



ÁLGEBRA APLICADA

Prof: Maglis Mujica

Autores: Juan Nocetti

Renzo Giacometti

Ionas Josponis

Año 2023

Índice

Introducción	2
Objetivos	2
Marco Teórico	3
Herramientas	4
Desarrollo	4
Selección de tweets	6
Conclusiones	12
Conclusiones personales	13
Pasos a futuro	13
Bibliografía	13

Introducción

En este proyecto, hemos desarrollado un programa en Python que analiza tweets para determinar su nivel de positividad, neutralidad o negatividad, así como evaluar la calidad de las palabras utilizadas en estos tweets. Utilizando conceptos matemáticos como vectores y herramientas de programación, exploramos tweets relacionados con una sesión musical para lograr analizar donde caen en una escala emocional.

Objetivos

- 1- Diseñar un programa en Python el cual analice tweets para ver que tan positivos, neutros o negativos son.
- 2- Crear una funcionalidad que devuelva la calidad de las palabras de cada tweet.
- 3- Buscar una serie de tweets para probar el programa.

Marco Teórico

Vector: Lista finita de números que se pueden encontrar representados numéricamente de dos formas distintas:

Vector columna:

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

Vector renglón: [1, 2, 3]

Elementos: Los elementos son los valores que están dentro de los vectores.

Tamaño (Dimensión o longitud): Es el número de elementos que contiene.

Escalar: Es un vector que contiene un único elemento que es un número que pertenece al conjunto de números enteros. Se representa con letras griegas.

Subvector: Un subvector son ciertos elementos que están contenidos dentro de otro vector.

Producto entre vector-escalar: La multiplicación entre un vector y un escalar consiste en hacer el producto elemento a elemento. En Python lo representamos con '*'. Por ejemplo:

$$\vec{x} = \alpha \vec{v} = \alpha \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_{n-1} \end{bmatrix} = \begin{bmatrix} \alpha \cdot v_0 \\ \alpha \cdot v_1 \\ \vdots \\ \alpha \cdot v_{n-1} \end{bmatrix}$$

Producto interno de vectores: El producto interno entre dos vectores es una multiplicación de elemento a elemento respetando sus respectivos subíndices, luego de dicho producto sumamos el resultado de cada multiplicación para obtener el total. Por ejemplo:

$$\vec{a} \cdot \vec{b} = (1, 2, 3) \cdot (4, 5, 6) = 1 \cdot 4 + 2 \cdot 5 + 3 \cdot 6 = 4 + 10 + 18 = 32$$

Herramientas

- **IDE:** Es un entorno de desarrollo para poder programar, en nuestra aplicación usamos PyCharm de JetBrains y Visual Studio Code.
- **Python:** Lenguaje de desarrollo.
- **GitHub:** Es una herramienta para desarrollar software colectivamente.
- **Numpy:** Librería de Python que nos facilita algunas operaciones entre vectores.

Desarrollo

Para desarrollar este programa creamos dos clases principales, la clase utils y main.

La clase utils tiene un constructor donde se le pasa las palabras que queremos buscar su ocurrencia dentro de los tweets, también las positivas, negativas y neutras. Además contendrá todos los métodos necesarios para analizar la calidad de las palabras de los tweets y para calificar si el tweet es bueno, neutral o malo.

clearTweet: Es una función que elimina todos los signos de puntuación y devuelve el tweet “limpio”(sin signos de puntuación)

```
def clearTweet(self, tweet): # Eliminar signos de puntuación
    tweet = tweet.lower()
    clearedTweet = tweet.translate(str.maketrans('', '', string.punctuation))

    # Dividir el texto en palabras
    words = clearedTweet.split()

    return words
```

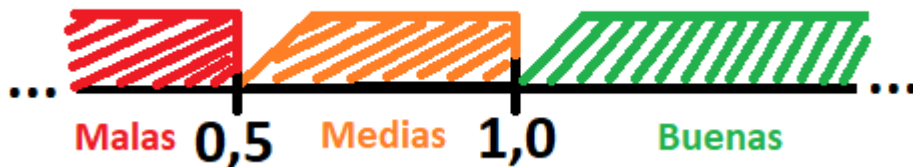
stringCounter: Es un método que devuelve el vector w donde en cada posición está la cantidad de veces que aparece dicha palabra en el tweet.

```
usage: -> ionasjospo +1*
def stringCounter(self, phrase): # Devuelve el vector w donde en cada posición está la cantidad de veces que aparece la palabra en el tweet.
    w = self.initialize()
    words = self.clearTweet(phrase)
    for actualWord in words:
        for i, word in enumerate(self.wordsToCount):
            if word == actualWord:
                w[i] += 1
    return w
```

classifyTweet: Devuelve el vector s (de tamaño 3) el cual contiene la cantidad de palabras positivas, negativas y neutras.

```
def classifyTweet(self, phrase):
    words = self.clearTweet(phrase)
    s = self.initialize()
    for w in words:
        if w in self.positiveWords:
            self.s[0] = self.s[0] + 1
        elif w in self.neutralWords:
            self.s[1] = self.s[1] + 1
        elif w in self.negativeWords:
            self.s[2] = self.s[2] + 1
    return self.s
```

avg(w) o avg(s): Devuelve la calidad de las palabras de cada tweet. Si el promedio es menor o igual a 1 el resultado va a ser con incertidumbre. Si el promedio es mayor a 1 tenemos suficiencia de palabras. Finalmente si es menor a 0.5 la calidad es mala.



```
def avg(self, w):
    n = len(w)
    alpha = (1 / n)
    r = alpha * sum(w)

    if 0.5 < r <= 1:
        return f"El tweet tiene una calidad media --> {r}"
    elif r > 1:
        return f"El tweet tiene una calidad buena --> {r}"
    else: # r < (0.5)
        return f"El tweet tiene una calidad mala --> {r}"
```

score(): Devuelve si el tweet es positivo(mayor a 0), neutral(0) o negativo(menor que 0).

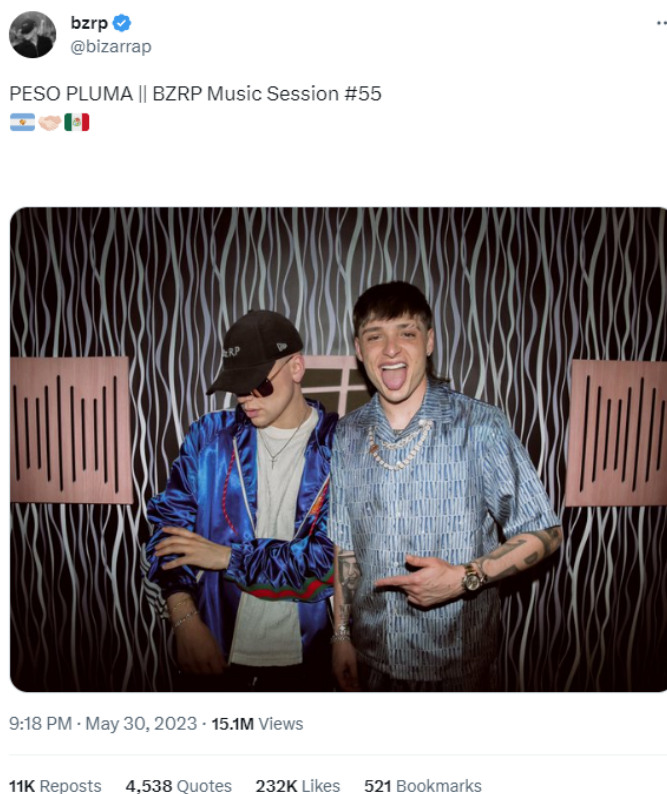
```
def score(self, s):
    v = np.array([1, 0, -1])
    score = s @ v
    if score > 0:
        return f"El tweet es positivo --> {s @ v}"
    elif score == 0:
        return f"El tweet es neutral --> {s @ v}"
    else:
        return f"El tweet es negativo --> {s @ v}"
```

Dentro de la clase main tenemos los tweets y llamamos a las funciones.

Selección de tweets

Nos basamos en la sesión del productor argentino conocido como “Bizarrap”, el cual sube contenido musical de artistas internacionales a redes sociales como youtube.

Específicamente nos enfocamos en una de estas creaciones en la cual participó el cantante mexicano “Peso Pluma”, ya que al cantar un género distinto a lo que usualmente se escucha en países latinoamericanos, generó controversia creando múltiples y distintas opiniones.



Los tweets elegidos fueron:

1:



Carla

@Caarlaap_21

...

Probablemente de las peores canciones serías que he escuchado y la peor bzip session de la historia, Peso Pluma en una bzip session no iba a acabar bien 🙄

2:



VIVI ❄️

@VIVI_Gam3

...

"Es una pena que no sea la 17472937462929372929 vez que sea una session de reguetón o trap. Cómo se atreve a hacer algo diferente? Cómo se atreve a poner instrumentos de verdad y no puro auto tune?"

1:29 p. m. · 1 jun. 2023 · 434 Reproducciones

6 Me gusta



3:



Vale

@_laparce_

...

A mi si me gusto la bzip session de peso pluma, por que tanto hate? 🙄

4:02 p. m. · 2 jun. 2023 · 25,9 mil Reproducciones

4 Reposts 2 Citas 245 Me gusta 2 Elementos guardados



4:



Pablo
@Pablo1106193181

...

Es un temazo y entre más lo escuchas más adictivo es

4:04 p. m. · 2 jun. 2023 · 566 Reproducciones

1 Me gusta



5:



Family Values Tour
@DobleUppercut

...

¿entonces la bzip session de peso pluma es una canción normal de peso pluma con un par de arreglos? porque la música está grabada con instrumentos reales por la banda y la letra también es de ellos.

1:37 a. m. · 1 jun. 2023 · 1.475 Reproducciones

1 Repost 36 Me gusta



6:



Futbol Gourmet
@Futbol_Gourmet

...

Hablan de mi Peso Pluma como si Quevedo tuviera otro tema que no sea su Session de BZRP 🤔🤔🤔🤔



follacamiones24 @follacamiones24 · 6 jul.

Os reto a todos y cada uno de mis seguidores que me digáis más de 3 canciones del caída de pluma este sin mirar spotify
twitter.com/Indie5051/stat...

6:44 p. m. · 6 jul. 2023 · 7.593 Reproducciones

3 Reposts 169 Me gusta 1 Elemento guardado



7:



Yagami TNG ✓
@yagamitng

...

La BZRP Session de Peso Pluma llega a 100M en YouTube en menos de un mes

A este paso, la polémica canción se va a convertir en una de las más vistas de la historia 🇲🇪

8:



Mariano_garmez
@MarianoGarmez

...


Bro es que si es famosa en México es seguro que será famosa si somos demasiados, a los que nos gusta nuestra música nacional la apoyamos y ya esta.
Independientemente del hete del exterior

3:04 p. m. · 23 jun. 2023 · 258 Reproducciones

1 Repost 6 Me gusta



9:



Kuvia
@Lucy_Gifer






...

La bzrp session de Peso Pluma no ha gustado porque es un género muy particular de México al que no estamos acostumbrados.

Pero con la sesión de Arcángel dijimos que Biza estaba empezando a ser repetitivo y aquí ha traído algo diferente, no podemos quejarnos en ese sentido.

4:03 a. m. · 1 jun. 2023 · 3.544 Reproducciones

2 Reposts 54 Me gusta



10:



Luego seleccionamos palabras:

```
words = ['mas', 'feliz', 'gusto', 'session', 'odio',  
'problema', 'peores', 'peor', 'temazo', 'adictivo']  
positive words = ['mas', 'temazo', 'feliz', 'gusto',  
'adictivo']  
neutral_words = ['session']  
negative words = ['odio', 'problema', 'peores', 'peor']
```

Con la ejecución del programa respondemos las primeras 3 preguntas planteadas:

- ¿Qué tuit es más positivo?
- ¿Qué tuit es más negativo?
- ¿Cuál es tu calidad promedio?

```
El tweet más positivo es: Es un temazo y entre más lo escuchas más adictivo es  
Score: El tweet es positivo --> 2.0  
El tweet más negativo es: Probablemente de las peores canciones serías que he escuchado y la peor bzrp session de la historia, Peso Pluma en una bzrp session no iba a acabar bien  
Score: El tweet es negativo --> -2.0  
El promedio de calidad de los tweets es de: 0.15
```

•¿Cómo interpretas avg(s) y score(s) ?

En primer lugar, el avg(s) lo interpretamos como el promedio de sentimiento de un cierto tweet, dependiendo completamente de las palabras seleccionadas anteriormente y el resultado de esta operación en caso de ser alto, será un tweet con influencia positiva, cuanto más cerca del 0 sea, significa que es neutro y en caso de ser un resultado menor a 0 tendremos un tweet negativo.

El score sentimental lo entendemos como una medida cuantitativa que indicará el sentimiento del tweet (al igual que el avg(s) depende de las palabras elegidas), tendremos como resultado la negatividad o positividad del enunciado. El resultado es calculado a partir de una operación de producto escalar de vectores, entre el vector de sentimientos (s) y un vector ponderado ([1,0,-1]).

•¿Cómo relacionas la calidad con score(s) y avg(s) ?

Los resultados de ambas funciones nos indican el tipo de enunciado con el que tratamos, analizando la calidad del sentimiento. Un resultado de calidad alto (valores altos de avg(s) y score(s)), representa una tendencia del enunciado a ser más positivo, en cambio un resultado de calidad bajo (valores bajos de avg(s) y score(s)) nos indicaría que tratamos con un texto con tendencia negativa.

Ambas son medidas cuantitativas, que dependen de diversos factores tales como una buena elección de palabras, ya sea por más apariciones en los tweets o por una buena clasificación de las mismas, y tweets que se basen en el mismo tema. El análisis será más exacto cuanto más palabras sean seleccionadas y cuanto más apariciones en los enunciados tengan. Esto nos permitirá ver de forma más explícita las diferencias entre los tweets, pudiendo obtener los que son positivos y aquellos que no.

Conclusiones

- 1- Se creó el programa correctamente, es capaz de realizar un análisis de sentimientos de cada tweet.
- 2- La funcionalidad para ver la calidad de palabras del tweet fue creada con éxito.
- 3- Los tweets fueron elegidos sobre una canción musical ya que todas las personas tienen gustos distintos por lo que se generan muchas opiniones.
- 4-Fue un trabajo que logró abarcar todo lo dado en clase hasta el momento.

Conclusiones personales

Pensamos que fue un proyecto que promueve el trabajo en equipo, ya que tiene muchas partes y combina el trabajo tanto en código, como en documentos y navegación de información, por lo tanto, permite que cada uno de nuestros integrantes presente más facilidades en algunas tareas y sea un trabajo más fácil de dividirse.

Se lograron los objetivos del proyecto y se remarcó todo lo visto en clase, esto fue gracias a una buena organización, comunicación y utilización de herramientas necesarias que facilitaron un gran trabajo.

Pasos a futuro

-Se puede mejorar el programa notoriamente si realizamos una clase Tweet que contenga los atributos utilizados(w, s, avg(w), avg(s), score()) ya que al momento de verificar cuál fue el más positivo y/o negativo sólo podíamos obtener el score del tweet no la frase, por lo que tuvimos que buscar a mano la variable del tweet para imprimirla. Ya que eran 10 tweet era fácil pero si usamos este programa para una cantidad significativamente mayor de tweets sería imposible.

-Otra mejora opcional es agregar una interfaz gráfica para visualizar mejor los datos.

Bibliografía

Usamos solamente las presentaciones de clase de Google Colabs.