```
import pandas as pd
import numpy as np


import matplotlib.pyplot as plt
import seaborn as sb
import plotly.figure_factory as ff
import plotly.graph_objects as go
import plotly.express as px

import os
import math
import time
import re

from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer


from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans


from sklearn.metrics.pairwise import cosine_similarity
from sklearn.metrics import pairwise_distances


news_art = pd.read_json("/content/News_Category_Dataset_v3.json", lines=True)
print(news_art)
```

```
                                                     link  \
0       https://www.huffpost.com/entry/covid-boosters-...
1       https://www.huffpost.com/entry/american-airlin...
2       https://www.huffpost.com/entry/funniest-tweets...
3       https://www.huffpost.com/entry/funniest-parent...
4       https://www.huffpost.com/entry/amy-cooper-lose...
...                                                   ...
209522  https://www.huffingtonpost.com/entry/rim-ceo-t...
209523  https://www.huffingtonpost.com/entry/maria-sha...
209524  https://www.huffingtonpost.com/entry/super-bow...
209525  https://www.huffingtonpost.com/entry/aldon-smi...
209526  https://www.huffingtonpost.com/entry/dwight-ho...

                                                 headline   category  \
0       Over 4 Million Americans Roll Up Sleeves For O...  U.S. NEWS
1       American Airlines Flyer Charged, Banned For Li...  U.S. NEWS
2       23 Of The Funniest Tweets About Cats And Dogs ...     COMEDY
3       The Funniest Tweets From Parents This Week (Se...  PARENTING
4       Woman Who Called Cops On Black Bird-Watcher Lo...  U.S. NEWS
...                                                   ...        ...
209522  RIM CEO Thorsten Heins' 'Significant' Plans Fo...       TECH
209523  Maria Sharapova Stunned By Victoria Azarenka I...     SPORTS
209524  Giants Over Patriots, Jets Over Colts Among  M...     SPORTS
209525  Aldon Smith Arrested: 49ers Linebacker Busted ...     SPORTS
209526  Dwight Howard Rips Teammates After Magic Loss ...     SPORTS

                                        short_description  \
0       Health experts said it is too early to predict...
1       He was subdued by passengers and crew when he ...
2       "Until you have a dog you don't understand wha...
3       "Accidentally put grown-up toothpaste on my to...
4       Amy Cooper accused investment firm Franklin Te...
...                                                   ...
209522  Verizon Wireless and AT&T are already promotin...
209523  Afterward, Azarenka, more effusive with the pr...
209524  Leading up to Super Bowl XLVI, the most talked...
209525  CORRECTION: An earlier version of this story i...
209526  The five-time all-star center tore into his te...

                     authors       date
0       Carla K. Johnson, AP 2022-09-23
1             Mary Papenfuss 2022-09-23
2              Elyse Wanshel 2022-09-23
3           Caroline Bologna 2022-09-23
4             Nina Golgowski 2022-09-22
...                      ...        ...
209522      Reuters, Reuters 2012-01-28
```

```
209523                    2012-01-28
209524                    2012-01-28
209525                    2012-01-28
209526                    2012-01-28

[209527 rows x 6 columns]
```

```
news_art.head()
```

	link	headline	category	short_description	authors	date
0	https://www.huffpost.com/entry/covid-boosters-...	Over 4 Million Americans Roll Up Sleeves For O...	U.S. NEWS	Health experts said it is too early to predict...	Carla K. Johnson, AP	2022-09-23
1	https://www.huffpost.com/entry/american-airlin...	American Airlines Flyer Charged, Banned For Li...	U.S. NEWS	He was subdued by passengers and crew when he ...	Mary Papenfuss	2022-09-23
2	https://www.huffpost.com/entry/funniest-tweets...	23 Of The Funniest Tweets About Cats And Dogs ...	COMEDY	"Until you have a dog you don't understand wha...	Elyse Wanshel	2022-09-23

```
news_art.tail()
```

	link	headline	category	short_description	authors	date
209522	https://www.huffingtonpost.com/entry/rim-ceo-t...	RIM CEO Thorsten Heins' 'Significant' Plans Fo...	TECH	Verizon Wireless and AT&T are already promotin...	Reuters, Reuters	2012-01-28
209523	https://www.huffingtonpost.com/entry/maria-sha...	Maria Sharapova Stunned By Victoria Azarenka I...	SPORTS	Afterward, Azarenka, more effusive with the pr...		2012-01-28
209524	https://www.huffingtonpost.com/entry/super-bow...	Giants Over Patriots, Jets Over Colts Among M...	SPORTS	Leading up to Super Bowl XLVI, the most talked...		2012-01-28

```
news_art.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 209527 entries, 0 to 209526
Data columns (total 6 columns):
 #   Column             Non-Null Count   Dtype
---  ------             --------------   -----
 0   link               209527 non-null  object
 1   headline           209527 non-null  object
 2   category           209527 non-null  object
 3   short_description  209527 non-null  object
 4   authors            209527 non-null  object
 5   date               209527 non-null  datetime64[ns]
dtypes: datetime64[ns](1), object(5)
memory usage: 9.6+ MB
```

Data Preprocessing

```
news_art = news_art[news_art['date'] >= pd.Timestamp(2018,1,1)] # Use pd.Timestamp instead of pd.timestamp
```

```
news_art.shape
```

```
(17257, 6)
```

```
news_art.isna()
```

	link	headline	category	short_description	authors	date
0	False	False	False	False	False	False
1	False	False	False	False	False	False
2	False	False	False	False	False	False
3	False	False	False	False	False	False
4	False	False	False	False	False	False
...	...	...	...	...	...	...
17252	False	False	False	False	False	False
17253	False	False	False	False	False	False
17254	False	False	False	False	False	False
17255	False	False	False	False	False	False
17256	False	False	False	False	False	False

17257 rows × 6 columns

```
news_art.isna().sum()
```

	0
link	0
headline	0
category	0
short_description	0
authors	0
date	0

dtype: int64

```
news_art = news_art[news_art['headline'].apply(lambda x: len(x.split())>5)]
print("Total number of articles after removal of headlines with short title:", news_art.shape[0])
```

```
Total number of articles after removal of headlines with short title: 17183
```

```
category_column = news_art['category']



unique_categories = category_column.unique()


num_unique_categories = len(unique_categories)
print("Number of unique categories:", num_unique_categories)
```

```
Number of unique categories: 36
```

```
print(unique_categories)
```

```
['U.S. NEWS' 'COMEDY' 'PARENTING' 'WORLD NEWS' 'CULTURE & ARTS' 'TECH'
 'SPORTS' 'ENTERTAINMENT' 'POLITICS' 'WEIRD NEWS' 'ENVIRONMENT'
 'EDUCATION' 'CRIME' 'SCIENCE' 'WELLNESS' 'BUSINESS' 'STYLE & BEAUTY'
 'FOOD & DRINK' 'MEDIA' 'QUEER VOICES' 'HOME & LIVING' 'WOMEN'
 'BLACK VOICES' 'TRAVEL' 'MONEY' 'RELIGION' 'LATINO VOICES' 'IMPACT'
 'WEDDINGS' 'COLLEGE' 'PARENTS' 'ARTS & CULTURE' 'STYLE' 'GREEN' 'TASTE'
 'HEALTHY LIVING']
```

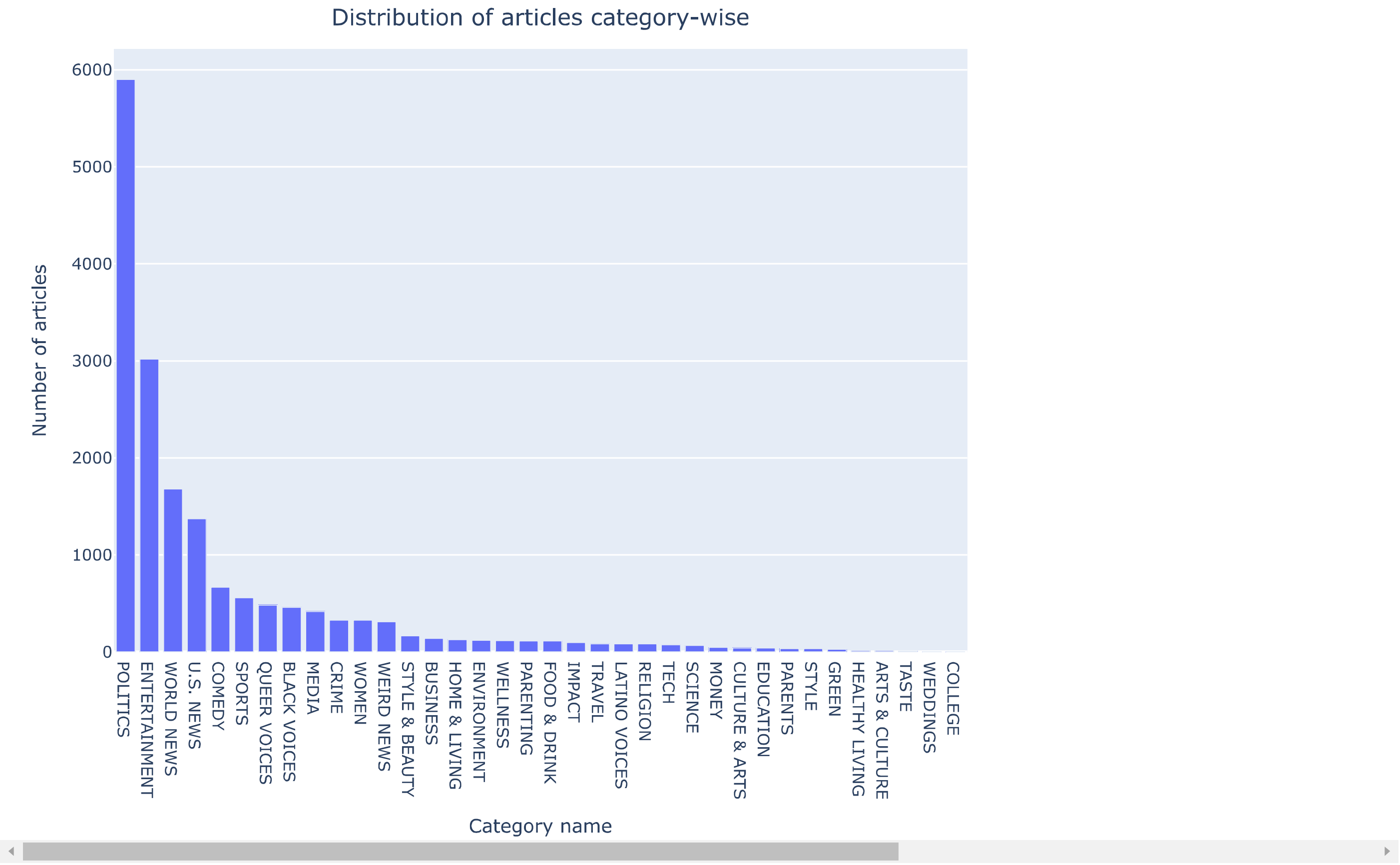
```
news_art.drop_duplicates(inplace = True)
news_art.shape
```

```
(17183, 6)
```

```
print("Total number of articles : ", news_art.shape[0])
print("Total number of authors : ", news_art["authors"].nunique())
print("Total number of unqiue categories : ", news_art["category"].nunique())
```

```
Total number of articles :  17183
Total number of authors :  2261
Total number of unqiue categories :  36
```

```
fig = go.Figure([go.Bar(x=news_art["category"].value_counts().index, y=news_art["category"].value_counts().values)])
fig['layout'].update(title={"text" : 'Distribution of articles category-wise','y':0.9,'x':0.5,'xanchor': 'center','yanchor': 'top'}, xaxis_t
fig.update_layout(width=800,height=700)
fig
```



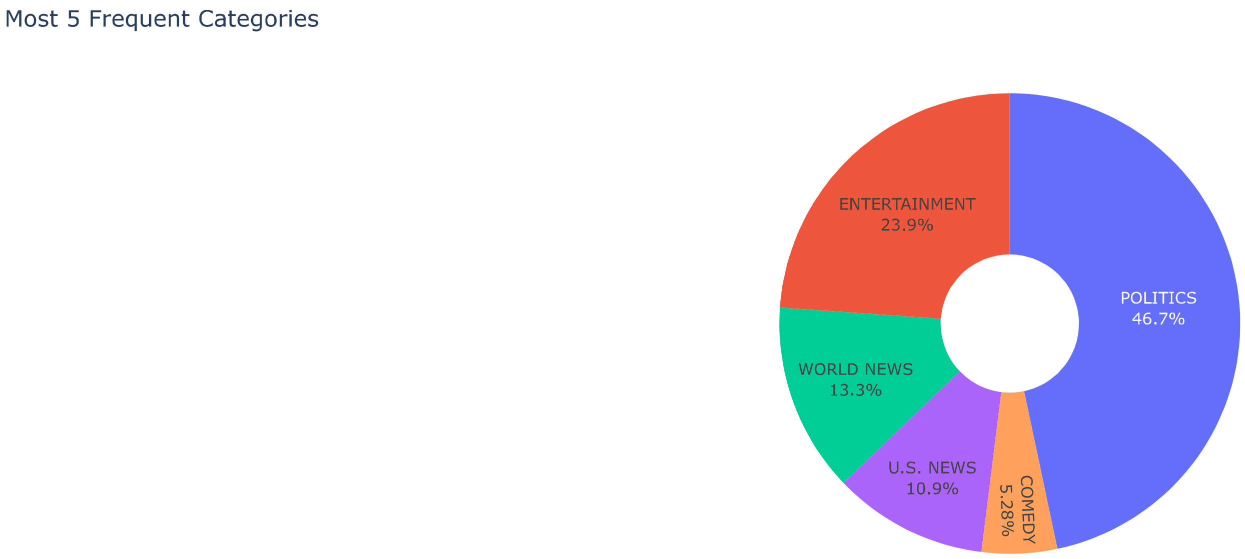
number of articles per month

```
news_articles_per_month = news_art.resample('m',on = 'date')['headline'].count()
news_articles_per_month
```

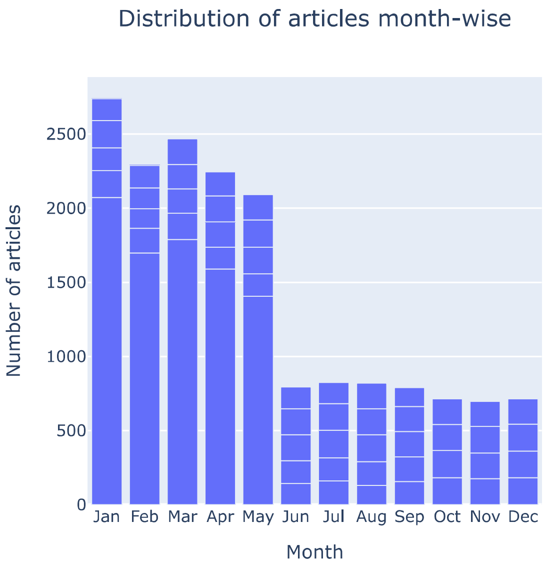
```
<ipython-input-26-12d4c520855b>:1: FutureWarning:

'm' is deprecated and will be removed in a future version, please use 'ME' instead.
```

	headline
date	
2018-01-31	2072
2018-02-28	1697
2018-03-31	1788
2018-04-30	1590
2018-05-31	1406
2018-06-30	143
2018-07-31	160
2018-08-31	130
2018-09-30	157
2018-10-31	182
2018-11-30	175
2018-12-31	181
2019-01-31	181
2019-02-28	168
2019-03-31	179
2019-04-30	147
2019-05-31	152
2019-06-30	153
2019-07-31	157
2019-08-31	161
2019-09-30	165
2019-10-31	183
2019-11-30	174
2019-12-31	181
2020-01-31	154
2020-02-29	132
2020-03-31	163
2020-04-30	170

```
most5_frequent_categories = news_art["category"].value_counts().head(5)

fig = px.pie(values=most5_frequent_categories.values, names=most5_frequent_categories.index,
       labels=most5_frequent_categories.index, title="Most 5 Frequent Categories", hole=.3,)

fig.update_traces(textposition='inside', textinfo='percent+label')
```

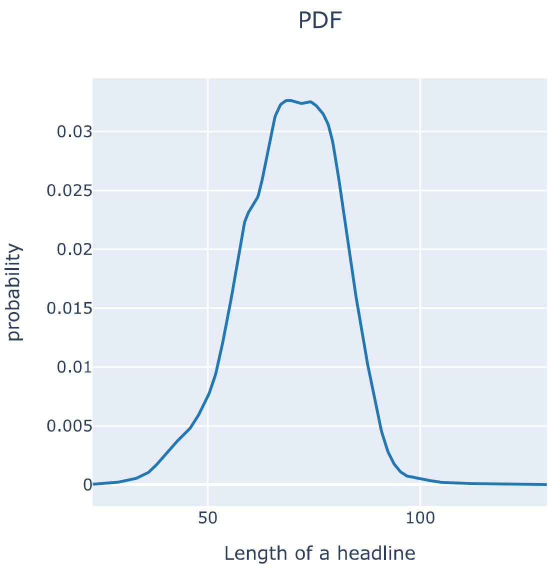


```
fig = go.Figure([go.Bar(x=news_articles_per_month.index.strftime("%b"), y=news_articles_per_month)])
fig['layout'].update(title={"text" : 'Distribution of articles month-wise','y':0.9,'x':0.5,'xanchor': 'center','yanchor': 'top'}, xaxis_titl
fig.update_layout(width=500,height=500)
fig
```



PDF FOR THE length of the headings

```
fig = ff.create_distplot([news_art['headline'].str.len()], ["ht"],show_hist=False,show_rug=False)
fig['layout'].update(title={'text':'PDF','y':0.9,'x':0.5,'xanchor': 'center','yanchor': 'top'}, xaxis_title="Length of a headline",yaxis_tit
fig.update_layout(showlegend = False,width=500,height=500)
fig
```



```
news_art.index = range(news_art.shape[0])
```

```
news_art["day and month"] = news_art["date"].dt.strftime("%a") + "_" + news_art["date"].dt.strftime("%b")
```

```
news_art.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17183 entries, 0 to 17182
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   link               17183 non-null  object
 1   headline           17183 non-null  object
 2   category           17183 non-null  object
 3   short_description  17183 non-null  object
 4   authors            17183 non-null  object
 5   date               17183 non-null  datetime64[ns]
 6   day and month      17183 non-null  object
dtypes: datetime64[ns](1), object(6)
memory usage: 939.8+ KB
```

```
news_art.iloc[10:20]
```

	link	headline	category	short_description	authors	date	day and month
10	https://www.huffpost.com/entry/bc-soc-wcup-cap...	World Cup Captains Want To Wear Rainbow Armban...	WORLD NEWS	FIFA has come under pressure from several Euro...	GRAHAM DUNBAR, AP	2022-09-21	Wed_Sep
11	https://www.huffpost.com/entry/man-sets-fire-p...	Man Sets Himself On Fire In Apparent Protest O...	WORLD NEWS	The incident underscores a growing wave of pro...	Mari Yamaguchi, AP	2022-09-21	Wed_Sep
12	https://www.huffpost.com/entry/fiona-threatens...	Fiona Threatens To Become Category 4 Storm Hea...	WORLD NEWS	Hurricane Fiona lashed the Turks and Caicos Is...	Dánica Coto, AP	2022-09-21	Wed_Sep
13	https://www.huffpost.com/entry/twitch-streamer...	Twitch Bans Gambling Sites After Streamer Scam...	TECH	One man's claims that he scammed people on the...	Ben Blanchet	2022-09-21	Wed_Sep
	https://www.huffpost.com/entry/virginia-	Virginia Thomas		Conservative activist Virginia	Eric Tucker and	2022-	

```
news_art_temp = news_art.copy()
```

Text Preprocessing

```
import nltk
nltk.download('stopwords')
nltk.download('punkt')
nltk.download('wordnet')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
True
```

```
stop_words = set(stopwords.words('english'))
```

```
for i in range(len(news_art_temp["headline"])):
    string = ""
    for word in news_art_temp["headline"][i].split():
        word = ("".join(e for e in word if e.isalnum()))
        word = word.lower()
        if not word in stop_words:
          string += word + " "
    if(i%1000==0):
      print(i)
    news_art_temp.at[i,"headline"] = string.strip()
```

```
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
11000
12000
13000
14000
15000
16000
17000
```

```
news_art = news_art[news_art['headline'].apply(lambda x: len(x.split())>5)]
print("Total number of articles after removal of headlines with short title:", news_art.shape[0])
```

```
Total number of articles after removal of headlines with short title: 17183
```

```
lemmatizer = WordNetLemmatizer()
```

```
for i in range(len(news_art_temp["headline"])):
    string = ""
    for w in word_tokenize(news_art_temp["headline"][i]):
        string += lemmatizer.lemmatize(w,pos = "v") + " "
    news_art_temp.at[i, "headline"] = string.strip()
    if(i%1000==0):
        print(i)
```

```
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
11000
12000
13000
14000
15000
16000
17000
```

```
headline_vectorizer = CountVectorizer()
headline_features   = headline_vectorizer.fit_transform(news_art_temp['headline'])
```

```
headline_features.get_shape()
```

```
(17183, 16483)
```

```
pd.set_option('display.max_colwidth', None)
```

```
def bag_of_words_based_model(row_index, num_similar_items):
    couple_dist = pairwise_distances(headline_features,headline_features[row_index])
    indices = np.argsort(couple_dist.ravel())[0:num_similar_items]
    df = pd.DataFrame({'publish_date': news_art['date'][indices].values,
               'headline':news_art['headline'][indices].values,
                'Euclidean similarity with the queried article': couple_dist[indices].ravel()})
    print("="*30,"Queried article details","="*30)
    print('headline : ',news_art['headline'][indices[0]])
    print("\n","="*25,"Recommended articles : ","="*23)
    #return df.iloc[1:,1]
    return df.iloc[1:,]

bag_of_words_based_model(133, 11)
```

```
============================== Queried article details ==============================
headline :  Stocks Dive For Truth Social SPAC Amid Merger Delay

 ========================= Recommended articles :  =======================
```

	publish_date	headline	Euclidean similarity with the queried article
1	2020-03-18	Solidarity In A Time Of Social Distancing	3.162278
2	2018-04-30	What A Year This Month Has Been	3.162278
3	2020-05-29	What Social Distancing Has Been Like For Only Children	3.162278
4	2018-03-31	What A Year This Month Has Been	3.162278
5	2018-02-07	Everything You Should Know About The Stock Market	3.162278
6	2018-02-21	All They Will Call You Will Be Deportees	3.162278
7	2020-04-13	A Pandemic Is Not A War	3.162278
8	2021-10-07	The Rudest Things You Can Do At A Hotel	3.316625
9	2018-02-14	Can There Be Equity In The Bike Lane?	3.316625
10	2018-01-12	No Shitholes In The Eyes Of Jesus	3.316625

```
tfidf_headline_vectorizer = TfidfVectorizer(min_df = 1)
tfidf_headline_features = tfidf_headline_vectorizer.fit_transform(news_art_temp['headline'])
```

```
def tfidf_based_model(row_index, num_similar_items):
    couple_dist = pairwise_distances(tfidf_headline_features,tfidf_headline_features[row_index])
    indices = np.argsort(couple_dist.ravel())[0:num_similar_items]
    df = pd.DataFrame({'publish_date': news_art['date'][indices].values,
                'headline':news_art['headline'][indices].values,
                 'Euclidean similarity with the queried article': couple_dist[indices].ravel()})
    print("="*30,"Queried article details","="*30)
    print('headline : ',news_art['headline'][indices[0]])
    print("\n","="*25,"Recommended articles : ","="*23)

    #return df.iloc[1:,1]
    return df.iloc[1:,]
tfidf_based_model(133, 11)
```

```
============================== Queried article details ==============================
headline :  Stocks Dive For Truth Social SPAC Amid Merger Delay

 ========================= Recommended articles :  =======================
```

	publish_date	headline	Euclidean similarity with the queried article
1	2022-04-28	Seth Meyers Spots ‘Desperate’ Moment That Indicates The State Of Truth Social	1.255788
2	2022-08-26	Donald Trump's Truth Social Reportedly Faces Major Money, Trademark Woes	1.256534
3	2019-09-05	Only 1 Person Remains Missing From California Dive Boat Fire	1.263386
4	2018-07-12	Justice Department Will Appeal Approval Of AT&T-Time Warner Merger	1.283725
5	2018-11-30	North Carolina Delays Certifying Results Of Congressional Race Amid Probe Of Irregularities	1.294606
6	2018-05-11	Michael Cohen Reportedly Paid $600,000 To Advise AT&T On Time Warner Merger	1.296806
7	2018-02-07	Everything You Should Know About The Stock Market	1.297108
8	2019-09-06	California Dive Boat Crew Reports Multiple Attempts To Save 34 Passengers	1.297519
9	2018-04-18	Here's The Truth About The Caravan Of Migrants Trump Keeps Going On About	1.300489

```
!pip install gensim
```

```
Requirement already satisfied: gensim in /usr/local/lib/python3.10/dist-packages (4.3.3)
Requirement already satisfied: numpy<2.0,>=1.18.5 in /usr/local/lib/python3.10/dist-packages (from gensim) (1.26.4)
Requirement already satisfied: scipy<1.14.0,>=1.7.0 in /usr/local/lib/python3.10/dist-packages (from gensim) (1.13.1)
Requirement already satisfied: smart-open>=1.8.1 in /usr/local/lib/python3.10/dist-packages (from gensim) (7.0.5)
Requirement already satisfied: wrapt in /usr/local/lib/python3.10/dist-packages (from smart-open>=1.8.1->gensim) (1.16.0)
```

```
from gensim.models import Word2Vec
from gensim.models import KeyedVectors
import pickle
import gensim.downloader as api
```

```
model_name = "word2vec-google-news-300"
model_path = api.load(model_name, return_path=True)
```

```
[==================================================] 100.0% 1662.8/1662.8MB downloaded
```

```
loaded_model = KeyedVectors.load_word2vec_format(model_path, binary=True)
```

```
def preprocess_text(text):
    text = re.sub(r'http\S+|www.\S+', '', text)
    text = text.lower()
    text = re.sub(r'[^a-z\s]', '', text)
    tokens = [word for word in text.split() if word not in stop_words]
    tokens = [lemmatizer.lemmatize(word) for word in tokens]
    return tokens
```

```
# Convert each headline to a vector using word embeddings
def get_embedding_vector(tokens):
    vectors = [loaded_model[word] for word in tokens if word in loaded_model]
    if vectors:
        return np.mean(vectors, axis=0)  # Average to get the headline embedding
    else:
        return np.zeros(loaded_model.vector_size)
```

```
# Apply preprocessing and embedding vectorization
news_art['tokens'] = news_art['headline'].apply(preprocess_text)
news_art['embedding'] = news_art['tokens'].apply(get_embedding_vector)

# Stack embedding vectors into a matrix
embedding_matrix = np.vstack(news_art['embedding'].values)


# K-Means clustering on embeddings
kmeans = KMeans(n_clusters=5, random_state=0)
clusters = kmeans.fit_predict(embedding_matrix)
news_art['cluster'] = clusters


# Example user read history (indices of articles read)
user_read_history = [10, 25, 75]

# Create a user profile by averaging vectors of read articles
def create_user_profile(embedding_matrix, user_read_history):
    user_profile = np.mean(embedding_matrix[user_read_history], axis=0)
    return user_profile

user_profile = create_user_profile(embedding_matrix, user_read_history)
```

Recommended Article based on cosine similarity

```
# Calculate similarity between user profile and all article vectors
similarity_scores = cosine_similarity([user_profile], embedding_matrix)[0]

# Get indices of the top N most similar articles
def recommend_articles(similarity_scores, N=5):
    recommended_indices = similarity_scores.argsort()[-N:][::-1]
    return recommended_indices
```