Práctica de laboratorio 5: Análisis de Sentimientos.

Castro Elvira D. [2022710039], Pineda Rugerio N. [20222710240], Castañón Hernández V. J. [2022710020], Galicia Cocoletzi N. [2022710234], Sánchez Zanjuampa M. A. [2022710029], y Nava Mendez E. U. [2021710144].

Profesor: Lauro Reyes Cocoletzi

***Resumen*—El análisis de sentimientos es una herramienta importante en el desarrollo del PLN. Los sentimientos y emociones humanas son conceptos subjetivos que son difíciles de entender para una máquina, sin embargo este tipo de contenido es importante para el funcionamiento de algunas aplicaciones del tipo chatbots como Chat GPT o asistentes virtuales como Alexa, Siri o Bixby. Pero en este trabajo se le da un enfoque menos complejo al utilizar la herramienta VADER para el análisis de algunos comentarios extraídos de Twitter para determinar el tipo de sentimiento que este denotando (bueno, malo o neutro) según las palabras que contienen.**

***Palabras clave*—Sentimientos, Emociones, Vader, NLTK**

I. Introducción

I-A. Marco teórico

I-A1. Análisis de textos: El análisis de sentimiento(sentiment analysis) o también conocida como minería de opiniones, es el proceso de analizar grandes volúmenes de texto para determinar si expresa un sentimiento positivo, un sentimiento negativo o un sentimiento neutros. De esta manera se puede obtener información que permita comprender las opiniones de las personas con respecto a un producto, servicio o tema en particular. Para el análisis de sentimientos se emplea métodos del procesamiento del lenguaje natural y machine learning, lo que permite contar con dos enfoques, basado en reglas o ML.

- Análisis de sentimiento basado en reglas.
- Análisis de sentimiento mediante machine learning

Se tiene diferentes maneras de analizar los sentimientos, entre los tres más populares son el análisis de sentimiento basado en la emoción, el de grano fino y el basado en aspectos (ABSA) se basan todos en la capacidad del software subyacente para calibrar algo llamado polaridad, el sentimiento general que transmite un fragmento de texto. Por lo general, la polaridad de un texto se puede describir como positiva, negativa o neutra.

I-A2. Análisis de sentimiento basado en reglas: En el enfoque basado en reglas, el software está entrenado para clasificar ciertas palabras clave en un bloque de texto basado en grupos de palabras, o léxicos, que describen la intención del autor.

I-A3. Análisis de sentimiento mediante machine learning: Con un enfoque de machine learning (ML), se utiliza un algoritmo para entrenar al software para medir el sentimiento en un bloque de texto utilizando palabras que aparecen en el texto, así como el orden en que aparecen.

I-A4. ¿Qué es NLTK Vader?:
Se trata de una herramienta de análisis de sentimientos basada en léxico y reglas específicamente calibrada para los sentimientos más comúnmente expresados. Al calcular un polarity scoreVader se obtienen cuatro métricas: compound, negative, neutraly positive. La compoundpuntuación calcula la suma de todas las calificaciones de léxico que está normalizada entre -1 (most negative)y +1 (most positive). Positive, negativey neutral representa la proporción del texto que cae en estas categorías.

I-A5. ¿Qué es NLTK subjectivity?:
La biblioteca nltk.corpus.subjectivity de NLTK (Natural Language Toolkit) proporciona un conjunto de datos de opiniones subjetivas en inglés. Este conjunto de datos se utiliza para entrenar y evaluar modelos de análisis de sentimientos. ¿Qué contiene? El conjunto de datos contiene dos subconjuntos:

- Subjectivity Gold Standard: Este subconjunto contiene 5,000 oraciones anotadas manualmente con su polaridad subjetiva (positiva, negativa, objetiva).
- Hough's Lexicon: Este subconjunto contiene 23,162 palabras y frases con su puntuación de subjetividad asignada manualmente (positiva, negativa, objetiva).

I-B. Aporte

- Con este trabajo se pretende dar una introducción para entender el análisis de sentimientos de un texto mediante el uso de herramientas como VADER.
- Se utiliza un dataset que contiene comentarios de Twitter, y es posible implementar VADER en las redes sociales en general, además de que debido a la polarización que hace entre *bueno, malo* o *neutro*, se puede utilizar para fines como evaluación de las opiniones de clientes en comercios electrónicos respecto a un producto o mejorar su servicio al cliente.
- En otro ámbitos, esta herramienta se puede escalar para el análisis de temas de mayor interés social como los discursos políticos o noticias.
- Se utilizan diferentes métricas para evaluar el análisis de sentimientos para tener mayor diversidad de información.

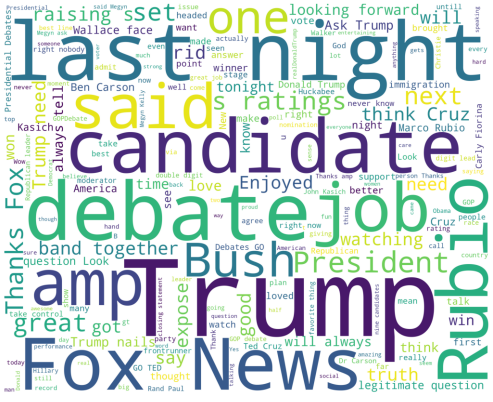
II. Desarrollo

II-A. Código 1. Análisis de sentimientos

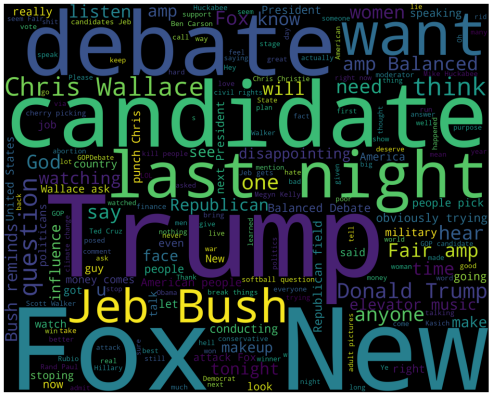
1. **Carga de bibliotecas y recursos** Se descargan los recursos necesarios como bibliotecas, recursos y el dataset.
2. **Preprocesamiento de los datos** Se divide el dataset en dos conjuntos para entrenamiento y prueba. en caso de ser necesario se realiza un preprocesamiento de los datos, por ejemplo retirar signos así como stopwords. De esta manera se procesan a fin de obtener las características necesarias para el entrenamiento.
3. **Entrenamiento del modelo** Con las características obtenidas del paso anterior, se procede a realizar el entrenamiento de un modelo, en caso de ser necesario.
4. **Obtener las predicciones del modelo** Una vez que el modelo ha sido entrenado, entonces se procede a realizar las pruebas correspondientes, mediante métricas como el accuracy, recuperación y F1-score.
5. **Analizar los resultados** Finalmente, se analizan los resultados obtenidos.

II-B. Código 2.1 Análisis de Tweets con Naive Bayes.

1. **Conjunto de datos:** Se cargan los datos del archivo Sentiment.csv, el cual contiene 13871 instancias de tweets clasificados como "positivos", "negativos.º "neutrales".
2. **Preprocesamiento de datos:** El preprocesamiento de datos consiste en eliminar las filas con sentimiento "Neutral". Quedando únicamente los tweets negativos y positivos.

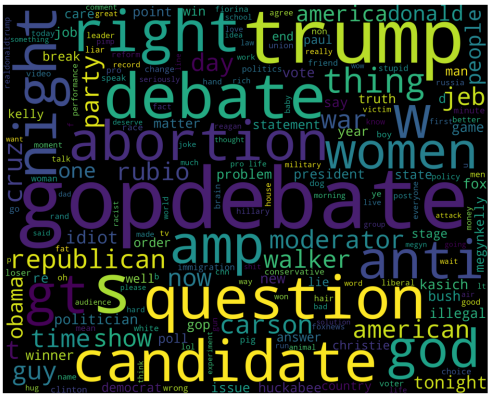


Nube de palabras positivas.



Nube de palabras negativas.

3. **División de datos:** Se dividen los datos en conjuntos de entrenamiento y prueba, en una proporción 90-10. Para optimizar el rendimiento del modelo se probó con una división de los datos 80-20 y 70-30.
4. **Visualización de palabras clave:** Se definen funciones para generar nubes de palabras (word clouds) que representan las palabras más frecuentes en los conjuntos de texto positivos y negativos.



Nube de Palabras en el conjunto de entrenamiento.

5. **Preparación de datos:** Se procesan los datos de entrenamiento, se eliminan stopwords y otras palabras no relevantes como aquellas que tienen menos de 3 caracteres, enlaces (http), menciones de usuario (), hashtags (#), y retweets (RT). Por último, se crean tuplas con el texto limpio del tweet, y con la etiqueta del sentimiento que le corresponde.
6. **Extracción de características:** Consiste en contar la presencia o ausencia de palabras en un tweet. Sin embargo, también se prueban con otra técnica, para evaluar el rendimiento del modelo, TF-IDF.

Algorithm 1 Extraer características: Conteo de palabras.

```
function EXTRACT_FEATURES(document)
    document_words ← set(document)
    features ← {}
    for each word in w_features do
        features['contains(%s)' %word] ← (word ∈ document_words)
    end for
    return features
end function
```

Algorithm 2 Extraer características: TF-IDF

```
function EXTRACT_FEATURES_TFIDF(tweet)
    text, sentiment ← tweet
    text ← join(text)
    features ← tfidf_vectorizer.fit_transform([text])
    features_list ← features.toarray()[0]
    return {'tfidf' : features_list,' sentiment' : sentiment}
end function
```

7. **Entrenamiento del modelo:** Se utiliza un clasificador Naive Bayes, utilizando las características extraídas, dependiendo del enfoque.
8. **Evaluación del modelo.** Se prueba el modelo entrenado con los datos de prueba y se cuentan el número de predicciones correctas para cada clase (positivo y negativo).
9. **Resultados.** Se imprimen los resultados de la evaluación del modelo, mostrando el número de predicciones correctas para las clases positivas y negativas; de igual manera, se muestra el accuracy del modelo.

II-C. Código 2.2 Análisis de Sentimientos con VADER.

- **Conjunto de datos:** Se utiliza el conjunto de opiniones subjetivas proporcionado por NLTK, que es un corpus llama "subjeivity", que contiene opiniones subjetivas y objetivas de una variedad de temas. Se definen dos conjuntos, uno de opiniones subjetivas y otro para opiniones objetivas, de 100 instancias cada uno.
- **División de datos:** Se divide cada conjunto en 80 instancias para entrenamiento y 20 para pruebas. Y se crea un conjunto de entrenamiento que combina tanto documentos subjetivos como objetivos, y uno de test de la misma forma.
- **Extracción de características:** Se utilizan todas las palabras de los documentos de entrenamiento para extraer características de unigramas que aparecen al menos 4 veces. Para la frecuencia de los unigramas se hace una búsqueda con valores de 1 a 5, para determinar cuál mejora el rendimiento del modelo.

Algorithm 3 Optimización de min_freq

```
min_freq_val ← [1, 2, 3, 4, 5]
for each val in min_freq_val do
    Entrenar y evaluar el clasificador con min_freq = val
end for
```

- **Entrenamiento del modelo:** Se utiliza el clasificador Naive Bayes para entrenar y se evalúa en el conjunto de prueba, calculando la precisión, el F-measure y el recall para las clases de subj y obj.
- **Análisis de sentimientos con VADER:** Se utiliza la herramienta de análisis de sentimientos VADER (Valence Aware Dictionary and sEntiment Reasoner) para analizar una serie de oraciones y párrafos. Se imprime la polaridad de cada oración según VADER, incluyendo su puntuación de negatividad, neutralidad, y positividad.

III. Resultados

III-A. Código 1.1 Análisis de sentimientos.

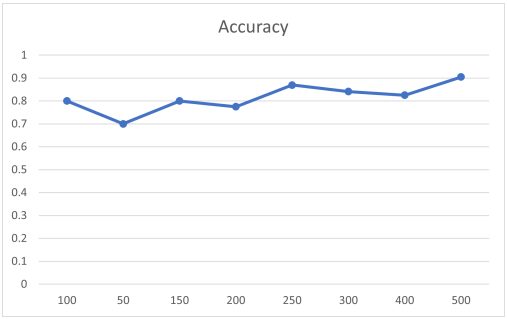
En este primer código se presentaron dos métodos de análisis de sentimientos. El primero consiste en el entrenamiento de un clasificador de Naive Bayes, mientras que el segundo utiliza NLTK Vader. Para el ejemplo del clasificador, se empleó un dataset disponible en NLTK, específicamente en la sección de subjetividad. La división de los datos se realizó considerando si las oraciones eran subjetivas u objetivas. Además, se tuvo en cuenta la negación de las palabras utilizando mark_negation(). De esta manera, se generan características unigramas de las oraciones, teniendo en cuenta la frecuencia de las palabras. Considerando como número mínimo de frecuencias de la palabra como 4, se obtuvieron los siguientes resultados.

Tabla I: **Resultados obtenidos en Accuracy y Fmeasure**

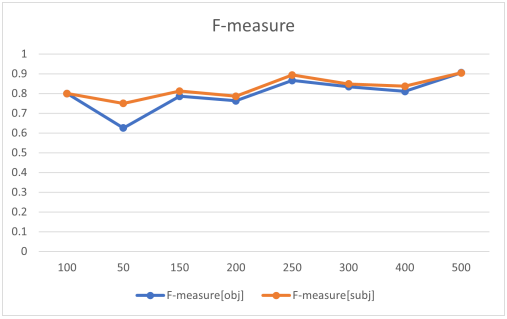
Prueba	n_iteraciones	Accuracy	Fmeasure[obj]	Fmeasure [subj]
1	100	0.8	0.8	0.8
2	50	0.7	0.625	0.75
3	150	0.8	0.7857	0.8125
4	200	0.775	0.7631	0.7857
5	250	0.87	0.8659	0.8936
6	300	0.841	0.8347	0.8480
7	400	0.825	0.8108	0.8372
8	500	0.905	0.9054	0.9045

Tabla II: **Resultados obtenidos en Precisión y Recall**

Prueba	Precision [obj]	Precision [subj]	Recall [obj]	Recall [subj]
1	0.8	0.8	0.8	0.8
2	0.833	0.6428	0.5	0.9
3	0.8461	0.7647	0.733	0.866
4	0.805	0.75	0.725	0.825
5	0.8936	0.8490	0.84	0.9
6	0.8727	0.8153	0.8	0.883
7	0.8823	0.7826	0.75	0.9
8	0.900	0.909	0.91	0.9



Gráfica de Accuracy según número de iteraciones.



Gráfica de F-measure según número de iteraciones.

Se probó para valores de min_freq = [1,2,3,4,5], basándonos en el n_iteraciones de 100. El resultado por cada valor es el siguiente:

Tabla III: **Resultados obtenidos en Accuracy, y Fmeasure**

Prueba	min_freq	Accuracy	Fmeasure[obj]	Fmeasure [subj]
1	1	0.85	0.8421	0.8571
2	2	0.85	0.833	0.8636
3	3	0.8	0.7894	0.8095
4	4	0.8	0.8	0.8
5	5	0.8	0.8	0.8

Tabla IV: **Resultados obtenidos en Precision y Recall**

Prueba	Precision [obj]	Precision [subj]	Recall [obj]	Recall [subj]
1	0.888	0.8181	0.8	0.9
2	0.9375	0.7916	0.7916	0.75
3	0.8333	0.7727	0.75	0.85
4	0.8	0.8	0.8	0.8
5	0.8	0.8	0.8	0.8

Gráfica de resultados según el mínimo de frecuencia.

Se tomaron los parámetros de n_iteraciones = 500 y min_frecuncia = 2, se obtuvo el siguiente resultado:

```
Accuracy: 0.915
F-measure [obj]: 0.9154228855721392
F-measure [subj]: 0.9145728643216081
Precision [obj]: 0.9108910891089109
Precision [subj]: 0.9191919191919192
Recall [obj]: 0.92
Recall [subj]: 0.91
```

Como se puede observarlos, los resultados obtenidos son equilibrados, por lo tanto, se puede decir que la clasificación realizada no se inclina hacia una de las dos clases.

III-B. Código 1.2 Análisis de sentimientos.

Haciendo uso de los parámetros:
De instancias: 100

Oraciones simples:

Unbelievably bad acting!!
compound: -0.6572, neg: 0.686
neu: 0.314, pos: 0.0

Poor direction.
compound: -0.4767, neg: 0.756
neu: 0.244, pos: 0.0

VERY poor production.
compound: -0.6281, neg: 0.674
neu: 0.326, pos: 0.0

The movie was bad.
compound: -0.5423, neg: 0.538
neu: 0.462, pos: 0.0

Very bad movie.
compound: -0.6732, neg: 0.694
neu: 0.306, pos: 0.0

Oraciones complicadas:

I like to hate Michael Bay films, but I couldn't fault this one
compound: 0.3153, neg: 0.157
neu: 0.534, pos: 0.309

It's one thing to watch an Uwe Boll film, but another thing entirely to pay for it
compound: -0.2541, neg: 0.112
neu: 0.888, pos: 0.0

It was one of the worst movies I've seen, despite good reviews.
compound: -0.7584, neg: 0.394
neu: 0.606, pos: 0.0

The script is not fantastic, but the acting is decent and the cinematography is EXCELLENT!
compound: 0.7565, neg: 0.092
neu: 0.607, pos: 0.301

Roger Dodger is one of the most compelling variations on this theme.
compound: 0.2944, neg: 0.0
neu: 0.834, pos: 0.166

Mediante SentimentIntensityAnalyzer().polarity_scores() se puede obtener el valor de cada palabra con respecto a si esta tiende a ser positiva o negativa. De esta manera permite obtener las tendencias de las oraciones que se presentan dentro del dataset, y, por lo tanto, permite obtener el sentimiento que transmite el autor.

III-C. Código 1.3. Análisis de Sentimientos Optimización.

III-C1. Optimización de min_freq.: Se probó para valores de min_freq = [1,2,3,4,5]. El resultado por cada valor es el siguiente:

1. Min_freq = 1.

```
Accuracy: 0.85
F-measure [obj]: 0.8421052631578947
F-measure [subj]: 0.8571428571428572
Precision [obj]: 0.8888888888888888
Precision [subj]: 0.8181818181818182
Recall [obj]: 0.8
Recall [subj]: 0.9
```

2. Min_freq = 2.

```
Accuracy: 0.85
F-measure [obj]: 0.8333333333333334
F-measure [subj]: 0.8636363636363636
```

```
Precision [obj]: 0.9375
Precision [subj]: 0.7916666666666666
Recall [obj]: 0.75
Recall [subj]: 0.95
```

3. Min_freq = 3.

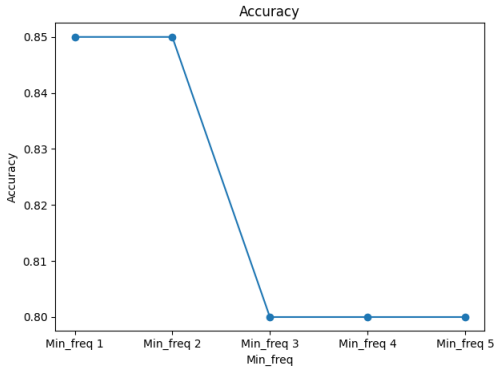
```
Accuracy: 0.8
F-measure [obj]: 0.7894736842105263
F-measure [subj]: 0.8095238095238095
Precision [obj]: 0.8333333333333334
Precision [subj]: 0.7727272727272727
Recall [obj]: 0.75
Recall [subj]: 0.85
```

4. Min_freq = 4.

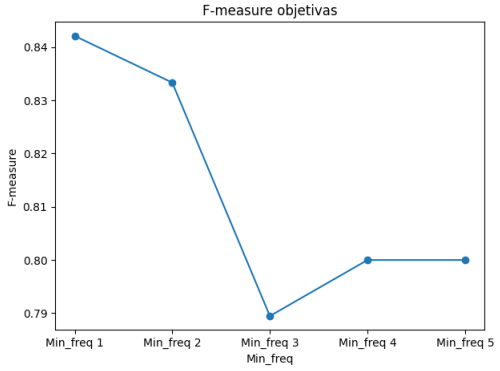
```
Accuracy: 0.8
F-measure [obj]: 0.8
F-measure [subj]: 0.8
Precision [obj]: 0.8
Precision [subj]: 0.8
Recall [obj]: 0.8
Recall [subj]: 0.8
```

5. Min_freq = 5.

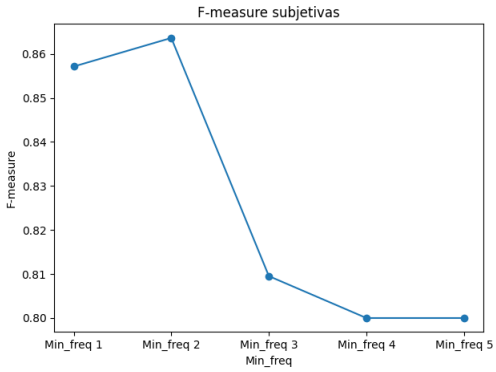
```
Accuracy: 0.8
F-measure [obj]: 0.8
F-measure [subj]: 0.8
Precision [obj]: 0.8
Precision [subj]: 0.8
Recall [obj]: 0.8
Recall [subj]: 0.8
```



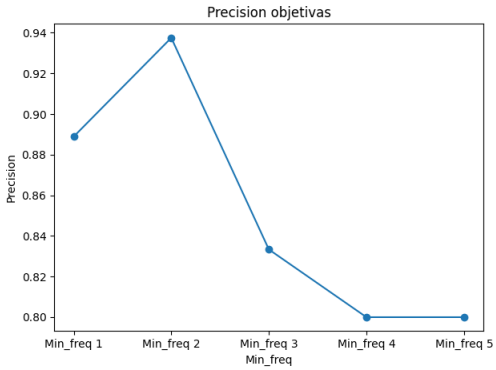
Gráfica de accuracy según el valor de min_freq.



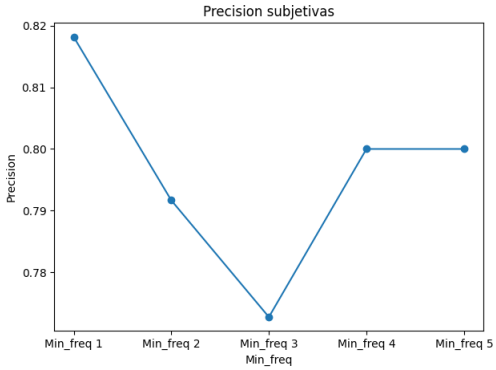
Gráfica de F_measure para opiniones objetivas según el valor de min_freq.



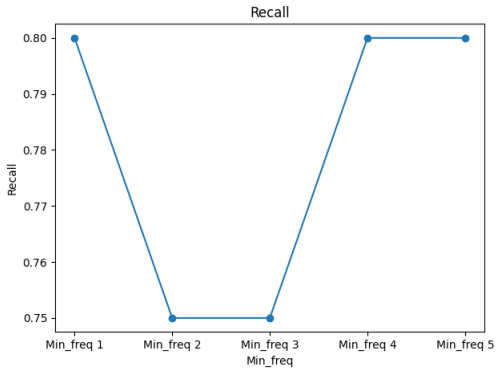
Gráfica de F_measure para opiniones subjetivas según el valor de min_freq.



Gráfica de Presición para opiniones objetivas según el valor de min_freq.



Gráfica de Presición para opiniones subjetivas según el valor de min_freq.



Gráfica de Recall según el valor de min_freq.

Al final los mejores resultados fueron con un min_freq = 1.

Asimismo, se probó para un $numerodeinstancias = [100, 500, 1000, 2000]$. Con un máximo de 2000 ya que solo existen 5000 instancias. Los resultados de las diferentes métricas de evaluación fueron los siguientes:

- Para 100 instancias.

```
Accuracy: 0.85
F-measure [obj]: 0.8421052631578947
F-measure [subj]: 0.8571428571428572
Precision [obj]: 0.8888888888888888
Precision [subj]: 0.8181818181818182
Recall [obj]: 0.8
Recall [subj]: 0.9
```

- Para 500 instancias.

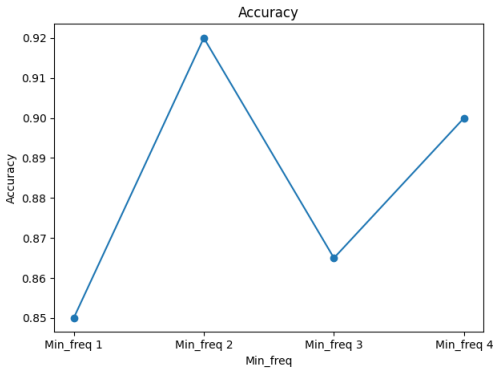
```
Accuracy: 0.92
F-measure [obj]: 0.92
F-measure [subj]: 0.92
Precision [obj]: 0.92
Precision [subj]: 0.92
Recall [obj]: 0.92
Recall [subj]: 0.92
```

- Para 1000 instancias.

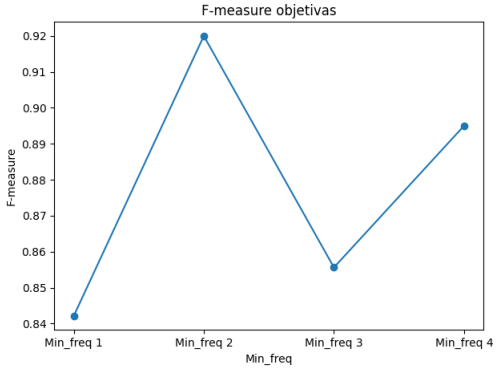
```
Accuracy: 0.865
F-measure [obj]: 0.8556149732620321
F-measure [subj]: 0.8732394366197184
Precision [obj]: 0.9195402298850575
Precision [subj]: 0.8230088495575221
Recall [obj]: 0.8
Recall [subj]: 0.93
```

- Para 2000 instancias.

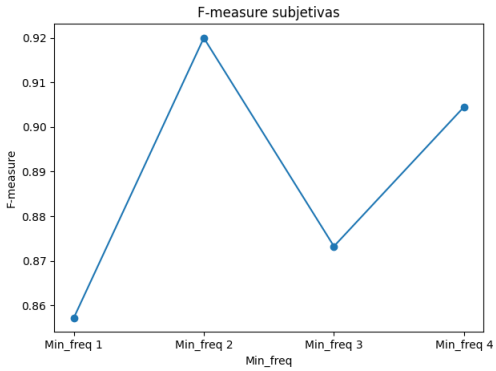
```
Accuracy: 0.9
F-measure [obj]: 0.8950131233595802
F-measure [subj]: 0.9045346062052505
Precision [obj]: 0.9419889502762431
Precision [subj]: 0.865296803652968
Recall [obj]: 0.8525
Recall [subj]: 0.9475
```



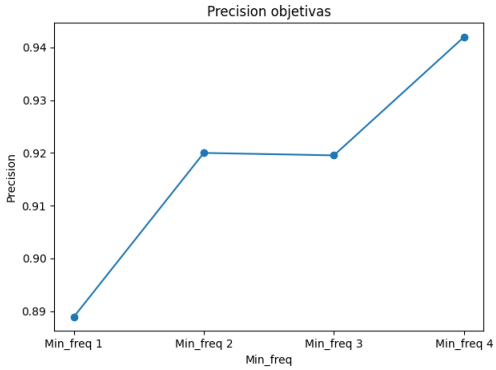
Gráfica de accuracy según el número de instancia.



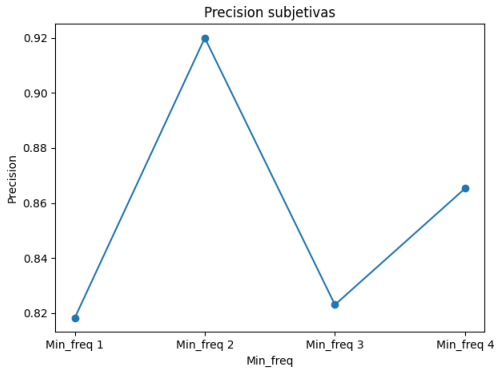
Gráfica de F_measure para opiniones objetivas según el número de instancia.



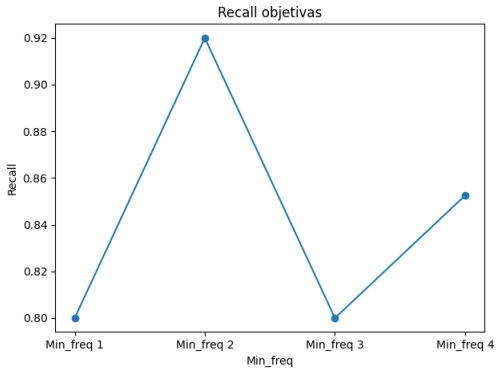
Gráfica de F_measure para opiniones subjetivas según el número de instancia.



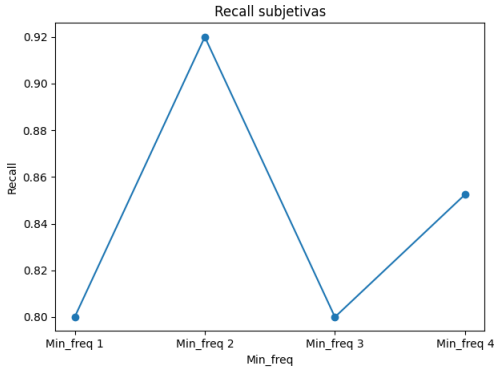
Gráfica de Precision para opiniones objetivas según el número de instancia.



Gráfica de Precision para opiniones subjetivas según el número de instancia.



Gráfica de Recall para opiniones objetivas según el número de instancia.

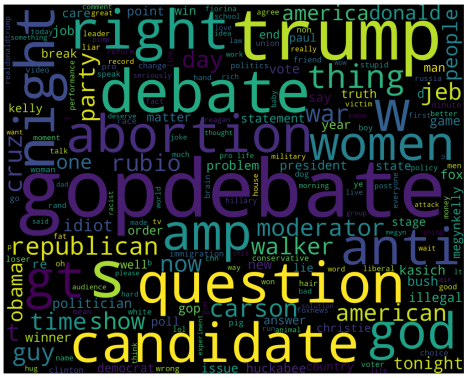


Gráfica de Recall para opiniones subjetivas según el número de instancia.

Como se puede observar gráficamente el pico se encuentra en la mayoría de métricas cuando se tienen 500 instancias.

III-D. Código 2. Análisis de Sentimientos.

III-D1. Utilizando la frecuencia de aparición de palabras.: Primero se muestra la nube de palabras generada con el set de entrenamiento para este caso:



Prueba 1. Nube de palabras en el set de entrenamiento.

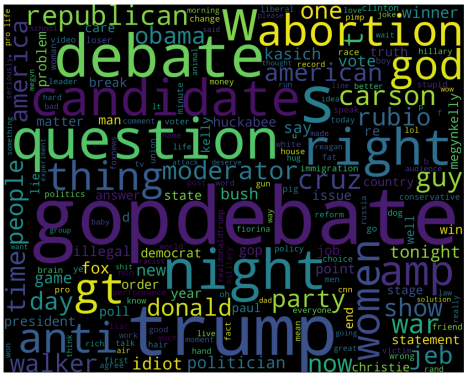
Se obtiene un vector de características de 14720 valores, esto es ya que utiliza cada palabra que aparece en el texto. Se obtuvo el siguiente número de palabras por cada polaridad:

[Negative]: 859/811
[Positive]: 223/88

Y se obtuvo un accuracy de:

Accuracy: 0.6577809798270894

III-D2. Utilizando TF-IDF.: Primero se muestra la nube de palabras generada con el set de entrenamiento para este caso:



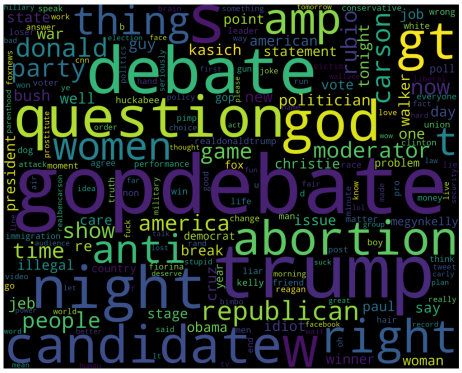
Prueba 2. Nube de palabras en el set de entrenamiento.

Se obtiene un vector de características de 11 valores, muchos menos que la extracción de características original, lo que reduce el tiempo de entrenamiento y validación del modelo. Y se obtuvo un accuracy de:

Accuracy: 0.7917867435158501

Después, se probó con distintas configuraciones para la división de datos de entrenamiento y prueba.

1. Para una división 90-10, el accuracy del modelo fue de: 0.79.
2. Para una división 80-20, se realizó el mismo procedimiento. A continuación se muestra la nube de palabras.

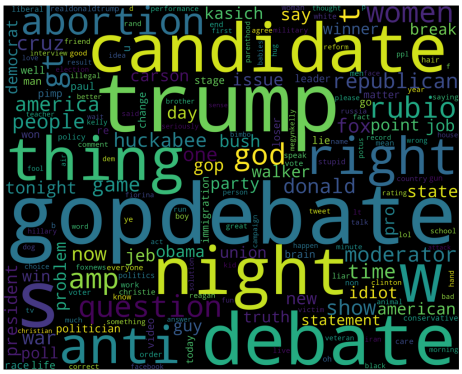


Nube de palabras para una división de datos 80-20.

Y se obtuvo un accuracy de:

Accuracy: 0.7809009009009009

3. Para una división 80-20, se realizó el mismo procedimiento. A continuación se muestra la nube de palabras.



Nube de palabras para una división de datos 70-30.

Accuracy: 0.7820759250360404

Por lo anterior, se concluye que la mejor configuración es con una división 90-10.

IV. Conclusiones

El análisis de sentimientos y la clasificación de textos son herramientas esenciales en el campo del procesamiento del lenguaje natural ya que nos ayudan a comprender las emociones humanas en torno a alguna tarea, sitio o actividad. Los resultados obtenidos en esta práctica proporcionaron una visión general de cómo los algoritmos interpretan el texto.

Se observó que las oraciones simples tienden a tener una fuerte negatividad, mientras que las oraciones más largas y complejas suelen ser neutrales. Esto puede deberse a la presencia de múltiples ideas o sentimientos en una sola oración, lo que puede disminuir el impacto de las palabras individuales.

Al comparar los algoritmos de Naive Bayes y NLTK Vader en la clasificación de textos, se encontró un equilibrio en los resultados, lo que indica que los algoritmos fueron capaces de manejar bien diferentes tipos de textos y contextos. Sin embargo, es importante tener en cuenta las limitaciones de estos algoritmos y seguir explorando formas de mejorarlos.

Finalmente, los ajustes en los parámetros del análisis demostraron que la frecuencia de las palabras y el número de instancias pueden tener un impacto significativo en los resultados. En particular, reducir la frecuencia de las palabras a 1 y aumentar el número de instancias a 500 mejoró la precisión del análisis.

Aunque estos algoritmos tienen sus limitaciones, también ofrecen oportunidades significativas para entender y analizar el texto, con futuras mejoras en estos algoritmos, podemos esperar obtener mejores resultados.

Referencias

[1] ¿Qué son las RNN? -AWS https://aws.amazon.com/es/what-is/recurrent-neural-network/#:~:text=Una%20red%20neuronal%20recurrente%20(RNN,salida%20de%20datos%20secuencial%20espec%C3%ADfica.. Recuperado el 12 de abril de 2024