PROY-NB04-HASHTAGS

December 9, 2018

1 Proyecto final. Datamining.

1.0.1 Análisis tweets UEFA Champions League Final 2018

1.0.2 Participantes:

Gonzalo de las Heras de Matías - Jorge de la Fuente Tagarro - Alejandro Amarillas Cámara - Sergio Sampio Balmaseda.

1.0.3 Notebook (4/4). Análisis de Hashtags.

1.0.4 Objetivo del notebook:

Este notebook analiza, entre otras cosas, patrones de asociación entre los hashtags usados durante la final.

Librerias

```
In [23]: import pandas as pd
         import numpy as np
         import matplotlib
         import matplotlib.pyplot as plt
         from collections import Counter
         from Library.Apriori import APriori
         from wordcloud import WordCloud, STOPWORDS
```

Funciones

```
In [29]: def GetNumHashtags(Palabra):
             hashtagsEncontrados = []
             for index, row in Hashtags.iterrows():
                 for hashtag in row["hashtag"]:
                     if Palabra in hashtag.lower():
                         hashtagsEncontrados.append(hashtag)
             DfhashtagsEncontrados = pd.DataFrame(hashtagsEncontrados, columns=["hashtag"])
             DfhashtagsEncontrados = DfhashtagsEncontrados["hashtag"].value_counts().reset_inde
             DfhashtagsEncontrados = DfhashtagsEncontrados[DfhashtagsEncontrados["hashtag"] > 
             return DfhashtagsEncontrados
```

title

Carga de datos

```
In [18]: Datos = pd.read_json("datos_limpios.json")
         len(Datos)
```

```
Out[18]: 330384
```

Desgranamos la hora de cada tweet para poder agrupar más fácilmente.

```
In [19]: Datos["hora"] = pd.to_datetime(Datos["hora"])
         Datos["min"] = Datos["hora"].dt.minute
         Datos["hour"] = Datos["hora"].dt.hour
         del Datos["hora"]
```

Extraemos la columna de hashtags.

```
In [20]: Hashtags = pd.DataFrame()
         Hashtags["hashtag"] = Datos["hashtag"]
         Hashtags = Hashtags[Hashtags["hashtag"] != "-1"]
         Hashtags = Hashtags[Hashtags["hashtag"] != -1]
         Hashtags = Hashtags.reset_index(drop=True)
         len(Hashtags)
```

```
Out[20]:                    hashtag
         0                    [lfc]
         1               [sportone]
         2  [halamadrid, aporla13]
         3              [kuzabalit]
         4  [aporla13, halamadrid]
```

2 1.- Mapa de hashtags más usados

Creamos un mapa de palabras con aquellas más usadas.

```
In [32]: palabras = []
         for line in Hashtags["hashtag"] :
             palabras.extend(line)

         # Contamos las palabras.
         frecuenciaPalabras = Counter(palabras)

         # Creamos el mapa de palabras.
         wordcloud = WordCloud(width=1600,
                               height=800,
                               max_words=200,
                               background_color="white",
                               max_font_size=150,
                               relative_scaling=0.5,
                               colormap=matplotlib.cm.Blues_r,
                               normalize_plurals=True).generate_from_frequencies(frecuenciaPala

         # Mostramos el mapa de palabras.
         plt.figure(figsize=(20,10))
         plt.imshow(wordcloud, interpolation="bilinear")
         plt.axis("off")
         plt.tight_layout(pad=0)
         plt.show()
```

2.1 2.2 Hashtags principales

Analizamos los hashtags más significativos y curiosos que aparecen en el mapa de hashtags.

2.1.1 2.2.1 Real Madrid

```
In [40]: madrid = GetNumHashtags("madrid").head()
         madrid
```

Out[40]:

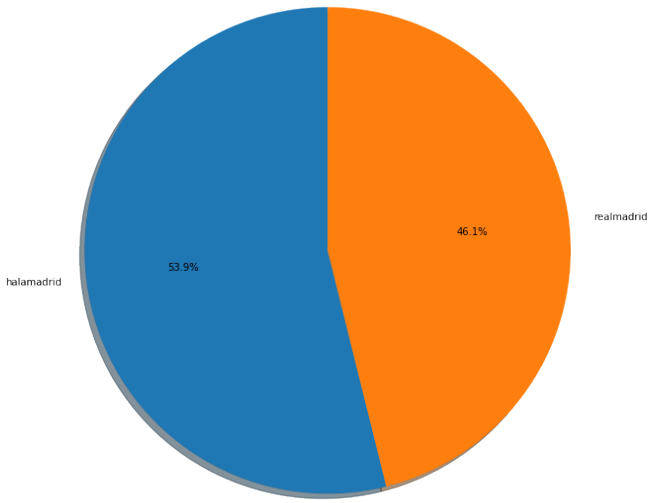
	index	hashtag
0	halamadrid	8001
1	realmadrid	6849
2	realmadridliverpool	1437
3	liverpoolvsrealmadrid	1030
4	madrid	570

```
In [41]: GetNumHashtags("aporla1").head()
```

Out[41]:

	index	hashtag
0	aporla13	2417
1	aporla14	162
2	aporla12	17

```
In [42]: fig, ax = plt.subplots(figsize=(20, 10))
         ax.pie(madrid.head(2)["hashtag"], labels=madrid.head(2)["index"], autopct='%1.1f%%', s
         ax.axis('equal')
         plt.show()
```



Resulta interesante como los aficionados del Real Madrid se tienen más unidos al lema 'Hala Madrid' que al nombre de su equipo.

2.1.2 2.2.2 Liverpool

```
In [43]: GetNumHashtags("liverpool").head()
```

```
Out[43]:                   index  hashtag
         0            liverpool     6487
         1       weareliverpool     2233
         2        realliverpool     1643
         3  realmadridliverpool     1437
         4          liverpoolfc     1102
```

```
In [44]: GetNumHashtags("walk").head()
```

```
Out[44]:                    index  hashtag
         0    youllneverwalkalone       45
         1      youneverwalkalone       19
         2  youwillneverwalkalone       17
```

```
In [45]: GetNumHashtags("ynwa").head()
```

```
Out[45]:   index  hashtag
         0  ynwa     2114
```

En cambio, los aficionados del liverpool se centran más en el nombre de su equipo.

2.2 2.3 Hashtags Ramos/Salah

```
In [46]: GetNumHashtags("ramos")
```

```
Out[46]:                    index  hashtag
         0                  ramos     1028
         1            sergioramos      445
         2              fuckramos      350
         3  shameonyousergioramos       50
         4       ramos_the_animal       44
         5          ramoscriminal       27
         6             dirtyramos       16
         7           fuckyouramos       13
```

```
In [47]: GetNumHashtags("salah").head()
```

```
Out[47]:            index  hashtag
         0          salah     3906
         1        mosalah      798
         2   mohamedsalah       89
         3  muhammedsalah       42
         4      momosalah       36
```

```
In [48]: GetNumHashtags("fuck")
```

```
Out[48]:            index  hashtag
         0     fuckramos      350
         1  fuckyouramos       13

In [49]: GetNumHashtags("wwe")

Out[49]:   index  hashtag
         0   wwe       57
```

Se remarca mucho la ausencia de Salah en la final y la culpa de Sergio Ramos.

2.3 2.4 Karius

```
In [50]: GetNumHashtags("karius")

Out[50]:          index  hashtag
         0       karius     1394
         1  loriskarius       16
```

Karius tuvo su propio hashtag debido a su actuación.

2.4 2.5 Hashtag extraños

2.4.1 2.5.1 Turquía

```
In [51]: GetNumHashtags("matchfixingcaseturkey")

Out[51]:                    index  hashtag
         0  matchfixingcaseturkey       59

In [52]: GetNumHashtags("justice")

Out[52]:                 index  hashtag
         0  allwewantisjustice       59
```

En principio parecían hashtags con tinte político, pero no es así. Se trata de una campaña en redes sociales, que todavía dura, en el que se reclama el campeonato de liga turca para el Trabzonspor debido a que se demostró una trama de amaños de partidos a favor del Fenerbahçe.

2.4.2 2.5.2 Política

```
In [53]: GetNumHashtags("political")

Out[53]:                            index  hashtag
         0  freecatalanpoliticalprisoners       24
```

Encontramos algunos tweets acerca de los políticos presos catalanes.

2.4.3 2.5.3 Caso de Tommy Robinson

```
In [37]: GetNumHashtags("tommy")
```

```
Out[37]:        index  hashtag
         0  freetommy       12
```

Algunos tweetos reclaman la libertad de Tommy Robinson, periodista que denunció a grupos musulmanes que violaban mujeres en baños públicos. Este periodista entró en prisión al enfrentarse a estos.

2.4.4 2.5.4 Gaza

```
In [39]: GetNumHashtags("gaza")
```

```
Out[39]:   index  hashtag
         0  gaza      342
```

Observamos algunos tweets acerca del conflico en la Franja de Gaza.

3 2.- Asociación de hashtags

A continuación, estudiamos las posibles reglas de asociación entre hashtags.

```
In [27]: apriori = APriori()
         apriori.Carga(Datos=Hashtags, Columna="hashtag")
         apriori.CalcularReglasDeConfianza(MinimoFreqSop=1000, Confianza=50.0, Echo=True)
```

```
Iniciando algoritmo apriori para patrones de asociación...

-------

Probando k = 1

Calculando items...

Calculando Frec.Soporte...

Calculando Soporte...
```

	Item	Frec. Soporte	Soporte
279	[aporla13]	2413	0.031504
394	[bale]	2072	0.027052
874	[championsleague]	1148	0.014988
878	[championsleaguefinal]	6214	0.081129
2218	[halamadrid]	7913	0.103311
2466	[innovateyourgame]	2547	0.033253

```
2652                 [karius]           1393  0.018187
2921                   [lfc]           5821  0.075998
2998             [liverpool]           6439  0.084067
3006           [liverpoolfc]           1099  0.014348
3021  [liverpoolvsrealmadrid]           1030  0.013448
4114                 [ramos]           1016  0.013265
4212          [realliverpool]           1642  0.021438
4215             [realmadrid]           6803  0.088819
4227    [realmadridliverpool]           1436  0.018748
4359                 [rmalfc]           2859  0.037327
4360                 [rmaliv]           7945  0.103729
4488                  [salah]           3901  0.050931
5304                    [ucl]           3045  0.039755
5318           [uclfinal2018]           4106  0.053607
5601         [weareliverpool]           2215  0.028919
5760                   [ynwa]           2109  0.027535
```

```
Probando k = 2
Calculando combinaciones
Calculando Frec.Soporte
Calculando Soporte
Filtrando Frec. Soporte mínimo >= 1000
```

```
                     Item  Frec. Soporte   Soporte
0   [aporla13, halamadrid]           1518  0.019819
1  [liverpool, realmadrid]           2560  0.033423
```

```
Probando k = 3
Calculando combinaciones
Calculando Frec.Soporte
Calculando Soporte
Filtrando Frec. Soporte mínimo >= 1000
Terminado
```

Reglas de asociación:

```
          r_1           r_2  soporte_r_1  soporte_r_2  confianza
```

```
0  [aporla13]  [halamadrid]         1518         2413       63.0
1  [aporla13]  [halamadrid]         1518         2413       63.0
```

#halamadrid y #aporla13 se dan juntas en un 63%. Son hashtags referentes del Real Madrid. Tiene sentido al ser 2 hashtag muy característicos del Real Madrid, pero llama la atención que el Liverpool no tenga una pareja de hashtags, como el #liverpool y su lema #ynwa.

```
In [30]: apriori = APriori()
         apriori.Carga(Datos=Hashtags, Columna="hashtag")
         apriori.CalcularReglasDeConfianza(MinimoFreqSop=500, Confianza=50.0, Echo=True)
```

```
Iniciando algoritmo apriori para patrones de asociación...

-------

Probando k = 1

Calculando items...

Calculando Frec.Soporte...

Calculando Soporte...
```

	Item	Frec. Soporte	Soporte
279	[aporla13]	2413	0.031504
394	[bale]	2072	0.027052
511	[beinucl]	971	0.012677
522	[believers]	572	0.007468
874	[championsleague]	1148	0.014988
878	[championsleaguefinal]	6214	0.081129
879	[championsleaguefinal2018]	825	0.010771
899	[championsxfox]	960	0.012534
1110	[cristiano]	532	0.006946
1114	[croisonsles]	889	0.011607
1693	[finalchampionstotal]	533	0.006959
2218	[halamadrid]	7913	0.103311
2466	[innovateyourgame]	2547	0.033253
2652	[karius]	1393	0.018187
2921	[lfc]	5821	0.075998
2998	[liverpool]	6439	0.084067
3006	[liverpoolfc]	1099	0.014348
3021	[liverpoolvsrealmadrid]	1030	0.013448
3041	[livrma]	544	0.007102
3154	[madrid]	563	0.007350
3444	[mosalah]	797	0.010406
4114	[ramos]	1016	0.013265

```
4212             [realliverpool]           1642  0.021438
4215                [realmadrid]           6803  0.088819
4227       [realmadridliverpool]           1436  0.018748
4359                    [rmalfc]           2859  0.037327
4360                    [rmaliv]           7945  0.103729
4369                      [rmcf]            816  0.010654
4488                     [salah]           3901  0.050931
5304                       [ucl]           3045  0.039755
5318              [uclfinal2018]           4106  0.053607
5601            [weareliverpool]           2215  0.028919
5760                      [ynwa]           2109  0.027535
```

```
Probando k = 2
Calculando combinaciones
Calculando Frec.Soporte
Calculando Soporte
Filtrando Frec. Soporte mínimo >= 500
```

```
                              Item  Frec. Soporte   Soporte
0             [aporla13, halamadrid]           1518  0.019819
1  [championsleaguefinal, liverpool]            520  0.006789
2     [championsleaguefinal, rmaliv]            514  0.006711
3              [croisonsles, rmaliv]            889  0.011607
4           [halamadrid, realmadrid]            648  0.008460
5                 [lfc, realmadrid]            519  0.006776
6                     [lfc, rmaliv]            536  0.006998
7                       [lfc, rmcf]            739  0.009648
8                       [lfc, ynwa]            633  0.008264
9            [liverpool, realmadrid]           2560  0.033423
```

```
Probando k = 3
Calculando combinaciones
Calculando Frec.Soporte
Calculando Soporte
Filtrando Frec. Soporte mínimo >= 500
Terminado
```

Reglas de asociación:

	r_1	r_2	soporte_r_1	soporte_r_2	confianza
0	[croisonsles]	[rmaliv]	889	889	100.0
1	[croisonsles]	[rmaliv]	889	889	100.0
2	[rmcf]	[lfc]	739	816	91.0
3	[rmcf]	[lfc]	739	816	91.0
4	[aporla13]	[halamadrid]	1518	2413	63.0
5	[aporla13]	[halamadrid]	1518	2413	63.0

Si reducimos la frecuencia soporte, obtenemos que #croisonsles y #rmaliv aparecen juntos siempre. Estos hashtag pertenecen a una cuenta de humor francesa, en la cual suben imágenes de protagonistas de actualidad con sus caras cambiadas a modo de caricatura. #rmcf y #lfc aparencen juntos en un 91%. También tiene mucho sentido pues son las siglas de ambos equipos de la final.

4 Referencias

https://www.kaggle.com/xvivancos/tweets-during-r-madrid-vs-liverpool-ucl-2018
https://github.com/pbugnion/gmaps
https://pandas.pydata.org/
Apuntes de la asignatura