

Reconocimiento de emociones

En este archivo se presentarán diversas alternativas para el reconocimiento y análisis de emociones en Hilos de diferente largo.

Para comodidad del lector primero se analizará solo un tweet, luego de presentar el método de análisis se procederá a trabajar todo el dataset.

Importación de librerías y apertura de archivos

In [1]:

```
import os
import random
import pandas as pd
```

In [34]:

```
from bokeh.io import output_notebook, show
from bokeh.plotting import figure
from bokeh.models import Span

output_notebook()
```

(<http://Bokeh5.0.12.13.ng>) successfully loaded.

In [2]:

```
from emotion_predictor import EmotionPredictor
```

Using Theano backend.

In [7]:

```
csv1 = pd.read_csv('five_ten.csv', encoding='iso-8859-1')
csv1_grouped_by_thread = csv1.groupby(['thread_number'])
threads1 = {}
kthreads1 = {}

csv2 = pd.read_csv('ten_fifteen.csv', encoding='iso-8859-1')
csv2_grouped_by_thread = csv2.groupby(['thread_number'])
threads2 = {}
kthreads2 = {}

csv3 = pd.read_csv('fifteen_twenty.csv', encoding='iso-8859-1')
csv3_grouped_by_thread = csv3.groupby(['thread_number'])
threads3 = {}
kthreads3 = {}

csv4 = pd.read_csv('twenty_twentyfive.csv', encoding='iso-8859-1')
csv4_grouped_by_thread = csv4.groupby(['thread_number'])
threads4 = {}
kthreads4 = {}

csv5 = pd.read_csv('twentyfive_thirty.csv', encoding='iso-8859-1')
csv5_grouped_by_thread = csv5.groupby(['thread_number'])
threads5 = {}
kthreads5 = {}
```

Threads como diccionario y como arreglo de string

Se tienen dos formatos para trabajar los datos. El primero es como un diccionario de threads, el cual la clave es el numero de thread y el valor es un arreglo de strings con todos los tweets del thread. La otra es un arreglo de strings, los cuales son todos los tweets unidos en orden separados con un \n

In [217]:

```

string = '\n'

for thread, data in dict(list(csv1_grouped_by_thread)).items():
    threads1[thread] = list(data['text'])
    kthreads1[thread] = string.join(list(data['text']))

for thread, data in dict(list(csv2_grouped_by_thread)).items():
    threads2[thread] = list(data['text'])
    kthreads2[thread] = string.join(list(data['text']))

for thread, data in dict(list(csv3_grouped_by_thread)).items():
    threads3[thread] = list(data['text'])
    kthreads3[thread] = string.join(list(data['text']))

for thread, data in dict(list(csv4_grouped_by_thread)).items():
    threads4[thread] = list(data['text'])
    kthreads4[thread] = string.join(list(data['text']))

for thread, data in dict(list(csv5_grouped_by_thread)).items():
    threads5[thread] = list(data['text'])
    kthreads5[thread] = string.join(list(data['text']))

Tthreads1 = list(kthreads1.values())
Tthreads2 = list(kthreads2.values())
Tthreads3 = list(kthreads3.values())
Tthreads4 = list(kthreads4.values())
Tthreads5 = list(kthreads5.values())

```

In [15]:

threads1["Thread 1"]

Out[15]:

['Extraordinary evidence at Treasury committee from Jon Thompson, CEO of HMR C on customs and Brexit today <https://t.co/DJhIQhmVwJ>', (<https://t.co/DJhIQhmVwJ>),
 "The Brexiter favourite Max Fac - would cost business between £17 and £20 bn a year\r\n\r\n\r\n- that's almost 1% of GDP\r\n\r\n\r\n- just?!", <https://t.co/0MwIcwre4t>", (<https://t.co/0MwIcwre4t>),
 "How does he arrive at the figure\r\n\r\n\r\n200m export consignments at an average cost of £32.50 each = £6.5bn (times two be?!", <https://t.co/KxnkU2QiVO>", (<https://t.co/KxnkU2QiVO>),
 "Theresa May's New Customs Partnership is much cheaper for business (almost zero cost) because it seeks to replicatâ?", <https://t.co/0LcsJHah0H>", (<https://t.co/0LcsJHah0H>),
 "Mr Thompson said he did not expect the EU to reciprocate over the customs partnership. \r\n\r\n\r\nWhat that means is UK collâ?", <https://t.co/9c3uhhnZGX>", (<https://t.co/9c3uhhnZGX>),
 "Both would not be ready by 2021. Max Fac needs 3 years. Customs Partnership requires 5, Mr Thompson said.\r\n\r\n\r\nThe bordâ?", <https://t.co/luLzgUsiR4>", (<https://t.co/luLzgUsiR4>),
 '"We think we can manage the risk - we think we can" he said. He didn't sound so sure. \r\n\r\n\r\nAnd "the potential backdoorâ?", <https://t.co/Ti1nbbjfpU>"! (<https://t.co/Ti1nbbjfpU>)

In [223]:

kthreads1

Out[223]:

```
{'Thread 1': 'Extraordinary evidence at Treasury committee from Jon Thompson, CEO of HMRC on customs and Brexit today https://t.co/DJhIQhmVwJ\nThe (https://t.co/DJhIQhmVwJ\nThe) Brexiter favourite Max Fac - would cost business between £17 and £20bn a year\r\n\r\n\r\n- that's almost 1% of GDP\r\n\r\n\r\n- jusâ?| https://t.co/0MwIcwre4t\nHow (https://t.co/0MwIcwre4t\nHow) does he arrive at the figure\r\n\r\n\r\n200m export consignments at an average cost of £32.50 each = £6.5bn (times two beâ?| https://t.co/KxnkU2QiVO\nTheresa (https://t.co/KxnkU2QiVO\nTheresa) May's New Customs Partnership is much cheaper for business (almost zero cost) because it seeks to replicate?| https://t.co/0LcsJHah0H\nMr (https://t.co/0LcsJHah0H\nMr) Thompson said he did not expect the EU to reciprocate over the customs partnership.\r\n\r\nWhat that means is UK collâ?| https://t.co/9c3uhhnZGX\nBoth (https://t.co/9c3uhhnZGX\nBoth) would not be ready by 2021. Max Fac needs 3 years. Customs Partnership requires 5, Mr Thompson said.\r\n\r\nThe bordâ?| https://t.co/luLzgUsiR4\n"We (https://t.co/luLzgUsiR4\n"We) think we can manage the risk - we think we can" he said. He didn't sound so sure.\r\n\r\nAnd "the potential backdoorâ?| https://t.co/Ti1nbbifpU'. (https://t.co/Ti1nbbifpU.)
```

In [224]:

Tthreads1

Out[224]:

```
['Extraordinary evidence at Treasury committee from Jon Thompson, CEO of HMRC on customs and Brexit today https://t.co/DJhIQhmVwJ\nThe (https://t.co/DJhIQhmVwJ\nThe) Brexiter favourite Max Fac - would cost business between £17 and £20bn a year\r\n\r\n\r\n- that's almost 1% of GDP\r\n\r\n\r\n- jusâ?| https://t.co/0MwIcwre4t\nHow (https://t.co/0MwIcwre4t\nHow) does he arrive at the figure\r\n\r\n\r\n200m export consignments at an average cost of £32.50 each = £6.5bn (times two beâ?| https://t.co/KxnkU2QiVO\nTheresa (https://t.co/KxnkU2QiVO\nTheresa) May's New Customs Partnership is much cheaper for business (almost zero cost) because it seeks to replicate?| https://t.co/0LcsJHah0H\nMr (https://t.co/0LcsJHah0H\nMr) Thompson said he did not expect the EU to reciprocate over the customs partnership.\r\n\r\nWhat that means is UK collâ?| https://t.co/9c3uhhnZGX\nBoth (https://t.co/9c3uhhnZGX\nBoth) would not be ready by 2021. Max Fac needs 3 years. Customs Partnership requires 5, Mr Thompson said.\r\n\r\nThe bordâ?| https://t.co/luLzgUsiR4\n"We (https://t.co/luLzgUsiR4\n"We) think we can manage the risk - we think we can" he said. He didn't sound so sure.\r\n\r\nAnd "the potential backdoorâ?| https://t.co/Ti1nbbifpU', (https://t.co/Ti1nbbifpU.)]
```

In [260]:

```
df_unidos1 = pd.DataFrame.from_dict(kthreads1, orient='index', columns=['Tweet'])
df_unidos2 = pd.DataFrame.from_dict(kthreads2, orient='index', columns=['Tweet'])
df_unidos3 = pd.DataFrame.from_dict(kthreads3, orient='index', columns=['Tweet'])
df_unidos4 = pd.DataFrame.from_dict(kthreads4, orient='index', columns=['Tweet'])
df_unidos5 = pd.DataFrame.from_dict(kthreads5, orient='index', columns=['Tweet'])

df_unidos1.index.name = "Hilo"
df_unidos1.reset_index(inplace=True)

df_unidos2.index.name = "Hilo"
df_unidos2.reset_index(inplace=True)

df_unidos3.index.name = "Hilo"
df_unidos3.reset_index(inplace=True)

df_unidos4.index.name = "Hilo"
df_unidos4.reset_index(inplace=True)

df_unidos5.index.name = "Hilo"
df_unidos5.reset_index(inplace=True)

df_unidos1.head()
```

Out[260]:

	Hilo	Tweet
0	Thread 1	Extraordinary evidence at Treasury committee f...
1	Thread 10	Here are some real ideas to fix the #immigrati...
2	Thread 100	This is such a shocking story in its details.\...
3	Thread 101	1) Attribution to an #Anon. Not wholly my wor...
4	Thread 11	Part 1: March 9ish. An excited #Qanon breaks d...

Detección de emociones.

Se importa el modelo a utilizar para la detección de emociones. Se trabajará con las emociones definidas por Ekman.

Se extraerán dos emociones por hilo, esto para posibles futuros análisis.

Existen dos alternativas para el output las emociones. La primera es a través de la predicción directa de la emoción, la cual entrega una tabla con la emoción principal de cada documento estudiado. La segunda es a través de una matriz de probabilidades, en la cual las columnas son las emociones y las filas los documentos a analizar. Cada valor de la matriz es la probabilidad de que el documento sea asociado a esa emoción.

In [17]:

```
ekman = EmotionPredictor(classification='ekman', setting='mc')
```

In [18]:

```
tweets = threads1["Thread 1"]
matriz = ekman.predict_probabilities(tweets)
tabla = ekman.predict_classes(tweets)
```

In [19]:

matriz

Out[19]:

	Tweet	Anger	Disgust	Fear	Joy	Sadness	Surprise
0	Extraordinary evidence at Treasury committee f...	0.005198	0.002203	0.022643	0.673286	0.009532	0.287138
1	The Brexiter favourite Max Fac - would cost bu...	0.013784	0.007089	0.054312	0.482430	0.112748	0.329637
2	How does he arrive at the figure\r\n\r\n\r\n\r\n20...	0.074924	0.011434	0.400213	0.295159	0.069761	0.148510
3	Theresa May's New Customs Partnership is much ...	0.012332	0.036829	0.179175	0.090255	0.018078	0.663330
4	Mr Thompson said he did not expect the EU to r...	0.030329	0.017072	0.060771	0.088922	0.277415	0.525490
5	Both would not be ready by 2021. Max Fac needs...	0.015301	0.005518	0.113649	0.403367	0.097253	0.364913
6	"We think we can manage the risk - we think we...	0.025122	0.007756	0.403552	0.187179	0.057420	0.318970

In [33]:

```
matriz.describe()
```

Out[33]:

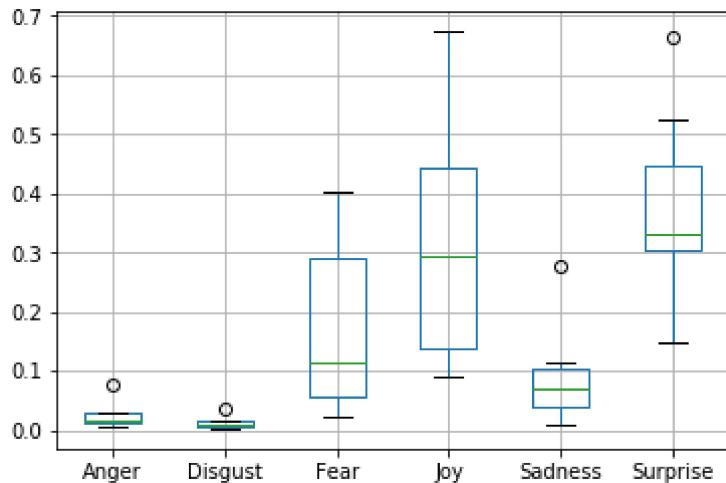
	Anger	Disgust	Fear	Joy	Sadness	Surprise
count	7.000000	7.000000	7.000000	7.000000	7.000000	7.000000
mean	0.025284	0.012557	0.176331	0.317228	0.091744	0.376856
std	0.023426	0.011694	0.162046	0.216998	0.090178	0.168362
min	0.005198	0.002203	0.022643	0.088922	0.009532	0.148510
25%	0.013058	0.006303	0.057542	0.138717	0.037749	0.303054
50%	0.015301	0.007756	0.113649	0.295159	0.069761	0.329637
75%	0.027726	0.014253	0.289694	0.442898	0.105001	0.445201
max	0.074924	0.036829	0.403552	0.673286	0.277415	0.663330

In [51]:

```
matriz.boxplot(column = ["Anger", "Disgust", "Fear", "Joy", "Sadness", "Surprise"])
```

Out[51]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x15e1a326e10>
```



In [37]:

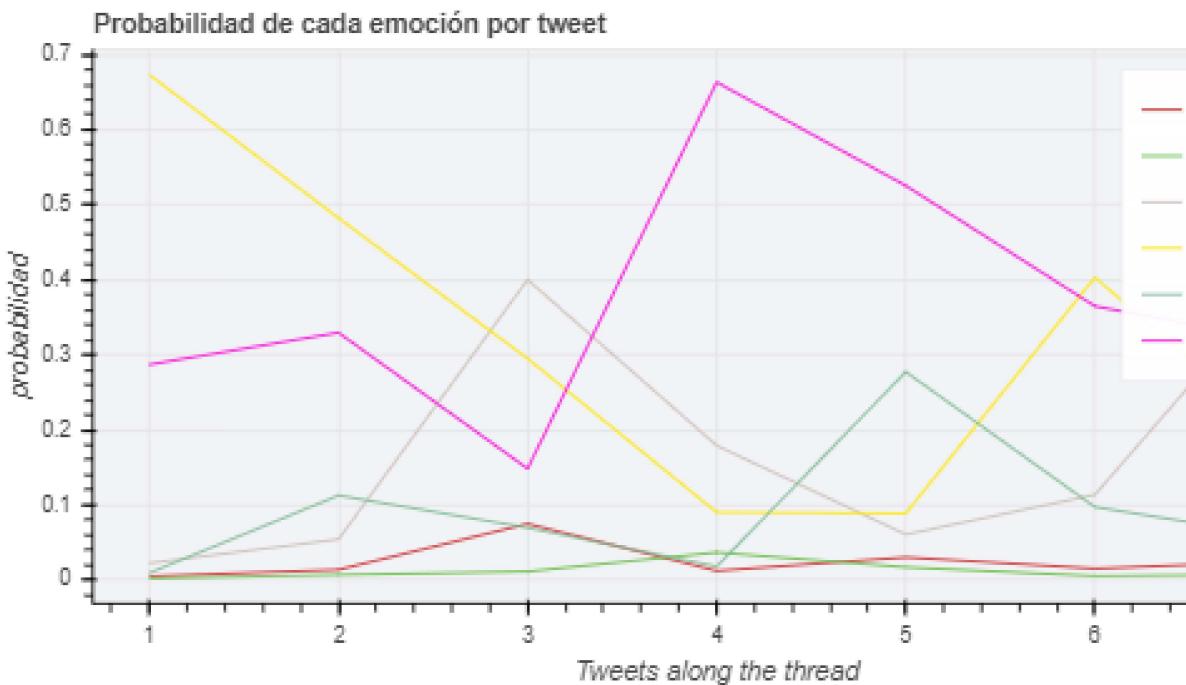
```

grafico = figure(plot_width=700,
                 plot_height=350,
                 title='Probabilidad de cada emoción por tweet',
                 background_fill_color="#f2f3f7",
                 y_axis_label='probabilidad',
                 x_axis_label='Tweets along the thread')

grafico.line(list(range(1,8)), matriz['Anger'].values, line_color='#cd2626', legend='Enojo')
grafico.line(list(range(1,8)), matriz['Disgust'].values, line_color='#64bd40', legend='Disgusto')
grafico.line(list(range(1,8)), matriz['Fear'].values, line_color='#cbbeb5', legend='Miedo')
grafico.line(list(range(1,8)), matriz['Joy'].values, line_color='#ffe700', legend='Alegria')
grafico.line(list(range(1,8)), matriz['Sadness'].values, line_color='#7cb890', legend='Pena')
grafico.line(list(range(1,8)), matriz['Surprise'].values, line_color='#ff00db', legend='Sorpresa')

show(grafico)

```



In [20]:

tabla

Out[20]:

	Tweet	Emotion
0	Extraordinary evidence at Treasury committee f...	Joy
1	The Brexiter favourite Max Fac - would cost bu...	Joy
2	How does he arrive at the figure\r\n\r\n\r\n\r\n\r\n20...	Fear
3	Theresa May's New Customs Partnership is much ...	Surprise
4	Mr Thompson said he did not expect the EU to r...	Surprise
5	Both would not be ready by 2021. Max Fac needs...	Joy
6	"We think we can manage the risk - we think we...	Fear

In [32]:

```
tabla.describe()
```

Out[32]:

	Tweet	Emotion
count	7	7
unique	7	3
top	Extraordinary evidence at Treasury committee f...	Joy
freq	1	3

Análisis de emociones según la matriz de probabilidades.

Se poseen varias alternativas para obtener la emoción del thread a través de la matriz, los cuales serán descritos a continuación:

In [46]:

```
matriz_valores = matriz.drop(columns ="Tweet")
matriz_valores
```

Out[46]:

	Anger	Disgust	Fear	Joy	Sadness	Surprise
0	0.005198	0.002203	0.022643	0.673286	0.009532	0.287138
1	0.013784	0.007089	0.054312	0.482430	0.112748	0.329637
2	0.074924	0.011434	0.400213	0.295159	0.069761	0.148510
3	0.012332	0.036829	0.179175	0.090255	0.018078	0.663330
4	0.030329	0.017072	0.060771	0.088922	0.277415	0.525490
5	0.015301	0.005518	0.113649	0.403367	0.097253	0.364913
6	0.025122	0.007756	0.403552	0.187179	0.057420	0.318970

Emoción media

Se definió como emoción media a la columna que posea el promedio más alto.

In [61]:

```
matriz_media = matriz_valores.mean().to_frame().transpose()
matriz_media
```

Out[61]:

	Anger	Disgust	Fear	Joy	Sadness	Surprise
0	0.025284	0.012557	0.176331	0.317228	0.091744	0.376856

In [84]:

```
matriz_media.sort_values(by = 0, axis = 1, ascending=False)
```

Out[84]:

	Surprise	Joy	Fear	Sadness	Anger	Disgust
0	0.376856	0.317228	0.176331	0.091744	0.025284	0.012557

In [88]:

```
matriz_emociones_media_max = list(matriz_media.sort_values(by = 0, axis = 1, ascending=False))
matriz_emociones_media_max
```

Out[88]:

```
['Surprise', 'Joy']
```

Emoción más intensa

Se definió como emoción más intensa la columna que posea el valor máximo más alto de todos.

In [127]:

```
matriz_emocion = matriz_valores.max().to_frame().transpose().sort_values(by = 0, axis = 1,
matriz_emocion
```

Out[127]:

	Joy	Surprise	Fear	Sadness	Anger	Disgust
0	0.673286	0.66333	0.403552	0.277415	0.074924	0.036829

In [128]:

```
matriz_emociones_max = list(matriz_emocion.sort_values(by = 0, axis = 1, ascending=False).i
matriz_emociones_max
```

Out[128]:

```
['Joy', 'Surprise']
```

Análisis de emociones según la tabla de emociones.

Para analizar las emociones según la tabla es posible considerar la frecuencia en la que aparecen las emociones.

Emoción Moda

Se definió como emoción moda como la emoción que presenta la mayor frecuencia de aparición

In [103]:

```
tabla.Emotion.value_counts().to_frame().transpose()
```

Out[103]:

	Joy	Surprise	Fear
Emotion	3	2	2

In [112]:

```
tabla_emociones_moda = list(tabla.Emotion.value_counts().to_frame().transpose().iloc[:,0:2])  
tabla_emociones_moda
```

Out[112]:

```
['Joy', 'Surprise']
```

Estudio de threads como un solo documento

Para estudiar los threads como un sólo documento, se unieron los tweets y luego se ingresó el dataset completo como una colección de documentos. cada fila es un thread y cada columna es la probabilidad de una emoción específica. En este caso solo se posee una forma de analizar las emociones.

Emoción Hilo

Al existir sólo un arreglo de probabilidades para cada thread, la emoción con mayor probabilidad y la emoción en la tabla es la misma, por lo que de este caso sólo se pueden obtener las primeras 2 emociones más probables para cada thread.

In [208]:

```
Ttweets = Tthreads1
Tmatrix = ekman.predict_probabilities(Ttweets).drop(columns = "Tweet")

for index, row in Tmatrix.iterrows():
    Tmatrix_emocion = row.to_frame().transpose().sort_values(by = index, axis = 1, ascending=False)
    Tmatrix_emocion_max = list(Tmatrix_emocion.iloc[:,0:2])
    print(Tmatrix_emocion_max)

['fear', 'surprise']
['Fear', 'Joy']
['Fear', 'Anger']
['Joy', 'Fear']
['Fear', 'Joy']
['Joy', 'Fear']
['Joy', 'Surprise']
['Fear', 'Joy']
['Joy', 'Surprise']
['Surprise', 'Joy']
['Fear', 'Joy']
['Joy', 'Fear']
['Fear', 'Joy']
['Anger', 'Surprise']
['Surprise', 'Joy']
['Fear', 'Surprise']
['Anger', 'Fear']
['Fear', 'Joy']
['Fear', 'Joy']
['Disgust', 'Fear']
```

Estudio del dataset completo por tweets separados.

Luego de presentar todos los métodos definidos para obtención la emoción para cada thread, se procede a calcularlos.

CSV five_ten

In [179]:

```
threads_estudio = threads1

tweets_analisis = []

emocion_media = []
emocion_presente = []
emocion_moda = []

etiquetas = ["Thread", "1ra Emocion", "2da Emocion"]

for hilo in threads_estudio:
    print(hilo)
    #Conseguir Emociones
    tweets_analisis = threads_estudio[hilo]
    matriz_hilo_analizado = ekman.predict_probabilities(tweets_analisis).drop(columns ="Twe
    tabla_hilo_analizado = ekman.predict_classes(tweets_analisis)

    #Emocion media
    matriz_media_analisis = matriz_hilo_analizado.mean().to_frame().transpose()
    matriz_emociones_media_max_analisis = list(matriz_media_analisis.sort_values(by = 0, axis = 1).tail(1))
    matriz_emociones_media_max_analisis.insert(0, hilo)
    print(matriz_emociones_media_max_analisis)

    #Emocion mas intensa
    matriz_emocion_analisis = matriz_hilo_analizado.max().to_frame().transpose().sort_values(by = 0, axis = 1)
    matriz_emociones_max_analisis = list(matriz_emocion_analisis.sort_values(by = 0, axis = 1).tail(1))
    matriz_emociones_max_analisis.insert(0, hilo)
    print(matriz_emociones_max_analisis)

    #Emocion moda
    tabla_emociones_moda_analisis = list(tabla_hilo_analizado.Emotion.value_counts().to_fra
    tabla_emociones_moda_analisis.insert(0, hilo)
    print(tabla_emociones_moda_analisis)

#Aregar a Lista de Listas

emocion_media.append(matriz_emociones_media_max_analisis)
emocion_presente.append(matriz_emociones_max_analisis)
emocion_moda.append(tabla_emociones_moda_analisis)

# Creacion de dataframes
dataframe_emocion_media = pd.DataFrame.from_records(emocion_media, columns = etiquetas)
dataframe_emocion_presente = pd.DataFrame.from_records(emocion_presente, columns = etiquetas)
dataframe_emocion_moda = pd.DataFrame.from_records(emocion_moda, columns = etiquetas)

# Guardado de dataframes

dataframe_emocion_media_1 = dataframe_emocion_media
dataframe_emocion_presente_1 = dataframe_emocion_presente
dataframe_emocion_moda_1 = dataframe_emocion_moda
['Thread 27', 'Joy', 'Surprise']
Thread 28
['Thread 28', 'Surprise', 'Sadness']
['Thread 28', 'Sadness', 'Surprise']
['Thread 28', 'Surprise', 'Fear']
Thread 29
['Thread 29', 'Surprise', 'Fear']
```

```
['Thread 29', 'Anger', 'Fear']
['Thread 29', 'Fear', 'Surprise']
Thread 3
['Thread 3', 'Surprise', 'Joy']
['Thread 3', 'Joy', 'Surprise']
['Thread 3', 'Surprise', 'Joy']
Thread 30
['Thread 30', 'Fear', 'Surprise']
['Thread 30', 'Fear', 'Surprise']
['Thread 30', 'Fear', 'Surprise']
Thread 31
['Thread 31', 'Surprise', 'Fear']
['Thread 31', 'Surprise', 'Fear']
```

CSV ten_fifteen

In [180]:

```

threads_estudio = threads2

tweets_analisis = []

emocion_media = []
emocion_presente = []
emocion_moda = []

etiquetas = ["Thread", "1ra Emocion", "2da Emocion"]

for hilo in threads_estudio:
    print(hilo)
    #Conseguir Emociones
    tweets_analisis = threads_estudio[hilo]
    matriz_hilo_analizado = ekman.predict_probabilities(tweets_analisis).drop(columns ="Twe"
    tabla_hilo_analizado = ekman.predict_classes(tweets_analisis)

    #Emocion media
    matriz_media_analisis = matriz_hilo_analizado.mean().to_frame().transpose()
    matriz_emociones_media_max_analisis = list(matriz_media_analisis.sort_values(by = 0, axis = 1).tail(1))
    matriz_emociones_media_max_analisis.insert(0, hilo)
    print(matriz_emociones_media_max_analisis)

    #Emocion mas intensa
    matriz_emocion_analisis = matriz_hilo_analizado.max().to_frame().transpose().sort_values(by = 0, axis = 1)
    matriz_emociones_max_analisis = list(matriz_emocion_analisis.sort_values(by = 0, axis = 1).tail(1))
    matriz_emociones_max_analisis.insert(0, hilo)
    print(matriz_emociones_max_analisis)

    #Emocion moda
    tabla_emociones_moda_analisis = list(tabla_hilo_analizado.Emotion.value_counts().to_fra
    tabla_emociones_moda_analisis.insert(0, hilo)
    print(tabla_emociones_moda_analisis)

#Aregar a Lista de Listas

emocion_media.append(matriz_emociones_media_max_analisis)
emocion_presente.append(matriz_emociones_max_analisis)
emocion_moda.append(tabla_emociones_moda_analisis)

# Creacion de dataframes
dataframe_emocion_media = pd.DataFrame.from_records(emocion_media, columns = etiquetas)
dataframe_emocion_presente = pd.DataFrame.from_records(emocion_presente, columns = etiquetas)
dataframe_emocion_moda = pd.DataFrame.from_records(emocion_moda, columns = etiquetas)

# Guardado de dataframes

dataframe_emocion_media_2 = dataframe_emocion_media
dataframe_emocion_presente_2 = dataframe_emocion_presente
dataframe_emocion_moda_2 = dataframe_emocion_moda

Thread 1
['Thread 1', 'Surprise', 'Fear']
['Thread 1', 'Fear', 'Surprise']
['Thread 1', 'Surprise', 'Fear']
Thread 10
['Thread 10', 'Surprise', 'Joy']

```

```
[ 'Thread 10', 'Surprise', 'Fear' ]
[ 'Thread 10', 'Surprise', 'Joy' ]
Thread 11
[ 'Thread 11', 'Surprise', 'Joy' ]
[ 'Thread 11', 'Joy', 'Surprise' ]
[ 'Thread 11', 'Surprise', 'Joy' ]
Thread 12
[ 'Thread 12', 'Joy', 'Surprise' ]
[ 'Thread 12', 'Joy', 'Surprise' ]
[ 'Thread 12', 'Joy', 'Surprise' ]
Thread 13
[ 'Thread 13', 'Surprise', 'Joy' ]
[ 'Thread 13', 'Surprise', 'Fear' ]
```

CSV fifteen_twenty

In [181]:

```
threads_estudio = threads3

tweets_analisis = []

emocion_media = []
emocion_presente = []
emocion_moda = []

etiquetas = ["Thread", "1ra Emocion", "2da Emocion"]

for hilo in threads_estudio:
    print(hilo)
    #Conseguir Emociones
    tweets_analisis = threads_estudio[hilo]
    matriz_hilo_analizado = ekman.predict_probabilities(tweets_analisis).drop(columns ="Twe
    tabla_hilo_analizado = ekman.predict_classes(tweets_analisis)

    #Emocion media
    matriz_media_analisis = matriz_hilo_analizado.mean().to_frame().transpose()
    matriz_emociones_media_max_analisis = list(matriz_media_analisis.sort_values(by = 0, axis = 1).tail(1))
    matriz_emociones_media_max_analisis.insert(0, hilo)
    print(matriz_emociones_media_max_analisis)

    #Emocion mas intensa
    matriz_emocion_analisis = matriz_hilo_analizado.max().to_frame().transpose().sort_values(by = 0, axis = 1)
    matriz_emociones_max_analisis = list(matriz_emocion_analisis.sort_values(by = 0, axis = 1).tail(1))
    matriz_emociones_max_analisis.insert(0, hilo)
    print(matriz_emociones_max_analisis)

    #Emocion moda
    tabla_emociones_moda_analisis = list(tabla_hilo_analizado.Emotion.value_counts().to_fra
    tabla_emociones_moda_analisis.insert(0, hilo)
    print(tabla_emociones_moda_analisis)

#Aregar a Lista de Listas

emocion_media.append(matriz_emociones_media_max_analisis)
emocion_presente.append(matriz_emociones_max_analisis)
emocion_moda.append(tabla_emociones_moda_analisis)

# Creacion de dataframes
dataframe_emocion_media = pd.DataFrame.from_records(emocion_media, columns = etiquetas)
dataframe_emocion_presente = pd.DataFrame.from_records(emocion_presente, columns = etiquetas)
dataframe_emocion_moda = pd.DataFrame.from_records(emocion_moda, columns = etiquetas)

# Guardado de dataframes

dataframe_emocion_media_3 = dataframe_emocion_media
dataframe_emocion_presente_3 = dataframe_emocion_presente
dataframe_emocion_moda_3 = dataframe_emocion_moda
In[10]
['Thread 10', 'Surprise', 'Joy']
['Thread 10', 'Surprise', 'Joy']
['Thread 10', 'Surprise', 'Fear']
Thread 11
['Thread 11', 'Fear', 'Surprise']
['Thread 11', 'Surprise', 'Joy']
```

```
['Thread 11', 'Fear', 'Surprise']  
Thread 12  
['Thread 12', 'Surprise', 'Fear']  
['Thread 12', 'Joy', 'Fear']  
['Thread 12', 'Surprise', 'Fear']  
Thread 13  
['Thread 13', 'Fear', 'Surprise']  
['Thread 13', 'Fear', 'Joy']  
['Thread 13', 'Fear', 'Surprise']  
Thread 14  
['Thread 14', 'Surprise', 'Fear']  
['Thread 14', 'Surprise', 'Joy']  
['Thread 14', 'Surprise', 'Fear']
```

CSV twenty_twentyfive

In [182]:

```

threads_estudio = threads4

tweets_analisis = []

emocion_media = []
emocion_presente = []
emocion_moda = []

etiquetas = ["Thread", "1ra Emocion", "2da Emocion"]

for hilo in threads_estudio:
    print(hilo)
    #Conseguir Emociones
    tweets_analisis = threads_estudio[hilo]
    matriz_hilo_analizado = ekman.predict_probabilities(tweets_analisis).drop(columns ="Twe"
    tabla_hilo_analizado = ekman.predict_classes(tweets_analisis)

    #Emocion media
    matriz_media_analisis = matriz_hilo_analizado.mean().to_frame().transpose()
    matriz_emociones_media_max_analisis = list(matriz_media_analisis.sort_values(by = 0, axis = 1).tail(1))
    matriz_emociones_media_max_analisis.insert(0, hilo)
    print(matriz_emociones_media_max_analisis)

    #Emocion mas intensa
    matriz_emocion_analisis = matriz_hilo_analizado.max().to_frame().transpose().sort_values(by = 0, axis = 1)
    matriz_emociones_max_analisis = list(matriz_emocion_analisis.sort_values(by = 0, axis = 1).tail(1))
    matriz_emociones_max_analisis.insert(0, hilo)
    print(matriz_emociones_max_analisis)

    #Emocion moda
    tabla_emociones_moda_analisis = list(tabla_hilo_analizado.Emotion.value_counts().to_fra
    tabla_emociones_moda_analisis.insert(0, hilo)
    print(tabla_emociones_moda_analisis)

#Aregar a Lista de Listas

emocion_media.append(matriz_emociones_media_max_analisis)
emocion_presente.append(matriz_emociones_max_analisis)
emocion_moda.append(tabla_emociones_moda_analisis)

# Creacion de dataframes
dataframe_emocion_media = pd.DataFrame.from_records(emocion_media, columns = etiquetas)
dataframe_emocion_presente = pd.DataFrame.from_records(emocion_presente, columns = etiquetas)
dataframe_emocion_moda = pd.DataFrame.from_records(emocion_moda, columns = etiquetas)

# Guardado de dataframes

dataframe_emocion_media_4 = dataframe_emocion_media
dataframe_emocion_presente_4 = dataframe_emocion_presente
dataframe_emocion_moda_4 = dataframe_emocion_moda

Thread 1
['Thread 1', 'Joy', 'Fear']
['Thread 1', 'Joy', 'Fear']
['Thread 1', 'Fear', 'Joy']
Thread 10
['Thread 10', 'Surprise', 'Joy']

```

```
[ 'Thread 10', 'Surprise', 'Joy' ]
[ 'Thread 10', 'Surprise', 'Joy' ]
Thread 100
[ 'Thread 100', 'Fear', 'Joy' ]
[ 'Thread 100', 'Joy', 'Fear' ]
[ 'Thread 100', 'Fear', 'Surprise' ]
Thread 11
[ 'Thread 11', 'Joy', 'Fear' ]
[ 'Thread 11', 'Joy', 'Fear' ]
[ 'Thread 11', 'Joy', 'Fear' ]
Thread 12
[ 'Thread 12', 'Joy', 'Surprise' ]
[ 'Thread 12', 'Joy', 'Surprise' ]
```

CSV twentyfive_thirty

In [183]:

```
threads_estudio = threads5

tweets_analisis = []

emocion_media = []
emocion_presente = []
emocion_moda = []

etiquetas = ["Thread", "1ra Emocion", "2da Emocion"]

for hilo in threads_estudio:
    print(hilo)
    #Conseguir Emociones
    tweets_analisis = threads_estudio[hilo]
    matriz_hilo_analizado = ekman.predict_probabilities(tweets_analisis).drop(columns ="Twe
    tabla_hilo_analizado = ekman.predict_classes(tweets_analisis)

    #Emocion media
    matriz_media_analisis = matriz_hilo_analizado.mean().to_frame().transpose()
    matriz_emociones_media_max_analisis = list(matriz_media_analisis.sort_values(by = 0, axis = 1).tail(1))
    matriz_emociones_media_max_analisis.insert(0, hilo)
    print(matriz_emociones_media_max_analisis)

    #Emocion mas intensa
    matriz_emocion_analisis = matriz_hilo_analizado.max().to_frame().transpose().sort_values(by = 0, axis = 1)
    matriz_emociones_max_analisis = list(matriz_emocion_analisis.sort_values(by = 0, axis = 1).tail(1))
    matriz_emociones_max_analisis.insert(0, hilo)
    print(matriz_emociones_max_analisis)

    #Emocion moda
    tabla_emociones_moda_analisis = list(tabla_hilo_analizado.Emotion.value_counts().to_fra
    tabla_emociones_moda_analisis.insert(0, hilo)
    print(tabla_emociones_moda_analisis)

#Aregar a Lista de Listas

emocion_media.append(matriz_emociones_media_max_analisis)
emocion_presente.append(matriz_emociones_max_analisis)
emocion_moda.append(tabla_emociones_moda_analisis)

# Creacion de dataframes
dataframe_emocion_media = pd.DataFrame.from_records(emocion_media, columns = etiquetas)
dataframe_emocion_presente = pd.DataFrame.from_records(emocion_presente, columns = etiquetas)
dataframe_emocion_moda = pd.DataFrame.from_records(emocion_moda, columns = etiquetas)

# Guardado de dataframes

dataframe_emocion_media_5 = dataframe_emocion_media
dataframe_emocion_presente_5 = dataframe_emocion_presente
dataframe_emocion_moda_5 = dataframe_emocion_moda

Thread 1
['Thread 1', 'Joy', 'Fear']
['Thread 1', 'Fear', 'Joy']
['Thread 1', 'Joy', 'Fear']
Thread 10
['Thread 10', 'Surprise', 'Joy']
```

```
[ 'Thread 10', 'Surprise', 'Fear' ]
[ 'Thread 10', 'Surprise', 'Joy' ]
Thread 100
[ 'Thread 100', 'Fear', 'Surprise' ]
[ 'Thread 100', 'Fear', 'Joy' ]
[ 'Thread 100', 'Fear', 'Surprise' ]
Thread 11
[ 'Thread 11', 'Joy', 'Surprise' ]
[ 'Thread 11', 'Fear', 'Joy' ]
[ 'Thread 11', 'Joy', 'Surprise' ]
Thread 12
[ 'Thread 12', 'Joy', 'Surprise' ]
[ 'Thread 12', 'Joy', 'Surprise' ]
```

Estudio del dataset completo por threads unidos.

CSV five_ten

In [278]:

```
Ttweets = Tthreads1
Tmatriz = ekman.predict_probabilities(Ttweets)
Tdf_unidos = df_unidos1

Tmatriz_emocion = []
Tmatriz_emocion_max = []

Temocion_max = []
dataframe_Temocion_max = []

unidos = []
u2 = []

unidos = Tdf_unidos.join(Tmatriz.set_index('Tweet'), on='Tweet')
u2 = unidos.set_index("Hilo").drop(columns = ["Tweet"])

#Sacar dos emociones maximas
for index, row in u2.iterrows():
    Tmatriz_emocion = row.to_frame().transpose().sort_values(by = index, axis = 1, ascending=False)
    Tmatriz_emocion_max = list(Tmatriz_emocion.iloc[:,0:2])
    Tmatriz_emocion_max.insert(0, index)
    print(Tmatriz_emocion_max)
    Temocion_max.append(Tmatriz_emocion_max)

#Matriz a dataframe
dataframe_Temocion_max = pd.DataFrame.from_records(Temocion_max, columns = etiquetas)

#Guardar Dataframe
dataframe_Temocion_max_1 = dataframe_Temocion_max

['Thread 1', 'Fear', 'Joy']
['Thread 10', 'Joy', 'Fear']
['Thread 100', 'Joy', 'Surprise']
['Thread 101', 'Fear', 'Surprise']
['Thread 11', 'Sadness', 'Fear']
['Thread 12', 'Anger', 'Joy']
['Thread 13', 'Sadness', 'Anger']
['Thread 14', 'Joy', 'Anger']
['Thread 15', 'Joy', 'Surprise']
['Thread 16', 'Joy', 'Fear']
['Thread 17', 'Fear', 'Surprise']
['Thread 18', 'Fear', 'Joy']
['Thread 19', 'Fear', 'Anger']
['Thread 2', 'Joy', 'Fear']
['Thread 20', 'Fear', 'Joy']
['Thread 21', 'Joy', 'Fear']
['Thread 22', 'Joy', 'Surprise']
['Thread 23', 'Fear', 'Joy']
['Thread 24', 'Joy', 'Surprise']
```

In [279]:

```
dataframe_Temocion_max_1.head()
```

Out[279]:

	Thread	1ra Emocion	2da Emocion
0	Thread 1	Fear	Joy
1	Thread 10	Joy	Fear
2	Thread 100	Joy	Surprise
3	Thread 101	Fear	Surprise
4	Thread 11	Sadness	Fear

CSV ten_fifteen

In [280]:

```
Ttweets = Tthreads2
Tmatriz = ekman.predict_probabilities(Ttweets)
Tdf_unidos = df_unidos2

Tmatriz_emocion = []
Tmatriz_emocion_max = []

Temocion_max = []
dataframe_Temocion_max = []

unidos = []
u2 = []

unidos = Tdf_unidos.join(Tmatriz.set_index('Tweet'), on='Tweet')
u2 = unidos.set_index("Hilo").drop(columns = ["Tweet"])

#Sacar dos emociones maximas
for index, row in u2.iterrows():
    Tmatriz_emocion = row.to_frame().transpose().sort_values(by = index, axis = 1, ascending=False)
    Tmatriz_emocion_max = list(Tmatriz_emocion.iloc[:,0:2])
    Tmatriz_emocion_max.insert(0, index)
    print(Tmatriz_emocion_max)
    Temocion_max.append(Tmatriz_emocion_max)

#Matriz a dataframe
dataframe_Temocion_max = pd.DataFrame.from_records(Temocion_max, columns = etiquetas)

#Guardar Dataframe
dataframe_Temocion_max_2 = dataframe_Temocion_max

['Thread 1', 'Fear', 'Surprise']
['Thread 10', 'Fear', 'Joy']
['Thread 11', 'Joy', 'Surprise']
['Thread 12', 'Surprise', 'Joy']
['Thread 13', 'Joy', 'Fear']
['Thread 14', 'Surprise', 'Sadness']
['Thread 15', 'Fear', 'Joy']
['Thread 16', 'Joy', 'Fear']
['Thread 17', 'Surprise', 'Fear']
['Thread 18', 'Fear', 'Joy']
['Thread 19', 'Joy', 'Fear']
['Thread 2', 'Disgust', 'Fear']
['Thread 20', 'Sadness', 'Joy']
['Thread 21', 'Fear', 'Joy']
['Thread 22', 'Fear', 'Joy']
['Thread 23', 'Joy', 'Surprise']
['Thread 24', 'Fear', 'Joy']
['Thread 25', 'Joy', 'Fear']
['Thread 26', 'Joy', 'Sadness']
```

In [281]:

```
dataframe_Temocion_max_2.head()
```

Out[281]:

	Thread	1ra Emocion	2da Emocion
0	Thread 1	Fear	Surprise
1	Thread 10	Fear	Joy
2	Thread 11	Joy	Surprise
3	Thread 12	Surprise	Joy
4	Thread 13	Joy	Fear

CSV fifteen_twenty

In [282]:

```
Ttweets = Tthreads3
Tmatriz = ekman.predict_probabilities(Ttweets)
Tdf_unidos = df_unidos3

Tmatriz_emocion = []
Tmatriz_emocion_max = []

Temocion_max = []
dataframe_Temocion_max = []

unidos = []
u2 = []

unidos = Tdf_unidos.join(Tmatriz.set_index('Tweet'), on='Tweet')
u2 = unidos.set_index("Hilo").drop(columns = ["Tweet"])

#Sacar dos emociones maximas
for index, row in u2.iterrows():
    Tmatriz_emocion = row.to_frame().transpose().sort_values(by = index, axis = 1, ascending=False)
    Tmatriz_emocion_max = list(Tmatriz_emocion.iloc[:,0:2])
    Tmatriz_emocion_max.insert(0, index)
    print(Tmatriz_emocion_max)
    Temocion_max.append(Tmatriz_emocion_max)

#Matriz a dataframe
dataframe_Temocion_max = pd.DataFrame.from_records(Temocion_max, columns = etiquetas)

#Guardar Dataframe
dataframe_Temocion_max_3 = dataframe_Temocion_max

['Thread 1', 'Joy', 'Fear']
['Thread 10', 'Joy', 'Fear']
['Thread 11', 'Fear', 'Joy']
['Thread 12', 'Joy', 'Disgust']
['Thread 13', 'Joy', 'Fear']
['Thread 13', 'Joy', 'Fear']
['Thread 14', 'Fear', 'Joy']
['Thread 15', 'Surprise', 'Fear']
['Thread 16', 'Joy', 'Fear']
['Thread 17', 'Joy', 'Fear']
['Thread 18', 'Surprise', 'Joy']
['Thread 19', 'Fear', 'Joy']
['Thread 2', 'Fear', 'Joy']
['Thread 20', 'Surprise', 'Joy']
['Thread 21', 'Fear', 'Anger']
['Thread 22', 'Joy', 'Fear']
['Thread 23', 'Joy', 'Sadness']
['Thread 24', 'Fear', 'Joy']
['Thread 25', 'Disgust', 'Fear']
```

In [283]:

```
dataframe_Temocion_max_3.head()
```

Out[283]:

	Thread	1ra Emocion	2da Emocion
0	Thread 1	Joy	Fear
1	Thread 10	Joy	Fear
2	Thread 11	Fear	Joy
3	Thread 12	Joy	Disgust
4	Thread 13	Joy	Fear

CSV twenty_twentyfive

In [284]:

```
Ttweets = Tthreads4
Tmatrix = ekman.predict_probabilities(Ttweets)
Tdf_unidos = df_unidos4

Tmatrix_emocion = []
Tmatrix_emocion_max = []

Temocion_max = []
dataframe_Temocion_max = []

unidos = []
u2 = []

unidos = Tdf_unidos.join(Tmatrix.set_index('Tweet'), on='Tweet')
u2 = unidos.set_index("Hilo").drop(columns = ["Tweet"])

#Sacar dos emociones maximas
for index, row in u2.iterrows():
    Tmatrix_emocion = row.to_frame().transpose().sort_values(by = index, axis = 1, ascending=False)
    Tmatrix_emocion_max = list(Tmatrix_emocion.iloc[:,0:2])
    Tmatrix_emocion_max.insert(0, index)
    print(Tmatrix_emocion_max)
    Temocion_max.append(Tmatrix_emocion_max)

#Matriz a dataframe
dataframe_Temocion_max = pd.DataFrame.from_records(Temocion_max, columns = etiquetas)

#Guardar Dataframe
dataframe_Temocion_max_4 = dataframe_Temocion_max
[ 'Thread 4', 'Fear', 'Disgust' ]
[ 'Thread 40', 'Joy', 'Sadness' ]
[ 'Thread 41', 'Joy', 'Fear' ]
[ 'Thread 42', 'Joy', 'Fear' ]
[ 'Thread 43', 'Fear', 'Joy' ]
[ 'Thread 44', 'Joy', 'Fear' ]
[ 'Thread 45', 'Joy', 'Fear' ]
[ 'Thread 46', 'Joy', 'Fear' ]
[ 'Thread 47', 'Fear', 'Joy' ]
[ 'Thread 48', 'Fear', 'Joy' ]
[ 'Thread 49', 'Fear', 'Joy' ]
[ 'Thread 5', 'Fear', 'Joy' ]
[ 'Thread 50', 'Joy', 'Surprise' ]
[ 'Thread 51', 'Fear', 'Joy' ]
[ 'Thread 52', 'Joy', 'Fear' ]
[ 'Thread 53', 'Joy', 'Fear' ]
[ 'Thread 54', 'Surprise', 'Fear' ]
[ 'Thread 55', 'Joy', 'Fear' ]
[ 'Thread 56', 'Fear', 'Joy' ]
[ 'Thread 57', 'Joy', 'Fear' ]
```

In [285]:

```
dataframe_Temocion_max_4.head()
```

Out[285]:

	Thread	1ra Emocion	2da Emocion
0	Thread 1	Fear	Joy
1	Thread 10	Joy	Fear
2	Thread 100	Joy	Fear
3	Thread 11	Fear	Surprise
4	Thread 12	Joy	Fear

CSV twentyfive_thirty

In [286]:

```
Ttweets = Tthreads5
Tmatriz = ekman.predict_probabilities(Ttweets)
Tdf_unidos = df_unidos5

Tmatriz_emocion = []
Tmatriz_emocion_max = []

Temocion_max = []
dataframe_Temocion_max = []

unidos = []
u2 = []

unidos = Tdf_unidos.join(Tmatriz.set_index('Tweet'), on='Tweet')
u2 = unidos.set_index("Hilo").drop(columns = ["Tweet"])

#Sacar dos emociones maximas
for index, row in u2.iterrows():
    Tmatriz_emocion = row.to_frame().transpose().sort_values(by = index, axis = 1, ascending=False)
    Tmatriz_emocion_max = list(Tmatriz_emocion.iloc[:,0:2])
    Tmatriz_emocion_max.insert(0, index)
    print(Tmatriz_emocion_max)
    Temocion_max.append(Tmatriz_emocion_max)

#Matriz a dataframe
dataframe_Temocion_max = pd.DataFrame.from_records(Temocion_max, columns = etiquetas)

#Guardar Dataframe
dataframe_Temocion_max_5 = dataframe_Temocion_max

['Thread 44', 'Fear', 'Joy']
['Thread 45', 'Joy', 'Surprise']
['Thread 46', 'Fear', 'Joy']
['Thread 47', 'Joy', 'Fear']
['Thread 48', 'Joy', 'Disgust']
['Thread 49', 'Joy', 'Fear']
['Thread 5', 'Joy', 'Surprise']
['Thread 50', 'Joy', 'Fear']
['Thread 51', 'Joy', 'Fear']
['Thread 52', 'Joy', 'Fear']
['Thread 53', 'Joy', 'Fear']
['Thread 54', 'Fear', 'Anger']
['Thread 55', 'Fear', 'Joy']
['Thread 56', 'Fear', 'Joy']
['Thread 57', 'Joy', 'Anger']
['Thread 58', 'Joy', 'Fear']
['Thread 59', 'Joy', 'Fear']
['Thread 6', 'Fear', 'Joy']
['Thread 60', 'Joy', 'Fear']
['Thread 61', 'Fear', 'Disgust']
```

In [287]:

dataframe_Temocion_max_5.head()

Out[287]:

	Thread	1ra Emocion	2da Emocion
0	Thread 1	Fear	Anger
1	Thread 10	Fear	Joy
2	Thread 100	Joy	Fear
3	Thread 11	Fear	Surprise
4	Thread 12	Joy	Surprise

Analisis de resultados

Luego de obtener las emociones de los threads tanto para los threads por separados, como para los tweets unidos, se procede a comparar los valores obtenidos

Emociones más presentes en el dataset en la primera emoción según métodos

Se obtendrán la frecuencia de todas las emociones presentes en el dataset, para así comparar la diferencia entre los métodos de clasificación.

CSV1

Emoción media

In [196]:

dataframe_emocion_media_1["1ra Emocion"].value_counts().to_frame().transpose()

Out[196]:

	Surprise	Fear	Joy	Sadness
1ra Emocion	58	20	20	3

Emoción más intensa

In [197]:

dataframe_emocion_presente_1["1ra Emocion"].value_counts().to_frame().transpose()

Out[197]:

	Surprise	Joy	Fear	Sadness	Anger	Disgust
1ra Emocion	38	30	24	4	3	2

Emoción moda

In [198]:

```
dataframe_emocion_moda_1["1ra Emocion"].value_counts().to_frame().transpose()
```

Out[198]:

	Surprise	Fear	Joy	Sadness
1ra Emocion	56	23	19	3

Emoción hilo

In [294]:

```
dataframe_Temocion_max_1["1ra Emocion"].value_counts().to_frame().transpose()
```

Out[294]:

	Joy	Fear	Surprise	Anger	Sadness	Disgust
1ra Emocion	46	31	12	5	4	3

CSV2

Emoción media

In [199]:

```
dataframe_emocion_media_2["1ra Emocion"].value_counts().to_frame().transpose()
```

Out[199]:

	Surprise	Joy	Fear	Sadness
1ra Emocion	61	20	16	1

Emoción más intensa

In [200]:

```
dataframe_emocion_presente_2["1ra Emocion"].value_counts().to_frame().transpose()
```

Out[200]:

	Surprise	Joy	Fear	Sadness	Anger
1ra Emocion	38	34	23	2	1

Emoción moda

In [201]:

```
dataframe_emocion_moda_2["1ra Emocion"].value_counts().to_frame().transpose()
```

Out[201]:

	Surprise	Joy	Fear	Sadness
1ra Emocion	56	21	17	4

Emoción hilo

In [295]:

```
dataframe_Temocion_max_5["1ra Emocion"].value_counts().to_frame().transpose()
```

Out[295]:

	Joy	Fear	Surprise	Sadness	Disgust	Anger
1ra Emocion	52	35	7	3	2	1

CSV3

Emoción media

In [202]:

```
dataframe_emocion_media_3["1ra Emocion"].value_counts().to_frame().transpose()
```

Out[202]:

	Surprise	Fear	Joy	Sadness
1ra Emocion	60	22	16	1

Emoción más intensa

In [203]:

```
dataframe_emocion_presente_3["1ra Emocion"].value_counts().to_frame().transpose()
```

Out[203]:

	Surprise	Joy	Fear	Anger	Sadness	Disgust
1ra Emocion	41	30	23	2	2	1

Emoción moda

In [204]:

```
dataframe_emocion_moda_3["1ra Emocion"].value_counts().to_frame().transpose()
```

Out[204]:

	Surprise	Fear	Joy	Anger	Sadness
1ra Emocion	58	21	18	1	1

Emoción hilo

In [296]:

```
dataframe_Temocion_max_3["1ra Emocion"].value_counts().to_frame().transpose()
```

Out[296]:

	Joy	Fear	Surprise	Disgust	Sadness	Anger
1ra Emocion	47	36	8	6	3	1

CSV4

Emoción media

In [205]:

```
dataframe_emocion_media_4["1ra Emocion"].value_counts().to_frame().transpose()
```

Out[205]:

	Joy	Fear	Surprise
1ra Emocion	48	31	21

Emoción más intensa

In [206]:

```
dataframe_emocion_presente_4["1ra Emocion"].value_counts().to_frame().transpose()
```

Out[206]:

	Joy	Fear	Surprise	Anger
1ra Emocion	48	29	21	2

Emoción moda

In [207]:

```
dataframe_emocion_moda_4["1ra Emocion"].value_counts().to_frame().transpose()
```

Out[207]:

	Joy	Fear	Surprise
1ra Emocion	42	31	27

Emoción hilo

In [297]:

```
dataframe_Temocion_max_4["1ra Emocion"].value_counts().to_frame().transpose()
```

Out[297]:

	Joy	Fear	Surprise	Disgust	Sadness	Anger
1ra Emocion	53	32	7	4	3	1

CSV5

Emoción media

In [193]:

```
dataframe_emocion_media_5["1ra Emocion"].value_counts().to_frame().transpose()
```

Out[193]:

	Joy	Surprise	Fear
1ra Emocion	67	19	14

Emoción más intensa

In [194]:

```
dataframe_emocion_presente_5["1ra Emocion"].value_counts().to_frame().transpose()
```

Out[194]:

	Joy	Fear	Surprise	Sadness
1ra Emocion	63	19	17	1

Emoción moda

In [195]:

```
dataframe_emocion_moda_5["1ra Emocion"].value_counts().to_frame().transpose()
```

Out[195]:

	Joy	Surprise	Fear
1ra Emocion	59	26	15

Emoción hilo

In [298]:

```
dataframe_Temocion_max_5["1ra Emocion"].value_counts().to_frame().transpose()
```

Out[298]:

	Joy	Fear	Surprise	Sadness	Disgust	Anger
1ra Emocion	52	35	7	3	2	1

In []: