Away Again T

# Preprocess article
onion_data = preprocess_string(onion_data, CUSTOM_FILTERS)

# Get sentence vector
onion_data = Sentence_Vector(onion_data)

# Get prediction
kmeans.predict(np.array([onion_data]))
```

```
C:\Users\jarc1\anaconda3\envs\env1\lib\site-packages\ipykernel_launcher.py:6: Deprecatio
nWarning: Call to deprecated `__getitem__` (Method will be removed in 4.0.0, use self.w
v.__getitem__() instead).
```

Out[21]:

```
array([0])
```

In [22]:

```
# News from BBC

bbc_data = "Nasa Mars 2020 Mission's MiMi Aung on women in space Next year, Nasa will s

# Preprocess article
bbc_data = preprocess_string(bbc_data, CUSTOM_FILTERS)

# Get sentence vector
bbc_data = Sentence_Vector(bbc_data)

# Get prediction
kmeans.predict(np.array([bbc_data]))
```

```
C:\Users\jarc1\anaconda3\envs\env1\lib\site-packages\ipykernel_launcher.py:6: Deprecatio
nWarning: Call to deprecated `__getitem__` (Method will be removed in 4.0.0, use self.w
v.__getitem__() instead).
```

Out[22]:

```
array([1])
```

In [ ]: