Trabajo Práctico 1: Análisis de Sentimientos

Introducción:

Objetivo y descripción de la tarea: Realizar una tarea de análisis de sentimientos sobre un conjunto de comentarios extraídos de Twitter en inglés con los hashtags: #coronavirus, #coronavirusoutbreak, #coronavirusPandemic, #covid19, #covid_19, #epitwitter, #ihavecorona, #StayHomeStaySafe, #TestTraceIsolate.

Los comentarios pertenecen a la segunda quincena de abril de 2020.

Este trabajo estaba realizado con dos conjuntos:

El primero con los ***tweets clasificados según los emoticones***, donde hay **6309 tweets en cada categorías** (alegría, apoyo, tristeza y enojo), es decir **25 236 en total.**

El segundo con los ***tweets sin clasificar,*** donde hay **3 000 000 tweets**

Parte 1:

Esta parte estaba realizada en Python con las librerías “pandas”, “numpy” y “re”

Primero, concatenamos los 15 archivos .csv de dataset para limpiar todos los datos con el archivo **“*limpieza.py*”**

Nos quedamos sólo con los tweets en inglés y la columna “text” del .csv. Sacamos todas la información que no es importante como las fechas, las URLs, los caracteres especiales, los números y solo los hashtags que hacen referencia al virus.

Después, definimos 4 categorías con los emoticones indicados en el enunciado del TP y donde agregamos otros que se encontraron en la lista completa de los emojis del link del TP.

Hay 13 emojis que representan la categoría “alegría”, 10 para la categoría “tristeza”, 7 para “enojo” y 7 para “apoyo”.

Si para una línea dada hay emoticones que indican más de un conjunto posible, dicha línea está almacenada en el .csv donde se ponen los tweets que no están en una categoría.

Cada una de las 4 categorías está almacenada en un archivo .csv.

Segundo: con el archivo ***"conjunto_etiquetado.py"***, vemos cuál es la categoría que contiene la menor cantidad de tweets y acortamos las otras al tamaño de la menor, para que cada una contenga el mismo número de tweets.

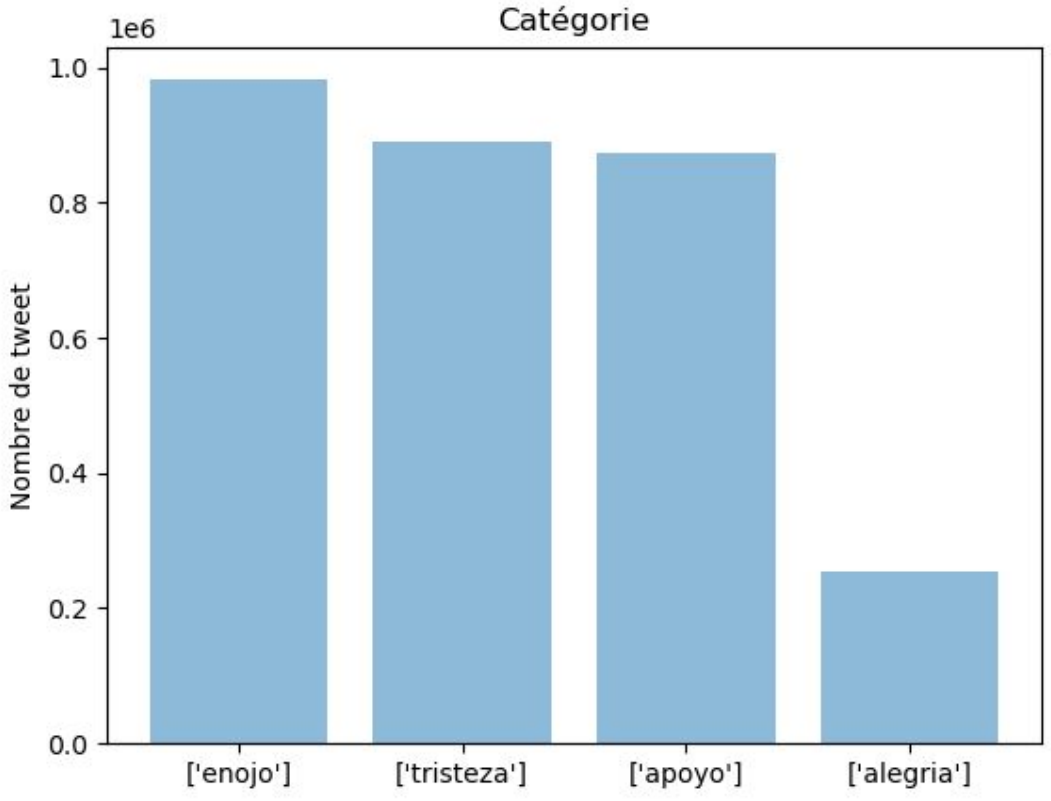
Concatenamos las 4 categorías de tweets etiquetadas juntas en el archivo .csv "conjunto_etiquetado_balanced.csv". Sacamos los emojis y podemos hacer el clasificador.

Finalmente, el archivo ***"classificateur.py"*** permite *tokenizar* los datos y entrenar un clasificador "Bayes Naïves". El 66% de los datos son para entrenarlos y el 33% restante para validar su precisión, *recall* y medida F1. Con el conjunto "conjunto_etiquetado_balanced.csv" de 25236 tweets, obtenemos la siguiente información:

```
(25236, 11464)
[[ 646  467  345  643]
 [ 115 1262  288  408]
 [ 102  207 1258  521]
 [ 116  187  376 1387]]
              precision    recall  f1-score   support

     alegria       0.66      0.31      0.42      2101
       apoyo       0.59      0.61      0.60      2073
       enojo       0.55      0.60      0.58      2088
    tristeza       0.47      0.67      0.55      2066

    accuracy                           0.55      8328
   macro avg       0.57      0.55      0.54      8328
weighted avg       0.57      0.55      0.54      8328

precision= 0.5467098943323727
nombre de tweet test  = 2999999
[enojo]       981516
[tristeza]    889149
[apoyo]       874472
[alegria]     254861
```

Aquí encontramos la matriz de confusión, la precisión, *recall* y medida F1 del clasificador entrenado.

Después, podemos clasificar cada línea del conjunto de los tweets que no estaban en una categoría y obtenemos los siguientes resultados:



Debate:

A pesar de todos los tweets que tenemos en cada categoría, la precisión del clasificador es de solamente 54,67% que no es un resultado suficientemente bueno. Al principio era de 45%, antes de que utilizáramos todos los archivos .csv y agregáramos otros emojis a cada categoría. Sin embargo, verificamos los .csv de cada categoría, revisamos aleatoriamente las frases de los tweets que estaban clasificados y que parecieron bastante buenos.

Además, probamos ver si el clasificador estaba funcionando bien, tomando frases como “I am happy, very happy”, “I am sad, very sad” (y otras para las dos otras), y clasificó bien en las buenas categorías.

Para terminar esta parte, vimos también que la cantidad de tweets estaba en el siguiente orden: “enojo”, la más importante; ”tristeza” y “apoyo”, donde la cantidad es aproximadamente la misma; y “alegría”, la categoría menos importante. Los porcentajes son los siguientes:

enojo = 32,71% ; tristeza = 29,63%; apoyo = 29,14%, alegría = 8,49%

numéro de tweets positivos = 37,63%; numéro de tweets negativos = 62,37%

Estos resultados pueden interpretarse así porque, durante esas dos semanas de abril, la gente estaba enojada por la cuarentena, con la cual debía quedarse en su casa y el enojo aumentó por culpa de todas las personas que no respetaron las medidas dirigidas a la prevención de la propagación del virus.

Hubo tristeza también con todos los muertos durante este periodo. La categoría “apoyo” podría representar a todas las personas que se fueron positivas y apoyaron a otras.

Sin embargo, después de la información del clasificador, vimos que hay demasiada incertidumbre para realmente interpretar los tweets.

Parte 2:

Se realizó esta parte en Python y con las librerías “pandas”, “nltk” y “sentiwordnet” (ya que no conocía Java, y aunque no conocía tampoco Python, preferí continuar utilizando este lenguaje de programación).

Ahora, por el conjunto de tweet clasificados, tomemos el archivo “conjunto_etiquetado_balanced.csv” y sustituyamos las etiquetas “apoyo” y “alegría” por “1” y los dos otros por “0”.

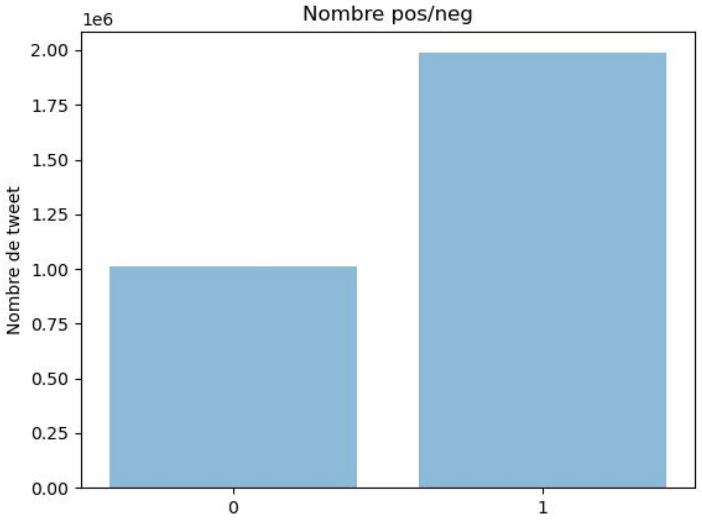
Segundo, tokenizemos los datos porque los tweets ya están limpios.

Entrenamos el clasificador y obtenemos la siguiente información:

```
[[1787 2367]
 [1130 3043]]
              precision    recall  f1-score   support

           0       0.61      0.43      0.51      4154
           1       0.56      0.73      0.64      4173

    accuracy                           0.58      8327
   macro avg       0.59      0.58      0.57      8327
weighted avg       0.59      0.58      0.57      8327

precision= 0.580040831031584
```

Aquí, la precisión del clasificador es más importante que en la última parte, con el 58% de precisión, pero vemos que con la matriz de confusión hay mucha confusión con algunos tweets negativos que están clasificados como positivos.

En cambio, la proporción de tweets positivos y negativos están invertida:



número de tweets positivos = 66,26% número de tweets negativos = 33,73%

Se verificó si se habían invertido los 1 y los 0, pero no. Esto se puede explicar con la precisión de los dos clasificadores que no está bien.

Pero, podemos ver que, después de la clasificación de los tweets según los emojis, el número de tweets en las categorías positivas es de 89 969 (80,65%) y la otra negativa de 21 576 (19,35%).

```
Python - conjunto_etiquetado.py:74 ✔

tweets enojos = 6309
tweets alegria = 35243
tweets apoyo = 54726
tweets tristeza = 15267
nb tweet 1 categorie = 6309
nb tweet total classifies pour entrainement = 25236
[Finished in 3.126s]
```

Además, este clasificador no es muy fiable porque se identificó un tweet clasificado como 1 donde estaba escrito: *"Coronavirus World surpasses two million confirmed cases"*.

Por otra parte, con este método, la clasificación toma mucho más tiempo.

Conclusión:

Con estos dos modelos, pudimos ver dos maneras para clasificar tweets que tratan el Covid. La precisión no resultó la esperada, probablemente porque depende mucho de los emojis que utilizamos y de la manera que definimos cada categoría. Con este TP se puede ver la dificultad de la disciplina de la ciencia de datos, donde muchos parámetros deben tomados en cuenta.