Categorizaci?n de noticias skl NJ

December 19, 2019

1 Categorización de noticias

1.1 Preguntas a responder con el análisis

- £Se pueden catalogar las noticias con la descripción y los titulares? Compare su clasificación con las categorías incluidas en el set de datos.
- £Existen estilos de escritura asociados a cada categoría?
- £Qué se puede decir de los autores?
- £Qué información útil se puede extraer de los datos?

1.1.1 Análisis exploratorio y carga de los datos

[1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from tensorflow.contrib import learn
import tensorflow as tf
import string
from random import randint
```

[2]:
```
df = pd.read_json("../Data/News_Category_Dataset_v2.json", lines = True,␣
↪convert_dates = True)
```

[3]:
```
df.head()
```

[3]:
```
           authors       category       date  \
0  Melissa Jeltsen          CRIME 2018-05-26
1    Andy McDonald  ENTERTAINMENT 2018-05-26
2       Ron Dicker  ENTERTAINMENT 2018-05-26
3       Ron Dicker  ENTERTAINMENT 2018-05-26
4       Ron Dicker  ENTERTAINMENT 2018-05-26

                                          headline  \
0  There Were 2 Mass Shootings In Texas Last Week...
1  Will Smith Joins Diplo And Nicky Jam For The 2...
2    Hugh Grant Marries For The First Time At Age 57
3  Jim Carrey Blasts 'Castrato' Adam Schiff And D...
4  Julianna Margulies Uses Donald Trump Poop Bags...
```

```
                                                  link  \
0  https://www.huffingtonpost.com/entry/texas-ama...
1  https://www.huffingtonpost.com/entry/will-smit...
2  https://www.huffingtonpost.com/entry/hugh-gran...
3  https://www.huffingtonpost.com/entry/jim-carre...
4  https://www.huffingtonpost.com/entry/julianna-...

                                   short_description
0  She left her husband. He killed their children...
1                           Of course it has a song.
2  The actor and his longtime girlfriend Anna Ebe...
3  The actor gives Dems an ass-kicking for not fi...
4  The "Dietland" actress said using the bags is ...
```

[4]:
```
df.describe()
```

[4]:
```
        authors  category                 date        headline  \
count    200853    200853               200853          200853
unique    27993        41                 2309          199344
top                POLITICS  2013-01-17 00:00:00  Sunday Roundup
freq      36620     32739                  100              90
first       NaN       NaN  2012-01-28 00:00:00             NaN
last        NaN       NaN  2018-05-26 00:00:00             NaN

                                                   link short_description
count                                            200853             200853
unique                                           200812             178353
top     https://www.huffingtonpost.comhttps://www.wash...
freq                                                  2              19712
first                                               NaN                NaN
last                                                NaN                NaN
```

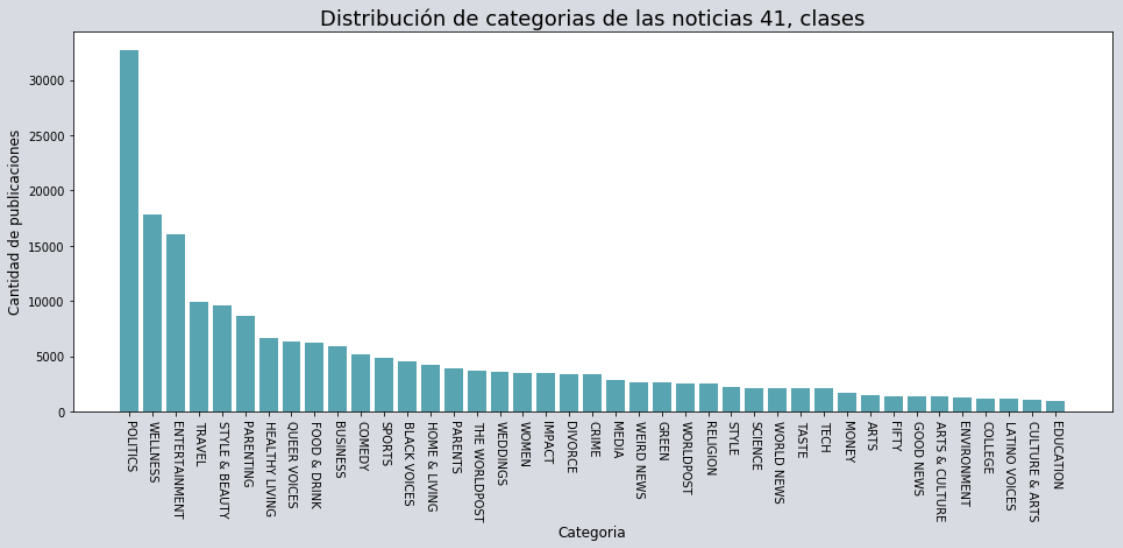
1.1.2 Variable category

[5]:
```
df_temp = pd.DataFrame(pd.value_counts(df['category']))
df_temp['Categoria'] = df_temp.index # organizar top categorias
df_temp = df_temp.rename(columns={'category': 'Frecuencia', 'Categoria':␣
 ↪'Categoria'})
df_temp = df_temp.reset_index(drop = True)
df_temp.head(10)
```

[5]:
```
   Frecuencia       Categoria
0       32739        POLITICS
1       17827        WELLNESS
2       16058   ENTERTAINMENT
3        9887          TRAVEL
4        9649  STYLE & BEAUTY
```

```
5        8677       PARENTING
6        6694  HEALTHY LIVING
7        6314    QUEER VOICES
8        6226    FOOD & DRINK
9        5937        BUSINESS
```

[6]:
```
plt.figure(figsize=(16,6), facecolor = '#d8dbe2') # tamaño del plot
plt.bar(df_temp.Categoria, df_temp.Frecuencia, color = '#58a4b0')
plt.xlabel("Categoria", fontsize = 12)
plt.xticks(rotation=270)
plt.ylabel("Cantidad de publicaciones", fontsize = 12)
plt.title("Distribución de categorias de las noticias {}, clases".
 ↪format(len(df_temp)), fontsize = 18)
plt.show()
```



[7]:
```
uni_c = df["category"].nunique()
len_c =len(df["category"])
len(df)
print("Se tienen {} categorias, y no tienen datos faltantes {} observaciones".
 ↪format(uni_c, len_c))
```

```
Se tienen 41 categorias, y no tienen datos faltantes 200853 observaciones
```

Se observan algunos nombres similares por tanto se limpian para tener menos catergorias y que el dataset este un poco mas balanceado

[8]:
```
def category_group(x):
    if x == 'THE WORLDPOST':
        return 'WORLDPOST'
    elif x == 'PARENTING':
```

```
        return 'PARENTS'
    elif x == 'ARTS' or x == 'CULTURE & ARTS':
        return 'ARTS & CULTURE'
    elif x == 'STYLE':
        return 'STYLE & BEAUTY'
    elif x == 'COLLEGE':
        return 'EDUCATION'
    elif x == 'TASTE':
        return 'FOOD & DRINK'
    elif x == 'WORLD NEWS':
        return 'WORLDPOST'
    else:
        return x

df['category'] = df.category.apply(category_group)
```

[9]:
```
uni_c = df["category"].nunique()
len(df)
print("Al limpiar las categorias se tiene {} categorias".format(uni_c))
```

```
Al limpiar las categorias se tiene 33 categorias
```

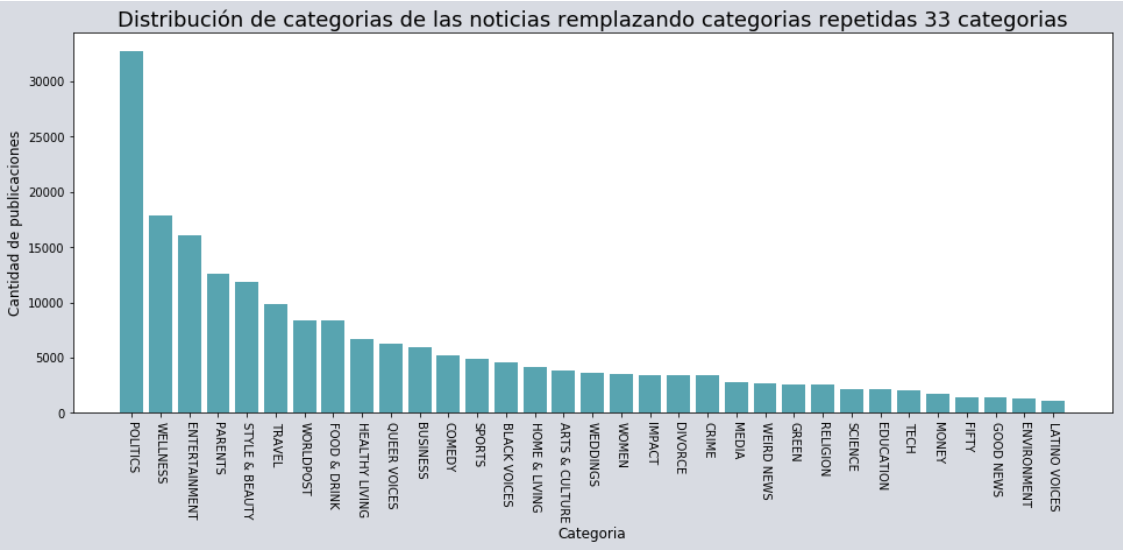
[10]:
```
df_temp_2 = pd.DataFrame(pd.value_counts(df['category']))
df_temp_2['Categoria'] = df_temp_2.index # organizar top categorias
df_temp_2 = df_temp_2.rename(columns={'category': 'Frecuencia', 'Categoria':␣
 ↪'Categoria'})
df_temp_2 = df_temp_2.reset_index(drop = True)
df_temp_2.head(10)
```

[10]:

	Frecuencia	Categoria
0	32739	POLITICS
1	17827	WELLNESS
2	16058	ENTERTAINMENT
3	12632	PARENTS
4	11903	STYLE & BEAUTY
5	9887	TRAVEL
6	8420	WORLDPOST
7	8322	FOOD & DRINK
8	6694	HEALTHY LIVING
9	6314	QUEER VOICES

[11]:
```
plt.figure(figsize=(16,6), facecolor = '#d8dbe2') # tamaño del plot
plt.bar(df_temp_2.Categoria, df_temp_2.Frecuencia, color = '#58a4b0')
plt.xlabel("Categoria", fontsize = 12)
plt.xticks(rotation=270)
plt.ylabel("Cantidad de publicaciones", fontsize = 12)
plt.title("Distribución de categorias de las noticias remplazando categorias␣
 ↪repetidas {} categorias".format(len(df_temp_2)),
          fontsize = 18)
```

```
plt.show()
```



[12]:
```
c1 = len(df[df['category'] == 'ENTERTAINMENT'])
c2 = len(df[df['category'] == 'POLITICS'])
c3 = len(df[df['category'] == 'WORLDPOST'])
print("ENTERTAINMENT = {} noticias, POLITICS = {} noticias, WORLDPOST = {}".
 ↪format(c1, c2, c3))
```

```
ENTERTAINMENT = 16058 noticias, POLITICS = 32739 noticias, WORLDPOST = 8420
```

[13]:
```
p = (c1 + c2 + c3)/len(df)*100
p = np.round(p, 2)
print("Las categorias ENTERTAINMENT, POLITICS y WORLDPOST son el {} % de las␣
 ↪noticias,".format(p))
print("un total de {} noticias".format(c1 + c2 + c3))
```

```
Las categorias ENTERTAINMENT, POLITICS y WORLDPOST son el 28.49 % de las
noticias,
un total de 57217 noticias
```

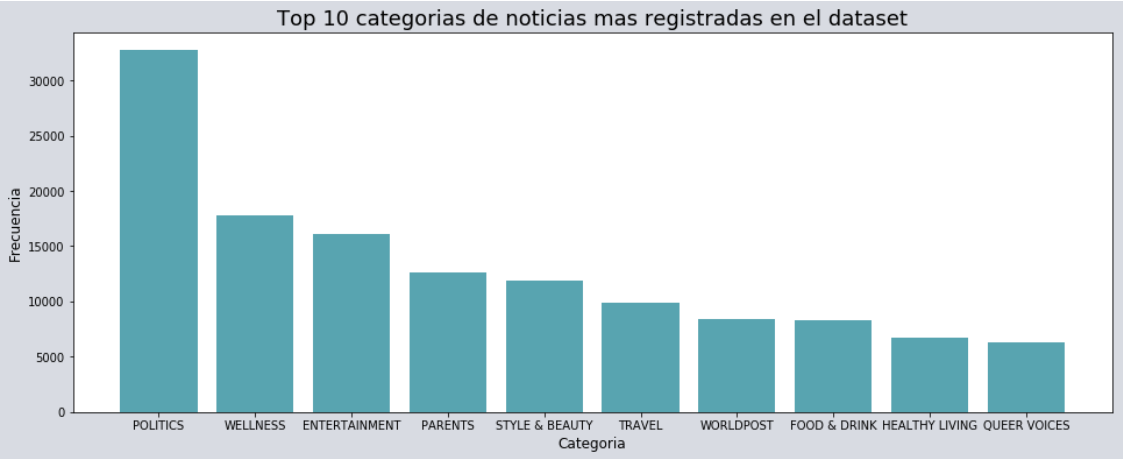
La variable **category** tiene 34 categorias pero nos estan distribuidas de manera uniforme *ENTERTAINMENT, POLITICS* y *WORLDPOST* tienen el 25.58% de todas las noticias

[14]:
```
top_category = pd.DataFrame(pd.value_counts(df['category']))
top_category['Cat'] = top_category.index # organizar top categorias
top_category = top_category.reset_index(drop = True)
top_category.columns = ['Frecuencia', 'Categoria']
top_category = top_category.iloc[0:10]
top_category
```

[14]:

	Frecuencia	Categoria
0	32739	POLITICS
1	17827	WELLNESS
2	16058	ENTERTAINMENT
3	12632	PARENTS
4	11903	STYLE & BEAUTY
5	9887	TRAVEL
6	8420	WORLDPOST
7	8322	FOOD & DRINK
8	6694	HEALTHY LIVING
9	6314	QUEER VOICES

[15]:
```
plt.figure(figsize=(16,6), facecolor = '#d8dbe2') # tamaño del plot
plt.bar(top_category.Categoria, top_category.Frecuencia, color = '#58a4b0')
plt.xlabel('Categoria', fontsize = 12)
plt.ylabel('Frecuencia', fontsize = 12)
plt.title('Top 10 categorias de noticias mas registradas en el dataset',␣
 ↪fontsize = 18)
plt.show()
```



1.1.3 Variable authors

[16]:
```
df["authors"].nunique()
```

[16]: 27993

[17]:
```
pd.value_counts(df['authors'])
```

[17]:
```
                                36620
Lee Moran
2423
Ron Dicker
1913
```

Reuters, Reuters
1562
Ed Mazza
1322
Cole Delbyck
1140
Andy McDonald
1068
Julia Brucculieri
1059
Carly Ledbetter
1054
Curtis M. Wong
1020
Mary Papenfuss
974
Bill Bradley
965
Dana Oliver
936
Sam Levine
893
David Moye
893
Michelle Manetti
876
Michelle Persad
875
Nina Golgowski
868
Igor Bobic
866
Ellie Krupnick
861
Dominique Mosbergen
784
Jamie Feldman
772
James Michael Nichols
764
Caroline Bologna
762
Rebecca Adams
753
Jenna Amatulli
711
Matthew Jacobs

```
702
Ryan Grenoble
698
Daniel Marans
669
Suzy Strutner
650
                              ...
Cara E Bigony, Contributor\nDoctoral Candidate in Counseling Psychology at
Fordham Univers...                    1
Skip Shuda, ContributorDigital market strategizing, new world exploring, big
picture dad                           1
Siddiquah Greene, ContributorPretty girl with the fro and a mind like woah
1
Ali Reff, ContributorOwner of Alice & Wonder, a Chicago-based apparel and gifts
shop                                  1
Rebecca Falconer and Lydia O'Connor
1
Laura Wattenberg, ContributorBaby Name Wizard
1
Idrees Ali and Megha Rajagopalan, Reuters
1
Krystal Hu, Yahoo Finance
1
Jim McKay, ContributorState Director, Prevent Child Abuse West Virginia
1
Kristin Wyatt, Associated Press
1
Liz Spikol, Contributor\nEditor-in-Chief, Curbed Philly
1
James Marshall Crotty, Contributor\nForbes Education Columnist; Author, 'How to
Talk American'; Di...                 1
Hiba Dlewati & Nawar Oliver, Syria Deeply
1
Brian Secemsky, M.D. and Joshua Liao, Contributors
1
Mica Rosenberg and Dan Levine, Reuters
1
Theresa Pickett, ContributorWriter
1
Donna Rockwell, ContributorClinical Psychologist, Mindfulness Teacher, Celebrity
Mental H...                           1
Mesfin Fekadu, AP
1
CafeMom, ContributorCafeMom is the meting place for moms
1
Melissa Sandgren, ContributorFeminist, Policy Nerd
```

```
1
Sarah Klein, Amanda L. Chan, Kate Bratskeir, and Gina Ryder, Contributors
1
Nicole Johnson, ContributorNicole is a fiction writer and blogger.
1
Dr. Rock Positano, Contributor\nHealth Columnist for the New York Daily News
1
Stephanie Sprenger, Contributorwriter, editor, blogger at Mommy, for Real
1
Paul Gale and John Trowbridge
1
Emily Stephenson and Luciana Lopez, Reuters
1
Jack Bouboushian, Courthouse News
1
Mariam Navaid Ottimofiore, ContributorExpat writer and blogger at 'And Then We
Moved To'                          1
Maureen Greenwood-Basken, Contributor\nExecutive Director, Universal Access
Project of the United Nat...        1
Penny Phillips, ContributorUnofficial ambassador for young women striving to
create their...                     1
Name: authors, Length: 27993, dtype: int64
```

[18]:
```
top_authors = pd.DataFrame(pd.value_counts(df['authors']))
top_authors['Aut'] = top_authors.index # organizar top categorias
top_authors = top_authors.reset_index(drop = True)
top_authors.columns = ['Frecuencia', 'authors']
top_authors = top_authors.iloc[0:11]
top_authors
```

[18]:

	Frecuencia	authors
0	36620	
1	2423	Lee Moran
2	1913	Ron Dicker
3	1562	Reuters, Reuters
4	1322	Ed Mazza
5	1140	Cole Delbyck
6	1068	Andy McDonald
7	1059	Julia Brucculieri
8	1054	Carly Ledbetter
9	1020	Curtis M. Wong
10	974	Mary Papenfuss

[19]:
```
plt.figure(figsize=(16,6), facecolor = '#d8dbe2') # tamaño del plot
plt.bar(top_authors.authors, top_authors.Frecuencia, color = '#58a4b0')
plt.xlabel('Autor', fontsize = 12)
plt.ylabel('Frecuencia', fontsize = 12)
plt.title('Top 10 autores de noticias registradas en el dataset', fontsize = 18)
```

```
plt.show()
```

[20]:
```
pd.value_counts(df['authors'] == '')
```

[20]:
```
False    164233
True      36620
Name: authors, dtype: int64
```

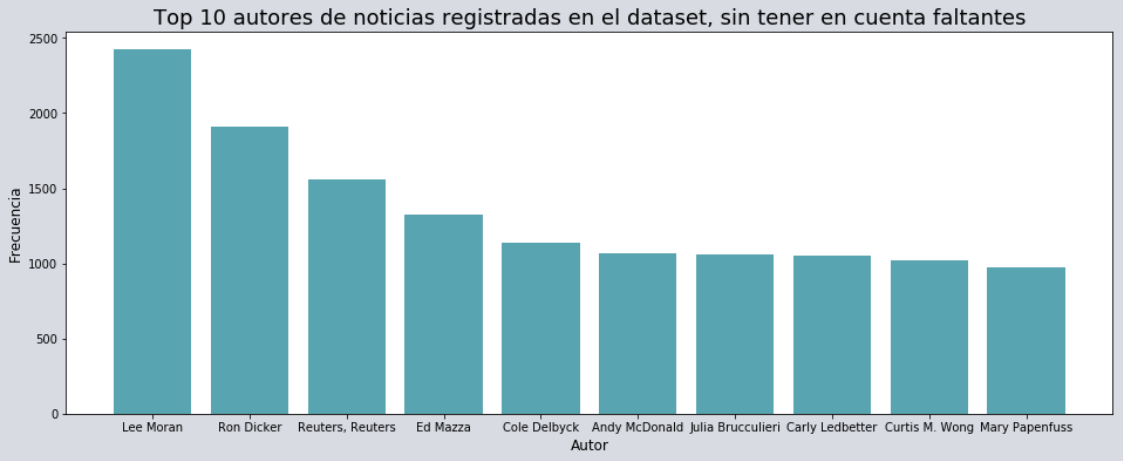
[ ]:
```

```

[21]:
```
36620/ len(df)
```

[21]:
```
0.18232239498538733
```

En el análisis se puede observar que se tienen **36620** registros sin el autor esto equivale al **18.23%** del total de los articulos por esta razón se procede a visualizar los datos sin tener en cuenta los valores que no tienen el autor

[22]:
```
top_authors_nw = top_authors.iloc[1:12] # no se seleccionan los datos sin autor
plt.figure(figsize=(16,6), facecolor = '#d8dbe2') # tamaño del plot
plt.bar(top_authors_nw.authors, top_authors_nw.Frecuencia, color = '#58a4b0')
plt.xlabel('Autor', fontsize = 12)
plt.ylabel('Frecuencia', fontsize = 12)
plt.title('Top 10 autores de noticias registradas en el dataset, sin tener en␣
 ↪cuenta faltantes', fontsize = 18)
plt.show()
```



1.1.4 Variable date

[23]:
```
date = pd.DataFrame(df['date'])
date.head()
```

[23]:
```
        date
0 2018-05-26
1 2018-05-26
2 2018-05-26
3 2018-05-26
4 2018-05-26
```

[24]:
```
type(date)
```

[24]:
```
pandas.core.frame.DataFrame
```

[25]:
```
date['Freq'] = 1
date.index = date['date']
date = date.drop(columns=['date'])
date.head()
```

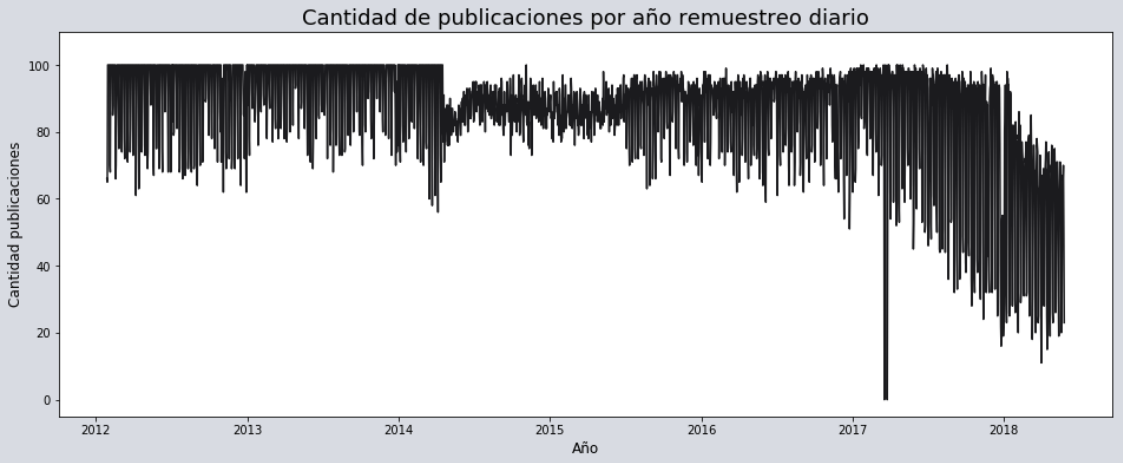
[25]:
```
            Freq
date
2018-05-26     1
2018-05-26     1
2018-05-26     1
2018-05-26     1
2018-05-26     1
```

[26]:
```
date_res_d = date.resample('D').count() # resample por día
date_res_d.head()
```

[26]:
```
            Freq
date
```

```
2012-01-28    66
2012-01-29    65
2012-01-30   100
2012-01-31   100
2012-02-01   100
```

[27]:
```
plt.figure(figsize=(16,6), facecolor = '#d8dbe2') # tamaño del plot
plt.plot(date_res_d, color = '#1b1b1e')
plt.xlabel('Año', fontsize = 12)
plt.ylabel('Cantidad publicaciones', fontsize = 12)
plt.title('Cantidad de publicaciones por año remuestreo diario', fontsize = 18)
plt.ylim([-5, 110])
plt.show()
```



[28]:
```
date_res_m = date.resample('MS').count() # resample por mes
date_res_m.head()
```

[28]:
```
            Freq
date
2012-01-01   331
2012-02-01  2693
2012-03-01  2880
2012-04-01  2769
2012-05-01  2899
```

[29]:
```
plt.figure(figsize=(16,6), facecolor = '#d8dbe2') # tamaño del plot
plt.plot(date_res_m, color = '#1b1b1e')
plt.xlabel('Año', fontsize = 12)
plt.ylabel('Cantidad publicaciones', fontsize = 12)
plt.title('Cantidad de publicaciones por año remuestreo mensual', fontsize = 18)
#plt.ylim([100, 700])
plt.show()
```

Frecuencias de la variable date por mes y por día

[30]:
```
# buscar frecuendia por mes y por día
date_t = pd.DataFrame(df['date'])
```

[31]:
```
date_t['Mes'] = pd.DatetimeIndex(date_t['date']).month_name()
date_t.head()
```

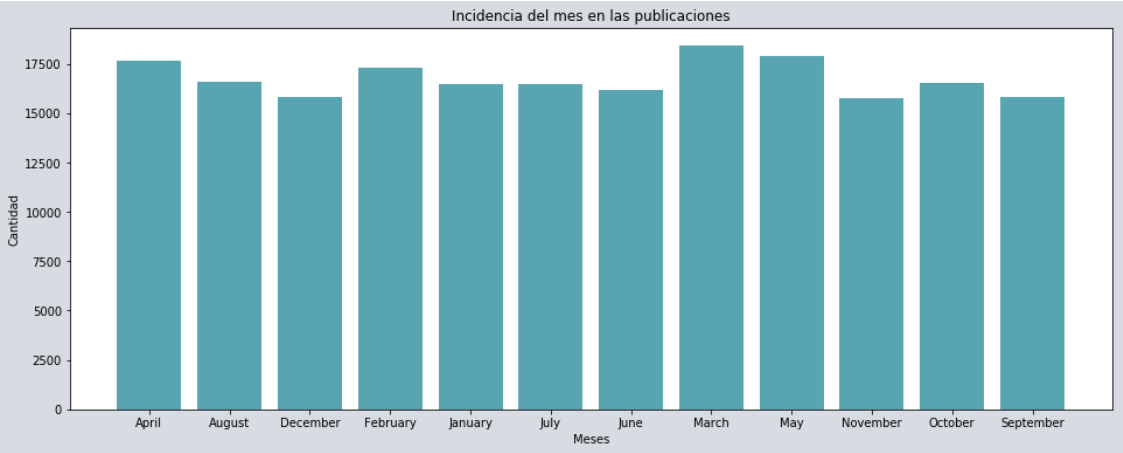
[31]:
```
        date  Mes
0 2018-05-26  May
1 2018-05-26  May
2 2018-05-26  May
3 2018-05-26  May
4 2018-05-26  May
```

[32]:
```
date_t_group = date_t
date_t_group = date_t_group.groupby(['Mes']).count()
date_t_group['Meses'] = date_t_group.index
date_t_group = date_t_group.reset_index(drop = True)
date_t_group
```

[32]:
```
     date      Meses
0   17635      April
1   16576     August
2   15816   December
3   17286   February
4   16493    January
5   16470       July
6   16183       June
7   18418      March
8   17917        May
9   15756   November
10  16505    October
```

```
11  15798  September
```

[33]:
```
plt.figure(figsize=(16,6), facecolor = '#d8dbe2') # tamaño del plot
plt.bar(date_t_group.Meses, date_t_group.date, color = '#58a4b0')
plt.xlabel('Meses')
plt.ylabel('Cantidad')
plt.title('Incidencia del mes en las publicaciones')
plt.show()
```



[34]:
```
mayor = date_t_group.date.max()
menor = date_t_group.date.min()
rango = mayor - menor
print("El mayor valor de publicaciones sumando los meses es: {} y el menor es :␣
 ↪{}, para un rango de {}".
      format(mayor, menor, rango))
```

```
El mayor valor de publicaciones sumando los meses es: 18418 y el menor es :
15756, para un rango de 2662
```

Con respecto a esta variable no se puede ver una estacionalidad clara ni un mes en el que sea mayor la incidencia de publicaciones el rango es de 2662

1.1.5 Variables authors & category

[35]:
```
top_authors_nw.head(10)
```

[35]:

	Frecuencia	authors
1	2423	Lee Moran
2	1913	Ron Dicker
3	1562	Reuters, Reuters
4	1322	Ed Mazza
5	1140	Cole Delbyck
6	1068	Andy McDonald

```
7         1059  Julia Brucculieri
8         1054    Carly Ledbetter
9         1020     Curtis M. Wong
10         974     Mary Papenfuss
```

[36]:
```
aut_cat = pd.merge(top_authors_nw, df) # joint de los datos en funcion de el␣
 ↪top 10 de los autores
```

[37]:
```
top = len(aut_cat)
all_pub = len(df)
aut_porcent = np.round(((top/all_pub) * 100), 2)
print("El top 10 de autores producen el {}% del total de los articulos".
 ↪format(aut_porcent))
```

```
El top 10 de autores producen el 6.74% del total de los articulos
```

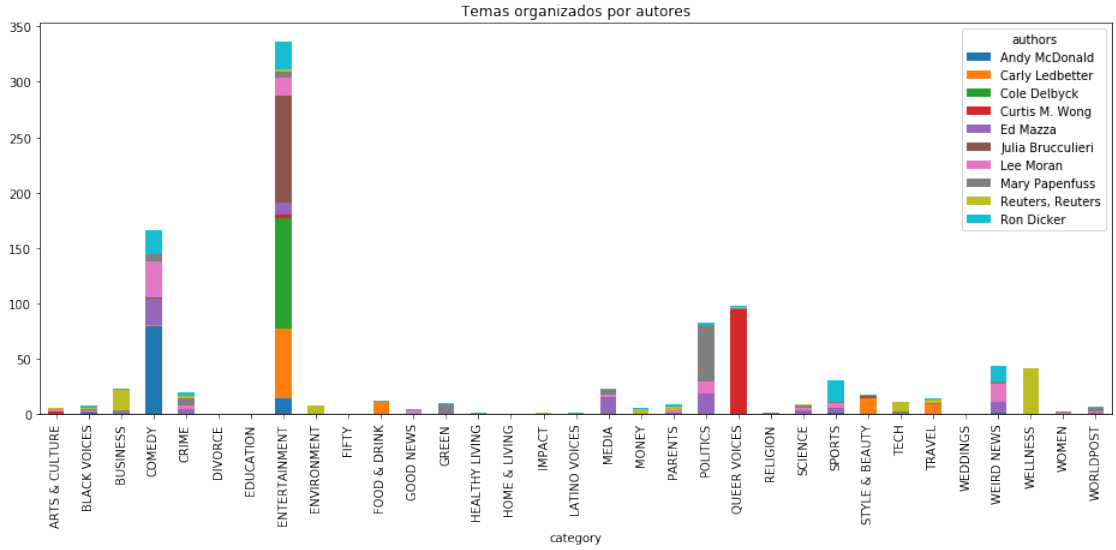
[38]:
```
aut_cat.Frecuencia = 1
aut_cat.head()
```

[38]:
```
   Frecuencia    authors  category       date  \
0           1  Lee Moran    COMEDY 2018-05-24
1           1  Lee Moran    COMEDY 2018-05-24
2           1  Lee Moran  POLITICS 2018-05-24
3           1  Lee Moran    COMEDY 2018-05-24
4           1  Lee Moran     MEDIA 2018-05-24

                                            headline  \
0  'Late Night' Writer's Breathless Royal Wedding...
1  Seth Meyers Gives Donald Trump Some Valuable M...
2  Chrissy Teigen Taunts Donald Trump Over Twitte...
3  Samantha Bee Torches ICE: 'Let's Shut It The F...
4  Jake Tapper Shreds Donald Trump With A Long Li...

                                               link  \
0  https://www.huffingtonpost.com/entry/royal-wed...
1  https://www.huffingtonpost.com/entry/seth-meye...
2  https://www.huffingtonpost.com/entry/chrissy-t...
3  https://www.huffingtonpost.com/entry/samantha-...
4  https://www.huffingtonpost.com/entry/jake-tapp...

                                  short_description
0       Then he bites his lip, like, Hmm, girl.
1             Stop trying to make Fetch happen."
2  Other tweeters, including Mexico's former pres...
3  "Im serious. Its awful. And we dont actuall...
4    "I could go on, but this is just an hour show."
```

Con respecto a estas dos variables se puede ver que el top de los 10 primeros autores en publicar artículos solo cubren el 6.74% de todas las publicaciones, el peso de las publicaciones que no

tienen autores es del 18.23% con 36620 publicaciones con la observación de la variable authors en blanco, esto hace que el 75.03% de las publicaciones tengan una variabilidad muy alta en cuanto a autores lo cual dificulta analiza esta variable

[39]:
```
plot = pd.crosstab(index = aut_cat['category'],
            columns = aut_cat['authors'] # tabla de frecuencias por categoria
                  ).apply(lambda r: r/r.sum() * 100, # promedio ponderado␣
 ↪categorias
                          axis=0).plot(kind='bar', stacked=True,
                                       figsize = (16, 6),
                                       title = "Temas organizados por autores"
                                      )
```



Se puede observar en los datos que los temas donde se publican mas noticias no son los mismo de los autores que mas publican, el top 10 de autores en publicar publican en la categoria ENTRERTAINMENT, esto hace mas desvalanceados los datos y al publicar un articulo en con estos datos la mayor porbabilidad es que sea de politica pero no de un autor del top 10

1.1.6 Variables authors, category & date

[40]:
```
aut_cat['Año'] = pd.DatetimeIndex(aut_cat['date']).year
```

[41]:
```
aut_cat.head()
```

[41]:

	Frecuencia	authors	category	date	\
0	1	Lee Moran	COMEDY	2018-05-24	
1	1	Lee Moran	COMEDY	2018-05-24	
2	1	Lee Moran	POLITICS	2018-05-24	
3	1	Lee Moran	COMEDY	2018-05-24	
4	1	Lee Moran	MEDIA	2018-05-24	

```
                                                headline  \
0  'Late Night' Writer's Breathless Royal Wedding...
1  Seth Meyers Gives Donald Trump Some Valuable M...
2  Chrissy Teigen Taunts Donald Trump Over Twitte...
3  Samantha Bee Torches ICE: 'Let's Shut It The F...
4  Jake Tapper Shreds Donald Trump With A Long Li...

                                                 link  \
0  https://www.huffingtonpost.com/entry/royal-wed...
1  https://www.huffingtonpost.com/entry/seth-meye...
2  https://www.huffingtonpost.com/entry/chrissy-t...
3  https://www.huffingtonpost.com/entry/samantha-...
4  https://www.huffingtonpost.com/entry/jake-tapp...

                                  short_description   Año
0       Then he bites his lip, like, Hmm, girl.  2018
1             Stop trying to make Fetch happen."  2018
2  Other tweeters, including Mexico's former pres...  2018
3  "Im serious. Its awful. And we dont actuall...  2018
4    "I could go on, but this is just an hour show."  2018
```

[42]:
```
Años_pub = aut_cat.Año.unique()
Años_pub
```

[42]:
```
array([2018, 2017, 2016, 2015, 2014, 2013, 2012])
```

[43]:
```
plt.figure(figsize=(16,8), facecolor = '#d8dbe2' ) # tamaño del plot
for i in Años_pub:
    f = aut_cat[aut_cat['Año'] == i]
    plt.scatter(f.authors, f.category, label = i, s = 100, alpha = 0.8)
    plt.legend(loc = 'upper left')
    plt.title("Publicaciones por autor y por año", fontsize = 18)
    plt.xlabel("Año", fontsize = 12)
    plt.ylabel("Categoria", fontsize = 12)
```

Con respecto a estas tres variables se puede observar como los autores que mas publican no son tan especializados y algunos no son tan activos en los diferentes años

1.2 Procesaminto de lenguaje natural de los datos

1.2.1 Alistamiento de los datos para determinar categoría según descripción y los titulares

[44]:
```
df_class = df[['category', 'headline', 'short_description']]
df_class.head()
```

[44]:
```
        category                                              headline  \
0          CRIME  There Were 2 Mass Shootings In Texas Last Week...
1  ENTERTAINMENT  Will Smith Joins Diplo And Nicky Jam For The 2...
2  ENTERTAINMENT    Hugh Grant Marries For The First Time At Age 57
3  ENTERTAINMENT  Jim Carrey Blasts 'Castrato' Adam Schiff And D...
4  ENTERTAINMENT  Julianna Margulies Uses Donald Trump Poop Bags...

                                  short_description
0  She left her husband. He killed their children...
1                          Of course it has a song.
2  The actor and his longtime girlfriend Anna Ebe...
3  The actor gives Dems an ass-kicking for not fi...
4  The "Dietland" actress said using the bags is ...
```

[192]:
```
df_class_joint = df_class
```

[193]:
```
p = df_class.headline + ' ' + df_class.short_description
# p = df_class.headline
df_class_joint['text'] = p
df_class_joint = df_class_joint.drop(columns=['headline','short_description'])
df_class_joint.head()
```

```
/home/oscar/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:3:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/indexing.html#indexing-view-versus-copy
  This is separate from the ipykernel package so we can avoid doing imports
until
```

[193]:

	category	text
0	CRIME	There Were 2 Mass Shootings In Texas Last Week...
1	ENTERTAINMENT	Will Smith Joins Diplo And Nicky Jam For The 2...
2	ENTERTAINMENT	Hugh Grant Marries For The First Time At Age 5...
3	ENTERTAINMENT	Jim Carrey Blasts 'Castrato' Adam Schiff And D...
4	ENTERTAINMENT	Julianna Margulies Uses Donald Trump Poop Bags...

[194]:
```
len(df_class_joint)
```

[194]: 200853

[195]:
```
df_class_joint.head()
```

[195]:

	category	text
0	CRIME	There Were 2 Mass Shootings In Texas Last Week...
1	ENTERTAINMENT	Will Smith Joins Diplo And Nicky Jam For The 2...
2	ENTERTAINMENT	Hugh Grant Marries For The First Time At Age 5...
3	ENTERTAINMENT	Jim Carrey Blasts 'Castrato' Adam Schiff And D...
4	ENTERTAINMENT	Julianna Margulies Uses Donald Trump Poop Bags...

[196]:
```
#limpieza de los datos
df_class_joint.text = [x.lower() for x in df_class_joint.text] # Convertir los␣
 ↪textos a minúscula
df_class_joint.text = [''.join(c for c in x if c not in string.punctuation) for␣
 ↪x in texts] # Eliminamos signos de puntuación
df_class_joint.text = [''.join(c for c in x if c not in '0123456789') for x in␣
 ↪texts] # Eliminamos los números
df_class_joint.text = [' '.join(x.split()) for x in texts] # Eliminar espacios␣
 ↪en blanco y separadores extras
```

[197]:
```
df_class_joint.head()
```

[197]:

	category	text
0	CRIME	there were mass shootings in texas last week b...
1	ENTERTAINMENT	will smith joins diplo and nicky jam for the w...
2	ENTERTAINMENT	hugh grant marries for the first time at age t...
3	ENTERTAINMENT	jim carrey blasts castrato adam schiff and dem...
4	ENTERTAINMENT	julianna margulies uses donald trump poop bags...

1.2.2 Modelo Naive Bayes

[198]:
```
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.naive_bayes import ComplementNB
```

[199]:
```
X_train, X_test, y_train, y_test = train_test_split(df_class_joint['text'],
                                                    df_class_joint['category'],
                                                    random_state = 137,
                                                    train_size = 0.8)
print("X_train {} y X_test {}".format(len(X_train), len(X_test)))
```

```
/home/oscar/anaconda3/lib/python3.7/site-
packages/sklearn/model_selection/_split.py:2179: FutureWarning: From version
0.21, test_size will always complement train_size unless both are specified.
  FutureWarning)

X_train 160682 y X_test 40171
```

[200]:
```
%%time
count_vect = CountVectorizer()
X_train_counts = count_vect.fit_transform(X_train)
tfidf_transformer = TfidfTransformer()
X_train_tfidf = tfidf_transformer.fit_transform(X_train_counts)
clf = ComplementNB(alpha = 0.8).fit(X_train_tfidf, y_train)
```

```
CPU times: user 17.6 s, sys: 110 ms, total: 17.7 s
Wall time: 17.7 s
```

[201]:
```
probe = pd.DataFrame(df_class_joint.text[X_test])
```

[202]:
```
clf.get_params
```

[202]:
```
<bound method BaseEstimator.get_params of ComplementNB(alpha=0.8,
class_prior=None, fit_prior=True, norm=False)>
```

[203]:
```
pos = randint(0, len(probe))
probe.index[pos]
```

[203]:
```
'lessons from the fast lane i was in the mood to try something entirely new and
different something that would shake up my routine and give me a fresh
perspective on life zipping around a racecourse at more than miles an hour
seemed like just the thing provided i didnt kill myself'
```

[204]:
```
pred = clf.predict(count_vect.transform([probe.index[pos]]))
pred[0]
```

[204]:
```
'WELLNESS'
```

[205]:
```
df_class_joint[df_class_joint['text'] == probe.index[pos]]
```

[205]:
```
        category                                                text
168182  WELLNESS  lessons from the fast lane i was in the mood t...
```

[206]:
```
%%time
pred_model = []
for i in range(len(X_test)):
    p = clf.predict(count_vect.transform([X_test.iloc[i]]))
    pred_model.append(p[0])
    if(i)%1000 == 0:
        print("Iteración {}".format(i))
```

```
Iteración 0
Iteración 1000
Iteración 2000
Iteración 3000
Iteración 4000
Iteración 5000
Iteración 6000
Iteración 7000
Iteración 8000
Iteración 9000
Iteración 10000
Iteración 11000
Iteración 12000
Iteración 13000
Iteración 14000
Iteración 15000
Iteración 16000
Iteración 17000
Iteración 18000
Iteración 19000
Iteración 20000
Iteración 21000
Iteración 22000
Iteración 23000
Iteración 24000
Iteración 25000
Iteración 26000
Iteración 27000
Iteración 28000
Iteración 29000
Iteración 30000
Iteración 31000
Iteración 32000
Iteración 33000
Iteración 34000
Iteración 35000
Iteración 36000
```

```
Iteración 37000
Iteración 38000
Iteración 39000
Iteración 40000
CPU times: user 12min 18s, sys: 1.96 s, total: 12min 20s
Wall time: 12min 24s
```

[207]:
```
len(pred_model)
```

[207]:
```
40171
```

[208]:
```
Acc_NB = pd.DataFrame()
```

[209]:
```
Acc_NB['Real'] = y_test
Acc_NB['Predicho'] = pred_model
```

[210]:
```
Acc_NB.head()
```

[210]:

	Real	Predicho
198274	ENTERTAINMENT	ENTERTAINMENT
9281	POLITICS	POLITICS
85726	WORLDPOST	POLITICS
156506	WELLNESS	WELLNESS
72313	LATINO VOICES	STYLE & BEAUTY

[211]:
```
Acierto = []
for i in range(len(Acc_NB)):
    if Acc_NB.Real.iloc[i] == Acc_NB.Predicho.iloc[i]:
        n = 1
    else:
        n = 0
    Acierto.append(n)
```

[212]:
```
Acc_NB['Acierto'] = Acierto
```

[213]:
```
Acc_NB.tail()
```

[213]:

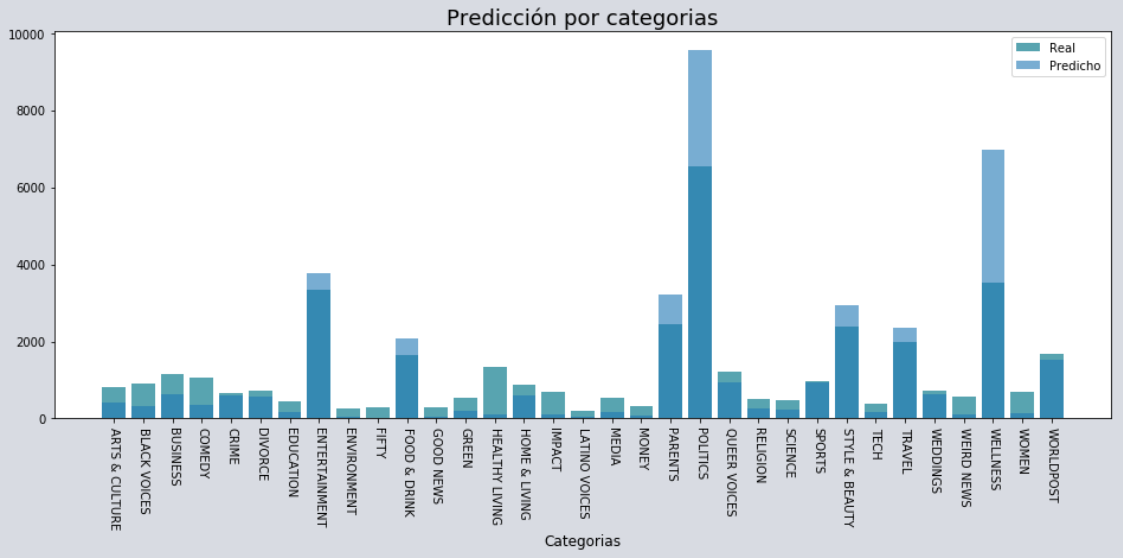
	Real	Predicho	Acierto
73859	POLITICS	POLITICS	1
6393	IMPACT	POLITICS	0
92933	SCIENCE	SCIENCE	1
4603	WORLDPOST	POLITICS	0
168889	WELLNESS	WELLNESS	1

[215]:
```
Acc_mod_NB = np.mean(Acc_NB.Acierto)
print("La efectividad del modelo es de {}%".format(np.round(Acc_mod_NB * 100,␣
 ↪2)))
```

```
La efectividad del modelo es de 61.32%
```

[257]:
```
Acc_NB_Plot_1 = Acc_NB.groupby(['Real']).count()
Acc_NB_Plot_2 = Acc_NB.groupby(['Predicho']).count()
```

[262]:
```
plt.figure(figsize=(16,6), facecolor = '#d8dbe2') # tamaño del plot
plt.bar(Acc_NB_Plot_1.index, Acc_NB_Plot_1.Acierto, label = 'Real', color =␣
↪'#58a4b0')
plt.bar(Acc_NB_Plot_2.index, Acc_NB_Plot_2.Acierto, label = 'Predicho', alpha =␣
↪0.6)
plt.xlabel('Categorias', fontsize = 12)
plt.title('Predicción por categorias', fontsize = 18)
plt.xticks(rotation=270)
plt.legend()
plt.show()
```



Se puede observar como el modelo sobre clasifica algunas categorias y otras no, para próximos analisis es bueno en base a esta información ajustar los pesos, de la categoria del modelo

1.2.3 Modelo Linear SVC

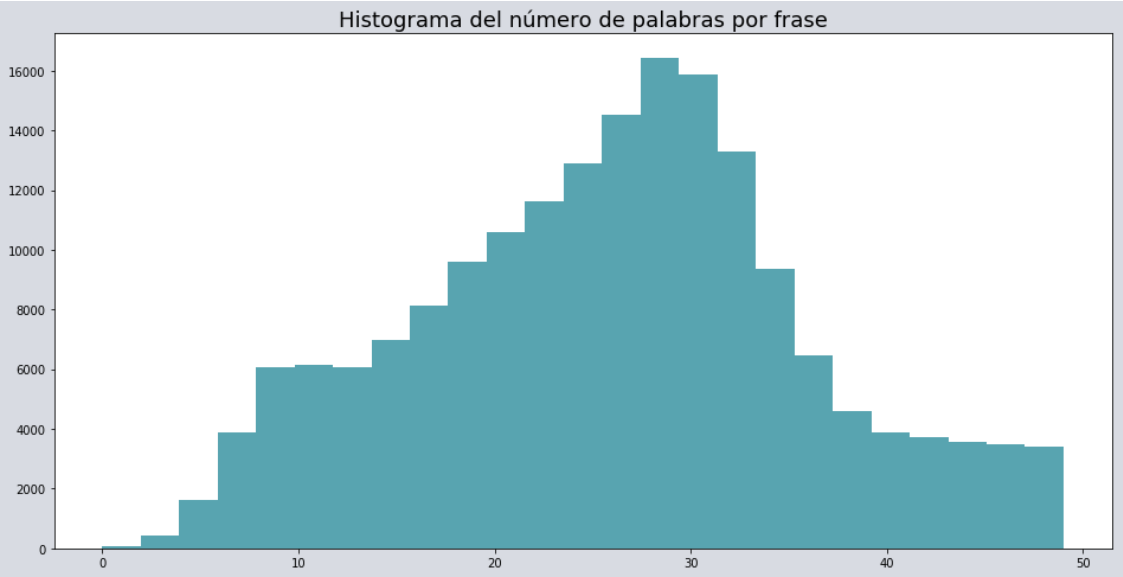
[65]:
```
texts = df_class_joint.text
df_class_joint.head()
```

[65]:

	category	text
0	CRIME	There Were 2 Mass Shootings In Texas Last Week...
1	ENTERTAINMENT	Will Smith Joins Diplo And Nicky Jam For The 2...
2	ENTERTAINMENT	Hugh Grant Marries For The First Time At Age 5...
3	ENTERTAINMENT	Jim Carrey Blasts 'Castrato' Adam Schiff And D...
4	ENTERTAINMENT	Julianna Margulies Uses Donald Trump Poop Bags...

[67]:
```
plt.figure(figsize=(16,8), facecolor = '#d8dbe2') # tamaño del plot
texts_lengths = [len(x.split()) for x in texts]
texts_lengths = [x for x in texts_lengths if x < 50]
plt.hist(texts_lengths, bins=25, color = '#58a4b0' )
```

```
plt.title("Histograma del número de palabras por frase", fontsize = 18)
plt.show()
```



[68]:
```
sentence_size = 48
min_word_freq = 3
```

[69]:
```
vocab_processor = learn.preprocessing.VocabularyProcessor(sentence_size,␣
 ↪min_frequency=min_word_freq)
vocab_processor.fit_transform(texts)
transformed_texts = np.array([x for x in vocab_processor.transform(texts)])
embedding_size = len(np.unique(transformed_texts))
```

```
WARNING:tensorflow:From <ipython-input-69-1c35fd779a57>:1:
VocabularyProcessor.__init__ (from
tensorflow.contrib.learn.python.learn.preprocessing.text) is deprecated and will
be removed in a future version.
Instructions for updating:
Please use tensorflow/transform or tf.data.
WARNING:tensorflow:From /home/oscar/anaconda3/lib/python3.7/site-
packages/tensorflow/contrib/learn/python/learn/preprocessing/text.py:154:
CategoricalVocabulary.__init__ (from
tensorflow.contrib.learn.python.learn.preprocessing.categorical_vocabulary) is
deprecated and will be removed in a future version.
Instructions for updating:
Please use tensorflow/transform or tf.data.
WARNING:tensorflow:From /home/oscar/anaconda3/lib/python3.7/site-
packages/tensorflow/contrib/learn/python/learn/preprocessing/text.py:170:
tokenizer (from tensorflow.contrib.learn.python.learn.preprocessing.text) is
deprecated and will be removed in a future version.
```

```
Instructions for updating:
Please use tensorflow/transform or tf.data.
```

[70]:
```
transformed_texts # texto como bag of words
```

[70]:
```
array([[   70,    73,  1281, ...,     0,     0,     0],
       [   42,  1570,  2859, ...,     0,     0,     0],
       [ 5744,  5103,  7760, ...,     0,     0,     0],
       ...,
       [ 4009,    75,  4942, ...,     0,     0,     0],
       [28704,  1570,  1046, ...,     0,     0,     0],
       [15537,  4538,  3574, ...,     0,     0,     0]])
```

[71]:
```
transformed_texts.shape
```

[71]:
```
(200853, 48)
```

[72]:
```
transformed_texts.size
```

[72]:
```
9640944
```

[73]:
```
category_text = np.array(df_class_joint.category)
category_text
```

[73]:
```
array(['CRIME', 'ENTERTAINMENT', 'ENTERTAINMENT', ..., 'SPORTS', 'SPORTS',
       'SPORTS'], dtype=object)
```

[74]:
```
from sklearn.preprocessing import OneHotEncoder
from sklearn.preprocessing import LabelEncoder

# encoder entero
label_encoder = LabelEncoder()
integer_encoded = label_encoder.fit_transform(category_text)

# encoder binario
onehot_encoder = OneHotEncoder(sparse=False)
integer_encoded = integer_encoded.reshape(len(integer_encoded), 1)
target = onehot_encoder.fit_transform(integer_encoded)
```

```
/home/oscar/anaconda3/lib/python3.7/site-
packages/sklearn/preprocessing/_encoders.py:368: FutureWarning: The handling of
integer data will change in version 0.22. Currently, the categories are
determined based on the range [0, max(values)], while in the future they will be
determined based on the unique values.
If you want the future behaviour and silence this warning, you can specify
"categories='auto'".
In case you used a LabelEncoder before this OneHotEncoder to convert the
categories to integers, then you can now use the OneHotEncoder directly.
  warnings.warn(msg, FutureWarning)
```

[75]:
```
integer_encoded[0]
```

[75]:
```
array([4])
```

[76]:
```
label_encoder.classes_
```

[76]:
```
array(['ARTS & CULTURE', 'BLACK VOICES', 'BUSINESS', 'COMEDY', 'CRIME',
       'DIVORCE', 'EDUCATION', 'ENTERTAINMENT', 'ENVIRONMENT', 'FIFTY',
       'FOOD & DRINK', 'GOOD NEWS', 'GREEN', 'HEALTHY LIVING',
       'HOME & LIVING', 'IMPACT', 'LATINO VOICES', 'MEDIA', 'MONEY',
       'PARENTS', 'POLITICS', 'QUEER VOICES', 'RELIGION', 'SCIENCE',
       'SPORTS', 'STYLE & BEAUTY', 'TECH', 'TRAVEL', 'WEDDINGS',
       'WEIRD NEWS', 'WELLNESS', 'WOMEN', 'WORLDPOST'], dtype=object)
```

[77]:
```
target[0] # categorias como one hot
```

[77]:
```
array([0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

[78]:
```
target.shape
```

[78]:
```
(200853, 33)
```

1.2.4 Modelo Linear SVC

[131]:
```
from sklearn.model_selection import train_test_split
from sklearn.svm import LinearSVC

model = LinearSVC(verbose = 1, max_iter = 300)
```

[132]:
```
X_train, X_test, y_train, y_test = train_test_split(transformed_texts,
                                                    df_class_joint.category,
                                                    test_size = 0.2,
                                                    random_state = 137)
```

[133]:
```
%%time
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

```
[LibLinear]CPU times: user 20min 43s, sys: 2.73 s, total: 20min 45s
Wall time: 20min 52s
```

```
/home/oscar/anaconda3/lib/python3.7/site-packages/sklearn/svm/base.py:922:
ConvergenceWarning: Liblinear failed to converge, increase the number of
iterations.
  "the number of iterations.", ConvergenceWarning)
```

[134]:
```
y_pred.shape
```

[134]:
```
(40171,)
```

[135]:
```
y_pred
```

[135]:
```
array(['WORLDPOST', 'STYLE & BEAUTY', 'POLITICS', ..., 'COMEDY',
       'ENTERTAINMENT', 'PARENTS'], dtype=object)
```

[136]:
```
Acc_SVC = pd.DataFrame()
```

[137]:
```
Acc_SVC['Real'] = y_test
Acc_SVC['Predicho'] = y_pred
```

[138]:
```
Acc_SVC.head()
```

[138]:

	Real	Predicho
198274	ENTERTAINMENT	WORLDPOST
9281	POLITICS	STYLE & BEAUTY
85726	WORLDPOST	POLITICS
156506	WELLNESS	POLITICS
72313	LATINO VOICES	WORLDPOST

[139]:
```
Acierto_SVC = []
for i in range(len(Acc_SVC)):
    if Acc_SVC.Real.iloc[i] == Acc_SVC.Predicho.iloc[i]:
        n = 1
    else:
        n = 0
    Acierto_SVC.append(n)
```

[140]:
```
Acc_SVC['Acierto'] = Acierto_SVC
```

[141]:
```
Acc_SVC.head()
```

[141]:

	Real	Predicho	Acierto
198274	ENTERTAINMENT	WORLDPOST	0
9281	POLITICS	STYLE & BEAUTY	0
85726	WORLDPOST	POLITICS	0
156506	WELLNESS	POLITICS	0
72313	LATINO VOICES	WORLDPOST	0

[142]:
```
Acc_mod_SVC = np.mean(Acc_SVC.Acierto)
print("La efectividad del modelo es de {}%".format(np.round(Acc_mod_SVC * 100,␣
 ↪2)))
```

```
La efectividad del modelo es de 10.99%
```

1.2.5 Red neuronal

[117]:
```
import tflearn # libreria simplificada por encima de tensor flow
```

[118]:
```
X_train_t, X_test_t, y_train_t, y_test_t = train_test_split(transformed_texts,
                                                            target,
                                                            test_size = 0.2,
                                                            random_state = 137)
```

[119]:
```
print(len(X_train_t))
print(len(X_test_t))
print(len(y_train_t))
print(len(y_test_t))
```

```
160682
40171
160682
40171
```

[120]:
```
X_train_t.shape
```

[120]:
```
(160682, 48)
```

[121]:
```
entradas = 48 # 48 por el largo del vector
capa1 = 60 # las neuronas son un valor que se debe ajustar dependiendo del␣
 ↪problema
clases = 33 # estas son las clases para la clasificación al final del␣
 ↪entrenamiento de la red neuronal
learning_Rate = 0.001
```

[122]:
```
def crear_modelo():
    tf.reset_default_graph() # este comando hace un reset en caso de tener algo␣
 ↪en memoria
    red = tflearn.input_data([None, entradas]) # esta es la entrada de los␣
 ↪datos de la red neuronal, None para enviar la información por lotes
    red = tflearn.fully_connected(red, capa1, activation='ReLU') # son las␣
 ↪conexiones de las capas de las red neuronal
    red = tflearn.fully_connected(red, clases, activation='softmax') # capa de␣
 ↪sálida en este caso por ser clasificación la activación es "softmax"
    red = tflearn.regression(red, optimizer = 'adam', # optimizador de␣
 ↪gradiante desendiente estocástico
                             learning_rate = learning_Rate, loss =␣
 ↪'categorical_crossentropy') # la tasa de aprendizaje es un valor pequeño
    modelo = tflearn.DNN(red, tensorboard_verbose = 3) # la variable de modelo␣
 ↪cargar toda la estructura red DNN red neruonal profunda tensorboard_verbose␣
 ↪= 3 para mostrar graficas
    return modelo
```

[123]:
```
modelo = crear_modelo() # para aplicar la función de crear_modelo con los␣
 ↪parametros anteriores
```

[124]:
```
%%time
modelo.fit(X_train_t, y_train_t, # variables de entrenamiento
           validation_set = (X_test_t, y_test_t), # % de datos que dejo para la␣
 ↪validación del modelo, no confundir con los datos de validación, los datos␣
 ↪de validación no los debe ver el modelo por temas de aver fitting
           show_metric = True, # muestra las metricas
           batch_size = 2000, # cantidad de batch de datos que voy a enviar por␣
 ↪iteración al modelo
           n_epoch = 200) # es que tantas veces se va a barrer e conjunto␣
 ↪completo de datos
```

```
Training Step: 16199  | total loss: 2.98672 | time: 1.855s
```

```
| Adam | epoch: 200 | loss: 2.98672 - acc: 0.1937 -- iter: 160000/160682
Training Step: 16200  | total loss: 2.98483 | time: 2.881s
| Adam | epoch: 200 | loss: 2.98483 - acc: 0.1946 | val_loss: 2.99892 - val_acc:
0.1928 -- iter: 160682/160682
--
CPU times: user 11min 48s, sys: 46 s, total: 12min 34s
Wall time: 10min 26s
```

1.2.6 Estilos de escritura

[125]:
```
from PIL import Image
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
```

[126]:
```
label_encoder.classes_
```

[126]:
```
array(['ARTS & CULTURE', 'BLACK VOICES', 'BUSINESS', 'COMEDY', 'CRIME',
       'DIVORCE', 'EDUCATION', 'ENTERTAINMENT', 'ENVIRONMENT', 'FIFTY',
       'FOOD & DRINK', 'GOOD NEWS', 'GREEN', 'HEALTHY LIVING',
       'HOME & LIVING', 'IMPACT', 'LATINO VOICES', 'MEDIA', 'MONEY',
       'PARENTS', 'POLITICS', 'QUEER VOICES', 'RELIGION', 'SCIENCE',
       'SPORTS', 'STYLE & BEAUTY', 'TECH', 'TRAVEL', 'WEDDINGS',
       'WEIRD NEWS', 'WELLNESS', 'WOMEN', 'WORLDPOST'], dtype=object)
```

[127]:
```
len(label_encoder.classes_)
```

[127]:
```
33
```

[128]:
```
p = df[['category', 'headline']]
p.head()
```

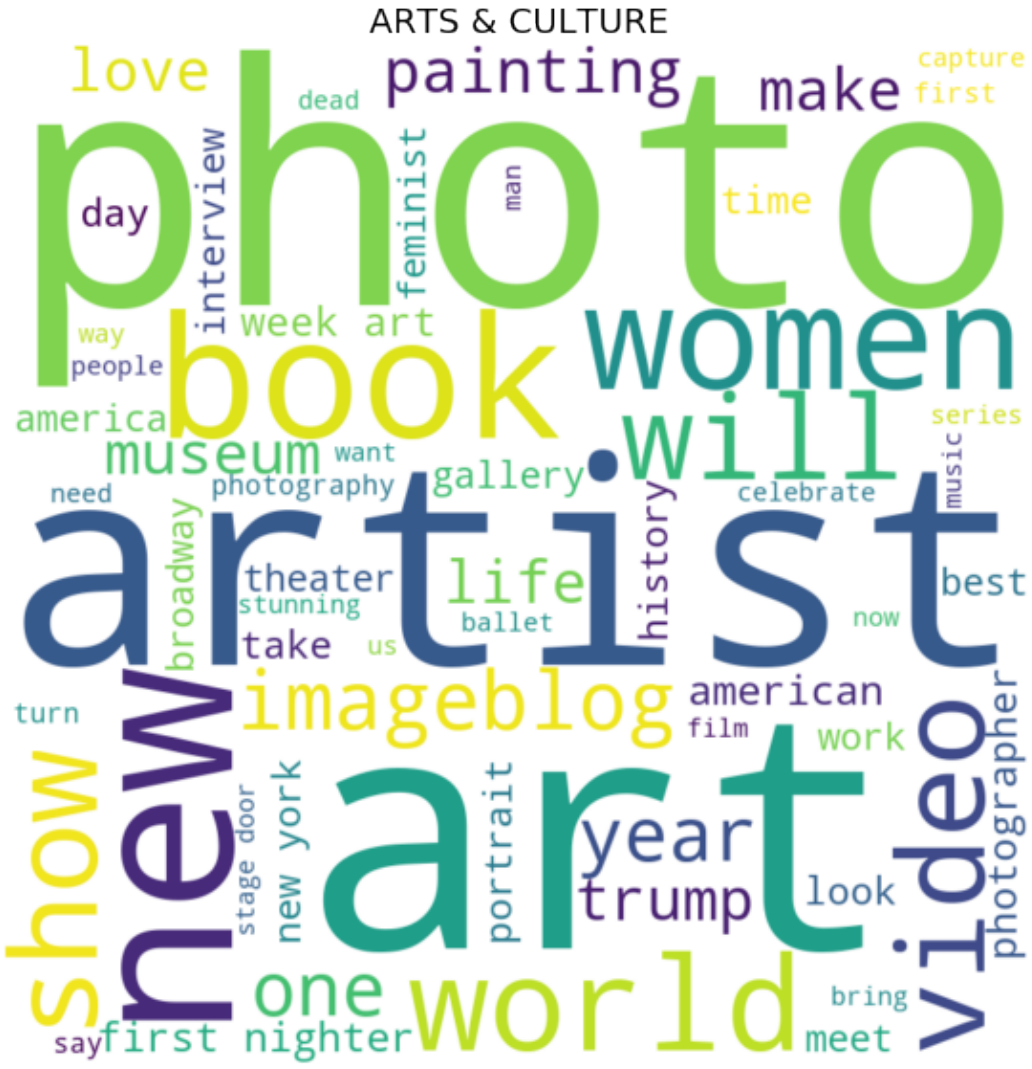
[128]:

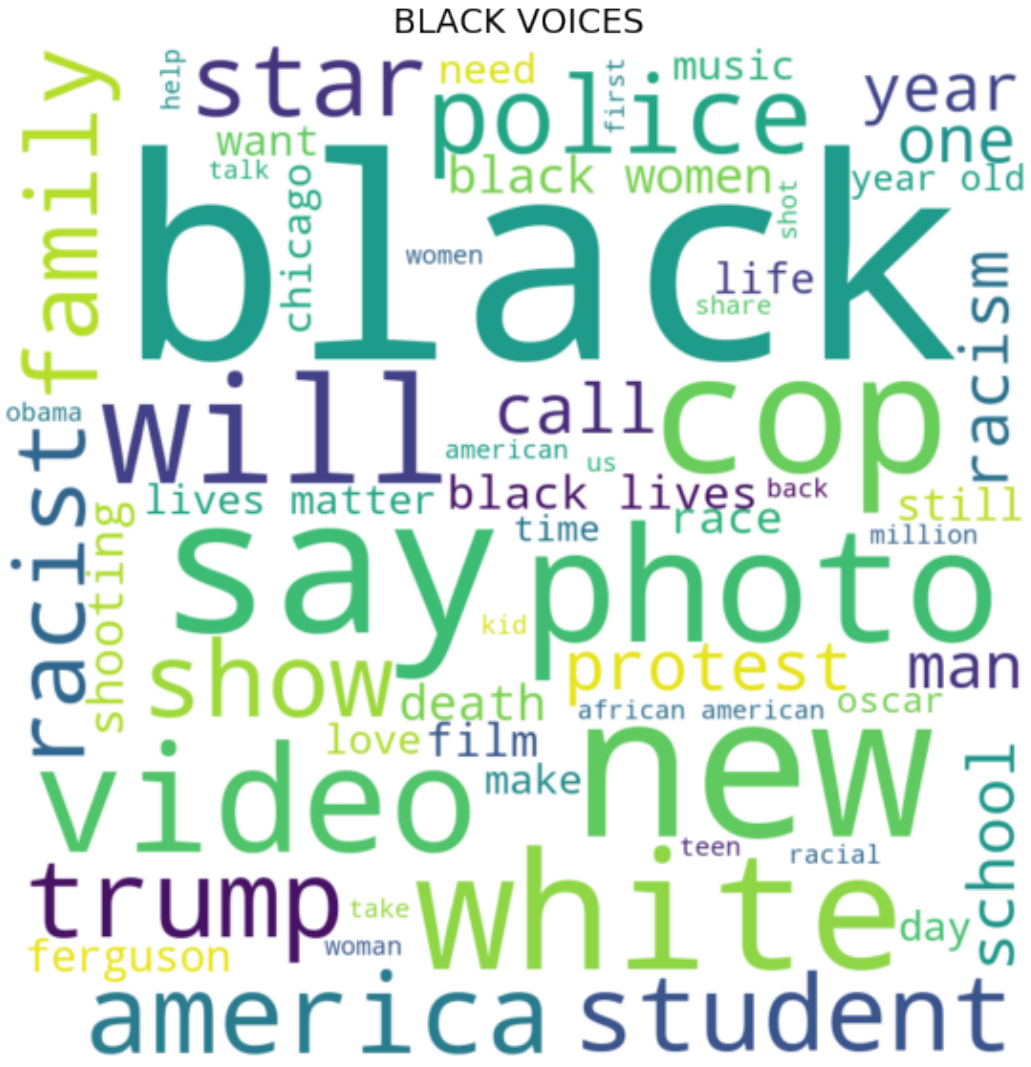
	category	headline
0	CRIME	There Were 2 Mass Shootings In Texas Last Week...
1	ENTERTAINMENT	Will Smith Joins Diplo And Nicky Jam For The 2...
2	ENTERTAINMENT	Hugh Grant Marries For The First Time At Age 57
3	ENTERTAINMENT	Jim Carrey Blasts 'Castrato' Adam Schiff And D...
4	ENTERTAINMENT	Julianna Margulies Uses Donald Trump Poop Bags...

[129]:
```
def my_cloud_wors(category_cloud):
    a = 'category'
    b = category_cloud
    c = p[p[a] == b]

    comment_words = ' '
    stopwords = set(STOPWORDS)

    # iterate through the csv file
    for val in c.headline:

        # typecaste each val to string
        val = str(val)
```

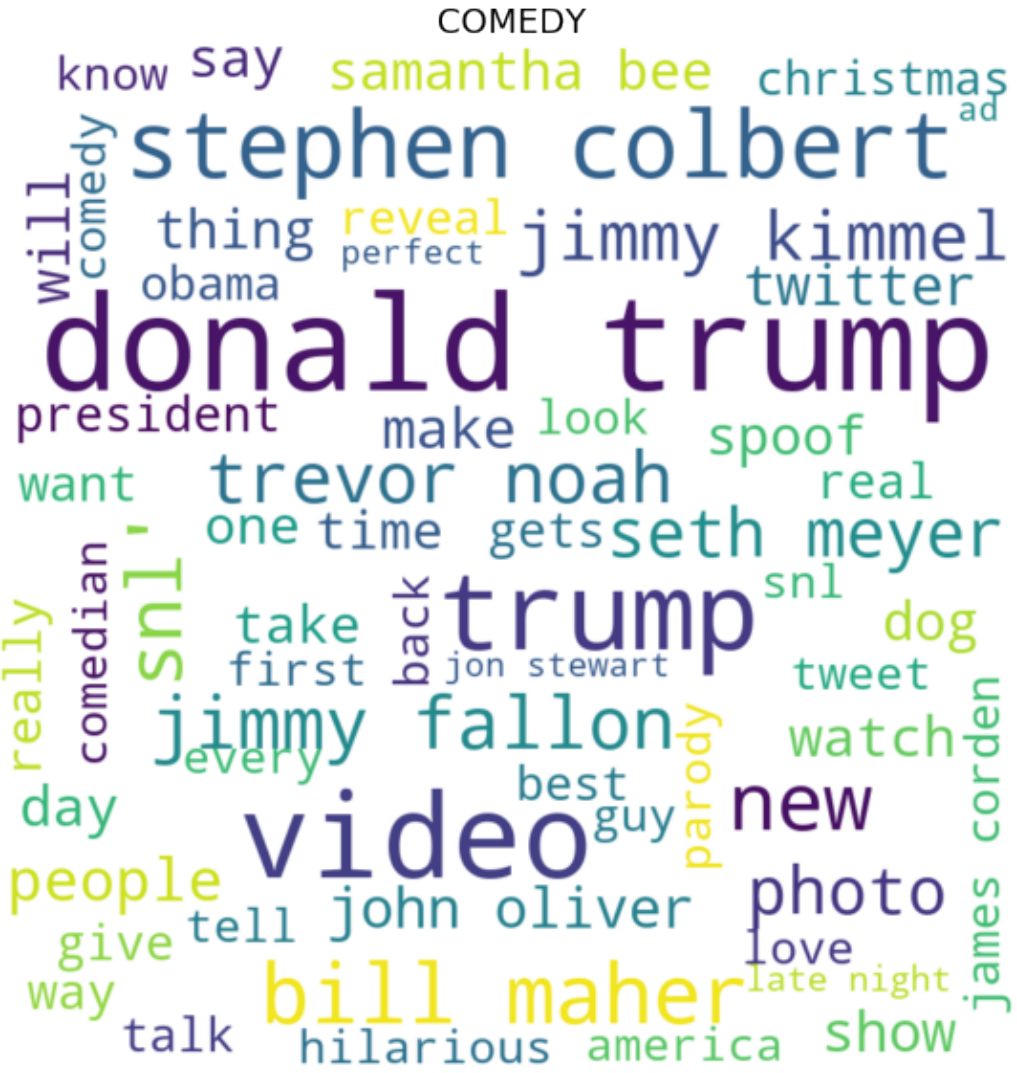
```
        # split the value
        tokens = val.split()

        # Converts each token into lowercase
        for i in range(len(tokens)):
            tokens[i] = tokens[i].lower()

        for words in tokens:
            comment_words = comment_words + words + ' '

    wordcloud = WordCloud(width = 800, height = 800,
                    max_words = 60,
                    background_color ='white',
                    stopwords = stopwords,
                    min_font_size = 10).generate(comment_words)

    # plot the WordCloud image
    plt.figure(figsize = (8, 8), facecolor = None)
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.title(category_cloud, fontsize = 18)
    plt.axis("off")
    plt.tight_layout(pad = 0)

    plt.show()
```

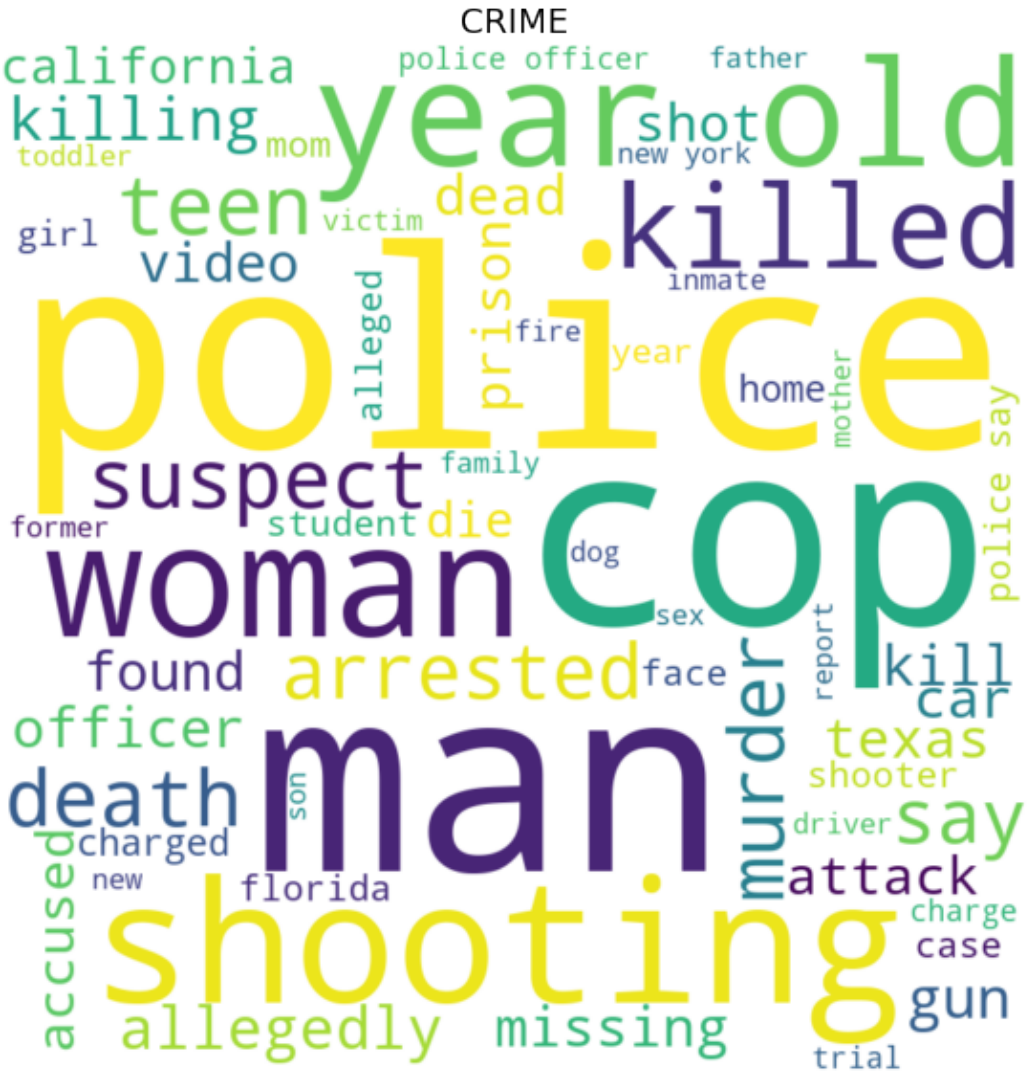
[130]:
```
for  i in label_encoder.classes_:
    my_cloud_wors(i)
```


ARTS & CULTURE
photo
artist
art
new
book
women
will
video
love
painting
make
capture
first
dead
day
interview
feminist
man
time
week
art
way
people
america
museum
want
gallery
photography
need
broadway
theater
stunning
ballet
take
us
life
history
celebrate
music
series
best
now
imageblog
american
film
work
turn
show
year
trump
look
portrait
stage door
new york
photographer
one
world
bring
meet
say
first
nighter

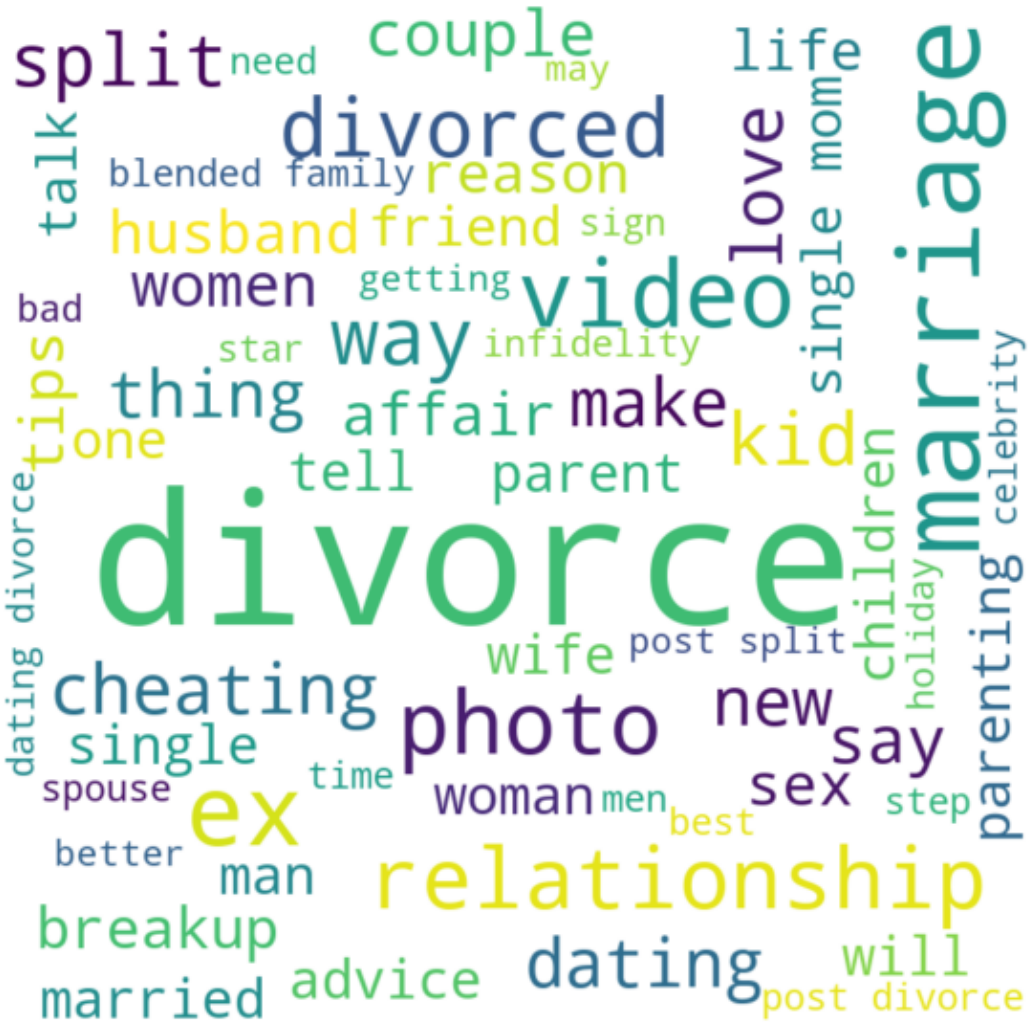

BLACK VOICES
black
police
star
will
cop
say
photo
new
video
white
america
student
trump
show
protest
family
racist
racism
year
one
year old
music
need
help
first
want
talk
chicago
black women
shot
women
life
share
call
obama
american
us
lives matter
black lives
back
still
race
time
million
shooting
kid
man
death
african american
oscar
love
film
make
school
teen
racial
take
day
woman
ferguson


BUSINESS
still
take
wall st
year
want
tax
customer
work
american
job
thing
know
ceo
plan
people
worker
deal
companies
report
company
video
women
business
help
leader
good
billion
time
america
world
state
brand
uber
step
day
secret
economy
founder
way
say
employee
may
make
one
need
will
amazon
walmart
success
business
money
best
reason
pay
women
bank
new
big
marketing
million
wall street

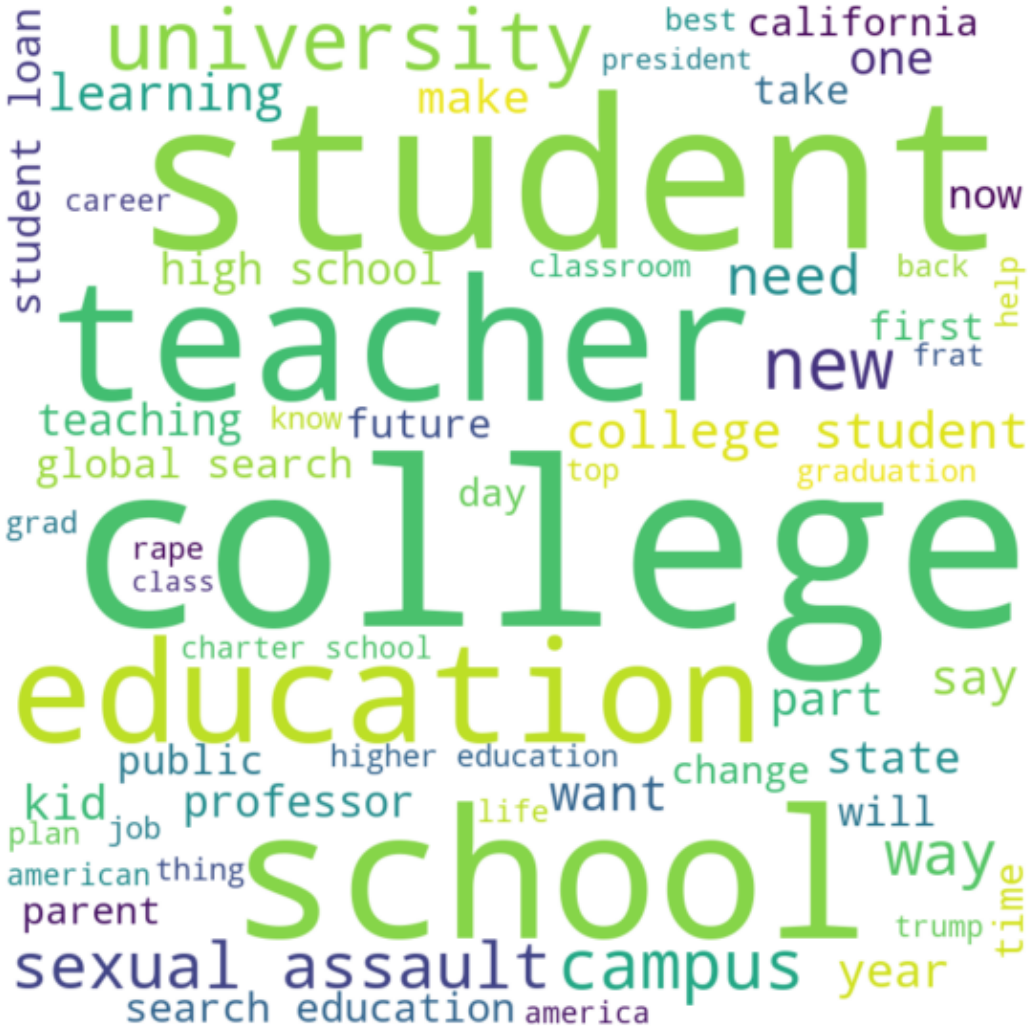

COMEDY
know
say
samantha bee
christmas
ad
stephen colbert
will
comedy
thing
reveal
perfect
jimmy kimmel
obama
twitter
donald trump
president
make
look
spoof
want
trevor noah
real
one
time
gets
seth meyer
snl
trump
really
comedian
snl
take
back
dog
first
jon stewart
tweet
jimmy fallon
watch
every
best
parody
james corden
day
guy
new
video
people
photo
tell
john oliver
give
love
late night
way
bill maher
talk
hilarious
america
show

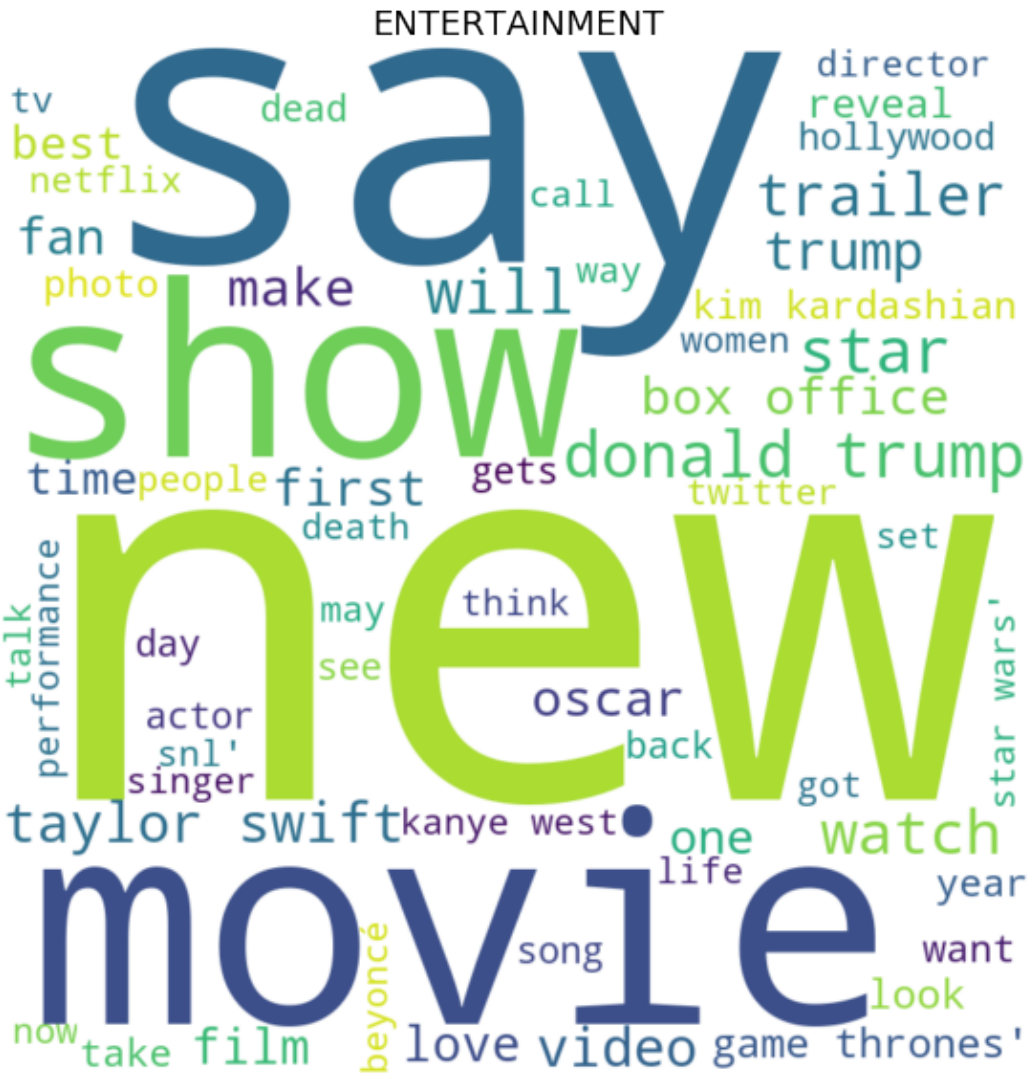

CRIME
police officer
father
california
killing
year
shot
old
toddler
mom
new york
teen
dead
victim
girl
video
killed
police
alleged
prison
inmate
fire
year
home
mother
police say
suspect
family
former
student
die
cop
woman
dog
sex
found
arrested
face
murder
report
kill
car
officer
man
texas
shooter
death
son
driver
say
charged
attack
accused
new
florida
charge
shooting
case
gun
allegedly
missing
trial

DIVORCE


split
need
couple
may
life
marriage
talk
divorced
blended family
reason
love
single mom
husband
friend
sign
women
getting
video
bad
way
infidelity
star
tips
thing
affair
make
kid
one
tell
parent
divorce
dating divorce
celebrity
children
wife
post split
holiday
cheating
photo
new
say
parenting
single
spouse
time
woman
men
sex
step
ex
best
better
man
relationship
breakup
dating
will
married
advice
post divorce

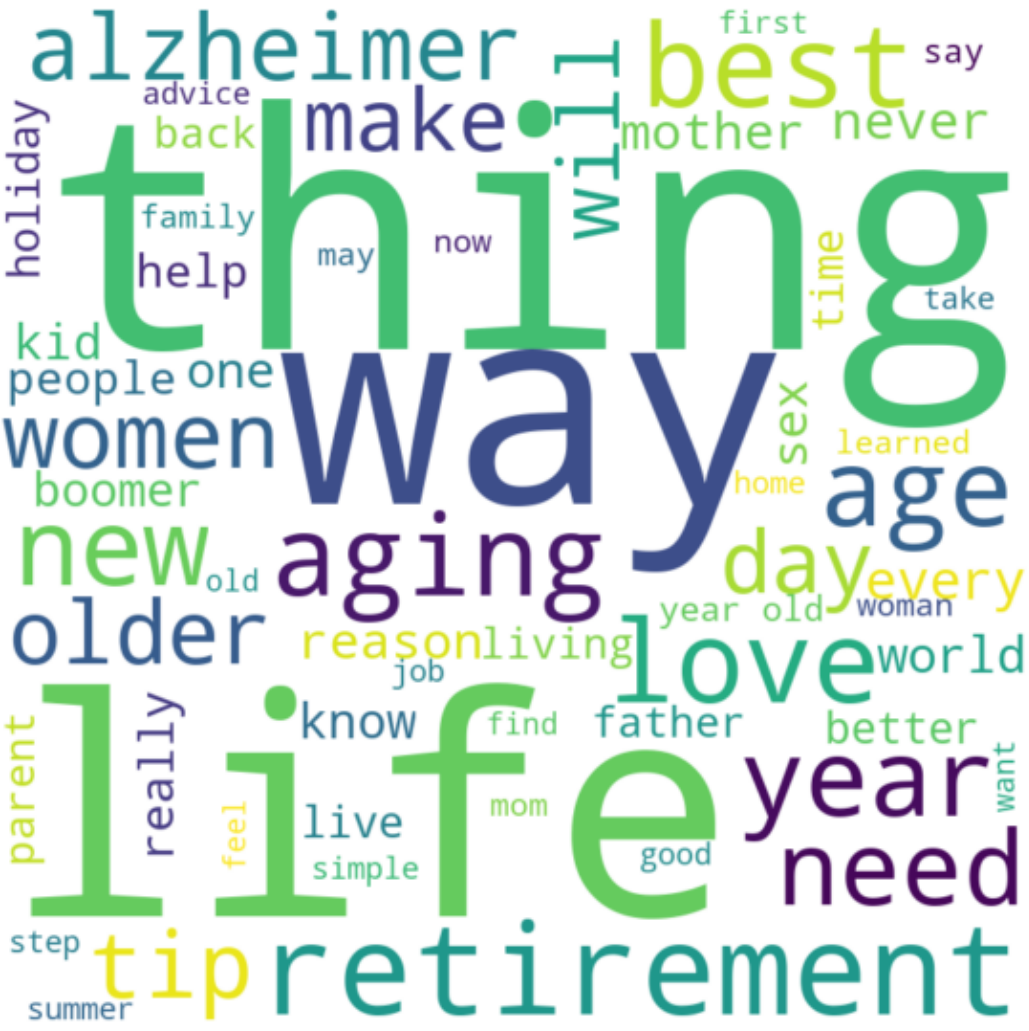
EDUCATION


university
best
california
president
one
learning
make
take
student loan
student
career
now
high school
classroom
need
back
help
teacher
first
new
frat
teaching
know
future
college student
global search
top
graduation
day
college
grad
rape
class
charter school
education
part
say
public
higher education
change
state
kid
professor
want
life
will
plan
job
way
american
thing
parent
time
trump
sexual assault
campus
year
search education
america


ENTERTAINMENT
say
director
tv
dead
reveal
best
hollywood
netflix
call
trailer
fan
trump
photo
make
will
way
kim kardashian
show
women
star
box office
donald trump
time
people
first
gets
twitter
death
set
new
talk
performance
may
think
day
see
oscar
actor
back
snl'
singer
got
star wars'
taylor swift
kanye west
one
watch
life
movie
year
song
want
look
beyoncé
now
take
film
love
video
game thrones'


ENVIRONMENT
climate change
video
photos week
extreme weather
weather photo
climate animal
animal photo
pictures
world
will
dog
storm
show
food
pet
water
fracking
free
tip
first
ocean
year
tornado
time
take
energy
way
obama
video animal
panda
big
snow
global warming
sea
make
hurricane sandy
act
earth day
day
home
oil
zoo
species
photo
life
national
cat
keystone xl
new
baby
green
whale
may
say
elephant
shark
capture
arctic
find

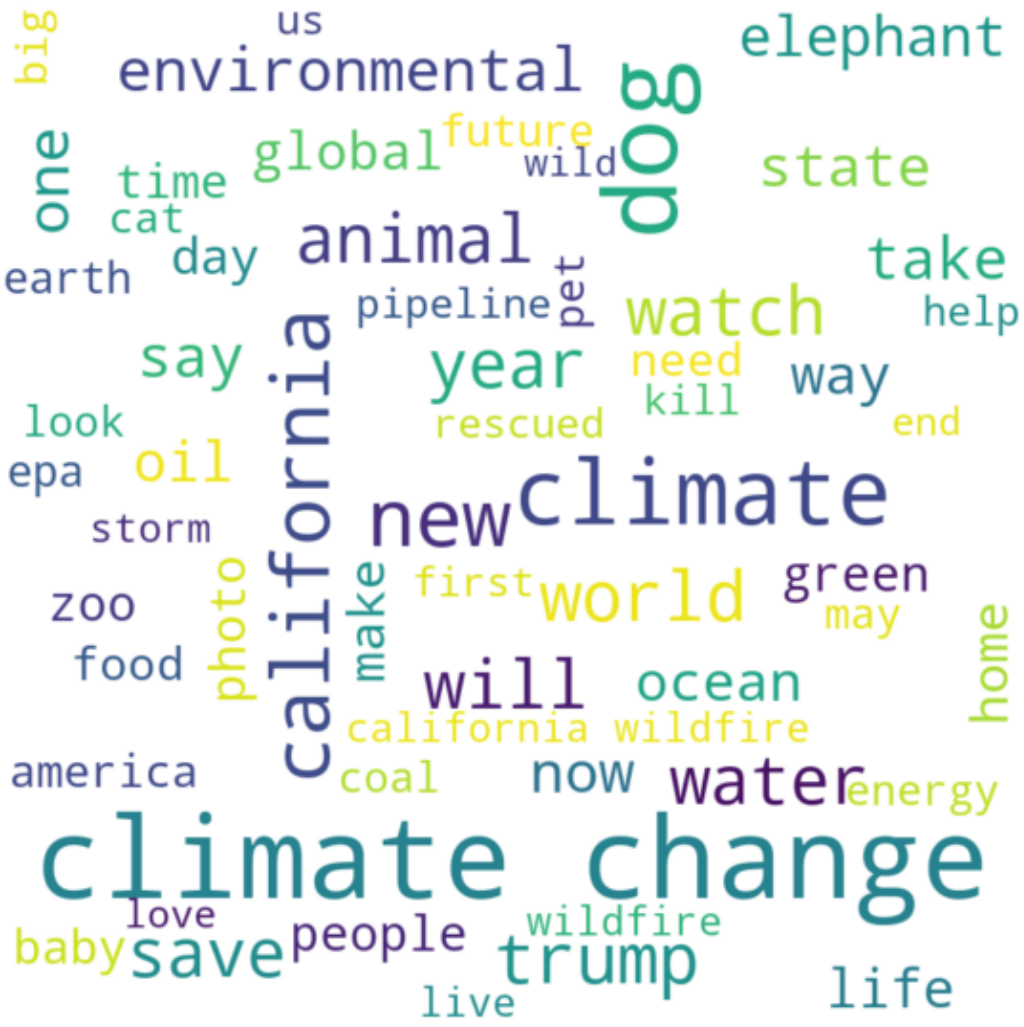
FIFTY

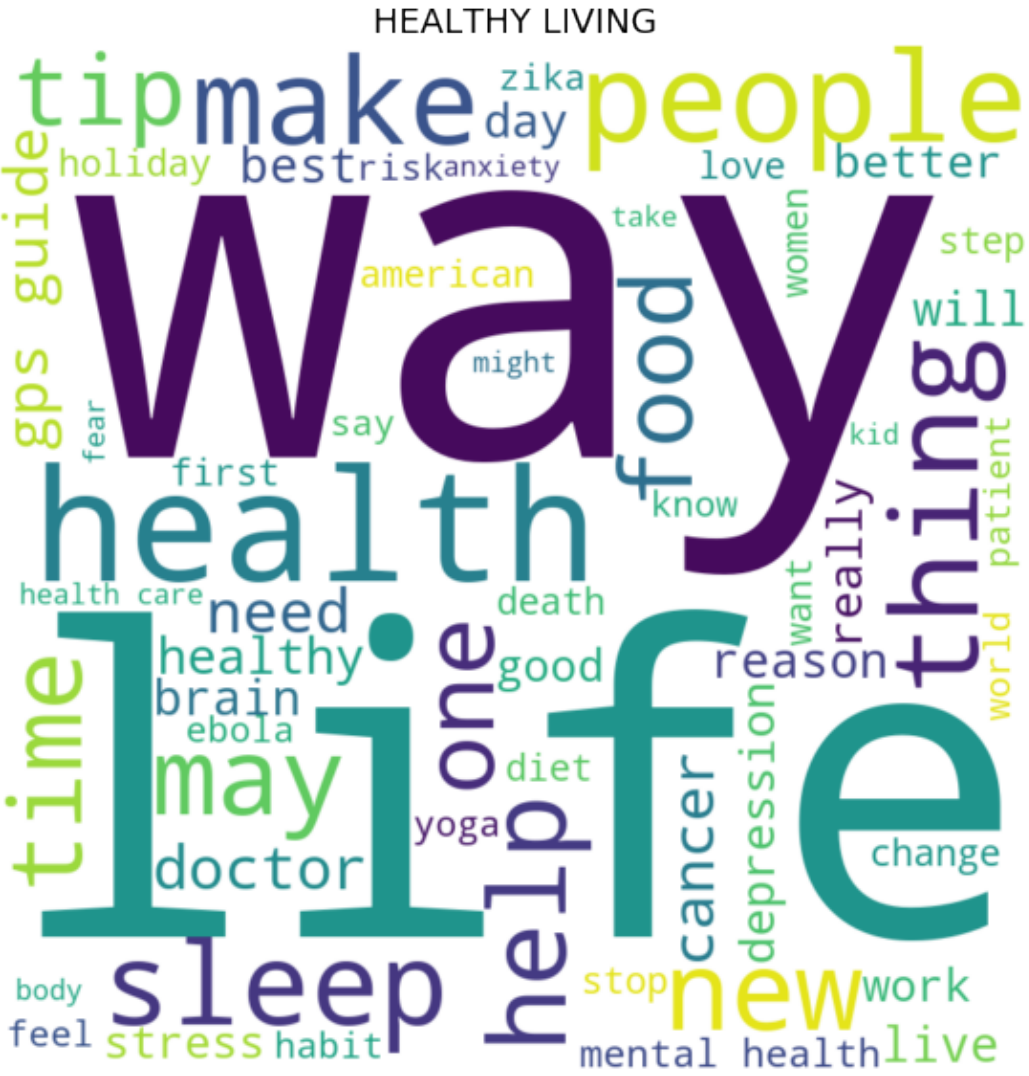

alzheimer
best
first
say
advice
make
will
mother
never
back
holiday
thing
family
help
may
now
time
take
kid
people
one
way
women
sex
learned
boomer
home
age
new
aging
day
every
old
year
old
woman
older
reason
living
love
world
job
know
find
father
better
life
really
parent
year
want
feel
live
mom
need
good
simple
step
tip
retirement
summer


FOOD & DRINK
cooking
summer
cooking cuff
better
beer
restaurant
drink
cheese
cake
pie
cream
salad
eat
food
soup
ice
chocolate
breakfast
day
way
chicken
pizza
need
best
make
dinner
delicious
recipe day
favorite
bacon
recipe
love
time
perfect
thanksgiving
dessert
recipes photo
healthy
coffee
video
new
wine
cookie
thing
cook
good
burger
one
holiday
photo
egg
chef
tip
will
easy
taste test
cocktail
top
taste
huffpost tastemakers
kitchen


GOOD NEWS
teen
saved
mom
adorable
family
show
cop
life
man
photo
gets
school
take
puppy
video
world
girl
find
car
turn
boy
day
cat
rescue
pit bull
homeless
rescued
make
people
watch
dog
kitten
pup
year old
time
buy
give
way
new
use
free
friend
save
student
home
stranger
year
gift
help
love
kid
animal
now
cancer
woman
baby
best
need
christmas
one

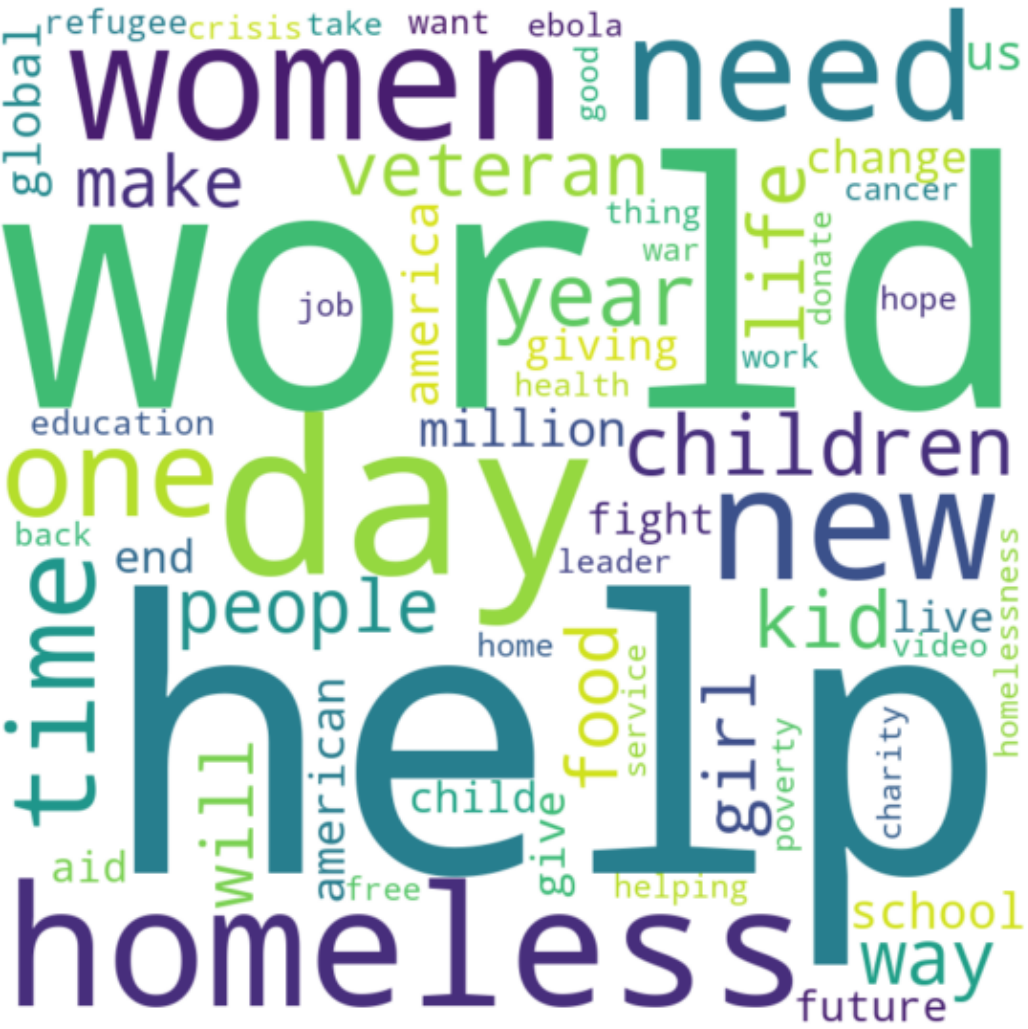
GREEN


big
us
elephant
environmental
dog
one
global
future
wild
state
time
cat
animal
take
earth
day
pipeline
pet
watch
help
say
year
need
way
california
look
rescued
kill
end
epa
oil
new
climate
storm
photo
make
first
world
green
zoo
may
food
will
ocean
home
california wildfire
america
coal
now
water
energy
climate change
love
wildfire
baby
save
people
trump
life
live

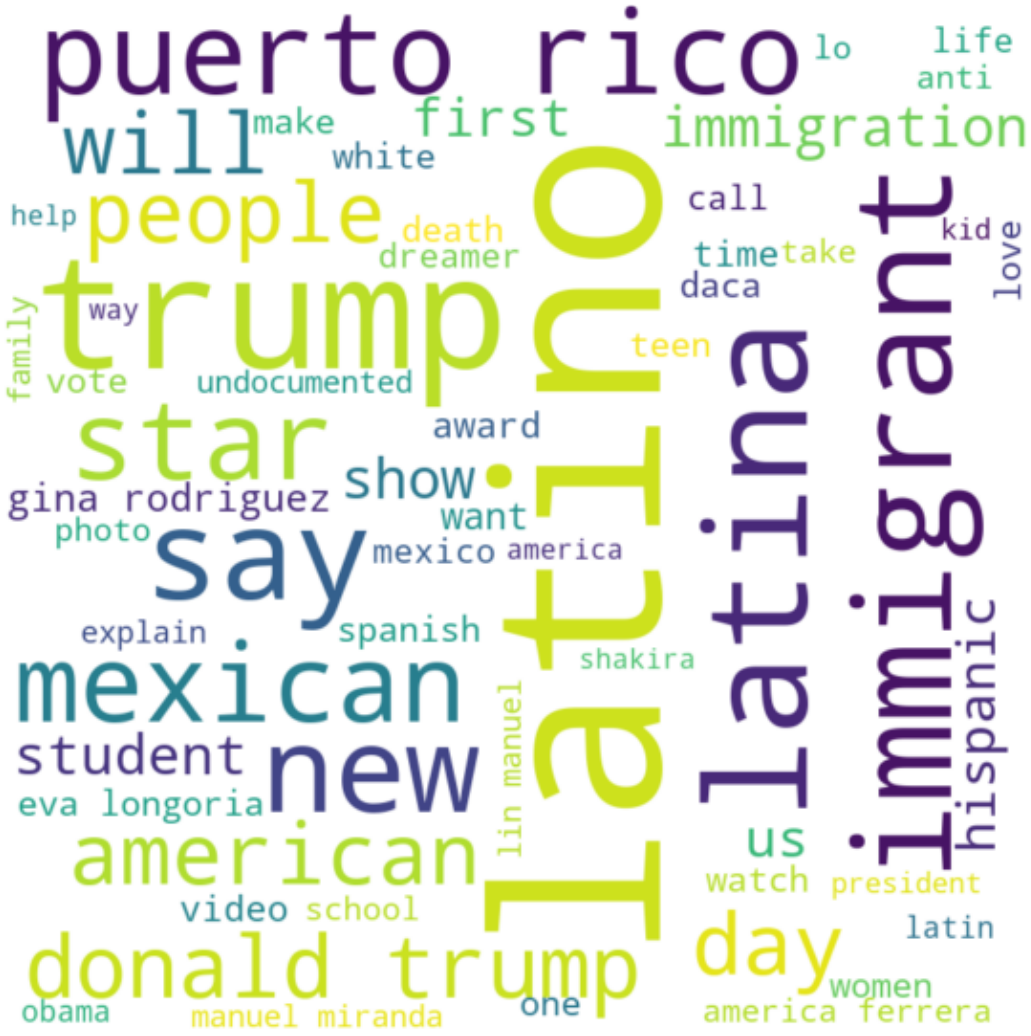

HEALTHY LIVING
way
life
health
people
make
tip
thing
time
help
sleep
new
one
may
food
doctor
need
healthy
brain
good
reason
cancer
depression
stress
habit
mental health
live
work
change
yoga
diet
death
ebola
zika
day
holiday
best
risk
anxiety
love
better
take
women
step
american
will
might
say
kid
first
know
fear
gps guide
patient
world
want
really
health care
body
feel
stop


HOME & LIVING
home finds
video
top
sprout home
show
style
us
look
life
make
remove
create
diy idea
weekend diy
outdoor
idea
space
way
furniture
best
photos craft
holiday
tiny home
country living
clean
design
makeover
color
new
home
one
gift idea
sale
summer
ebay vintage
roundup ebay
craft day
garden
new use
vintage home
house tour
finds photos
decor
photo
diy
spring
table
party
use
worth
thing
kitchen tip
perfect
designer
room
house
weekly roundup
will
apartment

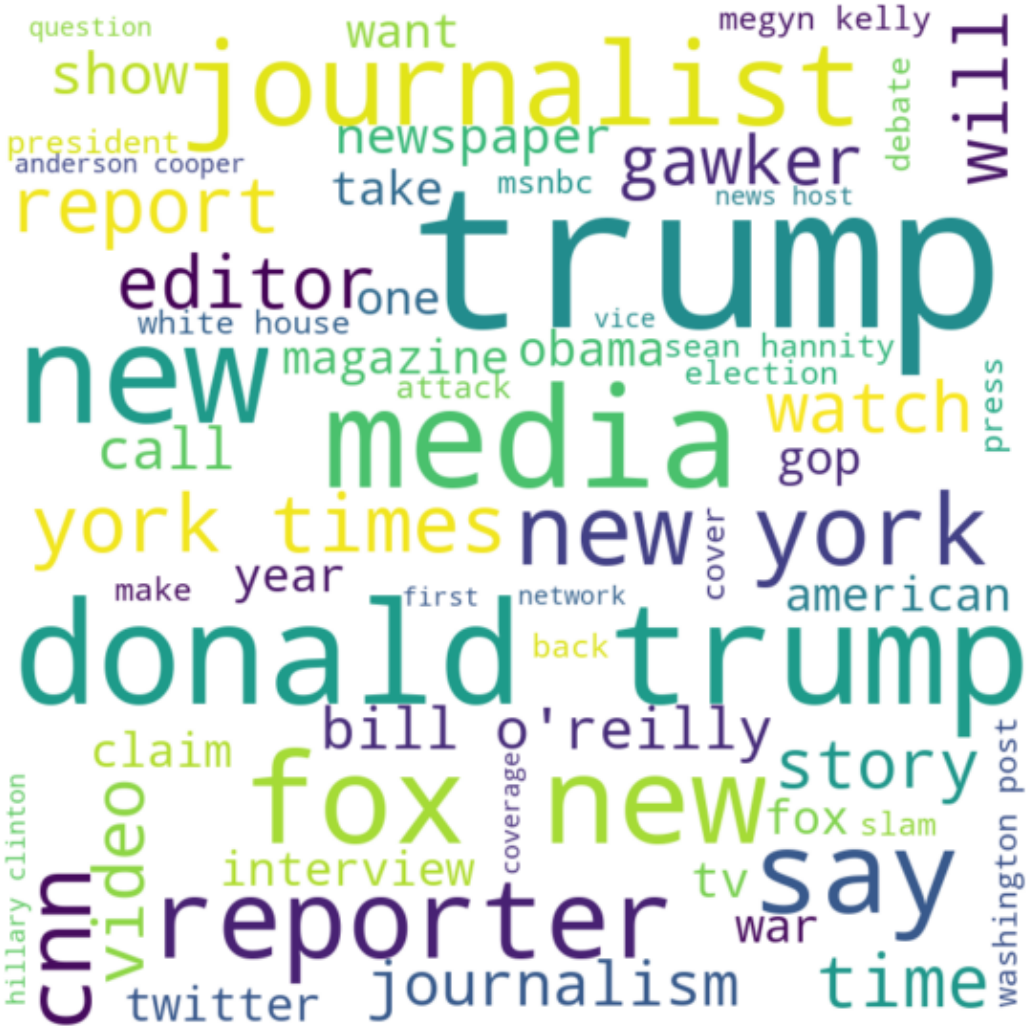
IMPACT


refugee
crisis
take
want
ebola
global
women
need
us
good
make
veteran
change
cancer
world
thing
war
life
donate
america
job
year
hope
giving
work
health
education
million
children
one
day
new
fight
back
end
leader
time
people
kid
live
video
home
food
homelessness
help
service
girl
american
poverty
charity
will
child
give
aid
free
helping
homeless
school
way
future

LATINO VOICES

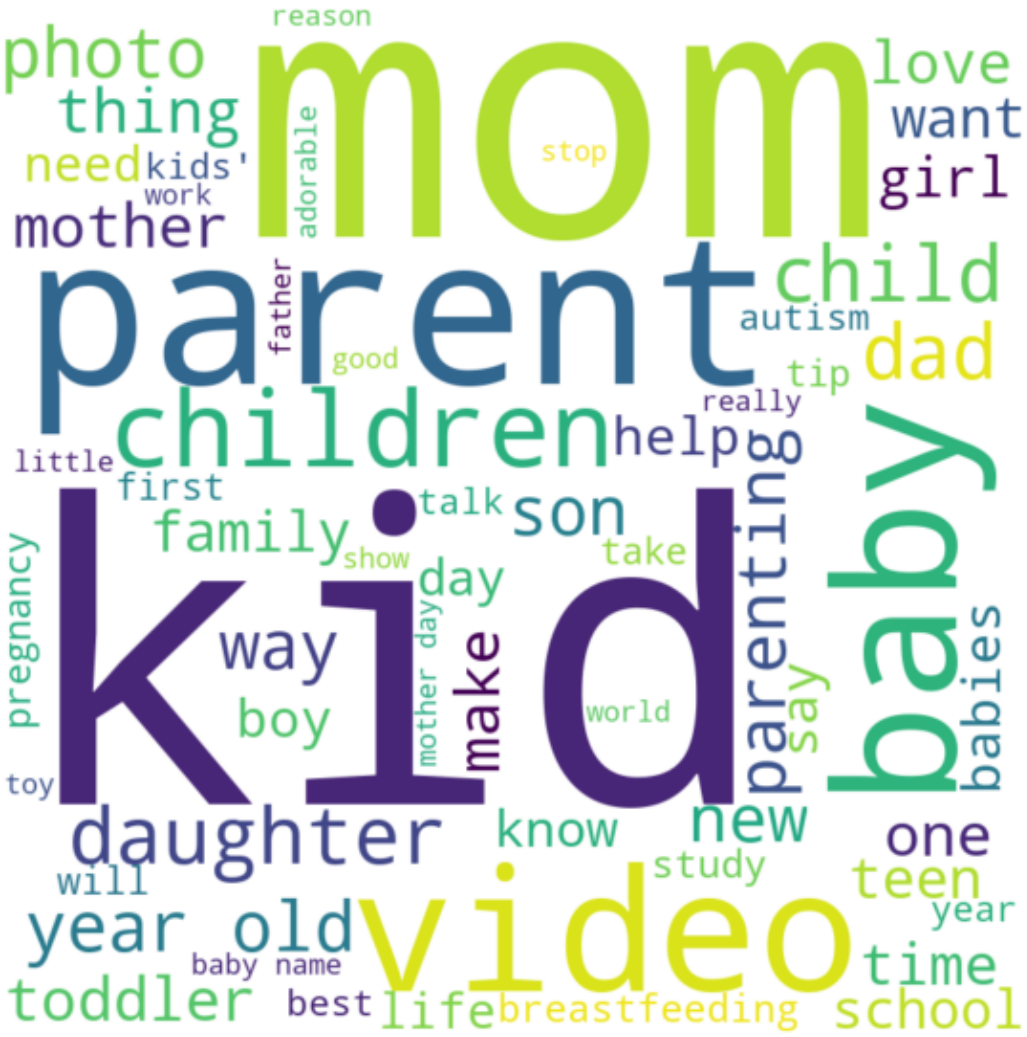


MEDIA

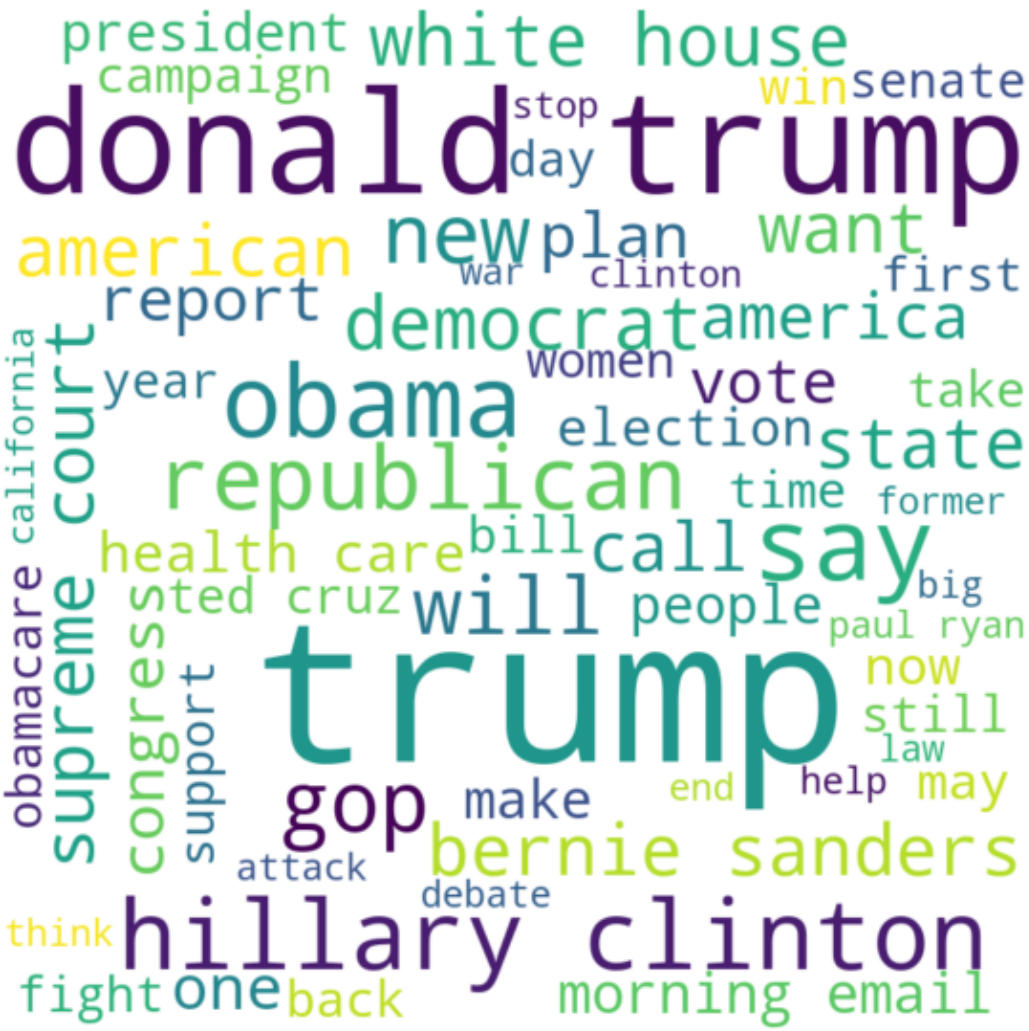

question
want
megyn kelly
show
journalist
will
president
newspaper
debate
anderson cooper
gawker
take
msnbc
news host
report
trump
editor
one
white house
vice
new
magazine
obama
sean hannity
election
attack
watch
media
press
call
gop
york times
new york
cover
make
year
first
network
american
donald trump
back
bill o'reilly
claim
story
fox new
fox
slam
hillary clinton
video
coverage
interview
tv
say
cnn
reporter
war
twitter
journalism
time
washington post


MONEY
financial
money
tax
million
bank
pay
way
credit card
consumer
time
best
job
car
will
taxes
free
tip
year
find
study
percent
make
report
avoid
day
new
plan
american
bad
people
retirement
price
college
mortgage
video
saving
student loan
bill
man
photo
sale
student
spending
fee
home
know
cash
loan
credit score
thing
cost
credit
buy
debt
customer
black friday
irs
five
may
save

PARENTS


reason
photo
mom
love
thing
want
need
kids'
stop
girl
work
adorable
mother
parent
child
father
autism
dad
good
tip
really
children
help
baby
little
first
talk
son
kid
family
show
take
pregnancy
day
parenting
way
mother day
make
babies
boy
world
say
toy
daughter
know
new
one
study
will
teen
year old
video
year
baby name
time
toddler
best
life
breastfeeding
school

POLITICS


president
white house
campaign
win
senate
donald trump
stop
day
american
new
plan
want
war
clinton
first
report
democrat
america
women
vote
take
year
obama
election
state
republican
time
former
bill
call
say
health care
big
ted cruz
will
people
paul ryan
trump
now
still
law
gop
make
end
help
may
bernie sanders
attack
debate
hillary clinton
think
fight
one
back
morning
email
supreme court
congress
support
obamacare
california


QUEER VOICES
queer
new
gay
transgender
lgbt
tran
love
say
star
trump
lgbtq
coming
video
year
life
show
people
gay men
gay marriage
kid
mom
anti gay
sex
lesbian
rupaul drag
now
photo
right
watch
couple
time
marriage equality
men
make
gender
come
first
one
community
america
meet
will
want
open
pride
sex marriage
family
man
take
teen
talk
two
look
hiv
women
black
back
son
activist
drag race'

RELIGION


sikh
say
jesus
american
make
man
gay
one
clergy
us
day
world
muslim
jewish
prayer
women
time
priest
take
pope
francis
jew
peace
america
bible
people
love
end
islamophobia
daily meditation
vatican
catholic
right
faith
religious
message
pope
god
church
need
now
first
new
mormon
death
religion
leader
will
bishop
call
year
look
show
christmas
christian
life
trump
pastor
evangelical
nun
spiritual


SCIENCE
nasa
universe
discovered
animal
see
people
image
mars
life
earth
astronaut
study suggests
astronomer
pluto
face
may
new
science
scientists say
new study
study shows
say
found
fossil
time
year
space
make
suggest
secret
brain
first
take
photo
day
women
ancient
watch
show
star
look
world
mission
reveal
spacecraft
video
way
climate change
dinosaur
help
sun
will
light
asteroid
moon
scientist
planet
one
big
human

SPORTS


watch
nfl
super bowl
player
say
open
year
day
mlb
nba
coach
thing
report
star
beat
first
nfl player
one
face
final
season
team
photo
football
win
soccer
fight
best
vs
will
play
hit
athlete
make
live
video
want
even
back
espn
goal
now
lebron james
game
fan
rio
time
new
look
olympic
take
baseball
need
world
sport
women
call
world cup
show
world series

STYLE & BEAUTY

TECH


fight
user
robot
data
iphone
world
smartphone
amazon
watch
use
online
microsoft
apple
company
instagram
want
digital
hacker
new
tech
show
app
best
youtube video
week
help
now
one
watch top
thing
still
photo
facebook
time
day
future
billion
internet
twitter
apple rumors
report
way
technology
apple
say
take
week
make
phone
may
year
top youtube
deal
uber
videos
will
google
million
samsung
privacy
people
fbi

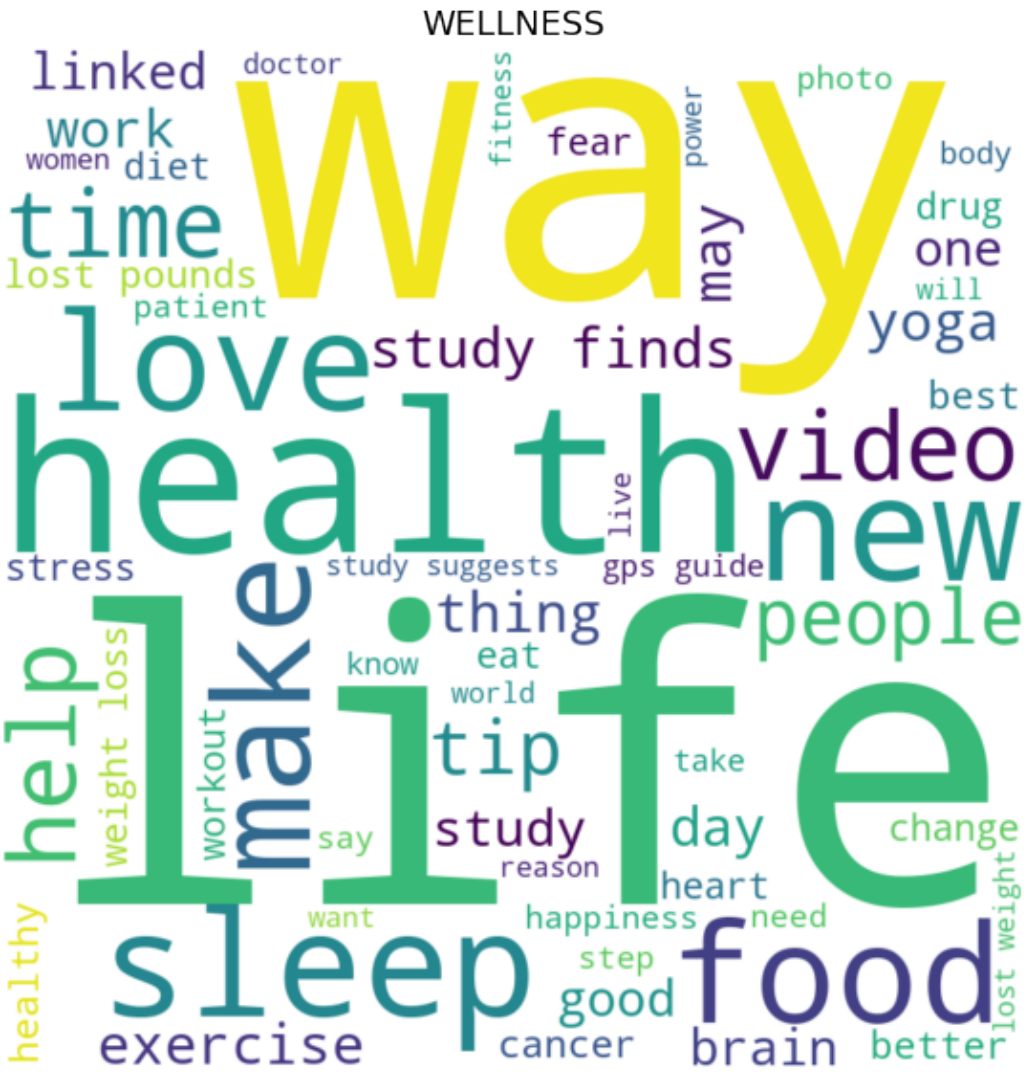
TRAVEL


travel
around world
flight
visit
beautiful
cities
plane
thing
destination
love
looney
front
hotel
london
reason
world
great
road trip
tsa
tourist
beach
city
new
day
go
island
tip
adventure
paris
see
best
airline
top
airport
place
way
time
stay
getaway
will
photo
traveler
make
guide
life
trip
summer
winter
town
tour
one
europe
traveling
new york
video
vacation
america
take
food
passenger
american

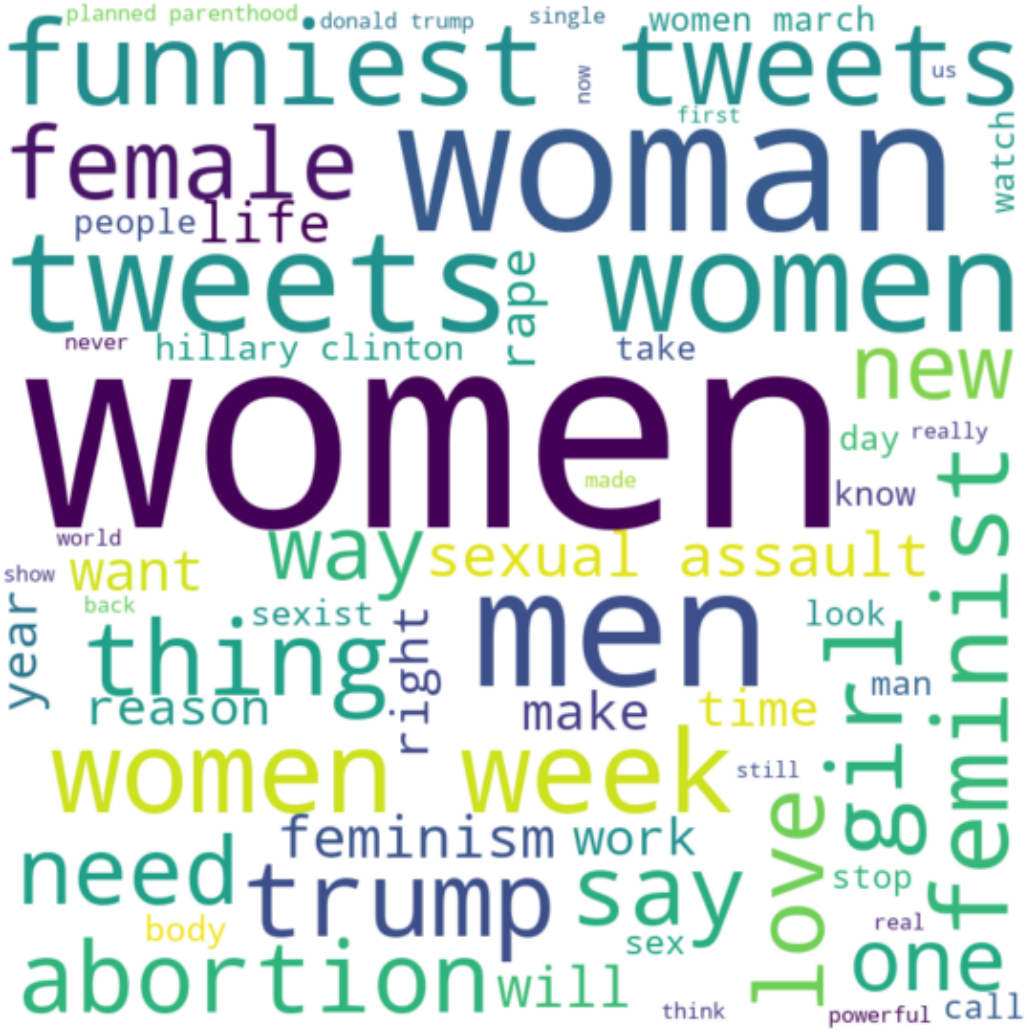
WEDDINGS

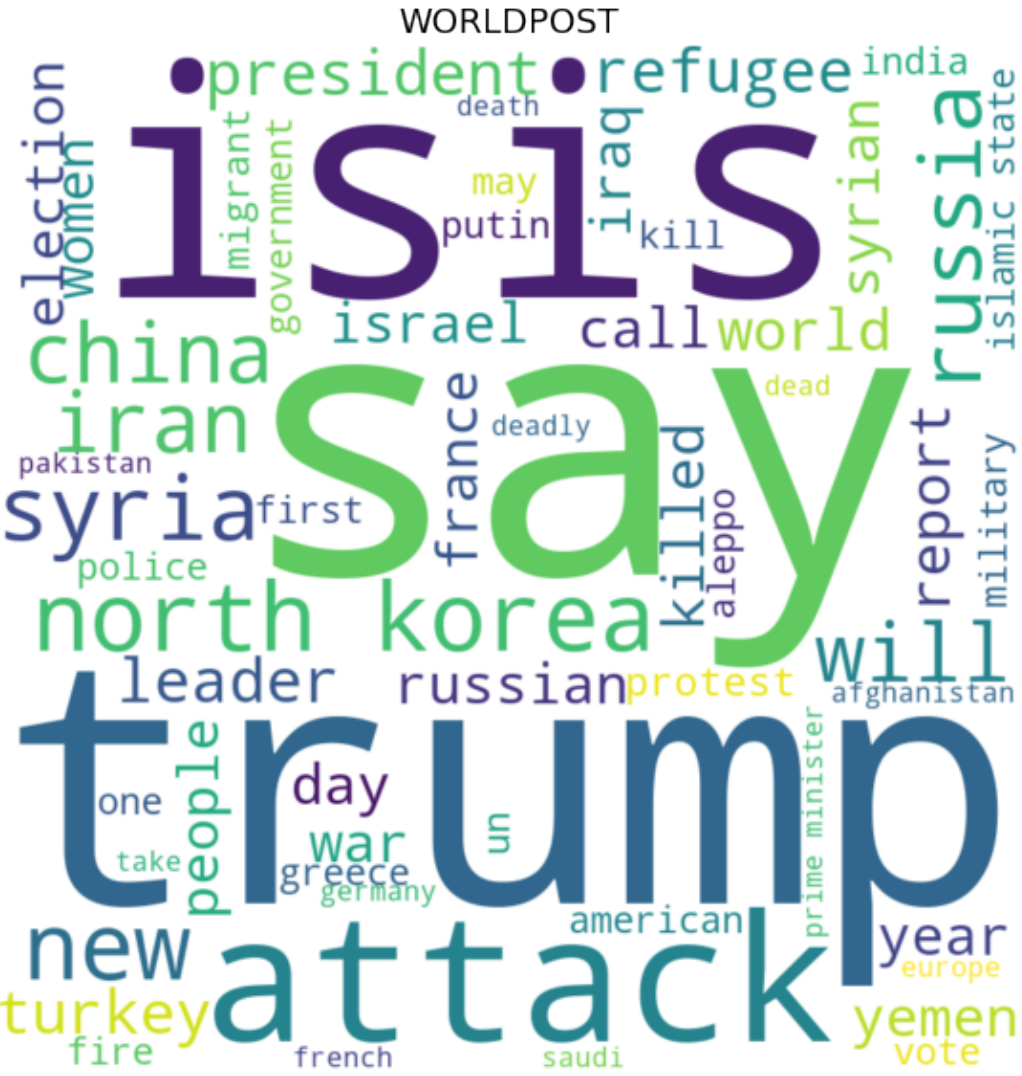

marry
bridesmaid
marriage
reason
secret
make
got married
wedding day
ring
talk
husband
couple
year
top
best
day
married weekend
want
wife
wedding
thingreal wedding
wedding guest
wedding dresses
wedding dress
idea
weddings couples
reveal
video
proposal
celebrity
women
wedding planning
star
new marriage proposal
relationship
big day
honeymoon
couples got
plan
one
photo
first
love
show
spouse
engagement
engaged
groom
will
trend
bridal
way
sex
say
bride
tip
engagement ring
getting married


WEIRD NEWS
shark
life
cop
video
cat
say
look
see
new
find
will
give
goes
internet
take fark
trump
man
year old
christmas
florida
guy
day
news quiz
photo
baby
now
accused
fire
weird new
woman
dog
real
home
truck
know
gets
police
show
make
want
human
fark weird
watch
year
turn
thing
sex
bear
allegedly
one
nsfw
car
take
found
people
time
donald trump
head
world
way


WELLNESS
linked
doctor
fitness
photo
work
women
diet
fear
power
body
way
time
drug
one
lost pounds
patient
may
will
yoga
love
study finds
best
health
video
new
stress
study suggests
gps guide
live
thing
people
life
help
weight loss
make
know
eat
world
tip
workout
take
study
day
change
say
reason
heart
want
happiness
need
healthy
sleep
food
step
good
lost weight
exercise
cancer
brain
better

WOMEN


planned parenthood
donald trump
single
women march
funniest
tweets
us
now
first
female
woman
watch
people
life
tweets
women
rape
never
hillary clinton
take
new
women
day
really
made
know
world
way
sexual assault
show
want
back
sexist
look
year
thing
men
right
girl
feminist
reason
make
time
man
women
week
still
love
feminism
work
need
trump
say
stop
body
sex
real
abortion
will
one
think
powerful
call


WORLDPOST
isis
say
trump
president
refugee
india
election
women
migrant
government
death
may
putin
iraq
kill
syrian
russia
islamic state
china
israel
call
world
dead
iran
deadly
pakistan
france
killed
report
military
syria
first
aleppo
police
north korea
will
leader
russian
protest
afghanistan
people
day
one
war
un
prime minister
take
greece
germany
american
year
new
attack
europe
turkey
yemen
fire
french
saudi
vote