Análisis de sentimientos en Twitter sobre COVID-19*

Juma Wendy[1[0]], Quisupangui Eduardo[1[1]], Morales Bryan[1[2]], Ramirez Kevin[1[3]], and Velasco Bryan[1[4]]

UNIVERSIDAD POLITÉCNICA SALESIANA
{UPS} https://www.ups.edu.ec/

Abstract. Twitter se ha convertido en una de las redes sociales más utilizadas para expresarse abiertamente con la cual se puede manifestar y compartir opiniones e ideas, esta plataforma social, es un servicio de comunicación bidireccional con el que puedes compartir información de diverso tipo de una forma rápida y sencilla. Por esta razón resulta una fuente ideal donde extraer información sobre estadísticas sociales, con la cual se pretende analizar esta información centrándose en el estudio de la polaridad. El Análisis de sentimientos en Twitter busca establecer la subjetividad de las opiniones expresadas sobre esta red social.

Keywords: tweets · Análisis de sentimiento · COVID · machine learning · regresión logística · procesamiento de datos · diccionario · distancia de jaccard · matriz de cosenos · Twitter.

1 Introducción

Twitter es una red social en la que los usuarios pueden publicar y compartir contenido de actualidad, como noticias o eventos en tiempo real, para este caso de estudio se utilizará los tweets que contenga temas relacionados al COVID-19 en Ecuador. El Análisis de Sentimientos es una tarea incluida en el ámbito del NLP (Natural Language Processing), con lo cuál, el analisís de sentimiento busca extraer opiniones y la polaridad de estas de uno o más documentos mediante la extracción de una serie de características que determinen cuán positivo o negativo es el texto.

Descripción del problema Analizar los sentimientos de nuevos tweets, basados en el corpus del tweet y su similitud con el diccionario de sentimientos. Utilizar dos métricas de similitud y etiquetar a los tweets entre positivos y negativos. El sistema debe ser capaz de tomar como entrada un dataset externo o consumir on-line una nueva consulta. Calcular el porcentaje de positivos y negativos.

* Supported by organization UPS.

Explicación del contexto
En el caso de este proyecto se busca encontrar la opinión de un conjunto de usuarios de Twitter acerca de un tema determinado, desarrollando un sistema que analice sentimientos de comentarios de Twiiter con base en Covid-19, para Ecuador.

Estado del arte

Análisis de sentimientos utilizando las métricas de Jaccard y Coseno

Jaccard es una métrica para puntuar dos conjuntos de datos a diferencia del solapamiento esta normaliza los documentos haciendo que no importe el tamaño para tener una puntuación más alta para su utilización, es necesario tokenizar el conjunto de datos que se quiera puntuar además de que los elementos solo pueden estar una vez en el conjunto. La forma de obtener la métrica es la siguiente, se toma los elementos que intersecan en los conjuntos y para no dar un peso mas grande a los conjuntos o documentos grandes se divide para la unión la fórmula es la siguiente:

$$J(A,B)=\frac{|A\cap B|}{|A\cup B|}$$

Fig. 1. Similutud Jaccard.

Implementación
Para analizar el sentimiento de los tweets se utilizo un diccionario de palabras positivas y negativas con el fin de comparar con las palabras del tweet, si este presentaba mas palabras positivas que negativas seria catalogado como un tweet positivo, así mismo si este presentaba mas palabras negativas que positivas seria catalogado como un tweet negativo, pero podría darse el caso de que tenga la misma cantidad de palabras será catalogado como neutro.
Tanto para Jaccard como para coseno se realizo procesamiento de lenguaje natural (NLTK).
Se elimino los términos frecuentes StopWords ya que estas nos dan más semántica en los tweets que sentimientos, además se realizó Stemming para poder dejar a las palabras en sus raíces lo cual garantizaba que la diferentes interpretaciones de una palabra queden como una sola por ejemplo (fallecer, falleció, fallecieron se reduce a fallec) por lo que nuestro diccionario de sentimientos no tendrá que

contener todos las posibles interpretaciones aumentando así la posibilidad de encontrar es sentimiento correcto.

Frecuencia del termino TF Y su pesado
Esto se obtiene mediante el conteo de veces que una palabra de nuestro diccionario está dentro de un tweet (número de veces que una palabra ocurre en el tweet). Por lo que una palabra que aparezca mas en un documento es más relevante pero no n veces más que este aparezca. La relevancia no se incrementa proporcionalmente con su frecuencia ya que con eso se dará más importancia a los documentos más grandes.
La puntuación de obtiene de la siguiente:
En el caso de que el TF sea 0 la puntuación será 0

$$w_{t,d} = 1 + \log_{10} \mathrm{tf}_{t,d} \quad \text{si } \mathrm{tf}_{t,d} > 0$$

Fig. 2. Pesado TF

Frecuencia del documento
Los términos poco comunes son mas informativos que los términos frecuentes lo mas deseable es asignar un peso alto para los términos raros. La diferencia entre los términos frecuentes y raros es que los frecuentes son menos informativos que los raros.
Para calcular el IDF se pasara el termino DF

$$\mathrm{idf}_t = \log_{10}(N/\mathrm{df}_t)$$

Fig. 3. Calculo IDF

N: es el número de documentos o tweets con los que estamos trabajando
Pesado TF-IDF
Es el valor final que representara la distancia entre la consulta y documento

$$w_{t,d} = tf_{t,d} \times idf_t = (1 + \log(tf_{t,d})) \times \log(N / df_t)$$

Fig. 4. Distancia TF-IDF

Similitud de coseno
Es el encargado de desarrollar el cálculo de una medida de similitud entre dos vectores distintos de cero dentro de un espacio interno, donde se mide el coseno del ángulo entre ellos

$$\cos(\vec{q},\vec{d}) = \vec{q} \bullet \vec{d} = \sum_{i=1}^{|V|} q_i d_i$$

Fig. 5. Similitud Coseno

1.1 Modelamiento del problema

Dado que es una tarea de aprendizaje supervisado, se nos proporciona un conjunto de datos de capacitación que consiste en Tweets etiquetados con "1" o "2" y un conjunto de datos de pruebas.

- etiqueta "1": sentimiento positivo
- etiqueta "2": sentimiento negativo

Ya que el conjunto de datos es un modelo de clasificación binaria re-catalogamos los sentimientos negativos como "0".

Construcción del modelo de predicción

Modelo de Regresión Logística
La regresión logística es un método útil para resolver problemas de clasificación binaria ya que puede resolver problemas de predicción, clasificación. La clasificación multinomial, categoría de clasificación, maneja problemas en los que se manejan múltiples clases presentes en una variable objetivo.

Carga de datos
Cargamos el conjunto de datos usando la función CSV de lectura de la librería de pandas, el conjunto de datos posee los tweets que incluyen "#Covid-19" y etiquetados como positivos "1" y negativos "0".

```
# Carga y reemplazo del dataset con los tweets etiquetados en Positivo = "1" y Negativos = "2"
import pandas as pd
from sklearn.model_selection import train_test_split
columnas = ['text','user','target']
data = pd.read_csv('lista-posi-nega-nueva3.csv',sep=';', encoding = "utf8", names=columnas)
# Reemplazo de etiquetado Negativo a "0"
data = data.replace(2, 0)
```

Fig. 6. Carga de conjunto de datos CSV y reemplazo de etiqueta.

División de datos
Para comprender el rendimiento del modelo una buena estrategia es dividir el conjunto de datos en training y en test usando la función train_test_split() proveniente de sklearn, las características de 3 parámetros, objetivo, tamaño de test_set, además usa random_set para seleccionar registros al azar. El conjunto de datos se divide en dos partes en una proporción de 70:30, lo que significa 70% de los datos se utilizaran para el entrenamiento o capacitación del modelo y el 30% para las pruebas de modelos.

```
# Division del dataset en Train & Test
x_train,x_test,y_train,y_test = train_test_split(x1,y1,test_size=0.3)
```

Fig. 7. División de dataframe en Train y Test usando la función $\text{train}_{t}est_{s}plit()$.

Desarrollo de modelo y predicción
Importamos los modelos de Regresión Logística y creamos un objeto utilizando la función LogisticRegression() proveniente de sklearn. Luego, se ajusta el modelo en el conjunto de train usando fit() y realice la predicción en el conjunto de prueba usando predict().

Procesamiento de Tweets
Uno de los procesos esenciales en las tareas de NPL es el procesamiento de datos.

- Lower: convertir las letras a minúsculas.
- Tokenizing: convertir los tweets en tokens. Las fichas son palabras separadas por espacios en un texto.
- Stopword: algunas palabras no contribuyen mucho al modelo de aprendizaje automático, por lo que es bueno eliminarlas.
- Stemming: eliminación de afijos (circunfijos, sufijos, prefijos, infijos) de una palabra para obtener una raíz de la palabra. Porter Stemmer es la técnica más utilizada porque es muy rápida.
- Vectorización de datos: proceso para convertir tokens en números. Es un paso importante porque el algoritmo de aprendizaje automático funciona con números y no con texto.

Se implementará la vectorización usando tf-idf, también otras técnicas como Bolsa de palabras.

```
def funcion_curacion(palabras):
    tokens = [re.sub(r'[-()\"#/@;:<>{}`+=~|.!?,''0-9,A¡""…¡¡¡»]', ' ', i.lower()).split() for i in palabras]
    stopW = stopwords.words('spanish')
    for h in tokens:
        for m in h:
            if m in stopW:
                h.remove(m)
        for m in h:
            if m in stopW:
                h.remove(m)
        for m in h:
            if m in stopW:
                h.remove(m)

    spanishStemmer = SnowballStemmer("spanish", ignore_stopwords=True)
    stemmer2 = [[spanishStemmer.stem(m) for m in h] for h in tokens]
    # print("\nSteamming:")
    # for i in stemmer2:
    #      print(i)
    return stemmer2
```

Fig. 8. Procesamiento de datos contenidos en los tweets.

Implementación del Modelo de Regresión Logística en Python para Análisis de Sentimientos

$$p(Y) = \frac{e^{\beta_0+\beta_1 x_1+\beta_2 x_2+.....+\beta_i x_i}}{1+e^{\beta_0+\beta_1 x_1+\beta_2 x_2+.....+\beta_i x_i}}$$

Al momento de evaluar la validez y calidad de un modelo de regresión logística múltiple, se analiza tanto el modelo en su conjunto como los predictores que lo conforman.

Ecuación de regresión logistica obtenida del modelo:

$$Y = \frac{e^{[0.65533467]+(-0.088X1)+(-0.271X2)+.....+(0.0X3971)+(0.0X3972)}}{1+e^{[0.65533467]+(-0.088X1)+(-0.271X2)+.....+(0.0X3971)+(0.0X3972)}}$$

Visualizacion de Análisis de Sentimientos Twitter

Una vez entrando el modelo podremos observar cómo clasifica el modelo de análisis de sentimientos. Mientras más datos de entrenamiento etiquete, más preciso será el clasificador.

Para este caso visualizamos como referencia 3 tweets de los 300 que posee el test.

```
# Training Logistics Regression model

model = LogisticRegression(solver='liblinear', random_state=0).fit(x11, y11)

LR_model = LogisticRegression(solver='lbfgs')
LR_model.fit(x11, y11)
y_predict_lr = LR_model.predict(x111)
print(accuracy_score(y111, y_predict_lr))
```

Fig. 9. Modelo de Regresión Logística.

```
Twitter de covid(Training)443 :
ANT entrega duplicados de licencias a los conductores en los domicilios » https://t.co/khcQMYk5LK https://t.co/IYNzhKffv9

Predicción-Y  443: --->1

Dataset-Y  443: --->1


Twitter de covid(Training)848 :
#Covid19 | Tras silencio inicial, Donald Trump reconoce los 100 000 muertos por coronavirus en EE.UU. »… https://t.co/7MTBKe3iJ8

Predicción-Y  848: --->0

Dataset-Y  848: --->0


Twitter de covid(Training)873 :
#EDITORIAL | 'Seguridad, otro flanco complejo en la pandemia'. Compartimos la opinion de EL COMERCIO »… https://t.co/NIeuPNHLEA

Predicción-Y  873: --->0

Dataset-Y  873: --->0
```

Fig. 10. Análisis de Sentimientos del dataframe.

API Twitter
La API de Twitter (interfaz de programación de aplicaciones) permite a los desarrolladores de software acceder e interactuar con datos públicos de Twitter. Los desarrolladores pueden interactuar con esta API escribiendo sus propios scripts o utilizando una de las bibliotecas de código abierto disponibles en diferentes lenguajes de programación.

2 Materiales y Métodos

2.1 Extracción de los tweets mediante la API de twitter:

Importación de las librerías:

```
Import twitter
```

```
Import tweepy
```

Para extraer los tweets debemos de tener una cuenta de desarrollador en twitter que nos generara una API con sus respectivas credenciales.

```
consumer_key = "BUJohQSMFB11NeW9Wi5khkymG"
consumer_secret = "yfihvSGhHoRHSadw2KPPbYgpRx7qqJwlfTdnGIv0rixmnIXoBm"
access_token= "1262539689224962050-3Sp0lWeeeDU683EjQhX0uyBuiwVLWC"
access_token_secret = "yGePV1nDq9w8UKv1DcRbOQB4kHS99eds4dcC0Wr9IuRIV"
```

Fig. 11. Clave de la API de Twitter

Se creó un curso con (api.search) que permite extraer un tema específico de los tweets con diferentes parámetros como (tema, idioma, fechas, localidad, coordenadas geográficas dando los valores de latitud y longitud y el número de tweets que queremos consultar.

```
lista = []
for tweet in tweepy.Cursor(api.search,
                           q="covid-19 -filter:retweets -filter:replies",
                           lang="es",
                           since="2020-06-06",
                           until ="2020-06-09",
                           locale="ECU",
                           geocode="-1.95529,-78.70604,330km",
                           tweet_mode="extended").items(5):

    lista.append(tweet._json["full_text"])
```

Fig. 12. Extración de los tweets

Herramienta textblob

TextBlob es una librería Python (2 y 3) para procesar datos de texto. Proporciona una API (interfaz de programación de aplicaciones) consistente para sumergirse en tareas comunes de procesamiento de lenguaje natural (PNL) tales como etiquetado de parte del habla, extracción de frases sustantivas, análisis de sentimientos.

Métricas de polaridad y subjetividad

Con TextBlob, obtenemos una métrica de polaridad y subjetividad. La polaridad es el sentimiento mismo, que va de un valor (-1) en negativos a (+1) como positivos. La subjetividad es una medida del sentimiento, y va de 0 a 1.

Librería necesaria para textblob:

Se deberá instalar

```
Pip install textblob
```

Se debe importar la librería TextBlob en su proyecto:

```
From textblob import TextBlob
```

Algoritmo

Código para poder realizar la polaridad de los tweets positivos y negativos:

Fig. 13. Polaridad de los tweets

Procesos importantes de textblob:
Se almacenará los tweets obtenidos en la variable analysis.

```
analysis = TextBlob(line)
```

Se realizará la traducción de los tweets almacenados de la variable analysis.

```
eng=analysis.translate(to='en')
```

Definimos una condicional mayor a "0" para dar una ponderación positiva a los tweets extraídos.

```
if eng.sentiment.polarity > 0:
print("positivo:\n",line)
```

Definimos una condicional menor igual a "0" para dar una ponderación negativa a los tweets extraídos.

```
if eng.sentiment.polarity <= 0:
print("negativo:\n", line)
```

Aquí indicara la ponderación total de los tweets positivos y negativos.

```
print("Precisi n positiva = {}% con {} tweets".format(
    pos_correct / pos_count * 100.0, pos_count))
```

```
print("Precisi n negativa = {}% con {} tweets".format(
    neg_correct / neg_count * 100.0, neg_count))
```

Software y hardware utilizado

Diagrama de bloques del proceso

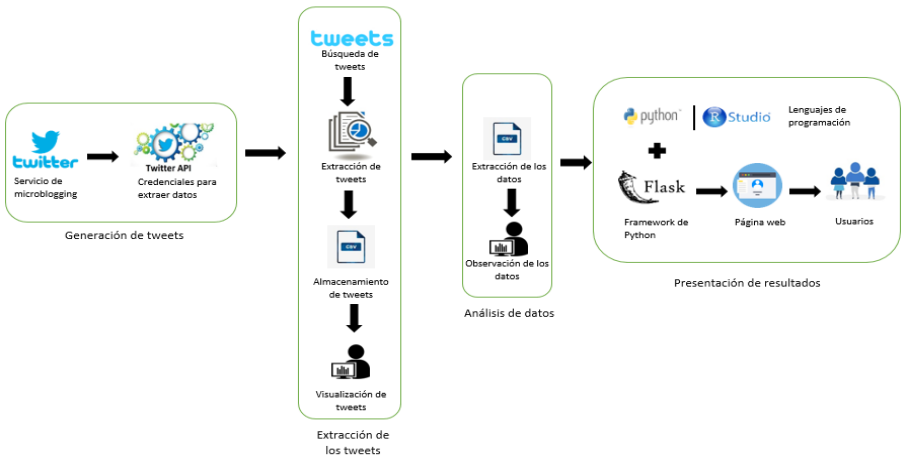

Fig. 14. Diagrama de Bloques

3 Experimentos

Gráficos

Nube de palabras

La nube de palabras es una representación visual de las palabras que conforman nuestro dataframe, donde se puede observar que para las palabras que aparecen con más frecuencia su tamaño es mayor al de las otras palabras.

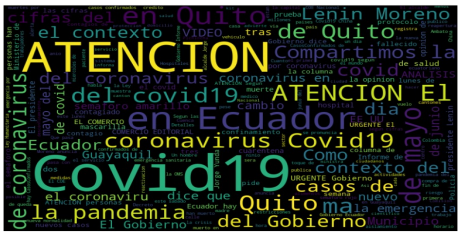

Fig. 15. Nube de palabras contenidas en el dataframe.

Tiempos de ejecución
Resultados en Terminal Python
Matriz de Confusión permite visualizar el desempeño del algoritmo que se esta ejecutando para el aprendizaje supervisado.

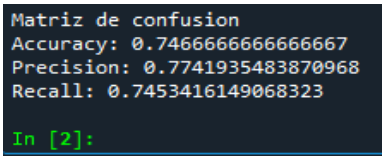

Fig. 16. Matriz de confusión en base al modelo de predicción.

Se observa una tasa de clasificación (Accuracy) del 74,6%, se considerada acpetable. La precisión del modelo (Precision) se refiere con qué frecuencia es correcta, cuando su modelo de regresión logística predice el 77,4% del tiempo. Recall: el modelo de regresión logística puede identificarlo el 74,5% del tiempo.

4 Conclusiones y Prospectivas

Conclusiones:

- Dentro de la minería de datos y aprendizaje automatizado las técnicas de clasificación son esenciales y representan aproximadamente el 70% de problemas de datascience.
- Uno de los algoritmos de aprendizaje de maquina mas simples es la regresión logística utilizados para la clasificación, ya que se puede usar como lineamiento para cualquier problema de clasificación o predicción.
- El Analisis de Sentimiento de Twitter permite conocer emociones que los usuarios de esta red social transmiten en sus publicaciones a traves de un analisis de polaridad demostrando una subjetividad en las opiniones expresadas en la plataforma.

– Con la librería de TextBlob se pudo analizar los sentimientos tanto como positivos y negativos, ayudando con la traducción de cualquier lenguaje y haciendo fácil su uso

Prospectivas:

– Al usar el conjunto de datos como el corpus para hacer un vector tf-idf, se debe utilizar la misma estructura vectorial para fines de training y test.

– Usando mejores técnicas de procesado de datos se puede obtener una mayor precisión del modelo.

5 Referencias

Proof. Para analizar los tweets tomamos como referencia a los articulos [3], donde la tarea primordial fue el procesamiento de datos [1], ademas de la documentacion en python proporcionada por [2, 4], que permitieron comprender temas necesarios para la eleboracion del proyecto.

References

1. Singh, T., Kumari, M. *Role of text pre-processing in twitter sentiment analysis.* [*Procedia Computer Science, 89(Supplement C)*] https://doi.org/549-554. (2016).
2. Pandarachalil, R., Sendhilkumar, S., Mahalakshmi, G. S. *Twitter sentiment analysis for large-scale data: an unsupervised approach.* [*Cognitive computation , 7(2)*] (2015). https://doi.org/254-262.
3. Kouloumpis, E., Wilson, T., Moore, J. *Twitter sentiment analysis: The good the bad and the omg!.* [*In Fifth International AAAI conference on weblogs and social media.*] (2011, July).
4. A. Deshwal and S. K. Sharma. *"Twitter sentiment analysis using various classification algorithms"* [*2016 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*] Noida, 2016, pp. 251-257, https://doi.org/10.1109/ICRITO.2016.7784960.