Tarea 2: Análisis de Sentimientos

César jair Tamez Juárez

May 2022

1 Datos

El dataset fue obtenido de Kaggle, fue creado por Aman Miglani, el nombre del dataset es “Coronavirus tweets NLP- Text Clasification” , consta de dos archivos CSV pero solo se trabajo con uno de ellos que cuenta con un total de 3,798 tweets que fueron extraídos de Twitter desde marzo del 2020, estos tweets fueron escritos por personas que viven en Estados Unidos, El dataset contiene la siguiente información, "UserName" el cual es un numero ya que por políticas de privacidad no se puede poner el nombre de la cuenta que lo escribió, "ScreenName", "Location" el cual es el lugar de donde proviene el tweet, "TweetAt" es la fecha en que se publicó el tweet, "OriginalTweet" el cual es el dato más importante ya que este es el texto o tweet que se escribió y por ultimó "Sentiment" el cual hace referencia al sentimiento del texto el cual puede ser negativo, positivo, neutral, extremadamente negativo y extremadamente positivo. A continuación, podemos ver una pequeña fracción del dataset para ilustrar la forma que tienen los textos: En total contamos con 599 tweets clasificados como “Ex-

1	UserName	ScreenName	Location	TweetAt	OriginalTweet	Sentiment
2	1	44953	NYC	02/03/2020	TRENDING: New Yorkers encounter empty supermarket shelves (pictured, Wegmans in Brooklyn), sold-out online grocers (FoodKick, MaxDelivery) as #coronavirus-fearing shoppers stock up https://t.co/Gr76pcrLWh https://t.co/ivMKMsqdT1	Extremely Negative
3	2	44954	Seattle, WA	02/03/2020	When I couldn't find hand sanitizer at Fred Meyer, I turned to #Amazon. But $114.97 for a 2 pack of Purell??!!Check out how #coronavirus concerns are driving up prices. https://t.co/ygbipBflMY	Positive
4	3	44955		02/03/2020	Find out how you can protect yourself and loved ones from #coronavirus. ?	Extremely Positive

Figure 1: Parte del Dataset

tremely Positive”, 947 tweets como “Positive”, 619 como “Neutral”, 1041 tweets como “Negative” y 592 tweets como” Extremely Negative”. Ahora hay que realizar el preprocesamiento de los Tweets para poder ser utilizados en un análisis o algoritmo

2 Preprocesamiento

Antes de iniciar el preprocesamiento se decidió eliminar algunas columnas de datos que consideramos que no son de mucha utilidad, se elimino la columna de ScreenName, Location y TweetAt, de esta manera solo nos quedamos con la información que seria el texto, el sentimiento y UserName que servirá como llave primaria. A continuación, explicaremos todo el preprocesamiento que se le realizo a los textos:

- Los textos originalmente tenían una forma similar a la siguiente:

  TRENDING: New Yorkers encounter empty supermarket shelves (pictured, Wegmans in Brooklyn), sold-out online grocers (FoodKick, MaxDelivery) as coronavirus-fearing shoppers stock up https://t.co/Gr76pcrLWh https://t.co/ivMKMsqdT1

- Lo primero que se realizo fue el normalizar los textos, es decir poner todas las letras del texto en minúsculas:

  trending: new yorkers encounter empty supermarket shelves (pictured, wegmans in brooklyn), sold-out online grocers (foodkick, maxdelivery) as coronavirus-fearing shoppers stock up https://t.co/gr76pcrlwh https://t.co/ivmkmsqdt1

- Después se decidió eliminar todos los números que aparecieran en el texto:

  trending: new yorkers encounter empty supermarket shelves (pictured, wegmans in brooklyn), sold-out online grocers (foodkick, maxdelivery) as coronavirus-fearing shoppers stock up https://t.co/grpcrlwh https://t.co/ivmkmsqdt

- Lo siguiente fue reducir los espacios en blanco a un solo espacio:

  trending: new yorkers encounter empty supermarket shelves (pictured, wegmans in brooklyn), sold-out online grocers (foodkick, maxdelivery) as coronavirus-fearing shoppers stock up https://t.co/grpcrlwh https://t.co/ivmkmsqdt

- Se eliminaron las direcciones URL que estuvieran en los textos

  trending: new yorkers encounter empty supermarket shelves (), sold-out online grocers () as coronavirus-fearing shoppers stock up

- Después se eliminaron los signos de puntuación, esto con el fin de eliminar los arrobas y hashtags que se suelen utilizar en los tweets:

  trending new yorkers encounter empty supermarket shelves soldout online grocers as coronavirusfearing shoppers stock up

- Finalmente se realiza la Tokenizacion del texto resultante, lo cual consiste en separar a cada palabra del texto:

  ['trending', 'new', 'yorkers', 'encounter', 'empty', 'supermarket', 'shelves', 'soldout', 'online', 'grocers', 'coronavirusfearing', 'shoppers', 'stock']

- Y a partir de estos tokens se obtiene el origen de cada una de las palabras con la lematización:

  ['trend', 'new', 'yorker', 'encount', 'empti', 'supermarket', 'shelv', 'soldout', 'onlin', 'grocer', 'coronavirusfear', 'shopper', 'stock']

Una vez que se realizó el preprocesamiento, los datos están listos para ser utilizados, por lo que el siguiente paso es realizar un análisis de sentimiento de cada uno de los textos esto con el fin de verificar que tan buenas son las clasificaciones originales.

3 Análisis de Sentimiento

Estos tweets fueron utilizados en un proyecto anterior de clasificación de textos, dicho proyecto no obtuvo buenos resultados, y al analizar las palabras correspondientes a cada una de las clasificaciones, estas no se diferenciaban mucho por lo que se consideró que las etiquetas correspondientes a cada palabra podrían estar incorrectas, debido a esto se realizara un análisis de sentimiento de los tweets para realizar una nueva clasificación, para esto utilizaremos tres librerías distintas, "textblob", "VaderSentiment" y utilizando "sentiwordnet", a continuación veremos como se utilizaron cada una de estas.

3.1 TextBlob

Con la librería "TextBlob" podemos obtener dos métricas con las cuales podemos clasificar textos, la polaridad y la subjetividad, la polaridad es un valor que va de -1 a 1 y este expresa el sentimiento en si del texto de manera objetiva, mientras que la subjetividad es un valor que va de 0 a 1 y esta expresa el sentimiento de manera subjetiva. Para nuestro trabajo se decidió utilizar la polaridad ya que su rango de valores era mucho más fácil de ajustar a nuestro numero de clases. Para realizar el análisis de sentimiento con esta librería se definieron las siguientes funciones:

```python
from textblob import TextBlob
# Función para obtener la polaridad
def getPolarity(review):
    return TextBlob(review).sentiment.polarity

# Función para realizar la clasificación
def analysis(score):
    if score <= -0.5:
        return 'Extremely Negative'
    elif score > -0.5 and score < 0:
        return 'Negative'
    elif score == 0:
        return 'Neutral'
    elif score > 0 and score < 0.5:
        return 'Positive'
    else:
        return 'Extremely Positive'
```

Figure 2: Parte del Dataset

Como podemos ver en la figura 2, la función "getPolarity" lo que hace es obtener la polaridad de cada uno de los textos, mientras que la función "analysis" realiza la clasificación de los textos, se tomo que todo texto con polaridad menor o igual a -0.5 seria "EXtremely Negative", mientras que los que tuvieran la polaridad entre -0.5 y 0 serian "Negative", simétricamente del lado positivo, si la polaridad estaba entre 0.5 y 1 se clasificaría como "Extremely Positive" y entre 0 y 0.5 seria "Positive", ientras que si la polaridad es igual a 0 este sería clasificado como "Neutral".

3.2 VaderSentiment

Con la librería "VaderSentiment", se puede obtener una métrica para el sentimiento del texto, al igual que la polaridad de la librería anterior, esta métrica también va de -1 a 1, esta librería lo que hace es encontrar para cada una de las palabras del texto las probabilidades la probabilidad de que estas sean negativas, neutras o positivas y a partir de estas probabilidades es que obtiene esa métrica final que nos dice la polaridad del texto. Al igual que en la librería anterior se definieron dos funciones que son las siguientes:

```python
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
analyzer = SentimentIntensityAnalyzer()

# Funcion para calcular el sentimiento vader
def vadersentimentanalysis(review):
    vs = analyzer.polarity_scores(review)
    return vs['compound']
#Funcion para realizar la clasificación
def vader_analysis(compound):
    if compound <= -0.5:
        return 'Extremely Negative'
    elif compound > -0.5 and compound < 0:
        return 'Negative'
    elif compound == 0:
        return 'Neutral'
    elif compound > 0 and compound < 0.5:
        return 'Positive'
    else:
        return 'Extremely Positive'
```

Figure 3: Parte del Dataset

Como podemos ver en la figura 3, la primer función es la que obtiene la polaridad de los textos, mientras que la segunda realiza la nueva clasificación, se tomaron los mismo valores de clasificación para realizar una mejor comparación.

3.3 SentiWordNet

Esta ultima realmente es parte de la librería "nltk", esta nos da acceso a una gran cantidad de recursos léxicos que son utilizados para calcular el sentimiento del texto, al igual que la librería anterior esta obtiene las probabilidades de que una palabra sea positiva, negativa o neutral y a partir de esta se obtiene una polaridad, dicha polaridad. El código que se realizó para esta librería fue el siguiente:

```
nltk.download('sentiwordnet')
from nltk.corpus import sentiwordnet as swn
def sentiwordnetanalysis(pos_data):
    sentiment = 0
    tokens_count = 0
    for word, pos in pos_data:
        if not pos:
            continue
        lemma = wordnet_lemmatizer.lemmatize(word, pos=pos)
        if not lemma:
            continue
        synsets = wordnet.synsets(lemma, pos=pos)
        if not synsets:
            continue
        # Take the first sense, the most common
        synset = synsets[0]
        swn_synset = swn.senti_synset(synset.name())
        sentiment += swn_synset.pos_score() - swn_synset.neg_score()
        tokens_count += 1
        # print(swn_synset.pos_score(),swn_synset.neg_score(),swn_synset.obj_score())
    if not tokens_count:
        return 'Neutral'
    if sentiment <= -0.5:
        return 'Extremely Negative'
    elif sentiment > -0.5 and sentiment < 0:
        return 'Negative'
    elif sentiment == 0:
        return 'Neutral'
    elif sentiment > 0 and sentiment < 0.5:
        return 'Positive'
    else:
        return 'Extremely Positive'
```

Figure 4: Parte del Dataset

Como podemos ver en la figura 4, en esta ocasión se realizó solo una función en la cual se obtienen las polaridades de los textos y se clasifican, dado que las polaridades tienen el mismo rango que en las librerías anteriores, se utilizo el mismo rango de clasificaciones.

4 Análisis y resultados

Bien una vez que vimos que se hizo con cada una de las librerías es momento de ver los resultados obtenidos. En la siguiente tabla podremos ver para cada una de las librerías y las etiquetas originales la cantidad de tweets asignados a cada una de las clases: Como podemos ver en la tabla de la figura 5, se obtuvieron resultados muy diferentes en el caso de la clase “Extremely Positive” en los datos originales contábamos con 599 tweets en esta clase pero cuando utilizamos la librería TextBlob estos se redujeron drásticamente a 197, mientras que con las librerías Vader y SWN se aumentaron llegando a 834 y 1107 respectivamente, algo parecido sucede en la clase “Extremely Negative”, por otro lado en el resto

	Extremely Positive	Positive	Neutral	Negative	Extremely Negative
Original	599	947	619	1041	592
Text Blob	197	1509	947	1036	109
Vader	834	847	584	853	680
SWN	1107	799	553	590	749

Figure 5: Tabla de resultados obtenidos

delas clases ocurre lo contrario a esto, con la librería Text Blob se aumentan las cantidades de tweets en comparación con la original, mientras que con Vader y SWN se reducen. En las siguientes graficas de pastel podemos apreciar de mejor manera la forma en que se clasificaron los tweets en cada uno de los casos:

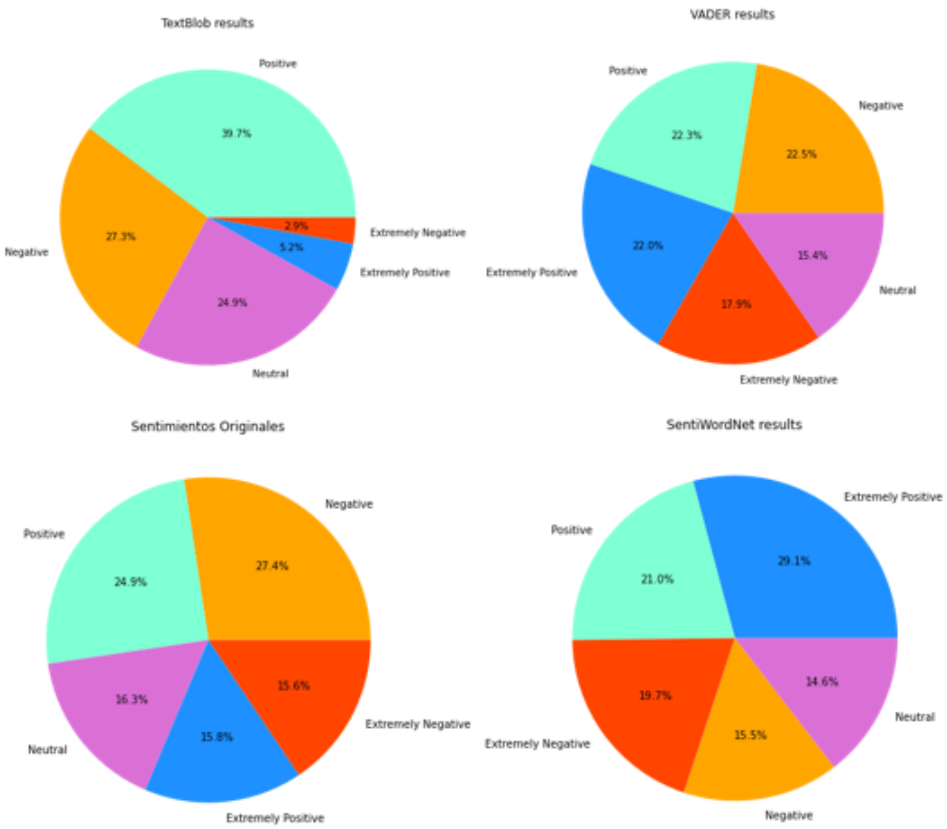

Figure 6: Gráficas de pastel

En la clasificación original se tenía que la mayoría de los textos pertenecían a las clases “Positive” y “Negative” con el 24.9% y 27.4% respectivamente, al realizar una nueva clasificación con las librerías Vader y TextBlob estas se mantuvieron como las clases con mayor porcentaje, con TextBlob los positivos aumentaron hasta el 39.7% mientras que los negativos se mantuvieron casi igual con un 27.3%, en el caso de Vader los positivos obtuvieron un 22.3% y los negativos un 22.5%. Cuando se utilizó la librería SentiWordNet la mayoría de los tweets se clasificaron como “Extremely Positive” y “Positive”. Parece que la clasificación hecha por la librería Vader es la que más se asemeja a la original, la librería TextBlob reduce en gran medida el porcentaje de tweets clasificados como “Extremely Positive” y “Extremely Negative”, mientras que la librería SentiWordNet parece haber hecho una clasificación muy distinta a la original. En la siguiente grafica de frecuencias se puede observar de mejor manera las diferencia y similitudes entre los resultados obtenidos en cada librería y las etiquetas originales: En la grafica de la figura 7, notamos como las líneas que representan a la librería

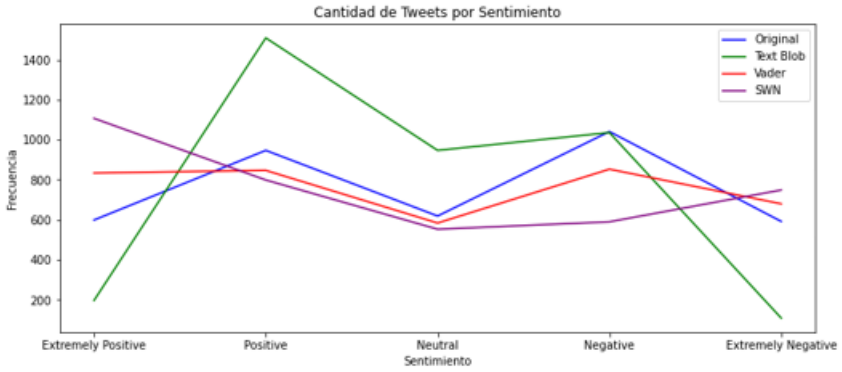

Figure 7: Gráfica de Frecuencias

Vader y TextBlob tienen formas parecidas a la de los datos originales mientras que la línea de la librería SentoWordNet tiene una forma muy diferente, la línea de la librería Vader es la que se asemeja más a la línea de las etiquetas originales.

Ahora analizaremos como se clasificaron cada uno de los tweets según su clasificación original, empezaremos con los tweets que originalmente estaba clasificados como “Extremely Positive”, se realizo un filtrado de la base final y se obtuvieron las siguiente graficas de pastel para ver cómo se clasificaron estos tweets con las otra librerías:

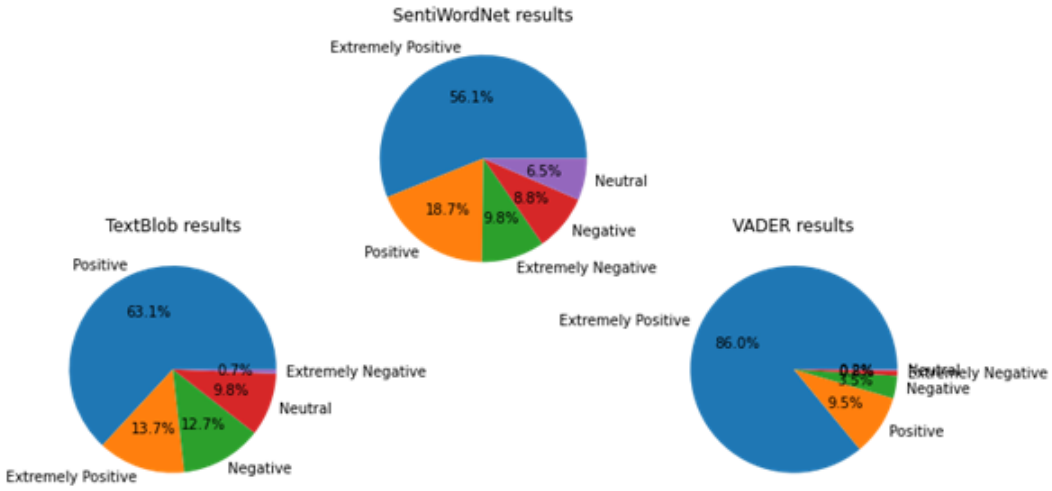

Figure 8: Gráficos de pastel para "Extremely Positive"

Como podemos ver en las graficas de la figura 8, con las librerías SentoWordNet y Vader la mayoría de estos tweets se clasificaron de la misma manera, siendo Vader el que obtuvo el porcentaje más alto, por otro lado, en el caso de la librería TextBlob la mayoría de estos tweets se clasificaron como “Positive”, tan solo el 13.7% fue clasificado igual a la original.
Ahora en el caso de los tweets clasificados originalmente como “Positive”, se obtuvieron los siguientes resultados:

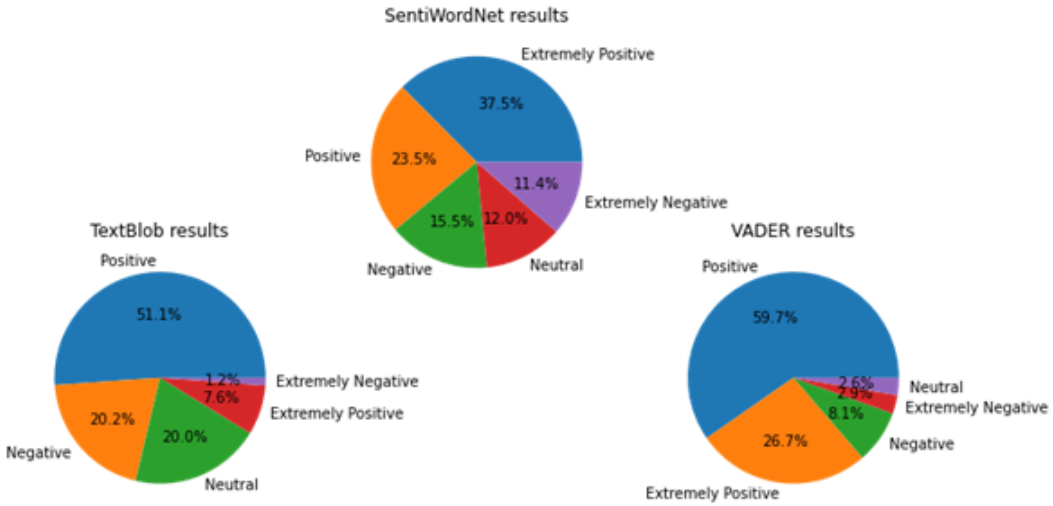

Figure 9: Gráficos de pastel para "Positive"

Como podemos ver en las graficas de la figura 9, las librerias TextBlob y Vader clasificaron a la mayoría de la misma manera, mientras que la libreria SentoWordNet tan solo clasifico al 23.5% como "Positive" se podria decir que cambio toda esta clase.
Ahora análizaremos a los tweets clasificados originalmente como "Neutral":

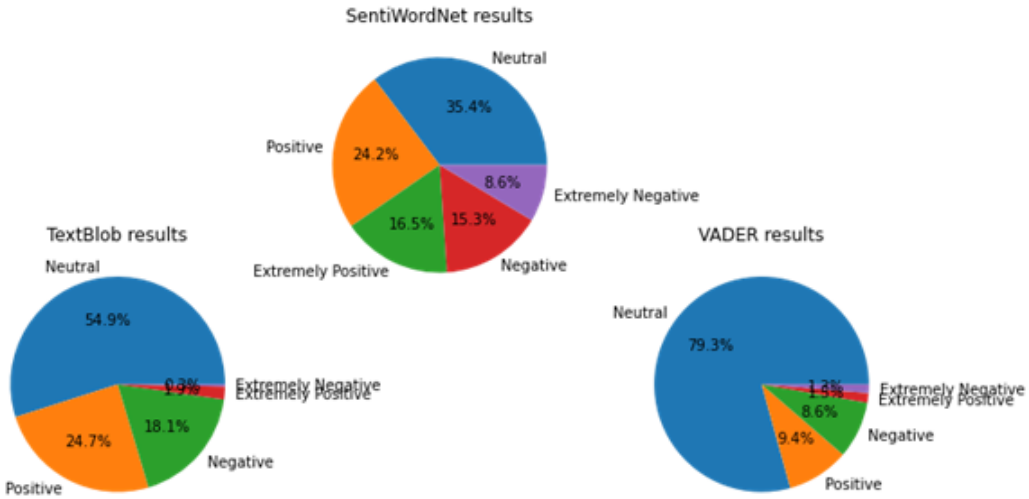

Figure 10: Gráficos de pastel para "Positive"

Podemos ver que una vez más las libreria TextBlob y Vader clasificaron a la mayoría de la misma manera y la libreria SentiWordNet reclasifico a la mayoría de estos tweets como "Positive", "Extremely Positive" y "Negative", tan solo al 35.4% los clasifico igual.

Ahora análizaremos a los tweets clasificados originalmente como "Negative":

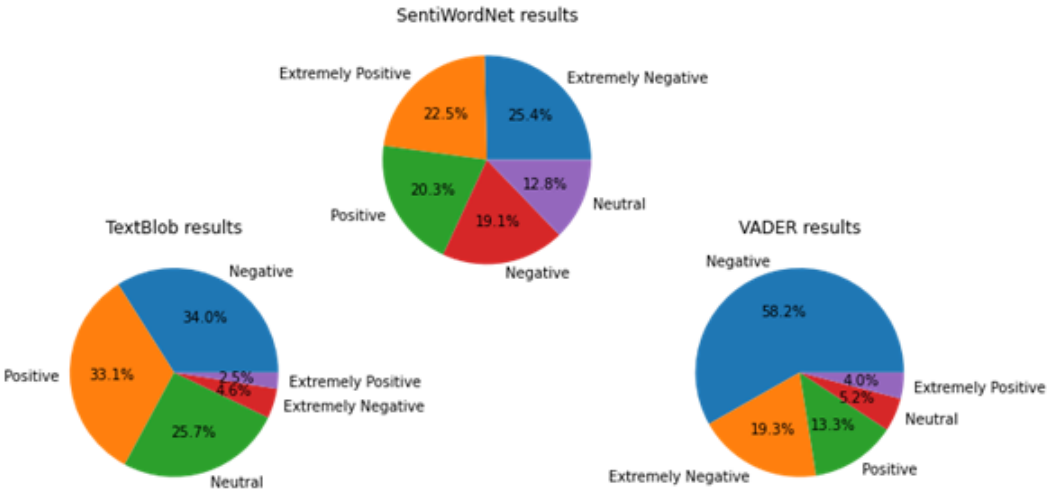

Figure 11: Gráficos de pastel para "Negative"

En este caso solo la librería Vader clasificó a la mayoría de los tweet igual, SentiWordNet reclasifico a la mayotiá de estos tweet como "Extremely Negative", "Extremely Positive" y "Positive", por su parte TextBlob reclasifico a la mayoria como "Neutral" y "Positive".
Ahora análizaremos a los tweets clasificados originalmente como "Extremely Negative":

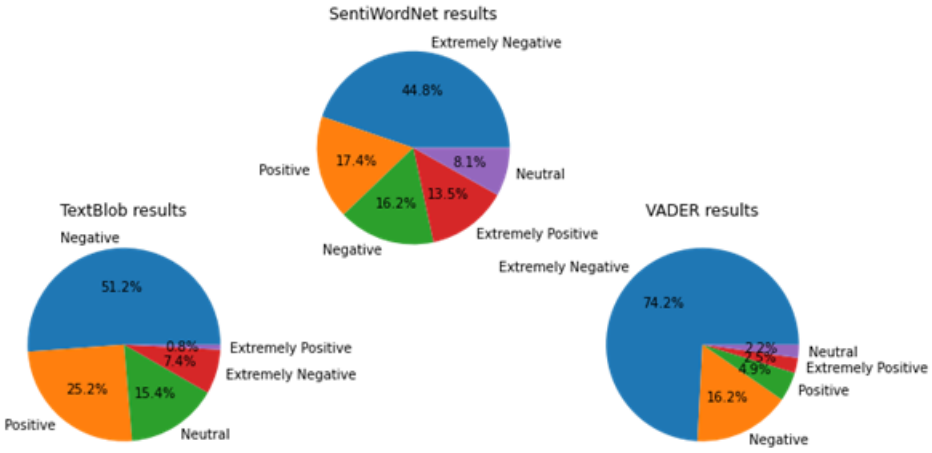

Figure 12: Gráficos de pastel para "Extremely Negative"

Por último, en este caso, la libreria Vader una vez más clasifica a la mayoría de la misma manera, mientras que la librería TextBlob reclasificó a estos tweets comó "Negative" y "Positive", por su parte la libreria SentiWordNet clasificó una gran parte de la misma manera pero una gran parte los reclasificó comó "Positive" y "Negative".

5 Conclusión

Podemos ver que a pesar de que estas tres librerías realicen trabajos parecidos, los resultados que obtuvimos con cada una de ellas fueron muy distintos entre sí, en específico la librería SentiWordNet, ya que está en la mayoría de las clases prácticamente reclasifico la mayoría de las etiquetas originales, por otro lado la librería de VaderSentiment fue la que obtuvo resultados más parecidos a las etiquetas originales, existe la posibilidad de que así fue como estos fueron clasificados obviamente con diferente rangos de clasificación ya que si no hubiéramos obtenido los mismos resultados, por último la librería TextBlob que a mi parecer obtuvo resultados intermedios ente las otras dos librerías, si bien sus resultados tenían cierto parecido a los originales como en el caso de la librería Vader, en el algunos casos si reclasifico muchos tweets como la librería SentiWordNet. Considero que sería muy interesante probar de nuevo el algoritmo de clasificación de texto, pero esta vez utilizando las etiquetas generadas por la librería SentiWordNet dado que esta fue la que realizo mas cambios en las etiquetas originales, también utilizaría las etiquetas generadas por TextBlob y realizaría una comparación de resultados con los resultados que obtuve con las etiquetas originales. En conclusión, con las tres librerías se puede realizar un buen análisis de sentimiento, pero teniendo en cuenta del trabajo previo con esta base de datos y de la sospecha de que las etiquetas podrían no ser las correctas, considero que para esta base de datos las librerías TextBlob y SentiWordNet fueron las que obtuvieron los resultados más interesantes, como futuro trabajo tenemos que entrenar un algoritmo de reconocimiento de texto con estas nuevas etiquetas para analizar mejor los resultados obtenidos.

References

[1] *Link de la base de datos original*
https://www.kaggle.com/datasets/saurabhshahane/ecommerce-text-classification

[2] *Link de Github*
https://github.com/CesarJairTJ/FCFM