PROY-NB02-ESTADISTICAS-GENERALES

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 (2/4). Estadísticas Generales.

1.0.4 Objetivo del notebook:

Este notebook se centra en mostrar estadísticas generales acerca de los tweets.

Librerias

```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import gmaps
        import gmaps.datasets
        import gmaps.geojson_geometries
        from collections import Counter
        from Library.Apriori import APriori
        from wordcloud import WordCloud, STOPWORDS
        from matplotlib.cm import viridis
        from matplotlib.colors import to_hex

        pd.options.display.max_colwidth = 500
        gmaps.configure(api_key="AIzaSyChafUDbO0RYZ7y7wsZKhgslvQXSDudURQ")
```

Funciones

```
In [2]: def RGB(Minimo, Maximo, Valor):
            """
            Función que calcula el valor RGB para el mapa de calor.
            """
            Minimo, Maximo = float(Minimo), float(Maximo)
```

title

```
    ratio = 2 * (Valor-Minimo) / (Maximo - Minimo)
    b = int(max(0, 255*(1 - ratio)))
    r = int(max(0, 255*(ratio - 1)))
    g = 255 - b - r
    return r, g, b
```

Carga de datos

```
In [3]: Datos = pd.read_json("datos_limpios.json")
        print("Hay", len(Datos), "registros")
```

```
Hay 330384 registros
```

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

```
In [4]: Datos["hora"] = pd.to_datetime(Datos["hora"])
        Datos["min"] = Datos["hora"].dt.minute
        Datos["hour"] = Datos["hora"].dt.hour
        del Datos["hora"]
```

Eliminamos 61 registros que son antes del partido a una hora distanciada de la hora del partido.

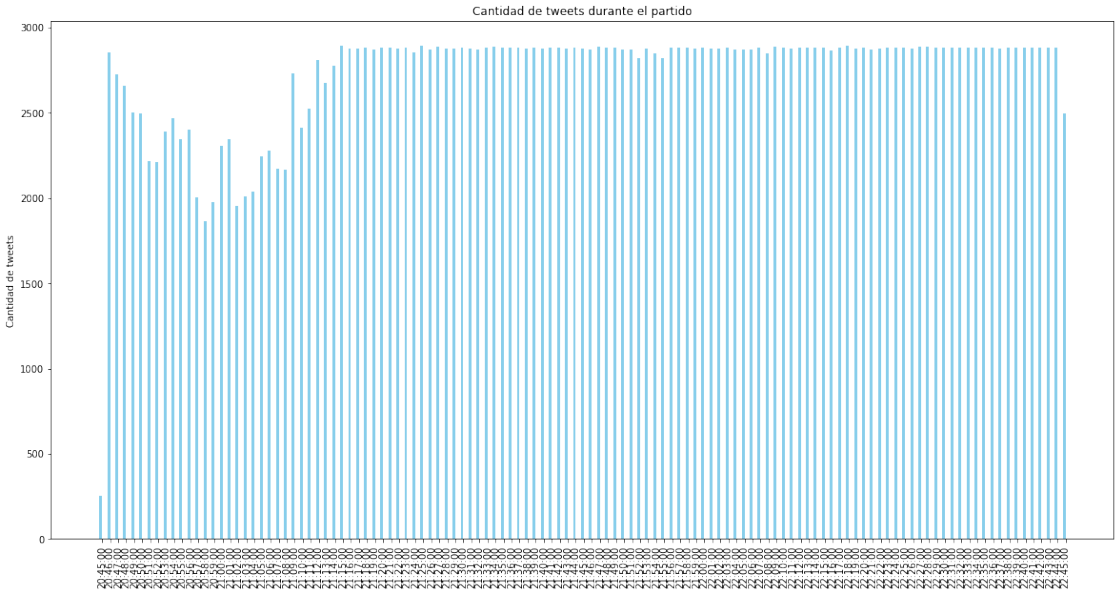
```
In [5]: Datos = Datos[Datos["hour"] > 15]
        print("Hay", len(Datos), "registros")
```

```
Hay 330323 registros
```

2 1.- Cantidad de tweets

En total hay más de 330.000 tweets.

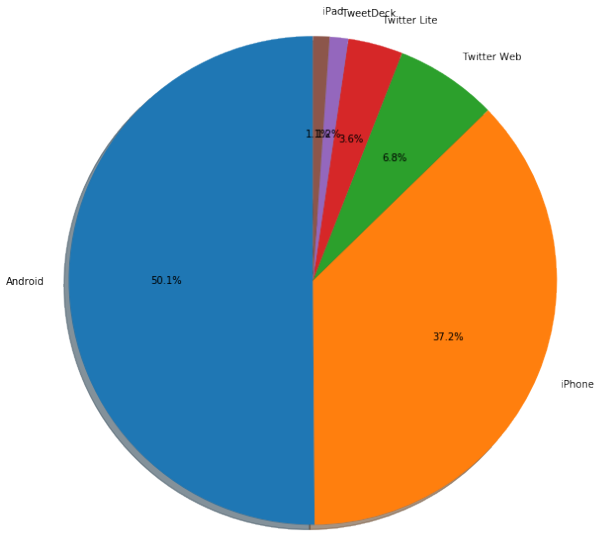
In [6]:
```
# Extraemos los campos necesarios y agrupamos por hora y minuto.
horas = Datos[['hour', 'min']]
horas = horas.groupby(['hour', 'min']).size().reset_index(name='counts')
horas["hora_str"] = ""
horas["hora"] = ""

# Creamos las etiquetas para la gráfica.
for index, row in horas.iterrows():
    if row["min"] < 10:
        minutos = "0" + str(row["min"])
    else:
        minutos = str(row["min"])
    horas.loc[index, "hora_str"] = str(row["hour"]) + ":" +minutos
    horas.loc[index, "hora"] = str(row["hour"]) + ":" + minutos + ":00"

# Preparamos la gráfica.
ind = np.arange(len(horas))
ancho = 0.35
fig, ax = plt.subplots(figsize=(20, 10))
rects1 = ax.bar(ind - ancho/2, horas["counts"], ancho, color='SkyBlue')
ax.set_ylabel('Cantidad de tweets')
ax.set_title('Cantidad de tweets durante el partido')
ax.set_xticks(ind)
ax.set_xticklabels(horas["hora"])
plt.xticks(rotation=90)

# Mostramos la gráfica.
plt.show()
```



Vemos que durante el comienzo del partido, que no ocurrió nada reseñable, los tweets acerca de la final son bajos. Sin embargo, desde el incidente con Sergio Ramos, la cantidad de tweets se dispara (hasta alcanzar el máximo número de tweets que podemos obtener gratuitamente de la API de Twitter).

3 2.- Fuente de los tweets

```
In [7]: Fuente = Datos["fuente"]
        Fuente = pd.DataFrame(Fuente.value_counts()).reset_index()
        Fuente = Fuente[Fuente["fuente"] > 3000]
        Fuente.head()
        fig, ax = plt.subplots(figsize=(20, 10))
        ax.pie(Fuente["fuente"], labels=Fuente["index"], autopct='%1.1f%%', shadow=True, starta
        ax.axis('equal')
        plt.show()
```



Observamos una amplia mayoría de tweets enviados desde dispositivos móviles. Coincide que Android está más extendido que iPhone. Llama la atención que solo un 2% de los tweets hayan sido enviados desde una herramienta para gestionar redes sociales como es TweetDeck.

4 3.- Países/Idiomas

Cargamos el dataset de paises y su correspondencia en idioma.

```
In [8]: DatosPaises = pd.read_csv("paises.csv", sep=";")
```

Agrupamos para calcular el número de tweets de cada país.

```
In [9]: # Capturamos los campos requeridos.
        IdiomasPaises = Datos[['usuario_idioma']]
        IdiomasPaises = IdiomasPaises.groupby(['usuario_idioma']).size().reset_index(name='cou
        IdiomasPaises["usuario_idioma"] = IdiomasPaises["usuario_idioma"].str.lower()
        IdiomasPaises["pais"] = ""
        IdiomasPaises = IdiomasPaises.reset_index(drop=True)

        # Preprocesamos el nombre de cada país e idioma para poder cruzar datasets.
        for index, row in IdiomasPaises.iterrows():
            pais = DatosPaises[DatosPaises["idioma"] == row["usuario_idioma"]]["pais"]
            if len(pais) > 0:
                if row["usuario_idioma"] == "en-gb":
                    IdiomasPaises.loc[index, "usuario_idioma"] = "en"
                IdiomasPaises.loc[index, "pais"] = pais.values[0]

        # Eliminamos los registros de los que no sepamos qué país es.
        IdiomasPaises = IdiomasPaises[IdiomasPaises["pais"] != ""]
```

```
IdiomasPaises = IdiomasPaises.reset_index(drop=True)

# Borramos el campo innecesario.
del IdiomasPaises["usuario_idioma"]

# Agrupos y contamos.
IdiomasPaises = IdiomasPaises.groupby(['pais']).sum().reset_index()
IdiomasPaises
```

Out[9]:

	pais	counts
0	Catalonia	1062
1	Denmark	170
2	France	13185
3	Germany	7258
4	Greece	1811
5	Indonesia	4089
6	Italy	6835
7	Japan	4884
8	Netherlands	1248
9	Poland	1583
10	Portugal	19531
11	Russia	1175
12	Spain	49442
13	Thailand	3859
14	Turkey	5653
15	Ukraine	284
16	United Arab Emirates	6995
17	United Kingdom	199323

Resulta llamativo como hay gran cantidad de tweets de cataluña. En el siguiente apartado de análisis de sentimientos, veremos qué dicen.

4.1 3.1.- Mapa de calor

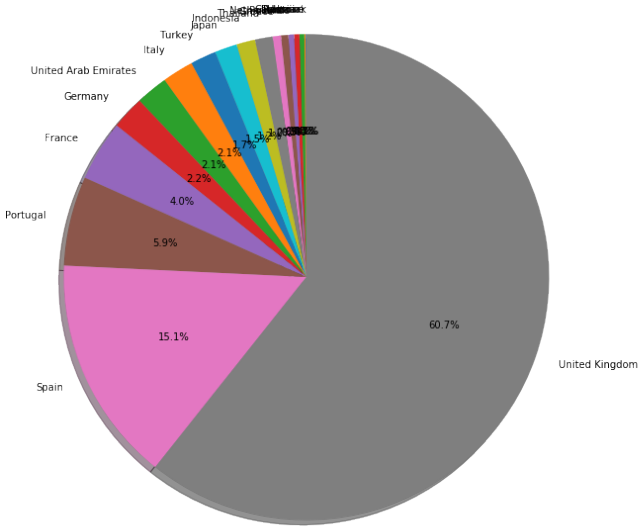
In [10]:
```
# Cargamos la geometría de cada país.
paises_geojson = gmaps.geojson_geometries.load_geometry('countries')

# Cálculo de valores para normalizar valores.
min_val = min(IdiomasPaises["counts"])
max_val = max(IdiomasPaises["counts"])
rango = max_val - min_val

# Calculamos cada color para el mapa de calor.
colores = []
for feature in paises_geojson['features']:
    nombre_pais = feature['properties']['name']
    try:
        if nombre_pais in IdiomasPaises["pais"].values:
```

```
                    # Extraemos del dataset el nž de tweets.
                    numTweets = IdiomasPaises[IdiomasPaises["pais"] == nombre_pais]["counts"]
                    # Calculamos el color.
                    r,g,b = RGB(min_val, max_val, numTweets)
                    # Lo parseamos el formato correcto.
                    color = (r, g , b, 1)
                else:
                    color = (0,0,0,0)
            except KeyError:
                print("Error")
                color = (0, 0, 0, 0)
            colores.append(color)

        # Creamos el mapa.
        fig = gmaps.figure()
        capa = gmaps.geojson_layer(
            paises_geojson,
            fill_color=colores,
            stroke_color=colores,
            fill_opacity=0.8)

        # Añadimos la capa y mostramos el mapa.
        fig.add_layer(capa)
        fig
```

```
Figure(layout=FigureLayout(height='420px'))
```

Observamos como los países que más tweets han escrito son aquellos de los participantes en la final. También sale Portugal, aunque en mucho menos medida. Esto se debe al seguimiento masivo que provoca en su país Cristiano Ronaldo.

```
In [11]: IdiomasPaises  = IdiomasPaises.sort_values(["counts"])
         fig, ax = plt.subplots(figsize=(20, 10))
         ax.pie(IdiomasPaises["counts"], labels=IdiomasPaises["pais"], autopct='%1.1f%%', shad
         ax.axis('equal')
         plt.show()
```



Llama la atención los Emiratos Árabes. Aunque es un país sin mucha tradición futbolística, cada vez más se va introduciendo en el fútbol. Prueba de ello, es como jugadores famosos juegan en sus últimos años en este país.

5 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