In [2]:
```
# This Python 3 environment comes with many helpful analytics libraries inst
# It is defined by the kaggle/python Docker image: https://github.com/kaggle
# For example, here's several helpful packages to load

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

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will l

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that
# You can also write temporary files to /kaggle/temp/, but they won't be sav
```

```
/kaggle/input/learn-ai-bbc/BBC News Train.csv
/kaggle/input/learn-ai-bbc/BBC News Sample Solution.csv
/kaggle/input/learn-ai-bbc/BBC News Test.csv
```

Load in the data

In [84]:
```
news_train = pd.read_csv('/kaggle/input/learn-ai-bbc/BBC News Train.csv')
news_test = pd.read_csv('/kaggle/input/learn-ai-bbc/BBC News Test.csv')
print(news_train)
print(news_test)
```

```
      ArticleId                                               Text  \
0          1833  worldcom ex-boss launches defence lawyers defe...
1           154  german business confidence slides german busin...
2          1101  bbc poll indicates economic gloom citizens in ...
3          1976  lifestyle  governs mobile choice  faster  bett...
4           917  enron bosses in $168m payout eighteen former e...
...         ...                                                ...
1485        857  double eviction from big brother model caprice...
1486        325  dj double act revamp chart show dj duo jk and ...
1487       1590  weak dollar hits reuters revenues at media gro...
1488       1587  apple ipod family expands market apple has exp...
1489        538  santy worm makes unwelcome visit thousands of ...

           Category
0          business
1          business
2          business
3              tech
4          business
...             ...
1485  entertainment
1486  entertainment
1487       business
1488           tech
1489           tech

[1490 rows x 3 columns]
     ArticleId                                               Text
0         1018  qpr keeper day heads for preston queens park r...
1         1319  software watching while you work software that...
2         1138  d arcy injury adds to ireland woe gordon d arc...
3          459  india s reliance family feud heats up the ongo...
4         1020  boro suffer morrison injury blow middlesbrough...
..         ...                                                ...
730       1923  eu to probe alitalia  state aid  the european ...
731        373  u2 to play at grammy awards show irish rock ba...
732       1704  sport betting rules in spotlight a group of mp...
733        206  alfa romeos  to get gm engines  fiat is to sto...
734        471  citizenship event for 18s touted citizenship c...

[735 rows x 2 columns]
```

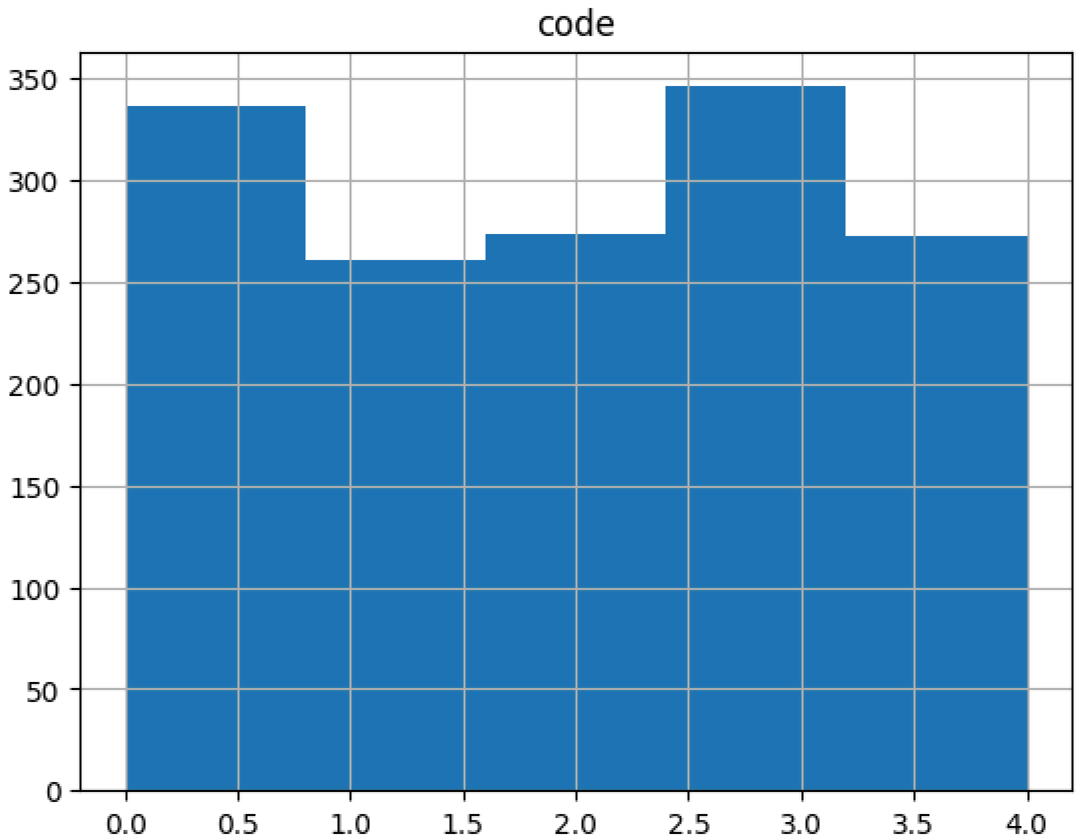
Exploratory Analysis

Make a numerical category column and run a histogram

In [4]:
```python
i = 0
code_dict = {}
for category in news_train.Category.unique():
    code_dict[category] = i
    i+=1

code_list = []
```

```
for idx in news_train.index:
    code_list.append(code_dict[news_train['Category'][idx]])

news_train['code'] = code_list

print(news_train.hist(column='code',bins=5))
print(code_dict)
```

```
[[<Axes: title={'center': 'code'}>]]
{'business': 0, 'tech': 1, 'politics': 2, 'sport': 3, 'entertainment': 4}
```

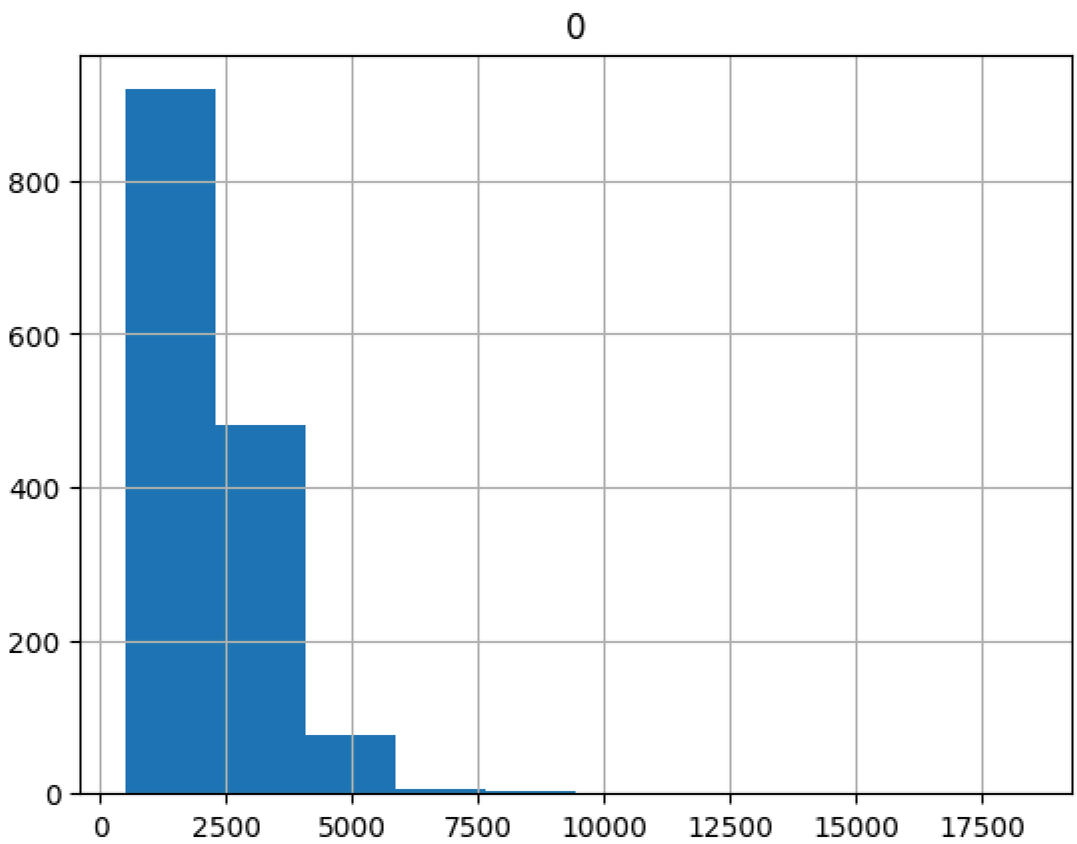


It seems that the training data is evenly spread out among the categories. This is good, as no category will have extreme bias in our model.

Text variability

In [5]:
```
len_list = []
for txt in news_train.Text:
    len_list.append(len(txt))

len_df = pd.DataFrame(len_list)

print(len_df.hist())
```

```
[[<Axes: title={'center': '0'}>]]
```



This shows that the vast majority of text documents have less than 2500 characters. Let's look at the most common words in each, removing for common stopwords. Since most of our articles are 2500 characters, lets rank the most common 250 words per category.

Text Uniqueness

In [6]:
```
from collections import Counter
from nltk.corpus import stopwords

counter_dict = {}

for category in news_train.Category.unique():
    tokens = []
    print(category)
    for txt in news_train[news_train['Category'] == category]['Text']:
        new_tokens = txt.split(" ")
        new_tokens = [x for x in new_tokens if x not in stopwords.words('eng
        tokens = tokens + new_tokens

    c = Counter(tokens).most_common(250)
    print(c)
    c = [x[0] for x in c]
    counter_dict[category] = c
```

```
business
[('', 7978), ('said', 876), ('-', 519), ('us', 497), ('mr', 393), ('would',
308), ('year', 302), ('also', 278), ('new', 273), ('firm', 242), ('company',
240), ('last', 235), ('market', 235), ('growth', 231), ('said.', 211), ('gov
ernment', 205), ('economic', 202), ('could', 198), ('bank', 196), ('econom
y', 189), ('sales', 184), ('may', 174), ('oil', 172), ('however', 163), ('on
e', 163), ('000', 161), ('shares', 161), ('world', 160), ('chief', 153), ('t
wo', 151), ('2004', 148), ('.', 142), ('financial', 139), ('uk', 134), ('bus
iness', 133), ('analysts', 133), ('deal', 132), ('companies', 130), ('chin
a', 128), ('prices', 123), ('expected', 119), ('people', 117), ('rise', 11
5), ('three', 113), ('group', 113), ('years', 112), ('country', 112), ('man
y', 112), ('dollar', 112), ('since', 110), ('yukos', 109), ('year.', 108),
('india', 107), ('still', 105), ('firms', 104), ('trade', 102), ('tax', 10
1), ('stock', 101), ('biggest', 100), ('told', 98), ('months', 98), ('intere
st', 98), ('time', 96), ('profits', 95), ('president', 94), ('figures', 94),
('rates', 94), ('executive', 94), ('made', 94), ('rate', 93), ('european', 9
2), ('countries', 92), ('investment', 92), ('spending', 91), ('first', 90),
('foreign', 90), ('offer', 89), ('strong', 88), ('set', 87), ('recent', 86),
('2005', 84), ('demand', 83), ('money', 83), ('high', 82), ('december', 82),
('news', 81), ('according', 81), ('quarter', 80), ('price', 80), ('cut', 8
0), ('rose', 79), ('state', 78), ('much', 78), ('next', 77), ('likely', 77),
('despite', 76), ('budget', 76), ('jobs', 76), ('united', 76), ('increase',
75), ('back', 75), ('deficit', 75), ('south', 74), ('pay', 74), ('hit', 73),
('part', 73), ('former', 72), ('make', 72), ('europe', 71), ('well', 71),
('investors', 71), ('share', 70), ('take', 70), ('london', 70), ('industry',
70), ('global', 68), ('russian', 68), ('exchange', 68), ('fall', 67), ('cost
s', 67), ('million', 67), ('month', 67), ('bid', 67), ('fell', 66), ('intern
ational', 66), ('club', 66), ('move', 66), ('eu', 66), ('record', 65), ('pu
t', 65), ('debt', 65), ('bankruptcy', 64), ('sale', 64), ('giant', 63), ('we
ek', 63), ('court', 63), ('euro', 62), ('car', 62), ('plans', 62), ('repor
t', 62), ('end', 62), ('january', 61), ('minister', 61), ('annual', 61), ('c
onsumer', 61), ('say', 61), ('number', 61), ('public', 60), ('current', 60),
('need', 58), ('largest', 57), ('exports', 57), ('shareholders', 57), ('russ
ia', 57), ('japan', 57), ('even', 56), ('seen', 56), ('euros', 56), ('profi
t', 56), ('boost', 56), ('continue', 55), ('higher', 55), ('main', 54), ('le
ss', 54), ('2003', 54), ('stake', 54), ('fraud', 53), ('german', 53), ('deut
sche', 53), ('work', 53), ('finance', 52), ('agreed', 52), ('buy', 52), ('pr
oduction', 52), ('previous', 52), ('airline', 52), ('indian', 52), ('earning
s', 51), ('trading', 51), ('lost', 51), ('unit', 51), ('already', 51), ('sec
ond', 51), ('talks', 51), ('commission', 51), ('retail', 50), ('including',
50), ('national', 50), ('major', 50), ('good', 50), ('2004.', 50), ('worldco
m', 49), ('says', 49), ('november', 49), ('low', 49), ('years.', 49), ('ge
t', 49), ('gm', 49), ('inflation', 48), ('markets', 48), ('added', 48), ('se
e', 48), ('although', 48), ('general', 48), ('mci', 47), ('value', 47), ('de
velopment', 47), ('future', 47), ('help', 47), ('came', 47), ('takeover', 4
7), ('early', 46), ('latest', 46), ('2005.', 46), ('case', 46), ('decision',
46), ('meeting', 46), ('house', 46), ('warned', 46), ('agreement', 46),
('&', 46), ('close', 45), ('cost', 45), ('total', 45), ('earlier', 45), ('da
ta', 45), ('glazer', 45), ('germany', 44), ('half', 44), ('announced', 44),
('reported', 44), ('another', 44), ('increased', 44), ('imf', 44), ('ebber
s', 43), ('economy.', 43), ('2003.', 43), ('analyst', 43), ('sold', 43), ('f
our', 43), ('federal', 43), ('market.', 43), ('air', 43), ('economist', 42),
('domestic', 42), ('10', 42), ('past', 42), ('standard', 42), ('lse', 42),
('level', 41)]
tech
[('', 9465), ('said', 815), ('people', 624), ('new', 349), ('mr', 349), ('al
```

```
so', 347), ('-', 338), ('would', 321), ('one', 316), ('mobile', 311), ('coul
d', 308), ('technology', 263), ('users', 249), ('digital', 238), ('use', 23
7), ('many', 234), ('music', 234), ('net', 228), ('software', 223), ('sai
d.', 216), ('phone', 209), ('us', 209), ('make', 208), ('like', 205), ('game
s', 198), ('microsoft', 188), ('used', 180), ('service', 180), ('first', 17
9), ('get', 176), ('uk', 172), ('million', 171), ('way', 163), ('internet',
162), ('computer', 160), ('video', 160), ('year', 159), ('online', 157), ('b
roadband', 155), ('world', 152), ('tv', 150), ('game', 147), ('number', 14
4), ('services', 144), ('phones', 139), ('time', 139), ('information', 138),
('search', 138), ('using', 135), ('according', 134), ('firms', 133), ('medi
a', 133), ('security', 133), ('content', 128), ('data', 128), ('system', 12
8), ('much', 126), ('two', 126), ('firm', 124), ('pc', 115), ('market', 11
5), ('web', 115), ('even', 114), ('take', 113), ('around', 112), ('bbc', 11
1), ('apple', 110), ('e-mail', 109), ('000', 108), ('already', 108), ('wor
k', 108), ('made', 108), ('next', 107), ('last', 106), ('news', 105), ('say
s', 104), ('research', 103), ('well', 101), ('help', 100), ('go', 99), ('com
panies', 99), ('see', 98), ('going', 96), ('show', 96), ('want', 96), ('son
y', 96), ('players', 96), ('consumers', 95), ('different', 95), ('site', 9
5), ('years', 94), ('part', 94), ('able', 93), ('home', 92), ('set', 90),
('devices', 88), ('still', 88), ('mobiles', 87), ('networks', 87), ('compan
y', 86), ('told', 85), ('three', 84), ('top', 84), ('consumer', 83), ('end',
82), ('.', 82), ('virus', 82), ('good', 81), ('europe', 80), ('gadget', 80),
('radio', 80), ('industry', 79), ('google', 79), ('gadgets', 79), ('report',
78), ('access', 77), ('every', 76), ('find', 76), ('may', 76), ('put', 76),
('european', 75), ('network', 75), ('need', 75), ('windows', 75), ('bt', 7
4), ('found', 74), ('spam', 74), ('portable', 74), ('technologies', 73), ('c
ontrol', 72), ('gaming', 71), ('via', 71), ('likely', 71), ('messages', 70),
('version', 70), ('free', 70), ('small', 70), ('sites', 70), ('play', 70),
('although', 69), ('offer', 69), ('2004', 69), ('back', 69), ('become', 69),
('year.', 69), ('hard', 68), ('say', 67), ('five', 66), ('months', 66), ('re
leased', 66), ('customers', 65), ('camera', 65), ('several', 65), ('called',
65), ('really', 64), ('currently', 64), ('personal', 64), ('computers', 63),
('looking', 63), ('latest', 63), ('means', 63), ('mac', 63), ('player', 63),
('machines', 62), ('come', 61), ('websites', 61), ('better', 60), ('popula
r', 60), ('almost', 60), ('device', 60), ('pcs', 60), ('available', 59), ('d
ownload', 59), ('programs', 59), ('machine', 59), ('2', 59), ('big', 58),
('images', 58), ('nintendo', 58), ('website', 58), ('systems', 57), ('makin
g', 57), ('group', 57), ('without', 57), ('attacks', 57), ('drive', 56), ('a
nother', 56), ('growing', 56), ('look', 56), ('getting', 56), ('wireless', 5
6), ('behind', 55), ('legal', 55), ('future', 55), ('electronics', 55), ('si
nce', 55), ('power', 55), ('10', 55), ('it.', 54), ('lot', 54), ('importan
t', 54), ('entertainment', 54), ('something', 54), ('mean', 54), ('mini', 5
4), ('launched', 53), ('due', 53), ('working', 53), ('less', 53), ('expecte
d', 53), ('blogs', 53), ('far', 52), ('director', 52), ('viruses', 52), ('ca
sh', 52), ('think', 52), ('let', 52), ('similar', 51), ('generation', 51),
('2005', 51), ('per', 50), ('xbox', 50), ('sold', 50), ('calls', 50), ('dr',
49), ('text', 49), ('files', 49), ('launch', 49), ('dvd', 49), ('cost', 49),
('might', 48), ('gamers', 48), ('high-definition', 48), ('cameras', 47), ('f
ilm', 47), ('allow', 47), ('money', 47), ('current', 47), ('analysts', 47),
('early', 46), ('seen', 46), ('share', 46), ('including', 46), ('rather', 4
6), ('across', 45), ('watch', 45), ('chief', 45), ('spyware', 45), ('storag
e', 45), ('products', 45)]
politics
[('', 9438), ('mr', 1071), ('said', 971), ('would', 710), ('-', 501), ('labo
ur', 469), ('government', 430), ('blair', 372), ('.', 370), ('people', 361),
('party', 334), ('election', 317), ('also', 308), ('new', 280), ('could', 27
```

```
2), ('said.', 271), ('minister', 265), ('brown', 251), ('uk', 223), ('told',
216), ('said:', 203), ('public', 201), ('prime', 194), ('howard', 191), ('pl
ans', 185), ('say', 169), ('secretary', 169), ('one', 167), ('tory', 166),
('tax', 164), ('general', 162), ('britain', 162), ('leader', 151), ('home',
150), ('next', 150), ('lord', 146), ('tories', 146), ('says', 145), ('chance
llor', 144), ('bbc', 143), ('two', 136), ('tony', 133), ('lib', 133), ('ge
t', 130), ('last', 128), ('make', 128), ('british', 128), ('spokesman', 12
6), ('000', 125), ('bill', 123), ('time', 118), ('police', 117), ('campaig
n', 117), ('made', 115), ('first', 115), ('liberal', 114), ('michael', 112),
('eu', 112), ('year', 111), ('council', 108), ('local', 105), ('law', 104),
('take', 104), ('want', 103), ('many', 99), ('mps', 98), ('part', 97), ('uki
p', 97), ('kennedy', 96), ('may', 94), ('political', 93), ('work', 90), ('ho
use', 89), ('way', 88), ('going', 88), ('years', 88), ('vote', 87), ('countr
y', 87), ('us', 87), ('set', 84), ('foreign', 84), ('expected', 84), ('bac
k', 84), ('good', 84), ('issue', 83), ('children', 82), ('parties', 82), ('f
ormer', 81), ('ministers', 81), ('believe', 81), ('help', 79), ('think', 7
8), ('like', 78), ('saying', 78), ('already', 78), ('election.', 78), ('euro
pean', 77), ('dems', 77), ('claims', 76), ('week', 76), ('immigration', 76),
('asylum', 76), ('put', 76), ('world', 75), ('office', 75), ('gordon', 74),
('increase', 74), ('london', 73), ('support', 72), ('men', 72), ('voters', 7
2), ('without', 71), ('conservative', 70), ('need', 70), ('report', 70), ('c
hange', 70), ('health', 70), ('commons', 69), ('pay', 69), ('rights', 68),
('number', 67), ('national', 67), ('plan', 67), ('charles', 66), ('conservat
ives', 66), ('democrats', 66), ('right', 65), ('iraq', 64), ('even', 64),
('kilroy-silk', 64), ('system', 64), ('see', 63), ('come', 63), ('human', 6
3), ('go', 62), ('whether', 62), ('mp', 62), ('lords', 62), ('legal', 62),
('still', 62), ('deal', 61), ('john', 61), ('war', 60), ('use', 60), ('hel
d', 60), ('shadow', 60), ('services', 60), ('called', 59), ('cabinet', 59),
('straw', 59), ('service', 59), ('give', 59), ('much', 59), ('policy', 59),
('used', 58), ('countries', 58), ('money', 58), ('four', 58), ('court', 58),
('taxes', 58), ('clear', 57), ('well', 57), ('schools', 57), ('evidence', 5
7), ('parliament', 56), ('news', 56), ('stand', 56), ('dem', 56), ('three',
55), ('since', 55), ('meeting', 55), ('david', 55), ('ms', 55), ('action', 5
4), ('given', 54), ('decision', 54), ('place', 54), ('members', 54), ('spend
ing', 53), ('minimum', 53), ('education', 53), ('commission', 52), ('budge
t', 52), ('move', 52), ('working', 52), ('role', 52), ('poll', 52), ('case',
52), ('chairman', 51), ('must', 51), ('powers', 51), ('issues', 51), ('powe
r', 51), ('debate', 51), ('end', 51), ('trust', 51), ('committee', 50), ('ch
ief', 50), ('clarke', 50), ('act', 50), ('day', 50), ('member', 49), ('affai
rs', 49), ('choice', 49), ('proposals', 49), ('needed', 49), ('cut', 49),
('wales', 48), ('england', 48), ('speech', 48), ('later', 48), ('politics',
48), ('able', 48), ('care', 48), ('big', 48), ('economy', 48), ('away', 48),
('claim', 47), ('allow', 47), ('key', 47), ('wage', 47), ('full', 46), ('les
s', 46), ('union', 46), ('far', 46), ('terror', 46), ('third', 46), ('went',
45), ('answer', 45), ('added.', 45), ('civil', 45), ('within', 45), ('radi
o', 45), ('better', 45), ('blunkett', 45), ('advice', 44), ('denied', 44),
('know', 44), ('id', 44), ('figures', 44), ('accused', 44), ('great', 44),
('group', 44), ('women', 44), ('conference', 43), ('got', 43), ('income', 4
3), ('show', 43), ('statement', 42), ('downing', 42)]
sport
[('', 9107), ('said', 356), ('first', 321), ('england', 313), ('-', 303),
('game', 285), ('win', 261), ('last', 255), ('two', 251), ('world', 248),
('would', 233), ('one', 230), ('back', 215), ('also', 214), ('new', 195),
('cup', 192), ('time', 191), ('players', 190), ('ireland', 181), ('play', 17
8), ('side', 172), ('could', 171), ('wales', 169), ('six', 167), ('second',
165), ('good', 161), ('three', 160), ('said.', 155), ('team', 152), ('year',
```

```
148), ('made', 145), ('get', 144), ('chelsea', 144), ('match', 140), ('fina
l', 136), ('coach', 135), ('france', 134), ('great', 131), ('take', 129),
('set', 125), ('club', 125), ('think', 125), ('said:', 123), ('told', 123),
('united', 121), ('well', 120), ('like', 119), ('since', 118), ('next', 11
8), ('still', 116), ('got', 116), ('open', 112), ('played', 112), ('internat
ional', 112), ('start', 110), ('make', 109), ('rugby', 109), ('going', 108),
('arsenal', 107), ('champion', 106), ('us', 105), ('go', 104), ('olympic', 1
01), ('injury', 100), ('ball', 100), ('games', 100), ('minutes', 99), ('bes
t', 99), ('league', 99), ('nations', 97), ('scotland', 97), ('williams', 9
6), ('playing', 95), ('right', 94), ('home', 93), ('roddick', 93), ('seaso
n', 93), ('years', 92), ('victory', 91), ('four', 91), ('know', 91), ('v', 9
1), ('chance', 90), ('way', 90), ('five', 89), ('another', 89), ('jones', 8
9), ('really', 87), ('want', 86), ('beat', 85), ('end', 85), ('top', 85),
('put', 85), ('former', 84), ('player', 84), ('grand', 83), ('lot', 83), ('l
eft', 83), ('number', 82), ('champions', 82), ('winning', 81), ('try', 80),
('took', 79), ('come', 78), ('manager', 78), ('title', 77), ('week', 77),
('came', 76), ('see', 76), ('even', 76), ('liverpool', 76), ('australian', 7
5), ('face', 74), ('break', 72), ('third', 71), ('boss', 71), ('robinson', 7
1), ('away', 71), ('goal', 70), ('points', 69), ('return', 69), ('half', 6
7), ('better', 67), ('european', 67), ('game.', 66), ('never', 66), ('lost',
66), ('football', 65), ('lead', 65), ('place', 65), ('italy', 64), ('give',
64), ('big', 64), ('bbc', 63), ('season.', 63), ('defeat', 62), ('referee',
62), ('j', 62), ('mark', 61), ('seed', 61), ('andy', 60), ('decision', 60),
('penalty', 59), ('much', 59), ('premiership', 59), ('early', 58), ('went',
58), ('ferguson', 58), ('missed', 58), ('nadal', 58), ('record', 57), ('manc
hester', 57), ('captain', 56), ('tennis', 56), ('squad', 56), ('it.', 56),
('long', 56), ('people', 56), ('despite', 55), ('french', 55), ('round', 5
5), ('test', 55), ('holmes', 55), ('.', 54), ('form', 54), ('zealand', 54),
('race', 54), ('g', 54), ('athens', 54), ('irish', 53), ('10', 53), ('sunda
y', 53), ('real', 53), ('added:', 53), ('slam', 52), ('hard', 52), ('wenge
r', 52), ('forward', 52), ('britain', 52), ('days', 52), ('mourinho', 52),
('madrid', 52), ('training', 51), ('ahead', 50), ('run', 50), ('given', 50),
('indoor', 50), ('work', 50), ('scored', 50), ('saturday', 49), ('looking',
49), ('career', 49), ('pressure', 49), ('drugs', 49), ('says', 48), ('gara',
48), ('spain', 48), ('every', 48), ('event', 48), ('centre', 47), ('openin
g', 47), ('tour', 47), ('need', 47), ('many', 47), ('later', 46), ('may', 4
6), ('american', 46), ('matches', 46), ('line', 46), ('south', 46), ('takin
g', 45), ('fourth', 45), ('hodgson', 45), ('lions', 45), ('athletics', 45),
('shot', 45), ('believes', 44), ('weeks', 44), ('fans', 44), ('newcastle', 4
4), ('davis', 44), ('difficult', 43), ('johnson', 43), ('always', 43), ('hap
py', 43), ('fa', 43), ('striker', 43), ('british', 43), ('gold', 43), ('kent
eris', 43), ('behind', 42), ('hope', 42), ('women', 42), ('david', 42), ('ad
mitted', 42), ('city', 42), ('bit', 41), ('contract', 41), ('men', 41), ('au
stralia', 41), ('henman', 41), ('important', 41), ('national', 41), ('whethe
r', 41), ('say', 41), ('failed', 41), ('dallaglio', 41), ('point', 40), ('go
als', 40), ('front', 40)]
entertainment
[('', 7267), ('film', 506), ('-', 464), ('best', 404), ('said', 383), ('als
o', 277), ('one', 249), ('us', 240), ('new', 232), ('music', 232), ('year',
213), ('show', 187), ('first', 184), ('number', 165), ('last', 159), ('acto
r', 158), ('uk', 157), ('band', 157), ('awards', 151), ('director', 148),
('mr', 148), ('.', 142), ('star', 140), ('top', 138), ('would', 138), ('tw
o', 137), ('tv', 135), ('said.', 134), ('british', 129), ('award', 123), ('f
ilms', 120), ('bbc', 120), ('people', 119), ('including', 115), ('three', 11
3), ('album', 109), ('actress', 109), ('years', 106), ('singer', 102), ('mad
e', 100), ('time', 97), ('stars', 93), ('million', 91), ('like', 87), ('come
```

```
dy', 86), ('festival', 84), ('oscar', 84), ('chart', 83), ('could', 82), ('m
ovie', 81), ('record', 79), ('hit', 78), ('five', 78), ('musical', 78), ('wo
rld', 77), ('make', 77), ('said:', 77), ('song', 77), ('well', 76), ('play',
76), ('london', 75), ('box', 74), ('sales', 74), ('big', 73), ('get', 73),
('took', 72), ('rock', 72), ('role', 71), ('hollywood', 70), ('000', 70),
('2004', 70), ('go', 70), ('series', 69), ('single', 66), ('many', 66), ('se
t', 65), ('book', 64), ('place', 63), ('man', 63), ('drama', 63), ('second',
62), ('theatre', 62), ('academy', 62), ('told', 61), ('starring', 61), ('avi
ator', 61), ('went', 61), ('office', 60), ('going', 60), ('pop', 59), ('thin
k', 59), ('four', 59), ('named', 59), ('nominated', 59), ('success', 58),
('life', 57), ('day', 57), ('win', 57), ('prize', 57), ('include', 56), ('gr
oup', 56), ('released', 56), ('children', 56), ('original', 56), ('since', 5
4), ('john', 54), ('ceremony', 54), ('nominations', 54), ('radio', 53), ('se
e', 53), ('former', 52), ('take', 52), ('love', 52), ('among', 51), ('indust
ry', 51), ('may', 50), ('company', 50), ('live', 50), ('work', 49), ('playe
d', 49), ('week', 49), ('debut', 49), ('television', 49), ('later', 49), ('n
ext', 49), ('charles', 49), ('third', 48), ('good', 48), ('came', 48), ('ver
sion', 48), ('money', 48), ('still', 47), ('got', 47), ('10', 47), ('america
n', 47), ('due', 47), ('ray', 47), ('christmas', 46), ('taking', 46), ('audi
ence', 45), ('following', 45), ('fans', 45), ('film.', 45), ('oscars', 45),
('great', 44), ('year.', 44), ('shows', 44), ('around', 44), ('home', 44),
('night', 44), ('golden', 44), ('paul', 43), ('singles', 43), ('sold', 43),
('really', 43), ('already', 43), ('release', 42), ('performance', 42), ('sup
porting', 42), ('want', 41), ('stage', 41), ('end', 41), ('never', 41), ('wo
man', 41), ('young', 41), ('part', 41), ('screen', 41), ('however', 40), ('i
ncluded', 40), ('dance', 40), ('died', 40), ('story', 40), ('know', 40), ('b
ecome', 40), ('days', 40), ('much', 39), ('los', 39), ('back', 39), ('produc
ers', 39), ('martin', 39), ('michael', 39), ('winners', 39), ('dollar', 39),
('jamie', 39), ('lee', 39), ('death', 38), ('biggest', 38), ('angeles', 38),
('held', 38), ('york', 38), ('vera', 38), ('1', 37), ('way', 37), ('career',
37), ('says', 37), ('winner', 37), ('elvis', 37), ('received', 36), ('accord
ing', 36), ('school', 36), ('producer', 36), ('special', 36), ('tour', 36),
('act', 36), ('drake', 36), ('despite', 35), ('seen', 35), ('digital', 35),
('songs', 35), ('saw', 35), ('found', 35), ('making', 35), ('20', 35), ('art
ists', 35), ('list', 35), ('added', 35), ('actors', 35), ('news', 35), ('rig
ht', 34), ('black', 34), ('critics', 34), ('host', 34), ('come', 34), ('popu
lar', 33), ('always', 33), ('court', 33), ('given', 33), ('members', 33),
('novel', 33), ('final', 33), ('sideways', 33), ('sir', 32), ('weekend', 3
2), ('channel', 32), ('history', 32), ('across', 32), ('baby', 32), ('famil
y', 32), ('show.', 32), ('awards.', 32), ('dead', 32), ('king', 32), ('fox
x', 32), ('recently', 31), ('age', 31), ('favourite', 31), ('expected', 31),
('hope', 31), ('studio', 30), ('ever', 30)]
```

There seems to be a lot of common words above, such as '', '-', 'said', 'mr', and so on. Let's remove the common ones and see if we can get a clearer deliniation of common words per category.

In [7]:
```python
unique_word_dict = {}
removable_words = []

for category in news_train.Category.unique():
    print(category)
    other_cats = list(news_train.Category.unique())
    other_cats.remove(category)
    used_words = []
```

```
    for cat in other_cats:
        used_words = counter_dict[cat] + used_words

    unique_word_dict[category] = []
    for word in counter_dict[category]:
        if word not in used_words:
            unique_word_dict[category].append(word)
        else:
            removable_words.append(word)

    print(unique_word_dict[category])
```

```
business
['growth', 'economic', 'bank', 'oil', 'shares', 'financial', 'business', 'ch
ina', 'prices', 'rise', 'yukos', 'india', 'trade', 'stock', 'interest', 'pro
fits', 'president', 'rates', 'executive', 'rate', 'investment', 'strong', 'r
ecent', 'demand', 'high', 'december', 'quarter', 'price', 'rose', 'state',
'jobs', 'deficit', 'investors', 'global', 'russian', 'exchange', 'fall', 'co
sts', 'month', 'bid', 'fell', 'debt', 'bankruptcy', 'sale', 'giant', 'euro',
'car', 'january', 'annual', 'largest', 'exports', 'shareholders', 'russia',
'japan', 'euros', 'profit', 'boost', 'continue', 'higher', 'main', '2003',
'stake', 'fraud', 'german', 'deutsche', 'finance', 'agreed', 'buy', 'product
ion', 'previous', 'airline', 'indian', 'earnings', 'trading', 'unit', 'talk
s', 'retail', 'major', '2004.', 'worldcom', 'november', 'low', 'years.', 'g
m', 'inflation', 'markets', 'mci', 'value', 'development', 'takeover', '200
5.', 'warned', 'agreement', '&', 'close', 'total', 'earlier', 'glazer', 'ger
many', 'announced', 'reported', 'increased', 'imf', 'ebbers', 'economy.', '2
003.', 'analyst', 'federal', 'market.', 'air', 'economist', 'domestic', 'pas
t', 'standard', 'lse', 'level']
tech
['mobile', 'technology', 'users', 'net', 'software', 'phone', 'microsoft',
'internet', 'computer', 'video', 'online', 'broadband', 'phones', 'informati
on', 'search', 'using', 'media', 'security', 'content', 'pc', 'web', 'appl
e', 'e-mail', 'research', 'sony', 'consumers', 'different', 'site', 'device
s', 'mobiles', 'networks', 'virus', 'gadget', 'google', 'gadgets', 'access',
'find', 'network', 'windows', 'bt', 'spam', 'portable', 'technologies', 'con
trol', 'gaming', 'via', 'messages', 'free', 'small', 'sites', 'customers',
'camera', 'several', 'currently', 'personal', 'computers', 'means', 'mac',
'machines', 'websites', 'almost', 'device', 'pcs', 'available', 'download',
'programs', 'machine', '2', 'images', 'nintendo', 'website', 'systems', 'att
acks', 'drive', 'growing', 'look', 'getting', 'wireless', 'electronics', 'en
tertainment', 'something', 'mean', 'mini', 'launched', 'blogs', 'viruses',
'cash', 'let', 'similar', 'generation', 'per', 'xbox', 'calls', 'dr', 'tex
t', 'files', 'launch', 'dvd', 'might', 'gamers', 'high-definition', 'camera
s', 'rather', 'watch', 'spyware', 'storage', 'products']
politics
['labour', 'blair', 'party', 'election', 'brown', 'prime', 'howard', 'secret
ary', 'tory', 'leader', 'lord', 'tories', 'chancellor', 'tony', 'lib', 'spok
esman', 'bill', 'police', 'campaign', 'liberal', 'council', 'local', 'law',
'mps', 'ukip', 'kennedy', 'political', 'vote', 'issue', 'parties', 'minister
s', 'believe', 'saying', 'election.', 'dems', 'claims', 'immigration', 'asyl
um', 'gordon', 'support', 'voters', 'conservative', 'change', 'health', 'com
mons', 'rights', 'plan', 'conservatives', 'democrats', 'iraq', 'kilroy-sil
k', 'human', 'mp', 'lords', 'war', 'shadow', 'cabinet', 'straw', 'policy',
'taxes', 'clear', 'schools', 'evidence', 'parliament', 'stand', 'dem', 'ms',
'action', 'minimum', 'education', 'poll', 'chairman', 'must', 'powers', 'iss
ues', 'debate', 'trust', 'committee', 'clarke', 'member', 'affairs', 'choic
e', 'proposals', 'needed', 'speech', 'politics', 'care', 'claim', 'key', 'wa
ge', 'full', 'union', 'terror', 'answer', 'added.', 'civil', 'within', 'blun
kett', 'advice', 'denied', 'id', 'accused', 'conference', 'income', 'stateme
nt', 'downing']
sport
['cup', 'ireland', 'side', 'six', 'team', 'chelsea', 'match', 'coach', 'fran
ce', 'open', 'start', 'rugby', 'arsenal', 'champion', 'olympic', 'injury',
'ball', 'minutes', 'league', 'nations', 'scotland', 'williams', 'playing',
'roddick', 'season', 'victory', 'v', 'chance', 'jones', 'beat', 'grand', 'le
ft', 'champions', 'winning', 'try', 'manager', 'title', 'liverpool', 'austra
lian', 'face', 'break', 'boss', 'robinson', 'goal', 'points', 'return', 'gam
```

```
e.', 'football', 'lead', 'italy', 'season.', 'defeat', 'referee', 'j', 'mar
k', 'seed', 'andy', 'penalty', 'premiership', 'ferguson', 'missed', 'nadal',
'manchester', 'captain', 'tennis', 'squad', 'long', 'french', 'round', 'tes
t', 'holmes', 'form', 'zealand', 'race', 'g', 'athens', 'irish', 'sunday',
'real', 'added:', 'slam', 'wenger', 'forward', 'mourinho', 'madrid', 'traini
ng', 'ahead', 'run', 'indoor', 'scored', 'saturday', 'pressure', 'drugs', 'g
ara', 'spain', 'event', 'centre', 'opening', 'matches', 'line', 'fourth', 'h
odgson', 'lions', 'athletics', 'shot', 'believes', 'weeks', 'newcastle', 'da
vis', 'difficult', 'johnson', 'happy', 'fa', 'striker', 'gold', 'kenteris',
'admitted', 'city', 'bit', 'contract', 'australia', 'henman', 'failed', 'dal
laglio', 'point', 'goals', 'front']
entertainment
['actor', 'band', 'awards', 'star', 'award', 'films', 'album', 'actress', 's
inger', 'stars', 'comedy', 'festival', 'oscar', 'chart', 'movie', 'musical',
'song', 'box', 'rock', 'hollywood', 'series', 'single', 'book', 'man', 'dram
a', 'theatre', 'academy', 'starring', 'aviator', 'pop', 'named', 'nominate
d', 'success', 'life', 'prize', 'include', 'original', 'ceremony', 'nominati
ons', 'love', 'among', 'live', 'debut', 'television', 'ray', 'christmas', 'a
udience', 'following', 'film.', 'oscars', 'shows', 'night', 'golden', 'pau
l', 'singles', 'release', 'performance', 'supporting', 'stage', 'woman', 'yo
ung', 'screen', 'included', 'dance', 'died', 'story', 'los', 'producers', 'm
artin', 'winners', 'jamie', 'lee', 'death', 'angeles', 'york', 'vera', '1',
'winner', 'elvis', 'received', 'school', 'producer', 'special', 'drake', 'so
ngs', 'saw', '20', 'artists', 'list', 'actors', 'black', 'critics', 'host',
'novel', 'sideways', 'sir', 'weekend', 'channel', 'history', 'baby', 'famil
y', 'show.', 'awards.', 'dead', 'king', 'foxx', 'recently', 'age', 'favourit
e', 'studio', 'ever']
```

Analysis Results

Since the categories share such unique words, a TF-IDF vectorizer is a great way to embed the words for machine learning. We will remove the common words to see if that gives a better result.

TF-IDF stands for Term Frequency - Inverse Document Frequency, and it is very good a picking important unique words for categorization efforts. TF-IDF looks at how many times a term appears in a text and then how common the term is amongst other documents.

Pre-Processing

Build the TF-IDF vectors.

One with all words in text, and one with removed words from the exploratory analysis above. The latter will be used in the supervised learning.

In [95]:
```python
from sklearn.feature_extraction.text import TfidfVectorizer
import matplotlib.pyplot as plt
from nltk.tokenize.treebank import TreebankWordDetokenizer
```

```
tfidf_df = news_train.copy()
del tfidf_df['Category']
tfidf_df = pd.concat([tfidf_df,news_test], ignore_index = True)

unsup_tfidf_vect = TfidfVectorizer(stop_words=stopwords.words('english'))
sup_tfidf_vect = TfidfVectorizer(stop_words=stopwords.words('english')+remov

unsup_tfidf = unsup_tfidf_vect.fit_transform(tfidf_df.Text)
unsup_tfidf_df = pd.DataFrame(unsup_tfidf.toarray(), columns=unsup_tfidf_vec

print(unsup_tfidf_df)
```

```
            00       000  0001  000bn  000m  000s  000th  001  001and  001st  \
ArticleId
1833       0.0  0.000000   0.0    0.0   0.0   0.0    0.0  0.0     0.0    0.0
154        0.0  0.000000   0.0    0.0   0.0   0.0    0.0  0.0     0.0    0.0
1101       0.0  0.022992   0.0    0.0   0.0   0.0    0.0  0.0     0.0    0.0
1976       0.0  0.019363   0.0    0.0   0.0   0.0    0.0  0.0     0.0    0.0
917        0.0  0.000000   0.0    0.0   0.0   0.0    0.0  0.0     0.0    0.0
...        ...       ...   ...    ...   ...   ...    ...  ...     ...    ...
1923       0.0  0.000000   0.0    0.0   0.0   0.0    0.0  0.0     0.0    0.0
373        0.0  0.000000   0.0    0.0   0.0   0.0    0.0  0.0     0.0    0.0
1704       0.0  0.000000   0.0    0.0   0.0   0.0    0.0  0.0     0.0    0.0
206        0.0  0.000000   0.0    0.0   0.0   0.0    0.0  0.0     0.0    0.0
471        0.0  0.000000   0.0    0.0   0.0   0.0    0.0  0.0     0.0    0.0

           ...  zooms  zooropa  zornotza  zorro  zubair  zuluaga  zurich  \
ArticleId  ...
1833       ...    0.0      0.0       0.0    0.0     0.0      0.0     0.0
154        ...    0.0      0.0       0.0    0.0     0.0      0.0     0.0
1101       ...    0.0      0.0       0.0    0.0     0.0      0.0     0.0
1976       ...    0.0      0.0       0.0    0.0     0.0      0.0     0.0
917        ...    0.0      0.0       0.0    0.0     0.0      0.0     0.0
...        ...    ...      ...       ...    ...     ...      ...     ...
1923       ...    0.0      0.0       0.0    0.0     0.0      0.0     0.0
373        ...    0.0      0.0       0.0    0.0     0.0      0.0     0.0
1704       ...    0.0      0.0       0.0    0.0     0.0      0.0     0.0
206        ...    0.0      0.0       0.0    0.0     0.0      0.0     0.0
471        ...    0.0      0.0       0.0    0.0     0.0      0.0     0.0

           zutons  zvonareva  zvyagintsev
ArticleId
1833          0.0        0.0          0.0
154           0.0        0.0          0.0
1101          0.0        0.0          0.0
1976          0.0        0.0          0.0
917           0.0        0.0          0.0
...           ...        ...          ...
1923          0.0        0.0          0.0
373           0.0        0.0          0.0
1704          0.0        0.0          0.0
206           0.0        0.0          0.0
471           0.0        0.0          0.0

[2225 rows x 29280 columns]
```

Building the Models

1: When you train the unsupervised model for matrix factorization, should you include texts (word features) from the test dataset or not as the input matrix? Why or why not?

Yes, and we will split them for the different models. We have no way of adding labeled categories to submission file without it.

2. 2) Build a model using the matrix factorization method(s) and predict the train and test data labels. Choose any hyperparameter (e.g., number of word features) to begin with.

Train NVM

In [77]:
```
from sklearn.decomposition import NMF

nmf = NMF(n_components = 5)

categories = nmf.fit_transform(unsup_tfidf_df)
nmf_categories = pd.DataFrame(categories, index=unsup_tfidf_df.index)

print(nmf_categories)
```

```
                 0         1         2         3         4
ArticleId
1833      0.005472  0.035981  0.001387  0.003490  0.040334
154       0.000000  0.000000  0.000000  0.000000  0.175231
1101      0.019100  0.019685  0.015790  0.003193  0.099109
1976      0.143770  0.000000  0.000000  0.000000  0.000000
917       0.007716  0.005010  0.009845  0.007808  0.058389
...            ...       ...       ...       ...       ...
1923      0.008213  0.019590  0.007218  0.000000  0.053152
373       0.000000  0.000000  0.013178  0.141749  0.010406
1704      0.024744  0.016952  0.014848  0.000697  0.020372
206       0.005198  0.010278  0.012124  0.000000  0.034851
471       0.033580  0.066458  0.015770  0.011994  0.014757

[2225 rows x 5 columns]
```

Now that we have the clusters, we can unpack the results and check against the actual labels.

In [78]:
```
train_pred = pd.DataFrame(columns={'ArticleId':[],'Category':[], 'Pred':[]})

for article_id in news_train.ArticleId:
    category = news_train[news_train['ArticleId'] == article_id]['Category']
    pred_list = nmf_categories.loc[article_id].tolist()

    train_pred.loc[len(train_pred)] = [
```

```
        article_id,
        category,
        pred_list.index(max(pred_list))
    ]

print(train_pred)
```

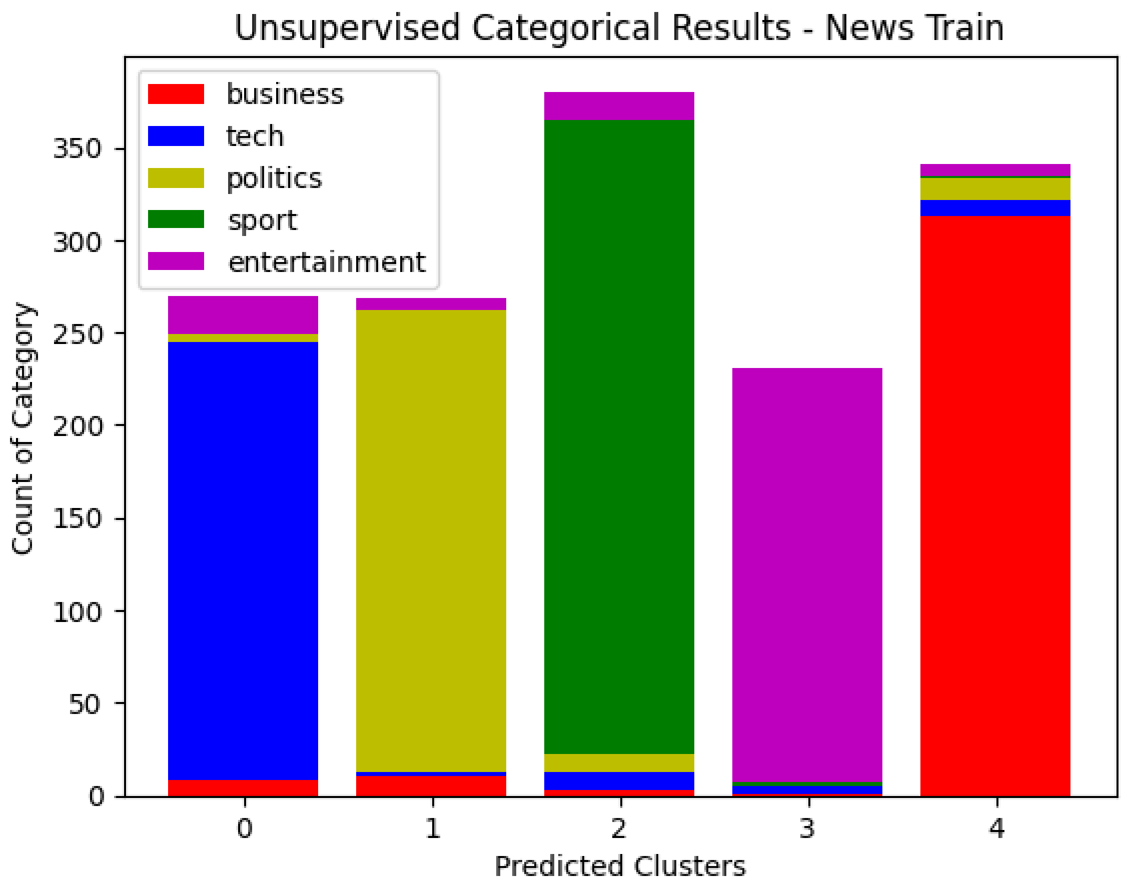
```
      ArticleId       Category  Pred
0          1833       business     4
1           154       business     4
2          1101       business     4
3          1976           tech     0
4           917       business     4
...         ...            ...   ...
1485        857  entertainment     3
1486        325  entertainment     3
1487       1590       business     4
1488       1587           tech     0
1489        538           tech     0

[1490 rows x 3 columns]
```

From the snapshot above, it seems the NVM did a decent job with only one error showing in entertainment. Let's visualize the results in a better way.

Visualize the Results

In [79]:
```
x = [0, 1, 2, 3, 4]

y_cats = {}

for category in news_train.Category.unique():
    cat_df = train_pred[train_pred['Category'] == category].copy()
    y_cats[category] = []
    for pred in x:
        y_cats[category].append(len(cat_df[cat_df['Pred'] == pred].index))

    y_cats[category] = np.array(y_cats[category])

y_true = []
for idx in train_pred.index:
    category = train_pred['Category'][idx]
    y_true.append(
        list(y_cats[category]).index(max(y_cats[category]))
    )

plt.bar(x, y_cats['business'], color = 'r')
plt.bar(x, y_cats['tech'], bottom = y_cats['business'], color='b')
plt.bar(x, y_cats['politics'], bottom = y_cats['business'] + y_cats['tech'],
plt.bar(x, y_cats['sport'], bottom = y_cats['business'] + y_cats['tech']
        + y_cats['politics'], color='g')
plt.bar(x, y_cats['entertainment'], bottom = y_cats['business'] + y_cats['te
        + y_cats['politics'] + y_cats['sport'], color='m')
plt.xlabel("Predicted Clusters")
plt.ylabel("Count of Category")
```

```
plt.legend(["business", "tech", "politics", "sport", "entertainment"])
plt.title("Unsupervised Categorical Results - News Train")
plt.show()

from sklearn.metrics import accuracy_score

print(f'Accuracy = {accuracy_score(y_true, train_pred["Pred"]):0.2f}')
```



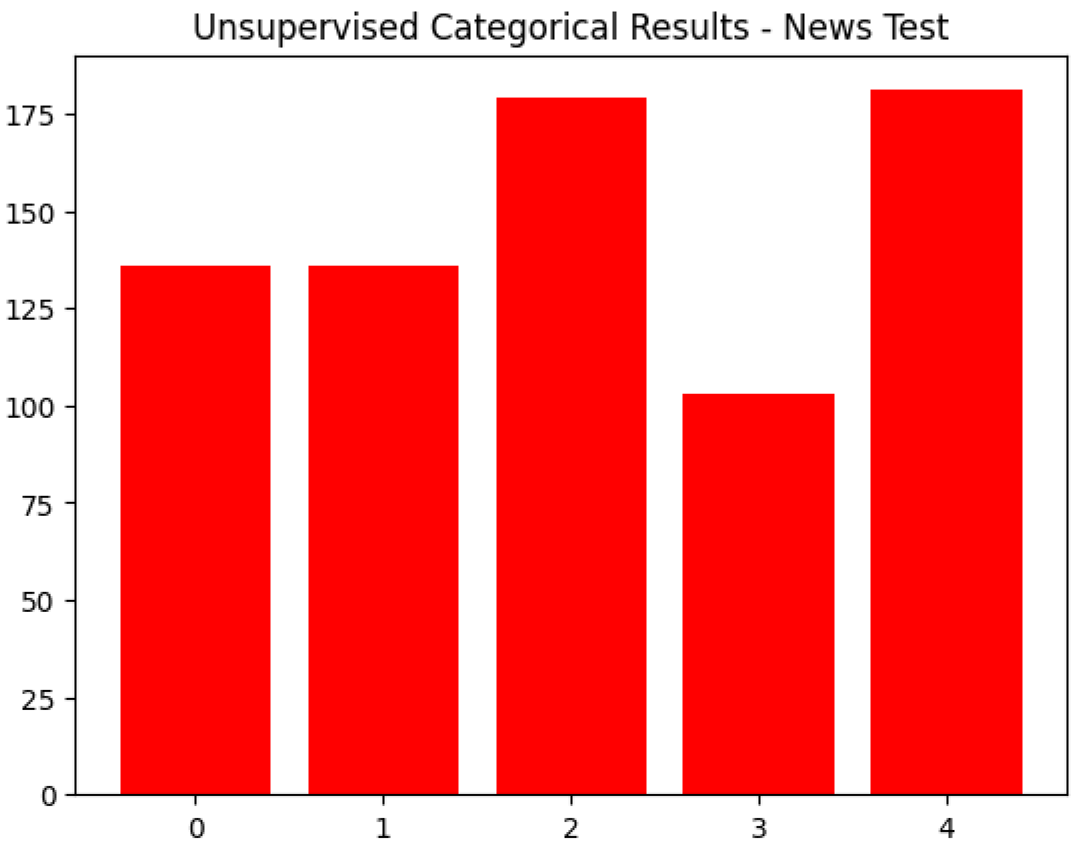

```
Accuracy = 0.92
```

92% accuracy is not bad! Let's try the test data.

Test NVM

In [85]:
```
test_pred = pd.DataFrame(columns={'ArticleId':[],'Pred':[]})

for article_id in news_test.ArticleId:
    pred_list = nmf_categories.loc[article_id].tolist()

    test_pred.loc[len(test_pred)] = [
        article_id,
        pred_list.index(max(pred_list))
    ]


x = [0, 1, 2, 3, 4]

y_cats = []
```

```
for cat in x:
    y_cats.append(len(test_pred[test_pred['Pred'] == cat].copy().index))

plt.bar(x, y_cats, color = 'r')
plt.title("Unsupervised Categorical Results - News Test")
plt.show()
```



This looks fairly balanced. We will need to add back in the categories from the train dataset in order to submit to kaggle and get the results.

In [86]:
```
category_translator = {}
for cat in news_train.Category.unique():
    df = train_pred[train_pred['Category'] == cat].copy()
    category_translator[df['Pred'].mode()[0]] = cat

submission = pd.DataFrame(columns={'ArticleId':[],'Category':[]})
for idx in test_pred.index:
    submission.loc[len(submission)] = [
        test_pred['ArticleId'][idx],
        category_translator[test_pred['Pred'][idx]]
    ]

submission.to_csv('/kaggle/working/submission.csv', index=False)
print(submission)
```

```
     ArticleId       Category
0         1018          sport
1         1319           tech
2         1138          sport
3          459       business
4         1020          sport
..         ...            ...
730       1923       business
731        373  entertainment
732       1704           tech
733        206       business
734        471       politics

[735 rows x 2 columns]
```

This yielded an accuracy of 92.653%, which is right on par with the train dataset!

Can we make it better?

Our nmf data is quite sparse. Scikit-learn's documentation on nmf says that an initialization method of Nonnegative Double Singular Value Decomposition is better for sparse data. Also, from this article, https://www.geeksforgeeks.org/beta-divergence-loss-functions-in-scikit-learn/, it states, "The Kullback-Leibler divergence is a good choice for data that is not perfectly non-negative." And, given the sparseness of our data, it makes sense to give that a shot!

In [91]:
```python
nmf = NMF(n_components = 5, init='nndsvd', beta_loss='kullback-leibler', sol

categories = nmf.fit_transform(unsup_tfidf_df)
nmf_categories = pd.DataFrame(categories, index=unsup_tfidf_df.index)

print(nmf_categories)
```

```
/opt/conda/lib/python3.10/site-packages/sklearn/decomposition/_nmf.py:1524:
UserWarning: The multiplicative update ('mu') solver cannot update zeros pre
sent in the initialization, and so leads to poorer results when used jointly
with init='nndsvd'. You may try init='nndsvda' or init='nndsvdar' instead.
  warnings.warn(
```

```
                    0         1         2         3         4
ArticleId
1833         0.024247  0.003666  0.000000  0.000000  0.073443
154          0.015323  0.000000  0.000000  0.000000  0.119783
1101         0.048865  0.000000  0.000000  0.000000  0.073561
1976         0.090131  0.000000  0.000000  0.000460  0.000000
917          0.026553  0.000000  0.000000  0.000000  0.100420
...               ...       ...       ...       ...       ...
1923         0.031808  0.000649  0.000000  0.000000  0.060395
373          0.000997  0.000000  0.010565  0.104412  0.005761
1704         0.068715  0.000000  0.000000  0.000000  0.000000
206          0.019335  0.000000  0.000000  0.000000  0.067510
471          0.050568  0.105453  0.000000  0.007478  0.000000

[2225 rows x 5 columns]
```

In [92]:

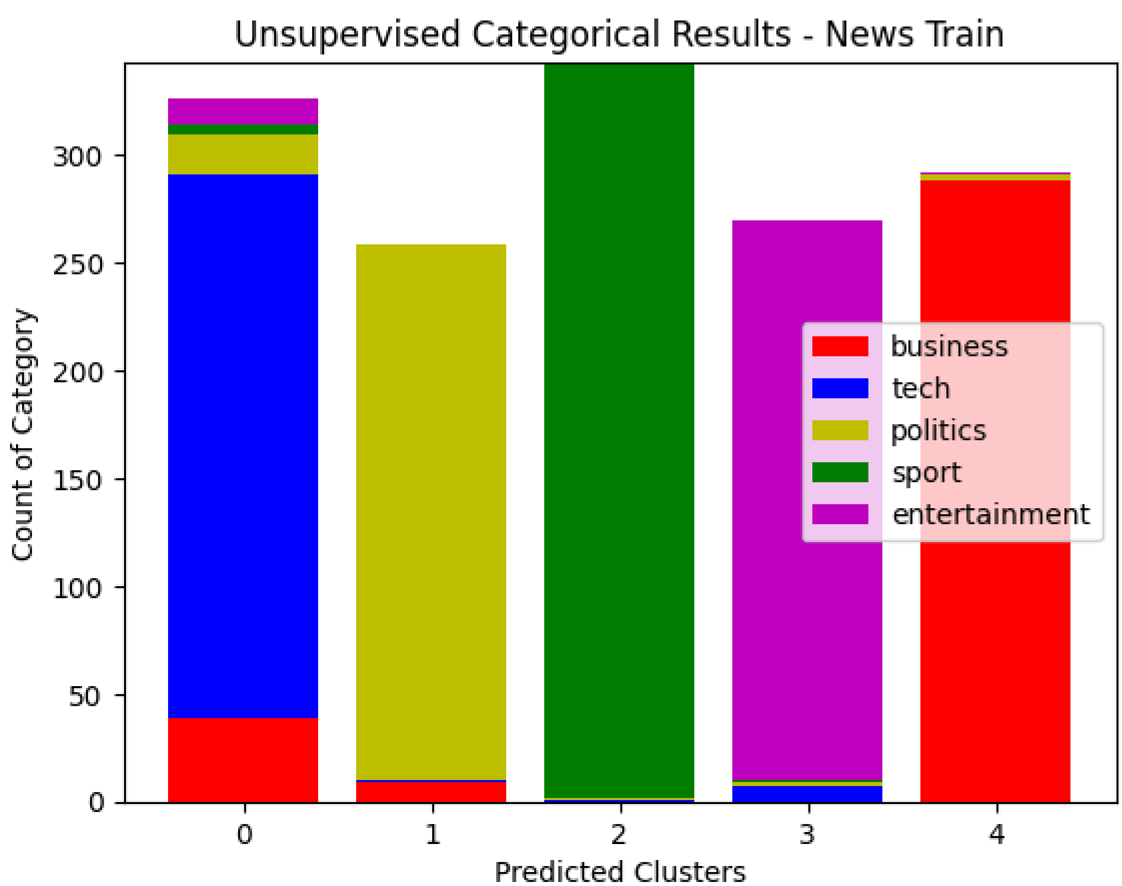
```
train_pred = pd.DataFrame(columns={'ArticleId':[],'Category':[], 'Pred':[]})

for article_id in news_train.ArticleId:
    category = news_train[news_train['ArticleId'] == article_id]['Category']
    pred_list = nmf_categories.loc[article_id].tolist()

    train_pred.loc[len(train_pred)] = [
        article_id,
        category,
        pred_list.index(max(pred_list))
    ]

print(train_pred)
```

```
      ArticleId       Category  Pred
0          1833       business     4
1           154       business     4
2          1101       business     4
3          1976           tech     0
4           917       business     4
...         ...            ...   ...
1485        857  entertainment     3
1486        325  entertainment     3
1487       1590       business     4
1488       1587           tech     0
1489        538           tech     0

[1490 rows x 3 columns]
```

In [93]:

```
y_cats = {}

for category in news_train.Category.unique():
    cat_df = train_pred[train_pred['Category'] == category].copy()
    y_cats[category] = []
    for pred in x:
        y_cats[category].append(len(cat_df[cat_df['Pred'] == pred].index))

    y_cats[category] = np.array(y_cats[category])

y_true = []
```

```
for idx in train_pred.index:
    category = train_pred['Category'][idx]
    y_true.append(
        list(y_cats[category]).index(max(y_cats[category]))
    )

plt.bar(x, y_cats['business'], color = 'r')
plt.bar(x, y_cats['tech'], bottom = y_cats['business'], color='b')
plt.bar(x, y_cats['politics'], bottom = y_cats['business'] + y_cats['tech'],
plt.bar(x, y_cats['sport'], bottom = y_cats['business'] + y_cats['tech']
        + y_cats['politics'], color='g')
plt.bar(x, y_cats['entertainment'], bottom = y_cats['business'] + y_cats['te
        + y_cats['politics'] + y_cats['sport'], color='m')
plt.xlabel("Predicted Clusters")
plt.ylabel("Count of Category")
plt.legend(["business", "tech", "politics", "sport", "entertainment"])
plt.title("Unsupervised Categorical Results - News Train")
plt.show()

from sklearn.metrics import accuracy_score

print(f'Accuracy = {accuracy_score(y_true, train_pred["Pred"]):0.2f}')
```




```
Accuracy = 0.93
```

Adding the nndsvd initialization method and a tighter beta loss range did increase the accuracy a bit, but it took significantly longer for the NVM to fit the data.

Supervised SVM

Now we will run a Support Vector Machine classifier on the cleaned data from above.

In [97]:
```
from sklearn.svm import SVC

sup_tfidf_vect.fit(news_train.Text)

sup_train = sup_tfidf_vect.transform(news_train.Text)

svm = SVC(C=1.0, kernel='linear', degree=3, gamma='auto')
svm.fit(sup_train, news_train.Category)
```

Out[97]:
```
▾ SVC
SVC(gamma='auto', kernel='linear')
```

Now, we will predict the category on the test data.

In [100…
```
sup_test = sup_tfidf_vect.transform(news_test.Text)
y_pred = svm.predict(sup_test)
print(y_pred)
```

```
['sport' 'tech' 'sport' 'business' 'sport' 'sport' 'politics' 'politics'
 'entertainment' 'business' 'business' 'tech' 'politics' 'tech'
 'entertainment' 'sport' 'politics' 'tech' 'entertainment' 'entertainment'
 'business' 'politics' 'sport' 'business' 'business' 'sport' 'business'
 'sport' 'sport' 'business' 'politics' 'tech' 'business' 'business'
 'sport' 'sport' 'sport' 'business' 'entertainment' 'entertainment' 'tech'
 'politics' 'entertainment' 'tech' 'sport' 'tech' 'entertainment'
 'business' 'politics' 'business' 'politics' 'business' 'business'
 'business' 'tech' 'business' 'tech' 'entertainment' 'sport' 'tech'
 'sport' 'entertainment' 'tech' 'politics' 'business' 'entertainment'
 'sport' 'tech' 'sport' 'sport' 'business' 'sport' 'business' 'politics'
 'tech' 'sport' 'tech' 'tech' 'tech' 'entertainment' 'politics' 'sport'
 'entertainment' 'entertainment' 'business' 'entertainment' 'business'
 'entertainment' 'business' 'tech' 'business' 'politics' 'sport' 'tech'
 'sport' 'sport' 'sport' 'sport' 'sport' 'sport' 'politics' 'sport'
 'politics' 'entertainment' 'business' 'sport' 'politics' 'sport'
 'politics' 'entertainment' 'sport' 'business' 'entertainment' 'sport'
 'politics' 'sport' 'politics' 'sport' 'politics' 'business'
 'entertainment' 'business' 'entertainment' 'entertainment' 'tech' 'sport'
 'business' 'entertainment' 'business' 'entertainment' 'business'
 'politics' 'politics' 'tech' 'business' 'business' 'politics' 'tech'
 'entertainment' 'sport' 'business' 'tech' 'sport' 'entertainment'
 'politics' 'sport' 'sport' 'entertainment' 'entertainment' 'tech'
 'business' 'tech' 'politics' 'tech' 'sport' 'sport' 'sport' 'sport'
 'entertainment' 'tech' 'business' 'tech' 'business' 'tech' 'business'
 'tech' 'entertainment' 'tech' 'tech' 'politics' 'business' 'politics'
 'business' 'business' 'entertainment' 'politics' 'tech' 'business'
 'business' 'tech' 'sport' 'politics' 'sport' 'politics' 'tech' 'tech'
 'politics' 'business' 'politics' 'entertainment' 'politics' 'business'
 'entertainment' 'sport' 'tech' 'tech' 'business' 'tech' 'politics'
 'business' 'sport' 'politics' 'business' 'entertainment' 'business'
 'business' 'sport' 'tech' 'business' 'sport' 'entertainment'
 'entertainment' 'sport' 'entertainment' 'sport' 'tech' 'business'
 'entertainment' 'sport' 'entertainment' 'sport' 'entertainment'
 'politics' 'business' 'tech' 'entertainment' 'business' 'politics'
 'business' 'tech' 'business' 'sport' 'politics' 'politics' 'politics'
 'politics' 'sport' 'business' 'business' 'politics' 'sport' 'politics'
 'business' 'sport' 'tech' 'business' 'politics' 'business' 'politics'
 'business' 'business' 'sport' 'tech' 'politics' 'entertainment' 'tech'
 'entertainment' 'tech' 'sport' 'sport' 'tech' 'sport' 'sport' 'sport'
 'entertainment' 'sport' 'politics' 'tech' 'tech' 'sport' 'business'
 'sport' 'business' 'sport' 'entertainment' 'business' 'business'
 'entertainment' 'politics' 'business' 'sport' 'sport' 'tech' 'sport'
 'sport' 'entertainment' 'business' 'sport' 'tech' 'politics'
 'entertainment' 'business' 'business' 'politics' 'sport' 'entertainment'
 'politics' 'business' 'sport' 'sport' 'tech' 'entertainment' 'sport'
 'business' 'tech' 'business' 'sport' 'politics' 'politics'
 'entertainment' 'politics' 'entertainment' 'politics' 'business'
 'politics' 'tech' 'business' 'sport' 'tech' 'entertainment' 'politics'
 'sport' 'politics' 'politics' 'tech' 'politics' 'sport' 'tech' 'politics'
 'tech' 'tech' 'entertainment' 'business' 'tech' 'politics' 'business'
 'politics' 'sport' 'tech' 'entertainment' 'entertainment' 'business'
 'sport' 'tech' 'tech' 'entertainment' 'tech' 'business' 'sport'
 'entertainment' 'tech' 'business' 'politics' 'business' 'tech' 'politics'
 'politics' 'sport' 'business' 'tech' 'sport' 'politics' 'politics'
 'business' 'tech' 'sport' 'politics' 'business' 'politics' 'politics'
```

```
'tech' 'entertainment' 'business' 'business' 'sport' 'sport' 'sport'
'tech' 'tech' 'politics' 'tech' 'tech' 'politics' 'business' 'sport'
'sport' 'entertainment' 'entertainment' 'sport' 'tech' 'tech' 'sport'
'tech' 'entertainment' 'politics' 'tech' 'sport' 'business' 'politics'
'entertainment' 'business' 'tech' 'sport' 'politics' 'business'
'business' 'politics' 'tech' 'sport' 'entertainment' 'business' 'tech'
'business' 'tech' 'sport' 'sport' 'politics' 'business' 'tech' 'sport'
'politics' 'business' 'tech' 'tech' 'politics' 'tech' 'business'
'politics' 'business' 'entertainment' 'business' 'entertainment'
'politics' 'entertainment' 'sport' 'business' 'business' 'business'
'sport' 'entertainment' 'business' 'entertainment' 'entertainment'
'sport' 'tech' 'entertainment' 'business' 'business' 'politics'
'entertainment' 'politics' 'politics' 'sport' 'business' 'business'
'politics' 'entertainment' 'entertainment' 'business' 'business' 'sport'
'politics' 'tech' 'tech' 'politics' 'business' 'sport' 'sport' 'politics'
'sport' 'tech' 'business' 'politics' 'sport' 'politics' 'tech' 'business'
'politics' 'tech' 'politics' 'politics' 'entertainment' 'tech' 'sport'
'sport' 'politics' 'business' 'tech' 'politics' 'sport' 'sport'
'entertainment' 'business' 'entertainment' 'entertainment' 'business'
'politics' 'sport' 'business' 'tech' 'tech' 'business' 'politics' 'sport'
'business' 'sport' 'business' 'politics' 'entertainment' 'sport'
'politics' 'tech' 'sport' 'politics' 'business' 'tech' 'politics' 'sport'
'politics' 'entertainment' 'sport' 'politics' 'business' 'business'
'business' 'tech' 'politics' 'politics' 'sport' 'business' 'tech' 'tech'
'tech' 'sport' 'tech' 'politics' 'tech' 'business' 'sport' 'business'
'politics' 'business' 'tech' 'tech' 'sport' 'tech' 'business' 'sport'
'business' 'business' 'business' 'politics' 'business' 'entertainment'
'entertainment' 'entertainment' 'politics' 'tech' 'tech' 'politics'
'entertainment' 'business' 'sport' 'sport' 'politics' 'entertainment'
'politics' 'sport' 'business' 'business' 'business' 'entertainment'
'tech' 'sport' 'business' 'politics' 'politics' 'tech' 'politics' 'sport'
'politics' 'business' 'tech' 'business' 'sport' 'sport' 'tech' 'sport'
'entertainment' 'tech' 'entertainment' 'tech' 'sport' 'politics'
'business' 'tech' 'politics' 'entertainment' 'entertainment' 'politics'
'business' 'business' 'tech' 'business' 'business' 'business' 'sport'
'entertainment' 'business' 'sport' 'business' 'sport' 'tech' 'business'
'politics' 'sport' 'business' 'sport' 'sport' 'entertainment' 'politics'
'tech' 'sport' 'business' 'sport' 'business' 'sport' 'sport' 'politics'
'tech' 'business' 'tech' 'business' 'sport' 'tech' 'business'
'entertainment' 'business' 'entertainment' 'sport' 'tech' 'business'
'business' 'business' 'politics' 'sport' 'entertainment' 'tech'
'business' 'sport' 'entertainment' 'business' 'entertainment' 'business'
'politics' 'sport' 'sport' 'business' 'tech' 'sport' 'business'
'business' 'business' 'entertainment' 'business' 'entertainment' 'tech'
'sport' 'politics' 'tech' 'politics' 'tech' 'sport' 'tech'
'entertainment' 'business' 'business' 'entertainment' 'politics' 'sport'
'sport' 'sport' 'entertainment' 'tech' 'politics' 'entertainment' 'sport'
'sport' 'politics' 'tech' 'politics' 'entertainment' 'sport'
'entertainment' 'sport' 'tech' 'tech' 'sport' 'sport' 'business' 'tech'
'entertainment' 'business' 'tech' 'business' 'business' 'sport'
'entertainment' 'politics' 'entertainment' 'business' 'politics'
'business' 'politics' 'sport' 'tech' 'tech' 'politics' 'entertainment'
'business' 'tech' 'entertainment' 'entertainment' 'politics' 'business'
'business' 'politics' 'politics' 'tech' 'sport' 'business'
'entertainment' 'politics' 'business' 'politics']
```

In [104…
```
submission = pd.DataFrame(columns={'ArticleId':[], 'Category':[]})
submission['ArticleId'] = news_test.ArticleId.tolist()
submission['Category'] = y_pred
submission.to_csv('/kaggle/working/submission.csv', index=False)
print(submission)
```

```
     ArticleId       Category
0         1018          sport
1         1319           tech
2         1138          sport
3          459       business
4         1020          sport
..         ...            ...
730       1923       business
731        373  entertainment
732       1704       politics
733        206       business
734        471       politics

[735 rows x 2 columns]
```

This had a 97.41% accuracy score! The supervised method certainly performs better and faster.

Discussion

The supervised SVM model performed better and faster than the unsupervised NMF method. The NMF method proved difficult to unpack the results as well, whereas the supervised model can simply be run by svm.predict(), yielding a list of the prediction results. The SVM can be accurate with less data as well.