Applying word2vec

Import data and packages:

In [ ]:
```
#import data frames with cleaned data from previous notebook

%store -r textdata

%store -r titledata
```

In [ ]:
```
#import packages

import pandas as pd

import numpy as np

from sklearn.model_selection import train_test_split

import gensim

from gensim.models import Word2Vec
```

In [ ]:
```
#import pre-trained word2vec model trained on Google News data

import gensim.downloader as api

wv = api.load('word2vec-google-news-300')
```

Apply word2vec to text column:

In [ ]:
```
#split into train and test sets

text_train, text_test = train_test_split(
    textdata,
    test_size=0.3,
    random_state=42
)
```

In [ ]:
```
#add column to dataframes containing average vector for each row's text

text_train['vectors'] = text_train['text'].apply(lambda x: wv.get_mean_vector(x, ignore_missing=True))
```

In [ ]:
```
text_test['vectors'] = text_test['text'].apply(lambda x: wv.get_mean_vector(x, ignore_missing=True))
```

In [ ]:
```
#view results

text_train
```

Out[ ]:

	text	label	vectors
7768	[swirl, revelation, allegation, russian, invol...	0	[-0.0003381749, 0.004170802, 8.591608e-05, 0.0...
20151	[despite, problem, plague, donald, trump, fail...	1	[0.0077793393, 0.019553736, 0.0042430665, 0.03...
1003	[tune, alternate, current, radio, network, acr...	1	[0.0028786482, 0.008605114, 0.0070612496, 0.03...
60898	[kiev, reuters, ukrainian, president, petro, p...	0	[-0.02019227, -0.0072652902, 0.0150532145, 0.0...
18689	[madman, merkel, demand, internet, publicly, r...	1	[0.0044224965, 0.002993722, 0.005907909, 0.022...
...	...	...	...
66235	[aboard, aquarius, rescue, ship, reuters, migr...	0	[0.011924975, 0.010988319, 0.0011468495, 0.025...
44512	[moscow, reuters, senior, russian, lawmaker, s...	0	[-0.006364392, 0.0013059269, 0.018351823, 0.01...
921	[trump, want, win, rust, belt, need, practice,...	1	[0.0034884035, 0.010201223, 0.00054162065, 0.0...
17458	[dubai, reuters, former, egyptian, prime, mini...	0	[-0.0059219073, 0.0037472998, 0.011917742, 0.0...
69180	[tim, kaine, cheer, end, white, majority, span...	1	[-0.00827276, 0.0054527237, 0.01909309, 0.0335...

40987 rows × 3 columns

In [ ]:
```
text_test
```

Out[ ]:

	text	label	vectors
33113	[new, york, reuters, michael, moore, left-wing...	0	[-0.00028460205, 0.0022248349, -0.0014604458, ...
56520	[catholic, church, decade, long, likely, even,...	1	[0.021414263, 0.006217401, 0.010361887, 0.0372...
24790	[megyn, kelly, fox, news, anchor, definitely, ...	1	[0.008622794, 0.010029217, -0.0054355743, 0.02...
5431	[peter, certo, wordseven, election, year, shoo...	1	[0.007797616, 0.005972583, 0.013798478, 0.0367...
16964	[something, interest, unz, review, recipient, ...	1	[0.009118676, 0.008844791, 0.008179908, 0.0458...
...	...	...	...
38902	[australia, stop, donation, corrupt, clinton, ...	1	[-0.006731601, -0.0014126656, 0.0031150207, 0....
48226	[ten, progressive, judge, virginia, decide, mu...	0	[0.005291016, 0.004906544, 0.01436485, 0.02873...
19398	[fort, lauderdale, fla, reuters, florida, gove...	0	[-0.005333254, 0.006447841, 0.0089601865, 0.02...
4343	[donald, trump, stun, neglect, disavow, nazi, ...	1	[0.001132972, 0.009400117, 0.009536006, 0.0369...
36825	[president, trump, nominate, christopher, wray...	1	[-0.0077695693, 0.011566349, 0.008971804, 0.02...

17567 rows × 3 columns

In [ ]:
```
#split train and test data into x and y

x_text_train_wv = text_train.vectors
x_text_test_wv = text_test.vectors

y_text_train_wv = text_train.label
y_text_test_wv = text_test.label
```

In [ ]:
```
# make vector arrays usable for modelling

x_text_train_wv_2d = np.stack(x_text_train_wv)

x_text_test_wv_2d = np.stack(x_text_test_wv)
```

In [ ]:
```
#store variables for later use

%store x_text_train_wv_2d
%store x_text_test_wv_2d
%store y_text_train_wv
%store y_text_test_wv
```

```
Stored 'x_text_train_wv_2d' (ndarray)
Stored 'x_text_test_wv_2d' (ndarray)
Stored 'y_text_train_wv' (Series)
Stored 'y_text_test_wv' (Series)
```

Apply word2vec to title column:

In [ ]:
```
#split title data into training and test sets

title_train, title_test = train_test_split(
    titledata,
    test_size=0.3,
    random_state=42
)
```

In [ ]:
```
title_test['vectors'] = title_test['title'].apply(lambda x: wv.get_mean_vector(x, ignore_missing=True))
```

In [ ]:
```
title_train['vectors'] = title_train['title'].apply(lambda x: wv.get_mean_vector(x, ignore_missing=True))
```

In [ ]:
```
#split train and test data into x and y

x_title_train_wv = title_train.vectors
x_title_test_wv = title_test.vectors

y_title_train_wv = title_train.label
y_title_test_wv = title_test.label
```

In [ ]:
```
# make vector arrays usable for modelling
```

```
x_title_train_wv_2d = np.stack(x_title_train_wv)

x_title_test_wv_2d = np.stack(x_title_test_wv)
```

In [ ]:

```
#store variables for later use

%store x_title_train_wv_2d
%store x_title_test_wv_2d
%store y_title_train_wv
%store y_title_test_wv
```

```
Stored 'x_title_train_wv_2d' (ndarray)
Stored 'x_title_test_wv_2d' (ndarray)
Stored 'y_title_train_wv' (Series)
Stored 'y_title_test_wv' (Series)
```