Non-Negative Matrix Factorization

In [1]:
```
import pandas as pd
```

In [2]:
```
npr = pd.read_csv('npr.csv')
```

In [3]:
```
npr.head()
```

Out[3]:

	Article
0	In the Washington of 2016, even when the polic...
1	Donald Trump has used Twitter — his prefe...
2	Donald Trump is unabashedly praising Russian...
3	Updated at 2:50 p. m. ET, Russian President Vl...
4	From photography, illustration and video, to d...

Preprocessing

In [5]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
```

In [6]:
```
tfidf = TfidfVectorizer(max_df = 0.95, min_df= 2, stop_words = 'english'
```

In [8]:
```
dtm = tfidf.fit_transform(npr['Article'])
```

In [9]:
```
dtm
```

Out[9]:
```
<11992x54777 sparse matrix of type '<class 'numpy.float64'>'
	with 3033388 stored elements in Compressed Sparse Row format>
```

NMF

In [11]:
```
from sklearn.decomposition import NMF
```

In [12]:
```
nmf_model = NMF(n_components = 7, random_state= 42)
```

In [13]:
```
nmf_model.fit(dtm)
```

Out[13]:
```
NMF(alpha=0.0, beta_loss='frobenius', init=None, l1_ratio=0.0, max_ite
r=200,
  n_components=7, random_state=42, shuffle=False, solver='cd', tol=0.0
001,
  verbose=0)
```

Displaying Topics

In [16]:
```
tfidf.get_feature_names()[50000]
```

Out[16]:
```
'transcribe'
```

In [17]:
```
len(tfidf.get_feature_names())
```

Out[17]:
```
54777
```

In [18]:
```
# Get the top 15 words for each topic
for index, topic in enumerate(nmf_model.components_):
    print(f"The TOP 15 words for TOPIC# {index}")
    print([tfidf.get_feature_names()[i] for i in topic.argsort()[-15:]])
    print('\n\n')
```

```
The TOP 15 words for TOPIC# 0
['new', 'research', 'like', 'patients', 'health', 'disease', 'percent'
, 'women', 'virus', 'study', 'water', 'food', 'people', 'zika', 'says'
]



The TOP 15 words for TOPIC# 1
['gop', 'pence', 'presidential', 'russia', 'administration', 'election
', 'republican', 'obama', 'white', 'house', 'donald', 'campaign', 'sai
d', 'president', 'trump']



The TOP 15 words for TOPIC# 2
['senate', 'house', 'people', 'act', 'law', 'tax', 'plan', 'republican
s', 'affordable', 'obamacare', 'coverage', 'medicaid', 'insurance', 'c
are', 'health']



The TOP 15 words for TOPIC# 3
['officers', 'syria', 'security', 'department', 'law', 'isis', 'russia
', 'government', 'state', 'attack', 'president', 'reports', 'court', '
said', 'police']



The TOP 15 words for TOPIC# 4
['primary', 'cruz', 'election', 'democrats', 'percent', 'party', 'dele
gates', 'vote', 'state', 'democratic', 'hillary', 'campaign', 'voters'
, 'sanders', 'clinton']



The TOP 15 words for TOPIC# 5
['love', 've', 'don', 'album', 'way', 'time', 'song', 'life', 'really'
, 'know', 'people', 'think', 'just', 'music', 'like']



The TOP 15 words for TOPIC# 6
['teacher', 'state', 'high', 'says', 'parents', 'devos', 'children', '
college', 'kids', 'teachers', 'student', 'education', 'schools', 'scho
ol', 'students']
```

Attach the topic labels that we discovered to the documents

In [19]:
```
topic_results = nmf_model.transform(dtm)
```

In [20]:
```
topic_results[0]
```

Out[20]:
```
array([0.        , 0.12075603, 0.00140297, 0.05919954, 0.01518909,
       0.        , 0.        ])
```

The values in the arrays are coefficients and we want the index of the highest coefficient

In [21]:
```
topic_results[0].argmax()
```

Out[21]:
```
1
```

In [22]:
```
## We will apply .argmax() to topic_results column
```

In [23]:
```
npr['Topic'] = topic_results.argmax(axis = 1)
```

In [24]:
```
npr.head()
```

Out[24]:

	Article	Topic
0	In the Washington of 2016, even when the polic...	1
1	Donald Trump has used Twitter — his prefe...	1
2	Donald Trump is unabashedly praising Russian...	1
3	Updated at 2:50 p. m. ET, Russian President Vl...	3
4	From photography, illustration and video, to d...	6

In [26]:
```
## Create a mapping of topic numbers to topic labels

mytopic_dict = {0:'health', 1: 'election', 2: 'legis', 3: 'politics',
                4: 'election', 5:'music', 6: 'education'}

npr['Topic_label'] = npr['Topic'].map(mytopic_dict)
```

In [27]:
```
npr.head(10)
```

Out[27]:

	Article	Topic	Topic_label
0	In the Washington of 2016, even when the polic...	1	election
1	Donald Trump has used Twitter — his prefe...	1	election
2	Donald Trump is unabashedly praising Russian...	1	election
3	Updated at 2:50 p. m. ET, Russian President Vl...	3	politics
4	From photography, illustration and video, to d...	6	education
5	I did not want to join yoga class. I hated tho...	5	music
6	With a who has publicly supported the debunk...	0	health
7	I was standing by the airport exit, debating w...	0	health
8	If movies were trying to be more realistic, pe...	0	health
9	Eighteen years ago, on New Year's Eve, David F...	5	music

In [ ]:
```

```