Corporación Universitaria Minuto de Dios

Semana 5: Análisis de sentimientos

Elaborado por:

Samir Romero Cárdenas

Procesamiento Natural Lenguaje NRC-50577

Especialización en Inteligencia Artificial

Tutora: Claudia Marcela Ospina Mosquera

12 de febrero de 2024

Contenido

Introducción

El análisis de sentimientos es una tarea crucial en el procesamiento del lenguaje natural que tiene como objetivo determinar la actitud general de un texto, ya sea positiva, negativa o neutral. En este informe, se realiza un análisis comparativo de sentimientos utilizando dos modelos diferentes: Naive Bayes y TextBlob, sobre un conjunto de datos que consiste en 5000 opiniones de películas. El objetivo es evaluar y comparar el rendimiento de ambos enfoques en la tarea de análisis de sentimientos.

Desarrollo

Naive Bayes:

Se utilizó el clasificador Naive Bayes, un modelo probabilístico simple pero efectivo para la clasificación de textos.

Se aplicó preprocesamiento de texto, que incluyó la eliminación de etiquetas HTML, conversión a minúsculas, eliminación de stopwords y stemming.

Se vectorizó el texto utilizando el método de Bolsa de Palabras (Bag of Words) utilizando la clase CountVectorizer.

Se dividió el conjunto de datos en conjuntos de entrenamiento y prueba.

Se entrenó el modelo Naive Bayes con el conjunto de entrenamiento y se evaluó su desempeño en el conjunto de prueba utilizando métricas como precisión, recall y F1-score.

TextBlob:

Se utilizó la librería TextBlob, una herramienta de procesamiento de lenguaje natural que proporciona una API sencilla para realizar análisis de sentimientos.

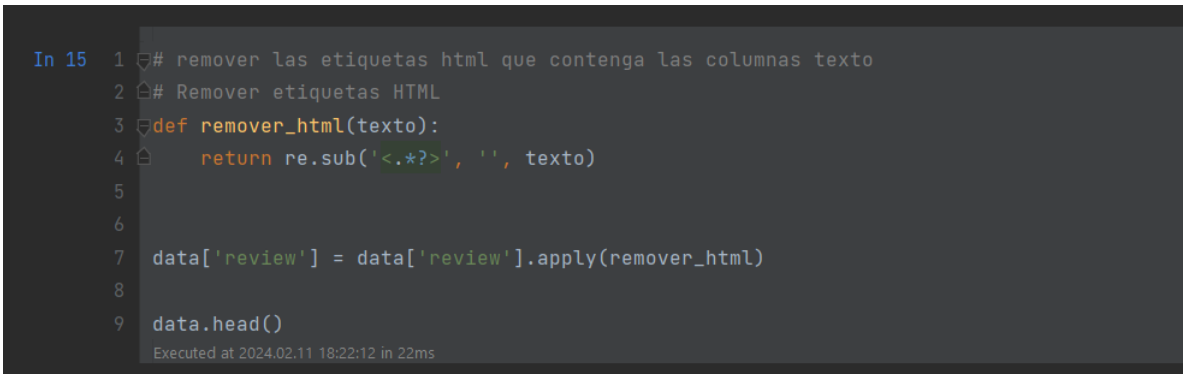
Se aplicó el análisis de sentimientos utilizando TextBlob, que calcula la polaridad del sentimiento de un texto.

Se clasificaron los sentimientos como positivos, negativos o neutros basándose en la polaridad calculada.

Se compararon las clasificaciones de sentimientos obtenidas con TextBlob con las etiquetas de sentimiento reales en el conjunto de datos.

Proceso de obtención y limpieza de datos.

Ilustración 1 Código para limpieza de datos.



```
In 15
# remover las etiquetas html que contenga las columnas texto
# Remover etiquetas HTML
def remover_html(texto):
    return re.sub('<.*?>', '', texto)


data['review'] = data['review'].apply(remover_html)

data.head()
Executed at 2024.02.11 18:22:12 in 22ms
```

Filtrado de palabras.

Se hace una división en palabras implementando la función **word_tokenize** de NLTK para dividir el texto en una lista de palabras, también se crea una lista de tokens que consisten únicamente en caracteres alfabéticos, eliminando números y otros caracteres especiales.

Ilustración 2 Word_tokenize.

```
In 16
# Utilizaremos una funcion para convertir el corpus en minuscula

data['review'] = data['review'].str.lower()
data.head()

Executed at 2024.02.11 18:22:13 in 27ms
```

Out 16 — 5 rows × 2 columns pd.Dataframe

	review	sentiment
0	i realli like thi summerslam due to the look of t…	positive
1	not mani televis show appeal to quit as mani diff…	positive
2	the film quickli get to major chase scene with ev…	negative
3	jane austen would definit approv of thi one ! gwy…	positive
4	expect were somewhat high for when i went to see …	negative

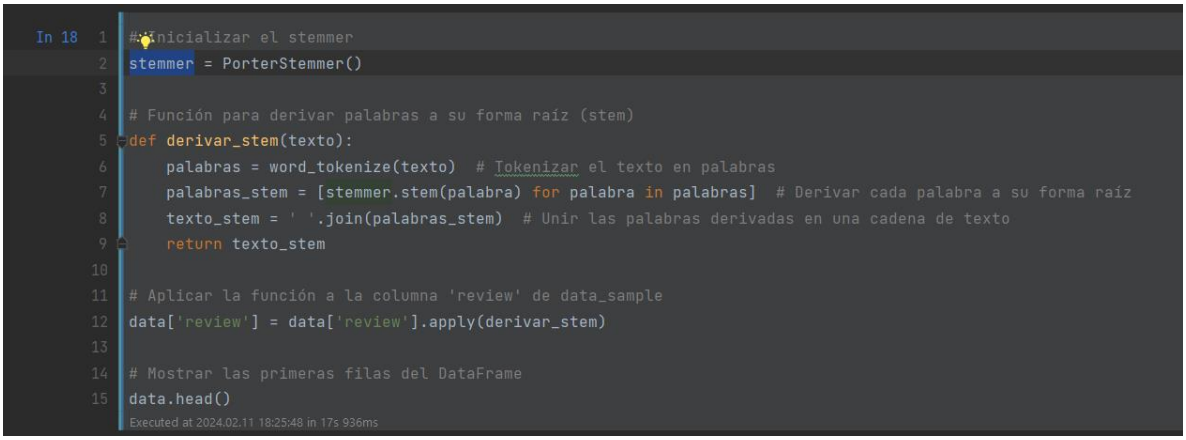
```
In 17
# eliminamos palabras vacias del corpus
# Remover stopwords
def remover_stopwords(texto):
    palabras = word_tokenize(texto)
    palabras_filtradas = [palabra for palabra in palabras if palabra.lower() not in stopwords.words('spanish')]
    return ' '.join(palabras_filtradas)

data['review'] = data['review'].apply(remover_stopwords)
data.head()
Executed at 2024.02.11 18:25:27 in 3m 13s 71ms
```

Stemmer.

Un stemmer es una herramienta utilizada en procesamiento de lenguaje natural (NLP) que se emplea para reducir las palabras a su forma raíz o "stem". El objetivo principal del stemming es reducir las variaciones de palabras a una forma base común para simplificar el análisis del texto.

Ilustración 3 Código de stemmer



Guardar corpus.

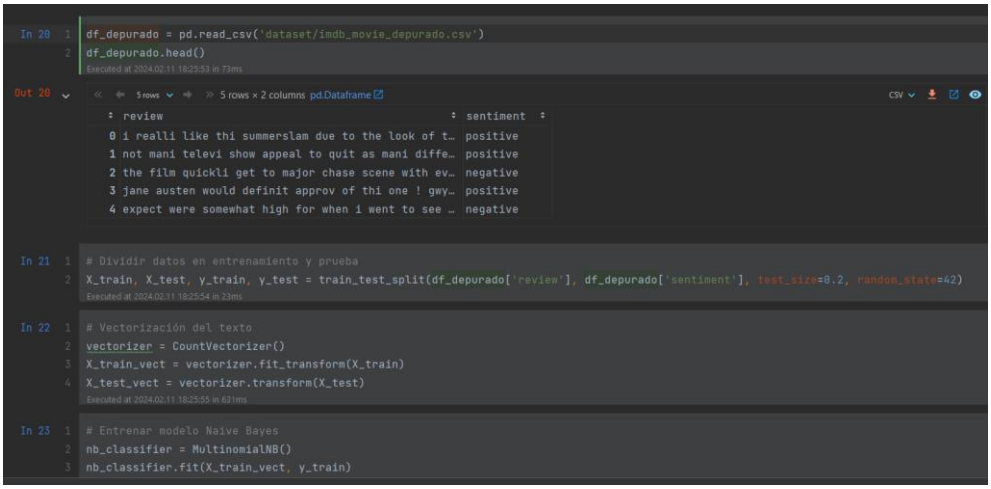
Después de la limpieza de datos en un corpus en el procesamiento de lenguaje natural (NLP), guardar el corpus limpio puede ser útil por varias razones: Responsabilidad, Eficiencia, Facilidad de acceso, Seguridad.

Ilustración 4 Código para guardar corpus

Naive Bayes.

El clasificador Naive Bayes es una técnica popular en el análisis de sentimientos debido a su simplicidad y eficacia. Funciona aplicando el teorema de Bayes con la suposición "ingenua" de independencia condicional entre cada par de características dado el valor de la variable de clase. En el análisis de sentimientos, las características podrían ser palabras o n-gramas, y la variable de clase podría ser el sentimiento (por ejemplo, positivo o negativo).

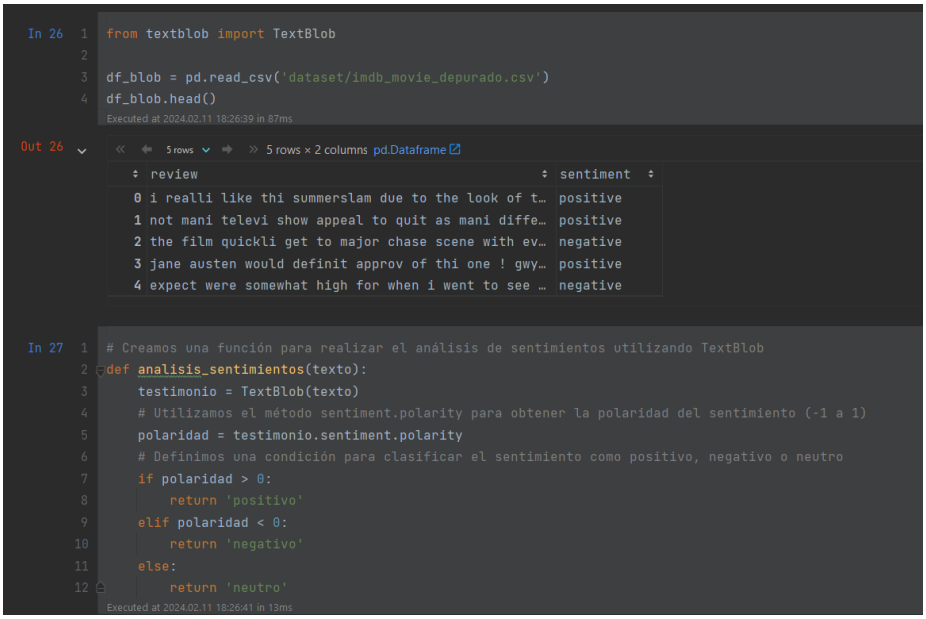
Ilustración 5 Código Implementando Naive Bayes.



TextBlob.

TextBlob es una biblioteca de procesamiento de lenguaje natural (NLP) en Python que proporciona una interfaz simple para realizar tareas comunes de NLP, como tokenización, análisis de sentimientos, análisis de sustantivos y nombres propios, traducción de texto, etc. Utiliza NLTK (Natural Language Toolkit) y Pattern, dos bibliotecas populares de NLP en Python, como su base.

Ilustración 6 Código implementando TextBlob



Resultados

Los resultados obtenidos en el análisis de sentimientos con Naive Bayes y TextBlob se presentan a continuación:

Naive Bayes

Ilustración 7 Precisión del modelo:

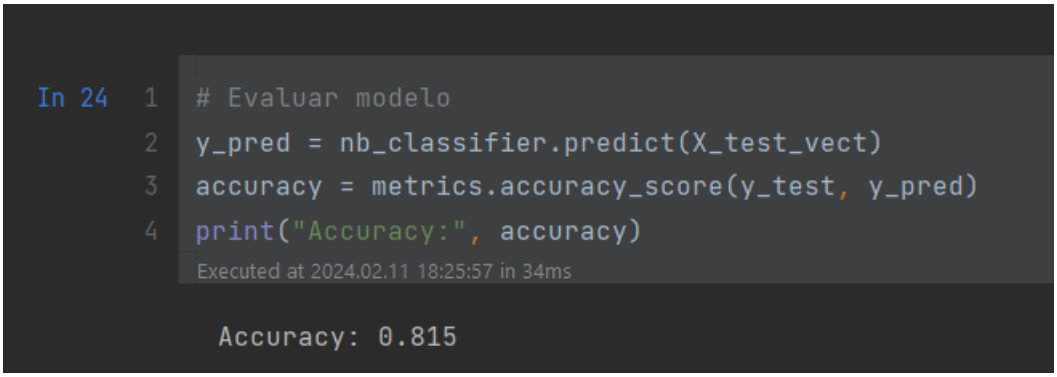
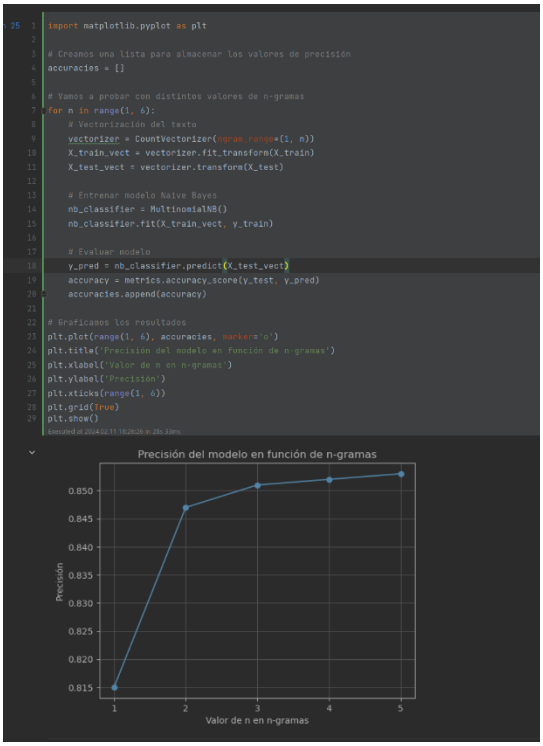


Ilustración 8 Gráfico de Precisión vs. n-gramas:



TextBlob

Ilustración 9 Resultados de clasificación de sentimientos:

```
In 29  1  # Mostramos las primeras filas del DataFrame con la nueva columna de sentimiento
       2  print(df_blob[['review', 'sentimiento_textblob']].head())
          Executed at 2024.02.11 18:27:39 in 50ms

                                              review sentimiento_textblob
   0  i realli like thi summerslam due to the look o...            positivo
   1  not mani televi show appeal to quit as mani di...            positivo
   2  the film quickli get to major chase scene with...            positivo
   3  jane austen would definit approv of thi one ! ...            positivo
   4  expect were somewhat high for when i went to s...            positivo
```

Ilustración 10 metrica de TextBlob

```
Métricas de clasificación para el análisis de sentimientos con TextBlob:
              precision    recall  f1-score   support

    negative       0.00      0.00      0.00    2481.0
    negativo       0.00      0.00      0.00       0.0
      neutro       0.00      0.00      0.00       0.0
    positive       0.00      0.00      0.00    2519.0
    positivo       0.00      0.00      0.00       0.0

    accuracy                           0.00    5000.0
   macro avg       0.00      0.00      0.00    5000.0
```

Conclusiones

Ambos enfoques han arrojado resultados interesantes. Mientras que el modelo Naive Bayes logró una precisión de aproximadamente el 81.5%, el análisis de sentimientos con TextBlob no fue tan exitoso, mostrando una precisión baja y una matriz de confusión que indica que la clasificación no fue significativa.

Una posible explicación de estos resultados podría ser la complejidad de las reseñas de películas y la capacidad de cada modelo para comprender el contexto y el tono de las mismas. Naive Bayes, al ser un modelo de aprendizaje supervisado, pudo haber capturado patrones más complejos en los datos de entrenamiento, lo que resultó en una mejor precisión en la clasificación de sentimientos. Por otro lado, TextBlob, al basarse en reglas heurísticas y en la polaridad de las palabras, puede no haber sido tan efectivo en este caso específico.

Una ventaja de TextBlob es su facilidad de uso y la capacidad de manejar análisis de sentimientos con un mínimo de configuración. Sin embargo, Naive Bayes proporciona más flexibilidad y control sobre el proceso de modelado, lo que puede ser beneficioso en escenarios donde se requiere un ajuste fino del modelo.

Para futuras investigaciones, sería interesante explorar enfoques más avanzados de aprendizaje automático y técnicas de procesamiento de lenguaje natural para mejorar aún más el rendimiento en el análisis de sentimientos de grandes conjuntos de datos de reseñas de películas.

Referencia

Srinivasa, B. (2018). Chapter 13. Deep learning for Text. En: Natural language processing and computational linguistics: A practical guide to text analysis with Python, Gensim, Spacy, and Keras. https://ebookcentral.proquest.com/lib/bibliouniminuto-ebooks/detail.action?docID=5446034&pq-origsite=summon

Campesato, O. (2021). Chapter 4. Algorithms and Toolkits (I). En: Natural language processing fundamentals for developers https://ebookcentral.proquest.com/lib/bibliouniminuto-ebooks/reader.action?docID=6647713

Ganegedara, T. (2018). Chapter 2: Understanding TensorFlow. En: Natural language processing with TensorFlow: Teach language to machines using python's deep learning library https://ebookcentral.proquest.com/lib/bibliouniminuto-ebooks/detail.action?docID=5405681&pq-origsite=summon

Casas, J., Lozano, T., y Bosch, A. (2019). Parte II: Redes neuronales artificiales. En: Deep learning: Principios y fundamentos https://login.microsoftonline.com/b1ba85eb-a253-4467-9ee8-d4f8ed4df300/saml2